

Przygotowanie środowiska uruchomieniowego PXE, pozwalającego na bezpośrednie uruchamianie systemów z sieci komputerowej

PXE (Preboot eXecution Environment) to technologia pozwalająca na uruchomienie systemu operacyjnego z innego komputera znajdującego się w sieci lokalnej.

Proces bootowania systemu operacyjnego w technologii PXE wygląda w uproszczeniu następująco:

- BIOS komputera wywołuje BIOS karty sieciowej
- BIOS karty sieciowej próbuje odnaleźć w sieci lokalnej serwer DHCP
- z serwera DHCP klient otrzymuje swój adres IP oraz adres IP serwera TFTP i nazwę pliku do załadowania, zwykle pxelinux.0 (PXE) lub gpxelinux.0 (gPXE)
- komputer klienta pobiera plik bootloadera, oraz inne pliki, które wynikają bezpośrednio z konfiguracji PXE / gPXE. W oryginalnym PXE wszystkie pliki przesyłane są przy pomocy protokołu TFTP, który niestety jest dość wolny. Z tego względu zwykle stosuje się także trzeci, pomocniczy serwer, zwykle NFS, ewentualnie HTTP lub FTP, który służy do przesyłania największych plików. Można też skorzystać z rozszerzonej wersji PXE – gPXE, która sama obsługuje protokół HTTP

Konfiguracja klienta

Konfiguracja komputera klienta ogranicza się do wykonania następujących kroków:

- ustawienia w BIOS-ie bootowania w pierwszej kolejności z sieci lokalnej (LAN),
- podłączenie karty sieciowej komputera do portu LAN routera.

Technologia PXE pochodzi z 1999 r., więc nawet stare komputery ją wspierają. Tak naprawdę, aby uruchomić system operacyjny poprzez PXE, komputer klienta nie musi posiadać żadnych urządzeń magazynujących dane, takich jak dyski twarde, pendrive-y, napędy optyczne czy stacje dyskietek, gdyż system operacyjny ładowany jest bezpośrednio do pamięci RAM. Warto wspomnieć o największej wadze PXE / gPXE jaką jest zapotrzebowanie na pamięć. Wynika ona z zapotrzebowania samego systemu operacyjnego oraz dodatkowej pamięci potrzebnej do obsługi bootloadera, i dodatkowo potrzebna jest pamięć do załadowania obrazu systemu operacyjnego.

Instalacja i konfiguracja serwera PXE w oparciu o system Linux - wersja Ubuntu 16.04

Założenia wstępne dla przykładowej konfiguracji:

- serwer posiada dwie karty sieciowe
- pierwsza karta to enp0s3, która będzie połączona z siecią lokalną (sieć wewnętrzna)
 - adres IP karty: 10.0.0.1/24
- druga karta sieciowa to enp0s8 zapewnia dostęp do internetu, oraz do konsoli.
 - karta ma przypisany dynamiczny adres IP

Głównym zadaniem serwera jest zapewnienie mechanizmu automatycznej konfiguracji kart sieciowych w sieci lokalnej, oraz dodatkowo możliwość uruchamiania i automatycznej instalacji wybranych systemów operacyjnych, korzystając tylko z łącza sieciowego.

Instalacja serwera tftpd

- `apt-get install -y tftpd-hpa`

Domyślny katalog server tftp znajduje się w katalogu `/var/lib/tftpboot`. Warto utworzyć zmienną lokalną `tftpboot_dir` wskazującą na katalog domowy serwera tftp, aby ułatwić sobie przeprowadzenie kolejnych kroków instalacyjnych.

Instalacja serwera dhcpd

- `apt-get install -y isc-dhcp-server`

Domyślny katalog domowy serwera dhcp znajduje się w katalogu `/etc/dhcp`.

Modyfikujemy domyślną konfigurację serwera dhcp na przedstawioną poniżej, zmiany należy wprowadzić w pliku **dhcpd.conf**:

```
subnet 10.0.0.0 netmask 255.255.255.0 {
range 10.0.0.10 10.0.0.200;
option subnet-mask 255.255.255.0;
option broadcast-address 10.0.0.255;
option domain-name "moj_serwer.com";
option domain-name-servers 192.168.1.1;
option routers 10.0.0.1;
filename "pxelinux.0";
}
```

```
host debian-8-pxeboot {
hardware ethernet 08:00:27:2A:23:4F;
fixed-address 10.0.0.2;
}
host ubuntu-16.04-pxeboot {
hardware ethernet 08:00:27:FE:46:C0;
fixed-address 10.0.0.3;
}
```

a następnie uruchamiamy serwer:

- `systemctl enable isc-dhcp-server`
- `systemctl restart isc-dhcp-server`

Konfiguracja obrazów systemów typu netboot

Wykonujemy polecenie (należy wpisać w jednym wierszu):

Pliki systemu Debian-8:

- `$ wget http://ftp.debian.org/debian/dists/jessie/main/installer-amd64/current/images/netboot/netboot.tar.gz -O debian-8-netboot.tar.gz`
- `$ mkdir debian-8-netboot`
- `$ tar xzf debian-8-netboot.tar.gz -C debian-8-netboot`
- `$ sudo cp -a debian-8-netboot/debian-installer ${tftpboot_dir}`

Pliki systemu Ubuntu-16_04:

- `$ wget http://archive.ubuntu.com/ubuntu/ubuntu/dists/xenial/main/installer-amd64/current/images/netboot/netboot.tar.gz -O ubuntu-16.04-netboot.tar.gz`
- `$ mkdir ubuntu-16.04-netboot`
- `$ tar xzf ubuntu-16.04-netboot.tar.gz -C ubuntu-16.04-netboot`
- `$ sudo cp -a ubuntu-16.04-netboot/ubuntu-installer ${tftpboot_dir}`

Zmienna `tftpboot_dir` powinna wskazywać na katalog domowy serwera tftp.

Konfiguracja pxelinux

Katalog domowy serwera tftp zawiera następujące pliki:

- `pxelinux.0` - bootowalny obraz plik
- `ldlinux.c32` - biblioteka wykorzystywana przez `pxelinux.0`

- pxelinux.cfg/default - symboliczny link do syslinux.cfg
- boot-screens/syslinux.cfg - plik konfiguracyjny pxelinux.0
- boot-screens/menu.cfg - menu startowe
- boot-screens/vesamenu.c32 - program obsługi obrazu VESA
- boot-screens/libcom32.c32 - biblioteka wykorzystywana przez vesamenu.c32
- boot-screens/libutil.c32 - biblioteka wykorzystywana przez vesamenu.c32
- debian-installer - instalator systemu linux - dystrybucja Debian 8
- ubuntu-installer - instalator systemu linux - dystrybucja Ubuntu 16.04

Wykonanie poniższych poleceń spowoduje, że pliki boot-screens/syslinux.cfg i boot-screens/menu.cfg zostaną utworzone, a pozostałe pliki zostaną skopiowane z obrazów netboot:

- \$ sudo cp ubuntu-16.04-netboot/ubuntu-installer/amd64/pxelinux.0 \${tftpboot_dir}/
- \$ sudo cp ubuntu-16.04-netboot/ubuntu-installer/amd64/boot-screens/ldlinux.c32 \${tftpboot_dir}/
- \$ sudo mkdir \${tftpboot_dir}/boot-screens
- \$ sudo cp ubuntu-16.04-netboot/ubuntu-installer/amd64/boot-screens/libcom32.c32 \${tftpboot_dir}/boot-screens
- \$ sudo cp ubuntu-16.04-netboot/ubuntu-installer/amd64/boot-screens/libutil.c32 \${tftpboot_dir}/boot-screens
- \$ sudo cp ubuntu-16.04-netboot/ubuntu-installer/amd64/boot-screens/vesamenu.c32 \${tftpboot_dir}/boot-screens
- \$ sudo mkdir \${tftpboot_dir}/pxelinux.cfg
- \$ cd \${tftpboot_dir}/pxelinux.cfg
- \$ sudo ln -s ../boot-screens/syslinux.cfg default

boot-screens/syslinux.cfg

```
path boot-screens
include boot-screens/menu.cfg
default boot-screens/vesamenu.c32
prompt 0
timeout 100
```

boot-screens/menu.cfg

Wyświetla następujące menu:

- Debian 8 automated install
- Ubuntu 16.04 automated install

- o Debian 8 manual install
- o Ubuntu 16.04 manual install

Pozycje instalowane automatycznie korzystają z odpowiednio przygotowanych plików "preseed".

Zawartość pliku menu.cfg:

```
menu hshift 13
menu width 49
menu margin 8
menu tabmsg
menu title Installer boot menu
label auto-debian-8
menu label ^Debian 8 automated install
kernel debian-installer/amd64/linux
append auto=true priority=critical vga=788 initrd=debian-
installer/amd64/initrd.gz
preseed/url=tftp://10.0.0.1/preseed/debian-8-preseed.cfg
label auto-ubuntu-16.04
menu label ^Ubuntu 16.04 automated install
kernel ubuntu-installer/amd64/linux
append auto=true priority=critical vga=788 initrd=ubuntu-
installer/amd64/initrd.gz
preseed/url=tftp://10.0.0.1/preseed/ubuntu-16.04 preseed.cfg
preseed/interactive=false
menu begin debian-8
menu title Debian 8
label mainmenu
menu label ^Back..
menu exit
include debian-installer/amd64/boot-screens/menu.cfg
menu end
menu begin ubuntu-16.04
menu title Ubuntu 16.04
label mainmenu
menu label ^Back..
menu exit
include ubuntu-installer/amd64/boot-screens/menu.cfg
menu end
```

Pliki konfiguracyjne preseed

Pliki konfiguracyjne preseed powinny zostać umieszczone w podkatalogu serwera tftp, `/${tftpboot_dir}/preseed`.

