Week # 01 Security Assessment Report

Prepared by: Shaheer Ahmad

Intern ID: DHC-6

Date: February 18, 2026

Internship Program: Developers Hub Cybersecurity Interns

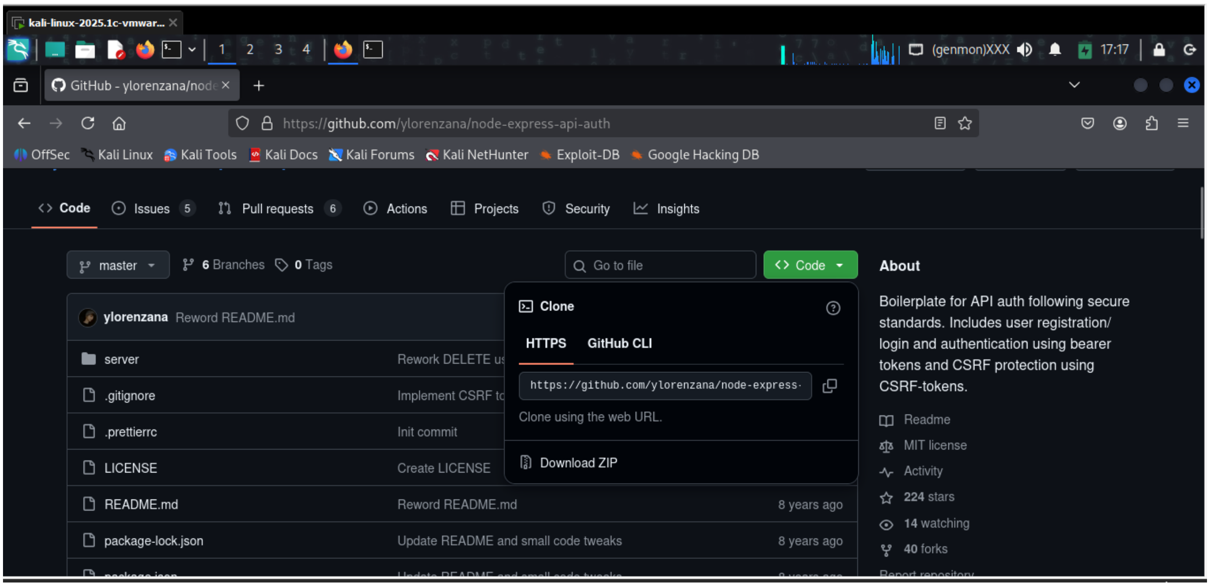
Table of Contents

1. Introduction

The first week focused on setting up the environment, exploring a sample application, and performing a basic vulnerability assessment. The objective was to identify common weaknesses in web applications and document findings for improvement.

2. Application Setup

- Cloned the **node-express-api-auth** repository from GitHub into the Kali Linux environment.
- Installed dependencies using `npm install` and launched the application with `npm start`.
- Accessed the application locally at **http://localhost:3000**, exploring the signup, login, and profile pages.
- This setup provided a controlled environment for testing authentication flows and security mechanisms.


kali-linux-2025.1c-vmwar...
GitHub - ylorenzana/node
https://github.com/ylorenzana/node-express-api-auth
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
Code Issues 5 Pull requests 6 Actions Projects Security Insights
master 6 Branches 0 Tags
Go to file
Code
Clone
HTTPS GitHub CLI
Download ZIP
17:17

https://github.com/ylorenzana/node-express

Clone using the web URL.

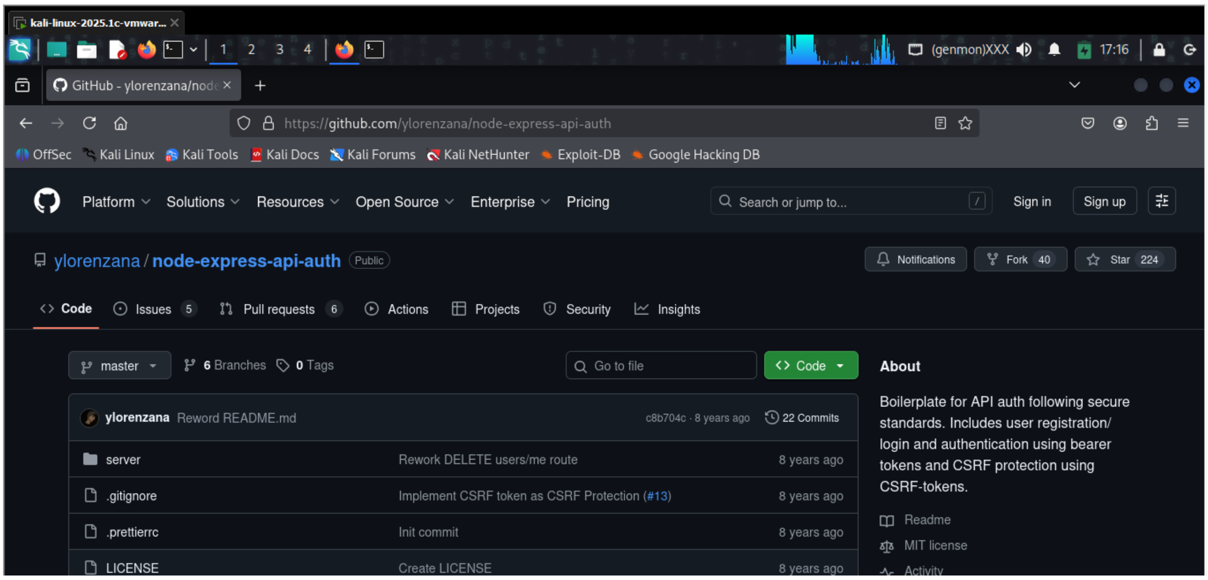
ylorenzana Reword README.md

server	Rework DELETE us	
.gitignore	Implement CSRF to	
.prettierrc	Init commit	
LICENSE	Create LICENSE	
README.md	Reword README.md	8 years ago
package-lock.json	Update README and small code tweaks	8 years ago

About

Boilerplate for API auth following secure standards. Includes user registration/ login and authentication using bearer tokens and CSRF protection using CSRF-tokens.

