Design Network Deception Solution

Presented By:

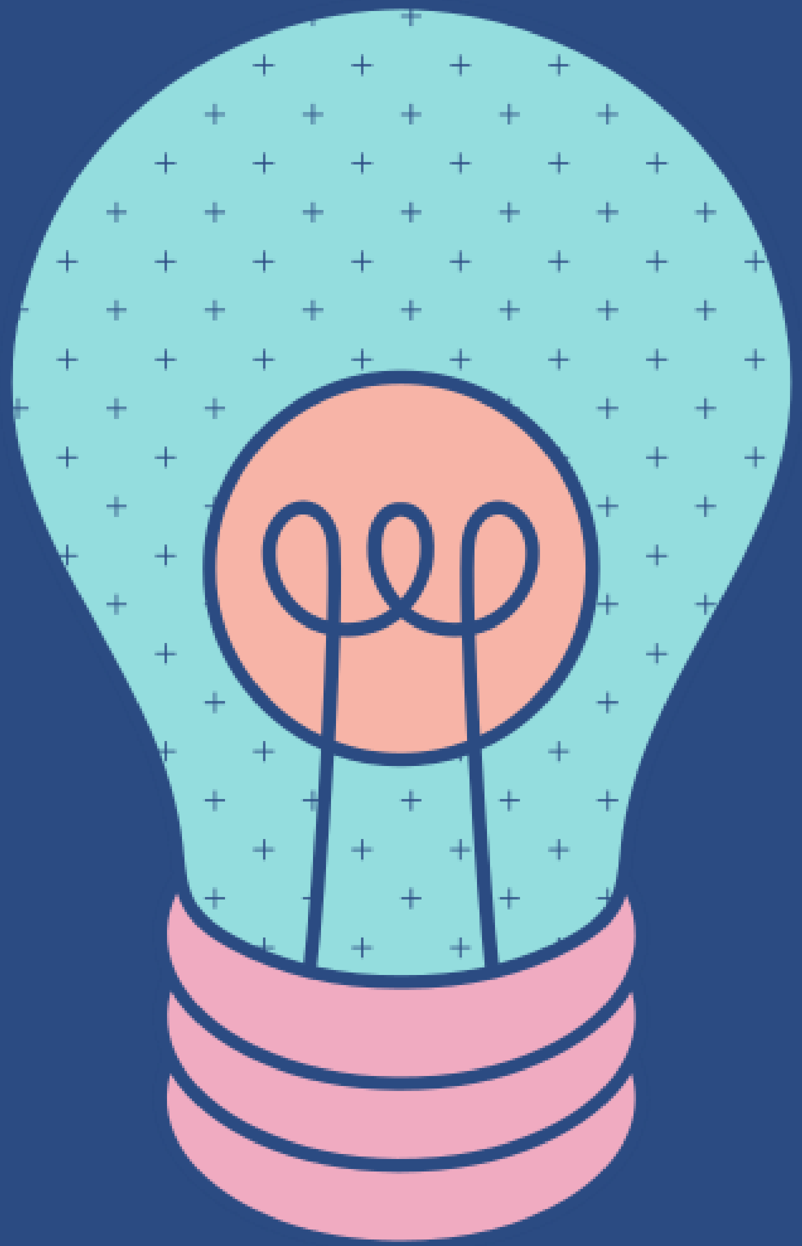
Fatma Mohamed Ahmed

Mayar Mohamed Saad

Manar Abdallah Tawfik

Salwa Youssef Attia Attia

Agenda



Introduction

Our project develops an automated network deception system for swift threat detection and enhanced proactive defenses.

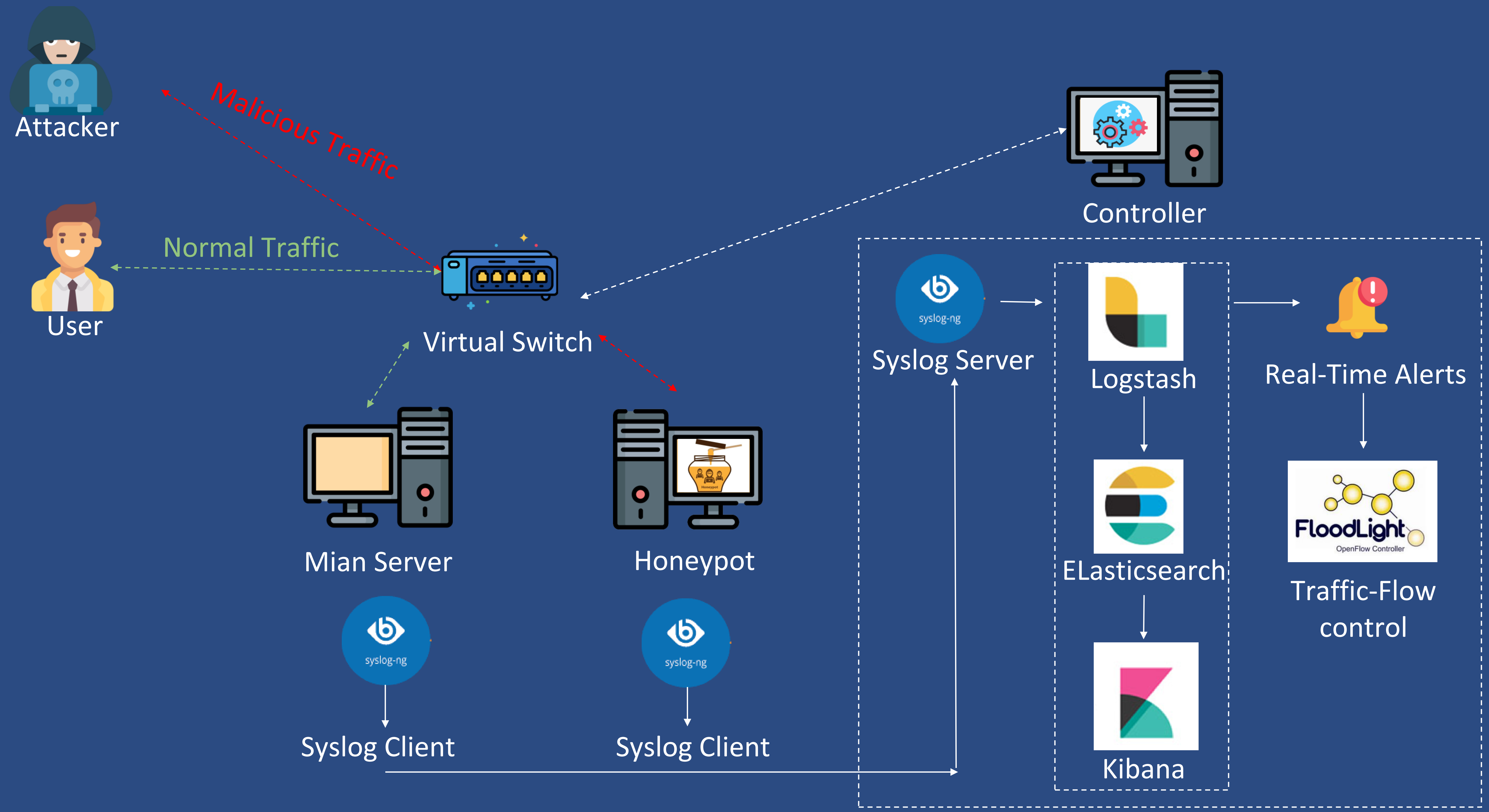
Problem Definition

Deploy Automated Decoys and Honeypots

Centralized Logging System

Real-Time Alerting System

Methodology and System Architecture



System Architecture

System Components

System Infrastructure

- KVM Virtualization Environment
- Open vSwitch (OVS)

