Penetration Testing results

1.

Attack Vector: SQL Injection:

Reason for choosing this vector:

SQL injection is one of the most common types of attack vector thus it is important to protect our application from this attack. In these types of attack the user tries to execute SQL queries to gain access as well as to perform malicious actions unknown to the owner. These attacks can lead to data security breach and heavy data losses.

High (Medium)	SQL Injection
Description	SQL injection may be possible.
URL	https://csye6225s19hagalavadigopalakrs.me/?query=query+AND+1%3D1+--+
Method	GET
Parameter	query
Attack	query AND 1=1 --
Instances	1
Solution	Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them. Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality! Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client. Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed characters in user input. Apply the principle of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application.
Other information	The page results were successfully manipulated using the boolean conditions [query AND 1=1 --] and [query AND 1=2 --] The parameter value being modified was stripped from the HTML output for the purposes of the comparison Data was returned for the original parameter. The vulnerability was detected by successfully restricting the data originally returned, by manipulating the parameter
Reference	https://www.owasp.org/index.php/Top_10_2010-A1 https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
CWE Id	89
WASC Id	19
Source ID	1

Screenshot 1: ZAP report without WAF

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	0
Low	0
Informational	0

Alert Detail

Screenshot 2: ZAP report with WAF

2.

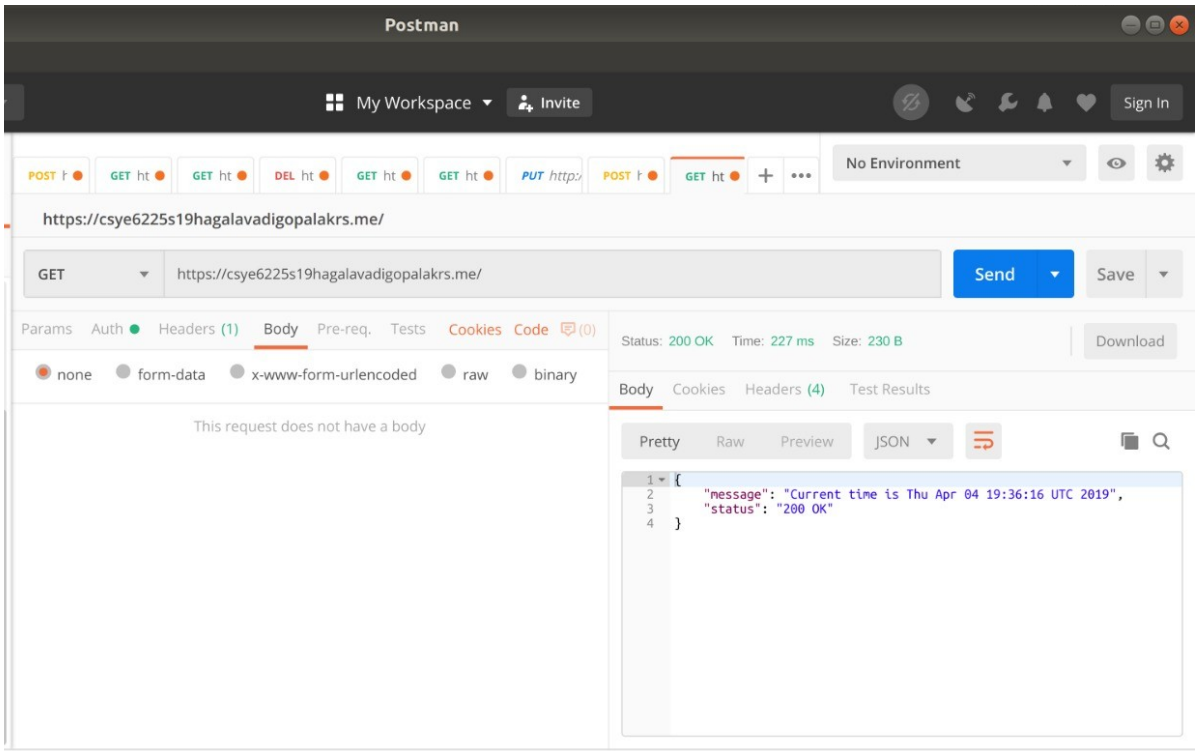
Attack Vector: IP Blacklist

This attack vector allows to matches IP addresses that should not be allowed to access content. IP address blacklisting is a method of protecting Web and other Internet servers from malicious attacks. This is accomplished by setting rules within server software or hardware routers regarding what traffic will be considered an attack, and then preventing the computers creating that traffic from connecting again.

