NHR@GÖTTINGEN

NHR Security / Admin Workshop

Dr. Tim Ehlers
tehlers@gwdg.de

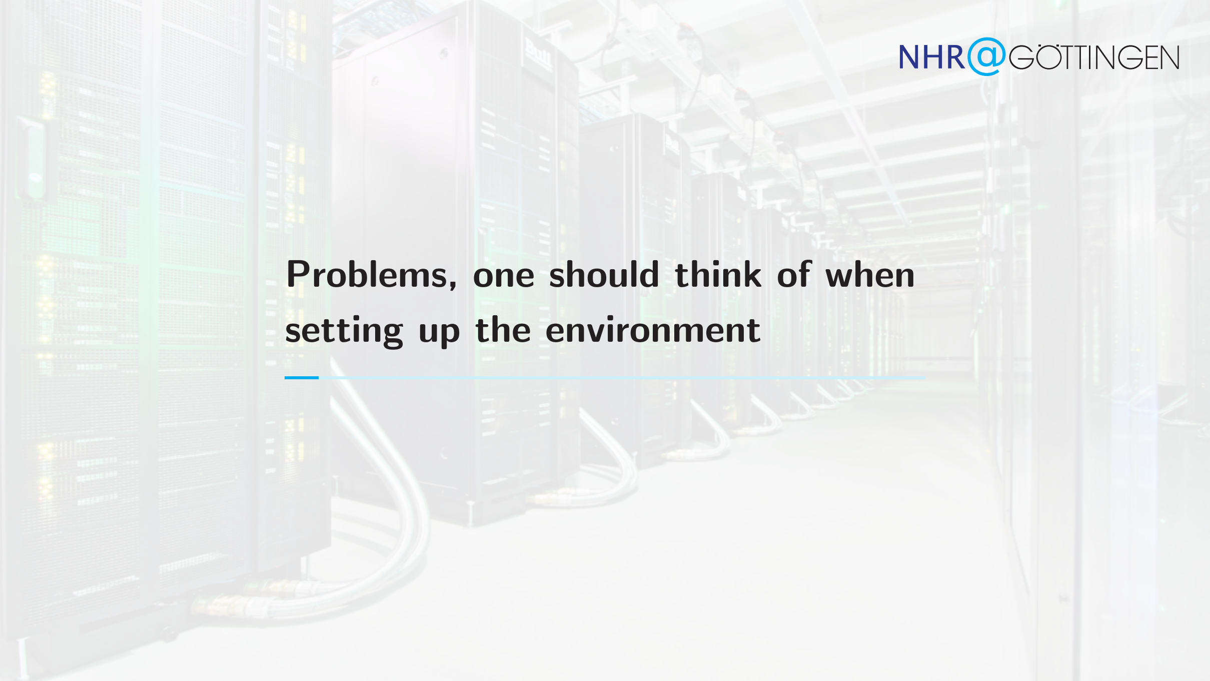
December 16, 2021

hpc-support@gwdg.de
GWDG – Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

Security in HLRN – Concept and Experiences

Outline

- Problems, one should think of when setting up the environment
- HLRN IV Security Concept (dev. by Hinnerk Stüben, Uni Hamburg)
- DFN-Cert recommendations



Problems, one should think of when setting up the environment

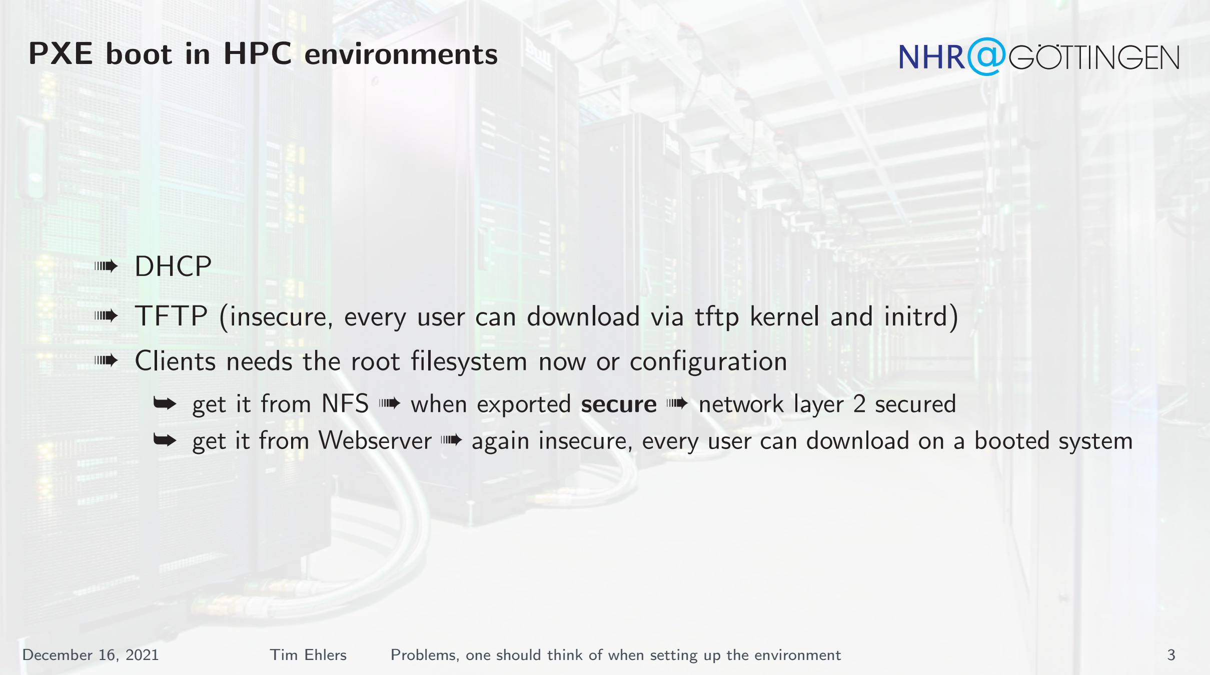
global filesystems

Mounting defaults options

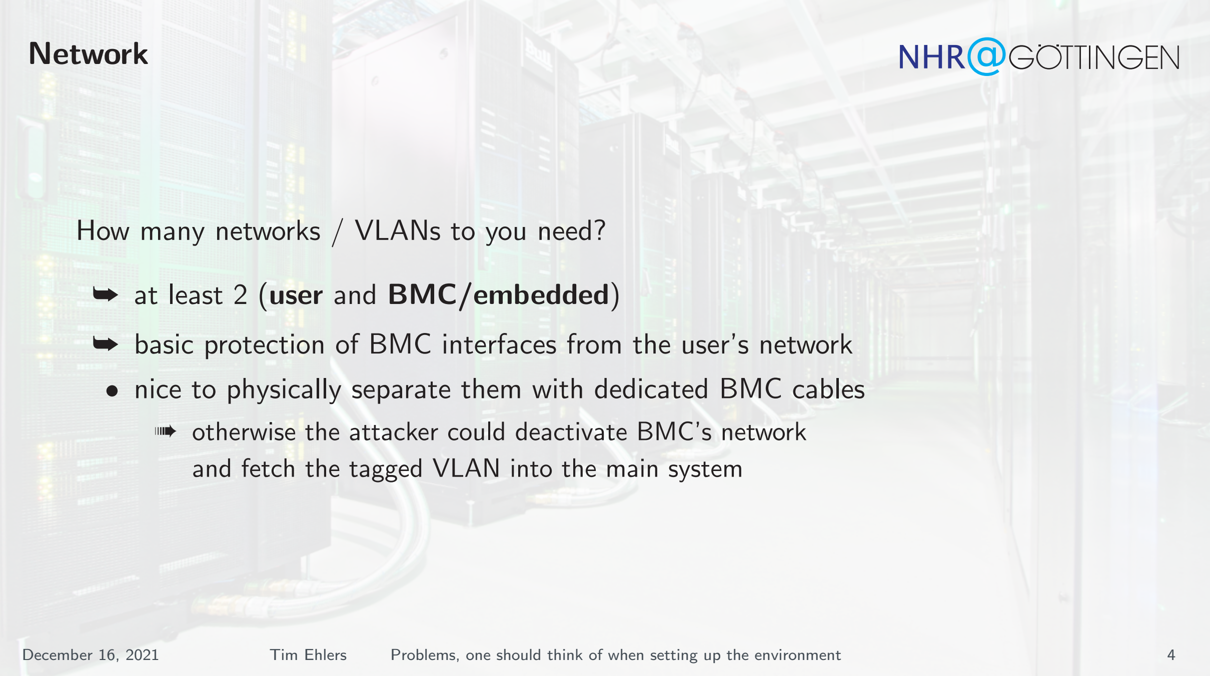
```
~ > grep nfs /etc/fstab
10.10.10.10:/sw /opt/sw nfs defaults 0 0
~ > mount | grep /opt/sw
10.10.10.10:/sw on /opt/sw type nfs (rw,relatime,vers=3,rsize=1048576,wsiz

~ > mount | grep lustre
10.10.10.10@o2ib:/work on /scratch type lustre (rw,flock,lazystatfs)
```