The Main-server

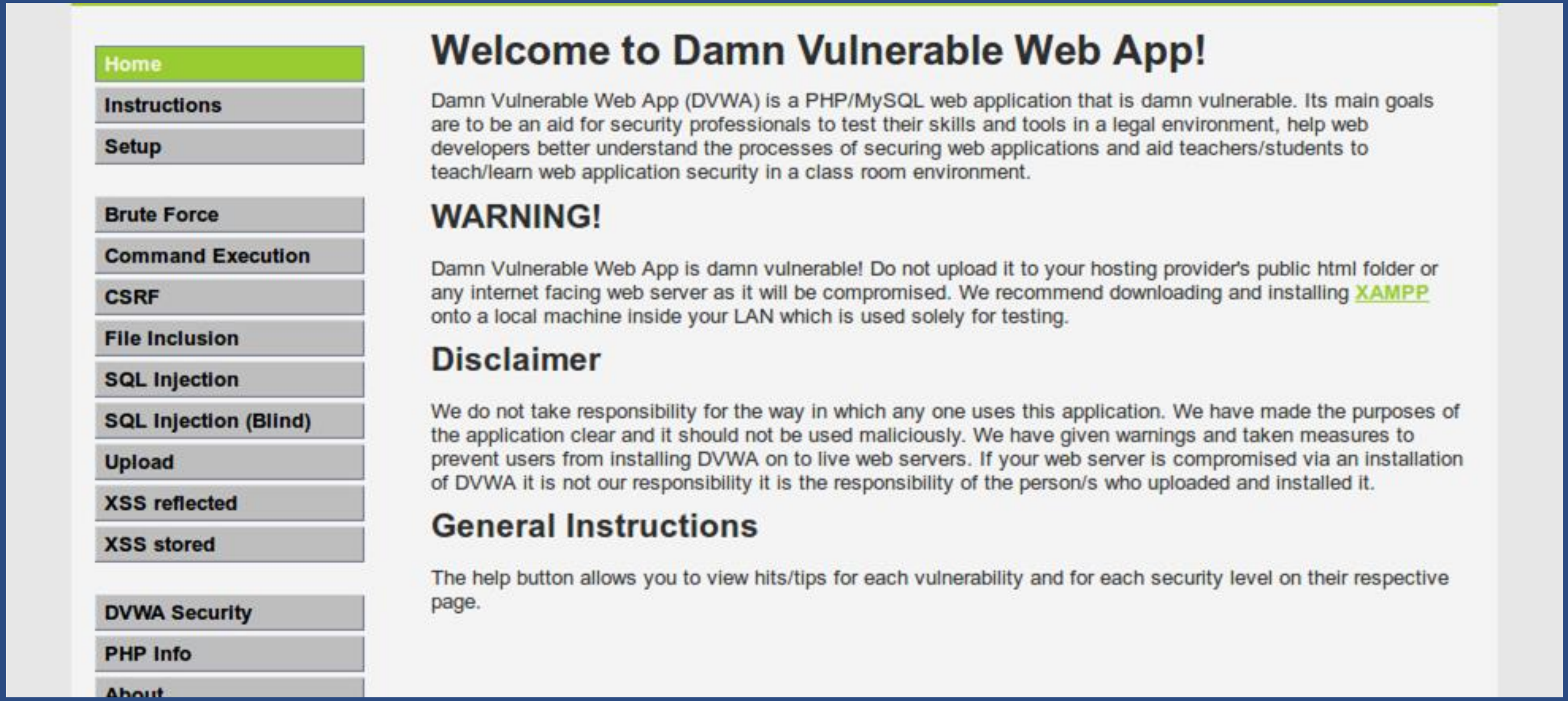
- Web-server
- SSH-server

Network Security Components

- Snort IDS
- ModSecurity WAF
- Syslog-ng
- ELK SIEM Solution

Dynamic Flow Rule Script

- Floodlight SDN Controller
- Python Script for SDN Flow Rules



Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

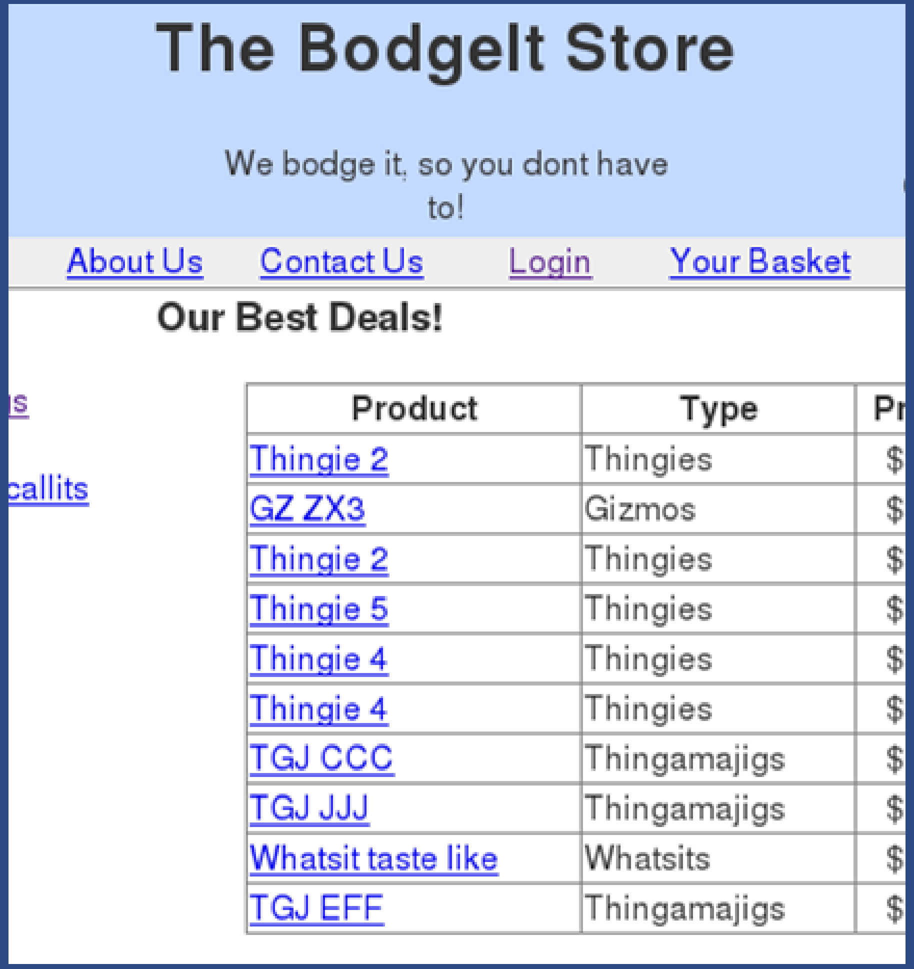
Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

