SAMODZIELNY ZAKŁAD SIECI KOMPUTEROWYCH
POLITECHNIKA ŁÓDZKA
90-924 Łódź ul. Stefanowskiego 18/22
tel./fax. (42) 636 03 00
e-mail: szsk@zsku.p.lodz.pl

Marcin Giełdziński

Zapory ogniowe typu NBAR

praca dyplomowa magisterska

Promotor:
dr inż. Michał Morawski

Dyplomant:
Marcin Giełdziński
nr albumu: 104825

Łódź, wrzesień 2005 r.

SPIS TREŚCI

Wykaz skrótów 4
1. Wstęp 6
2. Cel i zakres pracy 8
3. Bezpieczeństwo sieci komputerowych 9
3.1. Bezpieczeństwo systemu informacyjnego 10
3.2 Bezpieczeństwo przepływu danych przez sieć komputerową 11
3.3. Bezpieczeństwo przepływu danych - przepustowość 12
3.4. Jakość usług (*Quality of Service*) 16
3.4.1. Struktura *Quality of Service* 17
4. Rozpoznawanie aplikacji 19
4.1. Protokół IP 19
4.2. Numery portu TCP i UDP 23
4.3. Numer portu TCP, UDP i adres hosta 26
4.4. Wielkość pakietów 27
4.5. Inspekcja pakietów 28
5. Mechanizmy rozpoznające aplikacje 31
5.1. Rozwiązania sprzętowo-programowe 31
5.1.1. Cisco NBAR 31
5.1.2. Wise – TrafView 33
5.2. Agenci systemów operacyjnych 36
5.2.1. Ethereal 36
5.3. Cechy charakterystyczne protokołów 37
5.3.1. Protokół HTTP 38
5.3.2. Protokół SSH i SCP 40
5.3.3. Protokół eDonkey 41
5.3.4. SIP 42
6. Opis części praktycznej 45
6.1. Wstęp 45
6.2. Zakres 46
6.3. Opis programu 46
6.3.1. Wykorzystane narzędzia 46
6.3.2. Moduł główny 47

6.3.3. Ogólna zasada działania programu „NBAR monitor” ... 48
6.4. Mechanizm wtyczek ... 49
6.4.1. Wstęp ... 49
6.4.2. Konwencja pisania wtyczek ... 50
6.4.3. Dostarczone wtyczki ... 51
6.5. Cechy charakterystyczne protokołów ... 53
6.5.1. Protokół DHCP ... 53
6.5.2. Protokół DNS ... 53
6.5.3. Protokół FTP ... 54
6.5.4. Protokół HTTP i HTTPS ... 56
6.5.5. Protokół SSH i SCP ... 56
6.5.6. Protokół IMAP ... 56
6.5.7. Protokół POP3 ... 57
6.5.8. Protokół SMTP ... 57
6.5.9. Protokół sieci BitTorrent ... 57
6.5.10. Protokół sieci eDonkey ... 58
6.5.11. Protokół RTP ... 58
6.5.12. Protokół SIP ... 59
6.5.13. Protokół RDP ... 59
6.6. Testy i uwagi ... 60
7. Podsumowanie ... 62
8. Literatura ... 64
9. Załącznik 1 Instrukcja użytkownika programu „NBAR monitor” ... 68
9.1. Przeznaczenie programu ... 68
9.2. Wymagania ... 68
9.3. Instalacja ... 68
9.4. Opis interfejsu ... 69

Wykaz skrótów

Wszystkie skróty są w języku angielskim.

- Cisco IOS – *Cisco Internetwork Operating System*
- DARPA – *Defense Advanced Research Projects Agency*
- DHCP – *Dynamic Host Configuration Protocol*
- DLL – *Dynamic Link Library*
- DNS – *Domain Name System*
- FTP – *File Transfer Protocol*
- HTTP – *Hypertext Transfer Protocol*
- IANA – *Internet Assigned Numbers Authority*
- ICANN – *Internet Corporation For Assigned Names and Numbers*
- ICMP – *Internet Control Message Protocol*
- IMAP – *Interactive Mail Access Protocol*
- IP – *Internet Protocol*
- ISO – *International Organization for Standardization*
- LAN – *Local Area Network*
- MIME – *Multipurpose Internet Mail Extensions*
- NBAR – *Network-Based Application Recognition*
- OSI – *Open Systems Interconnect*
- P2P – *Peer-to-Peer*
- POP3 – *Post Office Protocol version 3*
- QoS – *Quality of Service*
- RDP – *Remote Desktop Protocol*
- RFC – *Request for Comments*
- RTP – *Real-Time Transport Protocol*
- SCP – *Secure Copy Protocol*
- SDP – *Session Descriptions Protocol*
- SFTP – *Secure FTP*
- SIP – *Session Initiation Protocol*
- SMTP – *Simple Mail Transfer Protocol*

- SSH – *The Secure Shell*
- TCP – *Transmission Control Protocol*
- ToS – *Type of Service*
- UDP – *User Datagram Protocol*
- URI – *Uniform Resource Indicator*
- VoIP – *Voice over IP*
- WAN – *Wide Area Network*
- WWW – *World Wide Web*

1. Wstęp

Ostatnie dwa dziesięciolecia były okresem burzliwego rozwoju telekomunikacji cyfrowej i różnorodnych systemów komputerowych. Przyczynił się do tego zarówno rozwój technik informatycznych jak i powszechniejsze stosowanie nowoczesnych, szerokopasmowych mediów transmisyjnych. Stworzono efektywne języki programowania i systemy operacyjne, opracowano nowe typy mikroprocesorów, pamięci operacyjnych i pamięci masowych a także wprowadzono do powszechnego użytku łącza światłowodowe i kanały satelitarne. Wszystkie te czynniki miały swój istotny wpływ na dynamiczny rozwój sieci komputerowych, zarówno rozległych, obejmujących swoim zasięgiem poszczególne kraje i całe kontynenty, jak i lokalnych, zaspokajających potrzeby jednej instytucji [3].

Sieci komputerowe stały się integralną częścią infrastruktury współczesnej organizacji. Właściwie trudno sobie dziś wyobrazić instytucję nieposiadającą komputerów połączonych w sieć lokalną oraz niekorzystającą z zasobów i usług Internetu. Technologie informacyjne i telekomunikacyjne, w tym sieci komputerowe, są obecne w życiu prywatnym i zawodowym milionów ludzi na całym świecie. Kierowanie kadrami, finansami czy inwestycjami wymaga od menedżera użycia wyspecjalizowanych programów komputerowych operujących na dużych bazach danych, często rozproszonych. Szybko rozwijające się technologie informacyjne i telekomunikacyjne zmieniają warsztaty pracy prawie we wszystkich dziedzinach. W konsekwencji, efektywność pracownika zależy w dużym stopniu od umiejętności posługiwania się środkami i narzędziami technologii informacyjnej i telekomunikacyjnej. Szczególną rolę odgrywają zasoby i usługi sieci Internet, która dla każdej organizacji, jest źródłem informacji (w tym o najnowszych technologiach) oraz środkiem wymiany informacji. Internet to współczesna platforma dla działalności handlowej i finansowej a także arena kooperacji między firmami [4].

W związku z tym, w dzisiejszych czasach, bardzo ważnym aspektem działalności firmy stało się zapewnienie bezpieczeństwa łączności z Internetem. Bezpieczeństwo to rozumiane jest jako dostarczenie łącza internetowego, o przepustowości pozwalającej na bezkonfliktowe działanie aplikacji wykorzystujących sieć Internet oraz na niedopuszczaniu transmisji, która mogła by być szkodliwa dla działalności firmy.

Stale rozwijające się systemy informatyczne, transmitujące coraz więcej danych oraz wykorzystanie łącz internetowych nie zawsze zgodne z profilem działalności firmy, powodują, że coraz częściej na infostradzie zdarzają się zatory oraz korki. Jeżeli są to sytuacje trwające ułamki sekund wówczas można określić je mianem normalnych, ale kiedy trwają dłużej, uniemożliwiając skorzystanie z systemów informatycznych, wówczas zagrożone jest bezpieczeństwo firmy, która narażona jest na duże straty. Bardzo często problem ten związany jest ze złym wykorzystaniem dostępnej przepustowości łącza internetowego, a nie z jego rozmiarem.

Panaceum na tego rodzaju problemy okazują się wszelkiego rodzaju mechanizmy optymalizujące wykorzystanie dostępnej przepustowości łącz internetowych oraz dynamiczne zapory ogniowe, które mogą na „bieżąco" reagować na procesy zachodzące w sieci i blokować szkodliwe dla działalności firmy transmisje.

U podstaw działania tych mechanizmów leży właściwe rozpoznanie aplikacji, które generują ruch w sieci. To dzięki takiej identyfikacji możliwa jest późniejsza analiza, podział na klasy ruchu oraz stosowanie odpowiednich algorytmów zarządzających klasami, w celu optymalizacji wykorzystania łącza internetowego. Dzięki niej dynamiczne zapory ogniowe mają aktualne informacje o typach aplikacji przesyłających dane przez sieć, przez co mogą bardzo szybko, w sposób „inteligentny", reagować na pojawienie się niepożądanych transmisji.

Funkcjonalność polegającą na rozpoznawaniu aplikacji, które generują ruch w sieci, określić można pojęciem używanym przez firmę CISCO *Network Base Application Recognition* - NBAR. Ten rodzaj rozpoznawania aplikacji bazuje na wiedzy o wykorzystywanych przez nie protokołach i analizie transmitowanych danych, co daje pełny obraz procesów zachodzących w sieci komputerowej. Taki rodzaj rozpoznawania aplikacji umożliwia administratorom lepsze i właściwsze wykorzystywanie łącz internetowych przez co bezpieczeństwo działalności firmy nie jest zagrożone.

2. Cel i zakres pracy

Celem pracy jest przegląd metod analizy ruchu w sieciach IP, umożliwiających rozpoznanie rodzaju aplikacji generującej ten ruch, a także praktyczna realizacja takiego mechanizmu identyfikującego, który może dostarczać informacje pozwalające ustalić dynamiczne reguły sterowania zaporą ogniową jak i agenta sterującego klasami jakości usług (ang. *Quality of Service – QoS*).

W niniejszym opracowaniu przedstawiono problem bezpieczeństwa sieci komputerowych, a zwłaszcza jednego z jego czynników, bezpieczeństwa transmisji danych oraz rolę jako odgrywają w jej zapewnieniu mechanizmy rozpoznające typ aplikacji.

Praca zawiera opis najbardziej znanych metod analiz ruchu w sieciach IP, których celem jest ustalenie rodzaju aplikacji generującej ruch. Przedstawia firmowe rozwiązania realizujące tę funkcjonalność oraz opisuje cechy charakterystyczne dla przykładowych protokołów, dzięki którym możliwa jest ich identyfikacja.

Ostatnia części pracy przedstawia stworzony program, realizujący funkcję identyfikującą protokoły na podstawie ruchu w sieci.

Tematyka pracy koncentruje się wokół protokołu IP, z racji jego popularności i powszechności w spotykanych na co dzień sieciach.

3. Bezpieczeństwo sieci komputerowych

W ostatnim czasie sieci komputerowe stały się integralną częścią infrastruktury współczesnej organizacji, w której ważną rolę odgrywają zasoby i usługi sieci Internet. Wyjątkowej wagi nabiera wówczas zagadnienie bezpieczeństwa sieci. Wszystkie operacje, usługi oraz serwisy powinny funkcjonować właściwie oraz być odporne na wszelkiego rodzaju efekty niepożądane [4].

Ma to ogromne znaczenie wówczas, gdy firmy zajmują się taką działalnością jak m.in. handel elektroniczny, czy bankowość elektroniczna. Bezpieczeństwo odgrywa ważną rolę w sytuacji, wykorzystania sieci komputerowych do przesyłania danych, zaczynając od poczty elektronicznej, informacji niezbędnych do działania firmy a kończąc na video konferencjach czy też telefonii opartej o protokół IP (VoIP).

Powinno się pamiętać, że bezpieczeństwo to między innymi zapewnienie poprawności działania sieci komputerowej, czyli dostarczenie takich parametrów transmisji, aby wszystkie usługi działały poprawnie oraz kontrolowanie przesyłanych danych w celu wykrywania niedozwolonych transmisji.

Problem parametrów łączności wiąże się głównie z dostarczeniem odpowiedniej przepustowości łącza internetowego (które można wykupić od dostawcy Internetu) lub w bardziej ekonomicznym podejściu, z zastosowaniem mechanizmów optymalizujących działanie sieci komputerowych. Problem wykrywania niedozwolonych transmisji wiąże się z zastosowaniem „inteligentnego" mechanizmu analizująco-reagującego na procesy zachodzące wewnątrz sieci, czego przykładem są dynamiczne zapory ogniowe.

U podstaw funkcjonowania takich mechanizmów znajduje się identyfikacja ruchu sieciowego, czyli rozpoznawanie typu aplikacji/protokołu, które generują ruch w sieci. Identyfikacja oparta jest na analizie, wyżej opisanego, ruchu z wykorzystaniem informacji zawartych w warstwie drugiej i trzeciej modelu OSI, jak również, w przypadku trudniejszych do analizy protokołów, informacji zawartych w warstwach wyższych aż do warstwy aplikacji. Takie rozpoznawanie określić można pojęciem wykorzystywanym przez firmę CISCO, mianowicie NBAR – *Network Base Application Recognition*.

3.1. Bezpieczeństwo systemu informacyjnego

Bezpieczeństwo systemu informacyjnego nie jest pojęciem łatwo definiowalnym. Polska Norma PN-I-13335 określa je jako "wszystkie aspekty związane z definiowaniem, osiąganiem i utrzymaniem poufności, integralności, dostępności, rozliczalności, autentyczności i niezawodności" [2]. Warunkiem uznania systemu informacyjnego za bezpieczny jest więc spełnienie przez niego następujących kryteriów:

- **poufności** (ang. *confidentiality*) - czyli ochrony informacji przed nieautoryzowanym dostępem;
- **integralności** (ang. *integrity*) - czyli ochrony przed nieautoryzowaną modyfikacją;
- **dostępności** (ang. *availability*) - czyli gwarancji uprawnionego dostępu do informacji;
- **rozliczalności** (ang. *accountability*) - czyli możliwości weryfikowania odpowiedzialności za korzystanie z systemu przez użytkowników;
- **autentyczności** (ang. *authenticity*) - czyli możliwości weryfikacji tożsamości użytkowników systemu i informacji w nim zawartych;
- **niezawodności** (ang. *reliability*) - czyli gwarancji spójnego i zamierzonego funkcjonowania systemu.

Rozważając różne aspekty bezpieczeństwa systemu informatycznego można wyróżnić:

- bezpieczeństwo fizyczne (ang. *physical security*)
  Określa sposób, w jaki obiekty systemu informacyjnego oraz jego użytkownicy są chronieni przed zagrożeniem fizycznym (np. przed kradzieżą dysków twardych) i bezpośrednią obserwacją (np. podsłuchem). W zakres bezpieczeństwa fizycznego wchodzą także procedury sprawdzające tożsamość personelu korzystającego z systemu [5].
- bezpieczeństwo komunikacyjne (ang. *communication security*)
  Polega na zapewnieniu poprawności przesyłania danych, uniemożliwienie ich podsłuchania oraz zmianie czy też zniszczeniu podczas transmisji.

- bezpieczeństwo logiczne (ang. *logical security*)
  Jest to ochrona informacji przed nieuprawnioną zmianą oraz jej pozyskaniem czy zniszczeniem [5].

Dalsza część pracy poświęcona jest aspektowi bezpieczeństwa komunikacyjnego systemu informacyjnego, a dokładniej jednego z jego elementów, bezpieczeństwa sieci komputerowej, w sensie poprawności transmisji danych.

3.2 Bezpieczeństwo przepływu danych przez sieć komputerową

Dane przesyłane przez sieć komputerową narażone są na różnego rodzaju zagrożenia, takie jak: przerwanie przepływu informacji (atak na dyspozycyjność), przechwycenie informacji przez nieuprawnioną osobę (atak na poufność), modyfikacja informacji bez wiedzy adresata (atak na nienaruszalność), podrobienie informacji, czyli podszycie się pod nadawcę i wysłanie fałszywych danych (atak na autentyczność).

Bezpieczeństwo przepływu danych przez sieć komputerową należy również rozumieć jako zapewnienie poprawności transmisji danych, polegające na dostarczeniu takich parametrów transmisji, aby wszystkie aplikacje, bez przeszkód, mogły komunikować się ze sobą.

W każdej firmie niedopuszczalna jest sytuacja, kiedy nie można odebrać poczty elektronicznej, czy otworzyć w przeglądarce internetowej strony WWW, w chwili kiedy jedna osoba lub grupa pracowników posługuje się oprogramowaniem (związanym i niezwiązanym z działalnością firmy), wykorzystującym prawie całe dostępne pasmo łącza internetowego.

W przypadku, kiedy firma wykorzystuje Internet do komunikacji z kontrahentem, korzystając z video-konferencji, czy telefonii VoIP, niedopuszczalna jest sytuacja, w której parametry transmisji są na tyle złe, że niemożliwie jest zrozumienie rozmówcy, czy też nawiązanie połączenia. Niedopuszczalna jest również sytuacja, w której połączenie zostanie przerwane, np. dlatego, że któryś z pracowników pobiera film z serwera ftp, czy z sieci Peer-to-Peer. Wówczas obok danych i zasobów, które należy chronić, zagrożony jest jeden z podmiotów bezpieczeństwa a mianowicie reputacja.

W konsekwencji utraty wiarygodności, uniemożliwieniu dostępu do danych, czy to też ograniczenia do nich dostępu poprzez niewłaściwe działającą sieć

komputerową, firma narażona jest na poważne straty finansowe. Przykładem ilustrującym ten problem jest poniższa tabela.

Tabela 1. Koszt utraty / braku dostępu do danych poszczególnych sektorów gospodarki USA [6]

Sektor gospodarki (USA)	Strata dochodu całego sektora z tytułu 1h przestoju
Energetyka	**2,8 mln $**
Telekomunikacja	**2,0 mln $**
Produkcja	**1,6 mln $**
Instytucje finansowe	**1,4 mln $**
IT	**1,3 mln $**
Ubezpieczenia	**1,2 mln $**
Handel detaliczny	**1,1 mln $**
Farmacja	**1,0 mln $**
Bankowość	**999 tys. $**

Biorąc pod uwagę powyższe dane, niebagatelne staje się zapewnienie poprawności transmisji danych jako jednego z czynników bezpieczeństwa przepływu informacji przez sieć komputerową.

3.3. Bezpieczeństwo przepływu danych - przepustowość

W celu zapewnienia bezpieczeństwa sieci komputerowej, w sensie poprawności transmisji danych, należy zadbać o dostarczenie odpowiednich parametrów transmisji, aby wszystkie aplikacje wykorzystujące sieć lokalną i Internet mogły sprawnie działać.

Wyróżniamy następujące parametry:

- **przepustowość** (ang. *bandwidth*)
  Ilość informacji, jaka może być skutecznie przesyłana między dwoma punktami końcowymi przez sieć w danej jednostce czasu, przepustowość mierzona jest w bitach na sekundę (bit/s).
- **opóźnienie (ang. *delay)***
  „Odstęp czasu pomiędzy chwilą wysłania pakietu z jednego punktu w sieci i odebrania go w innym punkcie w sieci" [7].

- **jednorodność opóźnienia** (ang. *jitter*)

  „Zakres, w którym zmienia się wartość opóźnienia mierzona dla pakietów należących do tego samego strumienia”[7]. Na ten parametr szczególnie wrażliwe są aplikacji czasu rzeczywistego. Jednorodność opóźnienia związana jest z własnością sieci IP, polegającą na dzieleniu informacji na pewną ilość pakietów, z których każdy może być przesłany przez sieć inną trasą pomiędzy nadajnikiem i odbiornikiem, co powoduje, że każdy pakiet może posiadać odmienne opóźnienie. Nie ma gwarancji, że dotrą one do odbiornika w kolejności, w jakiej zostały wysłane z nadajnika. [19]
- **straty pakietów**

  Ilość pakietów wysłanych przez nadajnik, a niedostarczonych do odbiornika, z reguły wyrażana w procentach.

Spośród tych parametrów, najbardziej znacząca i najbardziej wpływająca na poprawność działania sieci komputerowej, jest przepustowość, zarówno łącz sieci lokalnej, jak i przyłącza dostarczonego przez dostawce Internetu (ISP – *Internet Service Provider*). Przepustowość łącz sieci lokalnej jest, w większości przypadków, bardzo duża, kształtująca się w typowych wartościach 10Mb/s, 100Mb/s lub 1Gb/s, co jest wystarczające do zapewnienia poprawnej działalności systemu informatycznego w obrębie Intranetu. Przepustowość łącza Internetowego z reguły jest niższa i silnie zależy od potrzeb odbiorcy, kształtując się w granicach od 256 kb/s do 10 Gb/s (w skrajnych przypadkach nawet więcej). Odpowiednio dobrana przepustowość, w praktyce, godzi dwa kryteria: cenę przyłącza i wymagania systemów informatycznych, dlatego rzadko kiedy stosuje się rozwiązania polegającego na dostarczeniu każdemu użytkownikowi sieci dedykowanego łącza internetowego, ale wykorzystuje się jedno (bądź kilka) łącz. Łączna przepustowość współdzielona jest przez wszystkich użytkowników i ze względu na cenę łącza nie będzie ono na tyle pojemne, aby mogło zaspokoić maksymalne żądania aplikacji użytkowników.

Przykładem ilustrującym ten problem może być sieć komputerowa w średniej firmie, której przepustowość wynosi 100Mb/s, a łącze z Internetem 1Mb/s. W takiej sytuacji aplikacja, która w obrębie Intranetu firmy może wykorzystywać przepustowość dochodzącą nawet do 100Mb/s, komunikując się poprzez Internet ma do dyspozycji maksymalnie łącze o szybkości 1 Mb/s. Zatem aplikacja chcąca połączyć się poza obszar Intranetu musi ograniczyć swoje żądania, co do przepustowości,

do wartości 1 Mb/s. Jeżeli inna aplikacja będzie chciała również wykorzystać łącze z Internetem, wówczas ma do dyspozycji przepustowość 1Mb/s, odpowiednio pomniejszoną o wartość wykorzystywaną przez pierwszą aplikację. Może to być przyczyną sytuacji, w wyniku której chwilowa ilość aplikacji, wykorzystująca łącze z Internetem jest tak duża, że nowa aplikacja nie może się połączyć. Szybkość transmisji programów, które ją wykorzystują, spada do wartości powodujących pogorszenie działania, bądź nawet przerywanie połączenia. Taka sytuacja może trwać od kilku sekund do kilkudziesięciu minut.

Jeżeli takie problemy mają charakter przejściowy i krótkotrwały nie jest konieczne rozbudowywanie łącz z Internetem, ponieważ wiąże się to z dużymi kosztami. Należy wówczas zastosować różnego rodzaju mechanizmy pozwalające na wydajniejsze wykorzystanie dostępnej przepustowości przyłącza internetowego.

Ruch generowany w sieci komputerowej można podzielić na dwa rodzaje:

- związany z protokołami sterującymi działaniem sieci komputerowej;
- generowany przez aplikacje klienckie.

Ruch związany z protokołami sterującymi działaniem sieci komputerowej, zapewniający poprawność funkcjonowania sieci, zajmuje znikomą część dostępnej przepustowości łącza. Sieć jest najbardziej wykorzystywana przez aplikacje klienckie, które rozpoznawane są za pomocą stosowanego protokołu.

W przykładowej sieci firmy, generowany przez ten rodzaj aplikacji, ruch można podzielić na związany oraz niezwiązany z działalnością firmy.

Przykładami ruchu sieciowego związanego z funkcjonowaniem firmy może być ruch skojarzony z protokołami:

- HTTP [40], SECURE-HTTP [42], wykorzystywany przez przeglądarki internetowe;
- SMTP [38], POP3 [39], IMAP [44], wykorzystywany przez klientów poczty elektronicznej;
- RTP (*Real Time Protocol*) [45], wykorzystywany przez aplikacje czasu rzeczywistego np. do przesyłania głosu, czy obrazu video;
- VoIP służący do obsługi telefonii opartej o protokół IP;
- inne, niestandardowe stworzone na potrzeby konkretnych firm.

Do ruchu sieciowego, należącego do działalności firmy, można zaliczyć: FTP (*File Transport Protocol*) [34], który wykorzystywany jest przez aplikacje klientów FTP do przesyłania plików. Do tej grupy nie zaliczają się protokoły związane z obsługą aplikacji typu *Peer-to-Peer*.

W celu lepszego wykorzystania łącza internetowego można zastosować mechanizm, który na podstawie analizy ruchu w sieci komputerowej, rozpoznaje rodzaj aplikacji, a dostarczone przez niego informacje służą zarówno do ustalania dynamicznych reguł na zaporach ogniowych, jak i dokładnej konfiguracji mechanizmów zarządzania przepustowością łącza internetowego.

Mechanizmy zarządzania przepustowością łącza internetowego:

- priorytyzacja ruchu sieciowego;
- *Quality of Service* (informacje dla agenta sterującego klasami QoS);
- dynamiczna zapora ogniowa, która np. mając informacje o wykryciu niedozwolonego, mniej preferowanego w firmie ruchu, ogranicza go, bądź blokuje.

Mechanizm wykrywania i rozpoznawania aplikacji nazywany jest w zależności od firmy. Przez firmę CICO określana jest mianem *Network-Based Application Recognition* - NBAR.

Od stosowanych mechanizmów zarządzania rodzajami ruchu i ochrony sieci, nie mniej ważna jest jego identyfikacja. To ona jest podstawą dalszych działań, dzięki którym możemy rozpoznać, jaki typ ruchu przechodzi przez sieć komputerową. Dzięki identyfikacji można podzielić ruch sieciowy na dowolne grupy a posiadając wiedzę na temat protokołów/aplikacji, generujących ruch w sieci, można go bardzo dokładnie zmierzyć. W konsekwencji pozwala to na zastosowanie, bazujących na zarządzaniu klasami ruchu sieciowego, odpowiednich mechanizmów poprawiających działanie sieci komputerowej. Do wyżej wymienionych mechanizmów można również zaliczyć dynamiczne zapory ogniowe, które na podstawie uzyskanych informacji mogą, w sposób automatyczny, zablokować niedozwoloną transmisję. Takie działania nie dopuszcza do niewłaściwego wykorzystania łącza internetowego oraz chronią przed nieuprawnionym dostępem. Dzięki temu zapewniony jest odpowiedni poziom bezpieczeństwa przepływu danych, będący jednym z podstawowych elementów bezpieczeństwa systemu informatycznego.

Na zwrócenie szczególnej uwagi zasługuje, często pomijana lecz bardzo ważna, funkcja jaka odgrywają programy analizujące oraz rozpoznające aplikacje/protokoły, które implementują funkcjonalność definiowaną pojęciem NBAR.

Przedstawione powyżej rozważania dotyczą jednego z parametrów transmisji. Aby zapewnić wszystkie parametry należy zastosować inne mechanizmy, które spowodują, że na określonym poziomie, zagwarantowana będzie nie tylko przepustowość, ale również opóźnienie, jednorodność opóźnienia oraz straty pakietów. Zbiór wszystkich mechanizmów określany jest mianem jakości usług (ang. *Quality of Service*).

3.4. Jakość usług (*Quality of Service*)

Sieci teleinformatyczne rozwijają się na całym świecie w bardzo szybkim tempie. Coraz częściej firmy wdrażające technologie teletransmisyjne ustanawiają nowe rekordy efektywnego przesyłania danych. Struktury sieci (korporacyjnych, naukowych, i innych) stają się szybsze i wiele rozleglejsze niż parę lat temu. Pomimo rozpowszechnienia szybkich kanałów transmisyjnych koszty eksploatacyjne są nadal wysokie. Rzadko zdarza się, aby jakaś duża i szybka sieć była podłączona do Internetu łączem odpowiadającym jej realnej przepustowości. Zazwyczaj takie łącze ma kilkakrotnie niższą przepustowość niż sieć do niego podłączona. Coraz częściej jednak nowe usług tj. aplikacje czasu rzeczywistego, wymaga stałego a zarazem szerokiego pasma transmisyjnego. Z tego powodu dla pewnej grupy danych trzeba zagwarantować wysoką jakość usług sieciowych, aby wysyłane pakiety docierały bez opóźnień i strat części ramek [9].

W literaturze występuje wiele definicji jakości usług określających „*Quality of Service*" jako zbiór mechanizmów, które mają zapewnić dostarczenie przewidywalnego poziomu jakości usług sieciowych, poprzez zapewnienie określonych parametrów transmisji danych, w celu osiągnięcia satysfakcji użytkownika [19]. Cztery podstawowe i najczęściej rozważane parametry to: przepustowość, opóźnienie, jednorodność opóźnienia oraz straty pakietów (por. roz. 3.3).

3.4.1. Struktura *Quality of Service*

Wydajny mechanizm ***Quality of Service*** **(**QoS) powinien zawierać sposoby na [20]:

- zakomunikowanie dokonywanych rezerwacji wybranym węzłom sieci;
- faktyczne zapewnienie nienaruszalności zarezerwowanych zasobów w obrębie węzła sieci.

Praktyczna implementacja takiego mechanizmu powinna zostać dodatkowo wzbogacona o [20]:

- interfejs dla aplikacji chcących dokonać rezerwacji zasobów sieciowych;
- scentralizowany model zarządzania działającymi mechanizmami QoS.

O ile dwa ostatnie problemy są ściśle związane z platformą sprzętową, na której dokonywana jest implementacja QoS, o tyle w dziedzinie pierwszych dwóch problemów istnieje szeroka gama proponowanych rozwiązań. Poniżej przedstawione zostały najpopularniejsze i zarazem najczęściej stosowane rozwiązania.

Mając na myśli sposoby na komunikowanie dokonywanych rezerwacji wybranym węzłom sieci, wyróżnia się [20]:

- sygnalizowanie (ang. *signal*) - zrealizowane za pomocą specjalnego protokołu warstwy III modelu OSI. Przykładem takiego protokołu może być RSVP (*Resource ReSerVation Protocol*) zdefiniowany w dokumencie RFC 2205 [41].
- znakowanie pakietów (ang. *packet marking*) - polega na umieszczeniu informacji odnośnie QoS w obrębie nagłówka pakietu IP, przykładem tego może być pole ToS (*Type of Service*).

Jedną z metod zapewniających nienaruszalność zarezerwowanych zasobów w obrębie węzła sieci jest mechanizm zarządzania buforami (wejściowych i wyjściowych) na interfejsach węzła sieci, do którego trafiają pakiety IP. Do wyżej opisanej metody zalicza się m.in. [20]:

- kolejkowanie pakietów (ang. *packet scheduling/queueing*) - polegające na zastosowaniu określonego algorytmu układania pakietów w buforze, obecnie wyróżnia się m.in.: kolejkowanie priorytetowe, klasowe oraz kolejkowanie „sprawiedliwe”[21],

- kształtowanie przepływu (ang. *traffic shaping*) - są to algorytmy ograniczające prędkości przepływu, stosowane są w miejscach, gdzie występuje konieczność spowolnienia zbyt szybko nadchodzących danych [21];
- unikanie zatorów (ang. *congestion avoiding*) - jest to mechanizm, który to nie pozwala, aby doszło do sytuacji zbytniego zapełnienia buforów [21].

Dostarczanie przewidywalnego poziomu jakości usług sieciowych ma bardzo duży wpływ na bezpieczeństwo sieci komputerowej pod względem poprawności przepływu danych. Ustalając właściwą politykę zarządzania siecią oraz dostarczania odpowiednich parametrów transmisji dla różnego rodzaju klas ruchu sieciowego, zapewniamy, właściwy sposób działania aplikacji, wykorzystujących sieć komputerową jak i Internet.

Nie dopuszczanie do sytuacji, w której nie można przeprowadzić video konferencji, czy to rozmowy za pomocą aplikacji typu VoIP z bardzo ważnym kontrahentem, jest sposobem ochrony takich podmiotów bezpieczeństwa jak np. reputacja. Poprzez zapewnienie odpowiednich parametrów transmisji dla określonego rodzaju ruchu (określonych aplikacji) będących strategicznymi pod względem funkcjonowania firmy, chronimy również firmę przez niepotrzebnymi stratami wynikającymi z przestojów. Nie dochodzi wówczas do sytuacji, w której, ze względu na wykorzystanie prawie całego dostępnego pasma łącza (np. poprzez pobieranie danych z sieci typu Peer-to-Peer), aplikacje nie działają w sposób prawidłowy.

4. Rozpoznawanie aplikacji

W celu lepszego wykorzystania dostępnego łącza internetowego, należy wdrożyć mechanizmy, które dostarczają określony poziom jakości usług. Mechanizmy te opierają się na identyfikacji określonych rodzajów ruchu i stosowaniu dla nich odpowiednich reguł. Jednym z najważniejszych sposobów identyfikacji jest rozpoznawanie aplikacji, w sensie typu, generujących ruch w sieci. Jednocześnie trzeba zauważyć, że rozpoznawanie opierać się będzie na podstawie wiedzy o protokole, który wykorzystuje aplikacja.

Na początku rozdziału przedstawione zostały podstawowe zagadnienia, umożliwiające lepsze zrozumienie sposobów na rozpoznawanie rodzajów aplikacji opisanych w podrozdziałach od 4.2 do 4.5.

4.1. Protokół IP

Protokół IP (wersja 4) jest swoistym kamieniem węgielnym Internetu. W chwili obecnej jest on najczęściej używanym protokołem warstwy III, zarówno w sieciach rozległych, jak i w sieciach lokalnych. Pomimo projektów protokołów nowej generacji, mających zastąpić IP (np. IPv6), protokół ten pozostaje najbardziej powszechnym protokołem sieciowym [20]. Z tego powodu, niniejsza praca skupia się na metodach analizy ruchu w sieciach IP.

Protokół IP zdefiniowany został w roku 1981, w ramach projektu DARPA. Motywacją do jego opracowania była chęć stworzenia sieci działającej w oparciu o przełączanie pakietów. Zadaniem pakietów IP jest transport, zawartej w nim, informacji, pomiędzy nadawcą a odbiorcą [20]. Protokół IP dostarcza procedur wystarczających do przesyłania danych między maszynami znajdującymi się w połączonych sieciach. Protokół definiuje strukturę i format pakietów oraz sposób ich adresowania, nie realizuje jednak żadnych funkcji związanych z poprawnością transmisji, a w szczególności nie identyfikuje pakietów, które mają być przesłane ponownie (retransmitowane). Protokół IP potrafi także wykonywać wielu procesów związanych z odtwarzaniem prawidłowej sekwencji pakietów (pakiety podróżujące różnymi drogami mogą docierać do celu w innej kolejności niż zostały nadane).

Tym samym jest to protokół bezpołączeniowy; nie zapewnia stałego kanału komunikacyjnego [23].

Dopiero współpraca protokołu IP oraz jednego z protokołów warstwy wyższej (warstwy transportu) umożliwia wygodne przesyłanie danych na duże odległości. Przykładami protokołów, które podczas transmisji korzystają z protokołu IP, są TCP i UDP. W takich przypadkach, określając dwa współdziałające protokoły, używa się ich nazw rozdzielonych ukośnikiem np. TCP/IP, UDP/IP. W przypadku TCP komunikacja połączeniowa symulowana jest w kanale bezpołączeniowym poprzez wymianę pakietów i potwierdzeń ich odbioru [23].

Pakiet (inaczej: datagram) jest to podstawowa porcja danych transportowana w (trzeciej) warstwie sieci modelu OSI. Składa się z informacji binarnych obejmujących dane oraz nagłówek [1]. Jest to struktura logiczna, ponieważ datagram stanowi jeden ciąg bitów, nie ma żadnych specjalnych symboli, oddzielających nagłówek pakietu od jego danych. Nagłówek zawsze znajduje się na początku przesyłanego datagramu i zawiera pola charakterystyczne dla danego protokołu.

Poniżej przedstawiona została budowa nagłówka pakietu IP.

Rys 1. Nagłówek pakietu IP

Szczegółowe omówienie funkcji wszystkich pól nagłówka wykracza poza zakres pracy (informacje te można znaleźć w [36]). Z punktu widzenia analizy ruchu w sieciach IP, istotne są następujące pola:

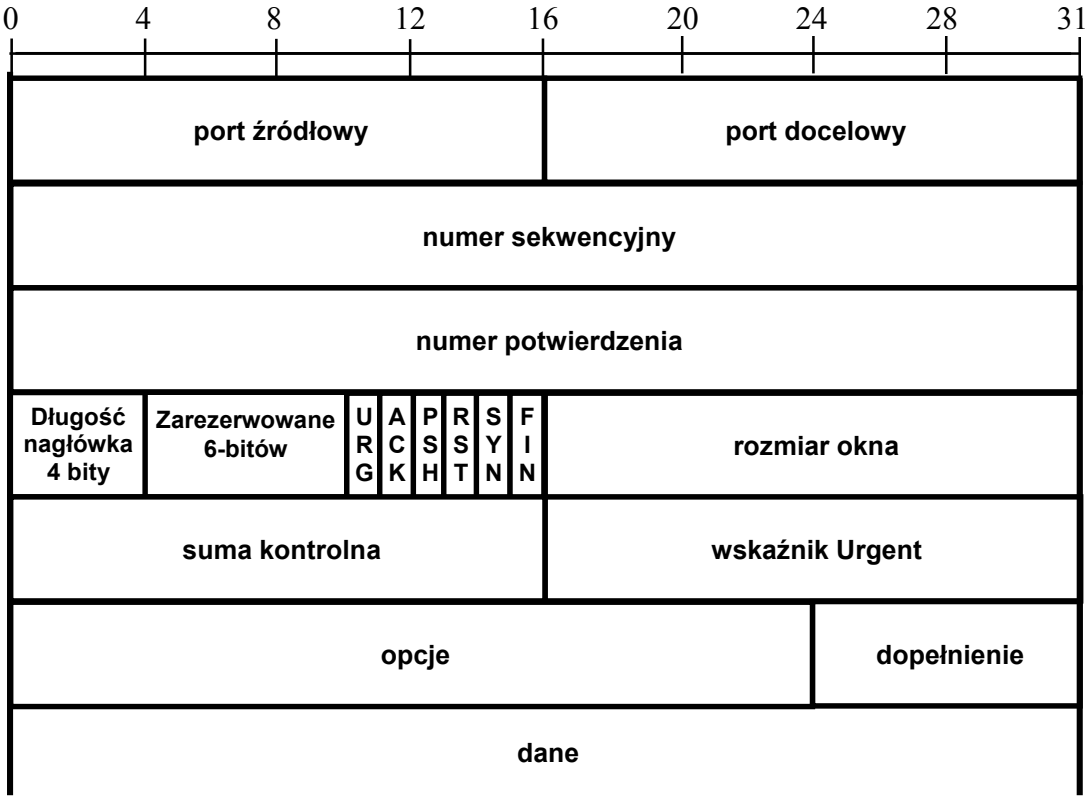
- ToS (ang. *Type of Service*) – określa rodzaj usługi przenoszonej przez dany pakiet;

- protokół (ang. *Protocol*) – zawiera kod, określający protokół warstwy wyższej, której dane są enkapsulowane w pakiecie IP;
- adres nadawcy (ang. *Source Address*) – adres IP komputera źródłowego;
- adres odbiorcy (ang. *Destination Address*) – adres IP komputera docelowego.

4.1.1.Protokół TCP

Protokół TCP (*Transmission Control Protocol*) działa w warstwie czwartej - transportowej modelu OSI. Służy do przesyłania danych między warstwą sieciową a warstwami wyższymi. Protokół oferuje niezawodną usługę dostarczania danych, opartą na strumieniu bajtów i nawiązaniu połączenia. TCP gwarantuje, że wiadomość zostanie dostarczona i aplikacja otrzyma dane w odpowiedniej kolejności. Protokół TCP stosując mechanizm okna transmisji dąży do zoptymalizowania ruchu w sieci oraz uzyskania maksymalnego przepływu danych w poszczególnych połączeniach. W przeciwieństwie do IP, jest to protokół połączeniowy

Budowę nagłówka pakietu TCP dokładnie definiuje dokument RFC 793 [37].



Rys 2. Nagłówek pakietu TCP

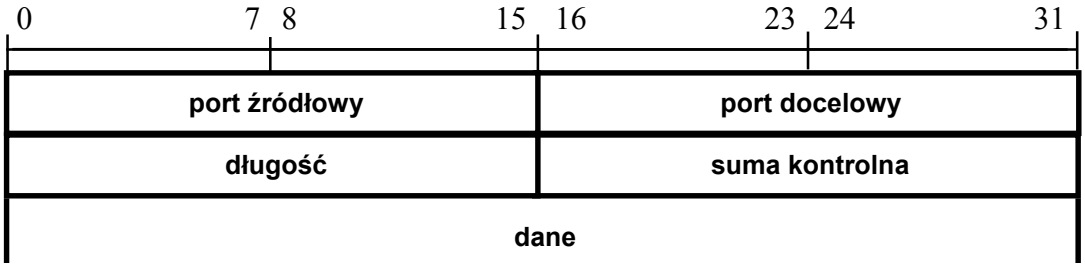
Ze względu na analizę ruchu w sieciach IP, istotnymi polami nagłówka są:

- numer portu źródłowego (ang. *Source Port*) – określa numer portu TCP nadawcy pakietu;
- numer portu docelowego (ang. *Destination Port*) – oznacza numer portu TCP odbiorcy pakietu.

4.1.2.Protokół UDP

Protokół UDP, podobnie jak TCP, jest protokołem warstwy czwartej modelu OSI, służącym do przesyłania danych między warstwą sieciową a warstwami wyższymi. Jest to protokół dużo prostszy od TCP, nie gwarantuje dostarczenie danych ani przesyłania ich w odpowiedniej kolejności. Protokół UDP jest protokołem bezpołączeniowym.

Na poniższym rysunku przedstawiona jest budowa nagłówka pakietu UDP [35].



Rys 3. Nagłówek pakietu UDP

Podobnie jak w przypadku protokołu TCP, ważnymi polami nagłówka są:

- numer portu źródłowego (ang. *Source Port*) – określa numer portu UDP nadawcy pakiet;
- numer portu docelowego (ang. *Destination Port*) – oznacza numer portu UDP odbiorcy pakietu.

4.2. Numery portu TCP i UDP

Aplikacje, usługi oraz programy, komunikujące się przez sieć komputerową, wykorzystując protokoły warstwy czwartej modelu ISO, TCP bądź UDP, używają określonych numerów portów.

Porty są logiczną strukturą, która pozwala na zidentyfikowanie aplikacji, uruchomionej w systemie operacyjnym, do której ma trafić określony pakiet. W nagłówku każdego pakietu TCP i UDP znajdują się pola określające numer portu nadawcy jak i odbiorcy. Dzięki temu gdy oprogramowanie sieciowe otrzyma pakiet UDP lub TCP, może za pomocą numeru portu rozpoznać do jakiej aplikacji należy przesyłka, i skierować nadesłane dane do właściwego programu.

Numery portów zawierają się w przedziale od 1 do 65535, które dzieli się na trzy grupy:

- „dobrze znane numery portów" (ang. *Well Known Ports*), zakres od 1 do 1023 (numer 0 jest zarezerwowany i niewykorzystywany), określone zostały „odgórnie" przez organizację IANA (*Internet Assigned Numbers Authority*) [30], obecnie zajmuje się nimi organizacja ICANN [31]; portów tych używają tylko procesy systemowe, uruchamiane przez administratora lub też programy wykonywane przez uprzywilejowanych użytkowników;
- porty zarezerwowane (ang. *Registered Ports*), zakres od 1024 do 49151, używane są one przez zwykłe procesy użytkowników oraz przez programy wykonywane przez zwyczajnych użytkowników (innych niż uprzywilejowani i administratorzy);
- dynamiczne lub/i prywatne numery portów, mieszczą się w zakresie od 49152 do 65535 i wykorzystywane są m.in. przez aplikacje, które ustalają numer portu podczas połączenia.

Poniżej przedstawione są przykładowe numery portów przyporządkowane dla protokołów związanych z standardowymi, najczęściej spotykanymi usługami oraz programami:

Tabela 2. Przykładowe numery portów wykorzystywane przez protokoły usług/aplikacji [30].

Protokół	Numer portu/protokół	Opis
FTP	20/tcp 20/udp	File Transfer Protocol (Wymiana danych - tryb aktywny)
FTP	21/tcp 21/udp	File Transfer Protocol (Ruch sterujący)
SSH	22/tcp 22/udp	SSH Remote Login Protocol
Telnet	22/tcp 22/udp	Telnet
SMTP	25/tcp 25/udp	Simple Mail Transfer (Poczta elektroniczna)
DNS	53/tcp 53/udp	Domain Name Server
HTTP	80/tcp 80/udp	HyperText Transfer Protocol (przeglądarki stron WWW)
POP3	110/tcp 110/udp	Post Office Protocol - Version 3 (Poczta elektroniczna)
IMAP	143/tcp 143/udp	Internet Message Access Protocol (Poczta elektroniczna)
IRC	194/tcp 194/udp	Internet Relay Chat Protocol
RMT	411/tcp 411/udp	Remote MT Protocol
DC++	412, 1412/tcp (domyślnie) 412, 1412/udp (domyślnie) 411/tcp (serwery) 411/udp (serwery)	Direct Connect (możliwość zmiany portów)
3m-image-lm	1550/tcp 1550/udp	Image Storage license manager 3M Company
Gadu-Gadu	1550/tcp	Gadu-Gadu
Kazaa	zmienne najczęściej używany: 1214/tcp 1214/udp	Kazaa
eMule	4662-4672/tcp (domyślnie) 4662-4672/udp (domyślnie)	eMule (możliwość zmiany portów)

Usługi mogą używać tych samych numerów portów pod warunkiem, że korzystają z innego protokołu (np. TCP albo UDP). Niektóre usługi korzystają jednocześnie z tego samego numeru portu i obydwu protokołów np. DNS korzysta jednocześnie z portu 53 protokołu TCP i UDP.

Rozpoznawanie aplikacji z wykorzystaniem wiedzy o numerze portu, z jakiego korzysta dana usługa/program jest metodą bardzo prostą. Wystarczy sprawdzać pola w nagłówkach protokołów TCP bądź UDP, w których zawarta jest informacja o numerach portów nadawcy i odbiorcy. Aby otrzymać odpowiedź, do jakiej aplikacja dane były adresowane, uzyskane informacje należy porównać z listą znanych numerów portów.

Nie bez znaczenia jest kierunek, w którym pakiet „podążą". W zależności od niego należy porównać odpowiednie pola nadawcy i odbiorcy w nagłówku pakietu TCP bądź UDP. Dla przykładu: jeżeli pakiet związany z ruchem sterującym FTP, przychodzi do komputera, w polu „*Source port*" (numer portu źródłowego) pojawia się wartość 21, natomiast jeżeli pakiet jest wysyłany z komputera wówczas wartość taka pojawia się w polu „*Destination port*" (numer portu docelowego) nagłówka pakietu TCP bądź UDP.

Rozpoznawanie aplikacji na podstawie używanych numerów portów, w odniesieniu do programów/usług korzystających z zakresu portów 1 - 1023 jest metodą dobrą, jednak dla aplikacji używających portów większych niż 1023, metoda ta obarczona jest już dużym błędem. Jak widać w tabeli numer 2, istnieją protokoły aplikacji, które używają tego samego portu co zarejestrowane aplikacje np. port TCP 1550. Niemożliwe jest stwierdzenie, który program generuje ruch wykorzystujący dany port.

Ponadto każdą usługę można tak skonfigurować, aby wykorzystywała inny niż standardowo przydzielony (wykorzystywany) numer portu. Przykładem jest protokół HTTP (używany przez przeglądarki stron WWW), wystarczy ustawić serwer WWW, aby działał na innym porcie niż domyślny. Wówczas, aby skorzystać ze strony uruchomionej na takim serwerze należy jawnie podać numer portu np. wpisując w adres URL http://przykładowa.stron.pl:8080.

Opisany powyżej mechanizm rozpoznawania nie ma zastosowania do aplikacji, które podczas nawiązywania połączenia, dynamicznie ustalają numery

wykorzystywanych portów. Przykładem tego mogą być aplikacje klientów FTP wymieniające dane w trybie pasywnym, które wykorzystując port nr 21 TCP, nawiązują połączenie i ustalają jego parametry, a wymiana danych następuje poprzez inne porty (ustalone między klientem a serwerem w fazie komunikacyjnej, z zakresu 49152 - 65535).

Innym przykładem są programy klientów sieci P2P, które pomimo domyślnie ustawionych numerów portów (np. DC++ [27]) lub puli, możliwych do wykorzystania, portów np. eMule [28], Bittorent [25], posiadają funkcje ustawienia dowolnego numeru portu (co jest niekiedy zalecane przez twórców programów). Powoduje to, że wiedza o wykorzystywanych numerach portów staje się bezużyteczna do identyfikacji protokołu, używanego przez dany typ aplikacji.

4.3. Numer portu TCP, UDP i adres hosta

Metoda ta pozwala na rozpoznanie aplikacji na podstawie wiedzy o używanym przez nie numerze portu i adresie, z jakim się komunikują. Przykładami takich aplikacji są programy działające w sieci Napster, służącej do wymiany plików muzycznych między użytkownikami. Klient sieci Napster w celu wyszukania pliku łączy się z centralnym serwerem, na którym znajduje się baza danych dostępnych w danym momencie plików i adresów komputerów, które je udostępniają [18].

Znajomość numeru portu i adresu serwera pozwala na jednoznaczne zidentyfikowanie ruchu kontrolnego sieci Napster ponieważ sama wymiana plików odbywa się bez udziału centralnego serwera. Klienci pobierający i udostępniający plik są połączeni bezpośrednio ze sobą. Taki rodzaj połączenia nazywamy punkt-punkt (ang. *Peer-to-Peer*). Zablokowanie ruchu kontrolnego powoduje uniemożliwienie korzystania z sieci Napster, co wykorzystywane jest przez administratorów sieci akademickich, które z powodu tego rodzaju ruchu były często przeciążone

Innymi przykładami architektury sieci P2P (ang. *Peer-to-Peer*), wykorzystujących centralny serwer są sieci: Audiogalaxy, OpenNap, WinMX, Direct Connect [18].

Drugim typem sieci P2P, są sieci wykorzystujące kilka serwerów, na których znajdują się bazy danych. Zasada działania tej sieci jest podobna jak w powyższym przykładzie. Dlatego aby wykryć ruch kontrolny, należy znać numer portu i adres serwera, znajdującego się najbliżej klienta sieci P2P. Jedyna różnica polega na tym,

iż program identyfikujący posiada informację o adresach IP serwerów i numerach wykorzystywanych portów. Metoda ta ma zastosowanie do konkretnych przypadków sieci komputerowych, dla których istnieje wiedza dotycząca serwerów sieci P2P znajdujących się „najbliżej" monitorowanej sieci. W przypadku globalnym rozmiar bazy danych oraz koszt obliczeń zbyt duży, aby można byłoby zastosować taką metodę.

Architekturę z wykorzystaniem kliku serwerów z bazami danymi posiadają sieci: eDonkey oraz eDonkey2000 [18].

Metoda opisywana w tym podrozdziale nie da się już skutecznie zastosować do sieci typu *Peer-to-Peer*, o architekturze, w której nie wykorzystuje się serwerów z bazami danymi, a wszelka komunikacja dokonuje się między aktywnymi hostami danej sieci. Przykładami są sieci KaZaA i Gnutella.

4.4. Wielkość pakietów

Wszelkie informacje przesyłane przez sieć komputerową między aplikacjami, transportowane są w pakietach protokołów warstwy czwartej (transportowej) modelu OSI. Są to przeważnie pakiety TCP bądź UDP, które z kolei są enkapsulowane („zamykane") w pakiety protokołu IP, służącego do wymiany informacji między komputerami znajdującymi się w sieci komputerowej.

Pakiety związane z protokołami aplikacji interaktywnych (wymagającymi szybkiej wymiany informacji) mają mały rozmiar w porównaniu do protokołów, które służą do wymiany danych. Przykładami aplikacji interaktywnych są: Telnet oraz programy używające protokółu SSH. Sztandarowym przykładem aplikacji, wymieniającej dane (pliki) za pomocą dużego rozmiaru pakietów, są programy klientów FTP. Rozmiar wysyłanych pakietów nie jest stały i może różnić się w zależności od rodzaju wysyłanych danych bądź wiadomości. Średni rozmiar datagramów pozostaje w przybliżeniu taki sam dla danego rodzaju protokołu.

Analizując średnie rozmiary pakietów można, w przybliżeniu, określić do jakiej aplikacja/usługi (używającej określonego protokołu) należy dany datagram. Jest to metoda dająca mały stopień prawdopodobieństwa poprawnych klasyfikacji, ale może posłużyć jako jedno z kryteriów rozpoznających programy, generujące ruch w sieci. Dzięki jej zastosowaniu, można zawęzić zakres przeszukiwań do protokołów (np. interaktywnych), bądź do protokołów należącej do grupy o określonym rozmiarze

pakietów. W celu dalszego, dokładniejszego wykrycia rodzaju aplikacji, generującej ruch w sieci komputerowej należy stosować inne kryteria.

Przykładem identyfikacji, z wykorzystaniem kryterium rozmiaru pakietów może być schemat działania protokołów SSH [33] i SCP [24], które dzielą port TCP 22. Protokół SCP używany jest do transmisji plików zatem średni rozmiar pakietów jest dużo większy niż SSH, który przesyła jedynie kody wciśniętych klawiszy i wiadomości odświeżające ekran. Ustawiając wartość graniczną kryterium rozpoznającego na rozmiar pakietu 100 bajtów, istnieje duże prawdopodobieństwo, że pakiety, o wielkości mniejszej niż 100 B, przychodzące na port 22 TCP należeć będą do protokołu SSH [9].

Metoda identyfikująca rodzaj aplikacji generującej ruch w sieci komputerowej na podstawie wiedzy o wielkości pakietów, jest bardzo przydatna w przypadku wszelkiego ruchu zaszyfrowanego. Rozpoznanie może odbyć się jedynie na podstawie wiedzy o rodzaju wykorzystywanego protokołu (TCP bądź UDP) bądź numerze wykorzystywanego portu i wielkości pakietu, (ponieważ dane są zaszyfrowane). Przykładem wykorzystania takiej identyfikacji są wspomniane wcześniej protokóły SSH i SCP.

4.5. Inspekcja pakietów

Komputery wymieniające informacje przez sieć komputerową (wsp. w rozdziale 4.4) komunikują się za pomocą protokołu warstwy trzeciej (sieciowej) modelu OSI. Informacje te mają postać pakietów (IP), które w swoim polu danych zawierają datagramy protokołów warstwy wyższej, transportowej (przeważnie TCP, UDP). Protokoły te w swoich polach danych zawierają informacje wymieniane przez aplikacje. Dane mają określoną strukturę.

