

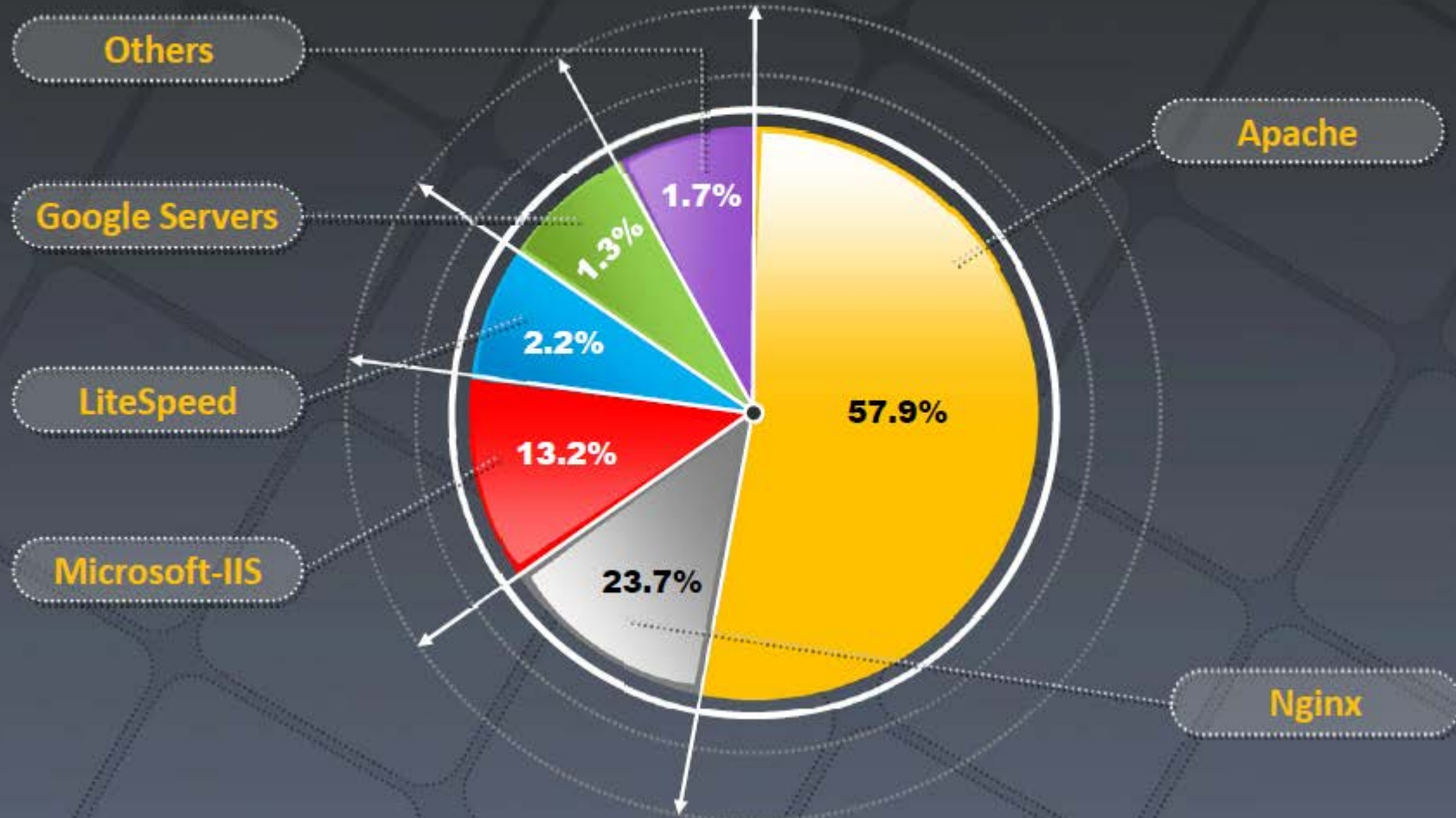
Hacking Webservers

Module 11

Unmask the **Invisible Hacker**.



Webserver Market Shares



<http://w3techs.com>

Module Objectives

- Understanding Webserver Concepts
- Understanding Webserver attacks
- Understanding Webserver Attack Methodology
- Webserver Attack Tools



- Countermeasures against Webserver Attacks
- Overview of Patch Management
- Webserver Security Tools
- Overview of Webserver Penetration Testing



Module Flow

CEH
Certified Ethical Hacker



**Webserver
Concepts**

1



**Webserver
Attacks**

2



**Attack
Methodology**

3



**Webserver
Attack Tools**

4



**Counter-
measures**

5



**Patch
Management**

6



**Webserver
Security Tools**

7

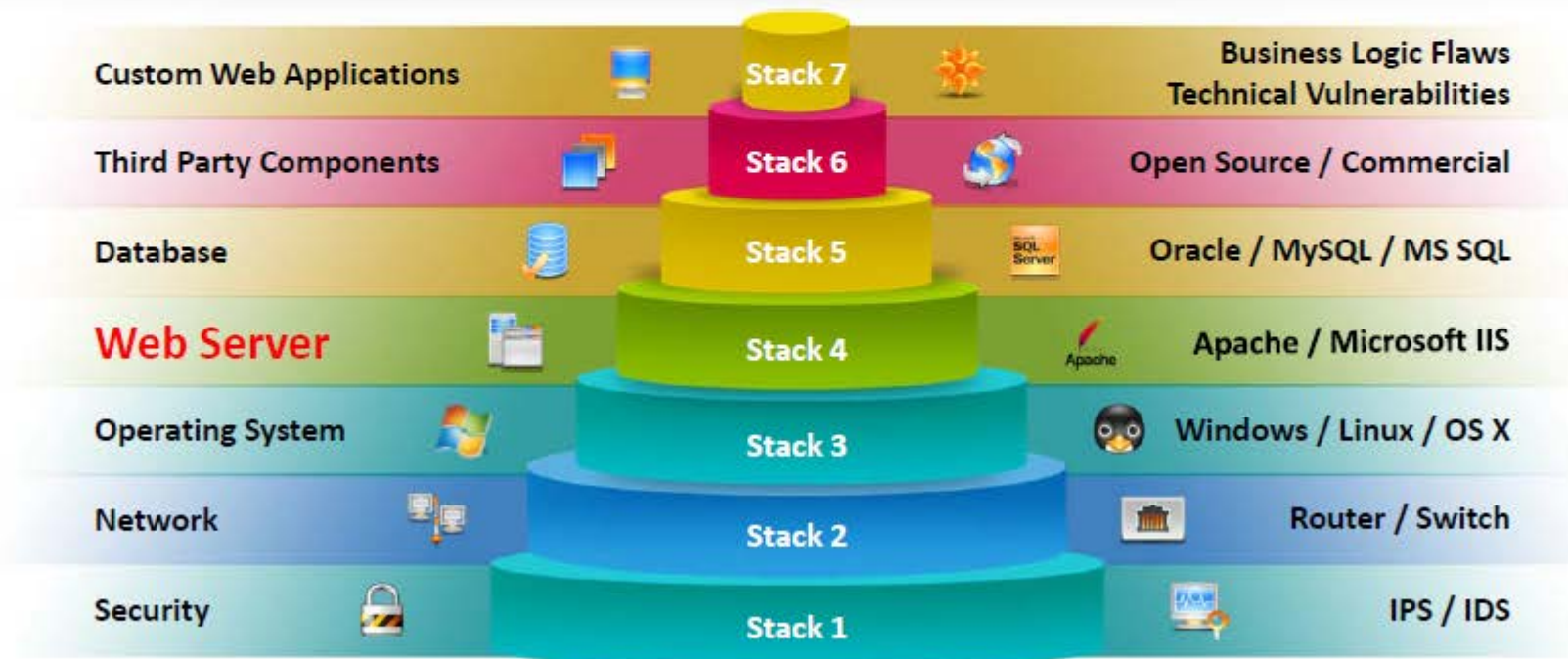


**Webserver
Pen Testing**

8

Web Server Security Issue

- Web server is a program (both hardware and software) that hosts websites; attackers usually target **software vulnerabilities** and configuration errors to compromise web servers
- Nowadays, **network** and **OS level attacks** can be well defended using proper network security measures such as firewalls, IDS, etc., however, web servers are accessible from anywhere on the web, which makes them **less secured** and **more vulnerable** to attacks



Why Web Servers Are **Compromised**



➔ **Improper** file and directory **permissions**

➔ Installing the server with **default settings**

➔ **Unnecessary services** enabled, including content management and remote administration

➔ **Security conflicts** with business ease-of-use case

➔ **Lack of proper security policy**, procedures, and maintenance

➔ **Improper authentication** with external systems

➔ **Default accounts** with their default or no passwords

➔ **Unnecessary** default, backup, or sample **files**

➔ **Misconfigurations** in web server, operating systems, and networks

➔ **Bugs** in server software, OS, and web applications

➔ **Misconfigured SSL certificates** and encryption settings

➔ Administrative or **debugging functions** that are **enabled** or accessible on web servers

➔ Use of **self-signed certificates** and default certificates

Impact of **Webserver Attacks**

01

Compromise of user accounts



02

Website defacement

03

Secondary attacks from the Website

04

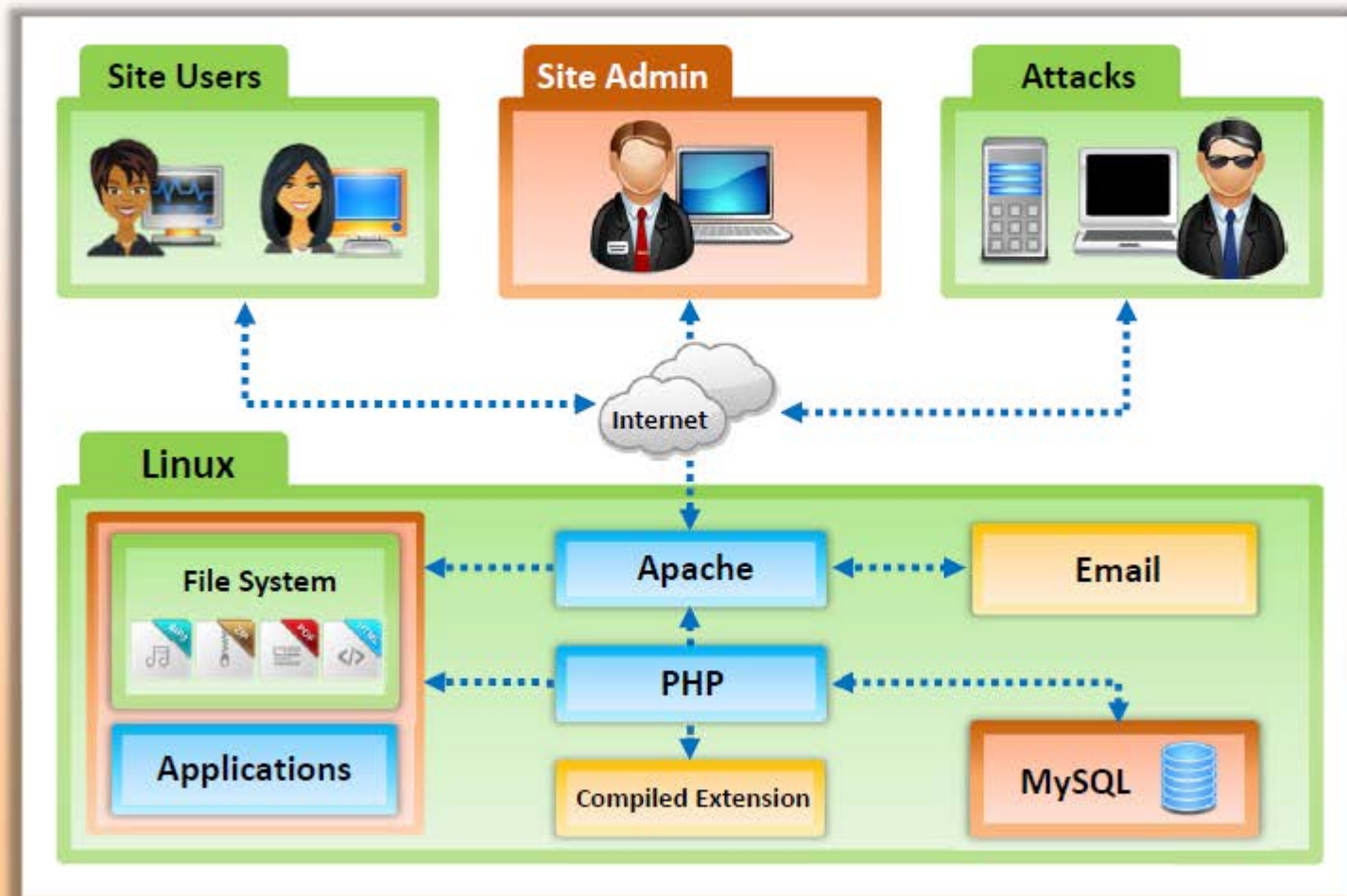
Root access to other applications or servers

05

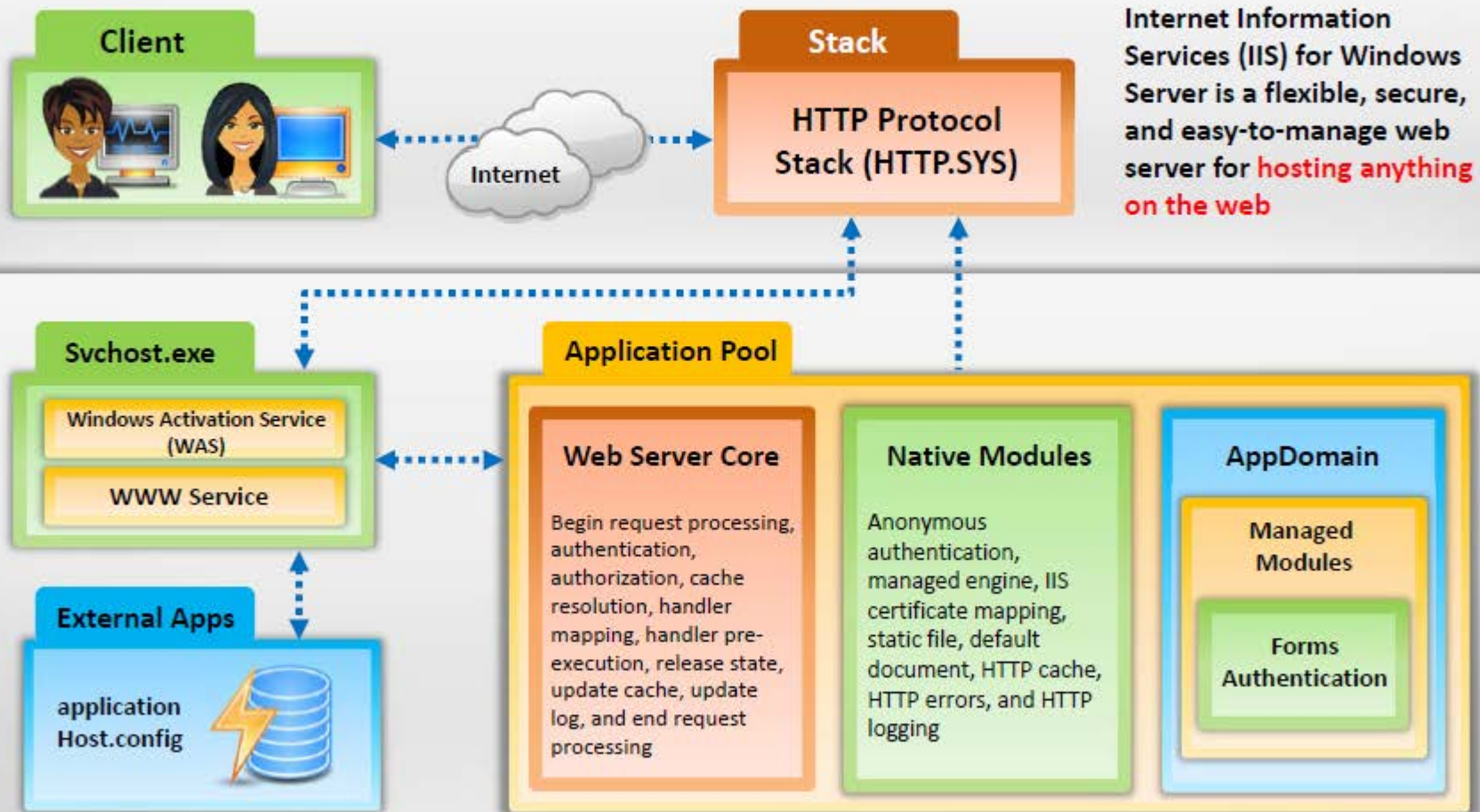
Data tampering and data theft



Open Source Webserver Architecture



IIS Web Server Architecture



Module Flow

CEH
Certified Ethical Hacker



**Webserver
Concepts**

1



**Webserver
Attacks**

2



**Attack
Methodology**

3



**Webserver
Attack Tools**

4



**Counter-
measures**

5



**Patch
Management**

6



**Webserver
Security Tools**

7



**Webserver
Pen Testing**

8

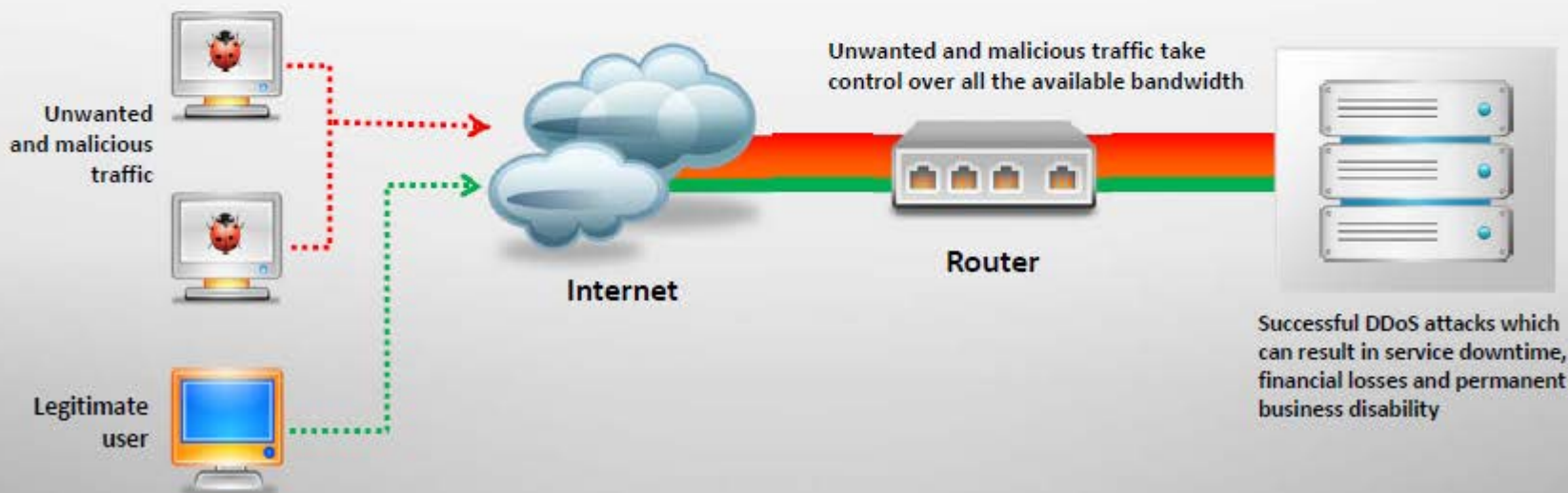
DoS/DDoS Attacks



Attackers may send numerous **fake requests** to the web server which results in the **web server crash** or become unavailable to the legitimate users

