Student:	Email:
Fernando Parra	fparra1@msudenve.edu

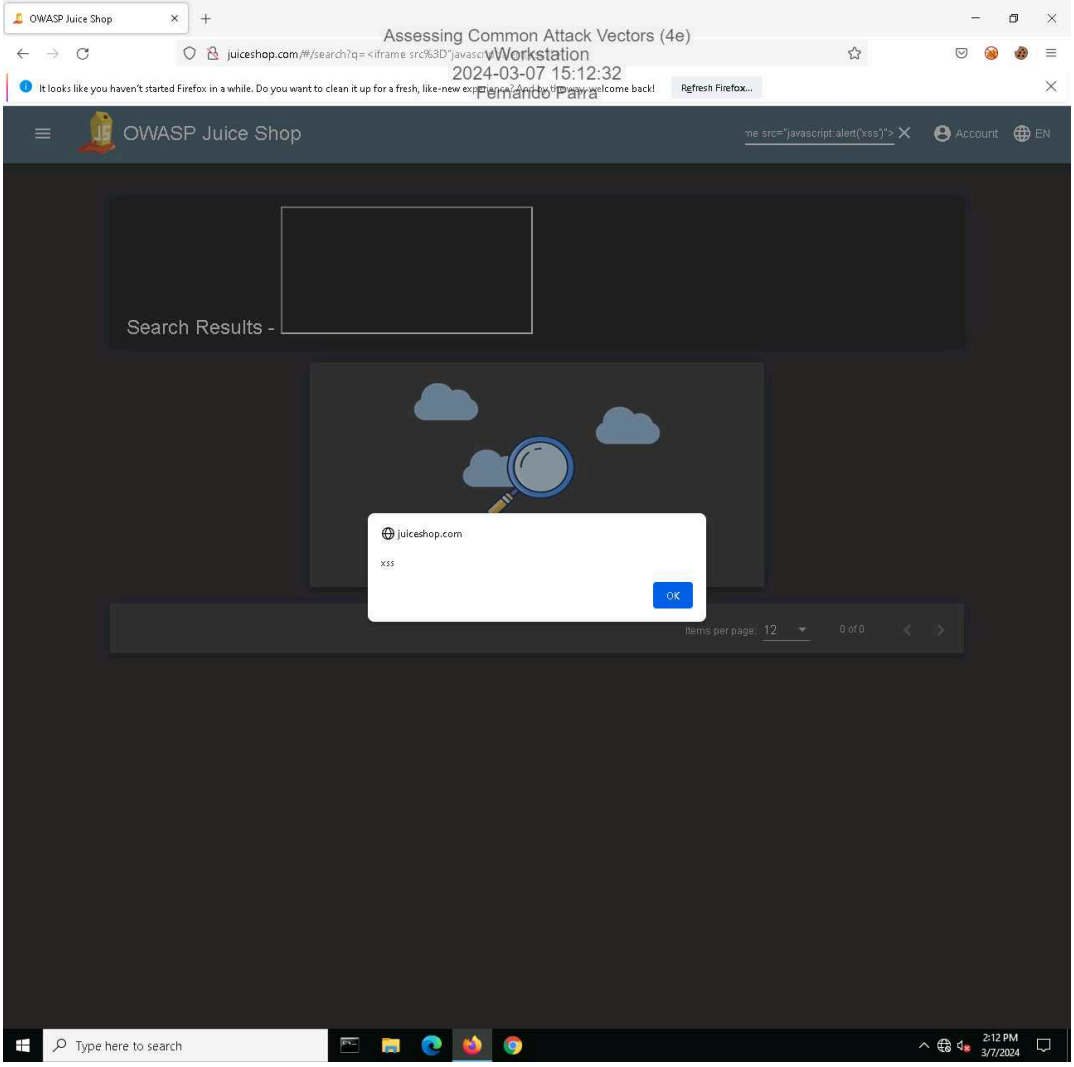
Time on Task:	Progress:
9 hours, 38 minutes	100%

