Making Linting Easier

Rob Stradling
Sectigo

CA/Browser Forum F2F 62
Bergamo

My motivation

crt.sh

Linting newly logged certs is too slow

Updating linters is really awkward

Need to add support for pkilint

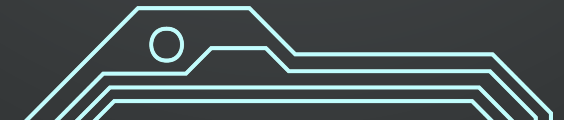
Sectigo's CA

Need to future-proof performance

Updating linters is a bit awkward

Ongoing code modernization

Conclusion: Linter integration revamp needed



Ecosystem observations

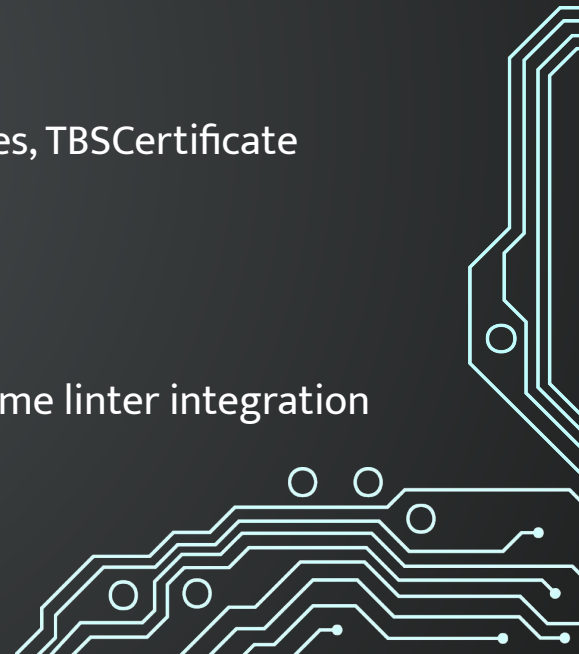
SC-75 proposes that *"the CA SHOULD implement a Linting process to test the technical conformity of the Certificate to be issued with these Requirements"*

Best practice: Use multiple linters

Integrating linters into a CA system is awkward: different languages, TBSCertificate input, etc

Performance of some linters is poor, when naively integrated

Many CAs are tackling, and some appear to struggling with, the same linter integration issues independently. Why not collaborate on solutions?



Summary of the pain points

Integration and Updating

Performance and Scalability

Basically, everything is painful!

pkimetal: A PKI meta-linter

Encasing something in metal makes it stronger!

https://www.collinsdictionary.com/dictionary/english/meta
“going beyond or higher, transcending”

Doesn’t replace any other linters

A one-stop, easy-to-use solution for all your linting needs

pkimetal: Features

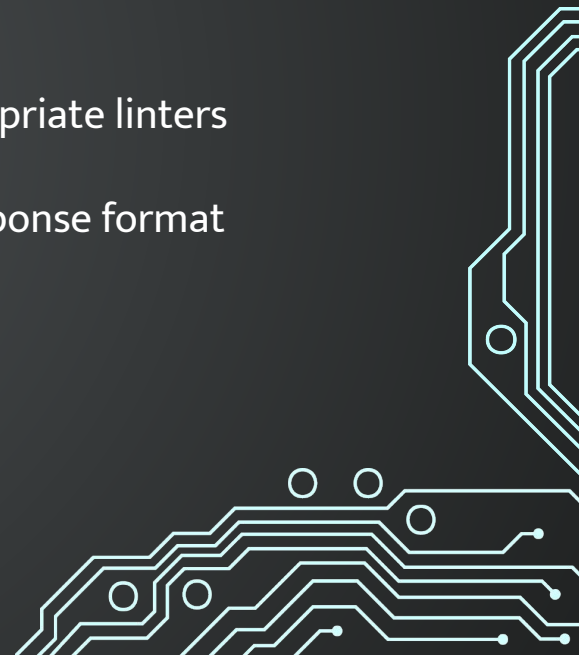
Provides access to multiple linters via a single, simple REST API call

