



# HackerFrogs Afterschool Beginner Cybersecurity Skills

## Web App Exploitation Basics with Natas CTF: Part 1

Class:	Web App Exploitation
Workshop Number:	AS-WEB-01
Document Version:	1.0
Special Requirements:	None

# Table of Contents

<a href="#"><u>Introduction</u></a> .....	3
<a href="#"><u>What is Web Exploitation?</u></a> .....	3
<a href="#"><u>Disclaimer on Web Exploitation</u></a> .....	3
<a href="#"><u>Workshop Guide Format Notes</u></a> .....	4
<a href="#"><u>Brief Description of Capture The Flag (CTF) Games and Natas CTF</u></a> .....	4
<a href="#"><u>Natas CTF Game Information</u></a> .....	4
<a href="#"><u>Part 0 – HTTP Source With Natas 0</u></a> .....	5
<a href="#"><u>Part 1 – Keyboard Shortcuts With Natas 1</u></a> .....	8
<a href="#"><u>Part 2 – Directory Indexing With Natas 2</u></a> .....	10
<a href="#"><u>Part 3 – Robots.txt With Natas 3</u></a> .....	13
<a href="#"><u>Part 4 – HTTP Referer Headers With Natas 4</u></a> .....	15
<a href="#"><u>Part 5 – Setting Cookies With Natas 5</u></a> .....	18
<a href="#"><u>Summary</u></a> .....	24
<a href="#"><u>Extra Credit</u></a> .....	24
<a href="#"><u>Extra Research</u></a> .....	25

# Introduction

Welcome to HackerFrogs Afterschool! HackerFrogs Afterschool is a series of online workshops meant to teach cybersecurity skills and concepts to beginners who may not be familiar with the field of cybersecurity. This lesson is the introductory lesson of web app exploitation, which is a very prominent and in-demand specialization in cybersecurity. This lesson will cover the following learning objectives:

- Interact with the web applications using web browser software.
- Interact with web applications using the cURL program
- Read relevant information from web page source code

## What is Web Exploitation?

Web exploitation is the intentional abuse of systems and applications accessed via the internet by a user, which benefits that user in some way. This benefit can take the form of access to unauthorized data or systems, the deletion of data, denial of website services, or any number of other things. In order to learn web exploitation, we must learn about the internet, how websites work, and a good amount about the technologies and software found in that space.

Most students know at least the basics of using a web browser and how to use websites, so we will build on those skills to establish a collection of techniques to explore and exploit web applications found on the internet.

## Disclaimer on Web Exploitation

It is important to note, that at no time during our study of web exploitation will we engage in testing or exploitation of websites or internet spaces that:

- a) we do not own, or
- b) are not intentionally setup as learning environments for web exploitation education

## Workshop Guide Format Notes

When following instructions in this document, there will be instructions to type specific commands in the CLI window or in a search bar. These commands will be displayed in **red**, and will have a different font applied to them, as well as a light grey background color. For example:

```
this is a sample command text line
```

However, if the input is meant to be put into the web browser or other non-CLI program, it will be colored **red**, and **bolded** but will not be in a different font nor will it have a light-grey highlight. For example:

**This is a sample user input text line**

If there are code snippets in this document, they look similar to the format above, except that the text will be black. For example:

```
this is a sample code snippet line
```

## Brief Description of Capture The Flag (CTF) Games and Natas CTF

Capture the Flag (CTF) games are training exercises in the field of cybersecurity. The goal of a CTF exercise is to “capture the flag” through use of cybersecurity skills. In this context, “capture” means to gain access to a file or other resource, and “flag” refers to a secret phrase or password.

The CTF game we will be playing to learn basic web app exploitation skills is called Natas, which is hosted on the OverTheWire CTF network. The OverTheWire network hosts several different CTF games covering different cybersecurity topics, such as cryptography and binary exploitation, among others. The Natas CTF game is made up of many levels of increasing difficulty. The initial levels (0-5) cover the target web app learning objectives for this lesson.

## Natas CTF Game Information

In our web browser, navigate to the following URL:

<https://overthewire.org/wargames/natas/>

# Natas

Natas teaches the basics of serverside web-security.

Each level of natas consists of its own website located at <http://natasX.natas.labs.overthewire.org>, where X is the level number. There is **no SSH login**. To access a level, enter the username for that level (e.g. natas0 for level 0) and its password.

Each level has access to the password of the next level. Your job is to somehow obtain that next password and level up. **All passwords are also stored in /etc/natas\_webpass/**. E.g. the password for natas5 is stored in the file /etc/natas\_webpass/natas5 and only readable by natas4 and natas5.

Start here:

Username: natas0

Password: natas0

URL: <http://natas0.natas.labs.overthewire.org>

There are a few important details on this page:

- 1) Each level of the Natas CTF game has its own website.
- 2) There is no SSH login (remote login) for any of the challenges.
- 3) Access to the level requires the user to enter a username and password.
- 4) Each level of the game contains a password that is used to access the next level of the game.
- 5) Passwords for the next level are also stored on the webserver in the **/etc/natas\_webpass/** directory.
- 6) We're provided with a username, password, and URL for the first level of the game.

Let's start the game with Natas level 0.

## Part 0 – HTTP Source With Natas 0

### Step 1

In our web browser, navigate to the following webpage:

<http://natas0.natas.labs.overthewire.org>

When prompted, enter the following username and password:

Username: **natas0**

Password: **natas0**

## Step 2

Once logged in, there is a hint on the webpage for how to complete the level:

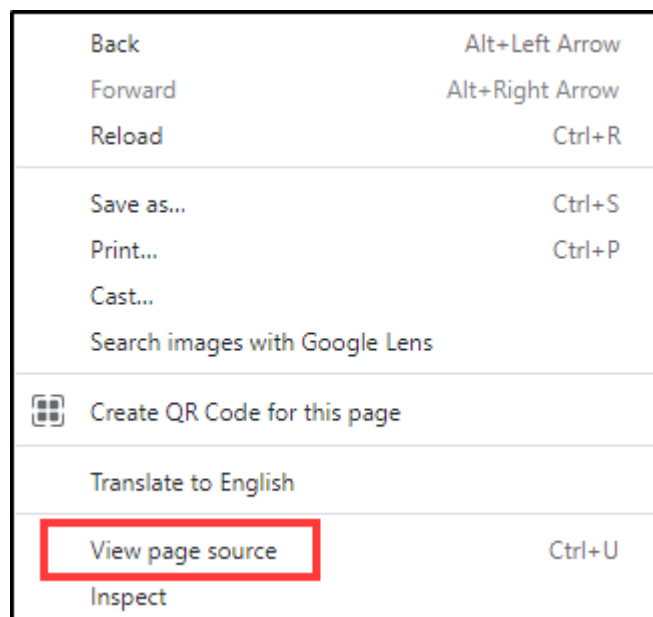
You can find the password for the next level on this page.