Można wyróżnić dwa rodzaje formatów przesyłanych informacji:

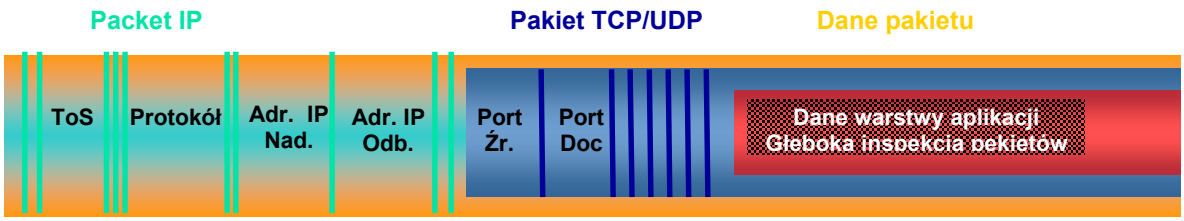
- wiadomości tekstowe – dane wymieniane są w postaci ciągów znaków o określonym kodowaniu, z reguły są to znaki ASCI, początkowy ciąg znaków stanowi nagłówek wiadomości, a kolejne znaki (oddzielone od nagłówka jakimś specyficznym znakiem, stanowią treść wiadomości;

- pakiety – dane wymieniane są za pomocą datagramów, których budowa podobna jest do pakietów warstwy niższych: kilka początkowych bitów stanowią pola nagłówka datagramu, po którym następuje pole danych.

Każda z aplikacji (w sensie typu) używa specyficznego zbioru reguł, czyli protokołu, co pociąga za sobą charakterystyczną budowę wykorzystywanego formatu przesyłanych informacji. Każdy protokół używający do komunikacji pakietów, bądź wiadomości tekstowych, ma inną składnię oraz budowę. Protokoły warstwy aplikacji różnią się między sobą ciągami przesyłanych znaków lub budową nagłówka, czyli ilością i znaczeniem poszczególnych pól datagramu. Dzięki czemu protokoły posiadają takie cechy, które pozwalają na jednoznaczne ich odróżnienie.

Inspekcja pakietów polega na analizie danych generowanych przez aplikacje, które znajdują się w warstwach wyższych, w celu znalezienia konkretnych właściwości. Umożliwia to bardzo dokładną ich identyfikację.

Poniższy rysunek przedstawia pola w nagłówkach pakietów, które są istotne i pomocne w identyfikacji protokołów aplikacji.



Rys. 4. Pola pakietów, wykorzystywanych w rozpoznawaniu protokołów aplikacji [26].

Zaletą inspekcji pakietów jest nie ograniczanie się do badania nagłówków pakietów IP, TCP i UDP, ale również analiza pakietów i wiadomości tekstowych, przesyłanych w warstwie aplikacji. Są to informacje, które znajdują się w polach danych protokołów transportowych pozwalające programom monitorującym w sposób precyzyjny określić, do jakiego protokołu należą analizowane pakiety, co pozwala na bardzo dokładną identyfikację aplikacji generujących ruch w sieci.

W porównaniu do metod przedstawionych w poprzednich podrozdziałach opisana powyżej metoda wymaga więcej czasu i mocy obliczeniowej, w zamian dając największe prawdopodobieństwo poprawnej klasyfikacji.

Rozważając, przedstawione metody, pozwalające na rozpoznawanie rodzaju aplikacji, która generuje ruch w sieci komputerowej można dojść do wniosku, że jedyną skuteczną metodą, stosowaną samodzielnie, jest inspekcja pakietów. Pozostałe metody stosowane oddzielnie są słabym mechanizmem identyfikującym protokoły, natomiast, gdy zastosuje się je razem, dają one dużo większe prawdopodobieństwo poprawnych identyfikacji protokołów. Mimo wszystko nie dadzą one takich rezultatów jak analiza poprzez głęboką inspekcję pakietów.

5. Mechanizmy rozpoznające aplikacje

Rozdział ten poświęcony jest firmowym rozwiązaniom, które na podstawie ruchu w sieciach komputerowych rozpoznają, typu aplikacji, które generują dany ruch. W dalszej części pracy opisane zostały cechy charakterystyczne protokołów, które mogą służyć do rozpoznawania w.w. aplikacji.

5.1. Rozwiązania sprzętowo-programowe

Poniżej przedstawione zostały mechanizmy rozpoznające aplikacje, które bazują na architekturze składającej się ze specjalistycznego oprogramowania stworzonego na platformę sprzętową konkretnego producenta.

5.1.1. Cisco NBAR

Cisco NBAR to jeden z najbardziej reprezentatywnych przykładów mechanizmów rozpoznających aplikacje, na podstawie generowanego przez nie ruchu, w sieciach IP. Od tego rozwiązania zaczerpnięta została nazwa NBAR - *Network-Based Application Recognition*, wykorzystana w tytule pracy.

NBAR to jedna z funkcji systemu operacyjnego urządzeń firmy Cisco, dokładnie systemu Cisco IOS (*Cisco Internetwork Operating System*). Jest to „inteligentny" mechanizm klasyfikujący, który umożliwia identyfikację różnego rodzaju aplikacji, protokołów sieciowych oraz trudnych do sklasyfikowania protokołów, używających dynamicznie przydzielanych numerów portów TCP i UDP. Kiedy aplikacja zostaje rozpoznana i sklasyfikowana przez NBAR, mogą zostać uruchomiane odpowiednie serwisy sieciowe dla danej aplikacji. NBAR, poprzez klasyfikacje pakietów oraz przez współpracę z mechanizmami *Quality of Service* (QoS) stosowanymi dla sklasyfikowanego rodzaju ruchu [14], sprawia, że przepustowość łącza sieciowego jest używana efektywniej.

NBAR posiada funkcje rozpoznające, które umożliwiają identyfikację aplikacji i protokołów z warstw od trzeciej do siódmej modelu OSI.

NBAR klasyfikuje [14]:

- aplikacje, które używają statycznie przydzielonych numery portów TCP i UDP;
- protokoły IP nie wykorzystujące protokołów TCP i UDP: EGP, EIGRP, GRE, ICMP, IPINIP, IPSec;
- aplikacje używające dynamicznie przydzielonych numerów portów TCP i UDP, klasyfikacja takich aplikacji wymaga inspekcji stanowej; jest to zdolność do wykrycia połączenie na podstawie analizy fazy połączenia, w której dokonywany jest przydział portów;
- aplikacje, których klasyfikacja oparta na głębokiej inspekcji pakietów;
- aplikacje, których klasyfikacja ruchu: HTTP na podstawie URL, nazwy hosta lub pola typu MIME; Citrix ICA; ruch RTP na podstawie pola „*Payload type*".

NBAR posiada funkcjonalność nazwaną „*Protocol Discovery*" (wykrywanie protokołów), która umożliwia rozpoznanie protokołu używanego przez aplikację. *Protocol Discovery*, w czasie rzeczywistym, analizuje ruch w sieci w celu poszukiwania znanych wzorców (cech charakterystycznych) protokołów aplikacji, za pomocą, których możliwa jest ich identyfikacja. NBAR rozpoznaje protokoły, których wzorce zaimplementowane są w systemie operacyjnym IOS. Aby dodać nowe definicje, pozwalające na rozpoznanie nowych aplikacji, konieczne jest zainstalowanie pakietu PDLM. *Packet Description Language* (PDL) występuje w dwóch wersjach: jako uaktualnienie bazy lub jako cały zbiór definicji.

NBAR *Protocol Discovery* zajmuje się także dostarczaniem funkcjonalności prowadzenia statystyk dla protokołów, które transmitowane są przez aktywne interfejsy sieciowe. Statystyki zawierają m.in. całkowitą ilość i rozmiar pakietów wychodzących i przychodzących na dany interfejs oraz szybkość ich transmisji. *Protocol Discovery* przechowuje również kluczowe charakterystyki wykonywanych statystyk dla poszczególnych protokołów, w celu ewentualnego ich wykorzystania w definiowaniu klas i polityk QoS [14].

Oprócz zdefiniowanych, w pakietach PDLM, cech charakterystycznych protokołów, istnieje możliwość tworzenia własnych kryteriów rozpoznających protokoły aplikacji, a co z tym idzie samodzielnego rozwijania możliwości identyfikujących mechanizmu NBAR [14].

Jak wspomniano wcześniej, CISCO NBAR umożliwia również klasyfikowanie ruchu RTP na podstawie pola „*Payload type*", które dokładnie charakteryzuje typ

przesyłanych danych. Stwarza to jeszcze większe możliwości klasyfikacji, pozwalające na dokładniejsze zastosowanie metod QoS dla, bardzo wrażliwego na opóźnienia, ruchu przenoszącego głos i obraz w czasie rzeczywistym.

Identyfikacja oraz klasyfikacja ruchu to bardzo ważny etap podczas stosowania usług *Quality of Service*. Dzięki temu etapowi administratorzy sieci komputerowych mogą bardziej efektywnie stosować funkcje QoS w sieciach, w których bardzo często uruchamiane są różnego rodzaju aplikacje. NBAR daje administratorom możliwość spostrzeżenia różnorodności protokołów oraz wielkości ruchu generowanych przez aplikacje, wykorzystujących do komunikacji w.w. protokoły. NBAR umożliwia użytkownikom zdefiniowanie klas ruchu sieciowego, dla których można stosować, różnego rodzaju i na różnym poziomie, usługi sieciowe, pozwalające na lepsze zarządzanie siecią.

5.1.2. Wise – TrafView

Wise – TrafView firmy ETRI to system służący do mierzenia i analizy ruchu w sieci opartym o przepływy.

W dzisiejszych czasach, sieci komputerowe (zwłaszcza sieci rozległe WAN) osiągają szybkość transmisji dochodzącą nawet do terabitów na sekundę (Tbps). W sieciach komputerowych transmitowane są różnego rodzaju dane związane m.in. ze strumieniowym przesyłaniem obrazu i dźwięku (*Streaming media*) (np.: *Windows Media, Real Media, Quicktime*), wymianą plików w sieciach *Peer-to-Peer*, grami sieciowymi, lub nawet atakami sieciowymi. Nie oparte o przepływy pomiary ruchu w sieci są niewystarczające dla takiego rodzaju sieci [15].

Typowe, oparte o przepływy, pomiary definiują przepływ jako zbiór pakietów posiadających wspólne właściwości, analizowanych w pewnym miejscu sieci przez pewien okres czasu. W celach pomiarowych używa się przeważnie pięciu nagłówków pakietu, jednak do pomiarów w dzisiejszych sieciach typowe podejście jest mało wystarczające. Po przeprowadzeniu wielu prób wykazano, że jednoznaczne zidentyfikowanie przepływu, w sytuacji kiedy nowe aplikacje takie jak: P2P, programy przesyłające strumieniowo dane, gry komputerowe, używają dynamicznie przydzielanych numerów portów, jest niemożliwe. Przedstawiona analiza ma duże ograniczenie szczególnie w przypadku gdy większość nowych aplikacji, używa numerów portów nie zarejestrowanych w organizacji IANA.

W celu lepszej identyfikacji przepływów, potrzeba więcej szczegółów, dlatego oprócz danych zawartych w nagłówkach pakietów (podejście standardowe), wykorzystuje się również informacje zawarte w warstwie aplikacji.

Firma ETRI w swoim systemie, wykonuje pomiary ruchu w sieci, wykorzystując koncept przepływów. Definiując przepływ jako zbiór pakietów, które posiadają identyczne wartości pól: adres źródłowy i docelowy IP, numer portu docelowego i źródłowego oraz rodzaj protokołu nie istnieje potrzeba analizowania wszystkich pakietów, gdyż prawie każdy pakiet z danego przepływu zawiera zduplikowane informacje. Wystarczające okazuje się znalezienie w pakiecie odpowiedniej sygnatury, pozwalającej na identyfikację konkretnej aplikacji. Z dużym prawdopodobieństwem można stwierdzić, iż pozostałe pakiety z danego przepływu należą właśnie do tej aplikacji [15].

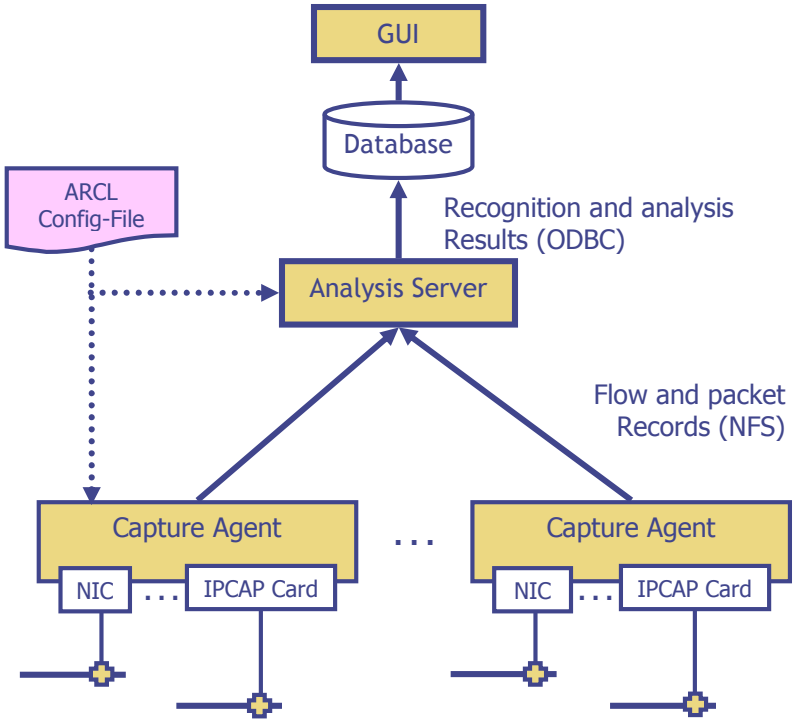
Wise Traf-View stosuje w szeroki sposób, ulepszone, firmowe mechanizmy rozpoznające aplikacje takie jak:

- klasyfikacja aplikacji internetowych (pięć grup);
- „dopasowywanie" sygnatur;
- korelacja przepływów;
- rozpoznawanie na podstawie dynamicznie przydzielanych portów;
- niektóre heurystyki.

Wise Traf-View (WTV) posiada również możliwość klasyfikacji na pod-przepływy, czyli jak miało to miejsce w przypadku CISCO NBAR, ruchu HTTP na np. HTTP_REQ, HTTP_REP, HTTP_REQACK itd.

Wise Traf-View wykorzystuje własny język konfiguracyjny ARCL *(Application Recognition Configuration Language*), który w sposób przejrzysty definiuje, wykorzystywane do rozpoznawania pakietów, wygenerowanych przez dany typ aplikacji, właściwości danego protokołu.

Na poniższym rysunku została przedstawiona architektura systemu WTV



Rys. 5. Architektura systemu *Wise Traf View* [15]

System WTV nie jest zależny od zastosowanego sprzętu sieciowego. Jego działanie polega na „podsłuchiwaniu" transmisji odbywającej się w sieci, poprzez wykorzystanie rozgałęźników sygnału (ang. *splitter*), zarówno elektrycznego jak i optycznego. Sygnał zawierający nie przetworzone pakiety trafia do agentów przechwytujących (ang. *Capture Agent*), którzy dokonują klasyfikacji na przepływy. Następnie dane enkapsulowane są w specjalne struktury (rekordy), wysyłane do serwera analizującego. Serwer rozpoznaje i analizuje dostarczone dane, po czym wyniki swoich działań zapisuje w bazie danych. Dostarczony interfejs graficzny umożliwia wizualizację otrzymanych wyników [15].

Wyżej przedstawiony mechanizm nie obciąża, w sensie mocy obliczeniowej, sprzętu sieciowego jak ma to miejsce w przypadku rozwiązania firmy CISCO (wyłączając przypadki, w których urządzenia posiadają osobne układy realizujące w.w. funkcjonalność). Dają one dużą elastyczność w kwestii analizy w dowolnym miejscu sieci oraz uniezależnia urządzenia sieciowe od stosowanego sprzętu monitorującego. *Wise Traf View* jest systemem biernym, który dostarcza, bardzo dużo i wnikliwych

informacji. System nie ma bezpośredniego wpływu na zastosowane, w monitorowanej sieci, mechanizmy poprawiające wykorzystanie dostępnej przepustowości łącza sieciowych. Wyżej opisane działanie wywiera rozwiązanie przedstawione w rozdziale 5.1.1.

CISCO NBAR analizuje, rozpoznaje oraz klasyfikuje ruch w sieci. Na podstawie tej analizy i podziału na zdefiniowane grupy ruchu sieciowego, stosowane są techniki *Quality of Service*.

Niezależność systemu monitorująco-analizującego, od sprzętu sieciowego, nie jest w żadnym wypadku wadą, gdyż na podstawie uzyskanych analiz, administratorzy sieci komputerowych, mają dużą wiedzę o rodzaju i ilości ruchu transmitowanego poprzez sieć. Dzięki czemu mogą równie dobrze, stosować własne, lub firmowe rozwiązania, poprawiające wydajności sieci komputerowej. W przypadku firmy CISCO analiza i działanie zintegrowana jest w jednym urządzeniu, co czyni je wygodniejszym w użyciu.

5.2. Agenci systemów operacyjnych

5.2.1. Ethereal

Ethereal to najbardziej znany, prawdopodobnie najlepszym, rozpowszechniany na zasadzie oprogramowania typu open source, analizator sieci.

Najważniejsze cechy tego analizatora można przedstawić w następujących punktach [29]:

- posiada możliwość analizowania zarówno pakietów kierowanych do komputera, na którym zainstalowany jest Ethereal, jak i „podsłuchiwanie” pozostałych, nie adresowanych do danego komputera;
- działa w środowisku systemu operacyjnego, komputer wraz z zainstalowanym programem może analizować pakiety w dowolnym miejscu sieci;
- przechwytuje pakiety z sieci a także czyta pakiety zapisane w specjalnych plikach;
- wspiera format filtrów programu Tcpdump;
- posiada możliwość rekonstrukcji sesji TCP i możliwość jej wyświetlenia w kodzie ASCII, EBCDIC, heksadecymalnie oraz w postaci tablic języka C;

- działa na ponad 20 platformach, opartych zarówno o system UNIX jak i Windows;
- czyta dane z różnego rodzaju sieci np. Ethernet, Token-Ring, 802.11 Wireless i innych;
- rozpoznaje 706 protokołów (stan na dzień 15.08.2005) i dzięki temu, że jest to oprogramowanie typu open source, ciągle rozwijane są jego możliwości o identyfikację nowych protokołów;
- uzyskane analizy mogą być zapisywane lub drukowane jako zwykły teks lub jako Postscript;
- posiada możliwość czytania plików (z zapisanymi w nim pakietami – *capture files*), które zostały stworzone przez inne programy „podsłuchujące" (*sniffery*), *routery* i inne narzędzia sieciowe. Wspiera również format zapisywania przechwytywanych pakietów, oparty na popularnej bibliotece libpcap.

Program jest jeszcze w wersji beta, mimo tego jest on bardzo dobrym analizatorem sieci. Dostarcza bardzo dużo wnikliwych informacji na temat przechwyconych pakietów, a także umożliwia rozpoznawanie bardzo dużej ilości protokołów [16].

Ethereal prawdopodobnie nie wykorzystuje mechanizmu przepływów lecz analizuje każdy pakiet z osobna. Jest to analizator, który dostarcza informacje po jakimś okresie czasu dlatego nie musi bardzo szybko przetwarzać przechwyconych pakietów a jedynie buforuje i sukcesywnie je analizuje. Według autora program dokonuje tego prawdopodobnie poprzez porównywanie struktury i danych pakietu z wzorcami, które zostały zdefiniowane w programie. Proces ten odbywa się bardzo sprawnie.

5.3. Cechy charakterystyczne protokołów

W poniższych podrozdziałach przedstawione zostały cechy charakterystyczne protokołów, które poprzez analizę generowanego ruchu, można wykorzystywać do rozpoznawania typu aplikacji.

Ogólnie protokoły można podzielić na dwie grupy. Do pierwszej grupy należą protokoły, których identyfikacja z dużym prawdopodobieństwem poprawnej klasyfikacji, opiera się na informacjach zawartych w nagłówkach pakietu IP, TCP bądź

UDP (por. podrozdziały 5.3.1 i 5.3.2). W drugiej grupie znajdują się protokoły, które wymagają inspekcji pakietów (analiza zawartości danych pakietu) w celu poprawnego rozpoznania protokołu (por. podrozdziały 5.3.3 i 5.3.4.).

5.3.1. Protokół HTTP

HyperText Transfer Protocol (HTTP) to jednokierunkowy, bezpołączeniowy protokół warstwy aplikacji, który leży u podstaw działania *World Wide Web*. Protokół ten określa sposób formatowania i przesyłania dokumentów oraz komendy sterujące pracą serwerów internetowych i przeglądarek WWW [23].

Cechą charakterystyczną HTTP jest wykorzystywanie do komunikacji protokołu TCP i portów o numerach 80, 8008, 8080. Największe prawdopodobieństwo poprawnej klasyfikacji protokołu HTTP w oparciu o numery portów, istnieje wówczas gdy, serwery WWW działają z wykorzystaniem portu 80 protokołu TCP. Numery portów z zakresu 1-1024 są zastrzeżone i przydzielane tylko i wyłącznie przez organizacje IANA (wsp. w rozdziale 2). dlatego porty z tego zakresu, nie powinny być używane przez inne protokoły niż zdefiniowane w organizacji.

W protokole TCP jak i UDP istnieją dwa pola, które określają wykorzystywane porty: numer portu źródłowego i portu docelowego. Charakteryzując protokół (po konkretnym numerze), należy sprawdzić, kierunek transmisji. Jeżeli pakiet przechodzi przez urządzenie, na którym zainstalowany jest program analizujący, z sieci wewnętrznej do sieci Internet, wówczas (jeżeli pakiet związany jest z protokołem HTTP) numer portu docelowego będzie równy 80. Jeśli pakiet przychodzi do sieci wewnętrznej, sytuacja będzie odwrotna - pole portu źródłowego równe jest 80.

Do zidentyfikowania pakietu należącego do protokołu HTTP, należy sprawdzić czy:

- pakiet transmitowany jest za pomocą protokołu TCP, analiza pola „protokół" nagłówka pakiety IP (por. rozdz. 4.1), w którym w którym zawarty jest pakiet TCP;
- numer portu, w zależności od kierunku transmisji pakietu, wynosi 80; informacja zawarta w polach portu źródłowego i docelowego nagłówka pakietu TCP (por. rozdz. 4.1.1).

Sytuacja w sieciach IP odbiega czasami od tego co zostało zdefiniowane w organizacji IANA. Na porcie 80 niekiedy transmitowane są pakiety związane z innym rodzajem protokołu. Co wykorzystywane jest przez hackerów, używających tego portu, do uzyskania dostępu do sieci z zainstalowaną zaporą ogniową, w której nie stosowane są dodatkowe zabezpieczenia, a filtrowanie pakietów odbywa się tylko na podstawie analizy numerów portu.

W celu dokładniejszego rozpoznania można zastosować inną metodę identyfikującą, tzw. inspekcję pakietów, polegającą na sprawdzaniu danych znajdujących się zaraz po nagłówkach IP i TCP. Identyfikacja opiera się na analizie ciągów znaków (ponieważ protokół HTTP jest protokołem tekstowym; wymiana informacji polega na przesyłaniu odpowiednich wiadomości tekstowych).

Format przesyłanych wiadomości tekstowych:

```
HTTP-message   =   | Full-Request         ; HTTP/1.0 messages
                   | Full-Response


Full-Request   = Request-Line
                            *( General-Header
                            | Request-Header
                            | Entity-Header )
                           CRLF
                           [ Entity-Body ]

Request-Line = Method SP Request-URI SP HTTP-Version CRLF


Full-Response   = Status-Line
                            *( General-Header
                            | Response-Header
                            | Entity-Header )
                           CRLF
                            [ Entity-Body ]

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF
```

Schemat 1. Format wiadomości HTTP w wersji 1.0 [40]

Analizując powyższą konwencję wiadomości, zauważyć można, że każdy komunikat, niezależnie od tego czy jest on typu „żądanie”, czy „odpowiedź”, w pierwszej linii zawiera ciąg znaków „HTTP”. W celu lepszej identyfikacji pakietu jako HTTP, należy sprawdzać, czy w ciągu danych między końcem nagłówka TCP

a pierwszym znakiem końca linii i „powrotu karetki” (CRLF) występuje wspomniany ciąg znaków.

Dokładna analiza jest ważną kwestią nie tylko podczas precyzyjnych pomiarów ruchu, ale ogrywa znaczącą rolę w funkcjonowaniu dynamicznych zapór ogniowych. Dzięki niej można sprawdzić czy port 80 wykorzystywany jest przez protokół HTTP, pozwalając dynamicznej zaporze zareagować odpowiednio szybko i zablokować ewentualnie niepożądaną transmisję.

5.3.2. Protokół SSH i SCP

SSH (*Secure Shell*) to standard protokołów komunikacyjnych używanych w sieciach komputerowych TCP/IP, w architekturze klient-serwer. Protokół SSH służy do terminalowego łączenia się ze zdalnym komputerem. Jest on następcą protokołu Telnet, z tą różnicą, że przesyłane przez niego dane są zaszyfrowane [24]. Protokół SSH dostarcza**,** użytkownikowi taką samą funkcjonalność jak ma to miejsce w przypadku Telnetu: przesyłanie przez sieć komputerową, wyświetlanych na ekranie, znaków oraz kodów wciśniętych klawiszy (ale w postaci zaszyfrowanej).

Istnieją dwie wersje protokołu SSH: SSH1 i SSH2. W drugiej wersji, możliwe jest użycie dowolnych sposobów szyfrowania danych i czterech różnych sposobów rozpoznawania użytkownika, podczas gdy SSH1 obsługuje tylko stałą listę kilku sposobów szyfrowania i dwa sposoby rozpoznawania użytkownika (klucz RSA i zwykłe hasło) [24].

SCP (*Secure Copy Protocol*) jest protokołem transportowym, służącym do przesyłania plików przez sieć komputerową. Protokół SCP i SSH używają identycznej techniki szyfrowania danych i rozpoznawania użytkownika. Można powiedzieć, że te dwa protokoły należą do jednej rodziny protokołów ogólnie nazwanych SSH, do których zaliczamy również protokoły służące do przesyłania plików (SFTP), zdalnej kontroli zasobów, tunelowania i wielu innych zastosowań [24].

Protokoły z rodziny SSH (np. SSH i SCP) korzystają domyślnie z protokołu TCP i portu 22, czyli współdzielą ten sam port 22, dlatego klasyfikacja na podstawie numeru portu nie da jednoznacznej identyfikacji, ale zawęzi ją, z bardzo dużym prawdopodobieństwem, do w.w. protokołów.

Ze względu na zaszyfrowanie danych ograniczone jest również zastosowanie identyfikacji protokołu poprzez inspekcję pakietów. Biorąc pod uwagę fakt, iż protokół

SSH służy jedynie do przesyłania znaków i kodów wciśniętych klawiszy, a SCP do przesyłania plików, zaobserwować można, że średnia wielkość pakietów protokołu SCP jest dużo większa od wielkości pakietów protokołu SSH. Wykorzystując te informacje, można zastosować mechanizm klasyfikujący, który sprawdza rozmiar, transmitowanych przez port 22, pakietów i rozgranicza je pod względem wielkości (większe niż 100 B jako SCP, a pozostałe jako SSH) [10].

Wyżej opisana metoda obarczona jest jednak pewnym marginesem błędu, ponieważ przez port 22 TCP mogą być przesyłane pakiety związane z innymi protokołami niż SSH i SCP (tak być nie powinno); nie zawsze pakiety sesji SSH mają rozmiar mniejszy niż 100B, czasami dochodzi on do 400 B.

Zastosowanie kryterium identyfikującego, w postaci numeru portu i wielkości pakietu odnośnie protokołów SSH i SCP, daje dużą ilość poprawnych klasyfikacji.

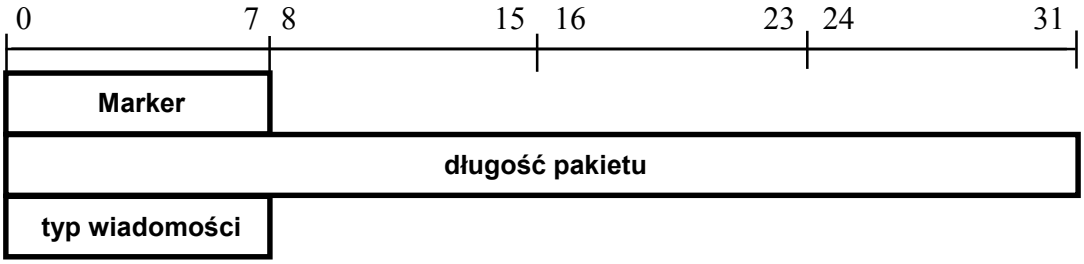
5.3.3. Protokół eDonkey

Protokół eDonkey („Osiołek") jest wykorzystywany, w sieci typu *Peer-to-Peer*, do wymiany plików miedzy użytkownikami Internetu.

W skład sieci eDonkey wchodzą klienci i serwery a każdy klient łączy się z głównym serwerem za pomocą protokołu TCP. Podczas fazy „sygnalizacyjnej", do której należy wszelka komunikacja mająca na celu wyszukanie pożądanej treści, do głównego serwera wysyłane jest żądanie wyszukiwania. Kiedy klient otrzyma listę użytkowników (od których może pobrać dane) ustanawia z nimi bezpośrednie połączenie, z wykorzystaniem protokółu TCP, pytając każdego użytkownika o inny kawałek pliku [10].

Po przeanalizowaniu fazy „sygnalizacyjnej" i fazy „pobierania", do której należy wszelka komunikacja związana z pobieraniem pliku, widocznym jest, że pakiety mają ustalony format nagłówka.

Na poniższym rysunku przedstawiony został schemat nagłówka występującego po nagłówku TCP.



Rys. 6. Nagłówek pakietu eDonkey [10]

Pakiet zawiera następujące pola:

- „Marker", który ma zawsze wartość heksadecymalną równą 0xe3;
- długość pakietu (ang. *packet Length*), określa w bajtach rozmiar treści wiadomości eDonkey (nie licząc długości pół „Marker" i „długość pakietu").

Po przeanalizowaniu powyższych cech charakterystycznych, w celu rozpoznania pakietu eDonkey można zastosować następujące sygnatury identyfikujące:

- pierwszy bajt zaraz po nagłówkach IP i TCP stanowi marker eDonkey, o wartości heksadecymalnej 0xe3;
- liczba zawarta w następnych czterech bajtach powinna być równa rozmiarowi całego pakietu (IP) (w bajtach) minus wielkości nagłówków IP i TCP oraz minus 5 bajtów (pierwsze pola nagłówka eDonkey).

Aby stwierdzić, czy dany pakiet związany jest z protokołem eDonkey należy zanalizować pakiety pod kątem występowania w.w. właściwości.

5.3.4. SIP

Protokół SIP (*Session Initiation Protocol*) jest protokołem warstwy aplikacji odpowiedzialnym za ustalanie, zmianę i kończenie multimedialnych sesji zestawianych pomiędzy uczestnikami połączenia w sieciach IP.

Protokół SIP zapewnia szeroki zakres usług, takich jak telefonia internetowa, przeprowadzanie konferencji multimedialnych, rejestracja i przeadresowywanie usług oraz upraszczanie połączeń z sieciami VPN. Jak wskazuje nazwa, protokół *Session Initiation Protocol* odpowiada za sygnalizację sesji, czyli wymianę komunikatów

w celu nawiązania sesji komunikacyjnej przekazującej głos lub multimedia. SIP wywodzi swoje korzenie z kilku inicjatyw IETF (*Internet Engineering Task Force*) i jest ściśle związany z technologiami i standardami dotyczącymi weba i poczty elektronicznej [13].

Z przyczyn wydajnościowych, protokół SIP używa protokołu UDP (*User Datagram Protocol*), użycie zaś protokołu TCP jest opcjonalne. Ze względu na zawodną naturę protokołu UDP, SIP zawiera własne mechanizmy retransmisji, w tym, podczas ustalania sesji, trójstronną wymianę komunikatów pomiędzy węzłami [13].

Protokołu SIP używa się do zaproszenia uczestników do sesji, natomiast protokół SDP (*Session Descriptions Protocol*) dekoduje główną część komunikatu SIP, zawierającą informacje o rodzaju mediów, których uczestnicy sesji mogą i chcą użyć. Natychmiast po wymianie i potwierdzeniu informacji wszyscy uczestnicy powinni być zorientowani co do adresów IP uczestników, dostępnego pasma i rodzaju medium. Następnie rozpoczyna się przekazywanie danych z użyciem odpowiedniego protokołu transportowego (bardzo często jest to protokół RTP). Podczas całej sesji uczestnicy, wysyłając dodatkowe komunikaty protokołu SIP, aktualizują sesję poprzez np. dodanie nowego zbioru mediów, nowych uczestników sesji lub dokonanie innych zmian [13].

Miejsca przeznaczenia w protokole SIP są reprezentowane przez wskaźniki URI (*Uniform Resource Indicator*), które mają formaty takie same jak adresy poczty elektronicznej. Oznacza to konieczność użycia systemu DNS do przekształcenia nazw hostów i domen na adresy IP. Wsparcie systemem DNS odgrywa kluczową rolę w integracji protokołu SIP z technologiami umożliwiającymi implementację weba i poczty elektronicznej, technologiami ściśle związanymi ze wskaźnikami URI i ich odwzorowaniem [13].

Wiadomość SIP ma następującą postać:

```
generic-message  =  start-line
                    *message-header
                    CRLF
                    [ message-body ]

start-line  =  Request-Line / Status-Line

Request-Line  =  Method SP Request-URI SP SIP-Version CRLF

Status-Line  =  SIP-Version SP Status-Code SP Reason-Phrase CRLF
```

Schemat 2. Format wiadomość SIP [43].

Zauważyć można, iż niezależnie od tego, czy to jest wiadomość typu „żądanie”, czy „odpowiedź”, zawsze w pierwszej linii wiadomości (do znaku powrotu karetki i następnej linii - CRLF), występuje ciąg znaków „SIP/2.0”.

Aby rozpoznać czy przechwycony pakiet, związany jest z protokołem SIP, należy sprawdzić, czy zaraz po nagłówku IP i TCP, do pierwszego znaku CRLF, występuje ciąg znaków „SIP”. Taka inspekcja pakietów, w celu znalezienia sygnatury jednoznacznie identyfikującej protokół SIP, sięga w swojej analizie, do danych przesyłanych w warstwie aplikacji.

6. Opis części praktycznej

6.1. Wstęp

W dzisiejszych czasach, sieci komputerowe (zwłaszcza sieci rozległe WAN) osiągają szybkość transmisji dochodzącą nawet do tera bitów na sekundę (Tbps). Transmitowane są różnego rodzaju dane, związane m.in. ze strumieniowym przesyłaniem obrazu i dźwięku *(Streaming media*), wymianą plików w sieciach *Peer-to-Peer*, grami sieciowymi, czasem nawet związane z atakami sieciowymi. Ze względu na rosnące oczekiwania użytkowników, rozmiary danych stają się coraz większe, jest ich coraz więcej i transmitowane są coraz częściej. Przy nie zmieniającej się budowie sieci komputerowej, pojawiają się zatory. Usługi sieciowe takie jak np. telefonia internetowa (inne wrażliwe na opóźnienia), działają bardzo źle, lub ich działanie jest przerywane, ponieważ inni użytkownicy pobierają treści/dane, które zajmują prawie całe dostępne pasmo łącza internetowego [15].

Jednym ze sposobów zapobiegania tej sytuacji jest zwiększenie przepustowości sieci (przyłącza internetowego), jednak rzadko kiedy zaspokaja to oczekiwania użytkowników, którzy mając szybsze łącze, będą prawdopodobnie pobierać więcej i z szybszą prędkością, rezerwując prawie całe dostępne pasmo łącza internetowego. Bardzo dobrym rozwiązaniem okazuje się zastosowanie mechanizmów poprawiających wydajność działania sieci komputerowej, jak np. *Quality of Service*. Jeśli zależy nam na dużym bezpieczeństwie można zastosować dynamiczne zapory ogniowe, które, w zależności od sytuacji występującej w sieci, w sposób „inteligentny" blokują niedozwolony rodzaj ruchu.

Powinno się przy tym pamiętać, że nie mniej ważna od stosowanych mechanizmów zrządzających ruchem, jest jego identyfikacja. Jest to podstawa dalszych działań, dzięki którym możliwe jest rozpoznanie, rodzaju ruchu przechodzącego przez sieć komputerową. Wykorzystując otrzymane informacje można blokować niedozwolony rodzaj ruchu (dynamiczne zapory ogniowe), a także lepiej skonfigurować mechanizmy optymalizujące wykorzystanie dostępnego łącza internetowego (np. agent sterujący klasami QoS). Ma to szczególne znaczenie w działaniach dążących do zapewnienia odpowiedniego poziomu bezpieczeństwa sieci komputerowych.

