WİRESHARK VE AĞ ADLİ ANALİZİ

Teknik Analiz Raporu

Hazırlayanlar

İlayda Aytaş Kezban Nur Yahşi

14 Şubat 2026

WİRESHARK DA FİLTRELEME VE TCP

Temel Filtreleme Kavramları

Wireshark'ta filtreleme, ağ trafiğinden yalnızca ilgili paketleri görüntülemek için kullanılan güçlü bir özelliktir. İki ana filtreleme türü vardır:

- Capture Filter: Yakalama sırasında uygulanan filtreler (Berkeley Packet Filter syntax)
- Display Filter: Yakalanan paketleri görüntülerken uygulanan filtreler

TCP Filtreleme Örnekleri

```
tcp                 # Tüm TCP trafiği

tcp.port == 80        # 80 portundaki TCP trafiği

tcp.flags.syn == 1    # SYN flag'i set edilmiş paketler

tcp.flags.rst == 1    # RST flag'i set edilmiş paketler

tcp.stream eq 0       # Belirli bir TCP stream'i

ip.addr == 192.168.1.1 and tcp  # Belirli IP'den TCP trafiği
```

TCP (Transmission Control Protocol)

TCP'nin Temel Özellikleri

TCP, güvenilir, bağlantı odaklı bir transport layer protokolüdür:

- Güvenilir İletim: Paket kaybını tespit eder ve yeniden gönderir
- Sıralı Teslimat: Paketlerin doğru sırada teslim edilmesini garantiler
- Akış Kontrolü: Alıcının kapasitesine göre gönderim hızını ayarlar
- Hata Kontrolü: Checksum ile veri bütünlüğünü kontrol eder

TCP'nin OSI Modelindeki Yeri

- Layer 4 (Transport Layer): Uygulama katmanları arasında güvenilir iletişim sağlar
- End-to-End Protocol: Kaynak ve hedef arasında direkt iletişim kurur

TCP Başlık (Header) Yapısı

TCP protokolü, her veri paketinin başına bir "başlık" ekler. Bu başlık, paketin nereye gideceğini ve nasıl yönetileceğini belirleyen kontrol bilgilerini içerir. Standart bir TCP başlığı genellikle **20 bayt** boyutundadır.

Temel Bileşenler:

- **Kaynak ve Hedef Portlar:** Verinin hangi uygulamadan çıktığını ve hangisine gideceğini belirler (Örn: HTTP için 80, HTTPS için 443).
- **Sıra Numarası (Sequence Number):** Veri parçalarının doğru sırayla birleştirilmesini sağlar.
- **Onay Numarası (Acknowledgment Number):** Alınan verinin onaylandığını ve bir sonraki hangi baytın beklendiğini belirtir.
- **Kontrol Bayrakları (Flags):** Bağlantı durumunu yönetir (SYN, ACK, FIN, RST, PSH, URG).
- **Pencere Boyutu (Window Size):** Akış kontrolü için kullanılır; alıcının ne kadar veri kabul edebileceğini bildirir.

Bağlantı Kurulumu: Üçlü El Sıkışma (3-Way Handshake)

TCP, veri göndermeye başlamadan önce istemci ve sunucu arasında güvenli bir hat kurar. Bu sürece "üçlü el sıkışma" denir.

1. **SYN (Synchronize):** İstemci, bağlantı kurma isteğini belirtmek için bir sıra numarasıyla birlikte SYN paketi gönderir.
2. **SYN-ACK:** Sunucu isteği alır, kendi sıra numarasını oluşturur ve istemciye "Mesajını aldım, ben de hazırım" der.
3. **ACK (Acknowledge):** İstemci bu onayı alır ve bağlantının kurulduğunu teyit eden son bir paket gönderir. Artık veri transferi başlayabilir.

