Dynamiczne zarządzanie tożsamością użytkowników w przestrzeniach inteligentnych

Boiński T.

8 stycznia 2010

Streszczenie. Omówiono zagadnienie uwierzytelniania oraz przedstawiono różnice pomiędzy rozwiązaniami dostarczanymi przez systemy Linux oraz Windows. Zaprezentowano dwie metody zapewnienia scentralizowanego systemu uwierzytelniającego. Obie metody były przetestowane w praktyce w sieci laboratorium komputerowego katedry KASK. Na tej podstawie wskazano metodę zdolną do zapewnienia bezpieczeństwa przestrzeni inteligentnej, w której to użytkownicy będą mogli uwierzytelniać się za pomocą innych atrybutów niż tylko login i hasło.

Słowa kluczowe. zapewnienie bezpieczeństwa, uwierzytelnianie, integracja systemów, LDAP, Active Directory

1 Klasyfikacja metod uwierzytelniania

Uwierzytelnianie[1] jest to proces weryfikacji tożsamości zadeklarowanej przez użytkownika, urządzenie lub usługę biorącą udział w procesie wymiany informacji. Klasycznie proces uwierzytelniania oparty był o zasadę "co wiesz" - potwierdzenie tożsamości odbywa się poprzez podanie informacji znanej tylko uwierzytelniającemu się podmiotowi, np. loginu i hasła. Proces ten, mimo swojej prostoty, nadal jest wykorzystywany, głównie do uwierzytelniania użytkowników systemów komputerowych i usług.

Wraz z rozwojem kryptografii i metod uwierzytelniania proces ten został rozszerzony o zasadę "co masz", czyli pewien element będący w posiadaniu uwierzytelniającego się podmiotu, np. token, certyfikat, karta inteligentna, itp. Taki rodzaj zabezpieczeń zwiększa bezpieczeństwo operacji uwierzytelniania, przez co często spotykany jest np. w systemach bankowych. Uwierzytelniając się podajemy numer wygenerowany przez token i PIN. W przypadku, gdy jakiemuś intruzowi uda się podsłuchać hasło, nie będzie w stanie wygenerować kolejnego numeru identycznego z naszym tokenem. Jeśli napastnik zdoła skraść nasz token, będzie on bezużyteczny, jeżeli nie zna hasła. Musiałby zarówno poznać hasło i skraść token, co jest zadaniem dużo trudniejszym i mniej prawdopodobnym. Z biegiem czasu, dzięki rozwojowi biometryki, podejście "co masz" rozszerzone zostało o cechy biometryczne specyficzne dla danej jednostki, np. odcisk palca, owal twarzy, źrenica oka itp.

W systemach komputerowych, niezależnie od przyjętej formy uwierzytelniania, dane podane przez użytkownika najczęściej porównywane są z wzorcem

przechowywanym w bazie danych. W związku z tym wyróżnić możemy trzy metody uwierzytelniania:

- pojedyncze, jednorodne stanowisko – zazwyczaj występuje tutaj jedna forma uwierzytelniania, wprowadzone informacje porównywane są z wzorcem zapisanym w lokalnej bazie danych, brak jest konfliktów przy dostępie do danych,
- rozproszone, jednorodne środowisko – występuje tutaj jedna lub więcej form uwierzytelniania, wzorzec zapisany jest w centralnej bazie danych ulokowanej na innym serwerze niż system, do którego użytkownik próbuje się uwierzytelnić. Zachodzi konieczność zapewnienia równoległego dostępu do danych przez wielu klientów jednocześnie, jednak wszyscy oni komunikują się z bazą za pomocą tego samego protokołu i oczekują informacji zwrotnej w tym samym formacie, przez co jest relatywnie prosty do implementacji. Najczęściej stosowanymi rozwiązaniami są Lightweight Directory Access Protocol (LDAP)[2] dla systemów opartych o Linux oraz Active Directory[3] dla systemów opartych o Windows,
- rozproszone, heterogeniczne środowisko – występuje tutaj wiele mechanizmów uwierzytelniania użytkowników oraz konieczność ich adaptacji do wybranej wspólnej zewnętrznej bazy danych lub synchronizacji danych pomiędzy rożnymi systemami składowania informacji. Niestety rodzi to poważne problemy implementacyjne lub logistyczne, zależnie od wybranego wariantu.

