# Experiment 06

**Aim:** Identify the vulnerable website for Clickjacking, CSRF, SSRF

**Tools:** Burp Suite, Port Swigger, Kali Linux

**Theory:** Clickjacking, CSRF and SSRF

1. **Clickjacking:** Clickjacking is a technique used by malicious actors to trick users into clicking on something different from what they perceive they are clicking on. This is usually done by overlaying transparent or opaque layers over a legitimate website, with the intention of capturing the user's clicks and interactions. The user may believe they are clicking on a harmless element of the visible page, but they are unwittingly interacting with hidden elements, potentially leading to unintended actions such as installing malware, revealing personal information, or making unauthorized transactions.
2. **CSRF:** CSRF stands for Cross-Site Request Forgery. It is a type of attack where a malicious website tricks a user's browser into making unintended requests to a different website where the user is authenticated. The attacker exploits the fact that the user is logged into a legitimate website, and without the user's knowledge, sends forged requests from the user's browser to the target website. To execute a CSRF attack, the attacker typically lures the victim to visit a malicious website or click on a malicious link. Once the victim is on the malicious page, the attacker's code can automatically make requests to the target website using the victim's active session, potentially causing unwanted actions. Protecting against CSRF typically involves techniques such as using anti-CSRF tokens, which are unique tokens generated by the server and embedded in forms or requests. These tokens are then validated upon submission to ensure that the request originates from the legitimate user and not from a malicious third party.
3. **SSRF:** SSRF stands for Server-Side Request Forgery. It is a type of security vulnerability where an attacker can manipulate the server into making unintended requests to other resources on the internet. In an SSRF attack, the attacker typically exploits functionalities of the vulnerable server that allow it to make HTTP requests to other servers. By manipulating parameters or input fields, the attacker tricks the server into sending requests to internal systems, other servers on the internet, or services that should not be accessible from the server. For example, an attacker might use SSRF to make the server request internal resources or even attack other systems within the same network that are not supposed to be accessible externally. To prevent SSRF attacks, it is essential to validate and sanitize user inputs, restrict the server's ability to make requests to external resources, and implement proper access controls and firewall rules to limit the server's reach.

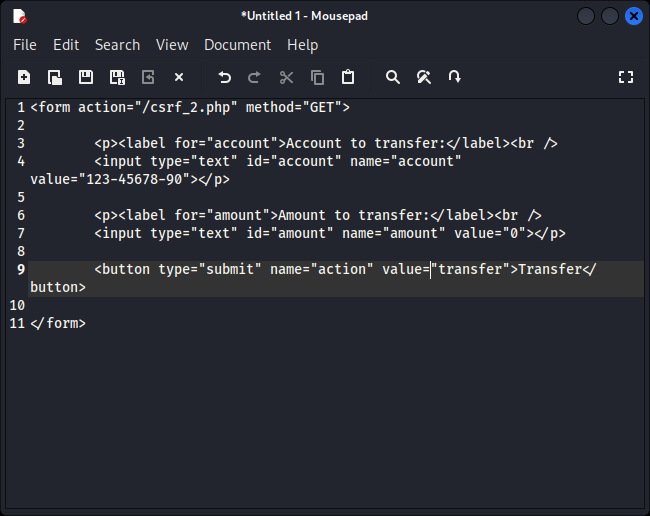
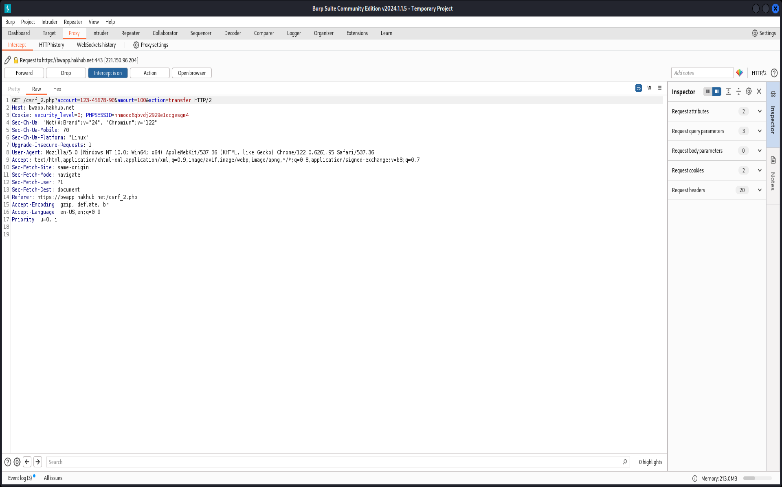
Identifying specific vulnerable websites for security exploits such as Clickjacking, CSRF, and SSRF is not appropriate or ethical. These vulnerabilities can exist in any web application, regardless of its size or popularity. It is essential for website developers and administrators to be aware of these vulnerabilities and take steps to mitigate them through proper security practices, such as input validation, access controls, and implementing security headers. However, it's worth noting that many websites have historically been vulnerable to these types of attacks, leading to widespread exploitation by malicious actors. Security researchers and ethical hackers often discover vulnerabilities in popular websites and report them to the respective organizations to prompt fixes and enhance security measures.