Bağlantıyı Sonlandırma

Veri iletimi bittiğinde bağlantı rastgele kesilmez; her iki tarafın da onayıyla kapatılır. Bu genellikle **Dörtlü El Sıkışma (4-way handshake)** ile yapılır:

1. Taraf A: **FIN** (Bitirelim mi?)
2. Taraf B: **ACK** (Tamam, isteğini aldım.)
3. Taraf B: **FIN** (Benim için de uygun, bitirebiliriz.)
4. Taraf A: **ACK** (Anlaşıldı, hoşça kal.)

RST (Reset) Paketi

RST Paketinin Amacı

RST paketi, TCP bağlantısının anormal şekilde sonlandırılması için kullanılır:

- Immediate Termination: Bağlantıyı derhal kapatır
- No Graceful Shutdown: Normal FIN/ACK süreci olmadan sonlanır
- Error Indication: Bağlantıda bir sorun olduğunu belirtir

RST Gönderilme Durumları

1. Port Kapalı

Client → Server: SYN

Server → Client: RST+ACK (Port kapalı)

2. Beklenmeyen Paket

Established Connection

Unknown Source → Server: Data

Server → Unknown Source: RST

3. Bağlantı Timeoutı

Client → Server: Data

(Uzun süre cevap yok)

Wireshark'ta RST Analizi

- Filter: `tcp.flags.reset == 1`

Client → Server: RST (Connection timeout)

PCAP İNCELEMESİ VE FİLE FORENSİCS

1. Wireshark ile PCAP İncelemesi

PCAP (Packet Capture) dosyaları, ağ üzerinden geçen tüm trafiğin ham kaydıdır. Analiz sürecinde izlenen temel adımlar şunlardır:

- **İstatistikleri Kontrol Etme:** "Protocol Hierarchy" ile hangi protokolün (TCP, HTTP, DNS vb.) ne kadar yoğun kullanıldığına bakılır.
- **Filtreleme:** `ip.addr == x.x.x.x` veya `http` gibi filtrelerle samanlıktaki iğne aranır.
- **Akışı Takip Etme (Follow Stream):** Parçalanmış paketleri birleştirerek tüm bir sohbeti veya veri transferini metin olarak görmemizi sağlar.

2. HTTP Object Export (Veri Çıkarma)

Eğer trafik şifrelenmemişse (HTTP), ağ üzerinden indirilen dosyaları sanki kendi bilgisayarımıza indirmişiz gibi geri kazanabiliriz.

Adımlar:

1. `File` -> `Export Objects` -> `HTTP...` yolunu izle.
2. Açılan listede trafikten geçen tüm görselleri, scriptleri ve dosyaları görebilirsin.
3. İstediğin dosyayı seçip `Save` diyerek bilgisayarına kaydedebilirsin.

3. JPG ve EXE Dosyalarını Çıkarma (File Carving)

Dosya adli tıbbında, ağ trafiği veya disk imajı içinden dosya çıkarma işlemine **Carving** denir.

- **JPG Çıkarma:** Trafikte `Content-Type: image/jpeg` ifadesi aranır. Export Objects ile kolayca alınır.
- **EXE Çıkarma:** Genellikle `application/x-msdownload` olarak görünür. Şüpheli bir `.exe` aktarımı, sistemin ele geçirildiğinin (payload iletimi) en büyük kanıtıdır.

4. Dosya Uzantısı Değiştirme ve Forensics Analizi

Saldırganlar genellikle güvenlik duvarlarını veya kullanıcıyı atlatmak için dosyaların uzantısını değiştirir. (Örn: `malware.exe` dosyasını `tatil_fotografı.jpg` yapmak).

Magic Bytes (Sihirli Baytlar) Kavramı

Dijital Kimlik Kartı: Magic Bytes & File Signatures

Bir dosyanın uzantısını (örneğin `.jpg`) istediğiniz an `.txt` olarak değiştirebilirsiniz. Ancak dosyanın içeriği değişmez. İşletim sistemleri ve analiz araçları, dosyanın gerçek türünü belirlemek için **Magic Bytes (Sihirli Baytlar)** adı verilen veri kalıplarını kullanır.