6.2. Zakres

Celem części praktycznej pracy dyplomowej jest stworzenie aplikacji, która na podstawie ruchu generowanego w sieciach IP, rozpoznaje jakiego rodzaju aplikacje go generują. Utworzona aplikacja swoją funkcjonalnością przypomina działanie mechanizmu NBAR (firmy CISCO). Dzięki otrzymanym informacjom możliwe jest opracowanie zasad sterowania zaporą ogniową oraz agenta sterującego klasami QoS.

W założeniu, program mógłby pracować na serwerze, ruterze, znajdującym się w miejscu, przez które przechodzi cały ruch z sieci wewnętrznej do Internetu. Z tego powodu analiza skupia się na ruchu przychodzącym i wychodzącym z sieci wewnętrznej, z pominięciem ruchu lokalnego.

Program składa się z modułu głównego i dołączonych do niego wtyczek, które rozszerzają jego możliwości o rozpoznawanie nowych protokołów.

6.3. Opis programu

Program „NBAR monitor” składa się z dwóch części: modułu głównego (plik „NBAR monitor.exe”) i dołączanych do niego wtyczek (ang. *Plug-ins*), którymi są biblioteki DLL.

6.3.1.Wykorzystane narzędzia

Program został napisany w języku programowania C++ z wykorzystaniem:

- narzędzia do tworzenia wizualnego oprogramowania Microsoft Visual Studio .NET;
- środowiska developerskiego w wersji 2002 (Microsoft Development Environment 2002);
- biblioteki Microsoft .NET Framework 1.0;
- biblioteki WinPcap wersja 3.0 - jest to open source’owa biblioteka służąca do przechwytywania i analizy pakietów na platformach Win32.

6.3.2. Moduł główny

Działania realizowane przez moduł główny :

- Rozpoznawanie aktywnych interfejsów sieciowych i umożliwianie wyboru jednego z nich; na wybranym interfejsie dokonywany jest proces rozpoznawania typów aplikacji (protokołów). Nazwa wybranego interfejsu zapisywana jest do pliku konfiguracyjnego config.cfg, w formacie tekstowym.
- Wyszukiwanie wtyczek, które znajdują się w katalogu *„Plug-Ins”*, wczytywanie ich oraz odczytywanie numeru i nazwy rozpoznawanego przez wtyczkę protokołu.
- Podział przechodzącego przez dany interfejs ruchu na przepływy, definiowane jako grupy pakietów, posiadających takie same wartości w polach: numer IP nadawcy i odbiorcy, numer potu docelowego i źródłowego. Ruch generowany przez konkretną aplikację dzielony będzie na dwa przepływy: wychodzący i przychodzący. Przepływy ulegają przeterminowaniu, jeżeli przez 9 sekund nie zaklasyfikowano do niego pakietu. Liczba przepływów jest ograniczona do 300. Wartość ta została ustalona przez autora pracy w wyniku przeprowadzanych testów i uznana za wystarczającą dla działania aplikacji.
- W zależności od zakresu portów rozpoznanych w przepływie w celu rozpoznania protokołu, do którego mogą należeć dane pakiety, analizowanych jest pierwszych pięć pakietów – dla zakresu portów od 1024 i pierwsze 2 dla 1-1023, Do analizy nie liczone są pakiety, które związane są z fazą ustalania sesji TCP, czyli pakiety m.in. z ustawionymi flagami „SYN”, „ACK”, „SYN” i „ACK”.
- Dla każdego z przepływów zliczana jest ilość i rozmiar pakietów.
- Prezentuje w formie tabelki otrzymane wyniki.
- Zapis (do tablicy) wyników (będącego sumą analiz poszczególnych sesji), w celu późniejszego, ewentualnego wykorzystania przez inne programy. Wyniki przechowywane są do końca miesiąca, w którym program został uruchomiony.

6.3.3. Ogólna zasada działania programu „NBAR monitor”

a) Program sprawdza ilość aktywnych interfejsów sieciowych. Jeżeli nie zostanie znaleziony taki interfejs pojawia się komunikat: „Nie znaleziono interfejsów. Sprawdź, czy zainstalowałeś pakiet WinPcap”, a następnie komunikat: "Nie wybrano żadnego adaptera, program zakończy swoje działanie!!!" po czym program zakończy swoje działanie.

b) Jeżeli znaleziono aktywny interfejs, program szuka w katalogu „Config” plik config.cfg, jeżeli go znajdzie, sprawdza nazwę zapisanego w nim interfejsu. Jeżeli się zgadza z nazwą interfejsu aktywnego w systemie operacyjnym, automatycznie zostaje on wybrany, jeżeli nie lub plik nie istnieje pojawia się okno dialogowe „Wybór adaptera”, które pozwala na wybór interfejsu, na którym dokonywany będzie nasłuch. Nazwa wybranego interfejsu zapisywana jest do pliku konfiguracyjnego. Jeżeli nie wybierzemy interfejsu (tzn. naciśniemy anuluj) pojawia się komunikat "Nie wybrano żądnego adaptera, program zakończy swoje działanie!!!".

c) Po pomyślnym wyborze interfejsu, program szuka w katalogu *„Plug-Ins”* wtyczki o nazwach rozpoczynających się od ciągów „p_” i „i_” o rozszerzeniu „dll”. Następuje wczytywanie wtyczek oraz odczytywanie nazw i numerów protokołów, których rozpoznawanie umożliwiają wczytane plug-in’y.

d) Po uruchomieniu procesu nasłuchiwania (przycisk „Start”), uruchamiane są dwa wątki. Pierwszy z nich wykonuje następujące zadania:

- sprawdzanie czy pakiety związane są z ruchem lokalnym;
- szereguje ruch przechodzący przez interfejs na przepływy;
- zlicza ilość i rozmiar pakietów należących do konkretnego przepływu;
- analizuje pakiety w przepływach.

Drugi wątek odpowiada za:

- zapisywanie danych do tablicy globalnej;
- porządkowanie statystyk w tablicy globalnej, które odbywa się przy zmianie minuty i godziny;
- sprawdzanie ważności przepływów, jeżeli do przepływu nie zakwalifikowano pakietów w ciągu 9 sekund, wówczas przepływ uważany jest za nieważny;
- prezentowanie wyników działania na ekranie monitora.

Drugi wątek uruchamiany jest co 5 sekund, w celu wykonania powyższych czynności. W konsekwencji wyniki odświeżane są na ekranie co 5 sekund.

Dane z analiz poszczególnych sesji zapisywane są w tablica globalnej, której elementy są następującego typu:

```
typedef struct stat_prot{
       int nr_prot;            -- numer protkołu
       u_short kierunek_in;    -- kierunek przepływu
       CString nazwa;          -- nazwa rozpoznanego protokołu
       minuty t_minut[60];     -- statystyki w aktualnej minucie
       typ_czas t_godz[24];    -- statystyki w aktualnej godzinie
       typ_czas t_dzien[31];   -- statystyki w danym dniu
 }stat_prot;
```

gdzie:

```
typedef struct minuty{
       u_int poj;
       u_int ilosc_pakietow;
 }minuty;

typedef struct typ_czas{
       float poj;
       u_long ilosc_pakietow;
 }typ_czas;
```

6.4. Mechanizm wtyczek

6.4.1. Wstęp

Mechanizm wtyczek polega na dołączaniu w trakcie działania aplikacji dodatkowego kodu, mającego rozszerzać lub modyfikować działanie programu. Rozwiązanie to posiada następujące zalety [17]:

- daje bardzo szerokie możliwości, ograniczone jedynie przez dozwoloną ingerencję w aplikację macierzystą;
- działanie wtyczek jest szybkie – realizowane jest przez procesor, bez pośrednictwa interpretera jak to ma miejsce w przypadku skryptów;

- wtyczka może korzystać ze wszystkich funkcji udostępnionych w środowisku (w systemie operacyjnym), w szczególności może tworzyć okna w środowiskach okienkowych oraz wykorzystywać zainstalowane biblioteki (np. obsługujące grafikę trójwymiarową, czy przetwarzającą dźwięk);
- kod źródłowy wtyczki może zostać napisany w dowolnym języku programowania;
- kolejne wersje wtyczek mogą być wypuszczane niezależnie od kolejnych wersji samej aplikacji;
- wtyczki mogą być tworzone przez producentów nie związanych z twórcą samej aplikacji.

Mechanizm działania wtyczek posiada wiele zalet, dlatego zastosowany został w celu rozszerzania działania programu głównego, o możliwość rozpoznawania dodatkowych protokołów.

6.4.2. Konwencja pisania wtyczek

W celu zaimplementowania mechanizmu wtyczek w pracy wykorzystano dynamicznie łączone biblioteki DLL *(Dynamic Link Library)*. Wtyczki, tak samo jak program, napisane zostały w języku programowania C++.

Biblioteka łączona dynamicznie jest zbiorem funkcji oraz danych, które aplikacja może dynamicznie (tj. w trakcie wykonywania) przyłączać do swojego kodu. Kod źródłowy biblioteki to zwykły kod w języku programowania z dodatkowymi oznaczeniami niektórych funkcji lub danych, które będą eksportowane (tj. dostępne dla aplikacji, ładujących daną bibliotekę) [17].

Aby oznaczyć funkcję lub zmienną jako eksportowaną w bibliotece DLL (napisanej w języku C/C++) należy w jej definicji dopisać [12]:

- słowo kluczowe – „`extern`”(przed typem);
- „`declspec(dllexport”)`, przed lub za typem (ze słowem `extern`)

W celu zwiększenia czytelności kodu zdefiniowano `__declspec(dllexport)` jako `PLUG_API`, poprzez umieszczenie definicji:

```
#define PLUG_API __declspec(dllexport)
```

Zastosowano również następujący standard nazw funkcji i procedur:

- procedura zwracająca nazwę protokołu rozpoznawanego przez wtyczkę ma zawsze taką samą nazwę: `GetProtocolName`
- procedura zwracająca numer protokołu rozpoznawanego przez wtyczkę ma zawsze taką samą nazwę: `GetNrProtocol`
- funkcja rozpoznająca z wykorzystaniem cech charakterystycznych zdefiniowanych we wtyczce ma nazwę: `CheckProtocolIN`

Pisząc nowe biblioteki DLL, należy przestrzegać przedstawionego powyżej standardu nazewnictwa. W innym przypadku wtyczki nie będą działały prawidłowo.

Jest to spowodowane mechanizmem działania programu, który wywołuje procedury i funkcje o podanych w standardzie nazwach, ale z różnych, wczytanych uprzednio, bibliotek. Jeżeli zostanie stworzona wtyczkę z innymi nazwami funkcji i (umieszczona w katalogu „Plug-ins") wówczas nie będą wykorzystywane funkcje, które odbiegające od przyjętego nazewnictwa. Stosowanie nazwy `PLUG_API` nie jest konieczne, gdy przed lub za typem zostanie umieszczony (zawsze ze słowem `extern`) ciąg znaków: `__declspec(dllexport)`.

Wszystkie wtyczki, wykorzystywane w programie „NBAR monitor", znajdują się w katalogu „*Plug-Ins*". Można je podzielić na dwie grupy: rozpoznające protokoły po numerach portów i takie, które dokonują inspekcji pakietów.

Pierwsza grupa posiada nazwę pliku poprzedzoną przedrostkiem „p_" np. „p_DNS.dll", a druga z nich „i_", czego przykładem jest nazwa „i_RTP". Należy przestrzegać nazewnictwa plików, ponieważ wtyczki wyszukiwane są po przedrostkach.

Program obsługuje maksymalnie 48 wtyczek.

6.4.3. Dostarczone wtyczki

Poniżej przedstawione zostały nazwy plików bibliotek DLL, które dostarczono wraz z programem. Pliki podzielono na dwie grupy.

Grupa I (nazwana „rodziną I"): pliki Służące do rozpoznawania protokołów po numerach portów -wykorzystywane dla przepływów, w których pakiety transmitowane są poprzez porty z zakresu 1-1023:

p_DHCP.dll - protokół DHCP
p_DNS.dll - protokół DNS
p_FTP_a.dll - protokół FTP ruch kontrolny i wymiana danych w trybie aktywnym
p_HTTP.dll - protokół HTTP
p_HTTPS.dll - protokół HTTPS (HTTP transmisja szyfrowana)
p_IMAP.dll - protokół IMAP4 - transmisja szyfrowana i nieszyfrowana
p_POP.dll - protokół POP3 - transmisja szyfrowana i nieszyfrowana
p_SCP.dll - protokół SCP
p_SSH.dll - protokół SSH

Grupa II (nazwana „rodziną P"): pliki służące do rozpoznawania protokołów z wykorzystaniem inspekcji pakietów. Tego rodzaju pliki używane w analizie przepływów, w których pakiety transmitowane są przez porty o numerach większych od 1023:

i_BitTorrent.dll - protokół sieci BitTorrent
i_eDonkey.dll - protokół sieci eDonkey
i_RDP.dll - protokół RDP
i_RTP.dll - protokół RTP
i_SIP.dll - protokół SIP

Program „NBAR monitor" domyślnie (funkcja wbudowana w program), rozpoznaje protokół ICMP, oraz pakiety IP, które używają innego protokołu warstwy wyższej od ICMP, TCP i UDP (wyświetlana jest nazwa „inny w. wyzszej"). Jeżeli nie rozpoznano jakiegoś protokołu wykorzystującego protokół TCP lub UDP wówczas wyświetlana jest nazwa „inny TCP" lub „inny UDP". Klasyfikowanie przepływów na cztery wymienione grupy, ma miejsce wówczas, kiedy nie zostanie wczytana, żadna ze wtyczek.

6.5. Cechy charakterystyczne protokołów

6.5.1. Protokół DHCP

Dynamic Host Configuration Protocol (DHCP) to protokół komunikacyjny umożliwiający komputerom uzyskanie, od serwera danych konfiguracyjnych np. adresu IP hosta, adresu IP bramy sieciowej, adresu serwera DNS, maski sieci.

Protokół ten do komunikacji wykorzystuje porty 67 i 68 protokołu UDP. Jeżeli pakiet kierowany jest od klienta do serwera DHCP wówczas pola w nagłówku pakietu UDP mają następujące wartości: port źródłowy 67 -port docelowy 68. Jeżeli pakiet podróżuje w kierunku przeciwnym wówczas wartości te wynoszą: port źródłowy 68 - port docelowy 67.

Opisana właściwość protokołu DHCP, autor pracy wykorzystał do rozpoznawania pakietów należących do tego protokołu.

6.5.2. Protokół DNS

Wykorzystywany jest jako nośnik informacji pomiędzy serwerami, klientami systemu DNS, który umożliwia translację adresów IP na domenowe (i odwrotnie).

Cechą charakterystyczną protokołu DNS, wykorzystywaną przez autora pracy do jego identyfikacji, jest możliwość wysyłania pakietów za pomocą protokołów TCP i UDP z wykorzystaniem portu 53.

Sprawdzanie numeru portu zależy od kierunku transmisji pakietu: jeżeli pakiet przechodzi przez interfejs, na którym nasłuchuje program NBAR monitor, w kierunku „do świata", (z sieci wewnętrznej), wówczas sprawdzana jest wartość numeru portu docelowego w nagłówku protokołu TCP lub UDP (w zależności od tego który jest wykorzystywany). Jeżeli wartość ta wynosi 53, a wartość portu źródłowego jest większa od 1024, wówczas pakiet jest klasyfikowany jako należący do protokołu DNS. W przypadku pakietu przychodzącego, sytuacja jest odwrotna (numer portu docelowego powinien być większy od 1024, a źródłowego wynosić 53).

6.5.3. Protokół FTP

FTP (*File Transfer Protocol*) to bardzo powszechny protokół transportowy za pomocą, którego można przesyłać pliki przez sieci komputerowe. Klienci FTP, mogą działać w dwóch trybach: aktywnym i pasywnym. W obu przypadkach ruch kontrolny odbywa się z wykorzystaniem portu 21 protokołu TCP. W trybie aktywnym przesyłanie danych odbywa się poprzez port 20 protokołu TCP.

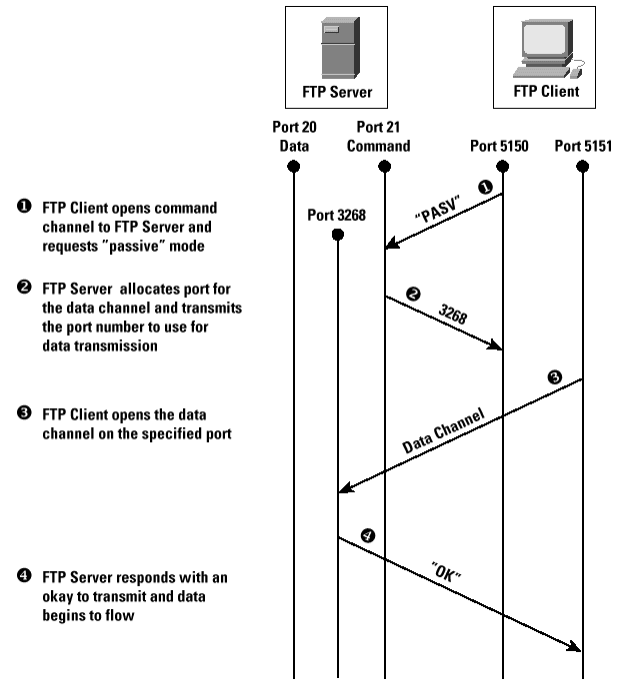
Program NBAR monitor rozpoznaje ruch kontrolny związany z protokołem FTP oraz transmisje danych odbywającą się w trybie pasywnym.

Sygnaturami służącymi do rozpoznawania protokółu FTP są:

- pakiety przesyłane są za pomocą protokołu TCP;
- pakiet wychodzący z sieci wewnętrznej (w zależności od rodzaju transmisji) ma w polu portu docelowego wartość 20 lub 21, pole źródłowe ma wartość większą od 1024;
- pakiet przychodzący - port docelowy jest większy od 1024, port źródłowy wynosi 20 lub 21.

Ze względów wydajnościowych nie zrealizowano rozpoznawania transmisji z wykorzystaniem protokołu FTP w trybie pasywnym, gdyż wymaga to stałego śledzenia sesji FTP, czyli ruchu kontrolnego realizowanego przez port o numerze 21. Wynika to z charakterystyki protokołu FTP w trybie pasywnym.

Działanie protokołu FTP przedstawia poniższy rysunek:



Rys. 7. Połączenie FTP realizowane w trybie pasywnym [11].

