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 Konienie.

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 Informacyjnych na kirunku 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

Komputer stał się nieodzowną częścią współczesnej kultury. Praktycznie każde gospodarstwo domowe posiada komputer wraz z dostępem do internet u. Urządzenia będące w posiadaniu prywatnych właścicieli, bardzo często są wykorzystywane do rozrywki lub innych czynności, których niewykonanie nie pociąga za sobą żadnych konsekwencji. Znaczna jednak część urządzeń znajduje się w posiadaniu dużych firm oraz ośrodków badawczych. Komputery te zazwyczaj są połączone ze sobą w sieć prywatną - intranet. Do ich połączenia konieczna jest zarówno rozbudowana struktura okablowania jak i zestaw urządzeń sieciowych. Wykorzystanie komputerów pozwala firmą na przyśpieszenie prac, przez co ich zysk znacząco wzrasta. Brak możliwości używania komputera, lub komunikacji poprzez infrastrukturę sieciową, niesie za sobą poważne straty finansowe. Konieczne jest zatem zapewnienie funkcjonowania całej infrastruktury, w każdej chwili, gdy jest ona potrzebna.

Urządzenia elektroniczne posiadają ograniczoną trwałość, przez co istnieje niezerowe prawdopodobieństwo awarii każdego z elementów sieci firmowej. Ponadto należy pamiętać, iż na urządzeniach uruchamiane jest oprogramowanie, które w znaczącej większości nie jest pozbawione błędów. Za awarię w danym systemie należy zatem uznać, nie tylko fizyczne uszkodzenie urządzenia, lecz także sytuacje, w której użytkownik zostaje pozbawiony dostępu do danej aplikacji. Warto również zauważyć, że użytkownik oczekuje od danej aplikacji dostarczenia usług o odpowiednim poziomie. Zatem należy również uznać, za awarię, sytuację, gdy usługi świadczone użytkownikowi nie są na satysfakcjonującym poziomie.

Użytkownik końcowy bardzo często nie jest w stanie udzielić precyzyjnej informacji o usterce, którą może on obserwować. Bardzo popularna jest sytuacja, w której użytkownik zgłasza, że nie działa aplikacja z której korzysta, natomiast faktyczną przyczyną błędu jest awaria bazy danych lub komunikacji pomiędzy nimi. Indywidualna diagnoza przy każdej awarii jest czasochłonna przez co czas do jej usunięcia wydłuża się, powodując straty finansowe. Od wielu lat w celu optymalizacji wykrywania i obsługi awarii stosuje się systemy monitorujące, które przedstawiają administratorowi w wygodny sposób stan wszystkich urządzeń oraz usług.

Zadaniem systemu monitorującego jest śledzenie stanu danego urządzenia i przedstawianie go administratorowi poprzez czytelny interfejs użytkownika. Stan urządzenia może być rozumiany bardzo szeroko. Istnieją systemy, które pozwalają na sprawdzanie, nie tylko czy urządzenia jest włączone, lecz również jego szczegółowych parametrów takich jak temperatura poszczególnych podzespołów czy zużycie prądu. Można wyróżnić dwa podstawowe rodzaje monitorowania:

Monitorowanie aktywne rodzaj monitorowania, w którym system monitorujący cyklicznie wykonuje sprawdzenie danego urządzenia lub usługi

Monitorowanie pasywne rodzaj monitorowanie, w którym status usługi lub urządzenia zgłaszany jest przez program zewnętrzny do systemu monitorującego

Każdy z sposobów monitorowania ma zarówno swoje wady i zalety. Wybór metody monitorowania, zależy zatem od charakterystyki wartości, którą monitorujemy. Jeśli wartość podlega, nieregularnym i krótkotrwałym zmianom stosuje się monitorowanie pasywne, aby nie było możliwe pominięcie owego zdarzenia. Natomiast jeśli dana wartość posiada charakterystykę zmieniającą się w sposób ciągły, należy korzystać wtedy z monitorowania aktywnego, które dokonuje próbkowania danej wartości w określonych odstępach czasu.

Sieci bardzo dużych przedsiębiorstw, nie posiadają płaskiej struktury. Ze względów bezpieczeństwa bardzo często składa się ona z sieci wirtualnych, czy wręcz odizolowanych od siebie podsieci. Ponadto sieci przedsiębiorstwa bardzo często zawierają zapory ogniowe, które filtrują ruch pomiędzy sieciami. Ze względu na fragmentacje sieci często nie jest możliwe zastosowanie prostego systemu monitorowania opisanego wcześniej. Konieczne jest zatem użycie rozproszonego systemu monitorowania.

Najprostszą realizacją rozproszonego systemu monitorowania jest użycie monitorowania pasywnego do monitorowania wszystkich urządzeń i usług, które nie są widoczne z sieci w której uruchomiony jest system monitorujący. Niestety wymaga to zmian w konfiguracji wszystkich urządzeń i uruchomienia na nim dodatkowego oprogramowania. Takie zmiany mogą nie być możliwe, na prostych urządzeniach, których kontrola odbywa się poprzez predefiniowany system producenta.

Możliwa jest również konfiguracja, wieloinstancyjnego systemu monitorowania. W każdej odizolowanej komórce sieci należy umieścić instancję systemu, która będzie zbierała dane z tej komórki sieci, zarówno w sposób aktywny jak i pasywny. Po wstępnym przetworzeniu takich danych muszą one zostać zsynchronizowane pomiędzy instancjami, a następnie umieszczone w instancji nadrzędnej lub innym miejscy docelowym. Rozwiązanie to posiada liczne zalety i jest bardzo często stosowane. Dodatkowo niektóre z systemów umożliwiają wymianę danych pomiędzy instancjami bez konieczności istnienia wyróżnionej instancji nadrzędnej.

Przedstawienie danych bieżących administratorowi jest często niewystarczające. Precyzyjna diagnoza awarii w możliwe krótkim czasie od jej wystąpienia jest bardzo ważna. Jednak istotna jest również możliwość analizy historycznych awarii, aby umożliwić wykrycie potencjalnej awarii jeszcze przed jej wystąpieniem. Są dostępne na rynku systemy, które pozwalają na gromadzenie danych o odczytach w bazach danych. Kolejnym krokiem może być analiza takiej bazy danych z wykorzystaniem systemu eksperckiego, który wykaże odpowiednie zależności i na tej podstawie możliwe będzie wykrycie awarii jeszcze przed jej wystąpieniem.

Współczesne korporacje posiadają nie tylko rozbudowaną infrastrukturę sieciową, lecz również bardzo dużą liczbę urządzeń mobilnych takich jak laptopy, tablety czy inne urządzenia specyficzne dla danej firmy. Bardzo często okazuje się, że poprawne działanie tych urządzeń wpływa znacząco na efektywność pracy, osób, które ich używają. Monitorowanie takiego urządzenie jest zadaniem nietrywialnym. Należy pamiętać, iż urządzenie mobilne może nie mieć chwilowej możliwości komunikacji z systemem monitorującym. Jeśli przerwy w łączności występują stosunkowo często, to w przypadku braku późniejszej synchronizacji danych można doprowadzić, do fałszywych predykcji systemu eksperckiego. Aby tego uniknąć konieczne jest zapewnienie dostarczenia wyników wszystkich pomiarów do systemu, kiedy tylko stanie się to możliwe. W przypadku klienta mobilnego niezwykle istotna jest również kwestia bezpieczeństwa. Urządzenia takie często nie pracują wewnątrz sieci formowej, lecz używają wielu różnych, niezaufanych sieci do komunikacji.

Dane zebrane podczas monitorowania klienta mobilnego mogą zawierać tajemnice handlowe firmy. Konieczne jest zatem zapewnienie zarówno poufności jak i integralności danych podczas synchronizacji.

Niestety, obecnie na rynku brak jest rozwiązań, które umożliwiałyby monitorowanie klienta mobilnego. Ważne jest dostarczenie odpowiedniego systemu, który pozwoli na kompleksowe monitorowanie wszystkich urządzeń występujących w firmie, zarówno mobilnych jak i statycznych. W związku z powyższym w niniejszej pracy wykonano rozbudowę popularnego systemu monitorowania, aby umożliwić monitorowanie przy jego użyciu zarówno urządzeń statycznych jak i mobilnych.

