Uniwersytet w Białymstoku

Wydział Matematyczno-Fizyczny

Instytut Matematyki

Paweł Kowalewski

implementacja serwera plików z zastosowaniem ldap nfs i samby

Praca dyplomowa napisana
pod kierunkiem
dr. Mariusza Żynela

Białystok 2005

Składam serdeczne podziękowania
dr. Mariuszowi Żynelowi za wszelką pomoc okazaną
podczas przygotowania niniejszej pracy.

Paweł Kowalewski

Spis treści

Wstęp

Od około 10 lat, czyli od początku ery „dot-com", komputer to dopiero komputer podłączony do sieci teleinformatycznej, dopiero sieć to komputer. W czasie tej ostatniej dekady gwałtownie rozwijały się różne technologie, a wśród nich LDAP, NFS i Samba, które są tematem tej pracy. Wszystkie one mają na celu ułatwienie wymiany danych i zarządzanie danym. Mają podnosić naszą produktywność.

Celem tej pracy jest przygotowanie dedykowanego serwera w sieci lokalnej, który pozwoli użytkownikom przechowywać na nim swoje pliki oraz logować się na swoje konto z dowolnego komputera w tej sieci. Z założenia użytkownik może logować się zarówno z Windows XP jak i Solaris. Na serwerze maja być zachowane nie tylko dane użytkownika, ale także personalne ustawienia pulpitu i aplikacji. Takie rozwiązanie daje dużą swobodę użytkownikom i nie powoduje „przywiązania" użytkownika do konkretnej stacji roboczej.

Centralne umieszczenie danych ułatwi wykonywanie kopii zapasowych danych. Administrator łatwiej będzie mógł również zarządzać tymi danymi, przydzielać miejsce oraz nadawać dostęp do konkretnych usług i aplikacji.

Postawiony w pracy cel można prawdopodobnie uzyskać przy pomocy różnego oprogramowania. W naszym projekcie udostępnianie zasobów dyskowych odbywa się za pomocą NFS i Samby w zależności od tego czy użytkownik loguje się z Solaris czy Windows, natomiast dane o kontach użytkowników przechowywane są w uniwersalnym serwerze usług katalogowych LDAP.

W pracy opisany został sposób instalacji oraz konfiguracji odpowiedniego oprogramowania na serwerze z systemem Solaris 10 x86, tak, aby uzyskać postawione cele. Ponadto, w ramach pracy zostały przygotowane skrypty wspomagające administratora w zarządzaniu tym serwerem.

Rozdział 1

LDAP

1.1 Ogólne informacje o LPAD

1.1.1 Czym jest LDAP

LDAP jest skrótem od Lightweight Directory Access Protocol w dosłownym tłumaczeniu to *Lekki Protokół Dostępu do Katalogu*. Oznacza to prosty, mało obciążający maszynę protokół dostępu do usług katalogowych, szczególnie do tych bazujących na usługach katalogowych X.500. LDAP pracuje w oparciu o protokół TCP/IP lub inne połączeniowe usługi transportu w sieci.

1.1.2 Powiązania i historia

LDAP wywodzi się w prostej linii z X.500. Ta starsza specyfikacja jest w istocie zestawem rekomendacji katalogowych opublikowanych przez ITU (International Telecommunications Union) w listopadzie 1993. Była ona szeroko zalecana przez firmy telekomunikacyjne do tworzenia wzajemnie połączonych ze sobą ogromnych i skomplikowanych serwerów katalogowych. Specyfikacja X.500 jest wielka i dosyć skomplikowana. LDAP wyłonił się jak gdyby z obaw i wątpliwości, czy będzie ją można uprościć i czy będzie dostępna dla przeciętnego użytkownika. Pokrewieństwo między LDAP i X.500 jest bardzo silne. LDAP został pomyślany jako metoda dostępu do katalogów X.500. Nie wymaga wprawdzie, ażeby to był konkretnie katalog X.500, ale używa jego terminów i definicji opisujących katalog. Schemat LDAP/X.500 może się wydać na pierwszy rzut oka nieco zawiły, ale wiele serwerów LDAP ma prosty schemat modyfikacji.

Usługi katalogowe mogą pełnić rolę np. sieciowej książki telefonicznej. Spisane są tam nazwiska, imona, adresy, numery telefonów itd. Dzięki LDAP z książki tej można korzystać, tak jak ze zwykłej książki Telekomunikacji Polskiej przeglądając po nazwach użytkowników, lub możemy wyszukiwać w sieci żądanych usług i zasobów. W dzisiejszych czasach większość dużych instytucje nie mogże poprawnie funkcjonować bez usług katalogowych. Uniwersalne

usługi katalogowe stają się także niezbędne w Internecie i innych sieciach publicznych.

W przypadku typowych dużych (i nie tylko) sieci rozproszonych zasoby i użytkownicy znajdują się w wielu różnych miejscach. Dzięki usługom katalogowym można po zalogowaniu się np. odnaleźć potrzebne zasoby lub innych użytkowników, z którymi chcemy nawiązać kontakt. Nazwy umieszczane są w postaci listy w bazie danych i udostępniane użytkownikom.

Typowa baza danych usług katalogowych ma organizację hierarchiczną. Podstawowym obiektem, występującym w takim drzewie jest jednostka organizacyjna (OU od Organizational Unit). Obiekt typu OU jest katalogiem głównym, zawierającym inne obiekty - zwykle inne jednostki organizacyjne lub obiekty typu liść (leaf objects). Obiekty typu liść nie zawierają w sobie informacji, gdyż same reprezentują rzeczywiste obiekty, takie jak serwery, drukarki albo użytkownicy. Katalogi w drzewie można rozwijać i zwijać, co pozwala na przeglądanie zawartych w nich obiektów typu liść. Administrator może wybrać obiekt, na którym chce przeprowadzić pewne czynności związane z zarządzaniem. Użytkownik może wybrać obiekt, z którego chce skorzystać, lub o którym chce się czegoś dowiedzieć, nie może natomiast niczym zarządzać.

1.1.3 Znaczenie i zastosowanie

LDAP jest dzisiaj ważnym protokołem, motorem napędowym procesu tworzenia katalogu. Jego najmocniejszą stroną jest prostota, symbolizowana "lekkoѕcią" w nazwie. Serwery katalogów — aplikacje dosyć skomplikowane — zostały tak uproszczone, że mogą być obsługiwane niemal przez wszystkich. Kluczowym słowem w LDAP jest podkreѕlona na wstępie "lekkoѕć", odzwierciedlająca niejako pragnienie stworzenia protokołu mocnych usług katalogowych, zdolnego rozwiązać podstawowe problemy, ale jednoczeѕnie wystarczająco nieskomplikowanego technologicznie, ażeby mógł być powszechnie używany. LDAP stał się teraz de facto standardem dostępu do informacji katalogowych. Jest on implementowany w różnych produktach bądź jako system identyfikujący, bądź też system pocztowy albo aplikacja handlu elektronicznego. Do dziѕ pojawiło się na rynku ponad 60 serwerów LDAP. Przypuszczalnie ok. 90% tych produktów to samodzielne serwery LDAP, podczas gdy w pozostałych 10% jest sprzedawane jako składnik zintegrowany z innymi aplikacjami. Druga częѕć nazwy — DAP — mówi o pewnych ograniczeniach, które LDAP stawia producentom oprogramowań. DAP w katalogach X.500 jest protokołem przepytującym katalog. W terminologii minionej olimpiady LDAP to DAP w lżejszej kategorii wagowej. W istocie jest on tylko częścią wielkiego fresku usług katalogowych. LDAP określa, z jakimi aplikacjami - od przeglądarki aż po złożony systemy zarządzania zasobami sieciowymi w przedsiębiorstwie — mogą "porozumiewać się" katalogi. W innych aspektach tworzenia, zarządzania i używania katalogu pozostawia programistom spory margines swobody.

1.1.4 Główne cechy

Jedną z osobliwości LDAP jest jego podobieństwo do bazy danych. Jeśli się zna popularną bazę SQL albo też inną relacyjną bazę danych, to LDAP może się jawić jako środek do otrzymania czegoś w rodzaju prostej bazy danych.Pierwsze wrażenie jest korzystne. LDAP można używać do uzyskiwania informacji ze standardowych baz danych. Udowadniają to m.in. Oracle i Computer Associates w swoich katalogach LDAP. Produkty tych firm stoją jak gdyby na szczycie ich relacyjnych baz danych.

LDAP definiuje operacje, które można przeprowadzać na katalogu — przeszukiwanie, modyfikowanie, dodawanie do niego danych i usuwanie. Wszystkie inne cechy relacyjnej bazy danych, jak na przykład transakcje, nie mają odpowiednika w dziedzinie LDAP. Serwer LDAP nie zapewnia porównywalnej mocy, przynajmniej nie przez interfejs LDAP. Niemniej jednak inwestowanie w taki serwer jest opłacalne. Wraz z LDAP uzyskuje się większą szybkość, niższe koszty, uproszczony model danych i łatwiejsze zarządzanie, a to wszystko przy stosunkowo prostej implementacji. Oprócz określenia metody dostępu do bazy danych, LDAP definiuje także sposób jej implementowania. Z technicznego punktu widzenia nie przypomina to katalogu LDAP. Jednakże LDAP tak silnie wpływa na konstrukcję bazy danych, że niejednokrotnie słyszy się o „katalogu LDAP". Trzeba przy tym zaznaczyć, że katalog zoptymalizowany przez LDAP jest łatwiejszy do zbudowania niż katalog ogólnego przeznaczenia. Gdyby tradycyjna baza danych mogła być dobrze odwzorowana (mapowana) w katalogu LDAP, wtedy nie byłoby nowych relacyjnych bazy danych, lecz bazy hierarchiczne w rodzaju Information Management System firmy IBM. Jest jednak tak, a nie inaczej, ponieważ LDAP został zaprojektowany jako hierarchiczne drzewo informacji odwzorowujące właśnie hierarchiczną strukturę organizacji, którą reprezentuje. Ponadto obejmuje on lokalizacje, takie jak kraj, miasto czy województwo, jednostki organizacyjne, oddziały i — jeszcze niżej — sale konferencyjne czy drukarki.

Duży wpływ na segmentację rynku produktów LDAP ma liczba wpisów do katalogu i tempo zapytań. Jeżeli utrzymuje się portal Web z 10 milionami klientów i ok. 1 milionem wizyt dziennie, wtedy potrzeba "ekstranetowego" lub "e—handlowego" serwera LDAP. W tego rodzaju produkcie użytkownik ma możliwość tworzenia szczególnego typu zapytań — na przykład o identyfikator ID. Rzeczywistość jest taka, że o zoptymalizowaniu katalogu dla wszystkich aplikacji nie może być mowy. iPlanet (dawniej Netscape), który chce utrzymać status pierwszego producenta katalogów ekstranet/handel elektroniczny, może załadować katalog z milionem wpisów w niespełna godzinę. Jest to niemal zwyczajowa operacja w katalogach LDAP — lepiej powtórnie załadować katalog, po wcześniejszym usunięciu z niego danych, niż przechowywać te dane i aktualizować je każdego dnia. Podobnie LDAP jest używany przez wiele aplikacji katalogowych, jak serwery identyfikujące i bramy pocztowe, ale są one "nastrajane" na konkretne API. Jak to pokazały testy porównawcze, produk-

ty różnią się nieraz bardzo poważnie wsparciem standardu LDAP. Jeśli więc nie ma się dostępu do kodu źródłowego, to łączenie różnych produktów tego protokołu może okazać się niemożliwe.

1.2 Uruchomienie serwera LDAP

W naszej pracy przedstawimy instalację całej usługi LDAP począwszy od instalacji serwerów po konfgurację klienta, pokażemy też podstawowe operacje na kliencie takie jak:

- tworzenie nowego użytkownika;
- modyfikacja danych o użytkowniku;
- usuwanie użytkownika z bazy danych

Wszystkie operacje tu pokazane zostały wykonane i przetestowane na komputerach laboratorium Instytutu Matematyki Uniwersytetu w Białymstoku. Używaliśmy komputerów PC z zainstalowanym systemem Solaris 10.

1.2.1 Instalacja serwera LDAP

Aby zainstalować serwer LDAP należy najpierw zaopatrzyć się w niezbędne pakiety, które dostarczone są wraz z systemem operacyjnym Solaris. Przede wszystkim musimy być zalogowani jako `root`. Następnie montujemy IV płytę instalacyjną Solaris 10 i przechodzimy do katalogu

```
/cdrom/sol_10_305_x86_4/Solaris_10/Product
```

po czym musimy dokonać instalacji odpowiednich pakietów w ustalonej kolejności. Przy każdej instalacji system wyświetli nam krótką charakterystykę i kroki instalacji m.in.

- nazwę pakietu;
- odczytanie informacje o pakiecie;
- sprawdzenie ilość dostępnego miejsca na instalację;
- kontrolę potencjalnego konfliktu z innymi wcześniej zainstalowanymi pakietami;
- sprawdzonie kolejności pakietów wymaganych do instalacji;
- wskazanie ilości już zainstalowanych składników;

Przy każdym dodawaniu nowego składnika system zapyta się czy chcemy kontynuować instalację wpisujemy: [y]- tak, [n]- nie lub [h]- krótki opis. Aby poprawnie zainstalować oprogramowanie serwera LDAP należy postępować zgodnie z krokami podanymi poniżej.

W terminalu, jako `root`, należy wpisac

```
pkgadd -d . nazwa_pakietu
```

gdzie `nazwa_pakietu` to kolejno:

1. `IPLTnls` (Nationalization Languages and Localization Support)— pakiet wspierający lokalizację oraz języki.
2. `IPLTnspr` (Portable Runtime Interface)— pakiet służący do łączenia się.
3. `IPLTnss` (Network Security Services)— pakiet zabezpieczający sieć.
4. `IPLTjss` (Processing package instance)— pakiet bezpieczeństwa dla Javy.
5. `IPLTpldap` (PerLDAP)— narzędzie, posiadające biblioteki Perla zwiększające efektywność i wypełniające działania LDAP.
6. `IPLTdsr` (Directory Server (root))— usługa katalogowa, instalowana w texttt/.
7. `IPLTdsu` (Directory Server (usr))— tworzenie usługi katalogowej dla użytkownika.
8. `IPLTadmin` (Administration Server)— usługi umożliwiające zarządzenie katalogami.
9. `IPLTcons` (Console Client Base)— tworzenie konsoli dla klienta.
10. `IPLTadcon` (Administration Server Console)— tworzenie konsoli administracyjnej.
11. `IPLTdscon` (Directory Server Console)— zarządzanie katalogami LDAP.
12. `IPLTadman` (Administration Server Documentation)— dokumentacja serwera administracyjnego.
13. `IPLTdsman` (Directory Server Documentation)— dokumentacja serwera usług katalogowych.

W ten sposób zakończyliśmy instalacje pakietów niezbędnych do uruchomienia serwera LDAP. Następnym krok to konfiguracja. Jednak zanim zaczniemy należy upewnić się, że nasz komputer (host) posiada nazwę domenową. Jeśli nie, to w tym celu plik `/etc/hosts` zaopatrujemy we wpis:

```
IP nazwa_hosta FQDN,
```

gdzie FQDN, to *Fully Qualified Domain Name*, czyli pełną nazwę domenową. Następnie wykonujemy polecenia:

```
/usr/bin/domainname domena.tld,
```

gdzie `domena.tld` to nazwa naszej domeny, a skrót `tld` to *top level domain* (w naszym przypadku jest to `im.uwb.edu.pl` i ten adres będziemy używać) oraz

```
/usr/bin/domainname > /etc/defaultdomain
```

1.2.2 Konfiguracja iPlanet Server Products

Po zainstalowaniu niezbędnego oprogramowania możemy przystąpić do konfiguracji. Należy w dalszym ciągu być zalogowanym jako `root` Wpisujemy w terminalu:

```
/usr/sbin/directoryserver setup
```

Krok 1. Po uruchomieniu programu ukazuje się okno powitalne oznajmiające, że właśnie rozpoczynamy konfigurację `"iPlanet Server Products"` i `"iPlanet Console"` na naszym komputerze. Podczas instalacji możemy:

- Wcisnąć [`enter`] w celu zaakceptowania proponowanej opcji;
- W każdym momencie instalacji możemy cofnąć się o krok używając kombinacji klawiszy 'Ctrl+B';
- Wcisnąć 'Ctrl+C', aby anulować konfigurację w każdym momencie;

Pierwsze zapytanie dotyczy chęci kontynuacji, akceptujemy zapytanie i rozpoczynamy proces instalacji.

Krok 2. Po pierwsze wybieramy, co chcemy skonfigurować i uruchomić. Możemy wybrać:

1. `iPlanet Servers.` Konfiguracja serwera iPlanet i zintegrowanej konsoli iPlanet
2. `iPlanet Console.` Samodzielna konsola jako aplikacja Javy .

W naszej konfiguracji wybieramy opcję [1].

Krok 3. Następnie wybieramy typ konfiguracji. I tak:

1. `Express Configuration` (konfiguracja ekspresowa). Szybka konfiguracja używająca najpopularniejszych opcji, przydatna, jeśli chcemy szybko uruchomić serwer .
2. `Typical Configuration` (konfiguracja typowa). Konfiguracja pozwalająca na wprowadzenie, swoich, niewielkich zmian.
3. `Custom Configuration` (konfiguracja eksperta). Samodzielnie ustawiamy większość opcji. Przeznaczone dla specjalistów i aministratorów.

Przeprowadzamy typową konfigurację, czyli zatwierdzamy opcję `[2]`.

Krok 4. Wybór składników:

1. `iPlanet Directory Suite`
2. `Administration Services`

Każdy ze składników posiada „podskładniki". Chcemy zainstalować wszystko, wpisujemy więc [`All`] i zatwierdzamy.

Krok 5. Składniki iPlanet Directory Suite:

1. `iPlanet Directory Server.`
2. `iPlanet Directory Server Console.`

Wybieramy [`1,2`], co oznacza wybór obydwu elementów.

Krok 6. Składniki Administration Services:

1. `iPlanet Administration Server.`
2. `Administration Server Console.`

Podobnie wybieramy [`1,2`].

Krok 7. W tym kroku należy wprowadzić domenową nazwę komputera, na którym konfigurujemy oprogramowanie serwera. Używamy schematu:

```
nazwa_hosta.im.uwb.edu.pl
```

Nie może to być „zmyślona" nazwa, musi to być domenowa nazwa naszego hosta, która jest określona w pliku `/etc/hosts`.

Krok 8. Teraz wybieramy użytkownika i grupę dla serwera iPlanet, który bkedzie uruchamiany z prawami tego użytkownika i grupy. Zaleca się, aby temu użytkownikowi i grupie nie przydzielać żadnych praw w systemie. Serwer administracyjny (*Administration Server* sam przydzieli pewne przywileje tej grupie użytkowników umożliwiając wykonywanie pewnych, specjalnych zadań. Należy zadbać, aby w systemie istniał wybrany użytkownik i grupa. Możemy wybrać sugerowaną przez instalatora nazwę lub wprowadzić własną.

W tej instalacji wykorzystajmy domyślne ustawienia [`nobody`] dla użytkownika i grupy. którzy, istnieją już w systemie i nie mają żadnych wyjątkowych uprawnień.

Krok 9. Informacje o serwerze iPlanet są przetrzymywane w katalogu konfiguracyjnym iPlanet, który być może już wcześniej został stworzony. Jeżeli tak to należy skonfigurować serwer, aby mógł być zarządzany przez konfigurację serwera. Aby to zrobić trzeba podać jego nazwę domenową, port, sufiks, DN, dane root'a.

Jednak my chcemy, aby to był samodzielny i nowy serwer, wiec wybieramy opcję [`no`].

Krok 10. Jeżeli posiadamy już serwer katalogów na tym komputerze to możemy teraz zrobić jego kopię zapasową, zawierającą takie dane jak: użytkownicy, grupy, itp. Wybierając [`yes`] trzeba podać nazwę hosta, port, sufiks, DN serwera, z którego robimy kopię. My konfigurujemy nasz pierwszy serwer, więc wybieramy [`no`].

Krok 11. W tym kroku musimy zdefiniować port `TCP`, przez który będziemy łączyć się. Standardowo jest to port `389` i tak zostawiamy.

Krok 12. Każdy serwer katalogów musi mieć unikalną nazwę, wybierzmy proponowaną lub wprowadźmy inną.

Krok 13. Teraz należy wprowadzić identyfikator administratora do konfiguracji serwera katalogów, oraz wpisać hasło.

My wybieramy standardowo

```
ID:              admin
hasło:           abcd1234
ponownie hasło:  abcd1234
```

Krok 14. Sufiks to jest korzeń naszego drzewa katalogów. Wpisujemy swoją nazwę lub (tak jak my) zatwierdzamy proponowane przez system:

```
Suffix [dc=im, dc=uwb, dc=edu, dc=pl].
```

Krok 15. Pewne działania na serwerze iPlanet wymagają konta administratora, nazwanego tutaj `,,Directory Manager"`. Możemy wprowadzić swoją nazwę lub potwierdzić proponowaną. My wybieramy nazwę domyślną `cn= Directory Manager` i wprowadzamy hasło `[abcd1234]`.

Krok 16. Domena administracyjna to część konfigurowanego serwera katalogowego używana do przechowywania informacji o oprogramowaniu iPlanet. Jeżeli zarządza się kilkoma programami w tym samym czasie, albo zarządza się informacjami różnych domen, można używać domen administracyjnych, aby utrzymywać dane oddzielnie. Jeżeli nie używa się kilku domen administracyjnych, należy wciśnąć [`enter`], aby wybrać ustawienie domyślne. W przeciwnym wypadku trzeba wprowadzić unikalne opisy dla odpowiednich domen.

