Ağ Güvenliği Proje Ödevi

Öğrenci Bilgileri: - **413518** - Berk Çağrı Laçin - **402498** - Enes Ceviz

---

Raw Socket Port Scanner - Teknik Rapor

Proje Özeti

Bu proje, uzak sunucularda port taraması yaparak açık servisleri tespit eden, işletim sistemi tahmininde bulunan ve network güvenlik analizi için kullanılan gelişmiş bir port scanner uygulamasıdır.

Temel Amaçlar:

- **Port Taraması:** Hedef sistemdeki açık TCP/UDP portlarını tespit etme
- **İşletim Sistemi Tespiti:** TTL ve window size analizi ile OS fingerprinting
- **Servis Tespiti:** Açık portlardaki çalışan servisleri belirleme
- **Banner Grabbing:** Servis versiyonları ve detaylarını toplama

Teknik Özellikler

Raw Socket İmplementasyonu

Program, gerçek raw socket teknolojisi kullanarak düşük seviyeli network paketleri oluşturur ve gönderir:

- **SOCK_RAW, IPPROTO_TCP:** Kernel bypass ile direkt TCP paket kontrolü
- **Manuel IP Header Oluşturma:** IP version, TTL, checksum manuel hesaplama
- **Manuel TCP Header Oluşturma:** TCP flags, sequence number, window size kontrolü
- **Custom Checksum Algoritması:** IP ve TCP checksum hesaplama
- **Packet Crafting:** SYN paketlerinin manuel oluşturulması

Tarama Teknikleri

1. **TCP SYN Scan (Stealth Scan)**
   - Raw socket ile SYN paketi gönderme
   - SYN+ACK yanıtı bekleme
   - Half-open connection tekniği
2. **TCP Connect Scan (Fallback)**
   - Standart socket bağlantısı
   - Raw socket başarısız olduğunda devreye girer

3. **UDP Scan**
   - Seçili portlar için (DNS, SNMP, etc.)
   - UDP paket gönderme ve yanıt analizi

Multi-Threading Yapısı

- **50 Eşzamanlı Thread:** Paralel port taraması
- **Mutex Koruması:** Thread-safe sonuç toplama
- **Batch Processing:** Port gruplarını paralel işleme

İşletim Sistemi Tespiti

- **TTL Analizi:** 64 (Linux/Unix), 128 (Windows), 255 (Cisco)
- **Window Size Analizi:** TCP window boyutu ile OS tespiti
- **Hop Count Hesaplama:** Network mesafesi analizi
- **Gelişmiş Fingerprinting:** Kombine analiz teknikleri

Kullanılan Teknolojiler

C++ Standart Kütüphaneleri

```
#include <sys/socket.h>     // Socket programlama
#include <netinet/ip.h>     // IP header yapıları
#include <netinet/tcp.h>    // TCP header yapıları
#include <arpa/inet.h>      // IP adres dönüşümleri
#include <thread>           // Multi-threading
#include <mutex>            // Thread synchronization
#include <chrono>           // Zaman ölçümü
```

Network Protokolleri

- **IPv4 Protocol Stack:** IP, TCP, UDP protokol implementasyonu
- **Berkeley Sockets API:** POSIX socket programlama
- **Raw Socket Programming:** Kernel-level packet manipulation

Sistem Çağrıları

- **socket():** Raw socket oluşturma
- **setsockopt():** Socket seçenekleri (IP_HDRINCL)
- **sendto():** Raw packet gönderme
- **recvfrom():** Yanıt paketi alma
- **select():** Non-blocking I/O

Güvenlik ve Yetkiler

Root Yetkisi Gereksinimleri

Program, raw socket kullanımı için root (sudo) yetkisi gerektirir:

```
sudo ./raw_scanner 192.168.1.1 1-1024
```

Neden Root Gerekli: - Raw socket oluşturma kernel seviyesi erişim gerektirir - IP header manipülasyonu privileged operation - Network interface'e direkt erişim

Yasal Durum

Bu program eğitim ve güvenlik analizi amaçlıdır. Yasal sorun teşkil etmez, ancak: - Sadece kendi sistemlerinizde veya izin verilen sistemlerde kullanın - Penetration testing için uygun yetkilendirme alın - Etik hacking prensiplerini takip edin