Readme
MIT license
Activity
224 stars
14 watching
40 forks


kali-linux-2025.1c-vmwar...
GitHub - ylorenzana/node
https://github.com/ylorenzana/node-express-api-auth
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
Platform Solutions Resources Open Source Enterprise Pricing
Search or jump to...
Sign in Sign up
Notifications Fork 40 Star 224
Code Issues 5 Pull requests 6 Actions Projects Security Insights
master 6 Branches 0 Tags
Go to file
Code
17:16

ylorenzana / node-express-api-auth Public

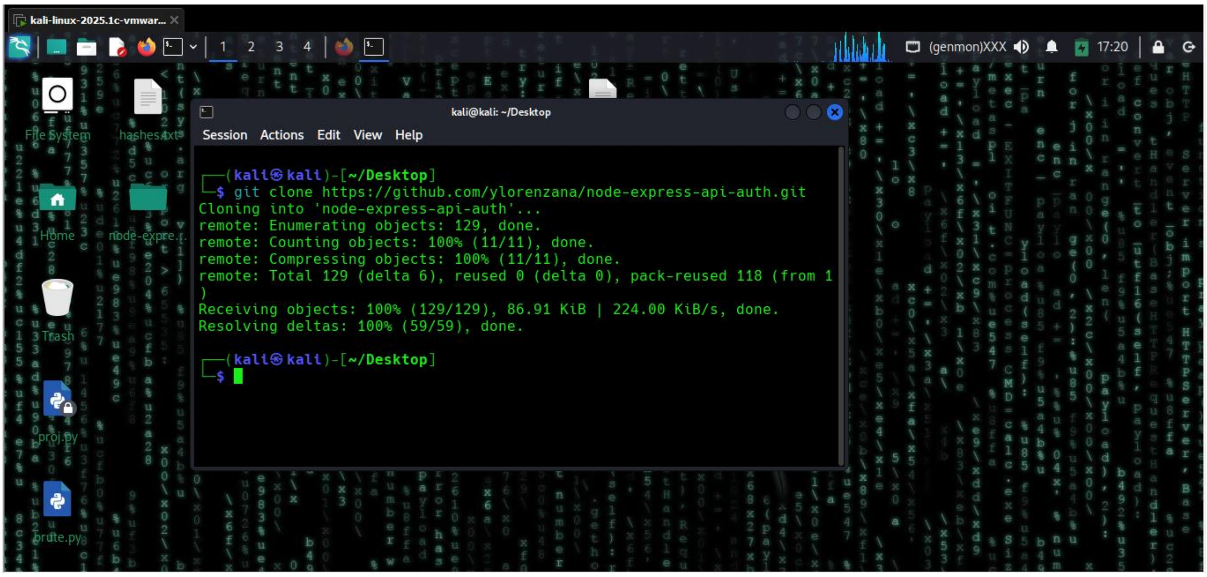
ylorenzana Reword README.md c8b704c · 8 years ago 22 Commits

server	Rework DELETE users/me route	8 years ago
.gitignore	Implement CSRF token as CSRF Protection (#13)	8 years ago
.prettierrc	Init commit	8 years ago
LICENSE	Create LICENSE	8 years ago

About

Boilerplate for API auth following secure standards. Includes user registration/ login and authentication using bearer tokens and CSRF protection using CSRF-tokens.

