Lab Exercise – systemd 1

Plan: in the following exercises we will explore important systemd commands, that help to setup a debugginging environment for systemd

prerequisites:

IPADDRESS of your VM named “trouble” is 192.168.2.160

see if existing SSH keys are present:

```
ls -al ~/.ssh
```

if not:

```
ssh-keygen
ssh-copy-id 192.168.2.160
```

now you are able to login passwordless to the VM “trouble” via

```
ssh -X root@192.168.2.160 or  ssh -X root@trouble
```

add an entry into /etc/hosts if hostname “trouble” is not listed

- 0.9 there are over 130 man pages.

How do you quickly determine the right manpage for an unknown option, parameter, directive?

- 1.0 is the journal persistent?

Per default logs are saved in /run/log/journal/, run is ephemeral

we have already booted the VM once, execute:

```
journalctl --list-boots
```

It will only display the most recent boot, so all journal information from previous boots is lost per default!

Directory /var/log/journal/ needs to exist, so that systemd-journald-service can save its data there.

Directory /etc/systemd lists some essential configuration files with its default values.

Open /etc/systemd/journald.conf and change the line that starts with Storage=

```
vi /etc/systemd/journald.conf
[...]
[Journal]
Storage=persistent
#Compress=yes
[...]
```

restart the journal service and display the last boots:

```
systemctl restart systemd-journald.service
```

verify, that directory /var/log/journal has been created.

Reboot and verify:

```
journalctl --list-boots
```

From now on all boots will be listed

During boot startup no messages are displayed.

To change this adapt the grub2 configuration to exclude options: “splash=silent quiet”

edit /etc/default/grub

```
GRUB_CMDLINE_LINUX_DEFAULT="resume=/dev/vda1 showopts"
```

and execute:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

and reboot.

Never edit `/boot/grub2/grub.cfg directly!`

• 1.1 In your KVM VM within virt-manager execute Ctrl+Alt+F10

by default log messages received by the journal daemon are forwarded to syslog, here the messages are also forwarded to the system console.

Which configuration file was used to enable that?

• 1.2 check if lamp server is installed as pattern

```
systemctl status apache2
```

execute:

```
systemctl
```

apache2 is not listed

• 1.3 Now execute:

```
systemctl list-unit-files|grep apache2
```

where does systemd define that a service is disabled although it is installed(as in this case apache2)?

• 1.4 What files are changed when you execute the following commands?

```
systemctl enable apache2
```

Take good note, which files were created by the command

```
systemctl disable apache2

systemctl mask apache2

systemctl start apache2
```

this will not work, because service is masked

```
systemctl unmask apache2

systemctl enable apache2

systemctl start apache2
```

rc scripts for Backward compatibility

• 1.5 how would you list all masked services?

• 2.0 execute:

```
rccron status
```

for Backward compatibility it is still possible to execute rc scripts via a symlink

• 2.1 what does the command actually execute ?

Hint: softlink

```
ls -l /usr/sbin/rc*
```

• 2.3 are these real init scripts?

To find out what the “service” command is doing, we will use shell debugging

include a "set -x" statement in /usr/sbin/service after the initial comments:

```
cp -p /usr/sbin/service /usr/sbin/service.orig

sed -i '4 a set -x' /usr/sbin/service
```

execute the command again:

```
rccron status
```

examine the output

to display the interesting lines do:

```
rccron status  2>&1|grep systemctl

# systemctl --full --no-legend --no-pager --type=service
--property=LoadState show cron.service
```

• 2.5 look up the options in the systemctl man page, some of the options above allow for automating sytemctl commands and assigning them to variables.

• 2.5.1 Init scripts use DefaultDependencies=No

What is the main difference this is causing?

Hint: Read systemd.service(5) and search for DefaultDependencies

• 2.6 what other types are available for systemctl command?

Hint: use bash completion

A:

```
systemctl -t <tab>
```

• 2.7 Display the contents of the cron service:

• 2.8 what does it depend on? (Look for Wants and Requires lines) Is there any ordering imposed on those dependencies? (After and Before lines - if there is an After line, the current unit should be started after the indicated unit.)

• 2.9 what are the units that depend on it (Look for WantedBy and RequiredBy)

• 2.10 look, that the cron service is running

```
ps aux|grep cron
killall -s SIGKILL cron
ps aux|grep cron
killall cron
ps aux|grep cron
```

why the difference?

Note your findings: