



**eLearnSecurity**  
Forging security professionals

# BURP SUITE



PRELIMINARY SKILLS | SECTION 1 MODULE 3 | LAB #5

**LAB**



# 1. DESCRIPTION

A local police department has hired you to pentest their website. They had a new website created by a web development company and they want to make sure that everything is secure and in order.

In this lab you will practice with Burp Suite, configuring the scope of the engagement, intercepting the communications with a webserver and *spidering* a target web application. You can access the target web application at the following address **10.100.13.5**.

# 2. GOAL

The goal of this lab is to test the given web application in order to find a hidden path that contains a restricted area. Once the hidden path is discovered, your goal will be to bypass the authentication exploiting a “*feature*” left over by the developers while “debugging” the area.

# 3. TOOLS

The best tools for this lab are:

- Web browser
- Burp Suite



## 4. STEPS

### 4.1. EXPLORE THE WEB APPLICATION

Explore the Web application at the address [10.100.13.5](http://10.100.13.5) and verify that everything works as intended. You should see the Police Department website.

### 4.2. CONFIGURING YOUR ARSENAL

Before starting analyzing the target application, configure your browser and Burp Suite. Do not forget to configure the ***scope of the engagement*** in order to analyze only requests and responses that belong to that scope, and filter the site map to show ***only in-scope*** items.

At the end of the configurations, perform a test to make sure that everything works as intended.

### 4.3. MAPPING THE TARGET APPLICATION

Some resources are hidden and performing an active crawling of the application, by following links, submitting forms, parsing responses, etc. requires time. Find a way to automate all of these operations in one click!

### 4.4. THE KEYSTONE

The automated mapping should have revealed a ***hidden path***. Explore the path and extract useful information to reach your goal. You should note that developers are full of “magic tricks”, find the one used in this application and you will find the **keystone**.



# SOLUTIONS

Please go ahead **ONLY** if you have **COMPLETED** the lab or you are stuck! Checking the solutions before actually trying the concepts and techniques you studied in the course, will dramatically reduce the benefits of a hands-on lab!



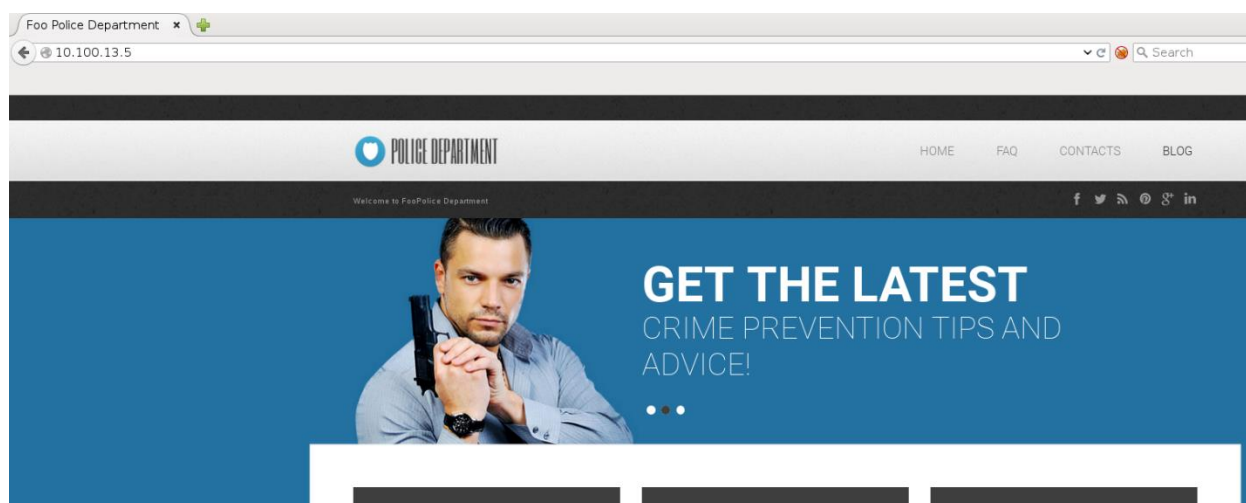
[This page intentionally left blank]



# 5. SOLUTIONS STEPS

## 5.1. EXPLORE THE WEB APPLICATION

You should see a page like this:



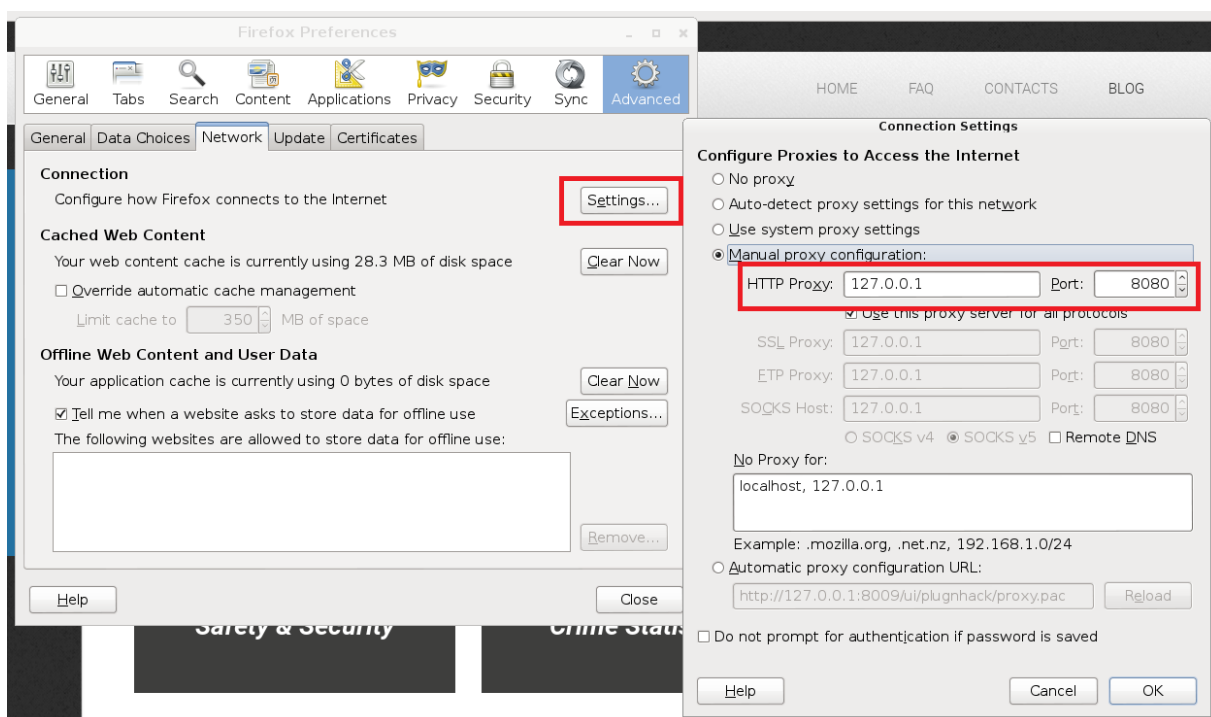
Welcome to **Foo Police Department!**



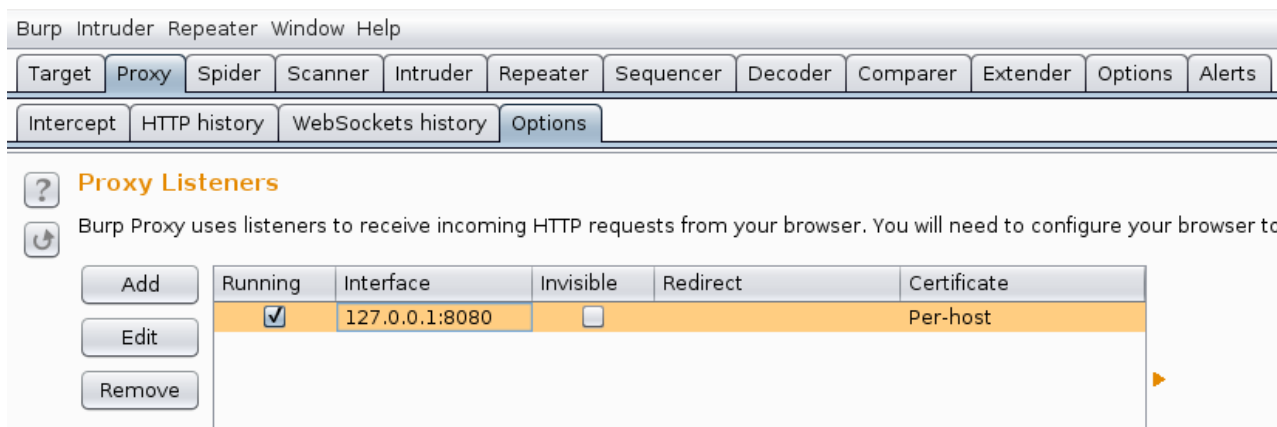
## 5.2. CONFIGURING YOUR ARSENAL

In order to analyze the traffic, spider the targeted web application and discover the hidden path of the restricted area. You need to setup the proxy both in your browser and in Burp Proxy.

### *In your browser*



### *In Burp Proxy*



In addition to the listener, it's a best practice to configure the proxy to intercept request and responses that belongs to the **targets in scope**:

CA certificate ...

### Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☒ Intercept requests based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
<input checked="" type="checkbox"/>		File extension	Does not match	(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$ ...
<input type="checkbox"/>	Or	Request	Contains parameters	
<input type="checkbox"/>	Or	HTTP method	Does not match	(get post)
<input checked="" type="checkbox"/>	And	URL	Is in target scope	

☐ Automatically fix missing or superfluous new lines at end of request  
☒ Automatically update Content-Length header when the request is edited

### Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

☐ Intercept responses based on the following rules:

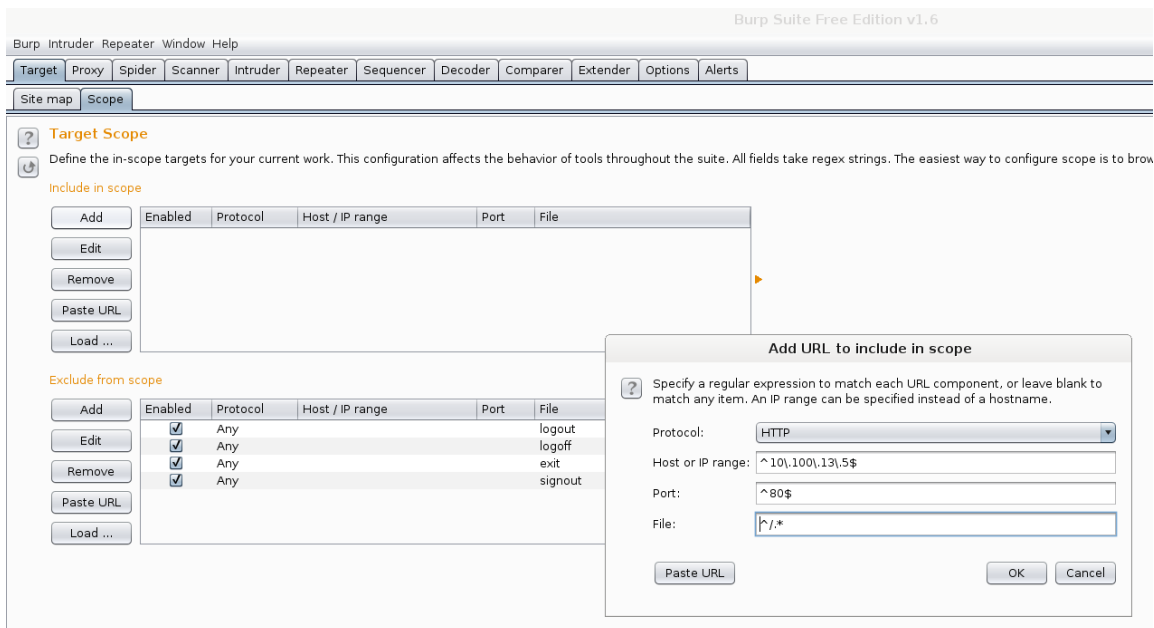
Enabled	Operator	Match type	Relationship	Condition
<input checked="" type="checkbox"/>		Content type he...	Matches	text
<input type="checkbox"/>	Or	Request	Was modified	
<input type="checkbox"/>	Or	Request	Was intercepted	
<input type="checkbox"/>	And	Status code	Does not match	^304\$
<input checked="" type="checkbox"/>	And	URL	Is in target scope	