Krok 17. Nasz serwer administracyjny jest odizolowany od innych aplikacji serwerowych za pomocą portu, dzięki któremu będziemy mieli bezpieczny i niezakłócony dostęp do jego zasobów. Należy wprowadzić numer portu zawierający się między wartościami `1024` a `65535` po to, aby serwer działał poprawnie. Nie należy używać numeru portu, który już gdzieś wcześniej został przydzielony.

Domyślny port to losowo wybrany numer niezajętego portu w systemie.

Krok 18. Administracja serwera jest możliwa dzięki pewnemu użytkownikowi w systemie. Ten użytkownik będzie miał inne uprawnienia niż jakikolwiek inny użytkownik korzystający z serwera. Tylko on będzie miał możliwość zapisywania konfiguracji. Jeżeli zalogujemy się na serwerze administracyjnym jako `root` będziemy mógli uruchamiać i wyłączać odpowiednie aplikacje na serwerze.

W tym oknie musimy wskazać nazwę użytkownika, który będzie miał takie przywileje. Domyślnie jest on nazwany `root` i taką nazwę wybieramy.

Po zatwierdzeniu nazwy `root‘a` początkowa konfiguracja serwera kończy się. System będzie teraz analizował dane, które wprowadziliśmy i spróbuje uruchomić serwer. Na ekranie pojawia się podsumowanie ustawień jakich wcześniej dokonaliśmy. Teraz uruchamiając polecenie

```
/usr/sbin/directoryserver startconsole
```

możemy rozpocząć zarządzanie serwerem.

1.3 Przygotownie iPlanet Directory Server do pracy

Po instalacji i początkowej konfiguracji serwera zajmijmy się właściwym ustawianiem `iPlanet Directory Server`, aby dysponował danymi i służył klientom LDAP. Aby rozpocząć proces należy uruchomić program z lokalizacji:

```
/usr/lib/ldap/idsconfig
```

Jeżeli będziemy w przyszłości uruchamiali `idsconfig` to możemy posłużyć się wygenerowanym plikiem konfiguracyjnym, aby pominąć zapytania systemu. Taki plik zostanie utworzony, jeśli uruchomimy interaktywny program `idsconfig` z opcją

```
-o nazwa_pliku_konfiguracyjnego.
```

Gdy uruchomimy go, będziemy musieli odpowiedzieć na kilka zapytań. Poniżej znajduje się przykładowy dialog:

Zaleca się wykonanie kopii zapasowej zanim zaczniemy coś zmieniać, aby uniknąć problemów. W każdym momencie konfiguracji możemy wcisnąć kombinację klawiszy Ctrl+C, aby przerwać konfigurację. Przy każdym pytaniu możemy zobaczyć krótką pomoc (wciskając `h` i zatwierdzając klawiszem `enter`).

1. W pierwszej kolejności system zapyta się nas czy chcemy rozpocząć konfigurację.

2. Musimy wprowadzić nazwę hosta, który chcemy skonfigurować do pracy z iDS. Będziemy konfigurować hosta o nazwie `lab-b05`.

3. Musimy wprowadzić port, przez który będziemy łączyć się. Jest to ten sam port, który podaliśmy przy konfiguracji serwera katalogów, czyli `389`.

4. W tym kroku należy wprowadzić nazwę i hasło zarządzającego katalogami(DN), wprowadziliśmy nazwę `cn=Directory Manager` oraz hasło `abcd1234`.

5. `(Domain to serve)`. Następnie wprowadzamy nazwę domenową serwera LDAP. W naszym wypadku konfiguracji jest to `im.uwb.edu.pl`)

6. Wprowadzamy domyślną lokalizację drzewa katalogów. Akceptujemy domyślną nazwę ustaloną przy konfiguracji serwera. Nazwa jest postaci `(dc=im,dc=uwb,dc=edu,dc=pl)`. Nazwa ta musi być znana serwerowi LDAP.

7. `(Profile name)`. Teraz wprowadzamy nazwę profilu konfiguracji klientów. Serwer katalogów może zgromadzić różne profile dla kilku grup klientów. Narzędzie inicjalizacji klienta
(`ldapclient`) może przyjąć wartość `default`, jeżeli innej nie podamy.

8. Musimy wprowadzić adresy IP serwera katalogów (domyślnego *Default Server List-* nie jest wymagane oraz preferowanego *Preferred Server List-*wymagane), z którego usług będziemy korzystać.

9. `(Default Search Scope)`. Teraz wybieramy pożądany zakres wyszukiwań danych w drzewie katalogów. Może to być opcja `one` [szukanie w głównym katalogu], lub wersję `sub` [szukanie przez wszystkie poziomy drzewa]. Wybierzmy opcję `one`.

10. `(Credential Level)`. Kolejno wybieramy metodę autoryzacji użytkownika. Wybieramy `anonymous`. Jeżeli wybierzemy `proxy`, wtedy zostaniemy poproszeni o nazwę `DN` oraz hasło.

11. `(Enable Follow Referrals)`. Kolejna opcja pozwala nam na to, aby klienci podążali za odniesieniami LDAP podczas wyszukiwania.

12. `(iDS Time Limit)`. Istnieje możliwość ustalenia maksymalnej ilości czasu, jaki serwer spędzi na pytaniu klienta przed rozłączeniem go. Wartość `-1` wskazuje na brak ograniczenia czasowego. Zezygnujmy z tej możliwości.

13. `(iDS Size Limit)`. Możemy także ustawić maksymalną ilość zapytań klienta, tego też nie zmieniamy i zostawiamy opcję [`n`]. Wartość `-1` oznacza brak limitu.

14. `(Enable crypt password storage)`. Następnie możemy żądać specjalnego przechowywania naszego hasła.

15. `(Authentication Method)`. Istnieje także możliwość specjalnej autoryzacji naszego klienta przez serwer w celu bezpieczniejszego przechowywania i dostępu do danych. Wybieramy opcję [`n`].

16. `(Search Time Limit)`. Ustawiamy termin przeszukiwania przez klienta katalogów na np. `30` sekund.

17. `(Profile Time to Live)`. W celach bezpieczeństwa wprowadzona została opcja długości sesji dla klienta, oznacza ona, że po danym czasie zostaje na nowo odświeżona konfiguracja klienta, co zmniejsza ryzyko, że zapominając się wylogować z serwera i opuszczając komputer ktoś inny zobaczy nasze dane.

18. `(Bind Limit)`. Kolejna wartość kontroluje czas reakcji klienta w sytuacji, gdy serwer jest nagle nieosiągalny. Wartość `10` jest optymalna dla pracy. Jest to coś podobnego do ustawiania czasu w sieci TCP, lecz nasze ustawienie odnosi się tylko do LDAP'a.

19. `(Service Search Descriptors Menu)`. Używając `SSD` możemy odrzucać lub zmieniać domyślne ustawienia dostarczanych usług. SSD może odrzucić lub zmieniać domyślne przeszukiwanie katalogu głównego (DN), zakres wyszukiwań oraz rodzaj filtrów wyszukiwania. SSD współpracuje ze wszystkimi usługami zdefiniowanymi w `nsswitch.conf`. Przed włączeniem SSD wskazane jest zapoznanie się z dokumentacją. Tej opcji także nie włączamy.

Po odpowiedzi na wszystkie pytania dostajemy tabelę z opisem naszej konfiguracji. Są tu zestawione wszystkie dane, które wprowadziliśmy.

```
            Nazwa usługi      Odpowiedź użytkownika

 1. Domain to serve                 :im.uwb.edu.pl
 2. Base DN to setup                :dc=lab-b05,dc=im,dc=uwb,dc=edu,dc=pl
 3. Profile name to create          :default
 4. Default Server List             :82.139.172.69
 5. Preferred Server List           :
 6. Default Search Scope            :one
 7. Credential Level                :anonymous
 8. Authentication Method           :
 9. Enable Follow Referrals         :FALSE
10. iDS Time Limit                  :
11. iDS Size Limit                  :
12. Enable crypt password storage   :FALSE
13. Service Auth Method pam_ldap    :
14. Service Auth Method keyserv     :
15. Service Auth Method passwd-cmd  :
16. Search Time Limit               :30
17. Profile Time to Live            :43200
18. Bind Limit                      :10
19. Service Search Descriptors Menu :
```

Po zakończonej konfiguracji zatwierdzamy zmiany, bądź wprowadzamy poprawki i w ten sposób kończymy konfigurację.

Rozdział 2

NFS

2.1 Udziały NFS

Podstawowy cel, jaki przed nami stoi w tej pracy, to umożliwienie użytkownikom sieci lokalnej na dostęp do swoich plików i folderów z dowolnej stacji roboczej, zarówno spod systemu Solaris jak i Windows. W tym rozdziale opisujemy mechanizm, który pozwala udostępniać i montować pliki i foldery pomiędzy maszynami z zainstalowanym systemem Unix (Solaris, Linux lub FreeBSD). Nazywa się go NFS, czyli Network File System.

Jak sama nazwa sugeruje, NFS jest to specjalny system plików i oprogramowanie, które pozwala na jednym komputerze – serwerze – udostępnić folder, a na innym – kliencie – zamontować go, tak jakby był to zwykły folder na dysku lokalnym komputera. Różnica jest tylko w czasie dostępu do danych, ale przy wydajnej sieci i serwerze różnica ta może przemawiać na korzyść NFS.

Dzięki zastosowaniu serwera plików, bo tak na ogól nazywa się komputer udostępniający swoje zasoby dyskowe, stacja robocza wymaga dysków o mniejszej pojemności, gdyż dane pobierane i zapisywane są przez sieć. Dodatkowo dzięki `NFS` możemy zredukować liczbę napędów `CD-ROM`, dysków elastycznych itp. poprzez współdzielenie tych urządzeń w sieci. Z powyższych zalet wynika jeszcze jedna, że koszty przechowywania i składowania danych znacznie obniżają się. Zastosowanie centralnego „magazynu danych” ułatwia wykonywanie kopii zapasowych (ang. backup) i zarządzanie danymi.

Zaskakujące dla wielu użytkowników (szczególnie tych, którzy nigdy nie pracowali na Unix) może być fakt, że klient serwera NFS nie musi wcale posiadać dysku twardego aby wystartować i uruchamiać aplikacje. W takiej sytuacji obraz systemu operacyjnego jak również wszystkie programy oraz ich dane pobierane są przez sieć.

2.2 Konfiguracja serwera NFS

Ważną cechą NFS jest brak potrzeby zakładania katalogu domowego na każdym komputerze, na którym pracujemy. Wystarczy, że katalog `/export/home` udostępni się na serwerze, a wtedy do katalogu domowego mamy dostęp przez sieć. W Solaris obecny jest dodatkowo mechanizm, zwany *automounter*, który, jak sama nazwa sugeruje, automatyzuje montowanie zasobów sieciowych NFS. Jeśli w plikach `/etc/auto_master` i `/etc/auto_home` na kliencie znajdują się odpowiednie wpisy, to folder domowy użytkownika, zostanie zamontowany samoczynnie, gdy tylko nastąpi odwołanie do niego. Wiersz

```
/home auto_home   -nobrowse
```

w pliku `/etc/auto_master` razem z wpisem

```
*       lab-b05:/export/home/&
```

mówi, że foldery domowe użytkowników należy montować z komputera o nazwie `lab-b05` i znajdują się one tam w folderze `/export/home`. Za znak `&` zostanie podstawiona nazwa użytkownika. Folder domowy zostanie zamontowany na kliencie w folderze `/home`. Tak więc np.: `/export/home/janek` z `lab-b05` będzie dostępny jako `/home/janek`.