Kullanım Kılavuzu

Derleme

```
g++ -std=c++11 -Wall -Wextra -O2 -pthread -o raw_scanner main.cpp
```

Temel Kullanım

```
# Tek port tarama
sudo ./raw_scanner 192.168.1.1 80

# Port aralığı tarama
sudo ./raw_scanner 192.168.1.1 1-1024

# Çoklu port tarama
sudo ./raw_scanner 192.168.1.1 22,80,443,3389

# Karma tarama
sudo ./raw_scanner 192.168.1.1 1-100,443,8080-8090
```

Hostname Desteği

```
sudo ./raw_scanner google.com 80,443
sudo ./raw_scanner localhost 1-65535
```

Çıktı Formatı

Program iki aşamalı çıktı verir:

1. **Gerçek Zamanlı Bildirimler:**

```
Port 22 açık: SSH [TCP-SYN]
Port 80 açık: HTTP (Apache/2.4.41) [TCP-Connect]
Port 443 açık: HTTPS [TCP-SYN]
```

2. **Final Rapor Tablosu:**

```
                    PORT TARAMA SONUÇLARI

Hedef IP: 192.168.1.1
Tespit edilen OS: Linux/Unix - Yakın (2 hop) (5840 Window - Linux) (TTL: 62)
Açık port sayısı: 3

PORT    DURUM   PROTOKOL      SERVİS

22      AÇIK    TCP-SYN       SSH
80      AÇIK    TCP-Connect   HTTP (Apache/2.4.41)
443     AÇIK    TCP-SYN       HTTPS
```

macOS Sınırlamaları

Raw Socket Kısıtlamaları

macOS, güvenlik nedeniyle raw socket kullanımını kısıtlar:

1. **Yanıt Filtreleme:**
   - Kernel seviyesinde raw socket yanıtları filtrelenir
   - SYN+ACK paketleri user space'e ulaşmayabilir
   - Program bu durumu handle eder ve fallback kullanır
2. **Sistem İntegrasyonu Koruması (SIP):**
   - System Integrity Protection raw socket erişimini sınırlar
   - Root yetkisi olsa bile bazı kısıtlamalar devam eder
3. **Firewall Etkileşimi:**
   - macOS firewall raw paketleri etkileyebilir
   - pfctl kuralları raw socket trafiğini bloke edebilir

Çözüm Stratejileri

Program bu sınırlamaları aşmak için: - **Hybrid Approach:** Raw socket + TCP Connect fallback - **Timeout Handling:** macOS'ta yanıt alamama durumu için özel timeout - **Graceful Degradation:** Raw socket başarısız olursa standart socket kullanımı

Performans Etkileri

- Raw socket başarısız olduğunda TCP Connect scan daha yavaş
- macOS'ta tarama süresi Linux'a göre %20-30 daha uzun olabilir
- Thread sayısı macOS'ta daha konservatif tutulmalı

Performans Özellikleri

Optimizasyonlar

- **50 Paralel Thread:** Maksimum eşzamanlı bağlantı
- **2 Saniye Timeout:** Hızlı yanıt için optimize edilmiş
- **Batch Processing:** Port gruplarını verimli işleme
- **Memory Efficient:** Düşük bellek kullanımı

Benchmark Sonuçları

- **1-1024 Port Tarama:** ~30-45 saniye (macOS)
- **1-65535 Full Scan:** ~15-20 dakika (macOS)
- **Bellek Kullanımı:** <10MB RAM
- **CPU Kullanımı:** Orta seviye, multi-core optimized

Gelecek Geliştirmeler

Planlanan Özellikler

- IPv6 desteği
- Daha gelişmiş OS fingerprinting
- XML/JSON çıktı formatları
- Nmap script engine benzeri eklentiler
- GUI arayüz seçeneği

İyileştirme Alanları

- macOS raw socket sınırlamalarını aşma teknikleri
- Daha hızlı tarama algoritmaları
- Gelişmiş servis tespit yöntemleri
- Steganografi ve evasion teknikleri

Geliştirici Notu: Bu program eğitim amaçlı geliştirilmiştir. Network güvenliği öğrenmek ve sistem yöneticilerinin güvenlik açıklarını tespit etmesine yardımcı olmak için tasarlanmıştır.