ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1997
Seria: INFORMATYKA z. 32 Nr kol. 1356

Grzegorz HRYŃ, Stanisław CIEŚLA
Politechnika Śląska, Instytut Informatyki

METODY ZAPEWNIENIA BEZPIECZEŃSTWA W SYSTEMIE WINDOWS NT

Streszczenie. W artykule przedstawiono charakterystykę modelu ochrony stosowanej w wieloprotokołowych środowiskach sieciowych na przykładzie systemu Windows NT. Przedstawiono także metody ochrony zasobów oraz opisano mechanizmy nadzoru.

METHODS OF SECURITY ASSURANCE IN WINDOWS NT NETWORK SYSTEM

Summary. This article provides an overview of the security model of contemporary network environment (Windows NT). It also explains methods of resource protection, and describes possibilities of auditing.

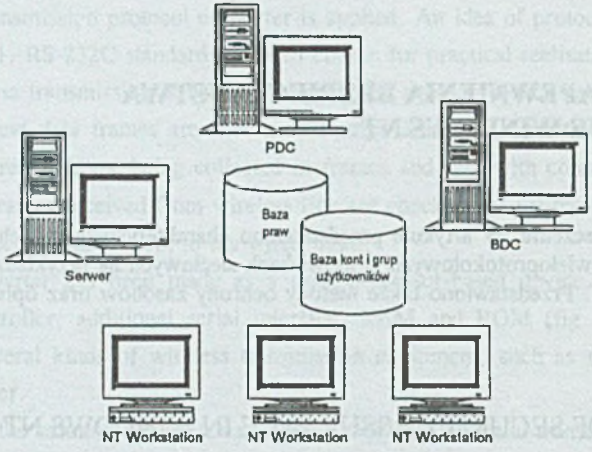
1. Bezpieczeństwo w domenie

Podstawową jednostką zapewniającą bezpieczeństwo w sieci komputerów opartych na systemie Windows NT jest domena [1]. Jest to zbiór serwerów współdzielący informacje dotyczące kont użytkowników oraz praw użytkowników i grup. Jeden z nich pełni rolę kontrolera domenowego (PDC – primary domain controller), przechowując scentralizowane bazy kont i przywilejów. Dodatkowo mogą istnieć wspierające kontrolery domenowe (BDC – backup domain controllers), które przejmują funkcje PDC w przypadku awarii tego ostatniego, co umożliwia funkcjonowanie domeny pomimo wyłączenia PDC.

W skład domeny mogą wchodzić również serwery nie pełniące funkcji kontrolerów domenowych, wtedy ich rola ograniczona jest do udostępniania zasobów, jak również komputery

z systemem NT Workstation, maszyny pracujące pod kontrolą systemów jednoużytkownikowych (MS-DOS, Windows for Workgroups czy Windows 95).

Zaletą takiego rozwiązania jest w pełni scentralizowany mechanizm ochrony zasobów w domenie, co w dużym stopniu utrudnia włamanie do sieci, a równocześnie ułatwia administratorowi zarządzanie, zmniejszając prawdopodobieństwo pozostawienia luki w systemie, gdyż informacja o grupach i użytkownikach jest przechowywana w jednym miejscu.



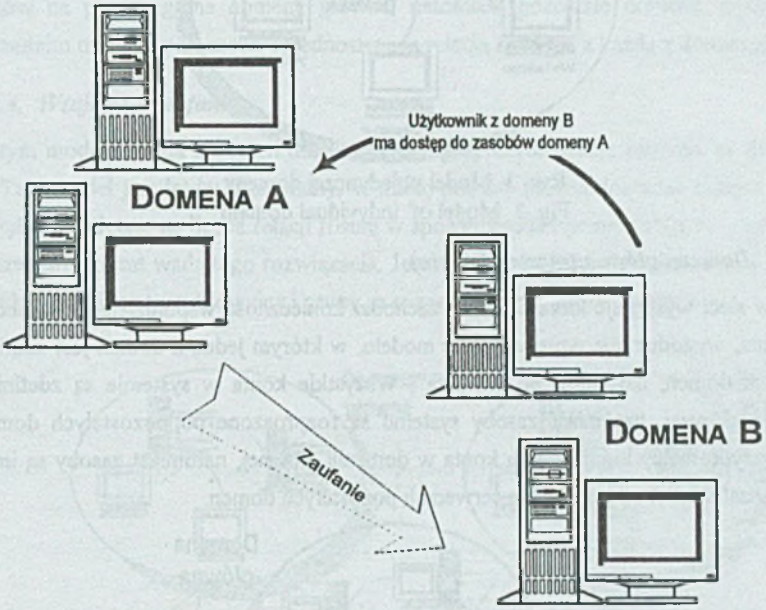
Rys. 1. Domena systemu Windows NT
Fig. 1. Domain of Windows NT operating system

