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

Guidelines and Practices for Multi-Party Vulnerability Coordination

Draft for public comment

Table of Contents

Introduction

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

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

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

59 This document differs from the ISO Vulnerability disclosure and handling standards (ISO/IEC
60 29147 and ISO/IEC 30111) in that the ISO standards provide basic guidance on the handling of
61 potential vulnerabilities in products. This document is a collection of best current practices that
62 consider more complex and typical real-life scenarios that extend past a single researcher notifying
63 a single company about a discovered vulnerability.

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

69

---

[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

102   **Vulnerability:** Weakness in software, hardware, or a service that can be exploited.
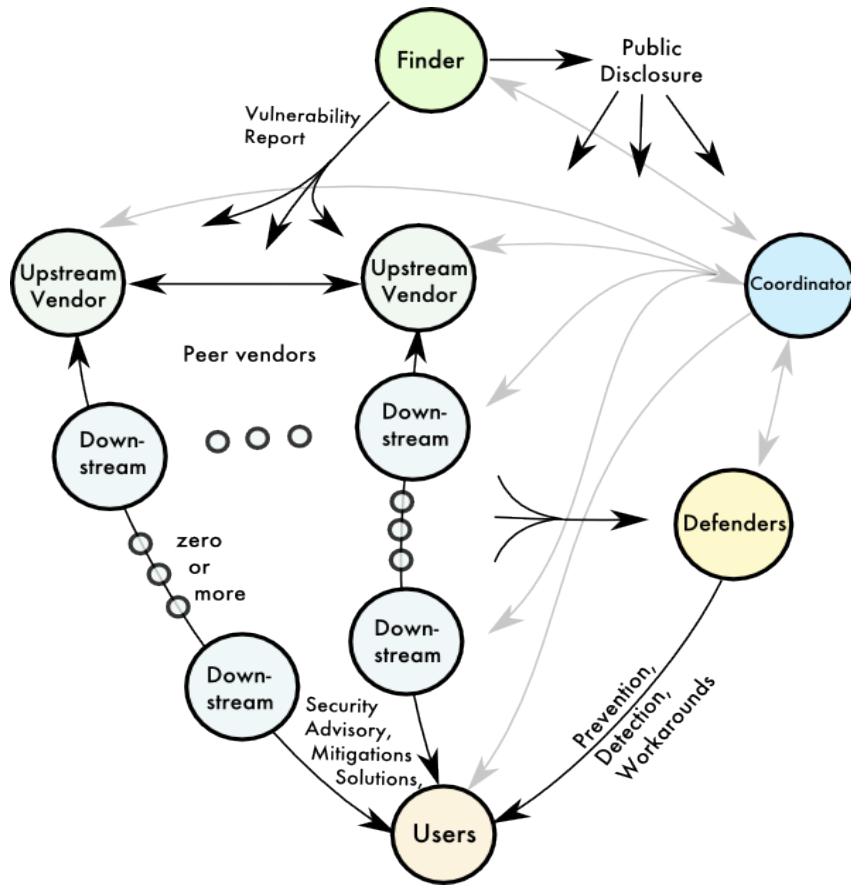
Figure 1: Stakeholder roles and communication paths

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

106

Multi-Party Disclosure Use Cases

108 Vulnerability disclosure can be a complicated process, especially when multiple parties (usually
109 multiple vendors) are involved. This section of the document is organized as a set of vulnerability
110 disclosure use cases, in rough order, from simple to complex. Significant attention is given to
111 coordinated, Multi-Party Disclosure (see Use Case 2: Vulnerability with Coordinated Disclosure).
112 Disclosure often deviates from the expected or ideal process, so within each use case are variants
113 that are common exceptions to the ideal use case. Within each variant are causes, preventions, and
114 responses. The collected set of preventions and responses are presented as practices that can be
115 used to reduce the occurrence and cost of expected variants.

116 Practices are denoted as strong recommendations ("should") or suggestions ("can," "could," or
117 "may").

118 At the conclusion of the use cases and variants, practices are rolled-up into the concluding section:
119 Guiding Concepts and Best Current Practices.

Use Case 0: No vulnerability

121 **Description**
122 This case is included for completeness, if there are no vulnerabilities, there is no need for
123 coordination.

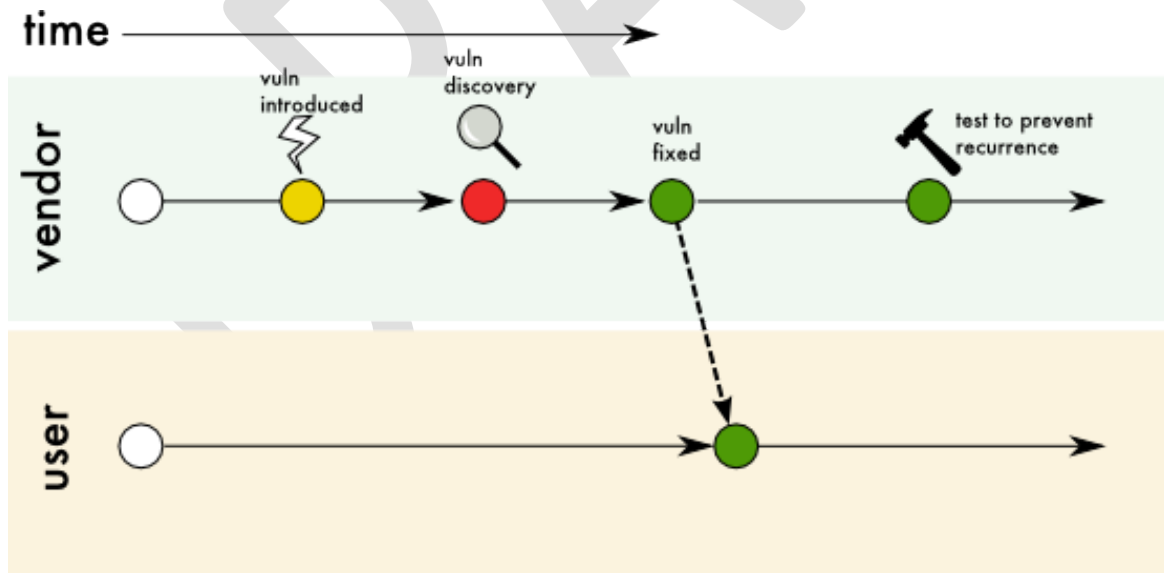### Use Case 1: Vulnerability with no affected users

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

126

127 **Description**
128 A vulnerability software or hardware with no users is a security hole that does not affect anyone
129 else in any way. Examples of this case include: products that are (a) non-production, experimental

130 (e.g., webgoat), (b) internal or for personal use, (c) never published or sold, or (d) under
131 development.

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

134 Coordination is not required, except:

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

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

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

145 *Causes*
146 • The affected product is not well tested.
147 • The affected product is deployed too soon.
148 • The affected product is deployed with known vulnerabilities.

149 *Prevention*
150 • Perform product penetration testing and or/scanning for known vulnerabilities prior to
151   release.
152 • Establish bug bounty programs to proactively identify vulnerabilities prior to release.
153 • Set clear expectations and baselines on beta quality versus ready for release requirements.

