

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych
Instytut Informatyki

Rok akademicki 2013/2014

Praca dyplomowa inżynierska

Krzysztof Opasiak

Rozproszony monitoring systemów komputerowych

Opiekun pracy:
dr inż. Piotr Gawkowski

Ocena

.....

Podpis Przewodniczącego
Komisji Egzaminu Dyplomowego



Kierunek: Informatyka

Specjalność: Inżynieria Systemów Informatycznych

Data urodzenia: 1990.12.28

Data rozpoczęcia studiów: 2010.10.01

Życiorys

Urodziłem się 28 grudnia 1990 w Koninie. Uczęszczałem do Szkoły Podstawowej numer 8 im. Powstańców Wielkopolskich w Koninie. Następnie uczęszczałem do Gimnazjum Towarzystwa Salezjańskiego w Koninie.

W latach 2006-2010 uczęszczałem do Technikum w Zespole Szkół im. Mikołaja Kopernika w Koninie. W trakcie nauki w tej szkole dwukrotnie przyznano mi stypendium Prezesa Rady Ministrów za bardzo dobre wyniki w nauce oraz wzorowe zachowanie. W roku 2010 ukończyłem z wyróżnieniem szkołę średnią, a następnie zdałem maturę oraz egzamin zawodowy uzyskując tytuł Technik Teleinformatyk.

W październiku 2010 roku rozpocząłem studia stacjonarne pierwszego stopnia na Wydziale Elektroniki i Technik Informatycznych na kierunku Informatyka.

.....
podpis studenta

Egzamin dyplomowy

Złożył egzamin dyplomowy w dn.20__r

Z wynikiem

Ogólny wynik studiów

Dodatkowe wnioski i uwagi Komisji

.....

Streszczenie

Praca ta prezentuje ...

Słowa kluczowe: *słowa kluczowe.*

Abstract

Title: *Thesis title.*

This thesis describes ...

Key words: *key words.*

Spis treści

1. Wprowadzenie	1
2. Monitorowanie klienta mobilnego jako monitorowanie rozproszone	2
2.1. Monitorowanie rozproszone klientów statycznych	2
2.2. Monitorowanie rozproszone klientów mobilnych	3
2.3. Wymagania systemu monitorowania klientów mobilnych	4
3. Architektura proponowanego systemu	7
3.1. Podział na moduły	7
3.2. Moduł podstawowy	8
3.3. Moduł odbioru danych	8
3.4. Moduł mobilny	9
4. Dostępne systemy monitorujące	11
4.1. Przegląd systemów dostępnych na rynku	11
4.1.1. System monitorowania Nagios	11
4.1.2. System monitorowania Icinga	11
4.2. Porównanie i wybór systemu monitorującego	11
5. Monitorowanie rozproszone z użyciem NSCA	12
5.1. Opis dodatku NSCA	12
5.1.1. Moduł wysyłający	12
5.1.2. Moduł odbierający	13
5.2. Bezpieczeństwo	15
5.3. Problemy z monitorowaniem klienta mobilnego	16
6. Architektura modułu odbioru danych	18
6.1. Analiza	18
6.2. Opis architektury	19
6.3. Szkielet programu	21
6.4. Moduł kryptograficzny	22
6.5. Moduł autoryzacji klienta	22
6.6. Moduł komunikacji z wykorzystaniem TCP	22
6.7. Moduł logowania	22
7. Protokół komunikacyjny	23
7.1. Podział na warstwy	23
7.2. Warstwa formowania wiadomości	23
7.3. Warstwa kryptograficzna	23
7.4. Warstwa integralności danych	23
7.5. Warstwa transportu logów	23
8. Testowanie i użytkowanie wykonanego systemu	24
8.1. Testowanie	24
8.2. Użytkowanie systemu	24
9. Podsumowanie	25
Bibliografia	26

1. Wprowadzenie

2. Monitorowanie klienta mobilnego jako monitorowanie rozproszone

2.1. Monitorowanie rozproszone klientów statycznych

Firmy działające obecnie na rynku posiadają bardzo rozbudowaną infrastrukturę informatyczną. Od bardzo wielu lat działy odpowiedzialne za utrzymanie infrastruktury informatycznej prowadzą ciągły monitoring zarówno urządzeń sieciowych jak i serwerów oraz stacji roboczych użytkowników. Bardzo wiele firm posiada również specjalistyczne urządzenia, które również muszą być podłączone do sieci i monitorowane w celu zapewnienia ciągłości procesów biznesowych danej firmy. Powyższe urządzenia rozumiane są jako klienci statyczne. Urządzenia tego typu zazwyczaj pracują nieprzerwanie i posiadają dobrze zdefiniowaną hierarchię. Wzajemne relacje pomiędzy tymi urządzeniami wynikają w dużej mierze z struktury sieci lecz mogą również wynikać z roli jaką pełnią one w danej organizacji. Dzięki monitorowaniu wszystkich urządzeń w danej sieci systemy monitorujące są w stanie wspierać administratora wskazując z bardzo dużym prawdopodobieństwem miejsce wystąpienia awarii.

Sieć w dużej firmie rzadko stanowi jedną całość. Zazwyczaj są to segmenty sieci oddzielone zaporami lub w ogóle oddzielnie sieci LAN lub VLAN. Taka separacja urządzeń pozwala na zwiększenie poziomu bezpieczeństwa, lecz jednocześnie utrudnia monitorowanie całej infrastruktury. Aby umożliwić monitorowanie całej sieci firmowej wykorzystywane jest monitorowanie rozproszone. Można wyróżnić dwie podstawowe konfiguracje monitorowania rozproszonego:

- Monitorowanie pasywne: Istnieje jedna, centralna instancja jądra monitorującego, do którego przesyłane są wyniki sprawdzeń poszczególnych usług. Każde urządzenie samo monitoruje swoje usługi i zgłasza rezultaty.
- Wieloinstancyjny system monitorujący: Istnieje wiele instancji jądra monitorującego. Typowo, każda wydzielona część sieci posiada swoją instancję. Każda instancja może posiadać zarówno usługi monitorowane aktywnie jak i pasywnie. Wyniki sprawdzeń przesyłane są następnie do jednej wybranej instancji, która gromadzi wszystkie dane.

Użycie monitorowania pasywnego dla wszystkich usług jest bardzo nie wygodnie i jednocześnie utrudnia konfiguracje, a także pozbawia administratora możliwości używania niektórych mechanizmów dostępnych wyłącznie dla urządzeń monitorowanych aktywnie. Ponadto wyniki sprawdzeń pasywnych nie są akumulowane, lecz wysyłane od razu po ich uzyskaniu. Oznacza to, że jeśli pojawi się chwilowy brak połączenia z serwerem, to wpisy dziennika zostaną zgubione. W przypadku, gdy jedynym celem systemu jest monitorowanie dostępności danej usługi zewnętrznej serwera, a nie jego parametrów wewnętrznych nie ma to większego

znaczenia. Kwestia ta staje się jednak istotna, gdy jednym z zadań systemu, jest gromadzenia i analiza danych historycznych. Wieleinstancyjny system monitorujący wymaga zdecydowanie więcej zasobów jednak pozwala na osiągnięcie znacznie wygodniejszego i bardziej niezawodnego systemu. Ponadto dzięki takiej konfiguracji nie ma potrzeby ingerencji w monitorowane serwery co redukuje ich obciążenie, a także zwiększa bezpieczeństwo. Warto również wspomnieć, iż istnieją systemy, które dają możliwość integracji wielu instancji jądra monitorującego. Dzięki temu administrator danej sieci ma możliwość monitorowania i konfigurowania wielu instancji. Niestety w wielu systemach rozwiązanie to posiada zamknięte źródła i jego wykorzystanie wymaga zakupu licencji. Rozwiązania oparte na istnieniu jednej centralnej instancji jądra systemu monitorującego, do której przesyłane są w ramach możliwości odczyty wykonane przez inne instancje, są zazwyczaj darmowe lecz wymagają dodatkowej instancji, zajmującej się agregacją danych.

2.2. Monitorowanie rozproszone klientów mobilnych

Rosnąca w ostatnich latach popularność technologii mobilnych przyczyniła się do pojawienia się w firmach bardzo dużej liczby urządzeń mobilnych, które wymagają zarówno zarządzania jak i monitorowania. Urządzenia mobilne są używane bardzo często przez przedstawicieli handlowych, a także przez menadżerów w celu umożliwienia wykonywania pracy poza obszarem firmy. Ponadto coraz więcej firm świadczących zaawansowane technicznie usługi wyposaża swoich pracowników w bardzo drogi sprzęt, który wymaga ciągłego monitorowania. Duże korporacje coraz częściej decydują się również na wyposażenie swoich pracowników w smartfony lub tablety, które mają ułatwić współpracę z firmą w trakcie podróży służbowych czy spotkań z klientami.

Klient mobilny posiada szereg cech, które znacząco odróżniają go od klientów statycznych. Przedewszystkim należy zauważyć, że urządzenia, o których mowa bardzo często pracują poza obszarem firmy. Wynika z tego iż nie zawsze możliwe jest utrzymywanie takich urządzeń w wirtualnej sieci prywatnej, gdyż urządzenie może znaleźć się w obszarze, gdzie nie ma dostępu do internetu. Ponadto nie zawsze konieczne jest, aby urządzenia mobilne pracowały podłączone do sieci firmowej, gdyż dla użytkownika często wymagany jest jedynie dostęp do internetu i inne funkcje tego urządzenia. Warto więc zdać sobie sprawę, że urządzenia te są często narażone na dostęp do sieci, o bardzo niskim poziomie zaufania i wielu zagrożeniach. Oznacza to w szczególności, iż urządzenie mobilne zazwyczaj posiada zmienny adres IP, który rzadko jest adresem globalnym. Również struktura sieci, z której korzystają klienty mobilne jest dynamiczna i znajduje się poza obszarem monitorowania administratorów danego przedsiębiorstwa. Znacząca większość klientów mobilnych dzięki kontaktom z sieciami poza firmową posiada, w przeciwieństwie do klientów statycznych, możliwość synchronizacji swojego czasu czy to z serwerami czasu światowego, czy też z sieci GSM.

Należy również zwrócić uwagę na duże rozproszenie klientów mobilnych. W przeciwieństwie do klientów statycznych, którzy zazwyczaj pracują w pewnych grupach lub fragmentach sieci, klienty mobilne są zazwyczaj rozpatrywane pojedynczo. Większość klientów mobilnych operuje w pełni samodzielnie, zatem grupa licząca grupę klientów wynosi zazwyczaj 1. Powoduje to, że w przeciwieństwie do klientów statycznych gdzie grup koniecznych do wydzielenia było zazwyczaj kilka

lub kilkanaście, w przypadku klientów mobilnych takich grup może być kilkaset lub nawet kilka tysięcy. Warto również dostrzec różnice w zasilaniu. Klienci mobilne zazwyczaj posiadają własne zasilanie, przez co każda operacja wykonywana na nim nie tylko spowalnia jego działanie, lecz również zmniejsza jego czas pracy pomiędzy ładowaniami. Przenośność klienta mobilnego zmienia również jego stopień bezpieczeństwa. Urządzenia mobilne stosunkowo często są gubione lub kradzione, co nie było możliwe w przypadku klientów statycznych. W związku z możliwością utraty urządzenia, nie powinno się na nim przechowywać tajnych danych, dzięki którym możnaby skompromitować cały system z którego korzysta klient.

Klient mobilny znacznie różni się swoją charakterystyką od klienta statycznego. Różni się również rodzaj monitorowanych usług. W przypadku klientów statycznych znaczna część wysiłków jest skierowana na pomiar usług świadczonych przez dany system dla innych systemów. Natomiast w przypadku klientów mobilnych znacznie większy nacisk jest położony na monitorowanie parametrów wewnętrznych danego klienta.

2.3. Wymagania systemu monitorowania klientów mobilnych

Klient mobilny posiada zdecydowanie odmienną charakterystykę niż klient statyczny. Dokonano zatem głębokiej analizy, jakie wymagania należały spełnić, aby dostarczyć system, który sprosta oczekiwaniom administratorów.

Odbiorcą systemu mają być duże firmy i korporacje, które posiadają bardzo rozbudowaną sieć wewnątrz firmy, a ponadto udostępniają swoim pracownikom urządzenia mobilne różnej klasy. Wśród tych urządzeń znajdują się przede wszystkim telefony oraz tablety z systemem operacyjnym Android oraz Windows Phone, a także liczne laptopy wyposażone w system Windows lub Linux. Konieczne jest zatem, aby system pozwalał na monitorowanie każdej z wspomnianych platform. Duże firmy oraz korporacje, zazwyczaj posiadają już oprogramowanie służące do monitorowania swojej infrastruktury sieciowej. Aby umożliwić administratorom łatwe zarządzanie oraz monitorowanie zarówno klientami mobilnymi jak i statycznymi, należy zapewnić integrację systemów monitorowania obu kategorii klientów. Dane odczytywane na urządzeniu mobilnym mogą zawierać zarówno dane prywatne pracownika, jak i tajemnice handlowe firmy. Oba te rodzaje danych należą do kategorii poufnych i powinny być należycie chronione. Ponieważ urządzenie mobilne będzie pracowało często poza siecią firmową, podczas tworzenia systemu należy zwrócić szczególną uwagę na kwestię bezpieczeństwa przesyłanych danych. Ponieważ system, musi przysyłać dane poprzez sieć publiczną, konieczne jest również zapewnienie odporności systemu na ataki zewnętrzne oraz na próby przekazywania sfałszowanych danych do systemu. Wszystkie wymagania stawiane przed omawianym systemem zostały zebrane w 2.1.

Tablica 2.1: Wymagania systemu monitorowania klienta mobilnego

Kod	Nazwa	Opis
W1	Spójność danych	System musi zapewnić, że wpisy dziennika nie zostaną zgubione. System musi zapewniać spójność danych pomiędzy serwerem, a klientem mobilnym.
W2	Integralności	System musi zapewnić, że wpisy dziennika dostarczone do serwera nie zostały w żaden sposób zmodyfikowane lub dodane.
W3	Autentyczność	System musi zapewnić, że odebrane dane pochodzą od uprawnionego klienta.
W4	Poufność	System musi zapewniać poufność danych przesyłanych od klienta poprzez szyfrowanie.
W5	Dodawanie algorytmów	System musi być niezależny od algorytmu kryptograficznego stosowanego podczas przesyłania danych. Ponadto system musi umożliwiać dodawanie w prosty sposób nowych algorytmów kryptograficznych.
W6	Uwierzytelnienie klienta	System musi zapewnić możliwość uwierzytelnienia klienta.
W7	Wymienne algorytmy uwierzytelnienia klienta	System musi być niezależny od algorytmu uwierzytelnienia klienta. Ponadto system musi umożliwiać dodanie w prosty sposób nowych algorytmów uwierzytelnienia klienta.
W8	Uwierzytelnienie serwera	System musi zapewniać, iż wpisy dziennika zostaną przesłane tylko do wyznaczonego, uprawnionego serwera.
W9	Odporność na zgubienie urządzenia	System musi być odporny na zgubienie urządzenia. Oznacza to iż zgubienie urządzenia nie może powodować kompromitacji całego systemu.
W10	Dostarczanie w wiele miejsc	System musi umożliwiać przekazywanie danych do wielu podsystemów monitorujących, bez konieczności ich retransmisji z klienta mobilnego.
W11	Reguły definiowane dla każdego klienta	System musi umożliwiać definiowanie reguł dotyczących miejsc przeznaczenia dla każdego klienta indywidualnie.
W12	Oszczędność pasma	System powinien minimalizować ilość przesyłanych danych. Ponadto powinien skrócić do minimum czas oczekiwania na potwierdzenie przetworzenia przesłanych danych.
Kontynuacja na następnej stronie		

Tablica 2.1 – Kontynuacja z poprzedniej strony

Kod	Nazwa	Opis
W13	Integracja z istniejącymi systemami	System monitoringu klienta mobilnego musi mieć możliwość integracji i współpracy z istniejącymi systemami monitorowania klienta statycznego.
W14	Analiza danych bieżących	System musi umożliwiać prezentację oraz analizę danych bieżących, a także posiadać możliwość reagowania na wystąpienie zdefiniowanych przez użytkownika zdarzeń.
W15	Analiza danych historycznych	System musi umożliwiać analizę zadanych danych historycznych włączając w to ich graficzną reprezentację.
W16	Kontrola danych wejściowych	System musi prowadzić kontrolę danych wejściowych od klientów. Konieczne jest aby system umożliwiał definiowanie jakie dane mogą być dostarczane przez jakich klientów.
W17	Łatwość dodawania nowych sprawdzeń	System musi umożliwiać dodawanie w łatwy sposób możliwości monitorowania nowych usług i parametrów.
W18	Klient dla platformy Android	System musi udostępniać klienta pozwalającego na monitorowanie urządzeń opartych na platformie Android
W19	Klient dla platformy Windows Phone	System musi udostępniać klienta pozwalającego na monitorowanie urządzeń opartych na platformie Windows Phone
W20	Klient dla platformy Windows 8	System musi udostępniać klienta pozwalającego na monitorowanie urządzeń opartych na platformie Windows 8
W21	Klient dla platformy Linux	System musi udostępniać klienta pozwalającego na monitorowanie urządzeń opartych na platformie Linux

3. Architektura proponowanego systemu

3.1. Podział na moduły

Klient mobilny, zdefiniowany w 2 jest urządzeniem, co do którego, nie można zakładać, że powinno mieć nieprzerwany dostęp do sieci internet. Ponadto należy zauważyć zmienność zarówno geograficznego miejsca użytkowania jak i zmienność, wykorzystywanej infrastruktury sieciowej. Dodatkowo, należy odnieść się do wymagań, w których zawarta jest konieczność minimalizowania zużycia energii przez kłietna mobilnego. Ciągłe utrzymywanie połączenia z serwerem, powodowałoby znaczne zużycie energii. Współpraca klienta mobilnego z infrastrukturą publiczną nie pozwala również, na założenie, iż klient mobilny posiada globalny adres IP¹. Powoduje to brak możliwości odpytywania klienta mobilnego o jego stan.

Brak możliwości odpytywania klienta mobilnego o jego stan, wymusza istnienie elementu systemu, znajdującego się, na urządzeniu mobilnym. Element ten musi zatem minotorować urządzenie, na którym się znajduje, a następnie, gdy pojawi się taka możliwość przekazywać dane do podsystemu centralnego. Przekazanie danych powinno odbyć się w sposób zapewniający poufność i integralność przesyłanych danych. Konieczne jest również uniemożliwienie fałszowania danych przez inne urządzenia oraz ich przechwycenia. Tak postawione wymagania w kwestii bezpieczeństwa powodują konieczność istnienia elementu odpowiedzialnego za odebranie w sposób bezpieczny danych z sieci zewnętrznej, a następnie przekazanie tych danych do podsystemu odpowiedzialnego za ich właściwe przetwarzanie.

System powinien również umożliwiać monitorowanie infrastruktury statycznej. Niezbędne, jest również udostępnienie danych bierzących administratorowi. Kolejnym z wymagań jest możliwość prezentacji i analizy danych historycznych dotyczących wszystkich rodzajów klientów. Konieczne jest zatem istnienie elementu systemu, który jest odpowiedzialny, za monitorowanie klientów statycznych, a także umożliwi przetwarzanie danych dostarczonych przez inne moduły od klientów mobilnych.

System, który spełni wymagania postawione w 2 powinien składać się conajmniej z poniższych modułów:

- Moduł podstawowy, odpowiedzialny za bezpośredni monitoring klientów statycznych oraz analizę danych od klientów mobilnych.
- Moduł odbioru danych, odpowiedzialny za przekazywanie wpisów dziennika od klientów mobilnych do modułu podstawowego.
- Moduł mobilny, odpowiedzialny za monitorowanie klientów mobilnych i przekazywanie danych do modułu odbioru danych.

¹ Globalny adres IP - adres protokołu internetowego działającego w warstwie sieciowej, pozwalający na unikalną identyfikację urządzenia w ramach całej sieci Internet.

Dobre praktyki programistyczne, a także dbałość o możliwości rozwoju systemu, nakazują umożliwienie komunikacji, pomiędzy modułami poprzez dobrze zdefiniowane interfejsy. Zapewni to wymiennność poszczególnych modułów systemu i pozwoli na lepsze dostosowanie całego systemu do oczekiwań konkretnego klienta. Należy zatem zwrócić szczególną uwagę, na zapewnienie bezpiecznego i elastycznego protokołu komunikacji pomiędzy klientem mobilnym, a modułem odbioru danych. Ważna jest również komunikacja, pomiędzy modułem odbioru danych, a modułem podstawowym. Należy ją zorganizować w sposób, który umożliwi współpracę z różnymi, modułami podstawowymi.

3.2. Moduł podstawowy

Moduł podstawowy stanowi rdzeń całego systemu monitorowania. Moduł ten został zbudowany wykorzystując system monitorowania Icinga. System monitorujący Icinga został szeroko opisany w 4.1.2. Cieszy się on uznaniem środowiska administratorów, a jego możliwości konfiguracji umożliwiają budowę rozległego systemu monitorowania rozproszonego zarówno dla bardzo dużej sieci, jak i dla dużej liczby klientów mobilnych.

Tutaj będzie opis tego modułu

3.3. Moduł odbioru danych

Moduł ten odpowiedzialny jest za odbieranie danych od klientów mobilnych i przekazywanie ich do modułu podstawowego. Stawiane wymagania, nakładają na ten moduł możliwość obsługi wielu klientów, używających wielu różnych platform. Ze względu na wymianę danych z klientem mobilnym poprzez sieć Internet, moduł ten musi umożliwiać bezpieczną wymianę danych. Szeroko rozumiany model bezpieczeństwa realizowany przez ten moduł składa się z następujących elementów:

- poufność: przekazywane dane mogą zawierać tajemnice handlowe firmy oraz dane prywatne pracownika, dlatego ich transmisja powinna być szyfrowana. Wykorzystanie kryptografii asymetrycznej wiąże się z przechowywaniem dużej liczby kluczy klientów, a także wymaga od klienta mobilnego większej ilości obliczeń. W związku z powyższym powinny zostać wykorzystane, algorytmy symetryczne o kluczach generowanych każdorazowo dla nawiązywanego połączenia, aby nie było konieczności przechowywania ich na urządzeniu mobilnym.
- integralność: przekazywane dane powinny być przyjmowane tylko jeśli jest pewność, iż nie zostały one zmodyfikowane przez stronę trzecią. Aby to uzyskać, stosuje się funkcję skrótu, której wynik jest dołączany do wiadomości. Konieczne jest zatem, żeby moduł odbioru danych potrafił obliczać i weryfikować funkcje skrótu klasy co najmniej SHA-2.
- uwierzytelnienie klienta: konieczność komunikacji poprzez sieć globalną, stwarza ryzyko odbioru danych od nieuprawnionych urządzeń. Konieczne jest zatem, aby moduł ten umożliwiał przeprowadzenie weryfikacji tożsamości klienta. Zgodnie z wymaganiami, proces weryfikacji, powinien być niezależny od tego modułu i musi być możliwe definiowanie nowych, dowolnych metod uwierzytelnienia klienta.

- uwierzytelnienie serwera: klient mobilny korzysta z różnych infrastruktur sieciowych. Część z nich może być narażona na ataki z zewnątrz. Konieczne jest zatem, aby moduł mobilny umożliwił uwierzytelnienie się klientowi, czyli zapewnienia, że klient połączył się z autentycznym serwerem.
- kontrola odbieranych danych: urządzenia mobilne, narażone są na nieautoryzowany dostęp. Konieczne jest, aby moduł odbiorczy kontrolował otrzymywane dane i przyjmował tylko te, które klient jest uprawniony przysyłać.

Zapewnienie bezpieczeństwa odbieranych nie jest jedynym zadaniem modułu odbiorczego. Dane po odebraniu i odpowiedniej weryfikacji powinny być przekazane do zdefiniowanych przez administratora aplikacji. Konieczne jest, aby moduł odbiorczy umożliwił przekazanie danych do dowolnej aplikacji, w tym w szczególności, do wielu z nich jednocześnie. Przekazywanie danych do wielu miejsc może być widziane przez klienta jako znaczne opóźnienie w realizacji odbioru jego danych. Należy zatem minimalizować czas pomiędzy odebraniem danych, a potwierdzeniem ich przetworzenia. Gdy moduł odbiorczy dokona potwierdzenia przetworzenia danych, jego obowiązkiem jest zapewnienie ich fizycznego dostarczenia do zdefiniowanych miejsc docelowych. Brak możliwości dostarczenia danych do miejsca docelowego powinien zostać zaraportowany jako błąd, a dane przechowane na potrzeby późniejszej synchronizacji.

Obecnie, na rynku dostępne są dodatki do systemów monitorujących pozwalające na odbieranie danych będących wynikiem pasywnych sprawdzeń wykonywanych u klientów mobilnych. Wskazana jest próba użycia tych narzędzi. Możliwa jest jednak sytuacja, w której narzędzia przeznaczone dla klientów statycznych nie będą spełniały wymagań stawianych przed tym modułem odbiorczym. Należy wówczas rozważyć modyfikację istniejącego narzędzia lub napisanie nowego od podstaw.

3.4. Moduł mobilny

Moduł ten jest odpowiedzialny za monitorowanie zadanych parametrów urządzenia mobilnego. Ponieważ klienty mobilne są od siebie niezależne i mogą oprować bez możliwości komunikacji ze sobą, konieczne jest aby każdy klient mobilny posiada swoją instancję tego modułu, która jest odpowiedzialna za monitorowanie jego urządzenia. Ten element systemu musi posiadać budowę modułową. Najważniejsze z wymagań odnoszących się do tego modułu narzuca, aby możliwe było w jak najprostszy sposób dodawanie możliwości sprawdzania nowych parametrów.

Implementacja tego modułu musi uwzględniać uwarunkowania sprzętowe jak i systemowe platformy na której się znajduje. Urządzenia mobilne są zazwyczaj zasilane z własnych akumulatorów dlatego konieczne jest zastosowanie mechanizmów, które pozwolą na zredukowanie zużycia energii związanego z systematycznym wykonywaniem sprawdzeń. Należy również wspomnieć, iż moduł mobilny odpowiedzialny jest za nadawanie każdemu z odczytów stempla czasu uniwersalnego² dokonywanego pomiaru. Na podstawie dokonanej charakterystyki klienta mobilnego, można poczynić założenie, iż klient posiada dostęp do punktów synchronizacji czasu. Jest wiele dostępnych metod synchronizacji czasu na urządzeniu mobilnym, między innymi pobranie czasu z sieci GSM czy też z serwerów czasu światowego, przez co nie stanowi to dla klienta mobilnego poważnego wymagania.

² Czas uniwersalny - średni astronomiczny czas słoneczny na południku zerowym.

Klient mobilny po zebraniu porcji wpisów dziennika o rozmiarze zgodnym z polityką administratora, lub po upływie określonego czasu powinien przesłać posiadane wpisy dziennika do modułu odbiorczego, a po uzyskaniu potwierdzenia usunąć je z urządzenia w celu oszczędności pamięci. Różnorodność platform dostępnych na rynku sprawia, iż nie można wymagać od modułu odbiorczego dostarczenia uniwersalnej implementacji protokołu komunikacyjnego. Wymaga się zatem, aby klient mobilny używał protokołu komunikacyjnego zgodnego z protokołem modułu odbiorczego. Konieczne jest również, aby klient mobilny posiadał możliwość definiowania metod uwierzytelnienia. Należy również zapewnić możliwość sprawdzenia autentyczności serwera, z którym nawiązuje się połączenie.

W ramach systemu monitorowanie funkcjonować będzie wiele instancji modułu mobilnego. Instancje te mogą używać bardzo wielu platform. W chwili pisania tej pracy nie znaleziono na rynku żadnej aplikacji przeznaczonej, na platformę mobilną, która spełniałaby to założenie. W omawianym systemie wykorzystano moduł mobilny, przeznaczony dla platformy Android, który został zaprojektowany i zaimplementowany przez Pana Marcina Kubika. Szczegółowy opis tej implementacji klienta mobilnego można znaleźć w [praca_kubika].

4. Dostępne systemy monitorujące

4.1. Przegląd systemów dostępnych na rynku

4.1.1. System monitorowania Nagios

4.1.2. System monitorowania Icinga

4.2. Porównanie i wybór systemu monitorującego

5. Monitorowanie rozproszone z użyciem NSCA

5.1. Opis dodatku NSCA

NSCA - Nagios Service Check Acceptor jest to dodatek do systemów monitorujących opartych o system Nagios, więc również systemu Icinga. Pozwala on na wykorzystanie mechanizmów pasywnego monitorowania z systemu innego niż ten na którym uruchomione jest oprogramowanie monitorujące. Program ten został napisany w całości w języku C i wydany na licencji pozwalającej na wgląd do kodu źródłowego. Wykorzystuje on plik zewnętrznych komend i nie integruje się z jądrem monitorującym. Dzięki temu możliwe jest jego wykorzystanie zarówno w systemie Nagios jak i jego klonach takich jak system Icinga. Dodatek ten składa się z dwóch modułów:

- moduł wysyłający (`send_nsca`) służący do wysyłania wyników sprawdzeń z monitorującego systemu do centralnego serwera, na którym umieszczony jest rdzeń systemu monitorującego odpowiedzialny za przetwarzanie wyników sprawdzeń,
- moduł odbierający (`nsca`) służący do odbierania wyników sprawdzeń od klientów i dostarczaniu ich do pliku komend zewnętrznych danego systemu monitorującego.

5.1.1. Moduł wysyłający

Ta część dodatku uruchamiana jest na systemie, na którym funkcjonuje jakiś mechanizm sprawdzający, który generuje wpisy dziennika. Wpisy te po utworzeniu, przekazywane są do programu wysyłającego. Moduł wysyłający, po uruchomieniu odczytuje ustawienia z pliku konfiguracyjnego, a następnie próbuje połączyć się z serwerem. Po udanej próbie połączenia otrzymuje pakiet inicjujący, który zawiera:

- wektor inicjalizacyjny: używany do celów kryptograficznych, wygenerowany przez serwer pseudolosowy ciąg znaków, konieczny do inicjalizacji algorytmu kryptograficznego,
- stempel czasu: czas odczytany przez serwer w chwili nadejścia połączenia od klienta.

