This README written by Jeffrey Harnois

This project contains:

Mission 3

VM2-

Simple-https-server2.py - part of the infastructure, responses to the real traffic being generated

Key.pem - The private key involved in the https traffic, private key of the server

Server.pem - the certificate generated by the CA for the server to allow connection

Attack_server.py - opens up a connection on port 4443 that listens for instructions from the Command and Control unit to execute on the infected host

Queue_exit_resp.py - Due to the fact that we can't access the TCP headers from the server, we send traffic and modifies the packets going back to the C&C from the attack server

Server_scapy_test.py - looks at packets from C&C and looks at the information stored in the urgent pointer. It attempts to run the command that was in the urgent pointer and writes the result into a file (not included due to the fact that the script deletes the file before it starts). It then sends what is in the file to the C&C using the attack server

VM3-

run_snort - a batch file needed to run as root that will implement any of the Snort rules that are contained in the local.rules file

Local.rules - contains all of the rules for Snort that will alert or drop packets

Barnyard2.conf - all the configurations for Barnyard2, the software that will allow snort to write to a MySQL database that is running on the computer

Snort.conf - a file containing all of the configurations we modified for Snort to run on the LAN and take our IP addresses into account while attempting to run Snort.

databaseInterface.py - a file that has functions that would allow us to be able to communicate with our Barnyard2, MySQL database. We attempted to use this as part of our defense before our NFQUEUE became an issue

VM4-

Real_traffic.py - a file that will create the real traffic before our attack or defense is being implemented, it just generates a bunch of HTTP requests to the simple-https-server2 on port 443

VM5-

Attack.py - the server set up to talk with the infected hosts and send messages through the urgent pointer to

Instruction_response.py - This file is used to read the response from what the webserver send back to the C&C and translates it, putting it into a file so that the attack server and read the response.

Queue_edit.py - as the packets are being sent out by the attack server, this file uses NFQUEUE to change the headers before it is sent. Is mainly focuses on changing the urgent pointer header to be able to store the malicious instructions.

VM6-

Gen_background_traffic.py - a multithreaded python script that will blast the entire network with packets that attempts to overload Snort as well as create cover traffic for the real attack packets that are going to and from the infected hosts and the C&C

Server.pem - see VM2