Forum of Incident Response and Security Teams, Inc. (FIRST.Org)

Summer 2017 | 17

Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure

This document is open for public comment. For more information, see <https://first.org/global/sigs/vulnerability-coordination>.

Table of Contents

Introduction

Events in the recent past have highlighted the need for real improvements in the area of vulnerability coordination. Historically, foundational work on best practices, policy, and process for vulnerability disclosure have focused on bi-lateral coordination and did not adequately address the current complexities of multi-party vulnerability coordination. Factors such as a vibrant open source development community, the proliferation of bug bounty programs, third party software, and the support challenges facing CSIRTs and PSIRTs or bug bounty programs are just a few of the complications. Examples such as Heartbleed[1] highlight coordination challenges.

This document is the outcome of an effort between the National Telecommunications and Information Administration (NTIA)[2] and FIRST to address such challenges. The purpose of this document is to assist in improving multi-party vulnerability coordination across different stakeholder communities.

The Industry Consortium for Advancement of Security on the Internet (ICASI) proposed to the FIRST Board of Directors that a Special Interest Group (SIG) be considered on Vulnerability Disclosure. After holding meetings at the FIRST Conferences in Boston and Berlin, ICASI formally requested FIRST to charter a SIG to review and update vulnerability coordination guidelines. The first part of this work is collaboration with the National Telecommunications and Information Administration (NTIA) to address multi-party coordination. Subsequent work will address bi-lateral coordination and approaches to notification.

This document differs from the ISO Vulnerability disclosure and handling standards (ISO/IEC 29147 and ISO/IEC 30111) in that the ISO standards provide basic guidance on the handling of potential vulnerabilities in products. This document is a collection of best current practices that consider more complex and typical real-life scenarios that extend past a single researcher notifying a single company about a discovered vulnerability.

This document is a compendium of coordination resource documents and recommended methods for reporting/updating coordination directories. The guidelines contain a common set of 'guiding concepts', and vulnerability coordination best practices that include use cases or examples that describe scenarios and disclosure paths. This document is targeted at vulnerabilities that have the potential to affect a wide range of vendors and technologies at the same time.

[1] http://heartbleed.com/
[2] https://www.ntia.doc.gov/other-publication/2016/multistakeholder-process-cybersecurity-vulnerabilities

Definitions

Within the context of this document, the following definitions apply. Definitions that are available in ISO/IEC 29147:2014[3] are used with minimal modification.

Advisory: Announcement or bulletin that serves to inform, advise, and warn about a vulnerability of a product.

Coordinator: Optional participant that can assist vendors and finders in handling and disclosing vulnerability information.

Defender: Stakeholder who is responsible for defending against attacks. A defender can be a system administrator, vendor, or provider of defensive technologies or services. Defenders may detect vulnerable systems, detect and respond to attacks, and perform vulnerability response and management.

Disclosure: Act of initially providing vulnerability information to a party that was not believed to be previously aware. The overall disclosure process typically includes multiple disclosure events.

Exposure: Time between the discovery of a vulnerability and the time a vulnerability can no longer be exploited.

Finder: Individual or organization that identifies a potential vulnerability in a product or service. For example, a finder may be an external security researcher.

Mitigations: Actions that reduce the likelihood of a vulnerability being exploited or the impact of exploitation.

Remediation: Patch, fix, upgrade, configuration, or documentation change to either remove or mitigate a vulnerability.

Vendor: Individual or organization that developed the product or service or is responsible for maintaining it.

Peer Vendor: Vendor at the same horizontal level of the supply chain. Peer vendors may be independent implementers of the same technology (e.g., OpenSSL and GnuTLS) or downstream users of the same upstream technology (e.g., Red Hat and SuSE).

Upstream Vendor: Vendor that provides a product or technology to a downstream vendor.

Downstream Vendor: Vendor that receives a product or technology from an upstream vendor for use in the downstream vendor's product, technology, or service.

---

[3] http://www.iso.org/iso/catalogue_detail.htm?csnumber=45170

Vulnerability: Weakness in software, hardware, or a service that can be exploited.

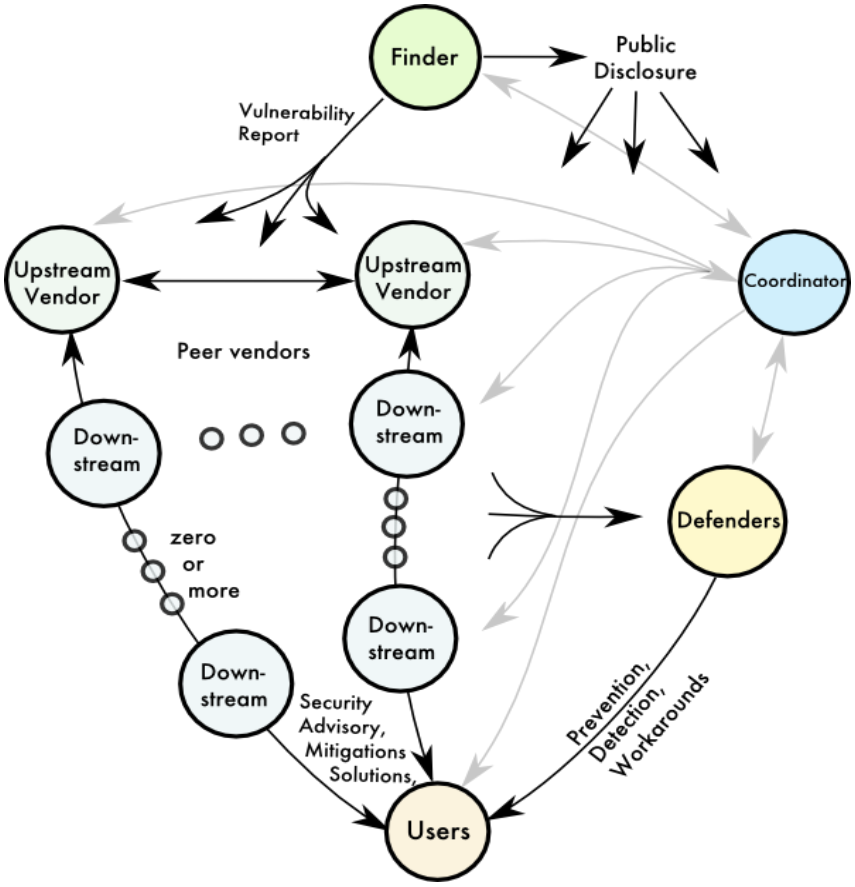

Figure 1: Stakeholder roles and communication paths

Figure 1 shows the relationships and communication paths between stakeholder roles.