☒ Automatically update Content-Length header when the response is edited

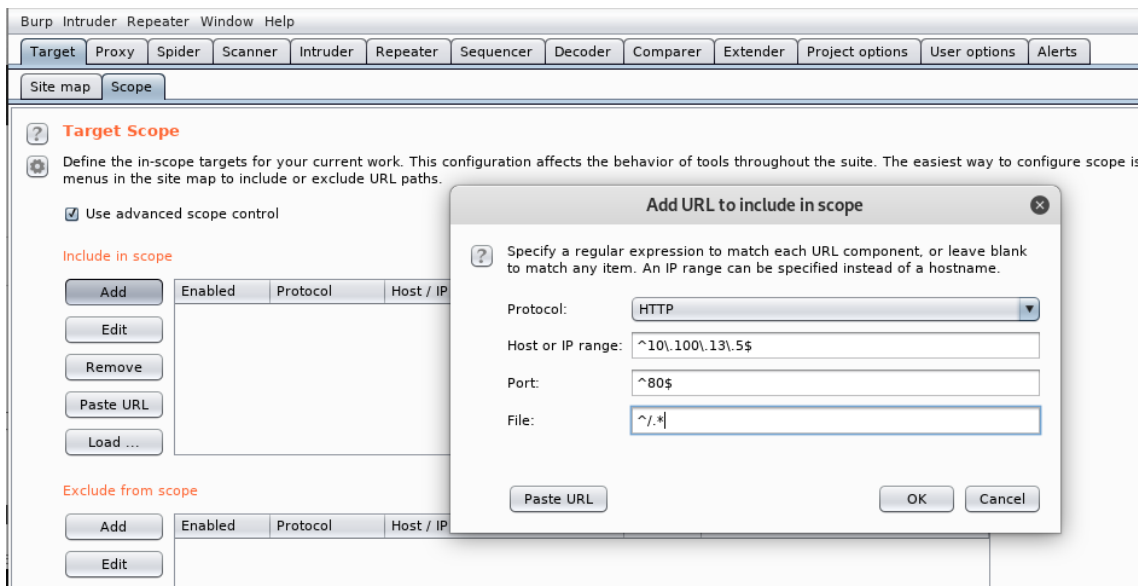




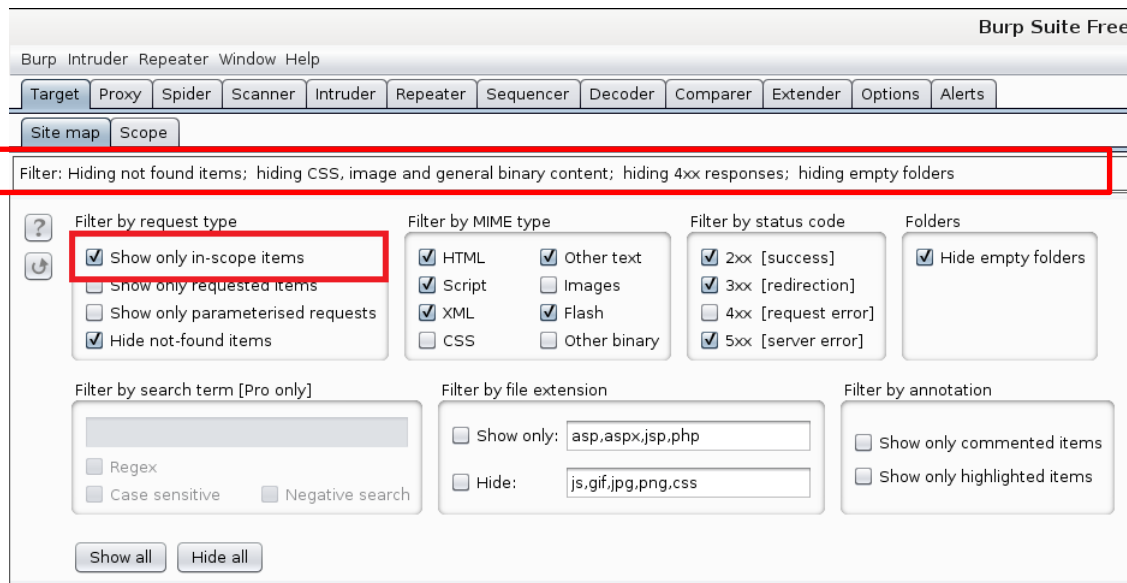
To configure the scope of engagement browse the tab **Target** and then **Scope**. To add a URL to the scope you can paste the link or type it manually.



