Session Hijacking

In session hijacking attacks, the attacker takes advantage of insecure session identifiers, finds a way to obtain them, and uses them to authenticate to the server and impersonate the victim.

An attacker can obtain a victim's session identifier using several methods, with the most common being:

- Passive Traffic Sniffing
- Cross-Site Scripting (XSS)
- Browser history or log-diving
- Read access to a database containing session information

As mentioned in the previous section, if a session identifier's security level is low, an attacker may also be able to brute force it or even predict it.

Session Hijacking Example

Proceed to the end of this section and click on `Click here to spawn the target system!`. Use the provided Pwnbox or a local VM with the supplied VPN key to reach the target application and follow along. Don't forget to configure the specified vhost (`xss.htb.net`) to access the application.

A quick way to specify this (and any other) vhost in your attacking system is the below:

Session Hijacking

```
MisaelMacias@htb[/htb]$ IP=ENTER SPAWNED TARGET IP HERE
MisaelMacias@htb[/htb]$ printf "%s\t%s\n\n" "$IP" "xss.htb.net csrf.htb.net oredirect.htb.net
```

Part 1: Identify the session identifier

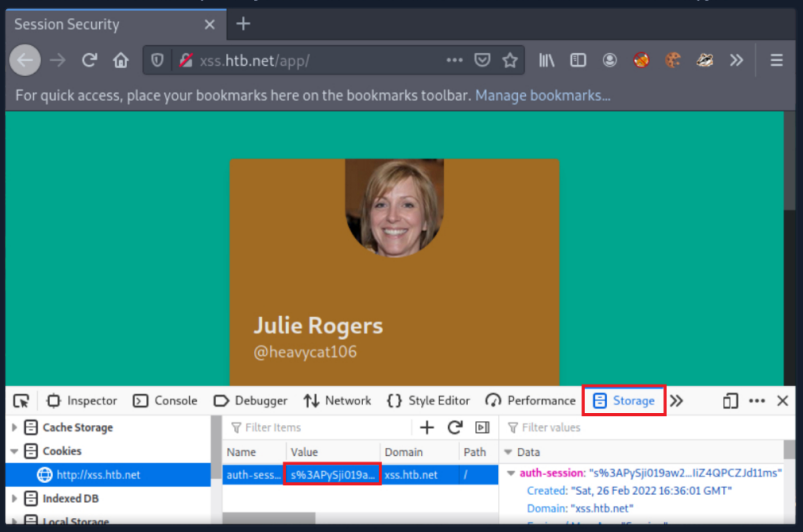
Navigate to `http://xss.htb.net` and log in to the application using the credentials below:

- Email: heavycat106
- Password: rocknrol

This is an account that we created to look into the application!

You should now be logged in as "Julie Rogers."

Using Web Developer Tools (Shift+Ctrl+I in the case of Firefox), notice that the application is using a cookie named `auth-session` most probably as a session identifier. Double click this cookie's value and copy it!



Part 2: Simulate an attacker

Now, suppose that you are the attacker and you somehow got access to the `auth-session` cookie's value for the user "Julie Rogers".

Go to Questions

Table of Contents

Introduction to Sessions

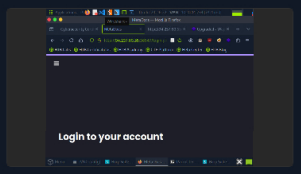
Session Attacks

Session Hijacking
Session Fixation
Obtaining Session Identifiers without User Interaction
Cross-Site Scripting (XSS)
Cross-Site Request Forgery
Cross-Site Request Forgery (GET-based)
Cross-Site Request Forgery (POST-based)
XSS & CSRF Chaining
Exploiting Weak CSRF Tokens
Additional CSRF Protection Bypasses
Open Redirect
Remediation Advice

