Module 14
Hacking Web Servers

Ansh Bhawnani

Web server concepts

Module 14

1. Web server Introduction

Module 14

Web server concepts

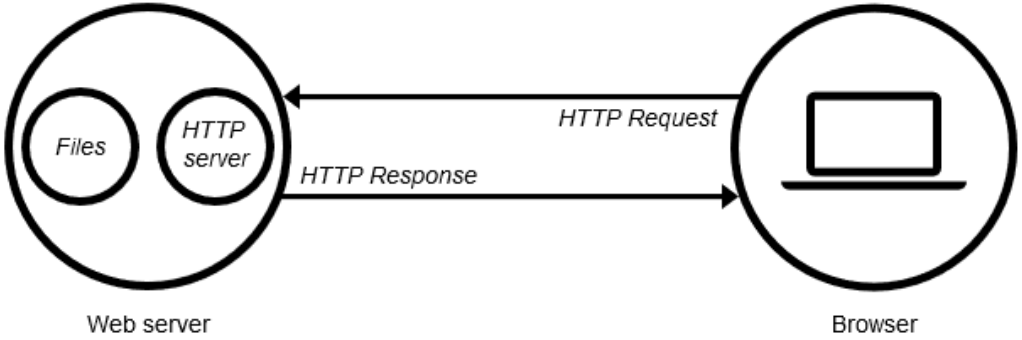
- **On the hardware side,**
  - A web server is a computer that stores web server software and a website's component files (e.g. HTML documents, images, CSS stylesheets, and JavaScript files).
  - It is connected to the Internet and supports physical data interchange with other devices connected to the web.

Web server concepts

- **On the software side,**
  - A web server includes several parts that control how web users access hosted files, at minimum an HTTP server.
  - An HTTP server is a piece of software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages).
  - It can be accessed through the domain names (like mozilla.org) of websites it stores, and delivers their content to the end-user's device.

Web server concepts

- At the most basic level, whenever a browser needs a file which is hosted on a web server, the browser requests the file via HTTP.
- When the request reaches the correct web server (hardware), the ***HTTP server*** (software) accepts request, finds the requested document (if it doesn't then a 404 response is returned), and sends it back to the browser, also through HTTP.



Web server concepts

- A **static web server**, or stack, consists of a computer (hardware) with an HTTP server (software). We call it "static" because the server sends its hosted files "as-is" to your browser.
- A **dynamic web server** consists of a static web server plus extra software, most commonly an ***application server*** and a ***database***. We call it "dynamic" because the application server updates the hosted files before sending them to your browser via the HTTP server.

2. Types of Web Servers

Module 14

Web server concepts

Apache HTTP Server

- This is the most popular web server developed by the Apache Software Foundation.
- Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more.
- About 40% of the web server machines run the Apache Web Server.

Web server concepts

IIS Web Server

- A Microsoft product, IIS is a server that offers all the features such as Apache. Since it's not an open source, adding personal modules as well as modifying becomes a bit difficult.
- It supports all the platforms that run Windows operating system. Additionally, you also get good customer support, if there is any issue.

Microsoft IIS

Web server concepts

Nginx Web Server

- Nginx is the next open source web server after Apache. It comprises of IMAP/POP3 proxy server.
- The significant features offered by Nginx are high performance, stability, simple configuration and low resource usage.
- No threads are used to handle the requests by Nginx, instead a highly scalable event-driven architecture that uses small and predictable amount of memory under load is utilized. It has become popular recently and hosts about 20% of all the domains globally.

Web server concepts

LiteSpeed Server

- A high-performance Apache drop-in replacement, LiteSpeed (LSWS) is the 4th popular web server on the internet.
- When you upgrade your web server to LiteSpeed, you will experience improved performance that too with low operating cost.
- It has the ability to load Apache configuration files directly and can replace the Apache within 15 minutes without any downtime.
- LSWS replaces all the Apache functions which other front-end proxy solutions can't do to simplify the use and make the transition from Apache smooth and easy.

Web server concepts

Apache Tomcat Server

- An open source ***Java servlet*** container, Apache Tomcat functions as a web server. Java servlets are Java equivalent to other dynamic web content technologies such as PHP and ASP.NET.
- Sun Microsystems donated Tomcat's code base to the Apache Software Foundation in 1999 which became a top-level Apache project in 2005. Currently, it powers just under 1% of all websites.
- Apache Tomcat is typically used to run Java applications.

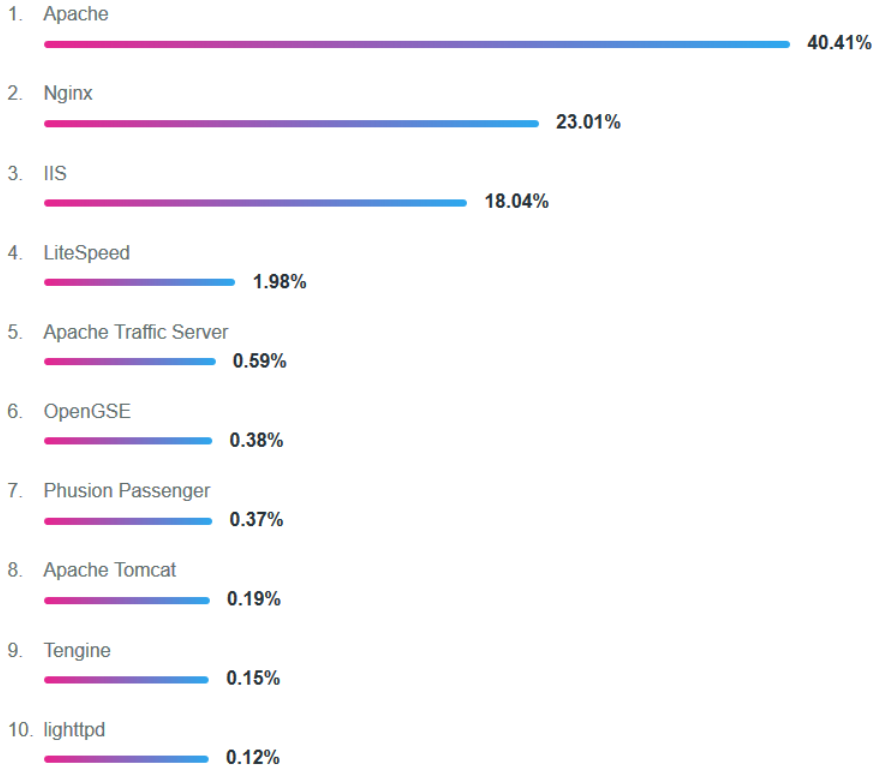
Web server concepts

Node.js Server

- Node.js is basically a server-side JavaScript environment that is used for network applications such as web servers.
- Node.js differs from other popular web servers because it is mainly a cross-platform runtime environment for building network applications with.
- An event-driven architecture is applied by Node.js which is capable of asynchronous I/O. Due to these design choices throughput and scalability are optimized in web applications which helps to run real-time communication and browser games

