HOW TO ACCESS SINGLE-USER MODE WITHOUT PASSWORD

I was asked to reset root password on some long forgotten Debian box. It was an easy and straightforward task, but, as there are some interesting pitfalls, I will describe the whole process of acquiring root shell without password using single-user mode and a couple of ways to prevent it.

What is single-user mode?

To access root shell without password you need to have physical access to the machine. Then you can modify kernel parameters to boot system into single-user mode which is just a single super user maintenance/recovery mode with all services disabled.

How to access single-user mode?

Default Debian configuration will require password before executing single-user mode and this is a standard behavior found in today's Linux distributions. To boot into this mode you need to turn on computer, access GRUB menu and select *Recovery mode* entry.
In case the *Recovery mode* menu entry is not available, you need to perform five simple steps in order to modify kernel parameters list.

1. Turn on computer.

2. Access GRUB menu.

3. Edit existing menu entry (use `e` key).

4. Add single keyword (alternatively you can use `-s` or `S`) to the Linux kernel parameters list.

5. Press `CTRL-X` or `F10` while still in edit mode to continue boot process.
How single-user mode is protected?

It is protected by using `sulogin` utility which is invoked by init process when system goes into single-user mode. You can verify this behavior manually by opening `/etc/inittab` file and looking for single-user runlevel definition.

```
$ cat /etc/inittab

[...]

# What to do in single-user mode.

~:S:wait:/sbin/sulogin

[...]
```

You can change it to shell interpreter if you do not want to enter password.

```
[...]
```
# What to do in single-user mode.

~~:S:wait:/bin/sh

[...]

## How to overcome the above protection?

You can modify default behavior and specify your own command run as init process as long as you can define kernel parameters.

So, according to the above statement you can get around this protection mechanism and boot into single-user mode to access root shell without password by specifying init option in the kernel parameters list.

```
setparams 'Debian GNU/Linux, with Linux 3.2.0-4-amd64'
load_video
insmod gzio
insmod part_msdos
insmod ext2
set root=(hd0,msdos1)
search --no-floppy --fs-uuid --set=root a42a1275-66d3-4b2a-8f1d-6268\e20b2b93
echo 'Loading Linux 3.2.0-4-amd64 ...'
linux /boot/vmlinuz-3.2.0-4-amd64 root=UUID=a42a1275-66d3-4b2a-8f1d-\e2686238b2b8 ro quiet init=/bin/sh
echo 'Loading initial ramdisk ...'
initrd /boot/initrd.img-3.2.0-4-amd64
```

How to protect against such attacks?
**BIOS**

Disable boot from external devices and lock boot device to the used one. Password protect BIOS settings. It is a weak protection but an important one, as circumventing it will surely draw an attention.

**GRUB**

Disable generation of recovery mode menu entries and lock down boot-loader to require authentication before accessing command line.

**Encryption**

This way is suitable only for personal devices but complements the above-mentioned methods with very strong protection. Full disk encryption will surely prevent access to the configuration files.

**Emergency mode**

Use this mode in case of emergency when you need to enter directly single-user mode without executing any other commands or startup scripts.

To start this mode use `-b` or `emergency` kernel option in the same way as the above ones.
Please note that emergency shell configuration is hard-coded and will use `sulogin` utility. Download `sysvinit` package source code if you want to modify it.