**Capture the Flag: Boot-to-Root / Raven.ova**

Omeed Rabbani, Matt Parker, Jeffrey Maupin, Adam Miller  
UCI Cybersecurity Bootcamp, May 2020

“My primary goal of hacking was the intellectual curiosity, the seduction of adventure.”

- Robert California

**Introduction**

One very important thing you need to know to be a good cyber defender is how to be a good cyber attacker. In other words, if you are going to be successful in protecting computer systems and networks, you have to understand the techniques, tactics, and procedures (TTPs) used by hackers to exploit them. For our final project at the UCI Cybersecurity Boot Camp, we were provided with a virtual machine called Boot-to-Root or Raven, a [publicly available](https://www.vulnhub.com/entry/raven-1,256/#), intentionally vulnerable system. For this Capture the Flag (CTF) exercise, four “flags” were hidden in various locations on the machine. Each flag was represented by a title (“flag1”, “flag2”, etc.) and an arbitrary hash value. The goal - our goal - was to hack the box and find the flags.

**Executive Summary**

The first major discovery in our attack, after some reconnaissance and scanning, was that the server was running an old version of Wordpress (WP). We brute-forced the WP password of one of the users, which ended up also being the user’s SSH password. Once logged in with SSH, a grep command led us straight to two of the four flags. A few carefully crafted greps later, we found a MySQL username and password. In MySQL, a search for entries with the word ‘flag’ led us to the remaining flags, which were stored in the WP database table wp\_posts. Finally, we were able to gain root (a bonus win!) by taking advantage of a user’s privileges when running Python.

As we will discuss in more detail in our conclusion, basic security hygiene would have been enough to prevent our attack. One of the two wordpress users had a password we couldn’t guess after running our brute-force attack for a few days straight before giving up. If both users had such good passwords, we never would have been able to guess either. And if the user whose password we guessed had used a different password for SSH from what he used for WP, we would have been stuck at square one. Read on to learn more about our methods and the various ways we could have been thwarted by better security on the part of the victim.

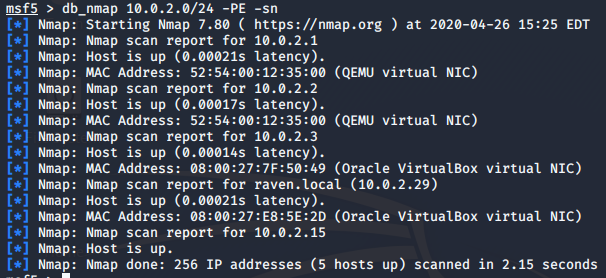
**Attack Narrative**

This assessment involved the attempted compromise of a given virtual machine, named 'Raven'. Each phase of the test is documented below.

**Reconnaissance**

**Initial Reconnaissance**

*The test began with a basic nmap host discovery scan. A single target (****10.0.2.29****) was discovered.*