3. Web Server Market Shares

Web server concepts



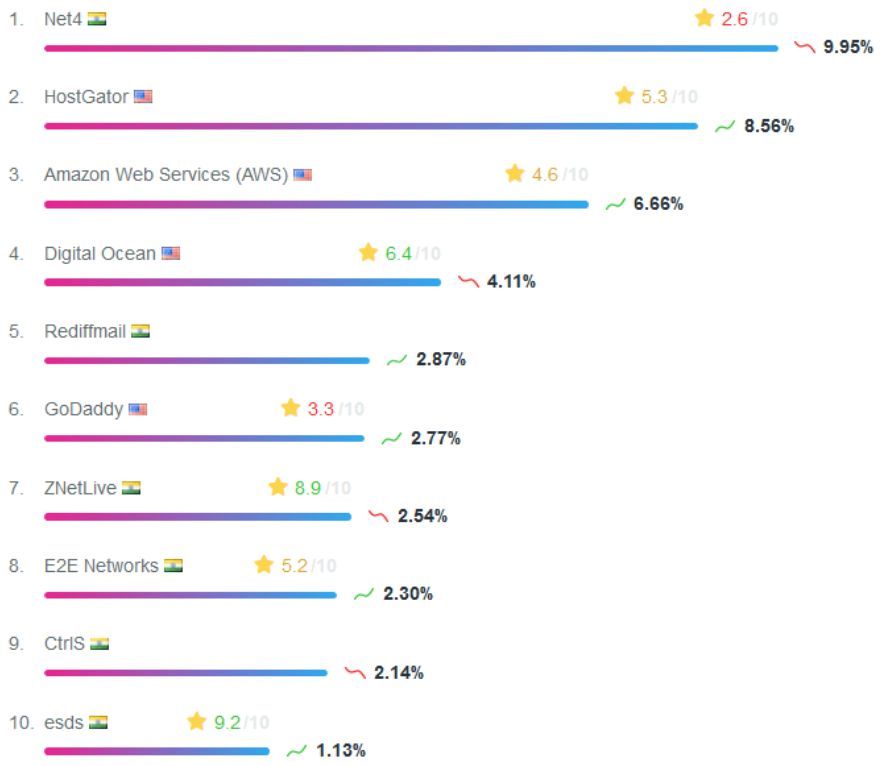
Web server concepts

Apache

Top 5 CMS hosted

WordPress	78.98%
Wix	11.74%
Sitefinity	3.47%
Joomla	3.1%
Squarespace	2.71%

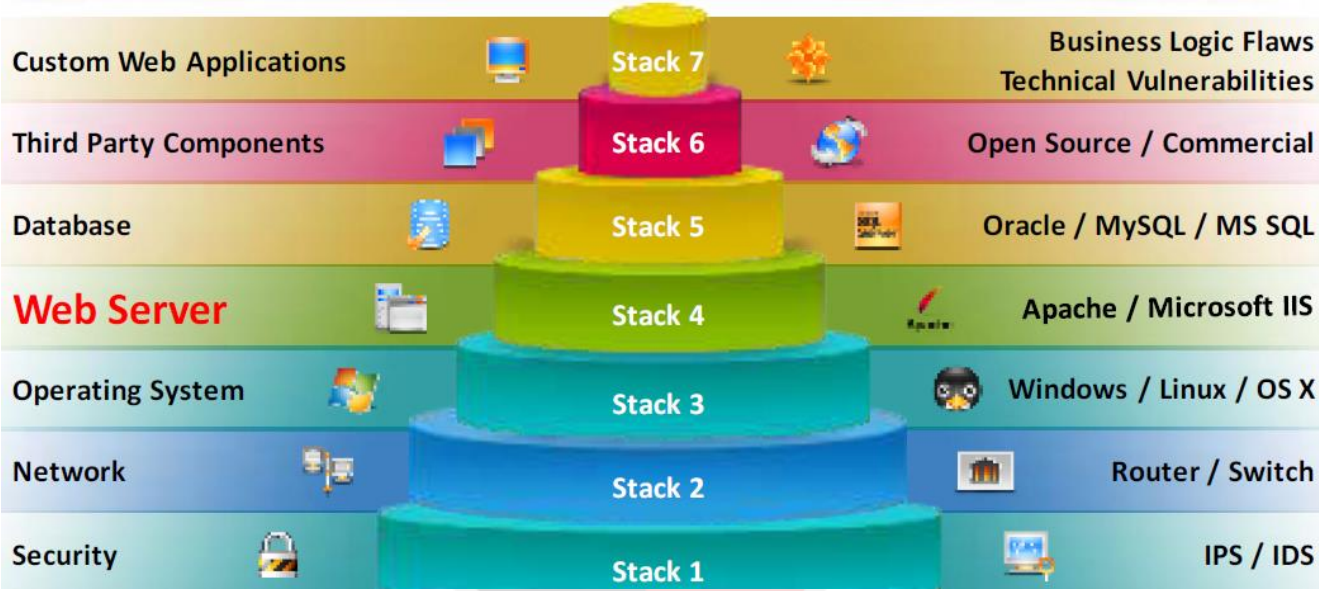
Web server concepts



4. Web Server Security Issue

Module 14

Web server concepts



5. Why Web Servers Are Compromised

Module 14

Web server concepts

- 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.

Web server concepts

- Default accounts with their default or no passwords.
- Unnecessary default, backup, or sample files.
- Misconfiguration 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.

