



**Politechnika Śląska**

**Instytut informatyki**

**Wydział Automatyki, Elektroniki i Informatyki**

**44-100 Gliwice, ul. Akademicka 16**

**PRACA MAGISTERSKA**

# **Analiza skuteczności ataków DDoS na najpopularniejsze współczesne systemy operacyjne**

Konrad Malewski

[kmalewski\\_at\\_gmail\\_dot\\_com](mailto:kmalewski@gmail.com)

Promotor: prof. dr hab. inż. Tadeusz Czachórski

---

GLIWICE 2005

Copyright (c) 2005 Konrad Malewski  
Udziela się zezwolenia do kopiowania rozpowszechniania i/lub  
modyfikację tego dokumentu zgodnie z zasadami Licencji GNU Wolnej  
Dokumentacji w wersji 1.1 lub dowolnej późniejszej opublikowanej przez  
Free Software Foundation; wraz z całą zawartością.  
Kopia licencji załączona jest w sekcji zatytułowanej "GNU Free  
Documentation License" lub pod adresem  
<http://www.fsf.org/licensing/licenses/fdl.txt>

1.	Wstęp.....	4
2.	Protokoły sieciowe .....	6
2.1	ISO OSI .....	6
2.2	Łącze danych – warstwa Ethernetu .....	8
2.3	Protokół IPv4.....	8
2.4	Protokół IPv6.....	11
2.5	Protokół TCP.....	13
2.6	Protokół UDP .....	16
2.7	Protokół ICMP .....	17
2.8	Fragmentacja danych w sieciach IPv4 oraz IPv6.....	19
2.8.1	Kontrola Fragmentacji w IPv4 .....	19
2.8.2	Kontrola Fragmentacji w IPv6 .....	21
3.	Luki, ataki sieciowe.....	24
3.1	Klasyfikacja luk w systemie.....	24
3.2	Kim jest potencjalny włamywacz?.....	25
3.3	Jakie są cele włamywaczy? .....	25
3.4	Modele ataków .....	27
3.5	Przyczyny ataków .....	27
3.6	Klasyfikacja ataków DoS .....	29
3.3.1	Ataki zużywające zasoby sieciowe .....	29
3.3.2	Ataki wykorzystujące domyślne zachowanie protokołu .....	32
3.3.3	Ataki wynikające z błędu w oprogramowaniu .....	33
3.7	Metody ukrywania ataku .....	34
4.	Jak wzmocnić stos TCP/IP .....	35
4.1	Wzmacnianie stosu w systemach Windows .....	35
4.2	Wzmacnianie stosu w systemie Linux .....	37
5.	Programy wykonane w ramach pracy dyplomowej .....	39
5.1	Program „DDoS Generator” .....	39
5.1.1	Jakie możliwości daje program? .....	39
5.1.2	Biblioteki wykorzystane do napisania programu .....	40
5.1.3	Instrukcja użytkownika .....	44
5.1.4	Przykłady użycia programu w rzeczywistym środowisku .....	48
5.2	Program mierzący czas połączenia – ConnectTime .....	50
5.3	Przykładowa usługa sieciowa IPv4/IPv6 – SocketListener .....	51
5.4	Programy mierzący obciążenie procesora – WinProcTime oraz Load .....	51
6.	Analiza wpływu ataków DOS/DDOS na popularne systemy operacyjne.....	52
6.1	Opis stanowiska wykorzystanego przy implementacji ataku.....	52
6.2	Opis ataku i pobierania wyników .....	53
6.3	Wyniki .....	54
6.3.1	System „Windows XP” .....	54
6.3.1.1	System niezabezpieczony – wersja ze zintegrowanym SP1 .....	54
6.3.1.2	System ze wszystkimi uaktualnieniami do dnia 20 maja 2005 .....	59
6.3.2	System „Windows 2003” .....	64
6.3.2.1	System niezabezpieczony .....	64

6.3.2.2	System zabezpieczony uaktualnieniami do 20 maja 2005 .....	68
6.3.3	System „Windows Longhorn build 5048” .....	70
6.3.4	System „Linux – debian” .....	72
6.3.5	Wpływ obecności ściany ogniowej na skuteczność ataku DDoS .....	74
6.3.5.1	Agnitum Outpost Firewall PRO – wersja 2.6.452.403 .....	74
6.3.5.2	Sygate Personal Firewall – wersja 5.5.2710.0 .....	76
6.3.5.3	Zone alarm – wersja 5.5.94.0 .....	78
7.	Podsumowanie.....	80
8.	Bibliografia.....	81
9.	Dodatki .....	82
	GNU Free Documentation License .....	83

*Gdy ktoś umie atakować, sprawi, że wróg nie będzie wiedział,  
gdzie wystawić obronę. Gdy ktoś potrafi się bronić, sprawi,  
że przeciwnik nie będzie wiedział gdzie atakować.*

- Sun Tzu, *Sztuka Wojny*

## 1. Wstęp

Obserwując rozwój sieci Internet w ciągu ostatnich lat, trudno nie dostrzec, że stała się ona dla większości źródłem nie tylko wiedzy, ale również narzędziem pracy czy sposobem na wypoczynek. Dzięki Internetowi czas komunikacji pomiędzy ludźmi skrócił się do ułamków sekund, a prezentowane w sieci materiały często odzwierciedlają niesamowity przekrój różnych kultur i myśli. Obserwując tę różnorodność, nie sposób nie zgodzić się z Marshalllem McLuhanem, który twierdził, że masowe media obalają bariery czasowe i przestrzenne, umożliwiając ludziom komunikację na masową skalę. W tym sensie Ziemia staje się wioską, a każdy użytkownik - sąsiadem. Idea globalnej wioski, w której wszyscy użytkownicy dzieliliby się myślą, współuczestniczyli w tworzeniu przyszłości - jest piękna, lecz okazała się niestety jedynie futurologiczną mrzonką.

Jak w każdym środowisku, tak i w rodzinie internautów, istnieją idealiści-marzyciele i ludzie twardo stąpający po ziemi; współlistnieją ludzie pracujący dla dobra ogółu oraz cyber-przestępcy. Ci ostatni, kierowani różnymi motywami, niszczą idylliczną wizję „Globalnej wioski”, wywołują strach wśród użytkowników i niszczą efekty wysiłku intelektualnego pracowitych mieszkańców „wioski”.

Trzeba uświadomić sobie, że zagrożenie ze strony wszelkiego rodzaju przestępców komputerowych istnieje pomimo wysiłków wielu znakomitych specjalistów, narzędzia służące czynieniu zniszczeń są coraz prostsze w użytkowaniu, a efekty ich wykorzystania przynoszą coraz większe straty.

Celem tej pracy jest zbadanie odporności najpopularniejszych systemów operacyjnych na ataki DDoS (ang. Distributed Denial of Service). W ramach pracy powstał program *DDoS Generator* oraz programy pomocnicze (opisane w rozdziale 5) za pomocą których możliwe było przeprowadzenie testów oraz akwizycja wyników. Eksperymenty były przeprowadzane na niezabezpieczonych systemach (w domyślnych konfiguracjach) oraz po przeprowadzeniu uaktualnień i dodatkowych modyfikacji „wzmacniających” stos TCP/IP. Przeprowadzone badania miały określić, jakich problemów należy oczekiwać przy podłączaniu serwera do sieci, oraz jakie są przesłanki, które prowadzą do stwierdzenia, że serwer jest ofiarą ataku.

Dodatkowo został przeprowadzony eksperyment, którego celem było określenie wpływu korzystania ze ściany ogniowej (ang. *firewall*), na skutki ataków DDoS.

Podczas przeprowadzania badań wykryto szereg nieprawidłowości w działaniu stosów TCP/IP, a jeden z błędów został nawet zgłoszony do CVE (ang. *Common Vulnerabilities and Exposures*) i została mu przydzielona sygnatura CAN-2005-1649.

Praca została podzielona na szereg rozdziałów:

Rozdział 2 – Opisuje model ISO/OSI wraz z poszczególnymi warstwami oraz wskazuje na potencjalne cele ataku.

Rozdział 3 – Charakterystyka luk w oprogramowaniu, włamywaczy. Stara się nakreślić postać włamywacza oraz zrozumieć sposób jego myślenia.

Rozdział 4 – Opisuje metody zabezpieczania stosów sieciowych.

Rozdział 5 – Opisuje programy, które zostały wykonane w ramach niniejszej pracy.

Rozdział 6 – Zawiera testy skuteczności ataków DDoS wykonane za pomocą programów. opisanych w rozdziale 5.

Rozdział 7 – Podsumowanie.

Rozdział 8 – Spis ilustracji i rysunków.

Rozdział 9 – Bibliografia.

Rozdział 10 – Dodatki.

## 2. Protokoły sieciowe

### 2.1 ISO OSI

Każde urządzenie w sieci, chcące wymieniać dane z innym urządzeniem, ma do czynienia z modelem ISO OSI (ang. *International Standard Organization - Open System Interconnect*), który jest modelem odniesienia. Model ten określa, jak ma wyglądać ogólny schemat wymiany danych pomiędzy poszczególnymi protokołami. Poniższy rysunek pokazuje, jakie warstwy znajdują się OSI, a co za tym idzie - na jakie abstrakcyjne części da się podzielić każdą przesyłaną w sieci wiadomość.

7. Aplikacji
6. Prezentacji
5. Sesji
4. Transportowa
3. Sieciowa
2. Łączy danych
1. Fizyczna

**Tabela 1. Warstwy ISO/OSI.**

O ile model ISO/OSI upraszcza schemat komunikacji pomiędzy obiektami w modelu sieci, dając wytyczne, co do roli poszczególnych protokołów i uniemożliwiając stworzenie monolitycznego rozwiązania dedykowanego, którego zastosowanie utrudniłoby wymianę danych pomiędzy różnymi architekturami, o tyle jego zastosowanie dostarcza atakującemu więcej celów, z których zaatakowanie któregośkolwiek powoduje destabilizację całego mechanizmu komunikacyjnego.

Pierwsza warstwa – fizyczna – jest odpowiedzialna za transmisję bitów między węzłami sieci. Określa fizyczną postać sygnałów elektrycznych w medium transmisyjnym – natężenie, amplitudę czy inne parametry elektryczno-fizyczne (np. maksymalną odległość pomiędzy węzłami, sposób dekodowania sygnałów itp.).

Warstwa łączy danych zajmuje się przekazywaniem danych pomiędzy sąsiednimi węzłami sieci. Wadą niektórych łączy fizycznych jest ich podatność na zakłócenia natury elektrycznej. Aby zapewnić poprawność przesyłanych danych, warstwa ta powinna zawierać mechanizmy

zapewniające poprawność transmitowanych danych w medium – sumę CRC (ang. *Cyclic Redundancy Check*).

Trzecia warstwa – sieciowa – zapewnia transport bloków danych, bądź to z węzła do węzła, bądź z sieci do sieci. W tym drugim przypadku warstwa ta powinna zapewnić transport danych po optymalnej trasie uwzględniając nasycenia łączy pośrednich (ang. *routing*) oraz odpowiednie dopasowanie wielkości bloku danych (zwanego dalej pakietem) do możliwości medium, czyli fragmentację.

Za bezbłędne dostarczenie pakietów pomiędzy węzłami końcowymi w sieciach odpowiedzialna jest warstwa transportowa. Zapewnia ona usługi połączeniowe pomiędzy zdalnymi systemami, zapewniając jednoznaczne przyporządkowanie każdego pakietu do działającej w systemie aplikacji. W celu zmaksymalizowania przepustowości łącza stosuje odpowiednią strategię zarządzania retransmisjami.

W warstwę sesji wbudowane są mechanizmy służące zapewnieniu uporządkowanej wymianie danych pomiędzy połączonymi segmentami warstwy prezentacji oraz umożliwieniu określenia stanu konwersacji. Jako przykład można podać sesje http.

Warstwa prezentacji ukrywa wewnętrzną architekturę systemu przed warstwą aplikacji. W tej warstwie dokonuje się ewentualne szyfrowanie, deszyfrowanie, kompresja lub dekompresja danych. Dokonywana jest konwersja danych z postaci sieciowej na postać charakterystyczną dla architektury.

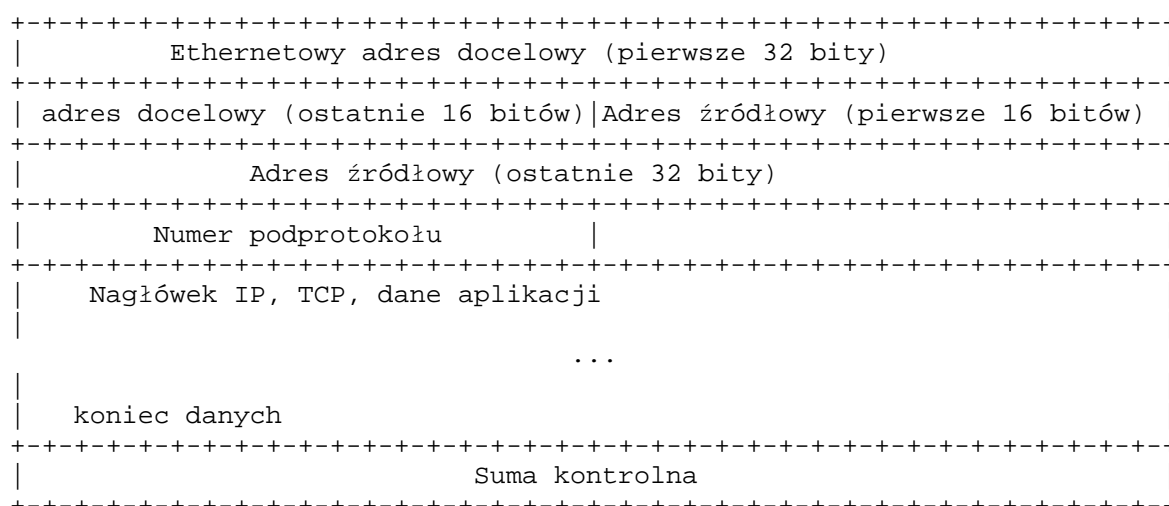
Ostatnia warstwa – aplikacji – umożliwia dostęp poszczególnych programów do środowiska ISO/OSI. Sprawia, że działający w systemie proces nie musi się odwoływać do sprzętu przy próbie komunikacji, lecz dostaje zunifikowany interfejs, za pomocą którego dokonuje konwersacji.

Patrząc na model ISO/OSI oczami atakującego, nie trudno dostrzec, że podział wiadomości na warstwy dostarcza celów ataku. Ataki mogą być skierowane na dowolną z warstw niezależnie, co może doprowadzić do zatrzymania bądź uniemożliwienia transmisji. O ile ataki na warstwę fizyczną wymagają ingerencji w sprzęt, a sam atak może dotyczyć wyłącznie jednego produktu (np. tylko określonych „hubów”), o tyle ataki na warstwy powyżej sieciowej mogą mieć charakter niezależny sprzętowo, co przy założeniu uniwersalności ataku czyni go bardzo groźnym z punktu widzenia bezpieczeństwa.



## 2.2 Łączy danych – warstwa Ethernetu

Warstwa ta umożliwia komunikację pomiędzy sąsiednimi węzłami w sieci LAN. Zamieszczona na końcu nagłówka suma kontrolna, której wyliczaniem zajmuje się karta sieciowa, gwarantuje poprawne dostarczenie pakietu do urządzenia docelowego.



Rysunek 1. Pola ramki ethernetowej.

Przed polem adresu docelowego znajduje się preambuła o długości 8 bajtów, która służy do synchronizacji kart sieciowych odbiorcy i nadawcy.

Atakujący, który posiada możliwość manipulowania danymi warstwy ethernetowej, uzyskuje nieograniczone możliwości kształtowania ruchu w sieci. Dodatkowo może on fałszować adres źródłowy, myląc ofiarę i podszywając się pod inny komputer w sieci.

Wykrycie intruza w takim przypadku jest zadaniem niełatwym i wymagającym sprawdzenia wszystkich możliwych fizycznych połączeń do sieci oraz medium transmisyjnego – jako że atakujący może znajdować się w obrębie sieci LAN. Innym rozwiązaniem, o wiele skuteczniejszym, może być zakup zaawansowanych *switchy*, do których należy wprowadzić informację o podłączonych końcówkach. Tak ustawione urządzenia nie pozwalają na transmisję pakietu, którego adres źródłowy nie znajduje się na liście dostępu dla danego portu urządzenia.

## 2.3 Protokół IPv4

Najczęściej wykorzystywanym protokołem transmisyjnym w sieci Internet jest protokół IP w wersji czwartej. Z tego właśnie względu może stać się obiektem ataku odmowy usługi.

Protokół ten został ustandaryzowany przez określający jego funkcjonalność dokument RFC numer 791.

Do charakterystycznych cech protokołu IP należy w pierwszym rzędzie zaliczyć to, że jest on bezpołączeniowy oraz nie zapewnia poprawności przesyłanych danych. Beipołączeniowość oznacza, iż nie zapewnia on przyporządkowania danych do konwersacji. Brak poprawności danych wywodzi się z tego, że suma kontrolna zawarta w nagłówku dotyczy jedynie samego nagłówka. System przetwarzając pakiet IP wie od jakiego systemu nadeszły dane, ale nie wie, czy są one poprawne. Jeżeli nagłówek IP zostanie uszkodzony, to system przetwarzający taki pakiet odrzuci go bez dokonywania analizy.

Inną ważną cechą IP jest możliwość przesyłania danych we fragmentach. Cecha ta jest przydatna w sieciach, które różnią się maksymalnymi rozmiarami pakietu w warstwie łącza danych. Dane IP mogą być pofragmentowane minimalnie na paczki o rozmiarze 8 bajtów.



**Rysunek 2. Format nagłówka IPv4.**

*Wersja* – 4 bity – pole określające wersję nagłówka.

*IHL* – 4 bity – określa długość nagłówka. Wartość tutaj wpisana określa liczbę podwójnych słów.

*TOS* – 8 bitów – określa, jaką strategię przyjąć przy przesyłaniu pakietu. Umożliwia na przykład zmuszenie routera, do jak najszybszego przesłania pakietu przez kanał lub z najmniejszą liczbą przeskoków, czy też wybór łącza o najlepszej niezawodności.

*Całkowita długość* – 16 bitów – wskazuje na długość pakietu, mierzoną w oktetach. Przez całkowitą długość rozumie się rozmiar nagłówka IP oraz wielkość danych. Pole to umożliwia ustawienie maksymalnej wielkości pakietu na 65535, co w praktyce jest rzadko wykorzystywane. Atakujący wysyłając pakiety IP o dużych rozmiarach może zakłócić działanie sieci poprzez zużycie całego dostępnego pasma. W praktyce

sugeruje się, aby każde urządzenie sieciowe było w stanie odebrać pakiet o długości co najmniej 576 oktetów.

*Identyfikator* – 16 bitów – pole to jest wykorzystywane przy składaniu pofragmentowanych pakietów. Dzięki niemu system operacyjny dokonuje przyporządkowania fragmentu do oryginalnej wiadomości.

*Flagi* – 3 bity – pole zawierające przełączniki sterujące fragmentacją pakietu.

Bit 0: zarezerwowany – musi być równy 0

Bit 1: (DF – ang. *Dont fragment*) 0 = Fragmentacja jest możliwa, 1 = Fragmentacja zabroniona.

Bit 2: (MF – ang. *More fragments*) 0 = Ostatni fragment, 1 = Fragmenty w drodze.

*Przesunięcie fragmentu* – 13 bitów – określa pozycję aktualnego fragmentu w pakiecie.

Przesunięcie jest liczone w porcjach po 8 bajtów, także maksymalna wielkość przesunięcia to  $2^{13} \cdot 8$  bajtów czyli 65536 bajtów.

*TTL* – 8 bitów – określa maksymalny czas, który pakiet IP może spędzić w sieci Internet.

Czas jest mierzony w sekundach. Jeśli jednak w jakimś systemie przetwarzającym pakiet przebywał krócej niż sekundę, to pole TTL jest zmniejszane o 1. W przypadku gry wartość tego pola w którymkolwiek momencie spadnie do 0, to pakiet jest niszczone, a do nadawcy wysyłany jest odpowiedni komunikat informacyjny protokołu ICMP.

*Podprotokół* – 8 bitów – pole określa protokół wyższej warstwy.

*Suma kontrolna nagłówka* – 16 bitów – zapewnia poprawność danych zawartych tylko w nagłówku. Jest ona weryfikowana i przeliczana za każdym razem, gdy pakiet jest przetwarzany przez urządzenie sieciowe.

*Adres źródłowy i adres docelowy* – 32 bity każde z pól – określają adresy nadawcy i odbiorcy.

Wiele cech protokołu IP w wersji 4 sprawia, że jest on obiektem zainteresowania hakerów. Największą jego wadą jest to, iż nie zapewnia on autentyczności przesyłanych danych. Każdy intruz, posiadający odpowiedni dostęp do infrastruktury sieciowej, może fałszować pakiety, wprowadzając „cel” w błąd. Protokół ten jest protokołem zawodnym, co można traktować zarówno jako wadę, jak i zaletę. Zaletą jest osiągana przez ten fakt prostota protokołu, co wpływa na wydajność przetwarzania pakietów. Dodatkowo, jeśli uwzględnimy możliwość fragmentacji danych oraz możliwość zamienienia kolejności pakietów, to stanowi to kolejną metodę ataku sieciowego (np. zmuszenie zdalnego komputera do zaalokowania dużej ilości pamięci czy też zmuszenie komputera do próby scalenia pakietu z fragmentów).

## 2.4 Protokół IPv6

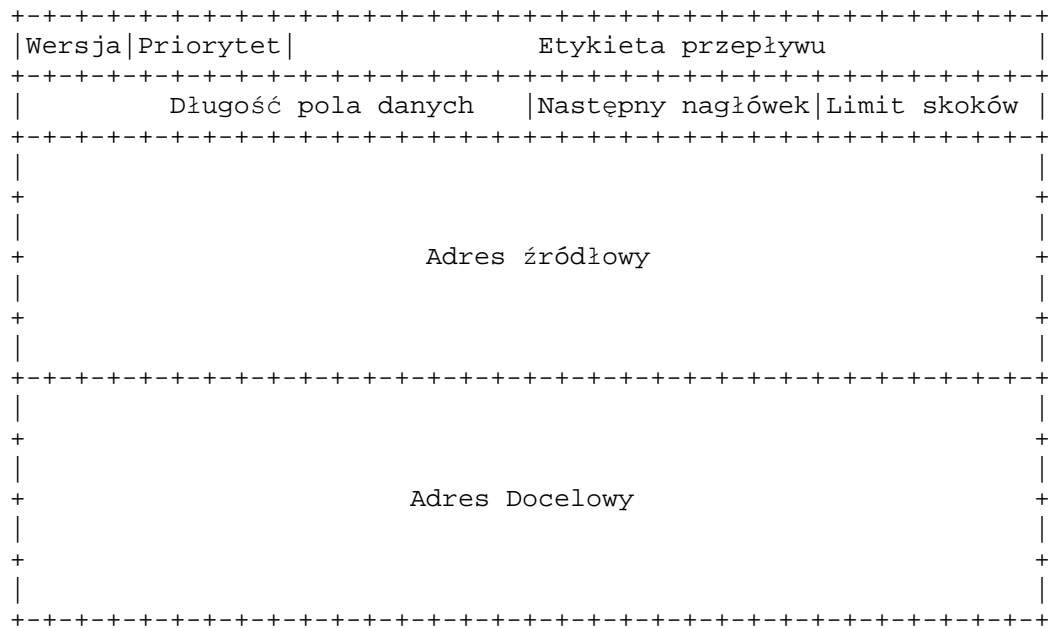
Protokół IPv6 znajduje się w tej samej warstwie ISO/OSI co IPv4. Zajmuje się tym samym, czym zajmuje się protokół IPv4, a więc przekazywaniem danych pomiędzy węzłami.

Niektóre nowe cechy IPv6 w porównaniu do IPv4 to:

- Rozmiar adresu – adres IPv6 ma 128 bitów, co powoduje, że każde urządzenie może posiadać własny adres IP, a pula adresów jest nie do wyczerpania w najbliższej przyszłości.
- Format nagłówka – zupełnie inny niż w czwartej wersji IP. Usunięto z niego rzadziej wykorzystywane opcje, przez co jest bardziej optymalny pod względem wykorzystywania.
- Nagłówki dodatkowe – założeniem protokołu IPv6 jest to, że powinien on mieć te same możliwości co IPv4. Osiąga się to poprzez dołączanie w miarę potrzeb dodatkowych nagłówków. Datagram IPv6 składa się zawsze z podstawowego nagłówka, po którym - w odróżnieniu od protokołu IPv4 używającego pojedynczego nagłówka - mogą się znajdować dodatkowe nagłówki.
- Lepsze wsparcie dla multimediiów – IPv6 umożliwia zestawienie pomiędzy nadawcą a odbiorcą kanału cechującego się wysoką wydajnością lub niskim kosztem. Dokonuje się tego jest poprzez ustawienie ścieżki przesyłu.
- Możliwość autoryzacji i zapewnienie bezpieczeństwa danych – poprzez nagłówki dodatkowe – *Authentication Header* (AH) oraz *Encrypted Security Payload* (ESP).
- Jumbogramy – czyli pakiety których pole danych przekracza 65535 bajtów. Cecha ta jest wykorzystywana w sieciach, w których medium jest w stanie przenieść więcej niż 65 kilobajtów w jednym ładunku.
- Rozszerzalność – protokół IPv6, dzięki możliwości dołączania dodatkowych informacji, jest bardziej elastyczny od swojego poprzednika czyniąc go bardziej uniwersalnym.
- Autokonfiguracja IP – która na podstawie adresu fizycznego przydziela adres IP o zasięgu lokalnym (*Local-link*).
- Nowy tryb komunikacji grupowej – anycast – pozwalający na wysłanie pakietu do grupy. Jego zaletą jest to, że pakiet zostanie odebrany wyłącznie przez jeden komputer w grupie.

- Zmiana polityki fragmentacji datagramów – fragmentacja odbywa się po stronie nadającego.
- Wsparcie dla IPv4 – poprzez mapowanie adresów IPv4 w IPv6.

Protokół używa nowego ulepszanego w stosunku do IPv4 nagłówka:



**Rysunek 3. Format nagłówka IPv6.**

*Wersja* – 4 bity – wersja protokołu; w tym wypadku równa 6.

*Priorytet* – 4 bity – wartość ta pozwala routerom rozróżniać potrzeby konkretnej konwersacji z danego źródła. Wartości zostały podzielone na dwa zakresy. Od 0 do 7 w przypadku, gdy źródło danych zajmuje się kontrolą zatorów oraz 8 do 15 w przypadku, gdy wymagany jest przesył ze stałą prędkością.

*Etykieta przepływu* – 24 bity – pole to jest używane przez routery do zapewnienia odpowiedniej jakości usług (QoS – ang. *Quality of Service*), dla konwersacji, które wymagają odpowiedniej obsługi. Pozwala to uniknąć tzw. problemu naruszenia warstw (ang. *The Layer Violation Problem*), z którego powodu routery musiały sięgać do wyższych warstw ISO/OSI aby zapewnić odpowiednie zasoby na potrzeby połączenia.

*Długość pola danych* – 16 bitów – długość pola danych za nagłówkiem. Wartość 0 oznacza, że informacja o rozmiarze danych znajduje się w dodatkowym nagłówku (a dokładnie – w nagłówku hop-by-hop).

*Następny nagłówek* – 8 bitów – określa, co znajduje się za nagłówkiem IPv6. Używane wartości są takie same jak w IPv4.

*Limit skoków* – 8 bitów – wartość ta jest zmniejszana o 1 przez każde urządzenie, które przesyła pakiet dalej. Pakiet jest kasowany, gdy wartość ta osiągnie 0.

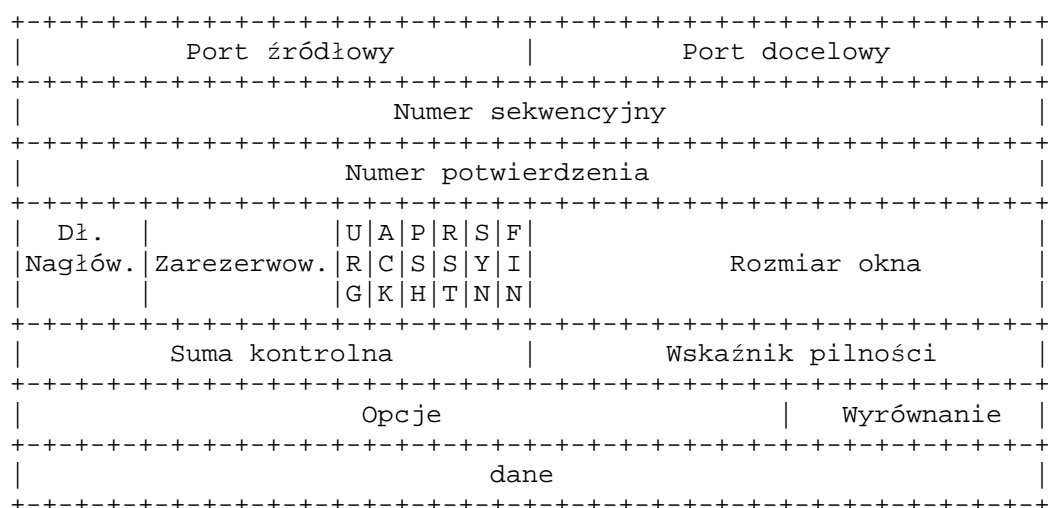
*Adres źródłowy* – 128 bitów – adres nadawcy.

*Adres docelowy* – 128 bitów – adres odbiorcy; odbiorca wymieniony w tym pakiecie nie musi być końcowym odbiorcą, jeśli obecny jest nagłówek routingu.

Protokół IPv6, który w obecnym stanie znajduje się w fazie testowania i wdrażania, posiada wiele zalet w stosunku do IPv4. Jego zaletą jest to, że znacznie lepiej organizuje miejsce w nagłówku, którego wielkość jest stała i zawsze równa 40 bajtów – oktetów. Istnienie dodatkowych nagłówków umożliwia rozszerzanie protokołu i dostosowywanie go do aktualnych potrzeb Internetu.

## 2.5 Protokół TCP

Protokół TCP w warstwie transportowej tworzy wirtualny kanał, w którym dane przesyłane są bezbłędnie (wykorzystywany jest mechanizm sum kontrolnych). Transmisja w kanałach odbywa się dwukierunkowo, a każda z dwu stron może wysłać dane w dowolnym momencie. Protokół ten jest zorientowany połączeniowo, co oznacza, że istnieje schemat nawiązywania połączenia. Fakt istnienia połączenia gwarantuje rozróżnialność połączeń w obrębie wielu połączeń pomiędzy tymi samymi systemami końcowymi. TCP operuje na strumieniowym mechanizmie przesyłania danych, co powoduje, iż nie ma gwarancji odebrania danych w tylu częściach, w ilu je nadano.



**Rysunek 4. Format nagłówka TCP.**

*Port źródłowy oraz port docelowy* – 16 bitów każdy – są to logiczne wartości pozwalające systemowi operacyjnemu na jednoznaczne przyporządkowanie pakietu do połączenia – kanału komunikacyjnego, który z kolei jest skojarzony z aplikacjami po stronie stacji źródłowej i docelowej.

*Numer sekwencyjny* – 32 bity – jest to licznik bajtów wysłanych podczas transmisji przez kanał. Wartość ta wybierana jest losowo w momencie połączenia i zwiększana o N wraz z każdym pakietem, gdzie N - liczbę oktetów w pakiecie.

*Numer potwierdzenia* – 32 bity – pole używane przy nawiązanym połączeniu. Jego zadaniem jest powiadamianie nadawcy o ilości poprawnie odebranych danych. Razem z poprzednim polem tworzy to mechanizm sterowania przepływem oraz usuwania błędów transmisji.

*Długość nagłówka* – 4 bity – określa, gdzie zaczynają się dane, a liczone jest w 32 bitowych słowach.

*Zarezerwowane* – 6 bitów – pole zawsze równe 0

*Flagi* – 6 bitów – używane przy określaniu i zmianie stanu połączenia.

URG: oznaczenie pola wskaźnika pilności,

ACK: pole potwierdzenia zawiera aktualne dane,

PSH: funkcja przepychania,

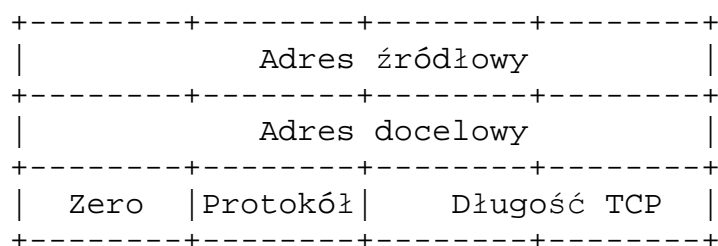
RST: zresetuj połączenie,

SYN: żądanie synchronizacji numerów sekwencyjnych,

FIN: ustawiane przy zakańczaniu połączenia.

*Rozmiar okna* – 16 bitów – liczba bajtów, którą zdalny system jest w stanie zaakceptować. Umożliwia to sterowaniem szybkości transmisji. Jeśli pole to jest równe 0 to nadawca musi przerwać transmisję. Wznowienie transmisji następuje w momencie wpisania przez odbiorcę liczby większej od 0.

*Suma kontrolna* – 16 bitów – zapewnia poprawność danych. Liczona jest dla pseudonagłówka (którego schemat ideowy przedstawia rysunek) oraz danych (wraz z nagłówkiem TCP).



**Rysunek 5. Format pseudonagłówka TCP.**





Protokół TCP dzięki temu mechanizmowi zapewnia pewność połączenia. Na połączenie muszą wyrazić zgodę dwie strony – strona usługodawcy (serwera) i strona usługobiorcy (klienta). Istnieje możliwość wykorzystania różnych metod do zaatakowania protokołu TCP. Jedną z możliwości jest wykorzystanie faktu, że w momencie nawiązywania połączenia następuje rezerwacja zasobów. W ten sposób można zmusić zdalny komputer do wykorzystania całej dostępnej dla stosu TCP pamięci, uniemożliwiając podłączenie się innym maszynom.

Fakt ten wykorzystałem do przeprowadzenia ataku *SYN Flood*[AIE-1] za pomocą programu DDoS, który potrafi wygenerować właśnie taki typ ataku, co opisuję w niniejszej pracy.

Istnieje wiele mechanizmów zabezpieczających przed atakiem tego typu. Należy tutaj wymienić przede wszystkim syncookie stosowane w linuxie, syncache z freebsd oraz wewnętrzny mechanizm windowsów z linii NT.

Innym rodzajem ataku jest atak wykorzystujący błąd w implementacji stosu TCP/IP o nazwie *Land*[AIE-1]. Dzięki niemu, możliwe jest całkowite zablokowanie systemu na czas około 10 sekund. Atak ten polega na ustawieniu adresu docelowego równego *adresowi źródłowemu* (równemu adresowi ofiary) oraz ustawieniu takich samych wartości w polu *port źródłowy*, *port docelowy*. Zaatakowana maszyna zachowuje się tak, jak gdyby sfałszowany pakiet był pakietem autentycznym i odpowiada zgodnie ze schematem konwersacji. W konsekwencji serwer zmuszony jest do wysyłania pakietu do samego siebie. Atak ten jest o tyle ciekawy, że był znany już w 1997 roku. Oznacza to, że należy baczniej się przyglądać nowym stosom TCP/IP, ponieważ w ich implementacjach popełniane są wciąż te same błędy.

## **2.6 Protokół UDP**

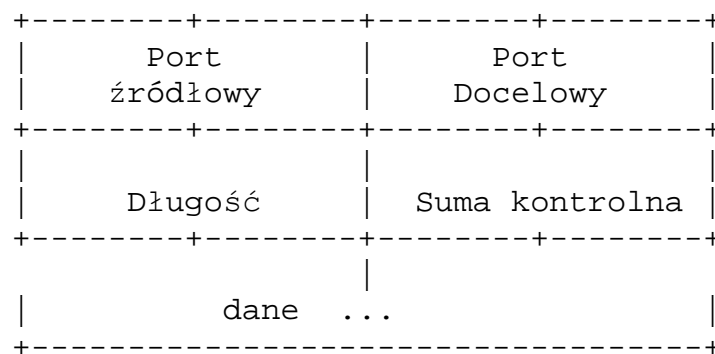
Protokół UDP, ze względu na poprawność przesyłanych danych oraz rozróżnialność danych w obrębie komunikacji pomiędzy dwoma komputerami, jest podobny do TCP. Na tym kończą się podobieństwa; zadania protokołu TCP i UDP są różne:

- Ciąg pakietów UDP nie tworzy strumienia danych, każdy pakiet należy traktować osobno.
- W przypadku UDP brak jest gwarancji dostarczenia pakietów; nie można polegać na tym, że pakiet nadany, zostanie odebrany po drugiej stronie kanału.
- Wysyłając dwa pakiety – założmy A i B – UDP nie daje gwarancji odebrania tych pakietów w tej samej kolejności przez odbiorcę.

- W bardzo rzadkich przypadkach, po wysłaniu pakietu A, odbiorca może dostać dwukrotnie ten sam pakiet.
- UDP jest szybszy od TCP; TCP w przypadku odebrania pakietu n+1 a nie odebraniu pakietu n, oczekuje na retransmisję pakietu n (co może zająć nawet około 3 sekund).  
Powoduje to, że dane w pakiecie n+1 mogą być już nieaktualne.

Wymienione wyżej różnice sprawiają, że protokół UDP znakomicie sprawdza się w przypadku transmisji multimedialnych czasu rzeczywistego, w których wymagane jest jak najszybsze dostarczenie danych do odbiorcy.

Na potrzeby protokołu UDP został wymyślony nowy format nagłówka, który realizuje cele założone dla UDP:



**Rysunek 7. Format nagłówka UDP.**

Każde z pól nagłówka UDP ma 16 bitów.

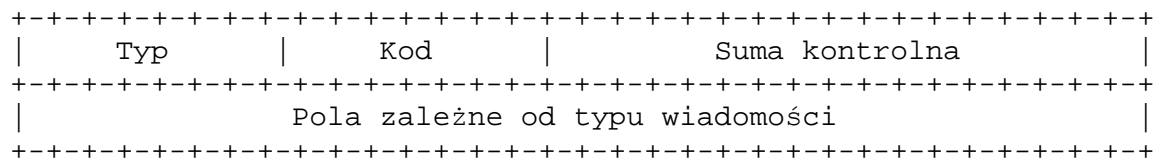
Protokół UDP, pomimo swojej prostoty, stwarza również możliwość wykorzystania do przeprowadzenia ataku. W tym wypadku ataki nie opierają się głównie na cechach charakterystycznych protokołu, ale na usługach oferowanych za jego pośrednictwem. Atakami takimi są na przykład *fraggle* oraz *pingpong*[PT-1], które wykorzystywały usługi ECHO oraz CHARGEN. Ataki te posługiwały się serwerami z zainstalowanymi usługami, które były wykorzystywane jako powielacze natężenia ruchu.

## **2.7 Protokół ICMP**

Protokół ICMP jest ściśle związany z IP i pełni funkcję informacyjną. Głównym jego przeznaczeniem jest :

- informowanie o braku możliwości dostarczenia pakietu,

- określenie czasu potrzebnego na dotarcie do zdalnego serwera i z powrotem (tzw. Ping),
- wyznaczenie trasy wędrówki pakietów,
- chwilowe żądanie wstrzymania nadawania,
- informowanie o przekroczonym czasie życia pakietu.



**Rysunek 8. Format nagłówka ICMP.**

*Typ* – 8 bitów – identyfikuje komunikat.

*Kod* – 8 bitów – dalsze informacje o komunikacie.

*Suma kontrolna* – 16 bitów – zapewnia poprawność nagłówka lub nagłówka i danych.

Protokół ICMP umożliwia nie tylko badanie i diagnostykę łącza, ale również sterowanie przepływem danych. Cecha ta sprawia, że protokół ten znalazł się w centrum zainteresowania intruzów, chcących przeprowadzić zdalny atak.

Najprostszym atakiem wykorzystującym protokół ICMP jest atak zwany *smurfing*[AIE-1]. Polega on na zalewaniu serwera sieciowego pakietami „ping”, co spowoduje, że jego łącze do sieci może zostać szybko wysyczone. Wykorzystuje się tutaj technikę powielaczy, o której będzie mowa w kolejnym rozdziale.

Innym rodzajem ataku jest atak logiczny – *ping of death*[AIE-1], wykorzystujący dodatkowo mechanizmy fragmentacji pakietu IP. Atakujący wysyła wiele fragmentów komunikatu ICMP, które po złożeniu mają rozmiar znacznie większy od maksymalnego rozmiaru pakietu IP.

Istnieje wiele mechanizmów zabezpieczenia się przed atakami z wykorzystaniem protokołu ICMP. Jednym z nich jest na przykład blokowanie wszystkich komunikatów ICMP z wyjątkiem ICMP ECHO REQUEST. Rozwiązanie to ma jednak swoje wady. Podstawową z nich jest utrata wszystkich przywilejów płynących z możliwości ICMP.

## 2.8 Fragmentacja danych w sieciach IPv4 oraz IPv6

Patrząc na model ISO/OSI, należy zauważyć, że pierwsze dwie warstwy - fizyczna oraz łącza danych, są zależne od medium po którym odbywa się transmisja. Nakłada to ograniczenie na ilość przenoszonych tą drogą danych. Dla jednego medium może to być 1500 bajtów (Ethernet), a dla innego na przykład 17966 (Cisco ATM). Ponieważ protokół IP jest w stanie przenosić ładunki o rozmiarach przekraczających maksymalną wielkość pakietu w medium – tak zwanego MTU (ang. *Maximal Transmission Unit*), konieczna jest fragmentacja.

### 2.8.1 Kontrola Fragmentacji w IPv4

Fragmentacja w protokole IPv4 wykonywana jest zarówno po stronie klienta, jak i poprzez maszyny pośrednie. Za kontrolę fragmentacji odpowiedzialne są trzy pola w nagłówku – *Identyfikator*, *Flagi* oraz *Przesunięcie fragmentu*.

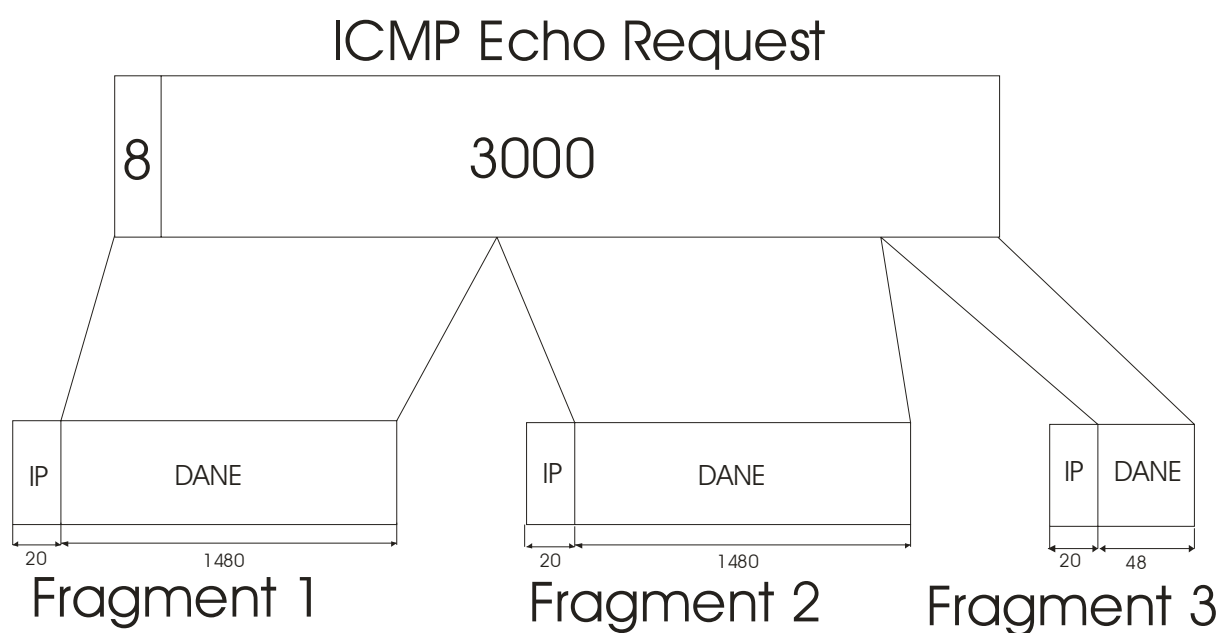
16 bitowe pole *Identyfikator* – zawiera liczbę całkowitą, pozwalającą odróżnić fragmenty pomiędzy różnymi połączeniami. Na przykład, jeśli komunikat ma rozmiar 5000 i zostanie podzielony na 4 fragmenty, to w tych czterech fragmentach pole *identyfikator* powinno być takie samo i mieć unikalną wartość w ramach połączeń pomiędzy dwoma rozmawiającymi węzłami.

Pole *znaczników* posiada 3 bity kontrolujące fragmentację. Pierwszy z bitów (określany popularnie bitem RF) nie jest używany przy większości transmisji – wykorzystywany jest jedynie przez programy sondujące, potrafiące na podstawie odpowiedzi wywnioskować, jaki system operacyjny został zainstalowany na zdalnym komputerze. W normalnych przypadkach pole to jest równe 0. Ustawiając drugi ze znaczników (DF) na wartość 1 - bezwzględnie zakazujemy podziału fragmentu na części. W sytuacji, gdy pakiet na którejś z maszyn nie może zostać podzielony, następuje jego odrzucenie, a do nadawcy wysyłany jest komunikat diagnostyczny. Ostatnia z flag (nazywana MF) powiadamia odbiorcę, czy aktualnie odebrany fragment jest ostatni (flaga ustawiona na wartość 1) czy spodziewać się więcej fragmentów z pakietu (flaga ustawiona na 0).

*Przesunięcie fragmentu* jest 13 bitowym polem zawierającym informację o miejscu w oryginalnej wiadomości, w którym znajdują się dane przesyłane w aktualnym fragmencie. Umożliwia to poprawne scalenie fragmentów w całość wiadomości. Zwiększenie tego pola o 1 oznacza przesunięcie danych o 8 bajtów. Oznacza to, że najmniejszy fragment ma rozmiar właśnie 8 bajtów.

Najlepiej zilustrować to przykładem:

Użytkownik chce wysłać komunikat ICMP Echo Request o rozmiarze 3000 bajtów. Ponieważ MTU w sieci LAN jest równe 1500, a ilość danych w minimalnym pakiecie IP równa jest 1480 bajtów, to pakiet o rozmiarze 3000+nagłówek = 3008 bajtów zostanie podzielony na  $3008/1480 =$  dwa pakiety o rozmiarze danych 1480 bajtów i jeden pakiet o rozmiarze danych 48 bajtów.



**Rysunek 9. Podział komunikatu ICMP na fragmenty w protokole IPv4.**

Fragment 1:

Identyfikator: 1000

Flagi

RF: 0

DF: 0

MF: 1

Przesunięcie fragmentu: 0

Fragment 2:

Identyfikator: 1000

Flagi

RF: 0

DF: 0

MF: 1

Przesunięcie fragmentu: 185

Fragment 3:

Identyfikator: 1000

Flagi

RF: 0

DF: 0

MF: 0

Przesunięcie fragmentu: 370

## 2.8.2 Kontrola Fragmentacji w IPv6

Fragmentacja w protokole odbywa się za pomocą specjalnego dodatkowego nagłówka (nie będącego częścią nagłówka tak, jak to jest w IPv4) opcji IPv6. Należy podkreślić, że za fragmentację danych odpowiedzialne jest wyłącznie źródło danych a nie (jak w przypadku IPv4) węzły pośrednie. Obecność nagłówka jest sygnalizowana wartością 44 w polu *Następny Nagłówek*, a jego format jest następujący:

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Nast.  Nagł.  | Zarezerwowane | Przesunięcie fragmentu | Res | M |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Identyfikator                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

**Rysunek 10. Format nagłówka fragmentacji IPv6.**

*Następny Nagłówek* – 8 bitów – określa, jakiego protokołu dane znajdują się za aktualnym nagłówkiem.

*Zarezerwowane* – 8 bitów – pole inicjalizowane na 0 przy wysyłaniu; przy odbieraniu ignorowane.

*Przesunięcie fragmentu* – 13 bitów – przesunięcie fragmentu w oryginalnej wiadomości; przesunięcie jest wyliczane względem fragmentowanej części oryginalnego pakietu.

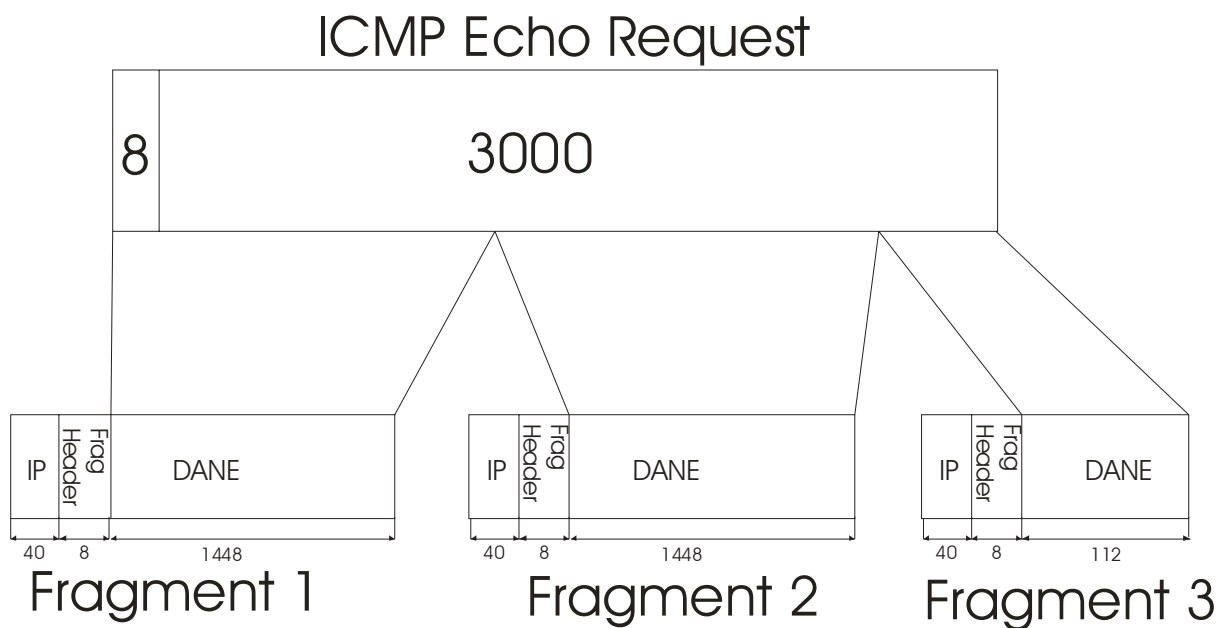
*Res* – 2 bity – zarezerwowane pole równe 0 przy wysyłaniu; przy odbieraniu ignorowane.

*M* – 1 bit – flaga, która ustawiona oznacza więcej fragmentów; wyzerowana – ostatni fragment.

*Identyfikator* – 32 bity – pole zawierające liczbę całkowitą, jednoznacznie przyporządkowującą fragment do składanego pakietu.

Przykład:

Jako przykładem posłużę się poprzednim przypadkiem. MTU jest równe 1500 bajtów. Maszyna chce wysłać komunikat ICMP Echo Request o rozmiarze 3000 bajtów.



**Rysunek 11. Podział komunikatu ICMP na fragmenty w protokole IPv6.**

Opcje IP nie różnią się we fragmentach IP (poza polem *Długość pola danych*). Pakiety różnią się natomiast polami w nagłówku fragmentacji.

Fragment 1:

Następny nagłówek: ICMv6 (0x3A)

Przesunięcie fragmentu: 0

M: Tak

Identyfikator: 0x0000029A

Fragment 2:

Następny nagłówek: ICMv6 (0x3A)

Przesunięcie fragmentu: 1449

M: Tak

Identyfikator: 0x0000029A

Fragment 3:

Następny nagłówek: ICMv6 (0x3A)

Przesunięcie fragmentu: 2896

M: Nie

Identyfikator: 0x0000029A



### 3. Luki, ataki sieciowe

Ataki sieciowe mogą mieć miejsce w wyniku istnienia luk w oprogramowaniu. Ogólnie rzecz biorąc: luka oprogramowania to podatność systemu na wykonywanie działań, które niezgodne z zamierzeniem programisty – twórcy systemu – powodują utratę kluczowych zasobów systemowych.

#### 3.1 *Klasyfikacja luk w systemie*

W literaturze powstało wiele klasyfikacji luk systemowych, których część można znaleźć pod [WCU-1]. Jedną z nich jest klasyfikacja dokonana przez Internet Security Systems ([www.iss.net](http://www.iss.net)), w ramach której wymienia się następujące kategorie:

- luki w oprogramowaniu, sprzęcie lub inne powstałe w procesie produkcji,
- luki wprowadzone do systemu przez administratorów, na skutek ich niewiedzy lub celowo (na przykład aby uprościć czynności administracyjne),
- luki wprowadzane do systemu przez użytkowników w trakcie korzystania z systemu.

Wśród luk wprowadzanych przez producenta wyróżnić można: błędy, łaty aktualizacyjne, które nie były odpowiednio przetestowane, pakiety naprawcze lub nowe narzędzia administracyjne oraz domyślne konfiguracje systemu, zawierające niezabezpieczone usługi.

Luki wprowadzone przez administratorów to przede wszystkim: niepoprawna, niezgodna z wymogami bezpieczeństwa konfiguracja systemu np. włączone dodatkowe usługi systemowe, które nie są niezbędne do działania systemu, czy też rezygnacja z wymuszenia minimalnej długości hasła.

Luki wprowadzane przez użytkownika powstają przeważnie na skutek ignorowania reguł bezpieczeństwa np. nie aktywowanie domyślnego *firewalla* czy nie stosowanie aktywnego skanera antywirusowego.

Luki, jak już wcześniej wspomniałem, mogą być potencjalną przyczyną ataku. Istnieje wiele definicji ataków. Mianem atak może być określana „każda czynność, które powoduje naruszenie spójności, poufności bądź dostępności systemu i przetwarzanych nim danych” czy też „każda czynność albo ciąg czynności wzajemnie powiązanych podejmowanych przez intruza celem stworzenia zagrożenia dla systemu przez wykorzystanie obecnych w nim luk i słabości”. Definicje te mają znaczenie opisowe i określają większość znanych ataków,

w tym również wykorzystujące socjotechnikę, dla których luką w systemie są jego użytkownicy.

### **3.2 Kim jest potencjalny włamywacz?**

Włamywaczem nazywamy osobę, która inicjuje atak. Każdego włamywacza charakteryzuje wiedza teoretyczna, praktyczne umiejętności, czas i miejsce działania, poziom kwalifikacji etc. Włamywacz może działać sam lub w grupie. Ogólnie da się podzielić włamywaczy na następujące kategorie:

- hakerzy – realizują ataki dla własnej satysfakcji, chcąc podnieść samoocenę albo status w środowisku,
- szpiegdy – organizują ataki celem pozyskani poufnych informacji,
- terroryści – ich celem jest szantaż,
- szpiegdy gospodarczy – szpiegdy, którzy działają z pobudek ekonomicznych, dla korzyści finansowych, lub którzy chcą przynieść straty konkurencji,
- zawodowi przestępcy – działający dla własnego zysku,
- wandal – ich jedynym celem jest przyjemność czerpana z niszczenia.

Najczęściej winą za włamania obarcza się hakerów. Ocena ta jest bardzo niesprawiedliwa, gdyż etyka większości hakerów nie pozwala na wandalizm. Pojęcie hakera ewoluowało przez lata. Początkowo – hakerem nazywano osobę wysoko wykwalifikowaną, która potrafiła zmusić komputer do wykonywania czynności nie ujętych w dokumentacji. Później był on wysokokwalifikowanym specjalistą przewodzącym w swoim środowisku. Etykieta „hakera” została wykoślawiona w latach siedemdziesiątych. Hakerzy zaczęli włamywać się do systemów operacyjnych i łamać zabezpieczenia banków. To właśnie dlatego pojęcie haker w dzisiejszych czasach ma tak pejoratywny wydźwięk. W dużej mierze odpowiedzialność za to ponoszą media, które dla rozgłosu mianem „hakera” określały każdego przestępcę komputerowego.

### **3.3 Jakie są cele włamywaczy?**

Każdy przeciętny użytkownik nie dopuszcza do siebie myśli, że może się stać celem ataku. Jest to częściowo prawdą, jednakże nie należy bagatelizować zagrożenia wynikającego z użytkowania sieci; zwykły szary użytkownik może być celem pośrednim, którego pokonania wymaga cel główny. We współczesnym świecie, większość ataków nie

jest przeprowadzona za pośrednictwem jednego komputera, lecz za pomocą wielu maszyn często zorganizowanych hierarchicznie, na której szczycie znajduje się komputer włamywacza. Warto jest znać motywy działania włamywacza, ponieważ pomaga to często zidentyfikować włamywacza i pozwala zabezpieczyć niektóre systemy, które są atrakcyjne dla szczególnej grupy intruzów.

Ogólna klasyfikacja celów włamywaczy przedstawia się następująco:

- Włamanie dla żartu.  
Jest to najczęstszy motyw ataków hakerów, którzy uwielbiają powodować niecodzienne sytuacje, tak jak niemiecki klub hakerski, który przejął kontrolę nad systemem sterowania budynku: gasił i zapalał światła w budynku tak, by mozaika oświetlonych okien tworzyła rysunki widoczne z zewnątrz.
- Zaspokajanie ciekawości.  
Wiele programów udowadniających istnienie luk w oprogramowaniu pojawia się w postaci gotowych narzędzi. Nazwy tych narzędzi oraz ich opisy przyciągają raczkujących wandal, którzy nie posiadają odpowiedniej wiedzy i umiejętności, a powodowani ciekawością testują oprogramowanie na losowo wybranych komputerach, często przyczyniając się do znacznych strat finansowych.
- Dla sławy i chwały.  
Podrastający wandal, kierowany chęcią wzbudzenia podziwu i strach. Osobnik taki szuka uznania wśród rówieśników albo w szerszym gronie obejmującym każdego, kto potrafi odebrać pocztę internetową. Zazwyczaj ich działania polegają na podmianie stron internetowych i umieszczaniu na nich nieparlamentarnych haseł.
- Polityka i ideologia.  
Hakerzy i wandal, to czasami osoby świadome politycznie (co nie jest dobrą nowiną dla całej reszty użytkowników Internetu) i chcąc być aktywni manifestują swoje przekonania na stronach www.
- Korzyści finansowe.  
Jest to podstawowy motyw działania włamywaczy przestępców – pod tym względem świat cyberprzestrzeni niczym nie różni się od świata materialnego.
- Odwet.  
Przyczyną odwetu może być rozczarowanie zbyt niską pensją, błędami w zarządzaniu, czy też niesprawiedliwym traktowaniem – nie mówiąc już o zwolnieniu pracownika. Aż dziw, że polski Internet istnieje i ma się dobrze!

- Wandalizm.  
Jego przejawami są przede wszystkim ataki odmowy usługi – czyli tymi, o których niniejsza praca traktuje. Celem większości jest zablokowanie jakiejś usługi lub spowodowaniem utraty danych przez ofiarę.
- Inne motywy.  
Wszystko, co nie da się sklasyfikować za pomocą powyższego podziału.

### **3.4 Modele ataków**

Generalnie można wyróżnić dwa modele ataku:

- Tradycyjny.  
Atak ten opiera się na koncepcji „jeden na jednego” bądź „jeden na wielu”. Bardzo często atakujący korzysta z węzłów pośredniczących, umożliwiających ukrycie faktycznego źródła ataku. Atak taki z reguły nie jest atakiem siłowym (np. SYN-FLOOD), lecz wykorzystuje luki w oprogramowaniu, czy też atak socjotechniczny.
- Rozproszony.  
Ataki rozproszone są względną „nowością” na rynku komputerowym. Jedno z pierwszych doniesień o możliwości skutecznego przeprowadzenia ataku rozproszonego pojawiło się we wrześniu 1998 roku w ośrodku Naval Surface Warfare Center. Model ten wymaga przyjęcia nowej koncepcji zabezpieczeń i wymusza na programistach wdrażanie nowych mechanizmów zabezpieczeń przed włamaniami. Ataki rozproszone opierają się na wielu koncepcjach np. „wielu na wielu”, „wielu na jednego”, jednakże nie da się jednoznacznie określić schematu połączeń pomiędzy atakującymi węzłami. Często korzysta się z wielowarstwowych węzłów pośrednich, które komunikują się z niższą warstwą węzłów i tak dalej. Działanie takie ma na celu ukrycie źródła ataku.

### **3.5 Przyczyny ataków**

Bardzo trudno jednoznacznie określić powody ataku; przyczyną ataku może być wszystko. W większości przypadków da się je jednak sprowadzić do następujących kategorii:

- Deklarowanie, że dany system jest nie do złamania.

Lepszego powodu do tzw. hakowania nie można dać hakerowi! Deklaracja ta jest o wiele skuteczniejsza niż poproszenie grupy hakerskiej o przetestowanie oprogramowania (tak hakerzy mogą pomagać znajdować dziury, jeśli zapewni im się odpowiednia dla nich nagrodę – w postaci np. umieszczenia w oknie informacyjnym aplikacji). Przykładem może być tutaj udane włamanie do oprogramowania Oracle 9i, którego kampania promocyjna etykietowana była hasłem „Oracle9i. Baza nie do złamania. Nie da się jej złamać. Nie da się do niej włamać”. W przeciągu roku znaleziono w aplikacji wiele błędów, mających wpływ na poziom bezpieczeństwa baz danych.

- Liderowanie danemu segmentowi rynku.

Na przykład cyber-wojna pomiędzy dwoma firmami, uczelniami czy ... akademikami

- Wymuszanie pożądaných przez atakującego działań.

Przykładem tu może być atak na witrynę RIAA pod koniec lipca 2002 roku. RIAA [TIQ-1] jest firmą, której nie lubi 99 procent społeczeństwa Internetu, a którą lubi 99 procent społeczeństwa związanego z wielkimi koncernami fonograficznymi. Firma zajmuje się zwalczaniem wymiany nielegalnych plików w sieciach p2p.

- Świadczenie usług finansowych, pośredniczenie w transakcjach handlu elektronicznego.

Pieniądże przechowywane są w bankach, o tym nie należy przekonywać nikogo – nawet włamywaczy. Rzeczywistość znowu znajduje swoje odzwierciedlenie w cyberświecie.

- Dysponowanie znaną nazwą i marką.

Witryny dużych firm odwiedzane są przez dużą liczbę osób. Stąd włamanie na taką witrynę szybko przynosi „sławę”.

- Prowadzenie działalności związanej z łącznością i bezpieczeństwem systemów informatycznych.

Hakerzy uwielbiają włamywać się na witryny firm, które zajmują się bezpieczeństwem.

- Inne przyczyny, których ilość i analiza wykracza poza ramy niniejszej pracy.

### **3.6 Klasyfikacja ataków DoS**

Przeprowadzono wiele klasyfikacji ataków. Ataki mogą być dzielone na pasywne i aktywne, wewnętrzne i zewnętrzne, celowe i niecelowe itp.. Aby nie wprowadzać zamieszania dużą liczbą podziałów rozmaitych ataków, skupię się na klasyfikacji ataków DoS.

Ataki DoS można podzielić na [CTA-1]:

- Ataki zużywające zasoby systemowe
  1. Ping Flood
  2. DoS - SYN Flood
  3. DDoS Syn Flood
  4. Distributed Reflected Denial of Service (DRDoS)
  5. Napcha
  6. UDP Flood
- Ataki wykorzystujące domyślne zachowanie protokołu
  1. Smurf
  2. Atak na DNS
- Ataki wynikające z błędu w oprogramowaniu
  1. Land
  2. Ping of Death
  3. Atak fragmentami Teardrop

#### **3.3.1 Ataki zużywające zasoby sieciowe**

Ataki zużywające zasoby systemowe mają na celu przeciążenie oraz zużycie zasobów systemowych dostępnych dla ofiary. Zasobami systemowymi może być: przepustowość sieci lub też ilość dostępnej pamięci operacyjnej, jak również czas procesora. Ataki takie, przeprowadzone poprzez dużą ilość komputerów pośredniczących, mogą na przykład wykorzystać całą przepustowość dostępną pomiędzy ofiarą a ISP, co uniemożliwi ofierze jakiegokolwiek korzystania z zasobów Internetu. Ruch generowany przez atak może zostać podzielony na dwie kategorie. Jedną z nich jest atak bezpołączeniowy (wykorzystuje „gołe” pakiety IP, UDP oraz protokół ICMP), nastawiony na konsumpcję przepustowości łącza. Drugim typem ataku jest atak nastawiony na protokoły połączeniowe (takie jak TCP), które

nie tylko zużywają przepustowość, ale również mogą unieruchomić urządzenia pośrednie (routery, switchy, ściany ogniowe etc.).