In this level, we will learn the concept of web page source code, often just called web page source. The password for the next level is in the source code, but we can't see it right now.

## Step 3

Let's reveal the source code for this web page.

Put your mouse cursor over the white box on the web page and click the right-side button on your mouse. On the menu that appears, click on the **View page source** option. See screenshot below (**View page source** option outlined in red):



## Step 4

A new webpage tab will appear in the browser. In the new page, we can see the web page source code. Even if we don't understand everything in the source code, we should pay attention to one part, which is underlined in red in the screenshot below:

```
11 <body>
12 <h1>natas0</h1>
13 <div id="content">
14 You can find the password for the next level on this page.
15
16 <!--The password for natas1 is gtVrDuiDfck831PqWsLEZy5gyDz1clto -->
17 </div>
18 </body>
19 </html>
```

## CONTEXT

In the HTML language, programmers are able to leave comments in their code that is not displayed when the webpage is loaded. To start a comment in HTML, the programmer inputs these characters:

```
<!--
```

And to end the comment, the programmer inputs these characters:

```
-->
```

Any text that is input between the start and end of the comment will be included in the HTML code as a comment, like the underlined text in the screenshot above. Web page developers often leave comments in their code to make the code easier to understand.

The passwords in the Natas CTF game can also be called “flags”, and in CTF games the objective is to gain access to the game's flag. In the next level (natas 1) we'll use the flag from this level (gtVrDuiDfck831PqWsLEZy5gyDz1clto) to log in.

## SPECIAL PASSWORD NOTE

The passwords for the Natas CTF game levels are changed every few months, so the passwords presented in this document are most likely out of date. Please record the Natas CTF level passwords as you work through the levels.

## Part 1 – Keyboard Shortcuts With Natas 1

### Step 1

In the web browser, navigate to the following webpage:

<http://natas1.natas.labs.overthewire.org>

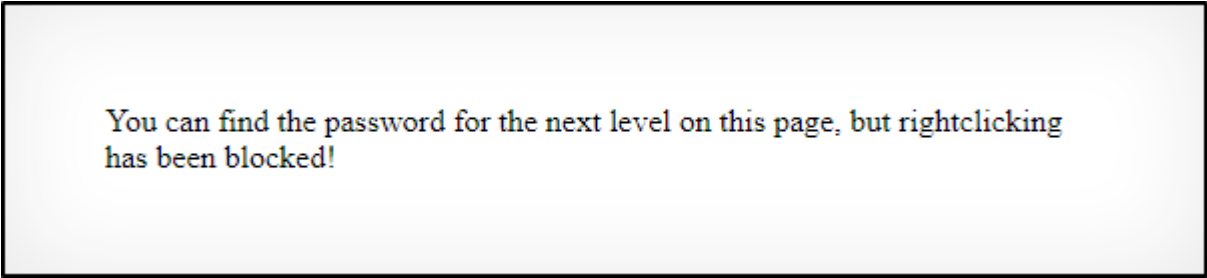
When prompted, enter the following username and password:

Username: **natas1**

Password: obtained from Natas level 0

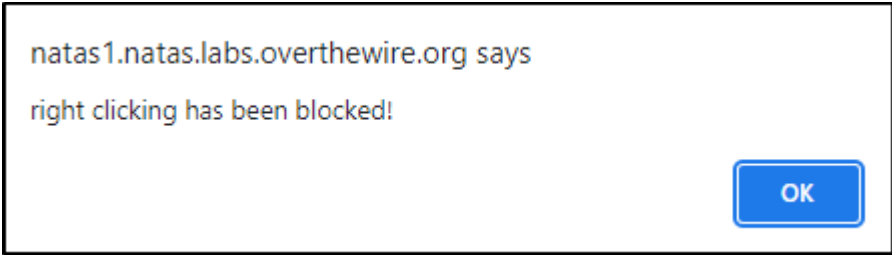
### Step 2

Upon successful login, we can see the following message:



You can find the password for the next level on this page, but rightclicking has been blocked!

If we try the method we used in the previous level to access the webpage source code (right-clicking on the webpage) we will find it doesn't work, and the following message appears:



natas1.natas.labs.overthewire.org says  
right clicking has been blocked!