6. Impact of Webserver Attacks

Module 14

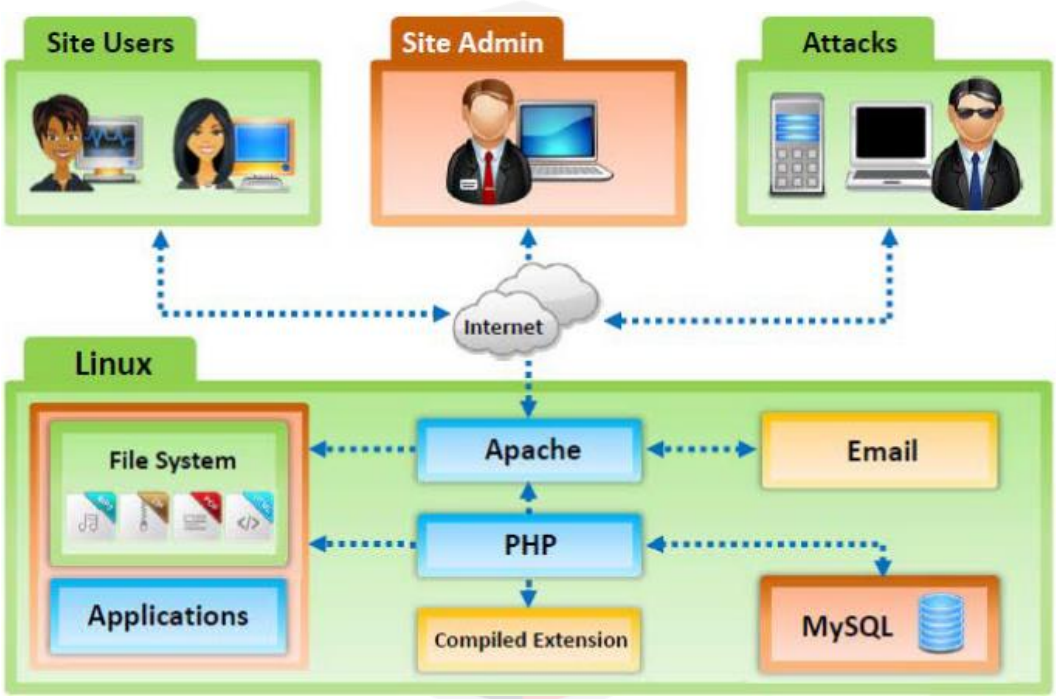
Web server concepts

- Compromise of user accounts.
- Website defacement.
- Secondary attacks from the Website.
- Root access to other applications or servers.
- Data tampering and data theft.

7. Open Source Webserver Architecture

Module 14

Web server concepts



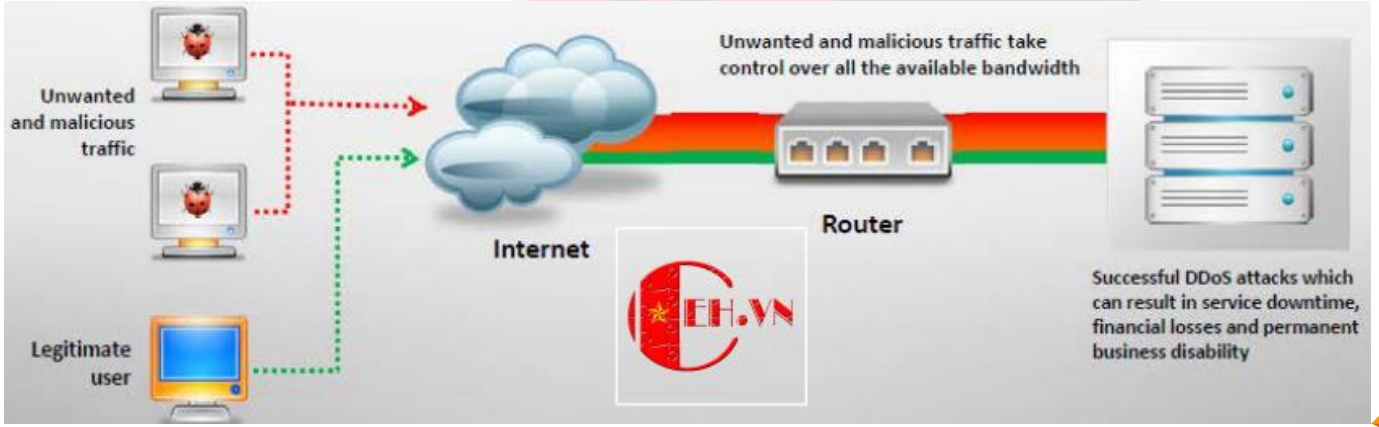
Web Server Attacks

Module 14

1. DoS/DDoS Attacks

Web Server 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.
- Attackers may target high profile web servers such as banks, credit card payment geteways, government owned services, etc. to steam user credentials.



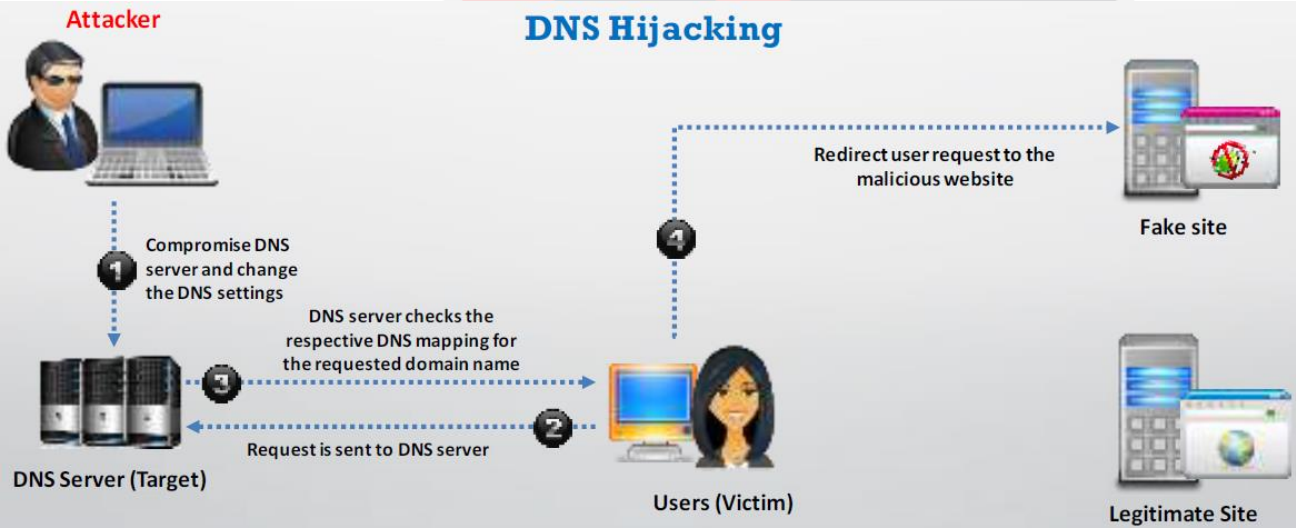
Web Server Attacks

- To crash the webserver running the application, attacker targets the following services by consuming the webserver with fake requests:
  - Network bandwidth
  - Server memory
  - Application exception handling mechanism
  - CPU usage
  - Hard disk space
  - Database space

2. DNS Server Hijacking

Web Server Attacks

- 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.