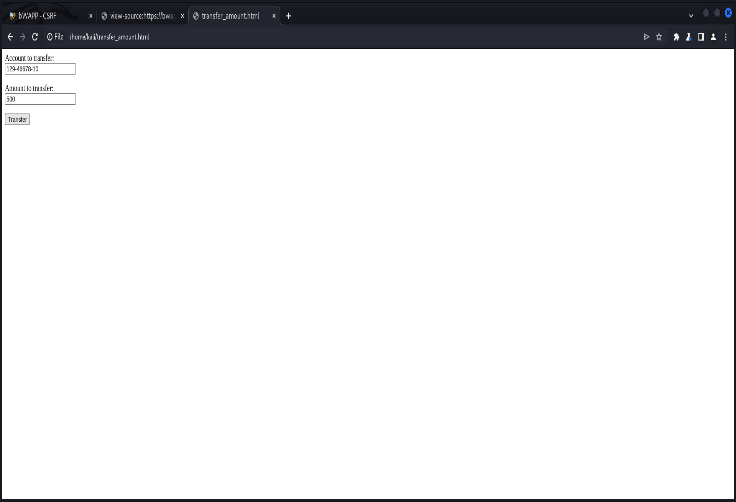
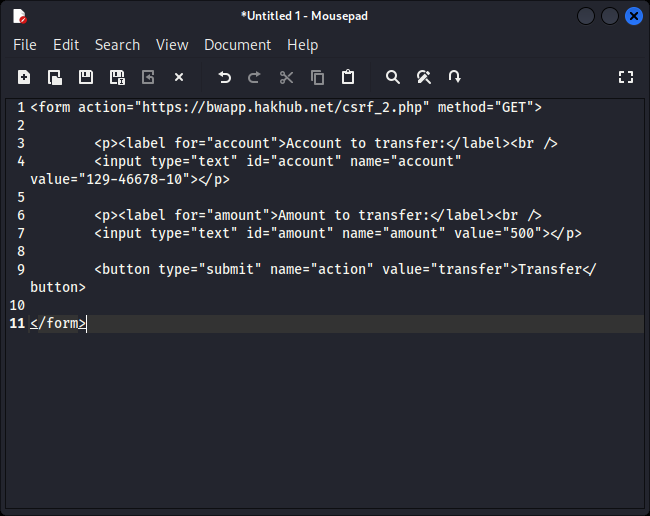
# Implementation:

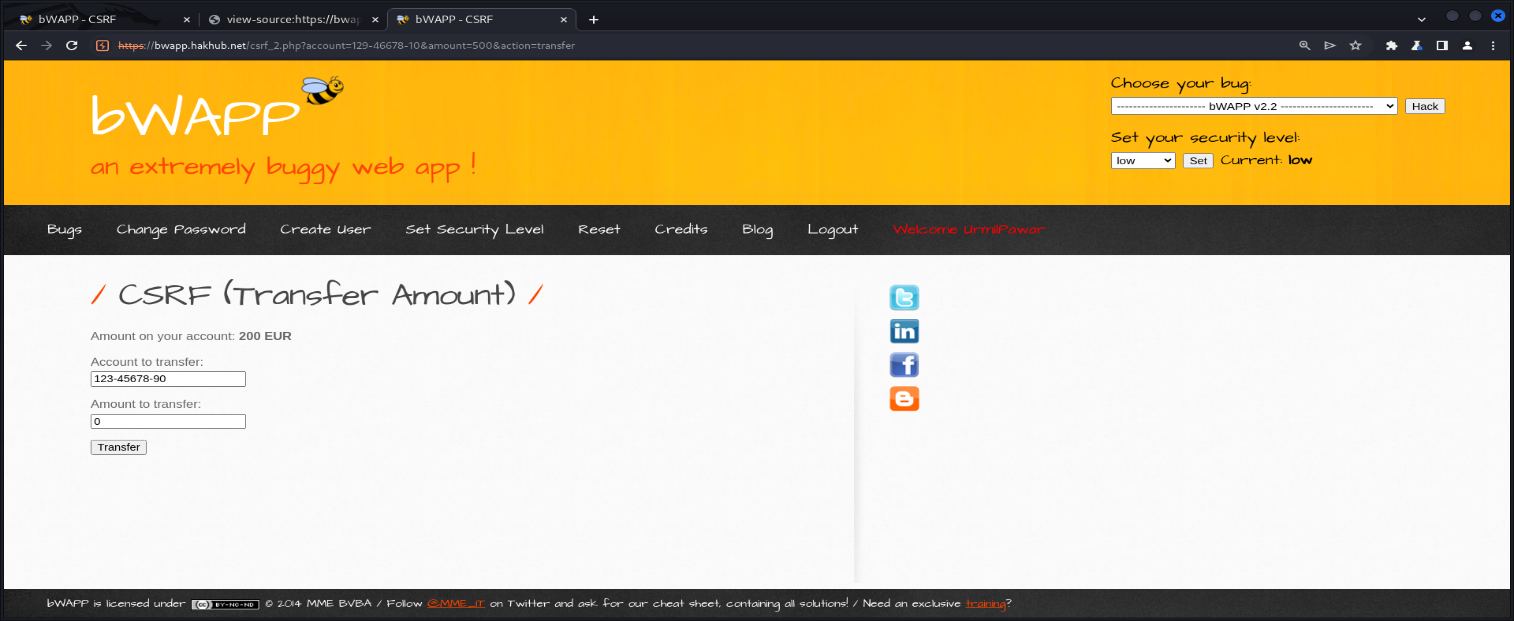
# 1. Clickjacking

# 2. CSRF

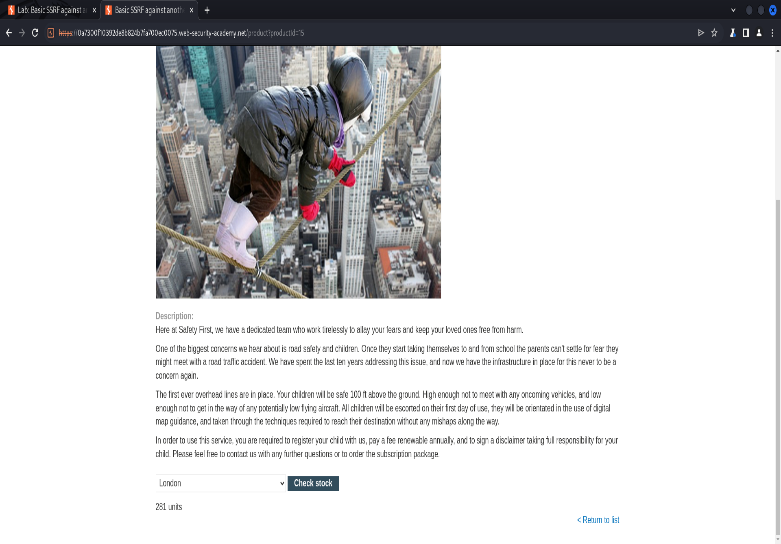
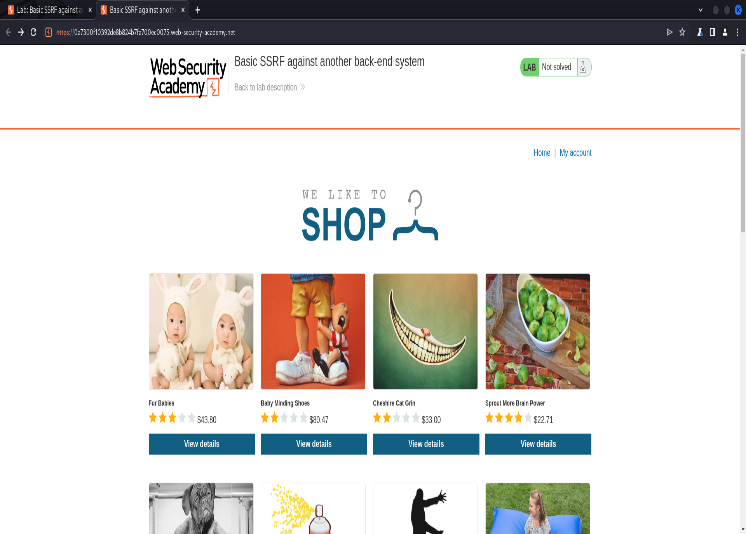
# 

