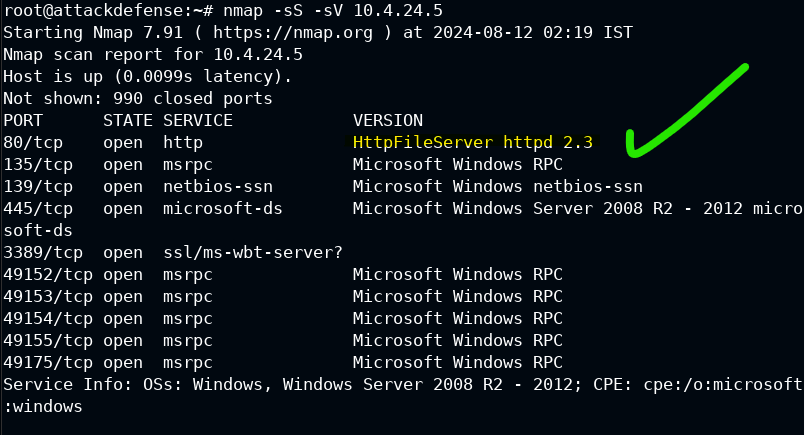
Start by scanning the target.

We’re only interested in our good friend HFS as far as the



…

Reminder to use ip addr to find your own address as a listening port. In this case it’s eth1. But that’s just for this particular environment.

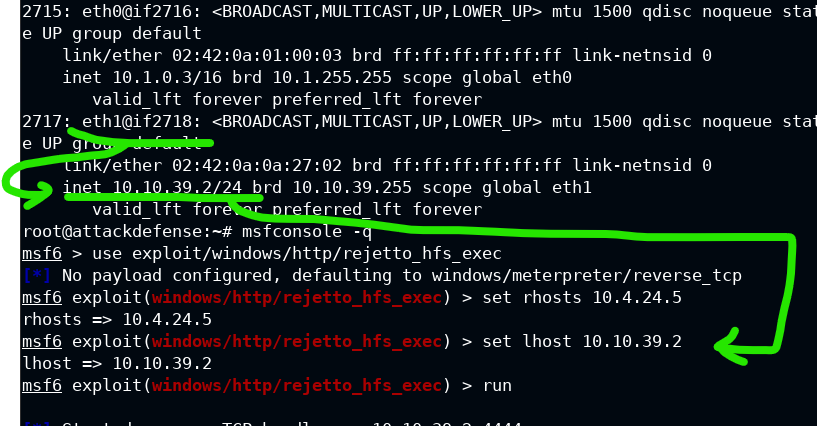
When using hfs exec, set the lhost to your own ip, which should be the default.

Rhost is the target machine’s ip address.

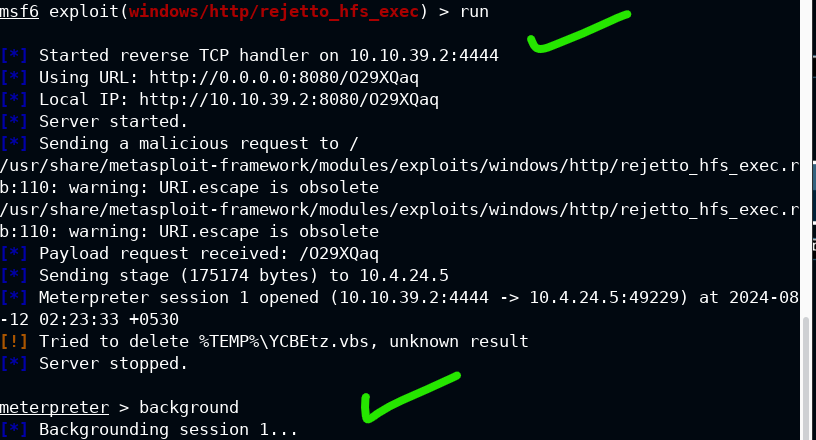
If you put these on a graph imagine your computer on the left, and the target on the right.

Lhost 🡪 Rhost

[[my machine ]] ---- 733t H4Xx 🡪 [[target machine]]



Meterpreter is opened, but we can send it to the background



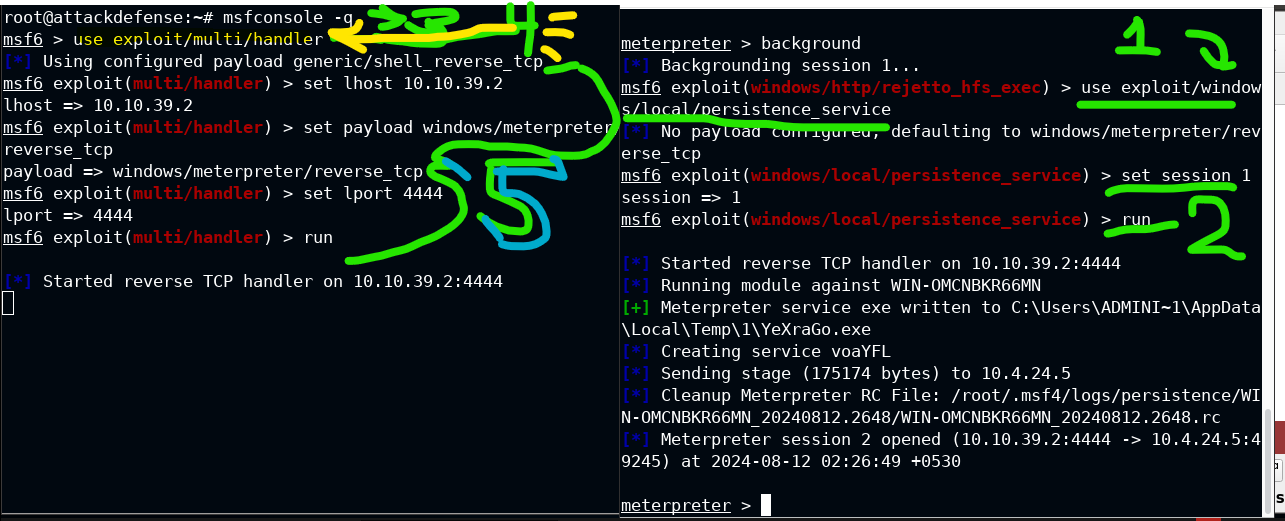
This is tricky now.

We want to set up persistence\_service, so we have 2 terminals open. In msfconsole, we 1) use the exploit for the persistence service, 2) set the session to the one we put in the background, and run it.

While this is active, we 3) switch to a second terminal with msfconsole (the -q just skips the intro ascii art)

Use the multi\_handler here, where we need to 4) set the lhost to our device, the payload as the meterpreter\_reverse\_tcp (since it was specifically a meterpreter session we opened in the background on the other terminal’s interface.)

The lport can be anything not active, but 4444 is usually fine. Then 5) run the program, it should givea reverse tcp handler started message.



Next, we have to reboot. The session, and that should give an elevated meterpreter?

Wait, the instructors weren’t clear about this… what did we even do all of this for? There was no objective even set from the beginning….