

# Week 3: Advanced Security and Final Reporting

Name: Muhammad Bin Qasim

## Network Scanning with Nmap:

By using Nmap, I discovered devices and services running on a network.

**Version Check:** I confirmed Nmap version **7.98** is installed on a Windows platform.

**localhost Scan:** A basic scan was run on the local machine (127.0.0.1). It identified several open ports, including:

1. **Port 135:** Microsoft RPC.
2. **Port 445:** Microsoft-ds (File sharing).
3. **Port 3000:** This is where the web application was later found running.

**Service Detection:** Using the -sV flag, I identified specific versions of the software running on these ports to look for potential weaknesses.

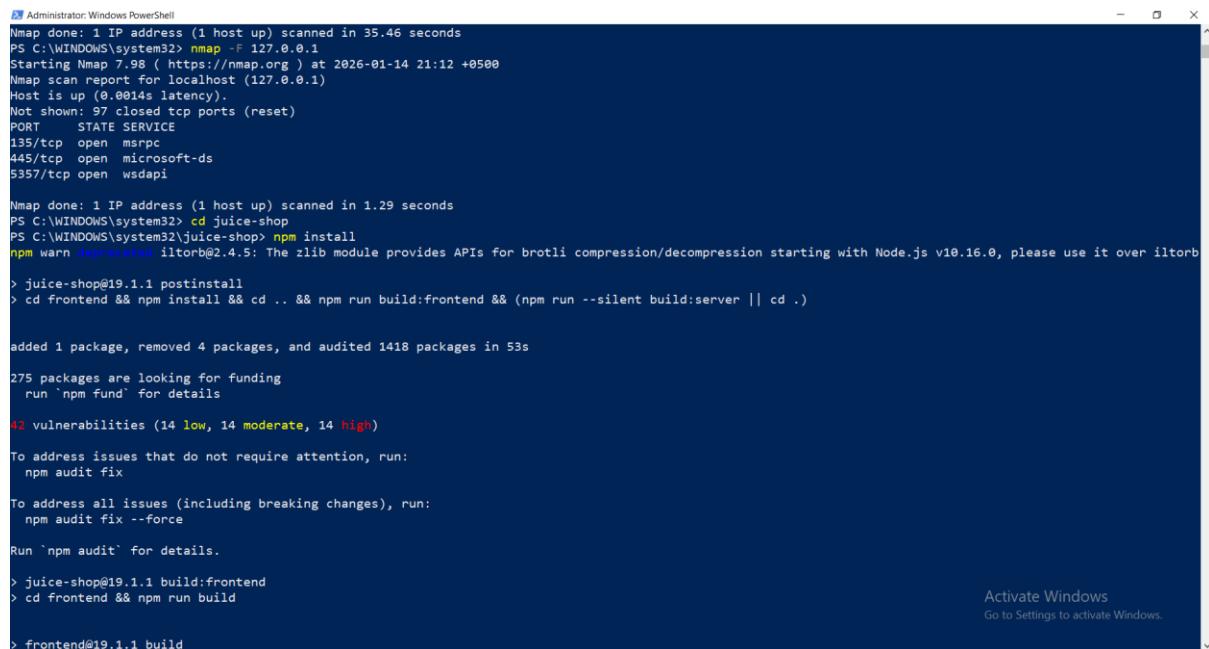
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> nmap --version
Nmap version 7.98 ( https://nmap.org )
Platform: i686-pc-windows-windows
Compiled with: nmap-liblua-5.4.8 openssl-3.0.17 nmap-libssh2-1.11.1 nmap-libz-1.3.1 nmap-libpcre2-10.45 Npcap-1.86 nmap-libdnet-1.18.0 ipv6
Compiled without:
Available nsock engines: iocp poll select
PS C:\WINDOWS\system32> nmap 127.0.0.1
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-14 21:10 +0500
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000010s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
445/tcp    open  microsoft-ds
2179/tcp   open  vmdrp
2382/tcp   open  ms-olap3
5357/tcp   open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
PS C:\WINDOWS\system32> nmap -sV 127.0.0.1
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-14 21:11 +0500
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
135/tcp    open  msrpc          Microsoft Windows RPC
445/tcp    open  microsoft-ds?
2179/tcp   open  vmdrp?
2382/tcp   open  ms-olap3?
5357/tcp   open  http           Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 35.46 seconds
```



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window displays a command-line session for setting up the OWASP Juice Shop application. The session starts with an Nmap scan of the local host (127.0.0.1), followed by navigating to the juice-shop directory and running an npm install command. This command installs the application and performs a security audit, identifying 42 vulnerabilities. The session concludes with a build command. A watermark in the bottom right corner reads "Activate Windows Go to Settings to activate Windows."