Po otrzymaniu pakietu inicjującego moduł rozpoczyna czytanie wpisów z standardowego wejścia programu. Wszystkie wpisy dziennika muszą być odpowiednio sformatowane. Poszczególne pola informacyjne muszą być rozdzielone pojedynczą tabulacją, a cały wpis zakończony znakiem nowej linii. Wpisy dotyczące urządzenia powinny zawierać następujące pola:

- nazwa urządzenia: krótka nazwa urządzenia, którego stan jest przekazywany,
- stan: numerycznie wyrażony kod stanu urządzenia,
- odczyt: dodatkowe wartości odczytów opisujące stan urządzenia.

Natomast wpisy dotyczące usługi świadczonej przez to urządzenie, lub innego rejestrowanego paramatru tego urządzenia powinny zawierać następujące pola:

- nazwa urządzenia: krótka nazwa urządzenia na którym uruchomiona jest usługa,
- opis usługi: nazwa usługi danego urządzenia, której dotyczy wpis
- stan: numerycznie wyrażony kod stanu usługi,
- odczyt: dodatkowe wartości odczytów opisujące stan usługi.

Łatwo zauważyć, że żadne z pól wpisu w dzienniku nie zawiera stempla czasu wymaganego przez jądro sprawdzające przy zapamiętywaniu odczytu pasywnego. Dzieje się tak, gdyż program NSCA posiada zdefiniowaną własną politykę określania czasu wpisu w dzienniku. Do każdego pakietu zawierającego wpis dziennika dodawany jest stempel czasu otrzymany w pakiecie inicjującym od modułu odbierającego. Właściwy stempel czasu, który trafia do jądra sprawdzającego nadawany jest natomiast przez moduł odbierający.

Kolejnym krokiem działania modułu jest obliczenie cyklicznego kodu nadmiarowego CRC32 dla danego pakietu. Po dołączeniu obliczonego kodu do pakietu pakiet jest szyfrowany. Algorytm szyfrujący stosowany do szyfrowania pakietów został wcześniej zainicjalizowany wektorem pseudolosowych danych odebranych w pakiecie inicjalizacyjnym od modułu odbierającego. Po zaszyfrowaniu dane są wysyłane, a moduł wysyłający, bez oczekiwania na potwierdzenie przetworzenia przez serwer, rozpoczyna przetwarzanie kolejnego wpisu dziennika.

5.1.2. Moduł odbierający

Demon, który stanowi moduł odbierający funkcjonuje na tym samym systemie operacyjnym na którym znajduje się jądro systemu monitorującego. Ta część odpowiedzialna jest za odbieranie danych od klientów i przekazywanie ich do jądra programu monitorującego. Moduł ten może pracować w jednym z poniższych trybów:

- samodzielny demon jednoprosesowy: uruchomiony w tle demon, który nasłuchuje na przychodzące połączenia od klientów i po nadejściu połączenia jest ono obsługiwane przy użyciu jednego procesu z jednym wątkiem,
- samodzielny demon wieloprosesowy: uruchomiony w tle demon, którego proces główny nasłuchuje na nadejście połączeń od klientów, gdy takie połączenie nadejdzie proces jest duplikowany i każdy z klientów obsługiwany jest w innym procesie potomnym,
- demon zintegrowany z inetd: w systemie uruchomiony jest demon inetd, który nasłuchuje na połączenia od klientów na konkretnym gnieździe, a gdy nadejdzie połączenie od klienta uruchamiany jest proces demona NSCA, który obsługuje nowe połączenie i kończy się wraz z zakończeniem obsługi klienta

Do przekazywania wpisów dziennika używany jest mechanizm pasywnego monitorowania dostępny w systemach z rodziny Nagios. Aby możliwe było wykorzystanie tego mechanizmu konieczne jest zapewnienie demonowi dostępu do pliku zewnętrznych komend systemu monitorującego. Ponieważ plik zewnętrznych komend jest potokiem nazwanym, chroniony jest on przez Uniksowy system uprawnień użytkowników. Zapewnienie dostępu do takiego bytu może się odbyć na dwa sposoby.

Pierwszym, polecanym przez twórców systemów monitorujących, jest uruchamianie demona NSCA jako procesu tego samego użytkownika co proces jądra systemu monitorującego. Drugim sposobem jest modyfikacja praw dostępu do omawianego pliku, tak aby umożliwić dostęp użytkownikowi z którego uprawnieniami uruchomiony jest demon NSCA. Przy zastosowaniu drugiego rozwiązania zalecana jest szczególna ostrożność, gdyż dostęp do pliku zewnętrznych komend daje bardzo duże możliwości ingerencji w system monitorujący.

Komunikacja modułu odbierającego z klientem rozpoczyna się od nadejścia połączenia od klienta. Gdy moduł odbierający otrzyma nowe połączenie zostanie wysłany pakiet inicjalizujący, którego zawartość została opisana w 5.1.1. Po przesłaniu pakietu inicjalizującego połączenie, moduł odbierający oczekuje na dane od klienta. Każdy wpis dziennika przesyłany jest przy użyciu pakietu o poniższych polach:

- wersja protokołu: aktualnie używana wersja protokołu komunikacyjnego,
- kod CRC32: kod CRC32 bieżącego pakietu,
- stempel czasu: stempel czasu pochodzący z pakietu inicjalizującego przesłanego klientowi,
- kod statusu: kod stanu usługi/hosta powiązany z przesyłanym wpisem
- nazwa hosta: nazwa klienta, który podlegał sprawdzeniu. Nie jest konieczne aby był to ten sam klient, który dostarcza dane,
- opis usługi: nazwa usługi, która podlegała sprawdzeniu lub pusty napis jeśli sprawdzenie dotyczy hosta,
- wynik sprawdzenia: napis wygenerowany przez wtyczkę, która dokonywała sprawdzenia, zawierający dodatkowe dane na temat stanu urządzenia lub usługi

Pakiety są zaszyfrowane z użyciem algorytmu oraz klucza symetrycznego pochodzącego z pliku konfiguracyjnego. Po odebraniu spodziewanej ilości danych, następuje próba odszyfrowania odebranych danych. Sprawdzenie poprawności odebranych danych i jednocześnie weryfikacja uprawnień odbywa się poprzez kontrolę zawartości pola CRC32. Jeśli wartość znajdująca się w tym polu, zgadza się z wartością wyliczoną dla całości otrzymanych danych, to pakiet jest przyjmowany, w przeciwnym zaś razie pakiet zostanie odrzucony. Dalsze przetwarzanie otrzymanego pakietu rozpoczyna się od porównania bieżącego stempla czasu z tym pochodzącym z odebranego pakietu. Jeśli różnica pomiędzy nimi jest zbyt duża, dane zostają odrzucone. Ostatnią czynnością wykonywaną przez moduł odbierający jest zapisanie odebranego wpisu do pliku zewnętrznych komend jądra systemu monitorującego.

