

CIS Hardening for isar- cip-core

About Me



Quirin Gylstorff

Software Engineer @Siemens Linux Expert Center



@gylstorffq



quirin.gylstorff@siemens.com

Why CIS hardening

- Products need a documented security baseline (CRA, IEC 62443-4-2).
- Typical practice: hand-written post-install scripts per product
- No proof the kernel / rootfs actually match the policy

Goals

Tailored for product use cases

Hardening and validation are separate components

Overridable from downstream layers

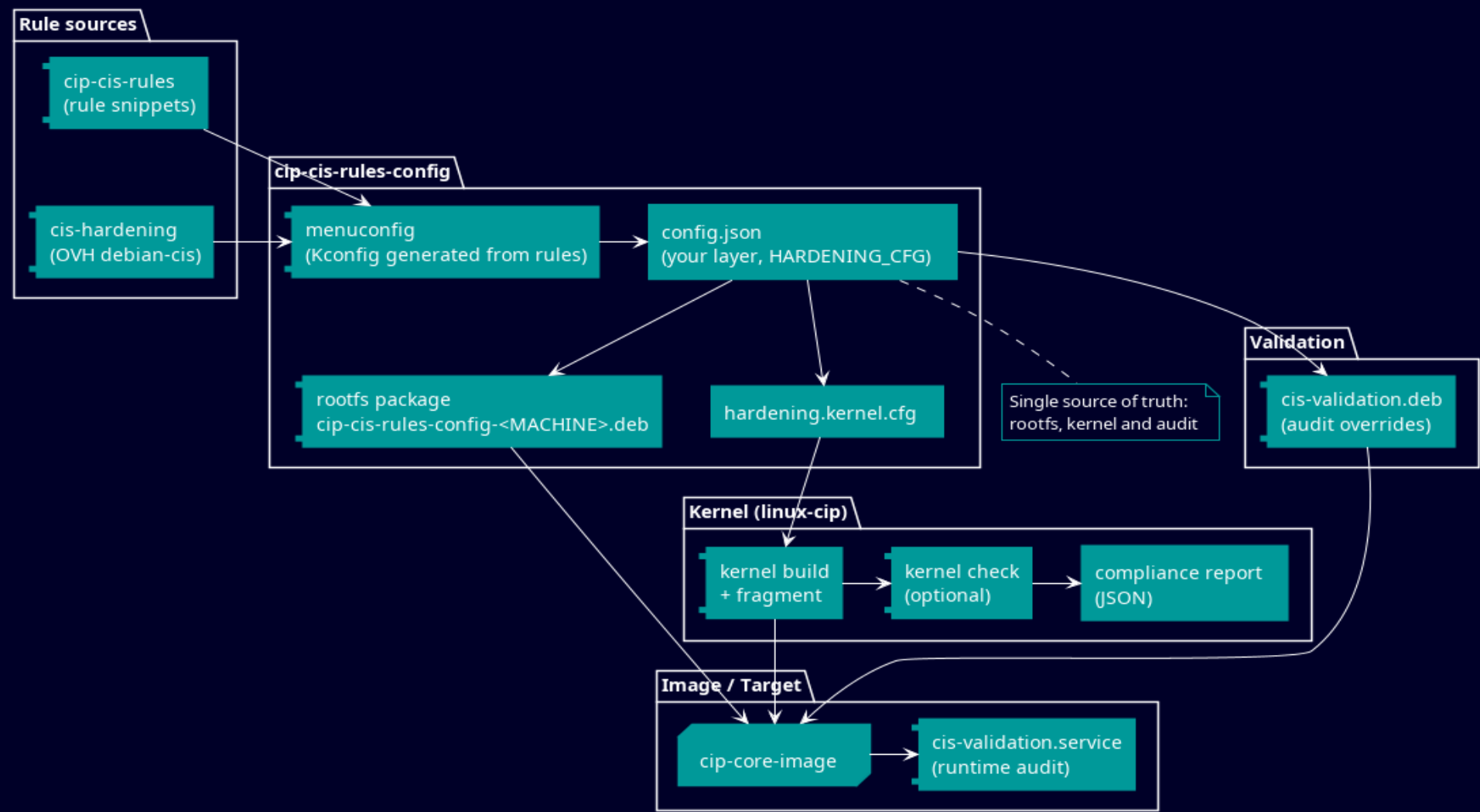
No default config shipped