1.1. Zależności pomiędzy domenami

W skład danej sieci może wchodzić więcej domen. Aby skorzystać z zasobów innej domeny, użytkownik musi mieć w niej swoje własne konto, co powoduje kłopoty w synchronizacji praw i przywilejów w takiej sieci, utrudniając pracę administratorom. Aby uniknąć takiej sytuacji, można zdefiniować relację zaufania (trust relationship) [1]. Jest to połączenie pomiędzy domenami pozwalające użytkownikowi posiadającemu konto w jednej z domen na dostęp do zasobów innej domeny. Połączenie takie ma ściśle określony kierunek: jedna z domen „ufa" innej, co pozwala na dostęp do jej zasobów użytkownikom z zaufanej domeny, pomimo że nie mają w niej swoich kont.

Dodatkowo użytkownicy z domeny B mogą rejestrować swoje zadania (logować się) na komputerze należącym fizycznie do domeny A. Nie zachodzi natomiast związek odwrotny – użytkownicy w domenie A nie mają dostępu do zasobów domeny B i nie mogą logować się na komputerach należących do domeny B. Aby zapewnić taką możliwość, należy zdefiniować dodatkowo relację zaufania z domeny B do domeny A.

Relacje zaufania nie są przechodnie [3], co oznacza, że jeżeli domena A „ufa" domenie B, a ta z kolei „ufa" domenie C, to domena A nie „ufa" domyślnie domenie C. Taką zależność trzeba definiować dodatkowo.



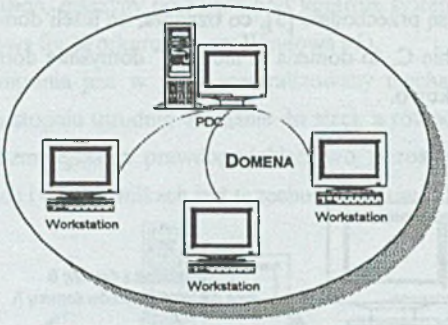
Rys. 2. Zaufanie w domenach
Fig. 2. Trust relationship in domains

1.2. Konfiguracja domen

W zależności od potrzeb i rozmiarów sieci, którą należy skonfigurować, możliwe są różne scenariusze budowy domen w takiej sieci [1 i 3]:

1.2.1. Pojedyncza domena

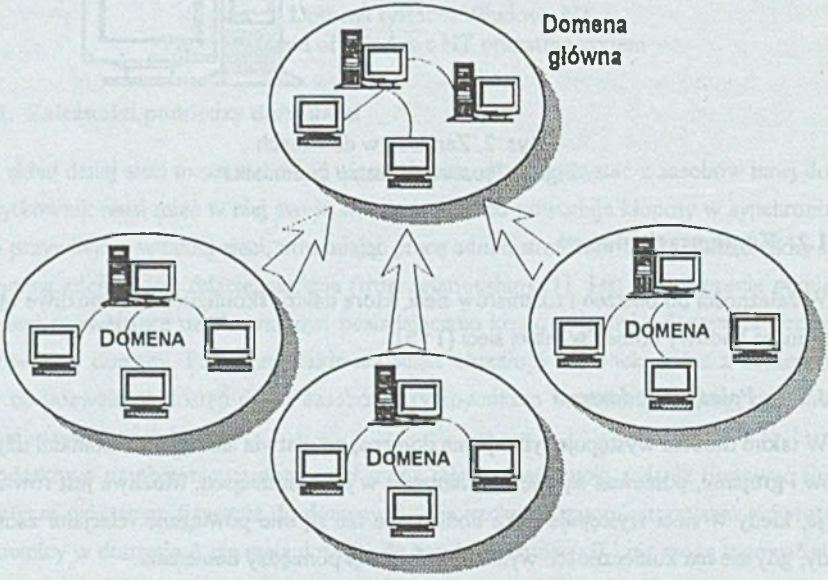
W takim modelu występuje tylko jedna domena, co ułatwia zarządzanie kontami użytkowników i grupami, ponieważ są one zdefiniowane w jednym miejscu. Możliwa jest również sytuacja, kiedy w sieci występuje kilka domen, ale nie są one powiązane relacjami zaufania – wtedy, gdy nie ma konieczności wymiany informacji pomiędzy domenami.



