

## SQL Injection Attack:

Below is the SQL injection attack performed on the WebApp.

SQL injection attack results: a. WAF Enabled:

When Web Application Firewall is enabled SQL injection attack is forbidden with 403 error code as shown below.

The screenshot shows a terminal window on an Ubuntu 64-bit system. The terminal is running a SQLmap attack on a Joomla! website. The output shows various warnings and informational messages from SQLmap, including target URL content analysis, parameter detection, and the final result of the attack.

```
Mon 14:59**
amrithprabhu@ubuntu: ~/Desktop/R/Testing/sqlmap

File Edit View Search Terminal Tabs Help

centos@ip-10-0-4-215:~$ sshpass
[14:58:19] [INFO] testing if the target URL content is stable
[14:58:19] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. If no dynamic nor injectable parameters are detected, or in case
[14:58:19] [WARNING] how do you want to proceed? ((Continue/(s)tring/(r)egex/(q)uot) C
[14:58:21] [INFO] searching for dynamic content
[14:58:21] [CRITICAL] target URL content appears to be heavily dynamic. sqlmap is going to retry the request(s)
[14:58:22] [WARNING] target URL content appears to be too dynamic, switching to '--test-only'
[14:58:22] [INFO] testing if (custom) POST parameter 'JSON email' is dynamic
[14:58:22] [INFO] (custom) POST parameter 'JSON email' appears to be dynamic
[14:58:22] [INFO] (custom) POST parameter 'JSON email' appears to be dynamic
[14:58:22] [INFO] testing for SQL injection on (custom) POST parameter 'JSON email'
[14:58:22] [INFO] testing 'AND Boolean-based blind - WHERE or HAVING clause'
[14:58:22] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[14:58:22] [INFO] testing 'MySQL >= 5.0 error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[14:58:22] [INFO] testing 'PostgreSQL and error-based - WHERE or HAVING clause'
[14:58:22] [INFO] testing 'Microsoft SQL Server/Sybase and error-based - WHERE or HAVING clause (IN)'
[14:58:22] [INFO] testing 'Oracle and error-based - WHERE or HAVING clause (SQLType)'
[14:58:22] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[14:58:22] [INFO] testing 'MySQL inline queries'
[14:58:22] [INFO] testing 'PostgreSQL inline queries'
[14:58:22] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[14:58:22] [INFO] testing 'PostgreSQL >= 8.1 stacked queries (comment)'
[14:58:22] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[14:58:22] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[14:58:22] [INFO] testing 'MySQL >= 5.0.12 and time-based blind'
[14:58:22] [INFO] testing 'PostgreSQL >= 8.1 and time-based blind'
[14:58:22] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[14:58:22] [INFO] testing 'Oracle and time-based blind'
[14:58:22] [INFO] testing 'Generic UNION query (NULL) - 1 to 30 columns'
[14:58:22] [WARNING] (custom) POST parameter 'JSON email' does not seem to be injectable
[14:58:22] [INFO] testing if (custom) POST parameter 'JSON password' is dynamic
[14:58:22] [INFO] (custom) POST parameter 'JSON password' appears to be dynamic
[14:58:22] [INFO] (custom) POST parameter 'JSON password' appears to be dynamic
[14:58:22] [INFO] testing for SQL injection on (custom) POST parameter 'JSON password'
[14:58:22] [INFO] testing 'AND Boolean-based blind - WHERE or HAVING clause'
[14:58:22] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[14:58:22] [INFO] testing 'MySQL >= 5.0 and error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[14:58:22] [INFO] testing 'PostgreSQL and error-based - WHERE or HAVING clause'
[14:58:22] [INFO] testing 'Microsoft SQL Server/Sybase and error-based - WHERE or HAVING clause (IN)'
[14:58:22] [INFO] testing 'Oracle and error-based - WHERE or HAVING clause (SQLType)'
[14:58:22] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[14:58:22] [INFO] testing 'MySQL inline queries'
[14:58:22] [INFO] testing 'PostgreSQL inline queries'
[14:58:22] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[14:58:22] [INFO] testing 'PostgreSQL >= 8.1 stacked queries (comment)'
[14:58:22] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[14:58:22] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[14:58:22] [INFO] testing 'MySQL >= 5.0.12 and time-based blind'
[14:58:22] [INFO] testing 'PostgreSQL >= 8.1 and time-based blind'
[14:58:22] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[14:58:22] [INFO] testing 'Oracle and time-based blind'
[14:58:22] [INFO] testing 'Generic UNION query (NULL) - 1 to 30 columns'
[14:58:22] [WARNING] (custom) POST parameter 'JSON password' does not seem to be injectable
[14:58:22] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind
[14:58:22] [WARNING] of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=spacecomment') and/or switch '--random-agent'
[14:58:22] [WARNING] [CRITICAL] http error codes detected during run!
403 (Forbidden) - 96 times, 404 (Not Found) - 1 times, 405 (Method Not Allowed) - 3 times

[*] ending @ 14:58:47 (2019-04-08)

amrithprabhu@ubuntu: ~/Desktop/R/Testing/sqlmap
```

