PAKETLER ve ÇERÇEVELER
Kaynak

İÇERİK

Packets and Frames (Paketler ve Çerçeveler)

Ağ dünyasında verinin nasıl taşındığını anlamak, siber güvenliğin temel taşıdır. Bir dosyayı veya mesajı olduğu gibi kablodan gönderemeyiz; bu verinin yönetilebilir parçalara bölünmesi gerekir. İşte burada **Packet** (Paket) ve **Frame** (Çerçeve) kavramları devreye giriyor. Bir üniversite öğrencisi olarak bu konuyu sadece "veri parçası" diyerek geçemeyiz; hangi katmanda ne isim aldığını ve içine ne eklendiğini bilmemiz şart.

Paket ve Çerçeve Ayrımı: OSI Katmanlarındaki Yerleri

Her ne kadar ikisi de "küçük veri parçası" olsa da, OSI modelindeki yerleri ve içerdikleri bilgiler bakımından birbirlerinden tamamen farklıdırlar. Aradaki farkı anlamak için **Encapsulation** (Kapsülleme) sürecini hatırlamamız gerekiyor.

1. Packet (Paket) - OSI Katman 3 (Network Layer)

Bir **Packet**, OSI modelinin 3. katmanı olan **Network Layer** (Ağ Katmanı) birimidir. Bu aşamada veri, yönlendirme (routing) yapılabilmesi için gerekli olan mantıksal adresleme bilgilerini içerir.

- **İçerik:** Temel olarak bir **IP Header** (IP Başlığı) ve asıl taşınan veri olan **Payload** kısmından oluşur.
- **Kritik Nokta:** Eğer bir yerde **IP adresi** mevzubahis ise, orada konuştuğumuz birim kesinlikle bir **pakettir**.

2. Frame (Çerçeve) - OSI Katman 2 (Data Link Layer)

Veri, 3. katmandan 2. katman olan **Data Link** (Veri Bağlantısı) katmanına indiğinde, paketimiz bir **Frame** (Çerçeve) içine hapsedilir.

- **İçerik:** Çerçeve, paketi sarmalar ve üzerine fiziksel adresleme olan **MAC adreslerini** ekler.
- **İşlem:** Paket bir zarfın içindeki mektupsa, çerçeve o mektubun konulduğu ve üzerine fiziksel adreslerin yazıldığı zarfın kendisidir. Alıcı cihaz zarfı (çerçeveyi) açtığında,

içindeki mektubun (paketin) nereye iletileceğini anlar.

Neden Parçalara Bölüyoruz? (Verimlilik ve Darboğaz)

Büyük bir veriyi (örneğin 10 GB'lık bir oyun dosyasını veya yüksek çözünürlüklü bir kedi fotoğrafını) tek bir blok halinde göndermek ağ için felakettir.

- **Bottleneck (Darboğaz) Önleme:** Veriler küçük parçalara bölünerek gönderildiğinde, ağ hatlarını tek bir büyük dosya ile tamamen işgal etmemiş oluruz. Bu sayede aynı anda birçok kullanıcı ağ kaynaklarını paylaşabilir.
- **Hata Yönetimi:** Eğer 3 parçaya bölünmüş bir fotoğrafın (örneğin bir kedi fotoğrafı örneği) 2. paketi yolda kaybolursa, sadece o küçük paket tekrar istenir. Eğer veri tek parça olsaydı, en ufak bir hatada tüm veriyi baştan göndermek zorunda kalırdık.
- **Yeniden Birleştirme:** Veriler alıcı bilgisayara ulaştığında, bu küçük parçalar protokoller aracılığıyla sıraya dizilir ve orijinal bütün (resim, dosya vb.) tekrar oluşturulur.

İnternet Protokolü (IP) Paket Yapısı ve Header Bilgileri

İnternet üzerindeki milyarlarca cihazın birbiriyle sorunsuz konuşabilmesi için paketlerin yapısı **Standardizasyon** kurallarına bağlıdır. Bir IP paketi gönderilirken, verinin önüne eklenen **Headers** (Başlıklar), o paketin ağdaki pasaportu gibidir.

