

# **Tytuł pracy :NIS i NIS+ Instalacja i opis konfiguracji**

Autorzy: Maciej Bolec , Rafał Krzysztofiak IV FDS

## **STRESZCZENIE**

Projekt ten zawiera informację na temat NIS i NIS+.

Pierwsze rozdziały tego dokumentu mają na celu ogólne zapoznanie czytelnika z problematyką dotyczącą działania poszczególnych systemów NIS i NIS+:

- wyjaśnienie nazw
- wybór odpowiedniego systemu w zależności od potrzeb użytkownika
- zasada działania

Kolejne rozdziały mają na celu zapoznanie czytelnika z bardziej praktyczną stroną NIS i NIS+ :

- instalacja
- możliwości i konfiguracja poszczególnych systemów
- problemy występujące podczas pracy
- przykłady

## SPIS TREŚCI

Streszczenie .....	1
1. Wstęp(wyjaśnienie pojęć).....	3
2. Wybór odpowiedniego systemu- NIS , NIS+, NYS (wady i zalety).....	3
3. Działanie poszczególnych systemów(Nis i Nis+) .....	4
3.1 Zasada działania systemu NIS .....	4
3.2 Programy NIS'a.....	4
3.3 Pliki NIS'a.....	5
3.4 Zasada działania systemu NIS+.....	6
3.5 Pliki i katalogi NIS+ .....	7
3.6 Tablice NIS+.....	7
3.7 Struktura domen NIS+ (schemat) .....	8
3.8 NIS+ Domian(schemat).....	9
4. Niektóre komendy oraz przykłady ich wykorzystania .....	9
4.1 Komendy .....	9
4.2 Przykłady .....	10
4.3 Składnia komendy: nistbladm .....	10
5. Ustawienie NIS'a(konfiguracja krok po kroku) .....	11
5.1 Sprawdzamy czy jesteśmy serwerem, serwerem slave czy klientem.....	11
5.2 Wykorzystane oprogramowanie .....	11
5.3 Demon ypbind .....	12
5.4 Uzupełnienie(Portmapper RPC).....	14
6. Konfiguracja klienta NIS .....	15
6.1 Ustawianie przy użyciu Tradycyjnego NIS-a.....	15
6.2 Ustawianie klienta NIS używając NYS.....	15
6.3 Ustawianie klienta NIS używając glibc 2.x.....	16
6.4 Zasada działania pliku nsswitch.conf .....	16
7. Konfiguracja serwera NIS .....	17
7.1 Konfiguracja serwera NIS opartego na "ypserv".....	17
7.2 Ograniczenie dostępu do serwera NIS.....	18
7.3 Zasada działania programu rpc.ypxfrd .....	19
7.4 Zmiany haseł i uaktualnianie baz danych NIS'a- program rpc.yppasswdd .....	19
7.5 Weryfikacja instalacji NIS/NYS .....	20
8. Ustawienia NIS+(konfiguracja krok po kroku) .....	21
8.1 Wykorzystane oprogramowanie .....	21
8.2 Ustawienie klienta NIS+.....	21
8.3 NIS+, keylogin, login i PAM.....	22
8.4 Zasada działania pliku nsswitch.conf .....	22
Literatura .....	24

## 1. WSTĘP(WYJAŚNIENIE POJĘĆ)

**NIS** - Jest to skrót od Network Information Service-sieciowa służba informacyjna, element protokołów NFS obsługujący zarządzanie kluczami i sprawdzanie tożsamości klientów sieciowego systemu plików NFS. NIS przechowuje także odwzorowania nazw użytkowników na (zaszyfrowane) hasła oraz odwzorowania nazw komputerów na ich adresy sieciowe.[3]

Celem jego jest dostarczanie informacji, która musi być znana na sieci dla wszystkich komputerów w tej sieci.

W standardowej bibliotece Linux-a - libc - jest obsługa dla NIS, która dalej zwana jest "tradycyjnym NIS-em".

Informacja, która najprawdopodobniej będzie dystrybuowana to:

- login/hasła/katalogi domowe (/etc/passwd)
- informacje o grupach (/etc/group)

Jeśli na przykład informacja o naszym hasle jest zapisana w bazie haseł NIS, będziemy mogli się zalogować na wszystkich maszynach na sieci, które mają uruchomionego klienta NIS.[1]

**NIS+** - network information service (plus :- ) - właściwie to taki rozbudowany nis.

NIS+ został zaprojektowany przez Sun Microsystems Inc. jako zastępca NIS-a z lepszym bezpieczeństwem i lepszą obsługą wielkich instalacji. [1]

Pisząc o NIS i NIS+ należy również wspomnieć o NYS:

**NYS** - jest to nazwa projektu, który powstał z nis+, yp i switch i zarządzany jest przez petera erikssona <peter@ifm.liu.se>. między innymi zawiera powtórna implementację kodu nis-a (=yp), która używa funkcjonalności name service switch biblioteki nys. [1]

## 2. WYBÓR ODPOWIEDNIEGO SYSTEMU- NIS , NIS+ , NYS (WADY I ZALETY)

Wybór między NIS a NIS+ jest prosty - używajmy NIS-a jeśli nie musimy używać NIS+ czy nie musimy stosować zaostrzonego bezpieczeństwa. Z NIS+ jest o wiele więcej kłopotów jeśli chodzi o administrację (jest nim całkiem łatwo zarządzać od strony klienta, ale od strony serwera jest to bardzo trudne). Innym problemem jest fakt, że wsparcie dla NIS+ w Linux-ie jest ciągle w fazie rozwoju - potrzebujemy najnowszej biblioteki glibc, albo musimy poczekać na glibc 2.1. Istnieje wersja zastępcza glibc z obsługą NIS+ dla libc5. [1]

### libc 4/5 z tradycyjnym NIS czy NYS ?

Wybór pomiędzy "tradycyjnym NIS" czy kodem NIS w bibliotece NYS jest wyborem pomiędzy lenistwem i rozwojem a elastycznością i zamięłowaniem do przygody.

Kod "tradycyjnego NIS" jest w standardowej bibliotece C i istnieje już długo i czasem cierpi z powodu swojego wieku i pewnej nieelastyczności.

Kod NIS w bibliotece NYS wymaga rekompilacji biblioteki libc, aby włączyć do niej kod NYS (albo możemy wziąć już skompilowaną wersję libc od kogoś kto już ją skompilował).

Inną różnicą jest to, że "tradycyjny NIS" ma pewną obsługę NIS Netgroups, której NYS nie ma. Z drugiej strony NYS pozwala obsługiwać Shadow Passwords w sposób przezroczysty. "Tradycyjny NIS" nie obsługuje Shadow passwords przez NIS.

Zapomnijmy o tym wszystkim jeśli używamy nowej biblioteki GNU C 2.x (aka libc6). Ma ona rzeczywistą obsługę NSS (Name Switch Service), co czyni ją bardzo elastyczną oraz zawiera obsługę następujących map NIS/NIS+: aliases, ethers, group, hosts, netgroups, networks, pro-

tocols, publickey, passwd, rpc, services i shadow. Biblioteka GNU C nie ma żadnych problemów z shadow password przez NIS. [1]

### 3. DZIAŁANIE POSZCZEGÓLNYCH SYSTEMÓW(NIS I NIS+)

#### 3.1 Zasada działania systemu NIS

W całej sieci musi być przynajmniej jedna maszyna działająca jako serwer NIS. Możemy zrobić więcej serwerów NIS, każdy dla innej "domeny" NIS - albo możemy mieć współpracujące serwery NIS, gdzie jeden ma być głównym serwerem NIS (master) a wszystkie inne są tak zwanymi slave NIS servers (to znaczy dla pewnej "domeny" NIS!) - albo możemy to pomieszać.

Serwery slave posiadają tylko kopię baz danych NIS i otrzymują te kopie od głównego serwera NIS kiedy tylko robione są jakieś zmiany w głównej bazie. W zależności od liczby komputerów w naszej sieci, możemy zdecydować się na instalację jednego lub większej ilości serwerów slave. Kiedy tylko serwer NIS jest unieruchamiany (goes down) albo jest zbyt wolny w odpowiedziach na żądania, klient NIS-a podłączony do tego serwera spróbuje znaleźć ten, który działa albo jest szybszy.

Bazy danych NIS są w tak zwanym formacie DBM, pochodzącym od baz danych ASCII. Na przykład, pliki `/etc/passwd` i `/etc/group` mogą być bezpośrednio zamienione na DBM przy pomocy oprogramowania translacyjnego ASCII-na-DBM ("makedbm" - dostarczany wraz z serwerem). Główny serwer NIS powinien posiadać obie bazy - tak ASCII jak i DBM.

Serwery slave zostaną powiadomione o każdej zmianie w mapach NIS, (poprzez program "yppush") i automatycznie uaktualnią owe zmiany, aby zsynchronizować swoje bazy danych. Klienci NIS nie muszą tego robić ponieważ oni zawsze łączą się z serwerem NIS, aby odczytać informacje zapisane w bazach danych DBM.

Stare wersje ypbind wysyłają adres rozgłoszeniowy (broadcast), aby znaleźć działający serwer NIS. Jest to niebezpieczne ponieważ każdy może zainstalować serwer NIS i odpowiedzieć na takie zapytanie. Nowsze wersje ypbind (3.3 czy mt) pobierają adres serwera z pliku konfiguracyjnego - i nie ma potrzeby wysyłania adresu rozgłoszeniowego.[1]

#### 3.2 Programy NIS'a

**ypserv - server NIS'a**

`ypserv [-d]`

**ypbind - demon komunikacyjny NIS'a**

`ypbind [-s]`

**yppush - polecenie uaktualnienia map na serwerach podrzędnych**

`yppush [-v] [-d domain] mapname`

**ypxfrd - demon uaktualniający mapy**

`ypxfrd`

**ypxfr - polecenie pobrania aktualnych map z serwera**

ypxfr [-f] [-h host] [-d domain] [-s domain]

**ypoll - polecenie pobrania aktualnego numeru wersji map**

ypoll [-h host] [-d domain] mapname

**ypwhich - polecenie pobrania nazwy serwera obsługującego maszynę**

ypwhich [-d domain] [-m] mapname

ypwhich -x

**ypset - polecenie przełączenia się na wskazany serwer**

ypset [-d domain] [-h host] server

**ypcat - polecenie wyświetlenia zawartości mapy**

ypcat [-k] [-d domain] mapname

ypcat -x

**ypmatch -polecenie wyszukania w mapie rekordów o zadanych kluczach**

ypmatch [-d domain] [-k] key mapname

ypmatch -x

[2]

### 3.3 Pliki NIS'a

<i>NIS maps</i>	<i>NIS+ tables</i>	<i>files</i>
passwd.byname	passwd.org_dir	/etc/passwd
passwd.byuid	passwd.org_dir	/etc/passwd
group.byname	group.org_dir	/etc/group
group.bygid	group.org_dir	/etc/group
publickey.byname	cred.org_dir	/etc/publickey
hosts.byaddr	hosts.org_dir	/etc/hosts

hosts.byname	hosts.org_dir	/etc/hosts
mail.byaddr	mail_aliases.org_dir	/etc/aliases
mail.aliases	mail_aliases.org_dir	/etc/aliases
services.byname	services.org_dir	/etc/services
services.byservicename	services.org_dir	/etc/services
rpc.bynumber	rpc.org_dir	/etc/rpc
rpc.byname	rpc.org_dir	/etc/rpc
protocols.bynumber	protocols.org_dir	/etc/protocols
protocols.byname	protocols.org_dir	/etc/protocols
networks.byaddr	networks.org_dir	/etc/networks
networks.byname	networks.org_dir	/etc/networks
netmasks.bymask	netmasks.org_dir	/etc/netmasks
netmasks.byaddr	netmasks.org_dir	/etc/netmasks
ethers.byname	ethers.org_dir	/etc/ethers
ethers.byaddr	ethers.byname	/etc/ethers
bootparams	bootparams	/etc/bootparams
auto.master	auto_master.org_dir	/etc/auto_master
auto.home	auto_home.org_dir	/etc/auto_home
auto.direct	auto_direct.org_dir	/etc/auto_direct
auto.src	auto_src.org_dir	/etc/auto_src

[2]

### 3.4 Zasada działania systemu NIS+

NIS+ to nowa wersja "network information nameservice" z Sun-a. Największą różnicą pomiędzy NIS i NIS+ jest obsługa kodowania danych i autentykacja poprzez bezpieczne RPC w NIS+.

Model nazewnictwa w NIS+ jest zbudowany w postaci struktury drzewiastej. Każdy węzeł w drzewie odpowiada obiektowi NIS+, których mamy sześć typów: katalog, pozycja (entry), grupa, dołączenie, tablica i prywatne.

Katalog NIS+, który tworzy podstawę przestrzeni nazw w NIS+ nazywa się katalogiem "root". Są dwa specjalne katalogi NIS+: `org_dir` i `groups_dir`. Katalog `org_dir` składa się z wszystkich tablic administracyjnych, takich jak `passwd`, `hosts` i `mail_aliases`. Katalog `groups_dir` składa się z grup obiektów NIS+, które używane są do kontroli dostępu. Kolekcja `org_dir`, `groups_dir` i ich katalogów nadrzędnych to domena NIS+.[1]

### 3.5 Pliki i katalogi NIS+

- ❖ `/etc/nsswitch.conf` - główny plik konfiguracyjny
- ❖ `/var/nis` - katalog zawierający wszystkie pozostałe pliki NIS+
- ❖ `/var/nis/NIS_COLD_START` - plik startowy
- ❖ `/var/nis/NIS_SHARED_DIRCACHE` - lokalna pamięć notatnikowa
- ❖ `/var/nis/nazwa_komputera` - katalog zawierający wszystkie obiekty domeny [2]

