

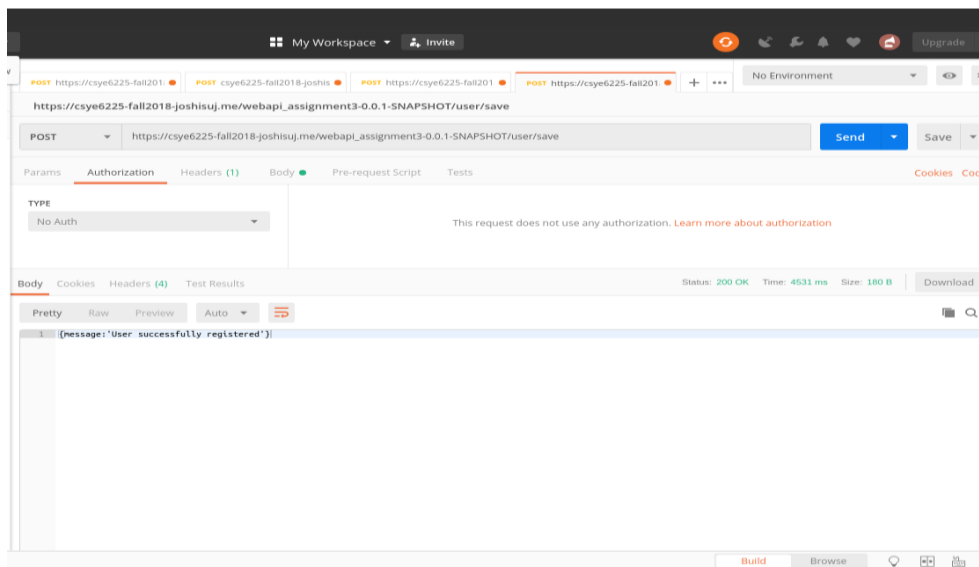
# NETWORKS STRUCTURES AND CLOUD COMPUTING

## ASSIGNMENT10

The following are the examples of attack vector and how we mitigated it

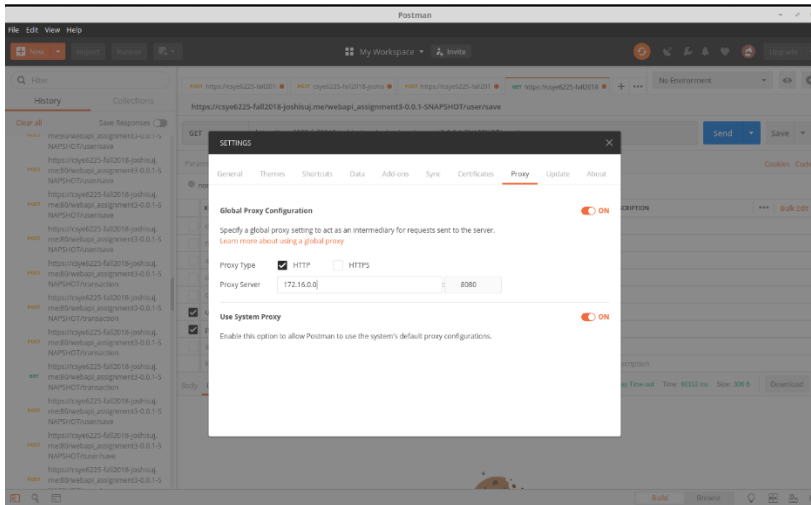
1. IP address Blocking: - IP Address blocking is a security measure that prevents a connection between a specific or group of IP addresses and a mail, web or Internet server. This is usually done to ban or block any undesirable sites and hosts from entering the server or node and causing harm to the network or individual computers. IP blocking is usually used by companies to prevent intrusion, allow remote access as well as limit the kinds of websites that can be accessed by employees in order to keep productivity high. Schools and other academic institutions also use IP address blocking for protection against unauthorized access of confidential records and data and for enforcing censorship.

Before mitigation we could access the web application through any IP address



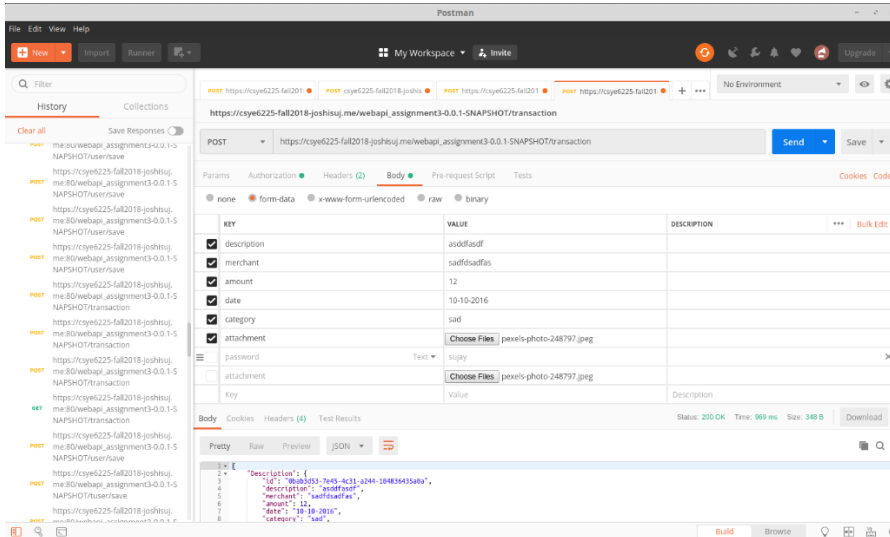
We mitigate this problem by blacklisting certain IP's

After Mitigation

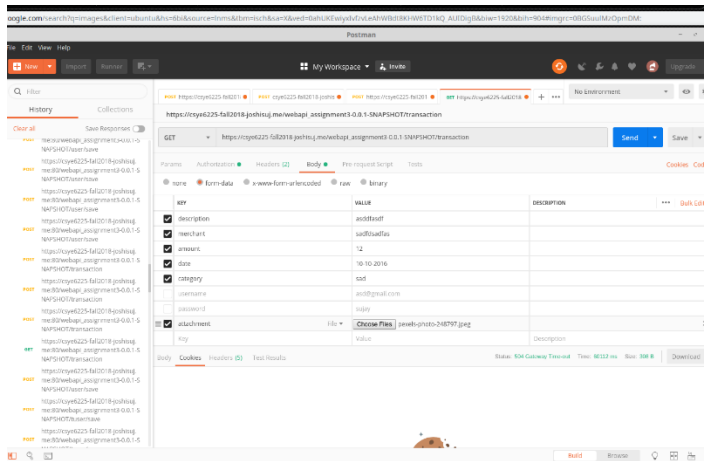


2. The body of the post request greater than certain bytes are not forwarded  
When attacker attack the web site they often try to attach large files or type text which is very large which often generates load on the web site, tending to cause denial of service. Therefore, to mitigate this issue we limit the request in the body to be less than 4096 bytes.

## Before mitigation



## After mitigation



3. **SQL Injection:** - SQL injection is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

We have tested the SQL injection technique using Automatic SQL injection and database takeover tool.

Before Mitigation

```
File Edit View Search Terminal Help
[14:50:11] [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=payloadcomment').
[*] ending @ 14:55:32 /2018-11-25/

[centos@ip-10-0-0-203 ~]$ python sqlmap.py -u "https://csye6225-fall2018-joshijw.me:80/webapi_assignment3-0.0.1-SNAPSHOT/transaction?id=b0b0d57-44be-4102-9948-8ca096594030" --batch

(1.2.11.15dev)
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 @ 15:01:09 /2018-11-25/

12:01:09 [INFO] testing connection to the target URL
12:01:09 [INFO] checking if the target is protected by some kind of WAF/IPS
12:01:12 [INFO] testing if the target URL content is stable
12:01:12 [INFO] target URL content is stable
12:01:12 [INFO] testing if GET parameter 'id' is dynamic
12:01:12 [WARNING] GET parameter 'id' does not appear to be dynamic
12:01:12 [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
12:01:12 [INFO] testing for SQL injection on GET parameter 'id'
12:01:12 [INFO] testing 'AND boolean-based blind' - WHERE or HAVING clause
12:01:12 [INFO] testing 'Boolean-based blind' - Parameter replace (original value)
12:01:12 [INFO] testing 'MySQL > 5.6 AND error-based' - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
12:01:12 [INFO] testing 'PostgreSQL AND error-based' - WHERE or HAVING clause
12:01:12 [INFO] testing 'Microsoft SQL Server/Oracle AND error-based' - WHERE or HAVING clause (IN)'
12:01:12 [INFO] testing 'Oracle AND error-based' - WHERE or HAVING clause (XMLType)'
12:01:12 [INFO] testing 'MySQL > 5.6 error-based' - Parameter replace (FLOOR)'
12:01:12 [INFO] testing 'MySQL inline queries'
12:01:12 [INFO] testing 'Microsoft SQL Server/Oracle inline queries'
12:01:12 [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
12:01:12 [INFO] testing 'Microsoft SQL Server/Oracle stacked queries (comment)'
12:01:12 [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
12:01:12 [INFO] testing 'MySQL > 5.6.12 AND time-based blind'
12:01:12 [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
there seems to be a continuous problem with connection to the target. Are you sure that you want to continue with further target testing? [Y/n] N
it looks like the back-end DBMS is 'PostgreSQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'PostgreSQL' extending provided level (1) and risk (1) values? [Y/n] Y
12:01:12 [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
12:01:12 [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
12:01:12 [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
12:01:12 [INFO] user aborted during detection phase
how do you want to proceed? (1) skip current test (2) end detection phase (3) next parameter (4) change verbosity (5) quit [1]
12:01:12 [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 1 times, 502 (Bad Gateway) - 2 times
12:01:12 [INFO] user quit
[*] ending @ 15:12:07 /2018-11-25/

[centos@ip-10-0-0-203 ~]$ python sqlmap.py -u "https://csye6225-fall2018-joshijw.me:80/webapi_assignment3-0.0.1-SNAPSHOT/transaction?id=b0b0d57-44be-4102-9948-8ca096594030" --batch
```