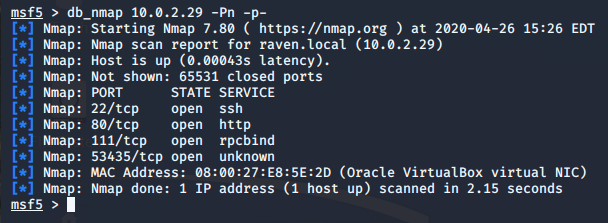
**

**Enumeration and Vulnerability Analysis**

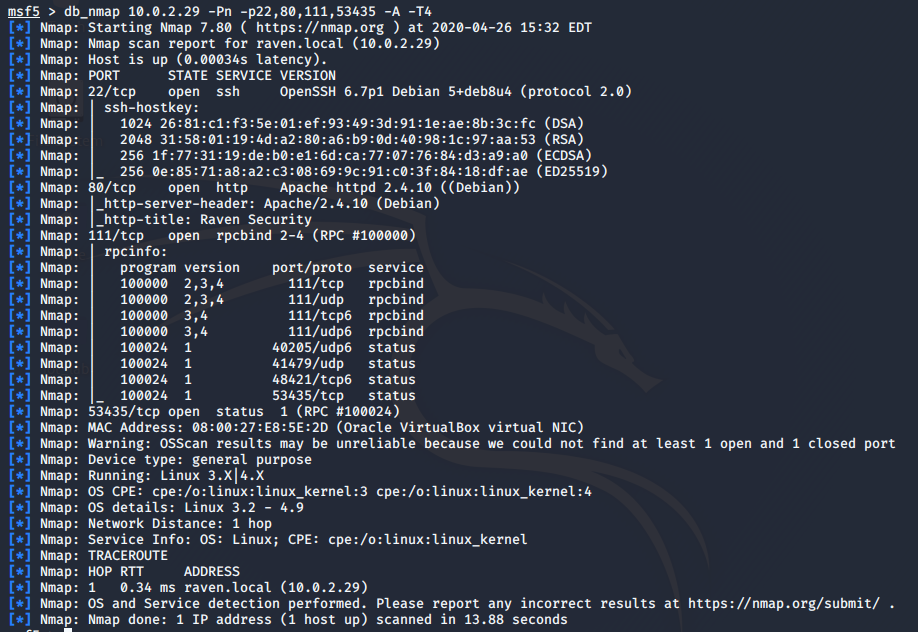
This section summarizes the most critical vulnerabilities affecting the target network.

**Service Discovery**

*The IP discovered during initial reconnaissance was used to perform a port scan against the target, which returned open ports* ***22****,* ***80****,* ***111*** *and* ***53435****.*

****

*A service scan was performed against the open ports. Of note are the services* ***SSH*** *and* ***HTTP.***

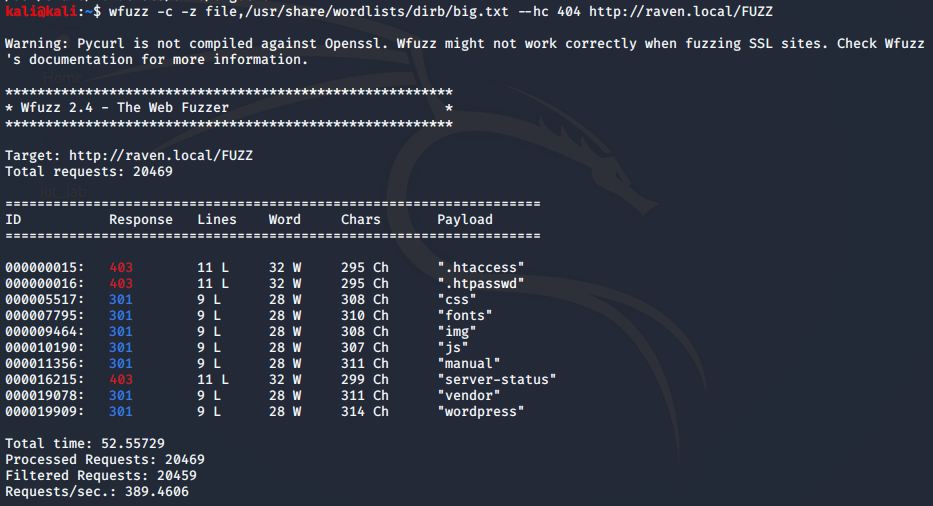
****

|  |  |  |  |
| --- | --- | --- | --- |
| **IP Address** | **Operating System** | **Vulnerabilities** | **Risk (Low/Med/High)** |
| 10.0.2.29 | Linux 3.2-4.9 | Exposed Web Server | Medium |
| 10.0.2.29 | Linux 3.2-4.9 | Exposed SSH Server | Medium |

**Web Server Analysis**

**Web Server Enumeration**

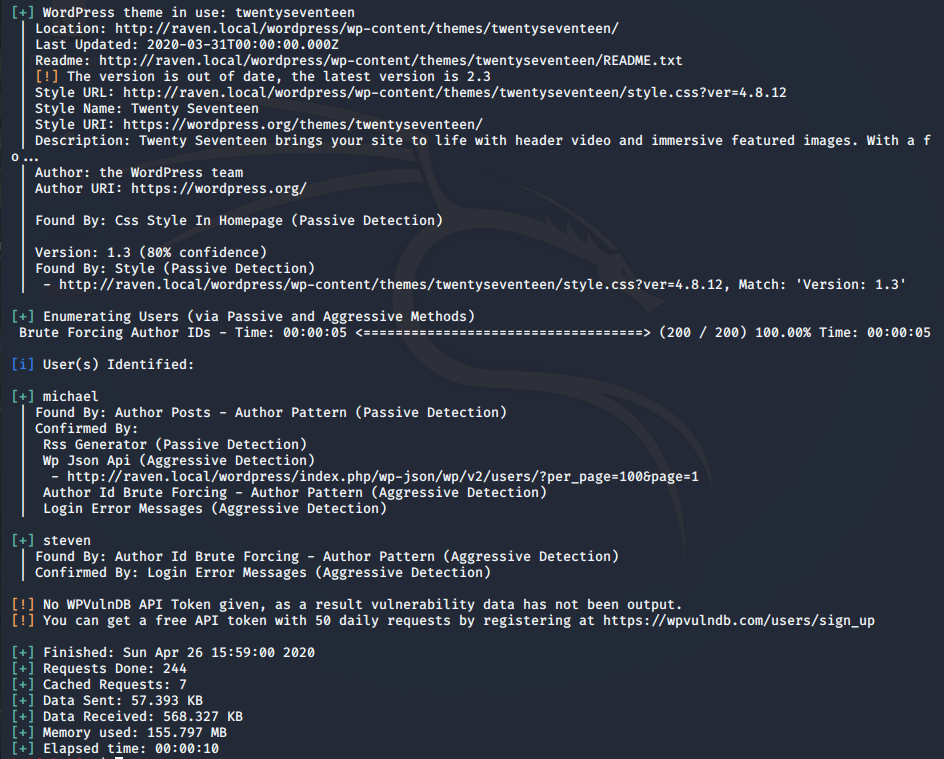
*WFuzz was used to discover web pages on the exposed web server. The dictionary used may be found natively on Kali Linux (/usr/share/wordlists/dirb/big.txt). The tool was configured to ignore 404 responses in order to return any page with any positive response. It may be seen that* ***WordPress*** *is installed on this server.*

****

**Wordpress Enumeration**

*WPScan was used to perform automated enumeration on the WordPress application. Two users were discovered:* ***michael*** *and* ***steven****.*

*Command:* ***wpscan -e -u -url raven.local/wordpress***

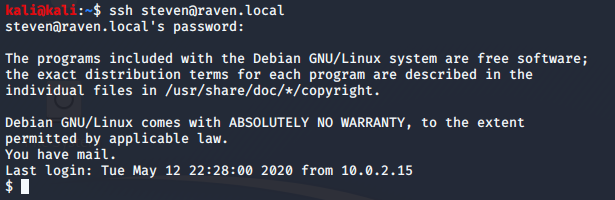
****

*WPScan was used again to perform a dictionary-based password attack against the discovered users. The dictionary used may be found natively in Kali Linux (/usr/share/ncrack/top50000.pwd). One set of valid credentials was discovered:* ***steven:pink84***

****

****

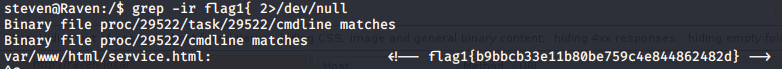
**Network Analysis**

*The discovered credentials were used to successfully log in to the server via ssh.*

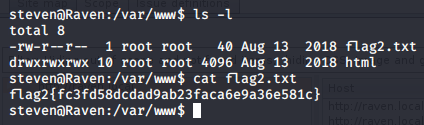
**Post-Exploitation Exploration and Privilege Escalation**

