

Attacking Web Servers

Module 13

Hacking Web Servers

A webserver, which can be referred to as the hardware, the computer, or the software, is the computer application that delivers content that can be accessed through the Internet.

Lab Scenario

Most of the online services are implemented as web applications. Online banking, search engines, email applications, and social networks are just a few examples of such web services. Web content is generated in real time by a software application running at server side. Hackers attack web servers to steal credentials, passwords, and business information. They do this using DoS (DDoS) attacks, SYN flood, ping flood, port scan, sniffing attacks, and social engineering attacks. In the area of Web security, despite strong encryption on the browser-server channel, Web users still have no assurance about what happens at the other end. We present a security application that augments web servers with trusted co-servers composed of high-assurance secure co-processors, configured with a publicly known guardian program. Web users can then establish their authenticated, encrypted channels with a trusted co-server, which can act as a trusted third party in the browser-server interaction. Systems are constantly being attacked, and IT security professionals need to be aware of common attacks on web server applications. Attackers use sniffers or protocol analyzers to capture and analyze packets. If data is sent across a network in clear text, an attacker can capture the data packets and use a sniffer to read the data. In other words, a sniffer can eavesdrop on electronic conversations. A popular sniffer is Wireshark. It's also used by administrators for legitimate purposes. One of the challenges for an attacker is to gain access to the network to capture data. If attackers have physical access to a router or switch, they can connect the sniffer and capture all traffic going through the system. Strong physical security measures help mitigate this risk.

As a penetration (pen) tester or ethical hacker for an organization, you must provide security to the company's web server. You must perform checks on the web server for vulnerabilities, misconfigurations, unpatched security flaws, and improper authentication with external systems.

Lab Objectives

The objective of this lab is to help students learn to detect unpatched security flaws, verbose error messages, and much more.

The objective of this lab is to:

- Perform Web Server Security Reconnaissance
- Detect unpatched security flaws like Shellshock bug
- Crack remote passwords

Lab Environment

To carry out this, you need:

- Windows Server 2016 and Windows Server 2012 machines
- A web browser with Internet access
- Administrative privileges to run tools

Lab Duration

Time: 50 Minutes

Overview of Web Server

Most people think a web server is just the hardware, but a web server is also the software application. A web server delivers web pages on request to clients using the Hypertext Transfer Protocol (HTTP). This means delivery of HTML documents and any additional content that may be included, such as video, images, style sheets, and scripts. Many generic web servers also support server-side scripting using Active Server Pages (ASP), PHP, or other scripting languages. This means that the behavior of the web server can be scripted in separate files, while the actual server software remains unchanged. Web servers are not always used for serving the Web. They can also be found embedded in devices such as printers, routers, and webcams, and serving only a local network. The web server may then be used as a part of a system for monitoring and/or administering the device in question. This usually means that no additional software has to be installed on the client computer, since only a browser is required.

Lab Tasks

Recommended labs to demonstrate webserver hacking:

- Performing Web Server Reconnaissance using **Skipfish**
- Footprinting a Web Server using the **httprecon** Tool
- Footprinting a Web Server using **ID Serve**
- **Uniscan** Web Server Fingerprinting in Kali Linux
- Cracking **FTP Credentials** using **Dictionary Attack**

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS
ABOUT THIS LAB.

Performing Web Server Reconnaissance using Skipfish

Skipfish is a web application (deployed on a webserver) security reconnaissance tool that performs recursive crawl and dictionary-based probes on applications.

Lab Scenario

Every attacker tries to collect as much information as possible about the target web server. The attacker gathers the information and then analyzes the information in order to find lapses in the current security mechanism of the web server.

Lab Objectives

The objective of this lab is to help the students learn how to:

- a. Perform web server reconnaissance using Skipfish

Lab Environment

To perform the lab, you need:

- Windows Server 2016 machine
- Windows Server 2012 virtual machine
- Kali Linux virtual machine

Lab Duration

Time: 15 Minutes

Overview of the Lab

This lab demonstrates how to perform security reconnaissance on a webserver and examine the findings.

Lab Tasks

1. Click **Start**, click the **downwards arrow** and then click **Wampserver64** to launch the WampServer application.



FIGURE 1.1: Starting WampServer

2. Log in to the **Kali Linux** virtual machine and launch a command line terminal.

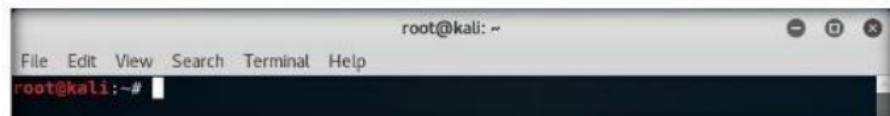


FIGURE 1.2: Launching a Command Line Terminal

3. Perform security reconnaissance on a web server using Skipfish. The target is the wordpress website **http://[IP Address of Windows Server 2012]**.
4. Specify the output directory and load a dictionary file based on the web server requirement. In this lab we are naming output directory as **test**.
5. Type **skipfish -o /root/test -S /usr/share/skipfish/dictionaries/complete.wl http://[IP Address of Windows Server 2012]:8080** and press **Enter**.

Note: IP address may vary in your lab environment.

```
root@kali:~# skipfish -o /root/test -S /usr/share/skipfish/dictionaries/complete.wl http://10.10.10.12:8080
```

A screenshot of a Kali Linux terminal window. The title bar says "root@kali: ~". The command line shows "root@kali:~# skipfish -o /root/test -S /usr/share/skipfish/dictionaries/complete.wl http://10.10.10.12:8080".

FIGURE 1.3: Initiating the Scan

6. Upon receiving this command, Skipfish performs a heavy **brute-force attack** on the web server by using **complete.wl** dictionary file, creates a directory named **test** in the **root** location, and stores the result in **index.html** inside this location.
7. Before beginning the scan, Skipfish displays some tips. Press **Enter** to start the security reconnaissance.

```
Welcome to skipfish. Here are some useful tips:  
1) To abort the scan at any time, press Ctrl-C. A partial report will be written to the specified location. To view a list of currently scanned URLs, you can press space at any time during the scan.  
2) Watch the number requests per second shown on the main screen. If this figure drops below 100-200, the scan will likely take a very long time.  
3) The scanner does not auto-limit the scope of the scan; on complex sites, you may need to specify locations to exclude, or limit brute-force steps.  
4) There are several new releases of the scanner every month. If you run into trouble, check for a newer version first, let the author know next.  
More info: http://code.google.com/p/skipfish/wiki/KnownIssues  
NOTE: The scanner is currently configured for directory brute-force attacks, and will make about 241435 requests per every fuzzable location. If this is not what you wanted, stop now and consult the documentation.  
Press any key to continue (or wait 60 seconds)...
```

A screenshot of a Kali Linux terminal window. The title bar says "root@kali: ~". The command line shows "root@kali:~#". The screen displays the Skipfish welcome message and tips, followed by a note about configuration, and a prompt to press any key to continue.

FIGURE 1.4: Initiating the Scan

8. Skipfish scans the web server as shown in the following screenshot:

```
root@kali: ~
File Edit View Search Terminal Help
- 10.10.10.12 - 0:00:54.7188.1/s), 131963 kB in, 20045 kB out (2780.7 kB/s)
- 10.10.10.12 - 0:00:54.7678.3/s), 132030 kB in, 20071 kB out (2779.7 kB/s)
Scan statistics:: 0:00:54.8348.6/s), 132096 kB in, 20097 kB out (2778.9 kB/s)
Scan statistics:: 0:00:54.8978.2/s), 132162 kB in, 20122 kB out (2777.1 kB/s)
    Scan time : 0:00:54.9408.0/s), 132226 kB in, 20147 kB out (2775.6 kB/s)
    Scan time : 0:00:54.9888.3/s), 132284 kB in, 20170 kB out (2774.9 kB/s)
HTTP requests : 108654 (1980.4/s), 132333 kB in, 20189 kB out (2773.7 kB/s)
    Compression : 0 kB in, 0 kB out (0.0% gain)    etried, 0 drops, 0 val
    HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops, 0 val
TCP handshakes : 1149 total (96.2 req/conn)    rged3 dict      4 par, 0 val
    TCP faults : 0 failures, 0 timeouts, 3 purged3 dict      4 par, 0 val
External links : 32 skipped 0 done (36.59%)      3 dict      4 par, 0 val
    Reqs pending : 1907 0 done (36.59%)      3 dict      4 par, 0 val
Database statistics:6 total, 91 done (36.99%)      3 dict      4 par, 0 val
Database statistics:6 total, 91 done (36.99%)      3 dict      4 par, 0 val
    Pivots : 246 total, 92 done (37.40%)      3 dict      4 par, 0 val
    Pivots : 246 total, 93 done (37.80%)      3 dict      4 par, 0 val
    In progress : 0 pending, 140 init, 10 attacks, 3 dict      4 par, 0 val
Missing nodes : 0 spotted dir, 92 file, 0 pinfo, 146 unkn, 4 par, 0 val
    Node types : 1 serv, 3 dir, 100 file, 0 pinfo, 138 unkn, 4 par, 0 val
Issues found : 7 info, 0 warn, 0 low, 0 medium, 0 high impactdates
    Dict size : 2292 words (77 new), 110 extensions, 256 candidates
Signatures : 77 total
```

