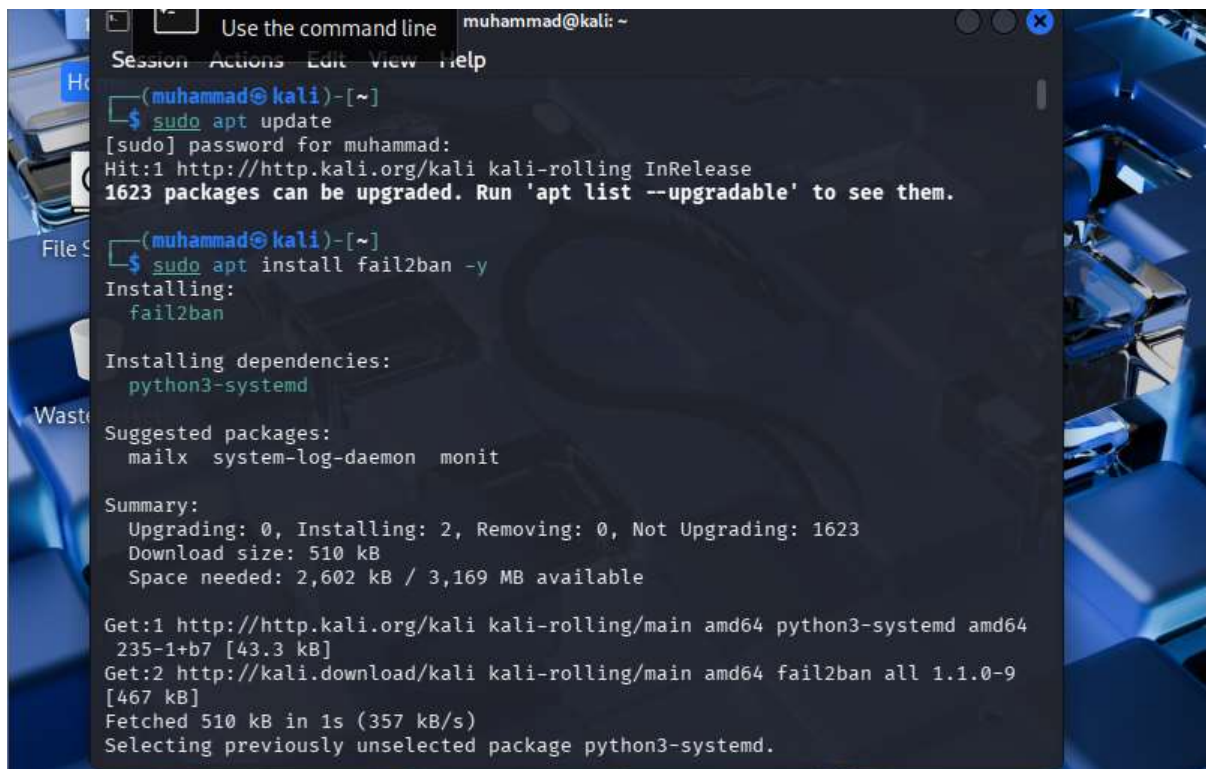


Name: Muhammad Bin Qasim

Week 4: Advanced Threat Detection & Web Security

Enhancements

Installing Security Tools: I working on a Kali Linux terminal. First, update the system's package list to ensure everything is current. Then, install a program called fail2ban. This is a security tool designed to protect computers from brute-force login attacks. The terminal displays the successful download and installation process.

A screenshot of a Kali Linux terminal window. The window title is "Use the command line" and the user is "muhammad@kali: ~". The terminal shows the following commands and output:

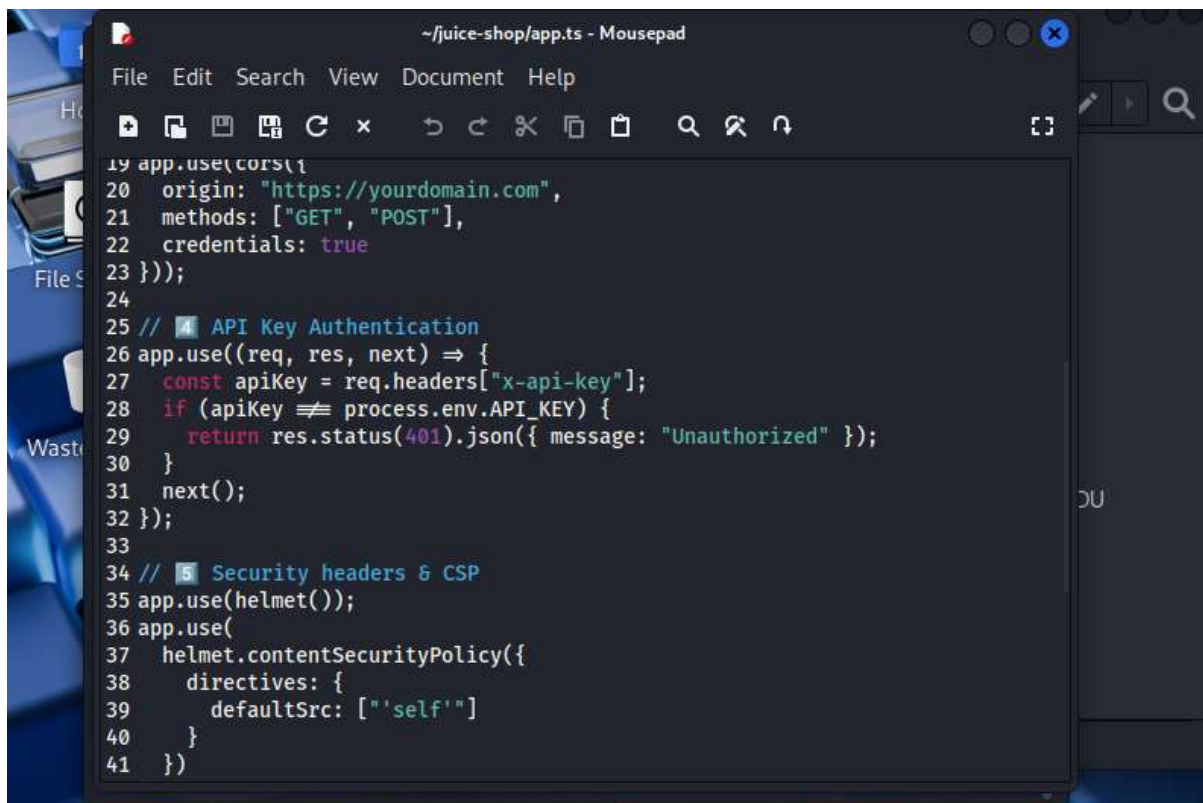
```
(muhammad@kali)-[~]  
$ sudo apt update  
[sudo] password for muhammad:  
Hit:1 http://http.kali.org/kali kali-rolling InRelease  
1623 packages can be upgraded. Run 'apt list --upgradable' to see them.  
  
(muhammad@kali)-[~]  
$ sudo apt install fail2ban -y  
Installing:  
fail2ban  
  
Installing dependencies:  
python3-systemd  
  
Suggested packages:  
mailx system-log-daemon monit  
  
Summary:  
Upgrading: 0, Installing: 2, Removing: 0, Not Upgrading: 1623  
Download size: 510 kB  
Space needed: 2,602 kB / 3,169 MB available  
  
Get:1 http://http.kali.org/kali kali-rolling/main amd64 python3-systemd amd64  
235-1+b7 [43.3 kB]  
Get:2 http://kali.download/kali kali-rolling/main amd64 fail2ban all 1.1.0-9  
[467 kB]  
Fetched 510 kB in 1s (357 kB/s)  
Selecting previously unselected package python3-systemd.
```

Configuring fail2ban: Turns on the fail2ban service and sets it to start automatically. Next, open its configuration file and check its current status to see if any malicious IP addresses have been blocked from accessing the system via SSH. At the bottom, they try to install a web package called express-rate-limit.

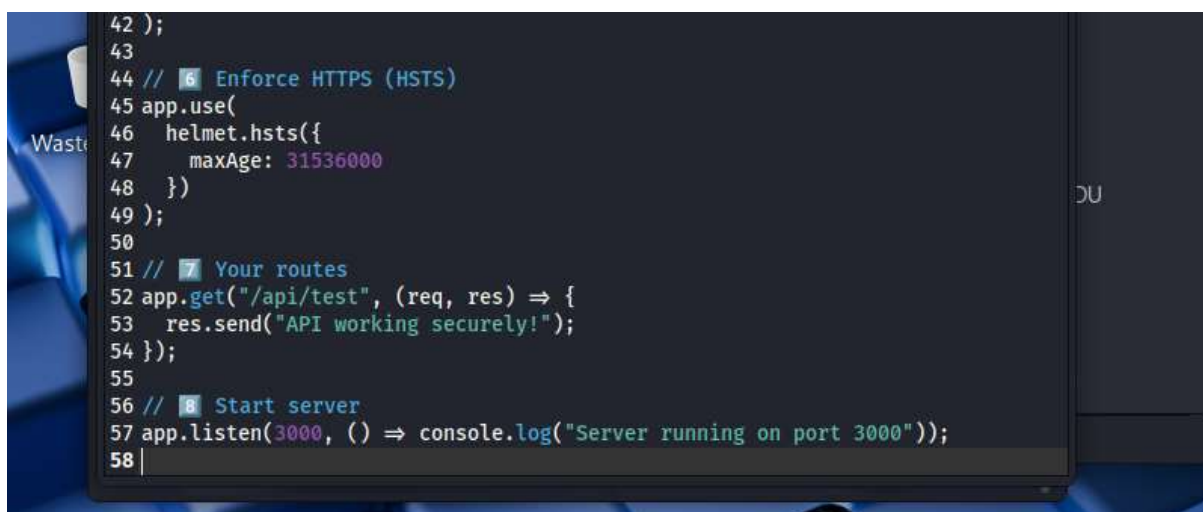
```
muhammad@kali: ~  
Session Actions Edit View Help  
(muhammad@kali)-[~]  
$ sudo systemctl enable fail2ban  
Synchronizing state of fail2ban.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable fail2ban  
Created symlink '/etc/systemd/system/multi-user.target.wants/fail2ban.service' -> '/usr/lib/systemd/system/fail2ban.service'.  
(muhammad@kali)-[~]  
$ sudo systemctl start fail2ban  
(muhammad@kali)-[~]  
$ sudo nano /etc/fail2ban/jail.local  
(muhammad@kali)-[~]  
$ sudo fail2ban-client status sshd  
Status for the jail: sshd  
- Filter  
| - Currently failed: 0  
| - Total failed: 0  
| - Journal matches: _SYSTEMD_UNIT=ssh.service + _COMM=sshd  
- Actions  
| - Currently banned: 0  
| - Total banned: 0  
| - Banned IP list:  
(muhammad@kali)-[~]  
$ npm install express-rate-limit
```

