**COMP 116 Fall 2020 - Capture The Flag**

**Group 16: Sam Berman, Rob Lester, Ryan Dreher, Alex Maleno**

Submission Key: YUx6am2F

Executive Summary

We spent a week finding vulnerabilities on a system. Many hours were spent parsing through the CTF site as a team, looking for discrepancies, oddities, or anything that just looked strange. In Capture The Flag, the mystery locations seem to encourage motivation in itself. Each flag caused random spurts of creativity, a new path to travel down in an attempt to solve the answer. When someone found something interesting, all four of us would devote our attention to that aspect in order to analyze and figure out if there might be a key hidden somewhere in there. We used many web security and basic vulnerability tactics. Here is more specifically what we did for the challenges we solved:

**Challenge 1:**

Methodology: We realized that anything refreshing of the page is essentially a GET request, so we just appended the word “FLAG” to the url based on the hint which provided a download file which held the key.

Path: <http://34.74.7.86/flag>

Key{9a9b1e01094c99c4458bcbcb438fceb0423da19ddcfcbeb0affe01ace047bbcc}

Graphical user interface, text, application, chat or text message

Description automatically generated

**Challenge 2:**

Methodology: We downloaded the two images of fauci found by traversing through site, ran strings and hexdump on both, and found that logo.jpg hexdump (command xxd) revealed a key in the text on the right hand side.

Path: <http://34.74.7.86/logo.jpg>

Key{0ad3c4cef5cc0ceca6da3163d4de6e8f15b5f4f36284b7c6}

Text

Description automatically generated

**Challenge 3:**

Methodology:

In this question we used our big brains and ran some Nmap on the website's IP. At first, one may think “this is a website not a machine, why does it have an IP?” Such concepts did not matter to us during this moment of curiosity. This predictive push, this attentive advance, led us to a Kali VM where we tried Wireshark, Netcat, tshark, TCPDump, and the like to get some clues.

But once we ran Nmap, that’s when our brains clicked. The neurons fired once we saw port 80 was open, and nmap found a Git repository! To the internet we went, and after clicking on Firefox in the VM enough times to crash VirtualBox, we visited 34.74.7.86/.git/ chrome on our host machine. There, we saw a file named FLAG. Bingo. Back to the VM we went to download and open the file. Inside was the key. Boomshakalaka.

Path:<http://34.74.7.86/.git/FLAG>

Key{640fb2489f4765d5de57dc99c14953b1fb6955f3768c2c60f7f30d41ea00f8e4}

A screenshot of a computer

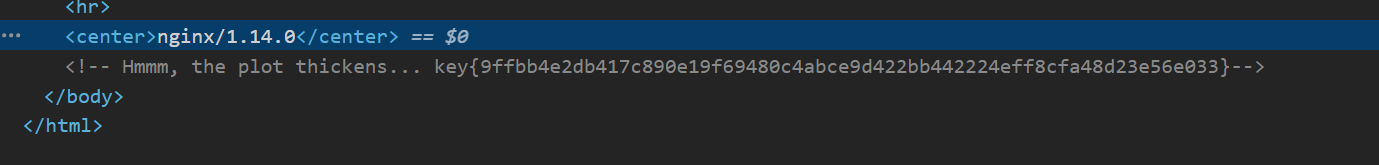
Description automatically generated with medium confidence

**Challenge 5:**

Methodology: Inspecting the source code of the 404 page on the Fake Voter Registration page. Easy-peasy.

Path: <http://34.74.7.86/main.php>

Key{9ffbb4e2db417c890e19f69480c4abce9d422bb442224eff8cfa48d23e56e033}



**Challenge 7:**

Methodology: After logging in to the admin page, we noticed that there was a login.php, and figured “why not try to go to a logout.php?” This took us to the logout page, but only after we had previously logged in on the admin page. Once there, the key was in plain sight.

