A Progress Report on the CVE Initiative

Robert Martin / Steven Christey / David Baker

The MITRE Corporation

MOTIVATION FOR CVE

Common Vulnerabilities and Exposures (CVE) is an international, community-based effort, including industry, government, and academia, that is working to create an organizing mechanism to make identifying, finding, and fixing software product vulnerabilities more rapid and efficient. A few years ago, each of us was faced with a cacophony of naming methods for defining individual security problems in software. This made it difficult to assess, manage, and fix vulnerabilities and exposures when using the various vulnerability services, tools, and databases along with the software suppliers' update announcements and alerts. For example, Table 1 shows how in 1998 each of a dozen leading organizations used different names to refer to the same well-known vulnerability in the phf phonebook CGI program. Such confusion made it hard to understand which vulnerabilities an organization faced and which ones each tool was looking for (or not looking for). Then, to get the fix to the identified vulnerability, users still had to figure out what name the vulnerability or exposure was assigned by their software supplier.

Table 1 - Vulnerability Tower of Babel, 1998

Organization	Name referring to vulnerability
AXENT (now Symantec)	phf CGI allows remote command execution
BindView	#107—cgi-phf
Bugtraq	PHF Attacks—fun and games for the whole family
CERIAS	http_escshellcmd
CERT	CA-96.06.cgi_example_code
Cisco Systems	HTTP—cgi-phf
CyberSafe	Network: HTTP 'phf' attack
DARPA	0x00000025 = HTTP PHF attack
IBM ERS	ERS-SVA-E01-1996:002.1
ISS	http—cgi-phf
Symantec	#180 HTTP server CGI example code compromises http server
SecurityFocus	#629—phf Remote Command Execution Vulnerability

Driven by a desire to develop an integrated picture of what was happening on its corporate networks and while trying to properly research options for selecting some new network security tools, the MITRE Corporation[1] (http://www.mitre.org) began designing a method to sort through this vulnerability naming confusion. The approach involved the creation of a unified reference list of vulnerability and exposure names that were mapped to the equivalent items in each tool and database. In January 1999, MITRE presented a paper at the 2nd Workshop on Research with Security Vulnerability Databases at Purdue University [1] that outlined the concept and approach for what today is known as the Common Vulnerabilities and Exposures Initiative (http://cve.mitre.org). The primary product of this Initiative is the CVE List, a reference list of standard names for vulnerabilities and exposures.

The CVE List was envisioned as a simple mechanism for linking vulnerability-related databases, tools, and concepts. It was believed to be critical for the information-security community to concur with the CVE approach and begin incorporating the common names into their various products and services. Therefore, CVE's role was limited to that of a logical bridge to avoid competing with existing and future commercial efforts.

[1] MITRE is a not-for-profit company that works in the public interest to provide systems engineering, research and development, and information technology support to the U. S. government

Although the CVE name itself was simple in concept, there would be nothing simple about implementing the CVE Initiative. To be successful, all existing vulnerability information would have to be examined and compared to determine which parts of this overall set of information referred to the same problem. Then, unique and consistent descriptions for each problem would have to be created, and the technical leaders of the information security community would have to be brought together to agree on the descriptions. The CVE List would have to be broadly distributed for commercial vendors and researchers to adopt it. A CVE compatibility evaluation process would have to be designed to verify vendor claims of support for the CVE names in products and services, and policies would have to be created to encourage the use of CVE-compatible products. The CVE Initiative would also have to be an ongoing effort since new vulnerabilities are always being discovered, and at an increasing rate. Finally, the CVE Initiative had to include international participation in both those helping with the development of the CVE List, and by the vendor community and other organizations using the common names in their products and services.

To guide the various aspects of the CVE Initiative to enable the adoption of the CVE List as a common mechanism for referring to vulnerabilities and exposures, CVE has targeted five specific areas of activity. These focuses are:

- Uniquely naming every publicly known information security vulnerability and exposure.
- Injecting CVE names into security and vendor advisories.
- Establishing CVE usage in information security products as common practice.
- Having CVE usage permeate policy guidelines about methodologies and purchasing, included as requirements for new capabilities, and introducing CVE into training, education, and best practices suggestions.
- Convincing commercial software developers to use CVE names in their fix-it sites and update mechanisms.

The remainder of this paper will describe the various challenges, solutions, and approaches that the CVE Initiative has undertaken (or faced) in the development of the various elements of the CVE Initiative.

IMPLEMENTING THE CVE INITIATIVE

After a positive response from the Purdue CERIAS Workshop, MITRE formed the CVE Editorial Board in May 1999 with 12 commercial vendor and research organizations, which worked to come to agreement on the initial CVE List with MITRE as moderator. During this same time, a MITRE team worked to develop a public Web site to host the CVE List, archive discussions of the Editorial Board, and host declarations of vendor intent to make products CVE-compatible. The CVE Initiative was publicly unveiled in September 1999. The unveiling included an initial list of 321 entries, a press release teleconference, and a CVE booth that was staffed with the Editorial Board members at the SANS 1999 technical conference. It was a very powerful message to attendees to see the CVE booth staffed by competing commercial vendors working together to solve an industry problem. There was a large audience of system administrators and security specialists in attendance, who had been dealing with the same problem that motivated the creation of the CVE Initiative.

As the volume of incoming vulnerability information increased for both new and legacy issues, MITRE established a content team to help with the job of generating CVE content. The roles and responsibilities for the Editorial Board were formalized. MITRE worked with vendors to put CVE names in security advisories as vulnerabilities were announced, and worked with the CVE Senior Advisory Council to develop policy recommending the use of CVE-compatible products and services and to find ways of funding the CVE Initiative for the long-term. Since the beginning, MITRE has promoted the CVE Initiative in and at various venues, including hosting booths at industry tradeshows, interviewing with the media, publishing CVE-focused articles in national and international journals [2], and presenting CVE-focused talks in public forums and conferences.

THE CVE LIST

The CVE Initiative has had to address many different perspectives, desires, and needs as it developed the CVE List. The common names in the CVE List are the result of open and collaborative discussions of the CVE Editorial Board (a deeper discussion of the Board can be found later in this paper), along with various supporting and facilitating activities by MITRE and others. With MITRE's support, the Board identifies which vulnerabilities or exposures to include on the CVE List and agrees on the common name, description, and references for each entry. MITRE maintains the CVE List and Web site, moderates Editorial Board discussions, analyzes submitted items, and provides guidance throughout the process to ensure that CVE remains objective and continues to serve the public interest.

CVE Candidates versus CVE Entries

CVE candidates are those vulnerabilities or exposures under consideration for acceptance into the official CVE List. Candidates are assigned special numbers that distinguish them from CVE entries. Each candidate has three primary items associated with it: number (also referred to as a name), description, and references. The number is an encoding of the year that the candidate number was assigned and a unique number N for the Nth candidate assigned that year, e.g., CAN-1999-0067. If the candidate is accepted into CVE these numbers become CVE entries. For example, the previous candidate number would have an eventual CVE number of CVE-1999-0067, where the "CAN" prefix is replace with the "CVE" prefix. The assignment of a candidate number is not a guarantee that it will become an official CVE entry.

Data Sources and Expansion of the CVE List

Throughout the life of the CVE List, MITRE has relied on other data sources to identify vulnerabilities. As a result, MITRE can concentrate on devising the standard names, instead of "reinventing the wheel" and conducting the research required to find the initial vulnerability reports. Before CVE was publicly released in September 1999, a "draft CVE" was created and submitted to the Editorial Board for feedback. ISS, L-3 Security (later acquired by Symantec), SANS, and Netect (later acquired by BindView) provided information that was used to help create the draft CVE. Data was also drawn from other sources including Bugtraq and NTBugtraq posts, CERT advisories, and security tools such as NAI's CyberCop Scanner, Cisco's NetSonar, and AXENT's NetRecon.

In November 1999, two months after the first version of the CVE List was made available, MITRE asked Editorial Board members to provide a "top 100" list of vulnerabilities that should be in the CVE List, which produced over 800 total submissions. Contributing organizations were Purdue CERIAS, ISS, Harris, BindView, Hiverworld (later nCircle), Cisco, L-3 Security (later acquired by Symantec), and AXENT (later acquired by Symantec). At this time, MITRE also began processing newly discovered vulnerabilities, using the periodic vulnerability summaries published by SecurityFocus, Network Computing/SANS, ISS, and the National Infrastructure Protection Center (NIPC).

To manage the volume of vulnerabilities that were submitted, MITRE began developing the submission matching and refinement process that is described later in this paper.

In Summer 2000, MITRE again sought to expand the CVE List to include older "legacy" problems that were not in the original draft CVE, this time receiving copies of the vulnerability databases from 10 organizations, which contained a total of approximately 8,400 submissions. The contributors were AXENT (now Symantec), BindView, Harris Corporation, Cisco Systems, Purdue University's Center for Education and Research in Information and Security (CERIAS), Hiverworld (now nCircle), SecurityFocus, Internet Security Systems (ISS), Network Associates, L3 (now Symantec), and the Nessus Project. These contributions were made while newly discovered issues were also being processed in parallel. In the following year, MITRE expanded its support staff and improved its processes and utilities for dealing with the increasing volume of information.

Of the 8,400 legacy submissions received in Summer 2000, MITRE has thus far eliminated 2,500 submissions that duplicated existing candidates or entries, or did not meet the CVE definition of a vulnerability or exposure. An additional 3,900 submissions require additional information from the source that provided them (generally due to lack of references or vague descriptions), and 1,100 have been set aside for more detailed examination and study. Many of these 1,100 vulnerability submissions describe insecure configurations and require further study. Configuration problems are difficult to identify with CVE because configuration is system-dependent, such problems are not as well studied as software implementation errors, and they could be described at multiple levels of abstraction. MITRE's research and analysis is currently focusing on the Windows-based portion of these configuration problems.

The remaining 900 legacy submissions formed the basis of 563 CVE candidates that were proposed to the Board in September 2001. A small number of submissions from November 1999 still remain, mostly due to the lack of sufficient information to create a candidate.

While MITRE processes the remaining legacy submissions and conducts the necessary background research, MITRE continues to receive between 150 and 500 new submissions per month from ISS, SecurityFocus, Neohapsis, and the National Infrastructure Protection Center. Each month, an additional 10 to 20 specific candidates are reserved before a new vulnerability or exposure is publicly known, with the candidate number then included in vendor and security community member alerts and advisories. To date, a variety of individuals and organizations have reserved candidate numbers for use in public announcements of vulnerabilities, including ISS, Rain Forest Puppy, BindView, Compaq, Silicon Graphics, IBM, the Computer Emergency Response Team Coordination Center (CERT/CC), Microsoft, Hewlett-Packard, Cisco Systems, and Red Hat Linux.

Because there was an increased emphasis on creating legacy candidates during the summer of 2001, a backlog of submissions for recent issues developed. Candidates for those issues should be created by early 2002, and additional processes will be implemented to avoid such backlogs in the future. One avenue that is being explored to address this problem is the active pursuit of vendors and researchers to include CVE candidate names in their initial advisories and alerts.

Growth of the CVE List since Inception

As previously mentioned, the first version of the CVE List was released in September of 1999, it contained 321 CVE entries that MITRE had researched and reviewed with the initial Editorial Board members. As Figure 1 shows, the number of entries in the CVE List stands at 2,032 entries as of early May 2002, while candidates number 2,325. Notable increases occurred in November 1999, September 2001, and February/March 2002 in conjunction with the growth of the list as described in the previous section. The CVE Web site now tracks some 4,350 uniquely named vulnerabilities and exposures, which includes the current CVE List, recently added legacy candidates, and the ongoing generation of new candidates from recent discoveries.

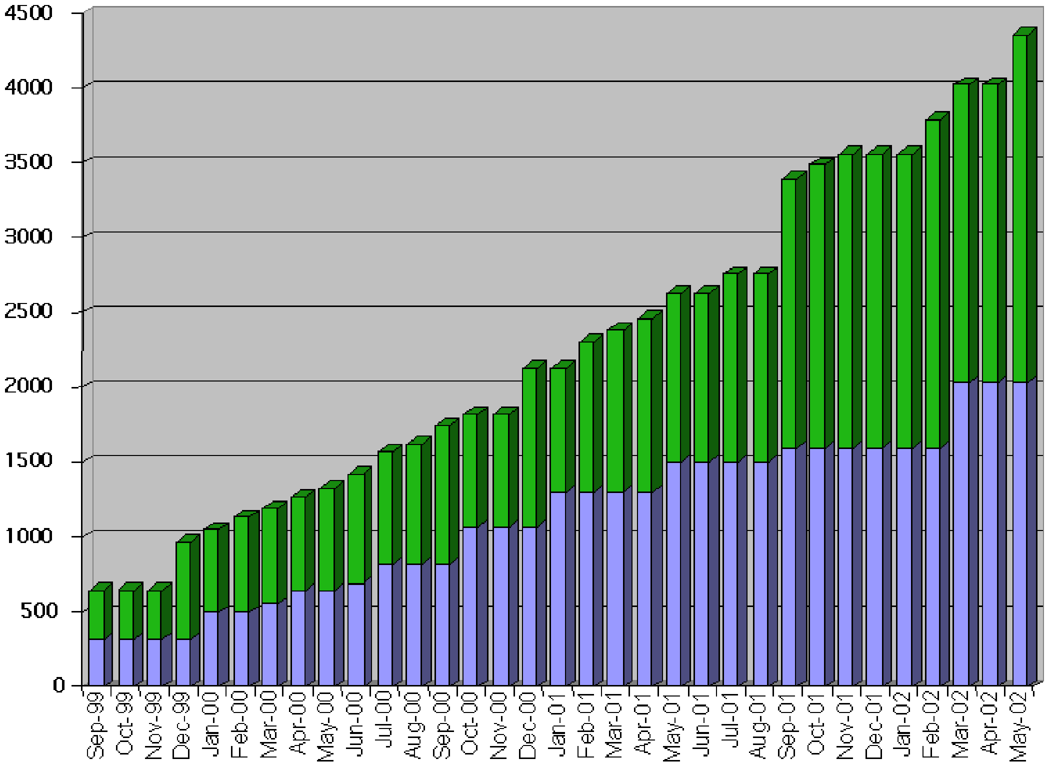

Figure 1. CVE growth over time

The Process of Building the CVE List: The Submission Stage – *Stage 1 of 3*

The CVE review process is divided into three stages: the initial submission stage, the candidate stage, and the entry stage. MITRE is solely responsible for the submission stage but is dependent on its data sources for the submissions. The Editorial Board shares the responsibility for the candidate and entry stages, although the entry stage is primarily managed by MITRE as part of normal CVE maintenance.

Content Team. For the CVE project, MITRE has a content team whose primary task is to analyze, research, and process incoming vulnerability submissions from CVE's data sources, transforming the submissions into candidates. The team is led by the CVE Editor, who is ultimately responsible for all CVE content.

Conversion Phase. During the submission stage, MITRE's CVE Content Team, which consists of MITRE security analysts and researchers, collects raw information from various sources, e.g. the various Board members who have provided MITRE with their databases, or publishers of weekly vulnerability summaries. Each separate item in the data source (typically a record of a single vulnerability) is then converted to a "submission," which is represented in a standardized format that facilitates processing by automated programs. Each submission includes the unique identifier that is used by the original data source.

Matching Phase. After this conversion phase, each target submission is automatically matched against all other submissions, candidates, and entries using information retrieval techniques. The matching is based primarily on keywords that are extracted from a submission's description, references, and short title. The keywords are weighted according to how frequently they appear, which generally gives preference to infrequently seen terms such as product and vendor names and specific vulnerability details. Keyword matching is not completely accurate, as there may be variations in spelling of important terms such as product names, or an anomalous term may be given a larger weight than a human would use. The closest matches for the target submission (typically 10) are then presented to a content team member, who identifies which submissions are describing the same issue (or the same set of closely related issues) as the target submission.

Once matching is complete, all related submissions are combined into submission groups, which may include any candidates or entries that were found during matching. Each group identifies a single vulnerability or a group of closely related vulnerabilities. These groups are then processed in the next phase, called “refinement.”

Refinement Phase. Typically, a content team member is assigned a batch of 20 or more submission groups, which usually includes both duplicate submissions and new issues.

During refinement, the team member analyzes a submission group and determines whether one or more of the submissions identify an existing CVE item. If so, then the analyst notes any additional references that are in the new submission, but not the existing CVE item, so that the existing CVE item’s references can be extended.

If there are submissions from the group that do not describe an existing CVE item, then the a team member makes the following assessment:

- Apply CVE content decisions to decide whether any candidates should be created.
- Apply CVE content decisions to decide how many candidates must be created.

(Content decisions are covered in a later section.)

For each candidate to be created, the analyst does the following:

- List the associated references using CVE’s canonical reference format.
- Create a description.
- Determine if there is vendor acknowledgement.
- Identify any related content decisions.
- Identify other supporting information such as the date the problem was announced, high-level operating system (OS), whether the issue is remotely or locally exploitable, and a few other attributes. This information is used to group sets of candidates later in the process, or to provide tailored voting ballots to individual Editorial Board members.
- Identify any keywords that could help in later submission matching (as well as the CVE Web site’s search engine). Typically the keywords include alternate spellings or terms that were not explicitly necessary for the description.
- Identify the rationales for acknowledgement and content decisions in the “analysis” section.

In some cases, an analyst may choose to delay analysis of a submission group (or a portion of the group) when an issue is unusually complex or if other individuals need to be consulted.

Submission refinement is a bottleneck because deep analysis is sometimes required to understand the reported problem, apply the content decisions, find vendor acknowledgement, research the references, and write the descriptions. Refinement is especially difficult for new analysts because there is a large amount of detail and background knowledge that is required before the analyst can be comfortable and productive in doing refinement.

For each action that the content team member undertakes - whether identifying a duplicate, rejecting a submission, or suggesting the creation of a new candidate - a “refinement group” is produced. One or more refinement groups are produced from the original submission group, depending on how many separate issues were in the original submission group.

Editing Phase. After refinement, the CVE Editor reviews the work of the analysts, occasionally making modifications to follow CVE style, ensure that CVE content decisions are being followed, or to do advanced research. Occasionally, submissions may be added or removed from the refinement groups. The Editor provides feedback to the analyst for the purposes of training or to raise certain issues. Since the submission matching may not always find all related submissions, typically due to spelling inconsistencies across submissions, the Editor may merge multiple refinement groups that were produced by different analysts.

The Editor then processes the resulting refinement groups. New candidate numbers are assigned to the groups that identify new issues (the "assignment" phase in the candidate stage).

After candidate assignment, each data source is provided with a backmap from their submission to the associated CVE items (whether newly created candidates, or existing candidates or entries). The backmap can reduce the amount of effort that the data source needs to perform to maintain a mapping to CVE. After the backmaps for the candidates are generated, the associated submissions are removed from the submission pool. In addition to backmaps, a new type of map referred to as a "gapmap" is also provided to the information source. The gapmap identifies the newly created candidates that were not found in the data source's original set of submissions, which may make the source aware of additional security problems that they had not seen previously.

In some cases, the submission stage may be entirely bypassed. This usually happens when an individual or organization reserves a candidate number in order to include it in the initial public announcement of a vulnerability, as described in further detail in a later section.

The Process of Building the CVE List: The Candidate Stage – *Stage 2 of 3*

Assignment Phase. Candidates are normally created in one of three ways. They are refined by the content team using submissions from CVE's data sources; they are reserved by an organization or individual who uses it when first announcing a new issue; or they are created "out-of-band" by the CVE Editor, typically to quickly create a candidate for a new, critical issue that is being widely reported.

Proposal Phase. The CVE Editor organizes candidates into clusters of 20 to 50 candidates. For new issues, the clusters are typically grouped by the initial public announcement dates of the candidates. For legacy issues, the clusters may be created according to other criteria that make the clusters more manageable for Editorial Board members to work with, such as Unix vendor advisories. The candidate clusters are then proposed to the Board for review and voting.

Voting Phase. Editorial Board members review proposed candidates, registering their feedback with a vote and optional commentary. Votes include ACCEPT, MODIFY (signifying the need for a minor change), REJECT, RECAST (signifying the need for a major change), REVIEWING (member is actively reviewing the candidate but does not have a vote ready), and NOOP (no opinion). A Board member may ACCEPT or MODIFY a candidate if (1) it has been acknowledged by the vendor, (2) if the issue has been replicated by the voting Board member, (3) if the issue has been reported or replicated by someone whom the member trusts, or (4) if there is independent confirmation from another party. MITRE is considering whether (4) is sufficient to establish the veracity of a candidate. One issue that has not yet been resolved is how to deal with "permanent" candidates that may be real, but never receive enough positive votes to be accepted as official entries.

Modification Phase. The candidate may be modified based on feedback from Board members. (More information on this is in the "Modification" section below).

Interim Decision Phase. The CVE Editor decides when the review of a candidate is complete or has come to a standstill. The Editor casts a single ACCEPT or REJECT vote, then gives Board members a "last call" opportunity to post any final comments or objections (at least 4 business days). If there are extensive comments or objections that require additional voting, the candidate may be returned to the Modification phase.

Final Decision Phase. If the CVE Editor determines that no sufficient grounds exist for changing the vote made in the Interim Decision, the decision becomes final. If the candidate is ACCEPTed, the Editor announces to all Board members that the candidate will be placed into CVE, and identifies the CVE name(s) that will be assigned to the new entry. If the candidate is REJECTed, the Editor notes the reason for rejection.

The Process of Building the CVE List: The Entry Stage – *Stage 3 of 3*

Publication Phase. If the candidate has been ACCEPTed, the candidate is converted into an entry by changing the name from CAN-YYYY-NNNN to CVE-YYYY-NNNN and removing the voting record. The new entry is then added to the next version of the CVE List.

Modification Phase. The entry may need to be modified in simple ways, e.g., to clarify the description or add more references. (More information on this is in the following section).

Modifications and Deletions in the CVE List and Candidates List

Modification. Most candidates and entries are modified by adding more references (such as additional vendor advisories), or through small changes to descriptions (such as fixing typos and clarifying the issue). Candidate modifications are normally not explicitly presented to the Editorial Board or the public, due to the number and frequency of changes that take place. For entries, the Editorial Board is notified of basic modifications at least four business days before the new CVE version is targeted for creation.

For CVE users who want to track modification in the CVE List, MITRE provides "version difference reports" that detail which entries have changed, and how they have changed, between two versions. For various reasons, this capability was not made available for candidates, but the Cassandra project being led by Purdue CERIAS now offers a change monitoring report that includes changes to candidates (https://cassandra.cerias.purdue.edu/CVE_changes/).

Some modifications may be substantial. For example, a candidate may need to be split into multiple items, or multiple candidates may need to be merged into a single item (RECAST). This will happen if a content decision was not applied properly when the candidate was first created, or if new information forces such a change. In some cases, a RECAST may be required for entries. The procedure for recasting candidates and entries has not been completely defined, because most of these changes are due to content decisions that have not been finalized yet. However, it is certain that the procedure will cover including forward pointers from any recast item to the correct items.

In other cases, a description and/or the set of references may be vague enough that the item could appear to describe more than one different vulnerability. This happened more frequently in the early days of CVE when the utility of references in deconflicting similar issues, and the importance of having necessary details in the descriptions, was not fully understood. Vague descriptions and missing references can lead to mapping errors in CVE-compatible products/services. Vendor security advisories with vague information present a special challenge: the issue is likely to be real (otherwise the vendor would not have reported it), but the issue could already be identified in a different CVE item. Consultation with the vendor may clear up any ambiguity, but it is not always possible or feasible.

Deletions. There may be several reasons why a candidate or entry should be "Deleted" from its associated list:

- If it is a duplicate of another CVE item.
- If further analysis shows that the vulnerability does not exist (e.g., the original report was incorrect).
- If the item needs to be recast.

Since any number of CVE-compatible products and services could be using older CVE identifiers, it is important to keep a record of what happens to each item that must be "deleted." A *candidate* is deleted by rejecting it. An *entry* is deleted by deprecating it. The process is the same for candidates and entries, and includes the following:

- An announcement is made to the Editorial Board that the item will be rejected or deleted.
- At least four business days are allowed for Board members to raise any questions (for candidates, this takes place in the Interim Decision phase)
- A Final Decision is made to Reject or Deprecate the item.
- All references for the item are deleted.
- The description is removed, and replaced with a statement that says that the item has been Rejected (for candidates) or Deprecated (for entries).
- A short reason for the action is included in the description.
- If the item is a duplicate, a reference is made to the correct CVE item(s).
- The change is noted in the next CVE difference report.
- The item remains in its associated list, so that there is always a record of what happened to it.

The references and descriptions are removed so that (1) it is clear to everyone that the item is no longer identifying the original vulnerability, and (2) the item is not returned as a result of keyword searches.

CANDIDATE RESERVATION AND CANDIDATE NUMBERING AUTHORITIES

Candidate reservation allows responsible researchers, vendors, and incident response teams to include candidate numbers in the initial public announcement of a vulnerability. It ensures that a candidate number is instantly available to all CVE users and makes it easier to track vulnerabilities over time.

The basic process is:

1) There is a request for one or more candidate number(s).
2) MITRE reserves the candidate number(s) and provides the number(s) to the requester, and creates "blank," content-free candidate(s) on the CVE Web site.
3) The requester shares the candidate number(s) with all parties involved in the disclosure.
4) The requester includes the candidate number(s) in the vulnerability advisory.
5) The requester makes the candidate(s) public and notifies MITRE.
6) MITRE updates the candidate(s) on the CVE Web site to provide the details.
7) MITRE proposes the candidate(s) to the Editorial Board.
8) If a candidate is accepted as an official CVE entry, then the requester updates the number in the advisory.

If a candidate was reserved and the issue was never made public, the candidate will be deleted. This is referred to as "Releasing" the candidate. Since the candidate was never public—and in some cases, the candidate was never assigned to a specific vulnerability—it is deleted entirely.

Candidate Numbering Authorities

Candidate Numbering Authorities (CNAs) are organizations that distribute CVE candidate numbers to researchers and information technology vendors for inclusion in first-time public announcements of new vulnerabilities, without directly involving MITRE in the details of the specific vulnerabilities. On an as-needed basis, MITRE provides a CNA with a pool of candidate numbers for the CNA to assign.

CNAs can help the CVE Initiative in several ways. When they function as intermediaries between a vulnerability researcher and the affected vendor, they can provide a candidate number without notifying MITRE of the vulnerability, which reduces the risk of accidental disclosure of vulnerability information. They increase the scope and visibility of CVE candidates by providing additional access points for researchers and vendors to obtain candidate numbers. They can utilize existing working relationships with researchers and vendors, which the affected parties may not have formed with MITRE. If they are already an integral part of the normal process by which vulnerabilities are disclosed, their participation prevents the addition of another party (i.e., MITRE) from interfering with that process or causing further delays. Finally, their participation relieves MITRE of some potentially labor-intensive tasks, allowing it to dedicate resources to other aspects of CVE that need attention.

Requirements to be a CNA

A CNA must be a major software vendor with a significant user base and an established security advisory capability, or an established third party that typically acts as a neutral interface between researchers and vendors. MITRE also functions as a CNA in a limited capacity.

The CNA must be an established distribution point for first-time vulnerability announcements. It must have a member of the Editorial Board who performs technical tasks. In keeping with the CVE requirement to identify public issues, the CNA must only assign candidates to security issues that will be made public. Finally, it must follow responsible disclosure practices that are accepted by a significant portion of the security community. Responsible disclosure is important for CVE because otherwise, it is more likely that duplicate or inaccurate information will be introduced into CVE.

CNA Tasks

CNAs must consistently apply documented CVE content decisions (with exceptions made for technical subtleties or incomplete documentation). They must also coordinate the exchange of candidate numbers across all involved parties (vendor, researcher, response team, etc.) in order to reduce the risk of producing duplicate candidate numbers. CNAs must notify MITRE when candidates have been publicly announced. Since disclosure practices directly impact the accuracy of CVE, CNAs must recommend best practices in vulnerability disclosure to both researcher and vendor. A CNA must verify that the reported vulnerability has not already been assigned a CVE or candidate number.

MITRE is working to increase the number of CNAs. Some of the greatest challenges are educating CNAs about content decisions and determining the process for exchanging candidate numbers across multiple parties, especially if more than one party reserves candidates from MITRE.

Communications from CNAs to MITRE

The following series of communications occur between CNAs to MITRE:

1) The CNA requests a pool of candidate numbers.
2) The CNA announces the publication of a new candidate, which allows MITRE to update the candidate information on the CVE Web site.

3) The CNA may need to consult with MITRE regarding CVE content decisions.
4) The CNA notifies MITRE of suspected abuses of the CVE process by researchers.
5) The CNA notifies MITRE and other parties when duplicate candidates are detected.

The primary method of communication is through email, although telephone discussions are sometimes necessary when a problem is particularly complex with respect to CVE content decisions or the nature of the vulnerability.

A third party CNA must also maintain awareness of all vendors and CNAs who utilize candidate numbers. Since a third party might gain a competitive advantage by initially providing candidate numbers to a limited audience (outside of the researcher and vendor), the CNA should not publish CVE candidate numbers in any manner which might provide it with an economic, technical, or political advantage over its competitors.

A vendor CNA must clearly advertise their security point of contact. They must provide the candidate to other affected parties, e.g., other vendors, researchers, or response teams. They must include candidate numbers in their own advisories. They can only use their pool of candidates for vulnerabilities in their own products. They must apply CVE content decisions to determine the proper number of candidates to assign, even if the content decisions are contrary to the vendor’s own criteria. If the issue does not meet the vendor’s minimum risk level for releasing an advisory, the CNA should still provide candidates for that issue. Finally, when an issue has already been published and assigned a candidate, the vendor must use the existing candidate number.

Vendor Liaisons

As can be seen by the requirements for a CNA, it can be resource-intensive and technically difficult to act as a CNA. Many vendors may want to participate properly in the CVE Initiative but not have the capability or desire to act as a CNA. A vendor liaison may work with another CNA to obtain or verify CVE candidates in the liaison’s own products, and include candidate numbers in its advisories.

Researcher Responsibilities

The researcher must reserve candidates for a particular vulnerability from only one CNA. If the affected software vendor is a CNA, then the researcher must obtain the candidate from the vendor. The researcher needs to provide the CNA with enough details for the CNA to apply CVE content decisions. The researcher has to coordinate the exchange of the candidate number across all involved parties. Finally, the researcher must include the candidate number in an advisory and publish the information through known reliable channels (vendor or response team), or known public channels with peer review (such as Bugtraq or NTBugtraq).

Researchers could adversely affect the reservation process in several different ways that could impact the overall quality of CVE. For example, the researcher’s disclosure process may frequently result in duplicate candidates, e.g., by refusing to work with a vendor. The researcher may frequently publish issues that are discovered to be false or so error-prone as to cause their associated candidates to be rejected by the Editorial Board. It is the responsibility of MITRE and the CNAs to identify and resolve such abuses.

CONTENT DECISIONS

CVE content decisions are the guidelines used to ensure that CVE items are created in a consistent fashion, independent of who is doing the creation. There are two major types of content decisions: Inclusion and Abstraction. Inclusion content decisions specify whether a vulnerability or exposure should go into CVE. Abstraction content decisions specify what level of abstraction (level of detail) a vulnerability should be described at, e.g., whether a particular security issue should be given one candidate or five candidates.

There are differences between many vulnerability databases or products in the type of content they include, as well as the level of abstraction. These differences occur within the same database or product. Because of this variety and the flat structure of the CVE name, CVE cannot be flexible enough to account for these differences. It is important for vulnerability analysts to be aware of these differences. As such, CVE content decisions not only document the guidelines for creating content, they often indicate areas in which there is inconsistency across vulnerability information sources. Quantitative analyses of vulnerabilities that use CVE-normalized data can be more easily replicated, and the CVE content decisions help to ensure that the data is normalized in a predictable fashion.

Two of the most commonly used content decisions (CDs) are shown in Table 2 below. They also highlight some of the most common discrepancies across vulnerability information sources. These CDs were revised many times over a period of a year and a half, but they were stabilized in early 2001 when they were modified to make them less sensitive to the amount of information that is available for a vulnerability. From an academic perspective this approach is not optimal, but it is proving to be repeatable and less likely to cause candidates to become split or merged when new information becomes available after the initial analysis has been performed.

Table 2 - The SF-LOC and SF-EXEC Content Decisions

CD:SF-LOC: multiple security flaws in the same executable, but possibly in different lines of code	**CD:SF-EXEC: multiple executables exhibiting the same problem**
1) CD:SF-LOC only applies when there may be multiple bugs that appear in the same executable (modulo the codebase, i.e., all “ps” executables in Unix are treated as the same). 2) SPLIT (create separate CANs) between problems of different types (e.g., buffer overflow versus symlink problem). 3) SPLIT between problems of the same type if problem X appears in some version that problem Y does not. 4) MERGE problems of the same type within the same version. Explicitly list the different problems in the description.	1) CD:SF-EXEC only applies when there are multiple executables in the same package that demonstrate the same problem. 2) “The same package” basically means “bundled executables that perform related functions that are not distributed separately.” Microsoft Word and PowerPoint are not the same package (they can be installed separately). The set of executable programs that support the lpd capability in UNIX are the same package. 3) SPLIT when the problems are of different types. 4) SPLIT when the problems are in different versions (for some definition of “version” that effectively describes the package). 5) MERGE when the problems are of the same type. Explicitly identify the separate affected “components” or executables in the package.

CD:SF-LOC is less sensitive to the lack of detailed information such as source code, exploits, or attack traces. However, it’s still sensitive to changes in version information. Problems that occur in libraries pose special challenges for this content decision, because they could be exhibited at several points within the same executable, or in many different executables. Ultimately, while this CD is intended to minimize the amount of information that is required to produce results, it is still dependent on critical information sources such as the vendor of the vulnerable product.

CD:SF-EXEC is also susceptible to error if the problem occurs in a library or other common codebase.

There are approximately 15 other content decisions currently defined for CVE, some of which are identified in the “Scope of the CVE List” section.

CVE EDITORIAL BOARD

The CVE Editorial Board includes prominent information security specialists from numerous information-security-related organizations around the world, including commercial security-tool vendors, academic and research institutions, and government agencies. MITRE invites other information security experts to participate on an as-needed basis, based upon recommendations from other Board members, or MITRE's own identification of gaps within the current representation. Archives of Board meetings and discussions are publicly available on the CVE Web site.

Members of the Editorial Board have different roles and tasks in support of the CVE Initiative. There are four roles: Technical Members, Liaisons, Advocates, and Emeritus Members. Each Board member has one primary role but can take other roles. Technical members participate in the creation, design, review, maintenance, and applications of the CVE List. Liaisons represent a significant constituency, related to or affected by CVE, in an area that does not necessarily have technical representation on the Board. In some cases, a liaison may represent an individual organization, which may include software vendors. Advocates actively support or promote CVE in a highly visible fashion. This role is reserved for respected leaders in the security community who help bring credibility to the CVE Initiative and give CVE a wider reach outside of the security community. Emeritus members were formerly active and influential in the CVE Initiative and are recognized for their significant contributions.

Board members must meet the minimum levels of effort for the tasks that they undertake, which varies across tasks. If a Board member participates in multiple tasks, then the minimum expectations for each individual task may be lowered accordingly.

All members perform Consultation and Awareness tasks. Consultation includes participating in Board meetings, or discussion of ad hoc issues related to CVE content or Editorial Board processes such as content decisions, Board membership, or CVE compatibility. Awareness includes participating in Board meetings and/or reading meeting summaries, and regularly reading posts on the Editorial Board mailing lists.

Many members also perform outreach by actively promoting CVE and educating the public about it, or introducing various contacts to the CVE Initiative. Occasionally, some Board members may participate in activities that are undertaken under the Board context, but not directly related to CVE.

Technical members regularly perform one or more of the following tasks:

- Voting: The primary task for most technical members is to review, comment on, and vote on CVE candidates proposed to the Editorial Board. Some members vote regularly. Others vote on an ad hoc basis, e.g., when there is an effort to reach a specific content goal.
- Content provider: Some Board members provide their vulnerability databases to MITRE for conversion into candidates, which ensures that CVE content is as complete as possible. Others are actively involved in candidate reservation. Others may be CNAs, which are authorized to assign CVE candidate numbers to security issues before they are publicized.
- CIEL: Members participate in the review and development of the Common Intrusion Event List (CIEL), a "CVE-for-IDS" which is currently being drafted by MITRE and is discussed later.

Liaisons perform one or more of the following tasks.

- Community Education: The liaison must educate the liaison's own community about CVE, where appropriate.
- Board Education: The liaison must educate the Editorial Board about the needs and interests for CVE of the liaison's community, where appropriate.

- Voting: If the member is a software vendor liaison, the member must vote on candidates related to that vendor's products.

Liaisons may undertake other technical tasks.

A liaison that represents a constituency beyond an individual organization must be visible and active in the liaison's constituency community. A liaison that represents an individual organization must be able to effectively communicate with the relevant parts of that organization. Software vendor liaisons must be able to effectively communicate with the vendor's security and product development teams.

Advocates' tasks include endorsing CVE to constituencies that will benefit from it, fostering better communication between constituencies, participating in Editorial Board activities (especially in decisions related to Board structure and strategic activities), and consulting when needed.

GUIDING THE DIRECTION OF CVE - THE CVE SENIOR ADVISORY COUNCIL

The CVE Senior Advisory Council was established to ensure that the CVE Initiative receives the sponsorship—including funding and guidance—required to maximize the effectiveness of CVE in supporting government efforts to improve the nation's ability to identify and respond to vulnerabilities and information assurance attacks or issues. The CVE Senior Advisory Council is composed of senior executives in U.S. government agencies, many of which provide (or provided) funding for CVE.

The Council provides business planning oversight and prioritization of new CVE and related services, discusses CVE and related security policy implications for the Federal Government, and identifies materials and resources that might be useful for Government CIOs and senior managers.

The Council promotes the adoption of CVE at the strategic level, works to assure funding for core CVE activities over the long term including outreach to Government organizations and agencies, and acts as a catalyst for CVE and related activities. The Council also brings to CVE their insights on community needs and possible areas for new CVE-related services.

Council membership is extended to the senior executives of those government organizations that provide funding for core CVE activities, as well as other executives that have the background and ability to help the Council achieve the stated objectives.

One of the Council's main roles is to provide strategic guidance for the direction of CVE. For example, the Council has encouraged MITRE to involve the various Information Sharing and Analysis Centers (ISACs) more closely in CVE, conduct outreach to large organizations outside of the security industry, define qualitative goals, and concentrate more on addressing the needs of the IDS segment of the security tools industry with respect to CVE.

CVE COMPATIBILITY

The basic premise of the CVE List is that there be one name for a vulnerability or exposure. A CVE-compatible product or service must understand the CVE names for vulnerabilities and allow the user to interact with the product/service in terms of those CVE names. This does not mean that the product/service only uses CVE names for vulnerabilities, but rather that in addition to its own native label for a vulnerability, it is aware of the CVE name for that vulnerability. This support for CVE names is central to the concept of CVE compatibility. The CVE-compatible tool, Web site, database, or service must use CVE names in a way that allows users to correlate its information with other repositories, tools, and services that also use CVE names, as shown in Figure 2.

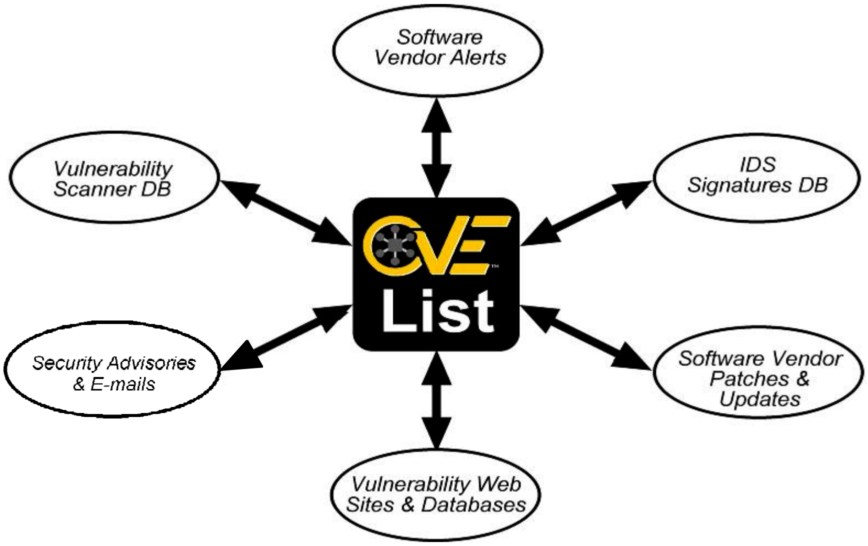
Software Vendor Alerts
Vulnerability Scanner DB
IDS Signatures DB
CVE List
Security Advisories & E-mails
Software Vendor Patches & Updates
Vulnerability Web Sites & Databases

Figure 2. Cross-Linking through the CVE List.

Uses of CVE Compatibility
Integrating vulnerability services, databases, Web sites, and tools that incorporate CVE names will provide an organization with more complete and efficient coverage of security issues. For example, a report from a vulnerability scanning tool that uses CVE names will enable the organization to quickly and accurately locate fix information in one or more of the CVE-compatible databases and Web sites.

It is also possible to determine exactly which vulnerabilities and exposures are covered by each CVE-compatible tool or service, because the CVE List provides a baseline for comparison. After determining which of the CVE entries apply to its platforms, operating systems, and commercial software packages, an organization can compare this subset of the CVE List to any particular tool's or service's coverage.

Network and security trade journals are already referring to CVE name support as a desirable feature in product reviews and comparisons of scanners and IDS devices [3] and [4]. Similarly, the National Institute for Science and Technology (NIST) has published a recommendation to all federal government agencies and services for the use of CVE-compatible products and services whenever possible [5].

Just as other types of information security products tend to focus on a particular core function or capability, platform, or types of issues, the various products, services, and repositories that strive to meet the CVE compatibility requirements will focus on different portions of the CVE List. For example, some deal with Unix while others focus on Windows NT; some focus on network-based or host-based vulnerabilities. Users must evaluate CVE-compatible items against their organization's specific needs in terms of platform and software product coverage.

The CVE Compatibility Requirements
At its core, CVE compatibility involves four basic requirements:

- Customers are able to use CVE names to inquire about scope, content, or coverage and then receive any related information.

- Customers are able to obtain output that includes all related CVE names.
- The owner of the item makes a good-faith effort to ensure that the item's mapping from its own elements to CVE names remains accurate as the CVE List and the compatible item are updated over time.
- Standard documentation includes a description of CVE compatibility, and the details of how customers can use the CVE-related functionality of their tool, database, Web site, or service.

In general, vendors are given flexibility to implement the requirements in a variety of ways. Users can then determine which features or implementations are best suited to their needs.

The CVE Compatibility Evaluation Process

MITRE's current approach for establishing the compatibility of a product or service involves two phases. The first requires the completion of a short informational "CVE Compatibility Declaration Form," which is used to register an organization's declaration of intent with respect to CVE compatibility. The organization is asked to review the compatibility requirements and then make a statement regarding whether the organization believes that its product or service currently fulfills the requirements, or if the organization is working towards fulfilling the requirements. This phase of the CVE compatibility process does not result in an official evaluation by MITRE; rather, MITRE only reviews the declaration. As long as the products or services are commercially available, the declaration and an endorsement quote from the vendor (if desired) is posted on the CVE Web site. This phase of the compatibility process has been in effect since October 1999 when the CVE Initiative started and can be done very quickly. It makes the vendor aware of high-level expectations for CVE compatibility and establishes the proper communication channels between MITRE and the organization.

When the organization believes that its product or service has obtained full compliance with the CVE compatibility requirements, it may then request a formal review and evaluation, which begins the second phase. In development for the last year, this formal process has a "branding program" and logo to indicate successful completion of the compatibility evaluation. A major component of this phase requires specific details about how an organization has satisfied each of the mandatory CVE compatibility requirements. The organization must complete an extended "CVE Compatibility Requirements Evaluation Form," which requires the signature of an authorized representative of the submitting organization. Additionally, the organization provides MITRE with the CVE-related user documentation for the product or service.

The organization's statements and documents are evaluated, and MITRE arranges to verify the accuracy of the mapping between CVE names and the organization's underlying data repository. Upon completion of this review, the organization's detailed evaluation form and supporting statements will be posted on the CVE Web site for public review and use, along with MITRE's concurrence with the organization's statement. MITRE then provides the organization with the special CVE-compatible logo and formally gives them permission to use the CVE-compatible logo and term "CVE-compatible."

While the second phase takes more effort than the first phase for both the submitting organization and MITRE, it has been designed to minimize the expense to both. This approach avoids an evaluation process that would make it too expensive for freeware or smaller software vendors to obtain compatibility. By using the questionnaire and statement of compatibility, the level of effort is kept reasonable, while making a good effort to verify that the submitting organization properly understands and correctly implements the CVE compatibility requirements. The publication of the organization's statement on the CVE Web site allows end users to compare how different products satisfy the requirements and then the market can then decide which specific implementations are best.

MITRE started internal testing for the second phase of the CVE compatibility assessment process in February 2002. A "beta test" will then be conducted with a small number of organizations in the March/April time-frame, to be followed by a public roll-out.

Growth of CVE-Compatible Products and Services

The list of organizations declaring CVE-compatible products and services is continuously expanding and is international in scope. As of early May 2002, 55 organizations are working toward compatibility. For a current list, visit the CVE Web site at (http://cve.mitre.org/compatible/).

The number of products and services that are working toward CVE compatibility has grown significantly over time. In October 1999, 15 products intended to be CVE-compatible, six months later, the number had doubled to 30, exceeding 50 by July 2001. After an increase in activity in recent months, there are 81 products or services on the way to CVE compatibility as of May 2002. 27 other organizations are working on declarations for 55 additional products or services.

CHALLENGES AND OPPORTUNITIES

As CVE moves forward, it faces a variety of challenges and opportunities. Challenges include renumbering the CVE List; identifying the proper scope for the CVE List; and addressing the impact of vulnerability disclosure practices on CVE accuracy (including vendor acknowledgement and replication of the vulnerability). At the same time, opportunities include analyzing vulnerability causes, improving vulnerability testing methods and veracity, facilitating large-scale quantitative comparisons of security tools and databases, filling in some gaps in research (such as analysis of configuration problems and developing a low-level taxonomy of vulnerabilities), and delivering real improvements in the way organizations manage the risks from vulnerabilities and exposures.

Challenges in the Current Naming Scheme

The current naming scheme allows humans to easily distinguish between CVE candidates and entries (CVE-YYYY-NNNN vs. CAN-YYYY-NNNN). This distinction was chosen early in the CVE Initiative, partially based on how names are handled in other fields. However, CVE names are normally not considered atomic in data processing operations, and as such they may not be found easily by most search mechanisms. Also, the differing candidate and CVE numbering schemes cause maintenance and search problems.

Search engines may separate the name into three different terms (CAN, YYYY, NNNN) because the hyphen is sometimes considered a word separator, which can make it difficult to easily find information on the Internet using CVE names. In other cases, a search engine may need to be modified to quote the hyphen parts of the CVE name. Finally, the encoding of the year in the name may cause some problems with misuse, as it does not necessarily reflect the year in which the vulnerability was first publicized. In addition, the sequence number within the name may represent a small information leak if a candidate number is reserved for an issue long before the issue is made publicly known.

The differences between the candidate name and the CVE entry name can be difficult to manage. For example, when a candidate becomes an official entry, all CVE-compatible vendors need to update their databases to convert the candidate number to a CVE number, which can be labor intensive. In addition, users might still search for the candidate number instead of the CVE number; and some CVE-compatible products or services may not find the associated CVE entry if the user uses the candidate number in the search. To avoid this problem, each CVE-compatible product/service would need to implement a specialized function. Some omit the CAN- and CVE- prefixes outright, but this prevents a user from knowing whether the item is a candidate or an entry. The CVE Web site handles these discrepancies flexibly, but it required specialized code. Many CVE-compatible tools are not as flexible, and such a capability is not required because CVE compatibility does not require the use of candidates.

A solution would be to construct the CVE names in a way that minimizes these types of implementation problems. Using just a number would not be suitable, because numbers are so commonly used in so many databases and search engines that it could be difficult to properly distinguish a CVE number from other numbers. A scheme in which a symbol (CVE) is prepended to a number (e.g., CVE12345) could work better. If such a scheme is adopted, then the status of a CVE item—whether candidate or entry—could be noted as a separate field in CVE.

While a change to the naming scheme may provide substantial benefits, the utility of CVE would be lost if the names change too often. CVE-compatible vendors will incur high maintenance costs if and when CVE moves to a new naming scheme. Educating the public will be an additional challenge. Therefore, MITRE and the CVE Editorial Board must give strong and thoughtful consideration to any new scheme. The naming scheme should only be changed once, and there should be a period of time in which the original scheme is still supported.

Scope of the CVE List

The scope of the CVE List has been discussed and debated many times during the evolution of CVE. The discussion has generally focused on two questions:

- What types of security issues are included on the list?
- What type of information is included with each issue, and the format of CVE information?

Not only do people define "vulnerability" differently, which will impact what would or would not be included on the CVE List in and of itself, but they also have different ideas regarding which types of issues should be included on the CVE List. Some of these issues are formalized in content decisions (prefaced by "CD:").

- Vulnerabilities and exposures in beta software (CD:EX-BETA). These types of vulnerabilities are reported fairly often, but it is sometimes argued that beta software is expected to be buggy. In general, such vulnerabilities are excluded from CVE, with the following exceptions: (1) if the software is in wide distribution; or (2) if the software is consistently released in beta versions instead of final versions, e.g., the ICQ program.
- Vulnerabilities and exposures in online services such as free Web-based mail services, online banking, and e-commerce, etc. (CD:EX-ONLINE-SVC). Such problems are normally addressed with a single fix on the server, by the service provider, and do not require any action by its customers.
- Problems in a network client that cause a denial of service whose scope is limited to the client, which can be addressed by restarting the client, and which can only be exploited by a passive attack (CD:EX-CLIENT-DOS). For example, if a Web browser cannot handle a certain sequence of characters, but the problem can only be triggered by enticing a user to visit a particular Web site and it only causes the client to crash, then that issue would not be added to CVE.
- Malicious code such as viruses, worms, and Trojan horses (this category excludes back doors that were deliberately inserted by the developer). Technically, the presence of such malicious software satisfies CVE's definition of a vulnerability. However, attempting to identify and catalog all malicious code would expand the size of CVE significantly, making it unusable for too many people. In addition, it is believed that defining standard names for malware is best left to the anti-virus community.
- Vague reports of vulnerabilities, even in vendor advisories (CD:VAGUE).
- Issues that are related to security policy violations. Policies such as minimum password length and password aging, approved services, and conformance to specific software versions vary across each

enterprise, so it is difficult to create CVE items that try to capture such policies. Insecure configurations often fall into this category.

- Issues that are not necessarily proven to be "exploitable." For example, many Linux vendors release an advisory for an issue that may have security implications, even if an exploit is not known to exist. This often happens with buffer overflows, format string vulnerabilities, and signedness errors.
- Issues that are related to intrusion detection "events" that are not easily described in terms of vulnerabilities or exposures, e.g., port scanning.

One difficulty of these decisions is that some vulnerability scanners, intrusion detection systems, databases, and services may identify some of the security issues that fall into one or more of the above categories of items. Some end users may also wish to see some of these types of problems addressed by CVE. For example, one of the most frequently asked questions is whether CVE is devising a standard name for viruses. (Many end users have had difficulty dealing with viruses that have multiple names from different vendors.)

In most of these "exception cases," it has not yet been decided whether these types of security problems will be included or excluded from the CVE List. These content decisions (which are further described later in this paper) are periodically discussed by the Editorial Board. MITRE typically creates candidates for beta software, client-side DoS, and vague vulnerability reports. However, these candidates are "labeled" with the associated draft content decisions, and they will not be accepted as official entries until sufficient discussion has taken place by the Editorial Board and the content decisions have been sufficiently evaluated for completeness and repeatability. For intrusion detection events, MITRE is creating the Common Intrusion Event List (CIEL), which is described elsewhere in this paper.

The second area of debate about the scope of the CVE List focuses on the type of information that should be included with each issue, and the format of CVE information. CVE entries currently have three fields: name, description, and references. Candidates are included with additional information such as votes from Editorial Board members and the phase, which identifies how far the candidate has progressed through the review process. End users of CVE sometimes ask for additional fields beyond what is currently provided, including risk level, operating system, product vendor, fix information, and greater levels of detail in the descriptions. Such information is not required for the purpose of naming vulnerabilities, but the request for this additional information does indicate that some consumers wish to use CVE as a vulnerability database, or they want an easier way to identify the set of CVE names that they care about. There are two main concerns with respect to making this information available: (1) it increases the workload on MITRE and the Editorial Board, and (2) it would expand CVE's scope more directly in competition with commercial security vulnerability database vendors.

While MITRE has decided not to adopt these previous types of suggested additions to the information in the CVE List, in other cases, MITRE is considering making available additional information that is specifically related to how CVE content is managed. For example, candidates include an "analysis" field that describes how content decisions were applied (e.g., why a particular level of abstraction was chosen), how vendor acknowledgement was determined, and other information that may indicate why a candidate was created in the way it was. Other information that is included is a reference to the particular content decisions that affect the candidate, the date that the vulnerability was publicly announced, what specific modifications were made to the candidate, whether the vendor has acknowledged the problem, and the dates of each phase that the candidate has reached (e.g., proposal, modification, and interim decision).

Other users of CVE would like to obtain more precise change logs for each candidate or entry. Some of this information is made available to Editorial Board members for voting purposes. Because voting ballots appear in the Editorial Board mailing list archives, some of the information is publicly viewable, but it cannot be extracted easily. However, this information would be useful to a certain portion of CVE users, such as those who may want to know why a candidate that has sufficient ACCEPT votes has not been promoted to an official entry. There are plans to make some of these fields more easily accessible in the future.

Labeling each candidate or entry with a “confidence level” that represented a level of certainty that the report was correct was also considered. Some candidates identify vulnerabilities in uncommon software, which are reported by researchers who are unknown to the voting Board members. Subsequently, there may not always be a strong level of confidence that the issue is real or accurately described. Confidence is now “encoded” within the recommended voting guidelines for when Board members can ACCEPT a candidate, but it was decided that an overt and separate field would not be created.

Another request that is received fairly often is to provide the CVE List as an XML document. Work in that direction has started but is not complete as of this writing.

CVE has also been approached about translating the CVE List and CVE Web site into other languages. While interested in supporting the use of CVE by groups that do not have English as their native language, CVE’s resources will not allow such efforts by CVE. However, by the time this paper is published, a Chinese translation of the CVE Web site, including a translated version of the CVE List and candidates will be available on a site hosted by another organization. A licensing mechanism and coordination process was devised so that other organizations that are interested in hosting similar sites in other languages can be accommodated. CVE’s main focus in these translation arrangements is to ensure the quality and integrity of the CVE Initiative while broadening its international reach.

Addressing the Needs of IDS Tools with CIEL

Many events that are detected by IDSs do not have a clear association with vulnerabilities or exposures, such as port mapping, protocol decodes, failed binary integrity checks, and generic attack signatures. For cases in which an event overlaps CVE (e.g., an attempt to exploit a specific vulnerability), the CVE descriptions focus on the nature of the security problem as opposed to how it may be exploited. A number of Editorial Board members and others involved in IDS work have expressed the desire to have CVE encompass all IDS events.

MITRE is currently building a draft list for IDS events, referred to as CIEL (Common Intrusion Event List, pronounced “seal”), that is sometimes informally described as “a CVE for intrusion detection.” It is intended to provide a naming scheme for all network- or host-based events that may be useful in detecting intruder activities, but are not directly associated with CVE items. MITRE is monitoring the efforts of the IETF Intrusion Detection Working Group (IDWG) to identify areas of overlap with CIEL. The IDWG is addressing the larger needs for information exchange across IDSes, but CIEL could be used to satisfy the IDWG’s requirement for a common attack name.

Several assumptions will be guiding the development of CIEL: there is a wider variety of IDS events than vulnerabilities, there is more variety across IDSs in the level of abstraction (or level of detail) than there is in vulnerability scanners and databases, and there is not much well-defined and commonly accepted terminology in the IDS arena.

In early 2002, MITRE plans to create a CIEL working group under the CVE Editorial Board. Discussions will be held on a separate mailing list. As of this writing, MITRE is still expanding the Editorial Board to include other members of the CIEL working group.

Managing Risk with CVE Compatibility
The increase in CVE-compatible products and services can change the way that organizations use security tools and data sources to address their operational security posture. For example, an organization can use CVE-compatible products and services to improve its response to security advisories. CVE-compatible advisories include CVE entries of vulnerabilities that scanners can check for, and an IDS can be examined for appropriate attack signatures for the vulnerability described in the advisory. The incorporation of CVE names and CVE-compatible products and services provides a more structured and predictable process for handling advisories than most organizations currently possess.

Along similar lines, when a group of concerned security professionals last year composed a list of the 10 most common critical Internet security threats, they included CVE names for them [6]. Orchestrated by the SANS Institute, the effort represented the consensus of a wide variety of security experts. To help ensure specificity and make the recommendations actionable, each suggestion included the appropriate CVE names, totaling 68, and detailed the specific issue areas for a variety of platforms and products. The recent update to the SANS list [7], which is now co-sponsored by the FBI, has grown to a list of the 20 most common, critical Internet security threats and now includes 125 CVE names.

Additionally, as shown in Figure 4, compatible products and services can be used by an organization to check over an ongoing attack with its CVE-compatible IDS system (A). In a CVE-compatible IDS, specific vulnerabilities that are susceptible to the detected attack are provided as part of the attack report. This information can be compared against the latest vulnerability scan by a CVE-compatible scanner (B) to determine whether the enterprise has one of the vulnerabilities or exposures that can be exploited by the attack. If it does, a CVE-compatible fix database at the vendor of the software product or a vulnerability Web site (C), can identify the location of the fix for a CVE entry (D), if one exists.

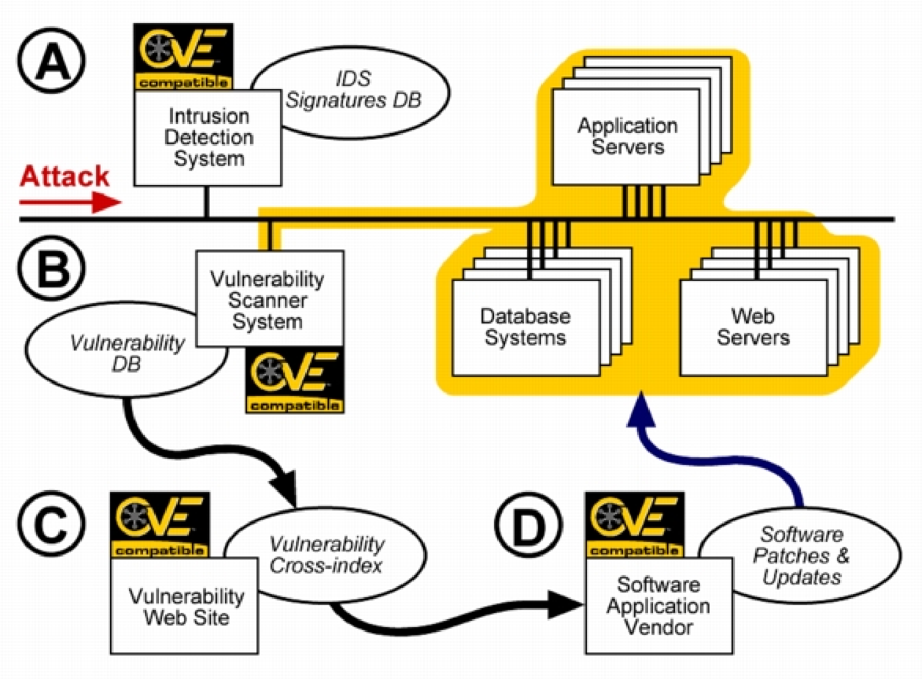

Figure 4. A CVE-Enabled Process

In addition, for systems that an organization builds or maintains for customers, CVE-compatible advisories and announcements can help directly identify any need for software fixes from the commercial vendors of those systems. For security issues in software that is distributed by multiple vendors, CVE names can help users to determine when different advisories are referring to the same vulnerability [8].

SUMMARY OF PROGRESS

Here is one way of looking at progress against the CVE strategy:

- CVE is gradually approaching the goal of uniquely naming every publicly known security relevant software mistake. More than half of all known software mistakes are now either included on the CVE List or are under review.
- CVE names are now regularly included in ISS X-Force, CERT/CC, Microsoft, and Red Hat advisories, and have been included in advisories from Rain Forest Puppy, BindView, COMPAQ, SGI, IBM, and HP on a less consistent basis.
- CVE usage in information security products and services now stands at over 100 that are either available or in development, with more being announced regularly.
- CVE usage has been included in a recent draft recommendation from the United State National Institute of Science and Technology (NIST).
- Various trade journals have started using support for CVE names as a review item in articles.
- For two years in a row the SANS Top Ten guidance (now co-issued by the United States Federal Bureau of Investigation [FBI]) has included CVE names.
- The CVE e-newsletters are subscribed to by over 2100 different organizations from over 70 countries worldwide, and the CVE Web site is being visited from individuals in over 125 countries on a regular basis.
- Of the dozen companies this has been discussed with, several are considering adding CVE name support to their fix-it sites and update mechanisms.

Acknowledgment

The summary work contained in this article was funded by the MITRE Corporation. It is based on the composite effort of all those working on the Common Vulnerabilities and Exposures Initiative, including but not limited to the CVE Editorial Board, the CVE Advisory Council, and CVE-compatible vendors.

References

1. D. E. Mann and S.M. Christey, "Towards a Common Enumeration of Vulnerabilities," *2nd Workshop Research with Security Vulnerability Databases*, Purdue University, West Lafayette, Ind., 1999; http://cve.mitre.org/docs/cerias.html (current Mar. 2002).
2. R. A. Martin, "Managing Vulnerabilities in Networked Systems," *IEEE Computer Society's Computer Magazine,* Vol. 34, No. 11, November 2001; http://www.computer.org/computer/co2001/ry032abs.htm (current Mar. 2002).
3. J. Forristal and G. Shipley, "Vulnerability Assessment Scanners," *Network Computing,* 8 Jan. 2001; http://www.networkcomputing.com/1201/1201f1b2.html (current Mar. 2002).
4. P. Mueller and G. Shipley, "To Catch a Thief," *Network Computing,* 20 Aug. 2001; http://www.networkcomputing.com/1217/1217f1.html (current Mar. 2002).
5. A. Saita, "CVE-Use Recommendations Open For Comment," *Security Wire Digest,* Vol. 4, No. 9, 4 Feb. 2002; http://www.INFOSECURITYMAG.COM/digest/2002/02-04-02.shtml#1b (current Mar. 2002).
6. W. Jackson, "Top 10 System Security Threats Are Familiar Foes," *Government Computer News,* Aug. 2000; http://www.gcn.com/state/vol6_no8/news/812-1.html (current Mar. 2002).
7. S. Bonisteel, "'Top 10' List of Net Security Holes Grows to 20," *Newsbytes.com,* 2 Oct. 2001; http://www.newsbytes.com/news/01/170713.html (current Mar. 2002).
8. Mark J. Cox, "'Chinese Whisper' security advisories," *LinuxWorld.com*, 21 Jan 2002; http://www.linuxworld.com/site-stories/2002/0121.whisper.html (current Mar. 2002).