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

2. Dostępne systemy monitorujące

2.1. Podział systemów monitorujących

2.1.1. Systemy aktywne

2.1.2. Systemy pasywne

2.2. Przegląd systemów dostępnych na rynku

2.2.1. System monitorowania Nagios

2.2.2. System monitorowania Icinga

3. Monitorowanie klienta mobilnego jako monitorowanie rozproszone

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 i posiadają dobrze zdefiniowaną hierarchię. Wzajemne relacje pomiędzy tymi urządzeniami wynikają w dużej mierze z struktury sieci lecz mogą również wynikać z innych zależności. 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 instancja jądra monitorującego, do którego przesyłane są wyniki sprawdzeń poszczególnych usług. Każdy serwer sam 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 monitorowania klienta stacjonarnego, nie ma to większego dla wykrywania awarii, jednak wprowadza to luki do wykresów analizujących dane historyczne. Wieleinstancyjny 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 obciążenie, a także zwiększa bezpieczeństwo. Warto również wspomnieć iż istnieje dodatek do systemów z rodziny Nagios o nazwie Nagios Fusion, który umożliwia integrację wielu instancji jądra sprawdzającego bez konieczności tworzenia centralnej jego instancji i przesyłania wyników wszystkich sprawdzeń. Do przesyłania wyników sprawdzeń dokonywanych czy to pasywnie na serwerach czy też aktywnie przez różne instacje jądra sprawdzającego służy dodatek NSCA opisany w 4.

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 zdać sobie sprawę, że urządzenia te są często narażone na dostęp do sieci, o bardzo niskim poziomie zaufania i wielu zagrożeniach. Oznacze to w szczególności, iż urządzenie mobilne zazwyczaj posiada zmienny adres IP, który rzadko jest adresem globalnym. Również struktura sieci, z której korzysta klienty mobilne jest dynamiczna i znajduje się poza obszarem monitorowania administratorów danego przedsiębiorstwa. Znacząca większość klientów mobilnych dzięki kontaktom z sieciom 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 grupa liczność grupy klientów wynosi zazwyczaj 1. Powoduje to, że w przeciwieństwie do klientów statycznych gdzie grup klientów 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 nich 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 skierowana na pomiar usług świadczonych przez dany system dla innych systemów. Natomiast w przypadku klientów mobilnych znacznie większy nacisk jest położony na monitorowanie parametrów wewnętrznych danego klienta.

3.3. Wymagania systemu monitorowania klientów mobilnych

Odmienna charakterystyka klienta mobilnego powoduje, że wymagania stawiane przed systemem monitorowania takich urządzeń są rózne od tych, które stawiano projektując systemy do monitoringu klienta. Wymagania, jakie zdaniem autora, powinno sie postawić zebrano w 3.1.

Na podstawie 3.1 jasno można stwierdzić, iż dostępne na rynku systemy nie spełniają wymagań stawianych przed systemem monitorowania klienta mobilnego. Obecnie brak jest na rynku spejalistycznego systemu przeznaczonego do monitorowania klientów mobilny. Ponadto istnienie takiego systemu byłoby nieuzasadnione, gdyż oddzielenie zarządzania klientami statycznymi i mobilnymi niosłoby za sobą dodatkowe koszty jak i komplikacje całego systemu. Łatwo można również zauważyć iż współczesne systemy przeznaczone dla klientów statycznych posiadają większość komponentów serwerowych niezbędnych do monitorowania jak i analizy danych klienta mobilnego. Niestety niektóre z komponentów tych systemów posiadają poważne uchybenia, które dyskwalifikują ich użycie przy monitorowaniu klienta mobilnego. Głównym uchybeniem wydaje się być kwestia bezpieczeństwa transportu wpisów dziennika oraz brak implementacji minimalnych wersji jądra sprawdzającego na platformy mobilne takie jak Androdid czy Windows Phone. Dokładny opis dodadku NSCA służącego do przekazywania wpisów dziennika w systemach z rodziny Nagios oraz opis jego uchybień w kwestii bezpieczeństwa zawarto w 4.