FIGURE 1.5: Skipfish Scanning the Web Server

9. Note that Skipfish takes some time (approximately 20 minutes) to complete the scan.

Note: You can press **Ctrl+C** to terminate the scan if it is taking longer.

```
root@kali: ~
File Edit View Search Terminal Help
Reqs pending : 31 248 done (99.20%) dict , 4 par, 0 val
Database statistics:0 total, 248 done (99.20%) dict , 4 par, 0 val
Database statistics:0 total, 248 done (99.20%) dict , 4 par, 0 val
    Pivots : 250 total, 248 done (99.20%) dict , 4 par, 0 val
    Pivots : 250 total, 248 done (99.20%) dict , 4 par, 0 val
    In progress : 0 pending, 0 init, 0 attacks, 2 dict , 4 par, 0 val
Missing nodes : 0 spotted dir, 238 file, 0 pinfo, 0 unkn, 4 par, 0 val
    Node types : 1 serv, 7 dir, 238 file, 0 pinfo, 0 unkn, 4 par, 0 val
Issues found : 14 info, 0 warn, 0 low, 0 medium, 0 high impactdates
    Dict size : 2295 words (80 new), 110 extensions, 256 candidates
Signatures : 77 total

[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 250
[+] Looking for duplicate entries: 250
[+] Counting unique nodes: 53
[+] Saving pivot data for third-party tools...
[+] Writing scan description...
[+] Writing crawl tree: 250
[+] Generating summary views...
[+] Report saved to '/root/test/index.html' [0x9b300471].
[+] This was a great day for science!
```

FIGURE 1.6: Completion of the Scan

10. On completion of the scan, Skipfish generates a report and stores it in the **test** directory (in **root** location). Navigate to **Home → test** and double-click **index.html** to view the scan result.

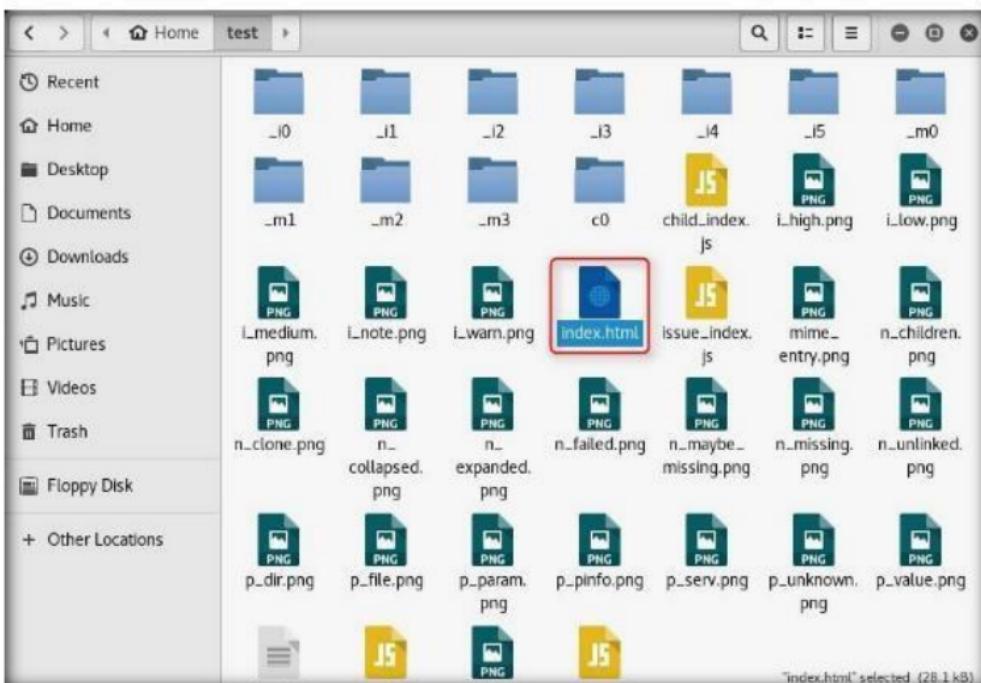


FIGURE 1.7: Viewing the Scan Result

11. The Skipfish crawl result appears in the web browser, displaying the summary overviews of document types and issue types found, as shown in the following screenshot:

Note: The scan result might vary in your lab environment.

A screenshot of a Mozilla Firefox browser window titled 'Skipfish - scan results browser - Mozilla Firefox'. The address bar shows 'file:///root/test/index.html'. The page content is from 'skipfish'.

Crawl results - click to expand:

http://10.10.10.12:8080/ 51
Code: 403, Length: 298, Resources: 10, Issues: 1, Application/xml; charset: iso-8859-1 [show trace]

Document type overview - click to expand:

- application/xhtml+xml (3)
- image/gif (21)
- image/png (19)
- image/svg+xml (1)

Issue type overview - click to expand:

- Numerical filename - consider enumerating (1)
- Incorrect or missing charset (low risk) (1)

FIGURE 1.8: Examining the Scan Result

12. Expand each node to view detailed information regarding the result.
13. Analyze an issue found in the web server. Click a node under the **Issue type overview** section to expand it.
14. Analyze the **Incorrect or missing charset** issue.

Skipfish - scan results browser - Mozilla Firefox

Skipfish - scan results br... +

file:///root/test/Index.html

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter

http://10.10.10.12:8080/ 15 51

Code: 403, length: 280, declared: text/html; charset: iso-8859-1 [show trace +]

Document type overview - click to expand:

- application/xhtml+xml (3)
- image/gif (21)
- image/png (19)
- image/svg+xml (1)

Issue type overview - click to expand:

- Numerical filename - consider enumerating (1)
- Incorrect or missing charset (low risk) (1)**
- Directory listing enabled (6)
- Resource not directly accessible (4)
- New 404 signature seen (2)
- New 'Server' header value seen (1)

NOTE: 100 samples maximum per issue or document type.

FIGURE 1.9: Examining the Scan Result

15. Observe the **URL** of the webpage associated with the vulnerability. Click the URL.

Skipfish - scan results browser - Mozilla Firefox

Skipfish - scan results br... +

file:///root/test/Index.html

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter

http://10.10.10.12:8080/ 15 51

Code: 403, length: 280, declared: text/html; charset: iso-8859-1 [show trace +]

Document type overview - click to expand:

- application/xhtml+xml (3)
- image/gif (21)
- image/png (19)
- image/svg+xml (1)

Issue type overview - click to expand:

- Numerical filename - consider enumerating (1)
- Incorrect or missing charset (low risk) (1)**
- 1. http://10.10.10.12:8080/icons/apache_pb.svg [show trace +]
- Directory listing enabled (6)
- Resource not directly accessible (4)
- New 404 signature seen (2)
- New 'Server' header value seen (1)

FIGURE 1.10: Examining the Scan Result

16. The webpage appears as shown in the following screenshot:



FIGURE 1.11: Examining the Scan Result

17. The php version webpage appears, displaying the details related to the machine, as well as the other resources associated with the web server infrastructure and php configuration.
18. Click **show trace** next to the URL to examine the vulnerability in detail.

Skipfish - scan results browser - Mozilla Firefox

Powered By: Apache 2.4

file:///root/test/Index.html#

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter

http://10.10.10.12:8080/index.html [15]

Code: 403, length: 298, decoded: Text/html, detected: application/xhtml+xml; charset: iso-8859-1 | show trace +

Document type overview - click to expand:

- application/xhtml+xml (3)
- image/gif (23)
- image/png (19)
- image/svg+xml (1)

Issue type overview - click to expand:

- Numerical filename - consider enumerating (1)
- Incorrect or missing charset (low risk) (1)
 - 1. http://10.10.10.12:8080/icons/apache_pb.svg [show trace +]
- Directory listing enabled (6)
- Resource not directly accessible (4)
- New 404 signature seen (2)
- New 'Server' header value seen (1)

file:///root/test/Index.html# issue or document type.

FIGURE 1.12: Examining the HTTP Trace

19. A HTTP trace window appears on the webpage, displaying the complete **HTML session**, as shown in the following screenshot:

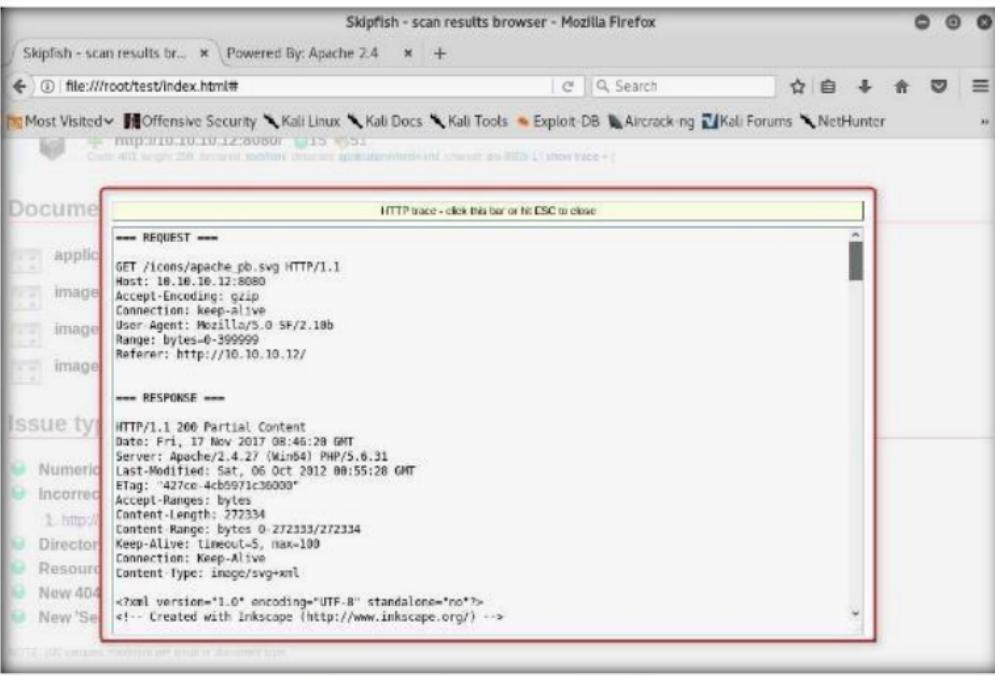


FIGURE 1.13: Examining the HTTP Trace

Note: If the window does not appear properly, hold down the **Ctrl** key and click the link.

20. Examine other vulnerabilities, and patch them in the process of securing the web server.

**PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS
ABOUT THIS LAB.**

Internet Connection Required

Yes

No

Platform Supported

Classroom

iLabs

Footprinting a Web Server using the httprecon Tool

The **httprecon** project undertakes research in the field of web server fingerprinting, also known as **http fingerprinting**.

Lab Scenario

Web applications can publish information, interact with Internet users, and establish an e-commerce/e-government presence. However, if an organization is not rigorous in configuring and operating its public website, it may be vulnerable to a variety of security threats. Although the threats in cyberspace remain largely the same as in the physical world (e.g., fraud, theft, vandalism, and terrorism), they are far more dangerous. Organizations can face monetary losses, damage to reputation, or legal action if an intruder successfully violates the confidentiality of their data. DoS attacks are easy for attackers to attempt because of the number of possible attack vectors, the variety of automated tools available, and the low skill level needed to use the tools. DoS attacks, as well as threats of initiating DoS attacks, are also increasingly being used to blackmail organizations. To be an expert ethical hacker and pen tester, you must understand how to perform footprinting on webservers.

Lab Objectives

The objective of this lab is to help students learn to footprint web servers. It will teach you how to:

- Use the httprecon tool
- Get webserver footprint

Lab Environment

To carry out the lab, you need:

- The **Httprecon** tool, available at **Z:\CEH-Tools\CEHv10 Module 13 Hacking Web Servers\Web Server Footprinting Tools\httprecon**. You can download the latest version of **httprecon** from the link

<http://www.computech/projekte/httprecon>. If you decide to download the **latest version**, then screenshots shown in the lab might differ.

- Windows Server 2016
- A web browser with Internet access
- Administrator privileges

Lab Duration

Time: 5 Minutes

Overview of httprecon

Httprecon is a tool for advanced **web server** fingerprinting, similar to **httpprint**. The goal is highly **accurate** identification of **httpd** implementations.

Lab Tasks

1. In Windows Server 2016 machine, navigate to **Z:\CEH-Tools\CEHv10 Module 13 Hacking Web Servers\Web Server Footprinting Tools\httprecon** and double-click **httprecon.exe** to launch the application.
2. If an **Open File - Security Warning** pop-up appears, click **Run**.
3. The main window of **httprecon** appears, as shown in the following screenshot:

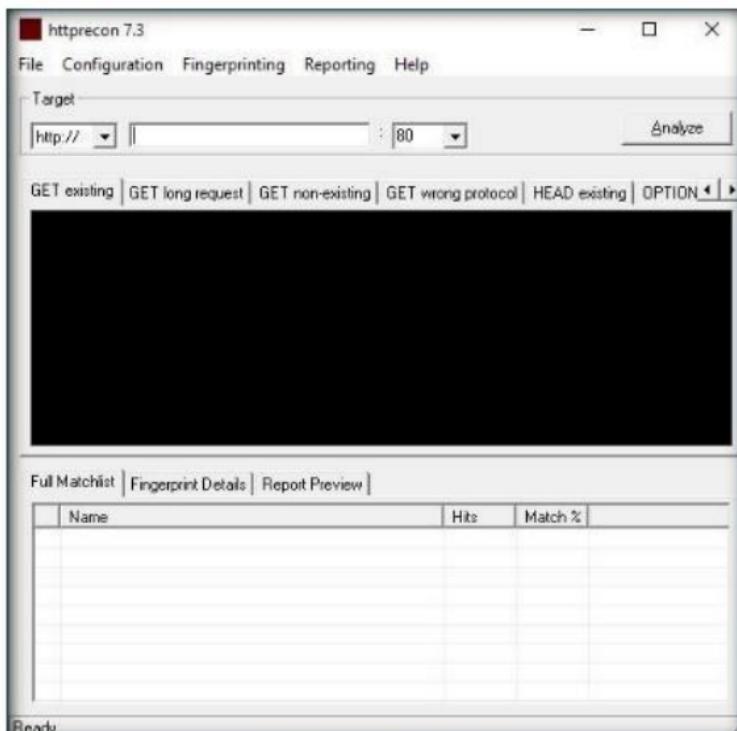


FIGURE 2.1: httprecon main window

- Enter the website URL (here, **www.certifiedhacker.com**) that you want to **footprint** and select the **port number (80)** in the **Target** section.
- Click **Analyze** to start analyzing the entered website.
- A **footprint** of the website as shown in the following screenshot:

The screenshot shows the httprecon 7.3 interface. At the top, there's a menu bar with File, Configuration, Fingerprinting, Reporting, and Help. Below the menu is a toolbar with a red box around the 'Analyze' button. The main area has tabs for Target (Apache 2.0.46) and Analyze. The Analyze tab is selected, showing a red box around the URL input field containing 'www.certifiedhacker.com' and the port dropdown set to '80'. Below the tabs is a row of buttons: GET existing, GET long request, GET non-existing, GET wrong protocol, HEAD existing, OPTION, and a right arrow. A large black box contains the server response headers. At the bottom, there's a Matchlist table and a status message 'Ready.'