Układ tej pracy jest następujący. Rozdział 2 zawiera opis oraz porównanie dostępnych na rynków systemów monitorowania oraz wykazuje ich braki. W rozdziale XXX zdefiniowano wymagania dla całego systemu monitorowania i na ich podstawie wybrano system, który podlegać będzie modyfikacjom. Rozdział XXX zawiera opis systemu Icinga, który wybrano do rozbudowy, oraz wykazuje jakie wymagania nie zostały przez ten system spełnione. W rozdziale XXX przedstawiono opracowany projekt systemu monitorowania, zgodnego z przedstawionymi wymaganiami. Rozdział XXX natomiast zawiera opis wykonanej implementacji oraz zaprojektowanego protokołu komunikacyjnego. W rozdziale XXX zawarto opis przebiegu testowania wykonanego systemu, a także przedstawiono sprawozdanie z jego użytkowania. Rozdział XXX natomiast, zawiera podsumowanie niniejszej pracy, a także wskazuje potencjalne możliwości rozwoju wykonanego systemu.

2. Dostępne systemy monitorujące

2.1. Przegląd systemów dostępnych na rynku

Na rynku dostępnych jest wiele bardzo różnych systemów monitorujących. Narzędzia z tej grupy możemy podzielić na dwie kategorie:

— Systemy dostępnościowe
— Systemu analityczne

Systemy monitorujące w których główny nacisk jest położony na zapewnienie ciągłej dostępności monitorowanych usług. Systemy te wspierają administratora w codziennych zadaniach, poprzez nieustanne monitorowanie aktualnego stanu sieci. Narzędzia te są wykorzystywane przede wszystkim do szybkiego powiadamiania oraz lokalizacji awarii.

Systemy analityczne, w kontekście monitorowania infrastruktury sieciowej, to systemy, które są nastawione na zbieranie i analizę posiadanych danych. Tego typu systemy nie są zazwyczaj wykorzystywane do powiadamiania czy lokalizacji awarii, ich zadaniem jest przede wszystkim gromadzenie danych dotyczących zużycia poszczególnych zasobów, czy też wskaźników jakości poszczególnych usług. Systemy tego typu posiadają zazwyczaj bardzo rozbudowane narzędzia służące do generacji i analizy wykresów na podstawie zebranych wcześniej danych.

W ostatnich latach można zauważyć wzrost popularności rozwiązań hybrydowych. Pozwalają one na kompleksowe zarządzanie infrastrukturą sieciową. Dzięki zastosowaniu takiego systemu administrator uzyskuje jeden interfejs, w którym może zarówno śledzić bieżący stan sieci i diagnozować awarie, jak również prowadzić analizę danych historycznych.

Przechowywanie danych zgromadzonych podczas monitorowania może odbywać się na różne sposoby. Podstawową techniką przechowywania danych, jeszcze 5 lat temu były płaskie pliki zawierające zgromadzone dane. Rozwiązanie tego typu jest bardzo uciążliwe, a sprawne zarządzanie zgromadzonymi danymi wymaga dużego wkładu pracy własnej administratora. Obecnie rozpowszechniają się techniki przechowywania zebranych danych w oparciu o bazy danych. Współcześnie używane typy baz danych to:

— Relacyjne bazy danych
— Cykliczne bazy danych[1]

Dane przechowywane w relacyjnych bazach danych zorganizowane są w postaci tabel, a powiązania pomiędzy danymi nazywane są relacjami. Taka organizacja bazy danych sprawia, że baza danych w której gromadzone są wyniki wraz z upływem czasu rośnie. Powoduje to zwiększenie zajętości przestrzeni dyskowej, a także

[1] ang. *Round Robin Database*

wpływa na czas wykonywania operacji. Dane są przechowywane w bazie do czasu, gdy użytkownik jawnie je usunie. Pozwala to na przeglądanie dowolnie długiego okresu historii, bez utraty dokładności, a także na dynamiczne zarządzanie czasem przechowywania danych.

Cykliczne bazy danych posiadają natomiast stały, definiowany podczas tworzenia rozmiar. Rozmiar ten określa o liczbę porcji danych jaka może być przechowywana w bazie. Jeśli rozmiar bazy przekroczy rozmiar zadany przy tworzeniu, wykonywana jest konsolidacja danych. Polega ona na wyliczeniu zadanych wartości w odpowiednich przedziałach i zachowanie ich w pojedynczych rekordach, a usunięcie dokładnych danych. Możliwe są trzy typy konsolidacji danych, minimum, średnia oraz maksimum. Rozmiar bazy danych jest definiowany w chwili jej tworzenia i późniejsza modyfikacja tego rozmiaru nie jest już możliwa. Ponadto należy zwrócić szczególną uwagę, na fakt iż dane są usuwane z bazy danych bez wiedzy użytkownika, przez co taka baza danych nie może zostać użyta do dokładnej analizy danych historycznych.

Każdy typ systemu, jak i rodzaj bazy danych posiada swoje zastosowanie. Należy zatem rozważnie zdefiniować wymagania jakie stawia się przed systemem. Po dokonaniu ich analizy i analizy możliwości konkretnego systemu dokonać wyboru systemu, który system najlepiej spełnia przedstawione wymagania.

2.1.1. System monitorowania Cacti

Jest to system monitorujący, rozwijany przez The Cacti Group Inc. i dystrybuowany na licencji GPL[2]. System bazuje na narzędziu RRDtool. Jest to narzędzie, które pozwala na wykorzystanie cyklicznej bazy danych do składowania pomiarów wartości w zadanym przedziale czasowym. Ponadto narzędzie dostarcza funkcji do generacji wykresów w kilku formatach. Dzięki wykorzystaniu wspomnianego narzędzia system ma bardzo prostą budowę i składa się z następujących elementów:

— interfejs użytkownika,
— dostawca danych.

Interfejs użytkownika został napisany w języku PHP. Do jego działania niezbędny jest serwer http np. Apache. Z poziomu interfejsu użytkownika możliwa jest graficzna konfiguracja całego systemu. Interfejs posiada klasyczna budowę. Składa się on z jednokolorowego paska menu, w którym zawarte są odnośniki do poszczególnych podstron oraz z pulpitu, na którym wyświetlane są wybrane dane. Interfejs umożliwia graficzne przedstawienie wyników w postaci wykresów. Format wykresu może być definiowany bezpośrednio przez użytkownika, lub można skorzystać z bogatej biblioteki gotowych szablonów. Dostęp do interfejsu zabezpieczony jest poprzez mechanizm uwierzytelnienia użytkownika systemu monitorującego. Możliwe jest definiowanie wielu użytkowników oraz ich uprawnienia. Każdy użytkownik ma możliwość definiowania własnego zestawu wykresów oraz pulpitów.

Dostawca danych jest to element systemu, który jest odpowiedzialny za faktyczne wykonywanie sprawdzeń danej wartości i przekazywanie ich do narzędzia RRDTool. System umożliwia wybór jednego z dwóch dostawców danych. Pierwszym z nich jest cmd.php, który jest prostym skryptem napisanym w języku php.

[2] ang. *General Public License* - popularna licencja oprogramowania o otwartych źródłach. Treść licencji można znaleźć w XXX

Umożliwia on monitorowanie aktywne urządzeń przy pomocy protokołu SNMP[3]. Skrypt cmd.php przeznaczony jest do monitorowania jedynie niewielkich sieci. Ze względów wydajnościowych, nie jest możliwe wykorzystanie go do monitorowania rozległej infrastruktury.

Drugim z możliwych do wyboru dostawców danych jest program Spine, nazywany również Cacttid. Jest to program napisany w języku C, który uruchomiony jest jako serwis systemowy na urządzeniu monitorującym. Umożliwia on monitorowanie urządzeń zarówno poprzez protokół SNMP jak i z wykorzystaniem innych metod. Możliwość dostarczenia własnych metod monitorowania opiera się na dostarczeniu skryptu lub pliku wykonywalnego, który będzie cyklicznie uruchamiany przez Cactid, a jako wyniki przekazywane w taki sam sposób jak z sprawdzeń operlających się na SNMP.

Żaden z dostawców danych nie umożliwia monitorowania danego urządzenia lub usługi w sposób pasywny. Cacti nie posiada również żadnego mechanizmu, który pozwoliłby na monitorowanie sieci w sposób rozproszony. Oznacza to, iż administrator musi zmienić konfigurację sieci, tak aby jeden serwer miał dostęp do każdego urządzenia, lub konfigurować i zarządzać osobną instancją w każdym serwerze. Jest to bardzo niewygodne i wręcz uniemożliwia monitorowania rozległych sieci przy pomocy Cacti.