Report Generated: Saturday, March 23, 2024 at 12:08 AM

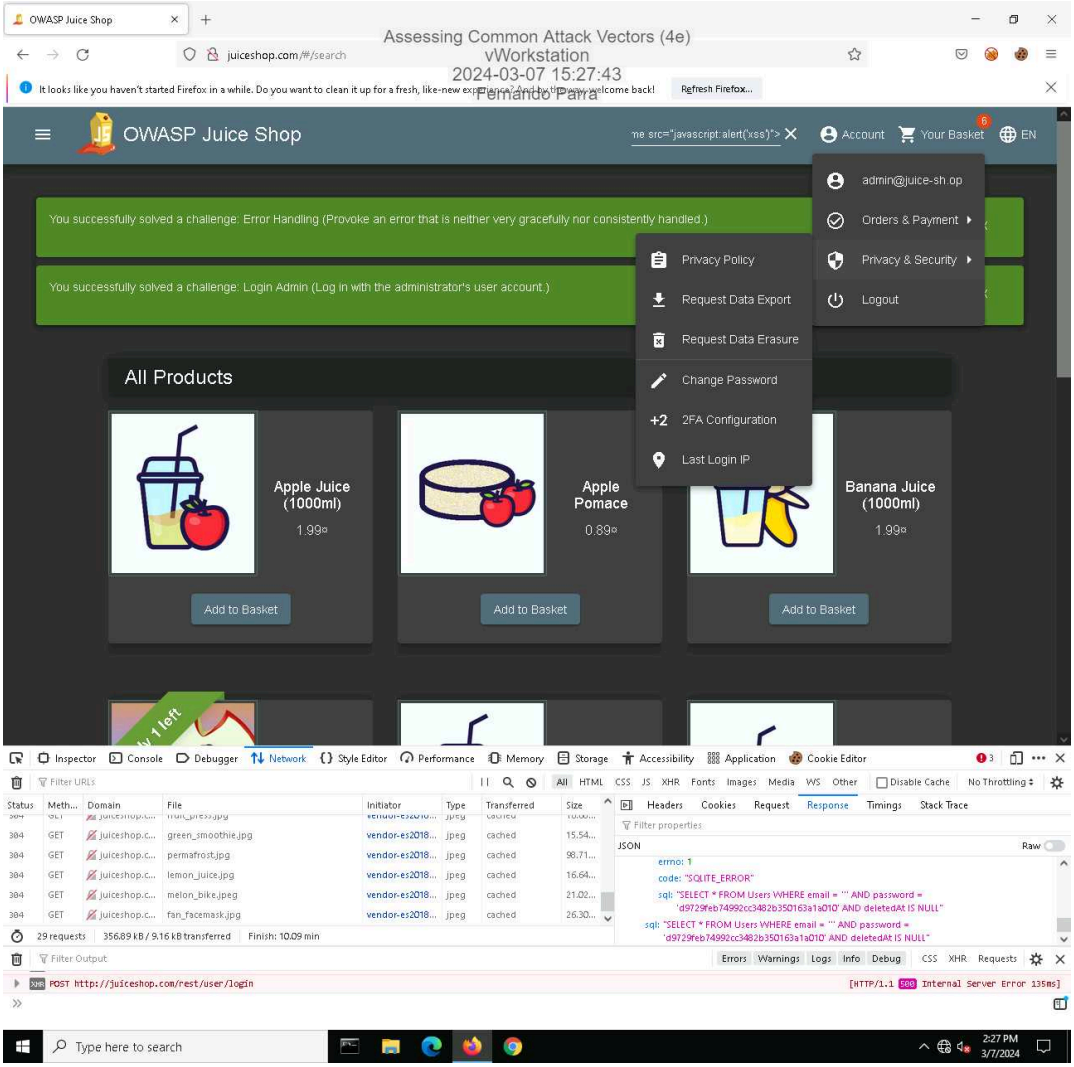
Section 1: Hands-On Demonstration

Part 1: Perform an Injection Attack

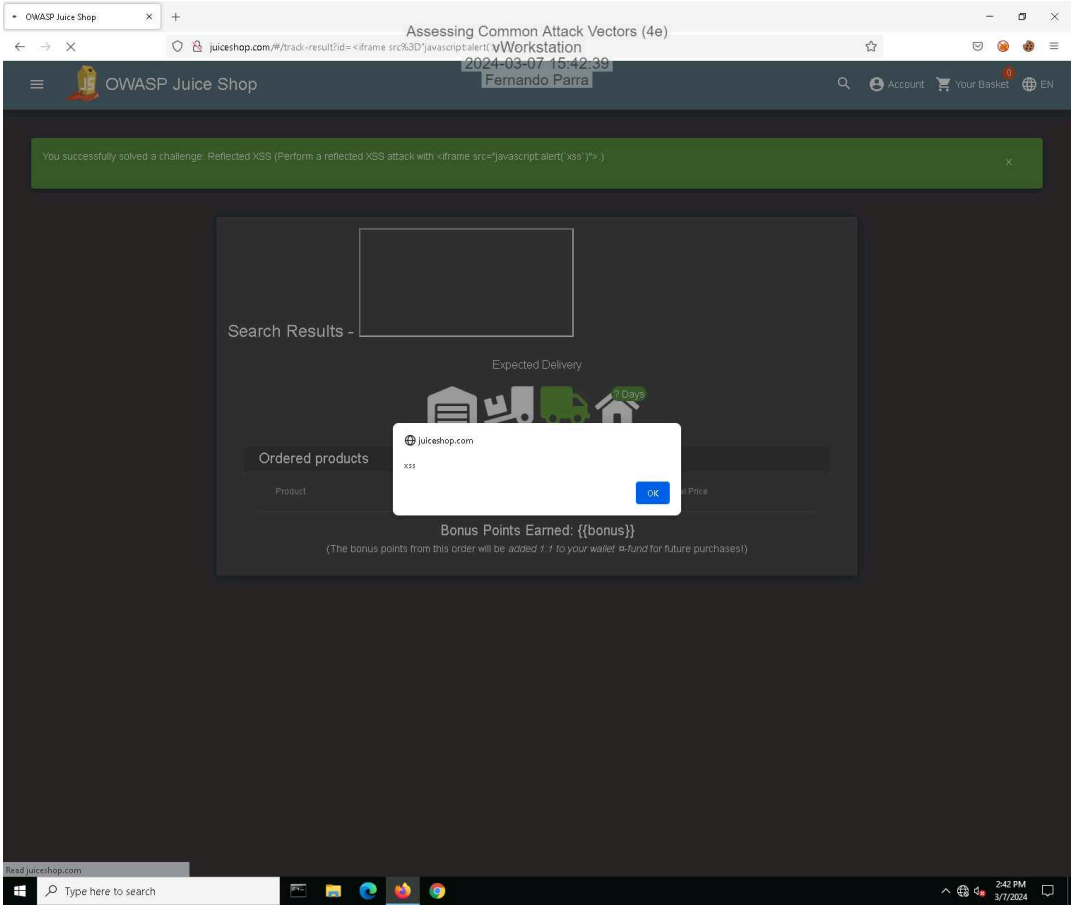
11. **Make a screen capture** showing the **DOM XSS dialog box**.



