Your TL;DR Summary of The CERT Guide to Coordinated Vulnerability Disclosure

Published on hackerone.com Oct. 26th, 2017

hackerone

CERT

Software Engineering Institute
Carnegie Mellon

The CERT Coordination Center at Carnegie Mellon University's Software Engineering Institute (SEI) recently released The CERT Guide to Coordinated Vulnerability Disclosure.

It is an amazingly detailed, clever, and complete guide to explaining the need for coordinated vulnerability disclosure (CVD), who should be involved, and how to react when the process hits bumps along the way.

But it's also 121 pages.

With your precious time in mind, we put together a summary of the CERT Guide. NOTE: Everything below is a direct cut-and-paste from their document unless contained within [square brackets]. Certain bolded text emphasis are added by us. We've also added a "further reading" section at the end, so be sure and check that out.

Let's go!

Background

CVD can be thought of as an iterative process that begins with someone finding a vulnerability, then repeatedly asking "what should I do with this information?" and "who else should I tell?" until the answers are "nothing," and "no one."

It's written for vulnerability analysts, security researchers, developers, and deployers; it's for both technical staff and their management alike.

1. Introduction

Vulnerability disclosures fall between two extremes:

1. Disclose everything you know about a vulnerability to everyone as soon as you know it.
2. Never disclose anything you know about a vulnerability to anyone.

Prior research into vulnerability disclosure practices has shown that neither approach is socially optimal.

The CERT/CC believes the Coordinated Vulnerability Disclosure (CVD) process provides a reasonable balance of these competing interests. The public and especially users of vulnerable products deserve to be informed about issues with those products and how the vendor handles those issues.

At the same time, disclosing such information without review and mitigation only opens the public up to exploitation. **The ideal scenario occurs when everyone coordinates and cooperates to protect the public.**

This coordination may also be turned into a public relations win for the vendor by quickly addressing the issue, thereby avoiding bad press for being unprepared.

Governments and international organizations also recognize the need for coordinated vulnerability disclosure practices.

2. Principles of Coordinated Vulnerability Disclosure

These principles include the following:

- **Reduce Harm** - (Balance) the ability for system defenders to take action while avoiding an increase in attacker advantage.
- **Presume Benevolence** - Assume that any individual who has taken the time and effort to reach out to a vendor or a coordinator to report an issue is likely benevolent and sincerely wishes to reduce the risk posed by the vulnerability.
- **Avoid Surprise** - Clearly communicating expectations across all parties involved in a CVD process.
- **Incentivize Desired Behavior** - Incentives can take many forms... recognition, gifts, money, employment.
- **Ethical Considerations** - The Usenix' System Administrators' Code of Ethics includes an ethical responsibility "to make decisions consistent with the safety, privacy, and well-being of my community and the public, and to disclose promptly factors that might pose unexamined risks or dangers."
- **Process Improvement** - Capture ideas that worked well and note failures. A successful CVD program feeds vulnerability information back into the vendor's Software Development Lifecycle, (and) helps encourage the search for and reporting of vulnerabilities while minimizing harm to users.
- **CVD as a Wicked Problem** - The goal of a solution is not to find an ultimate truth about the world, rather it is to improve conditions for those who inhabit it.

3. Roles in CVD

Certain roles are critical to the Coordinated Vulnerability Disclosure process, as described below:

- **Finder (Discoverer)** – the individual or organization that identifies the vulnerability
- **Reporter** – the individual or organization that notifies the vendor of the vulnerability
- **Vendor** – the individual or organization that created or maintains the product that is vulnerable
- **Deployer** – the individual or organization that must deploy a patch or take other remediation action
- **Coordinator** – an individual or organization that facilitates the coordinated response process

