MODULE 1 Analyzing Network Protocols With Wireshark

Packet that Matters:

Best way to begin capturing network traffic is collecting our analysis on a span mirror port on a switch near the client.

This will help to keep the traffic flowing and to avoid the traffic being captured by the switch hence making it smaller and easier to analyze.

Another way to create a smaller haystack is to filter the captured traffic to only the traffic that is relevant to the analysis.

We create custom columns, protocols, and custom filters to help us analyze the traffic.

Also saving protocol filterings as button for the quick access to those filters.

Core Protocols

Under the application data there are :

- UDP
- TCP
- IPV4
- IPV6
- ARP
- ICMP
- DNS
- TLS

Custom Profiles

Right click on down right on profile and select "New" and then create the custom profile.

After creating a new profile: go to edit and preferences to start customizing the profile.

Right of the bat: adding Delta time to the profile as title `delta` and type `Delta time displayed` to show the time difference between the packets.

Another way is to right click on any value down and click `add as column` to add the value as a column.

Creating and Saving buttons.

There are some filters that you keep using but you dont want to keep writing them. For example `tcp.flags.syn==1` which is the syn=1 flag where the communication starts.

in order to add it as button, I write this on the filter box and click on `+` to add it as a button.

it is also important to colorize the packets in order to create the visibility. view==>colouring rules, then write the filter you want to filter (like regex) and then do not forget to enable it. Also change the importance level by dragging up and down in the list.

PROTOCOLS IN DETAIL

1- ARP

NOTE THAT ARP does not resolve IPV6 addresses. Meaning IPv6 does not use ARP. IPV6 uses NDP(Neighbor Discovery Protocol) to resolve the addresses which replaces ARP.

Address Resolution Protocol (ARP) is a protocol used to resolve the MAC address of a host. meaning, it bridges the gap between layer 2 and layer 3.

ARP ne ise yarar nasil calisir?

Bir packet gonderecegin zaman, header olusturmak icin destination IP ve MAC address ihtiyacin var. sen baslangicta kendi IP ve mac adresini biliyorsun, bir de serverinkini.

Target'in MAC adresini bulmak icin ARP kullanilir. ilk once local arp cache'e bakar. eger varsa direkt cevap verir. eger yoksa `arp request` gonderir. Networke bunu broadcast olarak gonderir.

Bu broadcast domain icindeki devices will take this up, will check the IP address that is being resolved and will build and send a reply with its own mac address as the source mac address.

Yani aslida soyle:

18 numarali bilet kimde ogrenmek istiyorsun. Elindeki listede varsa zaten sorun yok, yoksa ortaliga bagiriyorsun `18 numarali bilet kimdee` yani `what is the mac address for this IP address`?

Herkes biletine bakip `benimki x`, `benimki Y` diye cevap veriyor.

Sen de sonunda ogrenmis oluyorsun ve header'i yaratip gonderiyorsun.

==> Arp requests are broadcasted to the network but responses are unicasted to the sender.

You check the ARP protocol if you see:

- Problems connecting to an application
- intermittent connectivity.
- Unicast flooding.

If the destination is visible (like Apple or Google,) it means that the destionation was on the Arp Cache already so it is not broadcasted but unicasted.

Hands on Demo:

**Lets disect a ARP request: **

```
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: BelkinIn_9d:02:73 (94:10:3e:9d:02:73)
    Sender IP address: 192.168.10.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.10.108
```

- Hardware type = Ethernet
- Protocol type = IPv4 Meaanign trying to resolve a mac address to IP address.
- Opcode = 1 meaning request
- Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00) so this is the information that is missing and being looked for. 00:00:00... means that the target mac address is not known.
- Target IP address: 192.168.10.108 is the IP address that is being looked for.

Now lets look at the ARP Response to this request:

```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Apple_e7:ce:6d (a4:5e:60:e7:ce:6d)
    Sender IP address: 192.168.10.108
    Target MAC address: BelkinIn_9d:02:73 (94:10:3e:9d:02:73)
    Target IP address: 192.168.10.1
```

- Opcode = 2 meaning reply
- Sender MAC address: Apple_e7:ce:6d (a4:5e:60:e7:ce:6d) so this is the mac address of the sender which was missing.
- Sender IP address: 192.168.10.108

As you can see, sender and target IP addresses match, so the ARP reply is valid and MAC address is resolved.

In some cases we see some weird ARP requests that sends requests for entire network. like couple hundred or maybe more limited or less limited numbers of ARP requests.

This `could be` and indicator to an attack.

