BIULETYN INSTYTUTU AUTOMATYKI I ROBOTYKI
NR 24, 2007

Organizacja środowiska pracy grupowej z wykorzystaniem serwera *Samba*

Janusz FURTAK, Łukasz STRZELECKI, Kamil RENCZEWSKI

Instytut Teleinformatyki i Automatyki WAT
ul. Kaliskiego 2, 00-908 Warszawa

STRESZCZENIE: W artykule przedstawiono strukturę oprogramowania i sposób konfiguracji serwera *SAMBA* w środowisku systemu *GNU/Linux*. Omówiono metody dostępu do danych zarządzanych przez usługę *SAMBA* z innych systemów oraz dokładnie opisano sposoby tworzenia środowiska do pracy grupowej przy wykorzystaniu tego serwera.

SŁOWA KLUCZOWE: serwer *SAMBA,* praca grupowa

1. Wstęp

Obecnie ze względu na dużą liczbę systemów operacyjnych występujących w różnych wersjach zarówno w przedsiębiorstwach prywatnych, jak i instytucjach publicznych istnieje problem dotyczący współdzielenia zasobów dyskowych. Systemy operacyjne wywodzące się z rodziny Unix wykorzystują powszechnie do współdzielenia tych zasobów sieciowy system plików NFS (*ang. Network File System*) opracowany przez firmę *Sun Microsystems*, natomiast systemy operacyjne produkowane przez *Microsoft Corporation* bazują na serwisach wiadomości blokowych SMB (*ang. Service Message Block*) [[1]]. Próby wykorzystania usługi NFS do współużytkowania tych samych zasobów przez komputery obsługiwane przez systemy operacyjne należące do obu z tych rodzin skazane są na niepowodzenie, gdyż implementacje tego systemu plików przeznaczone dla platformy *Microsoft* są mało efektywne i oferują mały zakres funkcji w stosunku do rozwiązań natywnych. Natomiast w przypadku SMB sytuacja wygląda inaczej. Usługa ta posiada zarówno darmowe, jak i komercyjne implementacje na wszystkie

popularne platformy, tj.:

- Microsoft Windows,
- Sun Solaris,
- Apple Mac,
- GNU/Linux.

Dodatkowo istnieją rozwiązania o otwartych kodach źródłowych, które posiadają zdecydowanie lepsze możliwości pod względem skalowalności i efektywności niż systemy komercyjne. Do takich rozwiązań należy dystrybucja[1] *SAMBA* wywodząca się z oprogramowania wytworzonego w 1991 roku przez Andrew Tridgellow'a i dostępna obecnie dla wszystkich wymienionych powyżej systemów operacyjnych. W dalszej części artykułu zostanie opisana podstawowa konfiguracja tego oprogramowania wykonana w systemie GNU/Linux[2]. W celu pokazania elastyczności i możliwości rozwiązania zostaną przedstawione sposoby tworzenia środowiska do pracy grupowej przy wykorzystaniu serwera *SAMBA*.

2. Struktura oprogramowania SAMBA

Oprogramowanie *SAMBA* opiera się na dwóch protokołach: NetBIOS oraz SMB. Pierwszy z nich opracowany został przez firmę *Sytek* na zlecenie firmy *IBM* w celu rozszerzenia możliwości PC BIOS. Oferuje on następujące trzy usługi:

- *usługa nazw* – umożliwia identyfikowanie komputerów w sieci na podstawie nazw,
- *usługa sesji* - umożliwia przesyłanie strumieni danych (np. odczyt lub zapis pliku),
- *usługa datagramowa* - umożliwia przesyłanie krótkich wiadomości pomiędzy komputerami pracującymi w sieci.

Wymienione usługi wykorzystuje oprogramowanie protokołu SMB, który w swojej oryginalnej postaci opracowany został w 1984 r. przez dr Barry'ego Feigenbaum'a na zlecenie firmy IBM. Od tego czasu przeszedł wiele zmian związanych ze zwiększającymi się wymaganiami użytkowników na przykład zostały dodane możliwości uwierzytelniania użytkowników. W 1996 roku po zaproponowaniu przez firmę *Sun Microsystems* standardu *WebNFS*, firma *Microsoft* opublikowała dokument opisujący *Common Internet File System*

[1] Dystrybucja tutaj jest rozumiana jako konkretna aplikacja programowa opisywanej w artykule usługi.
[2] Konfiguracje przeznaczone dla innych systemów operacyjnych są bardzo podobne.

(CIFS), który był opisem ówczesnej implementacji protokołu SMB. Od tamtego czasu oficjalnie funkcjonuje nazwa SMB/CIFS, a sam protokół nie ulega już tak dynamicznym zmianom jak we wcześniejszym okresie.

SMB jest protokołem warstwy aplikacji bazującym na NetBIOS, dzięki czemu możliwa była jego implementacja w różnych architekturach (rysunek 1). Opisywane w artykule programowanie usługi *SAMBA* jest implementacją protokołu SMB, która bazuje tylko na protokołach TCP/IP [[3]]. Ze względu na dominującą pozycję rodziny protokołów TCP/IP w sieciach komputerowych nie jest to znaczącym ograniczeniem.

