LAB CREATION GUIDE

Student Name: Manave Angelkumari & Abin Binu

Student ID: 10312082 & 10328278

Course: COMP-357 Advanced Pentesting

Instructor: Adam Abernethy

Exercises used: Juice Shop & Browser Exploitation Framework (BeEF)

Submission Date: 12/07/2025

1. Introduction

This Exercise shows how a browser-based attack works using the BeEF framework. The idea is to create a small environment where an attacker can hook a victim's browser and run different client-side exploits. I used a Kali Linux machine as the attacker and a Windows machine as the victim. For the second exercise, Juice Shop is used as the place where the malicious script is injected.

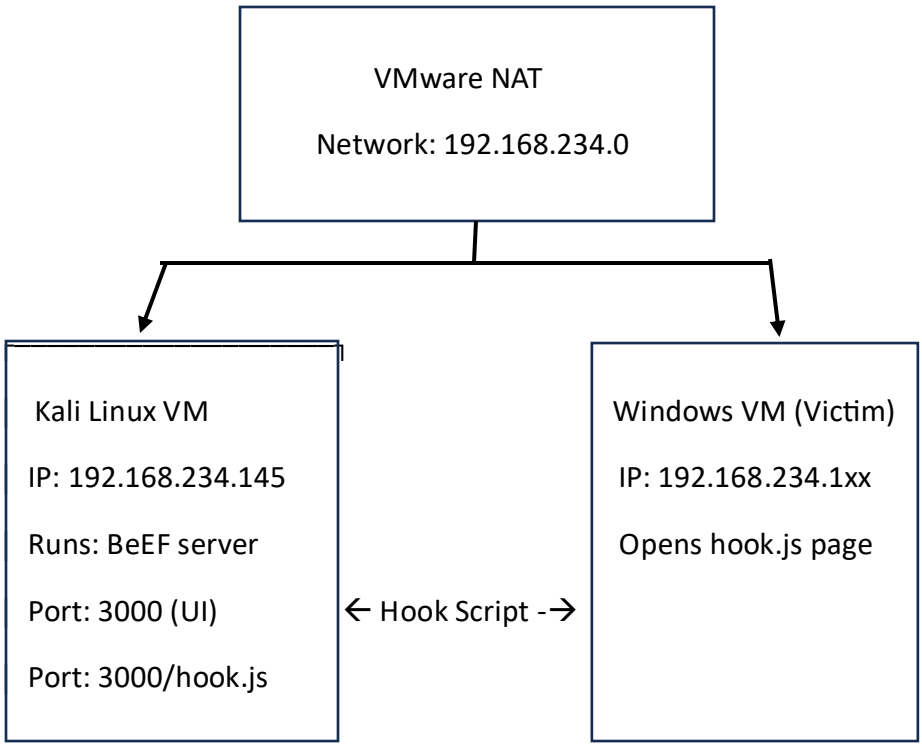
The goal is to make the setup easy to repeat from scratch, so anyone can recreate this small environment, start BeEF, hook a browser, and try out different modules.

2. Infrastructure Overview

- 2.1 Hosts Used:

Role	System	Description
Attacker	Kali Linux 2025 VM	Runs BeEF, controls hooked browsers
Victim	Windows 10/11 VM	Opens the page containing the hook.js script
Vulnerable App	Juice Shop	Used for the SQL injection

- 2.2 Network Layout:



Key Ports

Port	Use
3000	BeEF web interface + hook.js delivery
80/443	Juice Shop (Docker default)

3. Prerequisites

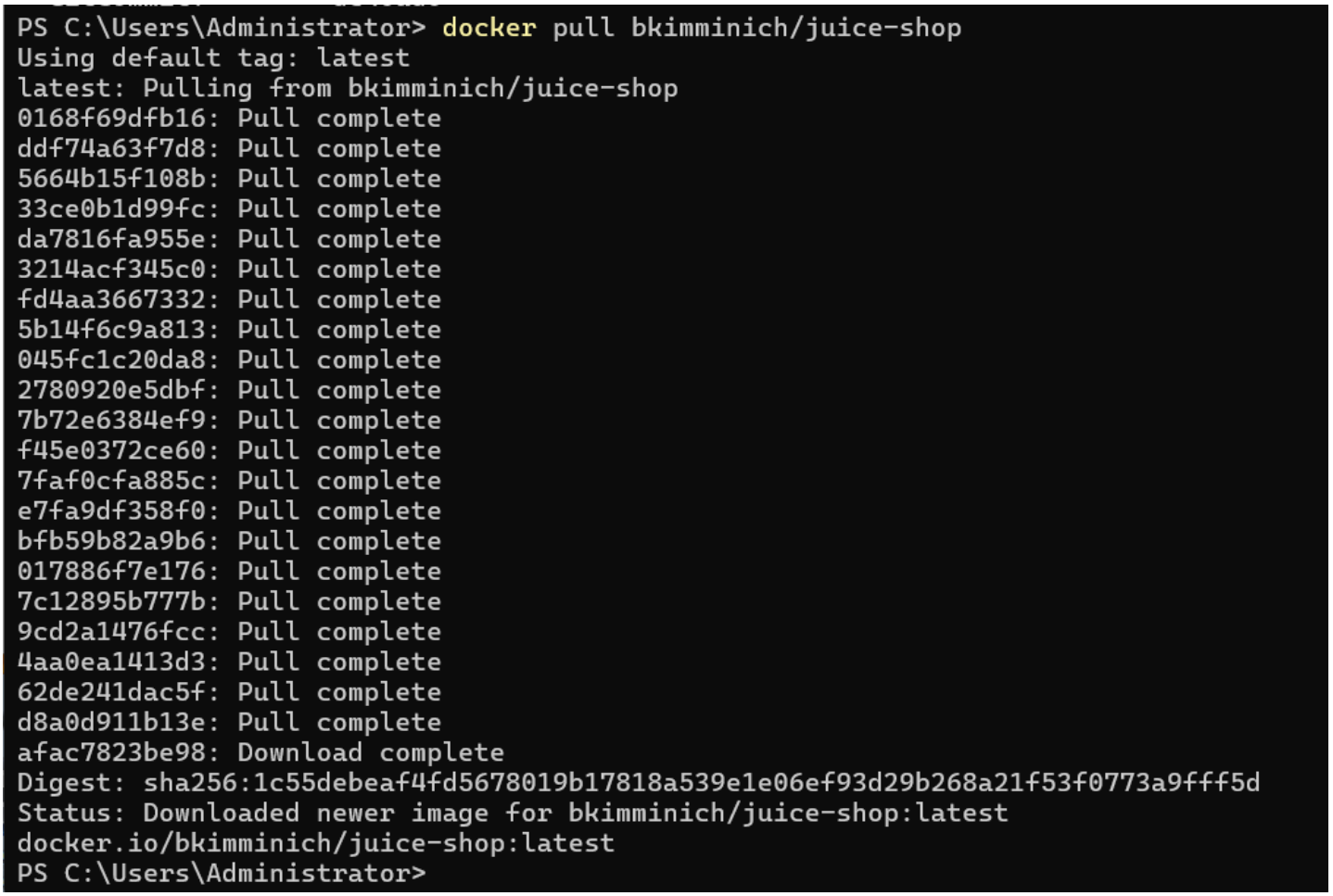
Before starting, we have to make sure that we have:

- VMware or VirtualBox
- Kali Linux updated
- Windows VM with Edge or Chrome
- Docker Desktop installed (for Juice Shop)
- Internet access for installation

4. Environment Setup Steps

- Step 1: Install Juice Shop:
  - For this, we use docker for the installation and successful running of the process, and the commands for this are;

    docker pull bkimminich/juice-shop
  - Capture below is of the command docker pull executed on PowerShell;

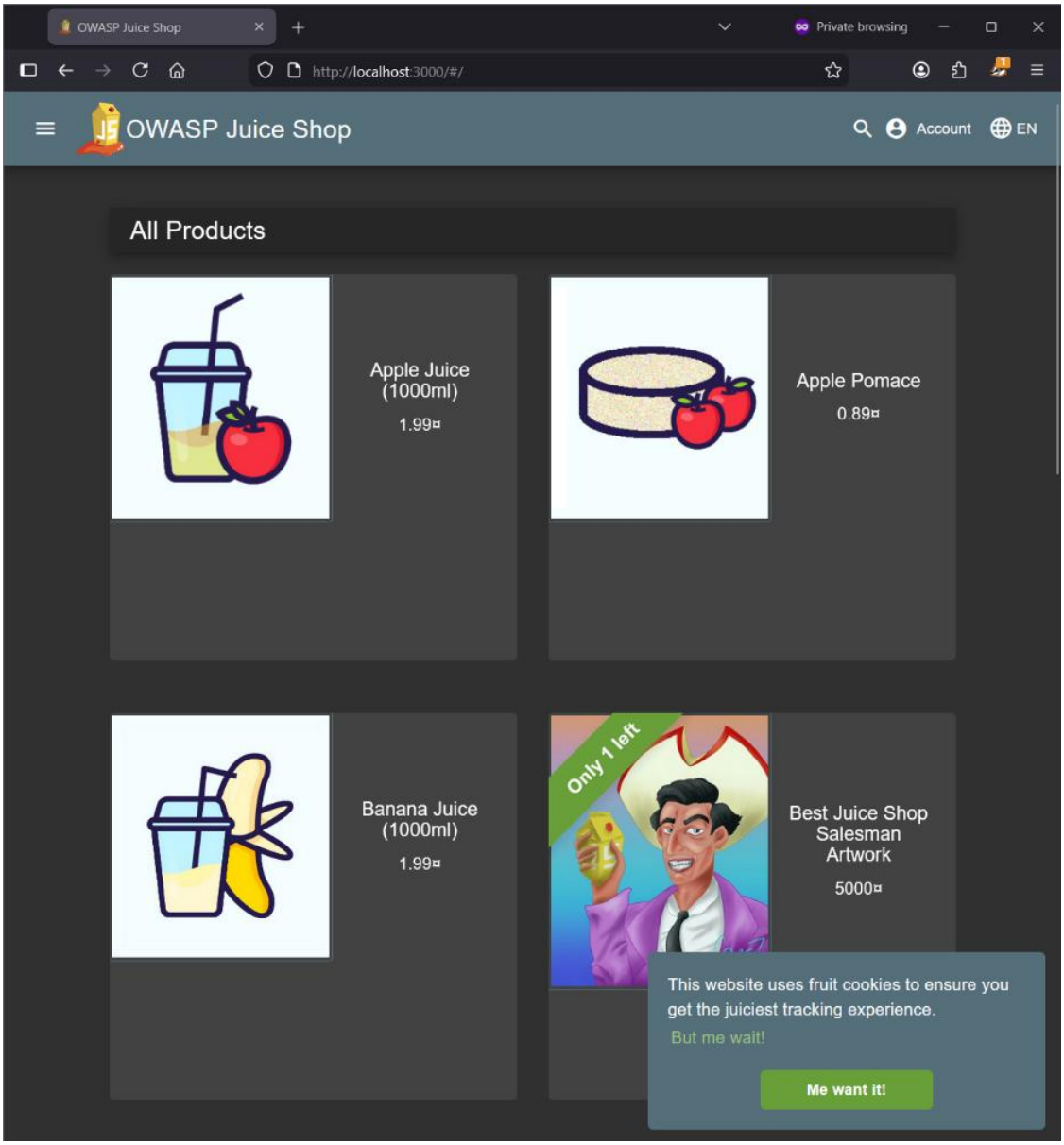


docker run -d -p 3001:3000 bkimminich/juice-shop

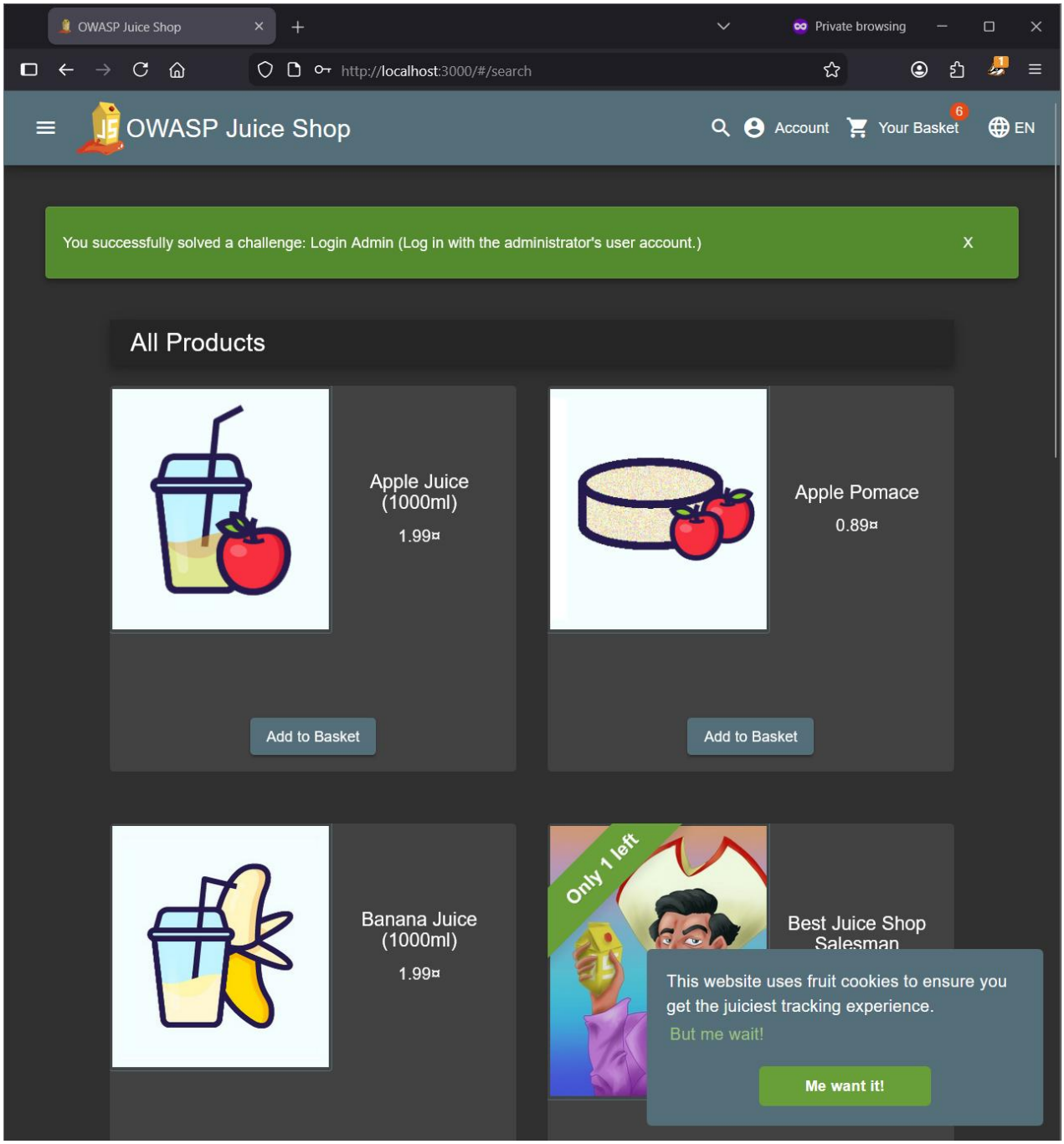
- Capture below is of the juice-shop run command executed on PowerShell;