nosuid, nodev

PXE boot in HPC environments



- ➠ DHCP
- ➠ TFTP (insecure, every user can download via tftp kernel and initrd)
- ➠ Clients needs the root filesystem now or configuration
  - ➡ get it from NFS ➠ when exported **secure** ➠ network layer 2 secured
  - ➡ get it from Webserver ➠ again insecure, every user can download on a booted system



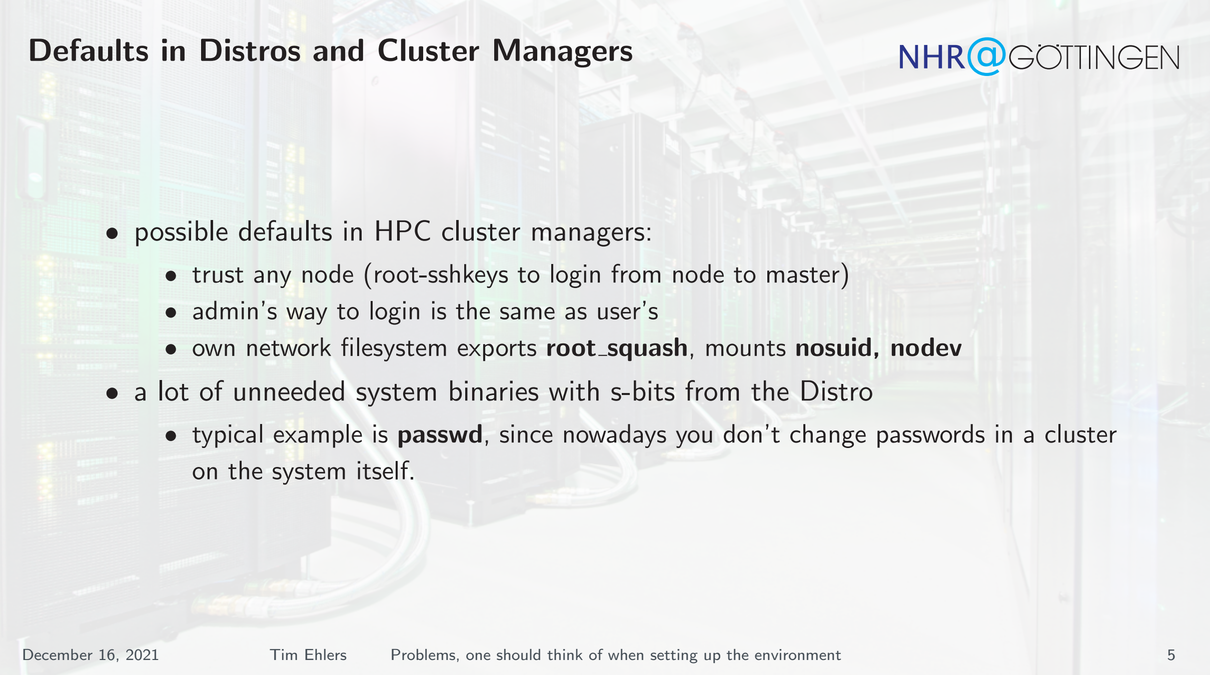
Network

NHR@GÖTTINGEN

How many networks / VLANs to you need?

- ➥ at least 2 (**user** and **BMC/embedded**)
- ➥ basic protection of BMC interfaces from the user's network
  - nice to physically separate them with dedicated BMC cables
    - ➠ otherwise the attacker could deactivate BMC's network and fetch the tagged VLAN into the main system

Defaults in Distros and Cluster Managers



- possible defaults in HPC cluster managers:
  - trust any node (root-sshkeys to login from node to master)
  - admin's way to login is the same as user's
  - own network filesystem exports **root_squash**, mounts **nosuid, nodev**
- a lot of unneeded system binaries with s-bits from the Distro
  - typical example is **passwd**, since nowadays you don't change passwords in a cluster on the system itself.