VISA

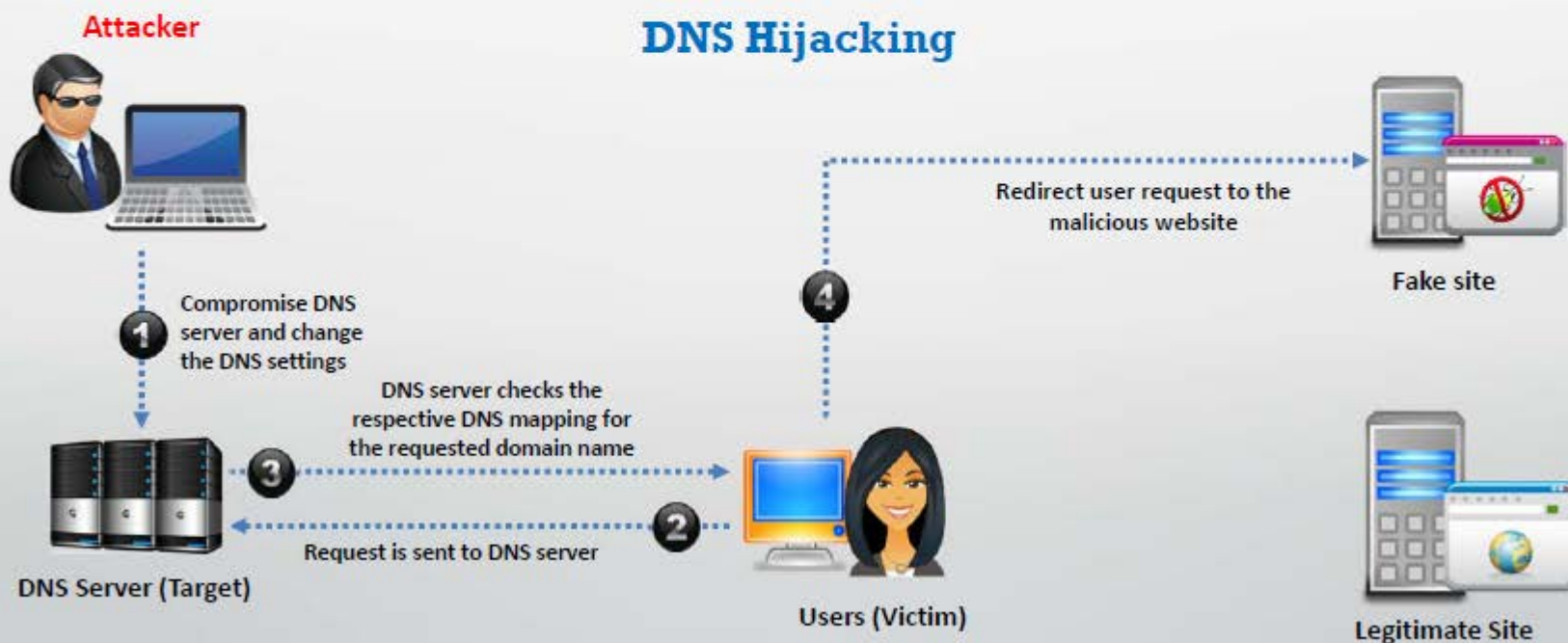
Attackers may target **high profile web servers** such as banks, credit card payment gateways, government owned services, etc. to **steal user credentials**



DNS Server Hijacking



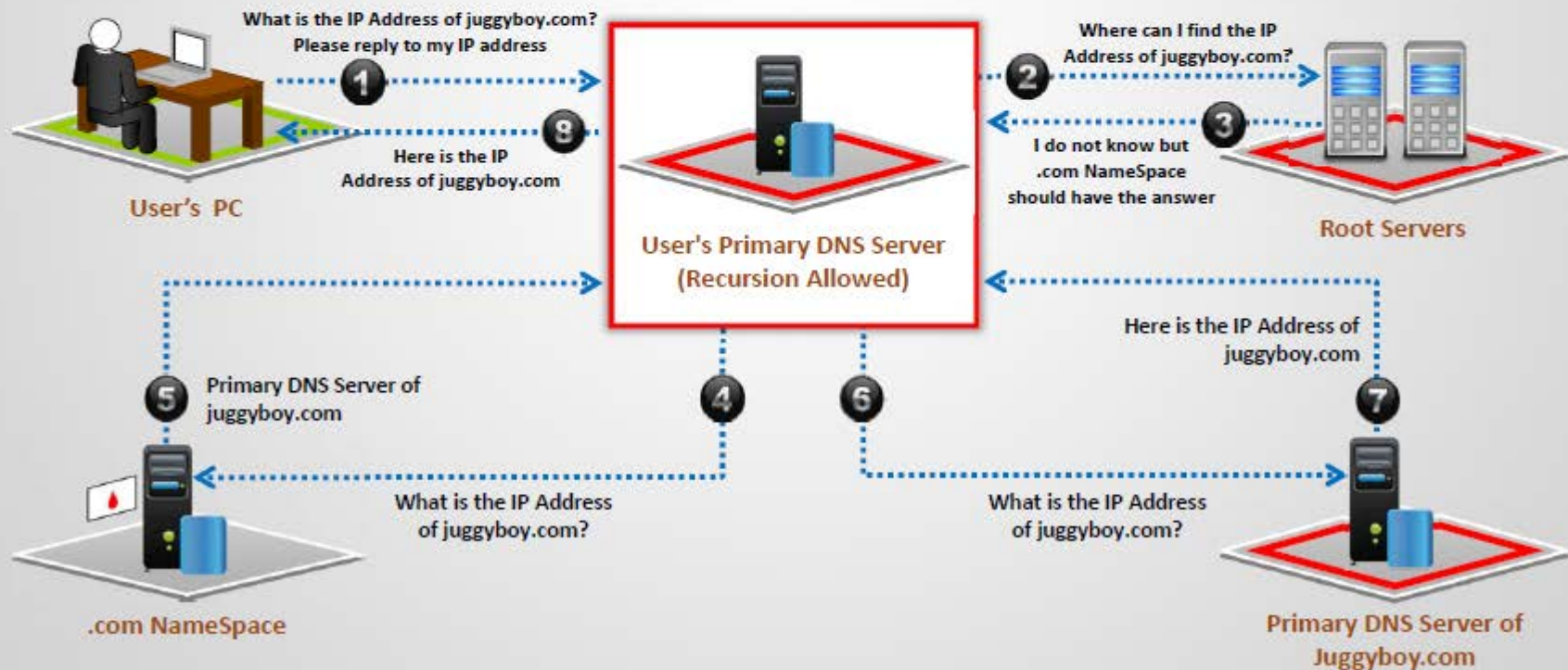
Attacker compromises DNS server and **changes the DNS settings** so that all the request coming toward the target web server should be redirected to his/her own malicious server



DNS Amplification Attack

Attacker takes the advantage of **DNS recursive method** of DNS redirection to perform DNS amplification attack

Recursive DNS Method



Directory Traversal Attacks

In directory traversal attacks, attackers use **../ (dot-dot-slash)** sequence to access restricted directories outside of the web server root directory

Attackers can use **trial and error method** to navigate the outside of root directory and access sensitive information in the system

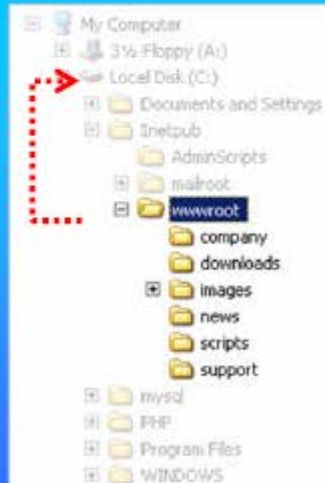


`http://server.com/scripts/..%5c../Windows/System32/cmd.exe?/c+dir+c:\`

Volume in drive C has no label.
Volume Serial Number is D45E-9FEE

Directory of C:\

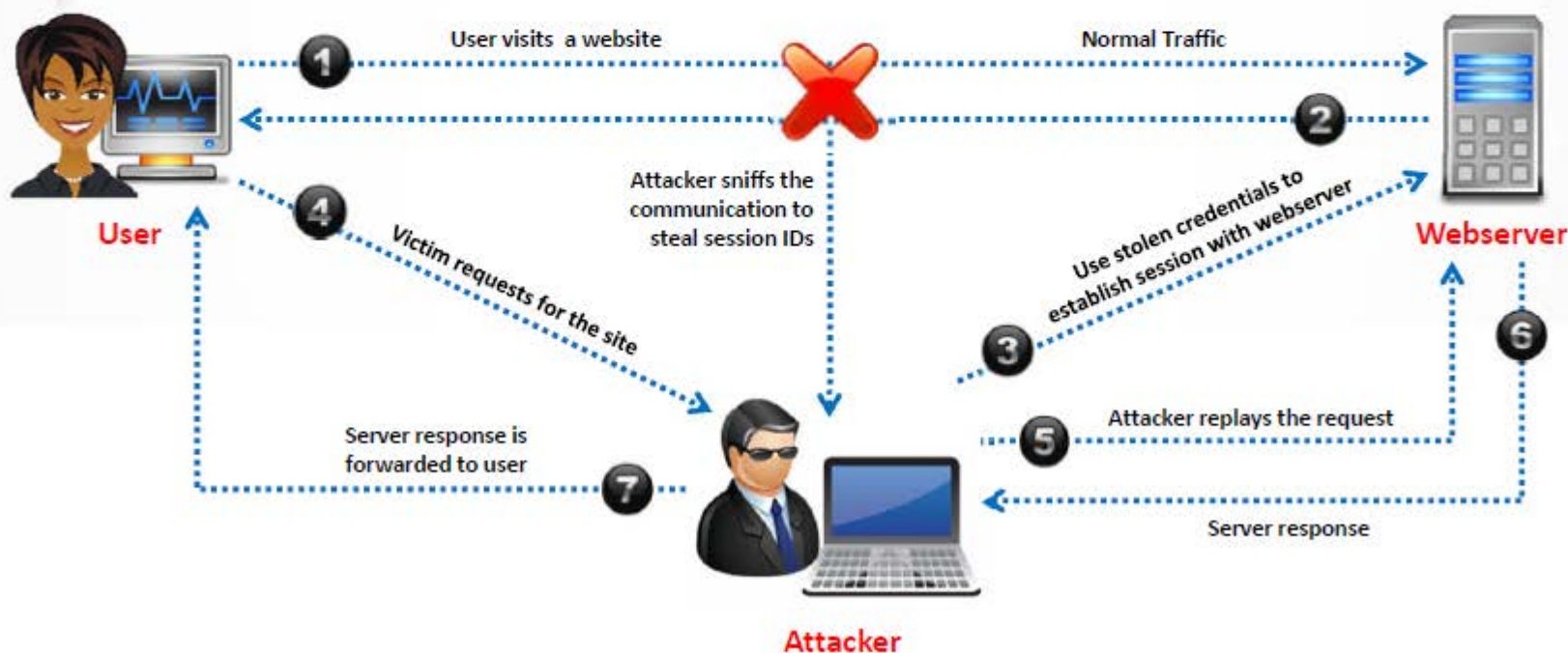
06/02/2013 11:31 AM	1,024 .rnd
09/28/2013 06:43 PM	0 123.text
05/21/2013 03:10 PM	0 AUTOEXEC.BAT
09/27/2013 08:54 PM	<DIR> CATALINA_HOME
05/21/2013 03:10 PM	0 CONFIG.SYS
08/11/2013 09:16 AM	<DIR> Documents and Settings
09/25/2013 05:25 PM	<DIR> Downloads
08/07/2013 03:38 PM	<DIR> Intel
09/27/2013 09:36 PM	<DIR> Program Files
05/26/2013 02:36 AM	<DIR> Snort
09/28/2013 09:50 AM	<DIR> WINDOWS
09/25/2013 02:03 PM	569,344 WinDump.exe
7 File(s) 570,368 bytes	
13 Dir(s) 13,432,115,200 bytes free	



Man-in-the-Middle/Sniffing Attack

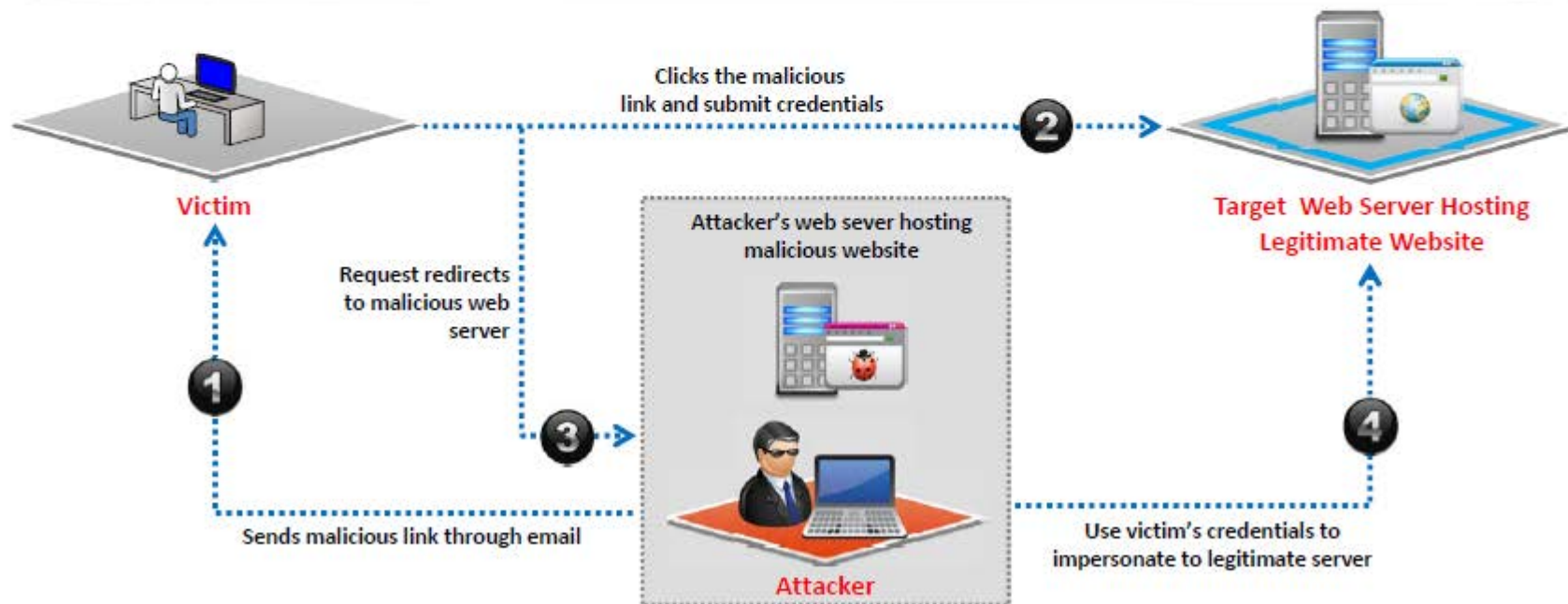
01 Man-in-the-Middle (MITM) attacks allow an attacker to access sensitive information by **intercepting and altering communications** between an end-user and webserver