2.1.2. System monitorowania Nagios

System Nagios został opublikowany w 1999 na licencji GPL. System od niemal 15 lat jest ciągle rozwijany i udoskonalany, zarówno przez autorów jak i przez szeroką społeczność. W systemie Nagios najwyższym priorytetem jest dbałość o zapewnienie dostępności wszystkich monitorowanych usług. Organizacja systemu zakłada, iż w sieci znajdują się urządzenia, które mogą świadczyć pewne usługi. Każde urządzenie jak i usługa może być w jednym z trzech stanów logicznych:

OK usługa działa poprawnie
WARNING monitorowane parametry przekroczyły stan ostrzegawczy
CRITICAL parametry usługi przekroczyły stan krytyczny, usługa lub urządzenie nie funkcjonuje

System posiada rozbudowane algorytmy określania stanu każdego urządzenia oraz usługi. Działanie usługi, jest zawsze zależne od stanu urządzenie, na którym dana usługa jest świadczona. Ponadto użytkownik może definiować zależności pomiędzy urządzeniami. System Nagios posiada rozbudowany system powiadamiania administratora o wystąpieniu awarii oraz o jej zakończeniu, lub innych zdefiniowanych wydarzeniach systemowych. Ponadto możliwe jest automatyczne wykonywanie zdefiniowanych programów lub skryptów, jeśli wystąpiło jakieś zdarzenie. Podstawowa wersja systemu składa się z następujących elementów:

— Interfejs graficzny
— Rdzeń monitorujący

[3] *Simple Network Management Protocol* – protokół zarządzania urządzeniami sieciowymi i uzyskiwania informacji o ich stanie. Zorganizowany w formie drzewa, gdzie każdy liść posiada globalnie unikalny identyfikator o ściśle określonym znaczeniu. Szeroko opisany w XXX

Interfejs graficzny został napisany w języku C z wykorzystaniem technologi CGI[4]. Jego wygląd jest zgodny z standardami z lat 90. Klasyczna strona WWW bez dynamicznie zmieniającej się treści. Dane odświeżane są na żądanie klienta, lub co określony czas. Wykorzystana technologia zakłada przesyłanie za każdym razem całego dokumentu HTML do klienta, w związku z czym generowany jest nadmierny ruch sieciowy. Widok użytkownika składa się z kliku części. Po lewej stronie widoczne jest klasyczne menu, umożliwiające użytkownikowi wybór treści. Na górze strony natomiast znajduje się podsumowanie aktualnego stanu monitorowanych urządzeń i usług. Centralną część okna zajmuje pulpit, który prezentuje użytkownikowi treść wybraną wcześniej z menu. Interfejs użytkownika umożliwia podgląd aktualnego stanu usług oraz urządzeń. Informacja ta może być wyświetlana w formie listy zawierającej urządzenie i usługi, lub w postaci mapy sieci, która pozwala na monitorowanie stanu urządzenia w korelacji z jego logicznym umieszczeniem w strukturze sieciowej. Możliwe jest również przeglądanie historii awarii oraz prostych wykresów zależności stanu urządzenia lub usługi w zadanym przedziale czasu. Dostęp do interfejsu chroniony jest przy pomocy autoryzacji uwierzytelnienia http. Możliwe jest definiowanie wielu użytkowników, jednak tylko z poziomu urządzenia na którym uruchomiony jest system monitorujący. Należy zauważyć również, że wszyscy użytkownicy posiadają takie same uprawnienia do wyświetlania danych oraz zarządzania.

Rdzeń monitorujący został zaimplementowany w języku C. Jest to centrum całego systemu, gdyż zajmuje się on przetwarzaniem wszystkich bieżących danych monitorowania, a następnie składowaniem ich w plikach. Ta cześć systemu jest odpowiedzialna za wykonywanie sprawdzeń w określonych odstępach czasu. Każde sprawdzenie odbywa się poprzez wykonanie komendy zdefiniowanej przez użytkownika. Komenda ta może zawierać zarówno wykonanie pliku binarnego jak i dowolnego skryptu. W ramach projektu Nagios, rozwijany jest zestaw wtyczek[5], czyli programów służących do monitorowania podstawowych usług oraz parametrów urządzeń. Dostępna jest bardzo duża liczba wtyczek, dzięki czemu system Nagios może monitorować w sposób aktywny wszystkie podstawowe parametry lub usługi. System umożliwia również monitorowanie dowolnych usług w sposób pasywny.

System posiada rozbudowane możliwości monitorowania rozproszonego. Niestety, do wykonania znacznej części z tych konfiguracji potrzebne są elementy systemu, które są dystrybuowane za opłatą. Istnieją również darmowe dodatki, które pozwalają na przechowywanie zgromadzonych danych zarówno w bazie relacyjnej jak i cyklicznej. Możliwa jest również częściowa integracja systemu Nagios z dodatkami lub systemami, które pozwalają na wizualizacje zgromadzonych danych.

2.1.3. System monitorowania Icinga

System Icinga powstał w 2009 roku jako klon (ang. fork) systemu Nagios. System został wzbogacony o wiele nowych elementów, a także poprawiono wiele błędów obecnych w systemie Nagios. Dzięki zachowaniu wstecznej kompatybilności zarówno wszystkie wtyczki jak i dodatki systemu Nagios mogą być wykorzystane

[4] *Common Gateway Interface* – znormalizowany interfejs służący do komunikacji pomiędzy serwerem www, a zewnętrznymi programami. Interfejs ten jest wykorzystywany do generowania stron internetowych na żądanie klienta. Zewnętrzny program generuje stronę w języku HTML, a następnie serwer przesyła ją do klienta. Szczegółowy opis można znaleźć w XXX

[5] Należy zwrócić uwagę na różne znaczenie słów wtyczka (ang. *Plugin*) oraz dodatek (ang. *Addon*).

w systemie Icinga. Pozyskano temu bardzo dużą bazę wtyczek, co umożliwia monitorowanie tych samych usług i urządzeń co przodek.

System Icinga został wyposażony w zupełnie nowy interfejs graficzny[6]. Został on zaimplementowany w języku PHP przy użyciu szkieletu aplikacji agavi. Jest on zatem oparty na technologii Ajax, dzięki której komunikacja z użytkownikiem, nie opera się na przesyłaniu całych stron w języku HTML, lecz na realizacji żądań generowanych poprzez język skryptowy wykonywany po stronie użytkownika. Dzięki zastosowaniu tej technologi, proces wyświetlania strony zużywa mniejszą część pasma, a serwer został odciążony. Nowy interfejs użytkownika jest w pełni dynamiczny, składa się on z rozszerzalnego menu po lewej stronie oraz pulpitów użytkownika w centralnej części. Możliwe jest otwieranie wielu pulpitów oraz wyświetlanie poszczególnych informacji w osobnych oknach, które można swobodnie przemieszczać w obszarze strony. Znacznej zmianie uległ również model bezpieczeństwa. W nowym interfejsie graficznym, każdy użytkownik, posiada swój zestaw zdefiniowanych uprawnień. Oznacza to, że możliwe jest ograniczenie użytkownikowi dostępu do danych o konkretnej usłudze lub zabronić wykonywania niektórych czynności administracyjnych. Zarządzanie użytkownikami oraz ich uprawnieniami możliwe jest również z poziomu graficznego interfejsu użytkownika, co znacząco podnosi wygodę użytkowania systemu.

Kolejną istotną różnicą, jest zmiana architektury systemu. System Nagios posiada budowę monolityczną, a współpraca pomiędzy poszczególnymi jego komponentami odbywa się w sposób bardzo zawiły i niejednorodny. System Icinga wprowadził natomiast budowę modularną. Wszystkie możliwe komponenty systemu zostały wyodrębnione, a do swobodnej komunikacji pomiędzy nimi zdefiniowano wygodne API. Taka budowa umożliwia przede wszystkim rozmieszczenie poszczególnych komponentów systemu na różnych fizycznych maszynach, co w przypadku dużych sieci może spowodować znaczący wzrost wydajności i niezawodności. Dostarczenie jednolitego REST API[7] umożliwia również prostsze tworzenie dodatków rozbudowujących możliwości systemu. W systemie Icinga rozbudowano także możliwości współpracy z bazą danych. System ten umożliwia współpracę, już nie tylko z bazą MySQL, lecz również z bazami PostgreSQL czy też z systemem zarządzania bazą danych firmy Oracle. Możliwość wykorzystania bazy danych Oracle, jest bardzo istotna, jeśli dane dotyczące pomiarów muszą być przechowywane przez długi czas, lub jeśli monitorowana infrastruktura jest bardzo rozbudowana.