```
Nmap done: 1 IP address (1 host up) scanned in 35.46 seconds
PS C:\WINDOWS\system32> nmap -F 127.0.0.1
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-14 21:12 +0500
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0014s latency).
Not shown: 97 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
5357/tcp  open  wsddapi

Nmap done: 1 IP address (1 host up) scanned in 1.29 seconds
PS C:\WINDOWS\system32> cd juice-shop
PS C:\WINDOWS\system32\juice-shop> npm install
npm warn deprecated iltorb@2.4.5: The zlib module provides APIs for brotli compression/decompression starting with Node.js v10.16.0, please use it over iltorb

> juice-shop@19.1.1 postinstall
> cd frontend && npm install && cd .. && npm run build:frontend && (npm run --silent build:server || cd .)

added 1 package, removed 4 packages, and audited 1418 packages in 53s

275 packages are looking for funding
  run `npm fund` for details

42 vulnerabilities (14 low, 14 moderate, 14 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

> juice-shop@19.1.1 build:frontend
> cd frontend && npm run build

> frontend@19.1.1 build
Activate Windows
Go to Settings to activate Windows.
```

## OWASP Juice Shop Setup:

The below screenshots show the setup of OWASP Juice Shop, which is a dangerously insecure web application used for security training.

**Installation:** I navigated to the juice-shop folder and run npm install. The system flagged 42 vulnerabilities (14 low, 14 moderate, and 14 high) within the application's dependencies.

**Starting the App:** After installation, I run npm start. The application successfully launched on Port 3000.

**Verification:** A web browser screenshot shows the All Products page of the Juice Shop, confirming the store is active and displaying items like "Apple Juice" and "Banana Juice."

```

Administrator: Windows PowerShell
run 'npm fund' for details
43 vulnerabilities (1 low, 18 moderate, 17 high, 7 critical)

To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 11.6.2 -> 11.7.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.7.0
npm notice To update run: npm install -g npm@11.7.0
npm notice
PS C:\WINDOWS\system32\juice-shop> npm start

> juice-shop@19.1.1 start
> node build/app

info: Detected Node.js version v24.12.0 (OK)
info: Detected OS win32 (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 20 of 20 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file runtime.js is present (OK)
info: Required file vendor.js is present (OK)
info: Required file tutorial.js is present (OK)
info: Required file styles.css is present (OK)
info: Required file main.js is present (OK)
info: Port 3000 is available (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Domain https://www.alchemy.com/ is reachable (OK)
info: Server listening on port 3000
PS C:\WINDOWS\system32\juice-shop> mkdir security-week2