Readme
MIT license
Activity

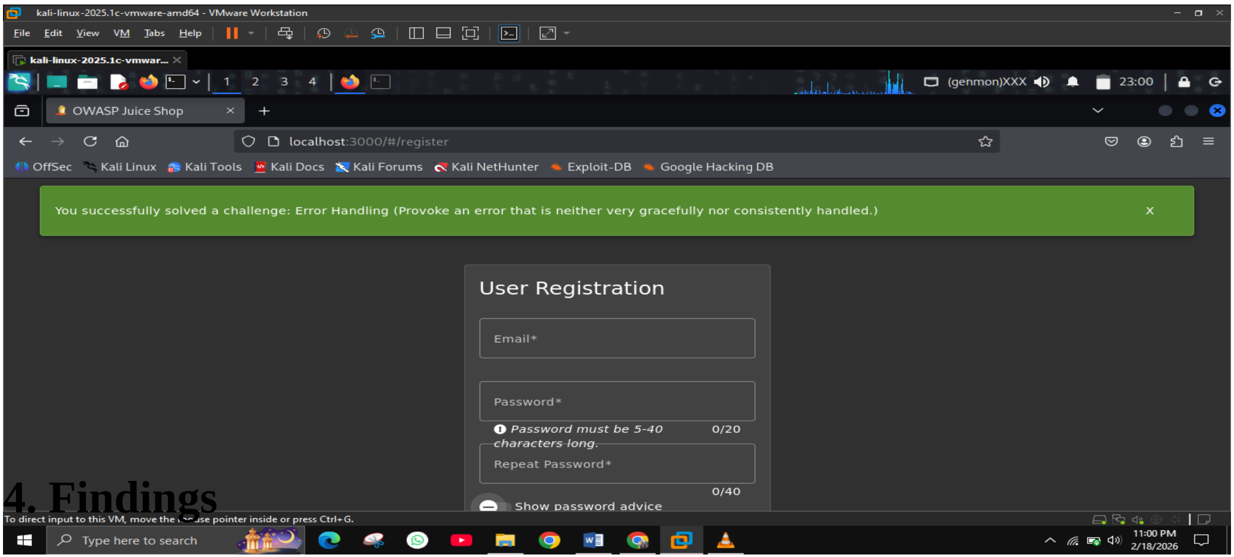


```
┌──(kali㉿kali)-[~/Desktop]
└─$ git clone https://github.com/ylorenzana/node-express-api-auth.git
Cloning into 'node-express-api-auth'...
remote: Enumerating objects: 129, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 129 (delta 6), reused 0 (delta 0), pack-reused 118 (from 1
)
Receiving objects: 100% (129/129), 86.91 KiB | 224.00 KiB/s, done.
Resolving deltas: 100% (59/59), done.

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

3. Vulnerability Assessment

The following tools and techniques were used to identify vulnerabilities:

- **OWASP ZAP**: Automated scanning revealed potential misconfigurations and insecure endpoints.
- **Browser Developer Tools**: Injected a simple XSS payload (<script>alert('XSS')</script>) into login and signup fields.
  - Result: The application displayed improper error handling ([object Object]), confirming weak input validation.
- **SQL Injection Test**: Attempted login with admin' OR '1'='1.
  - Observation: The application did not block this input, suggesting possible SQL injection risk.



kali-linux-2025.1c-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2025.1c-vmwar...
(genmon)XXX 23:00
OWASP Juice Shop
localhost:3000/#/register
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
Type here to search
11:00 PM 2/18/2026

You successfully solved a challenge: Error Handling (Provoke an error that is neither very gracefully nor consistently handled.)

User Registration

Email*

Password*

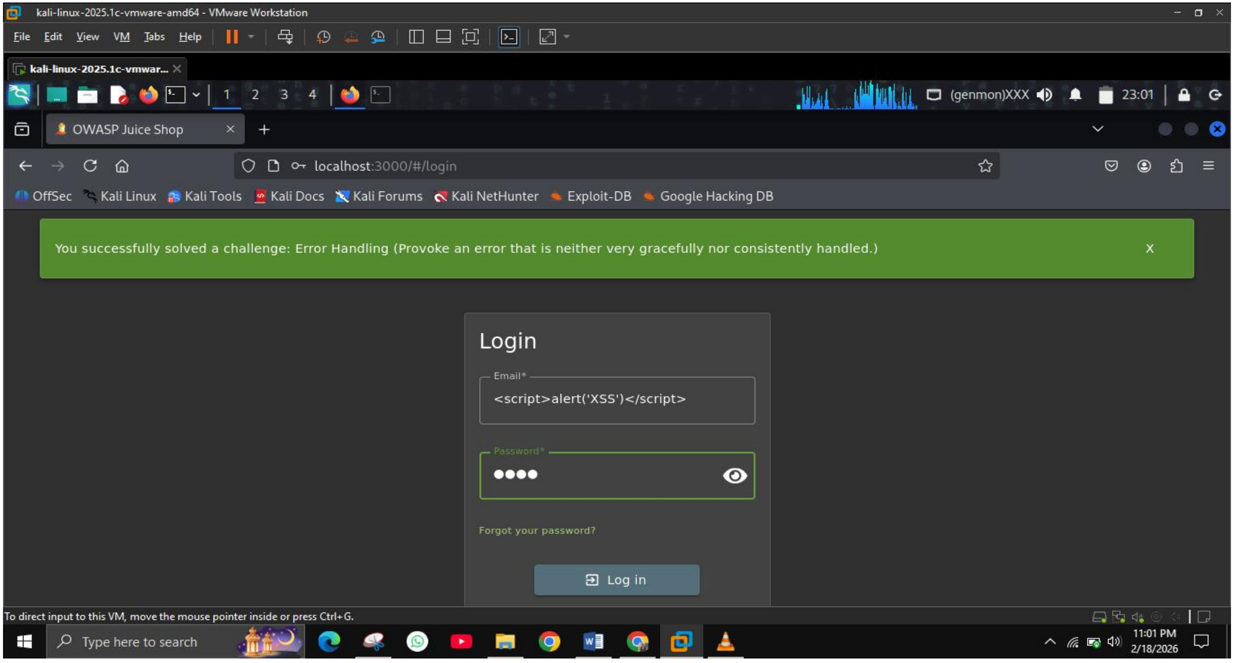
Password must be 5-40 characters long. 0/20

Repeat Password* 0/40

Show password advice

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

4. Findings



kali-linux-2025.1c-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2025.1c-vmwar...
(genmon)XXX 23:01
OWASP Juice Shop
localhost:3000/#/login
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
Forgot your password?
Log in
Type here to search
11:01 PM 2/18/2026

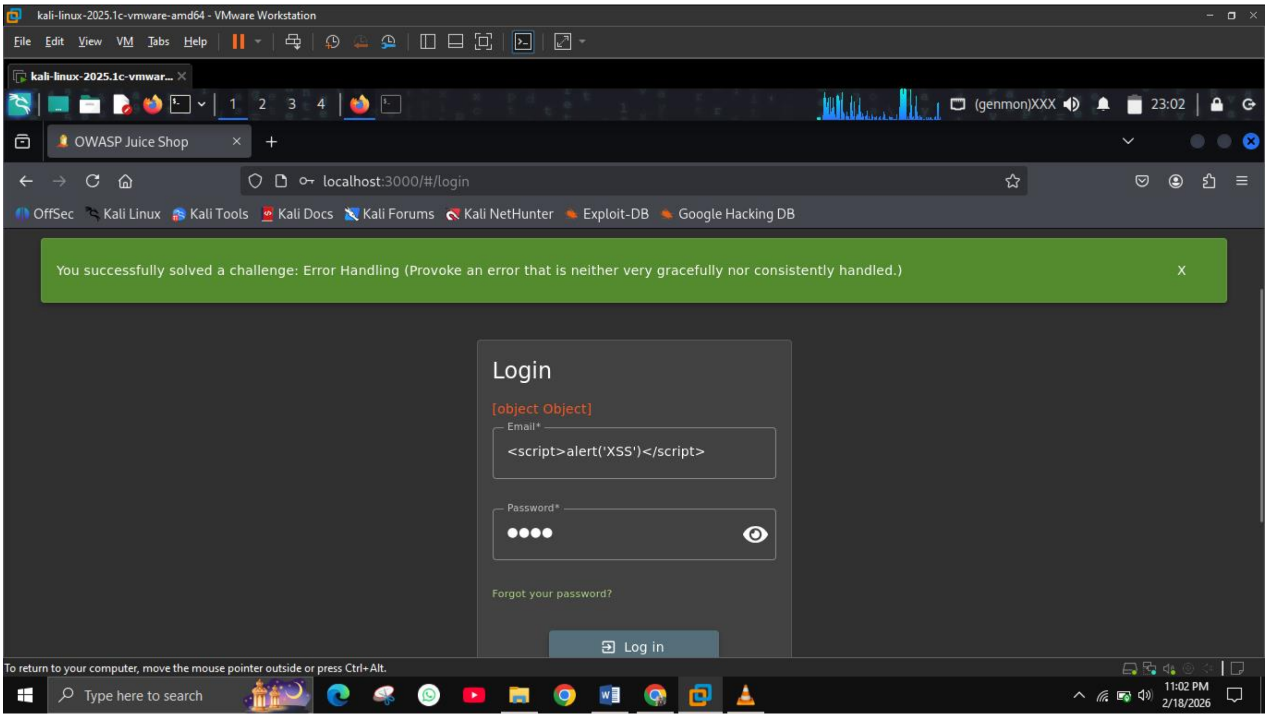
You successfully solved a challenge: Error Handling (Provoke an error that is neither very gracefully nor consistently handled.)

Login

Email*: <script>alert('XSS')</script>

