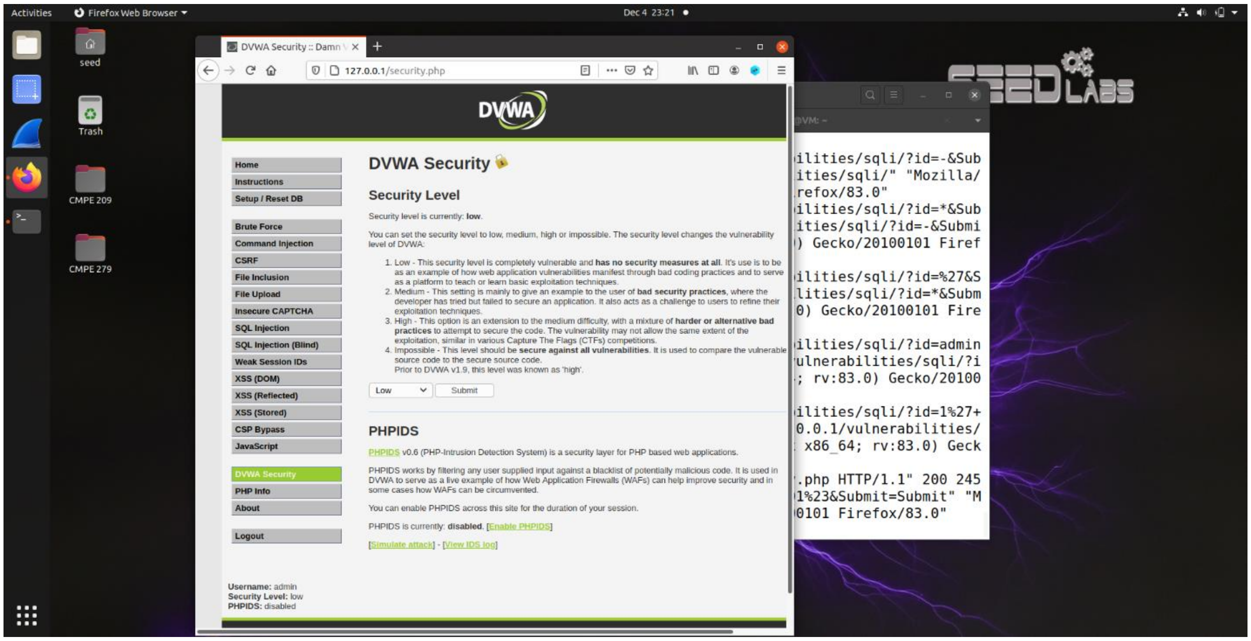
CMPE 279
Assignment - 3

Task -1 : Perform a SQL injection attack and retrieve the list of users in the user database (Low security)



DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

Low ▾ Submit

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

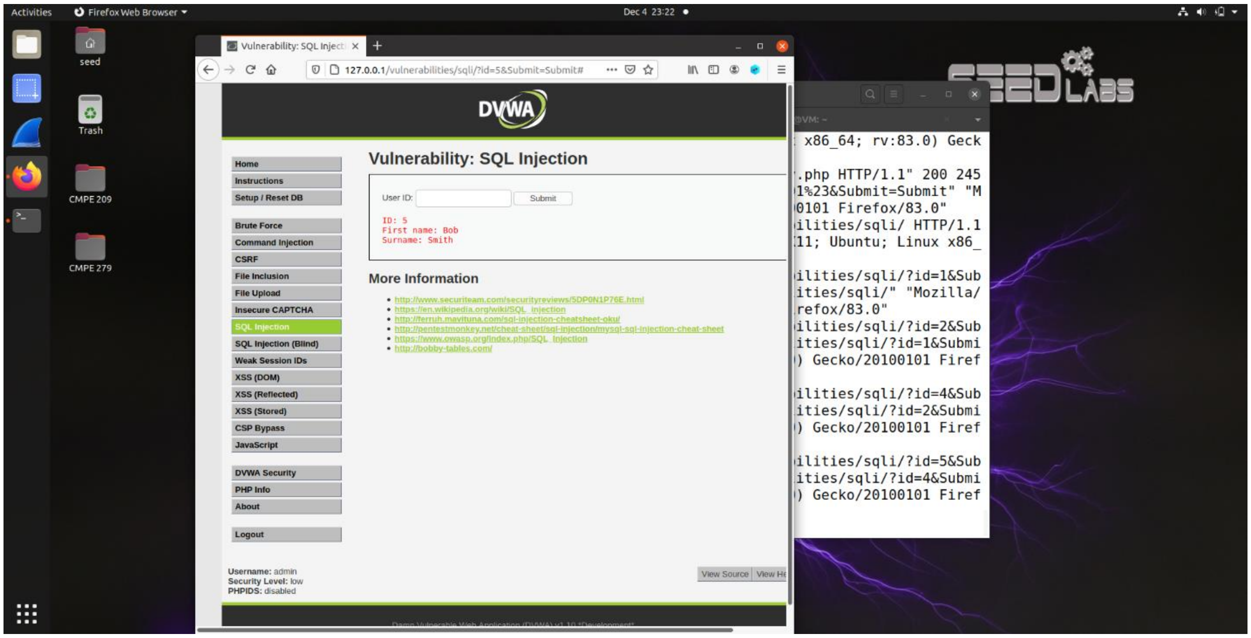
PHPIDS is currently: **disabled**. [Enable PHPIDS]

[Simulate attack] - [View IDS log]

Username: admin
Security Level: low
PHPIDS: disabled

```
ilities/sqli/?id=-&Sub
ities/sqli/" "Mozilla/
refox/83.0"
ilities/sqli/?id=*&Sub
ities/sqli/?id=-&Submi
) Gecko/20100101 Firef

ilities/sqli/?id=%27&S
lities/sqli/?id=*&Subm
0) Gecko/20100101 Fire

ilities/sqli/?id=admin
ulnerabilities/sqli/?i
; rv:83.0) Gecko/20100

ilities/sqli/?id=1%27+
0.0.1/vulnerabilities/
 x86_64; rv:83.0) Geck

.php HTTP/1.1" 200 245
1%23&Submit=Submit" "M
0101 Firefox/83.0"
```

Try getting single user data using user id : 5



Vulnerability: SQL Injection

User ID: Submit

```
ID: 5
First name: Bob
Surname: Smith
```

More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

Username: admin
Security Level: low
PHPIDS: disabled

```
 x86_64; rv:83.0) Geck

.php HTTP/1.1" 200 245
1%23&Submit=Submit" "M
0101 Firefox/83.0"
ilities/sqli/ HTTP/1.1
11; Ubuntu; Linux x86_

ilities/sqli/?id=1&Sub
ities/sqli/" "Mozilla/
refox/83.0"
ilities/sqli/?id=2&Sub
ities/sqli/?id=1&Submi
) Gecko/20100101 Firef

ilities/sqli/?id=4&Sub
ities/sqli/?id=2&Submi
) Gecko/20100101 Firef

ilities/sqli/?id=5&Sub
ities/sqli/?id=4&Submi
) Gecko/20100101 Firef
```

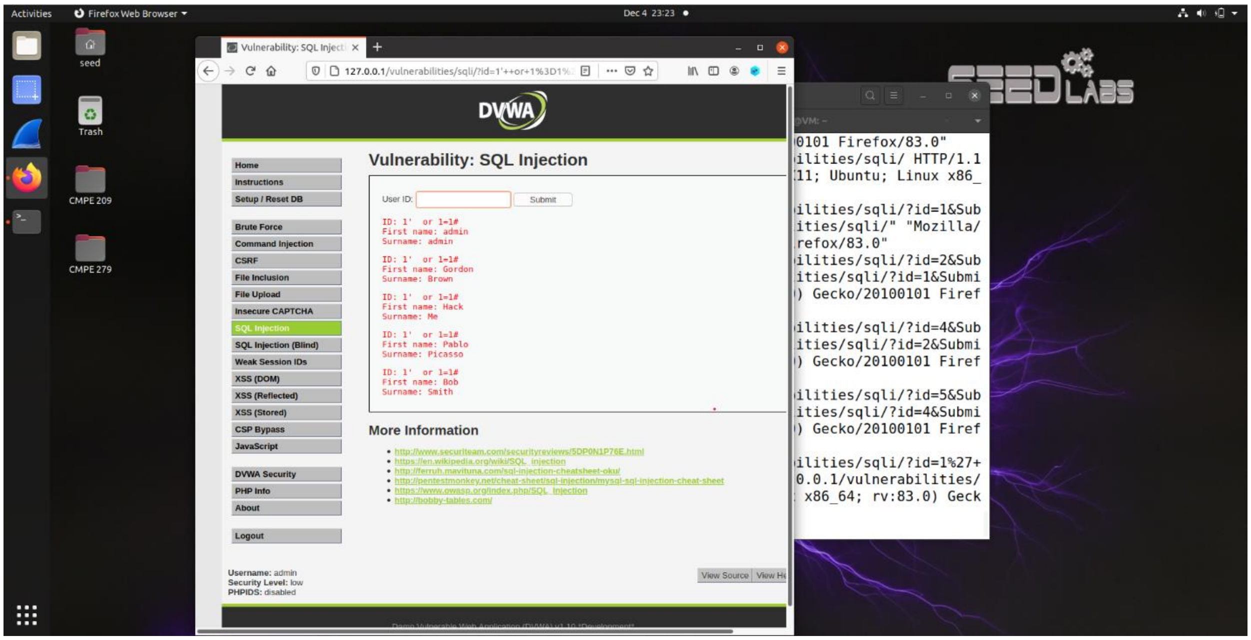
CMPE 279
Assignment - 3

Getting all Users in Database

I have used 1' or 1=1

'OR '1'='1 returns all rows from the queried table

This input string closes the first string parameter, which is supposed to be the user Id, and adds an always true condition. All database users will now be returned since the query was treated as True.



127.0.0.1/vulnerabilities/sqli/?id=1'++or+1%3D1%

Vulnerability: SQL Injection

User ID: Submit

```
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
```

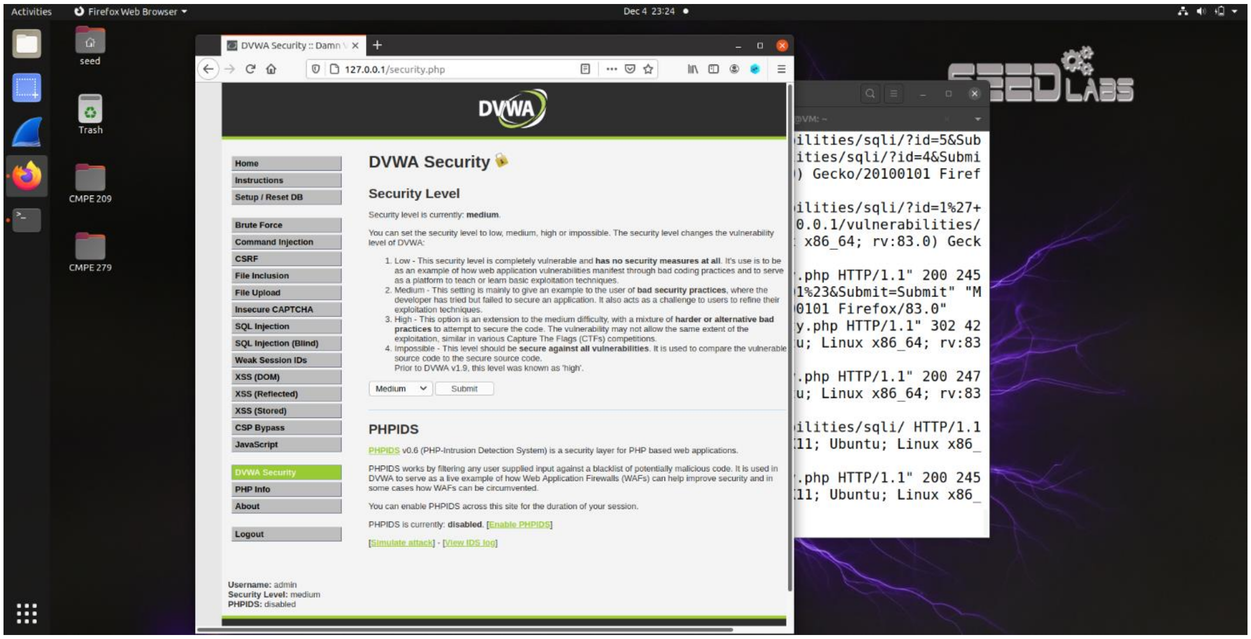
More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