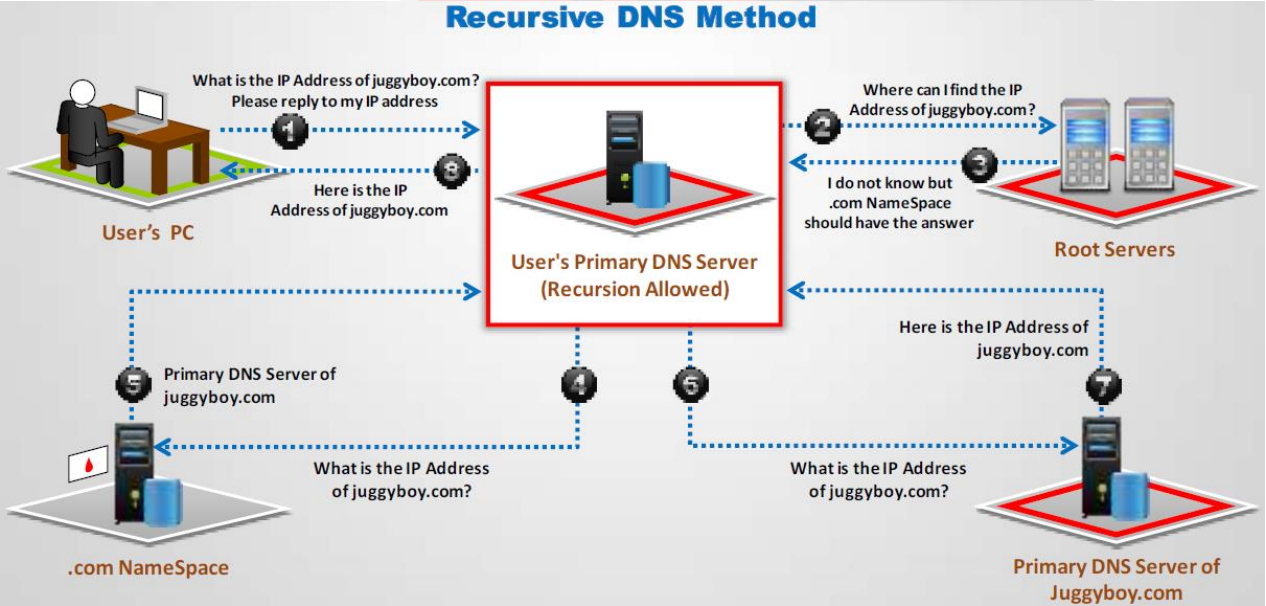


3. DNS Amplification Attack

Module 14

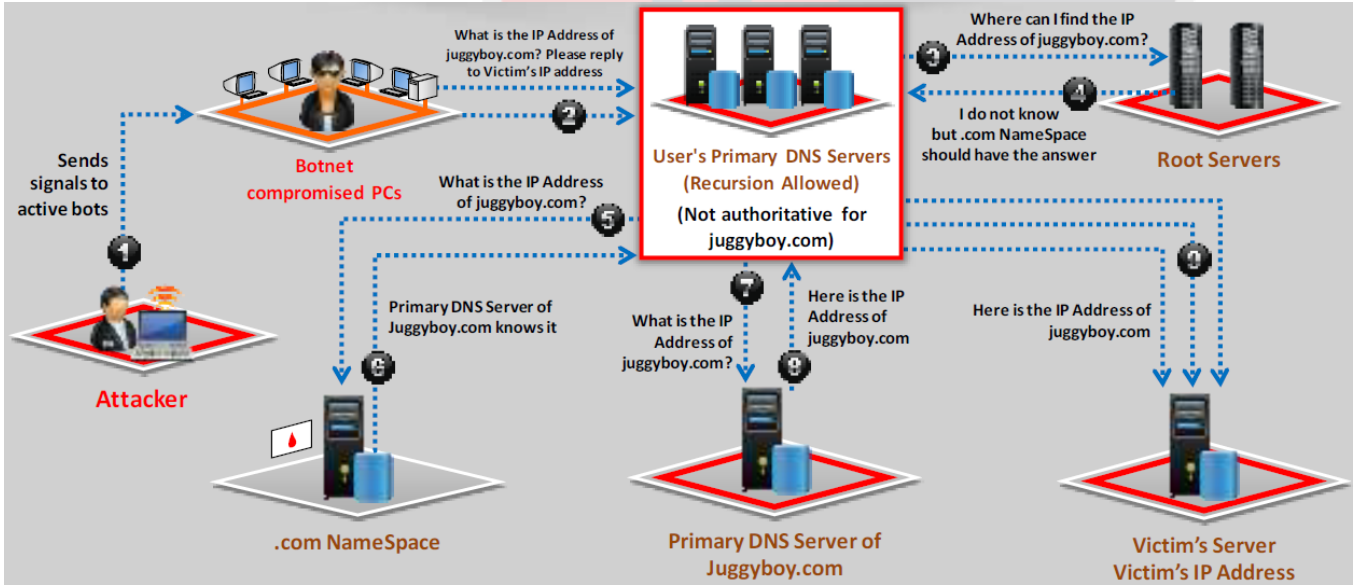
Web Server Attacks

- Attacker takes the advantages of DNS recursive method of DNS redirection to perform DNS amplification attack



Web Server Attacks

- Attacker uses compromised PCs with spoofed IP addresses to amplify the DDoS attacks on victims DNS server by exploiting DNS recursive method.



4. Directory Traversal

Module 14

Web Server Attacks

- The root directory is a specific directory on the server file system in which the users are confined. Users are not able to access anything above this root.
- 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.
- With a system vulnerable, an attacker can step out of the root directory and access other parts of the file system. This might give read access to restricted files, which could provide more information to further compromise the system.

Web Server Attacks

- In web applications with dynamic pages, input is usually received from browsers through GET or POST request methods. Here is an example of an HTTP GET request URL

```
GET http://test.webarticles.com/show.asp?view=oldarchive.html HTTP/1.1
Host: test.webarticles.com
```

- The attacker would assume that show.asp can retrieve files from the file system and sends the following custom URL.

```
GET http://test.webarticles.com/show.asp?view=../../../../../Windows/system.ini HTTP/1.1
Host: test.webarticles.com
```

Web Server Attacks

Preventing Directory Traversal attacks

- First of all, ensure you have installed the latest version of your web server software, and sure that all patches have been applied.
- Secondly, effectively filter any user input. Ideally remove everything but the known good data and filter meta characters from the user input. This will ensure that only what should be entered in the field will be submitted to the server.
- Use a good Web Application Vulnerability Scanner.

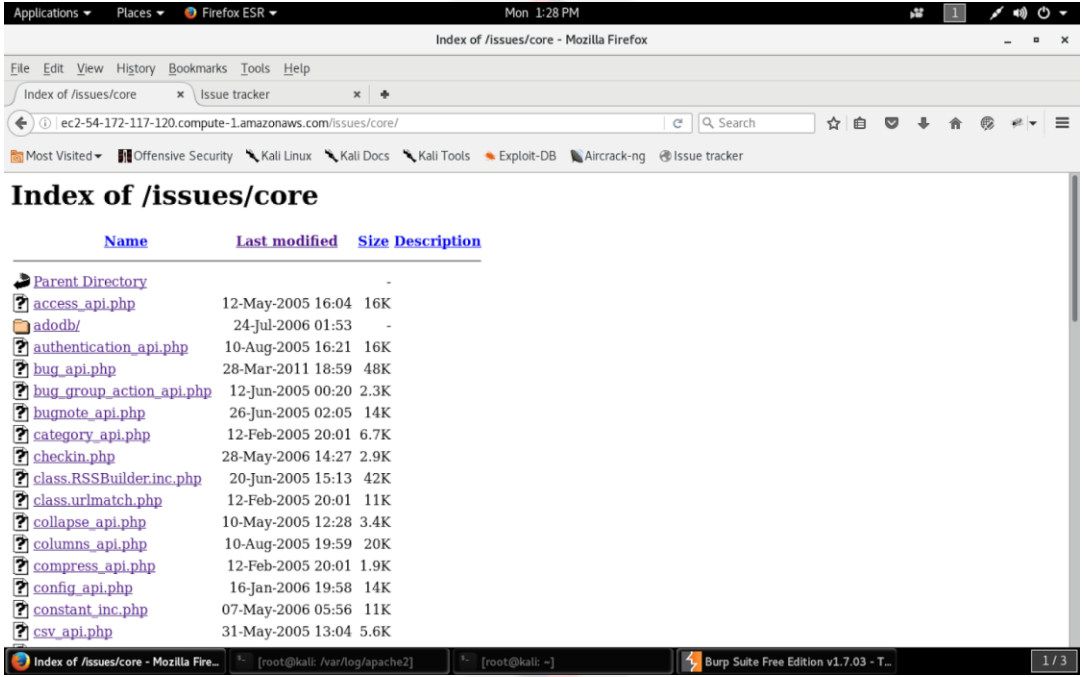
Directory Listing

Module 14

Web Server Attacks

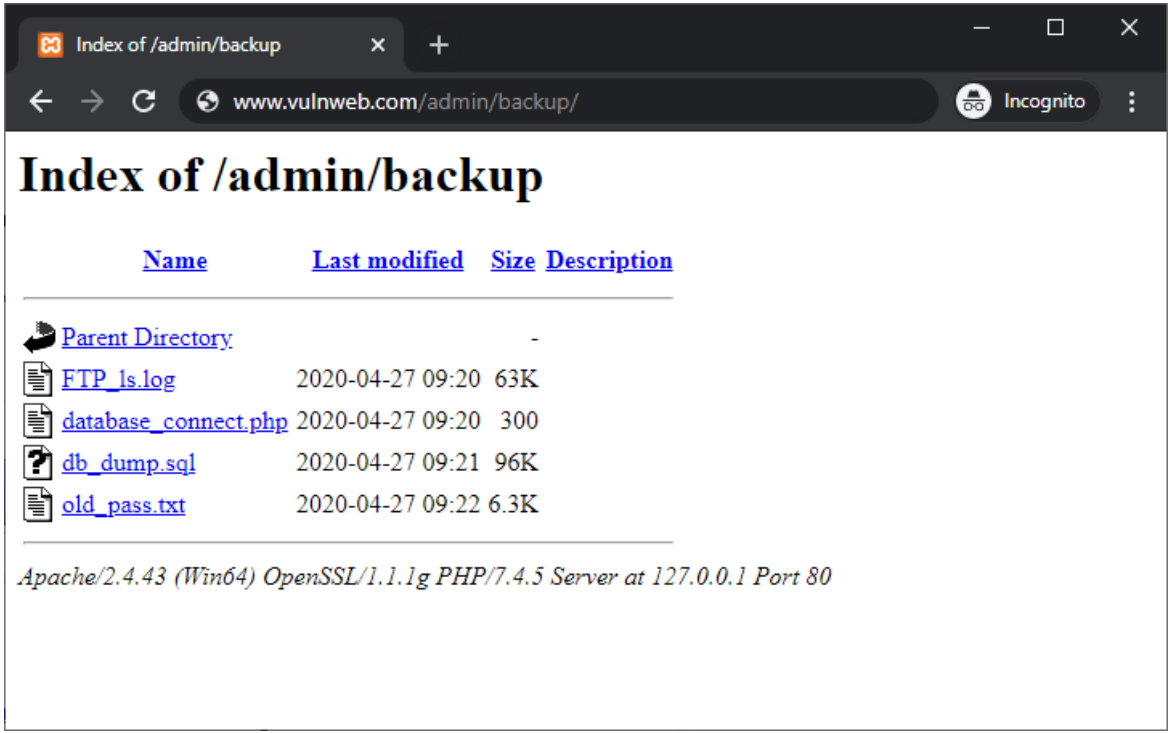
- Directory listing is a web server function that displays the directory contents when there is no index file in a specific website directory. It is dangerous because it leads to information disclosure.
- Even if directory listing is disabled, attackers might discover and exploit web server vulnerabilities. For example, there was an old Apache Tomcat vulnerability, where improper handling of null bytes (%00) and backslash (\) made it prone to directory listing attacks.
- Attackers might also discover directory indexes using cached or historical data contained in online databases. For example, Google's cache database

Web Server Attacks



Web Server Attacks

Web Server Attacks



How to Disable Directory Listing

- In *Apache*, You can disable directory listing by setting the ***Options*** directive in the Apache ***httpd.conf*** file by adding the following line:

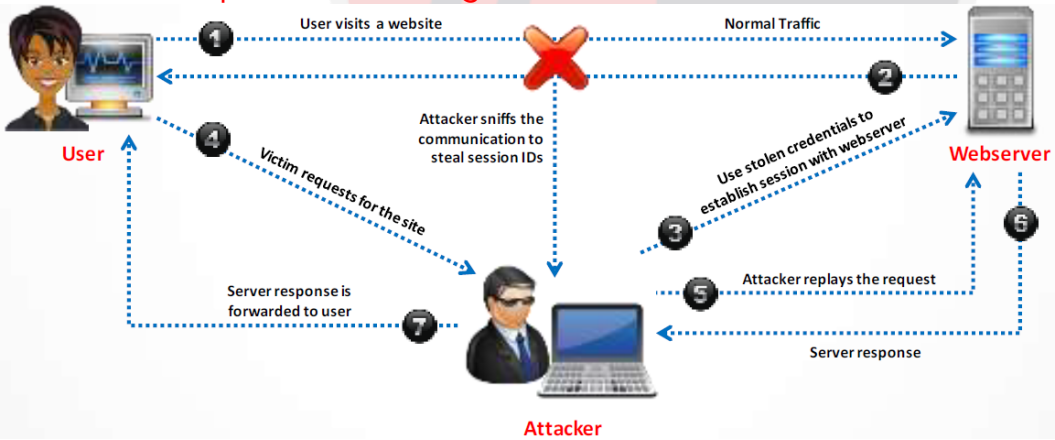
```
<Directory /your/website/directory>Options -Indexes</Directory>
```

- Or in *.htaccess* file as: Options –Indexes.
- Directory indexing is disabled by default in ***nginx*** so you do not need to configure anything. However, if it was turned on before, you can turn it off by opening the ***nginx.conf*** configuration file and changing autoindex on to autoindex off.

5. Man-in-the-Middle/Sniffing Attack

Web Server Attacks

- Man-in-the-Middle (MITM) attacks allow an attacker to access sensitive information by intercepting and altering communications between an end-user and webservers.
- Attacker acts as a proxy such that all the communication between the user and webserver passes through him.



6. Website Defacement

Module 14

Web Server Attacks

- 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 uses variety of methods such as MYSQL injection to access a site in order to deface it.

Web Server Attacks

7. Web Server Misconfiguration

Module 14

Web Server Attacks

- 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.
  - Sample Configuration, and Script Files.
  - Anonymous or Default Users/Passwords.
  - Verbose debug/error messages.
  - Misconfigured/Default SSL Certificates.
  - Unnecessary Services Enabled.
  - Remote Administration Functions.

Web Server Attacks

- This configuration allows anyone to view the server status page, which contains detailed information about the current user 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>
```

Web Server Attacks

- This configuration gives verbose error messages.
  - **php.ini** file:

```
display_error = On
log-errors = On
error-log = syslog
ignore_repeated_errors = Off
```

8. HTTP Response Splitting/CRLF Attack

Module 14

Web Server Attacks

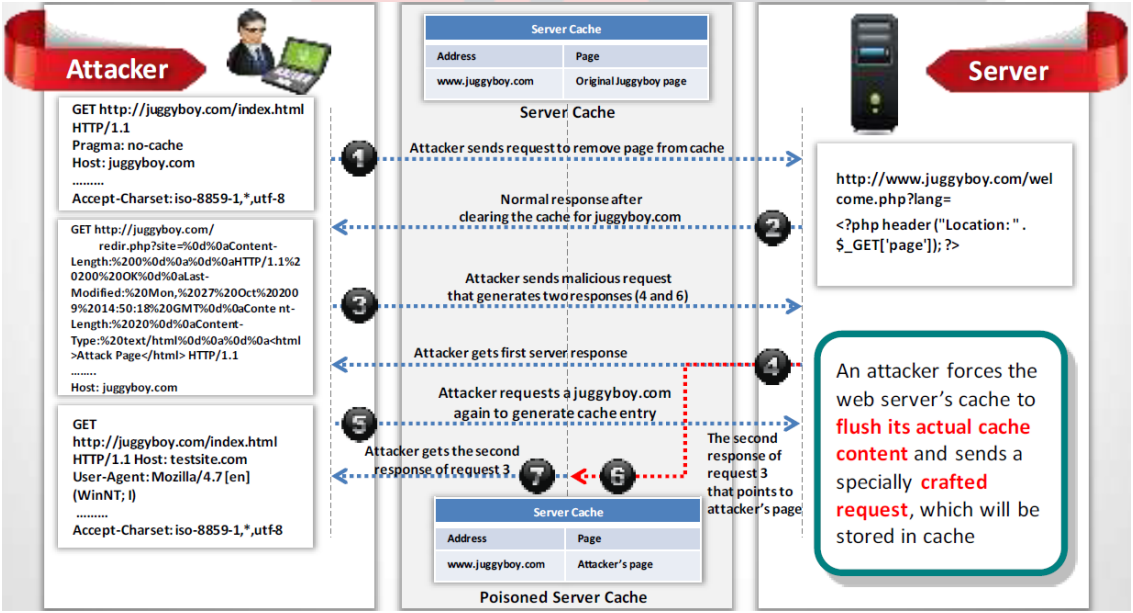
- HTTP response splitting attack involves adding header response data into the input field so that the server split the response into two responses.
- The attacker can control the second response to redirect user to a malicious website whereas the other responses will be discarded by web browser.
- The application must allow input that contains **CR** (carriage return, also given by %0d or **\r**) and **LF** (line feed, also given by %0a or **\n**) characters into the header AND the underlying platform must be vulnerable to the injection of such characters.
- These characters not only give attackers control of the remaining headers and body of the response the application intends to send, but also allow them to create additional responses entirely under their control.

10. Web Cache Poisoning Attack

Module 14

Web Server Attacks

- 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.



11. SSH Bruteforce Attack

Module 14

Web Server Attacks

- SSH protocols are used to create an encrypted SSH tunnel between two hosts in order to transfer unencrypted data over an insecure network.
- It works on TCP port 22.
- Attackers can bruteforce SSH login credentials to gain unauthorized access to a SSH tunnel.
- SSH tunnels can be used to transmit malwares and other exploits to victims without being detected.

12. Webserver Password Cracking

Module 14

Web Server Attacks

- 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.

13. Webserver Password Cracking Techniques

Module 14

Web Server Attacks

- 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 Server Attacks

Rainbow Tables

- A rainbow table works by doing a cryptanalysis very quickly and effectively.
- A rainbow table already computes hashes of the large set of available strings. There are two main steps in this:
- **Creating a Table**
  - Here, the hash of a string is taken and then reduced to create a new
    - hashMD5(12345678) = 25d55ad283aa400af464c76d713c07ad
    - hashMD5(25d55ad2) = 5c41c6b3958e798662d8853ece970f70

Web Server Attacks

- This is repeated until enough hashes in output chain. This represents one chain, which starts from the first plain text and ends at the last hash.
- After obtaining enough chains, we store them in a table.
- **Cracking the Password**
  - Starting off with the hashed text (the password) its checked if it exists in the database. If so, go to the start of the chain and start hashing until there is a match. As soon as the match is obtained, the process ceases and the authentication is cracked. The following flowchart explains the steps:

14. Web Application Attacks

Module 14

Web Server Attacks

- Vulnerabilities in web applications running on a webserver provide a broad attack path for webserver compromise.
  - Directory Traversal
  - Parameter/Form Tampering
  - Cookie Tampering
  - Command Injection Attacks
  - Buffer Overflow Attacks
  - Cross-Site Scripting (XSS) Attacks
  - Denial-of-Service (DoS) Attacks
  - Unvalidated Input and File injection Attacks
  - Cross-Site Request Forgery (CSRF) Attack
  - SQL Injection Attacks
  - Session Hijacking

Attack Methodology

Module 14

Attack Methodology

Webserver Attack Methodology

- Information Gathering
- Webserver Footprinting
- Mirroring Website
- Vulnerability Scanning
- Session Hijacking
- Hacking Webserver Passwords

1. Information Gathering

Module 14

Attack Methodology

- Information gathering involves collecting information about the targeted company.
- Attackers search the Internet, newsgroups, bulletin boards, etc. for information about the company.
- 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.
- **Note**: For complete coverage of information gathering techniques refer to Module 05: Footprinting and Reconnaissance

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.

2. Webserver Footprinting

Module 14

Attack Methodology

- Gather valuable system-level data such as account details, operating system, software versions, server names, and database schema details.
- Telnet a webserver to footprint a webserver and gather information such as server name, server type, operating systems, applications running, etc.
- Use tool such as ID Serve, httprecon, and Netcraft to perform footprinting.

3. Mirroring a Website

Module 14

Attack Methodology

- 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.

4. Vulnerability Scanning

Module 14

Attack Methodology

- Implement vulnerability scanning to identify weaknesses in a network and determine if the system can be exploited.
- Use a vulnerability scanner such as HP WebInspect, Acunetix Web Vulnerability Scanner, etc. to find hosts, services, and vulnerabilities.
- Sniff the network traffic to find out active systems, network services, applications, and vulnerabilities present.
- Test the web server infrastructure for any misconfiguration, outdated content, and known vulnerabilities.

Attack Methodology

Session Hijacking

- Sniff valid session IDs to gain unauthorized access to the Web Server and snoop the data.
- Use session hijacking techniques such as session fixation, session sidejacking, Cross-site scripting, etc. to capture valid session cookies and IDs.
- Use tools such as Burp Suite, Firesheep, JHijack, etc. to automate session hijacking.

5. Hacking Web Passwords

Module 14

Attack Methodology

- 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.

Web Server Attack Tools

Module 14

1. Metasploit

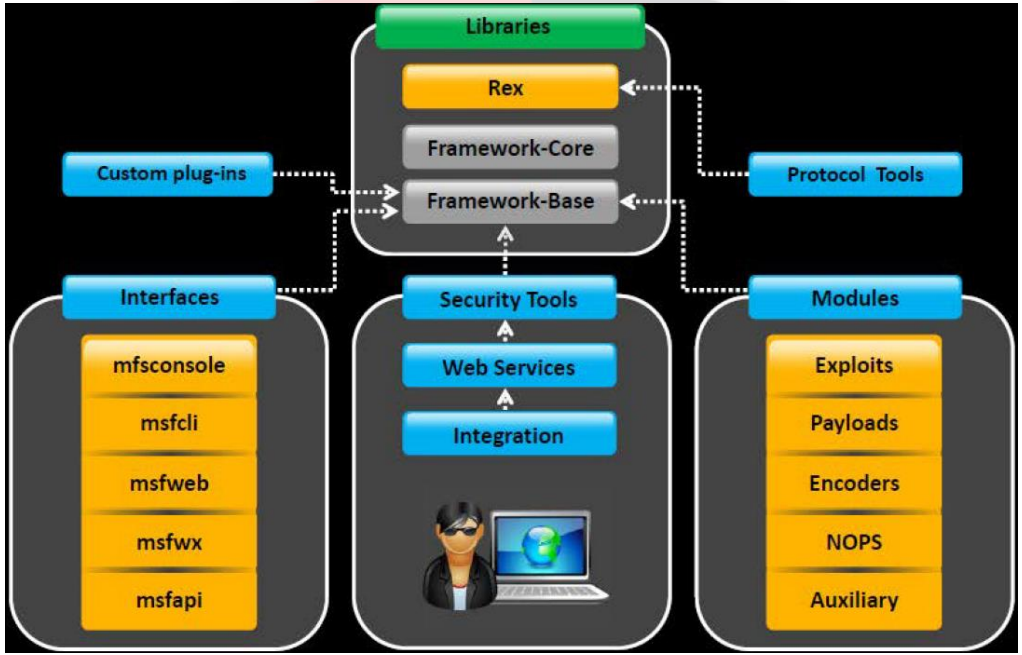
Module 14

Web Server Attack Tools

- 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.

Web Server Attack Tools

Metasploit Architecture



Web Server Attack Tools

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**:
  - Configuring Active Exploit
  - Verifying the Exploit Options
  - Selecting a Target
  - Selecting the Payload
  - Launching the Exploit

Web Server Attack Tools

Metasploit Payload Module

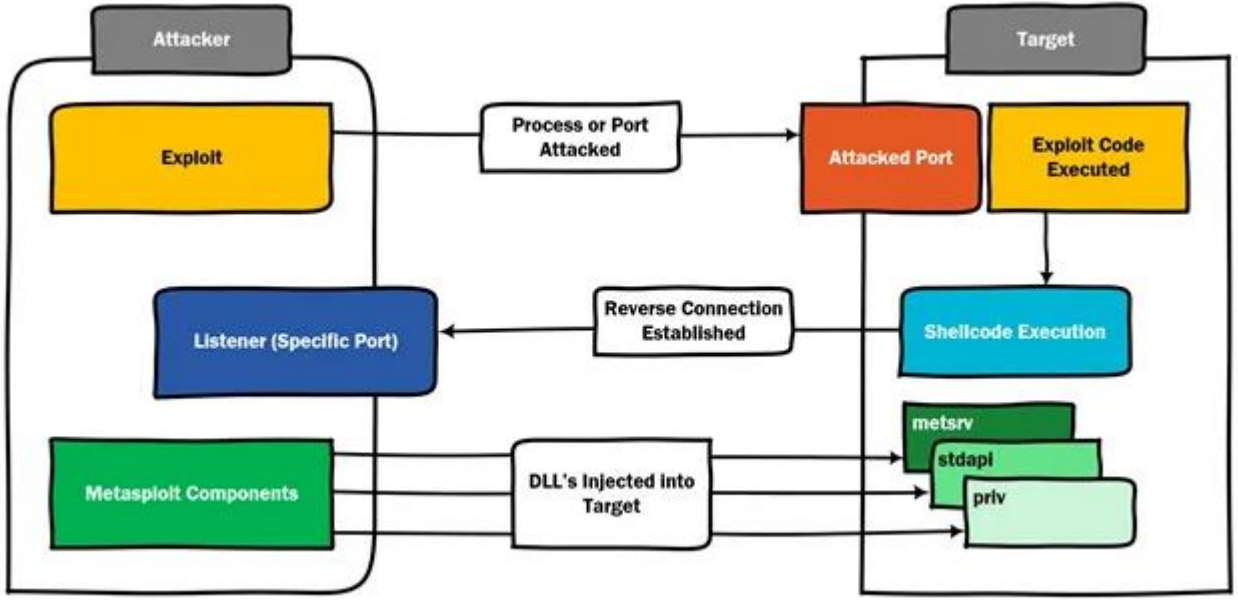
- Payload module establishes a communication channel between the Metasploit framework and the victim host.
- It combines the arbitrary code that is executed as the result of an exploit succeeding.
- To generate (stageless) payloads, first select a payload using the command:
  - msf > use windows/shell_reverse_tcp
  - msf payload(shell_reverse_tcp) > generate -h

Web Server Attack Tools

Metasploit Payload Module

- There are three types of payload modules provides by the Metasploit:
  - **Singles**: It is self-contained, fire-and-forget, completely standalone.
  - **Stagers**: It sets up a network connection between the attacker and victim.
  - **Stages**: It is downloaded by stagers modules.
  - **Stageless(New)**: The entire payload is sent in one hit and executed on the target machine.

Web Server Attack Tools



Web Server Attack Tools

Payload	Staged	Stageless
Reverse TCP	windows/meterpreter/reverse_tcp	windows/meterpreter_reverse_tcp
Reverse HTTPS	windows/meterpreter/reverse_https	windows/meterpreter_reverse_https
Bind TCP	windows/meterpreter/bind_tcp	windows/meterpreter_bind_tcp
Reverse TCP IPv6	windows/meterpreter/reverse_ipv6_tcp	windows/meterpreter_reverse_ipv6_tcp

Web Server Attack Tools

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.

Web Server Attack Tools

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)>

Web Server Attack Tools

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) >
```

2. Wfetch

Web Server Attack Tools

- 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.