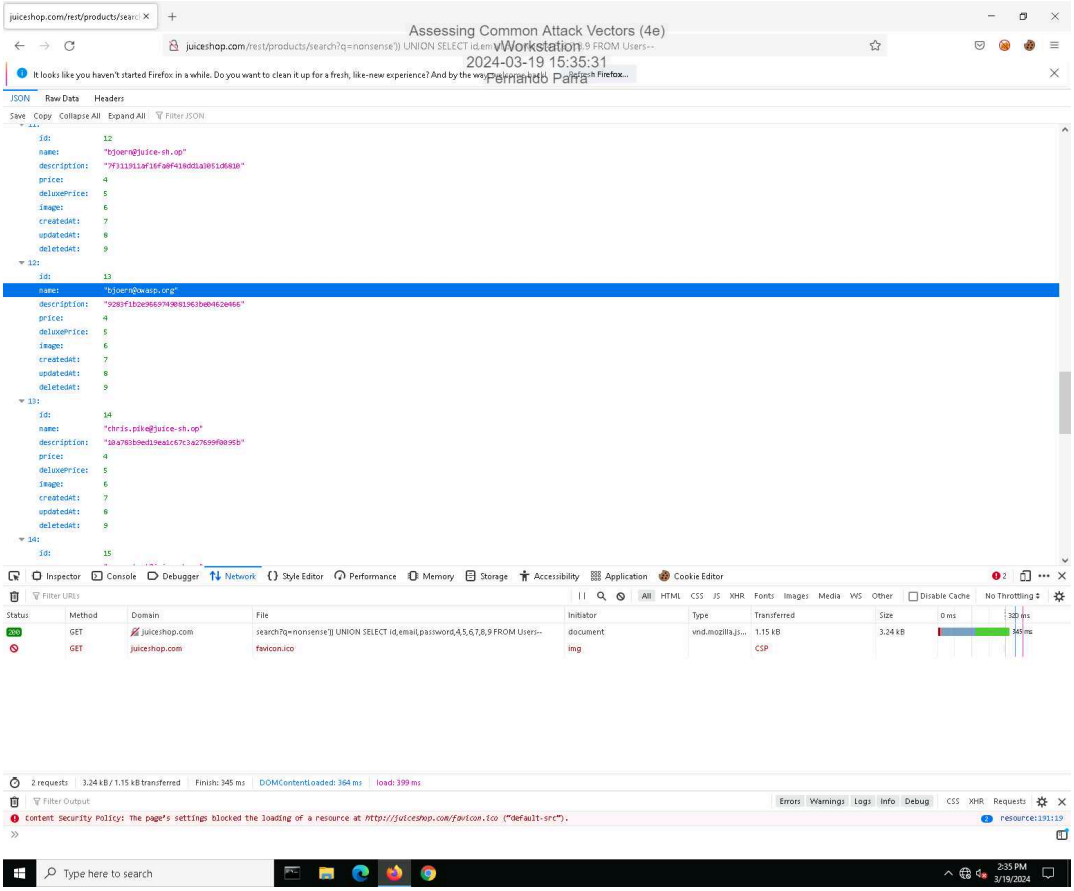
21. **Make a screen capture** showing the **successful admin login**.



26. **Make a screen capture** showing the **successful Reflected XSS injection**.

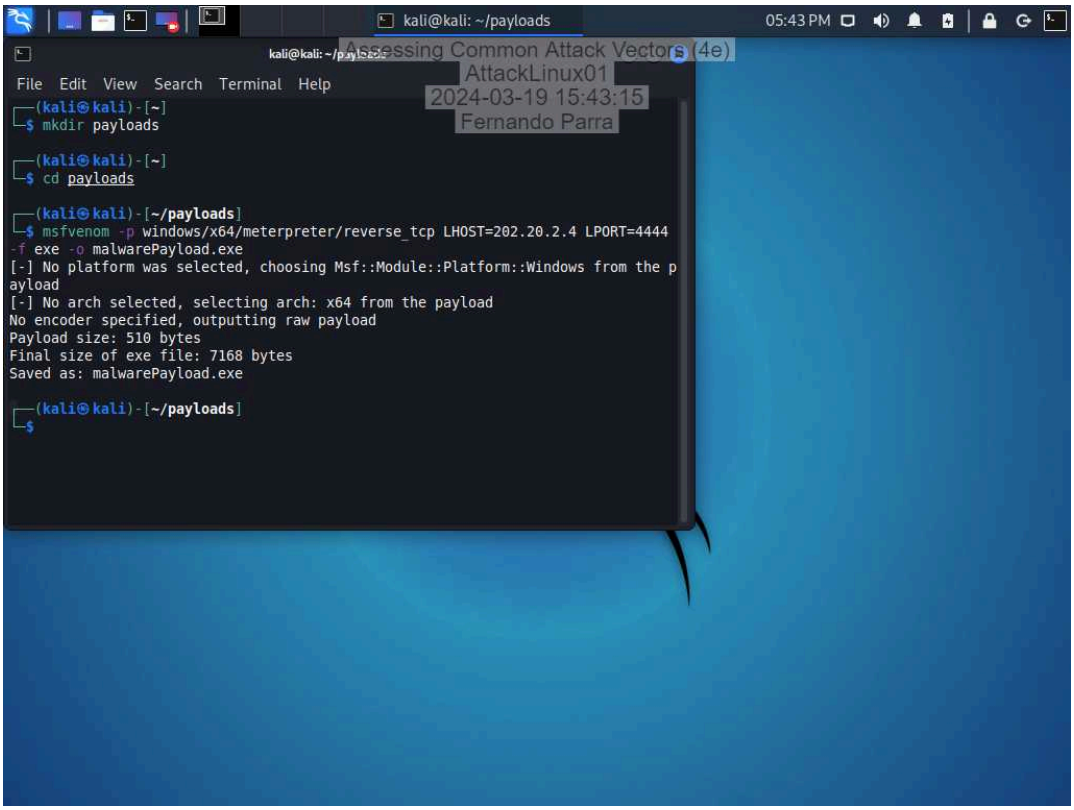


42. **Make a screen capture** showing the **user with the @owasp.org email**.