W rozdziale 6 niniejszej pracy przedstawiono nowy moduł systemu, kótry można zintegrować z systemami z rodziny Nagios. Natomiast rozdział 7 zawiera opis protokołu zdefiniowanego na potrzeby bezpiecznego przekazywania wpisów dziennika od klienta mobilnego. Moduł systemu odpowiedzialny, za generację wpisów dziennika jest poza zakresem tej pracy. Dla platformy Android taki moduł został opisany w. ...

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

4. Monitorowanie rozproszone z użyciem NSCA

4.1. Opis dodatku NSCA

NSCA - Nagios Service Check Acceptor jest to dodatek do systemów monitorujących opartych o system Nagios pozwalający 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 GPL. 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 wspominany 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 umieszczone jest jądro odpowiedzialne za przetwarzanie wyników sprawdzeń,
— moduł odbierający (nsca) służący do odbierania wyników sprawdzeń od klientów i dostarczaniu ich do jądra, które zajmuje się dalszym ich przetwarzaniem.

4.1.1. Moduł wysyłający

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

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

Po otrzymaniu pakietu inicjującego moduł rozpoczyna czytanie wpisów z standardowego wejścia. Wszystkie wpisy dziennika muszą być odpowiednio sformatowane. Poszczególne pola informacyjne muszą być rozdzielone pojedyńczą tablulacją, 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.

Natomast wpisy dotyczące konkretnej usługi tego urządzenia powinny zawierać następujące pola:

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

Łatwo zauważyć, że żadne z pól wpisu w dzienniku nie zawera stempla czasu wymaganego przez jądro sprwdzające 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 stepel czasu, który trafia do jadra sprawdzającego nadawany jest natomiast przez moduł odbierający.

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

4.1.2. Moduł odbierający

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

— samodzielny demon jednoprocesowy: 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 konkrentym 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 wykorzysanie tego mechanizmu konieczne jest zapewnienie demonowi dostępu do pliku zewnętrznych komend systemu monitorującego. Ponieważ plik zewnętrznych komend jest potokiem nazwanym, chroniony jest on przez Uniksowy system uprawnień użytkowników. Zapewnienie dostępu do takiego bytu może się odbyć na dwa sposoby. Pierwszym, polecanym przez twórców systemów monitorujących, jest uruchamianie demona NSCA jako procesu tego samego użytkownika co proces jądra systemu monitorującego. Drugim sposobem jest modyfikacja praw dostępu do omawianego

pliku, tak aby umożliwić dostęp użytkownikowi z którego uprawnieniami uruchomiony jest demon NSCA. Przy zastosowaniu drugiego rozwiązania zalecana jest szczególna ostrożność, gdyż dostęp do pliku zewnętrznych komend daje bardzo duże możliwości ingerencji w system monitorujący.

Komunikacja modułu odbirającego z klientem rozpoczyna się od nadejścia połączenia od klienta. Gdy moduł odbierający otrzyma nowe połączenie zostanie wysłany pakiet inicjalizujący, którego zawartość została opisana w 4.1.1. Po przesłaniu pakietu inicjalizującego połączenie, moduł odbierający oczekuje na dane od klienta. Każdy wpis dziennika przesyłany jest przy użyciu pakietu o poniższych polach:

— wersja protokołu: aktualnie używana wersja protokołu komunikacyjnego,
— kod CRC32: kod CRC32 bieżącego pakietu,
— stempel czasu: stempel czasu pochodzący z pakietu inicjalizującego przesłanego klientowi,
— kod statusu: kod stanu usługi/hosta powiązany z przesyłanym wpisem
— nazwa hosta: nazwa klienta, który podlegał sprawdzeniu. Nie jest konieczne aby był to ten sam klient, który dostarcza dane,
— opis usługi: nazwa usługi, która podlegała sprawdzeniu lub posty napis jeśli sprawdzenie dotyczy hosta,
— wynik sprawdzenia: napis wygenerowany przez wtyczkę, która dokonywała sprawdzenie, zawierajacy dodatkowe dane natemat stanu hosta/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 sie 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 bierzącego stempla czasu z tym pochodzącym z odebranego pakietu. Jeśli różnica pomiędzy nini jest zbyt duża, dane zostają odrzucane. Ostatnią czynnością wykonywaną przez moduł odbierajacy jest zapisanie odebranego wpisu do pliku zewnętrznych komend jądra systemu monitorującego.

Warto wspomnieć, że stempel czasu przesłany przez klienta nie jest dostarczany do jądra monitorujacego. Służy on jedynie określeniu odstępu czasu od inicjalizacji sesji do chwili otrzymania wiadomości i podjęciu decyzji o otrzymaniu, bądź odrzuceniu pakietu. Do systemu monitorującego trafia natomiast bierzący stempel czasu lokalnego serwera, na którym uruchomiony jest moduł odbierajacy i jądro systemu monitorujacego. Istotną, może się również okazać informacja, iż protokół komunikacyjny nie przewiduje przesyłania ACK[1], bądź też NACK[2]. Oznacza to, iż moduł wysyłający nie ma żadnej gwarancji ani informacji, że dane przesłane do modułu odbierającego zostaną dostarczone do jądra systemu monitorującego.

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

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

4.2. Bezpieczeństwo

Bezpieczeństwo monitorowania z użyciem dodatku NSCA opera 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, którym w przypadku tego protokołu jest przesyłany przez serwer w pakiecie inicjującym.

Wszystkie algorytmy symetryczne do prawidłowego działania wymagają, aby komunikujące się strowny współdzieliły pewien sekret jakim jest klucz używany do szyfrowania. Ujawnienie klucza symetrycznego wiąże się z kompromitacją całego systemu. W dodadktu 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 pozytywanie zweryfikowana są odrzucane.

Kryptografia zastosowana 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 kryptogrfii. Prawdopodobieństwo nie znalezienia kolizji po 200 000 prób wynosi poniżej 1%. Oznacza to iż jedynie w niespełna 1% przypadków koniecne będzie obliczenie więcej niż 200 000 kodów CRC przed znalezieniem kolizji. Prawdopodobieństwo nie znalezienia kolizji w zalezności 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 moduły 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 moze przesłać wpisy dziennka, udające wpisy pochodzące od zupełnie innych klientów. W szczególności jeśli atakujący uzyska klucz symetryczny, to nie

Tablica 4.1. Prawdopodopieństwo 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. Atakujący rozpoczyna atak, po czym przechwytuje pakiety z wpisami z dziennika, które mogą świadczyć o rozpoczęciu ataku i w zamian przesyła do serwera fałszywe pakiety informujące iż wszystkie usługi pracują normalnie.

4.3. Problemy z monitorowaniem klienta mobilnego

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

Problem monitorowania klienta mobilnego został szczegółowo opisany 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, który dyskryminują dodatek NSCA w zastosowaniach do monitorowania klienta mbilnego są:

— Bezpieczeństwo: mechanizmy bezpieczeństwa zawarte w protokole wymiany dancyh posiadają bardzo powazne luki. Zastosowanie CRC32 do sprawdzania spójności dancyh niesie za sobą bardzo duże ryzyko. Ponadto konieczność przechowywania na urządzeniu klucza symetrycznego, który kompromituje cały system znacząco osłaba 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 hosta i dowolnej usługi. Brak jest mechanizmu, który pozwolił by na kontrolę tego, jaki klient ma prawo informować o jakim hoście 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 iż wysłane przez klienta dane zostały przetworzone przez dodatek NSCA.
— 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 z systemem 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 dancyh do kilku systemów oraz definiowania reguł, które dane gdzie powinny trafić.

Powyższe wady zdecydowanie dyskryminują dodatek NSCA jako narzędzie do monitoringu klienta mobilnego. W związku z powyższym w tej pracy zaproponowano nowy protokół komunikacyjny, który został opisany w 7 oraz cały rzykładowy system do monitorowania zarówno klientów stacjonarnych jak i mobilnych opisany w 5.

5. Architektura proponowanego systemu

5.1. Podział na moduły

Monitorowanie klienta mobilnego jest zagadnieniem złożonym. Niestety nie jest możliwe zaadoptowanie bez modyfikacji żadnego z dostępnych na rynku systemów. Wymagania przedstawione w 3 wymuszają budowę systemu monitorowania, który będzie umożliwiał monitorowanie zarówno klienta mobilnego jak i statycznego. Napisanie całości takiego systemu jest zadaniem bardzo obszernym. Warto tutaj nadmienić, że jądro monitorujące systemu Icinga zostało napisane w języku C i zajmuje 315 000 linii kodu. Na tej podstawie uznano za niemożliwe napisanie systemu o funkcjonalności szerszej od wspominanego w ramach pracy inżynierskiej.

Kolejnym argumentem przeciw tworzeniu takiego systemu od podstaw jest konieczność przeprowadzenia obszernych testów. Większość systemów dostępnych na rynku posiada rozbudowany system testów autoamtycznych, a także szeroką społeczność, co powoduje, że kod w nich zawarty jest dobrze przetestowany, a jego jakoś jest bardzo wysoka.

W obec powyższych, bezsprzecznie słusznych argumentów, w ramach tej pracy inżynierskiej zaproponowano system monitorujący oparty o rozwiązania dostępne na rynku, które zostały zmodyfikowane, aby umożliwić monitoring klienta mobilnego. Proponowany system składa się z trzech modułów pełniących następujące funkcje:

— 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.

Elementy modułu podstawowego rozmieszczone są na serwerach sieci lokalnej, w której znajdują się klienty statyczne. Moduł ten zapewnia ich monitorowanie oraz stanowi interfejs dostępowy dla administratora. W przedstawionej kofiguracji istnieje tylko jedna instancja jądra monitorującego, lecz możliwa jest również konfiguracją zawierająca kilka rdzeni systemu monitorowania. Sposób wykonania takiej kofiguracji został omowiony w [wiki_icingi]. Moduł odbioru danych umieszczony jest na urządzeniu posiadającym dostęp do sieci, w której pracują klienty mobilne. W omawianej kofiguracji przyjęto, iż klienty mobilne oraz moduł odbioru danych mają dostęp do sieci Internet. Ponadto w celu uproszczenia omawianej kofiguracji i pominięcia dodatkowych elementów, które nie są przedmiotem tej pracy, poczyniono założenie, iż moduł odbierający dane oraz rdzeń monitorujący modułu podstawowego znajdują się na tym samym systemie operacyjnym. Możliwa jest jednak

konfiguracja, w której wspomniane elementy znajdują się na różnych urządzeniach. Moduł mobilny instalowany jest na urządzeniu mobilnym, które jest monitorowane przez system. Jest on bezpośrednio odpowiedzialny za wykonywanie zaplanowanych odczytów oraz przekazanie ich wyników do modułu odbierającego. Poszczególne elementy zostały omówione w kolejnych podrozdziałach. Moduły omawianego systemu są od siebie niezależne i komunikują się poprzez dobrze zdefiniowane interfejsy i protokoły. Możliwa jest zatem zmiana przedstawionej konfiguracji, w taki sposób, aby spełniała ona dodatkowe oczekiwania odbiorcy.

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 2.2.2. Cieszy się on uznaniem środowiska administratorów, a jego możliwości konfiguracji umożliwiają budowę rozległego systemu monitorowania rozproszonego zarówno dla bardzo dużej sieci, jak i dla dużej liczby klientów mobilnych.

Tutaj 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. Żaden z dostępnych na rynku systemów monitorowania, ani dodatek do takiego systemu, nie spełniał wymagań stawianych przed omawianym systemem. W związku z powyższym moduł odbioru doanych został zaprojektowany oraz zaimplementowany w ramach niniejszej pracy. Omawiany moduł jest niezależny od pozostałych. Możliwe jest jego użycie zarówno z innymi klientami mobilnymi, jaki z innym systemem monitorującym. Należy jedynie zapewnić implementacje protokołu komunikacyjnego po stronie klienta, oraz dostarczyć metodę, wprowadzania danych do modułu podstawowego. Omawiany moduł przeznaczony jest dla systemów z rodziny Linux. Składa się on z samodzielnego demona systemowego, którego szczegółowa architektura została omówiona w 6.

Moduł ten spełnia kilka bardzo ważnych funkcji. Podstawowym jego zadaniem jest odbieranie danych od klienta. Zgodnie z wymaganiami określonymi w 3 konieczne jest, aby system zapewniał zarówno poufność jak i integralność danych. Oba te wymagania są spełnione przez ten system, poprzez użycie kryptografi oraz funkcji skrótu, które pozwalają jednoznacznie zweryfikować integralność wiadomości. System jest niezależny od zastosowanego algorytmu szyfrowania danych. W omawianej konfiguracji, do transportu danych został wykorzystany algorytm AES pracujący w trybie wiązania bloków zaszyfrowanych (CBC). Jako długość klucza przyjęto 128 bitów, pomimo, iż jest to najmniejsza z dostępnych długosci klucza AES jest ona wystarczająca do tego zastosowania.

Kolejnym zadaniem omawianego modułu jest uwierzytelnienie klienta. Dla celów demonstracyjnych dostarczono dwa moduły uwierzytelnienia. Pierwszy z nich to uwierzytelnienie zawsze pozytywne, które pozwala na dostęp każdemu klientowi. Drugi natomiast, to proste uwierzytelnienie na podstawie nazwy użytkownika oraz

hasła przydzielonego przez administratora[1]. Możliwe jest skonfigurowanie dowolnej motody uwierzytelnienia klienta, zgodnie z polityką bezpieczeństwa stosowaną w danej sieci.

Odebrane od klienta dane są przechowywane przez moduł z wykorzystaniem pamięci trwałej, co umożliwa bardzo szybkie potwierdzenie odebrania danych, jeszcze przed przekazaniem ich do miejsc przeznaczenia. Omawiany moduł, na podstawie pliku konfiguracyjnego dostarcza dane do wskazanych miejsc docelowych. Możliwe jest definiowanie dowolnych miejsc przeznaczenia dla danych, co umożliwia przekazywanie danych pochodzących od klienta do wielu systemów bez konieczności ich retransmisji. Zastosowany plik pośredni gwarantuje, iż czas zapisu danych w miejsce docelowe, nie będzie wydłużał czasu oczekiwania na potwierdzenie przyjęcia danych od klienta. Warto zauważyć również, że dane odebrane przez ten moduł zostaną zawsze dostarczone do ich miejsc docelowych, lub jawnie usuniętę przez administratora. Oznacza to zatem, iż moduł gwarantuje dostarczenie danych, natomiast wykrywanie duplikatów danych jest wykonywane w miejsca docelowych dla danych.

5.4. Moduł mobilny

Moduł ten jest odpowiedzialny za monitorowanie zadanych parametrów urządzenia mobilnego. Każdy klient mobilny posiada swoją instancję tego modułu, która jest odpowiedzialna za monitorowanie jego urządzenia. Ten elemento systemu powinien posiadać budową modularną. Najważniejsze z wymagań odnoszocych się do tego modułu wymusza, aby możliwe było w jak najprostszy sposób dodawanie możliwości sprawdzania nowych parametrów.

Ponad to implementacja tego modułu musi brać pod uwagę architekturę sprzętową na której pracuje. 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 dokonywanego pomiaru. Na podstawie dokonanej charakterystyki klienta mobilnego, w niniejszej pracy poczyniono 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.

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 jest możliwe dostarczenie uniwersalnej implementacji protokołu komunikacyjnego dla wszystkich platform mobilnych. Oznacza to, iż przed klientem mobilnym stawia się wymóg zarówno implementacji protokołu

[1] Przedstawione moduły mają charakter czysto akdemicki i służą jedynie zaprezentowaniu niezależności modułu odbierania danych od wybranej metody uwierzytelnienia użytkownika. Wszystkie nazwy użytkowników oraz hasła, używane przez jedną z metod przechowywane są jawnym tekstem. Przed użyciem systemu należy skonfigurować metodę uwierzytelnienia urzytkownika zgodnie z polityką stosowaną w danej sieci.

komunikacyjnego jak i odpowiednich mechanizmów uwierzytelnienia klienta. Protokół komunikacyjny został szczegółowo opisany w 7.

W ramach omawianego systemu wykorzystano wiele instancji modułu mobilnego zaprojektowanego i zaimplementowanego przez Pana Marcina Kubika. Moduł ten został szeroko omówiony w [praca_kubika]. Całość systemu jest niezależna od platformy klienta mobilnego, zatem mozliwa jest współpraca całości systemu z klientami mobilnymi przeznaczonymi dla innych platform, jednak muszą one implementować protokół komunikacyjny wykorzystywany przez moduł pośredniczący.

6. Architektura modułu odbioru danych

6.1. Podział na moduły

6.2. Szkielet programu

6.3. Moduł kryptograficzny

6.4. Moduł autoryzacji klienta

6.5. Moduł komunikacji z wykorzystaniem TCP

6.6. Moduł pisarza potoku

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.