Aby jednak jakiekolwiek montowanie zasobów NFS na klienci było możliwe, wcześniej należy odpowiednie zasoby udostępnić. Aby udostępnić folder `/export/home` wystarczy jako `root` wpisać polecenie:

```
share /export/home
```

Udział będzie dostępny aż do restartu serwera NFS. Aby udział był dostępny zawsze, bez konieczności wołania `share` po każdym starcie komputera, należy dostosować plik `/etc/dfs/dfstab`. Składnia tego pliku jest identyczna jak składania polecenie `share` i wygląda następująco:

```
share [-F nfs] [-o opcje] [-d opis] pełna_ścieżka_udziału
```

gdzie

`-F nfs` wskazuje, że folder zostanie udostępniony jako NFS;

`-o opcje` to szczegółowe opcje dotyczące udziału, podając np. `-o ro` spowoduje, że udział będzie dostępny tylko do odczytu;

`-d opis` komentarz do udziału;

`pełna_ścieżka_udziału` pełna ścieżka do folderu, który chcemy udostępnić.

W przypadku naszego serwera plików, który konfigurujemy, dodajemy do `/etc/dfs/dfstab` wiersz:

```
share -d "User home directories" /export/home
```

Następnie wołamy

```
shareall
```

co spowoduje, że wszystkie udziały z pliku `/etc/dfs/dfstab` zostaną udostępnione. Sprawdzamy jeszcze jak jest ustawiona zmienna środowiskowa `NFSMAPID_DOMAIN` w pliku `/etc/default/nfs`. Aby po zamontowaniu katalogu użytkownika na kliencie, prawidłowo były odwzorowane właściciel, grupa i prawa dostępu, zmienna ta musi być tak samo ustawiona na serwerze jak i kliencie. Aby uniknąć nieporozumień, na serwerze wpisujemy

```
NFSMAPID_DOMAIN=im.uwb.edu.pl
```

później to samo wpiszemy na klientach.

Rozdział 3

Samba

3.1 Ogólna charakterystyka

Komputery w niemal każdej sieci lokalnej muszą posiadać możliwość łatwego przesyłania plików między sobą i drukowania dokumentów na zdalnych drukarkach. Wykonywanie takich operacji pomiędzy różnymi rodzinami systemów operacyjnych wymaga używania jednolitego protokołu komunikacyjnego. Samba jest właśnie takim pakietem narzędzi służącym do udostępniania zasobów w sieci, np. drukarek, plików. Dzięki niej można porozumiewać się pomiędzy różnymi systemami, w szczególności Unixowy komputer z Sambą może udawać serwer Windows i udostępniać następujące usługi:

- udostępnianie zasobów plikowych;
- udostępnianie drukarek;
- autoryzacja za pośrednictwem kontrolera domeny tak jak w Windows NT 4.0 Server i późniejszych;
- autoryzacja za pośrednictwem usługi `Active Directory` tak jak w Windows 2000/2003 Server i późniejszych;

Najważniejszymi cechami Samby są:

- proste udostępnianie plików (Windows: 3.1, 95, 98, Me, XP Home);
- zaawansowane udostępnianie plików (Windows: NT 4.0, 2000, XP Professional, 2003 Server);
- udostępnianie drukarek w sieci innym komputerom;
- współdziałanie serwera plików z kontrolerem domeny (Windows NT 4.0 Server);
- działanie w roli kontrolera domeny (Windows NT 4.0 Server);

Głównymi składnikami pakietu Samba są Unixowe demony, które zarządzają udziałami sieciowymi:

- `SMBD` umożliwia współdzielenie plików i drukarek w sieci SMB i zapewnia uwierzytelnienie;
- `NMBD` świadczy usługi Windows Internet Service i wspomaga przeglądanie zasobów sieci.

Korzystanie z Samby przynosi wiele korzyści m.in.:

- nie trzeba płacić dużych sum pieniędzy za serwer Windows, Samba jest darmowa i spełnia wszystkie funkcje serwera Windows.
- między komputerami z różnymi systemami istnieje niezakłucony przepływ danych, możliwe jest używanie urządzeń przez sieć niezależnie od systemu.

Systemy z serii Windows do komunikacji między sobą w zakresie udostępniania plików, drukarek i innych pokrewnych usług używają protokołu sieciowego SMB. Nazywamy to Microsoft Network, zaś potocznie częściej używa się nazwy `otoczenie sieciowe`. W komputerach z systemem Unix jest to sieć SMB. Szeroki wachlarz możliwości Samby pozwala na bardzo elastyczne dopasowywanie pakietu do potrzeb. W jednej sieci można umieszczać komputery z uruchomionym pakietem Samba jako serwer bądź jako klient. Duże znaczenie ma również fakt, że Samba nie ma ograniczenia ilości jednoczesnych połączeń, jak to jest w przypadku np.: Windows 2000 Professional (maksymalnie 10). Pod tym względem można ją porównywać jedynie do serwerowych wersji systemów Microsoftu. Samba udostępnia niemal wszystkie rodzaje zabezpieczeń oferowanych przez systemy Microsoftu, takie jak:

- szyfrowanie haseł;
- uwierzytelnianie przez kontroler domeny;
- restrykcje nakładane na zasoby;

Dodatkowo możemy korzystać z zabezpieczeń oferowanych przez systemy Unixowe, które nie są dostępne w Windows:

- konta użytkowników;
- system uprawnień plików;
- przydziały dyskowe.

Możemy wzbogacić sieć SMB o szyfrowanie całej transmisji danych przy pomocy protokołu SSL Te wszystkie cechy przemawiają na korzyść Samby zwłaszcza na tle systemów całkowicie nie przystosowanych do ochrony danych: Windows: 3.1, 95, 98, Me. W przypadku systemów Unixowych praktycznie nie istnieje problem wirusów. Spore kłopoty z napisaniem skutecznego wirusa dla Unix oraz całkowita odporność na Windowsowe wirusy, zapewniają mniejsze ryzyko utraty danych. Programy antywirusowe powstałe dla rodziny Unixa mają za zadanie jedynie kontrolować pliki wymieniane między systemami operacyjnymi Unix a Windows. Wiele z tych programów może współpracować z Sambą, np.: `Mks-Vir, Sophos, NOD-32`. Samba pracuje na wielu systemach z rodziny Unix, w tym na systemach o wolnym kodzie źródłowym: Solaris, GNU/Linux i BSD. Niskie wymagania sprzętowe, którymi cechują się zarówno wymienione systemy operacyjne jak i Samba, pozwalają na wykorzystanie mniej wydajnych maszyn niż w przypadku systemów Microsoftu. Także całkowity koszt utrzymania serwera plików z Unix jest dużo niższy niż Windows.

3.2 Możliwe zastosowania Samby

Istnieje możliwość łatwego wprowadzenia systemów Unixowych do sieci opartej na systemach typu Windows. Unixowa stacja robocza może zarówno korzystać z zasobów sieciowych jak i je udostępniać innym komputerom. Daje to możliwość stopniowego wprowadzania innych systemów operacyjnych i testowania ich w warunkach roboczych.

Ważną sprawą jest także możliwość zastąpienia Sambą istniejącego serwera plików czy też drukarek, opartego na domowych systemach Windows (3.1, 95, 98, Me). Pozwola to znaczne podniesiesć stabilność pracy całej sieci, bezpieczeństwo przechowywania danych oraz kontroli dostępu. Wnioskiem z powyższego jest podniesienie wydajności całego systemu oraz sieci. Dodatkowym atutem Samby jest możliwość prowadzenia bardzo dużej ilości aktywnych połączeń w jednym momencie co sprzyja rozrastającej się sieci, przy równoczesnym oszczędzeniu środków finansowych.

Rozrastająca się z czasem sieć wymaga centralnego uwierzytelniania użytkowników oraz centralnego udostępniania zasobów. SMB daje nam taką możliwość za pomocą oprogramowania zwanego kontrolerem domeny. Jego zadanie polega na centralnym zarządzaniu zasobami, użytkownikami i ich uwierzytelnianiem. Taką rolą spełnia oczywiście pakiet Samba.

3.3 Praktyczna realizacja

Poniżej przedstawiamy konkretny przykład konfiguracji Samby na komputerze lab-b05 w laboratorium IM. Zadanie konfiguracji było jednym z celów tej pracy

dyplomowej, a jednocześnie stanowi jeden z etapów przygotowania serwera plików. Nasz serwer Samby będzie pełnił rolę kontrolera domeny Windows. Ma to zagwarantować duży komfort pracy użytkownikom. Niezależnie od tego na jakim komputerze będą oni logowali, zawsze bądą mieli swój prywatny pulpit, ze swoim zestawem ikonek, plikami i katalogami. Realizowane jest to za pomocą mechanizmu zwanego *profile przechodnie*.

Poniższe instrukcje pochodzą z pliku konfiguracyjnego

```
/etc/sfw/smb.conf
```

Jego pełna treść znajduje się w dodatku na stronie 48.

Plik konfiguracyjny Samby składa się z sekcji, których nazwy umieszcza się w nawiasach kwadratowych:

```
[nazwa_sekcji]
```

Każda sekcja zawiera zestaw dyrektyw postaci:

```
opcja = wartość
```

Jeżeli dana opcja ma kilka wartości, oddzielamy je znakiem spacji. Poszczególne opcje umieszczone są w osobnych liniach. Komentarze w pliku rozpoczynają się znakiem `#` lub `;`. Ponieważ opcji konfiguracji Samby jest bardzo dużo, ograniczamy się tutaj do tych opcji, które występują w opisywanym pliku `/etc/sfw/smb.conf`

Pierwszym parametrem w sekcji `[global]`, dotyczącej globalnych ustawień serwera, jest

```
workgroup = im.uwb.edu.pl,
```

który określa grupę roboczą, w której serwer będzie rozsyłał informacje o sobie. Klienci, którzy chcą korzystać z udziałów tego właśnie serwera powinni być w tej samej grupie roboczej co serwer. W naszym przypadku grupą roboczą jest `im.uwb.edu.pl`. Linijka

```
server string = Samba Server
```

jest opisem, który widać w otoczeniu sieciowym.

O tym, jak duży ma być plik dziennika Samby mówi nam opcja

```
max log size = 50
```

Wartość podana jest w kilobajtach. Kiedy dane w pliku przekroczą zadaną wartość zostaje zmienione rozszerzenie pliku na `.old` i stworzony zostanie nowy plik dziennika. Kolejne przekroczenie danej wielkości powoduje zastąpienie pliku `.old` nowym.

Opcja

```
log level = 0
```

określa ilość rejestrowanych danych. Domyślnie jest to wartość 0 lub 1, ale jeżeli chcemy w przyszłości łatwo rozwiązać potencjalne problemy zalecana jest wartość 3. Wyższe wartości są niezalecane, ponieważ obciążają znacznie serwer.