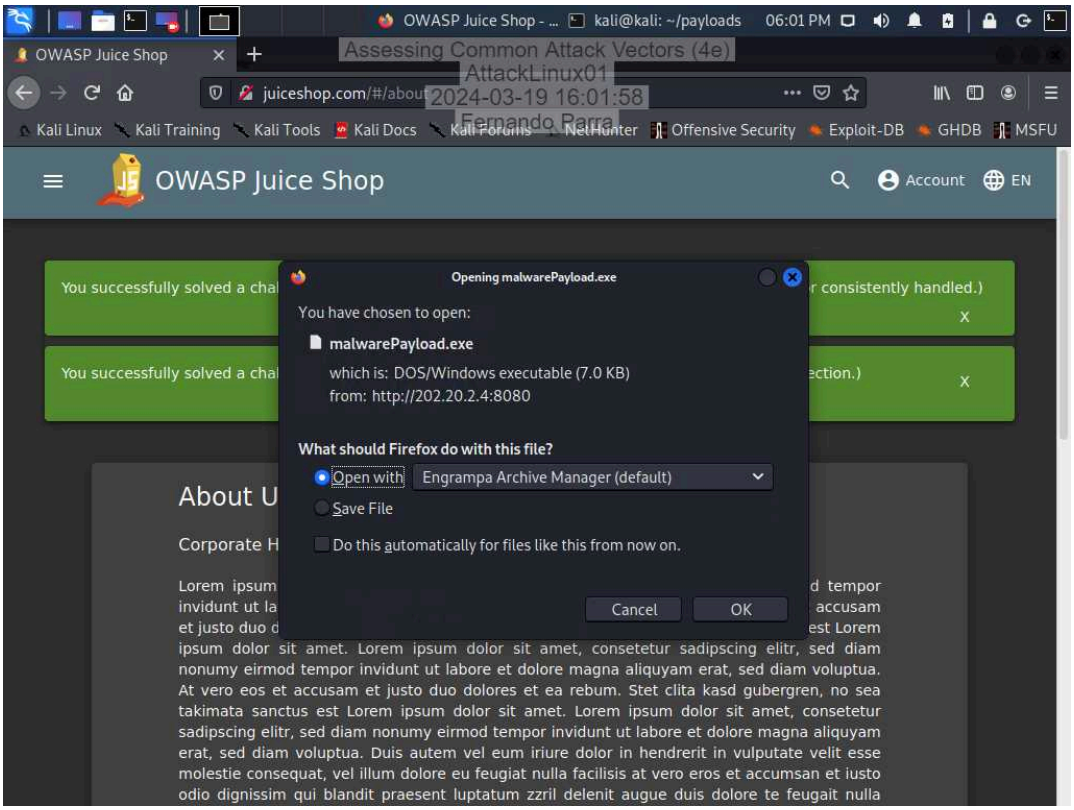


Part 2: Perform a Malware Attack

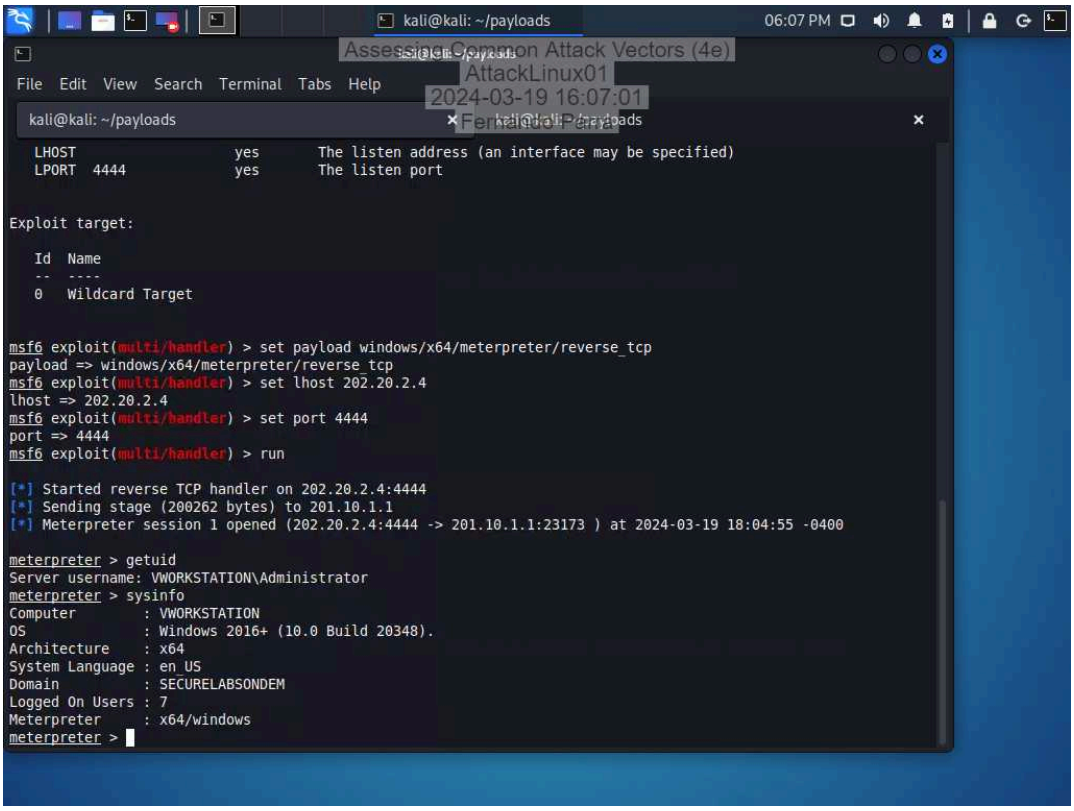
6. **Make a screen capture** showing the **msfvenom output**.