Multi-Party Disclosure Use Cases

Vulnerability disclosure can be a complicated process, especially when multiple parties (usually multiple vendors) are involved. This section of the document is organized as a set of vulnerability disclosure use cases, in rough order, from simple to complex. Significant attention is given to coordinated, multi-party disclosure (see Use Case 2: Vulnerability with coordinated disclosure). Disclosure often deviates from the expected or ideal process, so within each use case are variants that are common exceptions to the ideal use case. Within each variant are causes, preventions, and responses. The collected set of preventions and responses are presented as practices that can be used to reduce the occurrence and cost of expected variants. Practices are denoted as strong recommendations ("should") or suggestions ("can," "could," or "may"). After describing multi-party coordination and disclosure use cases and variants, recommended practices are collected into the concluding section: Guiding Concepts and Best Current Practices.

Use Case 0: No vulnerability

Description

This case is included for completeness, if there are no vulnerabilities, there is no need for coordination.

Use Case 1: Vulnerability with no affected users

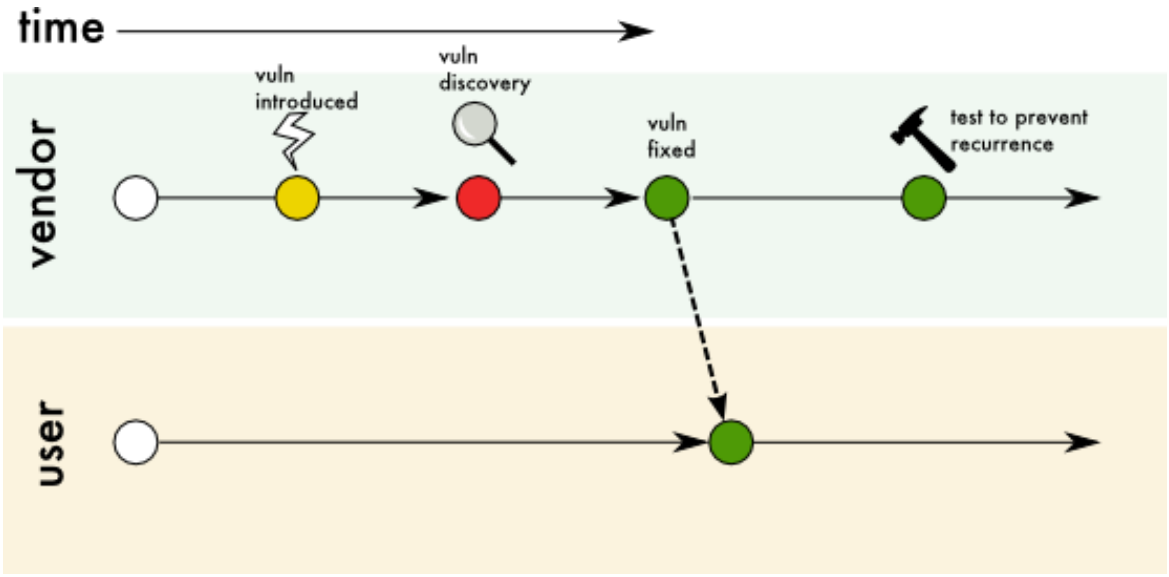

Figure 2: Use Case 1 Vulnerable product, but no affected users

Description

A vulnerability software or hardware with no users is a security hole that does not affect anyone else in any way. Examples of this case include: products that are (a) non-production, experimental (e.g., webgoat), (b) internal or for personal use, (c) never published or sold, or (d) under development.

Vulnerability is discovered and fixed before the product is deployed. Vendor takes steps to prevent recurrence of the vulnerability. No advisory required for users.

Coordination is not required, except:

- When the vulnerability can potentially exist in a similar product, protocol, or algorithm.
- When the vulnerability represents a new class of weaknesses not previously known.
- When the vendor is not reachable, but coordination with other affected stakeholders is taking place.
- When the vendor and researcher disagree.

Variant 1: Product is deployed before vulnerability is discovered or fixed

Description

The product is shipped and available with one or more existing vulnerabilities. The vendor discovers the vulnerabilities and corrects them. The vendor releases an updated version of the product and takes steps to prevent reoccurrence. The vendor, then, publishes an advisory.

Causes

- The affected product is not well tested.
- The affected product is deployed too soon.
- The affected product is deployed with known vulnerabilities.

Prevention

- Perform product penetration testing and or/scanning for known vulnerabilities prior to release.
- Establish bug bounty programs to proactively identify vulnerabilities prior to release. Set clear expectations and baselines on beta quality versus ready for release requirements.