Temel Kavramlar

- **Magic Bytes:** Bir dosyanın en başında (Header) yer alan, o dosya formatına özgü sabit bayt dizisidir.
- **File Signature (Dosya İmzası):** Magic bytes ve dosyanın sonundaki bitiş işaretlerinin (Trailer/Footer) toplamına verilen isimdir.
- **Dosya Başlığı (Header) Kontrolü:** Dosyanın ilk birkaç baytının, bilinen imza veritabanıyla (örneğin Gary Kessler'ın listesi) karşılaştırılması işlemidir.

Sık Karşılaşılan Dosya İmzaları

Aşağıdaki tablo, en yaygın dosya türlerinin "maskesiz" hallerini gösterir:

Dosya Türü	Uzantı	Hexadecimal (Büyülü Baytlar)	ASCII Karşılığı
Windows Executable	.exe / .dll	`4D 5A`	**MZ**
PDF Belgesi	.pdf	`25 50 44 46`	**%PDF**
JPEG Görsel	.jpg / .jpeg	`FF D8 FF`	**ÿØÿ**
ZIP Arşivi	.zip	`50 4B 03 04`	**PK..**

Forensics Neden Buna Bakar?

Siber saldırganlar, zararlı yazılımları (malware) gizlemek için sıklıkla **"Uzantı Aldatmacası"** yaparlar.

Örneğin elinizde `rapor.docx` isimli bir dosya var ama Word ile açılmıyor.Dosyayı bir Hex Editor (Örn: HxD) ile açtığınızda ilk iki baytın `4D 5A` olduğunu görüyorsunuz. Bu da bize şunu gösterir dosya aslında bir **EXE (çalıştırılabilir program)** dosyasıdır! Saldırgan, kullanıcının dosyaya çift tıklayıp virüsü çalıştırması için uzantısını değiştirmiştir.

RSA ŞİFRELEME MEKANİZMASI VE MATEMATİKSEL TEMELLERİ

RSA (Rivest-Shamir-Adleman), modern kriptografinin temel yapı taşlarından biridir ve asimetrik şifreleme sisteminin en yaygın kullanılan uygulamasıdır. 1977 yılında geliştirilmiş olan bu algoritma, günümüzde hala internet güvenliğinin omurgasını oluşturmaktadır.

1. Asimetrik Şifreleme Prensipleri

RSA, iki farklı anahtarın kullanımına dayanan asimetrik bir şifreleme sistemidir. Bu sistem, geleneksel simetrik şifreleme yöntemlerinden (tek anahtar kullanan sistemlerden) temel bir farklılık gösterir.

Public Key (Açık Anahtar)

Açık anahtar, herkesle paylaşılabilen ve yalnızca veriyi şifrelemek için kullanılan anahtardır. Bu anahtarın dağıtımında herhangi bir güvenlik riski yoktur çünkü açık anahtar ile şifrelenen bir veri, yalnızca karşılık gelen özel anahtar ile çözülebilir. İstemciler, sunucu ile güvenli iletişim kurmak için bu anahtarı kullanırlar.

Private Key (Özel Anahtar)

Özel anahtar, yalnızca sunucu tarafında gizli tutulan ve açık anahtar ile şifrelenmiş veriyi çözmek için kullanılan anahtardır. Bu anahtarın gizliliği, sistemin tüm güvenliğini garanti eder. Özel anahtarın ele geçirilmesi, o sunucu ile yapılan tüm şifreli iletişimin çözülebilmesi anlamına gelir.

2. Matematiksel Temel: Asal Sayı Çarpanlarına Ayırma Zorluğu

RSA'nın güvenliği, tamamen bir matematiksel zorluğa dayanır:

İki büyük asal sayının çarpımı kolaydır, ancak bu çarpımın sonucunu tekrar asal çarpanlarına ayırmak son derece zordur.

Örneğin, küçük sayılarla:

• 17 × 19 = 323 (çarpma kolay)

• 323'ü 17 ve 19'a ayırmak (faktörizasyon görece kolay)

Ancak modern RSA şifrelemede kullanılan sayılar 2048, 3072 veya 4096 bit uzunluğundadır. Bu seviyedeki bir sayıyı asal çarpanlarına ayırmak, günümüzün en güçlü süper bilgisayarları için bile pratik olarak imkansızdır ve milyonlarca yıl sürebilir. İşte RSA'nın gücü buradan gelmektedir.