Protokół ICMP (*Internet Control Message Protocol*), jak sama nazwa wskazuje, służy do kontroli przepływu pakietów, diagnozowania sieci oraz błędów pomiędzy komputerem klienta a docelową maszyną, z którą klient chce się połączyć. Ataki ICMP polegają na wysyłaniu jednego rodzaju komunikatu z maksymalną prędkością. Komunikatem tym jest ICMP Echo Request, który wysłany do zdalnego komputera powoduje wygenerowanie i wysłanie przez niego komunikatu ICMP Echo Reply. Komunikacja taka, w normalnych warunkach, służy do ustalenia dostępności zdalnego komputera oraz do detekcji opóźnienia występującego po drodze. Wysyłanie komunikatów ICMP z dużą prędkością powoduje wysycenie łącza ofiary i w efekcie uniemożliwienie transmisji do innych węzłów.

U podstaw ataków DoS SYN Flood, leży wykorzystanie schematu nawiązywania połączenia przez protokół TCP. Zgodnie z tym, o czym była mowa w rozdziale 2.5, wysłanie pakietu na otwarty port powoduje inicjalizację mechanizmu nawiązywania połączenia. Przy każdym nawiązywaniu połączenia pewne zasoby są alokowane na rzecz przyszłego połączenia. Oczywiście istnieje limit połączeń półotwartych (takich, w których tylko pierwsza część protokołu nawiązywania połączenia została zakończona), jednakże działa to tylko i wyłącznie na korzyść atakującego, ponieważ zaatakowana usługa w efekcie końcowym nie będzie w stanie obsłużyć nowych połączeń. Na rzecz limitowania połączeń pół otwartych natomiast przemawia fakt, iż w tak zabezpieczonym systemie, atakujący nie jest w stanie spowodować znacznego zużycia pamięci i czasu procesora. Gdy pakiet z ustawioną flagą SYN dotrze do systemu, po zaalokowaniu zasobów systemowych zmniejszana jest pula dostępnych połączeń w ramach gniazda nasłuchującego. Taka sytuacja utrzymuje się przez parę sekund, aż do momentu uznania przez system, że połączenie nie może dojść do skutku. Jednakże te parę sekund może być wystarczające, by nie pozwolić komuś innemu na połączenie się. Przy założeniu, że system obsługuje 5 połączeń na sekundę, a atakujący wysyła 10000 pakietów na sekundę, prawdopodobieństwo połączenia się z zaatakowanym serwerem jest znikomo małe. Celem tego ataku jest zablokowanie dostępu do usługi na zaatakowanym komputerze. Należy zaznaczyć, że atak ów nie powoduje w większości przypadków utraty danych znajdujących się na dyskach twardych. W dzisiejszych czasach ataki te są zarówno łatwo wykrywalne, jak i nieskuteczne, ponieważ w większości przypadków ofiary dysponują dobrym sprzętem i łączem.

Naturalną ewolucją ataku DoS jest atak DDoS (*Distributed Denial of Service*). Zmodyfikowaniu uległa liczba komputerów biorących udział w ataku oraz rozproszenie

źródła ataku, co spowodowało wzrost skuteczności ataku. Pomysł ataku pasuje do modelu rozproszonego – wiele komputerów koncentruje swój atak na jednej maszynie. Często komputery te mają różne lokacje a ich użytkownicy nie są świadomi tego, że biorą udział w ataku. Do przeprowadzenia ataku wykorzystywane jest oprogramowanie zwane *końmi trojańskimi*, które zainstalowane na komputerze ofiary, zazwyczaj niezabezpieczonej i na stałe podpiętej do Internetu, służy do sterowania atakiem. Zaatakowane maszyny nie są w stanie obsłużyć wielu połączeń w tym samym czasie, a wysyczone łącze powoduje, że „dobry pakiet” ma małe prawdopodobieństwo przedostania się.

Pomysłem, który powstał przy okazji DDoS jest atak RDDoS (*Reflected Distributed Denial of Service*). Innowacją w stosunku do poprzedniego ataku jest to, że nie jest on przeprowadzany bezpośrednio. Atakujący wysyła fałszywe pakiety z ustawioną flagą SYN do routerów internetowych, które zachowując się tak, jak w przypadku prób nawiązania połączenia przez ofiarę ataku, wysyłają pakiety z ustawioną flagą SYN/ACK (zgodnie z trzystopniowym schematem uzgadniania połączenia) w kierunku ofiary. Zostaje ona zalana falą pakietów pochodzących od routerów internetowych, dużej ilości pakietów pochodzących z komputerów podłączonych do sieci. W najszcześniejszym przypadku – gdy obiekt ataku ma zainstalowaną aktywną ścianę ogniową - zablokuje ona połączenie pomiędzy ofiarą a routerami, co uniemożliwi ofierze jakiegokolwiek korzystanie z sieci (sama się od sieci odetnie). Niektóre odmiany ataku używają protokołu BGP (*Border Gate Protocol*). Protokół ten jest wykorzystywany przez routery do wymiany pomiędzy sobą tablic routingu, zawierających informację o zakresach IP z którymi routery te mogą się komunikować. Większość wysoko-wydajnościowych głównych routerów posiada tę usługę, a działa ona na porcie 179. Wysłanie powodzi pakietów na ten port spowoduje odpowiedź w kierunku nadawcy określonego w pakiecie.

Bardzo ciekawym rodzajem ataków jest atak typu Naptha [CERT-1], który jest rozwinięciem ataku SYN Flood. W tym przypadku atakujący wykorzystuje słabości protokołów TCP a dokładniej stosów TCP/IP i ich niezdolności do zarządzania dużą ilością połączeń będących w stanie innym niż „SYN RECVD”. Intruz mający możliwość fałszowania pakietów i zmieniania stanu dużej liczbie nieistniejących połączeń, może doprowadzić do „zagłodzenia” zdalny system, wykorzystując każdy dostępny zasób. Ewentualnie atakujący może zmusić zdalny system do utrzymywania po n połączeń, z których każde jest w jednym z 11 stanów TCP. Pomysł ten został wykorzystany również przez systemy IDS np. system LaBrea [SF-1], pozwalający na stworzenie wirtualnych komputerów



(ang. *Honeypots*), które utrzymują z atakującym nieistniejące połączenia powodując skutki u atakującego takie, jak podczas ataku Naptha.

Protokół UDP nie jest zorientowany połączeniowo, co nie wymusza na nim utrzymania sesji oraz jej negocjacji pomiędzy serwerem a klientem. Ataki z wykorzystaniem tego protokołu sprowadzają się zazwyczaj do zalewania celu dużą ilością pakietów, co jest możliwe między innymi dzięki prostocie protokołu UDP.

### **3.3.2 Ataki wykorzystujące domyślne zachowanie protokołu**

Proste ataki polegające na zalewaniu pakietami, mogą być dodatkowo wzmocnione poprzez wykorzystanie protokołów wyższych warstw takich jak TCP, UDP, ICMP w przypadku wykorzystania warstwy sieci, oraz BGP, DNS czy HTTP w przypadku warstwy aplikacji. Ataki takie są o tyle niebezpieczne, że nie wykorzystują słabości protokołu, a jego domyślne zachowanie .

Atak SMURF wykorzystuje domyślny schemat protokołu ICMP dla wiadomości żądania odpowiedzi. Wiadomość ta, służąca oryginalnie do sondowania zdalnego systemu czy „żyje” oraz wyliczania czasu dotarcia do niego, ma tę właściwość, że można sondować zdalne systemy pakietami o różnej wielkości (od 8B do 65KB). Atak SMURF polega na wysłaniu wiadomości ICMP Echo Request ze sfalszowanym adresem źródłowym na adresy rozgłoszeniowe sieci. Adres rozgłoszeniowy jest specjalnym adresem, który służy do komunikacji grupowej. Jeśli jakaś maszyna wyśle komunikat na ten adres, to komunikat ten zostanie odebrany przez wszystkie komputery w sieci. Domyślnym zachowaniem stosu TCP/IP po odebraniu komunikatu ICMP Echo Request jest odpowiedź na adres źródłowy komunikatem ICMP Echo Reply. W wyniku wysłania przez atakującego komunikatu ICMP, komputer ofiary zostanie zalany pakietami odpowiedzi ze wszystkich komputerów znajdujących się w sieci. W sieciach opartych o systemy firmy Microsoft, atak ten nie ma racji bytu, ponieważ w większości systemów, komunikat ICMP wysłany na adres rozgłoszeniowy jest ignorowany. Również system oparty o jądro Linux, posiada zabezpieczenie przed tym atakiem, jednakże w większości dystrybucji opcja ta jest domyślnie wyłączona.

Ataki na serwery dns nie są rzadkością w świecie cyber-terroryzmu. Ataki te polegają na zalaniu serwera DNS fałszywymi pytaniami o adres. Jako, że odpowiedź DNS jest większa od zapytania, cel zostaje zalany dużymi pakietami, co doprowadza do wysycenia łącza ofiary.

Ataki te możliwe są wyłącznie dzięki niefrasobliwym administratorom, którzy niepoprawnie skonfigurowali serwer DNS.

### 3.3.3 Ataki wynikające z błędu w oprogramowaniu

W odróżnieniu od strategii omówionych wcześniej, ta grupa ataków stara się znaleźć tzw. piętę Achillesową atakowanego systemu. Cel ataku nie jest osiągany przez wysyłanie ogromnej ilości pakietów, unieruchamiających system, a poprzez sprecyzowany atak o mniejszym natężeniu niż atak zalewania pakietami, który wykorzystuje niewłaściwą implementację bądź cechy charakterystyczne danego stosu TCP/IP.

Atak Land [ISS-2] polega na wysłaniu do ofiary specjalnie przygotowanego pakietu, w którym adres docelowy jest równy źródłowemu oraz w niektórych odmianach - port źródłowy równy docelowemu (wraz z ustawioną flagą SYN). Niezabezpieczone systemy operacyjne mogą podczas próby interpretacji pakietu wpaść w pętlę, która zawiesi system na czas rzędu minuty lub też doprowadzić do zawieszenia systemu w ogóle. Atak ten był popularny w roku 1997 roku, kiedy odkryto go w systemie Windows95 [SMC-1]. Atak ten jest skuteczny również w dzisiejszych czasach [SFC-1], a jego implementacja (jak udało mi się udowodnić) w protokole IPv6 po dziś dzień [SFC-2] (23 maja 2005) powoduje efekt DoS.

*Ping of Death* jest atakiem wykorzystującym błędy w mechanizmie scalania fragmentów zaimplementowanym w różnych systemach operacyjnych. Polega on na wykorzystaniu założenia, o maksymalnej długości pakietu IP równym  $2^{16}$ , że wiadomość ICMP, może mieć rozmiar większy od 65536 bajtów oraz że fragmentując komunikat można zmusić zdalny system do próby złożenia pakietu większego od maksymalnego rozmiaru pakietu IP.

Ataki fragmentami polegają na wykorzystaniu błędów w mechanizmach składania fragmentów. Fragmenty wysłane przez atakującego z reguły albo nie prowadzą do scalenia w oryginalną wiadomość, albo powodują duże obciążenie procesora. Niektóre ataki fragmentami polegają na wysłaniu fragmentów, które na siebie zachodzą, co stanowi problem dla systemu operacyjnego, którego stos TCP/IP nie potrafi sobie poradzić z taką sytuacją. Inne ataki

z kolei polegają na wysłaniu wszystkich fragmentów oprócz ostatniego, co powoduje ciągłe próby „defragmentacji” pakietu – alokowania i zwalniania pamięci [DN-1].

### **3.7 Metody ukrywania ataku**

Aby atak się powiódł, atakujący musi umieć albo skutecznie go przeprowadzić w przypadku braku aktywnych mechanizmów ochronnych, albo ukryć atak korzystając z następujących technik:

- fałszowanie adresu źródła ataku,
- tworzenie mylących pakietów (ang. *decoy*),
- wykorzystanie cudzego komputera w fazie ataku,
- fragmentacja ataku,
- szyfrowanie,
- podszywanie się pod domyślne usługi,
- zmiany standardowych scenariuszy ataku,
- spowolnienie ataku,
- czyszczenie dziennika systemu,
- ukrywanie plików i danych,
- ukrywanie procesów.

## 4. Jak wzmocnić stos TCP/IP

Istnieje wiele metod uniknięcia ataku polegającego na odmowie usługi. Nie jest to łatwe, ponieważ sama metoda ataku oraz wykorzystywane mechanizmy są trudne do zniwelowania. Twórcy systemów operacyjnych zdawali sobie sprawę z powagi tego zagrożenia i dlatego wiele systemów ma wbudowane mechanizmy zapobiegające skutkom ataku. Oprócz ustawień systemów zawsze należy:

- zawsze mieć zainstalowane najnowsze uaktualnienia dostępne na stronie producenta,
- umieć przystosować konfigurację sieciową stosu TCP/IP do potrzeb serwera i znać zalety oraz wady tego rozwiązania.

### 4.1 Wzmacnianie stosu w systemach Windows

Poniższy tekst został skopiowany ze strony Microsoftu [WMC-2] i opisuje mechanizmy wbudowane w systemy z linii Windows NT, które wzmacniają stos TCP/IP. Ustawienia zmienia się za pomocą edytora rejestru systemowego (regedit.exe).

Na poniższej liście zestawiono i opisano wartości rejestru związane z protokołem TCP/IP, których skonfigurowanie pozwala wzmocnić stos protokołu TCP/IP na komputerach połączonych bezpośrednio z Internetem. Wszystkie te wartości ustawia się w następującym kluczu rejestru:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services

UWAGA: O ile tego specjalnie nie zaznaczono, wszystkie wartości mają format szesnastkowy.

Nazwa wartości: SynAttackProtect

Klucz: Tcpip\Parameters

Typ wartości: REG\_DWORD

Zakres prawidłowych wartości: 0 , 1 , 2

Domyślna wartość: 0

Ta wartość rejestru powoduje, że protokół TCP dopasowuje retransmisję pakietów SYN-ACK. Gdy użytkownik skonfiguruje tę wartość, to w przypadku ataku znanego jako SYN będą obowiązywały krótsze limity czasów połączeń.

Na poniższej liście zestawiono parametry, których można używać z tą wartością rejestru:

- 0 (wartość domyślna): Ustawienie parametru *SynAttackProtect* na 0 zapewnia zwykłą ochronę przed atakami typu SYN.
- 1: Ustawienie parametru *SynAttackProtect* na 1 zapewnia lepszą ochronę przed atakami typu SYN. Parametr ten powoduje, że protokół TCP dopasowuje retransmisję pakietów SYN-ACK. Jeśli parametr *SynAttackProtect* ma wartość 1, to w przypadku rozpoznania ataku SYN obowiązują krótsze limity czasu odpowiedzi na żądania połączeń. W celu rozpoznania ataku system Windows wykorzystuje następujące wartości:
  - *TcpMaxPortsExhausted*
  - *TCPMaxHalfOpen*
  - *TCPMaxHalfOpenRetried*
- 2: Ustawienie parametru *SynAttackProtect* na 2 zapewnia najlepszą ochronę przed atakami typu SYN. Wartość ta powoduje wymuszenie dodatkowych opóźnień przy informowaniu o połączeniach, a w wypadku rozpoznania ataku typu SYN, powoduje szybszą realizację żądań połączeń. Jest to ustawienie zalecane.

UWAGA: Po ustawieniu parametru *SynAttackProtect* na 2 nie działają następujące opcje gniazd:

- Skalowalne okna
- Parametry protokołu TCP skonfigurowane na poszczególnych kartach (w tym początkowa wartość RTT i rozmiar okna)

Nazwa wartości: *KeepAliveTime*

Klucz: *Tcpip\Parameters*

Typ wartości: *REG\_DWORD* - czas w milisekundach

Zakres prawidłowych wartości: 1 – 0xFFFFFFFF

Domyślna wartość: 7 200 000 (dwie godziny)

Parametr ten określa, jak często protokół TCP usiłuje sprawdzić, czy bezczynne połączenie jest wciąż aktywne, wysyłając pakiet utrzymania aktywności. Jeśli komputer zdalny jest

wciąż osiągalny, potwierdza pakiet utrzymania aktywności. Pakiety utrzymania aktywności nie są wysyłane domyślnie. Skonfigurowanie tego parametru dla danego połączenia można powierzyć programowi. Ustawieniem zalecanym jest 300 000 (5 minut).

## 4.2 Wzmacnianie stosu w systemie Linux

System Linux posiada o wiele prostsze a zarazem bardziej efektywne zabezpieczenie przed atakami SYN Flood. Należy zdać sobie sprawę, że atak SYN Flood jest skierowany na zasoby systemu ofiary. Jeśli ofierze w jakiś sposób uda się uniknąć alokowania zasobów przy nawiązywaniu połączenia, to atak ten nie odniesie skutku w ogóle. Na takiej właśnie zasadzie działa mechanizm SYN Cookie [CYT-1].

Aby uaktywnić mechanizm należy przy kompilacji jądra włączyć:

```
NETWORKING OPTIONS ----> IP : TCP syncookie support
```

Po pomyślnej kompilacji i instalacji jądra, należy ten mechanizm jeszcze włączyć za pomocą polecenia:

```
echo „1” > /proc/sys/net/ipv4/tcp_syncookie
```

Idea SYN Cookie polega na opóźnieniu alokowania zasobów na potrzeby połączenia do momentu przyjścia ostatniego z pakietów w trzyczęściowym modelu nawiązywania połączenia. Działanie Syn Cookie z kolei polega na takim dobraniu początkowej wartości numeru sekwencyjnego, aby przy odebraniu odpowiedzi można było stwierdzić, że pakiet ten należy do połączenia i należy zaalokować zasoby na jego potrzeby.

Innymi opcjami konfiguracyjnymi stos TCP/IP są pliki znajdujące się w katalogach /proc/sys/net/ipv4/ oraz /proc/sys/net/ipv6/. Możliwych opcji jest dużo i umożliwiają one bardzo szczegółowe dostosowanie parametrów stosu.

Do uniemożliwienia ataku typu SMURF na przykład służy opcja *icmp\_echo\_ignore\_broadcasts* której ustawienie powoduje, że system Linux będzie ignorował komunikaty ICMP Echo wysłane na adres rozgłoszeniowy.

Opcja *tcp\_max\_syn\_backlog* z kolei ustawia długość kolejki nowych połączeń. Gdy dodatkowo włączona jest opcja *tcp\_syncookie*, to w momencie przepełnienia kolejki backlog, wykorzystywany jest mechanizm SYN Cookie, a rozmiar kolejki jest ignorowany.

Do ustawienia ilości prób sprawdzenia aktywności połączenia TCP służy opcja *tcp\_keepalive\_probes*. W połączeniu z opcją *tcp\_keepalive\_time*, która określa co jaki czas połączenie jest sprawdzane pod kątem poprawności umożliwia szybszą rotację połączeń w systemie i sprawniejsze usuwanie martwych połączeń.

Wszystkie pozostałe opcje opisane są w dokumentacji jądra znajdującej się domyślnie w pliku „usr/src/linux/Documentation/filesystems/proc.txt”

## 5. Programy wykonane w ramach pracy dyplomowej

Programy napisane w ramach niniejszej pracy pracują pod kontrolą systemu Microsoft Windows (XP/2k/2k3). Do kompilacji programu *DDoS Generator*, wymagany jest kompilator Borland C++ Builder w wersji 6. Do skompilowania pozostałych programów należy użyć kompilatora c++ z pakietu Microsoft Visual Studio 2003 w przypadku *WinProcTime*, *ConnectTime*, *SocketListener* oraz kompilatora gcc w przypadku programu *Load*.

### 5.1 Program „DDoS Generator”

Program *DDoS Generator* jest programem napisanym pod platformę Windows NT i służy do wykonywania ataków DDoS w sieciach opartych o Ethernet. Jego prostota sprawia, że jest to narzędzie bardzo groźne w rękach wandalów czy też początkujących „hakerów”.

#### 5.1.1 Jakie możliwości daje program?

Program *DDoS Generator* potrafi generować wartości dla określonych poniżej pól na 3 sposoby:

- losowo,
- pojedyncze wartości,
- z listy wartości wczytywanej z pliku.

Pola, dla których generowane są wartości w wyżej wymieniony sposób to:

- adresy MAC (ang. *Media Access Control*): źródłowy oraz docelowy,
- adresy IPv4 oraz IPv6: źródłowy i docelowy,
- porty TCP: źródłowy oraz docelowy.

Ponadto program umożliwia wykorzystanie listy, w skład której wchodzi adresy MAC oraz powiązane z nimi adresy źródłowe.

Program *DDoS Generator* jest programem potrafiącym wykonywać ataki SYN Flood o różnym natężeniu ruchu. Do dyspozycji atakującego jest 5 ustawień natężenia pakietów:

- określona liczba pakietów na sekundę,
- liczba pakietów, która zajmie określoną przepustowość; wartość ta mierzona jest w kilobajtach na sekundę,



- brak ograniczenia ilości pakietów: wysyłanie z maksymalną prędkością umożliwiającą przez sterownik WinPCap,
- wysyłanie ściśle ograniczonej liczby pakietów z maksymalną prędkością,
- rozpoczęcie wysyłania od N pakietów na sekundę i zwiększanie co Z sekund ilości pakietów na sekundę o P.

Wartości losowe wykorzystane w programie liczone są na 3 sposoby:

- prekalkulowanych jest 100000 liczb losowych za pomocą funkcji *libnet\_get\_prand*,
- używana jest funkcja *libnet\_get\_prand*,
- używana jest wbudowana funkcja *rand*.

Program umożliwia wykorzystywanie dwóch bibliotek do generowania pakietów:

- WinPCap w przypadku generowania ruchu IPv4 oraz IPv6,
- LibNet w przypadku IPv4 (IPv6 nie jest obsługiwane dla platformy Windows).

*DDoS Generator* potrafi tworzyć logi z postępem operacji, które są zapisywane do pliku tekstowego. Każda linijka w pliku z logiem zawiera:

T - S - N

Gdzie T to czas w postaci HH:MM:SS.mmm; S to średnia liczba pakietów na sekundę od momentu rozpoczęcia ataku; N to ostatnia zarejestrowana liczba pakietów na sekundę.

Interfejs sieciowy, przez który wysyłane będą pakiety, użytkownik określa w fazie konfiguracji ataku. Do dyspozycji są wszystkie interfejsy wykrywane przez bibliotekę WinPCap. Jeśli jakiś interfejs nie umożliwia przesyłania pakietów, to program zasygnalizuje to błędem podczas próby wysyłania.

### 5.1.2 Biblioteki wykorzystane do napisania programu

Operacje wysyłania pakietów realizowane są w programie przez dwie biblioteki: WinPcap[WPI-1] oraz libnet[WPB-1].

Biblioteka WinPcap służy do niskopoziomowego dostępu do warstwy łącza. Przy jej pomocy dokonywane jest „wstrzykiwanie” pakietów do karty sieciowej, która następnie transmituje je w sieć. Funkcjonalność biblioteki libnet jest zbliżona do pierwszej z wymienionych bibliotek. Różnica polega na poziomie abstrakcji przy tworzeniu wiadomości

sieciowych. WinPcap umożliwia dowolne formatowanie pakietów w sieci, począwszy od warstwy łącza. Z kolei libnet posiada dodatkowe mechanizmy umożliwiające w łatwy sposób tworzenie pakietów z wyższych warstw ISO/OSI.

Każda z bibliotek wymaga inicjalizacji oraz dostarcza obiektu – urządzenia, na którym dokonuje się operacji (np. pobierania czy wysyłania). W przypadku WinPcap jest to zmienna typu `pcap_t`, a w przypadku libnet - `libnet_t`.

Inicjalizacja obu obiektów jest podobna.

W pierwszym kroku znajdujemy wszystkie karty sieciowe

```
int pcap_findalldevs (pcap_if_t** alldevsp, char* errbuf)
```

gdzie `alldevsp` jest strukturą (której deklaracja znajduje się w pliku `pcap.h`):

```
struct pcap_if {
    struct pcap_if *next;
    char *name;
    char *description;
    struct pcap_addr *addresses;
    bpf_u_int32 flags;
};
```

Z tej struktury interesować nas będą na razie 2 pola – `name` oraz `description`. Pierwsze jest adresem urządzenia podobnym do np. `"\\Device\\NPF_{3EF43412-DE35-46D6-B006-0248A5FF3D25}"` a drugie jest opisem zrozumiałym dla człowieka (może być równe 0 gdy biblioteka nie jest w stanie pobrać opisu urządzenia). Gdy wywołanie funkcji `pcap_findalldevs` zwróci NULL, oznacza to że nie mamy zainstalowanej biblioteki WinPcap.

W kolejnym kroku należy otworzyć urządzenie za pomocą wywołania procedury:

```
pcap_t* pcap_open_live (char * device,
                        int      snaplen,
                        int      promisc,
                        int      to_ms,
                        char*     ebuf)
```

Pierwszym argumentem jest nic innego, jak pole `name` zwrócone w strukturze `alldevsp`. Parametr `snaplen` definiuje maksymalny rozmiar pakietów. Jeśli pakiet jest większy niż rozmiar tu określony, to zostanie obcięty do podanej wartości. `Promisc` określa czy karta ma działać w trybie, w którym widziane są wszystkie pakiety (nawet te, których adres docelowy MAC jest różny od MAC-a karty sieciowej) i może przyjmować wartości: 0 lub

PCAP\_OPENFLAG\_PROMISCUOUS. Za pomocą *to\_ms*, można ustalić długość czasu, który zostanie poświęcony na złapanie pakietu. Dokładniej mówiąc – jeśli pakiet zostanie przechwycony przez bibliotekę, to wywołanie funkcji *pcap\_next* zakończy się natychmiast. W przeciwnym wypadku wywołanie tej funkcji zakończy się po czasie określonym przez zmienną *to\_ms* w przypadku braku pakietów w łączy.

W przypadku biblioteki libnet inicjalizacja odbywa się za pomocą wywołania funkcji:

```
libnet_t * libnet_init (int injection_type, char *device, char
*err_buf)
```

Pierwszy parametr określa tryb wstrzykiwania pakietów w sieć. W przypadku DDoS Generators jest on równy LIBNET\_LINK, co daje pełną kontrolę nad pakietem począwszy od warstwy łącza. Inne możliwości opisane są w pliku

libnet-structures.h

...

```
00171 #define LIBNET_LINK      0x00/* link-layer interface */
```

```
00172 #define LIBNET_RAW4      0x01/* raw socket interface
(ipv4) */
```

```
00173 #define LIBNET_RAW6      0x02/* raw socket interface
(ipv6) */
```

...

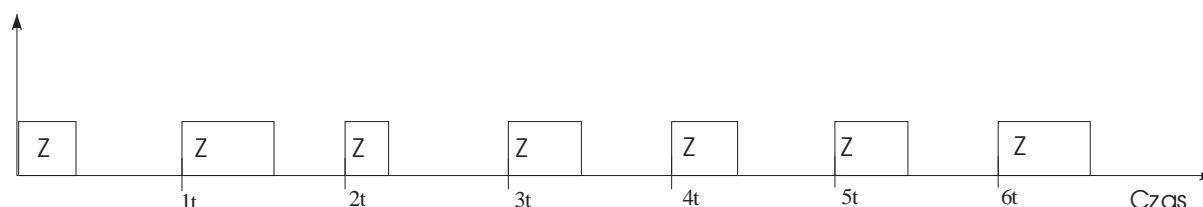
Kolejny parametr odpowiada parametrowi *device* z funkcji *pcap\_open\_live*.

Udana inicjalizacja umożliwia korzystanie z własności biblioteki. W przypadku biblioteki libnet należy ponadto stworzyć zmienną typu *libnet\_ptag\_t*, która będzie numerowała stos libnet. Dodatkowo istnieje możliwość zainicjalizowania generatora liczb pseudolosowych za pomocą funkcji *libnet\_seed\_prand*. W dalszym kroku czyścimy bufor pakietów wysyłanych za pomocą funkcji *libnet\_clear\_packet*. Każdy pakiet do wysłania tworzy się poprzez kolejne wywołania funkcji *libnet\_build\_tcp*, *libnet\_build\_ipv4* oraz *libnet\_build\_ethernet*. Wywołania te muszą być wykonywane w wyżej wymienionej kolejności, ponieważ pakiet budowany jest od najwyższej warstwy ISO/OSI do najniższej. Ostatnią rzeczą, jaką należy zrobić, jest wywołanie funkcji *libnet\_write*, która wyśle pakiet do karty sieciowej. Korzystanie z biblioteki WinPcap jest o wiele prostsze, ponieważ wywoływana jest tylko jedna funkcja, a mianowicie *pcap\_sendpacket*. Funkcja ta przyjmuje jako argument wskaźnik na dane do wysłania oraz rozmiar danych. O ile korzystanie z biblioteki WinPcap jest prostsze niż w przypadku libnet, o tyle biblioteka WinPcap nie wspiera tworzenia pakietów wyższych

warstw niż warstwa łącza. Oznacza to, że użytkownik WinPcap jest zmuszony do samodzielnego formatowania nagłówków oraz liczenia sum kontrolnych, podczas gdy biblioteka libnet policzy to automatycznie.

Deinicjalizacja bibliotek odbywa się za pomocą wywołań funkcji *pcap\_close* dla urządzenia WinPcap oraz *libnet\_destroy* dla libnet. Przed zamknięciem urządzeń bibliotek nie można zapomnieć o zwolnieniu zasobów zaalokowanych przez *pcap\_findalldevs*. Dokonuje się tego funkcją *pcap\_freealldevs*.

Wysyłanie pakietów dokonywane przez biblioteki jest procesem niedeterministycznym, zależącym od wielu czynników takich, jak: jakość urządzeń sieciowych, zajętość łącza oraz inne procesy zachodzące w karcie sieciowej. Ogólnie można przybliżyć to schematem:



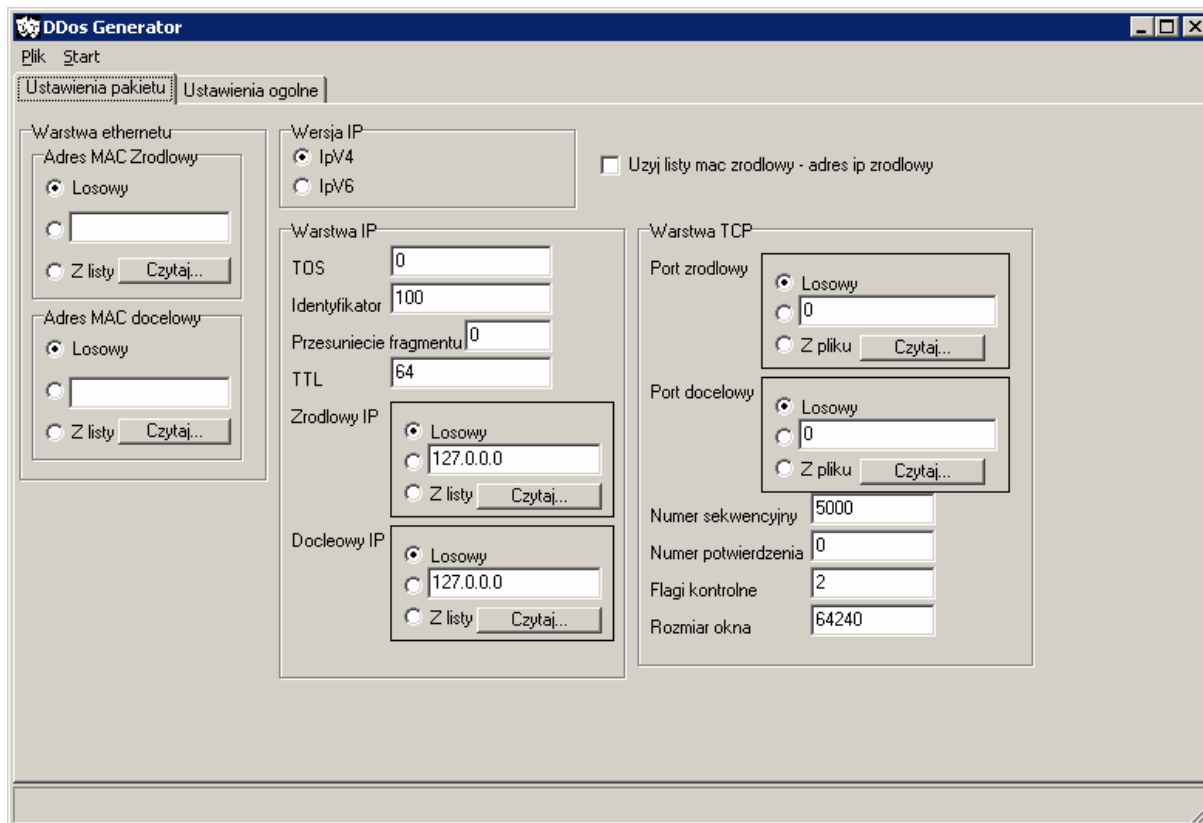
**Rysunek 12. Schemat czasu wysyłania pakietów do sieci.**

Czas  $Z$  jest czasem wysyłania pakietów w sieć. Pomiar ilości pakietów na sekundę następuje co  $1s$ , czyli czasie  $1t$ ,  $2t$ ,  $3t$  etc. Jeśli czas  $Z$  przekroczy  $1t$ , to mechanizm może ulec rozsynchronizowaniu. W programie widoczne to jest w momencie osiągania dużych prędkości wysyłania.

