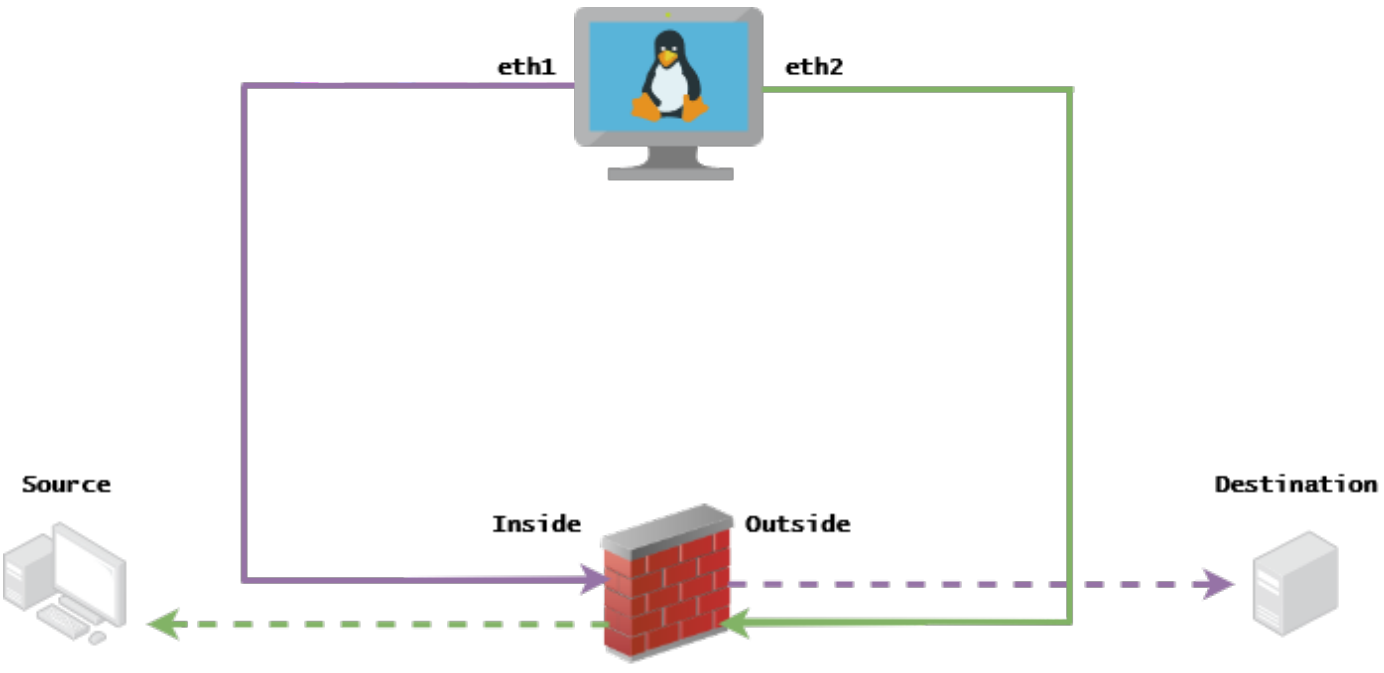
Configure TCP Replay with 2 NICs on Kali Linux

Contents

Introduction

This document describes TCP Replay to replay network traffic from PCAP files saved with packet capture tools.

Topology



Requisites

- VM with Kali Linux and two NICs
- FTD (Preferably managed by FMC)
- Linux knowledge to run commands.

Background Information

TCP Replayis a tool used to replay network traffic from pcap files saved with packet capture tools like wireshark or TCPdump. It can be useful for situations where you need to replicate traffic to test the outcome on network devices.

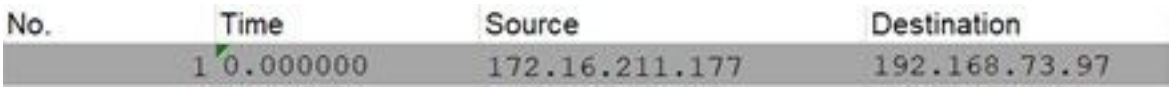
The basic operation of TCP Replay is to resend all packets from the input file(s) at the speed at which they were recorded, or a specified data rate, up to as fast as the hardware is capable.

There are other methods to perform this procedure, however, the purpose for this article is to achieve TCP Replay without the need of a middle router.

Implementation

FTD Configuration:

1. Configure the Inside/Outside interfaces with an IP on the same segment that you have on your packet captures:



No.	Time	Source	Destination
1	0.000000	172.16.211.177	192.168.73.97

- **Source:** 172.16.211.177
- **Destination**: 192.168.73.97

FMC > Devices > Device Management > Interfaces > Edit each interface

> **Tip**: It is best practice to assign each interface into a different VLAN to keep the traffic isolated.

Running-config (example)

```
interface Ethernet1/1
 nameif Outside
 ip address 192.168.73.34 255.255.255.0
!
interface Ethernet1/2
 nameif Inside
 security-level 0
 ip address 172.16.211.34 255.255.255.0
```

2. Configure static routes from the hosts to their gateways and fake ARP entries to them since these are non-existent gateways.

FMC > Devices > Device Management > Routes > Select your FTD > Routing > Static Route > Add Route

Running-config (example)

```
route Inside 172.16.211.177 172.16.211.100 1
route Outside 192.168.73.97 192.168.73.100 1
```

Use the LinaConfigTool backdoor to configure fake ARP entries:

1. Login to the FTD CLI
2. Go to expert mode
3. Elevate your privileges (sudo su)

LinaConfigTool Configuration example

```
/usr/local/sf/bin/LinaConfigTool "arp Inside 172.16.211.100 dead.deed.deed"
/usr/local/sf/bin/LinaConfigTool "arp Outside 192.168.73.100 dead.deed.deed"
/usr/local/sf/bin/LinaConfigTool "write mem"
```

3. Disable the equals sequence number randomization.

1. Create an Extended Access List: **Go to FMC > Objects > Access List > Extended > Add Extended Access List**Create the ACL with parameters "allow any any"
2. Disable sequence number randomization: **Go to FMC > Policies > Access Control > Select your ACP > Advanced > Threat Defense Service Policy**Add Rule and select **Global** Select your previously created **Extended ACL**Uncheck **Randomize TCP Sequence Number**

Running-config

```
policy-map global_policy
class class-default
set connection random-sequence-number disable
```

Linux Configuration:

1. Configure the IP for each interface (This is based on which one belongs to the inside subnet and the outside subnet) ifconfig ethX <ip_address> netmask <mask> example: ifconfig eth1 172.16.211.35 netmask 255.255.255.0
2. (Optional) Configure each interface into a different VLAN
3. Transfer PCAP file into the Kali Linux server (You can get the pcap file with tcpdump, captures on the FTD, etc)
4. Create a TCP Replay cache file with **tcpprep** tcpprep -i input_file -o input_cache -c server_ip/32 example: tcpprep -i stream.pcap -o stream.cache -c 192.168.73.97/32
5. Rewrite the MAC addresses with **tcprewrite** tcprewrite -i input_file -o output_file -c input_cache -C --enet-dmac=<ftd_server_interface_mac>,<ftd_client_interface_mac> example: tcprewrite -i stream.pcap -o stream.pcap.replay -c stream.cache -C --enet-dmac=00:50:56:b3:81:35,00:50:56:b3:63:f4
6. Connect NICs to the ASA/FTD
7. Replay the stream with **tcpreplay** tcpreplay -c input_cache -i <nic_server_interface> -I <nic_client_interface> output_file
   example: tcpreplay -c stream.cache -i eth2 -I eth1 stream.pcap.replay

Validation

Create packet captures on your FTD to test if the packets which arrive into your interface:

1. Create packet capture on the Inside interface cap i interface Inside trace match ip any any
2. Create packet capture on the Outside interface cap o interface Outside trace match ip any any

Run the tcpreplay and validate if the packets arrive into your interface:

Example scenario

```
firepower# show cap
capture i type raw-data trace interface Inside interface Outside [Capturing - 13106 bytes]
match ip any any
capture o type raw-data trace interface Outside [Capturing - 11348 bytes]
match ip any any
firepower# show cap i

47 packets captured

1: 00:03:53.657299 172.16.211.177.23725 > 192.168.73.97.443: S 1610809777:1610809777(0) win 8192
<mss 1460,nop,wscale 8,nop,nop,sackOK>
2: 00:03:53.657406 172.16.211.177.23726 > 192.168.73.97.443: S 3584167858:3584167858(0) win 8192
<mss 1460,nop,wscale 8,nop,nop,sackOK>
3: 00:03:53.803623 192.168.73.97.443 > 172.16.211.177.23726: S 2938484797:2938484797(0) ack
3584167859 win 64240 <mss 1380,nop,nop,sackOK,nop,wscale 7>
4: 00:03:53.803806 172.16.211.177.23726 > 192.168.73.97.443: . ack 2938484798 win 258
5: 00:03:53.804172 172.16.211.177.23726 > 192.168.73.97.443: P 3584167859:3584168376(517) ack
2938484798 win 258

firepower# show cap o

29 packets captured

1: 00:03:53.803638 192.168.73.97.443 > 172.16.211.177.23726: S 2938484797:2938484797(0) ack
3584167859 win 64240 <mss 1380,nop,nop,sackOK,nop,wscale 7>
2: 00:03:53.808078 192.168.73.97.443 > 172.16.211.177.23725: S 1639088682:1639088682(0) ack
1610809778 win 64240 <mss 1380,nop,nop,sackOK,nop,wscale 7>
3: 00:03:53.951717 192.168.73.97.443 > 172.16.211.177.23726: . ack 3584168376 win 501
4: 00:03:53.955776 192.168.73.97.443 > 172.16.211.177.23726: . 2938484798:2938486178(1380) ack
3584168376 win 501
5: 00:03:53.955806 192.168.73.97.443 > 172.16.211.177.23726: P 2938486178:2938487558(1380) ack
3584168376 win 501
```