cisco

tail-f

Secure ConfD Deployment

Per Andersson

ConfD Developer Days 2017

DISCLAIMER

This talk will discuss the vast security area, it is by no means complete or a reference work

Secure ConfD Deployment Talk

- Common ConfD Deployment
- Security trade offs
- Why is security important?
- Best Practices
- Development vs Deployment
- ConfD Secure Deployment Checklist

Common ConfD Deployment

Too Common Deployment

1. Establish the development environment
2. Develop
3. ???
4. Deploy

???

- Just deploying the ConfD development environment to production target
- The ConfD development environment is inherently insecure
- Not tightening or hardening the network, environment, OS, or application for production use
- Trading ease of use for security

What is security?

Sloppy definition:
Just say NO!

Security

- The degree of resistance to, or protection from, harm
- Applies to any vulnerable asset
- The most secure system is no system!
  - Nothing is solved by that

What is usability?

Sloppy definition:
Just say YES!

Usability

- Easy to use
- Easy to learn
- Usefulness: A usable system solves problems

Security
vs
Usability

Security vs Usability

- Eternal trade off
  - Increasing **security** generally decreases **usability**
  - Increasing **usability** generally decreases **security**

It’s trade offs all the way down

- It’s so secure you can’t come in
- It’s so secure you can’t even come out!
- What to protect
  - E.g. service uptime or data integrity?

Why is security important?

Ever been around when a breach or vulnerability affected your service or product?

(It is quite intense)

Why is Security Important?

- Risk of ending up with devices or entire network searchable on publicly available sites
- Denial of Service, leak of sensitive data, harm to assets or person…
- What’s the alternative? No security, no passwords, no cryptography, totally open systems...

Last week

Two vulnerabilities with massive impact

Last week

WPA2 rendered insecure

- Possible to eavesdrop and inject data

Last week

ROCA, CVE-2017-15361

- Affects RSA keys generated with vulnerable hardware
- Vulnerable chips manufactured since 2012
- Possible to calculate **private** key data from **public** key data
- Plausible for key length up to 2048 bits

It is quite intense

Best Practices

Best Practices

The entire organization must be security-minded

Best Practices

- Impossible to implement all security up front
- Security can’t just be bolt on after development is done
- Security is not a check box item, it is constant ongoing work

Best Practices

- Establish threat model
  - Identify threats
- Attack potential
- Sophisticated attacks
  - Covert or side channel attacks
  - Leveraging outside factors

Best Practices

- Study vulnerabilities and security techniques
  - Monitor CVE:s (Common Vulnerabilities and Exposures) and security mailing lists
  - OWASP (Open Web Application Security Project)
- Attack it yourself
  - Vulnerability scanner e.g. Nessus, OpenVAS, w3af, OWASP ZAP, Burp Suite...
  - Penetration testing e.g. with Metasploit
  - Integrate in Continuous Integration
- Peer reviews
- Security-oriented peer reviews

Best Practices

- Secure design
  - Input data is hostile, validate it
  - Fail gracefully with erroneous input
  - Assumptions are dangerous: “this input file will never grow beoynd 10 MB”
  - Safe defaults, fail securely don’t fail insecurely
  - Principle of least privilege

Best Practices

- Reduce number of moving parts, if possible
- Make every layer carry its own security
- Common security tip: Don’t roll your own crypto

Best Practices

- Development tool security
  - Audit compilers, libraries, third party components etc
  - Recent example: Malicious Python packages on PyPi, e.g. “`urllib4`”
  - Review source and verify signatures for third party components
- Sign deliverables
  - E.g. with OpenPGP or OpenSSL
  - Verify signed deliverables

Development vs Deployment

Development vs Deployment

- Install program installs ConfD suitable for development
- Changes are necessary to prepare ConfD for production deployment

Development vs Deployment

See Installing ConfD on a target system section in ConfD User Guide

Development vs Deployment

Evaluating and customizing `confd.conf` and other config files is fundamental

Development vs Deployment

Running ConfD in a development environment with some random merged configurations leaves a whole lot of wrongly set parameters, passwords, ports, default values etc.

