Methods to identify misconfigurations and bypass Web Application Firewalls

-Revanth Pendyala

Committee Members:

Dr. Yingshu Li

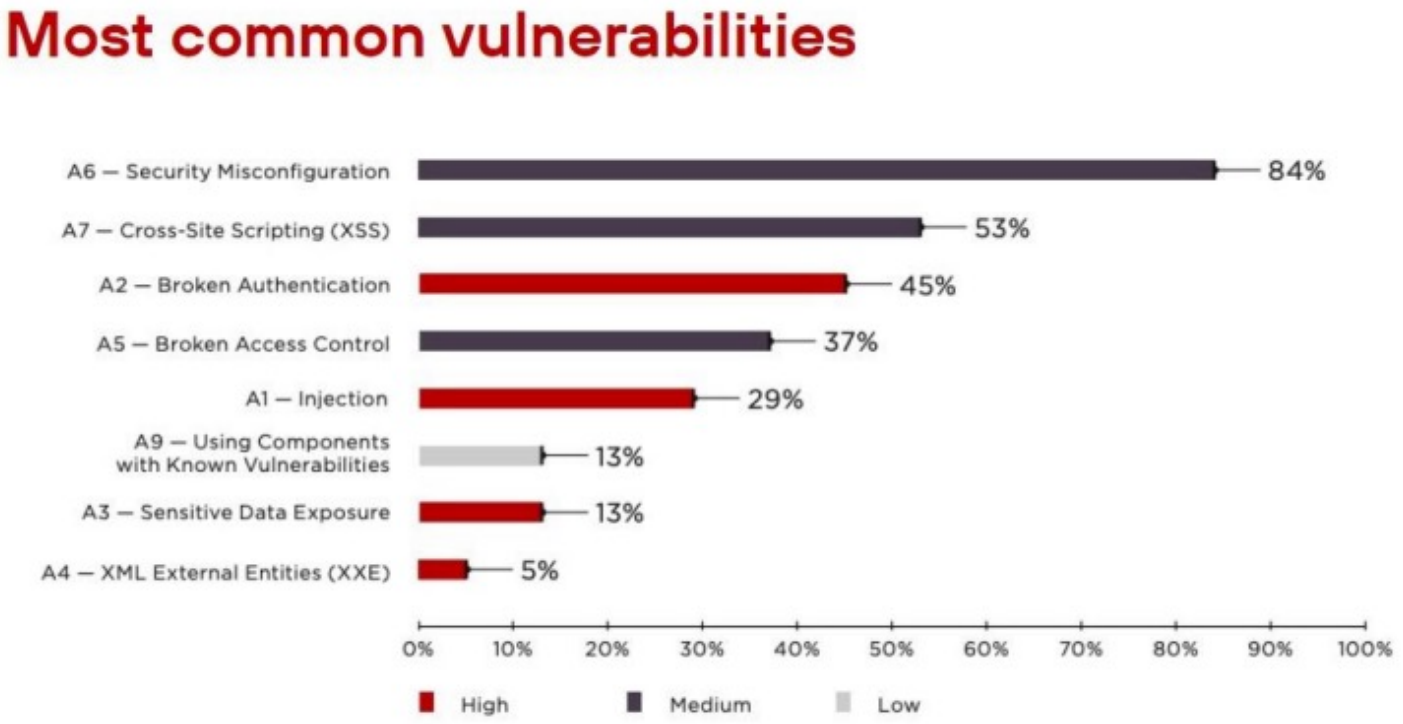
Dr. Yanqing Zhang

Index

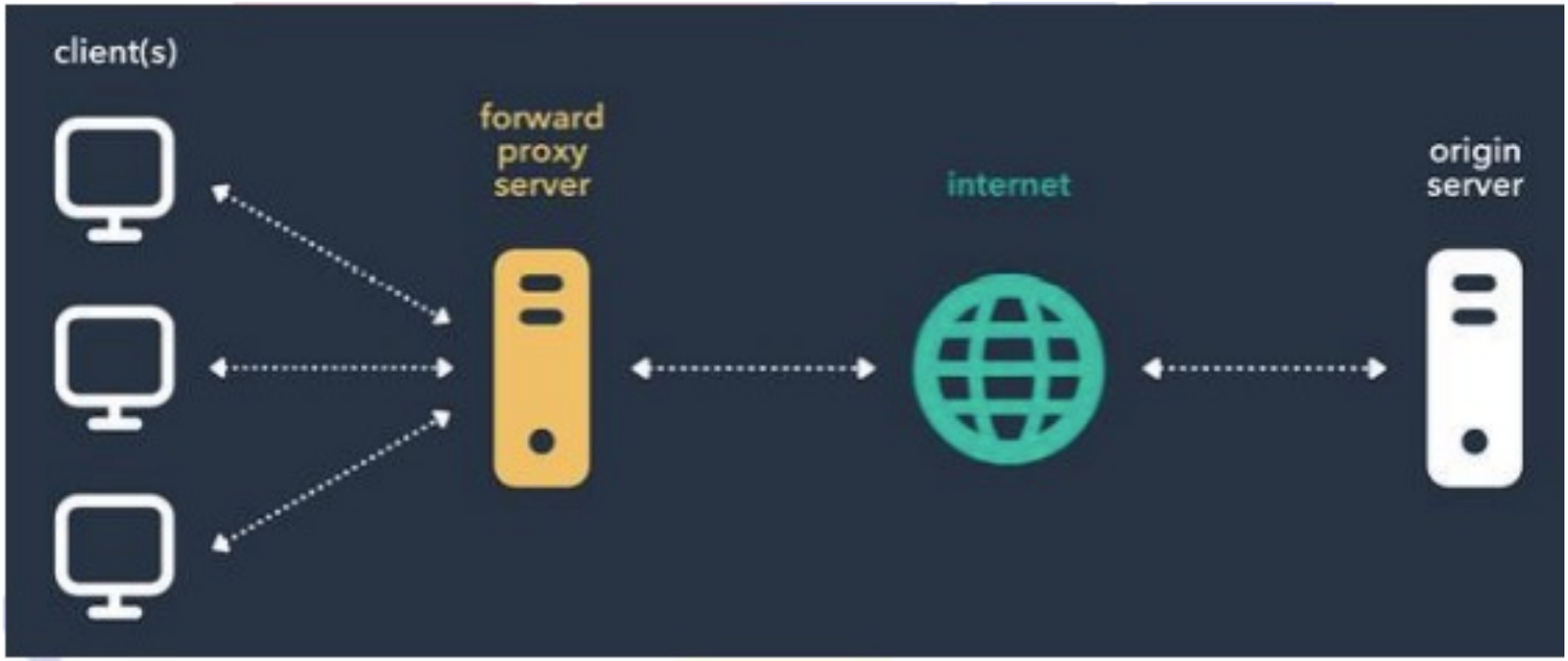
Problem Statement

- Web application attacks are increasing every day, year over year.
- The researchers ascribed 21% of the incidents in the Verizon 2021 Data Breach Investigation Reports to misconfigurations.
- A server owned and run by Capital One that included 1,40,000 SSN, 1 million Insurance Numbers, 80,000 bank account numbers, and some unknown PII was accessed by a former AWS software engineer who took advantage of a misconfigured WAF.
- **Goal**: To build an CLI tool to test for web application firewall(WAF) misconfigurations and bypass waf.