Bu matematiksel zorluk sayesinde, açık anahtar herkesle paylaşılabilir, ancak özel anahtar bu açık anahtardan elde edilemez. Saldırganların özel anahtara ulaşabilmesi için dev sayıların faktörizasyonunu gerçekleştirmeleri gerekir ki bu da mevcut teknoloji ile imkansızdır.

TLS/SSL EL SIKIŞMA (HANDSHAKE) VE ANAHTAR DEĞİŞİMİ

HTTPS bağlantısı kurulurken izlenen TLS/SSL el sıkışma süreci, iki tarafın güvenli bir iletişim kanalı oluşturmasını sağlar. Bu süreç, önce asimetrik şifreleme ile güvenli bir kanal açar, ardından performans nedeniyle simetrik şifrelemeye geçiş yapar.

1. El Sıkışma Aşamaları

TLS el sıkışması dört temel aşamadan oluşur ve her aşama farklı mesaj tipleri ile tanımlanır:

Aşama	Mesaj Tipi	Açıklama
1. Client Hello	Tip 1	İstemci sunucuya şifreli konuşma isteği gönderir, desteklediği TLS versiyonlarını, şifreleme yöntemlerini (cipher suites) ve Client Random değerini sunar.
2. Server Hello	Tip 2	Sunucu kullanılacak TLS versiyonunu ve şifreleme yöntemini seçer, dijital sertifikasını (Public Key dahil) ve Server Random değerini gönderir.
3. Client Key Exchange	Tip 16	İstemci, ürettiği Premaster Secret'ı sunucunun Public Key'i ile RSA şifrelemesi kullanarak şifreler ve gönderir. Sadece sunucu bu veriyi çözebilir.
4. Şifreli İletişim	-	Her iki taraf da Master Secret'ı üreterek AES gibi hızlı simetrik şifreleme algoritmasına geçer ve bundan sonraki tüm iletişim bu anahtar ile şifrelenir.

2. Anahtar Materyalleri ve Güvenlik Mekanizması

El sıkışma sürecinde üç kritik rastgele değer kullanılır:

Client Random

İstemci tarafından üretilen ve Client Hello mesajında gönderilen rastgele bir sayıdır. Bu değer şifrelenmemiş olarak ağ üzerinde iletilir.

Server Random

Sunucu tarafından o anki bağlantıya özel olarak üretilen rastgele bir sayıdır. Her yeni bağlantıda farklı bir değer alır ve Server Hello mesajında açık olarak gönderilir. Bu değer de şifrelenmemiştir.

Premaster Secret

İstemci tarafından üretilen 48 byte uzunluğundaki kritik rastgele veridir. Bu değer, sunucunun açık anahtarı (Public Key) ile RSA şifrelemesi kullanılarak şifrelenir ve Client Key Exchange mesajında gönderilir. Önemli olan şu ki:

- Sadece sunucu, kendi özel anahtarı (Private Key) ile bu şifreli veriyi çözebilir
- İstemci bile şifrelediği bu veriyi geri çözemez (tek yönlü şifreleme)
- Ağı dinleyen saldırganlar şifreli Premaster Secret'ı görebilir ama çözemez

Master Secret (Ana Gizli Anahtar)

Hem istemci hem sunucu, Premaster Secret, Client Random ve Server Random değerlerini bir hash fonksiyonundan geçirerek Master Secret'ı üretir. İki taraf da matematiksel olarak aynı sonuca ulaştığı için aynı Master Secret'a sahip olur. Bu ana gizli anahtardan, simetrik şifreleme için kullanılacak session anahtarları türetilir.

3. Saldırganların Konumu ve Güvenlik

Bu noktada kritik bir soru ortaya çıkar: Ağı dinleyen bir hacker ne görebilir?

