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

Effective March 5, 2020

Version 3.0

Approved by CVE Board on February 1, 2020

Table of Contents

List of Figures

1 Introduction

Common Vulnerabilities and Exposures (CVE) is a list of information security vulnerabilities and exposures that provides common identifiers for publicly known cybersecurity vulnerabilities. CVE makes it possible to share data across separate vulnerability capabilities (cybersecurity tools, repositories, and services) with this common enumeration. 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 CNA Program Structure

As the CNA Program has evolved and the number of CNAs has increased, it has become necessary to categorize CNAs into the four following categories:

1. Sub-CNA
2. Root CNA
3. Program Root CNA
4. CNA of Last Resort (CNA-LR).

Sub-CNAs are the most common and basic level of CNA. Each Sub-CNA assigns CVE IDs for vulnerabilities in their own products or their domain of responsibility, hereinafter referred to as scope, and submits the vulnerability information to the CVE List when they make the vulnerability public. Sub-CNAs are administered and mentored by Root CNAs.

Root CNAs manage a group of Sub-CNAs within a given domain or community, and train and admit new Sub-CNAs, CNAs-LR, and Root CNAs within that domain or community.

At the top of the CNA hierarchy is the Program Root CNA, which is a special type of Root CNA that oversees the entire CNA Program.

Since the Sub-CNAs administered by a Root CNA may not cover all of the vulnerabilities within that Root CNA's scope, each Root CNA is responsible for assigning a CNA-LR to fill in the gaps. CNAs-LR also assign CVE IDs, but their scopes cover all vulnerabilities for their Root CNA that are not already covered by one of its Sub-CNAs.

Finally, while not a type of CNA, the Secretariat role supports many of the CNA functions (such as providing infrastructure and publishing the CVE List) and therefore is included in this document.

Each of the different types of CNAs, and the Secretariat, are roles that can be fulfilled by any organization that meets the requirements. These roles are not mutually exclusive, and a single organization can fulfill more than one of these roles at the same time. For example, the organization that operates a Root CNA may also choose to act as the CNA-LR for its scope.

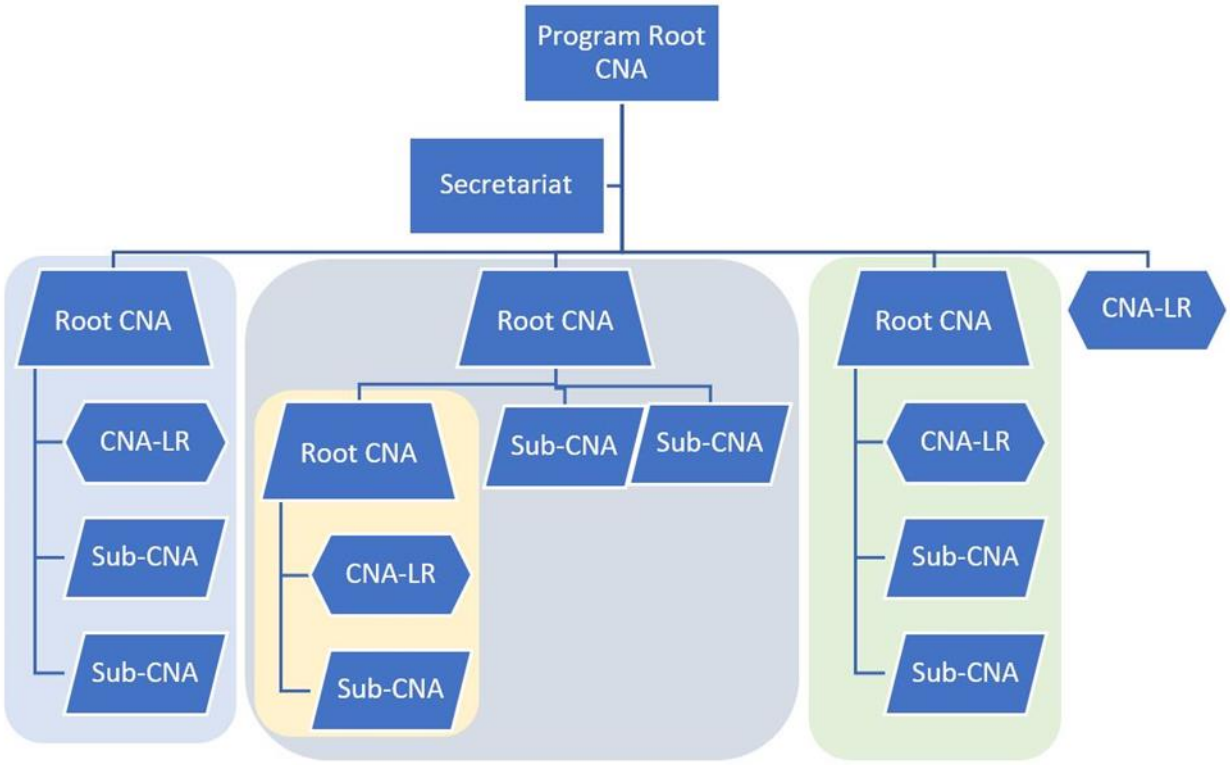

Figure 1. Federated CNA Structure

Figure 1 shows that different Root CNAs have different areas of responsibility; each colored box depicts a distinct scope. A portion of the gray box in the center of the figure is shaded yellow to indicate that part of that scope has been delegated to a particular Root CNA and its Sub-CNAs.

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 Program Root CNA. The Program Root CNA has the right to require remediation or impose sanctions on CNAs (of any type) that do not comply with the CNA Rules; however, Root CNAs are the main enforcement mechanism. Root CNAs are responsible for enforcing the rules within their area of responsibility; the Program Root CNA is the enforcement mechanism of last resort. The Root CNAs have the same level of enforcement ability as the Program Root CNA, including remediation or sanctions, within their areas of responsibility, which enables 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 CVE Entry submissions; and
- Revocation of CNA status.

The same flow, from Sub-CNAs to Root CNAs to the Program Root CNA, is followed to alert the next higher CNA when CVEs are assigned, or when reporting other programmatic data (see Figure 2, below). If the Root CNA agrees, the Sub-CNA can go directly to the Secretariat while keeping the Root CNA in the loop. The Program Root CNA provides CVE IDs to Root CNAs, and Root CNAs provide blocks of IDs to Sub-CNAs. If the Root CNA agrees, the Sub-CNAs can go to the Secretariat for blocks of IDs as well.

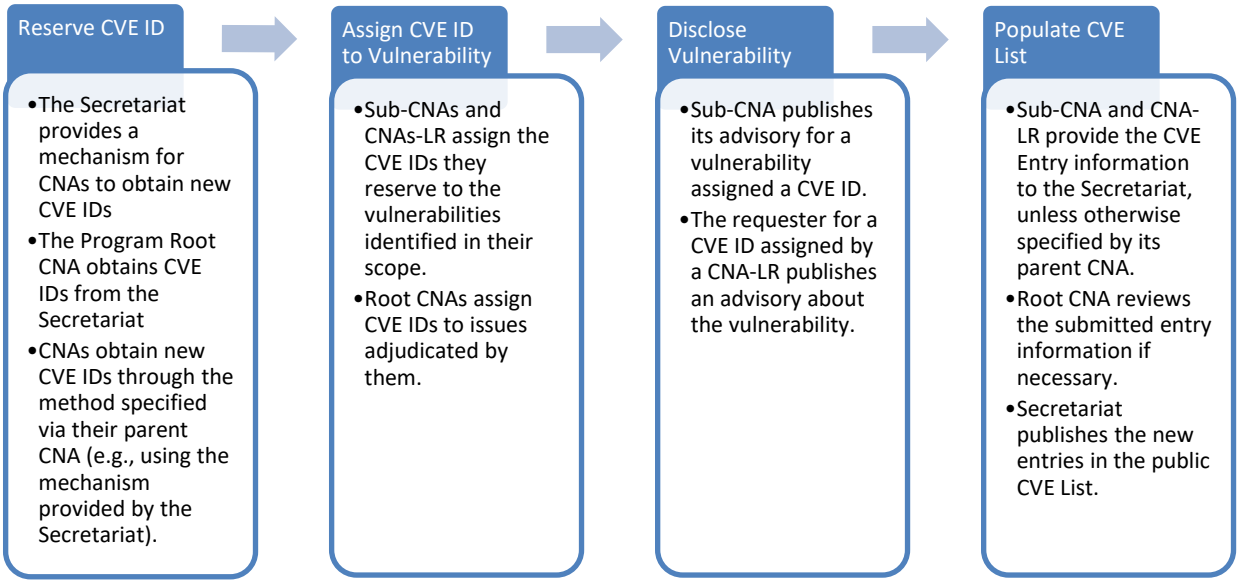

Figure 2. CNA CVE ID Lifecycle

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 domains or communities, while also maintaining consistency in the CVE assignment process and administration of the CNA Program.

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 the Secretariat and all CNAs (including Sub, Root, and CNA-LR), as well as those rules specific to Program and Root CNAs:

- Section 2: Sub-CNAs
- Section 3: Root CNAs
- Section 4: CNA of Last Resort
- Section 5: Secretariat
- Section 6: Program Root CNA
- Section 7: Assignment Rules
- Section 8: CVE Entry Requirements
- Section 9: Appeals Process
- Section 10: Defining a CNA's Scope
- Section 11: CNA Rules Update

2 Sub-CNAs

Sub-CNAs are the foundation of the CVE Program. Sub-CNAs (along with CNAs-LR) are the organizations that assign CVE IDs to vulnerabilities and generate the content for the CVE List. As such, most of the rules for Sub-CNAs revolve around CVE ID requests, assigning CVE IDs, and when and how to update the CVE List.

2.1 CVE ID Management Rules

2.1.1 MUST follow the rules for assigning CVE IDs; see Section 7 Assignment Rules.

2.1.2 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.

For example, if an operating system vendor discovers a vulnerability in a library from an upstream supplier, in addition to assigning the CVE ID to the vulnerability, the operating system vendor should attempt to contact the upstream supplier. This will help avoid duplicate CVE ID assignments and ensure that others affected by the vulnerability will be made aware of it.

2.1.3 When a vulnerability is reported to the CNA and a CVE ID is assigned to that vulnerability, the CNA MUST provide the CVE ID to the reporter.

This rule does not override any embargo rules established by the CNA.

2.1.4 Upon request by the Program Root CNA or by the CNA's Root CNA, a CNA MUST 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.

2.2 CVE Entry Management Rules

2.2.1 MUST provide CVE Entry information to the Secretariat through the process designated by its parent CNA; see Section 8 CVE Entry Requirements when a vulnerability assigned a CVE ID is made public by the CNA.

2.2.2 MUST provide CVE Entry information that meets the requirements listed in Section 8 CVE Entry Requirements.

2.2.3 SHOULD provide the CVE Entry information within 24 hours of publishing the CVE ID.

2.2.4 MUST have an established distribution point for in-scope vulnerability disclosures that is freely available to the general public without restrictions. The Sub-CNA's website MAY require registration but provide accounts for free without restriction to anyone. The Sub-CNA MAY provide additional details with restrictions as long as the information required in the reference is freely available; see Section 8 CVE Entry Requirements.

2.2.5 MUST provide information about where they will disclose the vulnerabilities to which they assign CVE IDs, such as (but not restricted to), a URL linked to the CNA's advisories. This information will be made public be the Secretariat as described in Section 8 CVE Entry Requirements.

2.3 CNA Record Management Rules

2.3.1 MUST provide Points of Contact (POCs) (e.g., email addresses, URLs, etc.) to the CNA's Root CNA and the Secretariat. This information will be made public to the Secretariat as stated in Section 5.2 Infrastructure Maintenance Rules.

2.3.2 MUST inform its Root CNA and the Secretariat when its POCs change.

2.3.3 MUST publish a disclosure (embargo) policy (see Appendix D Disclosure and Embargo Policies for additional information). This information will be made public by the Secretariat according to Section 5.2 Infrastructure Maintenance Rules.

2.3.4 MUST inform its Root CNA and the Secretariat when the CNA's scope changes.

2.3.5 MUST obtain its Root CNA's approval before changing scopes.

2.3.6 CNAs MUST provide a means (e.g., hyperlink, e-mail, web form) 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.). CNAs MUST provide the list publicly and to all levels above their own.

2.4 Administration Rules

2.4.1 MUST 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).

2.4.2 MUST operate under the CVE Terms of Use; see Appendix B Terms of Use.

2.4.3 MUST provide any documentation required to adjudicate disputes to the higher-level CNA.

3 Root CNAs

Root CNAs are the administrative arm of the CVE Program. Root CNAs can have Sub-CNAs, CNAs-LR, and other Root CNAs reporting to them. Any CNA that reports to a Root CNA is referred to herein as a child or child CNA of that Root CNA (i.e., parent CNA). Root CNAs are responsible for ensuring the other CNAs are following the CNA Rules, recruiting new CNAs, providing documentation for the CNAs, and handling issues with its child CNAs.

3.1 Child CNA Management Rules

One of the most important functions of a Root CNA is to manage the CNAs that report to it. Root CNAs are responsible for ensuring that its child CNAs understand the CNA Rules and are following them. Root CNAs ensure that the CVE Program provides its child CNAs with the necessary resources and guidance.

3.1.1 MUST provide a mechanism for child CNAs to obtain new CVE IDs.

3.1.2 MUST provide public documentation describing the specific process for submitting CVE assignments and other CVE requests.

3.1.3 MUST provide documentation on how its child CNAs should resolve issues with overlapping scopes.

3.1.4 MUST provide documentation on how to obtain new CVE IDs.

3.1.5 MUST maintain a public listing of the established assignment rules followed by the child CNAs in its domain.

3.1.6 When appropriate, a Root CNA MUST apply sanctions upon any child CNAs within its scope. The application of sanctions should occur as a last resort.

3.1.7 MAY sanction any CNA within its scope, regardless of whether the CNA reports to it. For example, if the Root CNA has a child Root CNA, then it can sanction the Sub-CNAs of the child Root CNA if necessary.

3.1.8 MUST be responsive to its child CNAs.

3.2 CNA-LR Management Rules

To ensure that there is a CNA to assign CVE IDs to all vulnerabilities within the Root CNA's scope, it must designate an organization to act as a CNA-LR. See Section 4 CNA of Last Resort (CNA-LR).

3.2.1 Root CNAs MUST designate an organization to perform the CNA-LR role for its scope.

3.2.2 A Root CNA MAY act as the CNA-LR for its scope.

3.2.3 If a Root CNA does not fulfill the CNA-LR role itself, it MUST designate some other organization for the role.

3.2.4 The Root CNA MAY designate its parent CNA's CNA-LR as its own if both its parent CNA and the CNA-LR agree.

3.3 Escalated Issues Rules

Parties who contend that a Root CNA's child CNA is not in compliance with the CNA Rules (e.g., not responding in a timely manner, refusing to assign a CVE ID to a vulnerability, not populating a CVE Entry in a timely manner, or inappropriately sanctioning a child CNA) may contact the Root CNA about the issue. The Root CNA will then judge whether the report is accurate and take any necessary actions. See Section 9 Appeals Process for a high-level description of the process.

3.3.1 MUST act as an escalation and adjudication point for issue resolution for its child CNAs.

3.3.2 MUST address CVE assignment issues from its child CNAs that require escalation.

3.3.3 MUST clearly document the dispute in the CVE Entry if the Root CNA assigns a CVE ID as the result of an escalated issue.

3.3.4 MUST provide documentation on how issues with a CNA can be escalated to the Root CNA.

3.3.5 MUST maintain a public contact method so that issues involving its child CNAs may be escalated.

3.3.6 Root CNAs MUST be responsive to escalation requests.

3.4 CNA Recruitment Rules

One of the unique responsibilities of a Root CNA is creating new CNAs within its scope. Root CNAs can create Sub-CNAs, CNA-LRs, and even other Root CNAs. When creating a new CNA, the Root CNA ensures that the candidate understands its responsibilities and can fulfill them.

3.4.1 MUST document and publish its process for creating new CNAs.

3.4.2 MUST notify the Secretariat and its parent CNA whenever child CNAs are established or removed.

3.4.3 MUST provide a public contact method to receive requests from candidate CNAs.

3.4.4 MUST be responsive to requests from candidates to become a CNA.

3.4.5 MUST ensure that the CNA candidate understands the responsibilities of its new role.

3.4.6 MUST ensure that the CNA candidate can fulfill the responsibilities of its new role.

3.4.7 MUST NOT create a new CNA with a scope that overlaps the scope of another CNA.

3.4.8 MUST NOT create a new CNA that is outside its scope.

3.4.9 SHOULD attempt to recruit new CNAs.

3.5 Administration Rules

3.5.1 MUST provide, to the Secretariat, a public list of POCs and web links for each child CNA in the Root CNA's domain.

3.5.2 MUST maintain and provide, to the Secretariat, a private list of individual POCs within each child CNA for use by CNAs only.

3.5.3 MUST notify the Secretariat and its parent CNA (if any) when sanctions are applied to a CNA.

4 CNA of Last Resort (CNA-LR)

Each Root CNA must designate a CNA-LR, which is similar to a Sub-CNA, but its scope is much broader (i.e., the Root CNA's scope minus the vulnerabilities covered by the Root's other child CNAs). The major differences between CNA-LRs and Sub-CNAs are their reporting requirements and the lack of control over their scopes, and a heightened need to collaborate with the other CNAs in their parents' scopes.

4.1 CVE ID Management Rules

4.1.1 MUST apply the assignment rules defined in Section 7 Assignment Rules.

4.1.2 The scope for a CNA-LRs is vulnerabilities within its parent Root CNA's scope that are not covered by another CNA's scope.

4.1.3 MUST be responsive to those requesting CVE ID assignments.

4.1.4 When a vulnerability is reported to the CNA and a CVE ID is assigned to that vulnerability, MUST provide the CVE ID to the reporter.

4.1.5 Upon request by the Program Root CNA or by the CNA's parent CNA, MUST 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.)

4.2 CVE Entry Management Rules

4.2.1 When submitting the entry information for a CVE ID, MUST follow the requirements in Section 8. CVE Entry Requirements.

4.2.2 MUST follow the process defined by its Root CNA to submit the entry information for the CVE IDs it assigns.

4.3 CNA Record Management Rules

4.3.1 MUST provide POCs (e.g., email addresses, URLs, etc.) to all levels above their own (including the Secretariat). This information will be made public be the Secretariat according to Section 5.2 Infrastructure Maintenance Rules.

4.3.2 MUST provide a means (e.g., hyperlink, e-mail) for the public to contact them regarding vulnerabilities. This information will be made public be the Secretariat according to Section 5.2 Infrastructure Maintenance Rules.

4.4 Administration Rules

4.4.1 MUST operate under the CVE Terms of Use; see Appendix B Terms of Use.

4.4.2 MUST provide any documentation required to adjudicate disputes to the higher-level CNA.

4.4.3 MUST 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 Secretariat

The Secretariat role is not a type of CNA, but supports many of the necessary functions for the CNAs to fulfill their responsibilities. The Secretariat maintains the CVE List, maintains the infrastructure used by the CVE Program (e.g., the website and mailing lists), and organizes the CVE Board and CVE Working Group meetings.

5.1 CVE List Maintenance Rules

The CVE List provides the official mapping of CVE IDs to vulnerabilities. The Secretariat is responsible for collecting this information from the CNAs and publishing it for use by the CVE community.

5.1.1 MUST maintain the CVE List and provide that information to the public.

5.1.2 MUST provide the CVE List to the public free of charge.

5.1.3 MUST maintain the Terms of Use under which the CVE List is published.

5.1.4 MUST 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 Secretariat 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 if necessary.)

5.1.5 MUST 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 Program Root CNA and can be updated based on the needs of the CVE community.

5.1.6 MUST maintain the CVE ID syntax (e.g., CVE dash year dash arbitrary number of digits).

5.2 Infrastructure Maintenance Rules

The Secretariat maintains the infrastructure to ensure the smooth assignment of CVE IDs and publication of CVE Entry information.

5.2.1 MUST provide a mechanism for CNAs to obtain CVE IDs for assignment.

5.2.2 MUST document and publish how to obtain CVE IDs.

5.2.3 MUST provide any necessary permissions to the infrastructure.

5.2.4 MUST provide a listing of all Root CNAs and Sub-CNAs, including public points of contact and web links. The Secretariat obtains this information from Root CNAs.

5.2.5 MUST maintain a private list of individual POCs for each Root and Sub-CNA for use by CNAs only.

5.2.6 MUST provide coordination of communication channels between Root CNAs.

5.2.7 MUST maintain a public listing of the established counting rules for the CVE Program, see Section 7 Assignment Rules.

5.2.8 MUST provide a mechanism through which CNAs can provide CVE Entry information to the CVE List.

5.2.9 MUST serve as a member, and the Board Moderator, of the Board.

5.3 Administration Rules

5.3.1 MUST accept metrics reports from Root CNAs quarterly, within one month of the calendar quarter.

5.3.2 MUST schedule and provide infrastructure for Board and CVE Working Group meetings.

5.3.3 MUST respond to inquiries by Root CNAs and Sub-CNAs in a timely manner; establish metrics for such responsiveness.

6 Program Root CNA

The Program Root CNA is at the top of the CNA hierarchy. It acts as the final arbiter for all disputes between CNAs and content-related decisions, develops the CNA Rules with approval from the CVE Board, recruits and onboards new CNAs, and ensures that all other CNAs are following CNA Rules.

6.1 Program Root CNA Rules

6.1.1 MUST follow the rules of a Root CNA as documented in Section 3 Root CNAs.

6.1.2 When necessary, the Program Root CNA SHOULD revise the CNA Rules.

6.1.3 MUST get the approval of the Board before implementing any revisions to the CNA Rules.

6.1.4 MUST inform the Secretariat when the CNA Rules are revised.

6.1.5 MUST decide on any issue escalated to it. There is no Root CNA above it so there is no one to escalate the issue to past the Program Root CNA.

6.1.6 MUST serve as a member of the Board.

7 Assignment Rules

7.1 What is a Vulnerability

The CVE Program does not adhere to a strict definition of a vulnerability. For the most part, CNAs are left to their own discretion to determine whether something is a vulnerability. Root CNAs may provide additional guidance to their child CNAs. This allows the program to adapt to definitions used in different industries, legal regimes, and cultures.

7.1.1 If a product owner considers an issue to be a vulnerability in its product, then the issue MUST be considered a vulnerability, regardless of whether other parties (e.g., other vendors whose products share the affected code) agree.

7.1.2 If the CNA determines that an issue violates the security policy of a product, then the issue SHOULD be considered a vulnerability.

7.1.3 If a CNA receives a report about a new vulnerability that has a negative impact, then the reported vulnerability MAY be considered a vulnerability.

7.2 How many Vulnerabilities

The CVE Program expects separate CVE IDs to be assigned to independently fixable vulnerabilities. If one vulnerability can be fixed without fixing the other, then the vulnerabilities should receive separate CVE IDs. The exception is when the vulnerabilities are independently fixable because they are in different products, but those products are affected because they share the same code, or the products are affected because they use the functionality of another product. "Product" in this case being a broad term that includes standards, application programming interfaces (APIs), and protocols.

7.2.1 CNAs MUST NOT assign the same CVE ID to more than one independently fixable vulnerability.

7.2.2 CNAs MUST NOT assign a CVE ID to a vulnerability that is dependent on another vulnerability. The dependent vulnerability should share the same CVE ID as the vulnerability it is dependent on. For example, if a buffer overflow occurs only when an integer overflow occurs, then the buffer overflow should share the same CVE ID as the integer overflow.

7.2.3 If a CNA is uncertain whether two issues are independently fixable, then the CNA SHOULD assign a single CVE ID.

7.2.4 If multiple products are affected by the same independently fixable vulnerability, then the CNA:

a. MUST NOT assign more than one CVE ID if the products are affected, because they share the vulnerable code. The assigned CVE ID will be shared by the affected products.

b. MUST assign different CVE IDs if the products do not share affected code.

c. SHOULD assign different CVE IDs if the CNA is uncertain whether the products share code.

7.2.5 If a product is affected by a vulnerability because it uses the functionality or specification of another product, then a CNA:

a. MUST assign a CVE ID to each known vulnerable implementation if there is a secure method of using the functionality or specification.

b. MUST assign a single CVE ID if there is no option to use the functionality or specification in a secure manner.

c. SHOULD assign different CVE IDs to each known vulnerable codebase if the CNA is uncertain whether there is a secure option.

7.3 CNA Scope

CNAs are restricted to assigning CVE IDs to vulnerabilities within their scope. Sometimes, a CNA's scope overlaps with another CNA's. Examples include researcher CNAs who discover

the same vulnerability at the same time and maintainer CNAs whose products are dependent on the same vulnerable library. In cases like these, CNAs are expected to negotiate between themselves to determine who should assign the CVE ID. If an agreement cannot be achieved, then the issue should be escalated to the appropriate Root CNA.

7.3.1 CNAs MUST NOT assign CVE IDs to vulnerabilities outside of their scope.

7.3.2 CNAs SHOULD assign CVE IDs within their scope.

7.3.3 If a vulnerability falls within the scope of multiple CNAs, the conflict MUST be resolved using the process defined by their Root CNA(s).

7.4 Requirements for Assigning a CVE ID

Not all vulnerabilities receive a CVE ID. CVE IDs are meant to help two or more parties communicate about a vulnerability. If assigning a CVE ID would not achieve this goal, then a CVE ID should not be assigned. The following rules are meant to ensure the goal is achieved.

7.4.1 CNAs MUST intend to make the vulnerabilities for which they assign CVE IDs public if they are not already.

7.4.2 CNAs MUST NOT assign CVE IDs to vulnerabilities they do not intend to make public.

7.4.3 CNAs MUST NOT assign a new CVE ID to a vulnerability that has already been assigned a CVE ID.

7.4.4 CNAs MAY assign a CVE ID to a vulnerability if:

a. The product or service is owned by the CNA,

b. The product or service is not customer controlled, and

c. The vulnerability requires customer or peer action to resolve.

7.4.5 CNAs MUST NOT assign a CVE ID to a vulnerability if the affected product(s) or service(s):

a. Are not owned by the CNA, and

b. Are not customer controlled.

7.4.6 CNAs MAY assign a CVE ID to a vulnerability if the affected product(s) or service(s):

a. Are not owned by the CNA and

b. Are customer controlled.

7.4.7 CNAs SHOULD NOT assign CVE IDs to vulnerabilities in products that are not publicly available or licensable.

7.4.8 CNAs SHOULD NOT consider factors other than those listed in these rules (e.g., whether the vulnerability affects end-of-life products) when deciding whether to assign a CVE ID. If they do, and the issue gets escalated to the CNA's parent CNA, then the parent CNA SHOULD assign a CVE ID.

8 CVE Entry Requirements

The following sections include what information a CVE Entry must contain when it is submitted by a CNA.

8.1 CVE Entry Information Requirements

This section defines the information required to be in a CVE Entry. The CVE Entry should contain enough information for users to understand to which vulnerability the CVE ID was assigned. The minimum required information is defined in this section. However, additional information can be provided and is encouraged.

8.1.1 MUST contain the name of an affected Product.

8.1.2 MUST contain the affected or fixed version(s).

8.1.3 MUST contain the CVE ID for the entry.

8.1.4 MUST contain at least one of the following:

a. Vulnerability Type
b. Root Cause
c. Impact

8.1.5 MUST contain at least one public reference. See Section 8.3 Reference Requirements for complete requirements for the public reference.

8.1.6 MUST contain a prose description. See Section 8.2 Prose Description Requirements for complete requirements for the prose description.

8.1.7 SHOULD indicate whether the CVE ID was assigned to a vulnerability affecting only products that are no longer supported.

8.2 Prose Description Requirements

This section defines the requirements for the prose description requirement defined in Section 8.1.6. While some of this information is included in CVE JSON fields, it is duplicated in the Description.

8.2.1 MUST provide enough information for a reader to have a reasonable understanding of what products are affected. If the affected products are not explicitly listed in the description, then the CNA MUST provide a reference that points to the known affected products.

8.2.2 SHOULD include the affected or fixed version(s).

8.2.3 MUST include one of the following:

a. Vulnerability Type
b. Root Cause
c. Impact

8.2.4 MUST NOT credit people by name in the description. CNAs are free to credit or acknowledge discoverers within their own advisories.

8.2.5 MUST include an English description. MAY include descriptions in other languages, but at least one of them must be English.

8.2.6 MAY contain information not listed here.

8.2.7 MUST NOT contain information that is not germane to describing the vulnerability (e.g., foul language or advertising).

8.3 Reference Requirements

This section defines the requirements for the public references.

8.3.1 MAY require registration or login, but MUST NOT have any other restrictions (e.g., be a paying customer).

8.3.2 SHOULD contain information about the vulnerability.

8.3.3 SHOULD use either the http, ftp, https, or ftps protocol.

8.3.4 MUST be accessible from the Internet.

8.3.5 The Terms of Use of the website MUST allow the CVE List to link to the URL.

8.3.6 MUST contain the minimum required information for a CVE Entry (See Section 8.1 CVE Entry Requirements). This information may be spread across multiple references.

8.4 Formatting

8.4.1 CNAs MUST submit CVE Entries in the format(s) approved by their Root CNA.

8.4.2 CNAs are always permitted to submit CVE Entries in the CVE minimum JSON specification.

9 Appeals Process

Each Root CNA can specify its own process for handling issues escalated to it. However, at a high level, the following process SHOULD be followed:

1. The party seeking to appeal a decision made by a Root CNA, or resolve a disagreement between Root CNAs, contacts the Program Root CNA at https://cveform.mitre.org/, and follows the steps below:
    a. Enter your email address.
    b. Select "Other" in the **Select a request type** field.
    c. Select "Issue" in the **Type of comment** field.
    d. Enter "Arbitration Request", in the **Please provide your question, issue, comment, etc.** field.
2. The Program Root CNA sets expectations for when a timely resolution may be available. Appeals of time-sensitive issues are prioritized, as determined by the Program Root CNA.
3. The Program Root 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 Program Root CNA may also engage the Board for their consideration of the issue.
4. The Program Root CNA communicates its decision to all relevant parties once the disagreement or appeal has been fully considered. This result is final.

10 Defining a CNA's Scope

Every CNA must provide a scope definition to indicate to what exactly they will assign CVE IDs. This scope should be published by the Secretariat, and be made available on the CNA's website (e.g., on their disclosure/security policy page). The scope should be a blanket statement (e.g., "All of Organization X's components"), a list of specific components covered, a list of specific components not covered, or a mix of covered and not covered. This scope statement should also specify coverage for components that are not part of the CNA's core business or purpose (for example, free tools).

A CNA should list in their scope whether they want to handle CVE assignments for end-of-life components according to the end-of-life rules or whether other CNAs should assume that responsibility if they have a need to reference such vulnerabilities. If a CNA specifies that it will not assign for end-of-life components, other CNAs may assign for those components.

CNA scope definitions for researchers and third-party coordinators may say “we will issue CVE IDs for components or projects that we are researching or coordinating unless they are otherwise covered by another CNA.”

For all CNAs, the published scope must be updated whenever a CNA’s scope changes. Scopes may change due to the introduction of new components (products, projects, etc.), mergers, sales, or acquisitions at an organizational level, or a change in process.

11 CNA Rules Updates

11.1 Rules for Updating the CNA Rules

11.1.1 The Program Root CNA maintains the CNA Rules.

11.1.2 Any suggested change should be proposed to Program Root CNA.

11.1.3 Any changes to the CNA Rules must be approved by the CVE Board.

11.1.4 A Root CNA may make additions to the CNA Rules. These additions would only apply to its child CNAs. For example, if there were a Root CNA for Spanish companies, that Root CNA may require its child CNAs submit entries in both Spanish and English.

11.1.5 A Root CNA MUST NOT make an addition to the CNA Rules that would contradict this document. For example, if there were a Root CNA for Spanish companies, that Root CNA may not require its child CNAs submit entries in only Spanish, as this would violate Section 8.2 Prose Description Requirements.

11.1.6 A CNA can apply to the CVE Board for an exception to a rule.

11.1.7 The CVE Board may approve exceptions to the CNA Rules.

11.1.8 Root CNAs cannot approve exceptions to the CNA Rules.

11.1.9 The Secretariat MUST document any exceptions approved by the CVE Board.

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.

A **child CNA** is any CNA within the hierarchy of a Root CNA. Root CNAs, Sub-CNAs, and CNA-LRs can be considered child CNAs.

The **CNA hierarchy** describes the relationship between Root CNAs and the other CNAs that report to it. If a Root CNA creates a new CNA, then the new CNA is within its hierarchy. CNA hierarchies are transitive. If Root CNA B reports to Root CNA A, then Root CNA B is in Root CNA A's hierarchy. If a CNA reports to Root CNA B, then that CNA is also in Root CNA A's hierarchy.

A **CNA of Last Resort (CNA-LR)** is an organization that is authorized to assign CVE IDs to vulnerabilities within a Root CNA's scope so long as there is not a Sub-CNA in the Root CNA's hierarchy responsible for the vulnerability or a CNA-LR lower in the Root CNA's hierarchy responsible for the vulnerability.

A **CVE Entry** is the descriptive data about the vulnerability included in the CVE List. The data includes, the CVE ID, product and version information, a prose description, and references. The full requirements for a CVE Entry can be found in Section 8.1. CVE Entry Information Requirements.

An **end-of-life** product is any product that is no longer supported by the owner or maintainer of the product.

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).

A **parent CNA** is the Root CNA to which that CNA reports. All Root CNAs also have a parent CNA, except for the Program Root CNA, which is at the top of the CNA hierarchy.

The **Program Root CNA** operates the CNA 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 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 a CNA regarding the issue).

A **reference** is the information about external sources where information about the vulnerability can be found. Requirements for a reference can be in Section 8.3 Reference Requirements.

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. Root CNAs may also manage other Root CNAs and CNAs-LR.

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

The **Secretariat** is the organization that performs the general secretary and administrative duties for the CVE Program, including maintaining the infrastructure, scheduling meetings, and publishing the CVE List.

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.

A **vulnerability** in the context of the CVE Program is defined by Section 7. Assignment Rules. 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)." NOTE: CNAs are not restricted to using this definition of a vulnerability.

A.1 CVE States

A.1.1 CVE ID States

- **Allocated** – When CVE IDs are first given to a CNA for later assignment to a vulnerability, they are in the allocated stated.
- **Assigned** – If a CVE ID has been associated with a vulnerability by a CNA, then the CVE ID is in the assigned state.
- **Public** – If the CVE ID is being used publicly to discuss a vulnerability, then it is in the public state. See Section 8.3 Reference Requirements for the requirements for the CVE Program to consider a CVE ID public.
- **Rejected** – If the CVE ID should no longer be used, then it is in the rejected state.

A.1.2 CVE Entry States

- **Reserved** – When a CVE ID has been allocated to a CNA, it is immediately added to the CVE List in the reserved state. The reserved entry is a placeholder until the information about the vulnerability is made public.
- **Populated** – When the information for the vulnerability is filled in the CVE List, the CVE Entry is considered populated. Populated can also be used as a verb to describe the process of filling the vulnerability details into the CVE Entry.
- **Rejected** – When the CVE ID and associated CVE Entry should no longer be used, the CVE Entry is in the rejected state. Rejected CVE Entries remain in the CVE List so that users can know when it is invalid.

Reserved but Public (RBP) – A term used to describe when the CVE ID is in the public state but the associated CVE Entry is in the reserved state. This happens when a CNA has published its advisory for the vulnerability but has not populated the CVE Entry.

Appendix B 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 C Process to Correct Assignment 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 Program Root 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 CVE Entries.

C.1 Dispute: CNA Rules Violations

In the event that a CVE Entry is published in violation of other CNA Rules, as reported to that CNA's Root CNA or the Program Root CNA, the offending CNA's Root CNA shall perform one of the following actions:

1. In the case the CVE Entry was published by an out-of-scope CNA, the offending CNA's Root CNA shall invite the in-scope CNA to review and edit the CVE Entry as appropriate. If at all possible, the originally published CVE ID should be preserved. If this results in a duplicate, split, or merged CVE Entry, the rules on handling those conditions listed below shall apply.
2. In the case where the CVE Entry was published with errors due to CNA rules violations (other than being out-of-scope), the offending CNA shall update the CVE Entry

accordingly to resolve those errors. If the issuing CNA refuses to do so, the offending CNA's Root CNA shall review and edit the CVE Entry as needed.

3. In the case where no CVE Entry was published because the in-scope CNA refuses to do so, the offending CNA's Root CNA shall be contacted to publish a CVE Entry. In many cases, this will lead to a disputed CVE Entry, which shall be handled according to the rules regarding disputing a CVE Entry, as noted below.

In all cases, repeated rules violations by CNAs can lead to the revocation of CNA status by the offending CNA's Root CNA.

C.2 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 the reason for the rejection is provided.

C.3 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.

C.4 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.

C.5 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. The CVE Program deals with these differing opinions as follows:

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 Entry 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 Section 7.1), or does not meet one of the other requirements for assignment in Section 7.4. 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 Section 7.4 are not things that can be disputed, per se. They either are true, or are not true.

Appendix D Disclosure and Embargo Policies

The CVE Program is not designed to provide or support the non-public aspects of coordinated vulnerability disclosure, and does not require specific disclosure or embargo practices. CNAs are required to provide a disclosure policy to the Secretariat (see Section 2.3.3). 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.
- If they are involved in a Bug Bounty program, how the rules of the Bug Bounty program affect their CVE assignment process.

Below 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:

- DHS CISA Vulnerability Disclosure Policy
- CERT Coordination Center Vulnerability Disclosure Policy
- ENISA Good Practice Guide on Vulnerability Disclosure
- ISO/IEC 29147 Vulnerability Disclosure
- NTIA “Early Stage” Coordinated Vulnerability Disclosure Template
- Open Source Responsible Disclosure Framework
- Bug Bounty - HackerOne Disclosure Policy
- Researcher - Rapid7 Disclosure Policy

List of Acronyms

Acronym	Definition
API	Application Programming Interface
CNA	CVE Numbering Authority
CNA-LR	CNA of Last Resort
CVE	Common Vulnerabilities and Exposures
ID	Identifier
POC	Point of Contact