Firepower Threat Defense Transparent Firewall Mode Advanced Concepts and Troubleshooting Tips

Contents

Introduction

This document describes a detailed explanation to understand the core concepts and elements from a Firepower Threat Defense (FTD) deployment in Transparent Firewall (TFW) mode. This article also provides useful tools and walkthroughs for the most common problems related to the transparent firewall architecture.

Contributed by Cesar Lopez and Edited by Yeraldin Sánchez, Cisco TAC Engineers.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco FTD transparent firewall mode knowledge

- Hot Standby Router Protocol (HSRP) concepts
- Address Resolution Protocol (ARP) and Internet Control Message Protocol (ICMP) protocols

It is highly recommended that the Firepower Configuration Guide Transparent or Routed Firewall Mode section is read to better comprehend the concepts described in this document.

Components Used

The information in this document is based on these software and hardware versions:

- Cisco Firepower 4120 FTD version 6.3.0.4
- Cisco Firepower Management Center (FMC) version 6.3.0.4
- Cisco ASR1001 IOS-XE Version 16.3.9
- Cisco Catalyst 3850 IOS-XE Version 16.9.3

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Transparent Firewall Advanced Concepts

MAC Address Table

While a firewall in routed mode relies on the routing table and ARP table to determine the egress interface and the necessary data to forward a packet to the next hop, the TFW mode uses the MAC address table to be able to determine the egress interface that is used to send a packet to its destination. The firewall looks at the destination MAC address field of the packet being processed and searches for an entry linking this address with an interface.

The MAC address table has these fields.

```
> show mac-address-table
interface mac address type Age(min) bridge-group
-----------------------------------------------------------------------------------------------
----
Outside 0050.56a5.6d52 dynamic 1 1
Inside 0000.0c9f.f014 dynamic 3 1
```

- Interface - This field holds the interface name from where this MAC address was dynamically learned or statically configured
- MAC address - MAC address record to store
- type - Method used to learn the entry. It can be dynamic or static
- Age(min) - Decremental timer in minutes displaying the time left before that entry is marked as dead. This timer only applies to dynamically learn entries
- bridge-group - Bridge group ID the interface belongs to

The packet forwarding decision is similar to a switch but there is a very important difference when it comes to a missing entry in the MAC table. In a switch, the packet is broadcasted through all interfaces except the ingress interface but in TFW, If a packet is received and there is no entry for the destination MAC address, the packet is dropped. It is discarded with Accelerated Security Path (ASP) drop code *dst-l2_lookup-fail*.

```
FTD63# show cap icmpin trace pack 1

7 packets captured

1: 00:20:22.338391 802.1Q vlan#20 P0 10.10.10.5 > 20.20.20.5 icmp: echo request
Result:
input-interface: Inside
input-status: up
input-line-status: up
Action: drop
Drop-reason: (dst-l2_lookup-fail) Dst MAC L2 Lookup Failed
```

This condition always happens for the first packet on an environment with dynamic learning enabled and without static entries for a destination if the MAC address wasn't seen before in a packet as a source MAC address.

Once the entry is added to the MAC address table, the next packet can be allowed conditioned to the firewall features enabled.

```
FTD63# show cap icmpin trace pack 2

7 packets captured

2: 00:20:27.329206 802.1Q vlan#20 P0 10.10.10.5 > 20.20.20.5 icmp: echo request
Phase: 1
Type: L2-EGRESS-IFC-LOOKUP
Subtype: Destination MAC L2 Lookup
Result: ALLOW
Config:
Additional Information:
Destination MAC lookup resulted in egress ifc Outside
```

> **Caution**: MAC Lookup is the first phase in the actions taken by the firewall. Having constant drops due to Failed L2 lookups can result in relevant packet loss and/or incomplete detection engine inspection. The affectation relies on the protocol or application capability to retransmit.

Based on the stated above, it is always preferable to have an entry learned prior to any transmission. TFW has multiple mechanisms to learn an entry.

MAC Address Table Learning Options

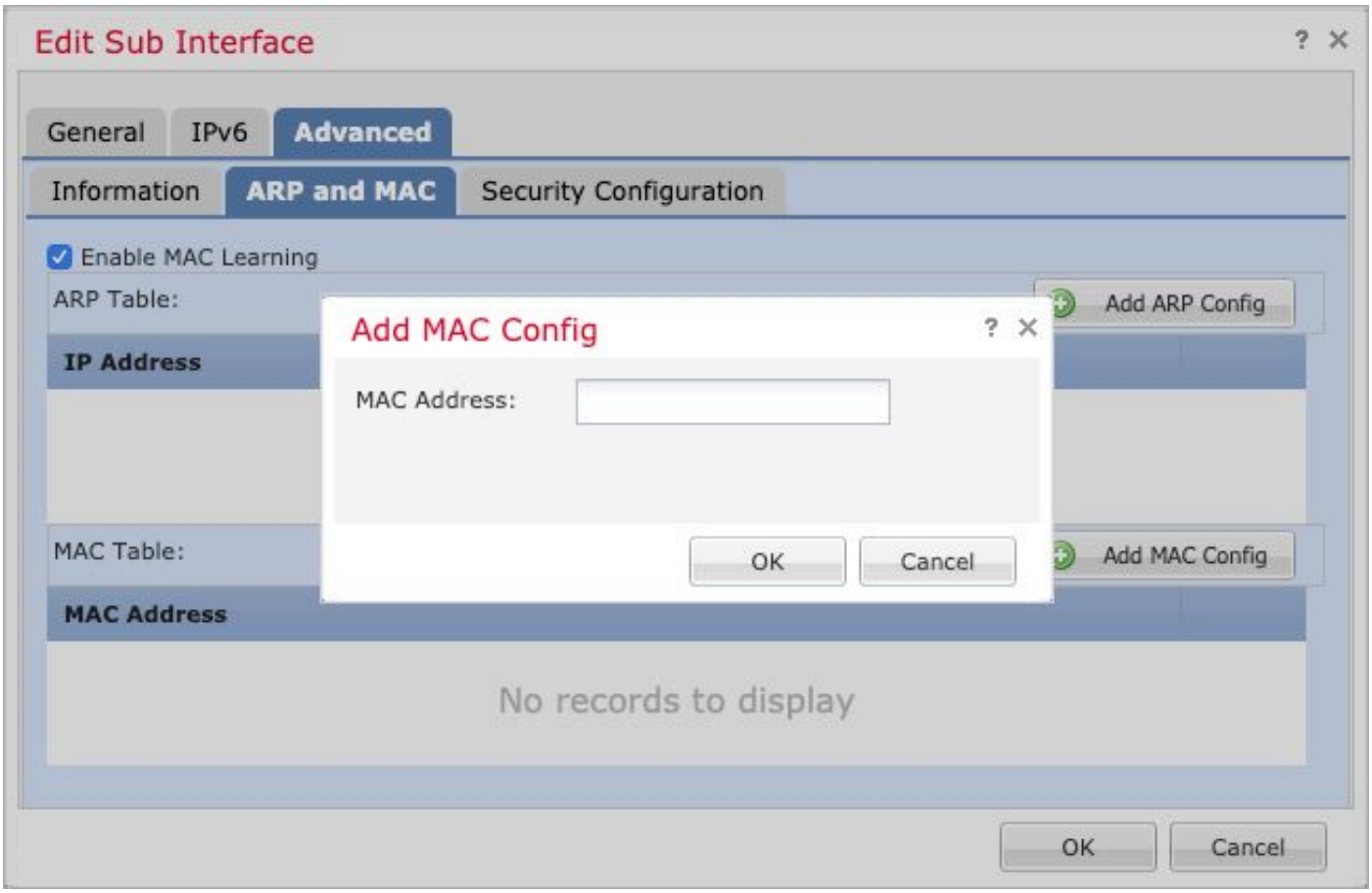
Static Entries

MAC addresses can be manually added to make the firewall always use the same interface for that specific entry. This is a valid option for entries that aren't susceptible to change. This is a common option when the static MAC is overwritten at the configuration level or by a feature at the next hop.

For example, in a scenario where the default gateway MAC address is always going to be the same on a Cisco Router as it was manually added to the configuration or if the HSRP virtual MAC address is going to remain the same.

In order to configure static entries in FTD managed by FMC, you can click on **Edit Interface / Subinterface > Advanced > ARP and MAC** and click on **Add MAC Config**. This adds an entry for the specific interface that is being edited from **Devices > Device Management >**

Interfaces section.



Dynamic Learning Based on Source MAC Address

This method is similar to what a switch does to populate the MAC address table. If a packet has a source MAC address that isn't part of the MAC table entries for the interface it was received, a new entry is added to the table.

Dynamic Learning Based on ARP Probe

If a packet arrives with a destination MAC address that is not part of the MAC table and the destination IP is part of the same network as the Bridge Virtual Interface (BVI), the TFW attempts to learn it sending an ARP request through all the bridge-group interfaces. If an ARP reply is received from any of the bridge group interfaces, it is then added to the MAC table. Note that, as it was mentioned above, while there is no reply to that ARP request, all packets are dropped with ASP code *dst-l2_lookup-fail*.

Dynamic Learning Based on ICMP Probe

If a packet arrives with a destination MAC address that is not part of the MAC table and the destination IP is NOT part of the same network as the BVI, an ICMP echo request is sent with a Time-to-Live (TTL) value equals to 1. The firewall expects an ICMP Time Exceeded message to learn the next-hop MAC address.

MAC Address Table Age Timer

The MAC address table Age timer is set to 5 minutes for each learned entry. This timeout value has two different stages.

Age Timeout First Stage

During the first 3 minutes, the MAC entry Age value isn't refreshed unless an ARP reply packet passing through the firewall with the source MAC address equals to an entry in the MAC address table. This condition excludes the ARP replies destined to the Bridge Group IP addresses. This means that any other packet that is not a through-the-box ARP reply is ignored during the first 3 minutes.

In this example, there is a PC with an IP address of 10.10.10.5 sending a ping to 10.20.20.5. The gateway IP address for 10.20.20.5 is 10.20.20.3 with MAC address 0000.0c9f.f014.

The destination PC creates an ARP update every 25 seconds causing constant ARP packets to go through the firewall.

```
FTD63# show mac-address-table
interface mac address type Age(min) bridge-group
-----------------------------------------------------------------------------------------------
----
Inside 00fc.baf3.d680 dynamic 3 1
Outside 0050.56a5.6d52 dynamic 5 1
Inside 0000.0c9f.f014 dynamic 5 1
Outside 40a6.e833.2a05 dynamic 4 1
```

A packet capture filtering ARP packets is used to match these packets.

```
> show capture

capture arp type raw-data ethernet-type arp interface Inside [Capturing - 1120 bytes]

>show capture arp

12 packets captured

1: 23:04:52.142524 802.1Q vlan#20 P0 arp who-has 10.20.20.3 (0:0:c:9f:f0:14) tell 10.20.20.5
2: 23:04:52.142952 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
3: 23:04:52.145057 802.1Q vlan#20 P0 arp who-has 10.20.20.3 (0:0:c:9f:f0:14) tell 10.20.20.5
4: 23:04:52.145347 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
5: 23:05:16.644574 802.1Q vlan#20 P0 arp who-has 10.20.20.3 (0:0:c:9f:f0:14) tell 10.20.20.5
6: 23:05:16.644940 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
7: 23:05:16.646756 802.1Q vlan#20 P0 arp who-has 10.20.20.3 (0:0:c:9f:f0:14) tell 10.20.20.5
8: 23:05:16.647015 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
9: 23:05:41.146614 802.1Q vlan#20 P0 arp who-has 10.20.20.3 (0:0:c:9f:f0:14) tell 10.20.20.5
10: 23:05:41.146980 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
11: 23:05:41.148734 802.1Q vlan#20 P0 arp who-has 10.20.20.3 (0:0:c:9f:f0:14) tell 10.20.20.5
12: 23:05:41.149009 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
```

The entry for 000.0c9f.4014 stays at 5 and never goes below that number.

```
> show mac-address-table
interface mac address type Age(min) bridge-group
-----------------------------------------------------------------------------------------------
----
Inside 00fc.baf3.d680 dynamic 3 1
Outside 0050.56a5.6d52 dynamic 5 1
```

```
Inside 0000.0c9f.f014 dynamic 5 1
Outside 40a6.e833.2a05 dynamic 4 1
```

Age Timeout Second Stage

During the last 2 minutes, the entry falls into a time period where the address is considered aged out.

```
> show mac-address-table
interface mac address type Age(min) bridge-group
-----------------------------------------------------------------------------------------------
----
Inside 00fc.baf3.d680 dynamic 5 1
Outside 0050.56a5.6d52 dynamic 3 1
Inside 0000.0c9f.f014 dynamic 2 1
Outside 40a6.e833.2a05 dynamic 3 1
```

The entry isn't removed yet and if any packet with the Source MAC address matching the table entry, including to-the-box packets, is detected, the Age entry is refreshed back to 5 minutes.

In this example, a ping is sent within this 2 minutes to force the firewall to send its own ARP packet.

```
> ping 10.20.20.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.20.20.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

The MAC address entry is set back to 5 minutes.

```
> show mac-address-table
interface mac address type Age(min) bridge-group
-----------------------------------------------------------------------------------------------
----
Inside 00fc.baf3.d680 dynamic 4 1
Outside 0050.56a5.6d52 dynamic 2 1
Inside 0000.0c9f.f014 dynamic 5 1
Outside 40a6.e833.2a05 dynamic 5 1
```

ARP table

First, it is essential to understand that the MAC Address table is entirely independent of the ARP table. While the ARP packets sent by the firewall to refresh an ARP entry can, at the same time refresh the MAC address table, these refresh processes are separate task and each has its own timeouts and conditions.

Even if the ARP table isn't used to determine the egress next-hop as in routed mode, it is important to understand the effect of the ARP packets generated and destined to the firewall identity IPs can have in a transparent deployment.

The ARP entries are used for management purposes and are only added to the table if a management feature or task requires it. As an example of a management task, if a Bridge Group has an IP address, this IP can be used to ping the destination.

```
> show ip
Management-only Interface: Ethernet1/4
System IP Address:
no ip address
Current IP Address:
no ip address
Group : 1
Management System IP Address:
ip address 10.20.20.4 255.255.255.0
Management Current IP Address:
ip address 10.20.20.4 255.255.255.0
```

If the destination is in the same subnet as the Bridge Group IP, it forces an ARP request and if a valid ARP reply is received, the IP/MAC entry is stored in the ARP table.

```
> show arp
Inside 10.20.20.3 0000.0c9f.f014 6
```

Unlike the MAC address table, the timer accompanying the interface/IP address/MAC address triplet is an increasing value.

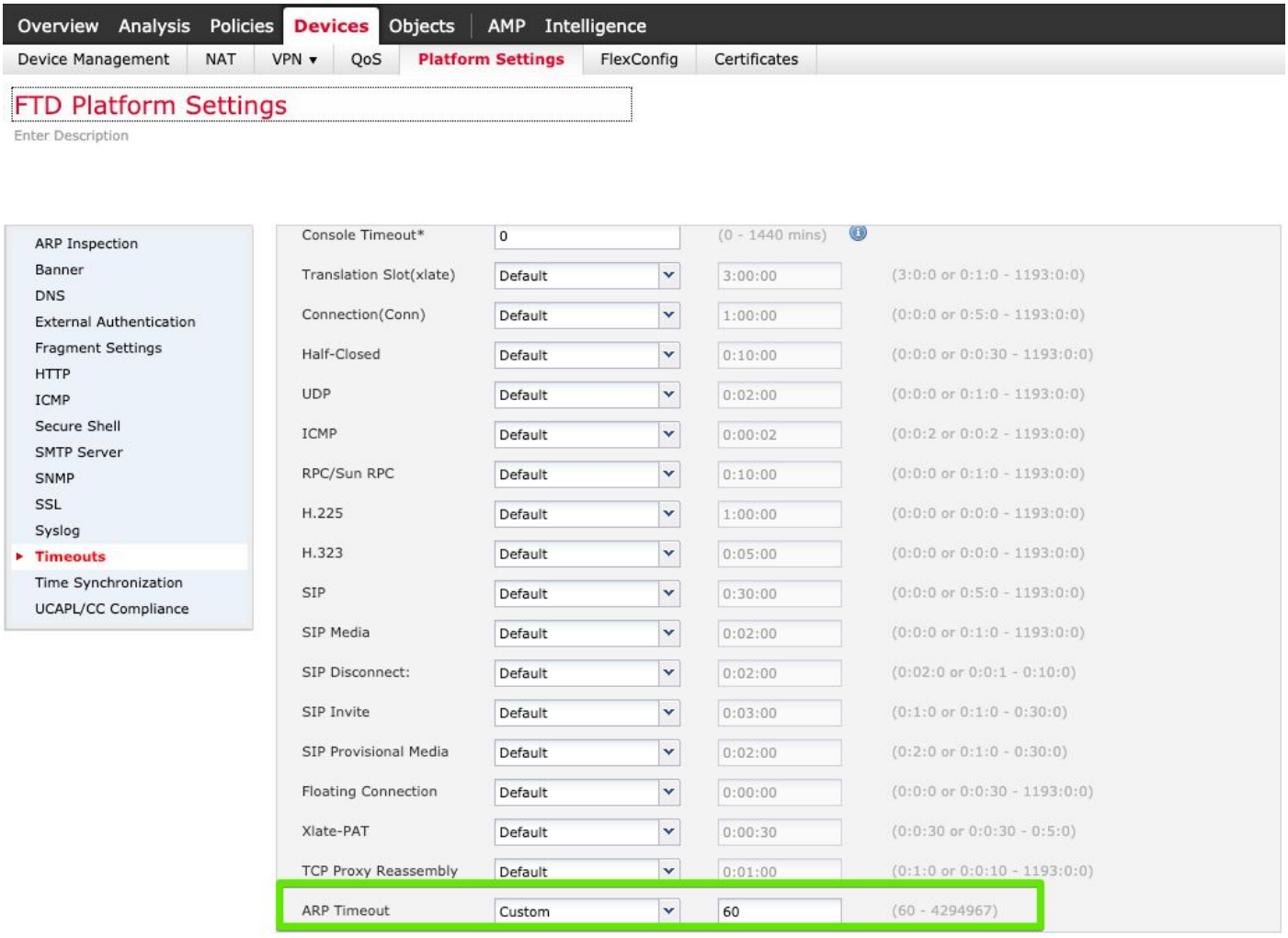
```
> show arp
Inside 10.20.20.3 0000.0c9f.f014 1
>show arp
Inside 10.20.20.3 0000.0c9f.f014 2
>show arp
Inside 10.20.20.3 0000.0c9f.f014 3
>show arp
Inside 10.20.20.3 0000.0c9f.f014 4
```

When the timer reaches an *n - 30* value where *n* is the ARP configured timeout (with a default of 14400 seconds), the firewall sends an ARP request to refresh the entry. If a valid ARP reply is received, the entry is held and the timer goes back to 0.

In this example, the ARP timeout was reduced to 60 seconds.

```
> show running-config arp
arp timeout 60
arp rate-limit 32768
```

This timeout is available to be configured at **Devices > Platform Settings > Timeouts** tab in FMC, as shown in the image.



Since the timeout is 60 seconds, an ARP request is sent every 30 seconds (60 - 30 = 30).

```
> show capture arp

8 packets captured

1: 21:18:16.779729 802.1Q vlan#20 P0 arp who-has 10.20.20.3 tell 10.20.20.4
2: 21:18:16.780111 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
3: 21:18:46.779744 802.1Q vlan#20 P0 arp who-has 10.20.20.3 tell 10.20.20.4
4: 21:18:46.780126 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
5: 21:19:16.779744 802.1Q vlan#20 P0 arp who-has 10.20.20.3 tell 10.20.20.4
6: 21:19:16.780111 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
7: 21:19:46.779729 802.1Q vlan#20 P0 arp who-has 10.20.20.3 tell 10.20.20.4
8: 21:19:46.780126 802.1Q vlan#20 P0 arp reply 10.20.20.3 is-at 0:0:c:9f:f0:14
```

The ARP entry is then refreshed every 30 seconds.

```
> show arp
Inside 10.20.20.3 0000.0c9f.f014 29
>show arp
Inside 10.20.20.3 0000.0c9f.f014 0
```

Troubleshoot Tips

Traffic Direction

One of the most difficult things to track down on a TFW is the traffic flow direction. Understanding

how the traffic flows helps to ensure the firewall is properly forwarding the packets to the destination.

Determining the right ingress and egress interface is an easier task on Routed mode as there are multiple indicators of the firewall involvement such as source and destination MAC addresses modification and Time-To-Live (TTL) value reduction from one interface to the other.

These differences aren't available on a TFW setup. The packet coming through the ingress interface looks the same as when it leaves the firewall in most of the cases.

Specific problems such as MAC flaps in the network or traffic loops could be harder to track without knowing where the packet entered and when it left the firewall.

To help differentiate ingress from egress packets, the trace keyword can be used in packet captures.

```
capture in interface inside buffer 33554432 trace trace-count 1000 match tcp host 10.10.220.42
host 10.10.241.225
capture out interface outside buffer 33554432 trace trace-count 1000 match tcp host 10.10.220.42
host 10.10.241.225
```

buffer - Increases the capture buffer in bytes. 33554432 is the maximum available value. In models such as 5500-X, Firepower appliances, or virtual machines, it is safe to use this size value as long as there are not dozens of captures already configured.

trace - Enables trace option for the specified captured.

trace-count - Allows a higher number of traces. 1000 is the maximum allowed and 128 is the default. This is also safe following the same recommendation as to the buffer size option.

> **Tip**: If you forget to add one of the options, you can add it without having to write the whole capture again by referencing the capture name and the option. However, the new option affects only the newly captured packets so a **clear capture *capname*** must be used to have the new effect since packet number 1. Example: **capture in trace**

Once packets have been captured, the command **show capture *cap_name* trace** displays the first 1000 (if the trace number was increased) traces of the ingressed packets.

```
FTD63# show capture out trace
1: 16:34:56.940960 802.1Q vlan#7 P0 10.10.241.225 > 10.10.220.38 icmp: time exceeded in-transit
Result: input-interface: outside input-status: up input-line-status: up Action: drop Drop-
reason: (dst-l2_lookup-fail) Dst MAC L2 Lookup Failed 2: 16:34:57.143959 802.1Q vlan#7 P0
10.10.220.42 > 10.10.241.225 icmp: echo request 3: 16:34:57.146476 802.1Q vlan#7 P0
10.10.241.225 > 10.10.220.42 icmp: echo reply Result: input-interface: outside input-status: up
input-line-status: up Action: drop Drop-reason: (dst-l2_lookup-fail) Dst MAC L2 Lookup Failed
```

This output is an example of the outside interface packet capture traces. This means that packet numbers 1 and 3 ingressed the outside interface and packet number 2 egressed the interface.

Additional information can be found in this trace such as the Action taken for that packet and the Drop-reason in case that the packet is dropped.

For longer traces and if you want to focus on a single packet, the command **show**

capture *cap_name* trace packet-number *packet_number*can be used to display the trace for that specific packet.

This is an example of an allowed packet number 10.

```
FTD63# show capture in detail trace packet-number 10

10: 20:55:31.118218 0000.0c9f.f014 0100.5e00.0066 0x8100 Length: 98 802.1Q vlan#20 P0
10.20.20.1.1985 > 224.0.0.102.1985: [udp sum ok] udp 52 [tos 0xc0] [ttl 1] (id 0) Phase: 1 Type:
L2-EGRESS-IFC-LOOKUP Subtype: Destination MAC L2 Lookup Result: ALLOW Config: Additional
Information: Destination MAC lookup resulted in egress ifc Outside Phase: 2 Type: CAPTURE
Subtype: Result: ALLOW Config: Additional Information: MAC Access list Phase: 3 Type: ACCESS-
LIST Subtype: Result: ALLOW Config: Implicit Rule Additional Information: MAC Access list Phase:
4 Type: FLOW-LOOKUP Subtype: Result: ALLOW Config: Additional Information: Found flow with id
2562905, using existing flow Phase: 5 Type: SNORT Subtype: Result: ALLOW Config: Additional
Information: Snort Verdict: (fast-forward) fast forward this flow Phase: 6 Type: CAPTURE
Subtype: Result: ALLOW Config: Additional Information: MAC Access list Result: input-interface:
Inside input-status: up input-line-status: up Action: allow
```

MAC Tracking

TFW makes all its forwarding decisions based on MAC addresses. During the traffic flow analysis, it is essential to ensure that the MAC addresses used as source and destination on each packet are correct based on the network topology.

The packet capture feature allows you to display the MAC addresses used using the ***detail*** option from the **show capture** command.

```
FTD63# show cap i detail

98 packets captured

   1: 20:55:06.938473 0000.0c9f.f014 0100.5e00.0066 0x8100 Length: 98
      802.1Q vlan#20 P0 10.20.20.1.1985 > 224.0.0.102.1985:  [udp sum ok] udp 52 [tos 0xc0]
[ttl 1] (id 0)
   2: 20:55:09.805561 0000.0c9f.f014 0100.5e00.0066 0x8100 Length: 98
      802.1Q vlan#20 P0 10.20.20.1.1985 > 224.0.0.102.1985:  [udp sum ok] udp 52 [tos 0xc0]
[ttl 1] (id 0)
```

Once you have located and interesting MAC address that requires specific tracking, the capture filters allow you to match it.

```
FTD63# capture in type raw-data trace interface inside match mac 0000.0c9f.f014 ffff.ffff.ffff
any

FTD63# show capture

capture in type raw-data trace interface inside [Capturing - 114 bytes] match mac 0000.0c9f.f014
ffff.ffff.ffff any

FTD63# show cap in detail 98 packets captured 1: 20:55:06.938473 0000.0c9f.f014 0100.5e00.0066
0x8100 Length: 98 802.1Q vlan#20 P0 10.20.20.1.1985 > 224.0.0.102.1985: [udp sum ok] udp 52 [tos
0xc0] [ttl 1] (id 0) 2: 20:55:09.805561 0000.0c9f.f014 0100.5e00.0066 0x8100 Length: 98 802.1Q
vlan#20 P0 10.20.20.1.1985 > 224.0.0.102.1985: [udp sum ok] udp 52 [tos 0xc0] [ttl 1] (id 0)
```

This filter is extremely useful when there are traces of MAC flaps and you want to find the culprit(s).

Mac-address-table Debug

MAC address table debug can be enabled to review each phase. The information provided by this debug helps understand when a MAC address is learned, refreshed, and removed from the table.

This section shows examples of each phase and how to read this information. In order to enable debug commands on FTD, you must access the Diagnostic CLI.

> **Warning**: Debugs can consume relevant resources if the network is too busy. It is recommended to use them in controlled environments or during low peak hours. It is recommended to send these debugs to a Syslog server if these are too verbose.

```
> system support diagnostic-cli
Attaching to Diagnostic CLI ... Press 'Ctrl+a then d' to detach.
Type help or '?' for a list of available commands.

FTD63# debug mac-address-table
debug mac-address-table enabled at level 1
```

Step 1. MAC address is learned. When an entry is not found in the MAC table already, this address is added to the table. The debug message informs the address and the interface where it was received.

```
FTD63# ping 10.20.20.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.20.20.3, timeout is 2 seconds:
add_l2fwd_entry: Going to add MAC 0000.0c9f.f014.
add_l2fwd_entry: Added MAC 0000.0c9f.f014 into bridge table thru Inside.
set_l2: Found MAC entry 0000.0c9f.f014 on Inside.
!add_l2fwd_entry: Going to add MAC 00fc.baf3.d680.
add_l2fwd_entry: Added MAC 00fc.baf3.d680 into bridge table thru Inside.
!!!!
```

If the MAC is learned through the ICMP method, the next message is shown. The entry enters the first stage of the timeout cycle where it doesn't refresh its timer based on the conditions listed in the MAC address Table Age Timer.

```
learn_from_icmp_error: Learning from icmp error.
```

Step 2. If an entry is already known, the debug informs about it. The debug also displays clustering messages which are irrelevant in standalone or HA setups.

```
set_l2: Found MAC entry 0000.0c9f.f014 on Inside.
l2fwd_refresh: Sending clustering LU to refresh MAC 0000.0c9f.f014.
l2fwd_refresh: Failed to send clustering LU to refresh MAC 0000.0c9f.f014
```

Step 3. Once the entry has reached the second stage (2 minutes before the absolute timeout).

```
FTD63# show mac-add
interface                  mac  address          type       Age(min)    bridge-group
-------------------------------------------------------------------------------------------
----
Inside                     00fc.baf3.d700        dynamic      3           1
```

```
Outside                   0050.56a5.6d52        dynamic      4           1
Inside                    0000.0c9f.f014        dynamic      2           1
Outside                   40a6.e833.2a05        dynamic      3           1

FTD63# l2fwd_clean:MAC 0000.0c9f.f014 entry aged out.
l2fwd_timeout:MAC entry timed out
```

Step 4. The firewall now expects new packets sourced with that address to refresh the table. If there are no more packets using that entry during those 2 minutes, the address is removed.

```
FTD63# show mac-address-table
interface mac address type Age(min) bridge-group
-----------------------------------------------------------------------------------------------
----
Inside 0000.0c9f.f014 dynamic 1 1
Outside 40a6.e833.2a05 dynamic 3 1
FTD63# l2fwd_clean:Deleting MAC 0000.0c9f.f014 entry due to timeout.
delete_l2_fromPC: Deleting MAC 0000.0c9f.f014 due to freeing up of entry
l2fwd_clean:Deleted MAC 0000.0c9f.f014 from NP.
```

Related Information

- **Firepower Management Center Guide, Version 6.3 - Chapter 3: Transparent or Routed Firewall Mode for Firepower Threat Defense**
- **Technical Support & Documentation - Cisco Systems**