<table>
<tr><td colspan="2">OSI</td><td colspan="4"></td><td>TCP/IP</td></tr>
<tr><td>Aplikacji</td><td colspan="4" rowspan="2">SMB</td><td rowspan="3">Aplikacji</td></tr>
<tr><td>Prezentacji</td></tr>
<tr><td>Sesji</td><td>NetBIOS</td><td rowspan="3">NetBEUI</td><td>NetBIOS</td><td>NetBIOS</td></tr>
<tr><td>Transportowa</td><td rowspan="2">IPX/SPX</td><td rowspan="2">DECnet</td><td>TCP, UDP</td><td>TCP, UDP</td></tr>
<tr><td>Sieci</td><td>IP</td><td>IP</td></tr>
<tr><td>Łącza danych</td><td>802.2 802.3
802.5</td><td>802.2
802.3
802.5</td><td>Ethernet II</td><td>Ethernet II</td><td>Ethernet lub inne</td></tr>
<tr><td>Fizyczna</td><td>Fizyczna</td><td>Fizyczna</td><td>Fizyczna</td><td>Fizyczna</td><td>Fizyczna</td></tr>
</table>

Rys. 1. Umiejscowienie protokołu SMB

Serwer *SAMBA* wykorzystuje następujące dwa demony (tzn. niezależnie aplikacje działające w tle, które realizują poszczególne usługi):

- **nmbd** - daemon nazw NetBIOS, odpowiada za poprawne tłumaczenie i rozpoznawanie nazw NetBIOS;
- **smbd** - daemon protokołu SMB, odpowiada za obsługę żądań klientów (dla każdego klienta uruchamiana jest oddzielna kopia daemona *smbd*).

Dane konfiguracyjne dla usługi *SAMBA* niezbędne do prawidłowego działania wymienionych demonów są przechowywane w pliku konfiguracyjnym o nazwie *smb.conf*. Usługa ta wykorzystuje porty podane w tabeli 1.

Tab. 1. Porty i protokoły wykorzystywane przez usługę *SAMBA*

Port	**Protokół**	**Używany przez demona**
135	TCP	Smbd
137	UDP	Nmbd
138	UDP	Nmbd
139	TCP	Smbd
445	TCP	Smbd rozproszonego systemu plików (DFS)

3. Konfiguracja podstawowa

Ze względu na różne sposoby instalacji oprogramowania w różnych systemach operacyjnych (począwszy od plików **.msi*, a skończywszy na pakietach **.deb*) na potrzeby niniejszego artykułu zostało przyjęte założenie, że podstawowa instalacja (bez żadnej konfiguracji) została już przez użytkownika wykonana, a pliki konfiguracyjne serwera *SAMBA* odpowiadające za główną konfigurację oraz hasła użytkowników to odpowiednio:

- */etc/samba/smb.conf*,
- /*etc/samba/smbpasswd*.

Dodatkowo w systemach wywodzących się z rodziny UNIX do montowania zdalnych zasobów udostępnianych przez serwer *SAMBA* konieczny jest program *smbmount*[3], który niejawnie będzie wykorzystywany przez komendę *mount*[4].

3.1. Główny plik konfiguracyjny

Zgodnie ze standardem przyjętym w systemach z rodziny Unix wszystkie pliki konfiguracyjne odnoszące się do usług systemowych powinny być umieszczone w katalogu */etc*, a dokładniej w podkatalogu o nazwie odpowiadającej usłudze. W przypadku serwera *SAMBA* jest to katalog */etc/samba*. Główny plik konfiguracyjny serwera *SAMBA* to *smb.conf*. Podzielony jest on zazwyczaj na sekcje odpowiadające poszczególnym udziałom, czyli udostępnianym zasobom. Każda sekcja rozpoczynana jest słowem kluczowym umieszczonym w nawiasach kwadratowych. Dodatkowo możliwe jest wykorzystanie w pliku konfiguracyjnym specjalnych zmiennych, co jest pomocne przy tworzenia zaawansowanej konfiguracji. Podstawowa postać pliku konfiguracyjnego jest bardzo prosta, a jej postać została przedstawiona na rysunku 2. Umożliwia ona dostęp wszystkim użytkownikom posiadających konto na serwerze do zasobu *public*.

Pierwsza sekcja o nazwie *global* określa opcje globalne odnoszące się do wszystkich aspektów działania serwera, a także do sekcji zdefiniowanych w dalszej części pliku. W przykładzie zostały zdefiniowane następujące pola:

- *netbios name* - nazwa NetBIOS serwera,
- *workgroup* - nazwa grupy roboczej lub domeny w przypadku pracy w architekturze domenowej,
- *server string* - opis serwera NetBIOS.

[3] Program smbmount jest dostępny na stronie WWW projektu SAMBA: http://samba.org.
[4] Zagadnienie to zostanie wyjaśnione w dalszej części artykułu.

```
; informacje ogolne
[global]
 netbios name = ITA
 workgroup = ITANET
 server string = File Server ITA

; katalog publiczny
[public]
 path = /tmp
 browseable = yes
 writable = yes
```

Rys. 2. Przykładowy plik konfiguracyjny serwera SAMBA

Kolejna sekcja o nazwie *public* opisuje zasób o takiej samej nazwie, a wykorzystane parametry oznaczają odpowiednio:

- *path* - ścieżka do udostępnianego zasobu,
- *browsable* - umożliwienie przeglądania zasobu przez wszystkich użytkowników,
- *writable* - umożliwienie zapisywania do udostępnianego katalogu.

