The Software Alliance

BSA

GUIDING PRINCIPLES FOR COORDINATED VULNERABILITY DISCLOSURE

Vulnerabilities are an unfortunate and inevitable byproduct of today's highly complex and increasingly interconnected software systems. As state-of-the art software and hardware engineering continues to evolve, code that is considered secure by today's standards can become susceptible to newly identified vulnerabilities in the future. Although pre-release coding practices are the core foundation on which secure software is built, experts agree that long-term security requires a lifecycle risk management process for addressing vulnerabilities that are identified post-release.

Because software vulnerabilities are often identified by external stakeholders, such as independent security researchers, it is critical for vendors to maintain procedures for processing such third-party reports. Fortunately, the information security community has developed a set of protocols known as "coordinated vulnerability disclosure" (CVD) to help vendors work with third-party stakeholders to mitigate potential risks to the public.

The guiding principle of CVD is that the public is best served when vulnerabilities are reported directly to vendors that can fix them and when public disclosures are delayed until the vendor has had an opportunity to develop, test, and deploy a patch to mitigate the underlying vulnerability. To operationalize this underlying principle, software vendors maintain CVD programs to respond to third-party vulnerability reports in a manner that minimizes the risk of malicious actors leveraging unpatched vulnerabilities to hack into systems.

92% of security researchers polled by an National Telecommunications and Information Administration survey report that they utilize CVD processes.

The goal of a CVD program is ultimately to protect users by:

(1) Ensuring that reported vulnerabilities are addressed.

(2) Minimizing the risk from vulnerabilities.

(3) Providing users with sufficient information to evaluate risks from vulnerabilities to their systems.

(4) Setting expectations to promote positive communication and coordination with the entity that reported the vulnerability.[1]

Importantly, CVD protocols recognize that there is no single, prescriptive approach for meeting these goals in all circumstances.

Instead, CVD is better thought of as a set of principles, policies, and procedures that can be tailored to accommodate an organization's structure and technical capacity, and adapted to address the unique considerations that may be implicated by individual vulnerabilities. Nonetheless, there are several important elements that are common to virtually all successful CVD programs.

[1] ISO/IEC 29147:2014, Information Technology-Security Techniques-Vulnerability Disclosure.

Common CVD Terminology

Vulnerability: Weakness of software, hardware, or online service that can be exploited to compromise the integrity, availability, or confidentiality of systems and/or data.

Exploit: A software program or sample code that, when executed against a vulnerable system, uses a security vulnerability to cause unintended and/or unanticipated behavior.

Finder: An individual, organization, or government that identifies a potential vulnerability.

Owner: The vendor, individual, or organization that created or maintains the product that is vulnerable to an identified vulnerability.

Coordinator: An organization, such as a national Computer Emergency Response Team (CERT) or a recognized bug bounty provider, that cooperatively works as an intermediary with finders and owners to privately disclose newly discovered vulnerabilities directly to the vendor of the affected product or service.

Bug Bounty: A formal program that provides incentives for finders to identify and report vulnerabilities in a vendor's software, hardware, and or/or online services.

Elements of a Successful Coordinated Vulnerability Disclosure Program

1. **Intake Mechanism:** CVD programs typically accept vulnerability reports through a dedicated channel (such as a portal or email address) and/or through existing customer service channels. Some CVD programs rely on a mix of these intake mechanisms. Regardless of the chosen intake mechanism, the mechanism should be clear, publicly known, regularly monitored, and adequately protected (e.g., allowing the receipt of encrypted messages) to ensure submissions are received and secure. The intake mechanism should include a means for confirming receipt of a vulnerability and establishing a channel of communication with the entity that reported it.

2. **Defining the Program Terms:** In defining program terms, the goal should be to establish clear rules of the road about the types of research that are permitted under the program, how vulnerabilities should be submitted, and how they will be processed. In exchange for a commitment to abide by the CVD program terms, some companies agree to forego potential legal claims for conduct that violates a legal right of the company but does not harm end-users.

   Program terms need not be highly complex, but they should provide enough information so that there is a mutual understanding between the vendor and security researcher about their respective roles and responsibilities. While program terms will vary, common items include:

   a. **Scope and Exclusions:** It is important to specify the products and properties that are in scope for the program (e.g., a specified domain, certain types of enterprise products, a category of physical products) and the types of vulnerabilities that can be reported through the program. Conversely, program terms may also exclude specific attack techniques that could create operational risks (e.g., phishing attacks, denial of service attacks).

   b. **Prohibitions and Limitations:** Program terms generally include prohibitions on illegal or harmful conduct that could create privacy risks or harm other end-users. To avoid uncertainty, such policies ideally leverage existing legal definitions as opposed to novel standards.

   c. **Engagement Expectations:** Program terms should identify the company's expectations, if any, such as that the researcher will meet applicable requirements in terms of submission quality and/or will not include personal information. Conversely, CVD program terms should also identify what the researcher can expect from the company in terms of response times, follow-up, and credit.

d. **Rules Regarding Disclosure:** To prevent harms to end-users that could arise if malicious actors are made aware of the existence of an unpatched vulnerability, program terms typically require researchers to withhold publication of vulnerability information until a remediation has been developed, tested, and released.

3 **Validation, Prioritization, and Remediation Processes:** Because the purpose of CVD is to protect end-users, encourage effective and substantive disclosures, and improve system security, any successful program will include robust processes for validating, prioritizing, and remediating reported vulnerabilities. Ideally, these processes should be integrated into the company's larger vulnerability management strategy so that all known vulnerabilities are addressed in a coherent and efficient manner. These processes should draw upon existing standards for vulnerability disclosure (ISO/IEC 29147) and management (ISO/IEC 30111). It is also important for vendors to maintain a communication channel to acknowledge receipt of vulnerability reports and share information about the status of the intake, validation, and remediation processes. Such a communication channel also enables the vendor to request additional information from the reporting party and ensure that mutual expectations are clear.

4 **Multi-Party Disclosure Processes:** Successful CVD programs also include a process to identify instances when disclosure of vulnerability information to outside stakeholders might be necessary or otherwise warranted. For instance, CVD reports that implicate vulnerabilities in third-party software or hardware will often need to be shared with the relevant third-party vendor that can ensure that the vulnerability is mitigated and/or patched. In some instances, a vendor may determine that mitigations will require cooperation from a range of outside stakeholder (e.g., hardware manufacturers, operating system vendors, core infrastructure providers), and will want to establish procedures for sharing necessary information in a confidential manner with the circle of companies that will need to develop mitigations. The inherent complexity of multi-party disclosure scenarios has given rise to its own set of norms regarding appropriate coordination efforts.[2]

5 **Resourcing and Governance:** Allocating the necessary resources to validate, triage, and mitigate vulnerabilities that are reported through a CVD program is critical. It is likewise important to develop governance structures to assign personnel with clear lines of responsibility, establish a risk-based mechanism for prioritizing the remediation of vulnerabilities, and provide clear guidance about how and when to disclose vulnerability information to external stakeholders. Successful CVD programs are typically integrated into the company's larger vulnerability management program. Companies that build successful programs make sure that there is coordination around the various different intake channels for vulnerabilities (e.g., CVD, penetration testing, reports from suppliers, incidents). For example, companies may use a unified tracking system for managing vulnerabilities discovered across contexts, make decisions based on information from each channel (e.g., identify targets for penetration testing based on reported vulnerabilities), and/or allocate resources in a way that allows effective support of these different sources of vulnerabilities.

6 **Iterative Learning:** Successful CVD programs should capture lessons learned from vulnerability reports to enable improvement of an organization's secure development practices. Ultimately, the goal of a CVD program isn't merely to patch individual vulnerabilities. Rather, CVD programs should be considered a mechanism for better protecting end-users, improving an organization's overall security posture by addressing vulnerabilities in a prioritized way commensurate with risk, and identifying trends that signal breakdowns in secure development practices.

[2] See, *Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure*, FIRST, available at https://first.org/global/sigs/vulnerability-coordination/multiparty/guidelines-v1.0.

The BSA Framework for Secure Software provides a comprehensive, outcome-based benchmark for software security. The Framework's "Secure Lifecycle" function identifies CVD as a core best practice for software developers, and provides guidance for implementing an effective CVD program.

For more information visit **www.bsa.org/softwaresecurityframework.**

Category	Subcategory	Diagnostic Statement	Relevant Standards and Informative Resources
SECURE LIFECYCLE			
Vulnerability Management (VM)	**VM.3.** The vendor maintains a coordinated vulnerability disclosure program.	**VM.3-1.** The vendor establishes a clearly defined and easily accessible intake mechanism to accept vulnerability information (email, portal, etc.).	ISO 29147; SAFECode "Fundamental Practices"; SAMM; ENISA Good Practice Guide on Vulnerability Disclosure; IoT Security Foundation Vulnerability Disclosure Best Practice Guidelines
		VM.3-2. A vendor's intake mechanism provides for secure and confidential communication of sensitive vulnerability information.	ISO 29147; SAFECode "Fundamental Practices"; IoT Security Foundation Vulnerability Disclosure Best Practice Guidelines
		VM.3-3. The vendor publishes, in simple and clear language, its policies for interacting with vulnerability reporters, addressing, at minimum: (1) how the vendor would like to be contacted, (2) options for secure communication, (3) expectations for communication from the vendor regarding the status of a reported vulnerability, (4) desired information regarding a potential vulnerability, (5) issues that are out of scope of the vulnerability disclosure program, (6) how submitted vulnerability reports are tracked, and (7) expectations for whether and how a reporter will be credited.	ISO 29147; ENISA Good Practice Guide on Vulnerability Disclosure; IoT Security Foundation Vulnerability Disclosure Best Practice Guidelines
		VM.3-4. The vendor maintains a system to record and track all reports of potential vulnerabilities.	ISO 29147
		VM.3-5. The vendor notifies vulnerability reporters of when reported vulnerabilities are remediated or mitigated.	ISO 29147

Outside Resources

CERT Guide to Coordinated Vulnerability Disclosure

ISO/IEC 29147 — Vulnerability Disclosure

ISO/IEC 30111 — Vulnerability Handling Processes

FIRST Best Practices for Multi-Party Disclosure Scenarios

DOJ Framework for Vulnerability Disclosure Programs

BSA Framework for Secure Software