Rys. 3. Model pojedynczej domeny
Fig. 3. Model of indyvidual domain

1.2.2. Domena główna (master domain)

Jeśli w sieci występuje kilka domen i zachodzi konieczność współdzielenia zasobów pomiędzy nimi, wygodne jest wprowadzenie modelu, w którym jedna z domen jest zaufaną dla pozostałych domen, natomiast im nie „ufa". Wszystkie konta w systemie są zdefiniowane w domenie głównej, natomiast zasoby systemu są rozproszone po pozostałych domenach. Wszyscy użytkownicy logują się na konta w domenie głównej, natomiast zasoby są im przydzielane w zależności od potrzeb na serwerach pozostałych domen.



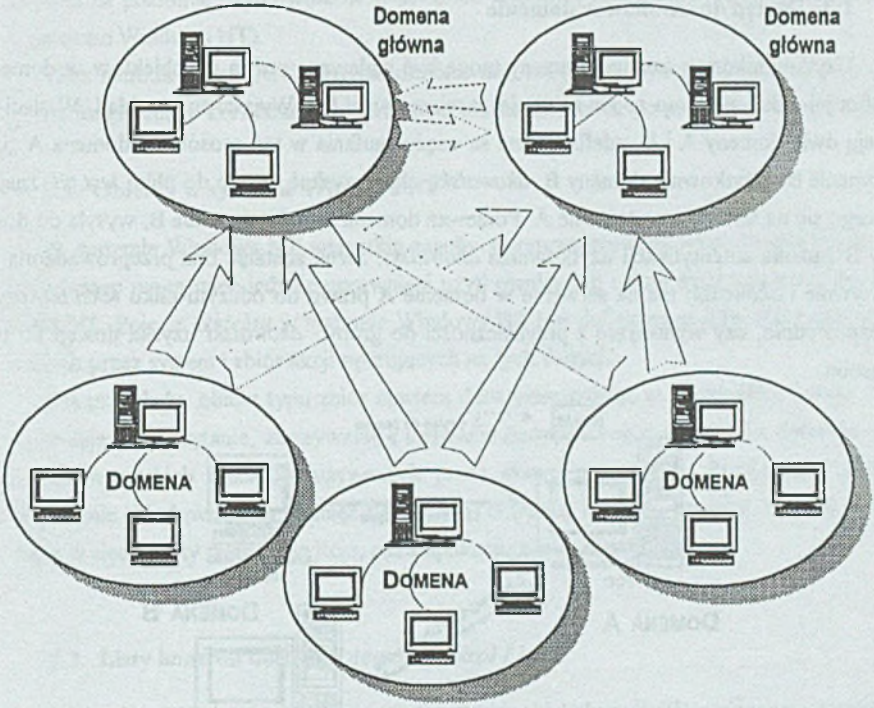
Rys. 4. Model systemu z domeną główną
Fig. 4. System with master domain

1.2.3. Wiele domen głównych (multiple master domain)

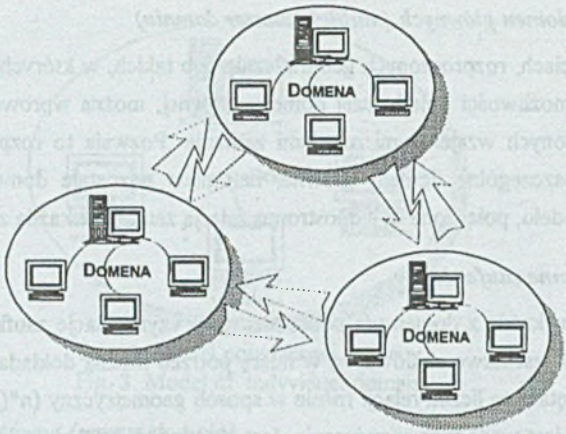
W dużych sieciach, rozproszonych geograficznie, lub takich, w których liczba użytkowników przekracza możliwości pojedynczej domeny głównej, można wprowadzić kilka domen głównych, połączonych wzajemnymi relacjami zaufania. Pozwala to rozproszyć bazę użytkowników na poszczególne domeny główne, natomiast pozostałe domeny, podobnie jak w poprzednim modelu, połączone są jednostronną relacją zaufania z każdą z domen głównych.