Jak łatwo zauważyć taka konfiguracja jest wyjątkowo prosta, ale równocześnie nie zapewnia wszystkich funkcji umożliwiających efektywną i bezpieczną pracę. Podstawową niedogodnością jest brak prywatnych zasobów dla użytkowników. Rozwiązaniem takiego problemu jest wykorzystanie specjalnej sekcji umożliwiającej użytkownikom dostęp do ich prywatnych katalogów domowych znajdujących się na serwerze. Na rysunku 3 została przedstawiona podstawowa konfiguracja udostępniająca katalogi domowe użytkowników.

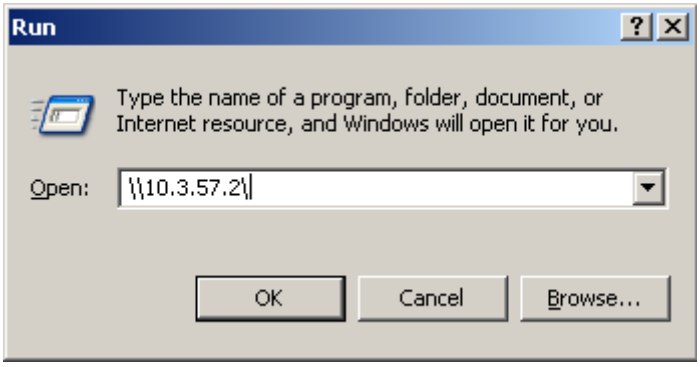
```
; katalogi domowe
[homes]
 path = /home/%u
 browseable = no
 valid users = %S
 read only = no
 guest ok = no
 inherit permissions = yes
```

Rys. 3. Plik konfiguracyjny – udostępnianie katalogów domowych użytkownikom

Warto zwrócić uwagę na fakt, że w tym przypadku zostały wykorzystane wcześniej wspomniane zmienne specjalne serwera *SAMBA*. Zmienna *%S* oznacza nazwę bieżącej usługi, a zmienna *%u* oznacza bieżącego użytkownika, dzięki czemu ścieżka określona przez parametr *path* będzie zawsze wskazywała właściwy dla użytkownika katalog domowy (przy założeniu, że nazwy katalogów domowych użytkowników odpowiadają ich nazwom kont i katalogi te znajdują się w katalogu /home.).

3.2. Dostęp do zasobów z systemu *Windows*

Dostęp do zasobów jest różny w przypadku systemów z rodziny Windows, jak i z rodziny UNIX. Dla systemów produkowanych przez firmę *Microsoft* standardowym postępowaniem jest wybranie z *Menu Start* opcji *Uruchom program* i wpisania nazwy sieciowej żądanego udziału. Przykład przedstawia rysunek 4. Po wykonaniu tych czynności powinno pojawić się okno logowania (rysunek 5). Po zalogowaniu do systemu użytkownicy zgodnie z zapisami utworzonego uprzednio pliku konfiguracyjnego uzyskają dostęp do publicznego katalogu *public* oraz do swojego prywatnego katalogu domowego.



Rys. 4. Montowanie zdalnych zasobów w systemie Microsoft Windows

Rys. 5. Uwierzytelnianie przed dostępem do zdalnych zasobów w systemie Microsoft Windows

3.3. Dostęp do zasobów z systemu UNIX

W systemach z rodziny UNIX standardowym sposobem dostępu do danych udostępnianych przez serwer *SAMBA* jest wykorzystanie programu *smbmount* lub standardowej komendy systemowej *mount*. Przykład został pokazany na rysunku 6.

```
# mount -t smbfs -o username=root //10.3.57.2/root /mnt/samba/
```

Rys. 6. Montowanie zasobu udostępnianego na serwerze SAMBA

Parametry wykorzystane w podanej komendzie oznaczają:

- *-t smbfs* - określenie typu systemu plików jako *smbfs* (czyli SMB File System),
- *-o username=root* - określenie nazwy konta użytkownika, który zamierza uzyskać dostęp do zasobów,
- *//10.3.57.2/root* - sieciowy adres zasobów, do których użytkownik zamierza uzyskać dostęp,
- */mnt/samba* - punkt montowania zdalnych zasobów.

4. Zaawansowana konfiguracja - architektura domenowa

Standardowe udostępnianie plików jest bardzo przydatne, jednak nie jest wystarczające przy codziennej pracy. W celu zapewnienia dodatkowych możliwości zostały wprowadzone konta domenowe, a wraz z nimi tzw. „wędrujące profile". Umożliwiają one użytkownikowi przede wszystkim dostęp do grupowych i prywatnych danych z poziomu dowolnej stacji dołączonej do domeny, o ile użytkownik poprawnie przejdzie proces uwierzytelniania. Technologia ta znana z rozwiązania *Microsoft Active Directory* dostępna jest także w serwerze *SAMBA*. Dodatkowo możliwe jest także przygotowanie dla użytkowników korzystających z oprogramowania *SAMBA* skryptów startowych zarówno grupowych (przeznaczonych dla określonej grupy użytkowników), jak i indywidualnych dopasowujących środowisko pracy użytkownika (np. montujących dodatkowe dyski sieciowe).

W celu odpowiedniego skonfigurowania serwera *SAMBA* należy uzupełnić uprzednio omówioną sekcję *global* o wpisy dotyczące konfiguracji domeny i skryptów startowych oraz dodać dwie sekcje specjalne:

- *netlogon* - logowanie użytkowników do kont domenowych,
- *profiles* - profile użytkowników przechowywane na serwerze.

4.1. Rozbudowa sekcji konfiguracyjnej *global*

Umożliwienie użytkowania kont domenowych przy wykorzystaniu oprogramowania *SAMBA* wymaga skonfigurowania serwera tej usługi jako kontrolera domeny. Wymusza to także w większości przypadków (w zależności od typu systemu operacyjnego klienta) ścisłego określenia sposobu wykorzystania serwerów *WINS*[5], DNS[6], a także tego, czy serwer *SAMBA* będzie pełnił dla klientów rolę serwera czasu. Odpowiednie wpisy, które muszą zostać umieszczone w pliku konfiguracyjnym zostały przedstawione na rysunku 7.

```
[global]
 ; ustawienia PDC
 local master = yes
 os level = 65
 domain master = yes

 ; server wins
 wins support = no
 name resolve order = lmhosts host bcast

 ; inne
 dns proxy = no
 time server = yes
```

Rys. 7. Dodatkowe wpisy w sekcji *global* w pliku konfiguracyjnym umożliwiające działanie serwerowi SAMBA jako kontrolerowi domeny

Wykorzystane w przykładzie na rysunku 7 opcje oznaczają odpowiednio:

- *local master = yes* - ustawienie serwera SAMBA jako lokalnego kontrolera domeny,
- *os level* - dzięki temu parametrowi możliwe jest wskazanie serwera SAMBA jako preferowanego serwera przeglądania[7],
- *domain master = yes* - ustawienie serwera SAMBA jako głównego kontrolera domeny,
- *wins support = no* - wyłączenie obsługi serwerów WINS,
- *name resolve order* - ustanowienie kolejności rozwiązywania nazw,
- *dns proxy = no* - wymuszenie sprawdzania w zasobach serwera DNS nazw, które nie mogły być odnalezione w zasobach serwera WINS,
- *time server = yes* - wymuszenie działania serwera SAMBA jako serwera czasu.

[5] WINS - Windows Internet Name Service
[6] DNS - Domain Name Service
[7] Im większa wartość parametru *os level* (z zakresu 0-255), tym większa szansa, że zostanie on preferowanym serwerem przeglądania.

Pełne wykorzystanie możliwości technologii kont domenowych, czyli automatyczne uruchamianie skryptów logowania, czy też utrzymywanie „wędrujących profili" wymaga jednakże dodatkowej konfiguracji. Serwer za pośrednictwem pliku konfiguracyjnego powinien być poinformowany o położeniu skryptów logowania, domyślnym dysku sieciowym logującego się użytkownika, a także umiejscowieniu danych logowania związanych z profilem użytkownika. Wszystkie wymienione opcje włączane są poprzez dodatkowe wpisy w sekcji *global* przedstawione na rysunku 8.

Dodatkowe opcje widoczne na rysunku 8 definiują:

- *domain logons = yes* - włączenie logowania do domeny,
- *logon script* - określenie skryptu uruchamianego po logowaniu użytkownika,
- *logon path* - określenie katalogu zawierającego dane profilu przy wykorzystaniu zmiennych serwera SAMBA,
- *logon drive* - określenie, który dysk zostanie automatycznie montowany użytkownikowi po zalogowaniu jako domowy,
- *root preexec* - określenie, który program i z jakimi parametrami powinien zostać uruchomiony przed zalogowaniem użytkownika; podana ścieżka określa specjalnie przygotowany skrypt zapisany w języku perl, który zostanie omówiony w dalszej części artykułu,
- *root postexec* - określenie, co powinno być wykonane po zalogowaniu użytkownika; w tym przypadku będzie to usunięcie skryptu logowania użytkownika, który został wytworzony przez skrypt wywołany przed logowaniem użytkownika.

```
[global]
 ; logowanie do domeny
 domain logons = yes
 logon script = %u.bat OR logon.bat
 logon path = \\%L\profiles\%u
 logon drive = H:

 root preexec = /home/netlogon/genlogon.pl %L %u %g %H
 root postexec = [ -f /home/netlogon/%u.bat ] && rm -f
                                              /home/netlogon/%u.bat
```

Rys. 8. Dodatkowe wpisy w sekcji *global* w pliku konfiguracyjnym umożliwiające wykorzystanie technologii kont domenowych SAMBA

4.2. Sekcja konfiguracyjna *netlogon*

Przy wykorzystywaniu skryptów logowania do domeny konieczne jest utworzenie dodatkowego katalogu, w którym będą one umieszczone. Zgodnie z zapisami widocznymi na rysunku 9 w przedstawianej przykładowej konfiguracji został utworzony do tego celu specjalny zasób *netlogon*, do którego ścieżka dostępu to */home/netlogon*. Dokładną konfigurację tego zasobu przedstawia rysunek 10.

