Common Vulnerabilities and Exposures (CVE) Numbering Authority (CNA) Rules

January 1, 2018

Version 2.0

Table of Contents

1. Overview

The Common Vulnerabilities and Exposures (CVE) Program's primary purpose is to uniquely identify vulnerabilities and to associate specific versions of code bases (e.g., software and shared libraries) to those vulnerabilities. The use of CVEs ensures that two or more parties can confidently refer to a CVE identifier (ID) when discussing or sharing information about a unique vulnerability. In this way, CVE is fundamental to the vulnerability management infrastructure.

The CVE Program's primary challenge is to satisfy the demand for timely, accurate CVE assignments, while rapidly expanding the scope of coverage to address the increasing number of vulnerabilities and evolving state of vulnerability management. The CVE Program is overseen by the CVE Board (hereinafter the Board). To address CVE's scalability challenge, the Board determined that the CVE Program must be federated and that CVE IDs should be produced both more quickly and in a more decentralized manner.

1.1. CVE Numbering Authorities (CNAs)

Operating under the authority of the CVE Program, CNAs are organizations that are authorized to assign CVE IDs to vulnerabilities affecting products within their distinct, agreed upon scope, for inclusion in first-time public announcements of new vulnerabilities. These CVE IDs are provided to researchers, vulnerability discoverers or reporters, and information technology vendors. Participation in this program is voluntary, and the benefits of participation include the ability to publicly disclose a vulnerability with an already assigned CVE ID, the ability to control the disclosure of vulnerability information without pre-publishing, and notification of vulnerabilities in products within a CNA's scope by researchers who request a CVE ID from them.

1.2. Federated CNA Structure

In a federated CNA structure, CNAs are categorized as Primary, Root, and Sub-CNAs (or just "CNAs", generically). Multiple Sub-CNAs may operate under the oversight of a Root CNA, while the Root CNAs operate under the oversight of a single, Primary CNA or another Root CNA. Sub-CNAs only assign CVEs for vulnerabilities in their own products or their domain of responsibility, hereinafter referred to as scope. Root CNAs manage a group of Sub-CNAs within a given domain or community, train and admit new Sub-CNAs, and are the assigners of last resort (i.e., no Sub-CNA exists for the scope) within that domain or community. The Primary CNA oversees the CVE Program, coordinates Root CNAs and Sub-CNAs, trains and admits new Root CNAs and Sub-CNAs, enables Root CNAs to administer their CVE scope, and is the assigner of last resort for requesters that are unable to have CVEs assigned at the Sub- or Root CNA levels.

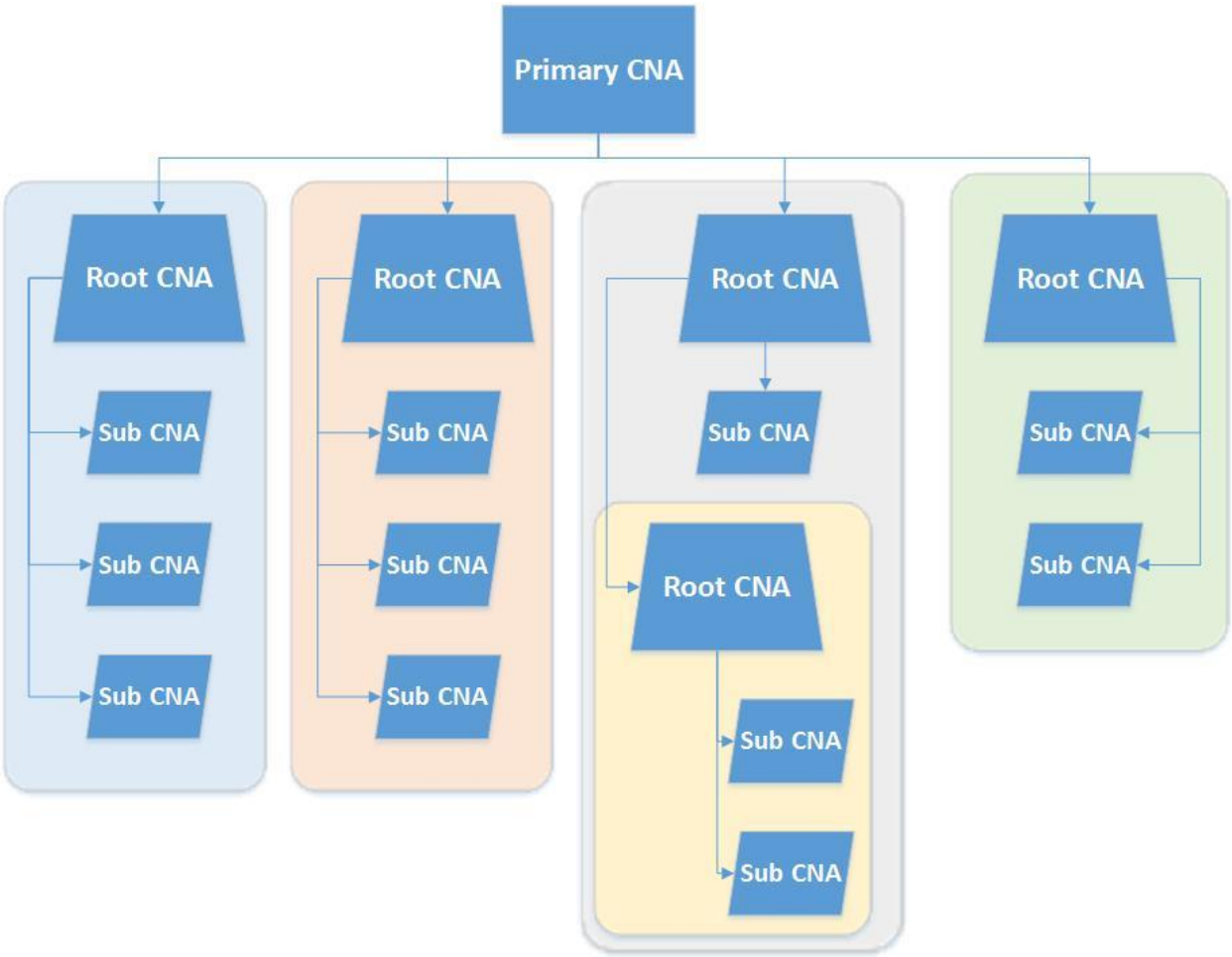

Figure 1. Federated CNA Structure

Figure 1. shows how different Root CNAs have different areas of responsibility. Each colored box is a distinctly described scope. For the gray box, part of the scope of the gray box has been delegated to a Root CNA and its Sub-CNAs, as indicated by the yellow box.