HLRN IV Security Concept (dev. by Hinnerk Stüben, Uni Hamburg)

HLRN requirements for the vendor

- only needed suid -bits on binaries
- only for operations needed services enabled / reachable
  - for example: mysqld + slurm
  - ➥ only slurmdbd needs to reach mylsqld
- all filesystems must be exported **root_squash**
  - especially NFS should not be exported with **insecure**
  - ➥ default in nfs-ganesha
- only needed filesystems should be mounted
  - all global filesystems must be mounted with **nosuid** and **nodev**

HLRN requirements for the vendor II

- sshd allows only from configured IPs to login (user / root)
- all sshd configs allow login only from a higher / same “class / layer”
- the admin always login a normal user with ssh-key over the jumphost
  - **sudo** with the password of the normal userID

  ➥ only way to become root in backend systems
- all services that can run as user, should **not** run as root

layer model in HLRN

- No login from lower layer to above
- lowest layer (S3): users login / interact
- S2: Services for users
- S1: Services for operation
- S0: admin login + background services

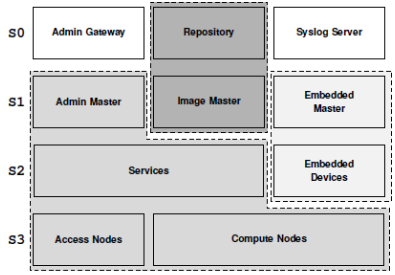
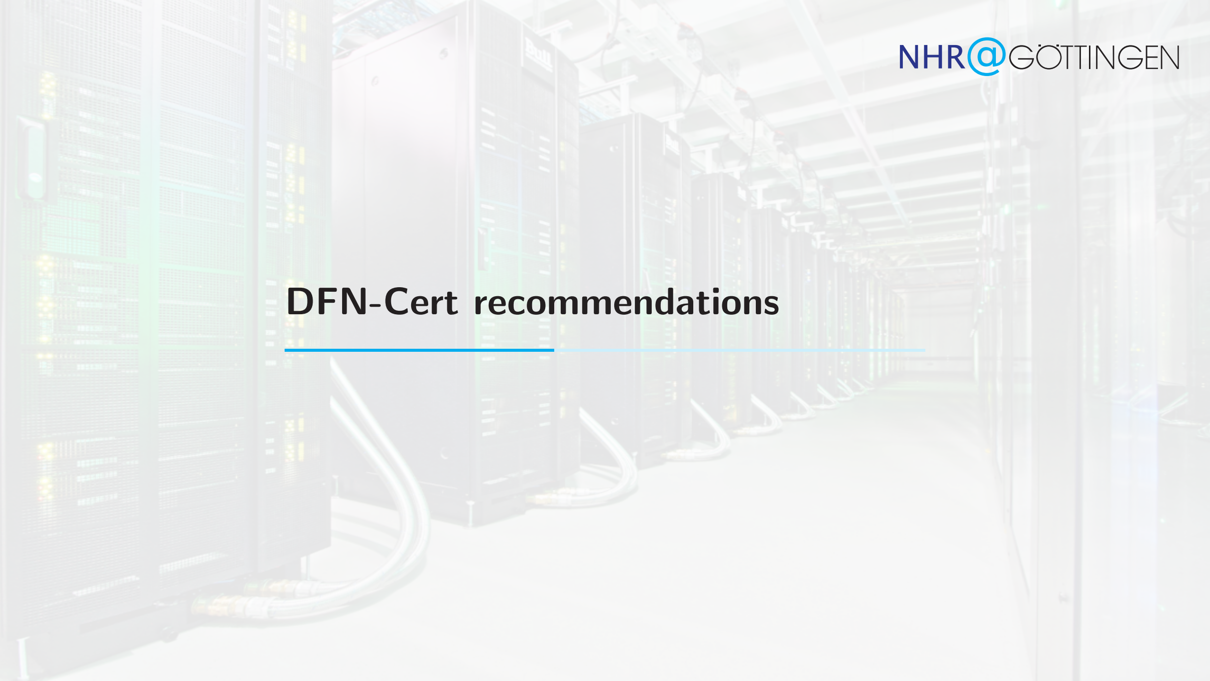
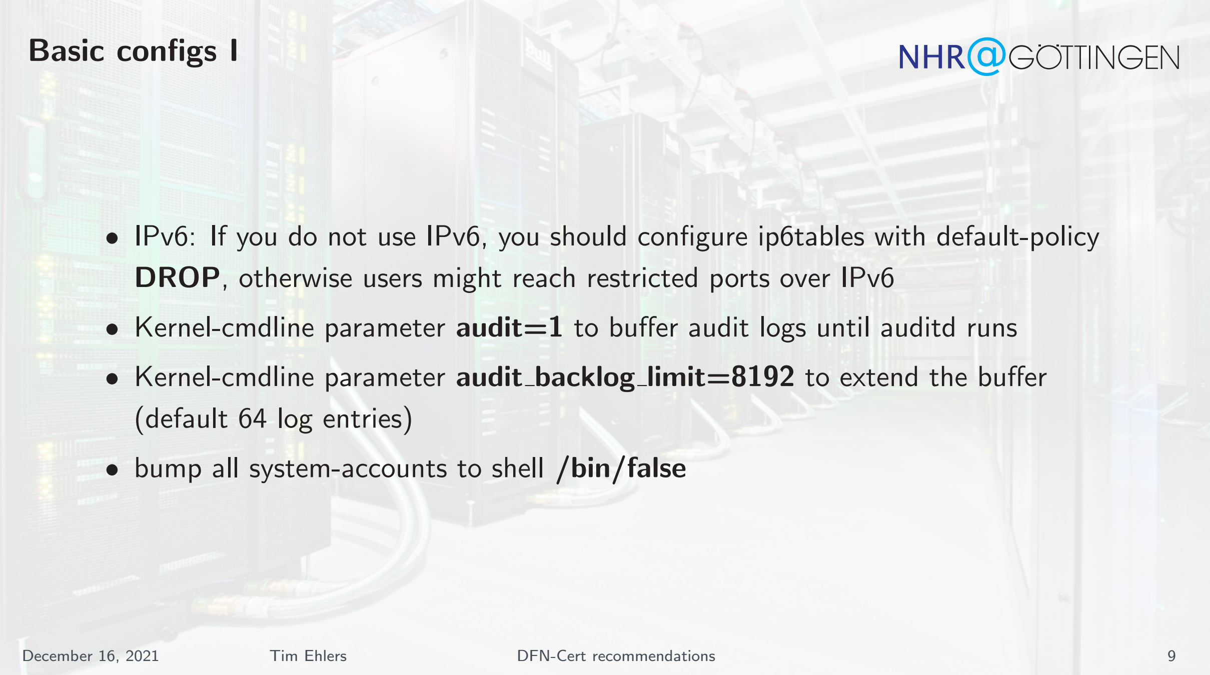