Innym rozwiązaniem byłoby podzielenie czasu  $1t$  na  $N$  kwantów. W każdym kwancie czasu wysyłany byłby 1 pakiet, a następnie program czekałby na następny kwant czasu, w którym może wysłać pakiet. Dałoby to bardziej „równomierne” pokrycie osi czasu. Jednakże biblioteki nie umożliwiają takiego zachowania w związku z wewnętrzną budową sterownika obsługującego wysyłanie pakietów, który nie daje możliwości wysłania pakietu w dowolnym czasie.

### 5.1.3 Instrukcja użytkownika

Po uruchomieniu programu ukaże się następujące okienko:



Rysunek 13. Postać programu po uruchomieniu.

Domyślnie aktywna zakładka zawiera opcje poszczególnych warstw: łącza, sieciowej oraz transportowej.

Wybór opcji warstwy łącza sprowadza się wyłącznie do określenia adresu MAC: źródłowego oraz docelowego. Do wyboru użytkownik ma 3 tryby wyboru każdego z adresów:

- losowy – preferencje losowania ustala się na zakładce ustawienia ogólne
- podany przez użytkownika w formacie XX:XX:XX:XX:XX:XX lub XX-XX-XX-XX-XX-XX, gdzie XX jest liczba heksadecymalna
- z listy zapisanej w pliku. Adresy w pliku powinny być rozdzielone znakami nowej linii.

W warstwie sieciowej istnieje możliwość wyboru używanego protokołu: IPv4 lub IPv6.

W przypadku użycia protokołu IPv4 można określić następujące opcje:

- TOS – wartość całkowita 8-bitowa,
- Identyfikator – wartość całkowita 16-bitowa,

- Przesunięcie fragmentu – wartość całkowita 13-bitowa,
- TTL – wartość całkowita 8-bitowa,
- Adresy IP: docelowy oraz źródłowy, których wyboru można dokonać w podobny sposób jak adresów MAC w warstwie łącza. Przy korzystaniu z listy adresów IP, format pliku jest podobny do formatu używanego w warstwie łącza z tą różnicą, że zamiast adresów MAC, lista powinna zawierać oddzielone znakami nowej linii adresy IP. Adresy IP muszą być podane w formie numerycznej [WAO-1] (ang. *dotted decimal*), ponieważ program nie rozwiązuje nazw na adresy IP.

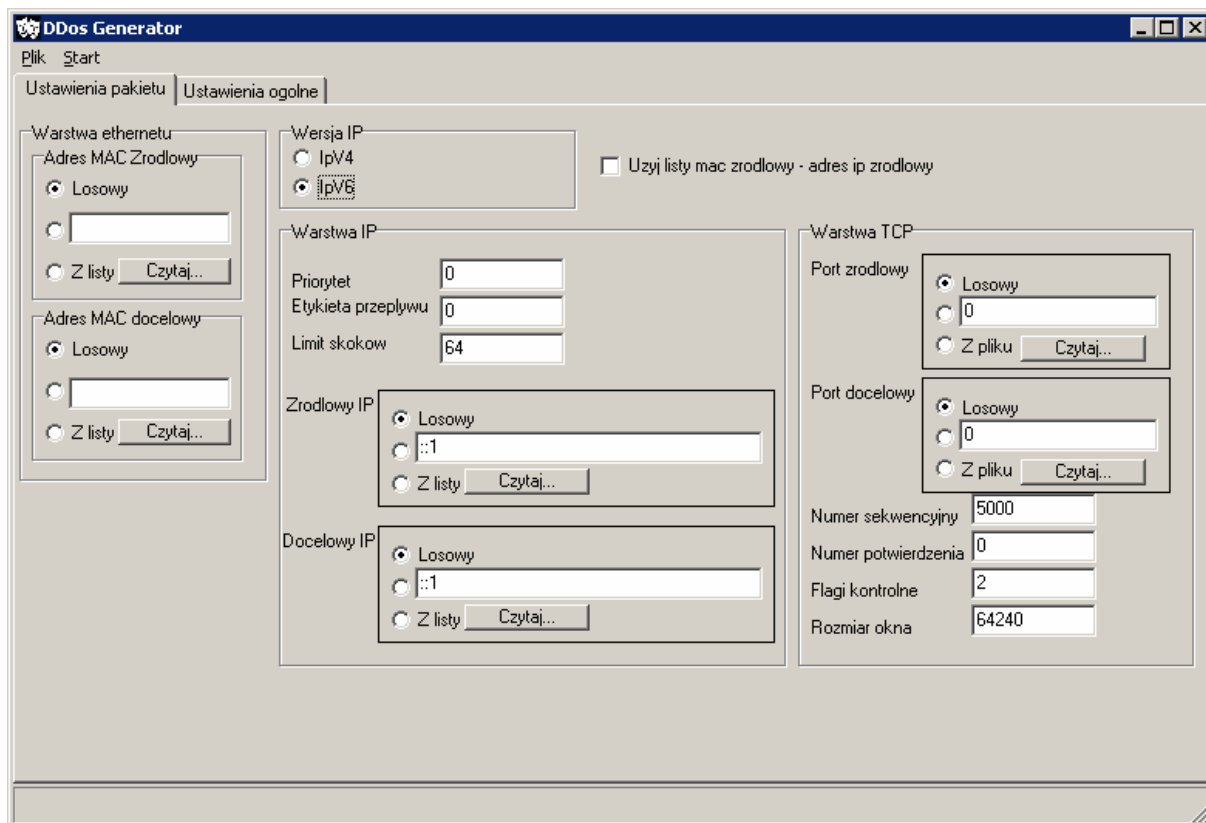
W wypadku korzystania z protokołu IPv6 użytkownik może ustawić następujące opcje:

- Priorytet – liczba całkowita 4-bitowa,
- Etykieta przepływu – liczba całkowita 24-bitowa,
- Liczba skoków – liczba całkowita 8-bitowa,
- Adresy IP: docelowy i źródłowy. Lista adresów, ma taki sam format jak w przypadku IPv4, a adresy muszą być w formie numerycznej IPv6 [WTC-1].

Opcjami warstwy transportowej są:

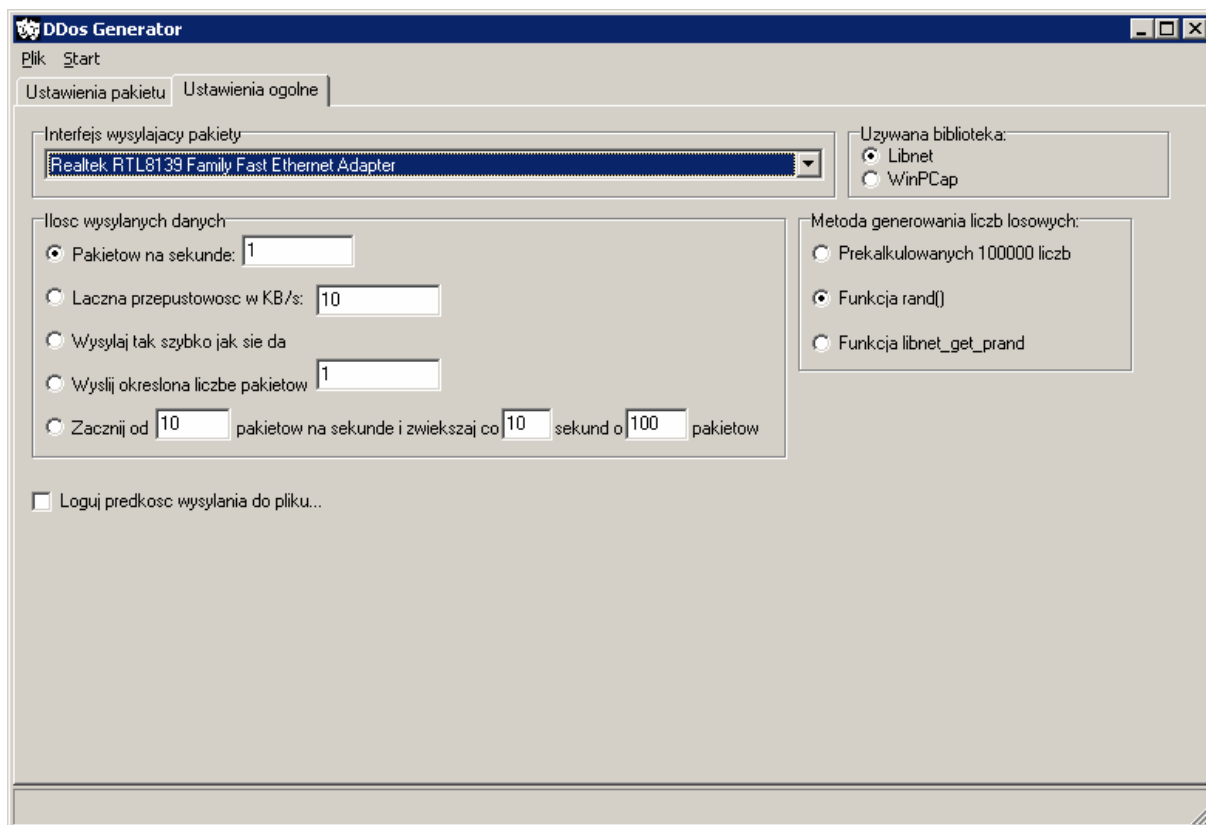
- Porty: źródłowy oraz docelowy – 16 bitowe liczby całkowite. Do wyboru są 3 podopcje: losowy, użytkownika oraz z pliku,
- Numer sekwencyjny – liczba całkowita 32-bitowa,
- Numer potwierdzenia – liczba całkowita 32-bitowa,
- Flagi kontrolne – liczba całkowita 6-bitowa,
- Rozmiar okna – liczba całkowita 16-bitowa.

W tej zakładce ustawić można również opcję, która spowoduje, że program będzie używał listy odwzorowań MAC->IP. Pozwala to na wykorzystywanie sieci lokalnej jako źródła ataku. Aby wejść w ten tryb należy włączyć opcję „Użyj listy mac zrodlowy - adres ip zrodlowy” i wskazać w kolejnym dialogu plik, zawierający listę. Plik z listą zawiera linię tekstu, z których każda zawiera parę „AdresIP AdresMAC”. Przy wysyłaniu danych adresy są brane kolejno z listy; elementem następnym po ostatnim z listy jest pierwszy.



Rysunek 14. Widok programu po ustawieniu protokołu na IPv6.

Ustawień konfiguracji wysyłania oraz opcji z wysyłaniem związanych dokonuje się na kolejnej zakładce:



Rysunek 15. Zakładka ustawień ogólnych.

Dostępne opcje obejmują:

- ustawienie interfejsu sieciowego używanego do wysyłania pakietów,
- określenie preferowanej biblioteki przy wysyłaniu: WinPcap dla IPv4,IPv6; libnet dla IPv4,
- wskazanie metody generowania liczb losowych,
- określenie natężenia ruchu,
- ustawienie pliku z logiem zawierającym aktualne natężenie ruchu.

Program wykrywa interfejsy sieciowe przy inicjalizacji. Jeśli w systemie nie ma bibliotek wymaganych przez program, to działanie aplikacji zakończy się po uprzednim wyświetleniu komunikatu o błędzie.

Wskazanie preferowanej biblioteki ma wpływ na wydajność aplikacji. Generalnie mówiąc - biblioteka WinPcap jest szybsza od biblioteki libnet. Użycie w aplikacji biblioteki libnet, miało charakter czysto edukacyjny, ponieważ interfejs programistyczny libnet w prosty sposób umożliwia tworzenie pakietów sieciowych.

Problem generowania liczb losowych nie jest problemem nowym w informatyce. Wiele wbudowanych generatorów ma tę wadę, że da się dla nich wyznaczyć okresu, po którym liczby losowe zaczynają się powtarzać. Program *DDoS Generator* daje możliwość korzystania z trzech sposobów generowania liczb losowych:

- użycie prekalkulowanych 100000 liczb – za pomocą *libnet\_get\_prand* i zwracanie kolejnych na żądanie modułu aplikacji,
- wykorzystanie wbudowanej funkcji *rand()*,
- użycie funkcji libnet o nazwie *libnet\_get\_prand*.

Używanie funkcji *rand* może spowodować to, że po pewnym czasie adresy IP czy porty, które mają być generowane losowo, zaczną się powtarzać, co nie jest zjawiskiem pożądanym w badaniach. Z kolei użycie funkcji *libnet* powoduje znaczny wzrost zapotrzebowania na czas procesora, co negatywnie wpływa na wydajność generowania pakietów.

Aplikacja pozwala na 5 sposobów generowania ruchu:

- ruch o stałym natężeniu; liczony w pakietach na sekundę,
- ruch o stałym natężeniu; liczony w kilobajtach na sekundę,
- ruch o maksymalnym natężeniu,
- ruch o maksymalnym natężeniu z ograniczeniem ilości pakietów,

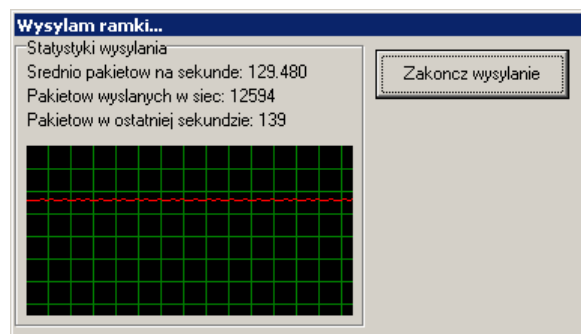


- ruch przyrostowy, którego natężenie początkowe jest określone i zwiększa się o podaną wartość co pewien podany czas.

Dodatkowo, dla celów analizy wyników, program posiada mechanizm logujący aktualne natężenie pakietów w danej sekundzie. Każda z linii w pliku zawiera trzy wartości oddzielone znakiem „-”:

- czas z dokładnością do milisekund,
- średnia liczba pakietów od początku generowania - w pakietach na sekundę,
- aktualne natężenie ruchu w pakietach na sekundę.

Gdy wszystkie opcje konfiguracyjne zostaną ustawione, należy wówczas wcisnąć przycisk *start*, który rozpocznie proces wysyłania pakietów do sieci oraz spowoduje pokazanie się okienka z postępem operacji.



Rysunek 16. Okienko z postępem operacji programu DDoS Generator.

Po naciśnięciu przycisku „Zakończ wysyłanie” proces wysyłania pakietów zostanie zakończony.

#### 5.1.4 Przykłady użycia programu w rzeczywistym środowisku

Opisywany program ma możliwość fałszowania wszystkich danych we wszystkich warstwach, począwszy od warstwy łącza. Daje to możliwość podszywania się pod dowolną maszynę w sieci.

Jako przykład podam sposób na wykorzystanie programu do zaatakowania jednego celu tak jakby przy wykorzystaniu wszystkich komputerów w sieci lokalnej. Pierwszym krokiem jest określenie adresu MAC routera w sieci lokalnej. Można tego dokonać poprzez otworenie dowolnej strony sieciowej lub jakiegokolwiek inne zainicjowanie komunikacji z komputerem

znajdującym się poza siecią lokalną. Następnie po wydaniu polecenia „arp -a” w okienku poleceń (wywołać je można za pomocą polecenia cmd.exe) ukaże się tzw. tablica ARP (ang. *Address Resolution Protocol*), która odwzorowuje adresy MAC na adresy IP w sieci lokalnej. Na maszynie testowej wyglądało to tak:

```
>arp -a
```

```
Interface: 157.158.181.42 --- 0x10004
```

Internet Address	Physical Address	Type
157.158.181.254	00-90-27-25-d2-f4	dynamic

Pozyskany adres MAC routera (00-90-27-25-d2-f4) wpisujemy w pole „Adres MAC docelowy” programu *DDoS Generator* i zaznaczamy obok przycisk wyboru nakazujący korzystanie z tego pola.

Następnie należy pobrać adresy MAC wszystkich użytkowników sieci. Można tego dokonać, podobnie jak w przypadku routera, poprzez wysyłanie do każdego z komputerów żądania komunikacji, a następnie sprawdzanie w tablicy ARP powiązania IP->MAC. Pomocnym tutaj może być program „Net Scan Tools” (<http://www.netscantools.com/>), który pozwala na wysyłanie komunikatów żądania echa do wszystkich komputerów w lokalnej sieci.

Gdy już posiadamy listę odwzorowań IP->MAC, należy zapisać ją do pliku w postaci:

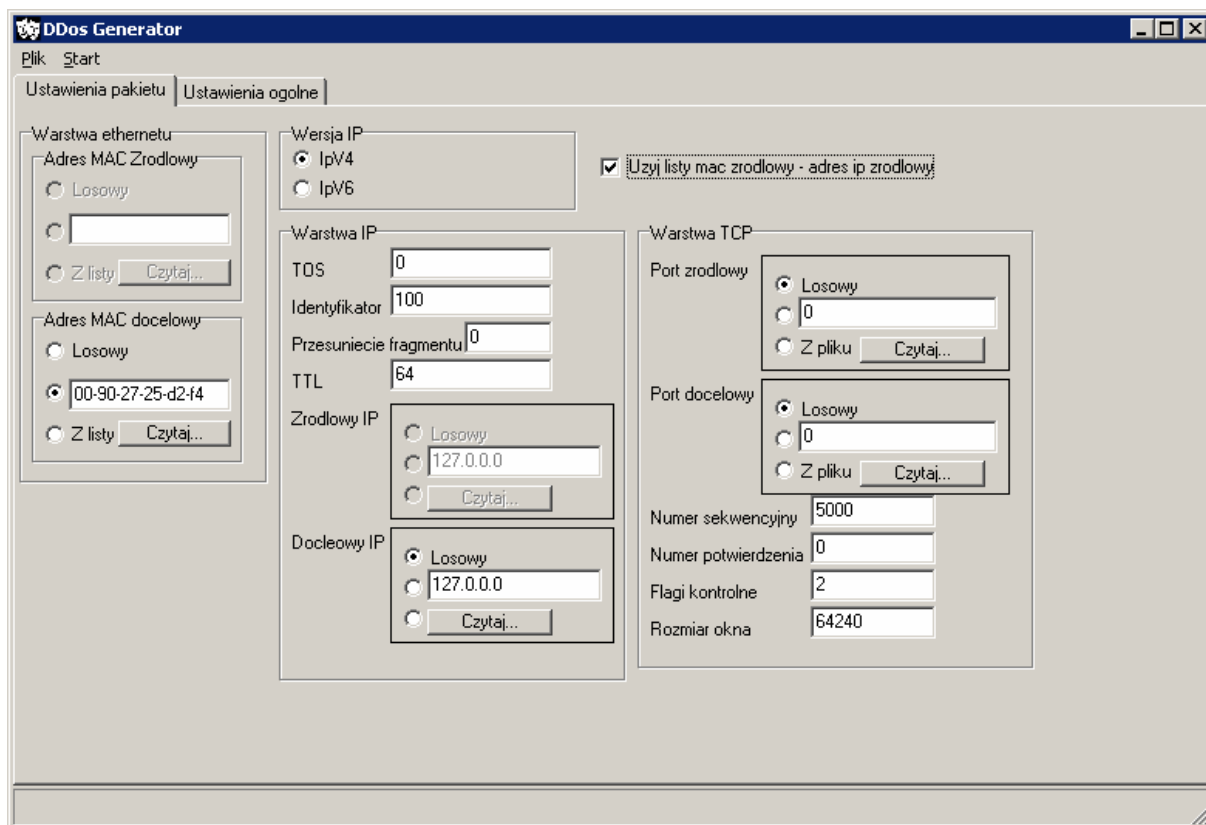
```
AdresIP MAC
```

```
AdresIP MAC
```

```
...
```

```
AdresIP MAC
```

Program *DDoS Generator* potrafi korzystać z takiej listy przy ustalaniu źródłowych adresów IP i powiązanych z nimi adresów MAC. Ustawienia tego dokonuje się poprzez zaznaczenie opcji „Użyj listy mac zrodlowy - adres ip zrodlowy” oraz wskazanie pliku z adresami.



Rysunek 17. Wygląd planszy programu po ustawieniu listy IP->MAC.

Ostatnim krokiem jest ustawienie celu ataku, portu docelowego oraz innych opcji, takich jak flagi kontrolne, TTL, TOS, natężenie ruchu pakietów, interfejs sieciowy używany do wysyłania pakietów itp.

Gdy to wszystko zostało ustawione, wystarczy nacisnąć przycisk „Start”.

## 5.2 Program mierzący czas połączenia – ConnectTime

Zadaniem programu ConnectTime jest badanie dostępności połączenia ze zdalnym hostem. Jeśli program zostanie uruchomiony bez parametrów - ukaże się komunikat z podpowiedzią dotyczącą parametrów.

```
>connecttime.exe
Sposob uzycia: connecttime.exe -t -a [-i -b]
-t docelowa maszyna, (postac: -t adres!port)
-d czas oczekiwania pomiedzy probami polaczenia (Domyslnie:
1000ms)
-l logowanie do pliku, (postac: -l wynik.txt)
-c ilosc prob polaczenia - 0 dla nieskonczonosci (Domyslne:0)
-r odbieraj dane z serwera zdalnego
-f PF_INET lub PF_INET6 dla protoko|u IPv4/IPv6 (Domyslnie
IPv4)
-v wlacza tryb gadatliwy
-s wysylaj dane do serwera zdalnego (po odebraniu)
```

Program generuje wyniki w postaci:

*Aktualny czas - B1 - C1 - B2 - C2 - B3 - C3*

B1,B2,B3 - Kod błędu - połączenia się do serwera, odbierania danych, wysyłania danych,

C1,C2,C3 - Czas łączenia się do serwera, odbierania i wysyłania danych.

Czas pokazywany jest w formacie: HH:MM:SS.mmm.

### **5.3 Przykładowa usługa sieciowa IPv4/IPv6 – SocketListener**

Program SocketListener symuluje działanie usługi sieciowej opartej o protokół IPv4 lub IPv6.

Po uruchomieniu programu bez parametrów przystępuje on do nasłuchu z domyślnymi parametrami. Gdy jako parametr programu poda się opcję „-h” ukaże się pomoc:

```
socketlistener.exe -a [-s -r ] [-f PF_INET|PF_INET6] [-v]
    -a adres nasłuchiwania, ("ip_address!port": 0.0.0.0!5555)
    -s wysyłaj dane po odebraniu
    -f PF_INET lub PF_INET6 dla nasłuchu IPv4/IPv6 (Domyślnie IPv4)
    -v tryb gadatliwy
    -r odbieraj dane przed wysłaniem
```

Program SocketListener można skompilować zarówno na platformie Windows jak i Linux.

### **5.4 Programy mierzący obciążenie procesora – WinProcTime oraz Load**

Programy te służą do mierzenia czasu zajętości procesora. Po wywołaniu programu WinProcTime z parametrem -h ukaże się pomoc:

```
usage:
    WinProcTime.exe [-s ms] [-l file]
    -s ms - czas pomiędzy pobraniami zużycia procesora
    -l file - określa plik z logiem
```

Program generuje wynik w postaci:

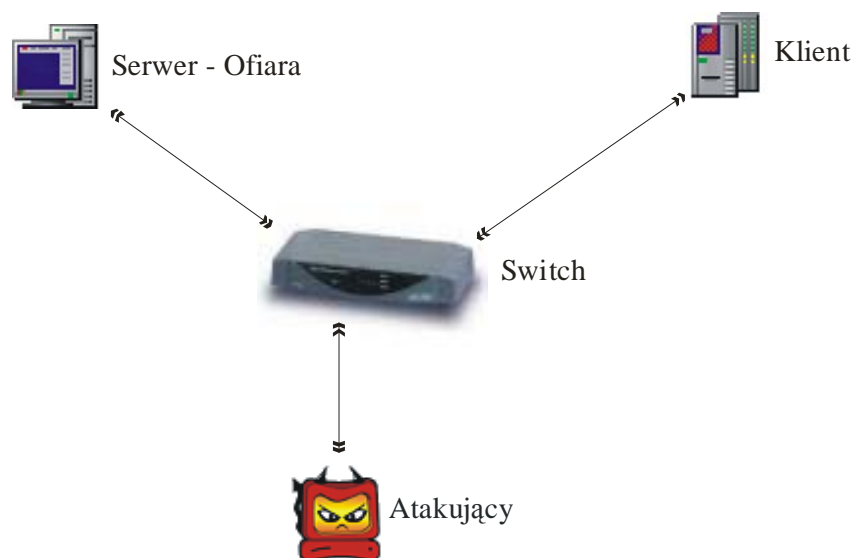
Czas – procentowe użycie procesora

Program Load z kolei nie przyjmuje żadnych parametrów, a jego uruchomienie powoduje wypisywanie na konsolę co 1000ms aktualnego obciążenia procesora.

## 6. Analiza wpływu ataków DOS/DDOS na popularne systemy operacyjne

### 6.1 Opis stanowiska wykorzystanego przy implementacji ataku

Stanowisko testowe składało się z trzech komputerów: ofiary, atakującego oraz klienta.



Rysunek 18. Schemat połączenia komputerów w stanowisku doświadczalnym.

#### Parametry komputerów

Ofiara:

Procesor: Athlon 2500+

Pamięć: 512MB DDR 3333

Karta sieciowa: 100MBit NForce

Atakujący:

Procesor: Duron 700 Mhz

Pamięć: 384 MB SDR 133 Mhz

Karta sieciowa: 100Mbit 3Com

Klient:

Procesor: Duron 800 Mhz

Pamięć: 512 MB DDR 266 Mhz

Karta sieciowa: 100Mbit Realtek RTL8139

Komputery były połączone switchem 100Mbit firmy Planet.

## 6.2 Opis ataku i pobierania wyników

Na popularne systemy operacyjne zostały przeprowadzone następujące ataki:

1. Atak „SYN Flood” [ISS-1]
2. Atak Land [ISS-2]

We wszystkich atakach został wykorzystany protokół zarówno IPv4 jak i IPv6.

Celem ataku pierwszego jest pokazanie istnienia problemu ataków SYN Flood oraz w miarę możliwości granicy, przy której serwis znajdujący się na maszynie serwera zaczyna odrzucać połączenia lub jest całkowicie zajęty. Granica ta, różna dla współczesnych systemów operacyjnych, może się okazać osiągalna dla przeciętnego użytkownika. Fakt ten powoduje, że atak ów może być przypuszczony przez każdego, kto posiada wystarczająco „mocne” łącze, co w dzisiejszych czasach nie jest trudne do osiągnięcia. Atak używa losowych adresów źródłowych, losowych adresów MAC oraz losowych portów źródłowych. Parametry natężenia pakietów zostały dobrane eksperymentalnie tak, aby test pojedynczej maszyny nie przekroczył jednej godziny, a wyniki prezentowały, w miarę możliwości, albo kres dolny albo kres górny ilości odrzucanych połączeń.

Atak Land ukazuje problem powtarzalności wykonywania błędów w implementacji stosów TCP/IP. Metoda ataku pojawiła się około 7 lat temu i aż trudno uwierzyć, że współczesne systemy operacyjne mogą być tak na nią podatne. Ponadto została stwierdzona skuteczność tego ataku w przypadku wykorzystania protokołu IPv6, co zostało zweryfikowane i oznaczone sygnaturą *CVE: CAN-2005-1649* oraz *Bugtraq ID 13658* na witrynie *securityfocus.com*. Test przeprowadzany był za pomocą programu *DDoS Generator*, który po skonfigurowaniu wysyłał jeden pakiet w kierunku ofiary. Na komputerze zaatakowanym uruchomiony był program *WinProcTime*, który mierzył obciążenie procesora.

Atakujący wykorzystywał program *DDoS Generator*, do wytworzenia ruchu o zmiennym natężeniu skierowanego do serwera-ofiary.

Rolę serwera-ofiary pełnił w testach program *SocketListener*, którego zadaniem jest nasłuchiwanie na określonym porcie, odebranie i obsłużenie połączenia skierowanego na ten port. Obsłużenie połączenia polega na wysłaniu porcji danych i odebraniu odpowiedzi od klienta. Testowany był protokół zarówno IPv4 jak i IPv6.

W roli klienta, wykorzystany został program ConnectTime, który co określony interwał czasu łączył się z maszyną serwera, odbierał i wysyłał porcje danych.

Programy DDosGenerator, ConnectTime oraz WinProcTime (w przypadku testów obciążenia procesora) generują pliki z logiem postępu operacji. Logi te były scalane za pomocą programu LogTimeSync, a następnie w przypadku logów z programu ConnectTime i DDosGenerator analizowane za pomocą programu ConnectStatistics. Plik wynikowy wykorzystywany jest do analizy i generowania wykresów. Niektóre wykresy zostały poddane dodatkowej obróbce – usunięto z nich część danych nie nadających się do interpretacji. Do synchronizacji czasu w logach wykorzystywany był program NetScanTools (<http://www.netscantools.com/>), który pozwalał na synchronizację zegara systemowego z zegarem serwera czasu – w tym przypadku wszystkie maszyny były przed testami synchronizowane z serwerem galaxy.uci.agh.edu.pl.

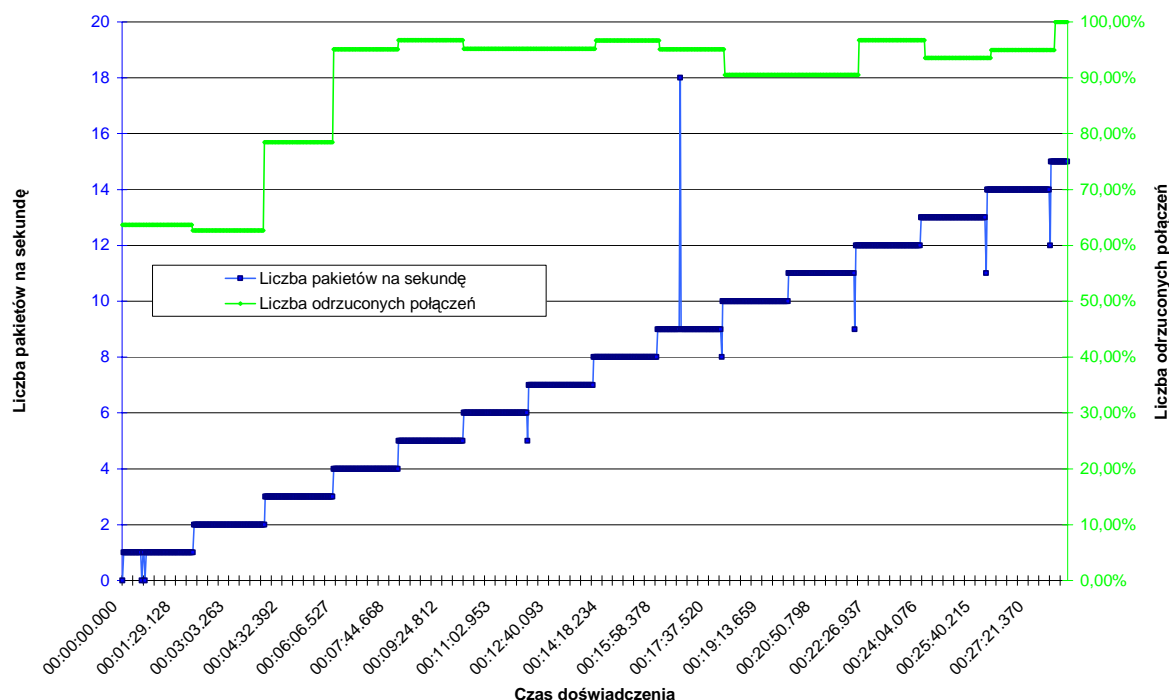
## **6.3 Wyniki**

### **6.3.1 System „Windows XP”**

System Windows XP jest najpopularniejszym systemem operacyjnym nie tylko wśród prywatnych użytkowników, ale również w biurach i dużych firmach. Często użytkownicy ci próbują uruchamiać różne serwisy świadczące usługi w sieci Internet. Testy wykazały, że nie jest to bezpieczne, ponieważ architektura tego systemu jest podatna na ataki sieciowe.

