

Forum:[Hack Forums](#)

Thread: [SQLMAP Volume 2: TIPS & TRICKS](#)

Hooded Robin - May 4, 2011, 10:52 pm

This is a follow up to my first article: SQLMAP Introduction and Tutorial. I have updated my blog to include a new article covering some basic tips and tricks for using sqlmap, and thought I would share here as well.

MyBlog Article:<http://kaoticcreations.blogspot.com/2011/05/sqlmap-volume-2-tips-tricks.html>

Original HF Post: <https://hackforums.net/showthread.php?tid=1251146>

SQLMAP Volume 2: Tips & Tricks:

1) You need to be safe, so this is crucial. I assume you already have a trusted VPN connected and now to be extra ninja we are also going to route traffic through proxies. We can accomplish this by simply setting the '--proxy=PROXY' parameter which will enable use of proxies during scanning.

EX: sqlmap.py -u <http://site.com/example.php?id=1> --proxy=http://proxysite.com:PORT

You can also supply credentials with the '--proxy-cred=user:pass' argument

2) You can speed scans up by increasing the THREAD count, since the default is set to 3 request at a time.

2a) If you set this beyond the default you may want to also set the '--delay=DELAY' parameter to allow some strategic delays between requests (or to speed up even further, although I have a feeling this leads to errors which other tools are prone to (cough..Havij)

2b) You can use the '-o' switch to enable all of the optimiZation features at once

2c) If you are performing time based attacks or blind injections then it may be a better idea to leave thread count alone and use the '--predict-output' argumernt which will save you a bit of time and will allow SQLMAP to perform some analysis on found results in session file to help speed things up (can slow things down in other scenarios where the '-o' option is much better choice.

3) You can have the tool load different USER-AGENTS in case a site or page has restrictions based on browser type. This can be accomplished by adding the '-a <insert/path/to/file>' parameter to your command string. It needs to be followed by the path to the file containing the USER-AGENTS to be used. You can run a few searches on Google to find some common ones or how to come up with your own custom USER-AGENT.

NOTE: this may have been replaced by --user-agent=<insert user agent details> with latest update

3a) there is a default list provided with installation that you can add to or see how they are modeled located at: '~\sqlmap\txt\user-agent.txt'

*3b) you can also just let SQLMAP do the deciding by issuing the '--random-agent'

4) If you need to fake out the referer in the request due to restrictions on server side you can change the default values by using the '--referer=<http://www.insert/REFERER/page/.com/>'

5) Need credentials to perform a more in depth scan? No problem! We can set the credentials using the '--auth-cred' followed by credentials in the typical 'user':"pass' format

EX: --auth-cred=user:password or --auth-cred=admin:IhazYourPazword!

6) You can load targets using GOOGLE and DORKS, by using the '-g' argument followed by the search dork in quotes ""

EX: sqlmap.py -g "inurl:index.php?id= site:us"

NOTE: It will work on any targets found in an interactive manner but only the arguments passes with original command will be used on each target so make sure you use some basics but not too much.

EX: sqlmap.py -g "inurl:index.php?id=" -b --current-user --current-db --is-dba --dbs

6a) You can also process more than one site request at a time from a file by changing the standard '-u' to '-r <path/to/load/HTTP/requests/from/>'

7) You can save and resume scans and data retrieved in session file (in "output" folder) to save time, or to pick things back up where you last left off. You need to use the '-s

<insert/path/to/session/file>/session' parameter to tell it where to look to parse results from.

NOTE: Point it directly at the session file and not just the folder it is in or it wont load properly.

You can also use the recorded scan details to learn a LOT. It contains info on the queries used to get all the details as well as everything found, so it is a good reference and learning tool (all dumps are converted to .CSV files which are above and beyond the log and session files that are saved).

8) You can use the '--eta' paramter to have the scans keep an ETA so you have an idea of how long things are going to take. This come in handy when doing dumps of database to give you an idea of how long it thinks it will take (very useful when blind or trying to make determination as to whether or not it makes sense to dump a questionable table in full or just what you need. Remember if it looks like it will take too long you can use the '-s' argument to pick it back up later See step 7 above)

9) You can use '--flush-session' to clear out the results stored for a session file if you need to just start over, or in case the admin has come along and made some changes since your last visit ;)

10) The last one I will leave you with is very important if you are working on Linux machine, like Backtrack, to make sure you have the latest version of SQLMAP as it is constantly being worked on and new developments constantly being released. This can be accomplished by a quick addition of the '--update' argument.

EX: sqlmap.py --update

NOTE: if you are working on Windows then you can either check the SQLMAP homepage often for updated version to download or you can try the Windows SVN client called TortoiseSVN GUI, and can be found here and is available for both 32 and 64 bit computers: <http://tortoisesvn.net/>

BONUS Material:

1) When attacking version 4 databases with no information schema to rely on there are still several options with SQLMAP:

1a) You can use the '--common-tables' and '--common-columns' arguments to try and bruteforce the table and column names. The default list is much greater than Havij and other tools and can easily be added to if you want to beef it up, it as well as many others used can be found at:

~\sqlmap\txt*.txt

2) If you dont get any positive results from injection but you have a gut feeling that the site is vulnerable then you can try increasing the '--level' or '--risk' arguments beyond the default level of 1. When you do this it will allow SQLMAP to perform more intense attacks and check for additional injection points such as the cookie field, user-agent field, and even the referer field. I typically will add '--level 3 --risk 3' to my command string if I dont get what I am looking for and have a strong feeling there is an injection point that is some how being overlooked.

3-X)=Updates recently introduced:

--batch allows you to walk away from the terminal and let SQLMAP make all of the decisions on your behalf fro all instances where it would usually prompt for interaction, a true hands free experience

--schema which can be used to enumeate the databases or schema

--parse-errors which tells SQLMAP to parse the error messages from response pages received

from queries

--mobile which like the user-agent argument allows SQLMAP to act as if it was a mobile device, which can be handy in testing many of the new mobile.site.com spin-off domains to help reach ever expanding consumer markets with very little concern for security or sanitization :)

Last, but certainly not least as this can be very helpful in adding to YOUR security:

--tor which enables SQLMAP to perform queries through the default TOR proxy setup address

Mini-Tut on the Switch: '--forms'

Say that you want to test against SQL injections a huge search form or you want to test a login bypass (typically only two input fields named like username and password), you can either pass to sqlmap the request in a request file (-r), set the \$POST data accordingly (--data)...

...or let SQLMAP do it for you!

Both of the above mentioned instances, and many others, appear as <form> and <input> tags in HTML response bodies and this is where this switch comes into play.

Provide SQLMAP with --forms as well as the page where the form can be found as the target url (-u) and SQLMAP will request the target url for you, parse the forms it has and guide you through to test for SQL injection on those form input fields (parameters) rather than the target url provided.

Hope these help you with the tool SQLMAP a little bit more. Still working on adding some more instructions and tutorials regarding the additional features that interact with the filesystem, system registry, and actual command execution with a little help from Metasploit. I am also planning a separate short article on how to perform injections via \$POST :)

Until next time...

Later - H.R.

TickTockMan - September 26, 2011, 8:09 am

Good tutorial! This should be helpful to those unfamiliar with sqlmap's more advanced features.

Hooded Robin - September 26, 2011, 8:11 am

and here are some more related ones if interested:

SQLMAP Basic Tutorial: <https://hackforums.net/showthread.php?tid=1251146>

Video: <https://hackforums.net/showthread.php?tid=1402691>

SQLMAP Tips & Tricks: <https://hackforums.net/showthread.php?tid=1270466>

SQLMAP and Metasploit in BT5: <https://hackforums.net/showthread.php?tid=1300361>

SQLMAP+Burpsuite for POST INjections: <https://hackforums.net/showthread.php?tid=1279189>