Abbildung LB.3: Sicherheitsschichten in der Systemkonfiguration des HLRN-IV.



DFN-Cert recommendations

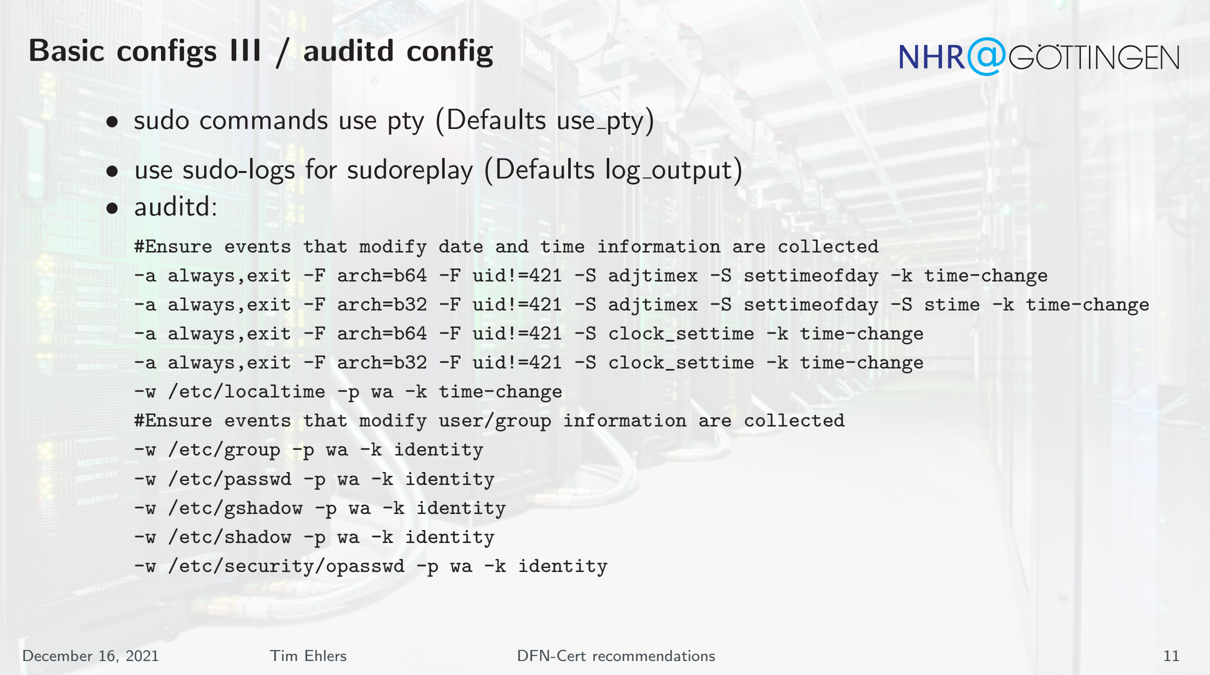
Basic configs I



- IPv6: If you do not use IPv6, you should configure ip6tables with default-policy **DROP**, otherwise users might reach restricted ports over IPv6
- Kernel-cmdline parameter **audit=1** to buffer audit logs until auditd runs
- Kernel-cmdline parameter **audit_backlog_limit=8192** to extend the buffer (default 64 log entries)
- bump all system-accounts to shell **/bin/false**

Basic configs II

- sshd (Encryption algorithms / Hashing algorithms)
  - `https://www.ssh-audit.com/hardening_guides.html`
  - `https://github.com/jtesta/ssh-audit`
  - **hmac-sha2-512** might be needed for older libssh
- SSL/TLS: check weak signature algorithms in certificates
  - for example MD2, MD4, MD5 or SHA1
- SSL/TLS: Vulnerable Cipher Suites for HTTPS
  - RC4 (CVE-2013-2566, CVE-2015-2808)
  - key length 64 bit (CVE-2015-4000)
  - RSA 1024 bit
- disable SSLv2 and SSLv3 in HTTPS



Basic configs III / auditd config

NHR@GÖTTINGEN

- sudo commands use pty (Defaults use_pty)
- use sudo-logs for sudoreplay (Defaults log_output)
- auditd:

```
#Ensure events that modify date and time information are collected
-a always,exit -F arch=b64 -F uid!=421 -S adjtimex -S settimeofday -k time-change
-a always,exit -F arch=b32 -F uid!=421 -S adjtimex -S settimeofday -S stime -k time-change
-a always,exit -F arch=b64 -F uid!=421 -S clock_settime -k time-change
-a always,exit -F arch=b32 -F uid!=421 -S clock_settime -k time-change
-w /etc/localtime -p wa -k time-change
#Ensure events that modify user/group information are collected
-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity
```

auditd config II

```
#Ensure events that modify the system's network environment are collected
-a always,exit -F arch=b64 -S sethostname -S setdomainname -k system-locale
-a always,exit -F arch=b32 -S sethostname -S setdomainname -k system-locale
-w /etc/issue -p wa -k system-locale
-w /etc/issue.net -p wa -k system-locale
-w /etc/hosts -p wa -k system-locale
-w /etc/sysconfig/network -p wa -k system-locale
#Ensure events that modify the system's Mandatory Access Controls are collected
-w /etc/selinux/ -p wa -k MAC-policy
-w /usr/share/selinux/ -p wa -k MAC-policy
#Ensure login and logout events are collected
-w /var/log/faillog -p wa -k logins
-w /var/log/lastlog -p wa -k logins
# optional für pam_faillock.so
-w /var/run/faillock/ -p wa -k logins
# optional für pam_tally2.so
-w /var/log/tallylog -p wa -k logins
```