✓ Client Random'u görebilir (şifrelenmemiş)

✓ Server Random'u görebilir (şifrelenmemiş)

✓ Şifrelenmiş Premaster Secret'ı görebilir

✗ Ancak Premaster Secret'ı çözemez (RSA ile korumalı)

✗ Dolayısıyla Master Secret'ı üretemez

Sonuç olarak, Client Random ve Server Random gibi açık veriler tek başına işe yaramaz. Kritik olan Premaster Secret, RSA şifrelemesi ile korunduğu için saldırganlar hiçbir şifreyi kıramaz. Sistem bu şekilde güvenliğini sağlar.

WIRESHARK İLE HTTPS TRAFİĞİNİN ANALİZİ

Wireshark, ağ trafiğini yakalayan ve analiz eden en popüler açık kaynaklı araçlardan biridir. Normal şartlarda HTTPS trafiği şifreli olduğu için Wireshark'ta 'Encrypted Application Data' olarak görünür ve içeriği okunamaz. Ancak özel koşullar altında bu trafik deşifre edilebilir.

1. HTTPS Deşifresi İçin Gereksinimler

Şifreli HTTPS trafiğini çözmek için şu kritik unsura ihtiyaç vardır:

Sunucunun RSA Private Key (Özel Anahtar) dosyası

Bu anahtar genellikle .pem, .key veya .p12 formatlarında olabilir. Özel anahtara sahip olunduğunda, Premaster Secret deşifre edilebilir ve dolayısıyla Master Secret yeniden hesaplanarak tüm session trafiği çözülebilir.

2. Wireshark Yapılandırma Adımları

HTTPS trafiğini deşifre etmek için izlenmesi gereken adımlar:

Adım	İşlem
1	Wireshark menüsünden Edit → Preferences seçeneğine gidin
2	Sol menüden Protocols → TLS sekmesini bulun ve açın
3	RSA Keys List alanında Edit butonuna tıklayın
4	Sunucu bilgilerini girin: IP Address (sunucu IP'si), Port (443 veya ilgili port), Protocol (http), Key File (özel anahtar dosyasının yolu)
5	Ayarları kaydedin ve PCAP dosyasını yeniden yükleyin veya canlı yakalamanızı yeniden başlatın

Doğru yapılandırma sonrasında, önceden 'Encrypted Application Data' olarak görünen paketler 'Decrypted Application Data' olarak görünür ve HTTP mesajları (GET, POST, vb.) okunabilir hale gelir.

3. Güvenlik İçgörüleri

Bu analiz, TLS/SSL güvenliğinin tamamen sunucu tarafındaki Private Key'in gizliliğine dayandığını gösterir. Eğer bir saldırgan bu anahtara erişebilirse, o sunucu ile yapılan tüm geçmiş ve gelecek iletişimleri çözebilir. Bu nedenle sunucu anahtarlarının korunması kritik öneme sahiptir.

NMAP PORT TARAMA TEKNİKLERİ VE SALDIRI İMZALARI

Nmap (Network Mapper), ağ keşfi ve güvenlik denetimi için kullanılan en yaygın araçlardan biridir. Port tarama, bir hedef sistemdeki açık portları tespit etmek için kullanılır ve saldırganlar için ilk keşif aşamasıdır.

1. Attack Signature (Saldırı İzi) Kavramı

Nmap çalıştırıldığında, ağ trafiğinde belirgin TCP kalıpları oluşur. Wireshark gibi araçlarla bu kalıplar tespit edildiğinde, ağda bir port taraması yapıldığı anlaşılır. Bu karakteristik kalıplara 'attack signature' (saldırı izi veya saldırı imzası) denir.

2. SYN Scan (-sS) - Stealth Tarama

SYN Scan, Nmap'in en popüler tarama yöntemidir ve 'yarı açık tarama' (half-open scan) olarak da bilinir.