Password*: ●●●●

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


kali-linux-2025.1c-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2025.1c-vmwar...
(genmon)XXX 23:02
OWASP Juice Shop
localhost:3000/#/login
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
Type here to search
11:02 PM
2/18/2026

You successfully solved a challenge: Error Handling (Provoke an error that is neither very gracefully nor consistently handled.) X

Login

[object Object]

Email* <script>alert('XSS')</script>

Password* ●●●●

Forgot your password?

Log in

- **Cross-Site Scripting (XSS)**: The application accepted script injections without sanitization.
- **Error Handling Issues**: Errors were exposed in raw form, making debugging information visible to attackers.
- **Potential SQL Injection**: Lack of input validation on login fields.
- **Weak Password Policy**: Password requirements were minimal (5–40 characters), with no enforced complexity.

5. Areas of Improvement

- Implement **input validation and sanitization** for all user inputs.
- Strengthen **password storage** by hashing with bcrypt.
- Improve **error handling** to avoid exposing raw system messages.
- Add **basic authentication hardening** (e.g., token-based sessions, CSRF protection).

6. Conclusion

Week 1 successfully established the testing environment and uncovered several vulnerabilities in the sample application. These findings set the stage for Week 2, where the focus will shift to fixing vulnerabilities using libraries such as validator, bcrypt, and jsonwebtoken, and enhancing overall application security.

Week # 02 Security Measures Implementation Report

Introduction

After identifying vulnerabilities in Week 1, the second week focused on applying security measures to strengthen the application. The goal was to fix issues such as XSS, SQL injection, weak password storage, and poor error handling, while also enhancing authentication and data transmission security.

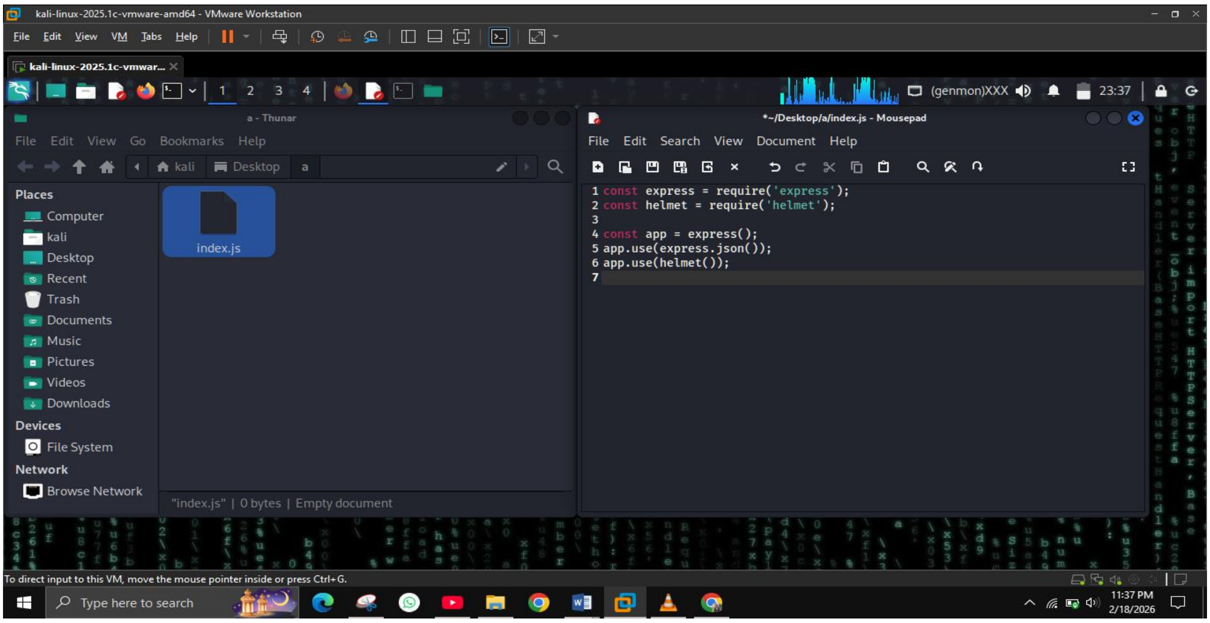
Fixing Vulnerabilities

1. Input Validation and Sanitization

- Installed the **validator** library (npm install validator).
- Applied validation checks in route handlers to ensure only properly formatted inputs are accepted.
- Example:

```
const validator = require('validator');

if (!validator.isEmail(email)) {

  return res.status(400).send('Invalid email');

}
```

- This prevents malicious scripts or malformed data from being processed.



kali-linux-2025.1c-vmware-amd64 - VMware Workstation
a - Thunar
index.js
"index.js" | 0 bytes | Empty document
*~/Desktop/a/index.js - Mousepad
23:37
11:37 PM
2/18/2026

```
const express = require('express');
const helmet = require('helmet');

const app = express();
app.use(express.json());
app.use(helmet());
```

2. Password Hashing

- Installed **bcrypt** (npm install bcrypt).
- Implemented password hashing before storing credentials in the database.
- Example:

```
const bcrypt = require('bcrypt');

const hashedPassword = await bcrypt.hash(password, 10);
```

- This ensures that even if the database is compromised, raw passwords are not exposed.

3. Enhanced Authentication

- Added **token-based authentication** using **jsonwebtoken** (npm install jsonwebtoken).
- Example:

```
const jwt = require('jsonwebtoken');

const token = jwt.sign({ id: user._id }, 'your-secret-key');

res.send({ token });
```

- This provides secure session handling and prevents unauthorized access.

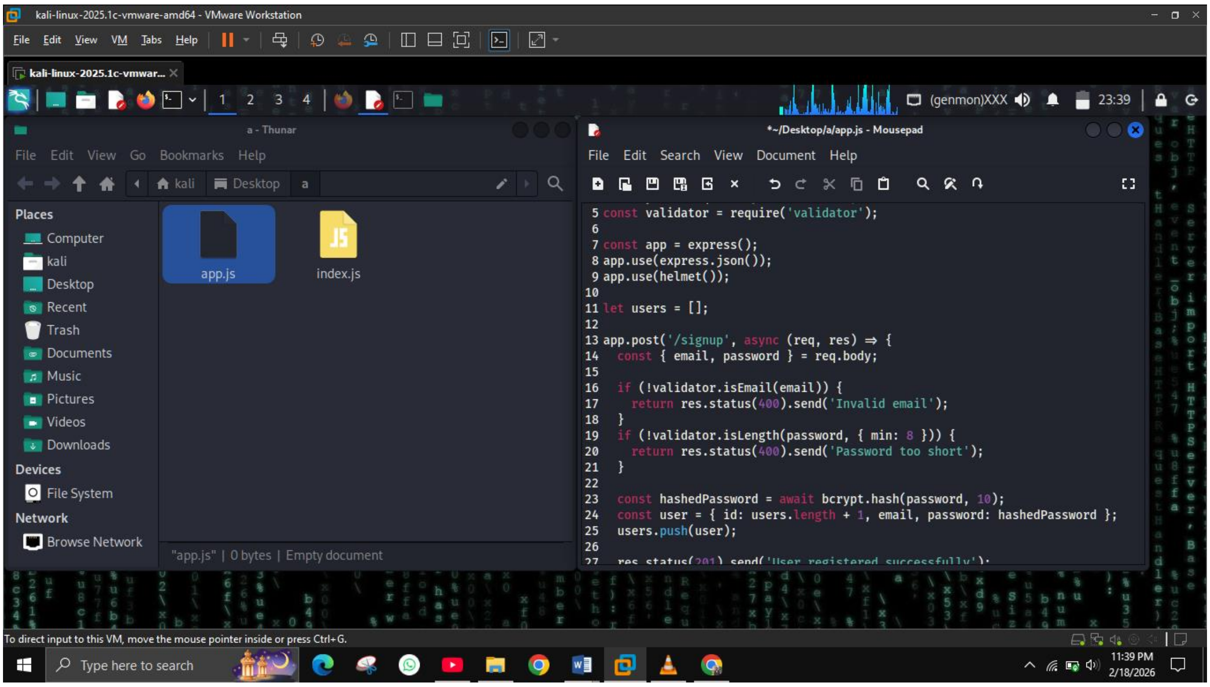
4. Secure Data Transmission

- Installed and configured **Helmet.js** (npm install helmet).
- Applied middleware to secure HTTP headers:

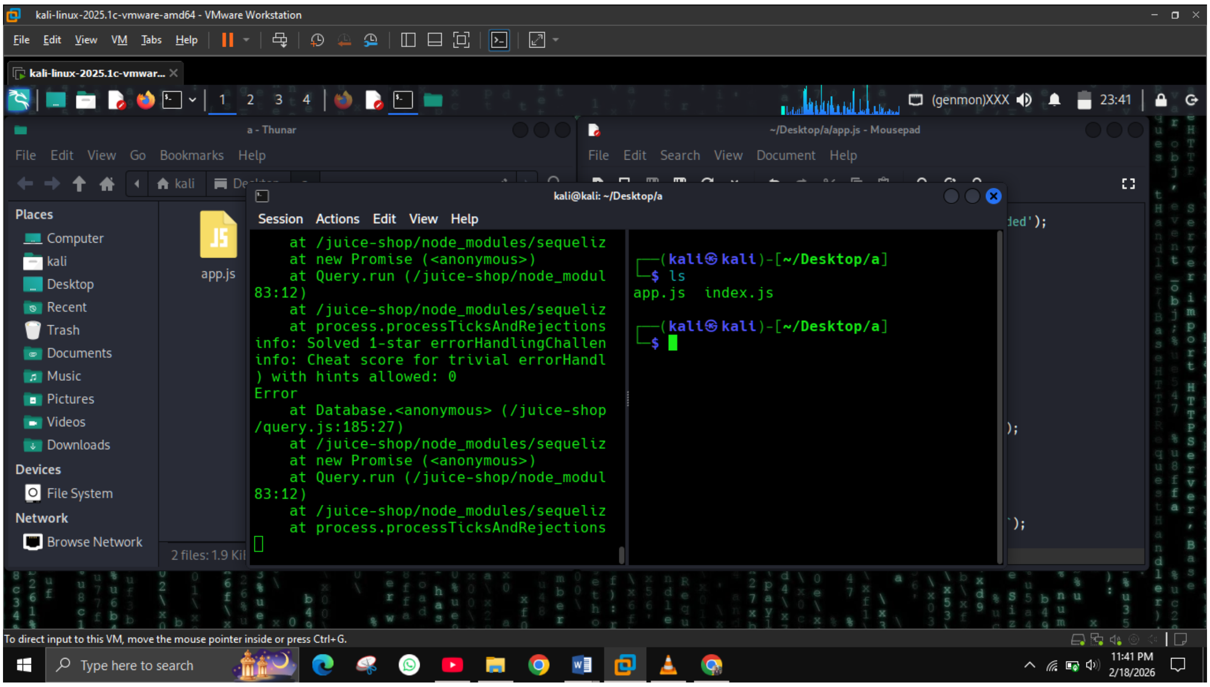
const helmet = require('helmet');

app.use(helmet());

- This reduces risks from common attacks like clickjacking and MIME-type sniffing.


kali-linux-2025.1c-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2025.1c-vmwar...
(genmon)XXX
23:39
a - Thunar
File Edit View Go Bookmarks Help
kali
Desktop
a
Places
Computer
kali
Desktop
Recent
Trash
Documents
Music
Pictures
Videos
Downloads
Devices
File System
Network
Browse Network
app.js
index.js
"app.js" | 0 bytes | Empty document
*~/Desktop/a/app.js - Mousepad
File Edit Search View Document Help
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Type here to search
11:39 PM
2/18/2026

