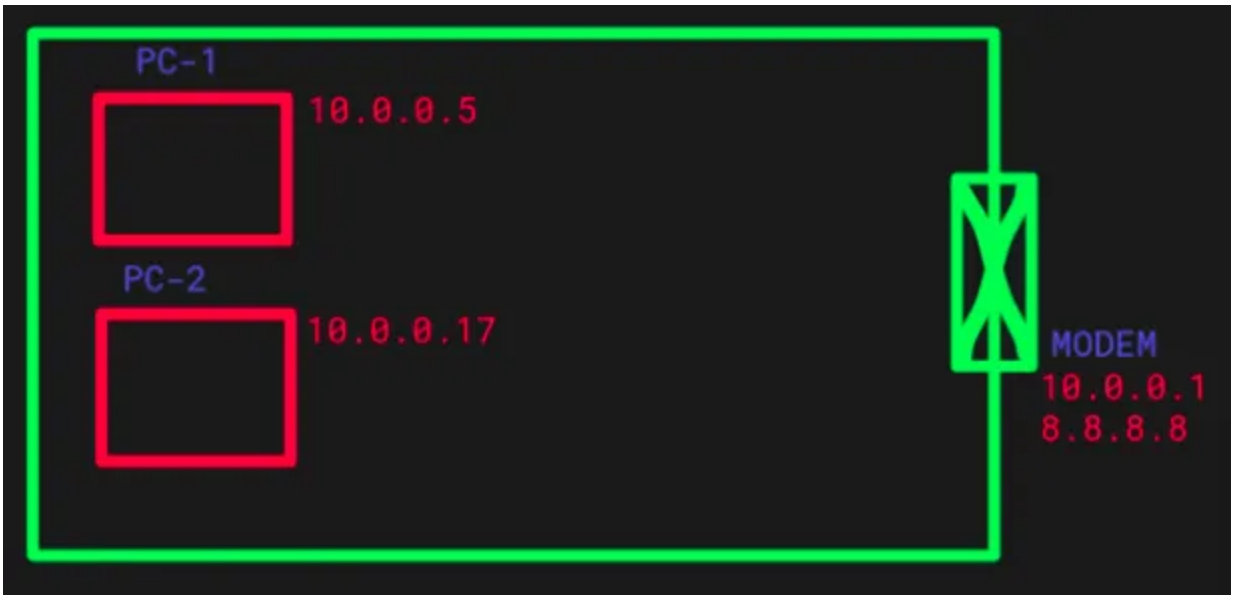
Bir Hacker'ın Gözünden Modern Web Nasıl Çalışır ? | MDISEC Neler Anlattı #5

İLKER YILMAZ
9 min read · 16 hours ago

Bu yazımızda da modern web'in nasıl çalıştığını, bilgisayarımızı açtığımızda nelerin meydana geldiğini ve daha fazlasını ele alacağız.

Günümüz Web dünyasına gelmeden önce temel konuları ele almakta fayda var. Evimizdeki yerel ağ yapısını şu şekilde düşünebiliriz;

Öncelikle bilgisayarımızı açtığımızda bir ip adresi alınmalıdır. IP adresi alırken de DHCP protokolü devreye girmektedir.

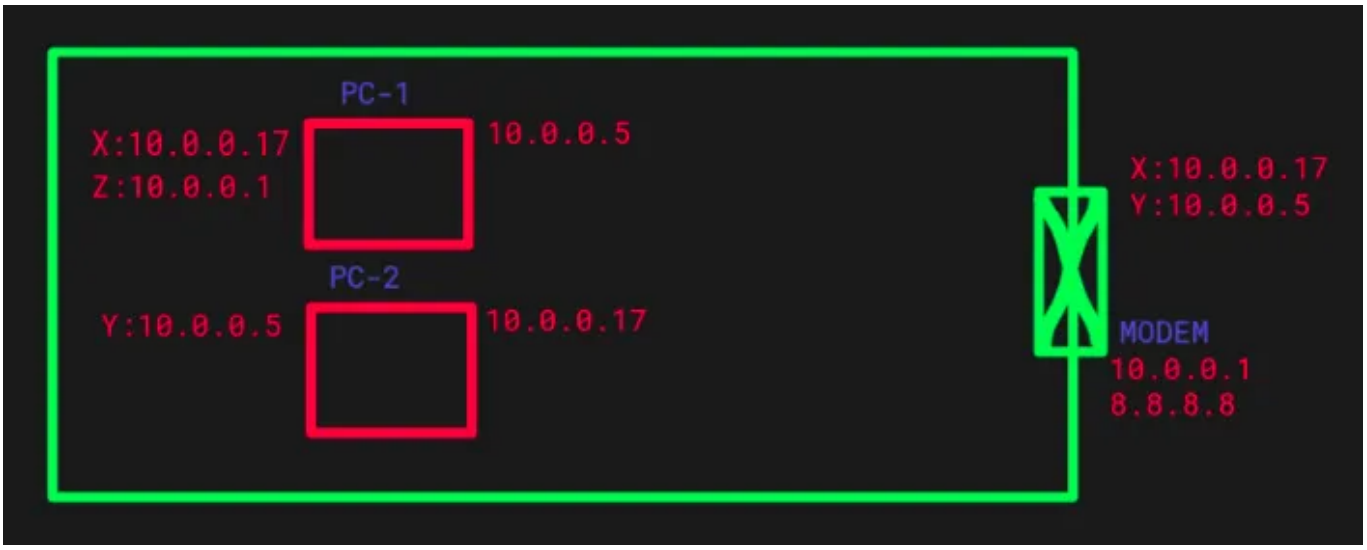


Burada görmüş olduğunuz düzen içerisinde PC-1 ve PC-2 veya PC-1 ve Modem iletişime geçeceği zaman ip adresi ile iletişim kurmadan önce layer-2'de yaşanan bir durum olmalı. Bu kısımda ARP (Address Resolution Protocol) devreye girmektedir. Yani burada iletişime geçecek olan cihazları düşünecek olursak, örneğin PC-1 ve PC-2 iletişim kurduğunda PC-1, PC-2'nin MAC adresini öğrenmelidir. Bu durumda bulunduğu ağ içerisinde bir duyuru mesajı yayınlayarak iletişime geçmek istediği kişiye mesajını iletmektedir. "Eeey 10.0.0.7 ben seninle konuşmak istiyorum" gibi bir mesaj yayınladığını düşünebiliriz.

ARP (Address Resolution Protocol)

ARP'ı genel olarak şöyle düşünebiliriz;
Şöyle bir durum hayal edelim. Örneğin içerisinde 200 kişinin olduğu bir sınıf düşünelim. Burada ismi "Ayşe" olan kişi ile iletişime geçmek istiyoruz. Fakat Ayşe'nin kim olduğunu bilmemekteyiz. O yüzden ayağa kalkıp "Eey Ayşe, Gel Beni Bul" deriz. 200 kişiden de adı Ayşe olan ayağa kalkıp der ki "Ben Ayşeyim Hadi Konuşalım". Temelde bunun canlandığını düşünelim. ARP temelde bu şekilde çalışmaktadır. Artık Ayşe'yi gördük ve tanıdık. Tekrar Ayşe ile konuşmak istediğimizde çağırmamıza gerek yoktur. O yüzden PC-1,

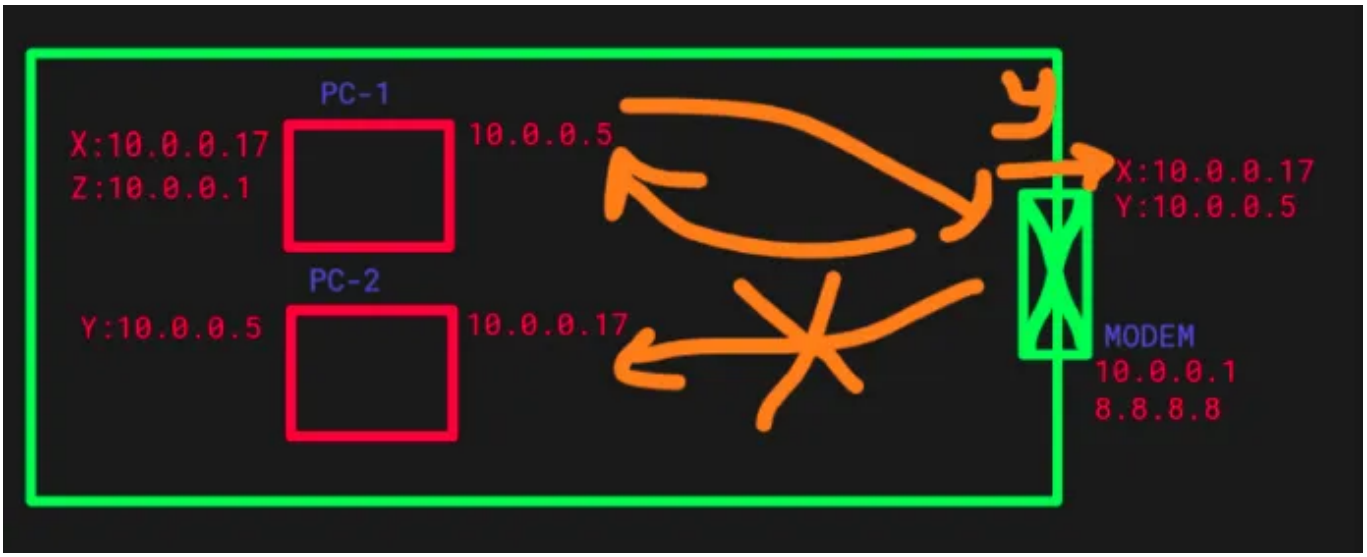
PC-2'nin MAC adresini öğrendiğinde kendi üzerinde bir ARP tablosu adı verilen yere yazar.