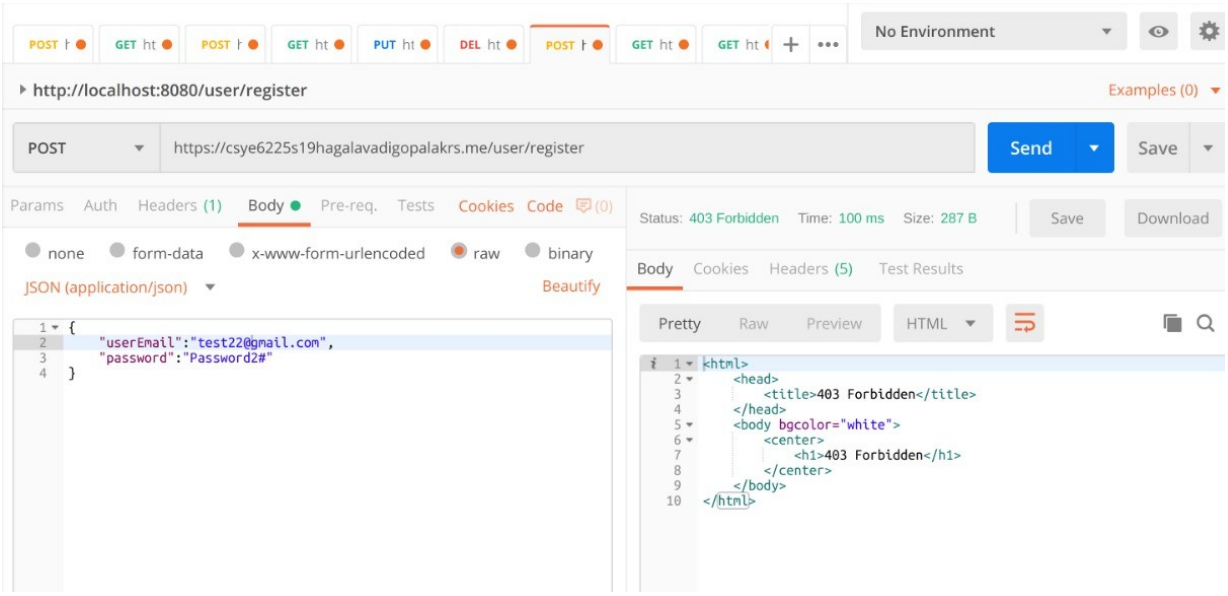
Results:

When the IP is not blocked the the malicious user is able to access the system.