W dalszej części niniejszej publikacji rozpatrzony zostanie jedynie ostatni przypadek stosowany w laboratorium komputerowym "Inteligentne usługi i dedykowane systemy internetowe" należący do Katedry Architektur Systemów Komputerowych[4]. W skład tego laboratorium wchodzą komputery klienckie oraz serwery działające pod różnymi wersjami systemów Windows (Vista, 2000 Serwer oraz 2008 Serwer) oraz Linux (openSuSE 11.1, Red Hat Enterprise Linux, CentOS, Rocks 4.1 Fuji) oraz sieć bezprzewodowa oparta o zabezpieczenia typu WPA Enterprise[5] i serwer RADIUS[6][7]. W niedalekiej przyszłości planowane jest rozszerzenie laboratorium o inne rodzaje uwierzytelnia niż za pomocą nazwy użytkownika i hasła – w pierwszej kolejności za pomocą kart inteligentnych z zapisanymi certyfikatami użytkowników.

W pierwszym rozdziale przedstawione zostaną typowe problemy napotykane w trakcie integracji systemów. Konkretne rozwiązania zostaną opisane w rozdziale trzecim. W ostatnim rozdziale zostanie wykorzystane zmodyfikowane rozwiązanie dla przestrzeni usług.

2 Problemy napotykane w trakcie integracji systemów Linux i Windows

Systemy Windows oraz Linux różnią się między sobą diametralnie, a różnice te są podstawą większości problemów pojawiających się w trakcie integracji środowisk złożonych z obu tych systemów. Sytuację dodatkowo komplikuje fakt, iż oba te systemy są dla siebie wzajemną konkurencją zarówno na polu rozwiązań serwerowych jak i stacji roboczych. Na szczęście od czasu porozumienia[8]

między firmami Novell i Microsoft można mówić o pewnym porozumieniu dotyczącym zapewnienia bezpieczeństwa, przez co bardzo wiele problemów da się przynajmniej częściowo zniwelować. Poniżej przedstawione zostały podstawowe różnice między oboma systemami.

Podstawowe różnice dzielące oba systemy to kwestie czysto technologiczne, wynikające z odmiennej architektury czy przyjętych rozwiązań. Poza tym różne są też rozwiązania dotyczące: zasad działania, kontroli użytkowników, dostępu do dokumentów itp. Do najważniejszych z nich można zaliczyć:

- identyfikacja użytkowników – zarówno system Linux jak i Windows nie rozróżniają użytkownika po nazwie (choć ta musi być unikatowa) a po ich identyfikatorach. W obu przypadkach zarówno każdy użytkownik jak i grupa mają swój unikatowy identyfikator, generowany jednak w inny sposób. Konieczne jest więc zarządzanie oboma sposobami numeracji,

- uprawnienia użytkowników – system Linux stosuje prosty, trzy-poziomowy system uprawnień – każdemu plikowi i katalogowi przypisany jest użytkownik oraz grupa. Uprawnienia sterowane są na poziomie – użytkownik, grupa oraz wszyscy pozostali i obejmuje prawo do odczytu, zapisu oraz wykonania. W systemie Windows możliwe jest tworzenie dowolnie skomplikowanej struktury przypisując prawa dostępu do plików i katalogów dowolnej liczbie użytkowników i grup. Możliwe jest również o wiele większe rozdrobienie uprawnień, np. odczyt pliku, odczyt uprawnień do pliku, odczyt rozszerzonych atrybutów pliku itp.

- obsługiwane systemy plików – system Windows obsługuje tylko system FAT oraz NTFS, przy czym instalacja systemu zalecana jest na systemie NTFS. System Linux obsługuje bardzo dużo systemów plików, w tym FAT i NTFS, jednak zaleca się by był instalowany na dyskach sformatowanych z wykorzystaniem natywnych dla niego systemów takich jak np. ext3[9],

- nazwy plików – system Linux obsługuje pliki o dowolnych nazwach, również zawierających znaki specjalne, takie jak "*", "?" itp. System Windows z kolei uniemożliwia zapis (a czasami nawet i odczyt) pliku zawierającego zabronione przez niego znaki specjalne. Ponadto nazwy plików tak zapisanych w systemie Windows wyświetlone będą nieprawidłowo,

- rozróżnianie wielkości znaków – systemy Linux są z natury "case sensitive", systemy Windows z natury "case insensitive" - nazwy plików są w tym przypadku zapisywane i odczytywane za uwzględnieniem wielkości liter, jednak wszelkie operacje na nich, jak porównywanie nazw, wyświetlanie zawartości katalogów, wyszukiwanie, uruchamianie, itp. odbywają się już z pominięciem tej informacji. Rodzi to poważne problemy w przypadku próby odczytu z poziomu systemu Windows katalogu zawierającego pliki bądź katalogi różniące się wielkością liter.

