Part 1 – Logging to PCAP Files

Execute the following commands on your az-fer device:

```
tcpdump -i any "net 192.168.0.0/16 or net 172.16.0.0/16" -w ~/apache-iis.pcap
```

This will create a PCAP file that can be analyzed with Wireshark later.

While this command is running, access your IIS and Apache pages from your client. **(Be sure your pages aren't cached.) When you have completed this step, stop the packet capture. (Ctrl-c)**

Analyzing Logged Packets

You will now read through the captured packets found in ***~/apache-iis.pcap*** and save specific packets to separate text files for later submission. Be careful here! Only include the correct number of packets specified in the instructions below.

Using Wireshark, open the file **apache-iis.pcap** and mark the following packets:

1. A DNS request from your Client, requesting the FQDN of the IIS server
2. A DNS reply to your Client with the FQDN of the IIS server
3. An HTTP request from your Client to the IIS Server
4. An HTTP reply from your IIS server to your Client
5. A DNS request from your Client, requesting the FQDN of the APACHE server
6. A DNS reply to your Client with the FQDN of the APACHE server
7. An HTTP request from your Client to the APACHE Server
8. An HTTP reply from your APACHE server to your Client

Export these packets to a **PCAP** file called **APIIS-transaction.pcap.** ***This file should only have 8 packets in it***.

Part 2 – Adjusting iptables To Log Dropped Traffic

Before beginning, rename your current iptables script to **scenario0.sh** to back up your CP3 work.

Read the following iptables command and understand what it does:

Original:

```
# An example of accepting MYSQL packets by source port
iptables -P FORWARD DROP
iptables -A FORWARD -p tcp --sport 3306 -j ACCEPT
```

Logged and Dropped:

```
# An example of logging (instead of accepting) MYSQL packets by source
port
iptables -P FORWARD DROP
iptables -A FORWARD -p tcp --sport 3306 -m limit --limit 10/min -j LOG
--log-prefix "DROPPED-"
```

In order to log dropped packets for any service either by source port or destination port, we can simply **edit and replace the line for the particular service and port that is to be dropped and logged.** (This is a visual example. Do not modify scenario0.sh.)

The example above will **LOG** up to **10 packets per minute** that match the **source port** of the **MySQL** service. After a packet is checked against this line and logged, it will continue down the **FORWARD** chain looking for a final match. When it gets to the end of the chain without finding a match, the packet will be dropped because the default policy of the **FORWARD** chain is set to **DROP**.

Creating iptables Scenario1 – scenario1.sh

Copy ***scenario0.sh*** to ***scenario1.sh***. Edit this iptables configuration of ***scenario1.sh*** using the methods mentioned above to block the following services:

- The SSH source port of the Linux server
- The FTP destination port of the Windows server (control port, not data port)

Execute the following command on your az-fer device:

```
tcpdump -i any -w "net 192.168.0.0/16 or net 172.16.0.0/16" ssh-
ftp.pcap
```

While this command is running, attempt to access your FTP server and SSH into az-ls from the client. **(Be sure to wait a few seconds to ensure each transaction is attempted and data is generated in the logs.) When you have completed this step, stop the packet capture.**

1. Using Wireshark, open the file **ssh-ftp.pcap**, mark the following packets and export to a PCAP file called **SHFTP-transaction.pcap.** ***This file should only have 7 packets in it.***
    - A DNS request from your Client, requesting the FQDN of the FTP server
    - A DNS Reply to your Client with the FQDN of the FTP server
    - An FTP request from your Client to the FTP Server
    - A DNS request from your Client, requesting the FQDN of the SSH server
    - A DNS Reply to your Client with the FQDN of the SSH server
    - An SSH request from your Client to the SSH Server
    - An SSH reply from your SSH server to your Client

2. In your */var/log/messages* file, find the following packets, and export to **DRPSSHFTP.log.** ***This file should only have two packets in it***.
    - A DROPPED packet with a destination port matching FTP (21)
    - A DROPPED packet with a source port matching SSH (22)

3. Using Wireshark, open the file **ssh-ftp.pcap,** mark the following packets and export to a file called **SSHFTPDR.pcap.** ***This file should only have two packets in it.***
    - A packet with a source port matching SSH and an **ID number in the IP packet info** that **matches** the dropped SSH packet **you found in step 2.**
    - A packet with a destination port matching FTP and an **ID number in the IP packet info** that **matches** the dropped FTP packet **you found in step 2.**

Part 3 – Submission

Blackboard Submission

Submit the following as separate, non-zipped files for your CP4 submission. There is no live demonstration for this checkpoint.

1. All .pcap and .log files created during this checkpoint. This includes:
    a. apache-iis.pcap
    b. APIIS-transaction.pcap
    c. ssh-ftp.pcap
    d. SHFTP-transaction.pcap
    e. SSHFTPDR.pcap
    f. DRPSSHFTP.log
2. All iptables scenario scripts created during this checkpoint. This includes:
    a. scenario0.sh
    b. scenario1.sh