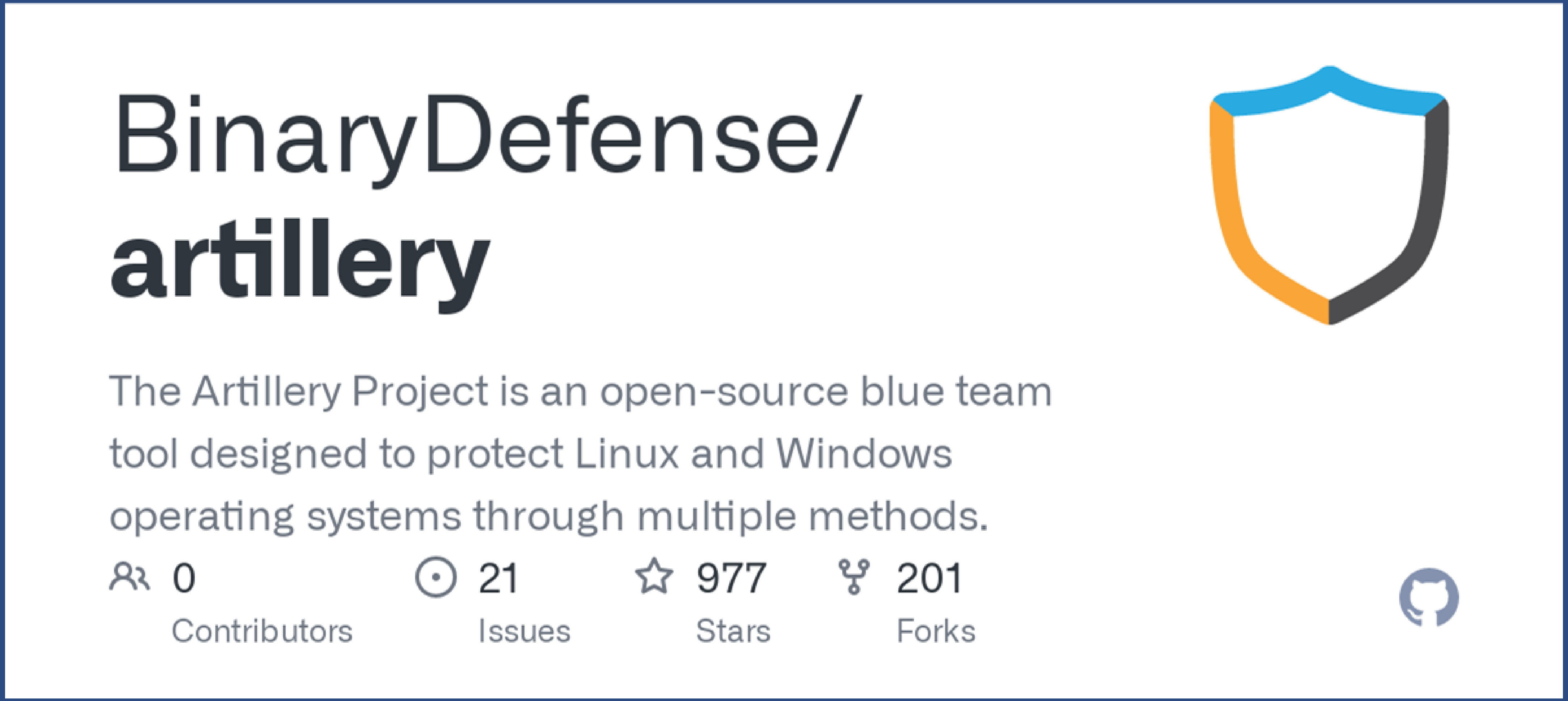
The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.



Our Best Deals!

Product	Type	Pr
Thingie 2	Thingies	$
GZ ZX3	Gizmos	$
Thingie 2	Thingies	$
Thingie 5	Thingies	$
Thingie 4	Thingies	$
Thingie 4	Thingies	$
TGJ CCC	Thingamajigs	$
TGJ JJJ	Thingamajigs	$
Whatsit taste like	Whatsits	$
TGJ EFF	Thingamajigs	$

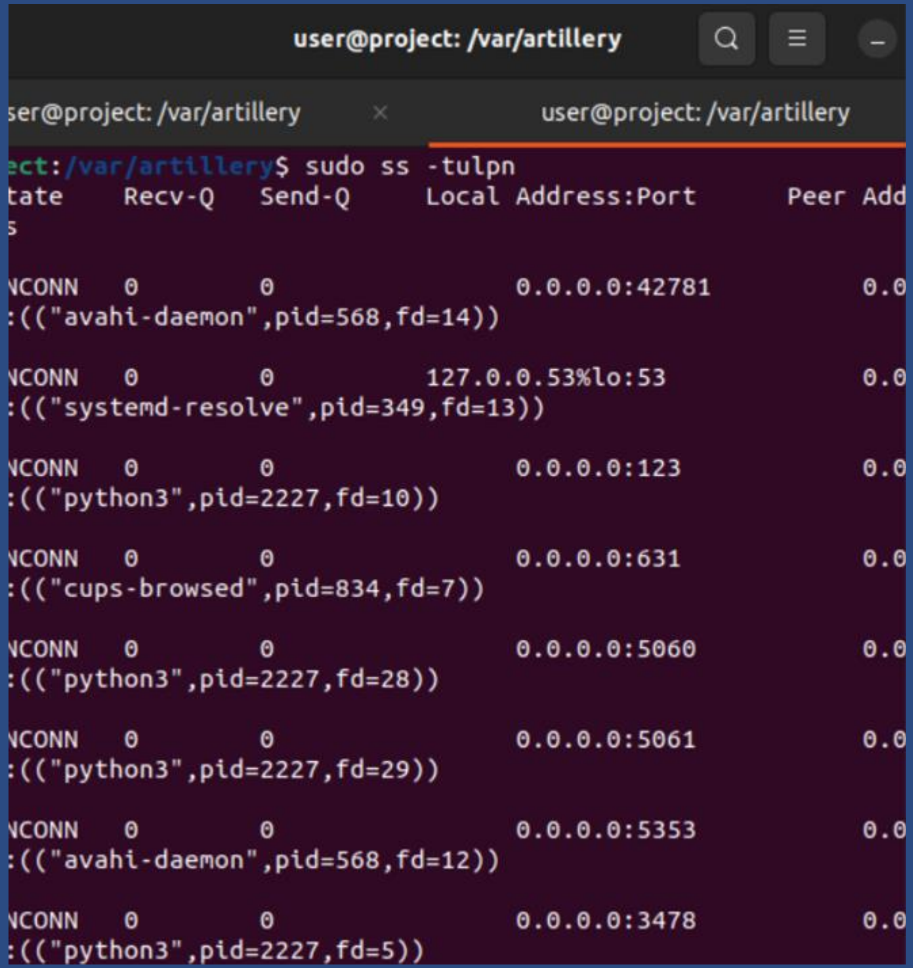
The Main-Server



BinaryDefense/
artillery

The Artillery Project is an open-source blue team tool designed to protect Linux and Windows operating systems through multiple methods.

0 Contributors · 21 Issues · 977 Stars · 201 Forks


user@project: /var/artillery
ser@project: /var/artillery
user@project: /var/artillery

```
ect:/var/artillery$ sudo ss -tulpn
tate     Recv-Q   Send-Q     Local Address:Port       Peer Add
s

NCONN    0        0              0.0.0.0:42781          0.0
:(("avahi-daemon",pid=568,fd=14))

NCONN    0        0          127.0.0.53%lo:53           0.0
:(("systemd-resolve",pid=349,fd=13))

NCONN    0        0              0.0.0.0:123            0.0
:(("python3",pid=2227,fd=10))

NCONN    0        0              0.0.0.0:631            0.0
:(("cups-browsed",pid=834,fd=7))

NCONN    0        0              0.0.0.0:5060           0.0
:(("python3",pid=2227,fd=28))

NCONN    0        0              0.0.0.0:5061           0.0
:(("python3",pid=2227,fd=29))

NCONN    0        0              0.0.0.0:5353           0.0
:(("avahi-daemon",pid=568,fd=12))

NCONN    0        0              0.0.0.0:3478           0.0
:(("python3",pid=2227,fd=5))
```

