Bo Nappie

Computer and Network Security

Web Application Security Assessment

Section 1: *Overview & Summary of Findings:*

This report has been designed to emulate a detailed step by step walkthrough/analysis of this exploit from beginning to end. Two web servers within the IP range 192.168.******* were intentionally designed with several vulnerabilities acting as challenges for a group of students to attempt to exploit (with permission). Challenges were grouped in three stages; Stage one being relatively less difficult than those in Stages two and three. Every challenge is presented with a riddle; For every challenge completed, five points is awarded to that student. If students choose to use a hint, it will cost them two points.

In order to gain access to the server, it's necessary to first begin working with Linux. Once on terminal (for MacOS), it is necessary to run the command,

nmap 192.168*********

Nmap is a free and open source tool used for scanning, fingerprinting, and host discovery with the ability to create specialized network traffic. It is necessary to run this command every time one wishes to access these servers. After the scan is completed, I used the command,

ssh -D1080 cbonap***@********

which creates an SSH tunnel that allows me to connect to the target servers.

Mozilla Firefox is a suitable browser for exploiting vulnerabilities of web servers as it offers a multitude of add-ons, insights on code, and data about the server. The add-ons I used for this exploit included, *Cookie Quick Manager,* which allowed me to easily manipulate cookies.

Section 2: Findings and Methodology:

This section provides an in depth break down on how every challenge was completed, in addition to a detailed analysis on every unsuccessful attempt at certain challenges. This was my first Capture the Flag. I likely spent a lot more time than necessary to complete a few of these challenges but I learned so much throughout the process. Walking away to take a break when I felt frustrated and coming back with a fresh mindset often proved to be a success. I would spend hours getting frustrated without any progress, and then I would come back the next day finally completing the challenge.

Challenge One; *Hidden in Plain Sight:*

Upon arriving to the web page, http://192.168.****/login.php, I was presented with a seemingly standard looking login page. At first glance there appeared to be no hints "in plain sight". Thus, I naturally began to click on all accessible links I possibly could, feeling unsure about my ability to complete any challenges, as this was only challenge one.

As my clicking spree proved to be without success, I shamefully had the urge to copy and paste the contents of the login page onto Google, as I was beginning to feel pretty hopeless. After completely highlighting the page in defeat, the hidden flag was finally revealed, "in plain sight". Needless to say, I felt a rush of relief following this discovery, and just a smidgen of my hope was restored that I could possibly complete these challenges.

Sometimes vulnerabilities are really just that easy, I definitely was overthinking the rigor of this specific challenge.

Challenge Two; *Mr. Robot:*

This challenge also initially confused me. I eventually realized I should look at the Power points. Here I found a slide titled "robots.txt", with a description that seemed to suit the riddle. From here I altered the URL to be,

*http://192.168.****/robots.txt*

Which led me to the correct page where the flag was located.

The Robots.txt file could be exploited during the reconnaissance phase of an attack. It could allow search engines to crawl and index your website, putting sensitive data at risk of.

Challenge Three; *Read the Fine Manual:*

The riddle included a link which brought me back to the login page. I actually completed this challenge prior to completing challenge 2, which I initially couldn't figure out. Looking at the URL,

*http://192.168.****/login.php*

I decided to delete "login.php" so I would no longer be on the login page. This brought me to the index where various files and pages were located. "README.txt" was obviously a very enticing name for a file; after opening the file, it revealed the proper flag for this challenge.

This challenge relates to the vulnerabilities posed by the data that could be revealed in a manual; the takeaway from this challenge is to read the manual, and also go to the Github page, where a lot of the vulnerabilities numerically listed coincide with the challenge order and act as additional hints for every challenge.

Challenge Four; *Turn Over Every Stone:*

The riddle for this challenge hinted at the flag being hidden in the source code of one of the web pages that I shouldn't necessarily have access to; the second half of this riddle didn't resonate with me until a few days later. I had looked through every page of source code that I felt I could; I felt stumped and frustrated, but still unwilling to use a hint. Admittedly, I went crazy over this challenge. The next day I came back to it, I began searching through all of the files of code, so I thought at the time. Frustrated, I decided to use a hint, but I was still stumped. The third day I came back to the hint, I read it once more, and it suddenly clicked. I recalled one link, "config.php" that would take me to a webpage that never seemed to load and decided to check the source code of this page, not knowing if it would work, but there it was, the flag.

This was another challenge that I felt foolish for overcomplicating. When looking at source code, programmers may leave notes for themselves that could unintentionally result in a very easy way for a hacker to gain intel about the structure of the server.

Challenge Five: *Who are You?;*

As the riddle stated to read the fine manual, naturally, I went to the index of pages, and chose the "about.php" page. The flag was located at the bottom of the page. I believe that this challenge reiterates a similar takeaway from challenge three. There is significant potential for fine manuals to reveal an immense load of sensitive intel to hackers that they could use to exploit any vulnerabilities on their target, and the information is essentially handed to the hackers, as it's a well put together, detailed packet meant to be read by others.

Challenge Six: *Jump the Turnstile:*

This challenge pertained to gaining access to the application even if you don't have a password. I found this one easier to complete as I recalled us going over this type of vulnerability in class. I decided to enter, **usernameAnon***@edu into the username field, and in the password field I entered, " ' or 1=1—", which is equivalent to a true statement, permitting me to login, despite not having a legitimate password; I could enter that into the username field as well, and it would also allow me to login.

This challenge emphasizes the importance of diligent coding practices to avoid sequel injection hacking techniques such as this one.

Challenge Seven: *Logo:*

I did recall going over a method in class that would change the logo of the website, but I was having trouble achieving this. I attempted to change the logo on various input fields on the server, but I had no luck. I attempted to use HTML scripts that would change the logo. I looked at the source code so that I knew what body I should reopen, and I attempted to write code that resembled the source code but change the logo statement with a different address for the logo image.

After trying the methods I could think of, I decided to do a Google search and I found that I could install a "Man in the Middle" sort of tool on Linux to accomplish this challenge, but after rereading the rules I concluded this would likely not have been an allowed or the intended method for achieving this challenge.

Challenge Eight: *Elevate Privileges:*

This challenge was relating to changing the access you have as a user. Realizing this challenge pertained to cookie manipulation, I went to Firefox add-ons to install "Cookie Quick Manager". After logging in with any true username and password, I went to the cookie settings, and altered the cookie that had said "user", to read "admin", saved the cookie, refreshed my page, and found I had access to more privileges in addition to finding the flag for this challenge.

Cookie settings are in complete control of the user. Through this challenge and our class lectures, I feel that I now have a much stronger understanding of cookies, their capabilities, and their potential vulnerabilities. Roles should not be placed in cookies, as the user could change their role and that is dangerous.