```
const validator = require('validator');

const app = express();
app.use(express.json());
app.use(helmet());

let users = [];

app.post('/signup', async (req, res) => {
  const { email, password } = req.body;

  if (!validator.isEmail(email)) {
    return res.status(400).send('Invalid email');
  }
  if (!validator.isLength(password, { min: 8 })) {
    return res.status(400).send('Password too short');
  }

  const hashedPassword = await bcrypt.hash(password, 10);
  const user = { id: users.length + 1, email, password: hashedPassword };
  users.push(user);

  res.status(201).send('User registered successfully');
```



```
    at /juice-shop/node_modules/sequeliz
    at new Promise (<anonymous>)
    at Query.run (/juice-shop/node_modul
83:12)
    at /juice-shop/node_modules/sequeliz
    at process.processTicksAndRejections
info: Solved 1-star errorHandlingChallen
info: Cheat score for trivial errorHandl
) with hints allowed: 0
Error
    at Database.<anonymous> (/juice-shop
/query.js:185:27)
    at /juice-shop/node_modules/sequeliz
    at new Promise (<anonymous>)
    at Query.run (/juice-shop/node_modul
83:12)
    at /juice-shop/node_modules/sequeliz
    at process.processTicksAndRejections
```

```
┌──(kali㉿kali)-[~/Desktop/a]
└─$ ls
app.js  index.js

┌──(kali㉿kali)-[~/Desktop/a]
└─$ 
```

Improvements Achieved

- **XSS Mitigation:** Inputs are now validated and sanitized.
- **Password Security:** Passwords are hashed with bcrypt, making them resistant to brute-force attacks.
- **Authentication Hardening:** JWT tokens ensure secure and scalable authentication.
- **Safer Data Transmission:** Helmet adds protective headers, reducing exposure to common web threats.

Conclusion

Week 2 successfully addressed the vulnerabilities identified in Week 1. By implementing input validation, password hashing, token-based authentication, and secure headers, the application is now significantly more resilient against common attacks. These measures lay the foundation for advanced penetration testing and logging in Week 3.

Week # 03 Advanced Security and Final Reporting

Introduction

Week 3 focused on advanced security practices, including penetration testing, logging, and documenting best practices. The objective was to validate the effectiveness of previous fixes, establish monitoring mechanisms, and prepare a checklist for ongoing security improvements.

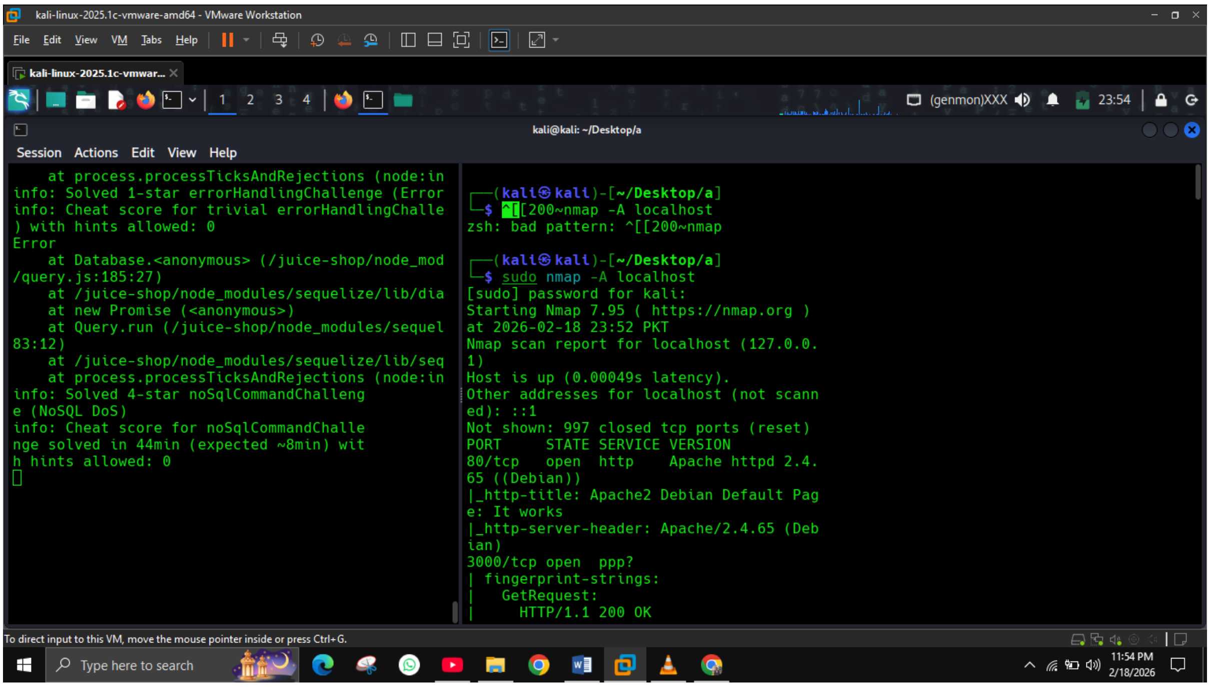
1. Basic Penetration Testing

Tools Used

- **Nmap:** Scanned localhost to identify open ports and services.
- **Browser-based Testing:** Attempted XSS, SQL injection, and weak password inputs in Juice Shop and the custom Node.js app.

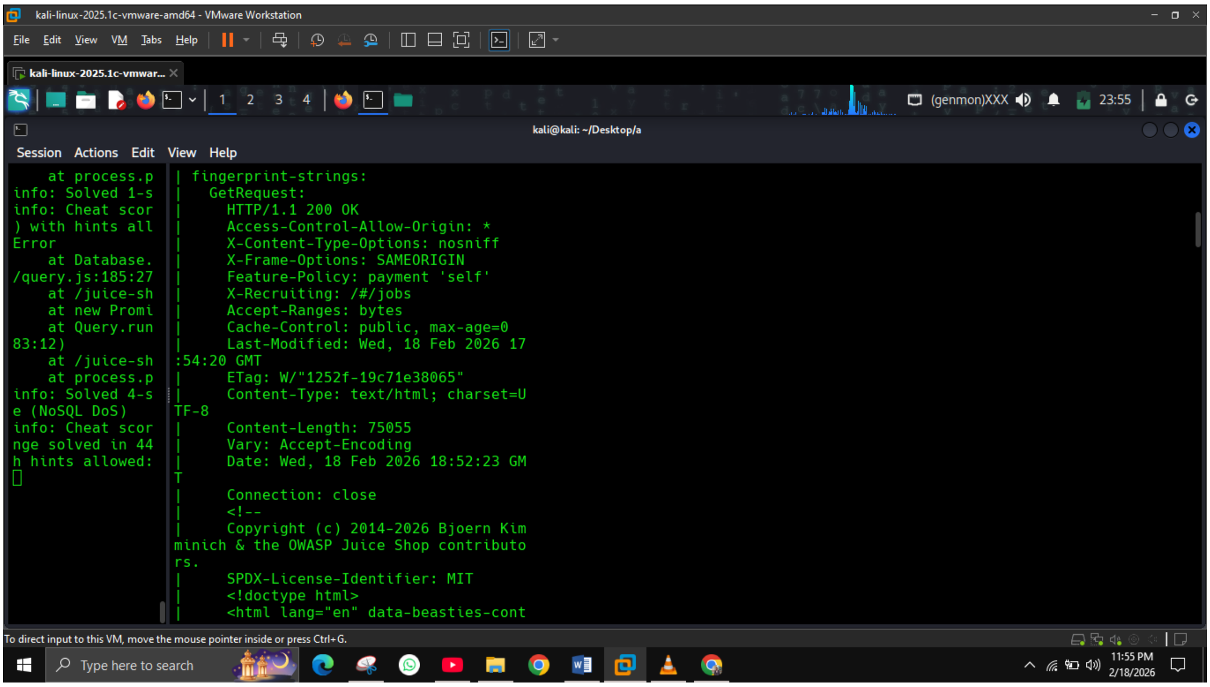
Findings

- **Port 80 (Apache):** Default Debian Apache page detected. Recommendation: disable or harden if not required.
- **Port 3000 (Juice Shop):** Application running with deliberate vulnerabilities. Confirmed headers like X-Frame-Options and X-Content-Type-Options are present, but XSS and SQL injection remain exploitable in Juice Shop (expected).
- **Port 3306 (MariaDB):** Database exposed with version info. Recommendation: restrict access to localhost only.
- **Custom Node.js App:**
    - XSS attempts blocked due to input validation.
    - SQL injection attempts failed due to sanitization.
    - Weak passwords rejected (minimum length enforced).


kali-linux-2025.1c-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2025.1c-vmwar...
(genmon)XXX
23:54
kali@kali: ~/Desktop/a
Session Actions Edit View Help
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Type here to search
11:54 PM
2/18/2026

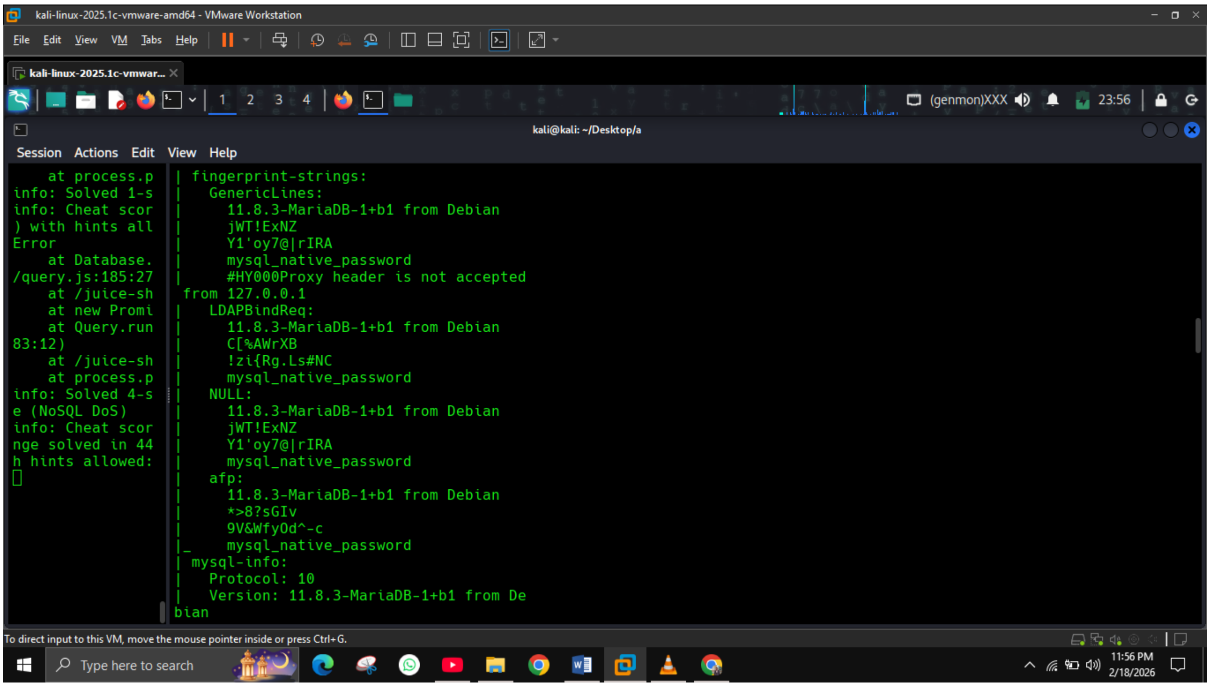
```
    at process.processTicksAndRejections (node:in
info: Solved 1-star errorHandlingChallenge (Error
info: Cheat score for trivial errorHandlingChalle
) with hints allowed: 0
Error
    at Database.<anonymous> (/juice-shop/node_mod
/query.js:185:27)
    at /juice-shop/node_modules/sequelize/lib/dia
    at new Promise (<anonymous>)
    at Query.run (/juice-shop/node_modules/sequel
83:12)
    at /juice-shop/node_modules/sequelize/lib/seq
    at process.processTicksAndRejections (node:in
info: Solved 4-star noSqlCommandChalleng
e (NoSQL DoS)
info: Cheat score for noSqlCommandChalle
nge solved in 44min (expected ~8min) wit
h hints allowed: 0
```