23. **Make a screen capture** showing the **Opening malwarePayload.exe dialog box**.



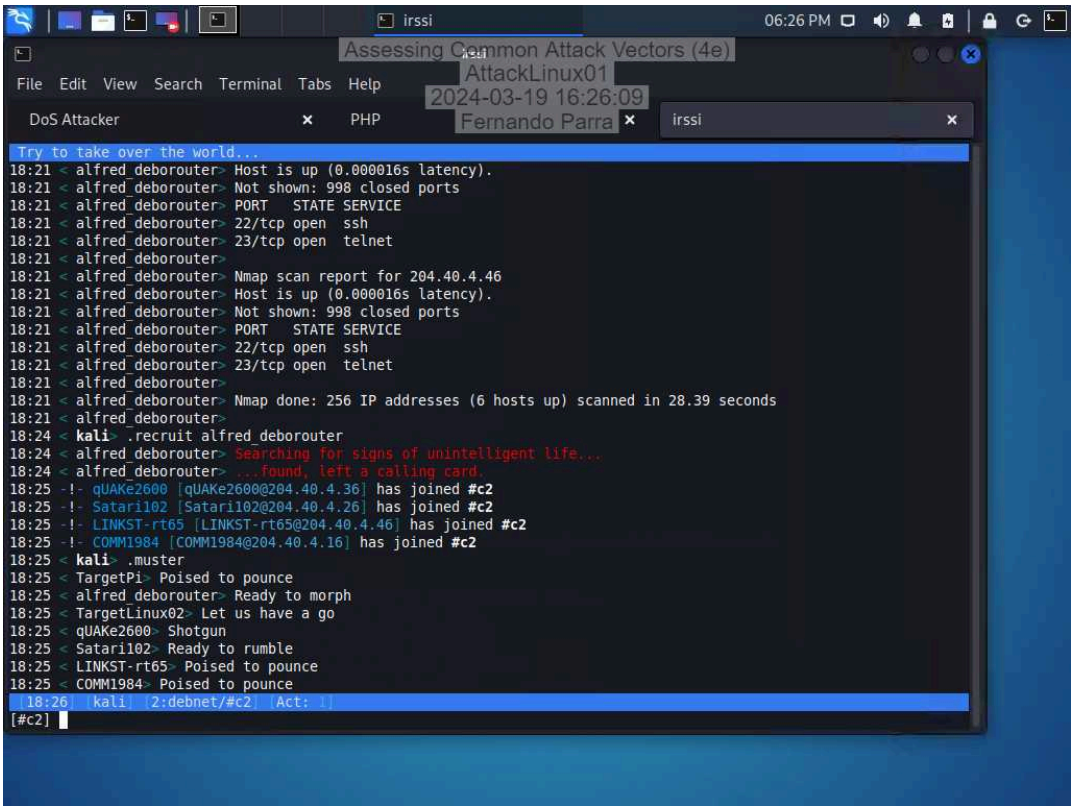
36. **Make a screen capture** showing the **output of the sysinfo command**.



