



Old Man GURU Magazine

*Wychodzi bardzo nieregularnie, kiedy wydaje mi się,
że mam coś ciekawego lub pożytecznego do napisania...*

Numer 37/2013

17 czerwiec 2013

Coraz częściej spotyka się instalacje terminalowe wykorzystujące system operacyjny ładowany z serwera sieciowego. Pomimo, że istnieje wiele rozwiązań oferujących taką możliwość celowe wydało mi się szczegółowe omówienie procesu sieciowego startu takiego terminala oraz konfiguracji wszystkich modułów programowych. Omówienie będzie dotyczyło najpopularniejszej obecnie metody wykorzystującej środowisko PXE (Preboot eXecution Environment).

Zaczynamy od serwera DHCP:

Pierwszym etapem jest konfiguracja serwera DHCP. Omówię ją na przykładzie systemu RedHat i jego pochodnych (CentOS, Fedora itp.), lecz dla innych systemów proces ten nie będzie się wiele różnił. Plik konfiguracyjny serwera DHCP dla IPv4 nosi nazwę `dhcpd.conf`. W systemach rodziny RedHat znajdziemy go w katalogu `/etc/dhcp`. W innych systemach (np. SuSE) znajdziemy go bezpośrednio w katalogu `/etc`. Plik ten wygląda mniej więcej w taki sposób:

```
localhost:/etc # more dhcpd.conf
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none; ddns-updates off;
allow booting;

option domain-name "aba.krakow.pl";
option domain-name-servers 192.168.1.1;
option routers 192.168.1.1;

# define rules to identify DHCP Requests from PXE and Etherboot clients.
class "pxe" {
    match if substrng (option vendor-class-identifier, 0, 9) = "PXEClient";
}
class "etherboot" {
    match if substrng (option vendor-class-identifier, 0, 9) = "Etherboot";
}

subnet 192.168.1.0 netmask 255.255.255.0 {
    option broadcast-address 192.168.1.255;
    pool {
        default-lease-time 180; # no long lease time required for booting
        max-lease-time 360; # booted system does its own dhcp request
        server-name "mybootserver";
        next-server mybootserver.; # in case your local DNS only handles
            # unqualified domains keep trailing '.'
        filename "pxelinux.0";
        allow members of "pxe";
        allow members of "etherboot"; # allow etherboot, too
        range 192.168.1.201 192.168.1.211;
    }
}
```

Zmodyfikujemy go nieco dla naszych potrzeb:

Część ogólna – dotycząca wszystkich terminali powinna wyglądać tak:

```
# DHCP Server Configuration file.
# see /usr/share/doc/dhcp*/dhcpd.conf.sample
# see 'man 5 dhcpd.conf'
#
# sieć 192.168.1.0/255.255.255.0
subnet 192.168.1.0 netmask 255.255.255.0 {
  # domyślnie przydzielamy adresy 192.168.1.100-130:
  range 192.168.1.100 192.168.1.130;
  # na jeden dzień
  default-lease-time 86400;
  # poinformujmy że hosty będą korzystać z domeny aba.krakow.pl
  option domain-name "aba.krakow.pl";
  # niech używają naszego routera jako serwera DNS:
  option domain-name-servers 192.168.1.1;
  option routers 192.168.1.1;
  # i dodatkowe info:
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.1.255;
}
```

Administrację systemem można sobie jednak bardzo ułatwić przyporządkowując na stałe adresy IP poszczególnym terminalom. Będziemy je identyfikować za pomocą adresów fizycznych MAC. Dodamy więc następujące wpisy:

```
host tomek0 {
  hardware ethernet 1C:C1:DE:9C:3C:10;
  fixed-address 192.168.1.120;
  option host-name "tomek0";
  filename "pxelinux.0";
}

host tomek1 {
  hardware ethernet 00:21:9B:F4:FC:24;
  fixed-address 192.168.1.121;
  option host-name "tomek1";
  filename "pxelinux.0";
}
.....
```

dzięki którym terminale nie tylko otrzymają określone adresy IP, lecz również nazwy (tomek0, tomek1 itd.). I to właściwie wszystko. Teraz wystarczy jedynie uruchomić serwer DHCP:

```
[root@centos dhcp]# service dhcpd start
Uruchamianie dhcpd: [ OK ]
```

i możemy przejść do następnego etapu konfiguracji.

