

CSYE 6225 - CLOUD COMPUTING & NETWORK STRUCTURES

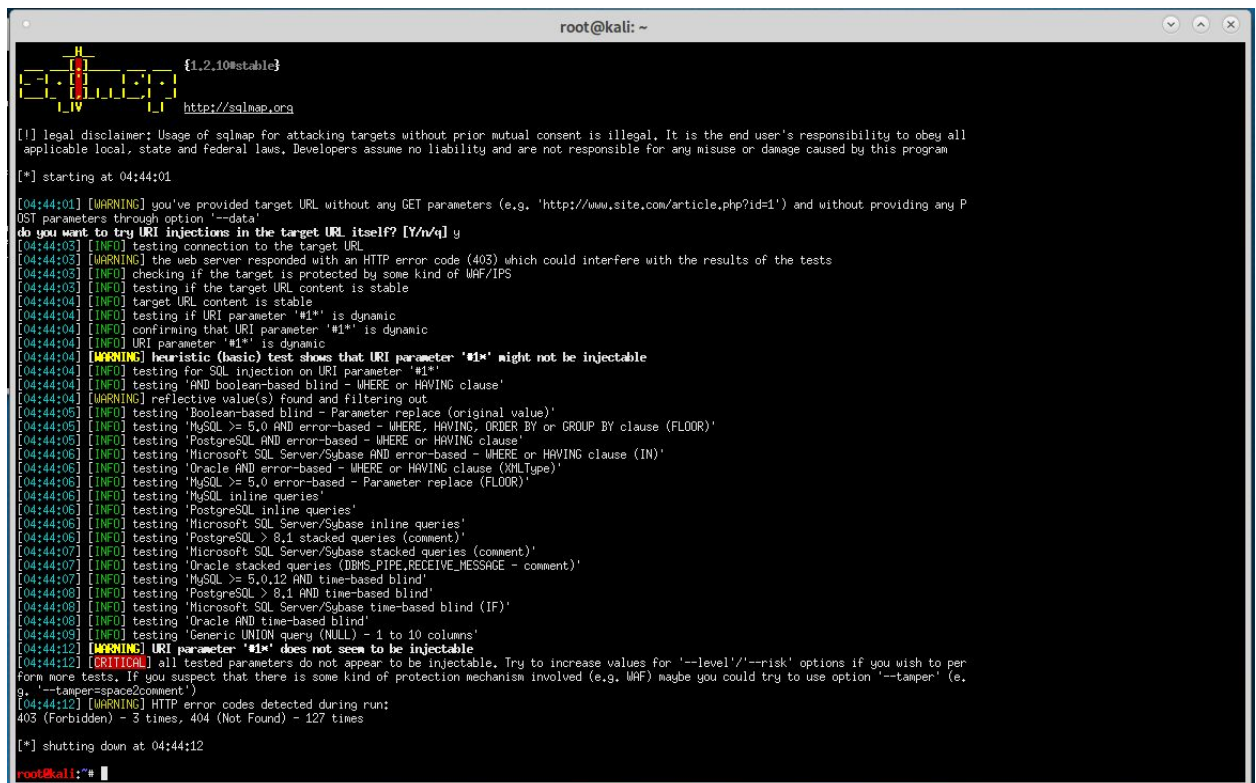
ASSIGNMENT 10- WEB APPLICATION FIREWALL

This assignment focuses on using AWS WAF to mitigate Web application vulnerabilities.

The 3 attack vectors of our UI are as follows:

1. 1st Top vulnerability - Mitigate SQL Injection Attacks:

Matches attempted SQLi patterns in the URI, QUERY_STRING, BODY, COOKIES



```
root@kali: ~
[!] 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 04:44:01

[04:44:01] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?id=1') and without providing any P
OST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] y
[04:44:03] [INFO] testing connection to the target URL
[04:44:03] [WARNING] the web server responded with an HTTP error code (403) which could interfere with the results of the tests
[04:44:03] [INFO] checking if the target is protected by some kind of WAF/IPS
[04:44:03] [INFO] testing if the target URL content is stable
[04:44:04] [INFO] target URL content is stable
[04:44:04] [INFO] testing if URI parameter '#1*' is dynamic
[04:44:04] [INFO] confirming that URI parameter '#1*' is dynamic
[04:44:04] [INFO] URI parameter '#1*' is dynamic
[04:44:04] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[04:44:04] [INFO] testing for SQL injection on URI parameter '#1*'
[04:44:04] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[04:44:04] [WARNING] reflective value(s) found and filtering out
[04:44:05] [INFO] testing 'boolean-based blind - Parameter replace (original value)'
[04:44:05] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[04:44:05] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[04:44:05] [INFO] testing 'Microsoft SQL Server/Subbase AND error-based - WHERE or HAVING clause (IN)'
[04:44:05] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[04:44:05] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[04:44:05] [INFO] testing 'MySQL inline queries'
[04:44:05] [INFO] testing 'PostgreSQL inline queries'
[04:44:05] [INFO] testing 'Microsoft SQL Server/Subbase inline queries'
[04:44:05] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[04:44:05] [INFO] testing 'Microsoft SQL Server/Subbase stacked queries (comment)'
[04:44:05] [INFO] testing 'Oracle stacked queries (DBMS_PIPE, RECEIVE_MESSAGE - comment)'
[04:44:05] [INFO] testing 'MySQL >= 5.0,12 AND time-based blind'
[04:44:05] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[04:44:05] [INFO] testing 'Microsoft SQL Server/Subbase time-based blind (IF)'
[04:44:05] [INFO] testing 'Oracle AND time-based blind'
[04:44:05] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[04:44:12] [WARNING] URI parameter '#1*' does not seem to be injectable
[04:44:12] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to per
form 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=space2comment')
[04:44:12] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 3 times, 404 (Not Found) - 127 times

[*] shutting down at 04:44:12
root@kali:~#
```

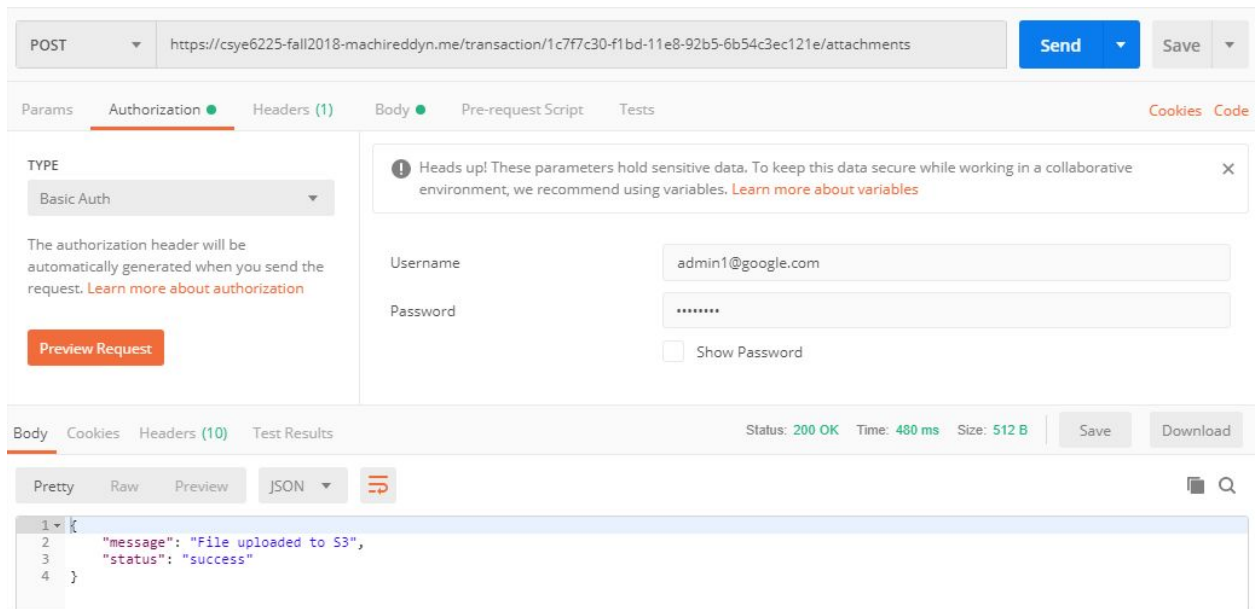
For our web application, none of the URLs are vulnerable and we have web level security because we are using sequelize node module which does not use web URL directly for sql parameter.

The waf is set up and in place to handle the vulnerability but since node module is handling the issue testing can't be done.