02 Attacker **acts as a proxy** such that all the communication between the user and webserver passes through him



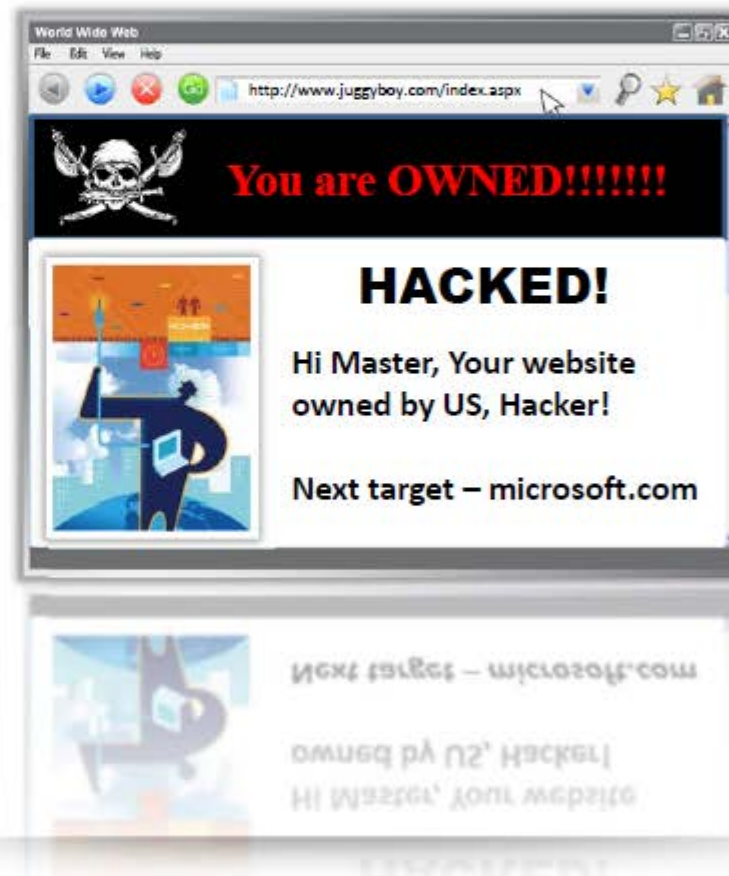
Phishing Attacks

- Attacker tricks user to submit **login details** for website that looks legitimate, but it redirect to the malicious website hosted on attacker web server
- Attacker **steals the credentials** entered and use it to impersonate with the website hosted on the legitimate target server
- Attacker then can perform **unauthorized** or **malicious operation** with the website target server



Website Defacement

- Web defacement occurs when an intruder **maliciously alters visual appearance of a web page** by inserting or substituting provocative and frequently offending data
- **Defaced pages exposes visitors to some propaganda** or misleading information until the unauthorized change is discovered and corrected
- Attackers use variety of methods such as **MYSQL injection** to access a site in order to deface it



Web Server Misconfiguration



Server misconfiguration refers to **configuration weaknesses** in web infrastructure that can be exploited to launch various attacks on web servers such as directory traversal, server intrusion, and data theft



Verbose Debug/Error Messages

Anonymous or Default Users/Passwords

Sample Configuration, and Script Files

Remote Administration Functions

Unnecessary Services Enabled

Misconfigured/Default SSL Certificates

Web Server Misconfiguration Example

This configuration allows anyone to view the **server status** page, which contains detailed information about the current use of the web server, including information about the **current hosts** and requests being processed



httpd.conf file on an **Apache** server

```
<Location /server-status>  
SetHandler server-status  
</Location>
```

This configuration gives **verbose error messages**



php.ini file

```
display_error = On  
log_errors = On  
error_log = syslog  
ignore_repeated_errors = Off
```


HTTP Response Splitting Attack



HTTP response splitting attack involves **adding header response data into the input field** so that the server split the response into two responses



The attacker can **control the second response to redirect user to a malicious website** whereas the other responses will be discarded by web browser

Server Code

```
String author =  
request.getParameter(AUTHOR_PA  
RAM);  
...  
Cookie cookie = new  
Cookie("author", author);  
cookie.setMaxAge(cookieExpirat  
ion);  
response.addCookie(cookie);
```

Input = Jason

```
HTTP/1.1 200 OK  
...  
Set-Cookie: author=Jason  
...
```

Input = JasonTheHacker\r\nHTTP/1.1 200 OK\r\n

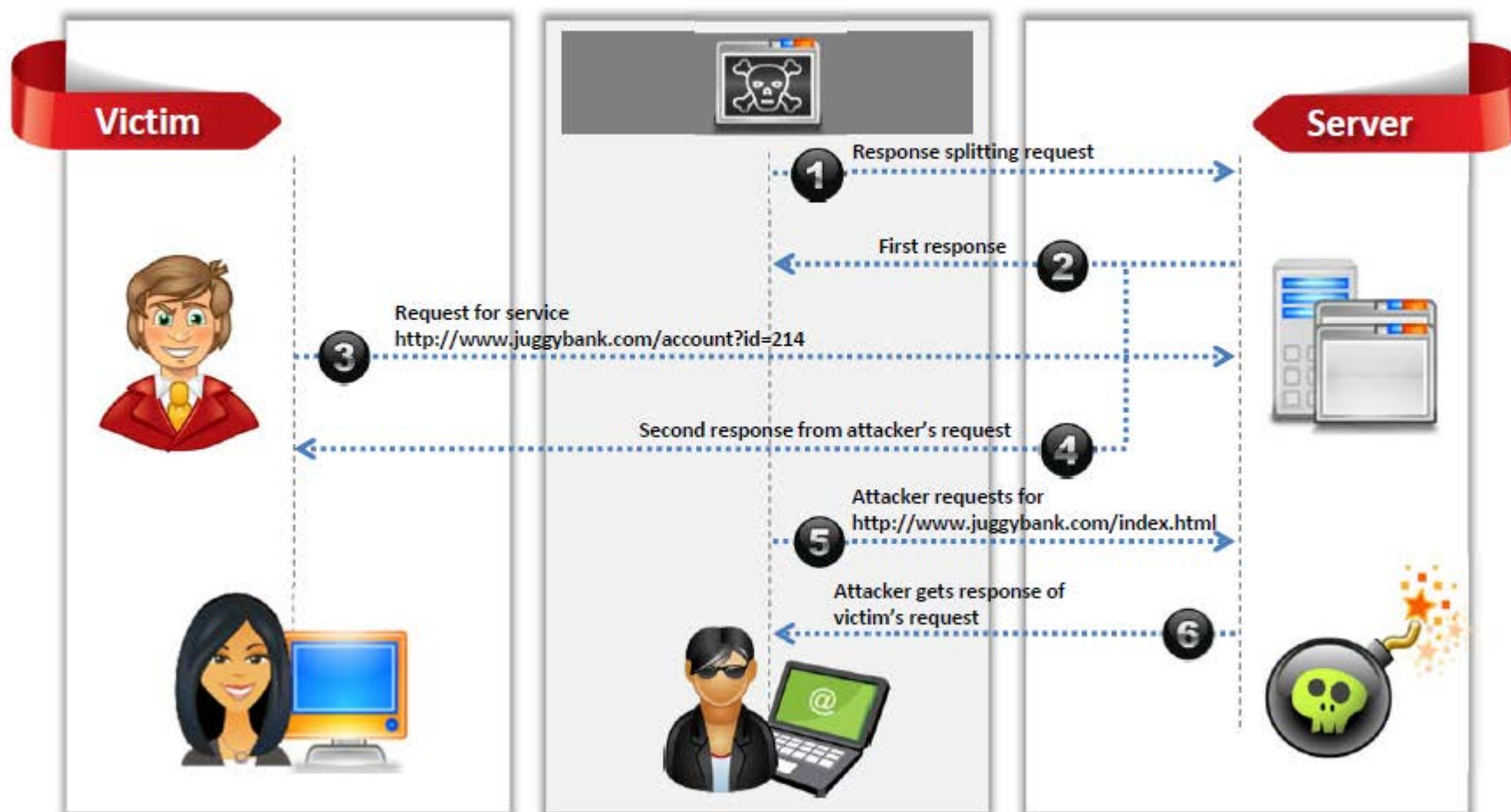
First Response (Controlled by Attacker)

```
Set-Cookie: author=JasonTheHacker  
HTTP/1.1 200 OK  
...
```

Second Response

```
HTTP/1.1 200 OK  
...
```


HTTP Response Splitting Attack (Cont'd)



Web Cache Poisoning Attack

CEH
Certified Ethical Hacker

Attacker



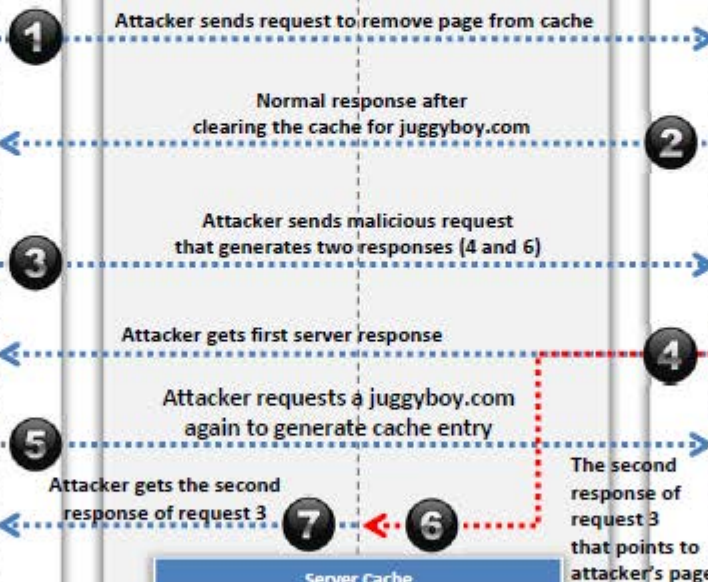
```
GET http://juggyboy.com/index.html
HTTP/1.1
Pragma: no-cache
Host: juggyboy.com
.....
Accept-Charset: iso-8859-1,*,utf-8
```

```
GET http://juggyboy.com/
redir.php?site=%0d%0aContent-
Length:%20%0d%0a%0d%0aHTTP/1.1%2
0200%20OK%0d%0aLast-
Modified:%20Mon,%2027%20Oct%20200
9%2014:50:18%20GMT%0d%0aConte nt-
Length:%20%0d%0aContent-
Type:%20text/html%0d%0a%0d%0a<html
>Attack Page</html> HTTP/1.1
.....
Host: juggyboy.com
```

```
GET
http://juggyboy.com/index.html
HTTP/1.1 Host: testsite.com
User-Agent: Mozilla/4.7 [en]
(WinNT; I)
.....
Accept-Charset: iso-8859-1,*,utf-8
```

Server Cache	
Address	Page
www.juggyboy.com	Original Juggyboy page

Server Cache



Server Cache	
Address	Page
www.juggyboy.com	Attacker's page

Poisoned Server Cache

Server



```
http://www.juggyboy.com/wel
come.php?lang=
<?php header ("Location: " .
$_GET['page']); ?>
```

An attacker forces the web server's cache to **flush its actual cache content** and sends a specially **crafted request**, which will be stored in cache

SSH Bruteforce Attack

1

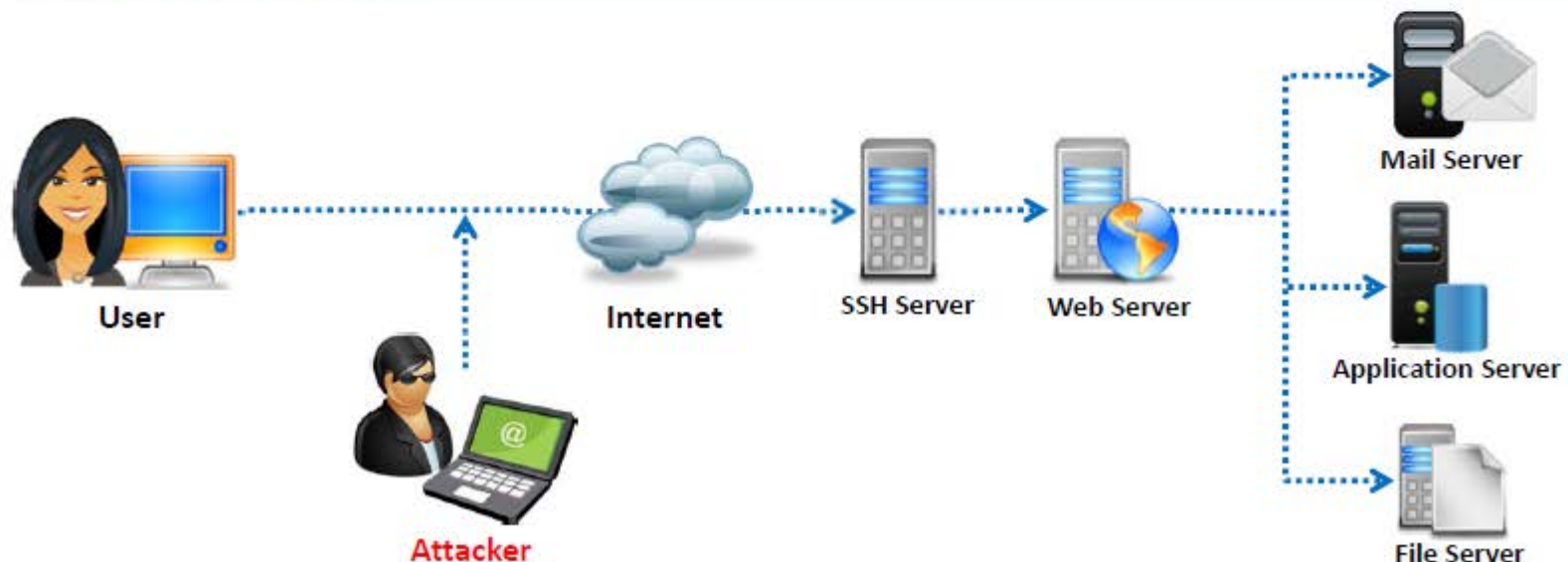
SSH protocols are used to create an **encrypted SSH tunnel** between two hosts in order to transfer unencrypted data over an insecure network

2

Attackers can brute force SSH login credentials to gain **unauthorized access to a SSH tunnel**

3

SSH tunnels can be used to **transmit malwares** and other exploits to victims without being detected



Webserver Password Cracking



An attacker tries to exploit weaknesses to hack **well-chosen passwords**



The most **common passwords** found are password, root, administrator, admin, demo, test, guest, qwerty, pet names, etc.



Attacker target mainly for:

- SMTP servers
- Web shares
- SSH Tunnels
- Web form authentication cracking
- FTP servers



Attackers use different methods such as **social engineering, spoofing, phishing**, using a Trojan Horse or virus, wiretapping, keystroke logging, etc.



Many hacking attempts start with **cracking passwords** and proves to the webserver that they are a **valid user**

Webserver Password Cracking Techniques



- Passwords may be cracked **manually** or with **automated tools** such as Cain & Abel, Brutus, THC Hydra, etc.
- Passwords can be cracked by using following techniques:



Guessing

A common cracking method used by attackers to guess passwords either by **humans** or by **automated tools** provided with dictionaries

Dictionary Attacks

A **file of words is run against user accounts**, and if the password is a simple word, it can be found pretty quickly

Brute Force Attack

The most time-consuming, but comprehensive way to crack a password. Every **combination of character is tried** until the password is broken.

Hybrid Attack

A hybrid attack works similar to dictionary attack, but it adds **numbers** or **symbols** to the password attempt

Web Application Attacks



Vulnerabilities in **web applications** running on a webserver provide a broad attack path for webserver compromise



Parameter/Form
Tampering



Cookie
Tampering



Unvalidated Input and
File Injection Attacks



SQL
Injection
Attacks



Session
Hijacking



Directory
Traversal



Denial-of-
Service (DoS)
Attack



Cross-Site Scripting
(XSS) Attacks



Buffer
Overflow
Attacks



Cross-Site Request
Forgery (CSRF)
Attack

Note: For complete coverage of web application attacks refer to Module 12: Hacking Web Applications

Module Flow



**Webserver
Concepts**

1



**Webserver
Attacks**

2



**Attack
Methodology**

3



**Webserver
Attack Tools**

4



**Counter-
measures**

5



**Patch
Management**

6



**Webserver
Security Tools**

7



**Webserver
Pen Testing**

8

Webserver Attack Methodology



**Information
Gathering**

01

**Webserver
Footprinting**



02



**Mirroring
Website**

03

**Vulnerability
Scanning**



04



**Session
Hijacking**

05

**Hacking
Webserver
Passwords**



06

Webserver Attack Methodology: Information Gathering



1

Information gathering involves collecting information about the **targeted company**

2

Attackers search the **Internet, newsgroups, bulletin boards**, etc. for information about the company

3

Attackers use **Whois, Traceroute, Active Whois**, etc. tools and query the Whois databases to get the details such as a domain name, an IP address, or an autonomous system number

WHOIS.netTM
Your Domain Starting Place...

Search box containing 'ebay.com' and a green 'GO' button.

WHOIS information for ebay.com:***

```
[Querying whois.verisign-grs.com]
[whois.verisign-grs.com]
Whois Server Version 2.0
Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.
Domain Name: EBAY.COM
Registrar: MARKMONITOR, INC.
Whois Server: whois.markmonitor.com
Referral URL: http://www.markmonitor.com
Name Server: NS1.P47.DYNECT.NET
Name Server: SJC-DNS1.EBAYDNS.COM
Name Server: SJC-DNS2.EBAYDNS.COM
Name Server: SMF-DNS1.EBAYDNS.COM
Name Server: SMF-DNS2.EBAYDNS.COM
Status: clientDeleteProhibited
Status: clientTransferProhibited
Status: clientUpdateProhibited
Status: serverDeleteProhibited
Status: serverTransferProhibited
Status: serverUpdateProhibited
Updated Date: 29-oct-2013
Creation Date: 04-aug-1995
Expiration Date: 03-aug-2018
<<
```

<http://www.whois.net>

Note: For complete coverage of information gathering techniques refer to Module 02: Footprinting and Reconnaissance

Webserver Attack Methodology:

Information Gathering from Robots.txt File

- The robots.txt file contains the **list of the web server directories and files** that the web site owner wants to hide from web crawlers
- Attacker can simply request Robots.txt file from the URL and retrieve the sensitive information such as **root directory structure, content management system information**, etc., about the target website



```
robots - Notepad
File Edit Format View Help
User-agent: *
Disallow: /wp-admin/
Disallow: /wp-includes/
Disallow: /*/download/confirmation.aspx?
Disallow: /ctl/
Disallow: /admin/
Disallow: /App_Browsers/
Disallow: /genuine/ajax/
Disallow: /App_Code/
Disallow: /App_Data/
Disallow: /App_GlobalResources/
Disallow: /bin/
Disallow: /Components/
Disallow: /Config/
Disallow: /contest/
Disallow: /genuine/survey/
Disallow: /controls/
Disallow: /DesktopModules/
Disallow: /HttpModules/
Disallow: /Install/
Disallow: /js/
Disallow: /software
Disallow: /software.aspx
Disallow: /windows/404.aspx?*|
Disallow: /Userlogin
Disallow: /testgallery
Sitemap: http://www.juggyboy.com/sitemap.xml
```


Webserver Attack Methodology: Webserver Footprinting



01

Gather **valuable system-level data** such as account details, operating system, software versions, server names, and database schema details

02

Telnet a webserver to footprint a webserver and gather information such as server name, server type, operating systems, applications running, etc.

