

# 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 nasıl çalışır?

Bir packet göndereceğin zaman, header oluşturmak için destination IP ve MAC address ihtiyacın var. sen başlangıçta kendi IP ve mac adresini biliyorsun, bir de serverinkini.

Target'in MAC adresini bulmak için ARP kullanılır. ilk önce local arp cache'e bakar. eğer varsa direkt cevap verir. eğer yoksa **arp request** gönderir. Network'e bunu broadcast olarak gönderir.

Bu broadcast domain içindeki 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 aslında şöyle:

18 numaralı bileti kimde öğrenmek istiyorsun. Elindeki listede varsa zaten sorun yok, yoksa ortalığa bağıriyorsun **18 numaralı bilet kimde?**

Herkes biletine bakıp **benimki x**, **benimki Y** diye cevap veriyor.

Sen de sonunda öğrenmiş oluyorsun ve header'i yaratıp gönderiyorsun.