**Incident Report:** CH-15078-Offensive-CAPTCHA-Exploit

**Date:** 10-20-2022

**Executive Summary:**

Using OWASP Juice Shop application and Burp Suite, submit 10 or more customer feedback forms (HTTP Requests) within 20 seconds to exploit the broken anti-automation features of the web app.

**Methodology:**

To complete this task successfully, I took the following actions, in order:

1. Researched the basics of what CAPTCHA is, its functionality, and the ways in which it adds a layer of security to web applications
   1. Learned a bit about their vulnerabilities and the ways in which they can be exploited by attackers
2. Based on this information, it became clear that the tool to use for this challenge was Burp Intruder:
   1. Used Portswigger and Github as resources to learn more about how Burp Intruder works:
      1. Attack types
      2. Payloads and positions
3. Navigated to OWASP Juice Shop contact page in browser
4. Opened Burp Suite:
   1. Used default settings to start new project
   2. Turned Intercept ON
5. Back in web browser:
   1. Completed an initial customer feedback form submission within the Juice Shop app:
      1. Author: anonymous
      2. Comment: great
      3. Rating: 5
      4. CAPTCHA Result: 21
   2. Clicked submit button to send HTTP POST Request to the web server
6. In Burp Suite:
   1. Selected POST request from HTTP History in Proxy tab
      1. Right-click → Send to Repeater
   2. Within Repeater tab:
      1. Reviewed the details of the HTTP Request and Response:
         1. HTTP Request parameters returned a Response status of “success”
            1. Indicated that the identifier payloads submitted were valid
      2. Clicked send button to resend original HTTP Request:
         1. Wanted to test what would happen if these same set of values in the HTTP Request was submitted again - Would the Response status still show as “success”?
      3. Results showed that this second Request was also successful
         1. Concluded that it was unnecessary to test additional Request parameters in order to analyze their returned Responses, based on the goals of this particular challenge
      4. Right-click → Send to Intruder
      5. In Intruder tab:
         1. In Positions tab:
            1. Clicked “clear” to remove highlighted payload positions

Would not need to test for valid identifiers based on the challenge requirements

* + - 1. In Payloads tab:
         1. Changed Payload type to “Null payloads”

Wanted this attack to repeatedly issue the Base Request unmodified

* + - * 1. In Payload Options:

Set “Generate” to 11

Task required 10+ HTTP Requests be submitted

* + - 1. Clicked “Start attack” button

**Findings/Solutions:**



