

Laboratory Exercise – Natas Level 0-6

1. Overview

In this lab exercise, students will learn new skills by setting up the Sublime Text editor with Python and using it to test web applications. Using the Brigante (2020) environment, students will be presented with progressive challenges from the OverTheWire website (Natas) in which they will use previously learned skills to find the Capture the Flag (CTF) flags.

2. Resources required

This exercise requires the Brigante VM running in the Cyber Range.

3. Initial Setup

For this exercise, you will log in to your Cyber Range account, select the Brigante (2020) environment, click “start” to start your environment, and then “join” to get to your Linux desktop.

4. Tasks

Task 1: Setting up SublimeText

The first thing we need to do is download and install the GPG key (GNU PGP Key). This allows for a secure download from the creators of SublimeText. Then we will configure the repo for the version of SublimeText we are downloading. The final step is to update and install the application. This is a CentOS Linux distribution, so some of the commands will be slightly different.

- In a root terminal, type the following: `yum install python python3-pip` when prompted type `y` and press enter.
- In a root terminal, type the following: `pip3 install requests-html` and press enter.
- In a root terminal, type the following:
`rpm -v --import https://download.sublimetext.com/sublimehq-rpm-pub.gpg`
and press enter. The terminal will return to the prompt if completed correctly.
- Type the following: `yum-config-manager --add-repo https://download.sublimetext.com/rpm/stable/x86\_64/sublime-text.repo`
and press enter.
- Type the following: `yum install sublime-text` and press enter. When prompted type `y` and press enter.

```
[root@ip-10-1-49-229 student]# yum-config-manager --add-repo https://download.sublimetext.com/rpm/stable/x86_64/sublime-text.repo
Loaded plugins: fastestmirror
adding repo from: https://download.sublimetext.com/rpm/stable/x86_64/sublime-text.repo
grabbing file https://download.sublimetext.com/rpm/stable/x86_64/sublime-text.repo to /etc/yum.repos.d/sublime-text.repo
repo saved to /etc/yum.repos.d/sublime-text.repo
[root@ip-10-1-49-229 student]#
```

```

=====
Package           Arch      Version      Repository      Size
=====
Installing:
sublime-text      x86_64     3211-1       sublime-text     13 M
=====
Transaction Summary
=====
Install 1 Package

Total download size: 13 M
Installed size: 33 M
Is this ok [y/d/N]: y
Downloading packages:
sublime-text-3211-1.x86_64.rpm | 13 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : sublime-text-3211-1.x86_64      1/1
  Verifying  : sublime-text-3211-1.x86_64      1/1

Installed:
sublime-text.x86_64 0:3211-1

Complete!
[root@ip-10-1-49-229 student]#

```

```

Terminal - root@ip-10-1-169-192:/home/student
File Edit View Terminal Tabs Help
=====
Installing:
sublime-text      x86_64     4126-1       sublime-text     20 M
=====
Transaction Summary
=====
Install 1 Package

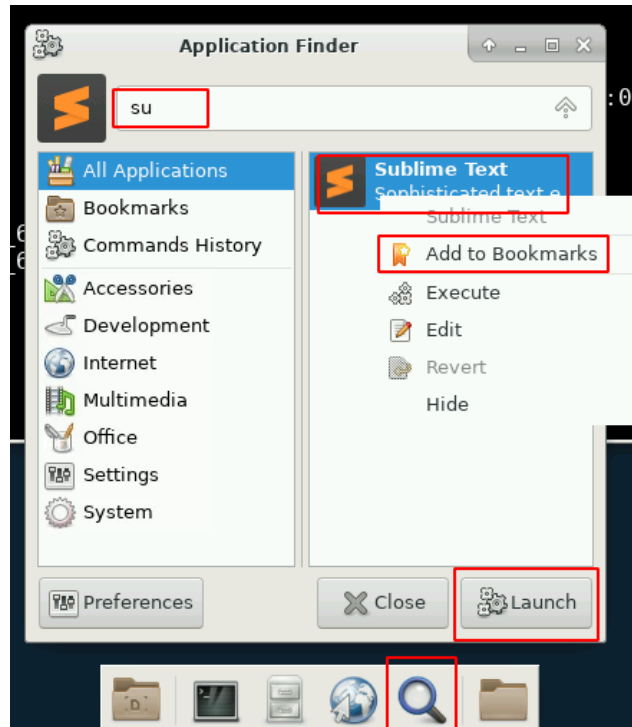
Total download size: 20 M
Installed size: 48 M
Is this ok [y/d/N]: y
Downloading packages:
sublime-text-4126-1.x86_64.rpm | 20 MB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : sublime-text-4126-1.x86_64      1/1
  Verifying  : sublime-text-4126-1.x86_64      1/1

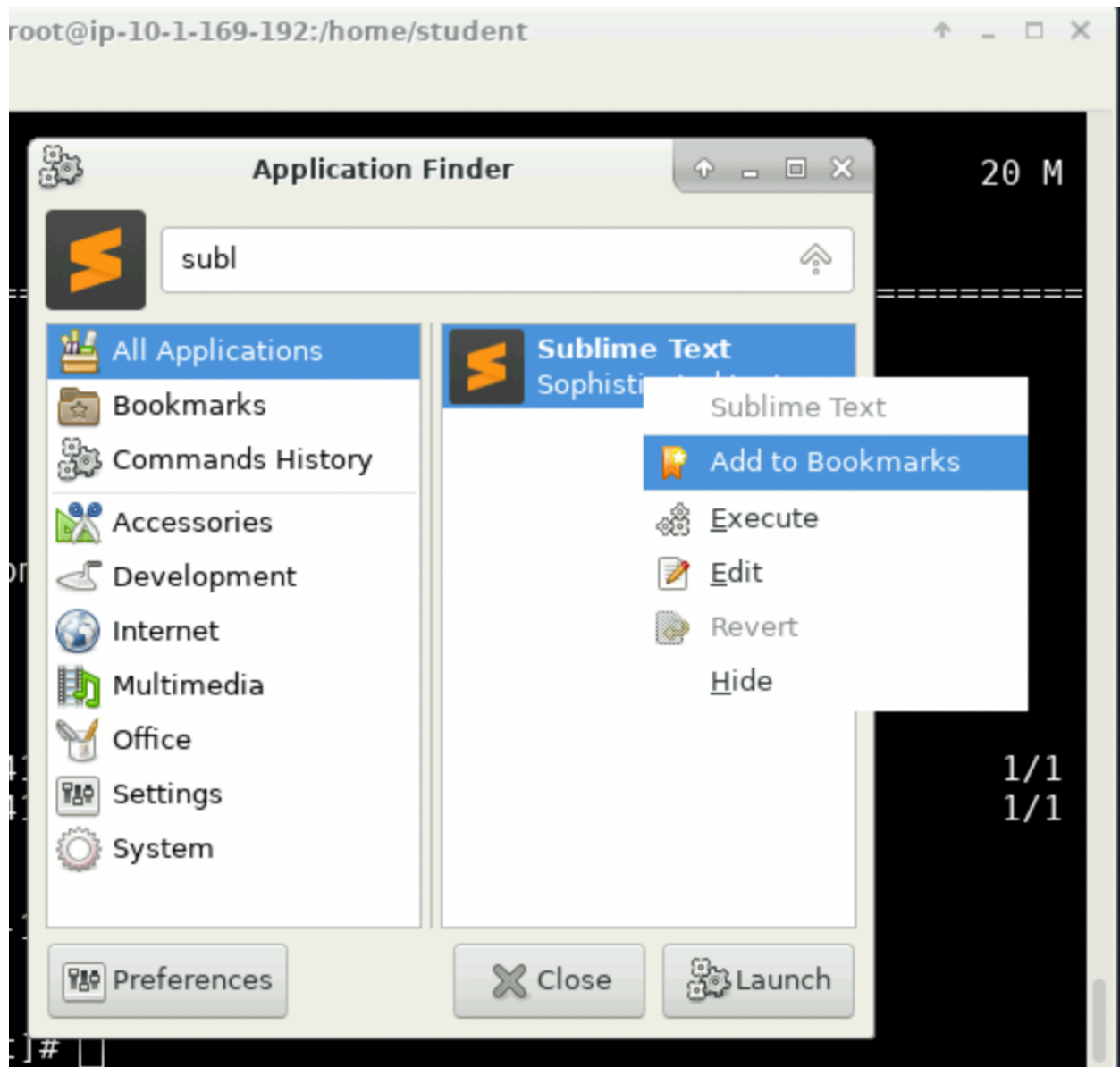
Installed:
sublime-text.x86_64 0:4126-1

Complete!
[root@ip-10-1-169-192 student]#

```

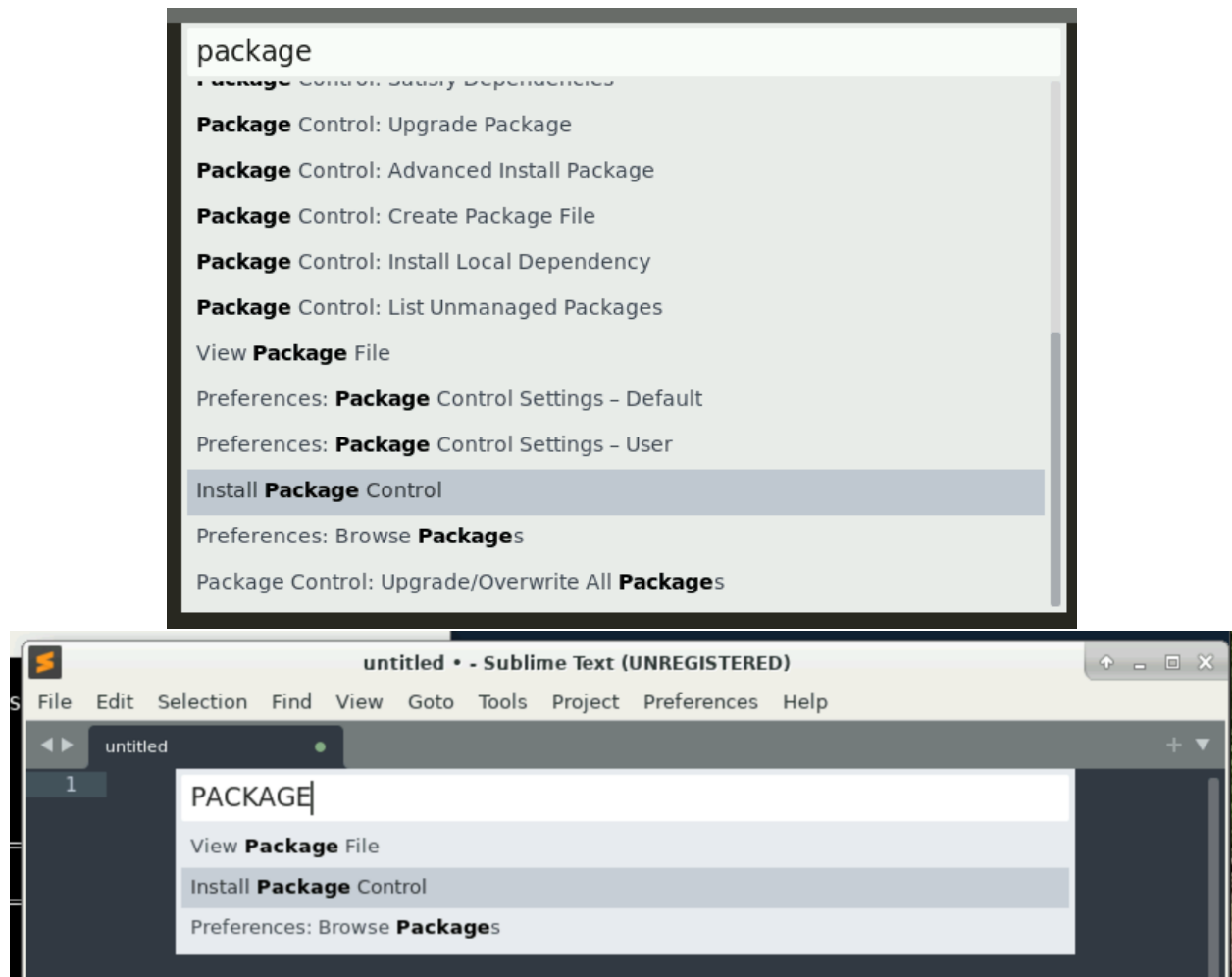
Now we can click on the magnifying glass at the bottom of the screen on the desktop and search for **Sublime Text**. Right click on the application logo and click **Add to Bookmarks**. In addition, you can click and drag the application logo to the desktop to create a shortcut. Once you have your shortcuts setup, click the **Launch** button.