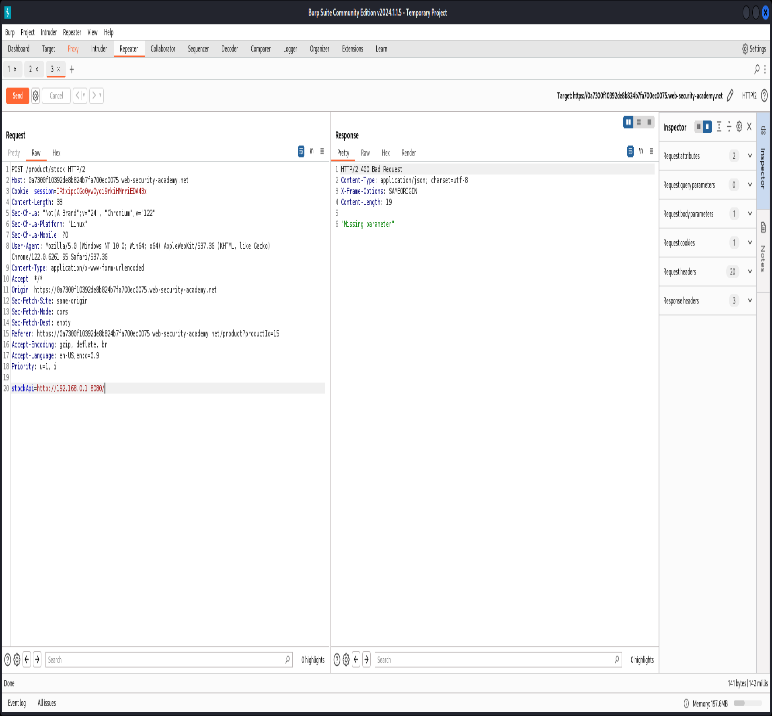
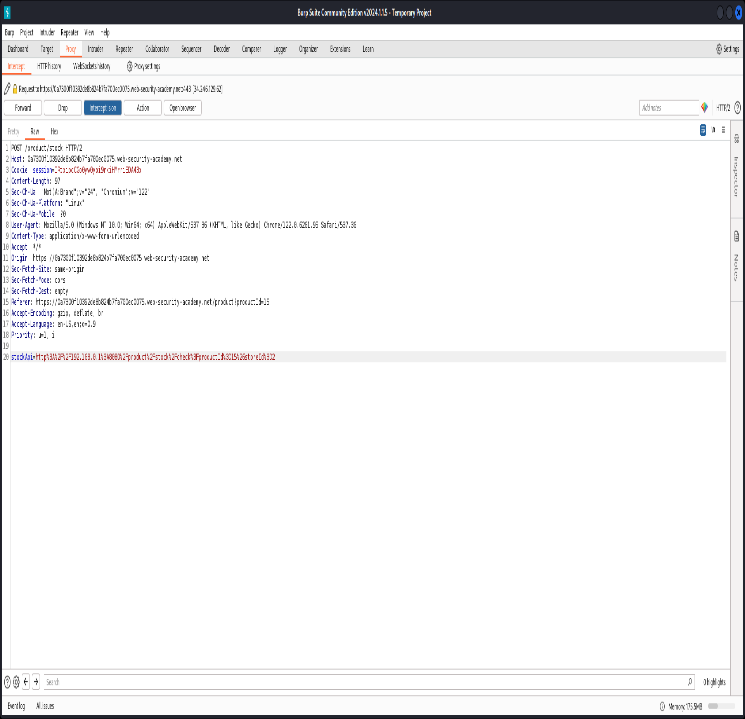