03

Use tool such as **ID Serve**, **httprecon**, and **Netcraft** to perform footprinting



NETCRAFT

Search Web by Domain

Explore 1,472,431 web sites visited by users of the Netcraft Toolbar 1st November 2013

Search: search tips

example: site contains .netcraft.com

Results for microsoft

First 500 sites returned

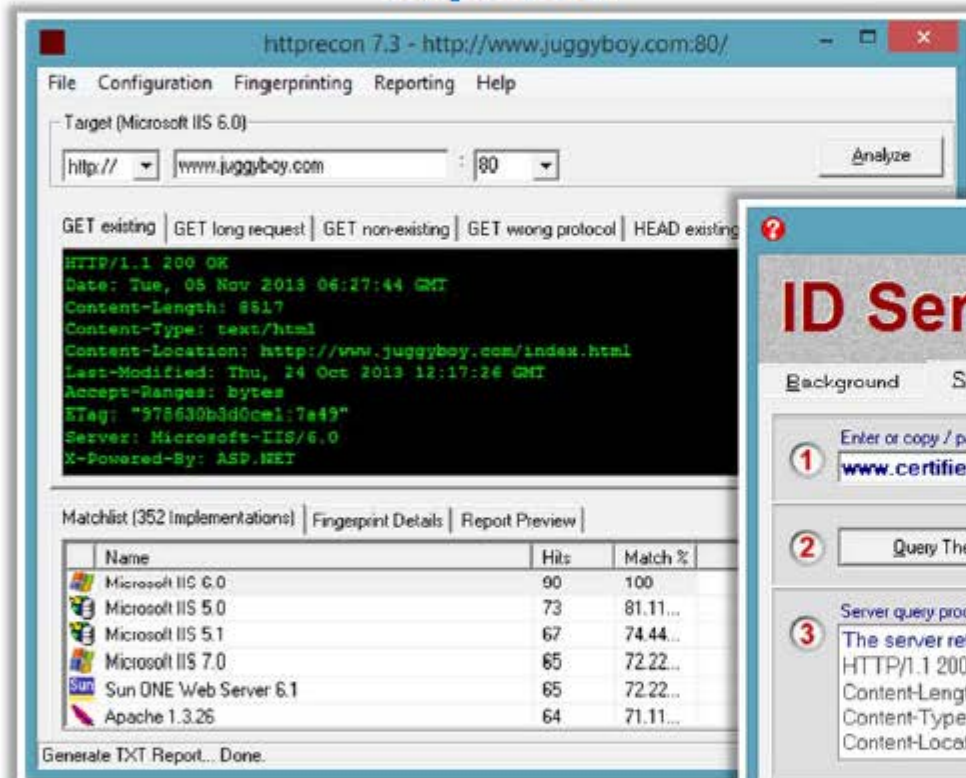
Site	Site Report	First seen	Netblock	OS
1. www.microsoft.com		august 1995	ms hotmail	citrix netcaler
2. go.microsoft.com		november 2001	ms hotmail	windows server 2008
3. support.microsoft.com		october 1997	microsoft corporation	unknown
4. technet.microsoft.com		august 1999	microsoft corporation	windows server 2012
5. windows.microsoft.com		june 1998	microsoft corporation	unknown
6. msdn.microsoft.com		september 1998	microsoft corporation	windows server 2012
7. social.technet.microsoft.com		august 2008	microsoft corporation	citrix netcaler
8. office.microsoft.com		november 1998	microsoft corporation	windows server 2008
9. answers.microsoft.com		august 2009	microsoft limited	windows server 2008
10. social.msdn.microsoft.com		august 2008	microsoft corporation	citrix netcaler
11. download.microsoft.com		august 1996	akamai international, bv	linux
12. login.microsoftonline.com		december 2010	microsoft corporation	windows server 2008
13. www.microsoftstore.com		november 2008	digital river ireland ltd.	FS big-ip
14. search.microsoft.com		january 1997	akamai technologies	linux
15. o15.officeedge.microsoft.com		may 2012	microsoft corporation	windows server 2008
16. www.update.microsoft.com		may 2007	microsoft corporation	windows server 2008
17. r.office.microsoft.com		november 2003	microsoft corporation	windows server 2008

<http://toolbar.netcraft.com>

Webserver Footprinting Tools



httprecon



<http://www.compute.ch>

ID Serve



<http://www.grc.com>

Enumerating Webserver Information Using Nmap

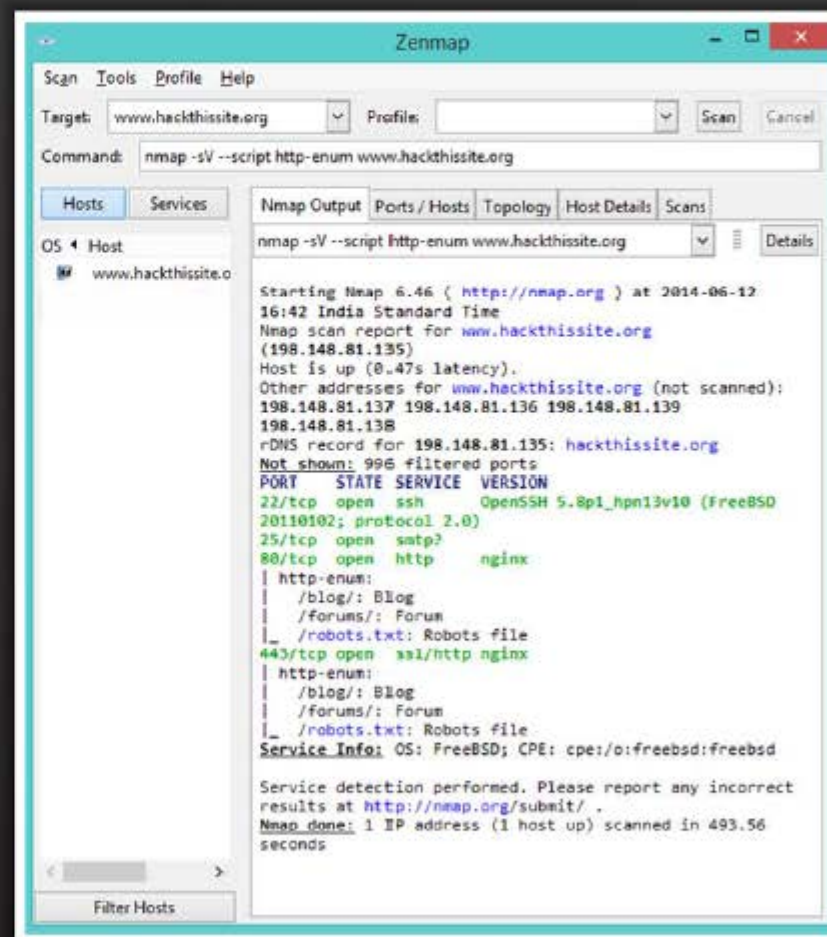
1 Attackers can use advanced **Nmap commands** and **Nmap Scripting Engine (NSE) scripts** to enumerate information about the target website

2 `nmap -sV -O -p target IP address`

3 `nmap -sV --script=http-enum target IP address`

4 `nmap target IP address -p 80 --script=http-frontpage-login`

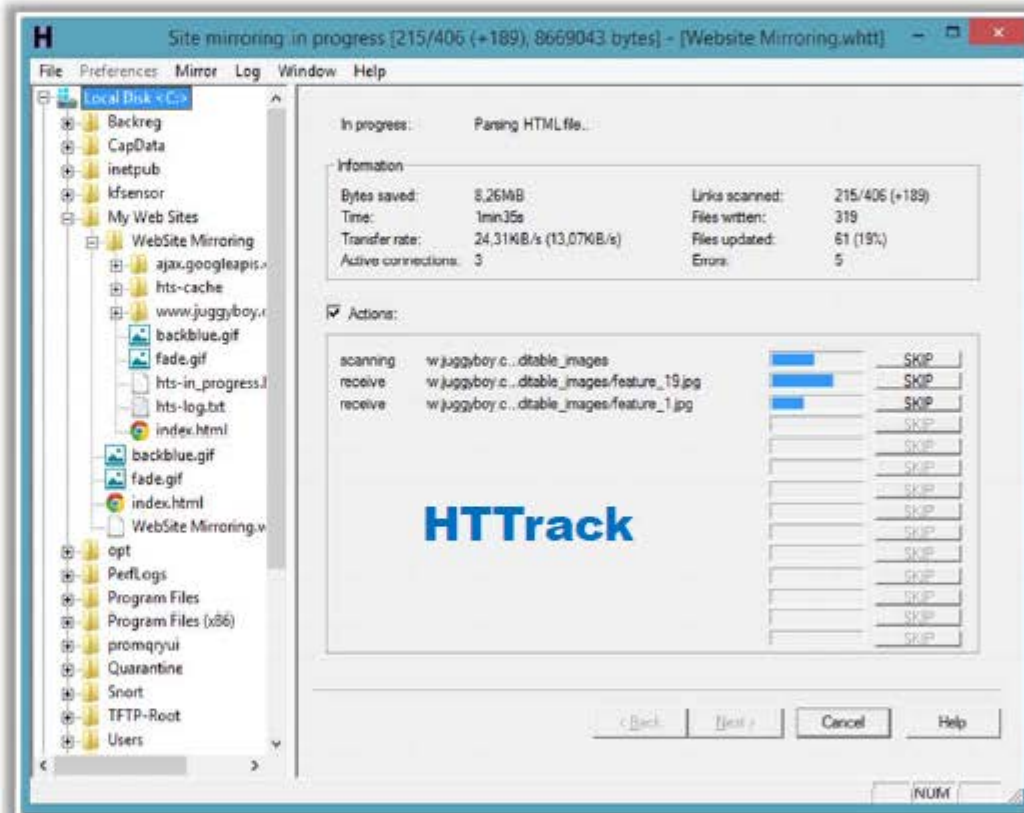
5 `nmap --script=http-passwd --script-args=http-passwd.root=/ target IP address`



<http://nmap.org>

Webserver Attack Methodology: Mirroring a Website

- Mirror a website to create a complete profile of the site's **directory structure, files structure, external links**, etc.
- Search for comments and other items in the **HTML source code** to make footprinting activities more efficient
- Use tools **HTTrack**, **WebCopier Pro**, **BlackWidow**, etc. to mirror a website



<http://www.httrack.com>

Webserver Attack Methodology: Vulnerability Scanning



01

Implement vulnerability scan to **identify weaknesses** in a network and determine if the system can be exploited

02

Use vulnerability scanners such as HP WebInspect, Acunetix Web Vulnerability Scanner, etc. to find **hosts**, **services**, and **vulnerabilities**

03

Sniff the network traffic to find out **active systems**, **network services**, **applications**, and vulnerabilities present

04

Test the **web server infrastructure** for any misconfigurations, outdated content, and vulnerabilities

Webserver Attack Methodology: Session Hijacking



1

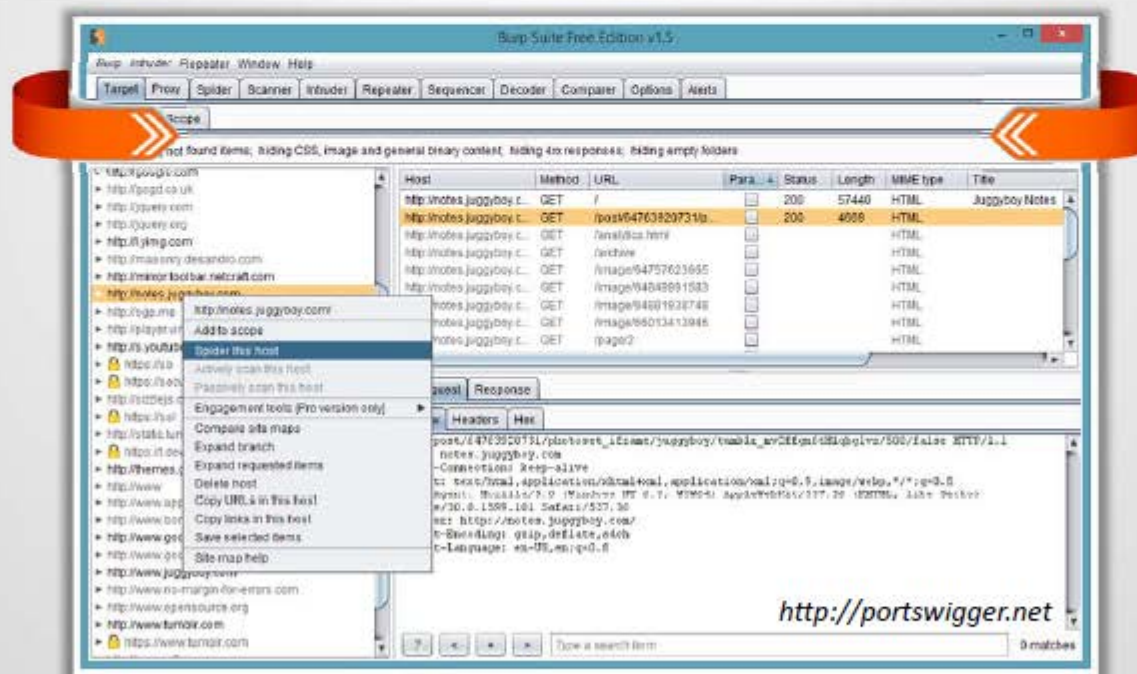
Sniff valid session IDs to **gain unauthorized access** to the Web Server and snoop the data

2

Use session hijacking techniques such as session fixation, session sidejacking, Cross-site scripting, etc. to **capture valid session cookies and IDs**

3

Use tools such as **Burp Suite**, **Firesheep**, **JHijack**, etc. to automate session hijacking



Note: For complete coverage of Session Hijacking concepts and techniques refer to Module 10: Session Hijacking

Webserver Attack Methodology: Hacking Web Passwords



Use password cracking techniques such as **brute force attack, dictionary attack**, password guessing to crack webserver passwords

Use tools such as **THC-Hydra, Brutus**, etc.

The screenshot shows the HydraGTK application window. The 'Output' tab is selected, displaying the following text:

```
Hydra v4.1 (c) 2004 by van Hauser / THC - use allowed only for legal purposes.  
Hydra (http://www.thc.org) starting at 2004-05-17 21:58:52  
[DATA] 32 tasks, 1 servers, 45380 login tries (l:1/p:45380), ~1418 tries per task  
[DATA] attacking service ftp on port 21  
[STATUS] 14056.00 tries/min, 14056 tries in 00:01h, 31324 todo in 00:03h  
[STATUS] 14513.00 tries/min, 29026 tries in 00:02h, 16354 todo in 00:02h  
[21][ftp] host: 127.0.0.1 login: marc password: success  
Hydra (http://www.thc.org) finished at 2004-05-17 22:01:38  
<finished>
```

At the bottom of the window, the command being executed is visible in the status bar:

```
hydra 127.0.0.1 ftp -l marc -P /tmp/passlist.txt -e ns -t 32
```