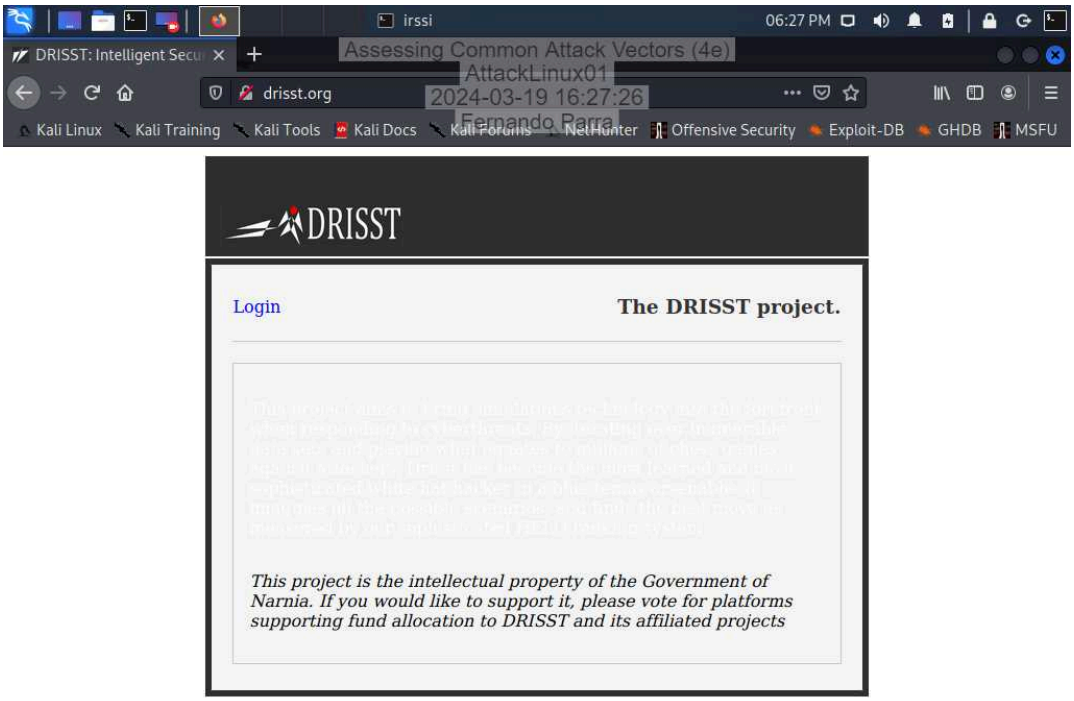
Section 2: Applied Learning

Part 1: Perform a Distributed Denial-of-Service Attack

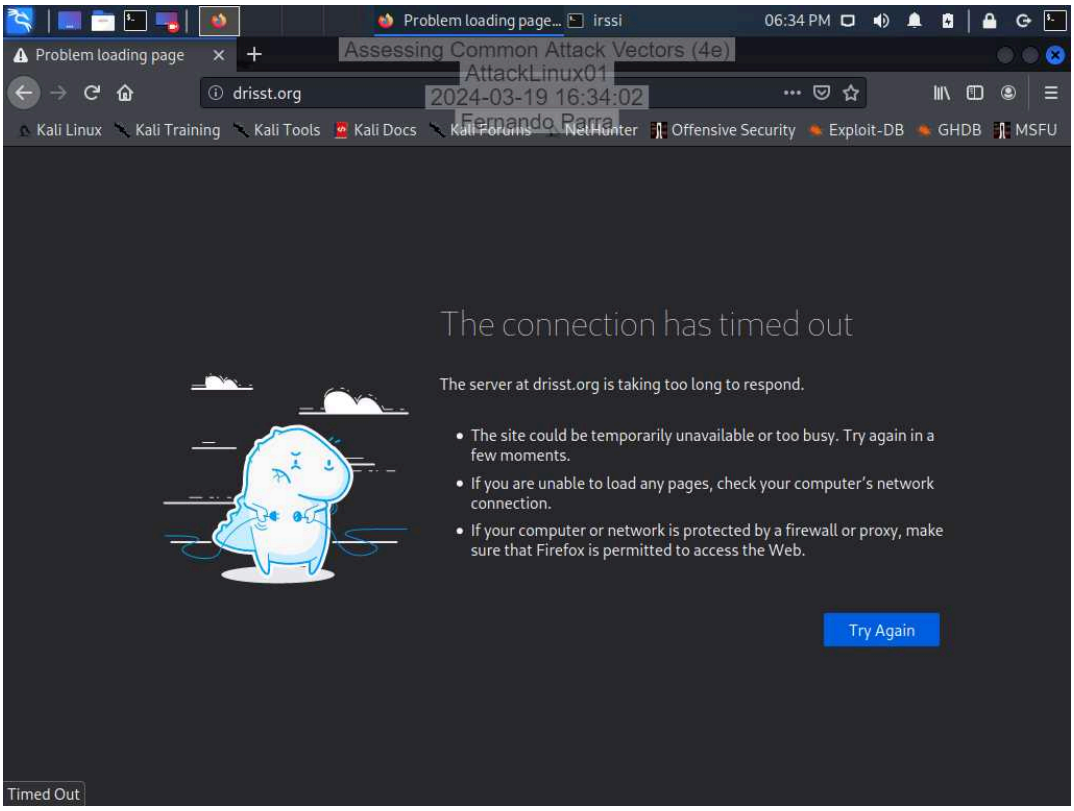
25. **Make a screen capture** showing the **newly recruited hosts**.



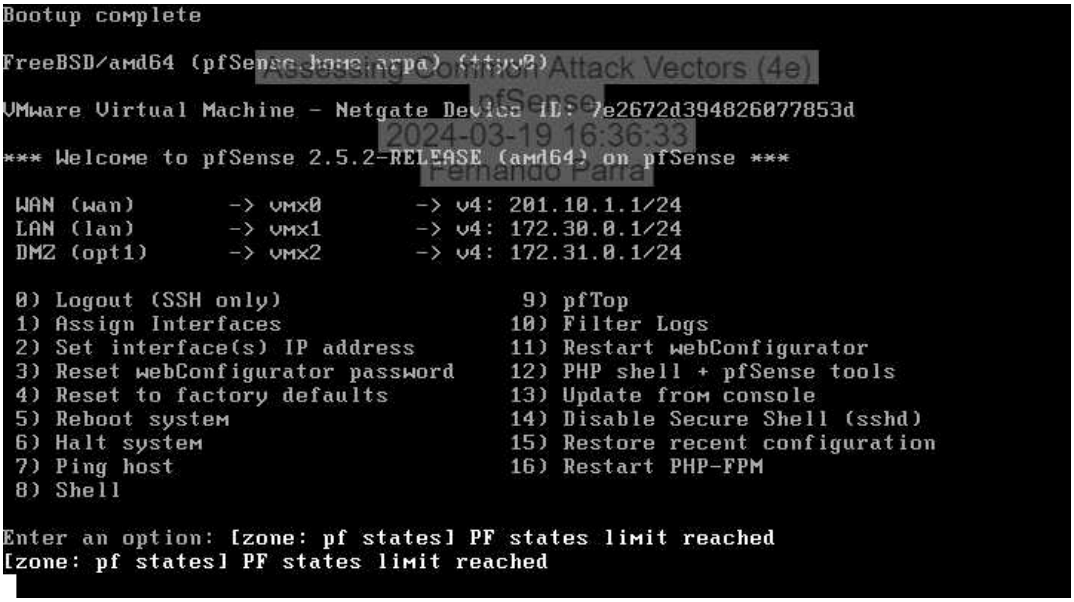
28. **Make a screen capture** showing the **drisst.org webpage**.



