Penetration Testing 1 - SQL Injection

1. Attack Vector

SQL Injection (To get all note information from an endpoint that is supposed to return one note when calling GET request)

2. Result

To implement SQL injection, firstly I opened an endpoint named “/notes/{Id}” to get the note information of a certain note.

figure

Secondly, to attack the web application using sqlmap in kali linux operating system, I used the phrase “1' or '1' = '1” as the note id in the url endpoint instead of the correct note id.
Then I started up sqlmap and use the url http://localhost:8080/notes/1' or '1' = '1 to send a GET request as an attack.

figure

Without WAF, the request returned all of the notes which means the attack succeeded.

figure

With WAF added to our EC2 instances, the request with invalid content in the url will be blocked, as shown in figure below.

figure

3. Why did you choose this specific attack vector

First of all, SQL injection in the top 1 attack as defined in OWASP, it is used to attack data-driven applications, in which diabolical 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.
Also, SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server
Most importantly, in our project, we used MyBatis to read and write the database. When using the phrase “1' or '1' = '1” injected into the sql, the where condition “where noteId='${noteId}'”will be replaced by “where noteId=1' or '1' = '1” which will always be true, so that all of the note will be

returned. Our web application is likely to be attacked by this kind of SQL injection.

```
<select id="getNoteByIdSQL" flushCache="true" parameterType="String"  resultMap="BaseResultMap">
    select
    *
    from note
    where
    noteId='${noteId}'
</select>
```

Penetration Testing 2 - Cross Site Scripting Attack

1. **Attack Vector**

   Cross Site Scripting Attack (XSS)

2. **Result**

   When uploading a html file as attachment:

   Without WAF, the POST request will be sent without any error, which means the cross site scripting attack succeeded.

   figure

   With WAF added to our EC2 instances, the POST request will be blocked, as shown in figure below.

   figure

3. **Why did you choose this specific attack vector**

   Cross Site Scripting Attack is the top 3 as defined in OWASP. Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy. Cross-site scripting carried out on websites accounted for roughly 84% of all security vulnerabilities documented by Symantec as of 2007. In 2017, XSS is still considered a major threat vector. XSS effects vary in range from petty nuisance to significant security risk, depending on the sensitivity of the data handled by the vulnerable site and the nature of any security mitigation implemented by the site's owner.
   In our project, the wafrXSSRule rule stipulates that user cannot upload html file to the web application in case that XSS attack will inject some invalid scripts to the app and get data that needs certain authorization to access.

Penetration Testing 3 - Abnormal requests

1. **Attack Vector**

   Abnormal requests with size restrictions

2. **Result**

   When sending content with a size of 4096 and above within the body:

   Without WAF, the POST request will be sent without any error, which means the abnormal requests with size restrictions attack succeeded.

   figure

   With WAF added to our EC2 instances, the POST request will be blocked, as shown in figure below.

   figure

3. **Why did you choose this specific attack vector**
   There might be