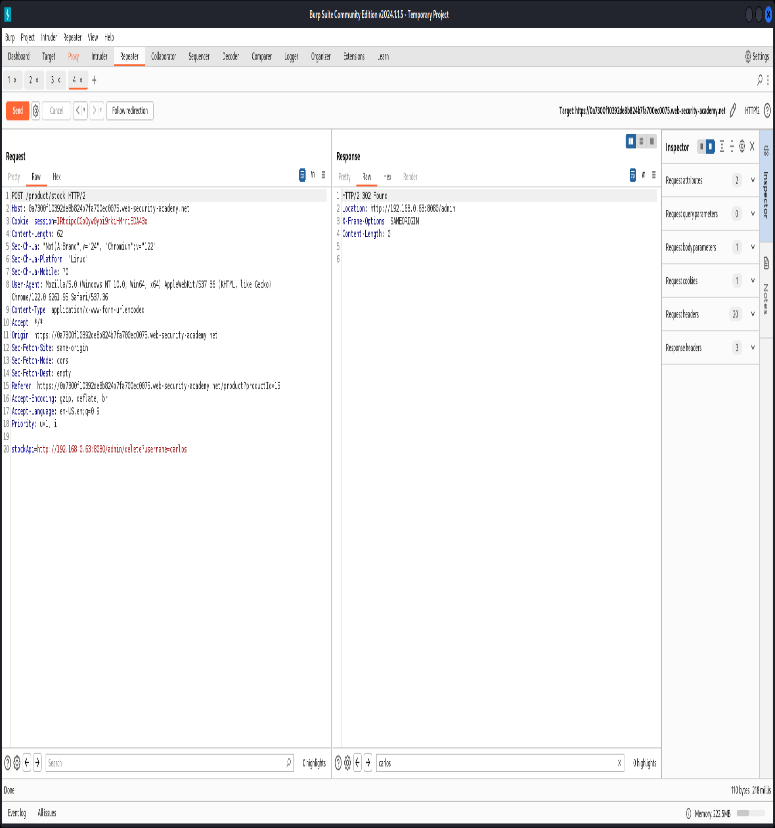
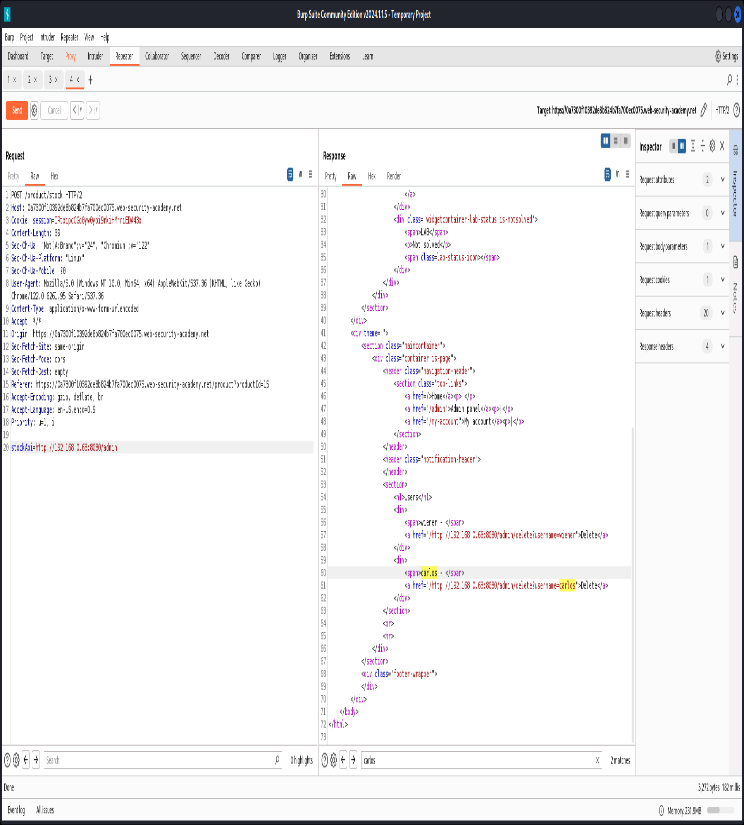