- przechowywanie danych uwierzytelniających oraz protokół uwierzytelniania użytkowników – system Windows centralne uwierzytelnianie użytkowników opiera o różne wersje Active Directory. Dla systemów opartych o Linux najczęściej stosowanym obecnie rozwiązaniem jest LDAP. Oba te rozwiązania wywodzą się ze wspólnych korzeni, lecz w podstawowej postaci są niekompatybilne ze sobą ze względu na różnice w zaimplementowanych atrybutach,

- mechanizm szyfrowania hasła i sposób jego zapisu – oba systemy szyfrują hasła w odmienny sposób, konieczne jest więc zapewnienie ich składowania na oba sposoby i synchronizacji w przypadku jego zmiany.

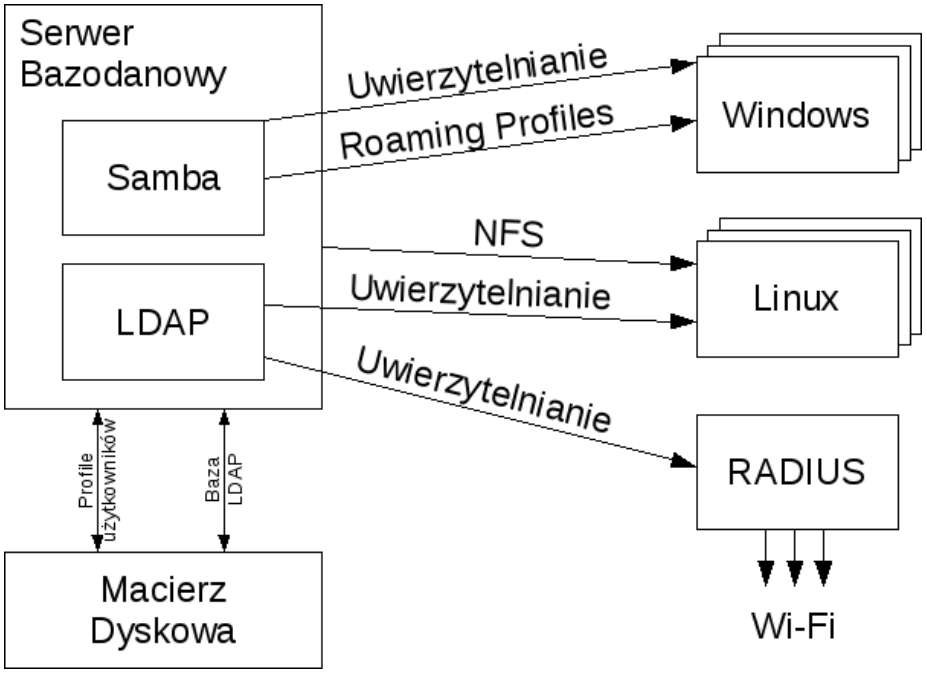
Wymienione powyżej różnice znacznie wpływają na cały proces integracji obu systemów. Dodatkowym problemem jest zamkniętość i komercyjność rozwiązań przeznaczonych dla systemów Windows, co utrudnia ich implementację dla systemów Linux, które z natury preferują rozwiązania otwarte i oparte o uznane standardy.

3 Integracja systemów Linux i Windows w praktyce

Problemy wynikające z różnic sygnalizowane w rozdziale drugim udało się przynajmniej w pewnym stopniu rozwiązać. W tym rozdziale przedstawione zostaną 2 alternatywne rozwiązania – jedno oparte o Samba[10] i LDAP[2] a drugie o Active Directory[3].

3.1 Samba + LDAP

Oryginalnie uwierzytelnianie użytkowników stosowane w laboratorium Katedry ASK oparte zostało o rozwiązania całkowicie otwarte, czyli LDAP oraz Samba. Architektura tego rozwiązania przedstawiona została na Rys. 1.