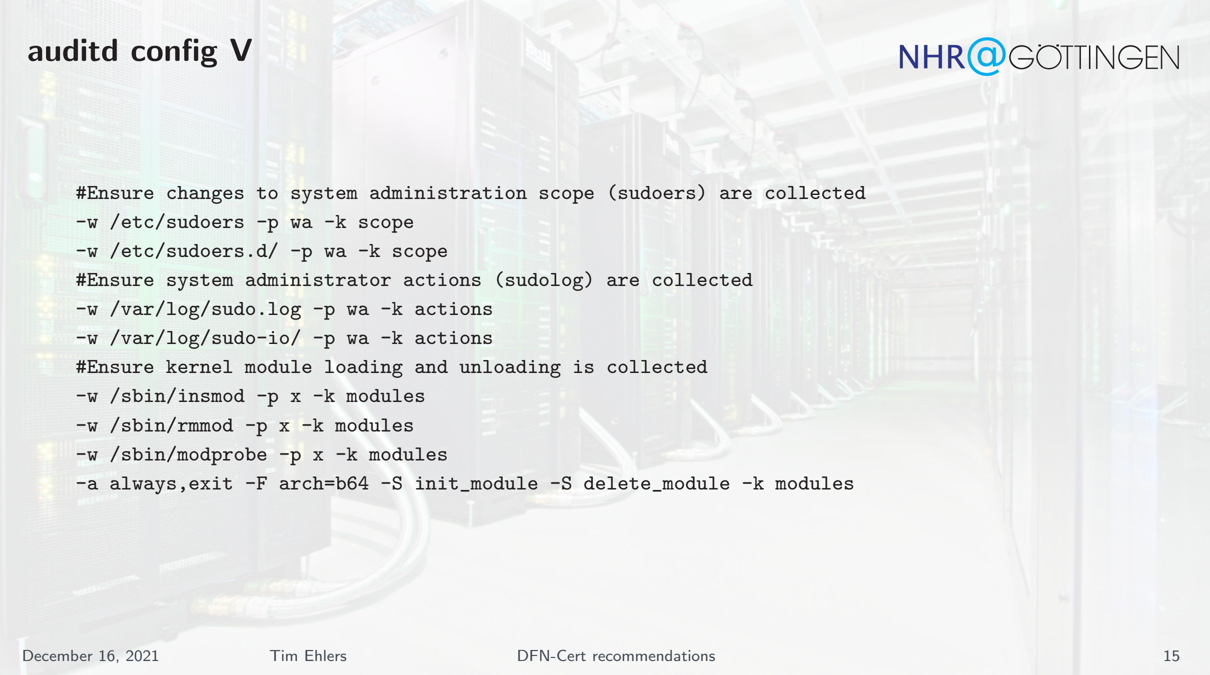
auditd config III

```
#Ensure session initiation information is collected
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k logins
-w /var/log/btmp -p wa -k logins
#Ensure discretionary access control permission modification events are collected
-a always,exit -F arch=b64 -S chmod -S fchmod -S fchmodat -F auid>=1000 -F auid!=4294967295\
-k perm_mod
-a always,exit -F arch=b32 -S chmod -S fchmod -S fchmodat -F auid>=1000 -F auid!=4294967295\
-k perm_mod
-a always,exit -F arch=b64 -S chown -S fchown -S fchownat -S lchown -F auid>=1000 -F\
auid!=4294967295 -k perm_mod
-a always,exit -F arch=b32 -S chown -S fchown -S fchownat -S lchown -F auid>=1000 -F\
auid!=4294967295 -k perm_mod
-a always,exit -F arch=b64 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr\
-S fremovexattr -F auid>=1000 -F auid!=4294967295 -k perm_mod
-a always,exit -F arch=b32 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr\
-S fremovexattr -F auid>=1000 -F auid!=4294967295 -k perm_mod
```

auditd config IV

```
#Ensure unsuccessful unauthorized file access attempts are collected
-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EACCES\
-F auid>=1000 -F auid!=4294967295 -k access
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EACCES\
-F auid>=1000 -F auid!=4294967295 -k access
-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EPERM\
-F auid>=1000 -F auid!=4294967295 -k access
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EPERM\
-F auid>=1000 -F auid!=4294967295 -k access
#Ensure successful filesystem mounts are collected
-a always,exit -F arch=b64 -S mount -F auid>=1000 -F auid!=4294967295 -k mounts
-a always,exit -F arch=b32 -S mount -F auid>=1000 -F auid!=4294967295 -k mounts
```