Warto wspomnieć, że stempel czasu przesłany przez klienta nie jest dostarczany do jądra monitorującego. Służy on jedynie określeniu odstępu czasu od inicjalizacji sesji do chwili otrzymania wiadomości i podjęciu decyzji o przyjęciu, bądź odrzuceniu pakietu. Do systemu monitorującego trafia natomiast bieżący stempel czasu serwera, na którym uruchomiony jest moduł odbierający i jądro systemu monitorującego. Do generacji stempla czasu wykorzystywany jest czas uniwersalny. Istotną, może się również okazać informacja, iż protokół komunikacyjny nie przewiduje przesyłania ACK¹, bądź też NACK². Moduł wysyłający, ma zatem

¹ ang. *Acknowledgement* – pozytywne potwierdzenie, powszechnie przyjęta nazwa komunikatu potwierdzającego przyjęcie i przetworzenie danych przez aplikację

² ang. *Negative-acknowledgement* – potwierdzenie negatywne, powszechnie przyjęta nazwa komunikatu oznaczająca odmowę przyjęcia lub przetworzenia odebranych danych

pewność, iż wysłane przez nie go dane zostaną dostarczone, gdyż używany jest protokół TCP, lecz nie ma żadnej gwarancji ani informacji, że dane przesłane do modułu odbierającego zostaną dostarczone do rdzenia systemu monitorującego.

5.2. Bezpieczeństwo

Bezpieczeństwo monitorowania z użyciem dodatku NSCA opiera się na kryptografii symetrycznej oraz cyklicznym kodzie nadmiarowym CRC32. Wiadomość inicjująca połączenie jest nieszyfrowana. Natomiast każda wiadomość zawierająca wpisy dziennika jest zaszyfrowana algorytmem wybranym podczas konfiguracji systemu. Dodatek NSCA korzysta z biblioteki libmcrypt i umożliwia użycie jednego spośród wielu algorytmów kryptografii symetrycznej, które zostały w niej zaimplementowane. Użytkownik posiada jedynie możliwość wyboru stosowanego algorytmu, natomiast jako tryb pracy stosowany jest tryb sprzężenia zwrotnego szyfrogramu. Tryb ten wymaga zawsze inicjalizacji zarówno kodera jak i dekodera tym samym wektorem początkowym, który w przypadku tego protokołu, jest przesyłany przez serwer w pakiecie inicjującym.

Wszystkie algorytmy symetryczne do prawidłowego działania wymagają, aby komunikujące się stowny współdzieliły pewien sekret jakim jest klucz używany do szyfrowania. Ujawnienie klucza symetrycznego wiąże się z kompromitacją całego systemu kryptograficznego. W dodatku NSCA klucz ten uzyskiwany jest z hasła, które musi być zapisane przez administratora systemu zarówno w części odbierającej jak i wysyłającej. Oczywiście jest, iż poza współdzieleniem klucza, wszystkie komunikujące się węzły muszą używać tego samego algorytmu kryptograficznego.

Algorytmy szyfrowania zapewniają tajność przesyłanej wiadomości, jednak w przypadku systemu monitorowania potrzebne jest również zapewnienie integralności wiadomości. Integralność w dodatku NSCA zapewniana jest poprzez cykliczny kod nadmiarowy CRC32. Obliczanie kodu CRC32 odbywa się poprzez dzielenie przesyłanego ciągu bitów przez dzielnik o długości 33 bitów, co daje kod CRC o długości 32 bitów. W celu sprawdzenia integralności, otrzymane bity są dzielone przez kod CRC. Jeśli reszta z dzielenia jest zero, oznacza to poprawną weryfikację integralności wiadomości. Jeśli reszta z dzielenia jest niezerowa oznacza to naruszenie integralności przesłanej wiadomości. W szczególności, taka sytuacja może się zdarzyć, gdy klient używa innego algorytmu kryptograficznego lub klucza. Pakiety, których integralność nie zostanie pozytywnie zweryfikowana są odrzucane.

Model bezpieczeństwa zastosowany w dodatku NSCA ma bardzo wiele wad. Największą z nich jest zastosowanie kodu CRC32 do sprawdzania integralności przesyłanych wiadomości. Kod ten można bardzo prosto i szybko obliczyć, a ponadto posiada on niewielką długość. Niestety jest on bardzo podatny na kolizje przez co nie powinien on być stosowany w kryptografii. Prawdopodobieństwo nie znalezienia kolizji po 200 000 prób wynosi poniżej 1%. Oznacza to iż jedynie w niespełna 1% przypadków konieczne będzie obliczenie więcej niż 200 000 kodów CRC przed znalezieniem kolizji. Prawdopodobieństwo nie znalezienia kolizji w zależności od liczby obliczonych kodów CRC32 przedstawiono w 5.1. Łatwość odnalezienia kolizji nie jest jedyną wadą modelu bezpieczeństwa zastosowanego w dodatku NSCA. Warto przypomnieć, iż wszystkie ustawienia zarówno modułu wysyłającego jak i odbierającego przechowywane są w plikach na dyskach odpowiednich urządzeń. Pliki te

Tablica 5.1. Prawdopodobieństwo nie znalezienia kolizji w zależności od liczby obliczonych kodów CRC32

Liczba obliczeń	Prawdopodobieństwo
50 000	74,7%
77 000	50,1%
78 000	49,2%
102 000	29,8%
110 000	24,5%
128 000	14,8%
150 000	7,3%
200 000	0,95%

zawierają również klucze symetryczne, które są stosowane w całym systemie. Oznacza to iż uzyskanie dostępu typu odczyt do takiego pliku powoduje utratę tajności danych przesyłanych w całym systemie. Ponadto przyjęty model bezpieczeństwa, nie zawiera żadnej weryfikacji danych pochodzących od klientów. Oznacza to, że każdy klient może przesyłać wpisy dziennika, udające wpisy pochodzące od zupełnie innych klientów. W szczególności jeśli atakujący uzyska klucz symetryczny, to nie tylko będzie mógł odczytywać informacje o wpisach przesyłanych od klientów, lecz także podszywać się pod klientów i przysyłać fałszywe wpisy. Taka luka może być wykorzystana przy ataku na jakąś usługę lub urządzenie. Atakujący rozpoczyna atak, po czym przechwytuje pakiety z wpisami dziennika, które mogą świadczyć o rozpoczęciu ataku i w zamian przysyła do serwera fałszywe pakiety informujące iż wszystkie usługi pracują normalnie.

5.3. Problemy z monitorowaniem klienta mobilnego

Dodatek NSCA jest powszechnie używany do monitorowania serwerów znajdujących się za zaporą, która uniemożliwia wykonywanie aktywnych sprawdzeń lub gdy charakterystyka monitorowanego parametru nie jest przystająca do cyklicznego odpytywania. Dodatek ten może być stosowany, w sieciach o statycznym charakterze, gdzie połączenia są stałe, a łączność nie ulega częstym przerwaniom. Ponadto należy być świadomym słabości modelu bezpieczeństwa stosowanego w protokole wymiany danych. Stosowanie dodatku NSCA poza zamkniętymi sieciami firmowymi może okazać się niebezpieczne i zawodne.

Zagadnienie monitorowania klienta mobilnego zostało szczegółowo opisane w 2. Niestety dodatek NSCA nie spełnia bardzo wielu z przedstawionych wymagań przez co nie powinien być on stosowany w systemach tego typu. Głównymi problemami, które dyskryminują dodatek NSCA w zastosowaniach do monitorowania klienta mobilnego są:

- Bezpieczeństwo: mechanizmy bezpieczeństwa zawarte w protokole wymiany danych posiadają bardzo poważne luki. Zastosowanie CRC32 do sprawdzania spójności danych niesie za sobą bardzo duże ryzyko. Ponadto konieczność przechowywania na urządzeniu klucza symetrycznego, którego ujawnienie kompromituje cały system znacząco osłabia stosowane mechanizmy bezpieczeństwa.
- Nadpisywanie stempla czasu: Moduł odbierający dodaje do każdego wpisu dziennika aktualny stempel czasu. Powoduje to brak możliwości przesyłania, historycznych danych zgromadzonych w skutek utraty dostępu do sieci.