UWAGA dla korzystających z ruterów z DHCP (np. Neostrada):

Nie ma potrzeby wyposażać naszego serwera w dwie karty sieciowe, choć oczywiście jest to możliwe. Wystarczy, że wyłączymy DHCP w routerze LiveBox i serwer DHCP będzie „wydawał” adresy z tej samej podsieci (w powyższym przykładzie 192.168.1.0), w której znajduje się ruter (192.168.1.1).

Serwer TFTP:

Niestety, sam serwer DHCP nie wystarczy, ponieważ PXE nie potrafi załadować do klienta większych plików (np. jądra i obrazu systemu operacyjnego). Start naszego terminala będzie więc następował w 2 etapach:

- w pierwszym terminal uzyska adres IP i pobierze klienta TFTP,
- w drugim zostanie załadowany program startowy, właściwe jądro systemu oraz obraz głównego systemu plikowego.

Konieczne będzie więc skonfigurowanie serwera TFTP.

W tym celu należy odszukać odpowiedni plik konfiguracyjny. W systemie RedHat 6.x znajduje się on w katalogu `/etc/xinetd.d` i nosi po prostu nazwę `tftp`:

```
# default: off
# description: The tftp server serves files using the trivial file transfer \
#              protocol. The tftp protocol is often used to boot diskless \
#              workstations, download configuration files to network-aware printers, \
#              and to start the installation process for some operating systems.
service tftp
{
    disable     = no
    socket_type = dgram
    protocol    = udp
    wait        = yes
    user        = root
    server       = /usr/sbin/in.tftpd
    server_args = -s /var/lib/tftpboot
    per_source  = 11
    cps         = 100 2
    flags       = IPv4
}
```

W pliku tym musimy podać nazwę katalogu głównego serwera TFTP – w powyższym przykładzie jest to `/var/lib/tftpd`. Opcja `-s` (secure) wymusza zmianę katalogu przy starcie serwera `tftp`. Nie należy zapomnieć o linii „`disable = no`” ponieważ w innym przypadku `xinetd` nie uruchomi nam serwera `tftp`, gdy będzie on potrzebny oraz o restarcie `xinetd`:

```
# /etc/init.d/xinetd restart
```

Nasz serwer oprogramowania dla terminali jest już prawie gotowy. Pozostaje skopiowanie do katalogu głównego serwera TFTP odpowiednich plików. Z oczywistych w dalszym ciągu tego opisu użyję jako przykładu oprogramowania terminalowego ABA-X4 opartego o dystrybucję TinyCore. Aby załadować to oprogramowanie niezbędne nam będą następujące pliki:

- `pxelinux.0`
- `vesamenu.c32` wraz z plikiem konfiguracyjnym
- jądro systemu linux – `vmlinuz`
- obraz głównego systemu plikowego – `core.gz`

Pliki te muszą się znaleźć w głównym katalogu serwera TFTP (`/var/lib/tftpboot`), zaś plik konfiguracyjny boot loadera w podkatalogu `/var/lib/tftpboot/pxelinux.cfg`. Plik ten domyślnie nosi nazwę `default`.



Pliki pxelinux.0, vesamenu.c32 (lub inny, np. menu.c32 lub linux.c32) wraz z plikiem konfiguracyjnym są częścią pakietu syslinux, pliki vmlinuz oraz core.gz to jądro oraz obraz głównego systemu plikowego oprogramowania ABA-X4.

Najprostszy plik konfiguracyjny (default) dla vesamenu.c32 może wyglądać np. tak:

```
default vesamenu.c32

prompt 1
timeout 5

display boot.msg

label vesa
label Core
  menu label ^Core Linux system
  kernel vmlinuz
  append initrd=core.gz
```

Jeśli serwer został poprawnie skonfigurowany po włączeniu terminala powinien on całkowicie automatycznie (5 sekund zwłoki) załadować podstawowy system operacyjny. Jego właściwości zależą od przygotowanego obrazu core.gz. Można oczywiście przygotować obraz, w którym zostaną umieszczone wszystkie konieczne moduły programowe, jednak taki obraz może być dość duży, co nie tylko wydłuży czas startu systemu, lecz również (zwłaszcza dla terminali dysponujących niewielką pamięcią RAM) może skutkować zakłóceniami w pracy.