b. WAF disabled:

When Web Application Firewall is disabled, firewall doesn't forbid the attack, however the API call is denied with error 502 (Bad Gateway) as the web application framework provides the security against SQL injection attack.

```
Activities Terminal Mon 15:25
amrithprabhu@ubuntu: ~/Desktop/IT/Testing/sqlmap
File Edit View Search Terminal Tabs Help

centos@ip-10-0-4-215: /webapp
amrithprabhu@ubuntu: ~/Desktop/IT/Testing/sqlmap

[15:22:09] INFO testing 'PostgreSQL stacked queries (heavy query - comment)'
[15:22:11] INFO testing 'PostgreSQL > 8.1.2 stacked queries (GILS - comment)'
[15:22:12] INFO testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[15:22:13] INFO testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[15:22:15] INFO testing 'Oracle stacked queries (heavy query - comment)'
[15:22:16] INFO testing 'IBM DB2 stacked queries (heavy query - comment)'
[15:22:17] INFO testing 'SQLite > 2.0 stacked queries (heavy query - comment)'
[15:22:18] INFO testing 'MySQL > 5.0.12 AND time-based blind'
[15:22:19] INFO testing 'MySQL > 5.0.12 OR time-based blind'
[15:22:22] INFO testing 'MySQL > 5.0.12 AND time-based blind (comment)'
[15:22:23] INFO testing 'MySQL > 5.0.12 OR time-based blind (comment)'
[15:22:24] INFO testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[15:22:27] INFO testing 'MySQL > 5.0.12 OR time-based blind (query SLEEP)'
[15:22:29] INFO testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP - comment)'
[15:22:30] INFO testing 'MySQL > 5.0.12 OR time-based blind (query SLEEP - comment)'
[15:22:31] INFO testing 'MySQL > 5.0.11 AND time-based blind (heavy query)'
[15:22:36] INFO testing 'MySQL > 5.0.11 OR time-based blind (heavy query)'
[15:22:38] INFO testing 'MySQL > 5.0.12 LIKE time-based blind'
[15:22:40] INFO testing 'MySQL > 5.0.12 RLIKE time-based blind (query SLEEP)'
[15:22:43] INFO testing 'MySQL AND time-based blind (LFI)'
[15:22:45] INFO testing 'MySQL OR time-based blind (LFI)'
[15:22:47] INFO testing 'PostgreSQL > 8.1 AND time-based blind'
[15:22:49] INFO testing 'PostgreSQL > 8.1 OR time-based blind'
[15:22:52] INFO testing 'PostgreSQL AND time-based blind (heavy query)'
[15:22:54] INFO testing 'PostgreSQL OR time-based blind (heavy query)'
[15:22:56] INFO testing 'Microsoft SQL Server/Sybase time-based blind (Er)'
[15:22:58] INFO testing 'Microsoft SQL Server/Sybase AND time-based blind (heavy query)'
[15:23:00] INFO testing 'Microsoft SQL Server/Sybase OR time-based blind (heavy query)'