### 3.6 Tablice NIS+

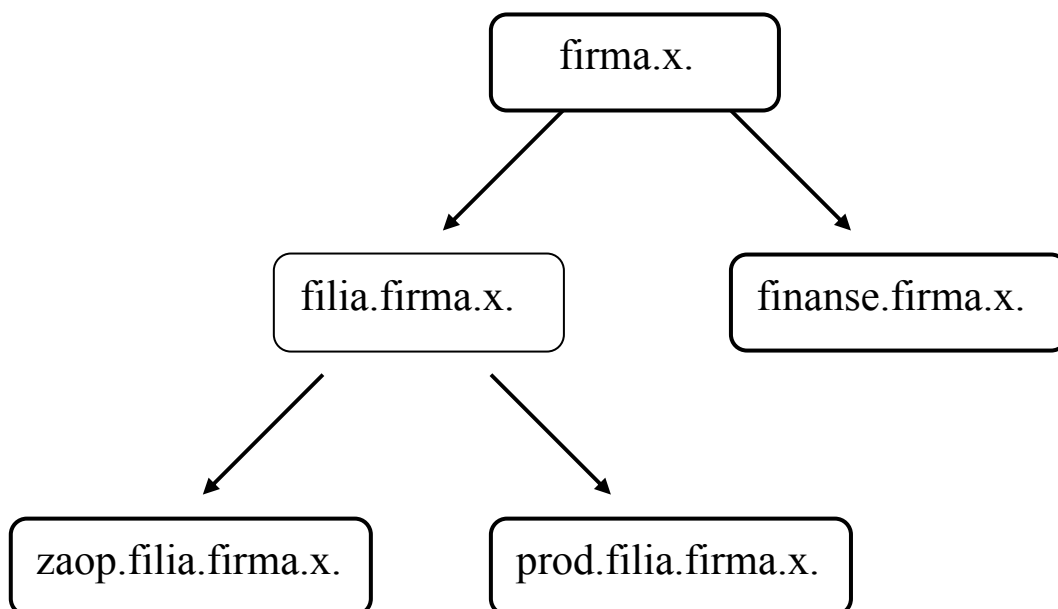
- **hosts** - lista maszyn
- **bootparams** - lokalizacja root'a i swap'a dla wszystkich klientów bezdyskowych
- **passwd** - informacje o użytkownikach i ich hasłach
- **cred** - informacje o kluczach prywatnych i publicznych użytkowników
- **group** - lista grup użytkowników
- **netgroup** - lista grup maszyn i użytkowników
- **aliases** - informacje o aliasach pocztowych
- **timezone** - informacja w jakich strefach czasowych znajdują się komputery
- **networks** - lista nazw sieci
- **netmasks** - lista masek dla każdej z sieci
- **ethers** - adresy sprzętowe komputerów
- **services** - symboliczne nazwy usług sieciowych
- **protocols** - symboliczne nazwy protokołów
- **rpc** - numery usług RPC



➤ **auto home** - plik sterujący montowaniem katalogów użytkowników

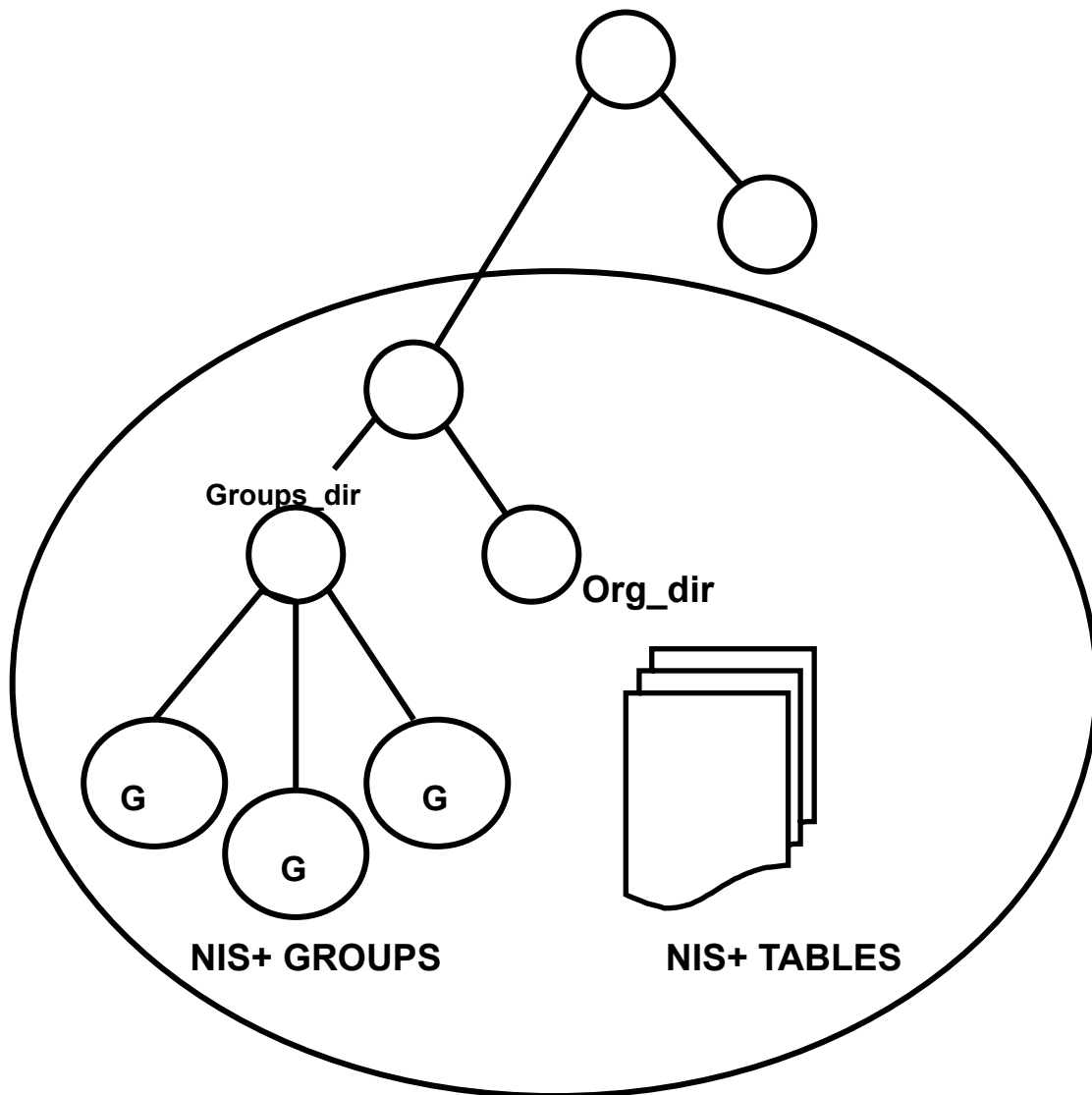
**auto master** - plik sterujący automounterem [2]

### 3.7 Struktura domen NIS+ (schemat)



[2]

### 3.8 NIS+ Domian(schemat)



[2]

## 4. NIKTÓRE KOMENDY ORAZ PRZYKŁADY ICH WYKORZYSTANIA

### 4.1 Komendy

- **domainname** - nadawanie nazwy domeny
- **nisgrpadm** - tworzenie, kasowanie i modyfikacja grup
- **nistbladm** - tworzenie, kasowanie i modyfikacja tablic

