Coordinated Vulnerability Disclosure
"Early Stage" Template and Discussion

NTIA Safety Working Group

Draft: November 4, 2016

Introduction

In the context of software vulnerability disclosure, safety critical manufacturers have relatively lower experience and higher consequences of failure. High trust, high collaboration interactions come from understanding mutual expectations and perspectives.

Software and other products are increasingly being examined and updated for security and safety. Nowhere is this more important than in the safety-critical industries on which we all depend.

Coordinated vulnerability disclosure is a way of channeling the energy and attention of the security research community into improving the safety and security of the broader community of users and the overall population.

Stakeholders representing a range of interests in this community recommend a considered approach which starts small to build experience, confidence, trust, and capacity. Firms contemplating their first steps into Coordinated Disclosure have many resources and references from multiple sources available to consult with as they develop their programs. What follows is a simple framing of what an ”early stage” coordinated disclosure program might look like.

The ultimate goal is to generate “high trust, high collaboration relationships with 3rd party researchers and their customers allowing them to improve their products, their customers' experiences, and even shareholder value.” This journey has taken many years for even the most sophisticated technical organizations. We present some insights into how those first few steps might be taken.

We believe this approach is especially important for safety-critical industries. The DMCA research exceptions went into effect in late October 2016. With softened fear of legal concerns, higher numbers of researchers are likely to engage in vulnerability research and disclosure. Organizations in those sectors should understand how the security research community may want to engage, and equip themselves with a flexible set of tools to successfully collaborate.

We define “safety critical industries” as those in which the potential for harm directly impacts humans. For example, an automobile, an embedded medical device (pacemaker or insulin pump), or carbon monoxide detectors.

When human safety is involved, the calculus of when and how to publicly disclose vulnerabilities is likely different than in other industries. Researchers may be more reluctant to disclose if they know a vulnerability has not been (or cannot be) fixed. In contrast, some researchers may be more likely to publicly disclose if they feel like it will motivate a vendor to fix vulnerabilities faster than going through their disclosure program.

Because safety critical systems may have higher consequences of failure, remediation should be handled both with all due haste and all due care. Remediation should be prioritized with a sense of urgency, to preserve safety, life, and trust. At the same time, validation and verification steps avoid unintended consequences. Any hard deadline for addressing vulnerabilities may both be too long and too short to safely address security vulnerabilities in safety critical systems.

Remediating vulnerabilities may take longer in safety critical industries than in other industries. This is because of the population that can potentially be impacted by updating software, and the level of danger that a partial or flawed update could have on a human - whose life and safety depends on a specific device. Vulnerability fixes may require more checks and testing to assure that updates do not result in (negative) unintended consequences. Additionally, the nature of software deployment especially in safety critical hardware means distributing a solution will likely take more time and effort than simply pushing code once, over the air, to repair all vulnerable devices.

We believe that human lives and health are so important to protect in safety critical industries that we encourage vendors and other parties who supply and build in this space to prioritize remediation. We urge this priority despite the likelihood that vulnerability remediation in safety critical industries is a more complex and lengthy process than fixing vulnerabilities in other industries. We also believe that researchers who are working to discover vulnerabilities in safety critical industries should become educated about the owner/operator's context and the vendors' priorities for mitigation, and emphasize the importance of protecting human users from harm and safeguarding their lives.

Below, we present a template of what a successful, lightweight, adaptable disclosure policy might look like, and then highlight some notable issues in developing such a policy. We also present a sample disclosure policy.

Template disclosure policy

We urge the creation/use of a simple, short document. These can fit on a single, readable page. Both automakers and medical device makers have already done this leveraging the template below.

Brand Promise

- Framework:
  - Audience: Customers and Market
  - Tone: "The safety & security of our customers is important to us…"
  - Optionally: Describe what you've already been doing
  - Optionally: a message to the researchers
    - "We are undertaking this program to give security researchers a point of contact so that they can directly submit their research findings so that they can be remediated quickly and efficiently."
    - Messaging to the research community is important.

Initial Program and Scope

- The term "Initial" is used to overtly suggest a program will evolve as capacity and confidence changes
- Framework:
  - Audience: Vulnerability Finders
  - Tone: Reasonable Phase 1 to build capacity
  - Explicit Scope:
    - Which Models (or Years) of Products and
    - Which Versions of Products and/or

        - Duration of scope (ie, until March 1, 2017)
    - Implicit Scope: Other Throttling mechanism (e.g. presence or lack of Recognition and/or Reward - see section below)
    - Optional: explicit out-of-scope
        - E.g. “Do not test on live patients.”
        - Danger of black-listing: could turn off some researchers, especially if it’s too long a list.

