Capitolis
Report: Offensive Security

1. Project Overview

We wanted to define what a user could see outside the organization. There are plenty of unsavory characters on the internet who like to see what's out there to hack just for fun. So, we thought, why not pretend I'm one of them and do some initial reconnaissance and try brute force on some passwords. This was an external test where I wanted to see if perhaps a user was lazy with credentials and I could use a dictionary attack to gain access to the login page and into the proper system. Overall, we wanted to understand how the site works and its potential vulnerabilities to the outside world. I focused on prod.capitolis and capitolis.com

2. Summary

The goal of this test is to identify and exploit vulnerabilities in the system in order to assess the system's security posture. Penetration tests are an important part of a comprehensive security strategy and can help organizations identify and fix vulnerabilities before they are exploited by attackers. I wanted to be able to determine the value of the compromised systems—i.e., how much financial impact would their incursion cost?

3. Methodology

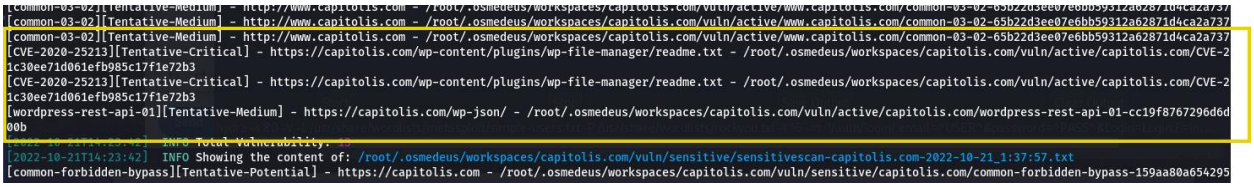
The scanning and discovery phase was used to discover how Capitolis's system would respond to various attempts at intrusion. I used automated penetration test tools to scan for initial vulnerabilities as well as performing a static and dynamic analysis. Static analysis inspects an application's code in an attempt to predict how it will react to an incursion. Dynamic analysis looks at an application's code as it runs, providing a real-time view of how it performs. I also hoped to gain more information about network systems, servers, and devices, as well as network hosts. An examination of the administrative interface revealed that it was vulnerable to a XSS scripting attack, which could be exploited to obtain interactive access to the website.

4. Findings & Risk Analysis

Key Findings:

I discovered various flaws that may be exploited by a malicious actor. The most important finding was:

- *CVE-2020-25213*
    - The File Manager (wp-file-manager) plugin before 6.9 for WordPress allows remote attackers to upload and execute arbitrary PHP code because it renames an unsafe example elFinder connector file to have the .php extension. This, for example, allows attackers to run the elFinder upload (or mkfile and put) command to write PHP code into the wp-content/plugins/wp-file-manager/lib/files/ directory. This was exploited in the wild in August and September 2020.

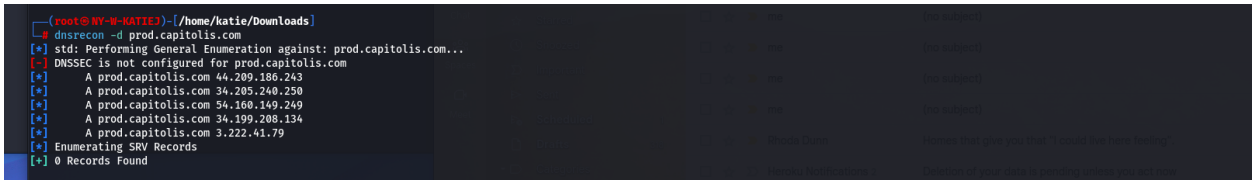


Website: prod.capitolis.com

1. DNS Recon

DNSRecon is a Python script that provides the ability to perform: Check all NS Records for Zone Transfers. Enumerate General DNS Records for a given Domain (MX, SOA, NS, A, AAAA, SPF and TXT). Perform common SRV Record Enumeration.

For example, an attacker may leverage such data to generate enough conditions for a denial-of-service (DoS) campaign. In DNSRecon, we proceed as follows:



DNS Recon at work

2. Gobuster

One of the primary steps in attacking an internet application is enumerating hidden directories and files. Doing so can often yield valuable information that makes it easier to execute a particular attack, leaving less room for errors and wasted time. There are many tools available to try to do this, but not all of them are created equally.

```
┌──(root㉿NY-W-KATIEJ)-[/home/katie/Downloads]
└─# gobuster dns -d prod.capitolis.com -t 50 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Domain:     prod.capitolis.com
[+] Threads:    50
[+] Timeout:    1s
[+] Wordlist:   /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
2022/10/21 10:57:38 Starting gobuster in DNS enumeration mode
===============================================================

===============================================================
2022/10/21 10:59:56 Finished
===============================================================
```

GoBuster at work

3. Nikto

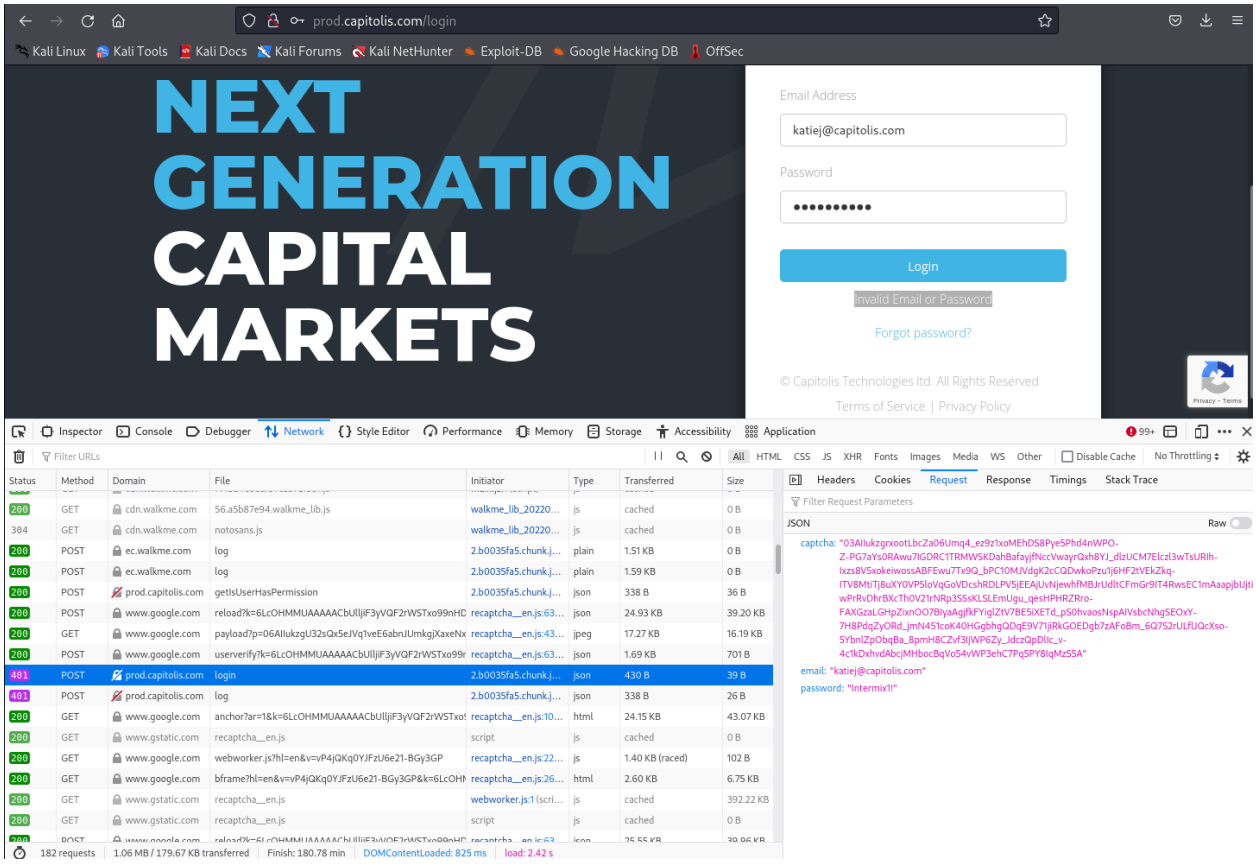
Nikto is an Open Source software written in Perl language that is used to scan a web-server for the vulnerability that can be exploited and can compromise the server. It can also check for outdated version details of 1200 servers and can detect problems with specific version details of over 200 servers.

```
┌──(root㉿NY-W-KATIEJ)-[/home/katie]
└─# nikto -h prod.capitolis.com
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          34.205.240.250
+ Target Hostname:    prod.capitolis.com
+ Target Port:        80
+ Message:           Multiple IP addresses found: 34.205.240.250, 54.160.149.249, 3.222.41.79, 44.209.186.243, 34.199.208.134
+ Start Time:        2022-10-21 11:38:37 (GMT-4)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ Retrieved x-powered-by header: Express
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server banner has changed from '' to 'awselb/2.0' which may suggest a WAF, load balancer or proxy is in place
+ /site.pem: Potentially interesting archive/cert file found.
+ /backup.cer: Potentially interesting archive/cert file found.
+ /prod.capitolis.alz: Potentially interesting archive/cert file found.
+ /prod.tar.lzma: Potentially interesting archive/cert file found.
+ /capitolis.tar: Potentially interesting archive/cert file found.
+ /prod_capitolis_com.jks: Potentially interesting archive/cert file found.
+ /prodcapitolis.tar: Potentially interesting archive/cert file found.
+ /backup.egg: Potentially interesting archive/cert file found.
+ /34.205.240.250.tgz: Potentially interesting archive/cert file found.
+ /site.egg: Potentially interesting archive/cert file found.
+ /prod.capitolis.com.jks: Potentially interesting archive/cert file found.
+ /prod_capitolis_com.egg: Potentially interesting archive/cert file found.
+ /34.205.240.250.egg: Potentially interesting archive/cert file found.
+ /capitolis.tar.bz2: Potentially interesting archive/cert file found.
+ /prodcapitolis.tar.lzma: Potentially interesting archive/cert file found.
+ /prod.egg: Potentially interesting archive/cert file found.
+ /prodcapitolis.war: Potentially interesting archive/cert file found.
+ /com.egg: Potentially interesting archive/cert file found.
+ /prodcapitoliscom.tgz: Potentially interesting archive/cert file found.
+ /prodcapitoliscom.tar.lzma: Potentially interesting archive/cert file found.
+ Allowed HTTP Methods: GET, HEAD
+ OSVDB-1201: /cgi/cgiproc?: It may be possible to crash Nortel Contivity VxWorks by requesting '/cgi/cgiproc?$' (not attempted!). Upgrade to version 2.60 or later.
+ /isapi/count.pl?: AN HTTPd default script may allow writing over arbitrary files with a new content of '1', which could allow a trivial DoS. Append /../../../../../ctr.dll to replace this file's
ts, for example.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3092: /js: This might be interesting...