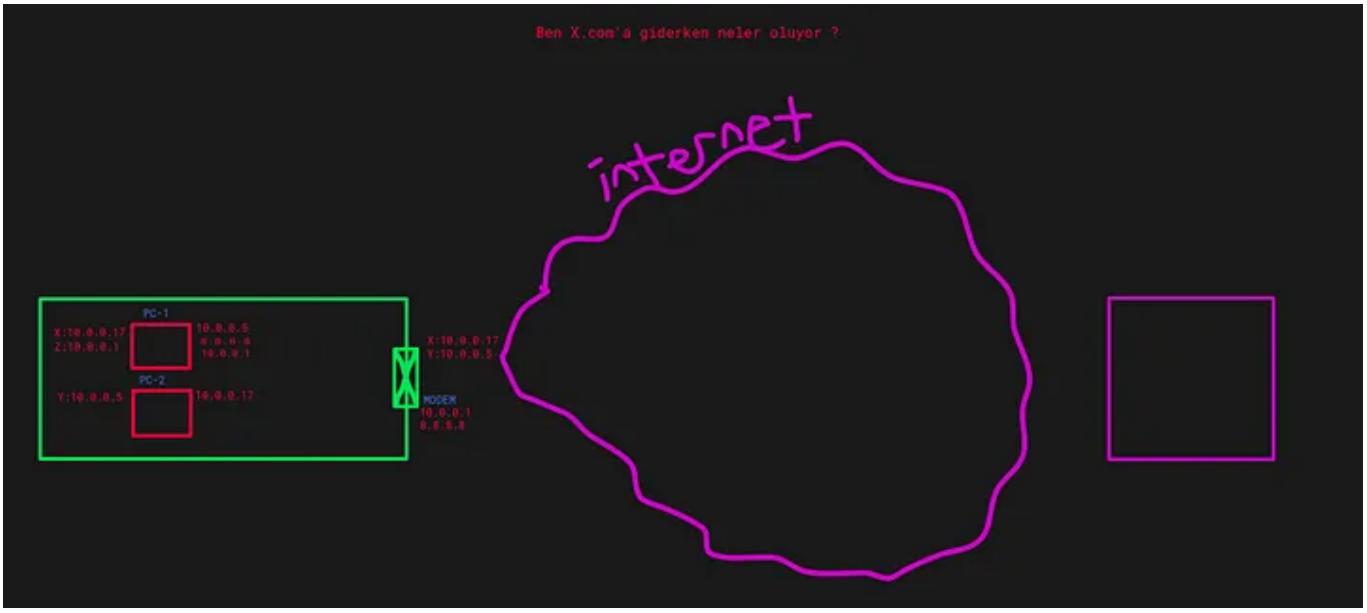
ARP Poisoning

Burada verdiğimiz örneğte 200 kişilik bir sınıfta "Ayşe Beni Bul" dediğimizde Ayşe'nin gelip bizi bulduğunu gördük. Fakat bu sınıfta herhangi bir igelip "Ben Ayşe'yim" dediğinde bunu doğrulayamıyoruz. Bu durum temel bir problem olarak karşımıza çıkmaktadır. O yüzden gerçekten Ayşe mi değil mi bilmiyoruz. ARP Poisoning dediğimiz olay da temelde bu durumdan dolayı meydana gelmektedir.

Burada eğer X değeri Y olarak güncellenirse artık gelen veriler PC-2'ye gitmek yerine PC-1'e gönderilecektir.



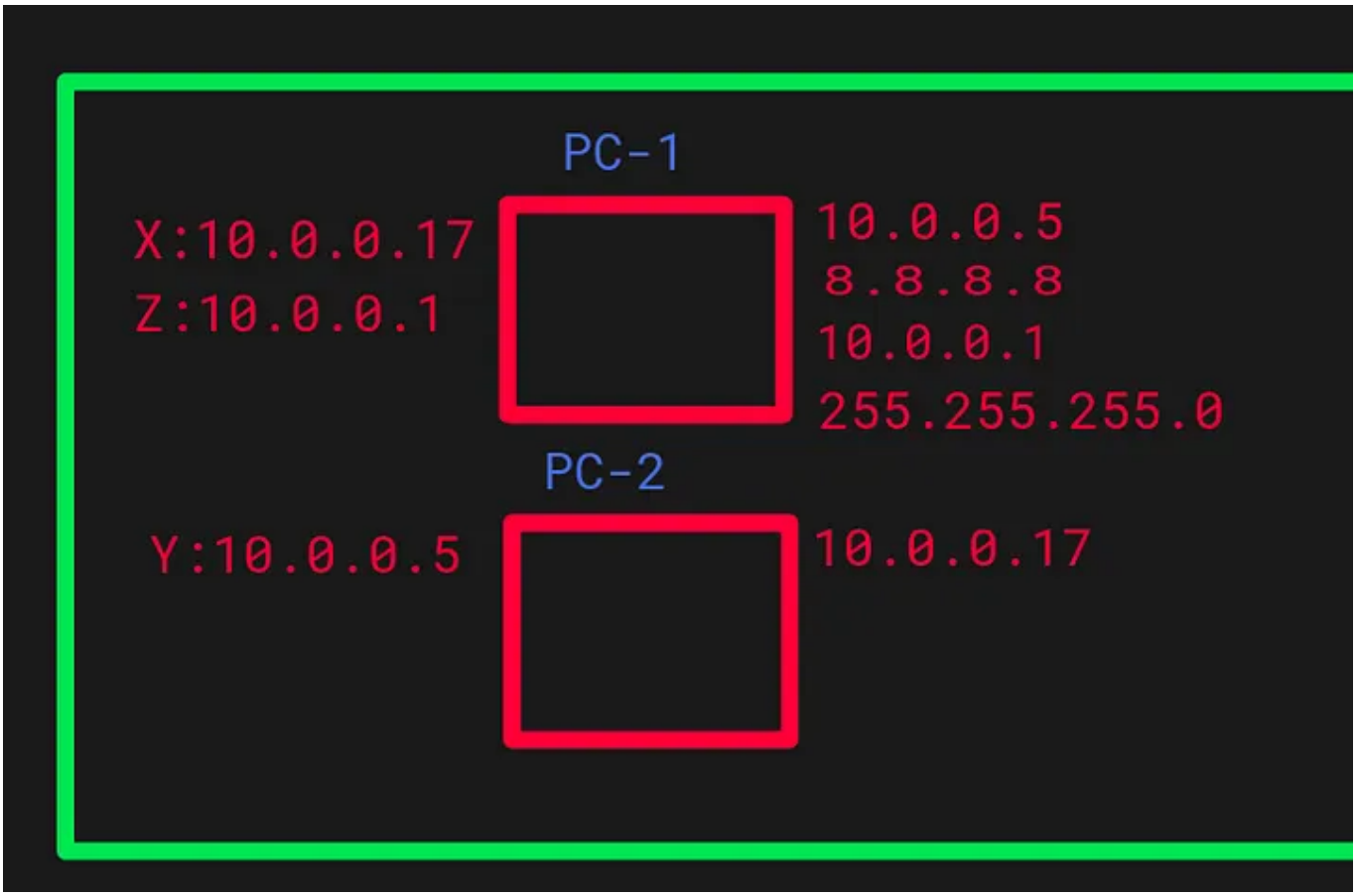
Buraya kadar bahsedilen kısım yerel ağımızda gerçekleşen kısımdı. Buna karşılık bir de internete bağlanmamız ve farklı dünyalara açılmamız da söz konusudur.



Subnet

Bir diğer konu da bulunduğumuz ortamda aradığımız kişinin bulunması ile ilgilidir. Örneğin sınıfta Ahmet isimli bir kişiyi aradığımızda eğer o sınıfta öyle biri yoksa kapının dışına çıkmamız gerekmektedir. Dolayısıyla

aradığımız kişinin aynı ortamda olup olmadığını bilmemiz gerekmektedir. Burada da devreye subnet meselesi girmektedir.



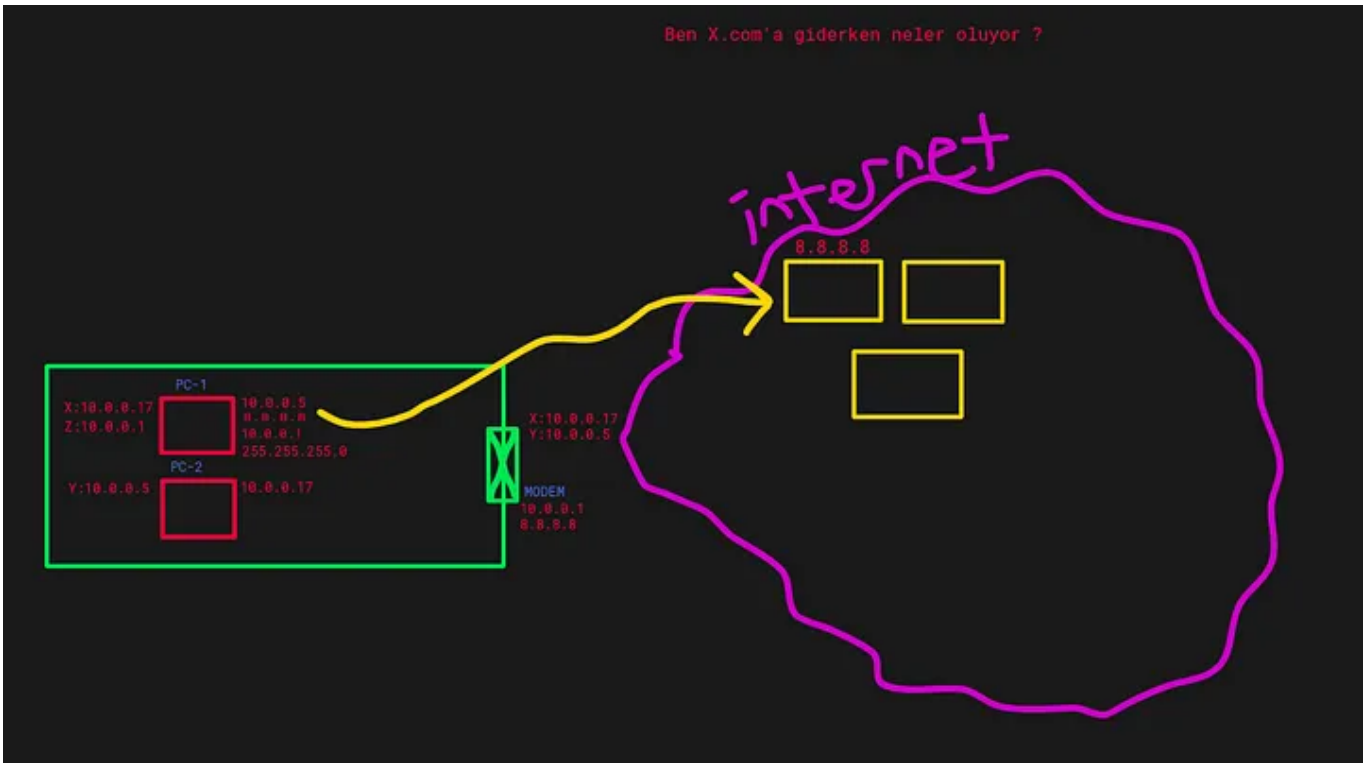
Tüm bunlar henüz bilgisayarı açtığımızda yaşanmaktadır. Google'a gitmedik henüz.