Kolejna ważną opcją jest

```
os level = 255
```

określająca poziom systemu, który Samba będzie pozorować podczas wyboru przeglądarki[1]. Liczba mówi, jaki system zostanie wybrany do obsługi przeglądarki. Wartości przypisywane systemom operacyjnym w wyborach przeglądarki przedstawiamy w tabelce:

System operacyjny	Wartość
Windows NT Server 4.0	33
Windows NT Server 3.51	32
Windows NT Workstation	17
Windows 98	2
Windows 95	1
Windows 3.1 for workgroups	1

Ustawienie wartości `255` daje pewność, że Samba będzie główną przeglądarką.

Jeśli chcemy, aby komputer w sieci z zainstalowaną Sambą był domyślną przeglądarką to należy w opcję

```
preferred master
```

ustawić na `yes`.

```
domain master=yes
```

za pomocą tego parametru można zadecydować, czy Samba ma być kontrolerem domeny.

```
local master = yes
```

oznacza czy Samba zaraz po uruchomieniu spróbuje zostać główną przeglądarką lokalną w sieci. Jednak ustawienie tej opcji nie gwarantuje, że zostanie ona wybrana. Za to ustawienie tej opcji na `no` zaprzepaszcza wszystkie szanse.

Proces nmbd może wyszukiwać nazw komputerów za pomocą usług serwera DNS, do tego służy opcja

[1]Nie chodzi tu o przeglądarkę internetową, lecz o mechanizm pozwalający śledzić (przeglądać ang. browse) w otoczeniu sieciowym listę udostępnionych folderów i drukarek

```
dns proxy = No
```

My jednak nie korzystamy z tej możliwości.

```
Security=user
```

Najczęściej parametr ten ma wartość `user`, co pozwala na udostępnianie zasobów zabezpieczonych indywidualnym hasłem dla każdego użytkownika. Podanie w tym miejscu wartości `share` umożliwia tworzenie zasobów dostępnych bez hasła; `server` z kolei pozwala na wskazanie serwera, który ma dokonać autoryzacji użytkownika.

Parametr

```
encrypt passwords
```

może przyjmować wartości `yes` lub `no`. Oznacza to włączenie lub wyłączenie szyfrowania haseł.

```
guest account
```

określa nazwę konta, które będzie używane podczas „gościnnego" dostępu do udziałów serwera Samby. Najczęściej tą nazwą jest `nobody`, ale w niektórych serwerach bywa tez `ftp`.

```
hosts allow
```

dzięki temu parametrowi można ograniczyć dostęp do serwera dla wybranych podsieci, bądź wręcz konkretnych komputerów (numerów IP). W naszym przypadku:

```
hosts allow = 192.168.2.3/255.255.255.0
```

Opcja

```
load printers
```

nakazuje Sambie utworzenie udziałów dla wszystkich dostępnych drukarek w sieci i załadowanie ich na listę przeglądania. Samba wykona tą czynność pod warunkiem, że plik

```
/etc/printcap
```

będzie zawierał odpowiednie wpisy. Dostępne możliwe ustawienia tego parametru to `yes/no`. Plik, który podaliśmy powyżej nie musi być jedynym, z którego Samba odczyta drukarki, możemy wprowadzić własny plik w opcji

```
printcap name = lpstat
```

gdzie `lpstat` jest nazwą pliku z pełną ścieżką zawierającego odpowiednie wpisy.

Opcja

```
printing
```

informuje Sambę o systemie druku używanym przez serwer. W Unixie istnieje kilka sposobów sterowania wydrukiem. Samba obsługuje siedem różnych typów:

Zmienna	definicja
BSD	System Berkeley Unix
SYSV	System V
AIX	System operacyjny AIX(IBM)
HPUX	Unix Hewlett-Packard)
QNX	System operacyjny czasu rzeczywistego QNX
LPRNG	LPR Next Generation (Powell)
SOFTQ	System SOFTQ
PLP	Portale Line Printer (Powell)

Parametr

```
bind interfaces only
```

sprawia, że procesy smbd i nmbd obsługują żądania pochodzące tylko z tych podsieci, które są wymienione w opcji `interfaces`.

```
domain logons
```

opcja ta pozwala włączyć możliwość logowania do domeny. W sieciach amatorskich teoretycznie nie ma to większego zastosowania, jednak umożliwia pozbycie się występującego niekiedy problemu, polegającego na odmowie klientowi dostępu do zasobów pomimo podania poprawnego hasła. Źródło tych kłopotów ewidentnie leży po stronie systemu Windows 9x, gdyż w W2K problem w ogóle nie występuje. Po włączeniu w Windows logowania do domeny problem znika.

Używanie logowania do domeny otwiera przed administratorem serwera możliwość uruchamiania na komputerach użytkowników praktycznie dowolnych programów bez ich zgody. (Podczas logowania z serwera jest ściągany i uruchamiany tzw. skrypt logowania - jeśli takowy istnieje - i użytkownik nie ma na to żadnego wpływu). Powiązany z poprzednia opcją jest parametr

```
logon path
```

który określa położenie profili przechodnich. Kiedy użytkownik loguje się, profil przechodni jest przekazywany z serwera do klienta i uaktualniany dla użytkownika. Natomiast po wylogowaniu zawartość profilu jest przechowywana spowrotem na serwerze aż do następnego zalogowania. Kolejnymi powiązanymi opcjami są

```
logon driver
logon home
```

Pierwsza określa literę na dysku w kliencie NT, na którą będzie mapowany katalog macierzysty podany w drugiej opcji `logon home`. Działa to tylko z klientami Windows.

```
wins support
```

Ustawienie `yes` powoduje, że Samba staje się serwerem Wins (jest to odpowiednik DNS dla nazw NetBIOS). Może to być przydatne, jeśli nasza sieć jest podzielona na kilka podsieci. Dzięki serwerowi Wins komputery z różnych podsieci będą mogły się widzieć w otoczeniu sieciowym. Oczywiście w takim przypadku konieczne jest odpowiednie skonfigurowanie systemów klienckich, by wiedziały, że mają się łączyć z serwerem Wins. Można to zrobić za pośrednictwem serwisu DHCP.

To, pod jaką nazwą nasz komputer jest widziany w sieci określa opcja `netbios name`.

Samba może uaktualniać standardowy Unixowy plik haseł, gdy użytkownik zmieni zaszyfrowane hasło. Zaszyfrowane hasła są przechowywane na serwerze Samby w pliku

```
/etc/stw/private/smbpasswd.
```

Wstawienie `yes` w linijce `unix password sync` spowoduje właśnie taką reakcje systemu.

```
pam password change = yes
```

Opcja ta pozwala na zmianę hasła użytkownika na serwerze, z klienta za pośrednictwem Samby przy wykorzystaniu modułu PAM.

Dalej w pliku konfiguracyjnym Samby znajdują się deklaracje udziałów. Możemy deklarować własne udziały, są jednak trzy udziały o specjalnym znaczeniu:

```
netlogon,
profile,
homes.
```

Udział `netlogon` wymagany jest, gdy konfigurujemy kontroler domeny. W udziale `profile` zapisywane są profile poszczególnych użtkowników, natomiast udział `homes` pozwala udostępniać foldery domowe użytkowników na serwerze plików. Każdy użytkownik będzie mógł zobaczyć, oczywiście, jedynie swój katalog domowy.

Poniżej opiszemy opcje, których użyliśmy podczas deklaracji wyżej wymienionych udziałów. Opcje te mogą występować, ale nie muszą, w deklaracji wszystkich udziałów.

Pierwsza opcja to komentarz (`comment`), kolejna zaś to ścieżka dostępu do skryptu logowania. Kiedy użytkownik zaloguje się w domenie, która zawiera skrypt startowy, zobaczy małe okno z wynikami pracy skryptu.

Kolejna linia pliku mówi o tym,czy wskazany udział powinien pojawiac się na liście zasobów udostępniającego go komputera. Wskazanie `no` powoduje niewyświetlanie udziałów.

Opcja

```
public
```

oznacza to samo co `guest ok.` czyli umożliwianie bądź nie gościnnego dostępu do udziałów. Kiedy gościnny użytkownik łączy się jego prawa dostępu są równe prawom nadanym użytkownikowi podanym w opcji `guest account`.

Opcja

```
writeable
```

pozwala na decydowanie o możliwości zapisu zmian przez klienta.

W Windows 95 NT każdy użytkownik może mieć swój własny profil. W profilu zapisane są informacje takie jak:

- wygląd pulpitu użytkownika,
- aplikacje widoczne w menu start,
- tło,

Jeśli profil przechowywany jest na lokalnym dysku twardym, to nazywamy go *profilem lokalnym*, natomiast profil przechowywany na serwerze i przesyłany do użytkownika za każdym zalogowaniem nazywamy *profilem przechodnim*. Ta druga możliwość jest bardzo przydatna, np. jeśli ktoś pracuje w biurze a

pracę musi dokończyć w domu- wszystkie ustawienia zostają przesłane np. do domu. Rozwiązuje to wiele problemów.

I tak opcja `path` mówi, w którym miejscu na dysku twardym znajduje się plik profilu.

Zestaw przywilejów jakie będą nadawane plikom tworzonym spod Windows określają opcje:

```
create mode = 0600
directory mode = 0700
```

Opcja

```
force group
```

ustala statyczny identyfikator grupy, który po uwierzytelnieniu użytkownika będzie używany we wszystkich połączeniach z udziałem. Opcja ta przypisuje określoną grupę każdemu plikowi i katalogowi utworzonemu przez klienta SMB.

```
profile acls = yes
```

Opcja jest konieczna do tego, aby prawidłowo działały profile przechodnie z systemami W2K i XP.

Sekcja `homes` ma specjalne znaczenie. Pozwala ona udostępniać przez otoczenie sieciowe foldery użytkowników. W sekcji tej można użyć większość parametrów dostępnych dla sekcji katalogów.

```
browseable
```

Pozwala zadecydować, czy zasób ma być wyświetlany przy „listowaniu" dostępnych katalogów na serwerze. Jeśli ustawimy ten parametr na no, katalog nie będzie widoczny, jednak dostęp do niego będzie nadal możliwy przez podanie pełnej ścieżki w postaci:

```
//nazwa_serwera/nazwa_zasobu
```

Rozdział 4

Konfiguracja klienta

4.1 Podłączenie klienta Solaris

Serwer LDAP możemy konfigurować tylko z pozycji klienta. Serwer nie może być swoim klientem. Aby dokończyć konfigurację serwera musimy skonfigurować klienta LDAP na innym komputerze.

Krok 1. Na komputerze, który ma być klientem LDAP wpisujemy:

```
ldapclient init 192.168.2.105
```

gdzie `192.168.2.105` to adres IP serwera LDAP. Ponieważ podczas przygotowania serwera LDAP skonfigurowaliśmy profil domyślny (o nazwie `default`), to zostanie on automatycznie użyty by zainicjalizować klienta. Mamy teraz dzięki temu mniej pracy i polecenie `ldapclient` nie wymaga stosu przedziwnych zaklęć.