1.2.4. Wzajemne zaufanie

W tym modelu każda z domen ufa pozostałej, przy czym relacje zaufania są dwukierunkowe. Taki model jest łatwo skalowalny, w miarę potrzeb można dokładać kolejne domeny, trzeba tylko pamiętać, że liczba relacji rośnie w sposób geometryczny (n*(n-1) – gdzie n jest liczbą domen), co jest wadą tego rozwiązania. Jest również trudniej zarządzać siecią opartą na tym modelu, ponieważ użytkownicy i grupy są rozproszeni pomiędzy domeny.



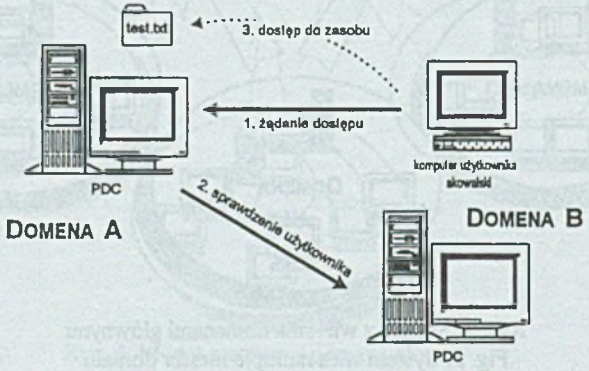
Rys. 5. System z wieloma domenami głównymi
Fig. 5. System with multiple master domain



Rys. 6. Model domen opartych na wzajemnym zaufaniu
Fig. 6. Model with trust relationship in every domain

1.3. Dostęp do zasobów w domenie

Użytkownikom z zaufanej domeny mogą być nadawane prawa do obiektów w domenie, która jej „ufa", pomimo że nie są w niej zarejestrowani [3]. Wyjaśni to przykład. W sieci istnieją dwie domeny A i B, zdefiniowane są więzy zaufania w ten sposób, że domena A „ufa" domenie B. Użytkownik domeny B, *akowalski*, chce uzyskać dostęp do pliku *test.txt*, znajdującego się na serwerze w domenie A. Ponieważ domena A „ufa" domenie B, wysyła do domeny B żądanie autentyfikacji użytkownika *akowalski*. Jeżeli zostanie ona przeprowadzona pozytywnie i *akowalski* ma na serwerze w domenie A prawo do odczytu pliku *tekst.txt*, czy to bezpośrednio, czy wynikające z przynależności do grupy, *akowalski* uzyska dostęp do tego zasobu.



Rys. 7. Dostęp do zasobów w domenach
Fig. 7. Rasource access in domains

2. Model ochrony zasobów

2.1. Wprowadzenie

Ochrona zasobów w wieloprotokołowych systemach operacyjnych polega na kontroli praw dostępu użytkowników do zbiorów drukarek i aplikacji. Ochrona zasobu przejawia się jego dostępnością jedynie dla uprawnionych użytkowników i równocześnie brakiem możliwości dostępu dla pozostałych użytkowników.

Znane są dwie podstawowe strategie ochrony zasobów. Pierwsza polega na związaniu z zasobem pewnego kodu dostępu i tylko jego znajomość pozwala użytkownikowi na dostęp do zasobu, innymi słowy – dostęp do zasobu może uzyskać każdy użytkownik znający ten kod (jako przykład ochrona na poziomie współdzielonego dostępu w systemie Windows 95, kiedy wymagane jest hasło dostępu), natomiast druga metoda polega na nadaniu poszczególnym użytkownikom określonych praw dostępu do zasobów systemu operacyjnego (jako przykład ochrona na poziomie użytkowników w systemie Windows 95 i mechanizm ochrony zasobów w systemie Windows NT).

Taka metoda zapewnia, że jedynie użytkownik, któremu przydzielono prawa do zasobu, będzie mógł z niego korzystać po poprawnym zidentyfikowaniu go przez system operacyjny.