The Honeypot

Network Security Components

Snort

- Alert DOS Attacks
- SSH brute Forcing
- Unauthorized access to assets

ModSecurity

- Secure the web server against OWASP Top 10 vulnerabilities

Syslog-ng

- Provide Real-Time Centralized Logging

ELK SIEM Solution

- Used for Logs Analysis and Visualization
- Real-Time Alerting

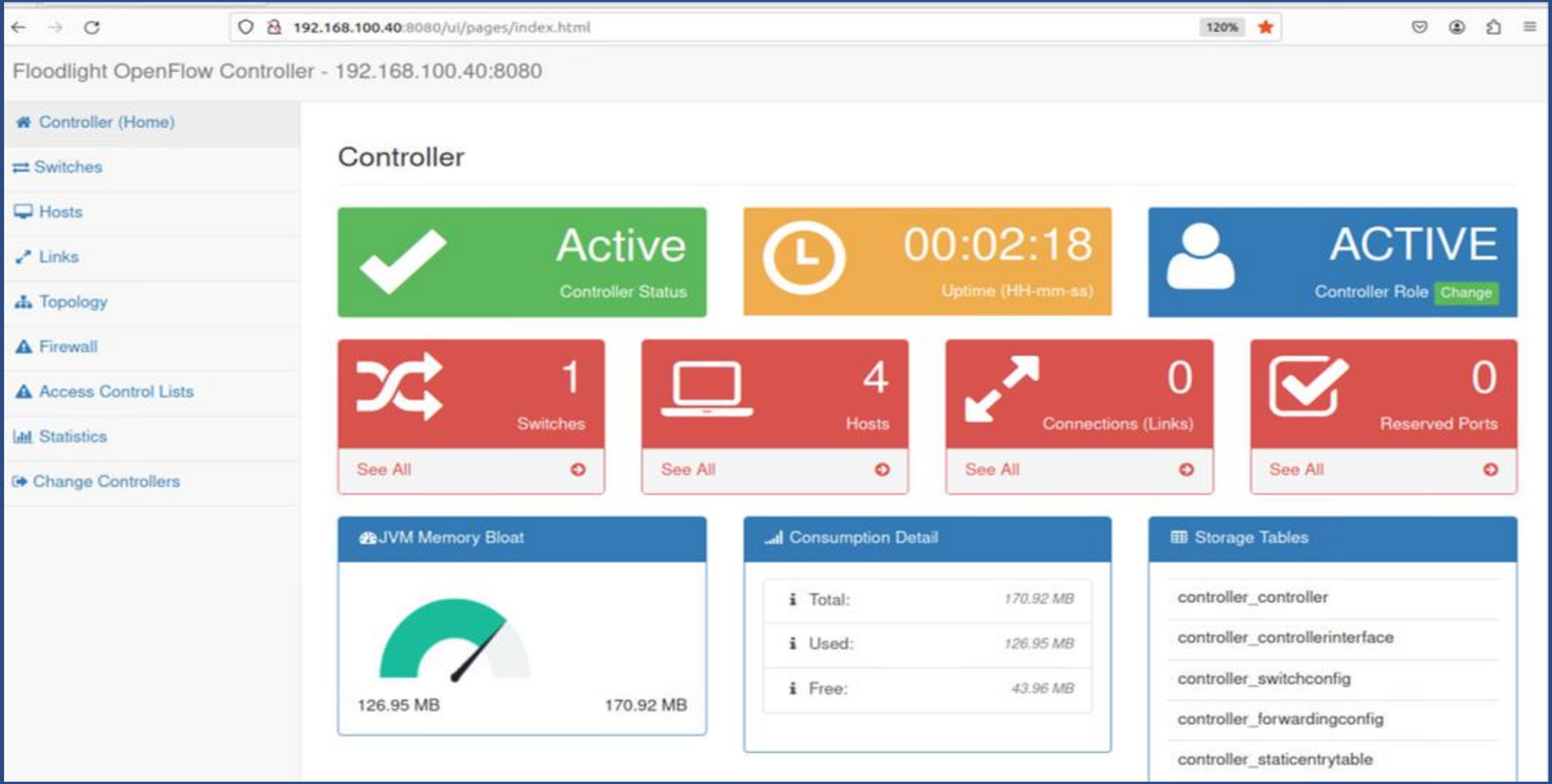
Dynamic Flow Rule Script

1. Monitor Alerts
2. Extract Malicious IPs
3. Push Flow Rule

Implementation

1. ModSecurity And Snort Alerts
2. Dynamic Threat Response
3. Centralized Alert Management
4. Automated Log Transmission

User-Friendly Interface



192.168.100.40:8080/ui/pages/index.html

Floodlight OpenFlow Controller - 192.168.100.40:8080

- Controller (Home)
- Switches
- Hosts
- Links
- Topology
- Firewall
- Access Control Lists
- Statistics
- Change Controllers

Controller

Active
Controller Status

00:02:18
Uptime (HH-mm-ss)

ACTIVE
Controller Role Change

1
Switches
See All

4
Hosts
See All

0
Connections (Links)
See All

0
Reserved Ports
See All

JVM Memory Bloat

126.95 MB 170.92 MB

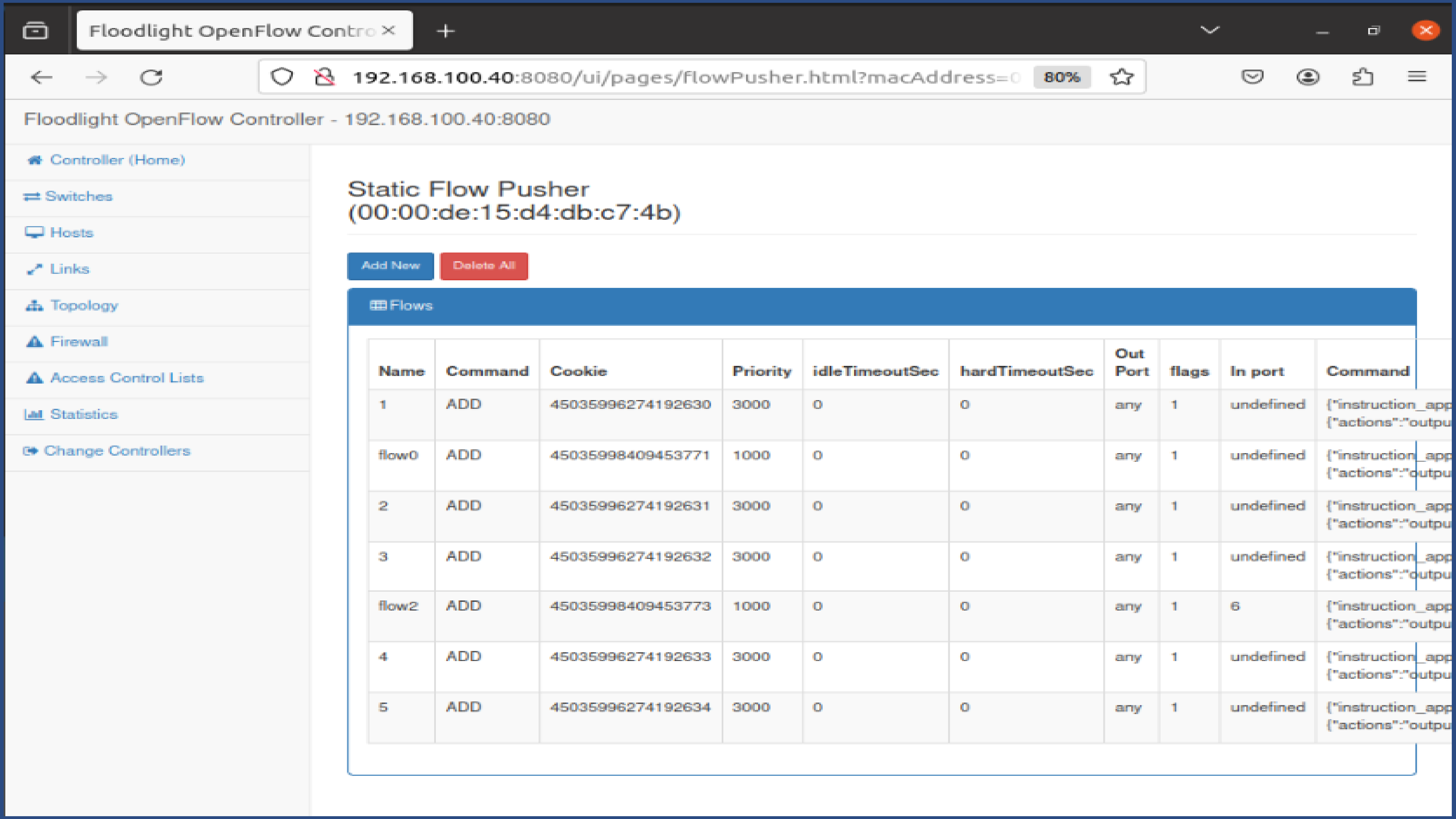
Consumption Detail

Total:	170.92 MB
Used:	126.95 MB
Free:	43.96 MB

Storage Tables

- controller_controller
- controller_controllerinterface
- controller_switchconfig
- controller_forwardingconfig
- controller_staticentrytable