Name	Hits	Match %
Apache 2.0.46	79	100
Apache 2.2.3	76	96.20...
Apache 2.0.55	75	94.93...
Apache 2.2.2	75	94.93...
Microsoft IIS 6.0	75	94.93...
Apache 1.3.37	74	93.67...
Apache 2.0.54	74	93.67...
Apache 2.2.22	74	93.67...

FIGURE 2.2: The footprint result of the entered website

- Scroll down the **Get existing** tab, and observe the server used (**Microsoft IIS**), its version (**6.0**), and the server-side application used to develop the webpages (**ASP.NET**).
- When attackers obtain this information, they research the vulnerabilities present in **ASP.NET** and **IIS version 6.0** and try to exploit them, which results in either full or partial control over the web application.

9. Click the **GET long request** tab, which lists all the GET requests. Then click the **Fingerprint Details** tab.

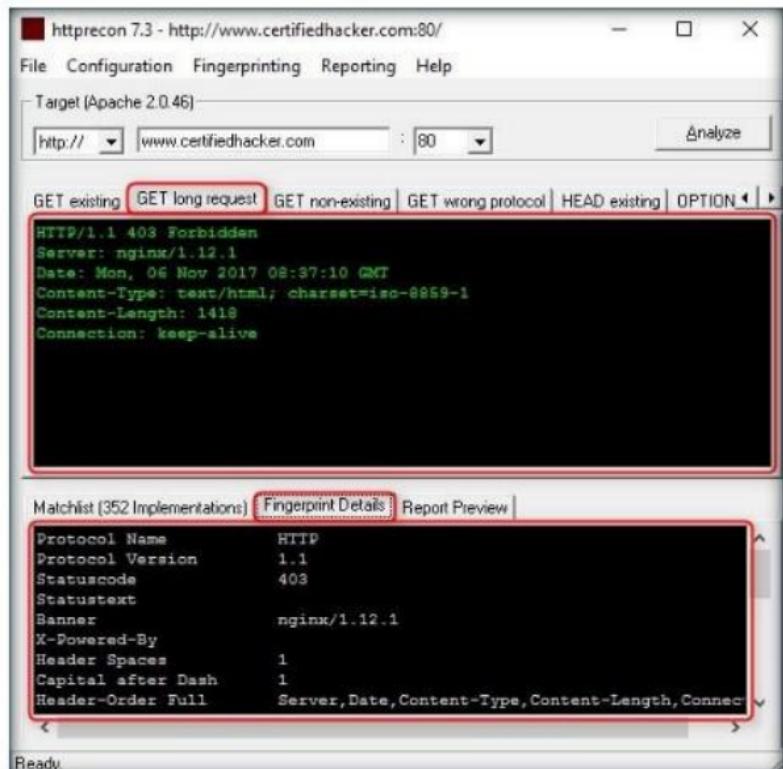


FIGURE 2.3: The fingerprint and GET long request result of the entered website

10. The details displayed in the screenshot above include the name of the protocol the website is using, and its version.
11. By obtaining this information, attackers can manipulate the vulnerabilities in HTTP to perform malicious activities such as sniffing over the HTTP channel, which might result in revealing sensitive data such as user credentials.

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required

Yes

No

Platform Supported

Classroom

iLabs

Footprinting a Web Server using ID Serve

ID Serve is a simple, free, small (26 Kbytes), and fast general-purpose Internet server identification utility.

Lab Scenario

Pen testers must be familiar with banner grabbing techniques to monitor servers and ensure compliance and appropriate security updates. This technique also helps in locating rogue servers or determining the role of servers within a network. In this lab you will learn the banner grabbing technique to determine a remote target system using ID Serve. In order to be an expert ethical hacker and pen tester, its important to understand how to footprint a webserver.

Lab Objectives

This lab shows how to footprint webservers and how to use ID Serve. It teaches how to:

- Use the ID Serve tool
- Get a webserver footprint

Lab Environment

To carry out the lab, you need:

- ID Serve located at **Z:\CEH-Tools\CEHv10 Module 13 Hacking Web Servers\Web Server Footprinting Tools\ID Serve**. You can also download the latest version of **ID Serve** from the link <http://www.grc.com/id/idserve.htm>. If you decide to download the latest version, then screenshots shown in the lab might differ
- Windows Server 2016
- A Web browser with Internet access
- Administrator privileges to run tools

Lab Duration

Time: 5 Minutes

Overview of ID Serve

ID Serve determines the domain name associated with an IP address. This process is known as a reverse DNS lookup and is useful when checking firewall logs or receiving an IP address. Not all IP addresses that have a forward direction lookup (Domain-to-IP) have a reverse (IP-to-Domain) lookup, but many do.

Lab Tasks

1. In Windows Server 2016 machine, navigate to **Z:\CEH-Tools\CEHv10 Module 13 Hacking Web Servers\Web Server Footprinting Tools\ID Serve** and double-click **idsrv.exe**.
2. If an **Open File - Security Warning** pop-up appears, click **Run**.
3. The main window of ID Server appears. Click the **Server Query** tab.



FIGURE 3.1: Welcome screen of ID Serve

4. In option 1, enter the URL (<http://www.certifiedhacker.com>) you want to **footprint** in the **Enter or copy/paste an Internet server URL or IP address** section.
5. Click **Query the Server** to start querying the website.

6. After the completion of the **query**, ID Serve displays the results of the entered website, as shown in the following screenshot:

The screenshot shows the 'ID Serve' application window. At the top, it says 'Internet Server Identification Utility, v1.02 Personal Security Freeware by Steve Gibson Copyright (c) 2003 by Gibson Research Corp.' Below the title bar are three tabs: 'Background', 'Server Query' (which is selected), and 'Q&A / Help'. A large red box highlights the URL input field, which contains 'http://www.certifiedhacker.com'. Step 1 is numbered 1 next to this field. Below the input field is a button labeled 'Query The Server' with a red border, step 2 is numbered 2 next to it. To the right of the button is a description: 'When an Internet URL or IP has been provided above, press this button to initiate a query of the specified server.' Step 3 is numbered 3 next to the response output area. The output shows server headers: 'Last-Modified: Thu, 10 Feb 2011 11:01:38 GMT', 'Accept-Ranges: none', 'Vary: Accept-Encoding', 'Content-Encoding: gzip', and 'Query complete.'. Step 4 is numbered 4 next to the 'Server identified itself as:' output area, which contains 'nginx/1.12.1'. Step 4 is highlighted with a red box. At the bottom of the window are three buttons: 'Copy', 'Goto ID Serve web page', and 'Exit'.

FIGURE 3.2: ID Serve detecting the footprint

Note: The result might vary in your lab environment.

7. By obtaining this information, attackers may perform vulnerability analysis on of that particular version of webserver and implement various techniques to perform exploitation.

Lab Analysis

Document all the server information.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required

Yes No

Platform Supported

Classroom iLabs

Uniscan Web Server Fingerprinting in Kali Linux

Uniscan is a simple Remote File Include, Local File Include and Remote Command Execution vulnerability scanner.

Lab Scenario

Webserver fingerprinting is an essential task for any penetration tester. Before proceeding to hacking/exploiting a webserver, it is critical for the penetration tester to know the type and version of the webserver as most of the attacks/exploits are specific to the type and version of server being used by the target. These methods help any penetration tester to gain information and analyze their target so that they can perform a thorough test and can deploy appropriate methods for mitigation of such attacks on the server.

Lab Objectives

The objective of this lab is to help the students learn how to perform fingerprinting of a webserver

Lab Environment

To perform the lab, you need:

- A computer running Windows Server 2012
- Kali Linux running as a virtual machine

Lab Duration

Time: 15 Minutes

Overview of Uniscan

Uniscan is an open source project and is preinstalled in kali linux distribution. It is a versatile server fingerprinting tools which not only performs the simple commands like ping, traceroute, nslookup, etc. but can also do static, dynamic and stress checks

on a web server. Apart from scanning websites, uniscan also has the feature of performing automated bing and google searches on provided IPs. Uniscan takes all this and combines them in a comprehensive report file for the user.

Lab Tasks

1. Click **Start** and then click **Wampserver64** to launch the WampServer application. Wait till all the services are running and the wamp icon in the taskbar turns green.