154   Use Case 2: Vulnerability with coordinated disclosure
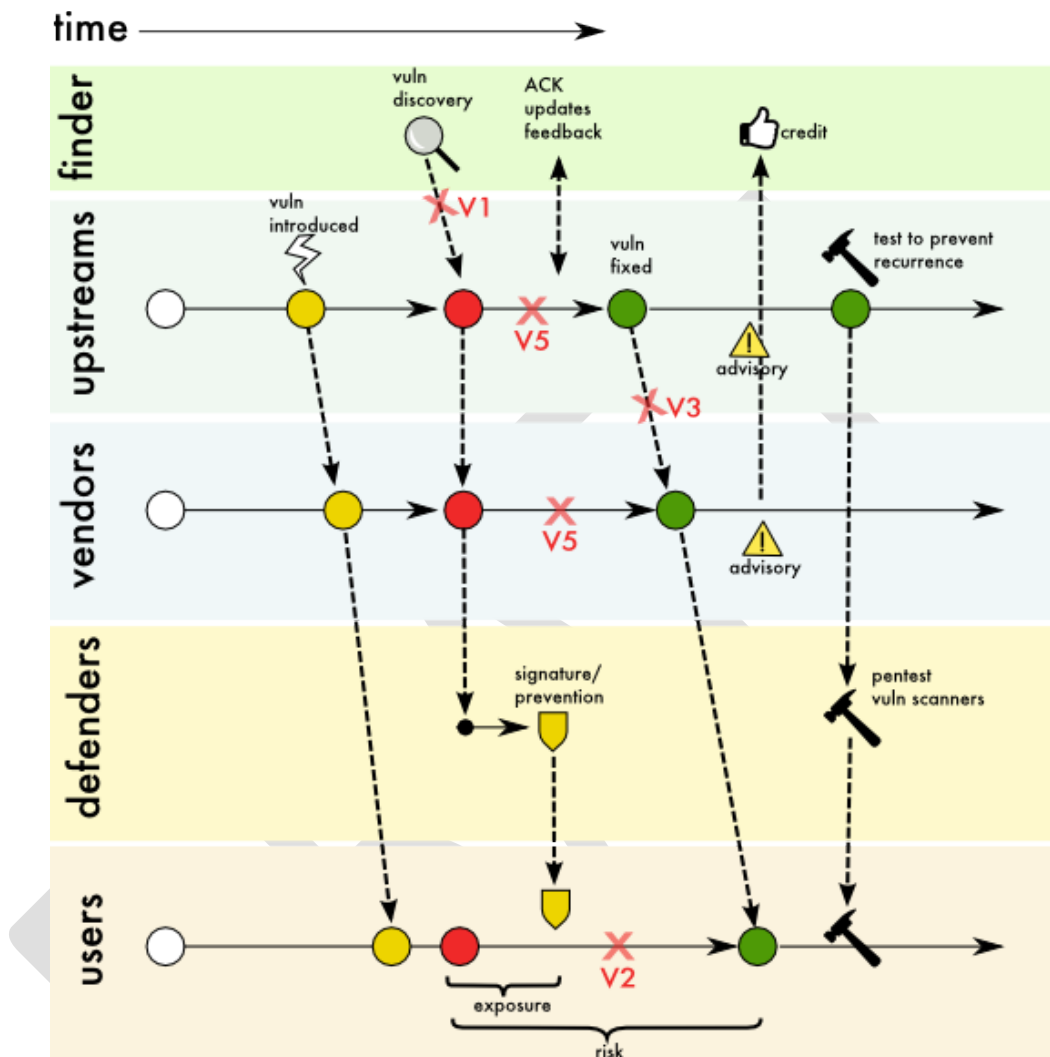
Figure 3: Vulnerability with coordinated disclosure

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

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

161   Finder

162   • Finder contacts the vendor using standard vulnerability reporting channels.

163 <u>Vendors</u>

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

167 <u>Defenders</u>

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

172 Users

173 • Deploy vendor patch / mitigation as soon as possible.

