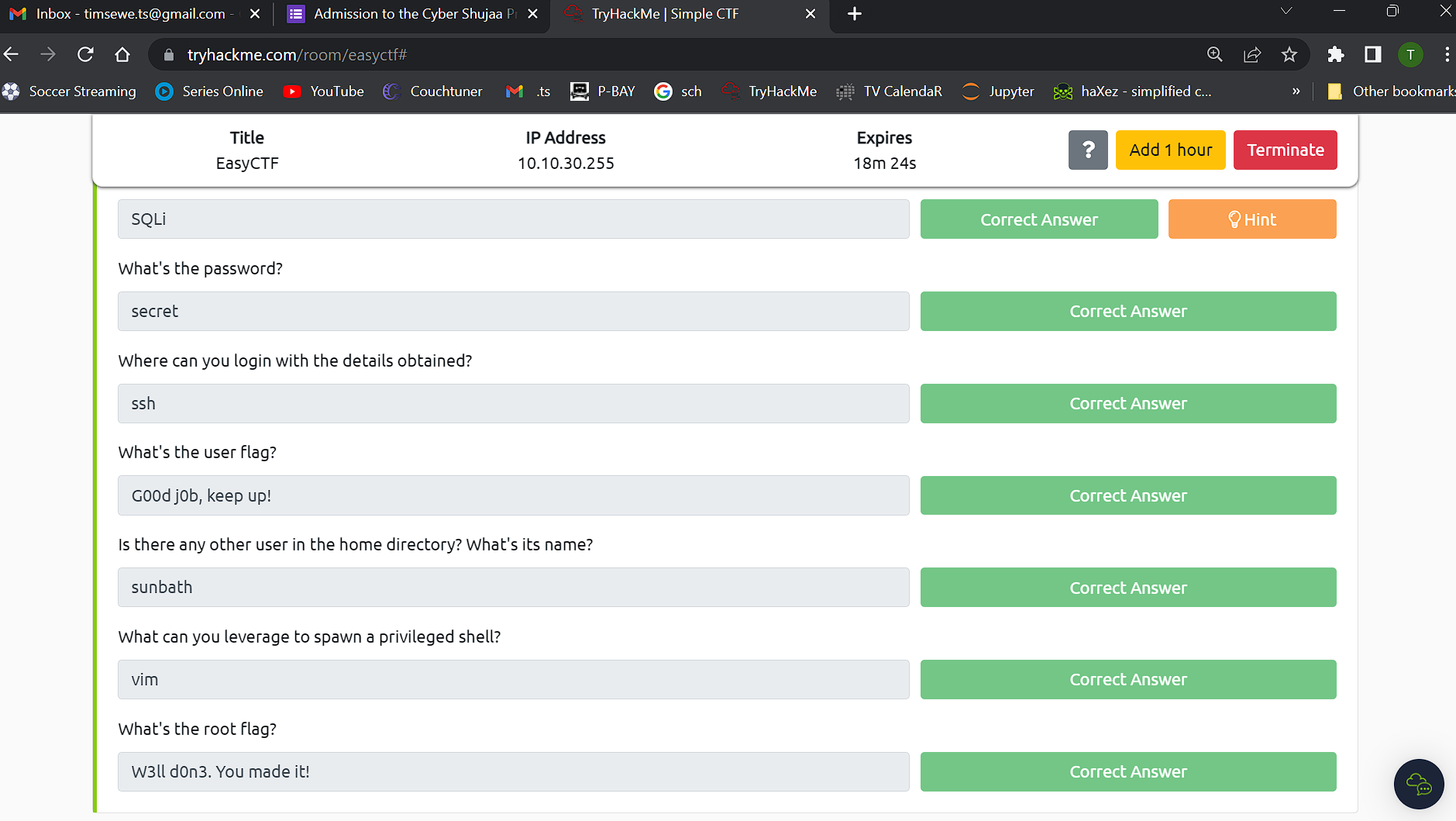
1. Were you able to complete the challenge? (Yes/No) Take a screen-shot showing your answers and completion level for the tasks.

Yes, I was able to complete the challenge.



2. What obstacles did you face while completing the challenge and how did you attempt to overcome them?

I didn’t face much serious obstacles but just the trick when submitting the root flag. If you weren’t keen enough you wouldn’t have noticed that they were using zero in place for letter O. A silly obstacle I would say but nonetheless it made me retype it every time thinking the page had an error.

3. What did you learn from this challenge?

From this challenge, I learned several important concepts and techniques in the field of cybersecurity and penetration testing:

1. Scanning and enumeration: The initial step in assessing a target system involves scanning for open ports and services. In this challenge, the nmap tool was used to identify ports 21 (FTP), 80 (HTTP), and 2222 (SSH) as open.
2. Web reconnaissance: Upon discovering the open port 80, further investigation was conducted by browsing the website hosted on that port. It was found to be the default Apache2 page, providing limited information.
3. Web page enumeration: To uncover additional pages or directories on the website, the "gobuster" tool was utilized. It successfully identified a webpage located at "/simple," which was then accessed for further exploration.
4. CMS Made Simple vulnerability: The "/simple" webpage was found to be a default page for the CMS Made Simple application, specifically version 2.2.8. By searching online for vulnerabilities related to this version, a known SQL injection vulnerability (CVE-2019-9053) was discovered.
5. Exploiting the vulnerability: Armed with knowledge of the SQL injection vulnerability, an exploit script was obtained and executed. The script utilized the CVE-2019-9053 vulnerability to extract information from the target system. The exploit was successful, yielding a username ("sunbath") and password ("secret").
6. Privilege escalation: After obtaining valid credentials, an attempt was made to log in to the target machine using SSH. Successful authentication granted access to the user's home directory, where the user flag ("user.txt") was found.
7. Privilege escalation via sudo: Running the command "sudo -l" revealed that the user "mitch" had permission to execute /usr/bin/vim without requiring a password. This information was leveraged to escalate privileges.
8. GTFOBins: GTFOBins (Get The F\*\*\* Out Binary) is a curated collection of Unix binaries that can be exploited for privilege escalation. By referring to GTFOBins, a command involving "vim" was identified as a means to spawn a privileged shell.
9. Obtaining root access: Following the GTFOBins instructions, a privileged shell was spawned. With escalated privileges, the root flag ("root.txt") was discovered in the root directory, signifying the completion of the challenge.

Overall, this challenge provided a beginner-level introduction to various fundamental concepts in CTFs, including scanning, enumeration, web reconnaissance, vulnerability research, exploitation, and privilege escalation.