The Common Vulnerability Reporting Framework

Dictionary of Elements

Last Update: 2012-05-07T10:26:00-08:00
Mike Schiffman, Cisco Systems, Inc., mschiffm@cisco.com

The Common Vulnerability Reporting Framework (CVRF) Dictionary of Elements is the definitive reference for the CVRF language, version 1.1.This is a living document managed by the Internet Consortium for Advancement of Security on the Internet (ICASI). It should be maintained and kept current with the CVRF Mindmap and CVRF Schema documents of the same version. Please note that CVRF 1.1 is not backward compatible with 1.0.

The following dictionary is grouped by schema, but is not derivative documentation from the schema documents themselves. The information contained here is more descriptive and complete.

Table of Contents

CVRF DOCUMENT ELEMENTS

Document Title

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Root

Document Title is a definitive canonical name for the document, providing enough descriptive content to differentiate from other similar documents, ideally providing a unique “handle.” While this field is largely up to the document producer, ICASI has some recommendations:

The title should be succinct and promptly give the reader an idea of what is to come. If the document producer also publishes a human-friendly document that goes hand-in-hand with a CVRF document, it is recommend that both documents use the same title. It is further recommended to include the manufacturer name with any product names mentioned in the title.

Examples:

-
  ```
  <DocumentTitle>

  Cisco IPv6 Crafted Packet Vulnerability

  </DocumentTitle>
  ```

-
  ```
  <DocumentTitle>

  CERT Vulnerabilities in Kerberos 5 Implementation

  </DocumentTitle>
  ```

-
  ```
  <DocumentTitle>

  Cisco Content Services Switch 11000 Series DNS Negative Cache of Information
  Denial-of-Service Vulnerability

  </DocumentTitle>
  ```

-
  ```
  <DocumentTitle>

  Symantec Brightmail AntiSpam Static Database Password

  </DocumentTitle>
  ```

-
  ```
  <DocumentTitle>

  HPSBUX02697 SSRT100591 rev.1 - HP-UX Running Java, Remote Unauthorized
  Access, Disclosure of Information, and Other Vulnerabilities

  </DocumentTitle>
  ```

-
  ```
  <DocumentTitle>

  Microsoft Vulnerability in the Microsoft Data Access Components (MDAC)
  Function Could Allow Code Execution

  </DocumentTitle>
  ```

-
  ```
  <DocumentTitle>

  Microsoft Vulnerability in Windows Explorer Could Allow Remote Code
  Execution

  </DocumentTitle>
  ```

Document Type

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Root

Document Type is a short canonical name, chosen by the document producer, which will inform the end user as to the type of document.

Examples:

-
  ```
  <DocumentType>
  Vulnerability Report
  </DocumentType>
  ```

-
  ```
  <DocumentType>
  Security Bulletin
  </DocumentType>
  ```

-
  ```
  <DocumentType>
  Security Notice
  </DocumentType>
  ```

Document Publisher

Data Type	container
Range	--
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Root
Children	Contact Details, Issuing Authority
Attribute	Type, Vendor ID
Attribute Data Type	enumerated list, string
Attribute Range	{Vendor, Discoverer, Coordinator, User, Other}, unrestricted
Attribute Required	yes, no

Document Publisher is a container that holds all the information about the publisher of the CVRF document, including attributes denoting the *Type* of publisher and an optional *Vendor ID* as well as optional elements for **Contact Details** and **Issuing Authority**.

Document Publisher is a required element, but the only required data is the *Type* attribute. The **Document Publisher** *Type* attribute is an enumerated list containing an array of different document publisher types. Types include:

- **Vendor**: Developers or maintainers of information system products or services. This includes all authoritative product vendors, Product Security Incident Response Teams (PSIRTs), and product resellers and distributors, including authoritative vendor partners.
- **Discoverer**: Individuals or organizations that find vulnerabilities or security weaknesses. This includes all manner of researchers.
- **Coordinator**: Individuals or organizations that manage a single vendor’s response or multiple vendors’ responses to a vulnerability, a security flaw, or an incident. This includes

all Computer Emergency/Incident Response Teams (CERTs/CIRTs) or agents acting on the behalf of a researcher.

- **User**: Everyone using a vendor's product.
- **Other**: Catchall for everyone else. Currently this includes forwarders, republishers, language translators, and miscellaneous contributors.

The optional *Vendor ID* attribute is a unique identifier (OID) that a vendor uses as issued by FIRST under the auspices of IETF. At the time of this writing, OID is a work in progress.

Example:

```
<DocumentPublisher Type="Vendor" VendorID="MarcusCom"/>
```

Contact Details

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Document Publisher

Contact Details contains information required to contact the document publisher.

Example:

```
<ContactDetails>

Name: Captain Sledge Fisthammer\r\nOrganization: International Space Explorers
of America\r\nPhone Number: 555-123-4567\r\nFax Number: 555-123-4568\r\nEmail
Address: sledge@foo.com

</ContactDetails>
```

Issuing Authority

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Document Publisher

Issuing Authority states the name of the issuing party and that party's authority to release the document. In particular, it addresses the party's constituency and responsibilities or other obligations. This element should also include instructions for contacting the issuer.

Example:

```
<IssuingAuthority>

The Juniper SIRT (Juniper Networks Security Incident Response Team) is the sole
authority regarding vulnerabilities in any Juniper Networks products or
services, and coordinates the handling of all aspects of such vulnerabilities
from initial discovery or report through public announcements and any
subsequent follow-on activities. Additional information is available at
http://www.juniper.net/support/security/report_vulnerability.html

</IssuingAuthority>
```

Document Tracking

Data Type	container
Range	--
Minimum Occurrences	1 (required)
Maximum Occurrences	1
Parent	Root
Children	Identification, Status, Version, Revision History, Initial Release Date, Current Release Date, Generator

The **Document Tracking** container contains all the attributes necessary to track a CVRF document.

Identification

Data Type	container
Range	--
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Document Tracking

The **Identification** container holds all the identifiers for the CVRF document. Required is the **ID** element, optional is the **Alias** element.

ID

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Identification

ID is a short, unique identifier used to refer to the document unambiguously in any context. The ID is a simple label. It is a string data type to provide for a wide range of numbering values, types, and schemes. Typically, the ID should be assigned and maintained by the original document issuing authority. It is recommended that the ID be a monotonically increasing value, or increasing in such a predictable manner that it does not contribute toward confusion or misnumbering. Careful consideration is required to ensure that construction of the ID does not contribute to confusion or collision with other labels.

Examples:

- `<ID>01</ID>`
- `<ID>29834841</ID>`
- `<ID>0xABCDEF</ID>`
- `<ID>100-200-301</ID>`

Alias

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Identification

Alias is an optional alternative ID used to refer to the document. Many vendors have one or more alternative or secondary IDs for documents and the **Alias** presents an interface to publish those alongside the primary ID.

Status

Data Type	enumerated list
Range	{Draft, Interim, Final}
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Document Tracking

Status refers to the condition of the document with regard to completeness and the likelihood of future editions.

Status types are:

- **Draft:** Pre-release, intended for issuing party's internal use only, or possibly used externally when the party is seeking feedback or indicating its intentions regarding a specific issue.
- **Interim:** The issuing party believes the content is subject to change.
- **Final:** The issuing party asserts the content is unlikely to change. "Final" status is an indication only, and does not preclude updates.

Issuing parties are strongly recommended to set **Status** to "Draft" when initiating a new document and to implement procedures to ensure that the status is changed to the appropriate value before the document is released.

Version

Data Type	token
Range	unrestricted.unrestricted.unrestricted.unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Document Tracking

Version is a simple counter to track the version of the document. This is a numeric tokenized field of the format "nn" - "nn.nn.nn.nn". It may be incremented in either major or minor notation to denote clearly the evolution of the content of the document. Issuing parties must ensure that this field is incremented appropriately, even for the least editorial or grammatical changes, when the field is used. It is validated using the following regular expression: `(0|[1-9][0-9]*)(\.(0|[1-9][0-9]*)){0,3}`.

Examples:

- `<Version>1.0</Version>`
- `<Version>1.0.1</Version>`

- `<Version>1.0.0.1</Version>`

Revision History

Data Type	container
Range	--
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Document Tracking
Children	Revision

Revision History should contain one **Revision** entry for each version/revision of the document, including the initial version and entries for each subsequent update.

Revision

Data Type	container
Range	--
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Revision History
Children	Number, Date, Description

Revision contains all the elements required to track the evolution of a CVRF document. Each change to a CVRF document should be accompanied by **Number**, **Date**, and **Description** elements.

Number

Data Type	token
Range	unrestricted.unrestricted.unrestricted.unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Revision

Number should contain the numeric version of the document. Like the **Version** element above, it is a numeric tokenized field of the format “nn” with up to four fields “nn.nn.nn.nn”. It is recommended that this be a monotonically increasing value. Minor revisions should be used for less-significant changes (for example, 1.0.0.0 to 1.0.0.1). Major, actionable changes should lead to a major increase of the version number (for example, 1.0 to 2.0).

Examples of such changes include:

- Any change to severity or impact
- The announcement of additional vulnerabilities
- The announcement of additional vulnerable products
- A significant change in remediation status

The most recent **Number** element should *always* match the **Version** element. It is validated using the following regular expression: `(0|[1-9][0-9]*)(\.(0|[1-9][0-9]*)){0,3}`.

Date

Data Type	dateTime
Range	--
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Revision

Date should record the date the revision was made. All dateTime values in CVRF require a time, and we recommend the inclusion of a time zone as well (ICASI endorses the use of Greenwich mean time [GMT] or “Zulu time”). If a time zone is excluded, Zulu should be assumed.

Description

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Revision

Description should be a short description of the changes made. It can describe the conditions that prompted the change or be a short list of the items changed.

Example:

```
<RevisionHistory>
    <Revision>
        <Number>1</Number>
        <Date>2011-11-26T00:00:00+00:00</Date>
        <Description>initial public release</Description>
    </Revision>
</RevisionHistory>
```

Initial Release Date

Data Type	dateTime
Range	--
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Document Tracking

Initial Release Date is the date (and time, optionally) that the document was initially released by the issuing party. All dateTime values in CVRF require a time, and we recommend the inclusion of a time zone as well (ICASI endorses the use of GMT or “Zulu time”). If a time zone is excluded, Zulu should be assumed.

Example:

```
<InitialReleaseDate>2011-11-26T00:00:00+00:00</InitialReleaseDate>
```

Current Release Date

Data Type	dateTime
Range	--
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Document Tracking

Current Release Date is the current date (and time, optionally) that the document was released by the issuing party. All dateTime values in CVRF require a time, and we recommend the inclusion of a time zone as well (ICASI endorses the use of GMT or "Zulu time"). If a time zone is excluded, Zulu should be assumed.

Example:

```
<CurrentReleaseDate>2011-11-26T00:00:00+00:00</CurrentReleaseDate>
```

Generator

Data Type	container
Range	--
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Document Tracking
Children	Engine, Date

The **Generator** container contains all the elements related to the generation of the document. These items will reference when the document was actually created, including the date it was generated and the entity that generated it.

Engine

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Generator

Engine will refer to the name and optional version of the engine that generated the CVRF document.

Date

Data Type	dateTime
Range	--
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Generator

Date will refer to the date the CVRF document was generated. Because documents are often generated internally by a document producer and exist for a nonzero amount of time before being released, this field can be different from the **Initial Release Date**. All dateTime values in CVRF require a time, and we recommend the inclusion of a time zone as well (ICASI endorses the use of GMT or "Zulu time"). If a time zone is excluded, Zulu should be assumed.

Example:

```
<Generator>
    <Engine>Mike Schiffman's sublime fingertips version 1.0</Engine>
    <Date>2012-02-27T00:00:00+00:00</Date>
</Generator>
```

Document Notes

Data Type	container
Range	--
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Root
Children	Note

Document Notes contains all of the document-level **Note** elements.

Note

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Document Notes
Children	--
Attribute	Type, Ordinal, Title, Audience
Attribute Data Type	enumerated list, positiveInteger, string, string
Attribute Required	yes, yes, no, no

Note is a place to put all manner of text blobs related to the document as a whole. It can be a concise summary of the overall document or a more compartmentalized and area-specific textual discussion. Depending on the need, there can be zero, one, or several **Note** elements in a given CVRF document.

The note should contain a compartmentalized textual discussion constrained by its *Type* attribute. *Type* can be one of the following:

- **General:** A general, high-level note (*Title* may have more information).
- **Details:** A low-level detailed discussion (*Title* may have more information).
- **Description:** A description of something (*Title* may have more information).
- **Summary**: A summary of something (*Title* may have more information).
- **FAQ**: A list of frequently asked questions.
- **Legal Disclaimer**: Any possible legal discussion, including constraints, surrounding the document.
- **Other**: Something that doesn't fit (*Title* should have more information).

Title and *Audience* are optional attributes to give human readers context around what they are about to read; *Title* should be a concise description of what is contained in the text, whereas *Audience* will indicate who is intended to read it.

For example, when *Type* is “General,” *Title* is “executive summary,” and *Audience* is “executives,” the note is a high-level overview designed for consumption by C-level decision makers. It should be brief and devoid of any technical details and jargon. On the other hand, when *Type* is “Details,” *Title* is “technical summary,” and *Audience* is “operational management and system administrators,” the note will be more detailed in nature and will contain more operational information.

Ordinal is a mandatory, locally significant value used to track notes inside a CVRF document at the root (document) level. It is provided to uniquely identify a **Note**. There should be one of these values for every **Note** inside **Document Notes**, and it is recommended that *Ordinal* should be instantiated as a monotonically increasing counter, indexed from 1. Each *Ordinal* that tracks a **Note** inside **Document Notes** is completely independent from an *Ordinal* tracking a **Note** inside **Vulnerability/Notes**.

Example:

```
<DocumentNotes>
    <Note Type=”General” Ordinal=”1” Title="Details" Audience="All">
    These are some details about a CVRF document intended for all stakeholders.
    </Note>
</DocumentNotes>
```

Document Distribution

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Root

Document Distribution should contain details about constraints, if any, for sharing the CVRF document with additional recipients. Constraints may include instructions on how to reproduce, share, copy, or otherwise distribute the document.

Aggregate Severity

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Root
Attribute	Namespace
Attribute Data Type	anyURI
Attribute Range	unrestricted
Attribute Required	no

Aggregate Severity is provided by the document producer to convey the urgency and criticality with which the vulnerability or vulnerabilities should be addressed. It is a document-level metric and applied to the document as a whole—not any specific vulnerability. The range of values in this field is defined according to the document producer's policies and procedures. These values can be understood only in the context of the document producer's stated practices. Therefore, the values may vary widely depending on the source of the document. The field is independent of—and in addition to—any other standard metric for determining the impact or severity of a given vulnerability (such as CVSS).

If one exists, the attribute *Namespace* should contain a URL pointing to the namespace so referenced.

Document References

Data Type	container
Range	--
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Root
Children	Reference

The **Document References** container should include references to any conferences, papers, advisories, and other resources that are related and considered to be of value to the document consumer. For every **Document References** container, there must be at least one **Reference** element, and each **Reference** element must contain one **URL** and one **Description**.

Reference

Data Type	container
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Document References
Children	URL, Description
Attribute	Type
Attribute Data Type	enumerated list
Attribute Required	yes
Attribute Default Value	External

Reference refers to resources related to the overall CVRF document. These may include a plaintext or HTML version of the advisory or other related documentation, such as white papers or mitigation documentation.

The *Type* attribute denotes the type of the document reference relative to the given document. The follow types are available:

- **External**: The default value indicates the reference is external to the document.
- **Self**: This indicates the related document is actually a direct reference to itself.

URL

Data Type	anyURI
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Reference

URL is the fixed URL or location of the reference.

Description

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Reference

Description is a descriptive title or the name of the reference.

Example:

```
<References>
    <Reference Type="External">
        <URL>http://foo.foo/bar/</URL>
        <Description xml:lang="fr">C'est un test de référence</Description>
    </Reference>
</References>
```

Acknowledgments

Data Type	container
Range	--
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Root
Children	Acknowledgment

The optional **Acknowledgments** container holds one or more **Acknowledgment** containers, which contain recognition of external parties.

Acknowledgment

Data Type	container
Range	--
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Acknowledgments
Children	Name, Organization, Description, URL

Acknowledgment contains recognition of external parties that reported noncritical/low-severity security issues or provided information, observations, or suggestions that contributed to improved security or improved documentation in future releases of the document producer's products. This may also contain recognition to external parties that contributed toward producing this document.

An acknowledgment container may contain three different types of child elements: **Name**, **Organization**, and/or a **Description**. All are described below.

Name

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Acknowledgment

The **Name** should contain the name of the party being acknowledged.

Organization

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Acknowledgment

The **Organization** should contain the organization of the party or if the **Name** is omitted, the organization itself that is being acknowledged.

Description

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Acknowledgment

The **Description** can contain any contextual details the document producers wish to make known about the acknowledgment or acknowledged parties.

If attributing to multiple organizations, each contributor should be grouped with that **Organization** within a single **Acknowledgment** container. An **Organization**-specific acknowledgment may be added within each **Acknowledgment** container using the **Description** element. If an overall general or aggregate acknowledgment is to be added, an **Acknowledgment** container that contains a single **Description** element may be used.

URL

Data Type	anyURI
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Acknowledgment

URL is the optional URL to the person, place, or thing being acknowledged.

Example:

```
<Acknowledgments>
    <Acknowledgment>
        <Name>
        Bartholomew McHandsome
        </Name>
        <Organization>
        FancyPants Inc.
        </Organization>
        <Description>
        We couldn’t have done it without you Bart!
        </Description>
        <URL>http://foo.foo/bar/</URL>
    </Acknowledgment>
</Acknowledgments>
```

PRODUCT TREE ELEMENTS

Product Tree

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Root
Children	Branch, Full Product Name, Relationship

The **Product Tree** container contains all the fully qualified product names that can be referenced elsewhere in the document (specifically when describing the products that are affected by a vulnerability using the **Product Statuses**, **Threats**, **CVSS Score Sets**, and **Remediation** containers). The **Product Tree** can have as many branches as needed, but each endpoint of the tree must be terminated with a **Full Product Name** element, which represents a product that can be referenced elsewhere.

This structure is a major change from CVRF 1.0, where affected products were direct child elements of **Vulnerability**; the change was necessary to meet the vastly different requirements of various organizations in the way they document their products. Also, in situations where a CVRF document contains more than one vulnerability, a separate product repository at the document level reduces the need to duplicate all product entries in each vulnerability.

The **Product Tree** can be kept simple (flat) or made more detailed (branched out). It also supports concatenating products to describe relationships, such as components contained in a product or products installed on other products.

Flat:
In the simplest case, a flat **Product Tree** would contain one or more **Full Product Name** elements at the root level, one for each product that needs to be described.

Branched:
In a more detailed **Product Tree**, the root element would contain one or more **Branch** elements at the root level, one for each class/type/category of product, each of which again contains one or more **Branch** elements until all desired categories and subcategories are described to the satisfaction of the document issuer. Then each open **Branch** element is terminated with the actual product item in the form of a **Full Product Name** element.

Concatenated:
No matter whether a flat or branched structure is chosen, you may need to be able to describe the combination of two **Full Product Name** elements, such as when a product is only vulnerable when installed together with another, or to describe operating system components. To do that, a **Relationship** element is inserted at the root of the **Product Tree**, with attributes establishing a link between two existing **Full Product Name** elements, allowing the document producer to define a combination of two products that form a new **Full Product Name** entry.

Grouped:
Once **Full Product Name** elements are defined, they may be freely added to logical groups, which may then be used to refer to a group of products. Given that it is possible for a product to be a member of more than one logical group, some areas of the CVRF document may not allow references to product groups to avoid ambiguity.

Branch

Data Type	choice
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Product Tree, Branch
Children	Branch, Full Product Name
Attribute	Name, Type
Attribute Data Type	string, enumerated list
Attribute Required	yes, yes

The **Branch** element is a choice element. A choice element behaves as a regular container that can have different child elements, but with the difference that only exactly one child element can be chosen. It is similar in concept to the "union" programming construct in which one variable can have one of several different predefined data types.

The **Branch** element contains a *Type-Name* pair as mandatory attributes to describe the characteristics of the current **Branch**. *Type* will describe the category of the branch in question. A full universe of values for 1.1 is shown below.

Branch *Types*:

- **Vendor**: The name of the vendor or manufacturer that makes the product
- **Product Family**: The product family that the product falls into
- **Product Name**: The name of the product
- **Product Version**: The product version, can be numeric or some other descriptor
- **Patch Level:** The patch level of the product
- **Service Pack**: The service pack of the product
- **Architecture**: The architecture for which the product is intended
- **Language**: The language of the product
- **Legacy**: A nonspecific legacy entry
- **Specification**: A specification such as a standard, best common practice, etc.

Name will contain the canonical descriptor or "friendly name" of the branch.

As for the child elements, each **Branch** can have either one of the following children:

- One **Full Product Name**. A single child element terminates the branch by describing a final product entry (described below).
- More **Branches**. Multiple additional **Branch** containers, which on their own can either terminate in a single **Full Product Name** element or yet more **Branch** containers.

Branch Example:

```
<Branch Type="Vendor" Name="Microsoft">
    <Branch Type="Product Family" Name="Windows">
        <Branch Type="Product Name" Name="Vista">
            <Branch Type="Service Pack" Name="1">
                <FullProductName ProductID="CVRFPID-0001">
                Microsoft Windows Vista Service Pack 1
                </FullProductName>
            </Branch>
            <Branch Type="Service Pack" Name="2">
                <FullProductName ProductID="CVRFPID-0002">
                Microsoft Windows Vista Service Pack 2
                </FullProductName>
            </Branch>
        </Branch>
    </Branch>
    <Branch Type="Product Family" Name="Office">
        <Branch Type="Product Name" Name="Word 2010">
            <Branch Type="Service Pack" Name="0">
                <Branch Type="Architecture" Name="x86">
                    <FullProductName ProductID="CVRFPID-0003">
                    Microsoft Word 2010 (32-bit editions)
                    </FullProductName>
                </Branch>
            </Branch>
        </Branch>
    </Branch>
</Branch>
```

Full Product Name

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Product Tree, Relationship, Branch
Attribute	Product ID, CPE
Attribute Data Type	token, CPE number or CPE URI
Attribute Required	yes, no

The **Full Product Name** elements define the endpoints of the **Product Tree** and occur directly at the root level, at the branch level, or as the result of a relationship between two products. The value of a **Full Product Name** element should be the product's full canonical name, including version number and other attributes, as it would be used in a human-friendly document.

Examples:

-
  ```
  <FullProductName ProductID="CVRFPID-0004">
  Microsoft Host Integration Server 2006 Service Pack 1
  </FullProductName>
  ```

-
  ```
  <FullProductName ProductID="CVRFPID-0005">
  Microsoft Office 2008 for Mac 12.3.1 Update
  </FullProductName>
  ```

The Common Platform Enumeration (*CPE*) attribute refers to a method for naming platforms. The structure for CPE is described at http://cpe.mitre.org. The *CPE* can be either an integer (if MITRE has an entry for the platform in question) or a candidate string from the vendor if no MITRE entry yet exists.

The *Product ID* attribute is required to identify a **Full Product Name** so that it can be referred to from other parts in the document. There is no predefined or required format for the *Product ID* as long as it uniquely identifies a product in the context of the current document. Examples include incremental integers or Globally Unique Identifiers (GUIDs).

Relationship

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Product Tree
Children	Full Product Name
Attribute	Product Reference, Relationship Type, Relates To Product Reference
Attribute Data Type	token, enumerated list, token
Attribute Required	yes, yes, yes

The **Relationship** element establishes a link between two existing **Full Product Name** elements, allowing the document producer to define a combination of two products that form a new **Full Product Name** entry.

This situation arises when a product is vulnerable only when installed together with another, or to describe operating system components. As a **Relationship** connects two existing products with each other, there need to be at least two **Full Product Name** entries present in the **Product Tree** before a Relationship element can be created.

Relationship elements live at the root of a **Product Tree**, and they have three mandatory attributes: *Product Reference* and *Relates To Product Reference* each contain the *Product ID* token for the two products that will form the relationship, and the *Type* attribute defines how the products are related.

Consider two previously constructed products with *Product IDs* CVRFPID-0001 and CVRFPID-0002. CVRF v1.1 supports the following **Relationship** *Type* values:

- **Default Component Of**: CVRFPID-0001 is a default component of CVRFPID-0002
- **Optional Component Of**: CVRFPID-0001 is an optional component of CVRFPID-0002
- **External Component Of**: CVRFPID-0001 is an external component of CVRFPID-0002
- **Installed On**: CVRFPID-0001 is installed on CVRFPID-0002
- **Installed With**: CVRFPID-0001 is installed with CVRFPID-0002

Once a **Relationship** element has been created, it needs to be completed by adding one **Full Product Name** element as a child, typically using a combination of the two related product names as a value.

Examples:

The first product is defined as:

```
<FullProductName ProductID="CVRFPID-0007">

Active Directory Lightweight Directory Service

</FullProductName>
```

And the second product is defined as:

```
<FullProductName ProductID="CVRFPID-0008">

Windows Vista Service Pack 2

</FullProductName>
```

And the relationship can then be defined as:

```
<Relationship ProductReference="CVRFPID-0007"

              RelationType="OptionalComponentOf"

              RelatesToProductReference = "CVRFPID-0008">

    <FullProductName ProductID=”CVRFPID-0009>

    Active Directory Lightweight Directory Service as an optional component of
    Windows Vista Service Pack 2

    </FullProductName>

</Relationship>
```

In another example, the first product is defined as:

```
<FullProductName ProductID="CVRFPID-0010">

Cisco AnyConnect Secure Mobility Client 2.3.185

</FullProductName>
```

And the second product is defined as:

```
<FullProductName ProductID="CVRFPID-0011">

Microsoft Windows

</FullProductName>
```

And the relationship can then be defined as:

```
<Relationship ProductReference="CVRFPID-0010" RelationType="InstalledOn"

              RelatesToProductReference="CVRFPID-0011">

    <FullProductName ProductID=”CVRFPID-0012>

    Cisco AnyConnect Secure Mobility Client 2.3.185 when installed on Microsoft
    Windows

    </FullProductName>

</Relationship>
```

Product Groups

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Product Tree
Children	Group

The **Product Groups** container defines whether **Full Product Name** elements in the product tree will be grouped into logical groups. If the container is present, at least one Group must be defined.

If groups are defined, products can be referred to using the Group ID attribute in many other parts of the document, rather than repeatedly having to list all members individually.

Whether groups are defined or not, the ability to reference each product individually in other parts of the document is not affected. In fact, the creator of a document can choose to use either direct product references or group references.

Note:

Given that a single product can be a member of more than one group, some areas of the CVRF document may not allow product references by group to avoid ambiguity.

Example:

We create two groups, CVRFGID-0001 and CVRFGID-0002. Both groups have four members, and ProductID CVRFPID-0001 is a member of both groups:

```
<ProductGroups>

    <Group GroupID="CVRFGID-0001">

        <ProductID>CVRFPID-0001</ProductID>

        <ProductID>CVRFPID-0002</ProductID>

        <ProductID>CVRFPID-0003</ProductID>

        <ProductID>CVRFPID-0004</ProductID>

    </Group>

    <Group GroupID="CVRFGID-0002">

        <ProductID>CVRFPID-0001</ProductID>

        <ProductID>CVRFPID-0010</ProductID>

        <ProductID>CVRFPID-0011</ProductID>

        <ProductID>CVRFPID-0099</ProductID>

    </Group>

</ProductGroups>
```

Group

Data Type	container
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Product Groups
Children	Description, Product ID
Attribute	Group ID
Attribute Data Type	token
Attribute Required	yes

Each **Group** container defines a new logical group of products that can then be referred to in other parts of the document to address a group of products with a single identifier. **Group** members are defined by adding one **Product ID** element for each member of the group.

The *Group ID* attribute is required to identify a **Group** so that it can be referred to from other parts in the document. There is no predefined or required format for the *Group ID* as long as it uniquely identifies a group in the context of the current document. Examples include incremental integers or GUIDs.

Description

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Group

Description is a short, optional description of the group.

Example:

```
<ProductGroups>
    <Group GroupID="CVRFGID-0001">
        <Description>The x64 versions of the operating system.</Description>
        <ProductID>CVRFPID-0001</ProductID>
        <ProductID>CVRFPID-0002</ProductID>
        <ProductID>CVRFPID-0003</ProductID>
        <ProductID>CVRFPID-0004</ProductID>
    </Group>
</ProductGroups>
```

Product ID

Data Type	token
Minimum Occurrences	2
Maximum Occurrences	unbounded
Parent	Group

The **Product ID** element defines a member of a group by referring to the unique *Product ID* attribute of a **Full Product Name** element.

Examples:

If the two products “Microsoft Windows Vista Service Pack 1” and “Microsoft Windows Vista Service Pack 2” have been defined in the product tree as follows:

```
<FullProductName ProductID="CVRFPID-0001">
Microsoft Windows Vista Service Pack 1
</FullProductName>
<FullProductName ProductID="CVRFPID-0002">
Microsoft Windows Vista Service Pack 2
</FullProductName>
```

They can both be made a member of the same group with Group ID “GRP-0001”:

```
<ProductGroups>
    <Group GroupID="GRP-0001">
        <ProductID>CVRFPID-0001</ProductID>
        <ProductID>CVRFPID-0002</ProductID>
    </Group>
```

```
</ProductGroups>
```

Later in the document, both products can be referenced together using the Group ID:

```
<Remediations>
    <Remediation Type="Vendor Fix">
        <Description>Security Update for Windows Vista</Description>
        <GroupID>GRP-0001</GroupID>
    </Remediation>
</Remediations>
```

The ability to reference both products individually will also be maintained (and in some cases required):

```
<Remediations>
    <Remediation Type="Vendor Fix">
        <Description>Security Update for Windows Vista</Description>
        <ProductID>CVRFPID-0001</ProductID>
        <ProductID>CVRFPID-0002</ProductID>
    </Remediation>
</Remediations>
```

VULNERABILITY ELEMENTS

Vulnerability

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Root
Children	Title, ID, Involvements, Notes, Discovery Date, Release Date, CVE, CWE, Threats, CVSS Score Sets, Remediation, Product Statuses, Acknowledgments, References
Attribute	Ordinal
Attribute Data Type	positiveInteger
Attribute Required	yes
Attribute Default Value	1

Vulnerability is a container for the aggregation of all fields that are related to a single vulnerability in the document. There may be zero, one, or many vulnerabilities in a single CVRF document.

Ordinal is a locally significant value used to track vulnerabilities inside a CVRF document. It is provided to enable specific vulnerabilities to be referenced from elsewhere in the document (or even outside the namespace of a document provided that a unique **Document Title** and **Revision** information are provided). There should be one of these values for every **Vulnerability** container in a document, and it is recommended that *Ordinal* should be instantiated as a monotonically increasing counter, indexed from 1.

Example:

```
<Vulnerability Ordinal="1" xmlns="http://www.icasi.org/CVRF/schema/vuln/1.1">

...

</Vulnerability>
```

Title

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability

Title gives the document producer the ability to apply a canonical name or title to the vulnerability. To avoid confusion, it is recommended that, if employed, this element commensurately match the nomenclature used by any numbering or cataloging systems references elsewhere, such as the **Document Title** or **CVE**.

Example:

```
<Title>February 2011 TelePresence Vulnerability Bundle</Title>
```

ID

Data Type	token
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability
Attribute	System Name
Attribute Data Type	string
Attribute Required	yes

ID gives the document producer a place to publish a unique label or tracking ID for the vulnerability (if such information exists).

General examples may include an identifier from a vulnerability tracking system that is available to customers, such as a Cisco bug ID, an ID from a Bugzilla system, or an ID from a public vulnerability database such as the X-Force Database. The **ID** may be a vendor-specific value.

The **ID** should not be used for CVE tracking numbers (MITRE standard Common Vulnerabilities and Exposures). CVE numbers should be specified using the separate CVE element. Values are tokenized and can be alphanumeric.

The attribute *System Name* indicates the name of the vulnerability tracking or numbering system that this **ID** comes from. Every **ID** value should have exactly one *System Name*. It is helpful if document producers use unique and consistent system names.

Example:

```
<VulnerabilityID>

    <Value SystemName="Cisco Bug ID">CSCso66472</Value>

</VulnerabilityID>
```

Notes

Data Type	container
Range	--
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability

Notes contains all the vulnerability-level **Note** elements.

Note

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Vulnerability
Attribute	Type, Ordinal, Title, Audience
Attribute Data Type	enumerated list, positiveInteger, string, string
Attribute Required	yes, yes, no, no

Notes is a place to put all manner of text blobs related to the vulnerability. Text should be limited to talking about the impacts, vectors, or caveats of this node and should not contain details to other vulnerabilities in the document. It is, however, acceptable to refer to a vulnerability that is not in the document for the purposes of pointing out a regression.

Akin to the **Document Notes** element, the note should contain a compartmentalized textual discussion constrained by its *Type* attribute. *Type* can be one of the following:

- **General:** A general, high-level note (*Title* may have more information).
- **Details:** A low-level detailed discussion (*Title* may have more information).
- **Description:** A description of something (*Title* may have more information).
- **Summary**: A summary of something (*Title* may have more information).
- **FAQ**: A list of frequently asked questions.
- **Legal Disclaimer**: Any possible legal discussion, including constraints, surrounding the vulnerability.
- **Other**: Something that doesn't fit (*Title* should have more information).

Title and *Audience* are optional attributes to give human readers context around what they are about to read; *Title* should be a concise description of what is contained in the text, whereas *Audience* will indicate who is intended to read it.

Ordinal is a mandatory, locally significant value used to track notes inside a CVRF document at the vulnerability level. It is provided to uniquely identify a **Note**. There should be one of these values for every **Note** inside **Vulnerability/Notes** and it is recommended that *Ordinal* should be instantiated as a monotonically increasing counter, indexed from 1. Each *Ordinal* that tracks a **Note** inside **Vulnerability/Notes** is completely independent from an *Ordinal* tracking a **Note** inside **Document Notes**.

Discovery Date

Data Type	dateTime
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability

The **Discovery Date** is the date the vulnerability was originally discovered. All dateTime values in CVRF require a time, and we recommend the inclusion of a time zone as well (ICASI endorses the use of GMT or "Zulu time"). If a time zone is excluded, Zulu should be assumed.

Release Date

Data Type	dateTime
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability

The **Release Date** is the date the vulnerability was originally released into the wild. All dateTime values in CVRF require a time, and we recommend the inclusion of a time zone as well (ICASI endorses the use of GMT or "Zulu time"). If a time zone is excluded, Zulu should be assumed.

Involvements

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability
Children	Involvement

The optional **Involvements** container holds one or more **Involvement** containers, which allow the document producers (or third party) to comment on their level of involvement in the vulnerability identification, scoping, and remediation process. Because there can be multiple Involvements containers, multiple parties can comment on their levels of involvement.

Involvement

Data Type	container
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Involvements
Children	Description
Attribute	Party, Status
Attribute Data Type	enumerated list, enumerated list
Attribute Range	{Vendor, Discoverer, Coordinator, User, Other}, {Open, Disputed, In Progress, Completed, Contact Attempted, Not Contacted}
Attribute Required	yes, yes

The **Involvement** container allows the document producers to comment on their level of Involvement (or engagement) in the vulnerability identification, scoping, and remediation process.

The attribute *Party* indicates the type of the producer issuing the status. It is identical to the **Document Publisher** attribute *Type*. Most of the time, both attributes will be the same because document producers will issue an **Involvement** status on their own behalf. However, if the document producer wants to issue a status on behalf of a third party and use a different type from that used in **Document Publisher**, that use is allowed by the schema. If this is the case, **Description** should contain additional context regarding what is going on.

The attribute *Status* indicates the level of involvement of *Party*.

The child **Description** (below) is an optional element used to give context about the involvement or engagement of the *Party*.

The final two status states, "Contact Attempted" and "Not Contacted," are intended for use by document producers other than vendors (such as research or coordinating entities).

Status types include:

- **Open:** This is the default status. It doesn't indicate anything about the vulnerability remediation effort other than the fact that the vendor has acknowledged awareness of the vulnerability report. The use of this status by a vendor indicates that future updates from the vendor about the vulnerability are to be expected.
- **Disputed:** This status indicates that the vendor disputes the vulnerability report in its entirety. Vendors should indicate this status when they believe that a vulnerability report regarding their product is completely inaccurate (that there is no real underlying security vulnerability) or that the technical issue being reported has no security implications.
- **In Progress:** This status indicates that some hotfixes, permanent fixes, mitigations, workarounds, or patches may have been made available by the vendor, but more information or fixes may be released in the future. The use of this status by a vendor indicates that future information from the vendor about the vulnerability is to be expected.
- **Completed:** The vendor asserts that investigation of the vulnerability is complete. No additional information, fixes, or documentation from the vendor about the vulnerability should be expected to be released.
- **Contact Attempted:** The document producer attempted to contact the affected vendor.
- **Not Contacted:** The document producer has not attempted to make contact with the affected vendor.

Each status is mutually exclusive—only one status is valid for a particular vulnerability at a particular time. As the vulnerability ages, a party's involvement could move from state to state. However, in many cases, a document producer may choose not to issue CVRF documents at each state, or simply omit this element altogether. It is recommended, however, that vendors that issue CVRF documents indicating an open or in-progress **Involvement** should eventually expect to issue a document as Disputed or Completed.

Description

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Involvement

The **Description** element will contain a thorough human-readable discussion of the **Involvement**.

Examples include:

```
<Involvements>
    <Involvement Party=”Vendor” Status=”In Progress”>
        <Description>
        Cisco acknowledges that the IronPort Email Security Appliances (ESA)
        and Cisco IronPort Security Management Appliances (SMA) contain a
        vulnerability that may allow a remote, unauthenticated attacker to
        execute arbitrary code with elevated privileges. A Mitigation is
        available.
        </Description>
    </Involvement>
</Involvements>

<Involvements>
    <Involvement Party=”Researcher” Status=”Contact Attempted”>
        <Description>
        We emailed the vendor on February 14, 2012 when the vulnerability was
        first discovered by our team.
        </Description>
    </Involvement>
</Involvements>
```

CVE

Data Type	token
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability

CVE contains the MITRE standard Common Vulnerabilities and Exposures (CVE) tracking number for the vulnerability. CVE is a standard for vulnerability naming that provides improved tracking of vulnerabilities over time across different reporting sources. More information about CVE is available at http://cve.mitre.org/.

Example:

```
<CVE>CVE-2006-0010</CVE>
```

CWE

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Vulnerability
Attribute	ID
Attribute Data Type	token
Attribute Required	yes

CWE contains the MITRE standard Common Weakness Enumeration (CWE). MITRE describes CWE in this way: “[CWE] is a formal list of software weakness types created to:

- Serve as a common language for describing software security weaknesses in architecture, design, or code.
- Serve as a standard measuring stick for software security tools targeting these weaknesses.
- Provide a common baseline standard for weakness identification, mitigation, and prevention efforts.”

More information about CWE is available at http://cwe.mitre.org/.

Examples:

-
  ```
  <CWE ID=”CWE-601”>
  URL Redirection to Untrusted Site ('Open Redirect')
  </CWE>
  ```

-
  ```
  <CWE ID=”CWE-602”>
  Client-Side Enforcement of Server-Side Security
  </CWE>
  ```

Product Statuses

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability
Children	Status

The optional **Product Statuses** container holds one or more **Status** containers, which will contain a subset of products chosen from **Product Tree** (see below). Each of the affected (and unaffected) products relating to the vulnerability will be referenced here, inside one or more **Status** containers.

Note there is a constraint in place to prevent a single product from being assigned two different (conflicting) **Status** elements within the scope of **Vulnerability**. Likewise, a **Status** child container cannot be tied to a **Product Group** due to the fact that a single product can be a member of more than one product group. Without this constraint, it would be possible to assign conflicting status information to one and the same product.

Status

Data Type	container
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Product Statuses
Children	Product ID
Attribute	Type
Attribute Data Type	enumerated list
Attribute Required	yes

The **Status** element contains one or more products as chosen from the **Product Tree**, and defines the status of this product in the mandatory *Status* attribute.

The *Type* attribute is an enumerated value that contains all the possible permutations of fixed, affected, and recommended versions of the products referenced inside the **Status** container. *Type* values include:

- **First Affected:** This is first version of the affected release known to be affected by the vulnerability.
- **Known Affected:** This version is known to be affected by the vulnerability.
- **Known Not Affected:** This version is known not to be affected by the vulnerability.
- **First Fixed:** This version contains the first fix for the vulnerability but may not be the recommended fixed version.
- **Fixed:** This version contains a fix for the vulnerability but may not be the recommended fixed version.
- **Recommended:** This version has a fix for the vulnerability and is the vendor-recommended version for fixing the vulnerability.
- **Last Affected:** This is the last version in a release train known to be affected by the vulnerability. Subsequently released versions would contain a fix for the vulnerability.

Example:

The three products “Microsoft Windows Vista (RTM),” “Microsoft Windows Vista Service Pack 1,” and “Microsoft Windows Vista Service Pack 2” have been defined in the product tree as follows:

```
<ProductTree>
    <FullProductName ProductID="CVRFPID-0000">
    Microsoft Windows Vista (RTM)
    </FullProductName>
    <FullProductName ProductID="CVRFPID-0001">
    Microsoft Windows Vista Service Pack 1
    </FullProductName>
    <FullProductName ProductID="CVRFPID-0002">
    Microsoft Windows Vista Service Pack 2
    </FullProductName>
</ProductTree>
```

If Windows Vista RTM and Service Pack 1 are known to be affected, and Service Pack 2 is known not to be affected, it can be documented as follows:

```
<Vulnerability Ordinal="1">
    <Product Statuses>
        <Status Type="KnownAffected">
            <ProductID>CVRFPID-0000</ProductID>
            <ProductID>CVRFPID-0001</ProductID>
        </Status>
        <Status Type="KnownNotAffected">
            <ProductID>CVRFPID-0002</ProductID>
        </Status>
    </Product Statuses>
</Vulnerability>
```

Product ID

Data Type	token
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Product

The **Product ID** element defines a product as having the status defined in the parent element's *Type* attribute. The reference is made using the unique *Product ID* attribute of a **Full Product Name** element that is defined in the **Product Tree**.

Note that a single **Product ID** may not be assigned more than one status type within the same **Vulnerability**.

Threats

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability
Children	Threats

The optional **Threats** container holds one or more **Threat** containers, which contain information about a vulnerability that can change with time (so called "vulnerability kinetics").

Threat

Data Type	container
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Vulnerability
Children	Description, Product ID, Group ID
Attribute	Type, Date
Attribute Data Type	enumerated list, dateTime
Attribute Required	yes, no

Threat contains the vulnerability kinetic information. This information can change as the vulnerability ages and new information becomes available. A given **Threats** container can contain one or more **Threat**.

A **Threat** container can be tied to one or more specific products by referencing these products using either the **Product ID** or **Group ID** child elements. If the **Threat** is meant to be general or nonspecific for all products, the **Product ID** and **Group ID** child elements should be omitted.

The *Date* attribute is optional. All dateTime values in CVRF require a time, and we recommend the inclusion of a time zone as well (ICASI endorses the use of GMT or “Zulu time”). If a time zone is excluded, Zulu should be assumed.

The *Type* of Threat is required and can be one of the following:

Impact: Impact contains an assessment of the impact on the user or the target set if the vulnerability is successfully exploited. (A description of the **Target Set** *Type* follows.) If applicable, for consistency and simplicity, this section can be a textual summary of the three CVSS impact metrics. These metrics measure how a vulnerability detracts from the three core security properties of an information system: Confidentiality, Integrity, and Availability.
Exploit Status: Exploit Status contains a description of the degree to which an exploit for the vulnerability is known. This knowledge can range from information privately held among a very small group to an issue that has been described to the public at a major conference or is being widely exploited globally. For consistency and simplicity, this section can be a mirror image of the CVSS “Exploitability” metric. However, it can also contain a more contextual status, such as “Weaponized” or “Functioning Code.”

Target Set: Target Set contains a description of the currently known victim population in whatever terms are appropriate. Such terms may include: operating system platform, types of products, user segments, and geographic distribution.

Description

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Threat

The **Description** element will contain a thorough human-readable discussion of the **Threat**.

Impact Example:

-
  ```
  <Threat Type=”Impact”>
      <Description>
      complete compromise of the integrity of affected machines
      </Description>
  ```

```
    </Threat>
```

Exploit Status examples:

-
  ```
  <Threat Type=”Exploit Status”>
      <Description>
       none
       </Description>
       <Date>2011-11-26T00:00:00+00:00</Date>
       <ProductID>CVRFPID-0000</ProductID>
  </Threat>
  ```

-
  ```
  <Threat Type=”Exploit Status”>
      <Description>
       proof of concept
       </Description>
       </Date>2011-11-26T00:00:00+00:00</Date>
  </Threat>
  ```

Target Set Examples:

-
  ```
  <Threat Type=”Target Set”>
      <Description>
       Financial Institutions
       </Description>
  </Threat>
  ```

-
  ```
  <Threat Type=”Target Set”>
      <Description>
       US Government Agencies
       </Description>
  </Threat>
  ```

-
  ```
  <Threat Type=”Target Set”>
         <Description>
         All versions of BIND 9.4.0 and lower
         </Description>
  </Threat>
  ```

Product ID

Data Type	token
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Threat

If the **Threat** pertains to a specific product, a **Product ID** element can be added to reference that product. The reference is made using the unique *Product ID* attribute of a **Full Product Name** element that is defined in the **Product Tree**. If a **Threat** applies to more than one

Product, you can add multiple **Product ID** elements accordingly, or add the **Product ID** element (see below) instead.

Group ID

Data Type	token
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Threat

If the **Threat** pertains to several products that have been logically grouped into a **Product Group**, the **Group ID** element can be added to reference that group of products. The reference is made using the unique *Group ID* attribute of a **Group** element that is defined in the **Product Tree**. If a **Threat** applies to more than one group of products, you can add multiple **Group ID** elements accordingly.

CVSS Score Sets

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability
Children	Score Set

The optional **CVSS Score Sets** container holds one or more of the **Score Set** containers.

Score Set

Data Type	container
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Vulnerability
Children	Base Score, Temporal Score, Environmental Score, Vector, Product ID

The **Score Set** container holds actual CVSS metrics. For more details about CVSS, see http://www.first.org/cvss/cvss-guide.html. The only required element of CVSS is the **Base Score**. If a value of the temporal or environmental score is set to "not defined," either **Temporal Score** or **Environmental Score** can be omitted.

A **Score Set** container can be tied to one or more specific products by referencing these products using the **Product ID** child element. If the **Score Set** is meant to be applied for all products, the *Product ID* attribute should be omitted.

Note there is a constraint in place to prevent having a single product assigned to two different score sets within the scope of a **Vulnerability**. Likewise, a **Score Set** cannot be tied to a **Product Group** due to the fact that a single product can be a member of more than one product group. Without this constraint, it would be possible to assign conflicting base score information to one and the same product.

Base Score

Data Type	float
Range	0.0 - 10.0
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Score Set

Base Score contains the numeric value of the computed CVSS base score, which should be a float from 0 to 10.0.

Temporal Score

Data Type	float
Range	0.0 - 10.0
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Score Set

Temporal Score contains the numeric value of the computed CVSS temporal score, which should be a float from 0 to 10.0.

Environmental Score

Data Type	float
Range	0.0 - 10.0
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Score Set

Environmental Score contains the numeric value of the computed CVSS environmental score, which should be a float from 0 to 10.0. This metric is typically reserved for use by the end user and is specific to the environment in which the affected product is deployed.

Vector

Data Type	string
Range	76 characters
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Score Set

Vector contains the official notation that displays all the values used to compute the CVSS base, temporal, and environmental scores. This notation will follow the guidelines set forth in the CVSS v2 documentation at http://www.first.org/cvss/cvss-guide.html#i2.4. Note the 76-character limitation.

CVSS Vector Example:

```
<Vector>

AV:N/AC:L/Au:N/C:P/I:P/A:C/E:P/RL:O/RC:C/CDP:H/TD:M/CR:H/IR:H/AR:H

<Vector>
```

Product ID

Data Type	token
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Score Set

If the **Score Set** pertains to a specific product, a **Product ID** element can be added to reference that product. The reference is made using the unique *Product ID* attribute of a **Full Product Name** element that is defined in the **Product Tree**. If a **Score Set** applies to more than one product, you can add multiple **Product ID** elements accordingly.

Note that a single **Product ID** may not be assigned to more than one **Score Set** within the same **Vulnerability**.

Remediations

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Vulnerability
Children	Remediation

The optional **Remediations** container holds one or more **Remediation** containers, which will have details on how to remediate a vulnerability.

Remediation

Data Type	container
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Remediations
Children	Description, Entitlement, URL, Product ID, Group ID
Attribute	Type
Attribute Data Type	enumerated list
Attribute Range	{Workaround, Mitigation, Vendor Fix, None Available, Will Not Fix}
Attribute Required	yes

The **Remediation** container holds specific details on how to handle (and presumably, fix) a vulnerability. A given **Remediations** container can contain one or more **Remediation** containers.

A **Remediation** container can be tied to one or more specific products by referencing these products using either the **Product ID** or **Group ID** child elements. If the **Remediation** is meant to be general or nonspecific for all products, the **Product ID** and **Group ID** child elements should be omitted.

The *Type* attribute is required and can be one of the following:

- **Workaround:** Workaround contains information about a configuration or specific deployment scenario that can be used to avoid exposure to the vulnerability. There may be none, one, or more workarounds available. This is typically the “first line of defense” against a new vulnerability before a mitigation or vendor fix has been issued or even discovered.

- **Mitigation:** Mitigation contains information about a configuration or deployment scenario that helps to reduce the risk of the vulnerability but that does not resolve the vulnerability on the affected product. Mitigations may include using devices or access controls external to the affected product. Mitigations may or may not be issued by the original author of the affected product, and they may or may not be officially sanctioned by the document producer.
- **Vendor Fix:** Vendor Fix contains information about an official fix that is issued by the original author of the affected product. Unless otherwise noted, it is assumed that this fix fully resolves the vulnerability.
- **None Available:** Currently there is no fix available. Description should contain details about why there is no fix.
- **Will Not Fix:** There is no fix for the vulnerability and there never will be one. This is often the case when a product has been orphaned, end-of-lifed, or otherwise deprecated. Description should contain details about why there will be no fix issued.

Optionally, **Remediation** can contain information and constraints about how to obtain fixes via the **Entitlement** element.

Description

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Remediation

The **Description** element will contain a thorough human-readable discussion of the Remediation.

Entitlement

Data Type	string
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Remediation

Entitlement contains any possible vendor-defined constraints for obtaining fixed software or hardware that fully resolves the vulnerability. This element will often contain information about service contracts or service-level agreements that is directed toward customers of large vendors.

Example:

```
<Entitlement>

Cisco customers with service contracts that entitle them to regular software
updates should obtain security fixes through their usual update channels,
generally from the Cisco website. Cisco recommends contacting the TAC only with
specific and imminent problems or questions.\r\nAs a special customer service,
and to improve the overall security of the Internet, Cisco may offer customers
free of charge software updates to address security problems. If Cisco has
offered a free software update to address a specific issue, noncontract
customers who are eligible for the update may obtain it by contacting the Cisco
TAC using any of the means described in the Contact Summary section of this
```

```
document. To verify their entitlement, individuals who contact the TAC should
have available the URL of the Cisco document that is offering the
upgrade.\r\nAll aspects of this process are subject to change without notice
and on a case-by-case basis. No particular level of response is guaranteed for
any specific issue or class of issues.

</Entitlement>
```

URL

Data Type	anyURI
Range	unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Remediation

URL is the optional URL to the Remediation.

Example:

```
<Remediation Type="Vendor Fix">

    <Description>

    this is an official fix for Test Product and here are the details...

    </Description>

    <URL>http://foo.foo/bar/</URL>

    <Product ID>CVRFPID-0000</Product ID>

</Remediation>
```

Product ID

Data Type	token
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Remediation

If the **Remediation** pertains to a specific product, a **Product ID** element can be added to reference that product. The reference is made using the unique *Product ID* attribute of a **Full Product Name** element that is defined in the **Product Tree**. If a **Remediation** applies to more than one Product, you can add multiple **Product ID** elements accordingly, or add the **Group ID** element (see below) instead.

Group ID

Data Type	token
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Remediation

If the **Remediation** pertains to several products that have been logically grouped into a **Product Group**, the **Group ID** element can be added to reference that group of products. The reference is made using the unique *Group ID* attribute of a **Group** element that is defined in the **Product Tree**. If a **Remediation** applies to more than one group of products, you can add multiple **Group ID** elements accordingly.

References

Data Type	container
Minimum Occurrences	0
Maximum Occurrences	unbounded
Parent	Vulnerability
Children	Reference

The **References** container should include citations to any conferences, papers, advisories, and other resources that are specific to the vulnerability section and considered to be of value to the document consumer. For every **References** container, there must be at least one **Reference** element and each **Reference** element must contain one **URL** and one **Description**.

Reference

Data Type	container
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	References
Children	URL, Description
Attribute	Type
Attribute Data Type	enumerated list
Attribute Required	yes
Attribute Default Value	External

The **Reference** element contains a description of a related document specific to a vulnerability section of a CVRF document. This may include a plaintext or HTML version of the advisory or other related documentation, such as white papers or mitigation documentation.

The *Type* attribute denotes the type of the document reference relative to the CVRF document itself. The following types are available:

- External: The default value indicates the reference is external to the CVRF document.
- Self: This indicates the related document is actually a direct reference to itself.

URL

Data Type	anyURI
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Reference

URL is the fixed URL or location of the reference.

Description

Data Type	string
Range	unrestricted
Minimum Occurrences	1
Maximum Occurrences	1
Parent	Reference

Description is a descriptive title or name of the reference.

Acknowledgments

Data Type	container
Range	--
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Root
Children	Acknowledgment

The optional **Acknowledgments** container holds one or more **Acknowledgment** containers, which contain recognition of external parties. This **Acknowledgments** container is different from the one at the document level because it is specifically related to the vulnerability in question.

Acknowledgment

Data Type	container
Range	--
Minimum Occurrences	1
Maximum Occurrences	unbounded
Parent	Acknowledgments
Children	Name, Organization, Description, URL

Acknowledgment contains recognition of external parties who were instrumental in the discovery of, reporting of, and response to the vulnerability. This element indicates collaboration with the security community in a positive fashion and is an important part of a notice or advisory. Care should be taken to ensure that individuals would like to be acknowledged before they are included.

External parties who have worked with the document producer may be recognized for their work. This should be applied liberally; if someone reports an issue and then discloses it publicly, that party might still be credited.

If the original discoverer is not concerned with recognition, or the issue was discovered internally by the document producer, this field can be omitted.

An acknowledgment container may contain three different types of child elements: **Name**, **Organization**, and/or a **Description**. All are described below.

Name

Data Type	String
Range	Unrestricted
Minimum Occurrences	0
Maximum Occurrences	Unbounded
Parent	Acknowledgment

The **Name** should contain the name of the party being acknowledged.

Organization

Data Type	String
Range	Unrestricted
Minimum Occurrences	0
Maximum Occurrences	Unbounded
Parent	Acknowledgment

The **Organization** should contain the organization of the party or, if the **Name** is omitted, the organization itself that is being acknowledged.

Description

Data Type	String
Range	Unrestricted
Minimum Occurrences	0
Maximum Occurrences	1
Parent	Acknowledgment

The **Description** can contain any contextual details the document producers wish to make known about the acknowledgment or acknowledged parties.

If attributing to multiple organizations, each contributor should be grouped with that **Organization** within a single **Acknowledgment** container. An **Organization**-specific acknowledgment may be added within each **Acknowledgment** container by the **Description** element. If an overall general or aggregate acknowledgment is to be added, an **Acknowledgment** container that contains a single **Description** element may be used.

Example:

```
<Acknowledgments>
    <Acknowledgment>
        <Name>[Name 1]</Name>
        <Name>[Name 2]</Name>
        <Organization>[OrgName]</Organization>
        <URL>http://foo.foo/bar/</URL>
    </Acknowledgment>
    <Acknowledgment>
        <Name>[Name 3]</Name>
        <Organization>[OrgName]</Organization>
        <Description>
        Vendor X would like to thank [Name 3] from [OrgName] for reporting this
        issue.
        </Description>
        <URL>http://foo.foo/bar/</URL>
    </Acknowledgment>
    <Acknowledgment>
        <Description>
        Vendor  X would like to thank the following researchers for their
        contributions to making this project more secure:  [Name 1], [Name 2],
        [Name 3]
        </Description>
        <URL>http://foo.foo/bar/</URL>
    </Acknowledgment>
</Acknowledgments>
```

END OF FILE.