Username: admin
Security Level: low
PHPIDS: disabled

```
0101 Firefox/83.0"
ilities/sqli/ HTTP/1.1
11; Ubuntu; Linux x86_

ilities/sqli/?id=1&Sub
ities/sqli/" "Mozilla/
refox/83.0"
ilities/sqli/?id=2&Sub
ities/sqli/?id=1&Submi
) Gecko/20100101 Firef

ilities/sqli/?id=4&Sub
ities/sqli/?id=2&Submi
) Gecko/20100101 Firef

ilities/sqli/?id=5&Sub
ities/sqli/?id=4&Submi
) Gecko/20100101 Firef

ilities/sqli/?id=1%27+
0.0.1/vulnerabilities/
 x86_64; rv:83.0) Geck
```

Task 2 : Switching the security level in DVWA to "Medium"



127.0.0.1/security.php

DVWA Security

Security Level

Security level is currently: **medium**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

Medium Submit

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [Enable PHPIDS]

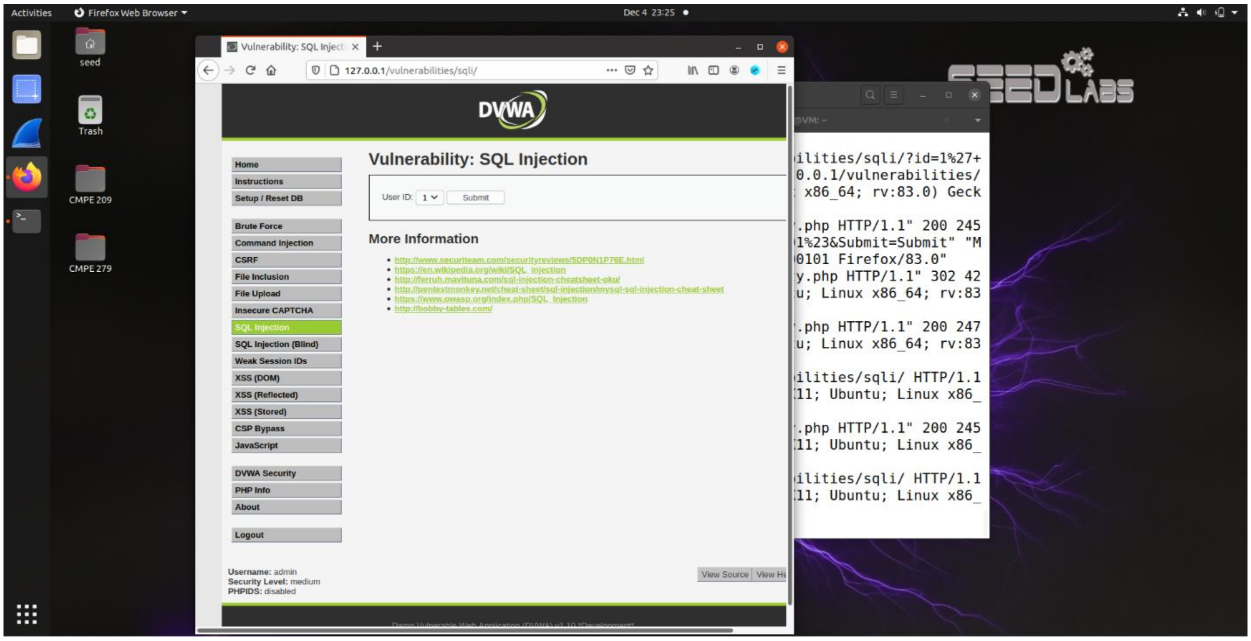
[Simulate attack] - [View IDS log]

Username: admin
Security Level: medium
PHPIDS: disabled

```
ilities/sqli/?id=5&Sub
ities/sqli/?id=4&Submi
) Gecko/20100101 Firef

ilities/sqli/?id=1%27+
0.0.1/vulnerabilities/
 x86_64; rv:83.0) Geck

.php HTTP/1.1" 200 245
1%23&Submit=Submit" "M
0101 Firefox/83.0"
y.php HTTP/1.1" 302 42
u; Linux x86_64; rv:83

.php HTTP/1.1" 200 247
u; Linux x86_64; rv:83

ilities/sqli/ HTTP/1.1
11; Ubuntu; Linux x86_

.php HTTP/1.1" 200 245
11; Ubuntu; Linux x86_
```

CMPE 279
Assignment - 3

I have changed the security to medium it won't work since there isn't a textbox where we can enter our input; instead, there is a dropdown list from which we must select the id from the pre-provided list. They have therefore solved the issue by removing the textbox entry and replacing it with dropdown options.



127.0.0.1/vulnerabilities/sqli/

Vulnerability: SQL Injection

User ID: 1

More Information

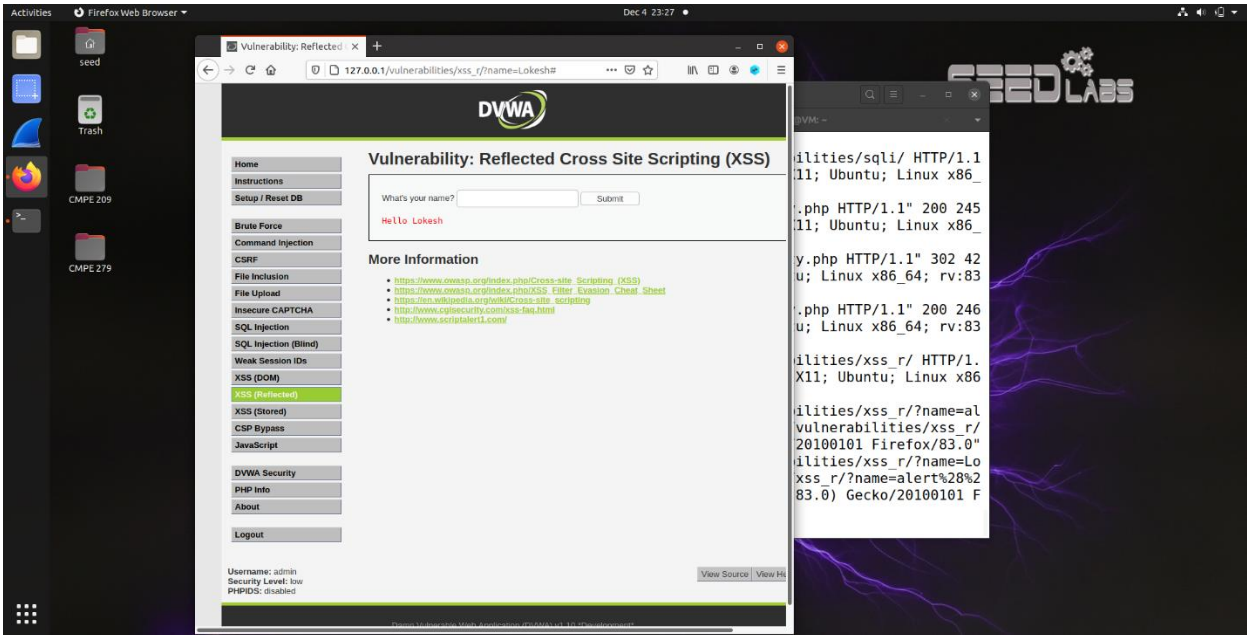
- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

Username: admin
Security Level: medium
PHPIDS: disabled

```
ilities/sqli/?id=1%27+
0.0.1/vulnerabilities/
 x86_64; rv:83.0) Geck

.php HTTP/1.1" 200 245
1%23&Submit=Submit" "M
0101 Firefox/83.0"
y.php HTTP/1.1" 302 42
u; Linux x86_64; rv:83

.php HTTP/1.1" 200 247
u; Linux x86_64; rv:83

ilities/sqli/ HTTP/1.1
11; Ubuntu; Linux x86_

.php HTTP/1.1" 200 245
11; Ubuntu; Linux x86_

ilities/sqli/ HTTP/1.1
11; Ubuntu; Linux x86_
```

Task 3 : reflected XSS attack

DVWA security level set to low



127.0.0.1/vulnerabilities/xss_r/?name=Lokesh#

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello Lokesh

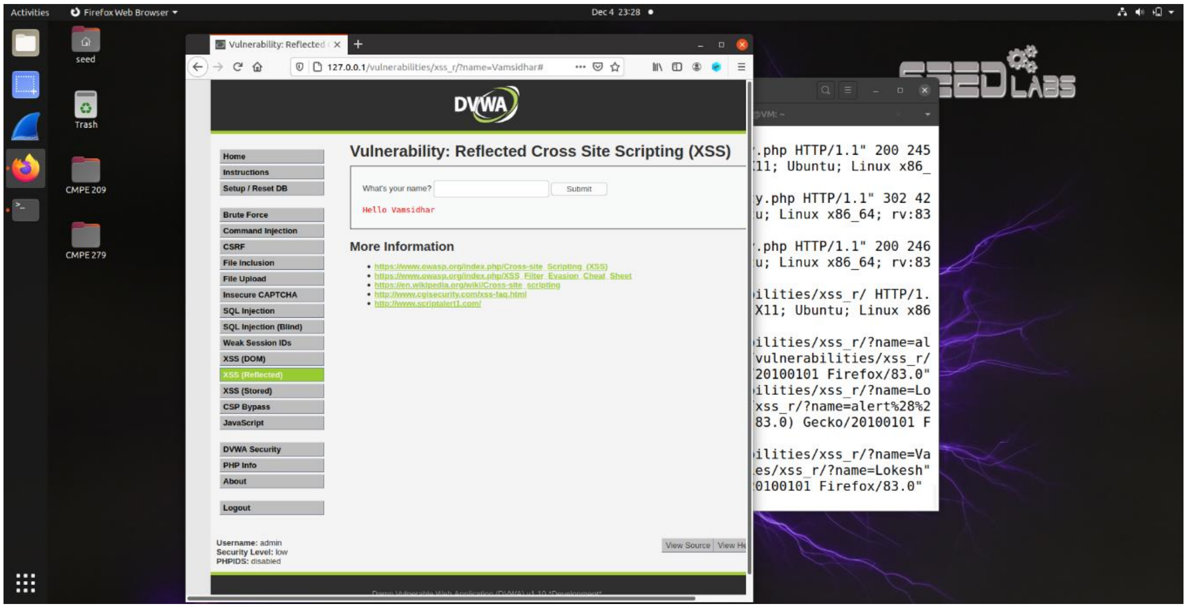
More Information

- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Username: admin
Security Level: low
PHPIDS: disabled

```
ilities/sqli/ HTTP/1.1
11; Ubuntu; Linux x86_

.php HTTP/1.1" 200 245
11; Ubuntu; Linux x86_

y.php HTTP/1.1" 302 42
u; Linux x86_64; rv:83

.php HTTP/1.1" 200 246
u; Linux x86_64; rv:83

ilities/xss_r/ HTTP/1.
X11; Ubuntu; Linux x86

ilities/xss_r/?name=al
vulnerabilities/xss_r/
20100101 Firefox/83.0"
ilities/xss_r/?name=Lo
xss_r/?name=alert%28%2
83.0) Gecko/20100101 F
```

CMPE 279
Assignment - 3



127.0.0.1/vulnerabilities/xss_r/?name=Vamsidhar#

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello Vamsidhar

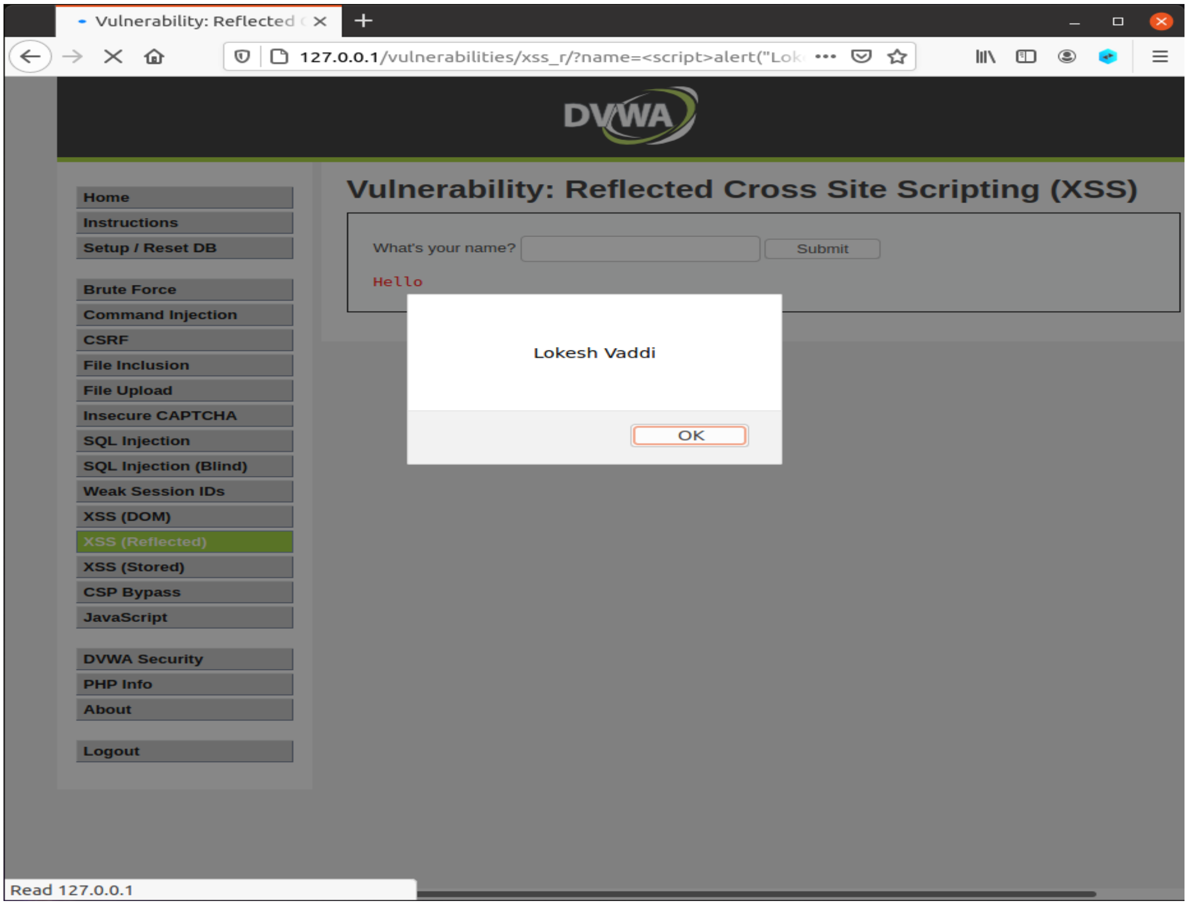
More Information

- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Username: admin
Security Level: low
PHPIDS: disabled

```
.php HTTP/1.1" 200 245
11; Ubuntu; Linux x86_

y.php HTTP/1.1" 302 42
u; Linux x86_64; rv:83

.php HTTP/1.1" 200 246
u; Linux x86_64; rv:83

ilities/xss_r/ HTTP/1.
X11; Ubuntu; Linux x86

ilities/xss_r/?name=al
vulnerabilities/xss_r/
20100101 Firefox/83.0"
ilities/xss_r/?name=Lo
xss_r/?name=alert%28%2
83.0) Gecko/20100101 F

ilities/xss_r/?name=Va
es/xss_r/?name=Lokesh"
0100101 Firefox/83.0"
```

The input I used for the XSS attack was <script>alert("Lokesh Vaddi")/script>. This works because the webpage wrongly escapes the input string. Any user input is accepted, and the result is inserted into the DOM. When the website is updated with user input, the malicious JavaScript I just provided will then be run, and the alert of our choice will show.



127.0.0.1/vulnerabilities/xss_r/?name=<script>alert("Lok

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello

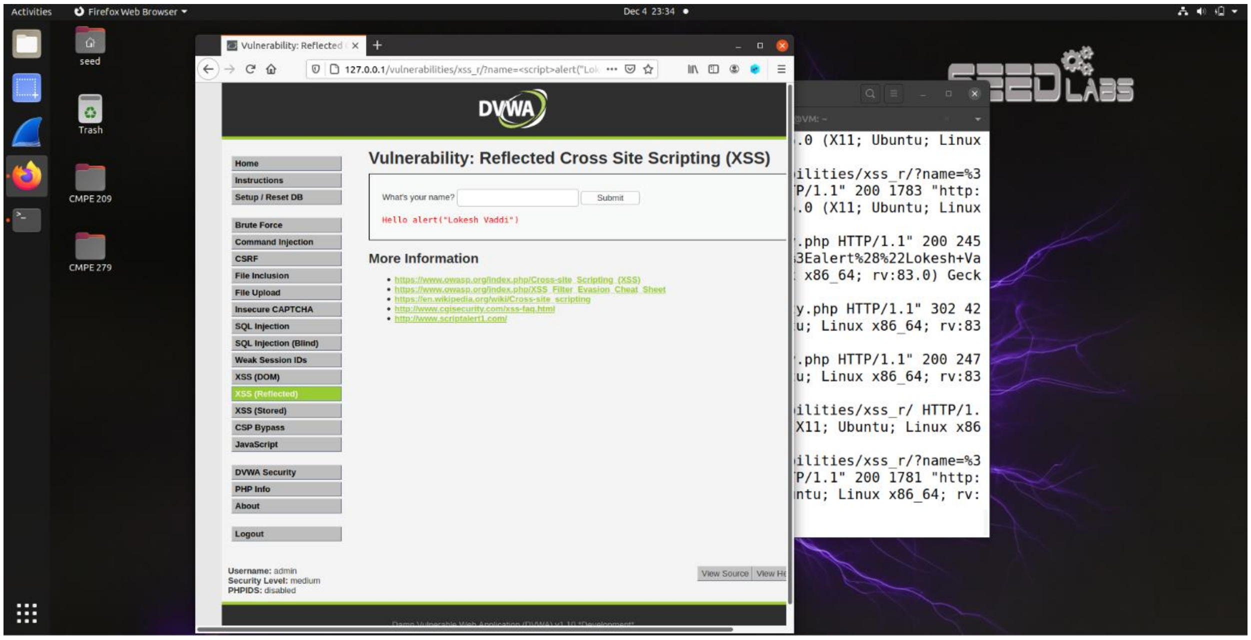
Lokesh Vaddi

Read 127.0.0.1

CMPE 279
Assignment - 3

switching the security level in DVWA to "Medium".

XSS does not work website might have validated user input.



127.0.0.1/vulnerabilities/xss_r/?name=<script>alert("Lok

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

```
Hello alert("Lokesh Vaddi")
```

More Information

- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Username: admin
Security Level: medium
PHPIDS: disabled

```
.0 (X11; Ubuntu; Linux

ilities/xss_r/?name=%3
P/1.1" 200 1783 "http:
.0 (X11; Ubuntu; Linux

.php HTTP/1.1" 200 245
3Ealert%28%22Lokesh+Va
 x86_64; rv:83.0) Geck

y.php HTTP/1.1" 302 42
u; Linux x86_64; rv:83

.php HTTP/1.1" 200 247
u; Linux x86_64; rv:83

ilities/xss_r/ HTTP/1.
X11; Ubuntu; Linux x86

ilities/xss_r/?name=%3
P/1.1" 200 1781 "http:
ntu; Linux x86_64; rv:
```

Team Members :

Lokesh Vaddi – 015999607

Vamsidhar Reddy - 015999191