Widoczne na rysunku 9 parametry *guest ok* oraz *read only* określają, że zasób ten nie może być przeglądany przez gości (niezalogowanych użytkowników lub użytkowników zalogowanych na specjalne „gościnne" konto), a także nie może być modyfikowany przez użytkowników, co jest skutecznym zabezpieczeniem przed nieuprawnioną modyfikacją.

W przedstawianej konfiguracji w tym zasobie będą znajdowały się skrypty [2] logowania poszczególnych użytkowników, a także główny skrypt, który określony został w sekcji *global* parametrem *root preexec* i będzie wywoływany przed każdym logowaniem dowolnego użytkownika. Zadaniem tego skryptu będzie tworzenie odpowiednich katalogów dla konta logującego się użytkownika w zasobach serwera (jeśli do tej pory nie istniały), a także tworzenie odpowiedniego skryptu logowania. Przykład takiego skryptu został przedstawiony na rysunku 10.

```
; skrypty logowania
[netlogon]
 path = /home/netlogon
 guest ok = no
 read only = yes
 browseable = no
```

Rys. 9. Konfiguracja zasobu *netlogon* zawierającego skrypty logowania użytkowników

```
#!/usr/bin/perl

# deklaracja wykorzystania modułów
use warnings;
use strict;
use File::Path;
#=========================================================
# przypisanie parametrów wejściowych do odpowiednich zmiennych
my $server = $ARGV[0];
my $user = $ARGV[1];
# pobranie dodatkowych danych o użytkowniku z pliku /etc/passwd
my ($login,$pass,$uid,$gid) = getpwnam($user) or die "$user not
in passwd file!\n";
my $group = $ARGV[2];
my $homedir = $ARGV[3];
my $home = "/home";
my $profile = "/home/profiles/".$user;
my $netlogon = "/home/netlogon/".$user.".bat";
#=========================================================
# utworzenie (jeśli nie istnieje) katalogu dla profilu
# użytkownika i nadanie odpowiednich uprawnień
unless( -d $profile){
    mkpath $profile;
    chown $uid, $gid, $profile;
    chmod 0755, $profile;
}

# utworzenie (jeśli nie istnieje) katalogu domowego
# i nadanie odpowiednich uprawnień
unless( -d $homedir){
    mkpath $homedir;
    open(FH, '>', $homedir."/.$user");
    print FH "exist";
    close(FH);
    chown $uid, $gid, $homedir;
    chmod 0700, $homedir;
}
#=========================================================
# utworzenie prostego skryptu logowania
open(FH,'>', $netlogon) or die "Cannot create file
                                                \"$netlogon\": $!";
# wyłączenie wypisywania wykonywanych operacji
print FH "@ echo off\r\n";
# montowanie dysku domowego, jeśli nie został wcześniej
# zamonotwany
print FH "IF NOT EXIST H:\\$user\\.$user NET USE H:
\\\\$server\\$user\r\n";
close(FH);
```

Rys. 10. Skrypt wywoływany przed logowaniem użytkownika

4.3. Sekcja konfiguracyjna *profiles*

Ostatnią sekcją wymaganą przy wykorzystywaniu kont domenowych jest sekcja *profiles*, czyli zasób zawierający profile użytkowników. Jego konfiguracja nieznacznie różni się od dotychczas prezentowanych ze względu na opcje zapewniające zgodność ze wszystkimi wersjami systemów klienckich. Widoczne jest to na rysunku 11.

```
; profile
[profiles]
 path = /home/profiles
 browseable = no
 writeable = yes
 default case = lower
 preserve case = no
 short preserve case = no
 case sensitive = no
 hide files = /desktop.ini/ntuser.ini/NTUSER.*/
 write list = @smbusers @root
 create mask = 0600
 directory mask = 0700
 csc policy = disable
```

Rys. 11. Konfiguracja zasobu *profiles* zawierającego profile użytkowników

Wspomniane opcje zapewniające poprawne działanie w przypadku wszystkich klientów serwera *SAMBA* to:

- *default case, preserve case, short preserve case, case sensitive* - opcje odpowiadają za wymuszenie wielkości znaków w nazwach plików (głównie ze względu na starsze wersje systemów Windows),
- *hide files, write list, create mask, directory mask, csc policy* - opcje dotyczące bezpieczeństwa, głównie praw dostępu do danych.

5. Środowisko pracy grupowej

Możliwości serwera *SAMBA* nie wynikają jedynie z zaimplementowanych w nim funkcji, ale także z możliwości wykorzystania charakterystycznych cech systemu operacyjnego, w którego ramach opisywany serwer działa. Doskonałym przykładem są tutaj dowiązania zarówno twarde, jak i symboliczne oferowane przez systemy z rodziny UNIX. Dzięki ich wykorzystaniu możliwe jest np. stworzenie w prosty sposób efektywnego środowiska pracy grupowej, w którym każdy użytkownik posiada własne konto domenowe (wraz

z wędrującym profilem"), zasoby dyskowe współdzielone z podgrupą roboczą, do której został przypisany (np. w czasie realizacji określonego projektu) oraz zasoby ogólnie dostępne dla wszystkich zalogowanych użytkowników.

