Yayıncı Avı Projesi

**Hazırlayanlar:**

* Ali Kemal Turan (2420191028)
* Fatma Yeliz Apaydın (2420191038)
* Burhan Yavaş (2420191018)

## **1. Proje Amacı**

Yayıncı Avı Projesi’nin amacı, çevrim içi yayın platformlarında (özellikle Twitch) gerçekleşen yayın trafiğini analiz ederek, yayıncı sunucularının IP aralıklarını, sunucu yapılandırmalarını ve kullanılan portları tespit etmek; bu tespitleri Wireshark ile manuel analiz ve Python tabanlı otomasyon ile gerçekleştirmektir.

## **2. Kullanılan Araçlar ve Yöntemler**

#### ***Kullanılan Dosyalar ve İçerikleri***

**Göruntuleme Filtreleri.txt**: Görüntüleme için kullanılan filtreler (örneğin, IP aralıkları ve portlar).

* **Yakalama filtreleri.txt**: Trafik yakalama için kullanılan filtreler.
* **https\_filtered\_raw.pcapng**: HTTPS trafiğine ait filtrelenmiş ham veri (341 KB).
* **rtmp\_filtered.pcapng**: RTMP trafiğine ait filtrelenmiş ham veri (39.754 KB).
* **twitch\_only\_rtmp\_filtered.pcapng**: Yalnızca belirli platforma ait RTMP trafiği (1.721 KB).
* **filtered\_tcp443.pcapng**: TCP 443 portuna ait filtrelenmiş trafik (8.809 KB).
* **twitch\_raw\_capture.pcapng**: Ham ağ trafiği yakalaması (41.421 KB).
* **ips.txt**: Tespit edilen IP adresleri listesi.
* **ips\_platform\_analysis.pdf**: IP analizleri ve coğrafi bilgiler (140 KB).

1. **Wireshark Analizi**
   1. Ham trafik yakalama dosyaları oluşturuldu:
      1. twitch\_raw\_capture.pcapng
      2. normal\_traffic.pcap
      3. capture.pcap
   2. **Capture Filter (Yakalama Filtreleri.txt)**
      1. Basit: tcp port 443
      2. Gelişmiş: tcp port 443 and (net 3.0.0.0 mask 255.0.0.0 or net 52.0.0.0 mask 255.0.0.0 or ...)
   3. **Display Filter (Görüntüleme Filtreleri.txt)**
      1. Basit: tcp.port == 443
      2. Gelişmiş: dns.qry.name contains "twitch.tv" || tls.handshake.extensions\_server\_name contains "twitch.tv"
      3. Final: DNS/TLS sorguları ve IP blokları aralığı mantıksal OR ile birleştirilmiş filtre.
2. **WHOIS ve CIDR Blok Tespiti**
   1. Wireshark’ta yakalanan örnek IP’ler Bulk ip lookup.txt ve WhoIS verilerine göre bulunan ip aralıkları.txt dosyalarında toplandı.
   2. ips\_platform\_analysis.pdf ile coğrafi ve servis bazlı analiz yapıldı.
   3. NetRange aralıkları CIDR formatına dönüştürüldü ve ips.txt içine eklendi.
3. **Python Otomasyon Script**
   1. sniff\_twitch\_final.py: Scapy ile canlı trafik dinleme
      1. CIDR destekli IP kontrolü (ipaddress modülü)
      2. DNS katmanında twitch.tv içeren sorguların tespiti
      3. Eşleşen paketlerin capture.pcap dosyasına yazılması
      4. sniff\_log.txt dosyasına zaman damgalı loglama (UTF-8)
      5. detected\_ports.txt ve detected\_twitch\_ports.txt ile port takibi
      6. 120 saniye sonra otomatik durma veya Ctrl+C ile manuel kesme
4. **Ek Rehber ve Analiz Dosyaları**
   1. sniff\_test\_twitch\_guide.txt, general\_sniff\_guide.txt: Script kullanımı ve inceleme adımları
   2. Not.txt: Çalışma notları

## **3. Proje Klasör Yapısı ve Üretilen Dosyalar**

Yeni/ # Proje ana dizini  
├─ Captures/ # Wireshark yakalama sonuçları  
│ ├─ filtered\_tcp443.pcapng  
│ ├─ https\_filtered\_raw.pcapng  
│ ├─ rtmp\_filtered.pcapng  
│ └─ twitch\_raw\_capture.pcapng  
├─ lookup verileri/ # IP araştırma ve WHOIS verileri  
│ ├─ Bulk ip lookup.txt  
│ ├─ WhoIS verilerine göre bulunan ip aralıkları.txt  
│ ├─ ips\_platform\_analysis.pdf  
│ ├─ Not.txt  
│ └─ Detaylı rapor/ # Ek rapor ve analiz  
├─ sniff\_test\_twitch (python otomasyon)/ # Otomasyon script ve çıktılar  
│ ├─ sniff\_twitch\_final.py  
│ ├─ ips.txt  
│ ├─ capture.pcap  
│ ├─ sniff\_log.txt  
│ ├─ detected\_ports.txt  
│ ├─ detected\_twitch\_ports.txt  
│ └─ Otomasyon Capture/ # Script çalışması sonrası klasör  
│ └─ ... # Otomatik oluşturulan pcap dosyaları  
├─ Görüntüleme Filtreleri.txt # Wireshark display filtre şablonları  
└─ Yakalama Filtreleri.txt # Wireshark capture filtre şablonları

## **4. Elde Edilen Bulgular**

* TLS SNI ve DNS sorguları kullanılarak doğrudan Twitch trafiği ayrıştırıldı.
* Amazon altyapısına ait IP blokları (/9, /10, /11, /12, /14 aralıkları) başarıyla belirlendi.
* Python script ile hem IP blokları hem de DNS sorgularını eşleyerek .pcap ve log dosyaları oluşturuldu.
* Windows 11 ortamında Türkçe karakter ve BOM kaynaklı hatalar giderildi.

## **5. Sonuç**

* Proje, Twitch yayın trafiğinin izlenmesi için hem Wireshark hem de Python otomasyonunu entegre eden eksiksiz bir çözüm sundu.