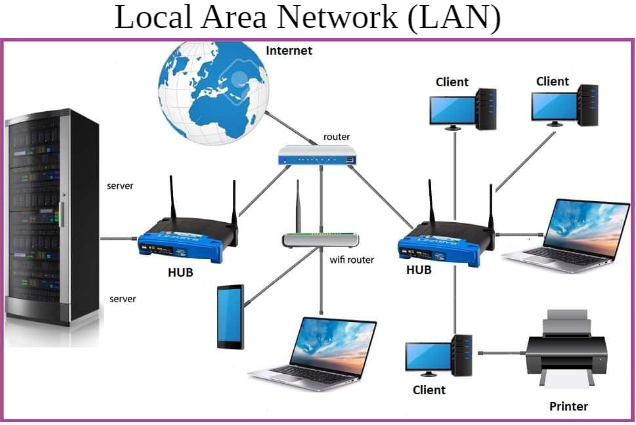
\*\*\*\*\*\*\*\*\*\*\*1.DERS\*\*\*\*\*\*\*\*\*05.03.2021\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ARPA.NET-->Amerikan Ordusu için geliştirilen ilk ağdır.73 yılında SOĞUK SAVAŞ'ın en üstte olduğu dönemdir.Ve çokça bu yıllarda bu şemalar vardır.

Daha sonra bu bizim bildiğimiz internete dönüşüyor.

**Local Area Network(LAN):** Fiziksel olarak aynı coğrafik bölgede bulunan bilgisayarların bir takım ağ bağlantı aygıtlarıyla birbirlerine bağlanmasıyla oluşan ağlara denir.

* ADSL modeminizin arkasında bulunan bilgisayarların bir arada oluşturduğu ağa deriz.
* LAN oldukça büyük bir ağı da kapsayabilir örneğin bizim TERZİOĞLU KAMPÜSÜ tamamı tek bir LAN’dan oluşuyor



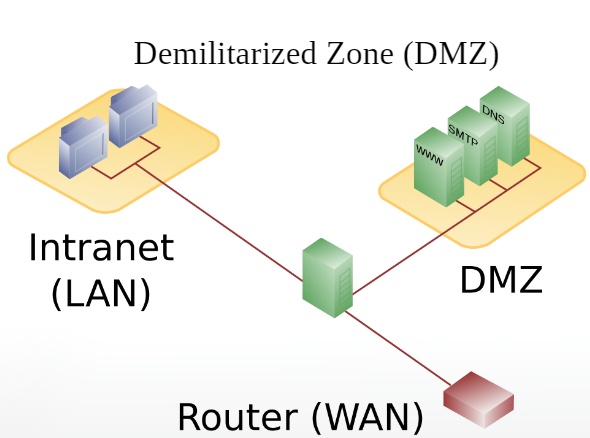
PAN(Personal Area Network): Bluetooth, Kızılötesi ile ulaşabildiğimiz kişisel alan ağlarıdır.

-Ders boyunca bilgisayar derken telefon tablet evin yerel alanına bağlı kol saati, kapı zili vs internete bağlanabilen tüm aygıtlardan bahsediliyor.

***Sunucu:***

* Bilgisayar ağlarında, diğer ağ bileşenlerinin erişebileceği, kullanımına ve/veya paylaşımına açık kaynakları barındıran bilgisayar birimi.

*Demilitarized Zone(DMZ)(Silahsızlandırılmış alan) :*

**

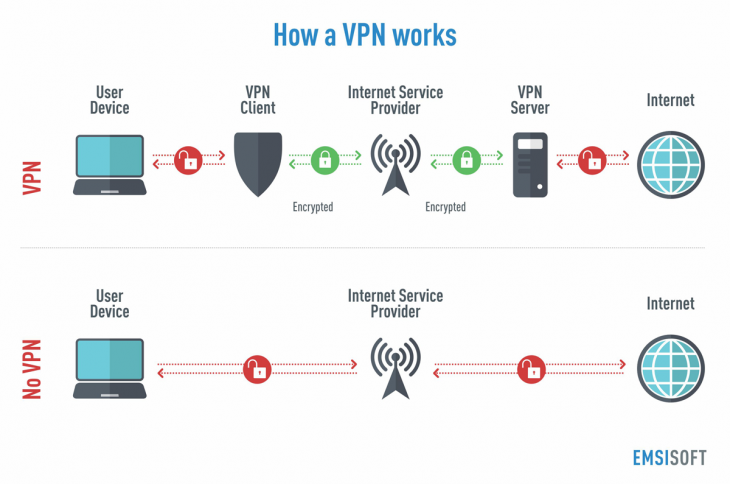
* Sunucularımızı bulundurduğumuz yerdir.
* Silahsızlandırılmış alan deriz çünkü dışarıdan içeri doğru gelen DMZe doğru gelen istekleri kabul ediyoruz.Doğal olarak yani. Sunucu var sonuçta istekleri kabul etmiyoruz diyemeyiz.
* Bir Yerel Ağ iki parçaya bölünmüş oluyor aslında.
* DMZ aslında LAN’ın bir parçasıdır.
* Her LAN’ın giriş ve çıkışında(zaten aynı yer)bulunan bir yönlendirici aygıtı vardır.
* Bu yönlendirici aygıt üzerinde sadece bu iş için tasarlanmış bir donanım olabileceği gibi bir bilgisayarda olabilir.
* Bu aygıt üzerinde bir takım kurallar yazıyoruz. (dışarıdan içeri gelen trafiğe şöyle içeriden dışarıya olan trafiğe şöyle davran gibi.)
* Eğer LAN içinde hiçbir sunucu hizmeti vermiyorsak oradaki gibi durumlarda bu yönlendirici üzerine diyoruz ki dışarıdan içeriye olan trafiği engelle diyoruz. Çünkü bu ev kullanıcıların yani ,LAN kullanıcılarının, dışarıdan gelecek saldırılara karşı bir önlem olması için yapıyoruz ama bulunduğumuz
* LAN içerisinde sunucularımızda var ise yani bir web sunucumuz,alan adı sunucusu,e-posta sunucu varsa o zaman dışarıdan gelen trafiği böyle düşüremeyiz. Ama bir takım sunucumuz var diye bütün kullanıcılarınızı dışarıdan gelecek olan bağlantı isteklerine de açmak istemeyiz.
  + O zaman yapılacak şey LAN’ı 2 parçaya bölmek olur.LAN var ve bir de sunucuları barındırdığımız DMZ kısmı var.

**Sanal Özel Ağ(VPN)-->**

*VPNSİZ* İNTERNETE NASIL BAĞLANILIR?:

* Bir istemci aygıtı var(bizim bilgisayarımız.) İnternet Servis Sağlayıcınıza bağlanıp internete çıkıyor.

*VPN* ile İNTERNETE BAĞLANMAK:



* Sanki kendi ağımızda değilde başka bir ağın içindeymişiz gibi davranıyor.
* İstemci bilgisayarın üzerinde bir VPN istemcisi var o servis sağlayıcısı aracılığıyla bir VPN sunucusuna kadar gidiyor. VPN istemcisi ve sunucusu arasındaki tüm trafik şifrelenip gönderilip alınıyor.
* Bu trafiği internet servis sağlayıcısı üzerinden geçirmemize rağmen şifrelendiği için servis sağlayıcı gönderip aldığımız verinin ne olduğunu bilemiyor ve bizim gönderdiğimiz internete bağlanma isteği sanki karşı taraftaki VPN sunucusu tarafından gönderilmiş gibi bir işleme tabii tutuluyor.
* Böyle olduğunda internet servis sağlayıcımız bizim bu internete ulaşmak istediğimiz adrese gitmemize engel oluyor olsa bile biz doğrudan o adrese değilde bu VPN istemcisi sayesinde VPN servere kadar gittiğimiz için bu internet servis sağlayıcının yaptığı kısıtlamalara da takılmamış oluyoruz.
* Bütün trafiğimiz VPN istemcisi ile VPN sunucusu arasında geçen yine de servis sağlayıcımızdan geçirdiğimizi ve sadece VPN istemci ile sunucu arasındaki trafiğin şifrelendiğini VPN sunucusundan çıkan trafik şifrelenmiyor. Dikkat!!!VPN sunucusu bizim üzerinden bütün trafiği akıttığımız bir sunucu olduğumuz için o sunucuya ne kadar güveniyoruz vs kısmını aklımızdan çıkarmamalıyız.

WAN(Wide Area Network)-->

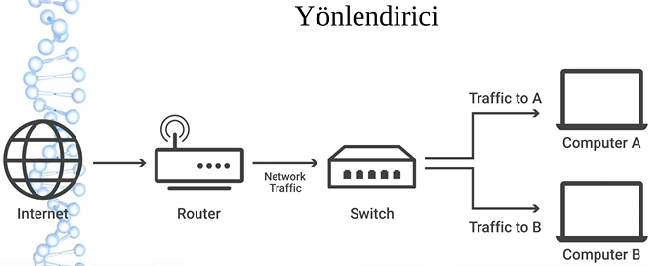
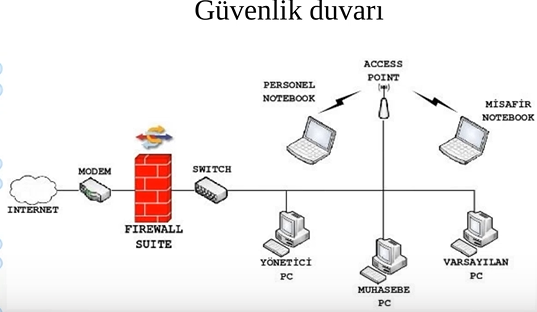
* Coğrafi olarak başka yerlerde olmalarına rağmen sanki hepsi aynı ağın içerisindeymiş gibi davranıyor.
* Okulun 20+ kampüsü var onların ağlarının birleştirilmesi gibi düşünüyoruz.
* Her yerleşke kendisi bir LAN oluştururken bunlar ağın dışına çıkan yönlendiriciler üzerinde yapılan tanımlamalarla deniyor ki hepiniz aslında şu WAN içerisindesiniz ve örneğin Bozcadada ağa bağlanan birisi sanki TERZİOĞLU kampüsü içerisindeki ağa bağlanan birisiymiş gibi davranabiliyor.

**İNTERNET:**

* LAN,WANlar hepsi bir araya geldiğinde birbirleri ile bağlandığında oluşan büyük ağa İNTERNET diyoruz.
* Herhangi bir WAN ya da LANın mutlaka internete bağlanması gerekmez. Sadece kendileri içinde makinelerin birbirleri içinde haberleştiği bir ağ olabilir.Evin içindeki ADSL bağlantısını kaybetsek dahi ağın içerisindeki bilgisayarlar aralarında haberleşebilir.

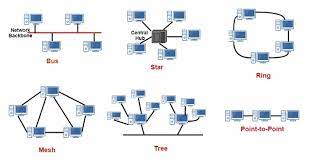
**AĞ AYGITLARI:**

1. **Tekrarlayıcı**
2. **köprü**
3. **switch**
4. **yönlendirici**
5. **güvenlik duvarı**

* **Tekrarlayıcı(Repeaterlar):**
  + OSI katman modelinde en alttaki katmanda çalışan cihazlardır.(1.Katman)
  + Aldıkları elektrik sinyallerini tekrar üretip karşıdaki cihaza iletirler.NEDEN BÖYLE BİR ŞEY YAPILIYOR? Çünkü elektrik sinyallerini taşıyan kabloların (koaksiyel kabloDAN fiber kablo) her kablo türünün elektrik kaybetmeden taşıyabildiği belli bir mesafe var.Bu mesafeden daha uzak bir mesafeye taşımak istiyorsak araya repeater koyarak bu işi çözebiliriz.
  + İçine kural yazamayız çünkü sadece elektrik sinyalini çoklayıp karşı tarafa iletir.
* **Köprü(Bridgeler):**
  + Tüm ADSL modemlerin çalışabildiği bir kip,moddur.Aldığı veriyi olduğu gibi başka bir cihaza aktarabilir.(repeaterler gibi)
  + Evde bulunan ADSL kendisine gelen veriyi alıp evdeki başka bir bilgisayara götürebilir.
    - Bunun için KÖPRÜ MODUNA almamız gerekir.
    - Köprü modunda repeaterdan farklı olarak MAC adresleriyle tanımlanır.ve iki ağ arasında bir tanımlama işlemi yapmış oluruz.
* **Switch**:
  + UTP kablo girebilecek portlar var. ADSLmodemlerimizde de.
  + Bu cihazlar 3. katmanda çalışan ağ aygıtlarıdır ve bu cihazları üçlü prizler gibi düşünebiliriz.
  + prizler bir taraftan gelen elektriği diğer portlarına çokluyorlarsa switchlerde bağlantı için giriş yaptığımız portlardan aldıkları ağ bağlantısını diğer portlarda çoklarlar.
  + Ama her bir portundan gelen dışarı giden isteği sadece ona götürürler ve tam tersi de aynı şekilde olur.
  + Üzerinde kurallar yazabildiğimiz aygıtlardır.
* **Yönlendirici(Rooterlar):**
  + LANların giriş ya da çıkışında bulunan aygıtlardır.
  + Mutlaka bir ağdan dışarı veri gönderebilmek için yönlendiriciye ihtiyacımız olur.
  + Bir LAN dediğimizde onun internete bağlanması gerekmez ama *internete bağlanmak için bir IP adresine sahip olmamız* ve karşı tarafa da gönderilen verinin ulaşabilmesi gerekir.
* **Güvenlik Duvarı(Firewall):**
  + Ağlarda ya yönlendirici ile aynı cihaz olur ya da onun önünde arkasında yönlendirici ile birlikte çalışan cihazlardır.
  + Üzerinde kural ailesi yazabiliriz.
  + Sadece LAN’ların girişinde bulunmazlar aynı zamanda WAN’ların giriş ve çıkışında bulunabilirler.
  + Vikipediye ulaşamadık bu FIREWALL’lar üzerinde tanımlanan kurallarla olan bir şeydi.
  + Trafik vikipediye gidiyorsa onları şuraya yönlendir diyordu ISP'ler.(Internet Servis Sağlayacıları)

**Ağ Topolojileri:**

1. **bus**
2. **yıldız**
3. **halka**
4. **mesh**

****

* **BUS:** Bilgisayarlar eğer birbirlerine bir kablo ile bağlılar, koaksiyel kablolarla, ve ayrıca bu makineler üzerinde bir başka ağ aygıtı
  + switchler falan sadece kablolar ile bağlıdır. Bu tip topolojilere BUS topolojisi denir.
  + Kablonun her iki ucunda da birer sonlandırıcı var .
  + Bir bilgisayardan çıkan veri başka bilgisayara veri yollamaya çalışırken herhangi bir sınırlayıcı olmadığı için ağdaki tüm bilgisayarlara gönderir bu veriyi.
* **YILDIZ(Star):**Çok yaygın olarak kullanılır.Ortadaki cihaz birbilgisayar olabileceği gibi bir switch de olabilir Çoğunlukla switchtir. Kablolu ise switchtir., kablosuzsa bir Access Pointtir.
  + Bu ortada bir erişim noktasının bulunması buradaki kablosuz erişim noktası ya da switch üzerinde bir iş. sis. çalıştığı için orada bazı kurallar yazabiliriz.
* **HALKA(Ring):**Başladığı yerde bitirirsek BUS RING’e döner. Ortada yine aktif bir cihaz bulunmaz sadece kablolarla makineleri birbirine bağlarız. Günümüzde ikisi de neredeyse hiç kullanılmaz.
* **MESH:** Yıldıza oldukça benzerdir.
  + Bilgisayarların her biri birbirlerine ayrı ayrı kablolar ile bağlılar ve ortada bu bağlantıyı yöneten bir switch yok.
  + Eğer saldırgan switchi ele geçirirse hem bağlantıyı manipüle edebilir hem de dinleyebilir.
    - MESHte ise herhangi iki bilgisayar üzerindeki veri dedicated bir kablo üzerinden yapıldığı için saldırganın aradan geçen veriyi dinleyebilmesi için doğrudan kablodan geçen veriyi dinlemesi gerekir. Aktif bir cihaz yoktur üzerinde.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2.DERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*11.03.21\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* ağ görevlerinden biri de--> sınırlı sistem kaynaklarının makineler arasında paylaştırılmasıdır.

**International Organization for Standardization**

\*Standart belirliyor

\*Merkezi İsviçre olan, 1947'de kurulmuş bir örgüt.

KATMANLI MİMARİLER:

**1-Open System Interconnection(OSI) Model**

\*1984'te ISO 7498 olarak yayınlanıyor.

\*7 katmanlı bir mimari ve protokollerden oluşuyor.

**2-Internet Protocol SUITE**

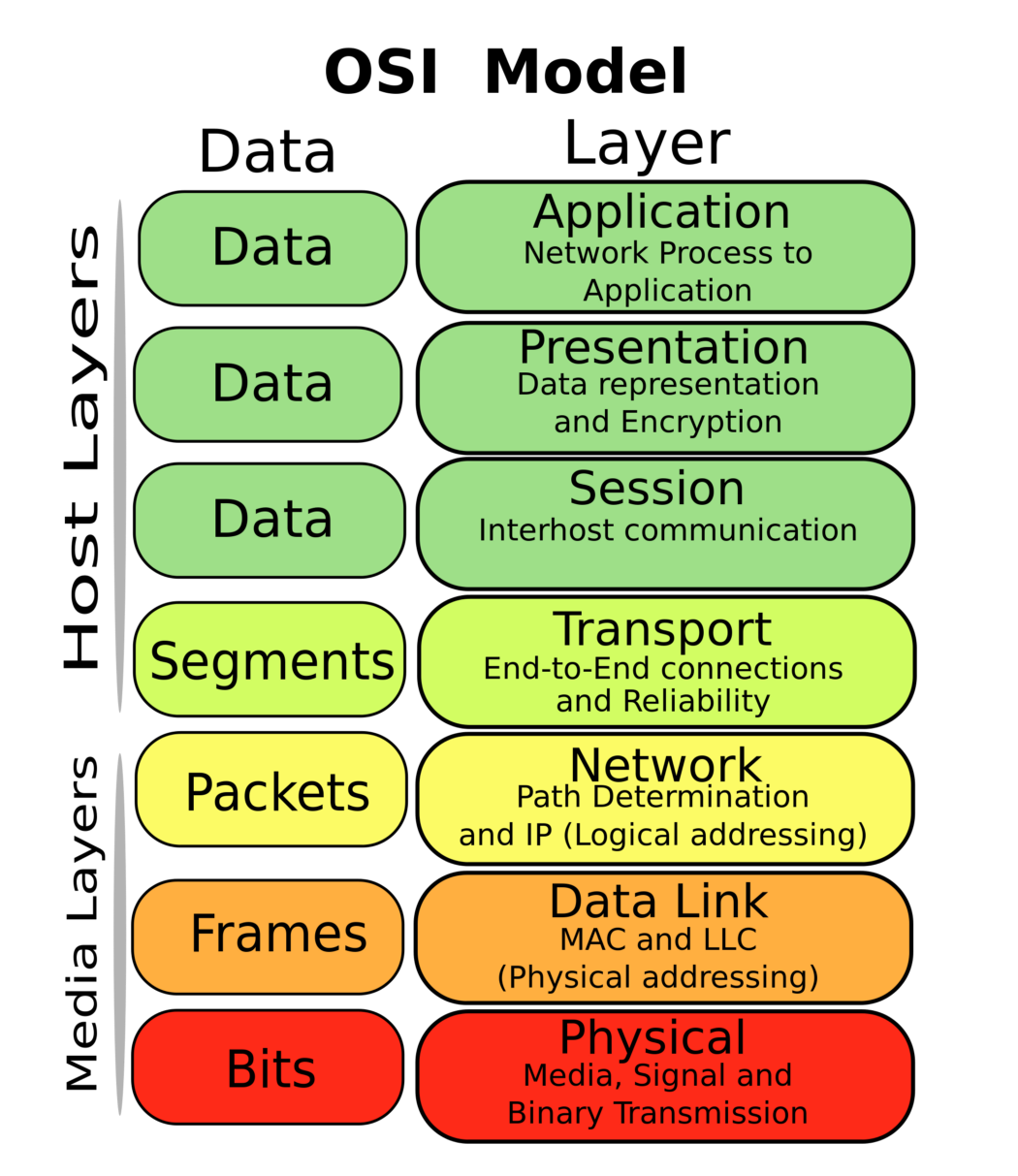
\*Internet Engineering Task Force(IETF)

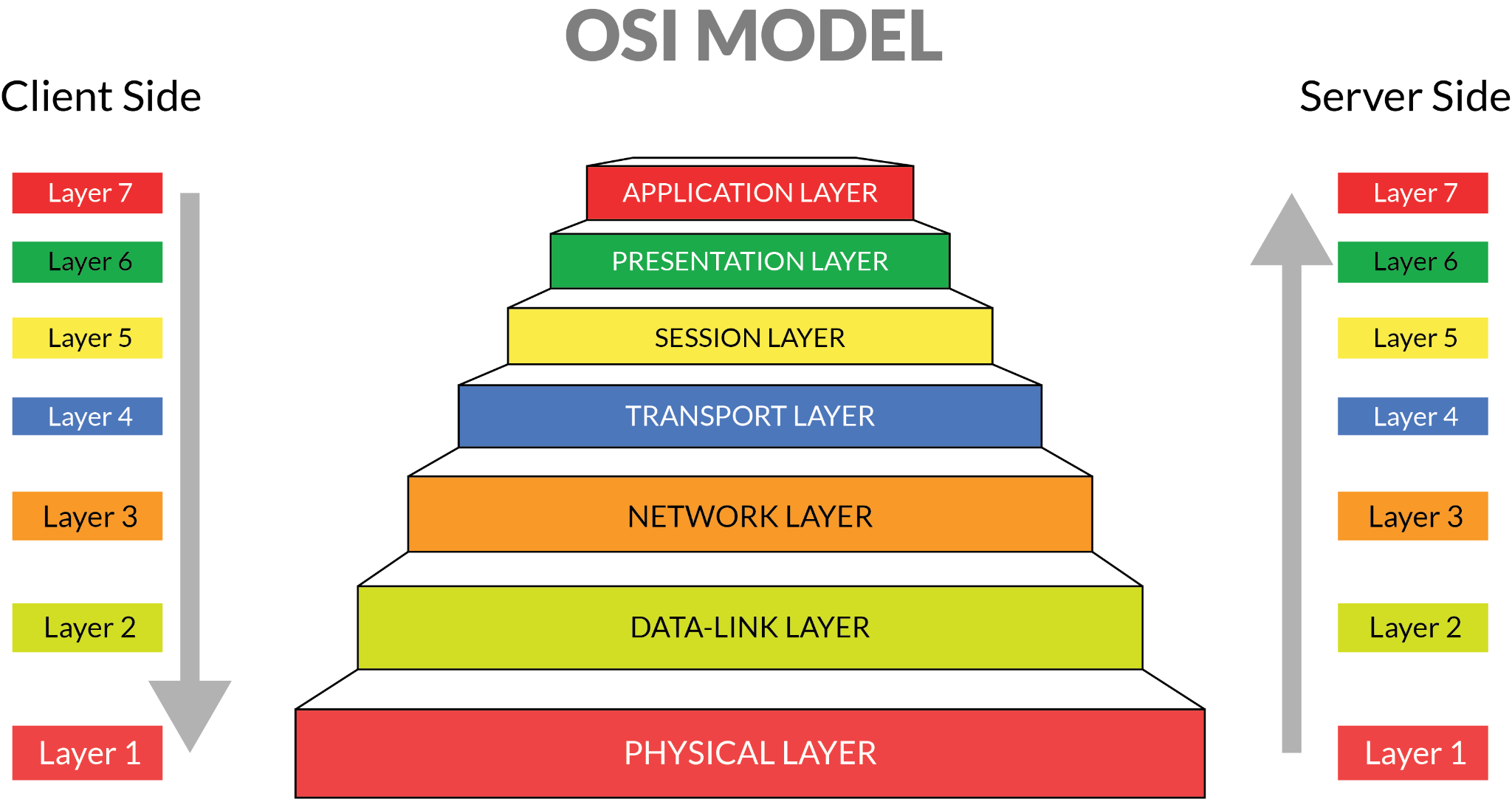
\*4 katmanlı bir mimari.

OSI MODEL

* OSI bir soyutlamadır!!!
* Bilgisayar üzerinde fiziki katmanlar değillerdir.
* OSI'de bir bilgisayarın diğer bilgisayara erişmesi için veriyi mutlaka 7 katmandan geçirmelidir.
* Karşı taraf fiziksel katmandan alıyor.

Bazen öyle cihazlar olur ki 1. katmanda çalışır sadece.

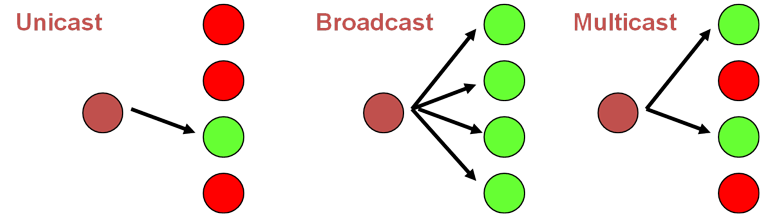
 en yukarı 7 en aşağı 1 olacak şekilde



* Veri her katmandan aşağı indirilirken üzerine bir başlık bilgisi ekleniyor.

**Veri Gönderim Türleri**

Broadcast-Multicast-Unicast



*BROADCAST:* O ağdaki bütün istemcilere yani nodelara,uçlara verinin gönderilmesi, çoklanmasıdır.

*MULTİCAST:* Bizim seçtiğimiz belli özellikleri taşıyan makinelere veriler gönderilir.(1+ makine)

*UNİCAST:* Bir uçtan bir uca verinin gönderilmesi demektir.

**KATMANLAR**

**1.KATMAN- FİZİKSEL KATMAN**:

* Ham veriyi 2 uç arasında taşımaktan sorumlu katmandır.
* Ham veri dediğimiz bizim gönderdiğimiz verilerin (bitler şekilde gönderiliyor). Sanki kablolardan 0lar 1ler aktarılıyormuş gibi düşünüyoruz ama hayır karşıya iletebilmek için radyoya,optik sinyallere ya da elektriğe çevirmemiz gerekir.
* Bu dönüşüm işleminden sorumlu katman FİZİKSEL KATMANdır.
  + Burada USB,BLUETOOTH,KIZILOTESI gibi protokoller fiziksel katmanın protokolleridir.

*Fiziksel katman problemleri:*

* Karşıya 0-1 leri nasıl ileteceğiz?
* Elektrik sinyallerinin bazı özellikleri olmalı. 5V 2 msn gönderdiğimde i 1 alacak yoksa 7Vu 3 sn gönderdiğimde mi 1 alacak.
* OSIden önce üreticilerin kararına kalmıştı. Örneğin sinyaller dışında veri gelince görmezden mi gelecek başka bir şey mi kabul edecek?
* Farklı ethernet kartı üreticileri 0 ve 1 in ne demek olduğunu belirlediğimizde artık bu standartlara uygun hem gönderen hem alabilen ağ cihazları alabiliyorlar.
* Bu katmanda veriler sadece elektrik ya da radyo sinyalleri ile taşınabildiği için 1.katmanda çalışan cihazlar(örn repeaterlar) bizim veri üzerinde oynama değiştirme filtreleme yapabileceğimiz cihazlar değildir.
* HTTP trafiğini şöyle taşı vs diyemeyiz çnk zaten verinin ne verisi olduğunu ve nereye gittiğini de fiziksel katman bilmiyor. O katman üzerinde tanımlanmış bir bilgi değildir.

**2.KATMAN -VERİ BAĞLANTISI KATMANI(DATA LİNK LAYER)**

* Kendi üzerindeki ağ katmandan verileri alıyor ve bir altındaki fiziksel katmana iletiyor.
* Veri bağlantısı katmanı donanım katmanına erişmek ve kullanmak ile ilgili kuralları belirler.
* Veri bağlantısı katmanı ağ üzerindeki diğer bilgisayarları tanımlama, kablonun o anda kimin tarafından kullanıldığının tespiti ve fiziksel katmandan gelen verinin hatalara karşı kontrolü görevini yerine getirir.
* Veri bağlantısı katmanı, fiziksel katmandan alınan bitleri veya bir üstteki ağ katmanından adlığı veri paketlerini çerçeve(frame) adı verilen bir formata dönüştürür.
* OSI de veri bir katmana gelmişse ve o makineden ayrılıyorsa her bir katmanı tek tek dolaşmalıdır. Bir katmanı atlatamayız.

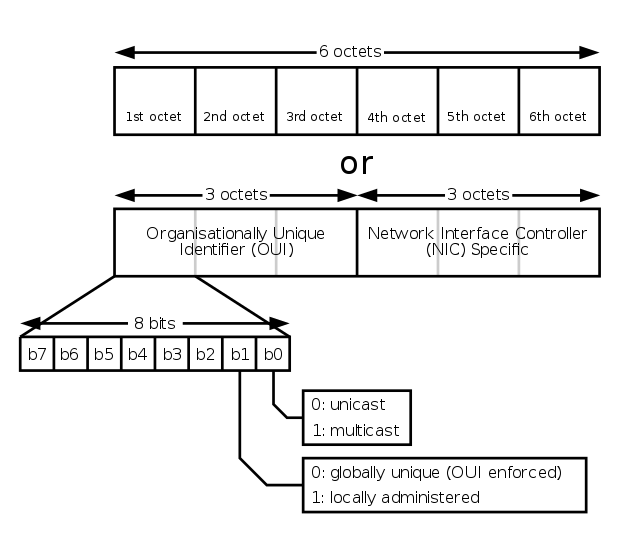
*iki alt katmanı vardır;*

**1-)Logical Link Control(LLC):**

* Veri bağlantı katmanının en üstünde çalışan protokolleri çoğaltır ve isteğe bağlı olarak akış denetimi(kullandığımız araç üzerinde haberleşme nasıl olacak aynı anda veri gönderip alınabilecek mi yoksa biri gönderirken diğeri dinleyecek vsvs bunlara karar verilecek.) onay ve hata bildirimi sağlar.
* Veri bağlantısının adreslenmesini ve kontrolünü sağlar.
* İstasyonları iletim ortamı üzerinden adreslemek ve kaynak ve alıcı makineler arasında gönderilip alınan verileri kontrol etmek için hangi mekanizmaların kullanılacağını belirler.
  + Gönderilip alınacak verilerin hangi mekanizmaları kullanacağı yani.

**2-)MAC katmanı(media access control):**

* Herhangi bir zamanda medyaya(veri haberleşmesi için kullandığımız kanal)( eskiden sadece kablo derken şimdi kablosuz erişim de çoklukla kullanılıyor.) kimlerin erişmesine izin verildiğini belirler.
* Alt katmanı veriyi hata kontrol kodu(CRC) alıcı(frammeden gelen mac adresini çıkarır) ve gönderenin(paketler) MAC adresleri ile beraber paketler(frame) ve fiziksel katmana aktarır.
* Alıcı tarafta da bu işlemleri tersine yapıp veriyi veri bağlantısı içindeki diğer alt katman olan LLC ye aktarmak görevi yine MAC alt katmanına aittir.
* ***media access control address(MAC ADRESİ)***
* Her ağ arayüzünün(NIC-NETWORK INTERFACE CONTROLLER) benzersiz bir MAC adresi vardır.
* Düşünüldüğünün aksine bilgisayarı değil doğrudan o donanımın kendisini tarif ediyor.
* Ethernet, Wi-Fi ve Bluetooth arayüzlerinin MAC adresleri vardır ve benzersizdir.
* MAC adresi bilgisayarı değil NIC'i işaret eder.



* MAC-48 2^48i temsil ediyor.
* Her oktet 8 bite karşılık gelir. Adresin tamamı 6 oktetten oluşur ve her oktet arası : ile ayrılır.6\*8=48
* ilk 24 bit yani ilk 3 oktet donanım üreticisini tarif eder. İlk 3 oktetine bakıp hangi firma arafından üretildiğini tespit etmek mümkün.
* Geri kalan 3 oktet ağ arayüzünü tarif ediyor.

MAC-48

örn:

54:bf:64:0d:2c:5e

* MAC adresi donanımın Salt Okunur kısmına ROM a yazılmış bir bilgidir.
* Makine üzerindeki MAC adresini kalıcı olarak değiştiremeyiz.MAC adresinin olduğu alan salt okunur bir alandır.
* Kalıcı olarak değiştirilmese de bir cihazdan çıkan her bilgi gibi farklı gösterilebilir.
* FF:FF:FF:FF:FF:FF broadcast adresidir.Artık ilk 3 oktet üretici bilgisi değil bu doğrudan bir kartı adreslemiyorda LAN daki bütün bilgisayarları adresliyor.
* İkinci katmanda yapılan veri alışverişinde kullanılır. Başka katmandalar ise MAC adresi daha üst katmanlarda kullanılan bir bilgi olmaz.
* Taraflardan biri farklı bir LAN'da ise iki taraf da karşının MAC adresini bilmez/ihtiyaç duymaz.

Laptopu kaybettiğimde MAC adresinden bulabilir miyim?Mümkün değil. LAN dışına taşınan bir bilgi olmadığı için MAC adresi, kullanılması mümkün değildir.

HATA TESPİTİ VE DÜZELTME

* Veri bağlantı katmanı, çerçevelemeye ek olarak, iletim hatalarını tespit etmek ve bunlardan kurtulmak için mekanizmalar da gerçekleştirebilir.
* Bir alıcının iletim hatalarını algılaması için, gönderenin gönderilen çerçeveye bir hata algılama kodu olarak fazladan bilgi eklemesi gerekir.
* Alıcı, bir hata tespit koduna sahip bir çerçeve elde ettiğinde, onu yeniden hesaplar ve alınan hata tespit kodunun hesaplanan hata tespit koduyla eşleşip eşleşmediğini doğrular.
* Bir hata tespit kodu, her N toplam bit sayısı dizisine karşılık gelen r yi(yedek -redundant- bitlerin miktarını) hesaplayan bir işlev olarak tanımlanabilir.
* En basit hata saptama kodu, bir alıcının iletilen N + r bitleri arasında tek bir biri etkileyen iletim hatalarını algılamasına izin veren **eşlik bitidir**.(parity bit)
* Birden fazla çevrilmiş bit varsa, o zaman kontrol yöntemi bunu alıcı tarafında açmayabilir. Eşlik hatası tespitinden daha gelişmiş yöntemler de mevcuttur.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3.DERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*18.03.21\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**3.Ağ Katmanı (Network Layer)**

●Veri paketinin farklı bir ağa gönderilmesi gerektiğinde(Bulunduğumuz ağın dışına çıkacaksa yani)(Farklı bir LAN’a gönderilecek ise) , pakete yönlendiricilerin kullanacağı bilginin eklendiği katmandır.

●Ağ katmanı, sunucular arası yönlendirme dahil olmak üzere kaynaktan hedefe paketin iletilmesinden sorumludur (veri bağlantı katmanı aynı bağlantı üzerindeki çerçevenin iletilmesinden sorumludur.)

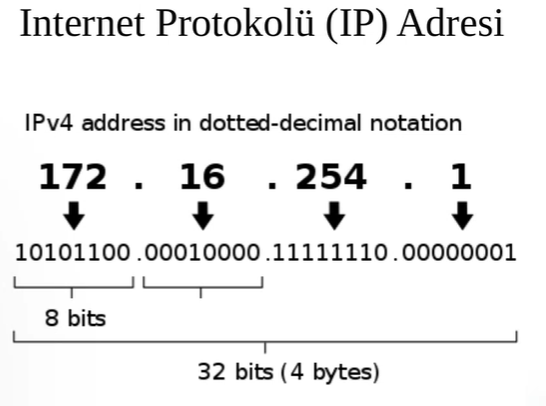
Ağ Katmanının İşlevleri

●Bağlantısız iletişim

●İstemci (host) adresleme

●Mesaj yönlendirme

\*Ağ katmanında mutlaka IP adreslemesi kullanmak gerekmiyor.Ama internete bağlanacak isek büyük ağa dahil olacak isek o zaman İnternet Protokolünü kullanmamız gerekir.



