**Summary:**

DC 1 is the initial machine in the DC series, which focuses on web vulnerabilities and often utilizes outdated versions of content management systems (CMS). The challenge with these machines is to exploit web vulnerabilities and find a method to gain control of the root user for each machine. DC 1 is a Linux machine that runs Drupal 7, which is susceptible to a remote code execution vulnerability.

To begin, no specific information was provided about the machine. Based on the virtual networking setup in my lab, I was aware of the IP address range in which the machine was located. I had Kali and DC1 running as virtual machines on an isolated NAT network, with Kali assigned the IP address 172.16.250.129. Using nmap ping scans, I discovered that DC1 was located at 172.16.250.131.

On Kali, I created a folder in the home directory for Vulnhub machines. Within that folder, I set up another specifically for DC1. This is where I copy or download exploits for review. Information from the target machine, such as credentials or configuration files, can also be saved here.

**Scanning & Enumeration:**

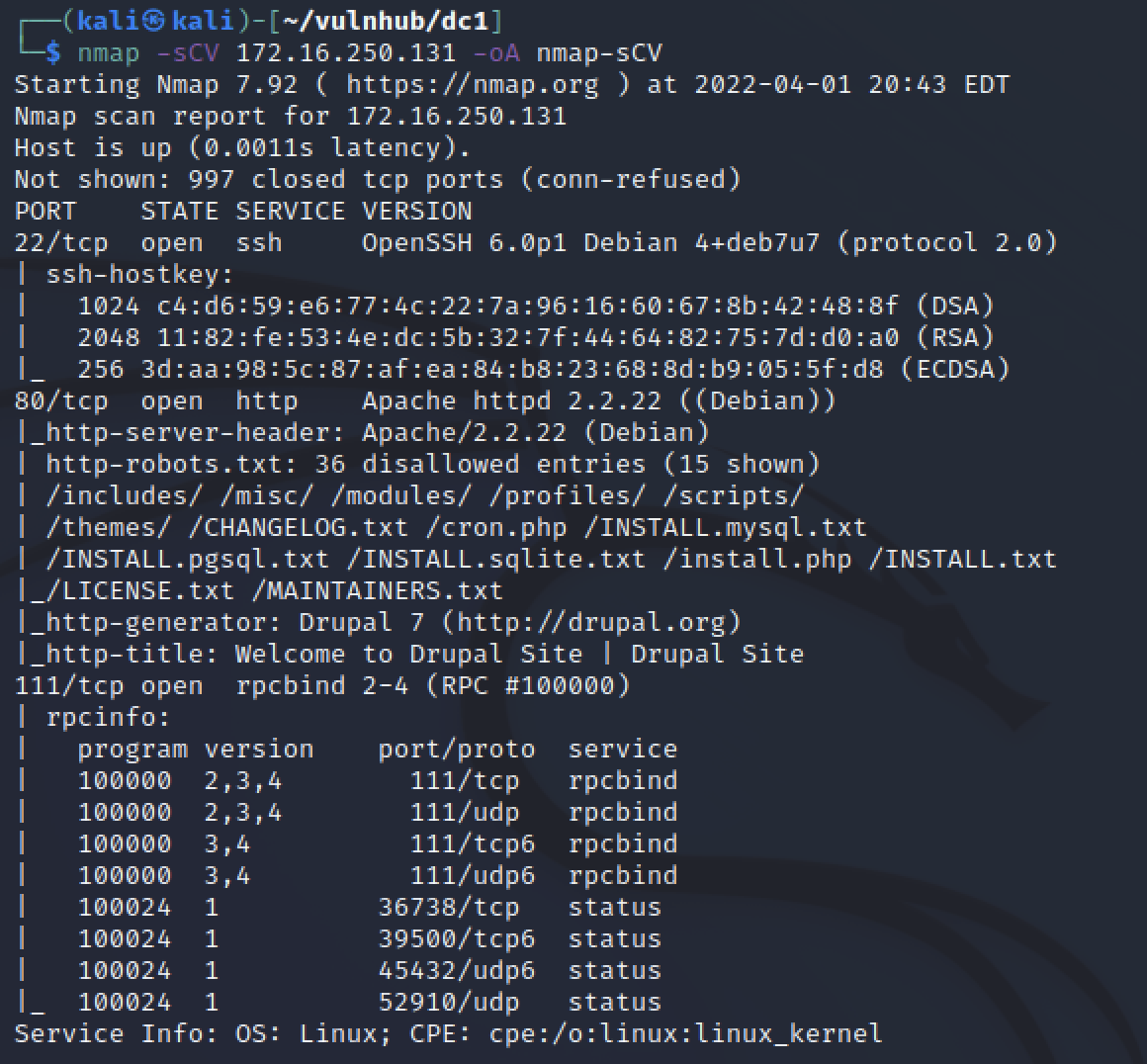
One of the steps in examining a new machine is to scan for open ports to determine the services that are operational. Nmap automates the process of port scanning and tries to identify the services by performing banner grabbing or checking for common details.

**Nmap**

To do a basic scan of DC1 I ran nmap -sCV 172.16.250.131 -oA nmap-sCV.