Use Case 2: Vulnerability with coordinated disclosure

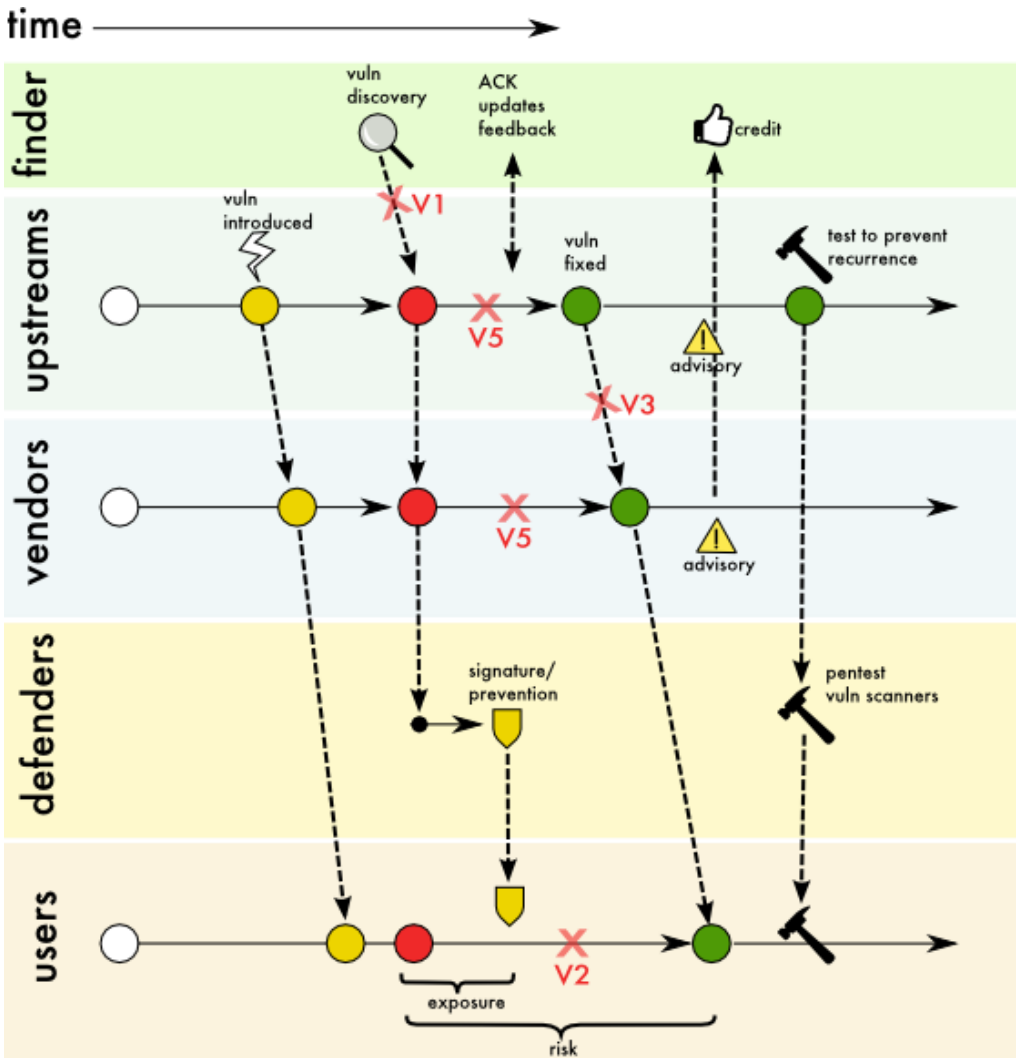

Figure 3: Vulnerability with coordinated disclosure

Description

Many security vulnerabilities are discovered after the product is released. Multiple stakeholders such as finders, upstream vendors, vendors, defenders, and users are involved in the coordinated disclosure effort. Stakeholders are encouraged to follow some guidelines set out by international bodies like ISO, to formulate the basis of their disclosure practice.

In a generalized coordinated disclosure process, the following stakeholders perform certain roles.

Finder

- Finder contacts the vendor using standard vulnerability reporting channels.

Vendors

- When vendors fix the problem, they communicate with upstream and downstream vendors at appropriate times as required.
- Vendors publish advisories as warranted.

Defenders

- Develop mitigations or signatures to detect and defend the users against vulnerability, without containing or inferring information that may assist a potential attacker.
- Request relevant test-cases from vendors to detect advanced threats based on recurring patterns.

Users

- Deploy vendor patch or mitigation as soon as possible.

Variant 1: Finder makes the vulnerability details public prior to remediation

Description

There may be instances in which a finder publicly releases details of a vulnerability prior to remediation, which can increase risk to affected users. Although a known active exploitation may prompt the finder to publicly disclose prior to remediation, other causes for disclosure include inability to establish contact with vendor and financial or other motivations for finder disclosure. Preventing public release prior to remediation is ideal, but in cases where early public release happens, quick response and communication of potential mitigations is paramount.

Causes

- Finder is unable to locate a vendor contact.
- Vendor does not respond to finder.
- Finder and vendor do not agree that report is a vulnerability (e.g., Vulnerability exists in an unsupported version of the product, but is fixed in the supported version of the product).
- Finder discloses to create pressure on vendor to fix or on the disclosure timeline.
- Finder is motivated by profit (e.g., finder's motivation is to sell a product or service that may detect or defend against the vulnerability).
- Finder is motivated by public recognition or fame.
- Miscommunication occurs between finder and vendor.
- Finder is insensitive to consumer security concerns.
- Finder believes vendor is insensitive to consumer security concerns.
- An active exploitation of the vulnerability is discovered.
- Vendor does not remediate the vulnerability.
- The number of vulnerable vendors it too large for the finder to deal with.
- Finder is concerned with legal issues associated with contacting vendor.

Prevention

- Vendors should provide currently accepted contact mechanisms, such as security@ email addresses and "slash security" (/security) web pages.
- All parties involved (including vendors, finders, and coordinators) should communicate their disclosure plans.
- All parties involved should provide their disclosure policies.
- There should be frequent communication with finder (including regular status updates).
- A coordinator can offer to analyze the vulnerability and educate either the vendor or the finder.
- Vendors can offer incentives such as safe harbor, credit, or bug bounties.
- All parties should avoid escalation to any extent possible (including legal action).
- All parties should advocate the Principle of Least Exposure.
- Vendors and coordinators should maintain an outreach program with finder community.
- Vendor should avoid individual points of failure for communication.
- When a larger number of vendors are involved, a coordinator can support communication and coordination between the vendors.

Response

- Contact finder to review vendor's coordinated disclosure policy.
- Express disappointment to the finder, yet remain positive while attempting to contain further leaks.
- Vendor may contact media.
- Vendor can align internal resources to patch the vulnerability with top priority.
- Vendor and/or finder may engage with a coordinator to mediate in case of disagreement.
- Vendor may provide mitigation advice to users through use of security advisory or blog.

Variant 2: Users do not deploy remediation immediately

Description

Providing remediation alone is not sufficient to reduce risk, deployment is also necessary. There may be instances in which users do not deploy either the remediation or the vendor suggested mitigations immediately after being made available by the upstream vendor. In general, users are strongly encouraged to apply, where possible, a risk-based approach in deciding how quickly they should deploy vendor-supplied remediation or mitigations when made available to help reduce potential risk of exploitation. Vendors responsible for issuing remediation or mitigations for critical and high severity vulnerabilities should communicate the availability of such, as broadly as possible, along with clear deployment and recommendations.

Causes

- Vendor has a history of providing low quality or untrusted security updates.
- It takes time and resources for users to test and deploy.
- Automatic patch updates are not available from the vendor.
- Automatic vendor patch updates are not enabled by the user.

- Older end-of-life/end-of-support version is installed and no security fix for that version/build will be released by vendor.
- Users do not fully understand the threat or criticality of the vulnerability.
- Users wait for multiple or bundle patches from the vendor.
- Users are not aware of the supply-chain for and components used in their systems.

Prevention

- Vendor can release fixes on a predetermined schedule (e.g., Patch Tuesday).
- When possible, vendors should not include non-security updates with security fixes (e.g., JRE model).
- Vendor should offer an automatic update process for users if possible.
- Users should enable automatic vendor patch updates if available.
- Vendor should test updates rigorously prior to security fix release.
- Vendor should publish the high-level version of their Secure Design Lifecycle (SDL) processes and publish disclosure policies to re-assure users.
- User should remove end-of-life/end-of-support systems from their environment.
- Vendor should eliminate extended support to legacy product versions that cannot be properly maintained and updated.
- Ensure product security advisory is clear on severity of the vulnerabilities, the impact of a successful exploitation, and the location of available download.
- Vendor may want to consider providing a bill of materials that includes information about third-party components.

Response

- Vendor should adopt a vulnerability scoring system standardization mechanism (e.g., Common Vulnerability Scoring System) to raise awareness for users on the severity of the vulnerability.
- Vendor should provide clear advisories and bulletins in machine readable format related to the vulnerability and fixes/remediations or mitigations.
- Vendor should provide any available mitigations or workarounds even if may cause some degradation of service.
- When possible, vendors should audit user's landscape and send a reminder if remediation has not been deployed.
    - Provide 1:1 support to critical users to break the trust-barrier and expedite remediation adoption.
    - Vendor can leverage existing customer support and sales channel to effectively communicate security bulletins to their users.
    - Vendor can inform their customer representatives through internal notification process so they can encourage customers to apply remediation.