The first thing to do is to check the IP origin of the ARP request. For that end, I will create a ARP profile in the wireshark.

as a new coulumn, I will add the `opcode` field which will filter requests and responds. that would be interesting in this case to see if there is any active subnets and actualy any response.

Here is the filter : `arp.opcode==2` This will show all the packets that responded to the requests.

Another very interesting filter is `arp.isgratuitous` which means that we are looking for arps that are gratuitous either as requests or replies. Grattuitous means that the sender and target mac addresses are the same. Meaning sending its own IP and Mac address which means advertising itself.

Once we know that all is fine, we can remove any protocol that is not needed in the trace. For a protocol to be removed we use `!{protocolName}` like `!arp`.

2- IPv4, IPv6 ve ICMP

Unlike ethernet, IP is a end to end protocol not a point to point protocol.

`192.168.1.8` is an IP address, `255.255.255.0` is a subnet.

IP header holds the information about the packet like version, header length, `DSCP(Differentiated Services Code Point)`, ECN(Explicit Congestion Notification), total length and so forth. Many of the times we will be dealing with `DSCP` part of it to troubleshoot where markings for the packet prioritizing is made.

Another important part is total length whici is the total amount of encapsulated packet including the header itself.

Next is the `identification` field which is used to identify the packet, it is either randomized or sequential which is used to `uniquely id` a packet from a station.

Helps figure out whether a packet is duplicated or not, and also help track application traffic behind a load balancer.

`flags` field helps to understand whether the packet is a fragment or not. or fragmentation is allowed or not.

`Time to Live` layer helps to see how many routers/or layer 3 switches a packet has hopped through on its way to destination.

`Protocol` field shows which protocol is coming next in the data payload. It could be TCP, ICMP, or other.

IP Fragmentation

Sometimes there is so much data that it is not possible to send it in a single packet.

Lets assume you are sending a data of 1500 bytes. But VPN tunnel has `MTU` 1400 bytes and rest is reserved for encapsulation. As long as the data is less than MTU, it can be sent in a single packet but if it is greater than that, `flags` field will be checked. If the flags are set to `MF` then it means that the packet is fragmented and needs to be sent in multiple packets(called fragments). Each fragment then holds in their header field on how to reassamble the packet.

Sometimes, like in encrypted traffic, the packet does not want to be intercepted or dissected.

If packet size is big and MTU is lower than that, and also `do not fragment` is set, then router will send an `ICMP` error message to the sender saying it cannot pass the packet.

TTL

For example when you ping to someplace, it gives you the TTL number.(like 51)

TTL is not a function of time, it is a decrementing counter!!. As the packet travels through the network, each router decrements the counter by 1. If the number is reduced to 0, meaning the packet has reached the destination, it will be dropped and ICMP error message will be sent back to the sender.

This works in both directions in the same manner. This way, we can estimate how many routers are there in the network.

These days TTL starts either at 255(cisco/solaris), or 128(windows), or 64(linux)

Understanding IP TTL

Questions based on a Pcap file(`IP TTL`)

- 1. How many unique IP stations are transmitting in this trace file?

go to statistics==>endpoints tab and do not count them manually!

- 2. How many unique IP conversations are there in this trace file?

go to statistics ==> converstaions and do not count them manually!

- 3. What conversation is the busiest? (By bytes)

in statistics ==> conversations, sort them by bytes. (104.17.208.240=192.168.10.108) was the busiest.

- 4.Set a filter for the conversations including address 104.19.162.127. How many packets match that filter?

set the filter `ip.addr==104.19.162.127` and on the lower right side is the visible.

- 5 What side of the conversation was this trace file captured on? Client or server? How can you tell?

we loook at the `info` column, `source` and `destination` columns.

they generally give an idea based on the ports (lets say destination port is 80, then it is a client side conversation)

A robust check would be to check the first packet, and take a look at the IP TTL. in my case, it is 64. **The reason for picking the FIRST packet is to capture the iunitial value of TTL becasue a random packet with TTL 64 or say 50 could belong to any possible TTLs of 255,128,64 but the first packet's ttl will give the initial counter number.**

Then I check a packet on the opposite direction(source,destionation places are changed.) then I see TTL is 51. This tells me that the `SERVER is 13 HOPS AWAY`!! `

because initial counter - current number = 13.

I know this is captured from client side.

REason 1: I dont have routing packets on the outbound direction. REason 2 : I do see coming back packets from the server.

Another way to know is to look at the delta time.