The options I passed with the command are listed below.

* sC and sV: run default scripts for found services and probe open ports to identiy services
* oA: create all major types of output (normal, XML, and grepable)
* host detection will happen first to check it’s online
* only common TCP ports will be scanned



There was only three ports found to be open. Port 22 for SSH, specifically OpenSSH 6.0.p1 on Debian Port 80 for HTTP or unencrypted web traffic. Specifically Apache 2.2.22 web server on Debian hosting a Durpal 7 site. The third port was 111 for RPC bind and I did not review that service. Just from the nmap results a lot of information was provided for this machine.

1. It was a linux machine running Debian
2. It had 3 known services running
   * SSH for access but without creds it is a hard target
   * HTTP for a website
   * RPC
3. HTTP scripts identified a lot of additional info
   * Confirmed web server and version from headers
   * robots.txt existed
   * Drupal 7 was probably the CMS in use

**Droopescan:**

Since Nmap revealed a Drupal website running on port 80, I utilized Droopescan. This tool is designed to extract specific information about Drupal instances, such as identifying their versions and listing the installed themes and plugins.

Nmap indicated that the version was 7, while Droopescan suggested it was most likely between 7.22 and 7.26. Obtaining specific version details can assist in narrowing down the available exploits later, increasing the likelihood of successful exploitation.

**Manual review**

Visiting the IP address through a standard browser like FireFox allows us to see the website as intended.

A screenshot of a computer

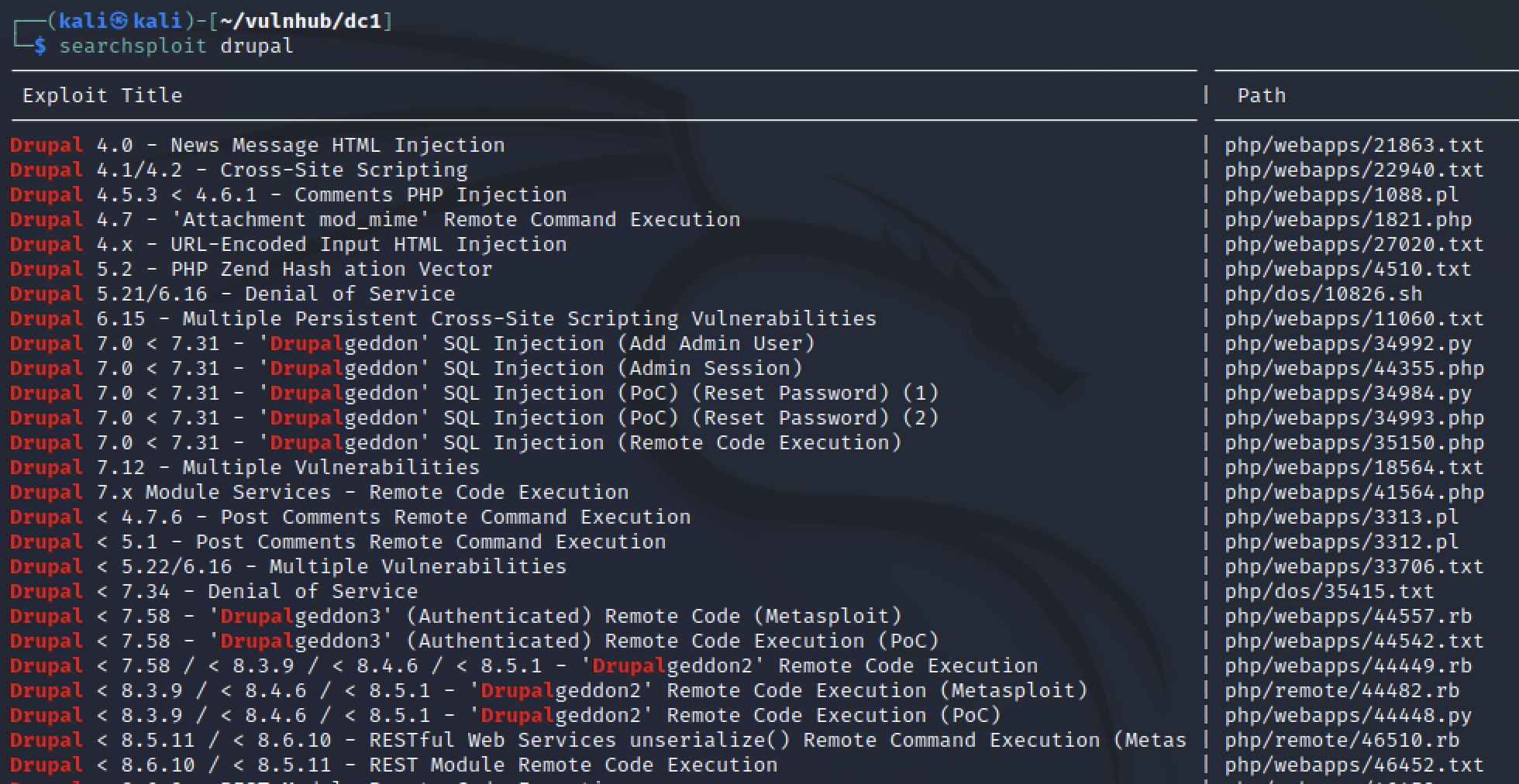
Description automatically generated

Before examining the site, I launched Burp Suite in the background to capture the requests. Other tools, such as Dirbuster, Wfuzz, or vulnerability assessment tools, can also run concurrently to further enumerate the website. The site displayed only default Drupal pages containing basic text information, with nothing that revealed details about its purpose, users, or developers.

Nmap indicated that a robots.txt file exists. Upon reviewing it and comparing it to the default file for Drupal 7, I found no custom information to glean. The admin login prompt is accessible, but we lack the necessary credentials. If the site had not been using the default template, we could have searched for identifying information to assist in brute-forcing some login attempts.

**Vulnerability Assessment**

Of the two services in nmap, only the drupal site was a clear target. A quick google search revealed that Drupal’s latest version is 9.x so it is likely there is public exploits for Drupal 7.22-26 Kali has searchsploit which reviews a local version of exploit-db for possible exploits to try.



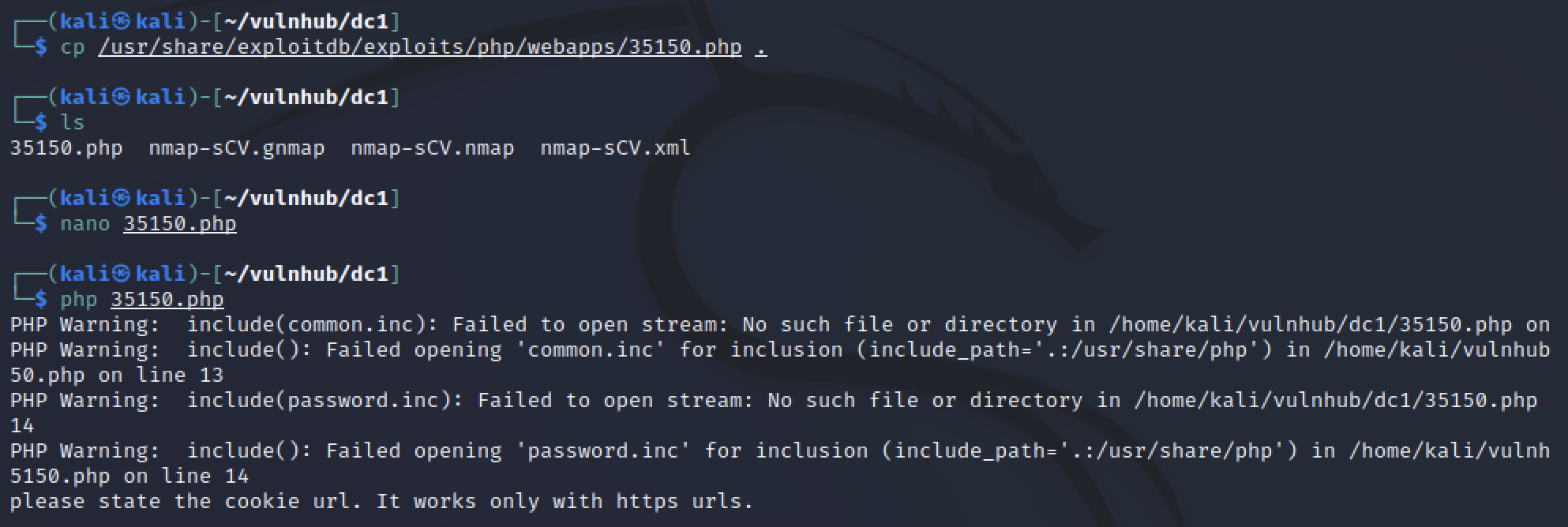
searchsploit drupal gave a lot of exploits to review! I could have narrowed down my search options by specifying version 7.22-26. However “Drupalgeddon” remote code execution (RCE) stood out to me immediately from the initial results.

There was also a Metasploit option which meant I could automatically exploit the vulnerability after just setting a few options.

**Exploitation:**

I tried a manual exploit first. I copied the RCE exploit into my local directory and reviewed it using nano. The exploit was written in PHP and required some files that I didn’t have but maybe it could run without them.

php 35150.php showed some errors that proved this exploit was not viable. The warnings of missing files was expected from reviewing the code. However “Please state the cookie” and “it works only with https urls” are bad messages. The target was running HTTP unencrypted and I did not have cookie values to provide. So this exploit was not possible.

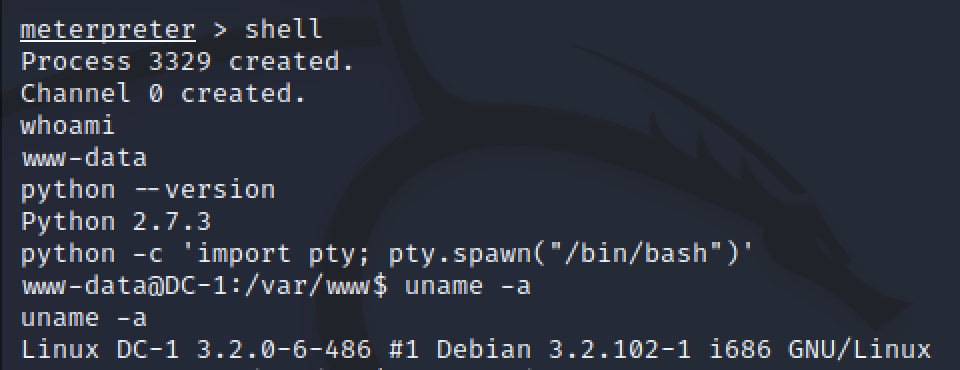


While there were other exploits to attempt, I took this as a sign to spin up Metasploit and get an easy win. Running msfconsole and searching for drupalgeddon showed me multiple exploits available. I used the one most relevant to Drupal 7.22-26. The only options required to set were the RHOSTS (target IP address) and LPORT (local port to listen for a reverse shell on).

A screenshot of a computer program

Description automatically generated

By default, this exploit used a meterpreter payload which ensured some safe handling of the reverse shell. Using shell gave a command prompt on the target machine. After some quick enumeration to check what user the shell connected as and what is on the machine, I improved the shell by adding TTY.



www-data user is restricted and does not have permission to access many parts of the linux system. The next thing to do was find a way to get unrestricted access to the machine.

**Privilege Escalation:**

There’s many things to check when searching for ways to get higher priveleges. G0tmi1k has a long standing blog post on [Linux Privelege Escalation](https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/) containing things to check on a linux host and S1ren references these in her streams.

Some of the things I checked were

* cat /etc/passwd users on the machine
* ls -la /etc/\* odd services or permissions (files writeable by everyone or not owned by root)
* ls -la /home user’s data, one existed called flag4
* ps aux | grep root root services
* netstat -antup local ports running services only for localhost

Automation can really improve the time spent enumerating for possible privelege escalation paths. It is smart to review each command first and understand the possible results, then build scripts to speed up the process. Some scripts are already public but without knowing what to look for the results will be hard to parse.

To be honest, this point was where I got stuck, not seeing access to users or any services I knew how to exploit. This is when S1ren reviewed SUID and GUID escalations. SUIDs is where calling a command as a low level user changes the active permissions to the root users. Some commands allow options that can be used to execute other commands like spawning a new shell session. The command to search for SUIDs is find / -perm -u=s -type f 2>/dev/null

* find: the command we’re using to search
* /: search in the base directory
* -perm -u=s: permissions for SUID
* -type f: search for files only
* 2>/dev/null: redirect errors to “null” (don’t display errors)

Running find we see there are 18 programs with the permission we want to review. For new hackers, we can use [GTFObins](https://gtfobins.github.io/#+suid) to learn about each of the programs that might spawn a new root shell. With experience this will be a faster process.

A computer screen shot of a computer

Description automatically generated

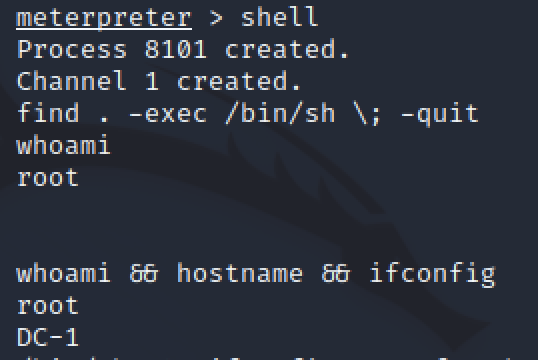
GTFObins shows that [find](https://gtfobins.github.io/gtfobins/find/#suid) can also be used to spawn a new root shell. Pasting from the website into our www-data shell I got multiple errors. Passing the -p parameter also made me close the www-data shell.

A screen shot of a computer screen

Description automatically generated

Meterpreter did not lose connection so it was possible to reconnect quickly. After removing the -p from the find command, I had a working shell as root.

**Root**



**Target Information**

OS: Debian Linux Services:

* OpenSSH on Port 22
* Apache 2.2.22 on Port 80 hosting a Drupal 7 site

Vulnerability: Drupalgeddon - Remote code Execution Privilege Escalation: find SUID

**Tools :**

Droopescan is a tool that I’ve not very experienced with. I find it helpful to quickly identify Drupal information.

SUIDs were a privilege escalation path that I did not fully understand. Having S1ren explain the concept and being able to play with the command helped me grasp it better. Future enumeration for privilege escalation will be easier since I know another option.

