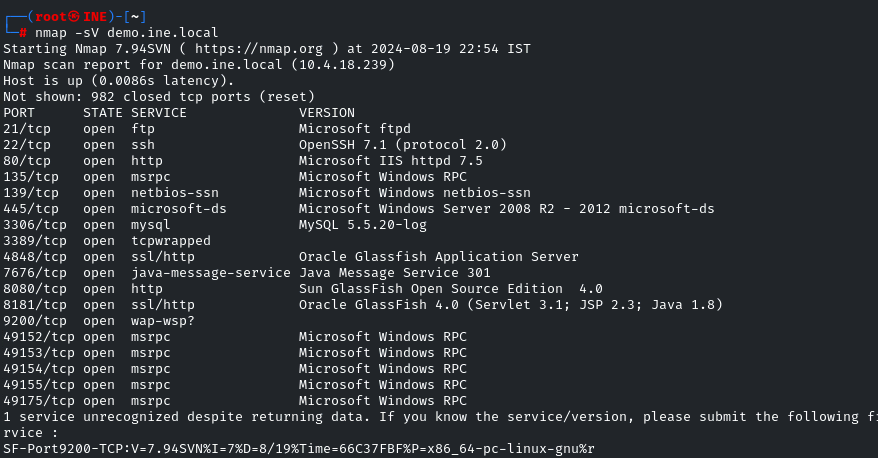
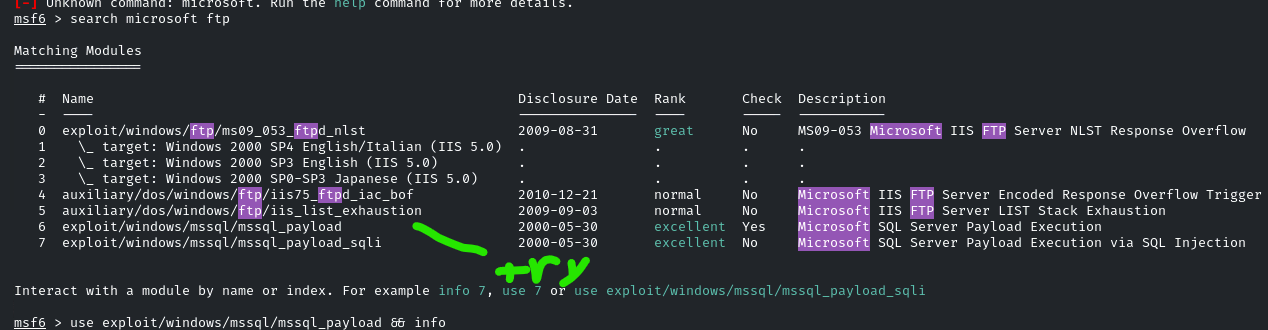
Starting with a service scan on the target demo.ine.local, a team is dividing tasks up to tackle.



Scenario: We’re attempting to crack ftp as a vector.

Let’s start with searching msfconsole for anything we can use.

Below, #6 looks ok. It’s got an excellent rank and might be easy to use.

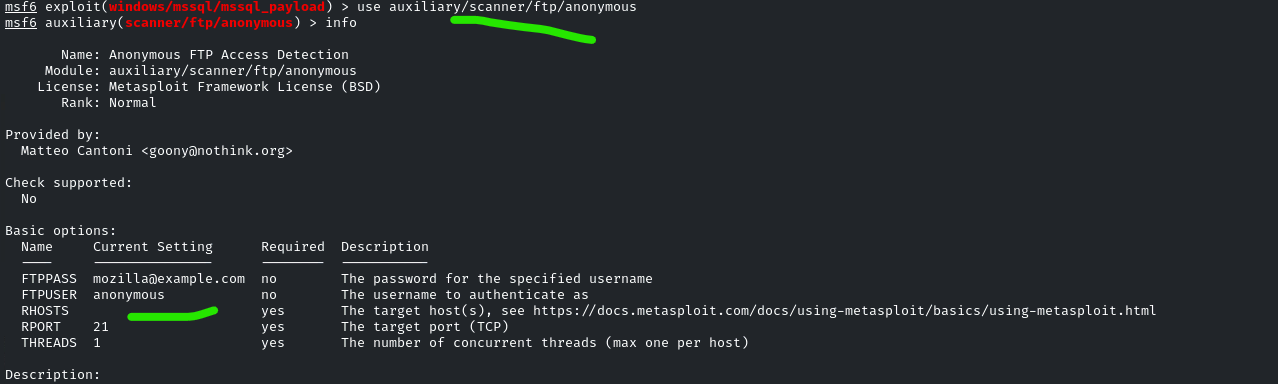
Well, while researching that, I just got straight up told to check for anonymous logins first, so let’s do that. Not sure why, but if the captain says to do it, it’s gotta be done somehow.