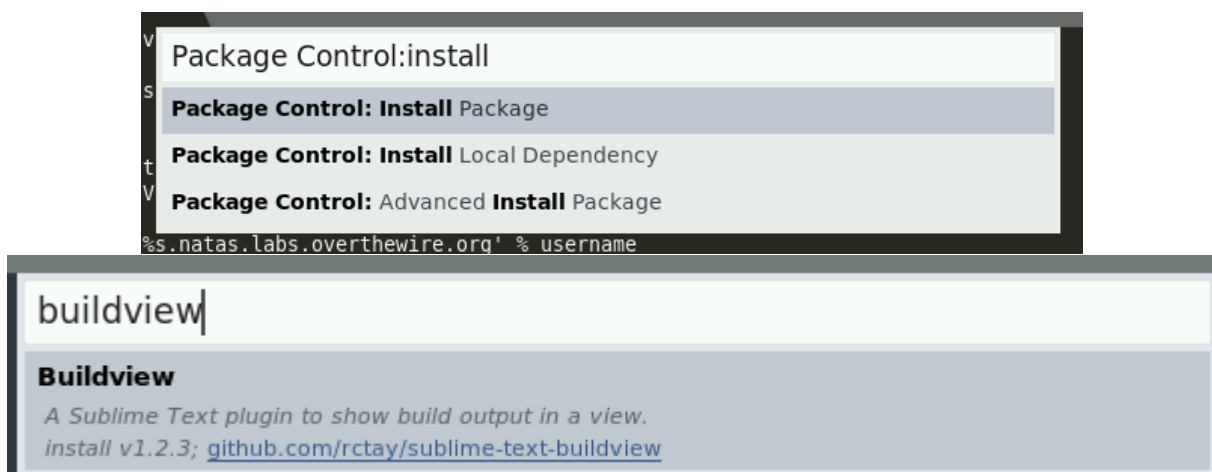


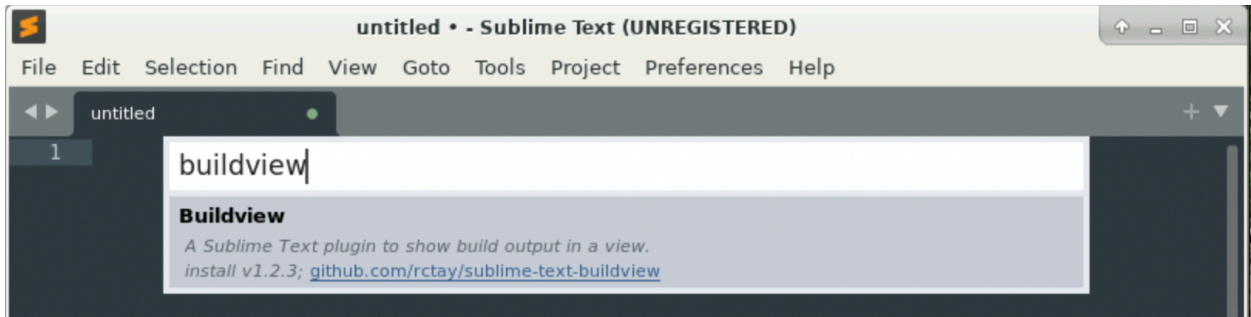
In Sublime Text, we need to set up a few things so that we can build and see our code. To do this, we will need to install the **Package Control** and **Buildview**.

- To install the Package Control, press ctrl+shift+p, and then in the search box, type **Install Package Control**.



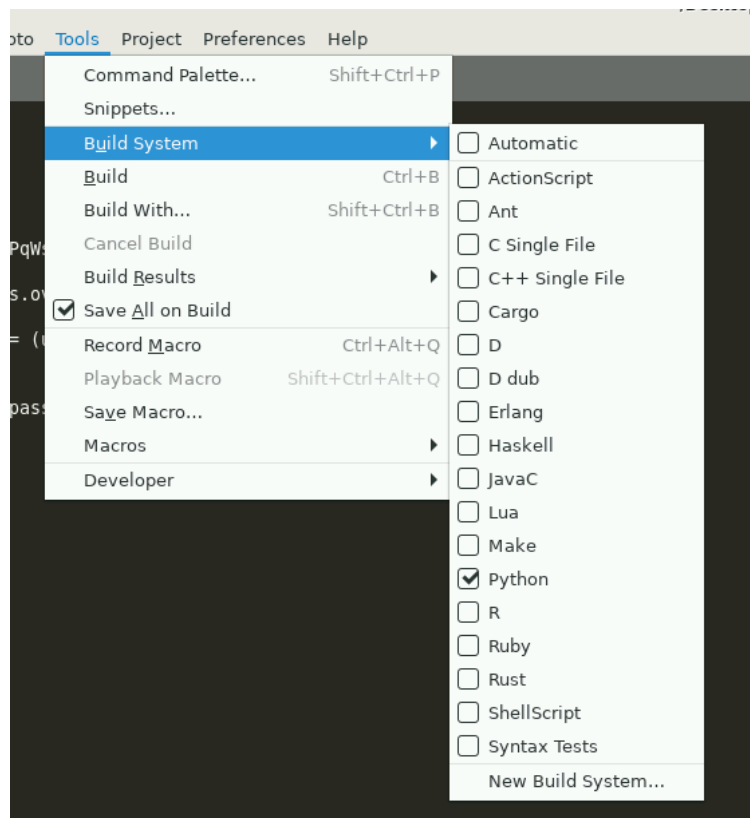
- To install a package press ctrl+shift+p and click on **Package Control: Install Package**.
- In the package control search box, type **buildview**.

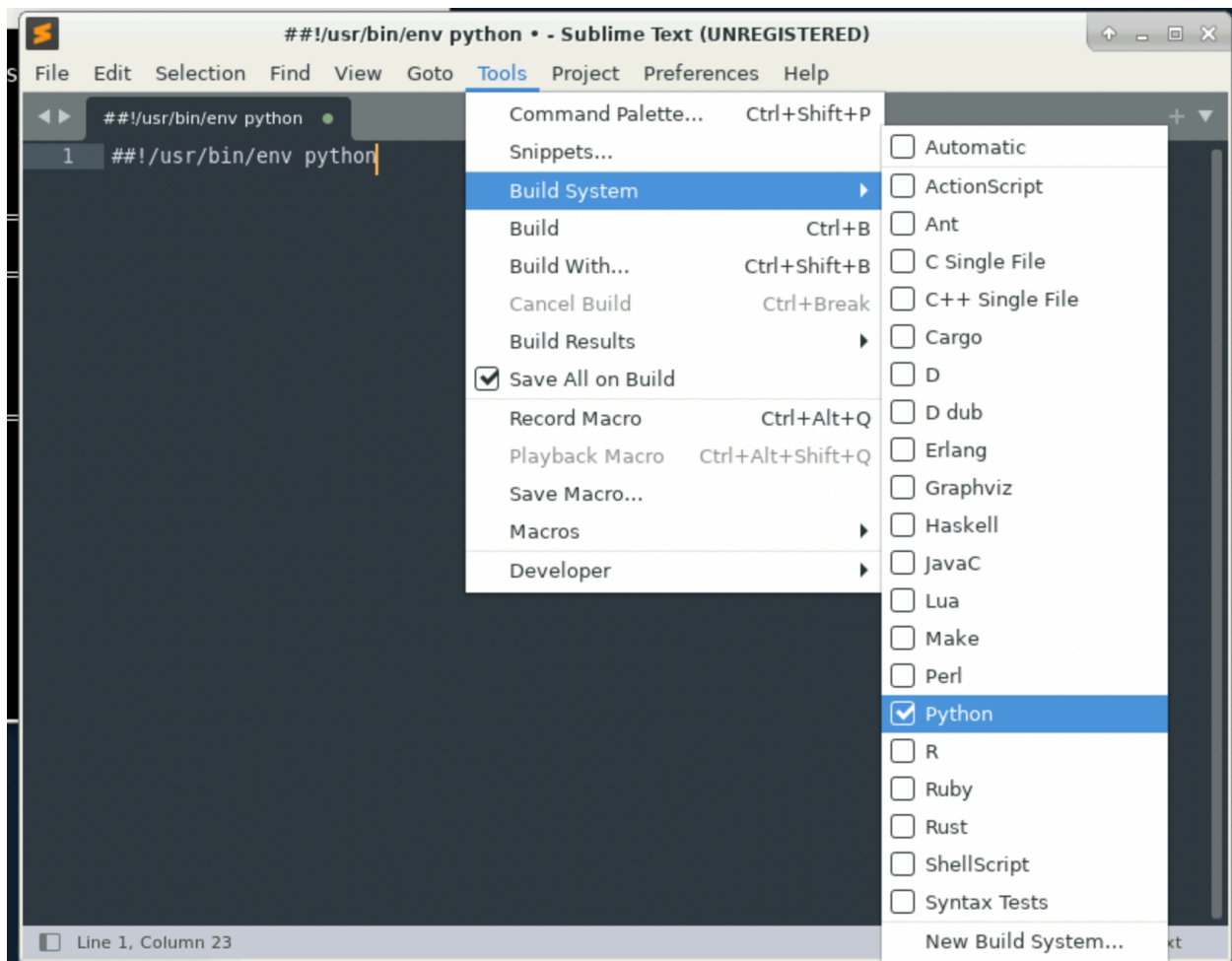




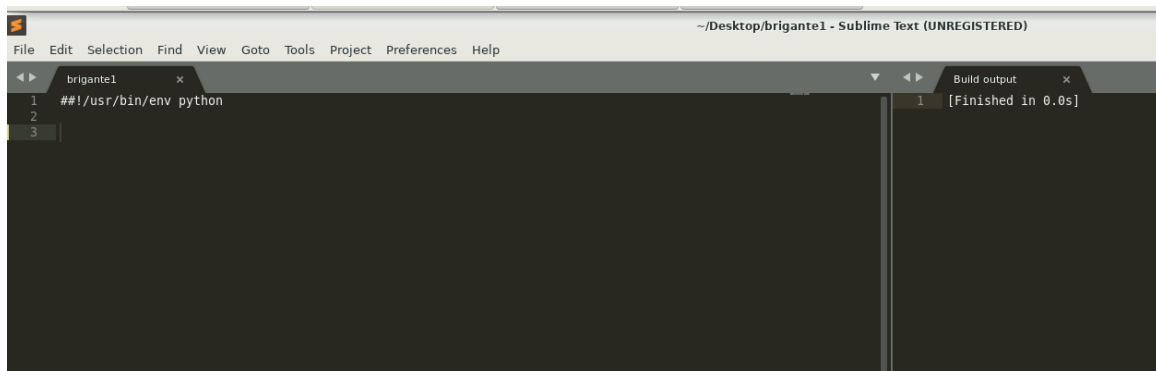
Now to build our code, we will have to set the build language.

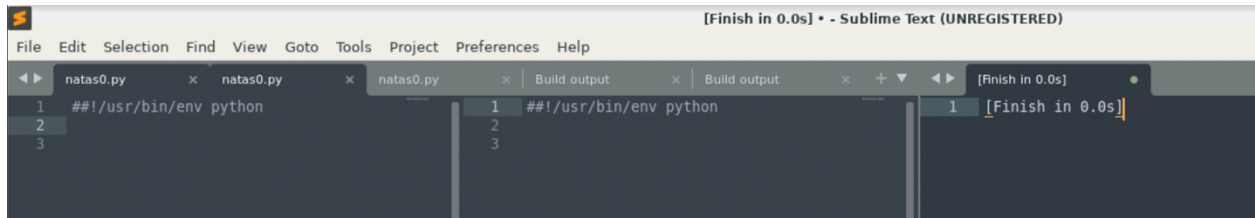
- In the SublimeText editor, type: `#!/usr/bin/env python`
- Click the **Tools** menu, select **Build System**, and then check the **Python** box.





- On the Desktop, create a folder named **natas**.
- In Sublime Text, save your file as **natas0.py** in the natas folder you just created on the Desktop.
- Now we need to exit out of Sublime Text and open it back up.
- To get the split screen, press alt+shift+2.
- Press ctrl+b to build the output.