Aşağıdaki tablo, bir paketin ağda güvenle ve doğru yere ulaşmasını sağlayan en kritik başlık alanlarını açıklamaktadır:

Başlık (Header)	Açıklama ve Teknik Fonksiyonu
Time to Live (TTL)	Paketin ağ üzerinde ne kadar süre "hayatta kalacağını" belirleyen bir sayaçtır. Eğer paket hedefine ulaşamazsa ve sürekli yönlendiriciler arasında dönerse, TTL değeri her sıçramada bir azalır ve sıfıra ulaştığında paket imha edilir. Bu, ağın sonsuz döngülerle kilitlenmesini (clog up) engeller.
Checksum	**Hata Kontrolü** sağlar. TCP/IP gibi protokollerde verinin bütünlüğünü (integrity) denetler. Gönderici bir matematiksel değer hesaplar; eğer veri yolda değişirse (bozulursa), alıcının hesapladığı değer tutmaz ve paket "corrupt" (bozuk) kabul edilerek atılır.
Source Address	**Kaynak Adresi**. Paketi gönderen cihazın IP adresidir. Alıcı cihazın, cevabı nereye döneceğini bilmesi için bu bilgi şarttır.
Destination Address	**Hedef Adresi**. Paketin gitmesi gereken nihai IP adresidir. Ağdaki yönlendiriciler bu adrese bakarak paketi hangi yola sokacaklarına karar verirler.

> **Özet Not:** Bir analiz aracında (Wireshark vb.) veri incelerken, MAC adreslerini görüyorsak **Frame** seviyesindeyizdir; eğer sadece IP adresleri ve verinin mantıksal

> akışını görüyorsak **Packet** seviyesindeyizdir. Unutma, her Frame bir Packet içerir ancak her Packet henüz bir Frame değildir (ta ki fiziksel katmana inene kadar).

TCP/IP ve Üçlü El Sıkışma (The Three-Way Handshake)

Siber güvenliğin bel kemiği olan **TCP (Transmission Control Protocol)**, verinin güvenilir, hatasız ve sıralı bir şekilde karşı tarafa ulaştırılmasını sağlayan protokoldür. Bir önceki konudaki OSI modelinin daha pratik ve güncellenmiş bir versiyonu olan **TCP/IP** modeli üzerine kuruludur.

TCP/IP Modeli Katmanları

OSI modelindeki 7 katman, TCP/IP modelinde daha sade bir yapıya (4 katman) indirgenmiştir. Veri yukarıdan aşağı inerken her katmanda üzerine yeni bilgiler eklenir (**Encapsulation**), alıcıda ise bu bilgiler tek tek soyulur (**Decapsulation**).

1. **Application (Uygulama):** Kullanıcının etkileşime girdiği katman (HTTP, FTP, SSH).
2. **Transport (Taşıma):** Verinin nasıl taşınacağının belirlendiği katman (TCP, UDP).
3. **Internet (İnternet):** Paketlerin yönlendirildiği katman (IP).
4. **Network Interface (Ağ Arayüzü):** Verinin fiziksel olarak kabloya/havaya çıktığı katman (Ethernet, Wi-Fi).

TCP'nin Karakteristiği: Bağlantı Tabanlı (Connection-Based) İletişim

TCP'yi UDP'den ayıran en büyük özellik **bağlantı tabanlı** olmasıdır. Yani, gerçek veri transferi başlamadan önce istemci (client) ve sunucu (server) arasında el sıkışılarak bir "anlaşma" sağlanır.

Avantaj ve Dezavantaj Tablosu

Avantajlar	Dezavantajlar
Veri Bütünlüğü (Integrity): Verinin tam ve hatasız ulaştığını garanti eder.	**Güvenilir Bağlantı Şartı:** Eğer küçük bir veri parçası bile kaybolursa, tüm veri bloku kullanılamaz ve tekrar gönderilmesi gerekir.
Senkronizasyon: Cihazların birbirini veri yağmuruna tutmasını (flooding) önler, verileri sıraya koyar.	**Yavaşlık:** UDP'ye göre çok daha yavaştır; çünkü onay mekanizmaları ek işlem gücü (overhead) gerektirir.
Sıralı İletim: Paketler farklı yollardan gitse bile alıcıda doğru sırada birleştirilir.	**Bottleneck (Darboğaz):** Bağlantı sürekli rezerve edildiği için yavaş bağlantılar diğer cihazları yavaşlatabilir.