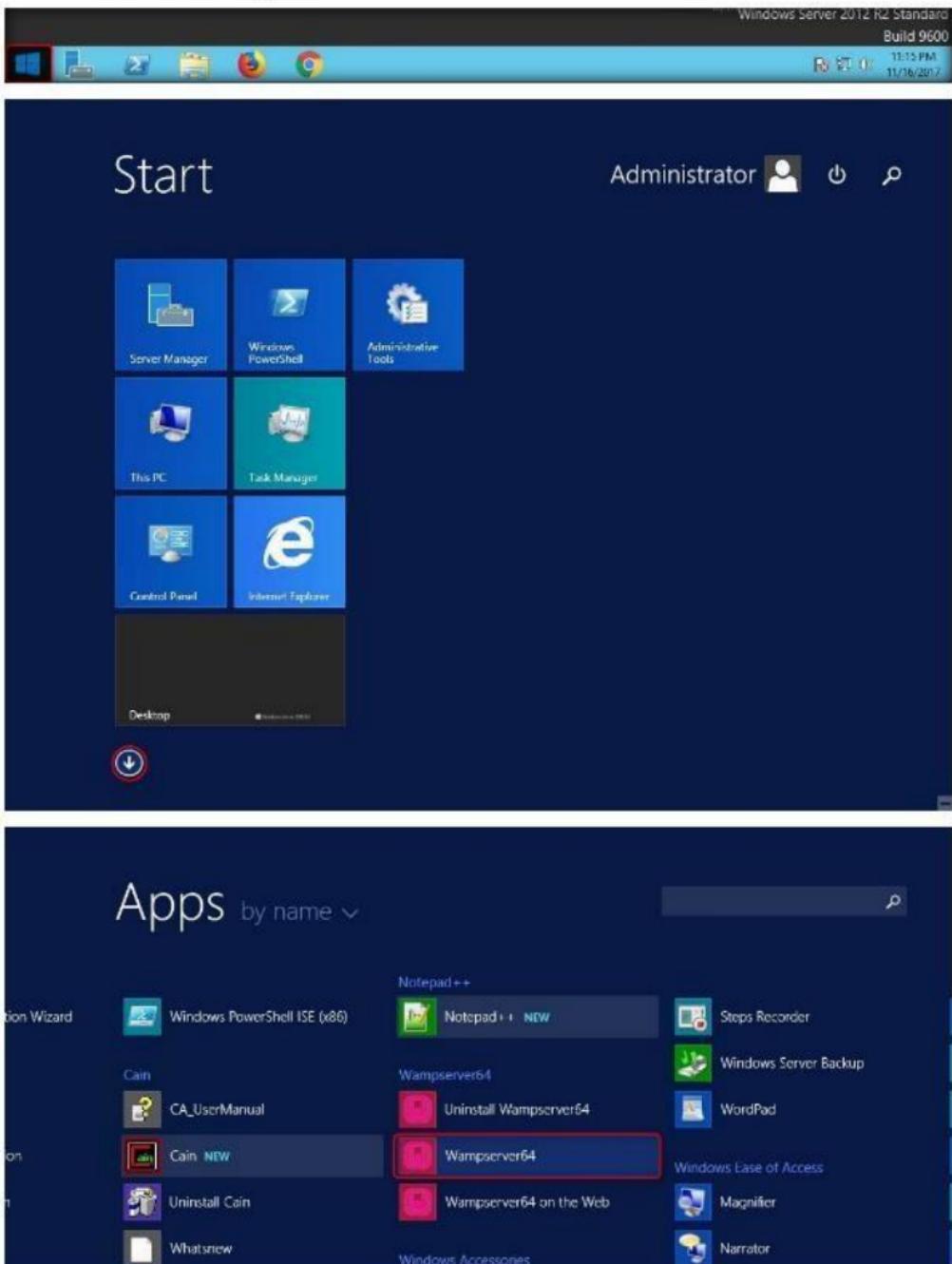


FIGURE 4.1: Starting WampServer

2. Log in to the **Kali Linux** virtual machine and launch a command line terminal.

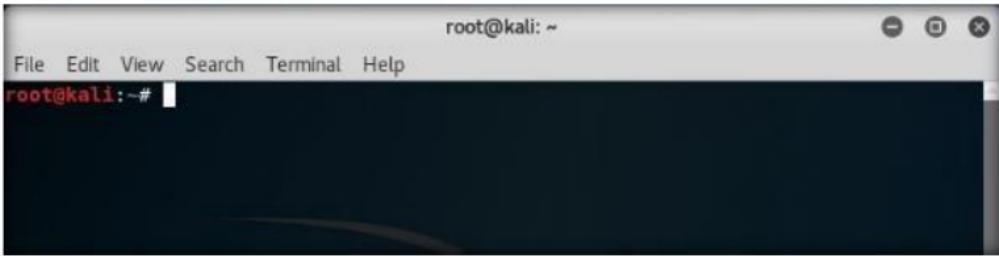


FIGURE 4.2: Launching a Command Line Terminal

3. In the terminal window, type **uniscan -h** and hit **Enter** to display the help options of uniscan.

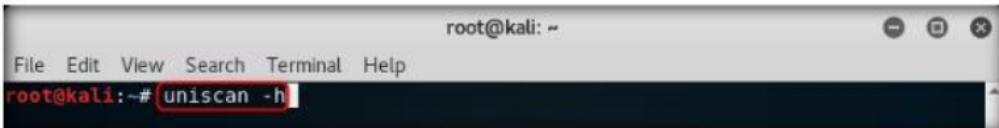


FIGURE 4.3: Uniscan hdp command

4. The help menu appears as shown in the screenshot. First use the **-q** command to search for the directories of the web server.

```
root@kali: ~
File Edit View Search Terminal Help
#####
# Uniscan project      #
# http://uniscan.sourceforge.net/ #
#####
V. 6.3

OPTIONS:
-h      help
-u      <url> example: https://www.example.com/
-f      <file> list of url's
-b      Uniscan go to background
-q      Enable Directory checks
-w      Enable File checks
-e      Enable robots.txt and sitemap.xml check
-d      Enable Dynamic checks
-s      Enable Static checks
-r      Enable Stress checks
-i      <dork> Bing search
-o      <dork> Google search
-g      Web fingerprint
-j      Server fingerprint

usage:
[1] perl ./uniscan.pl -u http://www.example.com/ -qweds
[2] perl ./uniscan.pl -f sites.txt -bqweds
[3] perl ./uniscan.pl -i uniscan
[4] perl ./uniscan.pl -i "ip:xxx.xxx.xxx.xxx"
[5] perl ./uniscan.pl -o "inurl:test"
[6] perl ./uniscan.pl -u https://www.example.com/ -r
```

FIGURE 4.4: Uniscan help menu

5. In the terminal window type **uniscan -u http://10.10.10.12:8080/CEH -q** and hit **Enter** to start the scan for directories.

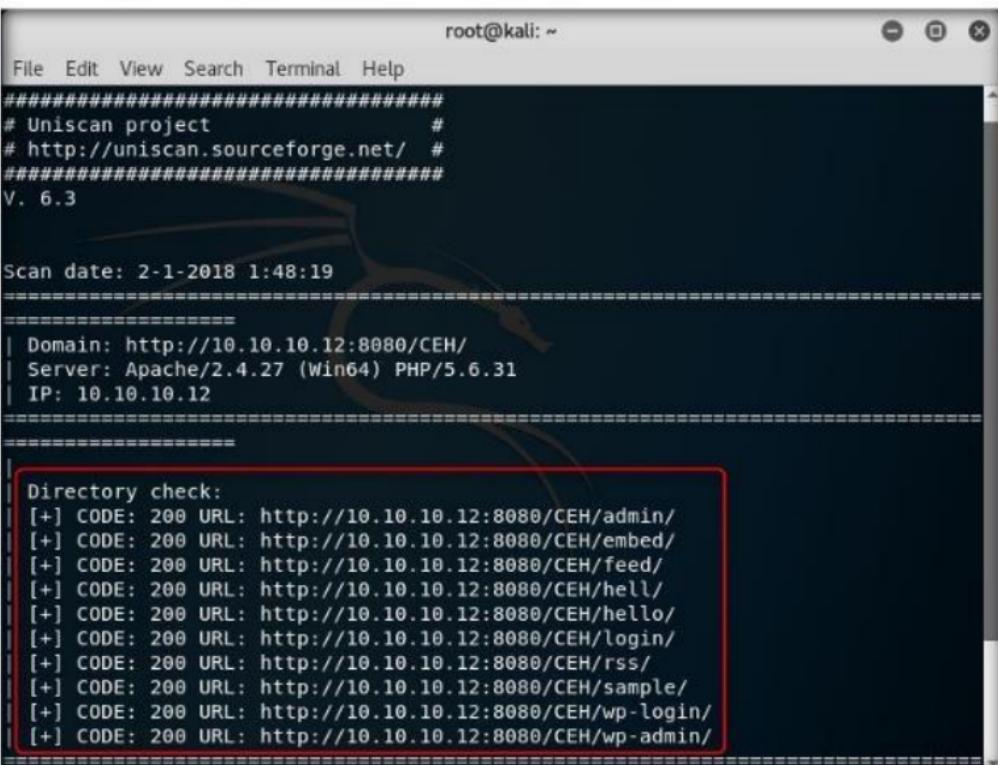


```
root@kali:~# uniscan -u http://10.10.10.12:8080/CEH -q
```

FIGURE 4.5: Run uniscan with -q command

6. Uniscan starts performing different tests on the webserver and finds out **web directories** as shown in the screenshot.