Poważną niedogodnością jest również konieczność każdorazowego przygotowania nowego obrazu w przypadku uaktualnień lub dołączania nowych modułów programowych. Ogranicza to znacznie elastyczność rozwiązania.

Oprogramowanie ABA-X4 wykorzystuje możliwość podłączania rozszerzeń (extensions). Rozszerzenia te zawierają moduły programowe (np. klienta protokołu RDP, SSH, przeglądarkę Firefox itp.). Szereg rozszerzeń jest dostępnych w Internecie – możliwe jest również przygotowanie rozszerzeń we własnym zakresie. Rozszerzenia te są udostępniane w trybie Read-Only, co zabezpiecza programy przed utratą integralności. Dla terminali i innych urządzeń ABA-X4 uruchamianych w trybie NetBOOT mogą być one przechowywane na serwerze i udostępniane za pomocą NFS.

PXELINUX (część popularnego projektu SYSLINUX) umożliwia „podmontowanie” katalogów zdalnych podczas startu systemu. W tym celu należy zmodyfikować plik default w katalogu pxelinux.conf dodając np. taki wpis:

```
label vesa
  menu label ^Core + NFS
  kernel vmlinuz
  append initrd=core.gz quiet vga=785 lang=pl_PL kmap=pl nfsmount=192.168.1.235:/var/lib/tftpboot/nfs tce=nfs/tce
```

Na serwerze przygotowujemy katalog /var/lib/tftpboot/nfs, oraz podkatalog tce:

```
/var/lib/tftpboot/nfs/tce
```

oraz wpisujemy odpowiednie dane do pliku /etc/exports i udostępniamy:

```
# more /etc/exports
/var/lib/tftpboot/nfs *(rw,no_root_squash) (możliwe jest także ograniczenie eksportu do wybranej podsięci)
# exportfs -a
```



Katalog ten będzie miał następującą strukturę:

```
drwxrwxr-x. 3 1001 ftp 4096 04-18 11:12 boot
-rw-rw-r--. 1 1001 ftp 2484 06-15 12:23 mydata.tgz
-rw-rw-r--. 1 1001 ftp 117 06-17 07:54 onboot.lst
drwxrwxr-x. 2 1001 ftp 4096 04-18 11:13 ondemand
drwxrwxr-x. 2 1001 ftp 4096 04-18 15:31 optional
-rw-rw-r--. 1 1001 ftp 0 04-18 11:13 xwbar.lst
```

Ze względu na wygodę administracji umieszczamy wszystkie pliki związane z obsługą systemu NetBOOT terminali w jednym podkatalogu – będzie możliwe proste utworzenie jego kopii awaryjnej.

Należy zwrócić uwagę na plik mydata.tgz . Zawiera on ustawienia konfiguracyjne i jest tworzony automatycznie podczas konfiguracji systemu i będzie odczytywany podczas każdego startu terminala.

Jeśli nie chcemy odtwarzać własnej konfiguracji i uruchomić konfigurację domyślną należy w linii „append” pliki default umieścić wpis „norestore”. Umożliwi to wprowadzenie oraz zapamiętanie nowej konfiguracji lub dodanie nowych programów lokalnych (extensions).