```
PS C:\Users\Administrator> docker run -d --name juice-shop -p 3000:3000 bkimminich/juice-shop
4b03ff22040d640261cfb0f10544d0d1b8840f407eebec27f6f2716672313723
PS C:\Users\Administrator>
```

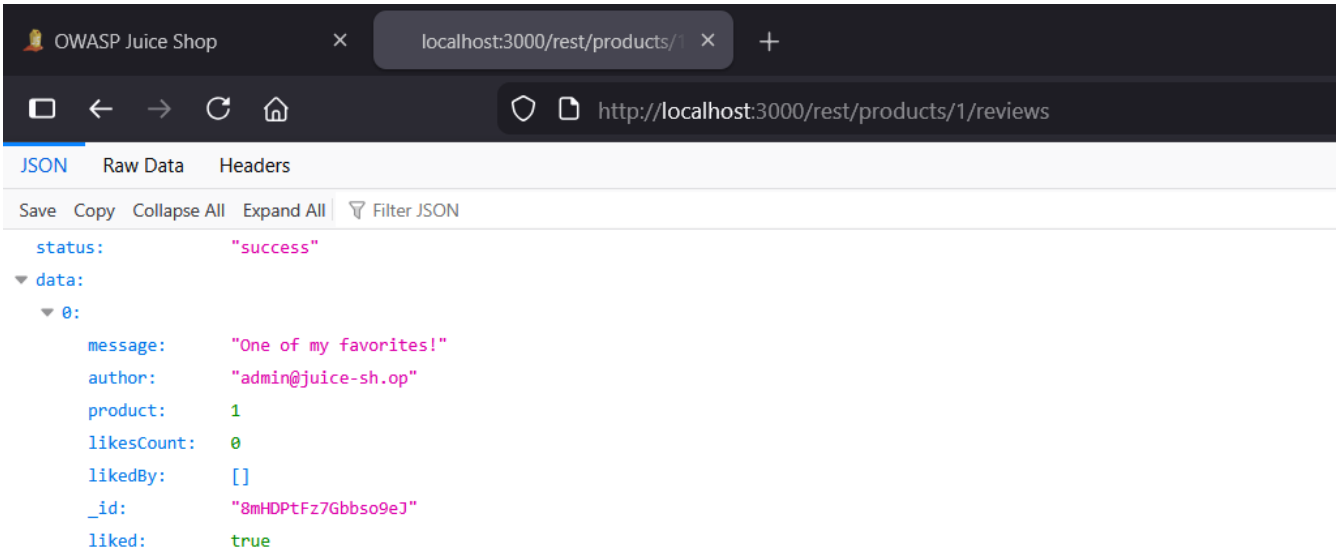
- After that, to access Juice Shop, the command used is; http://localhost:3001
- Capture below is of the main page of Juice Shop;



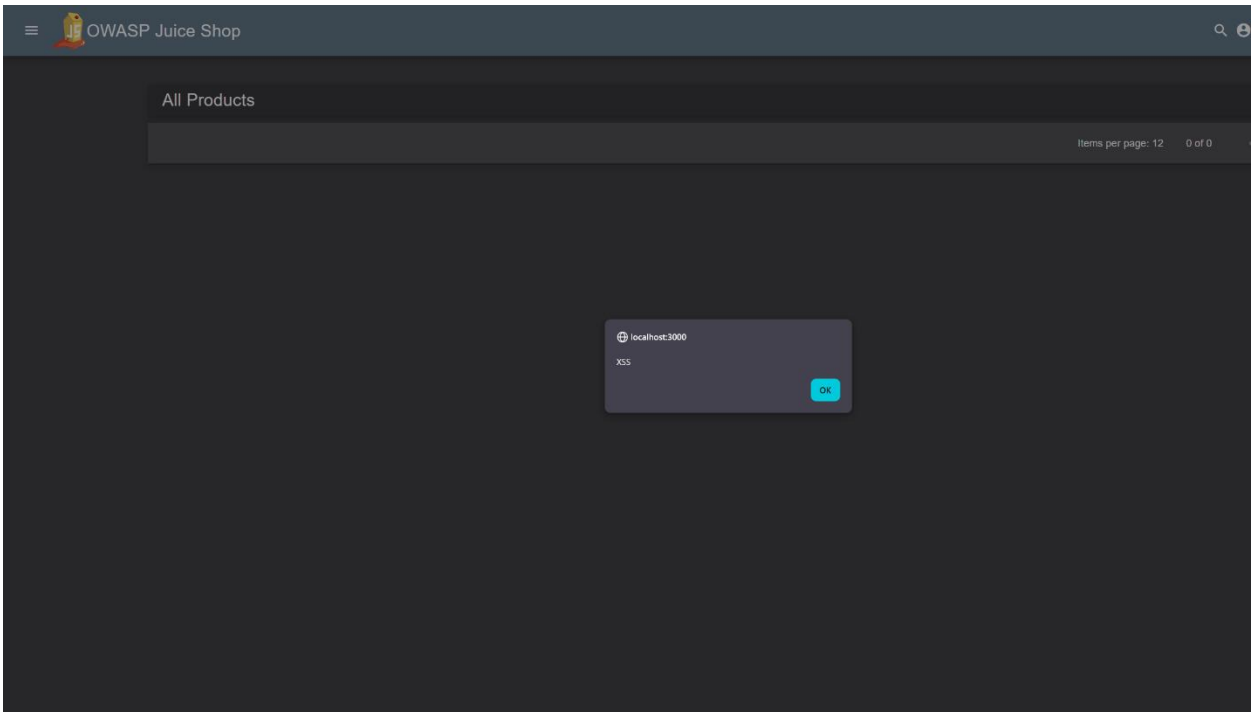
- After that, we need to perform an SQL injection attack to login,
  Username: ‘ OR 1=1—
  Password: abc
- Capture below is of the attacked performed properly and login successfully;