Note: Scroll to analyze the complete output of the scan. It might take approximately 10 minutes for the scan to finish.



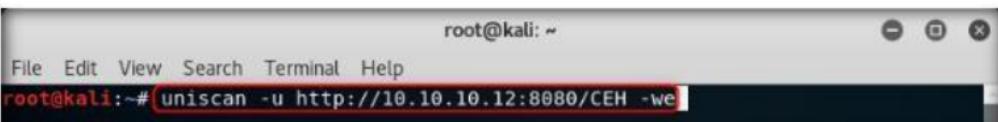
```
root@kali:~# uniscan -u http://10.10.10.12:8080/CEH -q
#####
# Uniscan project          #
# http://uniscan.sourceforge.net/ #
#####
V. 6.3

Scan date: 2-1-2018 1:48:19
=====
=====
| Domain: http://10.10.10.12:8080/CEH/
| Server: Apache/2.4.27 (Win64) PHP/5.6.31
| IP: 10.10.10.12
=====

=====
Directory check:
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/admin/
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/embed/
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/feed/
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/hell/
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/hello/
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/login/
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/rss/
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/sample/
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/wp-login/
[+] CODE: 200 URL: http://10.10.10.12:8080/CEH/wp-admin/
```

FIGURE 4.6: Uniscan showing found directories

7. Now we will run uniscan using two options together. Here **-w** and **-e** are used together to enable file check, robots.txt and sitemap.xml check. In the terminal window type **uniscan -u http://10.10.10.12:8080/CEH -we** and hit **Enter** to start the scan.



```
root@kali:~# uniscan -u http://10.10.10.12:8080/CEH -we
```

FIGURE 4.7: Uniscan command with -we option

8. Uniscan starts file check and shown output as shown in the screenshot.

Note: Scroll to analyze the complete scan result. It might take approximately 10 minutes for the scan to finish.

```
root@kali:~  
File Edit View Search Terminal Help  
| File check:  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/admin/index.php  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/.htaccess  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/index.php  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/LICENSE.TXT  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/license.txt  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/LICENSE.txt  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/readme  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/README  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/readme.html  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/search/htx/sqlqhit.asp  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/search/htx/SQLOHit.asp  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/search/sqlqhit.asp  
| [+] CODE: 200 URL: http://10.10.10.12:8080/CEH/search/SQLOHit.asp  
=====  
| Check robots.txt:  
| Check sitemap.xml:  
=====  
| Crawler Started:  
| Plugin name: Web Backdoor Disclosure v.1.1 Loaded.  
| Plugin name: E-mail Detection v.1.1 Loaded.  
| Plugin name: Code Disclosure v.1.1 Loaded.  
| Plugin name: Timthumb <= 1.32 vulnerability v.1 Loaded.  
| Plugin name: External Host Detect v.1.2 Loaded.
```

FIGURE 4.8: Uniscan displaying scan results

9. Now, we shall use the dynamic testing option by giving the command **-d**.
Type **uniscan -u http://10.10.10.12:8080/CEH -d** and hit Enter to start dynamic scan on the webserver.

```
root@kali:~  
File Edit View Search Terminal Help  
root@kali:~# uniscan -u http://10.10.10.12:8080/CEH -d
```

FIGURE 4.9: Run uniscan with -d option

10. Uniscan starts performing dynamic tests, giving more information about email-IDs, Source code disclosures and external hosts.

Note: Scroll to analyze the complete output of the scan. It might take approximately 10 minutes for the scan to finish.

```
root@kali: ~
File Edit View Search Terminal Help
E-mails:
[+] E-mail Found: license@php.net
[+] E-mail Found: wampserver@wampserver.invalid
[+] E-mail Found: alpha@zforms.ru
[+] E-mail Found: isaac@bennetch.org
[+] E-mail Found: m@tidakada.com
[+] E-mail Found: bennetch@gmail.com
[+] E-mail Found: admin@wampserver.invalid
[+] E-mail Found: klaus.hartl@stilbuero.de
[+] E-mail Found: humbedooh@apache.org
[+] E-mail Found: info@getid3.org
[+] E-mail Found: marijnh@gmail.com
[+] E-mail Found: crawleradmin.t-info@telekom.de
[+] E-mail Found: mike@hyperreal.org
[+] E-mail Found: kevinh@kevcom.com

Source Code Disclosure:
[+] Source Code Found: http://10.10.10.12:8080/phpmyadmin/doc/html/_sources/faq.txt
[+] Source Code Found: http://10.10.10.12:8080/phpmyadmin/doc/html/_sources/setup.txt
[+] Source Code Found: http://10.10.10.12:8080/phpmyadmin/doc/html/_sources/config.txt

Timthumb:

External hosts:
[+] External Host Found: https://dev.mysql.com
[+] External Host Found: https://www.apachefriends.org
[+] External Host Found: http://httpd.apache.org
[+] External Host Found: https://files.phomvadmin.net
```

FIGURE 4.10: Uniscan displaying scan results

11. Then uniscan displays the **PHP info** as given in the screenshot below.

```
root@kali: ~
File Edit View Search Terminal Help
PHPinfo() Disclosure:
[+] phpinfo() page: http://10.10.10.12:8080/?phpinfo=1
  System: Windows NT WIN-0JAQ7QJ8PAI 6.3 build 9600 (Windows Server 2012 R2 Standard Edition) AMD64
  PHP version: 5.6.31
  Apache Version: Apache/2.4.27 (Win64) PHP/5.6.31
  Server Administrator: wampserver@wampserver.invalid
  Server Root: C:/wamp64/bin/apache/apache2.4.27
  DOCUMENT_ROOT: C:/wamp64/www
  SCRIPT_FILENAME: C:/wamp64/www/index.php
  allow_url_fopen: On
  allow_url_include: Off
  disable_functions: <i>no value</i>
  OpenSSL Library Version: OpenSSL 1.0.2k 26 Jan 2017

Ignored Files:
http://10.10.10.12:8080/CEH/wp-content/themes/twentyseventeen/assets/js/skip-link-focus-fix.js?ver=1.0
http://10.10.10.12:8080/CEH/wp-content/themes/twentyseventeen/assets/css/ie8.css?ver=1.0
http://10.10.10.12:8080/CEH/wp-admin/css/ie.min.css?ver=4.9.1
http://10.10.10.12:8080/CEH/wp-content/themes/twentyseventeen/assets/js/jquery.scrollTo.js?ver=2.1.2
http://10.10.10.12:8080/CEH/wp-content/themes/twentyseventeen/style.css?ver=4.9.1
http://10.10.10.12:8080/CEH/wp-content/themes/twentyseventeen/assets/js/global.js?ver=1.0
http://10.10.10.12:8080/CEH/wp-admin/css/install.css?ver=20160228
http://10.10.10.12:8080/phpmyadmin/js/codemirror/lib/codemirror.css?v=4.7.4
http://10.10.10.12:8080/phpmyadmin/js/codemirror/addon/lint/lint.css?v=4.7.4
http://10.10.10.12:8080/CEH/wp-includes/js/jquery/jquery.js?ver=1.12.4
http://10.10.10.12:8080/CEH/wp-includes/css/dashicons.min.css?ver=4.9.1
http://10.10.10.12:8080/phpmyadmin/js/codemirror/addon/hint/show-hint.css?v=4.7.4
http://10.10.10.12:8080/CEH/wp-includes/css/install.min.css?ver=4.9.1
http://10.10.10.12:8080/CEH/wp-includes/js/jquery/jquery-migrate.min.js?ver=1.4.1
```

FIGURE 4.11: Uniscan displaying PHP info

12. After the scanning, navigate to **/usr/share/uniscan/report** and double-click **10.10.10.12.html** to view the scan report.