#### **6.3.1.1 System niezabezpieczony – wersja ze zintegrowanym SP1**

Testowanie rozpoczęto od uruchomienia na komputerze ofiary programu SocketListener i badania za pomocą programu ConnectTime zależności ilości poprawnie zakończonych konwersacji od ilości pakietów wysłanych przez atakującego (rys. 19).



Rysunek 19. Atak SYN Flood na port 5555 – protokół IPv4.

Jak wynika z wykresu, aby unieruchomić usługę pracującą pod tym systemem wystarczy wysłać stosunkowo niewiele pakietów na sekundę. Dzieje się tak między innymi dlatego, że ta wersja systemu nie respektuje parametru o nazwie *backlog* podanego jako argument funkcji *listen*.

*Int WSAAPI listen(IN SOCKET s, IN int backlog);*

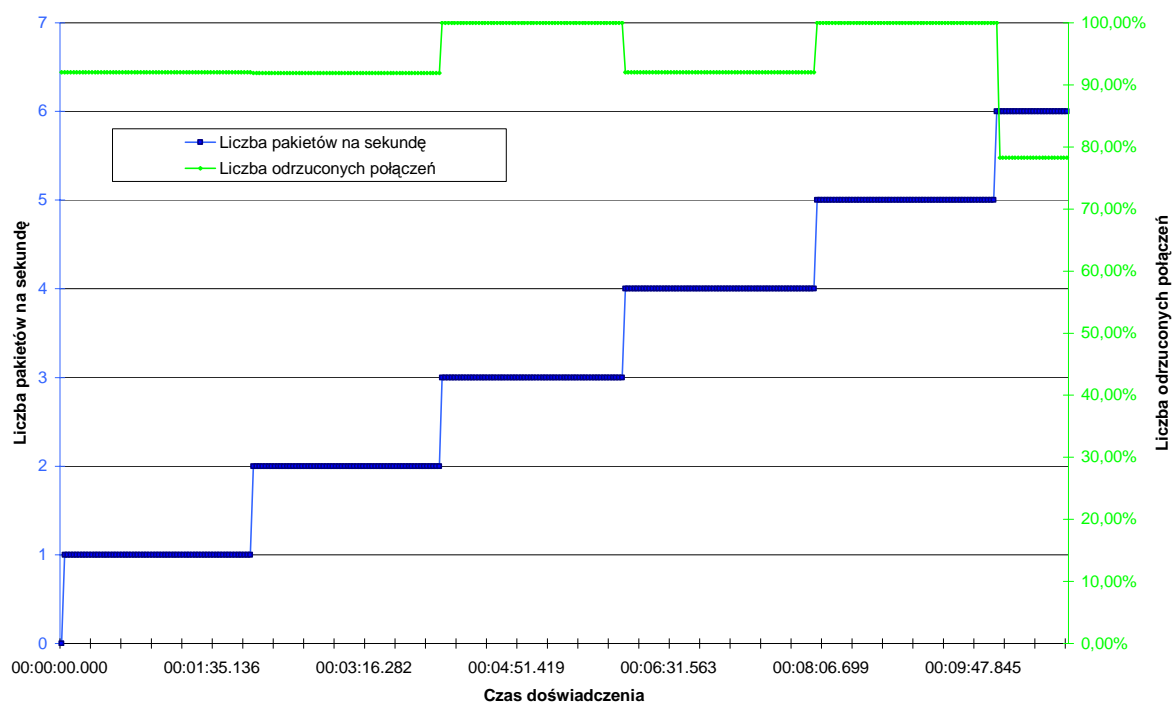
Wartość *backlog*, określa ile dla danej usługi system utrzymywać może połączeń nie w pełni ustanowionych. W dokumentacji MSDN napisano, że wartość ta ustawiona na *SOMAXCONN*, gwarantuje maksymalną liczbę połączeń półotwartych. Jest to zdefiniowane w plikach nagłówkowych:

- WinSock2.h  
#define SOMAXCONN 0x7fffffff
- WinSock.h  
#define SOMAXCONN 5

Ustawienie wartości *backlog* na wartość 0x7fffffff, jak udało się ustalić, nie powoduje zwiększenia kolejki połączeń oczekujących. W systemie Windows XP dla wszystkich programów działających w obszarze użytkownika wartość ta jest automatycznie zmniejszana do wartości 5.

Podobne wyniki otrzymano w przypadku protokołu IPv6 (rys. 20):



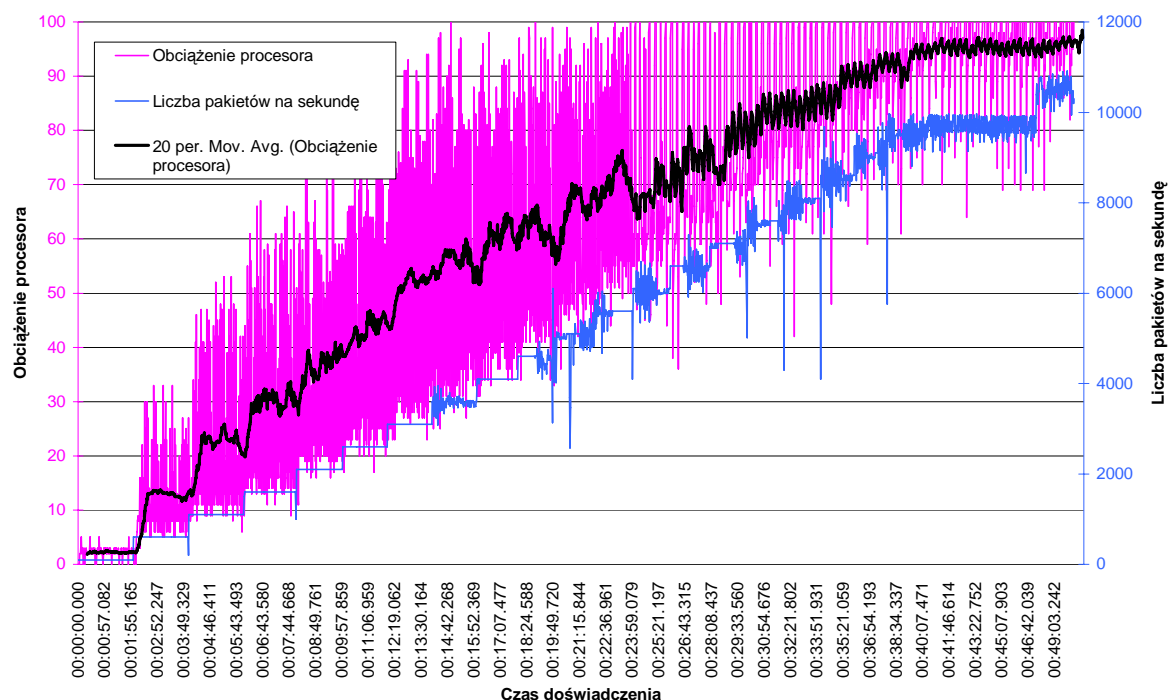


**Rysunek 20. Atak SYN Flood na port 5555 – protokół IPv6.**

Sytuacja wyglądała podobnie również i w tym przypadku. Z analizy wyników wynika, że czas *timeout* dla protokołu IPv6 jest dłuższy.

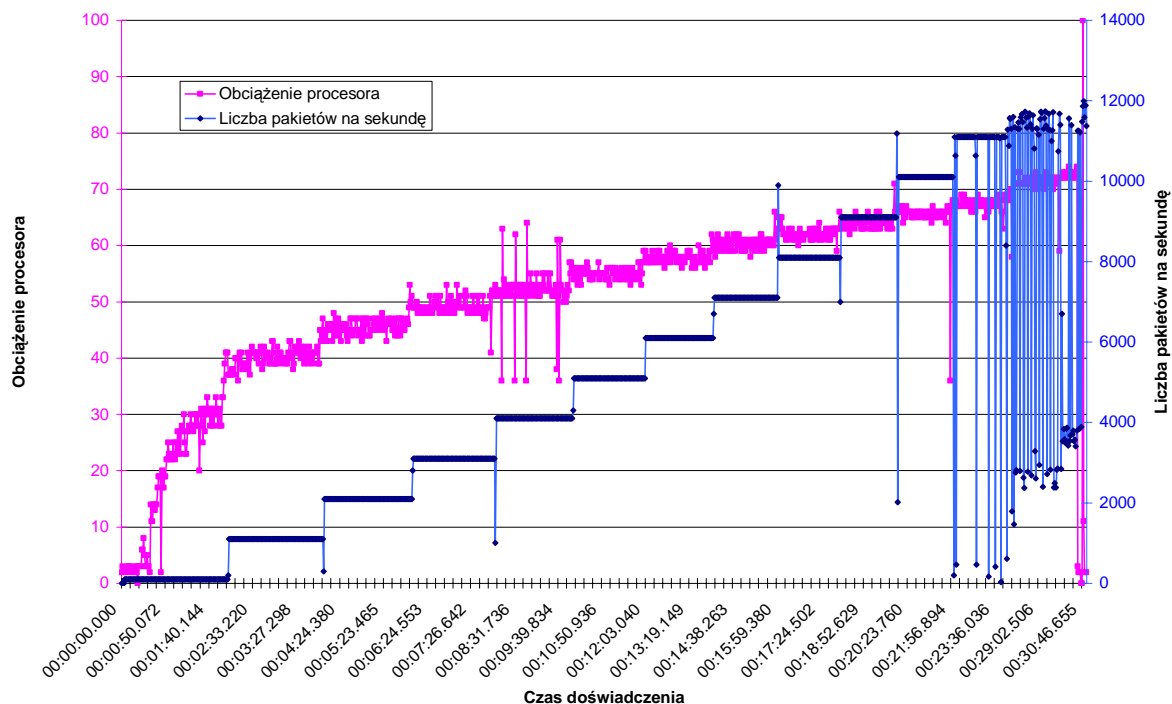
Przy badaniu czasu konwersacji dokonano dwóch interesujących spostrzeżeń:

- Wystarczy wysłać pakiet z ustawioną flagą SYN, aby system wywnioskował, że nastąpiło połączenie – funkcja *accept* zwraca identyfikator podłączonego gniazda. W przypadku, kiedy odebrany pakiet jest fałszywy, następuje próba wysłania i odebrania danych przez program *SocketListener*. Oczywiście próba ta kończy się niepowodzeniem.
- Tylko procesy pracujące w przestrzeni użytkownika mają limit półotwartych połączeń. Domyślnie otwarty port 139 [ISS-3], otwarty jest przez proces System (można to sprawdzić programem *Process Explorer* [SIS-1]), który nie posiada takich ograniczeń. Pozwoliło to wykonać kolejny test – a mianowicie test obciążenia procesora w zależności od liczby wysyłanych pakietów (rys. 21).

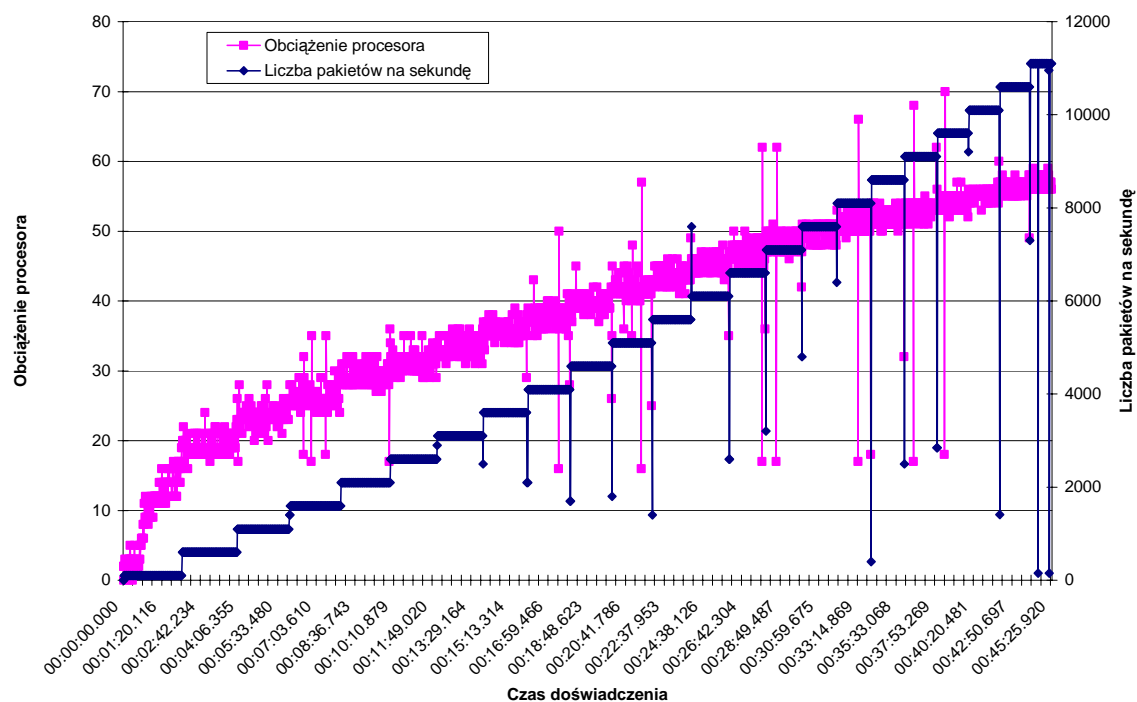


**Rysunek 21. SYN Flood na otwarty port 139 - IPv4.**

Z wykresu (rys. 21) wywnioskować można, że system Windows XP w domyślnej konfiguracji jest podatny na atak DDoS. Zalewanie pakietami zamkniętego portu IPv4 nie spowodowało wzrostu zajętości pamięci operacyjnej. W przypadku powodzi pakietów na port IPv6 sytuacja jest odmienna, ponieważ obciążenie procesora zależy wyłącznie od natężenia ruchu, a różnice w atakach na port otwarty i zamknięty nie są duże (rys. 21 oraz 22).

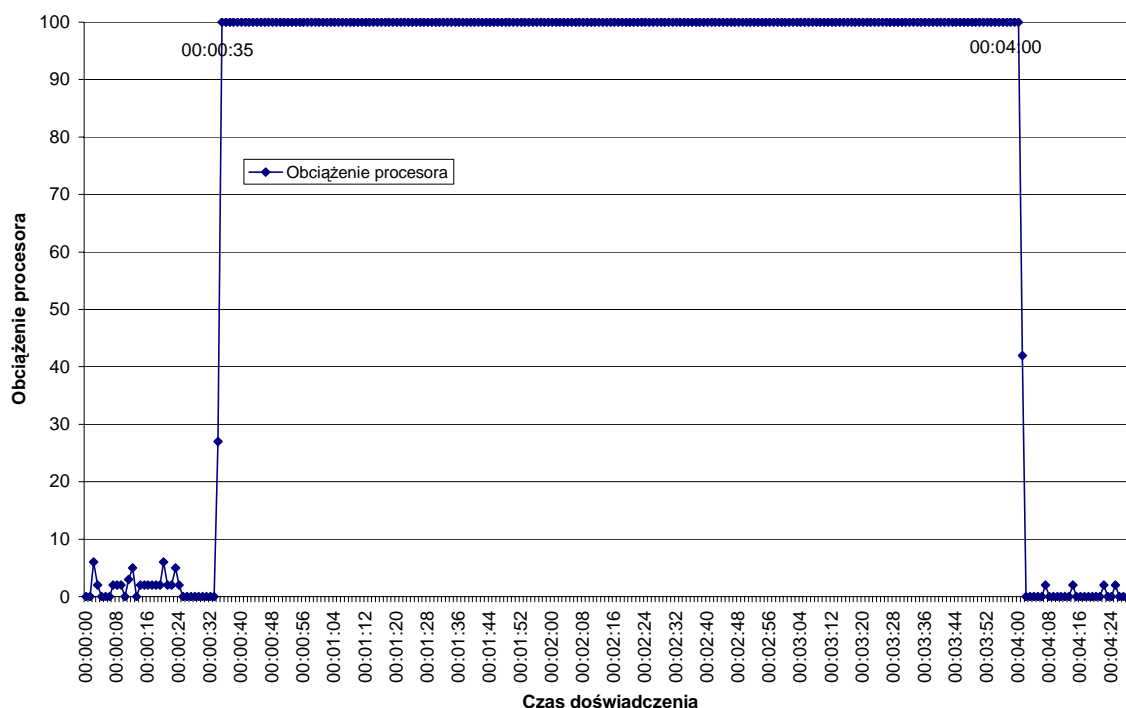


Rysunek 22. SYN Flood na zamknięty port IPv6.



Rysunek 23. SYN Flood na otwarty port IPv6.

Następnym krokiem było sprawdzenie podatkości tego systemu na atak land. Okazało się, że atak ten skuteczny jest wyłącznie w przypadku wykorzystania protokołu IPv6 (rys. 24).



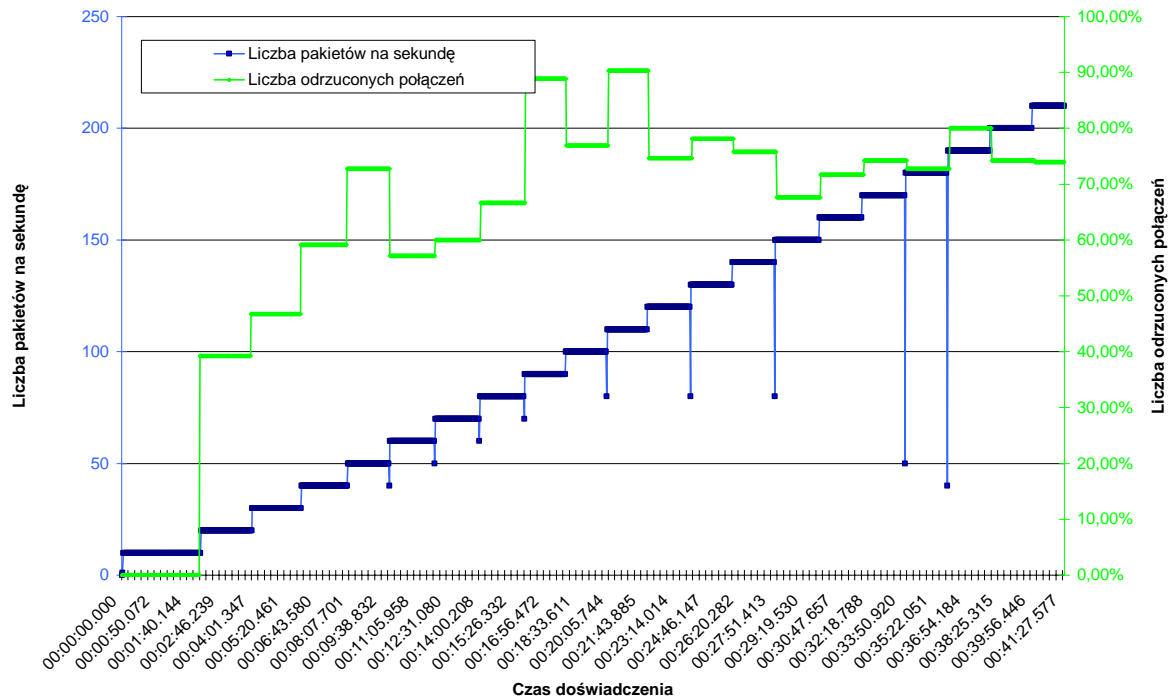
**Rysunek 24. Efekt ataku Land z użyciem IPv6.**

Wysłanie jednego pakietu na otwarty port spowodowało zawieszenie komputera na 3 minuty i 25 sekund. Pożądany efekt uzyskuje się natychmiast po odebraniu przez ofiarę feralnego pakietu. Eksperyment został przeprowadzony dla różnych wartości Hop Limit, jednakże zawsze czas efektu ataku Land był taki sam. Ponadto dało się zaobserwować jeszcze jeden fakt, a mianowicie program *SocketListener* zasygnalizował połączenie z lokalnej maszyny, ale dopiero wtedy, gdy obciążenie procesora spadło. Do nawiązanego połączenia nie dało się ani wysłać ani odebrać danych.

### 6.3.1.2 System ze wszystkimi uaktualnieniami do dnia 20 maja 2005

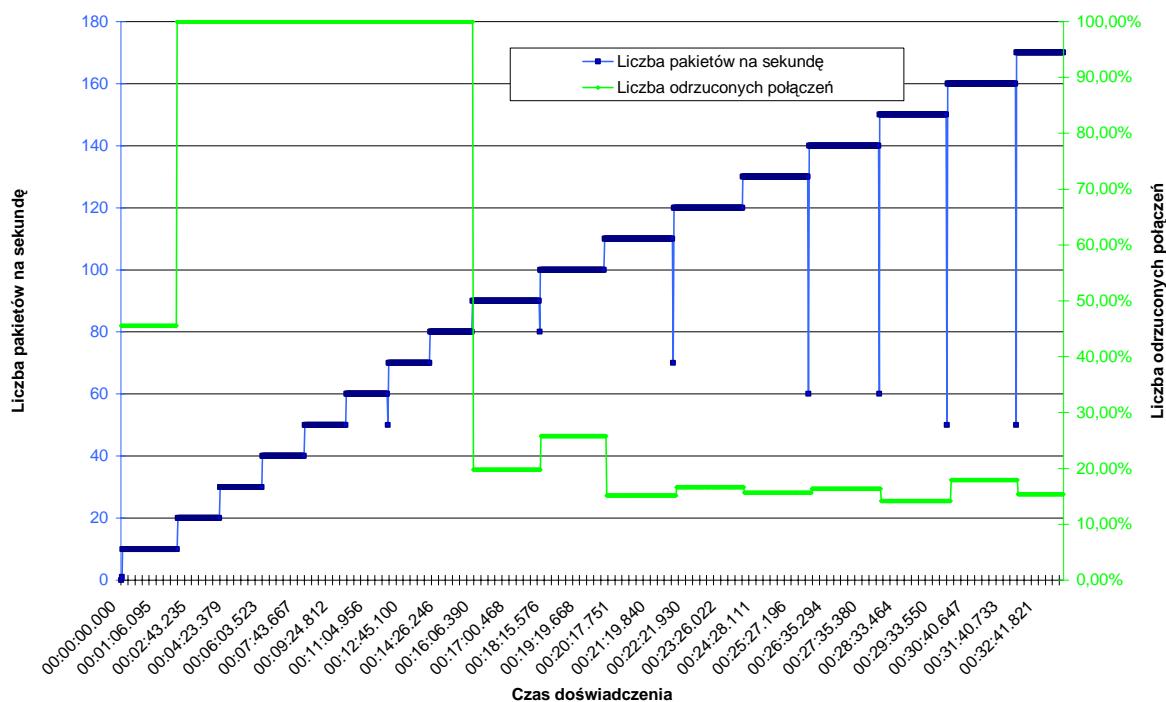
Uaktualnienia pobrane zostały przez udostępniony przez Microsoft mechanizm uaktualnień dostępny na witrynie [windowsupdate.microsoft.com](http://windowsupdate.microsoft.com). W skład uaktualnień wchodził Windows service pack 2 oraz uaktualnienia krytyczne do dnia 20 maja 2005r.

Nowe uaktualnienia zawierały dodatek w postaci ściany ogniowej, która została zintegrowana z systemem. Dla potrzeb testu została ona wyłączona, ponieważ jest ona wyłącznie statycznym filtrem pakietów i nie posiada mechanizmów ochronnych (np. takich jak obecne w ścianach ogniowych firmy Sygate systemy detekcji włamań).



**Rysunek 25 Atak SYN Flood na port IPv4.**

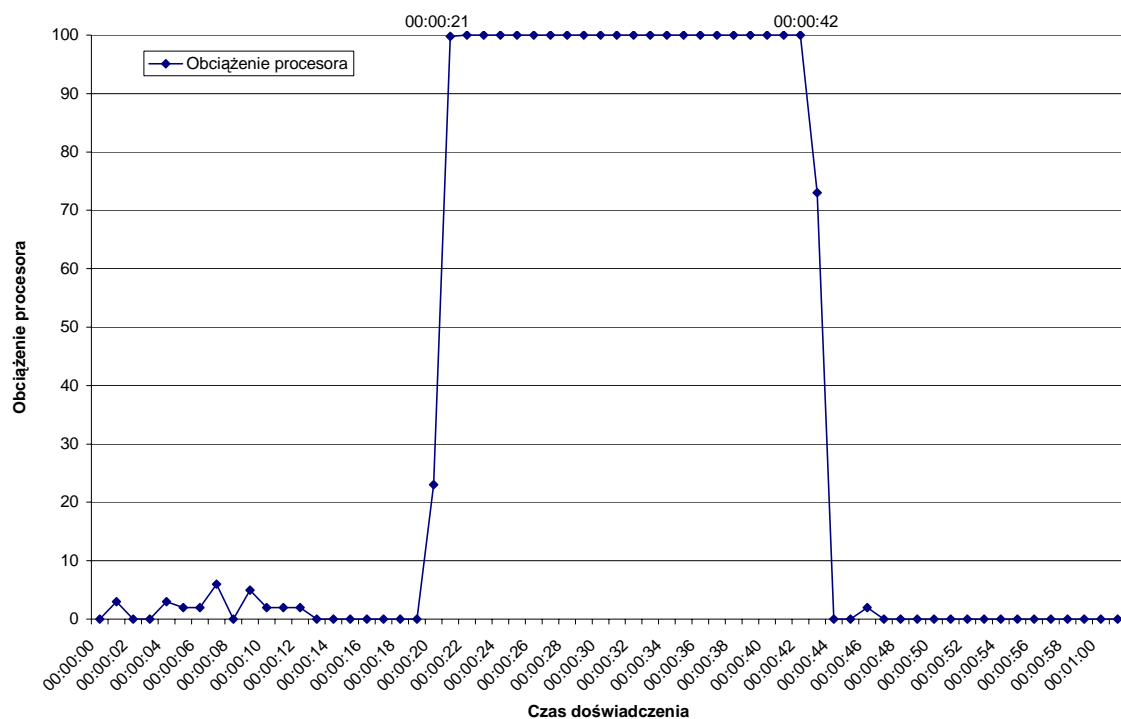
Z wykresu (rys. 25) wywnioskować można, że w porównaniu do stanu sprzed uaktualnienia - stos TCP/IP został wzmocniony. W tym przypadku, aby uzyskać efekt odrzucenia połączenia, należało użyć większego przyrostu ilości pakietów na sekundę, co w tym teście zostało ustalone na 10 pakietów na 120 sekund.



**Rysunek 26. Atak SYN Flood na port IPv6.**

Zachowanie się systemu zaatakowanego z wykorzystaniem protokołu IPv6 (rys. 26) jest ciekawe ze względu na początkowe „zatkanie” się kolejki połączeń oczekujących i „odetkanie” się jej przy pewnej ilości połączeń na sekundę. Sytuacja ta wymaga dodatkowych badań, których rozmiar wykracza poza granice niniejszego dokumentu.

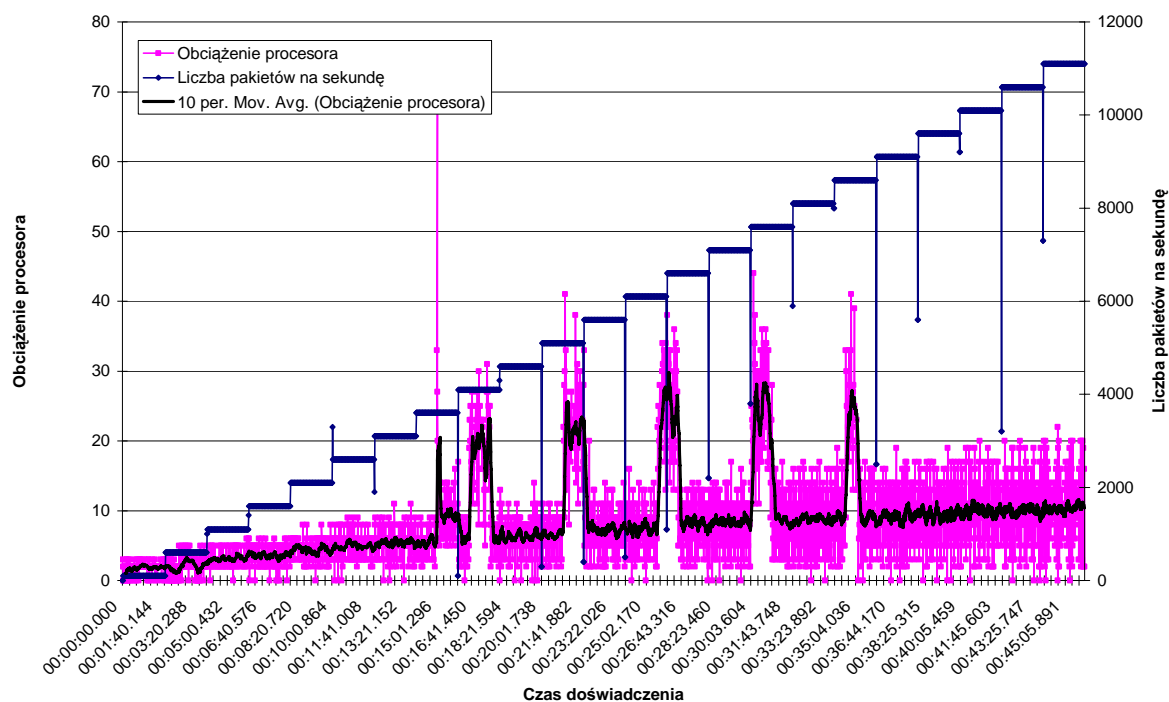
Kolejnym testem było sprawdzenie podatności systemu na atak Land (rys. 27). Okazało się, że system w dalszym ciągu jest podatny na ten atak. Jediną różnicą w stosunku do systemu bez uaktualnień był czas efektu ataku, który skrócił się do 21 sekund. Problem ten został zgłoszony przeze mnie do firmy Microsoft oraz na międzynarodową listę *bugtraq* (nadano temu problemowi bugtraqid=13658, a opis dostępny jest pod adresem <http://www.derkeiler.com/Mailing-Lists/NT-Bugtraq/2005-05/0006.html>).



**Rysunek 27 Efekt ataku Land z użyciem IPv6.**

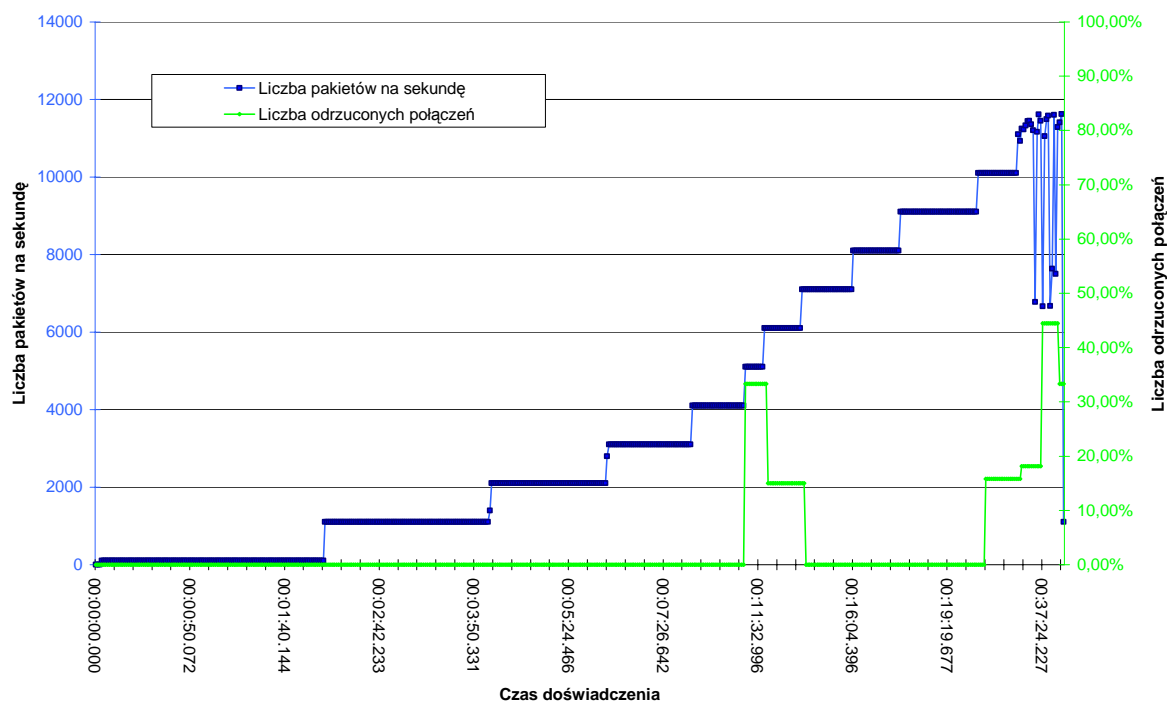
Protokół IPv4 jest zabezpieczony przed tym typem ataku. Wysłanie trefnego pakietu nie było ignorowane przez stos TCP/IP.