**IPv4 adres sınıflandırması**

* 32 bit(4byte)
* 4 tane oktetten oluşuyor.
* Her oktet 8 bitten oluşur.
* Yani her bir oktete 0-255 arasında sayılar yazmak mümkün.(256 tane)
* Bir IP adresinin hangi sınıftan oldugunu ilk oktetine bakarak anlarız.

A SINIFI:İlk oktet ağ adresini, kalan 3 oktet o ağın içerisindeki hostu yani istemciyi belli eder. 0.0.0.0 <---> 127.255.255.255

B SINIFI: İlk iki oktet sınıfı belli eder. 128.0.0.0. <---> 191.255.255.255

C SINIFI: İlk 3 oktet sınıfı belli eder geri kalan hostları. 192.0.0.0 <---> 223.255.255.255

012/8AT&T Bell Laboratories

–017/8Apple Computer Inc.

–019/8Ford Motor Company

–056/8US Postal Service

–022/8, 026/8, 029/8, 030/8 Defense Information Systems Agency

–193.255.97.0/24 Çanakkale Onsekiz Mart Üniversitesi

AYRILMIŞ (RESERVED) ADRESLER

10.0.0.0 - 10.255.255.255 A SINIFI

172.16.0.0 - 172.31.255.255 B sınıfı

192.168.0.0 - 192.168.255.255 C sınıfı BUNLAR HEP AYNI AĞDALAR.YEREL ALAN ADRESLERİDİR.

127.0.0.1 -> Loopback(Bilgisayarın kendisi ile haberleşmesi için kullandığı mantıksal bir aygıttır.)

169.254.x.y -> DHCP sunucusundan ip adresi alınmamış olması durumunda otomatik oluşturulur.Bir bilgisayarın internete bağlanabileceği bir IP adresi değildir.Eğer bilgisayarım IP adresi yapılandırmasını otomatik almak için ayarlanmışsa ve DHCP(cynamic host confg. protocol) dan IP adresini alamamış ise ya hiç IP adresi alamamış gibi davranır ya da kendisi otomatik 169.254 ile başlayan ve geri kalan iki oktetini kendisi ürettiği bir IP adresini almış gibi gösterir.

\*\*Bunu görür görmez erişemediğini anlamamız gerekir yani.

**SUBNET MASK:** Hangi ağın içerisinde bulunduğumuzu bize söyler.

* Bilgisayarımızdan çıkan bir paketin bizim LAN’ımızın içinde mi kalacak yoksa ağın dışına mı gönderileceğini bilmesi gerekir ki eğer aynı LAN içerisindeysek onunla 2.katman adreslemesi ile haberleşelim MAC adreslemesi ile.Eğer bizimle aynı ağda değilse o zaman onu bizim ağımızdan dışarı çıkaracak ve doğru yolu bulmasını sağlayacak yönlendiricimize yönlendirelim.Bunu bilebilmek için kullandığımız adres bilgisine NETMASK ya da ALT AĞ MASKESİ denir.

NETWORK HOST HOST HOST

CLASS A 255 0 0 0

NETWORK NETWORK HOST HOST

CLASS B 255 255 0 0

NETWORK NETWORK NETWORK HOST

CLASS C 255 255 255 0

kaç bit 1? 255-->8 oktet

NETBITS SUBNET MASK TOTAL-ADDRESSES

/20 255.255.240.0 4096

/21 255.255.248.0 2048

/22 255.255.252.0 1024

/23 255.255.254.0 512

/24 255.255.255.0 256

/25 255.255.255.128 128

/26 255.255.255.192 64

/27 255.255.255.224 32

/28 255.255.255.240 16

/29 255.255.255.248 8

/30 255.255255.252 4

242.2.4/25 (2 YE BÖLCEZ)

242.2.4.0 - 242.2.4.127 Bir ağın içinde

242.2.4.128 - 242.2.4.255 Diğer ağın içerisinde

242.2.4.0/26(4 e bölcez) 64 tane IP adresi olacak ağın içerisinde

242.2.4.0 - 242.2.4.63

242.2.4.64 - 242.2.4.127

242.2.4.128 - 242.2.4.191

242.2.4.1.192.-.242.2.4.255

* \*Eğer hedefteki makine ile bizim makinemiz aynı ağ içerisinde ise o zaman onunla ikinci katman adreslemesi üzerinden haberleşebilirim.
* \*Ama benim IP adresim ve kendi ağ maskem var.Benim Ip adresim ile ağ maskemi bir işleme tabi tutup kendi bulunduğum ağın adresini öğreneceğim.
* \*Sonra hedefteki Ip adresini ve elimdeki kendimle ilgili ağ maskesine bakıp hedefteki ağın adresini öğreneceğim.Bu ikisi aynı ise aynı ağdan yani 2.katmandan haberleşeceğim.
* Değilse o zaman bu veriyi hedefteki makineye ulaşması için ağımdaki yönlendiriciye ileteceğim göndereceğim paketi.
* 192.168.43.240/32 hiç 0 biti yok demek ki bir IP adresini gösteriyor.Ama o zaman neden direkt yazmadıkta /32 var?
  + \*Bu firewall yazarken işimize yarar.sadece o IP adresini göstermek için kullanırız.Sadece bu IP adresinden gelen verileri kabul et deriz.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*4.DERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*25.03.2021\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**AĞ KATMANI PROTOKOLLERİ**

1. IPv4
2. IPv6
3. ARP
4. ICMP
5. IPX

*örn-1:*

KAYNAK: 192.168.1.21/24 =255.255.255.0 (24 netmask)

HEDEF: 192.168.1.220

\*BU İKİ BİLGİSAYAR AYNI AĞDA mı yoksa ağları farklı mı.Bunu öğreneyim ki ona göre işlem yapayım yönlendirici(farklı) ya da aynı ağın içindeler ise bunu belirlemek için mutlaka netmaska ihtiyacım vardır.

IP ADRESİ 11000000.10101000.00000001.00010101

NET MASK 11111111.11111111.11111111.00000000

şimdi bu bitleri and işlemine tabii tutarsam

Ağ Adresi: 11000000 10101000 00000001 00000000

192 168 1 0

Broadcast:Bulundugum ağın en son adresi = 192.168.1.255

şimdi hedefteki ağ adresini kendi ağ maskemle ANDliycem

hedef: 11000000.10101000.00000001.11011100

netmask: 11111111.11111111.11111111.00000000

192 168 1 0

demek ki aynı ağın içerisindeler.

* Ağ adresi(ağı tarif ediyor), broadcast adresi(çoğa gönderim adresi)(veriyi ağ içindeki tüm bilgisayarlara göndermiş oluyoruz.) bilgisayarı adreslemek için kullanılamaz.

\*Burada yaptığımız işlem IP adresi ile Netmaskı bitwise işlemine tabi tutuyoruz yani AND işlemine.

1 olabilmesi için iki tarafta 1 olmalı.

AĞ ADRESİ: 11000000.110101000.00000001.00000000

192 168 1 0

broadcast: 192.168.1.255 #Bulunduğum ağdaki son IP adresine broadcast adresi denir.Veri yollarsak bu ağın iindeki tüm IP adreslerine veriyi yollamış olurum.

HEDEFTEKİ MAKİNE İÇİN HEDEF AĞ ADRESİNİ KENDİ NETMASK İLE İŞLEME TABİİ TUTACAGIM.

IP ADRESİ: 11000000.101010000.00000001.11011100

NETMASK: 11111111.11111111.11111111.00000000

AĞ ADRESİ: 1100000000.101010000.00000001.00000000

192.168.1.0

\*Demek ki aynı ağın içindeler!!!

örn-2:

KAYNAK: 192.168.1.21/25 =255.255.255.128

HEDEF: 192.168.1.220

IP ADRESİ 11000000.10101000.00000001.00010101

NET MASK 11111111.11111111.11111111.10000000

Kaynak AĞ ADRESİ: 11000000.110101000.00000001.00000000

192 168 1 0

Hedef ağ adresi :

IP ADRESİ: 11000000.101010000.00000001.11011100

NETMASK: 11111111.11111111.11111111.10000000

AĞ ADRESİ: 1100000000.101010000.00000001.10000000

192.168.1.128

\*Kaynak ve hedefteki ağ adresleri aynı değil. Demek ki veriyi karşı tarafa

ağ maskesi sayesinde hem kaynak hem hedef ağı bulabiliriz.

örn-3:

42.0.0.0/8 ilk oktet ağı kalan 3 oktet bilgisayarı tarif ediyor.

42.0.0.0 - 42.255.255.255'e kadar devam eder.

ağ adr. broadcast adr. ->Bilgisayar adresidir.

42.1.1.0 -->bilgisayar adreslemek için kullanabilir.

Hangi sınıftan bir IP adresi olduğuna bakmalıyız.Sonu sıfırsa mutlaka ağ adresidir diyemeyiz yani.

**BAŞLIK BİLGİSİ**

\*Data her katman indikçe.Ona bir BAŞLIK BİLGİSİ eklenir.

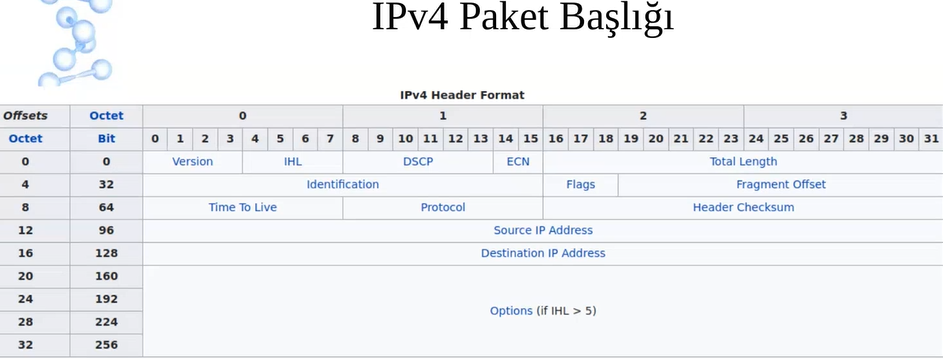
DATA application layer(SMTP,Telnet,FTP)

HEADER->DATA transport layer(TCP,UDP,ICMP)

HEADER->HEADER->DATA internet layer (IP)

HEADER->HEADER->HEADER->DATA network access layer(ETHERNET,FDDI,ATM vs)

***IPv4 Paket Başlığı:***

******

Başlık uzunluğu değişebilir.

**Version:** Başlık bilgisine baktığımız protokolun hangi protokol olduğunu söylüyor bize

**DSCD:** Type of service: Veriyi aciliyetine göre işaretleme imkanı vardı ama hep kullanılan bir şey değildir.Örneğin veri indiriyoruz ve aynı anda sesli konuşma yapıyoruz.Bu tip verileri burada tanımlıyoruz.

**ECN:** Ağda tıkanıklık var mı?vs

**Total length**: Gönderdiğim veri ile ilgili olan toplam uzunluktur.

**İdentification**: Fragları belirler.

**Flags:** 3 bitlik bir alandır.İlk bit sonra kullanılır diyerek rezerve edilmiş bir bit o yüzden hep 0 olarak gönderilir.

* Ondan sonra gönderilen bitler ise DF bayrağıdır. Eğer DF bayrağı varsa ve fragmantetiona uğraması gerekiyorsa ama 0 ise o zaman düşürülür.
* **MF:** Kendisinden sonra gelen başka fraglarda var anlamındadir.

**Fragment OFFSET:** Eğer fraga uğramışsa kaçıncı oldugunu bilgisini veriyor.

**TTL:** Bir paketin TTL değeri onun geçtiği her yönlendiricide düşürülür.Bu sayede sonsuza dek yönlendirilmesinin önüne geçilmiş olur.

**Protocol alanı:** Bir üst katmanın protokolu nedir bilgisini verir.UDP-->17 TCP-->6

**Header Checksum:** Başlığın özeti.Karşıya veri gönderilirken checksum alanı yoldaki verinin bozulmaya uğrayıp uğramadığını tespit edecek.

**FRAGMENTATİON:**

* Kullandığımız ağ her neyse bunun taşıyabileceği en büyük paket boyutundan(MTU) daha büyük ise daha küçük paketlere bölünmesi gerekir. Bu işleme FRAGMENTATION denir.
* Yönlendirici bir sonraki hop'un MTU'sundan daha büyük bir paket alınırsa iki şey yapılabilir;
  + \*Paket DF bayrağına sahipse paketi düşürür ve geriye bir ICMF mesajı döndürür.
  + \*Paketi MTU’ya uygun bir şekilde daha küçük parçalara ayırarak iletir.

**MTU:**

* Ağ katmanının taşıyabileceği maksimum veri boyutu

**TTL(time to live)** Her geçtiğimiz yönlendiricide bir düşürülüyor.Yönlendiriciler arasında sonsuza dek dönmemesini sağlıyor ve aynı zamanda bizim bir adrese ulaşırken hangi yönlendiricilerden geçtiğimizi de öğrenmemizi sağlıyor.

traceroute www.google.com

* 10 ile başlayan IP adresleri internette var olamazlar sadece yerel ağlarda kullanılırlar.

**HEADER CHECKSUM NASIL HESAPLANIYOR?**

Paket başlığı koyu, checksum altı çizili olacak şekilde;

**4500 0073 0000 4000 4011 b861 c0a8 0001**

**c0a8 00c7** 0035 e97c 005f 279f 1e4b 8180

\*Checksum alanı haricindeki başlık alanları toplanır.

4500 +0073+0000+4000+4011+c0a8+0001+c0s8+00c7 = 2479C

\*2+479C=479E *ilk biti ile geri kalanı topluyorum cnk 4 karakter elde etmeye calısıyorum*

\*Son adımda her basamağın hexadecimal tümleyeni alınır.

* B861

HEADER CHECKSUM NASIL DOĞRULANIR? hepsi tüm paket başlıkları toplanır.

4500+0073+0000+4000+4011+b861+c0a8+0001+c0a8+00c7=2fffd

fffd+2=ffff

\*Her basamağın hexadecimal tümleyeni alınır. 0000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*5.DERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*01.04.21\*\*\*\*\*

**IPv6 (in hexadecimal) ng yani next generation**

*2001: 0010000000000001*

*0DB8: 0000110110111000*

*AC10: 1010110000010000*

*FE01: 11111110000000001*

*0000:0000:0000:0000*

****

* İlk 4 blok ağ adresini gösterir
* Kalan 4 blok onun içindeki hostu adreslemek için kullanılır.
* IPv4ten; kullanım, çeşitleri ve bir ağ aygıtı üzerine atanabilen adres sayısı da bir miktar farklıdır.
* v6 adreslerinde v4te oldugu gibi A sınıfı B sınıfı C sınıfı adresler bir sınıflandırma yok.
* v4te farklı büyüklüklerde ip adresi alacak kadar küçük parçalara bölünebiliyordu.

v6da bölebildiğimiz en küçük ağın içerisinde 64 bit kadar(4 blok) ip adresi bırakmak gerekir.

**NEDEN IPv6’ya ihtiyaç duyuldu?**

* Temeldeki sebeplerden biri adaletsiz dağılımdır.
* v6nın çok daha geniş ip adres uzayı vardır.
  + IPv4: 4294967296
  + IPv6: 34028236692093846346374607431768211456(3.4x10\*\*38)
* Unicast:Tek bir IP adresini işaret eder .

Broadcast:Ağdaki bütün ip adreslerini mesaj göndermek için kullanılır.

Multicast: Bir grup IP adresi ile iletişime geçmek için kullanılır.

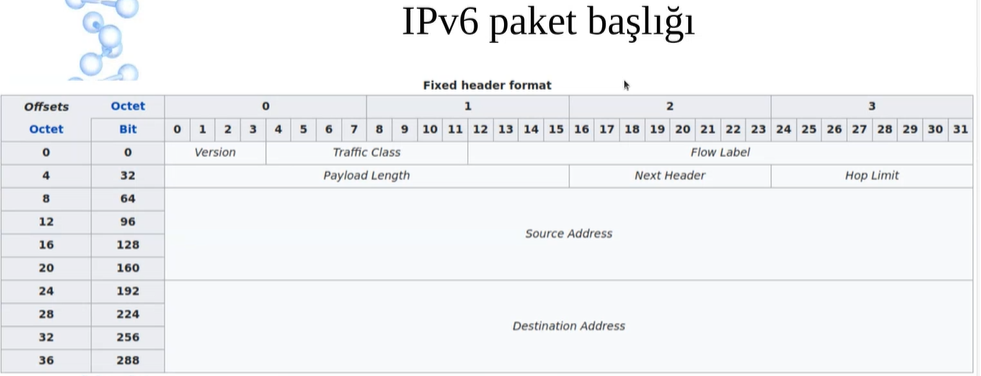
1. IPv4’te seçimlik bir özellik ama IPv6’da tümleşik olarak geliyor.

