CSYE 6225 : Network Structures and Cloud Computing

Description: This assignment aims at finding the security issues in the web application we built as part of previous assignments and try to come up with solutions for them.

Procedure: Try to come up with at least 3 attack vectors. Show the vulnerability in the application because of each attack vector. Install AWS Web Application Firewall (WAF) on the Load Balancer with correct AWS Rules to stop these type of attacks from happening.

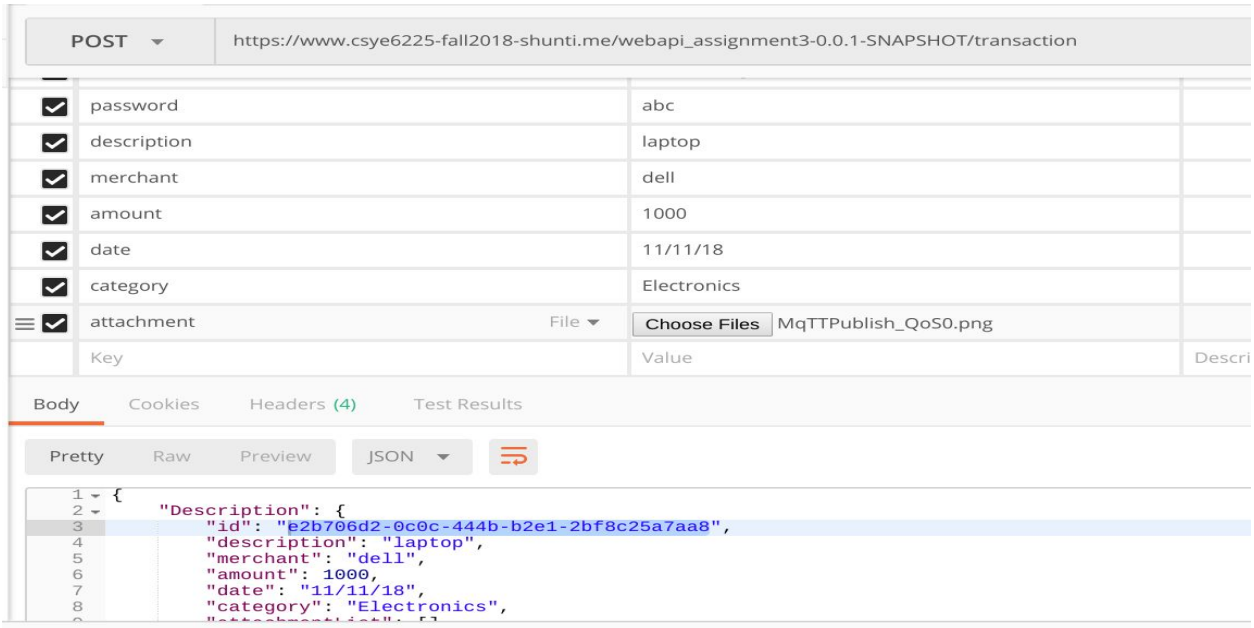
Test Cases:

CASE 1: Unlimited Size of Transaction Body

This is one of the most common vulnerability in a web application. Increase in the body size of a transaction needs exponential increase in processing speed of transactions and needs more storage space in the Data Base. Attackers can send huge number of false transaction requests in a small duration to cause the infrastructure to go out of processing capacity and storage capacity which may result in outage of service.