- **nisls** - listowanie zawartości katalogów (domen)
- **niscat** – wyświetlanie zawartości tablic
- **nismatch** - przeszukiwanie tablic
- **nismkdir** - tworzenie katalogów (domen)
- **nismkdir** - kasowanie katalogów (domen)
- **nisaddent** - wypełnianie tablic
- **nisinit** - inicjalizacja NIS+
- **nissetup** - utworzenie podstawowych tablic
- **keyser** - demon dystrybucji kluczy
- **rpc.nisd** – główny demon NIS+
- **nis\_achemgr** – zarządca pamięci notatnikowej
- **nisping** - wymuszenie uaktualnienia pamięci notatnikowych
- **nisspasswd** - zmiana hasła w NIS+
- **nissshowcache** - wyświetlenie aktualnej zawartości pamięci notatnikowej [2]

## 4.2 Przykłady

```

nisls org_dir
niscat passwd.org_dir
niscat -o passwd.org_dir
nismatch comp10 passwd.org_dir
nistbladm -m gid=11 "[name=comp10],passwd.org_dir"

```

## 4.3 Składnia komendy: nistbladm

**Create table** – utwórz tablicę

```

nistbladm -c [ -D defaults ] [ -p path ] [ -s sep ]
type colname=[flags][,access] ... tablename

```

**Destroy table** - skasuj tablicę

```

nistbladm -d tablename

```

**Add entry** - dodaj wiersz

```
nistbladm -a|-A [ -D defaults ] colname=value ...
tablename
nistbladm -a|-A [ -D defaults ] indexedname
```

**Modify entry** – zmień wiersz

```
nistbladm -m colname=value ... indexedname
```

**Remove entry** - skasuj wiersz

```
nistbladm -r|-R [ colname=value ... ] tablename
nistbladm -r|-R indexedname
```

**Update attributes** - poprawienie atrybutów

```
nistbladm -u [ -p path ] [ -s sep ] [ -t type ]
[ colname=access ... ] tablename
```

```
indexedname = = [ colname=value, ... ],tablename [2]
```

## 5. USTAWIENIE NIS'A(KONFIGURACJA KROK PO KROKU)

### 5.1 Sprawdzamy czy jesteśmy serwerem, serwerem slave czy klientem

Aby odpowiedzieć na to pytanie musimy rozważyć dwa przypadki:

1. Nasza maszyna będzie częścią sieci z istniejącymi serwerami NIS.
2. Nie mamy jeszcze żadnego serwera NIS w sieci.

W pierwszym przypadku potrzebujemy tylko programów klienta (ypbind, ypwhich, ypcat, yppoll, ypmatch). Najważniejszym programem jest ypbind. Program ten musi być uruchomiony przez cały czas, to znaczy powinien zawsze pojawiać się w liście procesów. Jest to procesdemon i musi być startowany z plików startowych systemu (np. /etc/init.d/nis, /sbin/init.d/ypclient, /etc/rc.local, /etc/rc.d/init.d/ypbind).

Jak tylko ypbind jest uruchomiony, nasz komputer staje się klientem NIS.

W drugim przypadku, jeśli nie mamy serwerów NIS, to będziemy także potrzebowali programu serwera NIS (zwykle zwanego ypserv). Jest jeszcze jeden darmowy serwer NIS, zwany "yps", napisany przez Tobiasza Rebera w Niemczech, który obsługuje concept master-slave, ale ma inne ograniczenia i nie jest wspierany od długiego czasu. [1]

### 5.2 Wykorzystane oprogramowanie

Biblioteka systemowa "/usr/lib/libc.a" (wersja 4.4.2 i nowsza) czy biblioteka dzielona "/lib/libc.so.x" zawierają wszystkie odwołania systemowe niezbędne do skompilowania oprogramowania klienta i serwera NIS. Do biblioteki glibc 2.x, potrzebujemy także biblioteki /lib/libnsl.so.1.

Niektórzy podawali, że NIS działa tylko z "/usr/lib/libc.a" w wersji 4.5.21 i nowszej, więc jeśli chcemy być zabezpieczeni, to nie należy używać starszych bibliotek. Oprogramowanie klienckie NIS możemy znaleźć w:

Adres	Katalog	Nazwa Pliku
ftp.kernel.org	/pub/linux/utils/net/NIS	yp- tools-2.2.tar.gz
ftp.kernel.org	/pub/linux/utils/net/NIS	ypbind-mt-1.4.tar.gz
ftp.kernel.org	/pub/linux/utils/net/NIS	ypbind-3.3.tar.gz
ftp.kernel.org	/pub/linux/utils/net/NIS	ypbind-3.3-glibc5.diff.gz
ftp.uni-paderborn.de	/linux/local/yp	yp-clients-2.2.tar.gz
ftp.icm.edu.pl	/pub/Linux/sunsite/ system/network/admin	yp-clients-2.2.tar.gz

**W Polsce używamy raczej tego ostatniego adresu.**

Jak już mamy oprogramowanie, postępujemy zgodnie z instrukcjami przychodzącymi wraz z nim. yp-clients 2.2 nadaje się do użytku z libc4 i libc5 aż do 5.4.20. libc 5.4.21 i glibc 2.x potrzebują yp-tools 1.4.1 lub nowsze. Nowe yp-tools 2.0 będą działać z każdą biblioteką libc dla Linux-a. Ponieważ był pewien błąd w kodzie NIS-a nie powinniśmy używać libc 5.4.21-35. W zamian użyjemy 5.4.36 lub nowszej, inaczej większość programów YP nie będzie działała. ypbind 3.3 będzie działać także ze wszystkimi bibliotekami. Jeśli używamy gcc 2.8.x lub nowszej wersji, egcs lub glibc 2.x, to należy nałożyć łatę ypbind-3.3-glibc5.diff na ypbind 3.3. Nie należy nigdy używać ypbind z yp-clients 2.2. ypbind-mt jest nowym, wielowątkowym demodem; do działania potrzebuje jądra w wersji 2.2 i glibc 2.1 lub nowszej.[1]

### 5.3 Demon ypbind

Odpowiednim miejscem dla demona ypbind będzie /usr/sbin. Niektórzy mogą mówić, że nie potrzebujemy ypbind w systemie z NYS. Jest to błędne, gdyż ypwhich i ypcat potrzebują go.

Należy to zrobić jako root. Inne programy (ypwhich, ypcat, yppoll, ypmatch) powinny znaleźć się w katalogu dostępnym dla wszystkich użytkowników, zwykle /usr/bin.

Nowsze wersje ypbind posiadają plik konfiguracyjny - /etc/yp.conf. Możemy tam na stałe wpisać serwer NIS .

Potrzebujemy tego pliku także do NYS.

Przykład:

```
ypserver voyager
ypserver defiant
ypserver ds9
```

Jeśli system może rozwinąć nazwę bez NIS, to można użyć tej nazwy. W przeciwnym wypadku trzeba użyć adresu IP. W ypbind 3.3 jest błąd; tylko ostatni adres zostanie użyty (ds9 w tym przykładzie). Wszystkie pozostałe pozycje zostaną zignorowane. ypbind-mt obsłuży to poprawnie i użyje tego, który odpowiedział pierwszy.

Dobrym pomysłem będzie przetestowanie ypbind przed wprowadzeniem go do plików startowych. Aby to zrobić postępujemy według schematu:

**a)** Należy upewnić się, że mamy ustawioną nazwę domeny YP. Jeśli nie, to wydajemy polecenie:

```
/bin/domainname domena.nis
```

Gdzie `domena.nis` powinno być jakimś łańcuchem zwykle nie związanym z DNS-em naszej maszyny! Powodem tego jest, to że zewnętrznym włamywaczom jest wtedy trochę trudniej zdobyć bazę danych z hasłami z naszego serwera NIS. Jeśli nie wiemy jaka jest nazwa domeny NIS w naszej sieci, to należy spytać administratora.