Skills Assessment

Session Security - Skills Assessment

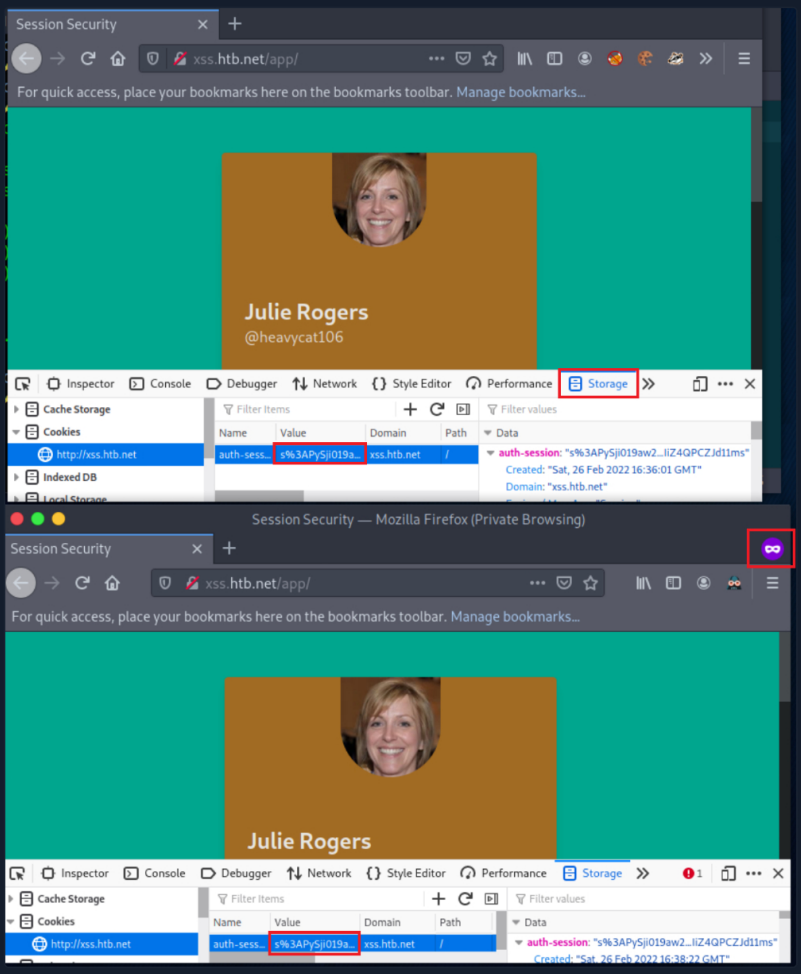
My Workstation



Interact Terminate Reset

Life Left: 40m

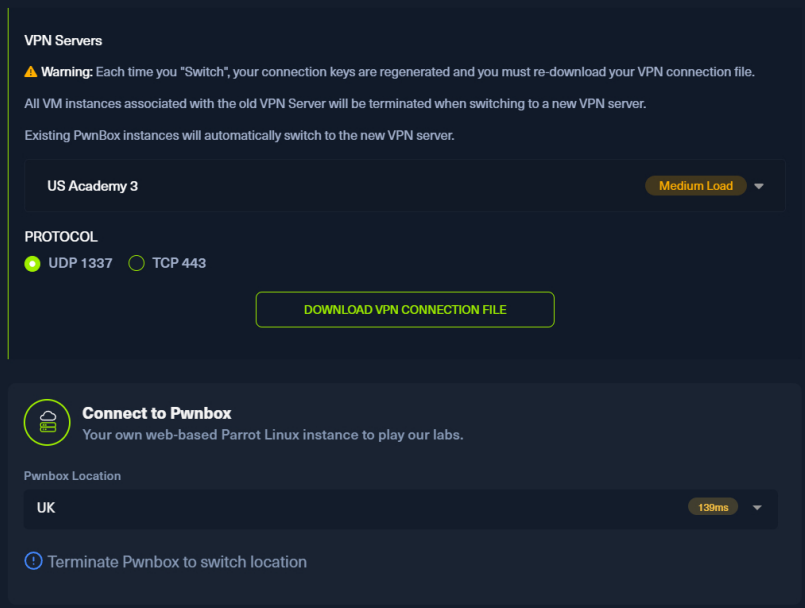
Open a `New Private Window` and navigate to `http://xss.htb.net` again. Using Web Developer Tools (Shift+Ctrl+I in the case of Firefox), replace the current `auth-session` cookie's value with the one you copied in Part 1. Reload the current page, and you will notice that you are logged in as "Julie Rogers" without using any credentials!