<https://www.thc.org>

Module Flow



**Webserver
Concepts**

1



**Webserver
Attacks**

2



**Attack
Methodology**

3



**Webserver
Attack Tools**

4



**Counter-
measures**

5



**Patch
Management**

6



**Webserver
Security Tools**

7



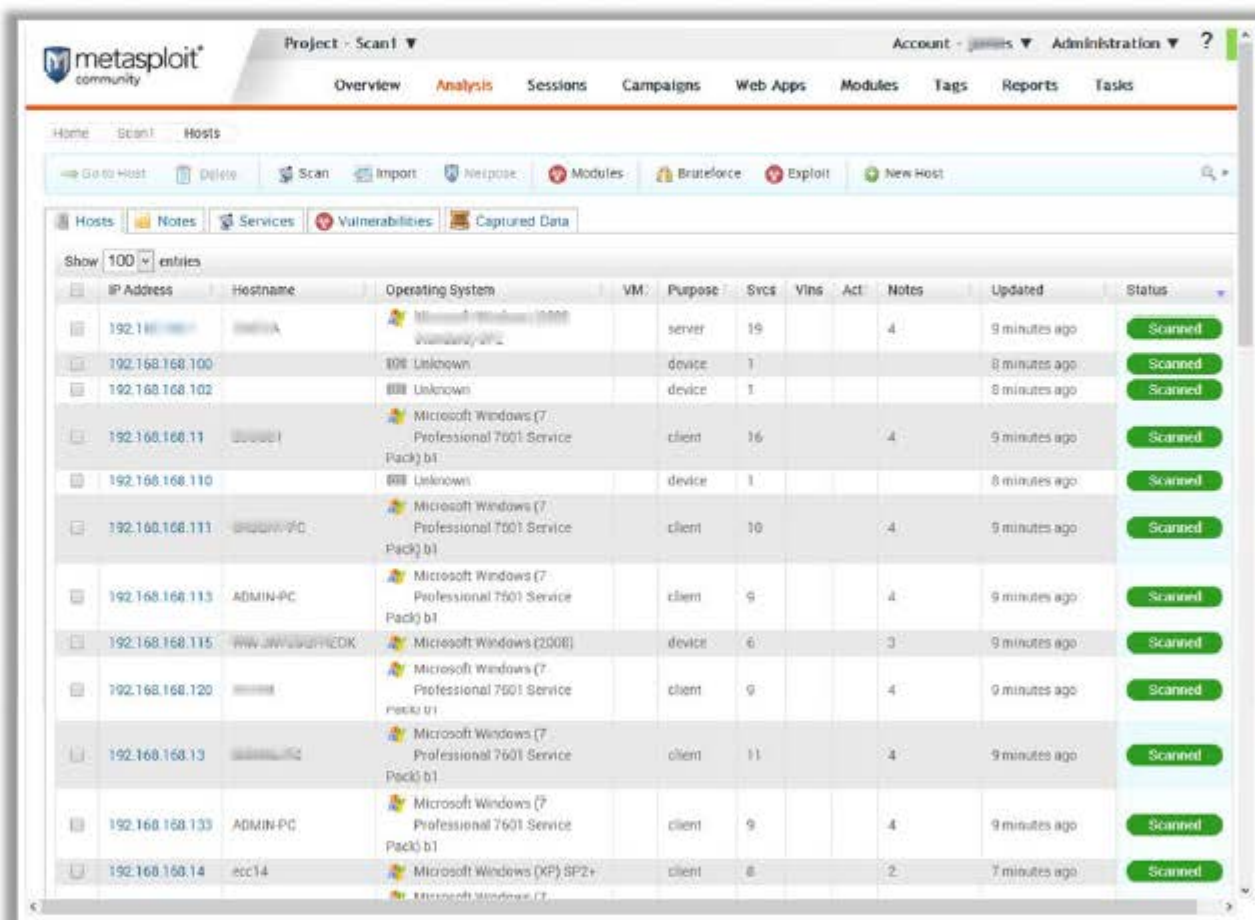
**Webserver
Pen Testing**

8

Webserver Attack Tool: Metasploit

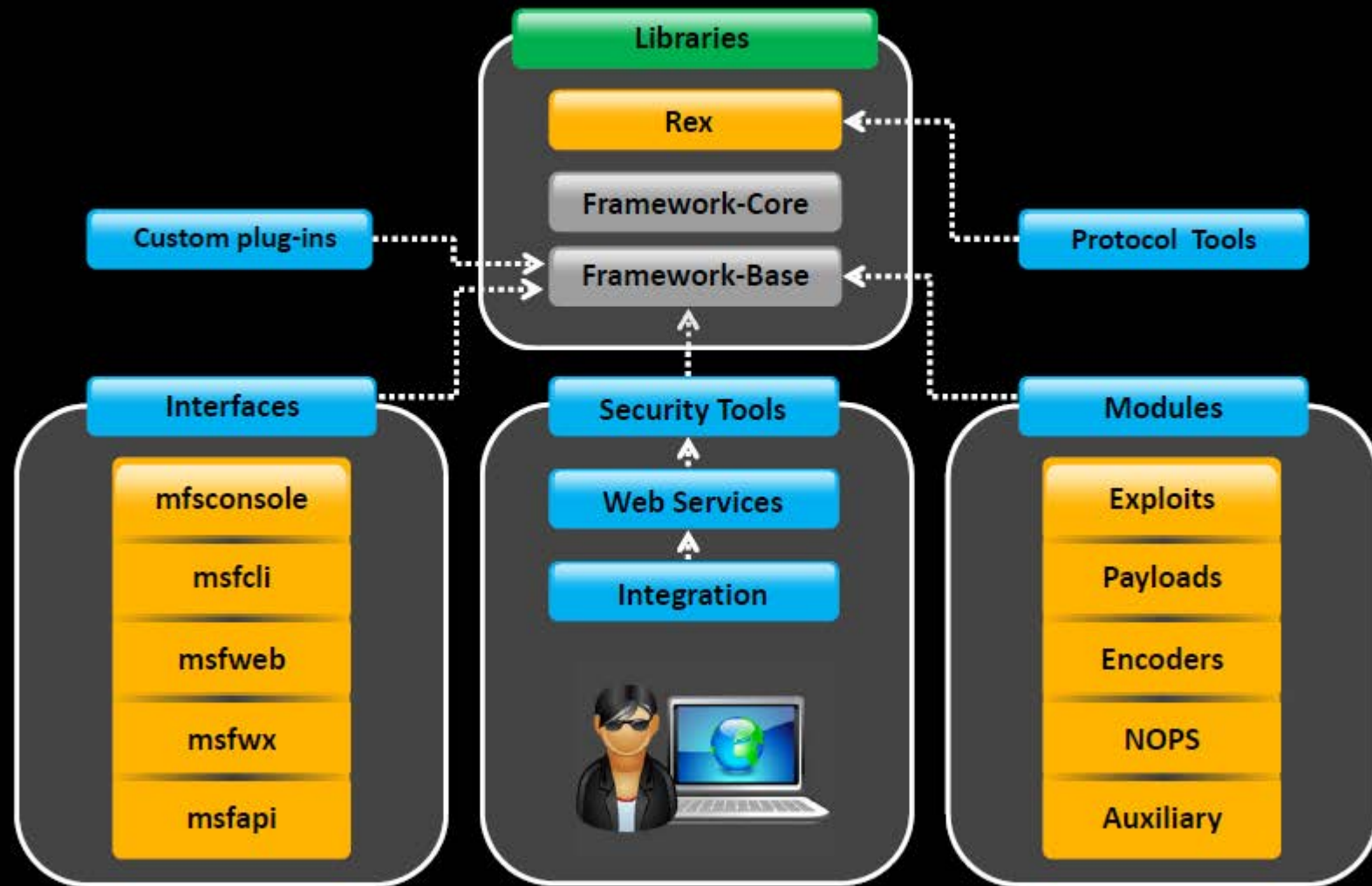


- The Metasploit Framework is a **penetration testing toolkit**, exploit development platform, and **research tool** that includes hundreds of working remote exploits for a variety of platforms
- It supports fully automated **exploitation of web servers**, by abusing known vulnerabilities and leveraging weak passwords via Telnet, SSH, HTTP, and SNMP



<http://www.metasploit.com>

Metasploit Architecture



Metasploit Exploit Module

- It is the basic module in Metasploit used to **encapsulate an exploit** using which users target many platforms with a single exploit
- This module comes with **simplified meta-information fields**
- Using a Mixins feature, users can also **modify exploit behavior dynamically**, brute force attacks, and attempt passive exploits



Steps to exploit a system follow the Metasploit Framework

- 1 Configuring Active Exploit
- 2 Verifying the Exploit Options
- 3 Selecting a Target
- 4 Selecting the Payload
- 5 Launching the Exploit

Metasploit Payload Module

- Payload module establishes a **communication channel** between the Metasploit framework and the victim host
- It combines the **arbitrary code** that is executed as the result of an exploit succeeding
- To generate **payloads**, first select a payload using the command:



```
C:\ Command Prompt

msf > use windows/shell_reverse_tcp

msf payload(shell_reverse_tcp) > generate -h

Usage: generate [options]

Generates a payload.

OPTIONS:

-b <opt> The list of characters to avoid:
'\x00\xff'

-e <opt> The name of the encoder module to use.

-h Help banner.

-o <opt> A comma separated list of options in
VAR=VAL format.

-s <opt> NOP sled length.

-t <opt> The output type: ruby, perl, c, or raw.

msf payload(shell_reverse_tcp) >
```


Metasploit Auxiliary Module



- Metasploit's auxiliary modules can be **used to perform arbitrary**, one-off actions such as port scanning, denial of service, and even fuzzing
- To run auxiliary module, either use the **run** command, or use the **exploit** command



Command Prompt

```
msf > use dos/windows/smb/ms06_035_mailslot
msf auxiliary(ms06_035_mailslot) > set RHOST 1.2.3.4
RHOST => 1.2.3.4
msf auxiliary(ms06_035_mailslot) > run
[*] Mangling the kernel, two bytes at a time...
```



Metasploit NOPS Module

- NOP modules generate a no-operation instructions used for blocking out buffers
 - Use **generate** command to generate a NOP sled of an arbitrary size and display it in a given format
- OPTIONS:

-b <opt>: The list of characters to avoid: '\x00\xff'
-h: Help banner
-s <opt>: The comma separated list of registers to save
-t <opt>: The output type: ruby, perl, c, or raw
msf nop(opty2) >



Generates a NOP sled of a given length

```
msf > use x86/opty2
msf nop(opty2) > generate -h
Usage: generate [options] length
```



Command to generate a 50 byte NOP sled

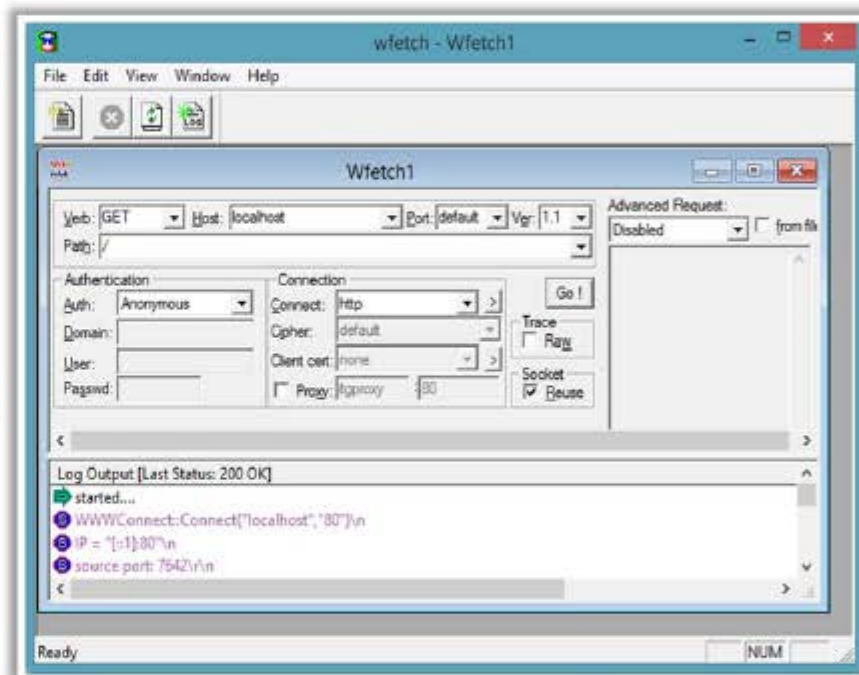
```
msf nop(opty2) > generate -t c 50
unsigned char buf[] =
"\xf5\x3d\x05\x15\xf8\x67\xba\x7d\x08\xd6\x
66\x9f\xb8\x2d\xb6"
"\x24\xbe\xb1\x3f\x43\x1d\x93\xb2\x37\x35\x
84\xd5\x14\x40\xb4"
"\xb3\x41\xb9\x48\x04\x99\x46\xa9\xb0\xb7\x
2f\xfd\x96\x4a\x98"
"\x92\xb5\xd4\x4f\x91";
msf nop(opty2) >
```


Webserver Attack Tool: Wfetch

CEH
Certified Ethical Hacker

WFetch allows attacker to fully customize an **HTTP request** and send it to a Web server to see the raw HTTP request and response data

It allows attacker to test the performance of Web sites that contain new elements such as **Active Server Pages (ASP)** or wireless protocols



<http://www.microsoft.com>



fully customize HTTP request

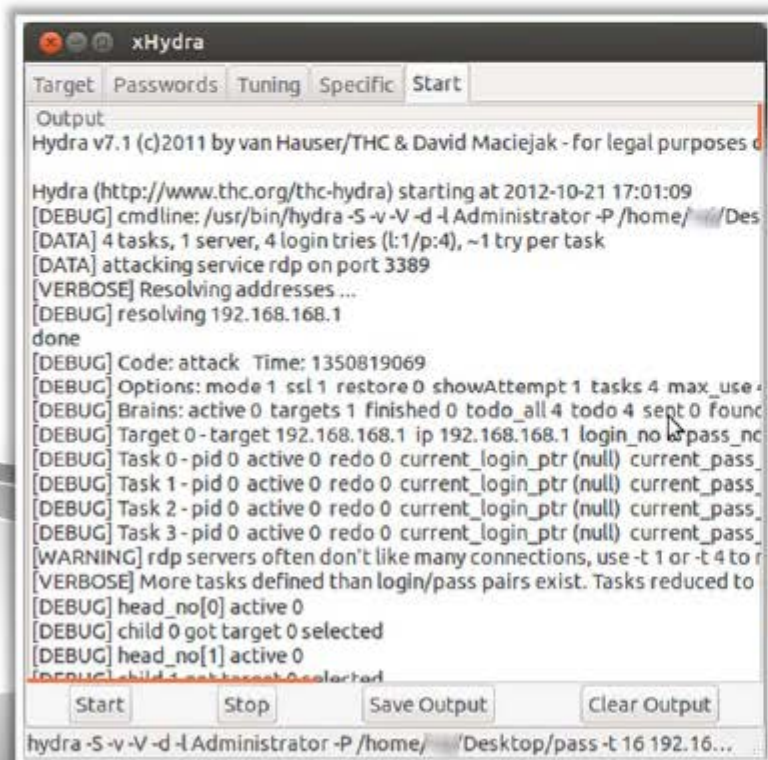


Web Password Cracking Tools: **THC-Hydra** and **Brutus**



THC-Hydra

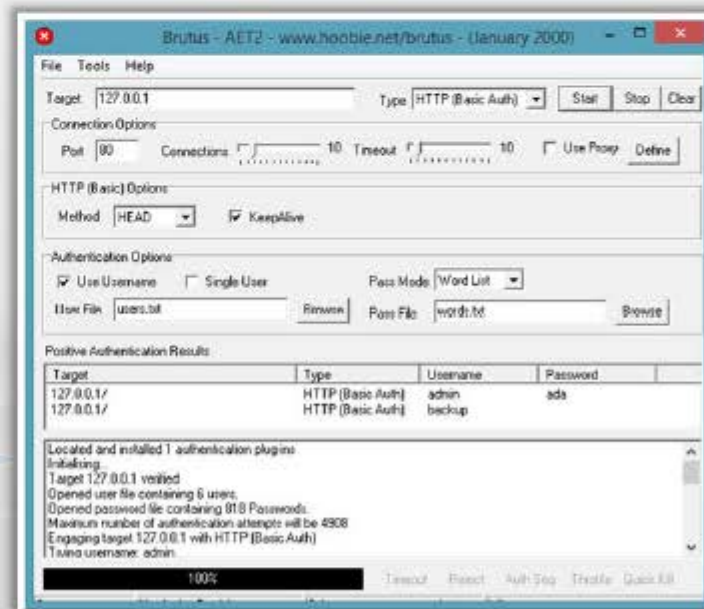
- Hydra is a parallelized **login cracker** which supports numerous protocols to attack



<http://www.thc.org>

Brutus

- It includes a multi-stage authentication engine and can **make 60 simultaneous target connections**
- It supports no user name, single user name, **multiple user name**, password list, combo (user/password) list and configurable brute force modes



<http://www.hoobie.net>

Module Flow



**Webserver
Concepts**

1



**Webserver
Attacks**

2



**Attack
Methodology**

3



**Webserver
Attack Tools**

4



**Counter-
measures**

5



**Patch
Management**

6



**Webserver
Security Tools**

7

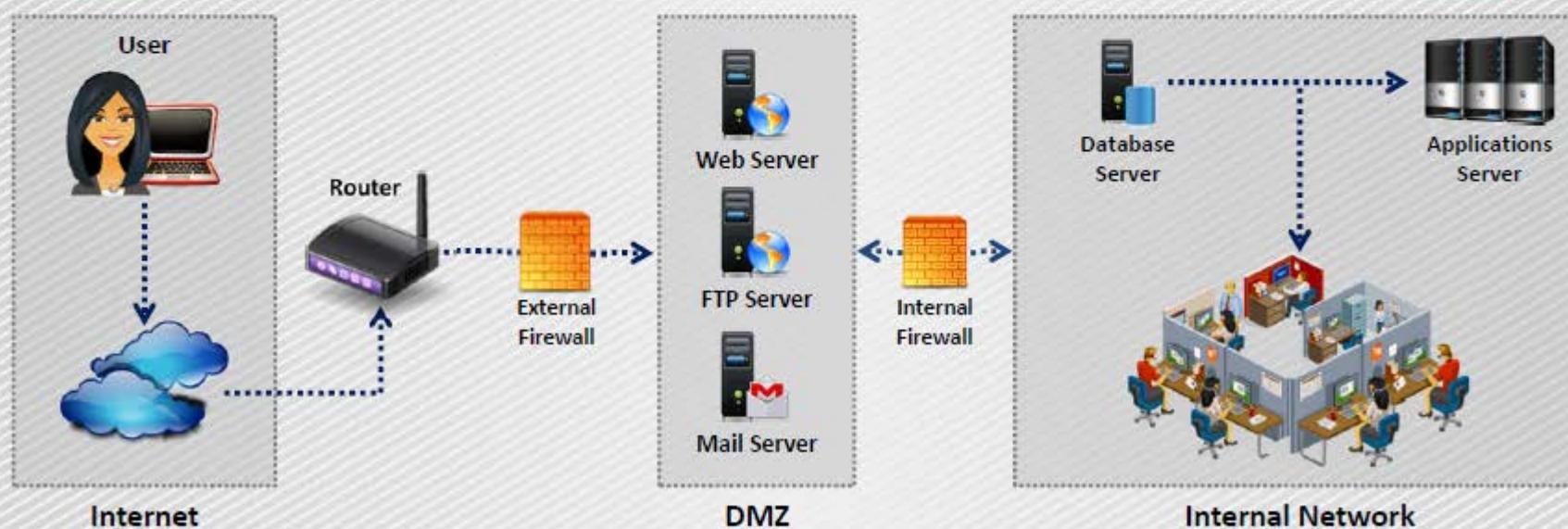


**Webserver
Pen Testing**

8

Place Web Servers in **Separate Secure Server Security Segment** on Network

- An ideal **web hosting network** should be designed with at least **three segments** namely Internet segment, secure server security segment often called demilitarized zone (DMZ), internal network
- Place the web server in **Server Security Segment** (DMZ) of the network isolated from public network as well as internal network
- The firewalls should be place for **internal network** as well as **Internet traffic** going towards DMZ