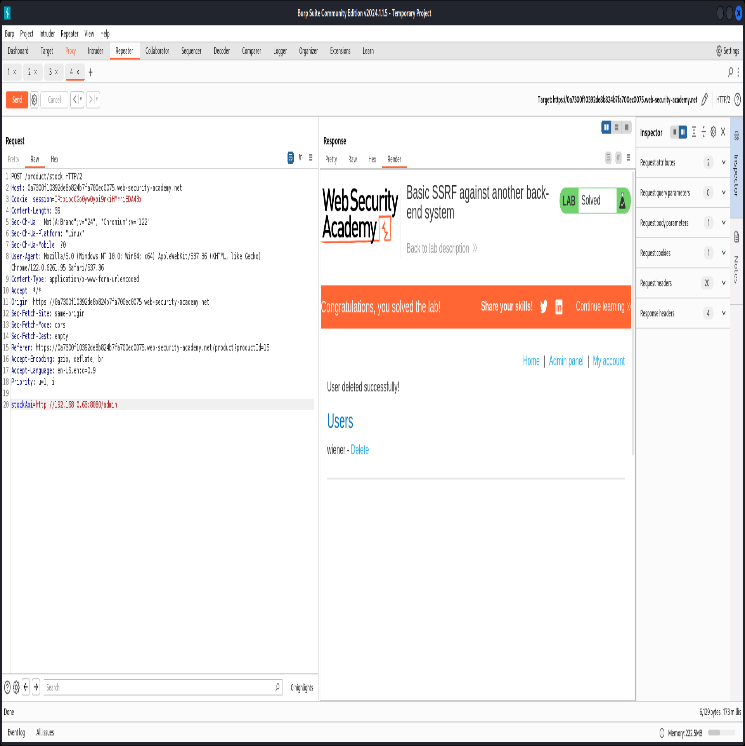
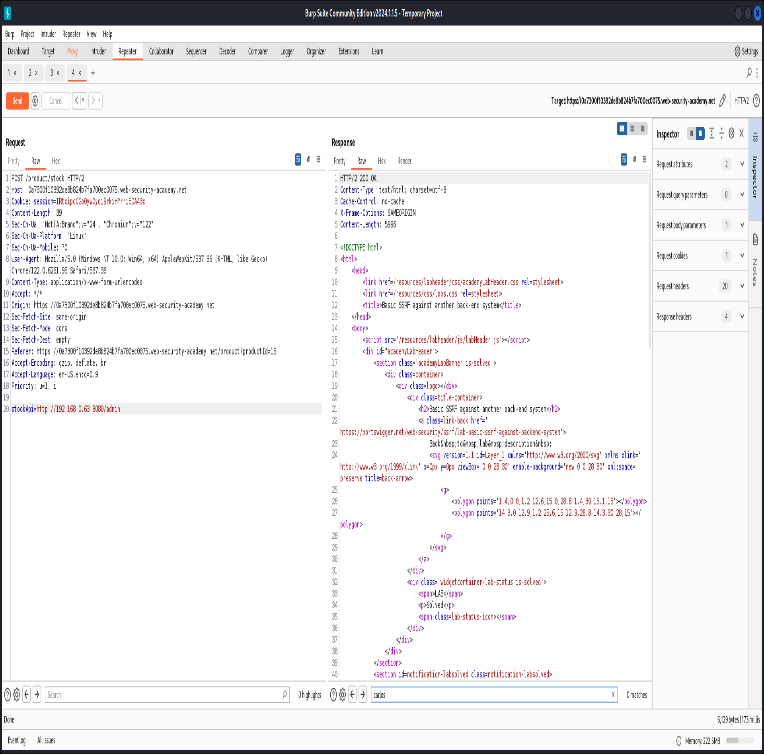