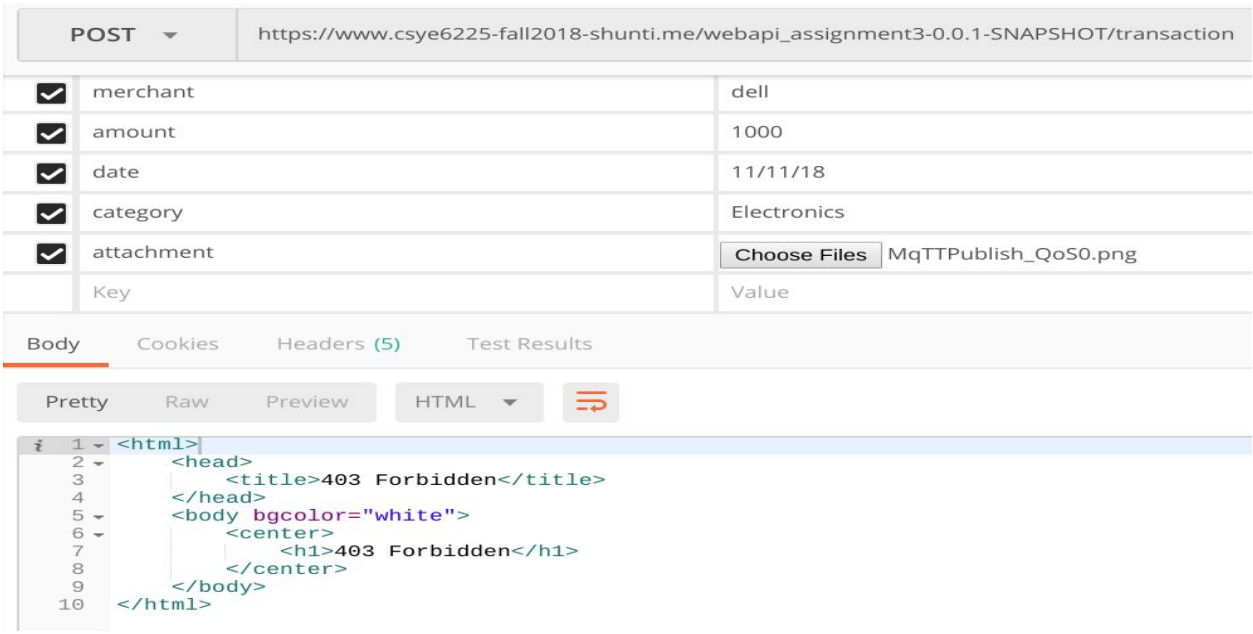
Our application was processing the transaction without considering the size of the transaction body (usually the attachment in the body can have huge size).

Tested the flaw in Postman:



Mitigation: Installed WAF on the load balancer with a constraint which blocks the transaction if the body is more than 50kb.

When tested in Postman, the transaction did not go through:

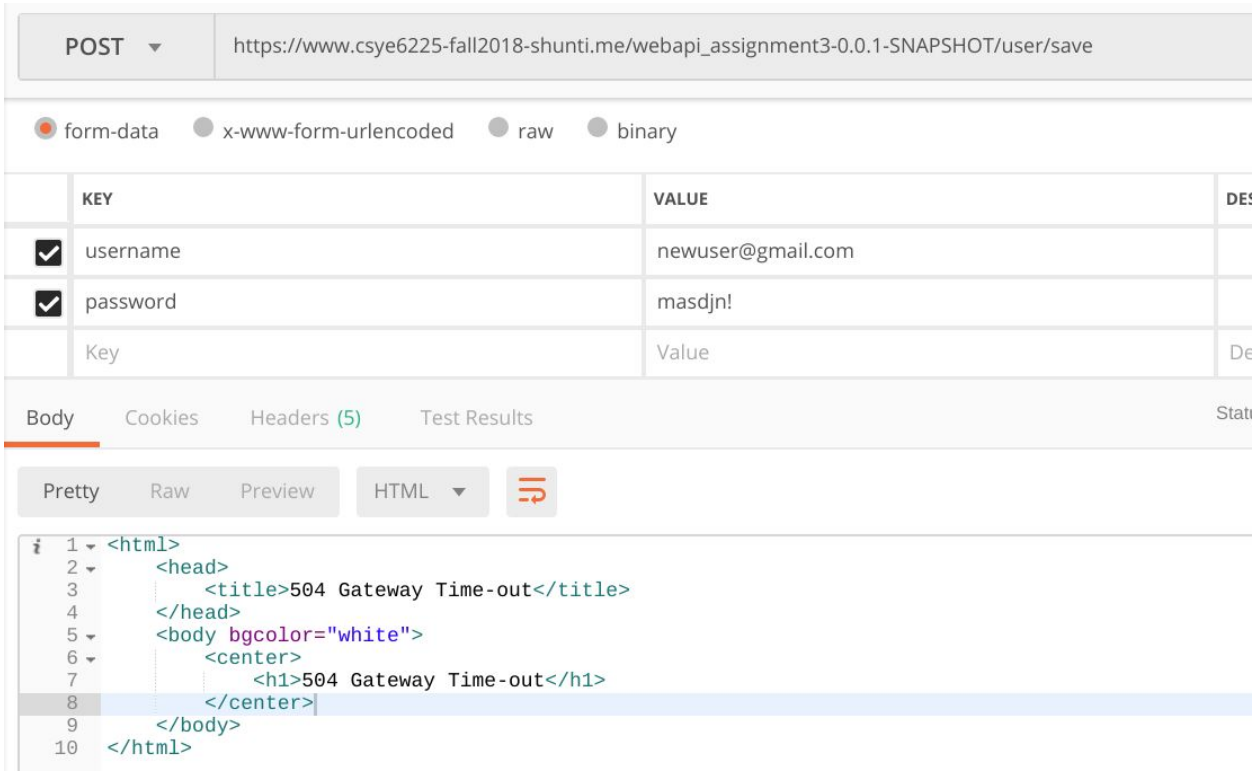


CASE 2: Suspicious users

There are many attackers who are constantly looking to hack into applications and are constantly trying to send malicious data/requests to find out the vulnerability in the application. So these IP addresses should be monitored and blocked from doing any type of transaction with the application.

The malicious users can be blocked using AWS WAF, by setting IP addresses or ranges of IP addresses to the block list. Any request/transaction from these IPs are not processed.
Here, my IP address is 65.96.221.118/32 which is put under blocked IP list

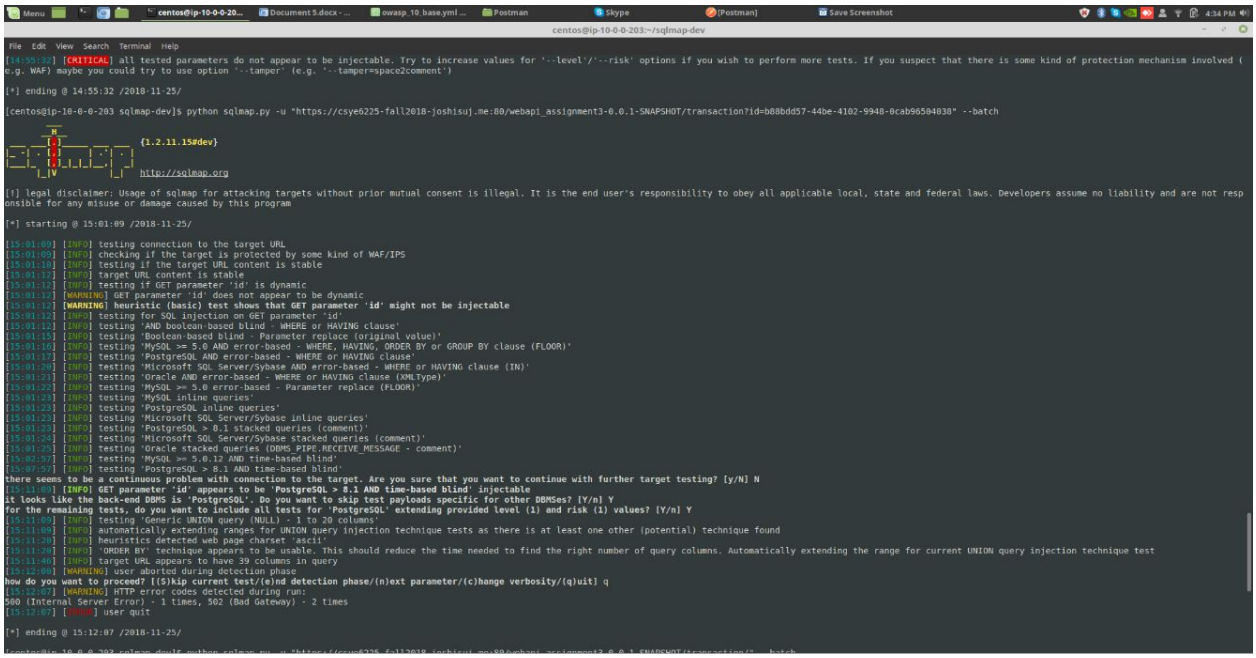
The transaction is clearly blocked from going through because of the firewall (New User registration is failed because the connection is forbidden).