Peki Bundan Sonra Ne Olacak ?

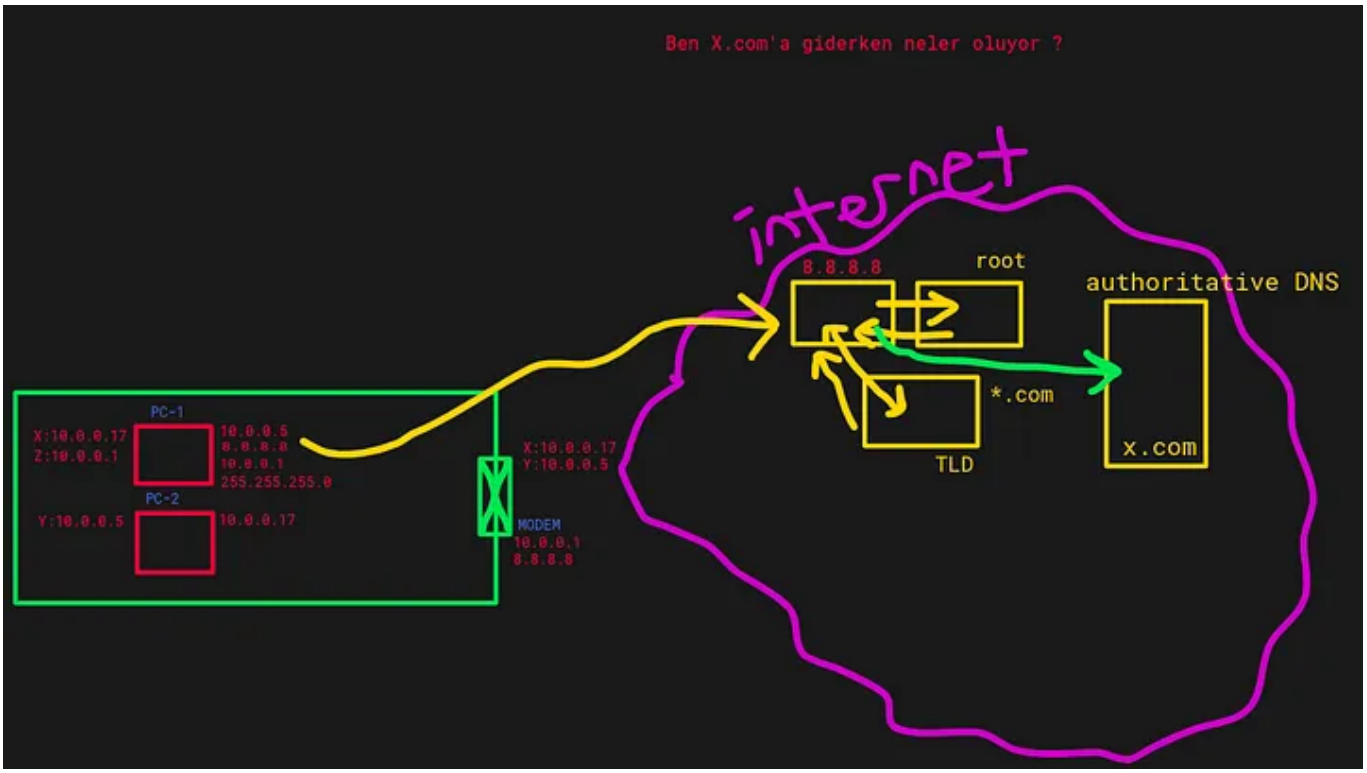
Bilgisayarı açtık, bu kısımdan sonra X.com'a gittiğimizde neler olacağını göreceğiz. Peki ilk önce ne olacak? Browser'ı açıp x.com yazdığımızda ne olmaya başlıyor ?

DNS ile Konuşmak

Bilgisayar öncelikle kendi host dosyasına bakmalıdır. Linux'ta bu /etc/hosts yolundadır. Hosts dosyasında bir domain'in isim çözümlendirmesinin ne olacağını belirleriz. Eğer hosts dosyasında x.com'a yönelik bir kayıt yoksa bu durumda DNS devreye girer. DNS'e sorar. DNS ise 8.8.8.8 olarak belirlenmişti burada. Öncelikle DNS ile konuşacağız ancak DNS ile konuşmak için yola çıkmalıyız.



Burada DNS protokolü bulunmaktadır ve DNS, UDP ve TCP kullanmaktadır. Burada X.com'a bağlanmak isteyen bilgisayar 8.8.8.8 DNS'ine giderek x.com'un ip adresini sorgular. Burada Resolver DNS x.com'un ip adresini bilmiyor ise başka bir Route DNS'e giderek x.com'un ip adresini öğrenmek istediğini belirtir. Route DNS de x.com'un ip adresini bilmediğini belirtir. Ancak Route DNS x.com'un ip adresini kimin bileceğinisöyler. 8.8.8.8 Resolver DNS ise bu kişiye gider. Bu da TLD olarak kabul edilebilir. Top Level Domain DNS'i gibi düşünebiliriz. Bu kişinin işi *.com'ları bilmektir. Burada TLD de x.com'u bilmiyorum der ancak x.com ile ilgili kayıtların tutulduğu yeri bildiğini söyler. Burası da Authoritative DNS'dir. Tüm bu yapıyı şu şekilde düşünebiliriz:

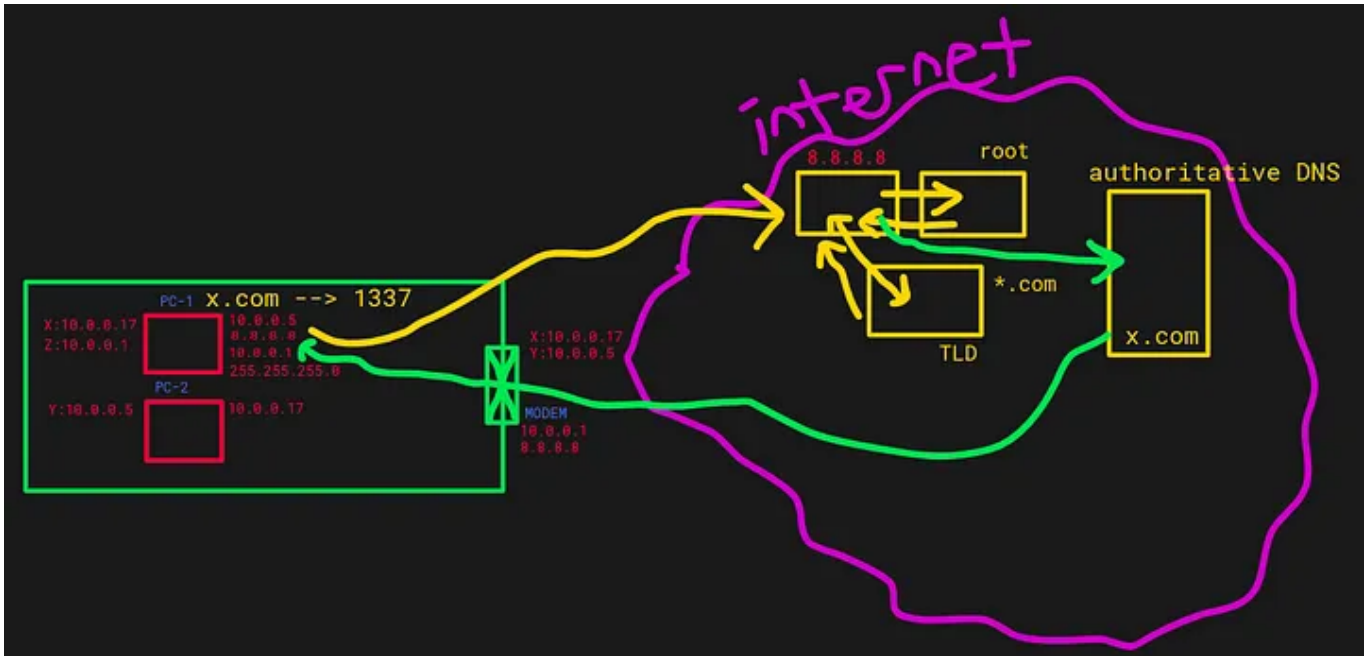


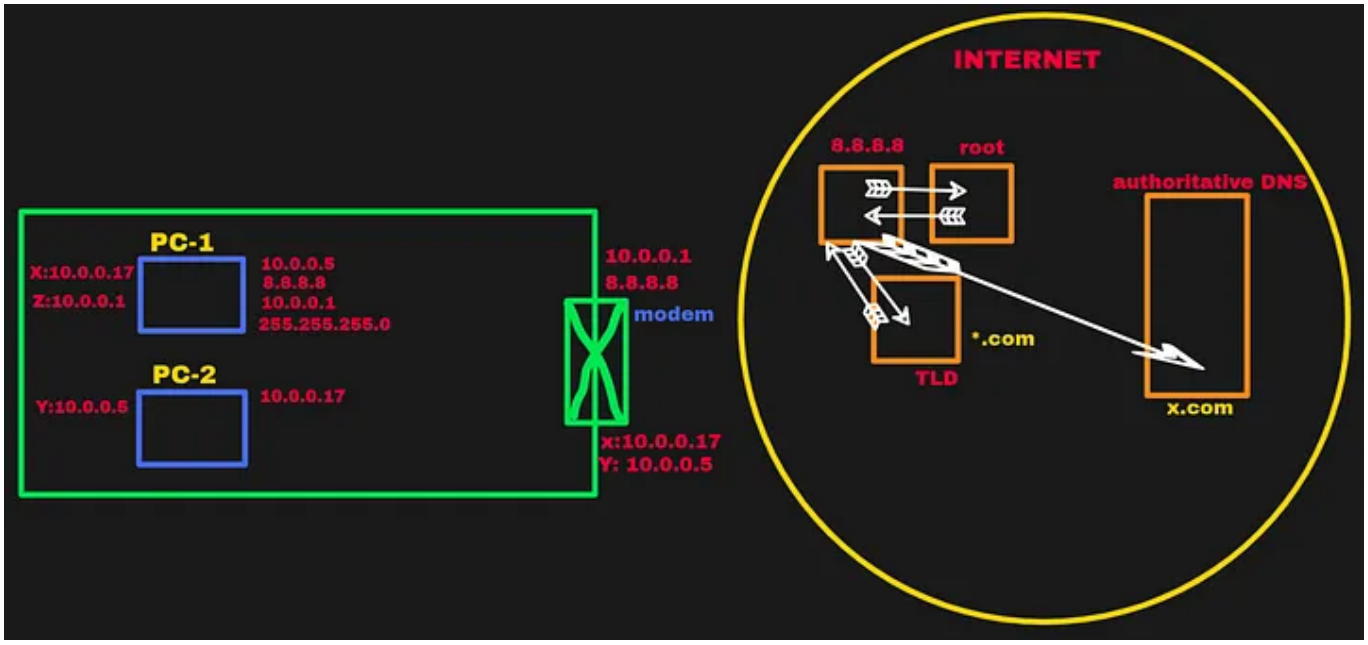
Bu da DNS bilgileri ile ilgili örnek bir komut ve çıktısı:

```
[ilker@cyberworld ~]$ dig NS hurriyet.com.tr

; <<>> DiG 9.18.19 <<>> NS hurriyet.com.tr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44715
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;hurriyet.com.tr.               IN      NS

;; ANSWER SECTION:
hurriyet.com.tr.        37127   IN      NS      ns-601.awsdns-11.net.
hurriyet.com.tr.        37127   IN      NS      ns-63.awsdns-07.com.
hurriyet.com.tr.        37127   IN      NS      ns-1789.awsdns-31.co.uk.
hurriyet.com.tr.        37127   IN      NS      ns-1172.awsdns-18.org.

;; Query time: 20 msec
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Mon Nov 13 22:14:43 +03 2023
;; MSG SIZE  rcvd: 183
```

Authoritative DNS'ten bilgisayarımıza artık gerekli cevap iletilmiş oldu. Bu aşamadan sonra da artık X.com'un ip adresini kolay ilerleyebilmek için 1337 olarak kabul edelim.

Genel yapımızı şimdi de bu şekilde düşünebiliriz:





Riskleri Konuşalım

Burada artık başka bir bilgisayar x.com'un ip adresini sorarsa döngü tekrar yaşanmaz ve direkt olarak cevap verilir. DNS sunucusunun TTL'lere göre cache'de tutma mekanizması devreye girer. O yüzden burada cache poisoning yapabilirseniz ve yanıltırsanız, x.com'a gidecek olan tüm http taleplerini kendi istediğiniz ip adresine çekebilir hale gelirsiniz.

Veya siz buradaki authoritative DNS sunucusunu ele geçirirseniz o kurum için oldukça sıkıntılı bir durum oluşabilir. MX kayıtlarını değiştirebilirsiniz, CNAME Records'larını değiştirerek tüm kayıtları üzerinize alabilirsiniz. TXT kayıtlarına istediğinizi girerek sertifika otoritelerinden sertifika issue ettirebilirsiniz. DNS internetin en zayıf halkasıdır düşüncesinin de temelde oluşmasının sebebi bu kısımlardan kaynaklanmaktadır.

Örnek bir MX Sonucu:

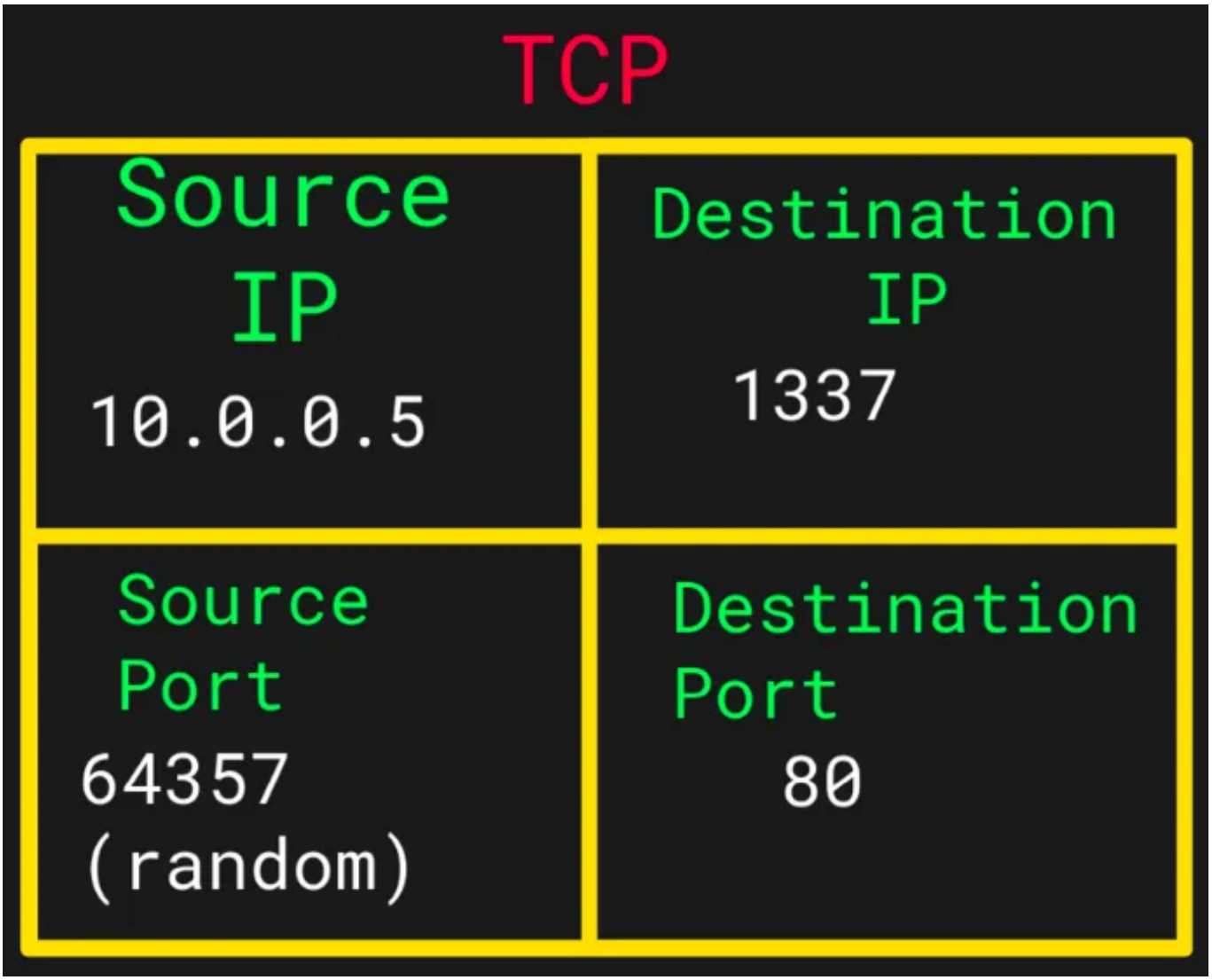
```
[ilker@cyberworld ~]$ dig MX hurriyet.com.tr

; <<>> DiG 9.18.19 <<>> MX hurriyet.com.tr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15519
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;hurriyet.com.tr.               IN      MX

;; ANSWER SECTION:
hurriyet.com.tr.        3600    IN      MX      0 hurriyet-com-tr.mail.protection.outlook.com.

;; Query time: 60 msec
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Thu Nov 16 06:59:35 +03 2023
;; MSG SIZE  rcvd: 103
```

Bu Kısımdan Sonra Mesele Nerelere Gelmeye Başlayacak Bakalım

Bir adet sunucumuz var. Bunun da ip adresi 1337 olarak kabul edilsin. gateway'den çıkıp internetten geçerek sunucuya erişecektir. Burada da TCP girmektedir.