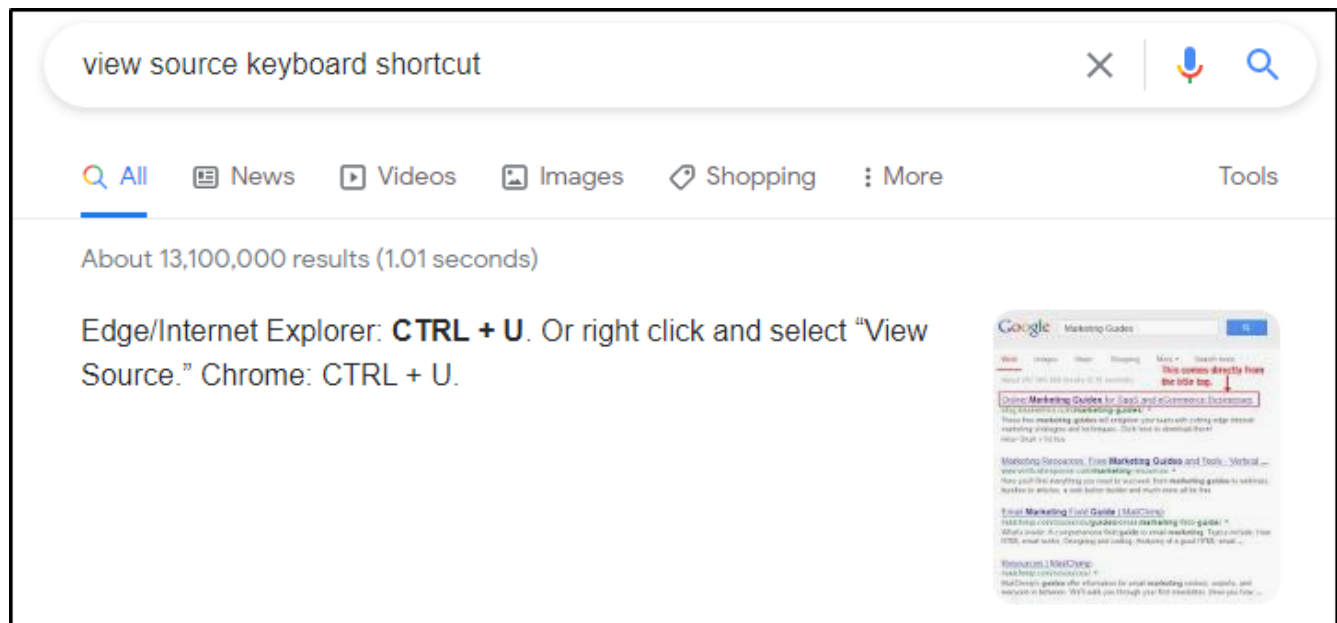
### Step 3

We will need to find an alternative to viewing the source code on the web page. Using a search engine (e.g., **Google**), search for the following search term:

**view source keyboard shortcut**



You should see results similar to the following screenshot:



#### Step 4

On the `natas1` webpage, use the keyboard shortcut **CTRL + U** to access the web page source. In the resulting page, we should see a HTML comment with the password, like the following screenshot (flag text underlined in red):

```
16
17 <!--The password for natas2 is ZluruAthQk7Q2MqmDeTiUiJ2ZvWY2mBi -->
18 </div>
19 </body>
20 </html>
```

## CONTEXT

There are a couple of things to note here. First, when solving CTF challenges, there will often be some obstacle to doing a task in the usual manner, so it's important to keep a flexible mindset when approaching problems. Second, when presented with a problem, it is often required for us to research possible solutions, and the go-to first step to solving technical problems is to do a search engine lookup for terms specific to the problem we're trying to solve (commonly referred to as **Googling**).

## Part 2 – Directory Indexing With Natas 2

### Step 1

In the web browser, navigate to the following webpage:

<http://natas2.natas.labs.overthewire.org>


When prompted, enter the following username and password:

Username: **natas2**

Password: obtained from Natas level 1

### Step 2

We see the following message after we login in:

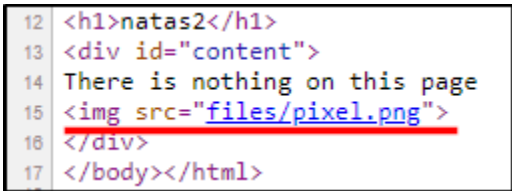


There is nothing on this page

Although the message isn't entirely accurate, the flag for this level is not on this page. This level requires us to do a bit of digging.

### Step 3

Open the webpage source with either the right-click method or the **Ctrl-U** keyboard shortcut. Along with the rest of the code, we can see this image source tag (underlined in red):



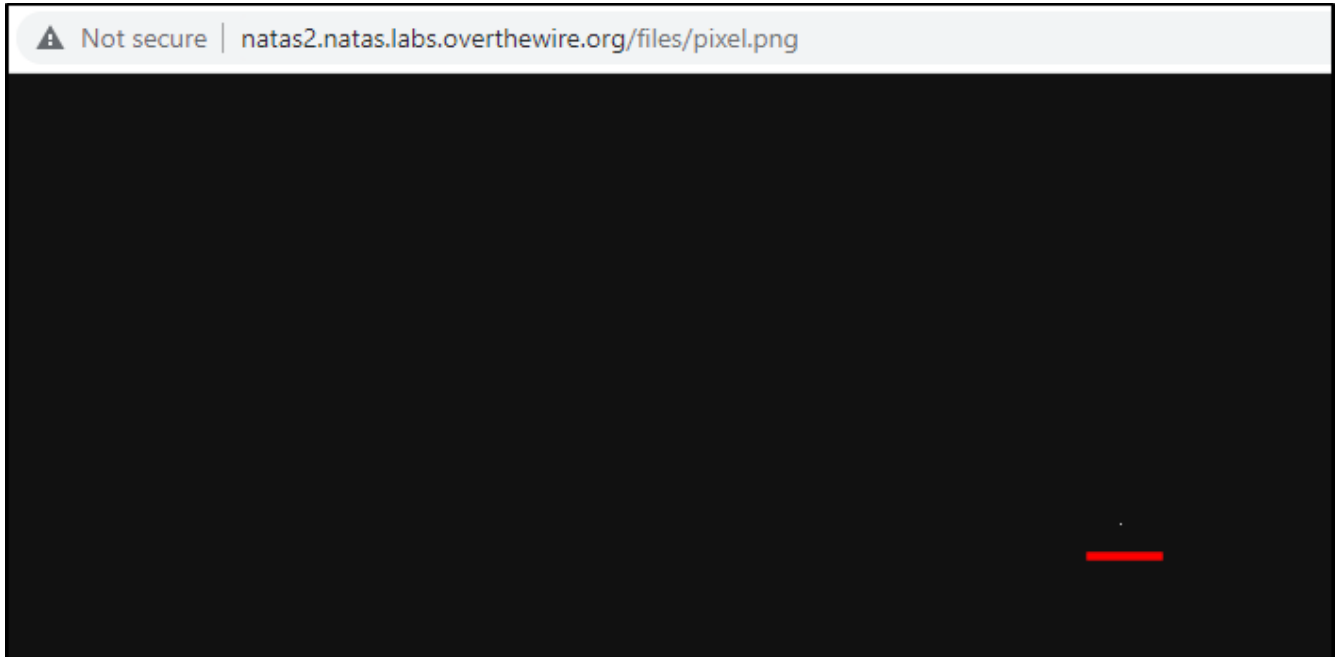
```
12 <h1>natas2</h1>
13 <div id="content">
14   There is nothing on this page
15   
16 </div>
17 </body></html>
```

## CONTEXT

HTML image tags ( `<img>` ) are used to display images on webpages. Considering there is no other significant content in the body of the source code, we should probably investigate this.