Single source of truth

Package (ssh, firewall) installation is product specific

Recipe Overview

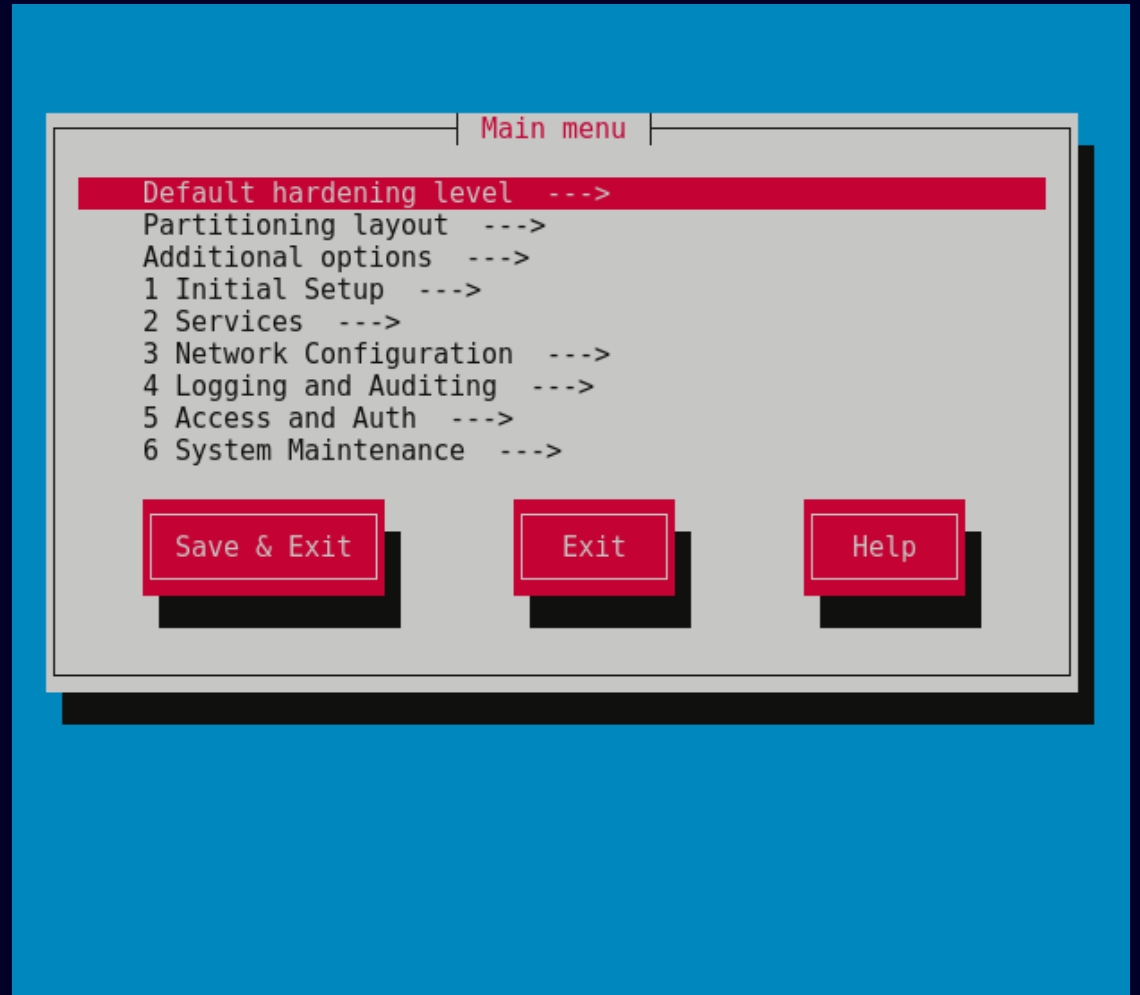
CIS Hardening in isar-cip-core — Build-time Data Flow



Hardening config via Kconfig

- Kconfig is **generated** from the OVH debian-cis ruleset at menuconfig time
- Menu contains:
 - hardening level choice (1–5)
 - one bool per CIS rule
 - justification string for each required but disabled rule
- Changing level resets rule defaults, keeps justifications

```
bitbake cip-cis-rules-config -c menuconfig
```