TCP Paketi

Burada bahsettiğimiz bağlantı 1337'nin 80 portuna gitmeye çalışmaktadır. Bu yaşanırken de TCP devreye girmektedir. Bir TCP header'ını en basit şekilde düşünecek olursak bu şekilde bir yapı ortaya koyabiliriz;



NAT

Burada bahsettiğimiz şey bi TCP paketidir. Bu TCP paketi gateway'e geldikten sonra source port'u 10.0.0.5 olarak internette gezemez. Burada da hayatımıza NAT (Network Address Translation) girmektedir. Buradaki router giden paketin source ip değerini 10.0.0.5 yerine başka bir ip adresiyle değiştirmektedir. Bu da ortamdaki iki bilgisayarın da internete çıkarken aynı ip adresini kullandıkları anlamına gelir. Bunun sebebi de internete bağlı cihaz sayısı ile ipv4 cihaz sayısı arasındaki uçurumdan dolayıdır.

Peki buraya gelen TCP Paketinin Tipi Nedir? Biraz da bu sorunun cevabını düşünelim.

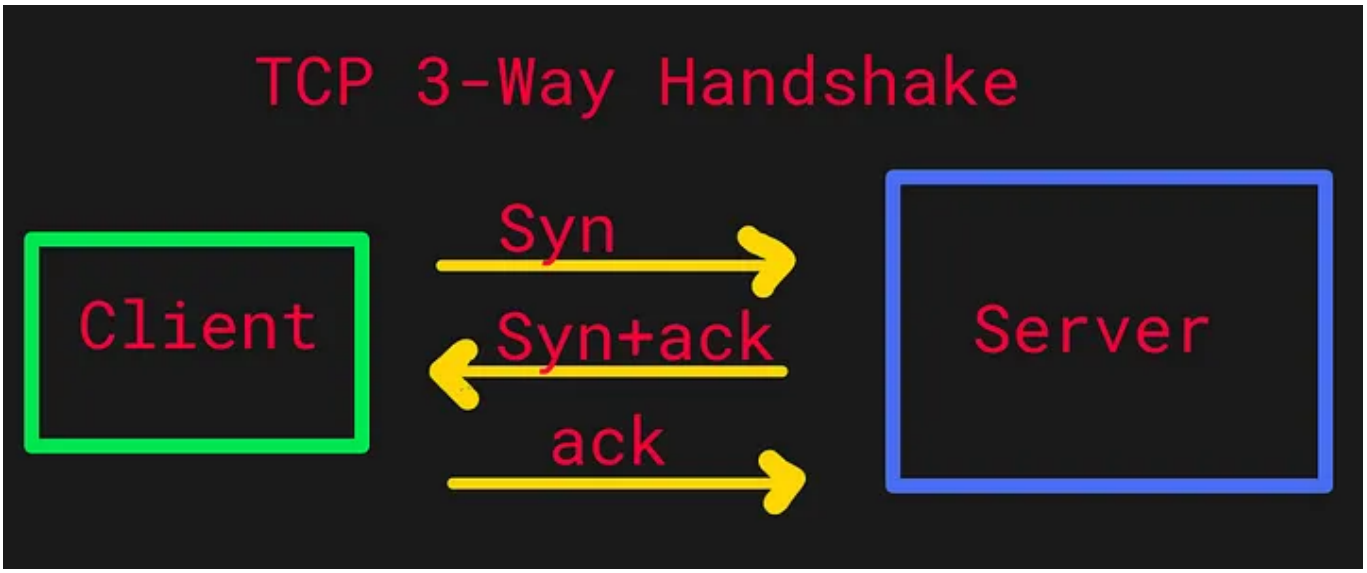
TCP 3-Way Handshake

Bu konuyu önceki yazımızda da ele almıştık. Bu yapıda bir canlandırma yapacak olursak şu şekilde bir konuşma geçmektedir;
Client: "Merhaba, seninle konuşmak istiyorum." **(Syn)**

Server: "Merhaba benimle konuşmak istediğini duydum ve seninle konuşmaya müsaitim" **(Syn+Ack)**

Client: "Merhaba, ben seninle konuşmak istediğimi söylemiştim sen de bunu duymuşsun ve benimle konuşmak için müsait olduğunu duydum. Hadi konuşalım." **(Ack)**

Burada kısaltılmış olarak kullanılan Syn ve Ack ifadeleri aslında ***synchronize*** ve ***acknowledgment*** ifadelerinin kısaltılmış halidir.



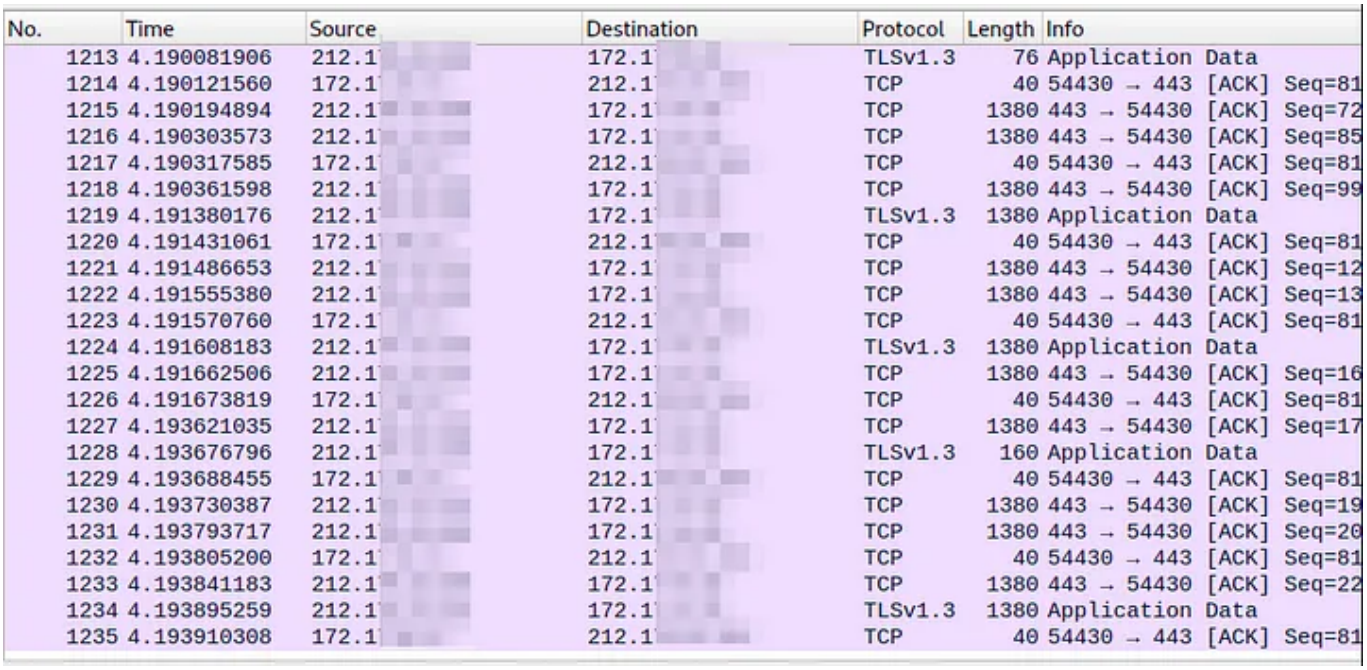
TCP 3-Way Handshake

Tüm bu anlatılanlar yaşanmadan HTTP'ye gelemeyiz. Bu mümkün değildir.

HTTP'ye Çıkma Vakti

Artık hedefimizin 80 portu ile üçlü el sıkışmayı (3-Way Handshake) tamamladık ve HTTP'ye çıkabiliriz. HTTP ifadesi de Hyper Text Transfer Protocol şeklinde açılabilir.