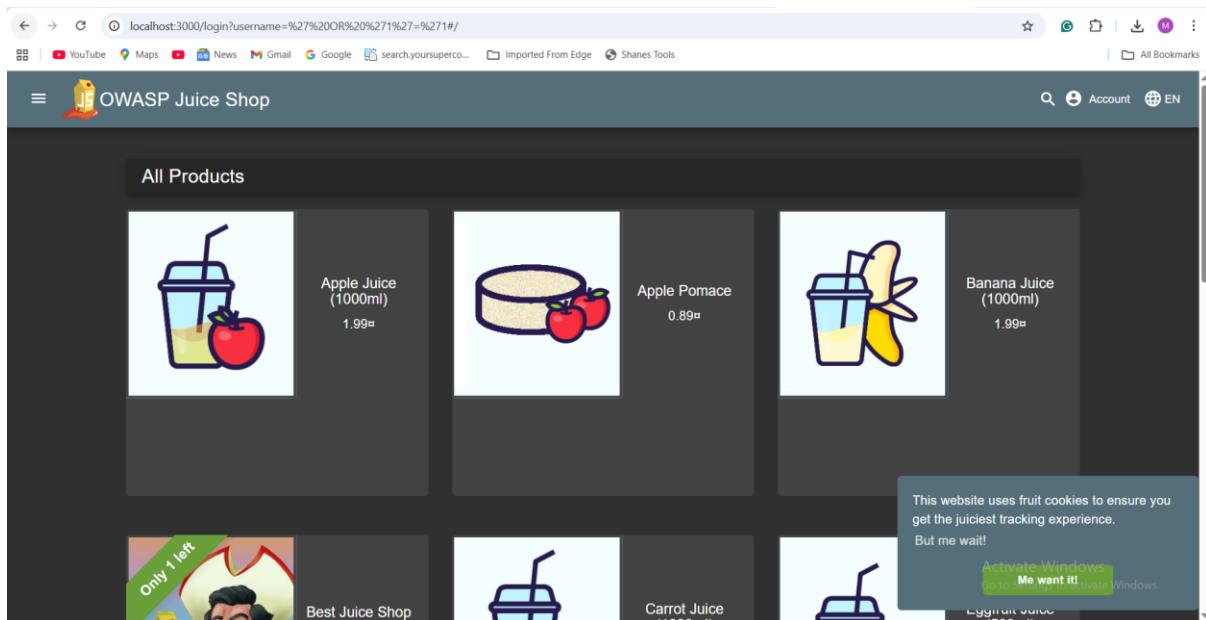
```

Activate Windows  
Go to Settings to activate Windows.

## Security Testing: SQL Injection:

Successful attempt to exploit a common web vulnerability called SQL Injection.

- 1. The Attack:** I entered a malicious string into the login URL: ' OR '1'='1
- 2. The Result:** This specific string is designed to trick the database into thinking a password is correct, even if it is not. The screenshot shows the user successfully bypassed the login screen and gained access to the store interface without a valid password.



## Implementation of Logging System:

Finally, I created a custom security logging tool to track activities within the environment.

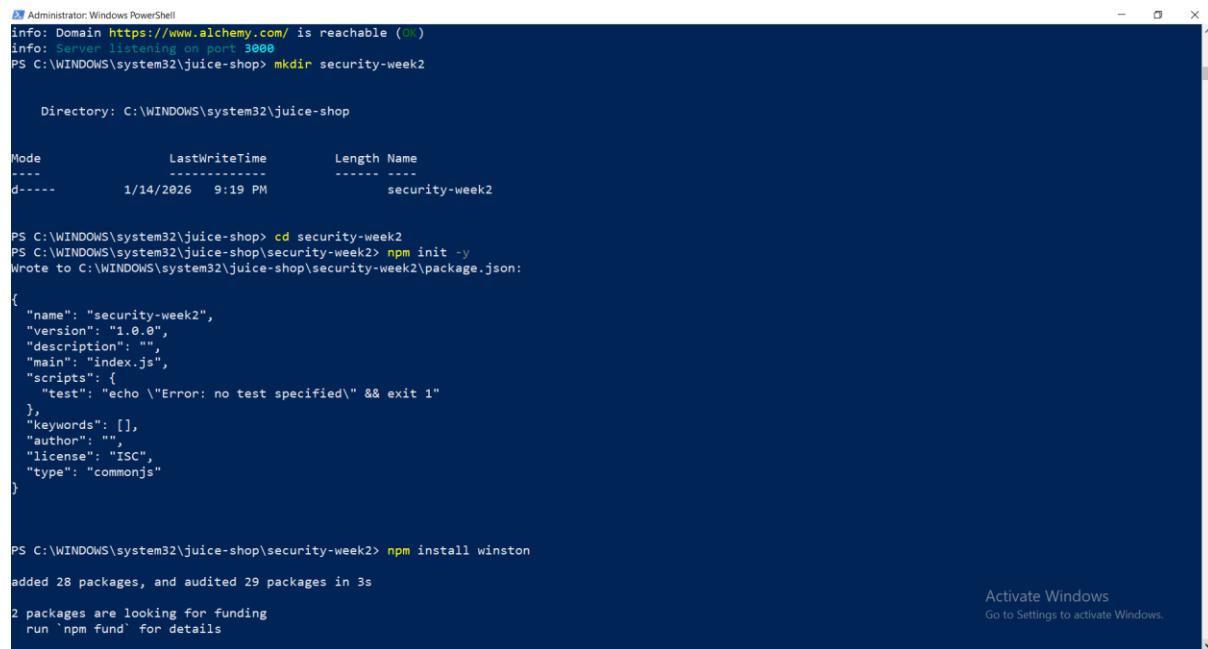
**Project Initialization:** A new directory named security-week2 was created, and a new Node.js project was started using npm init -y.

**Winston Library:** Installed **Winston**, a popular logging library for Node.js.

**Coding the Logger:** A file named logger.js was created.

**Testing:** I run node logger.js, and the terminal showed a successful log entry:

```
{"level": "info", "message": "Application started", ...}.
```



```
Administrator: Windows PowerShell
info: Domain https://www.alchemy.com/ is reachable (OK)
info: Server listening on port 3000
PS C:\WINDOWS\system32\juice-shop> mkdir security-week2

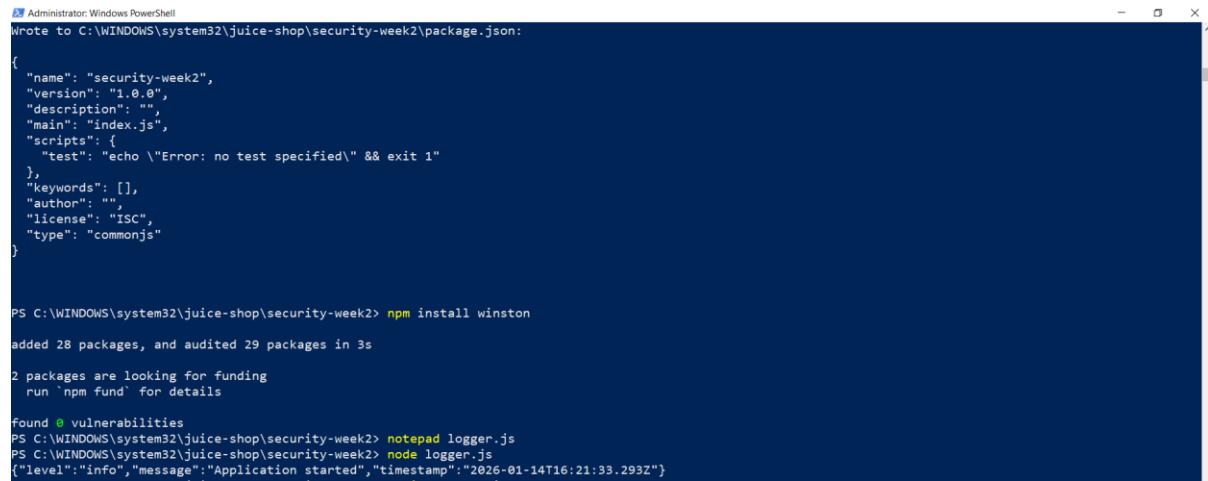
Directory: C:\WINDOWS\system32\juice-shop

Mode                LastWriteTime         Length Name
----                -----        ---- 
d----   1/14/2026 9:19 PM            security-week2

PS C:\WINDOWS\system32\juice-shop> cd security-week2
PS C:\WINDOWS\system32\juice-shop\security-week2> npm init -y
Wrote to C:\WINDOWS\system32\juice-shop\security-week2\package.json:

{
  "name": "security-week2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}

PS C:\WINDOWS\system32\juice-shop\security-week2> npm install winston
added 28 packages, and audited 29 packages in 3s
2 packages are looking for funding
  run 'npm fund' for details
Activate Windows
Go to Settings to activate Windows.
```



```
Administrator: Windows PowerShell
Wrote to C:\WINDOWS\system32\juice-shop\security-week2\package.json:

{
  "name": "security-week2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}

PS C:\WINDOWS\system32\juice-shop\security-week2> npm install winston
added 28 packages, and audited 29 packages in 3s
2 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
PS C:\WINDOWS\system32\juice-shop\security-week2> notepad logger.js
PS C:\WINDOWS\system32\juice-shop\security-week2> node logger.js
{"level": "info", "message": "Application started", "timestamp": "2026-01-14T16:21:33.293Z"}
```

## **Logger.js File:**

```
logger - Notepad
File Edit Format View Help
const winston = require('winston');

const logger = winston.createLogger({
  level: 'info',
  format: winston.format.combine(
    winston.format.timestamp(),
    winston.format.json()
  ),
  transports: [
    new winston.transports.Console(),
    new winston.transports.File({ filename: 'security.log' })
  ]
});

logger.info('Application started');

module.exports = logger;
|
```

---

## **At Last:**

Through this task, I learned how to use Nmap for network reconnaissance to identify open ports like 135, 445, and 3000 on a local system. I gained hands-on experience in web security by successfully using a SQL injection payload, ' OR '1'='1, to bypass the authentication of an OWASP Juice Shop application. Additionally, I developed a functional logging system using the Winston library in Node.js to record security events into a persistent security.log file.

---

## **GitHub Repository Link:**

<https://github.com/MBQ17/Week-3---Advanced-Security>

---