+ 7869 requests: 0 error(s) and 118 item(s) reported on remote host
+ End Time:           2022-10-21 11:43:22 (GMT-4) (285 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Nikto at work

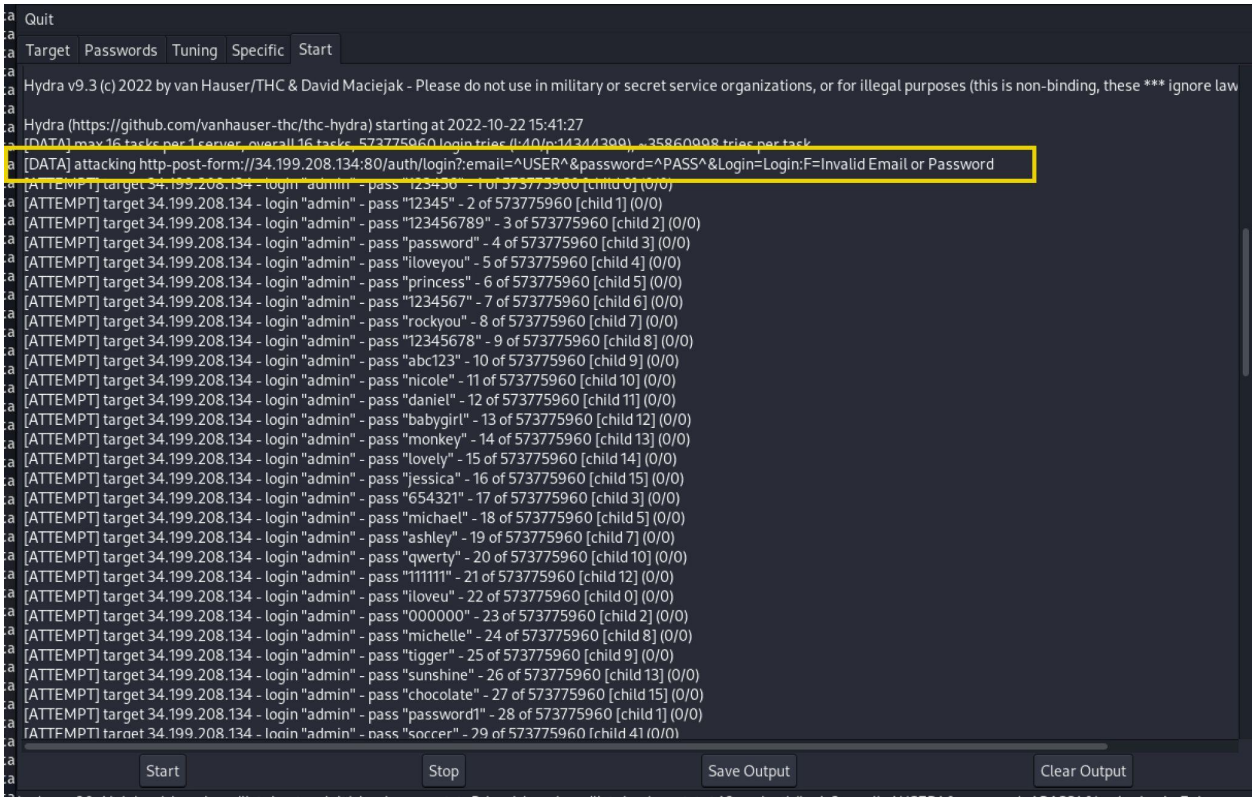
4. Hydra

- Capturing the post request gives me an idea of the parameters used and the specific url to plug into the command.



Capturing post request

- This is the brute force of the login page using a supplied word list and password list. I later changed the user to a cap user's email to see if perhaps I'd crack the password. I used the rockyou.txt plain text file that contains a list of commonly used password words. This file contains over 14,341,564 passwords that were previously leaked in data breaches.

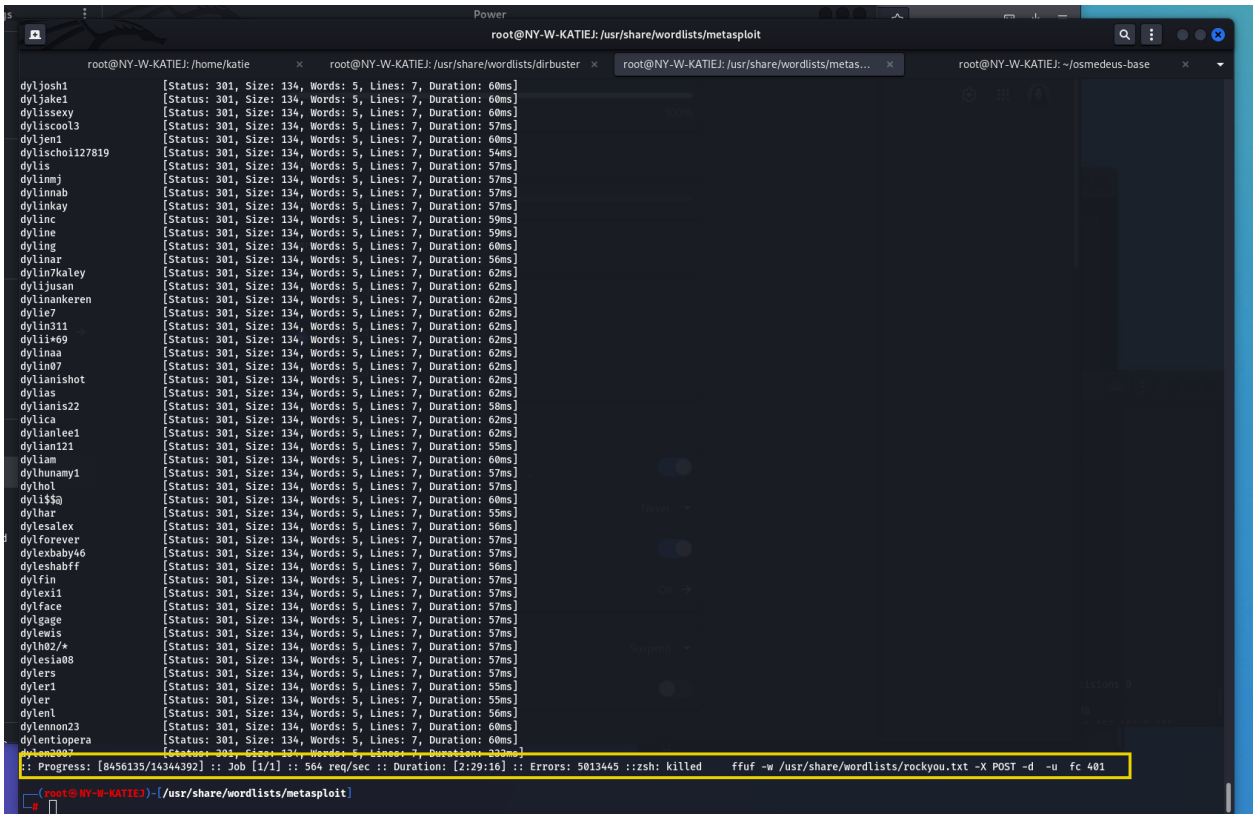


Highlighted portion refers to the command used to brute force the system.

- No matches were made; this was a relief.

FFUF

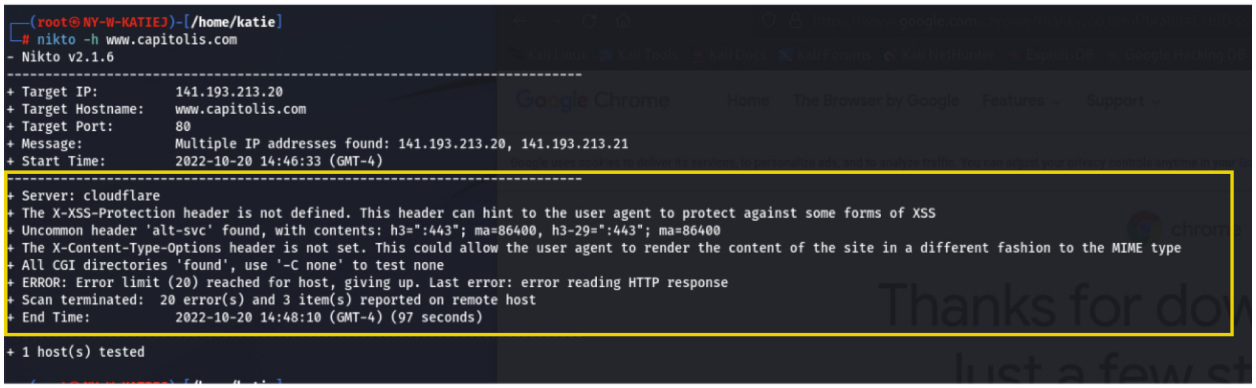
Ffuf is a fest web fuzzer written in Go that allows typical directory discovery, virtual host discovery (without DNS records) and GET and POST parameter fuzzing



After 2 hours 30 minutes and 846,135 the program errored out.

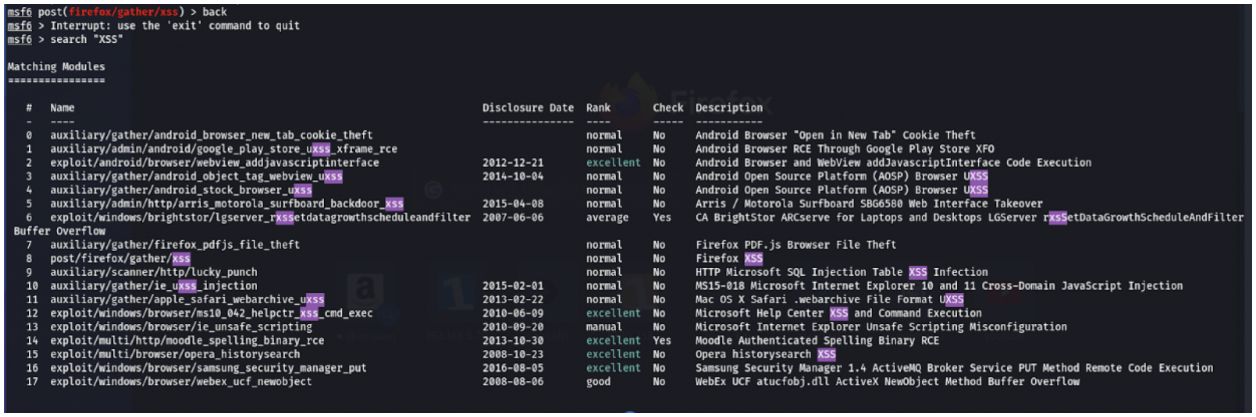
Website: Capitolis.com

- Using our old friend Nikto again, we see capitolis.com is a more vulnerable vector. Specifically:



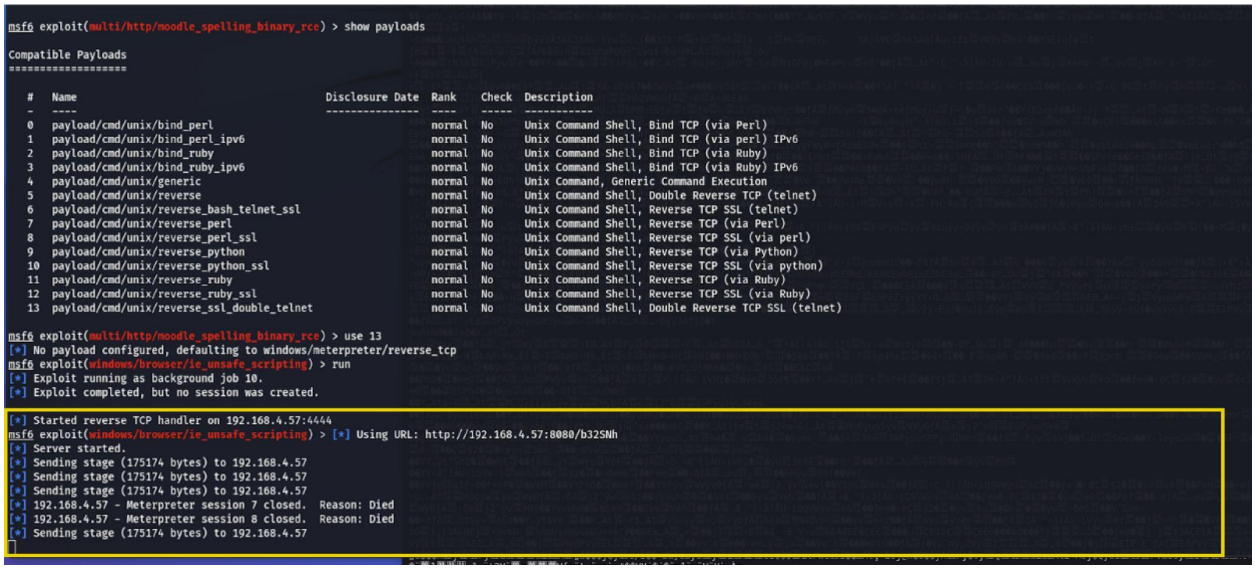
- Metasploit

The Metasploit framework is a very powerful tool which can be used by cybercriminals as well as ethical hackers to probe systematic vulnerabilities on networks and servers. Because it's an open-source framework, it can be easily customized and used with most operating systems.



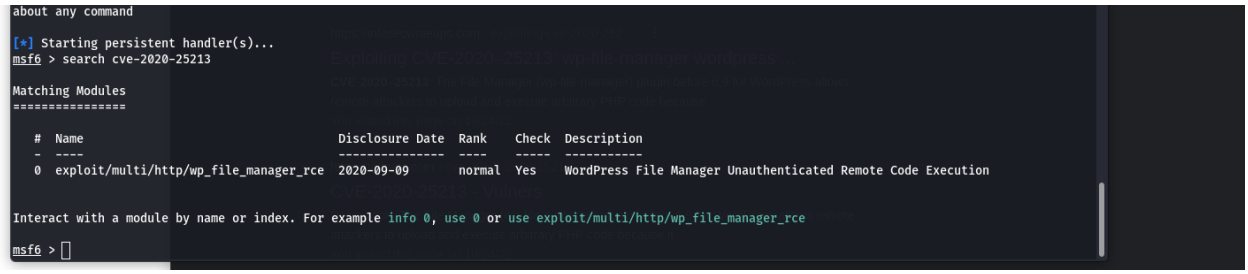
- Go root or go home

Searching for keywords provides exploits and ranks them from good to excellent as well as descriptions and creation dates. To run the exploit, you specify a payload and the remote host to execute on. I tried to set up an attack on the capitolis.com website but it died twice.

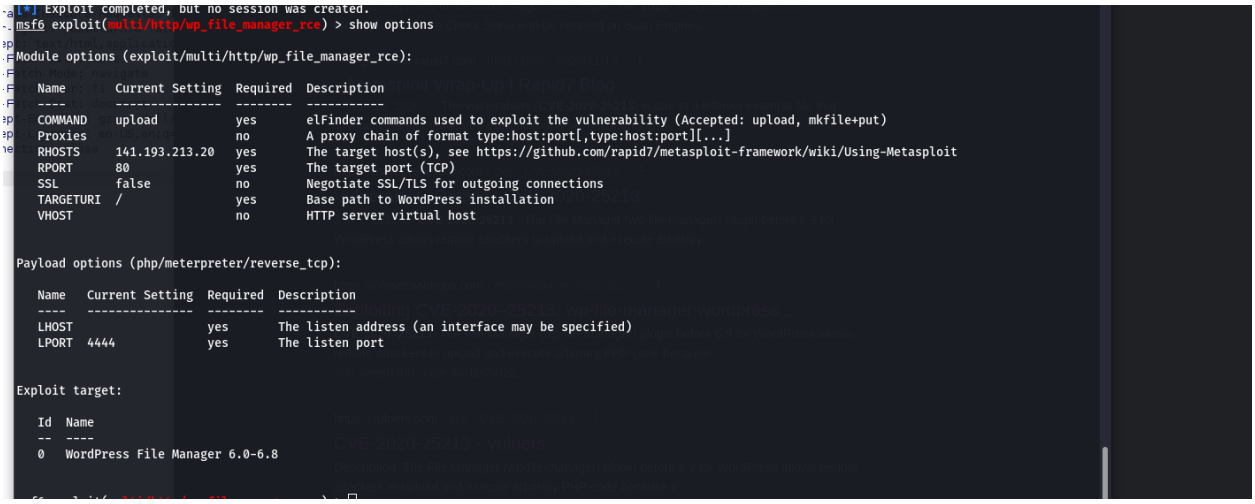


5. Rejection is Painful

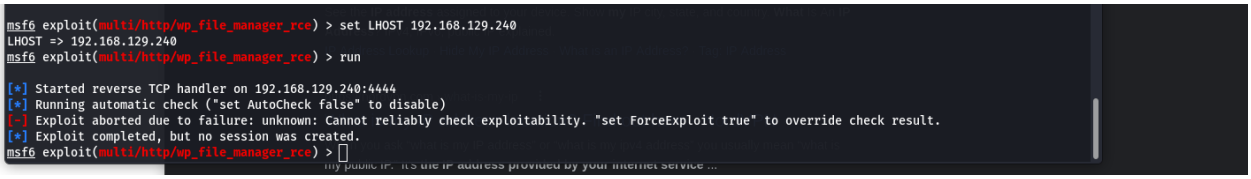
1. Metasploit Method
    - Searching the exploit in metasploit (of course it's there)



- The show options command will show you the available parameters for an exploit if used when the command line is in exploit context. The Flash exploit contains a total of 8 options from which only 4 are required:
    - Command (Required)
    - Proxies
    - RHOSTS (Required)
    - RPORT
    - SSL
    - TargetURI
    - VHOST
    - LHOST (Required)
    - LPORT (Required

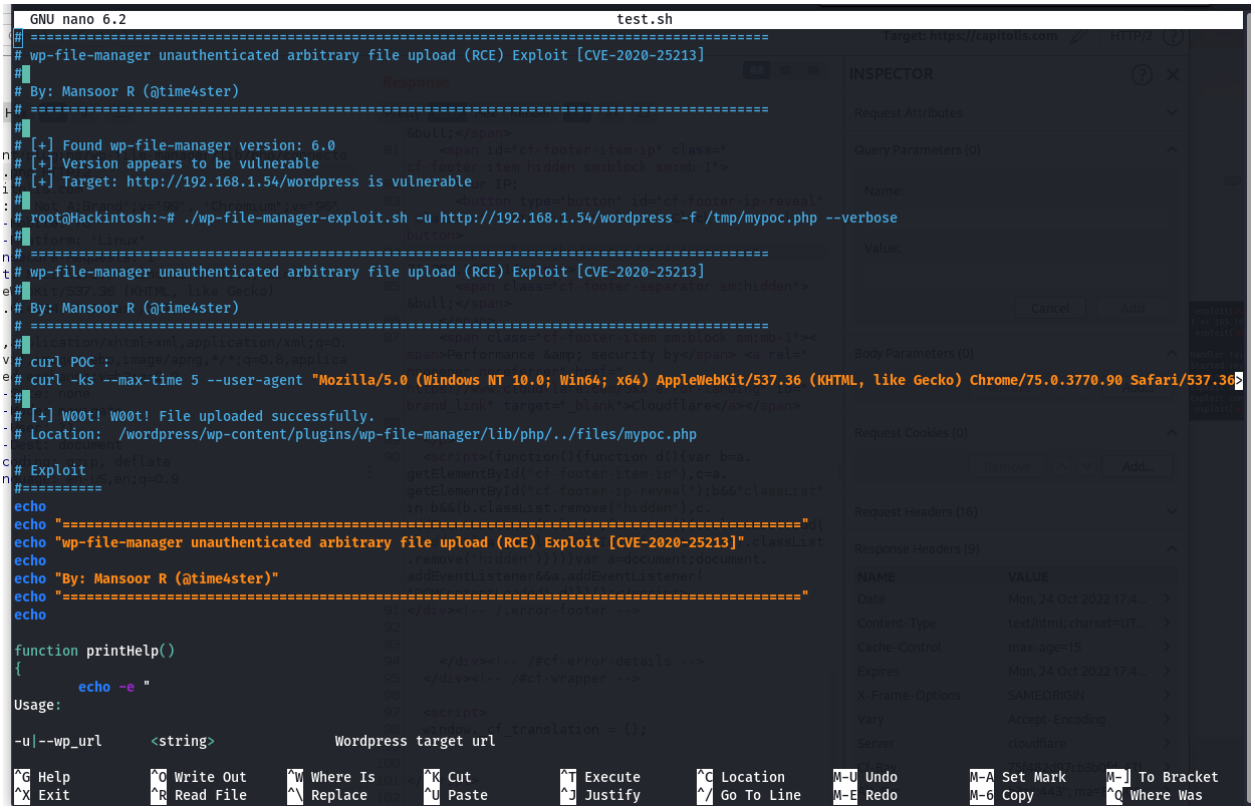


- Executing the command is unsuccessful



2. Premade script

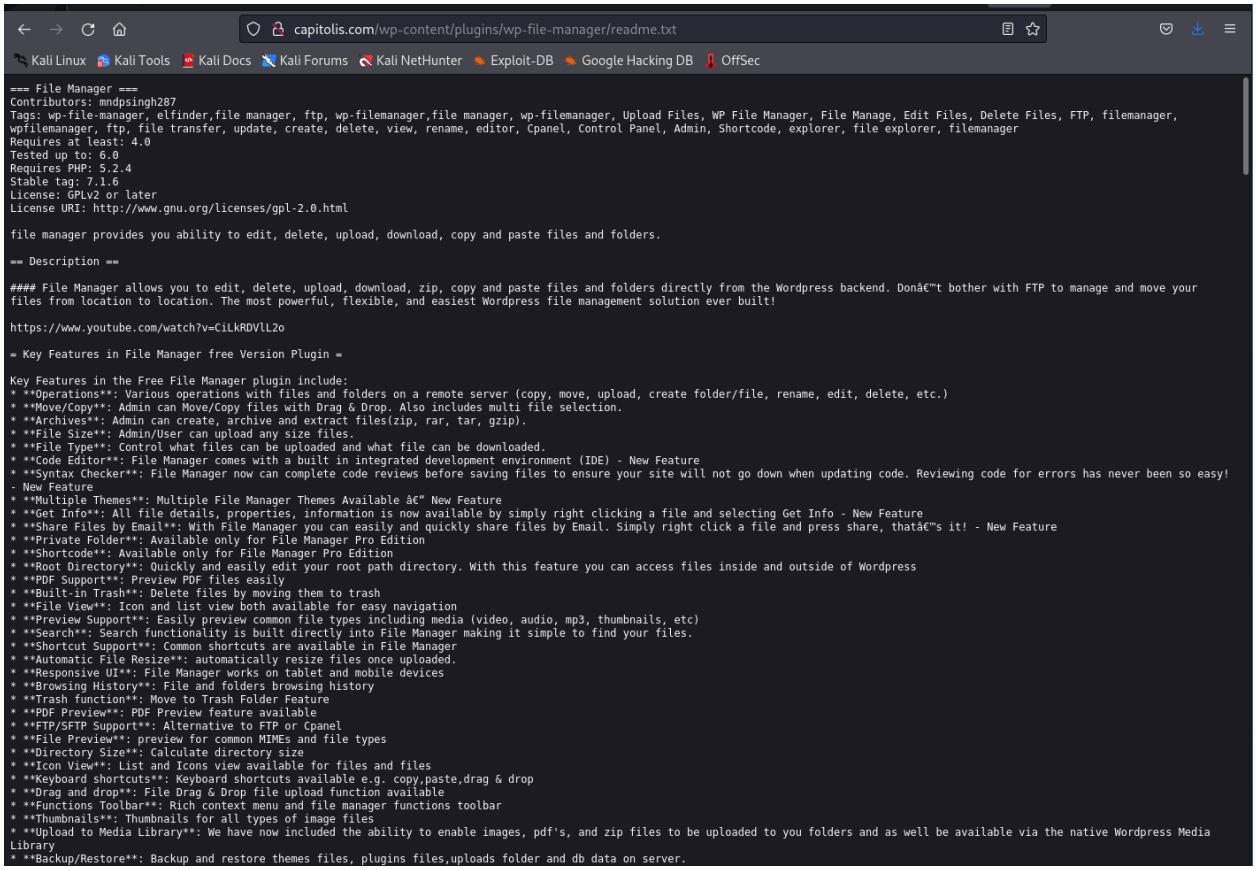
- Naturally a script exists that lets you deploy to the target of your choice:



- Not today satan

3. Editing the URL

The vulnerability lies in the URL: here is where you can preview the text file



But try to go one step further and WP knows….