Rysunek 1: Architektura systemu uwierzytelniającego opartego o Samba i LDAP

Jako centrum uwierzytelniające służy maszyna serwerowa z zainstalowanym systemem Linux. Pod kontrolą tego systemu działają dwie kluczowe aplikacje: serwer LDAP uwierzytelniający klientów linuksowych (zarówno aplikacje jak i usługi, np. serwer RADIUS) i przechowujący wszystkie niezbędne informacje na temat użytkowników i grup, czyli:

- nazwa użytkownika,
- informacje o profilu linuksowym (położenie katalogu domowego, shell),
- informacje o profilu windowsowym (położenie profilu zdalnego, punkt montowania profilu linuksowego),
- identyfikator użytkownika/grupy (windowsowy i linuksowy, przechowywane na osobnych polach),
- hasło użytkownika przechowywane w osobnych polach dla środowiska Linux i dla Windows,
- linuksowy identyfikator domyślnej grupy w przypadku użytkownika
- identyfikatory użytkowników należących do danej grupy w przypadku grup,
- serwer Samba symulujący domenę Windows NT i obsługujący klienty windowsowe.

Katalogi domowe użytkowników przechowywane są na zewnętrznej macierzy dyskowej i dostępne bezpośrednio tylko z serwera uwierzytelniającego. Jako, że jest to system linuksowy możliwe było zastosowanie systemu plików ext3. Dzięki temu wyeliminowany został problem rozróżniania rozmiarów liter w nazwach plików (obsługuje to sam system ext3) czy zakresu dopuszczalnych znaków – system ext3 pozwala na zapisywanie plików z wykorzystaniem dowolnych symboli, obsługuje więc wszystkie nazwy możliwe do utworzenia zarówno w systemach Windows jak i Linux. Niestety system Windows nie będzie w stanie odczytać plików i katalogów, których nazwy zawierają niewłaściwe dla niego znaki.

Serwer uwierzytelniający z kolei udostępnia katalogi domowe stacjom roboczym opartym o Linux za pomocą protokołu NFS[11] a klientom windowsowym za pośrednictwem Samba przez protokół SMB[12]. Oba te systemy nie wprowadzają dodatkowych ograniczeń jeżeli chodzi o dopuszczalne znaki w nazwach. Dodatkowo protokół NFS jest transparentny jeżeli chodzi o uprawnienia użytkowników. W przypadku klientów windowsowych Samba zajmuje się konwersją uprawnień odwzorowując trzy-poziomowy system linuksowy na odpowiednie grupy i użytkowników rozumianych przez system Windows.

Po bliższym przyjrzeniu się opisanemu rozwiązaniu można dojść do wniosku, że mamy tutaj do czynienia z dwoma niezależnymi systemami uwierzytelniania (LDAP dla Linux i Samba dla Windows) przechowującymi dane we wspólnej bazie, jednak na osobnych polach. Zachodzi więc konieczność synchronizacji wszystkich danych przy każdej modyfikacji zawartości bazy. Okazuje się to niestety bardziej skomplikowane niż się wydaje.

Przykładowo system Samba, uwierzytelniający klientów windowsowych, a korzystający z LDAP jako podległego systemu składowania danych jest świadom istnienia wielu pól przechowujących hasło, gdyż sam w LDAP się uwierzytelnia. Możliwe jest więc wymuszenie zapisu hasła na kilka sposobów i tym samym

synchronizacja haseł pomiędzy obiema metodami uwierzytelniania. Niestety odwrotna sytuacja nie jest możliwa. System Linux przy zmianie hasła w LDAP nie jest świadom istnienia klientów windowsowych i w związku z tym, w procesie zmiany hasła, do LDAP przekazywany jest jedynie jego skrót, co uniemożliwia wygenerowanie wpisu, jaki powinien znaleźć się w polach odpowiedzialnych za hasło dla systemów windowsowych. Podobny problem występuje w każdej podobnej sytuacji. Np. aby zablokować konto użytkownika należy zmodyfikować dwie wartości w LDAP – jedną odpowiedzialną za blokowanie konta dla klientów windowsowych, drugą dla klientów linuksowych.

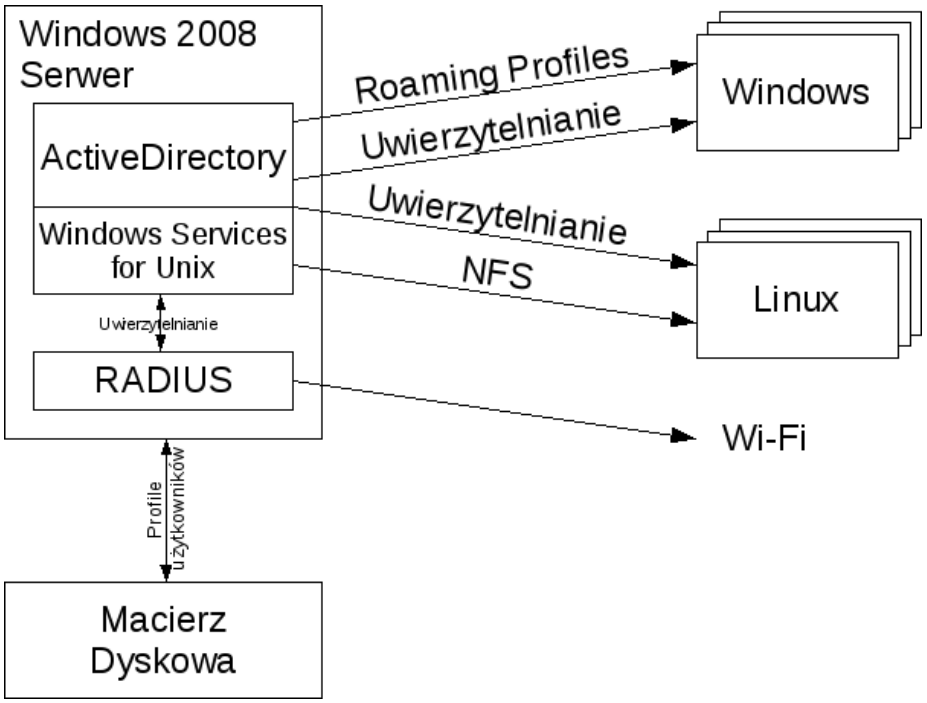
Rozwiązanie tego typu problemów wymaga zastosowania zewnętrznych narzędzi, które nie zawsze nadają się do udostępnienia użytkownikom. Tak było właśnie w przypadku wspomnianej zmiany hasła. LDAP wymaga loginu i hasła, żeby możliwe było zapisywanie w nim danych. Aby zwykły użytkownik mógł zmienić swoje wpisy w bazie danych musi znać login i hasło administratora LDAP lub dane te muszą być zapisane w jakiś sposób w aplikacji. Generuje to niepotrzebne zagrożenia bezpieczeństwa i dodatkowy nakład administracyjny związany z wykonywaniem czynności, które z powodzeniem użytkownicy mogliby wykonywać samodzielnie (jak aktualizacja własnych danych itp.).

Niedogodności te były jednym z powodów odejścia od opisywanego rozwiązania, jednak rzeczywisty problem był o wiele bardziej złożony. Opisany sposób zarządzania użytkownikami spełnia swoje zadanie w przypadku, gdy uwierzytelnianie odbywa się jedynie za pomocą loginu i hasła. W przypadku chęci rozszerzenia rozwiązania o uwierzytelnianie za pomocą kart inteligentnych czy innych nośników informacji napotykamy na dość poważne problemy. Z jednej strony producenci sprzętu koncentrują się jedynie na rozwiązaniach windowsowych całkowicie zaniedbując system Linux jeżeli chodzi o dostępność sterowników czy oprogramowania współpracującego z danymi rozwiązaniami. Generuje to konieczność samodzielnego zaimplementowania obsługi urządzeń i całego procesu autoryzacyjnego ręcznie. Z drugiej strony stosując to rozwiązanie pozbywamy się możliwości wykorzystania gotowych produktów dla Windows, gdyż wymagają one serwera również opartego o Windows. Dodatkowo wprowadzone warstwy pośredniczące (Samba i LDAP) w proces uwierzytelniania klientów windowsowych tworzy potrzebę modyfikacji kodu bardzo wielu systemów co dodatkowo komplikuje cały proces, wprowadzając dodatkowe ryzyko przy wdrożeniu alternatywnych metod uwierzytelniania a być może i uniemożliwiając ich realizację.

3.2 Active Directory

W związku z problemami opisanymi w ostatnich akapitach poprzedniego podrozdziału podjęto decyzję o migracji systemu uwierzytelniania z opartego o Linux, Sambę i LDAP na oparty o Windows 2008 Serwer i domenę Active Directory. Architektura nowego rozwiązania przedstawiona jest na Rys. 2.

To rozwiązanie jest zbliżone do przedstawionego w poprzednim podrozdziale. Jako centrum uwierzytelniające służy tutaj Active Directory. Wywodzi się ono z LDAP, jednak zostało zmodyfikowane i rozbudowane o funkcjonalność specyficzną dla systemów windowsowych przy jednoczesnym usunięciu elementów specyficznych dla środowisk opartych o system Linux. Współpraca z systemami linuksowymi wymaga instalacji Microsoft Windows Services for Unix[13], które od Windows 2003 Serwer dostępne są jako standardowy element systemu. Narzędzia



Rysunek 2: Architektura systemu uwierzytelniającego opartego o Active Directory

te rozszerzają schemat opisu Active Directory o pola niezbędne dla poprawnego funkcjonowania profilu użytkownika w Linux, czyli numeru użytkownika, numeru grupy, domyślnej powłoki systemowej i położenia katalogu domowego.

Hasła, w odróżnieniu od LDAP, przechowywane są w jednym polu. Zarówno użytkownicy Windows jak i Linux uwierzytelniają się poprzez protokół Kerberos[14]. Dzięki spójnemu systemowi uwierzytelniania wyeliminowana została konieczność synchronizacji haseł, uproszczone zarządzanie kontami (wystarczy pojedyncza blokada konta, możliwe jest dopuszczenie by użytkownik logował się tylko na wybranych komputerach, w określonych godzinach itp.). Użytkownicy mogą za pomocą standardowych narzędzi systemowych w dowolnym momencie zmienić swoje hasło, niezależnie od systemu operacyjnego z jakiego korzystają.

Dodatkowe usługi, zwłaszcza te wbudowane w system Windows 2008 Serwer, takie jak serwer RADIUS, bardzo łatwo integrują się z Active Directory niezależnie od tego, czy działają pod systemem Windows czy Linux. Nawet w przypadku, kiedy potrzebny byłby element pośredniczący w postaci pakietu Samba i Winbind nadal wszelkie dane uwierzytelniające znajdują się w jednej bazie danych, nie trzeba ich w żaden sposób synchronizować.

Podobnie jak w rozwiązaniu z podrozdziału pierwszego, dane użytkowników przechowywane są na macierzy dyskowej dostępnej w sposób bezpośredni tylko z poziomu serwera uwierzytelniającego. Dysk macierzy sformatowany jest do systemu plików NTFS natywnie wspieranego przez system Windows. Pliki i katalogi z kolei udostępniane są klientom windowsowym poprzez protokół SMB a linuksowym poprzez NFS. Implementację serwera NFS w wersji 3 zawarta jest

w wspominanych wcześniej Microsoft Windows Services for Unix. W świecie Linux od pewnego czasu króluje już wersja 4 jednak wszystkie systemy linuksowe umożliwiają nadal korzystanie z wersji 3, która jest wystarczająca w zdecydowanej większości zastosowań.

Odmiennie niż w przypadku rozwiązań linuksowych, rozwiązany jest problem odwzorowywania uprawnień dostępu do plików i katalogów. Domyślnie przez NFS pliki udostępniane są z uprawnieniami użytkownika "nobody". Serwer NFS przy każdym dostępie (odczycie atrybutów, odczycie pliku, zapisie danych, wykonanie pliku itp.) do pliku sprawdza rzeczywiste uprawnienia zapisane w tablicy partycji dysku macierzy. Dopiero na tej podstawie dostęp jest udzielany. Brak tutaj pełnej przeźroczystości tego rozwiązania znanej z działania serwera NFS pod Linux.

Serwer NFS, w przeciwieństwie do swojego linuksowego protoplasty, musi też przeprowadzać konwersję znaków użytych w nazwach plików i katalogów, gdyż system Windows, zarządzający dostępem do danych nie jest w stanie obsłużyć niektórych znaków specjalnych, w wyniku czego nie będzie możliwe utworzenie niektórych plików niezbędnych do działania części aplikacji linuksowych. Konieczne jest więc przygotowanie specjalnej tablicy odwzorowującej znaki nieobsługiwane na akceptowalne przez system Windows. Rozwiązanie to ma jednak kilka wad. Możliwe jest niestety tylko odwzorowanie znaku w inny, pojedynczy znak. Nie ma możliwości zastępowania jednego znaku jakimś ciągiem. Użyte z kolei znaki powinny posiadać jak najmniejsze prawdopodobieństwo pojawienia się w nazwach plików po stronie Windows, gdyż każda nazwa pliku zapisanego pod Linuksem zostanie zamieniona na obsługiwaną przez Windows. Niestety zamiana ta jest również przeprowadzana w drugą stronę, niezależnie od tego, czy plik został utworzony pod Windows czy pod Linux. W związku z tym trudne jest stworzenie takiej tablicy odwzorowującej znaki, która nie powodowałaby nieporozumień a generowała rozsądne nazwy.