“We will not take legal action if…”

- Framework:
    - Audience: Vulnerability Finders
    - Tone: Non-threatening, inviting, reasonable (minimize legal-ese)
        - Many researchers won’t be lawyers or have access to one
    - List: Short, unambiguous, more affirmative than prohibitive
        - (with key exceptions like "Not on equipment in use")
        - We’ve noticed a BlackList is a slippery slope and can set wrong tone for collaboration
    - What does and doesn’t get you sued.
        - Just a simple “If you follow these guidelines (referring to the entire disclosure policy itself) we will not pursue or support any legal action related to your research” which you can argue is working with/against blacklisting.
    - Should strive to be immutable/evergreen/unchanging
        - See also “Changing Disclosure Policy” section below.
        - Expectations for researchers about critical existential or safety issues.
            - Provide a way for researcher to understand what is at stake.
    - Parties should consider applicable State, Local, and National/Federal Laws (for example: CFAA and DMCA)
    - This should contain legal postures *only* (submission preferences come later)

Logistics - how to submit and ongoing communication

- Framework:
    - Audience: Vulnerability Finders (message from recipient)
    - Mechanism: For submission (where, how to)
    - Requirements for submission acceptance (different than for legal posture)
    - Tone: Reasonable for initial exchange and
    - When the researcher might expect to:
        - Initially hear back
        - What will shape future engagement (no one-size-fits-all for bugs)
    - Expectations: Communication and Responsibilities of parties
    - Other: E.g. Conflict Resolution

Technical details: Nonbinding submission preferences and prioritizations

- Expectation management, and helping the researcher understand the preferences and priorities of vendor

- Note: this section is typically going to be more evolving, maintained by the support and engineering team.
- Framework:
    - Living, evolving list of report preferences
        - Scope & classes of vulnerabilities
        - Style of reports
            - E.g. - if you only provide crash dumps from a fuzzer, then...
        - Tools
    - Content
    - Note: Putting too many restrictions here may:
        - Send the wrong tone
        - Solution: Ask your organization - What are your vulnerability disclosure program goals?
        - Explicit scoping or report throttling
            - See above “Initial Scope” but/and limiting scope to that which your team can handle.
            - The better you define your inputs, the less onerous the filters necessary
    - Submitting out of scope may be ignored, but *won’t* trigger a legal issue
        - Difference between ‘scope’ and ‘boundary’

Issues to Consider in Writing a Disclosure Policy

Restrictions on disclosure

A vulnerability in a system is a fact. The fact that one researcher does not disclose its existence does not guarantee that another will not find it--or has not already found it. Finders may have reasons to want to disclose the vulnerability publicly. A managed disclosure situation is preferable to one without control. Vendors may want to express preferences on when finders publicly talk about vulnerabilities. A few options are:

Do not publicly disclose:

1. Until it is fixed
2. Until a particular time frame after first submission
3. Until after giving us X days notice.
4. Mutually agreed-upon (or negotiated) timeline (example: 60 days for software; 6 months for hardware)

There are strong pros and cons for denying researchers any ability to go public. For example, if you say “no disclosures can happen until the bug is fixed” there may be at less risk of exploitation, but many researchers will not participate.

- What if they fear a vendor “sitting” on a bug?
- What if the fix takes 5 years?
- Some researchers may expect very fast turn-arounds for bugs, but some affected industries can’t turn on a dime.

[Different vulnerabilities may call for different policies, making it hard to have just a single blanket policy.]
[Options]

Defining Vulnerability Disclosure [Program] Scope

Any newly implemented vulnerability disclosure program may need to deal with a large, unanticipated volume of submissions. In the early stage, report volume can be reduced through explicit or implicit scoping in the disclosure policy. This has the effect of focusing researchers on the specific type of disclosure items the company is prepared to respond to while they endeavor to build capacity and experience.

For example, submissions could be explicitly scoped by limiting the program to the following:

- Only specified product model years
- Only select product make/model/year
- Only particular types of vulnerabilities

Implicit scoping may be influenced by the type, structure, and scale of incentives that may be awarded to researchers, if any incentives are used at all. A Coordinated Vulnerability Disclosure Program with no reward program is likely to only attract more altruistic types or hobbyists who want to share their findings with the company, but are not looking to be rewarded. Adding modest recognition and/or reward to the program could expand the scope to increased researcher participation. Rewards such as providing recognition on a wall of fame board or awarding a challenge coin and/or branded merchandise creates some incentive attracting researcher to pursue finding vulnerabilities.

Researchers are motivated to understand security flaws for a wide range of reasons, from a desire to solve an interesting problem to a desire to protect other users. Benefits of narrowing scope or having no financial incentive for reporting vulnerabilities include limiting the number of reported vulnerabilities and that reporting researchers may have more patience and/or less motivation to disclose during conference presentations which have deadlines, dates and could conflict with the organizational process.

In summary, an organization can utilize explicit and implicit scoping mechanisms to match its capacity to implement its disclosure program. As the organization builds capacity and experience responding to vulnerability disclosures, it can scale its program accordingly. With maturation of the organizational response capabilities explicit and implicit scope limitations may be relaxed so that more useful disclosures might be obtained. Additionally, vulnerabilities that fall outside the program scope may deserve appropriate consideration and response.

Changing the disclosure policy

As with any policy, at some point it may need to be changed or modified. The side effect of changing the disclosure policy is that it can make things difficult for researchers to navigate and difficult for vendors to track, or cause researchers to lose confidence in vendor promises and difficult for vendors to track. As such, we recommend minimizing changes if possible. While that may not be possible, the legal protections offered to researchers should not change much over time if trust is to be maintained.

Given that policies may change, some strategies to maintain trust include:

- Be transparent - explain why the disclosure policy is changing
    - Accept feedback on changes / listen to community
- Explicit duration of any given policy: this is good until X
- Include version control
    - For any change made, and archive prior versions (maybe archive ALL versions on your site)
    - Avoid abrupt or erratic changes in the policy, update on consistent time periods
- Researchers enroll, and become grandfathered into a given policy version
    - This puts a lot of responsibility on to the researcher and the vendor to track which policy version is being used.
    - [Light version: have a feed or email list for updates]
- Include explicit caveats about how the policy will change
    - This may result in a very long and complicated policy
    - Black lists will invariably grow
    - Potential solutions: white listing over black listing
- Declare certain parts of the policy immutable, *particularly legal protections and promises.*
    - Have a baseline - everything above this point is the basics, won’t change
        - Baseline = white list
        - Tie in with brand promise?
        - Should reflect high level goals of program or impacts of vulnerabilities rather than technical approaches?
        - Changes to whitelist (adding or removing) should be rare and accompanied with an explanation for the change
    - Here is the section that we may change - establish what might trigger a change

        - Changing = blacklist
        - May be used to throttle common or “low effort” vuln reports
        - May change as a result of enhanced security engineering / qa process
        - May be used to shift the focus to the newest or riskiest product
    - Can encourage researchers to check back, and archive what they [?]
    - Can subscribe to an RSS feed of updates

Resolving these issues will help inspire confidence among researchers, which will help promote the success of the policy.

Sample Vulnerability Disclosure Policy Template

ACME Corp.

ACME Corp., the leading manufacturer of embedded software widgets, is committed to ensuring the safety and security of our customers. Towards this end, ACME now formalizing our policy for accepting vulnerability reports in our products. We hope to foster an open partnership with the security community, and recognize that the work the community does is important in continuing to ensure safety and security for all of our customers.

We have developed this policy to both reflect our corporate values and to uphold our legal responsibility to good-faith security researchers that are providing us with their expertise.

Initial Scope
ACME’s Vulnerability Disclosure Program initially covers the following products:

- ACME Widgetsoft 3.1
- ACME Widget Module A
- ACME Widget Module B
- ACME Widget Controller
- ACME Widget Ethernet Gateway Module

While ACME develops a number of other products, we ask that all security researchers only submit vulnerability reports for the stated product list. We intend to increase our scope as we build capacity and experience with this process.

Researchers that submit a vulnerability report to us, once accepted and validated by our product security team, will be given full credit on our website.

Legal Posture
ACME Corp will not engage in legal action against individuals that submit vulnerability reports through our Vulnerability Reporting Form. We openly accept reports for the currently listed ACME products. We agree not to pursue legal action against individuals who:

- Engage in testing of systems/research without harming ACME or its customers.
- Engage in vulnerability testing within the scope of our vulnerability disclosure program and avoid testing against [ex. website].
- Test on products without affecting customers, or receive permission/consent from customers before engaging in vulnerability testing against their devices/software, etc.
- Adhere to the laws of their location and the location of ACME. For example, violating laws that would only result in a claim by ACME (and not a criminal claim) may be acceptable as ACME is authorizing the activity (reverse engineering or circumventing protective measures) to improve its system.}
- Refrain from disclosing vulnerability details to the public before a mutually agreed-upon timeframe expires.

How to Submit a Vulnerability
To submit a vulnerability report to ACME's Product Security Team, please utilize the following form <link to vulnerability reporting form>

Report Acceptance Criteria
We will use the following criteria to decide whether or not to accept the report. Report declines mean that the report was not of sufficient quality or was out of scope.

What we would like to see from you:

- Well written reports in English will have a higher chance of being accepted.
- Reports that include proof of concept code will be more likely to be accepted.
- Reports that include only crash dumps or other automated tool output will most likely not be accepted.
- Reports that include products not on the covered list will most likely be ignored.
- Include how you found the bug, the impact, and any potential remediation.
- Consideration for vulnerabilities that may have safety impact.
- Any plans for public disclosure.

What you can expect from us:

- A timely response to your email (within 2 business days).
- An open dialog to discuss issues.
- Notification when the vulnerability analysis has completed each stage of our review.
- An expected timeline for patches and fixes (usually within 120 days).
- Credit after the vulnerability has been validated and fixed.

If we are unable to resolve communication issues or other problems, ACME may bring in a neutral third party (such as CERT/CC or ICS-CERT) to handle the vulnerability, or may encourage you to disclose the vulnerability publicly.

[*Versioning*]
This document was created 26-July-2016. [We update or renew this policy every 90 days] Any updates will be noted below in the version notes.