System Icinga, nie tylko umożliwia rozmieszczenie modułów na różnych fizycznych maszynach, lecz również umożliwia wiele innych konfiguracji, które można wykorzystać podczas monitorowania rozproszonego. Szczególnie wartą zauważenia jest konfiguracja, w której występuje wiele równorzędnych instancji rdzenia monitorującego, natomiast wszystkie współpracują używając jednej bazy danych. Centralna baza danych stanowi źródło danych dla interfejsu graficznego. Taka konfiguracja umożliwia monitorowanie bardzo rozległej lub wielosegmentowej infrastruktury. Należy również nadmienić, iż wszystkie elementy niezbędne do konfiguracji takiego rozwiązania są darmowe.

[6] Skorzystanie z nowego interfejsu wymaga użycia modułu IDOUtils oraz bazy danych. Możliwe jest wykorzystanie również klasycznego interfejsu, który nie posiada takich wymagań.

[7] ang. *Representational state transfer* – lekka metoda przesyłania danych pomiędzy klientem a serwerem.

2.2. Podsumowanie

Współczesne systemy monitoringu, są bardzo bogato wyposażone i posiadają szereg zaawansowanych możliwości. Każdy z systemów oferuje unikalny zestaw rozwiązań, które z pewnością mogą zostać wykorzystane w wielu instytucjach. Porównując wszystkie omówione systemy, należy zwrócić szczególną uwagę, na różnice w ich możliwych zastosowaniach docelowych.

Systemy, takie jak Cacti zaliczane są do grupy systemów analitycznych. Ich celem jest zatem zapewnienie możliwości gromadzenia oraz analizy danych. Zbierane dane mają charakter pojedynczych, dokładnych wartości, na podstawie których prezentowane są użytkownikowi odpowiednie wykresy. Niestety ze względu na sposób gromadzenia danych - protokół SNMP, oraz ubogość metod ich gromadzenia systemy te, nie mogą być wzbogacone o funkcjonalność charakterystyczną dla systemów dostępnościowych.

Drugą grupę systemów stanowią natomiast systemy dostępnościowe, takie jak Nagios czy Icinga. Ich głównym celem jest monitorowanie bieżącego stanu infrastruktury i raportowanie użytkownikowi najświeższych informacji. Systemy te zostały również zaprojektowane, aby wspomagać administratora w lokalizacji awarii. Głównym typem danych na których operują te systemy jest stan urządzenia lub usługi. Zdefiniowanie odpowiednich poziomów kwantyzacji dla stanów pozwala na szybkie uzyskiwanie poglądowych informacji o stanie sieci. Podczas monitorowania gromadzone są również dane szczegółowe. Ich przetwarzaniem nie zajmują się już jednak same systemy monitorowania, lecz liczne dodatki do nich. Możliwe jest zatem rozbudowanie systemu tego typu, o dodatkowe elementy, które pozwolą uzyskać system hybrydowy. System taki będzie mógł pełnić rolę zarówno systemu dostepnościowego jak i analitycznego.

Wybierając system monitorujący, należy zatem dokonać szczegółowej analizy wymagań stawianych przed systemem. Szczegółowe porównanie wszystkich przedstawionych systemów monitorowania zawarto w 2.1.

Tablica 2.1: Porównanie systemów monitorowania

Nazwa systemu	Cacti	Nagios	Icinga
Podgląd stanu bieżącego	Nie	Tak	Tak
Podgląd danych historycznych	Tak	Tak, przez dodatek	Tak, przez dodatek
Dane w formie wykresu	Tak	Tak, przez dodatek	Tak, przez dodatek
Przechowywanie danych w bazie cyklicznej	Tak	Tak, przez dodatek	Tak, przez dodatek
Przechowywanie danych w bazie relacyjnej	Nie	Tak, przez dodatek	Tak, przez dodatek
Powiadomienia o awarii	Nie	Tak, email lub telefon	Tak, email lub telefon

Kontynuacja na następnej stronie

Tablica 2.1 – Kontynuacja z poprzedniej strony

Nazwa systemu	Cacti	Nagios	Icinga
Wsparcie w lokalizacji awarii	Nie	Tak, poprzez mapę logiczną sieci	Tak, poprzez mapę logiczną sieci
Obsługa SNMP	Tak	Tak, przez wtyczkę	Tak, przez wtyczkę
Zbieranie danych spoza SNMP	Tak, niewielka liczba dostępnych metod	Tak, bogaty zestaw wtyczek	Tak, bogaty zestaw wtyczek
Monitorowanie pasywne	Nie	Tak	Tak
Nowoczesny interfejs użytkownika	Nie	Nie	Tak, z wykorzystaniem technologi AJAX
Wielu użytkowników	Tak	Tak	Tak
Metoda uwierzytelnienia	Uwierzytelnienie wewnętrzne	Uwierzytelnienie http	Uwierzytelnienie wewnętrzne
Zarządzanie kontami użytkowników z interfejsu	Tak	Nie	Tak
Definiowanie uprawnień dla użytkowników	Tak, przez interfejs graficzny	Nie	Tak, przez interfejs graficzny
Modularność	Nie	Nie	Tak
Rozmieszczenie modułów na różnych urządzeniach fizycznych	Nie dotyczy	Nie dotyczy	Tak
Możliwość monitorowania rozproszonego z instancją nadrzędną	Nie	Tak	Tak
Możliwość monitorowania rozproszonego bez instancji nadrzędnej	Nie	Tak, konieczny płatny dodatek	Tak
Generacja raportów	Nie	Nie	Tak, z wykorzystaniem JasperReports
Możliwość monitorowania klienta mobilnego	Nie	Nie	Nie

Kontynuacja na następnej stronie

Tablica 2.1 – Kontynuacja z poprzedniej strony

Nazwa systemu	Cacti	Nagios	Icinga
Dostępność	Darmowy	Częściowo darmowy, wiele płatnych elementów i funkcjonalności	Darmowy
Licencja	GPL v2	GPL v3 (tylko darmowe elementy)	GPL v2

Przedstawione systemy monitorujące w znacznym stopniu zaspokajają zapotrzebowanie rynku na systemy monitorowania. Pojawia się jednak pewna nisza związana z monitorowaniem urządzeń mobilnych. Zadanie to nie jest trywialne i wymaga obecności dodatkowych mechanizmów zarówno na urządzeniu mobilnym, jak i w innych elementach systemu. Żaden z analizowanych systemów nie posiadał w swej implementacji ani w oficjalnych repozytoriach z dodatkami, oprogramowania, które pozwalałoby na monitorowanie parametrów urządzenia mobilnego.

3. Monitorowanie klienta mobilnego

3.1. Monitorowanie rozproszone klientów statycznych

Firmy działające obecnie na rynku posiadają bardzo rozbudowaną infrastruktę informatyczna. 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 klienty statyczne. Urządzenia tego typu zazwyczaj pracują nieprzerwanie lub w dobrze zdefiniowanych przedziałach czasowych 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 urządzemoe 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 możliwosci używania niektórych mechanizmów dostępnych wyłącznie dla urządzeń monitorowanych aktywnie. Ponadto wyniki sprawdzeń pasywnych nie są akumulowane, lecz wysyłane odrazu 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 jest to jednak błąd pomijalny. Błąd ten staje się jednak istotny, gdy jednym z zadań systemu, jest gromadzenie i analiza

danych historycznych. Wieloinstancyjny system minitorujący 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, że na przykład system Icinga, daje możliwość integracji wielu instancji jądra monitorującego, przy pomocy wspólnej bazy danych. Dzięki temu administrator danej sieci ma możliwość monitorowania i kofigurowania wielu instancji przy pomocy wspólnego interfejsu. Niestety w systemie Nagios rozwiązanie to zaliczane jest do cześci korporacyjnej tego systemu, przez co posiada zamknięte źródła i jego wykorzystanie wymaga zakupu licencji. Rozwiązania oparte na istnieniu jednej centralnej instancji jądra systemu monitorujacego, do której przesyłane są, gdy jest to możliwe odczyty wykonane przez inne instancje, są zazwyczaj darmowe lecz wymagają dodatkowej instancji, zajmującej sie agregacją danych. Nalezy również zwrócić uwagę, iż niektóre sytemy jak Cacti nie posiadają w ogóle możliwosći rozproszonego monitorowania.

