

ATTACK VECTORS

Attack Vectors:

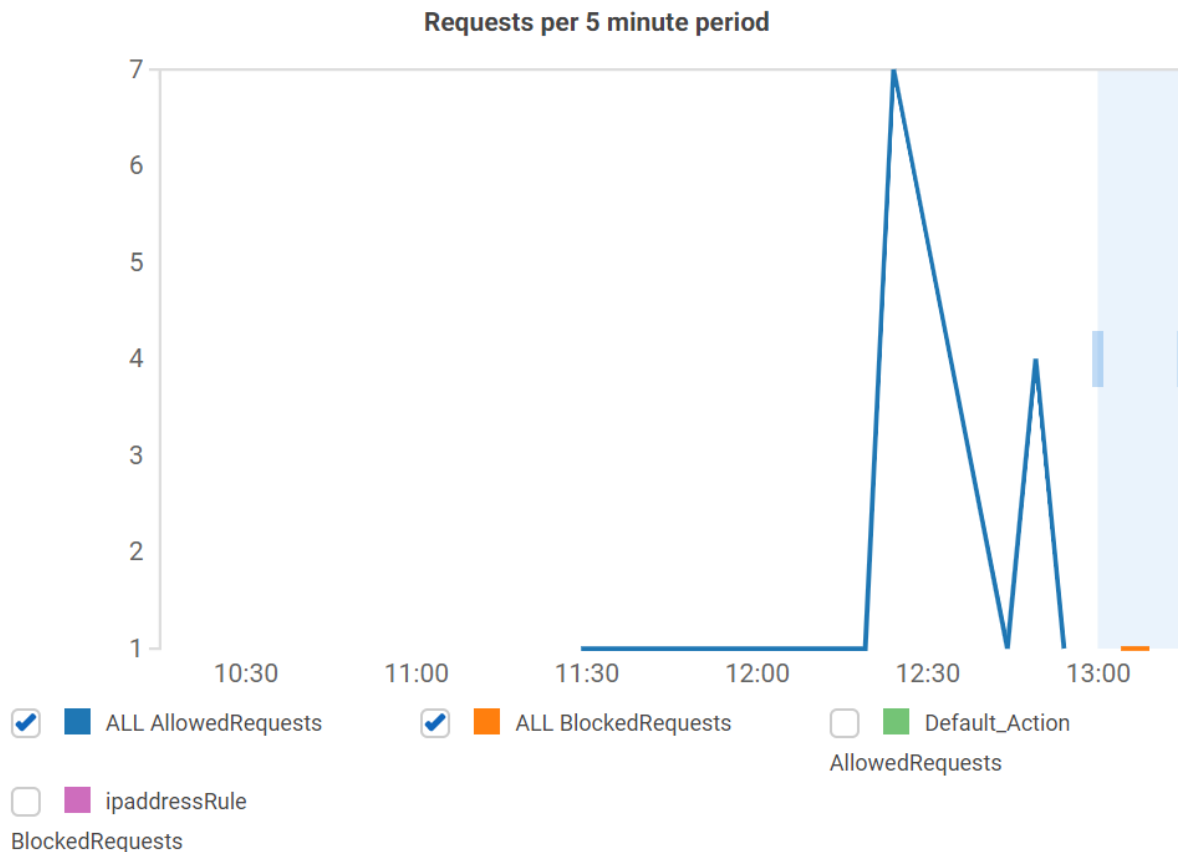
- IP address blocking
- SQL injection attack
- File size constraint
- Open port analysis

1. IP address blocking:

You can use this history to help identify and block IP addresses from malicious sources. This solution creates an AWS Lambda function that automatically parses access logs, counts the number of bad requests from unique source IP addresses, and updates AWS WAF to block further scans from those addresses.

Total requests and requests that match rules

The following CloudWatch graph shows the request count for each rule in this web ACL and for the default action. If you view the graph in the CloudWatch console, you can change additional settings and create an alarm. [View the graph in CloudWatch](#)



ATTACK VECTORS

- With WAF disabled:
All the requests from IP address 155.33.132.64 are allowed.