[15:23:02] INFO testing 'Oracle AND time-based blind'
[15:23:04] INFO testing 'Oracle OR time-based blind'
[15:23:06] INFO testing 'Oracle AND time-based blind (heavy query)'
[15:23:08] INFO testing 'Oracle OR time-based blind (heavy query)'
[15:23:10] INFO testing 'IBM DB2 AND time-based blind (heavy query)'
[15:23:12] INFO testing 'IBM DB2 OR time-based blind (heavy query)'
[15:23:13] INFO testing 'SQLite > 2.0 AND time-based blind (heavy query)'
[15:23:16] INFO testing 'SQLite > 2.0 OR time-based blind (heavy query)'
[15:23:18] INFO testing 'Informix AND time-based blind (heavy query)'
[15:23:20] INFO testing 'Informix OR time-based blind (heavy query)'
[15:23:22] INFO testing 'MySQL > 5.1 time-based blind (heavy query) - PROCEDURE ANALYSE (EXTRACTVALUE)'
[15:23:24] INFO testing 'MySQL > 5.0.12 time-based blind - Parameter replace'
[15:23:26] INFO testing 'MySQL > 5.0.12 time-based blind - Parameter replace (substitution)'
[15:23:28] INFO testing 'PostgreSQL > 8.1 time-based blind - Parameter replace'
[15:23:29] INFO testing 'Oracle time-based blind - Parameter replace (DBMS_LOCK.SLEEP)'
[15:23:32] INFO testing 'Oracle time-based blind - Parameter replace (DBMS_PIPE.RECEIVE_MESSAGE)'
[15:23:34] INFO testing 'MySQL > 5.0.12 time-based blind - ORDER BY, GROUP BY clause'
[15:23:36] INFO testing 'PostgreSQL > 8.1 time-based blind - ORDER BY, GROUP BY clause'
[15:23:38] INFO testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (DBMS_LOCK.SLEEP)'
[15:23:40] INFO testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (DBMS_PIPE.RECEIVE_MESSAGE)'
[15:23:42] INFO testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:23:44] INFO testing 'Generic UNION query (random number) - 1 to 10 columns'
[15:23:46] INFO testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[15:23:48] INFO testing 'MySQL UNION query (random number) - 1 to 10 columns'
[15:25:30] WARNING WARNING parameter 'User-Agent' does not seem to be injectable
[15:25:35] CRITICAL all tested parameters do not appear to be injectable. Try to increase values for '--level/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=spacecomment') and/or switch '--random-agent'

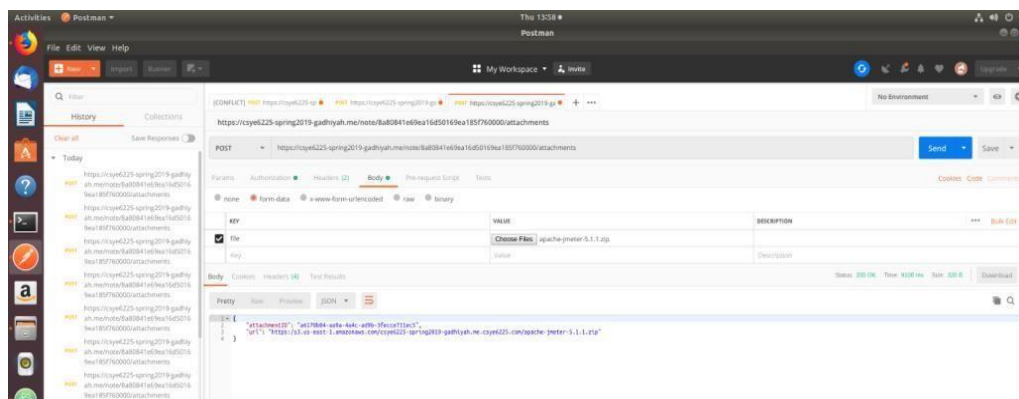
[*] ending @ 15:25:39 /2019-04-08/
amrithprabhu@ubuntu: ~/Desktop/IT/Testing/sqlmap
```

### File size restriction:

Firewall rule is added to block the file attachment to note if file size greater than 1MB.  
Without firewall: File with size 56 MB successfully attached to note and stored in S3  
as there is no such restriction.