**b)** należy uruchomić `"/usr/bin/portmap"` jeśli nie jest jeszcze uruchomiony.

**c)** tworzymy katalog `"/var/yp"` jeśli go jeszcze nie ma.

**d)** następnie uruchamiamy `"/usr/bin/ypbind"`

**e)** używamy polecenia `"rpcinfo -p localhost"`, aby sprawdzić czy ypbind zarejestrował swój serwis u portmapper-a. Coś takiego powinno się pojawić na ekranie:

```
program vers proto  port
 100000    2   tcp    111  portmapper
 100000    2   udp    111  portmapper
 100007    2   udp    637  ypbind
 100007    2   tcp    639  ypbind
```

lub

```
program vers proto  port
 100000    2   tcp    111  portmapper
 100000    2   udp    111  portmapper
 100007    2   udp    758  ypbind
 100007    1   udp    758  ypbind
 100007    2   tcp    761  ypbind
 100007    1   tcp    761  ypbind
```

Zależnie od wersji ypbind jakiej używamy.

**f)** możemy także uruchomić `"rpcinfo -u localhost ypbind"`. Polecenie to powinno dać mniej więcej taki wynik:

```
program 100007 version 2 ready and waiting
```

lub

```
program 100007 version 1 ready and waiting
program 100007 version 2 ready and waiting
```

Wynik ten zależy od wersji ypbind, którą zainstalowaliśmy. Ważną wiadomością jest tutaj tylko "version 2".

W tym momencie mamy możliwość używać programów klienckich NIS takich jak ypcat itp... Na przykład "ypcat passwd.byname" podaje nam całą bazę danych NIS z hasłami.

**Należy upewnić się, że ustawiliśmy nazwę domeny i stworzyliśmy katalog**

/var/yp

Ten katalog MUSI istnieć, aby ypbind poprawnie się uruchomił.

Aby sprawdzić czy nazwa domeny została ustawiona poprawnie używamy programu /bin/ypdomainname z pakietu yp-tools 2.2. Zawiera on funkcję yp\_get\_default\_domain(), która jest bardziej restrykcyjna. na przykład nie pozwala na domenę "(none)", która jest domyślną w Linux-ie. i stwarza wiele kłopotów.

Jeśli test się powiódł możemy teraz zmodyfikować pliki w skryptach startowych naszego systemu, tak żeby ypbind startował podczas startu systemu.

Należy ponownie uruchomić komputer i obserwować komunikaty podczas startu, żeby zobaczyć czy ypbind właściwie wystartował.[1]

## 5.4 Uzupełnienie(Portmapper RPC)

Aby móc uruchomić jakikolwiek z wymienionych programów, należy uruchomić program /usr/bin/portmap. Niektóre dystrybucje mają skrypt uruchamiający ten demon w plikach startowych /sbin/init.d/ czy /etc/rc.d/. Wszystko co trzeba zrobić, to uaktywnić go i zrestartować komputer.

Portmapper RPC (portmap(8)) jest serwerem, który zamienia numery programowe RPC na numery portów protokołu TCP/IP (albo UDP/IP). Musi być on uruchomiony, aby móc używać na tej maszynie odwołań RPC (co właśnie robi oprogramowanie klienta NIS/NIS+) do serwerów RPC (takich jak serwer NIS czy NIS+). Kiedy serwer RPC jest startowany, poinformuje on portmap-a na których portach nasłuchuje, i jakimi numerami programowymi RPC może służyć. Kiedy klient chce odwołać się przez RPC do danego numeru programowego, najpierw skontaktuje się z portmap-em na maszynie serwerowej, aby określić numer portu, do którego należy wysłać pakiety RPC.

Normalnie, standardowe serwery RPC są startowane przez inetd(8), więc portmap musi działać zanim wystartuje inetd.

Dla bezpieczeństwa portmapper potrzebuje serwisu czasu (Time Service).

Należy upewnić czy serwis ten jest włączony w /etc/inetd.conf na wszystkich hostach:

```
#
# Time service is used for clock synchronization.
#
time    stream  tcp      nowait  nobody  /usr/sbin/tcpd  in.timed
time    dgram  udp      wait    nobody  /usr/sbin/tcpd  in.timed
```

Na koniec jeszcze jedna ważna informacja-trzeba zresetować (kill -HUP) inetd-a po dokonaniu zmian w jego pliku konfiguracyjnym ! [1]

## 6. KONFIGURACJA KLIENTA NIS

### 6.1 Ustawianie przy użyciu Tradycyjnego NIS-a

Do sprawdzania nazw trzeba ustawić (albo dodać) "nis", do linii kolejności sprawdzania w pliku /etc/host.conf.

Na maszynach klienckich dodajemy następującą linię w /etc/passwd:

```
+:::~:
```

Możemy także użyć znaczków +/-, aby włączyć/wyłączyć lub zmienić użytkowników. Jeśli chcemy wyłączyć użytkownika guest dodajemy -guest do swojego pliku /etc/passwd. Chcąc użyć innej powłoki (np. ksh) dla użytkownika "linux" po prostu dodajemy do swojego /etc/passwd "+linux:::~/bin/ksh" (bez cudzysłowów). Pola, których nie chcemy zmieniać muszą pozostać puste. Możemy także użyć Netgroups do kontroli użytkowników.

Na przykład, aby pozwolić tylko miquels, dth, ed i wszystkim członkom grupy sieciowej sysadmins na login, ale żeby dane o kontaktach wszystkich innych użytkowników były dostępne używamy:

```
+miquels:::~:
+ed:::~:
+dth:::~:
+@sysadmins:::~:
-ftp
+:*:::~/etc/NoShell
```

Trzeba zauważyć, że na Linuksie możemy także zmienić pole z hasłem, jak to zrobiliśmy w tym przykładzie. Usunęliśmy także login "ftp", tak więc nie jest on już znany i anonimowe ftp nie będzie działać. [1]

Netgroup wyglądałoby tak:

```
sysadmins (-,software,) (-,kukuk,)
```

**WAŻNE:** Cecha netgroup jest zaimplementowana począwszy od libc 4.5.26. Jeśli mamy wersję biblioteki wcześniejszą niż 4.5.26, każdy użytkownik w bazie danych z hasłami ma dostęp do naszego Linux-a jeśli mamy uruchomiony "ypbind". [1]

### 6.2 Ustawianie klienta NIS używając NYS

Wszystko co jest potrzebne, to to, żeby plik konfiguracyjny (/etc/yp.conf) wskazywał na poprawn-y/e serwer(y). Także plik konfiguracyjny Name Service Switch (/etc/nsswitch.conf) musi być poprawnie ustawiony.

Należy zainstalować ypbind. Nie jest on potrzebny dla libc, ale narzędzia NIS(YP) go potrzebują.

Jeśli chcemy używać cechy włączania/wyłączania użytkownika (+/--guest/+@admins), musimy użyć "passwd: compat" i "group: compat" w pliku nsswitch.conf. W tym przypadku trzeba użyć "shadow: files nis".