## Step 4

Click on the link in the webpage source code. The web browser should display the image, which should look similar to the screenshot below(actual image underlined in red):



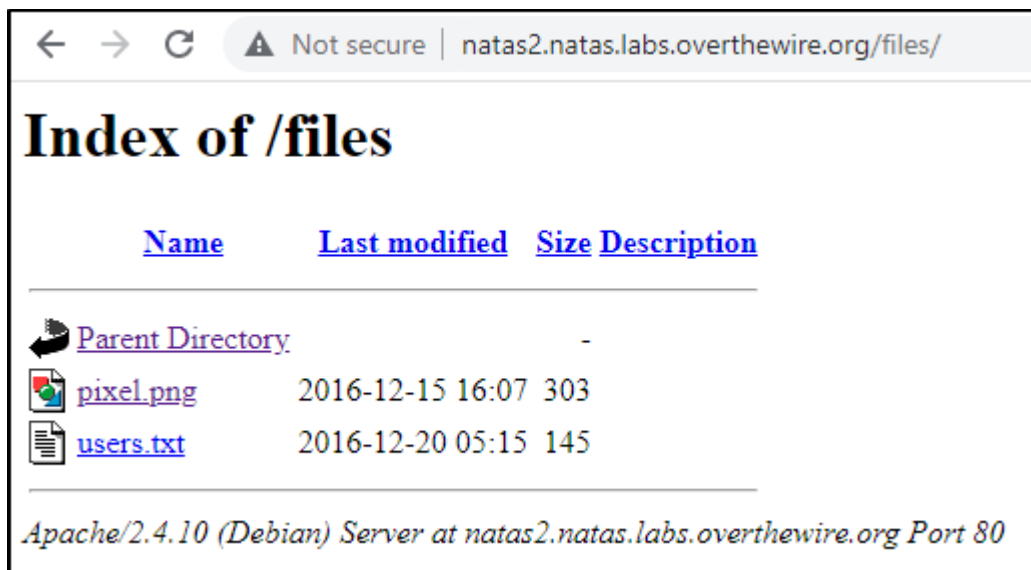
There's not much here either, except for the fact that the image is located in a web directory called **files**. Let's see if there are other files in that directory.

## Step 5

In the web browser address bar, manually change it to the following address:

<https://natas2.natas.labs.overthewire.org/files/>

The resulting page should look similar to the screenshot below:



## CONTEXT

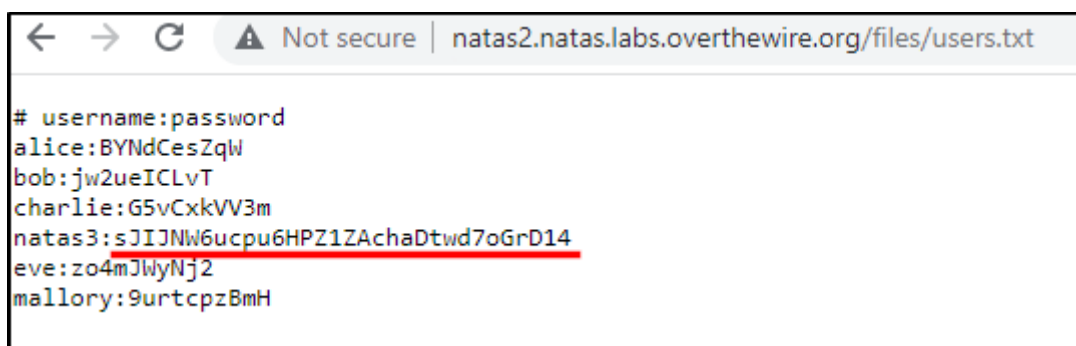
Webpage **directory indexing** (also called **directory listing**), is an insecure setting on websites that allows users to see the file contents of web directories. Although it is almost entirely absent from modern websites, it is still very common to find it on older websites.

### Step 6

Click on the **users.txt** file in the **/files** web directory. Alternatively we can browse to the following URL:

<http://natas2.natas.labs.overthewire.org/files/users.txt>

The resulting page should look similar to the screenshot below (this level's flag underlined in red):



## Part 3 – Robots.txt With Natas 3

### Step 1

In the web browser, navigate to the following webpage:

<http://natas3.natas.labs.overthewire.org>


When prompted, enter the following username and password:

Username: **natas3**

Password: obtained from Natas level 2

### Step 2

On login, we see the following message:

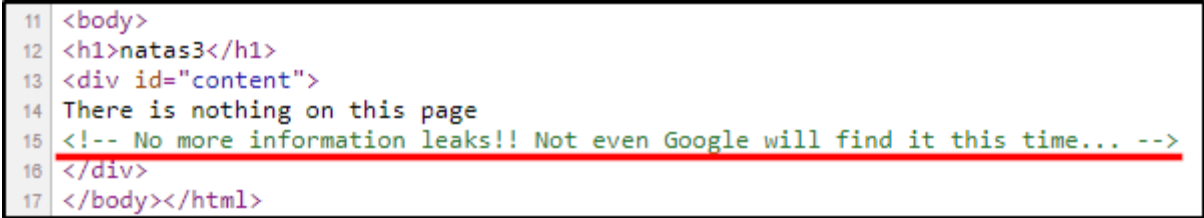


There is nothing on this page

Once again, we should take a look at the webpage source for more clues.

