**INCIDENT REPORT: BN-18977-User-Login**

**Date: 6/19/2024**

**Executive Summary:** I was tasked with logging into a specific user account (Bender) in the Juice Shop web application by exploiting an existing SQL injection vulnerability in the login form.

**Results**

* **Exploit Successful:** Gained unauthorized access to Bender's account.
* **Vulnerability:** SQL Injection in login form.

**Application Details:**

* OWASP Juice Shop: Latest version
* Burp Suite Community Edition: v2024.4.4.5

**Attack Narrative:**

1. **Intercepted Login Request:** Used Burp Suite to capture the HTTP POST request sent when attempting to log in as Bender.
2. **Configured Intruder Attack:**
   * **Attack Type:** Sniper
   * **Payload Position:** Injected after the email address (e.g., [bender@juice-sh.op](mailto:bender@juice-sh.op)[PAYLOAD])
   * **Payload List:** Auth\_Bypass.txt
3. **Launched Attack:** Initiated the Intruder attack to test the SQL injection payloads.
4. **Identified Successful Payloads:** Observed successful login attempts (200 OK status code) with the payloads listed below. These payloads either terminated the email string, bypassing domain validation, or created always-true conditions, allowing login regardless of the password.
   * **Payloads:**
   * '--
   * ' OR 1=1--
   * ' OR '1'='1
5. **Bypassed Login:** Inserted a successful payload into the intercepted login request and forwarded it. Successfully logged in as Bender without a password.

**Conclusion:**

This exercise demonstrates that a single SQL injection vulnerability can have far-reaching consequences, potentially allowing an attacker to access any user account within the application. It highlights the importance of secure coding practices.