Opisane w tym podrozdziale rozwiązanie zdecydowanie upraszcza implementacje alternatywnych metod uwierzytelniania użytkowników. Poprzez eliminację zbędnych warstw pośredniczących oraz możliwość wykorzystania gotowych rozwiązań dostarczanych przez producentów sprzętu zdecydowanemu uproszczeniu ulega implementacja dowolnie skomplikowanego systemu. Większość dostępnych na rynku produktów jest już przystosowana do współpracy z systemami Windows i Active Directory. Trudność pozostaje w implementacji rozwiązań dla systemu Linux, jednak wraz ze zacieśnieniem współpracy, prostszym dostępem do Active Directory i wzrostem popularności systemów linuksowych zadanie to staje się coraz prostsze.

4 Uwierzytelnianie w przestrzeni usług

Przestrzeń inteligentna to przestrzeń usług usług wywoływanych zdarzeniami wykonującymi się w tej przestrzeni. Dostęp do usług może wymagać zastosowania wspólnego z systemem operacyjnym, zintegrowanego mechanizmu zabezpieczającego te usługi.

Każdą z usług należy traktować jako niezależny podsystem. Każda z nich może być zlokalizowana na odrębnym fizycznym serwerze, co z kolei implikuje konieczność kontrolowania wszystkich połączeń do każdej z usług[15]. To z kolei wymaga zastosowania swego rodzaju zarządcy bezpieczeństwa, który kontroluje

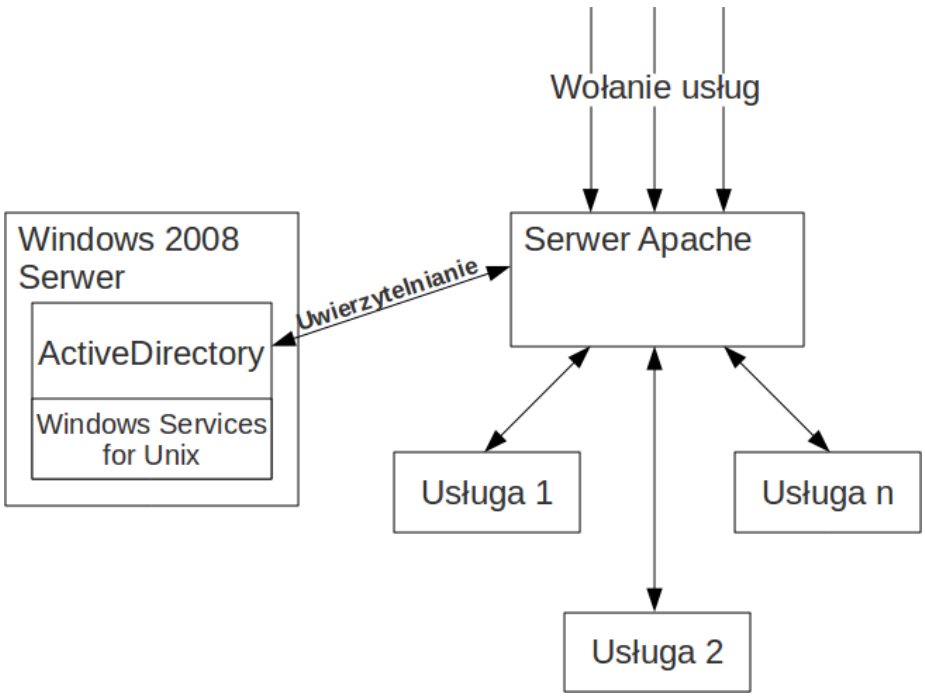
dostęp do każdej usługi niezależnie od źródła pochodzenia żądania. Kontrola taka odbywa się dwustopniowo:

1. weryfikacja klienta – sprawdzane jest czy dany fizyczny klient (np. komputer identyfikowany poprzez adres IP) ma prawo dostępu do wybranej usługi. Dzięki temu eliminowane są połączenia z nieznanych czy niezaufanych segmentów sieci.