auditd config V



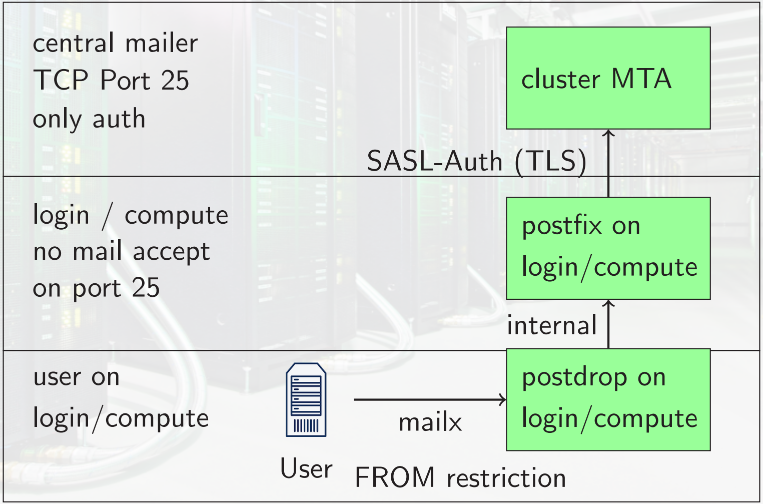
```
#Ensure changes to system administration scope (sudoers) are collected
-w /etc/sudoers -p wa -k scope
-w /etc/sudoers.d/ -p wa -k scope
#Ensure system administrator actions (sudolog) are collected
-w /var/log/sudo.log -p wa -k actions
-w /var/log/sudo-io/ -p wa -k actions
#Ensure kernel module loading and unloading is collected
-w /sbin/insmod -p x -k modules
-w /sbin/rmmod -p x -k modules
-w /sbin/modprobe -p x -k modules
-a always,exit -F arch=b64 -S init_module -S delete_module -k modules
```

auditd config VI

```
#Ensure use of privileged commands is collected
-a exit,always -F arch=b64 -F euid=0 -F uid!=0 -S execve -k audit-nhr
-a exit,always -F arch=b32 -F euid=0 -F uid!=0 -S execve -k audit-nhr
-a exit,always -F arch=b64 -F egid=0 -F gid!=0 -S execve -k audit-nhr
-a exit,always -F arch=b32 -F egid=0 -F gid!=0 -S execve -k audit-nhr
#OR
-a always,exit -F arch=b32 -S execve -C uid!=euid -F euid=0 -k setuid
-a always,exit -F arch=b64 -S execve -C uid!=euid -F euid=0 -k setuid
-a always,exit -F arch=b32 -S execve -C gid!=egid -F egid=0 -k setgid
-a always,exit -F arch=b64 -S execve -C gid!=egid -F egid=0 -k setgid

# make settings immutable
-e 2
```

Mailing I

Do you allow users to send mails? Do you have scripts sending mails?



central mailer
TCP Port 25
only auth

cluster MTA

SASL-Auth (TLS)

login / compute
no mail accept
on port 25

postfix on
login/compute

internal

user on
login/compute

User

mailx

postdrop on
login/compute

FROM restriction

Mailing II

- all nodes should have a shared secret for sending mails to central mailserver:

  ```
  smtp_sasl_auth_enable = yes
  smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
  ```

- central mailserver should not accept mails without sasl user/pass

  ```
  smtpd_sasl_auth_enable = yes
  smtpd_sasl_security_options = noanonymous
  ```

- all nodes should not accept mails on localhost:25 (mynetworks =)

Mailing III

NHR@GÖTTINGEN

- since postfix 3.6 you can prevent sending mails from false senders over mailx:

```
 local_login_sender_maps =
    inline:{ { root = *}, { postfix = * } },
    pcre:/etc/postfix/login_senders
~ > cat /etc/postfix/login_senders
# Allow both the bare username and the user@domain forms.
/(.+)/ $1 $1@hlrn.de $1@cm.cluster $1@glogin1.cm.cluster
```

- If you have an older postfix, you can just compile a new one and exchange only **/sbin/postdrop**