Countermeasures: Patches and Updates



01

Scan for existing vulnerabilities, patch, and update the **server software regularly**

02

Before applying any service pack, hotfix, or security patch, **read and peer review** all relevant documentation

03

Apply all updates, regardless of their type on an **"as-needed"** basis

04

Test the service packs and hotfixes on a representative **non-production environment** prior to being deployed to production

05

Ensure that service packs, hotfixes, and security patch levels are consistent on **all Domain Controllers (DCs)**

06

Ensure that **server outages** are scheduled and a complete set of **backup tapes** and emergency repair disks are available

07

Have a **back-out plan** that allows the system and enterprise to return to their original state, prior to the failed implementation

08

Schedule periodic service pack upgrades as part of operations maintenance and never try to have **more than two service packs behind**

Countermeasures: **Protocols**

01

Block all unnecessary **ports**, **Internet Control Message Protocol (ICMP) traffic**, and unnecessary protocols such as NetBIOS and SMB



02

Harden the TCP/IP stack and consistently apply the **latest software patches** and updates to system software



03

If using insecure protocols such as **Telnet**, **POP3**, **SMTP**, **FTP**, take appropriate measures to provide secure authentication and communication, for example, by using IPSec policies



04

If remote access is needed, make sure that the remote connection is secured properly, by using **tunneling and encryption protocols**










05

Disable **WebDAV** if not used by the application or keep secure if it is required



Countermeasures: Accounts

	Remove all unused modules and application extensions	✓
	Disable unused default user accounts created during installation of an operating system	✓
	When creating a new web root directory, grant the appropriate (least possible) NTFS permissions to the anonymous user being used from the IIS web server to access the web content	✓
	Eliminate unnecessary database users and stored procedures and follow the principle of least privilege for the database application to defend against SQL query poisoning	✓
	Use secure web permissions, NTFS permissions, and .NET Framework access control mechanisms including URL authorization	✓
	Slow down brute force and dictionary attacks with strong password policies, and then audit and alert for logon failures	✓
	Run processes using least privileged accounts as well as least privileged service and user accounts	✓

Countermeasures: Files and Directories

Eliminate unnecessary files within the **.jar files**



Disable serving of **directory listings**

Eliminate **sensitive configuration** information within the **byte code**



Eliminate the **presence of non web files** such as archive files, backup files, text files, and header/include files

Avoid mapping **virtual directories** between two different servers, or over a network



Disable serving certain **file types** by creating a resource mapping

Monitor and check all **network services logs**, **website access logs**, **database server logs** (e.g., Microsoft SQL Server, MySQL, Oracle) and OS logs frequently



Ensure the presence of **web application** or **website files** and **scripts** on a separate partition or drive other than that of the operating system, logs, and any other system files

Detecting Web Server Hacking Attempts



Use **Website Change Detection System** to detect hacking attempts on the web server

Website Change Detection System involves:



Running specific script on the server that detects any changes made in the existing executable file or new file included on the server



Periodically comparing the **hash values** of the files on the server with their respective master hash value to detect the changes made in codebase



Alerting the user upon any change detection on the server



For example: WebsiteCDS is a script that goes through your entire web folder and detects any changes made to the your code base and alert you using email

How to Defend Against Web Server Attacks



01

Ports

- 🟢 Audit the ports on server regularly to ensure that an **insecure** or unnecessary service is not active on your web server
- 🟢 Limit inbound traffic to **port 80 for HTTP** and **port 443 for HTTPS (SSL)**
- 🟢 Encrypt or restrict **intranet traffic**

02

Server Certificates

- 🟡 Ensure that **certificate data ranges** are valid and that certificates are used for their intended purpose
- 🟡 Ensure that the certificate has not been revoked and **certificate's public key** is valid all the way to a trusted root authority

03

Machine.config

- 🔴 Ensure that protected resources are mapped to **HttpForbiddenHandler** and unused **HttpModules** are removed
- 🔴 Ensure that **tracing is disabled** `<trace enable="false"/>` and **debug compiles** are turned off

04

Code Access Security

- 🔵 Implement **secure coding** practices
- 🔵 Restrict **code access security policy** settings
- 🔵 **Configure IIS** to reject URLs with `"../"` and install new patches and updates

How to Defend Against Web Server Attacks (Cont'd)



UrlScan

- UrlScan is a security tool that **restricts** the types of HTTP requests that IIS will process
- By blocking specific HTTP requests, the UrlScan security tool helps to **prevent potentially harmful requests** from reaching applications on the server
- UrlScan screens all incoming requests to the server by filtering the requests based on **rules** that are set by the administrator

Services

- UrlScan can be configured to filter HTTP query string values and other HTTP headers to **mitigate SQL injection** attacks while the root cause is being fixed in the application.
- It provides **W3C formatted logs** for easier log file analysis through log parsing solutions like Microsoft Log Parser 2.2

How to Defend Against Web Server Attacks (Cont'd)



01

- Apply **restricted ACLs** and block remote registry administration
- Secure the **SAM** (Stand-alone Servers Only)



02

Ensure that security related settings are **configured appropriately** and access to the metabase file is restricted with hardened **NTFS permissions**



03

Remove unnecessary ISAPI filters from the webserver



04

- Remove all unnecessary file shares including the **default administration shares** if not required
- Secure the shares with restricted **NTFS permissions**



05

Relocate sites and virtual directories to **non-system partitions** and use IIS Web permissions to restrict access



06

Remove all unnecessary **IIS script mappings** for optional file extensions to avoid exploiting any bugs in the ISAPI extensions that handle these types of files



07

Enable a **minimum level of auditing** on your web server and use NTFS permissions to protect the log files



How to Defend Against Web Server Attacks (Cont'd)



Do use a **dedicated machine** as a web server

Do physically protect the **webserver machine** in a secure machine room



Create **URL mappings** to internal servers cautiously

Do not connect an IIS Server to the **Internet** until it is fully hardened



Do not install the **IIS server** on a domain controller

Do not allow anyone to **locally log on** to the machine except for the administrator



Use server side **session ID tracking** and match connections with time stamps, IP addresses, etc.

Do configure a **separate anonymous user account** for each application, if you host multiple web applications



If a database server, such as **Microsoft SQL Server**, is to be used as a backend database, install it on a **separate server**

Limit the **server functionality** in order to support the web technologies that are going to be used



Use **security tools** provided with web server software and **scanners** that automate and make the process of securing a web server easy

Screen and filter the **incoming traffic request**



How to Defend against HTTP Response Splitting and Web Cache Poisoning



Server Admin

- Use latest **web server software**
- Regularly **update/patch OS** and webserver
- Run **web Vulnerability Scanner**



Application Developers

- Restrict web application access to **unique IPs**
- Disallow **carriage return** (%0d or \r) and line feed (%0a or \n) characters
- Comply to **RFC 2616** specifications for HTTP/1.1



Proxy Servers

- Avoid sharing **incoming TCP connections** among different clients
- Use different TCP connections with the proxy for different **virtual hosts**
- Implement “**maintain request host header**” correctly

How to Defend against DNS Hijacking



Choose an **ICANN** accredited **registrar** and encourage them to set **Registrar-Lock** on the domain name



Safeguard the **registrant account information**



Include DNS hijacking into **incident response and business continuity planning**



Use DNS monitoring tools/services to **monitor DNS server IP address and alert**



Avoid downloading **audio and video codecs** and other downloaders from untrusted websites



Install **antivirus** program and update it regularly



Change the **default router password** that comes with the factory settings

Module Flow



**Webserver
Concepts**

1



**Webserver
Attacks**

2



**Attack
Methodology**

3



**Webserver
Attack Tools**

4



**Counter-
measures**

5



**Patch
Management**

6



**Webserver
Security Tools**

7



**Webserver
Pen Testing**

8

Patches and Hotfixes

Hotfixes are an **update to fix a specific customer issue** and not always distributed outside the customer organization

A patch is a **small piece of software designed to fix problems**, security vulnerabilities, and bugs and improve the performance of a computer program or its supporting data

Users may be notified through **emails** or through the **vendor's website**

A patch can be considered as a **repair job to a programming problem**

Hotfixes are sometimes packaged as a set of fixes called a **combined hotfix** or **service pack**

What is Patch Management?



“Patch management is a process used to ensure that the **appropriate patches** are installed on a system and help fix known vulnerabilities”



An automated patch management process

Detect

Use tools to detect missing security patches

Assess

Asses the issue(s) and its associated severity by mitigating the factors that may influence the decision

Acquire

Download the patch for testing

Test

Install the patch first on a testing machine to verify the consequences of the update

Deploy

Deploy the patch to the computers and make sure the applications are not affected

Maintain

Subscribe to get notifications about vulnerabilities as they are reported

Identifying Appropriate Sources for Updates and Patches



1

First make a **patch management plan** that fits the operational environment and business objectives



2

Find appropriate **updates** and **patches** on the home sites of the applications or operating systems' vendors



3

The recommended way of tracking issues relevant to **proactive patching** is to register to the home sites to **receive alerts**

Installation of a Patch

01

Users can access and install security patches via the **World Wide Web**

Patches can be installed in two ways

Manual Installation

In this method, the user has to **download the patch** from the vendor and fix it



Automatic Installation

In this method, the applications use the **Auto Update** feature to update themselves



Implementation and Verification of a Security Patch or Upgrade

1



Before installing any patch **verify the source**

2



Use proper **patch management program** to validate files versions and checksums before deploying security patches

3



The patch management tool must be **able to monitor the patched systems**

4

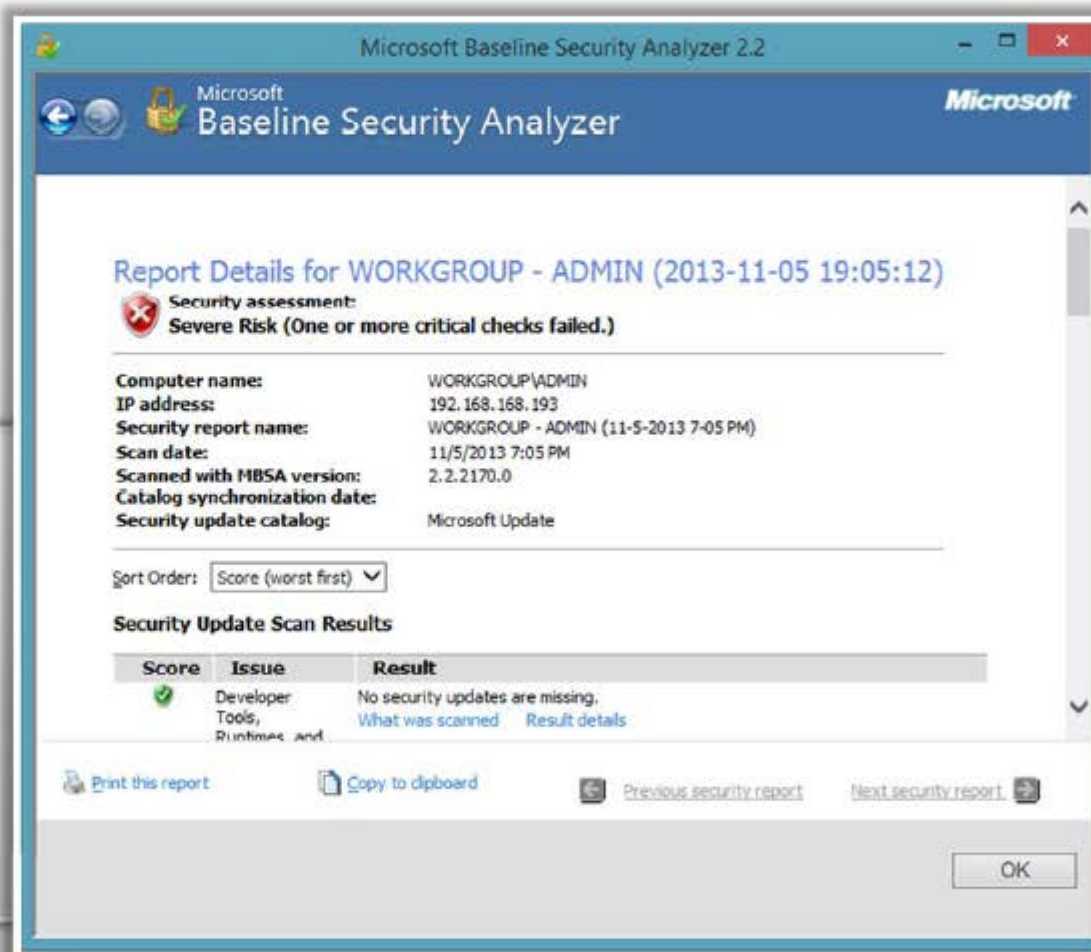


The **patch management team** should check for updates and patches regularly

Patch Management Tool: Microsoft Baseline Security Analyzer (MBSA)



- MBSA checks for **available updates** to the operating system, Microsoft Data Access Components (MDAC), MSXML (Microsoft XML Parser), .NET Framework, and SQL Server
- It also scans a computer for insecure **configuration settings**



<http://www.microsoft.com>

Patch Management Tools



Altiris Client Management Suite

<http://www.symantec.com>



Prism Suite

<http://www.newboundary.com>



GFI LanGuard

<http://www.gfi.com>



MaaS360® Patch Analyzer Tool

<http://www.maas360.com>



Kaseya Security Patch Management

<http://www.kaseya.com>



Secunia CSI

<http://secunia.com>



ZENworks® Patch Management

<http://www.novell.com>



Lumension® Patch and Remediation

<http://www.lumension.com>



Security Manager Plus

<http://www.manageengine.com>



VMware vCenter Protect

<http://www.vmware.com>

Module Flow



**Webserver
Concepts**

1



**Webserver
Attacks**

2



**Attack
Methodology**

3



**Webserver
Attack Tools**

4



**Counter-
measures**

5



**Patch
Management**

6



**Webserver
Security Tools**

7



**Webserver
Pen Testing**

8

Web Application Security Scanners: Syhunt Dynamic and N-Stalker Web Application Security Scanner

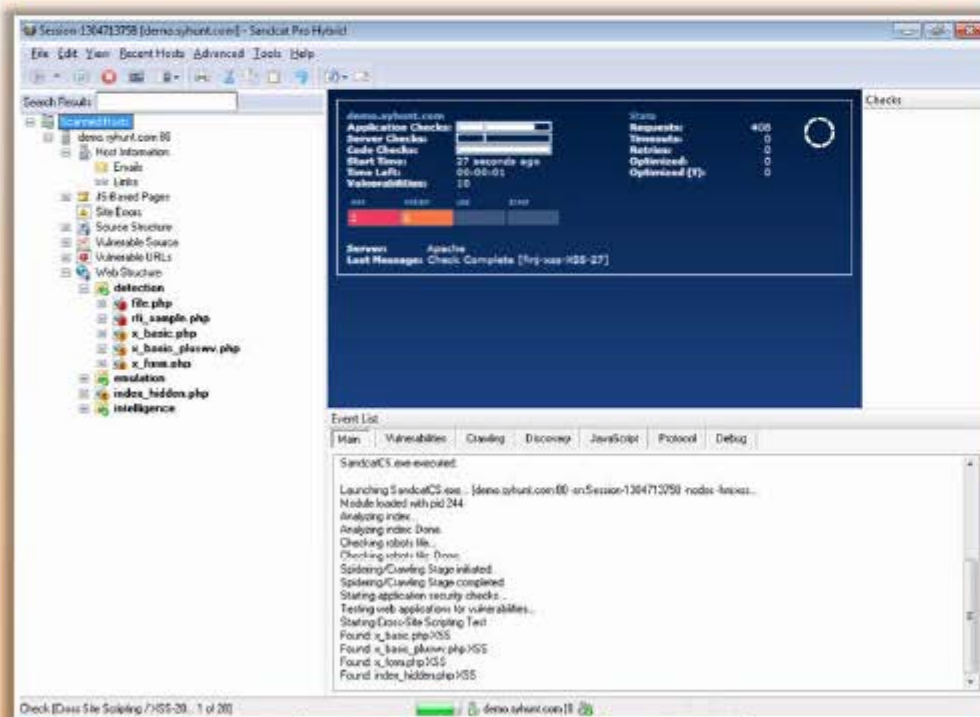


Syhunt Dynamic

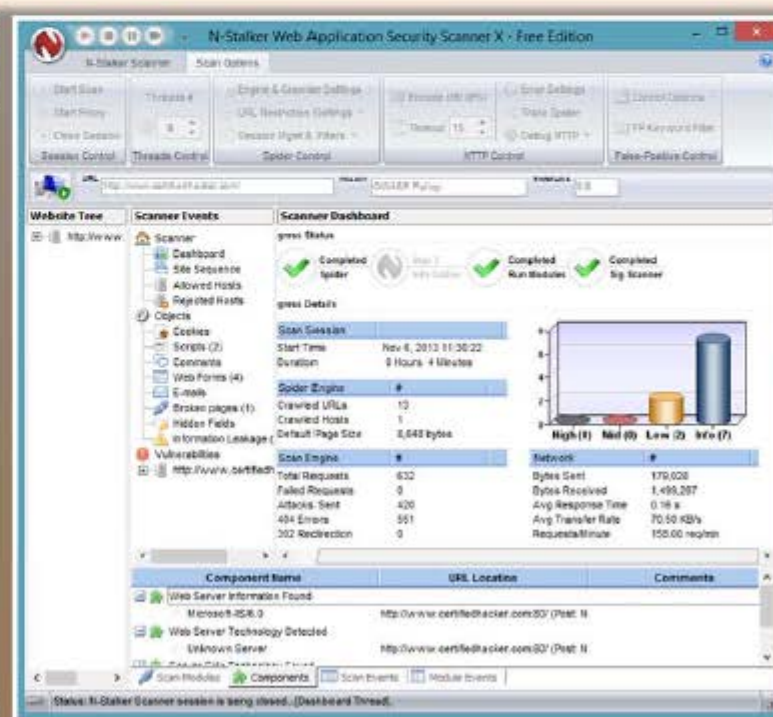
Syhunt Dynamic helps to automate **web application security** testing and guard organization's **web infrastructure** against various web application security threats

