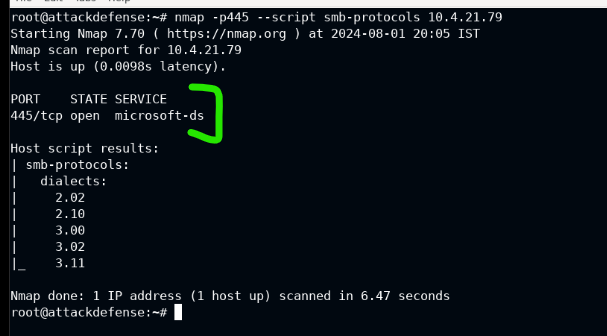
SMB ps server exec

The target ip is 10.4.21.79, port 445 has been identified OPEN in an nmap scan

Check protocols supported by smb

nmap -p445 --script smb-protocols 10.4.21.79



!!! ERROR FOUND IN LESSON!!!

The educational material misses the step of logging into msfconsole with the command:

Msfconsole -q.

Easy to overlook… anyway…

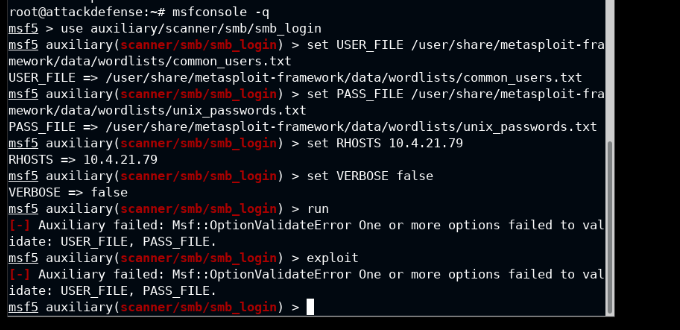
We’re using the exploit called Smb login. It’s essentially letting us brute force a bunch of options from our word lists that we load into it.

I had trouble loading the wordlists because of the long pathing, but taking some time to double-tab and read out the options each step of the way, I became more familiar with the hidden file structure that’s under the hood.

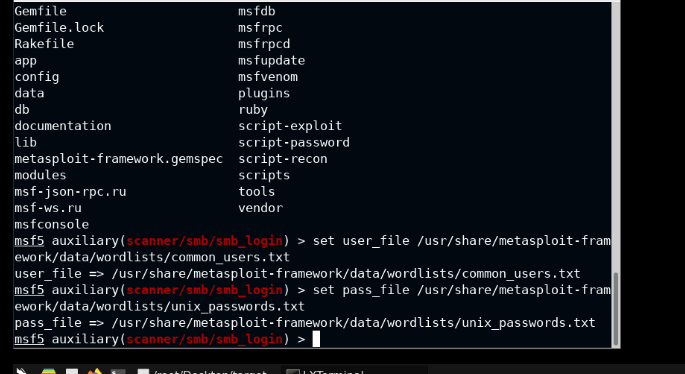
!! note for future improvement !!!

None of these lessons explain what the given file structure is like, and there’s minimal exploration of what’s actually surrounding all these folders within folders, leading to an obvious sense of disorientation.

I do not have the familiarity to make a better lesson plan, but I’m as student. This requires more refinement, and better educators… this whole thing of “figure it out on your own” is not a valid teaching methodology.

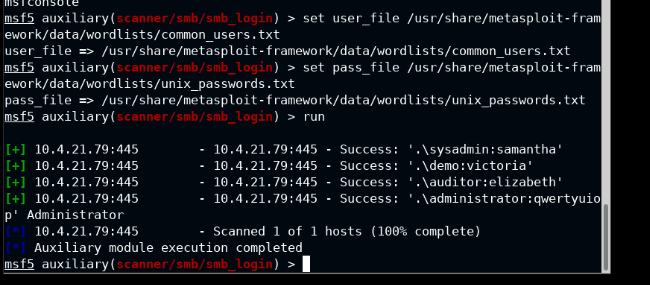


(above, there are errors in the loading process… requires some elbow-grease to get through)



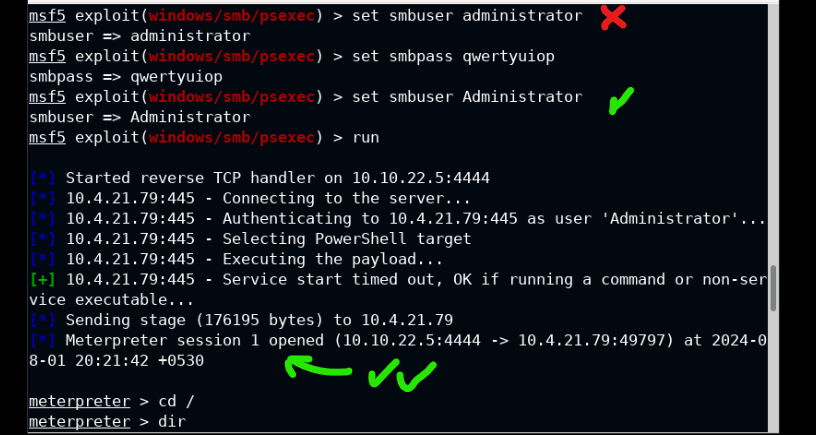
Using double-TAB, I can see what auto-fills after a letter or two, giving me a list of possible items. Through this, I have some kind of feedback as to what’s in the directories, and I’m able to figure out the correct pathing.

Actually executing the exploit does the following 🡪



This gives us some people we can log in as, and we want to jump straight to targeting the administrator. No fluff with lateral movement or priv. escalation, just right to the throat.

With credentials in hand, now we can run psexec to pop open a meterpreter shell



(Note: capitalization is important, so if you mess up inputting something like the username, then just re-input it to overwrite the data as shown in commands 1 and 3 above)

\*\*BONUS\*\*

Once we got our meterpreter shell open, we’ve cd’d into the main directory, but there’s a lot of junk here. I am going blind from screen eye-strain, so let’s make it simpler in searching for a file of a target name. We know we’re after something named “flag”, so how do we get it?

Grep isn’t working in this context, but we got some other witchcraft using /s /b to search for files whose name contains flag (which we surround with wildcards).

/s searches current dir and all sub-dirs, so if something’s embedded in a folder we can find it.

/b gives us the path of the file we hit

It’s a little different than regex, but it gets the job done. In our example, the path is just “flag.txt” with no sub-directories or anything because it’s in the current directory. At least we got rid of the clutter this way.