Source IP	URI	Matches rule	Action	Time (UTC)
▶ 71.6.143.90	/	DefaultAction	All ow	16:47:20
▶ 155.33.132.64	/librarymanageme... system-0.0.1- SNAPSHOT/user/re gister	DefaultAction	All ow	16:52:08
▶ 155.33.132.64	/librarymanageme... system-0.0.1- SNAPSHOT/user/re gister	DefaultAction	All ow	16:52:11
▶ 155.33.132.64	/librarymanageme... system-0.0.1- SNAPSHOT/user/re gister	DefaultAction	All ow	16:52:12
▶ 155.33.132.64	/librarymanageme... system-0.0.1- SNAPSHOT/user/re gister	DefaultAction	All ow	16:52:14
▶ 155.33.132.64	/librarymanageme... system-0.0.1-	DefaultAction	All ow	16:55:33

- With WAF Enabled:

When 155.33.132.63 IP address is added to the blacklisted IP addresses range. All the requests from this IP address are blocked by WAF.

IP match conditions

Create condition Delete

Filter US East (N. Virginia) Viewing 1 to 1 10

Name

☒ IPSet for blacklisted IP addresses

IPSet for blacklisted IP addresses

Add IP addresses or ranges Delete IP address or range

Filter by IP address or Viewing 1 to 3 of 3 IP descriptors Results per page 10

<input type="checkbox"/> IP addresses or range	IP version
<input type="checkbox"/> 169.254.198.15/32	IPV4
<input type="checkbox"/> 155.33.132.64/32	IPV4
<input type="checkbox"/> 169.254.0.0/16	IPV4

Request :Request made to /User/register endpoint with blocked IP address 155.33.132.64 gets 403 forbidden response from WAF application firewall.

ATTACK VECTORS

POST

https://csye6225-su19-honraoa.me/librarymanagementsystem-0.0.1-SNAPSHOT/user/register

Send

Save

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Cookies

Code

Comments (0)

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL BETA

JSON (application/json)

Beautify

1 {

2 "email": "honraoakshata31@gmail.com",

3 "password": "Abcd@123"

4 }

Body

Cookies

Headers (5)

Test Results

Status: 403 Forbidden

Time: 169ms

Size: 287 B

Save Response

Pretty

Raw

Preview

HTML

1 <html>

2

3 <head>

4 <title>403 Forbidden</title>

5 </head>

6

Sampled requests

To view new samples, choose **Get new samples**.

ipaddressRule

Get new samples

Sample data from 2019-08-08 16:59:55 to 17:14:55				
Source IP	URI	Matches rule	Action	Time (UTC)
▶ 155.33.132.64	/librarymanageme... system-0.0.1- SNAPSHOT/book/	ipaddressRule	Block	17:04:32
▶ 155.33.132.64	/librarymanageme... system-0.0.1- SNAPSHOT/user/register	ipaddressRule	Block	17:12:52

2. SQL INJECTION:

SQL Injection (SQLi) is a type of a injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. We choose this attack vector to check if our application has any SQL

ATTACK VECTORS

injection vulnerabilities. It turns out that this application is protected as we have used Prepared Statements (Parameterized queries).

Command:

```
python sqlmap.py -u " https://csye6225-su19-  
honraoa.me/librarymanagementsystem-0.0.1-SNAPSHOT/user/register/" --  
method=POST --  
data='{ "email": "honraoa.a@gmail.com", "password": "Test@123" }' --  
tamper=space2comment
```

- WAF disabled:

When WAF is disabled firewall doesn't forbid the attack but application security throws 400 bad request response to the sql injection attack .

```
[root@shashank@kali-vm ~]# python sqlmap.py -u "https://cys0622-sul9-honracc.me/librarymanagementsystem/0.0.1-SNAPSHOT/user/register" --method-POST --data '{"email":"shashan1314@gmail.com","password":"Test@223"}'
[+] http://10.10.10.10:8080/
[+] 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 @ 13:50:16 /2019-08-06/

SQLMap data found in POST data. Do you want to process it? [Y/n/q] Y
[*] SQLMap testing connection to the target URL...
[13:50:18] [WARNING] The web server responded with an HTTP error code (409) which could interfere with the results of the tests
[13:50:20] [INFO] Testing if the Target URL content is stable
[13:50:20] [INFO] target url content is stable
[13:50:30] [INFO] testing if (custom) POST parameter 'JSON email' is dynamic
[13:50:30] [INFO] (custom) POST parameter 'JSON email' appears to be dynamic
[13:50:30] [WARNING] heuristic (basic) test shows that (custom) POST parameter 'JSON email' might not be injectable
[13:50:30] [INFO] testing for SQL injection on (custom) POST parameter 'JSON email'
[13:50:30] [INFO] AND boolean-based blind - WHERE or HAVING clause
[13:50:30] [INFO] Boolean-based blind - Parameter replace (original value)
[13:50:30] [INFO] MySQL >= 5.0 and error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR!)
[13:50:30] [INFO] PostgreSQL AND error-based - WHERE or HAVING clause
[13:50:30] [INFO] Microsoft SQL Server/Sybase and error-based - WHERE or HAVING clause (IN)
[13:50:30] [INFO] Oracle and error-based - WHERE or HAVING clause (OR)
[13:50:30] [INFO] MySQL >= 5.0 error-based - Parameter replace (FLOOR)
[13:50:30] [INFO] MySQL inline queries
[13:50:30] [INFO] PostgreSQL inline queries
[13:50:30] [INFO] Microsoft SQL Server/Sybase inline queries
[13:50:30] [INFO] PostgreSQL >= 8.1 stacked queries (comment)
[13:50:30] [INFO] Microsoft SQL Server/Sybase stacked queries (comment)
[13:50:30] [INFO] Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)
[13:50:30] [INFO] MySQL >= 5.0.12 and time-based blind (query SLEEP)
[13:50:30] [INFO] PostgreSQL >= 8.1 and time-based blind
[13:50:30] [INFO] Microsoft SQL Server/Sybase time-based blind (IF)
[13:50:30] [INFO] Oracle and Time-based Blind
[13:50:30] [INFO] There are only half INTON tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n/q]
```

```

1 It is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? (Y/N) Y
2 [INFO] [13:50:33] [WARNING] (custom) POST parameter '350n_email' does not seem to be injectable
3 [INFO] [13:50:33] [INFO] testing if (custom) POST parameter '350n_password' is dynamic
4 [INFO] [13:50:33] [INFO] (custom) POST parameter '350n_password' appears to be dynamic
5 [INFO] [13:50:33] [WARNING] heuristic (basic) test shows that (custom) POST parameter '350n_password' might not be injectable
6 [INFO] [13:50:33] [INFO] testing for SQL injection on (custom) POST parameter '350n_password'
7 [INFO] [13:50:33] [INFO] testing AND boolean-based blind - WHERE or HAVING clause
8 [INFO] [13:50:34] [INFO] testing Boolean-based blind - Parameter replace (original value)
9 [INFO] [13:50:34] [INFO] testing WfSQL >= 5.0 AND error-based - WHERE or HAVING clause (FLOOR)
10 [INFO] [13:50:33] [INFO] testing PostgreSQL AND error-based - WHERE or HAVING clause
11 [INFO] [13:50:33] [INFO] testing Microsoft SQL Server/MySQL error-based - WHERE or HAVING clause (IN)
12 [INFO] [13:50:33] [INFO] testing Oracle AND error-based - WHERE or HAVING clause (NType)
13 [INFO] [13:50:32] [INFO] testing WfSQL >= 5.0 error-based - Parameter replace (FLOOR)
14 [INFO] [13:50:33] [INFO] testing WfSQL inline queries
15 [INFO] [13:50:33] [INFO] testing PostgreSQL inline queries
16 [INFO] [13:50:33] [INFO] testing Microsoft SQL Server/MySQL inline queries
17 [INFO] [13:50:32] [INFO] testing PostgreSQL >= 8.1 stacked queries (comment)
18 [INFO] [13:50:32] [INFO] testing Microsoft SQL Server/MySQL stacked queries (comment)
19 [INFO] [13:50:32] [INFO] testing Oracle stacked queries (open pipe receive MESSAGE comment)
20 [INFO] [13:50:37] [INFO] testing WfSQL >= 5.0.12 AND time-based blind (query SLEEP)
21 [INFO] [13:50:37] [INFO] testing PostgreSQL >= 8.1 AND time-based blind
22 [INFO] [13:50:38] [INFO] testing Microsoft SQL Server/MySQL time-based blind (IF)
23 [INFO] [13:50:38] [INFO] testing Oracle AND time-based blind
24 [INFO] [13:50:33] [INFO] testing Generic Union time query (UNION) - 1 to 10 columns
25 [INFO] [13:50:33] [WARNING] (custom) POST parameter '350n_password' does not seem to be injectable
26 [INFO] [13:50:33] [WARNING] (custom) POST parameter '350n_email' does not seem to be injectable
27 Try to use option --tamper (e.g., --tamper=space2comment) and/or switch --random-agent!
28 [INFO] [13:50:33] [INFO] HTTP error codes detected during run:
29 (Bad Request) - 151 times, 499 (Conflict) - 3 times
30
31 ending @ 13:50:39 / 2019-08-08/
32
33 whatwebkbnk1@kali:~/git/owasp-wsrf-scanner$ ./flood-bot-local-owasp-dead