TCP Paket Başlıkları (Headers)

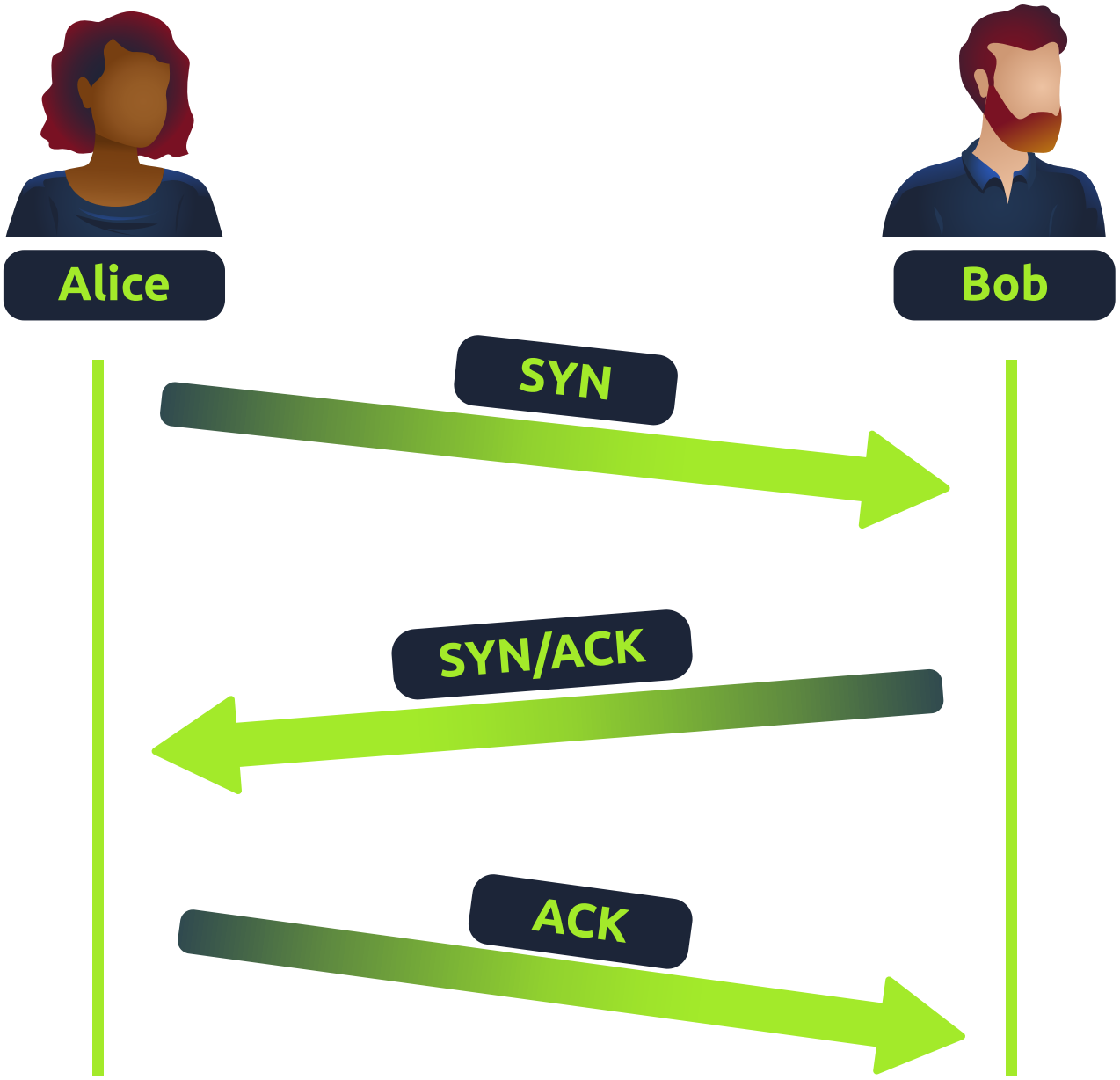
Bir TCP paketi, sadece veriden ibaret değildir; içinde iletişimi yöneten "etiketler" bulunur.