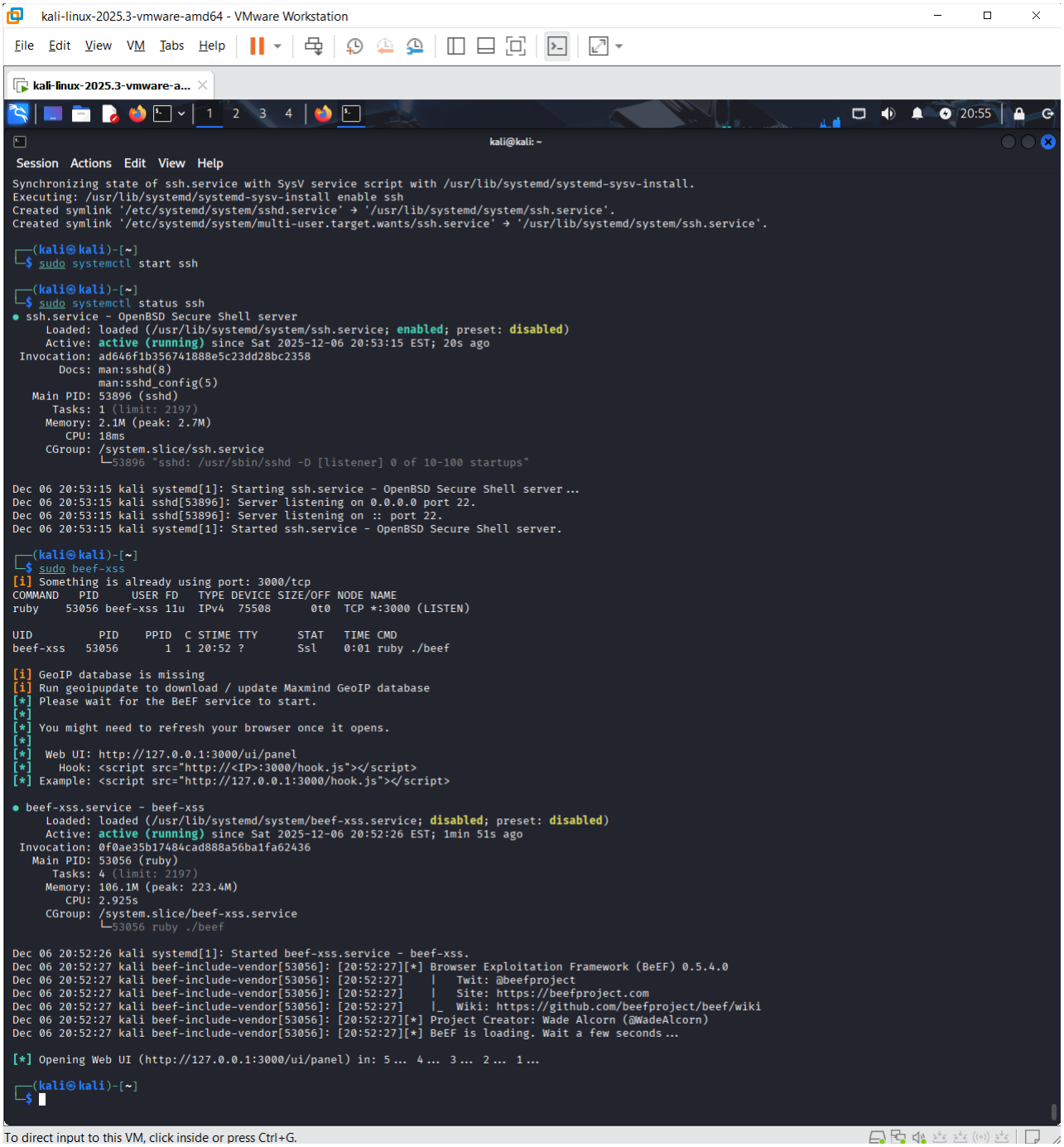
- As for the second exploit in juice shop, we will execute an IDOR attack to prove that broken access controls allow attackers to view other users' data just by modifying IDs in URLs.
- Capture below is of the successful data exposure by modifying IDs in URLs;



- For the third exploit in juice shop, we are going to perform a Stored XSS attack.
- Capture below is of the successful execution of the stored XSS attack;