**Local Enumeration and Flag Discovery**

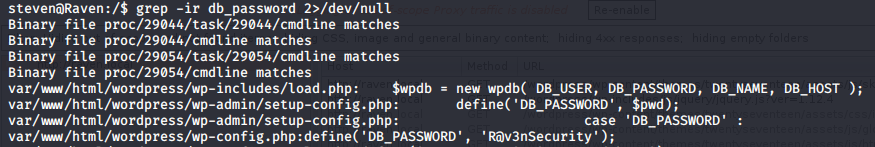
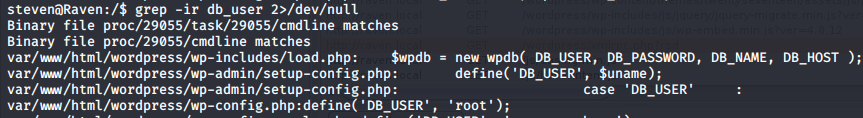
*The first flag was discovered using a simple grep command to find the string 'flag1' within the contents of a file. It was found in /var/www/html/service.html.*

***flag1{b9bbcb33e11b80be759c4e844862482d}***

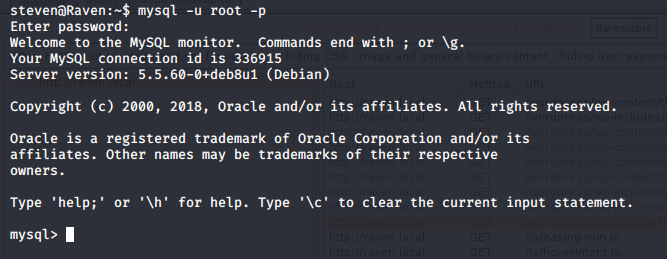
*The second flag was discovered sitting in a .txt file in /var/www/.*

***flag2{fc3fd58dcdad9ab23faca6e9a36e581c}***

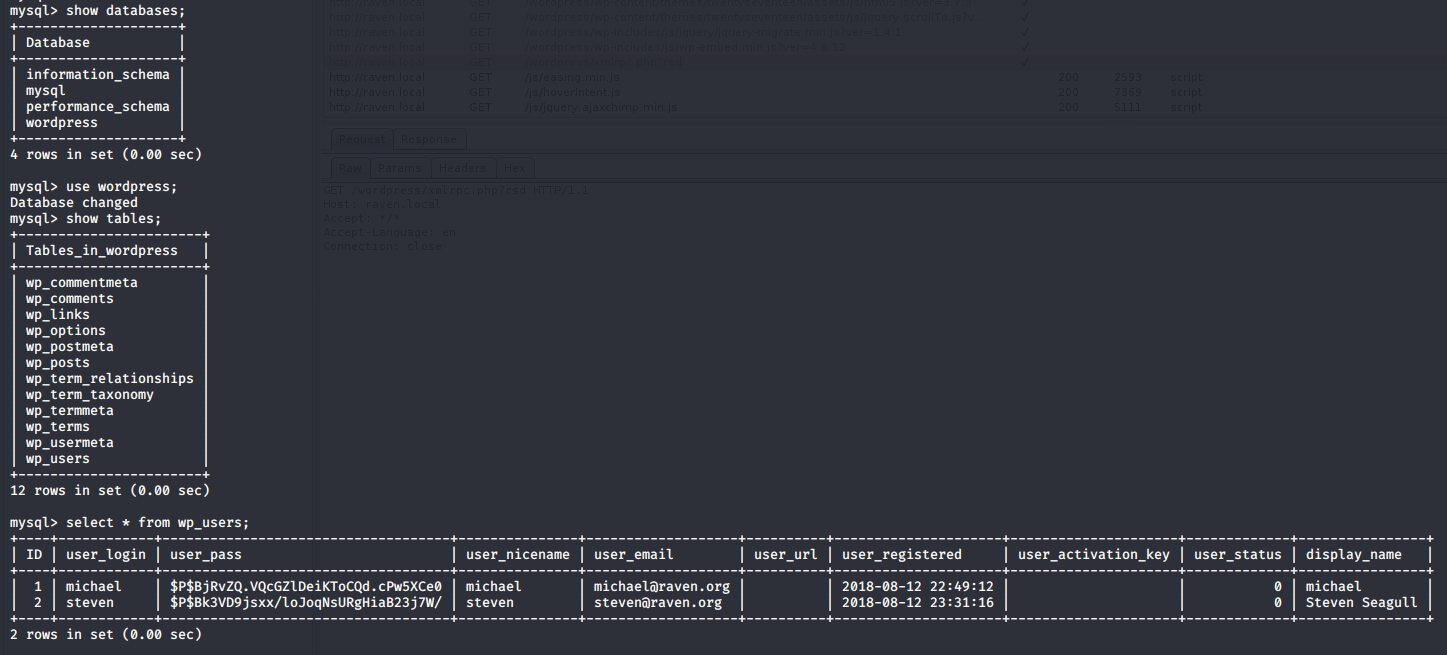
*Grep was used to look for potential credentials stored in plaintext. The WordPress database username (****root****) and password (****R@v3nSecurity****) were discovered in the file /var/www/html/wordpress/wp-config.*

