The Common Vulnerability Reporting Framework

An Internet Consortium for Advancement of Security on the Internet (ICASI) Whitepaper

Version 1.0
2011-05-01T10:20:00-08:00
Mike Schiffman, Cisco Systems, Inc., mschiffm@cisco.com

To date, a major gap exists in vulnerability standardization: there is no standard framework for the creation of vulnerability report documentation. Although the computer security community has made significant progress in several other areas, including categorizing and ranking the severity of vulnerabilities in information systems with the widespread adoption of the Common Vulnerabilities and Exposures (CVE) [1] dictionary and the Common Vulnerability Scoring System (CVSS) [2], this lack of standardization is evident in every vulnerability report, best practice document, or security bulletin released by any vendor or coordinator. In this white paper, a common and consistent framework is proposed for exchanging not just vulnerability information, but any security-related documentation. Originally derived from the Internet Engineering Task Force (IETF) draft Incident Object Description Exchange Format (IODEF) [3], The Common Vulnerability Reporting Framework (CVRF) is an XML-based language that will enable different stakeholders across different organizations to share critical security-related information in a single format, speeding up information exchange and digestion.

1. Introduction and Impetus

In recent years, significant progress has been made in information system vulnerability categorization and severity ranking. These advancements have tangibly benefited security operations such as cross-domain security analysis, first response planning, and patch management. The missing link is a single common framework for encoding security information, and the most obvious benefactor is the creation of vulnerability reports and similar documents to distribute across different organizations.

1.1 Problem Statement

Conventionally, the documentation of vulnerabilities is an ad hoc, producer-specific, and overtly nonstandard process. Document producers compile, collate, and produce their own versions of a vulnerability document that may or may not be similar to comparable reports by other vendors. To see an example of this, consider the 2009 multivendor "Outpost24 TCP" vulnerability report from major document producers such as Cisco [4], Microsoft [5], CERT [6], and Secunia [7]. Because each producer employs a unique and non-cooperative document structure, document consumers must manually parse individual reports to find information that is germane to their environments. Additionally, the documents are typically flat and do not facilitate or support automated processing.

1.2 Vulnerability Report Differences

For a solid example of this dissimilarity, see tables 1.1 through 1.4. Four vulnerability reports that detail the previously referenced Outpost24 TCP vulnerability are procured from the Internet. The reports are stripped bare of content and format and reduced to their core constituent components. Only a field title and a corresponding data type remain.

Field Title	Data Type
Summary	Text blob
Affected Products	Container
Vulnerable Products	List of text blobs
Products Confirmed Not Vulnerable	Bulleted list
Details	Text blob
Vulnerability Scoring Details	Text blob
Impact	Text blob
Software Versions and Fixes	Table
Workarounds	Text blob
Obtaining Fixed Software	Text blob
Exploitation and Public Announcements	Text blob
Status of this Notice	Text blob
Distribution	Text blob
Revision History	Table
Cisco Security Procedures	Text blob

Table 1.1 Cisco Systems Vulnerability Report

Field Title	Data Type
General Information	Container
Executive Summary	Text blob
Affected and Non-Affected Software	Container
Affected Software	Table
Non-Affected Software	Table

FAQ	Text blob
Vulnerability Information	Container
Severity Ratings and Vulnerability Identifiers	Table
0 or more vulnerabilities sorted by CVE	Container
Vulnerability Description	Text blob
Update Information	Container
Detection and Deployment Tools Guidance	Text blob
Security Update Deployment	Text blob
Other Information	Container
Acknowledgements	Text blob
Microsoft Active Protections Program	Text blob
Support	Text blob
Disclaimer	Text blob
Revisions	Bulleted list

Table 1.2 Microsoft Vulnerability Report

Field Title	Data Type
Target	Bulleted list
Access Vector	Bulleted list
Impact	Bulleted list
Remediation	Bulleted list
Details	Text blob
Impact	Text blob
Severity	Text blob
Vulnerability Coordination Information	Text blob
Vendor Information	Bulleted list
Remediation	Text blob
References	Bulleted list
Contact Information	Text blob
Revision History	Bulleted list

Table 1.3 CERT-FI Vulnerability Report

Field Title	Data Type
Secunia Advisory	String
Release Date	Date
Last Update	Date
Popularity	Integer
Comments	Text blob
Criticality Level	Enum
Impact	Enum
Where	Enum
Authentication Level	Text blob
Report Reliability	Text blob
Solution Status	Text blob
Systems Affected	Text blob
Approve Distribution	Text blob
Automated Scanning	Text blob
Operating System	Bulleted list
Secunia CVSS Score	Text blob

CVE References	Bulleted list
Description	Text blob
Solution	Text blob
Provided and/or Discovered by	Text blob
Changelog	Text blob
Original Advisory	Text blob
Other References	Text blob
Alternate/Detailed Remediation	Text blob
Deep links	Text blob

Table 1.4 Secunia Vulnerability Report

When examining vulnerability reports from different document producers, the differences are vast and the issues of correlation become clear. Because no vendor, research organization, or coordinator works within the same framework, vulnerability report formats vary wildly. This lack of a common documentation framework causes multiple problems.

1.2.1 Multiple Formats Mean More Complexity

First and most obviously, when there is more complexity, there is extra effort and overhead introduced. The attributes defined in one format need to be interpreted correctly, mapped (or partially mapped), and then converted into the attributes of another format.

For example, when multiple reports are compared, any unintentional misuse or subjective interpretations could trigger potential flaws when addressing vulnerability issues. Theoretically, the complexity of interpretations and mappings is $O(n^2)$ when sharing vulnerability information among n different formats. This is the same sort of complexity encountered with the classically flawed computer-science sorting algorithm known as a bubble sort. This algorithm requires multiple passes and comparisons over a list of items. As the input grows, the complexity grows quadratically, which is the same situation a user would encounter with multiple uncorrelated reporting formats.

1.2.2 Organizational Compatibility Issues

Additionally, multiple reporting formats create a substantial compatibility issue. When one format is updated by either the removal of an existing field or the addition of a new one, any organization that uses multiple formats must cope with these changes.

1.2.3 Stunted Extensibility

Finally, without the common framework for adding new attributes, each format has to be modified to satisfy the needs of any potential future attributes. It is nontrivial to make the same new attribute consistently defined and understood across all formats. This makes expansion of existing multiple frameworks extremely difficult.

2. Proposed Solution

As described above, while wildly different, each reporting format does contain certain portions of similar or even identical types of information (date fields, overview-type fields, impact and remediation fields).

The issue is that there is no common format or nomenclature among the reporting formats. To speed up the document production and consumption, a common, machine-readable format for security information exchange is required.

The proposed solution, the Common Vulnerability Reporting Framework (CVRF) is an XML-based framework that predefines a large number of fields designed with extensibility and robustness in mind. These fields will be consistent in naming and data type across the board, so any

organization that adopts and understands CVRF will have no problem in producing or reading CVRF documents from another CVRF-equipped organization. Resulting documents based on this framework will all adhere to a common format that will become commonplace and familiar for all users.

2.1 From Standardization Comes Cohesion

The Industry Consortium for the Advancement of Security on the Internet (ICASI), a vendor-neutral, industry-wide think tank that tackles international and multivendor security challenges, has adopted the CVRF project. Chaired by Cisco, the ICASI CVRF working group has assembled a team of experts that collectively have written, published, and studied many forms of vulnerability documentation.

The goal has been to establish a core team that could expand existing security documentation formats and subsequently integrate a best-of-breed solution into a common XML-based framework. The outcome of this goal is to create an open standard that brings consolidation and consistency to the security documentation space and grows organically among stakeholders.

When complete, CVRF will standardize security documentation in the form of an XML-based framework that is available to anyone who chooses to use it. Independent discoverers of bugs, large vendors, security coordinators, and end users of security response efforts worldwide will be able to write CVRF documents to share critical vulnerability-related information. CVRF will enable the acceleration of information dissemination and exchange as well as incident resolution. Ultimately, producers of vulnerability documents will benefit from a faster and more consistent report creation process, and users of the documents will be able to find relevant information more quickly and easily.

2.2 CVRF Roles

There are two contextual roles that users of CVRF will assume: a document producer and a document consumer. A given CVRF user may assume one or both of these roles.

2.2.1 Document Producer

Document producers are the top-level CVRF role. They create one or more CVRF documents and distribute them in a one-to-many fashion. The document producer handles all the details of production, including responsibility for the veracity of the information as well as the actual distribution (using such methods as HTTP, SOAP, or automated feeds). Typical document producers are the following:

- Vendors
- Coordinators
- Researchers

One or more document consumers are the assumed recipients of CVRF documents.

2.2.2 Document Consumer

A document consumer is any end user who examines and acts on information in CVRF documents. Typical document consumers are the following:

- Security practitioners
- Administrators

2.3 Approach

The CVRF working group used a two-pronged approach. The first part was based on industry

outreach to collect a wide sample of reports from the industry. The second part was to survey end users about the similarities and differences in current vulnerability reporting as well as what future reporting should address.

With this information, the working group defined a syntax that incorporates an array of elements to define structures typically found in conventional vulnerability reports or security advisories. Items such as vulnerability information, exploit status, affected platform, and remedial information are all cleanly and discretely represented.

Additionally, some existing element definitions are being incorporated or considered for future use. These definitions are already defined by other markup languages, such as Security Content Automation Protocol (SCAP) [8] standards, and include the Common Product Enumeration (CPE) [9], the Common Weakness Enumeration (CWE) [10], and the Open Vulnerability and Assessment Language (OVAL) [11].

Appendix A: CVRF FAQ v1.0

1. What is CVRF?

The Common Vulnerability Reporting Framework (CVRF) is an XML-based language that is designed to provide a standard format for the dissemination of security-related information. CVRF is intended to replace the myriad of nonstandard vulnerability reporting formats with one format that is machine readable. Current security documents such as vulnerability reports and security bulletins are produced in different formats that typically require manual consumption. CVRF provides a standard, rigid language that document producers (such as vendors, coordinators, and researchers) can use to generate a document in a common and expected format. Document consumers (such as security practitioners and administrators) will be able to parse and understand this format. Additionally, because CVRF is XML based, document consumers will be able to submit CVRF documents to automated parsers and processors for tasks such as priority escalation, trouble ticketing, patch management, and cataloging.

2. What problems does CVRF solve?

CVRF solves the problem of the missing standard in the security documentation space. At present, vulnerability documentation is an ad hoc, producer-specific, nonstandard process. For example, document producers compile, collate, and create their own versions of a vulnerability document that may or may not be similar to comparable reports by other producers. Because each producer employs a unique and non-cooperative document structure, users must manually parse individual reports to find information that is germane to their environments. Additionally, the documents are typically flat and do not facilitate or support automated processing.

3. Who should be interested in CVRF?

CVRF should be of interest to the following parties:

- Technology producers (as document producers)
- Large vendors (as document producers)
- Security firms and researchers (as document producers)
- CERTs (as document producers)
- Enterprise (as document consumers)
- Government (as document consumers)

4. Is CVRF centralized? Is there a CVRF repository?

CVRF is simply a standard for creating documents. It is not centralized. However, large organizations that support CVRF as document producers are encouraged to create their own repositories or document feeds for their user base of document consumers.

5. How is CVRF delivered to the end user?

Unlike OVAL, CVE, and other such standards, CVRF is designed to be delivered independently by a document producer to the document consumer. Although CVRF documents can be delivered in a variety of ways, work is in progress to recommend specific delivery methods. Methods under consideration include direct file-based downloads, RSS feeds, and SOAP-type web services.

6. Why would I want to use CVRF?

CVRF can provide the following advantages:

- Automated processing of vulnerability reports

- Time reduction in problem resolution
- Participation in a standard format with strong multivendor support

7. As a document producer, how can I implement CVRF?

To implement CVRF, download the CVRF schema (written in XML Schema) to use to validate your documents.

8. As a document consumer, how can I implement CVRF?

To process CVRF documents, download the CVRF parser when it is available.

9. Is there a CVRF parser?

As of this writing, there is no CVRF parser, but there are plans to release one by late-2010. Because CVRF is XML, it can be parsed by existing XML tools (for example, XSLT).

10. Is CVRF free?

CVRF is free. Absolutely. The goal is to provide a standard that the industry can use to provide documents from dissimilar sources in a standard way. The CVRF working group intends to transfer custodianship of CVRF to a standards body that will ensure that CVRF remains both stable and free for use.

11. Are there any sample CVRF documents?

Several sample CVRF documents will be available at the CVRF web page of the Industry Consortium for the Advancement of Security on the Internet (ICASI) website.

12. Are CVRF document producers required to support the MITRE enumeration standards CVE, CPE, and CWE?

While strongly encouraged, document producers are not currently required to support **CVE, CPE, and CWE** because they are optionally included in a given CVRF document.

13. Does CVRF support SCAP?

As of this writing, CVRF supports most of the Security Content Automation Protocol (SCAP) standards (CVE, CPE, and CWE), with plans to support more, including OVAL.

14. How does CVRF differ from OVAL?

The Open Vulnerability and Assessment Language (OVAL) provides an XML representation for determining whether software vulnerabilities and configuration issues exist on targeted systems. Although an OVAL definition can also contain information from an associated security advisory, such information is not the primary focus of the language, and the representation does not contain the breadth of elements that make it ideal for writing security bulletins and reports.

CVRF is focused on being a robust language for describing security documentation, able to be created and distributed by a document producer in real time without requiring a central repository.

15. Does CVRF support OVAL?

As of version 1.0, CVRF does not directly support OVAL, but it is possible for a CVRF document to reference any related OVAL patch or vulnerability definitions. The CVRF working group plans to support OVAL directly in future releases.

16. How does CVRF differ from CYBEX?

The Cybersecurity Information Exchange Network (CYBEX) is an ITU draft proposal that introduces "a common framework for providers and cybersecurity centers to exchange cybersecurity related information in a structured and trusted way; this exchange may occur locally or globally among all kinds of communities and entities." [12]. CYBEX is attempting to solve a number of problems by providing a framework for the trusted transfer of information security data. Vulnerability sharing is a small piece of that problem.

17. How does CVRF differ from the ISO-29147 document on responsible vulnerability disclosure?

Responsible vulnerability disclosure is an ongoing topic of debate in the information security community. Various entities support full disclosure or responsible disclosure. CVRF is designed to provide a standard machine-consumable method for vulnerability disclosure.

18. When will CVRF be available?

At the time of this writing, mid-2011 is the tentative timeframe to release an initial version of CVRF for use by the general public.

19. Who will be publishing CVRF content?

The goal is that any organization that publishes security documentation will employ CVRF. The adoption of CVRF will grow organically through time. An abridged list of organizations that have pledged to adopt and publish CVRF documents is at this link: http://www.icasi.org/CVRF.

20. How can I incorporate CVRF content into my environment?

How you choose to implement the CVRF content depends on your environment. CVRF will include an XML Schema and parser to get you started.

21. How is a CVRF document organized?

The CVRF language was designed to be as accommodating and extensible as possible, but all documents need a few core elements to be recognized as CVRF documents. These core elements include the following:

- Document Title
- Document Type
- Document Publisher
- Issuing Authority
- Document ID
- Document Status
- Document Version
- Document Revision History
- Document Initial Release Date
- Document Current Release Date
- Document Generator (Schema Version only)

All other elements are optional.

22. Why XML?

The benefits of using XML are numerous. For CVRF, the following benefits are key:

- More precise document structure
- Improved search and navigation
- Machine-readable format for automated processing

23. Why doesn't CVRF support specific presentation markup elements such as charts and tables?

CVRF is designed as a machine-readable document format. The goal is to present the data to enable automated systems to quickly and efficiently integrate the information into either case management or knowledge base solutions.

The CVRF specification also provides classification tags for data so that document producers may deliver many levels of information that are destined for different target audiences. The processors of this information can then selectively create custom notifications, e-mail messages, or reports that consist of only the information that an executive, management, or technical end user may need in a format that is familiar to those parties.

CVRF does provide a mechanism to link to a vendor-provided document that uses the vendor's own look and feel using the Related Document element.

24. Who is building CVRF?

ICASI has established a working group to research and build the CVRF standard. More information about ICASI is available at the following URL: http://www.icasi.org

25. Who owns CVRF?

No one owns the CVRF standard. It is open and free to be employed by anyone, anywhere, at any time. That said, some document producers may incorporate their own terms of service into each document. These terms may have content ownership implications.

26. How can I help the CVRF project?

Contact the CVRF project team at cvrf-feedback@memberws.org.

Appendix B: References

[1] “Common Vulnerabilities and Exposures,” MITRE, http://cve.mitre.org/, February 2010.
[2] “Common Vulnerability Scoring System,” FIRST, http://www.first.org/cvss, February 2010.
[3] “The Incident Object Description Exchange Format,” Danyliw R., Meijer, J., and Demchenko, Y., http://tools.ietf.org/id/draft-ietf-inch-iodef-10.txt, September 2006.
[4] “Cisco Security Advisory: TCP State Manipulation Denial of Service Vulnerabilities in Multiple Cisco Products,” Cisco Systems, Inc., http://www.cisco.com/en/US/products/products_security_advisory09186a0080af511d.shtml#@ID, September 2009.
[5] “Microsoft Security Bulletin MS09-048 – Critical: Vulnerabilities in Windows TCP/IP Could Allow Remote Code Execution (967723),” Microsoft Corporation, http://www.microsoft.com/technet/security/bulletin/ms09-048.mspx, September 2009.
[6] “CERT-FI Advisory on the Outpost24 TCP Issues,” https://www.cert.fi/haavoittuvuudet/2008/tcp-vulnerabilities.html, CERT-FI, September 2009.
[7] “Secunia Advisory SA36597,” http://secunia.com/advisories/36597/, September 2009.
[8] “Security Content Automation Protocol,” http://scap.nist.gov/
[9] “Common Platform Enumeration,” http://cpe.mitre.org
[10] “Common Weakness Enumeration,” http://cwe.mitre.org
[11] “Open Vulnerability and Assessment Language,” http://oval.mitre.org
[12] “Proposed Initial Draft for Rec. ITU-T X.cybex” https://datatracker.ietf.org/documents/LIAISON/file716.pdf, September 2009