Po uzyskaniu połączenia z serwerem FTP (dane chce przesyłać klient) na porcie 21 pojawia się polecenie „PASV". W odpowiedzi serwer przesyła numer portu, na którym można nawiązać połączenie. Następnie klient ustanawia nową sesję TCP z użyciem przesłanego numeru portu.

W czasie działanie aplikacji klienta FTP, kończy się i nawiązuje nowe połączenia w trybie pasywnym. Aby wykryć transmisję w trybie pasywnym należy sprawdzać każdy pakiet przesyłany przez port 21 i dokonać jego inspekcji, w celu sprawdzenia czy nie została przesyłana instrukcja „PASS". Jeżeli przesłano instrukcję, wówczas należy zanalizować odpowiedź i odczytać z niej przesłany numer portu, który wykorzystywany będzie w nowym połączeniu realizującym przesyłanie danych.

Program NBAR monitor, ze względów wydajnościowych, analizuje w tym przypadku pierwsze dwa pakiety należące do konkretnego przepływu, zatem nie dokonuje śledzenia całej sesji, w konsekwencji czego nie rozpoznaje ruchu typu FTP w trypie pasywnym.

6.5.4. Protokół HTTP i HTTPS

Program klasyfikuje pakiety do protokołu HTTP, wówczas gdy:

- pakiety przesyłane są za pomocą protokołu TCP;
- dla pakietu wychodzącego numer portu docelowego równy jest 80 – HTTP, 413 – HTTPS, a numer portu źródłowego jest większy od 1024;
- dla pakietu przychodzącego numer portu docelowego większy jest od 1024, a numer portu źródłowego jest równy 80 – HTTP, 413 – HTTPS;

W.w. protokoły zostały dokładnie opisane w rozdziale 5.3.1.

6.5.5. Protokół SSH i SCP

Program klasyfikuje przepływ jako SSH lub SCP, jeżeli w danym przepływie pakiety posiadają następujące właściwości:

- pakiety transmitowane są za pomocą protokołu TCP z wykorzystaniem portu 22;
- jeżeli rozmiar pakietu jest większy od 180 B wówczas klasyfikuje go jako SCP, w przeciwnym wypadku jako SSH.

Protokoły opisane zostały dokładnie w rozdziale 5.3.2.

6.5.6. Protokół IMAP

IMAP4 (*Internet Message Access Protocol*) jest internetowym protokołem pocztowym zaprojektowanym jako następca POP3 (*Post Office Protocol*). W przeciwieństwie do POP3, IMAP pozwala na zarządzanie wieloma folderami pocztowymi oraz pobieranie i operowanie na listach znajdujących się na zdalnym serwerze. IMAP pozwala także na pobranie nagłówków wiadomości i wybranie, które z wiadomości zostaną pobrane na komputer lokalny. Zdecydowanie zmniejsza to czas połączenia oraz eliminuje konieczność wchodzenia bezpośrednio na stronę w celu usunięcia wiadomości o zbyt dużym rozmiarze [24].

Cechą charakterystyczną protokołu IMAP4, wykorzystywaną w niniejszej pracy magisterskiej, do jego identyfikacji, jest wysyłanie pakietów za pomocą protokołów TCP z wykorzystaniem portu 143, w przypadku transmisji nieszyfrowanej i portu 993 (ew. portu 585) w transmisji zaszyfrowanej.

6.5.7. Protokół POP3

POP3 *Post Office Protocol version 3* jest internetowym protokołem warstwy aplikacji pozwalającym na odbiór poczty elektronicznej ze zdalnego serwera do lokalnego komputera poprzez połączenie TCP/IP. Jego możliwości ograniczają się jedynie do pobierania i kasowania poczty z serwera [24].

Cechą charakterystyczną protokołu POP3, wykorzystywaną przez autora pracy do jego identyfikacji, jest wysyłanie pakietów za pomocą protokołów TCP z wykorzystaniem portu 110 (w przypadku transmisji nieszyfrowanej) i portu 995 w transmisji zaszyfrowanej.

6.5.8. Protokół SMTP

SMTP (*Simple Mail Transfer Protocol)* jest protokołem komunikacyjnym, który opisuje sposób wymiany (przekazywania) poczty elektronicznej, między dwoma punktami (klient-serwer) [24].

Cechą charakterystyczną protokołu SMTP, wykorzystaną przez autora pracy, jest wysyłanie pakietów za pomocą protokołów TCP z wykorzystaniem portu 25.

6.5.9. Protokół sieci BitTorrent

Protokół BitTorrent jest protokołem sieci P2P, w której centralny serwer (ang. *tracker*) nie spełnia roli wyszukującego żądane treści, ale jedynie koordynuje działania klienta i zarządza połączeniami. Użytkownik sam wyszukuje interesujące go dane, które bardzo często zapisane są na stronach WWW w postaci linków. Jeżeli użytkownik ma zainstalowany program klienta sieci BitTorent, wówczas po kliknięciu takiego linka, automatycznie uruchamiany jest program i inicjowana jest faza pobierania pliku. W przypadku tej sieci nie występuje ruch sygnalizacyjny związany z wyszukiwaniem treści. Zatem w celu rozpoznawania transmisji typu BitTorrent należy się skupić na fazie pobierania pliku [10].

Komunikacja pomiędzy klientami sieci rozpoczyna się od wymiany wiadomości uzgodnienia, której nagłówek ma postać:

```
<a character(1 byte)><a string(19 byte)>
```

Pierwszy bajt ma zawsze stałą wartość „19”. A wartość ciągu znaków ma postać *„BitTorrent protocol”*

Na podstawie tych własności dokonywana jest identyfikacja ruchu związanego z protokołem BitTorrent. Jeżeli w pierwszych 5 pakietach danego przepływu pojawiają się pakiety, które:

- są enkapsulowane w pakiety TCP;
- zaraz po nagłówku pakietu IP i TCP w pierwszym bajcie znajduje się znak „19” (heksadecymalnie 0x13);
- w 19 bajtach, występujących po znaku „19”, znajduje się ciąg znaków odpowiadający napisowi „*BitTorrent protocol”*.

6.5.10. Protokół sieci eDonkey

Jeżeli w przepływie występują pakiety których pierwszy bajt (tzw. „Marker”) (występujący zaraz po nagłówku pakietu IP i TCP (lub UDP)), ma wartość heksadecymalną równą 0xe3 (starsza wersja protokołu) lub 0xc5 (wersja nowsza), ruch sklasyfikowano jako ruch eDonkey.

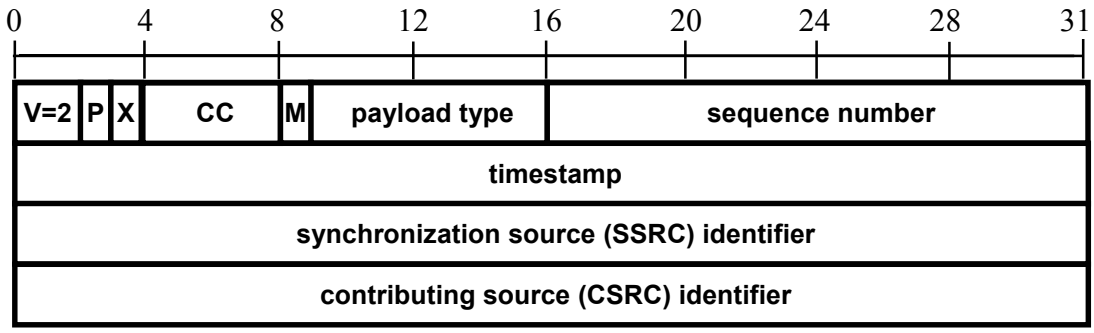
Protokół ten został dokładanie opisany w rozdziale 5.3.3.

6.5.11. Protokół RTP

RTP (*Real-time Transport Protocol)* jest protokołem transportowym, umożliwiającym dostarczanie przez Internet lub sieci intranetowe, w czasie rzeczywistym, danych wrażliwych na przerwy w transmisji (audycje radiowe lub telewizyjne, konferencje multimedialne). Jest to protokół niezbędny do obsługi wszelkich ciągłych potoków danych, interaktywnych symulacji rozproszonych oraz aplikacji służących sterowaniu i pomiarom. Usługi zapewniane przez RTP obejmują: identyfikację pakietów na podstawie zawartości (*payload-type*), numerowanie sekwencyjne, datowanie oraz śledzenie transmisji. Z uwagi na wysokie wymagania, RTP nie rozwija się aktywnie dopóki nie ma gwarancji odpowiedniej jakości transmisji. Protokół RTP działa w górnej warstwie protokołu UDP (*User Datagram Protocol*), który spełnia rolę mechanizmu przenoszenia [22].

Każdy pakiet RTP składa się z dwóch części: nagłówka pakietu i jego danych.

Budowę nagłówka RTP przedstawia poniższy rysunek.



Rys. 8. Nagłówek pakietu RTP [45].

Analizując protokół RTP i budowę pakietu zauważyć można, że każdy pakiet ma następujące własności:

- transmitowany jest za pomocą protokołu UDP;
- pierwsze dwa bity nagłówka pakietu (oznaczające wersję protokołu), które mają zawsze wartość 2;
- dla jednego przepływu, pole SSRC zawsze będzie miało taką samą wartość.

Te przedstawione powyżej cechy wykorzystane zostały w niniejszej pracy w celu identyfikacji przepływu jako RTP.

6.5.12. Protokół SIP

Autor pracy klasyfikuje ruch jako SIP, jeżeli w przepływie występują pakiety, które posiadają następujące właściwości:

- zaraz po nagłówku IP i TCP, do pierwszego znaku CRLF, występuje ciąg znaków „SIP”;

Protokół ten został dokładanie opisany w rozdziale 5.3.4.

6.5.13. Protokół RDP

RDP (*Remote Desktop Protocol)* jest protokołem stworzonym przez firmę Microsoft i wykorzystywanym do realizacji sesji terminalowych za pomocą programu

„Zdalny Pulpit”. RDP działa w sieciach o różnych topologiach, które wykorzystują różne protokoły sieciowe [32].

Transmisja za pomocą protokołu RDP jest szyfrowana, zatem jego identyfikacja odbywa się na podstawie analizy wykorzystywanych portów. Jeżeli komunikacja odbywa się poprzez sesję TCP na porcie 3389, wówczas pakiet taki klasyfikowany jest jako należący do protokołu RDP.

6.6. Testy i uwagi.

Program „NBAR monitor” przetestowany został w konfiguracji dwóch komputerów. Pierwszy z nich imitował rolę serwera, który udostępniał połączenie internetowe drugiemu komputerowi. Program dokonywał nasłuchu na interfejsie wewnętrznym pierwszego komputera i analizował ruch generowany przez drugi komputer.

Testy odbywały się również na pojedynczej maszynie, a nasłuch wykonywany był na interfejsie przez który realizowane było połączenie internetowe. W obu przypadkach główne testy działania programu, odbywały się na komputerze z procesorem AMD Athlon XP 2600+ z 512 MB pamięci RAM, pod systemem operacyjnym Windows XP z dodatkiem Service Pack 2.

Wyniki testów porównano z wynikami analiz, które wykonywał w tym samym czasie, na tym samym interfejsie, program Ethereal w wersji 0.10.11.

Program „NBAR monitor” rozpoznaje rodzaj aplikacji, która generuje ruch w sieci. Dokładnie te, których sygnatury identyfikujące zdefiniowane są w dołączanych do programu wtyczkach. Dzięki zastosowanej koncepcji podziału całego ruchu na przepływy jest to narzędzie bardzo wydajne, które jest w stanie podołać analizie ruchu z prędkością 1040 kb/s.

Ze względu na zastosowane kryterium 5 pakietów (dla portów większych od 1023), które muszą trafić do przepływu, aby został on jednoznacznie rozpoznany, program wyświetla wyniki analiz dopiero po ściągnięciu tej liczby. Założono, że ta ilość jest wystarczająca do poprawnego zidentyfikowania rodzaju protokołu w konkretnym przepływie. Jeżeli nie trafi 5 pakietów i upłynie czas 9 sekund, po których przepływ

zostanie unieważniony, wówczas dane te nie są brane do statystyk. Uznano że ilość odrzuconych pakietów jest pomijalnie mała w sytuacji kiedy analizowanych jest ok. 1200 pakietów na sekundę (przy szybkości transmisji 1040 kb/s).

Prezentowane wyniki różnią się nieznacznie od tych wykonywanych przez program Ethereal. Różnicą jest ilość pakietów, należących do przepływów do których nie trafiło 5 pakietów. Problem pojawia się również dla przepływów, które wykorzystują jeden z portów z zakresu 1-1023 (kiedy do przepływu trafi jeden pakiet).

Powyższy problem uwidacznia się w podczas analizy ruchu generowanego przez klienta sieci eDonkey, ponieważ realizowanych jest wiele prób połączeń. Skutkuje to znaczną ilością przepływów, w których ilość pakietów w jednym kierunku jest mniejsza niż 5.

Z analizy protokołu sieci BitTorent wynika, że pakiety zawierające wiadomości uzgodnienia przesyłane są na tyle rzadko, że ilość 5 analizowanych pakietów okazuje się niewystarczająca do poprawnych klasyfikacji.

Zauważyć można iż porównywanie sygnatur (identyfikujących pakiet danego protokołu), które występują na stałym miejscu jest bardzo efektywne i dużo wydajniejsze niż przeszukiwanie całego pakietu w poszukiwaniu sygnatury.

7. Podsumowanie

Mechanizmy rozpoznające typy aplikacji, na podstawie analizy generowanego ruchu, odgrywają bardzo ważną rolę w dzisiejszych sieciach komputerowych. Dostarczają one dokładnych informacji o rodzajach aplikacji oraz ilości transmitowanych danych, przez co dostarczany jest dokładny obraz wykorzystania dostępnego łącza internetowego. Informacje te umożliwiają administratorom sieci komputerowych podział ruchu sieciowego na określone klasy. Przykładem może być klasa związana z ruchem czasu rzeczywistego (wrażliwego na opóźnienia) lub z transmisjami, w których przesyłane są pliki. Dzięki temu możliwe jest zastosowanie różnorodnych algorytmów zarządzających tymi klasami, w celu optymalizacji działania sieci. Przykładem tego rodzaju mechanizmu jest wykorzystanie informacji o dostępności łącza internetowego do skonfigurowania agentów sterujących klasami QoS. Dzięki temu możliwe jest dostarczanie użytkownikom odpowiedniej jakości usług na wymaganym przez nich poziomie.

Mechanizmy identyfikujące typ aplikacji odgrywają ważną rolę w działaniu dynamicznych zapór ogniowych. Dzięki dostarczanym informacjom zapory mogą w sposób automatyczny decydować czy pojawiająca się transmisja jest zgodna z zdefiniowaną wcześniej polityką bezpieczeństwa i w zależności od tego zablokować ją.

Bardzo często analiza ruchu nie ogranicza się do protokołów warstwy transportowej, ale sięga aż do warstwy aplikacji. Jest to związane z coraz częstszym użyciem dynamicznie przydzielanych numerów portów, przez co niemożliwa jest ich identyfikacja za pomocą starszych metod. Głęboka inspekcja pakietów umożliwia również podział ruchu HTTP na mniejsze podklasy, np. pod względem zawartości fragmentu lub całości adresu URI lub typu przesyłanego pliku. Inspekcja umożliwia również podział ruchu RTP pod względem typu przesyłanych danych. Całość funkcjonalności implementuje mechanizm NBAR firmy CISCO. Inspekcja pakietów ogrywa ważną rolę w działaniu dynamicznych zapór ogniowych, ponieważ umożliwia bardzo dokładna identyfikacja ruchu, czego przykładem może być analiza ruchu przechodzącego przez port 80.

Znaczna ilość mechanizmów analizujących i rozpoznających typ aplikacji, na podstawie ruchu przez nie generowanego, opiera się na zastosowaniu idei przepływów. Definiując przepływ jako zbiór pakietów, posiadających identyczne

wartości określonych pól pakietów takich jak np.: adres źródłowy i docelowy IP, numer portu docelowego i źródłowego oraz rodzaj protokołu, nie istnieje wówczas potrzeba analizowania wszystkich pakietów przepływu. Niemal każdy pakiet z danego przepływu zawiera zduplikowane informacje i wystarczy zanalizowanie kilku, aby z dużym prawdopodobieństwem stwierdzić, iż pozostałe należą do tej samej aplikacji.

Reasumując, techniki rozpoznające typ aplikacji odgrywają bardzo ważną rolę w działaniach zmierzających do lepszego i bezpieczniejszego wykorzystania łącz internetowych. Szczególną rolę, w tym kontekście, spełnia analiza bazująca na inspekcji pakietów sięgająca informacji zawartych w warstwie aplikacji.

Bardzo dobrym przykładem aplikacji identyfikującej jest stworzony w ramach niniejszej pracy magisterskiej program „NBAR monitor", który spełnia wymogi stawiane przed nim w założeniach pracy a dzięki zastosowaniu mechanizmu przepływów jest bardzo wydajny. Umożliwia on rozpoznawanie typów aplikacji spośród transmisji o łącznej prędkości dochodzącej do 1 Mb/s. Na szczególną uwagę zasługuje również możliwość rozszerzania działania programu o wykrywanie nowych protokołów, co osiąga się przez tworzenie i dodawanie nowych wtyczek. Dzięki takiemu rozwiązaniu program „NBAR monitor" może służyć również jako dobre narzędzie do nauki metod analiz ruchu w sieciach IP.

