
w1r3 Network Documentation

Release 1.0

w1r3 Network

08.06.2014

| | | |
|----------|--|----------|
| 1 | Infrastructure | 3 |
| 1.1 | Team orientation | 3 |
| 1.2 | Accounts | 3 |
| 1.3 | Scripts | 3 |
| 1.4 | Kernel | 3 |
| 2 | Services | 5 |
| 2.1 | nya.ch - linux-kvm vps hosting | 5 |
| 2.2 | Arch Linux Repository | 10 |

Contents:

Infrastructure

Contents:

1.1 Team orientation

u know?

1.2 Accounts

Each user has an account in our ldap structure. It is possible to use the account credentials to authenticate on various services.

1.3 Scripts

We have a repository for scripts used in our infrastructure. The scripts are mainly helpers for users or for other software components.

This SOP is a guideline for writing new and maintaining existing scripts. It should help with the decision if it's a good idea to write a script for a specific function, how to document it and how to integrate it in our infrastructure.

1.3.1 When a script is a good idea

1.3.2 Why you shouldn't write one

1.3.3 Documentation

1.3.4 Integration

1.4 Kernel

Our Arch Linux powered servers are running a modified Linux kernel. The kernel is built with grsecurity patches and a working, semi-minimalist kernel configuration.

The linux-w1r3-xen kernel is tested with virtual private servers from Linode. The linux-w1r3-kvm kernel is tested with virtual private servers from nya.ch. We do not recommend to use this kernels in other infrastructures, because they are specifically designed and tested for the named environments.

Contents:

2.1 nya.ch - linux-kvm vps hosting

Contents:

2.1.1 virssh

`Virssh` ermöglicht dir die Kontrolle deiner VMs. Desweiteren gibt es dir die Möglichkeit, auf die Konsole der VM zuzugreifen.

Verbinden

Um `virssh` verwenden zu können benötigst du einen hinterlegten SSH-Key. Falls du noch keinen SSH-Key bei uns hinterlegt hast, wende dich an den Support.

```
ssh -t vmadmin@rachel.munich.wlr3.net
```

Dies zeigt dir die möglichen Befehle an.

Beispiel:

```
ALERT: NOT allowing command ""  
Allowed commands for you are:
```

```
ssh -t vmadmin@rachel.munich.wlr3.net sudo virsh list  
ssh -t vmadmin@rachel.munich.wlr3.net sudo virsh console example.nya.ch  
ssh -t vmadmin@rachel.munich.wlr3.net sudo virsh destroy example.nya.ch  
ssh -t vmadmin@rachel.munich.wlr3.net sudo virsh shutdown example.nya.ch  
ssh -t vmadmin@rachel.munich.wlr3.net sudo virsh start example.nya.ch  
ssh -t vmadmin@rachel.munich.wlr3.net sudo virsh start example.nya.ch --console
```