- **Source Port (Kaynak Port):** Gönderen cihazın rastgele (0-65535 arası, boşta olanlardan) seçtiği porttur.
- **Destination Port (Hedef Port):** Paketin gideceği servisin portudur. Rastgele değildir (Örn: HTTP için **80**, HTTPS için **443**).
- **Source/Destination IP:** Gönderen ve alan cihazların adresleri.
- **Sequence Number (Dizi Numarası):** Paketlerin sırasını takip etmek için ilk pakete atanan rastgele sayı.
- **Acknowledgement Number (Onay Numarası):** Alınan verinin teyidi için Sequence Number + 1 olarak belirlenen sayı.
- **Checksum:** Verinin yolda bozulup bozulmadığını kontrol eden matematiksel imza.
- **Flags (Bayraklar):** Paketin amacını belirtir (Bağlantı kurma, bitirme, sıfırlama vb.).
- **Data (Veri):** Asıl taşınan içerik (dosya parçası, mesaj vb.).

Üçlü El Sıkışma (Three-Way Handshake) Adımları

Bu süreç, iki yabancının konuşmaya başlamadan önce birbirine "Merhaba", "Merhaba, duyuyorum", "Tamam, ben de seni duyuyorum" demesine benzer.

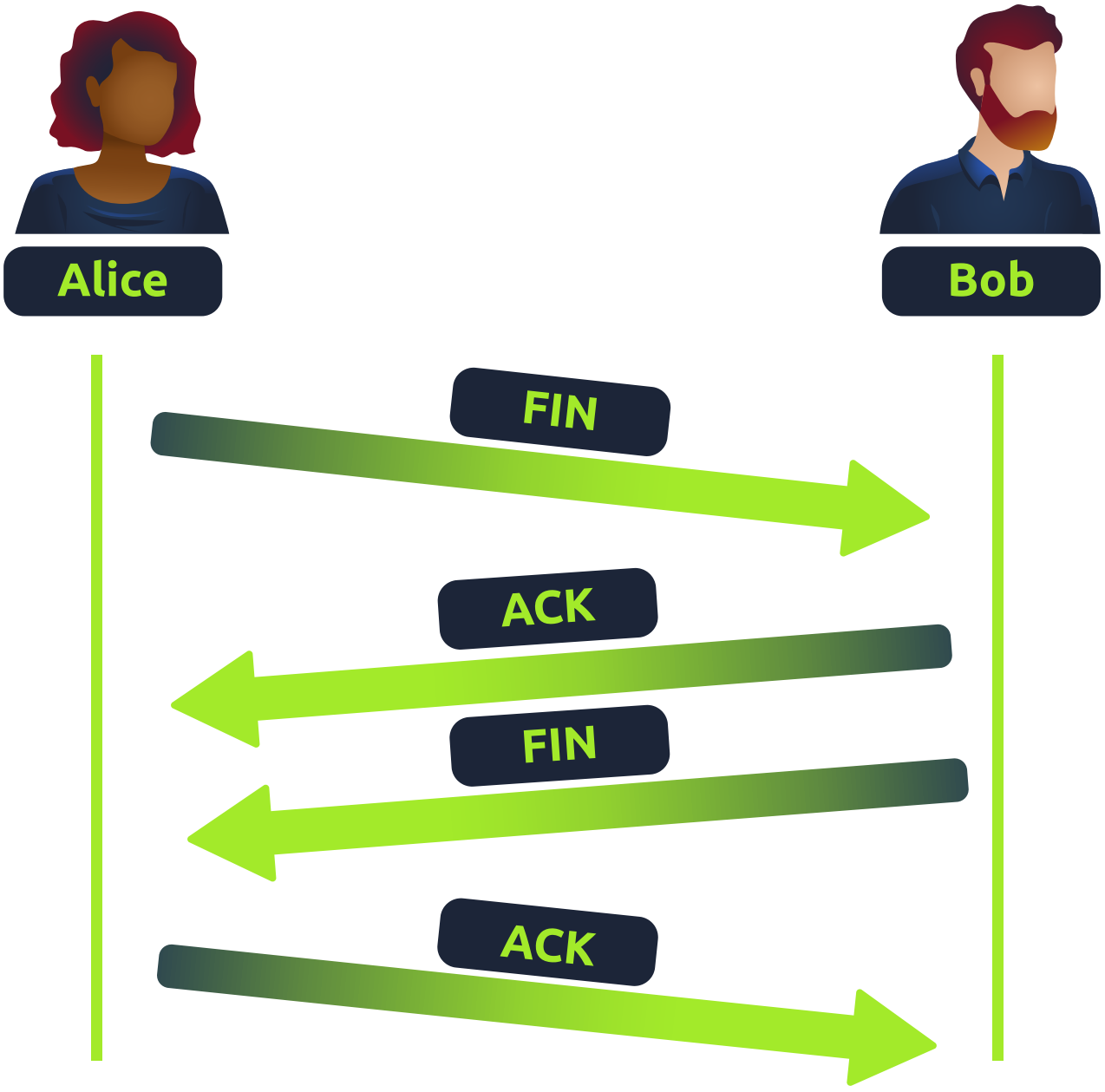
1. **SYN (Synchronize):** İstemci, sunucuya bağlantı kurmak istediğini belirten bir paket gönderir. İçinde kendi Başlangıç Dizi Numarası (**ISN**) bulunur (Örn: 0).
2. **SYN/ACK (Synchronize/Acknowledge):** Sunucu isteği alır; "Seninle senkronize olmayı kabul ediyorum ve ilk paketini aldığımı onaylıyorum" der. Kendi ISN değerini gönderir (Örn: 5000) ve istemcinin numarasını 1 artırarak onaylar (**ACK: 1**).
3. **ACK (Acknowledge):** İstemci, sunucunun cevabını aldığını onaylar. Artık bağlantı kurulmuştur (**Established**). Bundan sonraki paketler **DATA** bayrağı ile gönderilir.