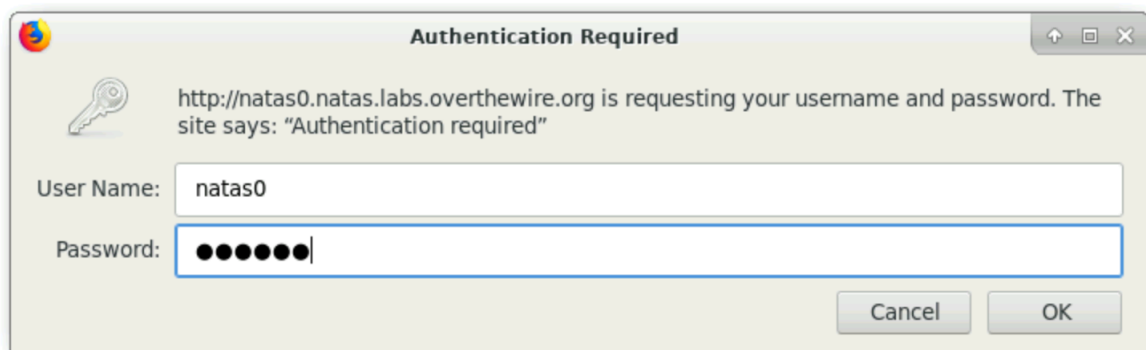
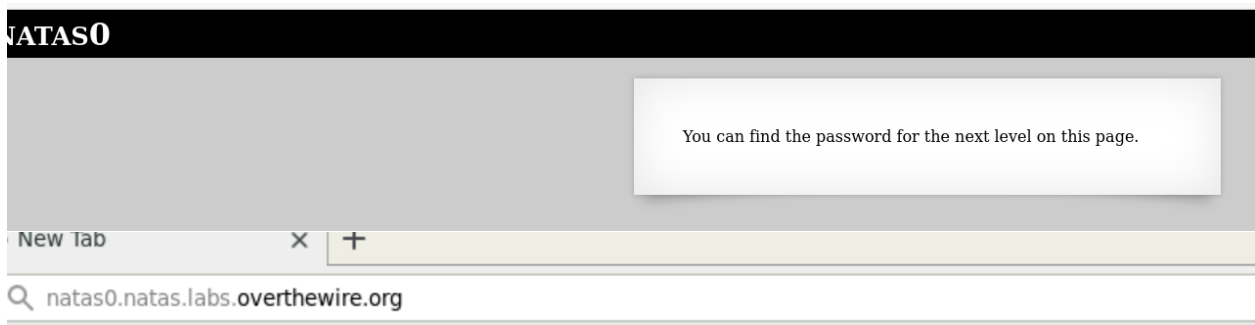
Task 2: Python Programming Natas 0-1

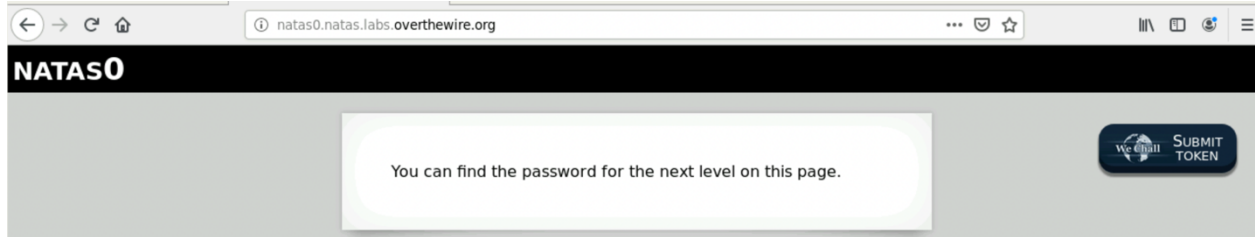
The goal in the lesson is not to be an expert in Python. The idea is to learn how Python can be used to parse information from a website using simple scripts. Don't focus too much on being a perfect Python programmer. Rather focus on the patterns that you see when programming and parsing information.

- In a browser, navigate to <https://overthewire.org/wargames/natas/natas0.html>

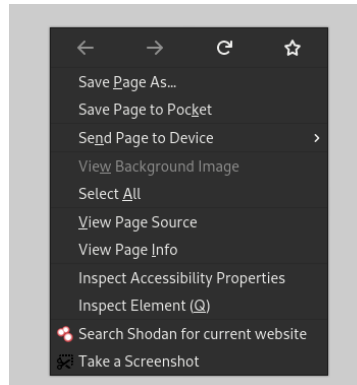
To the left, you can see all the challenges. They are progressive and will get more difficult as we proceed.

- Copy and paste the link from the page into a new tab in the browser.
<http://natas0.natas.labs.overthewire.org>
- Enter the username **natas0** and the **password natas0**.





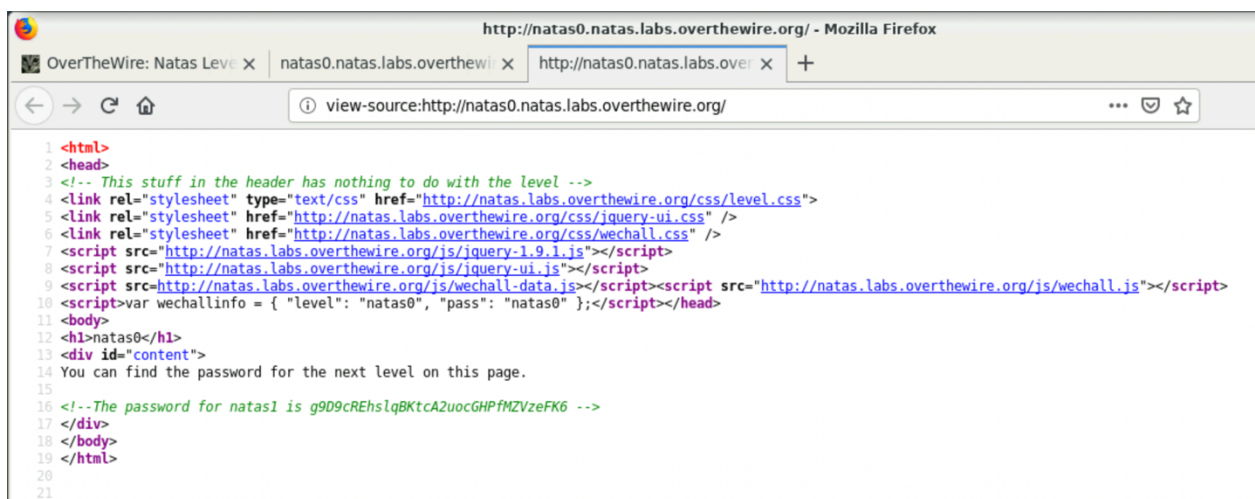
- Right click on the page and click **View Page Source**.



```

1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas0", "pass": "natas0" };</script></head>
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>
20
21

```



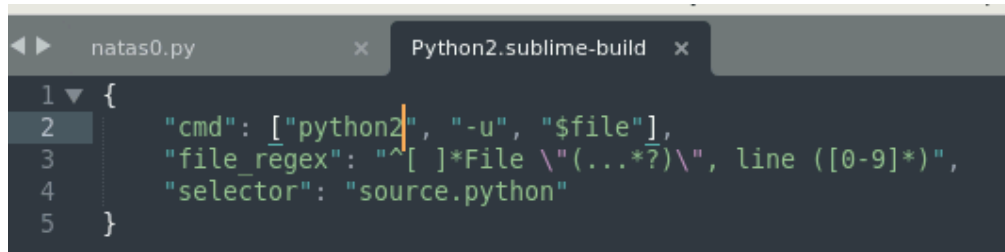
Natas 1 password: g9D9cRehslqBKtcA2uocGHPfMZVzeFK6

The password can be shown in the page source as a comment. When testing a web application, it is always important to view the source. Sometimes you can find sensitive information that leads to a deeper understanding of the application or its owners.

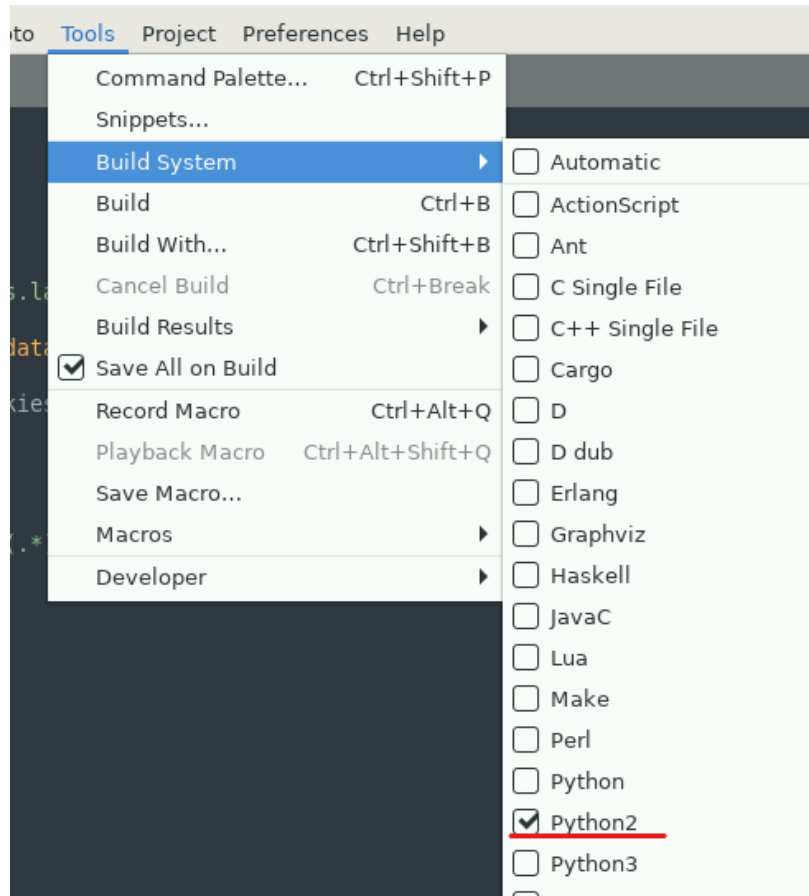
To prevent error messages, we can create a new python2 build system inside of SublimeText.

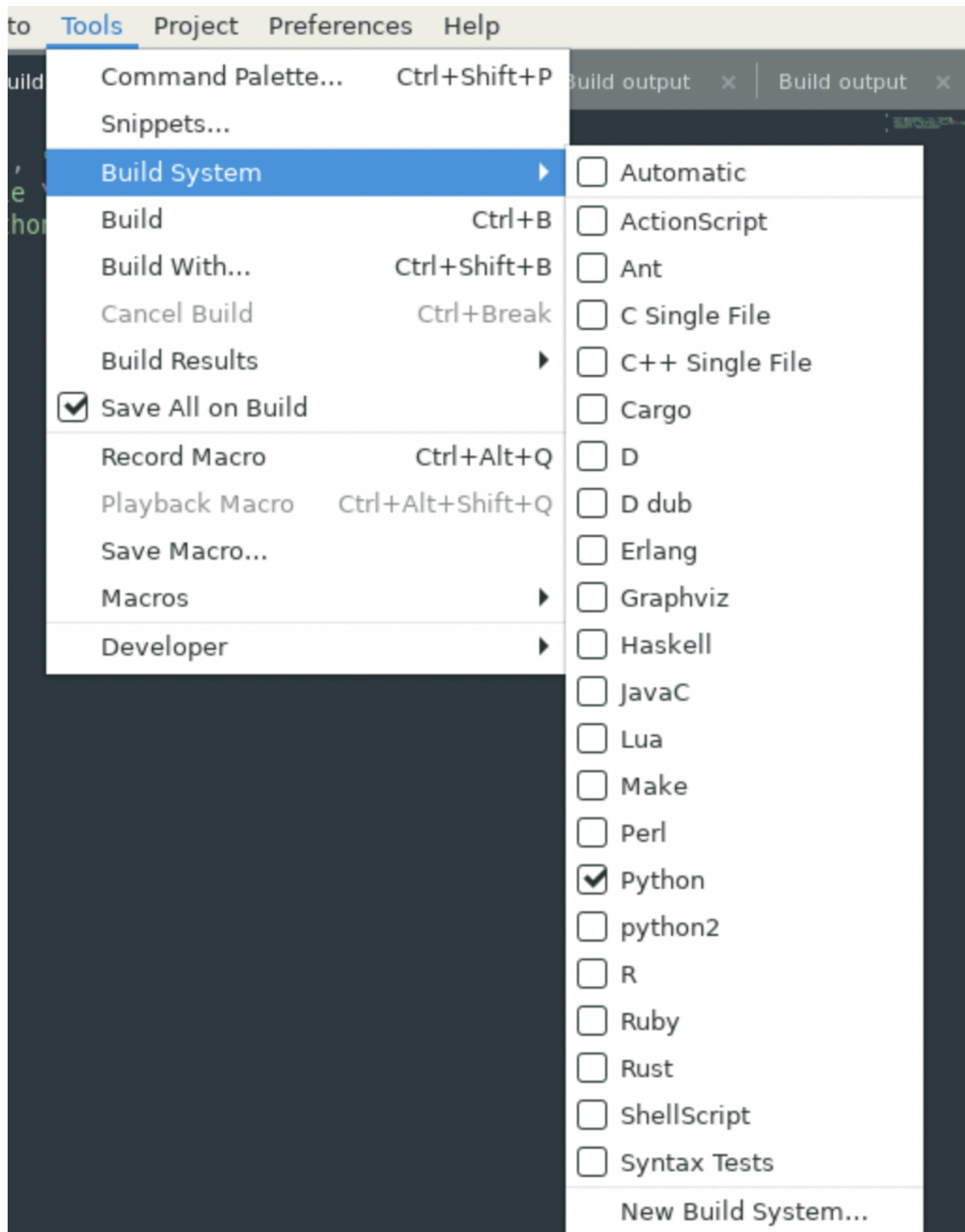
- In SublimeText, click **Tools -> Build System -> New Build System**
- Copy and paste the following code:

```
{  
    "cmd": ["python2", "-u", "$file"],  
    "file_regex": "^[ ]*File \"(...*?)\", line ([0-9]*)",  
    "selector": "source.python"  
}
```



- Click **File > Save As** and name the file **python2.sublime-build**. Then click **Save**.
- Click **Tools -> Build System -> python2**





- Exit out and restart SublimeText.

From this point on, be sure to pay close attention to the colors in the screenshots. A simple typo can lead to an error and the colors help pinpoint where the line error is. In Sublime Text editor, under the shebang (`#!/`), add the following to the `natas0.py` file (see image below):

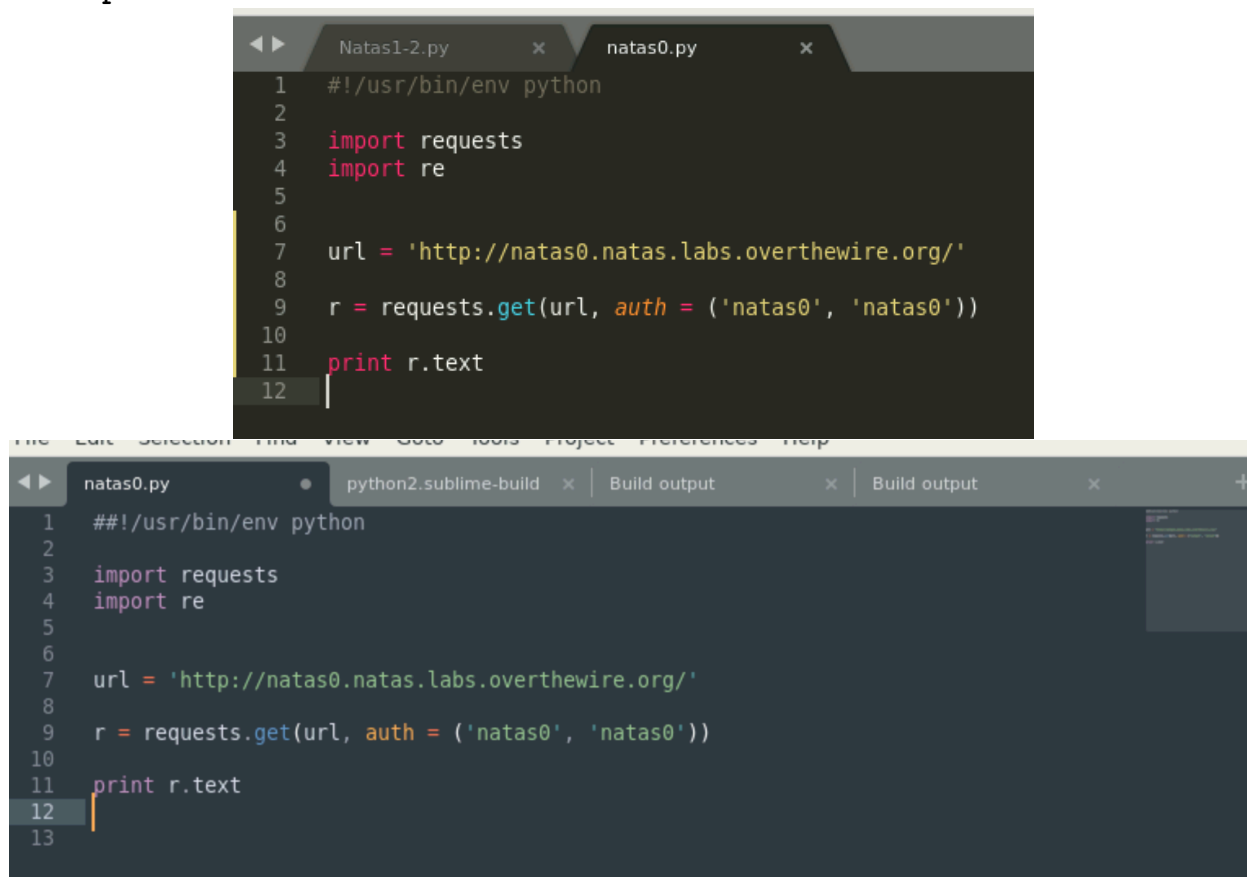
```
#!/usr/bin/env python

import requests
import re

url = 'http://natas0.natas.labs.overthewire.org/'

r = requests.get(url, auth = ('natas0', 'natas0'))

print r.text
```



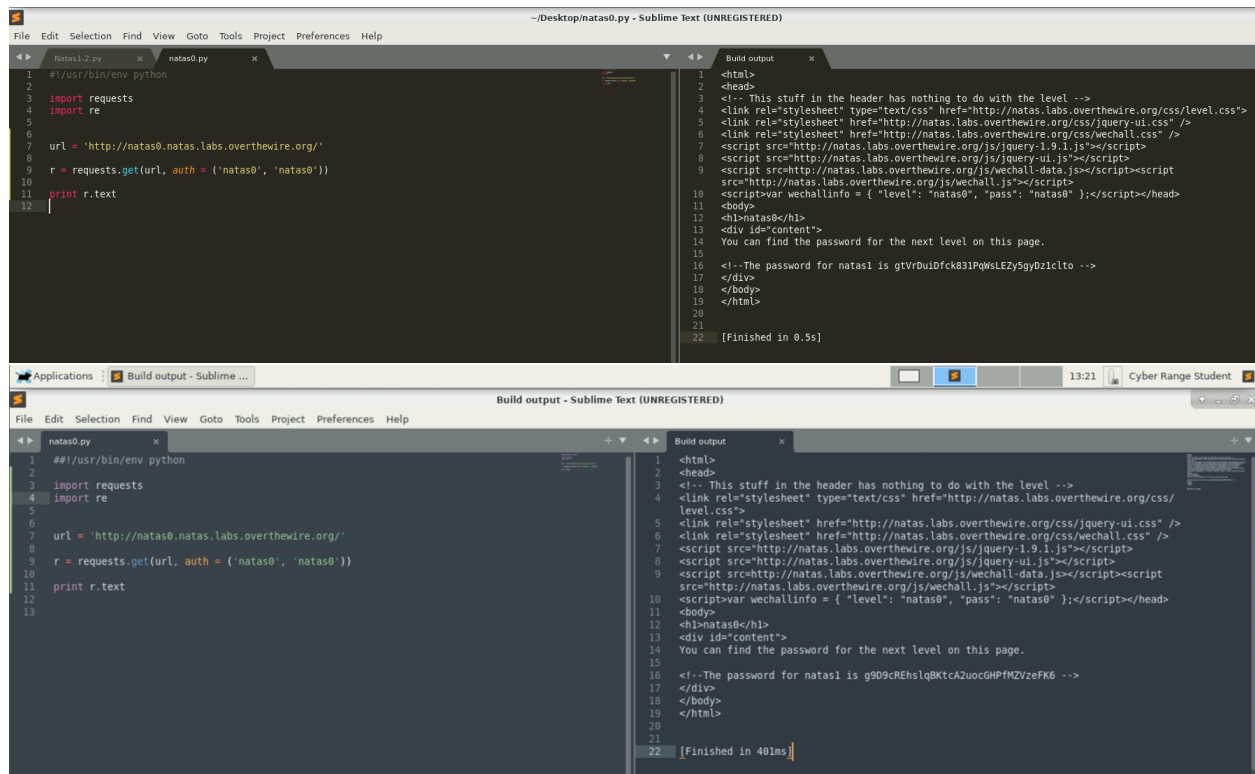
- Save the code with **CTRL+s** (you will want to do this frequently).

Python Code Breakdown:

- **import requests** - Imports the request module that allows http requests to be made.
- **import re** - Imports the module “Regular expressions” allowing characters such as the backslash (`\`) to be used without invoking their special meaning.
- **r = requests.get(url, auth = ('natas0', 'natas0'))** - This sets the variable `r` to access the get request with authorization using the username and password that we set (username first, then password).

- **print r.text** - prints the results of the variable r. In this case, the get request with authorization.
- In the **natas0.py** tab, press **CTRL+b** to build the program and output it in a new tab.

The first time you do this, you may have to click and drag the tab to the other side of the split screen to get the results in the screenshot below.



As you can see, the request is now in the build output tab and the next password for natas1 is there.

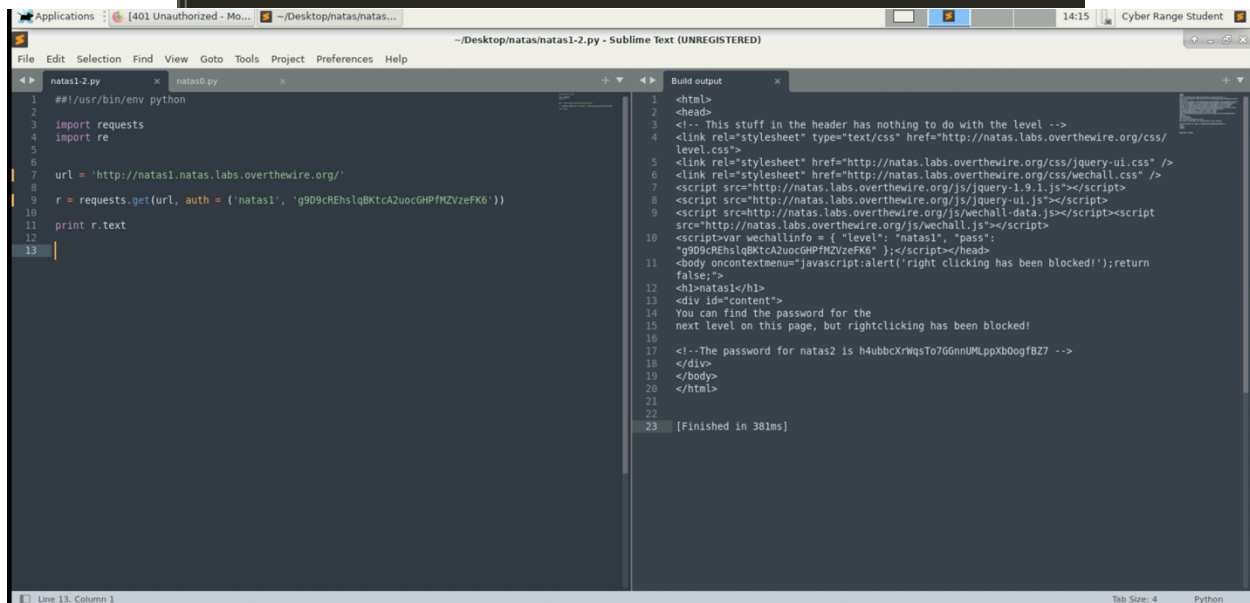
Task 3: Python Programming Natas 1-2

The next challenge is a bit easier since all we really need to do is change a few parameters to our code.

- In the **natas0.py** tab, change the **url** to **http://natas1.natas.labs.overthewire.org/**
- In the **r** variable, change the username 'natas0' to 'natas1'
- In the **r** variable, change the password 'natas0' to the password discovered in the previous task.
- Using the **CTRL+SHIFT+S**, save the file as **natas1-2.py** to the natas folder on your Desktop.
- In the **natas1-2.py** tab, press **CTRL+b** to build the program. See the images below.

```
Natas1-2.py x
1  #!/usr/bin/env python
2
3  import requests
4  import re
5
6
7  url = 'http://natas1.natas.labs.overthewire.org/'
8
9  r = requests.get(url, auth = ('natas1', 'gtVrDuiDfck831PqWsLEZy5gyDz1clto'))
10
11  print r.text
12
```

```
Build output x
1  <html>
2  <head>
3  <!-- This stuff in the header has nothing to do with the level -->
4  <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5  <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6  <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7  <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8  <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9  <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script
10 src="http://natas.labs.overthewire.org/js/wechall.js"></script>
11 <script>var wechallinfo = { "level": "natas1", "pass": "gtVrDuiDfck831PqWsLEZy5gyDz1clto"
12 };</script></head>
13 <body oncontextmenu="javascript:alert('right clicking has been blocked!');return false;">
14 <h1>natas1</h1>
15 <div id="content">
16 You can find the password for the
17 next level on this page, but rightclicking has been blocked!
18
19 <!--The password for natas2 is ZluruAthQk7Q2MqmDeTiUij2ZvWY2mBi -->
20 </div>
21 </body>
22 </html>
23 [Finished in 0.7s]
```



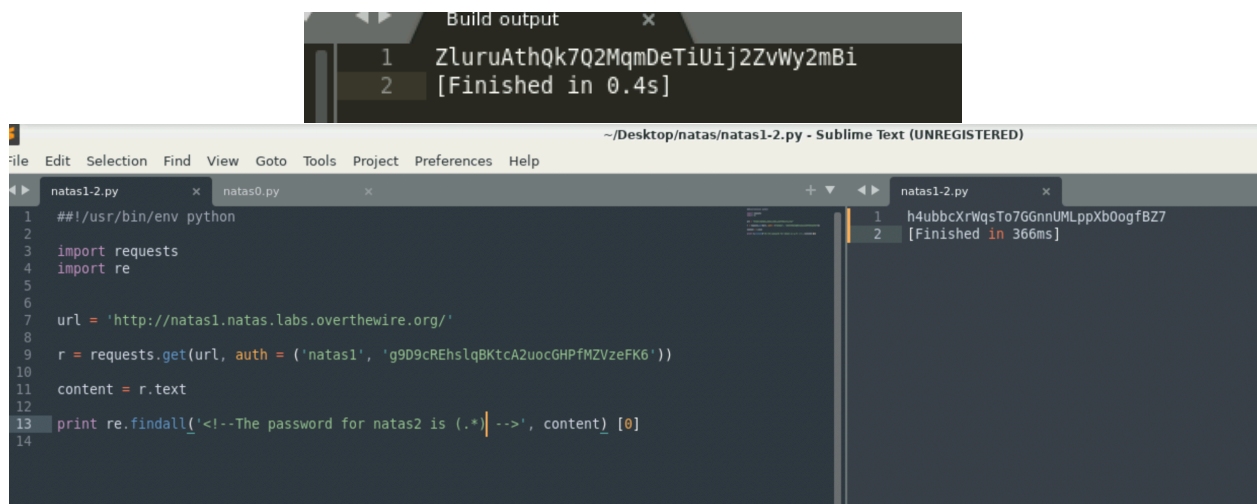
Natas 2 password: h4ubbcXrWqsTo7GGnnUMLppXb0ogfBZ7

It appears there is a pattern of “<!--the password for natas# is passwordhere -->”. Because of this, we can make a few adjustments to the Python code so we don’t have to keep looking

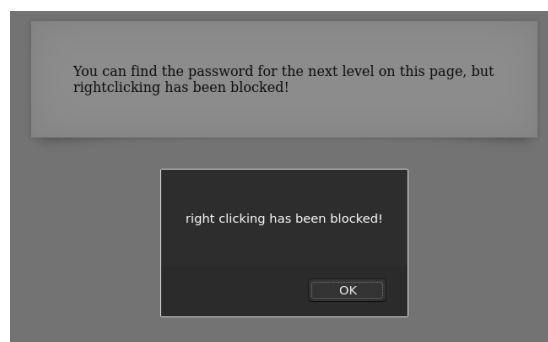
through the entire html output. Instead, we can use `re.findall` to search through the html page for this pattern.

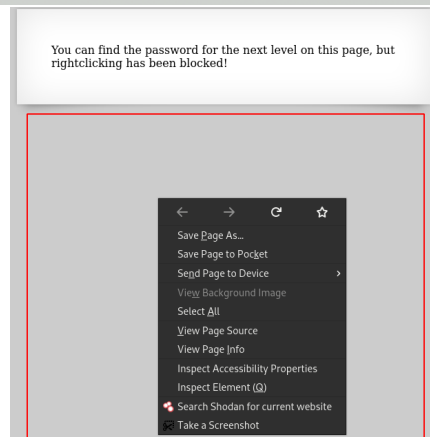
- In the `natas1-2.py` tab, delete `print r.text` and in its place add `print re.findall('<!--The password for natas2 is (.*) -->', content) [0]` (REMINDER: Don't forget to save the file every time you make a change.)
- Add the line `content = r.text` after the `r = requests.get...` as shown in the screenshot below
- In the `natas1-2.py` tab, press CTRL+b to build the program.

```
7 url = 'http://natas1.natas.labs.overthewire.org/'
8
9 r = requests.get(url, auth = ('natas1', 'gtVrDuiDfck831PqWsLEZy5gyDz1clto'))
10
11 content = r.text
12
13 print re.findall('<!--The password for natas2 is (.*) -->', content) [0]
```



- To complete this task on the site, you simply use the username and password from `natas1` at <http://natas1.natas.labs.overthewire.org> then right click outside of the container to view the source.





```

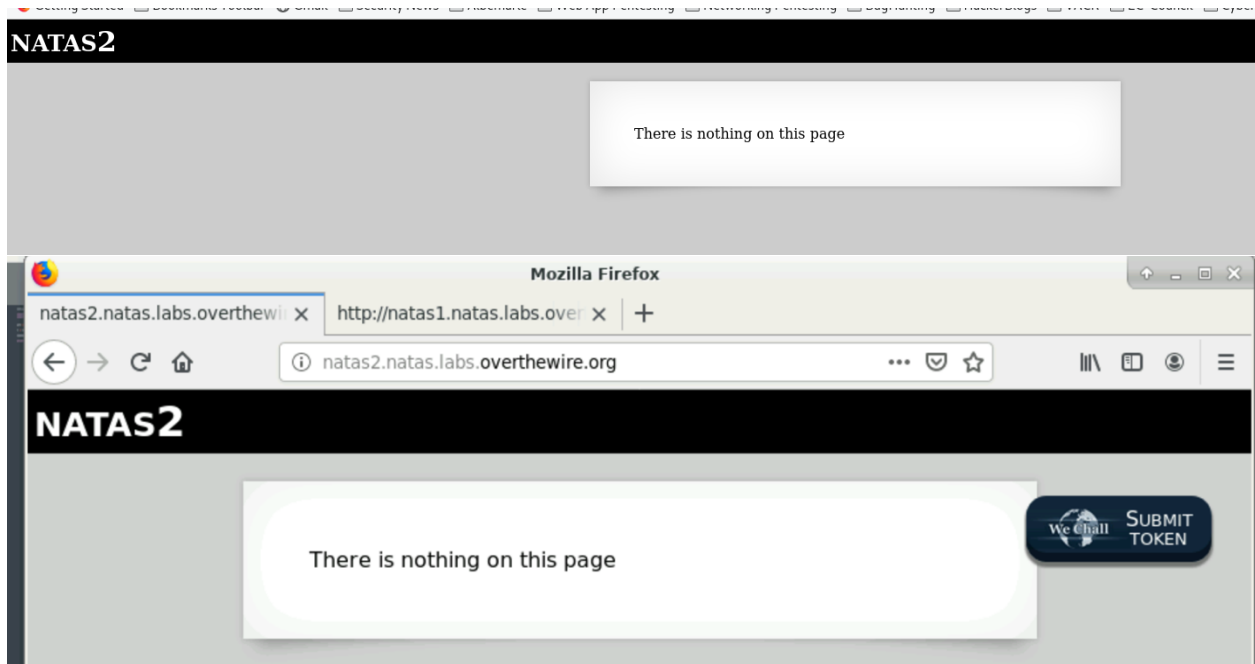
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas0", "pass": "natas0" };</script></head>
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>

```

```
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/we
10 <script>var wechallinfo = { "level": "natas1", "pass": "g9D9cREhslqBKtcA2uocGHPfMZVzeFK6" };</script></head>
11 <body oncontextmenu="javascript:alert('right clicking has been blocked!');return false;">
12 <h1>natas1</h1>
13 <div id="content">
14 You can find the password for the
15 next level on this page, but rightclicking has been blocked!
16
17 <!--The password for natas2 is h4ubbcXrWqsTo7GgnnUMLppXb0ogfBZ7 -->
18 </div>
19 </body>
20 </html>
21
22
```

Task 4: Python Programming Natas 2-3

Visit <http://natas2.natas.labs.overthewire.org/> for the next challenge and use the username natas2 and the password retrieved from the previous task.



- Right click on the page and view the source code


```

1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas2", "pass": "ZluruAthQk7Q2MqmDeTiUij2ZvWy2mBi" };</script></head>
11 <body>
12 <h1>natas2</h1>
13 <div id="content">
14 There is nothing on this page
15 
16 </div>
17 </body></html>
18

```



It appears that nothing is there. Let's check this out in the Python code. First we need to make a few changes. Since the pattern is no longer there we need to take this out for now. We also need to add a new print function.

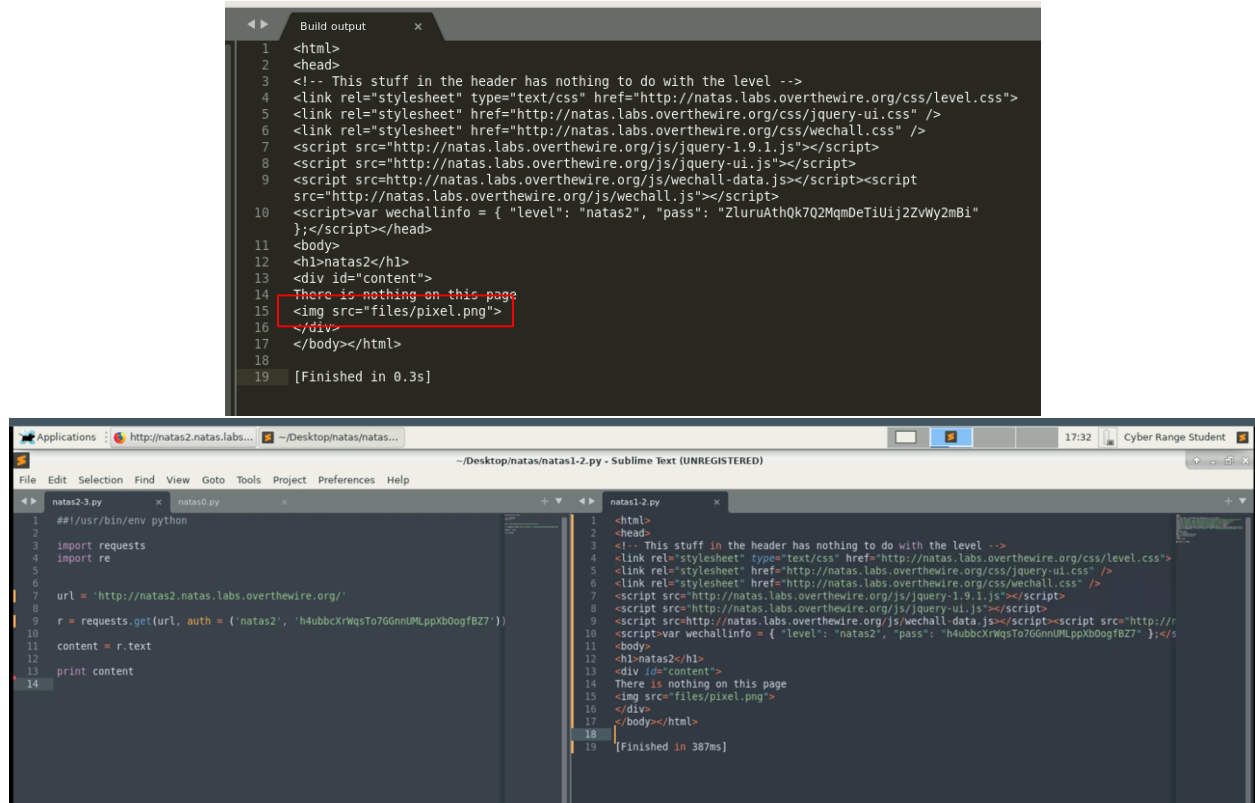
- Change all locations where you have natas1 to natas2.
- Delete the `print re.findall('<!--The password for natas2 is (.*) -->', content) [0]`
- Add `print content` to call the variable content.
- Save this file as `natas2-3.py` and then build the output.

```

natas2-3.py
1 #!/usr/bin/env python
2
3 import requests
4 import re
5
6
7 url = 'http://natas2.natas.labs.overthewire.org/'
8
9 r = requests.get(url, auth = ('natas2', 'ZluruAthQk7Q2MqmDeTiUij2ZvWy2mBi'))
10
11 content = r.text
12
13 print content
14