Systemy operacyjne rodziny Windows mają wbudowany mechanizm zabezpieczający przed atakami SYN Flood. Po włączeniu tego mechanizmu wszystkie połączenia IPv4 zakończyły się powodzeniem. Jednakże czynność ta zwiększyła obciążenie procesora (rys. 28).



**Rysunek 28. Obciążenie CPU przy ataku SYN Flood IPv4 na system zabezpieczony mechanizmem SynAttackProtect.**

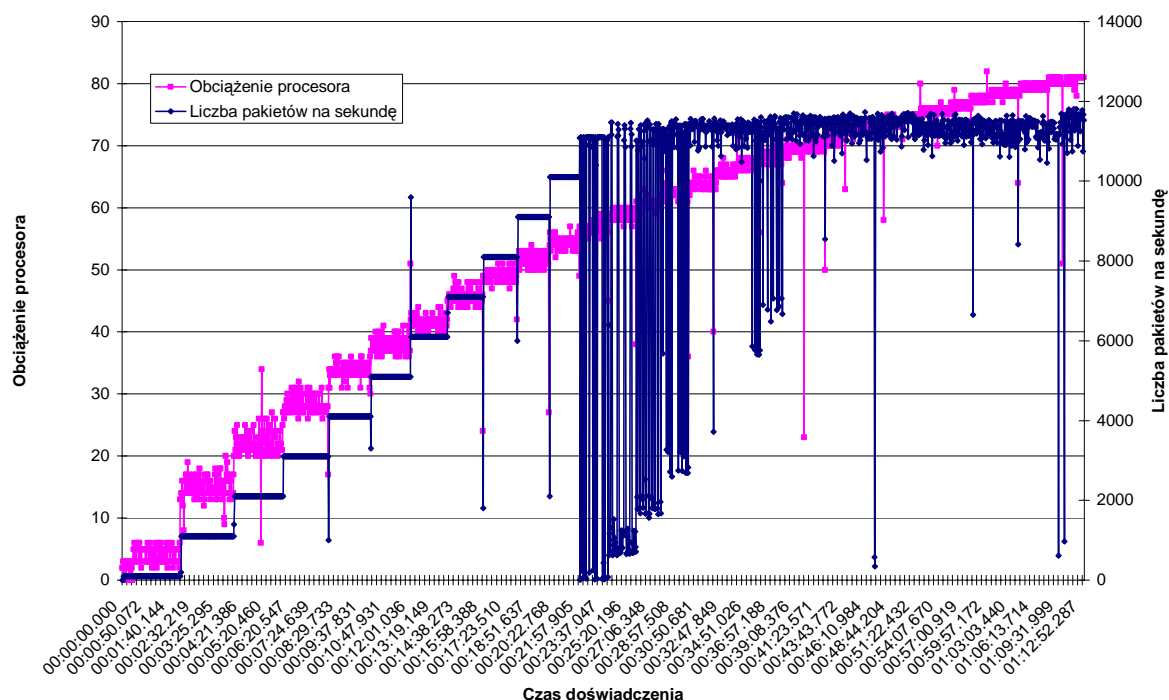
Atak na protokół IPv6 pokazał, że mechanizm SynAttackProtect w systemie Windows XP wymaga jeszcze wielu poprawek usprawniających zarządzanie połączeniami (rys. 29).



**Rysunek 29. Atak SYN Flood IPv6 na otwarty port w systemie zabezpieczonym mechanizmem SynAttackProtect.**



Podobnie, jak w systemie z wyłączonym mechanizmem SynAttackProtect, atak powoduje wzrost obciążenia procesora (rys. 30).



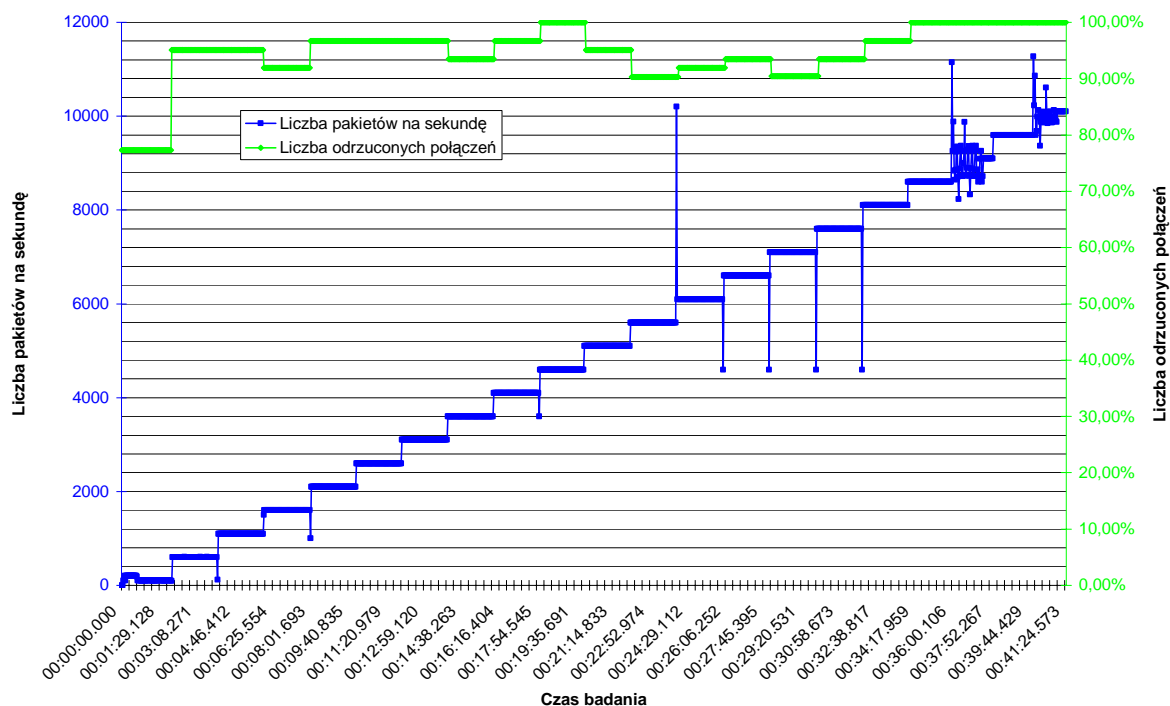
Rysunek 30. Obciążenie CPU przy ataku SYN Flood IPv6 na system zabezpieczony mechanizmem SynAttackProtect.

### 6.3.2 System „Windows 2003”

„Rodzina produktów Windows Server 2003 zamyka kolejny etap ewolucji serwerowych systemów operacyjnych Windows. System Windows Server 2003 został opracowany na podstawie systemu Windows 2000 Server, którego niezawodność, skalowalność i funkcjonalność została wielokrotnie potwierdzona w praktyce. W wyniku wprowadzenia dodatkowych funkcji otrzymano bardzo wydajną platformę infrastrukturalną wspomagającą zarządzanie sieciami, aplikacjami sieci Web i usługami sieciowymi XML — od grupy roboczej do centrum danych.” [WMC-1].

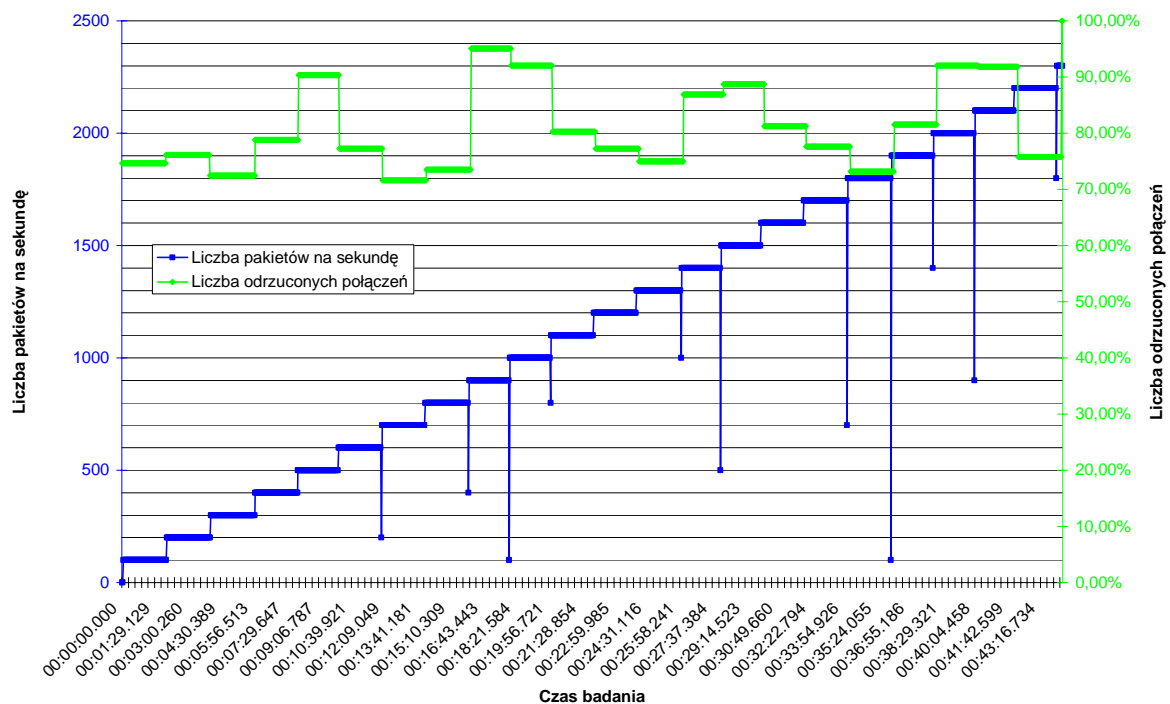
#### 6.3.2.1 System niezabezpieczony

Pierwszym testem, któremu został poddany flagowy produkt firmy Microsoft był test ilości odrzuconych połączeń (rys. 31).



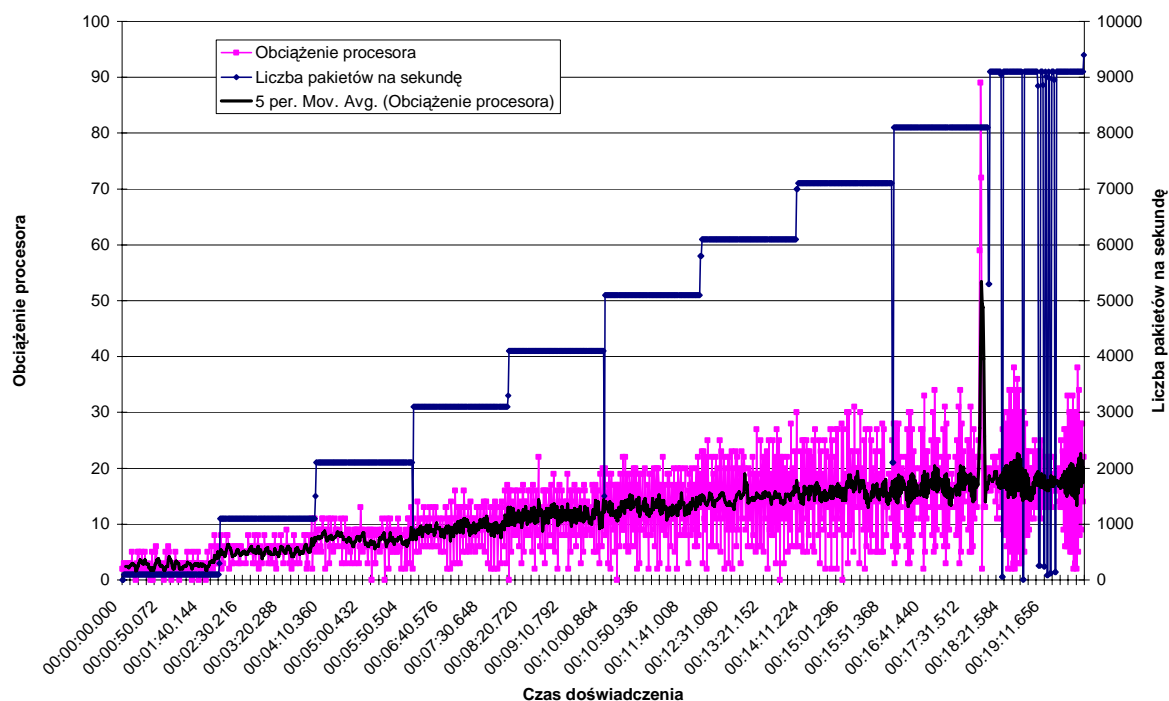
**Rysunek 31. Atak SYN Flood na otwarty port IPv4.**

Należy zauważyć, że Microsoft zrobił sporo w zakresie dostępności połączenia w porównaniu z Windows XP. Liczba pakietów potrzebnych do „zatkania” połączenia zwiększyła się o co najmniej jeden rząd. Jednakże system ten nadal jest podatny na ataki zalewania pakietami.



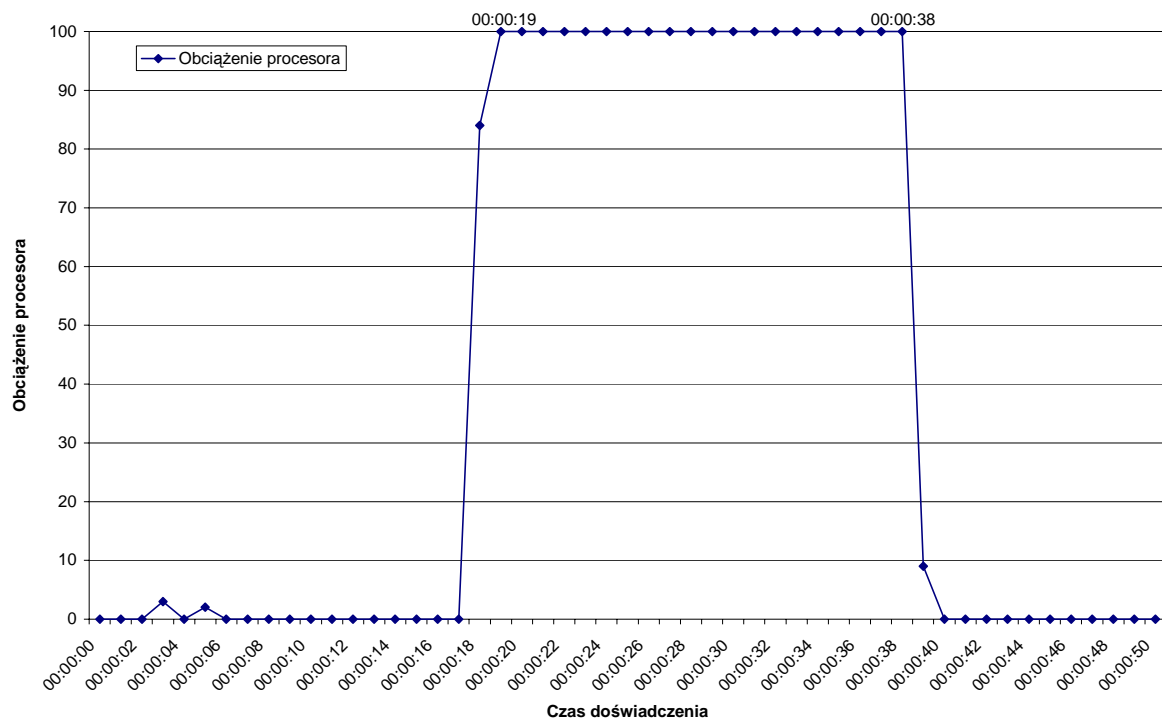
**Rysunek 32. Atak SYN Flood na otwarty port IPv6.**

Również w tym wypadku lepiej prezentuje się wykres pokazujący obciążenie procesora w zależności od natężenia ilości pakietów (rys. 33).

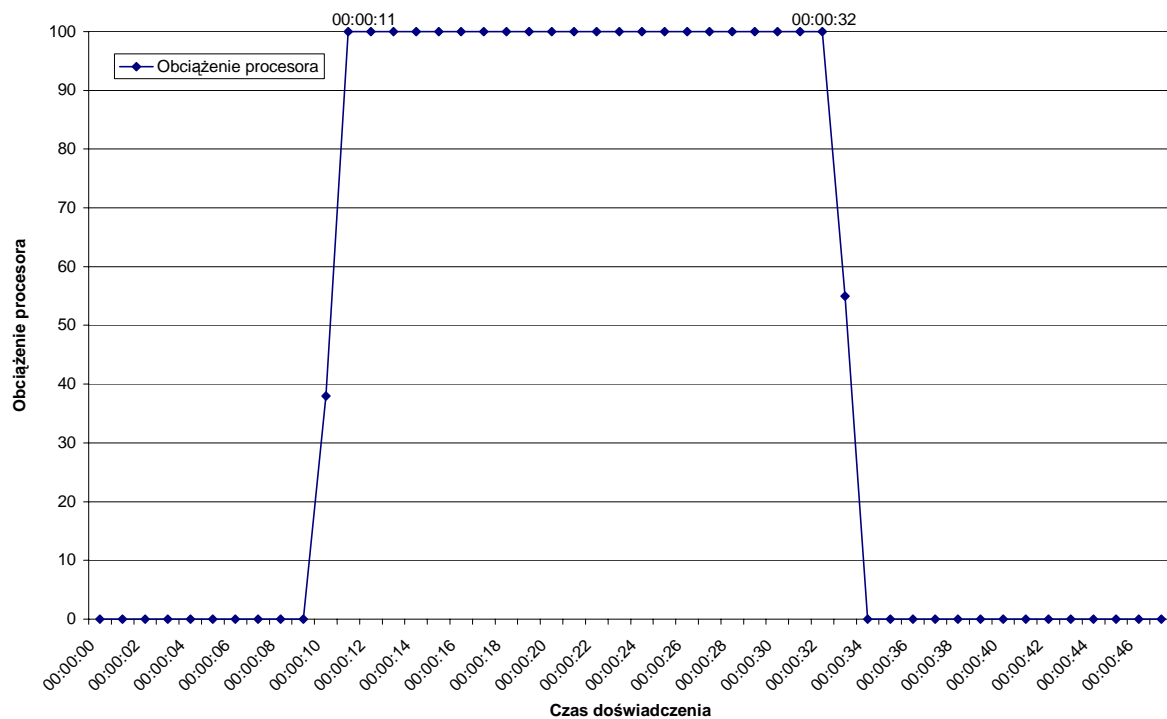


**Rysunek 33. Obciążenie procesora w zależności od ilości pakietów na sekundę – IPv6.**

Atak „Land” działał zarówno w IPv4 jak i IPv6 (rys 34, 35). W porównaniu do systemu Windows XP, skutki ataku uległy skróceniu do około 20 sekund.



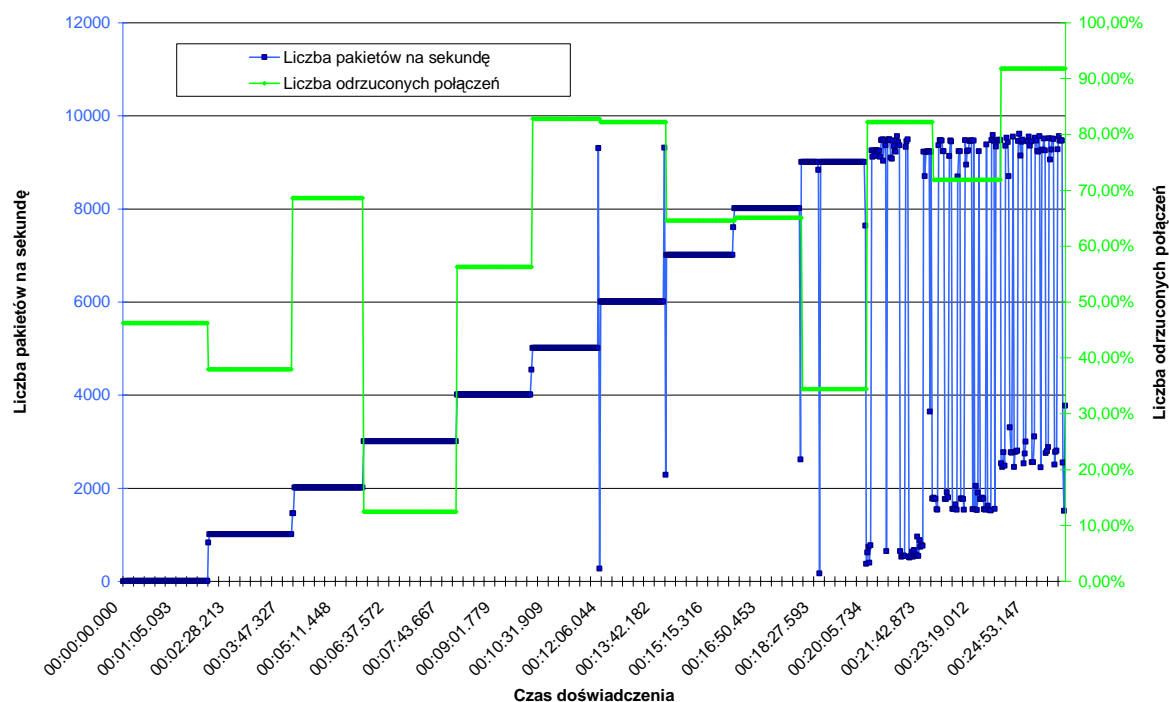
**Rysunek 34. Obciążenie procesora systemu zaatakowanego poprzez Land-IPv4.**



**Rysunek 35. Obciążenie procesora systemu zaatakowanego poprzez Land-IPv6.**

### 6.3.2.2 System zabezpieczony uaktualnieniami do 20 maja 2005

Przy wyłączonym mechanizmie SynAttackProtect atak SYN Flood na port TCP, używając protokołu IPv4, nie powiódł się w granicach natężenia ruchu umożliwianych przez program *DDoS Generator*. Natomiast atak z użyciem IPv6 zakończył się sukcesem (rys. 36).

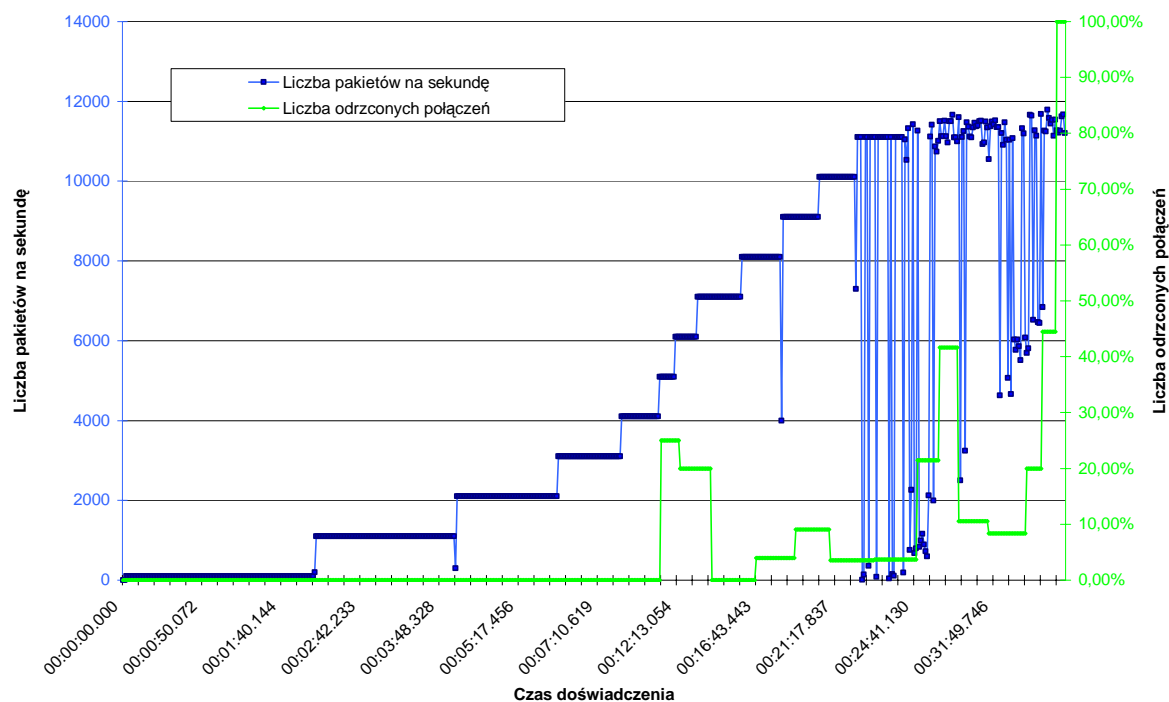


Rysunek 36. Atak SYN Flood na otwarty port IPv6.

Rozpatrując pod kątem zużycia procesora, nie udało się uzyskać efektu DoS dla dostępnego natężenia ruchu. Zalewanie zamkniętego portu IPv4 nie dało w ogóle żadnego efektu. Zalewanie otwartego portu, spowodowało wzrost zużycia procesora do około 10% . Ten sam typ ataku przeprowadzony z użyciem IPv6 powodował obciążenie procesora w granicy 20% dla portu zamkniętego oraz 30% dla otwartego.

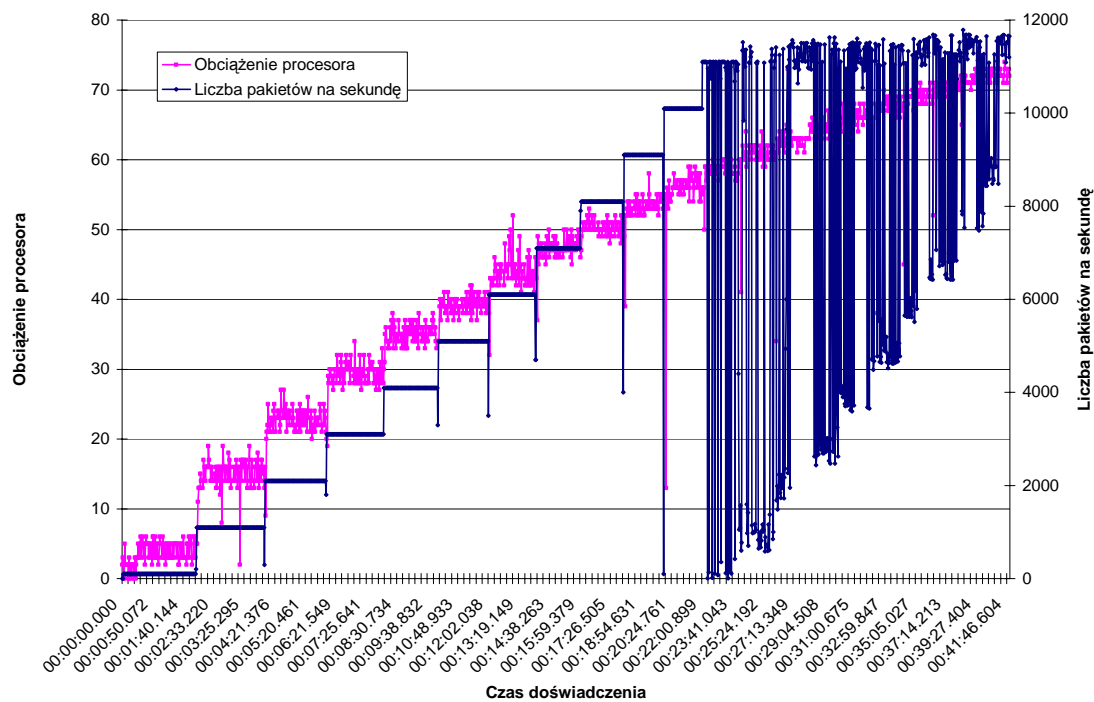
Atak Land powodował efekt DoS tylko w przypadku wykonywania go za pomocą protokołu IPv6. Stąd nasuwa się wniosek, że uaktualnienia Microsoftu dotyczyły jedynie protokołu IP w wersji 6, oraz iż mechanizmy obsługi TCP oraz IP mogą być oddzielne.

Po włączeniu mechanizmu SynAttackProtect sytuacja uległa poprawie, rozpatrując ją pod kątem ilości prawidłowo obsługiwanych połączeń przy użyciu IPv6 (rys. 37).



**Rysunek 37. Atak SYN Flood na otwarty port IPv6.**

Zmiana tego ustawienia spowodowała również wzrost zapotrzebowania na czas procesora (rys. 38).



**Rysunek 38. Obciążenie procesora w zależności od ilości pakietów na sekundę IPv6.**

Zalewanie pakietami zamkniętego portu IPv6 nie spowodowało wzrostu zużycia procesora. Dodatkową ciekawostką jest to, że włączenie mechanizmu SynAttackProtect powoduje, iż połączenie nie jest uznawane przez system za nawiązane w momencie nadejścia pierwszego pakietu (z ustawioną flagą SYN).

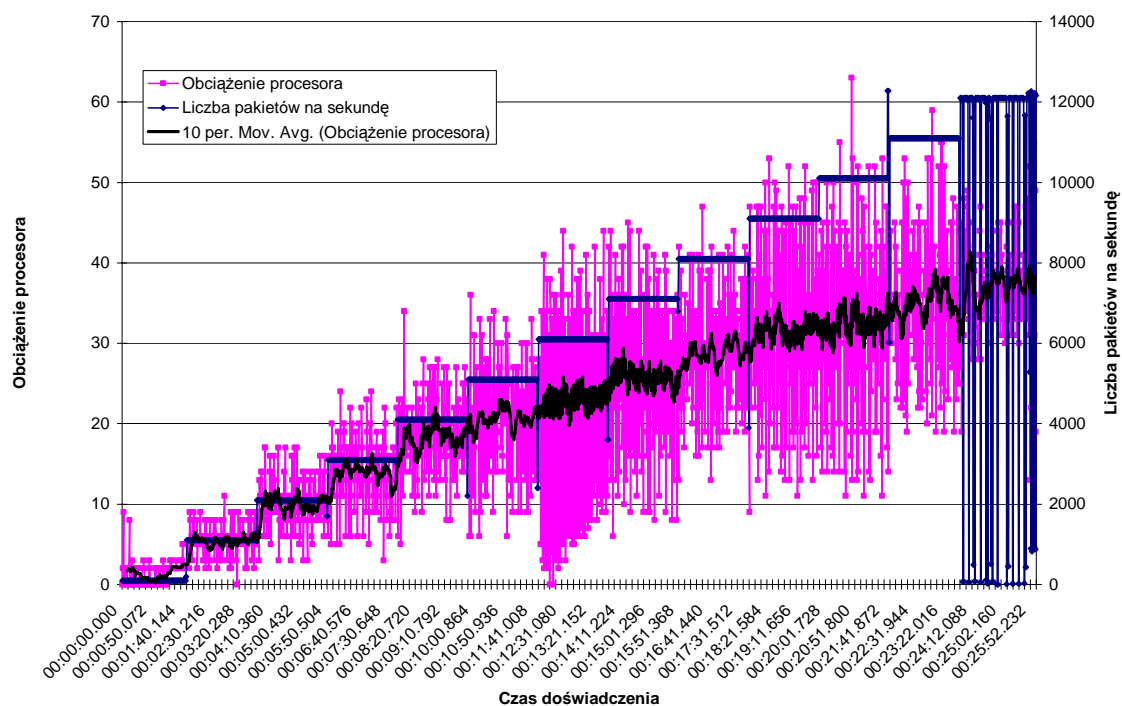
Mechanizm SynAttackProtect spowodował wzrost zużycia procesora przy ataku SYN Flood o około 4% w porównaniu z sytuacją, kiedy zabezpieczenie nie było włączone.

Włączenie SynAttackProtect nie zabezpiecza przed atakiem LAND. Tak jak w przypadku wyłączonego zabezpieczenia, Land działa wyłącznie w momencie użycia IPv6, a jego skutki są identyczne – zawieszenie komputera na około 20 sekund.

### **6.3.3 System „Windows Longhorn build 5048”**

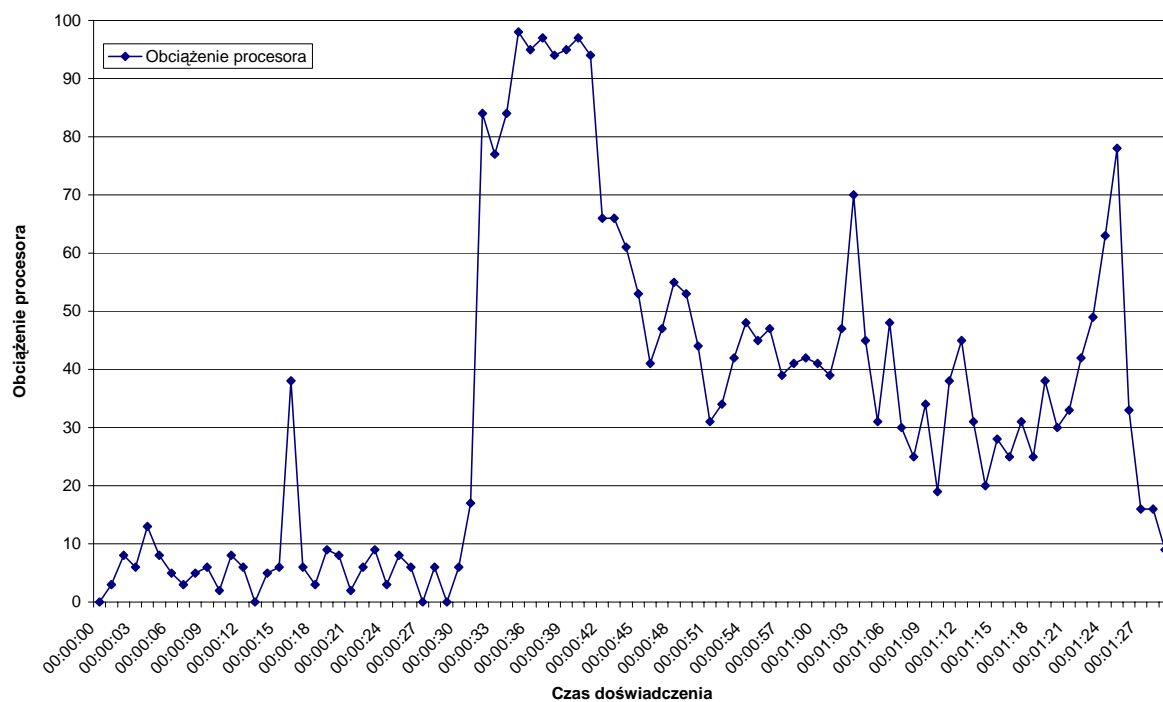
System Windows Longhorn będzie najnowszym produktem firmy Microsoft. Data jego premiery nie została jeszcze dokładnie ustalona, a wersja, która została udostępniona beta-testerom przez firmę Microsoft, pozostaje ciągle jeszcze fazie testów. Jednakże pomimo tego, wydaje się, że odporność tego systemu na ataki SYN Flood jest inna niż wszystkich do tej pory przetestowanych systemów.

Atak SYN Flood nie powiódł się. Zalewanie pakietami w granicach umożliwionych przez biblioteki WinPCap nie przyniosło odrzucenia połączenia zarówno przy wykorzystaniu IPv4, jak i IPv6. Inaczej było z obciążeniem procesora. SYN Flood z maksymalną prędkością na otwarty port ipv4 powoduje wzrost użycia procesora do 90%, a na zamknięty port około 20%. Ten sam atak na otwarty port IPv6 powoduje obciążenie procesora w granicach 40%, a na zamknięty port IPv6 zapotrzebowanie wynosi 20%.



**Rysunek 39. Obciążenie procesora podczas ataku SYN Flood na otwarty port - IPv6.**

Ciekawy efekt powodowało wysyłanie pakietów SYN z maksymalną prędkością (około 12000) na otwarty port 139 (rys. 40). Powodowało to wzrost zużycia procesora do 90%, a następnie spadek do 40%.



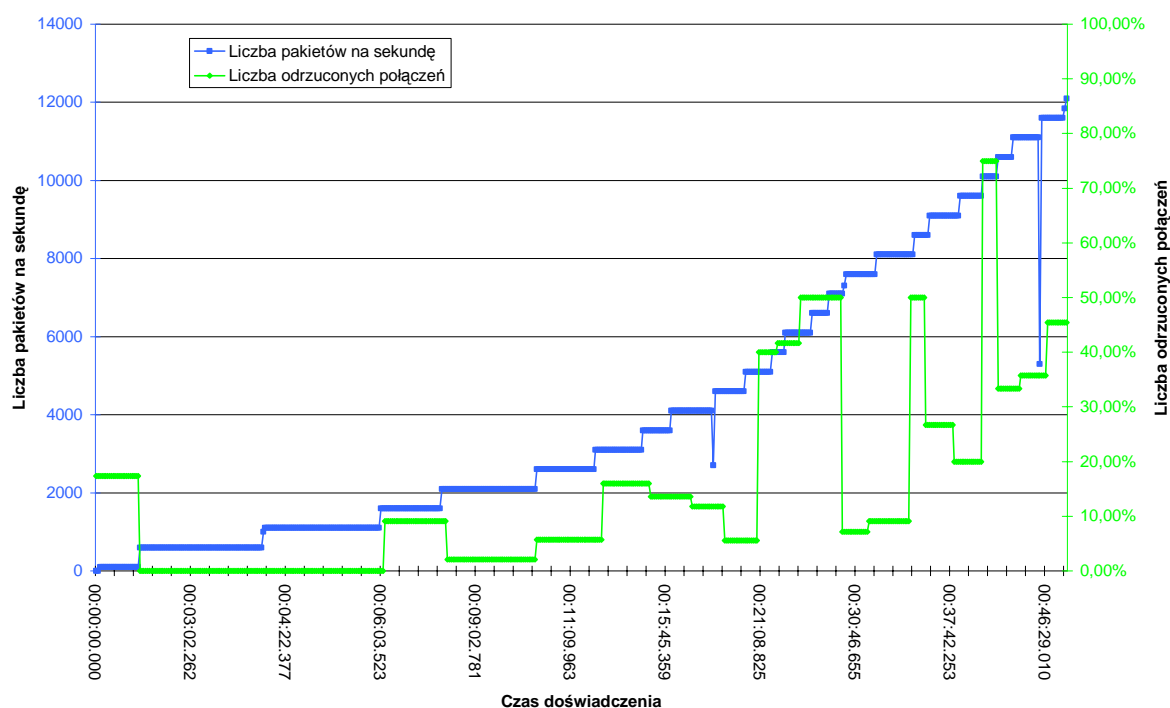
**Rysunek 40. Nietypowe zachowanie stosu TCP/IP przy maksymalnym natężeniu pakietów.**



### 6.3.4 System „Linux – debian”

System linux uważany jest za jedną z najstabilniejszych platform w zastosowaniach sieciowych. Jest on systemem na tyle dojrzałym, że jego zastosowanie nie sprowadza się wyłącznie do użytku w środowiskach akademickich, a popularny staje się także wśród tak użytkowników prywatnych, jak i w businessie. W badaniach wykorzystano dystrybucję debian-sarge, która cieszy się sławą stabilnego i wydajnego systemu. Testy zostały przeprowadzone na linuksie z jądrem w wersji „kernel-image-2.6.8-2-k7” ściągniętym z packages.debian.org.

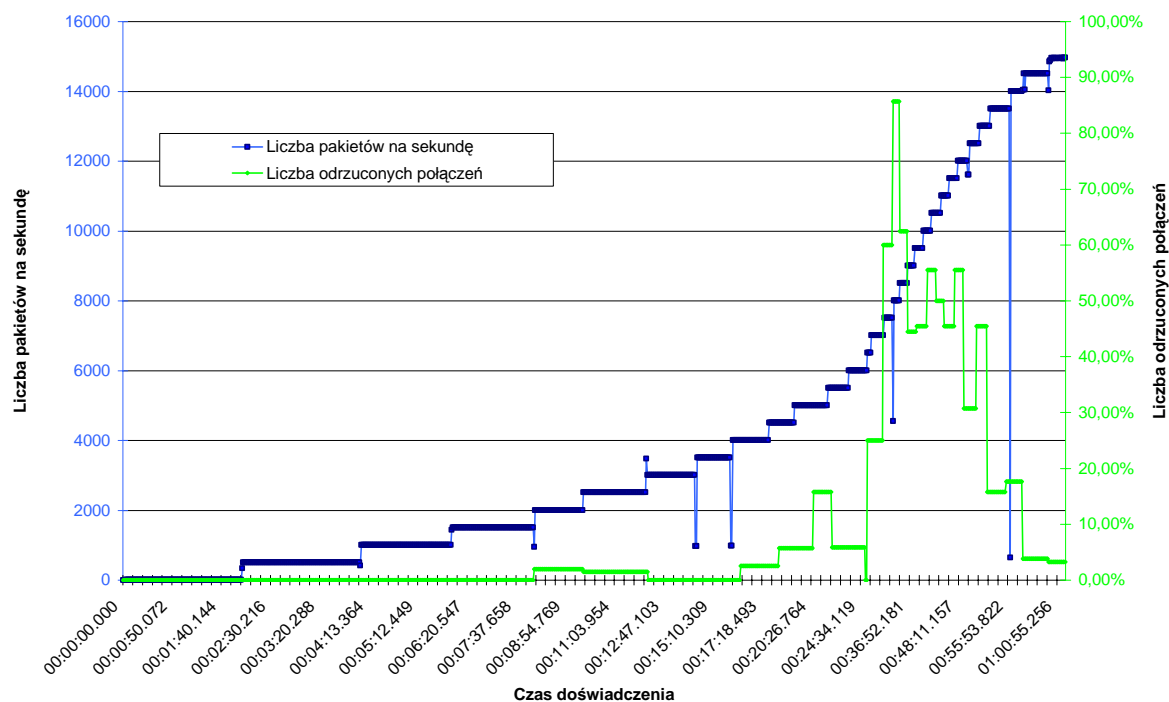
Pierwszym testem, który został przeprowadzony na tym systemie, był test odporności systemu na atak SYN Flood (rys. 41).



Rysunek 41. Atak SYN Flood na otwarty port IPv4.

Atak ten zakończył się sukcesem – system zaczął odrzucać połączenia przy stosunkowo niskim natężeniu pakietów. Po włączeniu mechanizmu *tcp\_syncookie* sytuacja uległa poprawie i system nie odrzucił ani jednego połączenia.

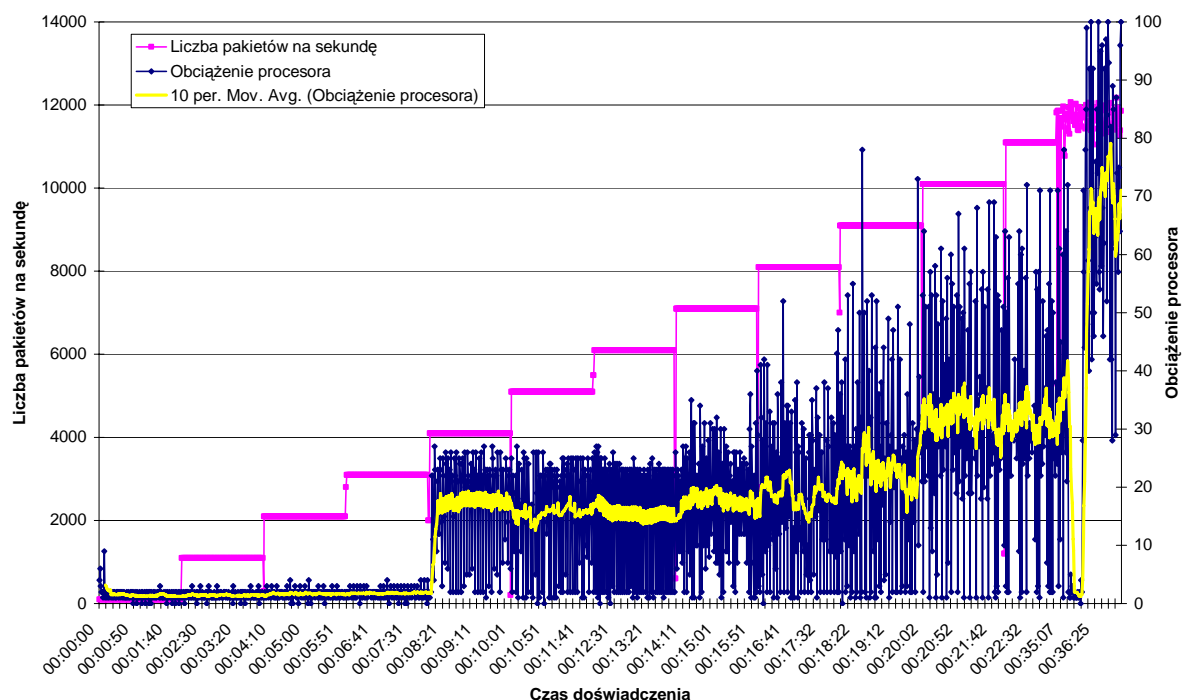
Podobne wyniki przyniósł test skuteczności ataku z wykorzystaniem IPv6 (rys. 42).



**Rysunek 42. Atak SYN Flood na otwarty port IPv6.**

Ataki Land nie przyniosły efektu DoS, zarówno w wersji IPv4 jak i IPv6.

Test obciążenia procesora w zależności od ilości pakietów wykazał, że jądro linuxa jest podatne na atak SYN Flood IPv6 jedynie wtedy, gdy celem jest zamknięty port TCP. Tylko wówczas można zaobserwować wzrost obciążenia procesora (rys. 43).



Rysunek 43. Obciążenie procesora w zależności od ilości pakietów IPv6.

Atak na otwarty port IPv6 oraz na port w dowolnym stanie IPv4 nie powoduje wzrostu obciążenia procesora.

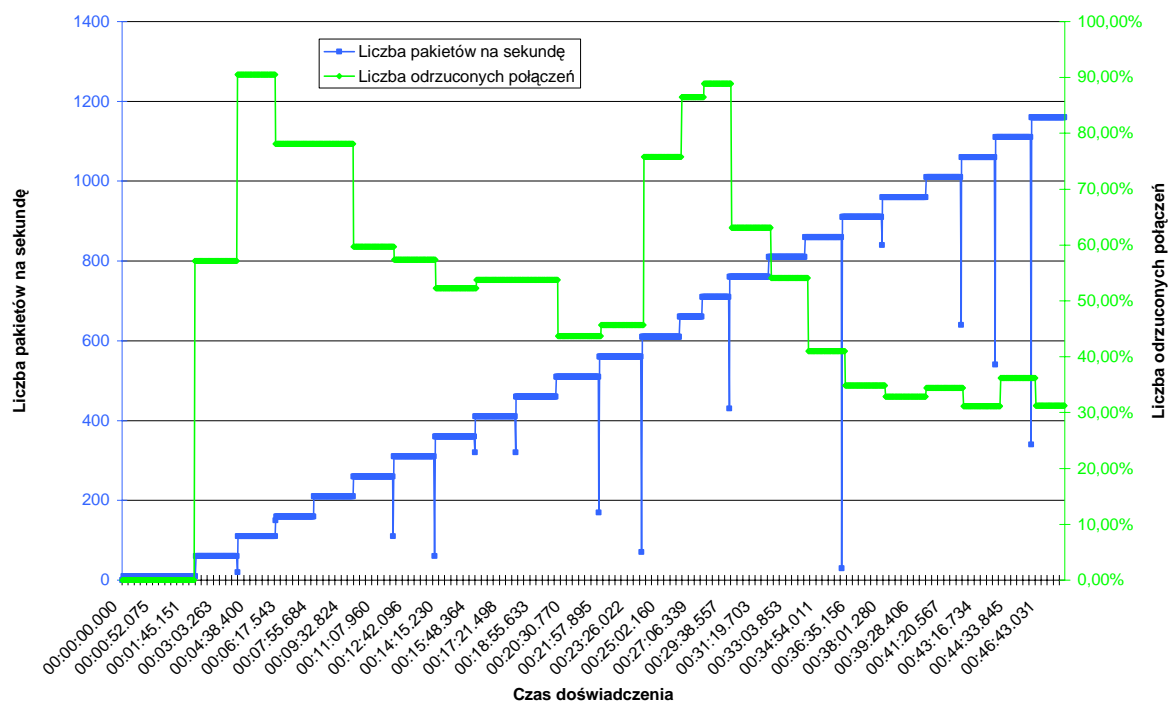
### 6.3.5 Wpływ obecności ściany ogniowej na skuteczność ataku DDoS

Ostatnim doświadczeniem było przeanalizowanie wpływu obecności popularnych ścian ogniowych na skuteczność ataku DDoS. Eksperyment miał na celu zweryfikowanie faktu, iż ściana ogniowa (zwana potocznie *firewallem*), jest skutecznym zabezpieczeniem przeciwko wszelkim atakom sieciowym. Test ten został przeprowadzony na systemie operacyjnym Windows XP wraz z zainstalowanymi najnowszymi uaktualnieniami (do dnia 20 marca 2005r.) oraz wyłączonym mechanizmem *SynAttackProtect*. Na potrzeby testu została również wyłączona ściana ogniowa wbudowana w system. Ataki zostały przeprowadzone wyłącznie z wykorzystaniem IPv4, ponieważ większość współczesnych ścian ogniowych nie wspiera IPv6.

#### 6.3.5.1 Agnitum Outpost Firewall PRO – wersja 2.6.452.403

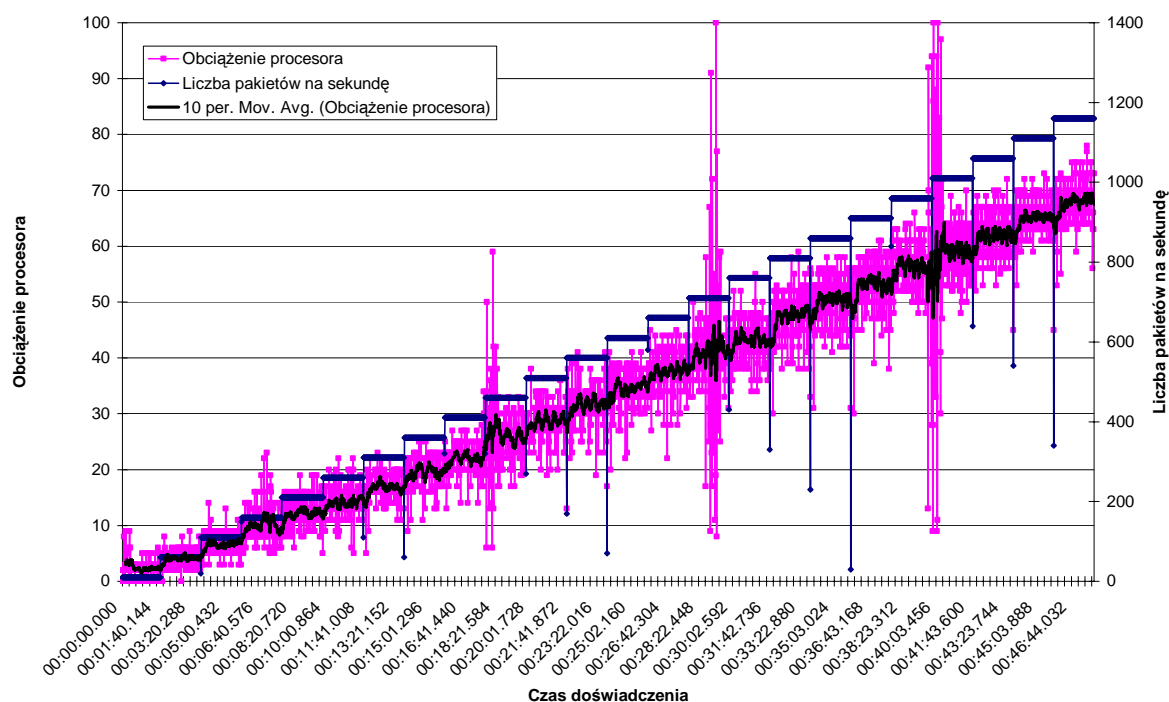
Agnitum Outpost Firewall PRO [WAC-1] jest jedną z najbardziej cenionych ścian ogniowych dostępną na platformie Windows.

Już pierwszy test – skuteczności ataku DDoS – wykazał, że firewall ten nie chroni przed skutkiem rozproszonego ataku sieciowego. Na obronę tego programu należy jednak dodać, że jako jedyny z testowanych wykrył rozproszony atak sieciowy (rys. 44).



**Rysunek 44. Test skuteczności ataku DDoS przy zainstalowanej ścianie ogniowej Agnitum Outpost Firewall.**

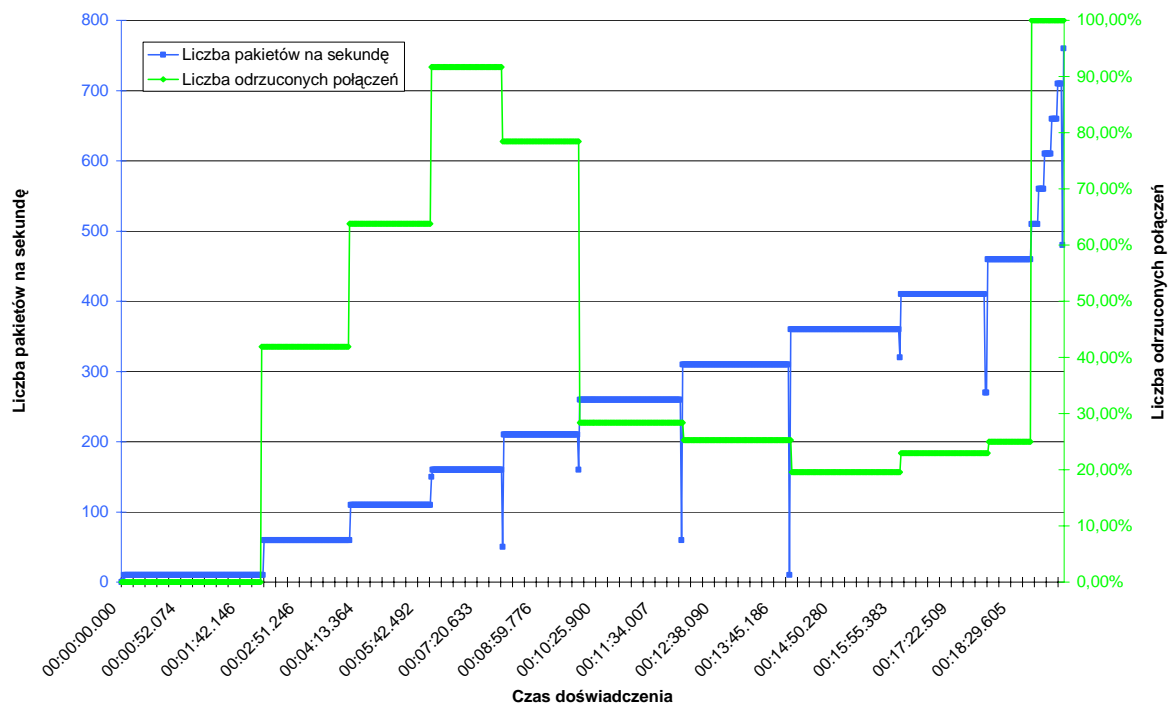
Dodatkowo podczas ataku, został wykonany test obciążenia procesora. Okazało się, że obecność ściany ogniowej zwiększa zapotrzebowanie systemu na czas procesora (rys. 45).



**Rysunek 45. Obciążenie procesora systemu podczas ataku DDoS i zainstalowanej ścianie ogniowej Agnitum Outpost Firewall.**

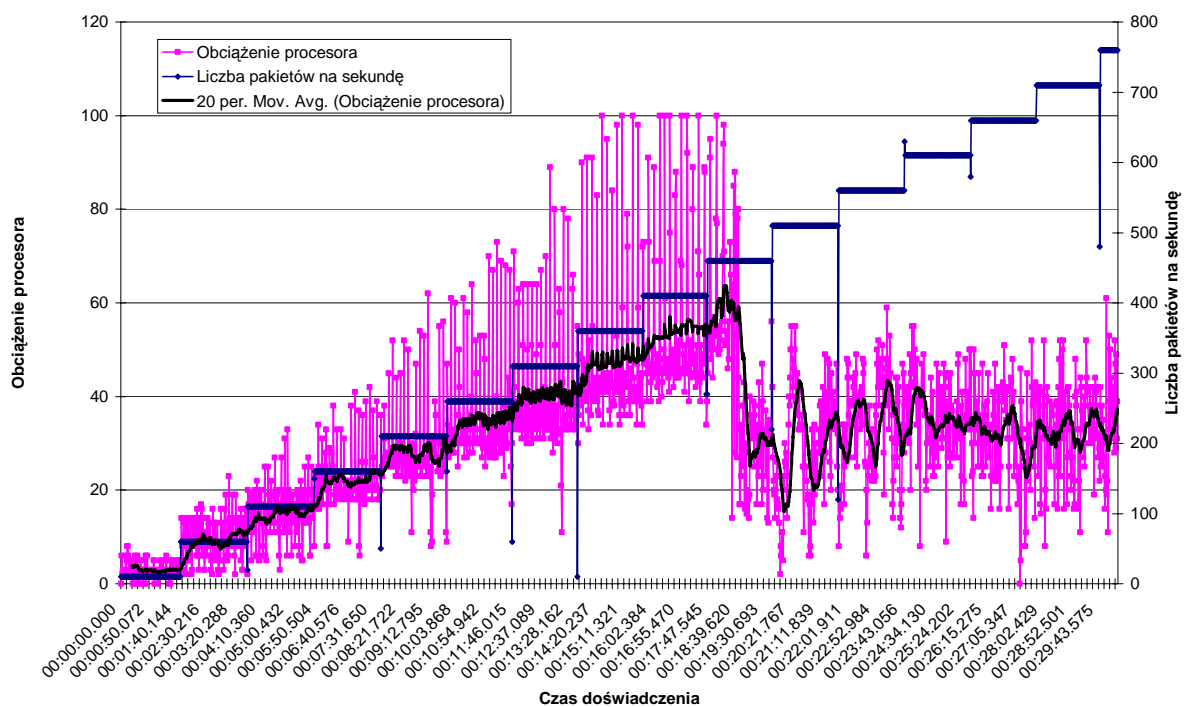
### 6.3.5.2 Sygate Personal Firewall – wersja 5.5.2710.0

Bardzo popularną ścianą ogniową jest również Sygate Personal Firewall [WSC-1], która wyznacza standard w dziedzinie ochrony sieci. Testy wykazały, że program ten również nie jest „panaceum” na ataki sieciowe. Już pierwszy test pokazał, że zaporę sieciową nie niweluje skutków ataku DDoS (rys. 46).



**Rysunek 46. Test skuteczności ataku DDoS przy zainstalowanej ścianie ogniowej Sygate Personal Firewall.**

Dodatkową wadą okazało obciążenie czasu procesora, które pojawiło się wraz z narastaniem natężenia pakietów w łączu (rys. 47).

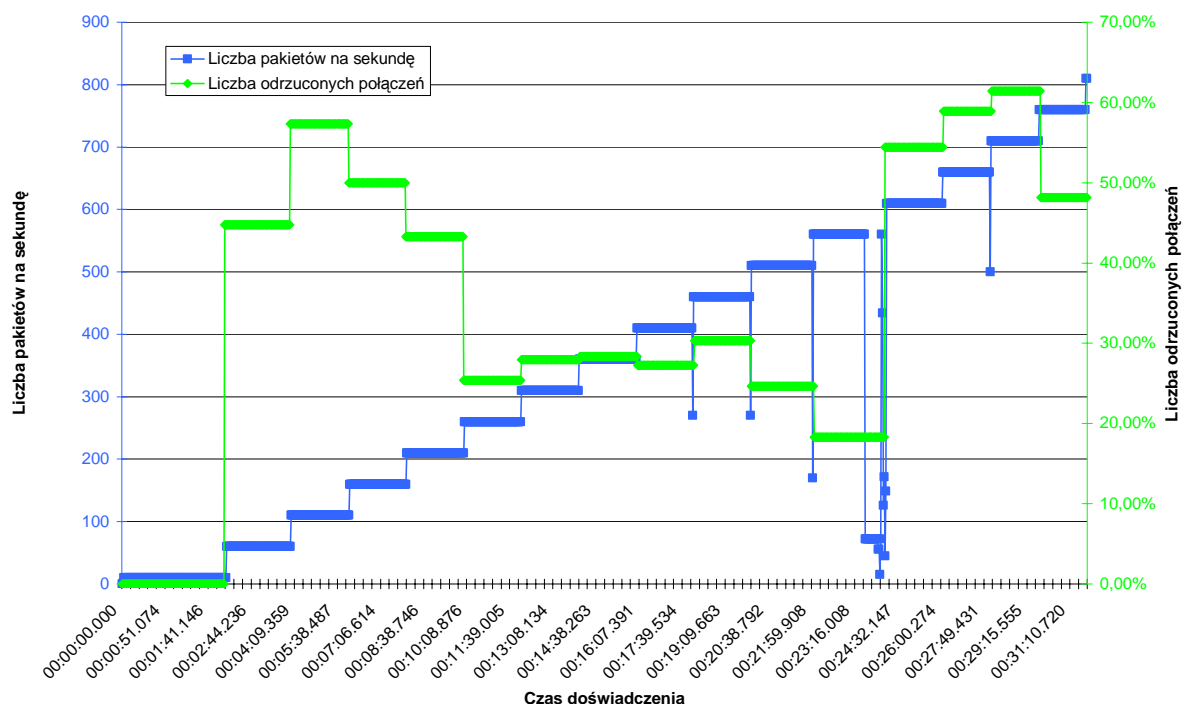


**Rysunek 47. Obciążenie procesora systemu podczas ataku DDoS i zainstalowanej ścianie ogniowej Sygate Personal Firewall.**

Ciekawym zjawiskiem okazał się spadek obciążenia procesora w około 18 minucie doświadczenia. W tym zdalnym czasie, program *ConnectTime* odnotował brak możliwości połączenia ze zdalnym serwerem.

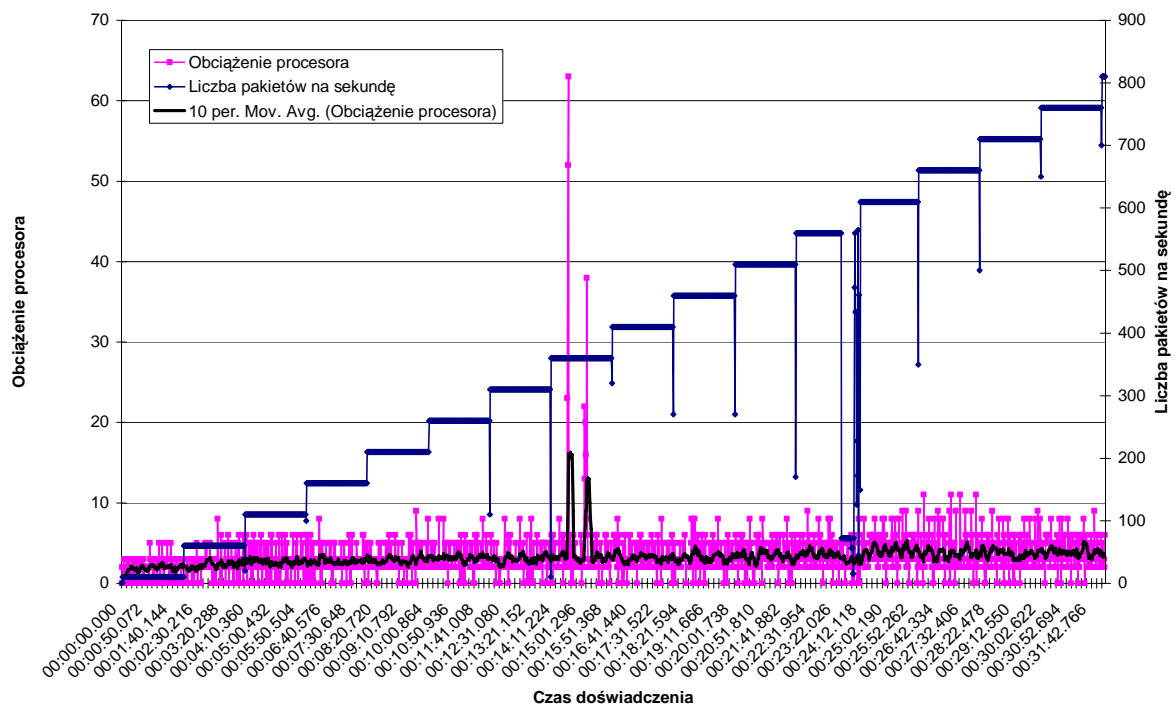
### 6.3.5.3 Zone alarm – wersja 5.5.94.0

Ściana ogniowa firmy Zone Labs [WZC-1] zajęła pierwsze miejsce w teście wykonanym przez autorów strony toptenreviews [PFS-1]. Autorzy testów ocenili produkt pod wieloma względami, jednakże nie przetestowali odporności ściany ogniowej na ataki DDoS. Testy przeprowadzone w ramach niniejszej pracy potwierdziły teorię, mówiącą o nieskuteczności tego typu zabezpieczeń przy atakach rozproszonych (rys. 48).