### Step 3

Use the right-click method or the **Ctrl-U** shortcut to access the webpage source. There is an interesting comment included in the code (comment underlined in red):



```
11 <body>
12 <h1>natas3</h1>
13 <div id="content">
14 There is nothing on this page
15 <!-- No more information leaks!! Not even Google will find it this time... -->
16 </div>
17 </body></html>
```

## CONTEXT

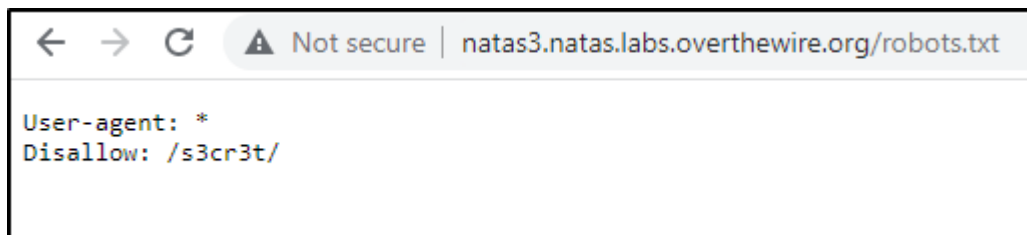
The reference to Google finding information has to do with how internet search engines (like Google) gather information about websites. One tool that they use are web crawling programs (called robots or bots) that visit websites and map out all of the different pages on the website according to the links provided on each page.

### Step 4

We will check if the website implements any blocked files or directories to web crawling robots by navigating to the following URL in our browser:

<http://natas3.natas.labs.overthewire.org/robots.txt>

The resulting page should look like the screenshot below:



## CONTEXT

If website administrators don't want search engine robots to map certain parts of their websites, they can specify which files or directories are allowed or not allowed to be mapped by including them in a file called **robots.txt**.

In this instance, the rules for web crawling robots on this site is that all robots (**User-agent: \***) are not allowed to map the directory **/s3cr3t/ (Disallow: /s3cr3t/)**.

### Step 5

If there's a **s3cr3t** (secret) directory on this website, we definitely want to check that out. Navigate to the following URL in our web browser:

<http://natas3.natas.labs.overthewire.org/s3cr3t/>

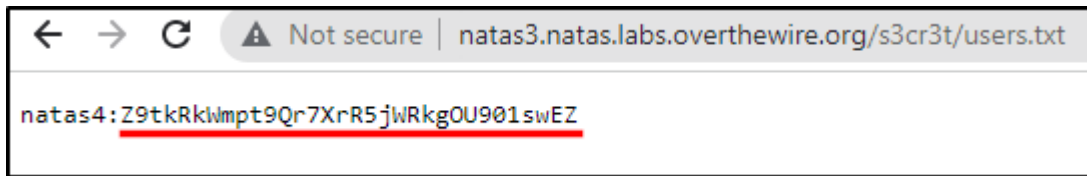
We should see that there is a **users.txt** file in the directory.

### Step 6

Access the **users.txt** file

<http://natas3.natas.labs.overthewire.org/s3cr3t/users.txt>

The contents of the file should look like the screenshot below (level flag underlined in red)



## Part 4 – HTTP Referer Headers With Natas 4

### Step 1

In the web browser, navigate to the following webpage:

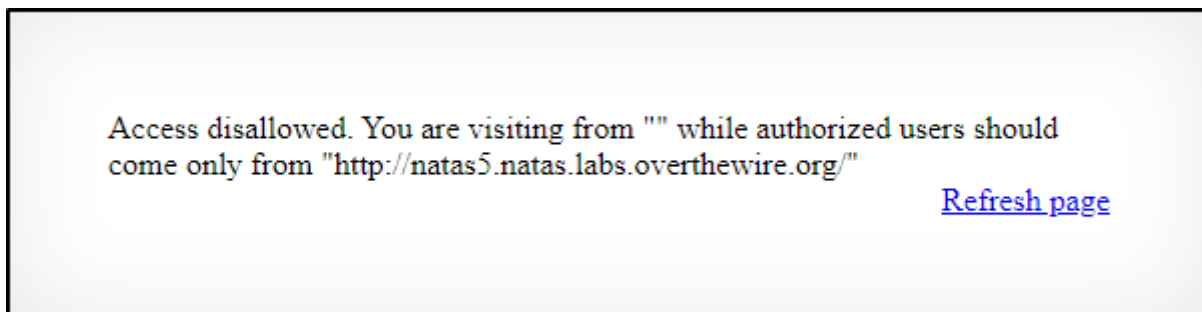
<http://natas4.natas.labs.overthewire.org>

When prompted, enter the following username and password:

Username: **natas4**

Password: obtained from Natas level 3

After login, we see the following on the landing page:



The message here is alluding to the HTTP Referer header, which is an additional parameter that can be sent with HTTP requests. The Referer header keeps track of which webpage or domain a user is coming from when they visit a particular webpage. We'll learn more about the Referer header as we work through this challenge.

### Step 2

Open a command-line interface on your computer. Instructions on how to do so with common operating systems, can be found at the links below:

macOS:

<https://www.idownloadblog.com/2019/04/19/ways-open-terminal-mac/>

Linux:

<https://itsfoss.com/open-terminal-ubuntu/>

Windows:

<https://www.wikihow.com/Open-Terminal-in-Windows>

### Step 3

In the command line terminal, use the following **cURL** command to send a request to Natas level 4, like we did in the web browser:

```
curl -v -u natas4:Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ  
http://natas4.natas.labs.overthewire.org
```

NOTE: The above command is meant to fit on a single line, but is presented here on two lines due to space constraints. Also, the string that is paired with the natas4 username is meant to be the password for natas4. Please replace it with our current password that we've been obtaining by working through the Natas CTF levels.

The first several lines of output should look similar to the screenshot below (HTTP headers outlined in red):

```
C:\Users\shyhat>curl -v -u natas4:Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ http://natas4.natas.labs.overthewire.org/
* Rebuilt URL to: http://natas4.natas.labs.overthewire.org/
* Trying 176.9.9.172...
* TCP_NODELAY set
* Connected to natas4.natas.labs.overthewire.org (176.9.9.172) port 80 (#0)
* Server auth using Basic with user 'natas4'
> GET / HTTP/1.1
> Host: natas4.natas.labs.overthewire.org
> Authorization: Basic bmF0YXM00lo5dGtSa1dtcHQ5UXI3WHJSNwpXUmtnt1U5MDFzd0Va
> User-Agent: curl/7.55.1
> Accept: */*
>
```

### CONTEXT

All virtually all HTTP requests are sent with HTTP headers. A brief summary of the headers present in this request are as follows:



<b>Host</b>	Specifies the host domain to which the request is sent. In this case it's the same address as the URL address.
<b>Authorization</b>	Provides credentials for access to a protected webpage. We see the username and password we use on the web browser to login.
<b>User-Agent</b>	Identifies which program is being used to send the request. The cURL program identifies itself here.
<b>Accept</b>	Specifies which types of data the requesting program is able to accept in the web server's response. The value here indicates that we will accept any type of data in the response.