33. **Make a screen capture** showing the **failed connection to drisst.org**.

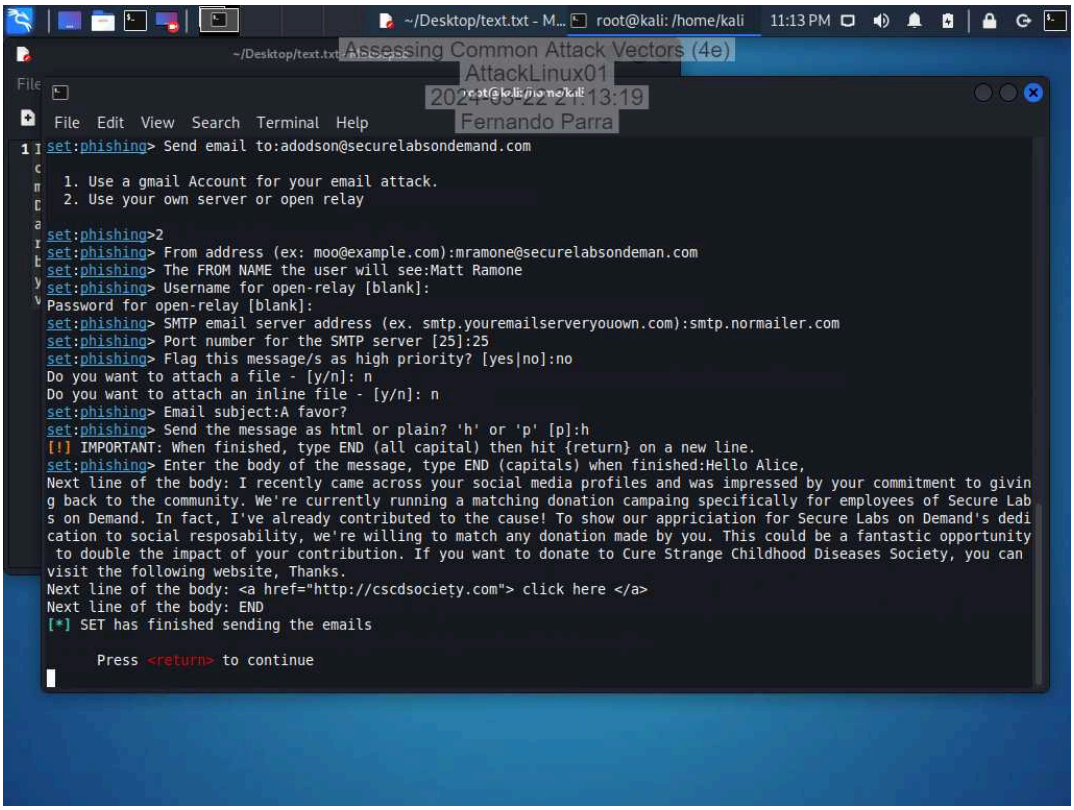


35. **Make a screen capture** showing the **"PF states limit reached" error message**.