After mitigating we got the Output be not injectable

```
File Edit View Search Terminal Help
[!] 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 @ 14:54:31 /2018-11-25/

12:01:09 [INFO] testing connection to the target URL
12:01:09 [INFO] checking if the target is protected by some kind of WAF/IPS
12:01:12 [INFO] testing if the target URL content is stable
12:01:12 [INFO] target URL content is stable
12:01:12 [INFO] testing if GET parameter 'id' is dynamic
12:01:12 [WARNING] GET parameter 'id' does not appear to be dynamic
12:01:12 [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
12:01:12 [INFO] testing for SQL injection on GET parameter 'id'
12:01:12 [INFO] testing 'AND boolean-based blind' - WHERE or HAVING clause
12:01:12 [INFO] testing 'Boolean-based blind' - Parameter replace (original value)
12:01:12 [INFO] testing 'MySQL > 5.6 AND error-based' - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
12:01:12 [INFO] testing 'PostgreSQL AND error-based' - WHERE or HAVING clause
12:01:12 [INFO] testing 'Microsoft SQL Server/Oracle AND error-based' - WHERE or HAVING clause (IN)'
12:01:12 [INFO] testing 'Oracle AND error-based' - WHERE or HAVING clause (XMLType)'
12:01:12 [INFO] testing 'MySQL > 5.6 error-based' - Parameter replace (FLOOR)'
12:01:12 [INFO] testing 'MySQL inline queries'
12:01:12 [INFO] testing 'PostgreSQL inline queries'
12:01:12 [INFO] testing 'Microsoft SQL Server/Oracle inline queries'
12:01:12 [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
12:01:12 [INFO] testing 'Microsoft SQL Server/Oracle stacked queries (comment)'
12:01:12 [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
12:01:12 [INFO] testing 'MySQL > 5.6.12 AND time-based blind'
12:01:12 [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
12:01:12 [INFO] testing 'Microsoft SQL Server/Oracle time-based blind (IF)'
12:01:12 [INFO] testing 'Oracle AND time-based blind'
12:01:12 [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
12:01:12 [WARNING] GET parameter 'id' does not seem to be injectable
12:01:12 [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=payloadcomment').
[*] ending @ 14:55:32 /2018-11-25/

[centos@ip-10-0-0-203 ~]$ python sqlmap.py -u "https://csye6225-fall2018-joshijw.me:80/webapi_assignment3-0.0.1-SNAPSHOT/transaction?id=b0b0d57-44be-4102-9948-8ca096594030" --batch
```