Variant 3: Missing communication between upstream and downstream vendors

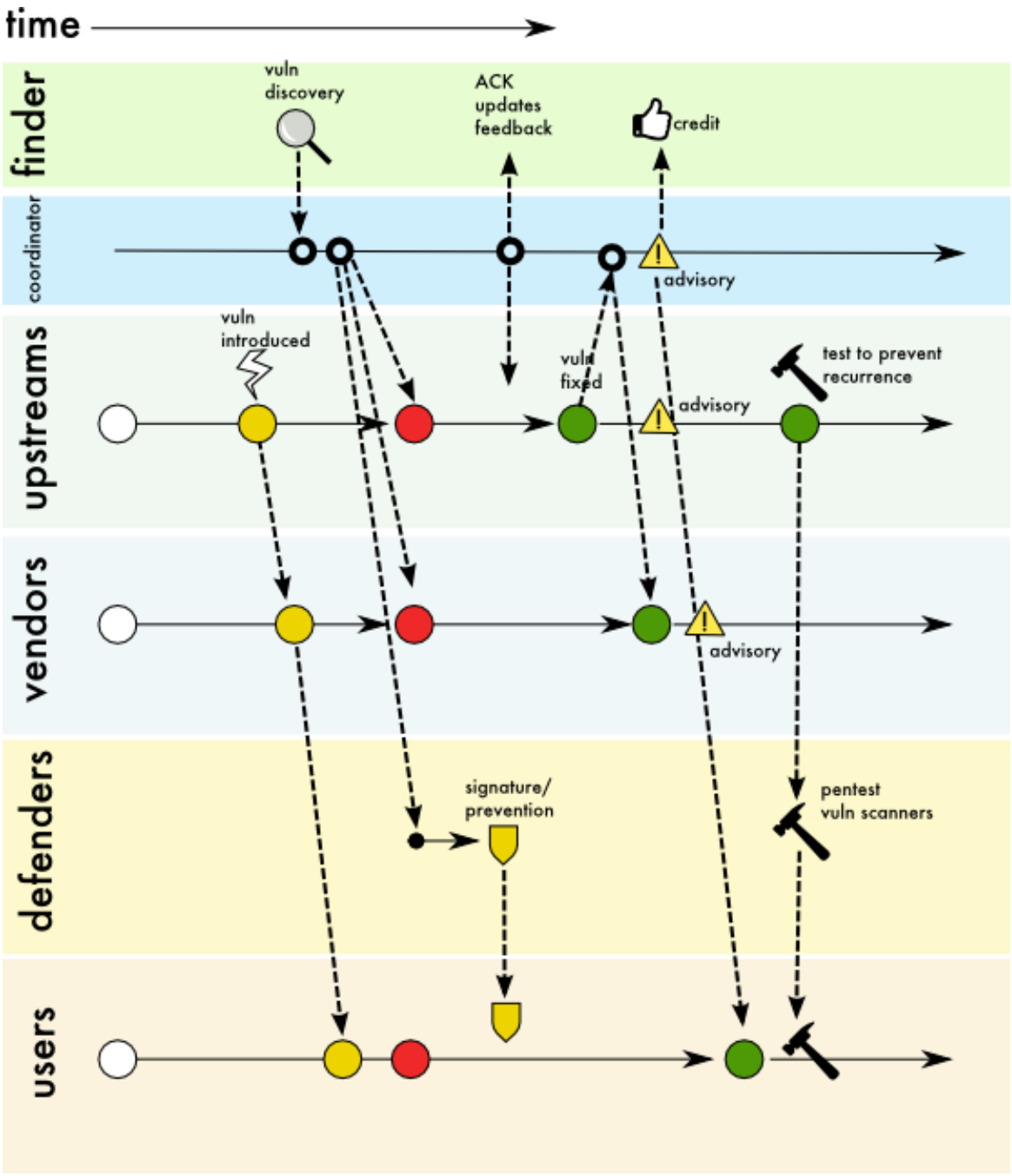

Figure 4: Use Case 2, Variant 3 Missing communication between upstream and downstream vendors

Description

Direct communication or a security disclosure could be missing between upstream vendors and downstream vendors or between vendors and users. A coordinator could facilitate receiving and distributing information back and forth to relevant parties at the various stages of remediation.

Causes

- Vendor fails to recognize vulnerabilities internally (e.g., a vendor may not track the vulnerabilities in third party components of their product).
- Vendor does not fully understand or is not aware of all downstream stakeholders.
- Vendor corrects the vulnerability,but does not inform all downstream stakeholders.
- Vendor fails to pre-establish trusted communication channels or NDAs with downstream stakeholders.
- Vendor fails to allow for sufficient downstream coordination and propagation time prior to public disclosure by the vendor.
- Vendor fails to communicate disclosure timeframe and set expectations with downstream stakeholders.

Prevention

- Vendor to establish an actionable public vulnerability coordination and disclosure policy, ideally describing the threshold for disclosure (e.g. severity).
- Vendor should consider communicating remediation/mitigations of all vulnerabilities regardless of severity rating or source of vulnerability report.
- Downstream vendors should consider keeping their components in-sync with upstream recommended release. Selectively patching security vulnerabilities can become tedious, error prone and expensive in the long run as source code can diverge between upstream and downstream instances. Downstream vendors may also miss security improvements or vulnerability fixes that do not get CVE assignments or get CVE assignments at a later date (e.g., CVE-2016-2108[4]).
- Vendor should consider tracking the use of third party components to develop better inventory and understanding of upstream and downstream dependencies.
- Vendor should pre-establish an upstream downstream trusted network for rapid communication and coordination (e.g., mailing lists such as the UEFI USRT).
- Vendor should clearly communicate disclosure timelines to downstream vendors.
- Vendor should anticipate the timeframes needed for downstream coordination.

[4] OpenSSL CVE-2016-2108: A vulnerability was fixed in OpenSSL June 2015 releases, but was not recognized as a vulnerability until May 2016. Downstream Vendors who upgraded their OpenSSL code base to the latest stable release in June 2015 had effectively resolved this vulnerability eleven months ahead of vendors who selectively patched only the CVE assigned vulnerabilities.

- Vendor could leverage coordinators for communication and coordination in the following ways:
    - A coordinator may receive a vulnerability report from a finder that affects multiple vendors and then distribute that report to affected upstream and downstream vendors.
    - A coordinator may receive a vulnerability report and resolution information from a vendor and help identify other affected vendors, possibly peer vendors and relay the information to them.
    - A coordinator may refer to the vendor directory to determine affected vendors.
    - A coordinator may also inform defenders at appropriate times to help mitigate or prevent attacks.
    - A coordinator may publish a public advisory in addition to vendor advisories to create awareness about the vulnerability and available remediation.