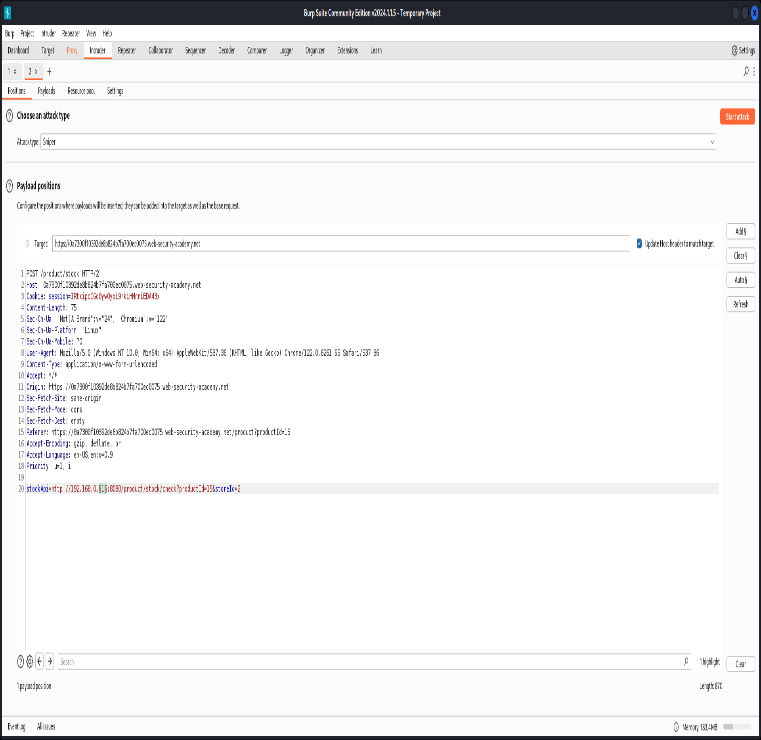
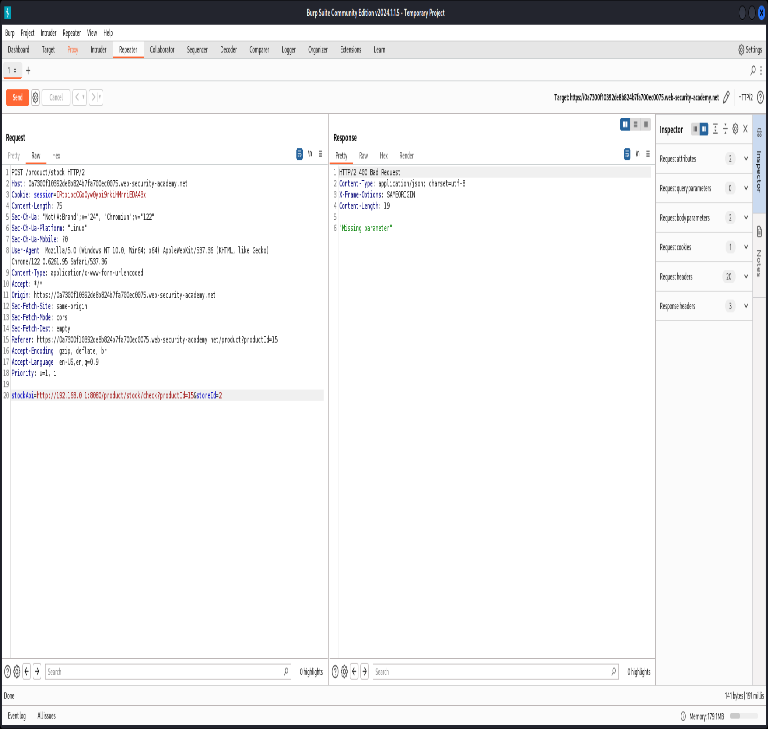
3. SSRF