* SLAAC(duruma bağlı olmayan otomatik adres yapılandırması)(Temel farklardan biridir:)
  + v4’te IP adres yapılandırması nasıl oluyor?
    - Bulunduğumuz ağda otomatik yapılandırma gönderen bir sunucu var.Diyez tipi sunucusu.Dynamic Host protokolü kullanan.
    - Ya da siz elle yazmış oluyorsunuz.Ama elle yazarsak o ağın içerisindeki ağ yapılandırmasını biliyor olmamız gerekir çünkü yönlendirici bizim oraya yazdığımız herhangi bir adresi dışarı çıkarmaz.Zaten yönlendiricimiz çıkarsa dahi cevap bize gelmez.Yönlendirme protokollerinde daha çok üzerinde dururuz.
    - Yani IPv4 adresi bizim hangi ağda olduğumuza bağlı olarak değişir.(3. katman adreslemesi)
    - Mac Adreslemesi: Bu donanımın üzerine yazılı ve bilgisayarımızdan çıkarken biz değiştirmediğimiz sürece(kalıcı değiştirmek mümkün değil) sabit olarak kalır.Duruma bağlı olmayan bir adres yapılandırmasıdır.
  + IPv6da o kadar çok IP adresimiz var ki ağın içerisinde özel IP adreslerini kullanalım ve ağın dışına bunu başka IP adresi ile çıkaralım gibi yapılmıyor.
  + **NAT:**Aynı ağ içerisinde bulunan birden fazla cihazın aynı IP’yi kullanarak internete erişebilmesini sağlayan yöntemdir.Çünkü IPv4te sınırlı sayıda idi.
  + Fakat aynı zamanda makinelere kendi istekleri dışında bağlantı yapılmasını da önlerdi.Yani NAT İŞLEVSİZ(IPV4’te çok işlevliydi halbuki)
    - Her istemcinin kendi başına internete çıkabilecek IP adresi var
    - Bunun da çekinceleri mevcut
      * Mobil cihazların sürekli aynı IP adresine sahip olması yüzünden izlenmesi mümkün.
        + Bunun önüne geçebilmek için “temporary addresses ler” kullanılabilir.
* IPsec
  + IPv4’te paket başlığı düz metin olarak karşıya gönderilir.
  + Paket başlığı seviyesinde uçtan uca şifrelenmiş veri haberleşmesi yapmak istersek kullanabileceğimiz bir protokol IPsidir.
  + Önce IPv6 için tasarlanıp IPv4’e backport ediliyor ve yaygın olarak kullanılıyor.
* Mobility(Gezici)
  + Yönlendiriciler üzerinde yapılacak yapılandırma ile aynı IP adresi ile farklı ağlardan erişilebilir olmak.
* Link-Local IPv6 adresi(IPv4’te yoktur.)
* fe80:0000:0000:0000:b8ef:3ddd:f74a:32f5
* fe80::b8ef:3ddd:f74a:32f5 ların sıkıştırılmış hali :: ile sağladık bunu.
  + fe80 ile başlayan IPv6 ile başlayan adreslere denir.
  + Bu tip bir IPv6 adresi ile internete çıkmak mümkün değildir.
  + Sadece LAN içinde geçerli
  + Asla 3.katman yönlendiricisini(rooter) aşamaz.
  + **Neden ihtiyacımız var?**
  + Çünkü bulunduğumuz ağın içerisinde çok fazla IPv6 adresi var.En küçük bloğun içerisinde bile 64 bit tane IPv6 adresi var.
  + Bizim otomatik yapılandırmayı dağıtan bu ya yönlendirici olabilir ya da sunucu olabilir onunla haberleşebilmek için kullandığımız bir adres tipidir.IPv6 fe80 ile başlıyorsa bulunduğum ağın dışına çıkamam gerçek bir IP adresi değildir.
  + Bize internet servisini sağlayan firmalar bize v6 adresini vermeye karar verirlerse o zaman bu ayarların modem üzerinde de yapılması gerekir ve bu link local v6 adreslerinin yanında 2001 ile başlayan bir başka IPv6 adresi de görebileceğiz. v4 ile v6 adreslemesinin temel farklarından biri de budur.
  + v4te bir ethernet kartına bir tane v4 adresi atayabiliriz. Eğer bir ethernet kartının birden fazla IPv4 adresini kullanmasını istiyorsan üzerinde sanal arayüzler oluşturup öyle kullanılabilir.
  + vda ise aynı arayüzün üzerinde ihtiyacımız kadar v6 adresini atayıp kullanabiliriz.
  + 2001 ile başlayanlar internete çıkabilir.

**IPv4-IPv6**

* IPv6’da adres kısaltması
  + 2001:0db8:000:000:000:ff00:0042:8329 yerine
  + 2001:db8::ff00:42:8329 yazılıyor
* IPv6’da loopback adresi
  + 0000:0000:0000:0000:0000:0000:0000:0001 yerine ::1 kullanılıyor.
  + IPv4’te 127.0.0.1
* İki farklı protokol olduklarından bir ara mekanizma olmadan haberleşmeleri mümkün değil.

**Yazılımların IPv6 desteği ne demek?**

****

* Yazılımlar 7. katmanda çalışırlar ve 3. katman bilgisiyle ilgilenmemeleri gerekmez mi?
  + DNS: DNS’in kullanım amacı son derece basittir, kolay anlaşılabilir ve kullanılabilir makine ve alan isimleri ile makine IP adresleri arasında çift taraflı dönüşümü sağlar. IP adreslerinin gündelik hayatta kullanımı ve hatırlanması pek pratik olmadığı için domain isimlendirme sistemi kullanılır.
* Yazılım dediğimiz şey çok büyük oranda 7. katmanda Uygulama katmanının üstünde çalışıyor. Ve OSI katman modelinin temel motivasyonu her bir katman üzerinde tanımlanmış bilgiler ve görevler var ve katmanlar birbirlerinden izole bir şekilde çalışıyorlar.v4’teki adreslemenin ve bir üst katmandaki yönlendirmeyle ilgili işlemlerinin 7.katman üzerinde çalışan yazılımlar ile bir ilişkisi olmaması gerekir.
* v6 IP’nin 6. versiyonu değil, versiyon alanında 6 yazan halidir.(4 bit)
  + the constant 6(bit sequence 0110).
* **traffic class:** ağ trafiğini sınıflandırmak için kullanılmış bir alandır.(6+2)
* **flow label(20 bit):** Bazı akışları daha önemli diye işaretleyebileceğim ve onlar yönlendiriciler üzerinde öncelikli olarak işlem görebilirler.
* **payload length:** Paketin peşine eklediğimiz payloadın uzunluğunu gösterir.
* **next header**: v4te protocol olarak geçiyordu.Bir üstteki protocolun ne olduğu bilgisi vardır.3.katman verisi karşı taraf bunu aldı aça aça yukarıya doğru gidiyor paket başlıklarını. Burada söylediğimiz şey bir üstteki protokol nedir.TCP mi UDP mi vs.
* **Hop Limit:** v4 teki TTL’e karşılık geliyor. Yani bizim gönderdiğimiz verinin en çok kaç yönlendiriciden geçerek karşıya gitmesini istiyoruz. 8 bitlik bir alandır yani en çok 255 oluyor.Bu da veri paketinin yönlendiriciler arasında sonsuza dek dolaşması sorununu çözüyor.
* **Source Address ve Destination Address:** 128 bitlik iki tane adres alanı vardır.

**Adres Çözümleme Protokolü (ARP) v4 ile mac adreslerini dönüştürür.**

* 2.katmandaki kullandığımız MAC adresiyle 3. katmandaki kullandığımız IP adresinin birbiri ile dönüşümü işini yapıyor.
* 2 çeşit ARP paketi vardır.
  1. **ARP Request:** İstemcilerden bir tanesi ulaşmak istediği IP adresin hangi MAC adresiyle ilişkili olduğu bilgisini soruyor. İlgili bilgisayarda bu sorguyu aldığında ARP request paketine ARP Reply ile cevap veriyor.Broadcast adresine gönderilir.Yani ağdaki tüm bilgisayarlara gönderiyor.
  2. **ARP Reply**: Kendi IP adresini kendi MAC adresini yazıyor ve bunu hedefteki makineye gönderiyor.Hedefteki makinenin hem IP hem MAC adresini zaten biliyordu.Onu doğrudan bu sorguyu gönderen bilgisayara gönderiyor ve böyle olunca artık ağ makinesi 2. katman üzerinden haberleşebilir duruma geliyor.***Unicast*** ile tek bir makineye gönderildiğini unutma.
* NOT: ARP protokolü SADECE AYNI YEREL AĞIN İÇERİSİNDE BULUNAN MAKİNELERDE KULLANILIR!!!
* **PEKİ DÖNEN ADRES DOĞRU MU NASIL BİLECEĞİZ? YA BAŞKASI KENDI**

**ADRESINI YOLLADIYSA?**

**ARP ZEHİRLENMESİ, ÇOK TEHLİKELİ BİR SALDIRI TÜRÜDÜR.Hem yönlendiriciyi hem de ağdaki diğer bilgisayarlı zehirlemek için ve dnsleri üzerine almak için kullanılan tehlikeli bir saldırı türüdür.**

**ARP tablosunda bu eşleştirmeler adresler tutulur. Sadece bizim iletişimde olduğumuz makineler ile olan adresler bir arada tutulur.**

**ARP Başlığı**

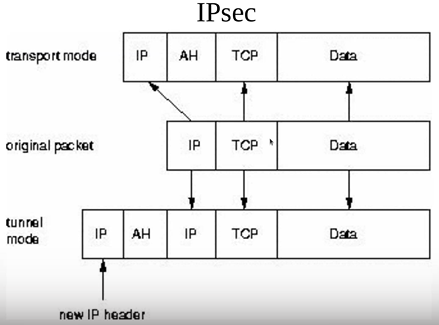
* Hardware type(HTYPE):
  + Kullandığımız donanımın tipini belirler.Ethernet ise 1 örneğin.
* Protocol type(PTYPE):
  + Gönderdiğimiz paket ARP Request mi yoksa ARP Reply paketimi ayırt etmek için kullanılacak.
* Hardware length(HLEN):
  + Eğer ether ise 6 yazarız.
* Protocol length(PLEN):
* Sender Protocol Address(SPA):
  + Donanım adres gönderici.
* Target Hardware Address(THA):
  + Alıcıya yönelik donanım adresi. Bu alanda istekler önemsenmez. Bir istek mesaji gönderilirken Varış Donanım Adresi’nin tamamı sıfır yapılır.
* Target Protocol Address(TPA)

**Komşu Keşfi Protokolü(Neighbor Discovery Protocol(NDP)) (arpnin v6 karşılığı)**

**Problemlere bakacak olursak;**

* Yönlendirici Keşfi(Router Discovery)
* Prefix Discovery
* Parameter Discovery
* Address Autoconfiguration
* Address Resolution
* Next-hıp Determination
* Neighbor Unreachability Detection
* Duplicate Address Detection
* Redirect
* **Yönlendirici Talep Mesajı(Router Solicitation):** Bu mesaj ağa bağlı yönlendiricilerin “Yönlendirici İlan” mesajlarını periyodik güncelleme zamanını beklemeden hemen yollamasını tetiklemek için yollanır.
* **Yönlendirici İlanı Mesajı(Router Advertisement)**: Ağa bağlı yönlendiriciler varlıklarını duyurmak ve ağa bağlanmak için gerekli parametreleri bildirmek için periyodik olarak ya da “Yönlendirici Talep Mesajına” cevaben “Yönlendirici İlan Mesajı” yayınlarlar.
* **Komşu Talep Mesajı(Neighbor Solicitation):** Bu mesaj ağa bağlı tüm düğümler tarafından diğer düğümlerin bağlantı katmanı adreslerinin bulunması ve daha önceden iletişim kurulmuş ve bağlantı katmanı adresleri komşu tamponuna(neighbor cache) eklenmiş komşuların erişilebilirliğinin kontrol edilmesi amacıyla kullanılır.
* **Komşu İlan Mesajı(Neighbor Advertisement):** Bu mesaj Komşu Talep mesajına cevap olarak ya da düğümde oluşan bağlantı katmanı adresi değişikliğinin ilan edilmesi amacıyla yayınlanır.
* **Yeniden Yönlendirme(Redirect):** Bu mesaj yönlendiriciler tarafından düğümlere yollanır. Mesaj içeriğinde belirli bir hedef IPv6 adresi için daha iyi bir yönlendirme yolunun varlığını belirtir.

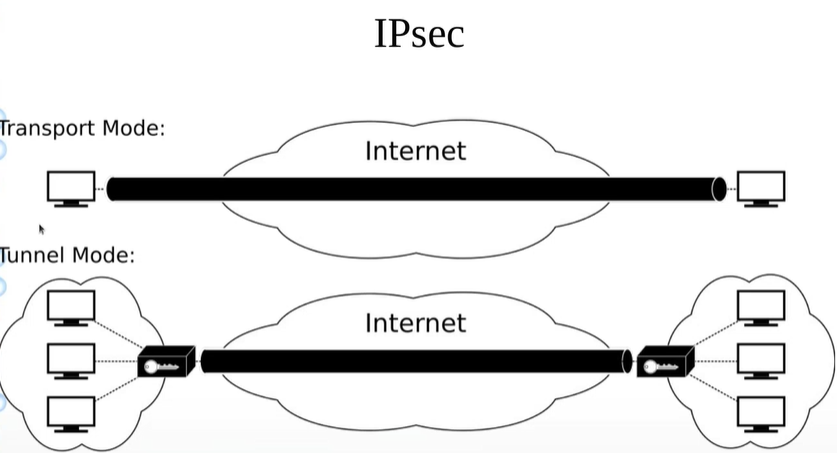
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*6.DERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*08.04.2021\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

****

* IP paket başlığından gönderilip alınan tüm veri düz metin olarak gönderiliyor.
* IPv4te checksum vardı.Paket başlığında bir alan bozulmuş mu bozulmamış mı bunu kontrol edebiliyorduk ama bu checksumın birden çok alan bozulduğunda yine bize paketin kaynağından çıktığı gibi bize ulaşmadığının garantisi yok.
* Gönderip aldığım tüm paket başlığı kısmını hem herkes görebiliyor hem de yönlendiricilerden geçerken üzerinde değişiklik yapılabiliyor.
* Biz çıktığı hali gibi isteriz.
* Data kısmı uygulama tarafından şifrelenebilir bir veridir.Biz HTTPS bağlantı kurmuşsak zaten gönderip aldığımız verinin kendisi şifreleniyor.
* Ama biz burada sadece verinin şifrelenmesini değil, paket başlığı seviyesinde bir şifreleme istiyoruz.Bunun için 2 kip vardır **transport** ve **tunnel mode.**

**IPsec:**

* **İnternet Protokolü Güvenliği, Internet Protokolü kullanılarak sağlanan iletişimlerde her paket için doğrulama ve şifreleme kullanarak koruma sağlayan bir protokol paketidir**
* **IPv6 için tasarlanmış ve sonra IPv4 için geriye uyumlu hale getirilmiştir.**
* IP ile haberleşen iki uç arasında uçtan uca şifreleme ve doğrulama yapılabilmesini sağlayan bir 3. katman protokolüdür.
* Simetrik şifreleme kullanılır.Veriyi hem şifrelerken hem deşifrelerken aynı anahtar
* kullanılıyor. Aynı gizli bilgi kullanılıyor.
  + Bütün işletim sistemlerinde kullanılabilir.
  + Veri uygulama seviyesinde değil ….IP seviyesinde.
  + kendi IP adresi ile gidebiliyorsa transport kipi gidemiyorsa tunnel ile karşıya taşınması gerekir.



Transport Mode:

Her iki uç arasında uçtan uca bir şifreleme yapmaktır.

* \*İkisininde temelde hedeflediği şeyler datayı ve onun üzerindeki TCP başlığını uçtan uca şifrelemektir.
* Her iki makinenin de IPsi varsa internete çıkacak o zaman bu ikisi arasında transport mode ile IPsec kullanmak mümkündür.
* Ama bazıları sadece yerel ağda olan IPlerdir. O zaman onları internete çıkarmak için birkaç yöntem gerekecek.
* Ortada bir veri var -data- onun üzerine yukarıdan aşağıya gelirken bir TCP başlığı eklenmiş onun üzerine de bir IP başlığı eklenmiş.
* **IP Paket Başlığı:** Gönderip alınan tüm veri düz metin olarak gönderiliyor.
  + IPv4 kısmında bir checksum kısmı vardı. Paket başlığında bir alanın bozulup bozulmadığı bilgisini tutardı.
  + Ama checksum birden fazla alan bozulduğunda yine paketin kaynağından çıktığı gibi bize ulaşıp ulaşmadığının garantisi yoktur.
  + Ayrıca gönderip aldığım bütün paket başlığını herkes görebilir. hem de yönlendiricilerden geçerken üzerinde değişiklik yapılabilir.
    - Bu hep bizim istediğimiz şey olmayabilir.
    - Zaten original packet kısmındaki DATA kısmı şifrelenebilir bir şeydir.Yani biz HTTPS bağlantısı kurmuşsak zaten gönderip aldığımız verinin kendisi şifreleniyor.
    - Ama bizim burada istediğimiz paket başlığı seviyesinde şifrelemedir.
    - Bunun için iki yöntem vardır.İkisininde amacı data ve onun üzerindeki TCP başlığını şifrelemedir.
      * Transport Mode: Uçtan uca şifreleme yapmak demek.
      * Tunnel Mode: Transport modundan bir miktar farklı olarak başka bir işlem yapılıyor.
  + Eğer her iki makinede kendi başlarına internete çıkabilecek IP adreslerine sahiplerse transport mode kullanmak mümkündür.Ama bazı IP adresleri yerel ağda haberleşmek için ayrılmış IP adresleridir.
* Her iki kipte de ilave bir AH var.Kimlik doğrulamak için kullanılıyor.Şifreleme işlemi için kullanıyoruz.

**Mobile IP:**

* İki tip adresten bahsetmiştik.İlki MAC adresi:Donanıma özel olarak üretilen ve bizim hangi ağdan bulunduğumuzdan bağımsız olarak HEP AYNı olan adresler.
* Diğerleri IP leri IPv4-v6 vs.Bunlar hangi ağda bulunduğumuza bağlıdır.Hatta bulunduğumuz ağda bizim ne zaman oturum açtığımıza bile bağlıdır.
* IPv6 ile birlikte kullanıma giren gezici IP adresi ise şöyledir yani Mobil olanı.
* 1+ ağ varsa(Home subnet ya da Visited Subnet), iki farklı ağ var yani, bunlardan birinin içindeyken kullanılan IP adresini başka ağa gidilince bunu ulaşılabilir bir halde tutabiliyor IPv6.Bu elbette hele ki ağın yönlendiricisi üzerinde ayarlar yapılmışsa olabilen bir şey.Bir mobil hoost var, mobil node (MN).Bu ilk başta kendi ağında, sonra başka bir ağa ziyarete gidiyor ve dışarıdan ona ulaşmak isteyen birisinin gönderdiği paketler esas bulunduğu ağa home subnete gidecek.Eğer yönlendiriciler üzerine ayarlar yapılmışsa ve dışarıdan biri MN ulaşmak isterse kendisine gelen verileri ziyaret ettiği yerdeki ağda taşıyabiliyor yönlendiriciler.
* **Bu aslında büyük kolaylıktır**
* **Yani IP adresi başka yere gidince de kullanılabilir ama aynı zamanda saldırılara açık hale gelir.**

**INTERNET CONTROL MESSAGE PROTOCOL(ICMP)**

* IP gibi karşıya veri taşımak için kullanılan bir protokol değildir.
* 3. katman protokolüdür ve IP adreslemesini kullanıyor.Bir üçüncü katman başka bir 3. katman protokolünü kullanabiliyor.Çünkü hedefe ulaşması için bir adresleme kullanması gerek.
* *Hataları raporlama için kullanılan, kontrol amaçlı bir protokoldür.Veri taşımayız.*
* ICMP, IP ile aynı katmanda olmasına karşın kendisi de IP yi kullanır.
* ICMP mesajları ağ yazılımının kendisince yorumlanır, yani ayrıca bir servis çalıştırmak gerekmez.
* 8-byte başlık ve değişken veri alanından oluşur.
* Son kullanıcının kullandığı uygulamaları: ping, traceroute(TTL’i +1 yaparak hedef makineye ulaşmayı amaçlıyor.TTL 0 olursa oradan ICMP mesajı döndürülüyor.
* traceroute bu protokolü kullanarak çalışıyor.3.katmanda çalışıyor.İzlenmek istemiyorsak ICMP zaten port kullanarak çalışan bir protokol değildir paketleri düşürerek bundan kurtulamayız.bunu durdurmak istiyorsak ICMP paketlerinin geçişine izin verme diyerek engel olabiliriz.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Offsets | Octet | 0 | 1 | 2 | 3 |
| Octest | Bit | 1234567 | 89**10**11**12**13**14**15 | 16**17**18**19**20**21**22**23** | 24**25**26**27**28**29**30**31** |
| 0 | 0 | Type | Code | Checksum | Checksum |
| 4 | 32 | Rest of header | Rest of header | Rest of header” | Rest of header |

ICMP BAŞLIĞI

* Type
  + ICMP type, see Control messages
* Code
  + ICMP subtype, see Control messages
* Checksum
  + Internet checksum (RFC 1071) for error checking, calculated from the ICMP header and data with value 0 substituted for this field.
* Rest of header
  + Four-byte, contents vary based on the ICMP type and code

INTERNET CONTROL MESSAGE PROTOCOL FOR IPv6(ICMPv6)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit offset | 0-7 | 8-15 | 16-31 |
| 0 | Type | Code | Checksum |
| 32 | Message body | Message body | Message body |

**TAŞIMA KATMANI(4. KATMAN):**

* Alt katmanlar(Transport Set) ve üst katmanlar (Application Set) arasında geçit görevi görür.Alt katmanlar verinin ne olduğuna bakmadan karşı tarafa yollama işini yaparken üst katmanlarında kullanılan donanım ile ilgilenmeden verinin kendisi ile uğraşabilirler.

PROTOKOLLER

1. TCP(Transmission Control Protocol)
2. UDP(User Datagram Protocol)
3. SCTP(Stream Control Transmission Protocol)
4. DCCP(Datagram Congestion Control)

NOT: Her bir katmana geldiğimizde o katmana ile ilgili yeni bir bilgi çıkıyor karşımıza. 2. katmanda donanım adresi(Bulunduğumu ağda hedefte bir donanım var ona erişmek için) bilgisi vardı.3. katmanda yani ağ katmanında IP adresi(Aynı ağdayken mac adresini dönüştürüp ona gönderiyorduk.Aynı ağda değilsek MAC adresi ile eşleştirmeyi yapmadan yönlendiriciye yolluyorduk bilgisi .Şimdi ise PORT bilgisi var.

* IP adresi ile kastettiğimiz şey makinenin internete bağlanan makinenin internet protokolü adresi idi yani bir makineyi kastediyor.
* Birden çok adres aynı makineye karşılık aynı adrese karşılık gelebilir. Aynı adres birden çok makineye de karşılık gelebilir.
* Ama şöyle bir senaryo olsun.Bir makineye ulaşmak istiyorum ama onun üzerindeki hangi servise ulaşmak istiyorum?
  + Bir makine üzerinde bir web sunucu olabilir aynı zamanda başka servislerde veriyordur örneğin veri tabanı hizmetinide sunuyordur.Biz o makineye ulaşmak istediğimizde onun üzerinde çalışan servislerin hangisine ulaşmak istediğimizi nasıl söyleyeceğiz?Sadece IP adresi ile yapamayız bunu.Öyle olsaydı tek bir çeşit servisin olması gerekirdi.

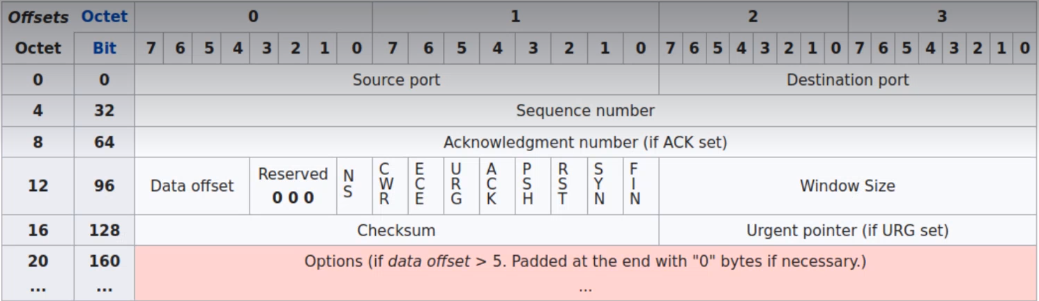
**Port:**

* Port ile bir IP adresi üzerindeki farklı servislere ulaşmak mümkün olur.
* 4.Katman bilgisi olduğundan ilk 3 katman protokolleri tarafından kullanılmaz
* 16 bitlik alan kaplar.
* Sunucular verdikleri servisler için sabit portları kullanırken istemciler rastgele(random) portlar kullanır.
* Bir servisin mutlaka bir porttan çalışması gibi bir zorunluluk yoktur.
* 0-1024 arası portlar
  + 20-21 FTP(File transfer
  + 80 HTTP
  + 53 dns
  + 443 HTTP Secure
  + 22 Secure Shell(SSH)
  + 123 NTP(Network Time Protocol)
* Portun açık olması demek o porta gelen istekleri dinleyen bir sunucu servisinin çalışıyor olması demektir.
* Portun kapatılması demek ya çalışan servisi durdurabiliriz ya da çekirdek seviyesinde makineye gelen paketlere bakıp onun hedefindeki porta gelmeden önce biz düşürebiliriz.
* FTP’ye bakarsak;
  + 21. porta baktığımızda hep açık olduğunu görürüz. ama FTP de 21. port bizim karşı tarafın kimlik kanıtlaması için kullanılan porttur.Kimlik kanıtlaması yaptığımızda o zaman iki bilgisayar IP adresi arasında 20. portta açılır ve FTP datayı kullanabilir yani karşıya data gönderip alabiliriz. Eğer 20. port açılmazsa bunu gerçekleştirmemişiz demektir.
  + İzin almadığımız hiçbir makine üzerinde tarama yapmamamız gerekir.
* NMAP: Karşıdaki makinenin hangi portlarından nasıl servisler çalıştırılıyor bunları söyler.
* Bütün iş. sis. için uygulaması mevcut.

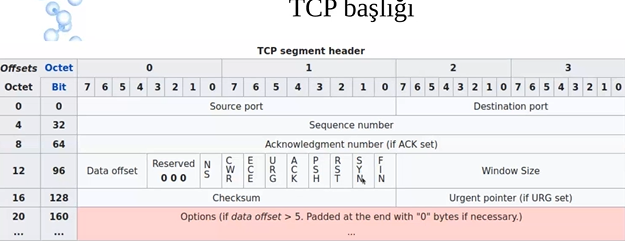
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*7.DERS\*\*\*\*\*\*\*\*\*15.04.21\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**TCP**:

* 4. katman protokolüdür.
* 2.katmanda adresleme bilgisi vardı donanım ile ilgili onu yerel ağda kullanıyorduk sadece.
* Bir üst katman olan 3. katmanda IP adresi vardı.
* 4. katmanda ilave olarak PORT bilgisi gelmişti.
* IP adresi bizim hangi makineye ulaşacağımızı söylerken PORT bilgisi bizim o makine üzerinden hangi servise ulaşacağımızı söylüyor idi.
* Her servis için atanmış bazı öntanımlı portlar olsa da servisler başka portlardan da çalışabiliyorlar.
* Gelişmiş bilgisayar ağlarında paket anahtarlamalı bilgisayar iletişiminde kayıpsız veri gönderimi sağlayabilmek için TCP protokolü yazılmıştır.
* HTTP,HTTPS,POP3,SSH ve FTP gibi internetin kullanıcı açısından en popüler protokollerinin veri iletimini yapar.
* Yönlendirme içinde kullanılır.
* TCP de UDP de daha alttaki katmanda IP yi kullanırlar.O yüzden TCP genellikle tek değil de TCP/IP diye anılır.
* Kullandığımız bütün uygulamalar alt tarafta ya TCP yi ya da UDP yi kullanırlar.
* Bağlantılı bir protokoldür ve üç fazla çalışır.
  1. Hedefle bağlantı gerçekleştirilir.
  2. Bağlantı gerçekleştikten sonra veri transferi yapılır.
  3. Veri transferi yapıldıktan sonra da bağlantı sona erdirilir.
* Karşı tarafa veri göndermeden önce karşı tarafın gerçekten kastettiği taraf olup olmadığını kontrol eder.Yani o IP adresine sahip mi değil mi kontrolünü yapar.



* Paket başlığında hangi tip bilgiler var bilmek önemlidir.Çünkü hangi katmanda bir iş yapacaksınız ve karşı tarafa iletmek istediğimiz bilgi o katmanda çalışıyor mu?
  + Eğer bir port bilgisini taşımak istiyorsak karşıya ve bunu uygulamanın içinden yapmayacaksak o zaman bunu en az 4. katmandan başlayarak taşımalıyım cunku onun haricinde 3-2 ve 1. katmanda port diye bir kavram yok.
  + Eğer karşı taraftaki makinenin özellikle şu porta bağlan demiyorsak o zaman 3. katmandan başlatabiliriz mesela verinin karşı tarafa transferini.
* Bizim kaynak makinedeki hangi portu kullandığımız bilgisini source porta yazabiliriz. Destination portta aynı şekilde.
* Örneğin HTTP nin portu 80.O zaman biz bir istemciden web sunucusuna bağlanmak istediğimizde 80’i destination port kısmına yazıyor olmalıdır.Burada ki source port 80 olmuyor elbette 1024’ün üstünde rastgele bir port açılıyor bilgisayarımızda ve karşı taraf ile böyle iletişim sağlıyoruz. veri bize dönerken sunucunun source portuna yazılırken 80 olur.
* örneğin yerelimizde rastgele açılan port 2000 olsun.bu port hedefteki tek bir makine portu ile iletişim kurabilirsiniz.Bir başka bağlantıya ihtiyac varsa makinede yeni bir port açılır karşı tarafa öyle istek gönderilir.
* Birçok uygulama 80. porta bağlanabilir farklı makine ya da aynı makinede olabilir ama bu bağlantılar için açılacak kaynak portun her seferinde yeniden belirlenmesi gerekir.

******

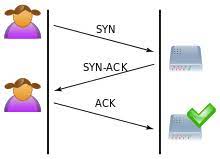
***TCP BAŞLIĞI***

* **Source port:** Veriyi gönderen tarafın kullandığı TCP portu.
* **Destination Port:** Hedef makinenin TCP portudur.
* **Sequence Number(Sıra Numarası):**
  + Eğer SYN bayrağı 1 ise; başlangıç sıra numarasıdır,
  + Eğer SYN bayrağı 0 ise; mevcut oturum için segmentin sıra numarasıdır.
* **ACK Number(Onay Numarası):** Alınan bir SYN paketine karşılık olan onay mesajı ACK biti ile gönderilir.
* **Başlık Uzunluğu(Data Offset):** TCP başlığının uzunluğunu gösterir.
* **Reserved:** İlerde kullanılmak üzere saklı tutulmuştur.
* **Window Size:** Akış kontrolü için kullanılır ve pencerenin boyutunu içerir.
* **Checksum:**Segmentin hatalı ulaşıp ulaşmadığını kontrol etmek için kullanılır.
  + 16-bit checksum alanı kaynak IP adresi,hedef IP adresi, TCP protokol adresi (0x0006) ve payload da içeren TCP başlığının bir özetini alarak başlığın hata kontrolünü yapmak için kullanılır.
  + Karşı taraf için güvenlik anlamı taşır ama her şeyi garanti etmez.
* **Urgent Pointer:**Bir verinin acil olarak iletilmek istendiği durumlarda kullanılır.
* **Option:**İhtiyaca bağlı seçimlik seçenekleri içerir.

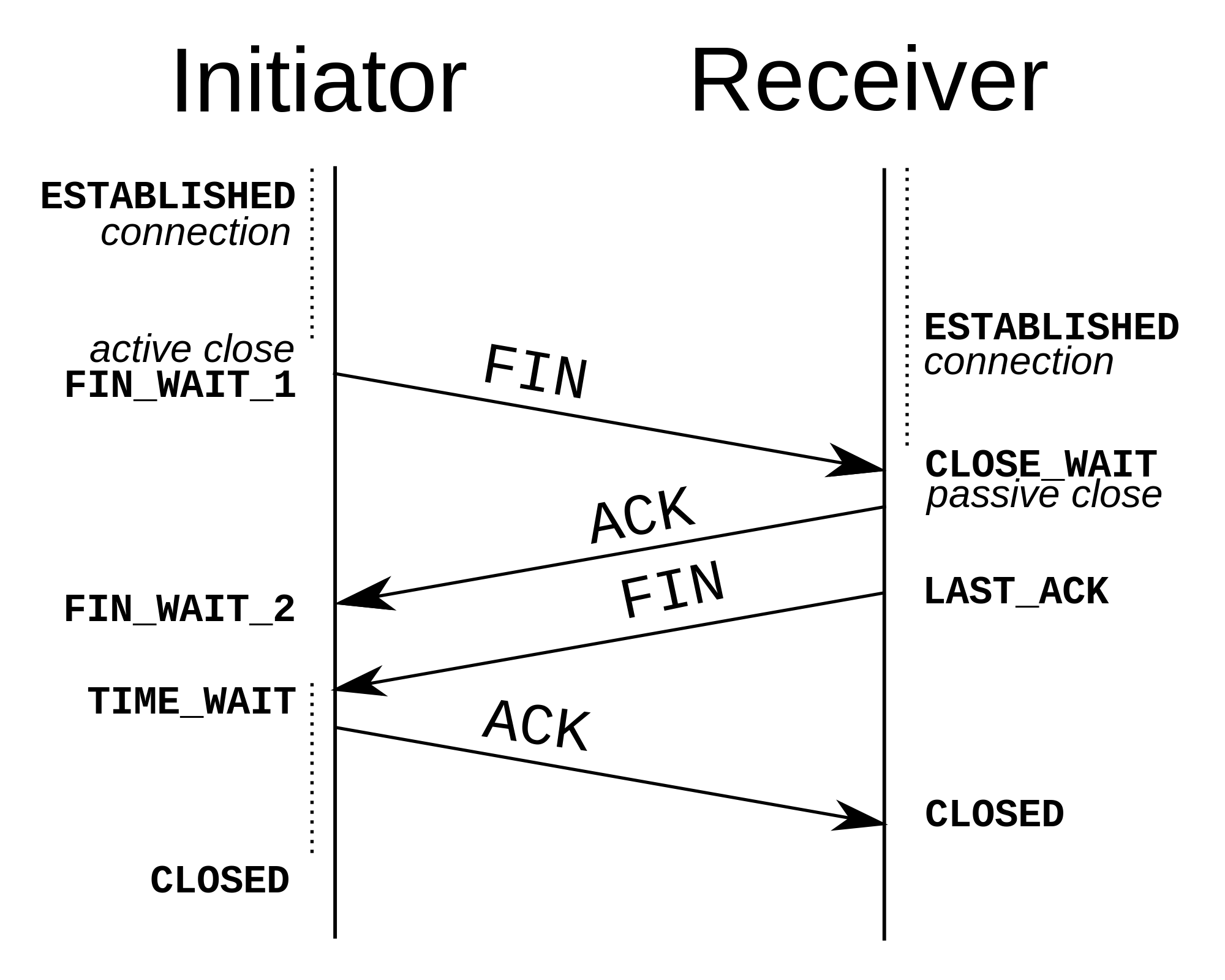
\*Burada sadece source bilgisi var ama hedefte hangi IP adresine ulaşacağım bilgisi yok cunku bu bilgi bize önceki katmanda belirtiliyor.

\*Bir bilgi birden fazla kez işlenmiyor OSI katmanında.

BAĞLANTI NASIL GERÇEKLEŞTİRİLİR?(Üçlü el sıkışma)

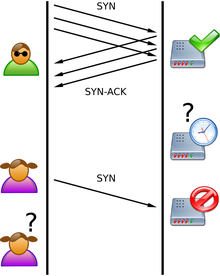
* OSI katmanları birbirinden soyutlarım her birine farklı görevler atar.
* Karşıya veri ulaşmadan önce bağlantıyı sağlamalıyız.illa client server olmasına gerek yok eşler arası da bağlantı olabilir. seq number rastgele x atanmış.
* 
* **SYN** seq=x
* **SYN-ACK** ack=x+1 seq=y
* ACK ack=y+1 seq=x+1
* SYN’yi karşıya gönderiyoruz yani karşının IP adresine ulaşıyor. Bunu neden yapıyoruz?İki tarafta birbirinin kimliğinden emin(IP adresi) oluyor.
* Sunucu da bu syni aldığını bana gösterebilmeli bunu syn-ack paketi geri göndererek yapıyor.
* MAC adresini farklı bir adresmiş gibi karşı tarafa gösterebiliriz.
* Kendi IP yerine başka bir IP adresi yazabiliriz veriyi böyle yollayabilriiz ama o zaman sunucu bunu alıp yerini değiştirip bize geri döndürürken bize değil değiştirilen IP adresine gider ve biz bunu değiştirip geri yollayamayız. TCP ille hangi IP adresi ile konustugumuzdan emin oluruz.

Bağlantı sona erdirilmeli:



FIN bayrağını etkinleştiririz.

*SYN FLOOD SALDIRI*

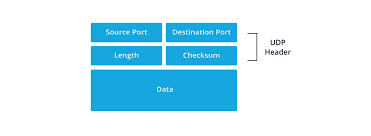


Belli bir IP adresinden gelen istekleri reddet dersek yine bu saldırıdan kurtulamayız cunku zaten bu IP adresinden asıl istek yollayamıyor ki.