Çalışma Mantığı

Normal TCP bağlantısı:

İstemci → SYN → Sunucu Sunucu → SYN/ACK → İstemci İstemci → ACK → Sunucu (Bağlantı kuruldu)

SYN Scan ise:

İstemci → SYN → Sunucu Sunucu → SYN/ACK → İstemci İstemci → RST (Bağlantıyı sonlandır)

Yani tam bağlantı kurulmaz, sadece 'port açık mı?' sorusuna cevap alınır. Bu yöntem 'stealth' (sessiz) olarak kabul edilir çünkü bazı eski sistemlerde log kaydı oluşturmaz.

Wireshark'ta Görünümü

• Aynı kaynak IP'den hedefe çok sayıda SYN paketi

• Farklı portlara yönlendirilmiş ardışık SYN paketleri

• ACK paketlerinin gelmemesi (RST ile sonlandırma)

Yetki Gereksinimleri

SYN Scan çalıştırmak için root (veya sudo) yetkisi gerekir çünkü ham paket (raw packet) oluşturma yetkisi gerektirir.

3. Connect Scan (-sT) - Tam Bağlantı Taraması

Connect Scan, tam TCP bağlantısı kuran tarama yöntemidir.

Çalışma Mantığı

Bu yöntemde üçlü el sıkışma (three-way handshake) tamamen gerçekleştirilir:

SYN → SYN/ACK → ACK Ardından bağlantı kapatılır.

SYN Scan vs Connect Scan

Özellik	SYN Scan (-sS)	Connect Scan (-sT)
Bağlantı Tipi	Yarım bağlantı	Tam bağlantı
ACK Paketi	Yok (RST gönderilir)	Var (tam el sıkışma)
Tespit Edilebilirlik	Daha stealth (sessiz)	Daha gürültülü
Yetki Gereksinimi	Root gerekir	Root gerekmez
Hız	Daha hızlı	Daha yavaş

4. Port Tarama Tespiti

Wireshark'ta port taramasını tespit etmenin klasik işaretleri:

• Tek kaynak IP adresinden tek hedef IP'ye yönelik trafik

• Sürekli değişen hedef port numaraları (örn: 80, 443, 22, 21, 53, 3306, 8080...)

• Kısa zaman diliminde çok sayıda bağlantı denemesi

• Ardışık SYN paketleri (SYN Scan için)

5. Wireshark Filtreleme Teknikleri

Port taramasını tespit etmek için kullanılabilecek Wireshark filtreleri:

SYN Tarama Filtresi

tcp.flags.syn==1 and tcp.flags.ack==0

Bu filtre yalnızca SYN flag'i açık olan ancak ACK flag'i olmayan paketleri gösterir. Bu, SYN Scan'in klasik imzasıdır.

Nmap Varsayılan Davranışı

Root yetkisiyle çalıştırıldığında:

sudo nmap hedef → Otomatik olarak SYN Scan (-sS) yapar

Root yetkisi olmadan çalıştırıldığında:

nmap hedef → Mecburen Connect Scan (-sT) yapar

WIRESHARK'TA TCP STREAM TAKİBİ VE ANALİZ TEKNİKLERİ

Ağ trafiği analiz edilirken, paketler genellikle parçalı ve dağınık halde görünür. Wireshark'ın 'Follow TCP Stream' özelliği, tek bir TCP bağlantısındaki tüm veriyi birleştirerek okunabilir hale getirir.

1. Follow TCP Stream Nedir?

Normal halde Wireshark'ta paketler şu şekilde dağınık görünür:

Paket 1: 'Mer'

Paket 2: 'ha'

Paket 3: 'ba'

Paket 4: 'nasıl'

Paket 5: 'sın?'

Follow TCP Stream ile bu paketler birleştirilerek:

'Merhaba nasılsın?' → Tek parça halinde!

2. Kullanım Adımları

Adım 1: Wireshark'ta herhangi bir TCP paketine sağ tıklayın

Adım 2: 'Follow' menüsünü seçin

Adım 3: 'TCP Stream' seçeneğine tıklayın

Sonuç olarak, o TCP bağlantısındaki tüm veri akışı tek bir pencerede görüntülenir.