3.2. 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 zauważyć, że urządzenia te są często narażone na dostęp do sieci, o bardzo niskim poziomie zaufania i wielu zagrożeniach. Oznaczia 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 siecią 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 pojedyńczo. Większość klientów mobilnych operuje w pełni samodzielnie, zatem liczność grupy klientów wynosi 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. Klienty 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 sie 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 ukierunkowana na pomiar usług świadczonych przez dany system na rzecz innych systemów. Natomiast w przypadku klientów mobilnych istotniejsze wydaje się być monitorowanie parametrów wewnętrznych danego klienta.

3.3. Wymagania systemu monitorowania klientów mobilnych

Klient mobilny posiada zdecydowanie odmienną charakterystykę niż klient statyczny. Dokonano zatem analizy, jakie wymagania należy spełnić, aby dostarczyć system, który sprosta oczekiwaniom administratów urządzeń mobilnych jak i statycznych.

Odbiorcą systemu mają być duże firmy i korporacje, które posiadają bardzo rozbudowaną sieć wewnątrz firmy, a ponadto udostępniają swoim pracownikom urządzenia mobilne różnej klasy. Wsród tych urządzeń znajdują się przedewszystkim 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 minitorowania 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 powiny 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 przesyłać 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 3.1.

Tablica 3.1: 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 uprawionego 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 umożliwać 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 kliena. Ponadto system musi umożliwaić 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	Systes musi być odporny na zgubienie urządzenie. 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 retransmisjii z klienta mobilnego.
W11	Reguły definiowane dla każdego klienta	System musi umożliwiać definowanie 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 3.1 – 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 bierzących	System musi umożliwiać prezentację oraz analizę danych bierzących, 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

3.4. Podstawowe decyzje projektowe

Przedstawione wymagania pozwalają na opracowanie systemu, który zaspokoi potrzebę monitorowania klienta mobilngo. System monitorujący stanowi duży zestaw programów wykonujacych się na różnych urządzeniach i w różnych kontekstach. Zaprojektowanie i implementacja od podstaw systemu monitorującego, który spełniłby wszystkie przedstawione wymagania, daleko wykracza, poza ograniczenia czasowe pracy inżynierskiej. Ponadto dobre praktyki programistyczne, nakazują możliwie szerokie wykorzystanie gotowych programów. Nalezy również pamietać, iż każdy program wymaga testowania i późniejszego utrzymania jego kodu. Wykorzystanie gotowego systemu, pozwala na uzyskanie niskim nakładem czasu systemu, który został już dokładnie przetestowany, a utrzymanie jego kodu zapewniane jest poprzez osoby zewnętrzne.

W związku z powyższym w niniejszej pracy podjęto decyzję, aby budowany system monitoringu klienta statycznego był oparty o jeden z dostępnych darmowych systemów monitorowania. Na podstawie analizy systemów monitorujacych dostępnych na rynku, dokonanej w 2, został wybrany system monitorujący Icinga.

Wybór ten podyktowany jest wieloma zaletami tego systemu. Przede wszystkim, nalezy zauważyć przemyślaną architekturę. Ponieważ posiada on budowę modularną, możliwe jest jego instalowanie na wielu odrębnych urządzeniach co znacząco moze przyśpieszyc analizę danych. Możliwe jest również monitorowanie infrastruktury w sposób rozproszony. System ten umożliwia zarówno monitorowanie pasywne, jak i wieloinstancyjne. Należy również wyróżnić system Icinga, gdyż jako jedyny udostępnia on w sposób darmowy, możliwość wspólnego zarządzania i podglądu wieloinstancyjnego systemu monitorującego. Nowoczesny i dynamiczny interfejs użytkownika dostarczany przez ten system, może być w łatwy sposób rozszerzany o dodatkowe funkcjonalności. Na szczególne uznanie zasługuje również rozbudowana i na bieżąco aktualizowana dokumentacja projektu. Najważniejszą z zalet jest jednak popularność tego systemu wśród administratorów. Dowodem popularności i wiarygodności systemu Icinga może być jego zastosowanie w ośrodku badań Europejskiej Organizacji Badań Jądrowych CERN.

Rdzeń monitorujący systemu Icinga jest przeznaczony dla systemu Linux, jednak możliwe jest uruchomienie go na większości systemów z rodziny Unix. W związku z powyższym wszelkie rozwiązania zaimplementowane w ramach tej pracy, są przeznaczone, dla tych samych systemów co jądro monitorujące.

4. System monitorowania Icinga

4.1. Opis systemu

System Icinga powstał jako klon systemu Nagios. Zachowana została kompatybilność wsteczna przez co możliwe jest używanie dodatków oraz wtyczek przeznaczonych dla systemu Nagios. Podstawowa konfiguracja systemu Icinga składa się dwóch modułów:

— Rdzeń monitorujący
— Interfejs użytkownika

Rdzeń monitorujący stanowi element centralny całego systemu. Do jego zadań należy przede wszystkim monitorowanie usług i urządzeń zgodnie z ustawieniami zawartymi w plikach konfiguracyjnych. System posiada możliwość monitorowania zarówno aktywnego jak i pasywnego. Monitorowanie aktywne wykonywane jest przy pomocy zewnętrznych programów lub skryptów nazywanych wtyczkami. Każda usługa oraz urządzenia posiada zdefiniowaną komendę sprawdzającą która definiuje jaką wtyczkę należy uruchomić oraz jakie dane do niej przekazać. Każda wtyczka posiada zdefiniowany zestaw danych wejściowych, które odpowiadają jej opcjom konfiguracyjnym. Po zakończeniu pomiaru wtyczka przekazuje dane o jego wynikach do systemu monitorującego. Przekazanie to odbywa się dwiema drogami. Pierwsza z nich to wartość zwrócona z programu, która determinuje w jakim stanie znajduje się urządzenie lub usługa. Zwrócona wartość powinna być jedną z następujących:

0 OK, wtyczka mogła wykonać sprawdzenie i usługa lub urządzenie jest w stanie OK
1 WARNING, wtyczka mogła wykonać sprawdzenie ale parametry urządzenia lub usługi przekraczają poziom ostrzegawczy.
2 CRITICAL, wtyczka mogła wykonać sprawdzenia ale parametry urządzenia lub usługi przekraczają poziom krytyczny.
3 UNKNOWN, wtyczka nie była w stanie wykonać sprawdzenia ze względu na dostarczenie nie prawidłowych parametrów wywołania lub niskopoziomowego błędu systemu.

Większość wtyczek dokonuje pewnych pomiarów, dlatego każde ich wykonanie gromadzi zdecydowanie więcej danych niż można przekazać poprzez jedną z czterech wartości. Przekazanie pozostałych informacji odbywa się poprzez dane tekstowe, wypisywane przez wtyczkę na standardowym wyjściu programu. Dane te rozdzielane są znakiem | na dwie grupy. Przed tym znakiem, znajdują się dane czytelna dla człowieka. Po znaku znajdują się dane wydajnościowe w formacie klucz=wartość, przeznaczone do analizy przez zewnętrzne programy np. do generacji wykresów. Znak | oraz druga grupa nie są obowiązkowe.

System Icinga umożliwia również monitorowanie w sposób pasywny. Dostarczanie wyników sprawdzenia pasywnego odbywa się poprzez plik komend zewnętrznych. Plik komend zewnętrznych jest to potok nazwany przez który poszczególne komendy trafiają do rdzenia monitorującego. Pełna lista dostępnych komend znajduje się w XXX. Zatem pewien dowolny program wykonuje sprawdzenia urządzenia lub usługi, po czym zapisuje wynik tego sprawdzenia zgodnie z formatem do potoku. Format przekazywanego sprawdzenia został opisany w XXX. Jeśli program wykonuje się na urządzeniu innym niż to na którym uruchomiony jest rdzeń monitorujący konieczne jest przesłanie danych do tego systemu. System Icinga posiada do tego celu dodatek NSCA, który został szeroko opisany w 4.4

Dane otrzymane od wtyczki są przez system monitorujący przechowywane wraz z innymi danymi potrzebnymi do monitorowania w plikach, a gdy przestają one być potrzebne, są kasowane. System Icinga udostępnia jednak możliwość przechowywania tych danych w bazie danych przy pomocy komponentu systemu IDOUtils. Został on dokładnie opisany w 4.2. Rdzeń monitorujący posiada także możliwość udostępniania danych wydajnościowych otrzymanych przez każdą wtyczkę dla zewnętrznych programów. Możliwe jest zdefiniowanie formatu danych wyjściowych. Po włączeniu eksportu danych wydajnościowych, rdzeń monitorujący będzie zapisywał do zdefiniowanych plików dane wydajnościowe pochodzące od wtyczek jak i czasy ich przybycia. Dane te są wykorzystywane np. przez dodatek inGraph opisany w 4.3.