```

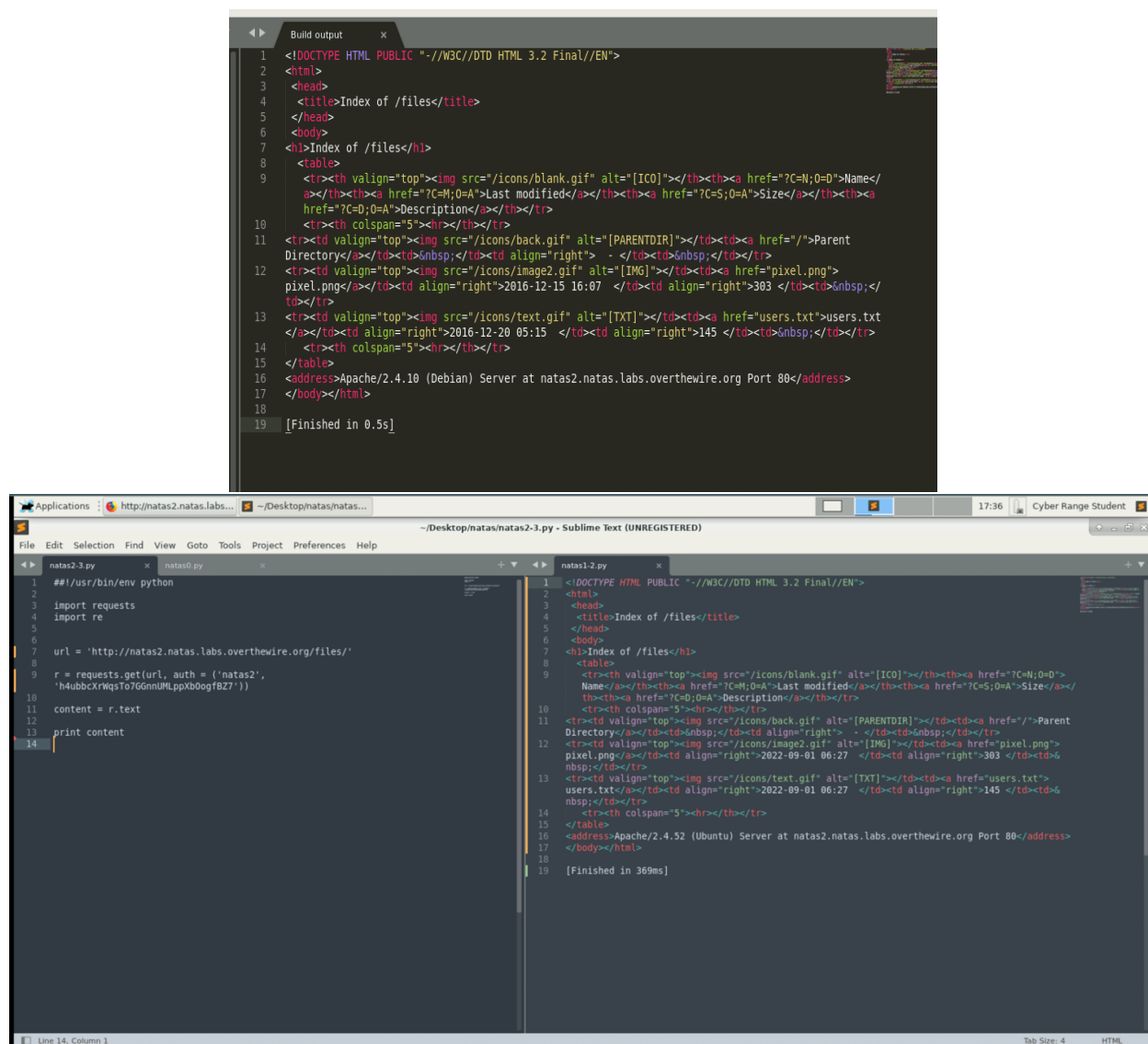
Notice the `img src` location. It appears to be in a directory called `files`.



- Change the url in the natas2-3 tab to match the img src location by appending the url with `/files/`.

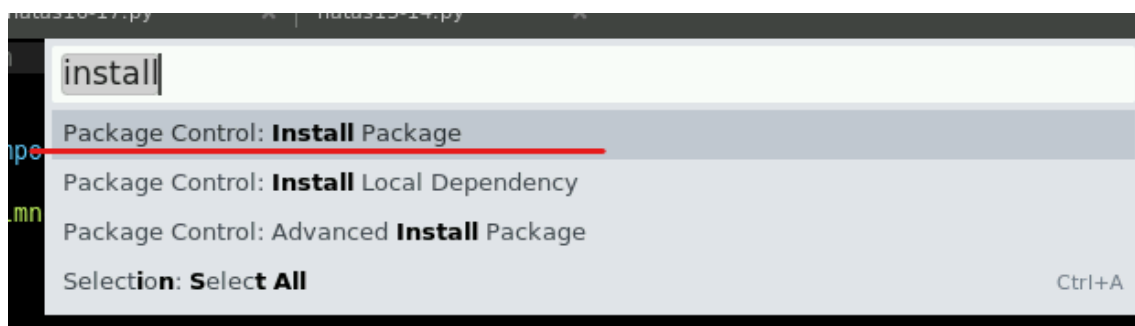
```
1 #!/usr/bin/env python
2
3 import requests
4 import re
5
6
7 url = 'http://natas2.natas.labs.overthewire.org/files/'
8
9 r = requests.get(url, auth = ('natas2', 'ZluruAthQk7Q2MqmDeTiUij2ZvWy2mBi'))
10
11 content = r.text
12
13 print content
14
```

It appears that we are able to navigate to the files folder. Note: if your build output loses its format coloring, use CTRL+SHIFT+P and search for **Set Syntax: HTML**. **IMPORTANT:** Be sure to complete this in the Build output tab.



This is hard to read. Let's install a package that will organize this code a little better.

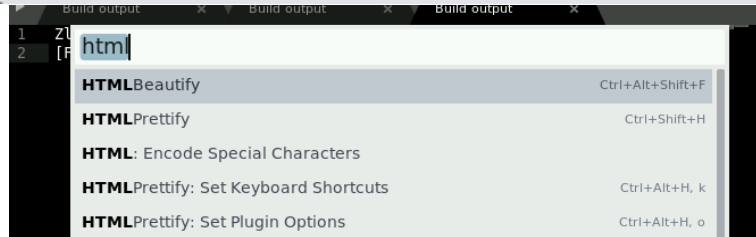
- In the build output tab, press **CTRL+SHIFT+P** and type **install package** and click **Package Control: Install Package** then search for **HTMLBeautify** and click on it to install.
- In the build output tab, press **CTRL+SHIFT+P** and type **HTMLBeautify** and click on it



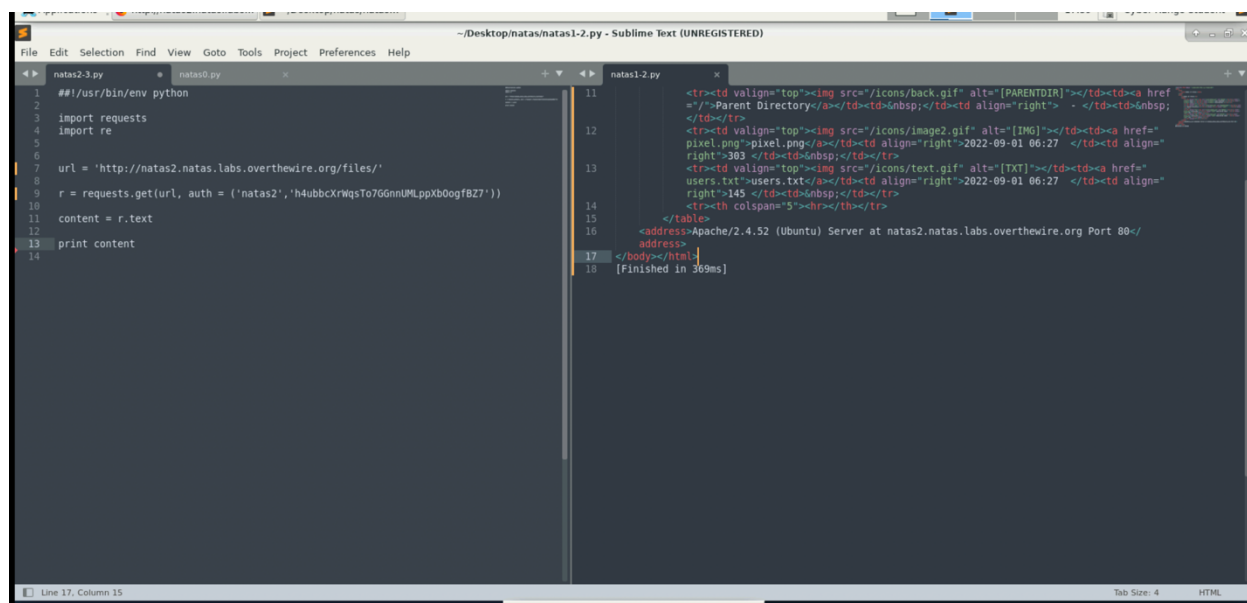
htmlb

HTMLBeautify

A plugin for Sublime Text that formats (indents) HTML source code. It makes code easier for humans
install v2018.03.01.21.41.57; github.com/rareyman/HTMLBeautify



```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
2 <html>
3   <head>
4     <title>Index of /files</title>
5   </head>
6   <body>
7     <h1>Index of /files</h1>
8     <table>
9       <tr><th valign="top"></th><th><a
10         href="?C=N;0=D">Name</a></th><th><a href="?C=M;0=A">Last modified</a></th><
11         th><a href="?C=S;0=A">Size</a></th><th><a href="?C=D;0=A">Description</a></
12         th></tr>
13       <tr><th colspan="5"><hr></th></tr>
14       <tr><td valign="top"></td><td>
15         <a href="/">Parent Directory</a></td><td>&nbsp;</td><td align="right">
16         </td><td>&nbsp;</td></tr>
17       <tr><td valign="top"></td><td><a
18         href="pixel.png">pixel.png</a></td><td align="right">2016-12-15 16:07 </td>
19         <td align="right">303 </td><td>&nbsp;</td></tr>
20       <tr><td valign="top"></td><td><a
21         href="users.txt">users.txt</a></td><td align="right">2016-12-20 05:15 </td>
22         <td align="right">145 </td><td>&nbsp;</td></tr>
23       <tr><th colspan="5"><hr></th></tr>
24     </table>
25     <address>Apache/2.4.10 (Debian) Server at natas2.natas.labs.overthewire.org Port 80
26   </address>
27 </body></html>
28 [Finished in 0.4s]
```



```
1 #!/usr/bin/env python
2
3 import requests
4 import re
5
6 url = 'http://natas2.natas.labs.overthewire.org/files/'
7
8 r = requests.get(url, auth = ('natas2', 'h4ubbcxRwqsTo7GgnnUMLppXb0ogf8Z7'))
9
10 content = r.text
11
12
13 print content
14
```

```
11 <tr><td valign="top"></td><td><a href=
=>/>Parent Directory</a></td><td align="right"> </td><td align="right">
</td></tr>
12 <tr><td valign="top"></td><td><a href=
pixel.png">pixel.png</a></td><td align="right">2022-09-01 06:27 </td><td align="
right">303 </td><td align="right"></td></tr>
13 <tr><td valign="top"></td><td><a href=
users.txt">users.txt</a></td><td align="right">2022-09-01 06:27 </td><td align="
right">145 </td><td align="right"></td></tr>
14 <tr><th colspan="5"><hr></th></tr>
15 </table>
16 <address>Apache/2.4.52 (Ubuntu) Server at natas2.natas.labs.overthewire.org Port 80</
address>
17 </body></html>
18 [Finished in 369ms]
```

This will have to do for now. Take notice of the **href="users.txt."** This looks interesting. Let's visit this location using our code.

```

4      <title>Index of /files</title>
5  </head>
6  <body>
7      <h1>Index of /files</h1>
8      <table>
9          <tr><th valign="top"></th><th><a href="?
C=N;0=D">Name</a></th><th><a href="?C=M;0=A">Last modified</a></th><th><a href="?
C=S;0=A">Size</a></th><th><a href="?C=D;0=A">Description</a></th></tr>
10         <tr><th colspan="5"><hr></th></tr>
11         <tr><td valign="top"></td><td><a href
="/>Parent Directory</a></td><td>&nbsp;</td><td align="right"> - </td><td>&nbsp;</td></tr>
12         <tr><td valign="top"></td><td><a href="
pixel.png">pixel.png</a></td><td align="right">2022-09-01 06:27 </td><td align="
right">303 </td><td>&nbsp;</td></tr>
13         <tr><td valign="top"></td><td><a href="
users.txt">users.txt</a></td><td align="right">2022-09-01 06:27 </td><td align="
right">145 </td><td>&nbsp;</td></tr>
14         <tr><th colspan="5"><hr></th></tr>
15     </table>
16     <address>Apache/2.4.52 (Ubuntu) Server at natas2.natas.labs.overthewire.org Port 80</
address>
17 </body></html>
18 [Finished in 369ms]

```

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
2  <html>
3      <head>
4          <title>Index of /files</title>
5      </head>
6      <body>
7          <h1>Index of /files</h1>
8          <table>
9              <tr><th valign="top"></th><th><a
href="?C=N;0=D">Name</a></th><th><a href="?C=M;0=A">Last modified</a></th><th><a
href="?C=S;0=A">Size</a></th><th><a href="?C=D;0=A">Description</a></th></tr>
10             <tr><th colspan="5"><hr></th></tr>
11             <tr><td valign="top"></td><td>
<a href="/>Parent Directory</a></td><td>&nbsp;</td><td align="right"> - </td><td>&nbsp;</td></tr>
12             <tr><td valign="top"></td><td><a
href="pixel.png">pixel.png</a></td><td align="right">2016-12-15 16:07 </td><td align="
right">303 </td><td>&nbsp;</td></tr>
13             <tr><td valign="top"></td><td><a
href="users.txt">users.txt</a></td><td align="right">2016-12-20 05:15 </td><td align="
right">145 </td><td>&nbsp;</td></tr>
14             <tr><th colspan="5"><hr></th></tr>
15         </table>
16         <address>Apache/2.4.10 (Debian) Server at natas2.natas.labs.overthewire.org Port 80
</address>
17 </body></html>
18 [Finished in 0.4s]

```

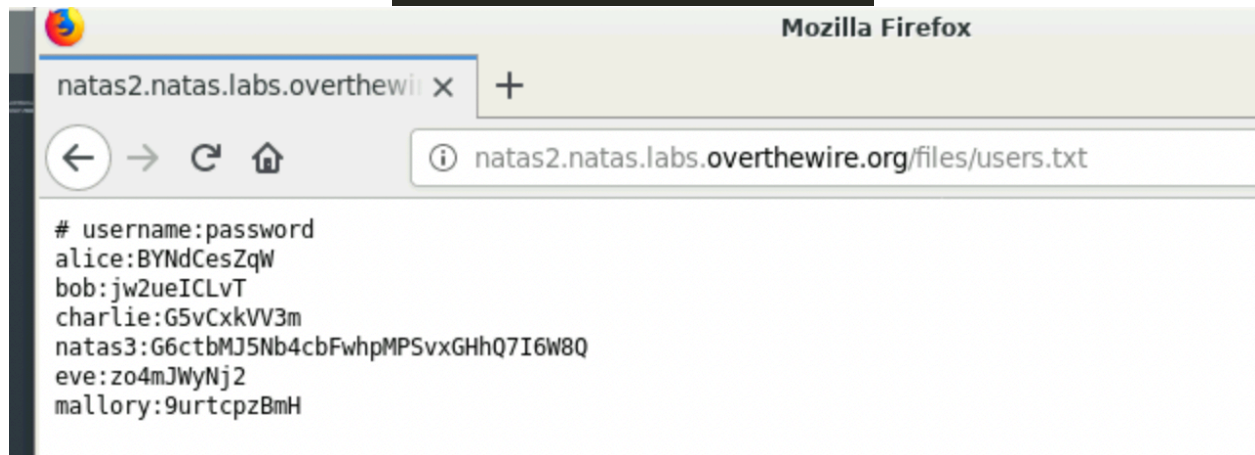
- In the natas2-3.py tab, change the url to match <http://natas2.natas.labs.overthewire.org/files/users.txt>
- Save the file and then build the program.

```

1  #!/usr/bin/env python
2
3  import requests
4  import re
5
6
7  url = 'http://natas2.natas.labs.overthewire.org/files/users.txt'
8
9  r = requests.get(url, auth = ('natas2', 'ZluruAthQk7Q2MqmDeTiUij2ZvWvy2mBi'))
10
11  content = r.text
12
13  print content
14

```

```
Build output x
1 # username:password
2 alice:BYNdCesZqW
3 bob:jw2ueICLvT
4 charlie:G5vCxxVV3m
5 natas3:sJIJNW6ucpu6HPZ1ZAchaDtdw7oGrD14
6 eve:zo4mJWyNj2
7 mallory:9urtcpzBmH
8
9 [Finished in 0.3s]
```



There you have it, the next password for natas3.

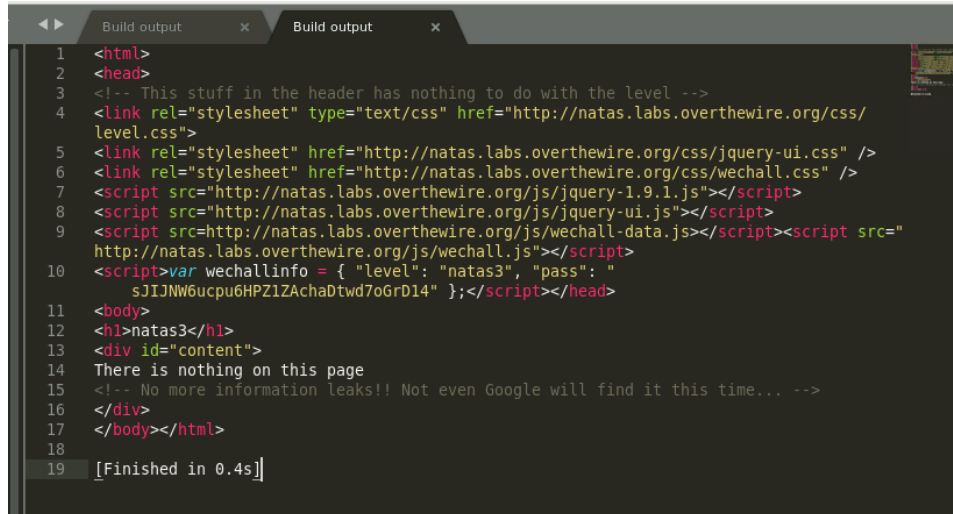
Task 5: Python Programming Natas 3-4

As completed in previous tasks, we want to change the Python code to match our new parameters.

- Take a look at the screenshot to check if you have all the parameters correct.

```
natas2-3.py x
1 #!/usr/bin/env python
2
3 import requests
4 import re
5
6
7 url = 'http://natas3.natas.labs.overthewire.org'
8
9 r = requests.get(url, auth = ('natas3', 'sJIJNW6ucpu6HPZ1ZAchaDtdw7oGrD14'))
10
11 content = r.text
12
13 print content
14
```

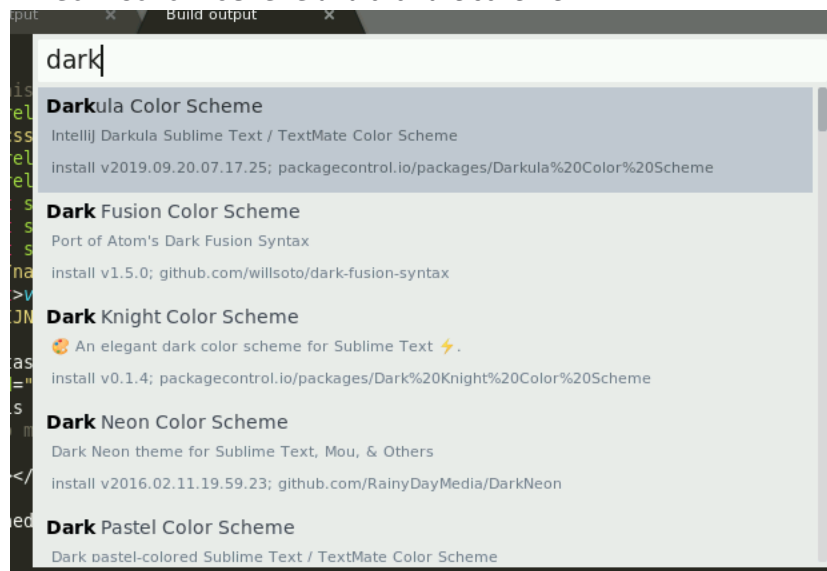
- Save the file as `natas3-4.py` and then build the program to look at the response.



```
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/
  level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="
  http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas3", "pass": "
    sJIJNW6ucpu6HPZ1ZAchaDtwD7oGrD14" };</script></head>
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>
18
19 [Finished in 0.4s]
```

I am not liking this color scheme as it is hard on the eyes. Let's use the package controller to install a better color scheme.

- Press **CTRL+SHIFT+P** and type **Package Control: Install Package**
- Click **Package Control: Install Package**
- Type **Dark Neon Color Scheme** and click the scheme.



- Press **CTRL+SHIFT+P** and type **UI: Select Color Scheme** and click the result.
- Click on **Dark Neon** (or the color scheme that you prefer).

Now let's look at the Build output.


```
Build output x Build output x
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="
http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas3", "pass": "
sJIJNW6ucpu6HPZ1ZAchaDtwD7oGrD14" };</script></head>
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>
18
19 [Finished in 1.4s]
```

I am not a fan of the inefficiencies of moving to the browser to complete the challenges. I will show some screenshots, but will not be walking through the browser method; however, at this point, you should be able to test these on your own if you are interested. Here is what it looks like on the webpage.

```
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas3", "pass": "sJIJNW6ucpu6HPZ1ZAchaDtwD7oGrD14" };</script></head>
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>
```

This hint should be easy to understand. If Google is not allowed to spider an application, this is defined in the robots.txt file. We covered this in earlier lessons. So, we should check to see if the page has a robot.txt file. From this point forward, **build the output** will mean you need to **press CTRL+B**.

- In the `natas3-4.py` file, append the url with `/robots.txt`, save the file, and then build the output.

```
Build output x Build
1 User-agent: *
2 Disallow: /s3cr3t/
3
4 [Finished in 0.3s]
```

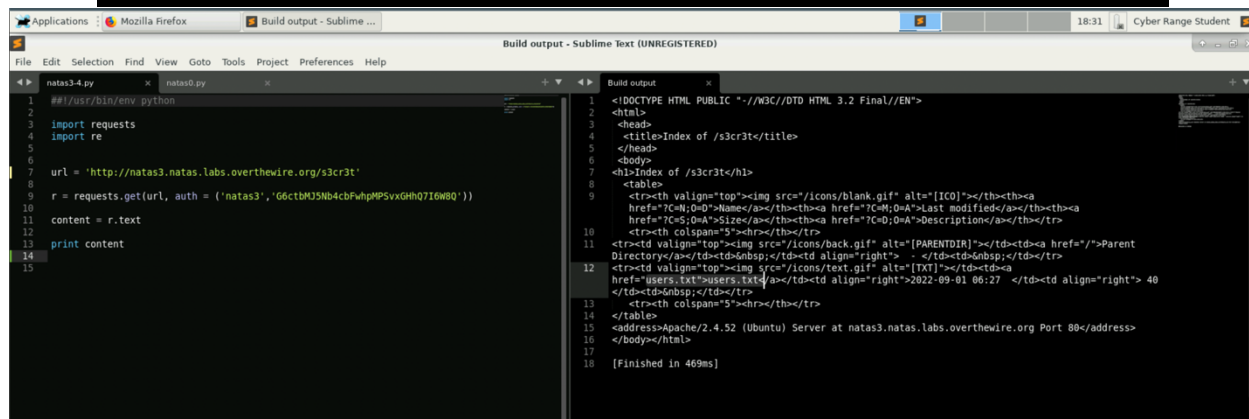
There seems to be a file that is not allowed to be crawled. Let's navigate to the location in our Python script.

- Change the url to `http://natas3.natas.labs.overthewire.org/s3cr3t`, save the file, and then build the output.

```
File Edit Selection Find View Goto Tools Project Preferences Help
natas2-3.py x
1  #!/usr/bin/env python
2
3  import requests
4  import re
5
6
7  url = 'http://natas3.natas.labs.overthewire.org/s3cr3t'
8
9  r = requests.get(url, auth = ('natas3', 'sJIJNW6ucpu6HPZ1ZAchaDtwD7o6rD14'))
10
11  content = r.text
12
13  print content
14
```

There appears to be another **users.txt** file. Let's navigate to this location in our Python script.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
  <head>
    <title>Index of /s3cr3t</title>
  </head>
  <body>
    <h1>Index of /s3cr3t</h1>
    <table>
      <tr><th valign="top"></th><th><a
href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><
th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></
th></tr>
      <tr><th colspan="5"><hr></th></tr>
      <tr><td valign="top"></td><td>
<a href="/">Parent Directory</a></td><td align="right"> -
</td><td align="right">
</td><td align="right">
</td></tr>
      <tr><td valign="top"></td><td><a
href="users.txt">users.txt</a></td><td align="right">2016-12-20 05:15 </td>
<td align="right"> 40 </td><td align="right">
</td></tr>
      <tr><th colspan="5"><hr></th></tr>
    </table>
    <address>Apache/2.4.10 (Debian) Server at natas3.natas.labs.overthewire.org Port 80
    </address>
  </body></html>
[Finished in 0.4s]
```



Natas 3 password: G6ctbMJ5Nb4cbFwhpMPsvxGHhQ7I6W8Q

- Change the url to `http://natas3.natas.labs.overthewire.org/s3cr3t/users.txt`, save the file, and then build the output.

The top screenshot shows a Python script named `natas3-4.py` in the Sublime Text editor. The script uses the `requests` library to fetch data from `http://natas3.natas.labs.overthewire.org/s3cr3t/users.txt` using a specific authentication token. The output window shows the result: `natas4:Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ`.

The bottom screenshot shows a similar script named `natas0.py` (though the tab says `natas3-4.py`). This script uses a different authentication token to fetch data from the same URL. The output window shows the result: `natas4:tk0cJlbzM4lTs8hbCmzn5Zr4434f6ZQm`.

Now we have the password for natas4! Be sure to save the script as **natas3-4.py**.

Task 6: Python Programming Natas 4-5

As completed in previous tasks we want to change the Python code to match our new parameters.

The screenshot shows a Python script named `natas4-5.py` in the Sublime Text editor. The script uses the `requests` library to fetch data from `http://natas4.natas.labs.overthewire.org` using a specific authentication token. The output window shows the result: `natas4:Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ`.

- Using the **CTRL+SHIFT+S**, save the file as **natas4-5.py** and then build the output.