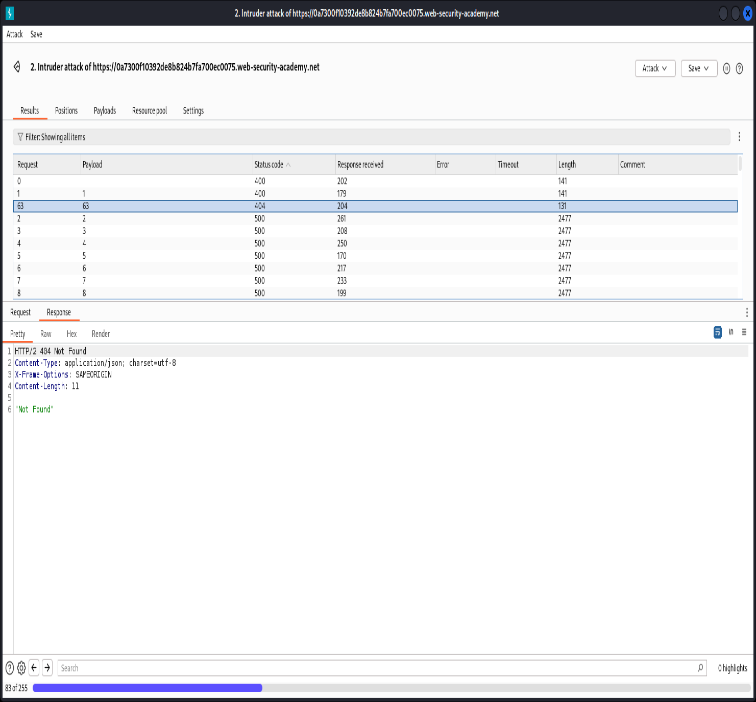
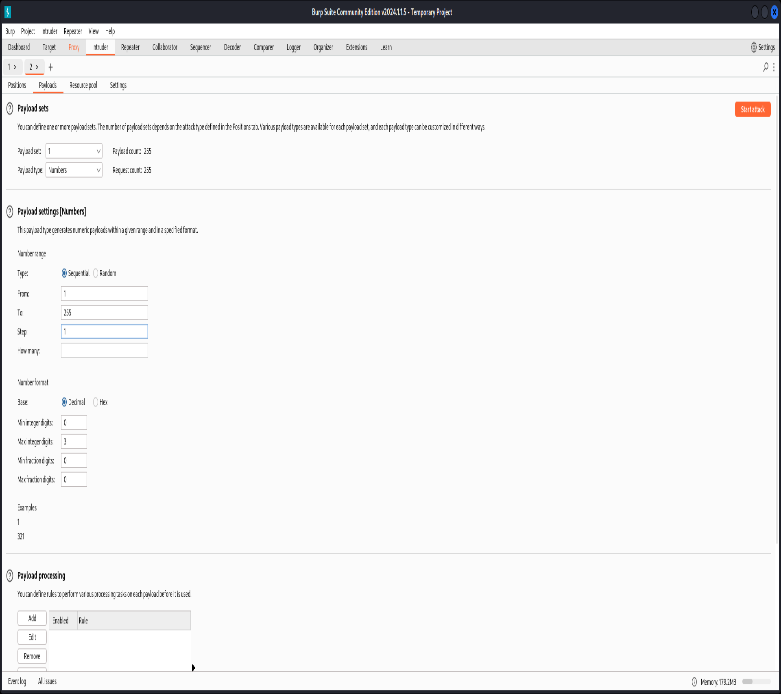












**Result and Discussion:**

1. Understanding Vulnerabilities: Research could focus on understanding the technical aspects of Clickjacking, CSRF, and SSRF vulnerabilities, including how they work, their potential impact on web applications, and real-world examples of their exploitation.
2. Tools and Techniques: Discussion could centre around the use of tools like Burp Suite, Port Swigger, and Kali Linux for identifying and exploiting these vulnerabilities. This could include exploring the functionalities of these tools, techniques for using them effectively, and best practices for conducting ethical hacking activities.
3. Mitigation Strategies: Discussion could focus on effective mitigation strategies for preventing Clickjacking, CSRF, and SSRF attacks. This could include techniques such as input validation, output encoding, implementing security headers, using anti-CSRF tokens, and network-level protections.
4. Real-World Impact: Discussion could explore the real-world impact of Clickjacking, CSRF, and SSRF vulnerabilities on organizations and individuals. This could include case studies of security breaches resulting from these vulnerabilities, the financial and reputational damage incurred, and lessons learned for improving security posture.

# Learning Outcomes:

1. Understanding of Vulnerabilities: By exploring vulnerabilities such as Clickjacking, CSRF, and SSRF in a controlled setting, individuals can gain a deeper understanding of how these vulnerabilities work and their potential impact on web applications.
2. Hands-On Experience with Tools: Using tools like Burp Suite, Port Swigger, and Kali Linux provides practical experience in conducting security assessments, penetration testing, and vulnerability analysis. Learners can become proficient in using these tools to identify and mitigate vulnerabilities effectively.
3. Knowledge of Attack Techniques: Individuals can learn various attack techniques and methodologies used by hackers to exploit vulnerabilities, including how attackers manipulate user interactions, forge requests, and abuse server-side functionality.

**Conclusion**:

For Faculty Use

| **Correction Parameters** | **Formative Assessment [40%]** | **Timely**  **Completion of**  **Practical [40%]** | **Attendance / Learning Attitude [20%]** | **Total** |
| --- | --- | --- | --- | --- |
| **Marks**  **Obtained** |  |  |  |  |