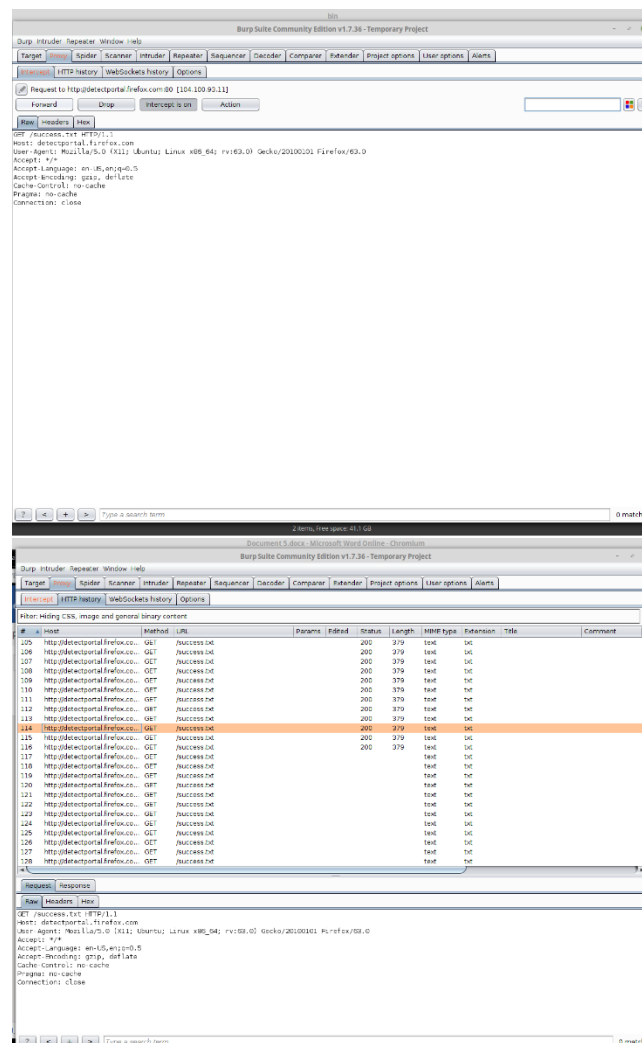
The screenshot shows the Postman application interface. A GET request is configured with the URL `https://csye6225-fall2018-joshijw.me/webapi_assignment3-0.0.1-SNAPSHOT/transaction`. The 'Authorization' tab is selected, showing 'Basic Auth' with fields for 'Username' (set to 'OR '1=1'' and 'Password' (set to '\*\*\*\*\*'). The 'Body' tab shows a JSON response: `{ "Description": "banther123", "Code": 401 }`. The status bar at the bottom indicates 'Status: 200 OK, Time: 617 ms, Size: 195 B'.

We have also implemented other security checks as follows

1. Cross-site request forgery, also known as one-click attack or session riding and abbreviated as CSRF (sometimes pronounced sea-surf<sup>1</sup>) or XSRF, is a type of malicious exploit of a website where unauthorized commands are transmitted from a user that the web application trusts. There are many ways in which a malicious website can transmit such commands; specially-crafted image tags, hidden forms, and JavaScript XMLHttpRequests, for example, can all work without the user's interaction or even knowledge.

We have tried to intercept the data between the client and server and tried to change the data between them and application provided security by not allowing use to forward random data packets