Bağlantıyı Sonlandırma ve Hata Durumları

TCP bağlantıları sistem kaynaklarını (RAM, CPU) rezerve ettiği için iş bitince kapatılmalıdır.

- **FIN (Finish):** İletişimi düzgünce (cleanly) kapatmak için kullanılır. "Benim gönderecek verim bitti" demektir. Karşı taraf da onaylayınca bağlantı kapanır.
- **RST (Reset):** Bağlantıyı aniden (abruptly) koparır. Genelde bir hata oluştuğunda veya kapalı bir porta paket gönderildiğinde "Hattı kes" anlamında kullanılır.



> **Neden Önemli?** Bir siber güvenlikçi için bu süreci bilmek kritik; çünkü Nmap taramaları (SYN Scan), paket analizleri ve hatta DoS saldırıları (SYN Flood) tamamen bu mantık üzerine kuruludur.

UDP/IP (User Datagram Protocol)

UDP (User Datagram Protocol), TCP'nin "hızlı ama umursamaz" kardeşidir. TCP'nin aksine, UDP **stateless** (durumsuz) bir protokoldür; yani iki cihaz arasında verinin gönderilmesi için sürekli bir bağlantı kurulmasına veya el sıkışmasına gerek duymaz. Veriyi yola çıkarır ve ulaşıp ulaşmadığını takip etmez.

UDP'nin Karakteristiği: Hız ve Esneklik

UDP, "gönder ve unut" (fire and forget) mantığıyla çalışır. Bu protokolde **Three-way handshake (Üçlü El Sıkışma)** gerçekleşmez, cihazlar arasında senkronizasyon kurulmaz. Bu durum, belirli kullanım senaryolarında UDP'yi vazgeçilmez kılar.