2.1.2 Rescue

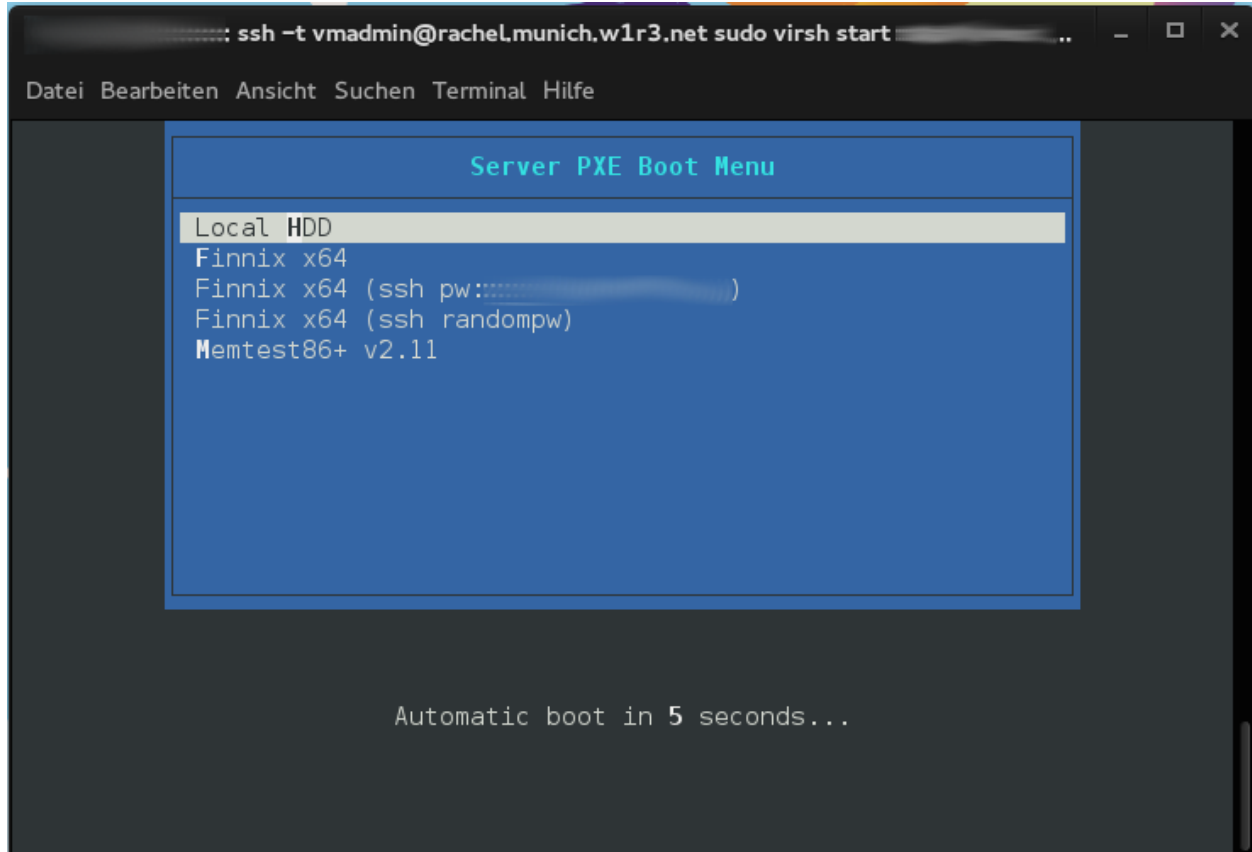
Da jeder einmal Fehler macht, bieten wir euch `Finnix` an, um diese zu korrigieren. `Finnix` kann auch zur Installation von `Arch Linux` verwendet werden.

Booten des Rescue-Systems

Zum Booten des Rescuesystems müssen wir als erstes die VM ausschalten, wonach wir sie mit folgendem Befehl wieder starten:

```
ssh -t vmadmin@rachel.munich.w1r3.net sudo virsh start example.nya.ch --console
```

Dies hat zur Folge, dass ihr auch gleich in der Konsole landet, um den Booteintrag auszuwählen.



Dort wählen wir dann einen der 3 Finnix Booteinträge aus (ich nehme hier jetzt den dritten)

```
ssh -t vadmin@rachel.munich.w1r3.net sudo virsh start
Datei Bearbeiten Ansicht Suchen Terminal Hilfe

Welcome to Finnix!

[*] Total memory: 2011MiB, shared ramdisk: 1597MiB
[*] System: QEMU Virtual version 0.13 1196MHz (4)
[*] Running Linux kernel 3.10.0-1-amd64-finnix on x86_64
[*] Generating SSH1 RSA host key... done
[*] Generating SSH2 RSA host key... done
[*] Generating SSH2 DSA host key... done
[*] Generating SSH2 ECDSA host key... done
[*] Starting OpenBSD Secure Shell server... sshd done
[*] New root/finnix password:
[*] Current system IP addresses:
    inet addr:31.47.227. Bcast:31.47.227.191 Mask:255.255.255.224
[*] Finnix 109 ready; 422 packages, 120M compressed

root@ttyS0:~#
```

2.1.3 Installation von Arch Linux

Wir benötigen unser Rescue-System, um Arch Linux zu installieren.

Installationsumgebung vorbereiten

Führe die folgenden Kommandos aus:

```
wget http://mirror.de.leaseweb.net/archlinux/iso/latest/arch/x86_64/root-image.fs.sfs
unsquashfs root-image.fs.sfs
cd squashfs-root
mount -o loop root-image.fs /mnt
/mnt/bin/arch-chroot /mnt
mkdir /run/shm
```

Einen Mirror auswählen

Wir bieten einen Mirror an, der alles schneller macht

```
echo "Server = http://mirror.w1r3.org/\$repo/os/\$arch" > /etc/pacman.d/mirrorlist
```

Partitionierung

Wir werden das ganze blockorientierte Gerät verwenden.

```
fdisk /dev/vda
```

Drücke n, um eine neue Partition zu erstellen, gefolgt von viermal Enter und w.