Installing Node Package Manager: Because the previous command failed, the system tells the user that npm (Node Package Manager) is not installed. Then types a command to install npm using the system's package manager. The screen shows a list of the required files being downloaded and set up. This tool is necessary for building JavaScript applications.

```
muhammad@kali: ~  
Session Actions Edit View Help  
- Banned IP list:  
(muhammad@kali)-[~]  
$ npm install express-rate-limit  
Command 'npm' not found, but can be installed with:  
sudo apt install npm  
Do you want to install it? (N/y)y  
sudo apt install npm  
Upgrading:  
libssl3t64 nodejs  
Installing:  
npm  
Installing dependencies:  
eslint  
gyp  
handlebars  
libjs-events  
libjs-inherits  
libjs-is-typedarray  
libjs-prettify  
libjs-regenerate  
libjs-source-map  
libjs-sprintf-js  
libjs-typedarray-to-buffer  
libjs-util  
libnode-dev
```

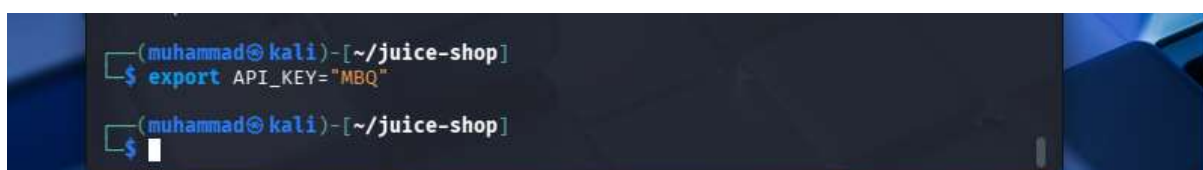



```
~juice-shop/app.ts - Mousepad
File Edit Search View Document Help
19 app.use(cors({
20   origin: "https://yourdomain.com",
21   methods: ["GET", "POST"],
22   credentials: true
23 }));
24
25 // 4 API Key Authentication
26 app.use((req, res, next) => {
27   const apiKey = req.headers["x-api-key"];
28   if (apiKey !== process.env.API_KEY) {
29     return res.status(401).json({ message: "Unauthorized" });
30   }
31   next();
32 });
33
34 // 5 Security headers & CSP
35 app.use(helmet());
36 app.use(
37   helmet.contentSecurityPolicy({
38     directives: {
39       defaultSrc: ["'self'"]
40     }
41   })
42 );
```



```
42 );
43
44 // 6 Enforce HTTPS (HSTS)
45 app.use(
46   helmet.hsts({
47     maxAge: 31536000
48   })
49 );
50
51 // 7 Your routes
52 app.get("/api/test", (req, res) => {
53   res.send("API working securely!");
54 });
55
56 // 8 Start server
57 app.listen(3000, () => console.log("Server running on port 3000"));
58 |
```

Setting an Environment Variable In this last screenshot, goes back to the terminal inside the juice-shop folder. Type an export command to create a temporary system variable named API_KEY and set its value to "MBQ". This secret key is exactly what the code in Image 6 will be looking for to allow access to the web app.



```
(muhammad@kali) - [~/juice-shop]
$ export API_KEY="MBQ"
(muhammad@kali) - [~/juice-shop]
$
```

Github Link:

<https://github.com/MBQ17/Week-4-Advanced-Threat-Detection-Web-Security-Enhancements>