- Brak dodatkowych mechanizmów uwierzytelnienia klienta: decyzja o przydzieleniu klientowi dostępu czyli akceptacji przesłanych przez niego wpisów dziennika podejmowana jest na podstawie znajomości przez niego algorytmu szyfrowania oraz klucza.
- Brak kontroli otrzymywanych danych: każdy klient, który zna klucz może przysłać wpisy dotyczące dowolnego urządzenia i dowolnej usługi. Brak jest mechanizmu, który pozwolił by na kontrolę tego, jaki klient ma prawo informować o jakim urządzeniu czy też usłudze.
- Brak potwierdzenia dostarczenia danych: klient wysyłający dane nie ma żadnej informacji o tym, czy jego dane zostały zaakceptowane czy odrzucone. Oznacza to brak możliwości synchronizacji danych na kliencie mobilnym i serwerze, gdyż nigdy nie mamy gwarancji, że wysłane przez klienta dane zostały przetworzone przez dodatek NSCA i przekazane do rdzenia monitorującego.
- Brak implementacji dla systemów mobilnych: moduł wysyłający jest aktualnie zaimplementowany jedynie na systemy Windows oraz Linux. Wiele współczesnych urządzeń mobilnych, które powinny być monitorowane funkcjonuje pod kontrolą systemu operacyjnego Android czy też Windows Phone.
- Przekazywanie danych tylko w jedno miejsce: dane odebrane przez moduł odbierający mogą być przekazane jedynie w jedno miejsce. Przy bardziej złożonych systemach, konieczna jest możliwość przekazywania danych do kilku systemów oraz definiowania reguł, które dane gdzie powinny trafić.

Powyższe wady zdecydowanie dyskryminują dodatek NSCA jako narzędzie do monitoringu klienta mobilnego. W związku z powyższym w tej pracy zaproponowano nowy protokół komunikacyjny, który został opisany w 7 oraz dodatek do systemów z rodziny Nagios, który implementuje ten protokół i umożliwia przekazywanie danych od klientów mobilnych. Dokładny opis modułu odbierającego dane znajduje się w 6. Moduł mobilny, z od którego dane otrzymuje wspomniany program, został zaprojektowany i zaimplementowany dla platformy Android przez Pana Marcina Kubika, a jego opis znajduje się w [praca_kubika].

6. Architektura modułu odbioru danych

6.1. Analiza

Na rynku brak jest rozwiązań dedykowanych do monitoringu klienta mobilnego. Rozwiązania przeznaczone dla klientów statycznych, takie jak dodatek NSCA, nie spełniają bardzo wielu wymagań, przez co ich użycie w budowanym systemie nie jest możliwe. W związku z powyższym zaprojektowany został i zaimplementowany moduł odbiorczy, stanowiący dodatek do systemów rodziny Nagios. Dodatek ten jest w pełni uniwersalny, można go wykorzystać, zarówno do monitorowania pasywnego klientów statycznych, jak i do monitorowania klientów mobilnych. Podczas projektowania oraz implementacji, szczególny nacisk położono na monitorowanie klienta mobilnego. Analiza wymagań przedstawionych w 2, doprowadziła do wyznaczenia podzbioru wymagań, które powinny być spełnione przez ten moduł. Należy zwrócić uwagę, że istnieje również znaczny podzbiór wymagań, powiązany z modułem odbioru danych, jednak ich realizacja jest uzależniona od wybranego protokołu komunikacyjnego, zatem one omówione w 7. W 6.1 przedstawiono podzbiór wymagań, odnoszących się do modułu odbiorczego, a także sposób ich realizacji.

Tablica 6.1: Realizacja wymagań przez moduł odbiorczy

Kod	Nazwa	Opis
W5	Dodawanie algorytmów	Możliwe jest to poprzez dostarczenie implementacji odpowiedniego interfejsu. Szerszy opis tego zagadnienia znajduje się w 6.4.
W7	Wymienne algorytmy uwierzytelnienia klienta	Możliwe jest definiowanie dowolnych metod autoryzacji klienta, poprzez dostarczenie implementacji odpowiedniego interfejsu. W celu komunikacji z klientem mobilnym został algorytm autoryzacji posiada dostęp do bezpiecznego kanału danych. Szerszy opis tego zagadnienia znajduje się w 6.5 ¹ .
Kontynuacja na następnej stronie		

¹ Konieczne jest dostarczenie również odpowiedniej implementacji algorytmu dla klienta mobilnego. Szczegóły dla platformy Android zostały opisane w [praca_kubika]

Tablica 6.1 – Kontynuacja z poprzedniej strony

Kod	Nazwa	Opis
W10	Dostarczanie w wiele miejsc	Moduł odbiorczy pozwala na przekazywanie danych do wielu lokalizacji i podsystemów docelowych, bez konieczności ich retransmisji. Konfiguracja reguł zapisana jest w pliku konfiguracyjnym. Szerszy opis implementacji tego mechanizmu znajduje się w 6.3.
W11	Reguły definiowane dla każdego klienta	Możliwe jest definiowanie reguł dostarczania danych od konkretnych klientów. Ponadto możliwe jest definiowanie grup klientów i reguł dla nich.
W12	Oszczędność pasma	Program stosuje wewnętrzne bufor, co umożliwia przesłanie potwierdzenia przetworzenia danych zanim jeszcze trafią one do miejsc docelowych. Szczegółowy opis tego mechanizmu znajduje się w 6.3
W13	Integracja z istniejącymi systemami	Moduł może być wykorzystywany z wieloma istniejącymi systemami monitorowania. Dodawanie nowych sposobów przekazywania danych do miejsca docelowego możliwe jest poprzez dostarczenie implementacji odpowiedniego interfejsu. Szerszy opis tego zagadnienia znajduje się w 6.3.
W16	Kontrola danych wejściowych	Program pozwala na definiowanie reguł, określających uprawnienia klientów do zgłaszania odczytów parametrów danego urządzenia czy też usługi. Reguły definiowane są w pliku konfiguracyjnym. Szerszy opis mechanizmu filtrowania danych znajduje się w 6.3.

6.2. Opis architektury

Wysokopoziomowa struktura logiczna programu zakłada istnienie dwóch podstawowych obiektów. Są nimi producenci danych oraz konsumenci. Dane od klienta mobilnego dostarczane są przez producentów danych, a więc muszą one zawierać implementacje protokołów komunikacyjnych oraz wszystkiego co z tym związane. Konsumenci danych natomiast zajmują się przekazywaniem danych do zewnętrznych programów. W celu przekazania danych od producenta danych do konsumentów, zgodnie z zasadami trasowania danych, potrzebny jest kanał komunikacyjny. Aby użyć kanału komunikacyjnego potrzebna jest wiadomość, a więc poprawnie sformatowana porcja danych.

Fizyczna struktura programu została utworzona z użyciem biblioteki Qt jako podstawowego szkieletu aplikacji. Wykorzystano również biblioteki boost oraz Crypto++. Moduł ten przeznaczony jest, podobnie jak system monitorujący Icinga

dla komputerów pracujących pod kontrolą systemu operacyjnego Linux i jest uruchamiany jako samodzielny serwis. Fizyczna struktura programu składa się z następujących elementów:

Szkielet programu Zawiera on elementy programu, konieczne do wytworzenia środowiska dla funkcjonowania pozostałych modułów oraz zarządzania nimi. Ponadto zawiera implementacje źródeł, a także ujść danych.

Moduł kryptograficzny Dostarcza on implementacji funkcji kryptograficznych, wymaganych podczas komunikacji z klientem mobilnym. Zawiera on zarówno algorytmy asymetryczne, konieczne do inicjalizacji kryptografii symetrycznej, jak i algorytmy symetryczne, służące do przesyłania danych.

Moduł autoryzacji klienta Zawiera on implementację algorytmów uwierzytelnienia klienta.