- `sudo mkdir ${tftpboot_dir}/preseed`

Konfiguracja dla systemu Ubuntu 16.04

Główne parametry pliku konfiguracyjnego:

- hasło użytkownika root - "ubuntu",
- zwykły użytkownik, login - ubuntu, hasło - ubuntu,

Zawartość pliku: `/${tftpboot_dir}/preseed/ubuntu-16.04-preseed.cfg`

```
d-i debian-installer/locale string en_US
d-i debian-installer/language string en
d-i debian-installer/country string JP
d-i keyboard-configuration/xkb-keymap select jp106
d-i passwd/user-fullname string
d-i passwd/username string ubuntu
d-i passwd/root-password password ubuntu
d-i passwd/root-password-again password ubuntu
d-i passwd/user-password password ubuntu
d-i passwd/user-password-again password ubuntu
d-i user-setup/allow-password-weak boolean true
d-i netcfg/choose_interface select auto
d-i netcfg/get_hostname string unassigned-hostname
d-i netcfg/get_domain string unassigned-domain
d-i mirror/country string manual
d-i mirror/http/hostname string http://jp.archive.ubuntu.com
d-i mirror/http/directory string /ubuntu
d-i mirror/http/proxy string
d-i clock-setup/utc boolean true
d-i clock-setup/ntp boolean true
d-i time/zone string Asia/Tokyo
d-i partman/confirm boolean true
d-i partman/choose_partition select finish
d-i partman/confirm_nooverwrite boolean true
d-i partman-auto/disk string /dev/[sv]da
d-i partman-auto/method string lvm
d-i partman-auto/choose_recipe select atomic
d-i partman-lvm/device_remove_lvm boolean true
d-i partman-lvm/confirm boolean true
```

```
d-i partman-lvm/confirm_nooverwrite boolean true
d-i partman-auto-lvm/guided_size string max
d-i partman-partitioning/confirm_write_new_label boolean true
d-i grub-installer/grub2_instead_of_grub_legacy boolean true
d-i grub-installer/only_debian boolean true
d-i grub-installer/bootdev string /dev/[sv]da
d-i pkgsel/update-policy select none
d-i finish-install/reboot_in_progress note
```

Konfiguracja dla systemu Debian 8

Główne parametry pliku konfiguracyjnego:

- hasło użytkownika root - "debian",
- zwykły użytkownik, login - debian, hasło - debian,

Zawartość pliku `$(tftpbboot_dir)/preseed/debian-8-preseed.cfg`

```
d-i debian-installer/locale string en_US
d-i debian-installer/language string en
d-i debian-installer/country string JP
d-i keyboard-configuration/xkb-keymap select jp106
d-i passwd/user-fullname string
d-i passwd/username string debian
d-i passwd/root-password password debian
d-i passwd/root-password-again password debian
d-i passwd/user-password password debian
d-i passwd/user-password-again password debian
d-i user-setup/allow-password-weak boolean true
d-i netcfg/choose_interface select auto
d-i netcfg/get_hostname string unassigned-hostname
d-i netcfg/get_domain string unassigned-domain
d-i mirror/country string manual
d-i mirror/http/hostname string ftp.jp.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string
d-i clock-setup/utc boolean true
d-i clock-setup/ntp boolean true
d-i time/zone string Asia/Tokyo
d-i partman/confirm boolean true
```

```
d-i partman/choose_partition select finish
d-i partman/confirm_nooverwrite boolean true
d-i partman-auto/disk string /dev/[sv]da
d-i partman-auto/method string lvm
d-i partman-auto/choose_recipe select atomic
d-i partman-lvm/device_remove_lvm boolean true
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true
d-i partman-auto-lvm/guided_size string max
d-i partman-partitioning/confirm_write_new_label boolean true
d-i grub-installer/grub2_instead_of_grub_legacy boolean true
d-i grub-installer/only_debian boolean true
d-i grub-installer/bootdev string /dev/[sv]da
d-i pkgsel/update-policy select none
d-i finish-install/reboot_in_progress note
```

Konfiguracja bramy sieciowej oraz mechanizmu NAT

Jeśli chcemy zapewnić dostęp do internetu komputerom w sieci lokalnej, należy przeprowadzić konfigurację bramy, oraz mechanizmu translacji adresów NAT.

Do tego zadania można wykorzystać funkcjonalności, jakie zapewnia dostępny firewall, w tym przypadku jest to program iptables.

Przykładowa konfiguracja bramy oraz nat:

```
route add -net 10.0.0.0/24 dev enp0s3
sysctl -w net.ipv4.ip_forward=1
iptables -F
iptables -t nat -F
iptables -t nat -A POSTROUTING ! -d 10.0.0.0/24 -o enp0s8 -j SNAT --
to-source <ip_karty_enp0s8>
```

Uwaga !!! - Dostępność repozytoriów z pakietami dla danej dystrybucji jest zwykle ograniczona czasowo. W przypadku wystąpienia trudności z dostępem do pakietów instalacyjnych, zalecane jest zaktualizowanie dystrybucji do nowszej wersji systemu.