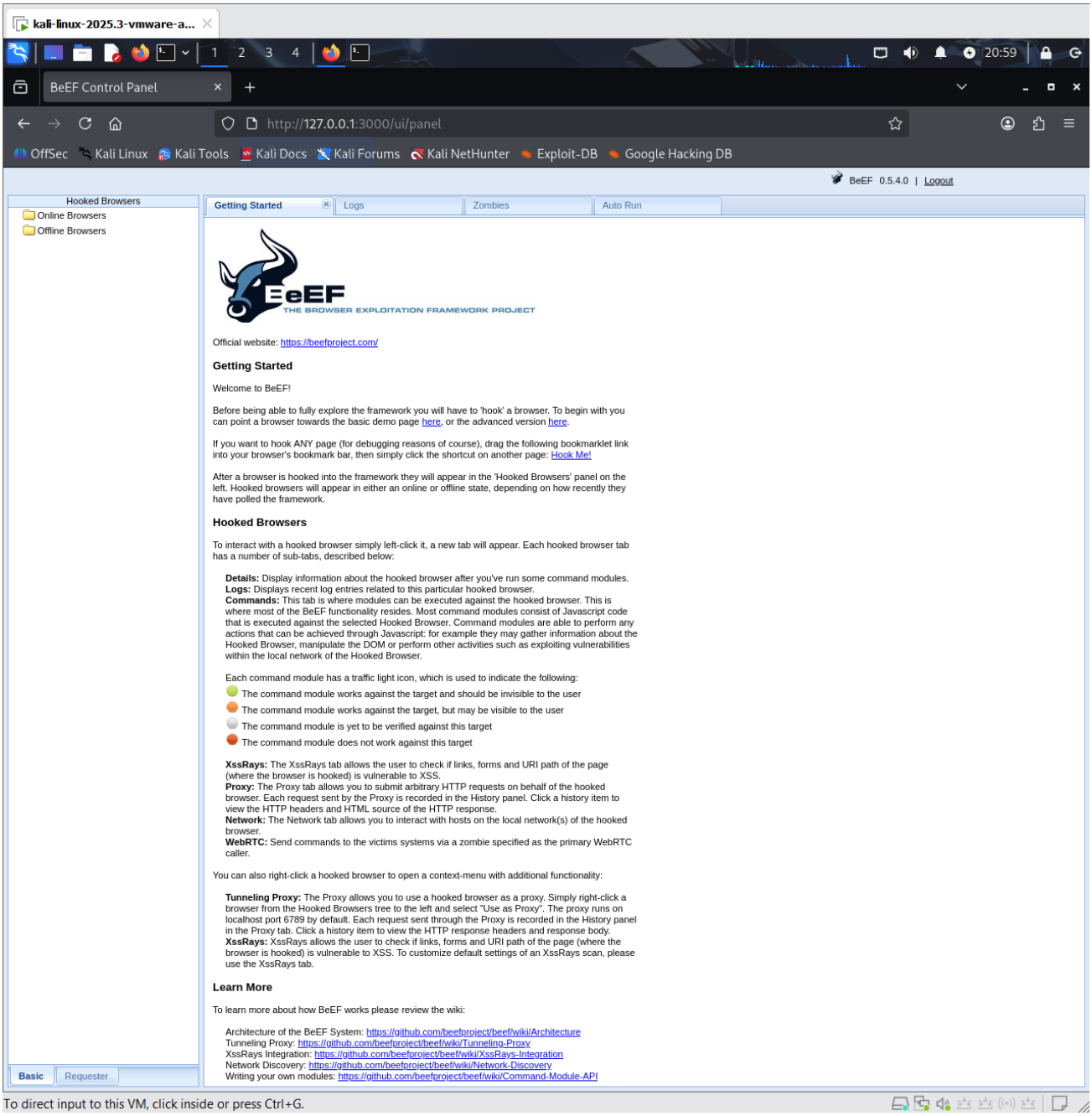


- Step 2: Install BeEF on Kali:
  - For this, we need to run the following steps inside kali;
    sudo apt update
    sudo apt install beef-xss -y
  - After that, the command used to start the service is;
    sudo systemctl start beef-xss
  - Capture below is of the BeEF service successfully running;



➢ Step 3: Access BeEF UI:

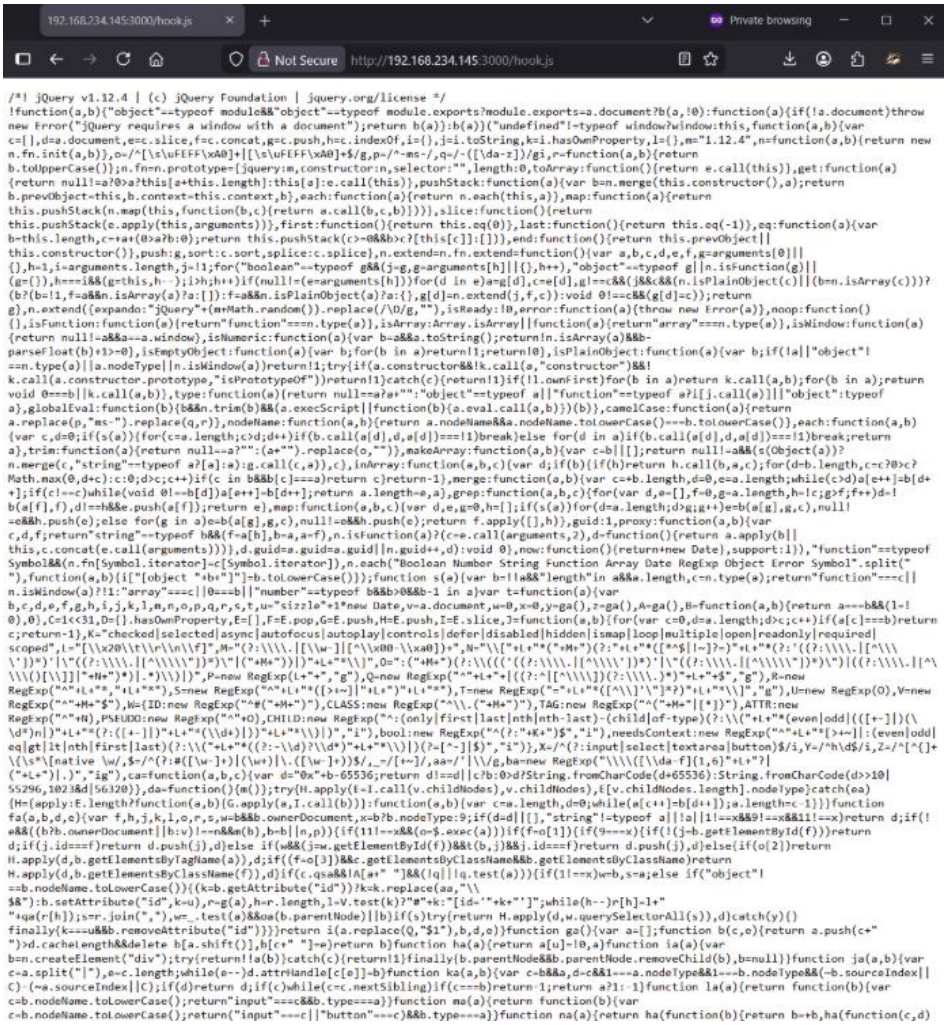
- Open this on Kali's browser:
  http://127.0.0.1:3000/ui/panel
- Given below are the default credentials, but while installing beef it will ask for a password:
  username: beef
  password: tcpip
- Capture below is of the BeEF dashboard after the successful login;



