



**POLITECHNIKA ŁÓDZKA**

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki

## **SYSTEMY DETEKCJI INTRUZÓW I AKTYWNEJ ODPOWIEDZI**

Praca magisterska wykonana przez  
**Macieja Skowrońskiego i Radosława Węzyka**  
pod kierunkiem **dr Macieja Szmita**  
w Katedrze Informatyki Stosowanej

Łódź 2006

## Spis treści:

Wstęp.....	4
Cel pracy .....	4
Układ pracy .....	4
Podział obowiązków .....	5
Metody badawcze i narzędzia .....	5
Rozdział pierwszy. Systemy IDS .....	5
1.1. Geneza systemów IDS (Maciej Skowroński).....	6
1.2. Zasada działania IDS-ów (Radosław Wężyk).....	9
1.3. Wykrywanie anormalnego zachowania (według: [Pieprzyk 2005], str. 416 i nast.) (Maciej Skowroński) .....	12
1.3.1. Statystyczne systemy IDS (według: [Pieprzyk 2005], str. 417 i nast.) .....	13
1.3.2. Systemy IDS oparte o wzorce predykowane (według: [Pieprzyk 2005], str. 419 i nast.) .....	14
1.4. Typy systemów IDS (Radosław Wężyk) .....	14
1.5. Przechwytywanie ruchu sieciowego w systemach IDS (Radosław Wężyk).....	20
1.5.1 Przechwytywanie ruchu w oparciu o tryb promiscuous.....	20
1.5.2. Przechwytywanie ruchu w sieciach zbudowanych w oparciu o przełączniki (ang. switch) .....	21
1.5.3. Przechwytywanie ruchu w sieciach VLAN (ang. Virtual Local Area Network)...	22
1.5.4. TAP (ang. Test Access Port) .....	22
1.5.5. Rozwiązania typu IN-LINE. ....	23
1.5.6. Systemy GIDS (ang. Gateway Intrusion Detection System). ....	24
1.5.7. Ukrywanie systemu IDS oraz praca interfejsów w trybie stealth (zobacz: [Baker 2004], str. 90) .....	24
1.6. Odpowiedzi systemów IDS (Maciej Skowroński).....	25
1.7. Problem fałszywych alarmów i gubienia pakietów (zobacz: [11]) (Radosław Wężyk) 26	
1.8. Architektura systemów IDS (Maciej Skowroński) .....	27
1.9. Metody komunikacji między elementami systemu IDS (Maciej Skowroński).....	32
1.10. Systemy IPS (Maciej Skowroński i Radosław Wężyk) .....	33
1.11. Systemy honeypot (Maciej Skowroński) .....	37
1.12. Ograniczenia systemów IDS (według: [Pieprzyk 2005], str. 438 i nast.) (Maciej Skowroński) .....	39
1.13. Snort – darmowy system detekcji intruzów (Maciej Skowroński i Radosław Wężyk).....	39
1.13.1. Tryby pracy Snorta.....	40
1.13.2. Ogólny schemat działania Snorta.....	44
1.13.3. Basic Analysis and Security Engine .....	48
Rozdział drugi. Implementacja. ....	51
2.1. Opis problemu (Maciej Skowroński) .....	51
2.2. Implementacja (Radosław Wężyk) .....	54
2.3. Preprocesor (Radosław Wężyk) .....	55
2.4. Generator profilu (Maciej Skowroński) .....	58
2.5. Opis użytkowy.....	64
2.5.1. Instalacja (Radosław Wężyk).....	64
2.5.2. Konfiguracja (Maciej Skowroński).....	65
2.5.3. Uruchamianie systemu Snort z preprocesorem AnomalyDetection. (Radosław Wężyk) .....	66
Rozdział trzeci. Badania empiryczne .....	70

3.1. Przedmiot badań.....	70
3.2. Wyniki pomiarów i wnioski.....	71
3.2.1. Porównanie otrzymanych wyników z wyznaczoną wartością średnią.....	74
3.2.2. Wykrywanie anomalii.....	74
3.2.2.1. Reguła 2 sigma.....	74
3.2.2.2. Reguła 2,5 sigma.....	75
3.2.2.3. Reguła 3 sigma.....	75
3.2.2.4. Liczba alertów a wartość mnożnika sigma.....	75
Zakończenie. Kierunki dalszych badań.....	80
Bibliografia.....	81
Załącznik 1: Wykresy przedstawiające wyniki pomiarów.....	83
Załącznik 2: Porównanie wyników i wartości średniej.....	132
Załącznik 3: Wykresy dla mnożnika sigmy równego 2.....	144
Załącznik 4: Wykresy dla mnożnika sigmy równego 2,5.....	157
Załącznik 5: Wykresy dla mnożnika sigmy równego 3.....	170
Załącznik 6: Wykresy zależności liczby alertów od wartości mnożnika sigmy.....	182
Załącznik 7: Słownik wyrażen obcojęzycznych i skrótów.....	186
Załącznik 8: Spis zawartości płyty CD-ROM.....	187
Spis ilustracji:.....	188
Spis tabel:.....	195
Spis wydruków:.....	195

## **Wstęp**

Łatwy i tani dostęp do Internetu dla osób prywatnych niosący ze sobą wygodę szybkiego uzyskania informacji, zapewniający atrakcyjne formy komunikacji oraz wiele rodzajów rozrywki powoduje ciągły wzrost liczby internautów. Informacje płynące w sieci są niezwykle ważne z punktu widzenia przedsiębiorstw i indywidualnych użytkowników. Numery kart kredytowych, rozliczenia bankowe, bazy danych to tylko nieliczne przykłady płynącej przez Internet tzw. informacji wrażliwej, której ochrona jest niezwykle ważna. Poważnym problemem administratorów sieci i serwerów jest utrzymanie stabilnego funkcjonowania systemów informatycznych tak, aby zapewnić użytkownikom możliwość komfortowej pracy. Do obowiązków administratorów należy zatem między innymi ochrona sieci przed działaniami hackerów, usuwanie podatności, wykrywanie i przeciwdziałanie próbom ataków, które nierzadko istotnie obciążają i destabilizują działanie systemów komputerowych. Dlatego od lat dąży się do stworzenia jak najlepszych narzędzi poprawiających bezpieczeństwo systemów komputerowych. Efektem tych prac było powstanie systemów wykrywania i przeciwdziałania atakom (IDS i IPS), które są bardzo zaawansowanymi systemami pozwalającymi na detekcję prób ataków oraz ich udaremnianie zanim atakującemu uda się złamać zabezpieczenia. Systemy te wykorzystują wiele różnych metod wyszukiwania śladów ataku w obserwowanym ruchu: od wyszukiwania wcześniej zdefiniowanych wzorców aż do przewidywania czynności wykonywanych przez użytkownika na podstawie jego wcześniejszych zachowań. Od końca lat dziewięćdziesiątych systemy IDS rozwijają się bardzo dynamicznie. Coraz częściej systemy tego typu instalowane są w firmach, sieciach osiedlowych, sieciach domowych a nawet w komputerach osobistych.

## **Cel pracy**

Celem niniejszej pracy było zaprojektowanie i implementacja oprogramowania służącego do detekcji anomalii ruchu sieciowego zintegrowanego z systemem IDS Snort oraz testy tegoż oprogramowania w przykładowej sieci LAN.

## **Układ pracy**

Praca niniejsza składa się z 3 rozdziałów oraz 8 załączników.

Rozdział pierwszy zawiera informacje dotyczące systemów IDS/IPS. Zaprezentowano w nim ich genezę, klasyfikację, konstrukcję oraz sposoby użycia. Opisany został również program Snort, jego budowa i możliwości.

Rozdział drugi zawiera opis techniczny i użytkowy zrealizowanego systemu detekcji anomalii. Przedstawiono w nim budowę i najważniejsze funkcje napisanych programów oraz opisano sposób ich instalacji i użytkowania.

W rozdziale trzecim zaprezentowane są wyniki otrzymane przy zastosowaniu napisanego systemu detekcji anomalii. Zawiera on zebrane dane i ich analizę statystyczną.

Do pracy dołączony został CD-ROM zawierający kod źródłowy i wykonywalny opracowanych programów.

## ***Podział obowiązków***

Każdy z autorów pracy zajął się opracowaniem poszczególnych zagadnień. Nazwisko autora danego rozdziału znajduje się w jego tytule. Część systemu przechwytyjąca i zapisująca dane zaimplementowana została przez Radosława Wężyka. Część aplikacji odpowiedzialna za analizę statystyczną przechwyconych danych zaimplementowana została przez Maciej Skowrońskiego. Analiza zebranych danych przeprowadzona została wspólnie przez obu autorów.

## ***Metody badawcze i narzędzia***

W ramach pracy zaprojektowany i zaimplementowany został preprocesor do programu Snort w wersji 2.4.4 oraz program generujący profil sieci. W pracy wykorzystano system Snort, ponieważ jest to najbardziej popularny system detekcji intruzów rozwijany na zasadzie licencji GPL. W związku z faktem, że Snort został napisany pierwotnie dla systemów z rodziny Unix, platformą dla tworzenia programów był Slackware Linux w wersji 10.1. System Snort zaimplementowany jest w języku C w związku z czym, dla zachowania spójności preprocesor napisany został w języku programowania C, natomiast generator profili w języku C++. Do kompilacji wykorzystany został kompilator g++ w wersji 3.4.5.

Zebrane dane poddane zostały analizie statystycznej wykorzystującej podstawowe parametry (średnia, wariancja). W oparciu o nie zostały wyznaczone progowe wartości klasyfikujące ruch jako prawidłowy lub nie. Na podstawie wrywkowych testów zgodności zebranych danych z rozkładem normalnym (test chi kwadrat) założono, że dane podlegają rozkładowi normalnemu.

## **Rozdział pierwszy. Systemy IDS.**

System detekcji intruzów ma za zadanie analizowanie ruchu sieciowego w poszukiwaniu podejrzanych aktywności sieciowych i określania czy są one atakiem.

Natomiast systemy IPS mają dodatkowo możliwość podjęcia działań uniemożliwiających atakującemu dokonanie ataku.

### **1.1. Geneza systemów IDS (Maciej Skowroński)**

Możliwość połączenia komputerów za pomocą sieci daje ich użytkownikom wiele udogodnień: umożliwia korzystanie z ogromnych zasobów, pozwala na prostą i szybką komunikację oraz zapewnia bardzo szybki dostęp do informacji z wielu źródeł. W każdej sieci komputerowej znajdują się jednak zasoby, których właściciele nie chcieliby udostępniać poza jej obrębem (np. serwery plików, serwery DNS) lub nawet po za obrębem własnego komputera (np. prywatna poczta elektroniczna, dokumenty). Z tego powodu stosuje się różnego rodzaju sprzęt i oprogramowanie mające na celu ochronę tych komputerów przed nieautoryzowanym dostępem. Podstawowym zabezpieczeniem hosta przed atakiem, stosowanym dzisiaj już bardzo powszechnie, są firewalle, zwane też „ścianami przeciwogniowymi” (zobacz: [Szmit 2005], str. 451). Termin ten pochodzi z budownictwa, gdzie określa on ścianę, która ma nie dopuścić do rozprzestrzeniania się ognia pomiędzy częściami budynku. Podobną rolę w sieci spełnia firewall. Oddziela on poszczególne części sieci i nie dopuszcza do przechodzenia zagrożeń pomiędzy nimi. Firewalle dostępne są dla praktycznie każdego systemu operacyjnego i w wielu przypadkach instalują się standardowo wraz z nim. W przypadku standardowej zapory przeciwogniowej, zasada działania sprowadza się do porównywania każdego pojedynczego pakietu przychodzącego i wychodzącego z bazą dostarczonych reguł. Zapora analizuje swoją bazę reguł w poszukiwaniu reguły, która może mieć zastosowanie w przypadku badanego pakietu, a następnie postępuje zgodnie z nią. Filtry tego typu nie potrafią zazwyczaj analizować zawartości pakietu i traktują każdy przychodzący pakiet niezależnie<sup>1</sup>. Takie zabezpieczenie zapewnia podstawowy poziom bezpieczeństwa danej sieci. Oczywiście istnieją też bardziej zaawansowane systemy tego typu, których możliwości są o wiele większe (zobacz: [Szmit 2005] str. 453 i nast.). Odpowiednie skonfigurowanie firewalla nie jest jednak sprawą prostą ani szybką, jednak jego poprawna konfiguracja zapewnia już pewien poziom bezpieczeństwa. Zapory sieciowe nie są jednak często wystarczającą ochroną. Wadą tych systemów jest to, że nie będą one blokowały prób włamań do systemu, jeśli atakujący wykorzystuje dozwolone przez zaporę połączenie (np. połączenie do serwera WWW, ftp). Firewall korporacyjny (ang. *corporate firewall*) nie

---

<sup>1</sup> Niektóre współczesne firewalle (np. iptables) posiadają już mechanizmy pozwalające na uwzględnianie pewnych statystyk pakietów

zabezpieczy również użytkowników znajdujących się wewnątrz ochranianej przez niego sieci przed atakiem przeprowadzonym z hosta będącego w obrębie tej sieci. W celu obrony przed takimi atakami zaczęto stosować systemy zwane Intrusion Detection System nazywane w skrócie IDS. Tłumacząc dosłownie, jest to system wykrywania wtargnięć. Nazywa się je również systemami detekcji intruzów. Jednak najpopularniejszym zabezpieczeniem są dzisiaj osobiste zapory (ang. *personal firewall*). W podstawowej wersji, są one w większości darmowe dla domowego użytkownika (np. Kerio Personal Firewall). Po za standardowymi możliwościami filtrowania pakietów umożliwiają one monitorowanie, które z programów próbują uzyskać połączenie z siecią i dają użytkownikowi możliwość zablokowanie takich prób. Istnieją też całe pakiety służące do ochrony komputera, w skład których wchodzi między innymi zaporą osobistą i system IDS (np. Norton Internet Security).

Początków systemów detekcji intruzów można upatrywać w roku 1980 (zobacz: [Endorf 2004], rozdział: The History of Intrusion Detection and Prevention), kiedy to James Anderson napisał raport dla U.S. Air Force zwracając w nim uwagę na możliwość wykrywania niedozwolonego korzystania z komputerów. W roku 1985 została powołana przez U.S. Navy specjalna grupa w celu rozwoju badań nad systemami detekcji intruzów. Zespół ten pod kierownictwem Doroty Denning opracował pierwszy prototyp systemu IDS, którym mógł analizować aktywność użytkowników. System ten został nazwany Intrusion Detection Expert System (IDES). W roku 1987 Denning opublikowała pracę pod tytułem "Intrusion Detection Model", w której opisała podstawowe zagadnienia dotyczące analizy zachowań. Metoda ta polega na poszukiwaniu niestandardowych zachowań, które odbiegają od przyjętych za poprawne, takich jak zależności pomiędzy pakietami lub danymi przesyłanymi poprzez sieć. W międzyczasie, w roku 1987, w Los Alamos National Laboratory, pracowano nad projektem o nazwie „Haystack”. Efektem tych prac było stworzenie systemu IDS pracującego w oparciu o wzorce oraz analizę statystyczną. Największą wadą tego systemu było to, że nie pracował on w czasie rzeczywistym a analizował wcześniej zgromadzone dane w trybie offline (według: [Pieprzyk 2005], str. 432). W roku 1989 Todd Heberlein zbudował system IDS zwany Network System Monitor (NSM). W przeciwieństwie do systemów IDES i „Haystack” NSM bazował na analizie ruchu sieciowego, a nie analizie logów systemowych. Część osób pracujących przy projekcie „Haystack” współpracowała z pracownikami uniwersytetu California-Davis oraz z Lawrence Livermore National Laboratory. Efektem tej współpracy było powstanie systemu IDS zwanego Distributed Intrusion Detection System (DIDS). Projekt ten rozszerzał możliwości systemu NSM możliwość detekcji intruzów na podstawie zaszyfrowanych danych (według:

[Pieprzyk 2005], str. 434). DIDS stał się podstawą dla stworzenia pierwszego komercyjnego systemu IDS o nazwie Net Stalker. Stalker oraz NSM będąc systemami komercyjnymi wpłynęły na wzrost zainteresowania oraz rozwój systemów IDS. W roku 1990 pojawiło się bardzo wiele usprawnień w technologii IDS. Christopher Klaus wraz z Thomasem E. Noonan założyli Internet Security Systems (ISS) oraz napisali Network-Based Intrusion-Detection System (NIDS) pod nazwą Real Secure. Intensywne prace rozwojowe prowadziło nadal U.S. Air Force opracowując Computer Misuse Detection System (CMDS) oraz Automated Security Incident Measurement (ASIM), który to był pierwszym systemem IDS łączącym w sobie rozwiązania programowe oraz sprzętowe. Część twórców systemu ASIM stworzyło The Wheel Group i skomercjalizowało produkt. W roku 1998 Cisco Systems wykupiło The Wheel Group, co pozwoliło na wprowadzenie funkcji IDS do produkowanych przez tę firmę routerów. W tym samym czasie laboratoria Haystack oraz twórcy systemu SAIC połączyli się tworząc Centrax Corporation i wypuszczając na rynek Host-Based Intrusion-Detection System (HIDS) przeznaczony dla platformy Windows NT. System ten nosił nazwę eNTrax. W roku 1998 swoją premierę miał Snort. Został on opracowany przez Marty Roescha. Snort jest programem typu open-source, który może pracować jako NIDS. W Roku 1999, Okena Systems opracowała jeden z pierwszych Intrusion Prevention System (IPS) pod nazwą StormWatch. Firma Okena Systems została wykupiona w roku 2003 przez Cisco Systems. Najważniejsze wydarzenia w historii systemów IDS zostały przedstawione w tabeli 1.

**Tabela 1: Historia systemów IDS. Źródło: [Endorf 2004], rozdział: The History of Intrusion Detection and Prevention.**

Rok	Wydarzenia
1980	James Anderson napisał raport dla U.S. Air Force zwracając w nim uwagę na możliwość wykrywania niedozwolonego korzystania z komputerów
1985	założony przez U.S. Navy został SRI International w celu rozwoju badań nad systemami detekcji intruzów
1987	- Doroty Denning opublikowała prace pod tytułem "Intrusion Detection Model" -w Los Alamos National Labortory pracowano nad projektem o nazwie „Haystack” - powstanie systemu IDS pracującego w oparciu o sygnatury
1989	Todd Heberlein zbudował system IDS zwany Network System Monitor (NSM) bazujący na analizie ruchu sieciowego



## **1.2. Zasada działania IDS-ów (Radosław Wężyk)**

Systemy IDS mogą być dedykowanymi rozwiązaniami sprzętowymi lub programowymi. Ich zadaniem jest wykrywanie prób włamań do chronionej sieci lub hosta. W celu realizacji tego zadania monitorują ruch w sieci lub logi systemowe w celu wychwycenia podejrzanych zdarzeń i następnie podjęcia odpowiedniego działania. Z punktu widzenia IDS podejrzanyymi zdarzeniami są m.in. skanowanie portów, ‘skanowanie’ sieci w celu uzyskania informacji na temat jej struktury, wyszukania słabych punktów oraz celów włamania – serwerów, aplikacji oraz usług(określanie konfiguracji sprzętowej, systemu operacyjnego, wersji oprogramowania na danych maszynach).

IDSy przechwytyją surowe pakiety IP a następnie poddają je analizie w poszukiwaniu sygnatur znanych ataków, anomalii w ruchu lub też działań, które mogą być przygotowaniem do ataku (np.: skanowanie portów).

Podstawowymi metodami wykrywania ataków są (zobacz: [Szmit 2005], str. 498):

- Metoda dopasowywania sygnatur,

W metodzie tej IDS ma swoją wcześniej zdefiniowaną bazę wzorców poszczególnych ataków. Dla każdego typu ataku zdefiniowany jest osobny wzorzec. System porównuje aktualnie zaistniałą sytuację ze wszystkim wzorcami. Gdy któryś z wzorców zostanie dopasowany, generowany jest alert. Zazwyczaj istnieje możliwość napisania własnych wzorców monitorujących istotne z naszego punktu widzenia rzeczy (na przykład: specyficzną zawartość pakietu). Liczba ataków stale się zwiększa i tylko dbanie o to, aby w bazie reguł były zawsze aktualne wzorce ataków zapewnia, że IDS będzie skutecznie działał.

- Badanie częstotliwości występowania zdarzeń,

System ustala pewne limity odnośnie do pewnych zdarzeń mogących zajść w sieci (np. próby nieudanego logowania). Kilkakrotne wystąpienie danego zdarzenia może wiązać się ze zwykłym ruchem generowanym przez użytkownika. Nadmierna częstotliwość występowania danej sytuacji może świadczyć o próbie ataku. W momencie wykrycia przekroczenia ustalonego limitu, generowany jest alert.

- Analiza statystyczna ruchu,

System analizuje standardowy ruch w sieci i na tej podstawie buduje jej profil. Profil taki może być również budowany dla konkretnego użytkownika. Zazwyczaj każda sieć ma specyficzne dla siebie cechy takie jak przeciętna wielkość pakietów IP, stosunek danych wysyłanych do danych odbieranych, czy też liczbę połączeń nawiązywanych w

danej jednostce czasu. Specyfika ruchu w sieci zmienia się też często w zależności od pory dnia czy też dnia tygodnia. Analizie statystycznej może podlegać praktycznie każdy aspekt wykorzystania sieci. Proces tworzenia profilu trwa zazwyczaj kilka tygodni. Po stworzeniu go IDS porównuje aktualną sytuację w sieci do stworzonego profilu i określa jej zgodność ze zbudowanym wcześniej profilem. W przypadku odkrycia nieprawidłowości generowany jest alert.

Zaawansowane systemy IDS wykorzystają wiele metod wykrywania (zobacz: [Szmit 2005], str. 499 i nast.). Ruch w sieci jest analizowany jednocześnie za pomocą kilku metod. Zapewnia to zwiększenie wydajności oraz efektywności systemu. Rozróżniamy kilka zaawansowanych metod detekcji:

- Stateful signatures – pełnostanowe sygnatury.

Metoda ta polega na porównywaniu zaistniałej sytuacji z bazą zawierającą pełnostanowe sygnatury. Sygnatury te oprócz wzorca ataku zawierają również rodzaj komunikacji, w którym dany atak może wystąpić. Wzorce porównywane są tylko w obrębie danej komunikacji (na przykład tylko w obrębie obustronnie otwartego obwodu wirtualnego TCP), co pozwala na szybsze i efektywniejsze działanie systemu. Przeprowadzana jest analiza kontekstowa, co pozwala na uniknięcie wielu fałszywych alertów.

- Protocol anomalies – anomalie protokołów.

Metoda ta wykrywa niezgodność zaobserwowanego ruchu z normami opisanymi w dokumentach Request For Comment (RFC).

- Backdoor detection – wykrywanie „tylnych drzwi”.

Metoda ta polega na wykrywaniu działania koni trojańskich w chronionych systemach oraz ataków wykorzystujących tak zwane „tylne drzwi”. Metoda ta porównuje zaistniały ruch z wzorcami działań oraz analizuje pakiety w poszukiwaniu charakterystycznych dla danego konia trojańskiego wpisów.

- Traffic anomalies – anomalie ruchu.

Metoda ta analizuje ruch w sieci w poszukiwaniu nienormalnych zachowań jak na przykład wielokrotne łączenie się z danym hostem w bardzo krótkim czasie.

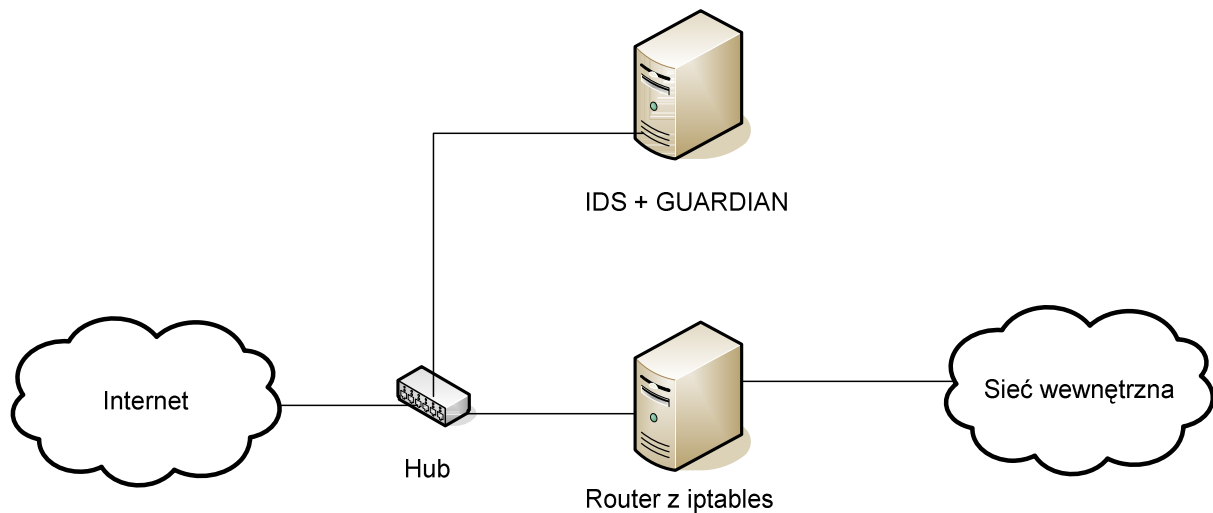
- Spoofing detection – wykrywanie podszywania.

Metoda ta analizuje ruch sieciowy w poszukiwaniu pakietów ze sfalszowanymi adresami nadawcy. Wykrywane jest to poprzez porównywanie adresu IP pakietu z adresami wykorzystywanymi w sieciach wewnętrznych.

- Layer 2 detection – monitorowanie warstwy drugiej.

Metoda ta polega na monitorowaniu ruchu na poziomie warstwy łącza danych. Umożliwia to weryfikację adresacji MAC. Metoda ta sprawdza się szczególnie wtedy, gdy IDS monitoruje również sieć wewnętrzną.

Sama analiza ruchu i generowanie alertów przeprowadzana przez IDS nie pozwala na zatrzymanie atakującego. Z tego powodu dla systemów tego typu powstały rozszerzenia, które pozwalają na zmianę ustawień zapory sieciowej opierając się na wygenerowanych przez system detekcji intruzów alertów (tak zwane systemy aktywnej odpowiedzi, ang. *active response*, które są czasem nazywane systemami IPS – Intrusion Prevention System). Przykładem może być współpraca programu Snort z aplikacją Guardian. Przykładowy schemat takiego systemu przedstawia rysunek 1.



**Rysunek 1: Schemat systemu aktywnej odpowiedzi. Źródło: opracowanie własne.**

W momencie wykrycia próby ataku, system detekcji wtargnięć generuje alert i zapisuje go w pliku z logami. Jest on następnie odczytywany przez Guardian, który generuje odpowiednie polecenie do zapory przeciwogniowej – w przypadku przedstawionego schematu jest nią popularna aplikacja iptables. Efektem tego jest zazwyczaj zablokowanie komunikacji pomiędzy atakującym a chronioną siecią. Blokada taka może być nakładana na określony czas, po którym jest ona zdejmowana. Rozwiązanie to ma jedną istotną wadę: w momencie, gdy system zaczyna blokować adresy, z których pochodzą ataki, wzrasta znaczenie problemu fałszywych alarmów. Atakujący może przeprowadzić w takim przypadku atak typu DoS (ang. *Denial of Service* – odmowa usług) polegający na przesłaniu do systemu IDS szeregu pakietów o zawartości odpowiadającej sygnaturom różnych rodzajów ataku, ze sfalszowanym adresem źródłowym. Adresy te zostaną w konsekwencji pozbawione możliwości korzystania z usług zabezpieczanego systemu. Jest to istotny

problem, ponieważ może on doprowadzić do zablokowania użytkowników, którzy korzystają w sposób legalny i nie będący żadnym zagrożeniem dla zasobów chronionej sieci.

Rozwinięciem koncepcji systemów IDS mogących reagować na zaistniałą sytuację, są systemy IPS. Systemy te współpracują z zaporą sieciową bez żadnych aplikacji pośredniczących. Dzięki temu czas potrzebny na reakcję staje się zdecydowanie krótszy. Ponadto systemy IPS oferują wiele innych możliwości, mogą na przykład uruchomić przygotowany wcześniej skrypt, wylogować użytkownika lub zablokować jego konto. IPS może równie poinformować administratora o zaistniałej sytuacji, wykorzystując do tego celu np. email, sms lub pager.

Jakość Systemu Detekcji Intruzów określają (zobacz: [Szmit 2005], str. 498 i nast.):

- wykrywalność ataków,
- liczba fałszywych alarmów,
- wydajność – w czasie analizy danych w sieci ‘gubienie’ pakietów musi być najmniejsze,
- baza i aktualizacja sygnatur,
- zakres reakcji,
- dostrajanie zabezpieczeń – możliwość ustawienia zabezpieczeń odpowiednich do chronionego systemu,
- zarządzanie zabezpieczeniami – przystępne narzędzia umożliwiające monitorowanie sieci, analizę niebezpiecznych zdarzeń oraz zarządzania ‘odległymi’ sensorami.

### ***1.3. Wykrywanie anormalnego zachowania (według: [Pieprzyk 2005], str. 416 i nast.) (Maciej Skowroński)***

System IDS, który wykrywa anormalne zachowanie użytkowników wykorzystuje do tego celu mierzalne cechy zachowania użytkowników w systemie. Na ich podstawie tworzy profil danego użytkownika. Zachowanie użytkownika można opisać w dwóch kategoriach:

- intensywność,
- niejednorodność.

Intensywność określa wartość każdego z wcześniej określonych parametrów mierzona w regularnych odstępach czasu. Jest ona ściśle powiązana z tym, jakiego typu aktywność prowadzi użytkownik. Wyróżnia się dwa typy charakterystyk intensywności:

- liczba wystąpień w jednostce czasu

- średnia ilość czasu przypadająca na jedno wystąpienie aktywności danego typu.

Każde z zarejestrowanych wystąpień może być następnie analizowane pod kątem parametrów pracy systemu (na przykład obciążenie procesora, liczba procesów w systemie czy też liczba operacji wejścia-wyjścia).

Niejednorodność ruchu generowanego przez danego użytkownika zwraca uwagę na to, że różne rodzaje aktywności mogą być powiązane ze sobą nie tylko tym, że wystąpiły, ale także tym, w jakiej kolejności nastąpiły. Przy czym zwraca się też uwagę na ich intensywność i uwarunkowania zewnętrzne.

Profil użytkownika może być zbudowany z wielu rodzajów informacji. Oto przykład kilku z nich:

- typ przejawianej aktywności (np. wysyłanie poczty, kompilacja programu, kopiowanie plików),
- kolejność występowania aktywności danego typu (np. zaraz po zalogowaniu odbierana i odczytywana jest poczta),
- kontekst występowania danej kolejności aktywności (np. inna kolejność czynności będzie gdy użytkownik znajduje się w biurze bezpośrednio przed komputerem, a inna gdy łączy się z nim zdalnie).

### **1.3.1. Statystyczne systemy IDS (według: [Pieprzyk 2005], str. 417 i nast.)**

Statystyczne systemy IDS jako podstawę detekcji anomalii wykorzystują wybrane podczas implementacji parametry opisujące aktywność użytkowników. Następnie każdemu z tych parametrów przypisywana jest wartość, która uznawana jest w późniejszych obliczeniach za wzorcową. Wartości te są określane na podstawie pomiarów dokonywanych przez system IDS. Dodatkowo każdemu z parametrów może być przyporządkowana odpowiednia waga odzwierciedlająca jego istotność. Określa się też wartość progową, która określa maksymalne odstępstwo zmierzonej wartości od wcześniej ustalonego wzorca. Wyznaczenie tej wartości odbywa się w sposób eksperymentalny poprzez analizę ilości alarmów fałszywych oraz nie wykrytych anomalii. Wpływ na ten parametr ma też polityka bezpieczeństwa. Wykrywanie anomalii w statystycznym systemie IDS odbywa się poprzez porównywanie aktualnie zaistniałej w sieci wartości do profilu użytkownika. Zaletą tego typu systemu jest ciągle uaktualnianie profilu użytkownika, dzięki czemu jest on zawsze aktualny.

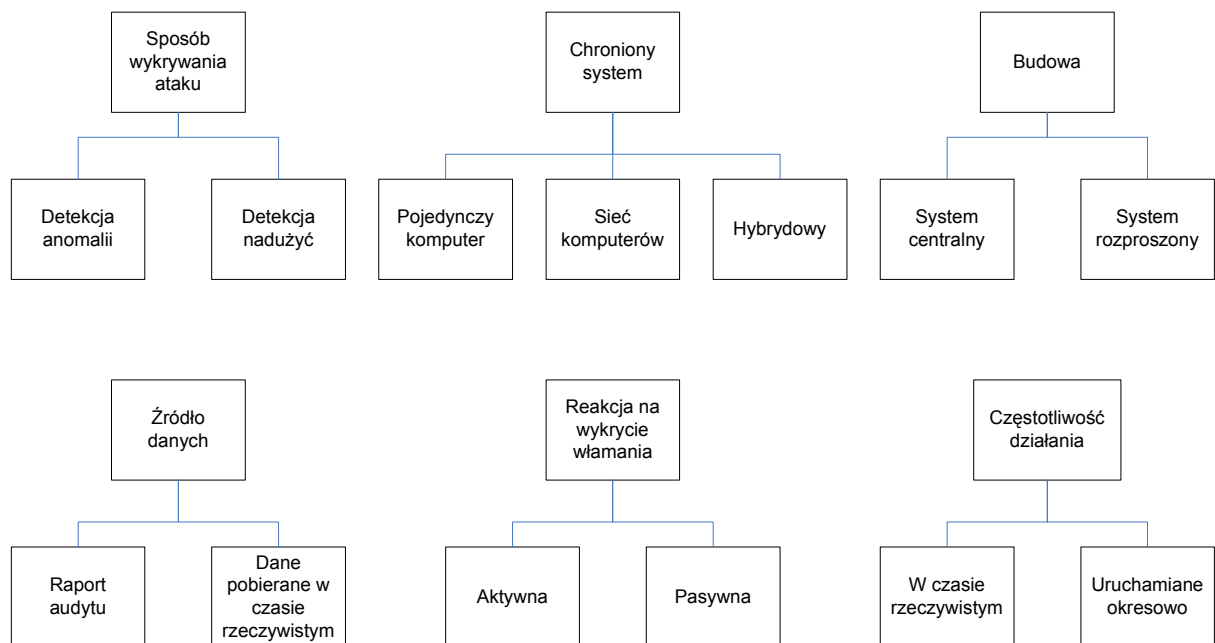
Taki typ systemu został stworzony w ramach tej pracy magisterskiej.

### 1.3.2. Systemy IDS oparte o wzorce predykowane (według:[Pieprzyk 2005], str. 419 i nast.)

Wykrywanie anomalii w oparciu o wzorce predykowane opiera się na założeniu, że można odróżnić normalne od anormalnego zachowania użytkownika na podstawie kolejnych wykonywanych przez niego czynności. W tym przypadku profil jest zbiorem typowych sekwencji zachowań. System IDS określa na podstawie tych zachowań czy dana sytuacja, w której można wyróżnić zapisaną wcześniej sekwencje jest zachowaniem normalnym czy też nie. Jeśli dana sekwencja była obserwowana wcześniej to wzrasta prawdopodobieństwo, że jest ona zachowaniem normalnym użytkownika. Wzorce predykowane najlepiej spisują się, gdy użytkownicy wykonują często powtarzające się kolejność zachowań.

### 1.4. Typy systemów IDS (Radosław Wężyk)

Klasyfikacje systemów IDS przedstawia rysunek 1:



Rysunek 2: Klasyfikacja systemów IDS. Źródło: [Dorosz 2/2002].

Systemy IDS możemy podzielić pod względem zasięgu działania oraz miejsca umieszczenia na systemy HIDS (ang. *Host-based IDS*, system IDS zorientowany na hosta) oraz NIDS (ang. *Network Based Intrusion Detection System*, system IDS zorientowany na sieć).

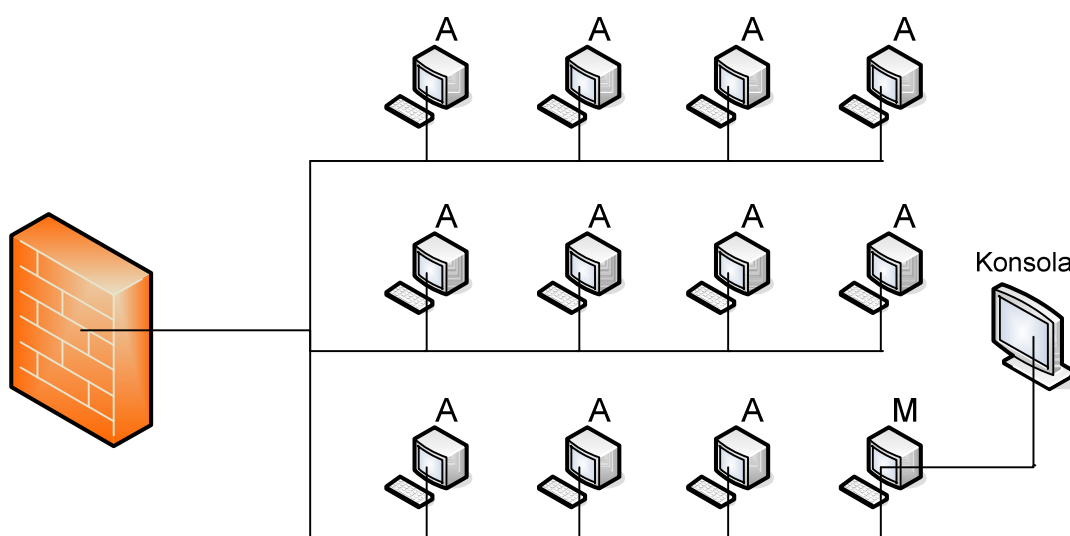
- HIDS

Systemy tego typu są to systemy IDS mające na celu ochronę komputera, na którym są zainstalowane. Z tego powodu instaluje się je tylko na konkretnym hoście, który ma zostać

poddany ochronie. Ten typ systemu stosowany jest zazwyczaj do ochrony ważnych punktów w sieci (np.: różnego rodzaju serwery usługowe). Instalowanie go na każdej maszynie w sieci było by bardzo czasochłonne, ponieważ wymagałoby osobnej instalacji i konfiguracji dla każdego hosta. HIDS może korzystać z systemu operacyjnego chronionego komputera, dzięki czemu może monitorować dostęp do konkretnych plików (np.: pliki systemowe, pliki dll, pliki wykonywalne), rejestru systemowego oraz czynności, które może wykonać tylko administrator systemu. HIDS może wygenerować alert w przypadku, gdy którykolwiek z tych zasobów został zmodyfikowany czy też osoba nieuprawniona próbowała uzyskać do niego dostęp. Może on również porównywać aktualny stan plików systemowych z zapamiętanym wcześniej. Współpraca systemów tego typu z systemem operacyjnym opiera się na analizie logów systemowych (zobacz: [Laing 2000], str. 8 i nast.). Z tego powodu, HIDS bazujący na współpracy z systemem operacyjnym hosta nie działa w czasie rzeczywistym. Przy dobrej konfiguracji może być on jednak bardzo blisko tego. HIDS nie wymaga zakupu dodatkowego sprzętu. Niewątpliwą zaletą tego systemu jest fakt, że możemy zweryfikować efekty ataku, który nastąpił. Jest to możliwe dzięki analizie logów systemu. Ponieważ HIDS pracuje na danym komputerze wykorzystuje część jego zasobów.

HIDS dzielą się na dwie kategorie (zobacz: [Axent 1999]):

1. HIDS typu „pojedynczy system” (ang. *single system*) panujący nad bezpieczeństwem jednej maszyny w oparciu o wykrywanie niebezpiecznych zdarzeń w logach
2. HIDS typu menadżer/agent – oprogramowanie agenta zainstalowane jest na każdym hoście w sieci i ma za zadanie monitorowanie logów i określonych plików. Każdy agent jest połączony z menadżerem, który z kolei podłączony jest do konsoli służącej monitorowaniu pracy sieci i wyświetlaniu informacji (zobacz rysunek 3). W momencie, gdy oprogramowanie agenta wykryje niebezpieczne zachowania, wysyłany zostaje alert do konsoli. Możliwe jest również podjęcie odpowiedniego działania przez oprogramowanie agenta np. wyłączenie procesu lub sesji związanej z atakiem, wyłączenie systemu itd. Taka konfiguracja pozwala administratorowi na wygodne zarządzanie i konfigurację oprogramowania agenta na wszystkich hostach, daje możliwość dodawania nowych reguł, aktualizację bazy sygnatur ataków oraz analizę logów ze wszystkich zarządzanych maszyn.



A – komputer z zainstalowanym oprogramowaniem agenta  
M – komputer z zainstalowanym oprogramowaniem menadżera

**Rysunek 3: Sieć chroniona przez HIDS typu menadżer/agent. Źródło: [Axent 1999].**

Główne zalety systemów HIDS to łatwa, wygodna i ściśle dostosowana konfiguracja, oraz swoboda reakcji w przypadku wykrycia włamania – tylko HIDS mogą wyłączyć stwarzający zagrożenie proces, wyłączyć konto użytkownika lub wykonać dowolną procedurę na danym komputerze.

Najważniejsze wady takich systemów to m.in. fakt, że nie zawsze działają w czasie rzeczywistym i zużywają zasoby komputera, na którym pracują. Ponadto należy pamiętać, że osoba włamująca się do systemu chronionego przez HIDS, w momencie zdobycia praw administratora może zarządzać procesami działającymi na komputerze np. ma możliwość wyłączenia agenta lub całego systemu detekcji intruzów.

- NIDS

Systemy NIDS są to systemy IDS, których celem jest ochrona danej sieci. Zasadniczą różnicą pomiędzy NIDS a HIDS jest to, że te drugie prowadzą obserwacje z punktu widzenia hosta, na którym pracują. NIDS jest to system monitorujący całą sieć, działający na zasadzie sniffera – przechwytyuje ramki i poddaje je analizie. Aby przechwytywanie było możliwe interfejs sieciowy musi być ustawiony w trybie ogólnym, czyli tzw. promiscuous. Wyjątkiem jest przypadek, gdy komputer, na którym pracuje system IDS jest bramą, wówczas ustawienie karty sieciowej w tryb ogólny nie jest warunkiem koniecznym działania IDSa. Należy jednak pamiętać, że przechwycony zostanie tylko ruch trasowany przez router. Warto, zatem w



takim przypadku również zadbać o to, by karta sieciowa była w trybie ogólnym, jeśli zależy na monitorowaniu ruchu wewnątrz sieci.

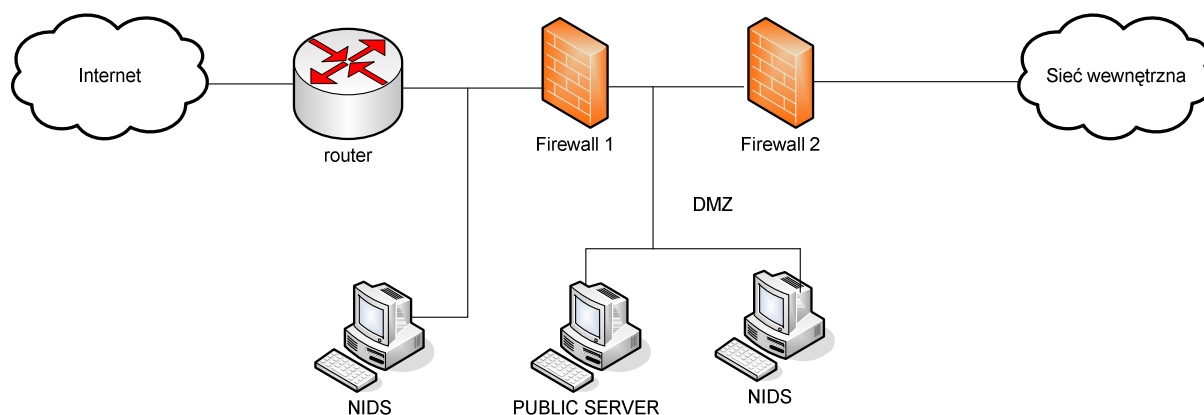
System tego typu pracujący na jednej maszynie może czuwać nad bezpieczeństwem całej sieci, a chronione komputery nie wymagają specjalnego oprogramowania. Działanie systemu typu Network-Based nie ogranicza się do analizy danych tylko w najwyższych warstwach modelu OSI/ISO. NIDSy monitorując zdarzenia na każdym poziomie warstwowego modelu OSI/ISO mają możliwość użycia wielu różnych technik związanych z właściwościami odpowiednich warstw. Stosując jednocześnie wiele różnych sposobów, w różnych warstwach zwiększają one wykrywalność ataków, a co za tym idzie bezpieczeństwo sieci. Najistotniejsze cechy tych systemów to m.in. działanie w czasie rzeczywistym oraz przezroczystość. NIDSy jednak, w przeciwieństwie do rozwiązań HIDS, nie mają możliwości wykonywania procedur na poszczególnych hostach i nie mają dostępu do właściwości ich systemu operacyjnego, logów, dziennika zdarzeń itd. Główną wadą systemów tego typu wg (według: [Pieprzyk 2005], str. 438 i nast.) jest brak informacji na temat wydarzeń w systemie operacyjnym hosta, do którego skierowany był dany ruch. System NIDS nie jest on w stanie określić czy system operacyjny przyjmie pakiet czy go odrzuci oraz jaki będzie efekt przyjęcia takiego pakietu. Zaletą tego typu IDS-ów jest to, że nie zależą od systemów operacyjnych działających na chronionych przez nie hostów, nie zabierają one ich zasobów systemowych ani nie wymagają instalowania na nich żadnego dodatkowego oprogramowania.

Aby NIDS mógł działać w czasie rzeczywistym musi być zainstalowany na odpowiednio wydajnej maszynie. Zwiększa to koszty uruchomienia takiego systemu.

Istnieją trzy podstawowe możliwości umieszczenia NIDS w sieci (zobacz: [Szmit 2005], str. 501):

- przed zaporą sieciową,
- w strefie DMZ,
- w sieci korporacyjnej,

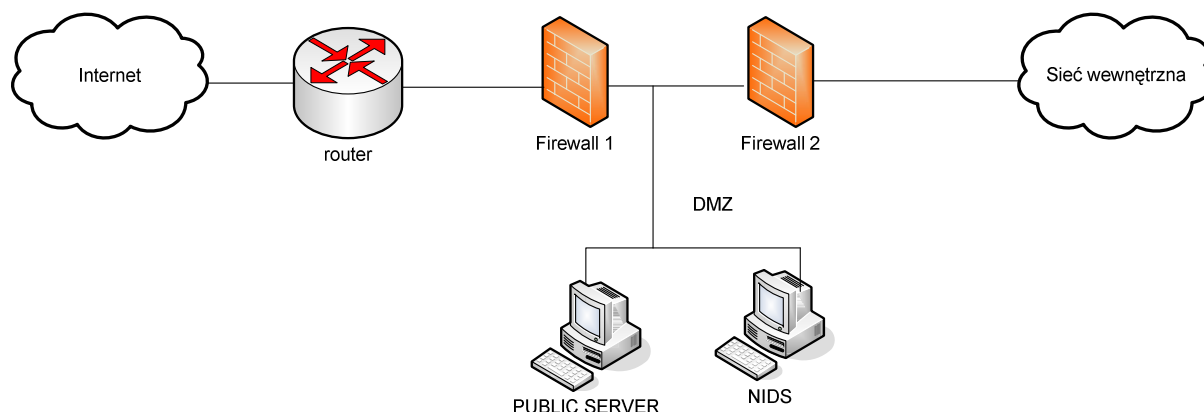
Jeśli NIDS zostanie umieszczony przed główną zaporą sieciową, tak jak na rysunku 4, to będzie on chronił sieć przed atakami z zewnątrz.



**Rysunek 4: NIDS przed główną zaporą sieciową (według: [Szmit 2005], str. 501)**

W tym punkcie występuje zazwyczaj bardzo duży ruch. NIDS powinien być zainstalowany na wydajnej maszynie, ponieważ będzie ona musiała w czasie rzeczywistym analizować duże ilości danych. Przy takim umiejscowieniu, będziemy otrzymywać informacje o wszystkich wykrytych próbach ataku, nawet o tych, które nie zostaną przepuszczone przez firewall, dlatego bardzo ważna jest poprawna konfiguracja. Złe skonfigurowanie może spowodować generowanie przez system wielu fałszywych alertów, co w efekcie zmniejszy skuteczność działania systemu.

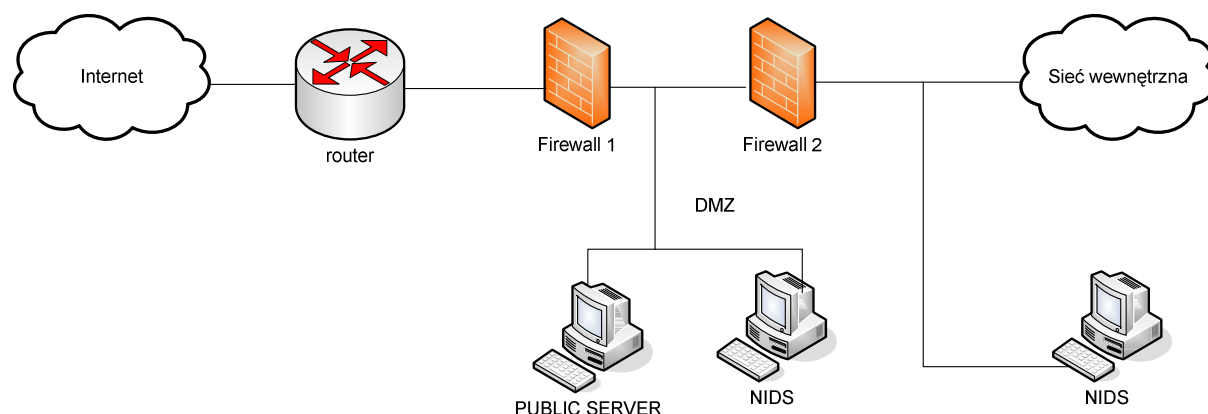
Umieszczając NIDS w obrębie strefy DMZ (ang. *de-militarized zone*) (zobacz: [Szmit, 2005], str. 501) zakłada się, że ruch jest już częściowo odfiltrowany poprzez pierwszą zaporę sieciową. Takie umiejscowienie przedstawia rysunek 5.



**Rysunek 5: NIDS w obrębie strefy DMZ (według:[Szmit 2005], str. 501)**

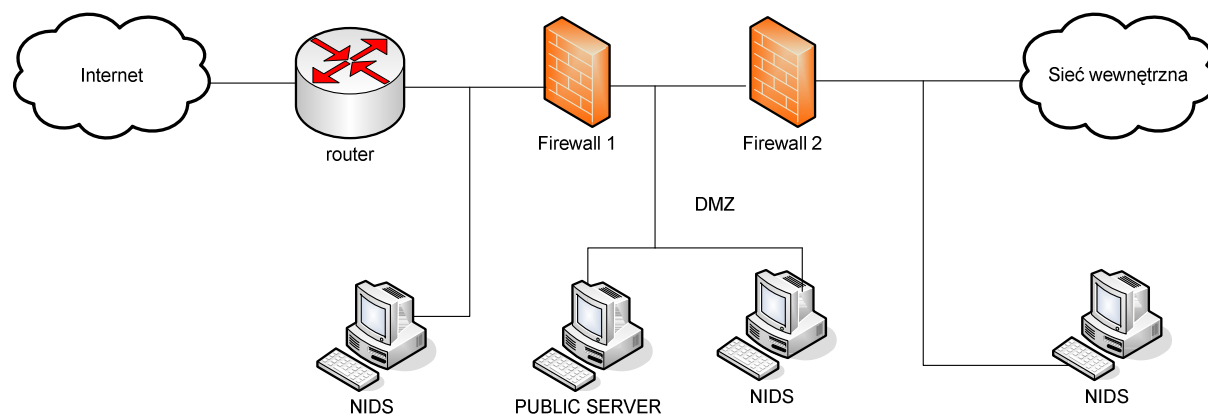
Dzięki temu można ograniczyć zbiór poszukiwanych ataków. Zmniejsza to wymagania wobec sprzętu.

Ostatnią możliwością umieszczenia NIDS jest zlokalizowanie go w obrębie sieci korporacyjnej (zob. rysunek 6).



**Rysunek 6: NIDS w obrębie sieci korporacyjnej (według: [Szmit 2005], str. 501)**

Takie umiejscowienie ma na celu monitorowanie ataków pochodzących z sieci lokalnej oraz wykrywanie ataków, które zostały przepuszczone przez firewall. Podobnie jak w poprzednim przypadku liczba możliwych w tym miejscu ataków jest mniejsza, przez co maleją wymagania sprzętowe. W przypadku tej lokalizacji generowanych jest mało fałszywych alarmów. Oczywiście, aby zapewnić jak największy poziom bezpieczeństwa powinno się umieścić NIDS w każdym z wymienionych powyżej punktów sieci. Taki przypadek przedstawia rysunek 7.



**Rysunek 7: NIDS w każdym z wymienionych powyżej punktów sieci (według: [Szmit 2005], str. 501)**

NIDS powinien pracować na innym hoście niż firewall. Jest to spowodowane tym, że atakujący może wykorzystać fakt, iż firewall wraz z systemem IDS znajdują się na jednym komputerze i wygenerować wielką liczbę fałszywych ataków. IDS zacznie je wszystkie rozpoznawać i stosownie reagować. Może to spowodować zużycie wszystkich zasobów danej maszyny, a co za tym idzie zabraknie ich dla firewalla. Efektem tego będzie odcięcie chronionej sieci od Internetu. Jest to przykład ataku typu DoS (zobacz: [Szmit 2005], str. 155 i nast.). W przypadku, gdy NIDS działa na osobnej maszynie przepływ pakietów zostanie zachowany.

## **1.5. Przechwytywanie ruchu sieciowego w systemach IDS**

### **(Radosław Wężyk)**

Praca systemów NIDS jest oparta o analizę przechwyconego ruchu sieciowego – na tym etapie aplikacja działa jak sniffer. Właściwa i wydajna akwizycja danych jest podstawą efektywnego działania systemu detekcji intruzów. Istnieje kilka sposobów przechwytywania ruchu opartych o właściwości urządzeń sieciowych, budowę sieci oraz zasoby sprzętowe.

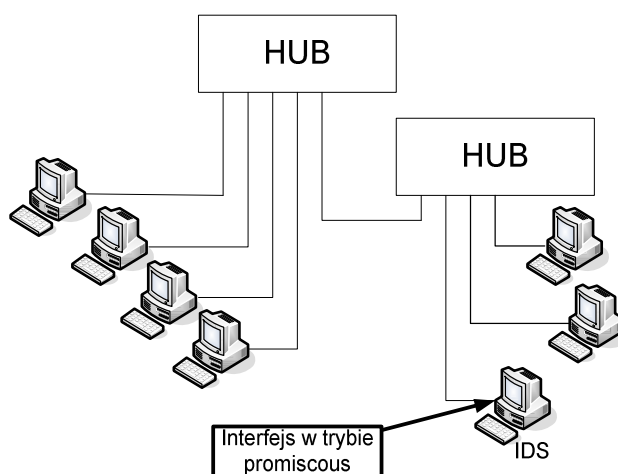
Systemy IDS ze względu na sposób akwizycji możemy podzielić na:

1. Systemy z jednym interfejsem sieciowym odpowiedzialnym za przechwytywanie ruchu:
  - Przechwytywanie ruchu w sieciach opartych na koncentratorach wykorzystujące interfejs sieciowy ustawiony w trybie promiscuous,
  - Przechwytywanie ruchu w sieciach opartych na przełącznikach klasycznych, przełącznikach z portem SPAN oraz sieciach VLAN,
  - Przechwytywanie ruchu przy użyciu urządzenia TAP (ang. *Test Access Port*).
2. Systemy typu in-line – co najmniej dwa interfejsy sieciowe:
  - Systemy in-line bez adresu IP,
  - Systemy GIDS.

### **1.5.1 Przechwytywanie ruchu w oparciu o tryb promiscuous.**

Ethernet zawsze jest zbudowany w logicznej topologii magistrali (zobacz: [Szmit 2005], str. 16), co oznacza, że każda ramka (właściwie jej kopia) trafia do każdego komputera w obrębie danej domeny kolizyjnej. To czy ramka jest zaadresowana do danego komputera sprawdza karta sieciowa. Dzięki możliwości ustawienia interfejsu sieciowego w specjalny tryb, nazwany trybem ogólnym (ang. *promiscuous*) możemy zapewnić, że cały ruch trafi do

naszego komputera. W przypadku sieci opartej wyłącznie na hubach podsłuch i przechwytywanie są proste w związku z faktem, że hub rozsyła otrzymane ramki na wszystkie porty (rys. 8).



Rysunek 8: Ruch z całej sieci jest przechwycony przez system IDS z interfejsem w trybie promiscuous.  
Źródło: opracowanie własne.

### 1.5.2. Przechwytywanie ruchu w sieciach zbudowanych w oparciu o przełączniki (ang. *switch*)

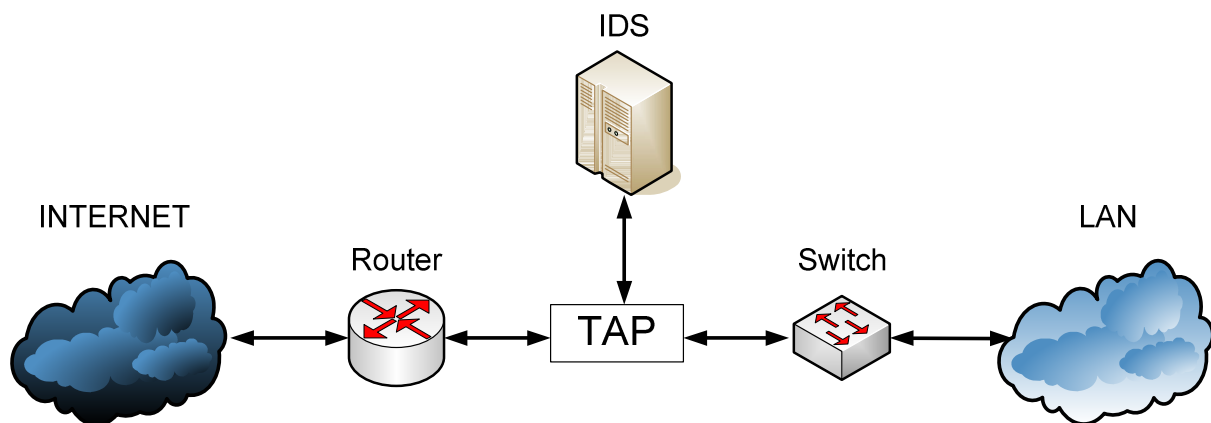
Switche są urządzeniami sieciowymi, które przesyłają ramki wyłącznie między dwoma portami (za wyjątkiem ramek broadcastowych). Decyzje, która ramka ma trafić na określony port podejmuje na podstawie tablicy adresów MAC. Switch wypełnia tablicę adresów od momentu włączenia zasilania. Efektem tego, przechwytywanie i podsłuch w klasyczny sposób (przy wykorzystaniu wyłącznie trybu promiscuous) stały się niemożliwe. Istnieją jednak metody umożliwiające podsłuch w takich sieciach. Są to m.in. ARP Spoofing, MAC flooding, duplikacja adresu MAC (zobacz: [Szmit 2005] str. 22 i nast.). W systemach IDS wykorzystuje się jednak głównie przełączniki z portem SPAN (ang. *Switch Port Analyzer*). Port SPAN umożliwia przesłanie na niego kopii ruchu z innego portu lub nawet ze wszystkich portów (zobacz: [Baker 2004] str. 87). IDS podłączony do portu SPAN otrzyma ramki, ale aby je przetworzyć interfejs musi pracować w trybie ogólnym. Aby przechwytywać cały ruch kierowany do/z sieci należy zadbać o to, aby IDS podłączony był do portu SPAN głównego (ang. *root*) switcha w monitorowanej sieci. Głównym problemem takiego rozwiązania jest przeciążenie switcha w momencie, gdy chcemy otrzymywać kopię ramek z większej liczby portów. Efektem przeciążenia przełącznika jest gubienie pakietów, które mają trafić na port SPAN oraz spadek wydajności sieci.

### 1.5.3. Przechwytywanie ruchu w sieciach VLAN (ang. Virtual Local Area Network).

Sieć VLAN jest to podział jednej sieci fizycznej na kilka sieci logicznych. Do ich tworzenia wykorzystuje się zarządzalne przełączniki sieciowe. Tworzenie sieci logicznej w najprostszych przełącznikach polega na odpowiednim pogrupowaniu portów przełącznika. W switchach zgodnych z IEEE 802.1Q możliwe jest tworzenie VLANów poprzez znakowanie ramek. Pozwala to na transmitowanie danych z wielu VLANów poprzez jedno łącze fizyczne (tak zwany trunking). Rozwiązanie problemu przechwytywania ruchu w sieciach VLAN opiera się o wykorzystanie switchy z portem SPAN. Takie rozwiązanie powoduje, że do systemu IDS docierają dane ze wszystkich VLANów. Należy zadbać, aby IDS był podłączony do głównego przełącznika w sieci. Ponadto w niektórych switchach należy pamiętać, aby port SPAN należał do wszystkich VLANów.

### 1.5.4. TAP (ang. *Test Access Port*)

TAP jest rozwiązaniem sprzętowym wstawianym bezpośrednio do segmentu sieci, którego funkcją jest wysłanie przepływającego przez nie ruchu na port, do którego podłączone jest urządzenie monitorujące np. IDS lub analizator protokołów.



Rysunek 9: TAP. Źródło: [16].

TAP posiada trzy interfejsy sieciowe: wejściowy, wyjściowy oraz interfejs łączący z urządzeniem monitorującym. Urządzenia TAP dzielą się na:

- Urządzenia przygotowane do pracy w sieci światłowodowej (ang. *Fibre TAPs*) – rozdzielają sygnał na dwa strumienie tak, aby jeden trafił na wyjściowy interfejs a drugi na interfejs monitorowany.
- Urządzenia przygotowane do pracy w sieciach opartych o medium miedziane (ang. *Copper TAPs*). Sygnał w tym przypadku jest wzmacniany tak aby mógł zostać odebrany zarówno przez system monitorujący jak i docelową sieć.

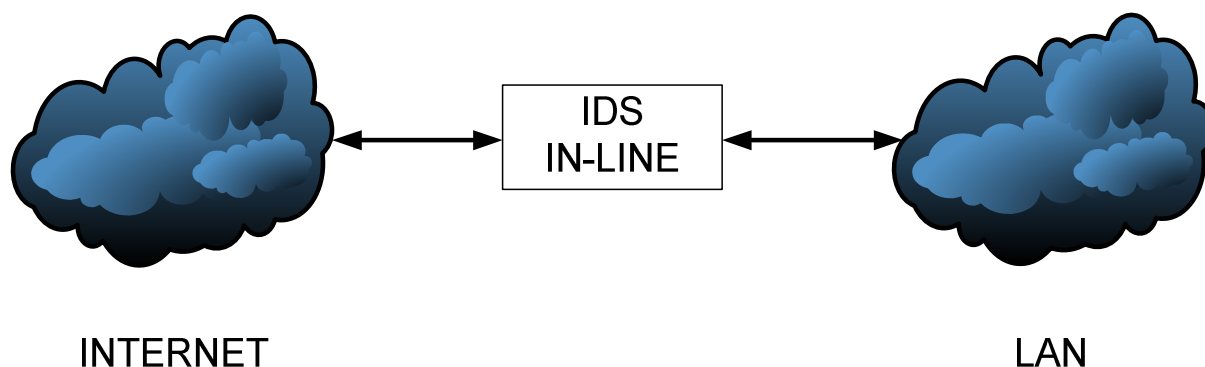
Żadne z powyższych rozwiązań nie wprowadza opóźnień ani nie wpływa na zawartość i strukturę danych.

Zalety rozwiązania TAP są następujące:

1. Na interfejs, do którego podłączony jest system monitorujący trafia cały ruch, który pojawi się na interfejsie wejściowym włączając błędy, uszkodzone i zniekształcone ramki itd.
2. TAP swoim działaniem nie wprowadza opóźnień i nie wpływa na wydajność sieci.
3. Wyklucza problem gubienia pakietów, jaki może mieć miejsce w switchach z portem SPAN o różnych prędkościach interfejsów. Wiele przełączników posiada kilka portów 10/100Mbps oraz jeden port 1Gbps lub kilka portów gigabitowych i jeden 10Gbps. W takich przypadkach rzadko wykorzystuje się najszybszy port jako port SPAN, więc łatwo sobie wyobrazić, że np. port SPAN 100Mbps przy dużym obciążeniu sieci nie będzie mógł przesłać całego ruchu z portu 1Gbps.
4. TAP jest niewidoczny dla sieci i atakującego. Jego interfejsy nie posiadają adresów IP i MAC.
5. Gdy TAP zostanie pozbawiony zasilania nie załamuje działania sieci – ruch dalej płynie przez niego.

#### **1.5.5. Rozwiązania typu IN-LINE.**

In-line jest sposobem umieszczenia systemu IDS w sieci zapewniającym, że cały ruch płynący do/z sieci przepływa przez system.



**Rysunek 10: Ruch sieciowy generowany między siecią LAN i Internetem zawsze przepływa przez system IDS typu in-line. Źródło: opracowanie własne.**

W przypadku IDS typu in-line pracującego w trybie stealth interfejsy sieciowe nie posiadają adresów IP. Zapewnia to częściową przeźroczystość i ukrycie systemu przed atakującymi. Często systemy tego typu posiadają trzeci interfejs sieciowy z przypisanym adresem IP służący do zarządzania systemem oraz komunikację np. z zewnętrzną bazą danych lub narzędziami służącymi wizualizacji wyników pracy.

#### **1.5.6. Systemy GIDS (ang. *Gateway Intrusion Detection System*).**

System IDS pracujący na komputerze będącym bramą (ang. *gateway*) zapewniającym dostęp do Internetu. Cały ruch do/z Internetu przepływa przez bramę, a więc aplikacja IDS ma możliwość monitorowania całego ruchu. Jest to jeden z przypadków, kiedy karty sieciowe nie muszą pracować w trybie ogólnym. Jest to spowodowane tym, że cały ruch kierowany jest do elementów systemu operacyjnego bramy odpowiedzialnych za routing w celu podjęcia decyzji routingu.

#### **1.5.7. Ukrywanie systemu IDS oraz praca interfejsów w trybie stealth (zobacz: [Baker 2004], str. 90)**

Zwykle użytkownicy konfiguruje IDS z dwoma interfejsami sieciowymi – jeden służy do przechwytywania ruchu, drugi do zarządzania systemem oraz wysyłania alertów. Możliwe jest ustawienie interfejsu nasłuchującego w sieci w tryb stealth w celu ukrycia systemu przed atakującym. Interfejs w trybie stealth nie posiada adresu IP i jest ustawiony w trybie promiscuous.

Innym sposobem wykrycia systemu IDS jest przechwycenie ruchu sieciowego zawierającego alerty. Jest to bardzo niebezpieczne, ponieważ w przypadku, gdy informacje te nie są szyfrowane atakujący może łatwo określić zestaw reguł systemu detekcji oraz



zlokalizować szczególnie chronione hosty w sieci. Można się przed tym ustrzec szyfrując ruch zawierający alerty lub stworzyć odrębną sieć służącą do zarządzania siecią i przesyłania alertów.

Aby system IDS był zupełnie niewidoczny, a karta sieciowa niemożliwa do wykrycia, można uniemożliwić karcie wysyłanie żadnych informacji na poziomie elektrycznym (zobacz: [Rehman 2003], str. 20). Na przykład w sieci Ethernet opartej o skrętkę należy zmodyfikować końcówkę RJ-45 – odciąć parę odpowiedzialną za transmisję (TX – piny 1 i 2) nie naruszając pary odpowiedzialnej za odbiór (RX – piny 3 i 6).

## **1.6. Odpowiedzi systemów IDS (Maciej Skowroński)**

W momencie, gdy IDS wykryje atak, reaguje na niego. Reakcji, jakie mogą być podjęte jest bardzo wiele, poczynając od wysłania powiadomienia o ataku pod wcześniej ustawiony adres aż po zablokowanie atakującego. Odpowiednia reakcja na zagrożenie jest sprawą kluczową dla bezpieczeństwa. Komercyjne systemy IDS oferują bardzo dużo możliwości konfiguracji tego parametru.

Ogólnie reakcje można podzielić na trzy typy (zobacz: [Bace], str. 20 i nast.):

- odpowiedzi aktywne,
- odpowiedzi pasywne,
- odpowiedzi mieszane.

Odpowiedzi aktywne są to akcje podejmowane w momencie wykrycia zagrożenia. Pierwsza z nich polega na tym, że po wykryciu ataku, IDS stara się zebrać jak najwięcej informacji na temat atakującego oraz przeprowadzanego ataku. Cel ten realizuje poprzez zwiększenie czułości swoich sensorów. Zebrane dane o atakującym ułatwią późniejszą analizę przebiegu ataku, dzięki czemu można będzie zabezpieczyć się przed nim w przyszłości. Dane te mogą również mieć duże znaczenie w śledztwie prowadzonym przez organy ścigania. Drugim typem odpowiedzi aktywnej jest zablokowanie atakującego. IDS sam nie jest w stanie zablokować atakującego, jednak może on podjąć działania, które mogą doprowadzić do zatrzymania atakującego lub do jego odstraszenia. Takimi działaniami mogą być na przykład „wkładanie” pakietów TCP z ustawioną flagą RST do połączenia atakującego, przekonfigurowanie routerów lub firewalli w celu zablokowania pakietów pochodzących od atakującego lub też portów, na których pracują usługi wykorzystywane przez napastnika. W ekstremalnych przypadkach IDS może zablokować cały ruch na danym

interfejsie sieciowym. Można wyróżnić też trzeci typ odpowiedzi aktywnej. Jest nim podjęcie działań ofensywnych przeciwko atakującemu. Celem tego typu zachowania jest zdobycie informacji o atakującym oraz jego odstraszenie. Ten typ odpowiedzi jest jednak ryzykowny, ponieważ może on być niezgodny z prawem. Może on też spowodować szkody niewinnym użytkownikom sieci. Taka reakcja może też spowodować nasilenie się ataków ze strony atakującego. Z tych powodów ten typ odpowiedzi nie jest implementowany w komercyjnych systemach IDS.

Odpowiedzi pasywne dostarczają informacji użytkownikom systemu na temat zaistniałej sytuacji i pozostawiają podjęcie odpowiednich kroków ludziom. Wiele komercyjnych rozwiązań polega wyłącznie na odpowiedziach tego typu. Przykładem odpowiedzi tego typu są alarmy i powiadomienia. IDS generuje je w celu poinformowania osób nadzorujących o wystąpieniu ataku. Najprostszą formą takiego powiadomienia jest wyświetlenie okienka na konsoli systemu IDS. Powiadomienia mogą być wysyłane również na telefon komórkowy bądź pager. Można spotkać się też z możliwością wysłania emaila, jednak funkcja ta może być łatwo zablokowana przez atakującego. Niektóre komercyjne systemy IDS potrafią zgłaszać ataki za pomocą systemu zarządzania siecią przy wykorzystaniu protokołu SNMP (ang. *Simple Network Management Protocol*). Dzięki takiemu rozwiązaniu w momencie wykrycia ataku możliwe jest szybkie przystosowanie sieci do przyjęcia tego ataku.

Odpowiedzi mieszane łączą w sobie możliwości oferowane przez odpowiedzi aktywne i pasywne.

### **1.7. Problem fałszywych alarmów i gubienia pakietów (zobacz: [1])** **(Radosław Wężyk)**

O jakości systemu IDS świadczy m.in. liczba fałszywych alarmów (ang. *false positives*) oraz liczba nie wykrytych ataków (ang. *false negatives*). Główną przyczyną gubienia pakietów jest fakt, że systemy nie nadążają analizować wszystkich danych w przypadku dużego obciążenia sieci.

Niezwykle istotna jest poprawna interpretacja danych. Systemy zabezpieczające powinny prawidłowo analizować dane – znać cechy protokołów w odpowiednich warstwach modelu ISO/OSI. System musi wziąć pod uwagę charakterystyczne właściwości poszczególnych protokołów – fragmentacje IP, brak potwierdzeń poprawności przesłania pakietów IP, brak kolejności dochodzących pakietów IP, retransmisje segmentów TCP itd.

Aby zminimalizować liczbę fałszywych alarmów oraz nie wykrytych ataków IDS powinien zadbać o to, żeby baza sygnatur była często aktualizowana, a porównania precyzyjnie i szczegółowo skonstruowane.

Zwiększenie wydajności systemów IDS uzyskuje się przez implementację dedykowanych sterowników dla kart sieciowych. Dzięki temu ramki trafiają do oprogramowania IDS z pominięciem stosu TCP/IP systemu operacyjnego.

Duży wpływ na wydajność systemów IDS mają dość skomplikowane i czasochłonne algorytmy opierające swe działanie na bazach sygnatur. Przeszukiwanie i porównywanie wszystkich dostępnych sygnatur jest czynnością czasochłonną i bezpośrednio wpływa na zmniejszenie wydajności systemów analizujących. Warto, zatem wprowadzić dodatkowe kryteria porównań i korzystania z odpowiednich rodzajów sygnatur dla określonego typu ruchu lub usług sieciowych – pozwoli to odrzucić część zbędnych porównań. Pamiętać należy również, że duży wpływ na wydajność systemu mają zasoby sprzętowe – im wyższa moc obliczeniowa jednostki, na której pracuje IDS tym lepiej. Zaawansowanym algorytmom, ale i również obsłudze rozbudowanych baz sygnatur oraz analizie ruchu mocno obciążonych sieci sprostać mogą tylko szybkie maszyny. Dla przykładu rozwiązania sprzętowo-programowe Cisco pracujące w sieciach Ethernet 10/100Mbit posiadają parametry rzędu: procesor 1.3 GHz i 1GB RAM. Wprowadzenie systemów IDS typu in-line pozwala na uniknięcie problemu gubienia pakietów i analizę danych w czasie rzeczywistym (zobacz: [11]). Podstawą takiego systemu jest fakt, że ruch wchodzi na jeden interfejs urządzenia, na którym jest zainstalowany IDS, następuje analiza a następnie wychodzi drugim interfejsem. Obecnie IDSy in-line mają wydajność 100Mbit/s – 500Mbit/s, co umożliwia efektywną pracę w sieciach Ethernet 100Mbit/s i w mało obciążonych sieciach gigabitowych. Przykładami systemów typu In-line są: Check Point SmartDefence, ISS RealSecure Guard, NetScreen IDP.

## **1.8. Architektura systemów IDS (Maciej Skowroński)**

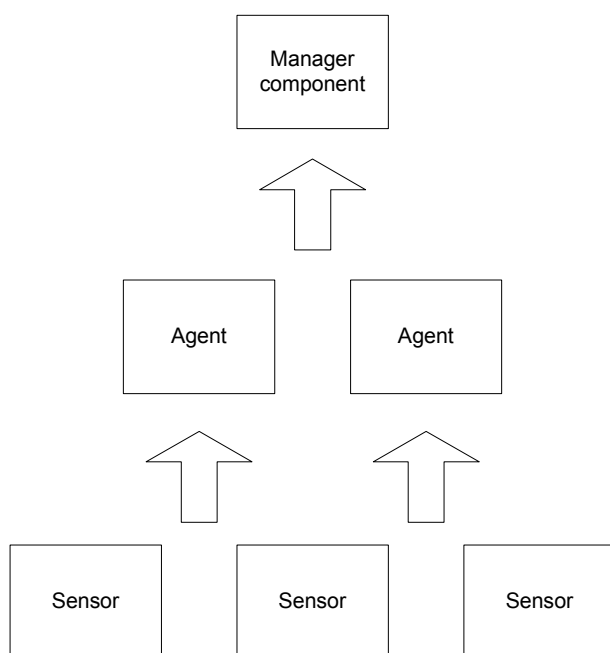
Można wyróżnić trzy różne typy architektur używane do konstrukcji systemów detekcji intruzów (zobacz.: [Endorf 2004], rozdział: Tiered Architectures):

- jednopoziomowa (ang. *Single-Tiered*),
- wielopoziomowa (ang. *Multi-Tiered*),
- Peer-to-Peer.

Architektura jednopoziomowa jest rozwiązaniem najprostszym. Jej cechą charakterystyczną jest to, że komponenty systemu same analizują przechwycone dane i nie przesyłają ich dalej. Przykładem systemu stworzonego w tej architekturze jest HIDS, który pobiera logi systemowe i je analizuje. Zaletą tej architektury jest jej prostota oraz niezależność od innych komponentów. Jej wadą jest brak współpracy pomiędzy komponentami, co zmniejsza możliwości systemu.

Komunikację pomiędzy poszczególnymi komponentami systemu detekcji intruzów wykorzystuje architektura wielowarstwowa. Schemat takiego systemu przedstawia rysunek 11. Podstawowymi elementami wchodzącymi w skład tej architektury są:

- czujniki (ang. *sensors*),
- analizatory lub agenci (ang. *analyzers, agents*),
- element zarządzający (ang. *Manager component*).



**Rysunek 11: Architektura wielowarstwowa. Źródło: [Endorf 2004].**

Czujniki są podstawowym i bardzo ważnym elementem systemów detekcji intruzów (zobacz: [Endorf 2004], rozdział: Sensors). Znajdują się one na samym dole architektury systemu. Ich zadanie jest z pozoru bardzo proste. Sprowadza się ono do zbierania danych a następnie przesyłania ich dalej. Można wyróżnić dwa typy czujników:

- network-based,
- host-based.

Czujniki typu network-based są to zazwyczaj urządzenia lub programy, które przechwytyują ruch sieciowy. Wielką zaletą tego typu czujników jest fakt, iż nie ma znaczenia liczba hostów, które znajdują się w sieci, z której czujnik zbiera informacje. Sensorów tego typu można użyć do monitorowania nawet całego ruchu wejściowego i wyjściowego w danej sieci. Teoretycznie tak skonfigurowany czujnik może wykryć zagrożenie każdego z hostów będących w sieci, którą monitoruje. Jeśli zostanie on poprawnie skonfigurowany, to nie będzie generował zbyt dużego ruchu w sieci. Przykładem programu, który może być wykorzystany jako czujnik jest tcpdump (zobcz: [I8]). Program ten zapisuje dane, które dotarły do interfejsu sieciowego pracującego w trybie promiscuous. Dane te mogą zostać następnie wykorzystane przez inne aplikacje, w szczególności przez IDS. Używanych jest wiele innych rozwiązań. Czujniki typu host-based, podobnie jak czujniki network-based, mogą przechwytywać dane, które dotarły do interfejsu sieciowego i przekazywać je dalej. Karta sieciowa nie jest jednak ustawiana w tryb promiscuous, przez co przechwytyuje on tylko dane zaadresowane do danego hosta. Większość sensorów typu host-based w wyniku swojej pracy produkuje logi. Są one później analizowane przez oprogramowanie działające na tej samej lub innej maszynie.

Czujniki przekazują zebrane przez siebie informacje do agentów, zwanych też analizatorami (zobacz: [Endorf 2004], rozdział: Agents). Pierwszy raz użyto ich w połowie lat dziewięćdziesiątych. Główną funkcją analizatora jest analiza danych w celu wykrywania anomalii w ruchu i naruszeń zasad bezpieczeństwa. Agenci są zazwyczaj wyspecjalizowani do pełnienia tylko jednej funkcji, na przykład analizy ruchu TCP, liczby prób nawiązania połączenia czy też czasu ich trwania. Zaletą tego rozwiązania jest to, że jeśli jeden z nich z jakiegoś powodu przestanie działać, nie wpłynie to na funkcjonalność innych. Oznacza to również, że można dodać lub usunąć nowy analizator bez konieczności przerywania pracy systemu. Rozwiązanie takie pozwala na dobieranie liczby pracujących elementów do aktualnych potrzeb. Agenci komunikują się między sobą wykorzystując do tego specyficzny protokół, pomimo że każdy z nich pracuje osobno i to zazwyczaj na osobnej maszynie. Każdy agent może otrzymywać jedynie część przechwyconych informacji z danego systemu, sieci czy też urządzenia. Jeśli analizator rozpozna jakiś atak lub anomalie, informuje o tym fakcie pozostałych agentów. Dzięki możliwości wzajemnej komunikacji, taka informacja wraz z aktualnie przetwarzanymi danymi pozwala stwierdzić agentowi, czy i u niego nie wystąpiła podobna próba ataku.

Agenci generują czasem również fałszywe alarmy. W takim przypadku błędne rozpoznanie zagrożenia ataku przez jednego z nich wpływa na całą resztę. Problem

falszywych alarmów jest jednym z największych problemów systemów detekcji intruzów. Zaawansowane systemy IDS pozwalają przeglądać alarmy za pomocą konsoli zarządzającej i dają operatorowi możliwość usunąć te fałszywe.

Pojawienie się systemów opartych na agentach było wielkim krokiem w historii rozwoju systemów detekcji intruzów. Rozwiązanie to przyniosło ze sobą wiele korzyści. Systemy stały się elastyczne i łatwo można je było dostosowywać do danych warunków. Użycie wielu jednocześnie pracujących agentów zwiększyło też efektywność systemów. Niezależność analizatorów zwiększyła niezawodność systemu.

Umieszczenie agenta jest prostsze niż poprawne umieszczenie sensora (zobacz: [Endorf 2004], rozdział: Agents Deployment Considerations). W przypadku HIDS, każdy analizator monitoruje tylko hosta, na którym pracuje. W przypadku systemów NIDS agenci umieszczani są w dwóch lokacjach. W miejscu gdzie będą najbardziej skuteczni lub gdzie będą najbardziej bezpieczni. Skuteczność agentów zależy w dużej mierze od umieszczenia czujników. Im bardziej lokalnie umieszczone są czujniki względem analizatorów, tym większa wydajność. Najlepszym rozwiązaniem jest posiadanie agenta i czujnika w obrębie tej samej sieci. Kwestia bezpieczeństwa agenta jest bardzo istotna (zobacz: [Endorf 2004], rozdział: Agent Security Considerations). W przypadku skutecznego ataku na niego atakujący może zatrzymać jego prace oraz może zdobyć informacje użyteczne do dalszego atakowania elementów systemu IDS. Dlatego przy umieszczaniu analizatora podobnie jak i czujnika należy zwrócić uwagę na jego bezpieczeństwo. Najlepiej gdyby agent pracował na dedykowanej dla niego maszynie. Zmniejsza to ryzyko dostania się do niego poprzez wykorzystanie luki w innym pracującym na danym komputerze oprogramowaniu. Szyfrowanie wymiany informacji pomiędzy poszczególnymi elementami systemu również może znacznie podnieść bezpieczeństwo.

Ostatnim elementem architektury trójwarstwowej jest element zarządzający (ang. *manager component*). To właśnie do niego agenci wysyłają informację o wykryciu ataku.

Manager wykonuje wiele funkcji (zobacz: [Endorf 2004], rozdział: Multi-Tiered Architectures):

- zbiera i wyświetla alerty otrzymane od agentów na konsoli oraz przechowuje wszystkie je w bazie danych,
- powiadamia administratora o wystąpieniu danego zdarzenia przy użyciu pagera lub wysyłając wiadomość sms,
- odszukuje dodatkowe informacje na temat zdarzenia, które miało miejsce,

- wysyła polecenia do danego hosta mające na celu zatrzymanie działającego na nim procesu,
- wysyła polecenia do firewalla wprowadzenia zmian w ustawieniach jego list dostępu,
- umożliwia użytkownikowi zarządzanie systemem.

Centralny punkt zarządzający systemu ułatwia sterowanie pracą całego systemu oraz analizowanie sytuacji. Dzięki niemu administrator może zarządzać sensorami, agentami i innymi podłączonymi systemami bez bezpośredniego dostępu do nich.

Zalety systemu trójwarstwowego to przede wszystkim większa efektywność oraz skuteczność. Systemy tego typu są w stanie zapewnić efektywność wykrywania, której nie są mogą osiągnąć systemy zbudowane na prostszej, jednowarstwowej architekturze. Dają one znacznie lepszy pogląd na zabezpieczenia danej sieci. Ich głównymi wadami są wysokie koszty oraz skomplikowanie systemu. Instalacja systemu trójwarstwowego oraz zarządzanie nim nie jest rzeczą łatwą.

Rozwiązanie bazujące na agentach nie jest oczywiście pozbawione wad. Największą wadą jest zapotrzebowanie na zasoby komputerów, na których uruchomieni są agenci. Potrzeba też dodatkowych interfejsów do komunikacji z innymi elementami systemu IDS. Efektem tego jest wzrost zapotrzebowania na moc obliczeniową oraz ilość pamięci, a co za tym idzie – wzrost kosztów. Problem stanowią też fałszywe alarmy generowane przez agentów.

Kolejną architekturą systemów IDS jest architektura Peer-to-Peer (zobacz: [Endorf 2004], rozdział: Peer-to-Peer Architecture), która wykorzystuje wymianę informacji pomiędzy systemami pracującymi na osobnych hostach. Przykładem takiej komunikacji może być komunikacja pomiędzy firewallami, routerami czy też przełącznikami. Informacja przekazywana jest pomiędzy urządzeniami bez potrzeby używania centralnego serwera. Główną zaletą tego rozwiązania jest jego prostota. Każdy system może uczestniczyć w przetwarzaniu informacji i wykorzystywać wyniki pracy innych systemów. Wadą tego rozwiązania jest brak zaawansowanych możliwości. Jednak funkcjonalność jest większa niż w pojedynczym systemie opartym na architekturze jednowarstwowej.

Jak widać z powyższego, architektura, w jakiej został stworzony dany IDS ma duży wpływ na jego możliwości i efektywność wykrywania ataków a także na jego koszt i stopień trudności jego konfiguracji.

## **1.9. Metody komunikacji między elementami systemu IDS (Maciej Skowroński)**

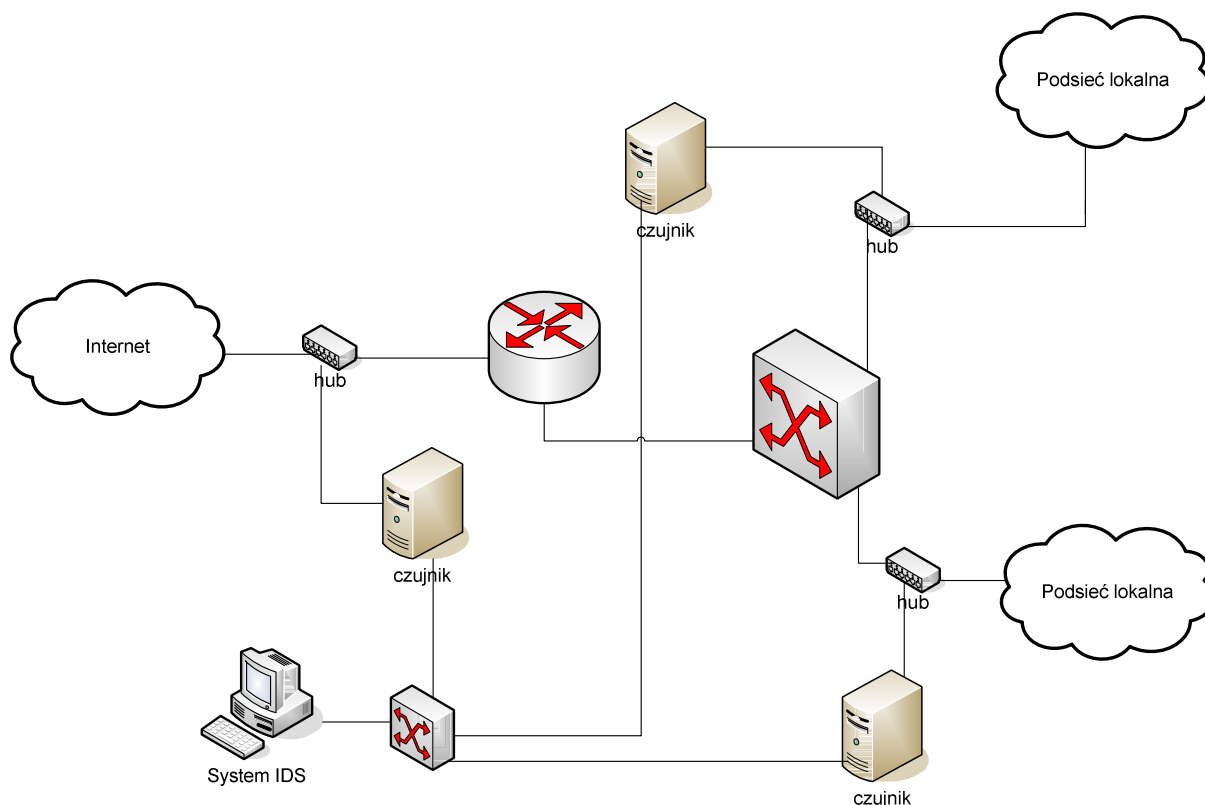
Poszczególne elementy systemu IDS do komunikacji między sobą mogą wykorzystywać:

- Istniejącą sieć, którą monitorują,
- Sieć dedykowaną.

Wykorzystanie istniejącej sieci jest rozwiązaniem najprostszym. Nie wymaga ono zakupu dodatkowego sprzętu ani dodatkowej konfiguracji. Informacje przesyłane pomiędzy poszczególnymi elementami systemu poruszają się w istniejącej już sieci, którą przesyłane są też inne informacje. Taka sytuacja stwarza możliwość przechwycenia lub wysłania spreparowanych przez siebie danych. Jest to niewątpliwe zagrożenie dla skutecznego działania systemu IDS. Atakujący może wykorzystać tę cechę do sparaliżowania pracy systemu poprzez wygenerowanie bardzo dużej ilości fałszywych alarmów. Sytuacji takiej można próbować zapobiec poprzez wprowadzanie szyfrowania danych. Metoda ta zwiększa obciążenie monitorowanej sieci.

Sieć dedykowana wymaga większych nakładów finansowych. Ta metoda komunikacji pomiędzy elementami systemu IDS wymaga stworzenia odrębnej sieci komputerowej przewidzianej wyłącznie dla systemu IDS. Sytuację taką przedstawia rysunek 12. Wszystkie elementy systemu zostają do niej dołączone poprzez zainstalowane w nich dodatkowe interfejsy sieciowe. Interfejsy służące do zbierania danych ustawia się tak, aby nie mogły wysyłać żadnych informacji do chronionej sieci (na przykład przez modyfikacje wtyczek).





**Rysunek 12: Schemat systemu IDS z dedykowaną siecią. Źródło: opracowanie własne.**

Taka metoda komunikacji jest zdecydowanie bezpieczniejsza od wykorzystania monitorowanej sieci. Praktycznie uniemożliwia atakującemu przechwytywanie informacji krążących wewnątrz systemu IDS. Dodatkowo metoda ta nie zwiększa obciążenia chronionej sieci. Jest jednak zdecydowanie kosztowniejsza, ponieważ wymaga zakupu dodatkowego sprzętu, jego instalacji oraz konfiguracji. Z punktu widzenia bezpieczeństwa sieci jest ona zdecydowanie lepsza, ponieważ nie przesyła danych ogólnie dostępnymi kanałami.

### **1.10. Systemy IPS (Maciej Skowroński i Radosław Wężyk)**

Systemy IPS (ang. *intrusion prevention system*) są rozwinięciem systemów IDS. Systemy IDS są zazwyczaj pasywne i generują tylko duże ilości alertów. Nie potrafią one samodzielnie jednak zablokować ataku i wymagają do tego zewnętrznych aplikacji. Systemy IPS natomiast, dzięki swojej integracji z zaporą sieciową, mogą zablokować podejrzany ruch w momencie wykrycia go. Każdy pakiet po pojawieniu się w systemie IPS jest porównywany z bazą sygnatur (o ile oczywiście mówimy o IPSie opartym o detekcję nadużyć) i – w przypadku, gdy okaże się on próbą ataku bądź też jej częścią – zostaje zablokowany i nie przepuszczony do środka chronionej sieci. Systemy IPS mogą również modyfikować zawartość przechodzących przez nie pakietów. Pozwala to na zapobiegania atakom jak i na

zmylenie atakującego. Oczywiście – podobnie jak w systemach IDS wzbogaconych o systemy aktywnej odpowiedzi – sprawdzanie i ewentualna modyfikacja każdego przechodzącego pakietu zajmuje czas, co wpływa na opóźnienie ruchu.

Podobnie jak w systemach IDS, można wyróżnić dwa podstawowe rodzaje systemów IPS:

- Host IPS (HIPS),
- Network IPS (NIPS).

Systemy HIPS podobnie jak i HIDS instalowane są na hoście, który ma być chroniony (zobacz: [Rash 2005], str. 10 i nast.). Zwykle są to rozwiązania programowe. Integrują się one z systemem operacyjnym i wykorzystują funkcje przez niego oferowane. Mogą monitorować działania jądra systemu, procesów, logi systemowych, klucze rejestru czy też konkretne pliki. W przypadku wykrycia zagrożenia podejmują one natychmiastowe działanie mające na celu powstrzymanie atakującego. Mogą to być na przykład zmiana praw dostępu do plików, uprawnienia użytkowników, wykasowanie plików bądź też dodanie nowych reguł do zapory sieciowej. W przypadku integracji systemu IPS z jądrem systemu operacyjnego, system IPS może nie dopuścić do wykorzystania przez aplikacje funkcji przez niego oferowanych.

Systemy NIPS oferują funkcjonalność podobną do systemu IDS połączonego z zaporą sieciową. Klasyfikowane są one czasami też jako In-line IDS lub Gateway IDS, o których była mowa wcześniej (zobacz: [Rash 2005], str. 6). NIPS posiada przynajmniej dwa interfejsy sieciowe, z których jeden pełni rolę interfejsu zewnętrznego a drugi interfejsu wewnętrznego. Główną różnicą pomiędzy systemami NIDS a NIPS jest fakt, że systemy NIPS umieszczone są na trasie pakietu. W odróżnieniu od systemów NIDS, NIPS potrafi nie dopuścić do przejścia żadnego z pakietów, który stanowi część ataku. System IDS po wykryciu ataku wysłałby informacje o tym do zapory sieciowej a ta dopiero zablokowałaby ruch pochodzący z danego adresu IP. Jednak zanim by do tego doszło przynajmniej jeden pakiet przeszedłby do wnętrza chronionej sieci. Systemy NIPS dzięki temu, że ich integralną część stanowi zaporę sieciową potrafią zablokować takie ataki. Systemy te umożliwiają również zmianę zawartości pakietów. Pozwala to na zablokowanie ataku w sposób niewidoczny dla atakującego i na przykład przekierowanie takiego ataku do systemu honeypot (pułapki) w celu zdobycia większej ilości informacji o atakującym. Można wyróżnić cztery rodzaje przeciwdziałań, jakie mogą podejmować systemy NIPS (zobacz: [Rash 2005], str. 7 i nast.). Zostały one pogrupowane ze względu na warstwę w modelu ISO/OSI, w jakiej funkcjonują:

- przeciwdziałania warstwy łącza danych,

- przeciwdziałania warstwy sieciowej,
- przeciwdziałania warstwy transportowej,
- przeciwdziałania warstwy aplikacji.

W obrębie warstwy łącza danych możliwe jest administracyjne zamknięcie portu w przełączniku, poprzez który atakujący dostał się do sieci. Jest to metoda stosowana w przypadku, gdy atak pochodzi z sieci lokalnej. Istotne jest, aby system mógł odblokować taki port po pewnym czasie, ponieważ stałe odcięcie jednego z portów mogłoby spowodować problemy z poprawnym funkcjonowaniem sieci.

W obrębie warstwy sieciowej, przeciwdziałanie może polegać na współpracy systemu IPS z zewnętrznymi zaporami sieciowymi lub routerami. Umożliwia to całkowite zablokowanie komunikacji z konkretnego adresu IP lub też z całą siecią. System IPS może wykonać to zadanie bez wykorzystywania zewnętrznych aplikacji bądź urządzeń. Istotne jest, aby system mógł po pewnym czasie odblokować wcześniej zablokowane adresy.

W obrębie warstwy transportowej system IPS może wysyłać pakiety TCP z ustawioną flagą RST w celu zresetowania połączenia TCP, lub wysyłać pakiety ICMP z różnymi kodami błędów przypadku połączeń UDP.

W obrębie warstwy aplikacji system IPS może zmieniać dane przesyłane w pakiecie. Operacja taka wymaga ponownego przeliczenia sumy kontrolnej pakietu.

Istnieją też inne rozwiązania bazujące na technologii systemów IPS:

- przełączniki warstwy siódmej (zobacz: [I3])

Przełączniki warstwy siódmej stworzono w celu wydajnego zarządzania pasmem dla różnych aplikacji i usług. Decyzje przełączania i trasowania oparte są o informacje z 3 najwyższych warstw modelu OSI. Urządzenia tego typu używane są, aby umożliwić rozkład obciążenia na poszczególne serwery. Firmy produkujące urządzenia tego typu implementują w nich elementy odpowiadające za bezpieczeństwo np. zabezpieczenie przed atakami DoS i DDoS. Detekcja niebezpiecznych zdarzeń oparta jest o bazę sygnatur, z możliwością aktualizacji. Switche warstwy siódmej nie wprowadzają zaawansowanych metod zapobiegania włamaniom, ale jako jedne z niewielu potrafią udaremnić ataki typu DoS i DDoS. Urządzenia te dysponują dużą mocą obliczeniową i są bardzo wydajne, co pozwala im łagodzić ataki DoS tak, aby nie miały znaczącego wpływu na wydajność i działanie chronionej sieci.

- Firewall aplikacyjny (zobacz: [I3])

Rozwiązania te są zorientowane na ochronę wyłącznie hosta, na którym są zainstalowane. IPSy tego typu nie sprawdzają budowy nagłówków pakietów, ich zawartości, lecz skupiają się na właściwościach uruchomionych aplikacji i zasobach systemowych. Ich zadaniem jest kontrola wywołań API, monitorowanie wykorzystania pamięci oraz prób jej przepełnienia, obserwowanie zależności uruchomionych aplikacji z systemem operacyjnym itd. Takie rozwiązanie zapewnia ochronę przed nieznanymi atakami i błędnie lub nieudolnie stworzonymi aplikacjami.

Firewalles aplikacyjne budują odpowiednie profile zachowań dla odpowiednich aplikacji. W przeciwieństwie do NIDS/NIPS typu in-line, oraz switchy warstwy siódmej omawiane rozwiązanie jest systemem typu „fail close”, co oznacza, że jeśli dane zachowanie nie jest zdefiniowane, IPS zabroni jego wykonania. Istotne jest, zatem aby upewnić się, że blokowane nie zostaną poprawne, nieszkodliwe zdarzenia.

- Switche hybrydowe

Rozwiązanie takie jest połączeniem firewalla aplikacyjnego oraz switcha warstwy siódmej. Switch hybrydowy nie opiera swojego działania o zestaw reguł typowy dla systemu NIDS/NIPS, lecz o profile analogiczne do budowanych w firewallach aplikacyjnych. Systemy te monitorują ruch sieciowy w poszukiwaniu zawartości zdefiniowanej w ogólnej polityce ochrony, ale posiadają również szczegółowe informacje o działających aplikacjach. Switche hybrydowe dysponują dużymi zasobami i wydajnością i zapewniają ochronę sieci przed atakami typu DoS i DDoS.

- Aplikacje oszukujące (ang. *deceptive applications*)

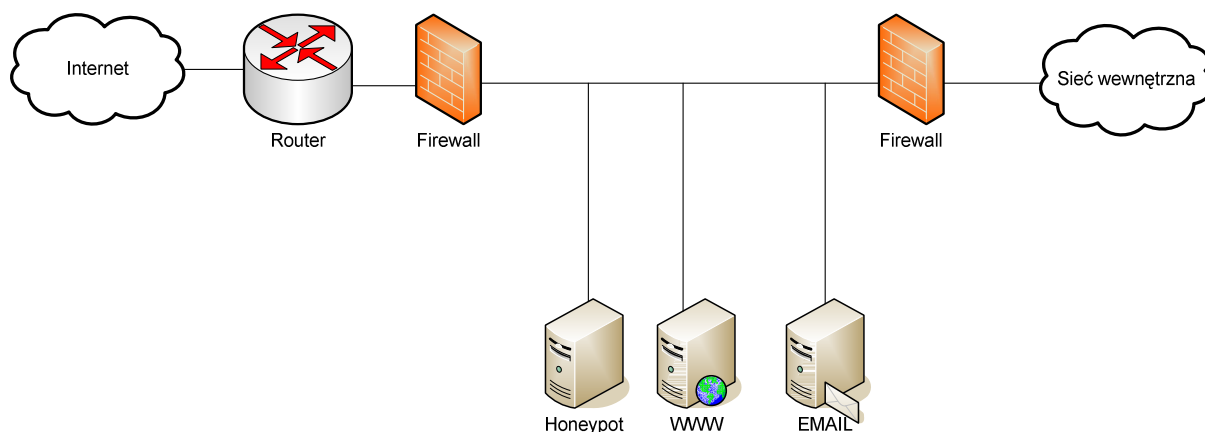
System tego typu monitoruje ruch sieciowy i na podstawie określonych reguł klasyfikuje go jako ruch bezpieczny lub niebezpieczny. W tej fazie jego działanie nie różni się od działania klasycznego systemu IDS. W momencie wykrycia prób łączenia się z nieistniejącymi usługami lub prób łączenia się z istniejącymi usługami w sposób podejrzany aplikacja wysyła do atakującego odpowiednio zbudowane pakiety.

- IPS oparte o klasyczny NIDS i firewall sygnaling

Rozwiązania oparte na działaniu klasycznego systemu NIDS, który po wykryciu ataku zmodyfikuje istniejące reguły na firewallu lub doda nowe. Przykładem takiego systemu może być współpraca programów Snort i Guardian. Guardian jest darmowym programem, który na podstawie alertów wygenerowanych przez Snorta modyfikuje reguły iptables. Blokuje on na określony czas adresy IP, które były źródłami ataku.

### 1.11. Systemy honeypot (Maciej Skowroński)

Systemy honeypot są to programy pozwalające symulować nieistniejące w danej sieci usługi sieciowe lub urządzenia (zobacz: [Szmit 2005], str. 274). Programy tego typu określa się w języku polskim jako przynęty lub pułapki. Stosowane są one w celu zmylenia atakującego, co do wielkości sieci oraz działających w niej serwerów usługowych. Realizowane jest to poprzez symulowanie pracy w sieci różnego typu urządzeń lub usług takich jak na przykład serwer WWW lub router. Udawane mogą być również usługi działające już w sieci. Napastnik po wykryciu w sieci takiego urządzenia bądź usługi będzie chciał przełamać zabezpieczenia. Nie stwarza to niebezpieczeństwa, ponieważ systemy te są tylko wirtualnymi usługami i nie udostępniają żadnych istotnych danych.



**Rysunek 13: Przykładowy schemat umieszczenia systemu Honeypot w sieci. Źródło: opracowanie własne.**

Systemy te instaluje się w dwóch podstawowych celach (według: [I11]):

- rejestrowania działań atakującego jak dowodów przeciwko niemu,
- określenia metody ataku jaką posłużył się napastnik.

Analiza tych informacji pozwala na lepsze zabezpieczenie rzeczywistych systemów działających w sieci.

Oprócz logów zebranych przez system honeypot przydatne mogą być również logi pochodzące z firewalla oraz informacje o ruchu przychodzącym do systemu honeypot zebrane przez sniffer. Honeypot może być umieszczony przed strefą DMZ, wewnątrz niej lub w sieci wewnętrznej. Systemy tego typu mogą być wykorzystane przez systemy IDS. W przypadku wykrycia atakującego system IDS może zmodyfikować ustawienia zapór sieciowych oraz routerów i przekierować ruch pochodzący od agresora do systemu honeypot. Systemy honeypot mają wiele zalet (według: [I12]):

- Zbierają niewielkie ilości informacji. Jest to spowodowane tym, że logują one tylko informacje w momencie, gdy ktoś komunikuje się z nimi.
- Redukują liczbę fałszywych alarmów, ponieważ z założenia każda komunikacja z systemem honeypot jest próba ataku.
- Potrafią wykrywać nowe rodzaje ataków. Jest to spowodowane tym, że – w przeciwieństwie do systemów typu IDS czy IPS – systemy honeypot zbierają dane dotyczące przebiegu ataku. Na ich podstawie można wykryć nowe typy ataków.
- Szyfrowanie transmisji nie wpływa na skuteczność ich działania, ponieważ system honeypot jest traktowany przez atakującego jako punkt końcowy komunikacji.
- Są to systemy bardzo dynamiczne i mogące być łatwo dostosowane do konkretnych wymagań.
- Mają małe wymagania sprzętowe.

Systemy te mają również pewne wady:

- Widzą tylko ataki przeprowadzane na ich samych. Nie potrafią wykryć ataków przeprowadzanych przeciwko innym systemom.
- Istnieje ryzyko, że system zostanie złamany przez atakującego i wykorzystany do prowadzenia dalszego ataku.

Systemy honeypot można podzielić pod względem interaktywności na dwa rodzaje:

- Nisko interaktywne,
- Wysoko interaktywne.

Nisko interaktywne systemy emulują przede wszystkim systemy i usługi. Atakujący na systemach tego typu może wykonać bardzo ograniczony zakres czynności. Zaletą tego typu honeypotów jest to, że są zazwyczaj od razu odpowiednio skonfigurowane i ich instalacja jest prosta. Systemy te minimalizują również ryzyko złamania ich zabezpieczeń przez atakującego, ponieważ oferują małą funkcjonalność. Mogą mieć za to problemy z odpowiednim zachowaniem w przypadku, gdy atakujący zastosuje nieznaną metodę ataku. Jest to spowodowane brakiem zbioru odpowiednich zachowań dla nowego ataku. Przykładem systemu tego typu jest Honeyd (zobacz: [116]). Jest to program w pełni darmowy opracowany dla systemów z rodziny UNIX.

Systemy wysoko interaktywne w przeciwieństwie do systemów nisko interaktywnych nie emulują systemów i usług. Tworzą one prawdziwe aplikacje i usługi, z którymi komunikować może się atakujący. Honeypoty tego typu mogą być skomplikowane w instalacji i konfiguracji. Zwiększone jest też ryzyko przejęcia systemu przez atakującego. Jest to

spowodowane tym, iż ma on do czynienia z prawdziwym systemem, a nie jego emulacją. Systemy tego typu zapewniają o wiele więcej informacji niż systemy nisko interaktywne. Informacje te są również dokładniejsze. Przykładem takiego systemu jest Symantec Decoy Server (zobacz: [I17]).

### **1.12. Ograniczenia systemów IDS (według: [Pieprzyk 2005], str. 438 i nast.) (Maciej Skowroński)**

Pomimo, że systemy IDS rozwijane są już od kilkunastu lat nadal istnieje wiele problemów do rozwiązania. Systemy nie spełniają w stu procentach oczekiwań ich użytkowników. Koszty stworzenia nowego systemu są wysokie, ponieważ brak jest gotowych wzorców tworzenia tego typu aplikacji. Tworzenie skutecznych systemów detekcji intruzów wymaga rozległej wiedzy z dziedzin takich jak sztuczna inteligencja, systemy operacyjne i sieci komputerowe. IDSy te tworzone są dla konkretnych platform systemowych. Efektem tego jest to, że niektóre funkcje dostępne są tylko dla danego systemu operacyjnego. Systemy IDS są mało efektywne, ponieważ ich twórcy starają się, aby były one w stanie wykryć jak najwięcej zagrożeń. Powoduje to wzrost ilości obliczeń do wykonania przy każdym zdarzeniu. Problemem jest również utrzymanie i konserwacja systemu IDS, ponieważ wymaga od osoby zarządzającej dużego doświadczenia i wiedzy nie tylko z zakresu bezpieczeństwa sieci komputerowych. Bardzo trudne jest ocenienie efektywności działania danego systemu w porównaniu z innymi systemami. Powodem tego jest trudność z uzyskaniem porównywalnych warunków testowych oraz powtarzalnego zachowania atakujących. Z tych powodów ciężko jest znaleźć informacje na temat efektywności systemów IDS a te, które można znaleźć są mało precyzyjne.

### **1.13. Snort – darmowy system detekcji intruzów (Maciej Skowroński i Radosław Wężyk)**

Snort jest systemem detekcji intruzów dystrybuowany w oparciu o licencję GPL (zobacz: [I10]). Został on napisany w 1998 roku przez Martina Roescha. Z założenia miał on być snifferem przeznaczonym dla sieci domowych (zobacz: [I5]). Udostępnienie Snorta na zasadach open source (zobacz: [I6]) nastąpiło w grudniu 1998 roku. W lipcu roku 1999 pojawiła się wersja 1.0. W grudniu 1999 roku ukazała się wersja 1.5. Wersja ta przyniosła

zmianę architektury programu na modułową i w takiej formie została zachowana ona do dziś. W połowie roku 2001 pojawiła się wersja Snort 1.8, która w porównaniu do wersji poprzednich zawierała między innymi szybki system wyjściowy oraz rozbudowane preprocesory odpowiedzialne za składanie pofragmentowanych pakietów IP i strumieni TCP. Od wersji 2.3 Snort oferuje już ponad 3000 reguł detekcji oraz wykrywanie niezgodności z protokołami. Prace rozwojowe nad program trwają intensywnie cały czas. Kod źródłowy jest dostępny do pobrania na stronie domowej projektu (zobacz: [118]).

### 1.13.1. Tryby pracy Snorta

Snort udostępnia kilka różnych trybów pracy. Są to (zobacz:[17]):

- sniffer,
- logger pakietów,
- NIDS,
- Inline.

W trybie sniffera (zobacz: [Rehman 2003], str. 58 i nast.) Snort przechwytyuje cały ruch sieciowy i wyświetla go na monitorze – działa jak typowy sniffer np. tcpdump. W zależności od potrzeb, przy pomocy odpowiednich opcji ustawianych przy włączeniu aplikacji możemy określić jak szczegółowe informacje chcemy poznać. Przechwytywanie ruchu oparte jest o bibliotekę libpcap. Snort uruchomiony w trybie sniffera bez dodatkowych opcji wyświetla podstawowe informacje o przechwyconym pakiecie, jak pokazano na wydruku poniżej

[illegible]

### Wydruk 1: Przechwycony pakiet (root@serwer:/# snort -v).

Powyższy wydruk zawiera następujące informacje:

- Data i czas przechwycenia pakietu,
- Źródłowy adres IP,
- Źródłowy port,



- Docelowy adres IP,
- Docelowy port,
- Użyty protokół warstwy transportowej lub sieciowej (ICMP),
- TTL (ang. *Time to Live*),
- ToS (ang. *Type of Service*),
- ID datagramu IP,
- Długość nagłówka IP w bajtach,
- Długość datagramu IP (bez nagłówka),
- Flaga datagramu IP dotycząca fragmentowania,
- Flagi pakietu TCP,
- Numer SEQ pakietu TCP,
- Numer ACK pakietu TCP,
- Wielkość okna pakietu TCP,
- Długość nagłówka TCP.

Możliwe jest uzyskanie bardziej szczegółowych informacji o przechwyconych pakietach przy użyciu dodatkowej flagi `-d`, co pokazano na wydruku 2.

```
04/11-17:01:21.258948 12.202.67.139:6346 -> 83.16.166.250:1033
TCP TTL:110 TOS:0x0 ID:51037 IpLen:20 DgmLen:172 DF
***AP*** Seq: 0x3FD66EA7 Ack: 0x27C83D1D Win: 0xFAE1 TcpLen: 20
2A 6F F2 57 74 1E D4 9F 3B DF 75 79 A8 DA 73 44 *o.Wt...;.uy..sD
8F FB 32 A4 F3 2C 1A 7E E7 79 B1 5D 27 96 8C 0D ..2...,.~.y.]'...
5C D7 5F 82 37 6D 9C C3 EF 35 21 FB 79 F2 2A 31 \._.7m...5!.y.*1
8A 85 6C 09 E0 14 9C 4B 11 3E 59 15 10 51 7C 9C ..1....K.>Y..Q|.
A9 03 B4 06 57 32 8A 72 95 AD 59 79 DD 84 8A 0E ....W2.r..Yy....
F9 D4 D2 1F E9 D3 45 BD BB 6B D7 A9 F7 C5 3D 24 .....E..k....=$
C6 47 9D C6 57 A2 FE 20 1E 33 75 D8 56 66 85 66 .G..W...3u.Vf.f
B1 23 C9 CA 58 D9 11 12 68 D2 C5 92 FE CC 90 22 .#..X...h....."
1A 2D D7 C2 .-..
```

**Wydruk 2: Przechwycony pakiet (root@serwer:/# snort -dv).**

Istnieje możliwość podejrzenia dodatkowo zawartości nagłówków ramek warstwy drugiej modelu OSI (zob. wydruk 3):

### Wydruk 3: Przechwycony pakiet (root@serwer:/# snort -dev).

Tryb pracy NIDS różni się przede wszystkim od wyżej wymienionych tym, iż nie loguje on wszystkich pakietów. Używa on do każdego przechwyconego pakietu zbioru reguł, które zostały wybrane w pliku konfiguracyjnym Snorta. Następnie w przypadku dopasowania

reguły, podejmując zapisaną w regule akcję. W trybie tym zapisywane są tylko te pakiety, które zostały dopasowane do sygnatur. Istnieje kilka możliwości logowania. Zostaną one omówione poniżej. Reguły są to pliki tekstowe zawierające opisy zagrożeń wraz z typem reakcji, jaka ma zostać podjęta po ich wykryciu. Do tworzenia reguł wykorzystywany jest prosty język opisowy, który jest jednak elastyczny i efektywny. Reguły można podzielić na dwie logiczne części:

Nagłówek reguły	Definicja reguły
-----------------	------------------

**Rysunek 14: Ogólna budowa reguł (według: [Rehman 2003] str. 79)**

W obrębie nagłówka znajdują się elementy takie jak:

- typ akcji jaka ma zostać podjęta po dopasowaniu reguły,
- protokół,
- adres źródłowy i docelowy,
- maski,
- port źródłowy i docelowy.

Akcja	Protokół	Adres Ź.	Port Ź.	Kierunek	Adres D.	Port D.
-------	----------	----------	---------	----------	----------	---------

**Rysunek 15: Składnia nagłówka reguły (według: [Rehman 2003] str. 79)**

W obrębie drugiej części reguły znajdują się treść alertu oraz informacje dotyczące działań, jakie mają być podjęte w obrębie danego pakietu.

Pierwszy elementem reguły jest akcja, jaka ma zostać podjęta w przypadku dopasowania pakietu. Istnieje pięć standardowych możliwości (zobacz: [Szmit 2005] str. 505):

- alert – wygenerowanie alertu ( alert może zostać wysłany m.in. do pliku lub na konsolę); pakiet jest logowany,
- log – zalogowanie pakietu – istnieją różne sposoby zapisu pakietu: m.in. zapis do pliku, bazy danych),
- pass – zignorowanie i przepuszczenie pakietu,
- activate – wygenerowanie alertu i uruchomienie innej reguły,

- dynamic – wywoływana wyłącznie przez regułę typu active.

Podczas pracy w trybie inline dostępne są dodatkowo trzy możliwości:

- drop,
- reject,
- sdrop.

Reguła drop przekaże do iptables polecenie zablokowania pakietu a sam Snort zaloguje informacje o nim. Reject zadziała tak samo jak drop, ale dodatkowo wyśle odpowiedź TCP reset bądź ICMP o nieosiągalności, w zależności od wykorzystanego protokołu. Reguła typu sdrop spowoduje zablokowanie przez iptables pakietu. Snort nic w tym przypadku nie zaloguje.

Począwszy od wersji 2.3.0 RC1 w Snorcie pojawiła się możliwość pracy w trybie inline. Oferuje on funkcjonalność systemu IPS. W trybie tym Snort pobiera pakiety z iptables zamiast z libpcap a następnie na podstawie reguł wpływa na samą konfigurację iptables. W trybie tym Snort oferuje możliwość wpływania na zawartość pakietów wychodzących z sieci.

Na przykład reguła w postaci:

```
alert tcp any any <> any 80 (msg: „tcp replace”; content:
“GET”; replace „BET”;
```

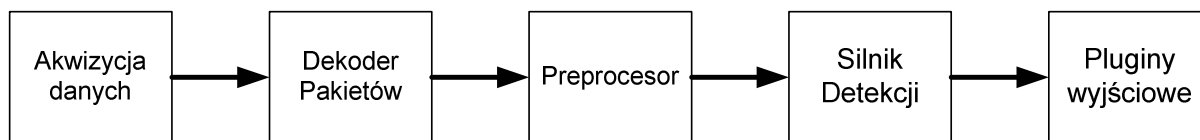
spowoduje monitorowanie ruchu na porcie 80 w poszukiwaniu słowa GET. Gdy zostanie ono odnalezione wewnątrz pakietu zostanie ono zastąpione słowem BET. Można w ten sposób zastępować dowolne ciągi znaków innymi. Jedynym warunkiem jest taka sama długość obu ciągów.

### 1.13.2. Ogólny schemat działania Snorta

Snort jest podzielony na kilka logicznych elementów, które współpracują ze sobą (zobacz: [Rehman 2003] str. 12 i nast.). Można wyróżnić kilka podstawowych jego części:

- dekodery pakietów,
- preprocesory,
- silnik detekcji,
- system logowania i alertów,
- moduły wyjściowe.

Przepływ danych pomiędzy poszczególnymi elementami systemu obrazuje rysunek 13.



Rysunek 16: Przepływ danych w programie Snort. Źródło: [15].

Dekoder pakietów za pośrednictwem biblioteki libpcap otrzymuje pakiety pochodzące z różnego typu interfejsów i przygotowuje je do przetworzenia przez kolejne elementy programu. Kolejnym krokiem jest przesłanie przechwyconych danych przez serie preprocesorów.

Preprocesory są to elementy Snorta, które mają za zadanie przygotować przechwycone dane do przetworzenia przez silnik detekcji lub też same poszukują śladów włamania. Zostały one dodane do Snorta począwszy od wersji 1.5. Umożliwiają one łatwe rozszerzenie funkcjonalności programu poprzez dopisywanie własnych modułów. Plik konfiguracyjny Snorta pozwala wybrać, które preprocesory mają zostać włączone a które nie. Podstawowymi preprocesorami, które wchodzi standardowo w skład Snorta 2.4.0 są (zobacz: [17]):

- Frag3,
- Stream4,
- sfPortscan,
- Telnet decode,
- RPC decode,
- Performance monitor,
- HTTP inspect,
- ASN.1 detection,
- X-Link2State Mini-Preprocessor.

Preprocesor Frag3 defragmentuje i normalizuje dane przychodzące w postaci fragmentów. Utrudnia to prowadzenie ataków, które wykorzystują drobno sfragmentowane pakiety. Stream4 jest to rozbudowany preprocesor analizującego strumienie TCP. Wykrywa niektóre formy skanowania portów, poprawność stanów połączeń oraz sum kontrolnych pakietów (TCP, UDP i ICMP). sfPortscan wykrywa próby skanowania portów. Jest to bardzo istotny element, ponieważ skanowanie portów jest zazwyczaj pierwsza faza ataku w przypadku, gdy atakujący nie zna atakowanego systemu. Preprocesor Telnet decode normalizuje dane przechwycone z ruchu odbywającego się za pośrednictwem protokołu telnet. RPC decode to preprocesor, który składa pofragmentowane rekordy RPC w jeden nie

pofragmentowany. Gdy aktywny jest Stream4, będzie on przetwarzać tylko ruch pochodzący od strony użytkownika. Performance monitor mierzy wydajność Snorta w pracy w czasie rzeczywistym oraz jego teoretyczną maksymalną wydajność. HTTP Inspect jest ogólnym dekoderym protokołu HTTP dla aplikacji użytkownika. Potrafi on zdekodować otrzymane dane, wyszukać w nich pól HTTP oraz je znormalizować. ASN.1 detection dekoduje pakiety lub porcje pakietów w poszukiwaniu złośliwego kodowania. X-Link2State Mini-Preprocessor ma za zadanie wykrywać próby wykorzystania luki X-Link2state w Microsoft Exchange Server (zobacz: [I7]). W ramach naszej pracy zaimplementowany został preprocesor, który loguje przepływający ruch oraz porównuje go ze stworzonym dla sieci profilem.

Kolejnym elementem w przepływie danych jest silnik detekcji (zobacz: [Rehman 2003], str. 14 i nast.). Jest to główny element programu odpowiedzialny za wykrywanie włamań. Do wyszukiwania śladów ataku, wykorzystywane są reguły. Reguły są pisane przy wykorzystaniu prostego języka, który to jednak daje możliwość tworzenia efektywnych reguł. W przypadku dopasowania którejś z reguł, podejmowana jest wybrana w niej akcja. To właśnie silnik detekcji jest elementem, którego praca trwa najdłużej. Duże znaczenie ma dla niego maszyna, na której pracuje oraz ilość reguł, które musi wziąć pod uwagę. Ważnym elementem jest również ruch w sieci. Jeśli jest on zbyt duży to silnik detekcji może nie dać rady sprawdzać go w czasie rzeczywistym lub wyrzucić pakiety. W zależności od wersji Snorta silnik detekcji działa inaczej. W wersjach z serii 1.x silnik detekcji zaprzestawał dalszego przetwarzania gdy tylko udało mu się dopasować regułę. Podejmował akcję taką jak była zapisana w tej regule. Nie analizował on sytuacji, gdy jeden pakiet spełnia kilka reguł. Był to niewątpliwie problem, ponieważ jeśli pakiet, który powinien zostać wykryty przez regułę wysokiego priorytetu, trafiał najpierw na regułę o niskim priorytecie, to zagrożenie, jakie z niego wynikało było klasyfikowane jako niskie. Powodowało to przedstawienie nie prawdziwej sytuacji. Problem ten został rozwiązany począwszy od wersji 2.0, w której został umieszczony napisany od nowa silnik detekcji. Jest on kilkakrotnie szybszy od poprzedniego i próbuje dopasować pakiet do wszystkich reguł zanim wygeneruje alert.

Następnym elementem jest system logowania i alertów. W zależności od tego, co wykryje silnik detekcji, pakiet może zostać zapisany lub też może zostać zgłoszony na jego podstawie alert. Sposób, w jaki zostanie to zrobione uzależnione jest od kolejnego elementu w przepływie danych, czyli od modułów wyjściowych. Zostały one wprowadzone do programu Snort począwszy od wersji 1.6 w celu zwiększenia przejrzystości wyników pracy programu dla jego użytkowników. W pliku konfiguracyjnym można wybrać kilka modułów wyjściowych. Domyślnie, plikiem wyjściowym Snorta jest plik tekstowy umiejscowiony w

/var/log/snort. Za pośrednictwem modułów wyjściowych można wybrać, jakiego typu dane pojawią się na wyjściu systemu logowania i alertów. W zależności od tego, jaki moduł zostanie wybrany, możliwe jest uzyskanie danych wyjściowych w różnych postaciach. W Snorcie 2.4.0 można wyróżnić następujące moduły wyjściowe (zobacz: [17]):

- alert\_syslog,
- alert\_fast,
- alert\_full,
- alert\_unixsock,
- log\_tcpdump,
- database,
- csv,
- unified,
- log\_null.

Moduł alert\_syslog przesyła alerty do demona syslog. Syslog jest systemem logującym i generującym logi dla zdarzeń systemowych.

Alert\_fast zapisuje alerty w szybkim formacie czasu rzeczywistego do konkretnego pliku tekstowego. Jest to szybka metoda zapisywania, ponieważ nie loguje ona całych nagłówków pakietów do pliku wyjściowego. Zapisywane są kolejno:

- czas,
- treść alertu,
- źródłowy i docelowy adres IP,
- źródłowy i docelowy port.

Aby uruchomić program Snort w trybie fast alert należy użyć przełącznika -A fast w linii poleceń.

W przeciwieństwie do alert\_fast, moduł alert\_full zapisuje alerty wraz z pełnym nagłówkiem pakietu. Jest to domyślny tryb zapisu. Miejsce zapisu może domyślnie być domyślny katalog dla logów programu, czyli /var/log/snort lub też inny katalog wybrany przez użytkownika. Wewnątrz katalogu zostaną stworzone katalogi, a wewnątrz których zapisywane będą zdekodowane pakiety, które wywołały alert. Nazwami katalogów będą adresy IP. Ten typ logowania znacznie spowalnia działanie Snorta i nie jest zalecany dla sieci z dużym ruchem.

Moduł `alert_unixsock` tworzy UNIX-domain socket (zobacz: [I9]), na którym mogą nasłuchiwać zewnętrzne programy bądź procesy w celu otrzymania danych od Snorta w czasie rzeczywistym.

Kolejny moduł, `log_tcpdump`, zapisuje dane wyjściowe w formacie plików programu `tcpdump` [I8]. Umożliwia to późniejszą analizę zgromadzonych danych za pomocą wielu dostępnych narzędzi do analizy logów w tym formacie.

Moduł `database` umożliwia zapis danych wyjściowych z programu do bazy SQL. Taki sposób przechowywania danych umożliwia łatwiejsze i szybsze tworzenie statystyk oraz prezentowania historii alertów niż w przypadku plików tekstowych. Moduł ten umożliwia aktualnie współpracę z następującymi bazami danych (zobacz: [I7]):

- `mssql`,
- `mysql`,
- `postgresql`,
- `oracle`,
- `odbc`.

Możliwa jest też obsługa bazy danych `prelude`, jednak nie jest ona wbudowana domyślnie. Aby została ona włączona należy użyć przełącznika `--enable-prelude` podczas wywoływania skryptu konfiguracyjnego.

Moduł `csv` umożliwia zapisywanie danych w formacie umożliwiającym łatwe zaimportowanie do bazy danych czy też na przykład do arkusza kalkulacyjnego.

Moduł `unified` został zaprojektowany jako najszybsza metoda logowania. Dane wyjściowe są zapisywane w postaci binarnej. Dzięki temu Snort pozostawia programom zewnętrznym przetworzenia tych danych jednocześnie odciążając się. Moduł ten tworzy dwa pliki wyjściowe: `alert` oraz `log`. Plik `alert` zawiera informacje temat samego wydarzenia, a plik `log` dokładne dane dotyczące pakietu, który je wywołał. Oba pliki zapisane są w formacie binarnym.

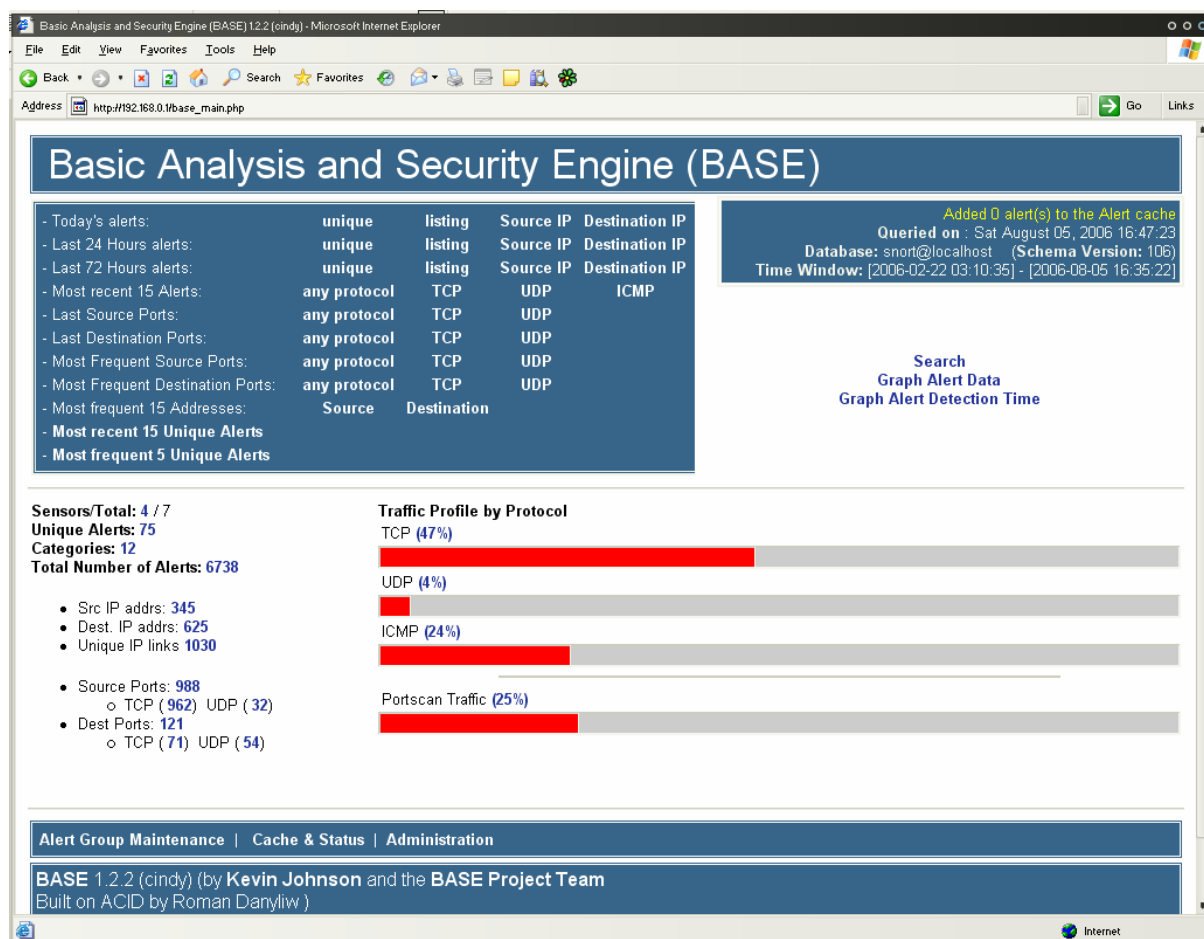
Poczynając od wersji 1.8.2 pojawił się moduł `log_null`. Wykorzystywany jest on w przypadkach, gdy chcemy stworzyć regułę, która wyświetli alert, jednak nie będzie logować informacji o nim.

### 1.13.3. Basic Analysis and Security Engine

BASE (ang. *Basic Analysis and Security Engine*) (zobacz: [I13]) jest to darmowy interfejs umożliwiający wygodne przeglądanie alertów pochodzących z systemu Snort. Zrealizowany



jest on w języku PHP, w formie strony WWW. Aby można było z niego korzystać, Snort musi zostać skonfigurowany tak, aby logował alerty do bazy danych obsługiwanej przez BASE (np. MySQL).



Rysunek 17: Okno główne interfejsu BASE. Źródło: opracowanie własne.

BASE został opracowany na bazie projektu ACID (zobacz: [I14]). Interfejs ten umożliwia tworzenie wielu rodzajów wykresów na podstawie alertów odczytanych z bazy danych Snorta. Przydatne są funkcje, które umożliwiają wyszukiwanie, oraz funkcje, które grupują wyświetlane alerty według różnych kategorii (alerty z dzisiejszego dnia, z ostatnich 24 lub 72 godzin, 15 najczęściej pojawiających się alertów, 15 ostatnich portów źródłowych lub docelowych, najczęściej pojawiające się porty źródłowe lub docelowe). Możliwe jest również grupowanie według protokołu (TCP, ICMP, UDP) bądź też według adresu docelowego lub też źródłowego. BASE pozwala również na wygodną analizę wykrytych alertów poprzez podgląd zawartości pakietów. Podgląd taki prezentuje nagłówki pakietu jak i przenoszone przez niego informacje. Przykład takiego podglądu przedstawia rysunek 11. BASE umożliwia również szybkie znalezienie informacji na temat wykrytego ataku oraz

adresu IP, z którego on pochodził. Przy każdym wykrytym alercie umieszczone są linki do stron opisujących dany atak oraz do stron umożliwiających identyfikację danego adresu IP.

The screenshot shows the Basic Analysis and Security Engine (BASE) interface in a Microsoft Internet Explorer browser. The address bar shows the URL: `http://192.168.0.1/base_qry_alert.php?submit=5230-5281-7523%sort_order=`.

The main heading is "Basic Analysis and Security Engine (BASE)". Below it, there is a "Home | Search" link and a "[ Back ]" link.

The "Queried on" timestamp is "Sat August 05, 2006 16:50:49".

The "Meta Criteria" section shows the signature: "[arachNIDS] [local] [snort] ICMP Large ICMP Packet" with a "...Clear..." link. Below this, the "IP Criteria" is "any", "Layer 4 Criteria" is "none", and "Payload Criteria" is "any". A red message states: "Added 0 alert(s) to the Alert cache".

The "Alert #0" section has a "[ First ]" link and a ">> Next #1-(1-8)" button.

The packet analysis section is divided into several tabs: "Meta", "IP", and "ICMP".

- Meta Tab:**
  - ID #:** 1 - 7
  - Time:** 2006-02-22 10:35:38
  - Triggered Signature:** [arachNIDS] [local] [snort] ICMP Large ICMP Packet
  - Sensor:** 192.168.0.1
  - Interface:** eth0
  - Filter:** none
  - Alert Group:** none
- IP Tab:**
  - Source Address:** 192.168.0.7
  - Dest. Address:** 213.25.5.57
  - Ver:** 4
  - Hdr Len:** 5
  - TOS:** 0
  - length:** 1478
  - ID:** 271
  - flags:** 0
  - offset:** 0
  - TTL:** 3
  - chksum:** 5671
  - Options:** none
- ICMP Tab:**
  - type:** (8) Echo Request
  - code:** (0) 0
  - checksum:** 59135
  - ID:** 512
  - seq #:** 3840

The "length" field in the IP tab is expanded to show the packet data in hexadecimal and ASCII format. The length is 1450 bytes. The data is displayed in a grid format with columns for hexadecimal values and their corresponding ASCII characters.

Rysunek 18: Podgląd pakietu w interfejsie BASE. Źródło: opracowanie własne.

## Rozdział drugi. Implementacja.

### 2.1. Opis problemu (Maciej Skowroński)

W ramach części praktycznej niniejszej pracy zaimplementowano system detekcji anomalii w ruchu sieciowym. W tym celu napisane zostały dwa programy:

- preprocesora AnomalyDetection do systemu Snort,
- programu ProfileGenerator generującego profil sieci.

Preprocesor ma dwie podstawowe funkcje:

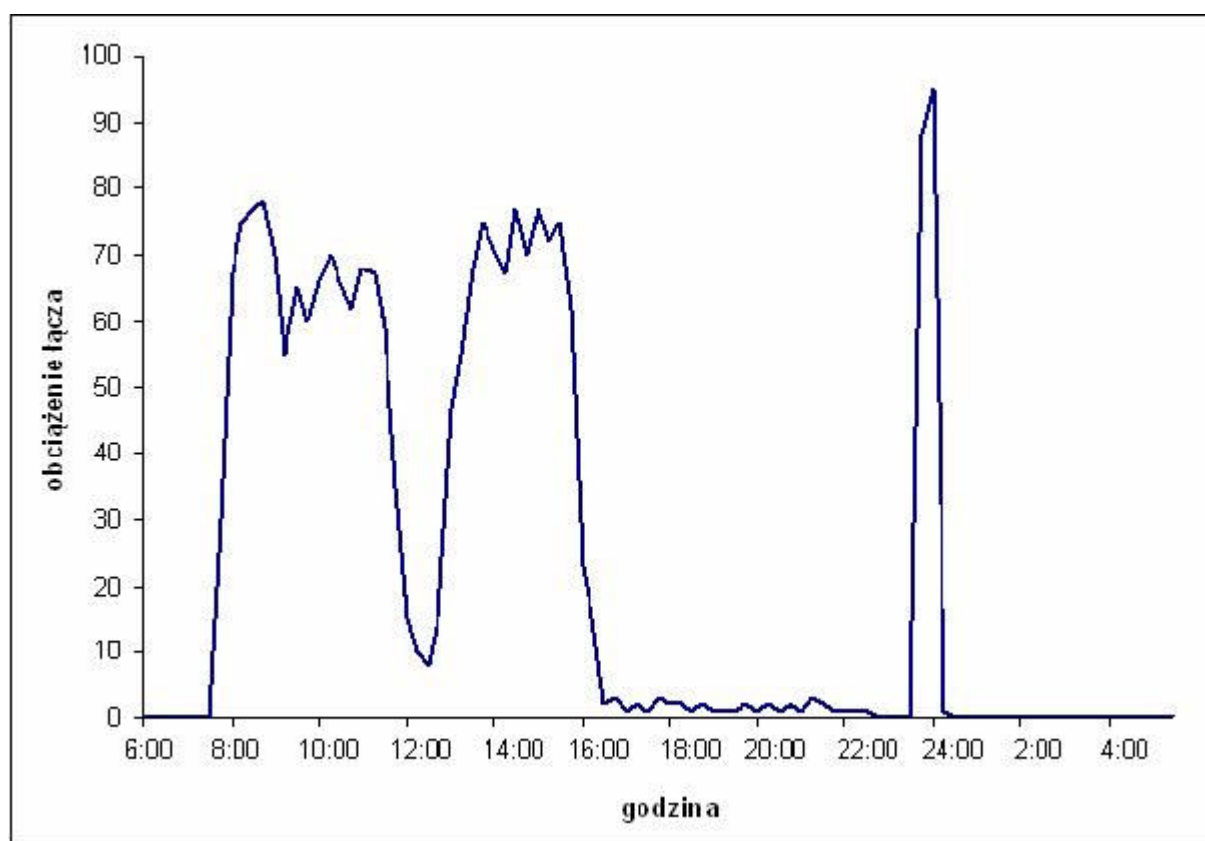
- zapisywanie do pliku przechwyconego ruchu w sieci,
- porównywanie aktualnego ruchu z danymi pochodzącymi z profilu.

W ramach zaimplementowanego rozwiązania, wykrywanie anomalii następuje w:

- Ruchu TCP,
- Ruchu ICMP,
- Ruchu UDP,
- Ruchu WWW,
- Ruchu DNS,
- Liczbie otwartych połączeń,
- Prędkości wysyłania danych,
- Prędkości odbierania danych.

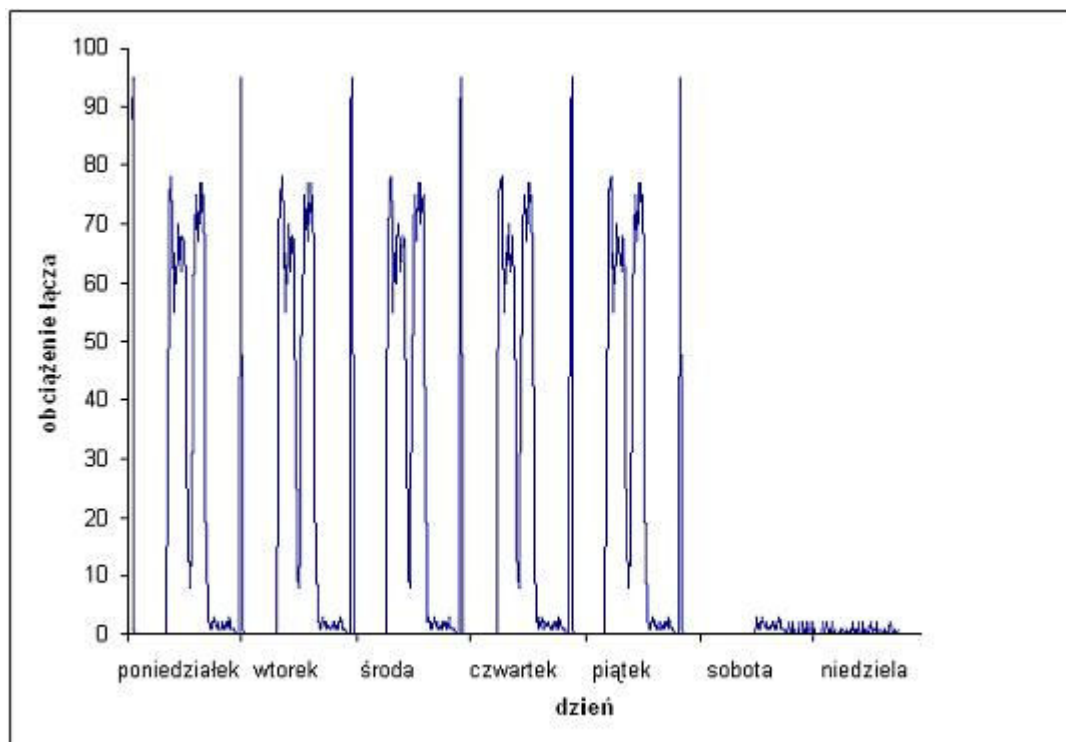
W ramach powyższych punktów sprawdzane są takie rzeczy jak prędkość wysyłania i odbierania w przypadku ruchu, czy też liczba nawiązywanych w danej jednostce czasu połączeń.

Program ProfileGenerator generujący profil sieci wykorzystuje do tego celu logi pochodzące z preprocesora. Efektem jego pracy jest plik tekstowy będący profilem sieci. Profil taki określa charakterystyczną dla danej sieci zależność między dniem i godziną a natężeniem ruchu danego typu w sieci. Profil taki bazuje na fakcie, że w sieci składających się z co najmniej kilkunastu komputerów można zaobserwować pewną powtarzalność w ruchu. Dobrym przykładem takiej sytuacji można zobrazować na przykładzie ruchu w sieci firmowej. Sytuację taką przedstawia rysunek 15.



**Rysunek 19:** Przykładowy wykres obciążenia łącza w ciągu dnia. Źródło: opracowanie własne.

W godzinach nocnych ruch w sieci jest praktycznie równy zeru. Spowodowane jest to tym, że nie ma w firmie pracowników, którzy wykorzystują sieć. W momencie rozpoczęcia pracy o godzinie siódmej rano, ruch bardzo gwałtownie wzrasta. Obciążenie łącza sięga 80%. Jest to spowodowane tym, że pracownicy po przejściu do pracy sprawdzają pocztę lub też czytają najnowsze wiadomości. Po około godzinie natężenie ruchu spada o około dziesięć procent i utrzymuje się na tym poziomie do przerwy obiadowej, która ma miejsce w okolicy godziny dwunastej. Po jej zakończeniu około godziny 13 pracownicy wracają do pracy i ponownie można zaobserwować wzrost ruchu. Takie obciążenie utrzymuje się do godziny szesnastej, czyli do końca pracy w firmie. Po tej godzinie ruch spada do bardzo niskiej wartości. Obciążenie sieci jest generowane przez pracowników, którzy musieli zostać w pracy po godzinach. Około godziny 22 ruch praktycznie zamiera. O północy wysyłane są dzienne raporty z pracy firmy. Owocuje to gwałtownym wzrostem obciążenia sieci. Taka sytuacja powtarza się każdego dnia roboczego. Pewne zależności można jednak zaobserwować również podczas analizy wykresów ruchu tygodniowego. Sytuację taką przedstawia rysunek 16.



Rysunek 16: Przykładowy wykres obciążenia łącza w ciągu tygodnia. Źródło: opracowanie własne.

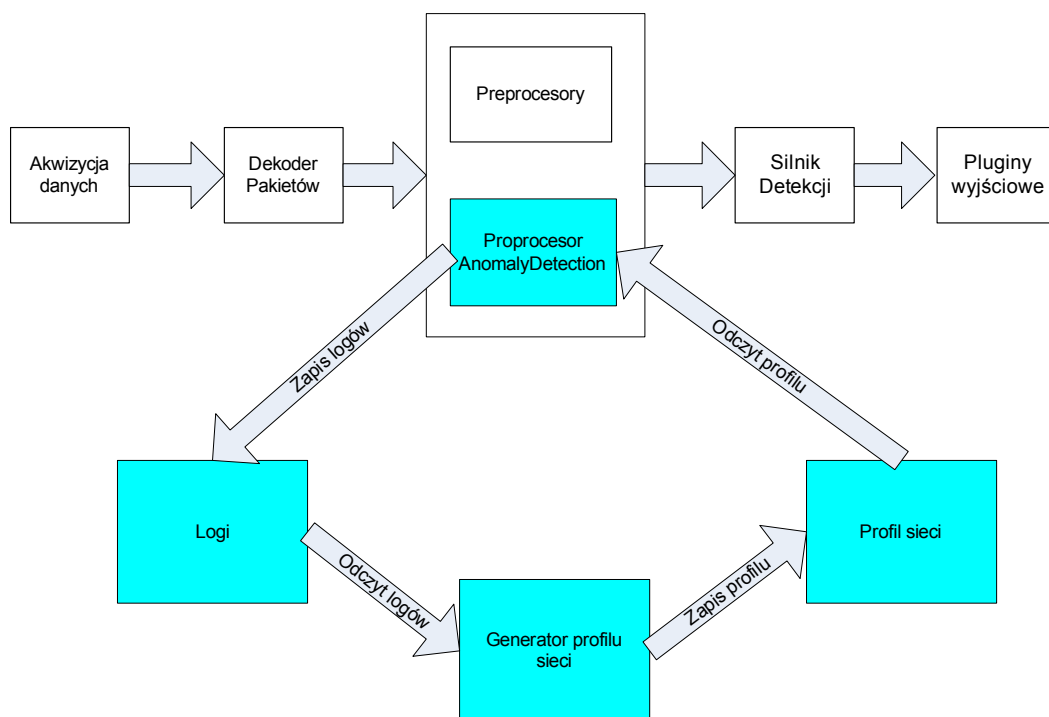
W przypadku dni roboczych podobna sytuacja powtarza się codziennie. Jednak w sobotę i niedzielę, gdy firma nie pracuje, ruch jest minimalny. Generowany jest on przez pracowników, którzy muszą dokończyć prace z tygodnia bądź też na przykład przez portiera przeglądającego strony WWW. Ponieważ firma nie pracuje, nie są wysyłane raporty. Sytuację spadku obciążenia można zaobserwować również podczas wszelkiego rodzaju dni wolnych od pracy. Można, zatem spodziewać się, że szereg czasowy opisujący natężenie ruchu będzie charakteryzował się co najmniej podwójną okresowością – dobową i tygodniową – a być może dodatkowo także roczną (można spodziewać się mniejszego ruchu w święta stałe np. w Boże Narodzenie czy w pierwszy dzień roku).

Stworzenie profilu takiej sieci pozwala na monitorowanie jej pod względem tego, czy jest ona wykorzystywana w sposób normalny. Jeśli nastąpi gwałtowny wzrost ruchu może to oznaczać że ktoś wykorzystuje sieć do przeprowadzania na przykład ataku typu DDoS lub, że nastąpiła awaria. Profil sieci powinien być budowany przez jak najdłuższy czas. Dzięki temu będzie on odpowiadał najlepiej sytuacji rzeczywistej w sieci. Pozwoli to na jego efektywniejsze wykorzystanie w przyszłości. Jest to spowodowane tym, że im dokładniejszy profil tym mniej fałszywych alarmów generowanych na jego podstawie. Należy również zwrócić uwagę na fakt, iż profil sieci może się zmieniać. Jeśli do pracy przyjdą nowi pracownicy lub też ilość pracowników się zmniejszy, bardzo prawdopodobne, że wpłynie to

na obciążenie sieci. Zmianie ulec mogą również przyzwyczajenia pracowników. Z tego powodu profil sieci powinien być często uaktualniany, aby pasował do istniejącej sytuacji. Należy jednak porównywać aktualny profil z profilem poprzednim i analizować ewentualne przyczyny zmiany parametrów sieci. Jest to spowodowane tym, że zmiany w sieci mogą być również celowym działaniem atakującego. Może on regularnie zwiększać określony typ ruchu i w ten sposób wpłynąć na wygenerowanie fałszywego profilu. Efektem tego przeprowadzony przez niego atak, na przykład ściągnięcie dużej ilości chronionych danych, pozostałby niezauważony przez system detekcji anomalii.

## **2.2. Implementacja (Radosław Wężyk)**

Schemat działania systemu przedstawiony jest na rysunku 17. Napisany preprocesor AnomalyDetection zapisuje do pliku logi, w których zawarte są informacje na temat ruchu w sieci. Logi te są następnie przetwarzane przez program, który ma za zadanie na ich podstawie utworzyć profil sieci. Profil ten jest następnie wczytywany przez preprocesor. Aby powstał profil, preprocesor powinien pracować jak najdłużej (najlepiej kilkadziesiąt tygodni) w trybie logowania. Umożliwi to generatorowi profilu zbudowanie profilu sieci, który będzie najlepiej charakteryzował daną sieć. Bloki zaznaczone kolorem niebieskim są częścią systemu napisanego w ramach tej pracy magisterskiej.



Rysunek 17: Schemat ideowy systemu. Źródło: opracowanie własne.

### 2.3. Preprocesor (Radosław Wężyk)

Program Snort wraz z preprocesorami napisany został w języku programowania C. Pliki źródłowe preprocesorów znajdują się w \$SNORT\_DIR/src/preprocessors. Struktura pliku źródłowego oraz pliku nagłówkowego preprocesora jest określona przez programistów systemu Snort. Tworzenie nowego preprocesora wymaga dokładnego poznania zasad działania i sposobu implementacji całego systemu Snort.

Na wydruku 4 przedstawiono strukturę pliku nagłówkowego preprocesora

```

1 #ifndef __SPP_PREPROCESOR_H__
2 #define __SPP_PREPROCESOR_H__
3 void SetupPreprocesor(void);
4 typedef struct _PreprocStruct{
5 int x;
6 char *string;
7 }PreprocStruct;
8 int PreprocCounter=0;
9 void mySecondFunction(char *str);
10 extern PreprocStruct mySecondStruct;
11 #endif

```

**Wydruk 4: Struktura pliku nagłówkowego preprocesora.**

Instrukcje w liniach 1-3 są wymagane – pierwsze dwie to instrukcje sterujące dla kompilatora a trzecia to prototyp funkcji rejestrującej bieżący preprocesor przy inicjalizacji preprocesorów przez Snorta. Pozostałe deklaracje są opcjonalne i mają sens w przypadku, gdy zmienne będą używane przez kilka preprocesorów.

Wydruk 5 przedstawia strukturę pliku źródłowego preprocesora

```

1 #include <sys/types.h>
2 #include "plugbase.h"
3 #include "decode.h"
4 #include "spp_preprocesor.h"
5 void PreprocesorInit(u_char* args);
6 void PreprocesorFunc();
7 void PreprocesorCleanExitFunction();
8 void PreprocesorRestartFunction();
9 void SetupPreprocesor(void){
10 printf("Rejestruje preprocesor...\n");
11 RegisterPreprocessor("preprocesor", PreprocesorInit);
12 }
13 static void PreprocesorInit(u_char* args){
14 ParsePreprocesorArgs((char*)args);
15 AddFuncToPreprocList(PreprocesorFunc);
16 AddFuncToCleanExitList(PreprocesorCleanExitFunction, NULL);
17 AddFuncToRestartList(PreprocesorRestartFunction, NULL);
18 }
19 void PreprocesorFunc(Packet* p){
20 printf("Przyszedl pakiet\n");
21 }
22 void PreprocesorRestartFunction(){
23 }
24 void PreprocesorCleanExitFunction(){
25 }

```

**Wydruk 5: Struktura pliku źródłowego preprocesora.**



Pierwsze cztery instrukcje dołączają niezbędne pliki nagłówkowe. Kolejne instrukcje deklarują prototypy kolejno:

- Funkcji inicjalizującej preprocesor
- Funkcji głównej preprocesora
- Funkcji wywoływanej przy wychodzeniu z programu
- Funkcji wywoływanej przy restarcie aplikacji

W liniach 9-12 zdefiniowana jest funkcja rejestrująca preprocesor w Snorcie (prototyp funkcji `RegisterPreprocessor()` znajduje się w pliku `plugbase.h`, jej argumenty to nazwa preprocesora z pliku konfiguracyjnego oraz funkcja inicjalizująca). Kolejne 6 linii to definicja funkcji inicjalizującej preprocesor. W ciele tej funkcji znajdują się kolejno:

- Funkcja analizująca argumenty z pliku konfiguracyjnego
- Funkcja dodająca funkcję główną preprocesora do listy preprocesorów Snorta
- Funkcja rejestrująca funkcję wywoływaną przy zamykaniu aplikacji przez `ctrl+c`
- Funkcja rejestrująca funkcję wywoływaną przy restartowaniu aplikacji

Od linii 19 znajdują się definicje funkcji głównej preprocesora wywoływanej przez Snorta w momencie przechwycenia pakietu oraz definicje funkcji wywoływanych przy zamykaniu i restartowaniu aplikacji. Te trzy funkcje budują (merytoryczną) funkcjonalność preprocesora.

Algorytm zawarty w głównej funkcji preprocesora `void PreprocessorFunc(Packet* p)` zlicza parametry określonych typów ruchu sieciowego. Do pliku logowane mogą być 28 wielkości zbierane w ustawionym przez użytkownika (w pliku konfiguracyjnym `snort.conf` jako parametr inicjalizujący preprocesor) czasie: pierwsze trzy to czas rozpoczęcia próbkowania, dzień, w którym miał miejsce pomiar oraz odcinek czasu próbkowania. Pozostałe 25 wielkości to parametry ruchu sieciowego.

Główne funkcje odpowiedzialne za funkcjonalność preprocesora to:

- `char* change_time()` - pobiera datę i czas z systemu i zwraca w postaci łańcucha tekstowego `dd-mm-rr gg:mm`. Funkcja ta została zaimplementowana aby logowana data i czas były w formacie wspieranym przez arkusz kalkulacyjny Excel.
- `void PreprocessorInit(u_char* args)` - funkcja inicjalizująca preprocesor. Jest ona odpowiedzialna za wczytanie utworzonego wcześniej profilu oraz za inicjalizację funkcji parsującej parametry wejściowe. Domyślnie profil znajduje się w `/etc/` w pliku o nazwie `profile.txt`.
- `void PreprocFunction(Packet *p, void *context)` – główna funkcja preprocesora wywoływana przez Snorta w momencie przechwycenia pakietu – w niej

znajduje się cały algorytm logujący parametry ruchu sieciowego do pliku w określonych interwałach czasowych oraz wykrywanie ruchu odbiegającego od załadowanego profilu. Działanie funkcji oparte jest o strukturę p typu `Packet` zdefiniowaną w pliku nagłówkowym `decode.h`, funkcje zaimplementowane w aplikacji Snort oraz standardowe biblioteki języka C.

- `void PreprocCleanExitFunction(int signal, void *data)` – funkcja wywoływana w momencie wyłączenia aplikacji Snort. Wyświetla informacje o preprocesorze, plikach zawierających logi oraz parametry ruchu sieciowego zebrane przez cały czas działania aplikacji:
  - liczba wszystkich pakietów TCP,
  - liczba wszystkich datagramów IP,
  - liczba wszystkich datagramów UDP,
  - liczba wszystkich pakietów ICMP,
  - liczba wszystkich pakietów ARP,
  - liczba wysłanych pakietów TCP,
  - liczba wysłanych datagramów UDP,
  - liczba wysłanych pakietów ICMP,
  - liczba odebranych pakietów TCP,
  - liczba odebranych datagramów UDP,
  - liczba odebranych pakietów ICMP,
  - liczba pakietów TCP wewnątrz sieci LAN,
  - liczba datagramów UDP wewnątrz sieci LAN,
  - liczba pakietów ICMP wewnątrz sieci LAN.

Wyświetlenie powyższych parametrów pozwala m.in. na sprawdzenie poprawności działania algorytmu zliczającego oraz wykrycie faktu gubienia pakietów.

- `void ParsePreprocesorArgs((char*)args)` – funkcja, która analizuje parametry startowe zapisane w pliku konfiguracyjnym programu Snort (domyślnie `/etc/snort.conf`) i w zależności od nich ustawiająca odpowiednie flagi. Dzięki niej użytkownik może wpłynąć na sposób działania preprocesora.

## **2.4. Generator profilu (Maciej Skowroński)**

Program generujący profil sieci został napisany w języku C++. Wczytuje on plik z logami pochodzącymi z preprocesora i zapisane w katalogu `/var/log` w pliku

snort\_log2.txt. Następnie przeszukuje je w poszukiwaniu rekordów opisujących daną godzinę danego dnia i wyznacza dla niej wartość średnią oraz odchylenie standardowe. Efektem pracy programu jest profil sieci stworzony dla całego tygodnia. W profilu określone są dwie wartości dla każdego zapisywanego parametru ruchu:

- średnia arytmetyczna,
- odchylenie standardowe.

Odchylenie standardowe wyznaczane jest z następującego wzoru:

$$\sigma_s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

**Wzór 1: Odchylenie standardowe.**

Gdzie:

$x_i$  – kolejne wartości szeregu,

$\bar{x}$  – średnia arytmetyczna z próby,

$n$  – liczba elementów szeregu.

Natomiast do estymacji odchylenia standardowego z populacji użyty został pierwiastek kwadratowy z wariancji nieobciążonej:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

**Wzór 2: Pierwiastek kwadratowy z wariancji nieobciążonej.**

Gdzie:

$x_i$  – kolejne wartości szeregu,

$\bar{x}$  – średnia arytmetyczna z próby,

$n$  – liczba elementów szeregu.

Za całkowitą populację przyjmowane są dane z całego czasu istnienia sieci. Ponieważ program opiera się na danych zebranych podczas kilku tygodni, wykorzystywany jest wzór na odchylenie standardowe z próby losowej.

Na podstawie wrywkowych testów zgodności z rozkładem normalnym (test chi kwadrat) założono, że dane podlegają rozkładowi normalnemu. Wówczas prawdziwe jest, że (według: [I19]):

- 68% wartości leży w odległości  $< 1 \sigma$  od wartości oczekiwanej
- 95,5% wartości leży w odległości  $< 2 \sigma$  od wartości oczekiwanej

- 99,7% wartości leży w odległości  $< 3\sigma$  od wartości oczekiwanej (tak zwana reguła trzech sigma)

Średnia arytmetyczna i odchylenie standardowe są określone dla każdej godziny każdego dnia tygodnia.

Plik profilu zapisywany jest w `/etc/` w pliku o nazwie `profile.txt`. Profil zawiera 168 linii. Wynika to z tego, że liczba godzin w tygodniu wynosi właśnie 168. Każda linia zawiera oddzielone przecinkami następujące pola:

1. Dzień,
2. Godzina,
3. Liczba pakietów TCP średnia,
4. Liczba pakietów TCP odchylenie standardowe,
5. Liczba wysłanych pakietów TCP średnia,
6. Liczba wysłanych pakietów TCP odchylenie standardowe,
7. Liczba odebranych pakietów TCP średnia,
8. Liczba odebranych pakietów TCP odchylenie standardowe,
9. Liczba pakietów TCP wewnątrz sieci LAN średnia,
10. Liczba pakietów TCP wewnątrz sieci LAN odchylenie standardowe,
11. Liczba datagramów UDP średnia,
12. Liczba datagramów UDP odchylenie standardowe,
13. Liczba wysłanych datagramów UDP średnia,
14. Liczba wysłanych datagramów UDP odchylenie standardowe,
15. Liczba odebranych datagramów UDP średnia,
16. Liczba odebranych datagramów UDP odchylenie standardowe,
17. Liczba datagramów UDP wewnątrz sieci LAN średnia,
18. Liczba datagramów UDP wewnątrz sieci LAN odchylenie standardowe,
19. Liczba pakietów ICMP średnia,
20. Liczba pakietów ICMP odchylenie standardowe,
21. Liczba wysłanych pakietów ICMP średnia,
22. Liczba wysłanych pakietów ICMP odchylenie standardowe,
23. Liczba odebranych pakietów ICMP średnia,
24. Liczba odebranych pakietów ICMP odchylenie standardowe,
25. Liczba pakietów ICMP wewnątrz sieci LAN średnia,
26. Liczba pakietów ICMP wewnątrz sieci LAN odchylenie standardowe,
27. Liczba pakietów z włączonymi flagami SYN i ACK średnia,

28. Liczba pakietów z włączonym flagami SYN i ACK odchylenie standardowe,
29. Liczba wysłanych pakietów port 80 (usługa WWW) średnia,
30. Liczba wysłanych pakietów port 80 (usługa WWW) odchylenie standardowe,
31. Liczba odebranych pakietów port 80 (usługa WWW) średnia,
32. Liczba odebranych pakietów port 80 (usługa WWW) odchylenie standardowe,
33. Liczba wysłanych pakietów port 53 (usługa DNS) średnia,
34. Liczba wysłanych pakietów port 53 (usługa DNS) odchylenie standardowe,
35. Liczba odebranych pakietów port 53 (usługa DNS) średnia,
36. Liczba odebranych pakietów port 53 (usługa DNS) odchylenie standardowe,
37. Prędkość wysyłania danych w kB/s (TCP/IP) średnia,
38. Prędkość wysyłania danych w kB/s (TCP/IP) odchylenie standardowe,
39. Prędkość odbierania danych w kB/s (TCP/IP) średnia,
40. Prędkość odbierania danych w kB/s (TCP/IP) odchylenie standardowe,
41. Prędkość wysyłania danych port 80 w kB/s (TCP/IP, WWW) średnia,
42. Prędkość wysyłania danych port 80 w kB/s (TCP/IP, WWW) odchylenie standardowe,
43. Prędkość odbierania danych port 80 w kB/s (TCP/IP, WWW) średnia,
44. Prędkość odbierania danych port 80 w kB/s (TCP/IP, WWW) odchylenie standardowe,
45. Prędkość wysyłania danych w kB/s (UDP) średnia,
46. Prędkość wysyłania danych w kB/s (UDP) odchylenie standardowe,
47. Prędkość odbierania danych w kB/s (UDP) średnia,
48. Prędkość odbierania danych w kB/s (UDP) odchylenie standardowe,
49. Prędkość wysyłania danych port 53 w kB/s (UDP, DNS) średnia,
50. Prędkość wysyłania danych port 53 w kB/s (UDP, DNS) odchylenie standardowe,
51. Prędkość odbierania danych port 53 w kB/s (UDP, DNS) średnia,
52. Prędkość odbierania danych port 53 w kB/s (UDP, DNS) odchylenie standardowe.

Pierwszą jego kolumną jest dzień tygodnia. Zapisany jest on w formie cyfry o wartości odpowiadającej dniu tygodnia poczynając od zera. W drugiej kolumnie znajduje się godzina zapisany w formacie dwudziestoczerogodzinnym. Następne kolumny zawierają wartości średnie i odchylenie standardowe dla poszczególnych typów ruchu w sieci.

Tabela nr 2: Przykład profilu dla ruchu TCP dla poniedziałku. Źródło: opracowanie własne.

dzień tygodnia	godzina	TCP średnia	TCP odchylenie	TCP up średnia	TCP up odchylenie	TCP down średnia	TCP down odchylenie
0	0	176602	279191	13927	11441.1	14038.1	11728.5
0	1	46489.2	17163.5	22956.9	7216.82	23514.8	9976.16
0	2	60117.2	19724.1	27849.6	8091.68	32247	11673.6
0	3	62629.6	21067.9	28353.5	8674.5	34258.6	12427.8
0	4	79522.2	20162	34485.2	8544.66	45019.5	11649
0	5	64170.9	14829.1	28469	6223.29	35685.2	8755.5
0	6	132303	235905	27117.5	9403.8	34344.1	12817.2
0	7	124981	205025	28394.7	15953.9	35277.2	20629.1
0	8	68081.8	36559.1	30729.5	15780	37259.6	20867.6
0	9	139062	232764	29850.5	12722.3	36358.8	16211.3
0	10	62186.8	42230.4	28022.3	17954.2	34115.7	24311.5
0	11	44678.2	14563	20741.5	7123.52	23869	7820.75
0	12	103238	133265	27356.4	10283.7	34543.6	11448.3
0	13	88277.9	21616.5	38602.4	10593.7	49620.2	11489.5
0	14	51582.1	23388.1	21561	10883.6	29924.8	13010.2
0	15	65969.7	30381.1	28826.5	12763.4	37062.5	17893.5
0	16	123786	188774	30279.5	13548.3	37541.2	17249
0	17	136963	196841	33822.5	19718.2	42376.5	22644.7
0	18	64084.4	48380.1	30676.2	22649.4	33192.3	25575
0	19	61456.2	38641.6	29317.1	18656.2	31916.6	19966.3
0	20	58949.1	46485.9	28607.7	22422.2	30130.2	24007
0	21	59533.2	45298.8	29215.1	22304.8	30153.5	22960.8
0	22	49002	26598.2	23262.6	13938.1	22948.2	13450.2
0	23	159680	210119	31748.3	9229.81	40530.6	13948.8

Generowanie alertów odbywa się za pomocą funkcji `GenerateSnortEvent`. W zależności od wykrytej anomalii generowany jest odpowiedni alert. Dla każdego z analizowanych parametrów mogą być wygenerowane dwa typy alertu:

- Duży ruch: too high traffic,
- Mały ruch: too low traffic.

Określenie tego, czy dany ruch jest zbyt duży czy zbyt mały jest przeprowadzane dzięki wczytanemu wcześniej profilowi. Preprocesor odczytuje z systemu aktualną godzinę i dzień tygodnia. Następnie odszukuje (w tabeli zawierającej odczytany profil) wiersz, w którym znajdują się dane dotyczące ruchu w tym dniu i o tej godzinie. Jeśli aktualna wartość natężenia ruchu danego typu (ang. bitrate) wyrażona w bitach na sekundę, jest większa od średniej, do której dodano wartość odchylenia standardowego pomnożonego przez dwa (reguła dwóch sigma), generowany jest alert o zbyt dużym ruchu.

$$W \notin \langle \overline{W} - 2\sigma; \overline{W} + 2\sigma \rangle$$

**Wzór 2: Reguła dwa sigma**

Gdzie:

$W$  - aktualna wartość natężenia ruchu danego typu,

$\sigma$  - odchylenie standardowe odczytane z profilu (obliczone dla danych historycznych).

W przypadku, gdy wartość ta jest mniejsza od średniej pomniejszonej o wartość odchylenia standardowego pomnożonego przez dwa, generowany jest alert o zbyt małym ruchu. Na rysunku poniżej pokazano wygenerowane przez preprocesor AnomalyDetection alerty.

The screenshot shows the 'Basic Analysis and Security Engine (BASE)' web interface. The main heading is 'Basic Analysis and Security Engine (BASE)'. Below it, there's a search bar and a 'Home | Search' link. A message states 'Added 0 alert(s) to the Alert cache'. The 'Queried on' date is 'Sat August 05, 2006 16:41:16'. The 'Meta Criteria' section shows 'time >= [ 08 / 05 / 2006 ] [ any time ]' and a 'Clear...' button. The 'IP Criteria' is 'any', 'Layer 4 Criteria' is 'none', and 'Payload Criteria' is 'any'. The 'Summary Statistics' section lists: 'Sensors /', 'Unique Alerts (classifications)', 'Unique addresses: Source | Destination', 'Unique IP links', 'Source Port: TCP | UDP', 'Destination Port: TCP | UDP', and 'Time profile of alerts'. Below this, it says 'Displaying alerts 1-50 of 79 total'. The main table lists alerts with columns: ID, Signature, Timestamp, Source Address, Dest. Address, and Layer 4 Proto. The table contains 14 rows of alert data.

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#0-(1-13979)	maly ruch daneUDPDownKB	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#1-(1-13978)	maly ruch daneUDPUpKB	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#2-(1-13977)	maly ruch daneTCPUKB	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#3-(1-13976)	maly ruch udpDNS_countDown	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#4-(1-13975)	maly ruch udpDNS_countUp	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#5-(1-13974)	maly ruch liczbaSYNACKp	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#6-(1-13973)	maly ruch icmp_countFpDown	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#7-(1-13972)	maly ruch icmp_countFp	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#8-(1-13971)	maly ruch udp_countFpDown	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#9-(1-13970)	maly ruch udp_countFpUp	2006-08-05 12:51:48	192.168.0.5:1059	193.17.41.53:443	TCP
#10-(1-14519)	duzy ruch lanTCP	2006-08-05 15:23:06	192.168.0.5:2045	85.186.68.138:7777	TCP
#11-(1-14520)	maly ruch tcpWWW_countUp	2006-08-05 15:23:06	192.168.0.5:2045	85.186.68.138:7777	TCP
#12-(1-14521)	maly ruch tcpWWW_countDown	2006-08-05 15:23:06	192.168.0.5:2045	85.186.68.138:7777	TCP
#13-(1-14522)	maly ruch daneTCPUKB	2006-08-05 15:23:06	192.168.0.5:2045	85.186.68.138:7777	TCP
#14-(1-14523)	maly ruch daneUDPUpKB	2006-08-05 15:23:06	192.168.0.5:2045	85.186.68.138:7777	TCP

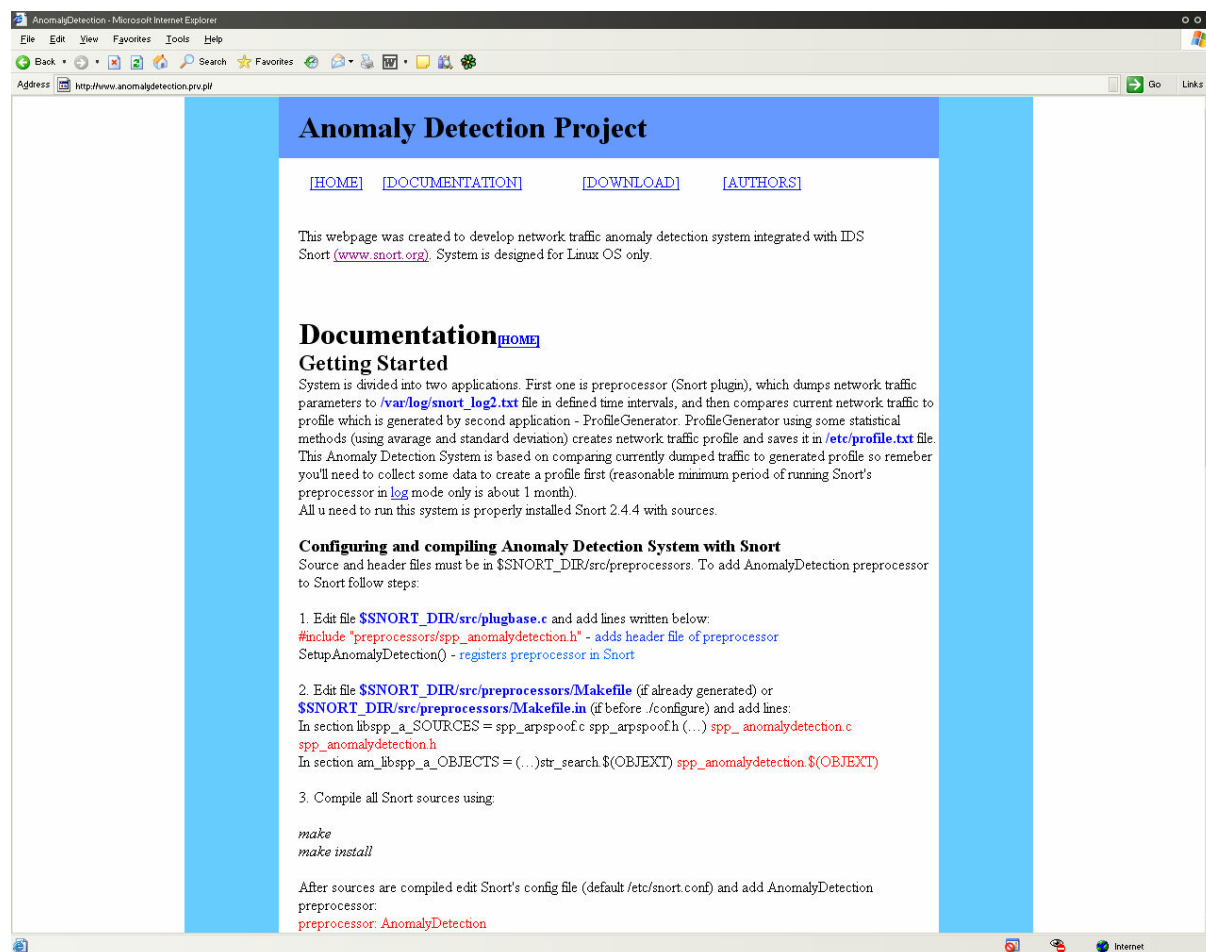
Rysunek 20: Przykład alertów wygenerowanych przez preprocesor AnomalyDetection. Źródło: opracowanie własne.

System Snort jest tak napisany, że główna funkcja preprocesora wywoływana jest w momencie przyjścia pakietu. Aby alert mógł zostać wygenerowany, funkcja GenerateSnortEvent musi dostać jako parametr pakiet. W związku z tym, w wygenerowanych przez preprocesor AnomalyDetection alertach można zobaczyć pakiety, które nie mają

związku z wygenerowanym alertem. Pakiety te są tymi, które aktualnie były przetwarzane w momencie, gdy został osiągnięty koniec czasu przewidziany na zliczanie.

## 2.5. Opis użytkowy

Pliki źródłowe preprocesora i programu do generacji profilu oraz pełna dokumentację pobrać można z domowej strony projektu (zobacz: [I21]). Pliki źródłowe programu Snort pobrać można z jego strony domowej (zobacz: [I18]).



Rysunek 21: Strona domowa opracowanego systemu. Źródło: opracowanie własne.

### 2.5.1. Instalacja (Radosław Wężyk)

Pliki źródłowe oraz nagłówkowe preprocesora muszą znajdować się w katalogu \$SNORT\_DIR/src/preprocessors. Aby dodać preprocesor do systemu Snort należy przed kompilacją całego systemu kolejno:

1. Wyedytować plik \$SNORT\_DIR/src/plugbase.c:
    - dodać pliki nagłówkowe naszego preprocesora :
- ```
#include "preprocessors/spp_anomalydetection.h"
```



- dodać do ciała funkcji `InitPreprocessors()` funkcję rejestrującą preprocesor `SetupAnomalyDetection()`.
2. Wyedytować plik `$SNORT_DIR/src/preprocessors/Makefile` (jeśli już został wygenerowany) lub `$SNORT_DIR/src/preprocessors/Makefile.in` (jeśli jeszcze nie został wygenerowany przez polecenie `./configure`) (zobacz: [199]) i dodać:
- W sekcji `libspp_a_SOURCES = spp_arpspoof.c spp_arpspoof.h (...)` dodać `spp_anomalydetection.c spp_anomalydetection.h`
  - W sekcji `am_libspp_a_OBJECTS = (...)` dodać `str_search.$(OBJEXT) spp_anomalydetection.$(OBJEXT)`

Następnie należy skompilować (lub przekompilować) Snorta za pomocą polecenia:

*make*

Następnie zainstalować poleceniem:

*make install*

Po kompilacji należy dopisać do pliku konfiguracyjnego Snorta, który znajduje się domyślnie w `etc/snort.conf`, w sekcji preprocesorów:

`Preprocessor: AnomalyDetection`

Jest to niezbędne do uruchomienia preprocesora wraz z systemem Snort.

### 2.5.2. Konfiguracja (Maciej Skowroński)

Do konfiguracji preprocesorów w programie Snort służą odpowiednie parametry konfiguracyjne dopisywane w pliku konfiguracyjnym. Umieszcza się je poprzedzone dwukropkiem bezpośrednio za nazwą preprocesora, w linii, w której jest on wpisany.

Możliwe jest użycie następujących parametrów:

- `alert` – użycie tego parametru spowoduje włączenie trybu generowania alertów.
- `log` – użycie tego parametru spowoduje włączenie trybu logowania.
- `time` – użycie tego parametru pozwala na określenie czasu podczas którego zbierane będą logi. Po tym słowie należy podać liczbę całkowitą dodatnią, która będzie odcinkiem czasu w sekundach.
- `sigma` – użycie tego parametru pozwala na określenie wartości mnożnika sigmy. Po tym słowie należy podać liczbę. W przypadku liczby niecałkowitej znakiem oddzielającym część całkowitą od części ułamkowej jest kropka.

Parametry te należy oddzielać między sobą znakiem spacji. Wartość czasu po słowie „time” oraz wartość mnożnika sigmy należy również oddzielić od niego poprzedzającego go słowa spacją. Inicjalizacja preprocesora może mieć na przykład postać:

```
Preprocessor: anomalydetection log alert time 600 sigma 2.5
```

Domyślnie preprocesor ma wyłączony tryb logowania i generowania alertów. Czas zbierania logów jest standardowo ustawiony na 600 sekund natomiast mnożnik sigmy ma wartość 2.

Istotne jest, aby najpierw stworzyć kompletny profil na podstawie przynajmniej jednego tygodnia. W przeciwnym wypadku włączenie opcji generowania alertów może spowodować niestabilność programu.

### **2.5.3. Uruchamianie systemu Snort z preprocesorem**

#### **AnomalyDetection. (Radosław Wężyk)**

Podczas włączania systemu niezbędne jest podanie przy użyciu flagi `-h` adresu IP hosta, na którym ma działać aplikacja oraz maski podsieci:

```
root@serwer:~# /usr/local/snort/bin/snort -c  
/usr/local/snort/etc/snort.conf -h 10.0.0.1/24
```

Pliki z logami znajdują się w katalogu `/var/log/`. Utworzone zostaną dwa pliki:

1. Plik o nazwie `snort_log.txt` zawierający przechwycone pakiety z ostatnich 5 minut próbkowania w następującym formacie:
  1. czas nadejścia pakietu,
  2. typ pakietu,
  3. flagi TCP,
  4. adres źródłowy,
  5. rozmiar pakietu,
  6. rozmiar nagłówka IP.

Plik ten pełni funkcję pomocniczą – służy do testowania i porównywania przechwytywanych danych w czasie rozbudowy aplikacji.

```
192.168.0.1 - Ntutty
Stream Trackers: 161
Stream flushes: 120
Segments used: 271
Stream4 Memory Faults: 0

=====
=====Anomaly Detection v1.00=====
Przechwycone pakiety:
liczba pakietow TCP : 3212
liczba datagramow IP : 3545
liczba datagramow UDP: 327
liczba pakietow ICMP : 6
liczba pakietow ARP : 36
ruch w LAN TCP:3212,UDP:327,ICMP:6
ruch zalogowany w /var/log/log_snort.txt
Final Flow Statistics
,----[ FLOWCACHE STATS ]-----
Memcap: 10485760 Overhead Bytes 16400 used(%0.427828)/blocks (44861/160)
```

Rysunek 22: Informacje pokazywane przy wyłączeniu preprocesora AnomalyDetection. Źródło: opracowanie własne.

2. Plik o nazwie `snort_log2.txt` zawierający parametry ruchu sieciowego zbierane w określonych przez użytkownika odcinkach czas od momentu włączenia systemu. Każda linijka pliku przedstawia 28 parametrów oddzielonych przecinkami – 3 parametry określające czas i datę oraz 25 parametrów ruchu sieciowego zebranych w ciągu ustalonego przez użytkownika w pliku konfiguracyjnym czasu.
  1. Czas i data rozpoczęcia próbkowania w formacie dd-mm-rr gg:mm,
  2. Dzień,
  3. Czas próbkowania (domyślnie 600 sekund),
  4. Liczba pakietów TCP,
  5. Liczba wysłanych pakietów TCP,
  6. Liczba odebranych pakietów TCP,
  7. Liczba pakietów TCP wewnątrz sieci LAN,
  8. Liczba datagramów UDP,
  9. Liczba wysłanych datagramów UDP
  10. Liczba odebranych datagramów UDP,
  11. Liczba datagramów UDP wewnątrz sieci LAN,
  12. Liczba pakietów ICMP,
  13. Liczba wysłanych pakietów ICMP,
  14. Liczba odebranych pakietów ICMP,

15. Liczba pakietów ICMP wewnątrz sieci LAN,
16. Liczba pakietów z włączonymi flagami SYN i ACK,
17. Liczba wysłanych pakietów port 80 (usługa WWW),
18. Liczba odebranych pakietów port 80 (usługa WWW),
19. Liczba wysłanych pakietów port 53 (usługa DNS),
20. Liczba odebranych pakietów port 53 (usługa DNS),
21. Prędkość wysyłania danych w kB/s (TCP/IP),
22. Prędkość ściągania danych w kB/s (TCP/IP),
23. Prędkość wysyłania danych port 80 w kB/s (TCP/IP, WWW),
24. Prędkość ściągania danych port 80 w kB/s (TCP/IP, WWW),
25. Prędkość wysyłania danych w kB/s (UDP),
26. Prędkość ściągania danych w kB/s (UDP),
27. Prędkość wysyłania danych port 53 w kB/s (UDP, DNS),
28. Prędkość ściągania danych port 53 w kB/s (UDP, DNS),

Plik z logami ma strukturę umożliwiającą łatwy import danych do arkusza kalkulacyjnego Microsoft Excel. Excel jest dobrym narzędziem do budowy statystyk, wstępnej analizy, kontroli poprawności działania tworzonych algorytmów oraz wizualizacji wyników.

Użycie programu generującego profil sieci sprowadza się do jego uruchomienia za pomocą polecenia:

```
./ProfileGenerator
```

Program automatycznie odczyta plik z logami preprocesora `snort_log2.txt` zlokalizowany w `/var/log`. Następnie wygenerowany zostanie na ich podstawie profil sieci. Zostanie on zapisany w pliku o nazwie `profile.txt` i umieszczony w katalogu `/etc/`. Program ten nie przyjmuje parametrów wejściowych.

Tabela nr 3 Statystyki ruchu sieciowego stworzone w oparciu o dane zebrane przez preprocesor w sieci złożonej z 30 komputerów. Źródło: opracowanie własne.

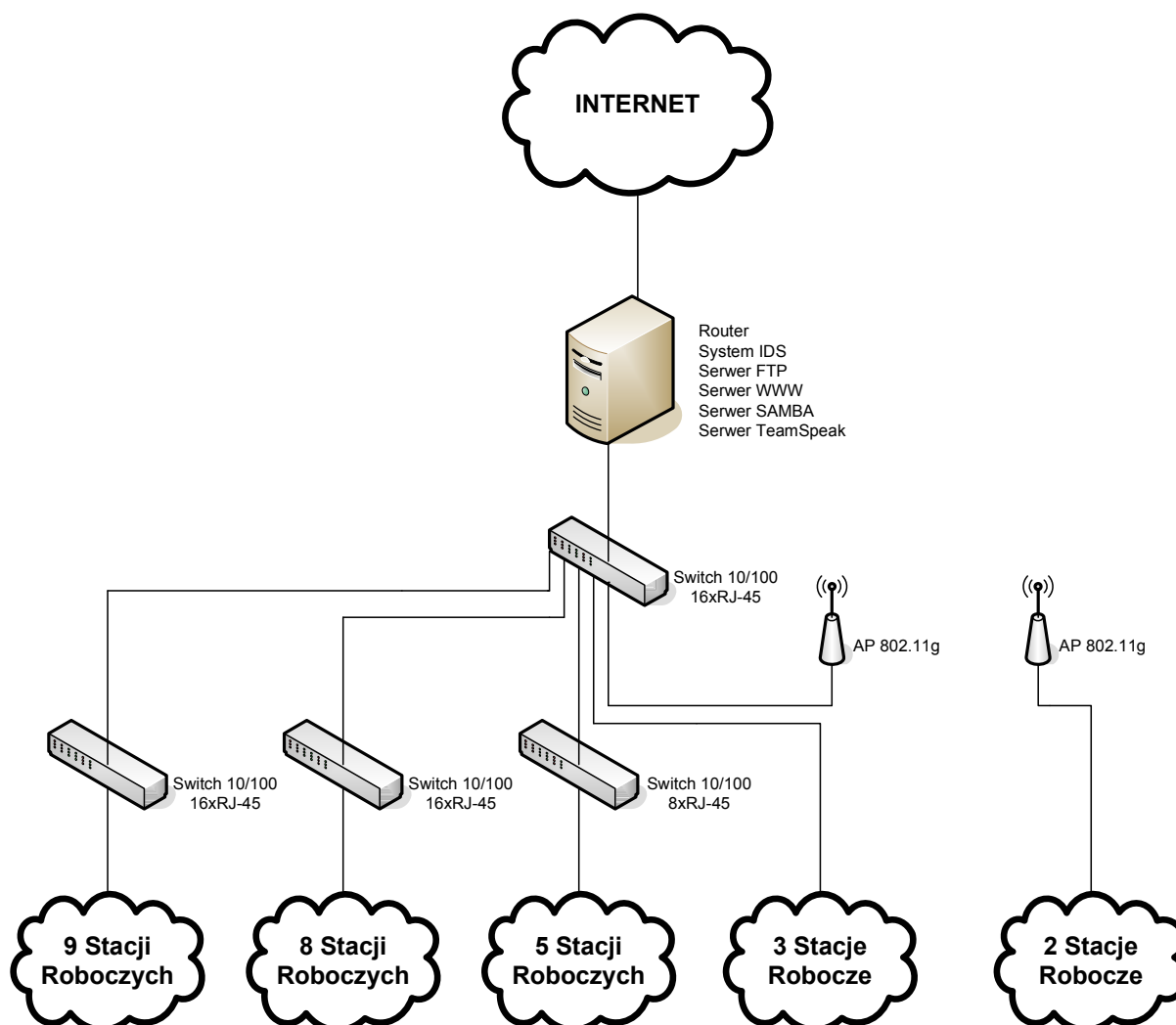
|                |       |                           | L I C Z B A P A K I E T Ó W |        |          |         |      |        |          |         |      |         |           |          |         |        |          |        |          |        |          |        | P R Ę D K O Ś Ć [kB/s] |        |          |        |          |  |  |  |
|----------------|-------|---------------------------|-----------------------------|--------|----------|---------|------|--------|----------|---------|------|---------|-----------|----------|---------|--------|----------|--------|----------|--------|----------|--------|------------------------|--------|----------|--------|----------|--|--|--|
| Czas           | Dzień | Czas probko-<br>wania [s] | TCP                         | TCP up | TCP down | TCP LAN | UDP  | UDP up | UDP down | UDP LAN | ICMP | ICMP Up | ICMP down | ICMP LAN | SYN&ACK | WWW up | WWW down | DNS up | DNS down | TCP up | TCP down | WWW up | WWW down               | UDP up | UDP down | DNS up | DNS down |  |  |  |
| 22-06-06 09:04 | Thu   | 600                       | 44503                       | 19224  | 25087    | 192     | 1556 | 758    | 791      | 7       | 5    | 0       | 5         | 0        | 917     | 768    | 1053     | 20     | 19       | 10,52  | 27,36    | 0,16   | 1,76                   | 0,06   | 0,08     | 0      | 0,01     |  |  |  |
| 22-06-06 09:14 | Thu   | 600                       | 48247                       | 20825  | 27422    | 0       | 1630 | 849    | 778      | 3       | 5    | 0       | 5         | 0        | 889     | 0      | 0        | 2      | 0        | 10,36  | 32,42    | 0      | 0                      | 0,07   | 0,08     | 0      | 0        |  |  |  |
| 22-06-06 09:24 | Thu   | 600                       | 51329                       | 21973  | 29126    | 230     | 1392 | 603    | 778      | 11      | 17   | 0       | 15        | 2        | 1002    | 935    | 1045     | 13     | 12       | 10,73  | 34,91    | 0,3    | 1,45                   | 0,05   | 0,08     | 0      | 0        |  |  |  |
| 22-06-06 09:34 | Thu   | 600                       | 26981                       | 12098  | 14864    | 19      | 1150 | 386    | 745      | 19      | 15   | 2       | 13        | 0        | 614     | 0      | 0        | 1      | 1        | 8,03   | 11,55    | 0      | 0                      | 0,03   | 0,07     | 0      | 0        |  |  |  |
| 22-06-06 09:44 | Thu   | 600                       | 29145                       | 13491  | 15634    | 20      | 1163 | 375    | 730      | 58      | 1    | 0       | 1         | 0        | 673     | 452    | 400      | 20     | 20       | 8,17   | 9,39     | 0,08   | 0,4                    | 0,03   | 0,07     | 0      | 0,01     |  |  |  |
| 22-06-06 09:54 | Thu   | 600                       | 35878                       | 15459  | 20314    | 105     | 1606 | 737    | 805      | 64      | 99   | 2       | 93        | 4        | 714     | 3002   | 6518     | 46     | 46       | 8,42   | 20,7     | 0,36   | 14,2                   | 0,07   | 0,09     | 0      | 0,01     |  |  |  |
| 22-06-06 10:04 | Thu   | 600                       | 43817                       | 18695  | 25071    | 51      | 1331 | 696    | 605      | 30      | 76   | 0       | 76        | 0        | 913     | 4530   | 9299     | 45     | 45       | 9,17   | 29,84    | 0,61   | 19,57                  | 0,07   | 0,07     | 0      | 0,02     |  |  |  |
| 22-06-06 10:14 | Thu   | 600                       | 73773                       | 28319  | 45405    | 49      | 1780 | 905    | 773      | 102     | 91   | 2       | 87        | 2        | 1059    | 14110  | 29929    | 90     | 89       | 10,06  | 73,65    | 1,47   | 65,35                  | 0,11   | 0,1      | 0,01   | 0,02     |  |  |  |
| 22-06-06 10:24 | Thu   | 600                       | 51383                       | 23600  | 27689    | 94      | 1936 | 1013   | 897      | 26      | 70   | 0       | 68        | 2        | 1033    | 4890   | 6923     | 66     | 66       | 11,28  | 26,3     | 1,23   | 11,85                  | 0,1    | 0,13     | 0,01   | 0,02     |  |  |  |
| 22-06-06 10:34 | Thu   | 600                       | 49506                       | 22677  | 26786    | 43      | 2088 | 1090   | 919      | 79      | 152  | 19      | 133       | 0        | 849     | 670    | 594      | 68     | 68       | 10,52  | 27,53    | 0,12   | 0,51                   | 0,13   | 0,13     | 0,01   | 0,02     |  |  |  |
| 22-06-06 10:44 | Thu   | 600                       | 78715                       | 34472  | 44195    | 48      | 1853 | 981    | 822      | 50      | 128  | 2       | 126       | 0        | 877     | 198    | 190      | 25     | 23       | 12,26  | 60,57    | 0,03   | 0,2                    | 0,14   | 0,1      | 0      | 0,01     |  |  |  |
| 22-06-06 10:54 | Thu   | 600                       | 6E+05                       | 37391  | 49418    | 472400  | 1923 | 1041   | 842      | 40      | 93   | 0       | 93        | 0        | 944     | 5777   | 10541    | 27     | 23       | 12,15  | 72,46    | 0,54   | 24,38                  | 0,15   | 0,11     | 0      | 0,01     |  |  |  |
| 22-06-06 11:04 | Thu   | 600                       | 73341                       | 32902  | 40302    | 137     | 2047 | 1080   | 934      | 33      | 78   | 0       | 76        | 2        | 1157    | 4621   | 6307     | 87     | 85       | 12,94  | 47,47    | 1,01   | 10,74                  | 0,15   | 0,14     | 0,01   | 0,03     |  |  |  |
| 22-06-06 11:14 | Thu   | 600                       | 77372                       | 34127  | 43203    | 42      | 1897 | 974    | 884      | 39      | 79   | 4       | 75        | 0        | 1072    | 2813   | 3914     | 24     | 24       | 11,16  | 54,08    | 0,74   | 6,96                   | 0,13   | 0,1      | 0      | 0,01     |  |  |  |
| 22-06-06 11:24 | Thu   | 600                       | 84884                       | 37358  | 47154    | 372     | 3637 | 1859   | 1686     | 92      | 87   | 21      | 66        | 0        | 1371    | 5229   | 6356     | 98     | 92       | 11,04  | 59,39    | 1,17   | 9,43                   | 1,71   | 0,19     | 0,01   | 0,03     |  |  |  |
| 22-06-06 11:34 | Thu   | 600                       | 91452                       | 39217  | 52097    | 138     | 4146 | 2059   | 1973     | 114     | 78   | 0       | 76        | 2        | 1219    | 10113  | 13242    | 63     | 63       | 11,01  | 69,94    | 2,02   | 23,55                  | 2,98   | 0,2      | 0,01   | 0,02     |  |  |  |
| 22-06-06 11:44 | Thu   | 600                       | 83548                       | 35829  | 47627    | 92      | 1661 | 834    | 770      | 57      | 54   | 0       | 54        | 0        | 1301    | 10762  | 14280    | 33     | 33       | 11,02  | 63,64    | 2,17   | 25,17                  | 0,23   | 0,09     | 0      | 0,01     |  |  |  |
| 22-06-06 11:54 | Thu   | 600                       | 69593                       | 31000  | 38514    | 79      | 2173 | 1184   | 962      | 27      | 61   | 0       | 61        | 0        | 1147    | 7986   | 9122     | 86     | 83       | 10,81  | 45,18    | 2,16   | 14,21                  | 0,59   | 0,14     | 0,01   | 0,03     |  |  |  |
| 22-06-06 12:04 | Thu   | 600                       | 56017                       | 25245  | 30640    | 132     | 1432 | 837    | 556      | 39      | 72   | 0       | 70        | 2        | 857     | 1538   | 1874     | 39     | 39       | 9,36   | 33,84    | 0,35   | 2,79                   | 0,16   | 0,07     | 0      | 0,02     |  |  |  |
| 22-06-06 12:14 | Thu   | 600                       | 63068                       | 29369  | 33483    | 216     | 2393 | 1263   | 1064     | 66      | 41   | 0       | 41        | 0        | 999     | 3957   | 4923     | 96     | 90       | 15,43  | 34,16    | 0,85   | 7,76                   | 1,24   | 0,12     | 0,01   | 0,03     |  |  |  |
| 22-06-06 12:24 | Thu   | 600                       | 1E+05                       | 46751  | 60414    | 92      | 5149 | 3048   | 2083     | 18      | 185  | 0       | 185       | 0        | 1140    | 15786  | 26525    | 97     | 88       | 19,02  | 88,5     | 1,61   | 59,07                  | 0,97   | 0,42     | 0,01   | 0,03     |  |  |  |
| 22-06-06 12:34 | Thu   | 600                       | 69577                       | 32549  | 36901    | 127     | 1917 | 1058   | 818      | 41      | 81   | 18      | 61        | 2        | 833     | 328    | 304      | 36     | 36       | 19,01  | 36,97    | 0,05   | 0,3                    | 0,23   | 0,12     | 0      | 0,02     |  |  |  |
| 22-06-06 12:44 | Thu   | 600                       | 70152                       | 31934  | 38199    | 19      | 6730 | 700    | 582      | 5448    | 24   | 0       | 24        | 0        | 931     | 1865   | 1943     | 37     | 37       | 15,14  | 42,77    | 0,64   | 2,68                   | 0,08   | 0,09     | 0      | 0,01     |  |  |  |
| 22-06-06 12:54 | Thu   | 600                       | 68026                       | 31735  | 36272    | 19      | 6584 | 618    | 481      | 5485    | 33   | 0       | 33        | 0        | 1061    | 2904   | 3283     | 32     | 31       | 15,64  | 35,51    | 0,8    | 4,47                   | 0,07   | 0,07     | 0      | 0,01     |  |  |  |
| 22-06-06 13:04 | Thu   | 600                       | 54271                       | 25862  | 28300    | 109     | 8566 | 745    | 553      | 7268    | 45   | 0       | 43        | 2        | 791     | 8      | 5        | 2      | 2        | 14,65  | 22,75    | 0      | 0                      | 0,1    | 0,08     | 0      | 0        |  |  |  |
| 22-06-06 13:14 | Thu   | 600                       | 59484                       | 27937  | 31547    | 0       | 5223 | 792    | 578      | 3853    | 23   | 0       | 23        | 0        | 839     | 1367   | 2256     | 23     | 23       | 14,97  | 28,24    | 0,19   | 4,15                   | 0,18   | 0,08     | 0      | 0        |  |  |  |
| 22-06-06 13:24 | Thu   | 600                       | 73178                       | 32895  | 40245    | 38      | 8496 | 2958   | 1806     | 3732    | 147  | 0       | 147       | 0        | 989     | 4487   | 8659     | 58     | 58       | 14,93  | 47,19    | 0,47   | 18,38                  | 0,42   | 0,49     | 0,01   | 0,01     |  |  |  |
| 22-06-06 13:34 | Thu   | 600                       | 73410                       | 33122  | 40269    | 19      | 4447 | 805    | 598      | 3044    | 35   | 0       | 35        | 0        | 970     | 3562   | 7295     | 91     | 88       | 15,74  | 47       | 0,36   | 16,42                  | 0,14   | 0,09     | 0,01   | 0,01     |  |  |  |
| 22-06-06 13:44 | Thu   | 600                       | 66050                       | 31526  | 34457    | 67      | 4233 | 641    | 471      | 3121    | 48   | 2       | 44        | 2        | 817     | 28     | 16       | 3      | 3        | 19,29  | 30,22    | 0      | 0                      | 0,08   | 0,06     | 0      | 0        |  |  |  |
| 22-06-06 13:54 | Thu   | 600                       | 68020                       | 31437  | 36545    | 38      | 3657 | 674    | 497      | 2486    | 31   | 0       | 31        | 0        | 804     | 587    | 725      | 20     | 20       | 15,31  | 37,82    | 0,13   | 1,18                   | 0,08   | 0,08     | 0      | 0        |  |  |  |

## Rozdział trzeci. Badania empiryczne.

### 3.1. Przedmiot badań

Przedmiotem badań była detekcja anomalii za pomocą napisanego oprogramowania w przykładowej sieci LAN. Napisany preprocesor został użyty do zbierania logów w sieci przedstawionej na rysunku 22. Jest to prosta sieć osiedlowa w skład, której wchodzi około 30 stacji roboczych. Snort pracował na routerze będącym jednocześnie bramą do Internetu. W sieci pracowały następujące usługi sieciowe:

- Serwer FTP,
- Serwer WWW,
- SAMBA,
- Serwer TeamSpeak.



Rysunek 23: Schemat sieci, w której zostały przeprowadzone pomiary. Źródło: opracowanie własne.

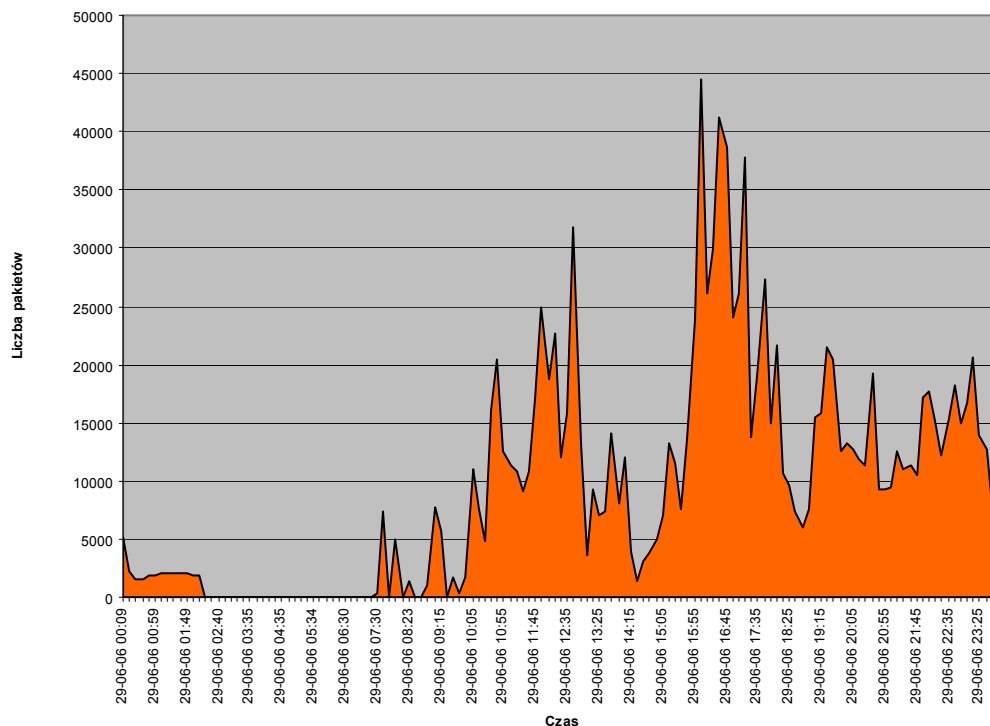
Komputery znajdujące się w tej sieci są w większości komputerami użytkowymi w warunkach domowych.

Dane zbierane były od 19 czerwca do 1 sierpnia. Profil porównany został z danymi pochodzącymi z jednego tygodnia (od 3.07.2006 do 9.07.2006). Przeprowadzone badanie objęło 6040 pomiarów 25 parametrów dokonywanych co 10 minut.

### **3.2. Wyniki pomiarów i wnioski**

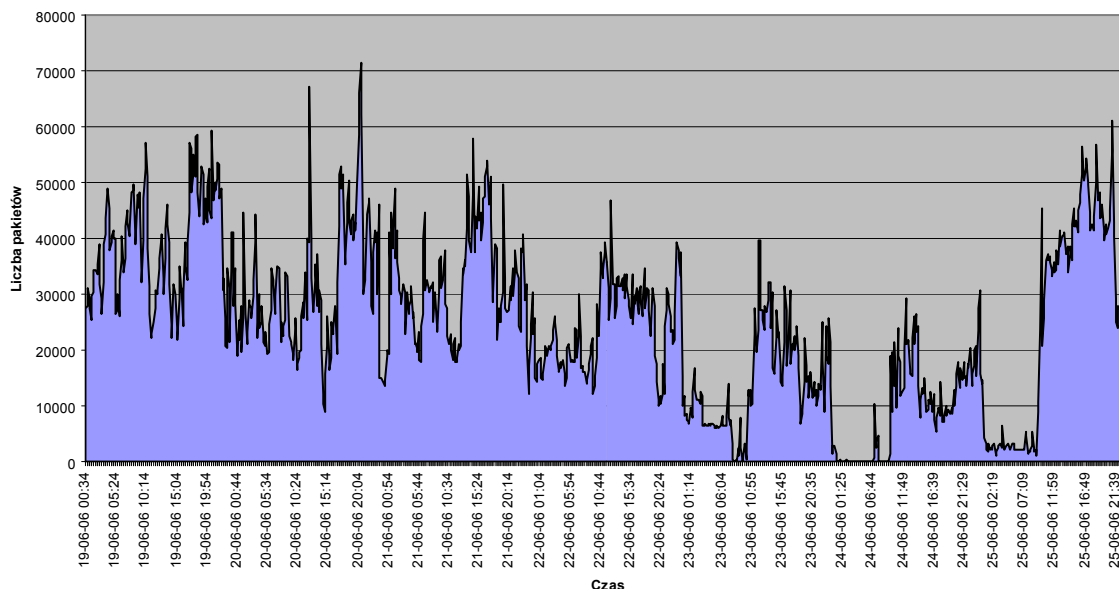
Szczegółowe wykresy przedstawiające otrzymane wyniki zostały przedstawione w załączniku 1 (rysunki od 28 do 99) . Na podstawie zebranych danych utworzony został profil badanej sieci.

Na wykresie z rysunku 25 przedstawiona jest statystyka pakietów TCP zebrana w ciągu 7 dni. Widoczne są wzrosty liczby pakietów w godzinach popołudniowych oraz wieczornych. W nocy liczba pakietów jest niewielka, czasami nawet równa zero. Takie zjawisko powtarza się we wszystkich typach obserwowanego ruchu sieciowego. Widoczny jest również wzrost wykorzystania łącza w weekendy (rysunek 25). Szczegółowy przebieg szeregu dobowego przedstawiony jest na rysunku 24, na którym wyraźnie widać wygaśnięcie ruchu sieciowego w godzinach nocnych. Na rysunku tym zaobserwować można również zależność natężenia ruchu od pory dnia. Ruch zwiększa się o godzinie 10, kolejny wzrost ruchu widoczny jest około godziny 15, co jest prawdopodobnie spowodowane tym, że ludzie po powrocie z pracy zaczynają korzystać z sieci.



**Rysunek 24: Statystyka wysłanych pakietów TCP (przebieg dobowy).** Źródło: opracowanie własne.

Na tygodniowym wykresie wysłanych pakietów TCP (rysunek 25) można zaobserwować, że liczba pakietów maleje w godzinach nocnych i utrzymuje dużą wartość w czasie weekendu.



**Rysunek 25: Statystyka wysłanych pakietów TCP (przebieg tygodniowy).** Źródło: opracowanie własne.



Ruch pakietów TCP (załącznik 1, rysunek 33) - wewnątrz sieci LAN związany jest głównie z usługami działającymi na routerze: serwer FTP, WWW, TeamSpeak, SAMBA. Na wykresach widoczne są duże wartości, które w przypadku obserwowanej sieci opisują ruch w protokołach FTP oraz SAMBA. Pozostałe usługi generują niewielki ruch, niewidoczny przy tak wyskalowanych wykresach. Wewnątrz sieci LAN w godzinach nocnych można zauważyć brak ruchu (załącznik 1, rysunek 34). Można, zatem spodziewać się, że obserwowany ruch charakteryzuje się podwójną okresowością: dobową (wzrost ruchu w godzinach popołudniowych) i tygodniową (wzrost ruchu w weekendy). Prawdopodobnie przy dłuższych szeregach da się zaobserwować również okresowość roczna (związana ze zjawiskami takimi jak początek i koniec wakacji czy święta stałe)

Na rysunkach 30 i 50 zawartych w załączniku 1, można zaobserwować bezpośredni związek między ruchem TCP a liczbą nowych połączeń. Wraz ze wzrostem liczby połączeń wzrasta ruch TCP.

Na rysunkach zawartych w załączniku 1 można zauważyć, że ruch TCP w ramach połączeń ze stronami WWW stanowi często prawie połowę całkowitego ruchu tego typu. Zależność tę widać na przebiegach dobowych (rysunek 32 i rysunek 52, załącznik 1) wysyłanych danych jak i na przebiegach dobowych (rysunek 31 i rysunek 54, załącznik 1) odbieranych danych. Wynika to z tego, że główną metodą wykorzystania badanej sieci przez użytkowników jest przeglądanie stron WWW.

Na wykresach przedstawiających statystyki wysłanych i odebranych pakietów DNS (rysunek 55 i rysunek 57, załącznik 1) widać wyraźnie cykl dobowy. W godzinach nocnych ruch tego typu praktycznie zamiera. Wzrasta od około godziny 7 rano, największe wartości osiąga w godzinach popołudniowych, a następnie spada do wartości minimalnych około godziny pierwszej w nocy. Porównując rysunki 55, 56, 57, 58 (załącznik 1) przedstawiające ruch UDP kierowany na port 53 (czyli zapytania do serwera DNS) z rysunkami 51, 52, 53, 54 (załącznik 1) przedstawiającymi ruch TCP kierowany na port 80 (czyli ruch WWW) można zauważyć ich podobieństwo. Zwiększenie ruchu TCP na tym porcie wynika z większej aktywności użytkowników przy przeglądaniu stron internetowych. Gdy któryś z użytkowników wprowadzi nowy adres strony WWW, system operacyjny wysyła zapytanie do serwera DNS w celu uzyskanie adresu IP, na którym dana strona się znajduje. Na wykresach widać wyraźnie, że w badanej sieci wraz ze wzrostem ruchu TCP kierowany na port 80 wzrasta w sposób proporcjonalny ruch UDP kierowany na port 53.

### **3.2.1. Porównanie otrzymanych wyników z wyznaczoną wartością średnią.**

Na wykresach zawartych w załączniku 2 (rysunki od 100 do 124) kolorem żółtym przedstawiono zebrane dane, kolorem niebieskim natomiast wyznaczoną przez program ProfileGenerator średnią wartość ruchu danego typu.

Można zauważyć, że zarejestrowane dane w niektórych przypadkach (na przykład ruch TCP - rysunek 100) znacznie odbiegają od wyznaczonej wartości średniej. Wynika to z tego, że dane zbierane były w sieci, której użytkownikami są w większości prywatne osoby. Efektem tego jest brak stałej regularności w korzystaniu z sieci (na przykład podczas urlopu wykorzystanie łącza może się zwiększyć, jeśli użytkownik pozostaje w domu i gra w gry sieciowe, lub wręcz spaść do zera, jeśli wyjedzie na wakacje). Innym powodem było też to, że początek zbierania danych przypadł na czas końca letniej sesji egzaminacyjnej na uczelniach (mniejsze wykorzystanie łącza przez studentów) i koniec roku szkolnego.

### **3.2.2. Wykrywanie anomalii**

Na wykresach zawartych w załącznikach 3, 4 i 5 (rysunki od 125 do 200) przedstawiono porównanie zapisanych parametrów ruchu z wyznaczonym profilem. Linia przerywaną zaznaczona jest granica wyznaczona przez dodanie i odjęcie od wartości średniej odchylenia standardowego pomnożonego przez odpowiednią wartość (2; 2,5; 3). Na większości wykresów widać ją tylko ponad danymi, które są zaznaczone na żółto. Jest to spowodowane tym, że gdy wartość graniczna schodziła poniżej zera jej wartość ustawiana była na zero (pojęcie ujemnego natężenia ruchu sieciowego nie ma sensu fizycznego). Czerwone krzyżyki oznaczają sytuacje, w których wygenerowany został alert.

#### **3.2.2.1. Reguła 2 sigma**

W załączniku 3 na wykresach 125 do 149 zebrano wyniki, dla których wartość graniczna ustalona została na  $\pm 2 \cdot \text{odchylenie standardowe (sigma)}$ . Ta wartość mnożnika sigmy jest domyślna w opracowanym systemie. Łączna liczba alertów przedstawionych na wykresach wyniosła 1574.

### 3.2.2.2. Reguła 2,5 sigma

W załączniku 4 przedstawione są wykresy (rysunki 150 do 175), na których wartość graniczna ustalona została na średnia  $\pm 2,5 \bullet$  sigma.

Oczywiście w porównaniu z wykresami dla mnożnika równego 2, liczba wygenerowanych alertów zmniejszyła się.

Łączna liczba alertów przedstawionych na wykresach wyniosła 1026.

### 3.2.2.3. Reguła 3 sigma

W załączniku 5 przedstawione są wykresy (rysunki 176 do 200), na których wartość graniczna ustalona została na średnia  $\pm 3 \bullet$  sigma.

W porównaniu do mnożnika 2 i 2,5 liczba alertów przy mnożniku równym 3 uległa zmniejszeniu. System jednak poprawnie wykrywa sytuacje, w których ruch wyraźnie odbiega od reszty profilu i generuje w tym momencie alert. Widać to bardzo wyraźnie na rysunkach 176, 179, 182, 194. 198.

Łączna liczba alertów przedstawionych na wykresach wyniosła 731.

### 3.2.2.4. Liczba alertów a wartość mnożnika sigma

Na rysunkach zawartych w załączniku 6 (rysunku od 201 do 207) przedstawiono jak zmienia się liczba generowanych alertów przy zmianie wartości mnożnika sigmy.

Jak można zauważyć na rysunkach z załącznika 6, największy spadek liczby alertów ma miejsce przy zmianie mnożnika sigmy z wartości 1 na 2. Przy zmianie z wartości 2 na 2,5 spadek jest już mniejszy. Przy dalszym zwiększaniu wartości mnożnika, liczba alertów spada zdecydowanie wolniej i powoli dąży do zera. Dokładne wyniki przedstawione są w tabeli 4.

Na rysunku zbiorczym (rysunek 26) widać, że prędkość spadku liczby alertów w przypadku ruchu ICMP, ICMP up, ICMP down, UDP up, UDP down, WWW up, WWW down, DNS up oraz DNS down jest wolniejsza niż innych analizowanych typów ruchu. Co istotne, nie spełniają one reguły 3 sigma znanej dla rozkładu normalnego. Na tej podstawie można wyciągnąć wniosek, że te typy ruchu nie mają rozkładu normalnego. Do analizy ruchu tego typu powinno się więc użyć innego rozkładu. Jednak w przypadku wszystkich analizowanych szeregów prawdziwa jest reguła 2 sigma dla rozkładu normalnego (to jest 95,5% wszystkich pomiarów mieści się w przedziale średnia  $\pm 2 \bullet$  odchylenie standardowe). Zatem charakter ruchu wydaje się być dość jednorodny, co zresztą zgodne jest z

oczekiwaniami i ze znanym z literatury przedmiotu podejściem, w którym ruch sieciowy rozpatruje się jako zjawisko o dużym współczynniku samopodobieństwa. Wykresy poszczególnych typów ruchu znajdują się w załączniku 6 (rysunki 201-207).

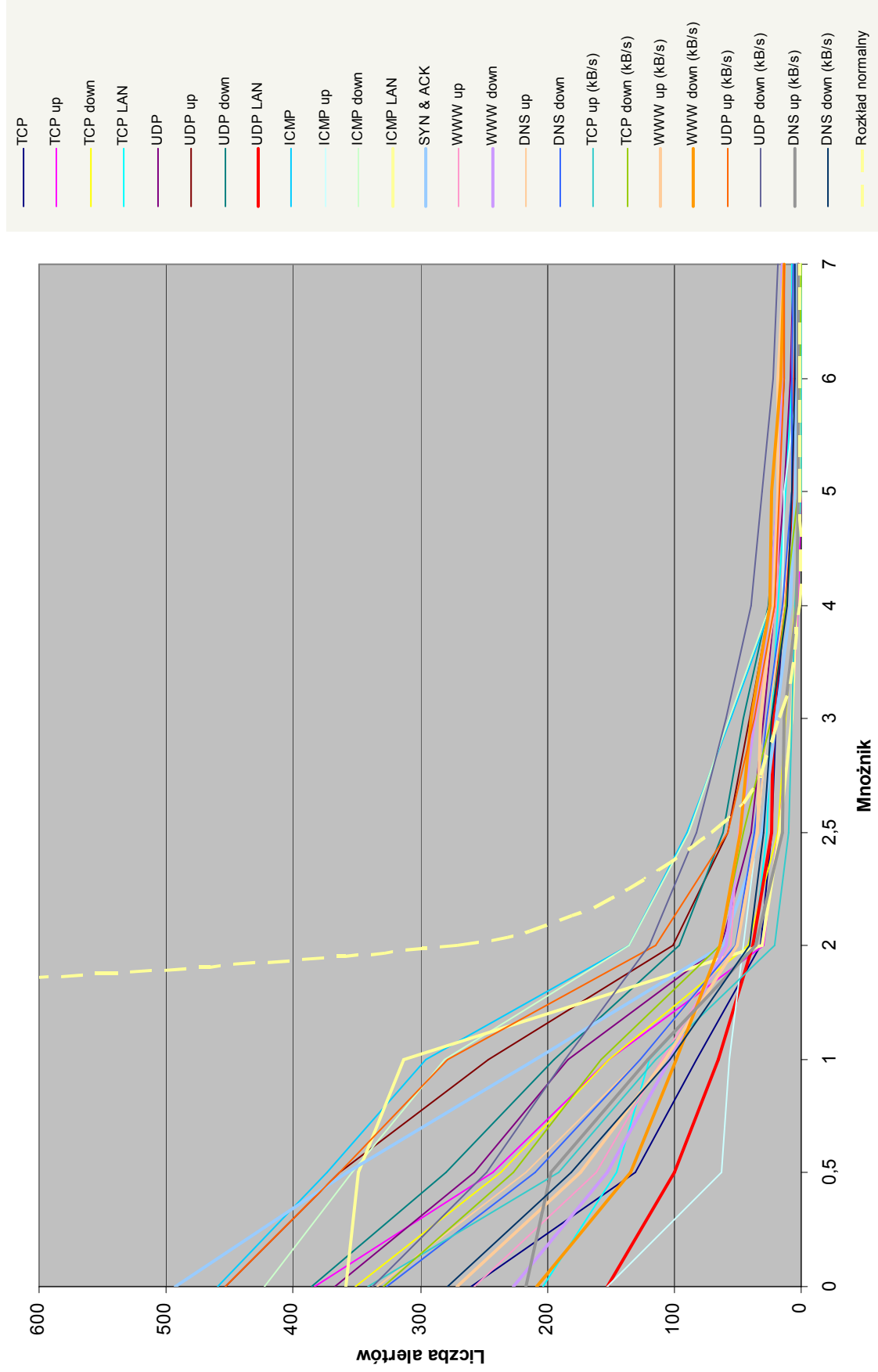
Rysunek 26 przedstawia zależność między mnożnikiem sigmy a liczbą wygenerowanych alertów. Widać na nim wyraźnie, że największy spadek liczby generowanych alertów jest przy przejściu z mnożnika 1 na 2. Największy spadek widoczny jest dla ruchu ICMP wewnątrz sieci LAN. Liczba wygenerowanych przez niego alertów spada z 313 do 31, a przy 2.5 sigma wynosi już 17.

Na podstawie rysunków 26 i 27 można wyciągnąć wniosek, że wartości mnożnika sigmy mniejsze od 2 są w praktyce nieużyteczne. Jest to spowodowane tym, że przy takiej wartości generowana byłaby zbyt duża liczba fałszywych alarmów typu false positive. Z rysunków tych widać również, że wartość mnożnika sigmy wpływa inaczej na różne rodzaje ruchu. Dla wyciągnięcia dalej idących wniosków wskazana jest jednak głębsza analiza charakteru ruchu na podstawie dłuższych szeregów czasowych, tym bardziej, że w różnych rodzajach sieci charakterystyka ruchu może mieć różny charakter (na przykład w zakładzie pracy nieczynnym w nocy większość ruchu będzie miała miejsce za dnia). Zaimplementowany preprocesor ma możliwość pracy z zadaną przez użytkownika wartością mnożnika sigmy.

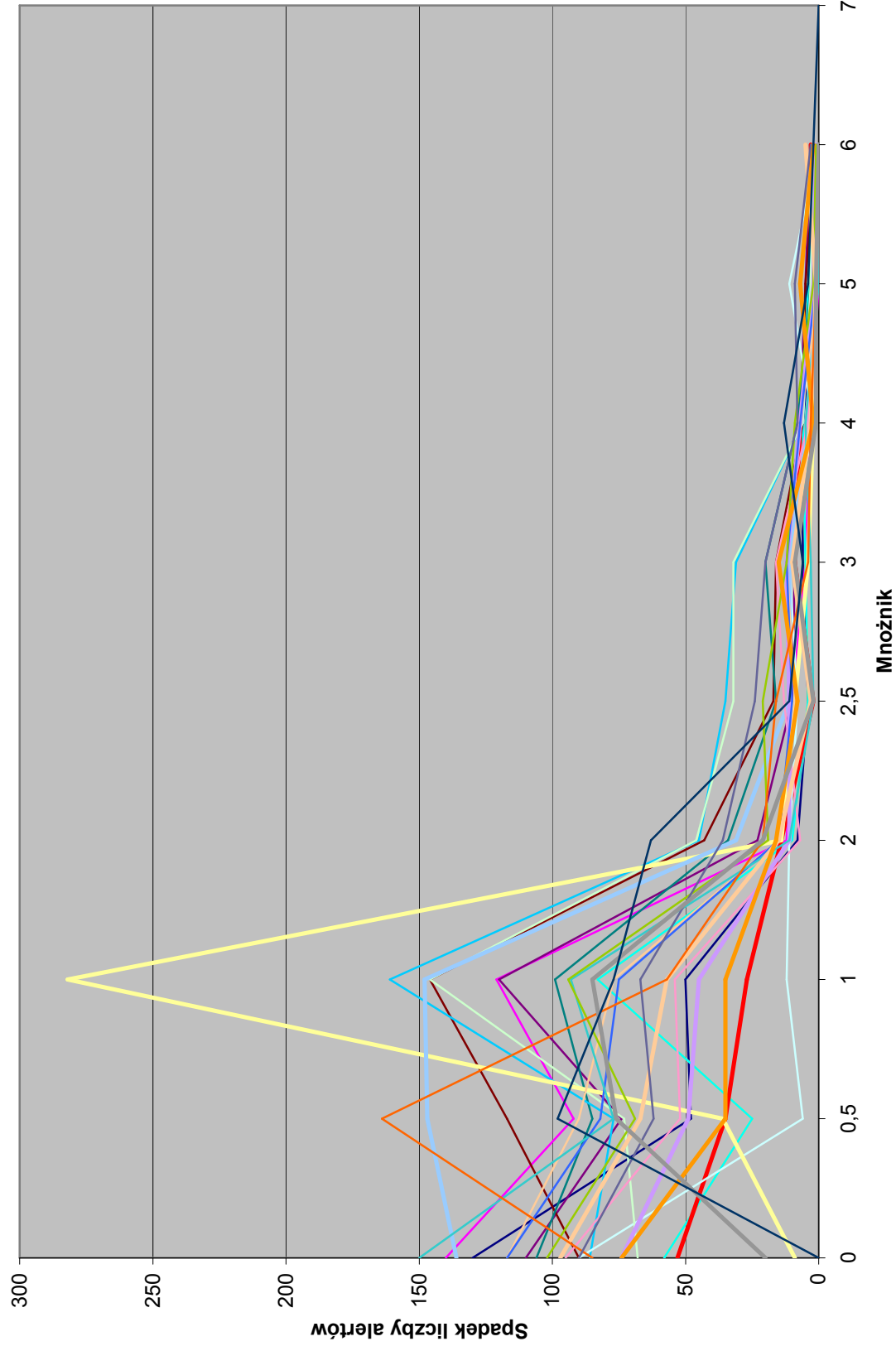
Na podstawie przeprowadzonych badań można stwierdzić, że metoda zastosowana przy tworzeniu systemu może być wykorzystywana do detekcji anomalii. Wymaga ona odpowiedniego doboru wartości mnożnika sigmy, odpowiedniego dla danej sieci i dla danego typu ruchu sieciowego.

Tabela 4: Zależność ilości alertów od mnożnika sigmy. Źródło: opracowanie własne.

| Mnożnik Sigma | TCP | TCP up | TCP down | TCP LAN | UDP | UDP up | UDP down | UDP LAN | ICMP | ICMP Up | ICMP down | ICMP LAN | SYN&ACK | WWW up | WWW down | DNS up | DNS down | TCP up | TCP down | WWW up | WWW down | UDP up | UDP down | DNS up | DNS down | Suma |
|---------------|-----|--------|----------|---------|-----|--------|----------|---------|------|---------|-----------|----------|---------|--------|----------|--------|----------|--------|----------|--------|----------|--------|----------|--------|----------|------|
| 0             | 260 | 383    | 351      | 203     | 367 | 454    | 386      | 153     | 459  | 153     | 422       | 358      | 493     | 257    | 227      | 334    | 326      | 341    | 329      | 271    | 208      | 454    | 338      | 217    | 279      | 8023 |
| 0,5           | 130 | 243    | 236      | 145     | 257 | 364    | 280      | 100     | 373  | 63      | 354       | 349      | 357     | 162    | 153      | 217    | 209      | 191    | 227      | 174    | 134      | 364    | 248      | 197    | 181      | 5708 |
| 1             | 82  | 151    | 151      | 120     | 183 | 247    | 195      | 65      | 296  | 57      | 281       | 313      | 210     | 110    | 104      | 127    | 127      | 114    | 158      | 107    | 99       | 279    | 186      | 121    | 104      | 3987 |
| 2             | 32  | 30     | 42       | 37      | 63  | 101    | 96       | 38      | 135  | 45      | 135       | 31       | 62      | 56     | 59       | 50     | 52       | 21     | 64       | 50     | 64       | 115    | 119      | 36     | 41       | 1574 |
| 2,5           | 24  | 18     | 19       | 27      | 40  | 58     | 62       | 24      | 90   | 34      | 89        | 17       | 31      | 49     | 48       | 35     | 37       | 10     | 45       | 35     | 48       | 58     | 83       | 15     | 30       | 1026 |
| 3             | 20  | 8      | 12       | 23      | 30  | 41     | 46       | 22      | 55   | 26      | 57        | 8        | 20      | 36     | 37       | 26     | 27       | 8      | 24       | 32     | 40       | 37     | 59       | 13     | 24       | 731  |
| 4             | 14  | 3      | 5        | 18      | 21  | 25     | 26       | 13      | 24   | 15      | 25        | 4        | 9       | 20     | 25       | 14     | 15       | 5      | 12       | 22     | 25       | 21     | 39       | 4      | 11       | 415  |
| 5             | 8   | 0      | 0        | 12      | 14  | 19     | 18       | 6       | 18   | 13      | 19        | 2        | 5       | 16     | 18       | 7      | 8        | 0      | 3        | 19     | 23       | 17     | 31       | 3      | 7        | 286  |
| 6             | 7   | 0      | 0        | 8       | 9   | 14     | 15       | 5       | 15   | 2       | 16        | 0        | 4       | 15     | 16       | 7      | 7        | 0      | 1        | 18     | 16       | 14     | 22       | 2      | 5        | 218  |
| 7             | 7   | 0      | 0        | 8       | 6   | 13     | 15       | 2       | 14   | 2       | 14        | 0        | 4       | 15     | 15       | 6      | 6        | 0      | 0        | 13     | 14       | 13     | 19       | 2      | 5        | 193  |



Rysunek 26: Zależność liczby alertów od mnożnika sigmy. Źródło: opracowanie własne.



Rysunek 27: Spadek liczby generowanych alertów przy zmianie mnożnika sigmy. Źródło: opracowanie własne.

## **Zakończenie. Kierunki dalszych badań.**

W ramach pracy napisany został preprocesor, który zapisuje informacje o dwudziestu pięciu różnych parametrach ruchu sieciowego i wykrywa anomalie w ich zachowaniu oraz program generujący profil (opis typowych zachowań) sieci na podstawie informacji preprocesora. Oprogramowanie to bazuje na porównywaniu aktualnie zaistniałej sytuacji do stworzonego wcześniej profilu sieci.

Przeprowadzone badanie trwało 6 tygodni i objęło 6040 pomiarów 25 parametrów dokonywanych co 10 minut (łącznie otrzymano 25 szeregów czasowych o długości 6040 każdy). Przeprowadzona została wstępna analiza statystyczna obejmująca wyrywkowe testy zgodności rozkładów empirycznych ruchu z rozkładem normalnym oraz badanie liczby alertów (zdarzeń polegających na odchyleniu wartości danego ruchu od średniej o więcej niż zadana liczba odchyleń standardowych).

Przygotowany w ramach niniejszej pracy program został uruchomiony i cały czas pracuje zbierając dane w testowanej sieci, więc długość szeregów czasowych ciągle rośnie, co pozwoli na przeprowadzenie w najbliższym czasie bardziej zaawansowanej analizy statystycznej otrzymywanych danych.

W ramach rozwijania zrealizowanego w tej pracy projektu planowane jest:

- tworzenie profili dla każdego hosta w sieci,
- analiza i rozpoznanie rozkładów statystycznych poszczególnych typów ruchu,
- implementacja analizy zależności pomiędzy poszczególnymi parametrami (np. stosunek liczby wysłanych pakietów TCP do liczby odebranych pakietów TCP),
- analiza poprawności uzyskanego wyniku przed dodaniem go do logów,
- zapis i analiza pakietów przechwyconych w momencie wykrycia anomalii.



# Bibliografia

## Książki

1. [Szmit 2005] Maciej Szmit, Marek Gusta, Mariusz Tomaszewski, „101 zabezpieczeń przed atakami w sieci komputerowej”, Helion, 2005
2. [Baker 2004] Andrew Baker, Jay Beale, Brian Caswell, Mike Poore, “Snort 2.1 Intrusion Detection Second Edition”, Syngress 2004
3. [Rehman 2003] Rafeeq Ur Rehman, “Intrusion Detecton Systems with Snort”, ISBN 0-13-140733-3
4. [Bace] Rebecca Bace, Peter Mell , „Intrusion detection systems”
5. [Endorf 2004] Carl Endorf, Eugene Schultz and Jim Mellander , „Intrusion Detection & Prevention”, ISBN 0072229543
6. [Rash 2005] Michael Rash, Angela Orebaugh, Graham Clark, Becky Pinkard, Jake Babbin, “Intrusion prevention and active response”, Syngress Publishing Inc., 2005, ISBN 1-932266-47-X
7. [Pieprzyk 2005] Josef Pieprzyk, Thomas Hardjono, Jennifer Seberry “Teoria bezpieczeństwa systemów komputerowych”, Helion, 2005, ISBN: 83-7361-678-0
8. [Laing 2000] Brian Laing , “How To Guide – Implementing a Network Based Intrusion Detect”, 2000

## Artykuły

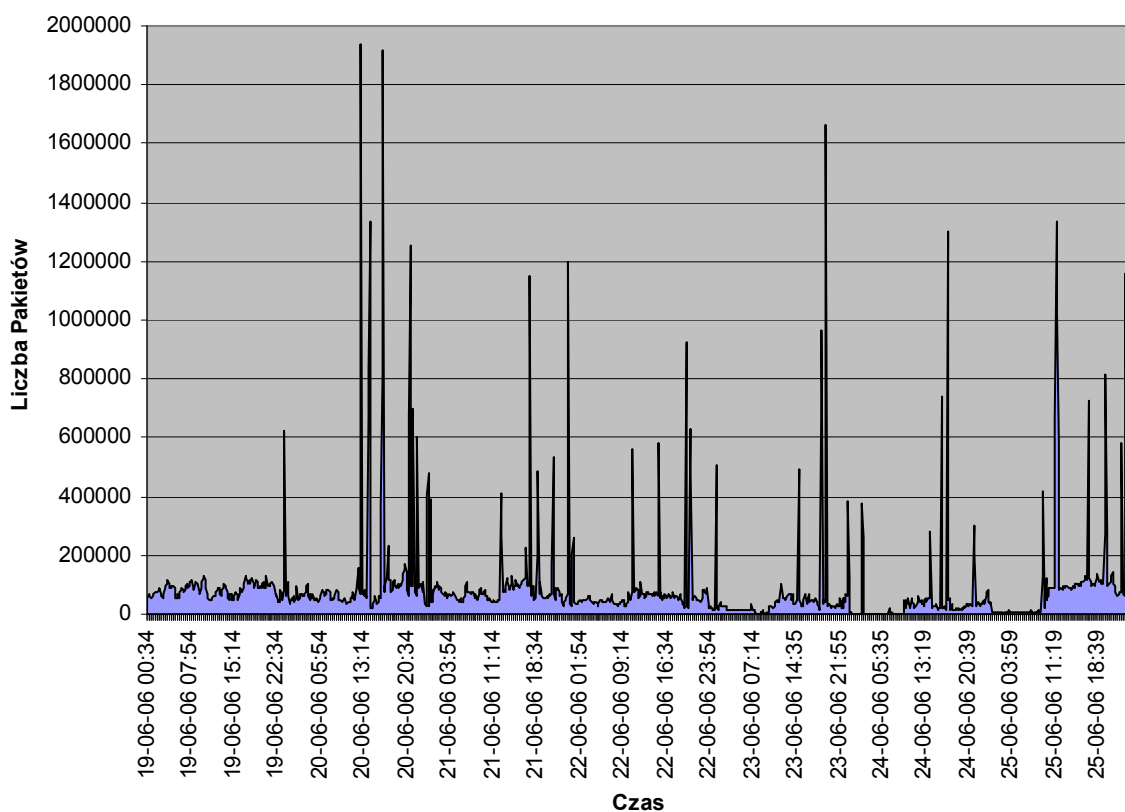
1. [Axent 1999] Axent, „Everything You Need To Know About Intrusion Detection”, Axent Technologies 1999
2. [Packer 2001] Ryon Packer, “Protecting the Network: NIDS: the logical first step in intrusion detection deployment”, Network Security Volume: 2001, Issue: 12, December 1, 2001
3. [Dorosz 2/2002] Piotr Dorosz, Przemysław Kazienko, “IDS – systemy wykrywania włamań- cz. II”, „IT FAQ – Information Technology FAQ” z 03.12.2002

## Adresy internetowe

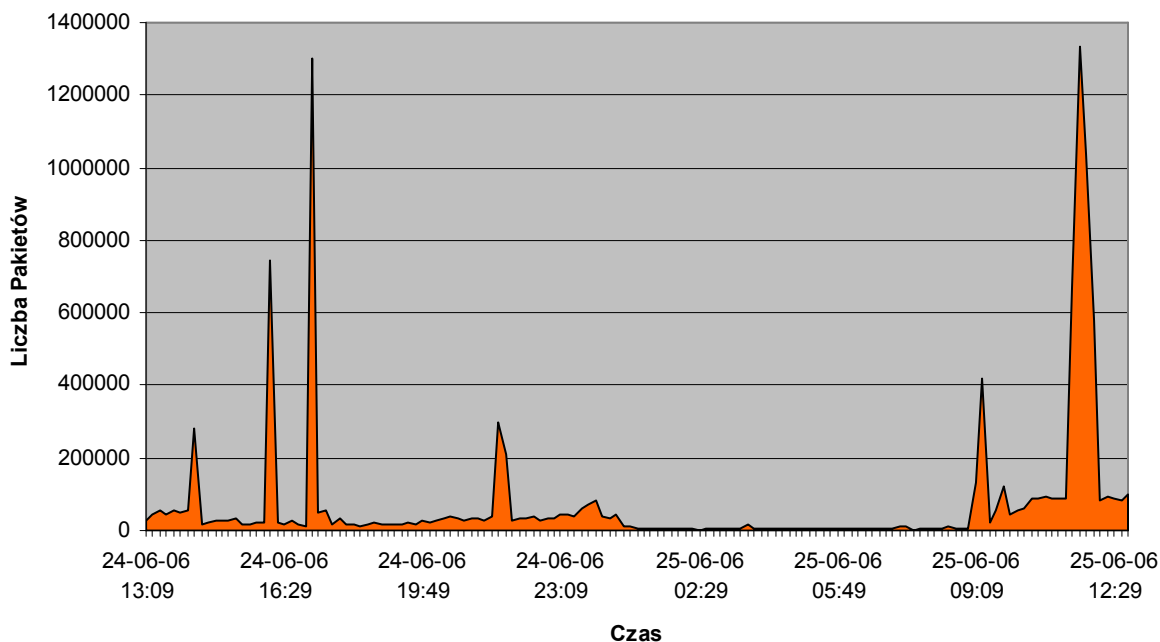
1. [I1][http://www.it-faq.pl/EditModule.aspx?tabid=397&mid=869&def=Cs\\_ITSCS\\_CMS\\_Articles\\_View&ArticleID=1334](http://www.it-faq.pl/EditModule.aspx?tabid=397&mid=869&def=Cs_ITSCS_CMS_Articles_View&ArticleID=1334) – problem fałszywych alarmów, (sprawdzony 7.06.2006)
2. [I2][http://www.nss.co.uk/WhitePapers/intrusion\\_prevention\\_systems.htm](http://www.nss.co.uk/WhitePapers/intrusion_prevention_systems.htm)
3. [I3]<http://www.securityfocus.com/infocus/1670> - opis systemów IDS, (sprawdzony 10.09.2006)

4. [I14]<http://hogwash.sourceforge.net/oldindex.html> - hogwash, (sprawdzony 10.07.2006)
5. [I15]<http://www.shmoo.com/~bmc/presentations/2005/linuxworldexpo/linuxworldexpo.ppt> – prezentacja na temat programu Snort, (sprawdzony 10.09.2006)
6. [I16] [www.netoptics.com/products/pdf/Taps-and-IDSs.pdf](http://www.netoptics.com/products/pdf/Taps-and-IDSs.pdf) - opis TAP, (sprawdzony 10.09.2006)
7. [I17] [http://www.snort.org/docs/snort\\_htmanuals/htmanual\\_2.4/rc1/](http://www.snort.org/docs/snort_htmanuals/htmanual_2.4/rc1/) - instrukcja obsługi programu Snort, (sprawdzony 10.09.2006)
8. [I18][www.tcpdump.org](http://www.tcpdump.org) - strona projektu tcpdump (sprawdzony 10.09.2006)
9. [I19][http://www.linuxia.de/minivend/mvdocs3/docindex/04.04.UNIX\\_domain\\_sockets.html](http://www.linuxia.de/minivend/mvdocs3/docindex/04.04.UNIX_domain_sockets.html) - opis UNIX-domain sockets, (sprawdzony 17.07.2006)
10. [I10]<http://en.wikipedia.org/wiki/GPL> - opis licencji GPL, (sprawdzony 10.09.2006)
11. [I11]<http://www.sans.org/resources/idfaq/honeypot3.php> – opis systemów honeypot, (sprawdzony 10.08.2006)
12. [I12]<http://www.newsforge.com/article.pl?sid=04/09/24/1734245> – opis systemów honeypot, (sprawdzony 10.09.2006)
13. [I13]<http://secureideas.sourceforge.net/>) – strona projektu BASE, (sprawdzony 11.08.2006)
14. [I14]<http://acidlab.sourceforge.net/> - strona projektu ACID, (sprawdzony 11.08.2006)
15. [I15]<http://www.pckurier.pl/archiwum/art0.asp?ID=5455> – artykuł na temat wykrywania podsłuchu w sieci, (sprawdzony 10.09.2006)
16. [I16]<http://www.honeyd.org/> - strona domowa projektu Honeyd, (sprawdzony 10.09.2006)
17. [I17]<http://www.symantec.com/region/pl/product/DecoyServer.html> – strona domowa Symantec Decoy Server, (sprawdzone 3.08.2009)
18. [I18]<http://www.snort.org> – strona domowa projektu Snort, (sprawdzony 10.09.2006)
19. [I19][http://pl.wikipedia.org/wiki/Odchylenie\\_standardowe](http://pl.wikipedia.org/wiki/Odchylenie_standardowe) – odchylenie standardowe, (sprawdzony 10.09.2006)
20. [I20][http://afrodita.unicauca.edu.co/~cbendon/snort/spp\\_kickstart.html](http://afrodita.unicauca.edu.co/~cbendon/snort/spp_kickstart.html) – opis rozwoju preprocesorów w programie Snort, (sprawdzony 10.09.2006)
21. [I21]<http://www.anomalydetection.prv.pl> – strona domowa stworzonego systemu

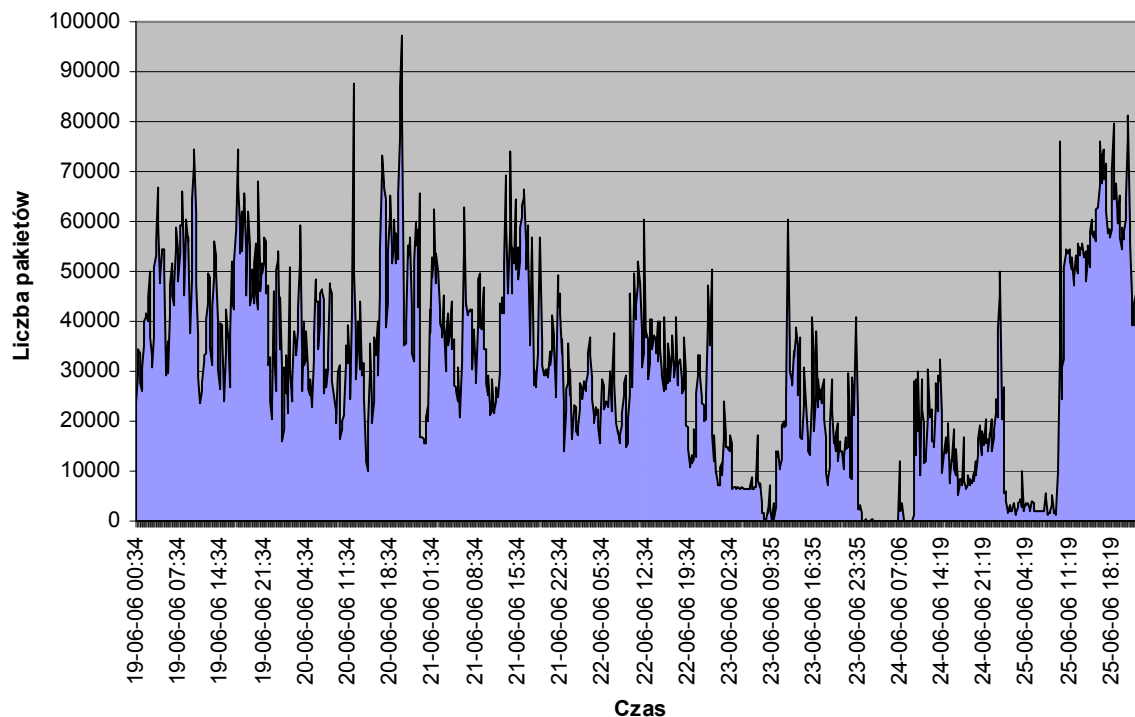
## Załącznik 1: Wykresy przedstawiające wyniki pomiarów.



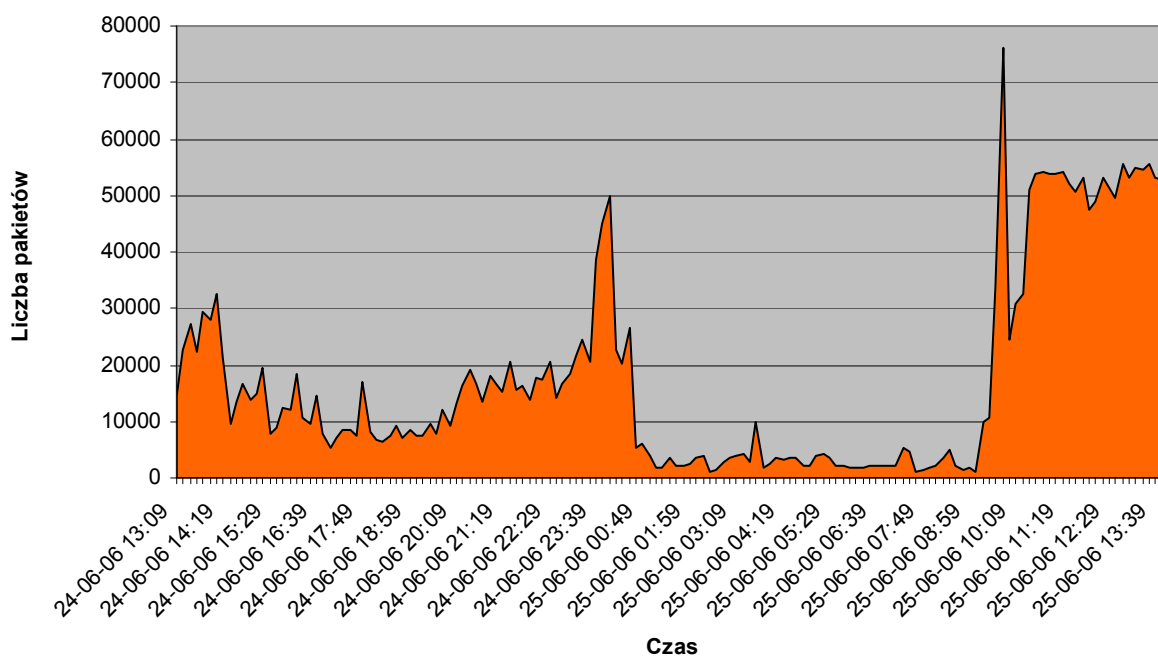
Rysunek 28: Statystyka pakietów TCP (przebieg tygodniowy). Źródło: opracowanie własne.



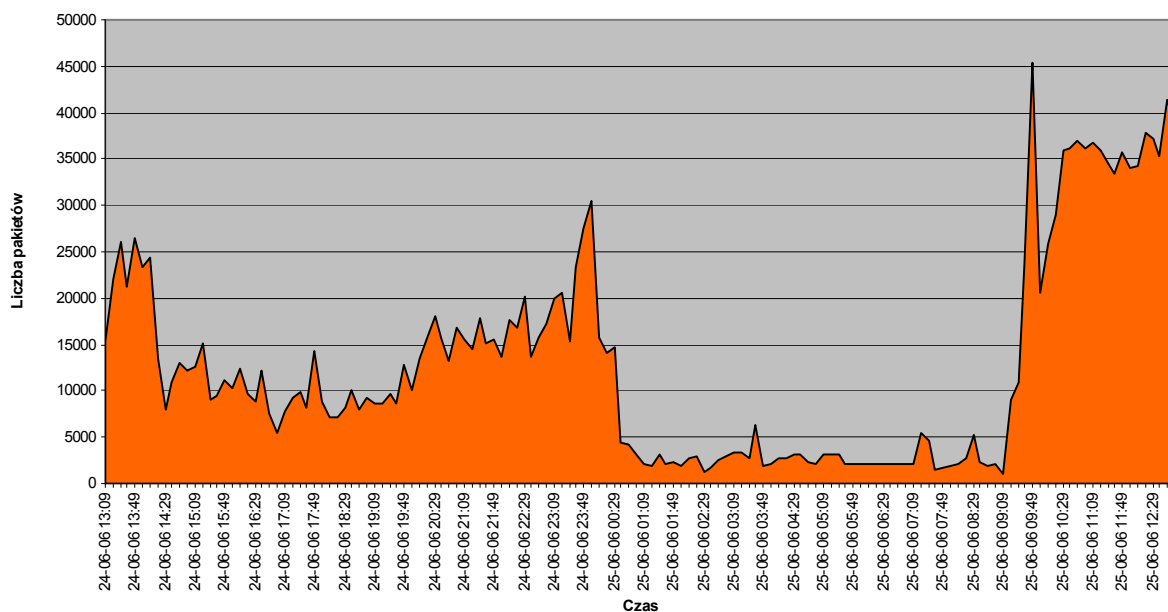
Rysunek 29: Statystyka pakietów TCP (przebieg dobowy). Źródło: opracowanie własne.



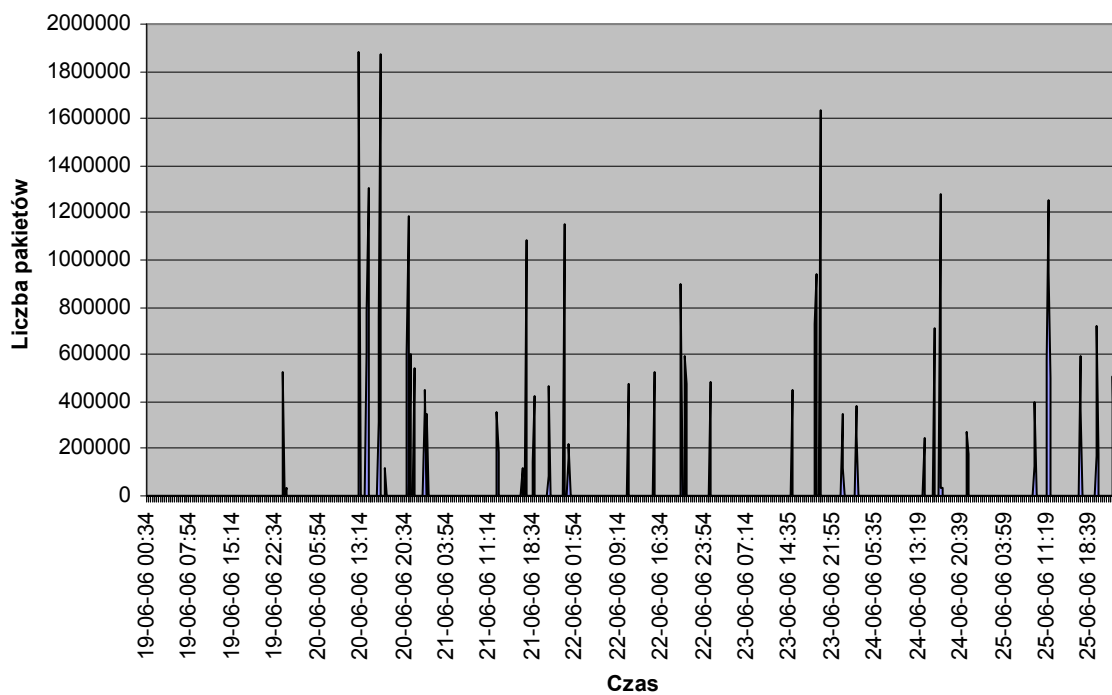
Rysunek 30: Statystyka odebranych pakietów TCP (przebieg tygodniowy). Źródło: opracowanie własne.



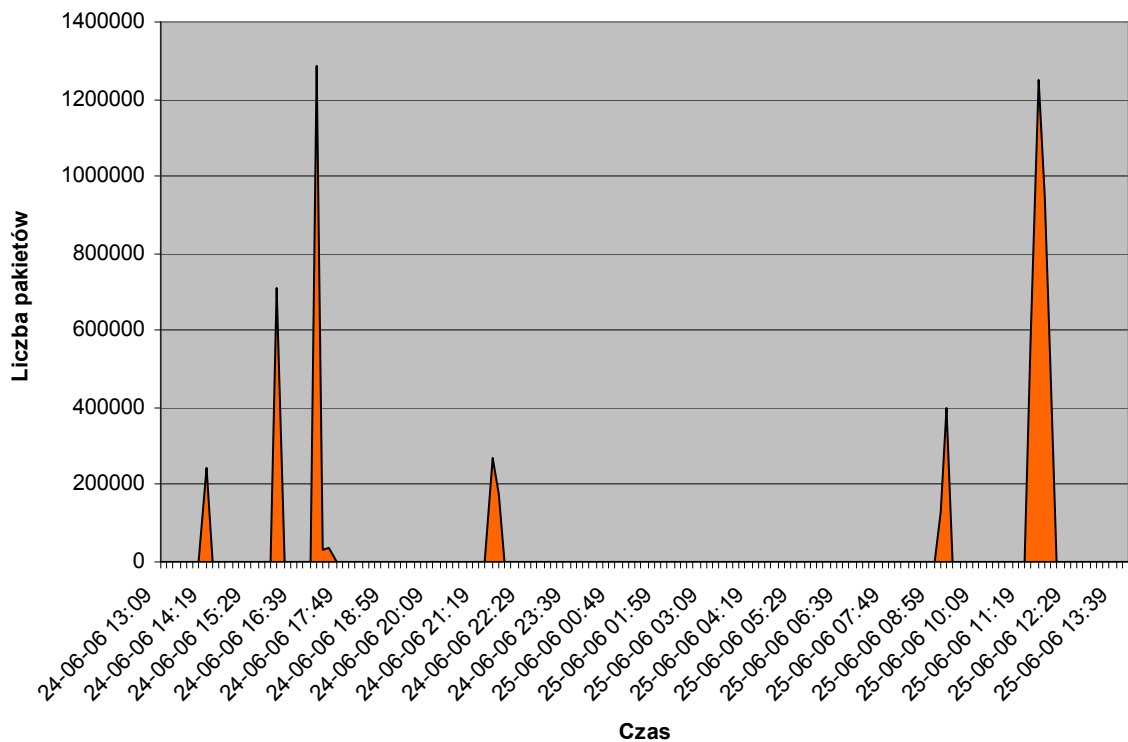
Rysunek 31: Statystyka odebranych pakietów TCP (przebieg dobowy). Źródło: opracowanie własne.



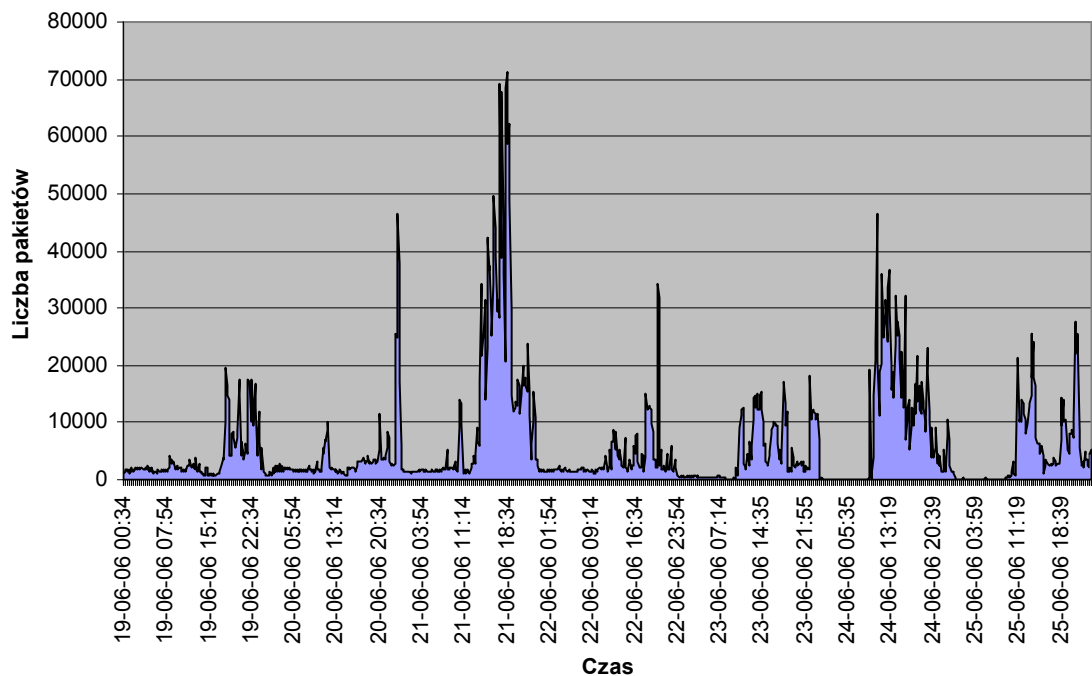
Rysunek 32: Statystyka wysłanych pakietów TCP (przebieg dobowy). Źródło: opracowanie własne.



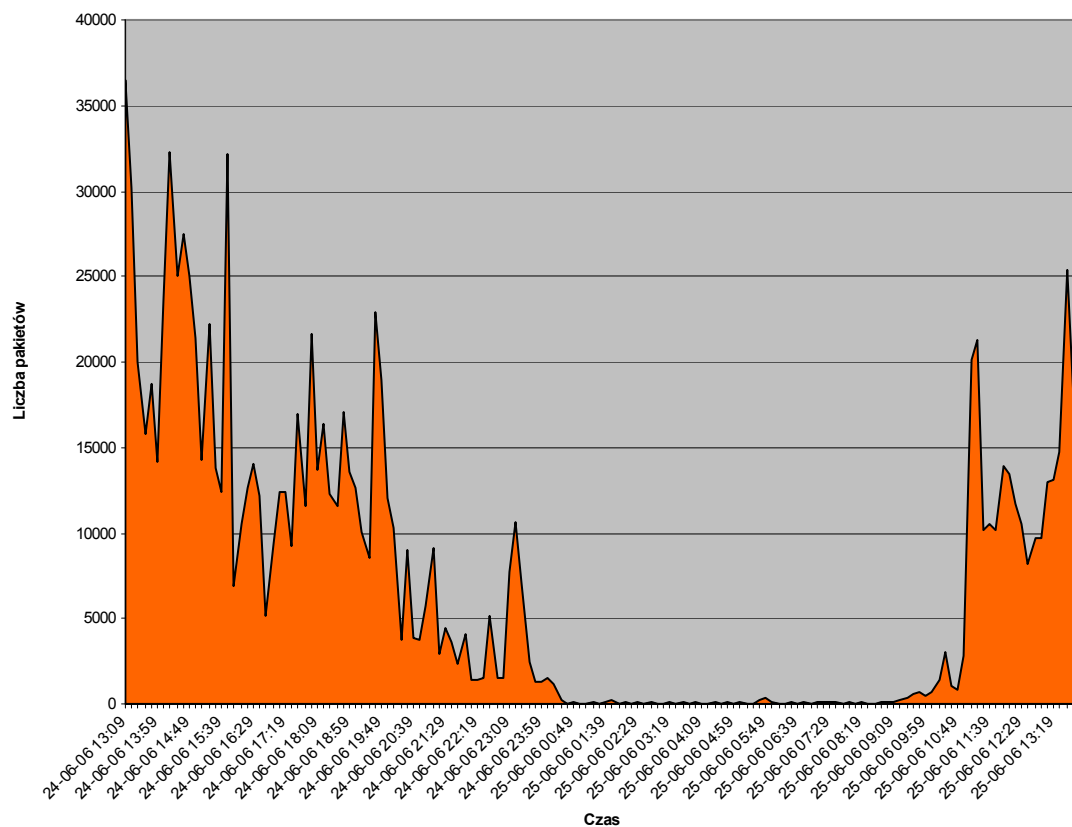
Rysunek 33: Statystyka pakietów TCP wewnątrz sieci LAN (przebieg tygodniowy). Źródło: opracowanie własne.



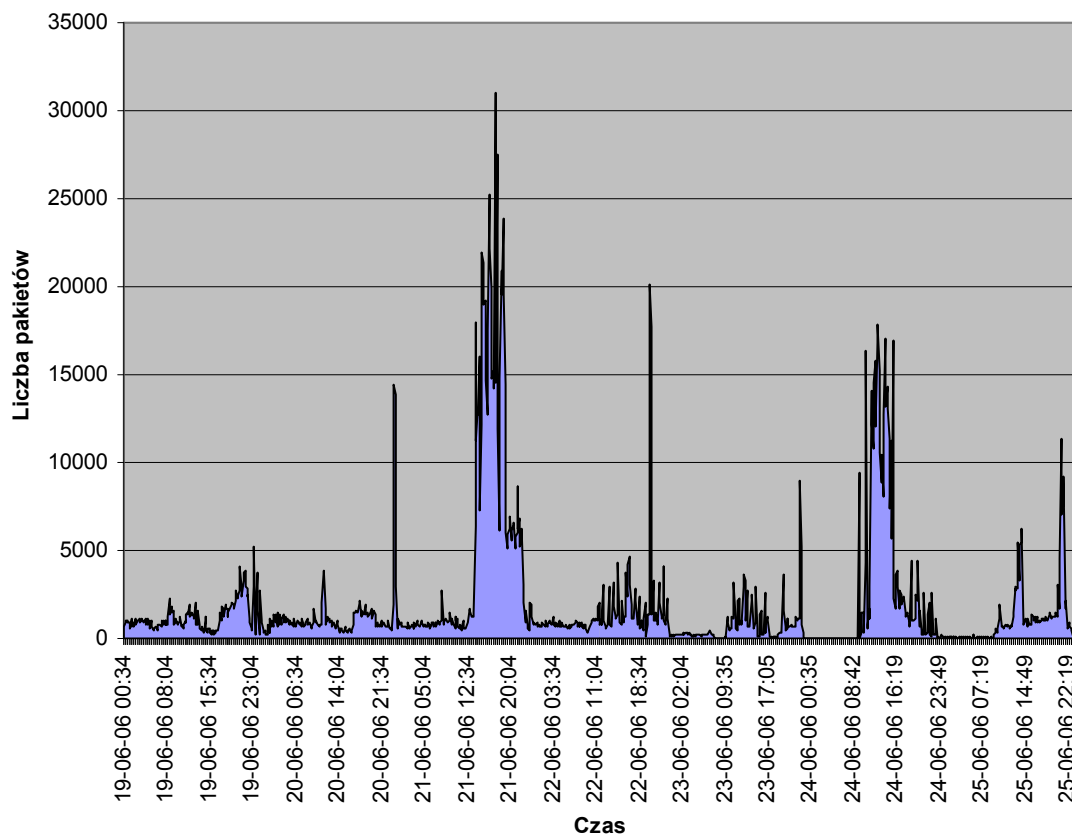
Rysunek 34: Statystyka pakietów TCP wewnątrz sieci LAN (przebieg dobowy). Źródło: opracowanie własne.



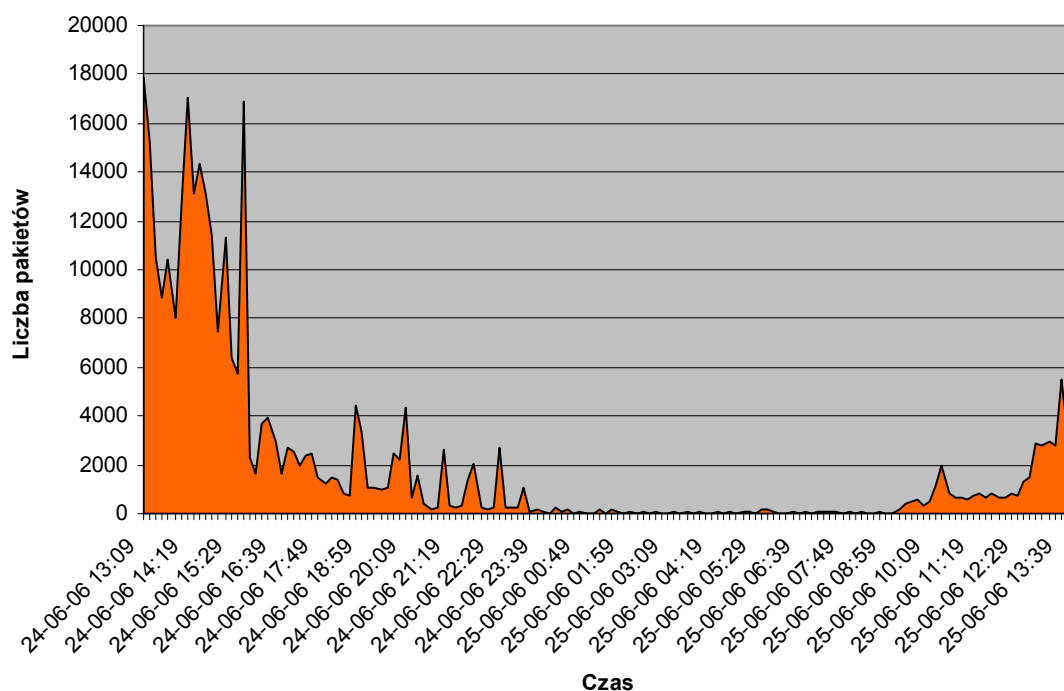
Rysunek 35: Statystyka pakietów UDP (przebieg tygodniowy). Źródło: opracowanie własne.



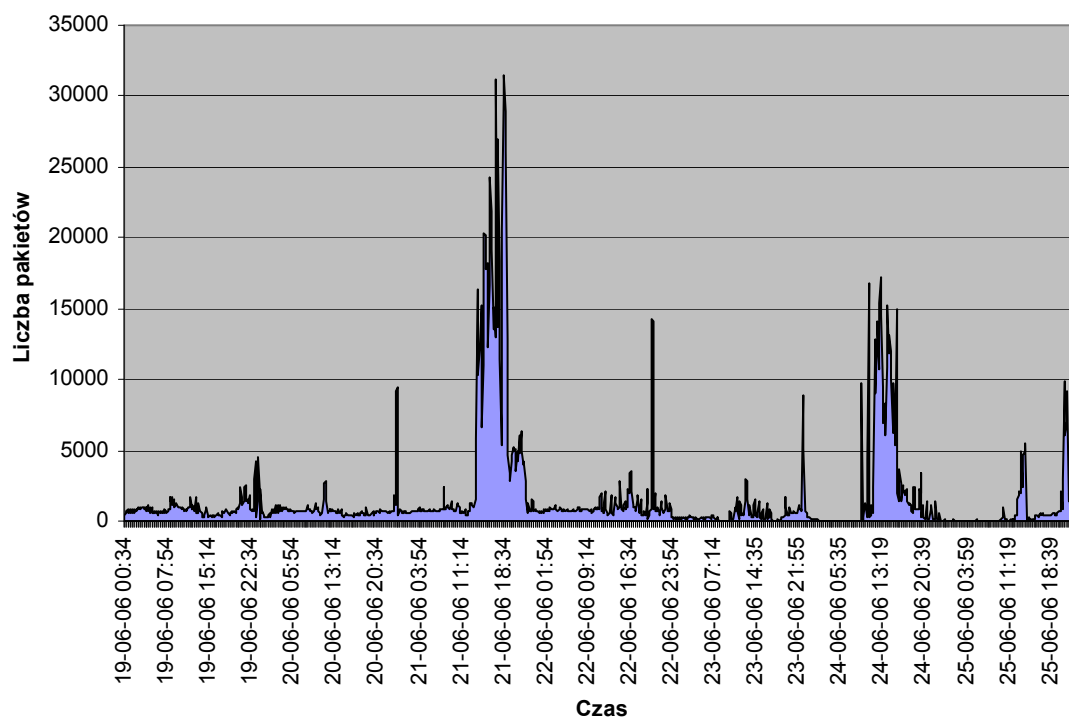
**Rysunek 36: Statystyka pakietów UDP (przebieg dobowy). Źródło: opracowanie własne.**



Rysunek 37: Statystyka wysłanych pakietów UDP (przebieg tygodniowy). Źródło: opracowanie własne.

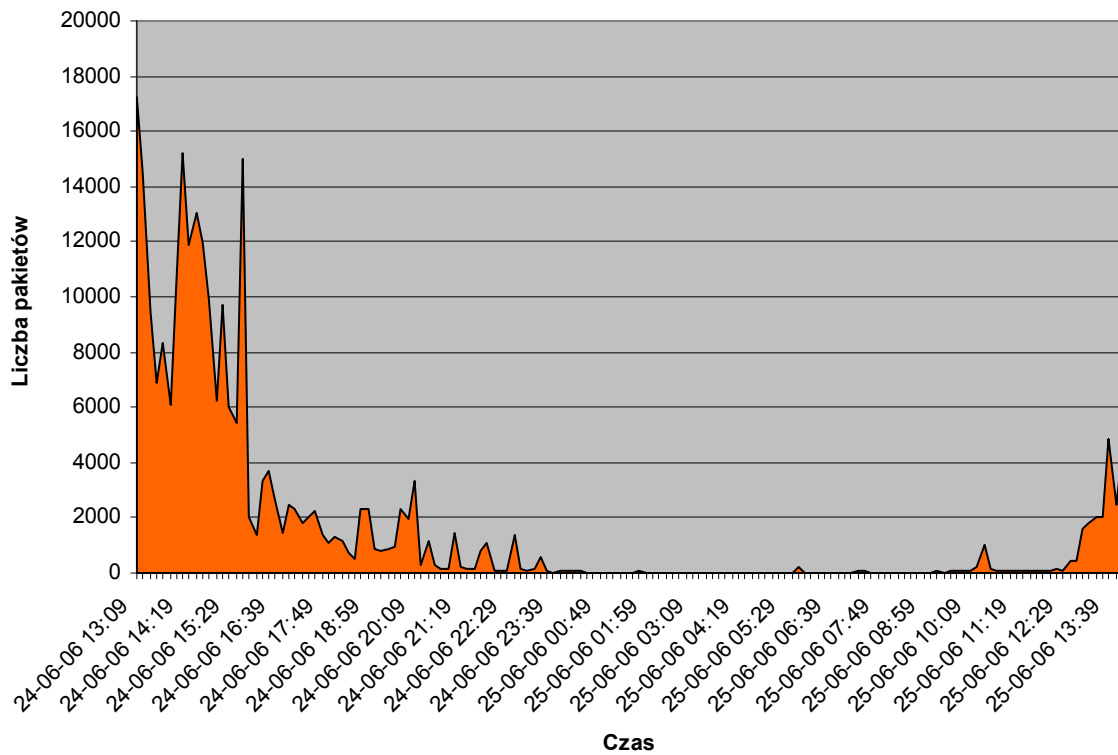


Rysunek 38: Statystyka wysłanych pakietów UDP (przebieg dobowy). Źródło: opracowanie własne.

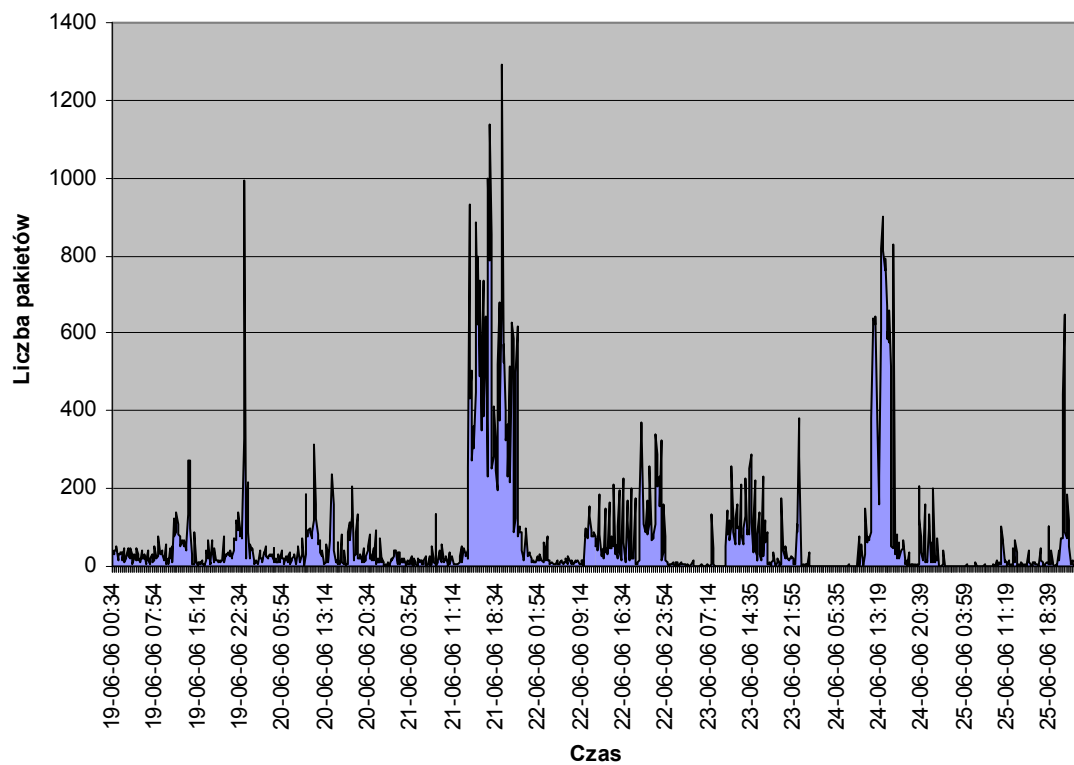


Rysunek 39: Statystyka odebranych pakietów UDP (przebieg tygodniowy). Źródło: opracowanie własne.

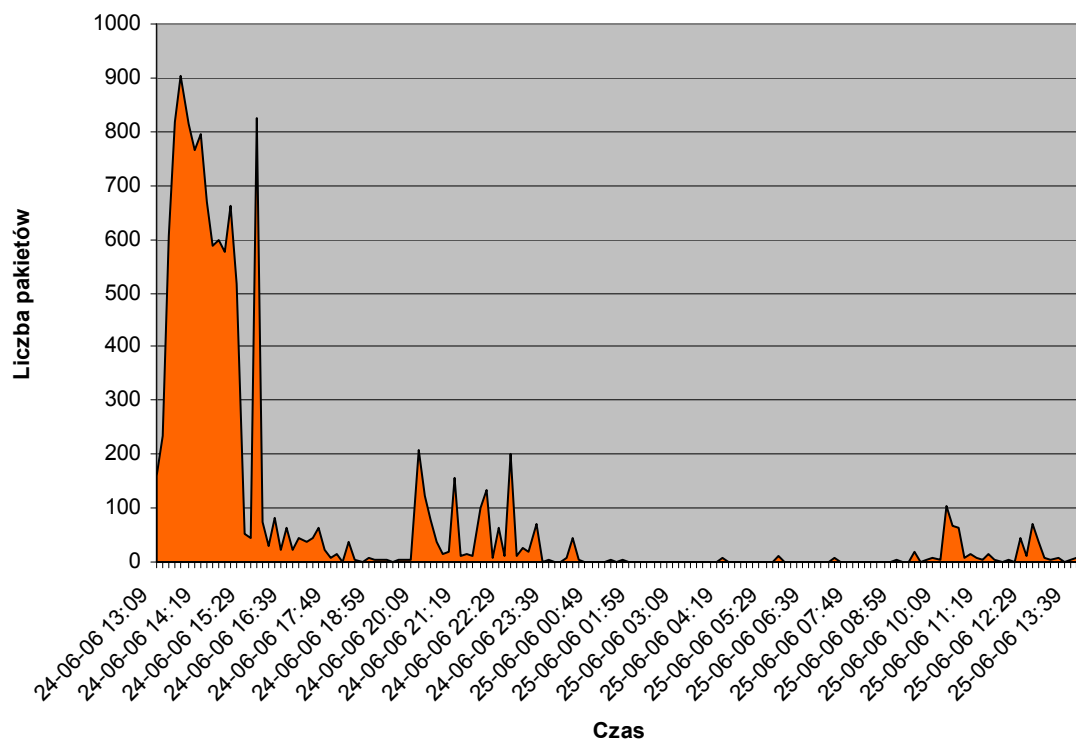




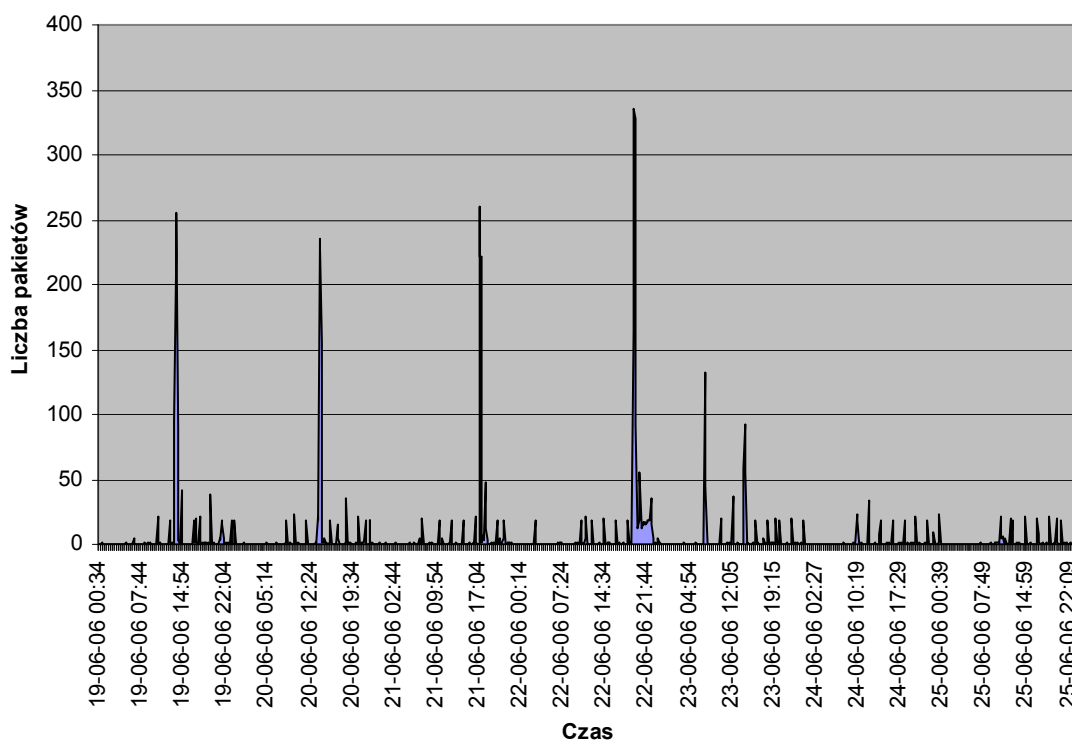
Rysunek 40: Statystyka odebranych pakietów UDP (przebieg dobowy). Źródło: opracowanie własne.



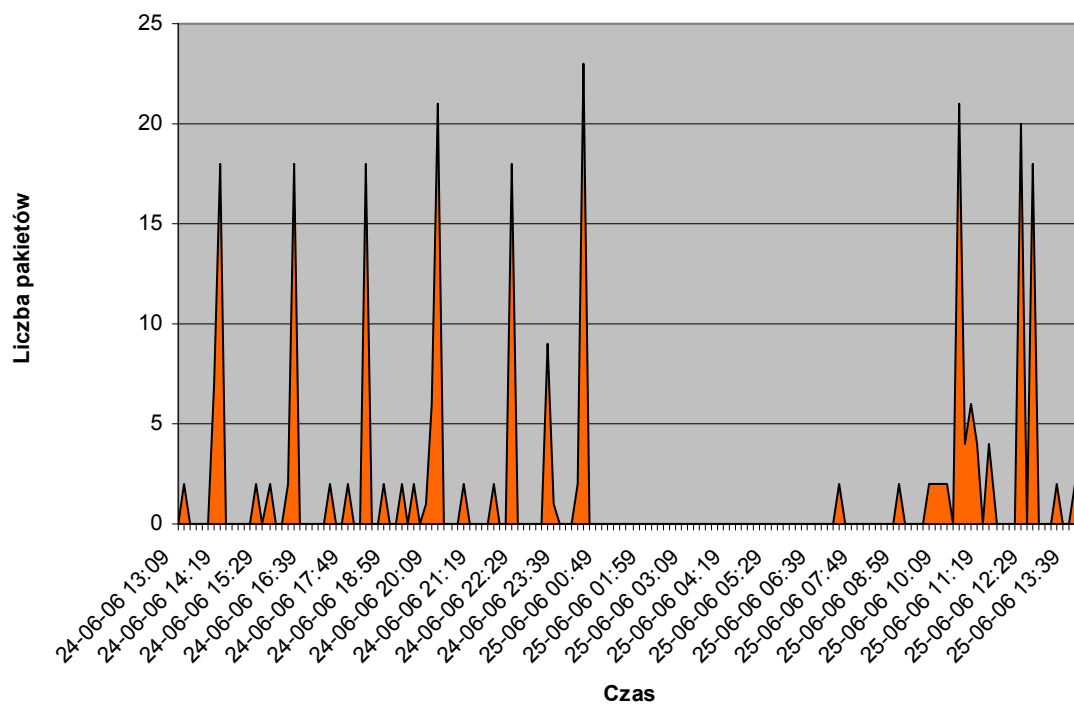
Rysunek 41: Statystyka pakietów ICMP (przebieg tygodniowy). Źródło: opracowanie własne.



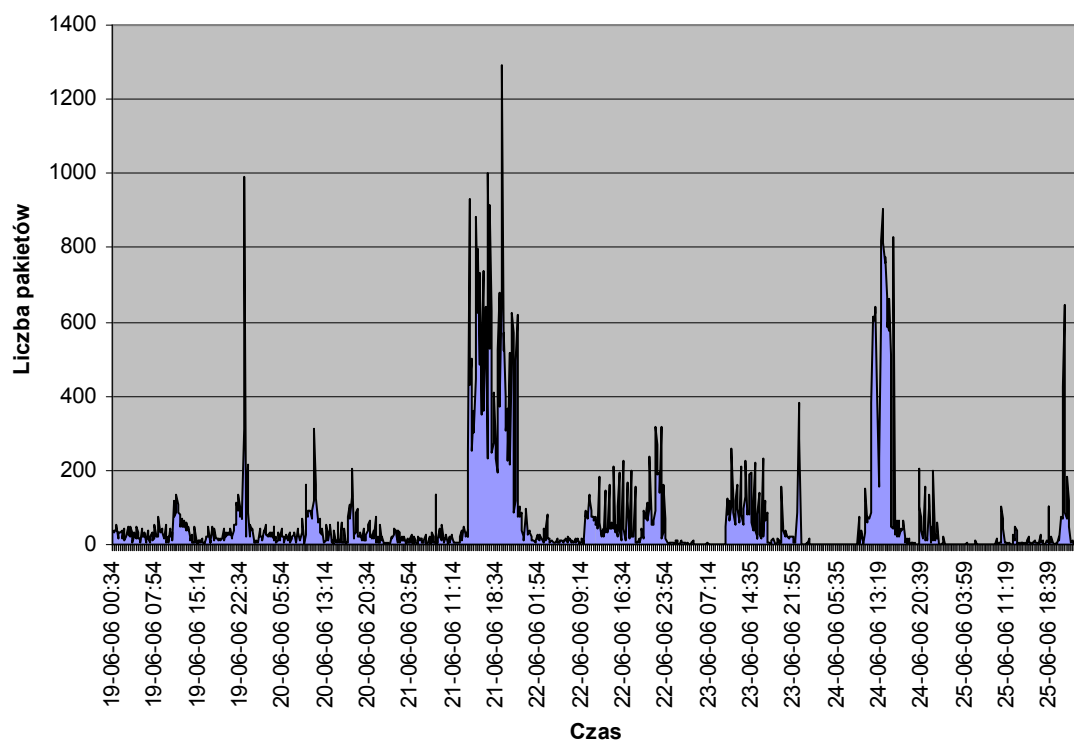
Rysunek 42: Statystyka pakietów ICMP (przebieg dobowy). Źródło: opracowanie własne.



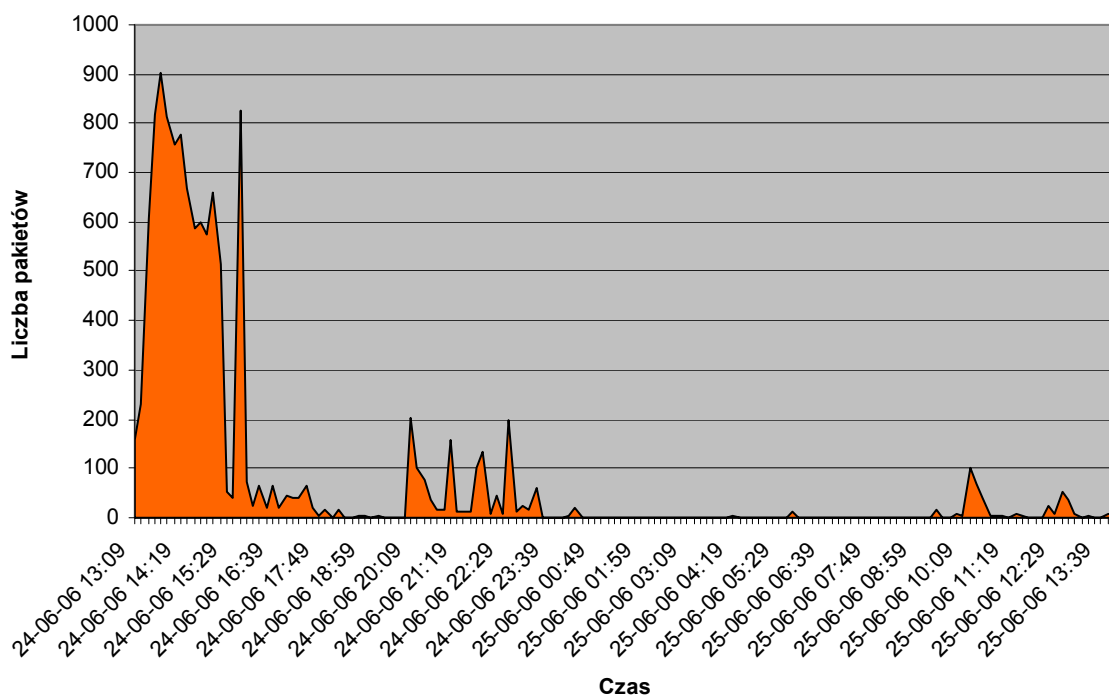
Rysunek 43: Statystyka wysłanych pakietów ICMP (przebieg tygodniowy). Źródło: opracowanie własne.



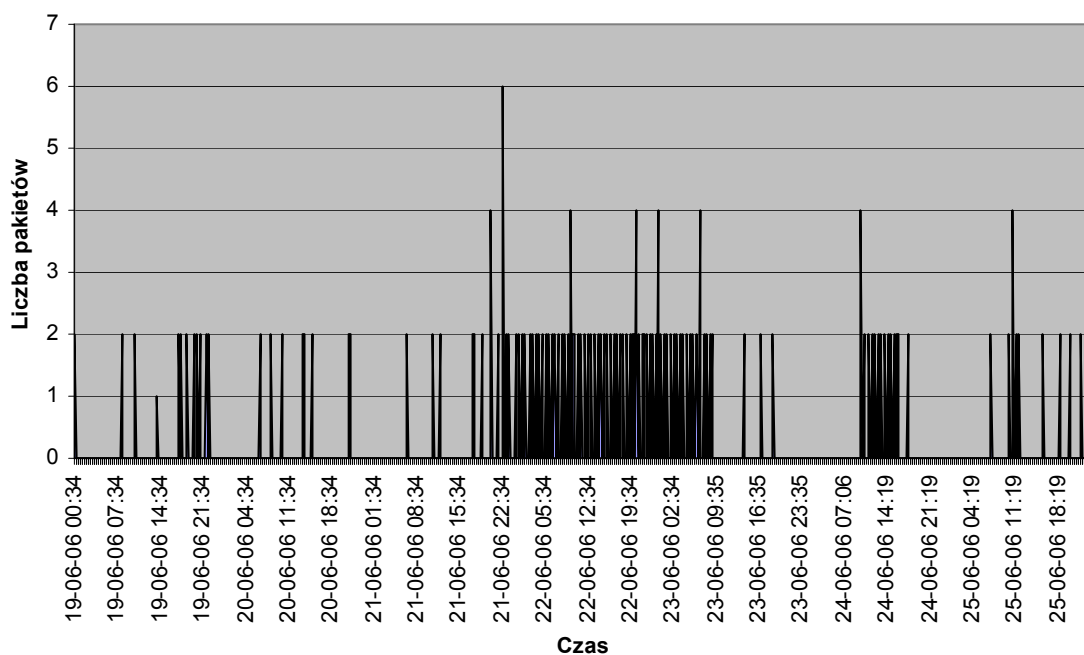
Rysunek 44: Statystyka wysłanych pakietów ICMP (przebieg dobowy). Źródło: opracowanie własne.



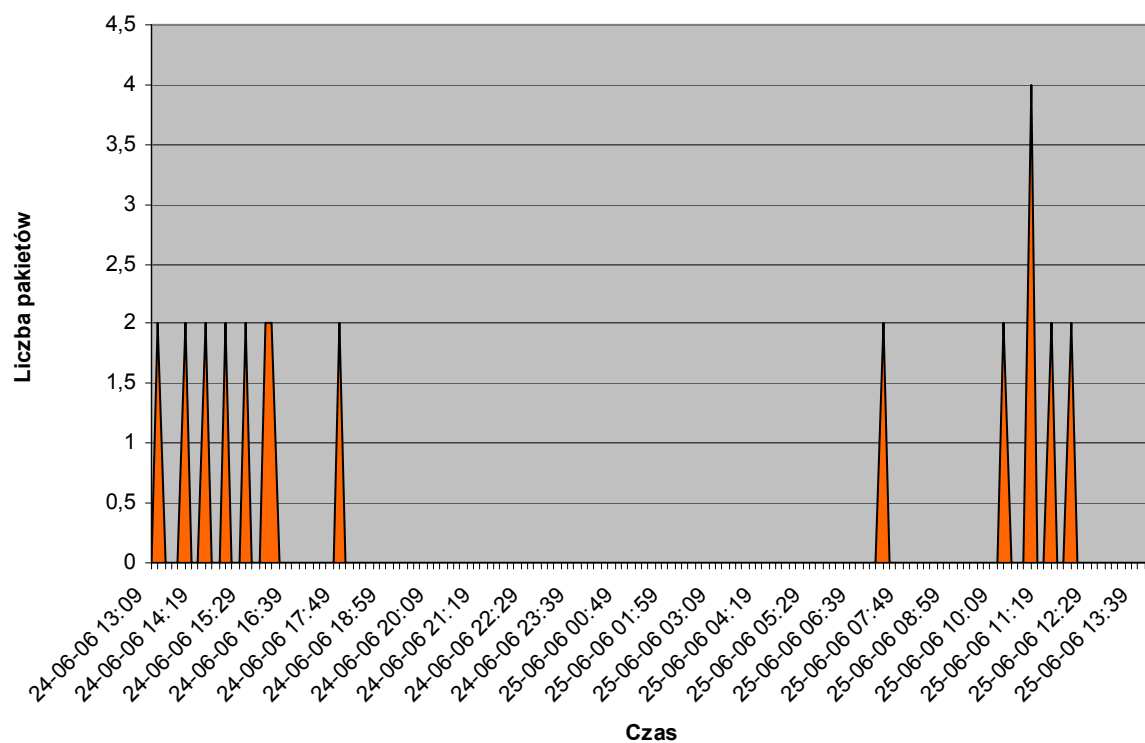
Rysunek 45: Statystyka odebranych pakietów ICMP (przebieg tygodniowy). Źródło: opracowanie własne.



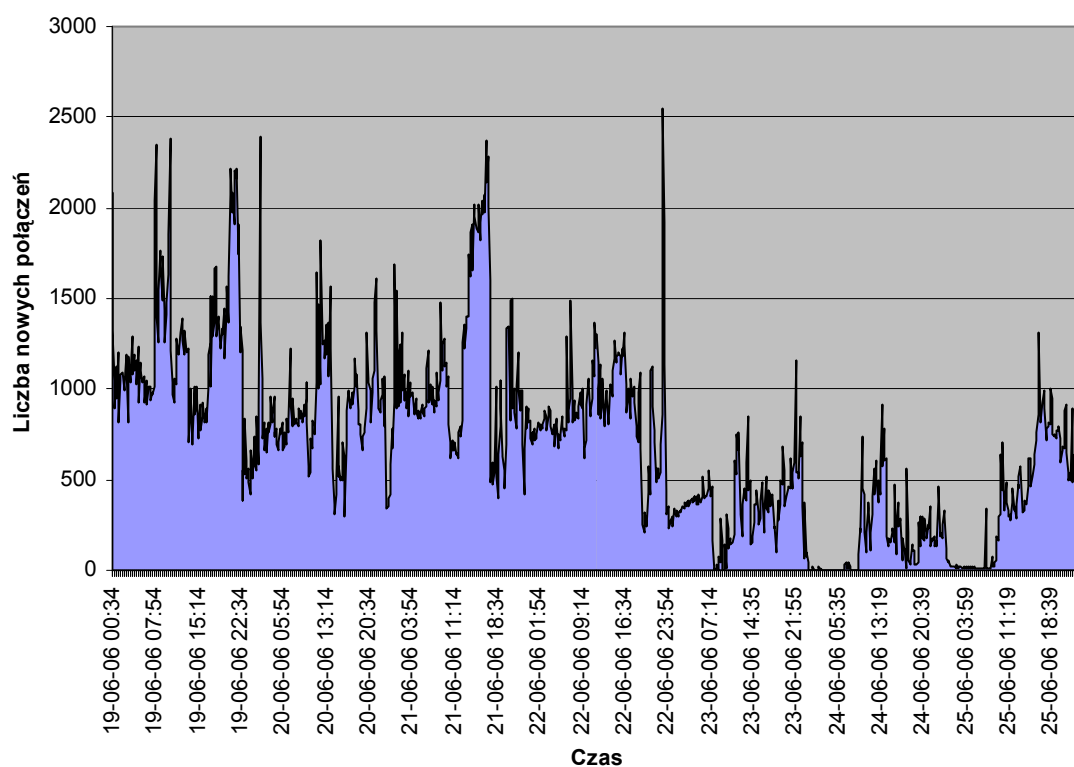
Rysunek 46: Statystyka odebranych pakietów ICMP (przebieg dobowy). Źródło: opracowanie własne.



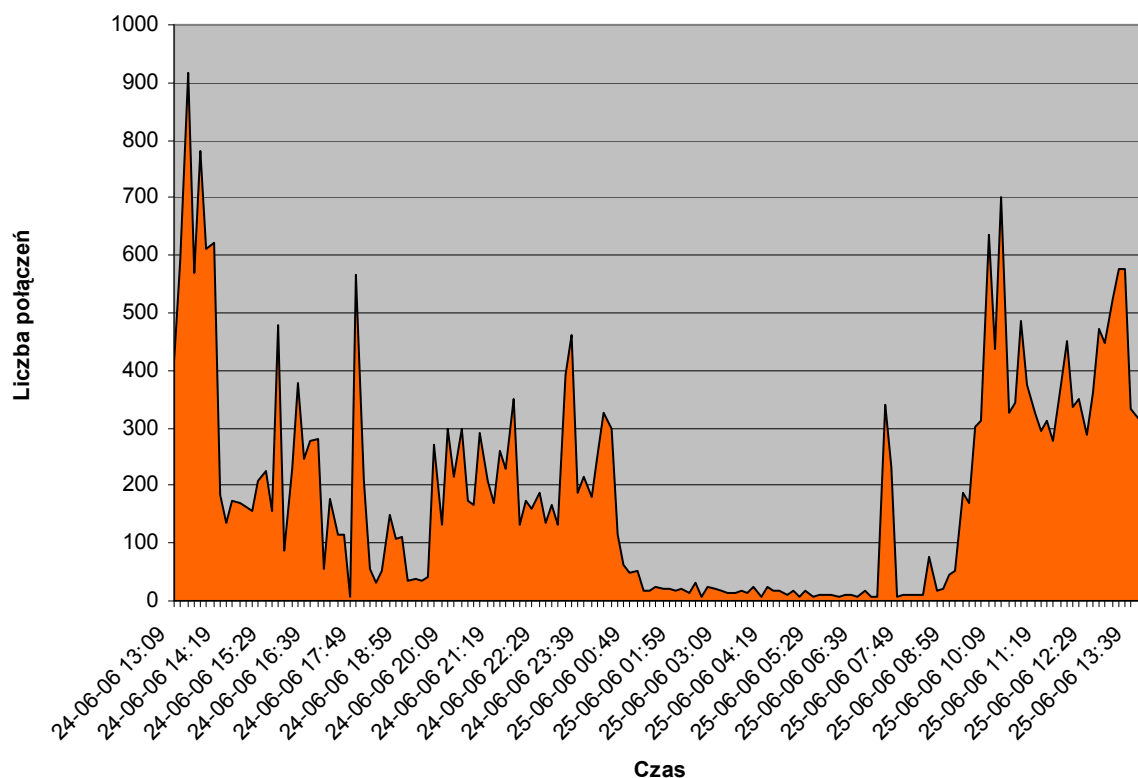
Rysunek 47: Statystyka pakietów ICMP wewnątrz sieci LAN (przebieg tygodniowy). Źródło: opracowanie własne.



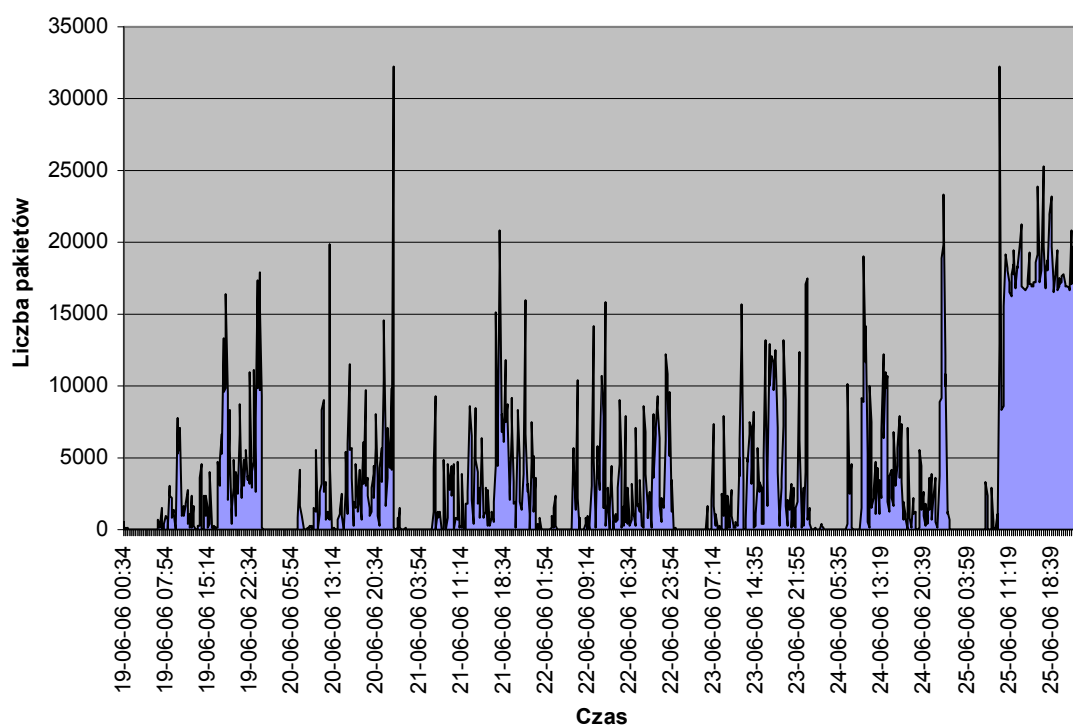
Rysunek 48: Statystyka pakietów ICMP wewnątrz sieci LAN (przebieg dobowy). Źródło: opracowanie własne.



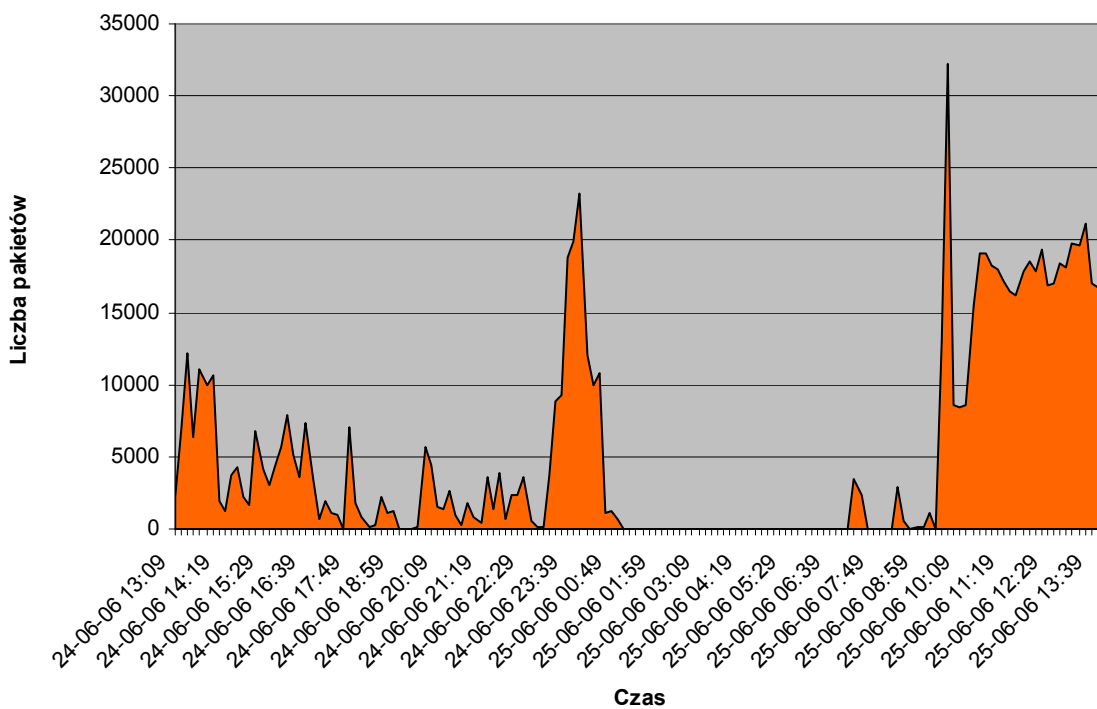
Rysunek 49: Liczba nowych połączeń (przebieg tygodniowy). Źródło: opracowanie własne.



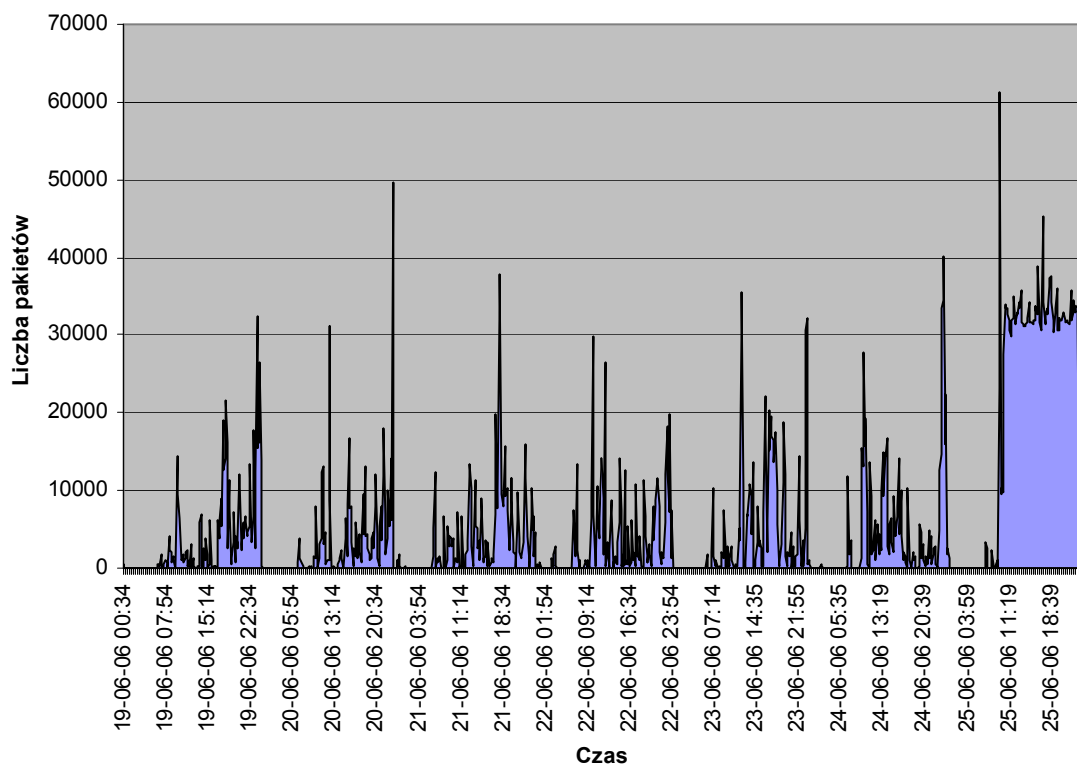
Rysunek 50: Liczba nowych połączeń (przebieg dobowy). Źródło: opracowanie własne.



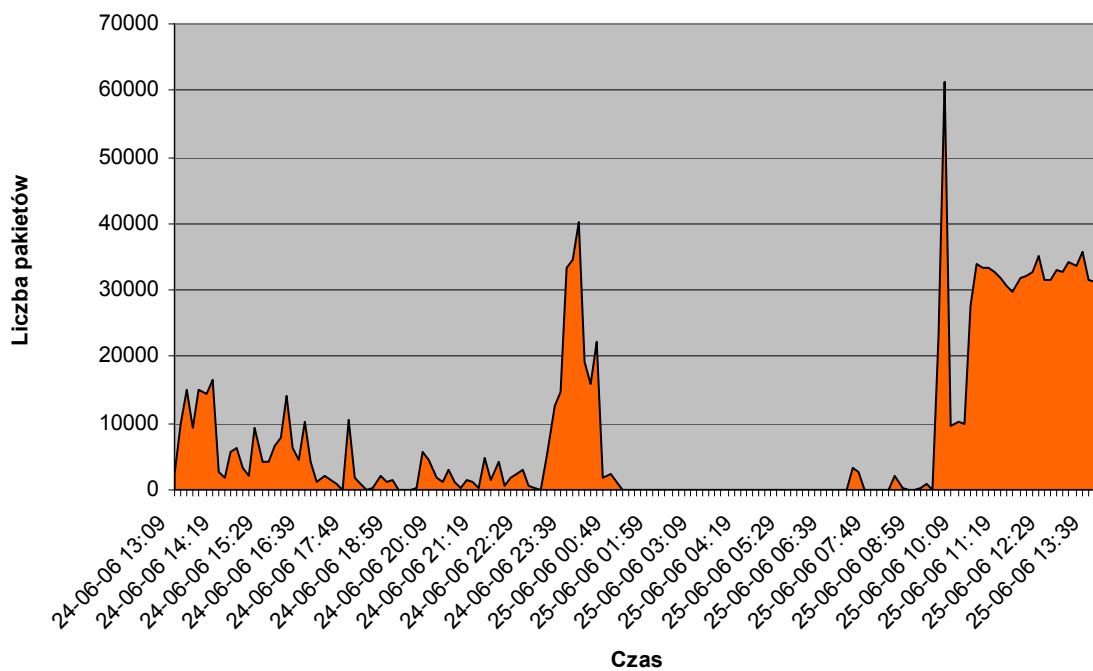
Rysunek 51: Statystyka wysłanych pakietów TCP port 80 (WWW) (przebieg tygodniowy). Źródło: opracowanie własne.



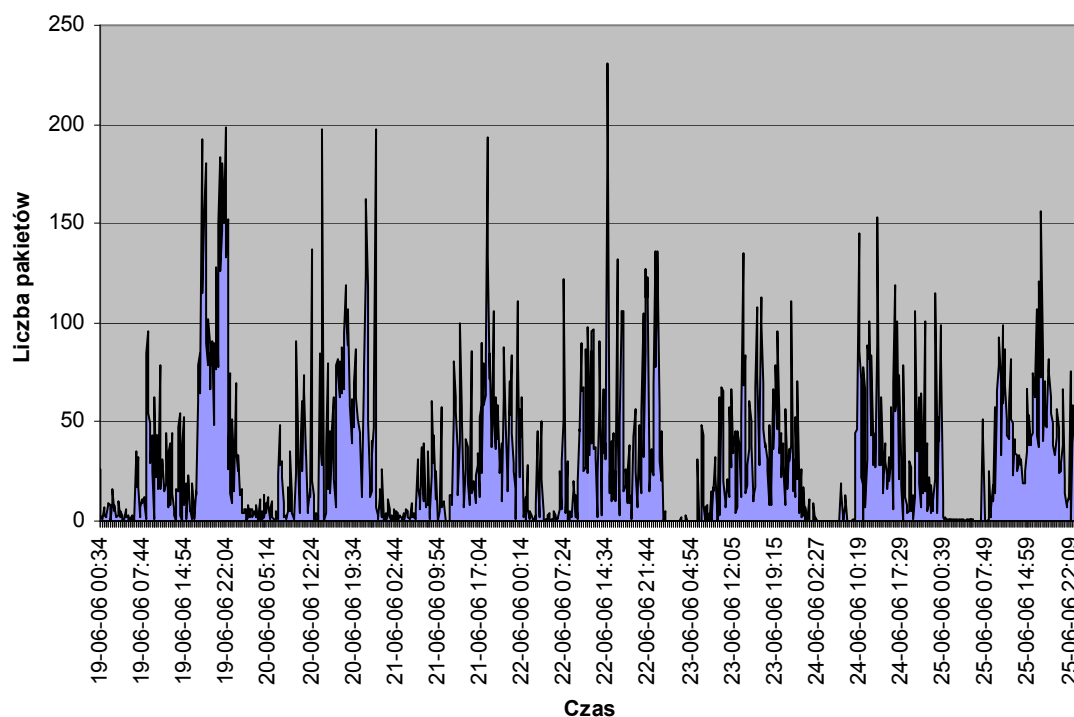
**Rysunek 52: Statystyka wysłanych pakietów TCP port 80 (WWW) (przebieg dobowy). Źródło: opracowanie własne.**



**Rysunek 53: Statystyka odebranych pakietów TCP port 80 (WWW) (przebieg tygodniowy). Źródło: opracowanie własne.**

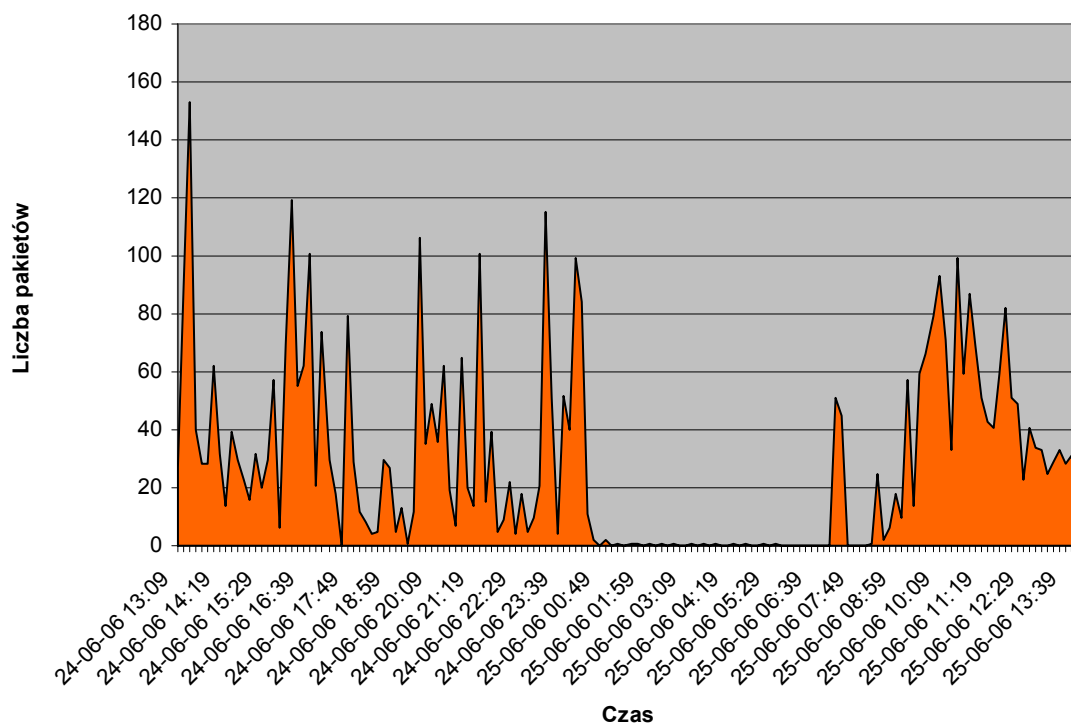


**Rysunek 54: Statystyka odebranych pakietów TCP port 80 (WWW) (przebieg dobowy). Źródło: opracowanie własne.**

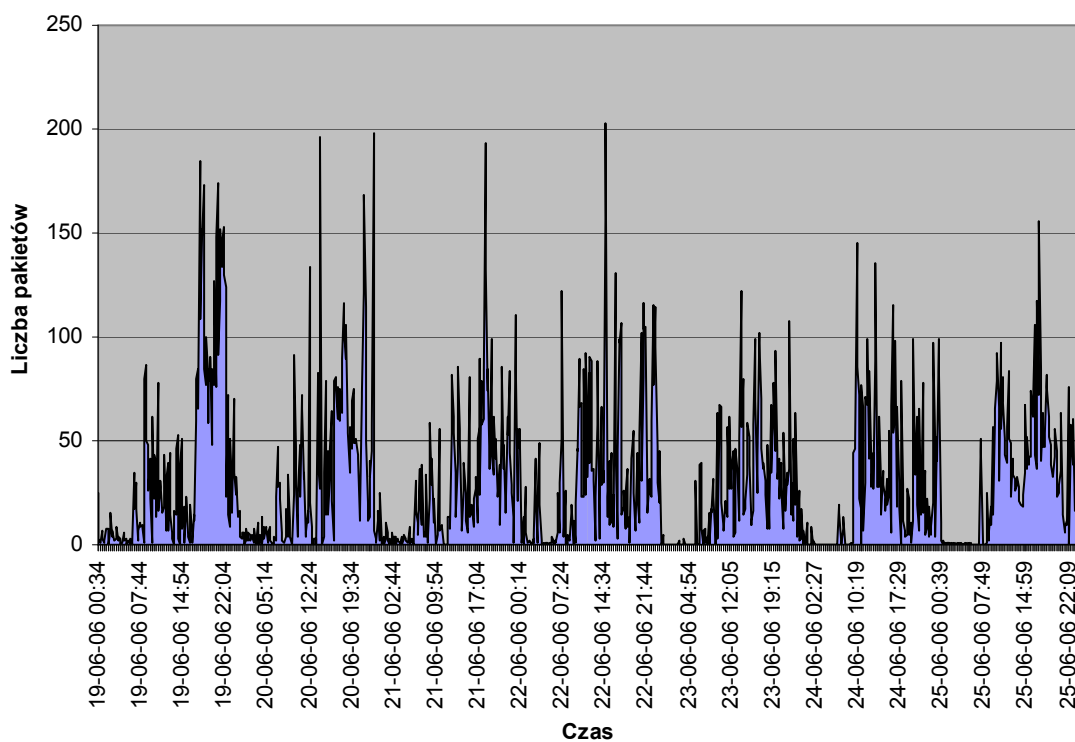


**Rysunek 55: Statystyka wysłanych pakietów UDP port 53 (DNS) (przebieg tygodniowy). Źródło: opracowanie własne.**

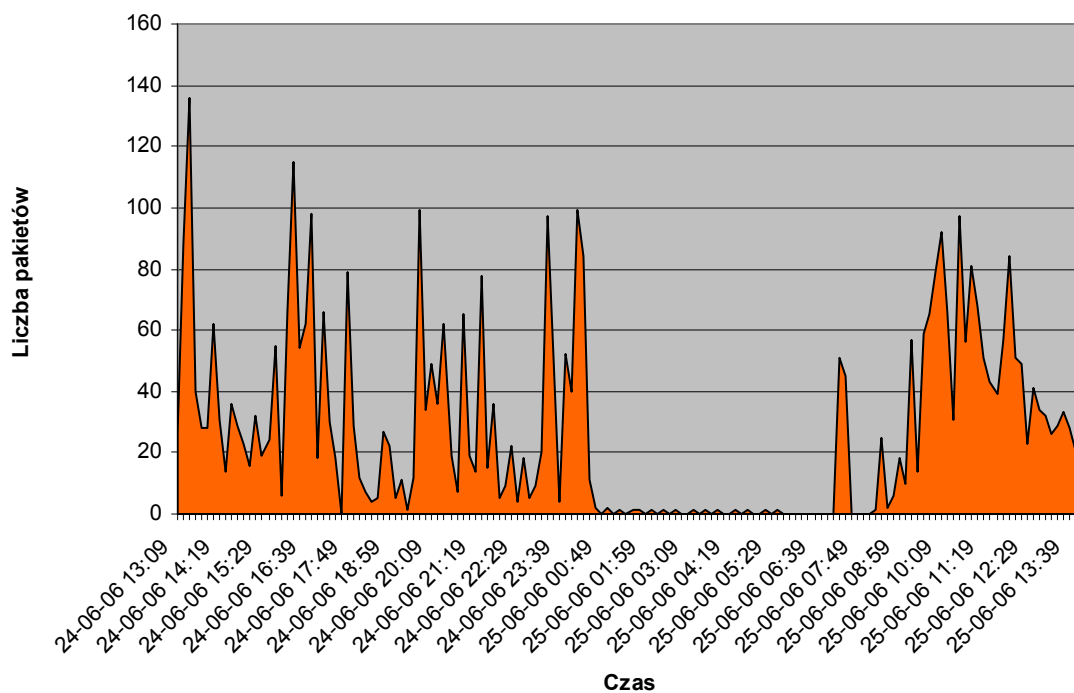




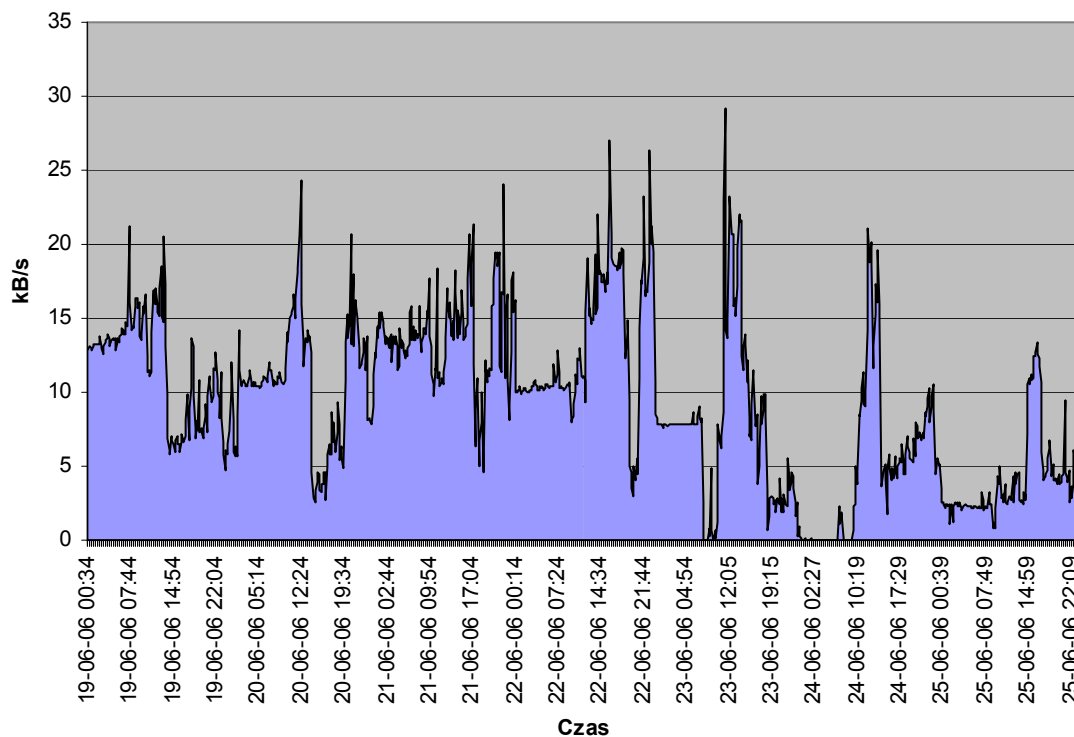
**Rysunek 56: Statystyka wysłanych pakietów UDP port 53 (DNS) (przebieg dobowy).** Źródło: opracowanie własne.



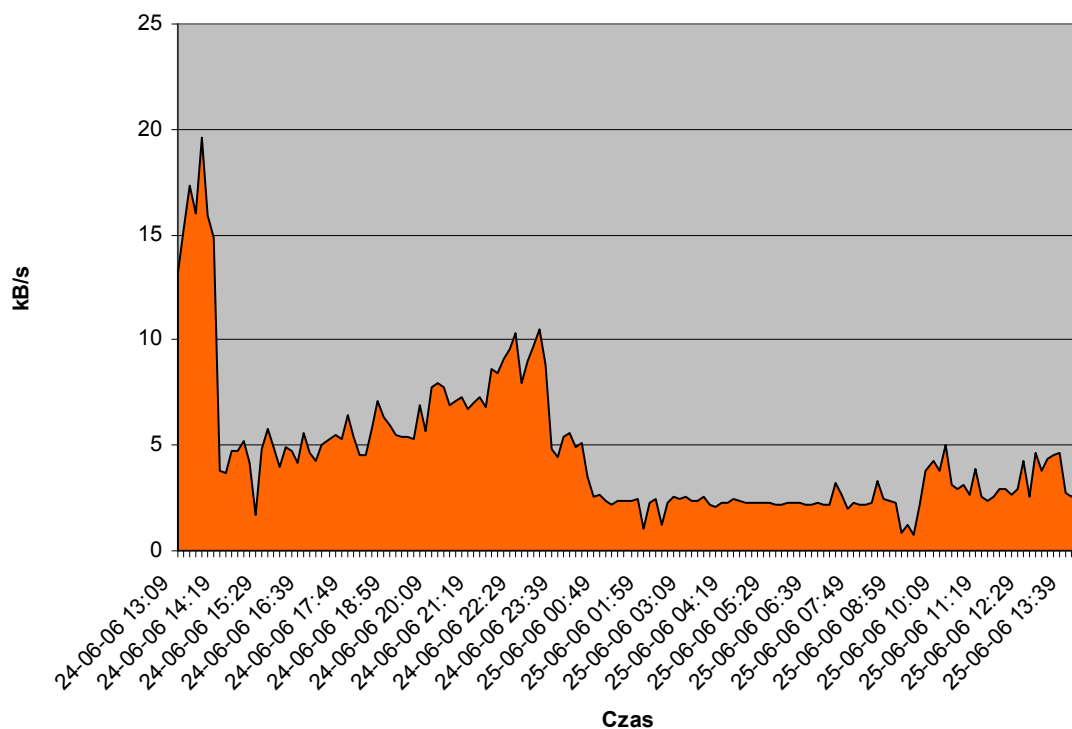
**Rysunek 57: Statystyka odebranych pakietów UDP port 53 (DNS) (przebieg tygodniowy).** Źródło: opracowanie własne.



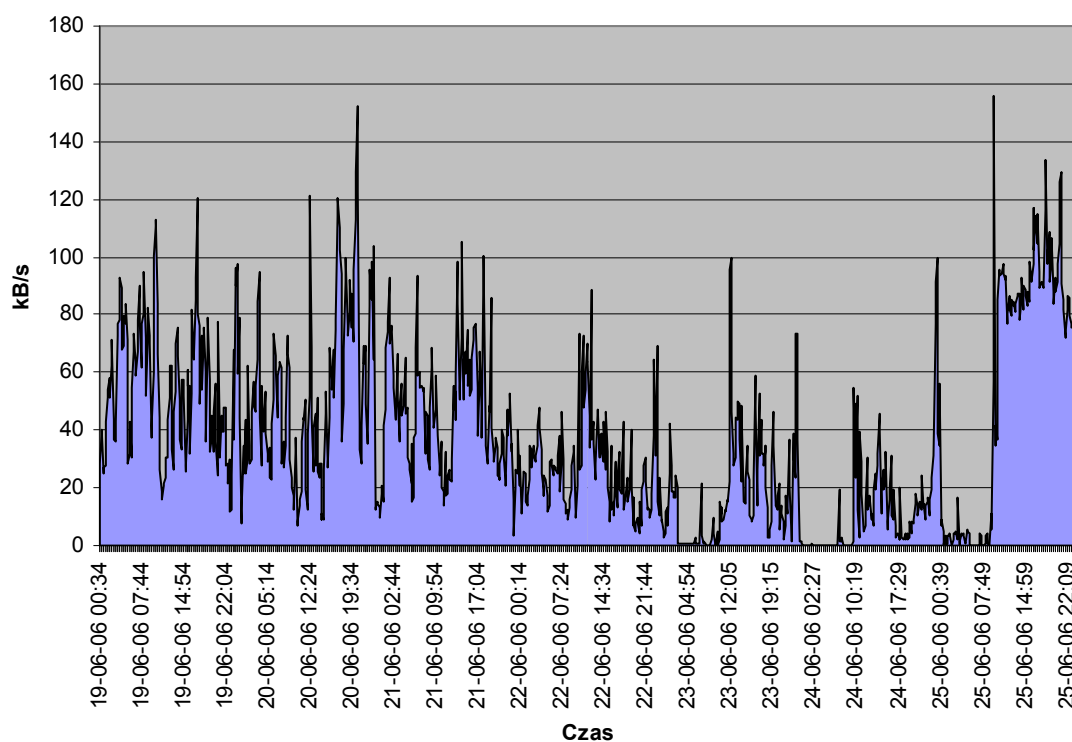
**Rysunek 58: Statystyka odebranych pakietów UDP port 53 (DNS) (przebieg dobowy). Źródło: opracowanie własne.**



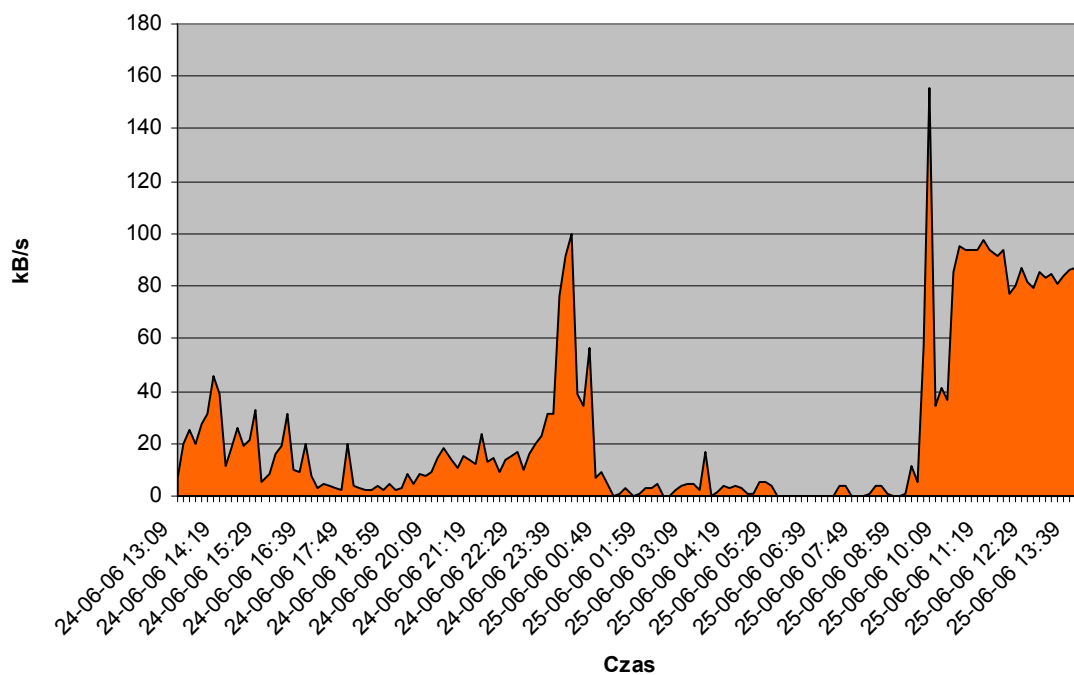
**Rysunek 59: Statystyka wysyłania pakietów TCP (przebieg tygodniowy). Źródło: opracowanie własne.**



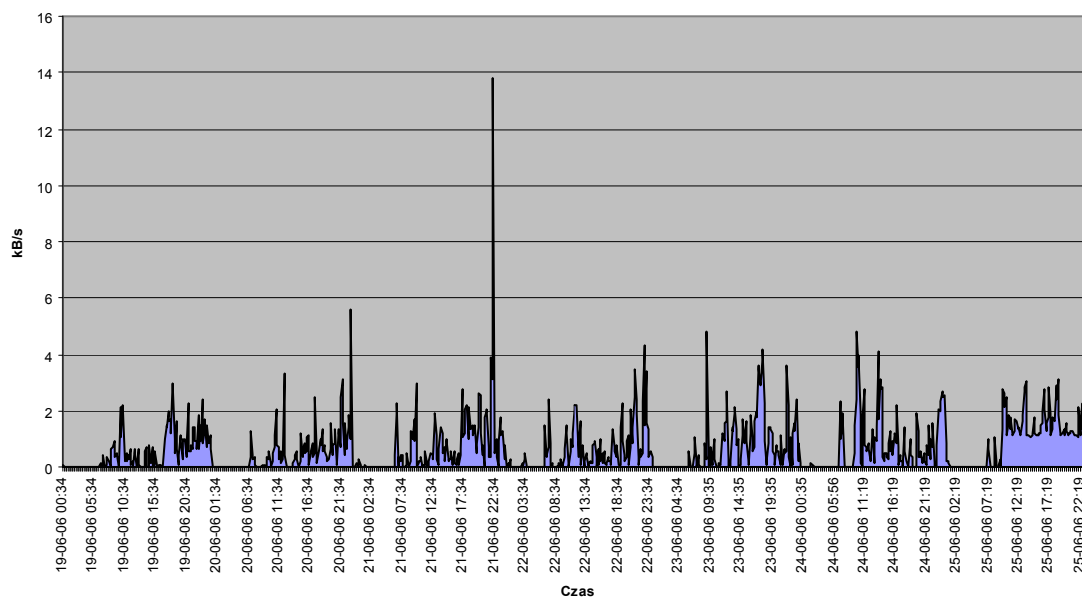
Rysunek 60: Statystyka wysyłania pakietów TCP (przebieg dobowy). Źródło: opracowanie własne.



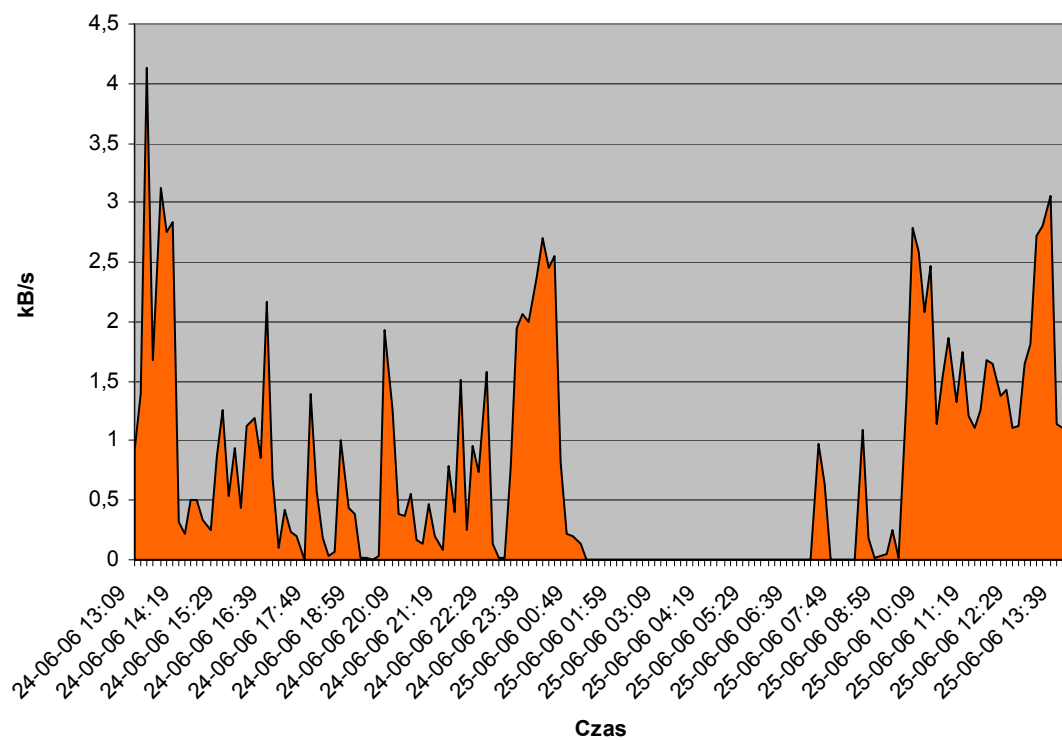
Rysunek 61: Statystyka odbierania pakietów TCP (przebieg tygodniowy). Źródło: opracowanie własne.



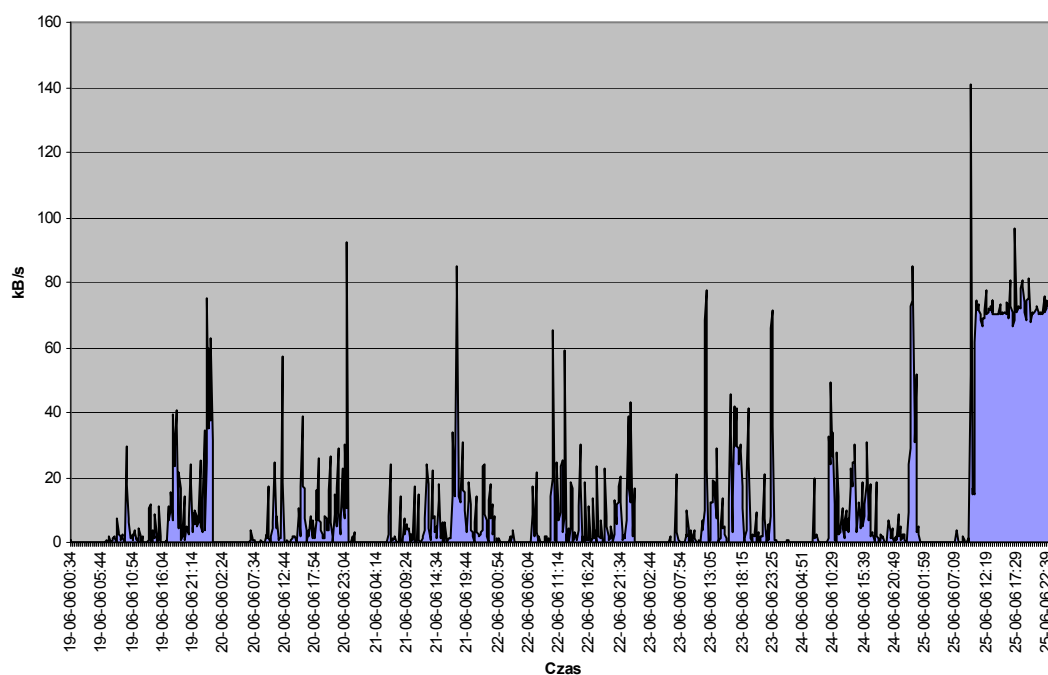
Rysunek 62: Statystyka odbierania pakietów TCP (przebieg dobowy). Źródło: opracowanie własne.



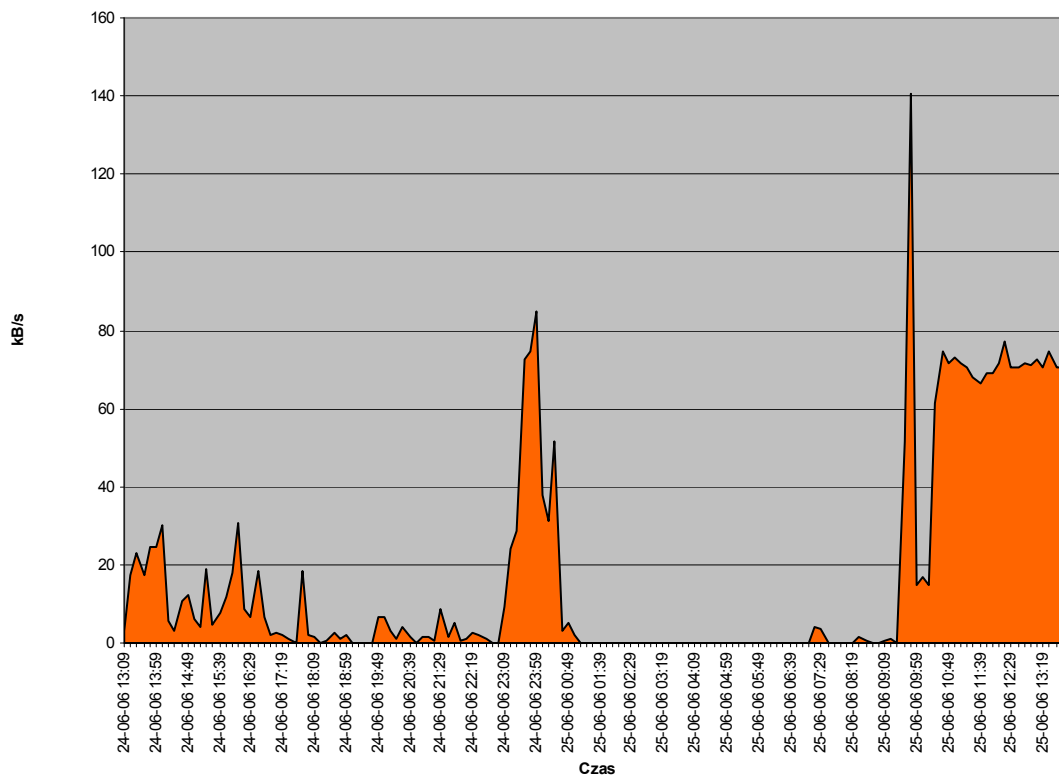
Rysunek 63: Statystyka wysyłania pakietów TCP na port 80 (przebieg tygodniowy). Źródło: opracowanie własne.



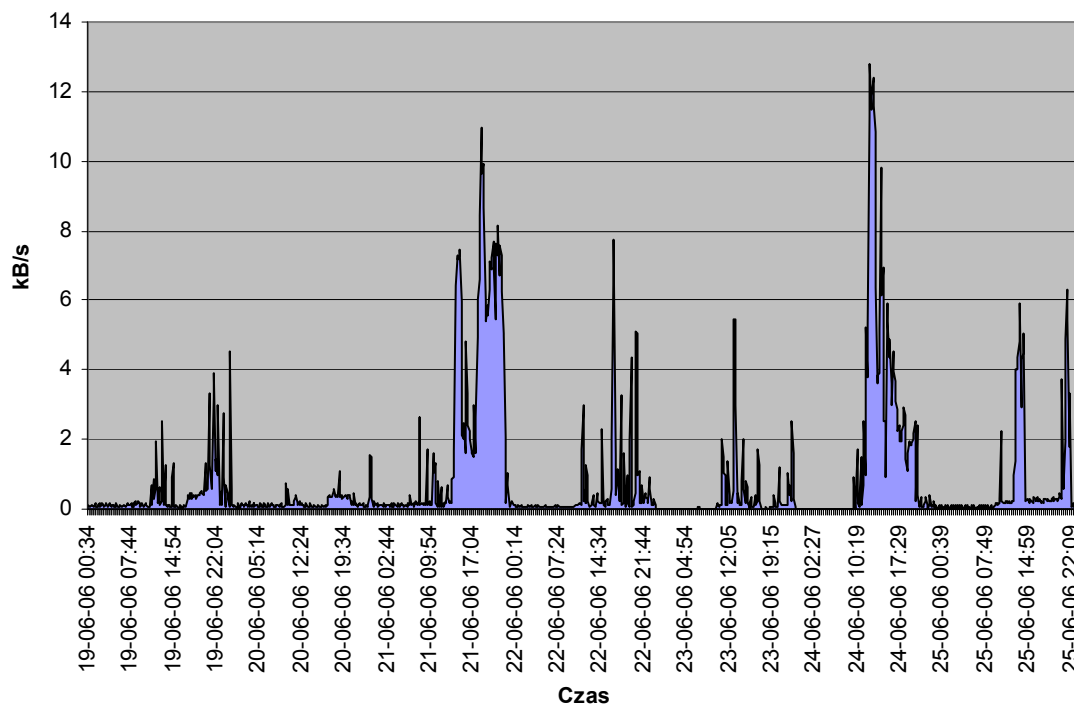
**Rysunek 64: Statystyka wysyłania pakietów TCP na port 80 (przebieg dobowy). Źródło: opracowanie własne.**



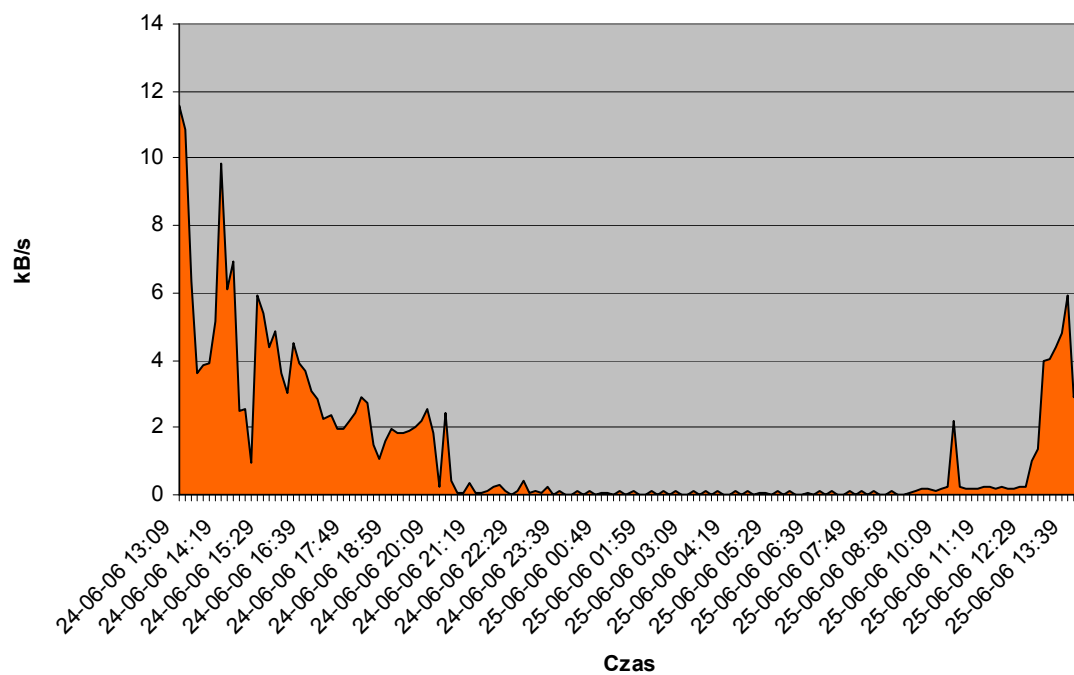
**Rysunek 65: Statystyka odbierania pakietów TCP z portu 80 (przebieg tygodniowy). Źródło: opracowanie własne.**



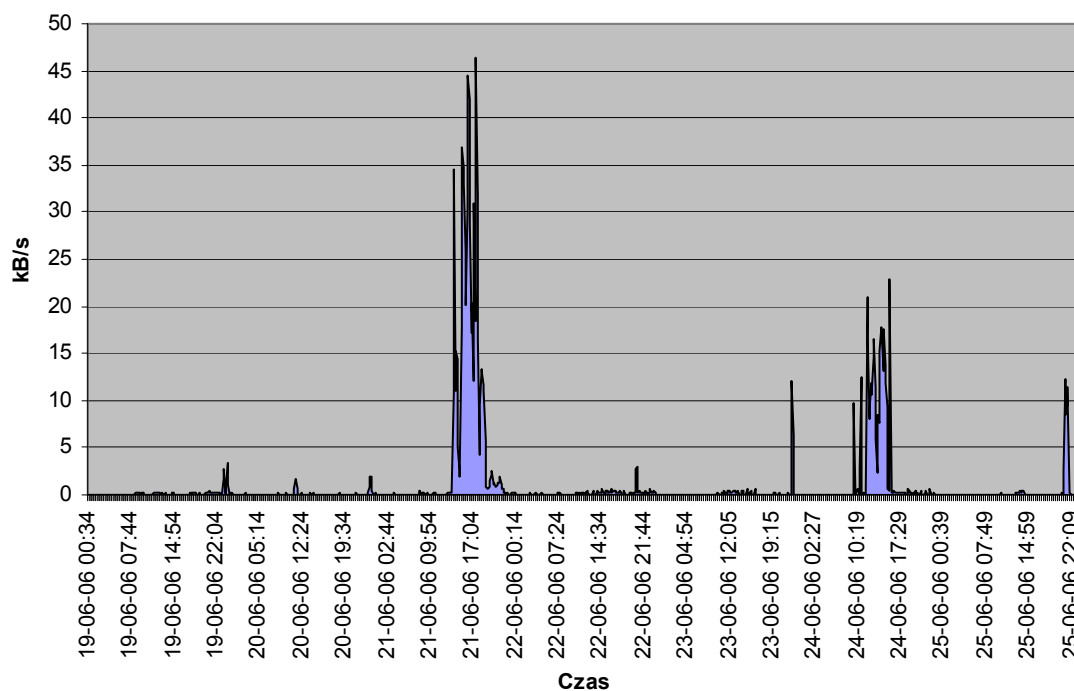
Rysunek 66: Statystyka odbierania pakietów TCP z portu 80 (przebieg dobowy). Źródło: opracowanie własne.



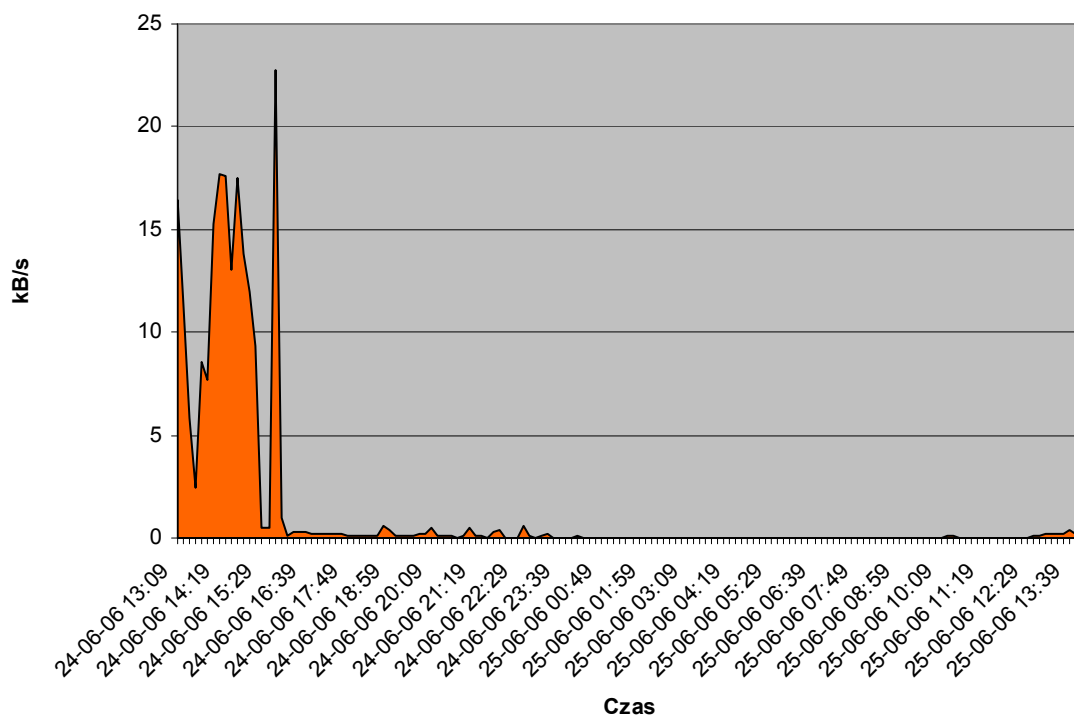
Rysunek 67: Statystyka wysyłania pakietów UDP (przebieg tygodniowy). Źródło: opracowanie własne.



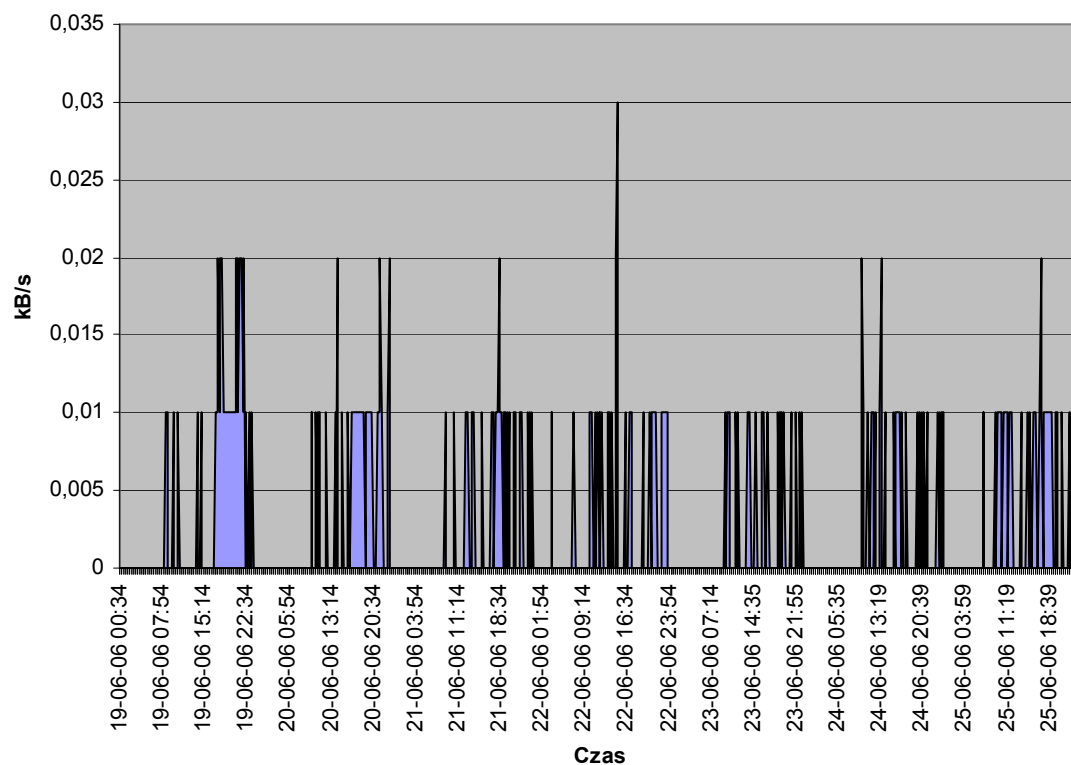
Rysunek 68: Statystyka wysyłania pakietów UDP (przebieg dobowy). Źródło: opracowanie własne.



Rysunek 69: Statystyka odbierania pakietów UDP (przebieg tygodniowy). Źródło: opracowanie własne.

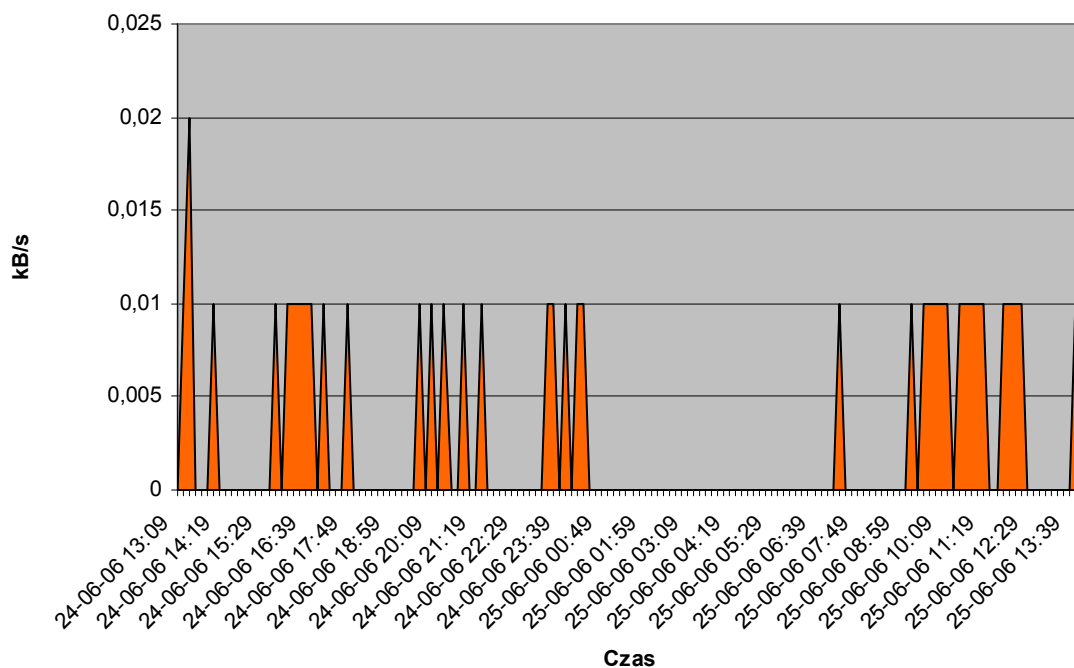


Rysunek 70: Statystyka odbierania pakietów UDP (przebieg dobowy). Źródło: opracowanie własne.

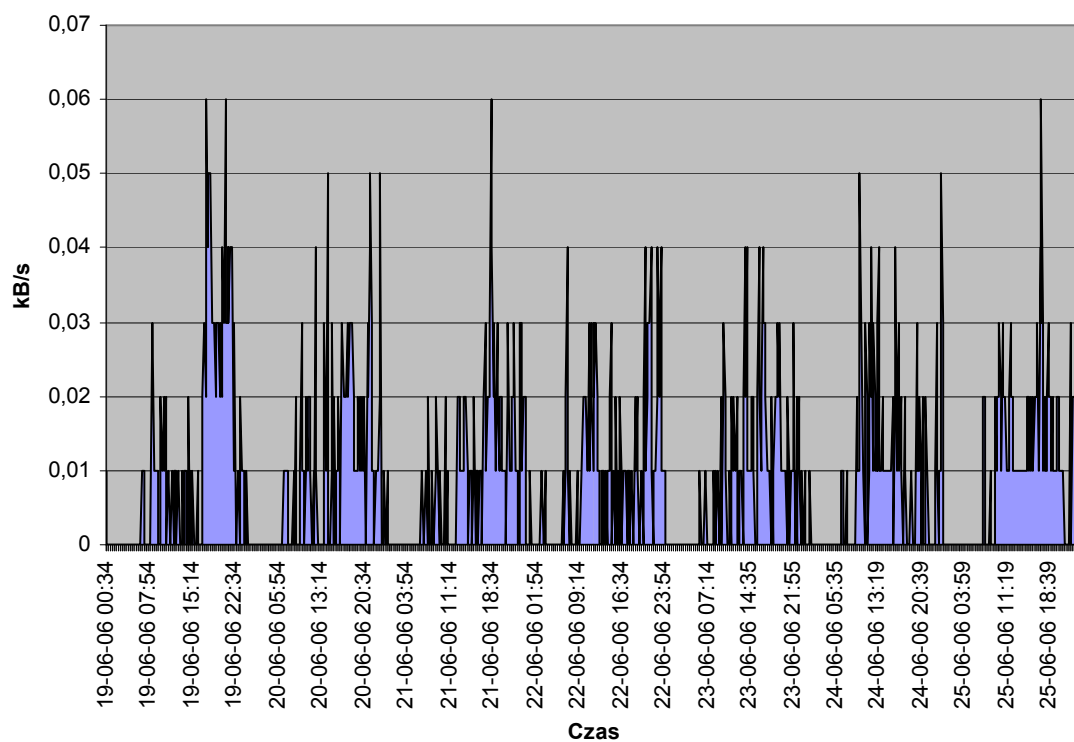


Rysunek 71: Statystyka wysyłania pakietów UDP na port 53 (przebieg tygodniowy). Źródło: opracowanie własne.

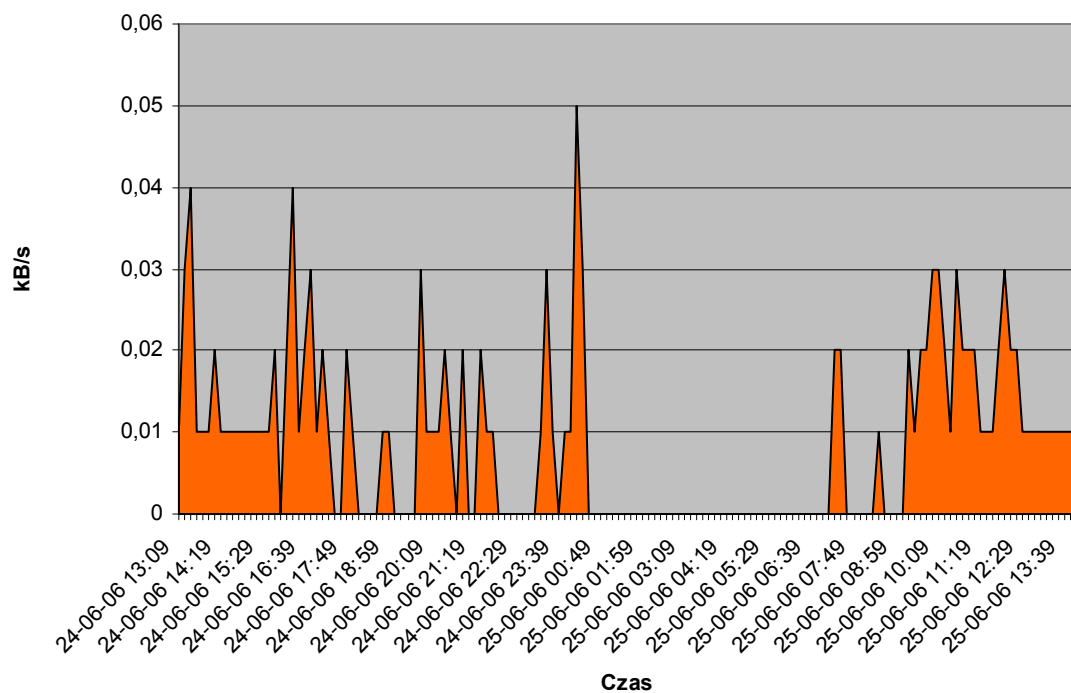




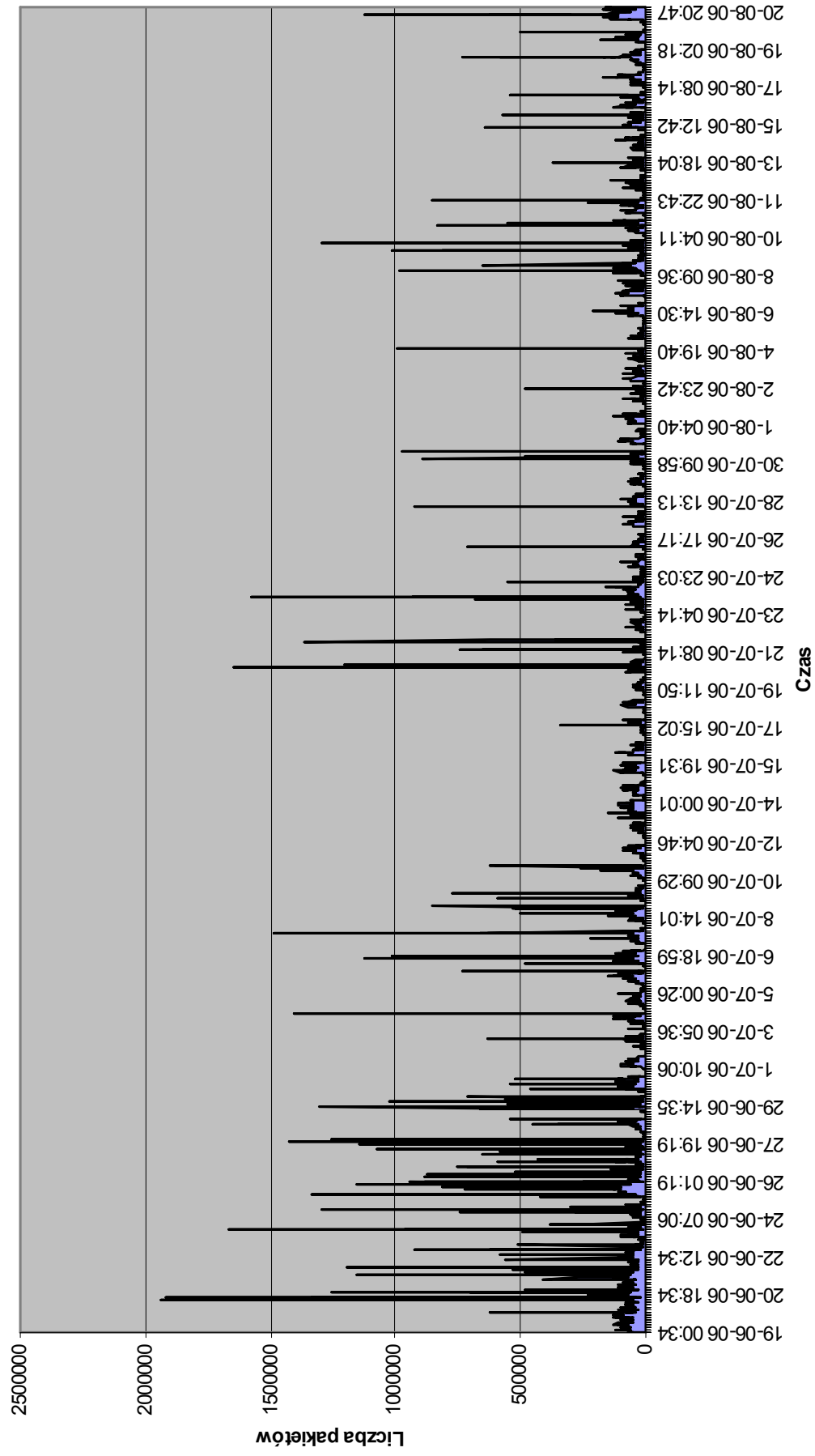
**Rysunek 72: Statystyka wysyłania pakietów UDP na port 53 (przebieg dobowy). Źródło: opracowanie własne.**



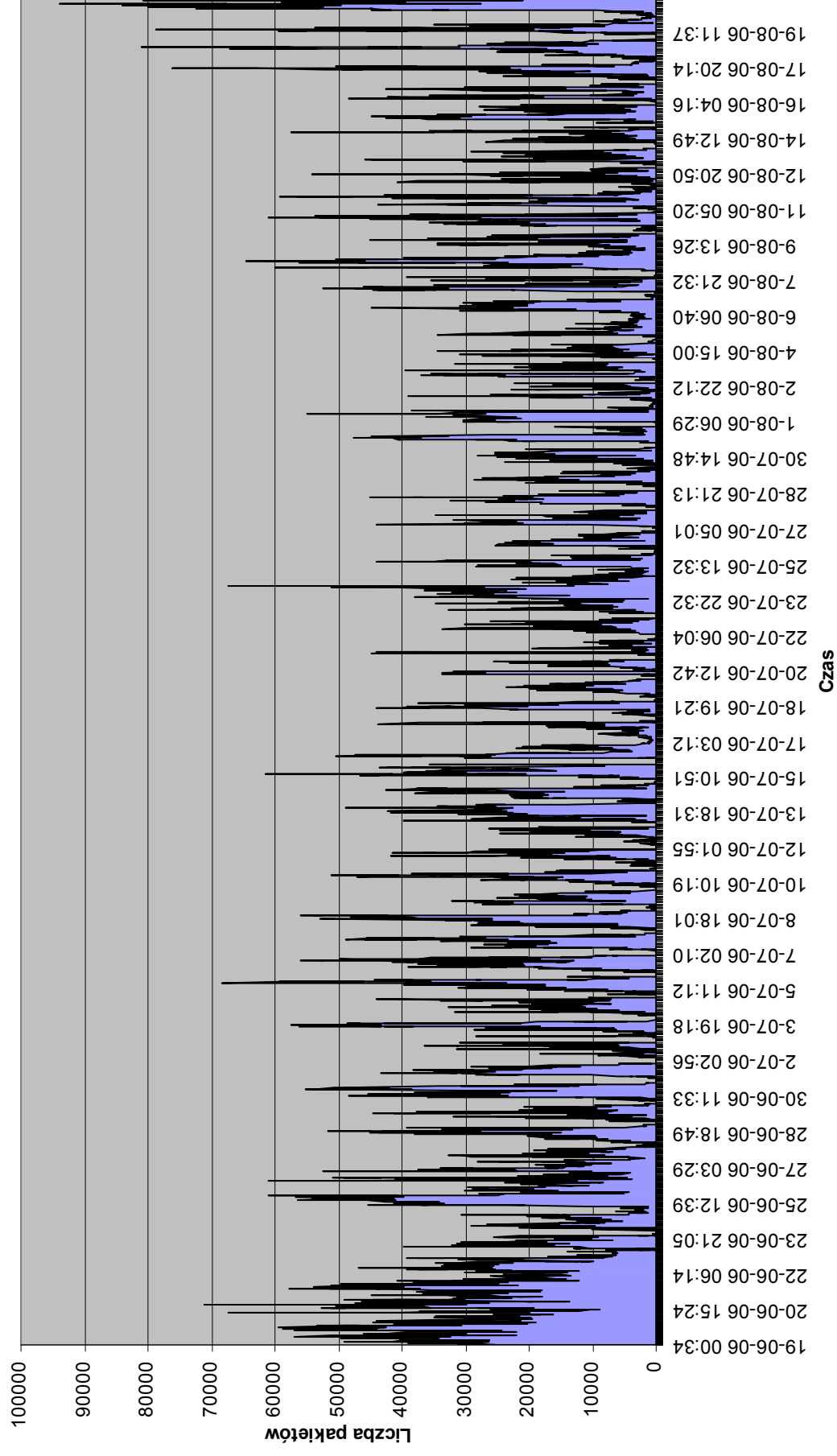
**Rysunek 73: Statystyka odbierania pakietów UDP z portu 53 (przebieg tygodniowy). Źródło: opracowanie własne.**



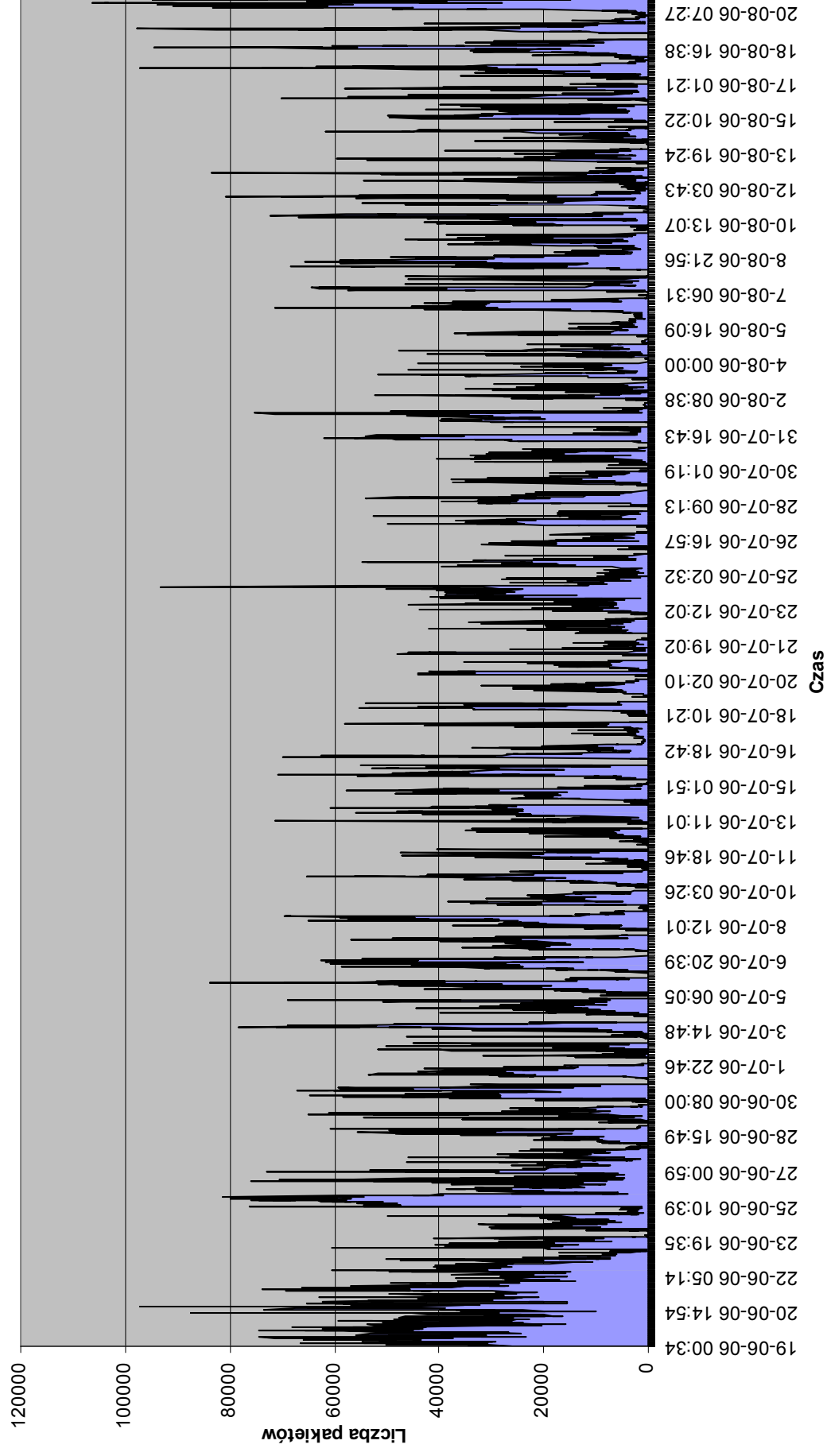
**Rysunek 74: Statystyka odbierania pakietów UDP z portu 53 (przebieg dobowy). Źródło: opracowanie własne.**



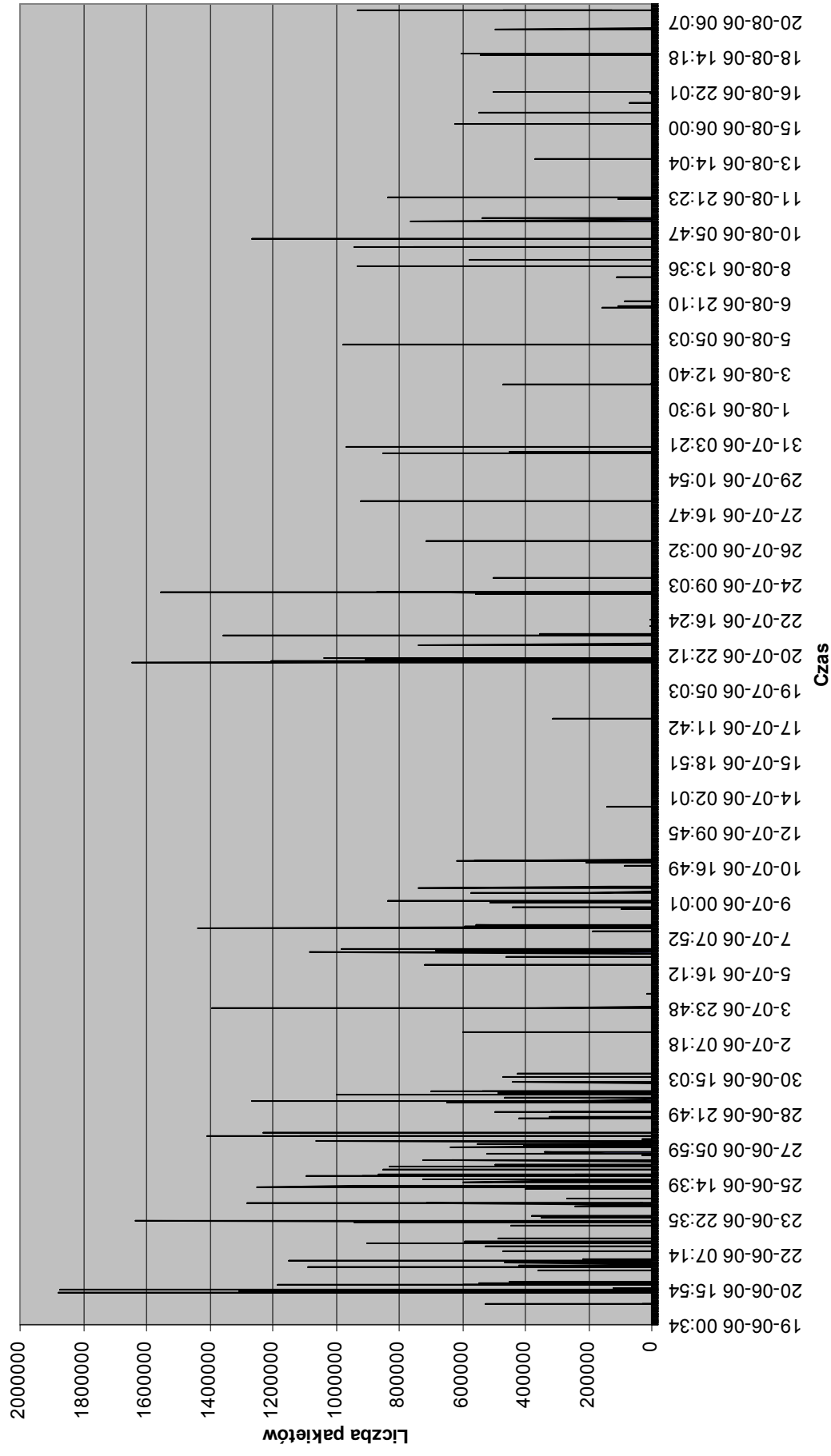
Rysunek 75: Statystyka pakietów TCP (przebieg miesięczny). Źródło: opracowanie własne.



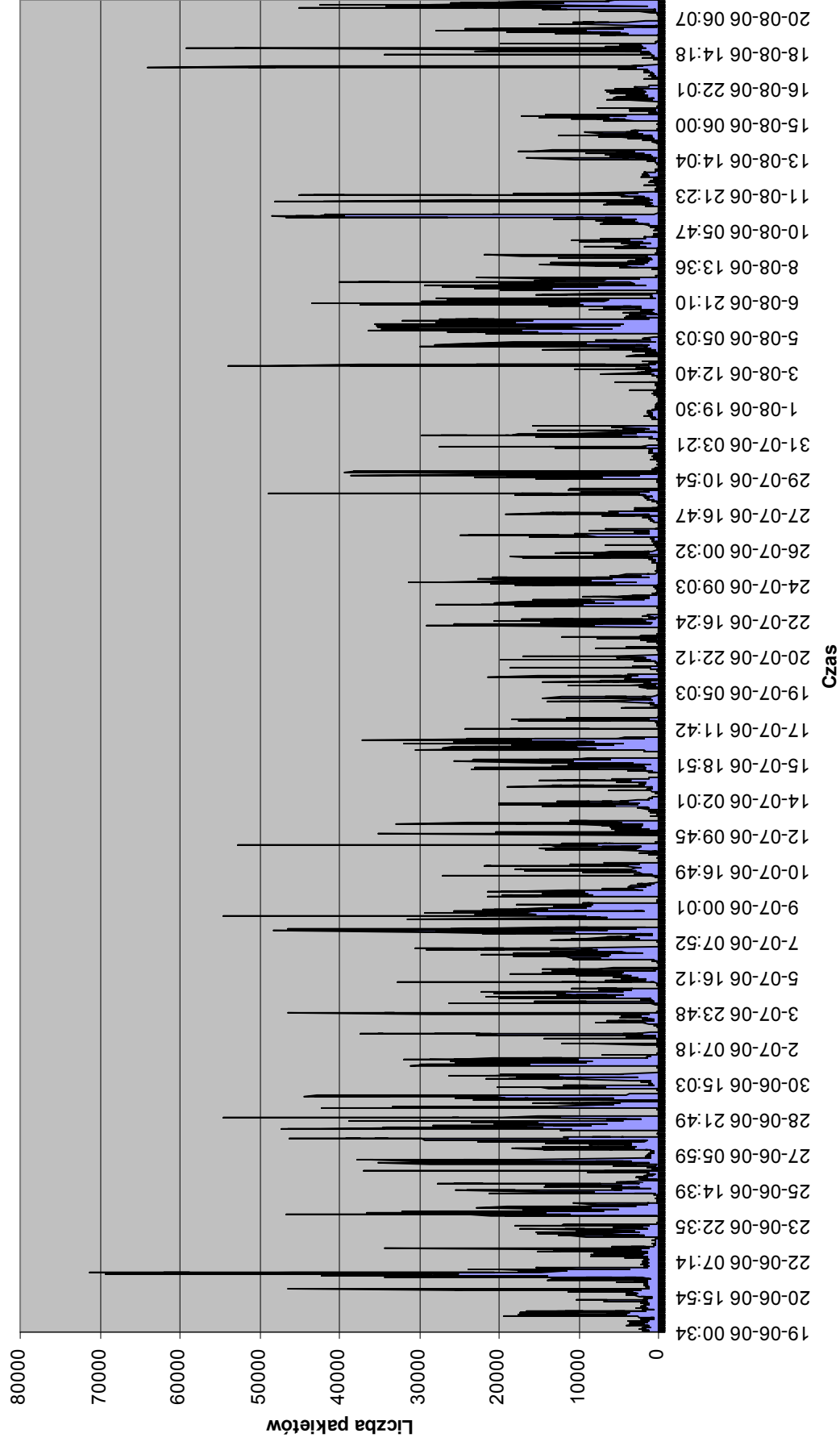
Rysunek 76: Statystyka wysłanych pakietów TCP (przebieg miesięczny). Źródło: opracowanie własne.



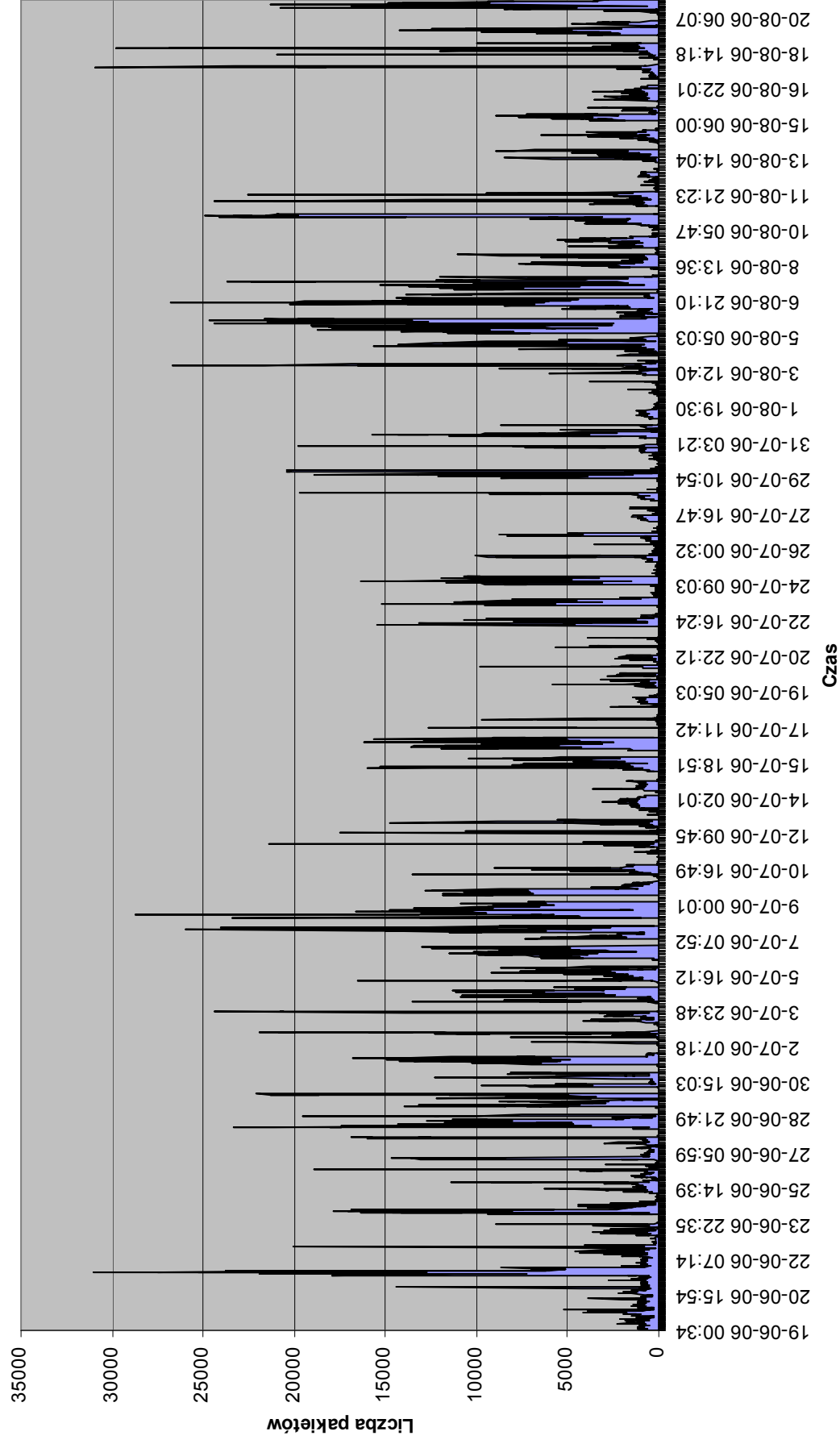
Rysunek 77: Statystyka odebranych pakietów TCP (przebieg miesięczny). Źródło: opracowanie własne.



Rysunek 78: Statystyka pakietów TCP wewnątrz sieci LAN (przebieg miesięczny). Źródło: opracowanie własne.

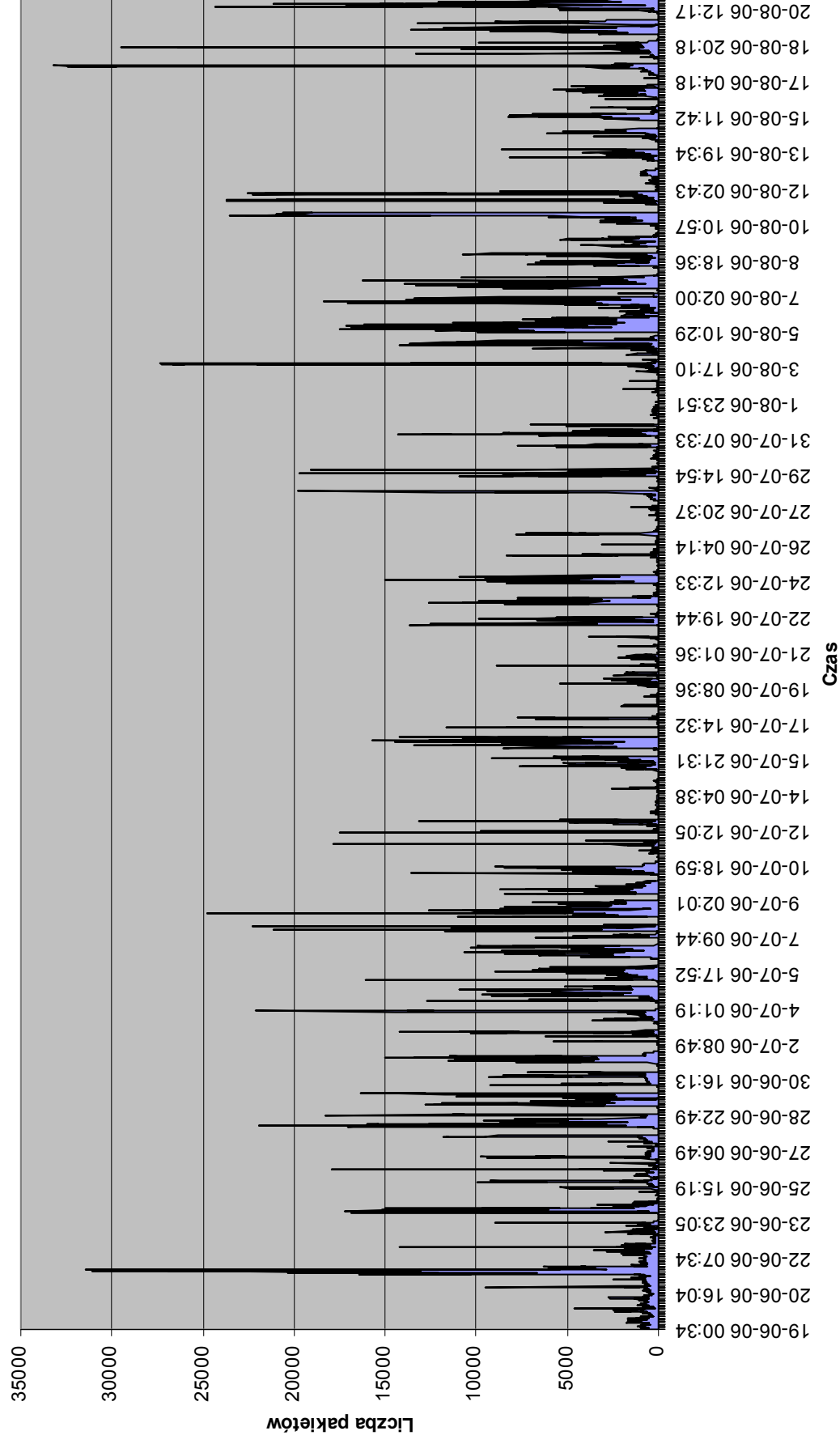


Rysunek 79: Statystyka pakietów UDP (przebieg miesięczny). Źródło: opracowanie własne.

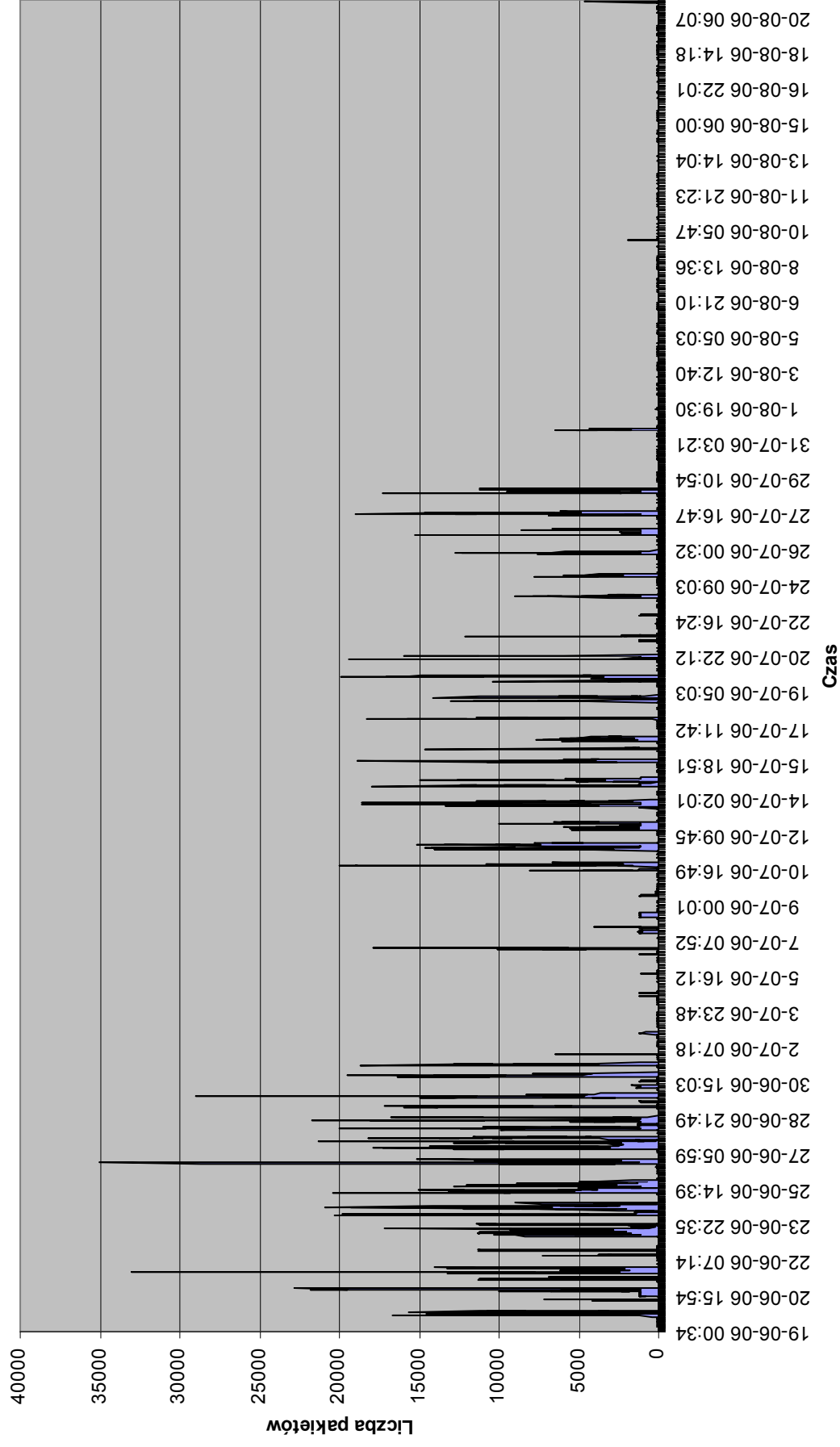


Rysunek 80: Statystyka wysłanych pakietów UDP (przebieg miesięczny). Źródło: opracowanie własne.

