# pkilint

Lint all the ASN.1-shaped things

# pkilint - goals

- Prototype “research project” started in late 2020
- High-level design goals:
  - PKI structure/document-agnostic
    - Don’t limit ourselves to just certificates
  - Support for validating multiple documents and their relationships
    - Other linters support validating only a single document at a time
  - Validate correct ASN.1/DER encoding
  - Allow for writing comprehensive tests that determine whether a specific field should be checked
    - Don’t require prescriptiveness in what is checked
  - Use a programming language that is accessible yet reasonably performant for high-volume issuance

# pkilint - architecture

- Each PKI structure is known as a document
  - Documents can have references to other documents
- Each document consists of nodes representing each field/element
  - Each node has a unique path within the document which identifies it
    - certificate.tbsCertificate.extensions.3.extnValue.subjectKeyIdentifier
- Validators check whether a node is applicable to the set of tests it performs and performs those tests
  - Validators can be made up of other Validators, allowing for easier organization of related tests
- Each node of the document is traversed, checking to see which Validators should run
  - “Visitor pattern”
- Findings are then gathered and presented as Results
- Use the popular ASN.1 Python package “pyasn1” and Russ Housley’s “pyasn1-alt-modules” package to supply ASN.1 definitions
  - Use “asn1ate” (ASN.1 to Python compiler) to fill in the missing pieces

# pkilint – example execution

Traversing an S/MIME certificate

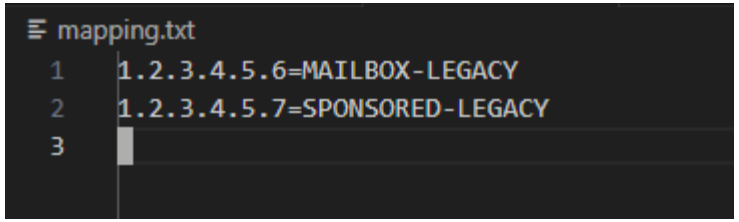
```
TimeCorrectEncodingValidator @ certificate.tbsCertificate.validity.notAfter
UtcTimeCorrectSyntaxValidator @ certificate.tbsCertificate.validity.notBefore.utcTime
UtcTimeCorrectSyntaxValidator @ certificate.tbsCertificate.validity.notAfter.utcTime
DomainComponentValidDomainNameValidator @ certificate.tbsCertificate.subject
SubscriberSubjectValidator @ certificate.tbsCertificate.subject.rdnSequence
RDNContainsUniqueTypesValidator @ certificate.tbsCertificate.subject.rdnSequence.0
RelativeDistinguishedNameContainsOneElementValidator @ certificate.tbsCertificate.subject.rdnSequence.0
SubjectAlternativeNameContainsSubjectEmailAddressesValidator @ certificate.tbsCertificate.subject.rdnSequence.0.0
OrganizationIdentifierAttributeValidator @ certificate.tbsCertificate.subject.rdnSequence.0.0
OrganizationIdentifierLeiValidator @ certificate.tbsCertificate.subject.rdnSequence.0.0.value.x520OrganizationIdentifier
RDNContainsUniqueTypesValidator @ certificate.tbsCertificate.subject.rdnSequence.1
RelativeDistinguishedNameContainsOneElementValidator @ certificate.tbsCertificate.subject.rdnSequence.1
SubjectAlternativeNameContainsSubjectEmailAddressesValidator @ certificate.tbsCertificate.subject.rdnSequence.1.0
RDNContainsUniqueTypesValidator @ certificate.tbsCertificate.subject.rdnSequence.2
RelativeDistinguishedNameContainsOneElementValidator @ certificate.tbsCertificate.subject.rdnSequence.2
SubjectEmailAddressInSanValidator @ certificate.tbsCertificate.subject.rdnSequence.2.0.value.emailAddress
MailboxAddressSyntaxValidator @ certificate.tbsCertificate.subject.rdnSequence.2.0.value.emailAddress
PrintableStringConstraintValidator @ certificate.tbsCertificate.subject.rdnSequence.0.0.value.x520OrganizationIdentifier.printableString
PrintableStringConstraintValidator @ certificate.tbsCertificate.subject.rdnSequence.1.0.value.x520OrganizationName.printableString
SmimeAllowedPublicKeyAlgorithmEncodingValidator @ certificate.tbsCertificate.subjectPublicKeyInfo.algorithm
RsaKeyValidator @ certificate.tbsCertificate.subjectPublicKeyInfo.subjectPublicKey.rSAPublicKey
GmailAllowedModulusLengthValidator @ certificate.tbsCertificate.subjectPublicKeyInfo.subjectPublicKey.rSAPublicKey
```

# pkilint - release

- Bundled with 6 command line programs, each wrapping a different linter
  - CABF S/MIME end-entity certificate
  - CABF TLS certificate (still preliminary)
  - PKIX certificate
  - PKIX and CABF CRL (CABF lints still preliminary)
  - PKIX OCSP
  - PKIX issuer and subject certificates relationship
- Easy-to-install Python package available on PyPi: https://pypi.org/project/pkilint/
- Source code available on Github: https://github.com/digicert/pkilint

# lint_cabf_smime_cert

- Command-line utility that wraps the S/MIME BR end-entity linter
- Requires that the S/MIME certificate type (validation level and generation) be specified
  - Specified explicitly on the command line (-t)
  - Via policy OIDs (-d)
    - CA/B Forum reserved policy OIDs
    - Mapping file (-m)
      - Text file with OID to type mappings
  - Heuristics (-g)
    - Uses attributes that are present in the subject DN to determine validation level
      - Always assumes LEGACY generation



# Thank you!