**

These credentials were used to log into MySQL.



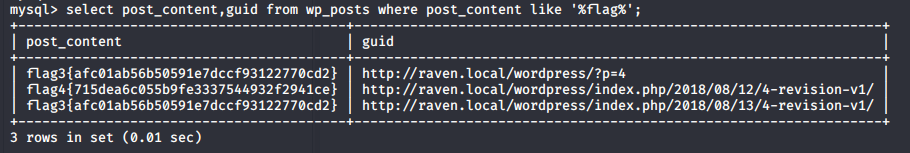
*Inside MySQL, basic enumeration reveals password hashes for the two discovered accounts.*

**

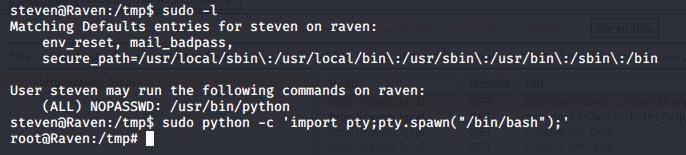
*Additionally, flags 3 and 4 may be seen within the database.*

***flag3{afc01ab56b50591e7dccf93122770cd2}***

***flag4{715dea6c055b9fe3337544932f2941ce}***



*The account used up to this point has been 'steven'. It is only a member of the group 'steven' and has no implicit administrator privileges. Its home directory is empty, and it does not appear to have access to any interesting SUID binaries. However, using the command 'sudo -l' to list available sudo commands reveals steven may run python as root without a password. From here, a simple python escape script provides a root shell.*

**

**Recommendations**

Being able to hack into the Raven box taught us some important lessons about what not to do when securing a computer system. Based on the results documented above, we would recommend the following steps to remediate the vulnerabilities identified on the target machine. These would be the types of actions a responsible administrator - or security analyst - would take in any situation.

**Web Server**

*Fuzzing the web server's pages revealed a hidden WordPress application. Assuming this application was not meant to be accessed publicly, it is advisable to move the service to a publicly inaccessible server.*

**Password**

*A WP administrator password was discovered using a common and relatively small dictionary. It is highly advisable to use good passwords, increasing minimum length as well as complexity. The same set of credentials discovered in WordPress were also valid for SSH. In addition to using a better password, users should not reuse credentials for multiple systems.*

**Hardening the Server**

*Allowing the steve account to run python as root is high risk and probably unnecessary; it allowed escalation from a fairly unprivileged user to root with minimal effort. It is advisable to practice the principle of least privilege and restrict such a permission, allowing specific, necessary scripts to be run at higher privileges rather than allowing python as a whole.*

**Conclusion**

The greatest lesson we can take away from this exercise is that - whether you do it for intellectual curiosity as Robert California does or you do it out of professional necessity - to be a great defender you need to become a great attacker first. Our team is excited to move on to hacking other intentionally vulnerable systems and, in doing so, learning more about how we can defend systems out in the real world.