Requests group new fields within a logical topics, then document reason why it should be included (satisfied XXX criteria, etc.). Format for including into spec, can be seen at: https://spdx.github.io/spdx-spec/8-annotations/#82-annotation-date

8.2.1 Purpose: Identify when the comment was made. This is to be specified according to the combined date and time in the UTC format, as specified in the ISO 8601 standard.
8.2.2 Intent: Here, the Annotation Date can serve as a verification as to when the actual review was done.
8.2.3 Cardinality: Conditional (Mandatory, one), if there is an Annotation.
8.2.4 Data Format:

For the fields to be included - key is to group them in a related area (ie. Vulnerabilities, Lifecycle, …) And then specify the purpose of including a field (use case satisfies, standard adheres to, etc.)

Source: https://spdx-xml.specchain.org/spdx-security/file/dochtml

Software Evidence Archive

A data type for automated software supply chain Product and version information metadata

1. Software Information. A data item for software product naming or version related information
   type = SoftwareInformationType
   a. Product Title Text. A data item for the product title
      type = LineTextType
      simpletype = LineTextSimpleType
      base = xs:string
      maxLength = 48
      pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
   b. Software Name Text. A data item for the Name of the software
      type = LineTextType
      simpletype = LineTextSimpleType
      base = xs:string
      maxLength = 48
      pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
   c. Software Org Text. A data item for the Organization of the software
      type = LineTextType
      simpletype = LineTextSimpleType
      base = xs:string
      maxLength = 48
      pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
   d. Version Information. A data item for version information
      type = VersionInformationType
     (1) Software Version Text. A data item for the raw Version of the software
         type = LineTextType
         simpletype = LineTextSimpleType
         base = xs:string

maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(2) Semantic Version Indicator. True if semantic version; false if not or not known.

type = SemanticVersionIndicatorType
simpletype = SemanticVersionIndicatorSimpleType
base = xs:boolean

(3) Semantic Version Text. A data item a software product version

type = SemanticVersionTextType
simpletype = SemanticVersionTextSimpleType
base = xs:string
pattern = (0|[1-9]\d*)\.(0|[1-9]\d*)\.(0|[1-9]\d*)(-(0|[1-9]\d*|\d*[a-zA-Z-][0-9a-zA-Z-]*)(\.(0|[1-9]\d*|\d*[a-zA-Z-][0-9a-zA-Z-]*))*)?(\+[0-9a-zA-Z-]+(\.[0-9a-zA-Z-]+)*)?

(4) Semantic Version. A data type for semantic versioning information. A normal version number MUST take the form X.Y.Z where X, Y, and Z are non-negative integers, and MUST NOT contain leading zeroes. X is the major version, Y is the minor version, and Z is the patch version. Each element MUST increase numerically. For instance: 1.9.0 -> 1.10.0 -> 1.11.0.(https://semver.org/spec/v2.0.0.html)

type = SemanticVersionInformationType

(a) API Name. A data item for the name of the required public API declaration

type = APINameType
simpletype = APINameSimpleType
base = xs:string
maxLength = 24
pattern = -?[0-9A-Za-z-]{0,24}

(b) Major Version. A data item for the Major version. The value zero (0.y.z) is for initial development. Anything may change at any time. The public API should not be considered stable. Version 1.0.0 defines the public API. The way in which the version number is incremented after this release is dependent on this public API and how it changes. Major version X (X.y.z | X > 0) MUST be incremented if any backwards incompatible changes are introduced to the public API. It MAY include minor and patch level changes. Patch and minor version MUST be reset to 0 when major version is incremented.

type = MajorVersionType
simpletype = VersionSimpleType
base = xs:integer

(c) Minor Version. A data item for the Minor version. Minor version Y (x.Y.z | x > 0) MUST be incremented if new, backwards compatible functionality is introduced to the public API. It MUST be incremented if any public API functionality is marked as deprecated. It MAY be incremented if substantial new functionality or improvements are introduced within the private code. It MAY include patch level changes. Patch version MUST be reset to 0 when minor version is incremented.

type = MinorVersionType

simpletype = VersionSimpleType
base = xs:integer

(d) Patch Version. A data item for the Patch version. Patch version Z (x.y.Z | x > 0) MUST be incremented if only backwards compatible bug fixes are introduced. A bug fix is defined as an internal change that fixes incorrect behavior.

