# Monitorowanie Sieci – „non-blocking content packet filtering”

praca inżynierska

prowadzący: **prof. dr hab. inż. Zbigniew Kotulski**

# Plan prezentacji

- Założenia projektu
- Sniffer
- Technologie
  - WinPcap
  - Windows Socket API
  - NDIS
- Projekt

# Założenia projektu

- Cel projektu: monitorowanie sieci pod względem przesyłanych w niej treści (non-blocking content filtering)
- Aplikacja w technologii klient-serwer
- Klient:
  - ukryty w tle sniffer
  - analiza wszystkich pakietów (możliwośc filtrowania)
  - wysyłanie wyselekcjonowanych informacji do serwera
- Serwer:
  - zdalne zarządzanie klientem/klientami
    - dynamiczna zmiana reguł filtrowania
    - aktywacja/dezaktywacja klientów
  - zbieranie i archiwizacja wyników (logów)

# Sniffer

- Cel: przechwytywanie i analiza danych przepływających przez sieć (śledzenie komunikacji)
- Karta sieciowa w trybie mieszanym (*promiscuous*)
- Używany w routerach, serwera proxy i innych stronach komunikacji
- Użycie: diagnostyka sieci, monitorowanie aktywności sieciowej, debugowanie aplikacji sieciowych

# WinPcap – ogólnie (1/4)

- Implementacja biblioteki libpcap dla środowiska Windows
- Otwarty standard
- Przechwytywanie/wysyłanie pakietów w warstwie łącza
- Filtrowanie pakietów
- Niskopoziomowy dostęp do sieci (sterownik)

# WinPcap – cechy (2/4)

- Darmowy
  - BSD open source licence
  - Dostępny kod
  - Możliwość użycia w komercyjnych aplikacjach
- Wysoka wydajność
  - Filtrowanie i buforowanie na poziomie jądra
- Popularny
  - Używany do: monitorowania sieci, snifferów, wykrywania intruzów, generatorów ruchu, testowania sieci …
  - Wykorzystywany przez wiele aplikacji: Wireshark, Snort, Nmap, Windump
- Przetestowany i wiarygodny
  - Używany od wielu lat
  - Ciągle rozwijany
- Łatwy w użyciu dla użytkownika
  - Mała „paczka”, która pozwala na instalacje

# WinPcap – cechy (3/4)

- Łatwy w użyciu dla programisty
  - Każda wersja zawiera „developer's pack": dokumentacja, biblioteka, dodatkowe pliki, przykłady gotowe do kompilacji
- Wielo-platformowy
  - Windows: NT, XP, 2000, Vista, 2003 Server
  - Starsze Windowsy: 95, 98, ME – brak wsparci i nie rozwoju
- Przenośny
  - Kompatybilny z libpcap
  - Można wykorzystywać z narzędziami dla Unix/Linux
- Dobra dokumentacja
  - Instrukcja krok po kroku jak wykorzystać WinPcap
- Dobra wsparcie
  - CACE Technology – mail, forum, telefon

# WinPcap\libpcap – wrappers (4/4)

- Standartowo biblioteka dla C/C++
- Implementacje dla innych środowisk:
  - jpcap – Java
  - Net::pcap – Perl
  - WinPcapNEt - .Net
  - Ruby/Pcap – Ruby
  - python-libpcap - Python

# WinSock – ogólnie (1/5)

- Opracowany w Berkeley interfejs gniazd zaadaptowany do Windows
- Obejmuje zestaw funkcji ogólnego zastosowania
- Umożliwia korzystnie z wielu protokołów komunikacyjnych
- Interfejs, z którego korzystają aplikacje nie zależny od protokołu, którym się porozumiewają
- Protokoły dostarczane na zasadzie usług (ujednolicony interfejs)
- W wywołaniach funkcji tylko typ usługi (bez dokładnej nazwy protokołu)
- Mechanizmy warstw poniżej warstwy aplikacji maskowane przed aplikacja
- Korzystanie z gniazd na zasadzie strumienia danych