Floodlight SDN Controller


Floodlight OpenFlow Contro
192.168.100.40:8080/ui/pages/flowPusher.html?macAddress=0
80%
Floodlight OpenFlow Controller - 192.168.100.40:8080
Controller (Home)
Switches
Hosts
Links
Topology
Firewall
Access Control Lists
Statistics
Change Controllers
Add New
Delete All
Flows

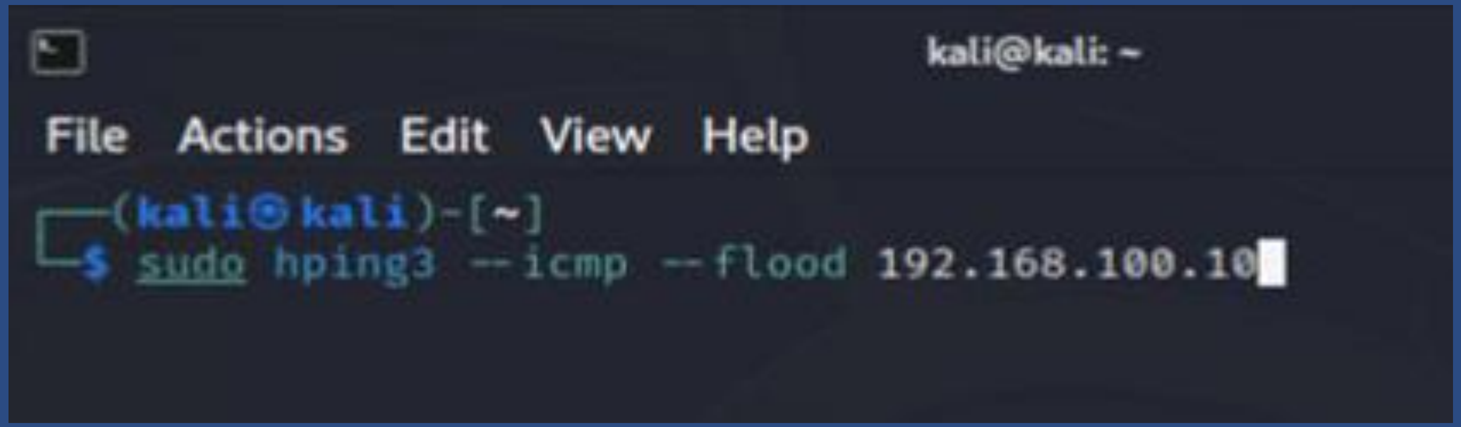
Static Flow Pusher
(00:00:de:15:d4:db:c7:4b)

Name	Command	Cookie	Priority	idleTimeoutSec	hardTimeoutSec	Out Port	flags	In port	Command
1	ADD	45035996274192630	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
flow0	ADD	45035998409453771	1000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
2	ADD	45035996274192631	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
3	ADD	45035996274192632	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
flow2	ADD	45035998409453773	1000	0	0	any	1	6	{"instruction_app {"actions":"outpu
4	ADD	45035996274192633	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
5	ADD	45035996274192634	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu

Test and Validation

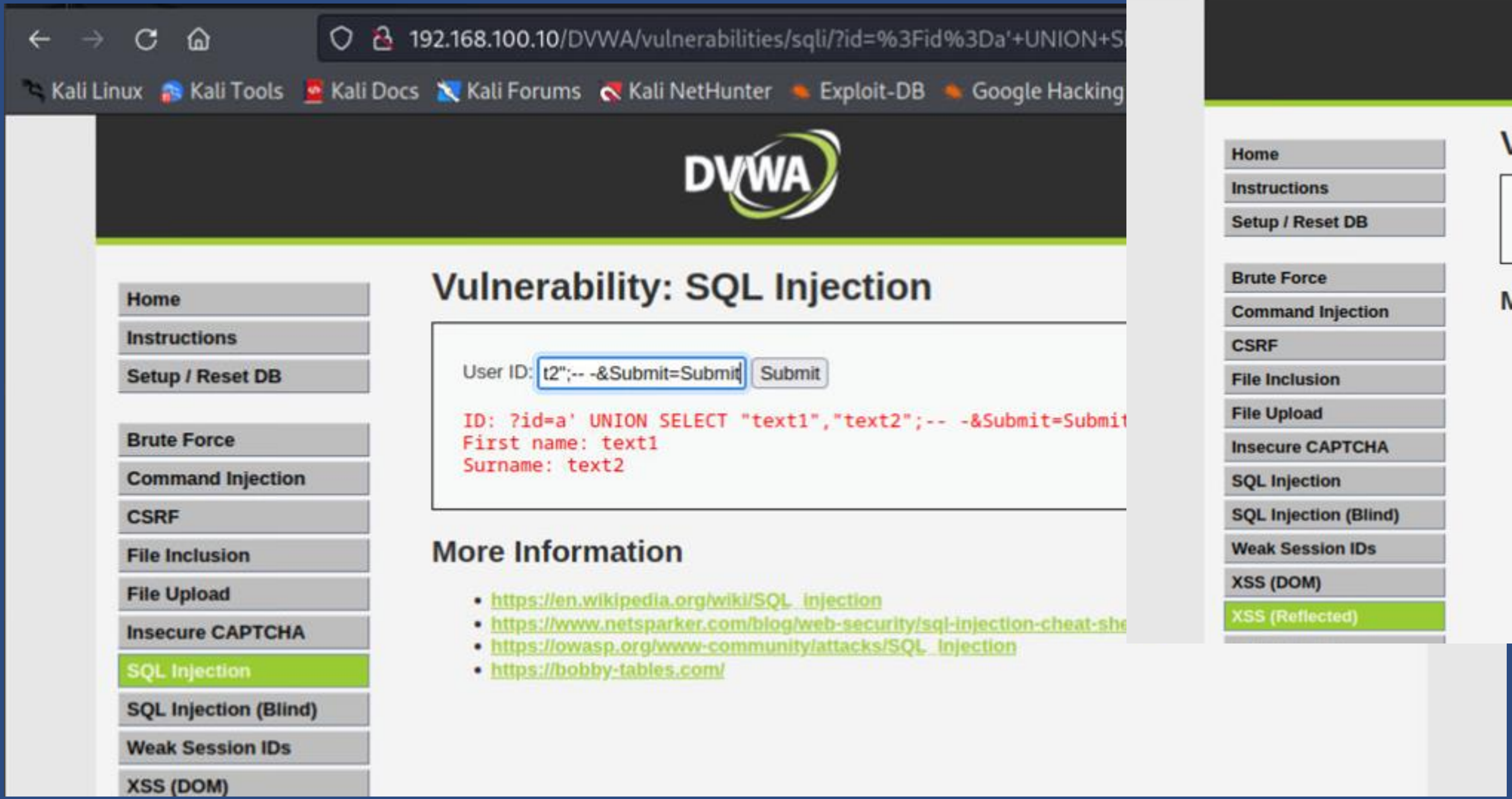
Simulated SSH attack

Simulated Dos attack



```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --icmp --flood 192.168.100.10
```