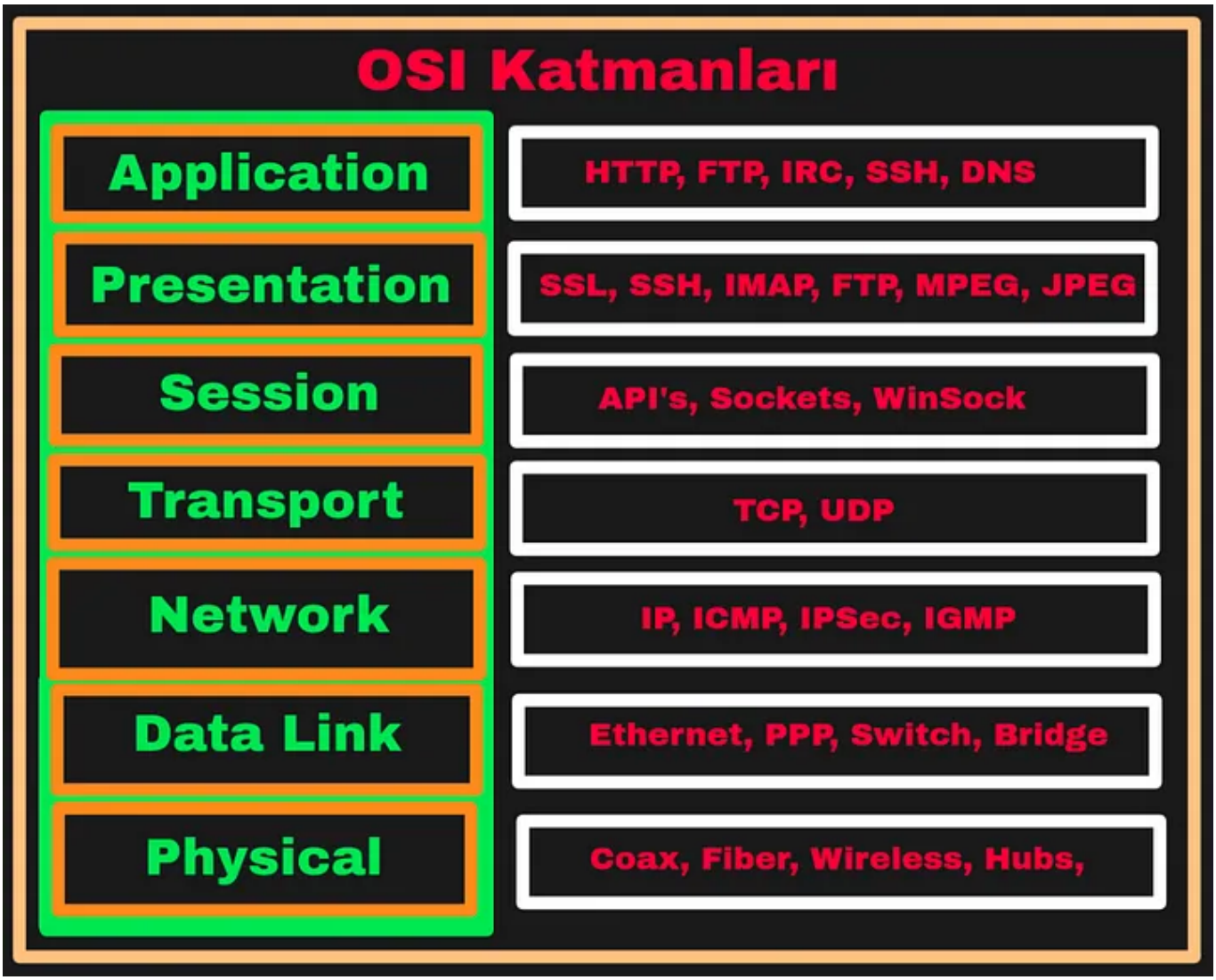
Aklımızda şöyle canlandıralım, bir adet HTTP request'i göndermekteyiz ve buna karşılık bir response almaktayız. OSI Modelinde 7. katmanda iken bu şekilde sadece 2 tane paket varken, alt katmanlarda yüzlerce TCP paketi gidip gelmektedir. Bunu da temede anlamanın en kolay yolu wireshark'ı açarak trafiği izlemektir. Burada görebileceğiniz üzere TCP paketlerinin sayısı oldukça fazladır.



No.	Time	Source	Destination	Protocol	Length	Info
1213	4.190081906	212.1	172.1	TLSv1.3	76	Application Data
1214	4.190121560	172.1	212.1	TCP	40	54430 → 443 [ACK] Seq=81
1215	4.190194894	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=72
1216	4.190303573	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=85
1217	4.190317585	172.1	212.1	TCP	40	54430 → 443 [ACK] Seq=81
1218	4.190361598	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=99
1219	4.191380176	212.1	172.1	TLSv1.3	1380	Application Data
1220	4.191431061	172.1	212.1	TCP	40	54430 → 443 [ACK] Seq=81
1221	4.191486653	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=12
1222	4.191555380	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=13
1223	4.191570760	172.1	212.1	TCP	40	54430 → 443 [ACK] Seq=81
1224	4.191608183	212.1	172.1	TLSv1.3	1380	Application Data
1225	4.191662506	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=16
1226	4.191673819	172.1	212.1	TCP	40	54430 → 443 [ACK] Seq=81
1227	4.193621035	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=17
1228	4.193676796	212.1	172.1	TLSv1.3	160	Application Data
1229	4.193688455	172.1	212.1	TCP	40	54430 → 443 [ACK] Seq=81
1230	4.193730387	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=19
1231	4.193793717	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=20
1232	4.193805200	172.1	212.1	TCP	40	54430 → 443 [ACK] Seq=81
1233	4.193841183	212.1	172.1	TCP	1380	443 → 54430 [ACK] Seq=22
1234	4.193895259	212.1	172.1	TLSv1.3	1380	Application Data
1235	4.193910308	172.1	212.1	TCP	40	54430 → 443 [ACK] Seq=81

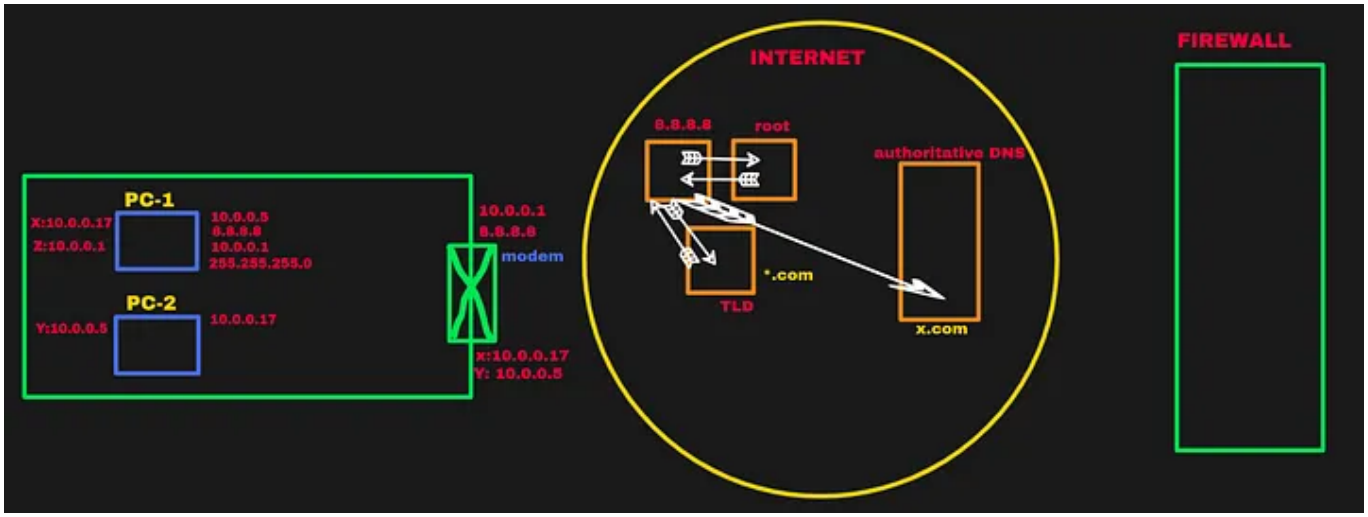
wireshark ile trafiği izleme

OSI Modelini de genel olarak şu şekilde gösterebiliriz:



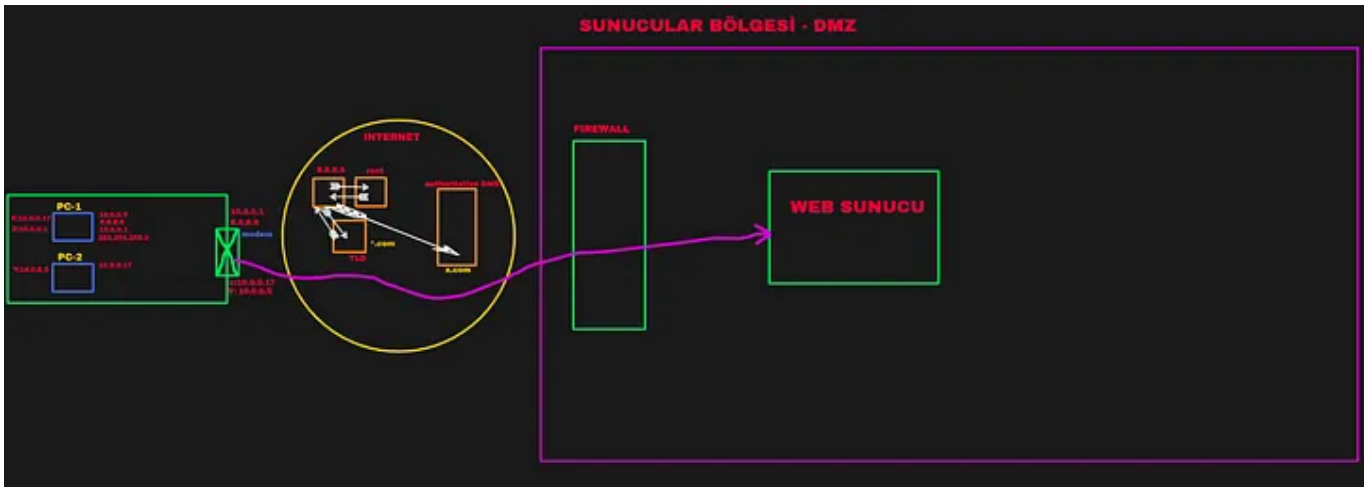
Günümüz Sistemlerine Bakış ve Firewall Hakkında

Günümüzde 80 portuna giderken aslında üzerinden geçtiğimiz bir firewall bulunmaktadır. Firewall genel olarak network yönetim aracı gibi düşünülebilir.



Günümüzdeki Sistemerin Genel Yapısı

Burada ancak firewall'den geçtikten sorna web sunucusuna gelinebilmektedir. Sağ tarafta görmüş olduğunuz kısım da kurumun sunucular bölgesidir. DMZ olarak da düşünebilirsiniz.