Software Used: Burp Suite



2. We have also tried to ping to application and check which all ports are open  
The application has on the https port open required for communication which is a good security in measure in our application  
Software Used: Nmap

```
sujoy@sujoy:~$ sudo nmap -vS "cye6225-fal12018-joshisuj.me"
Starting Nmap 7.60 ( https://nmap.org ) at 2018-11-25 22:44 EST
Nmap scan report for cye6225-fal12018-joshisuj.me (34.192.176.57)
Host is up (0.016s latency).
Other addresses for cye6225-fal12018-joshisuj.me (not scanned): 23.22.90.213 54.210.88.155
DNS record for 34.192.176.57: ec2-34-192-176-57.compute-1.amazonaws.com
Not shown: 995 filtered ports
PORT      STATE SERVICE
22/tcp    closed ssh
80/tcp    closed http
443/tcp   open  https
1300/tcp  closed mysql
8080/tcp  closed http-proxy

Nmap done: 1 IP address (1 host up) scanned in 28.83 seconds
sujoy@sujoy:~$ sudo nmap -vS "cye6225-fal12018-joshisuj.me"
Starting Nmap 7.60 ( https://nmap.org ) at 2018-11-25 22:46 EST
Warning: Hostname cye6225-fal12018-joshisuj.me resolves to 3 IPs, using 34.192.176.57.
Initiating SYN Stealth Scan at 22:46
Scanning cye6225-fal12018-joshisuj.me (34.192.176.57) [4 ports]
Completed SYN Scan at 22:46, 0.22s elapsed (1 total hosts)
Initiating parallel DNS resolution of 1 host, at 22:46
Completed Parallel DNS resolution of 1 host, at 22:46, 0.06s elapsed
Initiating SYN Stealth Scan at 22:46
Scanning cye6225-fal12018-joshisuj.me (34.192.176.57) [1000 ports]
Discovered open port 443/tcp on 34.192.176.57
Increasing send delay for 34.192.176.57 from 0 to 5 due to 11 out of 22 dropped probes since last increase.
Completed SYN Stealth Scan at 22:46, 27.80s elapsed (1000 total ports)
Nmap scan report for cye6225-fal12018-joshisuj.me (34.192.176.57)
Host is up, received reset (tl: 233 (0.016s latency)).
Other addresses for cye6225-fal12018-joshisuj.me (not scanned): 54.210.88.155 23.22.90.213
DNS record for 34.192.176.57: ec2-34-192-176-57.compute-1.amazonaws.com
Scanned at 2018-11-25 22:46:09 EST for 26s.
Not shown: 995 filtered ports
Reason: 995 no responses
PORT      STATE SERVICE REASON
22/tcp    closed ssh      reset ttl 233
80/tcp    closed http     reset ttl 233
443/tcp   open  https      syn-ack ttl 233
1300/tcp  closed mysql     reset ttl 233
8080/tcp  closed http-proxy reset ttl 233

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 28.79 seconds
Raw packets sent: 3017 (132.64Kbits) | Rcvd: 23 (9200)
sujoy@sujoy:~$
```

```
sujoy@sujoy:~$ sudo nmap -vS "cye6225-fal12018-joshisuj.me"
Completed Parallel DNS resolution of 1 host, at 22:46, 0.06s elapsed
Initiating SYN Stealth Scan at 22:46
Scanning cye6225-fal12018-joshisuj.me (34.192.176.57) [1000 ports]
Discovered open port 443/tcp on 34.192.176.57
Increasing send delay for 34.192.176.57 from 0 to 5 due to 11 out of 22 dropped probes since last increase.
Completed SYN Stealth Scan at 22:46, 27.80s elapsed (1000 total ports)
Nmap scan report for cye6225-fal12018-joshisuj.me (34.192.176.57)
Host is up, received reset (tl: 233 (0.016s latency)).
Other addresses for cye6225-fal12018-joshisuj.me (not scanned): 54.210.88.155 23.22.90.213
DNS record for 34.192.176.57: ec2-34-192-176-57.compute-1.amazonaws.com
Scanned at 2018-11-25 22:46:09 EST for 26s.
Not shown: 995 filtered ports
Reason: 995 no responses
PORT      STATE SERVICE REASON
22/tcp    closed ssh      reset ttl 233
80/tcp    closed http     reset ttl 233
443/tcp   open  https      syn-ack ttl 233
1300/tcp  closed mysql     reset ttl 233
8080/tcp  closed http-proxy reset ttl 233

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 28.25 seconds
Raw packets sent: 3017 (132.64Kbits) | Rcvd: 23 (9200)
sujoy@sujoy:~$ sudo nmap -vS "cye6225-fal12018-joshisuj.me"
Starting Nmap 7.60 ( https://nmap.org ) at 2018-11-25 22:48 EST
Nmap scan report for cye6225-fal12018-joshisuj.me (34.192.176.57)
Host is up (0.016s latency).
Other addresses for cye6225-fal12018-joshisuj.me (not scanned): 34.192.176.57 54.210.88.155
DNS record for 34.192.176.57: ec2-34-192-176-57.compute-1.amazonaws.com
Not shown: 995 filtered ports
PORT      STATE SERVICE
22/tcp    closed ssh
80/tcp    closed http
443/tcp   open  https
1300/tcp  closed mysql
8080/tcp  closed http-proxy

Nmap done: 1 IP address (1 host up) scanned in 21.96 seconds
sujoy@sujoy:~$ sudo nmap -vA "cye6225-fal12018-joshisuj.me"
Starting Nmap 7.60 ( https://nmap.org ) at 2018-11-25 22:49 EST
Nmap scan report for cye6225-fal12018-joshisuj.me (54.210.88.155)
Host is up (0.016s latency).
Other addresses for cye6225-fal12018-joshisuj.me (not scanned): 34.192.176.57 23.22.90.213
DNS record for 54.210.88.155: ec2-54-210-88-155.compute-1.amazonaws.com
Not shown: 995 filtered ports
PORT      STATE SERVICE
22/tcp    unfiltered ssh
80/tcp    unfiltered http
443/tcp   unfiltered https
1300/tcp  unfiltered mysql
8080/tcp  unfiltered http-proxy

Nmap done: 1 IP address (1 host up) scanned in 27.85 seconds
sujoy@sujoy:~$
```

Akash Bangera  
Rishabh Jain  
Punith Narayanswamy  
Sujay Joshi