Moduł komunikacji z użyciem TCP Dostarcza on implementację protokołu komunikacyjnego używanego do komunikacji z klientem opartego na protokole TCP.

Moduł logowania Pozwala on na przekazywanie wiadomości z dowolnych miejsc znajdujących się w innych modułach, która zawiera informacje o zaistniałym błędzie, lub innym zdarzeniu wymagającym poinformowania użytkownika.

Odwzorowanie podstawowych elementów struktury logicznej w fizyczną znajduje się w szkielecie aplikacji. Pozostałe elementy programu zapewniają elastyczność przy rozbudowie aplikacji. Szczególnym przykładem tego usługowego charakteru pozostałych modułów może być moduł kryptograficzny i moduł autoryzacji klienta. Zostały one zaprojektowane jako moduły dostarczające dobrze zdefiniowane usługi usługi dla pozostałych elementów programu. Udostępniają one generyczne interfejsy, które są następnie w nich implementowane. Utworzenie obiektów pochodzących z tych modułów uzależnione jest od bieżącej konfiguracji programu, a ich wartość może być bardzo dynamiczna. W celu ograniczenia wpływu rozrostu liczby klas w tych modułach wykorzystano wzorzec fabryki. Każdy z omawianych modułów zawiera klasę, w której rejestrowana jest dostępność poszczególnych algorytmów. Pozostałe moduły uzyskują instancje tych algorytmów, poprzez klasę fabryki. Jeśli dany algorytm jest dostępny, zostanie on wówczas przekazany wywołującemu i będzie on mógł używać go poprzez standardowy, zdefiniowany wcześniej interfejs. Należy wspomnieć, iż poszczególne klasy fabryk wykorzystują wzorzec projektowy nazywany singleton. Oznacza to iż istnieje w programie tylko i wyłącznie jedna instancja danej fabryki, co umożliwia zachowanie spójności i dostępności danych w całym systemie. Zastosowanie wzorca projektowego fabryki pozwala pozostałym obiektom, na korzystanie z obiektów, których typ faktyczny jest nieznan w trakcie kompilacji programu. Jest to niezbędne, gdyż faktyczny typ wykorzystywanego algorytmu kryptograficznego czy też algorytmu autoryzacji użytkownika, określany jest na podstawie pliku konfiguracyjnego.

W celu konfiguracji programu wykorzystano zewnętrzny plik w formacie XML. Umożliwia to zmianę konfiguracji programu bez konieczności jego ponownej kompilacji. Plik konfiguracyjny składa się z czterech zasadniczych sekcji:

Sekcja dostawców danych zawiera dane dostawców, którzy mają zostać uruchomieni podczas startu programu. Umożliwia przekazanie dodatkowych informacji do obiektu dostawcy np. adresu IP lub portu na którym powinien on oczekiwać na połączenia. Ponadto w tej sekcji definiowane są grupy dostawców.

Sekcja odbiorców danych zawiera dane odbiorców danych, którzy mają zostać uruchomieni podczas startu programu. Umożliwia przekazanie dodatkowych informacji do obiektu odbiorcy danych, takich jak ścieżka do pliku do którego należy zapisywać dane. Sekcja ta umożliwia również definiowanie grup odbiorców.

Sekcja definicji klientów zawiera definicję klientów oraz grup klientów. Każda definicja klienta składa się z następujących sekcji:

Sekcja autoryzacji zawiera dane o dozwolonych modułach autoryzacyjnych dla danego klienta. Umożliwia także dodatkową konfigurację instancji modułów przeznaczonych dla danego klienta.

Sekcja filtrowania zawiera urządzenia oraz usługi, których dane monitorowania mogą być przesyłane przez tego konkretnego klienta.

Sekcja definicji ścieżek danych zawiera definicję ścieżek danych w programie. Pozwala na definiowanie, do którego odbiorcy danych mają trafić dane odebrane od wskazanego klienta.

Podczas uruchamiania programu, plik konfiguracyjny zostaje przeczytany oraz sprawdzony pod kątem poprawności, zarówno składniowej jak i syntaktycznej. Niestety, ponieważ obiekty dostawców oraz odbiorców danych są dostarczane z zewnątrz prawidłowość ich ustawień nie może być sprawdzana na tym samym etapie. Jest to wykonywane dopiero w trakcie inicjalizacji danego obiektu. Należy zatem zawsze po pomyślnej analizie pliku konfiguracyjnego sprawdzić zawartość pliku dziennika wykonania programu.

6.3. Szkielet programu

Moduł ten zawiera podstawowe komponenty programu. Zawarto tutaj wszystkie czynności przygotowawcze związane z wczytaniem konfiguracji oraz powołaniem do życia obiektów wymaganych przez nią. Ponadto w module tym zawarto definicję podstawowych bytów logicznych programu. W zależności od pełnionej funkcji można wyróżnić następujące grupy obiektów:

Grupa obiektów konfiguracyjnych zawiera wszystkie obiekty używane zarówno do wczytania parametrów uruchomienia programu z linii poleceń, jak również obiekty odpowiedzialne za dostarczenie do programu konfiguracji zawartej w pliku.

Grupa obiektów zarządzających zawiera zarządcę programu oraz obiekty pomocnicze. Wykonywane są tutaj wszelkie czynności, które należy wykonać w trakcie uruchamiania programu, oraz powoływanie oraz niszczenie obiektów odwzorowań głównych obiektów logicznych programu.

Grupa obiektów producentów danych zawiera generyczny interfejs producenta danych, fabrykę, umożliwiającą uzyskiwanie obiektów z tej grupy oraz definicję dostępnych producentów danych.

Grupa obiektów konsumentów danych zawiera generyczny interfejs konsumenta danych, fabrykę, umożliwiającą uzyskiwanie obiektów z tej grupy oraz definicję dostępnych konsumentów danych.

Grupa obiektów kanału komunikacyjnego zawiera obiekty powiązane z kanałem komunikacyjnym pomiędzy producentami danych, a ich konsumentami. Zawiera również mechanizmy formatowania danych oraz buforów przeznaczone na dane oczekujące na przekazanie.

6.4. Moduł kryptograficzny**6.5. Moduł autoryzacji klienta****6.6. Moduł komunikacji z wykorzystaniem TCP****6.7. Moduł logowania**

7. Protokół komunikacyjny

7.1. Podział na warstwy

7.2. Warstwa formowania wiadomości

7.3. Warstwa kryptograficzna

7.4. Warstwa integralności danych

7.5. Warstwa transportu logów

8. Testowanie i użytkowanie wykonanego systemu

8.1. Testowanie

8.2. Użytkowanie systemu

9. Podsumowanie

Bibliografia

- [1] Michael D. Ernst. *Dynamically Discovering Likely Program Invariants*. Ph.D., University of Washington Department of Computer Science and Engineering, Seattle, Washington, 2000.
- [2] Michael D. Ernst. *Daikon Invariant Detector User Manual*. 2005.
- [3] Gajek Lesław, Kałużka Marek. *Wnioskowanie statystyczne - modele i metody*. Wydawnictwa Naukowo-Techniczne, wydanie trzecie, Warszawa 1993, 1996.
- [4] Piotr Nazimek. *Inżynieria programowania kart inteligentnych*. Warszawa, 2005.
- [5] Benjamin Jack R., Cornell C. Allin. *Rachunek prawdopodobieństwa, statystyka matematyczna i teoria decyzji dla inżynierów*. Wydawnictwa Naukowo-Techniczne, wydanie pierwsze, Warszawa 1977.
- [6] Łukaszek Władysław. *Podstawy statystycznego opracowania pomiarów*. Wydawnictwo Politechniki Śląskiej, wydanie trzecie, Gliwice 1995.