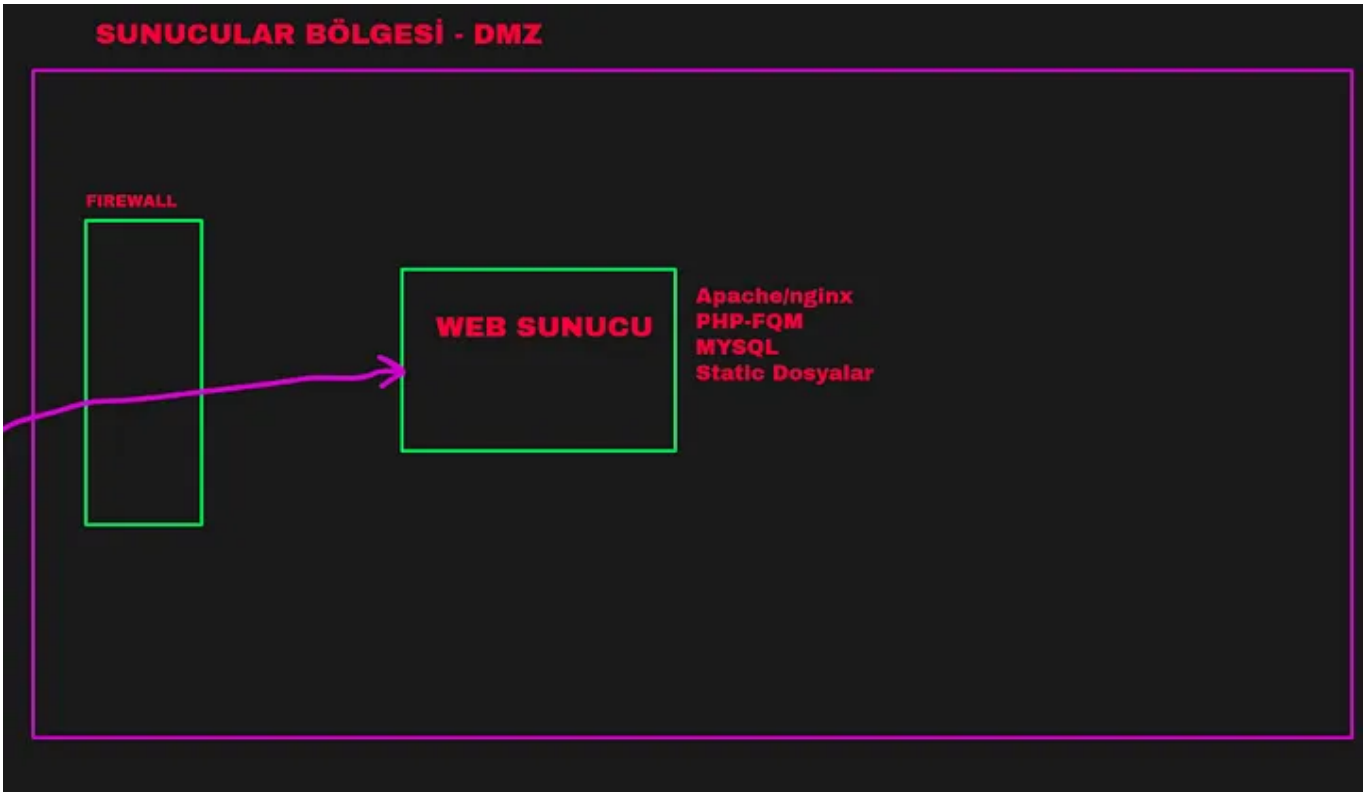
SYN Flood Attack

Hepiniz TCP Syn Attack konusunu muhtemelen görmüşsünüzdür. TCP Syn request'i gönderirsiniz, ip adresine de random bir ip adresi yazarsınız, paketi de TCP Syn paketi olarak gönderirsiniz, bu da web sunucusuna kadar gelir ve

web sunucusu başka bir cihaz ile konuşmaya başlar. Çünkü source kısmına random bir değer yazdınız. Yani TCP Syn Flood. Bu durumda da bu sunucunun kaynağı tükenir. Bu şekilde web sunucularının kaynağı tükendiği için ön taraftaki firewall'lar genellikle daha fazla güce sahip olurlar. Dolayısıyla web sunucusu olası bir DDOS Saldırısına karşı hazır olmak için daha fazla RAM'e ihtiyaç duymaz, bu görevi firewall halletmektedir.

Web Sunucusuna Geldik

Artık web sunucusuna gelmiş olduk. Bu web sunucusunun üzerinde de artık farklı hikayeler ve durumlar mevcuttur. Günümüzde tüm işlemlerin ve teknolojilerin bu şekilde tek bir web sunucusunda bulunması pek mümkün değildir.



Virtual Hosting

Dünya genelinde kaç tane ip adresi ve kaç tane domain olduğunu düşünecek olursak aradaki uçurumu görebiliriz. Bu yüzden tıpkı yerel ağda yaşanan NAT işlemi gibi web sunucuları da buna benzer bir olay yaşar. Bu Virtual Hosting meselesidir. VHOST da diyebiliriz.

Ayrıca HTTP GET Request'ine yazdığımız Host bilgisi buradaki web sunucusuna geldiğinde ilgili konfigürasyonda şu şekilde bir tanım vardır; Host alanında verilen host adı yazıyorsa cihazın çağırdığı dosyayı host içerisinden vermektedir. Örneğin index.html'i istiyorsa bu host alanındaki index.html verilecektir.

Bu sayede dünyada onlarca sayıda domain'in ip adresi aynı olurken bir sunucu üzerinde bu şekilde farklı uygulamalar halinde yer alabilmektedir. Buna da Virtual Hosting demekteyiz.

```
> GET / HTTP/2
> Host: hackerconf.stream
> User-Agent: curl/8.4.0
> Accept: */*
```

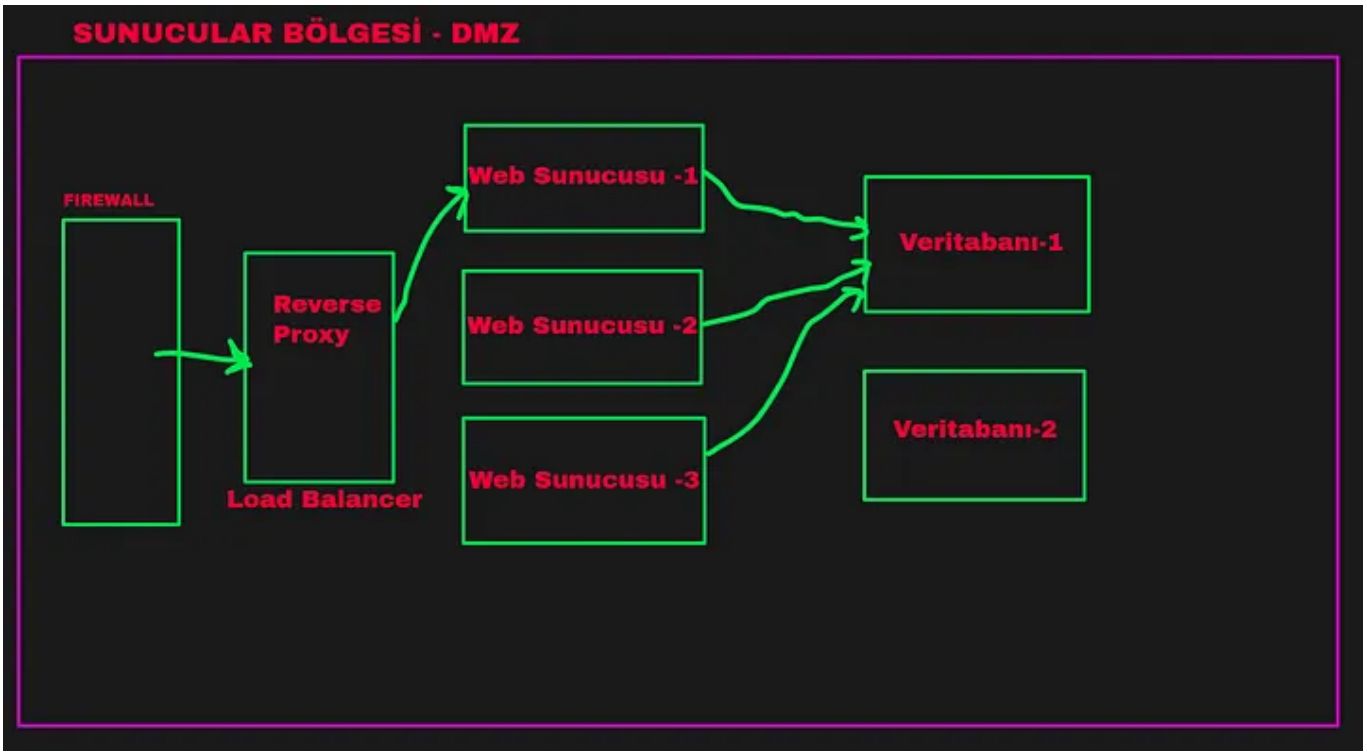
Sunucu Yetmediği Zaman Ne Yapılır? — Reverse Proxy

Buradaki yapıyı bir e-ticaret sitesi olarak düşünelim. Burada bazı problemler meydana gelebilmektedir. Örneğin black friday günlerinde çok fazla ziyaret ve satış yapıldığı için mevcut sunucu kendisi için yeterli olmamaktadır.

Günümüzde bu tarz problemleri çözebilmek için bu sunucudan birkaç tane daha kurabiliriz. 3 tane web sunucusu kurduğumuzu ve kaynak kodlarımızı da tüm sunuculara yüklediğimizi düşünelim. 2 tane de veritabanımız (database) olmuş olsun. Veritabanını da web sunucusunun dışında tutalım. Aynı zamanda firewall'den gelen request'leri bu yapılara göndermemiz gerektiği için bize Reverse Proxy ya da Load Balancer lazım olacaktır. Yani sizin gönderdiğiniz HTTP Request'ini alan Reverse Proxy bu request'i içeride

müsait olan sunucuya aktarmaktadır. Peki neye göre aktarmaktadır? Bu kısımda bazı sorunlar da yaşanabilmektedir.