Kaynak IP adresi değişince saldırıya uğramış oldugumuzu anlarız.

UDP:User Datagram Protocol

* Bağlantısız bir protokoldür.TCP de başlatıp sonlandırmak için üçlü el sıkışması gerekir.
* Karşı tarafın gerçekten kastettiği IP ye sahip mi değil mi emin olamıyor bağlantısız olması bunu kaybettirir.
* tcp de bu bağlantı işi ek paketler gerektirir ve maliyetlidir.
* Ses ve görüntü aktarımı gibi gerçek zamanlı veri aktarımlarında daha çok kullanılır çünkü çok az işlem yapmak gerekir TCP ye göre.
* Karşı tarafa daha hızlı veri alışverişi yapıyor olmak güvenlikten vazgeçmemizi gerektirir.
* ,Öyle bir paket olsaydı ki hem TCP kadar güvenli UDP kadar hızlı onu kullanırdık.
* Karşılıklı veri aktarımı için sadece iki protokol yoktur tabi daha mikro şekilde gönderilen protokoller vardır.



0-16 -->source port 16-31 → hedef port

* 3. katman paket başlığının içinde bize 4. katmanın protokolunun ne oldugu söylenmeli yoksa iş. sis. bunları nasıl işleyeceğini bilemez.
* IPv4te checksum alanı isteğe bağlı IPv6da ise zorunlu.

TCP-UDP

TCP:

Güvenilir: TCP mesaj iletimi,yeniden iletimi ve zaman aşımlarını yönetir.Mesajı iletmek için birden fazla deneme yapılır.Veri yolda kaybolursa, veriler yeniden gönderilir.TCP de, eksik veri olmaz, birden çok zaman aşımı durumunda bağlantı kesilir.

Sıralı: Bir bağlantı üzerinden sırayla iki mesaj gönderilirse, ilk mesaj ilk olarak alıcı uygulamaya ulaşacaktır.Veri segmentleri yanlış sırada geldiğinde, TCP, tüm veriler düzgün bir şekilde yeniden istenip uygulamaya gönderinceye kadar verileri buffer alır.

Ağır: Herhangi bir kullanıcı verisi gönderilmeden önce bir soket bağlantısı kurmak için üç paket gerektirir. Güvenilirlik ve tıkanıklık denetimini gerçekleştirir.

Akış: Veriler bir bayt akışı olarak okunur.

Paket başlığı arttıkça gönderilen veriden kısmak durumunda kalacağız ve karşıya bir seferde gönderilecek verinin miktarından kısmış oluruz.

UDP

Güvenilmez: Bir UDP mesajı gönderildiğinde, hedefine ulaşıp ulaşmayacağı bilinemez; yol boyunca kaybolabilir.Kabul, yeniden iletim veya zaman aşımı kavramı yoktur.

Sırasız: Aynı alıcıya iki mesaj gönderilirse, geldikleri sıra tahmin edilemez.

Hafif:Mesaj sıralaması, izleme bağlantısı gibi kavramlar olmadığından TCP’den daha hafiftir.

Datagrramlar: Paketler ayrı ayrı gönderilir ve yalnızca geldikleri takdirde bütünlükleri kontrol edilir.

Tıkanıklık Kontrolü Yoktur: UDP nin kendisi tıkanıklığı önlemez.Tıkanıklık kontrol önlemleri uygulama seviyesinde yapılmalıdır.

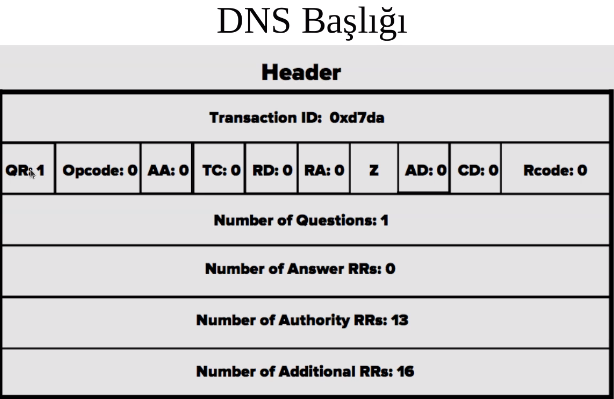
Çoğa Gönderim(Broadcasts): Bağlantısız bir protokol olduğundan, çoğa gönderim yayını yapabilir- gönderilen paketler, alt ağdaki tüm cihazlar tarafından alınabilecek şekilde adreslenebilir.

Çoklu Gönderim: Tek bir veri birimi paketinin çok sayıda istemciye çoğaltma olmadan otomatik olarak yönlendirebileceği çok noktaya yayın çalışma modu desteklenir.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*8.DERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*22.04.2021\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DNS(Alan Adı Sistemi):

* **En basit anlamıyla DNS, internet dünyasının telefon rehberi olarak ifade edilebilir.** İnternet dünyasında her bilgisayarın bir adı ve her sunucunun bir alan adı vardır. Bu alan adlarına ulaşabilmemizi sağlayan DNS, **telefon numaralarında olduğu gibi 10.30.60.100 gibi isimlerin IP adresi karşılığını tespit edebilmemizi sağlamaktadır.** Bu sayede tarayıcıya yazdığımız bir internet adresine ulaşabilmemiz sağlanmaktadır.
* Üçüncü katman iletişiminde iki tarafın adresleri sadece IP(v4-v6) adresi olarak bulunur.
  + Kullanıcılar bu adresleri hafızalarında tutamazlar.
  + Aynı IP adresinde birden fazla makine adı için servis verebilir.
  + Kullanıcılar ancak [www.comu.edu.tr](http://www.comu.edu.tr) gibi adresleri hafızalarında tutabilirler.
  + Servisi veren sunucunun IP adresi değişebilir.Bunun servisi etkilememesi gerekir.
* DNS hiyerarşik ve dağıtık bir sistemle çalışır.
* Neredeyse her internet servisi için temel bileşendir(Kullanıcının görüp tanıyabileceği makine adını ona göstermeliyiz ki o gerçekten kullanmak istediği servis bu mu kullanıcı karar versin.)
* alan adı <-> IP adresi dönüşümünü yapar.
* E-posta servisi için gereklidir.
* DNS kullanabilmek için bir makineye bir IP adresine gidip üzerinden bir servise ulaşmamız gerekir.Sadece bir IP adresine gitmek yetiyorsa o zaman 3.katman yeterli olur. Ama eğer IP adresi üzerindeki bir servise ulaşacaksak port kavramı yani 4.katmanı kullanmalıyız.Yani dns için en az 4. katmanı kullanıyor olmalıyız.
* genellikle *UDP 53.portu* kullanır.



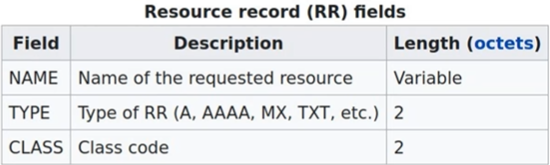
QR 0 = sorgu gönderdik

QR 1 = cevap aldık

OPCODE 0 karşıya standart sorgu

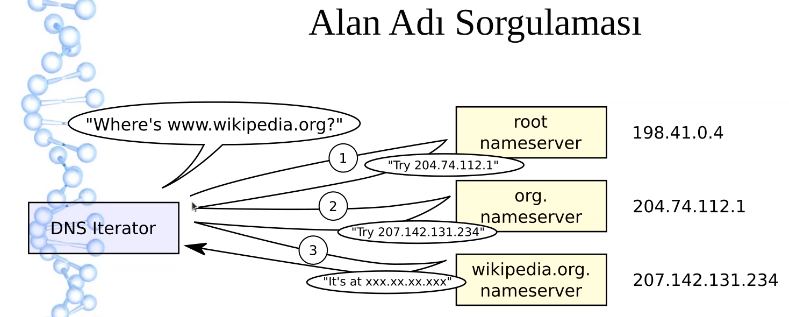
OPCODE 1 ters sorgu ya da server durumunu sorguladığımız anlamına gelir.

DNS Sorgusu:



**DNS sunucuları:**

* Yetkili sunucular
* Sadece çözümleme yapan sunucular.



Yepyeni bir alan adı ortaya çıktığını düşünelim.En yukarıda 13 tane root name server var.dünyanın farklı yerlerine yayılmış durumdalar.Sadece birine saldırıp ağ bağlantılarını tüm dünyada sekteye uğratmak mümkün değildir.

Önce root name servera sorup adresi sağdan sola doğru çözümlemeye çalışıyorum.

tüm IPleri aldıktan sonra alan adına bakan yetkili DNS sunucusuna ulaşmış oluyorum.

Her bir DNS sorgusu için bu işlem baştan çalıştırılmıor.Öyle olsaydı root nameserverlar üzerinde muazzam bir trafik oluşturmuş olurduk.

Gerçekte biz sorgulamayı son kullanıcılı bilgisayarı üzerinde yazılmış olan DNS sunucusu neyse sorguyu ona gönderiyor(evdeki modem).

(Eğer kendimiz DNS sunucumuzu değiştirip başka veri tanımlamadıysak)

Eğer bizim gönderdiğimiz sorgu DNS in belleğinde varsa sorgu ona iletiliyor yoksa o sorgu DNS in üstünde olan bir üstteki DNS sunucusuna gidip bu sorguyu gönderiyor.

NEDEN 13 TANE ROOT NAME SERVER VAR?

* 13 tanesi de hepsi aynı swervisi verip aynı bilgiyi birbirleri üzerinden eşliyorlar.Biri ulaşılamaz olsa bile geri kalanına ulaşılabilir.
* İlk bu altyapı tasarlandığında ağ üzerinde 512 bytelık veri taşınabiliyor.IP 32 bitlik alan yapıyor.
* Bunlardan 13 tanesini bir araya getirdiğimizde geriye ancak protokol ile ilgili bilgi bırakacak bir alan kalıyor.

KAYIT TÜRLERİ

Start of Authority(SOA):

IP addresses(A ve AAAA): A kaydı adres kaydı.2001 ile başlarlar.

SMPTP mail exchangers(MX) :e posta da kullanılan bır adres turudur.

Name Servers(NS)

Ters DNS kayıtları(PTR):DNS üzerinde bulunan Ip adresine atanmış makine adı

Takma İsimler(CNAME)

eposta sunucusundan mail yollama kısmını biz ayarlıyoruz.biz eve bu sunucuyu kurup comuden mail yolluyormus gibi gönderebiliriz ama karşıdaki e posta sunucusu maili kullanıcıya sunmadan önce ters dns kaydına da bakıyor.

Ters dns kaydı bizim sunucu IP adresi karşı tarafa gözüküyor.Bunu engelleyemeyiz.Çünkü mail göndermek için kullanılan protokol SNTP ve sntp de TCP kullanan bir protokoldür.Bu nedir? veri gönderip almayı başlatmadan önce 3lü el sıkışması yapmak gerekiyordu.3lü el sıkışması için de karşı tarafa veri gönderiyorum sonra karşı taraf ilişkili bi veri ben de ilişkili bir veri yolluyom.

tcp kullanıyorsan ipmizi gizleyemeyiz.

YEREL ADRES ÇÖZÜMLEMESİ

* /etc/hosts dosyası
* localhost ip adresi olarak 127.0.0.1 olarak bize döndürüyor.
* 10.10.10.42 [www.youtube.com](http://www.youtube.com) biz de yazabiliriz.YASAKLANMIŞTI YOUTUBE MESELA.Youtube ip adresi yerine teleküminasyın yasak adresi döndürülüyor.

DİG:

Dig(domain information groper)Linux sistemlerde **DNS** kayıtlarına bakmak için kullanılır. **DNS** adreslerine bakmak istediğiniz siteyi dig komutu ile aratma yapmanız mümkün ve sitede kullanılan tüm DNS kayıtları karşınıza çıkacaktır.

WHOIS:

DNS sunucusuna nasıl güveneceğiz?

* DNS, UDP kullanan güvensiz bir servistir.
* cevabın bizim sorgu gönderdiğimiz sunucudan geldiğinden emin olamayız.
  + DNS sunucusu bizimle aynı ağda ise MAC adresi spoof edilebilir,
  + Araya başkası girip(man-in-the-middle)DNS cevaplarını kendisi gönderebilir.

DNS’ sorusundan beklentilerimiz

* DNS sunucusunun doğru cevap vermesi
* Cevabı bizim sorguyu gönderdiğimiz DNS sunucusunun vermesi

DNSSEC

(Domain Name System Security Extensions)

* DNS istemcilerine (çözümleyicilerine), DNS verilerinin köken

kimlik doğrulaması, kimlik doğrulaması reddi ve veri bütünlüğünü sağlayan,

ancak geçerlilik veya gizlilik sağlamayan bir DNS eklentisidir.

* DNSSEC, sahte veya manipüle edilmiş DNS verilerinden (DNS önbellek

zehirlenmesi ile oluşturulan veriler gibi), uygulamaları korumak için (ve

bu uygulamalara hizmet eden çözümleyicileri önbelleğe almak)

kullanılacak şekilde tasarlanmıştır. DNSSEC korumasıyla gelen tüm

cevaplar dijital olarak imzalanmıştır. Bir DNS çözümleyici, dijital imzayı

kontrol ederek, bilgilerin alan sahibi tarafından yayınlanan ve yetkili bir

DNS sunucusunda sunulan bilgilerle aynı (yani; değiştirilmemiş ve tam)

olup olmadığını kontrol edebilir.

DNSSEC Nasıl Çalışır?

* DNSSEC'de her bölgenin ortak/özel bir anahtar çifti vardır. Bölgenin ortak anahtarı DNS kullanılarak

yayınlanır, özel anahtarı ise gizli kalır. Bir bölgenin özel anahtarı, o bölgenin bireysel DNS verilerini

işaretleyerek aynı zamanda DNS ile birlikte yayımlanan dijital imzaları oluşturur. DNSSEC katı bir

güvenlik modeli kullanır ve bu güven zinciri üst bölgeden alt bölgeye doğru gelişir. Yüksek düzeyli (üst)

bölgeler, düşük düzeyli (alt) bölgelerin ortak anahtarlarını imzalar veya onaylar. Bu bölgelerin yetkili

isim sunucuları; kayıt şirketleri, İSS'ler, web barındırma şirketleri veya web sitesi operatörleri (kayıt

sahipleri) tarafından yönetilebilir.

* Son kullanıcı bir web sitesine erişmek istediğinde, kullanıcının işletim sistemindeki çözümleyicisi, bir

İSS'de bulunan tekrarlamalı bir isim sunucusundan alan adı kaydını ister. Sunucu bu kaydı istedikten

sonra, bölgeyle ilgili olan DNSSEC anahtarını da ister. Bu anahtar, sunucunun aldığı bilgilerin yetkili

isim sunucusundaki kayda benzer olduğunu doğrulamasını sağlar.

* Tekrarlamalı isim sunucusu, adres kaydının yetkili isim sunucusu tarafından gönderildiğini ve aktarım

sırasında değiştirilmediğini belirlerse, alan adını çözümler ve kullanıcı siteye erişebilir. Bu sürece

doğrulama adı verilir. Adres kaydı değiştirilmişse veya belirtilen kaynaktan değilse, tekrarlamalı isim

sunucusu kullanıcının sahte adrese erişimine izin vermez. DNSSEC aynı zamanda bir alan adının

mevcut olmadığını da ispatlayabilir.

DNS over TLS

TLS üzerinden DNS, Transport Layer Security (TLS)

protokolü aracılığıyla şifrelemek ve sarmak için bir güvenlik

protokolüdür. Yöntemin amacı, aradaki adam saldırıları

yoluyla DNS verilerinin gizlice dinlenmesini ve

değiştirilmesini önleyerek kullanıcı gizliliğini ve güvenliğini

artırmaktır.

DoH (DNS over HTTPS)

* DoH, HTTPS protokolü aracılığıyla uzak DNS çözümlemesi

gerçekleştirmek için kullanılan bir protokoldür.

* Yöntemin amacı, DoH istemcisi ile DoH tabanlı DNS

çözümleyicisi arasındaki verileri şifrelemek için HTTPS

protokolünü kullanarak, aradaki adam saldırılarıyla DNS

verilerinin gizlice dinlenmesini ve değiştirilmesini önleyerek

kullanıcı gizliliğini ve güvenliğini artırmaktır.

* Şifreleme tek başına gizliliği korumaz, şifreleme sadece

verileri gizlemek için bir yöntemdir.

DNS sorgularını gizlemeye gerek var mı?

* Her web isteğinin başlangıcında bir DNS sorgusu yer alır.
* Bu sorgunun yanıtı alındıktan sonra erişim IP adresi ile

başlatılır.

* Bu IP adresiyle bağlantı kurduktan sonra erişmek

istediğimiz sunucu adını ve dosya adını HTTPS ile

şifrelediğimizden başkalarına görünemez durumdadır.

* Hangi alan adına eriştiği bilgisi de kullanıcının mahremiyeti

açışından gizli tutulması gereken bir bilgidir.