Web Application Attack



192.168.100.10/DVWA/vulnerabilities/sqli/?id=%3Fid%3Da'+UNION+S

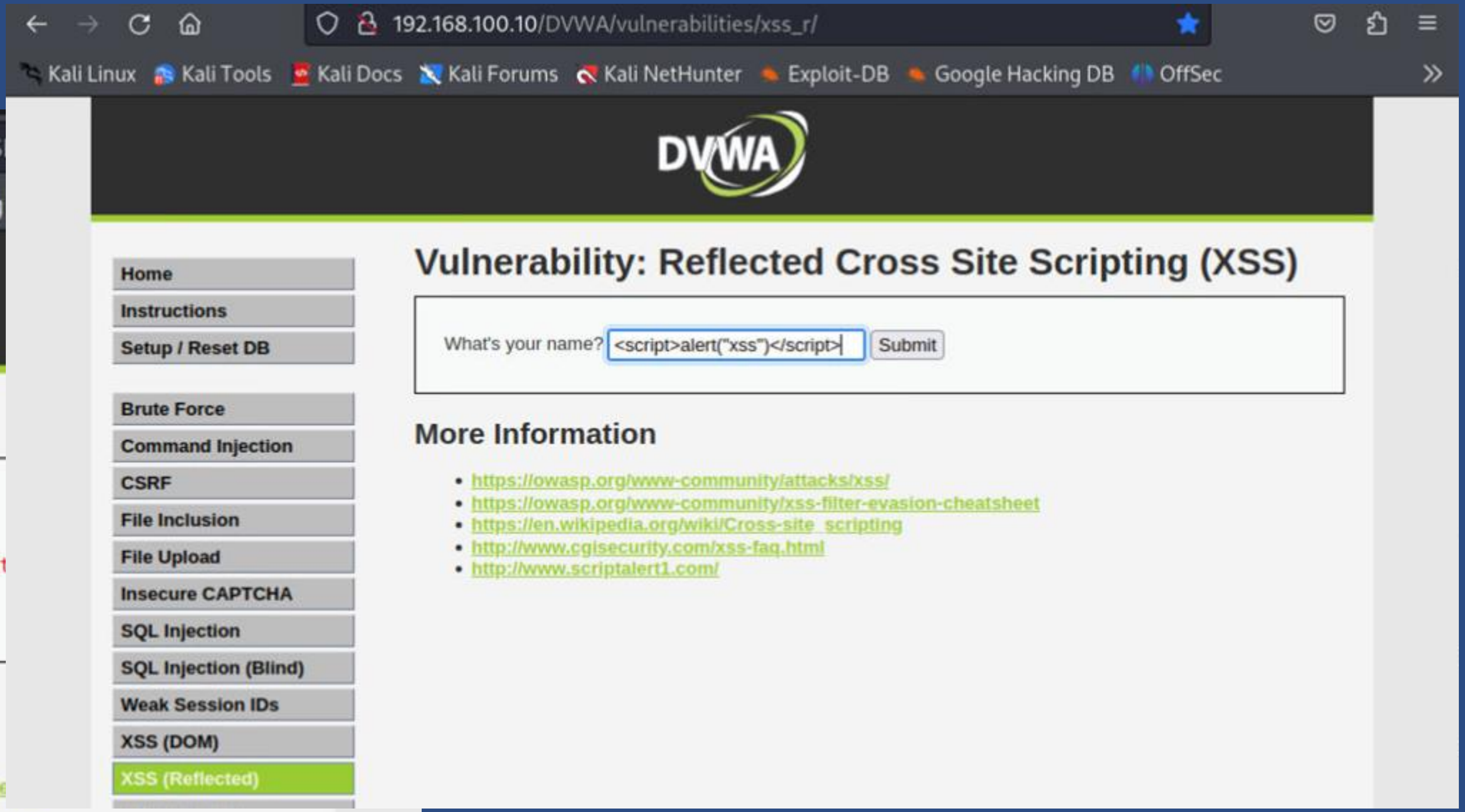
Vulnerability: SQL Injection

User ID: t2";-- -&Submit=Submit

```
ID: ?id=a' UNION SELECT "text1","text2";-- -&Submit=Submit
First name: text1
Surname: text2
```

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-she
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/



192.168.100.10/DVWA/vulnerabilities/xss_r/

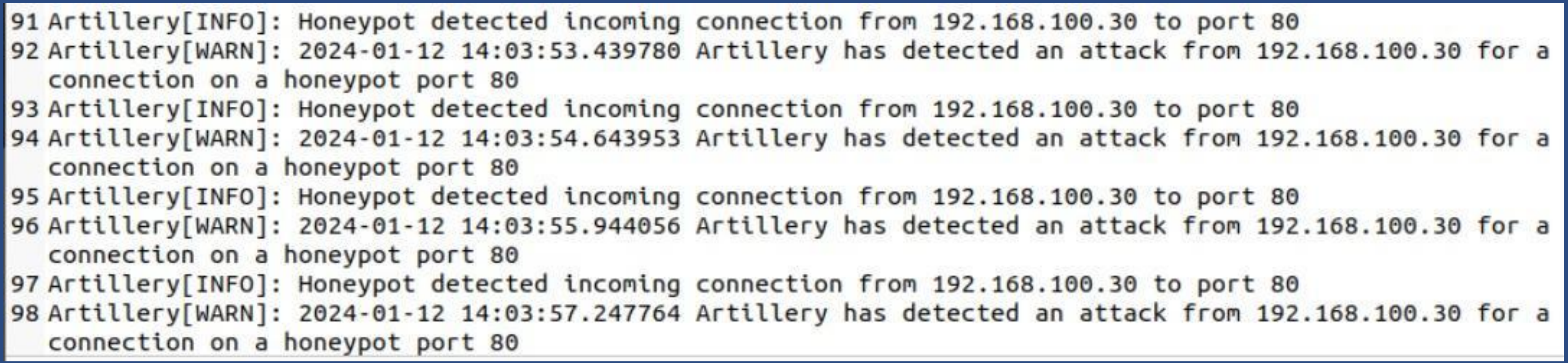
Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? <script>alert("xss")</script>

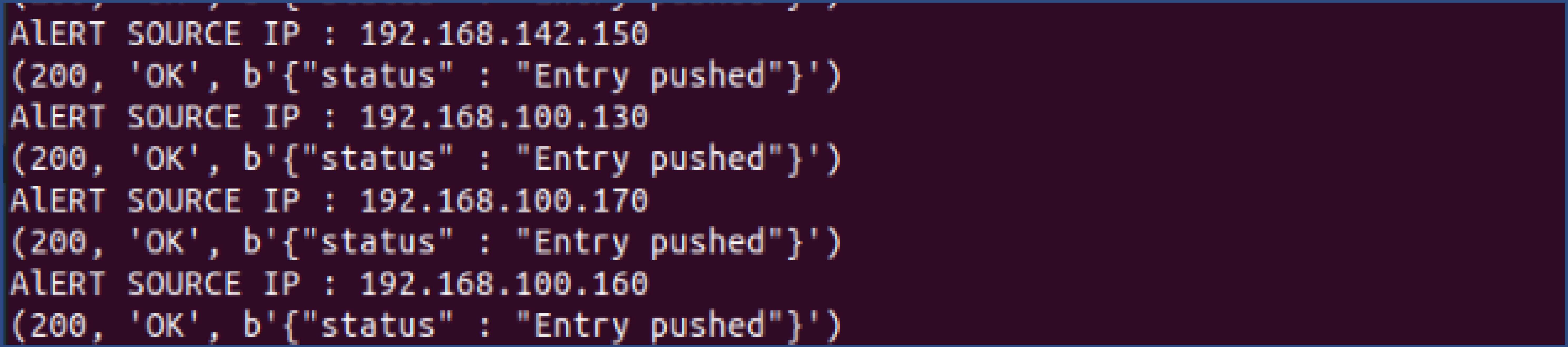
More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/xss-filter-evasion-cheatsheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Automated Response Mechanisms