Krok 2. Polecenie `ldapclient` powoduje zmiany w pliku `/etc/nsswitch.conf`. Jeśli wcześniej korzystaliśmy z usług DNS i w pliku `/etc/nsswitch.conf` mieliśmy wpis:

```
hosts: files dns
```

to po zawołaniu polecenia `ldapclient init` ten wiersz będzie wyglądał następująco:

```
hosts: ldap [NOTFOUND=return] files
```

Oznacza to, że już nie będziemy mogli korzystać z DNS. W naszej konfiguracji przyjęliśmy, że dalej korzystamy z DNS, więc dlatego odtworzyliśmy wersję pliku `nsswitch.conf` dla DNS poprzez kopiowanie:

```
cp /etc/nsswitch.dns /etc/nsswitch.conf
```

i dopisaliśmy `ldap` do `passwd, group` i `automaster`. Wynikowy plik załączony jest poniżej:

```
passwd:     files ldap
group:      files ldap

networks:   files
protocols:  files
rpc:        files
ethers:     files
netmasks:   files
bootparams: files
publickey:  files


netgroup:   files
automount:  files ldap
aliases:    files
services:   files
printers:   user files
```

Krok 3. Aby możliwe było korzystanie z bazy o użytkownikach na serwerze LDAP, musimy dostosować konfigurację modułu PAM (Pluggable Authentication Modul) odpowiedzialnego za autoryzację użytkowników w systemie Solaris. W tym celu należy zmodyfikować plik `/etc/pam.conf` do postaci takiej jak w dodatku na stronie 50.

Krok 4. Ostatnim krokiem jest dostosowanie klienta NFS do serwera. W Solaris 10 domyślnie po instalacji systemu klient NFS jest włączony. Należy jedynie zadbać, aby klient i serwer był w tej samej domenie. W tym celu zmieniamy wartość zmiennnej `NFSMAPID_DOMAIN` tak jak na serwerze (por. str. 16).

4.2 Podłączenie klienta Windows

Na stacji roboczej będącej klientem domeny Windows możliwe jest logowanie użytkowników zarejestrowanych na kontrolerze domeny. Tacy użytkownicy nie muszą posiadać konta na komputerze, przed którym siadają.

Podłączenie klienta Windows do serwera Samby pełniącego rolę kontrolera domeny wykonujemy w dwóch krokach, pierwszy na serwerze, drugi na kliencie. Zanim jednak zaczniemy rejestrować nowe stacje robocze, na serwerze, załóżmy nową grupę o nazwie `smbnode`. Można to zrobić poleceniem

```
groupadd smbnode
```

Musimy też zarejestrować konto `root` w serwerze Samby. Wykonujemy to poleceniem

```
smbpasswd -a root
```

i nadajemy hasło dla tego konta. Ponieważ jest to konto w Sambie nie w systemie, dla bezpieczeństwa, nie należy podawać w tym miejscu hasła systemowego `root`.

Krok 1. Zaczynamy od dopisania nazwy hosta klienta na serwerze plików. W tym celu wołamy polecenie

```
useradd -g smbnode -s /bin/false -d /dev/null lab-b04$
```

gdzie `lab-b04` to nazwa naszej stacji roboczej. Znak `$` na końcu nazwy hosta jest konieczny. Spowoduje to dodanie nowego konta użytkownika do `/etc/passwd`. Zauważmy, że ten użytkownik nie ma przypisanej powłoki shell i nie ma katalogu domowego. Jest to sztuczny użytkownik, zakładany wyłącznie po to, aby procesy związane z daną stacją roboczą wykonywane były z jego przywilejami.

Następnie rejestrujemy nazwę klienta w serwerze Samby wołając:

```
smbpasswd -a -m lab-b04
```

Krok 2. Na kliencie, czyli na komputerze z Windows NT 4.0, W2K lub XP, uruchamiamy aplet System w Panelu sterowania. Wybieramy zakładkę Nazwa komputera i klikamy przycisk Zmień aby przyłączyć się do domeny. W oknie dialogowym które się otworzy wpisujemy nazwę domeny (nie grupy roboczej). W naszym przypadku jest to `im.uwb.edu.pl`. Zostaniemy poproszeni o hasło `root` i komputer zostanie podłączony do domeny.

4.3 Przygotowanie katalogu LDAP

Następnym etapem jest wpisanie do katalogu LDAP danych potrzebnych do zdalnego logowania użytkownika. Można to zrobić na kilka sposobów:

1. użyć konsoli administracyjnej.
2. napisać ręcznie plik w formacie `ldif` zrozumiałym dla LDAP.
3. zaimportować odpowiednie pliki systemowe, takie jak `/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/auto_master`, `/etc/auto_home`

Wybieramy metodę (c). Z odpowiednich plików, podanych wyżej, usuwamy zbędne informacje, zostawiamy tylko to, co dotyczy użytkowników, których chcemy mieć w katalogu LDAP. Import wykonuje się za pomocą polecenia `ldapaddent`. Jeżeli w bieżącym katalogu mamy „wyczyszczone” kopię `/etc/passwd` to wołamy:

```
ldapaddent -D "cn=directory manager" -a simple
-f passwd passwd
```

Zostaniemy poproszeni o podanie hasła dla `directory manager`, które zostało wcześniej skonfigurowane przy pomocy iDSconfig. Analogicznie dodajemy bazy `group, shadow, auto_master, auto_home`. Można dodać więcej tych baz. Np. można dodać `hosts` i w ten sposób przestać korzystać z DNS.

W tym momencie nowo dodani użytkownicy do katalogu LDAP powinni być widoczni z klienta. Aby się o tym przekonać wpisujemy polecenie:

```
listusers
```

lub

```
getent passwd
```

Oba powinne zwrócić pełną listę użytkowników, tzn. użytkowników lokalnych na kliencie, czyli tych, których mamy w `/etc/passwd` oraz użytkowników z katalogu LDAP. Poniżej przedstawiona jest zawartość pliku `/etc/passwd`:

```
root:x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/:
webservd:x:80:80:WebServer Reserved UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
mariusz:x:1001:10::/export/home/mariusz:/bin/bash
```

Dla porównania, poniżej znajduje się wynik polecenia `listuser`:

```
janek           jan_kowalski
mariusz
marysia         marysia_mala
noaccess        No Access User
nobody          NFS Anonymous Access User
nobody4         SunOS 4.x NFS Anonymous Access User
```

Jak widać polecenie `listusers` wyświetla dodatkowo użytkowników Janek i Marysia, których nie mamy w pliku `/etc/passwd`. Możemy próbować zalogować się na jedo z tych kont. Możemy to zrobić przy pomocy polecenia

```
su - nazwa_użytkownika
```

Po tych zmianach możliwe jest już zalogowanie korzystając z danych o użytkownikach przechowywanych na serwerze LDAP.

Rozdział 5

Skrypty do zarządzania katalogiem LDAP

Przejdźmy teraz do przedstawienia podstawowych czynności związanych z administracją katalogu LDAP. Pokażemy w tym rozdziale jak dodawać nowych użytkowników, zmieniać ich dane i jak usunąć użytkownika z LDAP.

5.1 Dodawanie nowego użytkownika na serwerze plików

Dodanie nowego konta użytkownika na serwerze plików polega na:

1. utworzeniu katalogu domowego;
2. zarejestrowanie ścieżki do katalogu domowego;
3. wybraniu powłoki shell;
4. nadaniu hasła.

Pierwsze trzy punkty w systemie Solaris realizuje polecenie `useradd`, natomiast hasło nadaje się i zmienia przy pomocy polecenia `passwd`. Także, aby dodać nowe konto na serwerze plików administrator musi zawołać te dwa polecenia z odpowiednim zestawem opcji.

Informacje o założonym koncie znajdują się w dwóch plikach systemowych: `/etc/passwd` i `/etc/shadow`. Wbrew pozorom w pierwszym pliku przechowywany jest tylko identyfikator numeryczny, nazwa użytkownika, ścieżka do katalogu domowego, komentarz i powłoka shell. Natomiast w drugim pliku przechowywane są zaszyfrowane hasła.

Standartowo fizyczne katalogi domowe użytkowników na Solaris znajdują się w katalogu `/export/home`. Przedrostek export ma tutaj całkiem naturalne znaczenie, gdyż na serwerze plików katalog ten jest udostępniany, tak aby

katalogi domowe mogły być montowane na stacjach roboczych. Zwykle po zamontowaniu na stacji roboczej katalog domowy użytkownika znajduje się w katalogu `/home`. Inaczej zatem wygląda pełna ścieżka do katalogu użytkownika na serwerze a inaczej na stacji roboczej. Może to powodować nieprawidłowe działanie skryptów lub programów z których korzysta użytkownik. Aby temu zapobiec na serwerze plików modyfikujemy ścieżkę do katalogu domowego użytkownika poleceniem `usermod` wyrzucając przedrostek export. Modyfikacja ta zmienia tylko plik `/etc/passwd`, fizyczny katalog pozostaje w tym samym miejscu. Jeśli na serwerze plików mamy udostępniony katalog `/export/home` przez NFS i uruchomiony jest automounter to przy logowaniu użytkownika jego katalog z `/export/home` zostanie zamontowany w `/home`, tak jak to ma miejsce na stacji roboczej.

Aby usprawnić pracę administratora został napisany skrypt w Perlu, który realizuje powyższe zadanie. Skrypt uruchamia się następująco:

```
ldapuseradds [-u uid] [-c comment] [-d dir] [-s shell] login
```

Poszczególne opcje mają następujące znaczenie:

-u identyfikator numeryczny;

-c komentarz dopisywany do użytkownika, na ogół jego imię i nazwisko;

-d ścieżka do katalogu domowego;

-s ścieżka do programu shell;

login nazwa użytkownika.

Po uruchomieniu powyższego skryptu zostaniemy poproszeni o wprowadzenie hasła dla użytkownika.
Jeśli operacja zakończy się powodzeniem to na ekranie zostaną wypisane dwa wiersze: pierwszy z pliku `/etc/passwd`, drugi z `/etc/shadow` odpowiadające dodanemu użytkownikowi. Teraz należy dane te dodać do katalogu LDAP. Zadanie to wykonuje następny skrypt.

Pełny skrypt `ldapuseradds` znajduje się na stronie 39.

5.2 Dodawanie nowego użytkownika w katalogu LDAP

Po utworzeniu nowego konta na serwerze plików, aby dany użytkownik mógł logować się zdalnie do swego katalogu domowego, należy dodać do katalogu LDAP informacje o tym użytkowniku, jego identyfikator, hasło, ścieżkę do katalogu domowego, powłokę shell itd. Zadanie to można wykonać za pomocą konsoli administracyjnej, albo bardziej „ręcznie” wykorzystując program

`ldapaddent` wspomniany wcześniej. Ten właśnie program wykorzystujemy w skrypcie `ldapuseradds`, który automatyzuje proces dopisywania danych o użytkowniku do katalogu LDAP. Przy pomocy `ldapaddent` wystarczy zaimportować do LDAP dwa wiersze z plików `/etc/passwd` i `/etc/shadow` z serwera plików, odpowiadające nowemu użytkownikowi. Jeśli mamy na ekranie wynik skryptu `ldapuseradds`, to zadanie jest bardzo proste.