**Another way**

**I used this command “ifconfig” to know my ip address which is:192.168.70.129**

**A computer screen shot of a program

Description automatically generated**

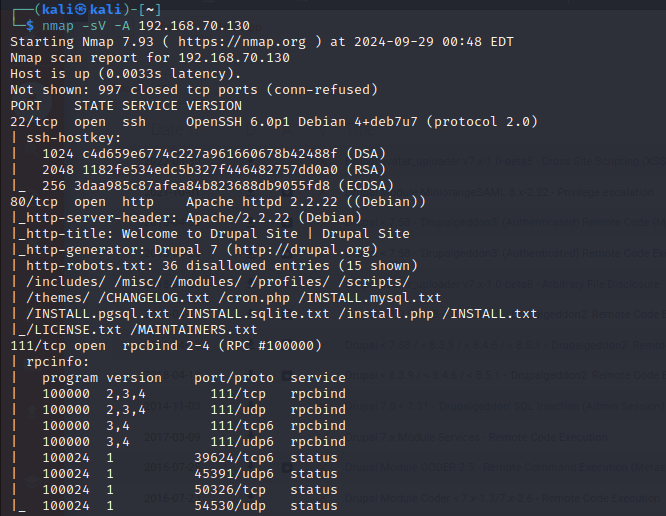
**This command for scanning the network and discover the target “Nmap ip address range”, and we found the target ip address which is :192.168.70.0/24**

**A screenshot of a computer

Description automatically generated**

**My target is 192.168.70.129**

Now we will scan the services running on the target ip that is obtained from previous step :**nmap -sV -A 192.168.70.130**



You will see there a lot of open ports and services such as **http 80**

**Go browser and write 192.168.70.130**

**A screenshot of a computer

Description automatically generated**

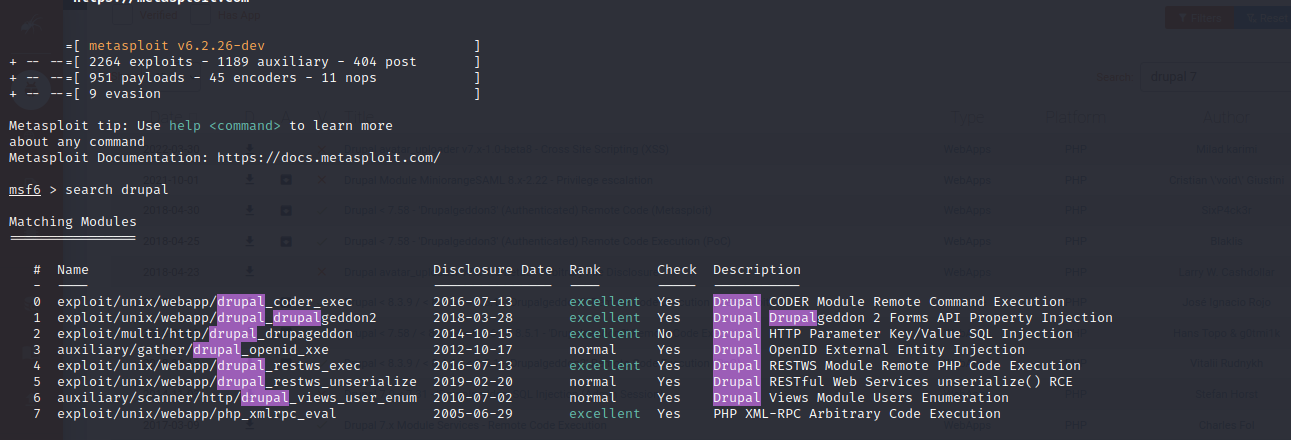
**I found fields that I can write into, so I can use sql injection**

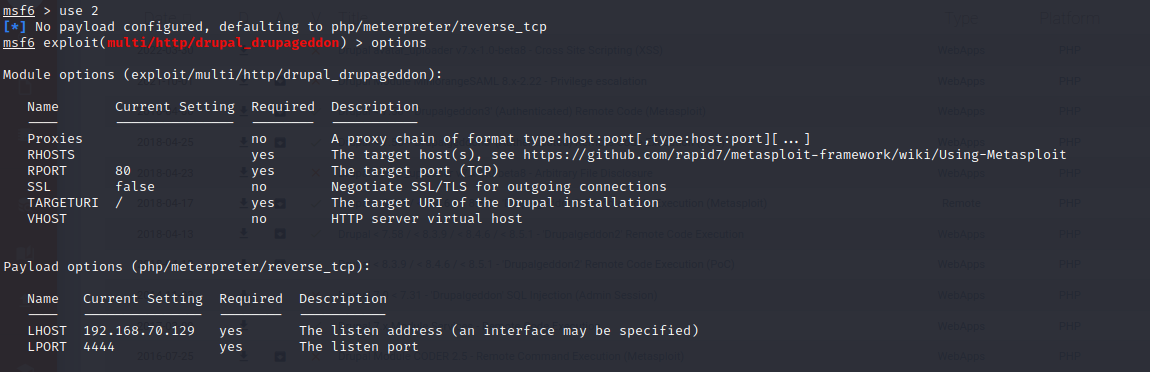
**Go to kali linux and use Metasploit framework**

**A screen shot of a computer

Description automatically generated**

**Search Drupal exploits**

****

**Use exploit/multi/http/drupal\_drupageddon**

**Give options set rhosts 192.168.70.130**

**A screen shot of a computer

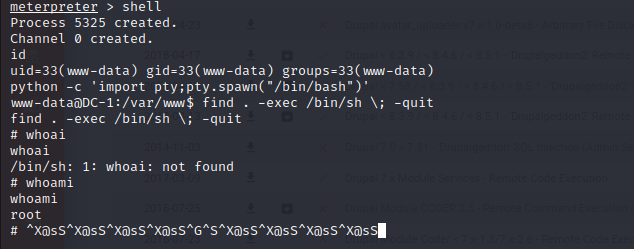
Description automatically generated**

**Now session 1 opened**

I write **python -c 'import pty; pty. spawn("/bin/bash")'** give me name of machine

I write **find . -exec /bin/sh \; -quit** up me to root

I now becomed root

****

**Now this machine attacked and with me permission of root**

**To close this bug, must close http protocol and use https**

**Findings**

* Service Enumeration: Discovered multiple services including HTTP on port 80 and SSH on port 22. The web service was outdated, leading to potential vulnerabilities.
* Weak Authentication Mechanisms: A weak username/password combination allowed access to a critical service.
* Privilege Escalation Vulnerability: A misconfigured sudo privilege allowed root access through an exploit.
* Sensitive Information Exposure: Important system files containing credentials were discovered post-exploitation.
* No Proper Logging: The system lacked logging mechanisms, making it difficult to detect unauthorized access.

**Recommendations**

1. Update Software and Services: All outdated software versions should be updated to mitigate known vulnerabilities.
2. Enforce Strong Authentication: Implement stronger password policies to avoid weak credentials being exploited.
3. Restrict Privileged Access: Restrict sudo permissions and implement the principle of least privilege to minimize potential privilege escalation vectors.
4. Enable System Logging: Configure proper logging and monitoring to detect unauthorized access or suspicious activity.
5. Patch Known Vulnerabilities: Regularly review vulnerability reports and apply patches to prevent exploitation of known issues.

DC-1

**Executive Summary:**

The engagement involved identifying the target system, scanning the network, discovering vulnerabilities, and exploiting them to gain administrative and root access. The key findings include multiple weaknesses in service configuration, such as the exposure of RPC services, SQL injection vulnerability in Drupal 7, and improper privilege management allowing SUID misconfigurations to escalate privileges. Recommendations include patching the Drupal CMS, enforcing stricter access control, and securing vulnerable services like RPC.

**Methodology:**

1. **Reconnaissance and Enumeration:**
   * **Tools:** ifconfig, nmap, OSINT techniques, Wappalyzer
   * **Actions:** Gathered information on the network and services, identified open ports and running services like HTTP, SSH, and RPC.
2. **Scanning and Vulnerability Assessment:**
   * **Tools:** Nmap was used to scan the network and identify running services and their versions. Drupal vulnerabilities were identified using OSINT tools like Wappalyzer.
3. **Exploitation and Gaining Access:**
   * **Techniques:** SQL injection was used to add an administrative user to the Drupal system. A Drupal module was exploited to gain shell access, and further steps led to the discovery of sensitive data.
4. **Maintaining Access and Privilege Escalation:**
   * **Techniques:** Reverse shell connections and privilege escalation via SUID misconfigurations. Persistence was maintained using root privileges obtained through a SUID exploit.
5. **Post-Exploitation and Lateral Movement:**
   * **Actions:** Explored the compromised system, collected flags, and confirmed administrative privileges. The database and sensitive configuration files were accessed.

A screenshot of a computer

Description automatically generated1) use “ifconfig ” to identify the ip of my machine to conclude our network ip which will be 🡪 192.168.74.0/24.