```
┌──(kali㉿kali)-[~/Desktop/a]
└─$ ^[[200~nmap -A localhost
zsh: bad pattern: ^[[200~nmap

┌──(kali㉿kali)-[~/Desktop/a]
└─$ sudo nmap -A localhost
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org )
at 2026-02-18 23:52 PKT
Nmap scan report for localhost (127.0.0.
1)
Host is up (0.00049s latency).
Other addresses for localhost (not scann
ed): ::1
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.
65 ((Debian))
|_http-title: Apache2 Debian Default Pag
e: It works
|_http-server-header: Apache/2.4.65 (Deb
ian)
3000/tcp open  ppp?
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
```


kali-linux-2025.1c-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2025.1c-vmwar...
(genmon)XXX
23:55
kali@kali: ~/Desktop/a
Session Actions Edit View Help
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Type here to search
11:55 PM
2/18/2026

```
    at process.p
info: Solved 1-s
info: Cheat scor
) with hints all
Error
    at Database.
/query.js:185:27
    at /juice-sh
    at new Promi
    at Query.run
83:12)
    at /juice-sh
    at process.p
info: Solved 4-s
e (NoSQL DoS)
info: Cheat scor
nge solved in 44
h hints allowed:
```

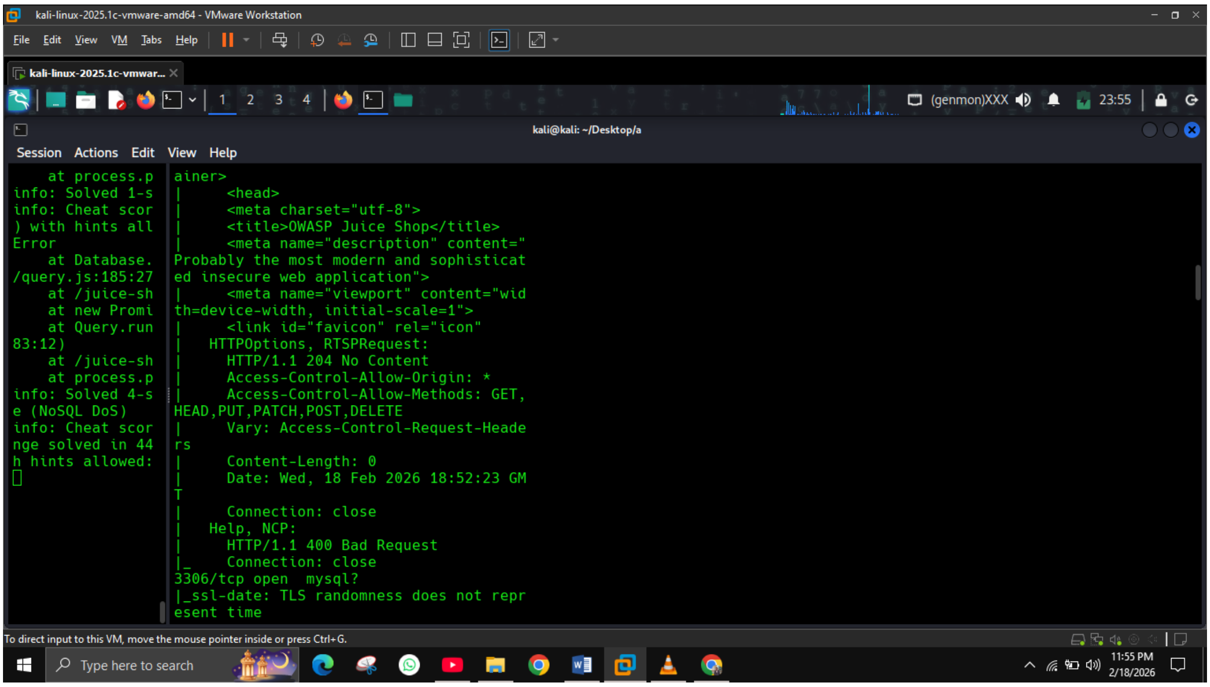
```
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Access-Control-Allow-Origin: *
|     X-Content-Type-Options: nosniff
|     X-Frame-Options: SAMEORIGIN
|     Feature-Policy: payment 'self'
|     X-Recruiting: /#/jobs
|     Accept-Ranges: bytes
|     Cache-Control: public, max-age=0
|     Last-Modified: Wed, 18 Feb 2026 17
:54:20 GMT
|     ETag: W/"1252f-19c71e38065"
|     Content-Type: text/html; charset=U
TF-8
|     Content-Length: 75055
|     Vary: Accept-Encoding
|     Date: Wed, 18 Feb 2026 18:52:23 GM
T
|     Connection: close
|     <!--
|     Copyright (c) 2014-2026 Bjoern Kim
minich & the OWASP Juice Shop contributo
rs.
|     SPDX-License-Identifier: MIT
|     <!doctype html>
|     <html lang="en" data-beasties-cont
```



kali-linux-2025.1c-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2025.1c-vmwar...
1 2 3 4
(genmon)XXX
23:56
kali@kali: ~/Desktop/a
Session Actions Edit View Help
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Type here to search
11:56 PM
2/18/2026