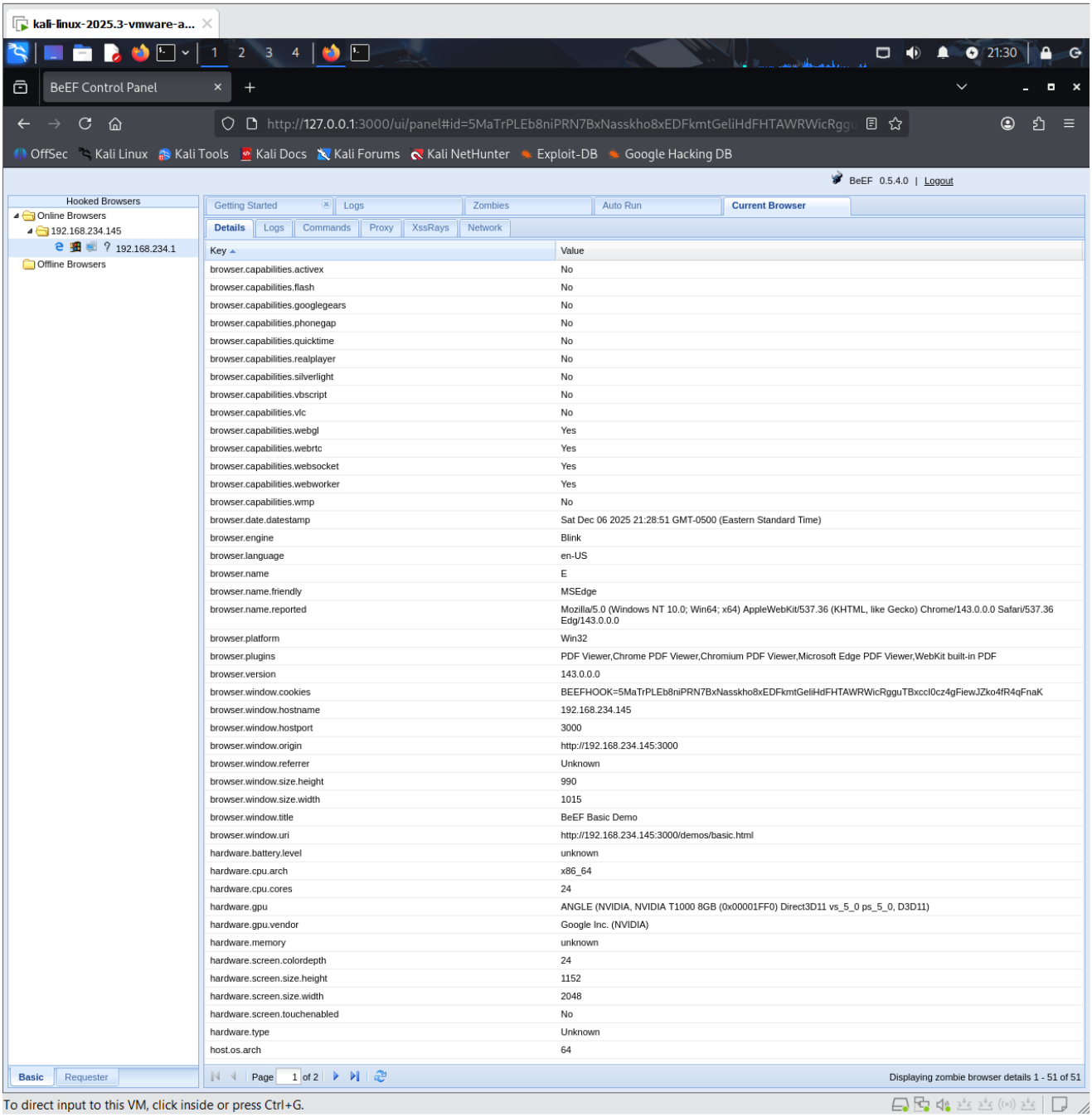
- Step 4: Obtain the Hook URL:
  - BeEF uses this script below to hook browsers which has the Vms Ip address: http://192.168.234.145:3000/hook.js

    After copying it, we will inject it into the victim's browser.

  - Capture below is of the hooks.js loading on the browser;



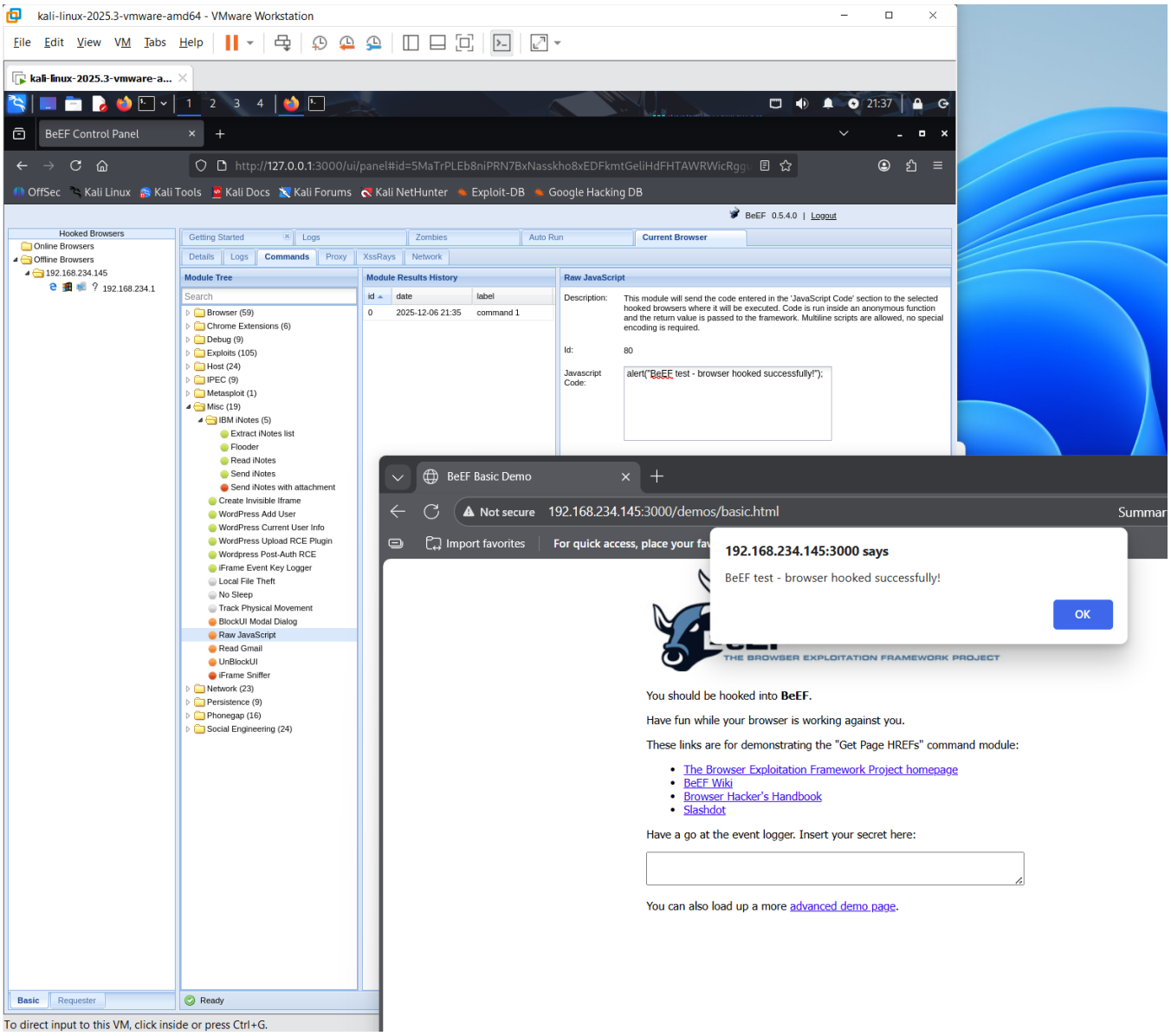
- Step 5: Confirm Browser Hooking:
  - Go back to Kali - BeEF UI.
    Under Online Browsers, we should be able to see our Windows machine appear (with IP).
  - Capture below is of the hooked browser listed in BeEF;