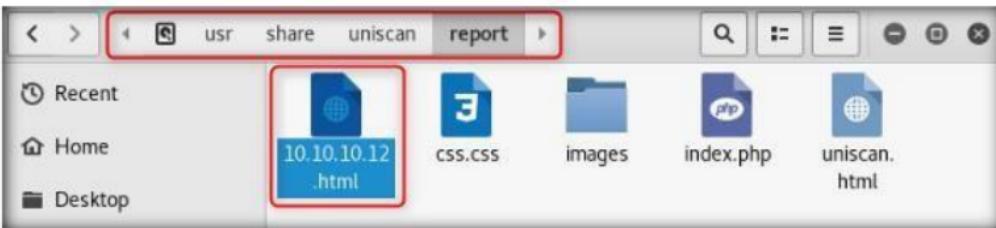


FIGURE 4.12: Scan report generated

13. The report opens in a browser giving you all the **scan details** in a more comprehensive way.

Uniscan Report - Mozilla Firefox

Uniscan Report

file:///usr/share/uniscan/report/10.10.10.12.html 133% C Search

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums

Uniscan
Web Vulnerability Scanner

CODE: 200 URL: http://10.10.10.12:8080/CEH/.htaccess
CODE: 200 URL: http://10.10.10.12:8080/CEH/index.php
CODE: 200 URL: http://10.10.10.12:8080/CEH/LICENSE.TXT
CODE: 200 URL: http://10.10.10.12:8080/CEH/license.txt
CODE: 200 URL: http://10.10.10.12:8080/CEH/LICENSE.txt
CODE: 200 URL: http://10.10.10.12:8080/CEH/README
CODE: 200 URL: http://10.10.10.12:8080/CEH/readme
CODE: 200 URL: http://10.10.10.12:8080/CEH/readme.html
CODE: 200 URL: http://10.10.10.12:8080/CEH/search/htx/sqlhit.asp
CODE: 200 URL: http://10.10.10.12:8080/CEH/search/htx/SQLHit.asp
CODE: 200 URL: http://10.10.10.12:8080/CEH/search/sqlhit.asp
CODE: 200 URL: http://10.10.10.12:8080/CEH/search/SQLHit.asp

[Check robots.txt:](#)

[Check sitemap.xml:](#)

FIGURE 4.13: View the scan report

Lab Analysis

Document the output and give your views about the security posture of the server.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required

Yes

No

Platform Supported

Classroom

iLabs

Cracking FTP Credentials using Dictionary Attack

A dictionary attack bypasses the authentication mechanism employed in a password-protected machine by trying numerous combinations of keywords present in a dictionary file.

Lab Scenario

In this phase of web server hacking, an attacker tries to crack web server passwords. The attacker tries all possible techniques to extract passwords, including password guessing, dictionary attacks, brute force attacks, hybrid attacks, pre-computed hashes, rule-based attacks, distributed network attacks, rainbow attacks, etc. The attacker needs patience, as some of these techniques are tedious and time-consuming. The attacker can also use automated tools such as Brutus, THC-Hydra, etc. to crack web passwords.

Lab Objectives

The objective of this lab is to help the students how to:

- Perform Nmap scan to find whether an ftp port is open
- Perform a dictionary attack using hydra

Lab Environment

To perform the lab, you need:

- Windows Server 2016 virtual machine
- Windows 10 virtual machine
- Kali Linux virtual machine

Lab Duration

Time: 10 Minutes

Overview of Dictionary Attacks

A Dictionary/wordlist contains thousands of words that are used by password cracking tools to break into a password-protected system. Dictionary attacks are often successful because many users insist on using ordinary words as passwords.

Lab Tasks

1. Before beginning this lab, launch the **Windows 10** virtual machine from **VMware Workstation** and log in.
2. Launch the **Kali Linux** virtual machine from **VMware Workstation** and log in.
3. Double-click **CEH-Tools** shared folder on Desktop and navigate to **CEHv10 Module 13 Hacking Web Servers**, and copy the **Wordlists** folder and paste it in the Home directory as shown in the screenshot:

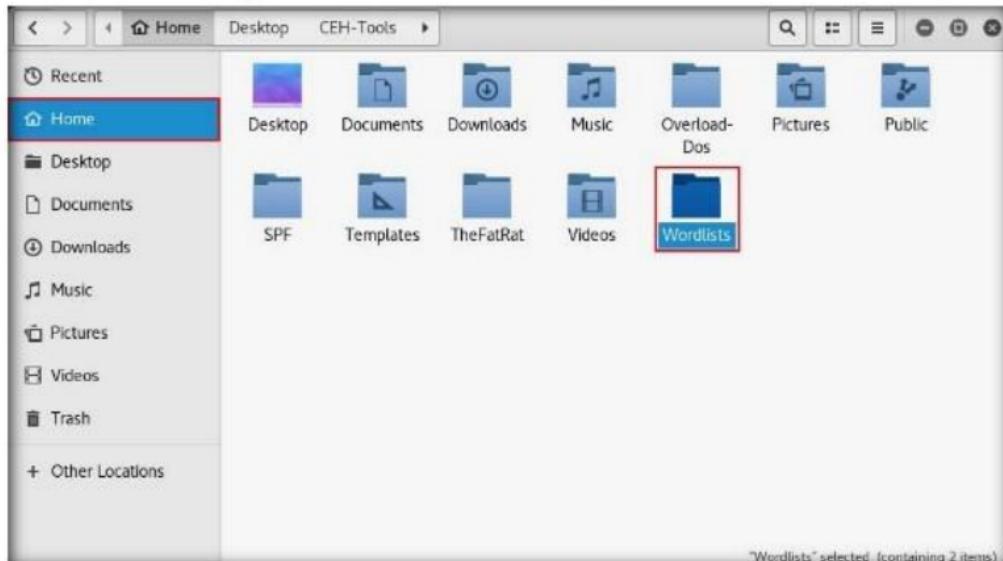


FIGURE 5.1: Wordlists folder in the Home directory

4. Perform an **Nmap scan** on the target machine (**Windows 10**) to check if the FTP port is open.

5. Launch a command line terminal in the **Kali Linux** machine and enter **nmap -p 21 [IP Address of Windows 10]**.

Note: In this lab, the IP Address of **Windows 10** is **10.10.10.10**.

```
root@kali:~# nmap -p 21 10.10.10.10
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-06 05:52 EST
Nmap scan report for 10.10.10.10
Host is up (-0.17s latency).

PORT      STATE SERVICE
21/tcp      open  ftp
MAC Address: [REDACTED] (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.48 seconds
root@kali:~#
```

FIGURE 5.2: Performing Nmap Port Scan

6. Observe that **port 21** is open in **Windows 10**.
7. Check if an FTP server is hosted on the Windows 10 machine.
8. Enter **ftp [IP Address of Windows 10]**. You will be prompted to enter user credentials, which implies that an FTP server is hosted on the machine and requires credentials.

Note: The IP Address of **Windows 10** in this lab is **10.10.10.10**.

```
root@kali:~# ftp 10.10.10.10
Connected to 10.10.10.10.
220 Microsoft FTP Service
Name (10.10.10.10:root):
```

FIGURE 5.3: Test for FTP Server

9. Try to enter random usernames and passwords in an attempt to gain ftp access.

Note: The password you enter will not be visible.

```
root@kali:~# ftp 10.10.10.10
Connected to 10.10.10.10.
220 Microsoft FTP Service
Name (10.10.10.10:root): james
331 Password required
Password:
530 User cannot log in.
Login failed.
Remote system type is Windows_NT.
ftp>
```

FIGURE 5.4: Test Log In

10. Perform an attack on the FTP server in an attempt to gain access to it.
11. This lab uses hydra.
12. Open a new command line terminal.
13. Type **hydra -L /root/Wordlists/Usernames.txt -P /root/Wordlists/Passwords.txt ftp://[IP Address of Windows 10]**.

