PacketSentry IDS/IPS Projesi Raporu

1. Kapak Sayfası

- **Başlık:** PacketSentry IDS/IPS Projesi Raporu
- **Adın:** Kadir Arıcı
- **Tarih:** 1 Haziran 2025
- **Versiyon:** 1.0
- **Hazırlayan:** Kadir Arıcı

2. İçindekiler

3. Giriş

Bu rapor, PacketSentry adlı ağ tabanlı bir Saldırı Tespit ve Önleme Sistemi (IDS/IPS) projesinin geliştirme sürecini ve sonuçlarını içermektedir. Proje, ağ trafiğini analiz ederek tehditleri tespit etme ve engelleme yeteneğine sahip bir sistem tasarlamayı amaçlamıştır. 1 Haziran 2025 itibarıyla, proje 5 modülden oluşmakta ve performans testi ile tamamlanmıştır. Proje, güvenlik analistlerine ve ağ yöneticilerine ağ güvenliğini artırma konusunda yardımcı olmayı hedeflemektedir.

4. Proje Amacı ve Kapsamı

Projenin temel amacı, ağ trafiğini gerçek zamanlı olarak izleyerek SQL Injection, XSS, Brute Force gibi tehditleri tespit etmek ve bu tehditleri engellemektir. Sistem, Linux, macOS ve Windows platformlarında çalışacak şekilde tasarlanmıştır. Kapsam, tehdit tespiti, engelleme, performans analizi ve platform bağımsızlığı olmak üzere dört ana bileşeni içermektedir. Ayrıca, sistemin kullanıcı dostu olması ve kolay yapılandırılabilir olması da hedeflenmiştir.

5. Geliştirilen Modüller

5.1. Modül 1: Arayüz Seçimi ve Paket Yakalama

- **Dosya:** `src/analyze_traffic.py`
- **Açıklama:** Bu modül, ağ arayüzlerini otomatik olarak tespit eder, kullanıcıya arayüz seçme imkanı sunar ve belirtilen arayüzden paketleri yakalar. TCP (port 80 ve 443) ve UDP trafiğini hedefler. Snort'un HTTP kuralıyla (`alert tcp any any -> any 80`) uyumlu çalışarak IDS/IPS analizi için temel oluşturur.
- **Özellikler:** Komut satırı desteği, otomatik filtreleme, akıllı dış ağ testi, önbellekleme, kullanıcı doğrulaması.

5.2. Modül 2: Tehdit Tespiti

- **Dosya:** `src/detect_attacks.py`
- **Açıklama:** Bu modül, ağ paketlerini analiz ederek SQL Injection, XSS, Brute Force gibi tehditleri tespit eder. Regex tabanlı kurallar ve Threat Intelligence API (henüz tam olarak entegre edilmedi) kullanılarak geliştirilmiştir.
- **Özellikler:** Esnek kural sistemi, tehdit puanlama, payload kodlama toleransı, zaman damgalı ve renkli çıktılar.

5.3. Modül 3: Tehdit Engelleme

- **Dosya:** `src/detect_and_block.py`
- **Açıklama:** Tespit edilen tehditlere göre IP engelleme yapar. Linux için iptables, macOS için pfctl ve Windows için netsh komutları desteklenir.
- **Özellikler:** Kalıcı engelleme, otomatik temizleme, JSON loglama detayları, macOS ve Windows için spesifik kurallar, performans iyileştirmesi.

5.4. Modül 4: Performans Testi

- **Dosya:** `src/performance_test.py`
- **Açıklama:** Sistem performansını ölçmek için saniyede işlenen paket (PPS), CPU kullanımı ve bellek tüketimi analiz edilmiştir.
- **Özellikler:** Kaynak izleme, test raporlama, komut satırı seçenekleri, Markdown formatında sonuç kaydetme.

6. Karşılaşılan Sorunlar ve Çözümler

- **Sorun:** Arayüz seçimi sırasında dış ağ testi başarısız oluyordu.
  - **Çözüm:** Ping ve HTTP testleri paralel olarak çalıştırılarak alternatif arayüz önerildi.
- **Sorun:** Windows'ta `netsh` komutları tutarsız çalışıyordu.
  - **Çözüm:** Hedef portun belirtilmesi zorunlu kılındı ve ek loglama eklendi.
- **Sorun:** Düzenleme aracı (edit_file) ile dosya güncelleme sırasında eşleşme hataları yaşanıyordu.
  - **Çözüm:** Dosya içeriği tekrar okunarak, tam eşleşen ORIGINAL bloklar kullanıldı. Alternatif olarak, manuel düzenleme önerildi.
- **Sorun:** Sanal ortam (venv) bulunamadı hatası.
  - **Çözüm:** Doğru dosya yolları kullanıldı ve sanal ortamın etkinleştirilmesi sağlandı.

7. Performans Testi Sonuçları

- **Test Ortamı:** macOS Monterey, 2.3 GHz Dual-Core Intel Core i5, 8 GB RAM
- **Test Türü:** Canlı trafik testi (arayüz: en0)
- **Test Süresi:** 5 saniye
- **Toplam İşlenen Paket Sayısı:** 22
- **Saniyede İşlenen Paket (PPS):** 4.40
- **Ortalama CPU Kullanımı:** %0.10
- **Ortalama Bellek Kullanımı:** 79.93 MB
- **Not:** Bu sonuçlar kısa süreli bir testten elde edilmiştir ve ağ trafiği yoğunluğuna göre değişebilir.

8. Sonuç ve Gelecek Çalışmalar

Proje, temel IDS/IPS işlevlerini başarıyla yerine getirmiştir. Belirtilen hedeflere ulaşılmış ve sistemin Linux, macOS ve Windows üzerinde çalışabilirliği doğrulanmıştır. Gelecekte yapılabilecek iyileştirmeler şunlardır:

- Stres testi için daha yoğun trafik simülasyonları eklenmeli.
- Kullanıcı arayüzü geliştirilerek sistem daha kullanıcı dostu hale getirilmeli.
- Threat Intelligence API entegrasyonu tamamlanmalı (AbuseIPDB).
- Loglama sistemi iyileştirilmeli ve olay korelasyonu eklenmeli.
- Farklı saldırı tipleri için test senaryoları oluşturulmalı.

9. Kaynaklar

- Scapy: https://scapy.net/
- Psutil: https://github.com/giampaolo/psutil
- AbuseIPDB API: https://www.abuseipdb.com/ (henüz tam entegrasyon sağlanmadı)