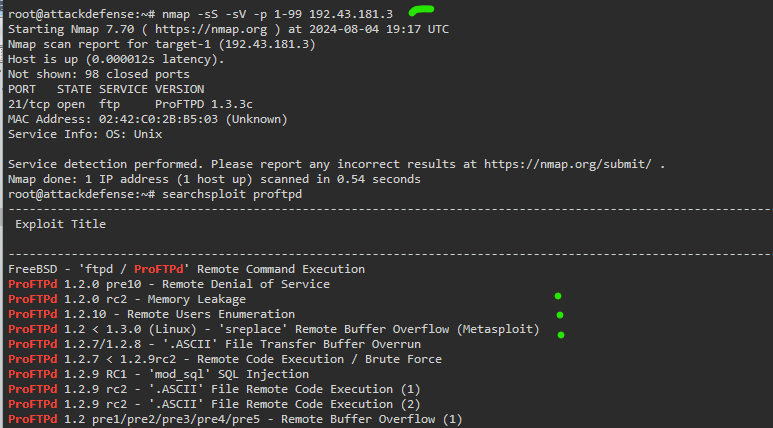
Start with a stealth scan on the target port, 192.43.181.3.

We only need the first 99 ports scanned. A teammate was working on this vector, and their notes mentioned a 2-digit port as an attack vector. Sadly, they forgot to note down what port it was. Rookie mistake in documentation.

Use -sS for a stealth scan, -sV to find versions running, and -p 1-99 for the first 99 ports

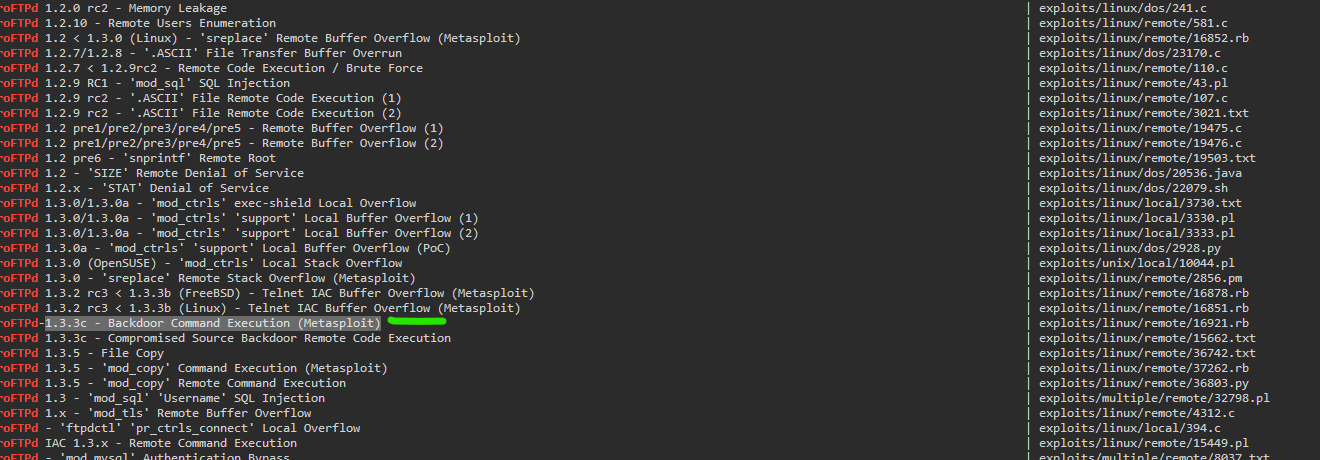
Hmm, seems port 21 is open, running ftp over proftpd 1.3.3c

Let’s searchsploit for proftpd



Scrolling down, we see a couple of results matching the version identified.