Inbound and outbound packets for the same IP alinir. aralarindaki delta time'a bakilir ki bu bizim ornekte 0.46ms. bu kisa bir sure. Sonra 2. packet'in infosuna bakilir. infoda `[SYN, ACK]` yaziyor. bu da clientta oldugumuzu gosteritor.

- 6. Is there any prioritization in traffic coming from the server? What priority marking is used?

I pick a packet coming from the server, open the `Differentiated Services Codepoint` and check if any marking. in my case, there was the marking `Assured Forwarding 11` so it says the packet is not going to be intercepted and dropped.

- 7. What IP flags are set on traffic coming from the server?

again, pick a packet coming from the server and open the `Flags` section. in my case it is `0x4000` which means `Don't Fragment`

- 8 Is the client using incrementing IP Identification numbers? or is it randomizing them? Pick 2 client packets coming one after another and check the `identification` in both of them. if the numbers are consecutive, then it is incrementing. if not, then it is random.

IP Fragmentation in action

I can use a filter and filter out all the packets with size 1500 and above. once filtered, in the infor section of the packet wireshark will say `Fragmented IP protocol`. Once I open the packet and check the **DSCP** field, it is set to 0000 00. More importantly, in the `Flags` segment, the flag set is `more fragments` meaning although the size of this packet is 1500, more packets are to follow.

`Fragment offset` field gives me the place of the packet in the total packet. (in my case was 0 meaning the first fragment.)

This is all possible due to `do not fragment` flag.

ICMP

Internet Message Control Protocol is a messaging suite used to send error messages to the sender. Used by both endpoints and infrastructire.

- outages
- network problems,
- routing problems
- port unreachable/ unavailable and more.

You send a SYN packet to server. Gateway is problematic and does not have a route to destination network. So it sends an ICMP error message (`Destination Unreachable`) to the sender.

Another way ICMP is used is during fragmenting a packet. Lets say ytou are sending a packet of 1500 bytes BUT flag says `Do not fragment` however MTU on the router is 1400 so it NEEDs to dissect the packet but it CANT. so it will reply with an ICMP error message(`Destionation Unreachable Fragmentation Needed`).

An ICMP packet :

```
Internet Control Message Protocol
   **Type: 8 (Echo (ping) request)**
   Code: 0
   Checksum: 0xe275 [correct]
   [Checksum Status: Good]
   Identifier (BE): 34209 (0x85a1)
   Identifier (LE): 41349 (0xa185)
   Sequence number (BE): 0 (0x0000)
   Sequence number (LE): 0 (0x0000)
   [No response seen]
   Timestamp from icmp data: Oct 28, 2019 22:00:31.237036000 +03
   [Timestamp from icmp data (relative): 0.000056000 seconds]
   Data (1592 bytes)
```

Some ICMP types:

- 0: Echo reply
- 3: Destination unreachable
- 5: Redirect
- 8: Echo request
- 11: Time exceeded(TTL Exceeded)
- 12: Parameter problem
- 30: Traceroute

Undear each type are codes that correspond(for gfiving more detailed error.)

LEts say the types is 3, `Destination Unreachable`. These are the possible codes:

- 0: Network unreachable
- 1: Host unreachable
- 2: Protocol unreachable
- 3: Port unreachable
- 4: Fragmentation required/ DF set
- 9: Network administratively prohibited

Let's go over some ICMP questions:

- 1 How many ICMP packets are in this trace file? go to display filter, write `icmp` and check the number down the right side.
- 2 What is the Type of ICMP message?

pick the icmp packet and under `ICMP`, you see the type and code.

- 3 What is the code value?

CODE:3 ==> so it is 3 and Port Unreachable.

So that packet went aalll the way through the network, hit the destionation but the port was not open.

- 4 What is the source IP of the sending station of these packets?

in the Source segment of the packet, it is available.

- 5 Why is this(with code 3) ICMP message being sent? Is something broken? If so, what?

we go above the first error opcoded ICMP packet and see the network traffic. in my case, there were 2 requests were made in sub-milisecond to same website on different IP addresses. one was successfully resolved whereas other gave the error. since the dirst one was successful, the other one was not needed anyways and was dropped.

One another way is to look ath the returned ICMP packet itself. nice thing about ICMP is that it shows the packet that triggered the error. check below:

`User Datagram Protocol, Src Port: 53, Dst Port: 39550`

so this informatiuon actually says that you TRIED to reach port 39550 but it was not open so here is the error.

- 6. Will the user experience any application problems from this behavior?

If there wasnt a subsequent successful connection, then the user would experience a problem.