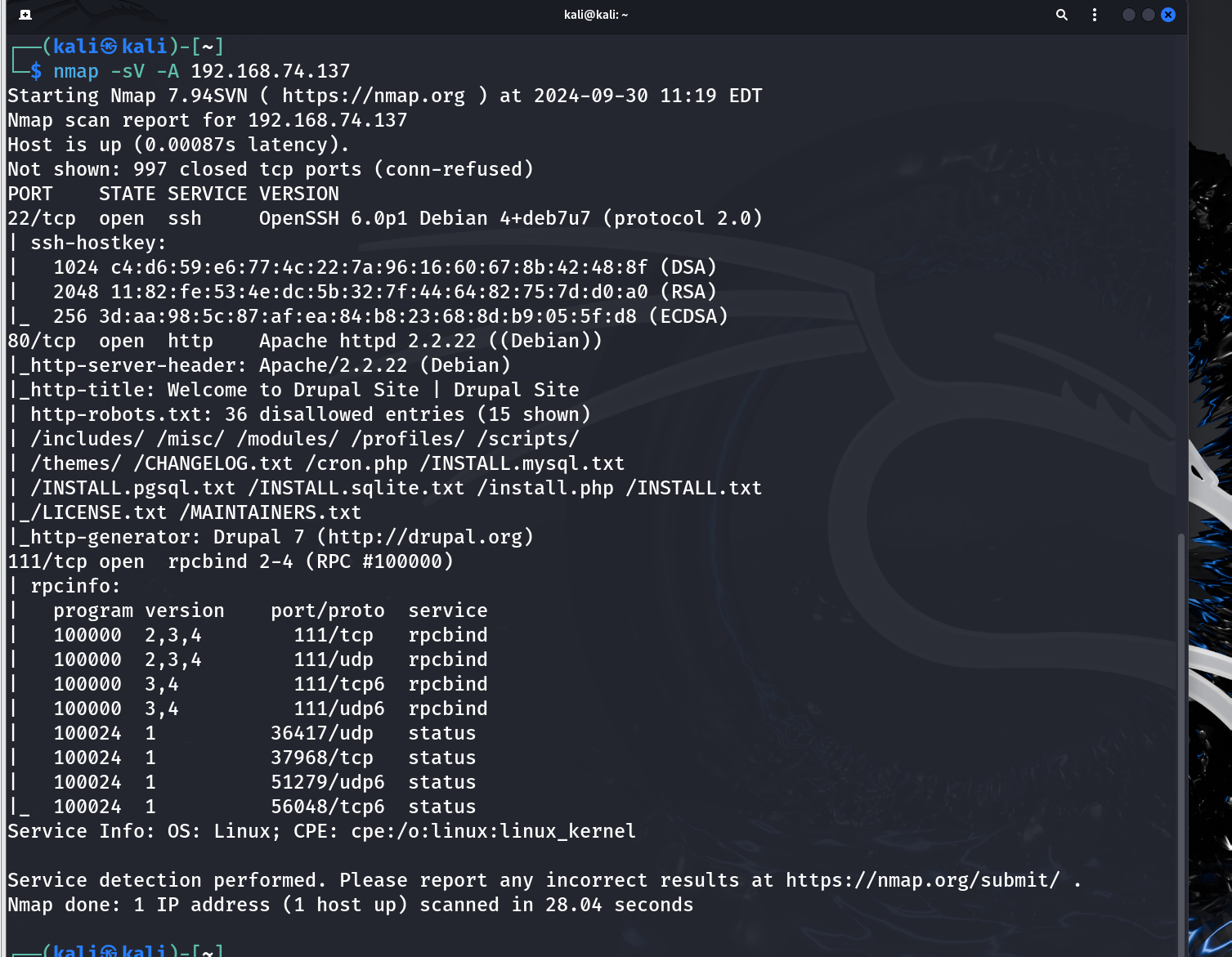
2) scan our network that connected by NAT

A computer screen shot of a program

Description automatically generated

3) **nmap**: This is the network scanning tool

1. **-sV:**
   * This option enables version detection. It attempts to determine the version of the services running on open ports.
2. **-A:**
   * This option enables several advanced and aggressive scan options:
     1. **OS Detection**: Attempts to determine the operating system of the target.
     2. **Version Detection**: As mentioned, it identifies the versions of services running.
     3. **Script Scanning**: Runs a set of Nmap scripts against the target to gather more information.
     4. **Traceroute**: Traces the path packets take to reach the target.



4) We see open port ssh, HTTP, and rpcbind. We come across an unusual protocol here - RPC. Briefly, this protocol processes requests for a given computer service, where that service is located on another computer on the same network.

HTTP PAGE

We see the main page with the login form:

A screenshot of a computer

Description automatically generated

5) The robots.txt file is a text file located in the root directory of a website. It provides instructions to web crawlers (robots) about which parts of the website should or should not be crawled and indexed.

A screenshot of a computer

Description automatically generated

6) **EXPLOIT - CREATING A NEW USER WITH SQL INJECTION**

Using the “Wappalyzer” Chrome plugin, I was able to find out which version of Drupal the website is running on. This is version 7.

A screenshot of a computer screen

Description automatically generatedCheck if there are any exploits in our local database.

7)We see many of them. The most convenient seems to be adding an account with administrator rights. username: SaraAdmin , password:sara .

A screenshot of a computer

Description automatically generated

Being logged in, we can immediately see that we have administrative rights.

Being logged in, we can immediately see that we have administrative rights A screenshot of a login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

While browsing the panel for additional information, we come across an interesting hint (?) Regarding the flag.

The note refers to the files / etc / passwd and / etc / shadow. The –exec flag most likely means access to a Linux shell.

8) Is there any way to get to the shell via the website with administrative rights?

A screenshot of a computer

Description automatically generatedApparently, there is a Drupal module that will give us this option. I install it and turn it on. After these steps, I have a shell on the home page.

A screenshot of a computer

Description automatically generated

A black screen with green text

Description automatically generatedA screenshot of a computer program

Description automatically generatedYou will immediately notice the file ‘flag.1.txt’. Browsing through its content, we see another hint: “Every good CMS needs a config file - and so do you”

9) So we google “Drupal config file directory”. We can see that it is located in the folder /sites/default / '' In it we have the configuration filesettings.php```. Already at the very beginning of the file, we see flag 2.

A screenshot of a computer screen

Description automatically generated

A screen shot of a computer

Description automatically generatedAs the note says, at the bottom, there are credentials for the database account. So let’s log in.