Avantaj ve Dezavantaj Tablosu

Avantajlar	Dezavantajlar
Hız: TCP'den çok daha hızlıdır çünkü onay mekanizmalarıyla vakit kaybetmez.	**Güvenilirlik Eksikliği:** Verinin karşı tarafa ulaşıp ulaşmadığını kontrol etmez.
Esneklik: Paketlerin ne hızda gönderileceği kontrolünü tamamen yazılıma (uygulama katmanına) bırakır.	**Veri Kaybı:** Paketler yolda kaybolursa veya sırası karışırsa, UDP bunu düzeltmeye çalışmaz.
Kaynak Tasarrufu: TCP gibi cihaz üzerinde sürekli bir bağlantı rezerve etmez, sistem kaynaklarını yormaz.	**Kötü Kullanıcı Deneyimi:** Stabil olmayan bağlantılarda ses/video kesilmelerine yol açar.

UDP Nerede Kullanılır?

UDP, veri kaybının "tolere edilebildiği" veya hızın, veri tamlığından daha önemli olduğu yerlerde kullanılır:

- **Video Akışı (Streaming):** Bir film izlerken tek bir pikselin verisinin kaybolması filmi durdurmamalıdır.
- **Sesli Sohbet (VoIP):** Konuşurken milisaniyelik ses kayıpları, tüm konuşmanın gecikmesinden daha kabul edilebilirdir.
- **Online Oyunlar:** Anlık tepkiler (lag olmaması) her şeyden önemlidir.
- **DNS Sorguları:** Hızlı yanıt almak için tercih edilir.

UDP Paket Yapısı ve Başlıkları (Headers)

UDP paketleri, TCP'ye göre çok daha basittir ve daha az başlık bilgisi taşır. Bu sadelik, paketlerin daha hızlı işlenmesini sağlar.

Başlık (Header)	Teknik Açıklama
Time to Live (TTL)	Paketin ağda sonsuza kadar dönüp trafiği tıkamaması için atanan yaşam süresi/sıçrama sayısıdır.
Source Address	Paketi gönderen cihazın IP adresi (yanıtın nereye döneceğini belirtir).
Destination Address	Paketin hedefindeki cihazın IP adresi.
Source Port	Gönderen cihazın rastgele (0-65535 arası) seçtiği çıkış portu.
Destination Port	Hedef cihazda çalışan servisin portu (Örn: DNS için **53**).
Data (Payload)	Taşınan asıl veri; örneğin bir ses dosyasının veya oyun verisinin baytları.

UDP Bağlantı Süreci: Stateless (Durumsuz) Akış

UDP'de bir bağlantı "kurulmaz", sadece veri akışı başlatılır.