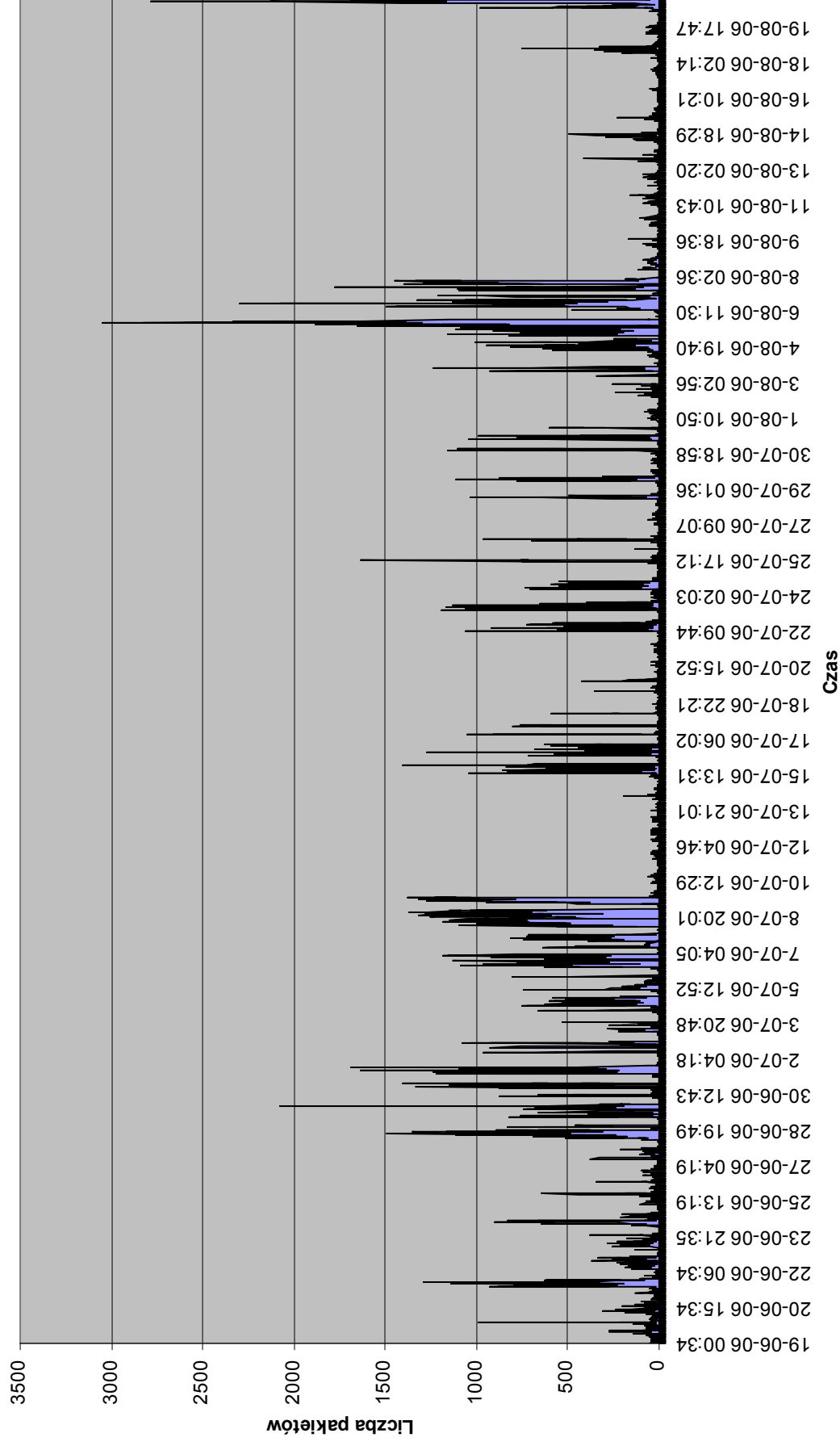




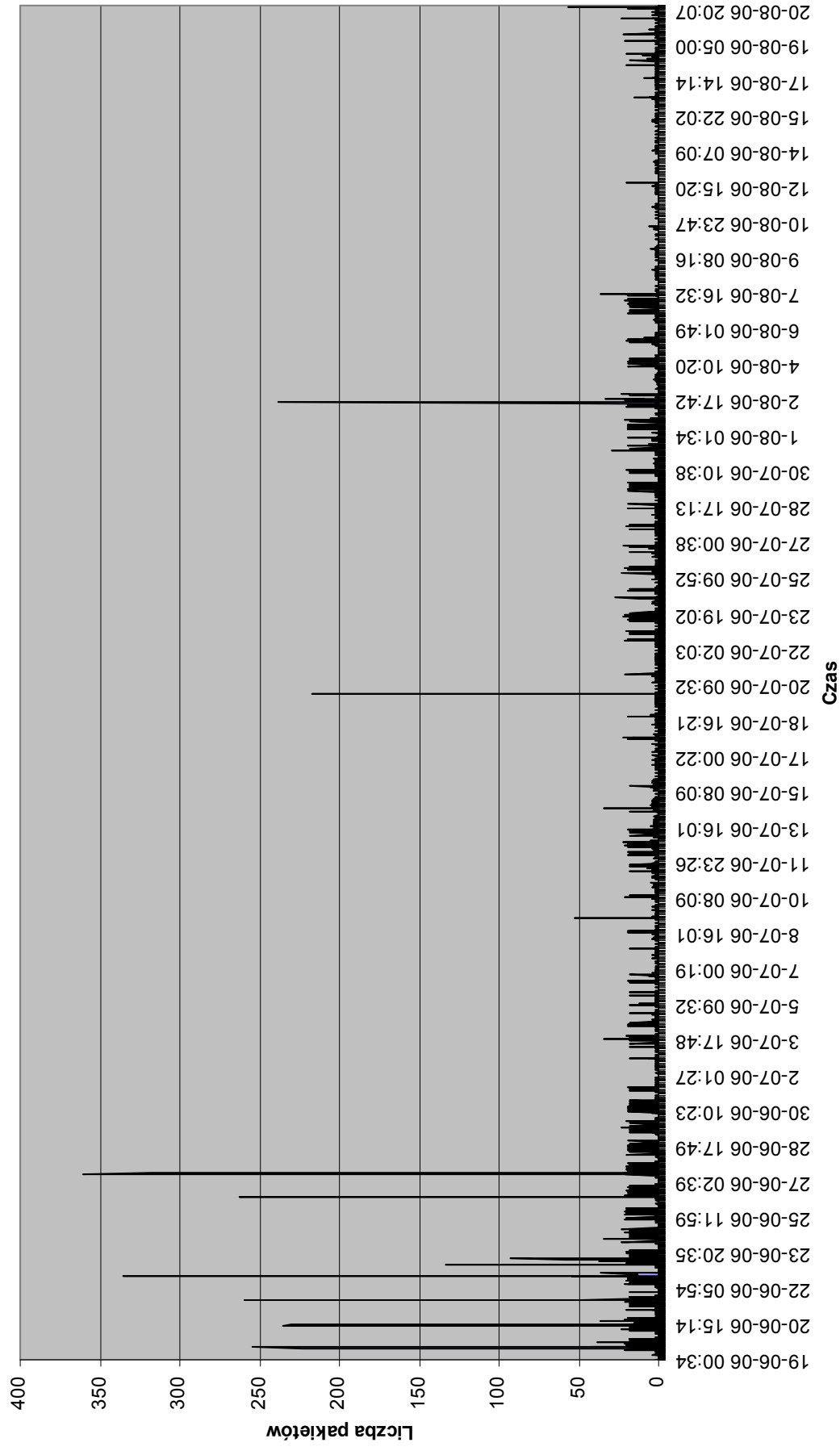
Rysunek 81: Statystyka odebranych pakietów UDP (przebieg miesięczny). Źródło: opracowanie własne.



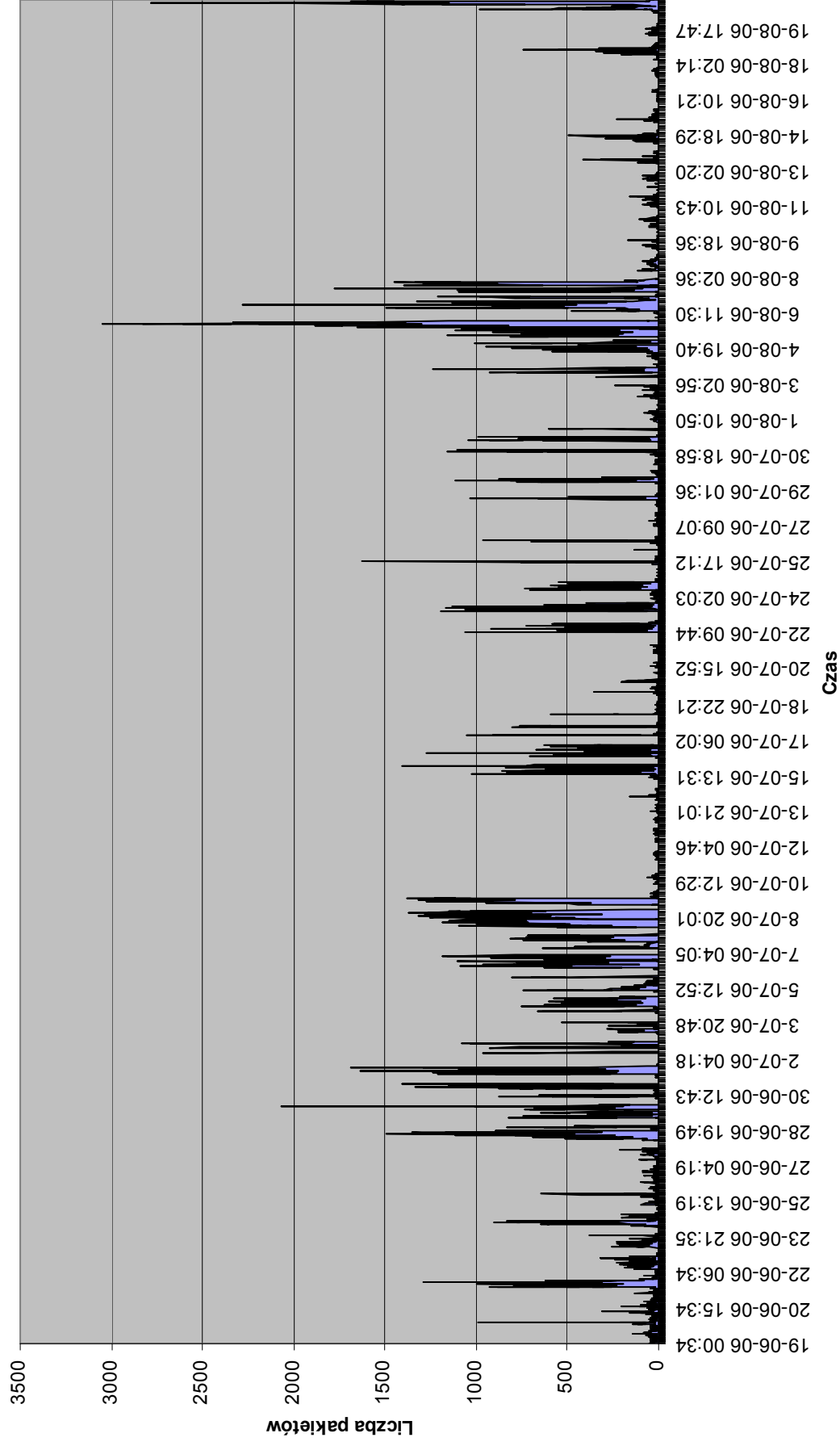
Rysunek 82: Statystyka pakietów UDP wewnątrz sieci LAN (przebieg miesięczny). Źródło: opracowanie własne.



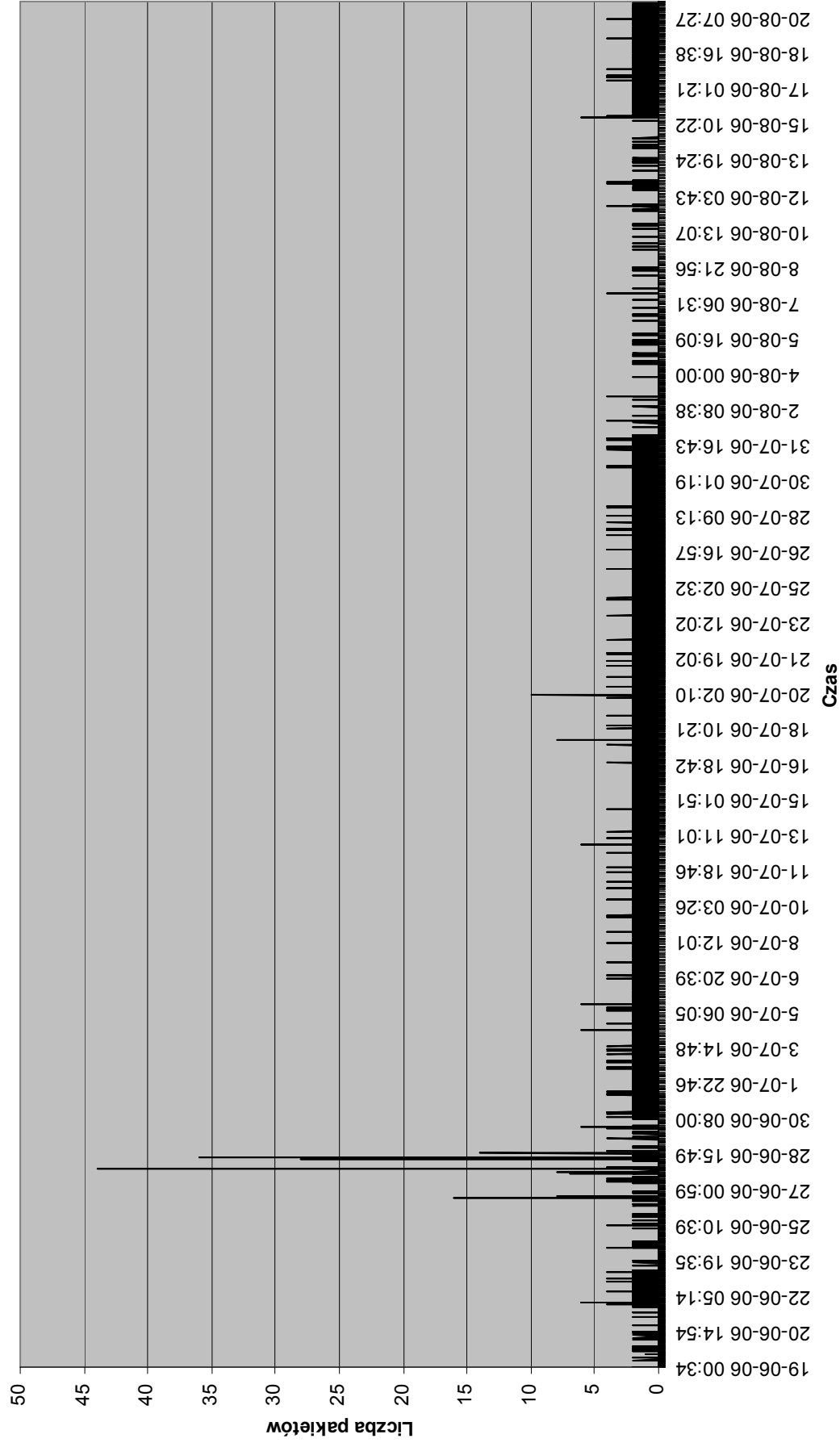
Rysunek 83: Statystyka pakietów ICMP (przebieg miesięczny). Źródło: opracowanie własne.

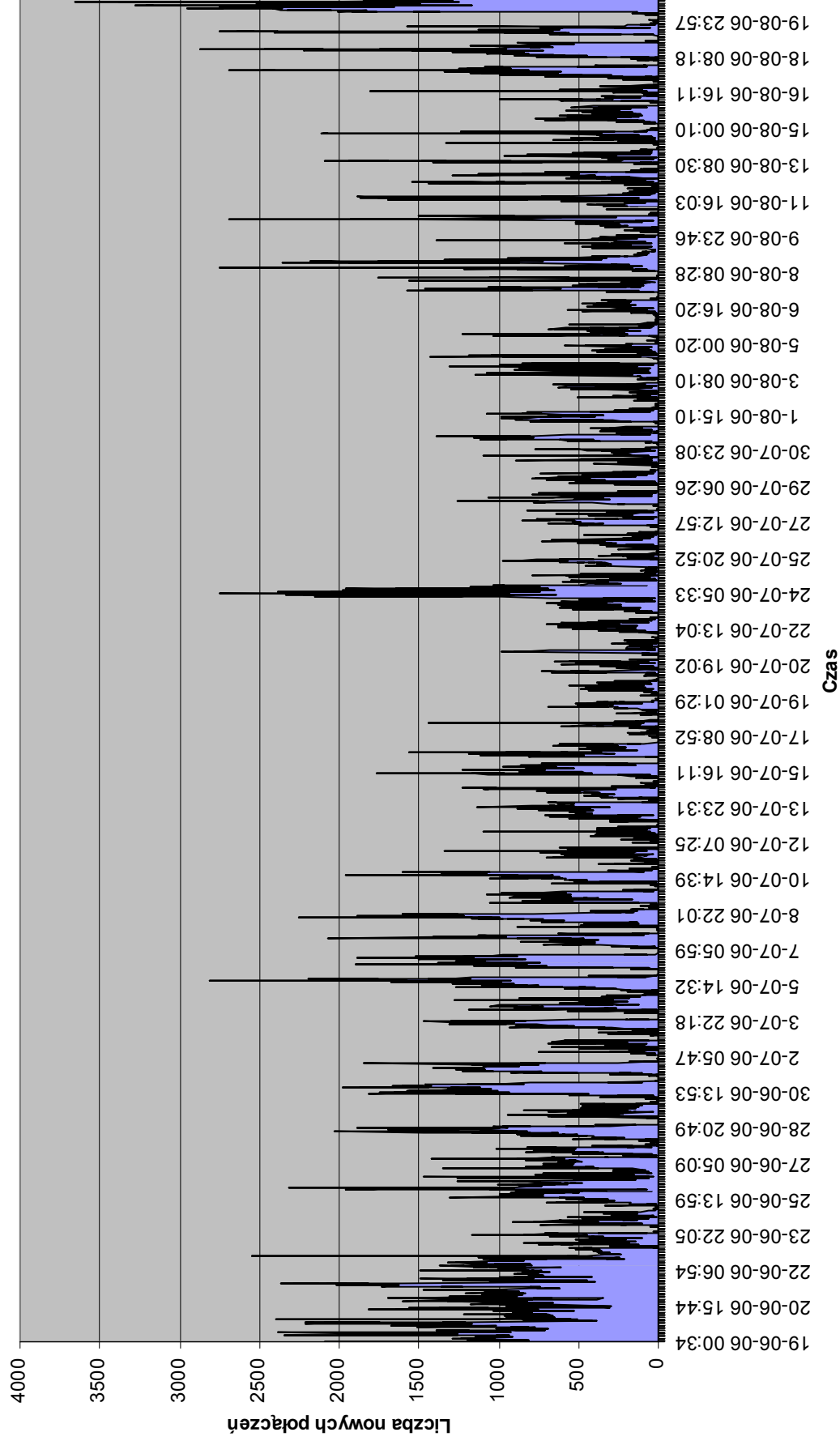


Rysunek 84: Statystyka wysłanych pakietów ICMP (przebieg miesięczny). Źródło: opracowanie własne.

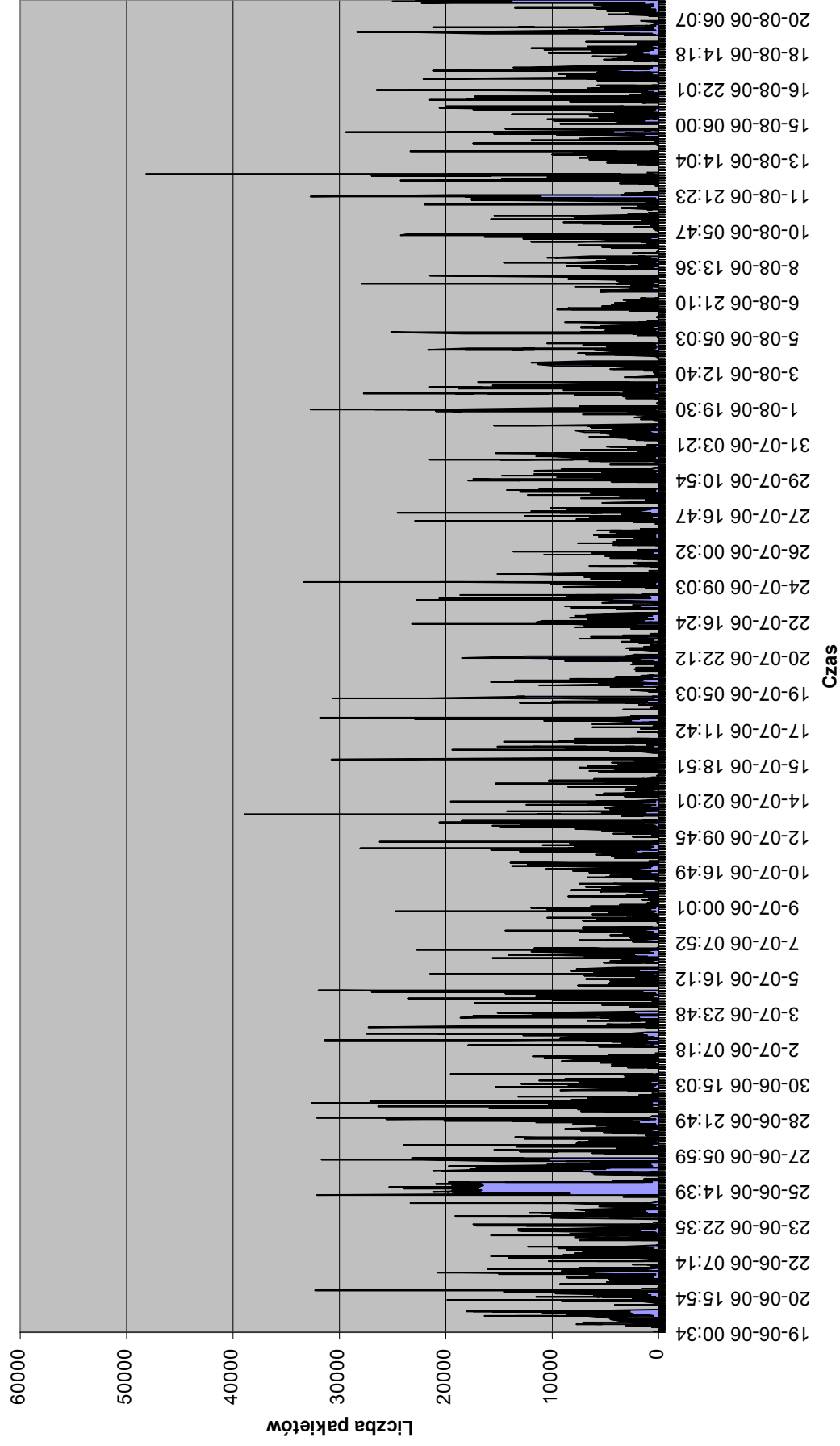


Rysunek 85: Statystyka odebranych pakietów ICMP (przebieg miesięczny). Źródło: opracowanie własne.



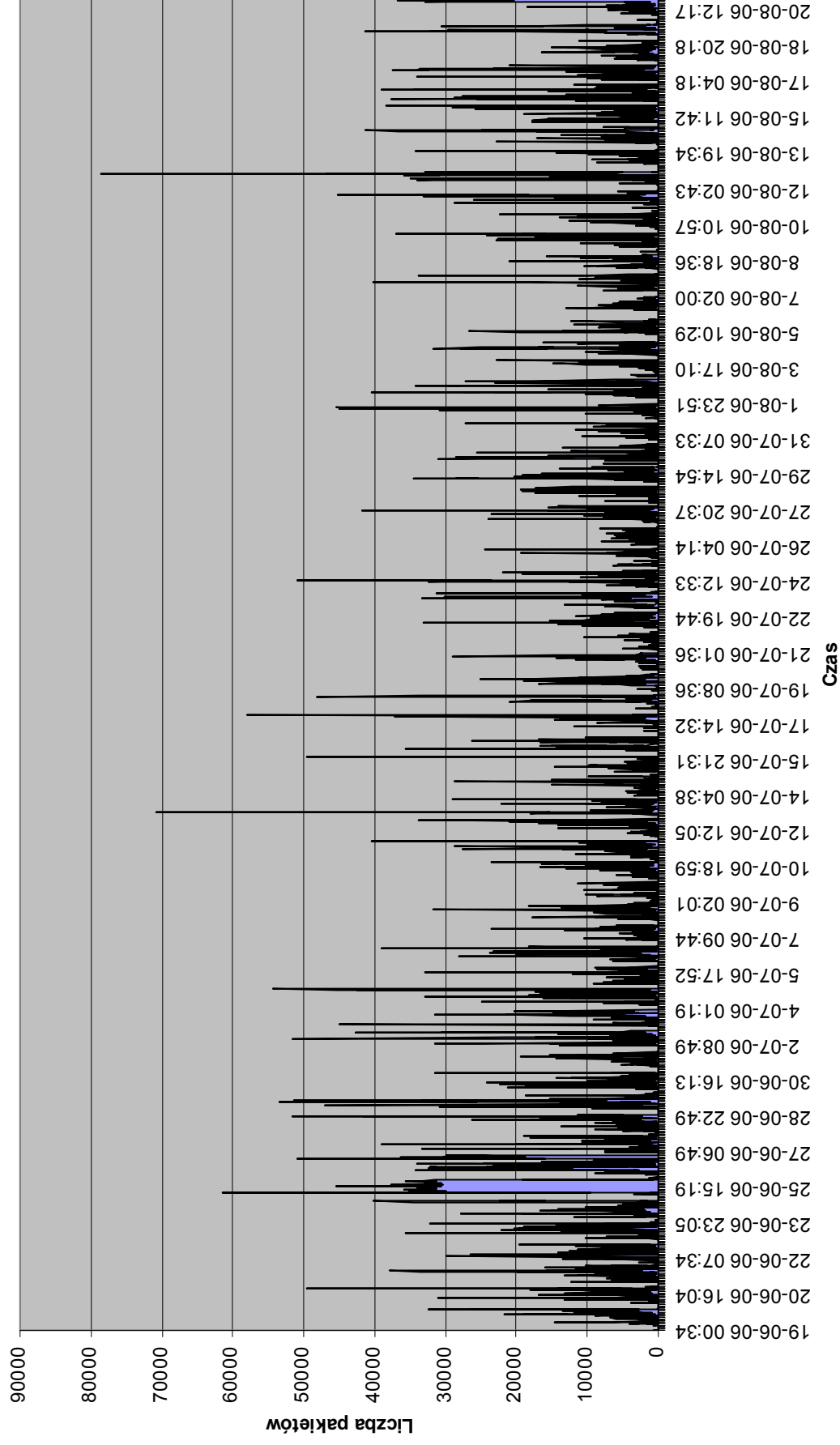


Rysunek 87: Liczba nowych połączeń (przebieg miesięczny). Źródło: opracowanie własne.

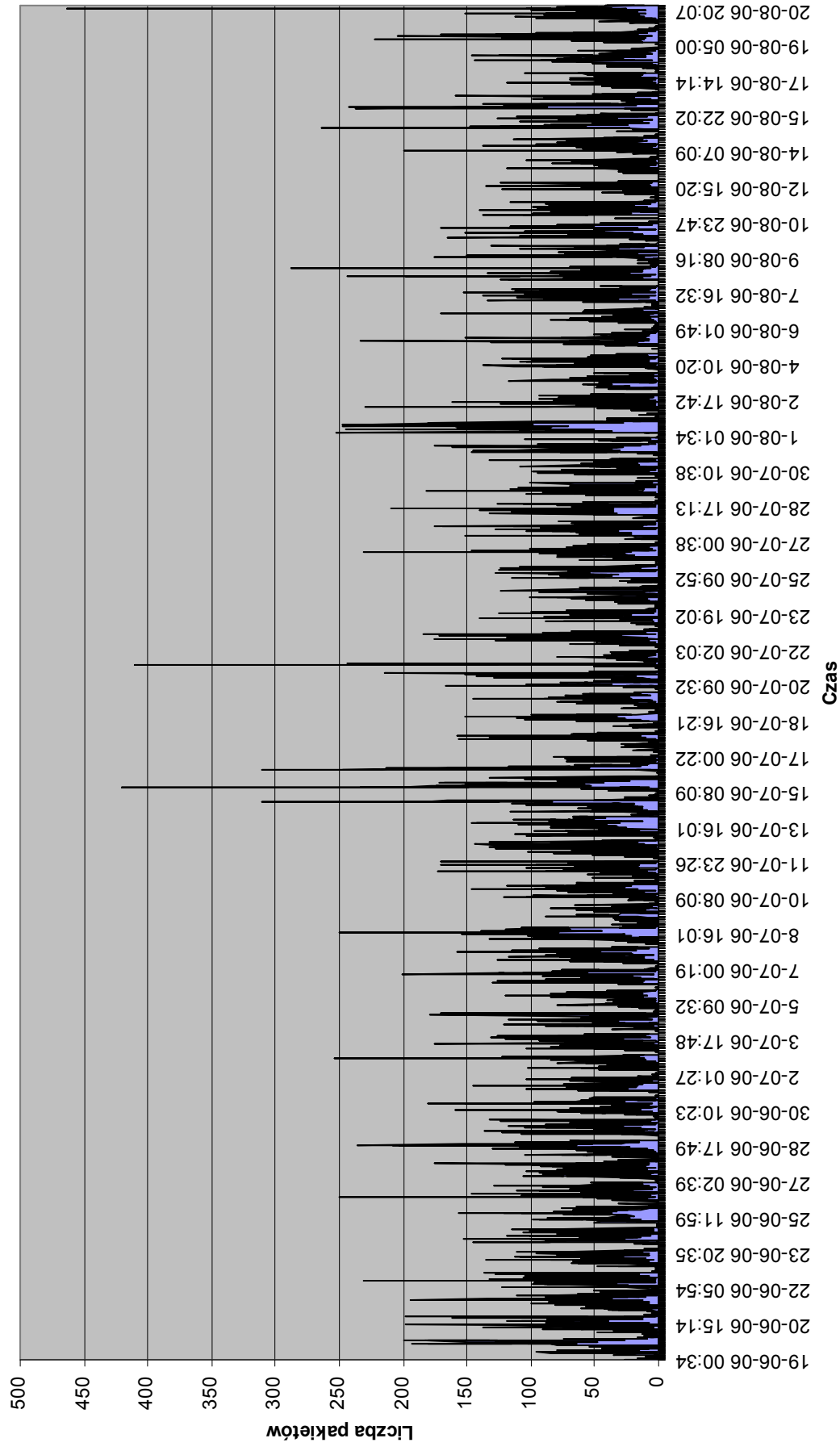


Rysunek 88: Statystyka wysłanych pakietów TCP port 80 (WWW) (przebieg miesięczny). Źródło: opracowanie własne.

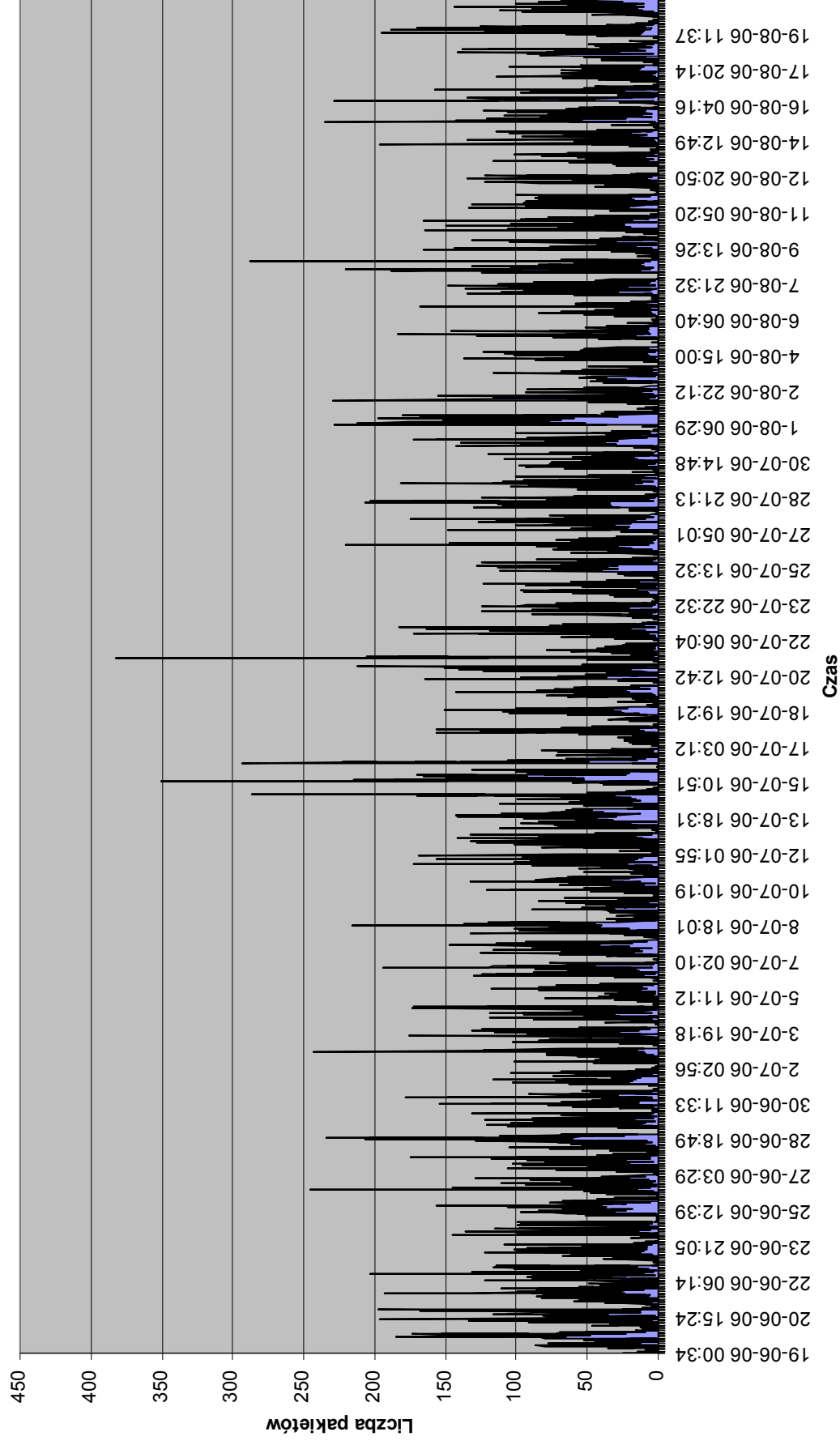




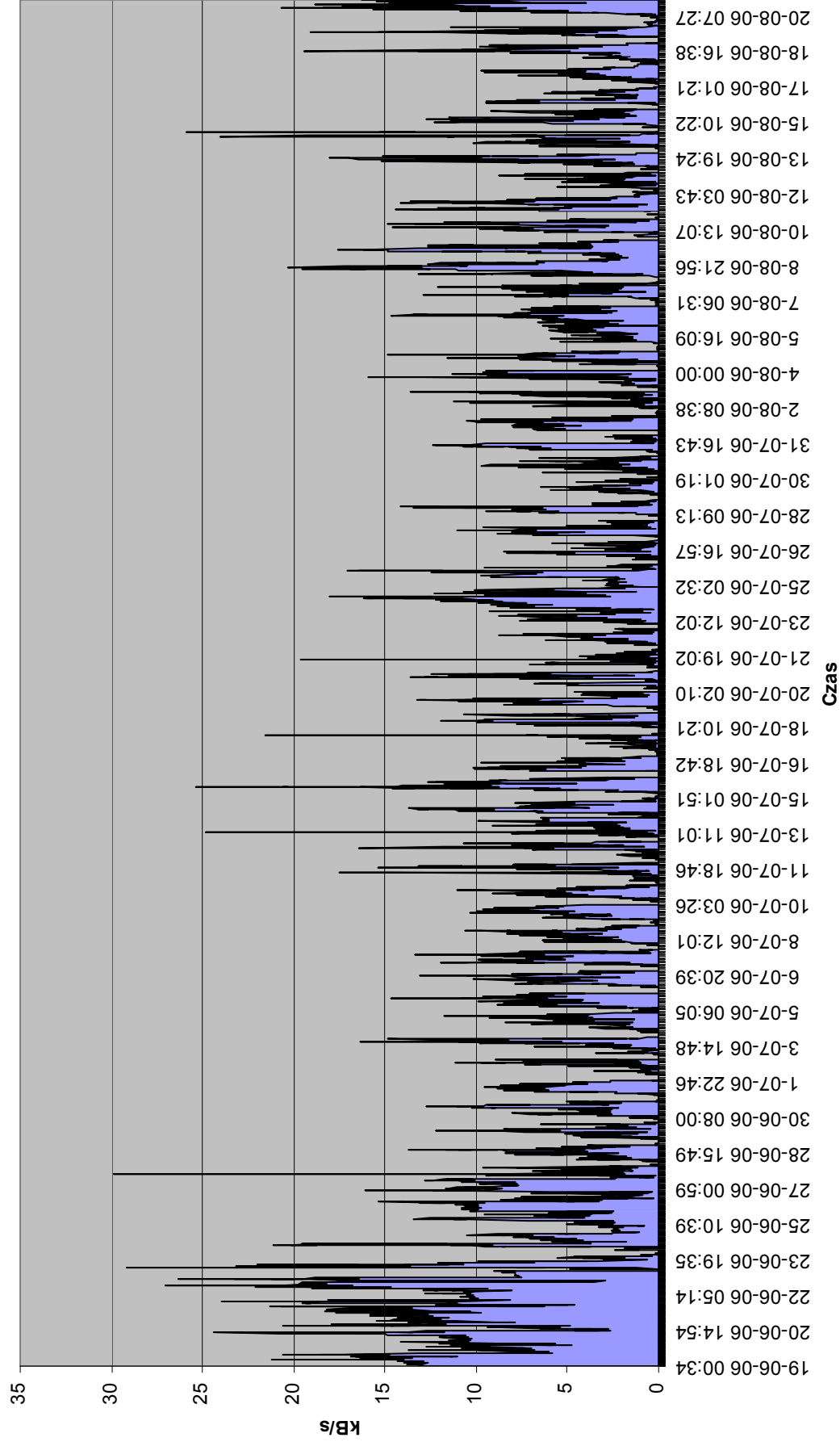
Rysunek 89: Statystyka odebranych pakietów TCP port 80 (WWW) (przebieg miesięczny). Źródło: opracowanie własne.



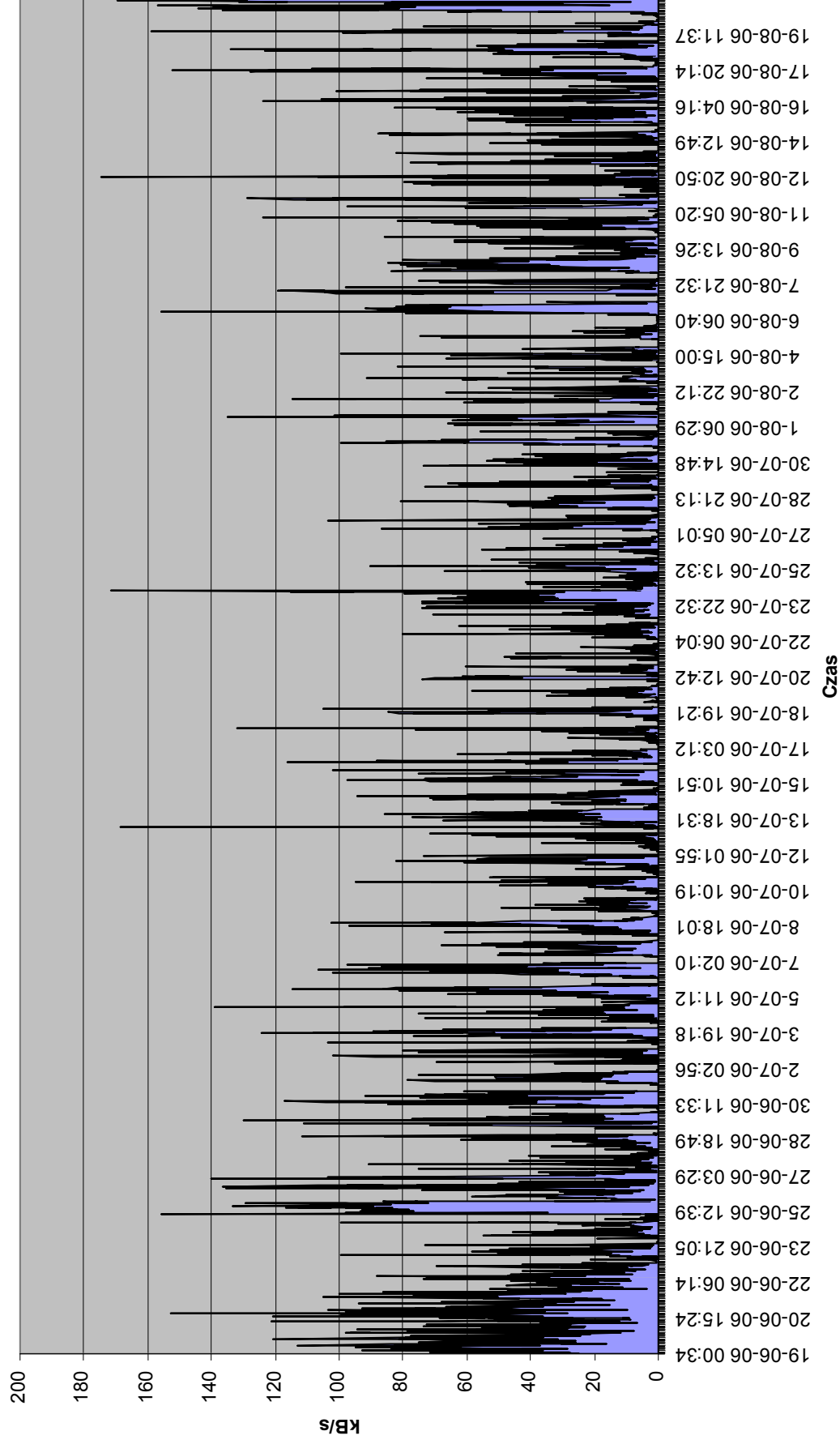
**Rysunek 90: Statystyka wysłanych pakietów UDP port 53 (DNS) (przebieg miesięczny). Źródło: opracowanie własne.**



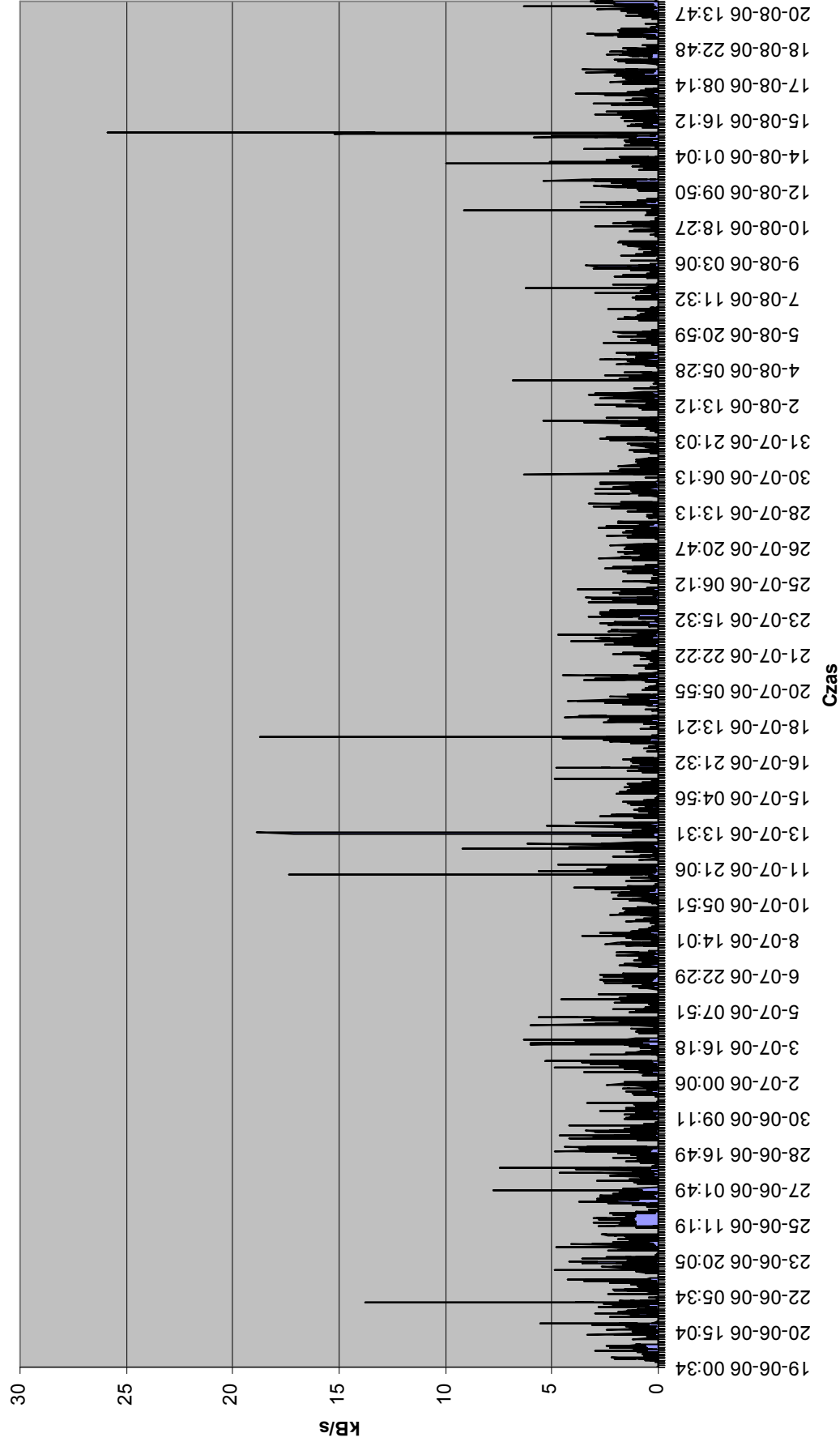
Rysunek 91: Statystyka odebranych pakietów UDP port 53 (DNS) (przebieg miesięczny). Źródło: opracowanie własne.



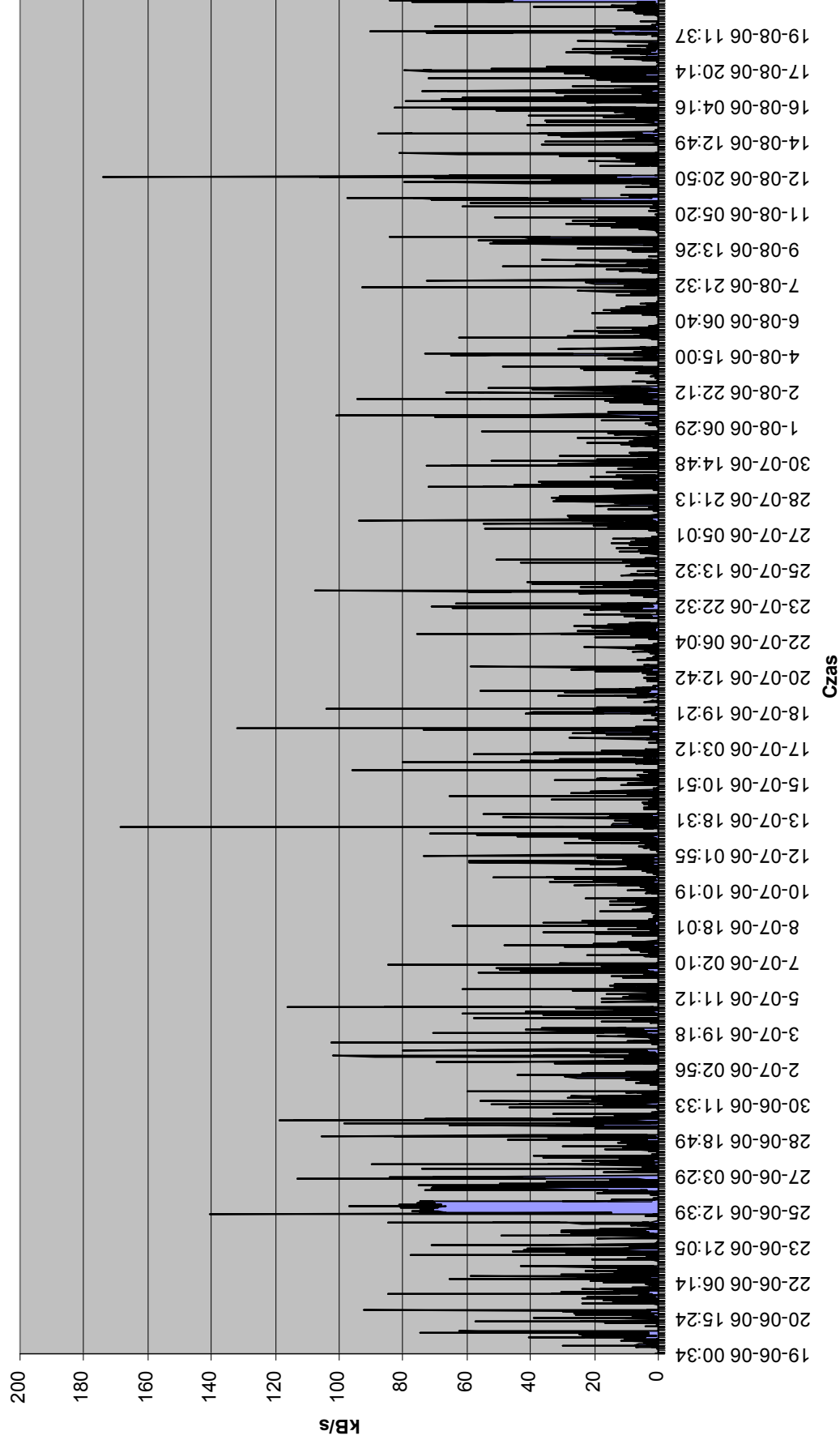
Rysunek 92: Statystyka wysyłania pakietów TCP (przebieg miesięczny). Źródło: opracowanie własne.



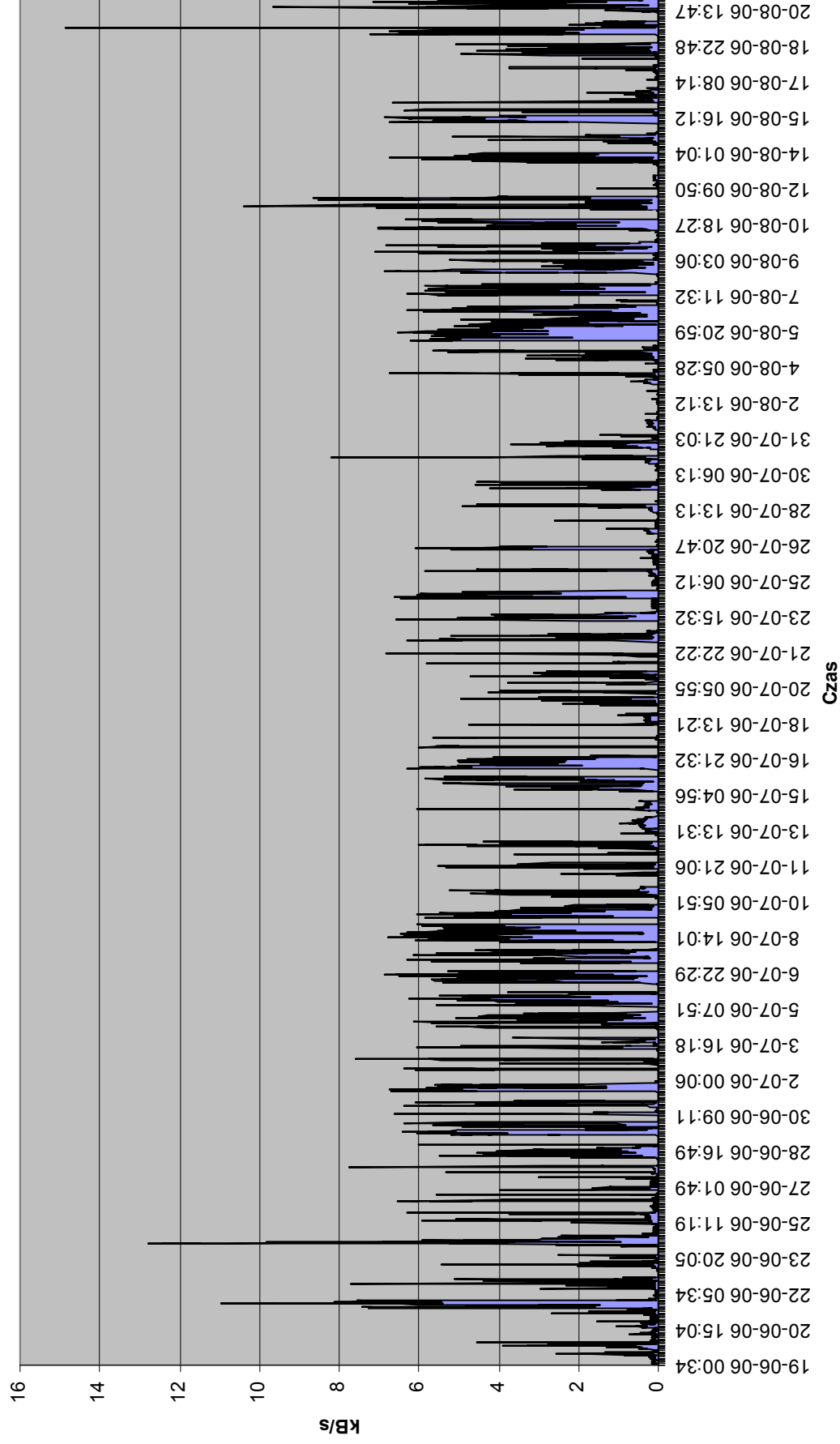
Rysunek 93: Statystyka odbierania pakietów TCP (przebieg miesięczny). Źródło: opracowanie własne.



Rysunek 94: Statystyka wysyłania pakietów TCP na port 80 (przebieg miesięczny). Źródło: opracowanie własne.

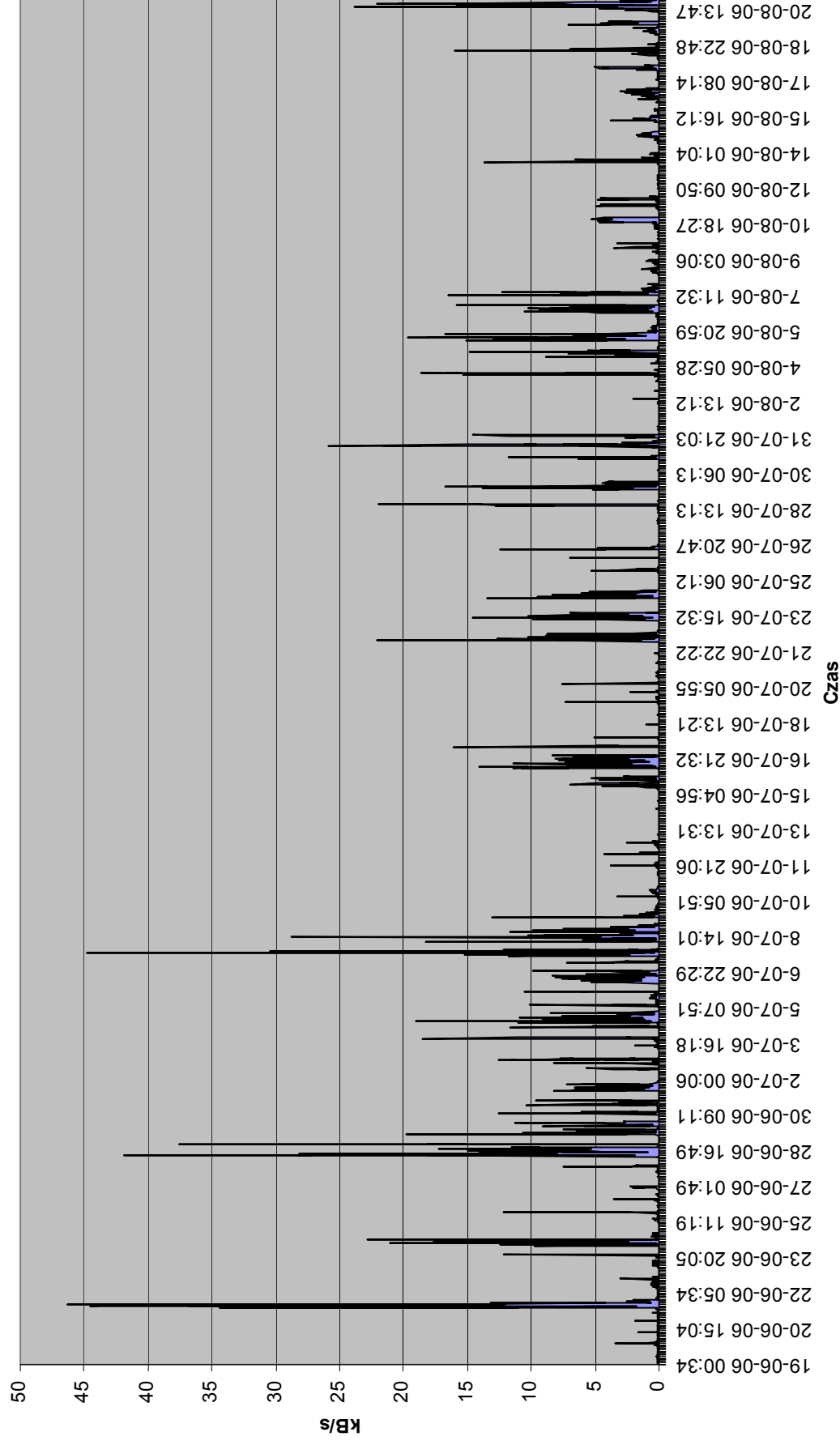


Rysunek 95: Statystyka odbierania pakietów TCP z portu 80 (przebieg miesięczny). Źródło: opracowanie własne.

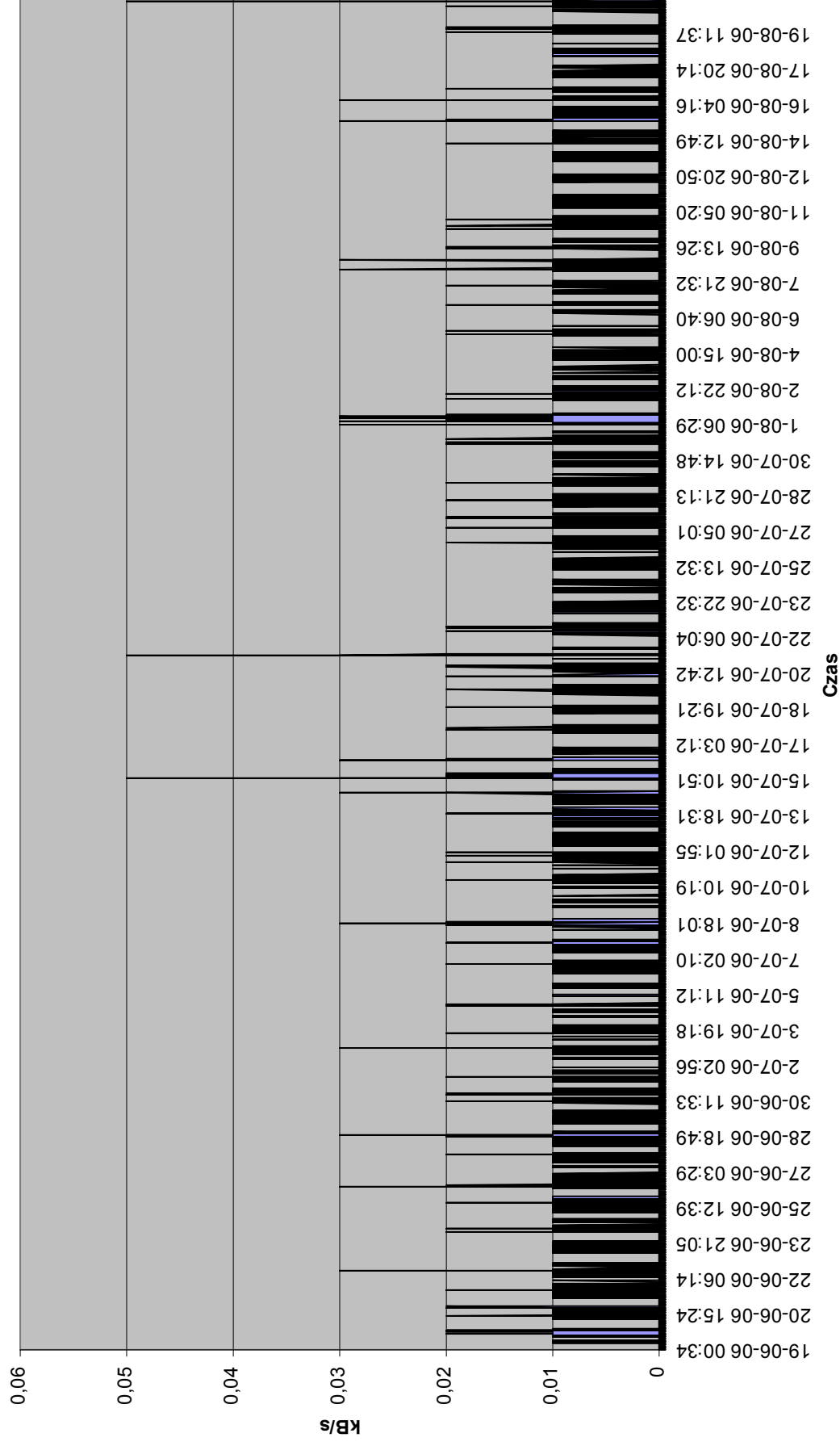


Rysunek 96: Statystyka wysyłania pakietów UDP (przebieg miesięczny). Źródło: opracowanie własne.

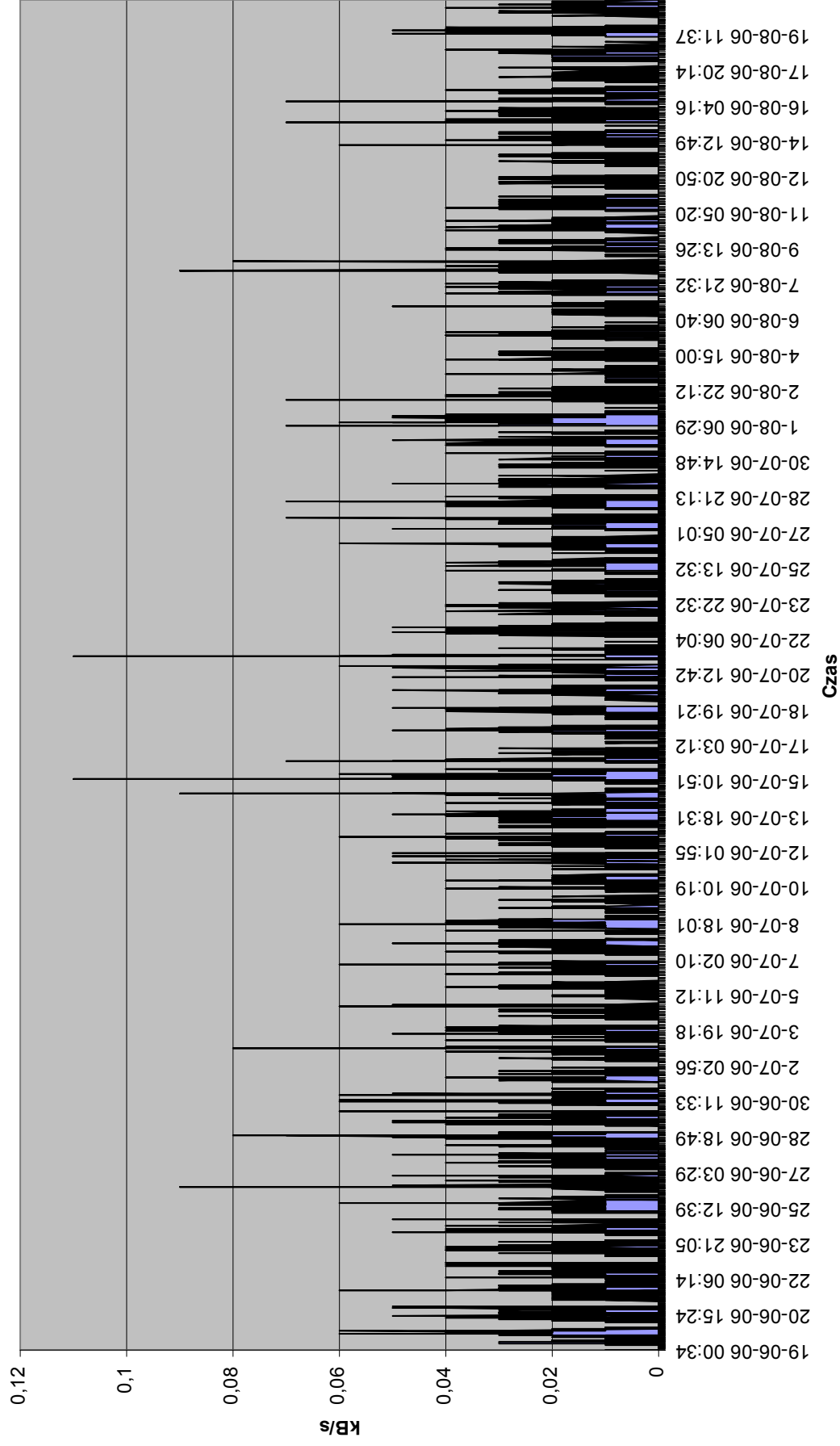




Rysunek 97: Statystyka odbierania pakietów UDP (przebieg miesięczny). Źródło: opracowanie własne.

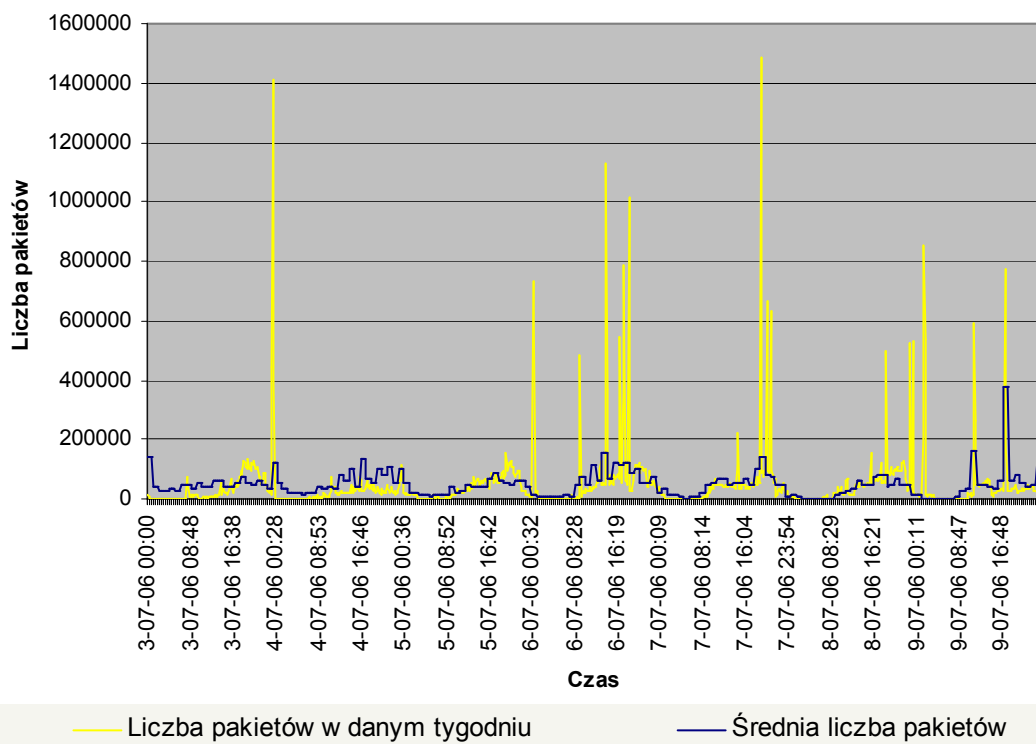


Rysunek 98: Statystyka wysyłania pakietów UDP na port 53 (przebieg miesięczny). Źródło: opracowanie własne.

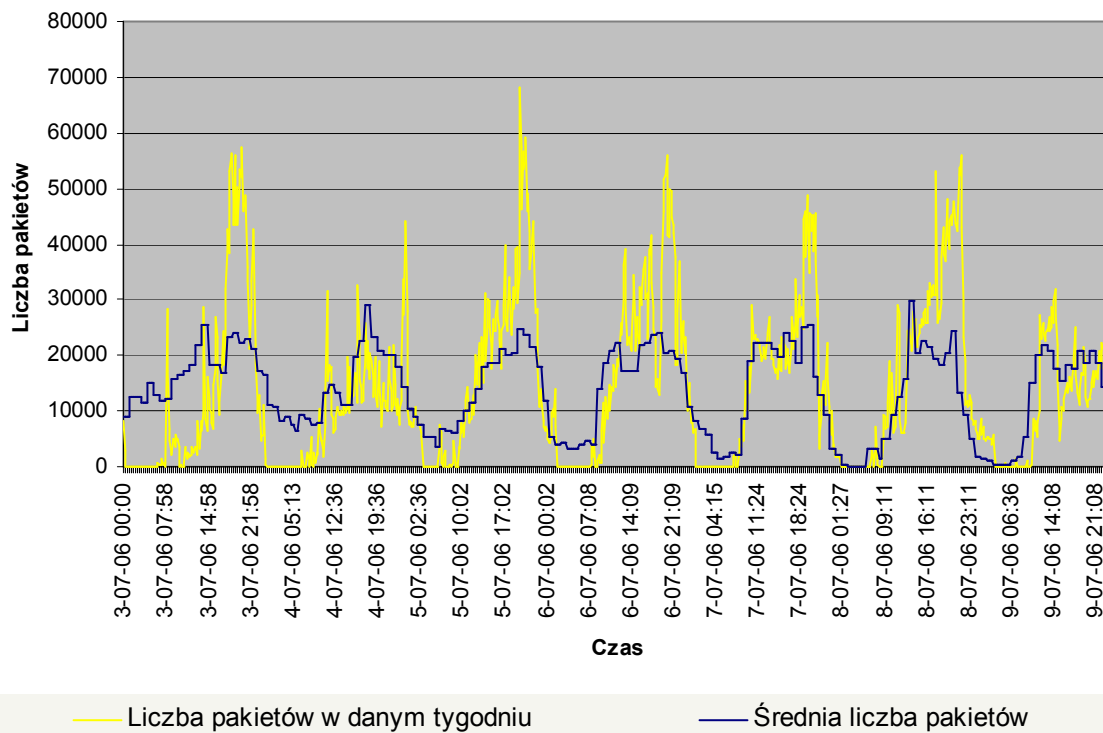


Rysunek 99: Statystyka odbierania pakietów UDP z portu 53 (przebieg miesięczny). Źródło: opracowanie własne.

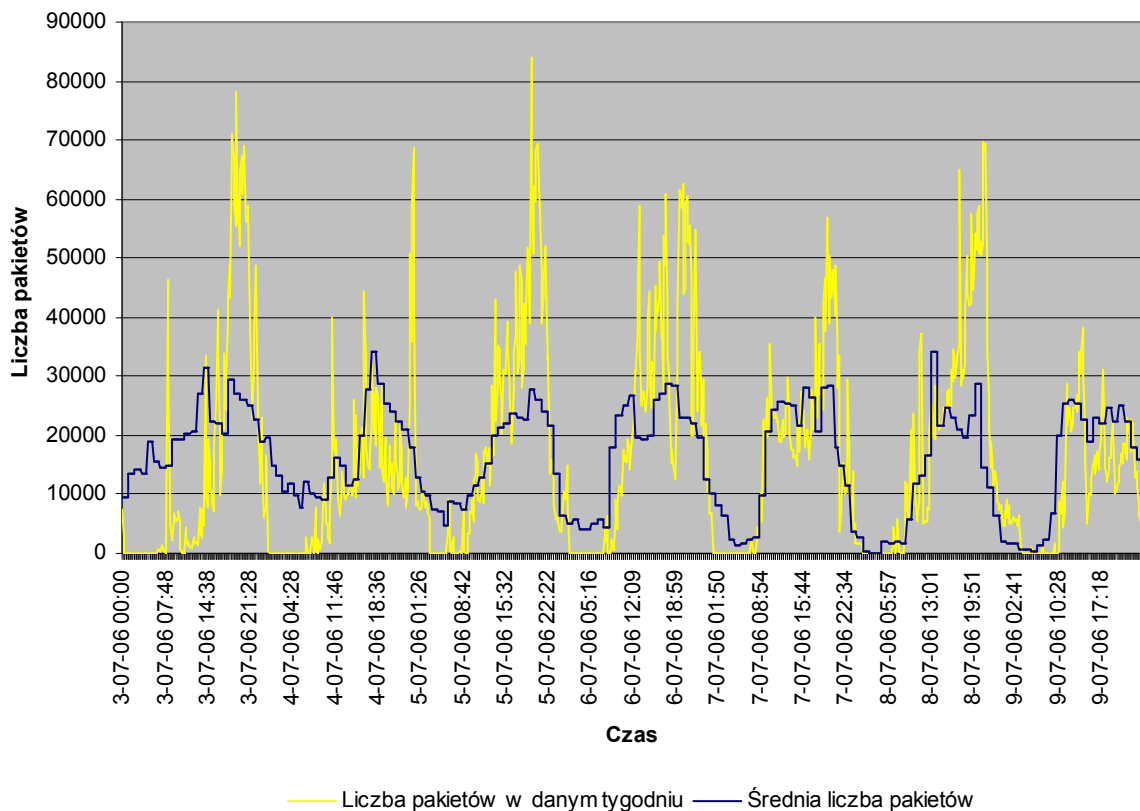
## Załącznik 2: Porównanie wyników i wartości średniej.



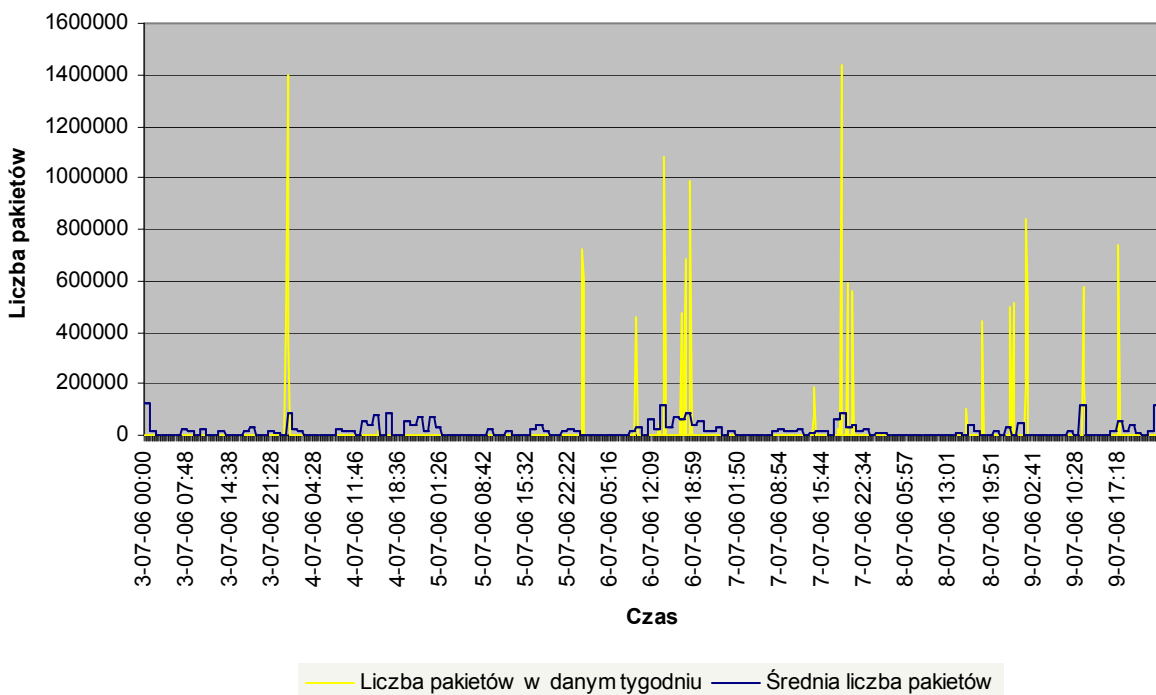
Rysunek 100: Statystyka pakietów TCP. Źródło: opracowanie własne.



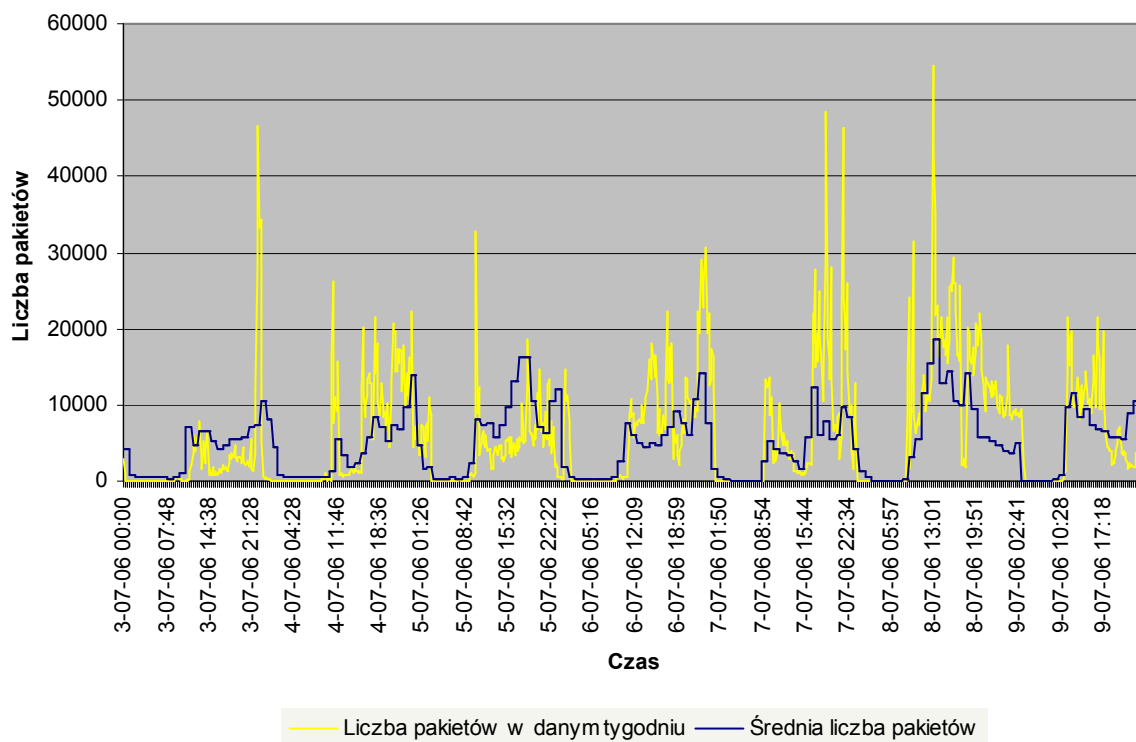
Rysunek 101: Statystyka wysyłanych pakietów TCP. Źródło: opracowanie własne.



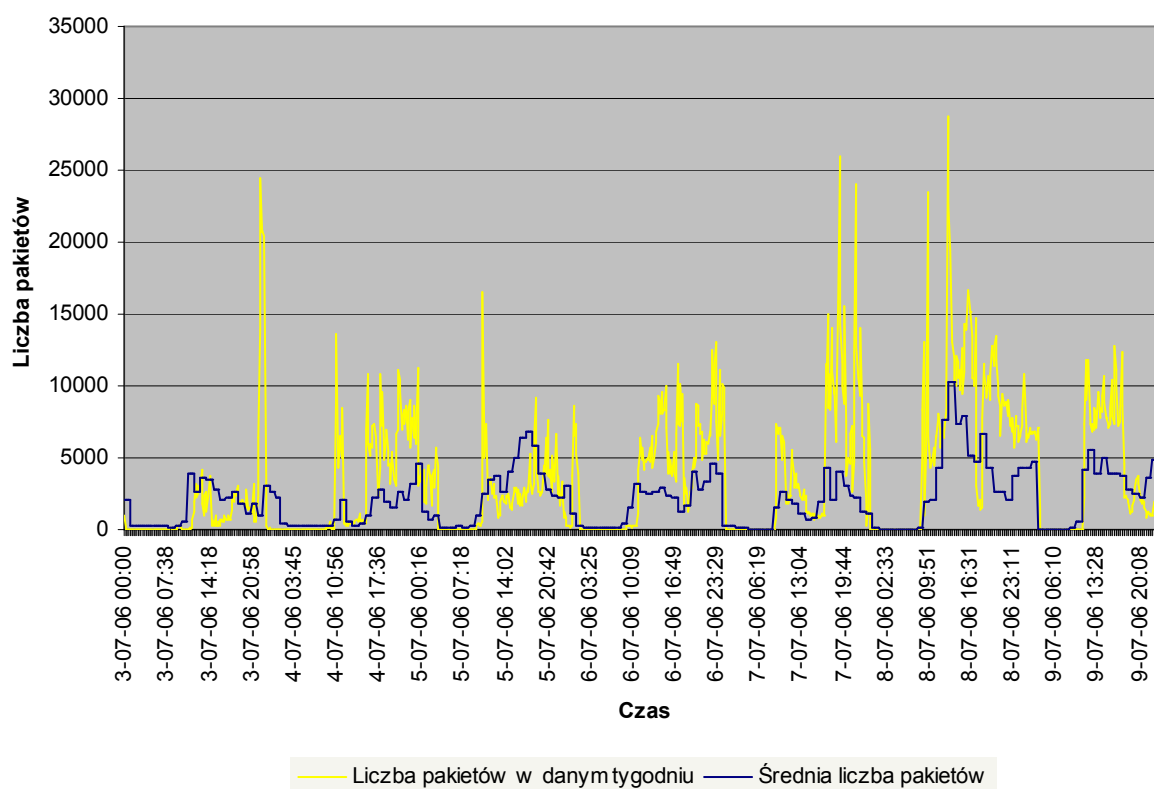
**Rysunek 102: Statystyka odebranych pakietów TCP. Źródło: opracowanie własne.**



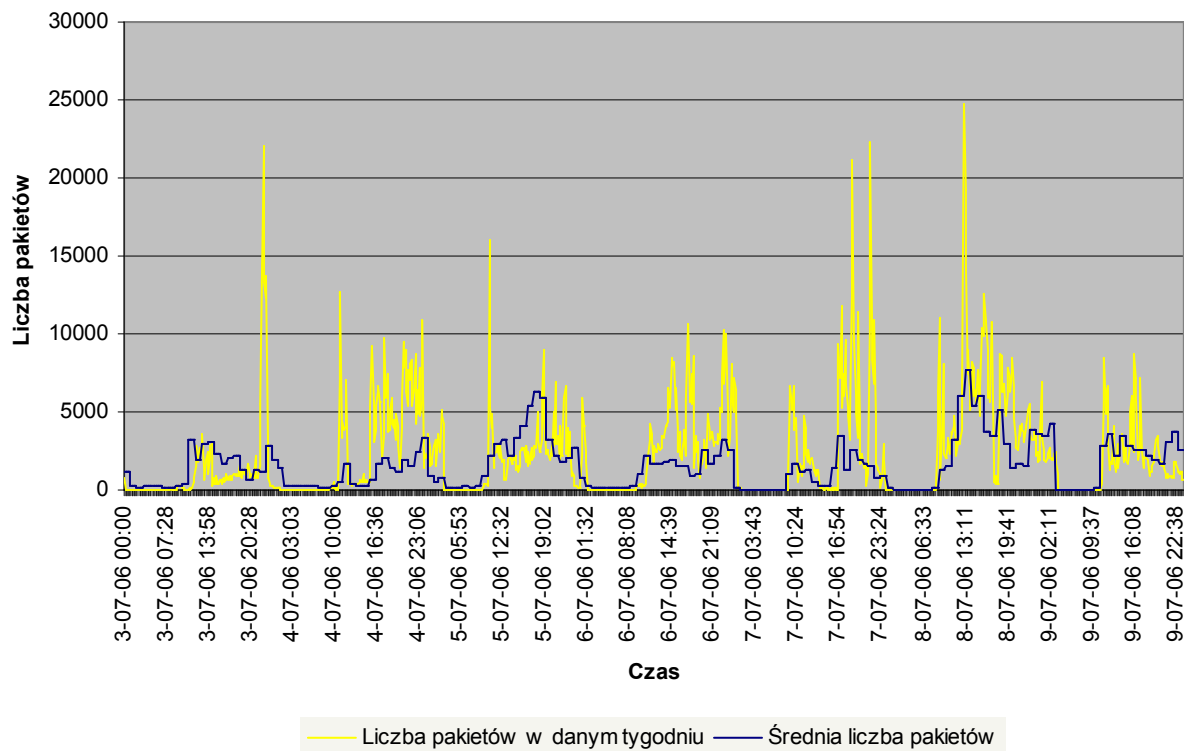
**Rysunek 103: Statystyka pakietów TCP wewnątrz sieci LAN. Źródło: opracowanie własne.**



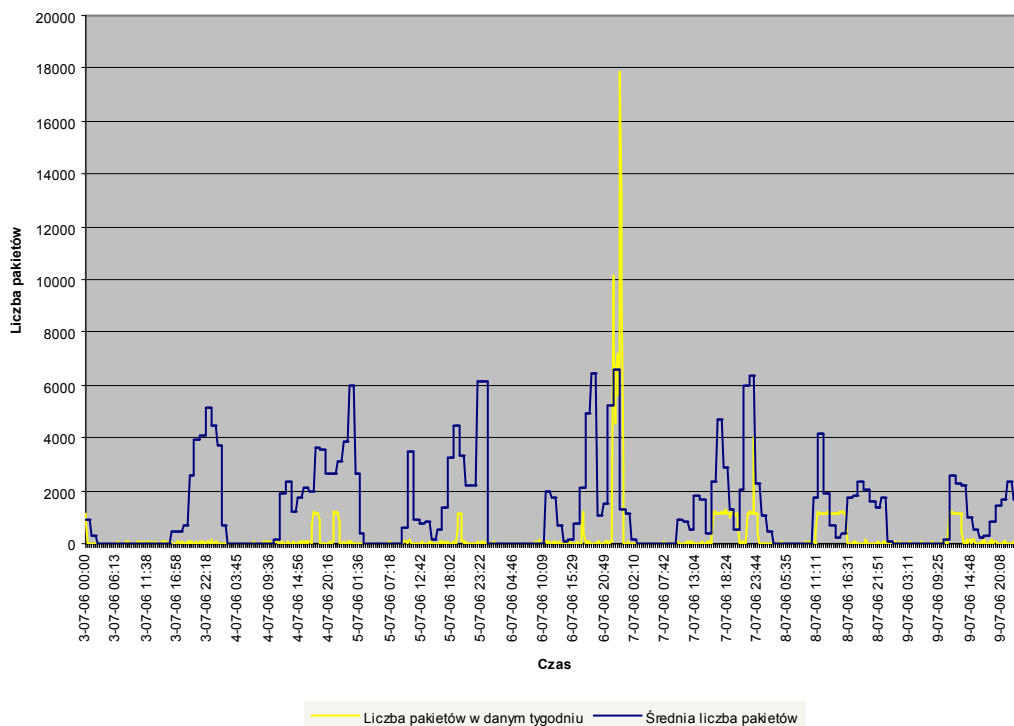
**Rysunek 104: Statystyka pakietów UDP. Źródło: opracowanie własne.**



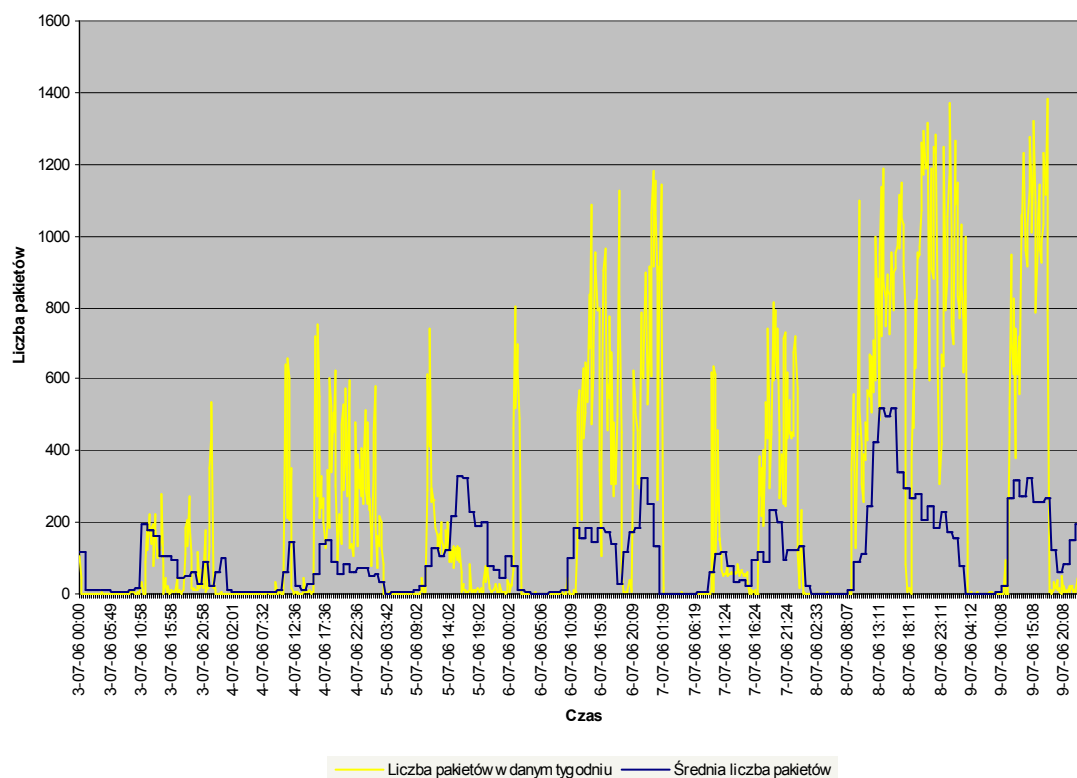
**Rysunek 105: Statystyka wysłanych pakietów UDP. Źródło: opracowanie własne.**



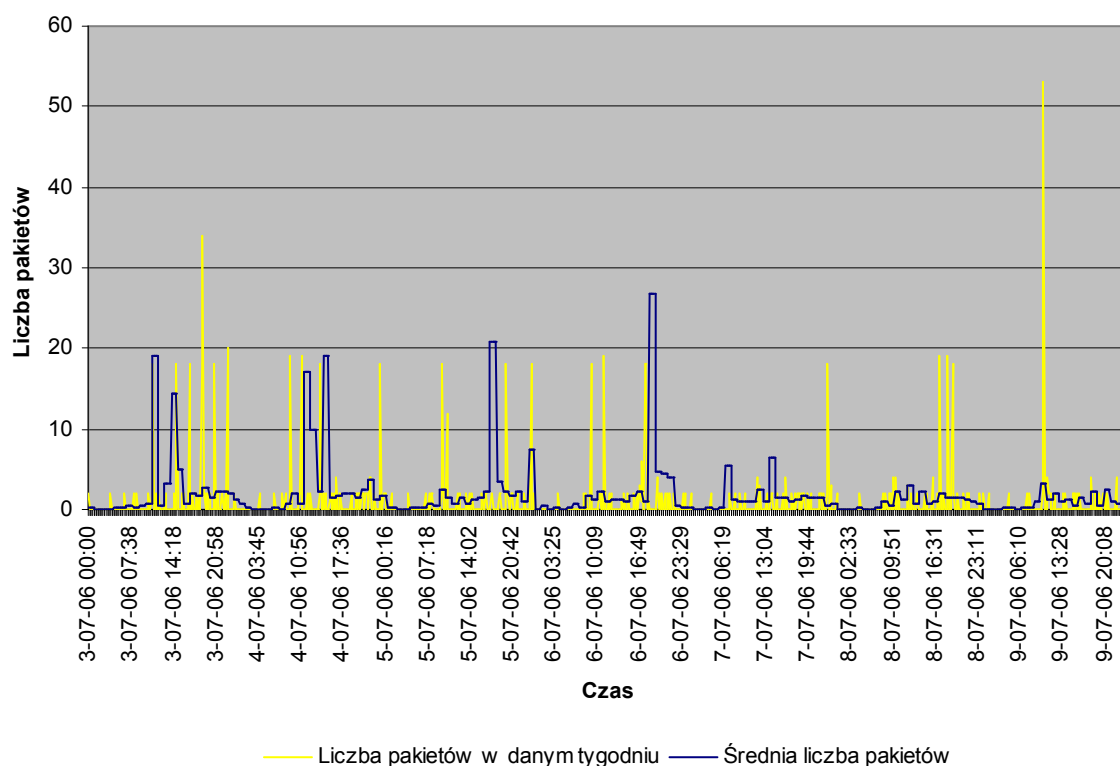
**Rysunek 106: Statystyka odebranych pakietów UDP. Źródło: opracowanie własne.**



**Rysunek 107: Statystyka pakietów UDP wewnątrz sieci LAN. Źródło: opracowanie własne.**

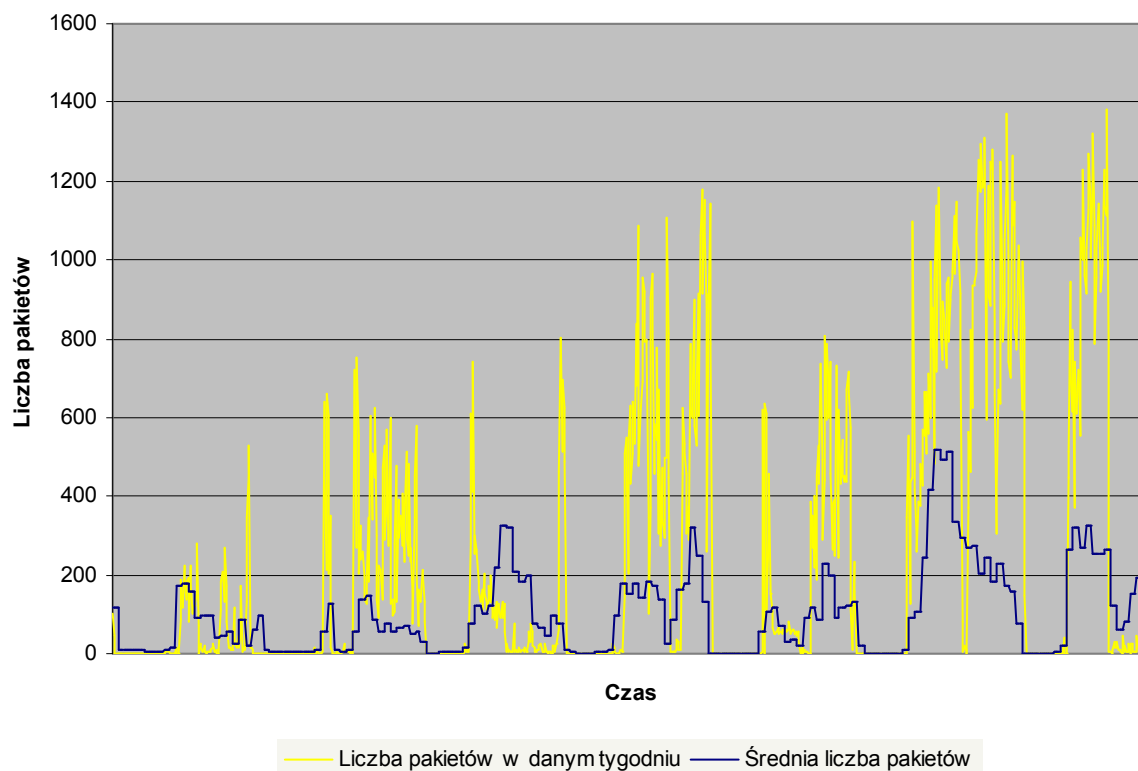


Rysunek 108: Statystyka pakietów ICMP. Źródło: opracowanie własne.

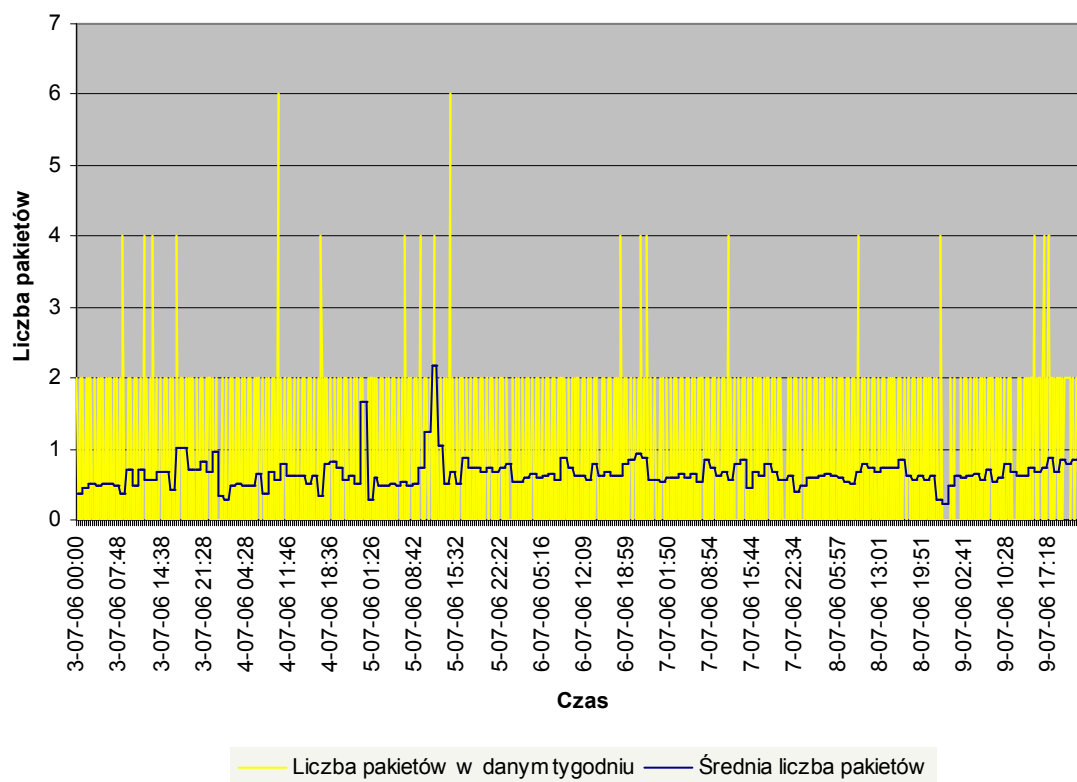


Rysunek 109: Statystyka wysłanych pakietów ICMP. Źródło: opracowanie własne.