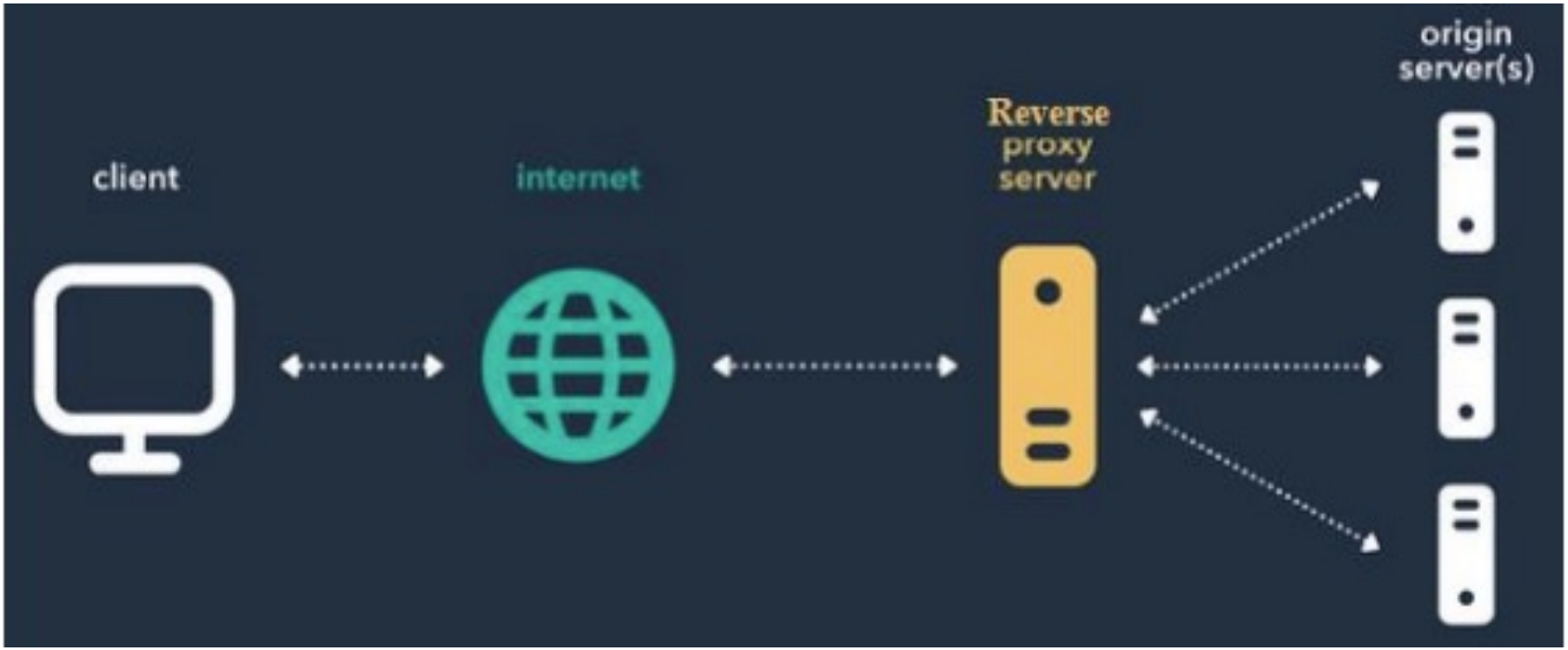
Vulnerabilities



Forward Proxy Architecture



Reverse Proxy Architecture

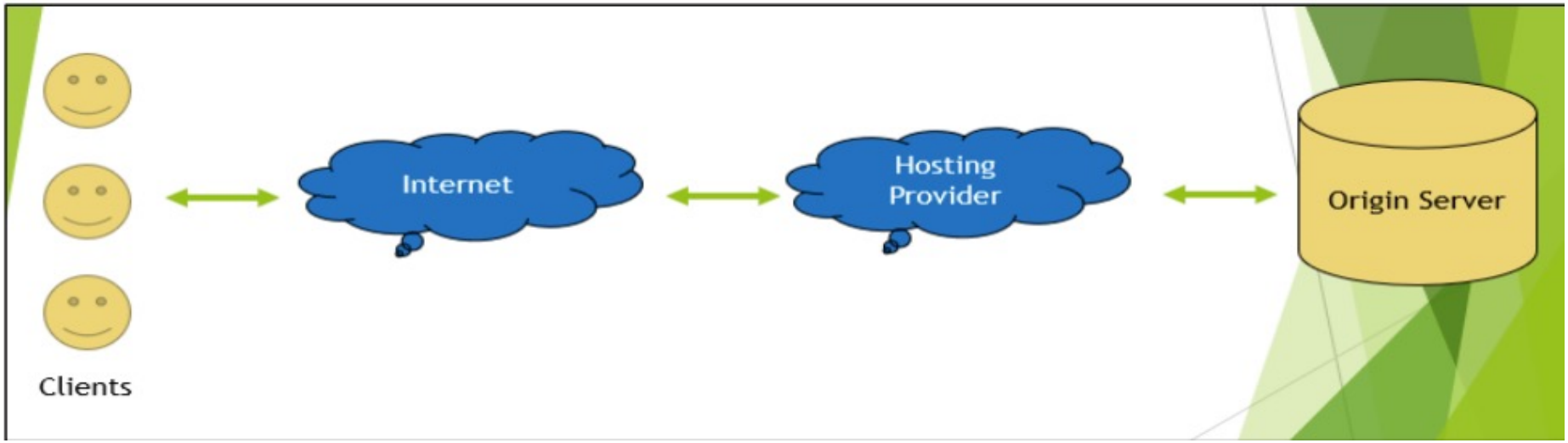


Introduction : WAF

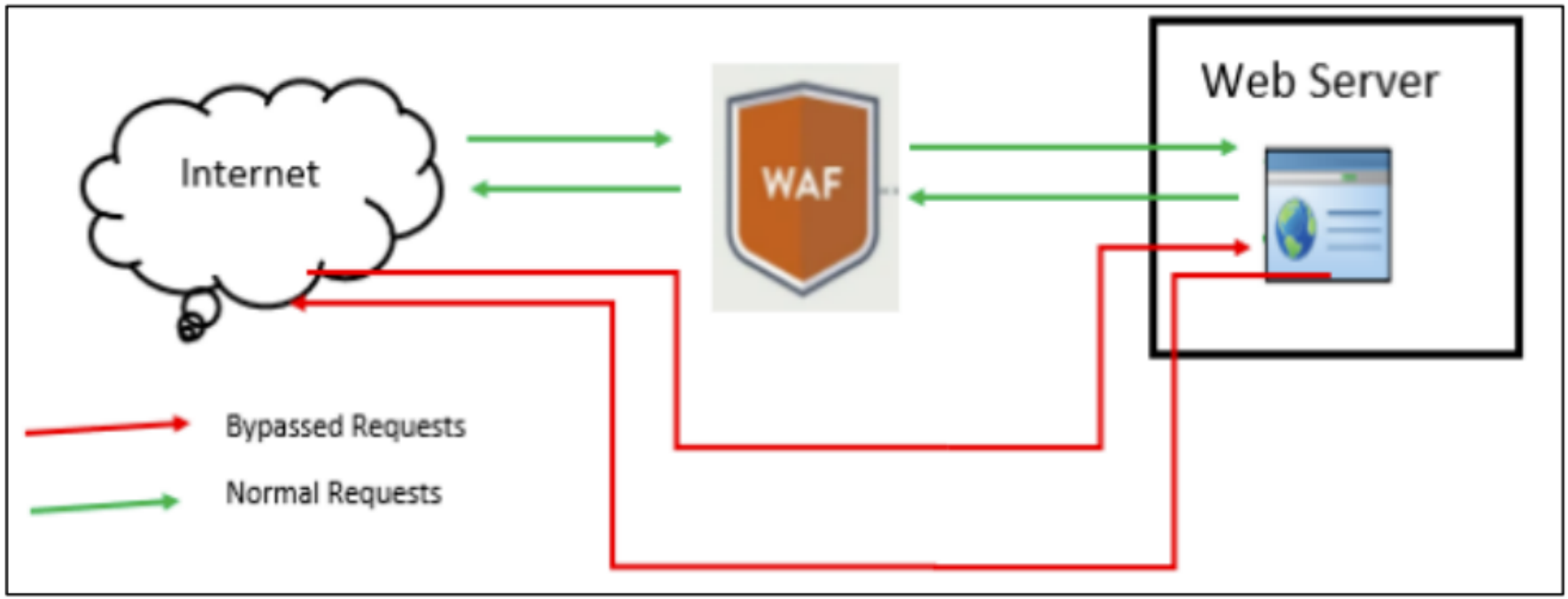
A WAF or web application firewall helps protect web applications by filtering and monitoring HTTP traffic between a web application and the Internet. It typically protects web applications from attacks such as cross-site forgery, cross-site-scripting (XSS), file inclusion, and SQL injection, among others.