2. 7th Top vulnerability - Mitigate abnormal requests via size restrictions:

Enforce consistent request hygiene, limit size of key elements

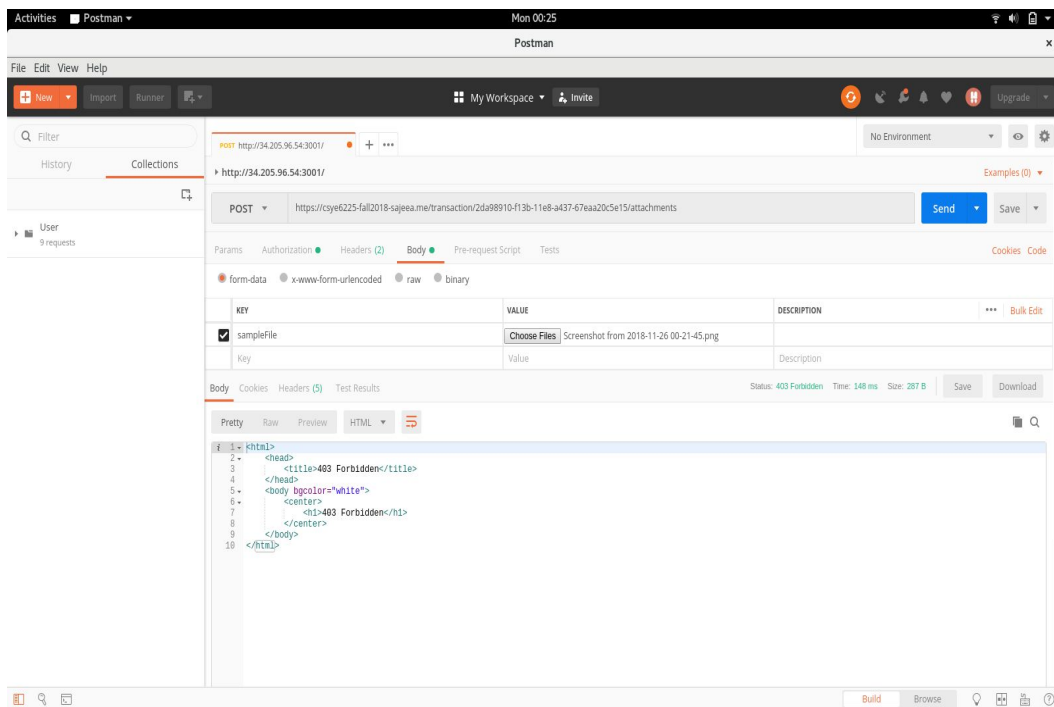
Without WAF:



With WAF:

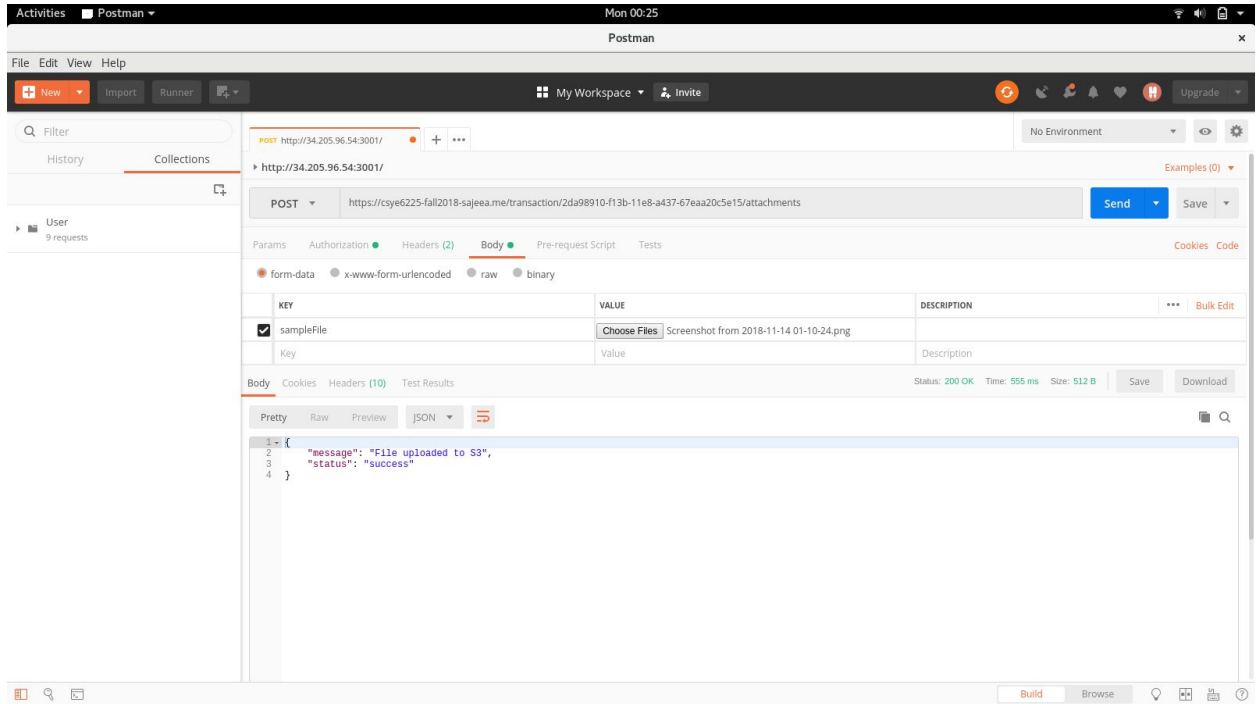
1. BODY size > 100 KB

With WAF, when the size of BODY was more greater than the max BODY size of 100 KB, it throws a 403 forbidden error. Hence, it mitigates the abnormal request via size restriction.