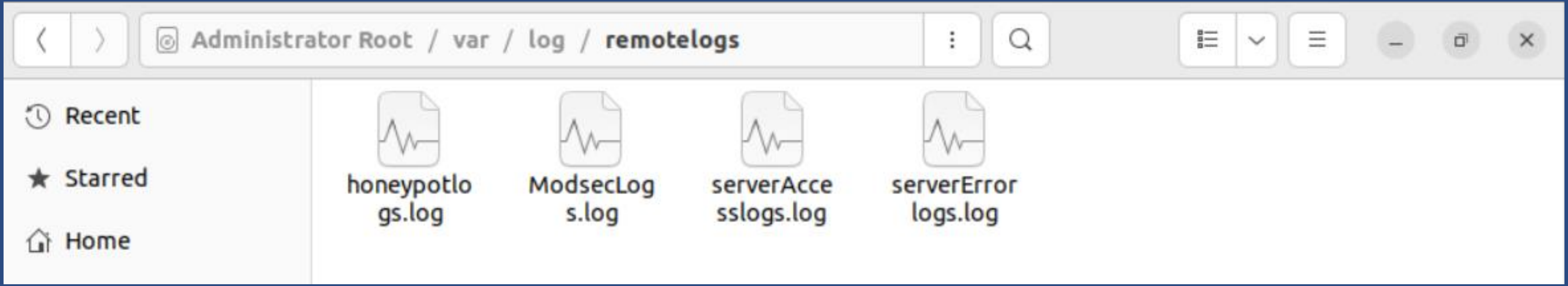


```
Artillery[INFO]: Honeypot detected incoming connection from 192.168.100.30 to port 80
Artillery[WARN]: 2024-01-12 14:03:53.439780 Artillery has detected an attack from 192.168.100.30 for a
connection on a honeypot port 80
Artillery[INFO]: Honeypot detected incoming connection from 192.168.100.30 to port 80
Artillery[WARN]: 2024-01-12 14:03:54.643953 Artillery has detected an attack from 192.168.100.30 for a
connection on a honeypot port 80
Artillery[INFO]: Honeypot detected incoming connection from 192.168.100.30 to port 80
Artillery[WARN]: 2024-01-12 14:03:55.944056 Artillery has detected an attack from 192.168.100.30 for a
connection on a honeypot port 80
Artillery[INFO]: Honeypot detected incoming connection from 192.168.100.30 to port 80
Artillery[WARN]: 2024-01-12 14:03:57.247764 Artillery has detected an attack from 192.168.100.30 for a
connection on a honeypot port 80
```



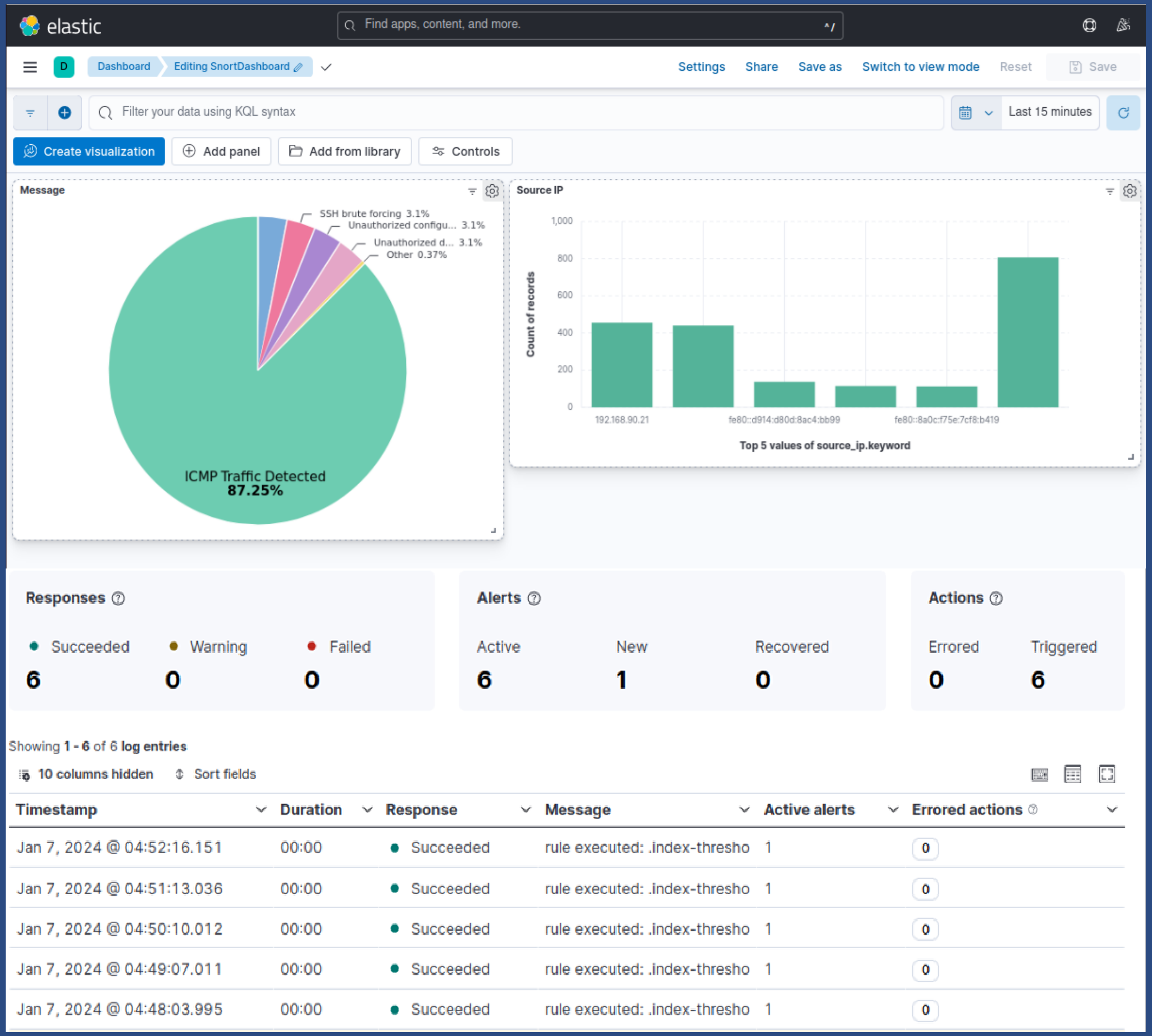
```
AlERT SOURCE IP : 192.168.142.150
(200, 'OK', b'{"status" : "Entry pushed"}')
AlERT SOURCE IP : 192.168.100.130
(200, 'OK', b'{"status" : "Entry pushed"}')
AlERT SOURCE IP : 192.168.100.170
(200, 'OK', b'{"status" : "Entry pushed"}')
AlERT SOURCE IP : 192.168.100.160
(200, 'OK', b'{"status" : "Entry pushed"}')
```

