Kurs administrowania systemem Linux 2019

Lista zadań na pracownię nr 13

Na zajęcia 27 maja 2019

Zadanie 1 (3 pkt). Przeczytaj podręczniki systemowe

`namespaces(7)`, `pid_namespaces(7)`, `user-namespaces(7)`, `namespace.conf(5)`, `unshare(2)`, `unshare(1)`, `setns(2)`, `nsenter(1)` i `lsns(8)`

oraz dokumentację jądra:

`https://www.kernel.org/doc/Documentation/unshare.txt`

i przygotuj krótkie omówienie implementacji przestrzeni nazw w Linuksie.

Za pomocą polecenia `debootstrap(8)` utwórz w katalogu `/target/` małą instalację Debiana. Wykonaj polecenia

```
host# unshare -imnpuf --mount-proc chroot /targret/ /bin/bash
guest# mount -t proc proc /proc
guest# ps -ef
guest# mount
guest# ip link
```

Zauważ, że uruchomiona powłoka twierdzi, że ma `PID=1` i `PPID=0` i „widzi" tylko własne procesy potomne. Zauważ, że punkt montażowy `/proc` jest jedynym punktem montażowym widocznym dla powłoki, a ponieważ został utworzony w jej przestrzeni nazw, to nie jest widoczny dla procesów poza powłoką. Jakie interfejsy sieciowe „widzi" ta powłoka? Wykonaj w systemie gospodarza polecenie `nsenter(1)` i dodaj do przestrzeni procesów gościa drugą powłokę. Co „widzi" druga powłoka? Jak ją widzi oryginalna powłoka gościa? Jaki ma PID? Za pomocą polecenia `lsns(1)` sprawdź, jakie przestrzenie nazw znajdują się w systemie.

Zadanie 2 (3 pkt). Przeczytaj podręczniki systemowe `croups(7)` i `cgroup-namespaces(7)` oraz dokumentację pakietu `cgroup-tools` (w tym poleceń `cgcreate(1)`, `cgget(1)`, `cgset(1)`, `cgexec(1)`, `lscgroup(1)` itp.) i dokumentację jądra:

`https://www.kernel.org/doc/Documentation/cgroup-v2.txt`
`https://www.kernel.org/doc/Documentation/cgroup-v1/*`

(w szczególności plik `cgroups.txt` zawierający opis grup zarządzania oraz pliki opisujące różne kontrolery) i przygotuj krótkie omówienie implementacji grup zarządzania w Linuksie. Napisz w C niewielki program `greedy`, który alokuje pamięć w ilości przekazanej mu jako parametr wywołania i kończy działanie wypisując komunikat, czy alokacja się powiodła, czy nie. Wykonaj polecenia

```
cgcreate -g memory:mygroup
echo 16000000 > /sys/fs/cgroup/memory/mygroup/memory.limit_in_bytes
echo 0 > /sys/fs/cgroup/memory/mygroup/memory.swappiness
cgexec -g memory:mygroup ./greedy 10000000
cgexec -g memory:mygroup ./greedy 20000000
```

Wyjaśnij, co się po kolei wydarzyło.

Zadanie 3 (2 pkt). Zapoznaj się z interfejsem przestrzeni nazw i grup zarządzania oferowanym przez SystemD, w szczególności z poleceniami `systemd-cgls(1)`, `systemd-cgtop(1)` i `systemd-nspawn(1)` oraz konfiguracją gospodarki zasobami w SystemD: `systemd.resource-control(5)`. Przygotuj ich krótkie omówienie. Za pomocą polecenia `systemd-nspawn(1)` uruchom instalację Debiana z katalogu `/target/` z zadania 1.

Zadanie 4 (2 pkt). Zapoznaj się z opisem kontenerów LXC: `lxc(7)` i podstawowymi poleceniami do zarządzania nimi: `lxc-create(1)`, `lxc-start(1)`, `lxc-stop(1)`, `lxc-attach(1)`, `lxc-console(1)`, `lxc-ls(1)`, `lxc-info(1)`, `lxc-monitor(1)` i przygotuj krótkie omówienie sposobów korzystania z kontenerów. Za pomocą polecenia

```
host# lxc-create -n guest1 -t debian -- -r stretch
```

utwórz kontener zawierający instalację Debiana Stretch. Na fizycznej maszynie utwórz interfejs `br0` typu *bridge*, nadaj mu adres prywatny, np. `10.0.1.1/24` i skonfiguruj kontener wirtualny tak, żeby jego interfejs wirtualny `veth0` był zmostkowany z `br0`. Nadaj interfejsowi `veth0` adres, np. `10.0.1.2/24`. Uruchom kontener i połącz się z nim za pomocą `ssh`. Za pomocą odpowiednich poleceń rozważanych w poprzednich zadaniach (`lsns` itp.) sprawdź, jakie przestrzenie nazw i grupy zarządzania zostały utworzone przez LXC dla kontenera `guest1`. Skonfiguruj go tak, żeby automatycznie uruchamiał się przy każdym starcie maszyny.

Zadanie 5 (2 pkt). Domyślnie LXC nie izoluje przestrzeni nazw użytkowników kontenera. Kontenery nieuprzywilejowane są bardziej bezpieczne, ale ich konfiguracja wymaga wykonania nieco większej liczby czynności. Załóż użytkownika `vm` w systemie macierzystym. Utwórz kontener, którego właścicielem jest użytkownik `vm` (dystrybucję Linuksa możesz wybrać według uznania). Uruchom ten kontener. Sprawdź, że wszystkie procesy kontenera (w tym jego `init`) oraz monitor LXC dla niego pracują jako użytkownik `vm`.

Zadanie 6 (2 pkt). Zapoznaj się z programem Firejail i przygotuj jego krótkie omówienie. Zademonstruj jak można izolować aplikacje za jego pomocą na przykładzie programów Firefox i LibreOffice. Pokaż, jak odebrać programowi Firefox prawo do odczytywania zawartości katalogu `~/.ssh/`, a programowi LibreOffice — możliwość połączeń internetowych.

Zadanie 7 (2 pkt). Zapoznaj się z biblioteką `libvirt` i poleceniem `virsh(1)` i przygotuj ich krótkie omówienie. Zademonstruj jak można zarządzań kontenerami LXC z poprzednich zadań za pomocą powłoki `virsh`.

Zadanie 8 (2 pkt). Zapoznaj się z programem Vagrant i przygotuj jego krótkie omówienie. Zademonstruj jak można zarządzań kontenerami LXC z poprzednich zadań za jego pomocą.