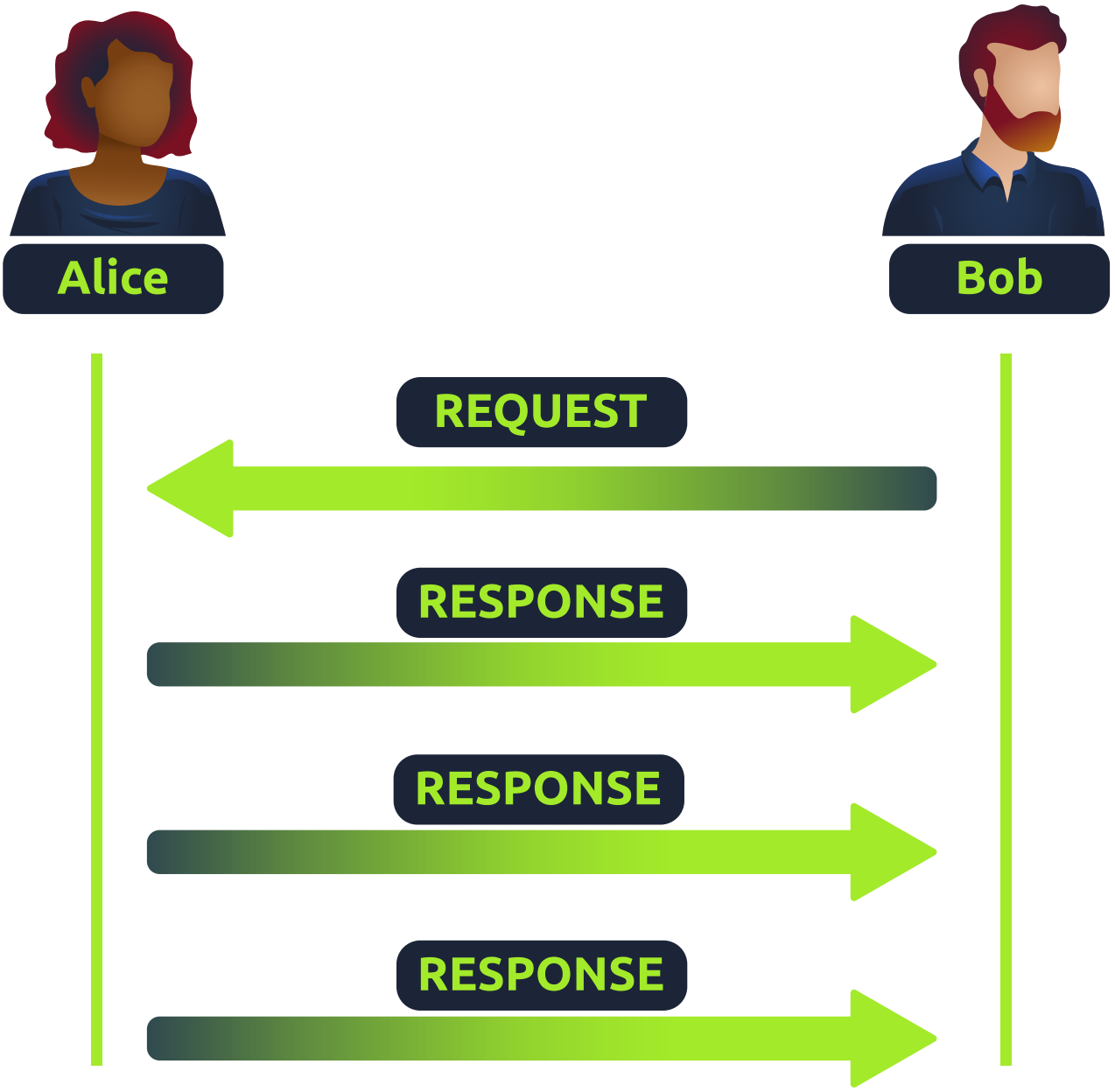
- **Onay Yok (No ACK):** Gönderici paketi gönderir, alıcı paketi aldığında "Aldım" diye bir mesaj (Acknowledgement) geri yollamaz.
- **Akış:** Alice, Bob'a veri paketlerini arka arkaya gönderir. Bob bu paketlerin bazılarını alamazsa veya paketler sırasız gelirse, Alice'in bundan haberi olmaz ve göndermeye devam eder.

Alice ve Bob Örneği (UDP Senaryosu)

1. **Alice:** "Sana bir video verisi gönderiyorum (Paket 1)."
2. **Alice:** "İşte devamı (Paket 2)."
3. **Alice:** "Ve son parça (Paket 3)."

   (Bu sırada Bob 2. paketi kaçırmış olsa bile Alice durmaz, Bob ise sadece eline geçenlerle resmi/videoyu oluşturmaya çalışır).

> **Siber Güvenlik Notu:** UDP'nin el sıkışma yapmaması, saldırganlar için bir avantaj olabilir. Örneğin **UDP Flooding** saldırılarında, saldırgan sahte kaynak IP'leri (spoofing) kullanarak hedefi paket yağmuruna tutabilir ve hedef sistem her paketi işlemek zorunda kalarak yorulur. TCP'deki gibi bir el sıkışma olmadığı için saldırganın kimliğini doğrulamak daha zordur.