In cases where requests or issues cannot be resolved by a given CNA, the issues are escalated to the next higher-level CNA. (Examples of such issues would be a CNA being unresponsive beyond expected timeframes or a disagreement with a CNA over whether or not an issue is a vulnerability.) Requests and issues at the Sub-CNA level can be elevated to Root CNAs, and requests and issues at the Root CNAs can be elevated to the Primary CNA. The same flow, from Sub-CNAs to Root CNAs to the Primary CNA, is followed to alert the next higher CNA when CVEs are assigned, or when reporting other programmatic data. The Primary CNA provides blocks of IDs to Root CNAs, and Root CNAs provide blocks of IDs to Sub-CNAs.

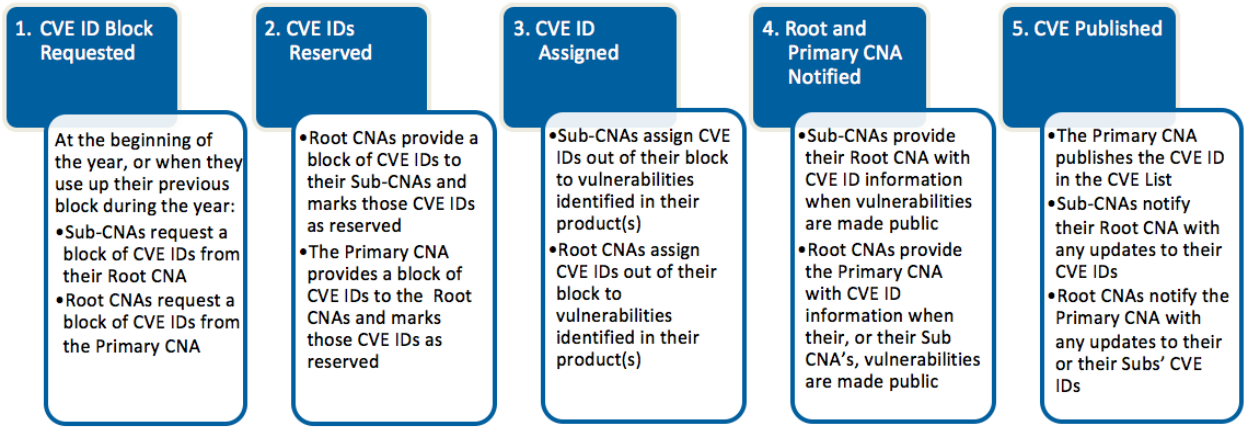

Figure 2. CNA CVE Request and Assignment Process[1]

1.3. Purpose and Goal of the CNA Rules

The purpose of establishing CNA Rules is to maintain consistency in the CVE assignment process and administration of the CNA program across all CNAs.

The goal of the CNA Rules is to provide the Root CNAs with the maximum flexibility to administer the CNA program within their respective communities, while also maintaining consistency in the CVE assignment process and administration of the CNA program.

The Primary CNA has the right to require remediation or impose sanctions on CNAs (of any type) who do not comply with these rules. However, Root CNAs are the main enforcement mechanism. That is, Root CNAs are responsible for enforcing the rules within their area of responsibility; the Primary CNA is the enforcement mechanism of last resort. The goal is for the Root CNAs to have the same level of enforcement ability as the Primary CNA, including remediation or sanctions, within their areas of responsibility, thereby enabling the federation of the CVE Program by implementing a de-centralized governance approach. Examples of remediation and sanctions include, but are not limited to:

- The development of training, guidance, or implementation materials for use by the CNAs;
- Retraining of CNA staff;
- Additional process documentation and reporting from a CNA;
- Reduction of the number of CVE IDs a CNA has available to assign at a time;
- Rejection of submissions; and
- Revocation of CNA status.

[1] Step 4 refers to "CVE information" that is provided to the Root CNAs and Primary CNAs. The information needed is listed in Appendix B.

The CNA rules, once adopted, will be reviewed at least annually, and more frequently based on lessons learned, if necessary.

1.4. Document Structure

This document is broken down into assignment, communication, and administration rules that apply to all CNAs, including Primary, Root, and Sub, as well as those rules specific to Primary and Root CNAs.

- Section 2: Rules for all CNAs
- Section 3: Rules for Root and Primary CNAs
- Section 4: CNA Candidate Process
- Section 5: Appeals Process

2. Rules for All CNAs

The following rules apply to all CNAs, regardless of level. They are related to assignment, communication, and administration. These rules, along with associated guideline and description documentation, create a concept of operations for all CNAs. (Note, CVE suggests, where possible and applicable, CNAs should conform to the vulnerability disclosure and vulnerability handling processes described in ISO/IEC 29147 and ISO/IEC 30111 in addition to the rules indicated below.)

All CNAs must adhere to the following rules:

2.1. Assignment Rules

1. Assign CVE IDs to security vulnerabilities in their scope as described by the CNA's Root CNA or the Primary CNA. CVE IDs should only be assigned to vulnerabilities that are or will be made public.[2] Vulnerabilities that will not be made public do not receive CVE IDs.

   Note: for a vulnerability to be considered "public", the following conditions must be met:

   - There must be a URL including information about the vulnerability accessible from the internet.
   - The Terms of Use of the website must allow the CVE List to link to the URL.
   - The document linked by the URL must contain the minimum required information for a CVE Entry (see Appendix B).

   Registration and login requirements are acceptable, but there cannot be other restrictions for accessing that content. Also, advisories that require payment for access are not considered public. That said, if you have a public advisory with the minimum required

[2] Disputes related to scope should be addressed by the next higher-level CNA first.

details with additional details available through paid access, the vulnerability is still considered public.

2. Only assign CVE IDs to security vulnerabilities when no lower level CNA exists which already covers a more constrained scope.

   Note: when assigning a CVE ID to a vulnerability in a bundled product, a CNA utilizing the bundled product in their own products may assign a CVE ID for the bundled product if:

   - The producer of the bundled product is not a CNA, and
   - The assigner coordinates with the producer of the bundled product or (if contact with the producer fails) the Root CNA for that bundled product.

3. Follow CVE counting rules established by the CVE Program as implemented by the Primary CNA. See Appendix C. This rule does not prevent Root CNAs and Sub-CNAs from establishing counting rules to augment the CVE counting rules established by the CVE Program. (Root CNAs can establish augmented counting rules for their scope, affecting all Sub-CNAs under them.) See 3.1.2.4 for communications rules related to such counting rules.
4. CNAs should update their upstream CNAs within 24 hours of the publication of a CVE ID. (The meaning of "publication" is discussed in Appendix A.)