## Step 4

We will now re-try the cURL command, but this time we will add the HTTP Referer header to the request, specifying the URL that was mentioned on the Natas level 4 landing page.

```
curl -v -H Referer:http://natas5.natas.labs.overthewire.org/ -u
natas4:Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ
http://natas4.natas.labs.overthewire.org
```

Please replace the **Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ** portion of the command with the current password for Natas level 4.

NOTE: The above command is meant to fit on a single line, but is presented here on three lines due to space constraints. If the command does not fit on one line while you are typing it, it won't be an issue.

Let's break down the different components of the cURL command:

`curl`

The name of the command itself

`-v`

The verbose switch, which outputs more output than normal

`-H Referrer:http://natas5.natas.labs.overthewire.org/`

The HTTP header argument, which allows us to send different header values

`-u natas4:Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ`

The user argument allows us to send a username and password along with our request

`http://natas4.natas.labs.overthewire.org`

The last argument to the command is the URL where the request will be sent

The first few lines of the output should look similar to the screenshot below (Referer header outlined in red):

```

C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/
tas.labs.overthewire.org
* Rebuilt URL to: http://natas4.natas.labs.overthewire.org/
* Trying 176.9.9.172...
* TCP_NODELAY set
* Connected to natas4.natas.labs.overthewire.org (176.9.9.172) port 80 (#0)
* Server auth using Basic with user 'natas4'
> GET / HTTP/1.1
> Host: natas4.natas.labs.overthewire.org
> Authorization: Basic bmF0YXM00lo5dGtSa1dtcHQ5UXI3WHJSNWpXUmtnT1U5MDFzd0Va
> User-Agent: curl/7.55.1
> Accept: */*
> Referer:http://natas5.natas.labs.overthewire.org/
>

```

So we see that cURL added the Referer header we specified to the request. In the rest of the output, we see that the request including the Referer header and the proper value specified, and so access was granted and we receive the flag for the level. See the screenshot below (flag string outlined in red):

```

Access granted. The password for natas5 is iX6IOfmpN7AYOQGpwn3fXpbaJVJcHfq
<br/>
<div id="viewsource"><a href="index.php">Refresh page</a></div>
</div>
</body>
</html>
* Connection #0 to host natas4.natas.labs.overthewire.org left intact

```

## Part 5 – Setting Cookies With Natas 5

### Step 1

In the web browser, navigate to the following webpage:

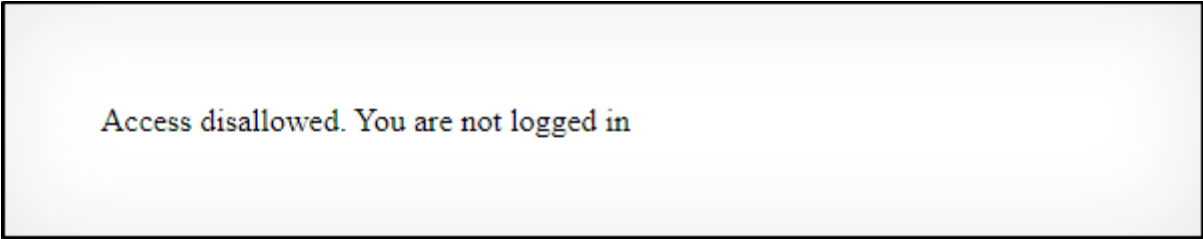
<http://natas5.natas.labs.overthewire.org>

When prompted, enter the following username and password:

Username: **natas5**

Password: obtained from Natas level 4

Upon successful login to the page, we should see this message:



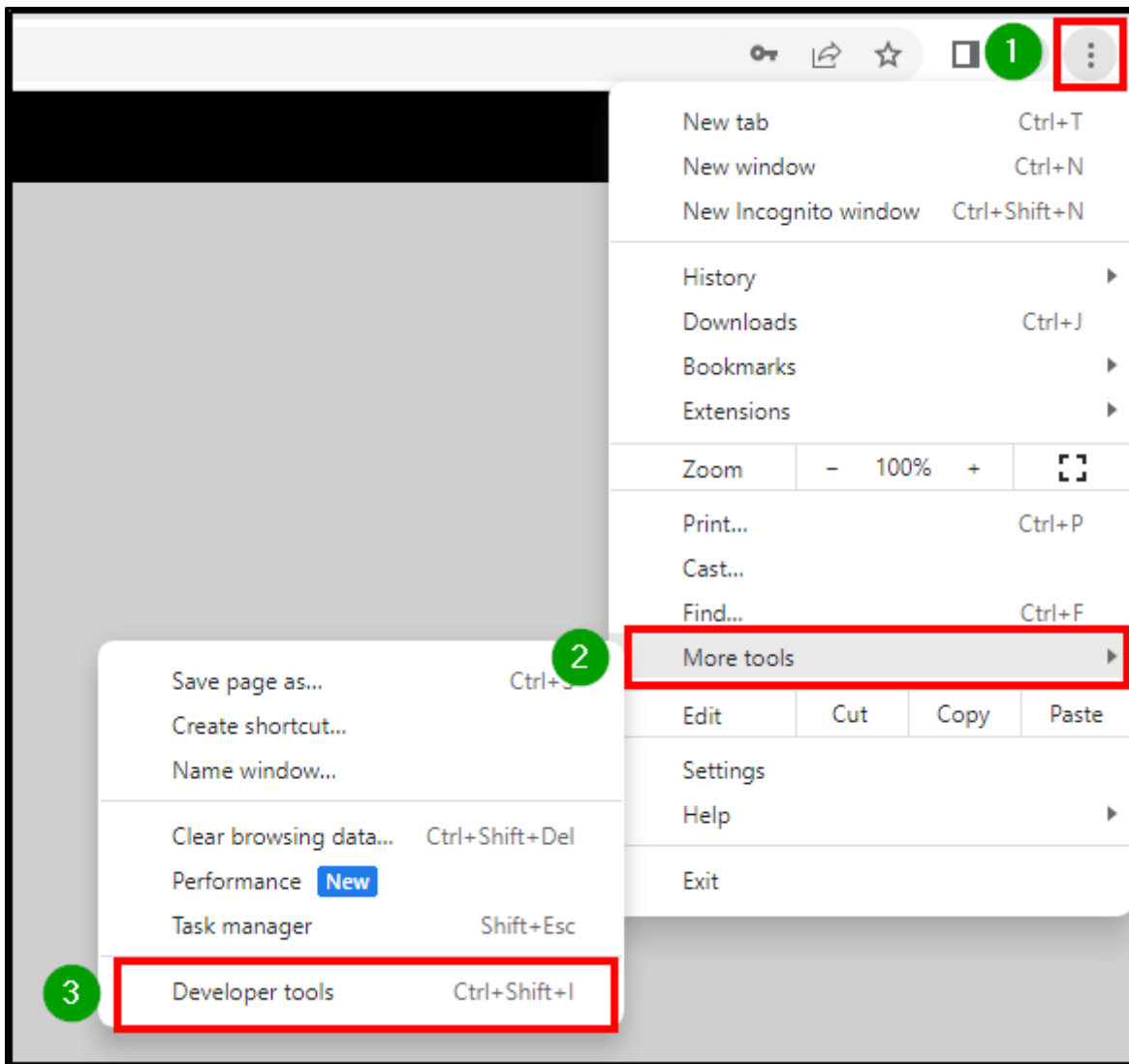
Access disallowed. You are not logged in