Let’s try metasploit’s backdoor through this channel as indicated below



1. Run another nmap scan, running a test as follows

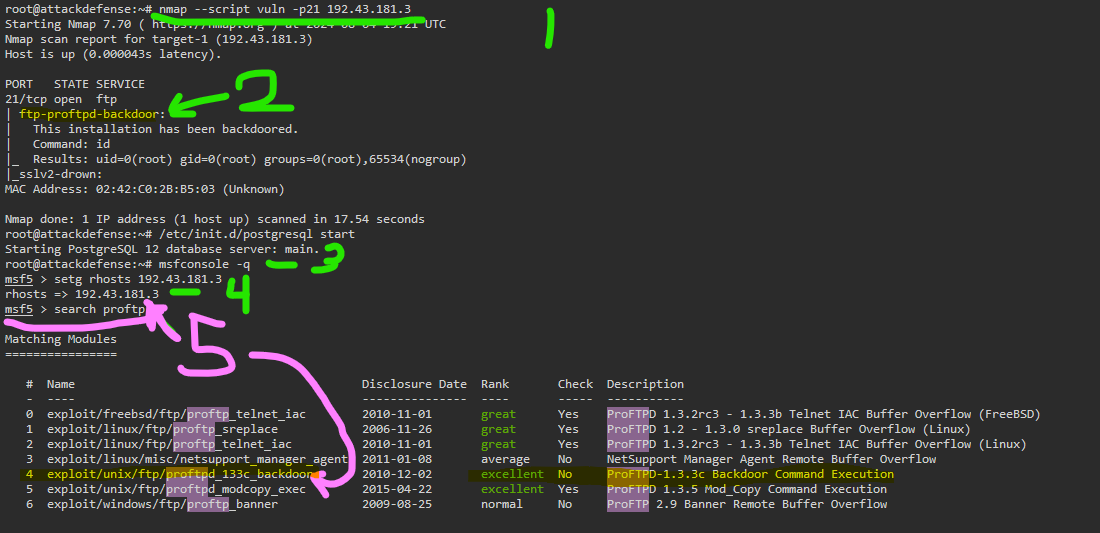
Command: nmap –script vuln -p21 192.32.181.3

Just to test if we can see there’s a vulnerability.

1. The extra info reveals to us that there is a backdoor, which matches our intended exploit.
2. Open msfconsole
3. set the rhosts globally (only one target machine here)
4. Search for proftp within msfconsole (when we were doing searchsploit before)

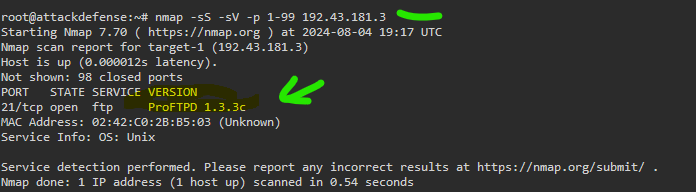
Option #4, as hilit in the list, is our backdoor.

See that name, 133c? Matches our version number for the service too.



Reminder of the version returned

Proftpd 1.3.3c



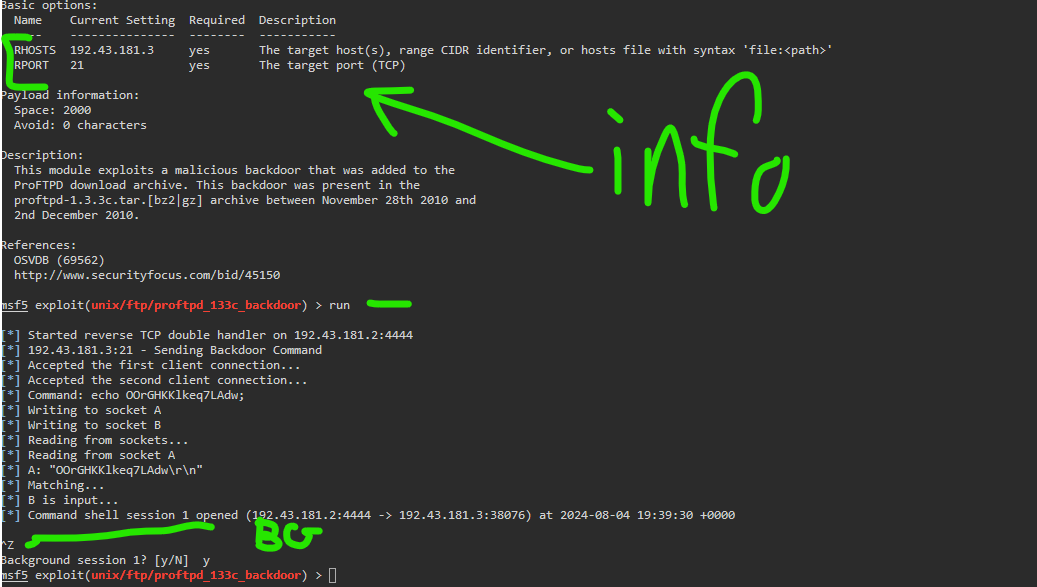
We got that double checked, so let’s look at the info for using the msf module called proftpd\_133c\_backdoor.

Only need the rhosts, and it assumes the port for ftp is standard, 21.

That’s the case, as was shown in the nmap scan, so we got everything configured

Run the exploit.

It opens a command shell, which we can set into the background with ctrl + z.



Back in Metasploit, we want to get the hashdup of this system.

Command: use post/linux/gather/hashdump

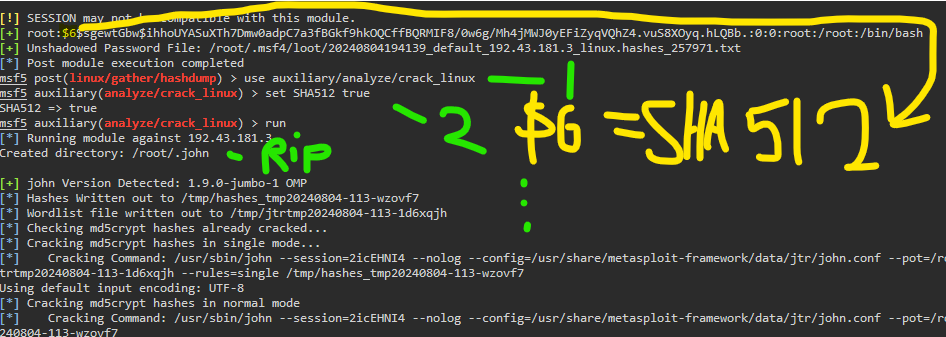
1. We see the info for this module, needing our session.
2. We have established our single session in the background, so we can indicate this with set SESSION 1.
3. Run the exploit.

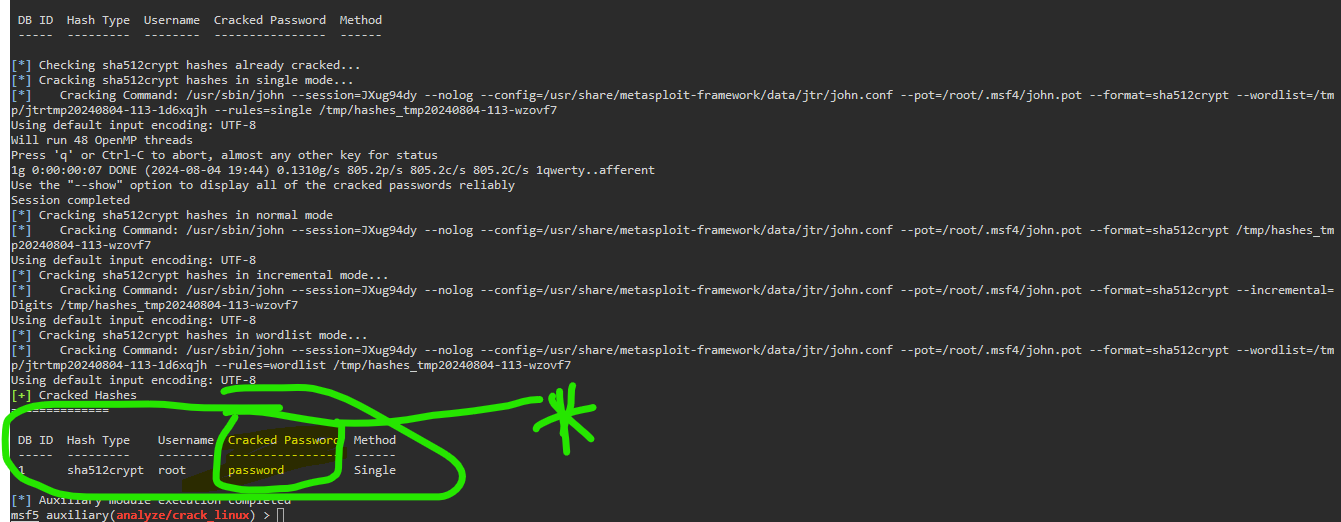
After running, we see info auto-grabbed about root.

Hmm, that $6 indicates some kind of hashing being done on the encoded info there. There’s a table explaining how these numbers relate to which hashing algo is used, but I think this is Sha-512.

Let’s point the auxiliary cracker to the target, indicating that it’s SHA512

This will use john the ripper for processing the hash.





We find the password here, and can now become root.

That’s all for this exercise