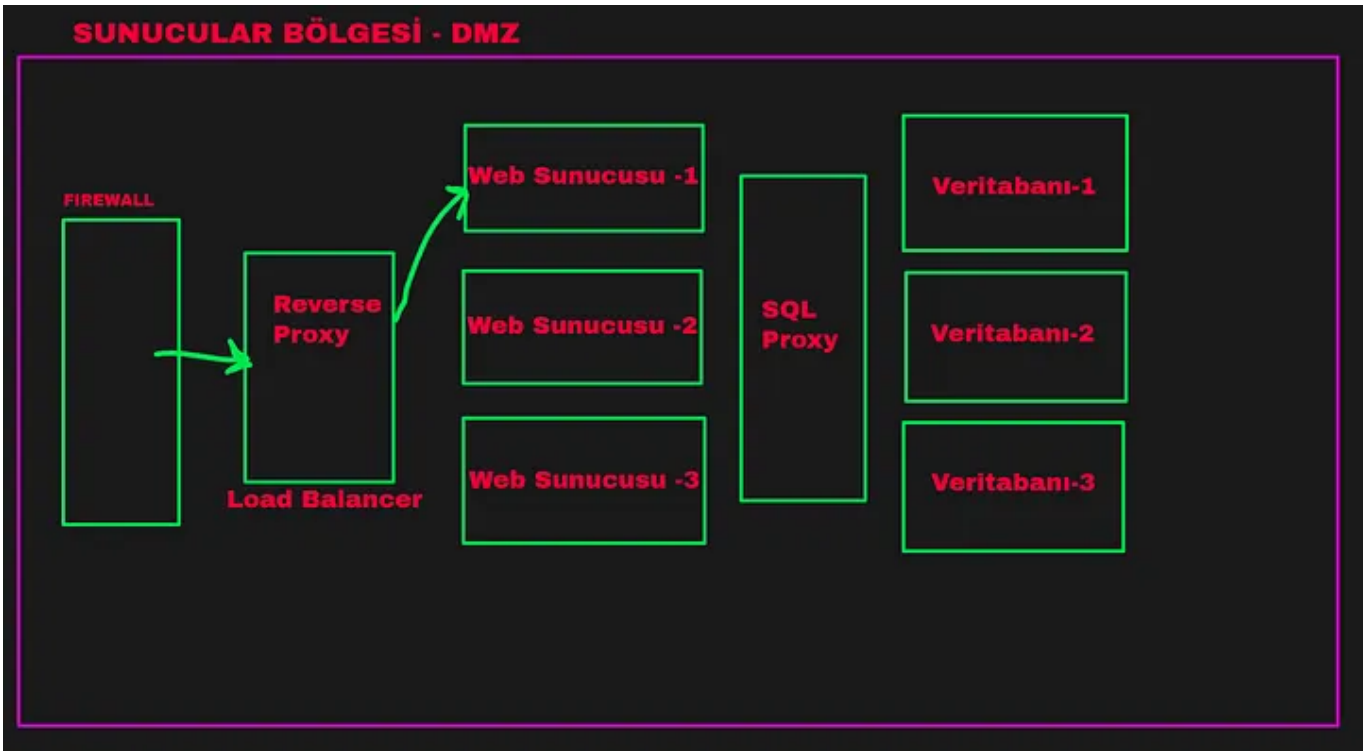
Reverse Proxy'nin web sunucusuna ilettiği request bu kısma geldikten sonra bu uygulamanın çalışırken oluşturduğu session eğer diskte tutuluyorsa artık bu request'lerin her seferinde aynı sunucuya gelmesi gerekmektedir. Çünkü oluşan session sadece tek bir sunucunun diskinde olmuş olur. Diğer sunucularda session bilgisi bulunmamış olur.



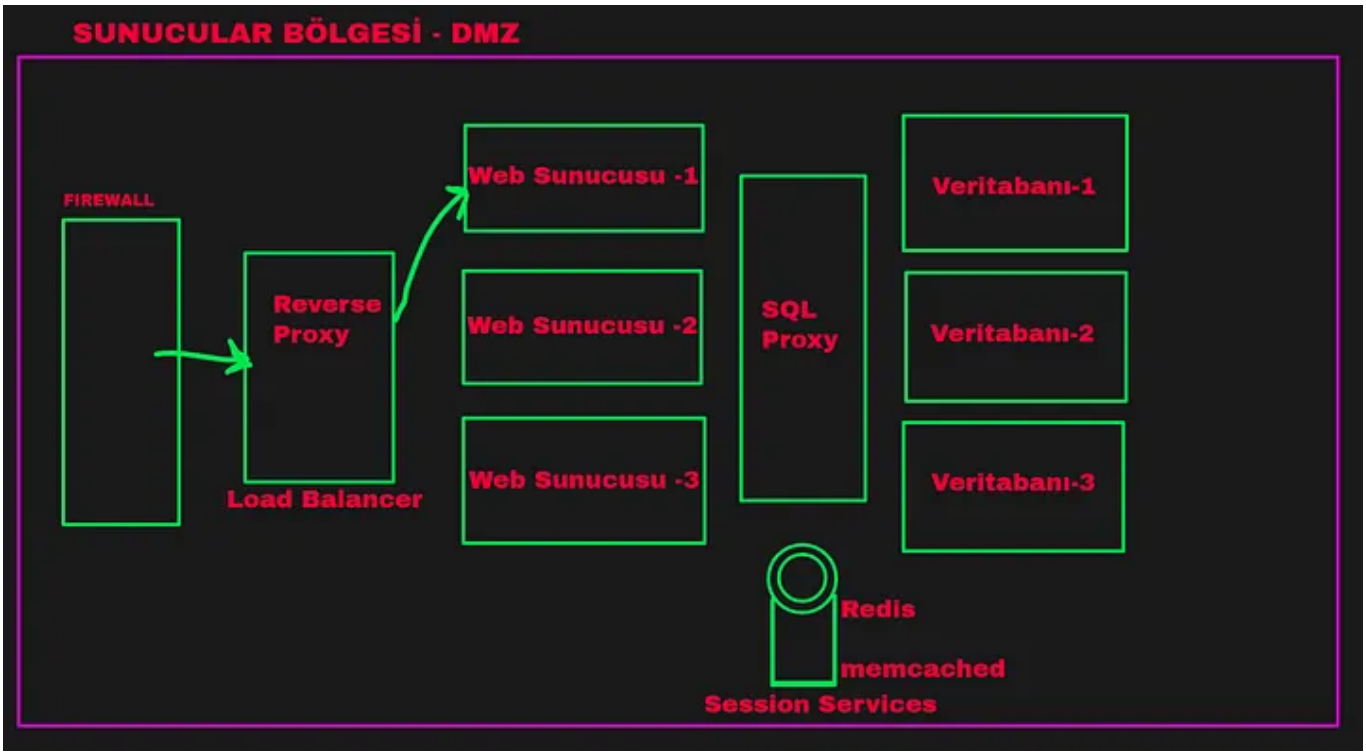
Diğer Sorunlar — Microservices Yapılarının Ortaya Çıkış Noktası

Bazı durumlarda tek bir veritabanı ya da 2 veritabanı yeterli olmayabilir. 3 tane veritabanı da kullanılabilir. Bu durumda ek olarak bir SQL Proxy yapısına da ihtiyacımız olacaktır. Bu SQL Proxy sayesinde hangi veritabanı

uygunsa o kullanılarak sorgulamalar yapılır. Günümüzde mikroservis yapılarının da ortaya çıkmaya başladığı nokta burasıdır.



Peki session meselesi ne olacak? Bu konuda daha önce şunları dile getirmiştik, eğer session bilgisi web sunucusundaki diskte tutulursa diğer web sunucuları bu bilgiye erişemeyecektir. Session bilgisi veritabanında tutulduğunda ise veritabanındaki query'ler yani sorgular artmaya başlayacaktır. Bu da istenmeyen bir durumdur. Session da kalıcı olmayan yani gelip geçici bir bilgi olduğu için bu bilgiyi başka ortamlarda tutalım diye bahsetmiştik ve Redis gibi session servisleri de bu şekilde ortaya çıkmaya başladı.



Backup

Tüm bunları konuştuktan sonra bir de bu yapıların backup'ının olması gerektiğini söylemeliyiz. Örneğin redis'te tutulan session bilgileri ayrıca bir yerde backup'ta tutulmalı. Çünkü redis'e erişilemezse kimse login olamayacaktır.

Statik Dosyalar

Ayrıca uygulamada bulunan static dosyaların tek bir web sunucusunda olmaması gerekmektedir. Çünkü olası bir değişiklik durumunda başka bir web sunucusuna geçildiğinde o static dosyalara artık erişilemeyecektir. Dolayısıyla bu sorunun çözümü için CDN (Content Delivery Network) mekanizmalarının olması gerekmektedir. CDN Sayesinde Cloud Flare ile de birlikte sizin uygulamanızdaki dosyalar vs. dünya üzerindeki çeşitli noktalara cache ile iletilebilmektedir. Tüm noktalardan sizin uygulama sunucunuza ya da CDN mekanizmanıza erişmelerine gerek kalmamaktadır.

Tüm bunların yanında sunucular bölgesinin aynısından bir tane daha bulunmaktadır. Eğer kullanılan hizmetlerde bir kesinti yaşanırsa hiçbir sorun yaşanmadan aynı şekilde devam edilebilmesi için bu şekidle bir kurtarıcı yapı da bulunmaktadır.

Tüm bu anlatılanları düşününce günümüz web dünyasının ne kadar kapsamlı bir yapıda olduğunu görebiliriz. Dolayısıyla bu sistemler üzerinde de envai çeşit saldırı yöntemleri, atak vektörleri düşünülebilir.

Buraya kadar okuduğunuz için teşekkürler, konu ile ilgili daha ayrıntılı bilgiye ulaşmak için kaynaktaki dersleri takip edebilirsiniz.

Sağlıcakla Kalın…

Kaynaklar:

1. (Bir Hacker'ın Gözünden Modern Web Nasıl Çalışır ? — YouTube)