STUDIA INFORMATICA 2000
Volume 21 Number 1 (39)

Daniel KORZEKWA
PWPT WASKO Sp. z o.o., Zakład Systemów Telekomunikacyjnych

ZASTOSOWANIE INTELIGENTNYCH AGENTÓW JAKO ALTERNATYWA DLA STANDARDU ZARZĄDZANIA SNMP

Streszczenie. W opracowaniu przedstawiono model zarządzania systemami komputerowymi w oparciu o protokół RPC. Opisano architekturę inteligentnego agenta oraz jego główne funkcje i możliwości. Omówiono procesy komunikacji z zarządcą oraz mechanizmy bezpieczeństwa agentów w środowisku rozproszonym.

TAKING ADVANTAGE OF INTELLIGENT AGENT AS AN ALTERNATIVE FOR THE SNMP MANAGEMENT STANDARD

Summary. This article describes computer systems management based upon the protocol RPC. It presents architecture of the intelligent agent as well as its main management functions and possibilities. The communication processes between the manager and the agent and the security mechanisms in the distributed environment are also described in the article.

1. Wprowadzenie

W rozbudowanych sieciach heterogenicznych istotną rolę pełnią systemy zarządzania, które zapewniają kontrolę nad pracą całej sieci. Najpopularniejszy obecnie standard zarządzania scentralizowanego korzystający z protokołu SNMP dobrze spełnia swoją rolę w sieciach lokalnych i nieskomplikowanych sieciach rozległych. W przypadku sieci składających się z wielu elementów aktywnych taki system zarządzania ma wiele wad, wśród których najpoważniejsze to: znaczne obciążenie sieci oraz prymitywny system bezpieczeństwa [10]. Co prawda w wersji SNMPv2 wprowadzono udogodnienie M2M (ang. manager to manager) pozwalające na proste zarządzanie rozproszone, jednak dalej weryfikacja klienta odbywała

weryfikacja klienta odbywała się jedynie na podstawnie hasła (ang. community name) [11]. Zawansowane mechanizmy bezpieczeństwa pojawiły się dopiero w wersji SNMPv3 [12]. Alternatywę dla SNMP stanowi system zarządzania oparty na rozproszonym środowisku obliczeniowym DCE RPC i inteligentnych agentach AI (ang. intelligent agent).

2. Opis inteligentnego agenta

Inteligentny agent AI został zaprojektowany do zarządzania serwerami wyposażonymi w systemy operacyjne wysokiego poziomu (np. Unix, Windows NT, OS/2) [4], które stanowią platformę dla rozbudowanych aplikacji (np. bazy danych, systemy finansowo-księgowe). Zarządzanie takimi zasobami wymaga zastosowania zaawansowanych funkcji zarządzających, które nie ograniczają się jedynie do prostego monitorowania parametrów systemowych.

Agent AI zajmuje się ciągłym monitorowaniem i zarządzaniem zasobów systemowych nie obciążając ich jednocześnie. Wystąpienie na stacji zarządzanej sytuacji zdefiniowanej w konfiguracji agenta powoduje wysłanie do serwera zarządzającego odpowiedniego komunikatu opisującego zdarzenie. Inteligencja agenta polega na umiejętności samodzielnego wykonania automatycznej akcji lub zlecenia jej operatorowi. Bardzo ważną funkcją agenta AI jest możliwość wykonywania zdalnych operacji przez serwer zarządzający na stacji zarządzanej. Jest to możliwe dzięki zastosowaniu protokołu zdalnych wywołań procedur RPC.

System umożliwiający zaawansowane aktywne zarządzanie musi zapewniać wysoki poziom bezpieczeństwa. W tym celu komunikacja agenta AI z serwerem zarządzającym została oparta na rozproszonym środowisku obliczeniowym DCE (ang. Distributed Computing Environment). Środowisko DCE udostępnia takie usługi w zakresie bezpieczeństwa, jak szyfrowanie przesyłanych danych oraz weryfikację i autoryzację klienta inicjującego połączenie z serwerem.

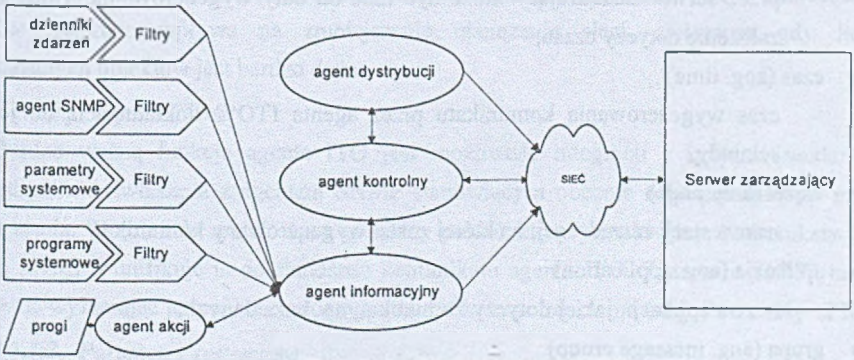
Zastosowanie agenta AI nie ogranicza się jedynie do zarządzania rozbudowanymi serwerami. Agent AI umożliwia także zaawansowane zarządzanie innymi aktywnymi elementami sieciowymi, jak rutery i przełączniki, które wyposażone są jedynie w agenta SNMP, lub urządzeniami nie wyposażonymi w funkcje zarządzania. W takich przypadkach można zastosować agenta proxy [3], który jest urządzeniem wyposażonym w agenta AI, pełniącym funkcję mediatora pomiędzy serwerem zarządzającym a zarządzanym obiektem.

3. Architektura agenta AI

Istnieje wiele rozwiązań firmowych pozwalających na zarządzanie z zastosowaniem inteligentnych agentów. Najpopularniejszymi obecnie produktami oferującymi takie rozwiązania są IT/Operations firmy Hewlett Packard oraz Tivoli firmy IBM.

W celu przedstawienia architektury agenta AI za przykład posłużył agent ITO opracowany przez firmę Hewlett Packard. Istotną zaletą agenta ITO jest przede wszystkim możliwość dowolnej konfiguracji i rozbudowy, co nie ogranicza zastosowań agenta do zaprogramowanych przez producenta funkcji, lecz daje szerokie możliwości adaptacji do własnych potrzeb.

Agent ITO zbudowany jest na zasadzie modułowego drzewa hierarchicznego (rys. 1) [2]. Głównym modułem pełniącym kontrolę nad pozostałymi modułami jest agent kontrolny (ang. control agent). Jest to jedyny proces umożliwiający dwustronną komunikację z serwerem zarządzającym (serwer ITO), co pozwala na interaktywne zarządzanie zasobami systemowymi. Obsługą komunikatów zajmuje się agent informacyjny (ang. message agent), który sprawuje kontrolę nad podagentami zajmującymi się zbieraniem informacji o monitorowanych zasobach. Wygenerowane przez podagentów komunikaty są filtrowane, a następnie zostają wysłane do serwera zarządzającego. Do każdego komunikatu może być dołączona akcja automatyczna lub inicjowana przez operatora. Funkcja ta jest obsługiwana przez agenta akcji (ang. action agent). Za konfigurację agenta ITO odpowiedzialny jest agent dystrybucji (ang. distribution agent), którego zadaniem jest aktualizacja konfiguracji dla wszystkich podagentów agenta ITO.



Rys. 1. Architektura agenta ITO
Fig. 1. Architecture of agent ITO

3.1. Agent kontrolny

Jest to główny proces agenta ITO kontrolujący pozostałe moduły. Pozwala na zdalne uruchamianie lub zatrzymywanie pozostałych podagentów. Pełni funkcję interfejsu dwukierunkowego pomiędzy klientem a serwerem ITO, co pozwala na aktywne zarządzanie zasobami stacji zarządzanej ze stacji zarządzającej.

3.2. Agent informacyjny

Agent informacyjny odbiera komunikaty od podagentów, a następnie przesyła je do serwera ITO. W przypadku braku komunikacji pomiędzy serwerem zarządzającym a agentem komunikaty są buforowane w oczekiwaniu na nawiązanie połączenia. Do każdego komunikatu może być dołączony rozkaz wykonania akcji automatycznej lub inicjowanej przez operatora. Rozkaz jest przesyłany do agenta akcji, który odpowiada za jej wykonanie.

Komunikat, oprócz opisu zdarzenia, zawiera szereg atrybutów pozwalających dokładnie określić jego źródło [2]. Opis atrybutów został przedstawiony poniżej:

- ważność (ang. severity)
  - definiuje poziom ważności komunikatu, możliwych jest pięć poziomów kolejno od najmniej do najbardziej ważnego : normalny (ang. normal), ostrzegawczy (ang. warning), drugorzędny (ang. minor), pierwszorzędny (ang. major), krytyczny (ang. critical),
- data (ang. date)
  - data wygenerowania komunikatu przez agenta ITO. W przypadku braku połączenia agenta z serwerem i buforowania danych data odebrania komunikatu przez serwer zarządzający może być inna od daty wygenerowania. Takie samo założenie dotyczy czasu,
- czas (ang. time)
  - czas wygenerowania komunikatu przez agenta ITO z dokładnością do jednej sekundy,
- węzeł (ang. node)
  - nazwa stacji zarządzanej, na której został wygenerowany komunikat,
- aplikacja (ang. application)
  - nazwa aplikacji, jakiej dotyczy komunikat, np. baza danych,
- grupa (ang. message group)
  - nazwa grupy, do której komunikat został zakwalifikowany, np. bezpieczeństwo,
- obiekt (ang. object)
  - nazwa obiektu skojarzonego z komunikatem, np. nazwa użytkownika.

Przykładowy komunikat wygląda następująco :

ważność : ostrzegawczy

data: 22/01/2000

czas: 22:45

węzeł: 10.0.10.12

aplikacja: baza danych

grupa: bezpieczeństwo

obiekt: użytkownik1

tekst komunikatu: Użytkownik [obiekt] podał błędne hasło przy próbie dostępu do [aplikacja] na serwerze [węzeł]

3.2.1. Źródła informacji

Źródła informacji można podzielić na cztery grupy: dzienniki zdarzeń, inni agenci zarządzania (np. agent SNMP), monitorowane parametry systemowe (np. obciążenie procesora) oraz programy systemowe, które bezpośrednio mogą wysłać komunikaty do agenta informacyjnego wykorzystując specjalny interfejs komunikacyjny. Za każdy rodzaj informacji odpowiada osobny moduł .

3.2.1.1. Dzienniki zdarzeń

Na dzienniki zdarzeń składają się wszystkie bazy informacji o pracy systemu i aplikacji. Zdarzenia mogą być zapisane w postaci jawnej (np. pliki tekstowe) lub niejawnie (np. informacje zapisane w relacyjnych bazach danych). W drugim przypadku najpierw dane zostaną przekonwertowane do postaci jawnej, a następnie zostaną przesłane do agenta informacyjnego. Zdarzenia mogą być filtrowane przy użyciu standardowych wyrażeń regularnych, w ten sposób można wychwycić jedynie interesujące informacje. Zastosowanie filtrów znacząco wpływa na zmniejszenie obciążenia sieci, zwłaszcza gdy liczba zarządzanych obiektów jest bardzo duża.

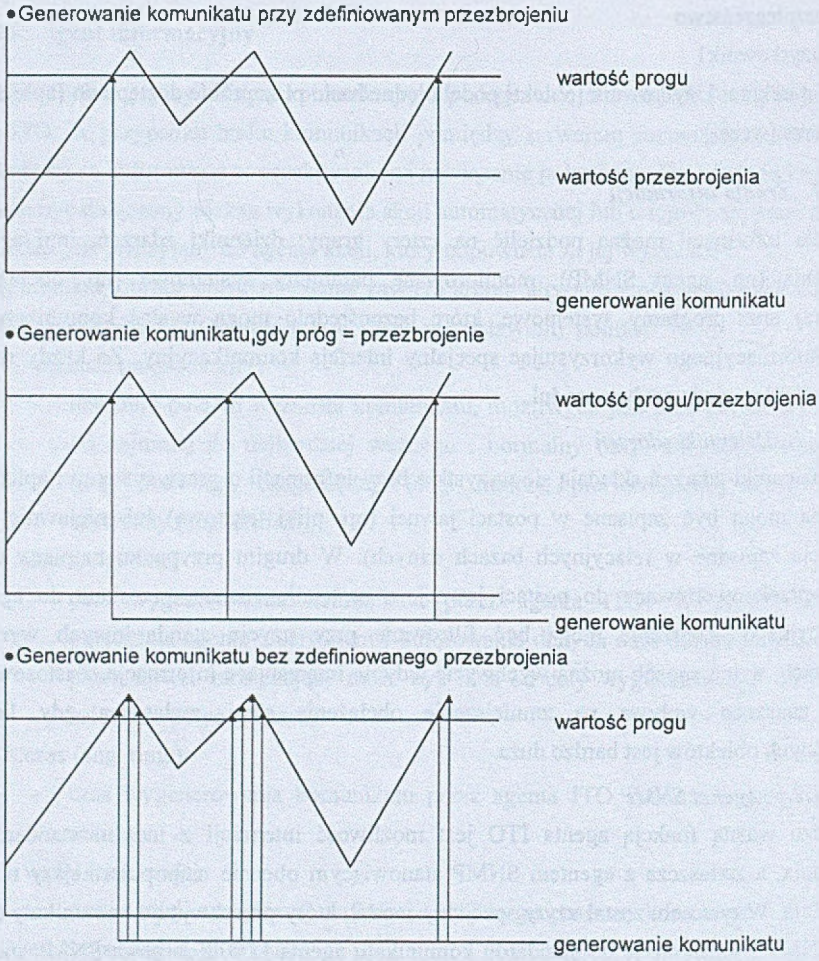
3.2.1.2. Agenci SNMP

Bardzo ważną funkcją agenta ITO jest możliwość integracji z innymi standardami zarządzania, a zwłaszcza z agentem SNMP stanowiącym obecnie najpopularniejszy model zarządzania. W tym celu został użyty specjalny moduł, który przechwytuje komunikaty (ang. traps) SNMP i unifikuje je do standardu komunikatu agenta ITO. Zdarzenia SNMP można filtrować wybierając jedynie te o zadanym identyfikatorze zdarzenia (ang. Event Object ID).

3.2.1.3. Parametry systemowe

Aktualny stan pracy zasobów systemowych (np. dyski, procesory, pamięć) opisywany jest za pomocą pojedynczych parametrów. W momencie przekroczenia zdefiniowanej wartości wysyłany jest specjalny komunikat (np. obciążenie procesora wynosi 95%, a zdefiniowa-

ny próg dla tego parametru ma wartość 90%). Aby uniknąć powielenia tego samego komunikatu w przypadku utrzymywania się wartości parametru powyżej progu, do dyspozycji są rożne warianty „przezbrojeń" (rys. 2) [2]. „Przezbrojenie" definiuje poziom, który musi osiągnąć wartość parametru, aby ponownie po przekroczeniu progu został wygenerowany komunikat.



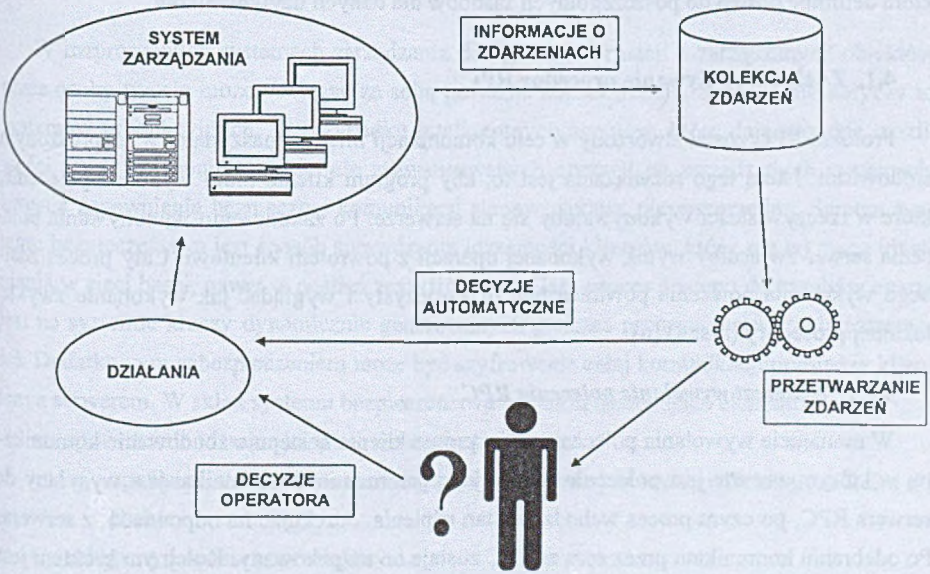
Rys. 2. Monitorowanie progów
Fig. 2. Thresholds monitoring

3.2.1.4. Programy systemowe

Do komunikacji pomiędzy programami systemowymi (np. program monitorujący pracę macierzy dyskowej) a agentem informacyjnym służy specjalny interfejs komunikacyjny. Interfejs zbudowany jest na zasadzie potoku FIFO (ang. pipe First In First Out). Informacje zapisane przez program systemowy do potoku FIFO następnie są odczytywane przez agenta informacyjnego w kolejności od najstarszego do najmłodszego [6]. Potoki są podstawową metodą komunikacji międzyprocesowej. Zapewniają prostą integrację agenta ITO z programami systemowymi.

3.3. Agent akcji

Komunikaty przesyłane do agenta informacyjnego mogą być skojarzone z rozkazami wykonania akcji. Rozkaz może być definitywny, to znaczy zostanie przesłany bezpośrednio do agenta akcji, lub też pasywny - wysłany do serwera zarządzającego, gdzie operator podejmie decyzję, czy wykonać akcję (rys. 3).



Rys. 3. Obsługa komunikatów
Fig. 3. Message service

4. Komunikacja agenta z serwerem

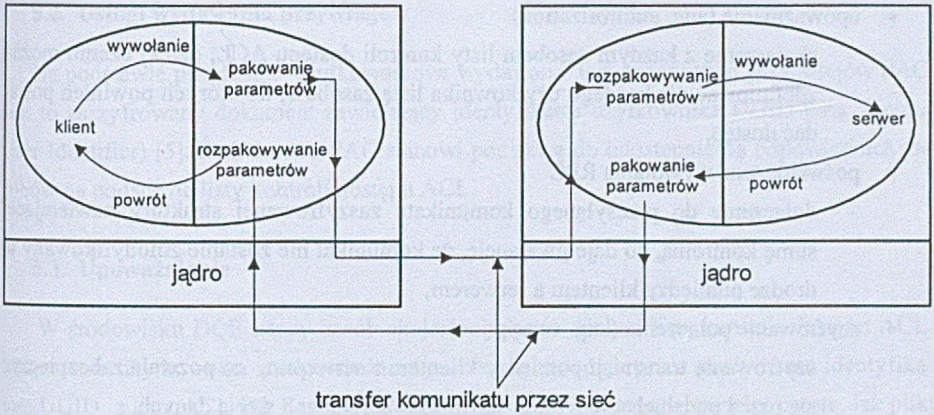
Komunikację pomiędzy agentem a serwerem zapewnia rozproszone środowisko obliczeniowe DCE RPC [2]. System DCE został zbudowany na podbudowie unixowych systemów operacyjnych, a w późniejszym okresie zaimplementowano go w systemach WINDOWS, VMS, OS/2 i innych. System rozproszony zbudowany na bazie środowiska DCE może być heterogeniczny, złożony z komputerów pochodzących od różnych dostawców [5]. System DCE jest niezależny od architektury sprzętowej. Automatycznie dokonuje konwersji różnych typów danych, a komunikacja pomiędzy maszynami nie zależy od używanych protokołów sieciowych. Komunikacja w środowisku DCE odbywa się za pomocą protokołu zdalnych wywołań procedur RPC [5]. Proces ten opiera się na wywoływaniu poleceń przez klienta na maszynie serwera, zaś wynik polecenia wykonanego na serwerze zwracany jest klientowi. W celu zapewnienia jak najlepszego bezpieczeństwa został zastosowany system potwierdzenia tożsamości oparty na systemie KERBEROS. Zaimplementowano także autoryzację polegającą na skojarzeniu z każdym zasobem listy kontroli dostępu ACL (ang. access control list), która definiuje dostęp do poszczególnych zasobów dla danych użytkowników.

4.1. Zdalne wywoływanie procedur RPC

Protokół RPC został stworzony w celu komunikacji między maszynami w rozproszonym środowisku. Istotą tego rozwiązania jest to, aby program klienta mógł wykonać polecenie, które w rzeczywistości wykonywałoby się na serwerze. Po zakończeniu wykonywania polecenia serwer zwracałby wynik wykonanej operacji z powrotem klientowi. Cały proces zdalnego wykonania polecenia powinien być przezroczysty i wyglądać jak wykonanie zwykłej lokalnej procedury (rys. 4) [6].

4.1.1. Schemat wywołanie polecenia RPC

W momencie wywołania polecenia przez proces klienta następuje zbudowanie komunikatu, w którym zawarte jest polecenie wraz z jego parametrami. Komunikat jest wysyłany do serwera RPC, po czym proces wchodzi w stan uśpienia, oczekując na odpowiedź z serwera. Po odebraniu komunikatu przez serwer RPC zostaje on rozpakowany. Kolejnym krokiem jest wykonanie polecenia na serwerze i zbudowanie komunikatu, który zawiera wynik polecenia. Gotowy komunikat zostaje wysłany z powrotem do klienta, gdzie następuje rozpakowanie komunikatu i zwrócenie wyniku macierzystemu procesowi.



Rys. 4. Zdalne wywołania procedur RPC
Fig. 4. Remote procedure calls RPC

5. Bezpieczeństwo w środowisku DCE

W rozproszonych systemach zarządzania dostęp do informacji z zarządzanych obiektów przez osoby trzecie może pociągać za sobą poważne konsekwencje. Szczególnie dotyczy to systemów produkcyjnych. W przypadku inteligentnych agentów, które dają szerokie możliwości nie tylko monitorowania, ale zaawansowanych operacji na zarządzanych systemach, kwestia zapewnienia bezpiecznej komunikacji stanowi sprawę pierwszorzędną. Sercem systemu bezpieczeństwa jest sposób sprawdzania tożsamości klientów, który nie wymaga ujawniania w sieci haseł, nawet w postaci zaszyfrowanej. Cały proces dostępu do zasobów oparty jest na systemie kluczy dynamicznie generowanych podczas procesu sprawdzania tożsamości. Dodatkowym zabezpieczeniem może być szyfrowanie całej komunikacji pomiędzy klientem a serwerem. W skład systemu bezpieczeństwa wchodzą następujące elementy:

- sprawdzanie tożsamości (ang. authentication)
  - upewnienie się, czy osoba inicjująca połączenie jest rzeczywiście tą, za którą się podaje,
- usługi wydawania przywilejów
  - po sprawdzeniu tożsamości klienta wydawane są świadectwa przywilejów PAC (ang. Privilege Attribute Certificates). Świadectwa PAC zawierają identyfikator klienta używany przy sprawdzaniu listy kontroli dostępu ACL, w celu przyznania dostępu do określonych zasobów,

- upoważnianie (ang. authrorization)
  - skojarzenie z każdym zasobem listy kontroli dostępu ACL, dzięki czemu można zdefiniować dla każdego użytkownika listę zasobów, do których powinien posiadać dostęp,
- poświadczone wywołania RPC
  - dołączenie do przesyłanego komunikatu zaszyfrowanej struktury zawierającej sumę kontrolną, co daje gwarancję, że komunikat nie zostanie zmodyfikowany w drodze pomiędzy klientem a serwerem,
- szyfrowanie połączenia (ang. encryption)
  - szyfrowanie transmisji pomiędzy klientem a serwerem, co pozwala zabezpieczyć się przed podsłuchaniem (ang. sniffmg) przesyłanych siecią danych.

5.1. Sprawdzanie tożsamości

Jest to pierwszy etap procesu bezpieczniej komunikacji pomiędzy klientem a serwerem. Został zbudowany na bazie systemu Kerberos i polega na potwierdzeniu tożsamości klienta. Usługa zabezpiecza przed udawaniem (ang. spoofing) przez intruza wiarygodnego użytkownika [5].

Na serwerze sprawdzania tożsamości zgromadzone są klucze, z których każdy skojarzony jest z osobnym klientem (użytkownikiem). Każdy użytkownik posiada jedynie prywatny klucz. Komunikaty wychodzące z serwera są kodowane kluczem użytkownika, dzięki czemu tylko użytkownik, do którego wysłany jest komunikat, może je rozkodować. Kiedy operacja potwierdzenia tożsamości zakończy się sukcesem, klient otrzymuje przepustkę (ang. ticket). Jest to zaszyfrowana struktura stanowiąca dowód, że klient jest rzeczywiście tym, za kogo się podaje. Struktura przepustki wygląda następująco :

przepustka = S, {klucz sesji, klient, termin ważności}Ks

gdzie:

S – serwer, dla którego przeznaczona jest przepustka,

klucz sesji – klucz używany do późniejszej komunikacji z klientem,

klient – identyfikacja klienta, z którym skojarzona jest przepustka,

termin ważności – okres,przez jaki przepustka będzie ważna,

Ks – prywatny klucz serwera.

Po poprawnej weryfikacji przepustki przesłanej przez klienta do serwera i zakodowanej kluczem Ks późniejsza komunikacja będzie szyfrowana kluczem sesji. Kolejnym etapem w procesie bezpiecznej komunikacji będzie nadanie klientowi świadectwa przywilejów PAC.

5.2. Usługi wydawania przywilejów

Na podstawie przepustki użytkownikowi wydawane jest świadectwo przywilejów PAC. Jest to zaszyfrowany dokument zawierający identyfikator użytkownika UUID (ang. Unique User Identifier) [5]. Świadectwo PAC stanowi podstawę do udostępnienia odpowiednich zasobów na podstawie listy kontroli dostępu ACL.

5.3. Upoważnianie

W środowisku DCE każdy zasób skojarzony jest z osobną listą kontroli dostępu ACL. Skojarzenie użytkownika z odpowiednimi zasobami następuje przez porównanie identyfikatora UUID z listą ACL [9]. Zasoby podzielone są na dwie kategorie: zasoby proste, jak pliki i rekordy baz danych, oraz zasobniki, czyli katalogi i tablice baz danych. W zasobnikach złożonych przechowywane są zasoby proste.

Lista ACL składa się z siedmiu atrybutów definiujących dostęp do zasobów:

- czytanie, zapisywanie, wykonywanie
  - są to standardowe parametry dostępu systemu UNIX,
- zmiana listy kontroli dostępu
  - pozwolenie na modyfikacje listy kontroli dostępu do zasobu,
- wstawianie i usuwanie z zasobnika
  - pozwolenie na wstawienie lub usunięcie elementów z zasobnika,
- testowanie zasobu
  - pozwolenie użytkownikowi na testowanie zawartości zasobu bez możliwości jego przeglądania, przykładowo można sprawdzić, czy w tablicy znajduje się dany rekord, mimo że do tego celu nie potrzebujemy praw do przeglądania zawartości tablicy.

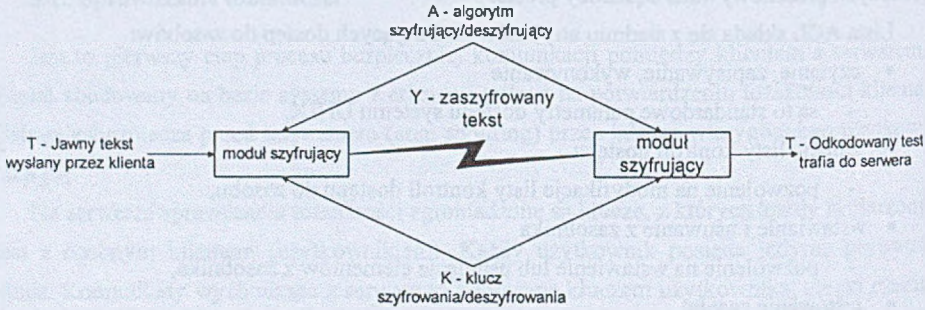
5.4. Poświadczone wywołania RPC

Żaden system zabezpieczeń nie da pełnej kontroli bez zapewnienia, że komunikat wysłany przez klienta w postaci nienaruszonej trafi do serwera. Modyfikacja może nastąpić w sposób przypadkowy, w momencie gdy na drodze pomiędzy klientem a serwerem jest wiele innych węzłów sieciowych, lub, co gorsze, modyfikacji może dokonać intruz. Dopiero na bazie poprawnej komunikacji pomiędzy klientem a serwerem można budować cały system innych zabezpieczeń.

Aby uchronić się przed modyfikacją komunikatu, dołączane jest do niego poświadczenie (ang. authenticator) [5]. Jest to zaszyfrowana struktura zawierająca sumę kontrolną opartą na algorytmie MD5 (ang. Message Digest 5).

5.5. Szyfrowanie połączenia

W systemach, w których oprócz przesyłania informacji służących do wykonywania określonych operacji na maszynie zdalnej, przesyłane są także informacje tajne, zachodzi potrzeba szyfrowania całej transmisji pomiędzy klientem a serwerem (rys. 5). Umożliwi to zabezpieczenie się przed podsłuchaniem (ang. sniffing) transmisji przez intruza. W środowisku DCE do zaszyfrowania przesyłanego komunikatu używany jest algorytm DCE DES parametryzowany kluczem, w którego posiadaniu są zarówno klient, jak i serwer [5].



Rys. 5. Szyfrowanie transmisji pomiędzy serwerem a klientem
Fig. 5. Encrypted transmission between server and klient

6. Podsumowanie

Zarządzanie systemami komputerowymi w oparciu o inteligentnych agentów jest nowoczesną metodą pozwalającą na zaawansowane zarządzanie zasobami systemowymi. Główną zaletą agenta AI jest jego inteligencja umożliwiająca przejęcie części zadań wykonywanych przez serwer zarządzający, dzięki czemu ruch sieciowy generowany przez agenta AI jest minimalny. Proste awarie pojawiające się na serwerze zarządzanym są usuwane bezpośrednio przez agenta AI. Generowanie bardzo dużego ruchu sieciowego w rozbudowanych rozproszonych systemach komputerowych jest jedną z wad agenta SNMP. Zastosowanie zaawansowanych mechanizmów bezpiecznej komunikacji jest kolejną ważną funkcją agenta AI, z kolei proces komunikacji agenta SNMP jest mało bezpieczny, co wyklucza go z poważnych zastosowań. Podsumowując można stwierdzić, iż funkcjonalność agenta SNMP jest wystar-

czająca w przypadku prostych urządzeń sieciowych, jak koncentratory i przełączniki. Natomiast w celu zarządzania rozbudowanymi serwerami komputerowymi należy stosować agentów AI, gdyż jedynie takie rozwiązanie zapewni wydajne i bezpiecznie zarządzanie.

LITERATURA

1. Rose, Marshall T.: The Simple Book. An Introduction to Management TCP/IP-based Internets. Englewood Cliffs, New Jersey, Prentice-Hall, 1989.
2. HP Openview IT/Operations Reference Manual by Hewlett Packard.
3. HP Openview IT/Operations Concept Guide by Hewlett Packard.
4. HP Openview IT/Operations Network and Platform Functions by Hewlett Packard.
5. Tanenbaum A. S.: Distributed Operating Systems. Prentince-Hall International Inc., 1995.
6. Shapley Gray J.: Interprocess Communications in UNIX. Second Edition, 1997.
7. Stallings W.: SNMP, SNMP2, and CMIP: The Practical Guide to Management Standarts. Addison-Wesley Publishing Co., Inc ©1993.
8. William Stallings, Ph. D. Network and Internetwork Security Principles and Practice, Copyright © 1994 by Prentice Hall, Inc.
9. Standard POSIX 1003.6, Access Control Lists ACL.
10. RFC 1157, A Simple Management Protocol (SNMPv1).
11. RFC 1910, User-based Security Model for SNMPv2.
12. RFC 2264, User-based Security Model (UsM) for version 3 of the Simple Network Management Protocol SNMPv3.

Recenzent: Dr inż. Andrzej Białas

Wpłynęło do Redakcji 3 kwietnia 2000 r.

Abstract

This article describes computer systems management based upon the protocol RPC. It presents architecture of the intelligent agent as well as its main management functions and possibilities (fig. 1). The structure of each ITO module of the agent and the functions of every module could be found in the article items. The mechanism of collecting information and

intelligent management functions are presented herein as well as all possible sources of information, which are used for generating system state messages. Fig. 2 shows threshold system, used by monitoring module for detecting and handling predefined situations in the managed node. The explanation of the process of performing automatic actions in case of situations mentioned above is showed on fig. 3. There were explain possibilities of integration agent AI with other management agents, for instance with agent SNMP. The article presents agent ITO communication processes and security mechanism in distributed enviroment as well as distributed computing enviroment DCE RPC, which is used by agent AI for communication with management server. Fig. 4. shows operation pattern of RPC protocol. It describes security procedures, which are used to improve security level of communication between agent AI and management server. There were described authorization and authentication security procedures. The article presents process of encrypted data transmission between agent AI and management server (fig. 5).