Development vs Deployment

None of the well over 80 `confd.conf` files included in examples are to be used as-is on a target system without a careful evaluation of the contents

ConfD Secure Deployment Checklist

Authentication

- Evaluate and customize `confd.conf` and `aaa_init.xml`
  - Local authentication / PAM / external authentication
  - No credentials in `aaa_init.xml` unless local authentication is used
    - Will be reset upon database re-initialization e.g. factory reset
    - Needs to be generated
    - High risk of participating in bot nets otherwise

Authentication

- Update all <password> tags
- Use secure ~~passwords~~ pass phrases
  - Not the default values
  - Not the account name
  - Something with high entropy (not a dictionary word with a number and special character)

Authentication (NACM Rules)

- Evaluate and customize `aaa_init.xml`
- Consider which groups (if any) are allowed to **change** NACM rules

Northbound Interfaces

- CLI, NETCONF, WebUI, RESTCONF, REST, SNMP, JSON-RPC, API bindings etc.
- Disable unused interfaces in `confd.conf`
- Interfaces used internally in product only listens on 127.0.0.1
- No DSA host key for NETCONF/CLI over SSH, use RSA
  - Unique host key per device

Northbound Interfaces

- Disable NETCONF over TCP (has no authentication)
  - Use the IPC port for ‘netconf-subsys’ if needed (for external SSH server)
- Lots of privileged ports
  - NETCONF over SSH (830), SSH (22), SNMP (161)
  - Requires root privileges or `CAP_NET_BIND_SERVICE` Linux capability
  - ConfD doesn’t drop privileges
  - Let ConfD bind to unprivileged (local) port and port forward to this one, instead of running as root

CLI

Evaluate/customize/add/remove clispec files

(custom/extension commands)

CLI

- Remove clispec files if CLI interface isn't provided
- Only provided as examples, not production ready
  - `$CONFD_DIR/src/confd/cli/confd.cli` (source)
  - `$CONFD_DIR/etc/confd/confd.ccl` (compiled)
- '`ssh`' and '`telnet`' commands included in the example for versions prior to ConfD 6.4 are problematic
  - Can be used to e.g. port forward to internal ports
- Consider using `/confdConfig/cli/restrictedFileAccess`
  - CLI user can't access files outside home directory tree

confd_dyncfg.yang

- `confd.conf` parameters in CDB
- Must not be exposed as-is to northbound interfaces
- Compile the YANG module with '`--export none`' option to `confdc`
- Possibly expose selected parameters via '`tailf:link`' or '`tailf:transform`'

Encryption Keys in confd.conf

- `/confdConfig/encryptedStrings`
  - In `confd.conf` and `confd_dyncfg.yang`
- Replace the “dummy” values, e.g. `0123456789abcdef`
- Unique, random keys per device
  - For ConfD HA (High Availability), keys must be equal on HA pair nodes

Encryption Keys

- ConfD installation does nothing to change encryption keys
  - You must change these yourself
- Change encryption keys dance
  - Decrypt data with old key
  - Generate new key
  - Encrypt data with new key

IPC port

- No authentication or authorization
  - Used by applications that are part of the product
- Minimal protection to listen only on `127.0.0.1`
  - `/confdConfig/confdIpcAddress/ip`
- Consider restriction via shared secret protected by OS file permissions
  - `/confdConfig/confdIpcAccesCheck` random, unique secret per device

Quick Summary

Quick Summary

- Don’t just install the development environment to production target
- Evaluate all configuration
- Don’t expose things that are sensitive or unused
- Don’t expose unencrypted northbound interfaces
- Ensure security practices are in place within the entire organization

Thank you for listening!

www.tail-f.com

Opinions?

- Would a preflight check method or script be helpful?
  - Check config files for e.g. default values
- Vulnerability scanner plugins?
- What are your challenges regarding security?
  - Degrades usability?
  - Deployments in high security networks, without access?