Response

- Vendor should identify a dedicated contact for upstream and downstream stakeholders, in addition to communicating via generic e-mail, like secure@example.com.
- Where possible, vendor should explain the situation to affected stakeholders to build transparency.
- Vendor should negotiate an agreed time frame with affected stakeholders prior to vulnerability disclosure.
- Vendor could leverage coordinators for communication and coordination.
- Vendor should utilize common vulnerability tracking and aggregation capabilities such as the NIST National Vulnerability Database (NVD)[5], Common Vulnerabilities and Exposures (CVE)[6], and the FIRST Vulnerability Database Catalog.[7]

Variant 4: Vendor makes the vulnerability details public prior to remediation

Description

Multi-party vulnerability disclosure often involves complex interaction among stakeholders. It is possible for a vendor to disclose the vulnerability details publicly prior to remediation. In many cases, such disclosure is accidental and a plan for damage control should be in place. A review of the incident afterwards should take place to prevent occurrences in the future.

Causes

- Vendor accidentally discloses.
- Vendor has gaps, or lack of policy and controls to handle and protect sensitive vulnerability-related information.

---

[5] https://nvd.nist.gov
[6] http://cve.mitre.org
[7] https://www.first.org/global/sigs/vrdx/vdb-catalog

Prevention

- Sharing communities could institute penalties for trust violations. (e.g., a sharing community member leak could lead to expulsion from that sharing community).
- Vendor should demonstrate the use of implemented policies and controls to correctly manage and limit access to sensitive vulnerability information (i.e., compliance with ISO/IEC 27001).
- Vendor should implement measure to secure communication channels such as implementing encryption of communication with external stakeholders.

Response

- Vendor should review the incident to understand the causes and reduce future occurrences.
- Vendor should implement and demonstrate new policies and controls for handling sensitive information.
- Vendor should implement sufficient auditing and logging of vulnerability information to enable quick and clear identification of the root causes of the leak.
- Vendor should understand why and where the vulnerability been leaked while attempting to prevent further damage.
- Vendor should analyze the situation and establish a priority remediation timeline.
- For transparency and damage control, the vendor should publish a statement to the public and to affected customers.

Variant 5: Vendor does not remediate a reported vulnerability

Description

There may be situations in which the vendor does not provide remediation to a vulnerability. There are many causes for such a scenario including the vendor no longer existing, the affected product no longer being supported, the vendor being unable to verify the finder's report, or the vendor not considering the report to be a vulnerability. Establishing clear communication and dialogue between the reporter and vendor is foundational to establishing a plan of action, whether that be remediation or mitigation.

Causes

- Finder and vendor fail to set clear expectations for remediation and disclosure.
- Vendor no longer exists.
- Vendor chooses not to fix. There could be several reasons for the vendor not fixing and identifying a vulnerability including:
    - Vendor no longer supports the product.
    - There are compatibility issues impacting the fix.
    - Vendor does not have the resources to fix the vulnerability.
    - Vulnerability remediation is prohibitively expensive.
    - The vulnerability is a low priority for the vendor.
- Vendor is unable to verify vulnerability.
- Vendor does not consider the report to be a vulnerability.

Prevention

- Vendor should clearly document product support timelines and limitations including end-of-life, end-of-support, and end-of-security-support dates.
- Finder should provide clear documentation and artifacts to support vulnerability verification.
- Both parties (vendor and finder), should clearly communicate and negotiate expectations and timelines, and acknowledge receipt of each communication.

Response

- Vendor could provide alternative list of supported products with similar functionality as affected end-of-life/end-of-security related products.
- Vendor should consult with legal resources to address potential liability and indemnity issues.
- Vendor should publish a statement explaining why no fix or remediation has occurred.

Variant 6: Missing communication between peer vendors impedes coordination

Description

Missing or poor communication between peer vendors can negatively impact coordination efforts. In some cases, this is due to lack of awareness of the uses and impacts of a common component or technology that make it difficult to identify and coordinate with affected peers. Use of third party coordinators and the investment in developing and maintaining an awareness of peer vendors are just two ways of managing these complexities in multi-party coordinated response.

Example 1. A vulnerability named "httpoxy" affected many CGI or CGI-like environments.

According to httpoxy.org, it was first discovered in 2001. Over the years the issue was rediscovered many times. Its impact on other peer CGI implementations was never investigated. In 2016 when an exploit was discovered in the wild, the issue was widely investigated across various CGI implementations and 14 CVE IDs were assigned.

Example 2. CVE-2008-1447

CVE-2008-1447 is a vulnerability in DNS protocol that was first mitigated by UDP source port randomization idea implemented in djbdns in 1999. While importance of this mitigation was emphasized on public mailing lists, many other DNS implementations lacked this mitigation until 2008. When a practical exploit for this vulnerability was demonstrated in 2008, the source port randomization mitigation was widely implemented.

Causes

- Vendor may not be aware that peers use the same component or technology, or may not be aware of all potentially affected peers.
- Vendor may find it difficult to identify or coordinate with affected peers.
- Vendor may intentionally withhold information for perceived competitive advantage.

- Vendor may fail to recognize an issue as a vulnerability (e.g., lack of CVE ID).

Prevention

- Vendor should develop and maintain awareness of peers (e.g., utilize FIRST directory to identify peers).
- Vendor should develop and maintain awareness of coordinators.
- Vendor should cooperate with peers on security measures to protect common customers.
- Vendor should recognize vulnerabilities and publish accordingly (e.g., assign CVE ID).

Responses

- Vendor can engage a coordinator.
- Vendor can publish vulnerability information optionally, including proof-of-concept tests (to the public or only to peers).

Variant 7: Coordinator makes vulnerability details public prior to remediation

Description

In this variant, a coordinator discloses vulnerability information publicly before remediation is ready. As in previous variants, disclosure may be accidental, or a coordinator may intentionally disclose due to the perceived defensive benefit. Also, similar to other variants setting and expectation, good communication can reduce accidental disclosures.

Causes

- Coordinator accidentally discloses.
- Confusion due to multiple coordinators working on the same or similar issues.
- The coordinator embargo period expires or coordinator determines vendor is not responsive.
- There is an active exploitation of the vulnerability and coordinator chooses to disclose.

Prevention

- To reduce confusion when multiple coordinators are involved, coordinators should select one coordinator as lead.
- Coordinator should develop and maintain awareness of and relationships with other coordinators.
- Coordinator should publish disclosure policy and expectations including timelines and expectations for vendor responsiveness.
- Coordinator and vendor should clearly determine disclosure timeline early in process.
- Vendor can choose not to engage with coordinators with a history of uncoordinated disclosure.
- Vendor should negotiate and try to meet timelines, and be responsive.

Responses

- Vendor can increase priority of response process.
- Vendor can release interim advisory.

Use Case 3: Public disclosure of limited vulnerability information prior to remediation

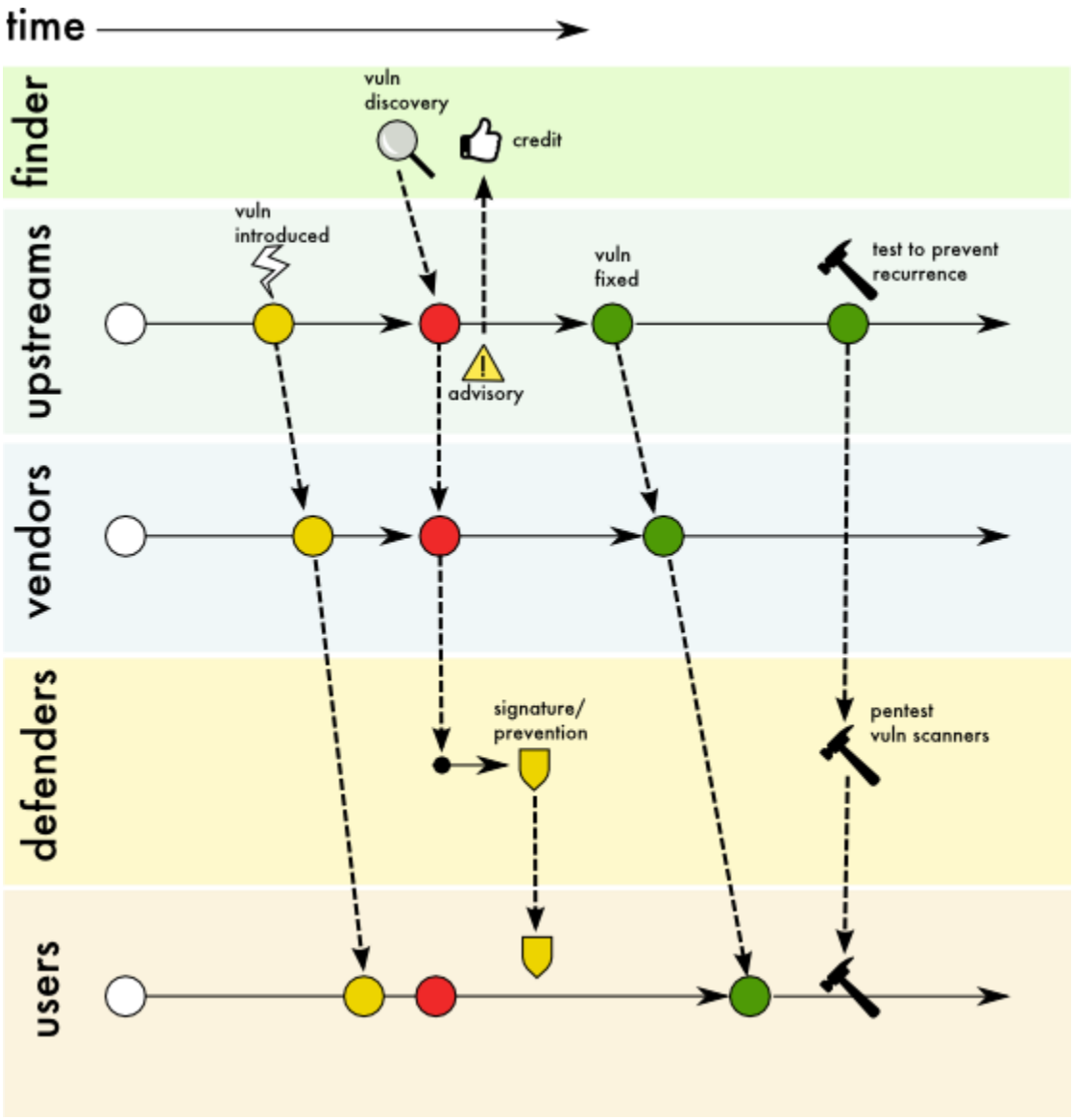

Figure 5: Use Case 3 Public disclosure of vulnerability and impact prior to remediation

Description

Some information about the vulnerability is published without giving any hints about the exploit. This use case is different than what is typically called "full-disclosure."

As a middle way between full public disclosure and a privately coordinated disclosure, a finder or a vendor may publish some preliminary notice about the existence of a vulnerability and its

disclosure timeline. Information disclosed may contain names of vulnerable product or component, worst case impact, and location of future advisories, but not provide any hints about exploiting the vulnerability such as source code changes or vulnerability type. This disclosure scenario is common when a large number of vendors are affected and maintaining confidentiality can be difficult.

Such advance notice helps all the responding parties (i.e., upstream vendors, downstream vendors, users and defenders) to plan and prepare to respond to the disclosure. Preparation may involve identifying potentially affected products and assets, identifying personnel responsible for analyzing the security fixes, making code changes or patching, testing, and solution delivery.

The variations, including causes, preventions, and responses from Use Case 2 also apply to Use Case 3.

Example 1. Vendor advance warning:

On April 28, 2016, OpenSSL project team announced a new software release with fixes for several 'high' severity security defects that was made available on May 3rd, 2016. The users and downstream vendors had five days to plan and prepare for taking response measures, thus minimizing the preparation time required for the responders.

Example 2. Vendor expected cadence:

Oracle published Critical Patch Update Advisories on a pre-determined quarterly schedule. According to Oracle[8], a pre-release announcement is also published five days prior to each Critical Patch Update release with a summary of affected products and risks. This notification serves as a trigger to initiate a customer's patching procedure.

Example 3. Researcher advance warning:

On 22nd March 2015, Stefan Metzmacher published an advance warning on website badlock.org, that a crucial security bug in Windows and Samba would be disclosed on April 12th, 2016. System administrators responsible for Windows or Samba server infrastructure were advised to be ready to patch their systems.

[8] http://www.oracle.com/us/support/assurance/leveraging-cpu-wp-164638.pdf

Use Case 4: Public disclosure or exploitation of vulnerability prior to vendor awareness

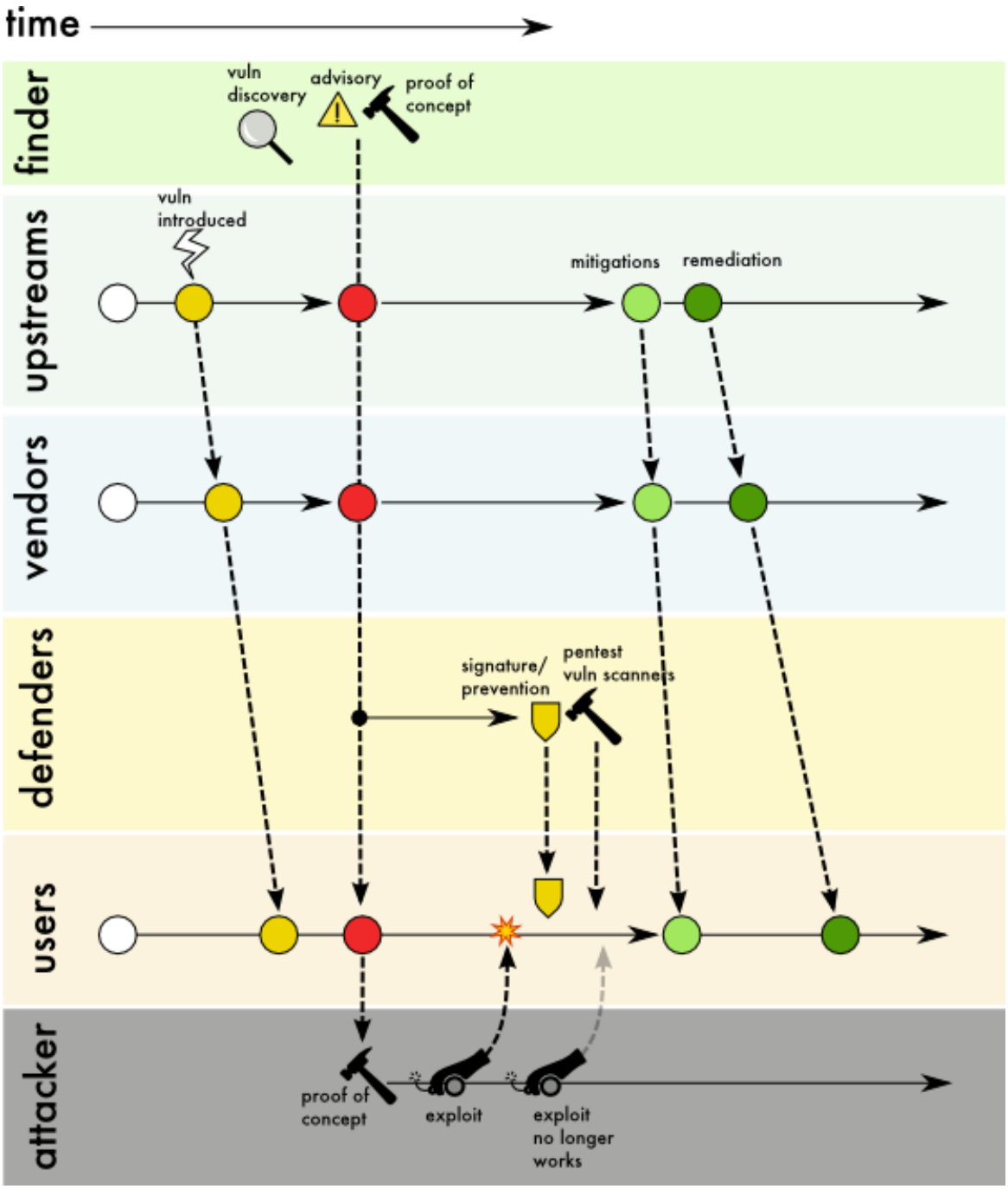

Figure 6: Use Case 4 Public disclosure or exploitation of vulnerability prior to vendor awareness

Description

When a vulnerability is discovered in a deployed product, the finder makes the information about the vulnerability accessible to anyone by methods such as publishing on the Internet, mailing lists, academic papers, or conferences. Disclosed information may include affected products and versions, proof of concept test cases that can trigger or demonstrate the vulnerability and detailed explanation of the defect or attack methodology. This disclosure is made without waiting for development or deployment of a remediation or mitigation. This type of disclosure is often referred to as "full disclosure"[9] or a "zero-day."

One of the main intentions here is to make users aware of the vulnerability as early as possible as a way to minimize exposure, with an assumption that there could be unknown attackers who may already know about the vulnerability and could be exploiting it.

An Internet survey of about 400 researchers, indicates that only 4% of the researchers follow full public disclosure versus 92% of researchers that follow some form of coordinated disclosure. While such disclosures are rare, vulnerability responders (vendors, defenders, users) should be prepared to handle disclosures anytime.

Example 1. A paper[10] presented at AppSec California in January 2015, described remote code execution under certain context related to Apache Commons Collection. Apache Commons project was not informed[11]. On November 2015, a blog post[12] was published containing exploits based on this paper for multiple products. None of the vendors or open source projects were directly notified prior to disclosure.

Variant 1: Finder publishes vulnerability details and vulnerability is exploited

Description

In this variant, a finder publicly discloses detailed vulnerability information without first having notified the vendor. Attackers can use this information to develop exploits and attack systems before vendors have prepared a remediation. Typically, attackers can develop attacks faster than vendors can develop a remediation and users can deploy them. This variant is commonly called a "zero-day" disclosure.

Causes

- The vulnerability report contains a proof of concept test or enough information to create a working exploit for the issue.
- Finder identifies previously unknown exploitation in the wild and publishes.

---

[9] Strictly speaking, "full disclosure" means publication of vulnerability details before remediation is available, either before or after notifying vendors.
[10] http://frohoff.github.io/appseccali-marshalling-pickles/
[11] https://commons.apache.org/proper/commons-collections/security-reports.html#Apache_Commons_Collections_Security_Vulnerabilities
[12] https://foxglovesecurity.com/2015/11/06/

Prevention

- The finder can withhold or delay proof of concept tests from the disclosure. Attackers would have to spend more time and effort to independently develop exploits, providing users some grace time to protect themselves.
- Addition of traceability information where possible in vendor disclosure advisory can be a deterrent to attackers.
- Vendors should monitor for public disclosures/discussions.

Response

- Vendor can provide a security advisory regarding mitigation and response.
- Vendor can accelerate patch testing and release.
- User can apply vendor fixes when available.
- User can apply workarounds provided by the vendor.
- User can apply workarounds for prevention or defenses recommended by the internal or external security community.
- User can use the proof of concept test to check for vulnerable assets.
- User can utilize security best practices to limit potential impacts.

Variant 2: Previously undisclosed vulnerability used in attacks

Description

In this variant, a vulnerability becomes publicly known because of its use in attacks. This variant is also referred to as a "zero-day" vulnerability or exploit, since vendors and defenders have not had a warning in advance. This is usually a very harmful scenario since vendors, defenders, and users rush to respond while under attack. Exploitation of a vulnerability in an attack can be considered as a disclosure of the vulnerability or a confirmation of its existence. The attacker typically wants the vulnerability and its exploitation to remain undetected and undisclosed.

Causes

- Incentives available for non-disclosure or exploitation are greater than incentives provided for disclosure.
- The vulnerability could be in a malware or a botnet in which case a disclosure is likely to make the nefarious software more secure.
- Incomplete vendor fixes may lure attackers to find closely related vulnerabilities.

Prevention

- Vendors should generally take steps to improve software security and reduce vulnerabilities. Such activity, generally referred to as Secure Software Development Lifecycle (SSDL) or Security Development Lifecycle (SDL), is beyond the scope of this document.[13]

[13] Coordinated vulnerability disclosure is often considered part of the deployment, maintenance, or support phases of a Secure Software Development Lifecycle.

- When vulnerabilities or weaknesses are found by a product assessment, make sure all the issues found are reported to appropriate stakeholders and resolved. Attackers are likely to be using the same security assessment tools and techniques, and may have encountered the same problems.
- To protect against malicious modifications and maintain supply chain integrity, vendors should produce tamper-proof or tamper-evident products.
    - Authenticity of source code or software should be verifiable using strong cryptography (e.g., use digital signatures or HTTPS while distributing software). Downstream vendors should verify authenticity of components included in their products.
    - Where possible, products should have signed, trusted, and verified execution enabled by default.
    - Consumers should verify authenticity of products that are to be used or deployed.
- Consumer/defender should continuously verify their deployments for unauthorized changes or anomalies.
- Forensically check returned or retired products for signs of compromise.

Response

- Vendor and defender should analyze exploits to determine the vulnerability.
- Where appropriate, vendors should consider providing a security advisory containing:
    - Acknowledgement of the problem
    - Development status of the remediation
    - Possible mitigations and workarounds
- Vendor can accelerate patch testing and release.
- Users can apply vendor fixes when available.
- Users can apply workarounds provided by the vendor.
- Users can apply workarounds for prevention or defenses recommended by the internal or external security community.
- Users can utilize security best practices to limit potential impacts.
- When prioritizing vulnerabilities or weaknesses found by any assessment (internal or by customers), vendors should consider that attackers can find the same or similar vulnerabilities.
- If defenders find incident indicators, then those should be reported to appropriate vendors or stakeholders for investigation.

Guiding Concepts and Best Current Practices

The following guidance is derived from the cases, variants, responses, and preventions discussed previously. The most important practices, particularly those that occurred repeatedly, are captured here. Stakeholders should carefully consider their actions, particularly notification and public disclosure, due to the widespread impact on other stakeholders in multi-party cases.

Establish a strong foundation of processes and relationships

- Establish and publish actionable public vulnerability coordination and disclosure policies and expectations, including timelines and thresholds for disclosure (e.g. severity).
- Develop and maintain awareness of peers and other potential stakeholder communities.
- Vendor should pre-establish upstream and downstream vendor relationships and communication channels to understand potential impacts and coordination timelines.
- Vendor should consider tracking the use of third party components to better develop inventory and an understanding of upstream and downstream dependencies.

Maintain clear and consistent communications

Prior to disclosure

- All parties should clearly and securely communicate and negotiate expectations and timelines.
- Vendors should provide currently accepted contact mechanisms, such as security@ email addresses and "slash security" (/security) web pages.
- All parties should acknowledge receipt of each communication.
- Vendor or coordinator should maintain frequent communication with finder including status updates and potential impacts to disclosure timeline.
- Finder should provide clear documentation and artifacts to support vulnerability verification.
- Vendor should clearly document product support timelines and limitations.
- All parties should avoid individual points of failure for communication.

After disclosure

- Vendor should provide clear advisories and bulletins in machine-readable format related to vulnerability fixes and mitigations (e.g., CVRF).
- Vendor should identify a dedicated contact for upstream and downstream stakeholders, in addition to communicating via generic e-mail, like secure@example.com.
- If needed, vendor should leverage coordinators for broad communication and coordination.
- All parties should utilize common vulnerability tracking and aggregation capabilities like the NIST National Vulnerability Database (NVD) and Common Vulnerabilities and Exposures (CVE).
- All parties should adopt a vulnerability scoring system standardization mechanism (e.g., CVSS) to raise awareness for users on the severity of the vulnerability.

Build and maintain trust

- All parties should implement measures to secure communication and handling of sensitive information. (e.g., implementing encryption of communication with external stakeholders).
- Vendor should test updates rigorously prior to security fix release.
- Vendor can establish bug bounty programs, credit or safe harbor, to proactively identify vulnerabilities prior to release.
- All parties should avoid escalation to any extent possible (including legal action). Stakeholders should encourage security research and coordinated disclosure within relevant legal frameworks. Legal or other coercive pressure, actual or perceived, often creates a chilling effect on desired security research.

Minimize exposure for stakeholders

- Vendor can release fixes on a predetermined schedule (e.g., Patch Tuesday).
- When possible, vendors should not include non-security updates with security fixes (e.g., JRE model).
- Vendor should offer an automatic update process for users if possible.
- User should enable automatic vendor patch updates if available.
- Vendor should establish and participate in upstream downstream trusted networks (e.g., vetted mailing lists such as the UEFI USRT for rapid communication and coordination).
- Vendor can provide any available mitigations or workarounds even if they may cause some degradation of service.
- Stakeholders should consider partial, preliminary public disclosure as described in Use Case 3.
- Downstream vendors should consider keeping their components up-to-date as soon as upstream vendors recommend a release.

Respond quickly to early disclosure

- Vendor should analyze the situation and establish a priority remediation timeline.
- Where possible, vendor can reach out to finder to define the scope of early disclosure and perform damage control.
- Vendor should provide communications to users regarding the vulnerability and potential mitigations (e.g., release an interim advisory).

Use coordinators when appropriate

- Coordinator can help connect researchers, vendors, and other stakeholders. This is particularly helpful when multiple parties (vendors) are involved or there is difficultly contacting a party (vendor).
- Coordinator can provide additional technical, impact, and scope analysis to researchers, vendors, and other stakeholders, particularly when there is disagreement.
- Coordinator should develop and maintain awareness of and relationships with other coordinators.

- To reduce confusion when multiple coordinators are involved, one coordinator should be selected as lead.

Acknowledgements

The Vulnerability Coordination SIG thanks all of its members and specifically the following contributors:

Pete Allor, Honeywell

Christa Anderson, Microsoft

Jerry Bryant, Microsoft

Vic Chung, SAP

Mark Cox, Red Hat

Beverly Finch, Lenovo

Jeroen van der Ham, NCSC-NL

Kent Landfield, McAfee

Magid Latif, Intel

Art Manion, CERT/CC

Klee Michaelis, Cisco

Bruce Monroe, Intel

Chandan Nandakumaraiah, Juniper

Kymberlee Price, Bugcrowd

Krassimir Tzvetanov, Fastly

Tania Ward, Dell EMC

Brian Willis, Intel

The SIG also thanks Allan Friedman of the National Telecommunications and Information Administration (NTIA) at the U.S. Department of Commerce.

Supporting Resources

ENISA Good Practice Guide on Vulnerability Disclosure (2015)
https://www.enisa.europa.eu/activities/cert/support/vulnerability-disclosure

NIAC Guide to Vulnerability Disclosure (2004)
https://www.dhs.gov/xlibrary/assets/vdwgreport.pdf

ISO/IEC 29147 Vulnerability Disclosure (2014)
http://www.iso.org/iso/catalogue_detail.htm?csnumber=45170
http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Vulnerability disclosure publications and discussion tracking
https://www.ee.oulu.fi/research/ouspg/Disclosure_tracking

Responsible Disclosure Guideline
https://www.ncsc.nl/english/current-topics/news/responsible-disclosure-guideline.html