Application Security Testing

## Summary

In this lab we are going to use the open source web testing tools OWASP ZAP and Burpsuite CE (already installed on Parrot) to perform a penetration test on the Google hosted pen testing site Gruyere: <http://google-gruyere.appspot.com/part1>

The goal of the pen test is to identify and demonstrate the most impactful finding.

Additional testing sites can be found here: <http://www.amanhardikar.com/mindmaps/Practice.html>

## Tools/Resources

Parrot: Burpsuite and OWASP’s ZAP

## Tasks

1. Start Gruyere <http://google-gruyere.appspot.com/start>
2. Setup a browser-ZAP-Burp-Gruyere proxy chain
3. Explore Gruyere
4. Attack Gruyere
5. Validate a ZAP finding. By replaying a finding request and capturing the altered behavior.

## Submission

Submit to Canvas an evidence report of what you consider the most serious finding to Canvas.  The evidence report should provide all the data needed for a developer to replicate the application flaw.

## Walkthrough

1. Start Gruyere <http://google-gruyere.appspot.com/start>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Start ZAP
   * No persistence needed
   * No updates needed
   * Using the proxy chain (Browser -> 8080 -> ZAP -> 8081 -> Burp -> Gruyere) . Zap will listen on 8080, Under Network -> Connection set HTTP Proxy to localhost:8081

A screenshot of a computer

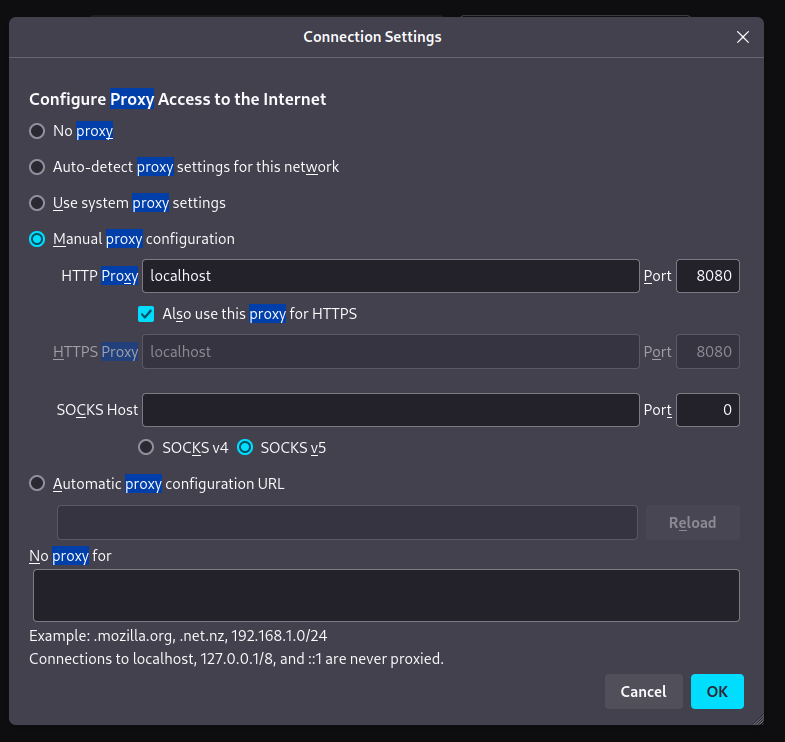
Description automatically generated

1. Start Burpsuite CE
   * When prompted: temporary and use defaults
   * Set listen proxy to 8081 and turn intercept off

A screenshot of a computer program

Description automatically generated

* + Return to browser, set proxy to 8080



* + Refresh Gruyere page and accept the security issue. The TLS connection is to ZAP not Gruyere. You need toad ZAP’s CA to your trusted authorities store. Download and import the CA cert, check allow for websites.

A screenshot of a computer

Description automatically generated

* + Validate requests are shown in both ZAP and Burp history.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* + Attack Gruyere. For the best results, spend some time playing with Gruyere. Login, fill in forms, etc. The more data you provide, even if fake, will inform ZAP. Once you have touched everything in Gruyere. In Zap, select Automated Scan, enter the Gruyere URL, click attack.
  + The attack will run for a few minutes. You should see a couple of dozen alerts with varying degrees of importance.

A screenshot of a computer

Description automatically generated

* + The number of alerts in your scan will be different depending on your interactive browser session.
  + Click on the Alerts tab to expand the findings to see details about specific findings.
  + ZAP alerts are findings until you prove that a request you make will alter the site’s behavior.
  + Expanding the XSS alert you will see the request ZAP sent, the response ZAP received, and guidance about the finding. Replicate the ZAP request in your browser to validate the finding is not a false positive.

https://google-gruyere.appspot.com/480430170828170014080201467021091172211/snippets.gtl?uid=%3C%2Fh2%3E%3CscrIpt%3Ealert(%22To%20fix%20this%20error%20call%20Jenny%20867-5309%22)%3B%3C%2FscRipt%3E%3Ch2%3E

A screenshot of a computer

Description automatically generated

## Additional Thoughts

CSP Cheat sheet: https://cheatsheetseries.owasp.org/cheatsheets/Content\_Security\_Policy\_Cheat\_Sheet.html

Would authenticating to Gruyere find more alerts?

Gruyere does not use SQL, However it is possible that ZAP may report SQLi. Why would ZAP find SQLi when there is no database?

What else can we configure in ZAP to get better results?

BurpSuite can be used to simulate a WAF

* Can a WAF eliminate all possible findings?
* Can WAF cause new findings?
* Does the IT or development team manage the WAF?