Źródła NYS są częścią źródeł libc 5. Kiedy uruchamiamy configure, za pierwszym razem należy odpowiedzieć "NO" na pytanie "Values correct", potem odpowiedzieć "YES", na "Build a NYS libc from nys". [1]



### 6.3 Ustawianie klienta NIS używając glibc 2.x

Glibc używa "tradycyjnego NIS", tak więc trzeba uruchomić ypbind. Plik konfiguracyjny Name Service Switch (/etc/nsswitch.conf) musi być poprawnie ustawiony. Jeśli używamy trybu kompatybilnego z passwd, shadow czy group, musimy dodać "+" na końcu tego pliku, i możemy używać cechy włączania/wyłączania użytkownika. Konfiguracja jest dokładnie taka sama jak pod Solaris-em 2.x.[1]

### 6.4 Zasada działania pliku nsswitch.conf

Plik /etc/nsswitch.conf określa kolejność w jakiej odbywa się sprawdzanie kiedy pojawi się żądanie pewnej informacji, tak samo jak plik /etc/host.conf, który określa kolejność sprawdzania adresów hostów. Na przykład linia:

```
hosts: files nis dns
```

określa, że funkcje sprawdzania adresów hostów powinny najpierw szukać w lokalnym pliku /etc/hosts, potem w bazie NIS i na końcu w DNS-ie (/etc/resolv.conf i named), gdzie jeśli nie znaleziono odpowiedzi pojawia się błąd. Ten plik musi być czytelny dla każdego użytkownika.[1]

Poprawny plik /etc/nsswitch.conf dla NIS:

```
#
# /etc/nsswitch.conf
#
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# Pozycja '[NOTFOUND=return]' oznacza, że poszukiwania powinny
# zakończyć się, jeśli poszukiwania w poprzedniej pozycji nic nie
# dały. Zauważ, że jeśli poszukiwania nie powiodły się z
# jakichś innych powodów (jak nie odpowiadający serwer NIS), to
# poszukiwania są kontynuowane z następną pozycją.
#
# Poprawne pozycje to:
#
#      nisplus          Use NIS+ (NIS version 3)
#      nis              Use NIS (NIS version 2), also called YP
#      dns              Use DNS (Domain Name Service)
#      files            Use the local files
#      db               Use the /var/db databases
#      [NOTFOUND=return] Stop searching if not found so far
#

passwd:      compat
group:       compat
# Jeśli mamy libc5 musimy użyć shadow: files nis
shadow:      compat

passwd_compat: nis
group_compat: nis
```

```
shadow_compat: nis

hosts:          nis files dns

services:      nis [NOTFOUND=return] files
networks:      nis [NOTFOUND=return] files
protocols:     nis [NOTFOUND=return] files
rpc:           nis [NOTFOUND=return] files
ethers:        nis [NOTFOUND=return] files
netmasks:     nis [NOTFOUND=return] files
netgroup:      nis
bootparams:    nis [NOTFOUND=return] files
publickey:     nis [NOTFOUND=return] files
automount:     files
aliases:       nis [NOTFOUND=return] files
```

passwd\_compat, group\_compat i shadow\_compat są obsługiwane tylko przez glibc 2.x. Jeśli nie ma zasad (rules) shadow w /etc/nsswitch.conf, glibc użyje zasady (rule) passwd do sprawdzenia. Jest więcej modułów szukania dla glibc jak hesoid. Więcej informacji można znaleźć w dokumentacji glibc. [1]

## 7. KONFIGURACJA SERWERA NIS

### 7.1 Konfiguracja serwera NIS opartego na "ypserv"

Oprogramowanie serwera NIS można znaleźć na:

Adres	Katatlog	Nazwa pliku
ftp.kernel.org	/pub/linux/utils/net/NIS	ypserv-1.3.6.tar.gz

Ustawianie serwera jest takie samo dla tradycyjnego NIS i NYS.

Należy skompilować oprogramowanie, aby wygenerować programy "ypserv" i "makedbm". Możemy skonfigurować program ypserv, aby używał pliku securenets albo tcp\_wrappers. Tcp\_wrappers są o wiele bardziej elastyczne, ale wielu ludzi ma z tym poważne kłopoty.

Jeśli mamy problemy z ypserv skompilowanym dla tcp\_wrappers, to przekompilujmy go do używania pliku securenets. ypserv --version, który pokaże jaką mamy wersję.

Jeśli uruchamiamy nasz serwer jako główny, trzeba określić jakie pliki mają być dostępne przez NIS i wtedy dodajemy lub usuwamy odpowiednie pozycje w /var/yp/Makefile. Zawsze należy zaglądać do plików Makefile i zapoznać się z opcjami na początku pliku.

Pomiędzy ypserv 1.1 a 1.2 była jedna duża zmiana. Od wersji 1.2, ypserv "kieszkuje" uchwyt plików (file handles). Oznacza to, że zawsze kiedy stworzymy nową mapę musimy wywołać makedbm z opcją -c. Trzeba upewnić się, że używamy nowego /var/yp/Makefile z ypserv 1.2 lub nowszego, albo dodajemy opcję -c do makedbm w pliku Makefile. Jeśli tego nie zrobimy, ypserv będzie wciąż używał starych map zamiast uaktualnionych.

Następnie trzeba zmodyfikować /var/yp/securenets i /etc/ypserv.conf.

Upewnijmy się czy portmapper (portmap(8)) jest uruchomiony i następnie uruchamiamy serwer ypserv.

Polecenie:

```
% rpcinfo -u localhost ypserv
```

powinno pokazać coś takiego:

```
program 100004 version 1 ready and waiting
program 100004 version 2 ready and waiting
```

Linii z "version 1" może nie być; zależnie od wersji ypserv i konfiguracji, której używamy. Tylko w klientach SunOS 4.x jest to wymagane.

Teraz wygenerujemy bazę danych NIS (YP). Na serwerze głównym uruchamiamy:

```
% /usr/lib/yp/ypinit -m
```

na slave upewnijmy się, że działa ypwhich -m.

Potem uruchamiamy:

```
% /usr/lib/yp/ypinit -s masterhost
```

To już wszystko, nasz serwer w tym momencie powinien działać.

-jeśli mamy większe problemy, możemy wystartować ypserv i ypbind i trybie śledzenia (debug) w innym oknie. Komunikaty pojawiające się w tym trybie powinny nam powiedzieć co jest nie tak.

-jeśli musimy uaktualnić mapę, to uruchamiamy make w katalogu /var/yp na master serwerze NIS. Uaktualni to mapę jeśli plik źródłowy jest nowszy i prześle pliki do serwerów slave. Nie należy używać ypinit do uaktualniania map.

Możemy zmodyfikować crontab root-a \*na serwerze slave\* i dodać poniższe linie:

```
20 * * * * /usr/lib/yp/ypxfr_1perhour
40 6 * * * /usr/lib/yp/ypxfr_1perday
55 6,18 * * * /usr/lib/yp/ypxfr_2perday
```

To upewni nas, że mapy NIS są aktualne, nawet jeśli jakieś uaktualnienie zostało przeoczone z powodu np. wyłączenia serwera slave podczas modyfikowania bazy głównej.

