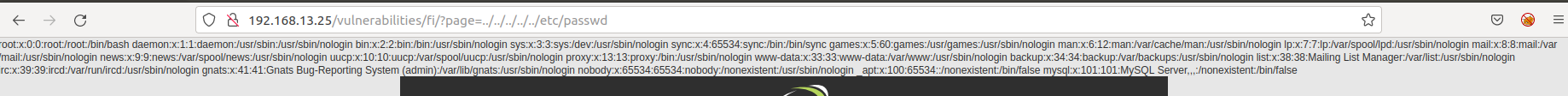
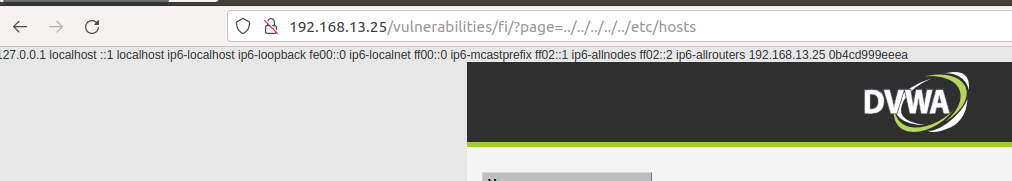
**Unit 15: Homework**

**Web Application 1: Your wish is my command injection:**

* **Command Payload:** *http://192.168.13.25/vulnerabilities/fi/?page=../../../../../etc/passwd*
* **Command Payload:** *http://192.168.13.25/vulnerabilities/fi/?page=../../../../../etc/hosts*

****

***Mitigation:***

* A server-side validation that does not allow selection of unintended directories could be put in place to prevent this injection happening. Other ways such as, Segregation of confidential files from the web server and accessible directories.

**Web Application 2: A Brute Force to be reckoned with:**

**Graphical user interface

Description automatically generated**

***Mitigation*:**

* To prevent a successful attack on an account is by either have stronger passwords, limit login attempts so when its unsuccessful, and a Two-Factor Authentication can be used.

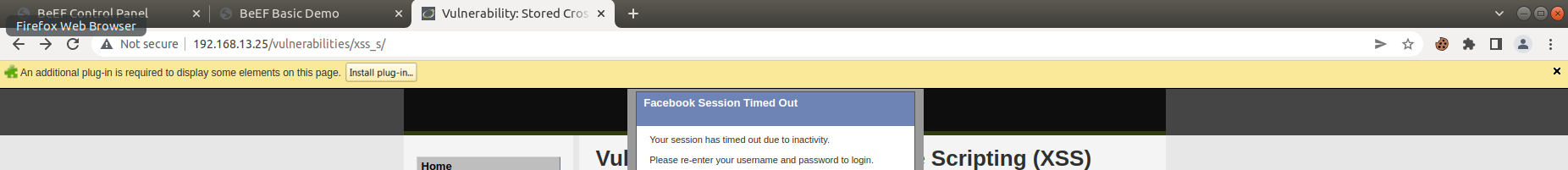
IP address monitoring can also help to see why an account keeps getting locked out.

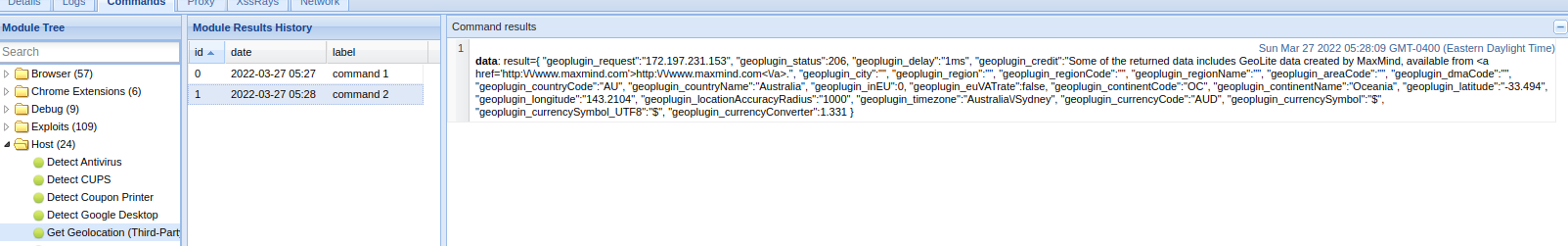
**Web Application 3: Where’s The BeFF:**

* **Social Engineering >> Pretty Theft**

Graphical user interface, application

Description automatically generated

* **Social Engineering >> Fake Notification Bar**
* **Host >> Get Geolocation (Third Party)**



***Mitigation:***

* Some Anti-Virus software application plug into your web page and prevent an exploit from loading. However, most applications can be fooled. Also, google Safe browsing API is used by Firefox and Chrome by default to prevent you from reaching a site.