8. Literatura

1. M. Czajkowski, „Encyklopedia Nowych Technologii", Edition 2000, 2002;
2. PN-I-13335-1:1999 „Technika informatyczna - Wytyczne do zarządzania bezpieczeństwem systemów informatycznych - Pojęcia i modele bezpieczeństwa systemów informatycznych", Polski Komitet Normalizacyjny, 1999;
3. K J. Woźniak, K. Nowicki, „Społeczeństwo Globalnej Informacji - Sieci LAN, MAN i WAN - protokoły komunikacyjne" Katedra telekomunikacji AGH Kraków, 2000;
4. P. Brudno, T. Ratajczak, „Problemy bezpieczeństwa systemów informatycznych organizacji" Politechnika Gdańska, 2004, URI: http://www.gazeta-it.p;
5. J. Bryl, „Bezpieczeństwo systemów e-Commerce", GazetaIT nr 3(33), 2005, URI: http://www.gazeta-it.p;
6. D. DiNunno, „IT Performance Engineering & Measurement Strategies: Quantifying Performance Loss" METAGroup, November 2000,
   URI: http://www.metagroup.com;
7. R. Stankiewicz, A. Jajszczyk, „Sposoby zapewnienia gwarantowanej jakości usług w sieciach IP", Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne, Nr 2/2002;
8. R. Wciseł, „Gwarantowana jakość usług", 2005,
   URI: www.brzeska.art.pl;
9. Y. Fei, J. Jones, K. Lakkas, Y. Zheng, „Measurement of the Usage of Several Secure Internet Protocols from Internet Traces" UCSD Department of Computer Science and Enggineering, 2005;
10. S. Sen, O. Spatscheck, D. Wang, „Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures", AT&T Labs-Research, 2004, URI: http://www.www2004.org/proceedings/docs/contents.htm;
11. T. M. Thomas, „One Byte at a Time: Is Your FTP Active or Passive?", Cisco Systems. Inc., URI: www.cisco.com;
12. M. Khanna, „Plug-In framework using DLLs", The Code Project, czerwiec 2002, URI: http://www.codeproject.com/dll/plugin.asp;
13. „Protokół SIP w sesjach multimedialnych" NetWorld, grudzień 2002, URI: http://www.networld.pl/;

14. „Network-Based Application Recognition and Distributed Network-Based Application Recognition”, Cisco Systems Inc., URI: http://cisco.com;
15. H. Chung, „Wise-TrafView, Flow-based Measurement and Analysis System”, ETRI, styczeń 2004;
16. R. Sharpe, E. Warnicke, U. Lamping, “Ethereal User's Guide”, URI: http://www.ethereal.com/docs/;
17. P. Dzierżak, „Wtyczki, rodzaje zasada działanie oraz tworzenie”, URI: http://sphere.pl/~przemek;
18. T. Giełdziński, Usługi wymiany plików typu peer to peer, Praca dyplomowa magisterska, Politechnika Łódzka, wrzesień 2004;
19. M. Karcewicz, „Quality of Service w sieciach LAN/WAN”, Politechnika Śląska, Wydział Organizacji i Zarządzania, 2005;
20. J. Turski, „Gwarantowana jakość usług (QoS) w sieciach IP”, Praca dyplomowa, Politechnika Łódzka, 2003;
21. M. Morawski, Materiały do wykładu „Technologie Sieci komputerowych”, Politechnika Łódzka, 2004;
22. Wydawnictwo Robomatic, Encyklopedia, URI: http://www.robomatic.pl/;
23. Neteopedia, WebStyle Systems, URI: http://www.ws-webstyle.com/cms.php/en/netopedia;
24. Wikipedia Wolna encyklopedia, URI: http://pl.wikipedia.org/;
25. Strona domowa programu BitTorrent, URI: http://www.bittorrent.com/;
26. Strona domowa firmy CISCO, URI: http://cisco.com;
27. Strona domowa programu DC++, URI: http://dcplusplus.sourceforge.net/;
28. Strona domowa programu eMule, URI: http://www.emule-project.net/;
29. Strona domowa poświęcona programowi Ethereal, Wprowadzenie, URI: http://www.ethereal.com/introduction.html;
30. Internet Assigned Numbers Authority (IANA), URI: http://www.iana.org/assignments/port-numbers;

31. ICANN - Internet Corporation For Assigned Names and Numbers, http://www.icann.org/;
32. Remote Desktop Protocol, MSDN,
URI: http://msdn.microsoft.com/;
33. Barrett, Silverman, Byrnes/o'Reilly, „SSH: The Secure Shell",
URI: http://www.snailbook.com/protocols.html;
34. A. Bhushan, „A File Transfer Protocol", Network Working Group, kwiecień 1971, RFC 114, URI: http://www.rfc-editor.org/rfc/rfc114.txt;
35. J. Postel, „User Datagram Protocol", Information Sciences Institute, sierpień 1980, RFC 768, URI: http://www.rfc-editor.org/rfc/rfc768.txt;
36. Darpa Internet Program, „Internet Protocol", (Specyfikacja protokołu IP), wrzesień 1981, RFC 791, URI: http://www.rfc-editor.org/rfc/rfc791.txt;
37. Darpa Internet Program, „Transmission Control Protocol", (Pierwsza specyfikacja protokołu TCP), sierpień 1981, RFC 793;
URI: http://www.rfc-editor.org/rfc/rfc793.txt
38. J. B. Postel, „Simple Mail Transfer Protocol", Information Sciences Institute University of Southern California, sierpień 1982, RFC 821
URI: http://www.rfc-editor.org/rfc/rfc821.txt;
39. J. Myers, M. Rose, „Post Office Protocol - Version 3", Network Working Group, maj 1996, RFC 1939, URI: http://www.rfc-editor.org/rfc/rfc1939.txt;
40. T. Berners-Lee, R. Fielding, H. Frystyk, "Hypertext Transfer Protocol - HTTP/1.0", Network Working Group, maj 1996, RFC 1945,
URI: http://www.rfc-editor.org/rfc/rfc1945.txt;
41. Ed. R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin; „Resource ReSerVation Protocol (RSVP) – Version 1" Specyfikacja funkcjonalna, wrzesień 1997, RFC 2205, URI: http://www.rfc-editor.org/rfc/rfc2205.txt;
42. E. Rescorla, A. Schiffman, "The Secure HyperText Transfer Protocol", Network Working Group, sierpień 1999, RFC 2660,
URI: http://www.rfc-editor.org/rfc/rfc2660.txt;
43. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, „SIP: Session Initiation Protocol", Network Working Group, czerwiec, 2002, RFC 3261, URI: http://www.rfc-editor.org/rfc/rfc3261.txt;
44. M. Crispin, „Internet Message Access Protocol - version 4rev1", Network Working Group, marzec 2003, RFC 3501, URI: http://www.rfc-editor.org/rfc/rfc3501.txt;

45. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, „RTP: A Transport Protocol for Real-Time Applications", Network Working Group, lipiec 2003, RFC 3550, URI: http://www.rfc-editor.org/rfc/rfc3550.txt;

9. Załącznik 1 Instrukcja użytkownika programu „NBAR monitor”

9.1. Przeznaczenie programu

Program „NBAR monitor” utworzony został na potrzeby pracy magisterskiej „Zapory ogniowe typu NBAR”. Jego głównym zadaniem jest analiza ruchu przechodzącego przez interfejs, na którym odbywa się nasłuch, w celu zidentyfikowania protokołu aplikacji generującej ten ruch. Obsługiwanym typem medium jest Ethernet. Program dodatkowo wyświetla informacje o ilości i rozmiarze zidentyfikowanych pakietów protokołów. Wyniki pokazywane są sumarycznie dla wszystkich sesji w danym dniu, a pamiętane do końca miesiąca.

9.2. Wymagania

Podstawowym wymogiem stawianym przed użytkownikiem chcącym korzystać z programu „NBAR monitor” jest posiadanie przez niego komputera klasy minimum Pentium 2 z procesorem 800Mz oraz z 128 MB pamięci. Sam program zajmuje mniej niż 1 MB.

Na komputerze musi być też zainstalowany systemem operacyjnym Microsoft Windows XP. Program wymaga zainstalowanie biblioteki WinPcap 3.0 oraz bibliotek mfc70.dll i msvcr70.dll.

9.3. Instalacja

W celu zainstalowania programu „NBAR monitor” należy wykonać niżej wymienione czynności:

- zainstalowanie biblioteki WinPcap w wersji 3.0, wersja instalacyjna znajduje się na dysku CD-ROM w katalogu: Cz praktyczna\Sterowniki\, plik o nazwie „WinPcap_3_0.exe”;
- przekopiowanie plików „mfc70.dll”, „msvcr70.dll” do katalogu: Windows\System32;

- przekopiowanie katalogu NBAR monitor z katalogu: Cz praktyczna w dowolne miejsce na dysku.

Katalog „NBAR monitor” posiada następujące podkatalogi:

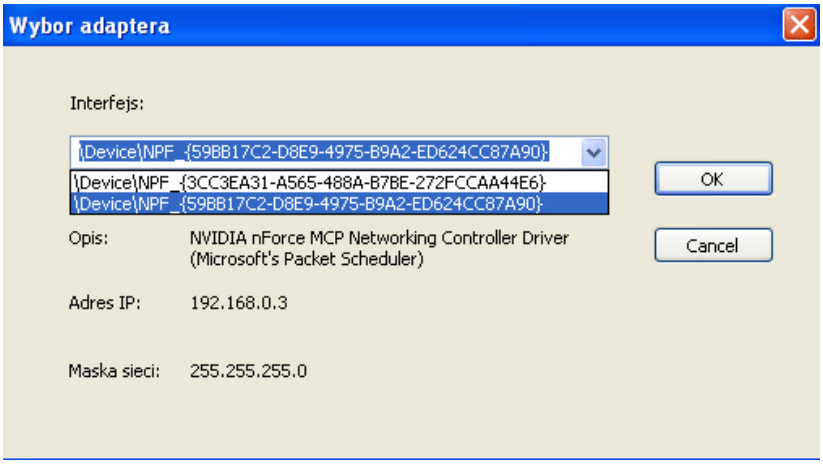
- „\Run\” – zawiera plik z programem „NBAR monitor.exe”, nazwa podkatalogu może być dowolna;
- „\Plug-Ins\” – zawiera wtyczki dostarczone wraz z programem;
- „\Config\” – zapisywany jest w nim plik konfiguracyjny „config.cfg”.

Powyższa struktura katalogów musi być zachowana, aby program działał poprawnie.

Program uruchamia się poprzez plik wykonywalny „NBAR monitor.exe” znajdujący się w katalogu „\Run\”.

9.4. Opis interfejsu

Przy pierwszym uruchomieniu programu, pojawia się okno wyboru adaptera (Rys.nr 9), na którym dokonywany będzie później nasłuch i analiza.



Rysunek 9. Okno wyboru adaptera

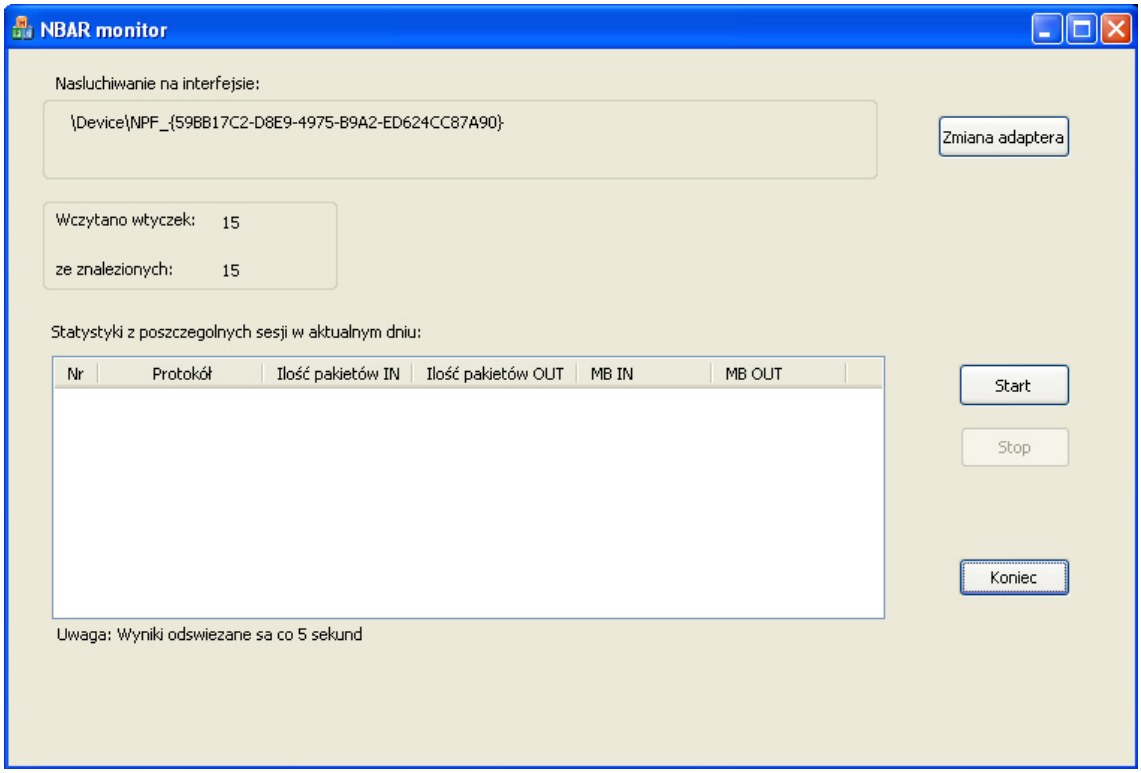
Klikając na listę rozwijaną pojawiają się nazwy aktywnych interfejsów. Po wybraniu któregoś z nich pojawia się jego opis oraz adres IP i maska sieci, jakie są skonfigurowane na danym interfejsie.

Po naciśnięciu przycisku „OK” nazwa interfejsu zapisywana jest do pliku „config.cfg” w katalogu „Config”, po czym program uruchamiany jest dalej.

Jeżeli naciśnięty zostanie przycisk „Cancel” wówczas pojawi się komunikat "Nie wybrano żadnego adaptera, program zakończy swoje działanie!!!" wówczas działanie programu zostanie przerwane.

Po wyborze adaptera program wyszukuje wtyczki i próbuje je wczytać. Wtyczki podzielone są na dwie grupy, z przedrostkiem „p_” i „i_”, jeżeli nie uda się wczytać wtyczek z któreś z grup pojawia się odpowiednio komunikaty „Nie wczytano żadnej wtyczki z rodziny P” i Nie wczytano żadnej wtyczki z rodziny I”.

Niezależnie od ilości wczytanych wyczek ukazuje się okno główne aplikacji, które przedstawia się następująco:



Rysunek 10. Okno główne aplikacji „NBAR monitor”

W sekcji u góry po lewej stronie zawarte są informację o aktualnie wybranym interfejsie. Obok niego znajduje się przycisk „Zmiana adaptera”, którego naciśnięcie powoduje pojawienie się okna przedstawionego na rysunku 9, w którym można dokonać zmiany adaptera. W tym wypadku, po naciśnięciu przycisku „Cancel” pojawia się komunikat „Nie zmieniono wyboru adaptera”.

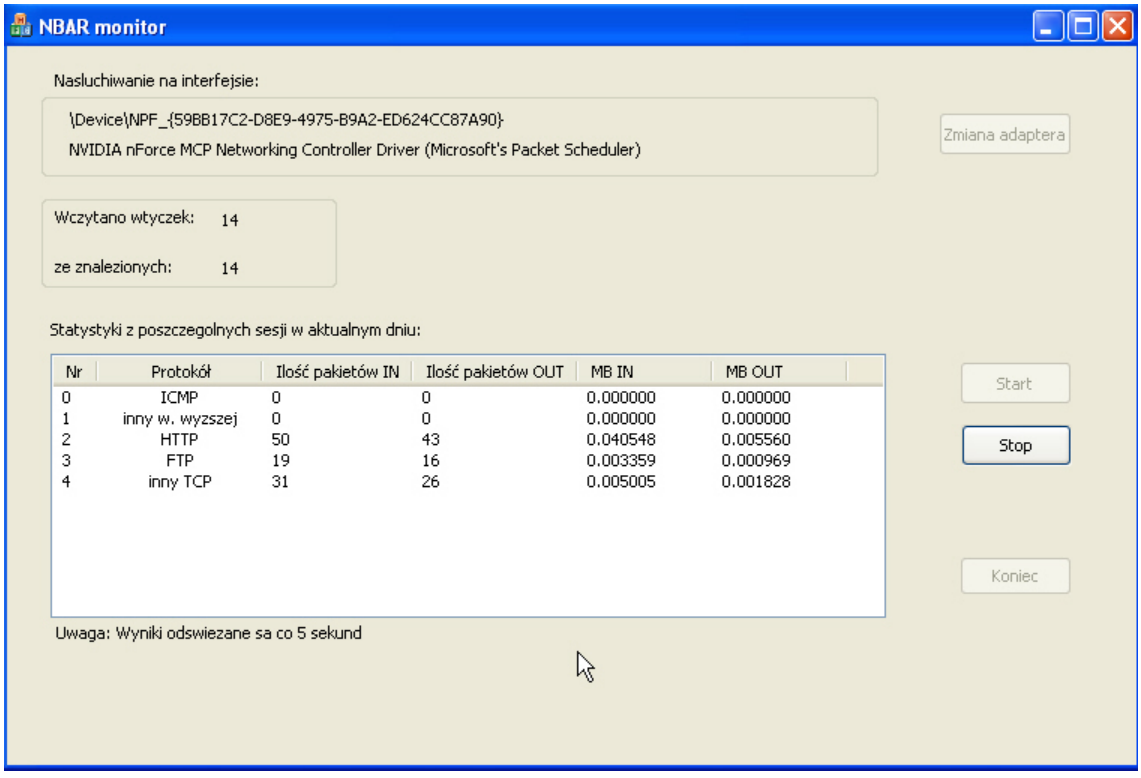
W następnej sekcji wyświetlona jest liczba znalezionych i wczytanych wtyczek.

W środkowej części w pokazywane są statystki z poszczególnych sesji w danym dniu. Informację te zawierają: nazwę rozpoznanego protokołu, ilość oraz rozmiar pakietów przychodzących i wychodzących w ramach konkretnego protokołu.

Obok tych statystyk znajdują się przyciski spełniające następujące role:

- Start – uruchamia proces rozpoznawania protokołów i wyświetlania wyników. Po jego wybraniu niemożliwe staje się naciśnięcie przycisków: „Zmiana adaptera", „Koniec" i „Start";
- Stop – kończy przedstawiony powyżej proces. Po jego naciśnięciu trzeba odczekać parę sekund, aby wątki poprawnie się zakończyły i wyświetliły wyniki;
- Koniec – kończy działanie programu, po uprzednim zwolnieniu wczytanych bibliotek.

Poniższy rysunek przedstawi wygląd okna programu po uruchomieniu procesu analizująco-wysświetlającego oraz przykładowe wyniki działania.



Rysunek 11. Wyniki działania programu „NBAR monitor"

Program standardowo w pierwszych dwóch pozycjach pokazuje ilość rozpoznanych pakietów ICMP i pakietów IP, które używają innego protokołu warstwy

wyższej od ICMP, TCP i UDP – „inny w. wyzszej”. Jeżeli nie zidentyfikowane protokołu, a pakiet transportowany jest za pomocą TCP lub UDP, wówczas statystki wyświetlane są pod nazwą „inny TCP” lub „inny UDP”. W sytuacji, kiedy nie zostanie wczytana, żadna wtyczka klasyfikowanie przepływów odbywać się będzie tylko na powyższe cztery grupy.