**Dns flood** : Saldırıları hedefteki DNS sunucuya normal isteklere cevap veremeyecek duruma gelinceye kadar DNS isteği gönderilmesi ya da sunucunun önündeki Firewall/IPS'in "session" limitlerini zorlayarak Firewall arkasındaki tüm sistemlerin erişilemeyecek duruma getirilmesi ile gerçekleştirilir. Bu saldırı tiplerinde çok sayıda DNS sorgusu yapılması gereklidir.Saldırıların gerçekleştirilmesi sırasında genellikle sahte Ip adresleri kulllanılır.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*9.DERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*06.05.2021\*\*\*\*\*\*\*\*\*

**YÖNLENDİRME PROTOKOLLERİ**

* Her ağda bir tane yönlendirici bulunur.
* Bu yönlendiriciler bizim göndermek istediğimiz hedef IP adresi bizimle aynı ağda değilse o yönlendiriciye kadar ulaştırma işini yapması için paketi doğrudan yönlendiriciye yönlendiriyorum
* Eğer hedef IP adresi benimle aynı ağda ise o zaman onun MAC adresini öğrenip doğrudan ona yönlendiriyod<um.
* Eğer iki bilgisayar birbiri ile haberleşecek ise ve ikisi de aynı ağda ise o zaman birinden diğerine giden paket yönlendirici üzerinde hiç geçmez.Yönlendirici ile ilgili bir işlem yapılmaz yani orada.
* O yüzden yerel ağın içerisindeki iki bilgisayarın birbiri ile haberleşmesi ile ilgili bir kural yazacaksak onu firewall üzerinden yönlendirici üzerinden yapamayız yani.
* Firewall üzerine kural yazarak ancak dışarıdan gelen isteklere engel olabiliriz.Çünkü bizim ağımızın içerisine geçebilmesi için bir paketin mutlaka yönlendiriciden geçmesi gerekir ki iç ağa girebilsin.
  + Ama yerel ağdaki makineler birbirleri ile haberleşirken eğer aynı ağdalar ise IP adresini MAC adresine dönüştürüp doğrudan onunla iletişime geçiyor.
* Peki aynı ağda olup olmadıklarını nasıl anlarız?
  + Kendi IP adresimize bakıp ağ maskesini kullanıyor idik. Ağ maskesi ile IP adresimizi bitwise and işlemine tabi tutuyorduk ve bulunduğumuz ağı tespit ediyoruz, sonra hedefteki IP adresi ile kendi ağ maskemizi aynı işleme tabii tutup hedefteki IP adresinin bulunduğu ağı tespit ediyoruz. Evet ise paketi yönlendiriciye göndermeden direkt hedefe gönderiyoruz.

route -n dediğimizde

destination 0.0.0.0 her ne olursa olsun demek istiyoruz.Tek bir bilgisayarı değil de

Paketler benim ağımdan dışarı nereye giterse gitsin gatewaye routera ver demiş oluyorum.

**STATİK YÖNLENDİRME**

* Yönlendirme tablosunun kalıcı olarak elle (manually) oluşturulmasıdır.
* Oraya biz kuralları yazıyoruz.
* Şu IP adres bloklarına gönderiliyorsa veri onları şu ethernet arayüzüne gönder
* Rotolar(yollar) sabittir ve yönlendiricilerden alınan bilgilerle güncellenmez.
* Bu tip bir yönlendirme ağdaki herhangi bir problemden etkilenmiyor.Niye? Zaten hiç kimseye bir şey sormuyoruz biz yani bu ağda yoğun bir trafik mi var karşı sunucu ulaşılabilir mi vs. Biz bunlarla ilgilenmiyoruz.
* Diğer yönlendiricilerle bir keşfetme işlemi yapılmadığından yönlendirici üzerinde çok az CPU ve ağ trafiği yükü oluşturur.
  + Karşıya göndermek istediğimiz trafiği azaltmıyor.Yönlendiriciler arasında ta amerikadan türkiyeye gelirken ki bu trafiği nasıl yöneticeği ile ilgili ayrıca bir yönlendiriciler arasında birbirlerine sorulup alınan bir ağ trafiği oluşturmuyor.
* Bunları sorup öğrenmediği için bunlar üzerindeki değişiklikten de etkilenmiyor.Fakat bu her zaman iyi bir şey değil çünkü gönderilip alınan trafikte bir bağlantı kopmuş olabilir benim elle yazdığım kuralı tekrar değiştirmem gerekir.
* İnsan hatasına açıktır.
* Hata toleransı içermez.Ağda bir değişiklik olduğunda veya statik yönlendirilmiş aygıtlar arasında bir problem olduğunda yeniden yönlendirme yapılamaz
* Yönetimi ve bakımı zahmetlidir. Yeni yönlendirici eklendiğinde veya değişiklik olduğunda yapılandırmayı yönetici elle yapmak zorunda kalır.

DİNAMİK YÖNLENDİRME

* Paketler için her seferinde farklı bir rotanın uygulanabilmesi demektir.

RIP

OSPF

BGP

RIP(routing Information Protocol)

* Ben iyi rota seçimi yaparken tek kriter olarak düğüm sayısı alınır.
* En fazla 15 düğüm kullanılır.Bu sayede yönlendirme döngüleri(loop) önlenmiş olur.
  + Saldırganlar looplara dönebilecek paketler gönderip bu bant genişliğimizi istemediğimiz şekilde harcamasınlar diye de yapılır.
* 16’dan sonra *destination unreachable* hatası döner.
* Yönlendirici bu protokolde kendisine bağlı olan ağları 30 sn’de bir komşu yönlendiricilere bildirir.Bu sayede bir yönlendirme tablosu tutulmuş oluyor.
* Taşıma protokolü olarak UDP kullanılır.
* 520 numaralı port RIP için ayrılmıştır.
* Diğer protokoller gibi parametre almadığından yapılandırılması kolaydır.

OSPF(Open Shortest Path First)

* Dijkstra algoritmasının gerçekleştirilmesidir.
* Önce en kısa yolu kullanan ve protokol olarakta IP kullanan bir yönlendirme algoritmasıdır.
* Bir yönlendirici ağın tüm topolojik haritasını oluşturur.
* Yönlendirici daha sonra yerel olarak tüm ağlara en kısa yol ağacını elde etmek için Dijkstra’nın en kısa yol algoritmasını kullanır, kendisini bu ağaçta kök olarak belirler.
* Yönlendiriciler arasında özel bağlantı maliyetleri ağ yöneticisi tarafından ayarlanır.

Gezgin satıcı problemi

Gezgin satıcı problemi şu şekilde tanımlanabilir:

* Bir seyyar satıcı var;
* Bu satıcı, mallarını n şehirde satmak istiyor;
* Öte yandan, mantıklı bir şekilde, bu satıcı bu şehirleri

mümkün olan en kısa şekilde ve her bir şehre maksimum bir

kere uğrayarak turlamak istiyor.

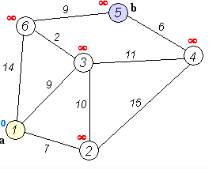
Problemin amacı, satıcıya bu en kısa yolu sunabilmektir.

Gezgin satıcı problemi

Satıcının n! değişik tur arasından seçim hakkı olacaktır.

Bu, 100 şehirlik bir tur için bile 9,33 \* 10^157 değişik

rota anlamına gelir.



BGP (Border Gateway Protocol)

* Taşıma protokolü olarak TCP (port 179) kullanır.Biz portu değiştirebiliriz ama bize gönderenlerin de bundan haberdar olması gerekir vsvs.
* Sınır Geçit Protokolü gerçekten de güvenilir, sağlıklı

çalışan bir yönlendirme protokolü olmasına karşın yavaş

bir yönlendirme protokolüdür.

* BGP ile haberleşmek için bir yönlendirici 3 tablo tutar.

1)Komşu Tablosu: Bu tabloya manuel olarak komşu

yönlendiriciler yazılır. Ayrıca hangi bilgiyi hangi yönlendiriciye

gönderebileceği bilgisi de bu tabloda tutulabilir.

2)BGP Tablosu: Dakikada bir komşu yönlendiricilerinden

ayaktayım-çalışıyorum(up and running) bilgisi alıp almadığını tutar.

3)IP Yönlendirme Tablosu: BGP tablosundaki en iyi rotaları

tutar.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*16.05.2021\*\*\*\*\*\*\*\*\*\*\*\*\*\*11.DERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HTTP - HTTPS

* Hypertext Transfer Protocol
* Hypertext Transfer Protocol Secure
* TSP’nin 80. portunu kullanıyor (HTTP)
* 443. portunu kullanıyor.(HTTPS)
* Farklı portlar ile de çalışabilir fakat bunu istemciye de haber vermek gerekir.

HTTP kodları:

* 1xx - Bilgilendirme yanıtları
* 2xx - Başarı yanıtları
* 3xx - Yönlendirme mesajları
* 4xx - İstemci hatası yanıtları
* 5xx - Sunucu hatası yanıtları

Telnet: Bir makineye HTTP ile bağlanmak istediğimizde bunu sadece tarayıcı ile yapmamız gerekmez.TELNET komutu ile Uzaktaki bir makine üzerinde oturum açmak için kullanılıyor.

* Şimdilerde onun yerine SSH kullanılıyor.
* Telnette karşıya gönderip aldığımız metinler düz metin olarak olurken SSH şifreli şekilde bunu sağlayan bir protokoldür.
* Sadece karşı tarafta oturum açmak için kullanılmıyor, bir port üzerinde işlem yapmak içinde kullanılıyor.Bir makinenin 80. portu açıksa o portu dinleyen bir sunucu servisinin çalışıyor olması demektir.
* Bir portun açık olması o portu dinleyen bir sunucu servisinin çalışıyor olması demek karşı tarafta bir makine üzerinde.
* HTTP de HTTPS de verileri düz metin olarak karşıya gönderiyoruz, sunucu da aldığı isteğe cevabı bize düz metin olarak gönderiyor. Bu ne demek? Araya giren herhangi birisi gönderip aldığımız tüm verileri okuyabilir.
* Bizim bağlantı isteğimizin geçtiği bütün nodelar anlamına geliyor.Bu ne demek?
  + Eğer biz sunucu ile aynı ağın içerisinde değilsek isteğimizi ağımızdaki yönlendiriciye gönderiyoruz.
  + Yönlendirici yönlendirme algoritmalarını kullanarak bunu karşı ağdaki sunucuya kadar ulaştırıyor.Demek ki gönderdiğim tüm istekler araya biri girmese daha yönlendiriciler üzerinden geçiyor ve yönlendiriciler üzerinden yapılan bir sorgulama ile bizim hangi web adreslerine nasıl istekler gönderip yanıtlar aldığımız sorgulanabilir.
  + HTTPSde veri uçtan uca şifrelendiği için araya giren bir 3. taraf ya da yönlendiriciler (ISP-Internet Service Provider) deşifreleyemezler.

HTTPS

* Gönderip aldığınız veriyi araya giren birinin anlam çıkaramayacağı şekilde şifrelediği için değiştirilmesinin de önüne geçer. Bu sayede HTTPS kullanarak gönderdiğimiz kullanıcı adı ve parolamız, kredi kartı bilgilerimiz ve hatta ulaştığımız sayfanın adresi trafiğimizi dinleyenlerce görülemez.
* İki temel hedefi vardır:
  + Ulaştığımız adresin gerçekte ulaşmaya çalıştığımız adres olduğundan emin olunmasıü
  + Sunucu ve istemci arasındaki trafiğin sadece bu iki taraf tarafından çözümlenebilecek şekilde şifrelenmesi
* Bu hedeflere ulaşabilmek için ilk yapılacak şey TLS bağlantısının kurulmasıdır. HTTP için kullandığımız istemci(çoğu durumda tarayıcımız) TLS bağlantısının kurulması için de aracılık eder.
* Taraflar aralarında aşağıdaki gibi anlaşırlar:
  + İstemci sunucuya bir “client hello” mesajı gönderir. Bu mesajda istemcinin desteklediği protokolün sürümü ve tercih ettiği kriptografik algoritmaların listesi bulunur.
  + İlk mesajı alan sunucu istemciye bir “server hello” mesajı gönderir.Bu mesajda ise istemciden aldığı listeden seçilen kriptografik algoritma ve oturum ID’si bulunur. Sunucu ayrıca sayısal sertifikasını da gönderir. Sunucunun istemciyi doğrulaması gereken durumlarda bir de “client certificate request” mesajı gönderilir ama biz bunun gerekmediğini varsayacağız.
* Github.com adresine github.com yazarak bağlanamam.Bu adresin önce bir IP adresine dönüştürülmesi lazım.Bunun için önce bir DNS sorgusu yapılması lazım.Daha sonra bu makine adına karşılık gelen IP yi öğrenip karşıya bağlanacağız.
* Bu aşamadan sonrasına devam edilebilmesi için istemcinin sunucu sertifikasını onaylaması gerekir.Bir sunucu sertifikası aşağıdaki bilgileri içerir.
  + Sertifikanın sahibi,
  + Sertifikanın geçerli olduğu alan veya makine adı,
  + Sunucunun açık anahtarı,
  + Sertifikanın geçerli olduğu tarih aralığı,
  + Sertifikanın sayısal imzası.
* Bu sertifika eğer istemcinin güvendiği bir Sertifika Otoritesi(CA) tarafından üretilmiş ise veya bu aşamada istemci sertifikayı güvenli bulursa sertifikanın onaylanması adımı geçilmiş olur.Aksi durumda iletişime devam edilmez.Tarayıcılarınızın bizi “bağlantı güvenli değil” diyerek uyardığı sayfaları gördüğümüz nokta burasıdır.
* İstemcinin hangi sertifika otoritelerine güvendiğini görebilmek kritik öneme sahiptir.
* İstersem güvensiz buduğum sertifikaları kaldırıp kendimiz ekleyebililiriz.Ama eğer kendimiz yapmadıysak kimsenin sertifikasına güvenip almak ço önerilmez.
* Kaynak kodu açık firefoxta bize göstewrdi
* ği sertifika oteritelerinden başkalarını kabul edip etmediğiniz görebilirizi ama yazılımın kaynak kodu açık değilse sadece onun söylediğine güvenmemiz gerekir.Bütün mekanizmayı da sertifika otoriteleri üzerine kurmuşken neye güvenip güvenmediğini bilmediğimiz tarayıcıları kullanmamız gerekir.
* Henüz sunucu ve istemci arasında yapılmasını planladığımız trafiğin başlamadığını unutmadan devam edelim.Bulunduğumuz durumda istemci sunucudan alığı sertifikaya güvenebileceğini doğruladı ama sunucunun bu sertifikanın gerçek sahibi olduğunu henüz doğrulamış değil. Sunucu bu sertifikayı kendisiyle iletişime geçen herkese gönderdiği için bu sertifikayı ele geçirmiş herhangi bir taraf bizi yanıltıyor olabileceğinden bunu da doğrulaması gerekiyor.Bunun doğrulanması da sertifika içinde bulunan sunucunun açık anahtarıyla gerçekleştiriliyor.İstemci bu açık anahtarı kullanarak sunucuya bir veri gönderdiğinde ancak sunucu o açık anahtarla ilişkili gizli anahtara sahipse veriyi veriyi deşifreleyebilecektir.
  + Sertifikayı aldık vs fakat güvenlikle ilgili her şeyi bitirmiş değiliz burada bizim yapmaya çalıştığımız şey,
  + Burada sunucu ile istemcimiz arasında uçtan uca şifreli sadece sunucu ile bizim anladığımız veriyi gönderip almak.
  + Fakat sunucu bize herkese gönderdiği bir veri gönderdi.
  + Sadece buna güvenerek güvenlik aşaması geçilmiş olmuyor.Sertifika içinde sunucunun açık anahtarı var.Bu anahtarı kullanarak şifreleyerek sadece sunucunun açacağı veriyi gönderebilirim.
* Artık istemci ve sunucu bu açık anahtarı ve bir anahtar değişim algoritmasını[2] kullanarak oturum boyunca verileri şifreli gönderip almak için bir ortak anahtar oluşturabilirler.
* Simetrik şifrelemede kullanılacak anahtar oluşturulduktan sonra istemci sunucuya bir “finished” mesajı bu anahtarla şifreleyerek gönderir. Bu noktadan sonra taraftar aralarında simetrik şifreleme algoritmalarından birini kullanılırlar.
* Mesajı alan sunucu da aynı anahtarla şifreletek “finished” mesajını istemciye iletir.Böylece her iki taraf da şifrelemede kullanacakları bir ortak anahtar üzerinde anlaşmış ve birbirlerini doğrulamış olurlar.
* Bu aşamadan sonra taraflar arasındaki iletişim standart HTTP de olduğu gibi yapılacaktır ama arada gidip gelen bütün veri şifrelendiği için HTTP de olduğu gibi üçüncü taraflara değil verinin görünmesi, ulaştığımız adresi bile görünmesi mümkün olmayacaktır.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*20.05.2021\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*12.Hafta\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AĞ ANALİZİ

* Bilgisayarınıza gelen ve giden ağ verilerini analiz etmek için çok sayıda yazılım kullanılmaktadır. **tcpdump** bunlar arasında en yaygın olarak kullanılanlardan biridir.
  + Her işletim sisteminde çalışan hali mevcuttur.
  + Çok sayıda arayüzü de mevcuttur.
  + BSD lisansına sahip bir özgür yazılımdır.