Skrypt `ldapuseradd` wołamy w następujący sposób:

```
ldapuseradd passwd_entry shadow_entry
```

W miejscu `passwd_entry` i `shadow_entry` wstawiamy wiersze pochodzące z odpowiednich plików, te dwa wiersze, które wyświetlił skrypt poprzedni. Po uruchomieniu `ldapuseradd` zostaniemy poproszeni o podanie hasła dla directory manager.

Pełny skrypt znajduje się na stronie 41.

5.3 Zmiana hasła

Od momentu dopisania danych o użytkowniku do katalogu LDAP, wszystkie zmiany dotyczące tego użytkownika muszą być wykonane w tym katalogu, gdyż z niego stacje robocze pobierają informacje niezbędne do zalogowania, a nie z serwera plików.

Jedną z częściej wykonywanych przez administratora operacji na bazie użytkowników jest zmiana hasła. Aby zmienić hasło użytkownika `janek` wykonujemy polecenie:

```
ldaplist -l passwd janek > /tmp/janek
```

Jego wynikiem jest zapisanie pliku `/tmp/janek`, który wygląda następująco:

```
dn: uid=janek,ou=people,dc=im,dc=uwb,dc=edu,dc=pl
        objectClass: posixAccount
        objectClass: shadowAccount
        objectClass: account
        objectClass: top
        uid: janek
        cn: janek
        uidNumber: 1003
        gidNumber: 1
        gecos: jan_kowalski
        homeDirectory: /home/jan
        loginShell: /bin/bash
        shadowLastChange: 12926
        shadowFlag: 0
```

Zmodyfikujemy zawartość tego pliku i wstawimy go do katalogu LDAP za pomocą programu `ldapmodify`. Nowe hasło podajemy w polu `userPassword` w postaci zaszyfrowanej, poprzedzając je nazwą metody użytej podczas szyfrowania. W tym celu, w linii poleceń, można napisać w Perlu mały skrypt. Uruchamiamy interpreter Perla poleceniem `perl` i piszemy:

```
print crypt("123456","ab"), "\n";
```

gdzie napis `123456` to nowe hasło. Po zakończeniu wciskamy `Ctrl+d` i program wypisuje na konsoli hasło w postaci gotowej do użycia:

```
ab01FAX.bQRSU
```

Modyfikujemy plik `/tmp/janek` do postaci jak poniżej:

```
dn: uid=janek,ou=people,dc=im,dc=uwb,dc=edu,dc=pl
        changetype: modify
        replace: userPassword
        userPassword: {crypt}ab01FAX.bQRSU
```

i wykonujemy polecenie

```
ldapmodify  -D "cn=directory manager" -r -f /tmp/janek -h 192.168.2.105
```

Podany adres IP 192.168.2.105 jest adresem naszego serwera LDAP. Od tego momentu użytkownik `janek` powinien logować się z nowym hasłem.

Jak widać operacja zmiany hasła jest dość kłopotliwa i wymaga wykonania szeregu poleceń. Dlatego też napisaliśmy w Perlu skrypt `ldappasswd`, który tę pracę wykonuje. Wywołuje się go w następujący sposób:

```
ldappasswd  [-H ldaphost] [-D ldapdomain] login
```

gdzie

-h adres serwera LDAP;

-d domena katalogu LDAP;

login nazwa użytkownika;

Po uruchomieniu skryptu zostaniemy poproszeni dwukrotnie poproszeni o podanie nowego hasła użytkownika, a następnie hasła `directory manager`.

Pełny skrypt znajduje się na stronie 42.

5.4 Zmiana danych użytkownika

Zmian w danych użytkownika w katalogu LDAP dokonuje się w analogiczny sposób jak w przypadku zmiany hasła. Aby uzyskać komplet danych o użytkowniku zapisanych w katalogu LDAP możemy użyć polecenie `ldaplist` tak jak w poprzednim podrozdziale.

Aby zmienić opis użytkownika, za który odpowiada pole `gecos` w katalogu LDAP, tworzymy plik `/tmp/janek` o następującej treści:

```
dn: uid=janek,ou=people,dc=im,dc=uwb,dc=edu,dc=pl
        changetype: modify
        replace: gecos
        gecos: Jan Kowalski
```

Aby nasze zmiany zostały wprowadzone należy wpisać polecenie:

```
ldapmodify  -D "cn=directory manager" -r -f
            /tmp/janek -h 192.168.2.105
```

W ten sposób udało się nam zmienić opis użytkownika `janek` z `jan_kowalski` na `Jan Kowalski`.

Aby usprawnić wykonywanie zmian napisaliśmy skrypt `ldapusermod`. Woła się go w poniższy sposób:

```
ldapusermod [-H ldaphost] [-D ldapdomain] [-c comment]
            [-d dir] [-s shell] login
```

Opcje mają następujące znaczenie:

-H adres serwera LDAP;

-D domena katalogu LDAP;

-c komentarz dotyczący użytkownika;

-d katalog domowy użytkownika;

-s program shell użytkownika;

login nazwa użytkownika;

Po uruchomieniu skryptu zostaniemy poproszeni o podanie hasła dla `directory manager`.

Skrypt znajduje się w dodatku na stronie 44.

5.5 Usunięcie konta użytkownika

Usuniecie danych z katalogu LDAP jest bardzo proste. Wystarczy zawołać polecenie `ldapdelete` i jako argument podać pełny klucz identyfikujący dane, tzw. *distinguished name*. Pełny klucz w przypadku konta użytkownika tworzy jego nazwa i domena katalogu LDAP.

Aby ułatwić pracę administratorowi dając mu komplet narzędzi do pracy z serwerem LDAP napisaliśmy skrypt `ldapuserdel`, który wykonuje zadanie usunięcia danych użytkownika z katalogu LDAP. Schemat jego wywołania jest następujący:

```
ldapuserdel [-H ldaphost] [-D ldapdomain] login
```

Poszczególne opcje mają następujące znaczenie:

-H adres serwera LDAP;

-D domena katalogu LDAP;

login nazwa użytkownika;

Skrypt ten nie usuwa danych i katalogu domowego z serwera plików. Tę trudno odwracalną operację pozostawiamy administratorowi.

Pelny skrypt usuwania konta użytkownika znajduje się w dodatku na stronie 45

Wszystkie skrypty zostały napisane w Perlu. Korzystają one z kilku wspólnych zmiennych. w których pamiętamy adres IP serwera LDAP i nazwę domeny LDAP oraz z dwóch wspólnych procedur do pobrania hasła i przekształcenia nazwy domenowej na postać `ldif`. Te wspólne fragmenty kodu zostały umieszczone we współdzielonym module `LDAP.pm`, który umieściliśmy w dodatku na stronie 38.

Dodatek A

Skrypty

Moduł LDAP.pm

```
#
# LDAP utility subroutines
#
# Last modified: Sept 20, 2005
#
# Copyright (c) 2005 Pawel Kowalewski & Mariusz Zynel.
#
# This  software  is  FREE.  You can use  and/or  redistribute  it for any
# purpose in either,  modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission  notice shall be included
# in all copies or substantial portions of this software.
#
# THIS  SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY  OF ANY KIND,
# EITHER  EXPRESSED OR IMPLIED.  IN NO EVENT WILL THE COPYRIGHT  HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

package LDAP;

use POSIX;
use Exporter;
@ISA = qw(Exporter);
@EXPORT_OK =  qw($self $host $domain get_passwd get_dc);

($self = $0) =~ s/.*\///;

$host = "192.168.2.105";
$domain = "im.uwb.edu.pl";

sub get_passwd {
    my $msg = shift;
    if (! $msg) {
        $msg = "Enter password:";
```

```
    }
    print $msg, " ";
    system('/usr/bin/stty', '-echo');  # Disable echoing
    my $passwd = <STDIN>;
    print "\n";
    system('/usr/bin/stty', 'echo');   # Enable echoing
    chomp $passwd;
    return $passwd;
}

sub get_dc {
    (my $domain = shift) =~ s/\./,dc=/g;
    return "dc=$domain";
}

1;
```

Skrypt dodania użytkownika do serwera LDAP

```
#!/bin/perl
#
# Add a user account to server
#
# Last modified: Jul 8, 2005
#
# Copyright (c) 2005 Pawel Kowalewski & Mariusz Zynel.
#
# This  software  is  FREE.  You can use  and/or  redistribute  it for any
# purpose in either,  modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission  notice shall be included
# in all copies or substantial portions of this software.
#
# THIS  SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY  OF ANY KIND,
# EITHER  EXPRESSED OR IMPLIED.  IN NO EVENT WILL THE COPYRIGHT  HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

use strict;
use Getopt::Std;

my ($uid, $comment, $dir, $m, $shell);

our ($opt_u, $opt_c, $opt_d, $opt_s);

(my $self = $0) =~ s/.*\///;
```

```
sub usage{
    die "Usage: $self [-u uid] [-c comment] [-d dir] [-s shell] login\n";
}

getopt("ucds");

if ($#ARGV < 0) {
  usage();
}

my $login = @ARGV[0];

if ($opt_u) {
$uid = "-u $opt_u";
}

if ($opt_c) {
$comment = "-c \"$opt_c\"";
}

if ($opt_d) {
$dir = "-d $opt_d -m";
} else {
$dir = "-d /export/home/$login -m";
}

if ($opt_s) {
$shell = "-s $opt_s";
} else {
$shell = "-s /bin/bash";
}

system("useradd $uid $comment $dir $shell $login");
system("passwd $login");
system("usermod -d /home/$login $login");

print "\n";

open(INFILE, "/etc/passwd") || die "ERROR: cannot open file /etc/passwd\n";
while (<INFILE>) {
print $_ if /^$login:/;
}
close(INFILE);

open(INFILE, "/etc/shadow") || die "ERROR: cannot open file /etc/shadow\n";
while (<INFILE>) {
print $_ if /^$login:/;
}
close(INFILE);

print "\n";
```

Skrypt dodawania użytkownika na kliencie LDAP

```
#!/bin/perl
#
# Add a user account to LDAP directory
#
# Last modified: Sept 20, 2005
#
# Copyright (c) 2005 Pawel Kowalewski & Mariusz Zynel.
#
# This  software  is  FREE.  You can use  and/or  redistribute  it for any
# purpose in either,  modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission  notice shall be included
# in all copies or substantial portions of this software.
#
# THIS  SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY  OF ANY KIND,
# EITHER  EXPRESSED OR IMPLIED.  IN NO EVENT WILL THE COPYRIGHT  HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

use strict;
use Getopt::Std;
use FindBin;
use lib $FindBin::Bin;
use LDAP qw($self $host $domain get_passwd get_dc);


sub usage{
    die "usage: $self passwd_entry shadow_entry\n";
}

if ($#ARGV + 1 != 2) {
    usage();
}

my $passwdentry = `mktemp /tmp/passwdentryXXXXXX`;
my $shadowentry = `mktemp /tmp/shadowentryXXXXXX`;

chomp $passwdentry;
chomp $shadowentry;

open(OUTFILE, ">$passwdentry") || die "ERROR: cannot open file
              $passwdentry for writing\n";
print OUTFILE @ARGV[0];
close(OUTFILE);
```

```
open(OUTFILE, ">$shadowentry") || die "ERROR: cannot open file
             $shadowentry for writing\n";
print OUTFILE @ARGV[1];
close(OUTFILE);

my $passwd = get_passwd("Enter Directory Manager password:");

system("ldapaddent -D \"cn=directory manager\" -w $passwd -a simple -f
                      $passwdentry passwd") == 0
        || die "\nERROR: ldapaddent passwd failed\n";
system("ldapaddent -D \"cn=directory manager\" -w $passwd -a simple -f
                      $shadowentry shadow") == 0
        || die "\nERROR: ldapaddent shadow failed\n";

unlink $passwdentry;
unlink $shadowentry;
```