3. THC-Hydra and Brutus

Module 14

Web Server Attack Tools

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

Countermeasures

Module 14

1. Patch Management

Module 14

Countermeasures

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.

Countermeasures

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.

Countermeasures

- **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.

Countermeasures

Identifying Appropriate Sources for Updates and Patches

- First make a patch management plan that fits the operational environment and business objectives.
- Find appropriate updates and patches on the home sites of the applications or operating systems' vendors.
- The recommended way of tracking issues relevant to proactive patching is to register to the home sites to receive alerts.

Countermeasures

Installation of a Patch

- 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.

Countermeasures

Implementation and Verification of a Security Patch or Upgrade

- Before installing any patch verify the source.
- Use proper patch management program to validate file versions and checksums before deploying security patches.
- The patch management tool must be able to monitor the patched systems.
- The patch management team should check for updates and patches regularly.

Countermeasures

Microsoft Baseline Security Analyzer (MBSA)

- Microsoft Baseline Security Analyzer (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.

Countermeasures

Microsoft Baseline Security Analyzer 2.3

Microsoft
Baseline Security Analyzer

Report Details for [redacted] (2016-04-27 22:11:15)

Security assessment:
Severe Risk (One or more critical checks failed.)

Computer name:	[redacted]
IP address:	192.168.99.128
Security report name:	[redacted] (2016-4-27 下午 10-11)
Scan date:	2016/4/27 下午 10:11 *** Microsoft recommends scanning on a weekly basis. This report is 52 days old. ***
Scanned with MBSA version:	2.3.2211.0
Catalog synchronization date:	
Security update catalog:	Microsoft Update

Sort Order: Score (worst first)

Security Update Scan Results

Score	Issue	Result
	Silverlight Security Updates	1 security updates are missing. What was scanned Result details How to correct this
	Windows Security Updates	34 security updates are missing. 2 service packs or update rollups are missing. What was scanned Result details How to correct this
	Developer Tools, Runtimes, and Redistributables Security Updates	No security updates are missing. What was scanned Result details
	SQL Server Security Updates	No security updates are missing. What was scanned Result details

Windows Scan Results

Administrative Vulnerabilities

Score	Issue	Result
	Automatic Updates	The Automatic Updates feature is disabled on this computer. What was scanned How to correct this

Countermeasures

Microsoft Baseline Security Analyzer (MBSA)

- Microsoft Baseline Security Analyzer (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.

Countermeasures

Patches and Updates

- Scan for existing vulnerabilities, patch, and update the server software regularly.
- Before applying any service pack, hotfix, or security patch, read and peer review all relevant documentation.
- Apply all updates, regardless of their type on an "as-needed" basis.
- Test the service packs and hotfixes on a representative non-production environment prior to being deployed to production.

Countermeasures

Patches and Updates

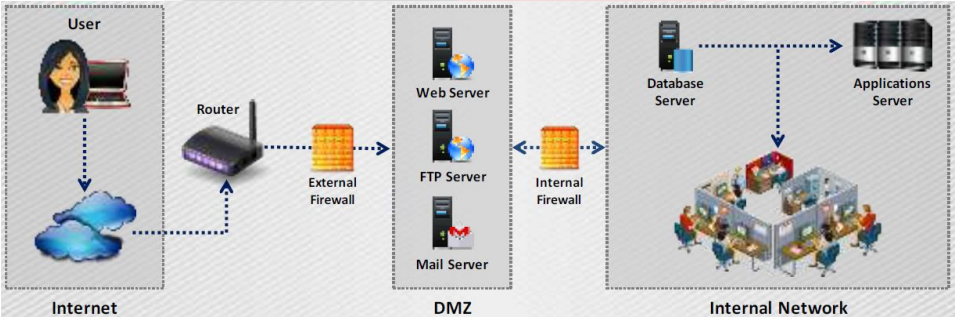
- Ensure that service packs, hotfixes, and security patch levels are consistent on all Domain Controllers (DCs).
- Ensure that server outages are scheduled and a complete set of backup tapes and emergency repair disks are available.
- Have a back-out plan that allows the system and enterprise to return to their original state, prior to the failed implementation.
- Schedule periodic service pack upgrades as part of operations maintenance and never try to have more than two service packs behind.

2. Web Servers in Separate Secure Segment

Module 14

Countermeasures

- 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.



3. Protocols

Countermeasures

- Block all unnecessary ports, Internet Control Message Protocol (ICMP) traffic, and unnecessary protocols such as NetBIOS and SMB.
- Harden the TCP/IP stack and consistently apply the latest software patches and updates to system software.
- 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.
- If remote access is needed, make sure that the remote connection is secured properly, by using tunneling and encryption protocols.
- Disable WebDAV if not used by the application or keep secure if it is required.

4. Accounts

Module 14

Countermeasures

- 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.

Countermeasures

- 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.

5. Files and Directories

Module 14

Countermeasures

- Eliminate unnecessary files within the .jar files.
- Eliminate sensitive configuration information within the byte code.
- Avoid mapping virtual directories between two different servers, or over a network.
- Monitor and check all network services logs, website access logs, database server logs (e.g., Microsoft SQL Server, MySQL, Oracle) and OS logs frequently.
- Disable serving of directory listings.

Countermeasures

- Eliminate the presence of non web files such as archive files, backup files, text files, and header/include files.
- Disable serving certain file types by creating a resource mapping.
- 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.

7. Detecting Web Server Hacking Attempts

Module 14

Countermeasures

- Use Website Change Detection System (WDS) 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.

8. How to Defend Against Web Server Attacks

Module 14

Countermeasures

- **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.
- **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 public key is valid all the way to a trusted root authority.

Countermeasures

- **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.
- **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.

Countermeasures

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.

Registry:

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

Countermeasures

- **IIS Metabase:**
  - Ensure that security related settings are configured appropriately and access to the metabase file is restricted with hardened NTFS permissions.
- **ISAPI Filters:**
  - Remove unnecessary ISAPI filters from the webserver.
- **Shares:**
  - Remove all unnecessary file shares including the default administration shares if not required.
  - Secure the shares with restricted NTFS permissions.

Countermeasures

- **Sites and Virtual Directories**:
  - Relocate sites and virtual directories to non-system partitions and use IIS Web permissions to restrict access.
- **Script Mappings:**
  - 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.
- **Auditing and Logging:**
  - Enable a minimum level of auditing on your web server and use NTFS permissions to protect the log files.

Countermeasures

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.

Countermeasures

- **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.

Countermeasures

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.

HACKING

Is an art, practised through a creative mind.