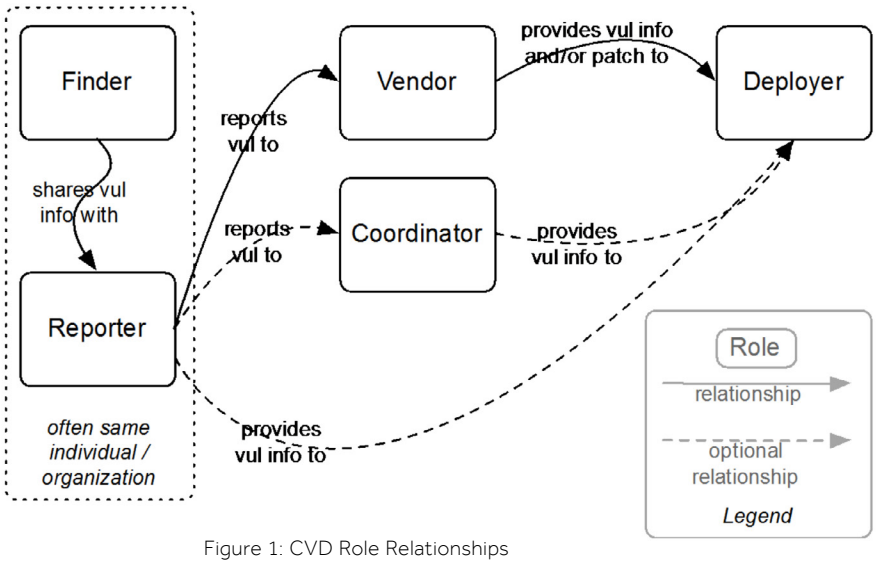

Figure 1: CVD Role Relationships

[Referring to Vendor]
The vendor is the party responsible for updating the product containing the vulnerability. Most often a vendor is a company or other organization, but an individual can also be a vendor. Many open source libraries are maintained by a single person or a small independent team; we still refer to these individuals and groups as vendors.

Having a mechanism to receive and track the disposition of vulnerability reports is an important first step in establishing a vendor's vulnerability response capability.

[Referring to Coordinator]
Complicated or complex CVD cases can often benefit from the help of a coordinator. A coordinator acts as a relay or information broker between other stakeholders.

In recent years, a new class of coordinator has emerged in the form of commercial bug bounty program providers. Many individual vendors have established programs to compensate security researchers for their efforts in discovering vulnerabilities in the vendor's products.

Creation of a bug bounty program has been noted as an indicator of maturity in vendors' vulnerability response efforts. In some cases, vendor bug bounty programs are enabled by other companies that provide tools and services to facilitate vulnerability coordination.

4. Phases of CVD

- **Discovery** – A researcher (not necessarily an academic one) discovers a vulnerability by using one of numerous tools and processes.
- **Reporting** – A researcher submits a vulnerability report to a software or product vendor, or a third-party coordinator if necessary.
- **Validation and Triage** – The analyst validates the report to ensure accuracy before action can be taken and prioritizes reports relative to others.
- **Remediation** – A remediation plan (ideally a software patch, but could also be other mechanisms) is developed and tested.
- **Public Awareness** – The vulnerability and its remediation plan is disclosed to the public.
- **Deployment** – The remediation is applied to deployed systems.

[Discovery]
Ultimately, we can't fix vulnerabilities we don't know about.

Many organizations hire application security testers or code auditors to look for vulnerabilities.

While such testing is certainly important and commendable, it is important to understand that absence of evidence is not always evidence of absence.

Finders should exercise an appropriate degree of care when performing vulnerability research.

This will help to alleviate legal concerns and limit the potential for damage to others. Likewise, organizations should make the rules and process for obtaining permission very clear and easy to find.

[Reporting]
Vendors need a mechanism to receive vulnerability reports from others. This reporting mechanism should be easy enough to use that it encourages rather than discourages reports. It can be as simple as a dedicated email address for reporting security issues, a secure web form, or a bug bounty program.

As a vendor, it is important to not treat reporters with suspicion or hostility. It's likely they have important information about your product, and they want to share it with you.