Centralized logging system



honeypotlogs.log
ModsecLogs.log
serverAccesslogs.log
serverErrorlogs.log

ELK Integration



Responses

● Succeeded	● Warning	● Failed
6	0	0

Alerts

Active	New	Recovered
6	1	0

Actions

Errored	Triggered
0	6

Showing **1 - 6** of 6 **log entries**

10 columns hidden   Sort fields

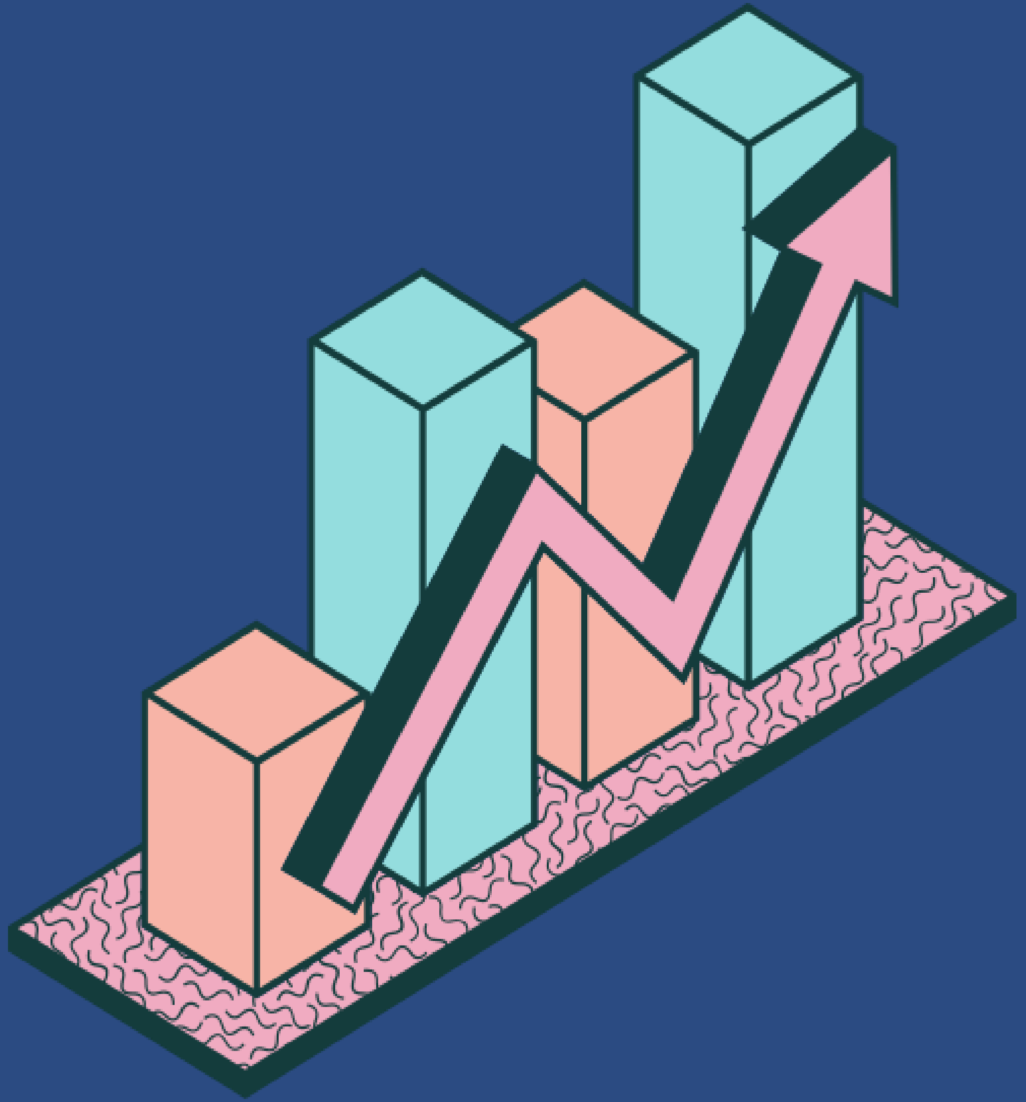
Timestamp	Duration	Response	Message	Active alerts	Errored actions
Jan 7, 2024 @ 04:52:16.151	00:00	● Succeeded	rule executed: .index-thresho	1	0
Jan 7, 2024 @ 04:51:13.036	00:00	● Succeeded	rule executed: .index-thresho	1	0
Jan 7, 2024 @ 04:50:10.012	00:00	● Succeeded	rule executed: .index-thresho	1	0
Jan 7, 2024 @ 04:49:07.011	00:00	● Succeeded	rule executed: .index-thresho	1	0
Jan 7, 2024 @ 04:48:03.995	00:00	● Succeeded	rule executed: .index-thresho	1	0

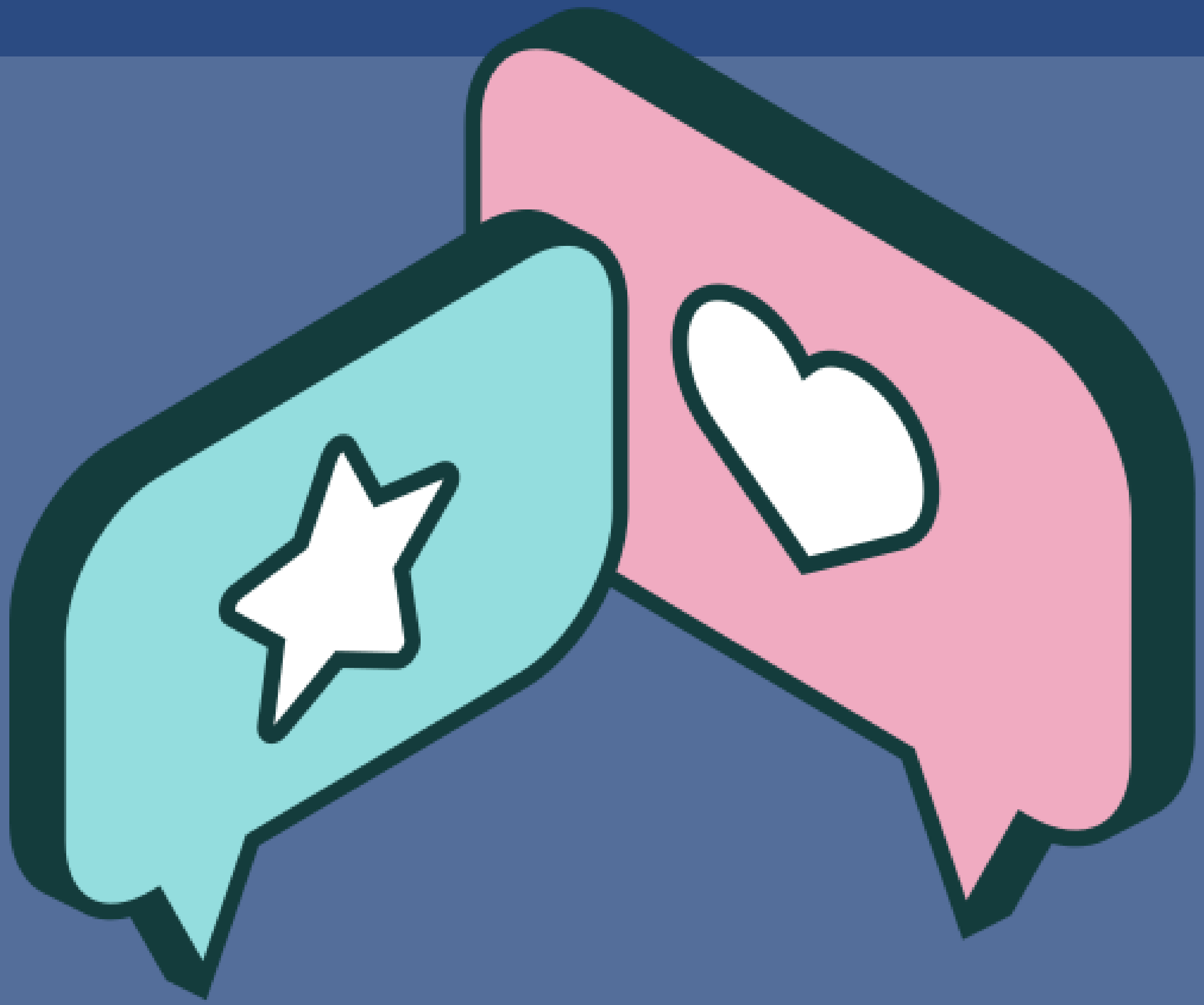
Conclusion

Innovative Deception Approach

Navigating a Technical Landscape

Meticulous Addressing of Requirements





Thank You!