Korzystanie oprogramowania systemu LINUX (protokół XDMCP):

Ponieważ konfiguracja XDMCP zależy od rodzaju dystrybucji systemu LINUX, którą wykorzystujemy na serwerze zagadnienie to potraktuję bardzo skrótowo. Poniższy, przykładowy opis dotyczy serwerów CentOS lub RedHat i środowiska GNOME:

Należy zmodyfikować plik /etc/gdm/custom.conf w następujący sposób:

```
# GDM configuration storage
```

```
[daemon]
```

```
[security]
```

```
DisallowTCP=false
```

```
[xdmcp]
```

```
Enable=true
```

```
DisplaysPerHost=16
```

```
[greeter]
```

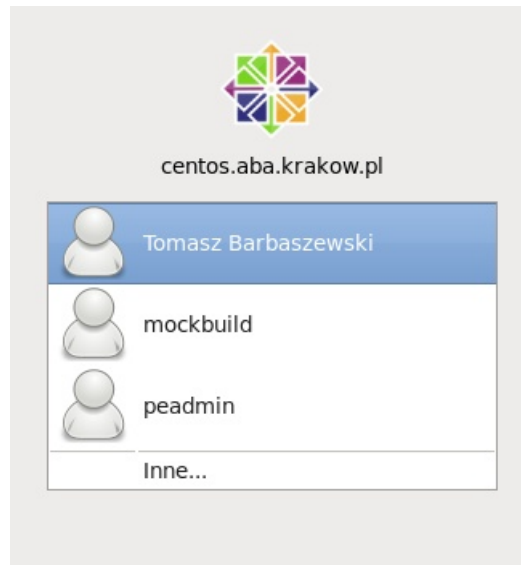
```
[chooser]
```

```
[debug]
```

Następnym krokiem jest dokonanie wpisu do pliku /home/tc/.xsession terminal tak, aby znalazła się w nim linia (jeśli wykorzystujemy serwer Xvesa):

```
Xvesa -br -screen 1024x768x32 -shadow -2button -mouse /dev/input/mice,5 -query <adres_ip_serwera>
```

Dzięki temu wpisowi terminal po załadowaniu oprogramowania połączy się automatycznie z wybranym serwerem i zostanie wyświetlona winiętka logowania wybranego systemu. Należy pamiętać o zapamiętaniu nowej konfiguracji na serwerze!



Po „zalogowaniu się” użytkownik terminala może pracować jako pełnoprawny użytkownik systemu LINUX (w tym przypadku CentOS).

Jeśli korzystamy z serwera Xvesa konieczny będzie jeszcze jeden krok – wprowadzenie mapowania polskiej klawiatury (układ programisty). W tym celu należy umieścić odwołanie do dostarczanego przez firmę ABA skryptu Xmodmap.sh (po rozpakowaniu oprogramowania ABA-X4 znajdzie się on w katalogu /var/lib/tftpboot) w pliku .bashrc użytkownika terminala lub w pliku /etc/bashrc (w tym przypadku mapowanie będzie realizowane dla wszystkich użytkowników).

Zapewne wiele osób będzie zainteresowanych różnicami pomiędzy LTSP a rozwiązaniem ABA-X4 NetBOOT. Zasadniczą różnicą jest znacznie większa elastyczność ABA-X4 dzięki prostej możliwości instalacji dodatkowego oprogramowania (także poprzez sieć). ABA-X4 NetBOOT może korzystać z wielu konsoli na terminalu (do 9 graficznych lub znakowych) w zależności od wielkości pamięci RAM), współpracować na różnych konsolach z różnymi serwerami, może bezpośrednio (bez udziału pośredniczącego systemu LINUX) łączyć się z serwerami MS Windows (z wersją Windows 8 oraz Windows Server 2012) oraz nawet wykonywać programy lokalnie.

Rozwiązanie ABA-X4 NetBOOT jest w pełni kompatybilne z rozwiązaniem ABA-X4 SelfBOOT (oprogramowanie ładowane z lokalnej pamięci FLASH) – może więc pracować także w trybie kiosku internetowego (łącznie z obsługą nakładek dotykowych).

Zachowana kompatybilność z rozwiązaniem TinyCore zapewnia dostęp do ogromnej liczby rozszerzeń dostępnych w Internecie.

Rozwiązanie ABA-X4 znacznie mniej obciąża sieć, ponieważ główny system plikowy (root) jest ładowany do pamięci RAM terminala (jako initrd), a nie obsługiwany przez NFS. Montowane w trybie „tylko do odczytu” programy (np. klient RDP lub VMWare OpenView) są w pełni zabezpieczone przed celowym lub przypadkowym uszkodzeniem.