N-Stalker Web Application Security Scanner

N-Stalker is a **WebApp Security Scanner** to search for vulnerabilities such as SQL injection, XSS, and known attacks



<http://www.syhunt.com>



<http://www.nstalker.com>

Web Server Security Scanners: **Wikto** and **Acunetix Web Vulnerability Scanner**



Wikto



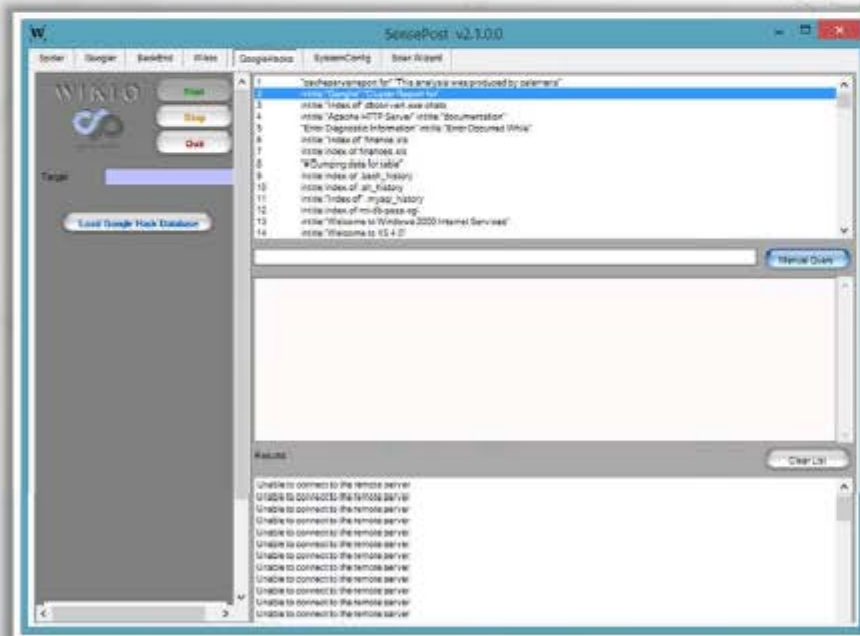
Wikto is a **web server security scanner** for windows

- Fuzzy logic error code checking
- Google assisted directory mining
- Back-end miner
- Real time HTTP request/response monitoring

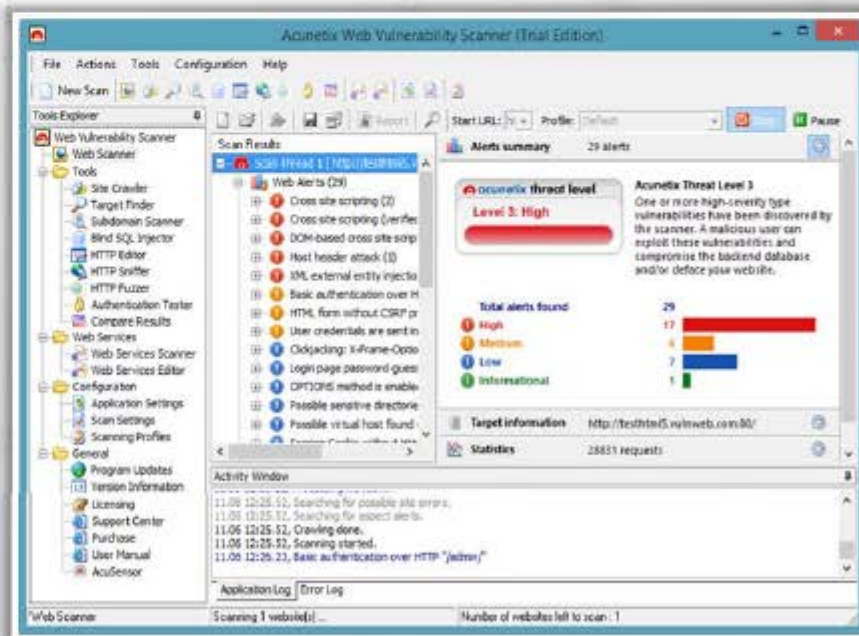


Acunetix Web Vulnerability Scanner

- Acunetix WVS **checks web applications** for SQL injections, cross-site scripting, etc.
- It includes advanced penetration testing tools to ease **manual security audit processes**, and also creates professional security audit and regulatory compliance reports



<http://www.sensepost.com>



<http://www.acunetix.com>

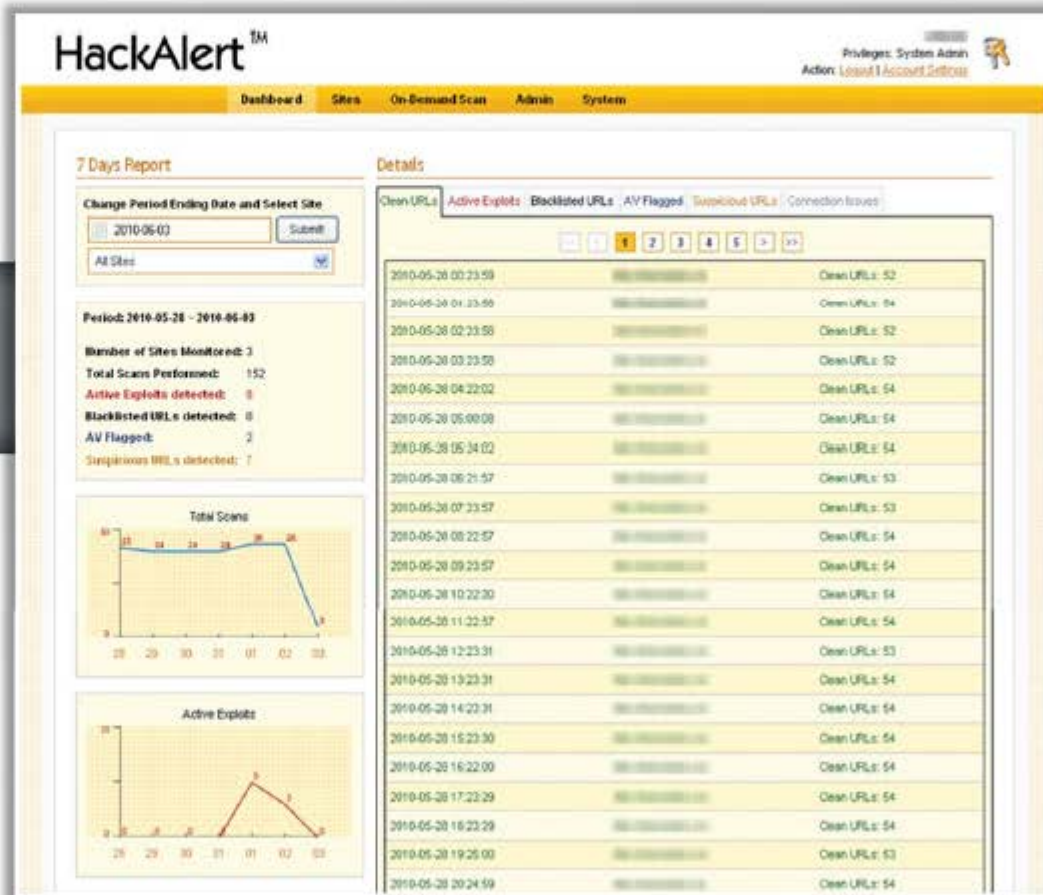
Web Server Malware Infection Monitoring Tool: HackAlert



HackAlert is a **cloud-based service** that identifies hidden zero-day malware and drive-by downloads in websites and online advertisements

Features

- Protects clients and customers from malware injected websites
- Identifies malware
- Displays injected code snippets
- Deploys as cloud-based SaaS
- Integrates with WAF or web server modules for instant mitigation

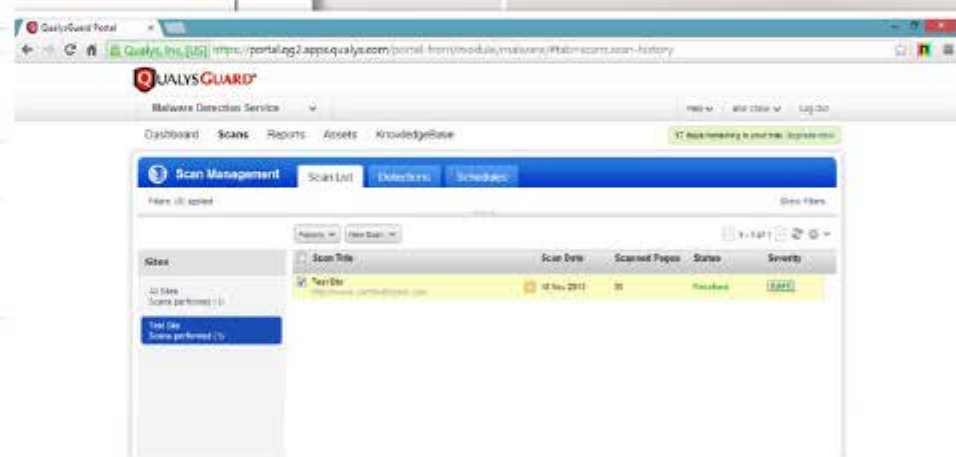
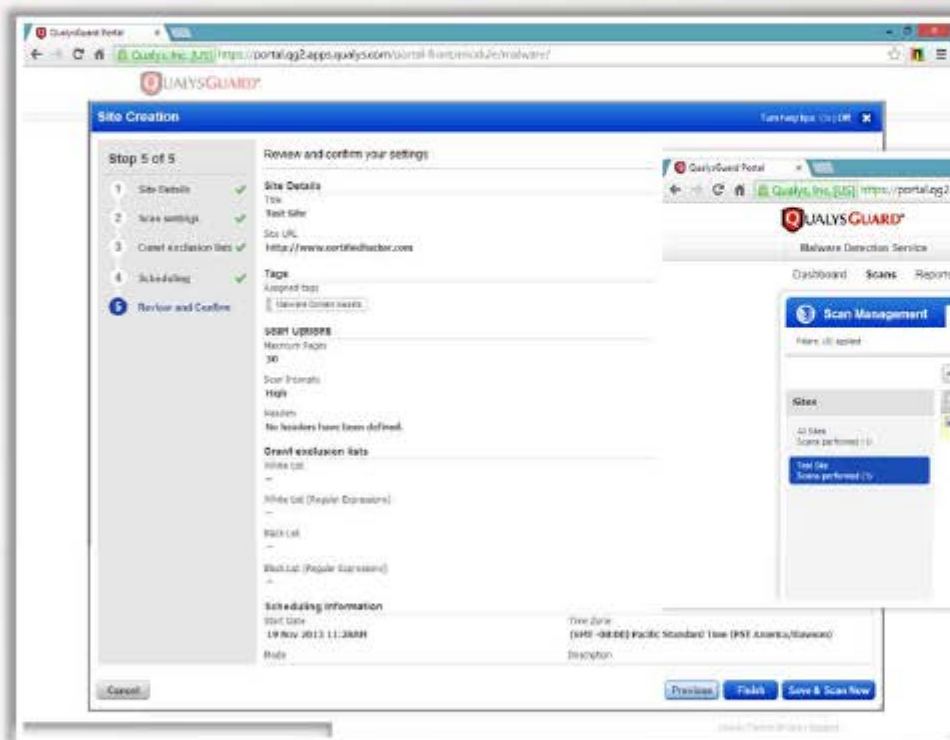


<http://www.armorize.com>

Web Server Malware Infection Monitoring Tool: QualysGuard Malware Detection



- QualysGuard® Malware Detection Service scans websites for **malware infections** and **threats**



<http://www.qualys.com>

Webserver Security Tools



Retina CS

<http://www.beyondtrust.com>



Arirang

<http://monkey.org>



Nscan

<http://nscan.hypermart.net>



**N-Stalker Web Application
Security Scanner**

<http://www.nstalker.com>



**NetIQ Secure Configuration
Manager**

<http://www.netiq.com>



Infiltrator

<http://www.infiltration-systems.com>



SAINTscanner

<http://www.saintcorporation.com>



WebCruiser

<http://sec4app.com>



HP WebInspect

<https://download.hpsmartupdate.com>



dotDefender

<http://www.applicure.com>

Module Flow



**Webserver
Concepts**

1



**Webserver
Attacks**

2



**Attack
Methodology**

3



**Webserver
Attack Tools**

4



**Counter-
measures**

5



**Patch
Management**

6



**Webserver
Security Tools**

7



**Webserver
Pen Testing**

8

Web Server Penetration Testing

- Web server pen testing is used to **identify, analyze, and report vulnerabilities** such as authentication weaknesses, configuration errors, protocol related vulnerabilities, etc. in a web server
- The best way to perform penetration testing is to **conduct a series of methodical and repeatable tests**, and to work through all of the different application vulnerabilities

Why Webserver Pen Testing?

Verification of Vulnerabilities

To exploit the vulnerability in order to test and fix the issue

Remediation of Vulnerabilities

To retest the solution against vulnerability to ensure that it is completely secure

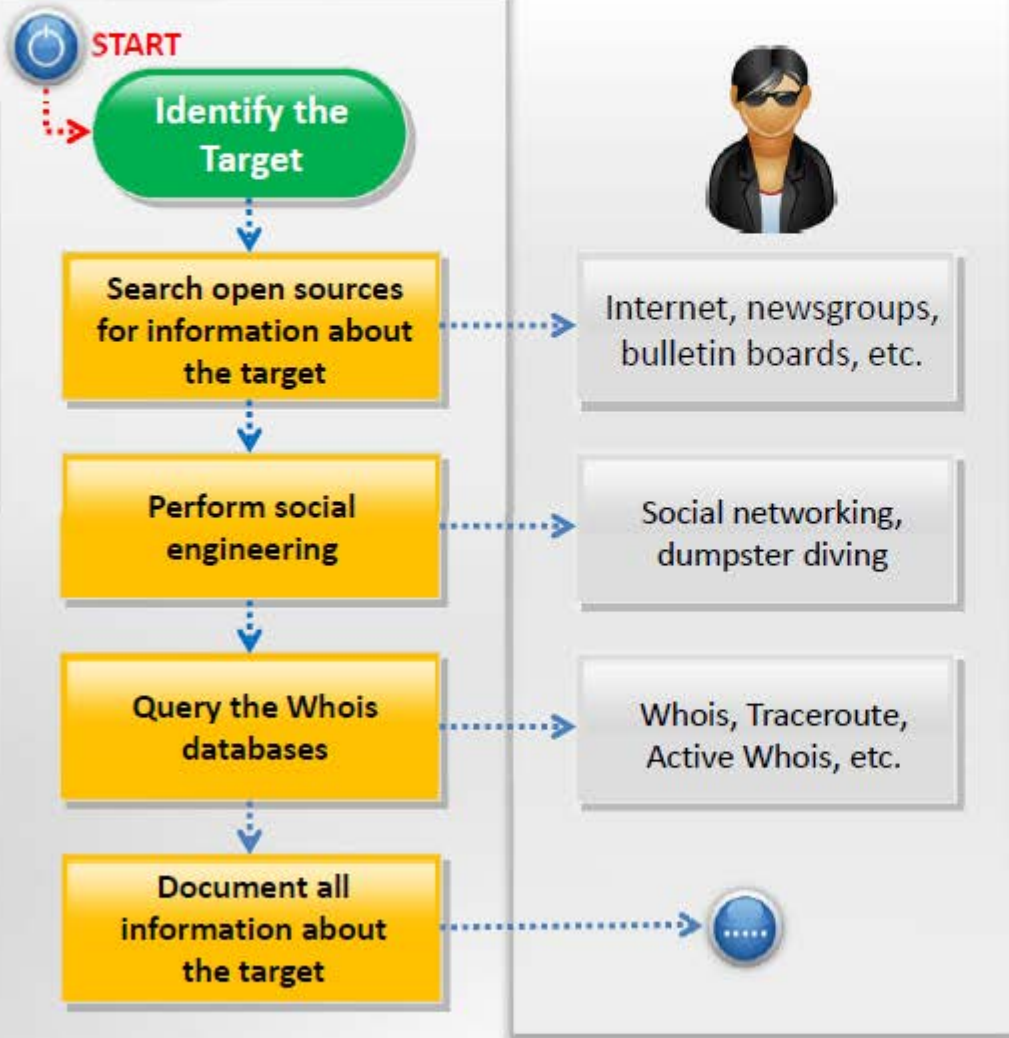


Identification of Web Infrastructure

To identify make, version, and update levels of web servers; this helps in selecting exploits to test for associated published vulnerabilities