A WAF operates through a set of rules often called policies. These policies aim to protect against vulnerabilities in the application by filtering out malicious traffic.

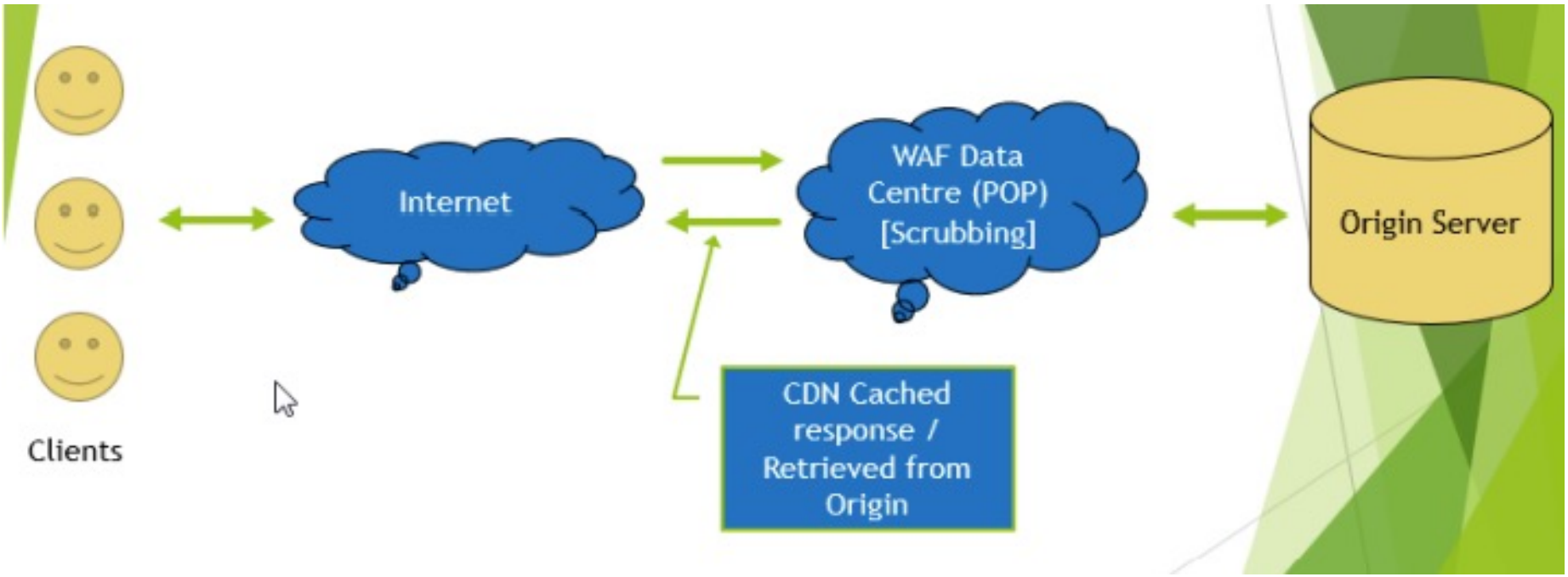
Traffic Flow Without WAF



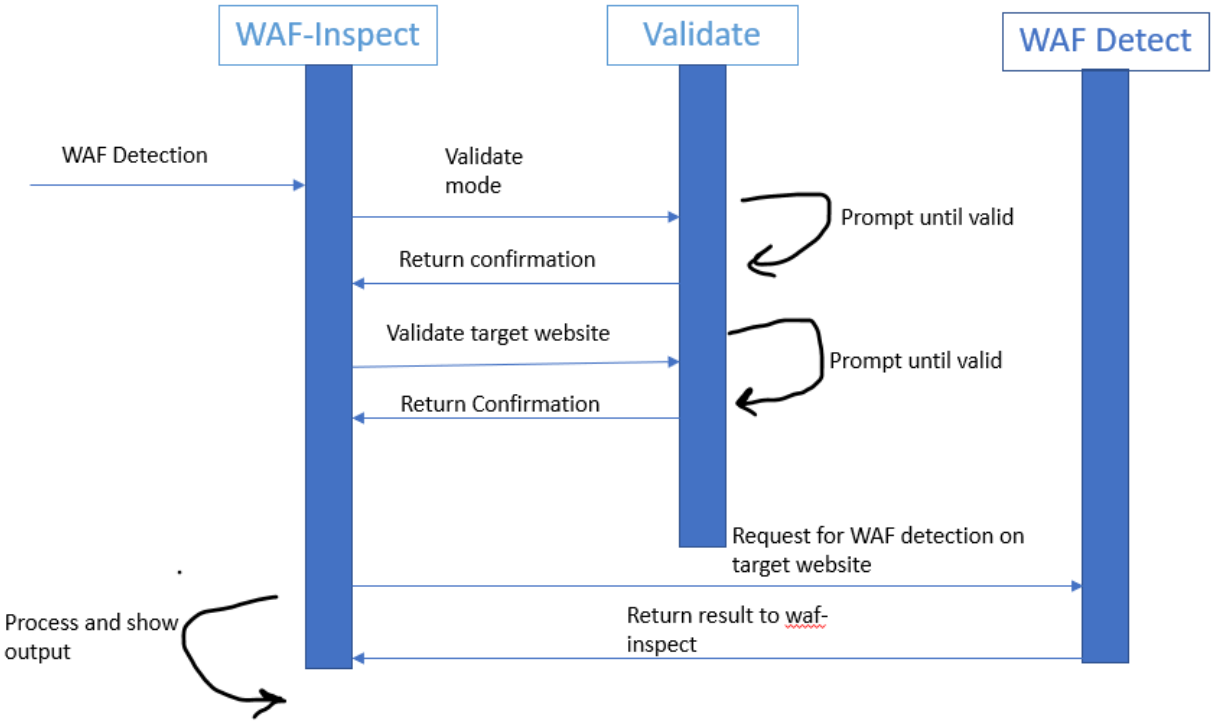
WAF Deployment



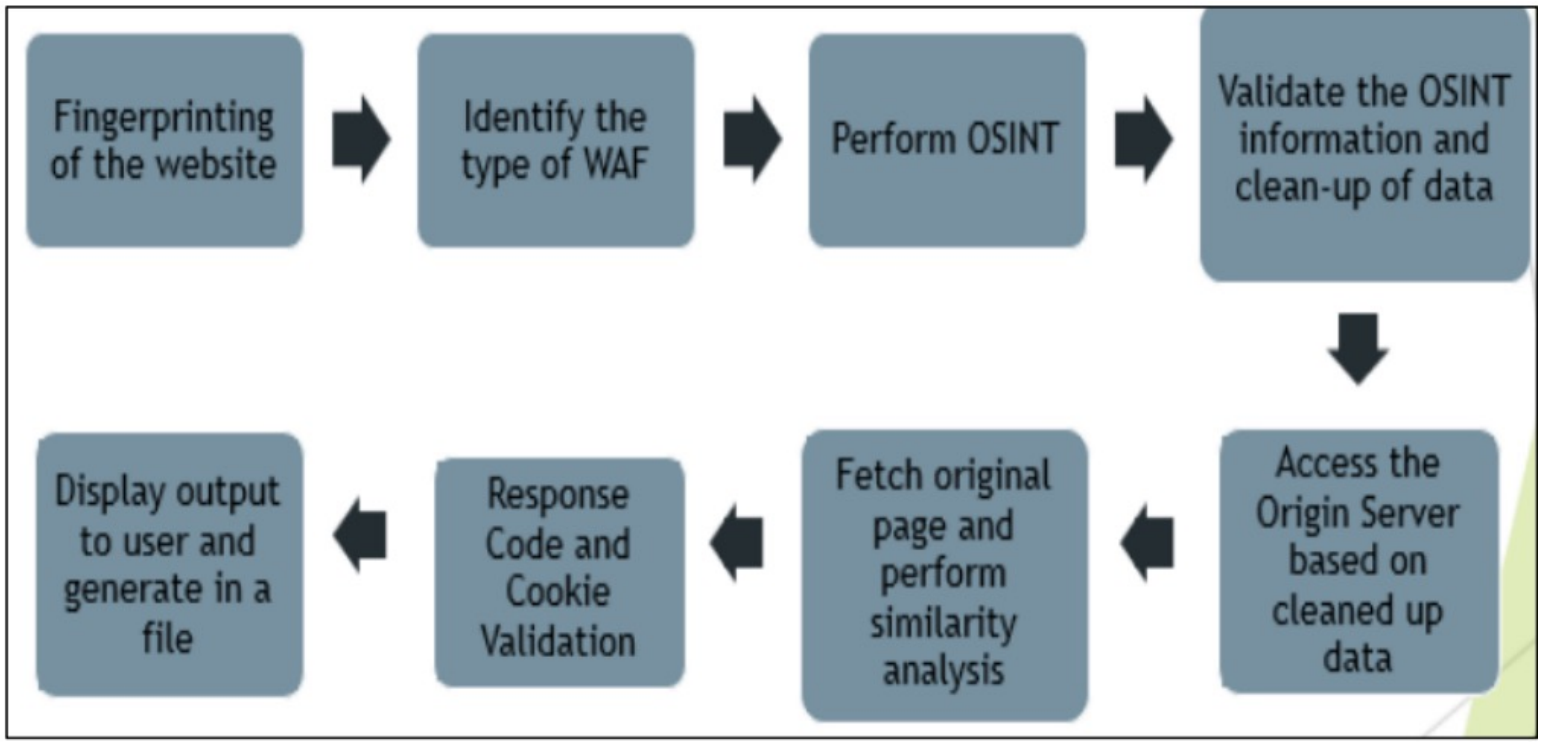
Traffic Flow With WAF Deployment



WAF Detection



OSINT Based Detection



Results

```
┌──(kali㉿  kali)-[~/Desktop/Project Code]
└─$ python3 waf-inspect.py

================================================================================

   WAF Inspect - Revanth Pendyala

================================================================================

Functionalities :

[1] WAF Detection
[2] OSINT Based WAF Bypass Test
[3] Custom WAF Bypass Test

Please enter the number of functionality you want to use : 1
Please enter the target website :www.epam.com

[+] Loading modules for WAF Detection
[+] Cheking if WAF in place
