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 . . . 1

2. Dostępne systemy monitorujące . . . 2

2.1. Podział systemów monitorujących . . . 2

2.1.1. Systemy aktywne . . . 2

2.1.2. Systemy pasywne . . . 2

2.2. Przegląd systemów dostępnych na rynku . . . 2

2.2.1. System monitorowania Nagios . . . 2

2.2.2. System monitorowania Icinga . . . 2

3. Monitorowanie klienta mobilnego jako monitorowanie rozproszone . . . 3

3.1. Charakterystyka problemu . . . 3

3.2. Wymagania funkcjonalne . . . 3

3.3. Wymagania niefunkcjonalne . . . 3

4. Monitorowanie rozproszone z użyciem NSCA . . . 4

4.1. Opis dodatku NSCA . . . 4

4.1.1. Moduł wysyłający . . . 4

4.1.2. Moduł odbierający . . . 5

4.2. Bezpieczeństwo . . . 7

4.3. Problemy z monitorowaniem klienta mobilnego . . . 8

5. Architektura proponowanego systemu . . . 10

5.1. Podział na moduły . . . 10

5.2. Moduł podstawowy . . . 10

5.3. Moduł odbioru danych . . . 10

5.4. Moduł mobilny . . . 10

6. Architektura modułu odbioru danych . . . 11

6.1. Podział na moduły . . . 11

6.2. Szkielet programu . . . 11

6.3. Moduł kryptograficzny . . . 11

6.4. Moduł autoryzacji klienta . . . 11

6.5. Moduł komunikacji z wykorzystaniem TCP . . . 11

6.6. Moduł pisarza potoku . . . 11

6.7. Moduł logowania . . . 11

7. Protokół komunikacyjny . . . 12

7.1. Podział na warstwy . . . 12

7.2. Warstwa formowania wiadomości . . . 12

7.3. Warstwa kryptograficzna . . . 12

7.4. Warstwa integralności danych . . . 12

7.5. Warstwa transportu logów . . . 12

8. Testowanie i użytkowanie wykonanego systemu . . . 13

8.1. Testowanie . . . 13

8.2. Użytkowanie systemu . . . 13

9. Podsumowanie . . . 14

Bibliografia . . . 15

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. Charakterystyka opoblemu

3.2. Wymagania funkcjonalne

3.3. Wymagania niefunkcjonalne

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

5.2. Moduł podstawowy

5.3. Moduł odbioru danych

5.4. Moduł mobilny

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.