Note: The IP Address of **Windows 10** in this lab is **10.10.10.10**, This IP Address might vary in your lab environment.

```
root@kali:~$ File Edit View Search Terminal Help
root@kali:~# hydra -L /root/Wordlists/Usernames.txt -P /root/Wordlists/Passwords.txt ftp://10.10.10.10
```

FIGURE 5.5: Attacking the FTP Server

14. Hydra tries various combinations of usernames and passwords (present in the Usernames.txt and Passwords.txt files) on the ftp server, and starts displaying the cracked usernames and passwords, as shown in the following screenshot:

```
root@kali:~$ File Edit View Search Terminal Help
root@kali:~# hydra -L /root/Wordlists/Usernames.txt -P /root/Wordlists/Passwords.txt ftp://10.10.10.10
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-12-19 07:48:54
[DATA] max 16 tasks per 1 server, overall 16 tasks, 41174 login tries (l:238/p:1
73), ~2574 tries per task
[DATA] attacking ftp://10.10.10.10:21/
[21][ftp] host: 10.10.10.10 login: Martin password: apple
[STATUS] 4738.00 tries/min, 4738 tries in 00:01h, 36436 to do in 00:08h, 16 active
```

FIGURE 5.6: Hydra Cracking User Credentials

15. On completion of password cracking, the **cracked credentials** appear as shown in the following screenshot:

```
root@kali:~# hydra -L /root/Wordlists/Usernames.txt -P /root/Wordlists/Passwords.txt ftp://10.10.10.10
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-12-19 07:48:54
[DATA] max 16 tasks per 1 server, overall 16 tasks, 41174 login tries (l:238/p:173)
, ~2574 tries per task
[DATA] attacking ftp://10.10.10.10:21/
[21][ftp] host: 10.10.10.10 login: Martin password: apple
[STATUS] 4738.00 tries/min, 4738 tries in 00:01h, 36436 to do in 00:08h, 16 active
[21][ftp] host: 10.10.10.10 login: Juggyboy password: green
[STATUS] 4728.00 tries/min, 14184 tries in 00:03h, 26990 to do in 00:06h, 16 active
[21][ftp] host: 10.10.10.10 login: Jason password: qwerty
[21][ftp] host: 10.10.10.10 login: Sheila password: test
[STATUS] 4706.00 tries/min, 32942 tries in 00:07h, 8232 to do in 00:02h, 16 active
[STATUS] 4701.62 tries/min, 37613 tries in 00:08h, 3561 to do in 00:01h, 16 active
1 of 1 target successfully completed, 4 valid passwords found
[WARNING] Writing restore file because 8 final worker threads did not complete until end.
[ERROR] 8 targets did not resolve or could not be connected
[ERROR] 16 targets did not complete
Hydra (http://www.thc.org/thc-hydra) finished at 2017-12-19 07:57:39
root@kali:~#
```

FIGURE 5.7: User Credentials Cracked Successfully

16. Try to log in to the ftp server using one of the cracked username and password combinations. In this lab, use Martin's credentials to gain access to the server.
17. Open a command line terminal and enter **ftp [IP Address of Windows 10]**.
18. Enter Martin's user credentials (**Martin/apple**) to check whether you can successfully log in to the server.
19. On entering the credentials, you will be able to successfully log in to the server. An ftp terminal appears as shown in the following screenshot:

```
root@kali:~# ftp 10.10.10.10
Connected to 10.10.10.10.
220 Microsoft FTP Service
Name (10.10.10.10:root): Martin
331 Password required
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> 
```

FIGURE 5.8: Logging in to FTP Server

20. Remotely access the FTP server hosted on the Windows 10 machine.

21. Enter **mkdir Hacked** to create a directory named **Hacked** through the **ftp** terminal.

The screenshot shows a terminal window titled "root@kali: ~". The user has connected to an FTP server at 10.10.10.10. They have logged in as "Martin" and entered the command "mkdir Hacked". The response "257 \"Hacked\" directory created." confirms the success of the operation. A red box highlights the command "mkdir Hacked".

```
File Edit View Search Terminal Help
root@kali:~# ftp 10.10.10.10
Connected to 10.10.10.10.
220 Microsoft FTP Service
Name (10.10.10.10:root): Martin
331 Password required
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> mkdir Hacked
257 "Hacked" directory created.
ftp>
```

FIGURE 5.9: Creating a Directory

22. Switch to the **Windows 10** virtual machine and navigate to C:\FTP.
23. View the directory named **Hacked**, as shown in the following screenshot:

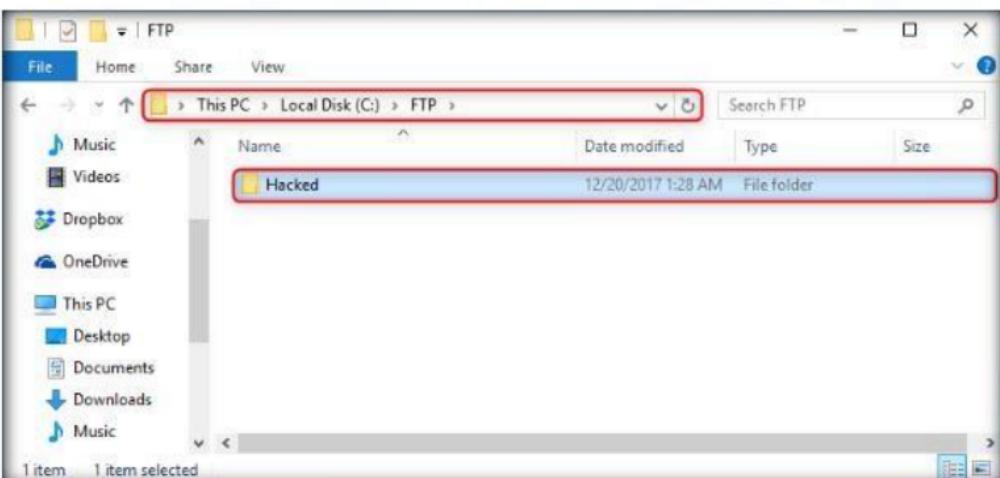


FIGURE 5.10: Viewing the Created Directory in Windows 10

24. You have successfully gained remote access to the **FTP server** by obtaining the credentials.
25. Switch to the **Kali Linux** virtual machine.

26. Enter **help** to view all the other commands which you can use through the FTP terminal.

```
root@kali:~  
File Edit View Search Terminal Help  
230 User logged in.  
Remote system type is Windows_NT.  
ftp> mkdir Hacked  
257 "Hacked" directory created.  
ftp> help  
Commands may be abbreviated. Commands are:  
!  
$ dir mdelete qc site  
account disconnect mdir sendport size  
append exit mget put status  
ascii get mkdir pwd struct  
bell glob mode quit system  
binary hash modtime quote sunique  
bye help mput recv tenex  
case idle newer rstatus tick  
cd image nmap rhelp trace  
cdup ipany nlist rename type  
chmod ipv4 ntrans reset user  
close ipv6 open restart umask  
cr lcd prompt rmdir verbose  
delete ls passive runique ?  
debug macdef proxy send  
ftp> [redacted]  
[redacted]
```

FIGURE 5.11: Viewing the Other FTP Commands

27. On completing the lab, enter **quit** to exit the FTP terminal.

```
root@kali:~  
File Edit View Search Terminal Help  
ftp> mkdir Hacked  
257 "Hacked" directory created.  
ftp> help  
Commands may be abbreviated. Commands are:  
!  
$ dir mdelete qc site  
account disconnect mdir sendport size  
append exit mget put status  
ascii get mkdir pwd struct  
bell glob mode quit system  
binary hash modtime quote sunique  
bye help mput recv tenex  
case idle newer rstatus tick  
cd image nmap rhelp trace  
cdup ipany nlist rename type  
chmod ipv4 ntrans reset user  
close ipv6 open restart umask  
cr lcd prompt rmdir verbose  
delete ls passive runique ?  
debug macdef proxy send  
ftp> quit  
221 Goodbye.  
root@kali:~# [redacted]
```

FIGURE 5.12: Exiting the FTP Shell

28. You have gained **remote access** to FTP server.

Lab Analysis

Document the output.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS
ABOUT THIS LAB.

Internet Connection Required

Yes No

Platform Supported

Classroom iLabs