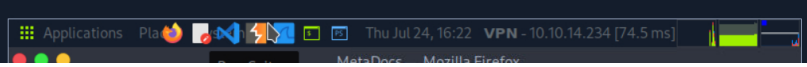


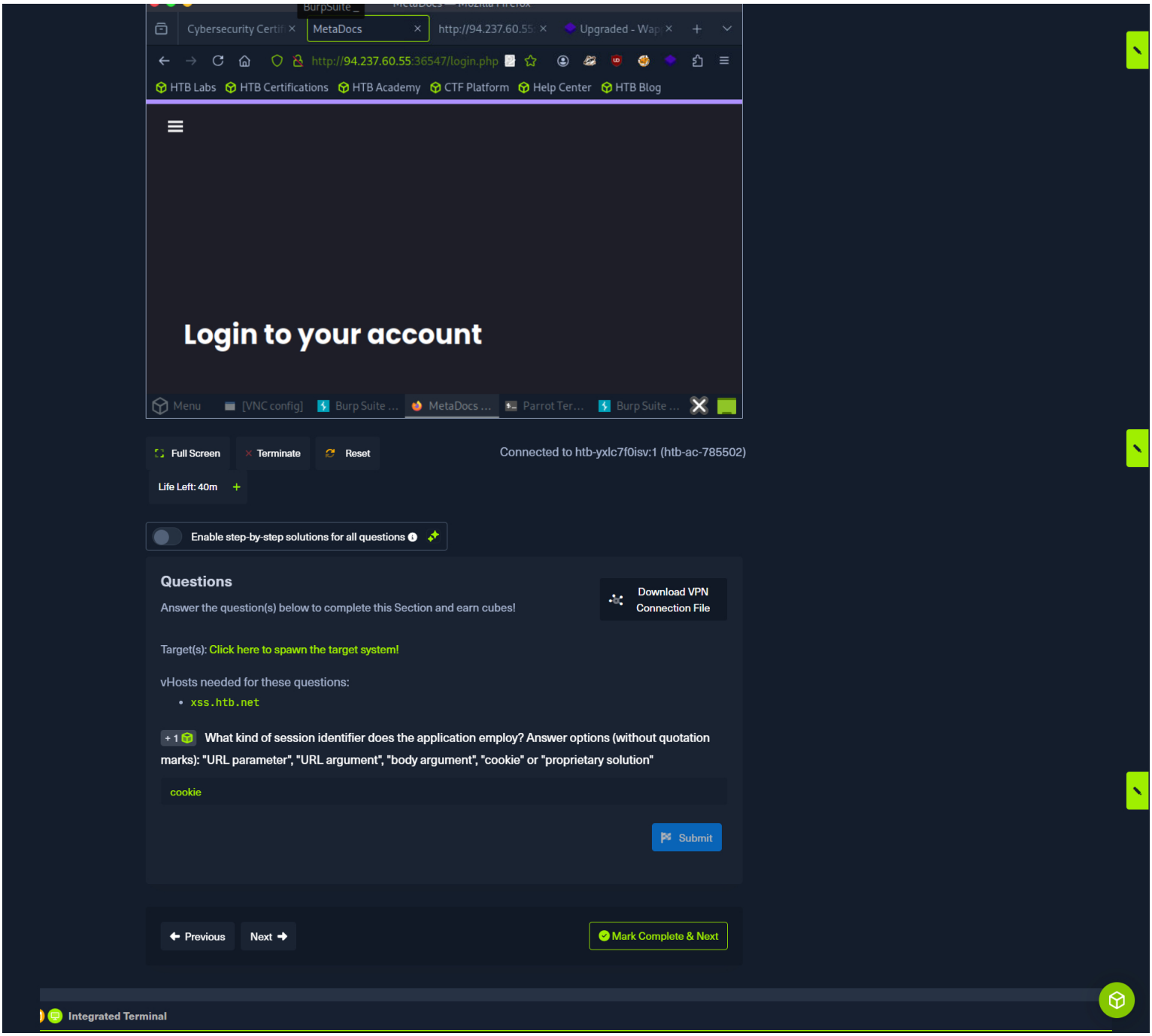
Congratulations! You just practiced your first session hijacking attack!

Please note that you could come across web applications that utilize more than one cookie for session tracking purposes.

In the following sections, we will cover how you can mount the most common session attacks in detail.






BurpSuite
Cybersecurity Certif
MetaDocs
http://94.237.60.55
Upgraded - Wap
http://94.237.60.55:36547/login.php
HTB Labs
HTB Certifications
HTB Academy
CTF Platform
Help Center
HTB Blog
Login to your account
Menu
[VNC config]
Burp Suite ...
MetaDocs ...
Parrot Ter...
Burp Suite ...
Full Screen
Terminate
Reset
Connected to htb-yxlc7f0isv:1 (htb-ac-785502)
Life Left: 40m
Enable step-by-step solutions for all questions
Questions
Answer the question(s) below to complete this Section and earn cubes!
Download VPN Connection File
Target(s): Click here to spawn the target system!
vHosts needed for these questions:
xss.htb.net
+ 1
What kind of session identifier does the application employ? Answer options (without quotation marks): "URL parameter", "URL argument", "body argument", "cookie" or "proprietary solution"
cookie
Submit
Previous
Next
Mark Complete & Next
Integrated Terminal