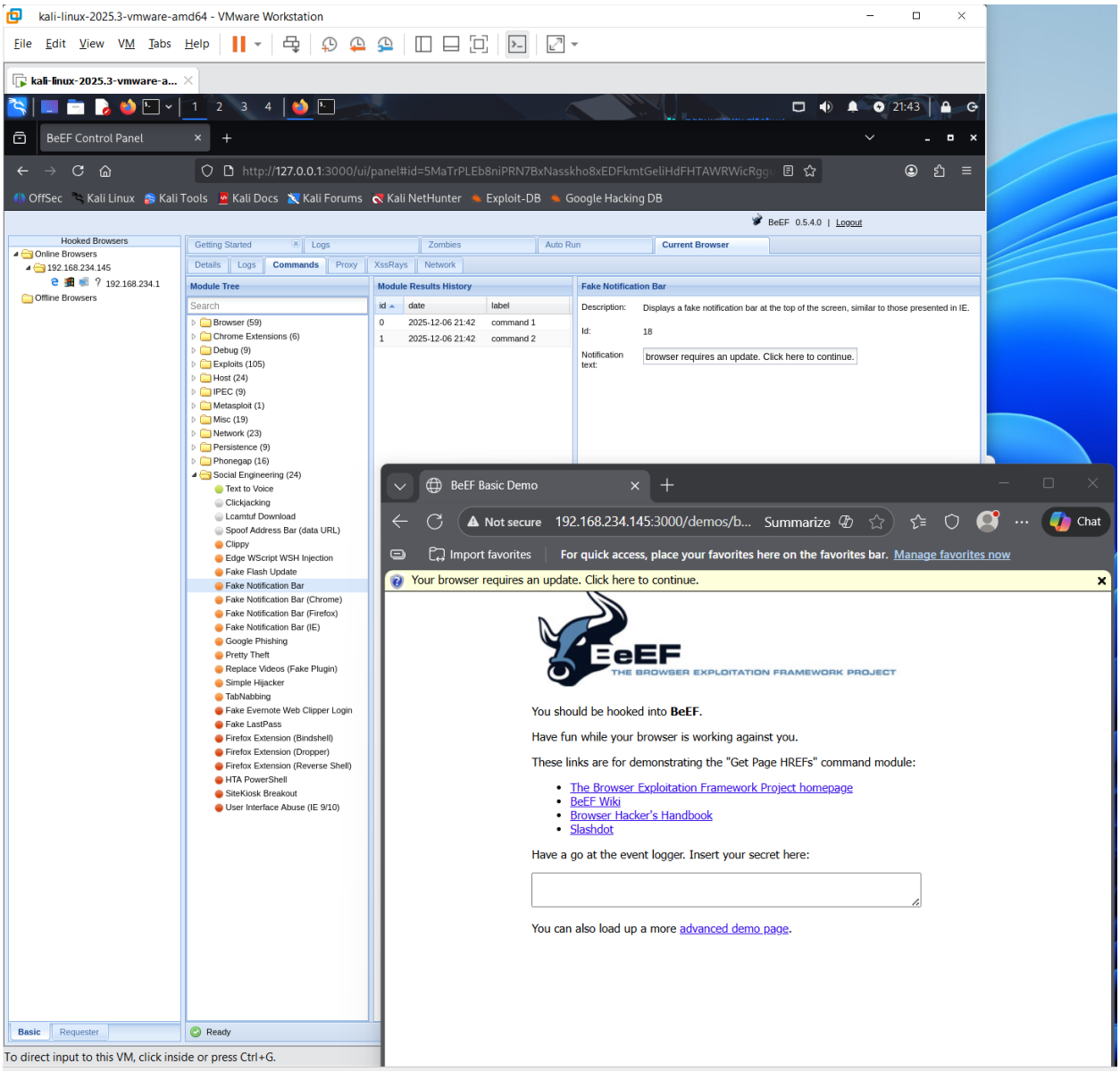
5. Running Attack Modules

For this part of the BeEF exercise, we run simple modules to show the attack is working:

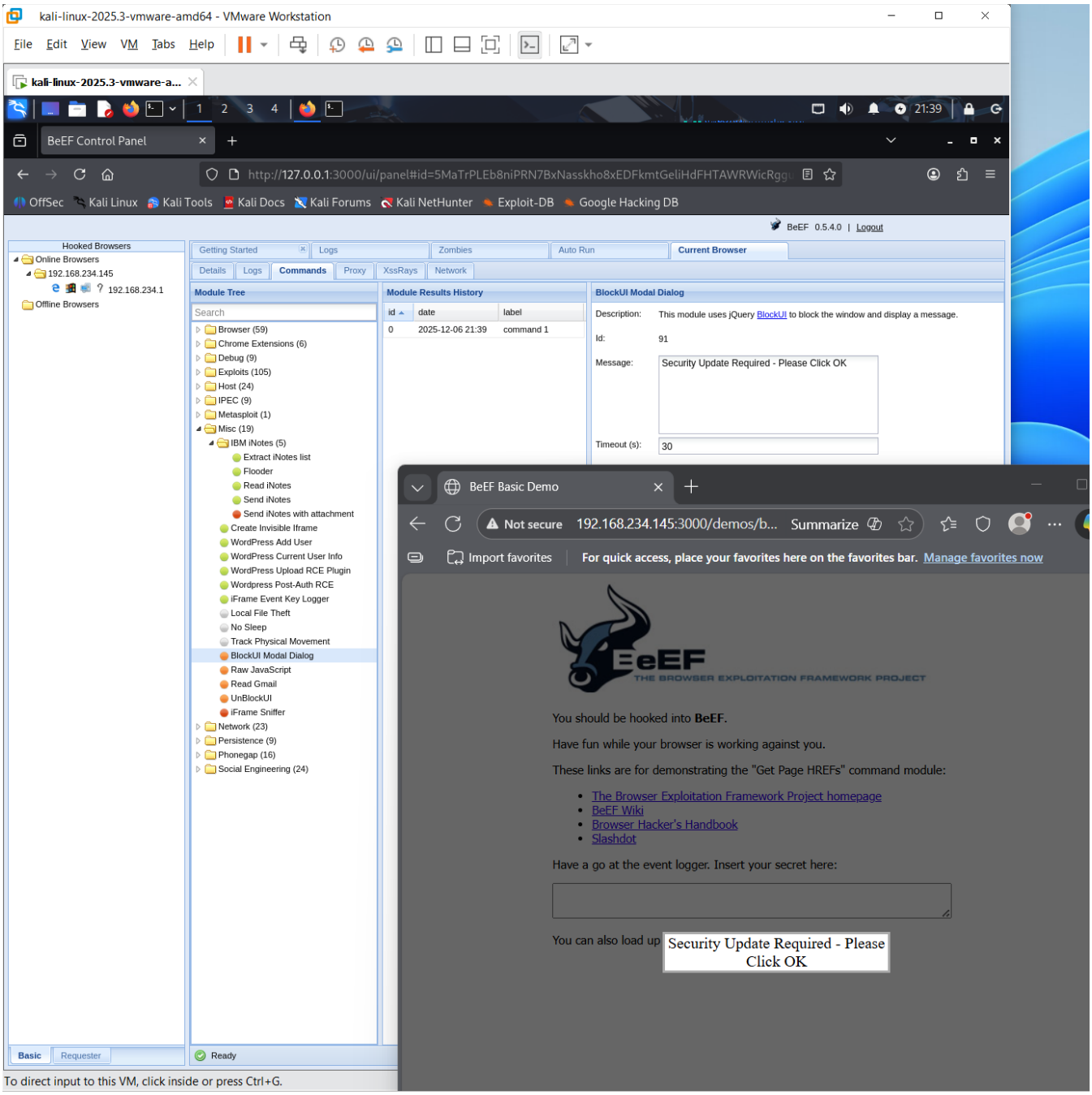
- Alert Popup:
  - Module: Raw JavaScript
  - Payload used for execution:
    alert('BeEF test - browser hooked successfully!')
  - Capture below is of windows popup on the browser;



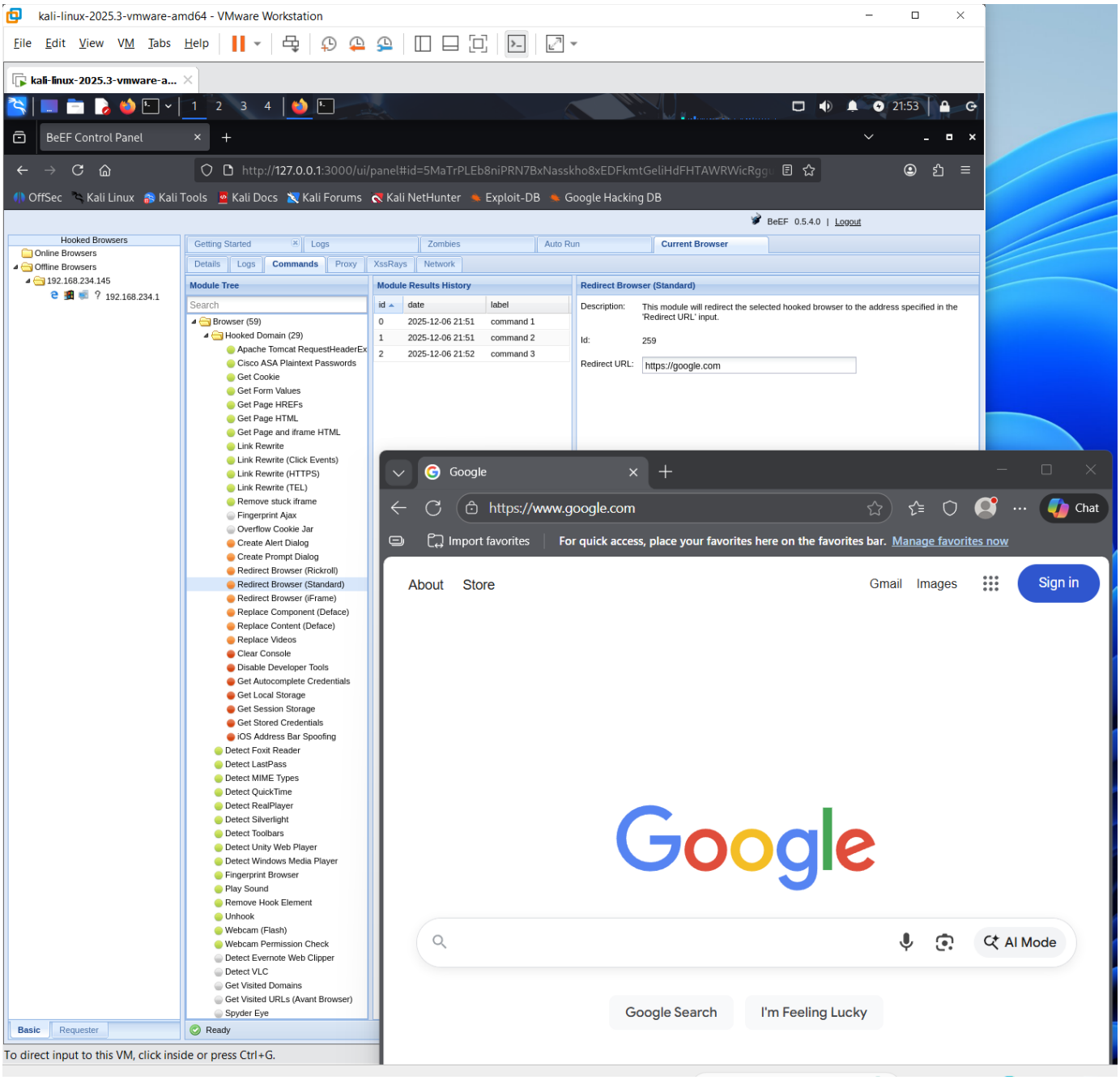
- Fake Notification Bar:
  - Module: Social Engineering - Fake Notification Bar
  - Payload used for execution: Your browser requires an update. Click here to continue.
  - Capture below is of the fake bar displayed on the browser;



- BlockUI Modal Dialog:
  - Module: Misc - BlockUI Modal Dialog
  - Payload used for the execution: Security Update Required — Please Click OK
  - Capture below is of Block UI message on the browser;



- Redirect Browser:
  - Module: Browser - Redirect Browser
  - Payload used for the execution: https://google.com
  - Capture below is of the browser redirected;



6. Final Environment Summary

Our final attack environment includes:

- Kali Linux VM running BeEF.
- Windows VM hooked through hook.js.
- Juice Shop or demo HTML page used as injection point.
- Ability to run multiple client-side modules.
- Evidences in the form of screenshots.