Supports (pre-)certificates, CRLs, and OCSP responses as inputs

Auto-detects the intended profile of the input, and runs the appropriate linters

Combines the findings from all of the linters into one uniform response format

Supports pre-issuance and post-issuance linting



pkimetal: More features

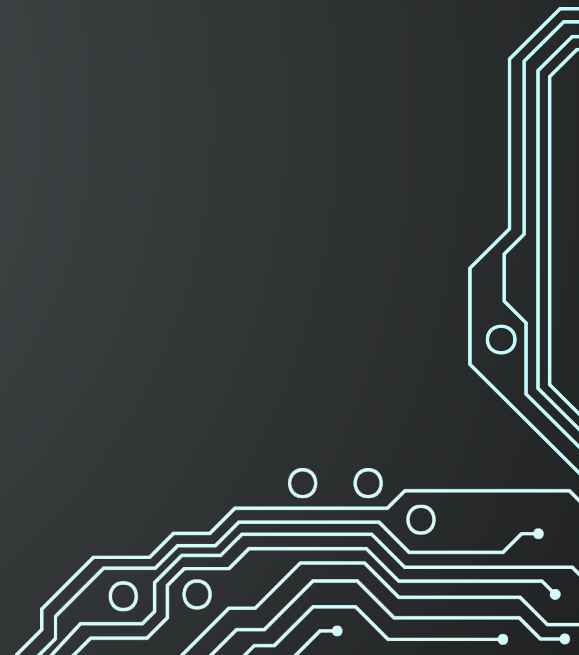
Optimized for performance

Can be configured to run multiple instances of each linter

Disable any linter you don’t want

Modular design - relatively easy to add new linters in the future

Dockerized



pkimetal: Supported linters

certlint (https://github.com/certlint/certlint)

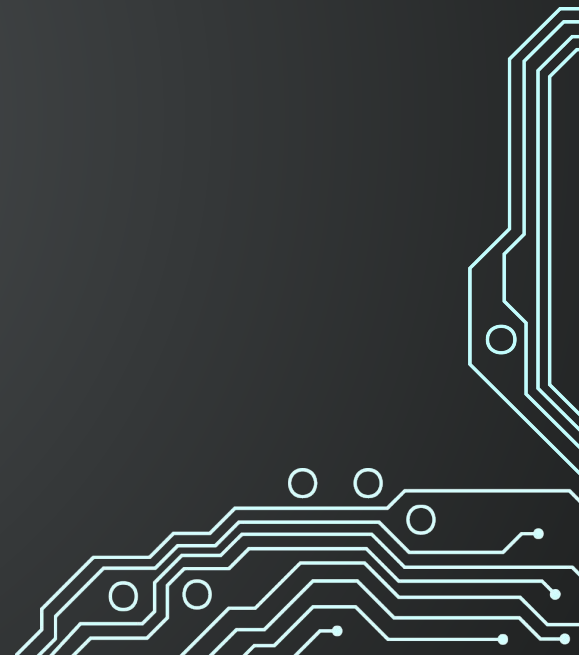
x509lint (https://github.com/kroeckx/x509lint)

zlint (https://github.com/zmap/zlint)

pkilint (https://github.com/digicert/pkilint)

dwklint (https://github.com/CVE-2008-0166/dwklint)

ftfy (https://github.com/rspeer/python-ftfy)



pkimetal: Status and Plans

Code will be open-sourced soon at https://github.com/pkimetal/pkimetal

Public instance will be available at (e.g.,) https://crt.sh/lintcert

crt.sh will use pkimetal for all of its linting functionality

If you're interested in using pkimetal, collaborating on further development, or have any feature requests, please reach out to me:
rob@sectigo.com

Any comments or questions?