System Icinga odziedziczył po systemie Nagios klasyczny interfejs oparty o technologię CGI. Ponieważ technologia ta jest już przestarzała, udostępniono użytkownikom również w pełni nowoczesny interfejs nazywany icinga-web. Jest to dynamiczny interfejs użytkownika zaimplementowany w języku PHP w oparciu o technologię AJAX. Różnice pomiędzy tymi interfejsami zauważalne są nie tylko w warstwie prezentacji lecz również w architekturze i warstwie komunikacji z rdzeniem monitorującym. Interfejs klasyczny wykorzystywał pliki generowane przez rdzeń monitorujący do pobierania aktualnych danych, przez co musiał on znajdować się na tym samym urządzeniu co rdzeń monitorujący. W przypadku interfejsu icinga-web komunikacja z rdzeniem monitorującym odbywa się poprzez bazę danych, dzięki czemu elementy te mogą znajdować się na różnych fizycznych urządzeniach. Należy również dostrzec różnice w bezpieczeństwie. Interfejs klasyczny zabezpieczony był jedynie poprzez mechanizm uwierzytelnienia http i wszyscy użytkownicy mieli dostęp do całego systemu. Interfejs icinga-web posiada swoją bazę użytkowników oraz ich uprawnień. Umożliwia to ograniczenie praw danego użytkownika np. tylko do wyświetlania stanu konkretnego urządzenia lub usługi.

4.2. Komponent IDOUtils

Komponent IDOUtils jest to zestaw programów dzięki którym możliwe jest składowanie informacji generowanych przez rdzeń monitorujący w bazie danych. W wersji dostępnej podczas pisania tej pracy wspierany były następujące systemy zarządzania baza danych:

— MySql,
— PostgreSQL,
— Oracle.

W celu zapewnienia funkcjonalności omawianego komponentu, konieczne jest utworzenie bazy danych o odpowiednim schemacie, który został opisany w XXX. Udostępnione zostały również skrypty SQL, które definiują odpowiednie tabele. Ponadto administrator musi zapewnić odpowiednia konfigurację bazy danych, w tym konto użytkownika i hasło, w taki sposób, aby umożliwić odpowiednim elementom komponentu IDOUtils dostęp do bazy danych.

W celu odciążenia urządzenia, na którym uruchomiony jest system Icinga. Komponent ten został podzielony, na kilka elementów, które mogą znajdować się na różnych urządzeniach. Można wyróżnić następujące elementy:

IDOMOD moduł rdzenia monitorującego, który pozwala mu na dostęp do bazy danych

LOG2IDO program pozwalający na import utworzonych wcześniej plików do bazy danych

FILE2SOCK program pozwalający na przekierowanie danych zapisywanych do pliku do gniazda TCP lub Unix

IDO2DB demon, który jest odpowiedzialny za wykonywanie operacji na bazie danych

Podstawowymi elementami całego komponentu są IDOMOD oraz IDO2DB. Moduł rdzenia IDOMOD ładowany jest przez rdzeń systemu Icinga tuż po starcie. Po załadowaniu zapewnia on spójny interfejs do uzyskiwania danych dla wszystkich pozostałych części rdzenia monitorującego. Ponieważ wykonywanie operacji na bazie danych może być czasochłonne nie powinno być to wykonywane przez rdzeń monitorujący. Z tego powodu powstał program IDO2DB. Jest on uruchomiony jako demon na dowolnym urządzeniu. Zadaniem tego serwisu jest fizyczna realizacja żądań na bazie danych.

Ponieważ rdzeń monitorujący oraz demon IDO2DB mogą znajdować się zarówno na jednym urządzeniu jak i na różnych urządzeniach konieczne jest zapewnienie odpowiednich mechanizmów komunikacji pomiędzy nimi. Gdy programy te znajdują się na różnych urządzeniach, jako mechanizm komunikacji wykorzystywane są gniazda TCP. W podstawowej konfiguracji dane przekazywane są w sposób niezaszyfrowany. Jeśli jednak istnieje potrzeba zapewnienia tajności oraz integralności przekazywanych danych możliwe jest użycie protokołu SSL[1]. W sytuacji, gdy oba programy uruchomione są na tym samym urządzeniu, w celu poprawy wydajności możliwe jest użycie gniazd protokołu Unix[2].

W celu zapewnienia możliwości migracji z środowiska, które korzystało wcześniej z przechowywania danych w plikach, został dostarczony program LOG2IDO. Pozwala on, na import danych historycznych do bazy danych. Program ten, analogicznie jak IDOMOD nie operuje bezpośrednio na bazie danych, lecz komunikuje się tymi samymi metodami co IDOMOD z demonem IDO2DB. Zarówno program LOG2IDO jak i moduł IDOMOD mogą kierować żądania do IDO2DB poprzez plik. W celu zapewnienia przekazywania tych danych z pliku do demona IDO2DB opracowano program FILE2SOCK. Jest to prosty program, który przekazuje dane zapisane do danego pliku do demona IDO2DB. Program ten nie zajmuje się w żadnym

[1] ang. *Secure Socket Layer* – protokół warstwy prezentacji, zapewniający poufność oraz integralność przesyłanych danych.

[2] and. *Unix Domain Socket* – metoda komunikacji między procesowej w systemach Unix. Posiada jednolite API jak gniazda domeny internetowej.

stopniu przetwarzaniem odczytaniem danych, lecz jedynie przesłaniem ich poprzez gniazdo internetowe lub Unix do demona IDO2DB.

4.3. Dodatek inGraph

inGraph jest to dodatek do systemów Icinga oraz Nagios, który umożliwia prezentację danych zgromadzonych poprzez system monitorujący w postaci wykresów. Dodatek ten został opracowany przez firmę NETWAYS GmbH i wydany na licencji GPL w wersji 3. Cechą, która odróżnia dodatek inGraph od innych rozwiązań, przeznaczonych do analizy danych historycznych jest wykorzystanie relacyjnej bazy danych do przechowywania danych otrzymanych od systemu monitorującego. Dzięki wykorzystaniu relacyjnej bazy danych, możliwe jest zarówno przeglądanie danych dokładnych w przedziale czasu w historii, jak i wykresów długoterminowych prezentujących trendy danej wartości.

Dodatek inGraph składa się z dwóch niezależnych elementów, komunikujących się poprzez XMLRPC[3]:

— interfejs graficzny
— rdzeń zbierający dane

Rdzeń zbierający oraz przetwarzający dane został napisany w języku Python. Jego zadaniem jest pobieranie danych od systemu monitorującego, dokonywanie ich przeliczeń, oraz umieszczanie ich wyników w bazie danych. Do pobierania danych z systemu monitorującego wykorzystano mechanizm udostępniania danych wydajnościowych. System monitorujący, musi eksportować dane przy pomocy formatu zrozumiałego dla dodatku inGraph. Demon zbierający dane dokonuje analizy otrzymanych danych, a następnie wykonuje wszystkie niezbędne obliczenia, a wyniki zapisuje w bazie danych MySql lub PostgreSQL. Ważną różnicą pomiędzy danymi składowanymi w tej bazie, a danymi przechowywanymi przez system monitorujący jest ich format. Systemy monitorujące, przechowują w postaci numerycznej jedynie skwantowany stan danej usługi lub urządzenia. Dodatek inGraph przechowuje natomiast w swojej bazie dane w postaci już przetworzonej. Oznacza to iż dokonywany jest rozbiór składniowy rezultatów pomiarów i w bazie danych zapamiętywane są pochodzące z tych rezultatów dane w postaci numerycznej.

Interfejs użytkownika został napisany w językach PHP oraz JavaScript. Umożliwia on podgląd danych zebranych i przetworzonych przez rdzeń dodatku. Interfejs może funkcjonować zarówno jako niezależny serwis jak i jako integralna część interfejsu systemu Icinga. Umożliwia on generację wykresów dla każdego z urządzeń oraz dla każdej z usług. Formaty wykresów, a także przedziały agregacji danych definiowane są w plikach konfiguracyjnych w formacie JASON[4]. Użytkownik po wybraniu usługi lub urządzenia uzyskuje interaktywny wykres przezentujący dane w zadanym okresie. Wszystkie wykresy wygenerowane przez program są w pełni konfigurowalne jak i edytowalne. Typ prezentowanych danych jest uzależniony od rozmiaru przedziału czasu w którym generowany jest wykres. Jeśli okno czasu jest odpowiednio małe, na wykresie zostaną przedstawione dane dokładne. W sytuacji,

[3] ang. *XML Remote Procedure Call* – zdalne wywołanie procedur przy użyciu XML. Metoda zdalnego wywoływania funkcji oparta na formacie XML. Szczegółowy opis w XXX.

[4] ang. *JavaScrip Object Notation* – lekki format tekstowy wymiany danych komputerowych. Szczegółowo opisany w XXX

gdy nie jest możliwe przedstawienie danych dokładnych, ze względu na rozmiar zadanego okresu czasu, dane są agregowane w przedziały, a na wykresie udostępniana jest wartość minimalna, maksymalna oraz średnia dla danego przedziału agregacji danych.

4.4. Dodatek NSCA

4.4.1. 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.

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 inicjujący: 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 dotyczącego 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.

Natomiast wpisy dotyczące usługi świadczonej przez to urządzenia, lub innego rejestrowanego parametru 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 dziennika nie zawiera stempla czasu wymaganego przez rdzeń sprawdzający 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 jadra 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 kryptograficzny 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.

Moduł odbierający

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

— samodzielny demon jedno procesowy: 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 wieloprocesowy: 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 rdzenia 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 4.4.1. 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 ACK[5], bądź też NACK[6]. Moduł wysyłający, ma zatem 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] ang. *Acknowledgement* – pozytywne potwierdzenie, powszechnie przyjęta nazwa komunikatu potwierdzającego przyjęcie i przetworzenie danych przez aplikację

[6] ang. *Negative Acknowledgement* – potwierdzenie negatywne, powszechnie przyjęta nazwa komunikatu oznaczająca odmowę przyjęcia lub przetworzenia odebranych danych

4.4.2. 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 strony 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. Oczywistym 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 poprawna 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 4.1. 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 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 przesłać wpisy dziennika, udające wpisy pochodzące od zupełnie innych klientów. W szczególności jeśli atakujący uzyska klucz symetryczny, to nie

Tablica 4.1. 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%

tylko będzie mógł odczytywać informacje o wpisach przesyłanych od klientów, lecz także podszywać się pod klientów i przesyłać 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 przesyła do serwera fałszywe pakiety informujące iż wszystkie usługi pracują normalnie.

4.5. Podstawowe konfiguracje rozproszone

Podstawowa konfiguracja systemu monitorującego Icinga składa się jedynie z rdzenia monitorującego oraz klasycznego interfejsu użytkownika. W tej konfiguracji, zarówno ustawienia systemu monitorującego, jak i dane o stanie usług i urządzeń znajdują się w plikach lokalnych. Jeśli nie zostaną użyte żadne dodatkowe mechanizmy transportu danych, obie części systemu Icinga będą musiały być wykonywane na jednym urządzeniu. Jeśli monitorowana infrastruktura jest bardzo rozbudowana, a administrator często i intensywnie korzysta z interfejsu graficznego, to umiejscowienie obu tych elementów na jednym urządzeniu może powodować jego znaczące obciążenie i zaburzenia w prawidłowym monitorowaniu infrastruktury. Należy również zwrócić uwagę na zagadnienie bezpieczeństwa takiego rozwiązania. Jeśli administrator chciałby udostępnić interfejs użytkownika poza monitorowaną sieć, musi on zezwolić na dostęp z zewnątrz do urządzenia, które monitoruje całą infrastrukturę. Obniża to bezpieczeństwo w sieci, gdyż atakujący może ukierunkować swoje działania właśnie na to urządzenie, a uzyskanie dostępu do niego pozwoli na ataki innych, być może słabiej zabezpieczonych urządzeń znajdujących się w sieci.

Podstawową metodą optymalizacji przedstawionej konfiguracji jest rozmieszczenie rdzenia monitorującego oraz interfejsu użytkownika na różnych urządzeniach fizycznych. Umożliwienie rozdzielenia tych dwóch bytów wymaga zapewnienia im wspólnego miejsca, w którym składowane będą dane konfiguracyjne, dane zawierające bieżący stan sieci oraz reprezentację powstałych zdarzeń. System Icinga wykorzystuje do tego celu relacyjną bazę danych. Klasyczny interfejs nie wspiera komunikacji poprzez bazę danych, dlatego należy wykorzystać interfejs icinga-web. Zapewnienie współpracy rdzenia monitorującego z baza danych odbywa się poprzez komponent IDOUtils opisany w 4.2. System składa się zatem z następujących elementów:

— rdzeń monitorujący

— baza danych
— interfejs graficzny

Dzięki modularnej budowie całego systemu możliwe jest umieszczenie każdego z wymienionych elementów na osobnym urządzaniu fizycznym. Umożliwia to odciążenie urządzenia, na którym uruchomiony jest rdzeń monitorujący. Ponadto zwiększone zostaje bezpieczeństwo całego rozwiązania, gdyż konieczne jest udostępnienie na zewnątrz jedynie serwera na którym znajduje się interfejs sieciowy. Urządzenie to musi mieć dostęp do bazy danych, lecz nie musi mieć dostępu do urządzenia, na którym umieszczony jest rdzeń monitorujący oraz do całej monitorowanej infrastruktury. Pozwala to na umieszczenie rdzenia monitorującego razem z monitorowaną infrastrukturą za zaporą ogniową, co ogranicza możliwości ingerencji w system monitorowania i infrastrukturę sieciową.

Przedstawiona architektura stanowi bardzo dobrą konfigurację dla firm posiadających jednolitą infrastrukturę sieciową o średniej wielkości. Istnieją jednak sieci dla których przedstawiona architektura może okazać się niewystarczająca. Jedna z takich sytuacji ma miejsce, gdy instytucja posiada sieć złożoną z kilku segmentów czy to ze względu na separacje czy też lokalizacje geograficzną. Przedstawiona architektura nie umożliwia monitorowania aktywnego, urządzeń znajdujących się za zaporą ogniową. Możliwe jest monitorowanie pasywne takich usług jednak wymaga ono ingerencji w monitorowane serwery. Kolejna z sytuacji ma miejsce, gdy monitorowana infrastruktura, jest na tyle rozbudowana, że urządzenie na którym uruchomiony jest rdzeń nie posiada wystarczającej ilości zasobów, aby monitorować wszystkie urządzenia i usługi. Obie te sytuacje wymagają monitorowania przy jednoczesnym użyciu wielu instancji rdzenia monitorującego.

Pierwszy z możliwych scenariuszy współpracy wielu instancji rdzenia monitorującego wymaga zastosowania dodatku NSCA omówionego w 4.4. Konfiguracja ta zakłada istnienie jednej wyróżnionej instancji rdzenia monitorującego, która będzie odpowiedzialna za przetwarzanie wszystkich wyników sprawdzeń, a także generacje zdarzeń i powiadomień. Konieczne jest również zapewnienie możliwości komunikacji z co najmniej jednego urządzenia w każdym segmencie sieci. Konfiguracja ta została oparta o mechanizm pasywnego sprawdzana usług i urządzeń. Instancja centralna posiada wszystkie usługi skonfigurowane w taki sposób, aby możliwe było dostarczanie pasywnych wyników sprawdzeń tych usług. Na tym samym systemie, co wyróżniona instancja rdzenia uruchomiony jest również serwis systemowy NSCA, który oczekuje na dane przesyłany z instancji roboczych. Każda z instancji roboczych może zarówno wykonywać monitorowanie aktywne jak i pasywne pewnej części usług lub urządzeń. Wyniki sprawdzeń nie są jednak przetwarzane przez instancję roboczą, lecz są przesyłane z użyciem send_nsca do instancji centralnej, w której następuje odpowiednie przetwarzanie.