2.2. Obiekty w systemie Windows NT

W systemie Windows NT wszystkie zasoby są reprezentowane jako obiekty [3], do których dostęp mogą mieć jedynie uprawnieni użytkownicy lub usługi systemowe systemu Windows NT. Pojęcie *Obiektu* w systemie Windows NT jest definiowane jako zbiór danych używanych przez system i zbiór akcji operujących na tych danych.

Dla przykładu, obiekt typu zbiór zawiera dane umieszczone w tymże pliku i zbiór funkcji pozwalający na czytanie, zapisywanie i usuwanie danych z tegoż zbioru. Ta definicja odnosi się do wszystkich obiektów używanych przez system operacyjny. Przykładami obiektów w systemie Windows 95 są: pamięć operacyjna, drukarki, katalogi, procesy, katalogi udostępniane w sieci, porty wejścia-wyjścia, okna aplikacji, zbiory, wątki.

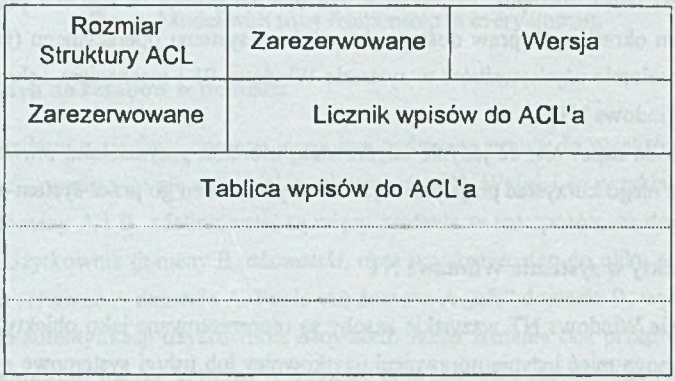
2.3. Listy kontroli dostępu (*Access Control Lists*)

Wszystkie funkcje używane do operowania na obiekcie są ściśle związane z konkretnym obiektem. Dodatkowo użytkownicy i grupy użytkowników, którym nadano prawa do używania tych funkcji, także są ściśle związani z tym obiektem. Tylko użytkownicy posiadający odpowiednie prawa mogą używać funkcji danego obiektu. Rezultatem tego jest, że funkcje ope-

rujące na danych jednego procesu nie są w stanie operować danymi należącymi do innego procesu.

Tak wprowadzona i rozumiana charakterystyka obiektów tworzy tak zwaną „ochronę wbudowaną" (*built-in security*). Prawa dostępu do każdego obiektu kontrolowane są poprzez strukturę zwaną Listą Kontroli Dostępu (*Access Control List - ACL*). Lista Kontroli Dostępu zawiera informacje o użytkownikach i grupach użytkowników, którzy mają dostęp do tego obiektu i posiadają prawa do wykonywania operacji na tym obiekcie [2 i 3].

Gdy jakiś użytkownik chce wykonać operację na obiekcie, system operacyjny weryfikuje użytkownika, sprawdza jego przynależność do grup i porównuje te informacje z Listą Kontroli Dostępu obiektu; wynik tego porównania determinuje, czy użytkownik wykona żądaną operację, czy też system operacyjny zabroni jej wykonania.

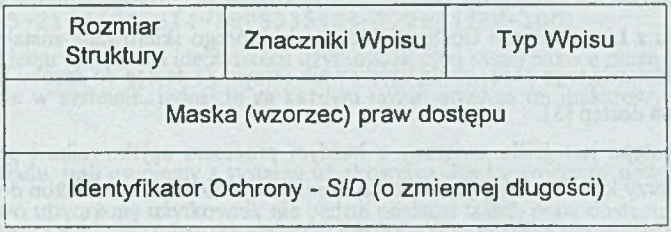


Rys. 8. Struktura Listy Kontroli Dostępu
Fig. 8. Access Control List data structure

Każdy użytkownik w systemie operacyjnym musi posiadać jednoznaczny identyfikator, który może zostać dodany do Listy Kontroli Dostępu danego zasobu; dotyczy to także aplikacji i usług systemowych, które mogą tak samo jak użytkownicy żądać dostępu do określonych zasobów. Gdy administrator lub osoba uprawniona nada prawa dostępu użytkownikowi do zasobu, identyfikator użytkownika jest dodawany do listy kontroli dostępu tego zasobu wraz z wyspecyfikowanymi prawami, jakie przydzielono dodawanemu użytkownikowi.