2.2. Communication Rules

1. Provide points of contact (POCs) (e.g., email addresses, URLs, etc.) to all levels above their own.
2. Publish a disclosure (embargo) policy and a description of its scope. See Appendix H for a discussion of disclosure and embargo policies.
3. If a CNA accepts requests from parties outside the CNA, provide a means (e.g., hyperlink, e-mail) for the public to contact them regarding vulnerabilities. CNAs can also provide guidelines for how to communicate with them, such as language restrictions ("English-only", "Japanese or English", etc.). Provide the list publicly and to all levels above their own.
4. Be responsive to inquiries from all CNAs and document those interactions in some way (archiving email correspondence or tracking via a trouble ticket would be sufficient, for example).
5. When a vulnerability is reported to the CNA and a CVE ID is assigned to that vulnerability, provide the CVE ID to the reporter. This rule does not override any embargo rules established by the CNA.
6. Notify the next higher-level CNA when CVEs are assigned and the associated vulnerability is made public. (The publication of the vulnerability can be made in any language, but the CVE ID entry must include English only. References to information related to the CVE ID in non-English languages would be included in the reference list for the CVE ID entry.)

7. Provide CVE information to the next higher-level CNA when a CVE ID is assigned and the associated vulnerability made public. For new CVE IDs, this information includes, at a minimum, the CVE ID used, product, affected or fixed version, the problem type, references, and a description on a per-ID basis. When a CVE ID is updated, the CVE ID and data change must be included.
    - This information must be provided in the format described in Appendix B, which describes in detail the expected information.
    - Information submitted will be subject only to the CVE Terms of Use.[3]
    - Root CNAs will send any CVE assignment information they collect, either from their Sub-CNAs or from their own assignments, to the next level up the CNA chain.
8. Have an established distribution point for in-scope vulnerability disclosures that is freely available to the general public without restrictions. (In addition to completely open web sites, this can include websites that require registration but provide accounts for free without restriction to anyone.)
9. Publish required CVE information in a standard format and presentation style. This format and style will be determined and managed by the CVE Board.
10. If a CVE ID is being assigned to a vulnerability, the CNA MUST make a reasonable effort to notify the maintainer of the code in which that vulnerability exists. (If the CNA is assigning for a vulnerability in their own product or codebase, this is inherently done.) For example, if an operating system vendor discovers a vulnerability in a printing library they distribute, in addition to assigning the CVE ID to the vulnerability, they should attempt to contact the upstream developer. This will help avoid duplicate CVE ID assignments as well as ensure others that are affected by the vulnerability will be made aware of it.
11. A CNA must provide a URL to a list the products for which they accept vulnerability reports, which is referred to as their "scope". When defining their scope, vendors and development projects should offer a blanket statement (e.g., "All of Company X's products"), a list of specific things covered, or a list of specific things not covered (or a mix of covered and not covered). For researchers and third-party coordinators, theirs might say "we will issue CVE IDs for products or projects that we are researching unless they are otherwise covered by another CNA". This would help direct folks away from them as a source for a CVE ID in anything in particular and instead point them to the proper CNA (or up to the Primary). The published scope must be updated whenever a CNA's scope changes. Scope may change due to the introduction of new projects or products; projects or products being set to "end-of-life" status; mergers, sales, or acquisitions at a company level; or a change in process.

2.3. Administration Rules

1. Operate under the CVE Terms of Use.

[3] https://cve.mitre.org/about/termsofuse.html

2. Track and provide metrics related to responsiveness[4] and CNA performance to higher level CNAs. These metrics will be provided quarterly to the next higher-level CNA. See Appendix G for details.
3. Provide any documentation required to adjudicate disputes to the higher-level CNA.
4. Upon request by the Primary CNA or by the CNA's Root CNA, provide a list of unused CVE IDs that have been reserved by the CNA. (This will typically be done on a yearly basis for the previous year's CVE ID reservations.)

3. Responsibilities of Root and Primary CNAs

In addition to following the rules that apply to all CNAs, both Root CNAs and the Primary CNA have responsibilities related to assignment, communication, and administration that they must perform. Adjudication mechanisms described in this section are intended to empower Root CNAs to effectively address various issues as they arise within their area of responsibility, with Primary CNA involvement being the last resort.

3.1. Root CNAs

All Root CNAs must adhere to the following rules:

3.1.1. Assignment Rules

1. Request CVE ID blocks from the Primary CNA.
2. Provide CVE ID blocks to Sub-CNAs from their CVE ID block.
3. Assign CVE IDs as a CNA when necessary within its scope per the CVE counting rules when none of their Sub-CNAs cover that scope. See Appendix C for assignment rules. Alternately, if a Root CNA does not themselves assign CVE IDs, they MUST escalate CVE ID requests up to the Primary CNA (or direct those requests accordingly).
4. Address CVE assignment issues from its Sub-CNAs that require escalation.
5. Provide public documentation describing the specific process for submitting CVE assignments and other CVE requests.

3.1.2. Communications Rules

1. Notify the Primary CNA whenever Sub-CNAs are established or removed.
2. Provide a public list of POCs and web links for each Sub-CNA in the Root CNA's domain. Provide this information to the Primary CNA.

[4] Responsiveness metrics may vary by CNA as determined by the unique circumstances of the particular CNA community. The Primary CNA responds to all CVE requests within 24 hours, which may or may not be an appropriate responsiveness goal for other CNAs. The purpose of responsiveness metrics is to ensure that CNAs are responsive to various types of requests from their various communities in time frames that are appropriate for those communities.

3. Maintain a private list of individual POCs within each Sub-CNA for use by CNAs only. Provide this information to the Primary CNA.
4. Maintain a public listing of the established counting rules followed by the Root CNA and Sub-CNAs in its domain.

3.1.3. Administration Rules

1. Accept metrics reports from Sub-CNAs. See 2.3.2. The format and instructions for sending metrics are determined by the Root CNA.
2. Submit metrics from Sub-CNAs quarterly, within two weeks of the quarter, to the Primary CNA. Quarters are based on the calendar year.
3. Act as an escalation and adjudication point for issue resolution for Sub-CNAs in its domain.
4. When appropriate, apply sanctions upon any Sub-CNAs within its domain and notify the Primary CNA. The application of sanctions should occur as a last resort.
5. Facilitate the enforcement of any administrative actions taken by the Primary CNA against a Sub-CNA.
6. Follow the CNA Candidate Process described in Section 4 when adding new Sub-CNAs.

3.2. Primary CNA

The Primary CNA must adhere to the following rules:

3.2.1. Assignment Rules

1. Provide CVE ID blocks to Root CNAs.
2. Maintain the CVE List, and provide that information to the public.
3. Assign CVE IDs as a CNA when necessary, per the CVE counting rules, when no Root CNAs cover that scope. See Appendix C.
4. Act as the CNA of last resort for assignment issues that require escalation.
5. Maintain a process for rejecting unused reserved CVE IDs each year. One example process would be: at the beginning of each calendar year, CNAs must indicate to the Primary CNA which CVE IDs from the previous calendar year were not assigned to a vulnerability. Those CVE IDs that were unused would be rejected. (CVE IDs for previous calendar years can always be requested from the Primary CNA if necessary.)
6. Maintain a process for rejecting assigned-but-unpopulated CVE entries based on an expiration period. For example, that period may be “if a CVE ID was assigned two years ago but the entry for it was not populated by the assigner, the CVE ID will be rejected”. The specific time frame should be publicly documented by the Primary CNA and can be updated based on the needs of the CVE community.

3.2.2. Communications Rules

1. Provide a listing of all Root CNAs and Sub-CNAs including public points of contact and web links. Obtain this information from Root CNAs.
2. Maintain a private list of individual POCs for each Root and Sub-CNA for use by CNAs only.
3. Provide coordination of communication channels between Root CNAs.
4. Respond to inquiries by Root CNAs and Sub-CNAs in a timely manner; establish responsiveness metrics for such responsiveness.
5. Maintain a public listing of the established counting rules for the CVE Program. See Appendix C.

3.2.3. Administration Rules

1. Serve as a member, and the Board Moderator, of the CVE Board.
2. Accept metrics reports from Root CNAs quarterly, within one month of the calendar quarter.
3. Act as the final arbiter for appeals regarding CNA assignment decisions and CNA program issues.
4. Act as an escalation point for issue resolution should this process fail at the Root CNA level.
5. When appropriate, apply sanctions upon any CNA.
6. Follow the CNA Candidate Process described in Section 4 when adding new Root CNAs.

4. CNA Candidate Process

The CVE Program, through both Root CNAs and the Primary CNA, adds qualified organizations (hereinafter referred to as candidates) as CNAs through the on-boarding process described in this section. The on-boarding process is designed to set expectations for CNAs regarding the oversight and administration of CVE assignment for products within their scope.

The goals of the CNA candidate process:

1. The candidate understands its roles and responsibilities.
2. Individual members of the new CNA's team are able to perform CVE assignment and counting processes.
3. Clear communication channels exist between CNAs and the rest of the CVE Program.

4.1. CNA Qualifications

A candidate is qualified if they meet the following criteria:

1. A candidate must be interested in becoming a CNA and willing to follow established CNA rules.

2. A CNA must be
    a. a vendor with a significant user base and an established security advisory capability or
    b. an established entity with an established security advisory capability that typically acts as a neutral interface between researchers and vendors or
    c. an established bug bounty service provider or
    d. an established vulnerability research team or
    e. an independent vulnerability researcher.

    A Root CNA may be a regional coordinator (such as a Computer Emergency Response Team [CERT] or a Computer Security Incident Response Team [CSIRT]) or a domain publisher (such as an Information Sharing and Analysis Center [ISAC] representing a particular sector). A CNA may also be a mature research organization.
3. The CNA must be an established distribution point or source for first-time product vulnerability announcements (which may concern their own products). In keeping with the CVE requirement to identify public issues, the CNA must only assign CVEs to security issues that will be made public. If the CNA is disclosing vulnerabilities for products or projects not their own (and not covered by another CNA), they must consistently publish a public vulnerability announcement for each assignment. (Refer to the definition of "vulnerability" in Appendix A for clarification on what products should and should not be considered when assigning a CVE ID.)
4. The CNA should follow coordinated disclosure practices as determined by the community which they serve. Coordinated disclosure practices reduce the likelihood that duplicate or inaccurate information will be introduced into CVE.

4.2. CNA On-Boarding Process

1. A candidate may be identified by a Root CNA, the Primary CNA, a member of the CVE Board, or they may approach the Root CNA, the Primary CNA, or a member of the CVE Board to request a CNA appointment.
2. The candidate is reviewed to determine whether it is qualified by the appropriate Root CNA or the Primary CNA, hereinafter referred to as the vetting CNA, using the guidance in this section. A Root CNA is appropriate if the candidate fits within the domain of the Root CNA.
3. The vetting CNA engages the candidate and shares information about becoming a CNA, including this document.
4. The candidate assigns a primary and secondary POC for initial coordination with the vetting CNA.
5. Anyone acting in a CVE analyst capacity at the candidate's organization will be given training by their vetting CNA, which will include:
    - Examples and exercises to work through with instruction and feedback;
    - Counting rules to review and follow.

During this training, an initial block of CVE IDs will be allocated to the candidate for use with their training. This block will be allocated by the vetting CNA. The Primary CNA will provide guidance and templates to assist with the creation of examples and exercises.

6. The candidate will document how CVE processes will be integrated into their operations.
    - The candidate's documentation will include how they will process new requests for CVE IDs, internally and externally. If the candidate will process external CVE assignment requests, processes to submit requests will be documented for public release.
    - All documentation will be shared with the vetting CNA and may also be shared publicly by the candidate.
7. The vetting CNA will review the candidate's documentation and work with the candidate to address any issues in their processes that may conflict with the established CNA rules.
8. The vetting CNA allocates the candidate a block of CVE IDs to assign.
9. The candidate's POCs are added to the appropriate communications channels.
10. After successfully completing the above, required steps, the candidate enters operational mode and is now considered a CNA. If the CNA was added by a Root CNA, the Root CNA notifies the Primary CNA.
11. The Primary CNA updates public documentation to include the new CNA and makes public announcements introducing the new CNA.

Any changes in a CNA's program, including staff changes or process changes, must be documented and shared with the CVE Program through a CNA's Root CNA or the Primary CNA.

5. Appeals Process

For situations where CVE assignment decisions are disputed, or where there is a disagreement between Root CNAs or between a Root CNA and one of their Sub-CNAs, the following process should be followed to resolve the issues:

1. The party seeking to appeal a decision made by a Root CNA, or resolve a disagreement between Root CNAs, contacts the Primary CNA at cve@mitre.org and asks for arbitration of the appeal.
2. The Primary CNA sets expectations for when a timely resolution may be available. Appeals of time-sensitive issues are prioritized, as determined by the Primary CNA.
3. The Primary CNA contacts the appropriate entities to collect information relevant to the issue. The CNAs involved in the dispute provide documentation per the rules established in this document. The Primary CNA may also engage the CVE Board for their consideration of the issue.
4. The Primary CNA communicates its decision to all relevant parties once the disagreement or appeal has been fully considered. This result is final.

Appendix A Definitions

These definitions give CNAs an understanding of terms that are used throughout the CVE Program. Whenever anyone within the CVE Program uses these terms in the context of CVE operations, CNAs should interpret the meanings of those terms based on these definitions.

An **access vector** or **extent** describes how a vulnerability may be exploited. Examples of this include a local exploit (by having a presence on the system), a physical exploit (by having physical access to the system), or a network-based exploit (where the vulnerability can be exploited through a network connection).

A **bundled product** is a product distributed with another product. In CVE terms, the developer of a bundled product included in another vendor's product is considered an **upstream** developer compared to the vendor distributing the bundled product, who is considered to be a **downstream** developer. For example, if Apple includes the apache server in Mac OS X, the apache server is considered a bundled product, and Apple is considered a downstream developer for the apache server, whereas Apache is considered the upstream developer for the apache server.

A **bug** is the flaw or design oversight leading to a potential vulnerability.

A **codebase** is a software component that is shared among multiple products.

A **configuration issue** is where a purposeful customization in the behavior of software results in an unintended state.

An **executable** file causes a computer to perform indicated tasks according to encoded instructions, as opposed to a data file that must be parsed by a program to be meaningful.

Hardware is defined interconnected electronic components which perform analog or logic operations on received and locally stored information to produce as output or store resulting new information or to provide control for output actuator mechanisms.
Electronic hardware can range from individual chips/circuits to distributed information processing systems. Electronic hardware is composed of hierarchies of functional modules which inter-communicate via precisely defined interfaces. (Definition is from https://en.wikipedia.org/wiki/Electronic_hardware)

A vulnerability is **independently fixable** when it can be fixed such that it does not fix any other reported vulnerabilities (i.e., is a separate code fix a possible approach to fix the vulnerability in question).

The **Primary CNA** operates the CVE Program, manages Root CNAs, trains and admits new Root CNAs, and is the assigner of last resort for requesters that are unable to have CVEs assigned at the Sub- or Root CNA levels.

A **problem type** is defined by a combination of attack model (e.g., symlink attack) and the type of mistake that causes the vulnerability (e.g., the product does not properly check permissions).

A product is **publicly available** when anyone can purchase or obtain legitimate access to it. This includes freeware, shareware, open source, and commercial products.

A vulnerability is **publicly known** when the issue has been published or divulged publicly (or is scheduled to be published by a researcher or vendor who has been in communication with the CVE Team regarding the issue).

Root CNAs manage a group of Sub-CNAs within a given domain or community, train and admit new Sub-CNAs, and are the assigners of last resort within that domain or community.

The **Scope** of a given CNA is its products or domain of responsibility.

A **software package** is a collection of separate, self-contained software components that are distributed as a single, monolithic object.

A **software product** is a collection of installable software distributed under a unique name by a particular vendor or development project.

A **software version** is a unique name for a particular revision of computer software. This includes commit IDs and other versioning identifiers. Within the CVE process, the specific version or versions affected by a vulnerability are key factors in the counting process.

Sub-CNAs assign CVEs for vulnerabilities in their scope, and operate under the management of Root CNAs.

The **U.S. Information Technology (IT) Sector** is defined by a set of functions performed by the entities that comprise the sector. Those functions provide: a) IT products and services; b) incident management capabilities; c) domain name resolution services; d) identity management and associated trust support services; e) Internet-based content, information, and communications services; and f) Internet routing, access, and connection services.

A **vulnerability** in the context of the CVE Program is defined by the Counting Rules as listed in Appendix C. In general, a vulnerability is defined as a weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, OR availability. Mitigation of the vulnerabilities in this context typically involves coding changes, but could also include specification changes or even specification deprecations (e.g., removal of affected protocols or functionality in their entirety).”

CVE ID Lifecycle Terminology

A CVE ID is considered assigned when any CNA has assigned that CVE ID to a vulnerability.

A CVE ID is considered reserved when

- The CVE ID has been allocated to a CNA for their potential use, or
- The Primary CNA has assigned a CVE ID to a vulnerability for a non-CNA CVE ID requester.

  Note: A reserved CVE ID may be in the reserved state without having been assigned. See https://cve.mitre.org/about/faqs.html#reserved_signify_in_cve_id for more discussion on reserved CVE IDs.

A CVE ID entry is populated when the description, references, and other meta information about the entry is added to the CVE List. A CVE ID entry marked as "reserved" or "rejected" is not considered populated.

A CVE ID is published when the CVE ID itself is used in a public forum outside of CVE. A CVE ID entry may not be populated when someone publishes the CVE ID. The CVE ID entry will not be populated until the minimal required information for CVE assignment has been communicated to the Primary CNA. Note: if the CVE ID is used publicly before anyone has updated the Primary CNA with the CVE ID information, the CVE ID entry may show as "reserved" or nonexistent in the CVE List until the Primary CNA has been updated.

Note the distinction between CVE IDs that are "published" versus those that are "populated". Published CVE ID exist outside of the CVE list. Populated CVE IDs are those that are found within the CVE List. These two terms help make the distinction between what is and is not included within the official CVE List.

Appendix B CVE Information Format

CNAs must provide CVE assignment information to the CNA level above them using one of the following formats. The use of these formats facilitates the automation of CVE assignment.

1. The preferred format for submitting CVE assignment information is using the JSON schema described here: https://github.com/CVEProject/automation-working-group/blob/master/cve_json_schema/DRAFT-JSON-file-format-v4.md

   Note: The JSON description includes guidance on entry formatting that is unique to the JSON schema and supersedes any formatting guidance listed in this Appendix.

2. In a flat file, use this format.

   [CVEID]:
   [PRODUCT]:
   [VERSION]:
   [PROBLEMTYPE]:
   [REFERENCES]:
   [DESCRIPTION]:
   [ASSIGNINGCNA]:

3. In a Comma Separated Values (CSV) file, each row should include each of these columns with CVE ID as a primary key.

There are no format limitations on the actual data, which allows for flexibility across products that may have unusual versioning or differing definitions, such as what a "problem type" means. The only exception to this is that references must be URLs. With or without this technical standard, the information referenced by each field is required for assigning a CVE. In all cases, the content included in CVE submission must be germane to the vulnerability. The Primary CNA reserves the right to modify or reject content included in CVE assignment if it is deemed inappropriate by the Primary CNA. Any information submitted as part of a CVE must be submitted in English, though CVEs may reference content in any language.

Where applicable, make use of industry standards when describing vulnerabilities.

As a general guideline, [PRODUCT] should include the vendor, developer, or project name as well as the name of the actual software or hardware in which the vulnerability exists.

[VERSION] should include the version, date of release, or whatever indicator that is used by vendors, developers, or projects to differentiate between releases. [VERSION] can be described with specific version numbers, ranges of versions, or “all versions before/after” a version number or date.

As mentioned above, [PROBLEMTYPE] can include an arbitrary summary of the problem, though Common Weakness Enumerations (CWEs) are an excellent standard to use in this field.

[REFERENCES] should be URLs pointing to a world-wide-web-based resource. For CSV and flat-file formats, they should be separated by a space. References should point to content that is relevant to the vulnerability and include at least all the details included in the CVE entry. Ideally, references should point to content that includes the CVE ID itself whenever possible. References must also be publicly available, as described in Section 2.1.1.

The [DESCRIPTION]: field is a plain language field that should describe the vulnerability with sufficient detail as to demonstrate that the vulnerability is unique. The required information listed above should be included in the [DESCRIPTION], as well as other details the author feels are relevant or necessary to show uniqueness.

Specifically, the [DESCRIPTION]: field could also include:

- An explanation of an attack type using the vulnerability;
- The impact of the vulnerability;
- The software components within a software product that are affected by the vulnerability; and
- Any attack vectors that can make use of the vulnerability.

Descriptions often follow this template:

[PROBLEM TYPE] in [PRODUCT/VERSION] causes [IMPACT] when [ATTACK]

where impact and attack are arbitrary terms that should be relevant to the nature of the vulnerability.

The [ASSIGNING CNA]: field should include the name of the assigning CNA. CNAs should use a consistent name to facilitate searches for CVE IDs that originate from them.

Following is an example of the reporting format in use. In this case, the Sub-CNA "BigCompanySoft" is assigning a CVE ID to versions of their product.

[CVEID]: CVE-2016-123455

[PRODUCT]: BIGCOMPANYSOFT SOFTWARE PRODUCT

[VERSION]: All versions prior to version 2.5

[PROBLEMTYPE]: Arbitrary Code Execution.

[REFERENCES]: http://bigcompanysoft.com/vuln/v1232.html

[DESCRIPTION]: CoreGraphics in BIGCOMPANYSOFT SOFTWARE PRODUCT before 2.5 allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a crafted BMP image.

[ASSIGNINGCNA]: BigCompanySoft

JSON Submission and Storage Format

The JSON schema will be reviewed periodically. The review cycle will follow a schedule similar to this example:

First 30 days (September)

- Open comment period including Board and CNAs.
- One or two Automation WG calls specifically set aside for discussion of proposed changes.
- At the end of this period, no additional suggestions will be included in the revision cycle.

Next 30 days (October)

- The community will work in one-week sprints (WG meetings and mailing list discussions) with a subset of the proposed revisions discussed during each sprint. Each subset is only to be discussed during that sprint.
- There will be four total sprints (making this part a four-week process).
- At the end of a sprint, if something was not resolved or discussed, it will not be included in the revision.
- When something is resolved, any changes based on it are included within the development branch at that time.

At the end of all sprints, the JSON format will be finalized and sent to the Board for approval.

Next 60 days (November and December)

- CNAs can use the development branch for testing new features and changes

The new JSON format would take effect on January 1 of the next year. This will give CNAs two months to implement any changes to their processes that become needed after the JSON format revised.

Appendix C Common Vulnerabilities and Exposures (CVE) Counting Rules

C.1. Purpose

This appendix provides the definition of, and guidelines for, the CVE vulnerability counting process. These guidelines should be used by any CVE Numbering Authorities (CNAs) who participate within the CVE Program.

C.2. Introduction

The nature and accuracy of the counting process underpins the value of a CVE. Correct counting reduces the likelihood of duplicate CVE IDs being assigned to a single vulnerability. Also, some reports of vulnerabilities may confuse or conflate multiple, separate software problems, and the counting process helps to differentiate between those vulnerabilities that are unique.

CVE IDs can be assigned to vulnerabilities in any code-based entity or standards upon which code-based entities are designed. This can include software, shared codebases, libraries, protocols, standards, hardware (e.g., firmware or microcode), hardware platforms, file formats, or data encodings.

C.3. Vulnerability Report

The following decision trees should be used when receiving a report for a single or multiple vulnerabilities. The decision trees are meant to be used together and are to be followed from top to bottom.

C.4. Counting Decisions

Use the following decision tree to determine how many vulnerabilities there are in a report.

NOTE: It is intended that one of CNT2.1 or CNT2.2 be completed, but not both (i.e., A CNA has the flexibility and choice to use the claim-based or security model-based inclusion decision).

CNT1	**Independently Fixable**: For each reported bug, determine if it can be fixed independently of the other bugs (i.e., a code fix can be created to fix only the bug in question)? A common indicator of independently fixable would be that the vulnerability affects a different version of the product than the other reported vulnerabilities. Note that this does not mean that the bugs are fixed independently; only that if the vendor chose to the bugs could be fixed independently.
	• If a vulnerability can be fixed independently from the others, go to CNT2. • If the vulnerabilities cannot be fixed independently, group the bugs together and go to CNT2. • If it is not clear whether the vulnerabilities can be fixed independently, group the bugs together and go to CNT2.

CNT2	**Vulnerability**: For each bug, apply the following decisions to determine if it is a vulnerability. If the bug does not meet the criteria, can you combine it with one or more other bugs to meet the following criteria (e.g. combine a permissions issue with predictable file name and a race condition to generate a symbolic link attack)?

CNT2.1	**Vendor acknowledgment**: Does the affected vendor acknowledge the bug as a vulnerability and does it also acknowledge a negative impact on security? Examples of negative impact could include; code execution, providing the attacker with extra privileges or information, causing a denial of service, etc. (i.e., see the definition of a vulnerability as defined by the CVE Program).
	• Yes: Continue to CNT3 • No: Continue to CNT2.2A OR CNT2.2B • Not sure: Continue to CNT2.2A OR CNT2.2B
CNT2.2A	**Claim-based**: Does the vulnerability report provide a demonstrated negative impact for the bug? Examples of negative impact could include; code execution, providing the attacker with extra privileges or information, causing a denial of service, etc. (i.e., see the definition of a vulnerability as defined by the CVE Program).
	• Yes: Continue to CNT3 • No: Do not assign a CVE ID. • Not sure: Continue to CNT3
CNT2.2B	**Security model-based**: Does the vulnerability report provide evidence of a mistake or design oversight in software that violates the security policy of the system?
	• Yes: Continue to CNT3 • No: Do not assign a CVE ID. • Not sure: Continue to CNT3

CNT3	**Shared Codebase, Library, Protocol**: Does the vulnerability affect a shared codebase, library, or protocol implementation issue? Note: consultation with the Root CNA is recommended when the vulnerability affects software covered by other CNAs

	• **For Shared Codebase**
	  ○ Affects a single product, assign one CVE ID and continue to INC1.
	  ○ Affects the same code in multiple products, assign a CVE ID to each affected codebase and continue to INC1.
	  ○ Affects multiple products but with different code, assign a CVE ID to each product and continue to INC1

	◦ Not sure or undefined, assign a CVE ID to each product and continue to INC1. • **For Libraries, Protocols, or Standards** ◦ If there is a way to use the library, protocol, or standard without being vulnerable, then assign a CVE ID to each affected codebase or product and continue to INC1. ◦ If the using the library, protocol, or standard requires the product to be vulnerable, assign a single CVE ID and continue to INC1. ◦ Not sure, assign a CVE ID to each affected codebase and continue to INC1.

C.5. Inclusion Decisions

Use this decision tree to determine if a vulnerability should be assigned a CVE ID (i.e., Does vulnerability meet the CVE inclusion decisions?). When multiple vulnerabilities are reported, this decision tree will need to be repeated for each issue.

NOTE: The Inclusion Decisions table describes an order to the inclusion decisions. However, so long as the vulnerability meets all of the conditions, it does not matter which order the decisions are executed in.

INC1	**In Scope of Authority**: Does the vulnerability report fall into the scope of authority for the CNA. CNAs can only assign CVE IDs to vulnerabilities that are within their scope of authority as defined by their root CNA.
	• Yes: Continue to INC2. • No: DEFER to appropriate CNA or Root CNA • Not sure: CONSULT Root CNA
INC2	**Intended to be Public**: Is the vulnerability report or the issue described currently published publicly or intended to be published to a publicly available location in the future? CVE IDs are intended to be public information and are not assigned to vulnerabilities that are intended to be private. See Section 2.1 for a description of what is considered “public”.
	• Yes: Continue to INC3. • No: Do not assign a CVE ID.
INC3	**Installable/Customer-controlled Software**: Is the vulnerability site-specific? Is it only in an online service (software-as-a-service), on a specific web site, or only offered through hosting solutions that are under the full control of the vendor? CVE IDs are assigned to products that are customer-controlled or customer-installable.

	• Yes: Do not assign a CVE ID. • No: Continue to INC4. • Not sure: Continue to INC4.
INC4	**Generally Available and Licensed Product**: Does the vulnerability affect software that is licensed and made generally available to the public? If the vulnerability only affects a version of software that was never made generally available to the publisher's or vendor's customers, the bug should not be assigned a CVE ID. CVE IDs are not assigned to bugs in malware, closed betas, commits that were fixed before a new release is issued, applications used only within a single organization (such as a unique, custom-built system).
	• Yes: Continue to INC5. • No: Do not assign a CVE ID. • Not sure: Continue to INC5.
INC5	**Duplicate**: Has the vulnerability already been assigned a CVE by you or does it already exist in the CVE List?
	• Yes: USE the existing CVE ID. • No: ASSIGN a CVE ID. • Not sure: ASSIGN a CVE ID.

Appendix D Terms of Use

LICENSE

Submissions: For all materials you submit to the Common Vulnerabilities and Exposures (CVE®), you hereby grant to The MITRE Corporation (MITRE) and all CVE Numbering Authorities (CNAs) a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute such materials and derivative works. Unless required by applicable law or agreed to in writing, you provide such materials on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

CVE Usage: MITRE hereby grants you a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Common Vulnerabilities and Exposures (CVE®). Any copy you make for such purposes is authorized provided that you reproduce MITRE's copyright designation and this license in any such copy.

DISCLAIMERS

ALL DOCUMENTS AND THE INFORMATION CONTAINED THEREIN PROVIDED BY MITRE ARE PROVIDED ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE MITRE CORPORATION, ITS BOARD OF TRUSTEES, OFFICERS, AGENTS, AND EMPLOYEES, DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Appendix E Process to Correct Counting Issues or Update CVE Entries

There are many places where the CVE ID assignment process can break down. Common causes of incorrect assignments include:

- Insufficient information, e.g., the codebase relationships are not sufficiently researched.
- Inadequate coordination, e.g., two CNAs assign separate CVE IDs without talking to each other.
- Human error, e.g., a typo in a report.

Since mistakes are inevitable, processes to correct them are necessary. The following sections describe different scenarios wherein the CVE ID assignment goes awry, and the corresponding resolution process.

In general, a CVE entry may be updated in order to:

- Add or update a reference;
- Update a description;
- Resolve the existence of a duplicate entry; or
- Reject an entry.

These updates may be initiated by:

- The CNA that assigned the CVE ID;
- A third-party with information not currently included in the CVE entry; or
- A Root or the Primary CNA resolving an issue with the CVE entry.

As part of a CNA's vulnerability management process, a CNA can choose whether they wish to vet any updates to CVE IDs that they assigned. The process for communicating those changes between CNAs and requesters will vary depending on the CNA. It is not a requirement that CNAs must vet changes to their CNA entries.

Reject: A CVE ID Should Not Have Been Assigned

There are many reasons why a CVE ID may be rejected, such as: further research determines the issue is not a vulnerability; a typo in an advisory causes the wrong CVE ID to be used; or the researcher decides to keep the vulnerability private. In these and other instances, the description for the CVE entry is updated to reflect that the CVE ID has been REJECTED and provides the reason for the rejection.

Merge: Multiple CVE IDs Assigned to One Vulnerability

The process for resolving multiple CVE IDs assigned to a single vulnerability (as defined by the counting decisions) is as follows:

1. Determine which CVE ID to associate with the issue.
2. Merge the information from the other CVE IDs into chosen CVE ID.

3. Update the CVE IDs that were not chosen with a REJECTED description that points to the chosen CVE ID as the correct one to use.

The following criteria is used to select which identifier will be associated with the issue:

1. PREFER THE MOST COMMONLY REFERENCED IDENTIFIER. This is roughly gauged by searching for all affected identifiers on a search engine and comparing results.
2. If the usage numbers of identifiers are about the same, then CHOOSE THE IDENTIFIER USED BY THE MOST AUTHORITATIVE SOURCE. The "most authoritative source" is roughly prioritized as: vendor, coordinator, researcher.
3. If the identifiers have the same level of authority, then CHOOSE THE IDENTIFIER THAT HAS BEEN PUBLIC FOR THE LONGEST PERIOD OF TIME.
4. If the identifiers have been public for the same amount of time, then CHOOSE THE IDENTIFIER WITH THE SMALLEST NUMERIC PORTION.

Note that the process described above is reserved for cases where the CVE IDs have clearly been assigned to the same vulnerability. If there is insufficient information to decide, the description of the CVE entries may be changed to indicate that they may be the same. For example, a NOTE sentence such as "This may be the same as <the-other-CVE-ID>" or "This may overlap <the-other-CVE-ID>" may be used.

Split: A Single CVE ID is Assigned when More than One is Required

The process for splitting a CVE entry into multiple CVE entries is as follows:

1. Determine which vulnerability should be associated with the original CVE ID.
2. Assign CVE IDs to the additional vulnerabilities.
3. Include a NOTE pointing to the original CVE ID in the descriptions of the CVE entries for the new CVE IDs.
4. Update description of the CVE entry for the original CVE ID with a NOTE saying that the entry has been split and point to the additional CVE IDs.

The following criteria is used to select which vulnerability is selected to be associated with the original CVE.

1. PREFER THE MOST COMMONLY ASSOCIATED VULNERABILITY. This is roughly gauged by searching for all of the vulnerabilities on a search engine and comparing results.
2. If the association number of the vulnerabilities are about the same, then CHOOSE THE VULNERABILITY WITH THE MOST SEVERE RISK. The risk for a vulnerability is determined by the CVSS score.
3. If the risks are roughly the same, CHOOSE THE VULNERABILITY WITH BROADEST RANGE OF AFFECTED VERSIONS.
4. If the vulnerabilities affect the same versions, CHOOSE THE VULNERABILITY THAT WAS DESCRIBED FIRST IN INITIAL PUBLICATION.

Dispute: Validity of the Vulnerability is Questioned

Not everyone shares the same definition of a vulnerability. One person's vulnerability is another person's security hardening opportunity, and another person's intended functionality. How does CVE deal with these differing opinions?

When an authoritative source disputes the validity of the vulnerability, "'** DISPUTED **" is added to the beginning of the description, and a short NOTE is added to the end explaining why the vulnerability is disputed. Ideally, the disputing party provides a link that can be added to the CVE as a reference, and a quote that can be used as the explanation in the NOTE. However, neither are required.

Note that marking a CVE entry as disputed is different from rejecting a CVE entry. Rejections are made because the issue is clearly not a vulnerability (it fails CNT2), the vulnerability is not made public (it fails INC2), the product isn't customer controlled (it fails INC3), or the product is not generally available (it fails INC4). Entries are disputed when there are differing opinions about it being a vulnerability or regarding the specific details of the vulnerability itself. The more binary cases of INC2, INC3, and INC4 are not things that can be disputed, per se. They either are or are not true.

Partial Duplicate

There are cases where two CVE IDs overlap in what software or hardware is affected by the same vulnerabilities. An example of this would be if CVE-2017-nnnn1 references Product1 versions 1.0, 2.0, and 3.0 and CVE-2017-nnnn2 is assigned to the same vulnerability and references Product1 versions 3.0, 4.0, and 5.0. In this situation, use the following process.

1. PREFER THE MOST COMMONLY REFERENCED IDENTIFIER. This is roughly gauged by searching for all affected identifiers on a search engine and comparing results. In our example above, CVE-2017-nnnn1 is used more often than CVE-2017-nnnn2. Therefore, CVE-2017-nnnn1 would reference versions 1.0, 2.0, and 3.0, and CVE-2017-nnnn2 would be changed to only reference versions 4.0 and 5.0. In both CVE entries, a note should be added to the effect "This CVE entry is related to [the other]."
2. If the usage numbers of identifiers are about the same, then CHOOSE THE IDENTIFIER USED BY THE MOST AUTHORITATIVE SOURCE. The "most authoritative source" is roughly prioritized as: vendor, coordinator, researcher. Again, if CVE-2017-nnnn1 is used by the most authoritative source, CVE-2017-nnnn1 would reference versions 1.0, 2.0, and 3.0, and CVE-2017-nnnn2 would be changed to only reference versions 4.0 and 5.0. In both CVE entries, a note should be added to the effect "This CVE entry is related to [the other]."
3. If the identifiers have the same level of authority, then CHOOSE THE IDENTIFIER THAT HAS BEEN PUBLIC FOR THE LONGEST PERIOD OF TIME. Again, if CVE-2017-nnnn1 has been public for the longest period, CVE-2017-nnnn1 would reference versions 1.0, 2.0, and 3.0, and CVE-2017-nnnn2 would be changed to only reference versions 4.0 and 5.0. In both CVE entries, a note should be added to the effect "This CVE entry is related to [the other]."

4. If the identifiers have been public for the same amount of time, then CHOOSE THE IDENTIFIER WITH THE SMALLEST NUMERIC PORTION. Since CVE-2017-nnnn1 uses a smaller numeric portion, CVE-2017-nnnn1 would reference versions 1.0, 2.0, and 3.0, and CVE-2017-nnnn2 would be changed to only reference versions 4.0 and 5.0. In both CVE entries, a note should be added to the effect "This CVE entry is related to [the other]."
5. If there are any disputes after this, CHOOSE THE IDENTIFIER THAT WAS POPULATED IN THE CVE LIST THE EARLIEST. Assuming CVE-2017-nnnn1 was populated earliest, CVE-2017-nnnn1 would reference versions 1.0, 2.0, and 3.0, and CVE-2017-nnnn2 would be changed to only reference versions 4.0 and 5.0. In both CVE entries, a note should be added to the effect "This CVE entry is related to [the other]."

Note that the process described above is reserved for cases where the CVE IDs have clearly been assigned to the same vulnerability. If there is insufficient information to decide, the description of the CVE entries may be changed to indicate that they may be the same. For example, a NOTE sentence such as "This may be the same as…" or "This may overlap…" may be used.

Appendix F Acronyms

Acronym	Definition
CERT	Computer Emergency Response Team
CSIRT	Computer Security Incident Response Team
CNA	CVE Numbering Authority
CVE	Common Vulnerabilities and Exposures
ID	Identifier
ISAC	Information Sharing and Analysis Center
POC	Point of Contact

Appendix G Quarterly Metrics

Per 2.3.2, every CNA must provide metrics to gauge their responsiveness to CVE requests and their general performance as a CNA. CNAs should collect and report on the following information to their Root CNA at a minimum. Root CNAs can request additional information.

For All CNAs

- Number of unique vulnerability reports received from external parties (assigned and not assigned CVE IDs)
  Rationale: This gives an idea of how much vulnerability disclosure activity there is in each CNA which can then be extrapolated to sectors or some other category of CNA.
- Average time between assignment of CVE ID and publication of CVE ID entry
  Rationale: Again, taken in aggregate, gives an idea for what the common time frames are, which can inform discussions of best practice.

For Root CNAs

- Number of times an issue was escalated to the Root CNA
  Rationale: How much of a Root CNA's time is spent dealing with escalations? Does it scale with the number of Sub-CNAs they have? Does it vary between sectors?
- Categories of escalated issues and percentage of total:
  - Dispute
  - Responsiveness
  - Misuse of CVE

  Rationale: What is the nature of the issues that Root CNAs are addressing, which can inform training, documentation, and process improvement.
- List of Sub-CNAs and New Sub-CNAs this quarter
  Rationale: Forces the periodic update of the full CNA directory.

Appendix H Disclosure and Embargo Policies

A disclosure and embargo policy should include the following information.
What process a third-party should expect when reporting a vulnerability to the CNA, including when the CNA will assign a CVE ID and when and how they will publish the CVE ID. Also, what expectations there are for the vulnerability reporter as far as their role in the disclosure process.
Communication guidelines and timelines, such as when a reporter should expect a response and what information the CNA is willing to discuss publicly. Just as important, the methods for contacting the CNA should be clearly described.
Guidelines describing what they consider to be vulnerabilities in their products. For example, they can stipulate which version of the CNT2 Counting Rule they use.
If they are involved in a Bug Bounty program, how do the rules of the Bug Bounty program affect their CVE assignment process?

Here are some examples of disclosure policies that can be used as a template for the development of a policy to be used by a CNA.

US CERT's vulnerability disclosure policy:
http://www.cert.org/vulnerability-analysis/vul-disclosure.cfm?

ENISA Good Practice Guide on Vulnerability Disclosure
https://www.enisa.europa.eu/publications/vulnerability-disclosure/at_download/fullReport

ISO/IEC 29147 Vulnerability Disclosure
https://www.iso.org/standard/45170.html

NTIA “Early Stage” Coordinated Vulnerability Disclosure Template
https://www.ntia.doc.gov/files/ntia/publications/ntia_vuln_disclosure_early_stage_template.pdf

Open Source Responsible Disclosure Framework
https://github.com/bugcrowd/disclosure-policy