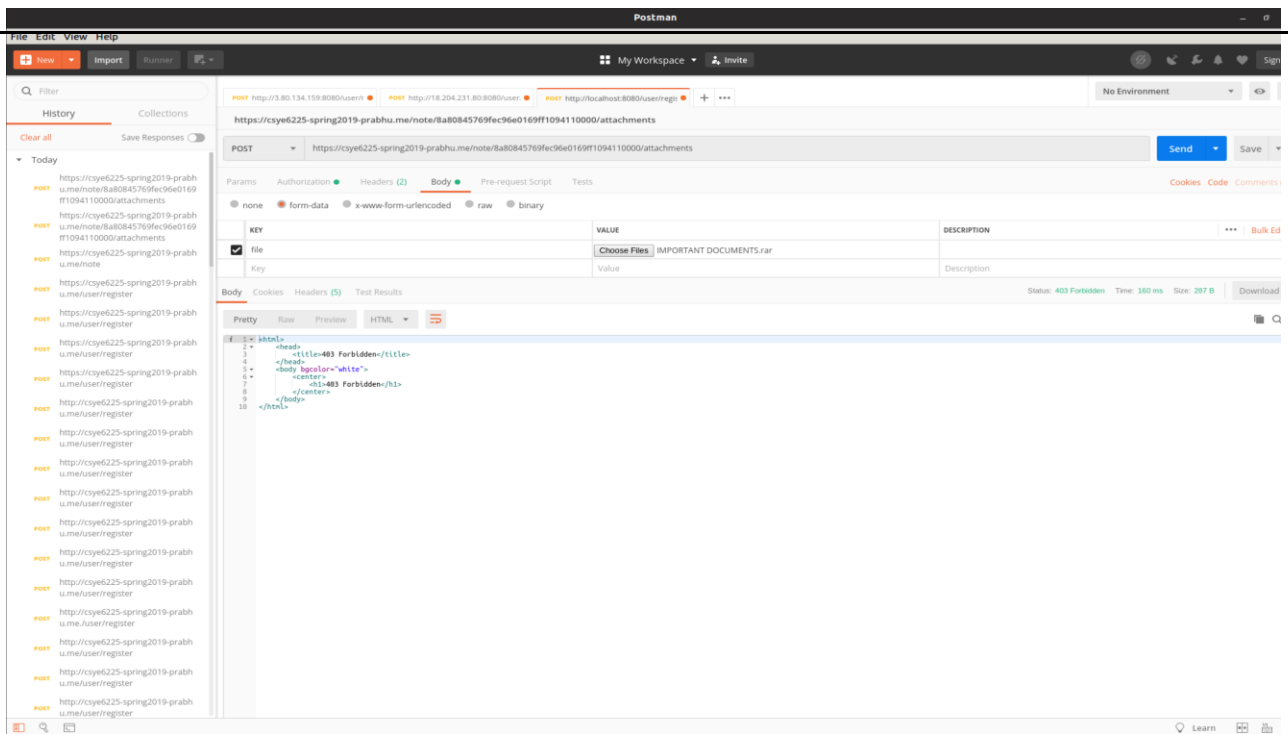
#### a. WAF disabled (To restrict file size):

Successfully able to attach the large sized file and stored into S3



#### b. WAF Enabled:

With WAF enabled there is restriction on the size of the file to be attached. Below screenshot shows failure to attach the file due to its size 403 forbidden error is thrown.  
When file with smaller size is used the attachment is successful and file stores to S3 refer the below screen shots.