IPv6

IPv4 has 4.3 billion IP addresses, 32 bit. IPv6 has 340 trillion trillion IP addresses, 128 bit.

lets take a look at a IPv6 address:

`2001:4860:4860:0000:0000:0000:0000:8888`

- 8 blocks of 4 hexadecimal values
- first 3 blocks(48 bytes) are unique network identifier (UNI)
- 4.th block is subnet identifier (SID)(ipv4 uses masks ipv6 uses identifier)
- rest of the 4 blocks are host identifier (HI). used for getting the packet into its destination

if there are multiple 0 blocks, it is by convention written as `::`

so above example would be `2001:4860:4860::8888`

The IPv6 ADdress Ranges:

Most commons are:

- Link Local Address Range: `fe80::/64` ==> local without a router

- Global Address Range : `2000::/3` ==> send `unicast` packet from one device to another
- Unique Local Address Range: `fc00::/7` ==> Router comms within an enterprise(private address space gibi)

Ipv6 is compatible with IPv4 using tunneling.

As for **IPv6 Header**:

first two header values are `Version` and `Priority/Traffic Class` which allows priorizitaion of the Ipv6 packets.

next comes `Flow Label` which works as a label for series of packets in a single flow to uniquely ID them.

Let's go over some IPv6 packet:

```
Internet Protocol Version 6, Src: 2001:1:2:3::1, Dst: 2001:1:2:3::2
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP:
CS0, ECN: Not-ECT)
    .... .... .... 0000 0000 0000 0000 0000 = Flow Label: 0x00000
    Payload Length: 60
    Next Header: ICMPv6 (58)
    Hop Limit: 64
    Source: 2001:1:2:3::1
    Destination: 2001:1:2:3::2
```

version: 6 ==> ipv6

next we can see DSCP or Traffic Class. (in this packet there is not one but normally it is expected)

Next is the `Flow Label` which is used to uniquely identify the packets in a flow. IPv6 automatically adds this value to the IPv6 header.

`Hop Limit` ==> how many times the packet can be forwarded. in this case is this is like TTL in IPv4. and last 2 are source and destination IPs with :: markation.

So as packet perspective, it is not very daunting.

3- UDP, DHCP, DNS

UDP

`User Datagram Protocol` does not require a handshake or establishing a connection. Due to its low overhead, it is generally used for `Time Sensitive applications` **that can handle small amount of data loss**. So time critical but NOT safety critical maybe. It is a very simple protocol without many of the segments that TCP has.

A UDP header consists of the following fields:

- Source Port

- Destination Port
- Length
- Checksum .

Let's go over some UDP packet and answer some questions:(Based on *Analyzing UDP pcap*)

- 1 . What Applications are using UDP in this trace file?

We can see on the `Protocol` column, DNS and SNMP are using the UDP protocol.

- 2 What UDP port number does the DNS server use?

Opening a DNS application and checking the UDP packet header will show the source and destination ports. Generally, UDP lives on the port 53. This pcap file is not exception to that.

Source: 2085

Destination: 53

- 3 Why do we see the ICMP message in packet 5?

ICMP packet's type is 3 and code is 3 which is `Destination Unreachable, Port Unreachable`.

So there has been a packet that was sent to a port that was not open.

The reason was that, there was a SNMP request made to an IP address on port 161 and that returned no port open as ICMP packet. How Did I know this happened because of the ICMP packet?

Because ICMP's source-destination should be used revereselu in the problematic packet. which was ICMP. then to further check, I checked the destination ports in both packets which were 161 and 161. meaning they were same so ICMP was a response to SNMP packet.

==> **Tip: port 161 is generally used for SNMP anyways.**

That's all for UDP for now.

DHCP

Client IP address is assigned by the DHCP server.

Lets say there are 2 hosts with unique MAC addresses , connected to a switch. In order to be able to communicate, they neeed unique IP addresses. DHCP helps assigning IP addresses dynamically from a central pool. DHCP helps these endpoints to discover servers that are offering addresing. So client , using `DHCP discover` asks **Who is the DHCP server?**

DHCP consists of 4 packets:

- 1) DHCP discover: Client *broadcasts* this to the network to find a DHCP server.
- 2) DHCP offer: Server sends this broadcast to the client to tell it where to go.

There could be more than one DHCP server in the network so can get multiple offers, client then decides which one to accept.

- 3) DHCP request: Client sends broadcast request to decided DHcP server to get an IP address.
- 4) DHCP ack: Server sends this to the client to tell it that it has an IP address.