Dla przykładu, użytkownik *Jan* posiada prawo czytania ze zbioru *Info.txt*, podczas gdy użytkownik *Piotr* posiada prawa czytania, pisania i usuwania do tego samego zbioru. Każdy taki wpis do Listy Kontroli Dostępu identyfikuje użytkownika lub grupę i prawa, jakie przyznano do danego obiektu. Wpisy do Listy Kontroli Dostępu są generowane dla każdego użytkownika, któremu nadano lub cofnięto prawa do danego obiektu. Wpisy zabraniające dostępu

są umieszczone na początku Listy Kontroli Dostępu, a dopiero po nich wyspecyfikowane są wpisy zezwalające na dostęp.



Rys. 9. Struktura Wpisu do Listy Kontroli Dostępu
Fig. 9. Access Control Entry data structure

2.4. Ochrona dostępu do zasobów

Problem dostępu do zasobów zaczyna się od momentu wejścia użytkownika do systemu. System Windows NT wymaga, by użytkownik został zweryfikowany (podał identyfikator i hasło) przed udostępnieniem mu jakiegokolwiek zasobu. Gdy użytkownik wchodzący do systemu zostanie poprawnie zweryfikowany (system sprawdzi jego identyfikator i hasło), zostaje mu przypisana pewnego rodzaju struktura, która jest z nim nierozerwalnie związana aż do opuszczenia przez niego systemu operacyjnego [1 i 3].

Strukturę tę możemy przyrównać do pewnego rodzaju żetonu dostępu lub karty identyfikacyjnej. Każdorazowo, gdy użytkownik żąda dostępu do zasobu, informacje pochodzące z jego żetonu dostępu są porównywane z wpisami w Liście Kontroli Dostępu zasobu, do którego wystąpiło żądanie dostępu, celem określenia, czy użytkownikowi należy zezwolić na dostęp czy też nie.

2.5. Obowiązkowa procedura identyfikacji użytkownika

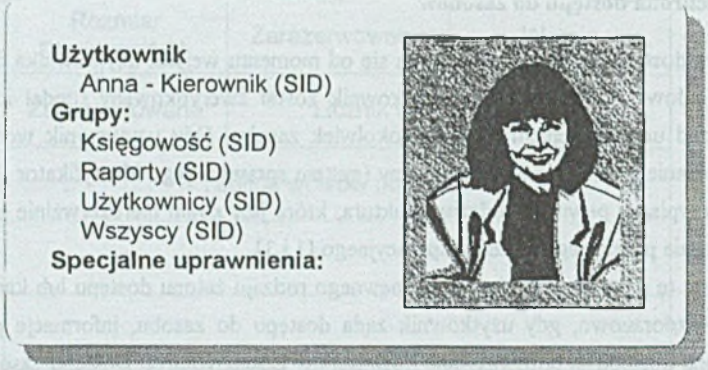
System Windows NT wymaga, by każdy użytkownik systemu wprowadził unikatowy identyfikator i hasło podczas wchodzenia do systemu. Gdy użytkownik zostanie pozytywnie zweryfikowany przez system, podsystem ochrony systemu operacyjnego tworzy dla tego użytkownika strukturę zwaną żetonem dostępu. Struktura ta zawiera informacje o nazwie użytkownika i grupach, do których użytkownik przynależy.

Użytkownik może korzystać z systemu operacyjnego dopiero po otrzymaniu utworzonego dla niego żetonu dostępu. Użytkownik, który wszedł do systemu podczas swojej pracy, jest identyfikowany przez system operacyjny właśnie poprzez ten żeton dostępu, który otrzymał podczas procedury wchodzenia do systemu.

2.6. Żeton dostępu

Gdy proces użytkownika żąda dostępu do obiektu, system Windows NT porównuje identyfikator użytkownika i listę grup, do których użytkownik przynależy, zapisane w żetonie dostępu procesu z Listą Kontroli Dostępu obiektu, do którego skierowane zostało żądanie dostępu. Na podstawie tego porównania użytkownik otrzymuje dostęp do zasobu lub też system nie zezwala na dostęp [3].

Żeton dostępu jest ściśle związany z każdym procesem użytkownika i pełni rolę jego „paszportu" przy każdorazowej próbie korzystania z zasobów systemu. Żeton dostępu też jest obiektem w rozumieniu systemu operacyjnego i posiada zarówno dane, jak i operujące na nim funkcje jak każdy inny obiekt w systemie.



Rys. 10. Postać żetonu dostępu
Fig. 10. Model of access token

2.7. Identyfikator ochrony (*Security ID – SID*)

Pomimo że w systemie zarówno użytkownicy, jak i grupy użytkowników są specyfikowani za pomocą unikatowych nazw, to system operacyjny identyfikuje użytkowników za pomocą identyfikatorów ochrony (*SID*), a grupy za pomocą identyfikatorów ochrony grup (*Group SID*) [2 i 3]. Identyfikatory te są unikatowe i używane w systemie do reprezentowania użytkowników i grup.

Identyfikatory ochrony są używane w żetonach dostępu i Listach Kontroli Dostępu zamiast nazw użytkowników i grup. Identyfikatory ochrony są tworzone na podstawie informacji o użytkowniku, czasie, dacie i domenie. Identyfikator ochrony jest reprezentowany jako unikatowy numer zgodny z poniższym schematem:

$$S\text{-}1\text{-}X\text{-}Y^{1}\text{-}Y^{2}\text{-}\ldots\ldots\ldots\text{-}Y^{n}$$

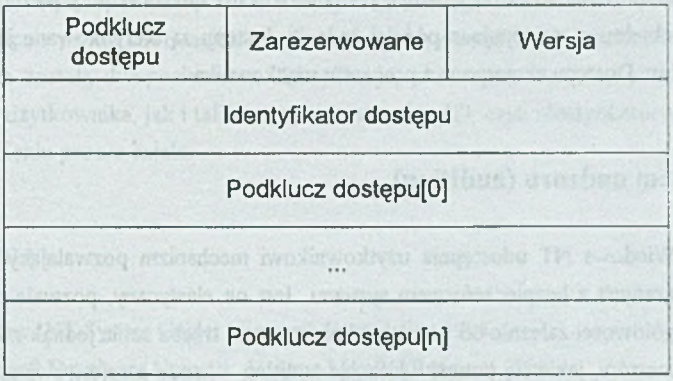
„S-1" oznacza, że jest to wersja pierwsza identyfikatora, X oznacza identyfikator dostępu, natomiast Y^1 do Y^n symbolizują podklucze dostępu.

Poniżej przedstawiony jest przykład identyfikatora:

```
S-1-5-21-76965814-1898335404-322544488-1001
```

Dzięki takiemu systemowi identyfikacji użytkownik o tej samej nazwie może być utworzony kilkakrotnie w systemie, jednakże za każdym razem otrzyma on unikatowy identyfikator ochrony.

Dla przykładu, jeśli usuniemy z systemu użytkownika *Jan* i utworzymy nowego użytkownika *Jan*, nowo utworzony użytkownik nie będzie posiadał takich praw dostępu do zasobów, jakie posiadał poprzedni użytkownik. Wynika to z faktu, że dla nowo utworzonego użytkownika o takiej samej nazwie generowany jest zupełnie nowy i unikatowy identyfikator ochrony i to właśnie on pełni funkcję identyfikacji użytkownika.



Rys. 11. Struktura Identyfikatora Ochrony
Fig. 11. Security ID data structure

2.8. Kontrola praw dostępu

System Windows NT porównuje informacje zawarte w żetonie dostępu z wpisami zawartymi w Liście Kontroli Dostępu celem określenia, czy użytkownik może z zasobu skorzystać czy też nie może. Podczas żądania dostępu do zasobu podsystem ochrony sprawdzając prawa dostępu użytkownika postępuje zgodnie z podanym poniżej schematem:

a) rozpoczyna od analizy początkowych wpisów Listy Kontroli Dostępu pod kątem wpisu zabraniającego użytkownikowi lub grupie, do której należy, żądanego rodzaju dostępu do zasobu,
b) w następnej kolejności sprawdza, czy żądany rodzaj dostępu jest wprost wyspecyfikowany jako możliwy dla użytkownika lub grupy, do której użytkownik należy,

c) powtarza punkty 1. i 2. dla każdego wpisu w Liście Kontroli Dostępu aż do momentu znalezienia pierwszego wpisu zabraniającego żądanego dostępu lub skompletowania wszystkich uprawnień koniecznych do zezwolenia na żądany rodzaj dostępu,
d) jeśli w wyniku analizy całej Listy Kontroli Dostępu nie zostanie odnaleziony wpis jednoznacznie określający, czy żądany dostęp ma zostać zrealizowany czy też nie, użytkownik nie otrzymuje zezwolenia na zrealizowanie żądanego dostępu.