The screenshot shows a web browser window displaying an HTML page. The page content is as follows:

```
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="
10 http://natas.labs.overthewire.org/js/wechall.js"></script>
11 <script>var wechallinfo = { "level": "natas4", "pass": "
12 Z9tkRkKmpt9Qr7XrR5jWRkg0U901swEZ" };</script></head>
13 <body>
14 <h1>natas4</h1>
15 <div id="content">
16 Access disallowed. You are visiting from "" while authorized users should come only
17 from "http://natas5.natas.labs.overthewire.org/"
18 <br/>
19 <div id="viewsource"><a href="index.php">Refresh page</a></div>
20 </div>
21 </body>
22 </html>
```

The browser shows an error message: "Access disallowed. You are visiting from "" while authorized users should come only from "http://natas5.natas.labs.overthewire.org/". Below the error message is a link: "Refresh page".

The background shows a Python script in a text editor:

```
1 #!/usr/bin/env python
2
3 import requests
4 import re
5
6 url = 'http://natas4.natas.labs.overthewire.org'
7
8 r = requests.get(url, auth = ('natas4', 'tK0cJ1bzM4lTs8hbCmzn5Zr4434fGZ0m'))
9
10 content = r.text
11
12 print content
13
14
15
```

This looks like a header issue. We know this because the site is referring to us visiting from "" which is null. This means we are missing a referrer in our header. We could fire up BurpSuite and capture the request and then change the header; however, this is not very efficient and we are already set up in Python. According to <https://2.python-requests.org/en/master/user/quickstart/#custom-headers> we can just add a header request into our script.

```
>>> url = 'https://api.github.com/some/endpoint'
>>> headers = {'user-agent': 'my-app/0.0.1'}
>>> r = requests.get(url, headers=headers)
```

source: <https://2.python-requests.org/en/master/user/quickstart/#custom-headers>

We need a referer header and it needs to come from the specified location.

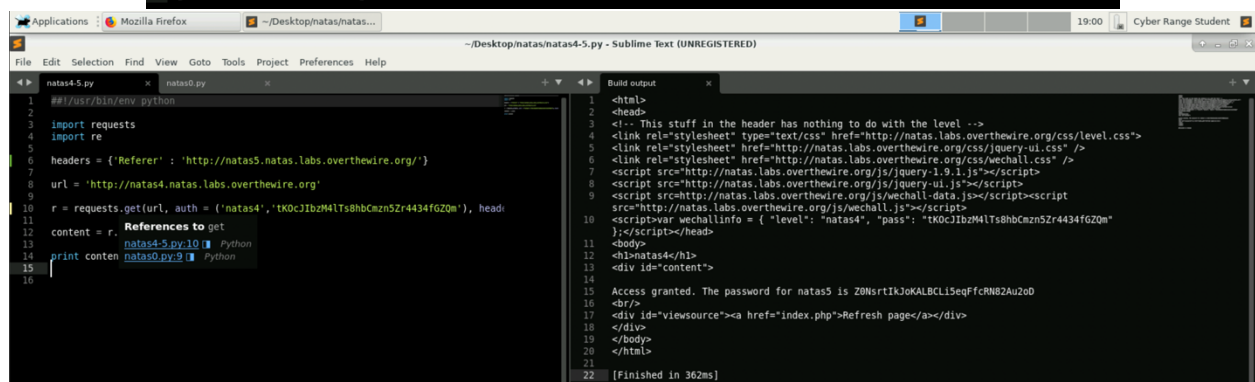
- In the `natas4-5.py` file, add `headers = { 'Referer' : 'http://natas5.natas.labs.overthewire.org/' }` as show in the screenshot below.
- Next, add `, headers = headers` inside the parentheses as show in the screenshot below, save the file, then build the output.

```
1 #!/usr/bin/env python
2
3 import requests
4 import re
5
6 headers = {'Referer': 'http://natas5.natas.labs.overthewire.org/'}
7
8 url = 'http://natas4.natas.labs.overthewire.org'
9
10 r = requests.get(url, auth=('natas4', 'Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ'), headers = headers)
11
12 content = r.text
13
14
15 print content
16
```

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas4", "pass": "Z9tkRkWmpt9Qr7XrR5jWRkgOU901swEZ" };</script></head>
<body>
<h1>natas4</h1>
<div id="content">

Access granted. The password for natas5 is ix6IOfmpN7AY0QGpWtn3fXpbaJVJcHfQ
<br/>
<div id="viewsource"><a href="index.php">Refresh page</a></div>
</div>
</body>
</html>

[Finished in 0.3s]
```



Natas 4 password: tkOcJIbzM4lTs8hbCmzn5Zr4434fGZQm

Let's narrow this script down even more.

- In the `natas4-5.py` file, comment out `print content` by adding `#` before it. This means when we build the output, this command will be ignored.
- Add `print re.findall('The password for natas5 is (.*)', content) [0]` below this comment. See image below.
- Save the script as `natas5-6.py` and then build the output.

```
nat5-6.py x
1  #!/usr/bin/env python
2
3  import requests
4  import re
5
6  headers = {'Referer' : 'http://nat5.natas.labs.overthewire.org/'}
7
8  url = 'http://nat4.natas.labs.overthewire.org'
9
10 r = requests.get(url, auth=('nat4', 'Z9tkRkKmpt9Qr7XrR5jWRkg0U901swEZ'), headers = headers )
11
12 content = r.text
13
14 #print content
15
16 print re.findall('The password for nat5 is (.*)', content) [0]
17
```

Natas 5: Z0NsrtIkJoKALBCLi5eqFfcRN82Au2oD

```
Build output x Build output x
1  iX6IOfmpN7AY0QGpwn3fXpbaJVJcHfq
2  [Finished in 1.5s]
```

And there we have it! The password for natas5.

Task 7: Python Programming Natas 5-6

As completed in previous tasks, we want to change the Python code to match our new parameters. We also want to comment out the line 16 the `print re.findall...` and the `headers` on line 6. We then want to delete the `#` (comment) on the `print content`. We want to add a `) #` on line 10. There are a lot of changes in this script, so they are shown in the red boxes in the screenshot below. The `#` is called commenting out and can be used for quick changes in the code. Make sure you save the file after making all the edits.

```
nat5-6.py x
1  #!/usr/bin/env python
2
3  import requests
4  import re
5
6  #headers = {'Referer' : 'http://nat5.natas.labs.overthewire.org/'}
7
8  url = 'http://nat5.natas.labs.overthewire.org'
9
10 r = requests.get(url, auth=('nat5', 'iX6IOfmpN7AY0QGpwn3fXpbaJVJcHfq')) #, headers = headers )
11
12 content = r.text
13
14 print content
15
16 #print re.findall('The password for nat5 is (.*)', content) [0]
```

- Build the output.

```
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas5", "pass": "ix6IOfmpN7AY0QGpWtn3fXpbaJVJcHfq" };</script></head>
11 <body>
12 <h1>natas5</h1>
13 <div id="content">
14 Access disallowed. You are not logged in</div>
15 </body>
16 </html>
17
18 [Finished in 0.4s]
```

```
1 #!/usr/bin/env python
2
3 import requests
4 import re
5
6 #headers = {'Referer': 'http://natas5.natas.labs.overthewire.org/'}
7
8 url = 'http://natas5.natas.labs.overthewire.org'
9
10 r = requests.get(url, auth = ('natas5', 'ix6IOfmpN7AY0QGpWtn3fXpbaJVJcHfq')) #, headers = headers )
11
12 content = r.text
13
14 print content
15
16 #print re.findall('The password for natas5 is (.*)', content) [0]
```

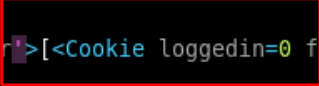
According to the output, we do not have access to the page. This means that the page is using cookies. Again we could use Burp Suite to capture a request and look at the cookies; however, we can do a quick check with `r.cookies`. Take a look at this site for more details on writing Python code and cookies: <https://2.python-requests.org/en/master/user/quickstart/#cookies>

- In the `natas5-6.py` file, add `print (r.cookies)`, save the file, and then build the output. See image below.

```
1 #!/usr/bin/env python
2
3 import requests
4 import re
5
6 #headers = {'Referer': 'http://natas5.natas.labs.overthewire.org/'}
7
8 url = 'http://natas5.natas.labs.overthewire.org'
9
10 r = requests.get(url, auth = ('natas5', 'ix6IOfmpN7AY0QGpWtn3fXpbaJVJcHfq')) #, headers = headers )
11
12 content = r.text
13
14 print content
15
16 print (r.cookies)
17
18 #print re.findall('The password for natas5 is (.*)', content) [0]
```



```
10 <script>var wechallinfo = { "level": "natas5", "pass": "ix6IOfmpN7AY0QGpWtn3fXpbaJVJcHfq" };</script></head>
11 <body>
12 <h1>natas5</h1>
13 <div id="content">
14 Access disallowed. You are not logged in</div>
15 </body>
16 </html>
17
18 <<class 'requests.cookies.RequestsCookieJar'>[Cookie loggedin=0 for
natas5.natas.labs.overthewire.org/>]>
19 [Finished in 0.3s]
```



```
1 #!/usr/bin/env python
2
3 import requests
4 import re
5
6 #headers = {'Referer': 'http://natas5.natas.labs.overthewire.org/'}
7
8 url = 'http://natas5.natas.labs.overthewire.org'
9
10 r = requests.get(url, auth = ('natas5', 'Z0NsrtIkJoKALBCLISeqFFcRN82Au20'))#_he
11
12 content = r.text
13
14 print content
15
16 print (r.cookies)
17
18 #print re.findall('The password for natas5 is (.*)', content) [0]
19
```

```
1 <html>
2 <head>
3 <!-- This stuff in the header has nothing to do with the level -->
4 <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
5 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
6 <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
7 <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
8 <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
9 <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script
src="http://natas.labs.overthewire.org/js/wechall.js"></script>
10 <script>var wechallinfo = { "level": "natas5", "pass": "Z0NsrtIkJoKALBCLISeqFFcRN82Au20"
};</script></head>
11 <body>
12 <h1>natas5</h1>
13 <div id="content">
14 Access disallowed. You are not logged in</div>
15 </body>
16 </html>
17
18 <<class 'requests.cookies.RequestsCookieJar'>[Cookie loggedin=0 for
natas5.natas.labs.overthewire.org/>]>
19 [Finished in 361ms]
```

It appears that the cookie name is **loggedin** and it currently **=0**. It looks like all we need to do is add a cookie named **loggedin** and set the value to **1**. In the **natas5-6.py** file, do the following:

- Comment out **print (r.cookies)**.
- Add **cookies = {'loggedin' : '1'}** under the url.
- Edit the **r = requests.get** function call by adding in **cookies=cookies** after url, and **exactly** as shown in the below image.
- Add **r.cookies = ['loggedin']** under **r = requests.get...** line 13 in the image below.
- Comment out **print content**.
- Uncomment the **print re.findall** line and change **natas5** to **natas6**.
- add **</div>** after the **(/*)** in the **print** function
- Save the file and then build the output.

```

1  #!/usr/bin/env python
2
3  import requests
4  import re
5
6  #headers = {'Referer' : 'http://natas5.natas.labs.overthewire.org/'}
7
8  url = 'http://natas5.natas.labs.overthewire.org'
9  cookies = {'loggedin' : '1'}
10
11 r = requests.get(url, cookies=cookies, auth=('natas5', 'ix6I0fmpN7AY0QGpwn3fXpbaJVJcHfq')) #, headers = headers )
12
13 r.cookies = ['loggedin']
14
15 content = r.text
16
17 #print content
18
19 #print (r.cookies)
20
21 print re.findall('The password for natas6 is (.*)</div>', content) [0]
22

```

There we have it, the level six password.

```

1  aGoY4q2Dc6MgDq4oL4YtoKtyAg9PeHa1
2  [Finished in 0.3s]

```

The screenshot shows the Sublime Text editor with a file named 'natas5-6.py'. The code in the editor is as follows:

```

1  #!/usr/bin/env python
2
3  import requests
4  import re
5
6  #headers = {'Referer' : 'http://natas5.natas.labs.overthewire.org/'}
7
8  url = 'http://natas5.natas.labs.overthewire.org'
9  cookies = {'loggedin' : '1'}
10
11 r = requests.get(url, cookies=cookies, auth=('natas5', 'Z0NsrtIkJoKALBCLiSeqFfcf
12
13 r.cookies = ['loggedin']
14
15 content = r.text
16
17 #print content
18
19 #print (r.cookies)
20
21 print re.findall('The password for natas6 is (.*)</div>', content) [0]
22

```

The Build output panel on the right shows the following output:

```

1  f0IvE0MDtPTgRhqmmvvaOt2EfXR6uQgR
2  [Finished in 356ms]

```

Natas 6 password: f0IvE0MDtPTgRhqmmvvaOt2EfXR6uQgR

In this lesson, we learned how to set up and use Sublime Text, and how to use Python to parse information from a web application. In the process, we gained access to a site by changing the referrer and we also discovered a cookie session name and modified the cookie to gain access.