```

- WAF enabled:

When WAF is enabled firewall thorws 403 forbidden access to sql injection attack

ATTACK VECTORS

```
akshata@akshata-virtual-machine:~/Cloud-Repo-Local/sqlmap-devs$ python sqlmap.py -u "https://csp0225-sul9-horrasa.m0/user/register" --method=POST --data=({'email':'qertyqerty@gmail.com','password':'Test@123'}) --tamper=space2comment

(1.3.0.40dev)
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 @ 13:26:32 /2019-08-08/

13:26:32 [INFO] loading tamper module 'space2comment'
JSON data found in POST data. Do you want to process it? [Y/n/q] Y
13:26:34 [INFO] testing connection to the target URL
13:26:34 [INFO] WAF/IPS identified as 'Aps WAF (Amazon)'
13:26:34 [CRITICAL] the web server responded with an HTTP error code (403) which could interfere with the results of the tests
13:26:34 [INFO] checking if the target is protected by some kind of WAF/IPS
13:26:35 [INFO] testing if the target URL content is stable
13:26:35 [INFO] target URL content is stable
13:26:35 [INFO] testing if (custom) POST parameter 'JSON_email' is dynamic
13:26:35 [WARNING] (custom) POST parameter 'JSON_email' does not appear to be dynamic
13:26:35 [WARNING] heuristic (basic) test shows that (custom) POST parameter 'JSON_email' might not be injectable
13:26:35 [INFO] testing for SQL injection on (custom) POST parameter 'JSON_email'
13:26:35 [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
13:26:35 [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
13:26:35 [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
13:26:36 [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
13:26:36 [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
13:26:36 [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
13:26:37 [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
13:26:37 [INFO] testing 'MySQL inline queries'
13:26:37 [INFO] testing 'PostgreSQL inline queries'
13:26:37 [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
13:26:38 [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
13:26:38 [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
13:26:38 [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
13:26:38 [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
13:26:39 [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
13:26:39 [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
13:26:39 [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] Y
13:26:41 [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
13:26:41 [WARNING] (custom) POST parameter 'JSON_email' does not seem to be injectable
13:26:42 [INFO] testing if (custom) POST parameter 'JSON_password' is dynamic
13:26:42 [WARNING] (custom) POST parameter 'JSON_password' does not appear to be dynamic
13:26:42 [WARNING] heuristic (basic) test shows that (custom) POST parameter 'JSON_password' might not be injectable
13:26:42 [INFO] testing for SQL injection on (custom) POST parameter 'JSON_password'
13:26:42 [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
13:26:43 [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
13:26:43 [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
13:26:43 [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
13:26:43 [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
13:26:44 [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
13:26:44 [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
13:26:44 [INFO] testing 'MySQL inline queries'
13:26:44 [INFO] testing 'PostgreSQL inline queries'
13:26:44 [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
13:26:44 [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
13:26:44 [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
13:26:45 [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
13:26:45 [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
13:26:45 [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
13:26:45 [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
13:26:45 [INFO] testing 'Oracle AND time-based blind'
13:26:46 [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
13:26:46 [WARNING] (custom) POST parameter 'JSON_password' does not seem to be injectable
13:26:47 [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
13:26:47 [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 155 times

[*] ending @ 13:26:57 /2019-08-08/

akshata@akshata-virtual-machine:~/Cloud-Repo-Local/sqlmap-devs$ ls -l
akshata@akshata-virtual-machine:~/Cloud-Repo-Local/sqlmap-devs$
```

3. File size constraint:

Any http request having body size greater than 200KB will not be allowed to process

- With WAF disabled:

When WAF is disabled there are no constraints on the body size so all the requests are processed .

ATTACK VECTORS

[illegible]

- With WAF enabled:

With body length exceeds the file constraint 200KB firewall will block that request and will return a 403 forbidden response.

Rules

[Create rule](#)[Delete](#)

FilterUS East (N. Virginia)

Viewing 1 to 210

Name	Type
<input checked="" type="radio"/> MyfileSizeRule	Regular
<input type="radio"/> ipAddressRule	Regular

MyfileSizeRule

[Edit rule](#)

When a request matches at least one of the filters in the size constraint condition [MyFileSize](#)

Filters in MyFileSize

The length of the Body is greater than or equal to 200000.

ATTACK VECTORS

POST

https://csye6225-su19-honraoa.me/librarymanagementsystem-0.0.1-SNAPSHOT/book/3e5629c2-9dba-4bc2...

Send

Save

Cache-Control	no-cache
Postman-Token	245c4abe-9850-4994-b2fd-ef1d19fc6730
Host	csye6225-su19-honraoa.me
Accept-Encoding	gzip, deflate
Content-Type	multipart/form-data; boundary=-----443016199629643302796256
Content-Length	215027
Connection	keep-alive

Body

Cookies

Headers (5)

Test Results

Status: 403 Forbidden Time: 123ms Size: 287 B Save Response

Pretty

Raw

Preview

HTML

```
1 <html>
2
3 <head>
4   <title>403 Forbidden</title>
5 </head>
6
7 <body bgcolor="white">
8   <center>
```

4. Checking all the open ports using nmap:

Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses. After running nmap command on domain we can see that only https 443 port open except that all the open port have been closed by the firewall.

```
akshata@akshata-virtual-machine:~/Downloads/apache-jmeter-5.1.1/bin$ nmap -v csye6225-su19-honraoa.me
Starting Nmap 7.60 ( https://nmap.org ) at 2019-08-09 17:10 EDT
Initiating Ping Scan at 17:10
Scanning csye6225-su19-honraoa.me (52.72.170.231) [2 ports]
Completed Ping Scan at 17:10, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:10
Completed Parallel DNS resolution of 1 host. at 17:10, 0.02s elapsed
Initiating Connect Scan at 17:10
Scanning csye6225-su19-honraoa.me (52.72.170.231) [1000 ports]
Discovered open port 443/tcp on 52.72.170.231
Increasing send delay for 52.72.170.231 from 0 to 5 due to 11 out of 14 dropped probes since last increase.
Completed Connect Scan at 17:10, 29.43s elapsed (1000 total ports)
Nmap scan report for csye6225-su19-honraoa.me (52.72.170.231)
Host is up (0.017s latency).
Other addresses for csye6225-su19-honraoa.me (not scanned): 54.210.226.206
rDNS record for 52.72.170.231: ec2-52-72-170-231.compute-1.amazonaws.com
Not shown: 999 filtered ports
PORT      STATE SERVICE
443/tcp   open  https
Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 29.53 seconds
```