3. Renk Kodları

Follow TCP Stream penceresi, iletişim yönünü ayırt etmek için renk kodlaması kullanır:

Kırmızı metin: İstemciden sunucuya giden veri

Mavi metin: Sunucudan istemciye gelen veri

Bu renk ayrımı sayesinde, hangi tarafın ne gönderdiği kolayca anlaşılabilir.

4. TCP Stream Filtreleme

Wireshark, her TCP bağlantısını otomatik olarak numaralandırır (stream 0, stream 1, stream 2, vb.). Belirli bir stream'e odaklanmak için filtre kullanılabilir:

tcp.stream eq 2 and ip.addr eq 192.168.1.158

Bu filtre şu anlama gelir:

• tcp.stream eq 2 → 3. TCP bağlantısını göster (indeks 0'dan başlar)

• ip.addr eq 192.168.1.158 → Bu IP adresini içeren paketleri göster

Stream numarası, iki cihaz arasındaki tüm konuşmayı temsil eder. Tek bir TCP bağlantısına ait tüm paketleri bir arada görüntülemeyi sağlar.

5. Kullanım Senaryoları

• HTTP isteklerini ve yanıtlarını tam olarak okuma

• FTP veya SMTP gibi metin tabanlı protokollerdeki komutları görme

• Şifrelenmemiş veri akışlarını analiz etme

• Uygulama katmanı iletişimini debuglama

PROTOCOL HIERARCHY STATISTICS

Wireshark'ın 'Protocol Hierarchy Statistics' özelliği, yakalanan tüm ağ trafiğinde hangi protokollerin ne kadar kullanıldığını gösteren özet bir rapordur.

1. Erişim ve Kullanım

Bu rapora erişmek için:

Statistics → Protocol Hierarchy

2. Sağladığı Bilgiler

Protocol Hierarchy raporu şunları gösterir:

• Her protokolün toplam paket sayısı
• Her protokolün toplam byte miktarı
• Trafiğin yüzdesel dağılımı
• Protokollerin hiyerarşik ilişkisi (örn: TCP içinde HTTP, HTTPS, SSH...)

3. Pratik Kullanım Alanları

Bu istatistik özellikle şu durumlarda faydalıdır:

Ağ trafiğinin genel kompozisyonunu anlama: PCAP dosyasında hangi protokollerin dominant olduğunu hızlıca görmek

Anormal trafik tespiti: Beklenmeyen protokollerin varlığını tespit etmek (örn: normalden fazla DNS trafiği, beklenmeyen SMB trafiği)

Performans analizi: Hangi protokollerin bant genişliğinin çoğunu kullandığını görmek

Güvenlik analizi: Şifreli (TLS/SSL) ve şifresiz trafik oranlarını karşılaştırmak

Hızlı ön değerlendirme: Büyük PCAP dosyalarını analiz etmeden önce genel bir fikir edinmek

SONUÇ VE DEĞERLENDİRME

Bu teknik inceleme, modern internet güvenliğinin temel yapı taşlarını ve pratik ağ güvenliği analiz tekniklerini kapsamlı bir şekilde ele almıştır. Çalışmada elde edilen ana çıkarımlar şunlardır:

Modern internet güvenliği, matematiksel kriptografi, protokol tasarımı ve pratik uygulama tekniklerinin bir bileşimidir. TLS/SSL protokolü, teorik kriptografik prensipleri pratik güvenlik çözümlerine dönüştürmüş ve milyarlarca kullanıcının verilerini korumaktadır.

Wireshark ve Nmap gibi araçlar, hem savunma hem de saldırı perspektifinden ağ güvenliğini anlamak için kritik öneme sahiptir. Bu araçların doğru kullanımı, güvenlik uzmanlarının sistemleri korumalarına, zafiyetleri tespit etmelerine ve olası saldırılara karşı önlem almalarına olanak tanır.

Sonuç olarak, internet güvenliği statik bir alan değil, sürekli gelişen ve yenilenen dinamik bir disiplindir. Bu raporda ele alınan konular, siber güvenlik uzmanlarının sahip olması gereken temel bilgi ve becerilerin sadece bir kısmını oluşturmaktadır.