Config json

Output of menuconfig

Per rule: `apply` and `justification`

Check into your layer, reference with
`HARDENING\_CFG`

Deployed as `\${DISTRO}-
hardening.config.json`

Installed to `/usr/share/cip-cis-rules/`

```
...  
  {  
    "rule": {  
      "name": "1.1.1.5",  
      "description": "disable squashfs",  
      "level": 2,  
      "apply": false,  
      "justification": "isar cip core  
uses squashfs for read-only root file  
systems"  
    }  
  },  
  ...
```

Rootfs hardening

- `cip-cis-rules-config-${MACHINE}` installs **only** snippets of rules with `apply: true`:
- The rules are stored in <https://gitlab.com/cip-playground/cip-cis-rules> and contain
 - `/etc/modprobe.d/` (disabled filesystems, USB storage)
 - `/etc/sysctl.d/`
 - `/etc/ssh/sshd_config.d/`
 - `/etc/sudoers.d/`
 - `/etc/security/limits.d/`
 - helper scripts for install-time actions (permissions, `pam-auth-update`, `login.defs`)

Kernel Hardening

- Enabled kernel rules become a config fragment, added to the linux-cip build
 - A report is generated and stored in the deploy dir
- ```
CIS hardening kernel config
Generated by generator/gen-
kernel-config.py - do not
edit.
Values: y=built-in,
n=disabled

CIS 1.1.1.1: Disable
Veritas VxFS filesystem
CONFIG_VXFS_FS is not set

CIS 1.1.1.2: Disable JFFS2
filesystem
CONFIG_JFFS2_FS is not set
```

```
"rule": {
 "name": "1.1.1.1",
 "description": "Disable Veritas",
 "compliant": true,
 "symbols": [
 {
 "symbol": "CONFIG_VXFS_FS",
 "expected": "n",
 "actual": "not set",
 "compliant": true
 }
]
},
```

# CIS validation

`cis-validation.service`  
runs OVH  
`hardening.sh --  
audit` at boot

Audits the  
**selected** policy, not  
the full upstream  
set

```
cis-validation[355]: OK 6.2.19_check_duplicate_groupname OK{No duplicate groupnames}
cis-validation[355]: OK 6.2.20_shadow_group_empty OK{No user belongs to shadow group} OK{No user
r primary group}
cis-validation[355]: AUDIT_SUMMARY PASSED_CHECKS:93 RUN_CHECKS:97 TOTAL_CHECKS_AVAIL:223 CONFORMIT
cis-validation[355]: CIS validation conformity is 95.87%, which is less than 100%.
systemd[1]: cis-validation.service: Main process exited, code=exited, status=1/FAILURE
systemd[1]: cis-validation.service: Failed with result 'exit-code'.
systemd[1]: cis-validation.service: Consumed 9.743s CPU time, 93.8M memory peak.

cis-validation[444]: OK 6.2.18_check_duplicate_username OK{No duplicate usernames}
cis-validation[444]: OK 6.2.19_check_duplicate_groupname OK{No duplicate groupnames}
cis-validation[444]: OK 6.2.20_shadow_group_empty OK{No user belongs to shadow group} OK{No u
cis-validation[444]: AUDIT_SUMMARY PASSED_CHECKS:97 RUN_CHECKS:97 TOTAL_CHECKS_AVAIL:223 CONFO
cis-validation[444]: CIS validation conformity is 100%.
```

## Using in your layer

Your layer:

```
FILESEXTRAPATHS:prepend :=
"${FILE_DIRNAME}/files:"

HARDENING_CFG = "product.config.json"
```

Workflow:

```
bitbake cip-cis-rules-config -c menuconfig →
copy JSON into layer → commit → build.
```

Config (kas/distro.conf or machine config)

kas/opt/cis-hardening.yml (or "Harden image according to CIS rules" in kas menu)

```
local_conf_header:
 enable-hardening: |
 CIP_CIS_HARDENING = "1"
 enable-validation: |
 CIP_CIS_HARDENING_VALIDATION = "1"
```

## Next Steps

- Next:
  - Integration into isar-cip-core v1.14
  - Update Process for new versions
- Call to action
  - Try it, send feedback to cip-dev