# WinSock – historia (2/5)

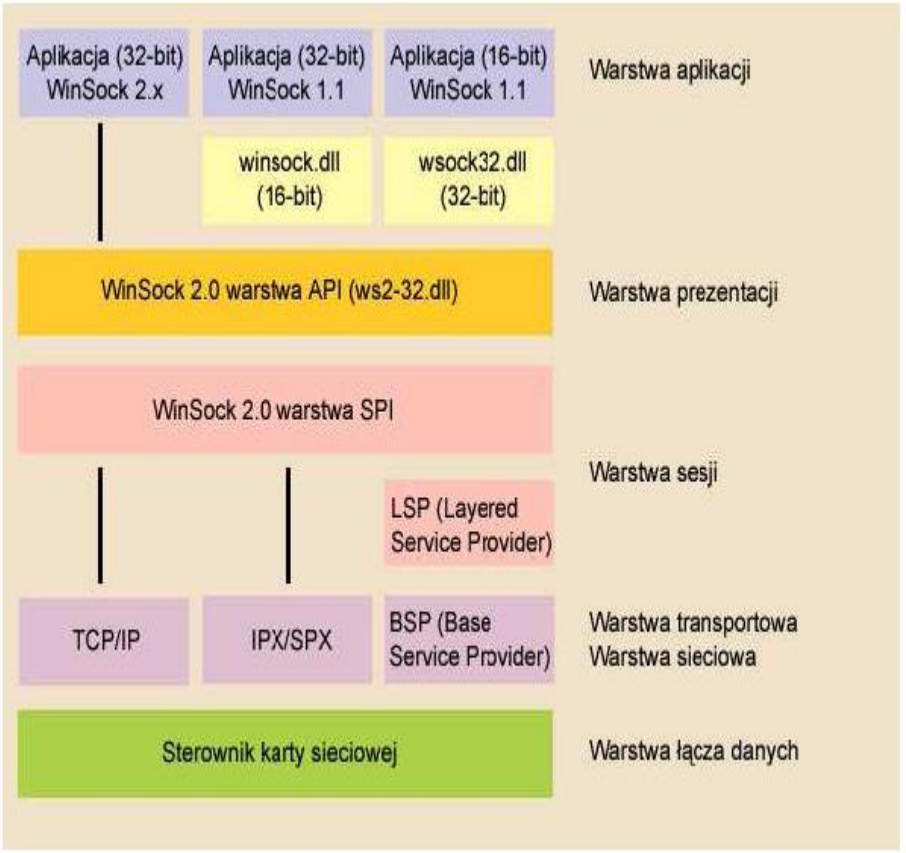
- WinSock wersja 1.1 (1993)
  - Standard aplikacji dla Windows
  - Obsługa jedynie protokołów stosu TCP/IP
- WinSock wersja 2.0
  - Możliwość korzystanie nie tylko z protokołów TCP/IP – SPI (Service Provider Interface)
  - QoS
  - Współdzielenie „socketów”
  - Interfejs zbliżony do ideologii interfejsu gniazd BSD Unix

# WinSock – LSP (3/5)

- LSP, czyli Layered Service Provider
- 2 typy usług:
  - Transportowe (np. TCP/IP)
  - Przestrzeni nazw (np. DNS)
- Usługi transportowe:
  - Podstawowe (BSP) – warstwa transportowa i sieciowa
    - Ustanawianie połączenia
    - Transfer danych
    - Obsługa błędów
  - Rozszerzające (LSP) – warstwa sesji
    - Wybrane funkcje komunikacyjne oparte na usługach podstawowych lub innych rozszerzonych

# WinSock – stos dawców usług (4/5)

- Pozwala na używanie w systemie 2 lub więcej usług o tych samych parametrach
- Wyszukiwanie przez warstwę SBP biblioteki WinSock
- Żądanie udostępnienia gniazda usługi zwraca do aplikacji deskryptor gniazda dla usługi najbliżej wierzchołka stosu dawców usług



# WinSock – podsumowanie (5/5)

- Wykorzystanie tych samych funkcji obsługujących gniazda niezależnie od wykorzystywanego protokołu transportowego
- Swoboda dodawania, oraz rozszerzania usług, "przezroczystość" (ang. transparency) usług rozszerzających podstawowe
- Funkcje protokołów transportowych
- Obsługa zdarzeń sieciowych
- Wady: zwiększenie wykorzystanie mocy obliczeniowej procesora i pamięci operacyjnej w systemie

# NDIS

- Network Driver Interface Specification
- Interfejs programowania aplikacji dla kart sieciowych
- Interfejs pomiędzy drugą (łącza danych) a trzecią (sieci) warstwą (dokładnie w LLC – podwarstwa warstwy łącza)
- „opakowanie” złożoności karty sieciowej
- Na stronie MSDN Microsoft jest tutorial „Writting NDIS Filter Drivers”

# NDIS Intermediate (IM) Driver

- Nowy rodzaj sterownika NDIS (od Windows NT 4.0 SP3)
- Przeźroczysta warstwa pomiędzy częścią transportowa sterownika NDIS i interfejsem karty sieciowej (NIC)
- Wykorzystanie:
  - Monitorowanie pakietów
  - Szyfrowanie
  - Filtrowanie pakietów
    - Odrzucanie/przepuszczanie pakietów
    - Zapis/opóźnianie pakietów
    - Kompresja/Dekompresja pakietów
    - Kierowanie pakietów

# Projekt (1/3)

- Wybrane technologie:
  - Windows
  - Java + jpcap
  - Filtry jako XML
  - Archiwizacja w plikach
- Serwer – proste GUI, komendy „konsolowe”
- Klient – aplikacja ukryta

# Projekt – klient (2/3)

- Filtrowanie pakietów według:
  - Protokołów
  - Adresów IP
  - Portów
- Rozbudowany filtr dla HTTP

# Projekt – serwer (3/3)

- Usługa na otwartym porcie
- Informacje o wszystkich uruchomionych aplikacjach klienckich
- Zarządzanie klientami
- Wysyłanie reguł filtrowania
- Archiwizacja danych do pliku
- Blokowanie wybranych interfejsów u klienta
- Brak możliwości blokowania pakietów

# Koniec

Pytania?