5.1. Wykorzystanie możliwości systemu operacyjnego

5.1.1. Struktura katalogów

W celu wytworzenia środowiska opisanego w poprzednim punkcie należy utworzyć odpowiednie poddrzewo katalogów */home*, w którym zostanie uwzględniony specjalny folder o nazwie *WORKGROUPS* zawierający katalogi odpowiadające poszczególnym podgrupom roboczym. Katalog ten powinien mieć specjalnie przydzielone prawo dostępu „x" (execute). Dodatkowo utworzone w nim katalogi odpowiadające podgrupom nie powinny nakładać żadnych ograniczeń związanych z uprawnieniami. Przykładowe poddrzewo katalogów przedstawiają rysunki 12 oraz 13.

```
# ls -l /home
razem 44K
drwxr-xr-x   2  ftp     nogroup  2006-12-22 14:08 ftp
drwxrwx---   3  jf      jf       2007-04-29 12:02 jf
drwx------  15  mak     mak      2005-12-04 22:31 mak
drwxr-xr-x   2  root    root     2007-07-01 23:20 netlogon
drwxrwxrwx   2  nobody  nogroup  2006-11-24 01:38 nogroup
drwxrwxrwx   2  root    nobody   2006-11-15 18:04 pdf-docs
drwxrwxr-x  13  root    nobody   2007-05-04 22:15 profiles
drwxr-xr-x   2  root    root     2006-12-21 10:08 services
drwxr-xr-x   7  st      st       2006-12-27 22:45 st
drwxrwxr-x   3  root    root     2005-11-27 06:12 system
d-----x---   5  root    ita      2007-05-04 22:12 WORKGROUPS
```

Rys. 12. Poddrzewo katalogów /home

```
# ls -l /home/WORKGROUPS/
razem 12K
drwxrwxrwx   2  root    root     2007-05-04 22:12 art
drwxrwxrwx   2  root    root     2007-05-04 22:12 itx
drwxrwxrwx   2  root    root     2007-05-04 22:12 rsw
```

Rys. 13. Poddrzewo katalogów /home/WORKGROUPS

Ostatnim elementem jest utworzenie w katalogach domowych wybranych użytkowników dowiązań symbolicznych do katalogów przeznaczonych dla konkretnych podgrup. I tak na przykład, jeżeli użytkownik *jf* będzie posiadał w swym katalogu domowym dowiązania do folderów *itx* oraz *art* (rysunek 14), a użytkownik *st* będzie posiadał w swoim katalogu domowym jedynie dowiązanie do katalogu *art* (rysunek 15), to będą oni mogli pracować na wspólnych danych jedynie w ramach grupy *art*. Natomiast użytkownik *jf* będzie mógł pracować na współdzielonych danych ze wszystkimi użytkownikami, którzy posiadają dostęp do *art* lub do *itx*.

```
# ls -l /home/jf
razem 12K
lrwxrwxrwx   1  root root 2007-07-02 08:16 art  ->
                                              ../WORKGROUPS/art/
lrwxrwxrwx   1  root root 2007-07-02 08:17 itx  ->
                                              ../WORKGROUPS/itx/
drwxr-xr-x   2  jf   root 2006-12-22 14:01 private_html
drwxr-xr-x   2  jf   root 2006-12-22 14:23 public_ftp
drwxr-xr-x   4  jf   root 2007-01-02 19:10 public_html
```

Rys. 14. Poddrzewo katalogów */home/jf*

```
# ls -l /home/st
razem 12K
lrwxrwxrwx   1  root root 2007-07-02 08:16 art ->
                                             ../WORKGROUPS/art/
drwxr-xr-x   2  st   root 2006-12-22 14:01 private_html
drwxr-xr-x   2  st   root 2006-12-22 14:23 public_ftp
drwxr-xr-x   4  st   root 2007-01-02 19:10 public_html
```

Rys. 15. Poddrzewo katalogów */home/st*

5.1.2. Automatyzacja zadań

Samodzielne tworzenie katalogów dla użytkowników, grup i podgrup dla wielu administratorów jest absorbujące, zwłaszcza w przypadku, gdy ma miejsce duża rotacja personelu. W takim przypadku logicznym wydaje się wykorzystanie skryptów automatyzujących zadania i minimalizujących czas poświęcany przez administratora na przygotowanie środowiska dla nowego użytkownika. Przykładowy skrypt tworzący wszelkie niezbędne katalogi i grupy oraz nadający im właściwe uprawnienia został przedstawiony na rysunkach 16, 17 i 18.

```
#!/bin/bash
# nazwa uzytkownika systemowego i jego grupy
WORKGROUP='ita'
# grupy uzytkownikow
# FORMAT: grupa uzytkownik0 uzytkownik1...
WORKGROUPS[0]="ita jf zs"
WORKGROUPS[1]="art jf st mak"
WORKGROUPS[2]="other zs"
# katalog z grupami uzytkownikow
WRK='/home/WORKGROUPS'
# domyslne haslo dla uzytkownikow
PASSWORD="haslo"
# ustawiamy opis uzytkownika
GECOS='ITX'
# miejsce, gdzie znajduja sie katalogi domowe uzytkownikow
HOMES='/home'
# plik z haslami uzytkownikow serwera SAMBA
SMBFILE='/etc/samba/smbpasswd'
#=========================================================
# funkcja pobierajaca uzytkownikow
function getusers() {
    USERS=""
    for i in $(seq 0 $(echo ${WORKGROUPS[*]} | wc -w));do
        COUNTER=0
        for j in ${WORKGROUPS[$i]};do
            [ -z $j ] && exit
            if [ $COUNTER != 0 ];then
                USERS+=" $j"
            fi
            COUNTER=$(($COUNTER+1))
        done
    done
    echo $USERS|perl -e 'use List::MoreUtils qw/uniq/; $_= <>;
my @a = split(q/\s+/,$_);print join " ", uniq @a;'
}
```