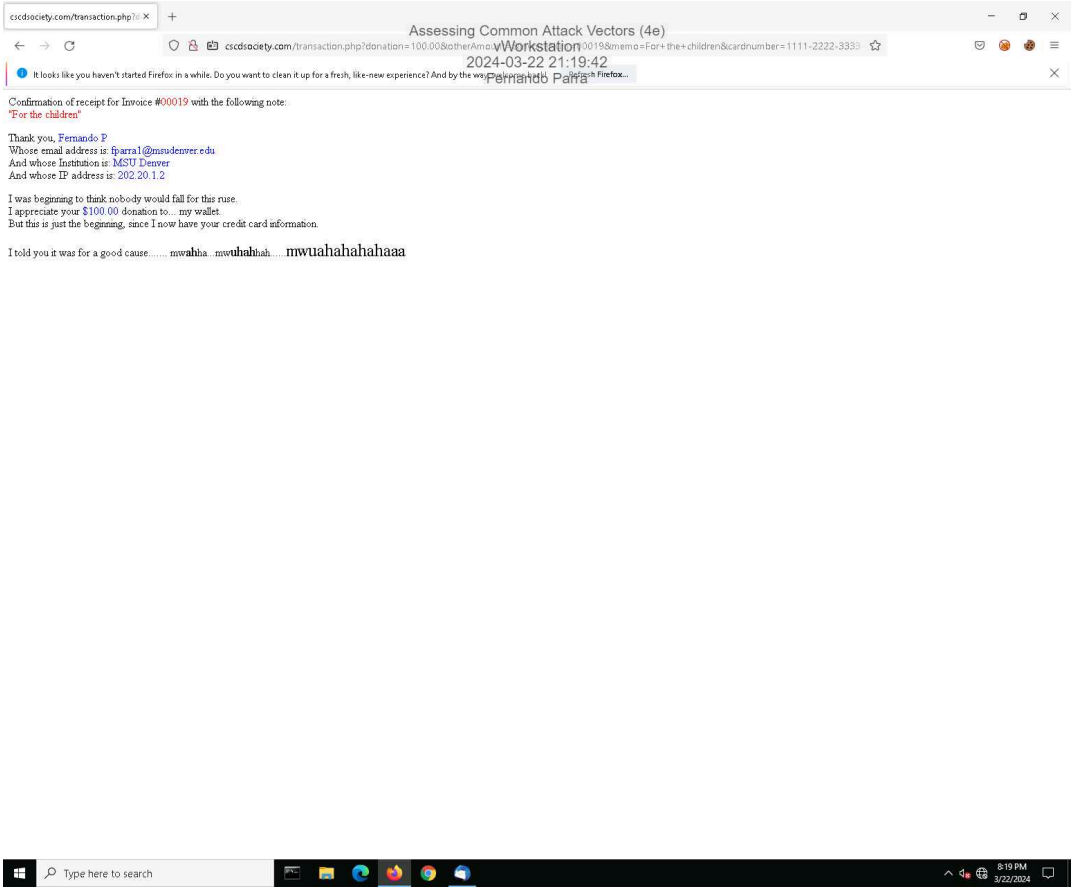


Part 2: Perform a Social Engineering Attack

24. **Make a screen capture** showing the **finished SET phishing email composition**.



36. **Make a screen capture** showing the **transaction.php page in the browser**.



Section 3: Challenge and Analysis

Part 1: Recommend Defensive Measures

Identify and **describe** at least two defensive measures that can be used against injection attacks. Be sure to cite your sources.

Parameterized Queries is a technique that separates data from the actual SQL statement. Instead of embedding the data directly into the query, it is treated as parameters and inserted later, preventing malicious code from being interpreted as part of the SQL statement itself. Another approach to protect against injection attacks is **input validation and sanitization**. This involves inspecting all user input for unexpected characters or commands that could be used for injection. For instance, special characters can be restricted or the format of allowed inputs can be defined. Sanitization can also involve encoding special characters to prevent them from being misunderstood by the application.

Reference
https://stackoverflow.com/questions/446551/from-what-do-sql-parameters-protect-you
https://www.linkedin.com/advice/0/how-can-you-prevent-injection-attacks-web-forms-ay5me

Identify and **describe** at least two defensive measures that can be used against malware attacks. Be sure to cite your sources.

Endpoint Protection Software and Updates are essential to keep your devices safe from malware and other security threats. Installing reputable antivirus and anti-malware software on all devices that can scan for suspicious activity and known malware signatures is highly recommended. This stops malware from infecting a device(s). Keeping this software up-to-date ensures it can detect the latest threats as well. Another security control that helps to prevent malware from damaging your system is **Application Whitelisting**. This restricts which applications are authorized to run on your system. By creating a whitelist of approved programs, any unauthorized software, including malware, will be blocked from executing, preventing it from causing harm.
Reference:
https://www.cisa.gov/news-events/news/understanding-anti-virus-software
https://www.cisa.gov/news-events/news/protecting-against-malicious-code#:~:text=Install%20and%20maintain%20antivirus%20software,in%20preventing%20and%20detecting%20infections.
https://www.crowdstrike.com/cybersecurity-101/application-whitelisting/
https://csrc.nist.gov/pubs/sp/800/167/final

Identify and **describe** at least two defensive measures that can be used against denial-of-service attacks. Be sure to cite your sources.

To protect against denial-of-service (DoS) attacks, two defensive measures can be implemented: **rate limiting** and **Content Delivery Network** (CDN). Rate limiting is a technique that restricts the number of requests that a single IP address or user can send to your server within a specific timeframe. This helps to prevent a flood of malicious traffic from overwhelming your system. Additionally, a CDN distributes your website's content across a network of servers that are geographically dispersed. If a DoS attack targets your main server, the CDN can handle the attack traffic while still delivering content to legitimate users from other servers.

References
https://www.cloudflare.com/learning/bots/what-is-rate-limiting/
https://aws.amazon.com/what-is/cdn/

Identify and **describe** at least two defensive measures that can be used against social engineering attacks. Be sure to cite your sources.

Two crucial measures to protect against social engineering attacks are user awareness training and multi-factor authentication (MFA). User awareness training educates employees on how to identify suspicious tactics, such as urgency, emotional manipulation, and requests for personal information. This knowledge helps employees to be more cautious and question communications before taking any action. MFA provides an additional layer of security by requiring a second verification factor, such as a code from a phone app, along with a password. This makes it significantly more difficult for attackers to access accounts, even if they manage to trick someone into revealing their password.

References
https://www.linkedin.com/pulse/role-employee-training-cybersecurity-risk-management-brian-kimathi
https://www.cisa.gov/MFA

Part 2: Research Additional Attack Vectors

Describe the additional attack vector you selected and **identify** at least two defensive measures that can be used against it. Be sure to cite your sources.

Session Hijacking is a malicious attack where a hacker gains access to an authorized user's session by stealing their session identifier, such as a session cookie. This identifier acts as a verification token between the user's browser and the web server. Once the attacker obtains this identifier, they can pretend to be a legitimate user and access their account and sensitive information without authorization.

HTTP Secure is a protocol that encrypts all communication between a user's browser and the web server. This encryption process scrambles the data, making it unreadable even if intercepted by an attacker. By implementing HTTPS, session identifiers become useless to attackers as they cannot decrypt the stolen information. Additionally, there is a type of cookie called secure cookies that has an additional security flag set. This flag restricts the cookie from being transmitted over unsecured HTTP connections, ensuring that even if an attacker steals the cookie on an unsecured network, they cannot use it to hijack the session on a secure HTTPS connection.

References:
https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/how-to-prevent-session-hijacking-attacks/
https://www.cloudflare.com/learning/ssl/what-is-https/
https://www.securiwiser.com/blog/how-hackers-can-pretend-to-be-you-online-by-stealing-cookies/
https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies