NTIA Multi-party* disclosure WG

(*vendor)

Scope

Vulnerability disclosure that affects multiple vendors, with a focus on activity by vendors, less by other stakeholders.

(References to “multi-party” primarily mean “multi-vendor.” “Party” is used to cover non-vendor roles, such as a coordinator.)

Problem Statement

There is no broad consensus on vulnerability coordination and disclosure best practices for vulnerabilities affecting multiple vendors.

Background

Development is increasingly likely to involve interdependencies between components (products, frameworks, libraries, services, devices, etc.) from different vendors or providers. This means that vulnerabilities and coordination efforts are more likely to involve multiple vendors. Interdependencies are often related to supply chain systems. There are different types of multi-vendor disclosure, usually related to the nature of the interdependencies.

Common examples include

Multi-vendor vulnerability		**Example**
1	Common components, likely free and open-source, often libraries One component with many users, breadth/horizontal	Heartbleed
2	Multiple vendor layers, shared responsibility Depth/vertical	Stagefright AOSP, OEM, provider

3	Service provider, X-as-a-Service Provider responsibility for some services, user for others	Virtualization or framework vulnerability
4	Development component Users need to take manual action	Microsoft ATL, Java de-serialization

Problems

Current state, thinking, practices, issues.

Information Sharing

In coordinated multi-party disclosure, defenders attempt to share privately with each other before attackers (and the general public) become aware. A common goal is to have fixes prepared and possibly deployed before attackers discover and exploit the vulnerability. The assumption that embargos are successful at limiting knowledge by attackers should be considered carefully. Common information security information sharing problems apply. Even more generally, keeping secrets is hard.

- Trust: Information sharing usually only works if participants trust each other and share expectations of privacy and use/further sharing. Without trust, Process and OpSec don't add much.
- Process: Format, protocol, procedures for sharing. Rules for sharing club, penalties for not following rules.
- OpSec: Keeping information confidential, following sharing rules.
    - vendor-sec being target for hackers
    - distros@openwall hosted on a server in Moscow(?)
    - PGP is hard, doesn't scale

The more vendors/parties are informed, the greater likelihood of a leak.

Three states of embargo:

1. Intact, only vendors/defenders know
2. Public leak, some defenders knew before the leak
3. Silent leak, embargo believed to be in force, but isn't

State #1 is more rare than one might think? Hard to keep secrets, state #3 is worse than #2. Attackers may independently discover a vulnerability, or learn about it through a leak.

Vendors (sharing club/embargo participants) may be concerned about being excluded from a coordination event, related to time needed to develop fixes. Pressure to comply with slowest vendor, since faster vendors fear being excluded from the next embargo if they try to protect their user base quickly. At least one vendor, OpenBSD, will not wait for embargo periods (http://www.openbsd.org/security.html).

Value Proposition

How much effort is sufficient (or worthwhile) for a given vulnerability? If vulnerabilities aren't special/unique/sparse, then going to lengths to keep one under wraps becomes less valuable (because attackers are likely to independently discover it or a different but equivalent vulnerability).

Complexity

Even in a case of single researcher, single vendor, single vulnerability, multiple products, versions, and release cycles can make preparing and deploying fixes complicated. What appears to be a single vendor vulnerability may still involve multiple business units or external vendors.

Adding multiple vendors adds complexity to the coordination process.

Perceptions

Perceptions of unknown lead to different policy. How do you perceive/believe:

- Ability of embargoes to maintain integrity
- Fairness of embargoes
- Chance that bug has been independently discovered
- Coordination effort of a bug is worth security improvement
- Existence of as-yet-undiscovered bugs

Unknown unknowns about attack activity contributes to inability to agree on a/the best disclosure plan. Can we reduce the gap between perceptions and reality? Many of these unknowns are difficult to measure.

Considerations

Keep the overall problem in mind: Lack of agreement on if and how to conduct multi-vendor/phased disclosure.

If you're going to try anyway, consider:

Fairness

- Does it need to be fair? Fair to whom?
- Participant ability to abide by sharing rules?

Effectiveness

- By what measure? Minimizing harm? Ability for vendor/provider to protect users?
    - Ability to push fixes, detection, or protection?
    - To large numbers of end users?
    - To providers or administrators (that can in turn protect users)?
    - To CIKR, other critical sectors?
    - To protection vendors/providers?

Whom to inform

- Other vendors, major customers/users?
- Protection vendors (IDPS, AV)?
  Note Microsoft MAPP (https://technet.microsoft.com/en-us/security/dn467918.aspx).
- Public safety, government, CIKR, regulators?

How long to wait

- Time between notifying vendor and producing fix
    - Identifying and notifying *all* vendors and waiting for *all* vendors to produce fixes?
- Time between producing fix and fix being fully/widely/significantly deployed
    - For all vendors/users?

Write down the criteria, possibly publish it so others understand how they will or will not be included.

As a minimal starting point, is there consensus for:

1. Vendors have (and publish) policies
2. Vulnerability is reported to implementer
3. Vendor (implementer) produces fixes (implementer ~= developer, creator)
4. *Vendor (implementer) informs others privately and sets embargo?*

5. Vendor (implementer) publishes fixes and vulnerability information
6. Downstream users (vendors, customers, users) monitor for and adopt fixes or otherwise respond

Without step #4, there really is no coordination. What if there are more than one implementers of a standard/protocol/format, and each is vulnerable? In this case step #2 becomes coordination.

Note OpenSSL policy (https://www.openssl.org/policies/secpolicy.html).

Suggestions

Survey existing practices? (ENISA might be doing this, CERT has some survey data)

Document problems/challenges, why previous agreement on previous standards/efforts has not been reached (with multi-vendor focus, also connect with Adoption/Awareness SWG).

Overall guidance should be high level principles (with consensus?), whereas prescriptive guidelines that may not serve all stakeholders/or be agreed upon (~30 year debate) should be considered and discussed, but not impede the overall process.

Attempt to find and document consensus on principles that help reduce risk in general, versus forcing one disclosure philosophy over another?

Provide support/advice for small/OSS/library projects (that are often at the root of a multi-vendor vulnerability).

Integrate multi-party disclosure into larger discussion? Or separate output?

2015-11-10 Meeting

Attendees

Amanda (Microsoft)

Chandan (Juniper)

Katie (HackerOne)

Art (CERT/CC)

Paul (Microsoft)

Kymberlee (BugCrowd)

Bruce (Oracle)

Ben (Google)

Meeting Notes

Next NTIA vulnerability disclosure meeting Dec 2, 10AM-4:30 PM, DC

ISO

Draft of ISO 29147 WD 2 to share with experts at Dec 2 NTIA meeting.

Trying to get 29147 made available at no cost.

Desire for ISO to generally match other disclosure policy/practices, or vice versa.

Researcher representation

Does this WG have sufficient researcher representation? If not, WG members should feel free to recruit a researcher or two. Note vendor focus.

Scope

Multi-vendor coordination

Multi-party, “party” to include coordinators, or other non-vendors, such as providers?

Multi-**vendor** is the focus of this WG (multi-stakeholder, all stakeholders, is the scope of the broader NTIA process)

At least two common scenarios:

1. Heartbleed-like, 3rd-party code/library (OpenSSL, libpng) used by many downstream vendors
2. the second scenario, hardware, software, services supply chain - e.g. all mobile devices with chip, phone, OS, app developers, mobile carriers
3. IoT-ish supply chain issues, like Android (OS, hardware, provider, end user)
4. Service/provider/XaaS vs. box product

Open Questions

Does this WG produce/provide input to other efforts, or does this WG produce “final product” documents?

Out of Scope

- Disclosure timelines
- Receiving vuln reports - part of ISO 29147 already - can be on Adoption WG scope.
- Fixing vulnerabilities internal to an org - part of ISO 30111 already - can be on Adoption WG scope.
- Publishing advisories - part of ISO 29147 already - can be on Adoption WG scope.

Issues / Challenges

More people in the know, greater chance of premature public disclosure

Phased notification/disclosure?

Deconfliction of disparate disclosure timelines requested

Inclusion of “defenders” -- AV, IDPS, network operators?

Who can coordinate disclosure amongst multiple parties/vendors?

- Where can researchers get assistance with coordinating disclosure to multiple parties
- Where can vendors get assistance with coordinating disclosure to multiple parties

Researcher reports to one vendor, turns out multiple vendors are affected, how is communication with researcher handled? By first vendor? Does first vendor act as coordinator with other vendors? Outside coordinator?

- **A problem statement**

DRAFT PROBLEM STATEMENT: There is no broad (industry) consensus on how vulnerability coordination and disclosure best practices should be structured for vulnerabilities affecting multiple parties, or due to the interdependencies between software solutions, their components, or services, and/or devices.

- **Some useful background on the issue**

Examples of multi-vendor/party coordination cases, like Heartbleed, ATL, Apache commons, a recent Android thing, SSL Renegotiation?

Significant additional complexity with multiple vendors involved

Current model (1): developer fixes, then full disclosure (see current OpenSSL disclosure policy)

Current model (2): ICASI USIRP and distros@ protocol - broadcast, multicast, unicast within a trusted group.

How to decide whom to notify and when (phased notification/disclosure?)

- **An approach, or multiple approaches to addressing the problem**
    - Existing efforts
        - ISO 29147 and 30111

        - FIRST Vulnerability Coordination SIG - https://www.first.org/global/sigs/vulnerability-coordination
        - CERT/CC and some national CSIRTs
    - Approach to identifying existing challenges is to look at ISO 29147 and 30111 mapping (ISO 29147 Figure 1) and highlight where support for multi party coordination needs to be added, highlighted, updated
- **Subcomponents that may need to be addressed**

Weaknesses with current coordination models:

- vendor-sec being target for hackers
- distros@openwall hosted on a server in Moscow!
- **Milestones towards a solution.**

Minimal starting point:

developer/upstream produces fixed software

user/downstream monitors and brings in fixes

(ref OpenSSL?)

Especially for upstream developers/vendors, at a minimum publish your policy, so downstream/users/vendors understand how they'll be receiving fixes

note that there is no agreement on if or how to run a multi-vendor embargo?

role of coordinator?

One may choose to notify privately/before public disclosure, without being prescriptive, would we describe factors that go into such choices?

- Evidence of exploitation (attackers know and are using)
- Speed/ability of affected users to adopt/deploy fixes
    - Cloud/service vs box,
        - Box auto-update vs. user/manual
- Perceived ability of embargoes to maintain integrity
- Perceived fairness of embargoes

perception of unknowns leads to different policies

unknown unknowns about attackers contributes to inability to agree on a/the best disclosure plan

Overall guidance should be high level principles that can be agreed, whereas prescriptive guidelines may not serve all stakeholders/or be agreed upon (~30 year debate) should be considered and discussed, but not impede the overall process.

Agreement of principles that help reduce risk in general, versus forcing one disclosure philosophy over another.

Concerns over embargo exclusion, tied to time to fix:

Pressure to comply with slowest vendor, whereas faster vendors fear being excluded from the next embargo if they try to protect their user base quickly.

Time between notifying vendor and producing fix

Time between producing fix and fix being fully/widely/significantly deployed

    Critical infrastructure

    Inform IDPS/detection vendors to provide stopgap before fixes are deployed?

worth describing problems/challenges, why previous agreement on previous standards/efforts not achieved (with multi-vendor focus)

collect on-the-ground practices

Where does multi-vendor hit ISO 29147/30111?

Support/advice for small/OSS/library projects

Three states of embargo:

4. intact, only defenders know
5. public leak, some defenders knew
6. silent leak, embargo believed to be in force, but isn't

#1 is rare, hard to keep secrets, #3 is worse than #2

With multi-version, multi-release updates on multi-products it becomes even more involved in attempting to coordinate a synchronized security update for a multi-vendor component.

Different product in different release cycle, varying methods of update capabilities, etc. make it extremely difficult, especially with common 3rd Party usage issues, SSL being one, to coordinate a single disclosure which could potentially identify a product that may be impacted but is still in the process of being updated.

A single vendor disclosure already has issues with different versions being supported, different release cycles, product relationships/dependencies, that are all complicating factors, before adding the multi-vendor angle.