Rys. 16. Skrypt tworzący środowisko dla użytkowników (część I)

W przypadku wykorzystania tego skryptu rola administratora ogranicza się do ustawienia głównych zmiennych tj.:

- *WORKGROUP* - nazwa głównej grupy roboczej,
- *WORKGROUPS* - tablica zawierająca podgrupy robocze i nazwy kont użytkowników do nich przynależących,
- *WRK* - katalog, w którym znajdują się katalogi podgrup[8],
- *PASSWORD* - domyślne hasło dla użytkowników,

[8] Zmiany wymaga jedynie w systemach innych niż *GNU/Linux.*

- *GECOS* - opis GECOS,
- *HOMES* - katalog zawierający katalogi domowe użytkowników.

```
# tworzymy uzytkownika bez mozliwosci logowania
groupadd $WORKGROUP
# tworzymy katalog, w ktorym beda znajdowaly sie katalogi grup
roboczych
mkdir -p $WRK
# zmieniamy grupe, do ktorej przynalezy katalog
chgrp $WORKGROUP $WRK
# minimalne uprawnienia
chmod 010 $WRK

# pobieramy uzytkownikow do utworzenia
USERS=$(getusers)
# haslo systemowe (wystarczy ustalic raz)
PASSWD=`echo $PASSWORD | perl -e 'use Crypt::PasswdMD5; my
$p=<>;chomp $p;
print unix_md5_crypt($p);'`

# petla po wszystkich kontach uzytkowników
for u in $USERS;do
  # sprawdzamy czy uzytkownik istnieje
  grep $u /etc/passwd >> /dev/null
  if [ $? == 1 ]; then
    # dodajemy uzytkownika do systemu
    adduser --ingroup $WORKGROUP $u --disabled-login \
                              --gecos $GECOS --shell /bin/false
    usermod -p "$PASSWD" $u

    # haslo dla samby
    SMBPASSWD=`echo $u:$PASSWORD | perl -e 'use Crypt::SmbHash;
    my $IN = <>;
    (my $USER = $IN) =~ s/:[^:]+$//;
    (my $PASSWD = $IN) =~ s/^[^:]+://;
    my ($lm,$nt);my $uid = (getpwnam($USER))[2]; ntlmgen
                                              $PASSWD, $lm, $nt;
    printf "%s:%d:%s:%s:[%-11s]:LCT-%08X\n", $USER, $uid, $lm,
                                                $nt, "U", time;'`
    # dodajemy uzytkownika do samby
    cat $SMBFILE | pcregrep -v "^$u:" > "$SMBFILE.bak"
    mv -f "$SMBFILE.bak" $SMBFILE
    echo "$SMBPASSWD" >> $SMBFILE
    echo -e "$PASSWORD\n$PASSWORD\n" | smbpasswd -s $u
  else
    echo -e "User \"$u\" already exists! Omiting...\a"
  fi
done
```

Rys. 17. Skrypt tworzący środowisko dla użytkowników (część II)

```
# tworzymy dowiazania
for i in $(seq 0 $(echo ${WORKGROUPS[*]} | wc -w));do
    COUNTER=0
    GRP=""
    for j in ${WORKGROUPS[$i]};do
        [ -z $j ] && exit
        if [ $COUNTER == 0 ];then
            GRP=$j
            mkdir -p $WRK/$GRP
            chmod 777 $WRK/$GRP
        else
            cd $HOMES/$j
            ln -s $WRK/$GRP
        fi
        COUNTER=$(($COUNTER+1))
    done
done
```

Rys. 18. Skrypt tworzący środowisko dla użytkowników (część III)

5.2. Wykorzystanie funkcji serwera *SAMBA*

Wykorzystanie właściwości systemu operacyjnego jest najczęściej stosowanym rozwiązaniem, zwłaszcza w przypadku łączenia usług oferowanych przez serwer *SAMBA* z innymi usługami systemowymi, takimi jak na przykład. FTP. Jednak dla wielu osób może wydawać się to zbędną komplikacją. W takich przypadkach oprogramowanie *SAMBA* zapewnia odpowiednie rozwiązania, które mogą być wdrożone już na poziomie pliku konfiguracyjnego. Przy tego typu konfiguracjach należy w przypadku każdego zasobu posłużyć się dodatkowymi parametrami:

- *valid users* - określa listę użytkowników uprawnionych do dostępu do zasobu;
- *write list* - określa listę użytkowników, którym udostępnia się prawo zapisu do zasobu.