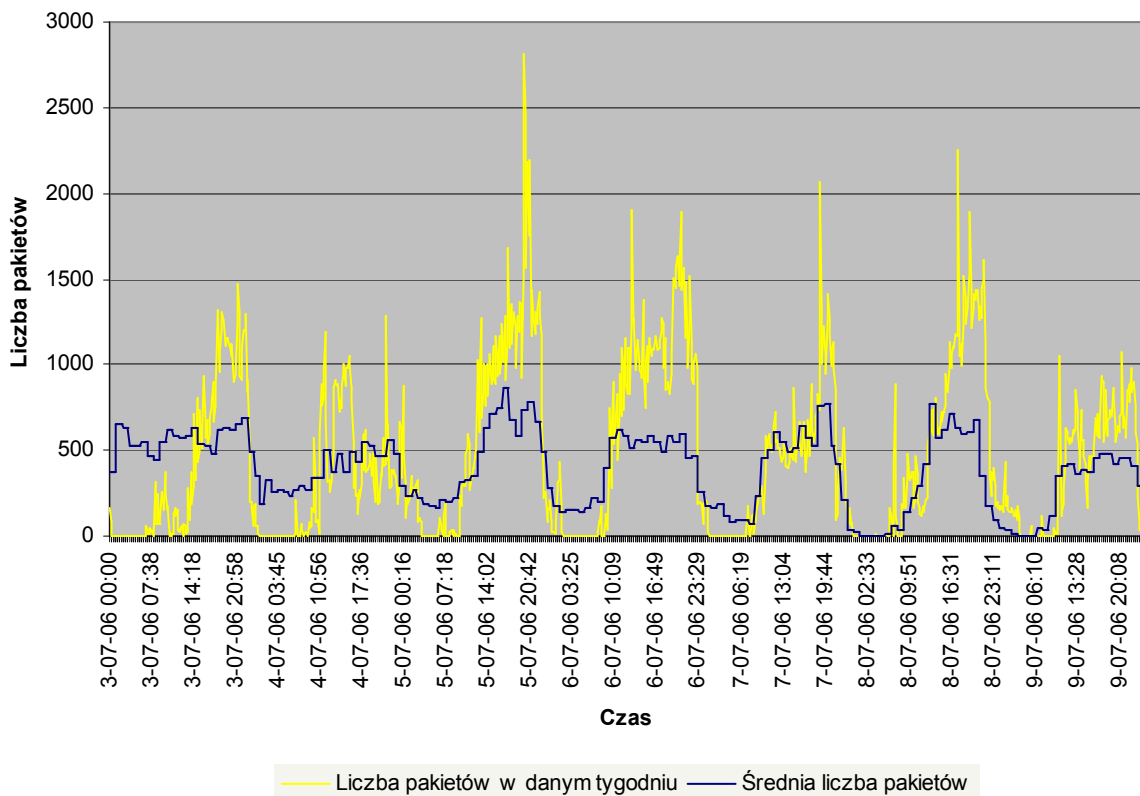




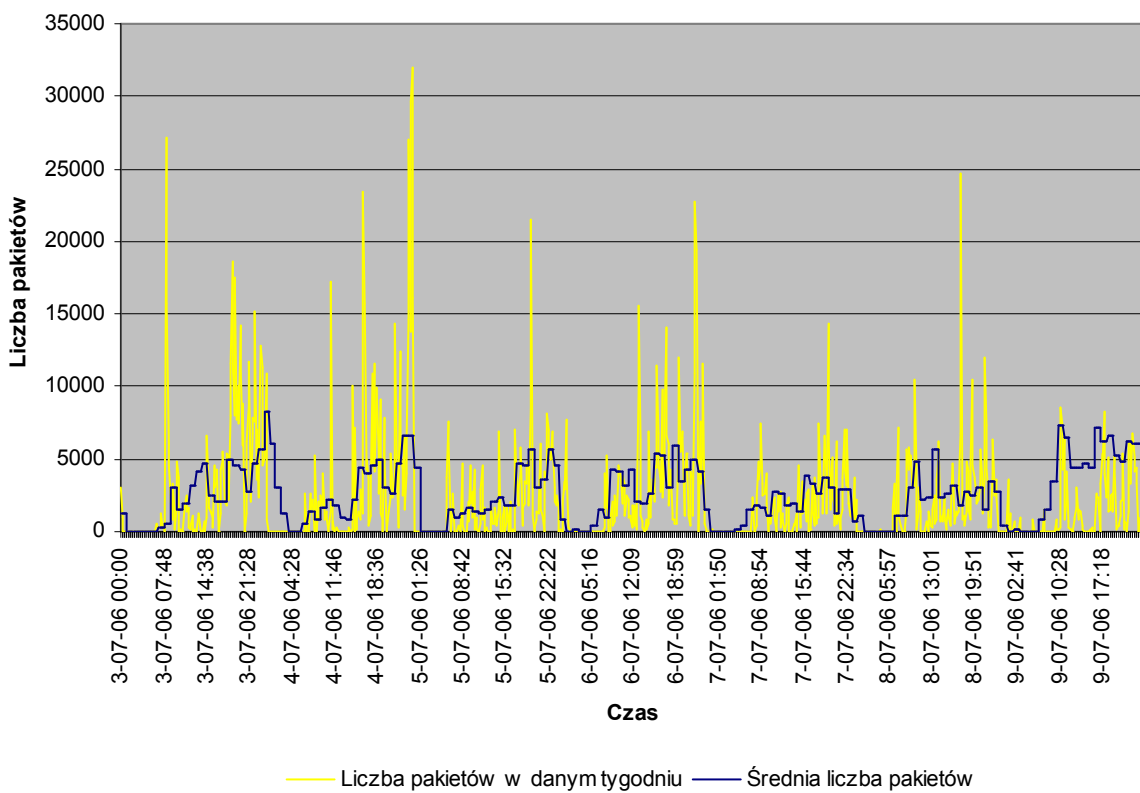
Rysunek 110: Statystyka odebranych pakietów ICMP. Źródło: opracowanie własne.



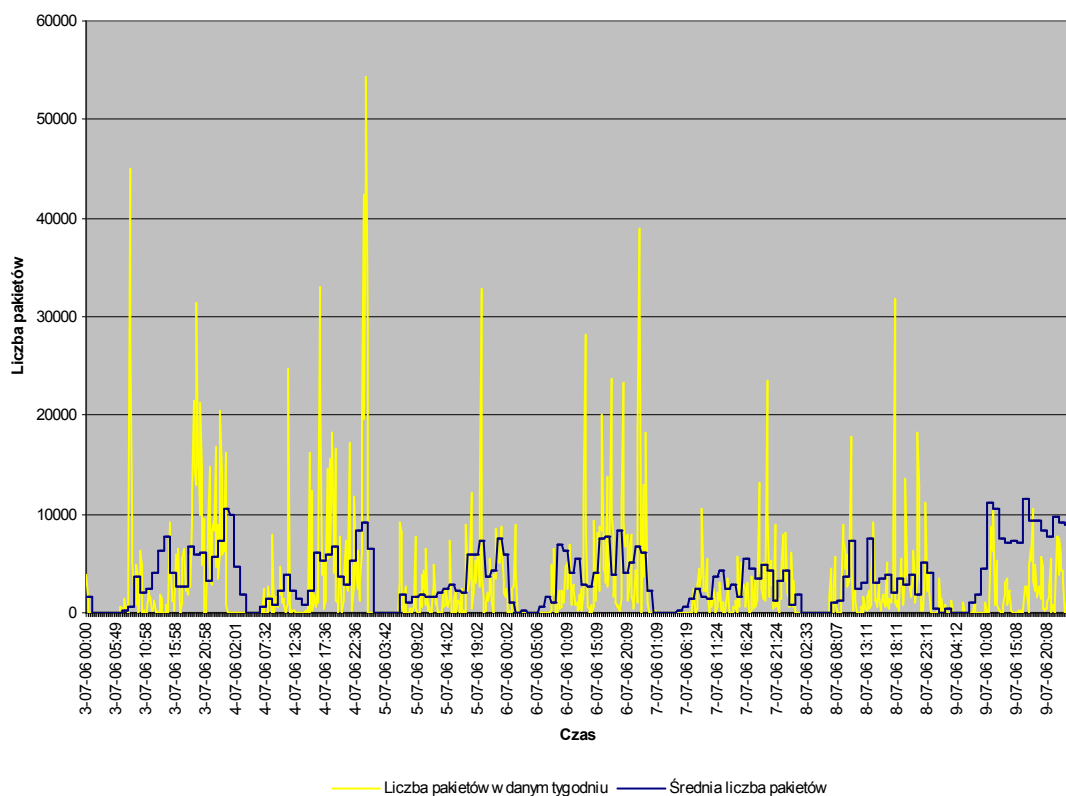
Rysunek 111: Statystyka pakietów ICMP wewnątrz sieci LAN. Źródło: opracowanie własne.



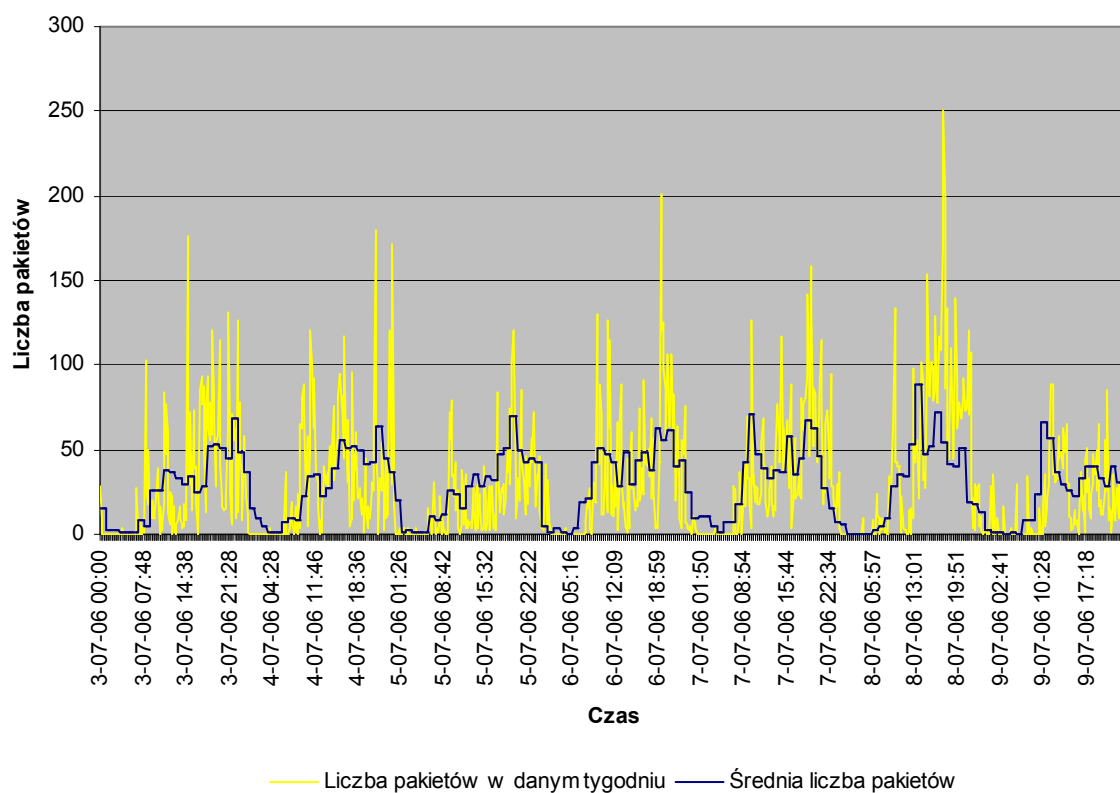
Rysunek 112: Statystyka nowych połączeń. Źródło: opracowanie własne.



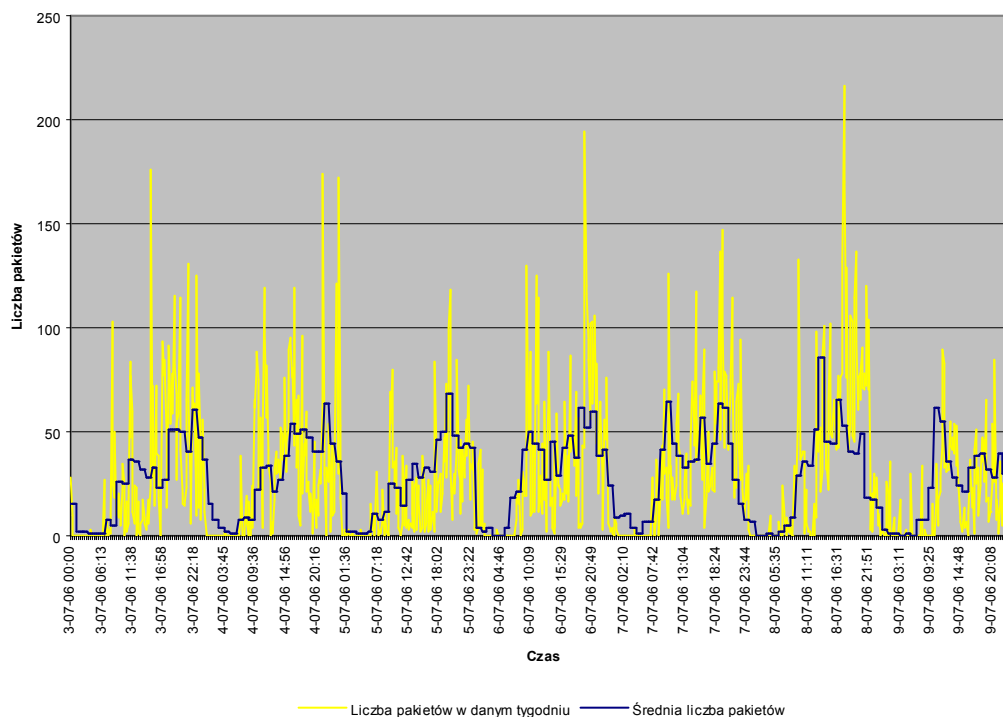
Rysunek 113: Statystyka wysłanych pakietów TCP na port 80 (WWW). Źródło: opracowanie własne.



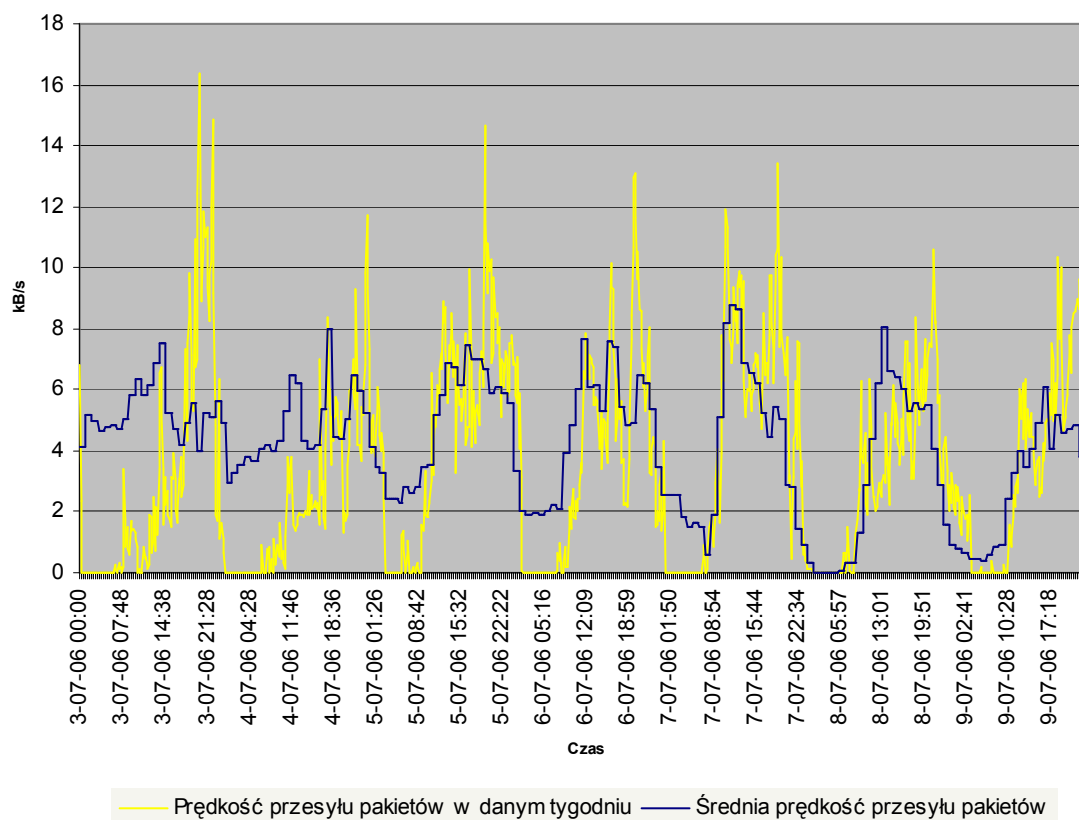
Rysunek 114: Statystyka odebranych pakietów TCP na port 80 (WWW). Źródło: opracowanie własne.



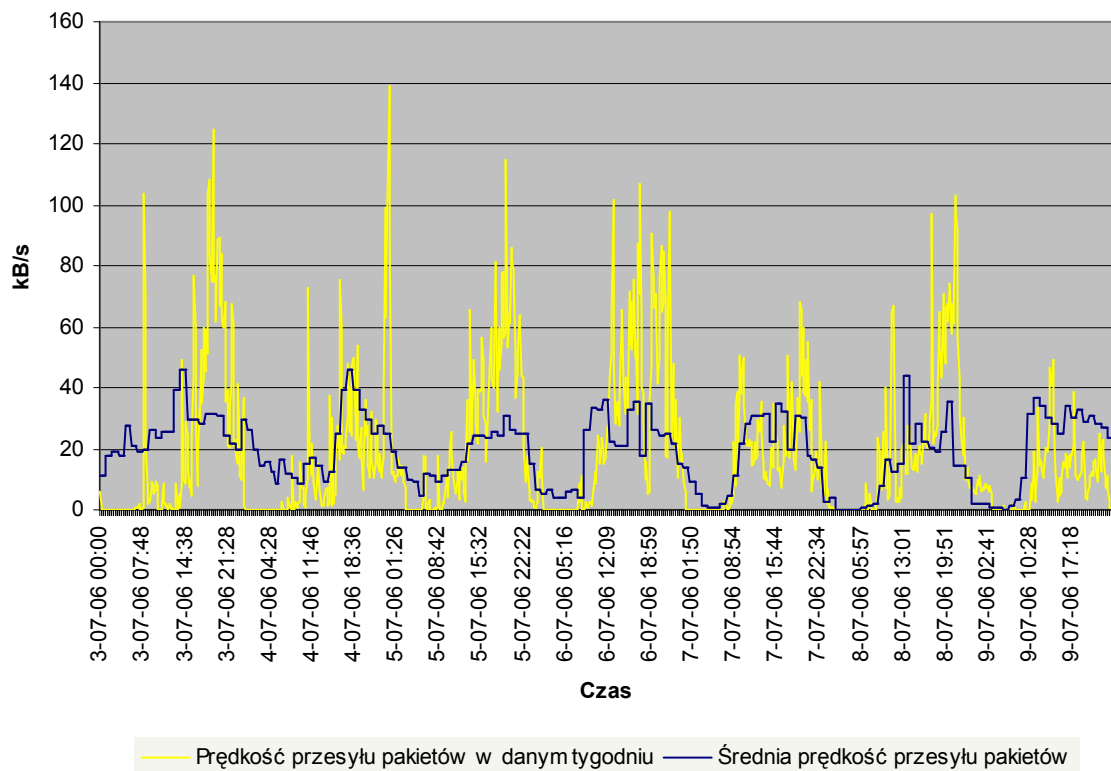
Rysunek 115: Statystyka wysłanych pakietów UDP na port 53 (DNS). Źródło: opracowanie własne.



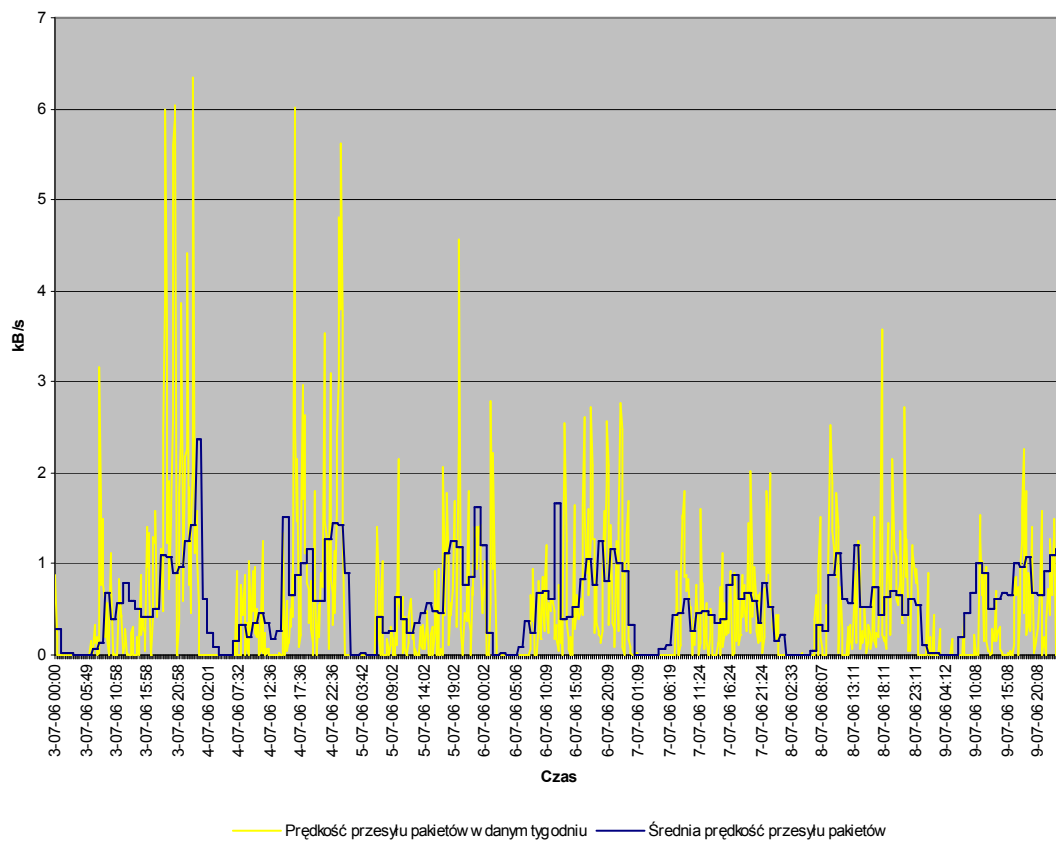
**Rysunek 116: Statystyka odebranych pakietów UDP na port 53 (DNS). Źródło: opracowanie własne.**



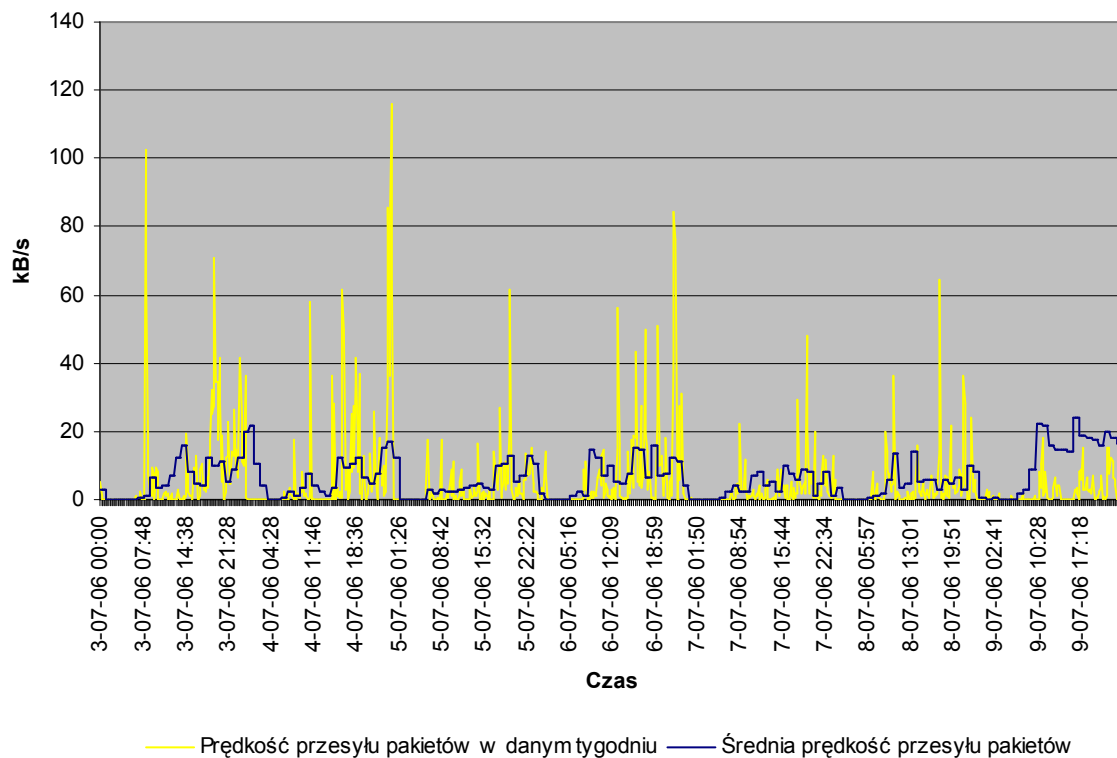
**Rysunek 117: Statystyka ruchu TCP (dane wysłane). Źródło: opracowanie własne.**



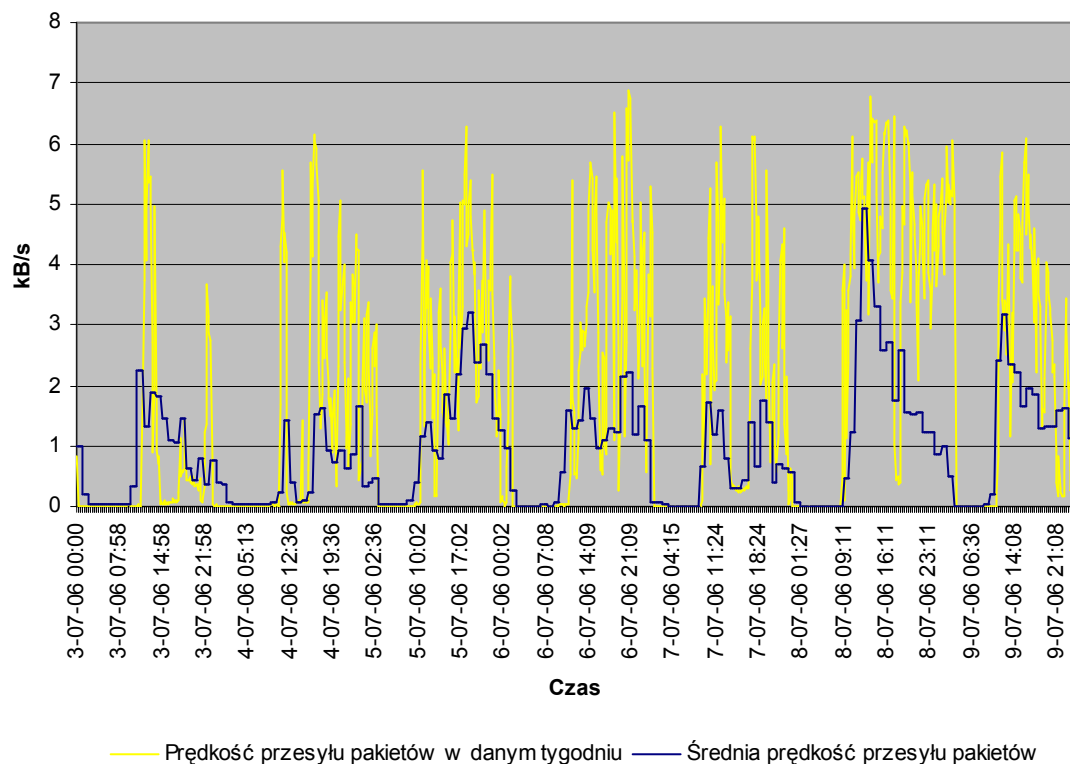
**Rysunek 118: Statystyka ruchu TCP (dane odebrane). Źródło: opracowanie własne.**



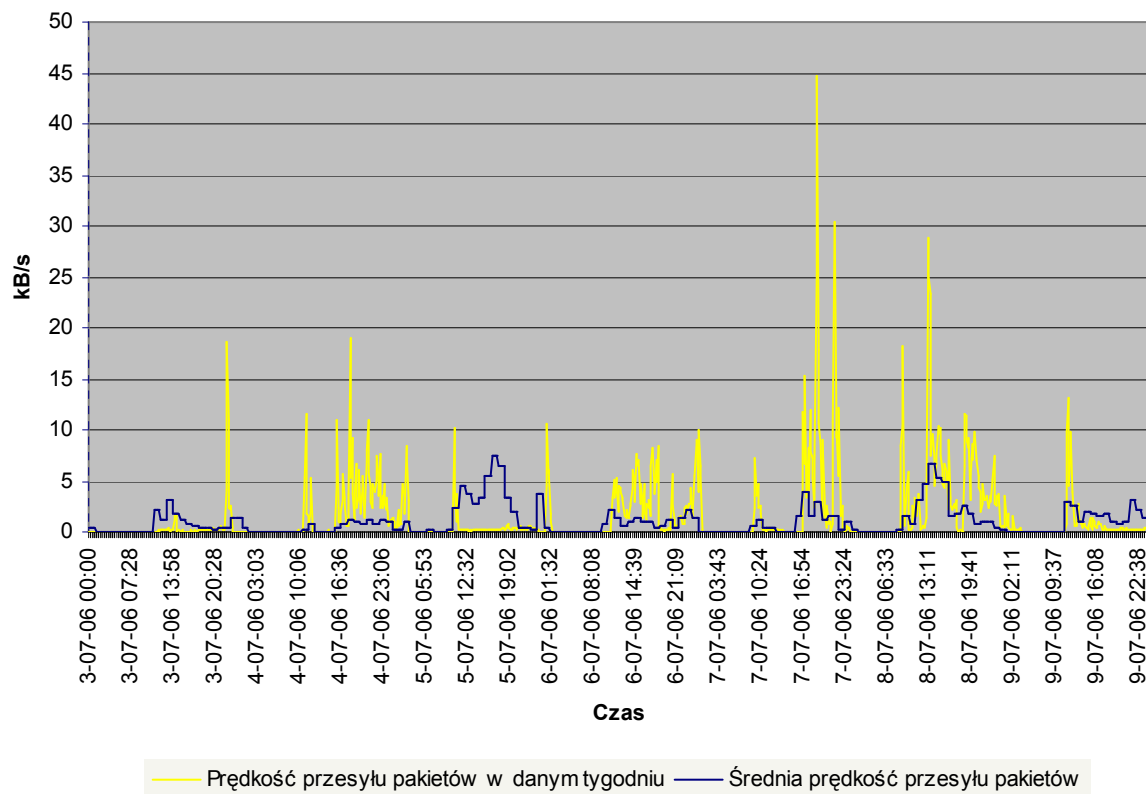
**Rysunek 119: Statystyka ruchu WWW (dane wysłane). Źródło: opracowanie własne.**



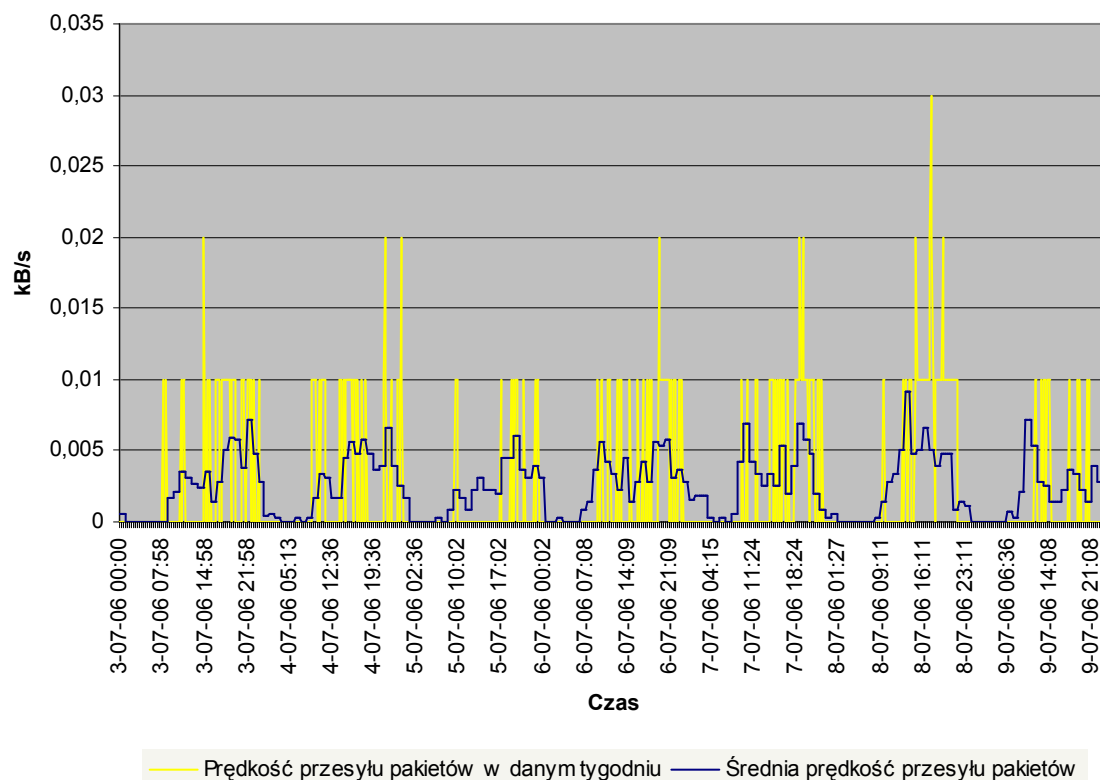
**Rysunek 120: Statystyka ruchu WWW (dane odebrane). Źródło: opracowanie własne.**



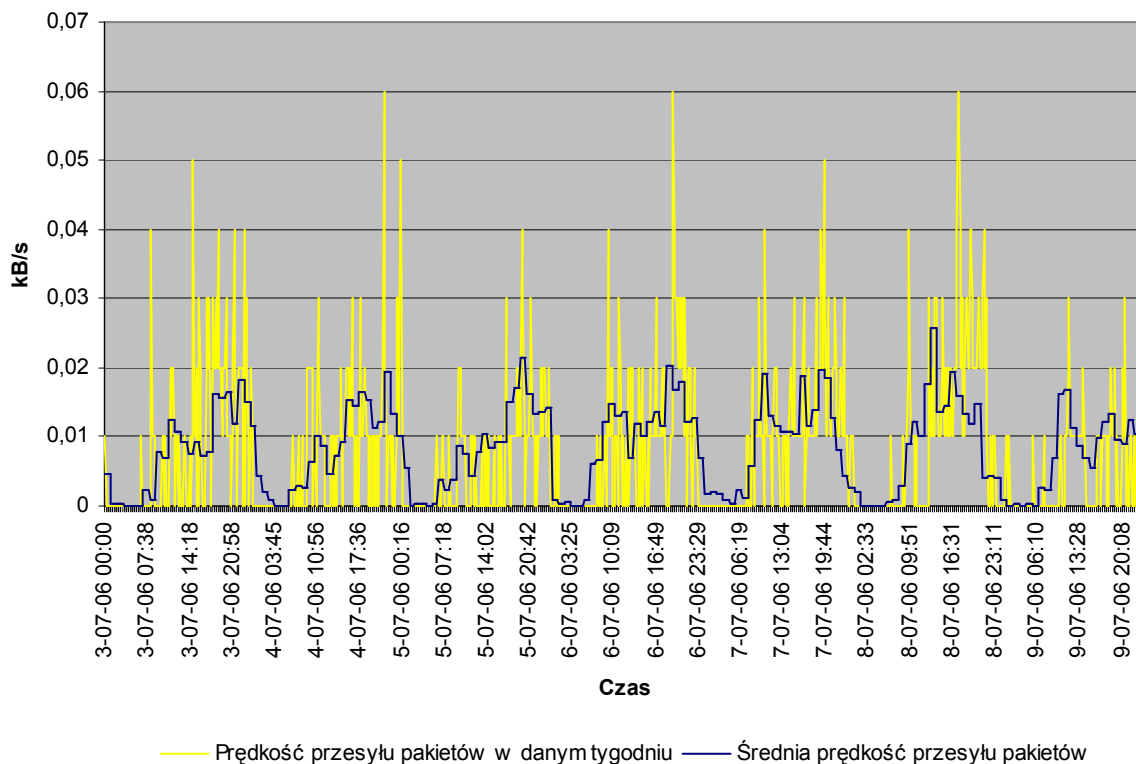
**Rysunek 121: Statystyka ruchu UDP (dane wysłane). Źródło: opracowanie własne.**



Rysunek 122: Statystyka ruchu UDP (dane odebrane). Źródło: opracowanie własne.

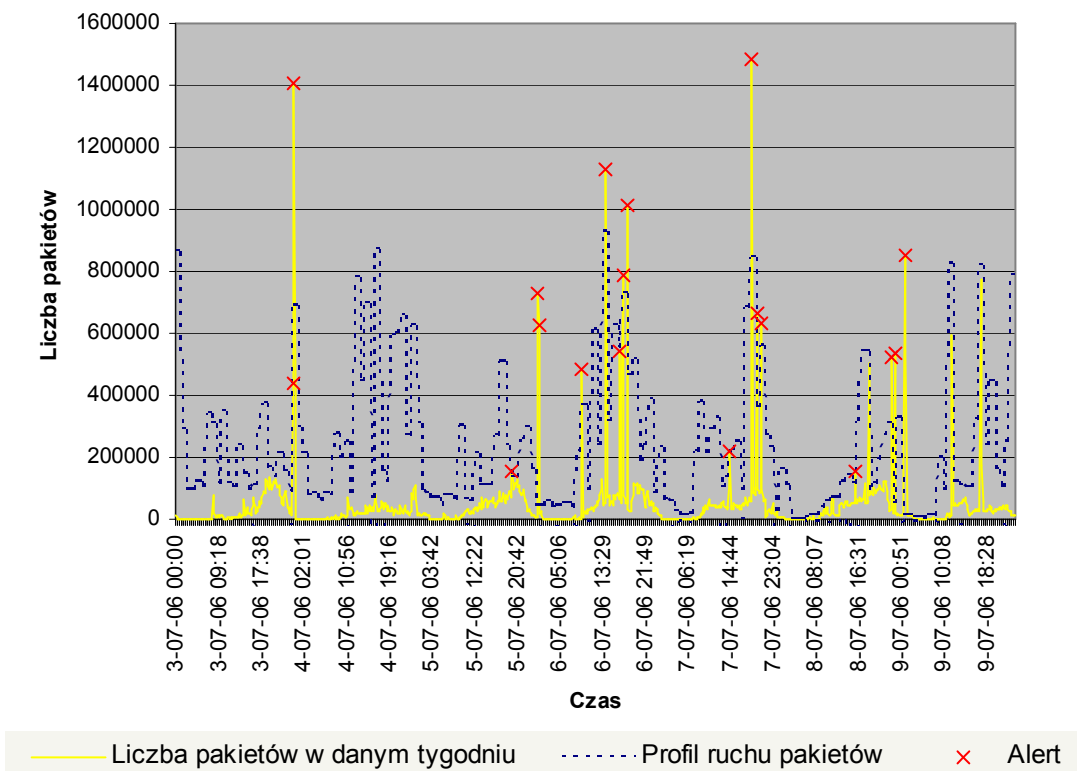


Rysunek 123: Statystyka ruchu UDP port 53 (dane wysłane). Źródło: opracowanie własne.



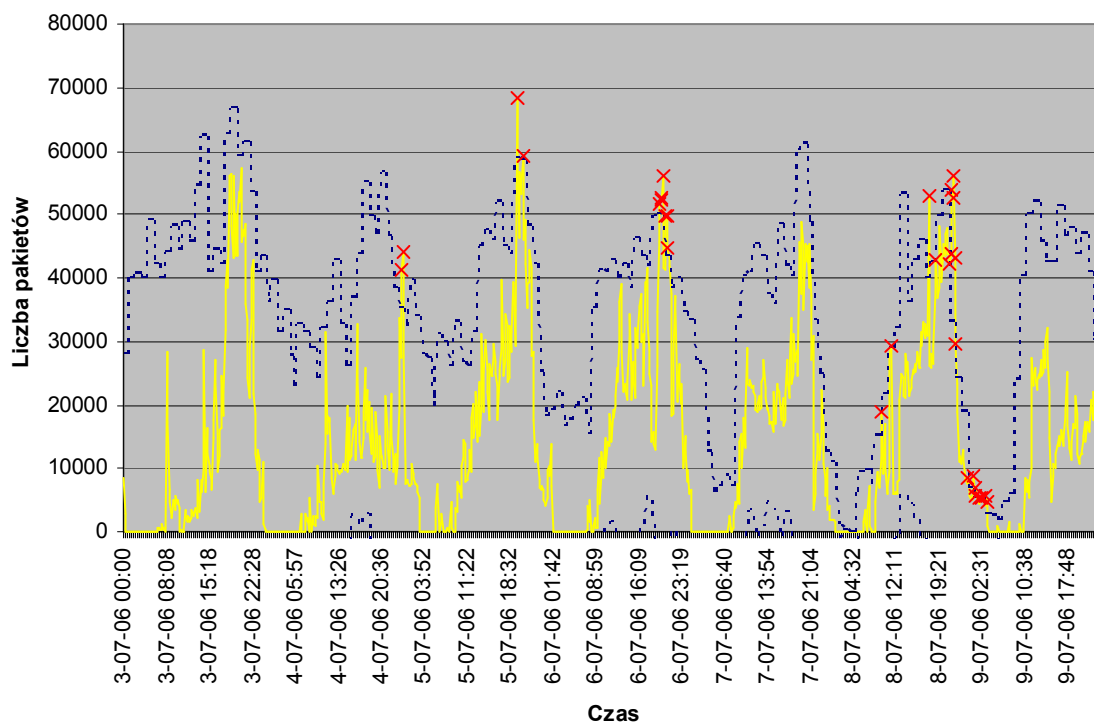
Rysunek 124: Statystyka ruchu UDP port 53 (dane odebrane). Źródło: opracowanie własne.

### Załącznik 3: Wykresy dla mnożnika sigmy równego 2.



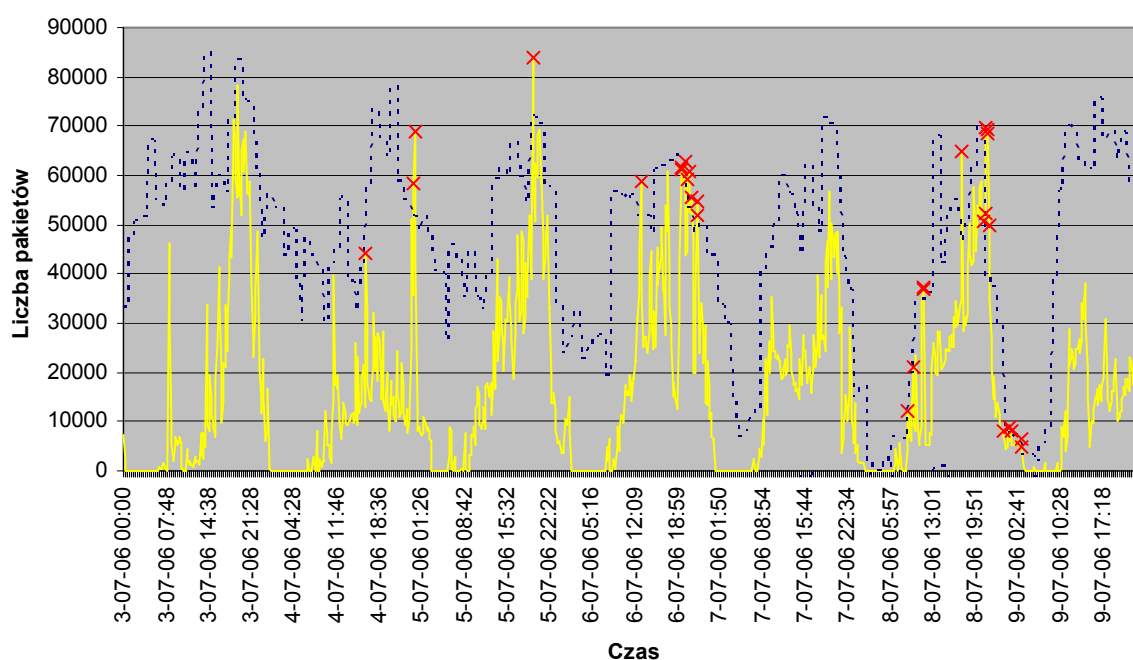
Rysunek 125: Statystyka alertów - ruch TCP. Źródło: opracowanie własne.





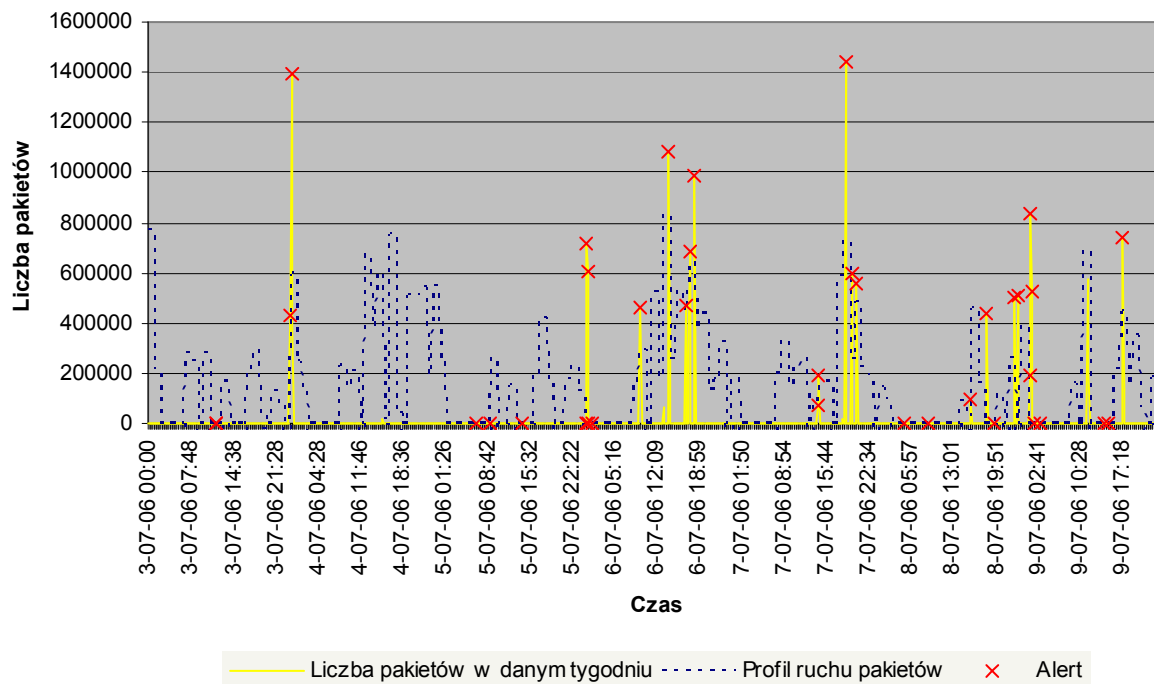
— Liczba pakietów w danym tygodniu    - - - - - Profil ruchu pakietów    x Alert

**Rysunek 126: Statystyka alertów – wysłane pakiety TCP. Źródło: opracowanie własne.**

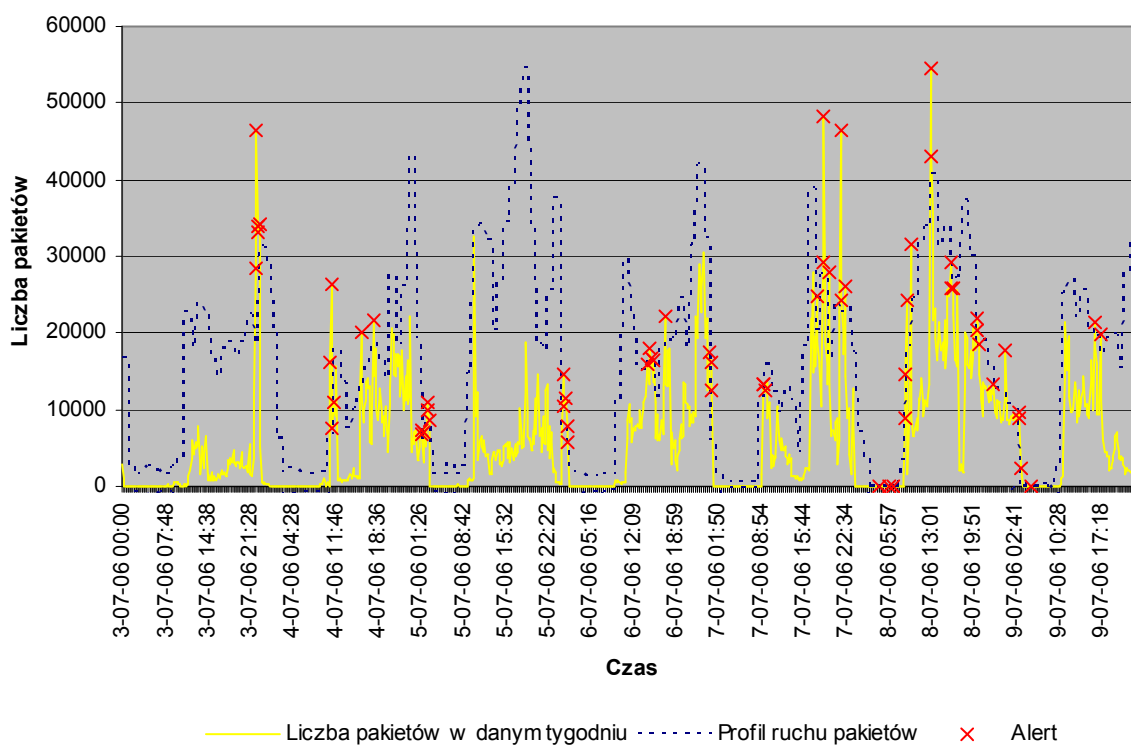


— Liczba pakietów w danym tygodniu    - - - - - Profil ruchu pakietów    x Alert

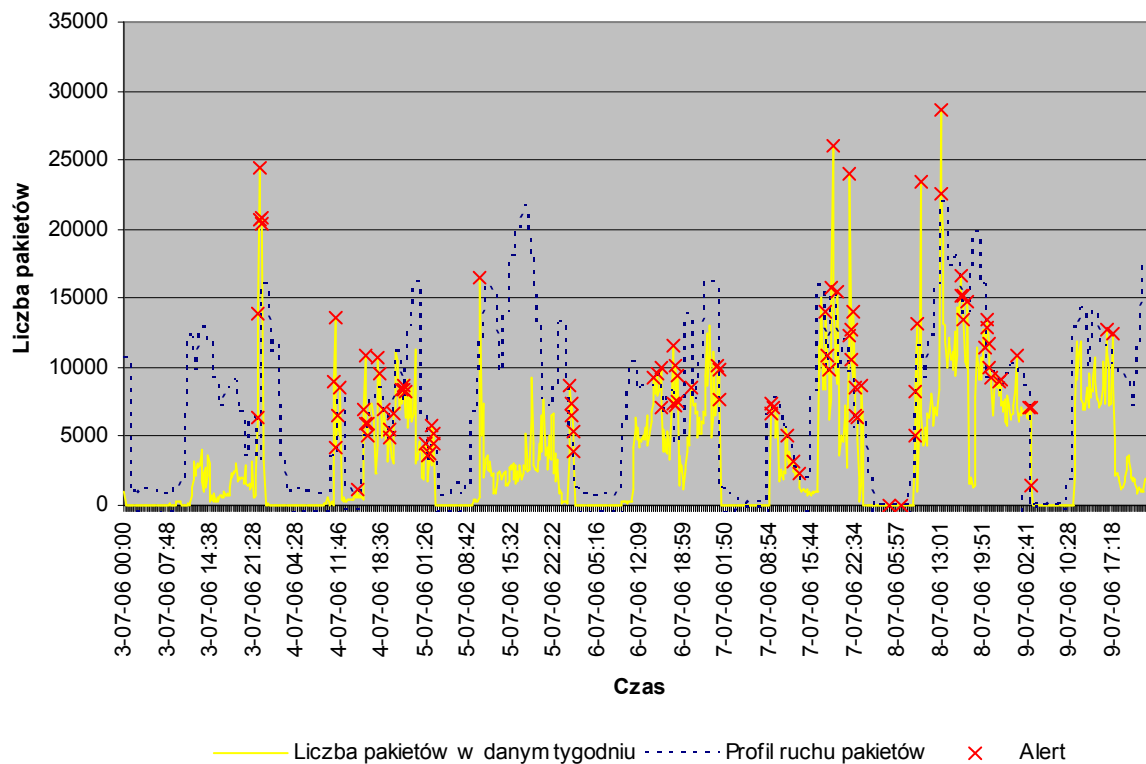
**Rysunek 127: Statystyka alertów – odebrane pakiety TCP. Źródło: opracowanie własne.**



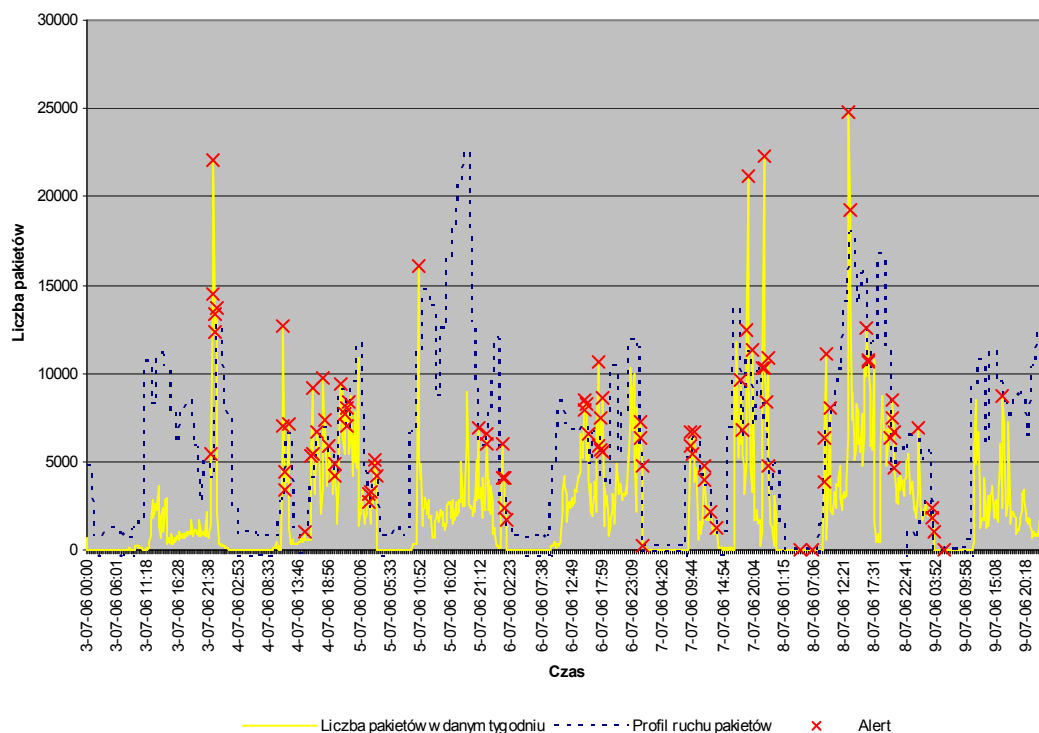
Rysunek 128: Statystyka alertów – pakiety TCP wewnątrz sieci LAN. Źródło: opracowanie własne.



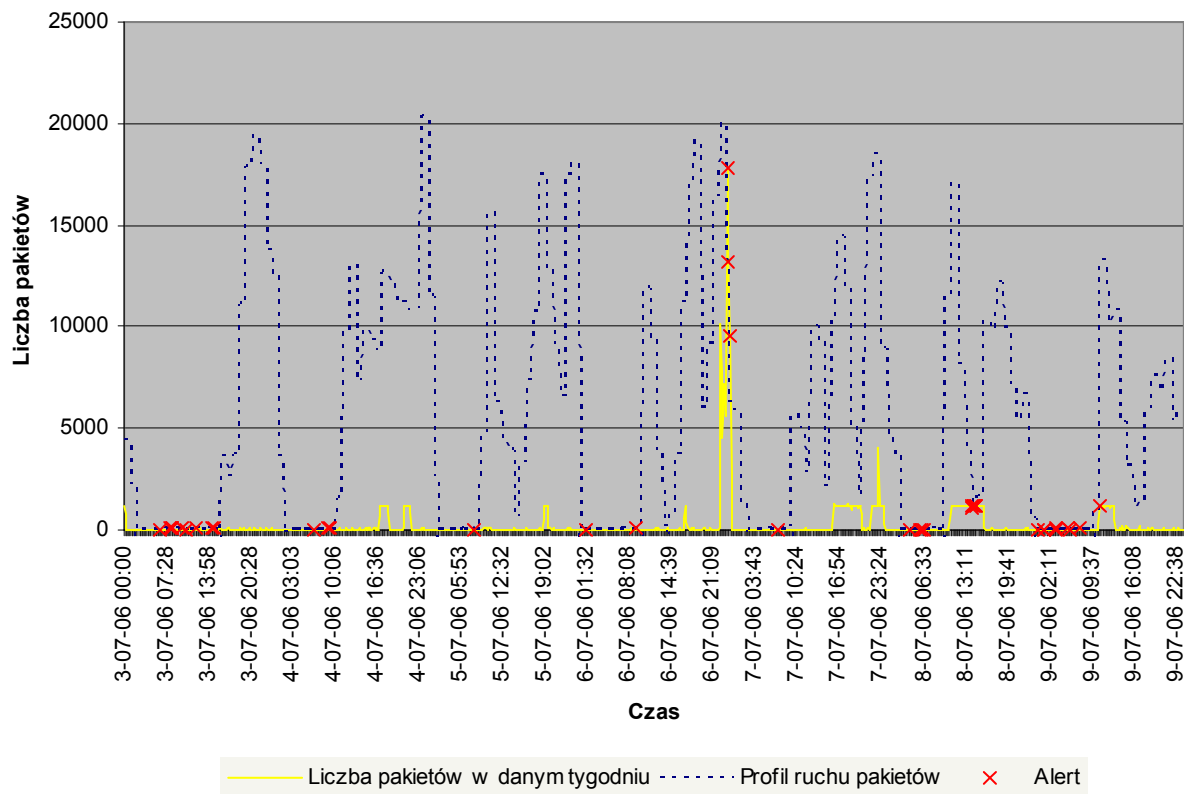
Rysunek 129: Statystyka alertów – pakiety UDP. Źródło: opracowanie własne.



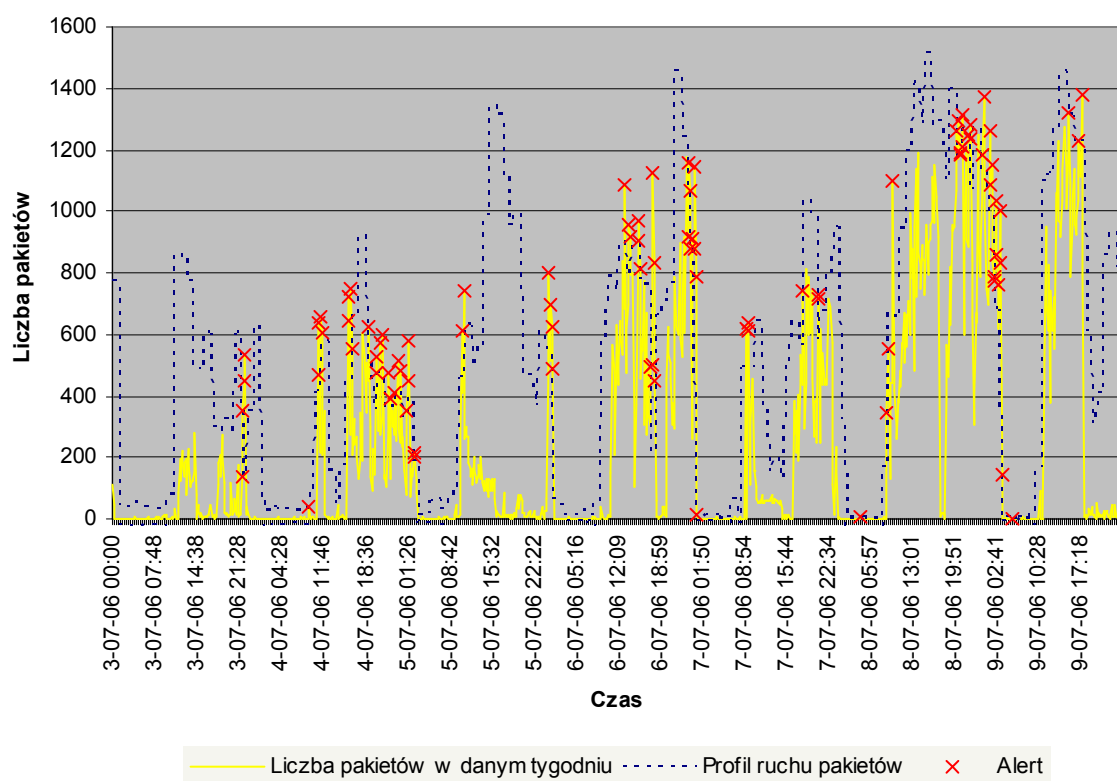
Rysunek 130: Statystyka alertów – wysłane pakiety UDP. Źródło: opracowanie własne.



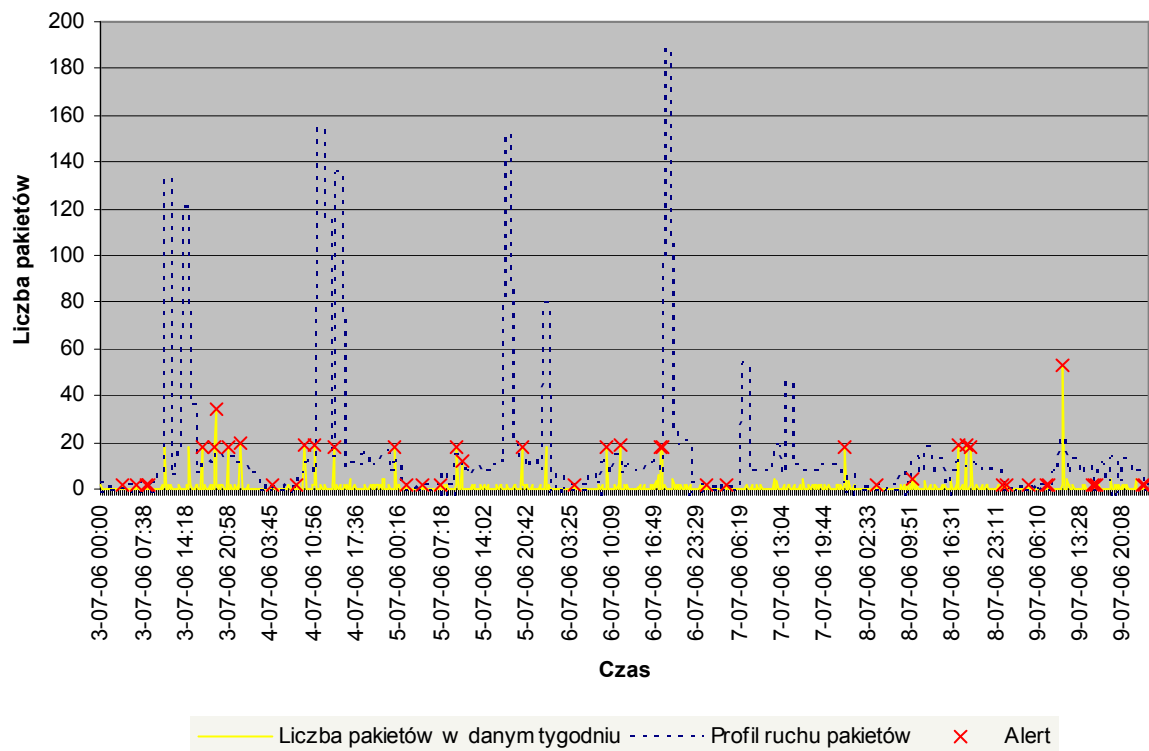
Rysunek 131: Statystyka alertów – odebrane pakiety UDP. Źródło: opracowanie własne.



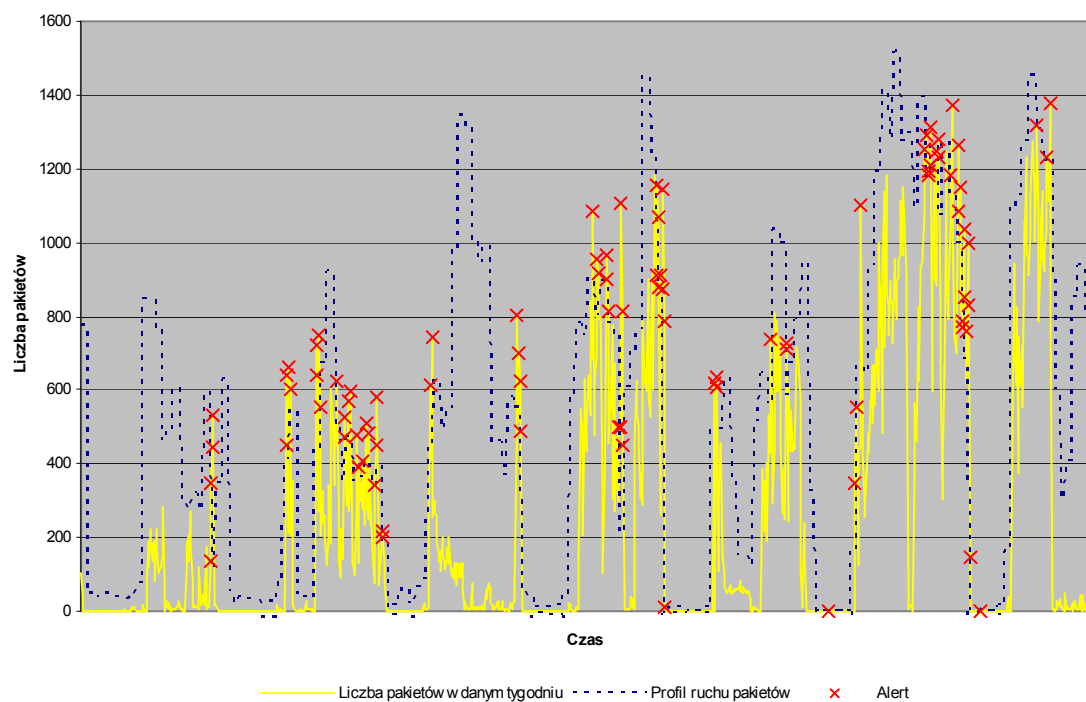
Rysunek 132: Statystyka alertów – pakiety UDP wewnątrz sieci LAN. Źródło: opracowanie własne.



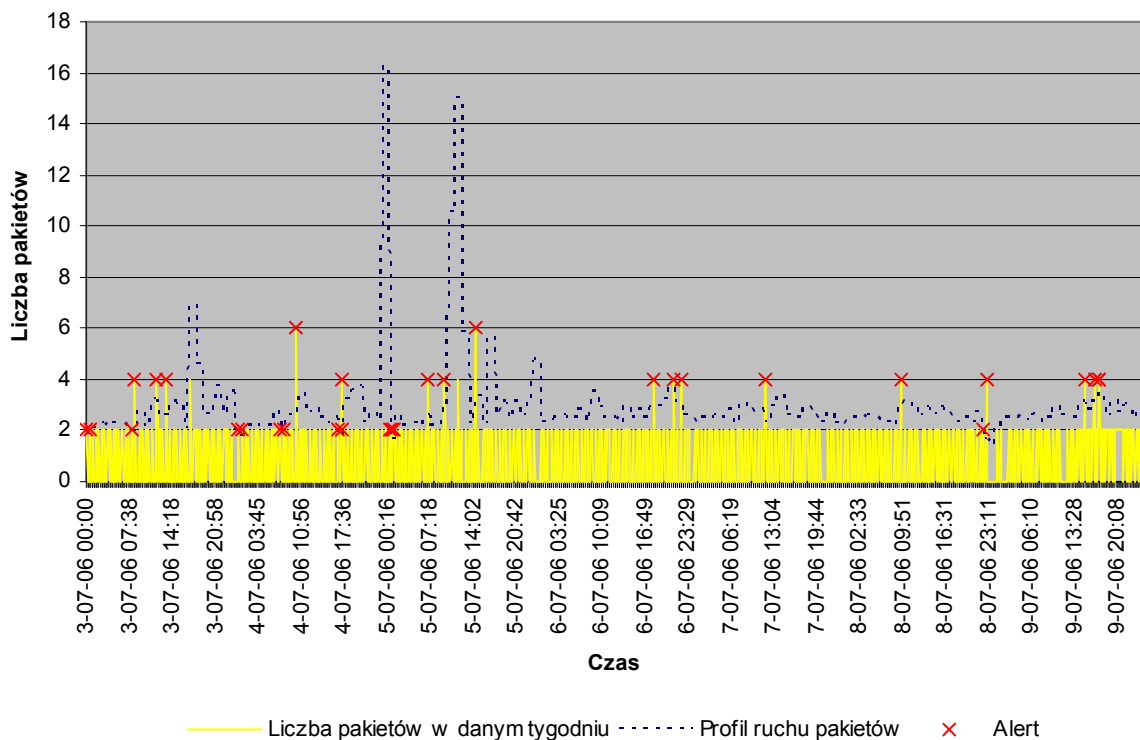
Rysunek 133: Statystyka alertów – pakiety ICMP. Źródło: opracowanie własne.



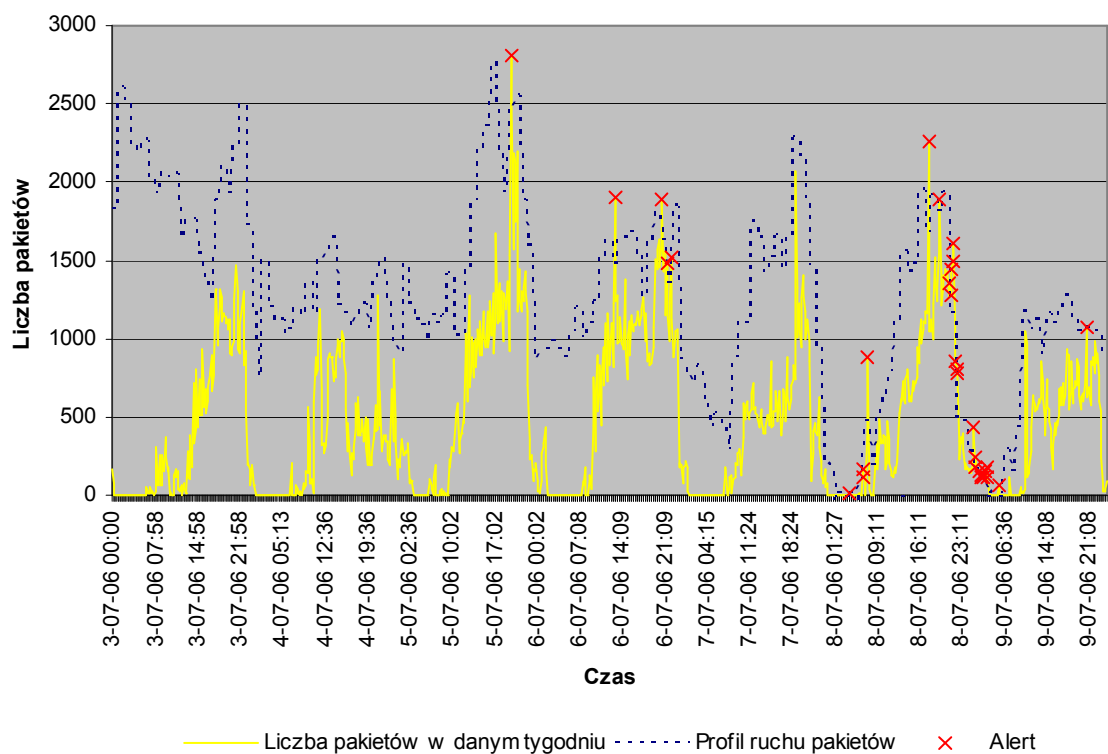
**Rysunek 134: Statystyka alertów – wysłane pakiety ICMP. Źródło: opracowanie własne.**



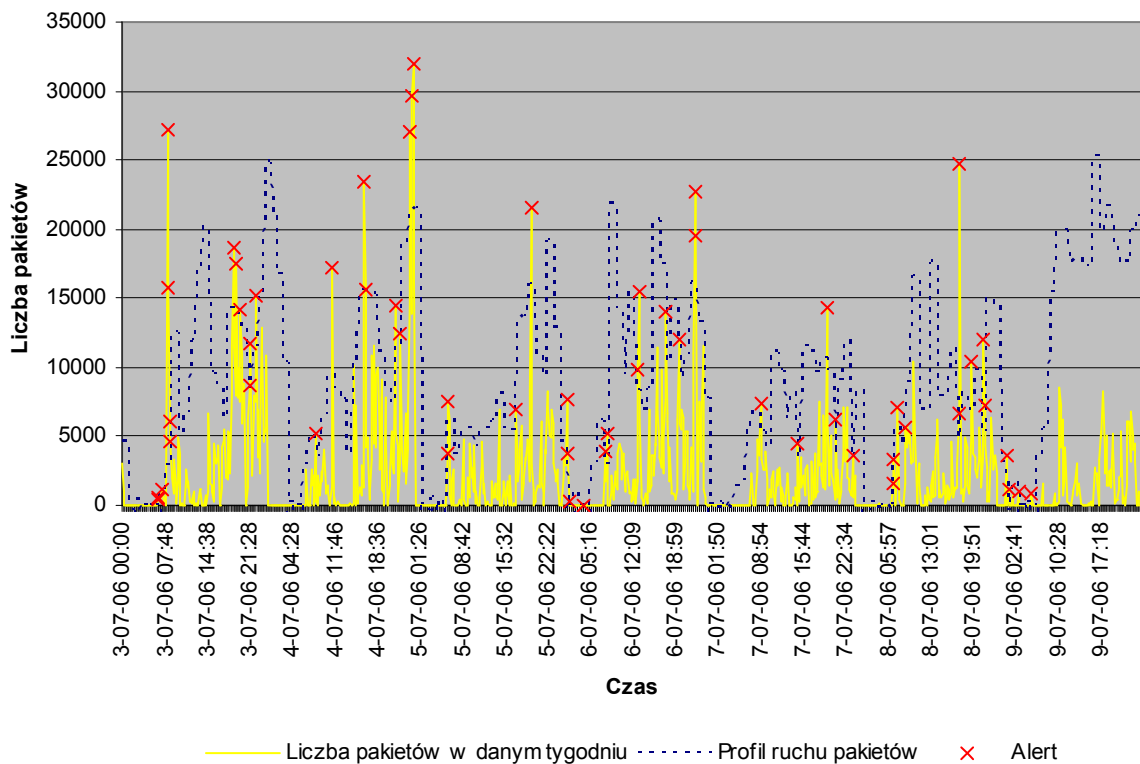
**Rysunek 135: Statystyka alertów – odebrane pakiety ICMP. Źródło: opracowanie własne.**



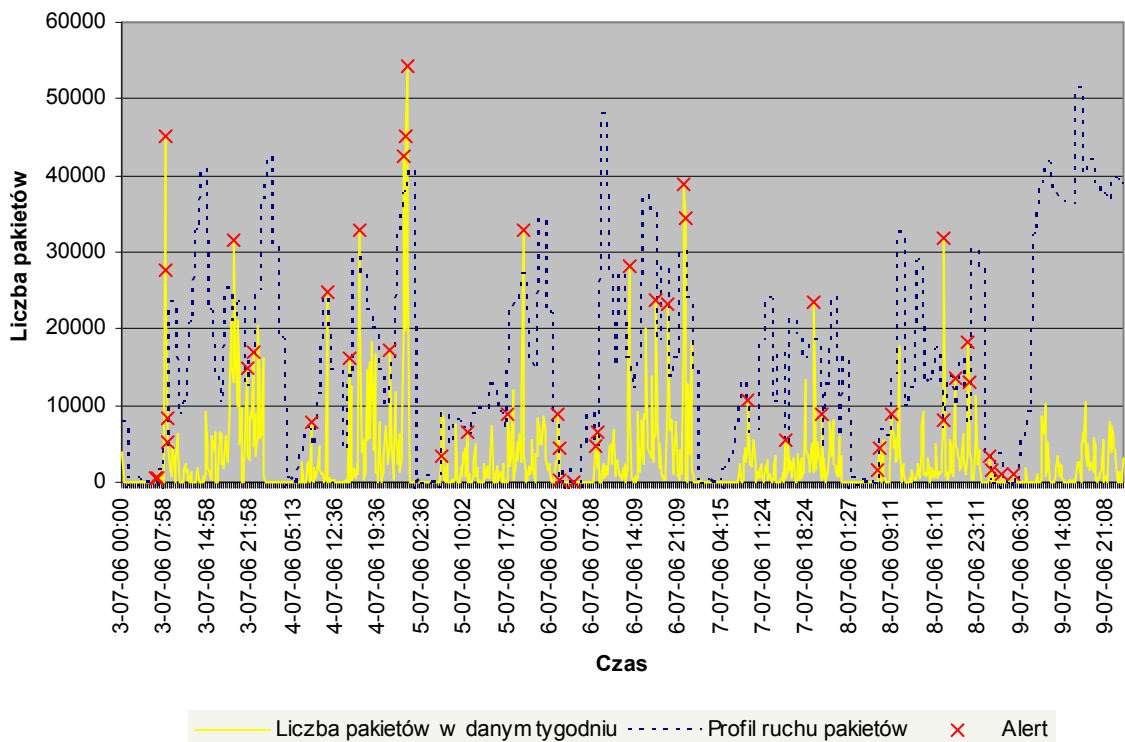
Rysunek 136: Statystyka alertów – pakiety ICMP wewnątrz sieci LAN. Źródło: opracowanie własne.



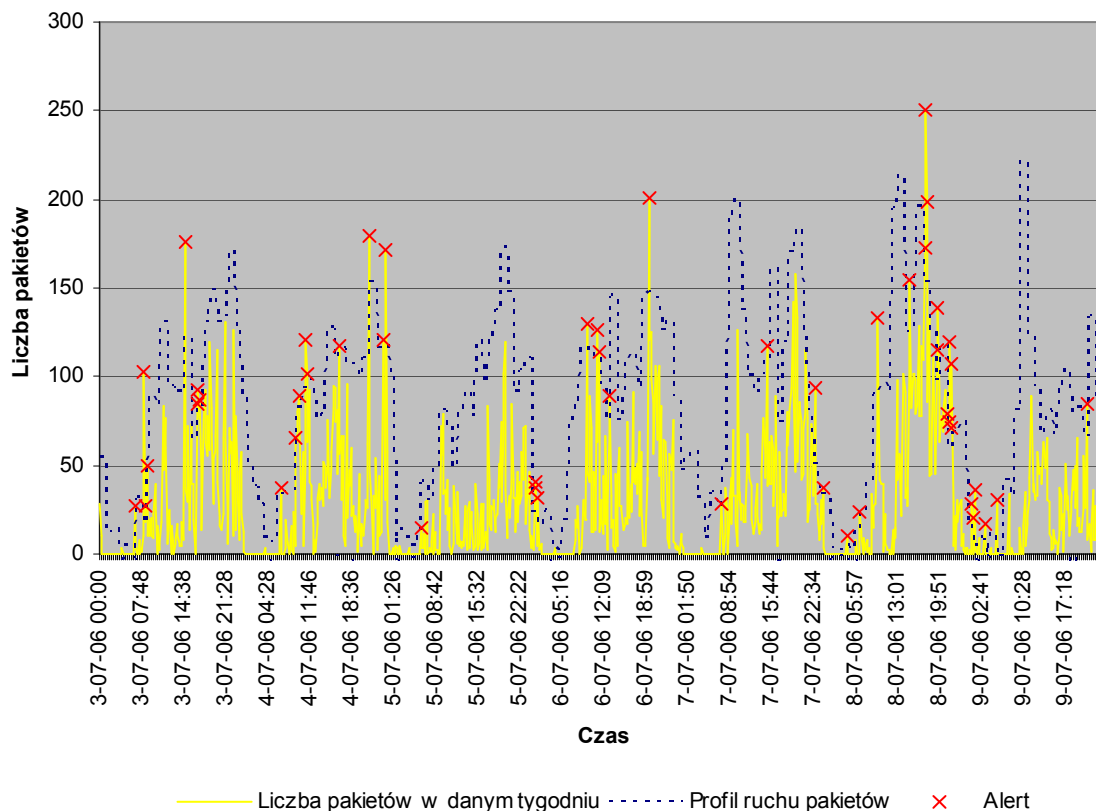
Rysunek 137: Statystyka alertów – nowe połączenia (TCP z flagami SYN i ACK). Źródło: opracowanie własne.



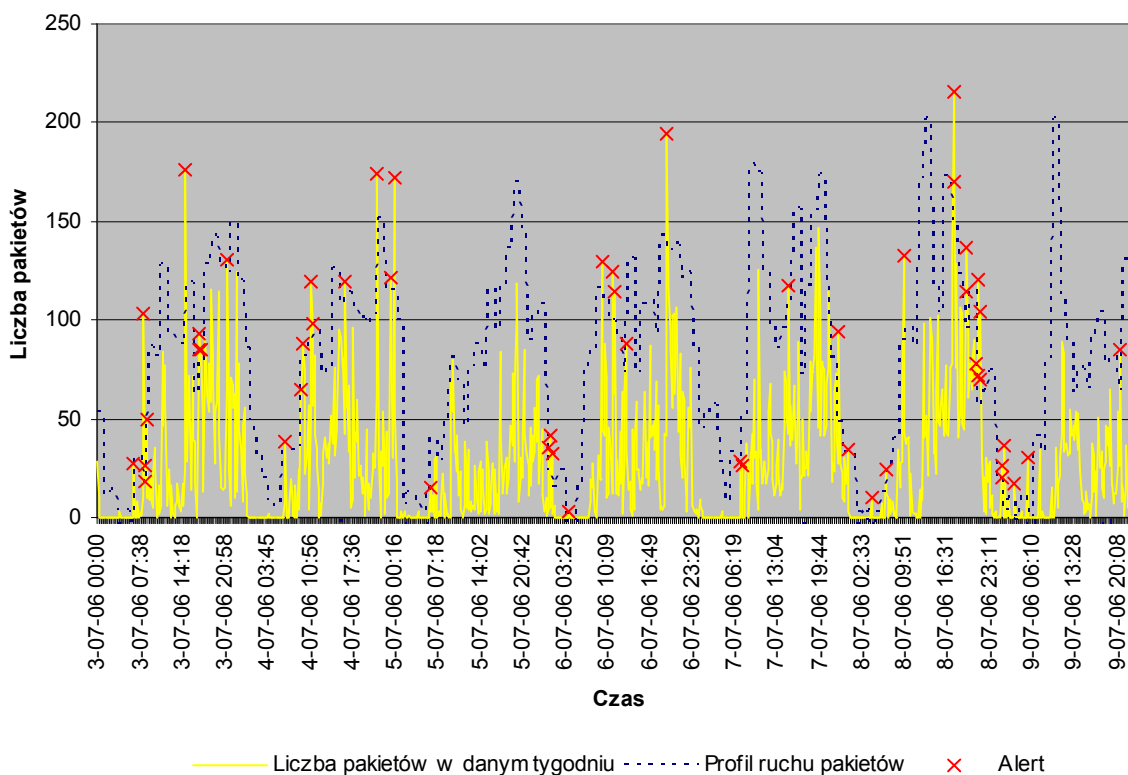
Rysunek 138: Statystyka alertów – wysłane pakiety TCP (port 80). Źródło: opracowanie własne.



Rysunek 139: Statystyka alertów – odebrane pakiety TCP (port 80). Źródło: opracowanie własne.

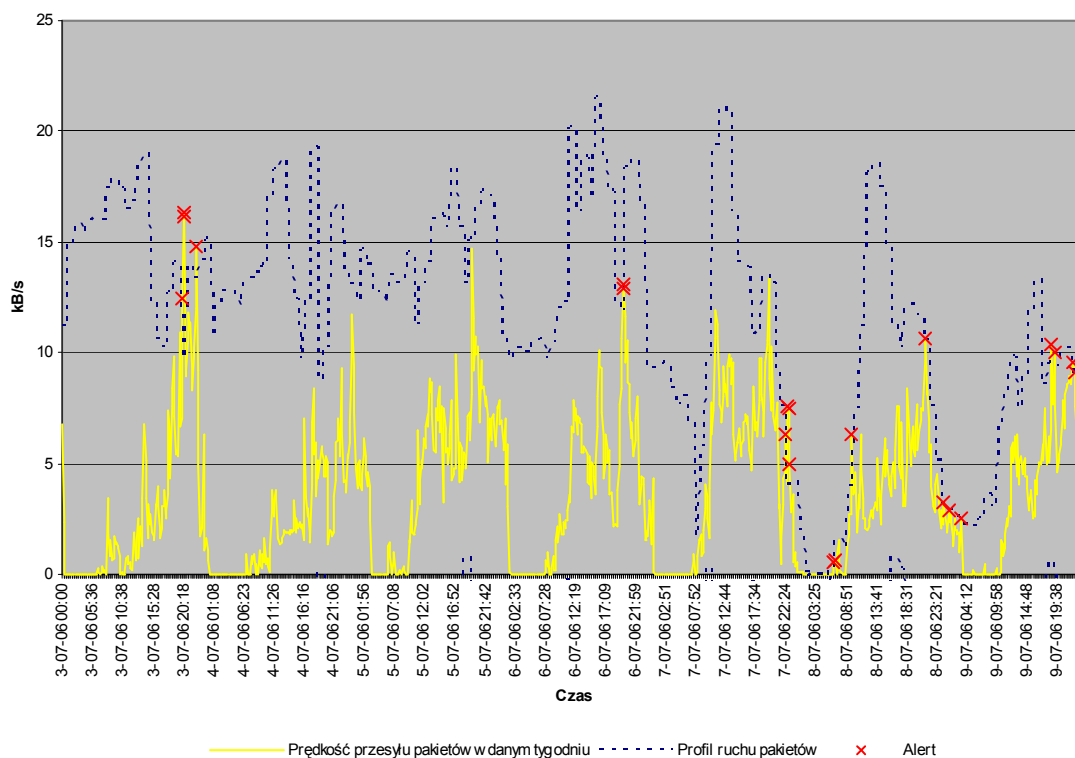


**Rysunek 140: Statystyka alertów – wysłane pakiety UDP (port 53). Źródło: opracowanie własne.**

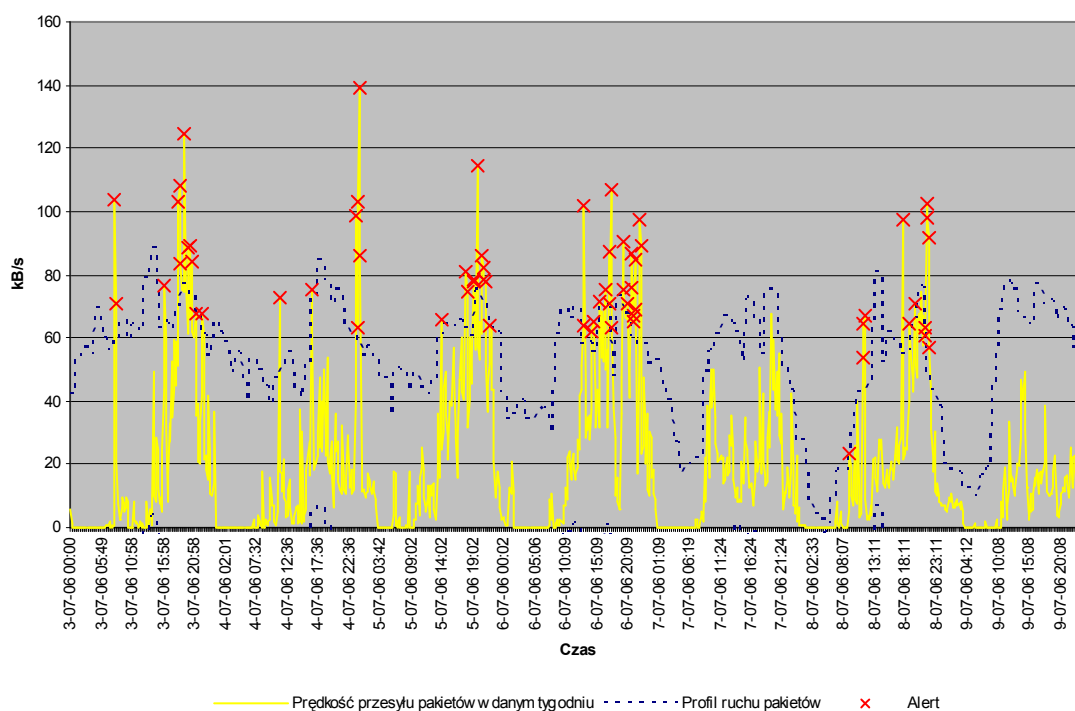


**Rysunek 141: Statystyka alertów – odebrane pakiety UDP (port 53). Źródło: opracowanie własne.**

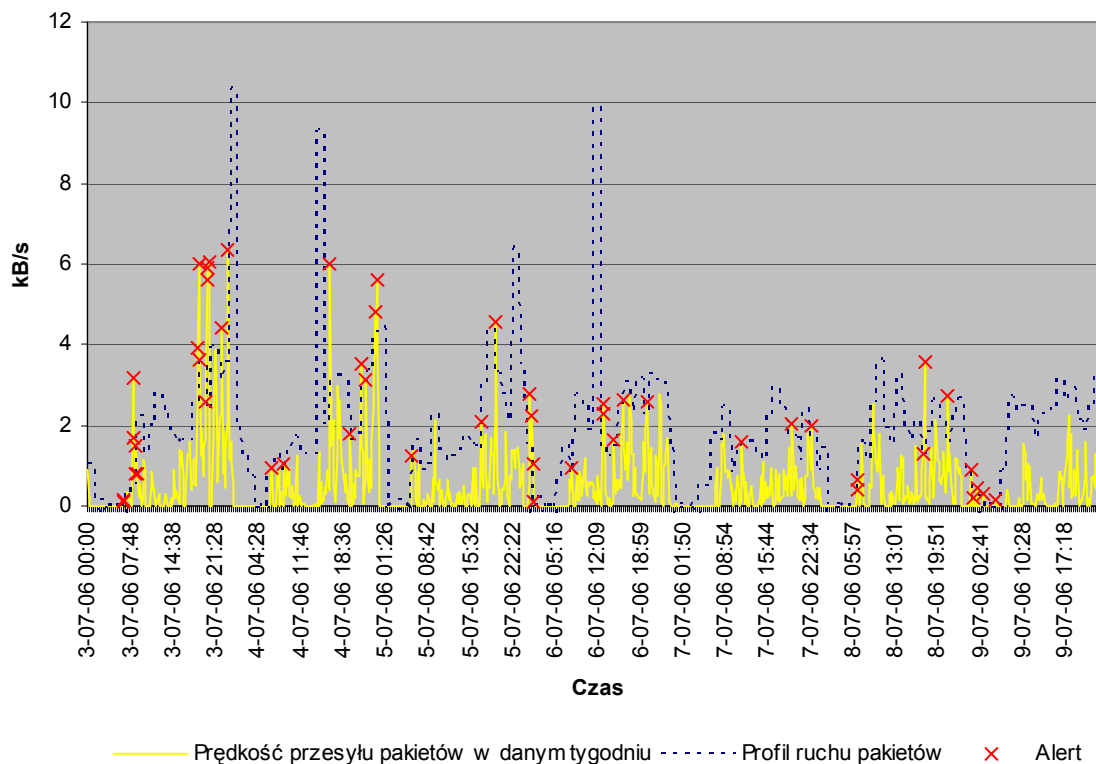




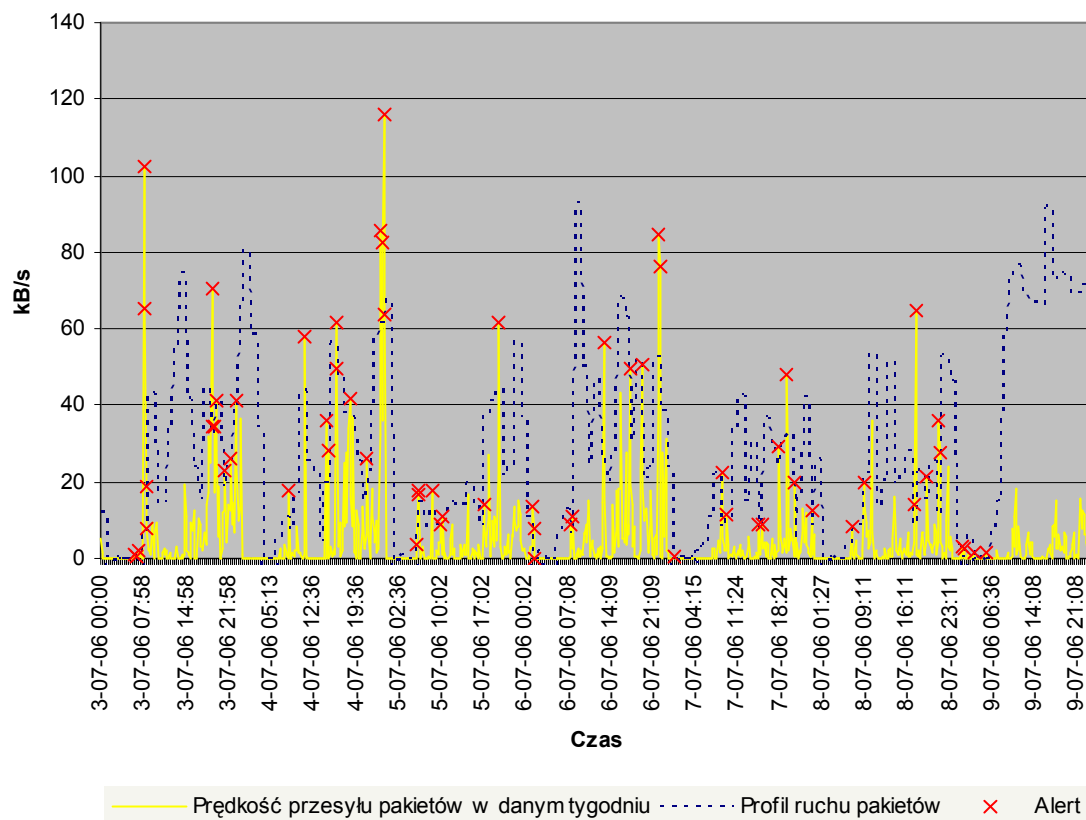
**Rysunek 142: Statystyka alertów – ruch TCP (dane wysłane). Źródło: opracowanie własne.**



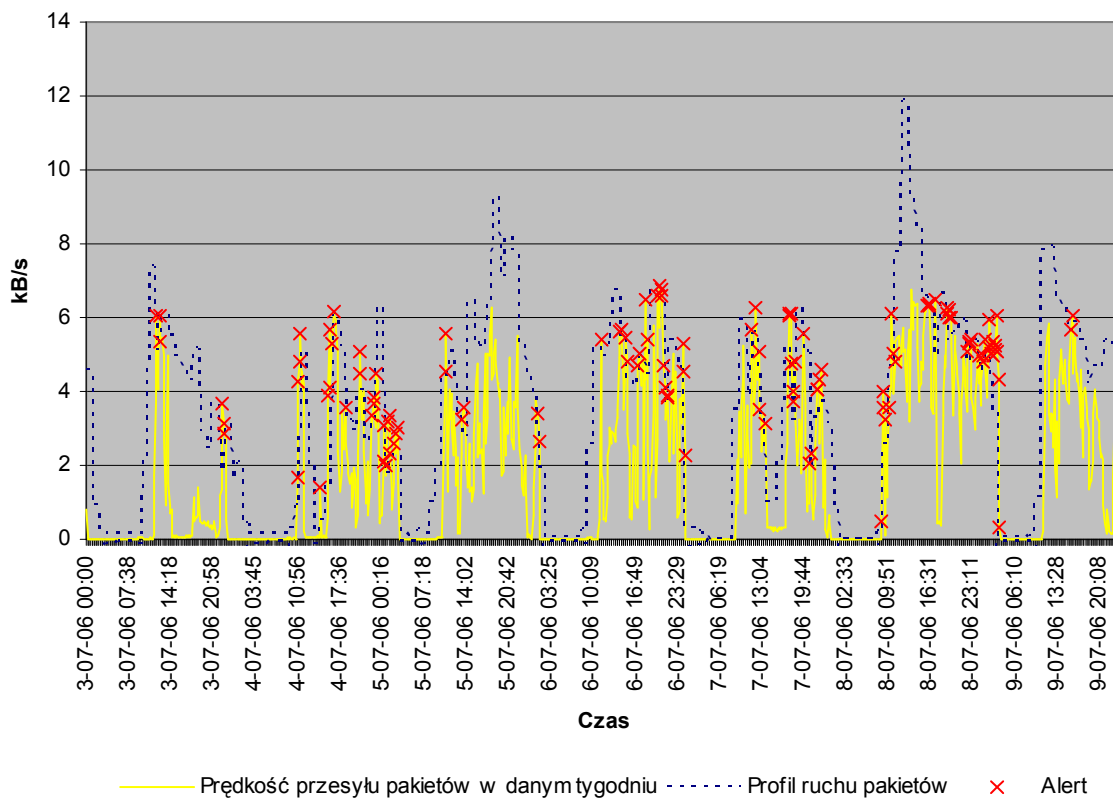
**Rysunek 143: Statystyka alertów – ruch TCP (dane odebrane). Źródło: opracowanie własne.**



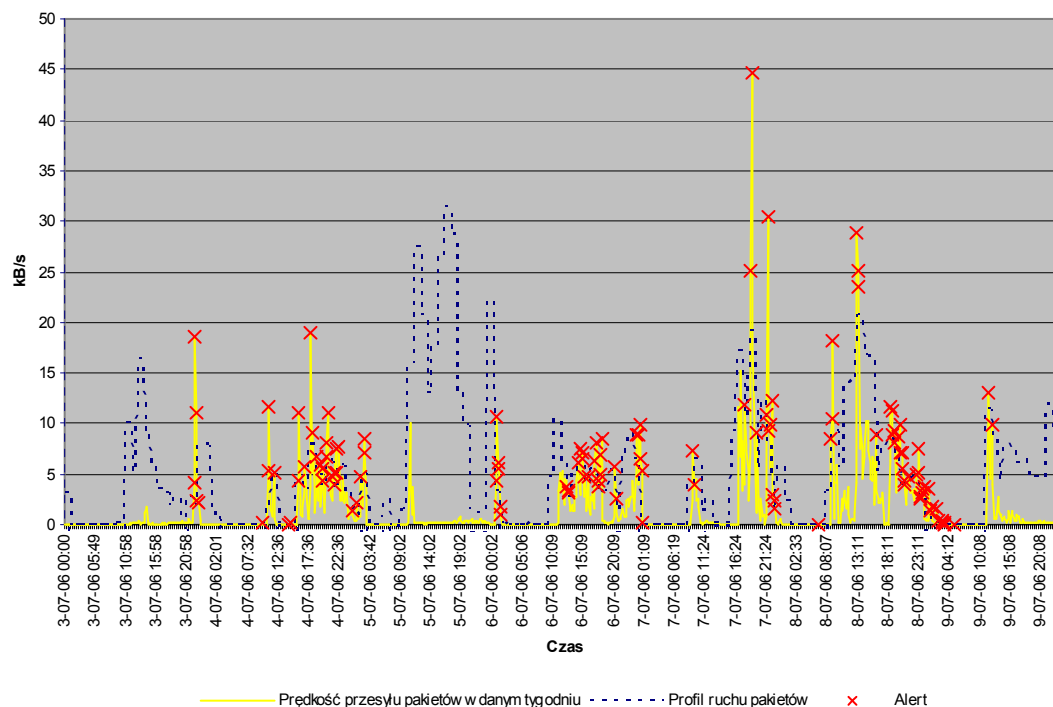
Rysunek 144: Statystyka alertów – ruch WWW (dane wysłane). Źródło: opracowanie własne.



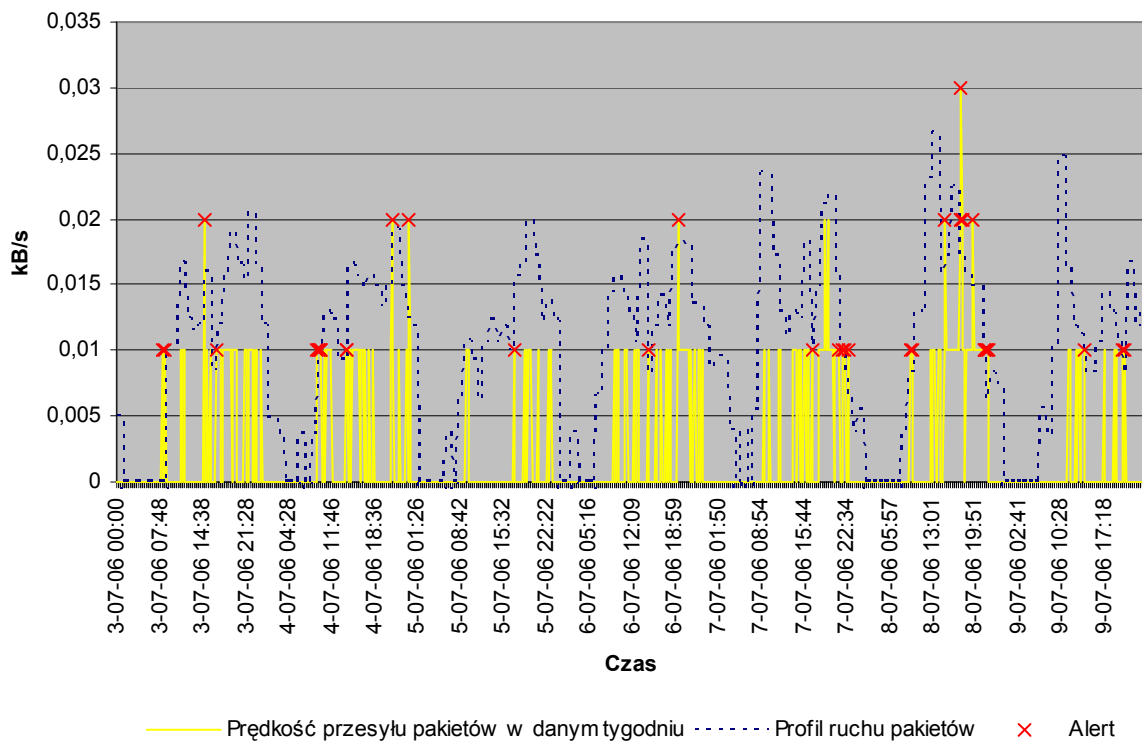
Rysunek 145: Statystyka alertów – ruch WWW (dane odebrane). Źródło: opracowanie własne.



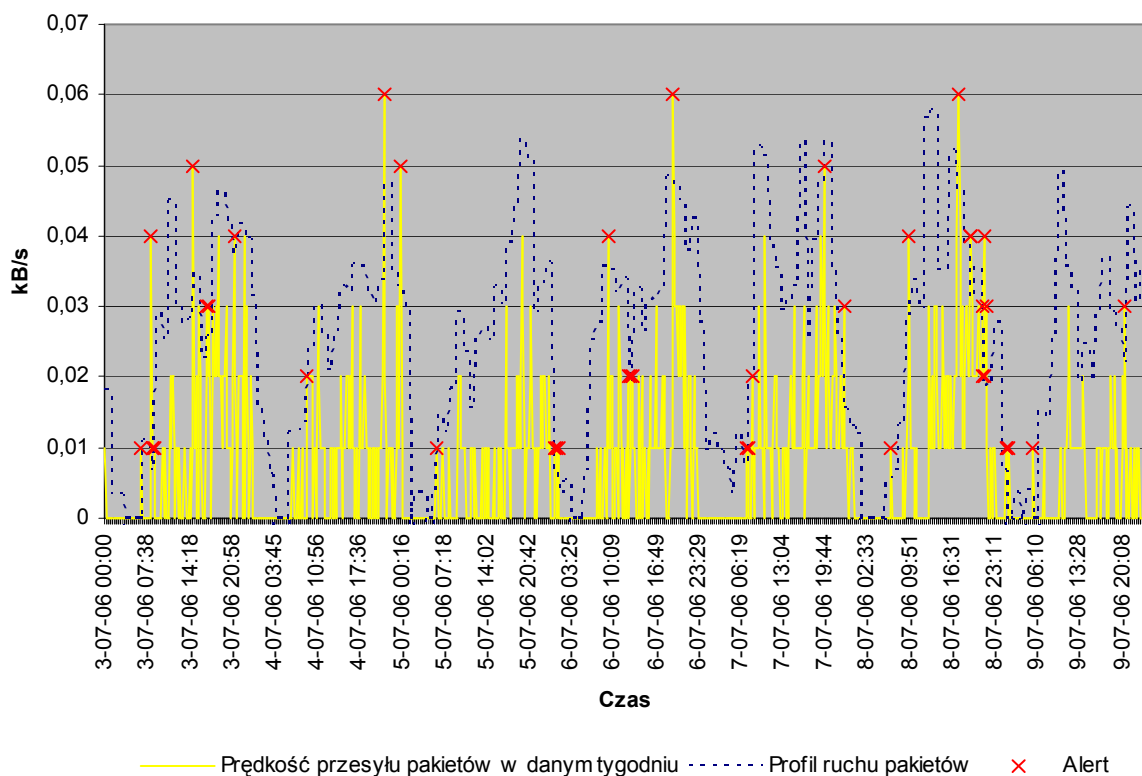
Rysunek 146: Statystyka alertów – ruch UDP (dane wysłane). Źródło: opracowanie własne.



Rysunek 147: Statystyka alertów – ruch UDP (dane odebrane). Źródło: opracowanie własne.

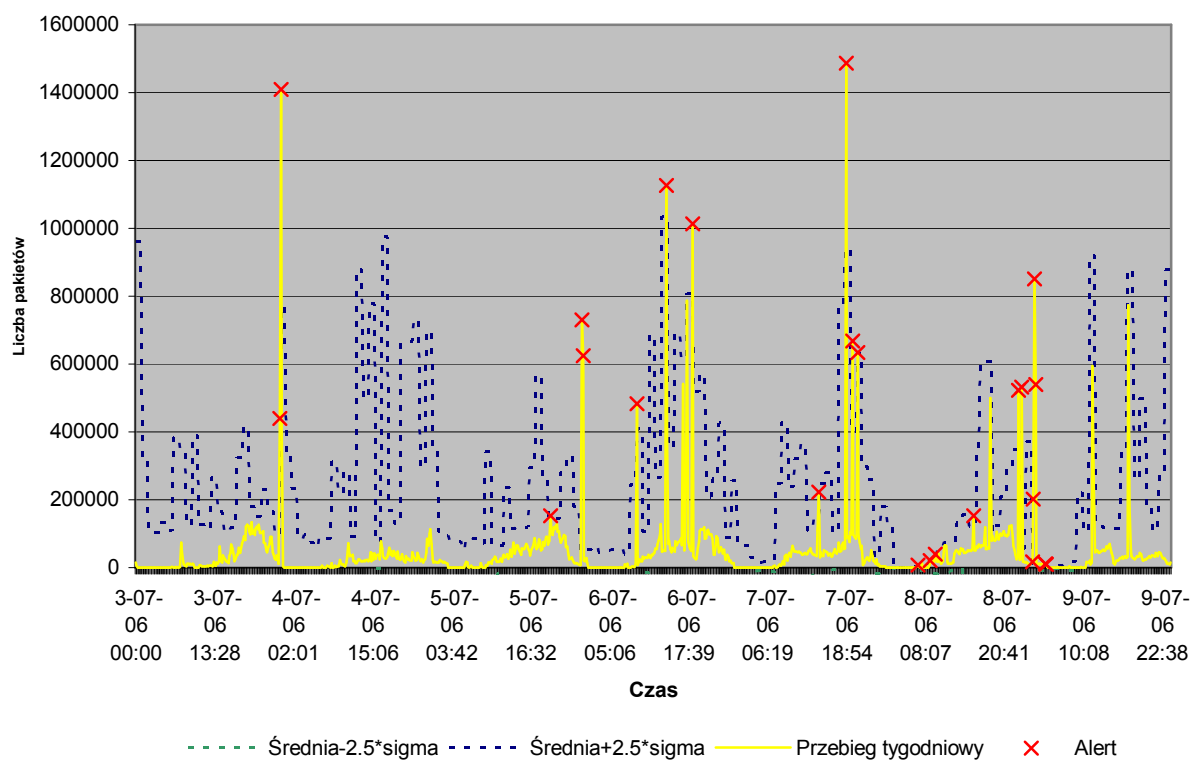


Rysunek 148: Statystyka alertów – ruch UDP port 53 (dane wysłane). Źródło: opracowanie własne.

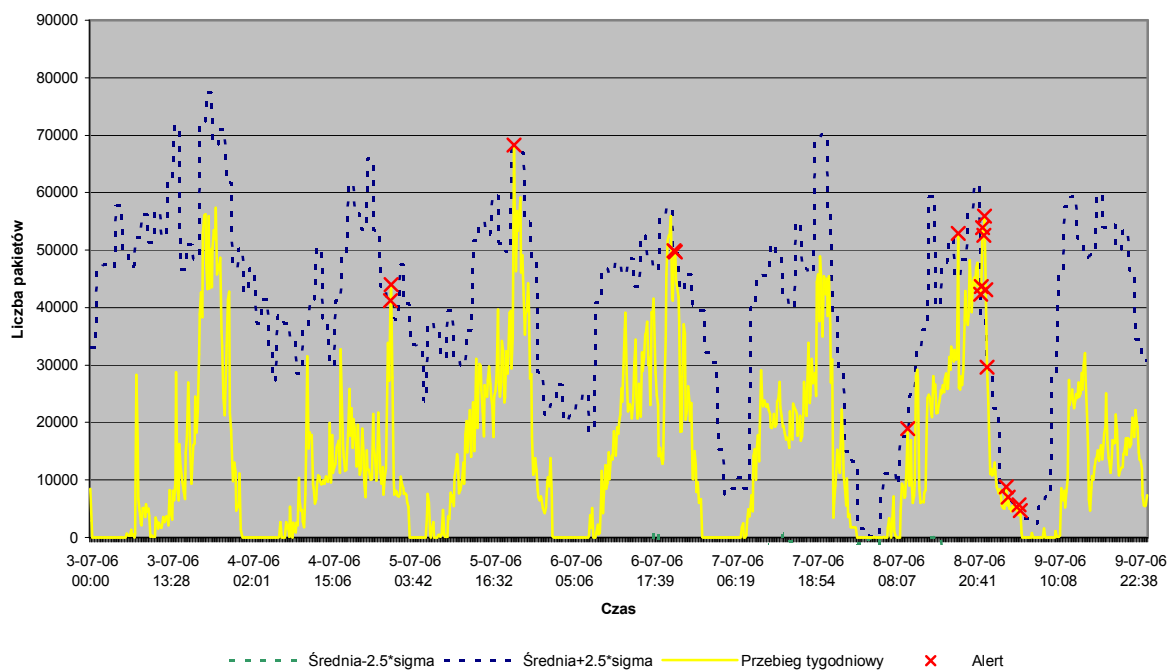


Rysunek 149: Statystyka alertów – ruch UDP port 53 (dane odebrane). Źródło: opracowanie własne.

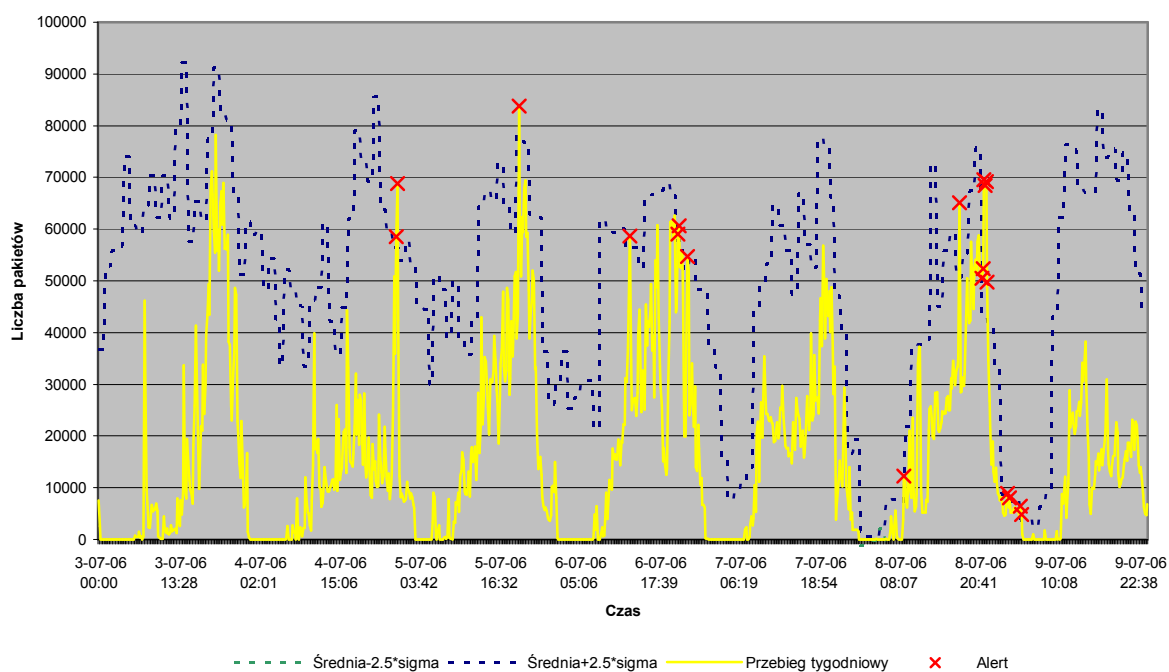
#### Załącznik 4: Wykresy dla mnożnika sigmy równego 2,5.



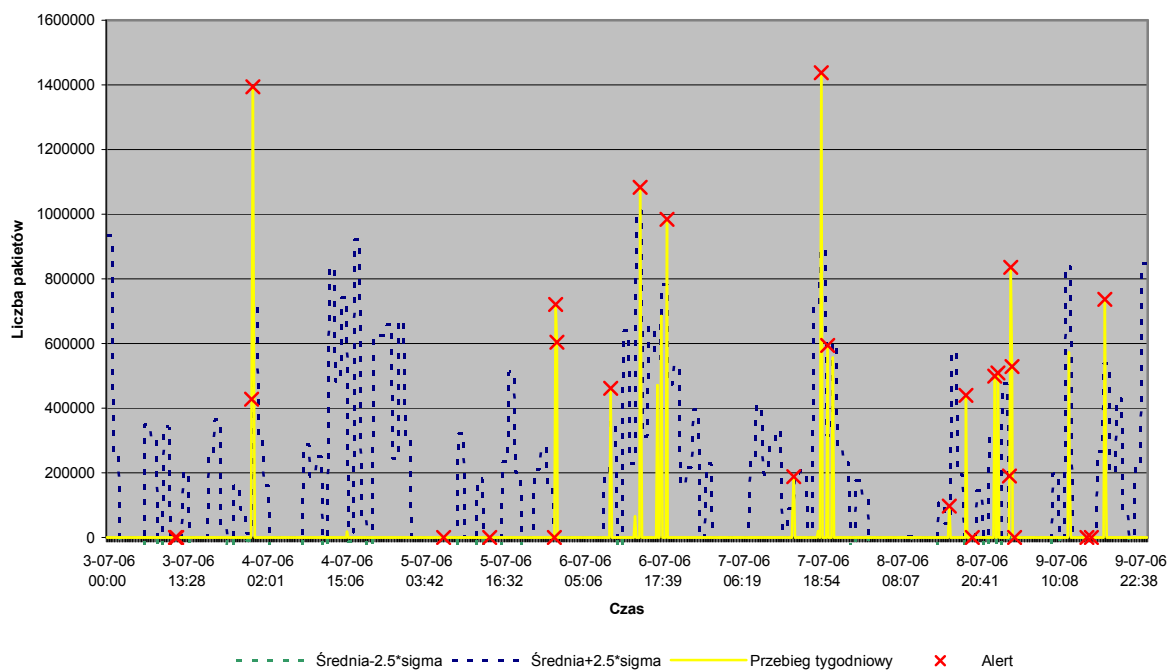
Rysunek 150: Statystyka alertów - ruch TCP. Źródło: opracowanie własne.



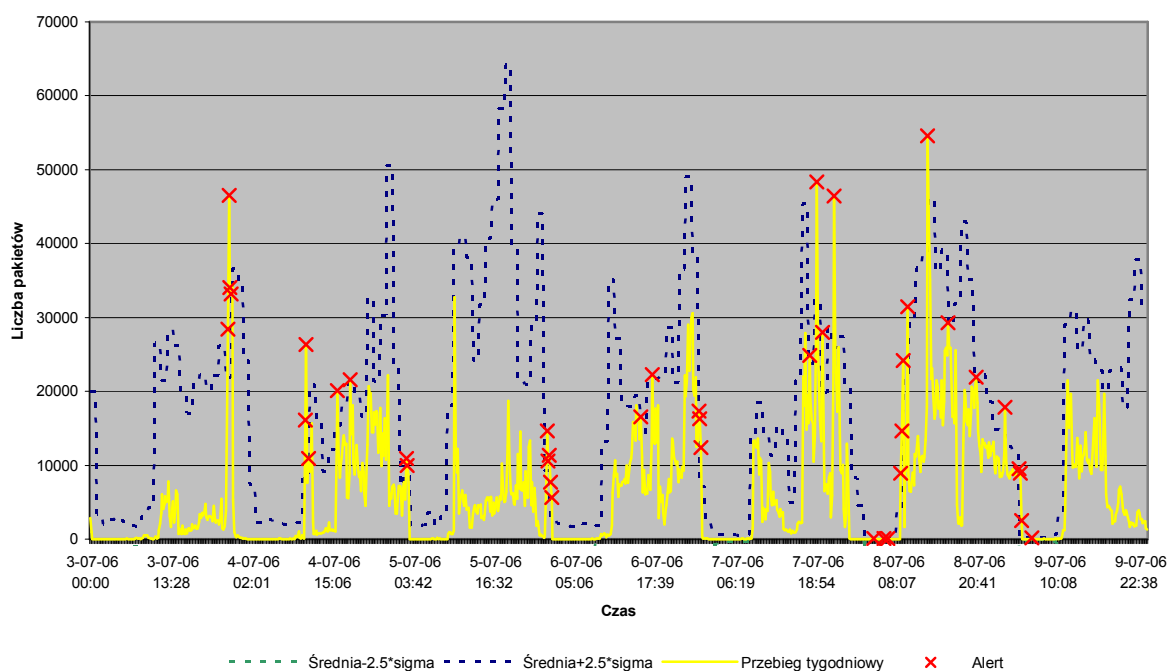
**Rysunek 151: Statystyka alertów – wysłane pakiety TCP. Źródło: opracowanie własne.**



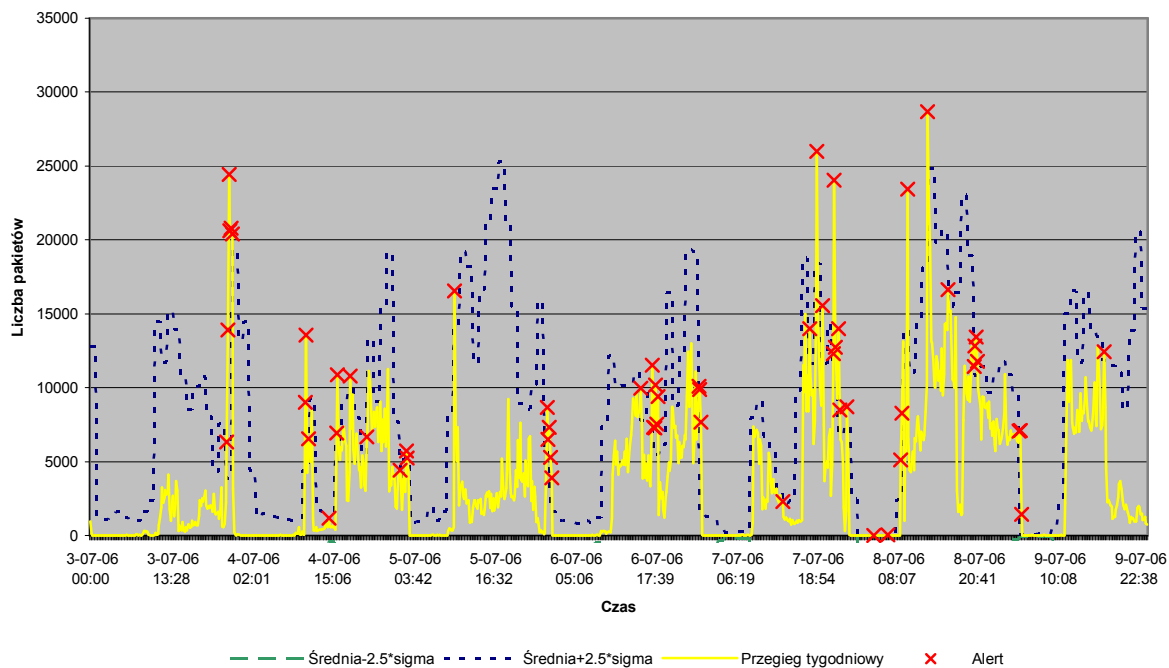
**Rysunek 152: Statystyka alertów – odebrane pakiety TCP. Źródło: opracowanie własne.**



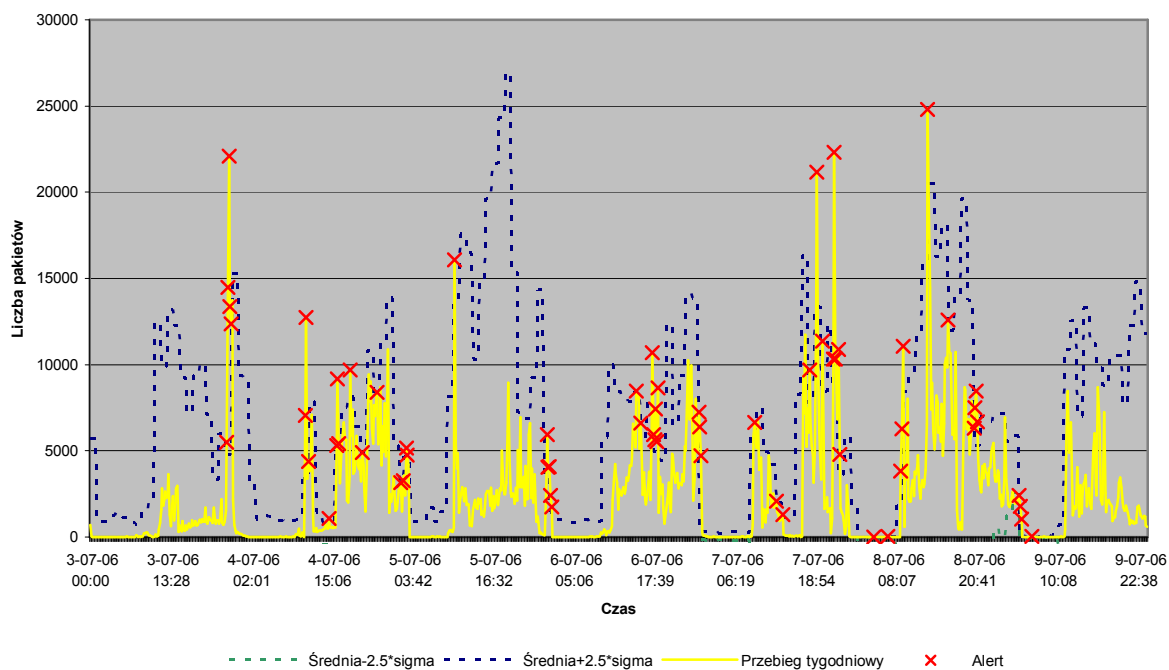
**Rysunek 153: Statystyka alertów – pakiety TCP wewnątrz sieci LAN. Źródło: opracowanie własne.**



**Rysunek 154: Statystyka alertów – pakiety UDP. Źródło: opracowanie własne.**

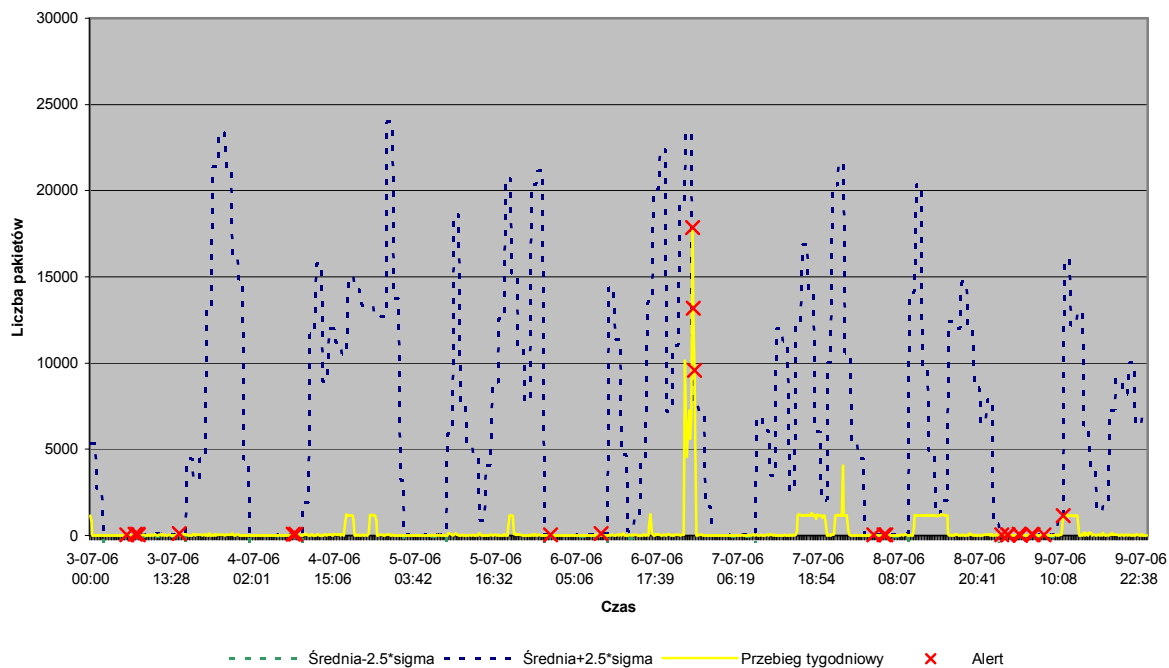


**Rysunek 155: Statystyka alertów – wysłane pakiety UDP. Źródło: opracowanie własne.**

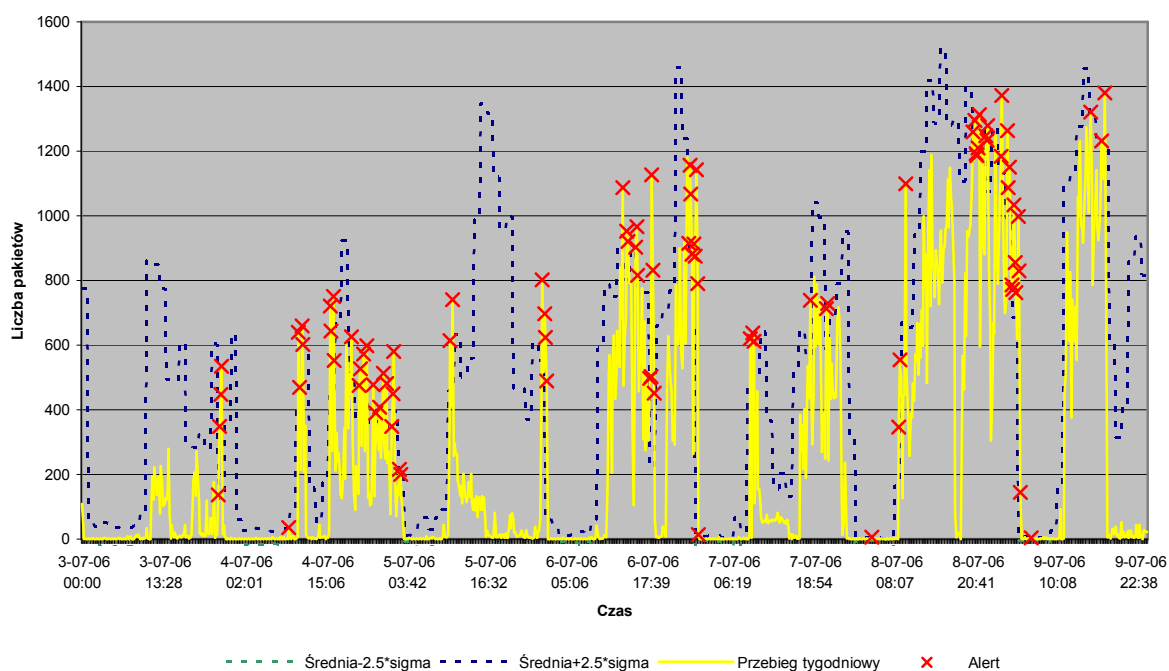


**Rysunek 156: Statystyka alertów – odebrane pakiety UDP. Źródło: opracowanie własne.**

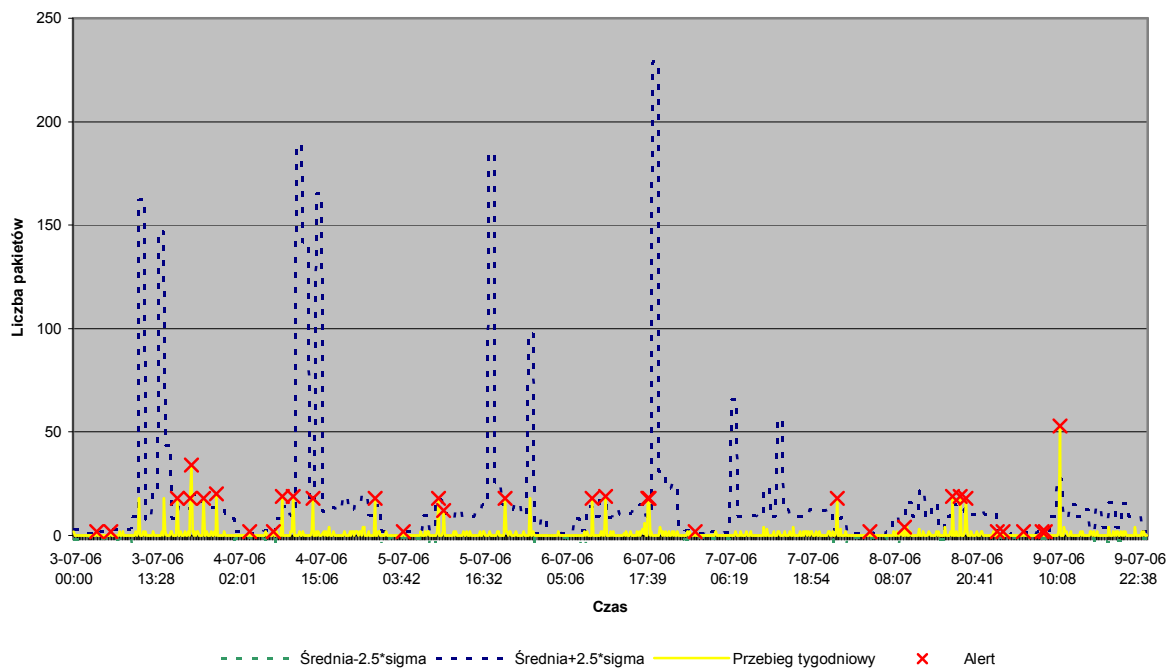




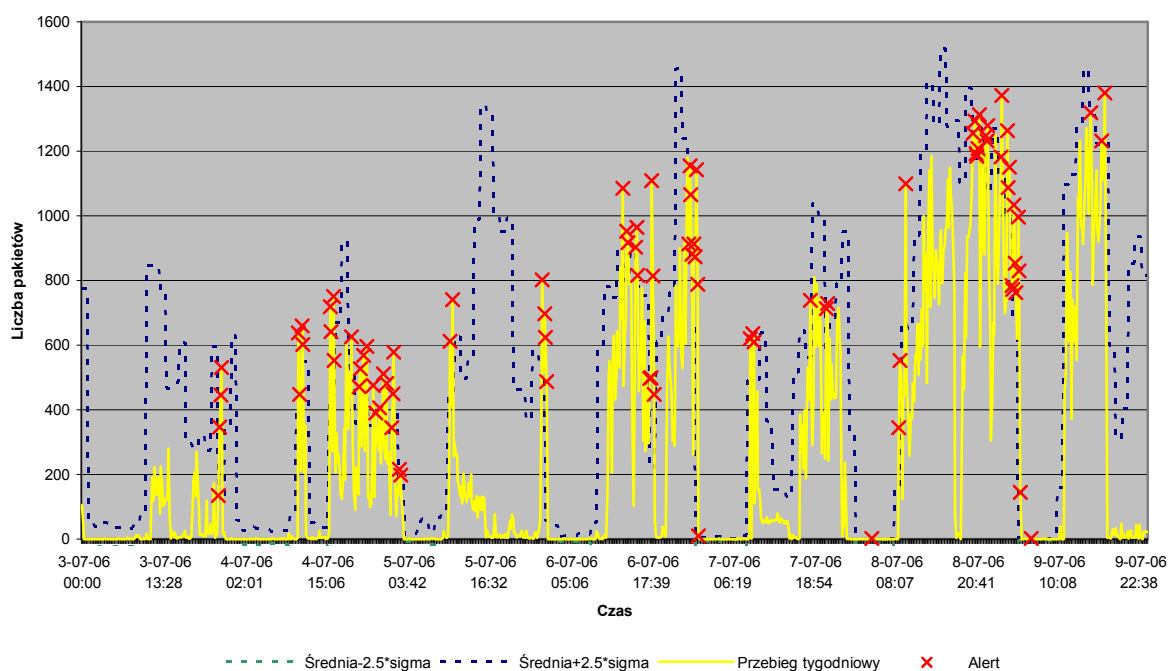
**Rysunek 157: Statystyka alertów – pakiety UDP wewnątrz sieci LAN. Źródło: opracowanie własne.**



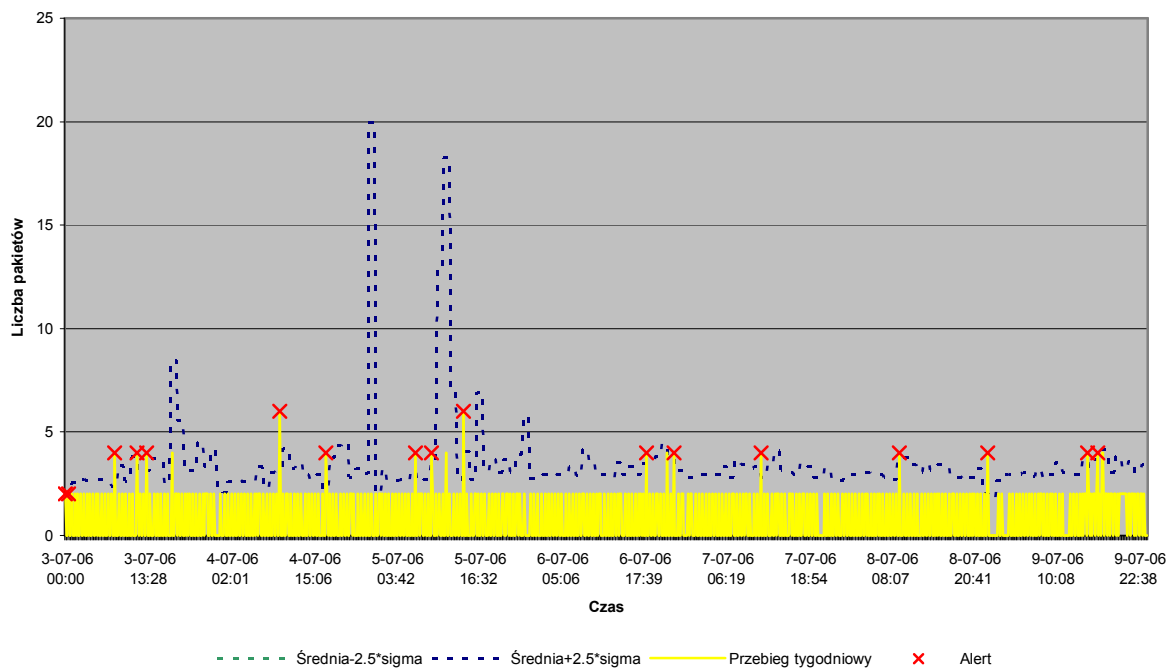
**Rysunek 158: Statystyka alertów – pakiety ICMP. Źródło: opracowanie własne.**



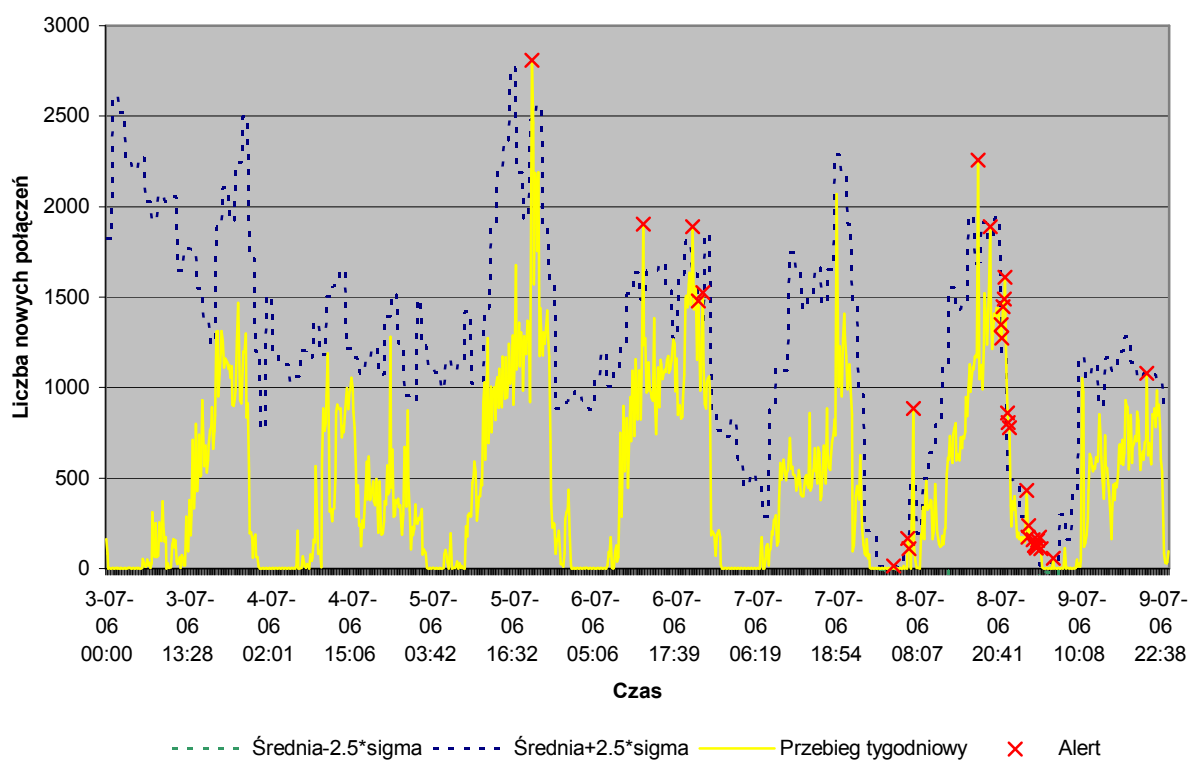
**Rysunek 159: Statystyka alertów – wysłane pakiety ICMP. Źródło: opracowanie własne.**



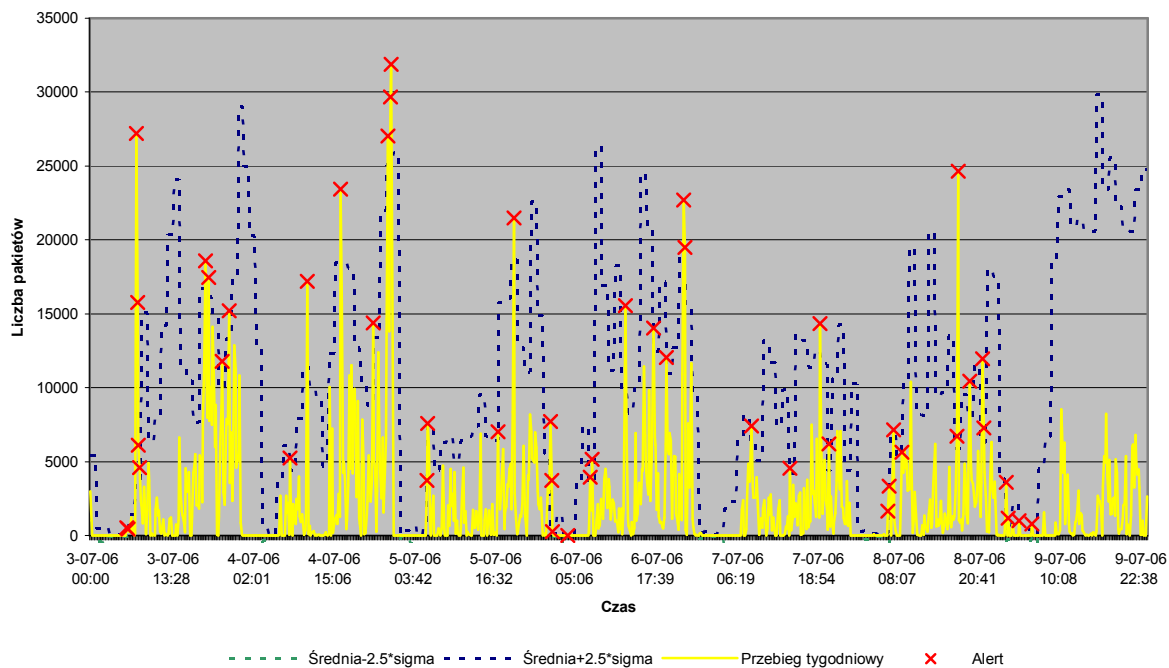
**Rysunek 160: Statystyka alertów – odebrane pakiety ICMP. Źródło: opracowanie własne.**



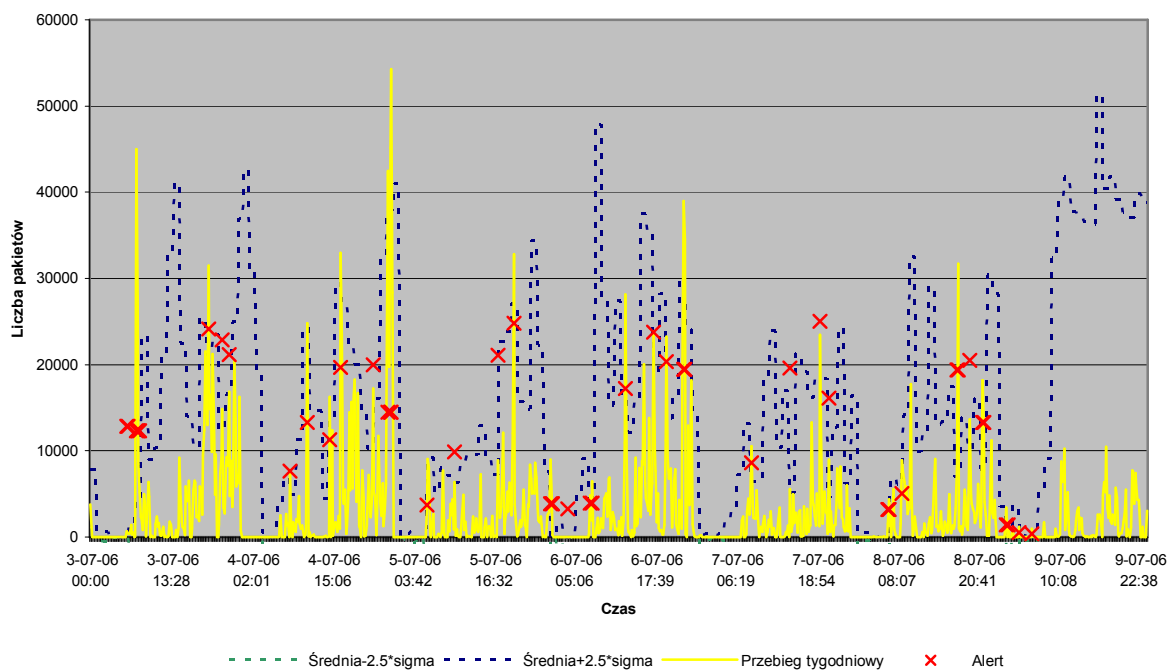
Rysunek 161: Statystyka alertów – pakiety ICMP wewnątrz sieci LAN. Źródło: opracowanie własne.



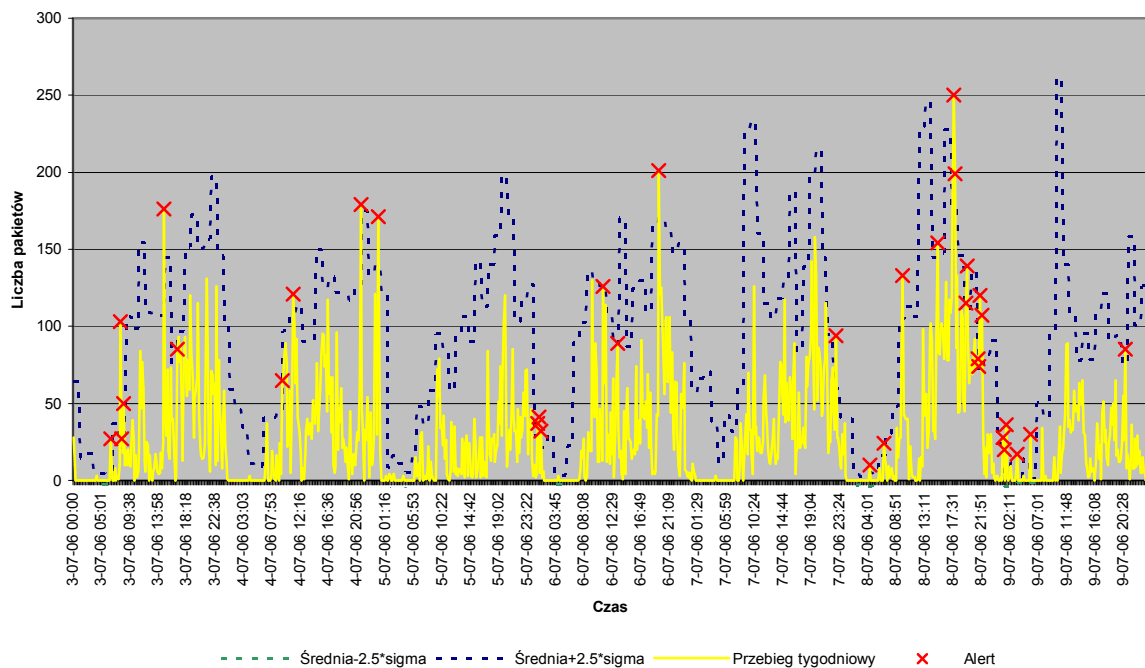
Rysunek 162: Statystyka alertów – nowe połączenia (TCP z flagami SYN i ACK). Źródło: opracowanie własne.



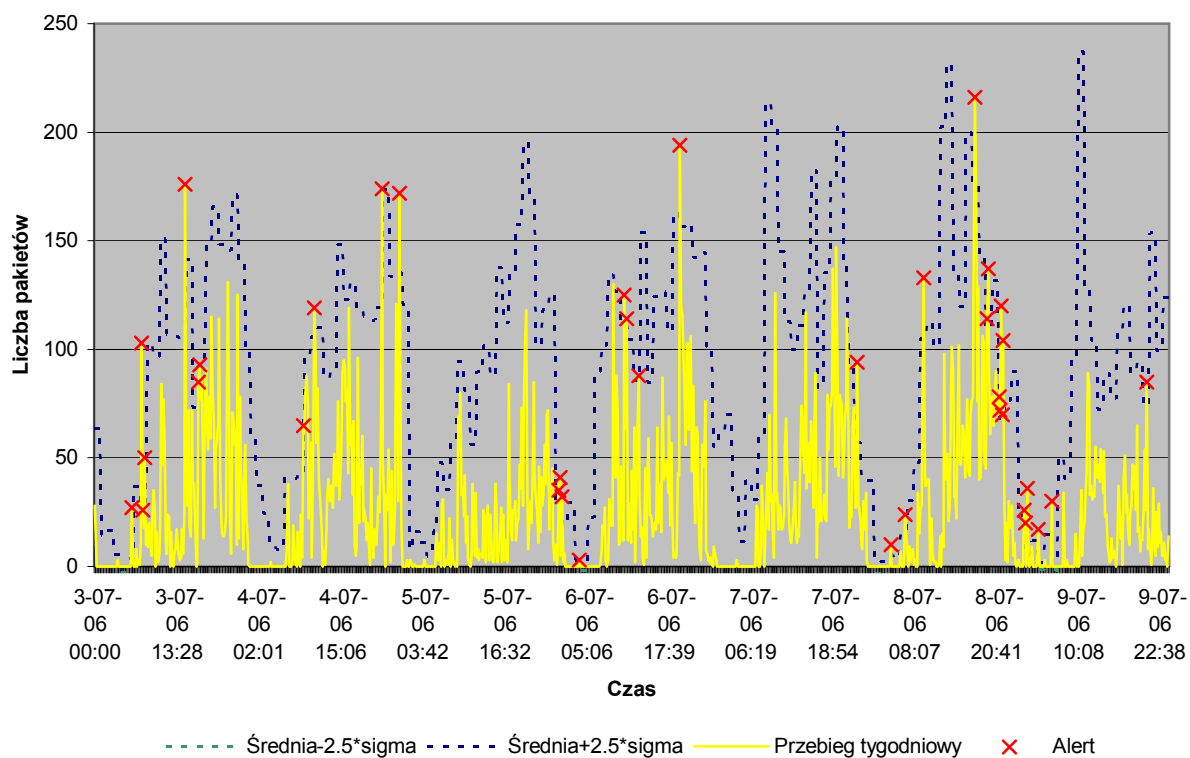
**Rysunek 163: Statystyka alertów – wysłane pakiety TCP (port 80). Źródło: opracowanie własne.**



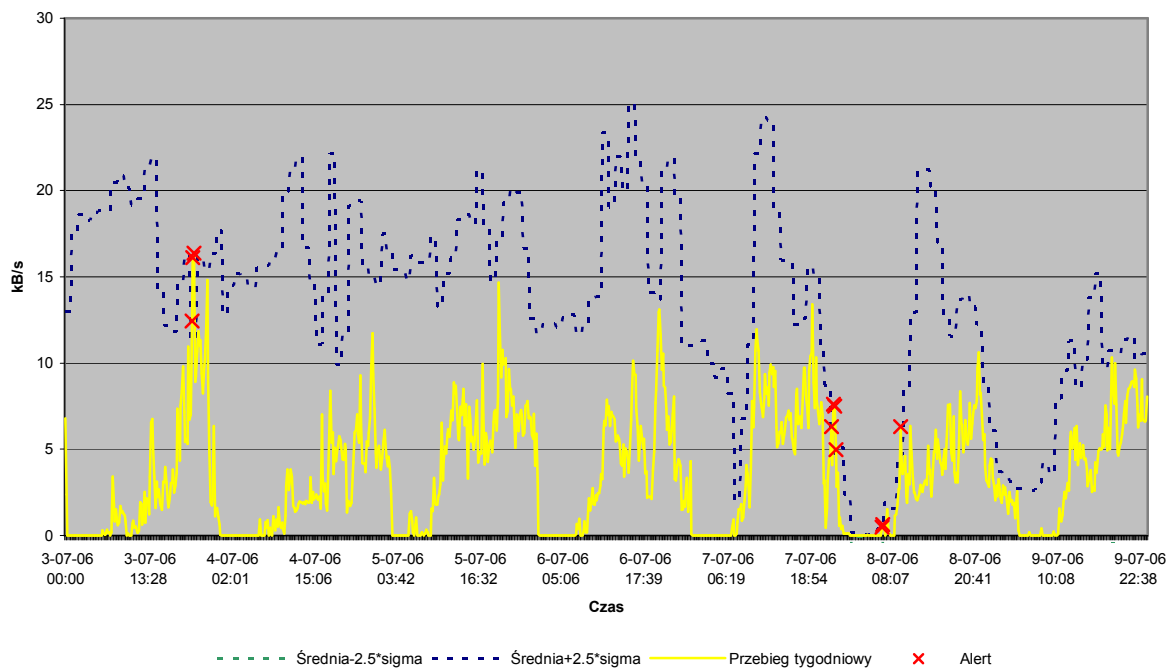
**Rysunek 164: Statystyka alertów – odebrane pakiety TCP (port 80). Źródło: opracowanie własne.**



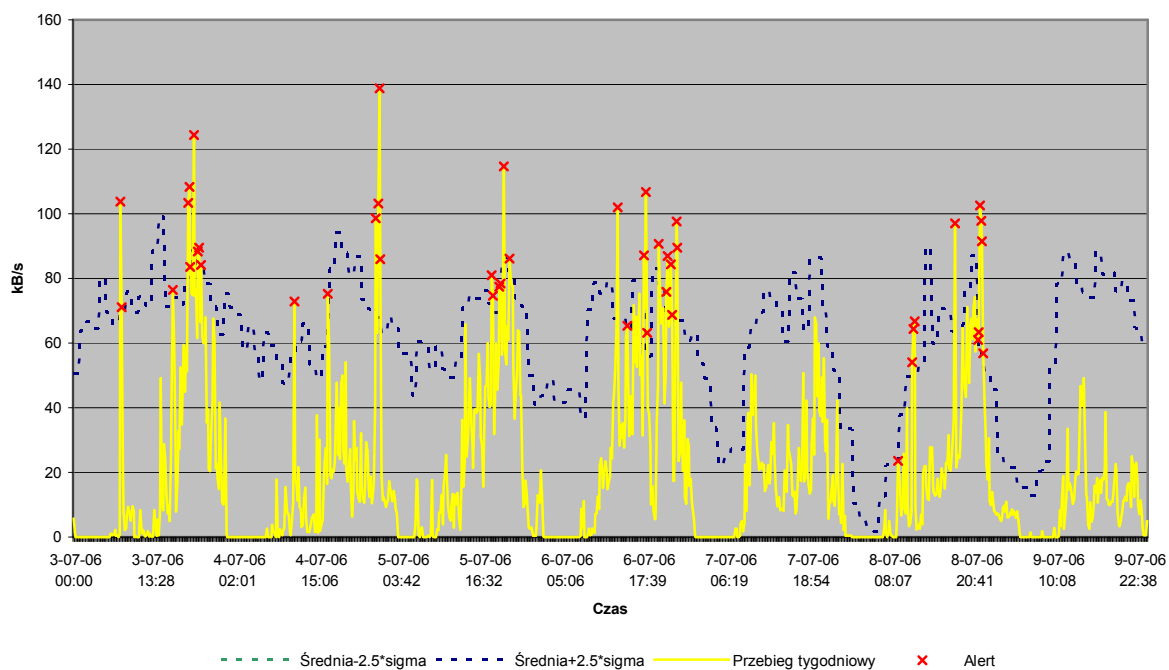
Rysunek 165: Statystyka alertów – wysłane pakiety UDP (port 53). Źródło: opracowanie własne.



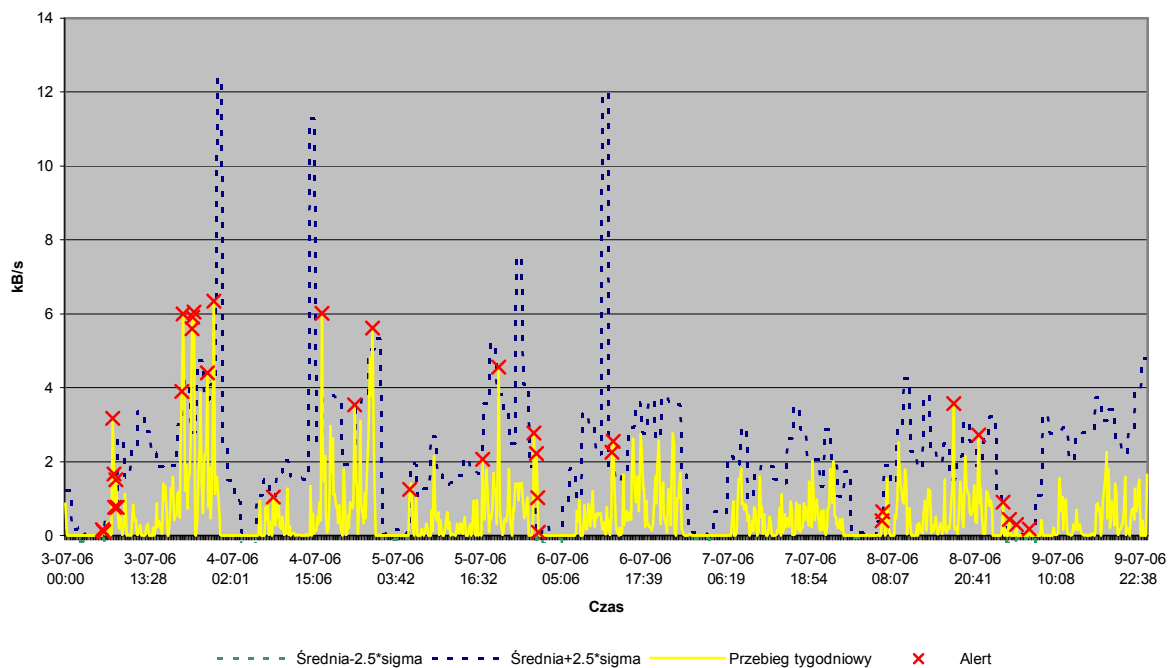
Rysunek 166: Rysunek 167: Statystyka alertów – odebrane pakiety UDP (port 53). Źródło: opracowanie własne.



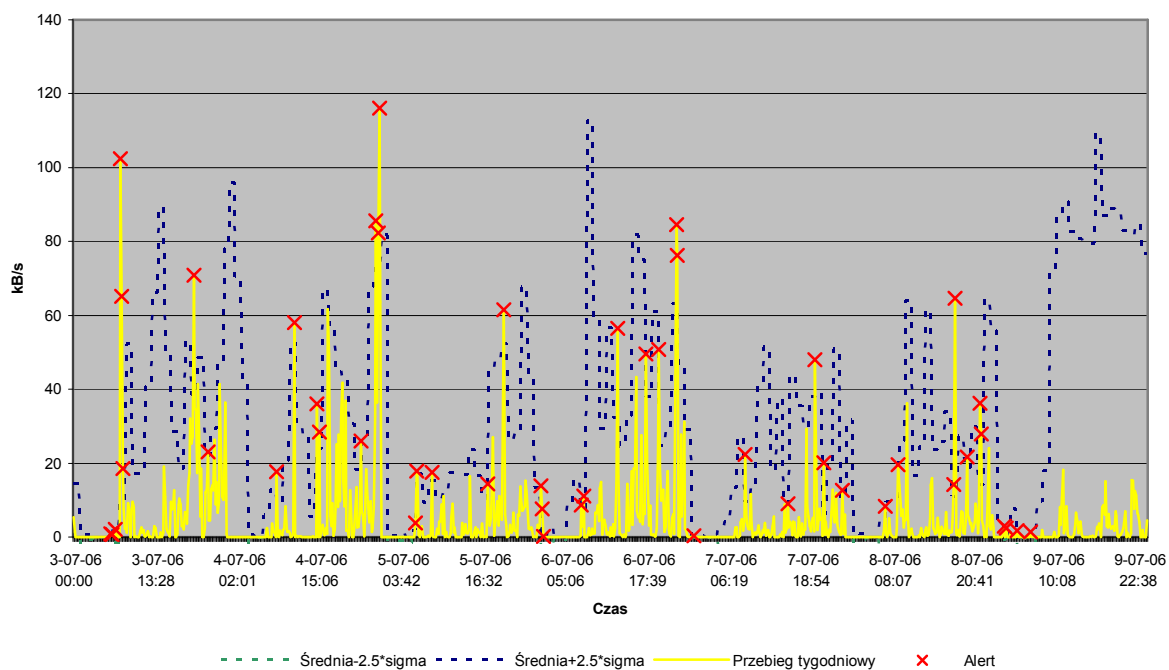
**Rysunek 168: Statystyka alertów – ruch TCP (dane wysłane). Źródło: opracowanie własne.**



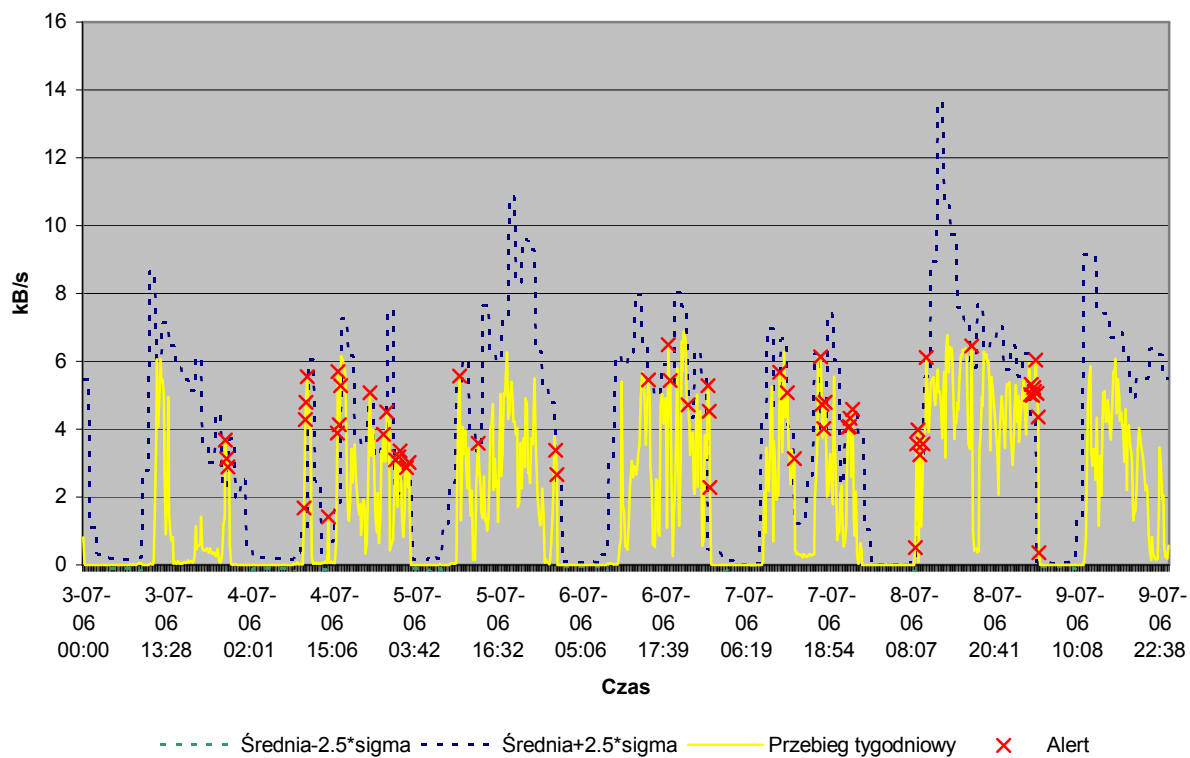
**Rysunek 169: Statystyka alertów – ruch TCP (dane odebrane). Źródło: opracowanie własne.**



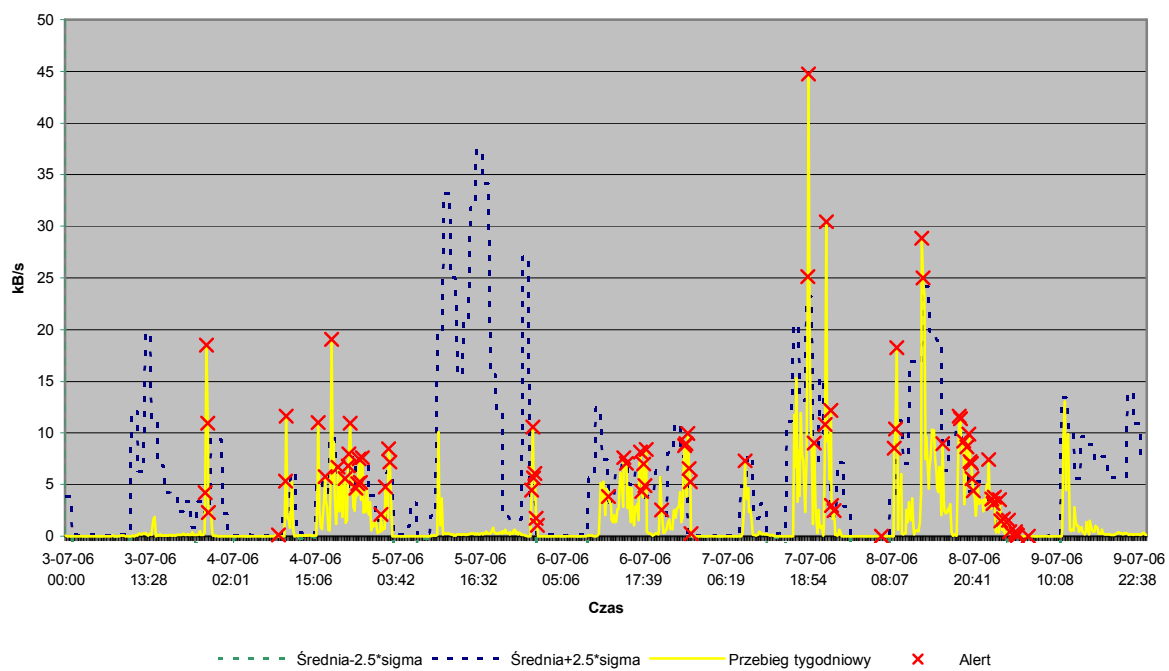
**Rysunek 170: Statystyka alertów – ruch WWW (dane wysłane). Źródło: opracowanie własne.**



**Rysunek 171: Statystyka alertów – ruch WWW (dane odebrane). Źródło: opracowanie własne.**

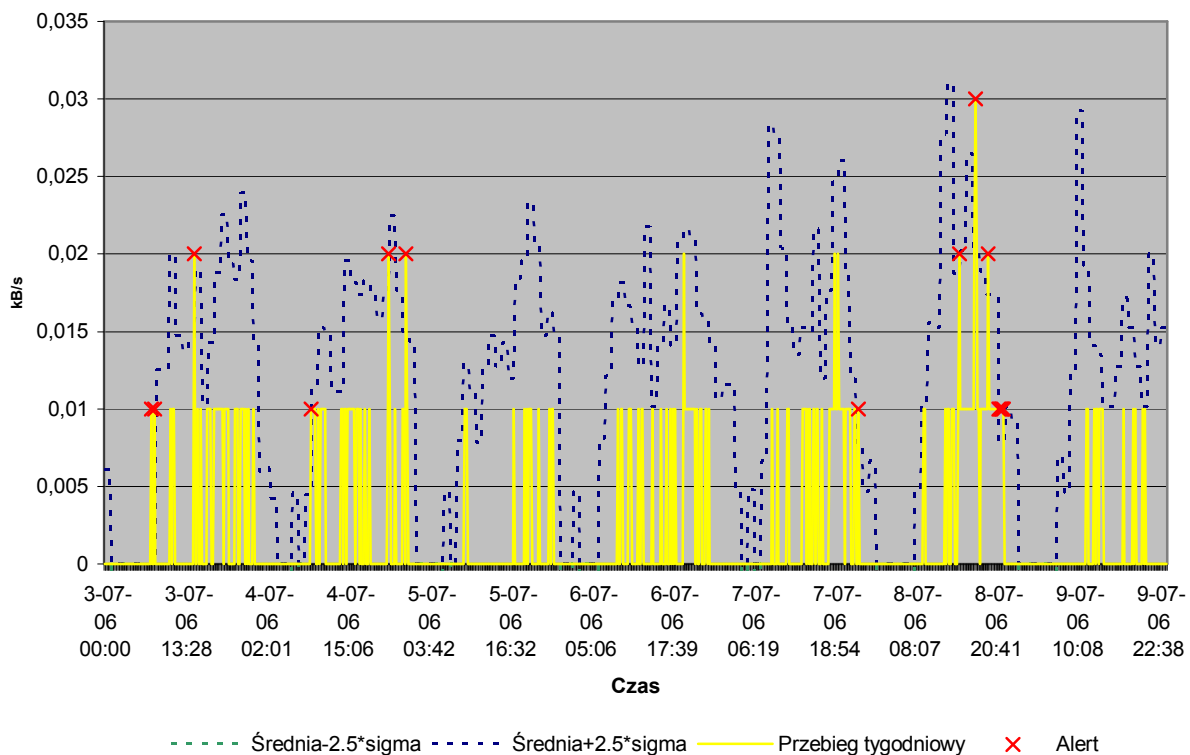


Rysunek 172: Statystyka alertów – ruch UDP (dane wysłane). Źródło: opracowanie własne.

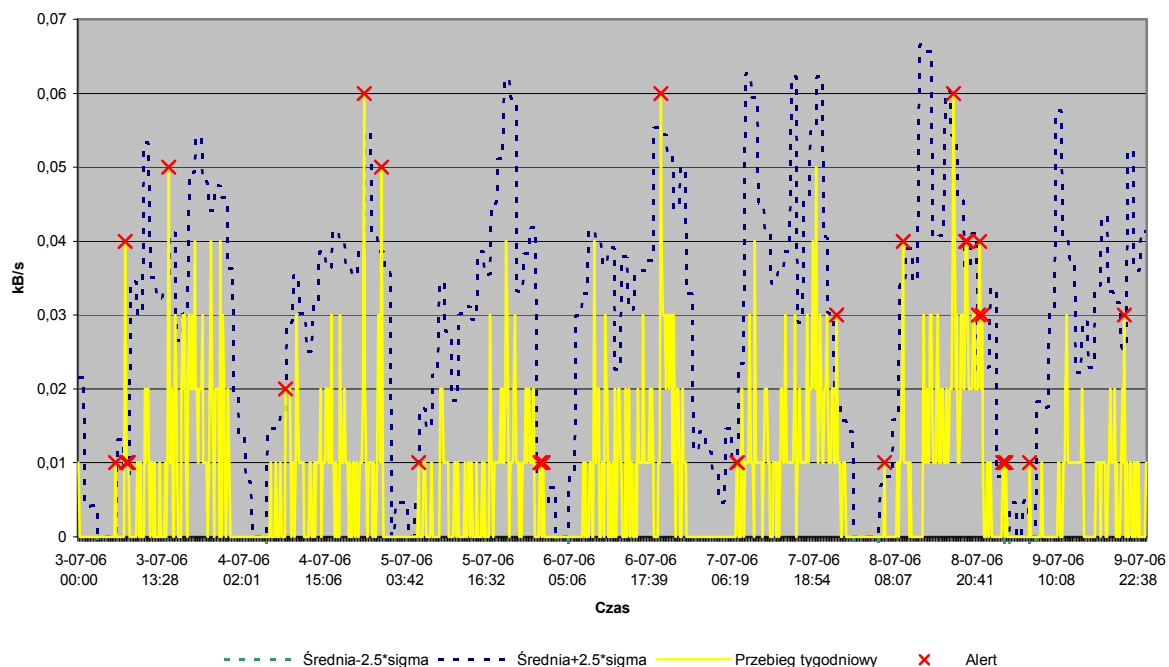


Rysunek 173: Statystyka alertów – ruch UDP (dane odebrane). Źródło: opracowanie własne.



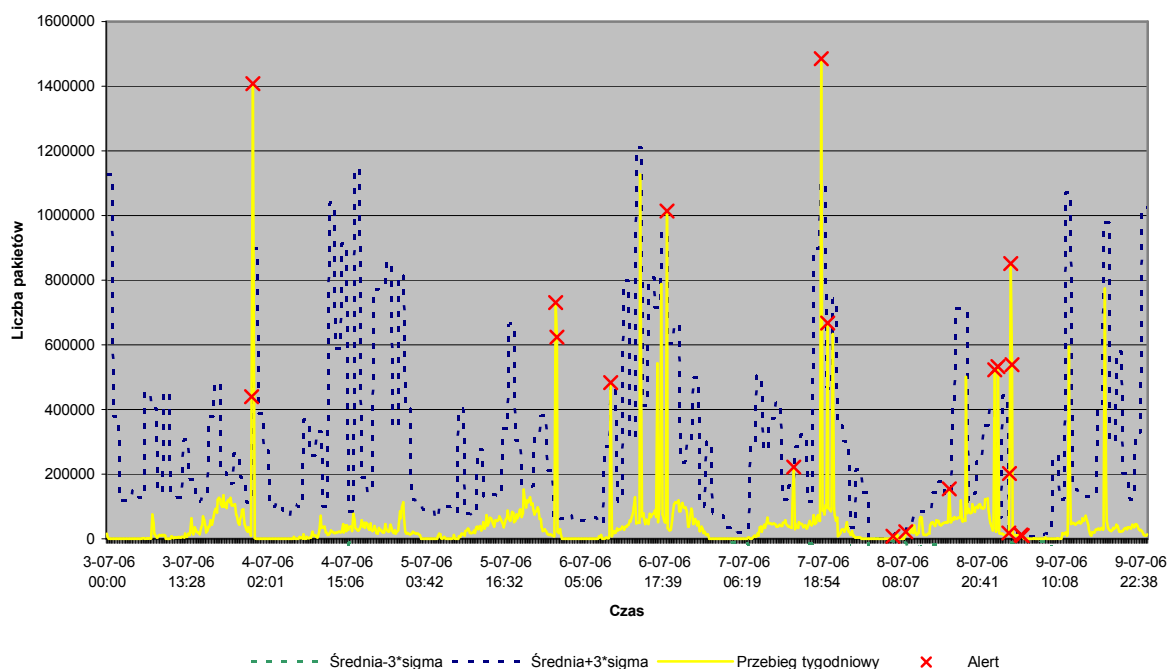


Rysunek 174: Statystyka alertów – ruch UDP port 53 (dane wysłane). Źródło: opracowanie własne.

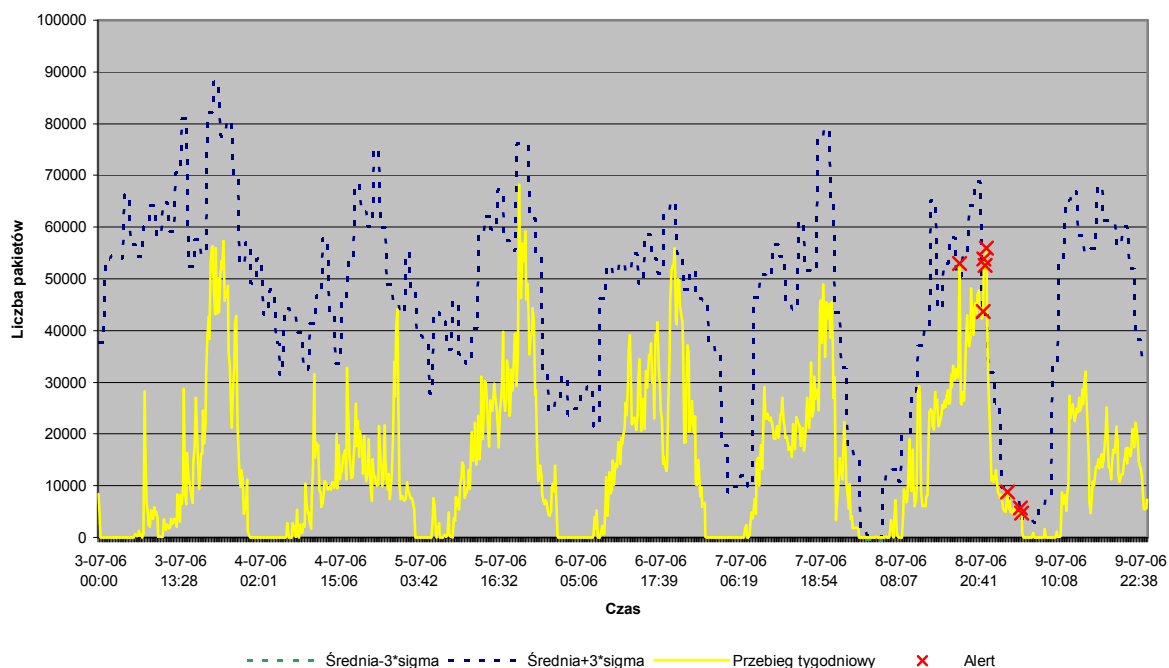


Rysunek 175: Statystyka alertów – ruch UDP port 53 (dane odebrane). Źródło: opracowanie własne.

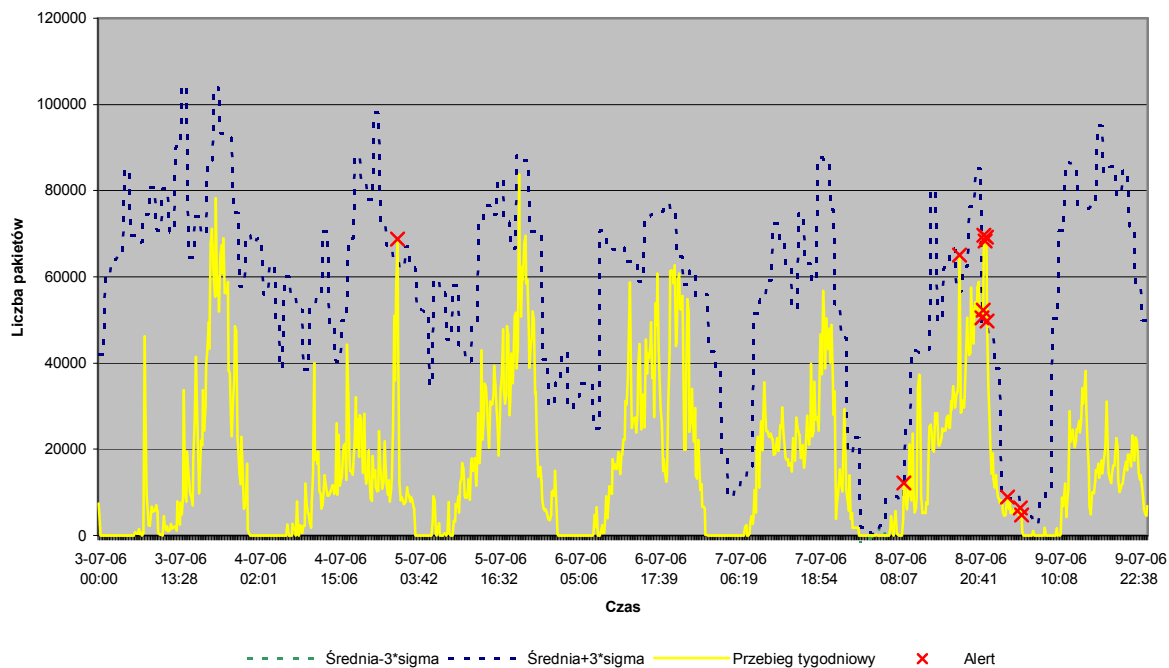
## Załącznik 5: Wykresy dla mnożnika sigmy równego 3.



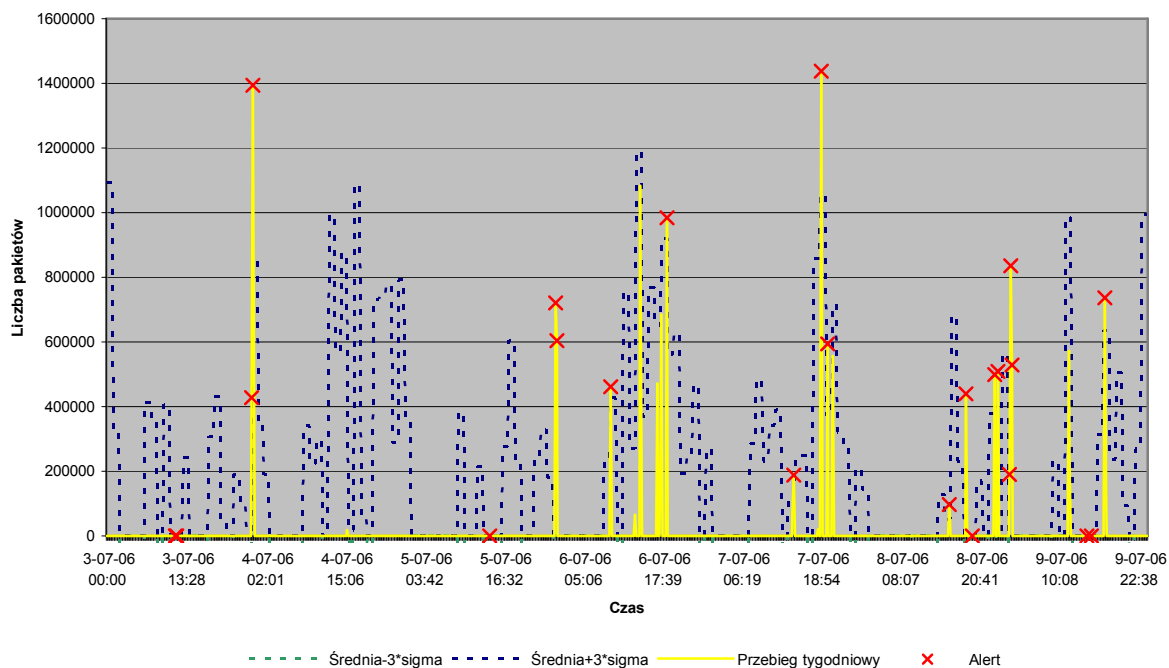
Rysunek 176: Statystyka alertów - ruch TCP. Źródło: opracowanie własne.



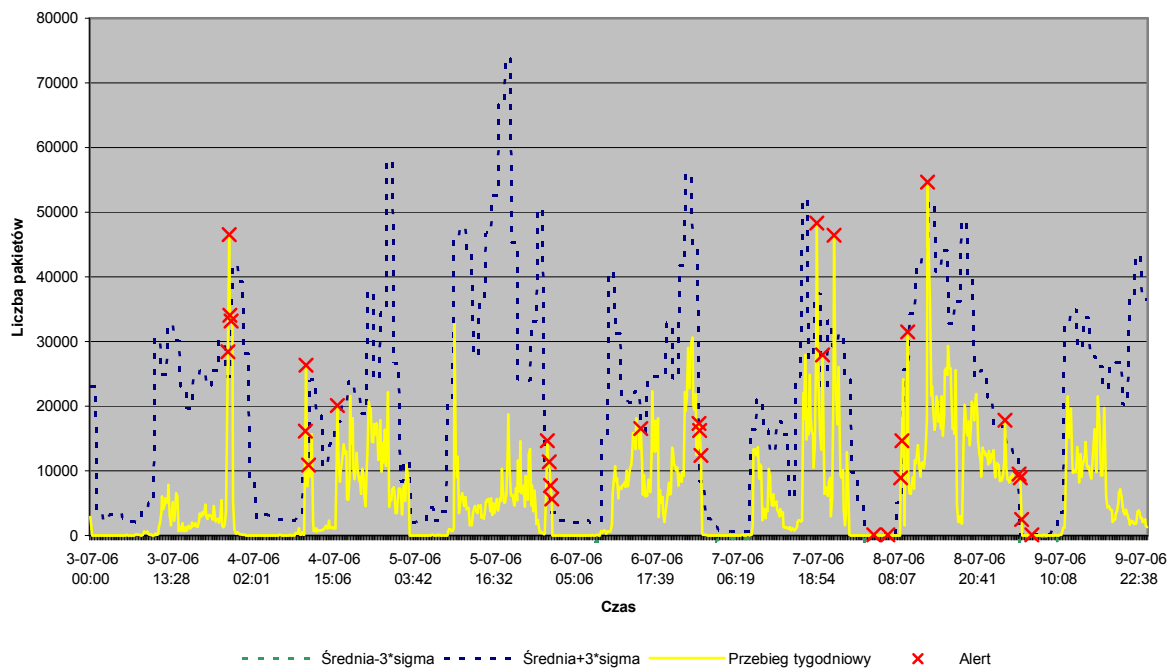
Rysunek 177: Statystyka alertów – wysłane pakiety TCP. Źródło: opracowanie własne.



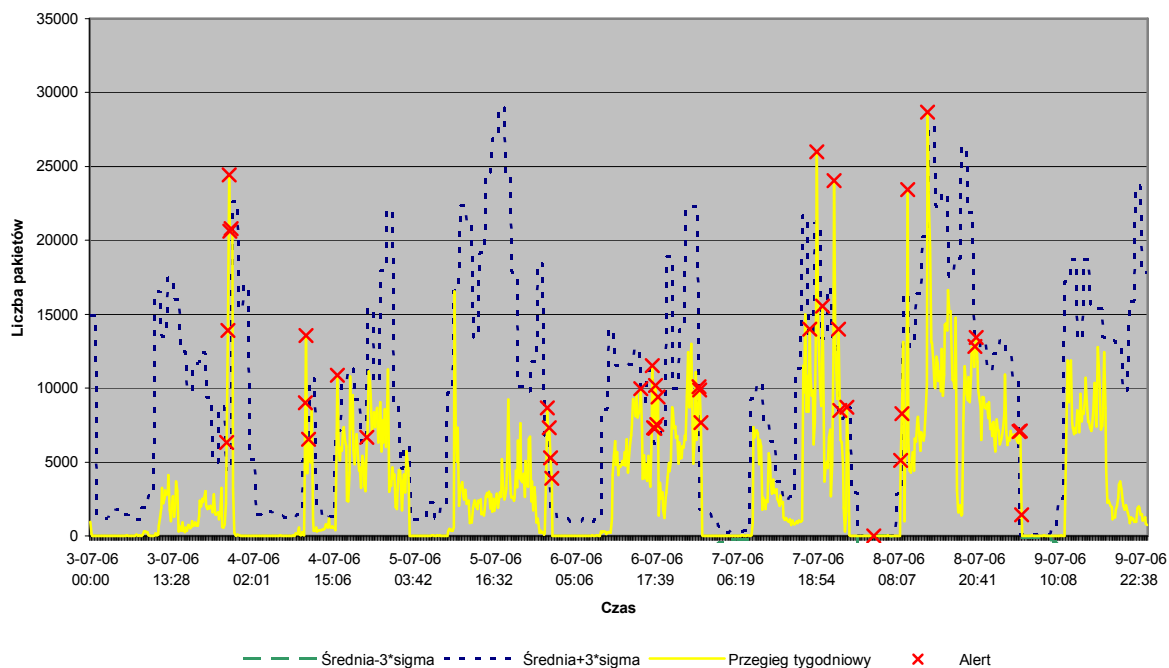
Rysunek 178: Statystyka alertów – odebrane pakiety TCP. Źródło: opracowanie własne.



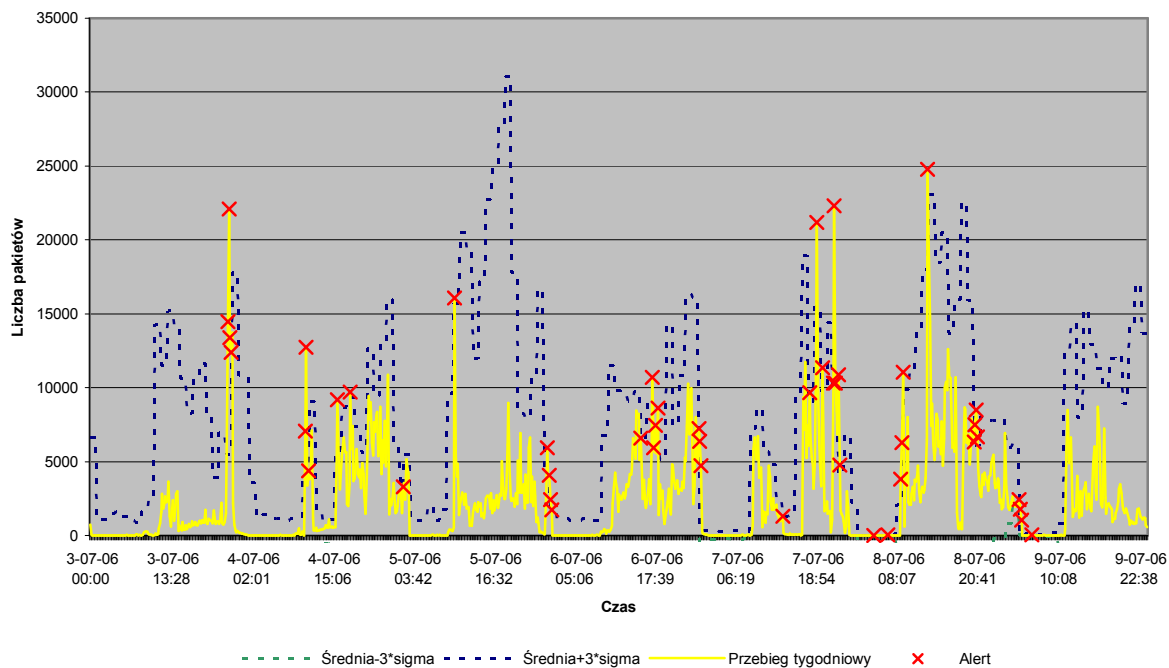
Rysunek 179: Statystyka alertów – pakiety TCP wewnątrz sieci LAN. Źródło: opracowanie własne.