The auxiliary module below can help

Command: use auxiliary/scanner/ftp/anonymous

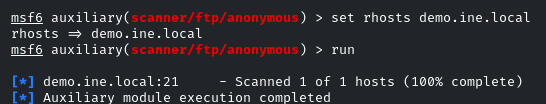
Checks if anonymous login is available… which should also be visible from just an nmap -sV scan… but apparently that’s not good enough? The captain isn’t being clear *why* this is needed…

Well, I just need to set the RHOSTS to the target ip address, it seems.



well, there’s no output, and that was a waste.

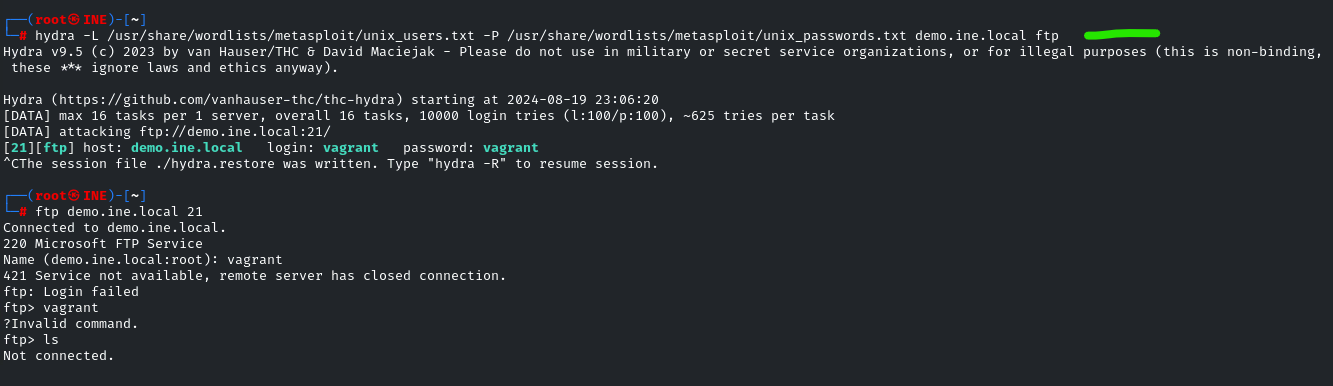
I could’ve told you that from the nmap -sV scan…



I got told to just ditch my previous attempt in looking at the msf modules and to pivot to a hydra bruteforce, so let’s go do that.

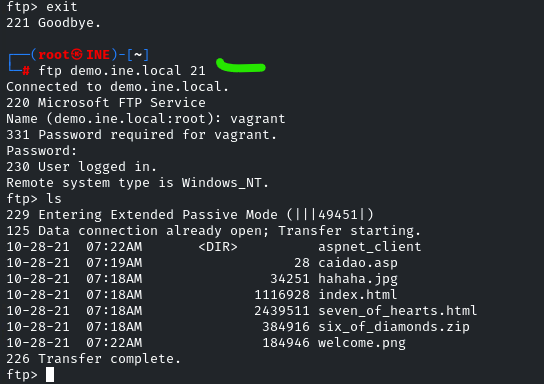
We find the user an pass as both being “vagrant”

Wait, why is login failing?



Okay, exit out and try again… must’ve typed in something wrong. It happens.