Portlar 101: Mantık ve Pratik (Ports 101)

Ağ dünyasında portlar, verinin bir cihaza girdikten sonra hangi "kapıdan" geçeceğini ve hangi servisle buluşacağını belirleyen kritik noktalardır. Bir liman (harbour) düşünün; devasa bir cruise gemisi, balıkçı tekneleri için ayrılmış küçük bir iskeleye yanaşamaz. Her geminin boyutuna ve amacına uygun bir rıhtımı (port) vardır. Bilgisayarlarda da durum aynıdır: **Portlar, verinin nereye "park edeceğini" belirleyen kuralları dayatır.**

Portların Temel Mantığı ve Sayısal Aralıklar

İletişim kurulduğunda (OSI modelinde hatırladığımız üzere), gönderilen veya alınan her veri bir port üzerinden akar. Bilgisayar sistemlerinde portlar **0 ile 65535** arasında sayısal bir değerle temsil edilir.

- **Standartlaşma İhtiyacı:** Eğer hangi uygulamanın hangi portu kullanacağı belli olmasaydı, ağ tam bir kaos olurdu. Bu yüzden belirli yazılım ve davranışlar için standart portlar belirlenmiştir.

- **Örnek:** Web trafiği için **Port 80** standardı belirlendiği için, Google Chrome veya Firefox gibi farklı tarayıcılar bu porttan gelen veriyi aynı şekilde yorumlayabilir. Tasarım farklı olsa da veri iletişim kuralı tektir.

Yaygın Portlar (Common Ports)

0 ile 1024 arasındaki portlar "Common Ports" (Yaygın/Tanınmış Portlar) olarak bilinir ve dünya çapında kabul görmüş servislere ayrılmıştır.

Protokol	Port No	Açıklama
FTP (File Transfer Protocol)	**21**	Dosya paylaşımı için kullanılır. Merkezi bir sunucudan dosya indirmeyi/yüklemeyi sağlar.
SSH (Secure Shell)	**22**	Sistemlere metin tabanlı (CLI) güvenli uzaktan erişim ve yönetim için kullanılır.
HTTP (HyperText Transfer Protocol)	**80**	İnternetin (WWW) temelidir. Web sayfalarını indirmek için kullanılır.
HTTPS (HTTP Secure)	**443**	HTTP'nin şifrelenmiş, güvenli halidir.
SMB (Server Message Block)	**445**	FTP'ye benzer ama daha gelişmiştir; dosya paylaşımının yanı sıra yazıcı gibi cihazların paylaşımını da sağlar.
RDP (Remote Desktop Protocol)	**3389**	Sisteme görsel bir masaüstü arayüzü ile bağlanmayı sağlar (SSH'ın grafiksel versiyonu gibi).

Kritik Teknik Detay: Standart Dışı Port Kullanımı

Önemli bir siber güvenlik notu: Bu protokollerin standart portları olsa da, bu bir zorunluluk değil, bir **teamüldür.**

- Bir sistem yöneticisi, güvenlik veya özel bir yapılandırma için web sunucusunu **80** yerine **8080** portunda çalıştırabilir.
- **Bağlantı Kuralı:** Eğer standart dışı bir port kullanılıyorsa, bunu belirtmek için IP adresinden sonra **iki nokta üst üste (:)** işareti kullanılır.
  - *Örnek:* `http://10.10.10.10:8080`

Uygulama Senaryosu (Practical Challenge)

Bu görevde bir IP adresine belirli bir port üzerinden bağlanmanın mantığını kavrıyoruz.

Senaryo:

1. Verilen site arayüzünü aç.
2. Hedef IP: **8.8.8.8**

3. Hedef Port: **1234**
4. Bağlantı kurulduğunda sunucu bize gizli bayrağı (**flag**) verecektir.

Neden 1234?

Çünkü hedef sistemde bayrağı veren uygulama, standart portlar (80, 22 vb.) dışında özel bir portta (1234) dinleme (listening) yapacak şekilde ayarlanmıştır. Siber güvenlikte yaptığımız **Port Scanning (Port Tarama)** işlemlerinin amacı da tam olarak budur: Standart veya standart dışı hangi kapıların (portların) açık olduğunu ve arkasında hangi servisin çalıştığını bulmak.

> **Not:** Bir siber güvenlik öğrencisi olarak, her zaman **nmap** gibi araçlarla 0-1024 arasını değil, mümkünse tüm port aralığını (65535'e kadar) taramanın önemini unutma. Bazen en kritik açıklar, hiç beklenmedik yüksek numaralı portlarda gizlidir.