**Rysunek 48. Test skuteczności ataku DDoS przy zainstalowanej ścianie ogniowej Zone Alarm.**

W odróżnieniu od innych produktów, program firmy Zone Labs nie zużywał tyle czasu procesora co konkurencyjne produkty (rys. 49).



**Rysunek 49. Obciążenie procesora systemu podczas ataku DDoS i zainstalowanej ścianie ogniowej Zone Alarm.**



## 7. Podsumowanie

Ataki DDoS, będą jeszcze z pewnością przez wiele lat spędzać sen z powiek administratorom systemów operacyjnych. Jest wielce prawdopodobne, że problem przybierze na wadze w momencie przejścia ze starego protokołu IPv4 na IPv6, ponieważ jak pokazano w tej pracy, nowy protokół wymaga usprawnienia i dalszego testowania, a wprowadzenie go w jego obecnej postaci, sprawia, że zwiększa się podatność systemu na ataki odmowy usługi. Włamywaczom dodatkowo sprzyja fakt zwiększania przepustowości łącz abonenckich, co ułatwia znacząco przeprowadzenie ataku DDoS i zwielokrotnia jego efekty.

Napisany w ramach niniejszej pracy program *DDoS Generator* pokazał, że narzędzia stosowane przez różnego rodzaju wandalów mogą być bardzo proste w obsłudze i nie wymagać dużej wiedzy z dziedziny sieci komputerowych. Program można wykorzystać nie tylko do przeprowadzania testów w wyizolowanej lokalnej sieci, ale również do badania zachowania się stosów TCP/IP systemów podłączonych do sieci Internet. Prostota oraz funkcjonalność programu sprawiają, że może on spowodować wiele szkód, gdy znajdzie się w niepowołanych rękach.

Test popularnych ścian ogniowych wykazał, że nie tylko nie obsługują one protokołu IPv6, ale również często wzmacniają skutki ataków rozproszonych. Z trzech testowanych ścian ogniowych, wyłącznie jedna zasygnalizowała rozproszony atak, aczkolwiek nie podjęła żadnych kroków mających na celu usprawnienie komunikacji. Dwie z trzech testowanych ścian ogniowych z kolei dodatkowo spowodowały wzrost zużycia procesora podczas ataku DDoS, co zwielokrotniło skutek tegoż ataku.

Najlepszym mechanizmem zabezpieczającym przed atakami odmowy usługi okazał się linuksowy mechanizm SYN Cookie. Nie dość, że nie spowodował on wzrostu obciążenia procesora tak jak mechanizmy systemu Windows, to jeszcze zapewnił bezproblemowe połączenie przy dowolnym natężeniu ruchu, które dało się osiągnąć poprzez program *DDoS Generator*.

Ponadto praca pokazała, że stosowanie poprawek i uaktualnień może znacząco poprawić wydajność mechanizmów sieciowych. W badanych systemach uaktualnienie poprawiło dostępność usługi tak, że do osiągnięcia podobnego skutku ataku DDoS, wymagane było wygenerowanie ruchu dziesięciokrotnie większego.

## 8. Bibliografia

- [AIE-1] <http://www.anml.iu.edu/ddos/types.html>
- [CERT-1] <http://www.cert.org/advisories/CA-2000-21.html>
- [CTA-1] [http://www-comnet.technion.ac.il/~cn1w03/docs/DDoS\\_Attacks\\_methods\\_new.doc](http://www-comnet.technion.ac.il/~cn1w03/docs/DDoS_Attacks_methods_new.doc)
- [CYT-1] <http://cr.yp.to/syncookies.html>
- [DN-1] [http://digital.net/~gandalf/Rose\\_Frag\\_Attack\\_Explained.htm](http://digital.net/~gandalf/Rose_Frag_Attack_Explained.htm)
- [ISS-1] [http://www.iss.net/security\\_center/advice/Exploits/TCP/SYN\\_flood/default.htm](http://www.iss.net/security_center/advice/Exploits/TCP/SYN_flood/default.htm)
- [ISS-2] [http://www.iss.net/security\\_center/advice/Exploits/TCP/land/default.htm](http://www.iss.net/security_center/advice/Exploits/TCP/land/default.htm)
- [ISS-3] [http://www.iss.net/security\\_center/advice/Exploits/Ports/139/default.htm](http://www.iss.net/security_center/advice/Exploits/Ports/139/default.htm)
- [PFS-1] <http://personal-firewall-software-review.toptenreviews.com/?ttreng=1&ttrkey=firewall+comparison>
- [PT-1] <http://pintday.org/whitepapers/dos-smurf.shtml>
- [SF-1] <http://labrea.sourceforge.net/labrea-info.html>
- [SFC-1] <http://www.securityfocus.com/archive/1/392354>
- [SFC-2] <http://www.securityfocus.com/bid/13658/info/>
- [SIS-1] <http://www.sysinternals.com/ntw2k/freeware/procexp.shtml>
- [SMC-1] <http://support.microsoft.com/kb/q177539/>
- [TIQ-1] <http://www.theinquirer.net/?article=10733>
- [WAC-1] <http://www.agnitum.com/products/outpost>
- [WAO-1] [http://www.atis.org/tg2k/\\_dotted\\_decimal\\_notation.html](http://www.atis.org/tg2k/_dotted_decimal_notation.html)
- [WCU-1] <http://www.csif.ucdavis.edu/~engle/projects/va/resources.html>
- [WMC-1] <http://www.microsoft.com/poland/windowsserver2003/opis/ocena/default.msp>
- [WMC-2] <http://support.microsoft.com/default.aspx?scid=kb;pl;315669#2>
- [WPI-1] <http://winpcap.polito.it/>
- [WPN-1] <http://www.packetfactory.net/libnet>
- [WSC-1] <http://www.sygate.com/products/sygate-personal-firewall-pro.htm>
- [WTC-1] [http://www.tcpipguide.com/free/t\\_IPv6AddressandAddressNotationandPrefixRepresentati.htm](http://www.tcpipguide.com/free/t_IPv6AddressandAddressNotationandPrefixRepresentati.htm)
- [WZC-1] [http://www.zonelabs.com/store/content/catalog/products/zap/zap\\_details.jsp?lid=ho\\_zap](http://www.zonelabs.com/store/content/catalog/products/zap/zap_details.jsp?lid=ho_zap)

## 9. Dodatki

Spis odnośników zawierających lukę wykrytą podczas eksperymentów:

Lista dyskusyjna NTBUGTRAQ:

<http://www.ntbugtraq.com/default.aspx?pid=36&sid=1&A2=ind0505&L=NTBUGTRAQ&P=R409&D=0&F=N&H=0&O=D&T=0>

<http://www.derkeiler.com/Mailing-Lists/NT-Bugtraq/2005-05/0006.html>

French Security Incident Response Team

<http://www.frsirt.com/english/advisories/2005/0559>

Lista BUGTRAQ:

bugtraq id 13658; <http://www.securityfocus.com/bid/13658/info/>

Internet Security Systems:

<http://xforce.iss.net/xforce/xfdb/20629>

Common Vulnerabilities and Exposures:

05.20.1 - CVE: CAN-2005-1649

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-1649>

SANS Institute - Computer Security Education and Information Security Training:

**@Risk: The Consensus Security Alert - Volume 4, Issue #20 - Mozilla Firefox**

Plik Edycja Widok Przejść Zakładki Narzędzia Pomoc

<http://www.sans.org/newsletters/risk/display.php?printer=Y#05.20.1>

**References:**

- Posting by Alex Wheeler  
<http://www.rem0te.com/public/images/zen.pdf>
- Novell Announcement  
<http://support.novell.com/cgi-bin/search/searchtid.cgi?/10097644.htm>
- ZENWorks Product Page  
<http://www.novell.com/products/zenworks/quicklook.html>
- SecurityFocus BID  
<http://www.securityfocus.com/bid/13678>

**(2) LOW: Windows XP/2003 IPv6 Land Attack**

**Affected:**  
Windows XP/2003

**Description:** Land attack, a denial-of service attack known since 1997, can be launched by directing IP packets with same source and destination IP addresses at the target machine. Windows XP SP2 and Windows 2003 SP1 server are reportedly vulnerable to this attack. By continuously sending a stream of malformed IPv6 (IP Protocol version 6) TCP "SYN" packets to the open ports on a Windows XP/2003 system, it is possible to cause a significant performance degradation thereby rendering the system unusable. Exploit code has been publicly posted.

**Status:** Microsoft has been informed about the vulnerability. No patches are available yet. Appropriate ingress/egress filtering would also defend against this attack. Another alternative is to use firewalls and router ACLs to block such attacks.

**Council Site Actions:** Council sites report they have a large number of systems that could potentially be affected by a local IPv6 attack. Nearly all of their affected systems will obtain the update through the public Windows Update site, or through their local SUS server, whenever Microsoft happens to release a patch for this. They are not planning to proactively block the attack, although it is possible that the vulnerability may cause them to temporarily hold off on further expansion of external IPv6 availability.

**References:**

- Posting and Exploit Code by Konrad Malewski  
<http://marc.theaimsgroup.com/?l=ntbugtraq&m=111633815018247&w=2>
- SecurityFocus BID  
<http://www.securityfocus.com/bid/13658>

**Other Software**

**(3) HIGH: Neteyes Nexusway Border Gateway Administrative Access**

**Affected:**  
Possibly all versions

**Description:** Nexusway, a networking product from Taiwan, is designed to be a border gateway product to connect multiple networks. The gateway's web administration contains multiple flaws that can be exploited to obtain administrative control over the device, or run arbitrary commands. The posting shows how to

Zakończono

Proxy: w3cache.polsk.gliwice.pl Adblock

# GNU Free Documentation License

Uwaga!

To jest nieoficjalne tłumaczenie Licencji GNU Wolnej Dokumentacji na język polski. Nie zostało opublikowane przez Free Software Foundation i pod względem prawnym nie stanowi warunków rozpowszechniania tekstów stosujących GNU FDL -- ustanawia je wyłącznie oryginalny angielski tekst licencji GNU FDL. Jednak mamy nadzieję, że pomoże ono lepiej zrozumieć Licencję osobom mówiącym po polsku.

Licencja GNU Wolnej Dokumentacji  
Wersja 1.1, marzec 2000

Copyright (c) 2000 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Zezwala się na kopiowanie i rozpowszechnianie wiernych kopii niniejszego dokumentu licencyjnego, jednak bez prawa wprowadzania zmian.

## 0. Preambuła

Celem niniejszej licencji jest zagwarantowanie wolnego dostępu do podręcznika, treści książki i wszelkiej dokumentacji w formie pisanej oraz zapewnienie każdemu użytkownikowi swobody kopiowania i rozpowszechniania wyżej wymienionych, z dokonywaniem modyfikacji lub bez, zarówno w celach komercyjnych, jak i nie komercyjnych. Ponad to Licencja ta pozwala przyznać zasługi autorowi i wydawcy przy jednoczesnym ich zwolnieniu z odpowiedzialności za modyfikacje dokonywane przez innych.

Niniejsza Licencja zastrzega też, że wszelkie prace powstałe na podstawie tego dokumentu muszą nosić cechę wolnego dostępu w tym samym sensie co produkt oryginalny. Licencja stanowi uzupełnienie Powszechnej Licencji Publicznej GNU (GNU General Public License), która jest licencją dotyczącą wolnego oprogramowania.

Niniejsza Licencja została opracowana z zamiarem zastosowania jej do podręczników do wolnego oprogramowania, ponieważ wolne oprogramowanie wymaga wolnej dokumentacji: wolny program powinien być rozpowszechniany z podręcznikami, których dotyczą te same prawa, które wiążą się z oprogramowaniem. Licencja ta nie ogranicza się jednak do podręczników oprogramowania. Można ją stosować do różnych dokumentów tekstowych, bez względu na ich przedmiot oraz niezależnie od tego, czy zostały opublikowane w postaci książki drukowanej. Stosowanie tej Licencji zalecane jest głównie w przypadku prac, których celem jest instruktaż lub pomoc podręczna.

## 1. Zastosowanie i definicje

Niniejsza Licencja stosuje się do podręczników i innych prac, na których umieszczona jest pochodząca od właściciela praw autorskich informacja, że dana praca może być rozpowszechniana wyłącznie na warunkach niniejszej Licencji. Używane poniżej słowo "Dokument" odnosić się będzie do wszelkich tego typu publikacji. Ich odbiorcy nazywani będą licencjobiorcami.

"Zmodyfikowana wersja" Dokumentu oznacza wszelkie prace zawierające Dokument lub jego część w postaci dosłownej bądź zmodyfikowanej i/lub przełożonej na inny język.

"Sekcją drugorzędą" nazywa się dodatek opatrzoney odrębnym tytułem lub sekcję początkową Dokumentu, która dotyczy wyłącznie związku wydawców lub autorów Dokumentu z ogólną tematyką Dokumentu (lub zagadnieniami z nią związanymi) i nie zawiera żadnych treści bezpośrednio związanych z ogólną tematyką (na przykład, jeżeli Dokument stanowi w części podręcznik matematyki, Sekcja drugorzędna nie może wyjaśniać zagadnień matematycznych). Wyżej wyjaśniany związek może się natomiast wyrażać w aspektach historycznym, prawnym, komercyjnym, filozoficznym, etycznym lub politycznym.

"Sekcje niezmiennicze" to takie Sekcje drugorzędne, których tytuły są ustalone jako tytuły Sekcji niezmienniczych w nocy informującej, że Dokument został opublikowany na warunkach Licencji.

"Treść okładki" to pewne krótkie fragmenty tekstu, które w nocy informującej, że Dokument został opublikowany na warunkach Licencji, są opisywane jako "do umieszczenia na przedniej okładce" lub "do umieszczenia na tylnej okładce".

"Jawna" kopia Dokumentu oznacza kopię czytelną dla komputera, zapisaną w formacie, którego specyfikacja jest publicznie dostępna. Zawartość tej kopii może być oglądana i edytowana bezpośrednio za pomocą typowego edytora tekstu lub (w przypadku obrazów złożonych z pikseli) za pomocą typowego programu graficznego lub (w przypadku rysunków) za pomocą ogólnie dostępnego edytora rysunków. Ponadto kopia ta stanowi odpowiednie dane wejściowe dla programów formatujących tekst lub dla programów konwertujących do różnych formatów odpowiednich dla programów formatujących tekst. Kopia spełniająca powyższe warunki, w której jednak zostały wstawione znaczniki mające na celu utrudnienie dalszych modyfikacji przez czytelników, nie jest Jawna. Kopię, która nie jest "Jawna", nazywa się "Niejawną".

Przykładowe formaty kopii Jawnych to: czysty tekst ASCII bez znaczników, format wejściowy Texinfo, format wejściowy LaTeX, SGML lub XML wykorzystujące publicznie dostępne DTD, standardowy prosty HTML przeznaczony do ręcznej modyfikacji. Formaty niejawne to na przykład PostScript, PDF, formaty własne, które mogą być odczytywane i edytowane jedynie przez własne edytory tekstu, SGML lub XML, dla których DTD i/lub narzędzia przetwarzające nie są ogólnie dostępne, oraz HTML wygenerowany maszynowo przez niektóre procesory tekstu jedynie w celu uzyskania danych wynikowych.

"Strona tytułowa" oznacza, w przypadku książki drukowanej, samą stronę tytułową oraz kolejne strony zawierające informacje, które zgodnie z tą Licencją muszą pojawić się na stronie tytułowej. W przypadku prac w formatach nieposiadających strony tytułowej "Strona tytułowa" oznacza tekst pojawiający się najbliżej tytułu pracy, poprzedzający początek tekstu głównego.

## 2. Kopiowanie dosłowne

Licencjodawca może kopiować i rozprowadzać Dokument komercyjnie lub niekomercyjnie, w dowolnej postaci, pod warunkiem zamieszczenia na każdej kopii Dokumentu treści Licencji, informacji o prawie autorskim oraz noty mówiącej, że do Dokumentu ma zastosowanie niniejsza Licencja, a także pod warunkiem nie umieszczania żadnych dodatkowych ograniczeń, które nie wynikają z Licencji. Licencjodawca nie ma prawa używać żadnych technicznych metod pomiarowych utrudniających lub kontrolujących czytanie lub dalsze kopiowanie utworzonych i rozpowszechnianych przez siebie kopii. Może jednak pobierać opłaty za udostępnianie kopii. W przypadku dystrybucji dużej liczby kopii Licencjodawca jest zobowiązany przestrzegać warunków wymienionych w punkcie 3.

Licencjodawca może także wypożyczać kopie na warunkach opisanych powyżej, a także wystawiać je publicznie.

### 3. Kopiowanie ilościowe

Jeżeli Licencjodawca publikuje drukowane kopie Dokumentu w liczbie większej niż 100, a licencja Dokumentu wymaga umieszczenia Treści okładki, należy dołączyć kopie okładek, które zawierają całą wyraźną i czytelną Treść okładki: treść przedniej okładki, na przedniej okładce, a treść tylnej okładki, na tylnej okładce. Obie okładki muszą też jasno i czytelnie informować o Licencjodawcy jako wydawcy tych kopii. Okładka przednia musi przedstawiać pełny tytuł; wszystkie słowa muszą być równie dobrze widoczne i czytelne. Licencjodawca może na okładkach umieszczać także inne informacje dodatkowe. Kopiowanie ze zmianami ograniczonymi do okładek, dopóki nie narusza tytułu Dokumentu i spełnia opisane warunki, może być traktowane pod innymi względami jako kopiowanie dosłowne.

Jeżeli napisy wymagane na którejś z okładek są zbyt obszerne, by mogły pozostać czytelne po ich umieszczeniu, Licencjodawca powinien umieścić ich początek (taką ilość, jaka wydaje się rozsądna) na rzeczywistej okładce, a pozostałą część na sąsiednich stronach.

W przypadku publikowania lub rozpowszechniania Niejawnych kopii Dokumentu w liczbie większej niż 100, Licencjodawca zobowiązany jest albo dołączyć do każdej z nich Jawną kopię czytelną dla komputera, albo wymienić w lub przy każdej kopii Niejawnej publicznie dostępną w sieci komputerowej lokalizację pełnej kopii Jawnej Dokumentu, bez żadnych informacji dodanych -- lokalizację, do której każdy użytkownik sieci miałby bezpłatny anonimowy dostęp za pomocą standardowych publicznych protokołów sieciowych. W przypadku drugim Licencjodawca musi podjąć odpowiednie środki ostrożności, by wymieniona kopia Jawną pozostała dostępna we wskazanej lokalizacji przynajmniej przez rok od momentu rozpowszechnienia ostatniej kopii Niejawnej (bezpośredniego lub przez agentów albo sprzedawców) danego wydania.

Zaleca się, choć nie wymaga, aby przed rozpoczęciem rozpowszechniania dużej liczby kopii Dokumentu, Licencjodawca skontaktował się z jego autorami celem uzyskania uaktualnionej wersji Dokumentu.

### 4. Modyfikacje

Licencjodawca może kopiować i rozpowszechniać Zmodyfikowaną wersję Dokumentu na zasadach wymienionych powyżej w punkcie 2 i 3 pod warunkiem ścisłego przestrzegania niniejszej Licencji. Zmodyfikowana wersja pełni wtedy rolę Dokumentu, a więc Licencja dotycząca modyfikacji i rozpowszechniania Zmodyfikowanej wersji przenoszona jest na każdego, kto posiada jej kopię. Ponadto Licencjodawca musi w stosunku do Zmodyfikowanej wersji spełnić następujące wymogi:

- \* A. Użyć na Stronie tytułowej (i na okładkach, o ile istnieją) tytułu innego niż tytuł Dokumentu i innego niż tytuły poprzednich wersji (które, o ile istniały, powinny zostać wymienione w Dokumencie, w sekcji Historia). Tytułu jednej z ostatnich wersji Licencjodawca może użyć, jeżeli jej wydawca wyrazi na to zgodę.
- \* B. Wymienić na Stronie tytułowej, jako autorów, jedną lub kilka osób albo jednostek odpowiedzialnych za autorstwo modyfikacji Zmodyfikowanej wersji, a także przynajmniej pięciu spośród pierwotnych autorów Dokumentu (wszystkich, jeśli było ich mniej niż pięciu).
- \* C. Umieścić na Stronie tytułowej nazwę wydawcy Zmodyfikowanej wersji.
- \* D. Zachować wszelkie noty o prawach autorskich zawarte w Dokumencie.
- \* E. Dodać odpowiednią notę o prawach autorskich dotyczących modyfikacji obok innych not o prawach autorskich.
- \* F. Bezpośrednio po notach o prawach autorskich, zamieścić notę licencyjną zezwalającą na publiczne użytkowanie Zmodyfikowanej wersji na zasadach niniejszej Licencji w postaci podanej w Załączniku poniżej.
- \* G. Zachować w notce licencyjnej pełną listę Sekcji niezmiennych i wymaganych Treści okładki podanych w notce licencyjnej Dokumentu.
- \* H. Dołączyć niezmienną kopię niniejszej Licencji.
- \* I. Zachować sekcję zatytułowaną "Historia" oraz jej tytuł i dodać do niej informację dotyczącą przynajmniej tytułu, roku publikacji, nowych autorów i wydawcy Zmodyfikowanej wersji zgodnie z danymi zamieszczonymi na Stronie tytułowej. Jeżeli w Dokumencie nie istnieje sekcja pod tytułem "Historia", należy ją utworzyć, podając tytuł, rok, autorów i wydawcę Dokumentu zgodnie z danymi zamieszczonymi na stronie tytułowej, a następnie dodając informację dotyczącą Zmodyfikowanej wersji, jak opisano w poprzednim zdaniu.
- \* J. Zachować wymienioną w Dokumencie (jeśli taka istniała) informację o lokalizacji sieciowej, publicznie dostępnej Jawnej kopii Dokumentu, a także o podanych w Dokumencie lokalizacjach sieciowych poprzednich wersji, na których został on oparty. Informacje te mogą się znajdować w sekcji "Historia". Zezwala się na pominięcie lokalizacji sieciowej prac, które zostały wydane przynajmniej cztery lata przed samym Dokumentem, a także tych, których pierwotny wydawca wyraża na to zgodę.
- \* K. W każdej sekcji zatytułowanej "Podziękowania" lub "Dedykacje" zachować tytuł i treść, oddając również ton każdego z podziękowań i dedykacji.
- \* L. Zachować wszelkie Sekcje niezmiennych Dokumentu w niezmiennionej postaci (dotyczy zarówno treści, jak i tytułu). Numery sekcji i równoważne im oznaczenia nie są traktowane jako należące do tytułów sekcji.
- \* M. Usunąć wszelkie sekcje zatytułowane "Adnotacje". Nie muszą one być załączane w Zmodyfikowanej wersji.
- \* N. Nie nadawać żadnej z istniejących sekcji tytułu "Adnotacje" ani tytułu pokrywającego się z jakkolwiek Sekcją niezmienną.

Jeżeli Zmodyfikowana wersja zawiera nowe sekcje początkowe lub dodatki stanowiące Sekcje drugorzędne i nie zawierające materiału skopiowanego z Dokumentu, Licencjodawca może je lub ich część oznaczyć jako sekcje niezmiennych. W tym celu musi on dodać ich tytuły do listy Sekcji niezmiennych zawartej w notce licencyjnej Zmodyfikowanej wersji. Tytuły te muszą być różne od tytułów pozostałych sekcji.

Licencjodawca może dodać sekcję "Adnotacje", pod warunkiem, że nie zawiera ona żadnych treści innych niż adnotacje dotyczące Zmodyfikowanej wersji -- mogą to być na przykład stwierdzenia o recenzji koleżeńskej albo o akceptacji tekstu przez organizację jako autorytatywnej definicji standardu.

Na końcu listy Treści okładki w Zmodyfikowanej wersji, Licencjodawca może dodać fragment "do umieszczenia na przedniej okładce" o długości nie przekraczającej pięciu słów, a także fragment o długości do 25 słów "do umieszczenia na tylnej okładce". Przez każdą jednostkę (lub na mocy ustaleń przez nią poczynionych) może zostać dodany tylko jeden fragment z przeznaczeniem na przednią okładkę i jeden z przeznaczeniem na tylną. Jeżeli Dokument zawiera już treść okładki dla danej okładki, dodaną uprzednio przez Licencjodawcę lub w ramach ustaleń z jednostką, w imieniu której działa Licencjodawca, nowa treść okładki nie może zostać dodana. Dopuszcza się jednak zastąpienie poprzedniej treści okładki nową pod warunkiem wyraźnej zgody poprzedniego wydawcy, od którego stara treść pochodzi.

Niniejsza Licencja nie oznacza, iż autor (autorzy) i wydawca (wydawcy) wyrażają zgodę na publiczne używanie ich nazwisk w celu zapewnienia autorytetu jakiegokolwiek Zmodyfikowanej wersji.

## 5. Łączenie dokumentów

Licencjodawca może łączyć Dokument z innymi dokumentami wydanymi na warunkach niniejszej Licencji, na warunkach podanych dla wersji zmodyfikowanych w części 4 powyżej, jednak tylko wtedy, gdy w połączeniu zostaną zawarte wszystkie Sekcje niezmiennych wszystkich oryginalnych dokumentów w postaci niezmodyfikowanej i gdy będą one wymienione jako Sekcje niezmiennych połączenia w jego nocy licencyjnej.

Połączenie wymaga tylko jednej kopii niniejszej Licencji, a kilka identycznych Sekcji niezmiennych może zostać zastąpionych jedną. Jeżeli istnieje kilka Sekcji niezmiennych o tym samym tytule, ale różnej zawartości, Licencjodawca jest zobowiązany uczynić tytuł każdej z nich unikalnym poprzez dodanie na jego końcu, w nawiasach, nazwy oryginalnego autora lub wydawcy danej sekcji, o ile jest znany, lub unikalnego numeru. Podobne poprawki wymagane są w tytułach sekcji na liście Sekcji niezmiennych w nocy licencyjnej połączenia.

W połączeniu Licencjodawca musi zawrzeć wszystkie sekcje zatytułowane "Historia" z dokumentów oryginalnych, tworząc jedną sekcję "Historia". Podobnie ma postąpić z sekcjami "Podziękowania" i "Dedykacje". Wszystkie sekcje zatytułowane "Adnotacje" należy usunąć.

## 6. Zbiory dokumentów

Licencjodawca może utworzyć zbiór składający się z Dokumentu i innych dokumentów wydanych zgodnie z niniejszą Licencją i zastąpić poszczególne kopie Licencji pochodzące z tych dokumentów jedną kopią dołączoną do zbioru, pod warunkiem zachowania zasad Licencji dotyczących kopii dosłownych we wszelkich innych aspektach każdego z dokumentów.

Z takiego zbioru Licencjodawca może wyodrębnić pojedynczy dokument i rozpowszechniać go niezależnie na zasadach niniejszej Licencji, pod warunkiem zamieszczenia w wyodrębnionym dokumencie kopii niniejszej Licencji oraz zachowania zasad Licencji we wszystkich aspektach dotyczących dosłownej kopii tego dokumentu.

## 7. Zestawienia z pracami niezależnymi

Kompilacja Dokumentu lub jego pochodnych z innymi oddzielnymi i niezależnymi dokumentami lub pracami nie jest uznawana za Zmodyfikowaną wersję Dokumentu, chyba że odnoszą się do niej jako do całości prawa autorskie. Taka kompilacja jest nazywana zestawieniem, a niniejsza Licencja nie dotyczy samodzielnych prac skompilowanych z Dokumentem, jeśli nie są to pochodne Dokumentu.

Jeżeli do kopii Dokumentu odnoszą się wymagania dotyczące Treści okładki wymienione w części 3 i jeżeli Dokument stanowi mniej niż jedną czwartą całości zestawienia, Treść okładki Dokumentu może być umieszczona na okładkach zamykających Dokument w obrębie zestawienia. W przeciwnym razie Treść okładki musi się pojawić na okładkach całego zestawienia.

## 8. Tłumaczenie

Tłumaczenie jest uznawane za rodzaj modyfikacji, a więc Licencjodawca może rozpowszechniać tłumaczenia Dokumentu na zasadach wymienionych w punkcie 4. Zastąpienie Sekcji niezmiennych ich tłumaczeniem wymaga specjalnej zgody właścicieli prawa autorskiego. Dopuszcza się jednak zamieszczanie tłumaczeń wybranych lub wszystkich Sekcji niezmiennych obok ich wersji oryginalnych. Podanie tłumaczenia niniejszej Licencji możliwe jest pod warunkiem zamieszczenia także jej oryginalnej wersji angielskiej. W przypadku niezgodności pomiędzy zamieszczonym tłumaczeniem a oryginalną wersją angielską niniejszej Licencji moc prawną ma oryginalna wersja angielska.

## 9. Wygaśnięcie

Poza przypadkami jednoznacznie dopuszczonymi na warunkach niniejszej Licencji nie zezwala się Licencjodawcy na kopiowanie, modyfikowanie, czy rozpowszechnianie Dokumentu ani też na cedowanie praw licencyjnych. We wszystkich pozostałych wypadkach każda próba kopiowania, modyfikowania lub rozpowszechniania Dokumentu albo cedowania praw licencyjnych jest nieważna i powoduje automatyczne wygaśnięcie praw, które licencjodawca nabył z tytułu Licencji. Niemniej jednak w odniesieniu do stron, które już otrzymały od Licencjodawcy kopie albo prawa w ramach niniejszej Licencji, licencje nie zostaną anulowane, dopóki strony te w pełni się do nich stosują.

## 10. Przyszłe wersje Licencji

W miarę potrzeby Free Software Foundation może publikować nowe poprawione wersje GNU Free Documentation License. Wersje te muszą pozostawać w duchu podobnym do wersji obecnej, choć mogą się różnić w szczegółach dotyczących nowych problemów czy zagadnień. Patrz <http://www.gnu.org/copyleft/>. Każdej wersji niniejszej Licencji nadaje się wyróżniający ją numer. Jeżeli w Dokumencie podaje się numer wersji Licencji, oznaczający, iż odnosi się do niego podana "lub jakkolwiek późniejsza" wersja licencji, Licencjodawca ma do wyboru stosować się do postanowień i warunków albo tej wersji, albo którejkolwiek wersji późniejszej opublikowanej oficjalnie (nie jako propozycja) przez Free Software Foundation. Jeśli Dokument nie podaje numeru wersji niniejszej Licencji, Licencjodawca może wybrać dowolną wersję kiedykolwiek opublikowaną (nie jako propozycja) przez Free Software Foundation.

Załącznik: Jak zastosować tę Licencję dla swojego dokumentu.

Aby zastosować tę Licencję w stosunku do dokumentu swojego autorstwa, dołącz kopię Licencji do dokumentu i zamieść następującą informację o prawach autorskich i uwagi o licencji bezpośrednio po stronie tytułowej.

Copyright (c) ROK TWOJE IMIE I NAZWISKO

Udziela się zezwolenia do kopiowania rozpowszechniania i/lub modyfikację tego dokumentu zgodnie z zasadami Licencji GNU Wolnej Dokumentacji w wersji 1.1 lub dowolnej późniejszej opublikowanej przez Free Software Foundation; wraz z zawartymi Sekcjami Niezmiennymi LISTA TYTUŁÓW SEKCJI, wraz z Tekstem na Przedniej Okładce LISTA i z Tekstem na Tyłnej Okładce LISTA.  
Kopia licencji załączona jest w sekcji zatytułowanej "GNU Free Documentation License"

Jeśli nie zamieszczasz Sekcji Niezmiennych, napisz "nie zawiera Sekcji Niezmiennych" zamiast spisu sekcji niezmiennych. Jeśli nie umieszczasz Tekstu na Przedniej Okładce wpisz "bez Tekstu na Okładce" w miejsce "wraz z Tekstem na Przedniej Okładce LISTA", analogicznie postąp z "Tekstem na Tyłnej Okładce"

Jeśli w twoim dokumencie zawarte są nieszablony przykłady kodu programu, zalecamy abyś także uwolnił te przykłady wybierając licencję wolnego oprogramowania, taką jak Powszechna Licencja Publiczna GNU, w celu zapewnienia możliwości ich użycia w wolnym oprogramowaniu.

Informacja prawie autorskim powyżej.

Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA

Tłumaczenie na język polski: Krzysztof Łabanowski, w: "Linux - podręcznik administratora sieci", Wydawnictwo RM, Warszawa 2000.

Poprawiono kilka literówek, zmieniono "wolnodostępny" na "wolny".

W.Kotwica (wkotwica(at)post.pl)

Część omawiająca sposób stosowania GNU FDL do własnej dokumentacji została podesłana przez Przemka Sarnowskiego (Przemysław.Sarnowski(at)nfosigw.gov.pl)