2.9. Optymalizacja kontroli dostępu do zasobów

Celem zoptymalizowania procedur kontroli dostępu do zasobów system Windows NT po pierwszym żądaniu dostępu do obiektu przez proces użytkownika tworzy listę dozwolonych rodzajów dostępu do obiektu zwaną Listą Nadanych Praw Dostępu i wiąże ją z procesem użytkownika.

W ten sposób Lista Kontroli Dostępu jest analizowana tylko raz przy pierwszym żądaniu dostępu do obiektu, a następujące później żądania dostępu są weryfikowane jedyne z Listą Nadanych Praw Dostępu skojarzoną z procesem użytkownika.

3. System nadzoru (auditing)

System Windows NT udostępnia użytkownikowi mechanizm pozwalający na śledzenie zdarzeń związanych z bezpieczeństwem systemu. Jest on elastyczny, pozwala na dobór poziomu szczegółowości zależnie od potrzeb administratora, trzeba sobie jednak zdawać sprawę z pewnych kosztów, jakie się ponosi włączając auditing – każde zarejestrowane zdarzenie to zmniejszenie wydajności systemu [1 i 3].

Windows NT może śledzić zdarzenia związane nie tylko z systemem, ale również z konkretnymi aplikacjami, jednak o ile zdarzenia systemowe są ściśle określone, o tyle twórca aplikacji decyduje, które zdarzenia z nią związane należy rejestrować.

System może rejestrować zdarzenia należące do jednej z poniższych kategorii:

a) Zarządzanie grupami i użytkownikami – zdarzenia te opisują zmiany w bazie kont i grup użytkowników, takie jak stworzenie użytkownika, usunięcie, zmiana przynależności do grupy, stworzenie nowej grupy itp.
b) Logon/logoff – śledzenie zarówno pomyślnych, jak i niepomyślnych prób logowania się i opuszczania systemu, zarówno poprzez sieć, interaktywnie (z komputera, dla którego prowadzone jest śledzenie), jak również jako część systemu (service).
c) Zmiany w systemie praw – zdarzenia opisujące zmiany w bazie praw, jak dodanie komuś przywileju dostępu do jakiegoś zasobu systemu, zmiana sposobu rejestracji w do-

menie (np. zezwolenie użytkownikowi na logowanie się na lokalnie serwerze, a nie tylko poprzez sieć).

d) Dostęp do obiektów systemu – śledzenie pomyślnych i niepomyślnych prób dostępu do zasobów, dla których zdefiniowano konieczność śledzenia.

e) Użycie praw przez użytkownika – śledzenie pomyślnych i niepomyślnych prób zmian w systemie dokonywanych przez użytkownika, jak np. przejęcie własności pliku lub katalogu, zmiana czasu.

f) Zdarzenia systemowe – obejmujące zamknięcie, ponowny start systemu, uruchomienie procesów rejestrujących użytkowników w systemie.

g) Śledzenie procesów – rejestracja zdarzeń obejmujących uruchomienie i zakończenie działania procesów w systemie.

Jednym z ważniejszych aspektów bezpieczeństwa jest określenie, kto jest odpowiedzialny za operację przeprowadzoną w systemie. Chociaż z każdym wątkiem w systemie jest związany identyfikator, wątek może działać w imieniu jakiegoś innego użytkownika (impersonation). W celu określenia pełnej informacji o użytkowniku, który przeprowadzał określoną operację, wprowadzona została dwupoziomowa identyfikacja, określająca zarówno pierwszoplanowy identyfikator użytkownika, jak i tak zwany impersonation ID, czyli identyfikator użytkownika, w którego imieniu proces działa.

LITERATURA

[1] Windows NT System Guide. Microsoft Corporation 1995.

[2] Microsoft Developer Network. Volume May 96. Redmont 1996.

[3] Microsoft TechNet. Volume 4, Issue 8, August 1996.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 18 listopada 1996 r.

Abstract

The security model of network system is described in this paper, concerning domain security and trust relationships. It explains how Windows NT tracks each user and each securable object, how users and groups are identified internally in the system. This article also provides

examples of Windows NT security, showing how Windows NT validates access requests and how it audits activities performed on protected objects.