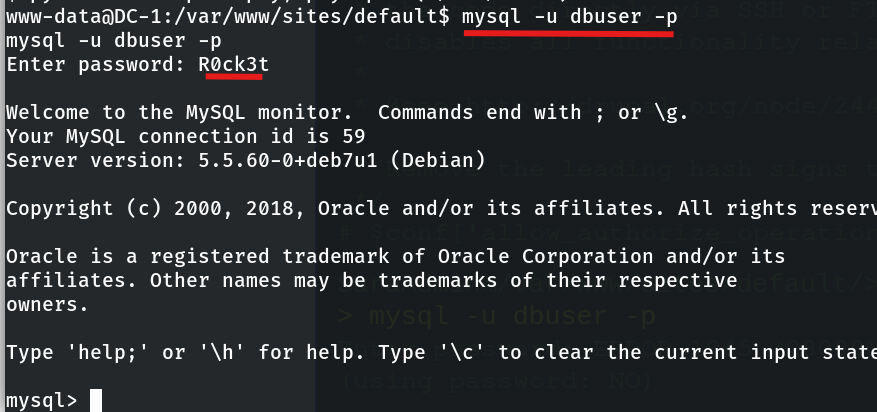
This command does not work. We get an access denied error, but it may be more because we are working in a web shell all the time.so, we will try the command again on reverse shell.

10) **Reverse Shell:**

A screenshot of a computer program

Description automatically generatedLet’s connect to our machine with the reverse shell

It will not be an aesthetic shell so let's make it even better with this command:

So let’s log in to the database again :

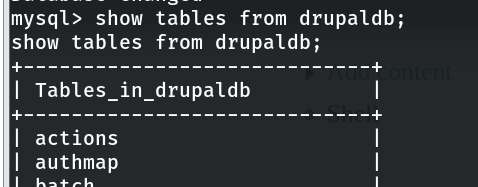
A screenshot of a computer

Description automatically generatedA screen shot of a computer code

Description automatically generated11)We see information\_schema which has basic information about the database and its structure. However, we are interested in the drupaldb base.

A screen shot of a computer program

Description automatically generated



12) ESCALATION OF POWERS:

A screenshot of a computer

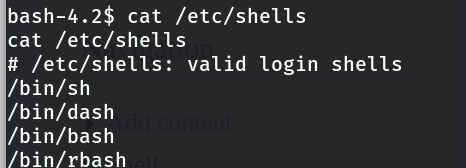
Description automatically generatedOne of the basic techniques for escalating privileges is through SUID programs (Those that need different privileges than those of the user executing the command) . To display them, execute the command:

Many of them are normal commands. For example, / usr / bin / passwd must have administrator privileges because it needs access to / etc / passwd and / etc / shadow. However, there is a command find which has unnecessary SUID status The find command has a built-in flag -exec (Recall the content of the flag3)

A black screen with white text

Description automatically generatedWhat we can do now is to execute the command '' find / etc / shadow -exec bash \;. In doing so, however, we are still the www-data user.

This is quite strange. Let’s try to do it with a different shell. To see what else we have to choose from, run cat / etc / shells



Let’s try with dash ``````find / etc / shadow -exec dash \;

A black background with white text

Description automatically generated

Apparently, it works :) Going to the / root directory, we see the file**thefinalflag.txt**

**Categorization of Steps:**

**Reconnaissance & Enumeration:**

* **Commands/Tools:** ifconfig, nmap -sV -A, Wappalyzer
* **Actions:** Gathered IP information and details about open services and versions.

**Scanning & Vulnerability Assessment:**

* **Tools:** Nmap, Drupal version enumeration (Wappalyzer)
* **Findings:** Open ports, service versions, and vulnerability to SQL injection in Drupal 7.

**Exploitation & Gaining Access:**

* **Tools/Frameworks:** SQL injection, custom exploits, Drupal module for shell access.
* **Actions:** Created an admin user, logged into the Drupal admin panel, installed a module to gain shell access.

**Maintaining Access & Privilege Escalation:**

* **Techniques:** Reverse shell, SUID misconfiguration exploitation (find /etc/shadow -exec).
* **Result:** Root access was obtained through the find command with elevated privileges.

**Post-Exploitation & Lateral Movement:**

* **Actions:** Explored the database and file system, located flags, confirmed root access, and completed the engagement by securing the final flag.

**Findings & Recommendations:**

* **Vulnerability 1: SQL Injection in Drupal 7**
  + **Impact:** Allows unauthorized creation of administrative accounts.
  + **Recommendation:** Upgrade to a more secure version of Drupal and implement stronger input validation.
* **Vulnerability 2: RPC Service Misconfiguration**
  + **Impact:** Exposes the system to unauthorized access via remote procedure calls.
  + **Recommendation:** Restrict access to RPC services and monitor network traffic for malicious activity.
* **Vulnerability 3: SUID Misconfiguration**
  + **Impact:** Allows privilege escalation via the find command.
  + **Recommendation:** Audit SUID programs regularly and remove unnecessary SUID flags.
* **Overall Risk:** The vulnerabilities found allow an attacker to escalate privileges and gain root access, posing a severe risk to the system's integrity. Immediate patching and access control hardening are recommended.