[Validation and Triage]
When a vendor or coordinator receives a vulnerability report, it's usually necessary to prioritize it along with other vulnerability reports already in progress, new feature development, and possibly other non-security bug fixes. But just because it was reported doesn't make it true.

Replication of the salient claims made in the report is an important step in the case handling process.

There are a number of heuristics for evaluating the severity of vulnerabilities. Perhaps the most commonly known of these is the Common Vulnerability Scoring System (CVSS). Vendors should ensure their analysts are trained in the chosen heuristic and understand its strengths and weaknesses so that its result can be overridden when necessary.

[Remediation]
Once the scope of the vulnerability has been adequately ascertained, it's time to prepare and test the fix (patch). The sequence of tasks tends to include identifying and isolating the vulnerability in the code; changing the code to eliminate the vulnerability; testing the changes, packaging the changes for distribution; and distributing the updated product.

[Public Awareness]
Knowledge of the existence of a vulnerability is often the key driver causing patches to be deployed. A silent fix is far less likely to be widely deployed than one that is clearly described.

Should you disclose at all? – Generally, the answer will be yes...

Many vulnerability reports can be similar, and sometimes a vendor or coordinator might receive multiple reports of similar vulnerabilities at the same time.

The most common identifier in use today is the CVE ID, which is meant as a globally unique identifier for a public vulnerability report. CVE IDs can be obtained from the CVE Project at MITRE or one of several CVE Numbering Authorities (CNAs) established by MITRE—typically the vendors of common software products themselves.

[Deployment]
Although we tend to think of the CVD process as ending with the disclosure of a vulnerability, if the fix is not deployed the rest of the exercise is futile. A patch that is quietly posted to a website and not well advertised is almost useless in protecting users from vulnerabilities.

Figure 2: Mapping CVD Roles to Phases

Roles → Phases	Finder	Reporter	Vendor	Corrdinator	Deployer
DISCOVERY	Finds vulnerabilities				
REPORTING	Prepares report	Reports vuls to vendor(s) and/or coordinators	Receives reports	Receives reports Acts as reporter proxy	
VALIDATION AND TRIAGE			Validates reports received Prioritizes report for response	Validates reports received Prioritizes report for response	
REMEDIATION		Confirms fix	Prepares patches Develops advice, workarounds	Coordinates multiparty response Develops advice, workarounds	
PUBLIC AWARENESS	Publishes report	Publishes report	Publishes report	Publishes report	Receives report
DEPLOYMENT					Deploys fix or mitigation

5. Process Variation Points

For those responsible for implementing the CVD process, defining a disclosure policy is an important first step. **A well-defined policy makes it clear what other participants in the CVD process can expect when they engage with you and establishes good relationships between finders, reporters, vendors, coordinators, and other stakeholders.**

Participants in Coordinated Vulnerability Disclosure iterate over the following questions:

1. What actions should I take in response to this knowledge?
2. Who else should I tell about it?
    1. What should I tell them?

The simplest instance of CVD is when there are only two parties involved: the finder of the vulnerability and the vendor who can fix the software. Most of the interesting cases in CVD involve more than two parties, as these are the cases where the most care must be taken.

Automation of the process can help somewhat, but the impact technology can have on the problem is limited by the inherent complexities involved in trying to get numerous organizations to synchronize their development, testing, and release processes in order to reduce the risk to users.

The FIRST Vulnerability Coordination SIG has published its "Guidelines and Practices for Multi-Party Coordination and Disclosure" which we strongly recommend reading.

The complexity of coordination problems increases rapidly as more parties are involved in the coordination effort.

6. Troubleshooting CVD

Any number of factors can lead to difficulty in making the initial connection between a would-be reporter and the party or parties that can do something about the vulnerability they want to report. Furthermore, even when you can find the vendor, not all vendors have established processes for receiving vulnerability reports.

The more transparent your process is—and the closer it is to what other folks are doing—the better you will be able to avoid problems. Good documentation is a start.

A presumption of benevolence is helpful when navigating the CVD process. Multiple things can go wrong in the disclosure process, but often these problems do not arise as a result of intentional acts of malice. So even if something has gone wrong, it's still good to give the benefit of the doubt to the good intentions of the involved stakeholders.

For vendors: A person who shows up at your door to tell you about a vulnerability in your product is not the enemy. That person is your friend.

7. Operational Considerations

Participating in a CVD process over time requires a set of tools and practices in order to be successful.

[Email vs. Web Forms]
Email is a simple, tried-and-true method of communication over the Internet. Although receiving information via email is convenient, it is not a very good mechanism for tracking multiple cases at once. Vendors...should consider setting up a web-based case tracking system instead.

Most vulnerability reports have a similar structure, making a web form a preferable method for receiving vulnerability reports for many organizations. To secure your web form, you will need to enable HTTPS and TLS, and obtain a TLS certificate from a Certificate Authority (CA).

[Bug Bounty Platforms]
A number of third-party CVD platforms now exist to facilitate communication between vendors and reporters. **Although they are often referred to as bug bounty platforms, often the "bounty" aspect is in fact optional—vendors can use bug bounty platforms to receive reports without needing to compensate reporters unless they choose to do so** [see HackerOne Response].

[Case and Bug Tracking]
Case tracking systems such as bug trackers or trouble ticket systems are often used by vendors, coordinators, and reporters for tracking vulnerability reports. Such systems can centralize the vulnerability response process and provide the ability to track individual cases. Case tracking systems also provide a means of collecting data about recurring security issues and the performance of the response process itself.

8. Open Problems in CVD

The units of work in CVD are vulnerability reports or cases. However, a single case may actually address multiple vulnerabilities. Teasing out how many problems are involved in a report can be tricky

at times. The implications of this in terms of the CVD process and the compilation of vulnerability databases is significant.

9. Conclusion

We do not live in an ideal world. In the world we find ourselves occupying, software-based systems exhibit complex behaviors, increasingly exceeding the limits of human comprehension.

We test for flaws, we probe for weaknesses, and we identify recurring patterns and themes that lead to undesired outcomes. We fix what we can, mitigate what we can't fix, and remain vigilant over what we can't mitigate. **We coordinate vulnerability disclosure because we realize we're all in this together.**

Additional Reading

We've summarized the document's bibliography to guide you to a few good resources, plus added a few others.

- SEI/CERT: How to Report a Vulnerability
- MITRE: Common Weakness Enumeration - a community-developed list of software weakness types
- Electronic Frontier Foundation: Coders' Rights Project Vulnerability Reporting FAQ
- U.S. Department of Defense: Vulnerability Disclosure Policy
- HackerOne: Vulnerability Disclosure Policy Basics: 5 Critical Components
- U.S. Department of Commerce: Multistakeholder Process for Cybersecurity Vulnerabilities
- U.S. Department of Justice: A Framework for a Vulnerability Disclosure Program for Online Systems
- U.S. Food and Drug Administration: Postmarket Management of Cybersecurity in Medical Devices
- U.S. National Telecommunications and Information Administration: "Early Stage" Coordinated Vulnerability Disclosure Template
- U.S. Federal Trade Commission: Start with Security: Lessons Learned from FTC Cases
- ISO/IEC 30111:2013: Vulnerability handling processes
- HackerOne: Vulnerability Disclosure Policy Basics
- HackerOne: The 5 Critical Components of a Vulnerability Disclosure Policy
- HackerOne: An Invitation to Hack: The Benefits and Risks of Vulnerability Disclosure and Bug Bounty Programs: Webinar with Wiley Rein LLP's Megan Brown and Matt Gardner
- HackerOne: 7 Steps to Hacker-Powered Security Success

Special thanks to Allen D. Householder and Art Manion for reviewing this document and allowing us to share their exceptional work.