Lab10- Firewalling and C2

In this lab, we're going to implement a command and control type of system and implement a firewall in order to protect against suspicious traffic. After all, if you're going to create a C2 server, you aren't going to want anyone to tamper with it. This assignment will be used as a basis for future labs. The specifics of these requirements are fairly light—you're welcome to use your creative interpretation on this task. It will overlap with future labs as well.

Instructions

Perform the following tasks in an environment of your choosing. The IA Lab has a number of useful resources for accomplishing these tasks, but you're welcome to accomplish this on your own as well. There is no one right or specifically wrong mechanism for completing this lab; implement this lab in a way that you see fit, assuming they meet the guidelines.

1. Create a network, using a pfSense firewall, connecting a server to the LAN interface and a 'compromised client' to the WAN interface.
    a. In this particular instance, the C2 server will have a static IP address, as will the client.
    b. Total of 3 VMs, feel free to use an OS if your choosing for the Client and Server
    c. If using the IA Lab, you can connect your WAN to the Projects_Internet interface, that'll gain you internet access. Feel free to use it for your 'compromised client' too, just be mindful that you're getting your IPs via DHCP.
2. Using Python, create a C2 Server that logs data reported back from the client.
    a. You're welcome to reuse work you completed in CSC840 for this task.
    b. This will live on your LAN network/server. You're the attacker in this situation, this server is your 'home base' of controlling your hoard of clients.
    c. Choose an appropriate port that the server will operate on. Keep in mind that some ports may appear suspicious compared to others (for example, TCP4444 would be a poor choice compared to 53 or 80).
    d. Ensure your connections are taking place over TCP
    e. Implement basic status codes for communication back and forth. At a minimum, implement the following commands the server can report back with:
        i. Shutdown the host
        ii. Shutdown the C2 client
        iii. Report the MAC
        iv. Upload a file at a given path
    f. The server can be interactive (IE: if you press 3 on the server's console, all the clients will report back their MAC address, otherwise they blindly sit idle waiting for instructions)
    g. Regarding the port you choose, try and make your traffic look legit. For example, if I'm on port 80, I'll obfuscate my traffic to appear like real HTTP traffic—(client requests out using "GET /someURL/ HTTP/1.1" etc)

3. Using Python, create a C2 client that reports to the server.
    a. Ensure the client can report the requested data when contacting the server
    b. The client shouldn't be interactive, it should sit in the background & not require user intervention to work. This would be poor malware after all.
    c. Ensure that your client properly fakes their traffic to look somewhat legit
    d. You may statically connect to the server or use a domain generation algorithm to determine your server's IP address from the client.
4. Compromise your client
    a. Load your "malicious code" onto the client
    b. Ensure the client is reporting back to the server

What to Submit

This lab is going to be mostly screenshot and python based. Ensure all of your screenshots are contained in a single PDF (not Word) document. Include your python scripts as standalone text files (also, do NOT paste any code into a Word document, that's a nasty thing to do).

Turn the following screenshots and other files in to the D2L Dropbox:

1. A screenshot showing your C2 client uploading a file to the server.
2. A screenshot showing your firewall rules in pfSense.
3. Upload your python scripts as separate documents to the Dropbox.

Grading

Your grades will be based off of a rubric that includes the following items:

1. Successful completion of C2 server
2. Successful completion of C2 client
3. Successful firewall implementation
4. Proper formatting, following good technology standards, and meeting submission guidelines.