That is the basic of DHCP and how a client gets a new IP address.

There are some extra steps when analyzing the DHCP packets.

After getting assigned with an IP address, the client wants to verify that it is the only one that is with that IP address, hence sends an ARP request (gratuitous) to the server, if any other client says `I have that IP address too`, then it informs the server saying I need anoter IP address.

for this end, the client sends an **DHCP Decline** to the server.

Then server marks this as `assigned alredy` and assigns the next available IP address in the dynamic pool.

during the DHCP process, the server will indicate to the client the *length of the lease* and the *time remaining* for the lease. meaning this is the timeframe this IP is valid. After that, the IP will be released to the dynamic pool for reassignment.

anytime during the `lease` the client can send a **DHCP renew** to the server to keep the IP address alive as long as it is available or server tells the client when to request `renew`

Lets go over some questions to better understand DHCP (analyzing DHCP pcap):

- 1 In the DHCP discover, what options is the client requesting?

Beware that the source IP address of the client that sends `dhcp discover` is 0.0.0.0 since it does not have a valid IP address yet.

in the initial discover packet we have the following options:

- DHCP message type: 1 (Discover)
- Client Identifier: that is the MAC address of the client.
- Host name: the name of the client.
- Parameter Request List: the list of options the client wants to receive. in my case it was `Subnet Mask`, `Router`, `Domain Name Server`.
- End of options: 0xFF.
- 2. Is this a broadcast packet?

When I look at that layer 2 of the packet `Ethernet II, Src: NetAlly_a1:17:9f (00:c0:17:a1:17:9f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)` it is clearly said that the destination is Broadcast.

- 3.What is the function of the ARPs in packets 2-4?

ARP packets 2-4 are exactly the same packets broadcasted 3 times. asking whether someone has the IP address that the client is trying to get. We assume that not response was given to this packet.(because in packet 5, DHCP will offer this IP to the client.)

BEWARE THAT WE ONLY ASSUME THAT ARP RESPONSE WAS NOT GIVEN. WHY DO WE ONLY ASSUME? BECUSE ARP REQUESTS ARE BROADCASTED SO WE CAN SEE THEM BUT RESPONSES ARE UNICASTED SO THERE IS A POSSIBILITY WE DONT TAP ON THEM

- 4 What is the server host name of the DHCP server?

in the DHCP offer packet, the server host name is `ecosystem.home.cisco.com`.

- 5 The server goes above and beyond. What options are offered to the client that did not appear in the discover packet?

One thing interesting is, in the offer packet, there are some options that are not mentioned in the discover packet. these are:

- Server Identifier: the IP address of the server.
- IP address lease time: the time the client has to renew the IP address.
- Rebinding time: the time the client has to rebind the IP address.
- Subnet Mask: the subnet mask of the network.
- Broadcast address: the broadcast address of the network.
- DNS.

- 6 In the request packet, why is there no relay agent?

checking the next DHCP packet(which is sent by the client), DHCP Request, the relay agent is not there. in the packet: Relay agent IP address: 0.0.0.0.

this is a typical behavior, you dont relay yourself, router do that. we see that relay agent IP address on the way back in the acknowledgement packet.

- 7 In the request packet, how does the client identify which server it wants an address from?

in the options section of the request packet, the client identifies the server using the option `DHCP server identifier`. to which the value is an IP address.

- 8 In the DHCP ACK - is this packet a broadcast?

When we look at the second layer headear (Ethernet II) this is what we see :

```
Ethernet II, Src: BelkinIn_9d:02:73 (94:10:3e:9d:02:73), Dst:
NetAlly_a1:17:9f (00:c0:17:a1:17:9f)
    Destination: NetAlly_a1:17:9f (00:c0:17:a1:17:9f)
    Source: BelkinIn_9d:02:73 (94:10:3e:9d:02:73)
```

So destination is clearly specified, hence it is a unicasted packet.

- 9 What is the lease time of the accepted address?

leaase time, rebinding time values are in the options section of the DHCP ACK packet.

in this case it is 24 hours.

- 10 What is the function of packets 8 and 9?

in my trace file, two packets coming after the DHCP ACK packets are ARP requests.

The use of these packets are to verify that the client has the IP address that the server has assigned to it. to they are broadcasting and advertising themselves to the network using gratuitous ARP.

in the first packet, it sends its own mack address and target IP address is it's own IP address. asking `IF anyone else has the same IP address` and the response is not given. the next packet is ARP announcement packet whath is again a broadcast and telling I have this IP address.

DNS