2. weryfikacja użytkownika – aby uzyskać dostęp do wybranej usługi i określić poziom tego dostępu użytkownik, który korzysta z usługi z uprawnionego do tego komputera, zobowiązany jest do podania odpowiednich danych uwierzytelniających (np. login i hasło), które weryfikowane są w jednym centralnym centrum uwierzytelniającym.

Aby uzyskać dostęp do wybranej usługi konieczne jest pozytywne uwierzytelnienie się użytkownika na obu wyżej wymienionych poziomach.

W laboratorium Katedry ASK rolę nadzorcy zabezpieczeń pełni serwer Apache[16]. Serwer ten przechwytuje żądania kierowane do dowolnej z usług z dowolnego źródła (Rys. 3). Następnie weryfikuje uprawnienia zarówno maszyny klienckiej (korzystając z swoich wewnętrznych reguł) jak i użytkownika (weryfikując dane uwierzytelniające z przechowywanymi w serwerze Active Directory).



Rysunek 3: Bezpieczeństwo przestrzeni usług

Same usługi zostały przeniesione do niezależnego segmentu sieci, co z kolei uniemożliwia wysłanie żądania z pominięciem nadzorcy zabezpieczeń.

5 Podsumowanie

Ciągły rozwój metod kryptograficznych, pojawianie się coraz to nowszych rozwiązań w dziedzinie zarządzania tożsamością użytkowników powoduje coraz większe problemy przy integracji systemów. Praktycznie z dnia na dzień pojawiają się nowe technologie, które z założenia mają uprościć codzienną pracę, a ostatecznie dodają się do ogólnego chaosu i różnorodności panującej na rynku informatycznym. Przez to stworzenie spójnego i rozszerzalnego rozwiązania umożliwiającego zarządzanie użytkownikami i usługami w środowiskach heterogenicznych jest coraz bardziej skomplikowane, zwłaszcza w pełni opartego na otwartych rozwiązaniach, które z powodu niedostępności dokumentacji nie nadążają z implementacją nowych, często zamkniętych technologii. Jednakże dzięki umowom, takim jak zawarta pomiędzy formami Novell i Microsoft, współpraca ta coraz częściej jest możliwa. Często płacimy za to pewien niewymierny koszt, tracąc część niezależności technologicznej czy wybierając po prostu gorsze rozwiązania. Często jednak koszt ten trzeba ponieść aby możliwe było zbudowanie podstawy do prawdziwej heterogenicznej przestrzeni inteligentnej.

Literatura

[1] Wikipedia, wolna encyklopedia, http://pl.wikipedia.org/wiki/Uwierzytelnianie

[2] The Internet Engineering Task Force (IETF), Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map, http://tools.ietf.org/html/rfc4510

[3] Microsoft TechNet, Active Directory Domain Services, http://technet.microsoft.com/pl-pl/windowsserver/dd448614(en-us).aspx

[4] Katedra Architektur Systemów Komputerowych, Laboratorium komputerowe "Inteligentne usługi i dedykowane systemy internetowe", http://lab527.eti.pg.gda.pl

[5] Wikipedia, wolna encyklopedia, Wi-Fi Protected Access, http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access

[6] The Internet Engineering Task Force (IETF), Remote Authentication Dial In User Service (RADIUS), http://tools.ietf.org/html/rfc2865

[7] Hassell, J., RADIUS - Securing Public Access to Private Resources, O'Reilly & Associates, 2002

[8] Microsoft & Novell Interoperability Collaboration, http://www.microsoft.com/interop/msnovellcollab/default.mspx

[9] Tweedie S., EXT3, Journaling Filesystem, http://olstrans.sourceforge.net/release/OLS2000-ext3/OLS2000-ext3.html

[10] Samba, http://www.samba.org

[11] Sun Microsystems, Inc., NFS: Network File System Protocol Specification, http://tools.ietf.org/html/rfc1094

[12] IBM, Microsoft, Server Message Block (SMB), ftp://ftp.microsoft.com/developr/drg/CIFS/

[13] Microsoft TechNet, Microsoft Windows Services for Unix, http://www.microsoft.com/windows/sfu

[14] MIT, Kerberos: The Network Authentication Protocol, http://web.mit.edu/kerberos/

[15] Krawczyk, H., Lubomski, P., Architektury systemów informatycznych wspomagających rozwój e-uczelni, Zeszyty naukowe wydziału ETI Politechniki Gdańskiej, nr 7, tom 17, s. 143-150, Politechnika Gdańska, Gdańsk 2009

[16] Apache Foundation, Apache HTTP Server, http://httpd.apache.org/