File upload

**Lab: Remote code execution via web shell upload**

Problem: It doesn't perform any validation on the files users upload before storing them on the server's filesystem.

Goal: upload a basic PHP web shell and use it to exfiltrate the contents of the file /home/carlos/secret. Submit this secret using the button provided in the lab banner.

Solution:

1-

A screenshot of a computer

Description automatically generated

2-A white background with black text

Description automatically generatedA screenshot of a chat

Description automatically generated

3-

A screenshot of a chat

Description automatically generated

4-

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

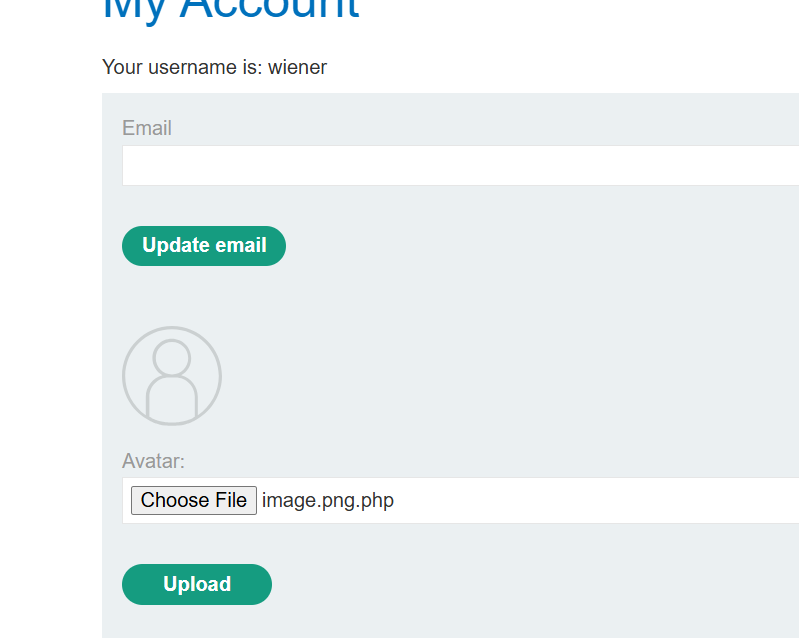
**Lab: Web shell upload via Content-Type restriction bypass**

Problem:  It attempts to prevent users from uploading unexpected file types, but relies on checking user-controllable input to verify this.

Goal: upload a basic PHP web shell and use it to exfiltrate the contents of the file /home/carlos/secret

Solution:

1-



A white background with black text

Description automatically generated

2-A screenshot of a computer

Description automatically generated

3-

A screenshot of a computer

Description automatically generated

4-

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Lab: Web shell upload via path traversal**

Problem:  The server is configured to prevent execution of user-supplied files, but this restriction can be bypassed by exploiting a [secondary vulnerability](https://portswigger.net/web-security/file-path-traversal).

Goal: upload a basic PHP web shell and use it to exfiltrate the contents of the file /home/carlos/secret. Submit this secret using the button provided in the lab banner.

Solution:

1-

A black screen with a white text

Description automatically generated

2-

A screenshot of a computer

Description automatically generated

3-

A screenshot of a computer

Description automatically generated

4-encoding🡪 ../exploit.php

A screenshot of a computer

Description automatically generated

5-

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

**Lab: Web shell upload via extension blacklist bypass**

Problem:  Certain file extensions are blacklisted, but this defense can be bypassed due to a fundamental flaw in the configuration of this blacklist.

Goal:  upload a basic PHP web shell, then use it to exfiltrate the contents of the file /home/carlos/secret. Submit this secret using the button provided in the lab banner.

Solution:

1-

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

2-

A screenshot of a computer

Description automatically generated

3-

A screenshot of a computer

Description automatically generated

4-

A screenshot of a computer

Description automatically generated

5-

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Lab: Web shell upload via obfuscated file extension**

Problem: This lab contains a vulnerable image upload function. Certain file extensions are blacklisted, but this defense can be bypassed using a classic obfuscation technique.

Goal: To solve the lab, upload a basic PHP web shell, then use it to exfiltrate the contents of the file /home/carlos/secret. Submit this secret using the button provided in the lab banner.

Solution:

**What is Code Obfuscation?**

As its name suggests, code obfuscation refers to a series of programming techniques designed to disguise elements of a program's code. It's the primary way that programmers can defend their work against unauthorized access or alteration by hackers or intellectual property thieves.

1-

A screenshot of a computer

Description automatically generated

2-

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

**Lab: Remote code execution via polyglot web shell upload**

Problem:  Although it checks the contents of the file to verify that it is a genuine image, it is still possible to upload and execute server-side code

Goal: To solve the lab, upload a basic PHP web shell, then use it to exfiltrate the contents of the file /home/carlos/secret. Submit this secret using the button provided in the lab banner.

Solution:

1-

A screenshot of a computer

Description automatically generated

2-A computer screen shot of a cat

Description automatically generated

3-

A screenshot of a computer

Description automatically generated

4-

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Lab: Web shell upload via race condition**

Problem: Although it performs robust validation on any files that are uploaded, it is possible to bypass this validation entirely by exploiting a race condition in the way it processes them.

Goal:  upload a basic PHP web shell, then use it to exfiltrate the contents of the file /home/carlos/secret

Solution:

1-

A close up of a sign

Description automatically generated