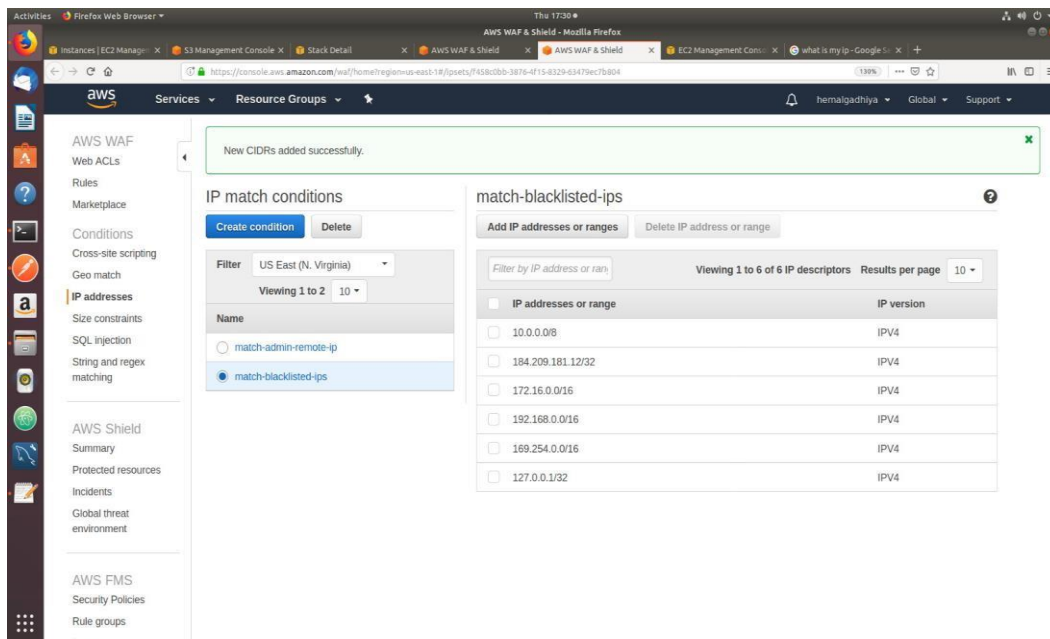


Firewall rule to block blacklist IP.

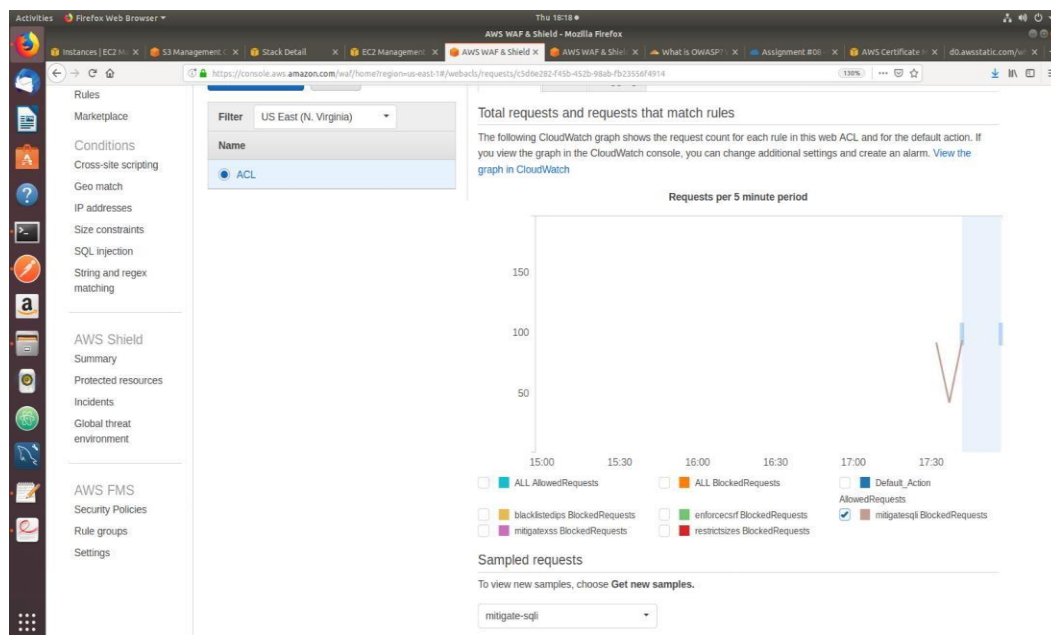
WAF rule can be set to block specific IP address (Blacklisting IP address). And when the web request (API call) is made from that IP address the HTTP request is dropped by the firewall.

In our example:

We determine the public IP address of our machine add that IP address to list of blacklisted IP addresses as below:



When API call is made from blacklisted IP's these calls will be blocked by WAF which can be seen from AWS console refer below screenshot:



Activities Firefox Web Browser This 18:19 AWS WAF & Shield - Mozilla Firefox

Instances | EC2 Management | Stack Detail | EC2 Management | AWS WAF & Shield | AWS WAF & Shield | What is OWASP | Assignment #01 | AWS Certificate | do.awsstatic.com/

https://console.aws.amazon.com/waf/home?region=us-east-1#/webacl/requests/C5d6e282-4455-452b-94ab-fb2335ef4914 138%

Source IP	URI	Matches rule	Action	Time (UTC)
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:33
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:33
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:33
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:34
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:34
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:35
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:36
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:36
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:37
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:39
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:39
▶ 155.33.133.46	/user/register/	mitigate-sqli	Block	00:42:39