type = PatchVersionType
simpletype = VersionSimpleType
base = xs:integer

(e) Pre-Release Version. A pre-release version MAY be denoted by appending a hyphen and a series of dot separated identifiers immediately following the patch version. Identifiers MUST comprise only ASCII alphanumerics and hyphen [0-9A-Za-z-]. Identifiers MUST NOT be empty. Numeric identifiers MUST NOT include leading zeroes. Pre-release versions have a lower precedence than the associated normal version. A pre-release version indicates that the version is unstable and might not satisfy the intended compatibility requirements as denoted by its associated normal version. Examples: 1.0.0-alpha, 1.0.0-alpha.1, 1.0.0-0.3.7, 1.0.0-x.7.z.92.

type = PreReleaseVersionType
simpletype = PreReleaseVersionSimpleType
base = xs:string
pattern = -[0-9A-Za-z-]{0,24}

(f) Build Metadata. A data item for Build metadata. Build metadata MAY be denoted by appending a plus sign and a series of dot separated identifiers immediately following the patch or pre-release version. Identifiers MUST comprise only ASCII alphanumerics and hyphen [0-9A-Za-z-]. Identifiers MUST NOT be empty. Build metadata SHOULD be ignored when determining version precedence. Thus two versions that differ only in the build metadata, have the same precedence. Examples: 1.0.0-alpha+001, 1.0.0+20130313144700, 1.0.0-beta+exp.sha.5114f85.(https://semver.org/spec/v2.0.0.html)

type = BuildMetaType
simpletype = BuildMetaSimpleType
base = xs:string
maxLength = 24
pattern = \+[0-9A-Za-z.-]{0,24}

(5) Semantic Version Date. A data type for date and time information

type = DateTimeType
simpletype = DateTimeSimpleType
base = xs:dateTime

(6) Python Semantic Version Indicator. True if python semantic version; false if not or not known.

type = PythonSemanticVersionIndicatorType
simpletype = PythonSemanticVersionIndicatorSimpleType
base = xs:boolean

(7) Python Semantic Version. A data item for a Python Distribution identified by a public version identifier which supports all defined version comparison operations. The version scheme is used both to describe the distribution version provided by a particular distribution archive, as well as

to place constraints on the version of dependencies needed in order to build or run the software.The canonical public version identifiers MUST comply with the following scheme: [N!]N(.N)*[{a|b|rc}N][.postN][.devN]
(https://www.python.org/dev/peps/pep-0440/#public-version-identifiers)
type = PythonSemanticVersionType

(a) Epoch Segment. A data type for the Python Epoch Segment with the scheme: [N!].
(https://www.python.org/dev/peps/pep-0440/#public-version-identifiers).
type = PythonEpochVersionType
simpletype = PythonEpochVersionSimpleType
base = xs:string
pattern = [(\d+!)?]

(b) Release Segment. A data type for the Python Release Segment with the scheme: N(.N)*.
(https://www.python.org/dev/peps/pep-0440/#public-version-identifiers).
type = PythonReleaseVersionType
simpletype = PythonReleaseVersionSimpleType
base = xs:string
pattern = [(\d+)(\.\d+)+]

(c) Pre-Release Segment. A data type for the Python Pre-Release Segment with the scheme: {a|b|rc}N.
(https://www.python.org/dev/peps/pep-0440/#public-version-identifiers).
type = PythonPreReleaseVersionType
simpletype = PythonPreReleaseVersionSimpleType
base = xs:string
pattern = [(\d+!)?]

(d) Post Release Segment. A data type for the Python Post-Release Segment with the scheme: .postN
(https://www.python.org/dev/peps/pep-0440/#public-version-identifiers).
type = PythonPostReleaseVersionType
simpletype = PythonPostReleaseVersionSimpleType
base = xs:string
pattern = [(\.?(post)\d*)?]

(e) Development Release Segment. A data type for the Python development release segment with the scheme: .devN:
[N!]N(.N)*[{a|b|rc}N][.postN][.devN]
(https://www.python.org/dev/peps/pep-0440/#public-version-identifiers).
type = PythonPreReleaseVersionType
simpletype = PythonPreReleaseVersionSimpleType
base = xs:string
pattern = [(\d+!)?]

(8) Python Version. An augmentation point for SemanticVersionType
type = PythonVersionTextType
simpletype = PythonVersionTextSimpleType
base = xs:string
pattern = [0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}(\..*)?

e. Grouping Text. A data item for a software family or relationship
type = LineTextType

simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

f. Release Notes Text. A data item for a release notes
type = ParagraphTextType
simpletype = ParagraphTextSimpleType
base = xs:string
maxLength = 512
pattern = [a-zA-Z0-9\.,\?/':;]{0,512}

2. File Information. A data item for file types and related Multipurpose Internet Mail Extensions (MIME) types
type = FileInformationType

a. File Name Text. A data item for a computer file name without extension
type = ComputerFileNameType
simpletype = ComputerFileNameSimpleType
base = xs:string
maxLength = 256
pattern = [a-zA-Z_\-/]{1,256}

b. File Extension Text. A data item for a computer file extension
type = FileExtensionType
simpletype = ComputerFileExtensionSimpleType
base = xs:string
maxLength = 24
pattern = \.[a-zA-Z]{0,24}

3. Authoritative Source Information. A data item for information about what is to be considered the authoritative source for a given artifact or source repository. This includes an URL, a hash of the source, and whether or not the source has been signed
type = AuthoritativeSourceInformationType

a. Source Hash Text. A data item a for a sha1 encoded string
type = Sha256TextType
simpletype = Sha256TextSimpleType
base = xs:string
length = 64
pattern = [A-Fa-f0-9]{64}

b. Source URI Text. A data item for a World Wide Web Consortium Uniform Reference Indicator
type = URITextType
simpletype = URITextSimpleType
base = xs:string
maxLength = 512
pattern = [a-zA-Z0-9\.,\?/\\':;_\-]{0,512}

c. Authoritative Digital Signature Indicator. True if has an authoritative digital signatiure; false if not.
type = AuthoritativeDigitalSignatureIndicatorType
simpletype = AuthoritativeDigitalSignatureIndicatorSimpleType
base = xs:boolean

4. Ecosystem Information. A data item for information pertaining to a software project's ecosystem which includes programming languages, references, number of committers, mailing list activity, overall sentiment, and other information
type = EcosystemInformationType
a. Ecosystem. A data item for software ecosystem information
type = EcosystemType
(1) Committer Quantity. A data item for the number of committers to a software repository
type = DataCountType
simpletype = DataCountSimpleType
base = xs:integer
pattern = [0-9]{0,10000000}
(2) Languages. A data item for languages used in a software repository
type = LanguagesType
(a) Language Text. A data item for the name of a programming language
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
minOccurs = 1
maxOccurs = 100
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
(3) Commit Logs. A data item for entries associated with software repository commit events
type = CommitLogsType
(a) Commit Log Text. A data item for an entry associated with software repository commit event
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
minOccurs = 1
maxOccurs = 100
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
b. Company Information. A data item for languages used in a software repository
type = CompanyInformationType
(1) NameText. A data item for a name
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
5. Dependency Information. A data item for a list of dependencies for a given software project derived directly from the artifact or source dependency definition file. Entries include names, versions, and vulnerabilities

type = DependencyInformationType
a. Dependencies. A data item for software dependencies
type = DependenciesType
(1) Dependency. A data item for software dependency information
type = DependencyType
minOccurs = 1
maxOccurs = 100
(a) NameText. A data item for a name
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
minOccurs = 1
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
(b) Dependency Version Text. A data item for a dependency version
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
minOccurs = 1
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
b. Compiler. A data item for languages used in a software repository
type = CompilerType
(1) NameText. A data item for a name
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
(2) Compiler Version Text. A data item for a compiler version
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

6. License Information. A data item for software license information for a software artifact or source repository
type = LicenseInformationType
a. License Category Code. A data item for a software license category
type = LicenseCategoryCodeType
simpletype = LicenseCategoryCodeSimpleType
base = xs:string
enumerations = FOSS, COTS, GOSS
b. License Code. A data item for a type of software license
type = LicenseCodeType
simpletype = LicenseCodeSimpleType
base = xs:string
minOccurs = 1

maxOccurs = 20
enumerations = 3DFX, 3DFX-PL, AAL, ACAA, ACDL, ACE, Adaptec-GPL, Adaptec(RESTRICTED), Adobe-AFM, Adobe-EULA, AFL, AFL-1.0, AFL-1.1, AFL-1.2, AFL-2.0, AFL-2.1, AFL-3.0, AFPL-Ghostscript, AgainstDRM, AGE-Logic, Agere-EULA, AGFA-EULA, AGFA(RESTRICTED), AGPL, AGPL-1.0, AGPL-1.0+, AGPL-3.0, AGPL-3.0+, Aladdin, Alfresco, Alfresco/FLOSS, Algorithmics, AMD, AMD-EULA, ANTLR-PD, AOL-EULA, Apache, Apache-1.0, Apache-1.1, Apache-2.0, Apache-style, Apache-v1.0, APL, APL-1.0, Apple(FontForge), Apple(Sample), APSL, APSL-1.0, APSL-1.1, APSL-1.2, APSL-2.0, APSL-style, Aptana, Aptana-1.0, ARJ, Arphic-Font-PL, Arphic-style, Artifex, Artistic-1.0-cl8, Artistic-1.0-Perl, Ascender-EULA, ATI-EULA, ATMEL-FW, ATT(Non-commercial), ATT-Source, ATT-Source-1.0, ATT-Source-1.2d, Baekmuk-Font, Baekmuk(Hwan), BancTec, Beerware, Bellcore, Bellcore-style, BH-Font, BH-Font-style, BISON, BitTorrent, BitTorrent-1.0, BitTorrent-1.1, BIZNET, BIZNET-style, BrainStorm-EULA, Broadcom-EULA, BSD-2-Clause, BSD-2-Clause-FreeBSD, BSD-2-Clause-NetBSD, BSD-3-Clause, BSD-3-Clause-Clear, BSD-3-Clause-Severability, BSD-4-Clause, BSD-4-Clause-UC, BSL-1.0, BSL-style, CA, Cadence, CATOSL, CATOSL-1.1, CC0-1.0, CC-BY, CC-BY-1.0, CC-BY-2.0, CC-BY-2.5, CC-BY-3.0, CC-BY-4.0, CC-BY-NC-1.0, CC-BY-NC-2.0, CC-BY-NC-2.5, CC-BY-NC-3.0, CC-BY-NC-4.0, CC-BY-NC-ND-1.0, CC-BY-NC-ND-2.0, CC-BY-NC-ND-2.5, CC-BY-NC-ND-3.0, CC-BY-NC-ND-4.0, CC-BY-NC-SA-1.0, CC-BY-NC-SA-2.0, CC-BY-NC-SA-2.5, CC-BY-NC-SA-3.0, CC-BY-NC-SA-4.0, CC-BY-ND-1.0, CC-BY-ND-2.0, CC-BY-ND-2.5, CC-BY-ND-3.0, CC-BY-ND-4.0, CC-BY-SA, CC-BY-SA-1.0, CC-BY-SA-2.0, CC-BY-SA-2.5, CC-BY-SA-3.0, CC-BY-SA-4.0, CC-LGPL, CC-LGPL-2.1, CCLRC, CCPL, CDDL, CDDL-1.0, CDDL-1.1, CECILL, CECILL-2.0, CECILL-B, CECILL-C, CECILL(dual), Cisco, Cisco-style, Citrix, ClArtistic, ClearSilver, CMake, CMU-style, CNRI-Python, CNRI-Python-GPL-Compatible, Combined_OpenSSL+SSLeay, CompuServe, Comtrol, Condor-1.0, Condor-1.1, CopyLeft[1], CopyLeft[2], CPL, CPL-0.5, CPL-1.0, CPOL, CPOL-1.02, Cryptogams, CUA-OPL-1.0, CUPS, CUPS-EULA, Cylink-ISC, Cypress-FW, DARPA, DARPA-Cougaar, Debian-social-DFSG, Debian-SPI, Debian-SPI-style, D-FSL-1.0, DMTF, DOCBOOK, DOCBOOK-style, DPTC, DSCT, DSL, Dyade, EBT-style, ECL-1.0, ECL-2.0, eCos-2.0, EFL, EFL-1.0, EFL-2.0, eGenix, Empty-file-no-data!, Entessa, Epinions, Epson-EULA, Epson-PL, ErlPL-1.1, EUDatagrid, EUPL-1.0, EUPL-1.1, FaCE, Fair, Fair-style, FAL-1.0, FAL-1.3, FedoraCLA, Flash2xml-1.0, Flora, Flora-1.0, Flora-1.1, Frameworx, Frameworx-1.0, FreeBSD-Doc, Free-PL, Free-SW, FTL, FTL-style, Fujitsu, Garmin-EULA, Genivia(Commercial), GFDL, GFDL-1.1, GFDL-1.1+, GFDL-1.2, GFDL-1.2+, GFDL-1.3, GFDL-v1.2, Ghostscript-GPL-1.1, Giftware, GNU-copyleft, GNU-Ghostscript, GNU-javamail-exception, GNU-Manpages, Google-BSD, Govt-restrict, GPL, GPL-1.0, GPL-1.0+, GPL-2.0, GPL-2.0+, GPL-2.0:3.0, GPL-2.0+:3.0, GPL-2.0+KDEupgradeClause, GPL-2.0-with-autoconf-exception, GPL-2.0-with-bison-exception, GPL-2.0+-with-bison-exception, GPL-2.0-with-classpath-exception, GPL-2.0+-with-classpath-exception, GPL-2.0-with-font-exception, GPL-2.0-with-GCC-exception, GPL-2.1[sic], GPL-2.1+[sic], GPL-3?, GPL-3.0, GPL-3.0+, GPL-3.0-with-autoconf-exception, GPL-3.0-with-GCC-exception, GPL-3.0+-with-GCC-exception, GPL-Meta, GPL-or-LGPL, GPL(rms), GPL-with-autoconf-exception, gSOAP, gSOAP-1.3b, H2, H2-1.0, Hacktivismo, Hauppauge, Helix/RealNetworks-EULA, HPND, IBM-Courier, IBM-JCL,

IBM-pibs, IBM-reciprocal, ID-EULA, IDPL, IDPL-1.0, IEEE-Doc, IETF, IETF-style,
IJG, ImageMagick, ImageMagick(Apache), ImageMagick-style, Imlib2, InfoSeek,
info-zip, InnerNet, InnerNet-2.00, InnerNet-style, Intel-EULA,
Intel(RESTRICTED), Intel-WLAN, Interbase-1.0, Interbase-PL, Interlink-EULA,
Intranet-only, IOS, IoSoft(COMMERCIAL), IPA, IPA-Font-EULA, IPL, IPL-1.0,
IPL-2.0, IPTC, IronDoc, ISC, Jabber, Java-WSDL4J, Java-WSDL-Policy,
Java-WSDL-Schema, Java-WSDL-Spec, JISP, JPNIC, JSON, KDE, Keyspan-FW,
KnowledgeTree-1.1, Knuth-style, Lachman-Proprietary, Larabie-EULA, LDP-1A,
LDP-2.0, Legato, Leptonica, LGPL, LGPL-1.0, LGPL-1.0+, LGPL-2.0, LGPL-2.0+,
LGPL-2.1, LGPL-2.1+, LGPL-2.1+-KDE-exception, LGPL-3?, LGPL-3.0, LGPL-3.0+,
LIBGCJ, Libpng, Link-exception, LinuxDoc, Linux-HOWTO, Logica-OSL-1.0, LPL-1.0,
LPL-1.02, LPPL, LPPL-1.0, LPPL-1.0+, LPPL-1.1, LPPL-1.1+, LPPL-1.2, LPPL-1.2+,
LPPL-1.3, LPPL-1.3+, LPPL-1.3a, LPPL-1.3a+, LPPL-1.3b, LPPL-1.3b+, LPPL-1.3c,
LPPL-1.3c+, MacroMedia-RPSL, Macrovision, Macrovision-EULA, Majordomo,
Majordomo-1.1, Mandriva, Mellanox, MetroLink, MetroLink-nonfree, Mibble,
Mibble-2.8, Microsoft, Migemo, MindTerm, MirOS, MITEM, MitreCVW-style,
Motosoto, MPEG3-decoder, MPL, MPL-1.0, MPL-1.1, MPL-1.1+, MPL-1.1-style,
MPL-2.0, MPL-2.0-no-copyleft-exception, MPL-EULA-1.1, MPL-EULA-2.0,
MPL-EULA-3.0, MPL-style, M-Plus-Project, MRL, MS-indemnity, MS-LPL, MS-LRL,
Multics, MX4J, MX4J-1.0, MySQL-0.3, MySQL/FLOSS, MySQL-style, NASA, NASA-1.3,
Naumen, NBPL-1.0, nCipher, NCSA, NESSUS-EULA, NGPL, Nokia, No_license_found,
non-ATT-BSD, None, NOSL-1.0, Not-for-sale!, Not-Free!, Not-Internet,
Not-OpenSource!, NOT-Open-Source!, NOT-public-domain, Novell, Novell-EULA, NPL,
NPL-1.0, NPL-1.1, NPL-1.1+, NPL-EULA, NPOSL-3.0, NRL, NTP, OASIS, OCL, OCL-1.0,
OCLC, OCLC-1.0, OCLC-2.0, OCL-style, ODbL-1.0, ODL, OFL-1.0, OFL-1.1, OGTSL,
OLDAP, OLDAP-1.1, OLDAP-1.2, OLDAP-1.3, OLDAP-1.4, OLDAP-2.0, OLDAP-2.0.1,
OLDAP-2.1, OLDAP-2.2, OLDAP-2.2.1, OLDAP-2.2.2, OLDAP-2.3, OLDAP-2.4,
OLDAP-2.5, OLDAP-2.6, OLDAP-2.7, OLDAP-2.8, OLDAP-style, OMF, Ontopia,
OpenCASCADE-PL, OpenGroup, OpenGroup-Proprietary, OpenMarket, Open-PL,
Open-PL-0.4, Open-PL-1.0, Open-PL-style, OPL-1.0, Oracle-Berkeley-DB, O'Reilly,
O'Reilly-style, OSD, OSL, OSL-1.0, OSL-1.1, OSL-2.0, OSL-2.1, OSL-3.0,
Paradigm, Patent-ref, PDDL-1.0, Phorum, PHP-3.0, PHP-style, Piriform,
Pixware-EULA, Polyserve-CONFIDENTIAL, Postfix, PostgreSQL, Powder-Proprietary,
Princeton, Princeton-style, Proprietary!, Public-domain, Public-domain(C),
Public-domain-ref, Public-Use, Public-Use-1.0, Python, Python-2.0,
Python-2.0.1, Python-2.0.2, Python-2.1.1, Python-2.1.3, Python-2.2,
Python-2.2.3, Python-2.2.7, Python-2.3, Python-2.3.7, Python-2.4.4,
Python-style, Qmail, QPL, QPL-1.0, QT(Commercial), QuarterDeck, Quest-EULA,
RCSL, RCSL-1.0, RCSL-2.0, RCSL-3.0, RealNetworks-EULA, RealNetworks-Unknown,
RedHat-EULA, Redland, Restricted-rights, RHeCos-1.1, Riverbank-EULA, RPL-1.5,
RPSL, RPSL-1.0, RPSL-2.0, RPSL-3.0, RSCPL, Ruby, SAX-PD, SciTech,
SCO(commercial), SCSL, SCSL-2.3, SCSL-3.0, SCSL-TSA, SCSL-TSA-1.0,
See-doc(OTHER), See-file, See-file(COPYING), See-file(LICENSE),
See-file(README), See-URL, Sendmail, SGI, SGI-B-1.0, SGI-B-1.1, SGI-B-2.0,
SGI-Freeware, SGI_GLX, SGI-Proprietary, SGI-style, SGML, SimPL-2.0, SISSL,
SISSL-1.1, SISSL-1.2, Skype-EULA, Sleepycat(Non-commercial), SMLNJ, SNIA,
SNIA-1.0, SNIA-1.1, SpikeSource, SPL, SPL-1.0, Stanford, Stanford-style,
SugarCRM-1.1.3, Sun-BCLA-1.5.0, Sun-EULA, Sun-IP, Sun-Java,

Sun(Non-commercial), SunPro, Sun(RESTRICTED), Sun-RPC, Sun-SCA, Sun(tm), SW-Research, Tapjoy, Tektronix, Tektronix-style, Trademark-ref, Trident-EULA, Trolltech, U-BC, U-Cambridge, U-Cambridge-style, UCAR, UCAR-style, U-Chicago, U-Columbia, UCWare-EULA, U-Del, U-Del-style, U-Monash, UnclassifiedLicense, Unidex, UnitedLinux-EULA, Unlicense, unRAR restriction, URA(gov't), USC, USC(Non-commercial), USC-style, US-Export-restrict, U-Utah, U-Wash(Free-Fork), VIM, Vixie, Vixie-license, VMware-EULA, VSL-1.0, Wash-U-style, Watcom, Watcom-1.0, WebM, Wintertree, WordNet-3.0, WTFPL, WTI(Not-free), WXwindows, X11, Xerox, Xerox-style, XFree86, XFree86-1.0, XFree86-1.1, Ximian, Ximian-1.0, XMLDB-1.0, Xnet, XOPEN-EULA, X/Open-style, Yahoo-EULA, YaST(SuSE), YPL, YPL-1.0, YPL-1.1, Zend-2.0, Zeus, Zimbra, Zimbra-1.2, Zimbra-1.3, Zlib, ZoneAlarm-EULA, ZPL-1.0, ZPL-1.1, ZPL-2.0, ZPL-2.1, Zveno

c. End Of Life Indicator. True if software has end of life status; false if not.

type = EndOfLifeIndicatorType
simpletype = EndOfLifeIndicatorSimpleType
base = xs:boolean

7. Vulnerability Information. A data item for vulnerability or virus information for a software artifact or source repository

type = VulnerabilityInformationType

a. Vulnerabilities. A data item for vulnerabilty information

type = VulnerabilitiesType

(1) Vulnerability. A data item for vulnerabilty information

type = VulnerabilityType
minOccurs = 1
maxOccurs = 100

(a) Vulnerability Identifier Text. A data item for a vulnerabilty identifier

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(b) Source Text. A data item describing the vulnerabilty source

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(c) Vulnerability Title Text. A data item for a vulnerability title

type = ParagraphTextType
simpletype = ParagraphTextSimpleType
base = xs:string
maxLength = 512
pattern = [a-zA-Z0-9\.,\?/':;]{0,512}

(d) Vulnerability Summary Text. A data item for a vulnerability summary paragraph

type = ParagraphTextType
simpletype = ParagraphTextSimpleType
base = xs:string
maxLength = 512
pattern = [a-zA-Z0-9\.,\?/':;]{0,512}

(e) Vulnerability Score Value. A data item for a vulnerability score

type = FloatValueType
simpletype = FloatValueSimpleType
base = xs:float

(f) Vulnerability Score Version Value. A data item for a vulnerability score version

type = FloatValueType
simpletype = FloatValueSimpleType
base = xs:float

(g) Vulnerability Score System Text. A data item for a vulnerability score system

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(h) Score Details. A data item for common vulnerability score details

type = ScoreDetailsType

1. Common Vulnerability Scoring System V2. A data type for Common Vulnerability Scoring System V2 (CVSSV2) Information

type = CommonVulnerabilityScoringSystemV2Type

a. Vector String Text. A data item for a vulnerability vector string

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

b. Access Vector Text. A data item for a vulnerability access vector

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

c. Access Complexity Text. A data item for a access complexity

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string

maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

d. Vulnerability Authentication Text. A data item for authentication

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

e. Confidentiality Impact Text. A data item for confidentiality impact

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

f. Integrity Impact Text. A data item for a integrity impact

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

g. Availability Impact Text. A data item for a availability impact

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

h. Vulnerability Base Score Value. A data item for a vulnerability base score

type = FloatValueType
simpletype = FloatValueSimpleType
base = xs:float

2. Common Vulnerability Scoring System V3. A data type for Common Vulnerability Scoring System V3 (CVSSV3) Information

type = CommonVulnerabilityScoringSystemV3Type
minOccurs = 0
maxOccurs = 1

a. Vector String Text. A data item for a vulnerability vector string

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

b. Access Vector Text. A data item for a vulnerability access vector

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

c. Access Complexity Text. A data item for a access complexity

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

d. Privileges Required Text. A data item for privileges required

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

e. User Interaction Text. A data item for user interaction

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

f. Vulnerability Scope Text. A data item for vulnerability scope

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

g. Confidentiality Impact Text. A data item for confidentiality impact

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

h. Integrity Impact Text. A data item for a integrity impact

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

i. Availability Impact Text. A data item for a availability impact

type = LineTextType

simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

j. Vulnerability Base Score Value. A data item for a vulnerability base score
type = FloatValueType
simpletype = FloatValueSimpleType
base = xs:float

k. Base Severity Text. A data item for base severity
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(i) Vulnerability Vector Text. A data item for a vector
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(j) Access Complexity Text. A data item for a access complexity
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(k) Vulnerability Authentication Text. A data item for authentication
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(l) Confidentiality Impact Text. A data item for confidentiality impact
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(m) Integrity Impact Text. A data item for a integrity impact
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(n) Availability Impact Text. A data item for a availability impact

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(o) Vulnerability Source Text. A data item for a vulnerabilty source
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(p) Assessment Check Text. A data item for a vulnerabilty source
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(q) Scanner Text. A data item for a vulnerabilty source
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(r) Recommendation Text. A data item for a recommendation paragraph
type = ParagraphTextType
simpletype = ParagraphTextSimpleType
base = xs:string
maxLength = 512
pattern = [a-zA-Z0-9\.,\?/':;]{0,512}

(s) Vulnerability Reference. A data item for reference details for a vulnerability
type = VulnerabilityReferenceType
minOccurs = 0
maxOccurs = 1

1. Type Of Reference Text. A data item for the type of reference
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}
2. Reference Source Text. A data item for the reference source
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

3. URI Text. A data item for a World Wide Web Consortium Uniform Reference Indicator

type = URITextType
simpletype = URITextSimpleType
base = xs:string
maxLength = 512
pattern = [a-zA-Z0-9\.,\?/\\':;_\-]{0,512}

4. Reference Text. A data item for a reference description

type = ParagraphTextType
simpletype = ParagraphTextSimpleType
base = xs:string
maxLength = 512
pattern = [a-zA-Z0-9\.,\?/':;]{0,512}

(t) Vulnerability Modified Date. A data type for date and time vulnerability modified

type = DateTimeType
simpletype = DateTimeSimpleType
base = xs:dateTime

(u) Vulnerability Published Date. A data type for date and time vulnerability published

type = DateTimeType
simpletype = DateTimeSimpleType
base = xs:dateTime

b. Virus Information. A data item for information collected from virus scans

type = VirusInformationType

(1) Known Virus Quantity. A data item for the number of known viruses

type = DataCountType
simpletype = DataCountSimpleType
base = xs:integer
pattern = [0-9]{0,10000000}

(2) Engine Version Text. A data item for a virus engine version

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(3) Virus Engine Text. A data item for an entry associated with software repository commit event

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(4) Scanned Directory Quantity. A data item for the number of directories scanned

type = DataCountType
simpletype = DataCountSimpleType

base = xs:integer
pattern = [0-9]{0,10000000}

(5) Scanned File Quantity. A data item for the number of files scanned
type = DataCountType
simpletype = DataCountSimpleType
base = xs:integer
pattern = [0-9]{0,10000000}

(6) Infected File Quantity. A data item for the number of files infected by a virus
type = DataCountType
simpletype = DataCountSimpleType
base = xs:integer
pattern = [0-9]{0,10000000}

(7) Data Scanned Text. A data item for an entry associated with software repository commit event
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(8) Data Read Text. A data item for an entry associated with software repository commit event
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(9) Time Text. A data item for a text time entry
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(10) File Notes Text. A data item a for a multi-line entry for file notes
type = ParagraphTextType
simpletype = ParagraphTextSimpleType
base = xs:string
maxLength = 512
pattern = [a-zA-Z0-9\.,\?/':;]{0,512}

(11) Engine Details. A data item for for details of a virus engine
type = EngineDetailsType

(a) Engine Version Text. A data item for a virus engine version
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

(b) Database Version Text. A data item for a virus database version
type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

8. Governance Risk Compliance. A data item for calculated risk and compliance to governance for a software artifact or source repository

type = GovernanceRiskComplianceType

a. Risk Code. A data item for the level of risk associated with a software artifact

type = RiskCodeType
simpletype = RiskCodeSimpleType
base = xs:string
enumerations = low, medium, high, critical

b. Statement Of Assurance Text. A data item for a statement of assurance

type = ParagraphTextType
simpletype = ParagraphTextSimpleType
base = xs:string
maxLength = 512
pattern = [a-zA-Z0-9\.,\?/':;]{0,512}

9. Delivery Information. A data item for delivery information including the target url and last delivered date and time for a previously signed Software Evidence Archive.

type = DeliveryInformationType

a. Delivering Organization Text. A data item for the Organization of Delivery for the security

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

b. Destination Text. A data item for a destination

type = LineTextType
simpletype = LineTextSimpleType
base = xs:string
maxLength = 48
pattern = [a-zA-Z0-9\.,\?/':;]{0,48}

c. Hash Text. A data item for a hash of the artifact delivered

type = Sha256TextType
simpletype = Sha256TextSimpleType
base = xs:string
length = 64
pattern = [A-Fa-f0-9]{64}

d. Last Registry Update Date. A data type for date and time of last registry update

type = DateTimeType
simpletype = DateTimeSimpleType

```
base = xs:dateTime
```