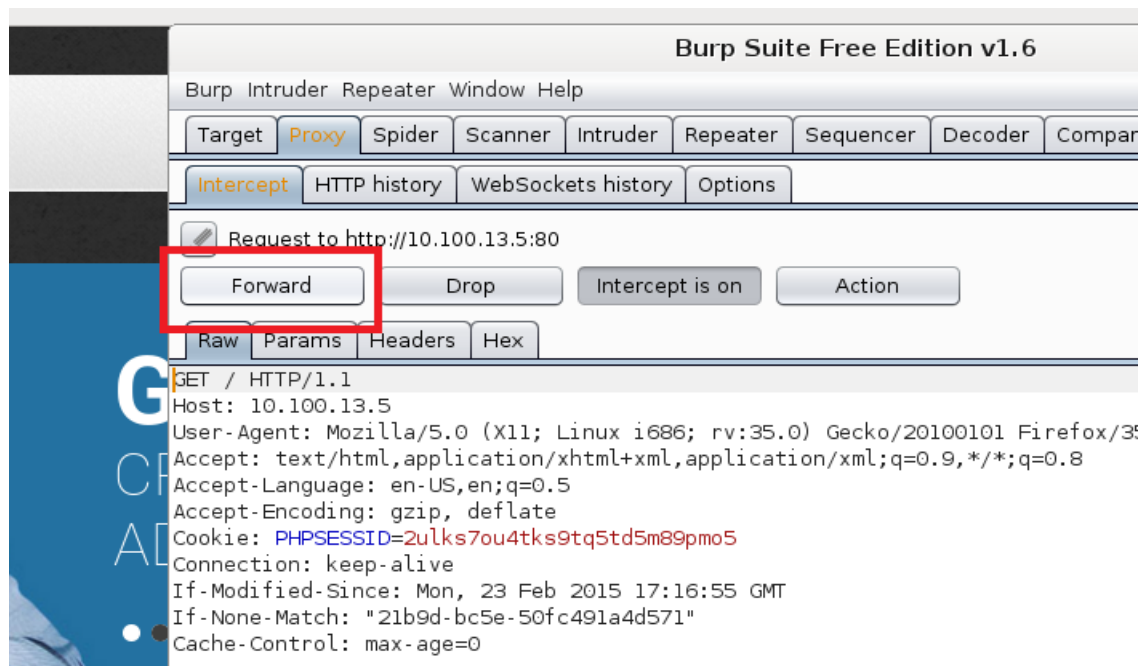
In latest version of Burp Suite, you will need to click the “Use advanced scope control” checkbox before you can specify the scope in that form, as shown in the screenshots below:



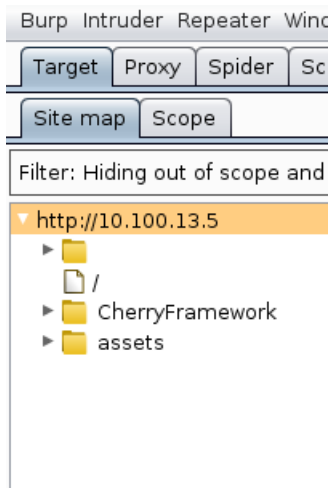
In the **site map**, configure the filter by request type adding a tick to “*Show only in-scope items*”. This will show you only the resources that belong to the scope defined previously.



To test if your configurations are working as intended, just refresh the link into the browser and verify that the intercept has captured your request. If not, be sure the *Intercept* button is toggled.