Kolejnym z możliwych scenariuszy współpracy wielu instancji rdzenia monitorującego jest wykorzystanie wspólnej bazy danych. Rozwiązanie to wymaga jedynie, aby wszystkie instancje rdzenia miały dostęp do jednej bazy danych. Wszystkie instancje są w pełni niezależne i każda z nich monitoruje w dowolny sposób pewną grupę usług i urządzeń. Wyniki monitorowania są przetwarzane, przez każda instancję niezależnie, a na podstawie ich przetwarzania generowane są odpowiednie zdarzenia. Przy użyciu komponentu IDOUtils wszystkie te dane są konsolidowane

w wspólnej bazie danych z której korzysta interfejs icinga-web. Dzięki wykorzystaniu nowego interfejsu możliwe jest równoczesna prezentacja wyników monitorowania pochodzących od wielu instancji, przy użyciu jednego interfejsu.

Oba rozwiązania posiadają zarówno zalety jak i wady. Rozwiązanie z użyciem dodatku NSCA zapewnia spójne przetwarzanie danych przez jedna instancje i łatwość konfiguracji dodatków wykorzystujących dane eksportowane przez jądro w postaci danych wydajnościowych. Niestety rozwiązanie to generuje znaczące obciążenie instancji centralnej, gdyż musi ona przetwarzać wszystkie wyniki sprawdzeń. Ponadto należy przypomnieć, że model bezpieczeństwa dodatku NSCA posiada poważne wady. Rozwiązanie oparte o wspólną bazę danych posiada rozproszony mechanizm przetwarzania sprawdzeń jak i zdarzeń dzięki czemu nie występuje w nim nadmierne obciążenie jednej z instancji. Ponadto awaria, dowolnej z instancji nie powoduje nigdy braku możliwości monitorowania całej sieci lecz jedynie jej fragmentu. Niestety w rozwiązaniu tym konieczna jest bardziej zaawansowana konfiguracja dodatków korzystających z danych wydajnościowych. Wybór konfiguracji zależy zatem silnie od infrastruktury w jakiej ma być ona zastosowana, a także od pozostałych elementów systemu, jakie będą wykorzystane.

4.6. Problemy z monitorowaniem klienta mobilnego

System Icinga nie posiada żadnego mechanizmu wsparcia dla klientów mobilnych. Istnieje wiele konfiguracji rozproszonych, a część z nich może być zaadoptowana do monitorowania klienta mobilnego. Należy pamiętać, iż element systemu obecny na urządzeniu mobilnym musi oszczędzać zarówno pamięć jak i czas procesora. Konfiguracja rozproszona z wspólną bazą danych w znaczący sposób zwiększyła by obciążenie klienta mobilnego. W związku z powyższym zdecydowano się rozważyć konfigurację rozproszoną z użyciem NSCA. Wymaga ona dostarczenia elementu systemu, który będzie znajdował się na urządzeniu mobilnym i monitorował je, a następnie przekazywał, gdy będzie to możliwe dane do instancji nadrzędnej, która będzie prowadziła analizę otrzymanych danych.

Wykorzystanie do celu komunikacji pomiędzy klientem mobilnym, a instancja nadrzędną dodatku NSCA niesie za sobą wiele problemów. Dodatek NSCA jest powszechnie używany do monitorowania serwerów znajdujących się za zaporą lub w wydzielonym segmencie sieci. 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 3. 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 zastosowaniu 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 czasy. 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 przesyłać 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.

Ze względu na powyższe wady, zastosowanie dodatku NSCA do monitorowania klienta mobilnego jest niemożliwe, gdyż stwarza poważne uchybienia w zakresie bezpieczeństwa. Wykorzystanie konfiguracji z nadrzędną instancją systemu, która będzie przeważała dane pochodzące od klientów mobilnych, zdecydowanie przystaje do charakterystyki monitorowania klienta mobilnego.

5. Architektura proponowanego systemu

5.1. Podział na moduły

Klient mobilny, zdefiniowany w 3 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 infrastruktóry sieciowej. Dodatkowo, należy odnieść się do wymagań, w których zawarta jest konieczność minimalizowania zużycia energii przez klietna mobilnego. Ciągłe utrzymywanie połączenia z serwere, powodowałoby znaczne zużycie energii. Współpraca klienta mobilnego z infrastrukturą publiczną nie pozwala również, na założenie, iż klient mobilny posiada globalny adres IP[1]. 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 przechwycenenia. 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 bierzącyhc 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 3 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 systamu, nakazują umożliwienie komunikacji, pomiędzy modułami poprzez dobrze zdefiniowane interfejsy. Zapewni to wymienność poszczególnych modułów systemu i pozwoli na lepsze dostosowanie całego systemu do oczekiwań konkretnego klienta. Należy zatem zwrócić szczególną uwagę, na zapewnienie bezpiecznego i elastycznego protokołkołu 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.

5.2. Moduł podstawowy

Moduł podstawowy stanowi rdzeń całego systemu monitorowania. Moduł ten został zbudowany wykorzystując system monitorowania Icinga. System monitorujący Icinga został szeroko opisany w 2.1.3. 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 bedzie opis tego modulu

5.3. 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 nastpujących 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 obliczen. 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 nieuprawionych 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 umożliwiał 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 kotrolował otrzymywane dane i przyjmował tylko te, które klient jest uprawiony przesyłać.

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 umożliwiał przekazanie danych do dowolnej aplikacji, w tym w szczególności, do wielu z nich jednocześnie. Przekazywanie dancyh 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 fizicznego dostarczenia do zdefiniowancyh miejsc docelowych. Brak możliwosci 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 wykonowanych u klientów mobilnych. Wskazana jest próba użycia tych narzędzi. Możliwa jest jednak sytuacja, w której narzędza przeznaczone dla klientów statycznych nie będą spełniały wymagań stawianych przed tym modułem odbiorczym. Należy wówczas rozważyć modifikację istniejącego narzędzia lub napisanie nowego od podstaw.

5.4. Moduł mobilny

Moduł ten jest odpowiedzialny za monitorowanie zadanych parametrów urządzenia mobilnego. Ponieważ klienty mobilne są od siebie nie zależne i mogą oprować bez możliwosci komunikacji ze sobą, konieczne jest aby każdy klient mobilny posiada swoją instancję tego modułu, która jest odpowiedzialna za monitorowanie jego urządzenia. Ten elementn systemu musi posiadać budową modularną. Najważniejsze z wymagań odnoszocych 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 równiez wspomnieć, iż moduł mobilny odpowiedzialny jest za nadawianie każdemu z odczytów stempla czasu uniwersalnego[2] 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, midzy 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. Nalezy 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].

6. Architektura modułu odbioru danych

6.1. Analiza

Na rynku brak jest rozwiazań 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 3, 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 dostep 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 bufory, 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 doceloweg 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 oni zawierać implementacje protokołów komunikacyjnych oraz wszystkiego co z tym związane. Konsumenci danych natomist 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 równiez 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 kryptografi 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 zaistaniałym 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 elemetów programu. Udostępniają one generyczne interfejsy, które są następnie w nich implementowane. Utworzenie obiektów pochodzących z tych modułów uzależnione jest od bierzącej konifguracji porgramu, a ich zawartość moze być bardzo dynamiczna. W celu ograniczenia wpływu rozrostu liczby klas w tych modułach wykorzystano wzorzec fabryki. Każdy z omawiancy 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 uzywać go poprzez standardowy, zdefiniowany wcześniej interface. Należy wspomnieć, iż poszczególne klasy fabryk wykorzystują wzorzec projektowy nazywany signleton. Oznacza to iż istnieje w programie tylko i wyłącznie jedna instancja danej fabryki, co umożliwa zachowanie spójności i dostępności danych w całym systemie. Zastosowanie wzorca projektowego fabryki pozwala pozostałym obiektom, na korzystanie z obiektów, których typ faktyczny jest nieznany 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 konfuguracji programu wykorzystano zewnętrzny plik w formacie XML. Umożliwaia to zmianę konfiguracji programu bez konieczności jego ponownej komilacji. Plik konfuguracyjny 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 nalezy 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 odbiekty dostawców oraz odbiorców danych są dostarczane z zewnętrz prawidłowość ich ustawień nie moze 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 zarządce programu oraz obiekty pomocnicze. Wykonywane są tutaj wszelkie czynności, które nalezy 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 bufory przeznaczone na dane oczekujace 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, Kałuszka 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.