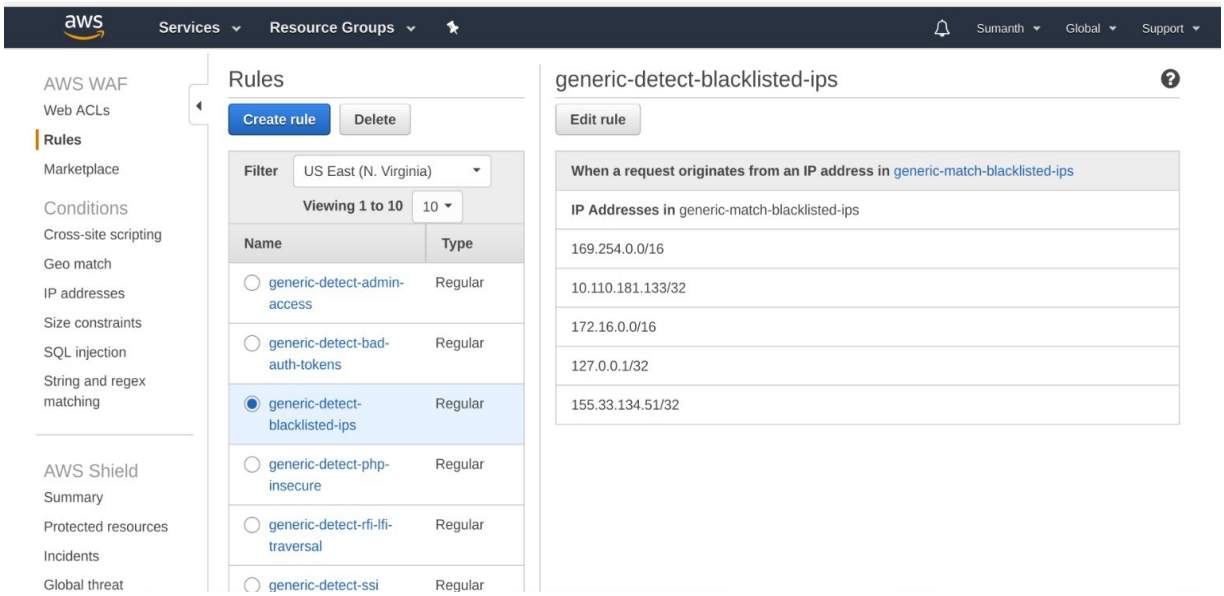
Once the IP is blacklisted, all request coming from the blacklisted IP are blocked and served with a 403 HTTP status code is received.



Screenshot 3: When WAF is not inplace to blacklist IP



Screenshot 4: Blacklisted IP receives 403 with WAF



Reason for choosing this vector:

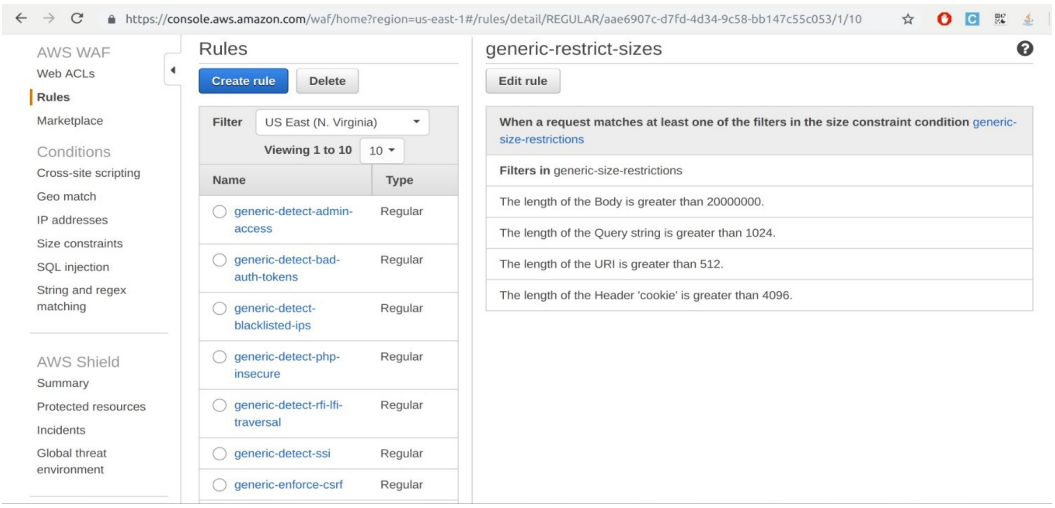
There are a lot of malicious users who try to break the system for there benefit or to simply cause harm to the system. It is important to block this user IPs, preventing them from accessing the system.

3.