2. BODY size < 100 KB

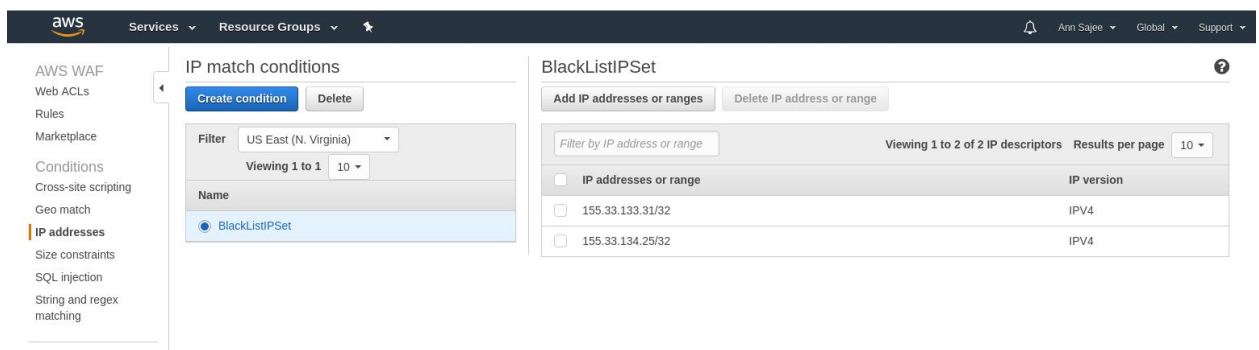
When the BODY size is less than 100KB the request was accepted.



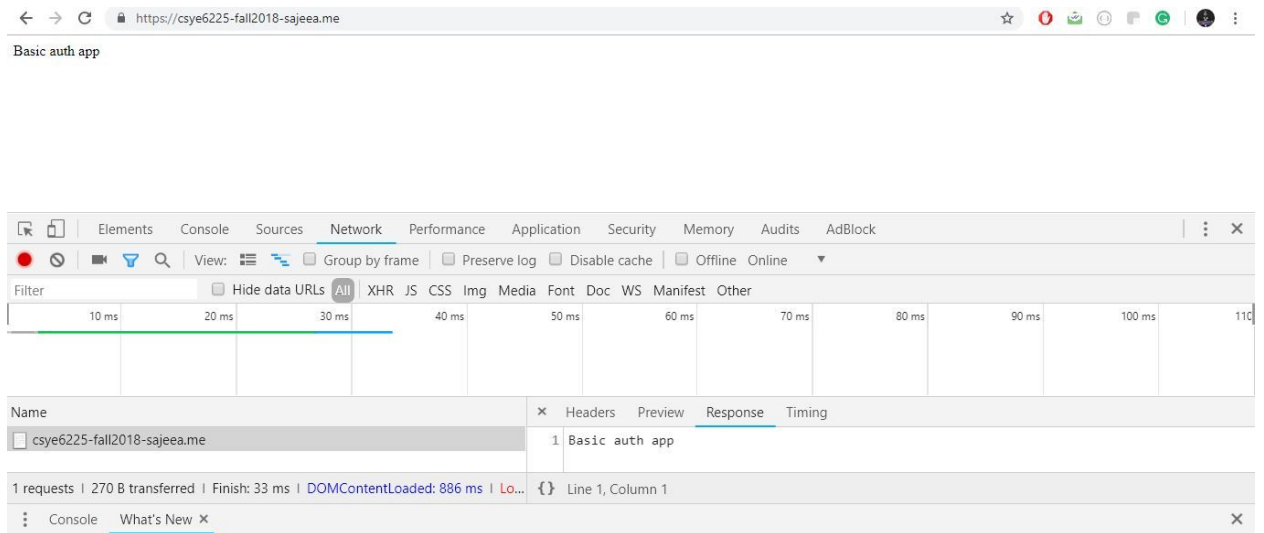
3. 10th Top Vulnerability - IP Blacklist:

Matches IP addresses that should not be allowed to access content

List of Blacklisted IP Addresses:



Without WAF:



With WAF:

403 Forbidden