Web Server Penetration Testing

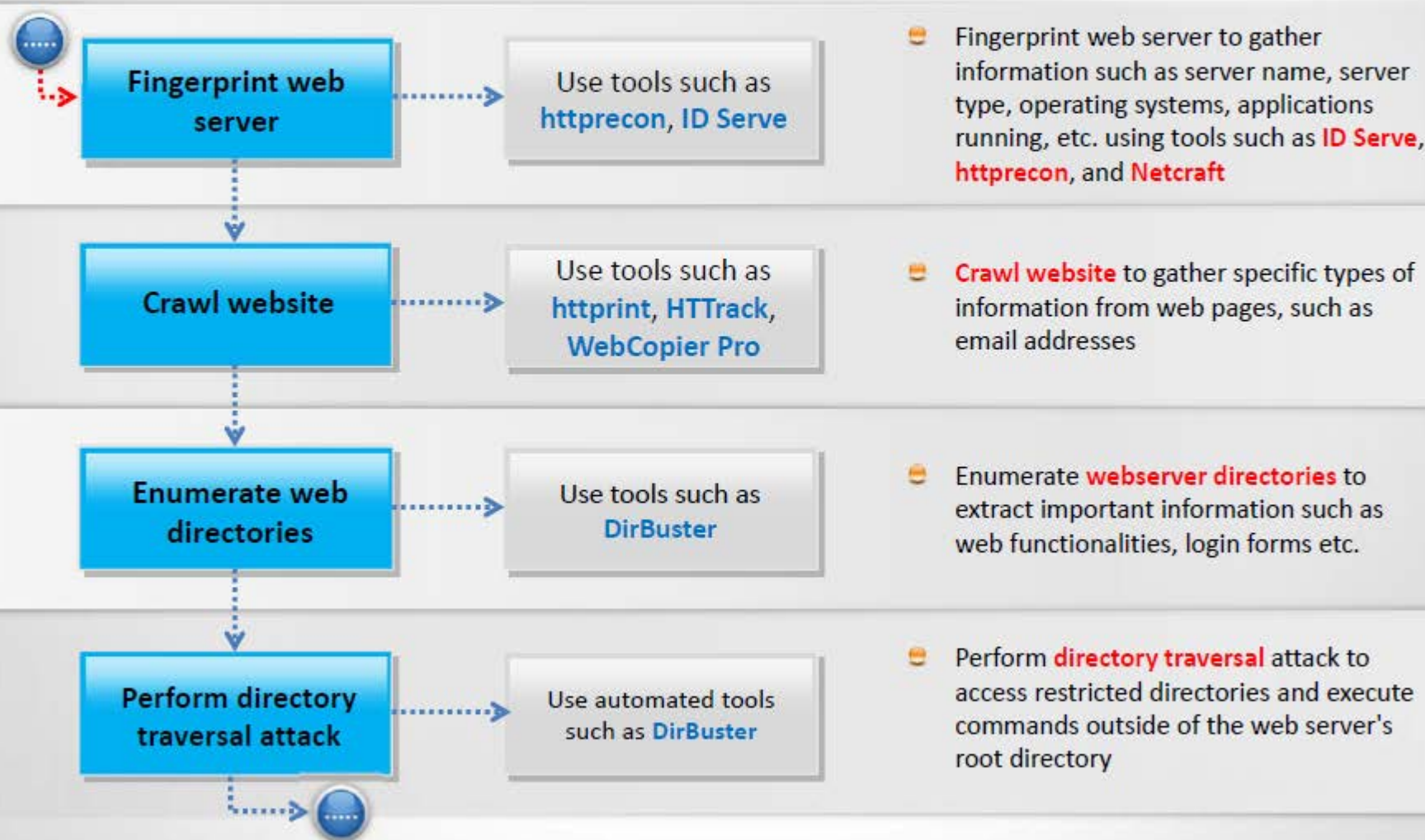
(Cont'd)



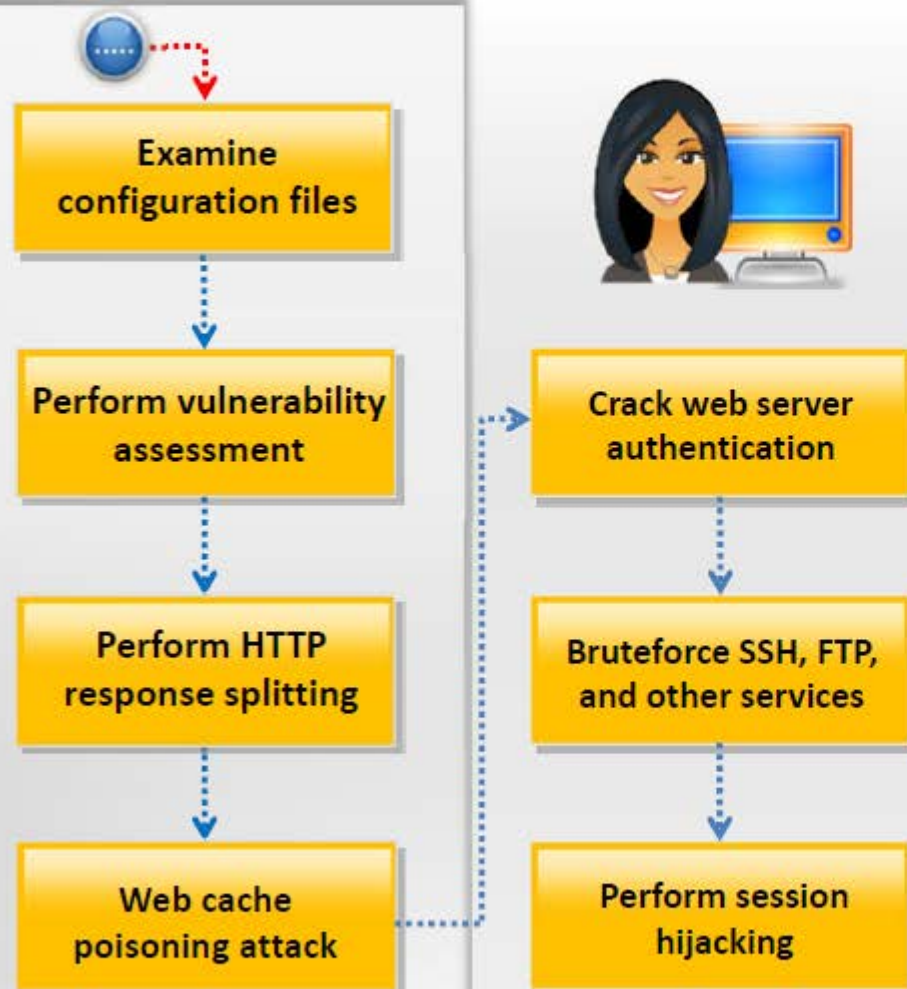
- Webserver penetration testing starts with **collecting as much information** as possible about an organization ranging from its physical location to operating environment
- Use **social engineering techniques** to collect information such as human resources, contact details, etc. that may help in **webserver authentication testing**
- Use **Whois database query tools** to get the details about the target such as domain name, IP address, administrative contacts, Autonomous System Number, DNS, etc.
- Note:** Refer Module 02: Footprinting and Reconnaissance for more information gathering techniques



Web Server Penetration Testing (Cont'd)



Web Server Penetration Testing (Cont'd)



- Perform vulnerability scanning to **identify weaknesses** in a network using tools such as **HP WebInspect**, **Nessus**, etc. and determine if the system can be exploited
- Perform HTTP response splitting attack to pass malicious data to a vulnerable application that includes the data in an HTTP response header
- Perform web cache poisoning attack to force the web server's cache to **flush its actual cache content** and send a specially **crafted request**, which will be stored in cache
- Bruteforce SSH, FTP, and other services login credentials to gain **unauthorized access**
- Perform session hijacking to **capture valid session cookies and IDs**. Use tools such as Burp Suite, Firesheep, Jhijack, etc. to automate session hijacking

Web Server Penetration Testing

(Cont'd)



Perform MITM attack

- Perform MITM attack to access sensitive information by **intercepting and altering communications** between an end-user and web servers

Perform web application pen testing

- Note:** Refer Module 12: Hacking Web Applications for more information on how to conduct web application pen testing

Examine webserver logs

- Use tools such as Webalizer, AWStats, Ktmatu Relax, etc. to **examine web server logs**

Exploit frameworks

Document all the findings

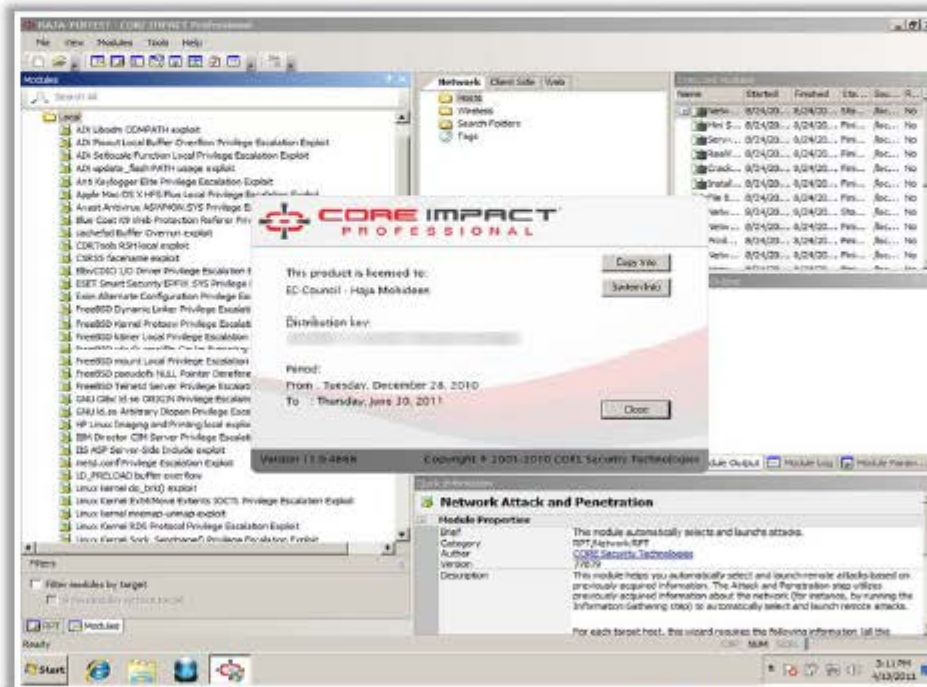
- Use tools such as **Metasploit**, **w3af**, etc. to exploit frameworks

Web Server Pen Testing Tool: CORE Impact® Pro



CORE Impact® Pro is the software solution for assessing and testing **security vulnerabilities** in the organization:

- Web Applications
- Network Systems
- Endpoint systems
- Wireless Networks
- Network Devices
- Mobile Devices
- IPS/IDS and other defenses

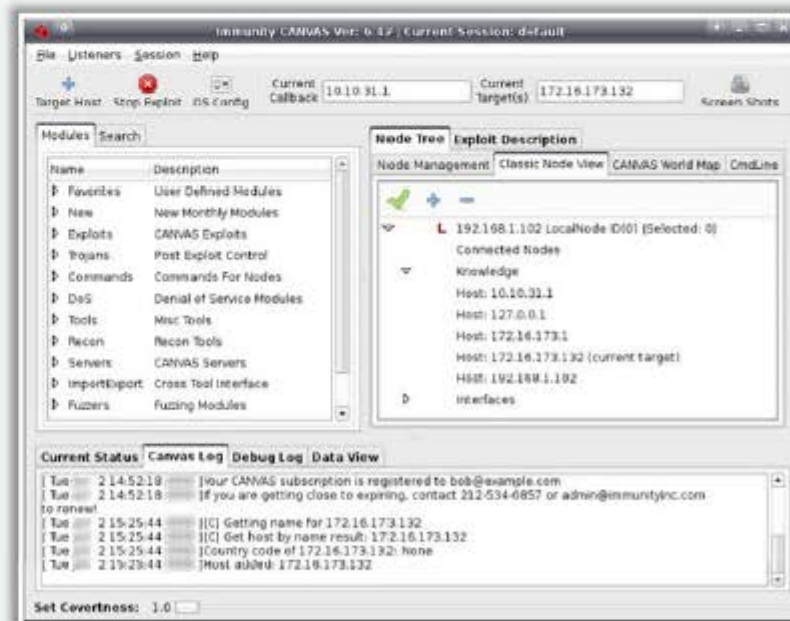
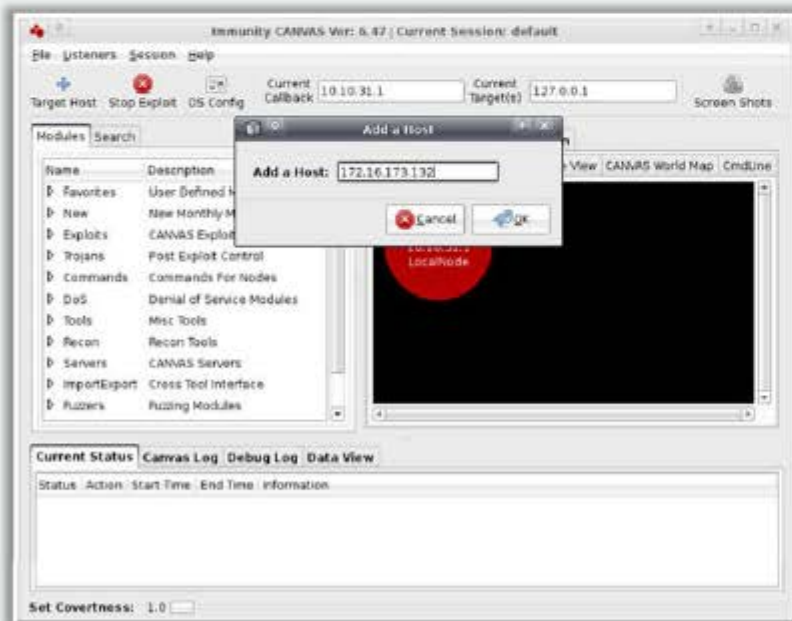


<http://www.coresecurity.com>

Web Server Pen Testing Tool: Immunity CANVAS

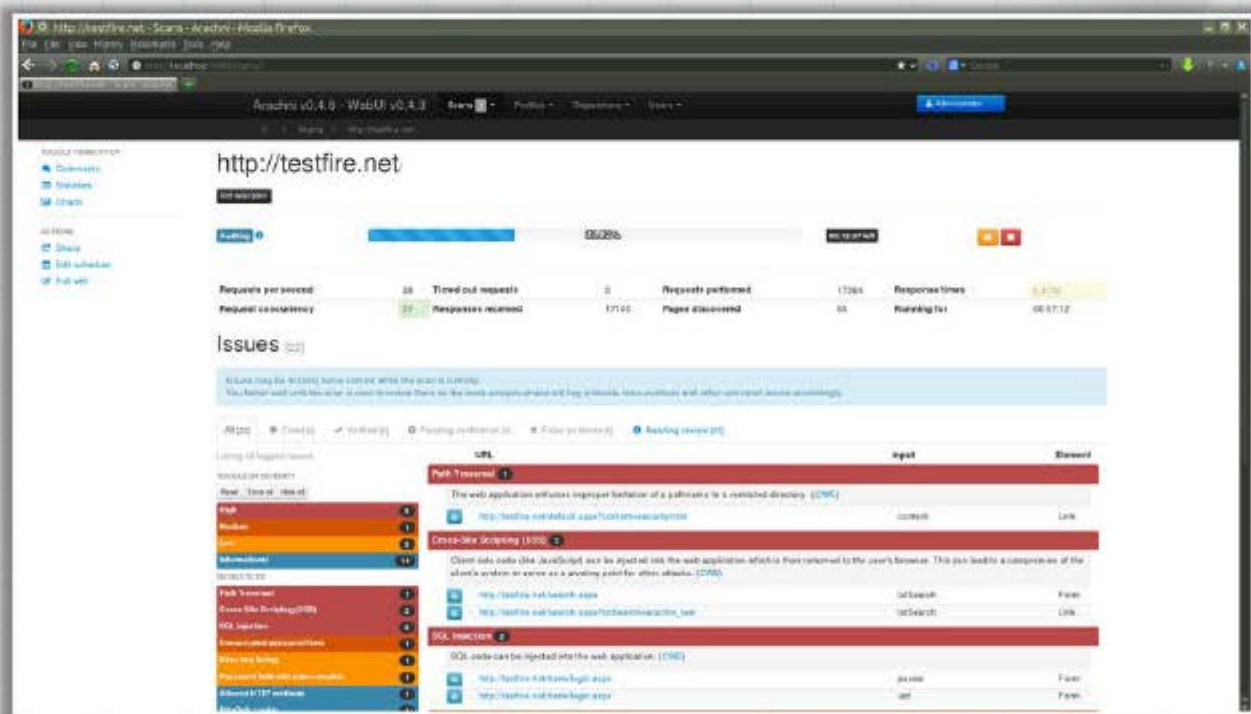


CANVAS is an automated exploitation system, and a comprehensive, reliable **exploit development framework** to security professionals and penetration testers



<http://www.immunitysec.com>

CEH
Certified Ethical Hacker



<http://www.arachni-scanner.com>

Module Summary



- ❑ Web servers assume critical importance in the realm of Internet security
- ❑ Vulnerabilities exist in different releases of popular web servers and respective vendors patch these often
- ❑ The inherent security risks owing to the compromised web servers have impact on the local area networks that host these websites, even on the normal users of web browsers
- ❑ Looking through the long list of vulnerabilities that had been discovered and patched over the past few years, it provides an attacker ample scope to plan attacks to unpatched servers
- ❑ Different tools/exploit codes aid an attacker in perpetrating web server's hacking
- ❑ Countermeasures include scanning for the existing vulnerabilities and patching them immediately, anonymous access restriction, incoming traffic request screening, and filtering