```
    at process.p
info: Solved 1-s
info: Cheat scor
) with hints all
Error
    at Database.
/query.js:185:27
    at /juice-sh
    at new Promi
    at Query.run
83:12)
    at /juice-sh
    at process.p
info: Solved 4-s
e (NoSQL DoS)
info: Cheat scor
nge solved in 44
h hints allowed:
```

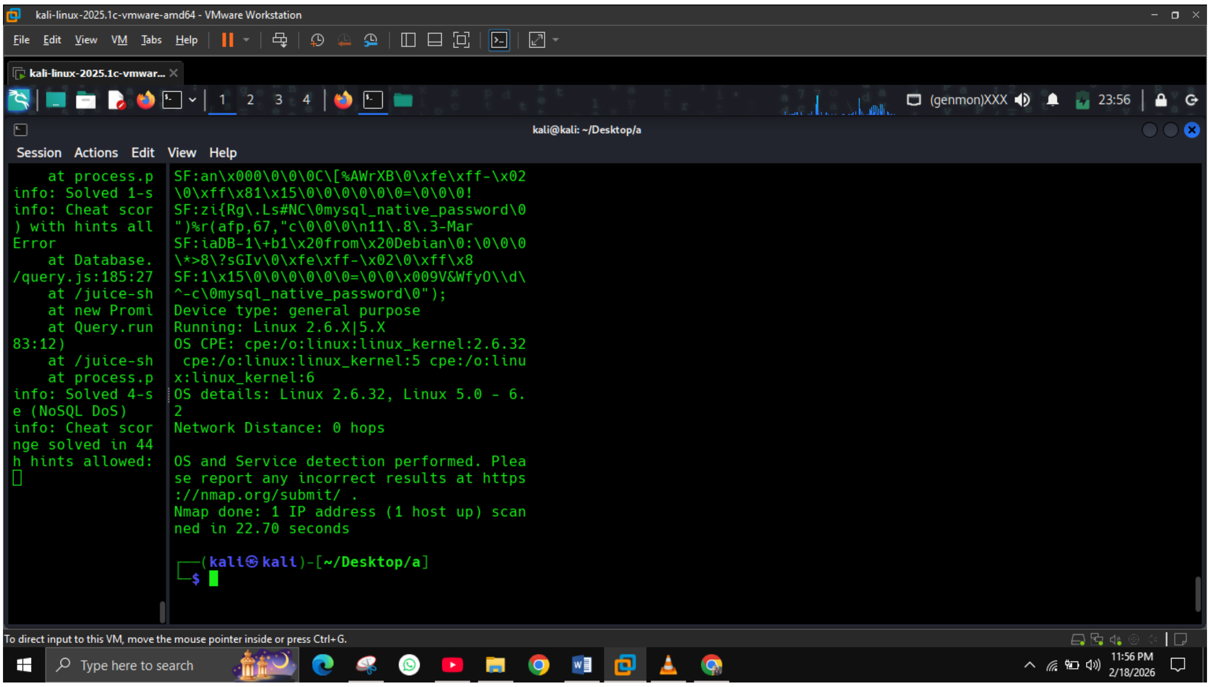
```
| fingerprint-strings:
|   GenericLines:
|     11.8.3-MariaDB-1+b1 from Debian
|     jWT!ExNZ
|     Y1'oy7@|rIRA
|     mysql_native_password
|     #HY000Proxy header is not accepted
 from 127.0.0.1
|   LDAPBindReq:
|     11.8.3-MariaDB-1+b1 from Debian
|     C[%AWrXB
|     !zi{Rg.Ls#NC
|     mysql_native_password
|   NULL:
|     11.8.3-MariaDB-1+b1 from Debian
|     jWT!ExNZ
|     Y1'oy7@|rIRA
|     mysql_native_password
|   afp:
|     11.8.3-MariaDB-1+b1 from Debian
|     *>8?sGIv
|     9V&WfyOd^-c
|_    mysql_native_password
| mysql-info:
|   Protocol: 10
|   Version: 11.8.3-MariaDB-1+b1 from De
bian
```



kali-linux-2025.1c-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2025.1c-vmwar...
1 2 3 4
(genmon)XXX
23:55
kali@kali: ~/Desktop/a
Session Actions Edit View Help
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Type here to search
11:55 PM
2/18/2026

```
    at process.p
info: Solved 1-s
info: Cheat scor
) with hints all
Error
    at Database.
/query.js:185:27
    at /juice-sh
    at new Promi
    at Query.run
83:12)
    at /juice-sh
    at process.p
info: Solved 4-s
e (NoSQL DoS)
info: Cheat scor
nge solved in 44
h hints allowed:
```

```
ainer>
|     <head>
|     <meta charset="utf-8">
|     <title>OWASP Juice Shop</title>
|     <meta name="description" content="
Probably the most modern and sophisticat
ed insecure web application">
|     <meta name="viewport" content="wid
th=device-width, initial-scale=1">
|     <link id="favicon" rel="icon"
|   HTTPOptions, RTSPRequest:
|     HTTP/1.1 204 No Content
|     Access-Control-Allow-Origin: *
|     Access-Control-Allow-Methods: GET,
HEAD,PUT,PATCH,POST,DELETE
|     Vary: Access-Control-Request-Heade
rs
|     Content-Length: 0
|     Date: Wed, 18 Feb 2026 18:52:23 GM
T
|     Connection: close
|   Help, NCP:
|     HTTP/1.1 400 Bad Request
|_    Connection: close
3306/tcp open  mysql?
|_ssl-date: TLS randomness does not repr
esent time
```



```
    at process.p
info: Solved 1-s
info: Cheat scor
) with hints all
Error
    at Database.
/query.js:185:27
    at /juice-sh
    at new Promi
    at Query.run
83:12)
    at /juice-sh
    at process.p
info: Solved 4-s
e (NoSQL DoS)
info: Cheat scor
nge solved in 44
h hints allowed:
```

```
SF:an\x000\0\0\0C\[%AWrXB\0\xfe\xff-\x02
\0\xff\x81\x15\0\0\0\0\0\0=\0\0\0!
SF:zi{Rg\.Ls#NC\0mysql_native_password\0
")%r(afp,67,"c\0\0\0\n11\.8\.3-Mar
SF:iaDB-1\+b1\x20from\x20Debian\0:\0\0\0
\*>8\?sGIv\0\xfe\xff-\x02\0\xff\x8
SF:1\x15\0\0\0\0\0\0=\0\0\x009V&WfyO\\d\
^-c\0mysql_native_password\0");
Device type: general purpose
Running: Linux 2.6.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
 cpe:/o:linux:linux_kernel:5 cpe:/o:linu
x:linux_kernel:6
OS details: Linux 2.6.32, Linux 5.0 - 6.
2
Network Distance: 0 hops

OS and Service detection performed. Plea
se report any incorrect results at https
://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scan
ned in 22.70 seconds

┌──(kali㉿kali)-[~/Desktop/a]
└─$ 
```

kali-linux-2025.1c-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2025.1c-vmwar...
1 2 3 4
(genmon)XXX
23:56
kali@kali: ~/Desktop/a
Session Actions Edit View Help
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Type here to search
11:56 PM
2/18/2026

2. Logging Implementation

Setup

- Installed **Winston** for structured logging.
- Configured both console and file logging (security.log).

Example Code

```
const winston = require('winston');

const logger = winston.createLogger({

  transports: [

    new winston.transports.Console(),

    new winston.transports.File({ filename: 'security.log' })

  ]

});

logger.info('Application started');
```

Usage

- Logs application startup.
- Records login attempts and failed authentications.
- Provides audit trail for suspicious activity.

3. Security Checklist

A simple checklist was created to ensure ongoing best practices:

- Validate all inputs (use validator).
- Hash and salt passwords (bcrypt).
- Use JWT for authentication.
- Secure headers with Helmet.
- Use HTTPS for data transmission.
- Restrict database access (bind to localhost).
- Log security events with Winston.
- Regularly run penetration tests (Nmap, browser-based attacks).

Conclusion

Week 3 validated the effectiveness of the security measures implemented in Week 2. Penetration testing confirmed that common attacks were blocked in the custom Node.js app. Logging was successfully integrated, providing visibility into security events. The final checklist ensures that best practices are consistently applied. Together, Weeks 1–3 provide a comprehensive cycle: **identification, remediation, validation, and documentation.**