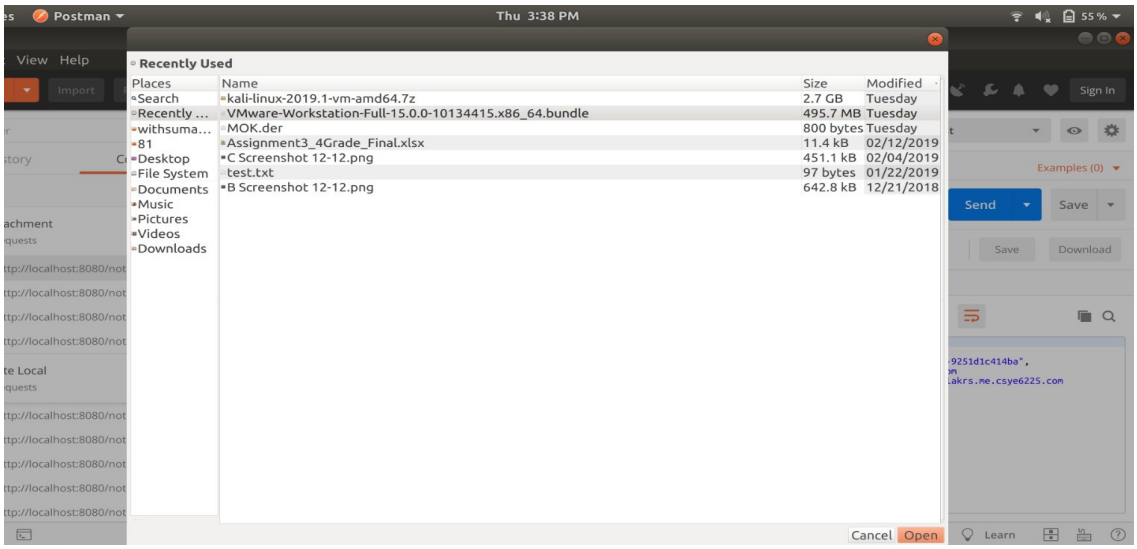
Attack Vector: Body size constrain

Reason for choosing this vector: The user is not allowed to post attachments larger than 20MB. This is implemented to restrict the size of the content to be within the allowed storage limit. We might have a limit on storing the content submitted by the user. To implement this limit, we stop the user from attaching receipts which are greater than the specified size say 20MB.

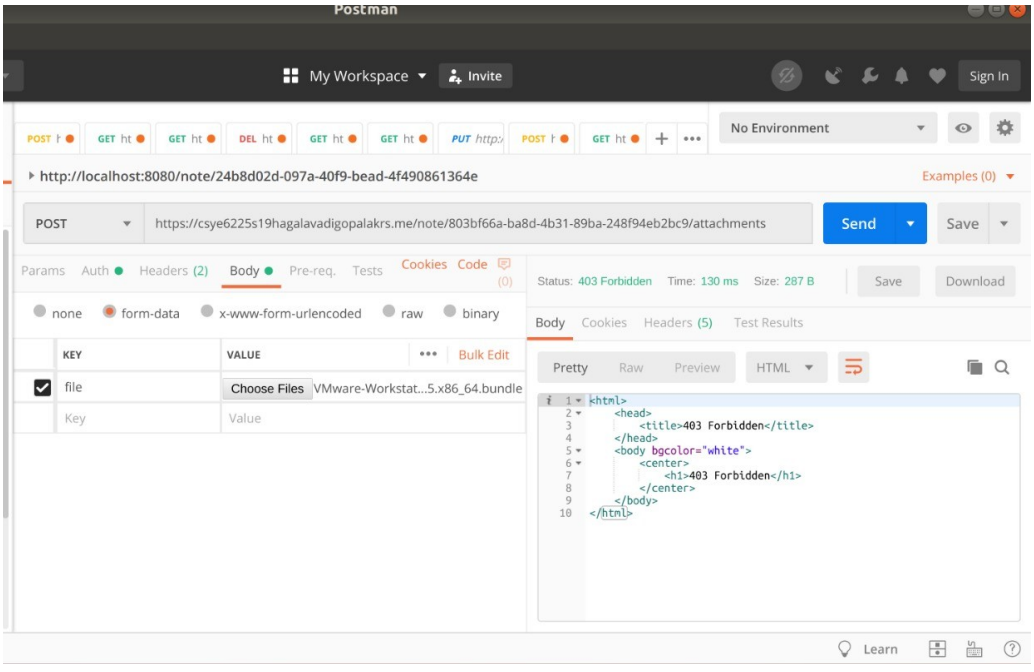
Results: A request that is larger than the specified size is not accepted by the web application



Screenshot 5: Setting WAF to restrict body size to 20MB



Screenshot 6: Trying to upload file size of ~500MB



Screenshot 7: request with more than 20MB body size receives 403