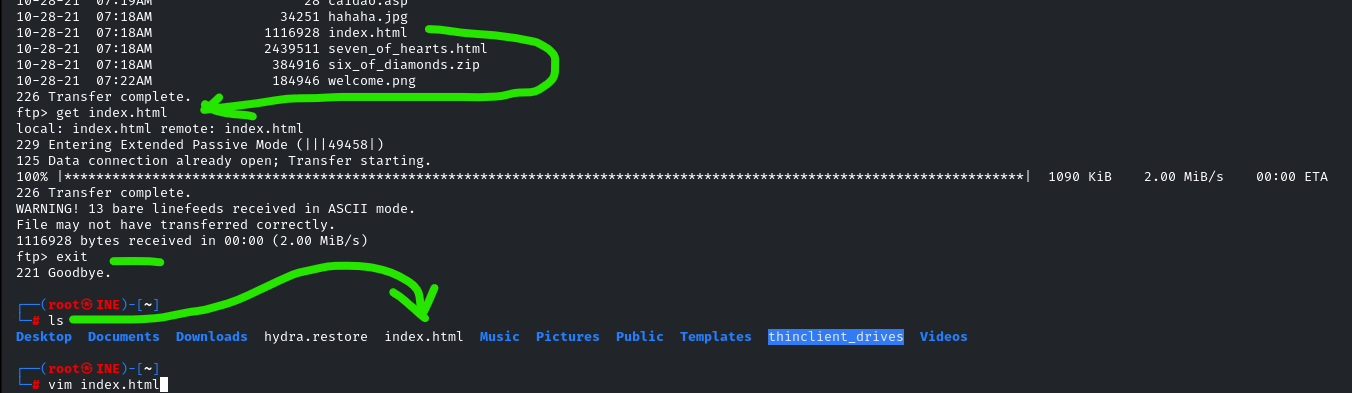
We’re in ftp via command line now



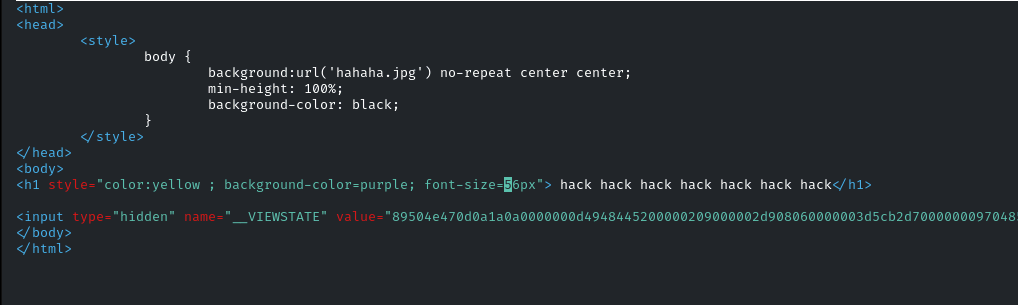
New orders: deface the index.html to prove we can compromise their data. They want something they can see clearly.

Use command: get index.html

That downloads the remote file, index.html, to the local attack machine.



…



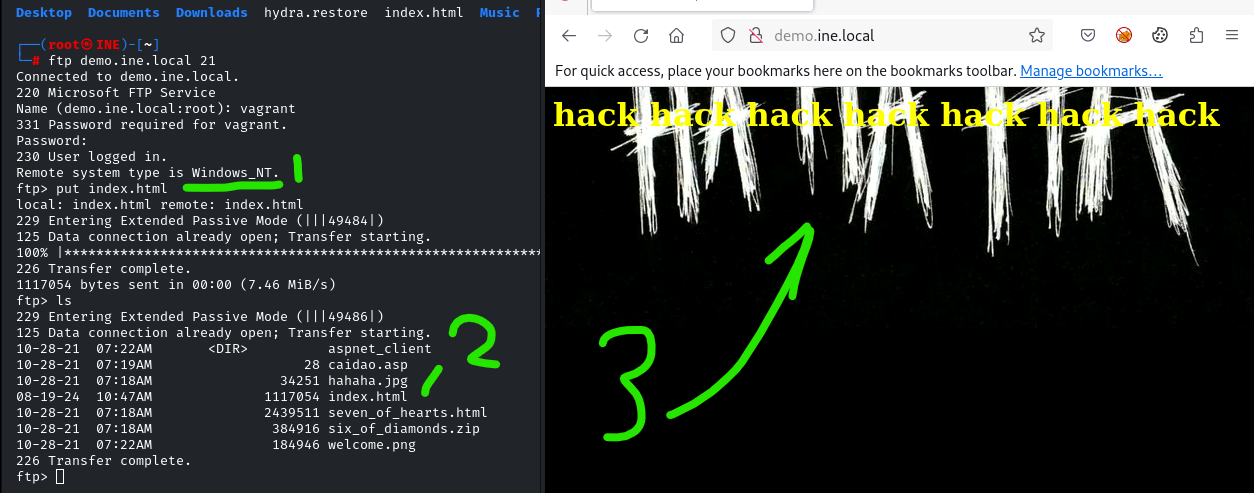
Edit some stuff in the webpage to make it obvious….

This h1 should catch their attention, hacky as it is.

Save the file locally. Next, we’ll put it on the remote victim machine

1. Reconnect to ftp with the previous credentials. Then use command: put index.html . By matching the previous filename, we overwrite it in place
2. We can see the index.html in the file structure. No new file, because existing one with same name is overwritten.
3. In the browser, we see the modifications with the h1 overlayed on the defaced website

NOTE: this is a bad visual for the lab. The lab was pre-configured to have this dumb “hahaha” image as the background as the default for the website. This is a baffling decision and another grain of sand in the bag.



Well, that’s all for defacement, I guess.