Path:  <http://34.74.7.86/logout.php>

key{57c6a1585882d10781010702b485f896abac245bad61e96e4ffc2bd92dcd9442}

Graphical user interface, application

Description automatically generated

**Challenge 8:**

Methodology: We knew that the Fake Voter Registration page would have some type of SQL injection involved with it. To actually access the page we used `' or '1'='1` as both the username and the password, which because of SQL injection always returns true, and therefore gave us access to the voter site. Once there, we played around with different inputs for a few minutes before trying the same string `' or '1'='1` on all the open fields, which worked, producing a list of all the voters in the database. From there, we simply Cmd-F’d for “key{“ , which produced the screenshot below.

Path: <http://34.74.7.86/main.php> and <https://35.211.66.169/>

Key{6c15abf0c60717485a059ee41610b8fd1c9e45021707a43c171d5961de14f811}

Table

Description automatically generated

**Challenge 9: README**

Methodology: We looked up the different contents of a Wordpress site, and found out that there is a page in every site directory called readme.html, so we tried navigating to readme.html in the URL and found the key.

Path: <http://34.74.7.86/readme.html>

key{ca5ee4f9735e64f881a9b64dae465ce793e817d324580b0207645cb47b161ce4}

Graphical user interface, text, application, email

Description automatically generated

**Challenge 10:**

Methodology: We looked up the architecture of a Wordpress website and found that they have a folder called wp-includes. Found that the frankenstein.jpg was in a subdirectory of this folder, so we traversed backwards from 34.7.74.86/wp-content/uploads/2015/10/frankenstein.jpg to 34.7.74.86/wp-content/uploads/2015/ and found a mislabeled file in 34.7.74.86/wp-content/uploads/2015/11 (was ASCII text but labeled .pcap) containing a flag at 34.7.74.86/wp-content/uploads/2015/11/ppe.pcap

key{88111518921f4173aa0810343a0ee306dce22c9d3142794a523d1e3449897161}

Text

Description automatically generated

Lessons Learned

1. Figuring out the problem - defining the scope and details of the problem well is the first step in attacking it. Once you learn enough about your target and the way it’s designed, then you can start to leverage specific knowledge about specific details and vulnerabilities of a target.
2. When it rains it pours--usually when you are able to find one specific vulnerability which can be broadly applied to different aspects of a target, it is usually a good idea to explore that further because it can lead to other vulnerabilities often.
3. It’s important to think more generally rather than trying specific tools and practices without any context--it was relatively useless to blindly use the different security and network analysis tools that we used this semester without a broader understanding of the problem as a whole. Rather than just procedural testing and exploitation of a target, it is far easier to go about this challenge by finding tools for specific problems rather than problems for specific tools.
4. Websites can be a lot more insecure than they seem--at first glance, it doesn’t look like there is much than can be done to the website, but with knowledge of the architecture and through exploitation practices, they can be far less secure than first appearance.
5. All of the tools fit together--throughout this course we have used different tools which have looked at and exploited vulnerabilities at different levels of the OSI, and in this project we were able to see how much of this came together in different capacities, for example, when navigating into the wp\_users database using SQLMap and then finding passwords with a specific type of MD5 encryption which we could use hashcat to help us decrypt, so it was valuable to see how everything fits together.

Conclusions

While admittedly we didn’t find as many flags as most groups, we explored a vast number of paths to each flag. Each flag was like a spelunking cave dive into different aspects of the web’s content, and every new open path revealed a new chamber. Excluding those chambers that were just 404 pages, of course. In our group, we worked hard to collaborate and let each other indulge their hunches. Each member contributed to the flag finding (but Rob came in clutch with the carry), and finding teamwork through detective work was a blast. Perhaps we’d be better off in an escape instead of a CTF competition, though we shouldn’t concede our abilities just yet.

If we play another CTF game in the future, we would spend more time analyzing and researching compartments of website infrastructure like WordPress, MongoDB, and SQL. We all had other classes on top of the CTF game, so we couldn’t spend as much time as we would’ve liked, but if we had more time we feel confident that we could’ve captured more flags.

This CTF game gave us valuable web security and vulnerability experience. This transferable-to-the-workforce experience gave us both technical experience and practice working with a group. Overall, a very enjoyable week.