We had to supply a valid username and password to see this page, so why does it say we're not logged in?

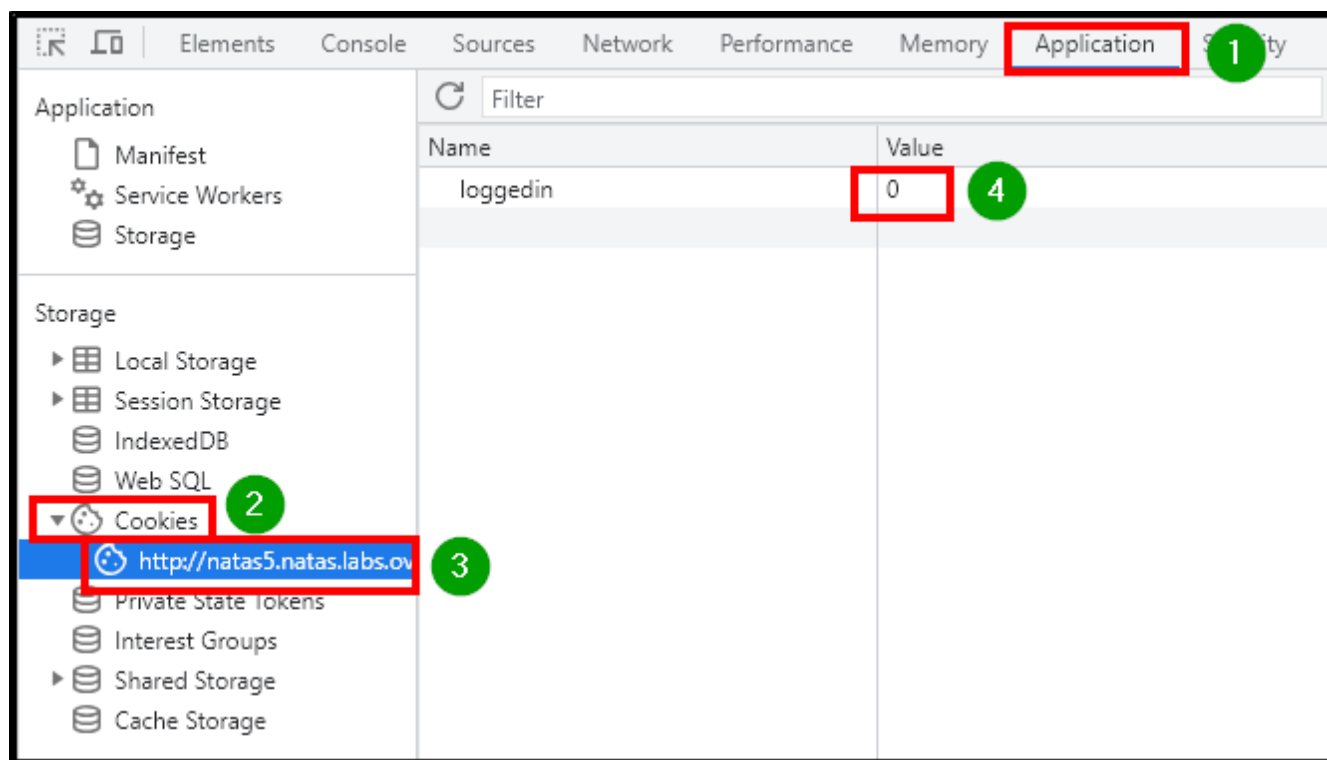
Typically, a web application starts a user session by sending an HTTP cookie from the webserver to the web browser. The values of the HTTP cookies can be found and modified in the web browser.