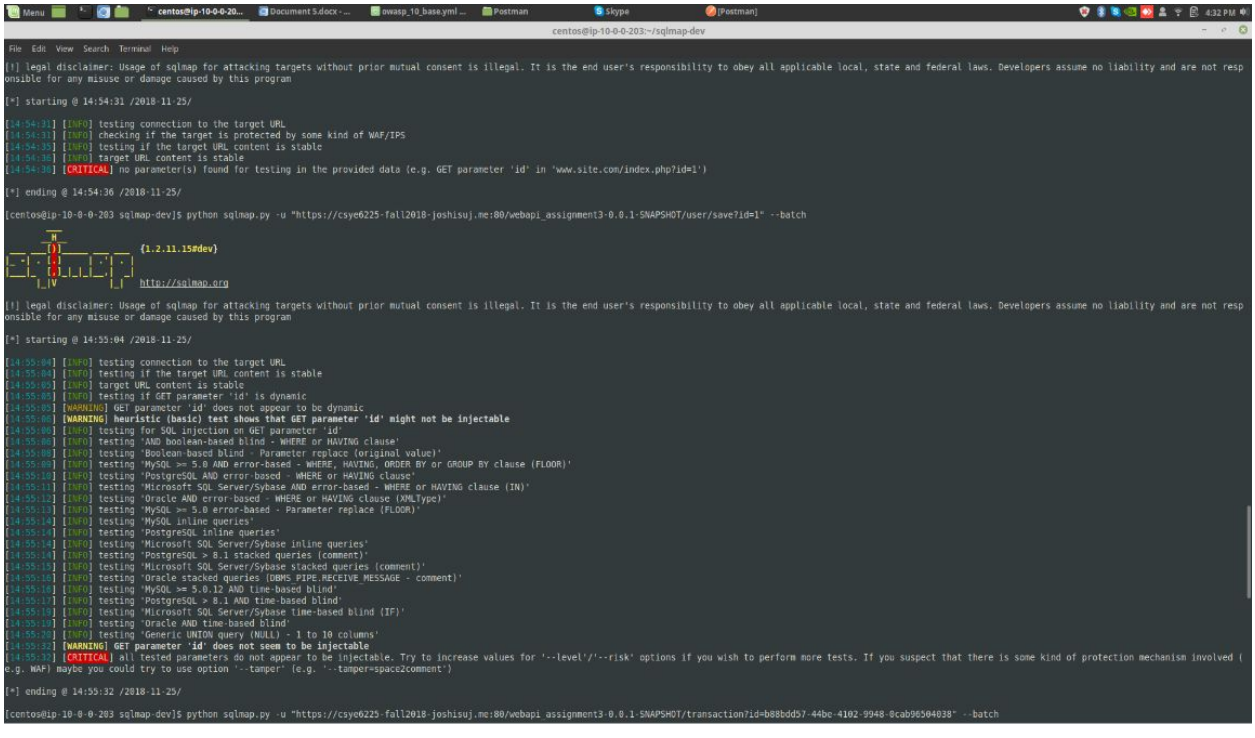
CASE 3: SQL Injection:

A SQL Injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.

We tried SQL injection technique using Automatic SQL injection and database takeover tool (as our application uses Hibernate I used SSH to connect to EC2 and then i tried to access the database).



After mitigating we got the Output cannot be injected:



Additional Security Checks:

We have also implemented the basic security constraints in the web application like SSL over HTTP, which will make the packets not decryptable. And the intercepted packets cannot be interpreted.

TCP packet data (user/save API) captured in WireShark:

Also we have only 1 secure port open:

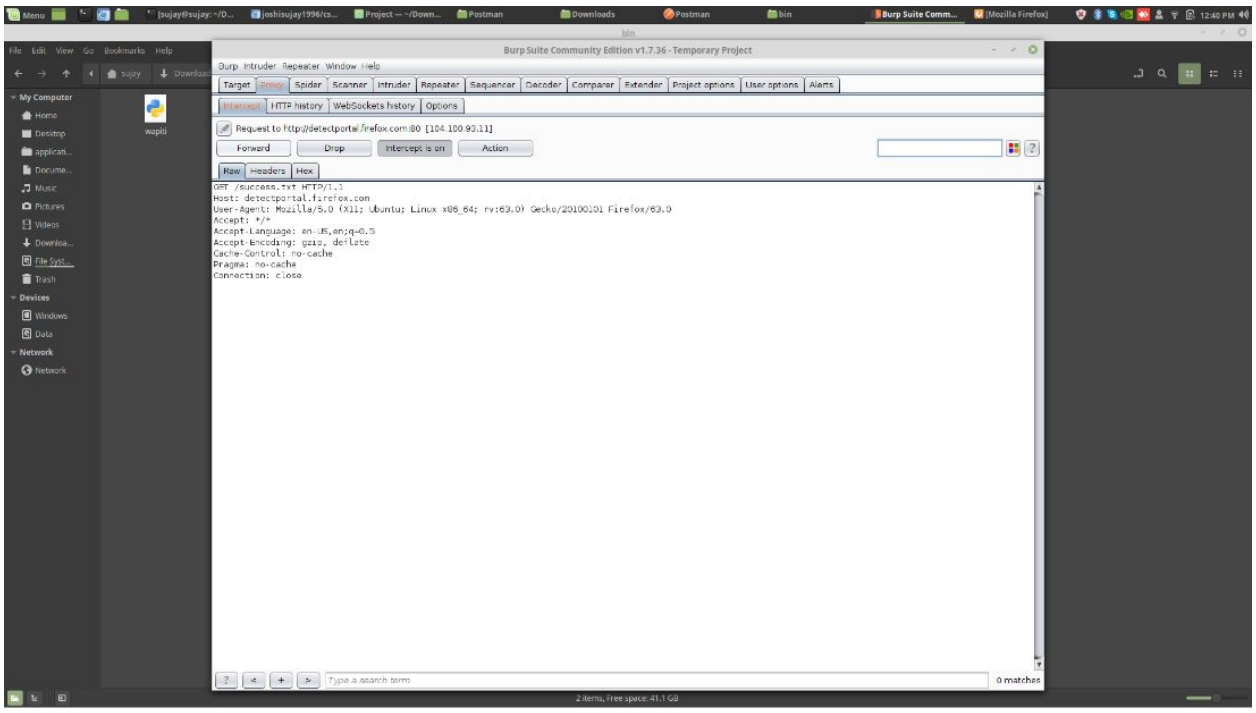
NMap screenshot:

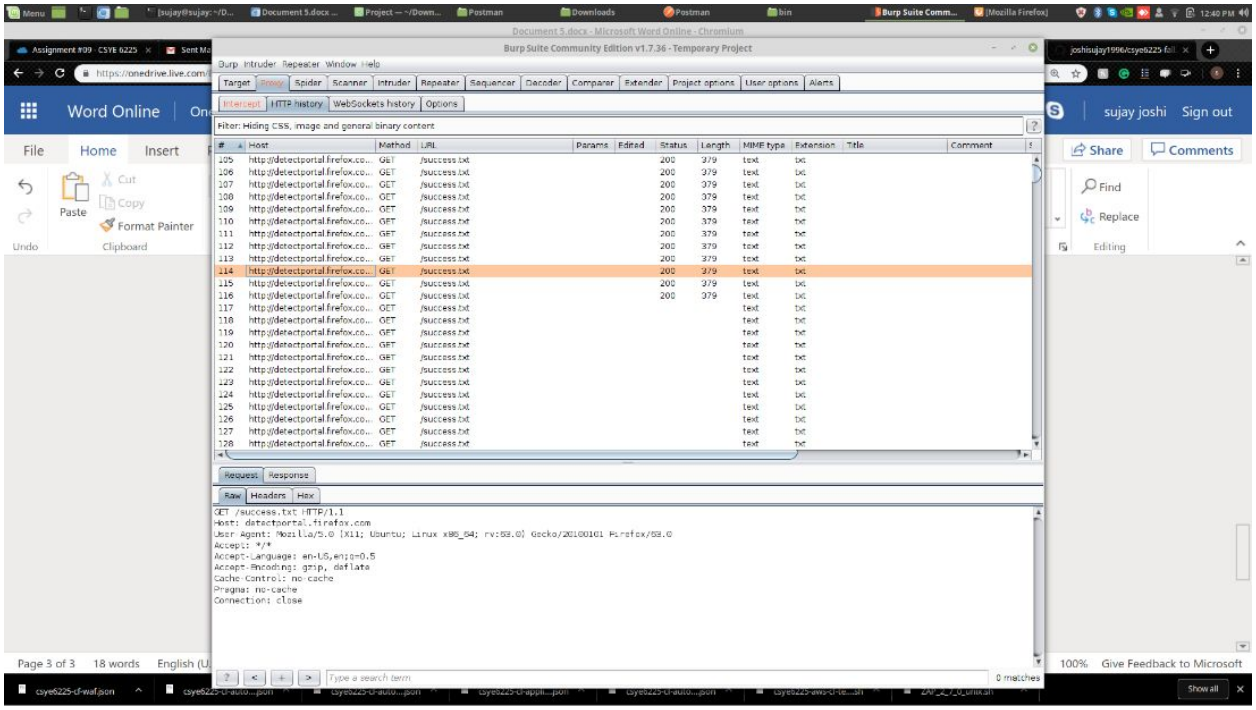
```
[ashwath@ashwathShunti ~]$ nmap www.csye6225-fall2018-shahdhw.me

Starting Nmap 7.60 ( https://nmap.org ) at 2018-11-26 17:39 EST
Nmap scan report for www.csye6225-fall2018-shahdhw.me (18.207.16.220)
Host is up (0.034s latency).
Other addresses for www.csye6225-fall2018-shahdhw.me (not scanned): 34.195.225.88
rDNS record for 18.207.16.220: ec2-18-207-16-220.compute-1.amazonaws.com
Not shown: 995 filtered ports
PORT     STATE  SERVICE
22/tcp   closed ssh
80/tcp   closed http
443/tcp  open   https
3306/tcp closed mysql
8080/tcp closed http-proxy
```

We have also tested Cross Site forgery request test using Brup Suite.

The application is secure because of the https connection:

Assignment 10





Conclusion: With the installation of AWS Web Application Firewall and SSL protocol implementation, we can conclude that the application is reasonably safe from attacks.