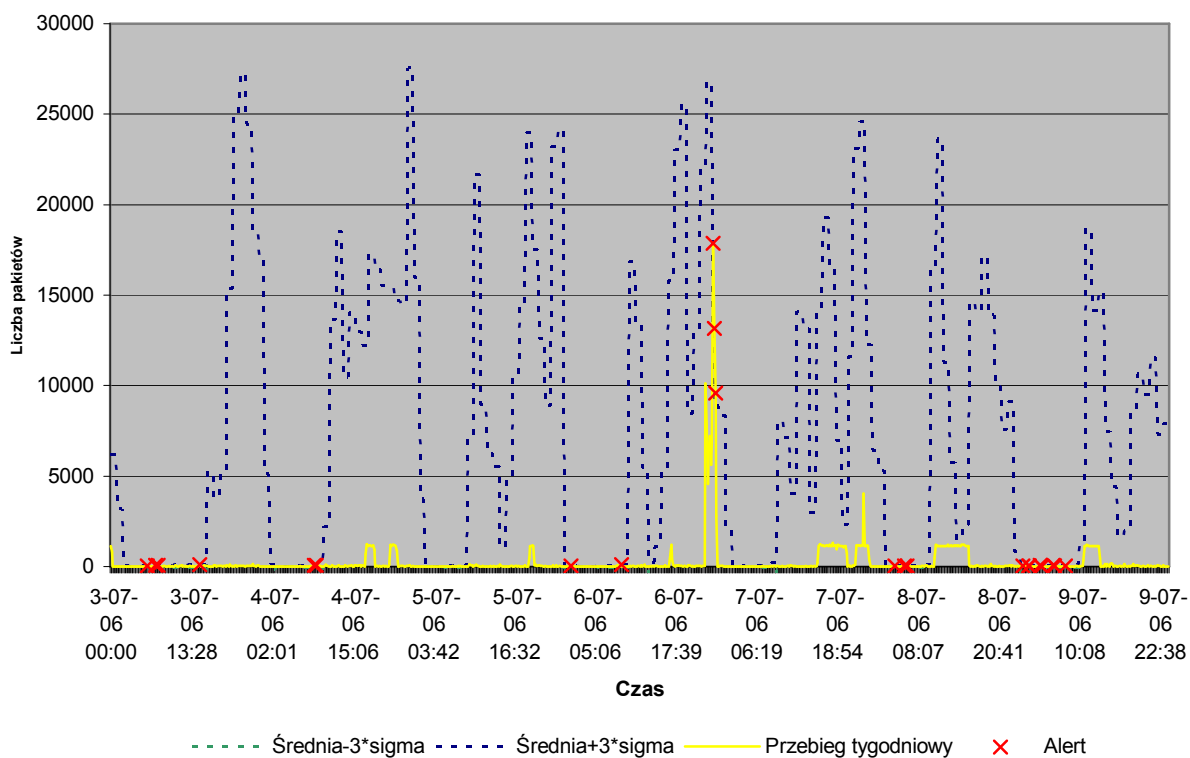
**Rysunek 180: Statystyka alertów – pakiety UDP. Źródło: opracowanie własne.**



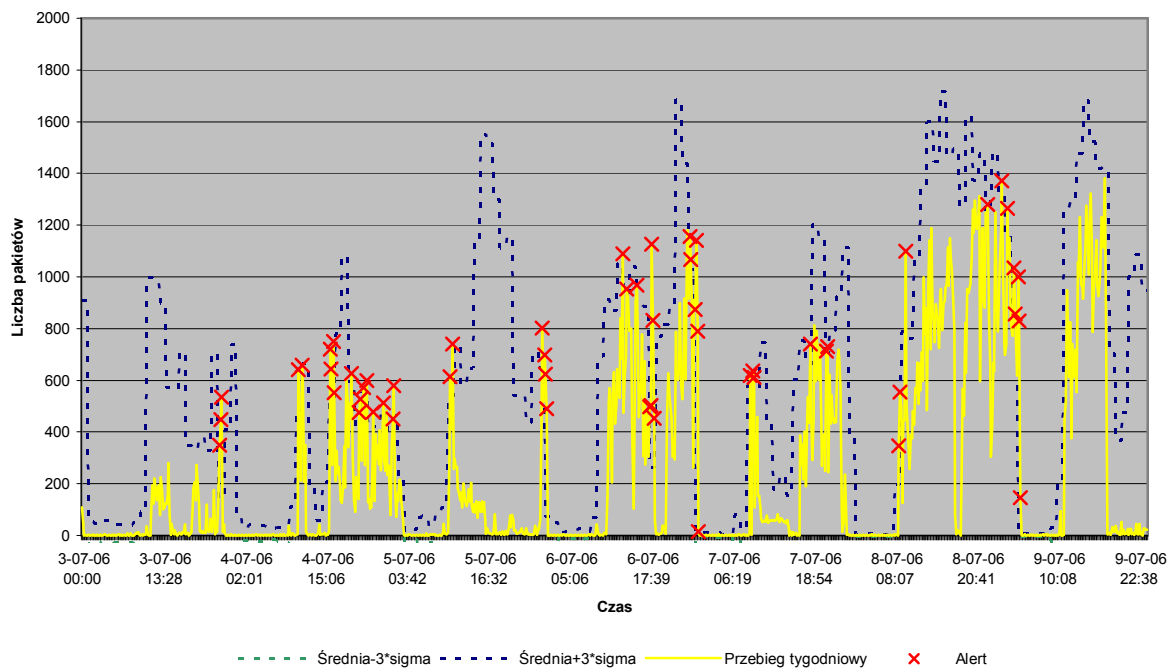
**Rysunek 181: Statystyka alertów – wysłane pakiety UDP. Źródło: opracowanie własne.**



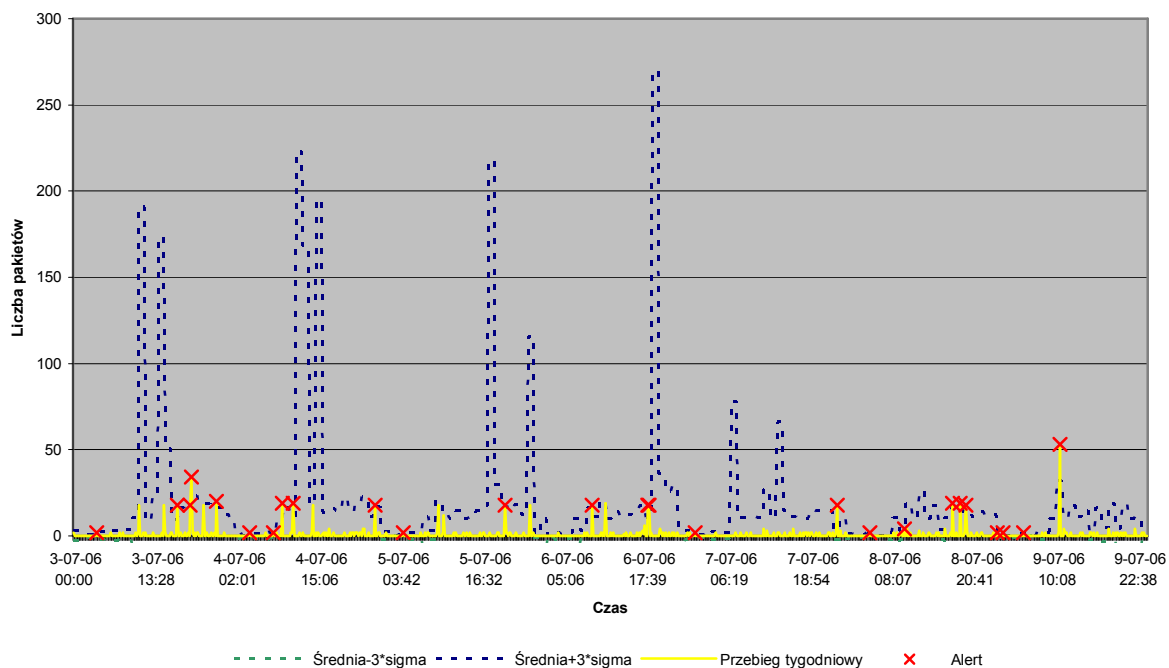
**Rysunek 182: Statystyka alertów – odebrane pakiety UDP. Źródło: opracowanie własne.**



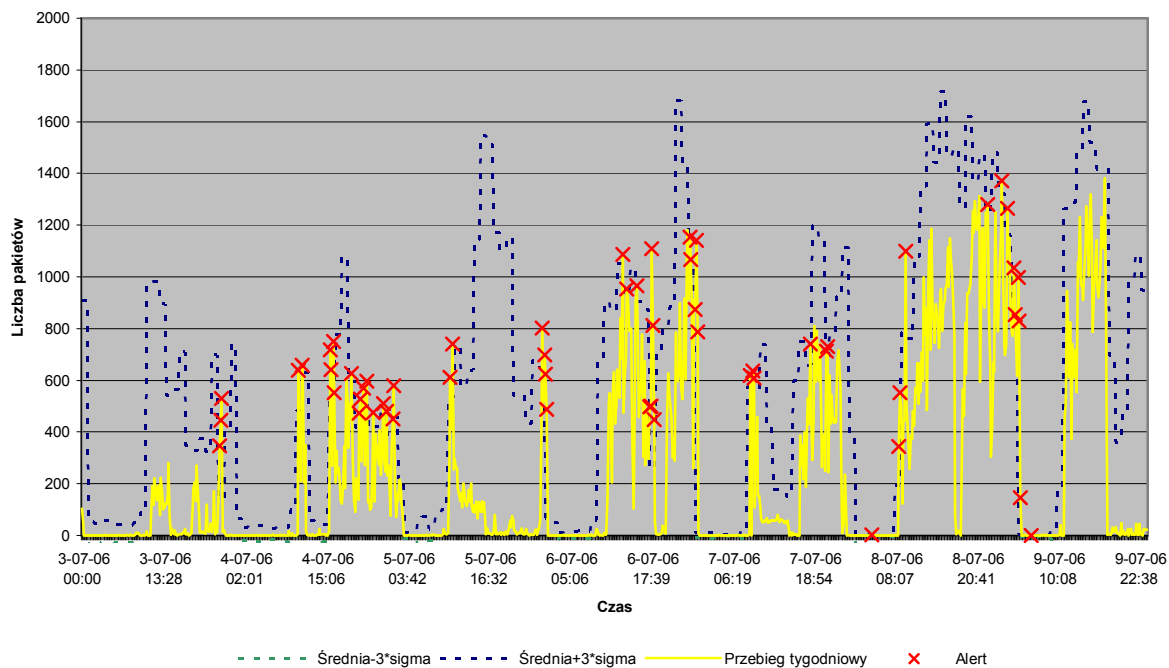
**Rysunek 183: Statystyka alertów – pakiety UDP wewnątrz sieci LAN. Źródło: opracowanie własne.**



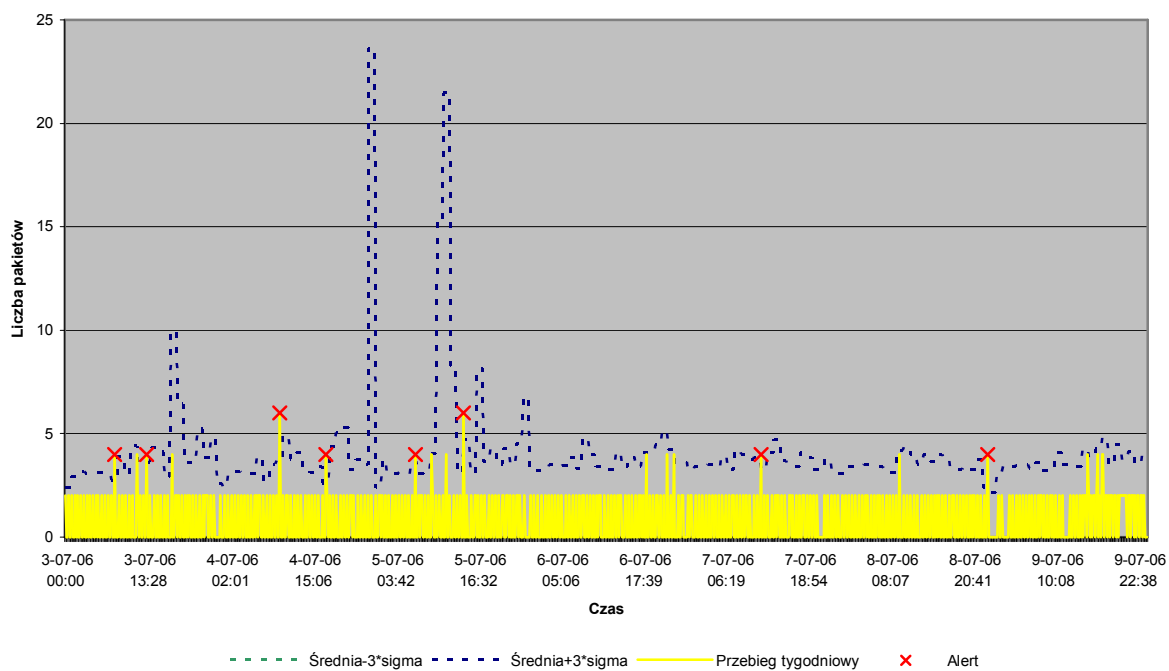
Rysunek 184: Statystyka alertów – pakiety ICMP. Źródło: opracowanie własne.



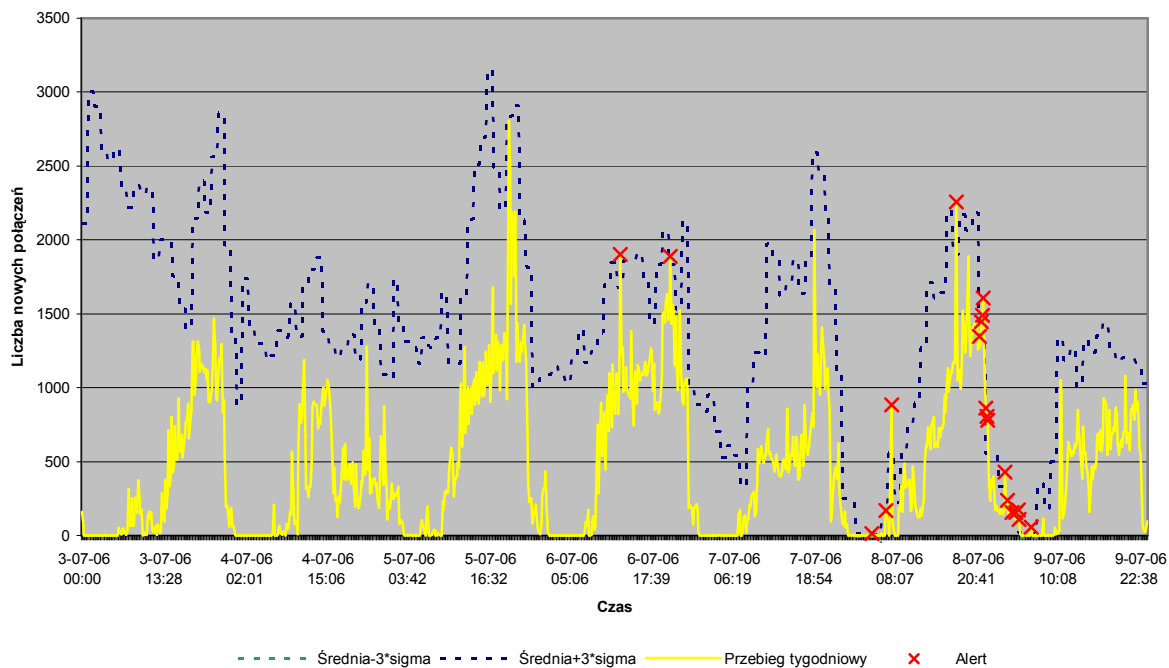
Rysunek 185: Statystyka alertów – wysłane pakiety ICMP. Źródło: opracowanie własne.



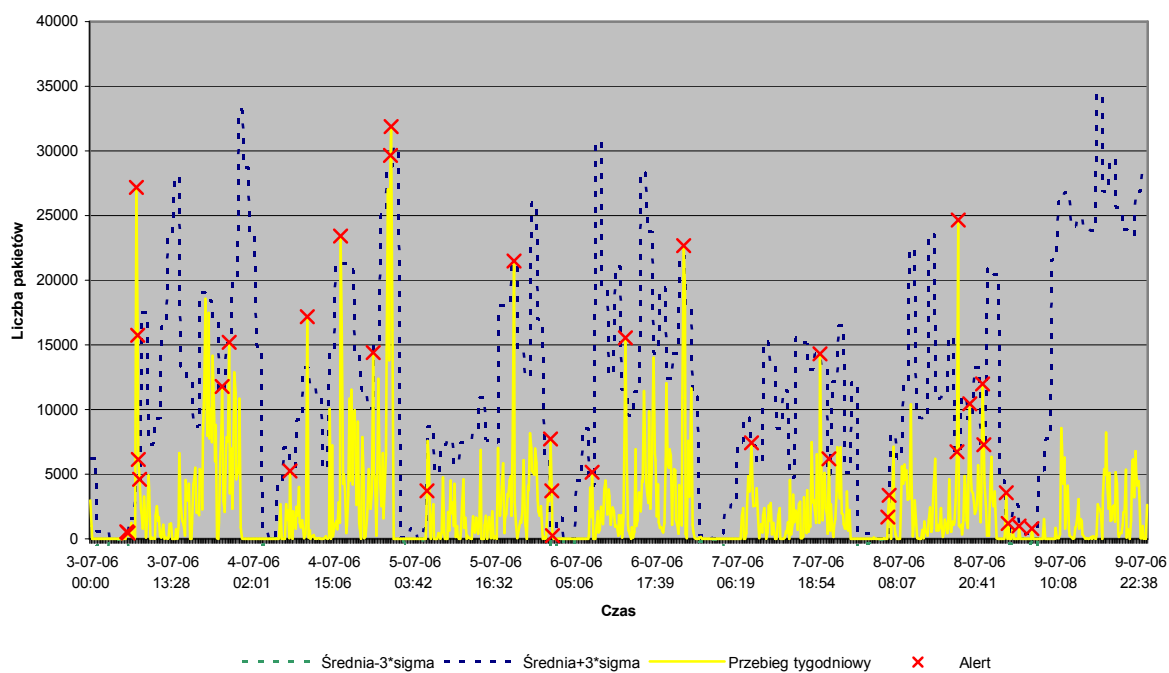
**Rysunek 186: Statystyka alertów – odebrane pakiety ICMP. Źródło: opracowanie własne.**



**Rysunek 187: Statystyka alertów – pakiety ICMP wewnątrz sieci LAN. Źródło: opracowanie własne.**

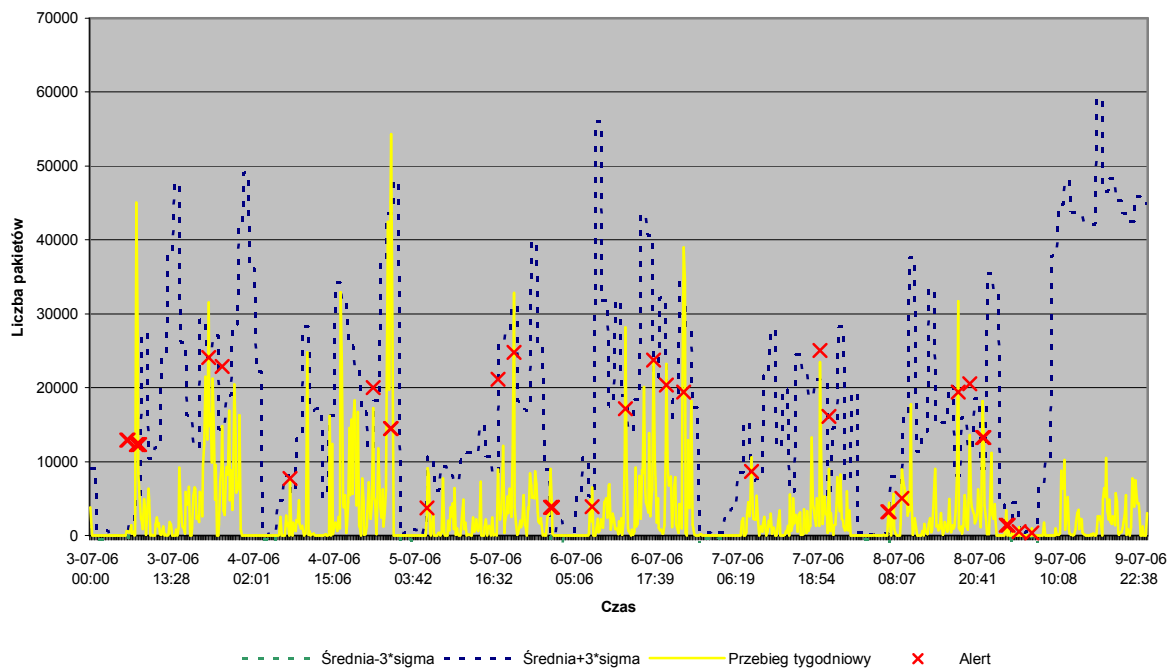


**Rysunek 188: Statystyka alertów – nowe połączenia (TCP z flagami SYN i ACK). Źródło: opracowanie własne.**

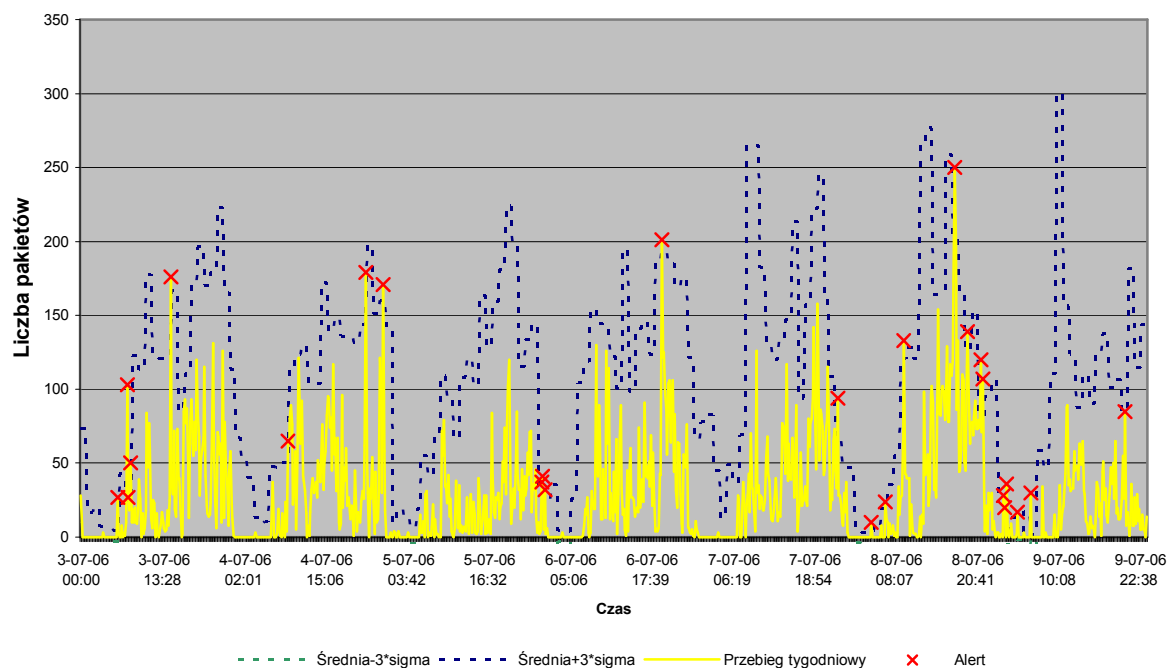


**Rysunek 189: Statystyka alertów – wysłane pakiety TCP (port 80). Źródło: opracowanie własne.**

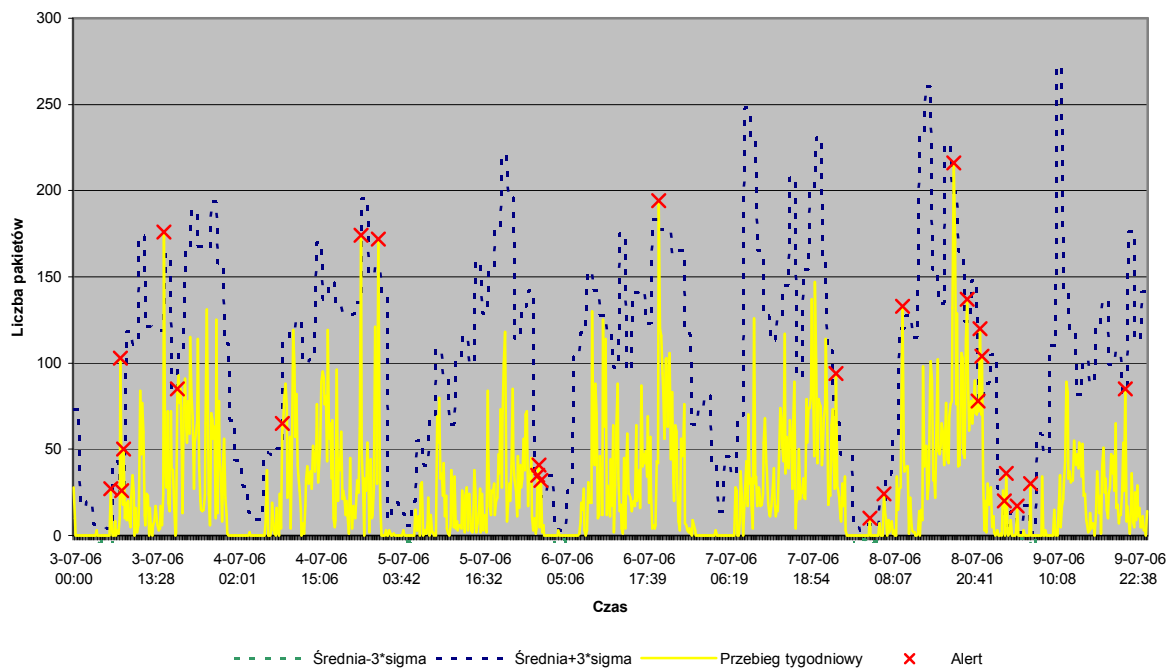




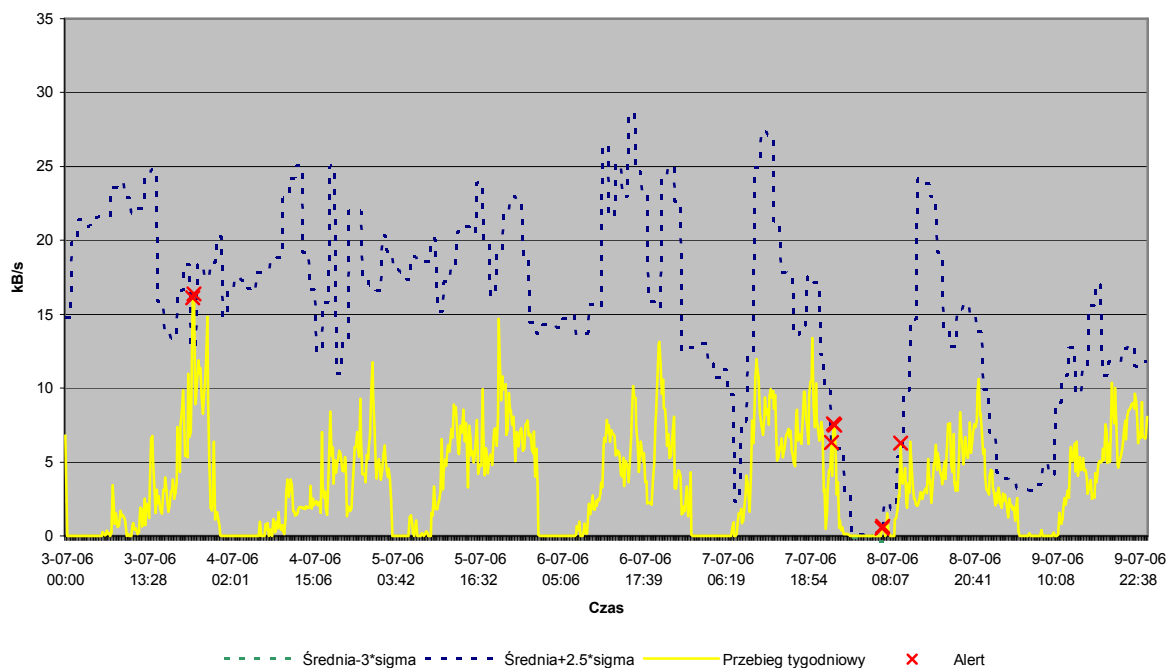
**Rysunek 190: Statystyka alertów – odebrane pakiety TCP (port 80). Źródło: opracowanie własne.**



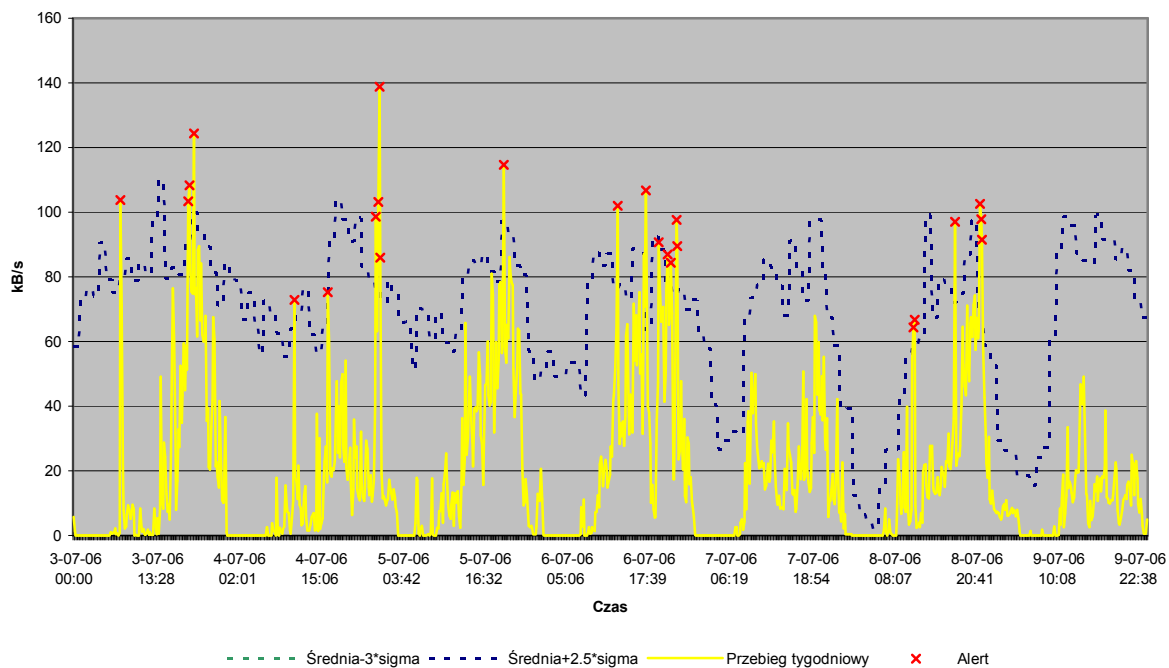
**Rysunek 191: Statystyka alertów – wysłane pakiety UDP (port 53). Źródło: opracowanie własne.**



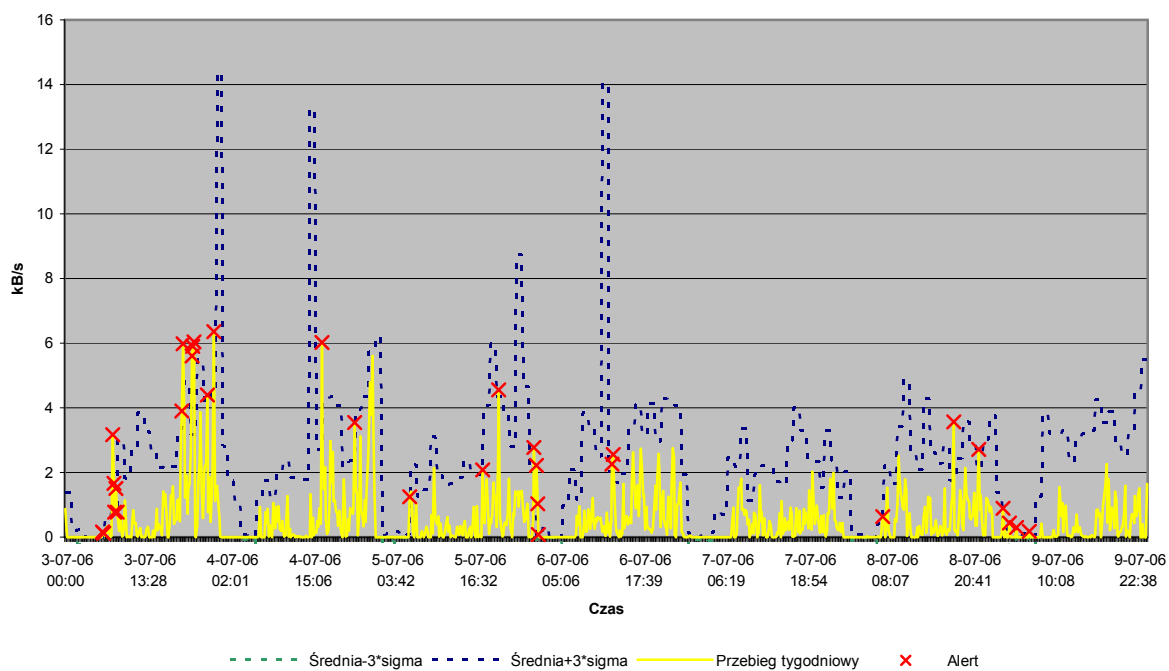
**Rysunek 192: Statystyka alertów – odebrane pakiety UDP (port 53). Źródło: opracowanie własne.**



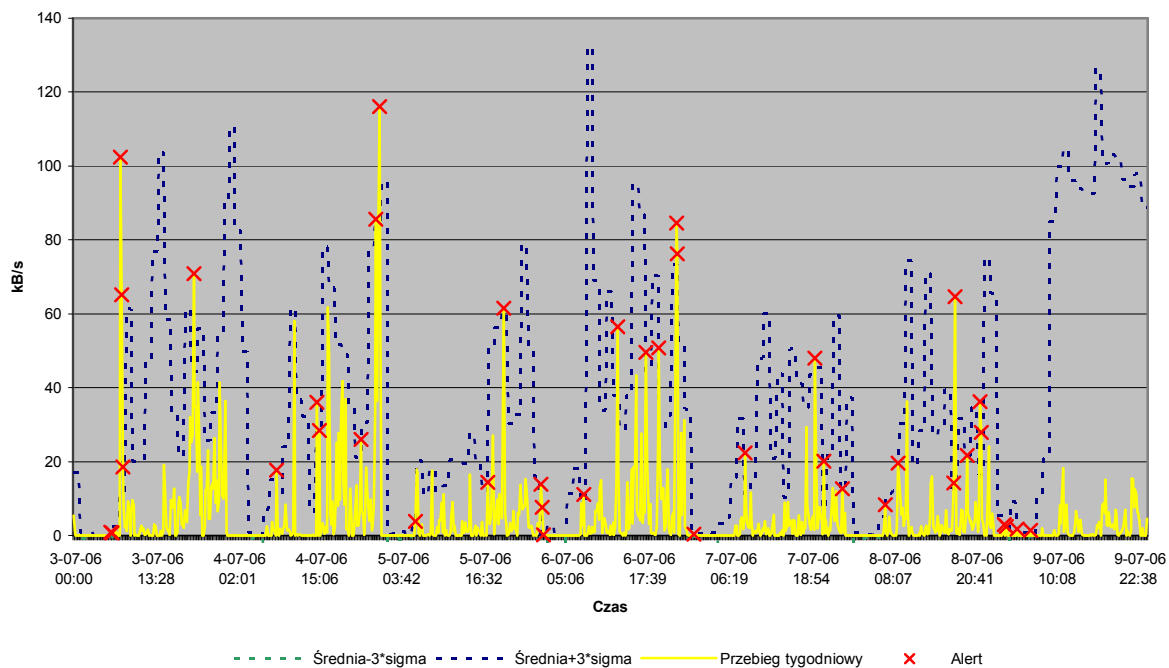
**Rysunek 193: Statystyka alertów – ruch TCP (dane wysłane). Źródło: opracowanie własne.**



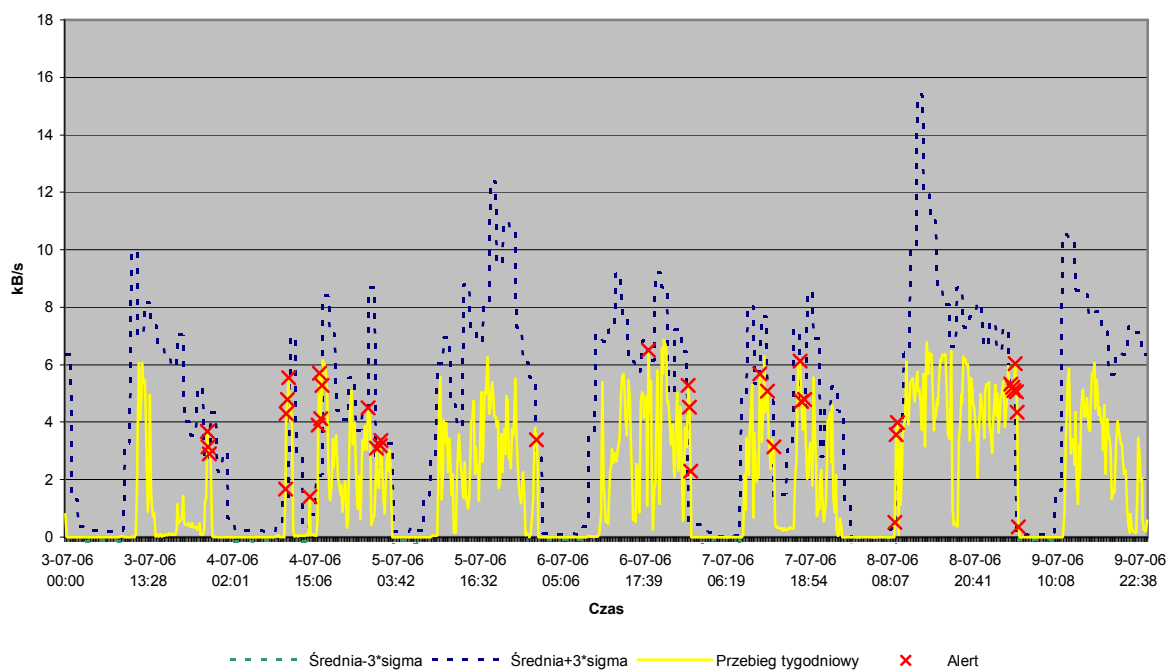
**Rysunek 194: Statystyka alertów – ruch TCP (dane odebrane). Źródło: opracowanie własne.**



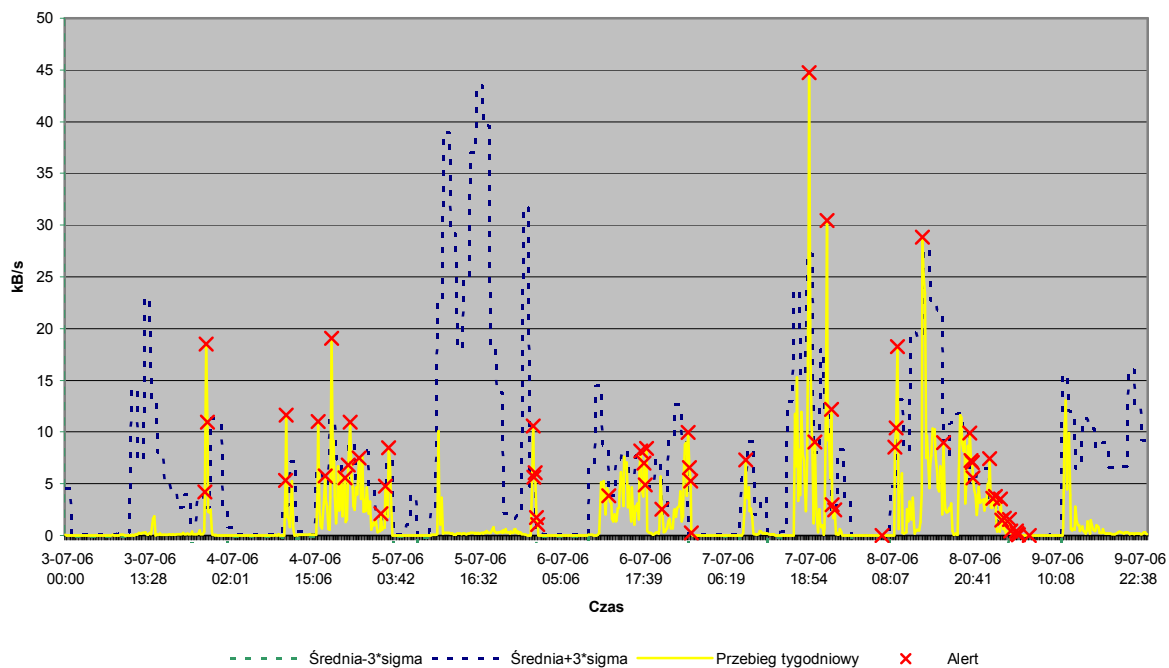
**Rysunek 195: Statystyka alertów – ruch WWW (dane wysłane). Źródło: opracowanie własne.**



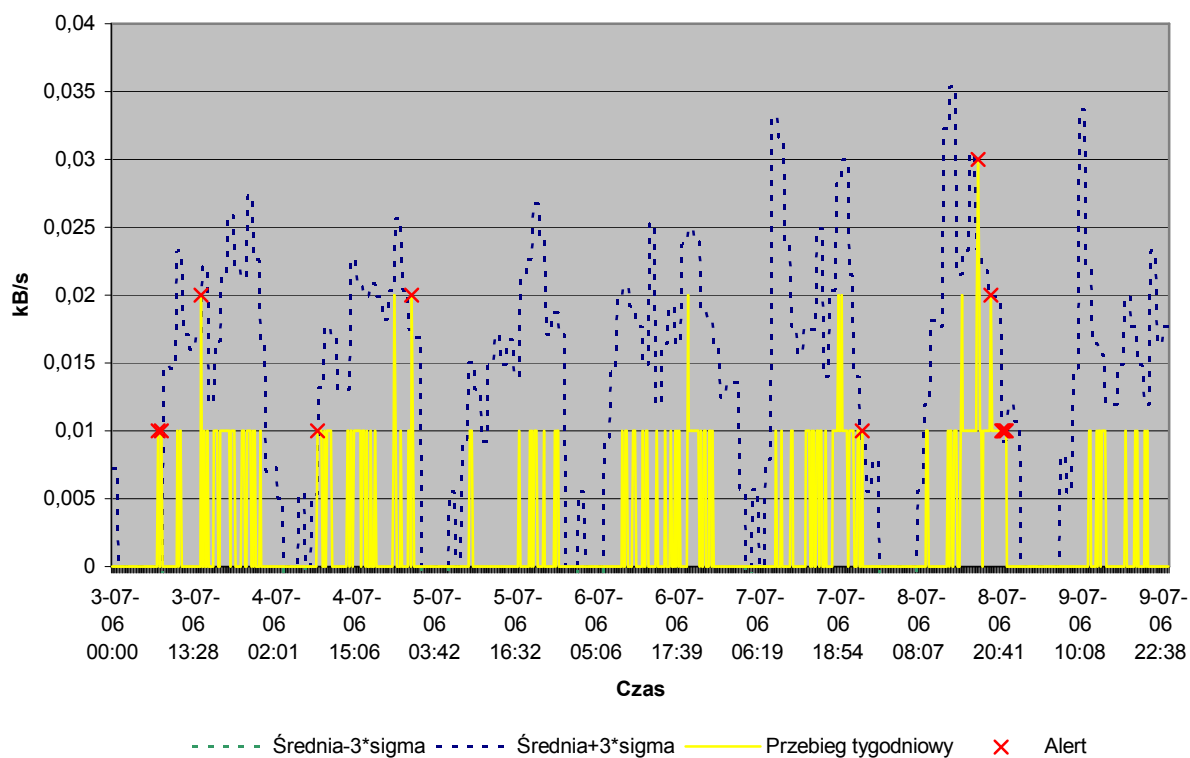
**Rysunek 196: Statystyka alertów – ruch WWWW (dane odebrane). Źródło: opracowanie własne.**



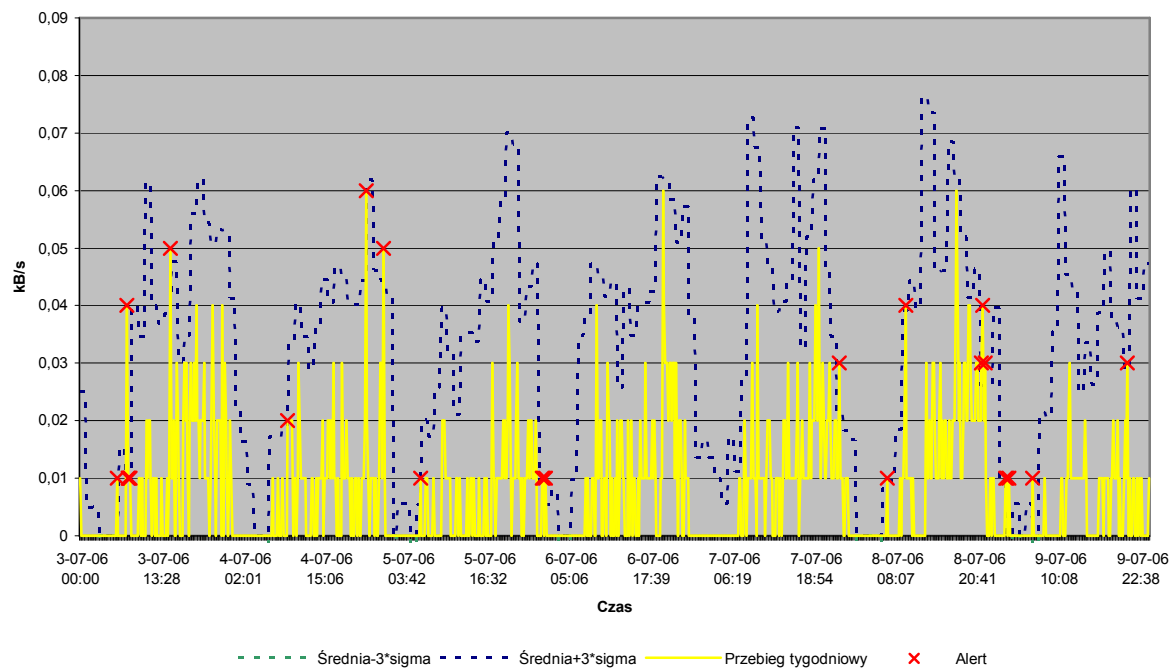
**Rysunek 197: Statystyka alertów – ruch UDP (dane wysłane). Źródło: opracowanie własne.**



**Rysunek 198: Statystyka alertów – ruch UDP (dane odebrane). Źródło: opracowanie własne.**

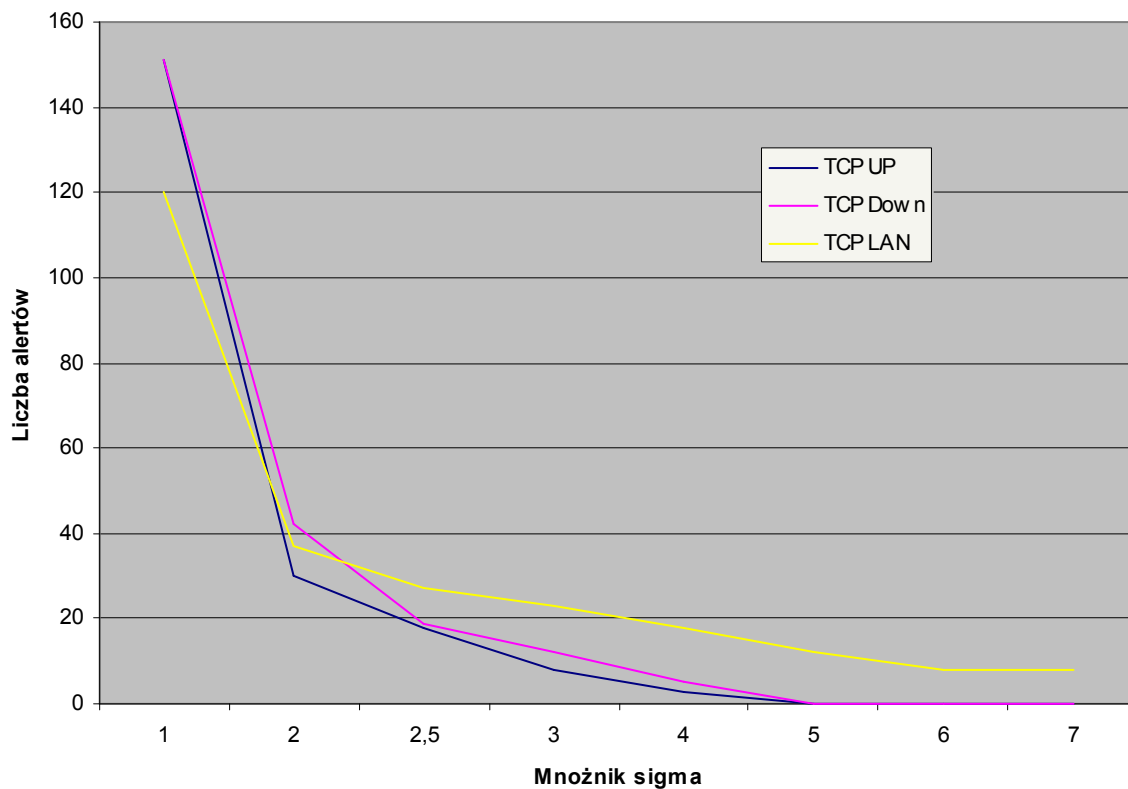


**Rysunek 199: Statystyka alertów – ruch UDP port 53 (dane wysłane). Źródło: opracowanie własne.**

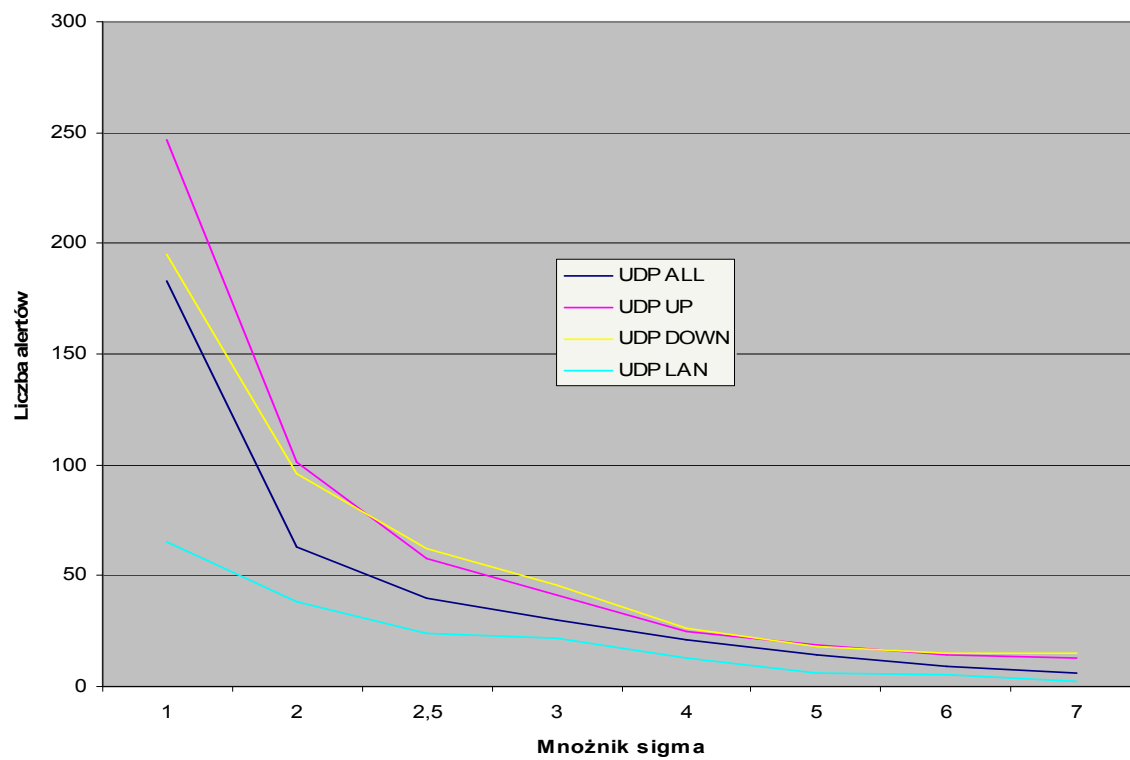


Rysunek 200: Statystyka alertów – ruch UDP port 53 (dane odebrane). Źródło: opracowanie własne.

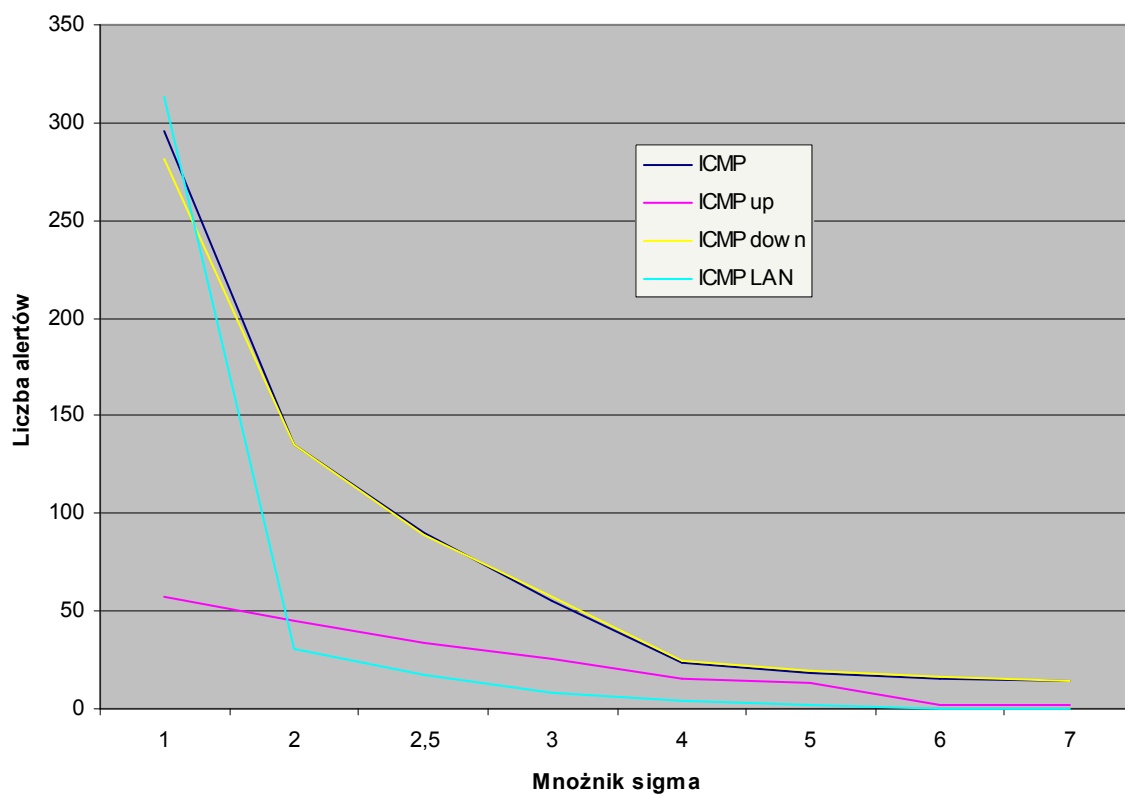
## Załącznik 6: Wykresy zależności liczby alertów od wartość mnożnika sigmy.



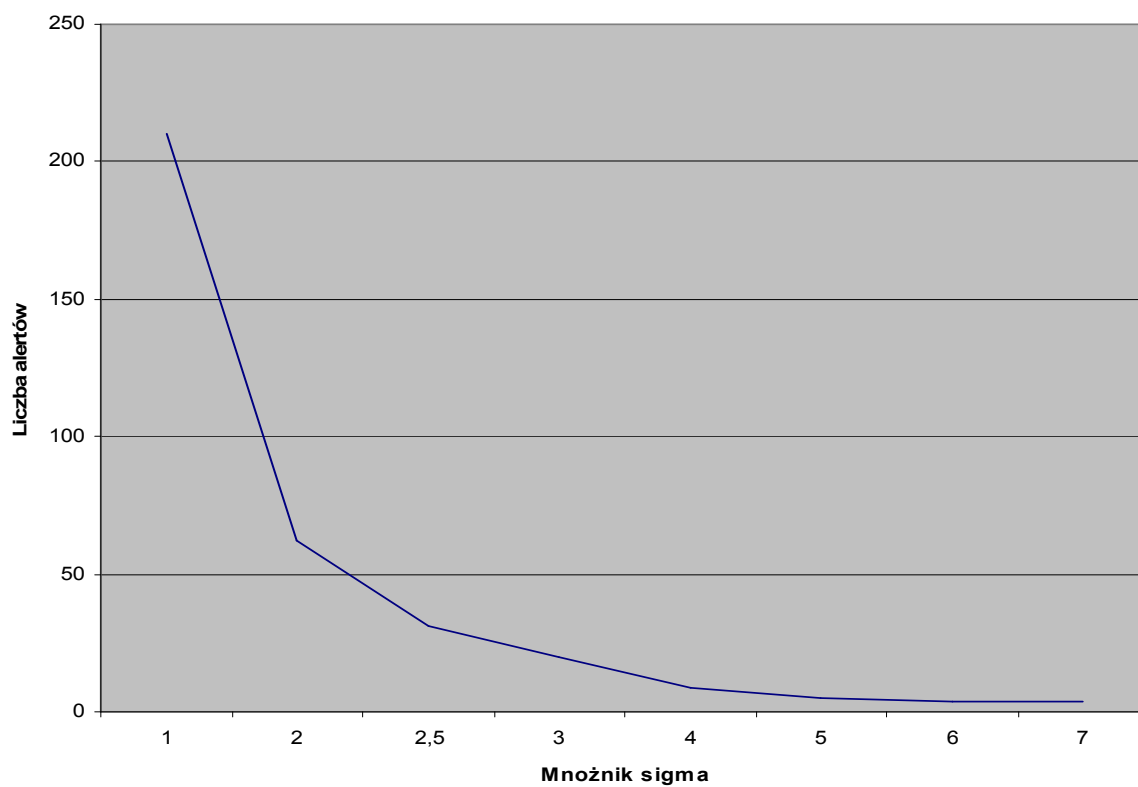
Rysunek 201: Zależność liczby alertów od mnożnika sigmy: ruch TCP. Źródło: opracowanie własne.



Rysunek 202: Zależność liczby alertów od mnożnika sigmy: ruch UDP. Źródło: opracowanie własne.

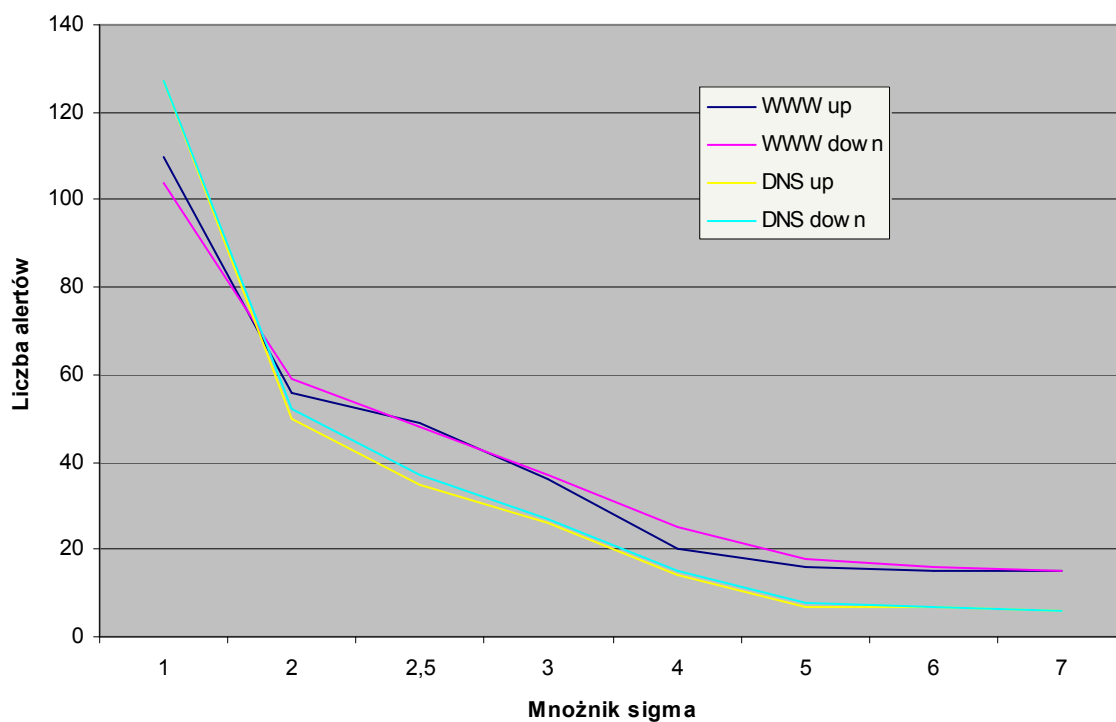


Rysunek 203: Zależność liczby alertów od mnożnika sigmy: ruch ICMP. Źródło: opracowanie własne.

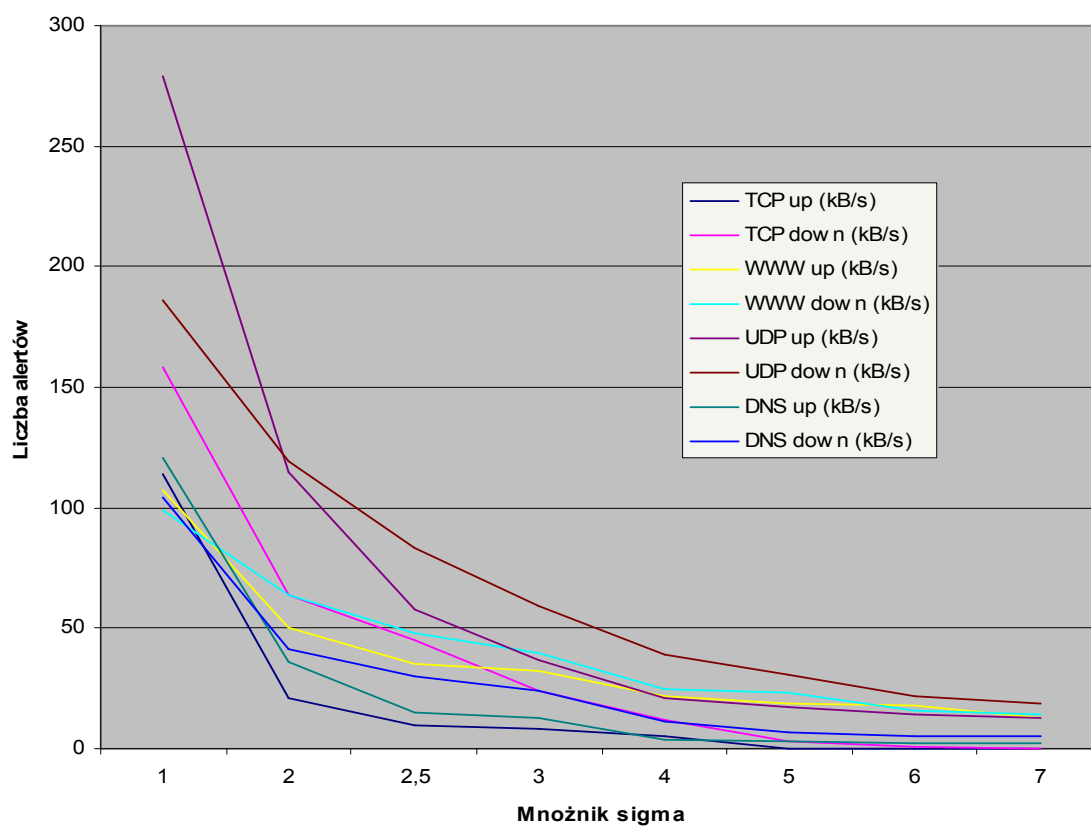


Rysunek 204: Zależność liczby alertów od mnożnika sigmy: liczba SYNACK. Źródło: opracowanie własne.

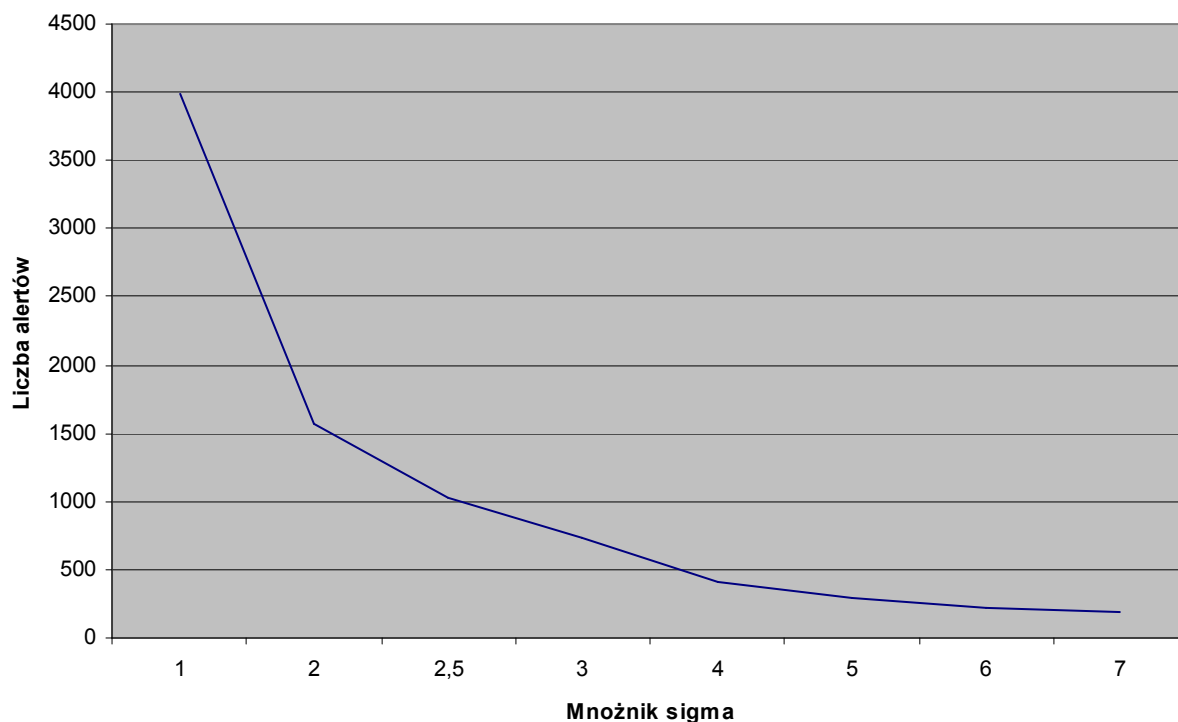




Rysunek 205: Zależność liczby alertów od mnożnika sigmy: ruch WWW i DNS. Źródło: opracowanie własne.



Rysunek 206: Zależność liczby alertów od mnożnika sigmy: szybkość ruchu. Źródło: opracowanie własne.



**Rysunek 207: Zależność liczby alertów od mnożnika sigmy: sumaryczna ilość alertów. Źródło: opracowanie własne.**

## **Załącznik 7: Słownik wyrażen obcojęzycznych i skrótów:**

- ARP - Address Resolution Protocol – protokół pozwalający na znalezienie adresu MAC przy znanym adresie IP
- BASE - Basic Analysis and Security Engine - interfejs do przeglądania alertów wygenerowanych przez Snorta
- DMZ - de-militarized zone – strefa zdemilitaryzowana
- DDoS – atak DoS przeprowadzany z wielu lokalizacji na raz
- DoS – denial of service - odmowa usługi
- Firewall – program realizujący filtrowanie pakietów, ściana przeciwoogniowa
- FTP – protokół transferu plików
- GIDS - Gateway Intrusion Detection System – system IDS będący bramą
- HIDS – Host Intrusion Detection System – system detekcji wtargnięć przeznaczony do pracy na pojedynczym hoście
- Hub – koncentrator sieciowy

- IDES – Intrusion Detection Expert System - opracowany pod kierownictwem Doroty Denning pierwszy prototyp systemu IDS, którym mógł analizować aktywność użytkowników.
- IDS – Intrusion Detection System, system detekcji wtargnięć
- ICMP - Internet Control Message Protocol
- IP – Internet Protocol
- IPS – Intrusion Prevention System – system zapobiegania wtargnięciom
- Iptables – popularna zaporę sieciową stosowaną w systemach operacyjnych z rodziny Unix
- LAN – Local Area Network - sieć lokalna
- MAC address – fizyczny adres interfejsu sieciowego
- NIDS – Network Intrusion Detection System – sieciowy system detekcji intruzów
- Promiscuous – tryb pracy karty sieciowej umożliwiający przechwytywanie pakietów nie zaadresowanych do danego hosta
- RFC – Request For Comments
- Router- urządzenie realizujące proces trasowania pakietów
- SAMBA – usługa sieciowa pozwalająca na wymianę plików i udostępnianie drukarek
- Sniffer – program do podsłuchu sieci
- SNMP - simple network management protocol – prosty protokół zarządzania siecią
- SPAN - Switch Port Analyzer – port w przełączniku, na który może być kopiowany ruch z innych portów tego switcha
- Stealh - niewidzialny
- Switch – przełącznik sieciowy
- TAP – test access port, urządzenie kopiujące ruch sieciowy
- TTL – Time To live – czas życia
- VLAN - Virtual Local Area Network – wirtualna sieć lokalna

## **Załącznik 8: Spis zawartości płyty CD-ROM**

- Pliki źródłowe:
  - Preprocesor AnomalyDetection (pliki: spp\_anomalydetection.c i spp\_anomalydetection.h)
  - Generator profilu ProfileGenerator (plik: profilegenerator.cpp)
  - Skompilowany program ProfileGenerator

- Wersja elektroniczna pracy

## Spis ilustracji:

|                                                                                                                                                      |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Rysunek 1: Schemat systemu aktywnej odpowiedzi. Źródło: opracowanie własne. ....                                                                     | 11 |
| Rysunek 2: Klasyfikacja systemów IDS. Źródło: [Dorosz 2/2002]. ....                                                                                  | 14 |
| Rysunek 3: Sieć chroniona przez HIDS typu menadżer/agent. Źródło: [Axent 1999]. ....                                                                 | 16 |
| Rysunek 4: NIDS przed główną zaporą sieciową (według: [Szmit 2005], str. 501).....                                                                   | 18 |
| Rysunek 5: NIDS w obrębie strefy DMZ (według:[Szmit 2005], str. 501) .....                                                                           | 18 |
| Rysunek 6: NIDS w obrębie sieci korporacyjnej (według: [Szmit 2005], str. 501).....                                                                  | 19 |
| Rysunek 7: NIDS w każdym z wymienionych powyżej punktów sieci (według: [Szmit 2005], str. 501).....                                                  | 19 |
| Rysunek 8: Ruch z całej sieci jest przechwycony przez system IDS z interfejsem w trybie promiscuous. Źródło: opracowanie własne. ....                | 21 |
| Rysunek 9: TAP. Źródło: [16].....                                                                                                                    | 22 |
| Rysunek 10: Ruch sieciowy generowany między siecią LAN i Internetem zawsze przepływa przez system IDS typu in-line. Źródło: opracowanie własne. .... | 24 |
| Rysunek 11: Architektura wielowarstwowa. Źródło: [Endorf 2004]. ....                                                                                 | 28 |
| Rysunek 12: Schemat systemu IDS z dedykowaną siecią. Źródło: opracowanie własne. ....                                                                | 33 |
| Rysunek 13: Przykładowy schemat umiejscowienia systemu Honeypot w sieci. Źródło: opracowanie własne. ....                                            | 37 |
| Rysunek 14: Ogólna budowa reguł (według: [Rehman 2003] str. 79).....                                                                                 | 43 |
| Rysunek 15: Składnia nagłówka reguły (według: [Rehman 2003] str. 79).....                                                                            | 43 |
| Rysunek 16: Przepływ danych w programie Snort. Źródło: [15]. ....                                                                                    | 45 |
| Rysunek 17: Okno główne interfejsu BASE. Źródło: opracowanie własne.....                                                                             | 49 |
| Rysunek 18: Podgląd pakietu w interfejsie BASE. Źródło: opracowanie własne.....                                                                      | 50 |
| Rysunek 19: Przykładowy wykres obciążenia łącza w ciągu dnia. Źródło: opracowanie własne. ....                                                       | 52 |
| Rysunek 20: Przykład alertów wygenerowanych przez preprocesor AnomalyDetection. Źródło: opracowanie własne.....                                      | 63 |
| Rysunek 21: Strona domowa opracowanego systemu. Źródło: opracowanie własne.....                                                                      | 64 |
| Rysunek 22: Informacje pokazywane przy wyłączaniu preprocesora AnomalyDetection. Źródło: opracowanie własne.....                                     | 67 |
| Rysunek 23: Schemat sieci, w której zostały przeprowadzone pomiary. Źródło: opracowanie własne. ....                                                 | 70 |
| Rysunek 24: Statystyka wysłanych pakietów TCP (przebieg dobowy). Źródło: opracowanie własne. ....                                                    | 72 |
| Rysunek 25: Statystyka wysłanych pakietów TCP (przebieg tygodniowy). Źródło: opracowanie własne.....                                                 | 72 |
| Rysunek 26: Zależność liczby alertów od mnożnika sigmy. Źródło: opracowanie własne. ....                                                             | 78 |
| Rysunek 27: Spadek liczby generowanych alertów przy zmianie mnożnika sigmy. Źródło: opracowanie własne.....                                          | 79 |
| Rysunek 28: Statystyka pakietów TCP (przebieg tygodniowy). Źródło: opracowanie własne.....                                                           | 83 |
| Rysunek 29: Statystyka pakietów TCP (przebieg dobowy). Źródło: opracowanie własne.....                                                               | 83 |
| Rysunek 30: Statystyka odebranych pakietów TCP (przebieg tygodniowy). Źródło: opracowanie własne.....                                                | 84 |
| Rysunek 31: Statystyka odebranych pakietów TCP (przebieg dobowy). Źródło: opracowanie własne. ....                                                   | 84 |

|                                                                                                                 |    |
|-----------------------------------------------------------------------------------------------------------------|----|
| Rysunek 32: Statystyka wysłanych pakietów TCP (przebieg dobowy). Źródło: opracowanie własne.                    | 85 |
| Rysunek 33: Statystyka pakietów TCP wewnątrz sieci LAN (przebieg tygodniowy). Źródło: opracowanie własne.       | 85 |
| Rysunek 34: Statystyka pakietów TCP wewnątrz sieci LAN (przebieg dobowy). Źródło: opracowanie własne.           | 86 |
| Rysunek 35: Statystyka pakietów UDP (przebieg tygodniowy). Źródło: opracowanie własne.                          | 86 |
| Rysunek 36: Statystyka pakietów UDP (przebieg dobowy). Źródło: opracowanie własne.                              | 87 |
| Rysunek 37: Statystyka wysłanych pakietów UDP (przebieg tygodniowy). Źródło: opracowanie własne.                | 88 |
| Rysunek 38: Statystyka wysłanych pakietów UDP (przebieg dobowy). Źródło: opracowanie własne.                    | 88 |
| Rysunek 39: Statystyka odebranych pakietów UDP (przebieg tygodniowy). Źródło: opracowanie własne.               | 88 |
| Rysunek 40: Statystyka odebranych pakietów UDP (przebieg dobowy). Źródło: opracowanie własne.                   | 89 |
| Rysunek 41: Statystyka pakietów ICMP (przebieg tygodniowy). Źródło: opracowanie własne.                         | 89 |
| Rysunek 42: Statystyka pakietów ICMP (przebieg dobowy). Źródło: opracowanie własne.                             | 90 |
| Rysunek 43: Statystyka wysłanych pakietów ICMP (przebieg tygodniowy). Źródło: opracowanie własne.               | 90 |
| Rysunek 44: Statystyka wysłanych pakietów ICMP (przebieg dobowy). Źródło: opracowanie własne.                   | 91 |
| Rysunek 45: Statystyka odebranych pakietów ICMP (przebieg tygodniowy). Źródło: opracowanie własne.              | 91 |
| Rysunek 46: Statystyka odebranych pakietów ICMP (przebieg dobowy). Źródło: opracowanie własne.                  | 92 |
| Rysunek 47: Statystyka pakietów ICMP wewnątrz sieci LAN (przebieg tygodniowy). Źródło: opracowanie własne.      | 92 |
| Rysunek 48: Statystyka pakietów ICMP wewnątrz sieci LAN (przebieg dobowy). Źródło: opracowanie własne.          | 93 |
| Rysunek 49: Liczba nowych połączeń (przebieg tygodniowy). Źródło: opracowanie własne.                           | 93 |
| Rysunek 50: Liczba nowych połączeń (przebieg dobowy). Źródło: opracowanie własne.                               | 94 |
| Rysunek 51: Statystyka wysłanych pakietów TCP port 80 (WWW) (przebieg tygodniowy). Źródło: opracowanie własne.  | 94 |
| Rysunek 52: Statystyka wysłanych pakietów TCP port 80 (WWW) (przebieg dobowy). Źródło: opracowanie własne.      | 95 |
| Rysunek 53: Statystyka odebranych pakietów TCP port 80 (WWW) (przebieg tygodniowy). Źródło: opracowanie własne. | 95 |
| Rysunek 54: Statystyka odebranych pakietów TCP port 80 (WWW) (przebieg dobowy). Źródło: opracowanie własne.     | 96 |
| Rysunek 55: Statystyka wysłanych pakietów UDP port 53 (DNS) (przebieg tygodniowy). Źródło: opracowanie własne.  | 96 |
| Rysunek 56: Statystyka wysłanych pakietów UDP port 53 (DNS) (przebieg dobowy). Źródło: opracowanie własne.      | 97 |
| Rysunek 57: Statystyka odebranych pakietów UDP port 53 (DNS) (przebieg tygodniowy). Źródło: opracowanie własne. | 97 |
| Rysunek 58: Statystyka odebranych pakietów UDP port 53 (DNS) (przebieg dobowy). Źródło: opracowanie własne.     | 98 |

|                                                                                                                     |     |
|---------------------------------------------------------------------------------------------------------------------|-----|
| Rysunek 59: Statystyka wysyłania pakietów TCP (przebieg tygodniowy). Źródło:<br>opracowanie własne.....             | 98  |
| Rysunek 60: Statystyka wysyłania pakietów TCP (przebieg dobowy). Źródło: opracowanie<br>własne.....                 | 99  |
| Rysunek 61: Statystyka odbierania pakietów TCP (przebieg tygodniowy). Źródło:<br>opracowanie własne.....            | 99  |
| Rysunek 62: Statystyka odbierania pakietów TCP (przebieg dobowy). Źródło: opracowanie<br>własne.....                | 100 |
| Rysunek 63: Statystyka wysyłania pakietów TCP na port 80 (przebieg tygodniowy). Źródło:<br>opracowanie własne.....  | 100 |
| Rysunek 64: Statystyka wysyłania pakietów TCP na port 80 (przebieg dobowy). Źródło:<br>opracowanie własne.....      | 101 |
| Rysunek 65: Statystyka odbierania pakietów TCP z portu 80 (przebieg tygodniowy). Źródło:<br>opracowanie własne..... | 101 |
| Rysunek 66: Statystyka odbierania pakietów TCP z portu 80 (przebieg dobowy). Źródło:<br>opracowanie własne.....     | 102 |
| Rysunek 67: Statystyka wysyłania pakietów UDP (przebieg tygodniowy). Źródło:<br>opracowanie własne.....             | 102 |
| Rysunek 68: Statystyka wysyłania pakietów UDP (przebieg dobowy). Źródło: opracowanie<br>własne.....                 | 103 |
| Rysunek 69: Statystyka odbierania pakietów UDP (przebieg tygodniowy). Źródło:<br>opracowanie własne.....            | 103 |
| Rysunek 70: Statystyka odbierania pakietów UDP (przebieg dobowy). Źródło: opracowanie<br>własne.....                | 104 |
| Rysunek 71: Statystyka wysyłania pakietów UDP na port 53 (przebieg tygodniowy). Źródło:<br>opracowanie własne.....  | 104 |
| Rysunek 72: Statystyka wysyłania pakietów UDP na port 53 (przebieg dobowy). Źródło:<br>opracowanie własne.....      | 105 |
| Rysunek 73: Statystyka odbierania pakietów UDP z portu 53 (przebieg tygodniowy). Źródło:<br>opracowanie własne..... | 105 |
| Rysunek 74: Statystyka odbierania pakietów UDP z portu 53 (przebieg dobowy). Źródło:<br>opracowanie własne.....     | 106 |
| Rysunek 75: Statystyka pakietów TCP (przebieg miesięczny).<br>Źródło: opracowanie własne.....                       | 107 |
| Rysunek 76: Statystyka wysłanych pakietów TCP (przebieg miesięczny). Źródło:<br>opracowanie własne.....             | 108 |
| Rysunek 77: Statystyka odebranych pakietów TCP (przebieg miesięczny). Źródło:<br>opracowanie własne.....            | 109 |
| Rysunek 78: Statystyka pakietów TCP wewnątrz sieci LAN (przebieg miesięczny). Źródło:<br>opracowanie własne.....    | 110 |
| Rysunek 79: Statystyka pakietów UDP (przebieg miesięczny).<br>Źródło: opracowanie własne.....                       | 111 |
| Rysunek 80: Statystyka wysłanych pakietów UDP (przebieg miesięczny). Źródło:<br>opracowanie własne.....             | 112 |
| Rysunek 81: Statystyka odebranych pakietów UDP (przebieg miesięczny). Źródło:<br>opracowanie własne.....            | 113 |
| Rysunek 82: Statystyka pakietów UDP wewnątrz sieci LAN (przebieg miesięczny). Źródło:<br>opracowanie własne.....    | 114 |
| Rysunek 83: Statystyka pakietów ICMP (przebieg miesięczny).<br>Źródło: opracowanie własne.....                      | 115 |

|                                                                                                                        |     |
|------------------------------------------------------------------------------------------------------------------------|-----|
| Rysunek 84: Statystyka wysłanych pakietów ICMP (przebieg miesięczny). Źródło:<br>opracowanie własne.....               | 116 |
| Rysunek 85: Statystyka odebranych pakietów ICMP (przebieg miesięczny). Źródło:<br>opracowanie własne.....              | 117 |
| Rysunek 86: Statystyka pakietów ICMP wewnątrz sieci LAN (przebieg miesięczny). Źródło:<br>opracowanie własne.....      | 118 |
| Rysunek 87: Liczba nowych połączeń (przebieg miesięczny).<br>Źródło: opracowanie własne.....                           | 119 |
| Rysunek 88: Statystyka wysłanych pakietów TCP port 80 (WWW) (przebieg miesięczny).<br>Źródło: opracowanie własne.....  | 120 |
| Rysunek 89: Statystyka odebranych pakietów TCP port 80 (WWW) (przebieg miesięczny).<br>Źródło: opracowanie własne..... | 121 |
| Rysunek 90: Statystyka wysłanych pakietów UDP port 53 (DNS) (przebieg miesięczny).<br>Źródło: opracowanie własne.....  | 122 |
| Rysunek 91: Statystyka odebranych pakietów UDP port 53 (DNS) (przebieg miesięczny).<br>Źródło: opracowanie własne..... | 123 |
| Rysunek 92: Statystyka wysyłania pakietów TCP (przebieg miesięczny). Źródło: opracowanie<br>własne. ....               | 124 |
| Rysunek 93: Statystyka odbierania pakietów TCP (przebieg miesięczny). Źródło:<br>opracowanie własne.....               | 125 |
| Rysunek 94: Statystyka wysyłania pakietów TCP na port 80 (przebieg miesięczny). Źródło:<br>opracowanie własne.....     | 126 |
| Rysunek 95: Statystyka odbierania pakietów TCP z portu 80 (przebieg miesięczny). Źródło:<br>opracowanie własne.....    | 127 |
| Rysunek 96: Statystyka wysyłania pakietów UDP (przebieg miesięczny). Źródło:<br>opracowanie własne.....                | 128 |
| Rysunek 97: Statystyka odbierania pakietów UDP (przebieg miesięczny). Źródło:<br>opracowanie własne.....               | 129 |
| Rysunek 98: Statystyka wysyłania pakietów UDP na port 53 (przebieg miesięczny). Źródło:<br>opracowanie własne.....     | 130 |
| Rysunek 99: Statystyka odbierania pakietów UDP z portu 53 (przebieg miesięczny). Źródło:<br>opracowanie własne.....    | 131 |
| Rysunek 100: Statystyka pakietów TCP. Źródło: opracowanie własne.....                                                  | 132 |
| Rysunek 101: Statystyka wysyłanych pakietów TCP. Źródło: opracowanie własne.....                                       | 132 |
| Rysunek 102: Statystyka odebranych pakietów TCP. Źródło: opracowanie własne.....                                       | 133 |
| Rysunek 103: Statystyka pakietów TCP wewnątrz sieci LAN.<br>Źródło: opracowanie własne.....                            | 133 |
| Rysunek 104: Statystyka pakietów UDP. Źródło: opracowanie własne.....                                                  | 134 |
| Rysunek 105: Statystyka wysłanych pakietów UDP. Źródło: opracowanie własne.....                                        | 134 |
| Rysunek 106: Statystyka odebranych pakietów UDP. Źródło: opracowanie własne.....                                       | 135 |
| Rysunek 107: Statystyka pakietów UDP wewnątrz sieci LAN.<br>Źródło: opracowanie własne.....                            | 135 |
| Rysunek 108: Statystyka pakietów ICMP. Źródło: opracowanie własne.....                                                 | 136 |
| Rysunek 109: Statystyka wysłanych pakietów ICMP. Źródło: opracowanie własne.....                                       | 136 |
| Rysunek 110: Statystyka odebranych pakietów ICMP. Źródło: opracowanie własne.....                                      | 137 |
| Rysunek 111: Statystyka pakietów ICMP wewnątrz sieci LAN.<br>Źródło: opracowanie własne.....                           | 137 |
| Rysunek 112: Statystyka nowych połączeń. Źródło: opracowanie własne.....                                               | 138 |
| Rysunek 113: Statystyka wysłanych pakietów TCP na port 80 (WWW). Źródło: opracowanie<br>własne. ....                   | 138 |

|                                                                                                          |     |
|----------------------------------------------------------------------------------------------------------|-----|
| Rysunek 114: Statystyka odebranych pakietów TCP na port 80 (WWW). Źródło: opracowanie własne.            | 139 |
| Rysunek 115: Statystyka wysłanych pakietów UDP na port 53 (DNS). Źródło: opracowanie własne.             | 139 |
| Rysunek 116: Statystyka odebranych pakietów UDP na port 53 (DNS). Źródło: opracowanie własne.            | 140 |
| Rysunek 117: Statystyka ruchu TCP (dane wysłane). Źródło: opracowanie własne.                            | 140 |
| Rysunek 118: Statystyka ruchu TCP (dane odebrane). Źródło: opracowanie własne.                           | 141 |
| Rysunek 119: Statystyka ruchu WWW (dane wysłane). Źródło: opracowanie własne.                            | 141 |
| Rysunek 120: Statystyka ruchu WWW (dane odebrane). Źródło: opracowanie własne.                           | 142 |
| Rysunek 121: Statystyka ruchu UDP (dane wysłane). Źródło: opracowanie własne.                            | 142 |
| Rysunek 122: Statystyka ruchu UDP (dane odebrane). Źródło: opracowanie własne.                           | 143 |
| Rysunek 123: Statystyka ruchu UDP port 53 (dane wysłane).<br>Źródło: opracowanie własne.                 | 143 |
| Rysunek 124: Statystyka ruchu UDP port 53 (dane odebrane).<br>Źródło: opracowanie własne.                | 144 |
| Rysunek 125: Statystyka alertów - ruch TCP. Źródło: opracowanie własne.                                  | 144 |
| Rysunek 126: Statystyka alertów – wysłane pakiety TCP. Źródło: opracowanie własne.                       | 145 |
| Rysunek 127: Statystyka alertów – odebrane pakiety TCP. Źródło: opracowanie własne.                      | 145 |
| Rysunek 128: Statystyka alertów – pakiety TCP wewnątrz sieci LAN. Źródło: opracowanie własne.            | 146 |
| Rysunek 129: Statystyka alertów – pakiety UDP. Źródło: opracowanie własne.                               | 146 |
| Rysunek 130: Statystyka alertów – wysłane pakiety UDP. Źródło: opracowanie własne.                       | 147 |
| Rysunek 131: Statystyka alertów – odebrane pakiety UDP. Źródło: opracowanie własne.                      | 147 |
| Rysunek 132: Statystyka alertów – pakiety UDP wewnątrz sieci LAN. Źródło: opracowanie własne.            | 148 |
| Rysunek 133: Statystyka alertów – pakiety ICMP. Źródło: opracowanie własne.                              | 148 |
| Rysunek 134: Statystyka alertów – wysłane pakiety ICMP. Źródło: opracowanie własne.                      | 149 |
| Rysunek 135: Statystyka alertów – odebrane pakiety ICMP. Źródło: opracowanie własne.                     | 149 |
| Rysunek 136: Statystyka alertów – pakiety ICMP wewnątrz sieci LAN. Źródło: opracowanie własne.           | 150 |
| Rysunek 137: Statystyka alertów – nowe połączenia (TCP z flagami SYN i ACK). Źródło: opracowanie własne. | 150 |
| Rysunek 138: Statystyka alertów – wysłane pakiety TCP (port 80). Źródło: opracowanie własne.             | 151 |
| Rysunek 139: Statystyka alertów – odebrane pakiety TCP (port 80). Źródło: opracowanie własne.            | 151 |
| Rysunek 140: Statystyka alertów – wysłane pakiety UDP (port 53). Źródło: opracowanie własne.             | 152 |
| Rysunek 141: Statystyka alertów – odebrane pakiety UDP (port 53). Źródło: opracowanie własne.            | 152 |
| Rysunek 142: Statystyka alertów – ruch TCP (dane wysłane).<br>Źródło: opracowanie własne.                | 153 |
| Rysunek 143: Statystyka alertów – ruch TCP (dane odebrane).<br>Źródło: opracowanie własne.               | 153 |
| Rysunek 144: Statystyka alertów – ruch WWW (dane wysłane).<br>Źródło: opracowanie własne.                | 154 |
| Rysunek 145: Statystyka alertów – ruch WWW (dane odebrane). Źródło: opracowanie własne.                  | 154 |



|                                                                                                                |     |
|----------------------------------------------------------------------------------------------------------------|-----|
| Rysunek 146: Statystyka alertów – ruch UDP (dane wysłane).                                                     |     |
| Źródło: opracowanie własne.....                                                                                | 155 |
| Rysunek 147: Statystyka alertów – ruch UDP (dane odebrane).                                                    |     |
| Źródło: opracowanie własne.....                                                                                | 155 |
| Rysunek 148: Statystyka alertów – ruch UDP port 53 (dane wysłane). Źródło: opracowanie własne. ....            | 156 |
| Rysunek 149: Statystyka alertów – ruch UDP port 53 (dane odebrane). Źródło: opracowanie własne. ....           | 156 |
| Rysunek 150: Statystyka alertów - ruch TCP. Źródło: opracowanie własne. ....                                   | 157 |
| Rysunek 151: Statystyka alertów – wysłane pakiety TCP. Źródło: opracowanie własne. ....                        | 158 |
| Rysunek 152: Statystyka alertów – odebrane pakiety TCP. Źródło: opracowanie własne. ....                       | 158 |
| Rysunek 153: Statystyka alertów – pakiety TCP wewnątrz sieci LAN. Źródło: opracowanie własne. ....             | 159 |
| Rysunek 154: Statystyka alertów – pakiety UDP. Źródło: opracowanie własne. ....                                | 159 |
| Rysunek 155: Statystyka alertów – wysłane pakiety UDP. Źródło: opracowanie własne. ....                        | 160 |
| Rysunek 156: Statystyka alertów – odebrane pakiety UDP. Źródło: opracowanie własne. ....                       | 160 |
| Rysunek 157: Statystyka alertów – pakiety UDP wewnątrz sieci LAN. Źródło: opracowanie własne. ....             | 161 |
| Rysunek 158: Statystyka alertów – pakiety ICMP. Źródło: opracowanie własne. ....                               | 161 |
| Rysunek 159: Statystyka alertów – wysłane pakiety ICMP. Źródło: opracowanie własne. ....                       | 162 |
| Rysunek 160: Statystyka alertów – odebrane pakiety ICMP. Źródło: opracowanie własne. ....                      | 162 |
| Rysunek 161: Statystyka alertów – pakiety ICMP wewnątrz sieci LAN. Źródło: opracowanie własne. ....            | 163 |
| Rysunek 162: Statystyka alertów – nowe połączenia (TCP z flagami SYN i ACK). Źródło: opracowanie własne.....   | 163 |
| Rysunek 163: Statystyka alertów – wysłane pakiety TCP (port 80). Źródło: opracowanie własne. ....              | 164 |
| Rysunek 164: Statystyka alertów – odebrane pakiety TCP (port 80). Źródło: opracowanie własne. ....             | 164 |
| Rysunek 165: Statystyka alertów – wysłane pakiety UDP (port 53). Źródło: opracowanie własne. ....              | 165 |
| Rysunek 166: Rysunek 167: Statystyka alertów – odebrane pakiety UDP (port 53). Źródło: opracowanie własne..... | 165 |
| Rysunek 168: Statystyka alertów – ruch TCP (dane wysłane).                                                     |     |
| Źródło: opracowanie własne.....                                                                                | 166 |
| Rysunek 169: Statystyka alertów – ruch TCP (dane odebrane).                                                    |     |
| Źródło: opracowanie własne.....                                                                                | 166 |
| Rysunek 170: Statystyka alertów – ruch WWW (dane wysłane).                                                     |     |
| Źródło: opracowanie własne.....                                                                                | 167 |
| Rysunek 171: Statystyka alertów – ruch WWW (dane odebrane).                                                    |     |
| Źródło: opracowanie własne.....                                                                                | 167 |
| Rysunek 172: Statystyka alertów – ruch UDP (dane wysłane).                                                     |     |
| Źródło: opracowanie własne.....                                                                                | 168 |
| Rysunek 173: Statystyka alertów – ruch UDP (dane odebrane).                                                    |     |
| Źródło: opracowanie własne.....                                                                                | 168 |
| Rysunek 174: Statystyka alertów – ruch UDP port 53 (dane wysłane). Źródło: opracowanie własne. ....            | 169 |
| Rysunek 175: Statystyka alertów – ruch UDP port 53 (dane odebrane). Źródło: opracowanie własne. ....           | 169 |
| Rysunek 176: Statystyka alertów - ruch TCP. Źródło: opracowanie własne. ....                                   | 170 |

|                                                                                                              |     |
|--------------------------------------------------------------------------------------------------------------|-----|
| Rysunek 177: Statystyka alertów – wysłane pakiety TCP. Źródło: opracowanie własne.....                       | 170 |
| Rysunek 178: Statystyka alertów – odebrane pakiety TCP. Źródło: opracowanie własne. ...                      | 171 |
| Rysunek 179: Statystyka alertów – pakiety TCP wewnątrz sieci LAN. Źródło: opracowanie własne. ....           | 171 |
| Rysunek 180: Statystyka alertów – pakiety UDP. Źródło: opracowanie własne. ....                              | 172 |
| Rysunek 181: Statystyka alertów – wysłane pakiety UDP. Źródło: opracowanie własne. ....                      | 172 |
| Rysunek 182: Statystyka alertów – odebrane pakiety UDP. Źródło: opracowanie własne. ..                       | 173 |
| Rysunek 183: Statystyka alertów – pakiety UDP wewnątrz sieci LAN. Źródło: opracowanie własne. ....           | 173 |
| Rysunek 184: Statystyka alertów – pakiety ICMP. Źródło: opracowanie własne. ....                             | 174 |
| Rysunek 185: Statystyka alertów – wysłane pakiety ICMP. Źródło: opracowanie własne. ..                       | 174 |
| Rysunek 186: Statystyka alertów – odebrane pakiety ICMP. Źródło: opracowanie własne. ..                      | 175 |
| Rysunek 187: Statystyka alertów – pakiety ICMP wewnątrz sieci LAN. Źródło: opracowanie własne. ....          | 175 |
| Rysunek 188: Statystyka alertów – nowe połączenia (TCP z flagami SYN i ACK). Źródło: opracowanie własne..... | 176 |
| Rysunek 189: Statystyka alertów – wysłane pakiety TCP (port 80). Źródło: opracowanie własne. ....            | 176 |
| Rysunek 190: Statystyka alertów – odebrane pakiety TCP (port 80). Źródło: opracowanie własne. ....           | 177 |
| Rysunek 191: Statystyka alertów – wysłane pakiety UDP (port 53). Źródło: opracowanie własne. ....            | 177 |
| Rysunek 192: Statystyka alertów – odebrane pakiety UDP (port 53). Źródło: opracowanie własne. ....           | 178 |
| Rysunek 193: Statystyka alertów – ruch TCP (dane wysłane).<br>Źródło: opracowanie własne.....                | 178 |
| Rysunek 194: Statystyka alertów – ruch TCP (dane odebrane).<br>Źródło: opracowanie własne.....               | 179 |
| Rysunek 195: Statystyka alertów – ruch WWW (dane wysłane).<br>Źródło: opracowanie własne.....                | 179 |
| Rysunek 196: Statystyka alertów – ruch WWW (dane odebrane).<br>Źródło: opracowanie własne.....               | 180 |
| Rysunek 197: Statystyka alertów – ruch UDP (dane wysłane).<br>Źródło: opracowanie własne.....                | 180 |
| Rysunek 198: Statystyka alertów – ruch UDP (dane odebrane).<br>Źródło: opracowanie własne.....               | 181 |
| Rysunek 199: Statystyka alertów – ruch UDP port 53 (dane wysłane). Źródło: opracowanie własne. ....          | 181 |
| Rysunek 200: Statystyka alertów – ruch UDP port 53 (dane odebrane). Źródło: opracowanie własne. ....         | 182 |
| Rysunek 201: Zależność liczby alertów od mnożnika sigmy: ruch TCP. Źródło: opracowanie własne. ....          | 183 |
| Rysunek 202: Zależność liczby alertów od mnożnika sigmy: ruch UDP. Źródło: opracowanie własne. ....          | 183 |
| Rysunek 203: Zależność liczby alertów od mnożnika sigmy: ruch ICMP. Źródło: opracowanie własne. ....         | 184 |
| Rysunek 204: Zależność liczby alertów od mnożnika sigmy: liczba SYNACK. Źródło: opracowanie własne.....      | 184 |
| Rysunek 205: Zależność liczby alertów od mnożnika sigmy: ruch WWW i DNS. Źródło: opracowanie własne.....     | 185 |

|                                                                                                                       |     |
|-----------------------------------------------------------------------------------------------------------------------|-----|
| Rysunek 206: Zależność liczby alertów od mnożnika sigmy: szybkość ruchu. Źródło:<br>opracowanie własne.....           | 185 |
| Rysunek 207: Zależność liczby alertów od mnożnika sigmy: sumaryczna ilość alertów.<br>Źródło: opracowanie własne..... | 186 |

## Spis tabel:

|                                                                                                                                     |    |
|-------------------------------------------------------------------------------------------------------------------------------------|----|
| Tabela 1: historia systemów IDS. Źródło: Endorf, 2004, rozdział: The History of Intrusion<br>Detection and Prevention.....          | 8  |
| Tabela 2: przykład profilu dla ruchu TCP dla poniedziałku.....                                                                      | 62 |
| Tabela 3: Statystyki ruchu sieciowego stworzone w oparciu o dane zebrane przez preprocesor<br>w sieci złożonej z 30 komputerów..... | 66 |
| Tabela 4: Zależność ilości alertów od mnożnika sigmy.....                                                                           | 77 |

## Spis wydruków:

|                                                                |    |
|----------------------------------------------------------------|----|
| wydruk 1. przechwycony pakiet (root@serwer:/# snort -v).....   | 40 |
| wydruk 2. przechwycony pakiet (root@serwer:/# snort -dv).....  | 41 |
| wydruk 3. przechwycony pakiet (root@serwer:/# snort -dev)..... | 42 |
| wydruk 4. Struktura pliku nagłówkowego preprocesora.....       | 56 |
| wydruk 5. Struktura pliku źródłowego preprocesora.....         | 56 |