## Step 2

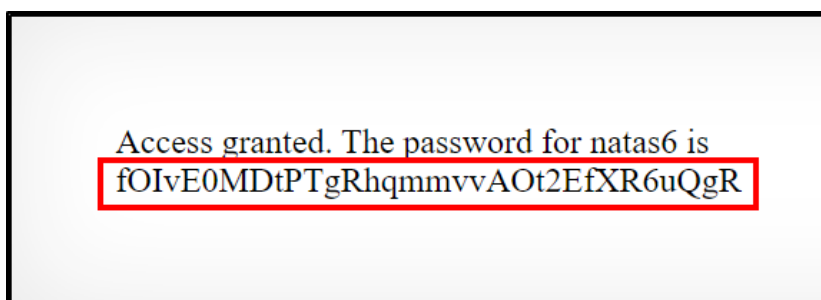
For this example, we are using the Google Chrome web browser. In the web browser, click on the three-dot icon located at the top-right portion of the browser window, then click on the Moore tools button, then the Developer tools button. See the screenshot below (relevant buttons outlined in red):



Next, click on the **Application** tab on the top row of the window that opens (if you don't see the Application tab, click on the left or right arrows on the top row until it becomes visible), then click on the **Cookies** button in the **Storage** section, then click on the natas5 URL, then double-click the **0** under the value column, changing the value to **1**, then pressing the enter key. See the screenshot below (relevant buttons outlined in red):



Refresh the web browser (keyboard F5 key) and we will see the following output (Natas 6 password outlined in red):



## CONTEXT

By going into the web browser cookie settings, we were able to locate the cookie associated with the Natas 5 webpage and modify its value, changing it from 0 (false) to 1 (true). The values of valid authentication cookies should not be any value that can be easily guessed, and typically cookie values are long alphanumeric strings that appear random.

We'll go over one more method of solving this level before we finish the up.

### Step 3

In our command line terminal, use cURL to see what is being sent in the request and being sent back in the response from the web server:

```
curl -v -u natas5:iX6IOfmpN7AYOQGPwtn3fXpbaJVJcHfq  
http://natas5.natas.labs.overthewire.org
```

NOTE: The above command is meant to be single-line command

Please replace the `iX6IOfmpN7AYOQGPwtn3fXpbaJVJcHfq` portion of the above command with the current Natas 5 password.

In the output of the command, we see the following in the webserver response (relevant portion outlined in red):

```
< HTTP/1.1 200 OK  
< Date: Mon, 25 Apr 2022 21:56:31 GMT  
< Server: Apache/2.4.10 (Debian)  
< Set-Cookie: loggedin=0  
< Vary: Accept-Encoding  
< Content-Length: 855  
< Content-Type: text/html; charset=UTF-8
```

### CONTEXT

Among the headers in the response, we see that the server has set a cookie for our session named **loggedin**, and its value is set to 0 (false). That would lead us to speculate that if the loggedin cookie were set to a value of 1 (true), then we would be considered “logged in” and given access to more content on the webpage.

Cookies are commonly used for a variety of purposes, including security, and ensuring that only users with a certain level of authorization can access sensitive webpages.

### Step 3

Use another cURL command to access the Natas 5 page, but this time, include a loggedin cookie with a value set to 1.

```
curl -v --cookie loggedin=1 -u
natas5:iX6IOfmpN7AYOQGPwtn3fXpbaJVJcHfq
http://natas5.natas.labs.overthewire.org
```

Again, the above command is all mean to be on a single line. Again, replace the iX6IOfmpN7AYOQGPwtn3fXpbaJVJcHfq portion of the above command with the current Natas 5 password.

In the first several lines of the output, we see that our request was sent to the server with a cookie, and in the response, the same cookie is set with the value that we sent. See screenshot below (relevant cookies outlined in red):

```
> GET / HTTP/1.1
> Host: natas5.natas.labs.overthewire.org
> Authorization: Basic bmF0YXM1Om1YNk1PZm1wTjdBWU9RR1B3dG4zZlhwYmFKVkpjSGZx
> User-Agent: curl/7.55.1
> Accept: */*
> Cookie: loggedin=1
>
< HTTP/1.1 200 OK
< Date: Mon, 25 Apr 2022 22:05:37 GMT
< Server: Apache/2.4.10 (Debian)
< Set-Cookie: loggedin=1
< Vary: Accept-Encoding
< Content-Length: 890
< Content-Type: text/html; charset=UTF-8
```

And since loggedin is set to 1, we gain access to all of the content on the page, as indicated by the final few lines of the output (level flag outlined in red):

```
<body>
<h1>natas5</h1>
<div id="content">
Access granted. The password for natas6 is aGoY4q2Dc6MgDq4oL4YtoKtyAg9PeHa1</div>
</body>
</html>
* Connection #0 to host natas5.natas.labs.overthewire.org left intact
```

## Summary

While working through the Natas CTF levels 0 to 5, we learned and used our understanding of the following topics to solve various challenges.

**In Natas level 0**, we learned about accessing webpage source from our web browser, and that developers can leave comments on webpages that can be seen in the page source, but not in the standard web browser view.

**In Natas level 1**, we learned that there are multiple ways to access a webpage's HTML source, and the benefit of web searches in problem solving.

**In Natas level 2**, we learned about the concept of directory indexing, an insecure configuration on websites that allows users to see the full contents of directories on a website.

**In Natas level 3**, we learned about robots.txt, a file that contains directories and / or files that the website developers don't want to be indexed by search engines. Sometimes these directories and files are of a sensitive nature, and if malicious users were to access the robots.txt file, it may lead them directly to potentially sensitive data on a website.

**In Natas level 4**, we learned about the HTTP Referer header, which is sometimes used by websites as a means of ensuring that users can only access a webpage if they came directly from another webpage or domain. However, it isn't a reliable method of security, since values in the HTTP Referer header can easily be spoofed.

**In Natas level 5**, we learned about HTTP Cookie headers, which can also be enumerated and exploited if the expected values used by the header are predictable.

In the next workshop, we'll continue to learn about web app exploits with Natas CTF, including LFI (Local File Inclusion) and injection exploits.

## Extra Credit

The following exercises are related to the concepts in this workshop, and can be challenged for extra practice (a free registered account at each of these websites is required to complete these challenges):

### **picoCTF – Scavenger Hunt**

<https://play.picoctf.org/practice/challenge/161>

NOTES: A registered account is required to do this exercise.

### **ctfLearn – My Blog**

<https://ctflearn.com/challenge/979>



## Extra Research

The following articles will help us further understand some of the concepts covered in this workshop:

Article on directory listing (aka directory indexing):

[https://portswigger.net/kb/issues/00600100\\_directory-listing](https://portswigger.net/kb/issues/00600100_directory-listing)

More information about the robots.txt file:

<https://www.cloudflare.com/en-ca/learning/bots/what-is-robots-txt/>

How HTTP cookies work:

<https://www.kaspersky.com/resource-center/definitions/cookies>

Until next time, HackerFrogs!