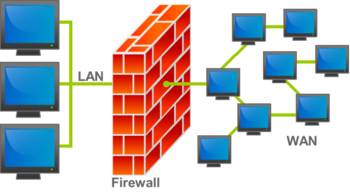
AĞ AYGITLARININ TESPİTİ

* **ip** komutu ile
  + Önceden ifconfig kullanılırdı.
* **nmcli** komutu ile
  + nmcli device status
  + nmcli device show
* **netstat** komutu ile (önmli bir komut kullanmayı öğren)

tcpdump KULLANIMI

* Bilgisayarımızda gelen giden veriyi analiz etmeliyiz ama neden uygulama seviyesinde yapmıyoruz?ÇNK örneğin 80.porta bağlı olmasa bile bir uygulama istek geliyor mu gelmiyor mu onu anlamak için kullanılıyor ağ analiz aygıtları ya da makinemize gelen ve cevabını herhangi bir uygulamanın değil iş. sis. verdiği icmp mesajlarını görmek isteriz.
* sudo tcpdump dst port 22 komutu
  + Bilisayarımıza gelen ssh paketlerini görüntülemiş oluruz.Bu verilerden bir şey anlamayız cnk şifreli.
* Parametreler birbirine and, or, not ile bağlanabilir.
  + sudo tcpdump dst port 42 and udp and ip6
* Çıktılar bir dosyaya yazılabilir.
  + sudo tcpdump port 42 -w capture\_file
* Kaydedilen çıktılar sonradan tcpdump ile okunabilir.
  + tcpdump -r capture\_file
* sudo tcpdump -i wls1 -tttt -c 5
  + 5 paketi yakala dur diyebiliriz.Paket sayısını biz vermezsek biz durdurana kadar çalışır ya da disk dolana kadar.
* İlgili portunu söyleyerek trafiği ayırt edebiliriz
  + DNS trafiğini ayırt etmek
  + sudo tcpdump -vVAs0 port 53

GÜVENLİK DUVARI

* Güvenlik duvarı, önceden belirlenmiş kuralları uygulayarak, ağa gelen ve ağdan dışarı çıkan trafiği gözleyen(monitoring) ve kontrol eden güvenlik sistemidir.
* Güvenlik duvarları güvenilen bir iç ağ ve güvenilmeyen dış ağ(internet) arasında bir bariyer oluştururlar.
* Ağ Güvenlik Duvarı
  + İki ya da daha fazla ağ arasındaki trafiği kontrol eder ve süzer(filter)
* Bilgisayar için güvenlik duvarı
  + Bilgisayara gelen ve giden ağ trafiğini kontrol eder.

iptables

* Linux çekirdeğinin 2.4 sürümünden bu yana kullanılmaktadır.
* Öncesinde ipchains kullanılmaktaydı.
* Linux çekirdeğinin geliştirdiği yeni güvenlik duvarı uygulaması nftables’dır.
* ACCEPT
* REJECT(Paketi düşürür ve bu durumu geri bildirir.)
* DROP(Paketi düşürür ve bildirim göndermez)
  + Mevcut kuralları listelemek:
    - sudo iptables -L -n -v
    - L: list rules
    - -v : Display detailed information
  + Gelen trafiği engellemek:
    - sudo iptables -P INPUR
    - sudo iptables -P FORWARS **DROP**
    - sudo iptables -P OUTPUT **ACCEPT**
  + Varsayılan kural atamak
    - sudo iptables -P OUTPUT ACCEPT
  + Bütün kuralları sıfırlamak
    - sudo iptables -F
  + Sadece belirli bir ağdan/adresten gelen trafiği engellemek
    - sudo iptables -A INPUT -i eth0 -s 10.0.0.0/8 -j DROP
    - sudo iptables -A INPUT -s 1.2.3.4 -j DROP
  + Belirli ağdan giden trafiği engellemek
    - sudo iptables -A OUTPUT -d 75.126.153.206 -j DROP
    - sudo iptables -A OUTPUT -d 192.168.1.0/24. -j DROP
  + Belirli bir porta gelen istekleri engellemek
    - sudo iptables -A INPUT -p tcp --dport 80 -j DROP
    - sudo iptables -A INPUT -p tcp -s 1.2.3.4 --dport 80 -j DROP
  + facebook.com’a giden trafiğe engel olmak
    - # iptables -A OUTPUT -p tcp -d facebook.com -j DROP
  + Bu işlem dns sorgusu gerektireceğinden şöyle de yapılabilir;
    - # host -t a [www.facebook.com](http://www.facebook.com)
    - [www.facebook.com](http://www.facebook.com) is an alias for star-mini.c10r.facebook.com.
    - star-mini.c10r.facebook.com has address 157.240.9.35
    - #whois 157.240.9.35 grep CIDR
    - CIDR: 157.240.0.0/16
    - # iptables -A OUTPUT -p tcp -d 157.240.0.0/16 -j DROP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*13.DERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*27.05.2021\*\*\*\*\*\*\*\*\*\*\*\***Güvenli İletişiminde Sertifika Onayı**

* HTTPS kullanıldığında sunucu sertifikası istemci tarafından doğrulanır.
* İstemciler(çoğunlukla tarayıcılar) güvenilen sertifika otoritelerinin (CA) bir listesini tutar.
  + Bu listeye ekleme yapılabilir.
  + Tarauoconon sadece bu CA’lara güvendiğinden nasıl emin olabiliriz?
* Sertifika ancak güveniken bir CA tarafından onaylanmışsa kabul edilir.
* Tarayıcının sadece bize gösterdiği sertifikalara gücendiğini başkalarına güvenmedidğini bizim bir şekilde doğrulamamız gerekir.

**SSH için sertifika onayı**

* HTTPSye çok benzer bir şekilde TLS bağlantısı yapıyor karşı taraf ile
* Karşı tarafta güvenli oturum açmayı sağlıyor.
* Her protokol SSHın içerisine tünellenebilir.Yani iki tarafta SSH bağlantısı yapıldıktan sonra aynı HTTPSde olduğu gibi uçtan uca veriler şifrelenip gönderilip alınır.İçerisinde ne gönderdiğimiz kısmı yönlendiriciden operatörden bağımsız bir şekilde görülemez.
* SSH istemcileri tarayıcılar gibi bu sertifikayı şu sertifika otoritesi onaylamış diye kontrol etmezler.Mekanizmaları böyle değildir.
* SSH için kullanılan sertifikalar ise hemen hemen her sunucu tarafında kendinden imzalı(self-signed)şeklinde üretilir.(TPS kullanır)
* HTTPSde de kullanılabilir fakat istenen bir şey değildir.SSHta ise başka çare yoktur.
* SSH bağlantısının bütün güvenliği bu sertifikanın güvenilir olması üzerine kurulu (bunu HTTPSde tarayıcımız yapıyor, SSHta ise bunu istemimizde yapamadığı için başka mekanizmalar bulmamız gerekir.)olduğundan istemcinin bu sertifikayı bir şekilde onaylaması gerekir.
* Karşı tarafın sertifikasının güvenilir olup olmadığını anlamak için;

**Sunucu anahtarının parmak izinin kontrolü**

* Bir SSH sunucusuna bağlanmak için ilk isteği yaptığımızda aşağıdaki gibi bir durumla karşılaşırız.
  + ssh scanme.nmap.org(komutunu yazınca)
    - *The authenticity of host ‘scanmeçnmap.org (45.33.32.156)’ can’t be established.*
    - *ECDSA key fingerprint is SHA256:8iz5L6iZxKJ6Y0Nmad4oMbC+m/+vI9vx5C5f+qTTGDc.*
    - *Are you sure you want to continue connecting (yes/no/[fingerprint])?*
* Karşı tarafın sertifikasını sunucu onaylamıyorsa bizim onaylamamız gerekir.
* Scanmenin parmak izi şudur diye söylüyor bunu nasıl onaylayabilriim.Biliyor muyum ki scanme.nmap.orgun sertifikasının parmak izinin ne olduğunu biliyor olmam gerekir ki ancak o zaman onaylayabileyim.
* Bu aşamada eğer sunucu anahtarının parmak izini bilmiyorsak onaylayabileceğimiz bir şey de yok demektir.”....” ifadesinin ezberlenmesi imkansız değilse de oldukça zordur.Not alınmış olmalıdır.

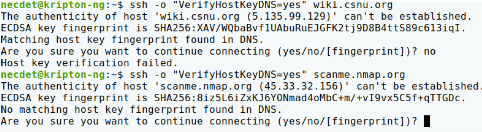
**Randomart Kullanmak**

* Sunucunun anahtarının parmak izi yerine ondan üretilmiş bir konsol görselini akılda tutmak daha kolaydır.Eğer SSH’ı “-o visualHostKey=yes” parametreiyle kullanırsak aşağıdaki gibi bir görsen görüp onu onaylayabilriiz.
* Neden sertifikaların parmak izini akılda tutmaya çalışıyoruz?

Çünkü bizim de yeni bir makineye geçtiğimiz zaman o sertifikayı onaylamamız gerekir.

Eğer bu ikisini de yapmıyorsak DNS kullanırız.

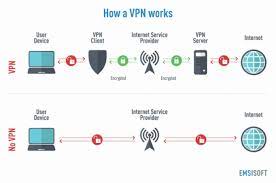
**DNS kullanmak**: Ip adreslerini makine adlarına , makine adlarını IP adreslerine çevirir.

* Makine adına karşılık gelen IP adresini nasıl DNS sunucuna sorabiliyorsak aynı şekilde sunucu anahtarının aprmak izini de sorabiliriz. Bu işlem “-oVerifyHostLeyDNS=yes” parametresiyle yapılır.Sunucuda böyle bir DNS kaydının girilip girilmemesine göre aşağıdaki gibi cevaplar.
* 
* Çok az sunucunun SSH kaydı DNS ile tutuluyor.
* DNS sunucusuna nasıl güveneceğiz?Hadi SSH sertifikasını tarayıcım onaylayamıyor ben onaylayayım dedim.
  + DNS, UDP53 kullanan güvensiz bir servistir.
  + Cevabın bizim sorgu gönderdiğimiz sunucudan geldiğinden emin olamayız.
    - DNS sunucu bizimle aynı ağda ise MAC adresi spoof edilebilir,
    - Araya başkası girip(man-in\*the-middle)DNS cevaplarnı kendisi gönderebilir.
    - DNS neden UDP kullanıyor?Çünkü çok basit küçük paketler yolluyoruz karşıya.
  + **DNS sorgusundan beklentiler**
    - DNS sunucusunun doğru cevap vermesi
    - Cevabı bizim sorguyu gönderdiğimiz DNS suncuusunun vermesi.
      * DNSe sorgu gönderdiğimizde bizxe cevap dönebiliyor ama cevabı bizim sorduğumuz dns sunucu değil kullandığımız operatörün dns sunucu verebiliyor.

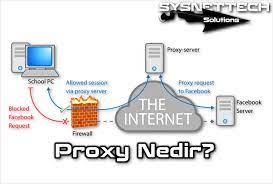
**BAĞLANTIYI YONLENDIRMEK ICIN 2 UYGULAMA**

* VPN
* Proxy

VPN

* Bizi bir takım engellemelerden kurtarıyor gibi dursada kendi içerisinde bir takım tehlikeler barındıran uygulamalardır.
* VPN sadece bir takım engellenmiş adresler var ve bunlara ulaşmak istiyoruz şeklinde kullanılmıyordur.
* Temel mantığı ; Biz isteklerimizi modemle servis sağlayıcıya gönderiyoruz o da internette nereye gönderiyorsa gönderiyor.Ama VPN kullandığımızda kendi makinemiz üzerinde VPN istemcisi kullanıyoruz.Biz trafiği VPN istemcimiz ile yapıyoruz.Örneğin bir tarayıcı çalıştırdığımızda o tarayıcı internette kendi başına istek yollamıyorda bu istek bizim VPN istemcimize gidiyor.
* Artık bizim bulunduğumuz ağın dışındaki VPN sunucusuna yapıyor ve bizim aramızdaki trafik uçtan uca şifrelenmiş oluyor.
* VPN ile sunucu arasındaki verileri herkes okuyamıyor Araya giren internet servis sağlayıcısı bile göremiyor. ama VPN sunucu sağlayıcılarda mı okuyamıyor?Her VPN sunucusuna güven olmaz!!!!11 Mutlaka bu servisi kimin çalıştırdığından haberdar olmalıyız.

**PROXY**



* Mantık olarak VPNe çok benziyor.Ama proxy için ayrı bir istemci çalıştırılmıyor.
* Kullandığımız araçlar her neyse onlar proxy bağlantısını gerçekleştirebiliyorlar.( Tarayıcı, FTP kullanacaksan onlar kendi içlerinde proxy ayarlarını barındırıyorlar
* Burada örneğin bir yere ulaşmak istediğimizde bulunduğumuz ağdaki firewall verinin bizim gödnerdiğimiz adrese gitmesine engel oluyor olabilir.Ya da bir VPN sunucu kurup VPN istemcisine yönlendirebiliriz trafiği. Ve firewalla takılmadan geçer.Çünkü veriler şifrelenip alındığı için firewall nasıl bir trafik olduğunu göremediği için bizim trafiğimizin nereyi hedeflediğini anlayıp kısıtlama uygulayamaz.
* Proxy bizim ağımızın içerisine de kurulabilir illa kısıtlamalardan kaçınmak için kullanmamıza gerek yok.
* Fakat ağımızın içerisindeyken hiçbir kısıtlamadan kurtulamayız.

NAT → Network address translation

Yerel ağda istemcilerimize elimizde eğer yeterince kendi başına internete çıkabilecek IP adresi yoksa ayrılmış IP adresleri gruplarınıni istemcilerimize dağıtıyoruz.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*14.DERS\*\*\*\*\*\*\*\*\*\*03.06.2021\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* Önce bilgisayarımızı açtığımızda bilgisayarın bir ağ yapılandırmasına ihtiyacı var.
* Bu ağ yapılandırması için iki farklı seçenek vardır.
  + 1-)Ağ yapılandırmasının elle yazılması.
  + 2-)Ya da ağ yapılandırmasını otomatik dağıtan bir sunucu varsa o da DHCP sunucusu idi. O otomatik olarak gönderebilir.
    - Bunu gönderebilmesi için sunucunun bizim önce Discovey paketi yollamamız lazım bulunduğumuz ağa. Yani bu ağda bir DHCP sunucusu var mı diye?
      * Ama bunu nereye göndereceğiz?Bulunduğumuz ağdaki sunucunun adresini bilmiyoruz. Hem bilsek dahi ondan nasıl cevap alacağız. Bilmiyoruz çünkü henüz bir ağ yapılandırması yok.
      * Hangi ağda olduğumuzu bilmediğimiz için broadcast adresine de gönderemeyiz.
* discovery(255.255.255.255)(Bu da bir broadcast ama sadece bulunduğum ağdaki kısıtlı iletileilecek bir adrestir.Bu discovery paketini gönderdiğimizde ağda bir sunucu var mı diye sormuş oluyoruz tüm ağa.)
  + offer
    - DHCP sunucusu da bu paketi alınca bize bir offer paketi gönderiyor.Burada bir DHCP sunucu var diye.
* request
  + ack
    - Aldığımız cevapta bizim hangi IP adresini hangi ağ maskesi aldığımızı, yönlendiricimzin adresini ve DNS sunucusunun adresini min. olarak barındırıyor.

***SORU:*** Peki bizim IP adresimiz yok LAN’a bize nasıl oluyorda DCP sunucusu cevap döndürüyor? Sonucta makinelerin birbirleri ile konuşabilmesi için bir IP adresi yapılandırmadına ihtiyacı var. Hadi biz bunu broadcast gödnerdik cevapta mı broadcast gönderiyor.? Eğer birden çok makine talepte bulunduysa broadcast geşen cevabın kendisine ait olduğunu biliyor?

* İkinci katman adreslemesi ile yani MAC adresi ile alıyor çünkü verileri yerel ağda iki makine birbirine MAC adresini kullanarakta adresleiyip gönderebiliyor. Buradan karşı taraftan gelen DCP paketinin içindeki veri ilk başta isteği yapan bilgisayara ulaşıyor.Bunları ne ile hangi protokolü kullanrak gönderiyor?

**UDP(istemci portu 68), (sunucu portu 67)**

**DHCP (Karşı taraftan aldık)**

* IP adresi
* Ağ maskesi
* Yönlendirici Ip adresi
* DNS suncuusu IP adresi
* Sunucunun gönderdiği diğer bilgiler.

ıp ve ağ maskesine neden ihtiyacım var. Çünkü ağ maskesini bildiğim zaman kendi IP adresimi de biliyorum. O zaman bulunduğum ağın ne olduğunu hesaplayabilirim.Hedefteki bir makineye ulaşmak istiyorsam o benmle aynı ağda mı değil mi bunu ayırt edebilirim.Eğer benimle aynı ağda değilse bu paketleri yönlendiriciye göndermem lazım. O zaman yönlendiricinin de IP adresine ihtiyacım var ve ulaşmak istediğim makinelerin hemen hepsine IP adresi ile değil de makine adı ile ulaşmak istiyorum.Çünkü o adreslerini biliyorum. O zaman bu dönüşümü yapabilecek olan DNS sunucunun IP adresine de ihtiyacım var.

**MAKİNE ADI ÇÖZÜMLEMESİ**

* Cache (Ön bellekte var mı?)(Daha önce sorgulamış mı)
* hosts dosyası
* DNS suncusu

Makine adının bir şekilde IP adresine dönüştürülmesi lazım. Çünkü gördük ki 3 ya da 4. katmanda paket başlıklarında makine adı yazabileceğimiz bir başlık yok.

İş. Sis.leri bu adın çözülmesi için yukarıdaki adımları takip eder.

**DNS sunucusuna sormak için; Ip adresini DHCP ile almıştık.**

* DNS suncusu aynı ağda mı yoksa dışında mı onu tespit etmeliyiz.

LAN içinde/ dışında

**IP-Netmask**

**DNS SUNUCUSU**

* DNS sunucusu LAN içinde ise; Verileri LAN dan göndermem gerekir, yönlendiriciye gerek yok.O zaman ne yapacagım elimde DNS sunucusunun IP adresi var ve ona ulaşmak istiyorum. O zaman bu IP adresine karşılık gelen MAC adresini öğrenmem lazım. Bu işlemi nasıl yapıyoruz?;
  + ARP request
    - Kaynak IP adresim, kaynak MAC adresim, karşı tarafın ulaşmaya calıstığım hedef IP adresi vardır.Hedefteki MAC adresi kısmı boş tabi.
  + ARP replay
  + UDP 53 DNS sorgusu paketini gönderiyorum.:(DNS sorgularının tamamı UDP ile yapılıyor).
    - Bu ne demek ? Karşı taraf bu cevabı benim sorduğum makine göndermiyor olabilir.Biri UDP kullanıyorsa UDP aynı zamanda IP de kullanıyor 3. katman adreslemesi olarak ama source IP adresi olarak kendi IP adresini değil de DNS sunucununun IP adresini kullanıp bize cevap döndürüyor olabilir.

Aynı ağda değilsem

YÖNLENDİRİCİ

* Yönlendiricinin IP adresini biliyorum sadece. Onu nereden aldım? DHCP suncusuundan aldım.Eğer yönlendirici ile DHCP aynı makine değilse benim yönlendiriciye verileri ulaştırmak için onun MAC adresini de bilmem gerekir. Çünkü aynı ağda olduğum bir makine ile iletişime geçmek istiyorum.
* Yönlendiric ile benim aynı ağda olmamam gibi bir durum söz konusu olabilir mi?
  + Olamaz. Çünkü LAN dan dışarı çıkabilmemiz için mutlaka yönlendiriciye ulaşmamız lazım. Yönlendirici de ağın dışında olursa nasıl ona kadar ulaştıracagız paketleri?
  + Çünkü bir ağdan çıkan paketleri başka bir ağa ulaştıabilmek için izlenmesi gereken algoritmalar var ve bu algoritmaları da yalnızca yönlendiriciler çalıştırabilir. Bizim ağımızın çıkışında bir bilgisayar başka bir ağdaki yönlendirciye ulaşabiliyorsa zaten o bilgisayarda bir yönlendirici vazifesini görüyor demektir.
* Yönlendiricinin IP adresi için ARP request’e gerek var mı?