Formatierung

```
mkfs.ext4 /dev/vda1
```

Das neue Dateisystem mounten

```
mount /dev/vda1 /mnt
```

Basis-Pakete installieren

Wir werden die Paketgruppen base und base-devel installieren, aber zuerst müssen wir unseren Schlüsselring generieren. Für diese Aufgabe brauchen wir haveged, weil Entropie fehlt. Das Zweite ist für das Bauen der Pakete.

```
sed -i "s/SigLevel = Required DatabaseOptional/SigLevel = Never/" /etc/pacman.conf
pacman -Sy haveged procps-ng --noconfirm
haveged -w 1024
pkill haveged
pacman -Rs haveged procps-ng --noconfirm
pacman-key --populate archlinux
pacman-key --refresh-keys
sed -i "s/SigLevel = Never/SigLevel = Required DatabaseOptional/" /etc/pacman.conf
pacstrap /mnt base base-devel
```

/etc/fstab generieren

```
genfstab -U -p /mnt > /mnt/etc/fstab
```

Chroot in das neue System

```
arch-chroot /mnt
```

Root-Passwort setzen

```
passwd
```

Installiere einige Pakete, die immer gut zu gebrauchen sind

Diese Pakete sind, außer GRUB, welcher als Bootloader genutzt wird, meist optional.

```
pacman -S vim zsh git grub openssh rsync ntp ferm atop htop mtr traceroute wget net-tools dnsutils
```

GRUB installieren und konfigurieren

Zuerst müssen wir die `/etc/default/grub` bearbeiten, um seriellen Konsolen-Support zu aktivieren. Ersetze `GRUB_CMDLINE_LINUX_DEFAULT` mit dem Folgenden:

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,38400n8"
```

Füge noch diese Zeilen hinzu:

```
GRUB_TERMINAL=serial
GRUB_SERIAL_COMMAND="serial --speed=38400 --unit=0 --word=8 --parity=no --stop=1"
```

Führe nun die Kommandos unten durch, um GRUB auf das blockorientierte Gerät zu installieren und generiere die Konfigurationsdatei.

```
grub-install --recheck /dev/vda
grub-mkconfig -o /boot/grub/grub.cfg
```

Netzwerk konfigurieren

Erstelle jetzt ein `netctl`-Profil und konfiguriere die Netzwerk-Einstellungen. Ändere die IP-Einstellungen zu den IPs, die dir gegeben wurden.

```
nano /etc/netctl/ethernet-static

Description='nya.ch connection'
Interface=ens3v1
Connection=ethernet

IP=static
Address='31.47.227.CHANGEME/CHANGE'
Gateway='31.47.227.CHANGEME'
DNS=('8.8.4.4')
DNSDomain='nya.ch'

IP6=static
Address6=('2a02:2260:2:1f60::CHANGEME/64')
Gateway6='2a02:2260:2:1f60::1'
```

Aktiviere das `netctl`-Profil

```
netctl enable ethernet-static
```

Neu starten

Jetzt ist es an der Zeit, in dein neues System zu starten.

```
exit
umount /mnt
exit
reboot
```

Zeitzone und Hostname

Setze die Zeitzone und den Hostnamen. Bitte benutze den Hostnamen, der dir gegeben wurde.

```
timedatectl set-timezone Europe/Berlin
hostnamectl set-hostname HOSTNAME
```

Locales generieren und setzen

Entferne nun das Kommentarzeichen in der Datei `/etc/locale.gen` für die Locales DE, AT, CH, und US (alle UTF-8) und führe das Kommando `locale-gen` aus.

Setze jetzt die Standard System-Locales und die Tastaturbelegung

```
localectl set-locale LANG="en_US.UTF-8"
localectl set-keymap de
```

Neu starten und Spaß haben

Es gibt nichts weiter zu sagen ;)

2.2 Arch Linux Repository

Diese Seite ist leer und wird aktualisiert, sobald es wieder eine produktive Repository gibt.