Serwer slave możemy dodać kiedykolwiek później. Najpierw upewnijmy się, że nowy serwer slave ma pozwolenie na kontaktowanie się z masterem NIS. Potem uruchamiamy

```
% /usr/lib/yp/ypinit -s masterhost
```

na nowym slawie. Na serwerze master dodajemy nazwę nowego serwera slave do pliku /var/yp/ypservers należy uruchomić make w </var/yp, aby uaktualnić mapy.[1]

## 7.2 Ograniczenie dostępu do serwera NIS

Jeśli chcemy ograniczyć dostęp do naszego serwera NIS będziemy musieli ustawić serwer NIS także jako klienta przez uruchomienie ypbind i dodanie pozycji + do /etc/passwd\_w połowie\_ (halfway) pliku z hasłami. Funkcje biblioteczne zignorują wszystkie normalne pozycje po pierwszej pozycji NIS i i resztę informacji zdobędą przez NIS. W ten sposób obsługiwany jest dostęp NIS.

Przykład:

```
root:x:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:
bin:*:2:2:bin:/bin:
sys:*:3:3:sys:/dev:
sync:*:4:100:sync:/bin:/bin/sync
```

```

games:*:5:100:games:/usr/games:
man:*:6:100:man:/var/catman:
lp:*:7:7:lp:/var/spool/lpd:
mail:*:8:8:mail:/var/spool/mail:
news:*:9:9:news:/var/spool/news:
uucp:*:10:50:uucp:/var/spool/uucp:
nobody:*:65534:65534:noone at all,,,,:/dev/null:
+miquels:::::
+*:::::/etc/NoShell
[ PO tej linii normalni użytkownicy ! ]
tester:*:299:10:Just a test account:/tmp:
miquels:1234567890123:101:10:Miquel van Smooren-
burg:/home/miquels:/bin/zsh

```

Użytkownik tester będzie istniał, ale będzie miał ustawioną powłokę na /etc/NoShell. miquels będzie miał normalny dostęp.

Alternatywnie można zmodyfikować plik /var/yp/Makefile i ustawić NIS, tak żeby używał innego źródłowego pliku z hasłami. W dużych systemach NIS-owe pliki z hasłami i grupami znajdują się zwykle w /etc/yp/. Jeśli tak zrobimy, to zwykle narzędzia jak passwd, chfn, adduser nie będą już działać i będziemy potrzebowali specjalnych własnoręcznie zrobionych wersji.

Chociaż yppswd, ypchsh i ypchfn będą oczywiście działać.[1]

### 7.3 Zasada działania programu rpc.ypxfrd

Program rpc.ypxfrd służy do przyspieszania przesyłu bardzo dużych map z serwera master na slave. Jeśli serwer NIS slave otrzyma informacje, że jest nowa mapa wystartuje program ypxfr, aby ją pobrać. ypxfr przeczyta zawartość mapy z serwera master przy pomocy funkcji yp\_all(). Proces ten może zająć do kilku minut jeśli są bardzo duże mapy, które muszą być zapisane przez biblioteki bazy danych.

Program rpc.ypxfrd przyspiesza proces pobrania przez pozwolenie serwerowi slave na skopiowanie mapy serwera master zamiast tworzenia jej od nowa. rpc.ypxfrd używa protokołu do przesyłu plików na podstawie RPC, tak że nie ma potrzeby na tworzenie nowej mapy.

rpc.ypxfrd może być uruchomiony przez inetd. Ale ponieważ startuje bardzo wolno, powinien być startowany przez ypserv. Musimy wystartować rpc.ypxfrd tylko na serwerze NIS master. [1]

### 7.4 Zmiany haseł i uaktualnianie baz danych NIS'a- program rpc.yppasswdd

Kiedy użytkownicy zmieniają swoje hasła, baza danych NIS z hasłami i przypuszczalnie inne bazy danych NIS, które zależą od bazy danych z hasłami, powinny być uaktualnione. Program "rpc.yppasswdd" jest serwerem, który odpowiedzialny jest za zmiany haseł i uaktualnianie baz danych NIS. rpc.yppasswdd jest zintegrowany z ypserv. Nie potrzebujemy starszych oddzielnych yppasswd-0.9.tar.gz czy yppasswd-0.10.tar.gz i nie należy ich już używać. rpc.yppasswdd z ypserv 1.3.2 ma pełną obsługę shadow. yppasswd jest teraz częścią yp-tool-.2.2.tar.gz.

Musimy uruchomić rpc.yppasswdd tylko na serwerze NIS master. Domyślnie użytkownicy nie mają prawa zmieniać swoich "pełnych nazw" czy powłoki. Możemy na to pozwolić opcją -e chfn czy -e chsh.

Jeśli nasze pliki passwd i shadow nie są umieszczone w innym katalogu niż /etc, musimy dodać opcję -D. Na przykład jeśli umieściliśmy wszystkie pliki źródłowe w /etc/yp i chcemy do-

stąpić swoim użytkownikom możliwość zmiany powłoki, musimy uruchomić `rpc.yppasswdd` z następującymi parametrami:

```
rpc.yppasswdd -D /etc/yp -e chsh
```

lub

```
rpc.yppasswdd -s /etc/yp/shadow -p /etc/yp/passwd -e chsh
```

Nic więcej do zrobienia już nie zostało. Musimy się tylko upewnić, że `rpc.yppasswdd` używa tych samych plików co `/var/yp/Makefile`. Błędy będą logowane za pomocą `syslog`. [1]

## 7.5 Weryfikacja instalacji NIS/NYS

Jeśli wszystko jest w porządku powinniśmy móc zweryfikować instalację kilkoma prostymi poleceniami. Zakładając na przykład, że nasz plik z hasłami jest dostarczany przez NIS, polecenie :

```
% ypcat passwd
```

powinno pokazać zawartość pliku z hasłami z NIS.

Polecenie :

```
% ypmatch userid passwd
```

(gdzie `userid` jest login-em jakiegoś użytkownika) powinno pokazać pozycję w pliku z hasłami dotyczącą tego użytkownika. Programy "ypcat" i "ypmatch" powinny być zawarte w dystrybucji NIS lub NYS.

Jeśli użytkownik nie może się zalogować, należy uruchomić następujący program po stronie klienta:

```
#include <stdio.h>
#include <pwd.h>
#include <sys/types.h>

int
main(int argc, char *argv[])
{
    struct passwd *pwd;

    if(argc != 2)
    {
        fprintf(stderr, "Stosowanie: getpwnam nazwa-uzytkownika\n");
        exit(1);
    }

    pwd=getpwnam(argv[1]);

    if(pwd != NULL)
    {
        printf("name.....: [%s]\n", pwd->pw_name);
        printf("password.: [%s]\n", pwd->pw_passwd);
        printf("user id...: [%d]\n", pwd->pw_uid);
        printf("group id.: [%d]\n", pwd->pw_gid);
        printf("gecos.....: [%s]\n", pwd->pw_gecos);
    }
}
```

```
printf("directory: [%s]\n",pwd->pw_dir);
    printf("shell....: [%s]\n",pwd->pw_shell);
}
else
    fprintf(stderr,"Nie znaleziono uzytkownika \"%s\"!\n",argv[1]);

exit(0);
}
```

Uruchomienie tego programu z użytkownikiem jako parametr, spowoduje wyświetlenie wszystkich informacji jakie uzyska funkcja `getpwnam` dla tego użytkownika. Powinno to pokazać, która pozycja jest niepoprawna. Najpopularniejszym problemem jest zapisanie "&ast;" w polu hasła.

Razem z biblioteką GNU C 2.1 (glibc 2.1) dostarczane jest narzędzie `getent`. Należy użyć tego programu zamiast powyższego na systemie z tą biblioteką. Możemy spróbować:

```
getent passwd
```

lub

```
getent passwd login
```

[1]

## 8. USTAWIENIA NIS+(KONFIGURACJA KROK PO KROKU)

### 8.1 Wykorzystane oprogramowanie

Klient dla NIS+ na Linuxa został napisany dla biblioteki GNU C 2. Jest także wersja dla `libc5`, ponieważ większość komercyjnych aplikacji jest z nią skompilowane i nie można ich przekompilować z biblioteką `glibc`. Z biblioteką tą i NIS+ są problemy: Nie mogliśmy dołączyć statycznie tej biblioteki do programów i programy skompilowane z tą biblioteką nie będą działały z inną.

Musimy ściągnąć i skompilować najnowszą bibliotekę GNU C 2.1 dla platformy Intelu albo biblioteki GNU C 2.1.1 dla platform 64-bitowych. Potrzebujemy także systemu opartego na `glibc` jak RedHat 5.x, Debian 2.x czy SuSE Linux 6.x.

W każdej dystrybucji musimy przekompilować kompilator `gcc/g++`, `libstdc++` i `ncurses`. W RedHat musimy wiele zmienić w konfiguracji PAM. W SuSE 6.0 musimy przekompilować pakiet `Shadow Passwords`.

Aby skompilować bibliotekę GNU C postępujemy zgodnie z instrukcjami dołączonymi do niej. Tutaj możemy znaleźć załataną `libc5`, opartą o źródła NYS i `glibc` zamiast standardowej `libc5`:

Adres	Katalog	Nazwa Pliku
ftp.kernel.org	/pub/linux/utils/net/NIS+	libc-5.4.44-ns1- 0.4.10.tar.gz

[1]

### 8.2 Ustawienie klienta NIS+

Po zainstalowaniu nowej biblioteki `libc` i `nis-tools` należy stworzyć listy uwierzytelniające dla nowego klienta na serwerze NIS+. Następnie trzeba sprawdzić czy działa portmap.

Potem sprawdzamy czy na naszym Linux-ie jest ten sam czas co na serwerze. W bezpiecznym RPC mamy tylko 3 minuty, w ciągu których listy uwierzytelniające są aktualne. Dobrym pomysłem jest uruchomienie na wszystkich hostach xntpd. Potem należy uruchomić

```
domainname domena.nisplus
nisinit -c -H <serwer NIS+>
```

aby zainicjalizować zimny start file.

Teraz należy zmienić nasz plik konfiguracyjny /etc/nsswitch.conf. Trzeba upewnić się, że jedynym serwisem po publickey jest nisplus ("publickey: nisplus") i nic więcej !

Potem uruchamiamy keyserw i upewnijmy się, że zawsze będzie startowany jako następny demon po portmapperze podczas startu. Następnie uruchamiamy:

```
keylogin -r
```

aby zapisać tajny klucz (secretkey) root-a na naszym systemie

"niscat passwd.org\_dir" powinno pokazać teraz wszystkie pozycje w bazie danych z hasłami.[1]

### 8.3 NIS+, keylogin, login i PAM

Kiedy użytkownik się loguje musi ustawić swój tajny klucz dla keyserw-a. Robi się to przez wywołanie "keylogin". Login z pakietu shadow robi to za użytkownika jeśli został skompilowany z biblioteką glibc 2.1. W przypadku login-u PAM musimy zainstalować pam\_keylogin-1.2.tar.gz i zmodyfikować plik /etc/pam.d/login, tak aby używał pam\_unix\_auth, a nie pwdb, które nie obsługuje NIS+.

Na przykład:

```
##PAM-1.0

auth      required      /lib/security/pam_securetty.so
auth      required      /lib/security/pam_keylogin.so
auth      required      /lib/security/pam_unix_auth.so
auth      required      /lib/security/pam_nologin.so
account   required      /lib/security/pam_unix_acct.so
password  required      /lib/security/pam_unix_passwd.so

session   required      /lib/security/pam_unix_session.so
```

### 8.4 Zasada działania pliku nsswitch.conf.

Plik /etc/nsswitch.conf określa kolejność w jakiej odbywa się sprawdzanie kiedy pojawi się żądanie pewnej informacji, tak samo jak plik /etc/host.conf, który określa kolejność sprawdzania adresów hostów. Na przykład linia:

```
hosts: files nis dns
```

określa, że funkcje sprawdzania adresów hostów powinny najpierw szukać w lokalnym pliku /etc/hosts, potem w bazie NIS i na końcu w DNS-ie (/etc/resolv.conf i named), gdzie jeśli nie

znaleziono odpowiedzi, to zwracany jest błąd.

Poprawny plik `/etc/nsswitch.conf` dla NIS:

```
#
# /etc/nsswitch.conf
#
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# Pozycja '[NOTFOUND=return]' oznacza, że poszukiwania powinny
# zakończyć się, jeśli poszukiwania w poprzedniej pozycji nic nie
# dały. Zauważmy, że jeśli poszukiwania nie powiodły się z
# jakichś innych powodów (jak nie odpowiadający serwer NIS), to
# poszukiwania są kontynuowane z następną pozycją.
#
# Poprawne pozycje to:
#
#     nisplus           Use NIS+ (NIS version 3)
#     nis               Use NIS (NIS version 2), also called YP
#     dns               Use DNS (Domain Name Service)
#     files             Use the local files
#     db                Use the /var/db databases
#     [NOTFOUND=return] Stop searching if not found so far
#

passwd:      compat
# dla libc5: passwd: files nisplus
group:       compat
# dla libc5: group: files nisplus
shadow:      compat
# dla libc5: shadow: files nisplus

passwd_compat: nisplus
group_compat:  nisplus
shadow_compat: nisplus
hosts:        nisplus files dns
services:     nisplus [NOTFOUND=return] files
networks:     nisplus [NOTFOUND=return] files
protocols:    nisplus [NOTFOUND=return] files
rpc:          nisplus [NOTFOUND=return] files
ethers:       nisplus [NOTFOUND=return] files
netmasks:    nisplus [NOTFOUND=return] files
netgroup:     nisplus
bootparams:   nisplus [NOTFOUND=return] files
publickey:    nisplus [NOTFOUND=return] files
automount:    files
aliases:      nisplus [NOTFOUND=return] files [1]
```



## LITERATURA

Praca została napisana w oparciu o materiały z zasięgnięte z Internetu oraz o forum dyskusyjne m.in.:

[1] <http://www.ict.pwr.wroc.pl/jtz/Html/NIS-HOWTO.pl.html#toc1>

[2] <http://www.desy.de/~sieversm/ypdoku/ypdoku/ypdoku.html>

[3] <http://wiem.onet.pl/wiem/0164e8.html>