Skrypt zmiany hasła użytkownika LDAP

```
#!/bin/perl
#
# Change user's password in LDAP directory
#
# Last modified: Sept 20, 2005
#
# Copyright (c) 2005 Pawel Kowalewski & Mariusz Zynel.
#
# This  software  is  FREE.  You can use  and/or  redistribute  it for any
# purpose in either,  modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission  notice shall be included
# in all copies or substantial portions of this software.
#
# THIS  SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY  OF ANY KIND,
# EITHER  EXPRESSED OR IMPLIED.  IN NO EVENT WILL THE COPYRIGHT  HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

use strict;
use Getopt::Std;
use FindBin;
use lib $FindBin::Bin;
use LDAP qw($self $host $domain get_passwd get_dc);


our ($opt_H, $opt_D);

sub usage {
```

```
    die "usage: $self [-H ldaphost] [-D ldapdomain] login\n";
}

getopt('H:D:');

if ($opt_H) {
    $host = $opt_H;
}

if ($opt_D) {
    $domain = $opt_D;
}

if ($#ARGV + 1 != 1) {
    usage();
}

my $login = @ARGV[0];

my $password = get_passwd("Enter new password:");
my $vpassword = get_passwd("Verify password:");

if ($password eq $vpassword) {

    if (length($password) < 6 || $password !~ /[^a-z].*[^a-z]/) {
        die "ERROR: password is trivial\n";
    }

    if (length($password) > 8) {
        die "ERROR: password is too long, max 8 characters\n";
    }

    $password = crypt($password, "ab");

    my $tmpfile = `mktemp /tmp/ldappasswdXXXXXX`;
    chomp $tmpfile;

    open(OUTFILE, ">$tmpfile") || die "ERROR: cannot open file
                  $tmpfile for writing\n";
    print OUTFILE "dn: uid=$login,ou=people," . get_dc($domain) . "\n";
    print OUTFILE "changetype: modify\n";
    print OUTFILE "replace: userPassword\n";
    print OUTFILE "userPassword: {crypt}$password\n";
    close(OUTFILE);

    my $passwd = get_passwd("Enter Directory Manager password:");
    system("ldapmodify -D \"cn=directory manager\" -w $passwd -h $host -r -f
                          $tmpfile");
    unlink $tmpfile;

} else {
    print "ERROR: entered passwords do not match\n";
```

```
}
```

Skrypt modyfikacji danych o użytkowniku LDAP

```
#!/bin/perl
#
# Change user's data in LDAP directory
#
# Last modified: Sept 20, 2005
#
# Copyright (c) 2005 Pawel Kowalewski & Mariusz Zynel.
#
# This  software  is  FREE.  You can use  and/or  redistribute  it for any
# purpose in either,  modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission  notice shall be included
# in all copies or substantial portions of this software.
#
# THIS  SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY  OF ANY KIND,
# EITHER  EXPRESSED OR IMPLIED.  IN NO EVENT WILL THE COPYRIGHT  HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

use strict;
use Getopt::Std;
use FindBin;
use lib $FindBin::Bin;
use LDAP qw($self $host $domain get_passwd get_dc);


our ($opt_H, $opt_D, $opt_c, $opt_d, $opt_s);
my ($comment, $dir, $shell);

sub usage {
    die "usage: $self [-H ldaphost] [-D ldapdomain] [-c comment] [-d dir]
                      [-s shell] login\n";
}

getopt('H:D:c:d:s:');

if ($opt_H) {
    $host = $opt_H;
}

if ($opt_D) {
    $domain = $opt_D;
}
```

```
if ($opt_c) {
    $comment = "replace: gecos\ngecos: $opt_c\n";
}

if ($opt_d) {
    if ($comment) {
        $dir = "-\n";
    }
    $dir .= "replace: homeDirectory\nhomeDirectory: $opt_d\n";
}

if ($opt_s) {
    if ($comment || $dir) {
        $shell = "-\n";
    }
    $shell .= "replace: loginShell\nloginShell: $opt_d\n";
}

if ($#ARGV + 1 != 1) {
    usage();
}

my $login = @ARGV[0];

my $tmpfile = `mktemp /tmp/ldappasswdXXXXXX`;
chomp $tmpfile;

my $passwd = get_passwd("Enter Directory Manager password:");

open(OUTFILE, ">$tmpfile") || die "ERROR: cannot open file
             $tmpfile for writing\n";
print OUTFILE "dn: uid=$login,ou=people," . get_dc($domain) . "\n";
print OUTFILE "changetype: modify\n";
print OUTFILE $comment . $dir . $shell;
close(OUTFILE);

system("ldapmodify -D \"cn=directory manager\" -w $passwd -h $host -r -f
                     $tmpfile") == 0
        || die "\nERROR: ldapmodify failed\n";

unlink $tmpfile;
```

Skrypt usuwający dane o użytkowniku LDAP

```
#!/bin/perl
#
# Delete user account from LDAP directory
#
# Last modified: Sept 20, 2005
#
# Copyright (c) 2005 Pawel Kowalewski & Mariusz Zynel.
#
# This  software  is  FREE.  You can use  and/or  redistribute  it for any
# purpose in either,  modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission  notice shall be included
# in all copies or substantial portions of this software.
#
# THIS  SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY  OF ANY KIND,
# EITHER  EXPRESSED OR IMPLIED.  IN NO EVENT WILL THE COPYRIGHT  HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

use strict;
use Getopt::Std;
use FindBin;
use lib $FindBin::Bin;
use LDAP qw($self $host $domain get_passwd get_dc);


our ($opt_H, $opt_D);

sub usage{
    die "usage: $self [-H ldaphost] [-D ldapdomain] login\n";
}

getopt('H:D:');

if ($opt_H) {
    $host = $opt_H;
}

if ($opt_D) {
    $domain = $opt_D;
}

if ($#ARGV + 1 != 1) {
    usage();
}

my $login = @ARGV[0];

my $passwd = get_passwd("Enter Directory Manager password:");

system("ldapdelete -D \"cn=directory manager\" -w $passwd -h
```

```
$host \"uid=$login,ou=people," . get_dc($domain) . "\"") == 0
        || die "\nERROR: ldapdelete passwd\n";
```

Dodatek B

Plik konfiguracyjny Samby

```
theta$ cat /etc/sfw/smb.conf
[global]
         workgroup = im.uwb.edu.pl
         server string = Theta Samba Server
         max log size = 50
         log level = 0
         os level = 255
         preferred master = yes
         domain master = yes
         local master = yes
         dns proxy = No
         security = user
         encrypt passwords = yes
         guest account = nobody
         hosts allow = 192.168.2.3/255.255.255.0
         load printers = no
         printcap name = lpstat
         printing = SYSV
         bind interfaces only = yes
         interfaces = 192.168.2.3/32
#        socket options = TCP_NODELAY IPTOS_LOWDELAY SO_KEEPALIVE
SO_RCVBUF=8192 SO_SNDBUF=8192

         domain logons = yes
         logon path = \\theta\profile\%u
         logon drive = H:
         logon home = \\theta\%u

         wins support = yes
         netbios name = theta
         unix password sync = yes
         pam password change = yes

[netlogon]
         comment = Netlogon
         path = /export/home1/netlogon
```

```
        browseable = no
        public = no
        writeable = no

[profile]
        comment = Profiles
        path = /export/home1/profiles
        create mode = 0600
        directory mode = 0700
        force group = smbuser
        profile acls = yes
        read only = no

[homes]
        comment = Home Directories
        create mode = 0600
        directory mode = 0700
        read only = no
        browseable = no
```

Dodatek C

Plik /etc/pam.conf

```
PAM configuration
login auth requisite pam_authtok_get.so.1
login auth required pam_dhkeys.so.1
login auth required pam_unix_cred.so.1
login auth required pam_dial_auth.so.1
login auth binding pam_unix_auth.so.1 server_policy
login auth required pam_ldap.so.1

rlogin auth sufficient pam_rhosts_auth.so.1
rlogin auth requisite pam_authtok_get.so.1
rlogin auth required pam_dhkeys.so.1
rlogin auth required pam_unix_cred.so.1
rlogin auth binding pam_unix_auth.so.1 server_policy
rlogin auth required pam_ldap.so.1

rsh auth sufficient pam_rhosts_auth.so.1
rsh auth required pam_unix_cred.so.1
rsh auth binding pam_unix_auth.so.1 server_policy
rsh auth required pam_ldap.so.1

ppp auth requisite pam_authtok_get.so.1
ppp auth required pam_dhkeys.so.1
ppp auth required pam_unix_cred.so.1
ppp auth required pam_unix_auth.so.1
ppp auth required pam_dial_auth.so.1

other auth requisite pam_authtok_get.so.1
other auth required pam_dhkeys.so.1
other auth required pam_unix_cred.so.1
other auth binding pam_unix_auth.so.1 server_policy
other auth required pam_ldap.so.1

passwd auth binding pam_passwd_auth.so.1 server_policy
passwd auth required pam_ldap.so.1

cron account required pam_unix_account.so.1
```

```
other account requisite pam_roles.so.1
other account binding pam_unix_account.so.1 server_policy
other account required pam_ldap.so.1

other session required pam_unix_session.so.1

other password required pam_dhkeys.so.1
other password requisite pam_authtok_get.so.1
other password requisite pam_authtok_check.so.1
other password required pam_authtok_store.so.1
```

Bibliografia

[1] Lightweight Directory Access Protocol,
`http://docs.sun.com/source/816-5616-10/ldap.htm`

[2] iPlanet Server Products,
`http://docs.sun.com/source/816-5572-10/2_instal.htm`

[3] iPlanet Directory Server,
`http://docs-pdf.sun.com/816-6094-10/816-6094-10.pdf`

[4] Dokumentacja komendy,
/usr/lib/ldap/idsconfig

[5] Dokumentacja komendy,
man ldapclient

[6] NFS Administration Guide,
`http://docs.sun.com/app/docs/doc/801-6634/6i10efsk3?a=view`

[7] Robert Eckstein,David Collier-Brown, Peter Kelly, *Samba*, O'Reilly, 2000
http://www.dronet.pl/dronet/samba.php