[+] www.epam.com    behind Cloudfront (Amazon) WAF.
[~] Number of requests: 2
```

Results

```
[+] Performing OSINT to find Origin Servers
[+] Origins discovered during OSINT :

172.67.164.180
104.21.66.222
104.24.114.228
104.24.115.228
172.67.164.180
13.127.132.185
166.62.28.6
50.63.202.65
50.63.202.81
166.62.28.6
166.62.28.166

[+] Origins left after exclusing WAF Public IPs :

172.67.164.180
104.21.66.222
104.24.114.228
104.24.115.228
172.67.164.180
13.127.132.185
166.62.28.6
50.63.202.65
50.63.202.81
166.62.28.6
166.62.28.166
```

Results

```
[+] Performing the Bypass test on HTTPS for the origin : 172.67.164.180

Bypass for the origin : 172.67.164.180
SSL Error, try with / without www for the origin : 172.67.164.180
--------------------------------------------------------------------------------

[+] Performing the Bypass test on HTTP for the origin : 104.21.66.222

[*] HTTP Response Code : 200
Bypass for the origin : 104.21.66.222
--------------------------------------------------------------------------------

[+] Performing the Bypass test on HTTPS for the origin : 104.21.66.222

Bypass for the origin : 104.21.66.222
SSL Error, try with / without www for the origin : 104.21.66.222
--------------------------------------------------------------------------------

[+] Performing the Bypass test on HTTP for the origin : 104.24.114.228

Connection Timed Out
No Bypass for the origin : 104.24.114.228
--------------------------------------------------------------------------------

[+] Performing the Bypass test on HTTPS for the origin : 104.24.114.228

Connection Timed Out
No Bypass for the origin : 104.24.114.228
--------------------------------------------------------------------------------

[+] Performing the Bypass test on HTTP for the origin : 104.24.115.228

Connection Timed Out
No Bypass for the origin : 104.24.115.228
```

Demo

Future Work

- Can be developed further to test for vulnerabilities like XSS, Injection Attacks.

Thank You