6. Kwestie bezpieczeństwa

Wraz z pojawianiem się nowych zagrożeń protokół SMB był rozszerzany o dodatkowe opcje związane z bezpieczeństwem. Zostały między innymi dodane opcje uwierzytelniania użytkowników na poziomie zasobów, czy też kont systemowych. Wprowadzone zostało podpisywanie protokołu i szyfrowanie

przesyłanych haseł. Wszystkie te opcje są obecnie obsługiwane przez serwer *SAMBA*. Dodatkowo należy zaznaczyć, iż zostały one nawet rozszerzone o dodatkowe możliwości jak na przykład ograniczenie dostępu do zasobów w ramach wybranych podsieci, albo z konkretnego interfejsu. W każdym przypadku uaktywnienie określonej funkcji sprowadza się do dodania do sekcji *global* w pliku konfiguracyjnym odpowiednich wpisów. Zostały one przedstawione na rysunku 19.

```
[global]
; opcje zwiazane z bezpieczenstwem
hosts allow = 127.0.0.0/8 10.3.57.0/24
security = user
encrypt passwords = true
interfaces = eth0
null passwords = no
hide dot files = yes
```

Rys. 19. Włączenie dodatkowych zabezpieczeń w sekcji global

Opcje widoczne na rysunku 19 oznaczają odpowiednio:

- *hosts allow* - określenie poszczególnych adresów IP lub podsieci, z których żądania dostępu będą honorowane,
- *security* - określenie poziomu bezpieczeństwa (możliwe są dwie opcje: *user* - poziom kont systemowych, *share* - poziom zasobu),
- *encrypt passwords=true* - określenie, czy hasła będą szyfrowane,
- *interfaces* - określenie, z których interfejsów sieciowych ruch będzie honorowany,
- *null passwords=no* - określenie, czy możliwe będzie logowanie bez podawania hasła (np. dla konta „gościnnego”),
- *hide dot files=yes* - ukrycie plików konfiguracyjnych specyficznych dla macierzystego systemu operacyjnego (jedynie w wypadku systemu z rodziny UNIX).

Warto zauważyć, iż poprawnie skonfigurowane wymienione powyżej podstawowe opcje zapewniają już stosunkowo wysoki poziom bezpieczeństwa. Dodatkowo przy uwzględnieniu istnienia systemowej zapory sieciowej zbudowanej z wykorzystaniem *IP Tables*[9] oraz programu antywirusowego regularnie sprawdzającego zasoby użytkowników przedstawione rozwiązanie wydaje się gwarantować wystarczające bezpieczeństwo dla większości zastosowań.

[9] Jest to sytuacja standardowa w przypadku zdecydowanej większości systemów z rodziny Unix.

7. Podsumowanie

Najefektywniejsze środowiska pracy zapewniają rozwiązania bazujące na jednorodnych systemach operacyjnych na przykład tylko na produktach firmy *Microsoft*, czy też *Sun Microsystems*. Niestety taka sytuacja jest niezmiernie rzadka ze względu na prowadzone przez instytucje projekty i badania, które ze swej natury związane są z odmiennymi środowiskami. W chwili obecnej nie istnieje idealne rozwiązanie, którym byłoby istnienie sieciowego systemu plików wspólnego dla wszystkich systemów operacyjnych, jednak wydaje się, że sytuacja nie jest krytyczna ze względu na liczne projekty, których celem jest stworzenie substytutu dla takiego rozwiązania. Niewątpliwie do tego typu programów należy opisywany w artykule serwer *SAMBA*. Jest to oprogramowanie stabilne, efektywne i dostępne dla większości popularnych systemów operacyjnych. Dodatkowo przy próbie zapewnienia wszystkich ważnych funkcji zespół rozwijający serwer *SAMBA* nie zbagatelizował kwestii bezpieczeństwa, dzięki czemu powstałe oprogramowanie nadaje się do wykorzystania w środowisku produkcyjnym. Warto także pamiętać, że opisywane oprogramowanie udostępniane jest na zasadach licencji *GPL*, dzięki czemu jest nieodpłatne i posiada publicznie dostępne kody źródłowe. Fakt ten jest o tyle ważny, iż stanowi gwarancję, że do oprogramowania nie zostaną dołączone żadne ukryte funkcje stanowiące potencjalne niebezpieczeństwo dla użytkowników.

Literatura

[1] Sharpe R., Potter T., Morris J.: *SAMBA*, Wydawnictwo Helion, Gliwice, 2002.

[2] Christiansen T., Torkington N.: *Perl - Receptury*, Wydanie drugie, Wydawnictwo Helion, Gliwice, 2004.

[3] Parker T., Sportack M.: *TCP/IP - Księga eksperta*, Wydawnictwo Helion, Gliwice, 2000.

Workgroup environment organization using SAMBA server

ABSTRACT: In this article the software structure and *SAMBA* server configuration procedure in GNU/Linux environment were presented. The access methods from other systems to data supported by *SAMBA* service were discussed and the ways of workgroup environment creating using *SAMBA* server were exactly described.

Key words: SAMBA server, workgroup working

Recenzent: dr hab. inż. Antoni Donigiewicz

Praca wpłynęła do redakcji 20.12.2007