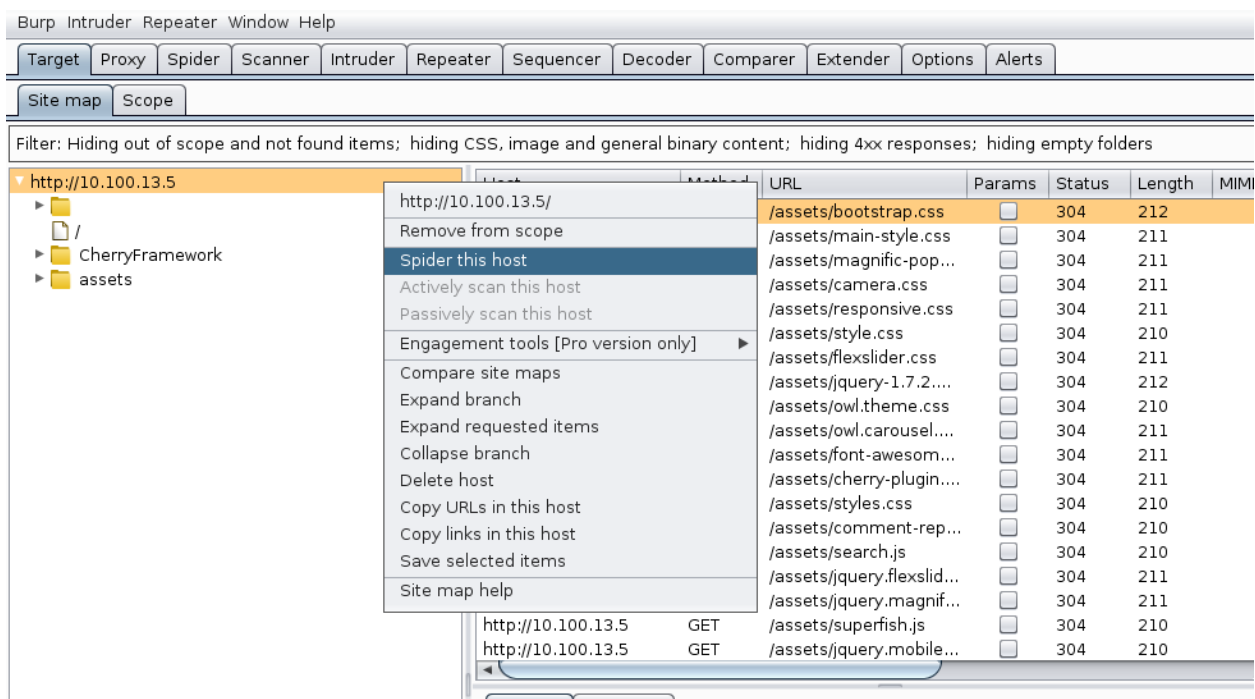


Once forwarded all the requests and responses, you should see the list of the resources exchanged in the *Target > Site map* tab:

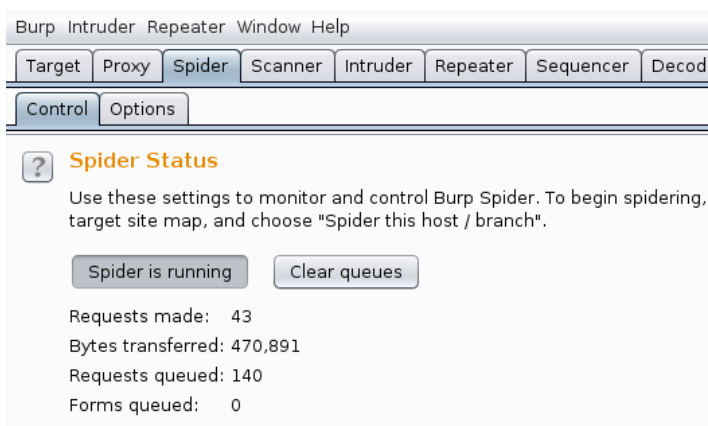


## 5.3. MAPPING THE TARGET APPLICATION

In order to automatically map the target web application we can use the Burp Spider tool. To do this, just right click on the target host in the site map list. Then select “**Spider this host**”:



In the Spider tab you'll see the status of this operation:



After a while, you should see a list of paths on the **Site Map** that were not listed before. One of them is the hidden area we are looking for:

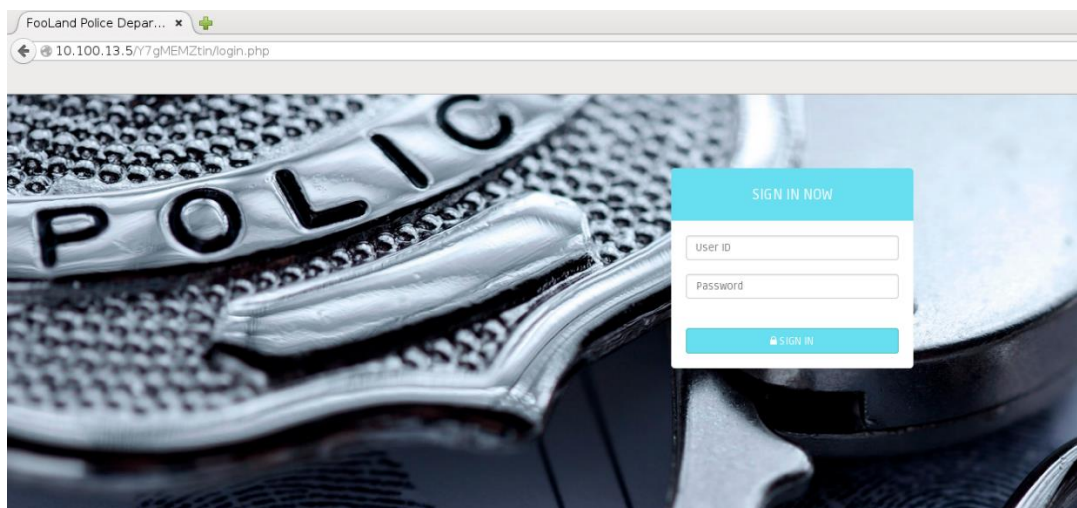
The screenshot shows the Burp Suite interface with the Site Map tab selected. The Site Map tree on the left shows a new folder named **Y7gMEMZtin** highlighted with a red rectangle. The main table on the right lists the discovered paths and their details.

Host	Method	URL	Params	Status	Length	MIME type
http://10.100.13.5	GET	/Y7gMEMZtin/		302	371	
http://10.100.13.5	GET	/Y7gMEMZtin/login.php		200	5514	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/...		200	2201	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...		200	1347	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...		200	1655	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/...		200	2098	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...		200	1535	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...		200	1412	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/js/		200	7229	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/...		200	7002	script
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...		200	1171	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/js/		200	4338	script
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...		200	16045	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...		200	32112	script
http://10.100.13.5	GET	/Y7gMEMZtin/assets/...	<input checked="" type="checkbox"/>	200	2201	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/...	<input checked="" type="checkbox"/>	200	2201	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/...	<input checked="" type="checkbox"/>	200	2201	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/...	<input checked="" type="checkbox"/>	200	2201	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...	<input checked="" type="checkbox"/>	200	1347	HTML



## 5.4. THE KEYSTONE

Visiting the *hidden path*, you should notice that the application exposes an authentication page. It requires a login and you don't have one.



The next step needs to analyze this page in order to find something useful to bypass the authentication.

Analyzing the server response to the ***login.php*** resource, you should have noticed that at the end of the file there is a debugging message. The developers implemented a simple login bypass to avoid the authentication during the debugging operations and forgot to remove the message in production.



Burp Intruder Repeater Window Help  
 Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts  
 Site map Scope  
 Filter: Hiding out of scope and not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status	Length	MIME type
http://10.100.13.5	GET	/Y7gMEMZtin/		302	371	
http://10.100.13.5	GET	/Y7gMEMZtin/login.php		200	5514	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/...		200	2201	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...		200	1347	HTML
http://10.100.13.5	GET	/Y7gMEMZtin/assets/f...		200	1655	HTML

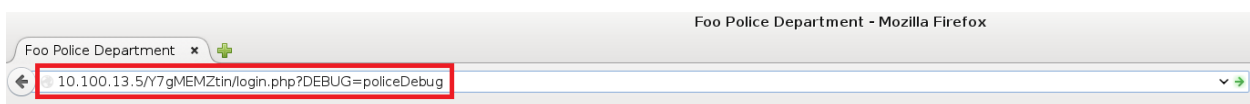
Site map: http://10.100.13.5  
 - /  
 - CherryFramework  
 - Y7gMEMZtin  
 - assets  
 - blog.html  
 - contacts.html  
 - faq.html  
 - robots.txt

Request Response  
 Raw Headers Hex HTML Render

```

<!--
*****
*****
*****  ALERT: Remove this in production!!  *****
*****
*****
DEBUGGIN MODE
To avoid the authentication page, send in the query string DEBUG=policeDebug
e.g. login.php?DEBUG=policeDebug
-->
  
```

Requesting the login path with the parameters suggested by the developers:



You will access the restricted area and reach the goal of this lab!

You also notify your client of your findings and successfully close your engagement.