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

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

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

Prevention

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

214 *Response*

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

222 **Variant 2: Users do not deploy remediation immediately**

223 *Description*

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

232 *Causes*

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

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

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

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

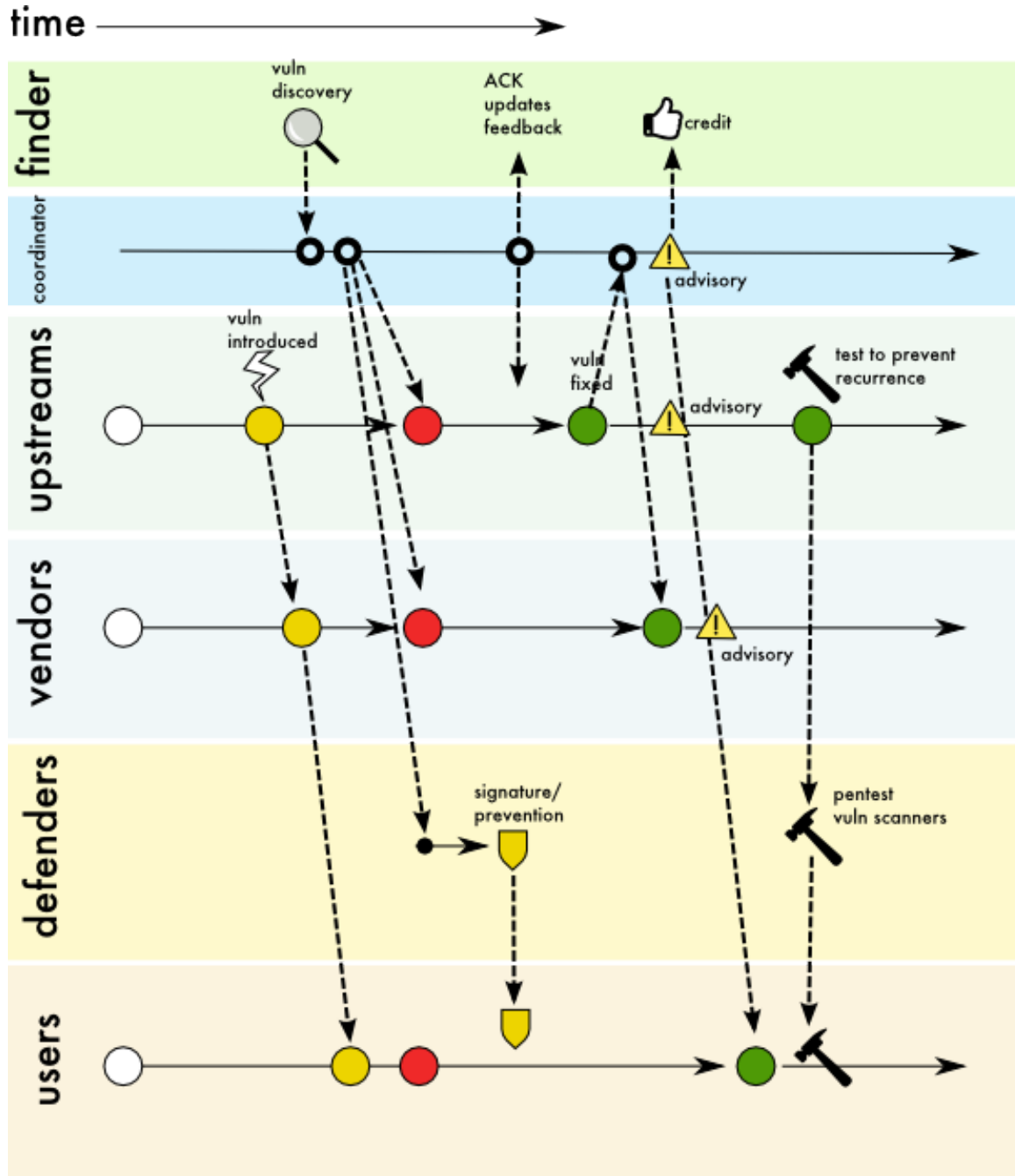274   **Variant 3: Missing communication between upstream and downstream vendors**

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

275

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

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

291 *Prevention*
292 • Vendor to establish an actionable public vulnerability coordination and disclosure policy,
293   ideally describing the threshold for disclosure (e.g. severity).
294 • Vendor should consider communicating remediation/mitigations of all vulnerabilities
295   regardless of severity rating or source of vulnerability report.
296 • Downstream vendors should consider keeping their components in-sync with upstream
297   recommended release. Selectively patching security vulnerabilities can become tedious,
298   error prone and expensive in the long run as source code can diverge between upstream
299   and downstream instances. Downstream vendors may also miss security improvements or
300   vulnerability fixes that do not get CVE assignments or get CVE assignments at a later date
301   (e.g., CVE-2016-2108[4]).
302 • Vendor should consider tracking the use of third party components to develop better
303   inventory and understanding of upstream and downstream dependencies.
304 • Vendor should pre-establish an upstream downstream trusted network for rapid
305   communication and coordination (e.g., mailing lists such as the UEFI USRT).
306 • Vendor should clearly communicate disclosure timelines to downstream vendors.
307 • Vendor should anticipate the timeframes needed for downstream coordination.
308 • Vendor could leverage coordinators for communication and coordination in the following
309   ways:

---

[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.

310   o A coordinator may receive a vulnerability report from a finder that affects multiple
311   vendors and then distribute that report to affected upstream and downstream
312   vendors.
313   o A coordinator may receive a vulnerability report and resolution information from a
314   vendor and help identify other affected vendors, possibly peer vendors and relay the
315   information to them.
316   o A coordinator may refer to the vendor directory to determine affected vendors.
317   o A coordinator may also inform defenders at appropriate times to help mitigate or
318   prevent attacks.
319   o A coordinator may publish a public advisory in addition to vendor advisories to
320   create awareness about the vulnerability and available remediation.

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

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

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

338 *Causes*
339 • Vendor accidentally discloses.
340 • Vendor has gaps, or lack of policy and controls to handle and protect sensitive vulnerability-
341   related information.

342 *Prevention*
343 • Sharing communities could institute penalties for trust violations. (e.g., a sharing
344   community member leak could lead to expulsion from that sharing community).

---

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

14

345   •   Vendor should demonstrate the use of implemented policies and controls to correctly
346   manage and limit access to sensitive vulnerability information (i.e., compliance with
347   ISO/IEC 27001).
348   •   Vendor should implement measure to secure communication channels such as
349   implementing encryption of communication with external stakeholders.

350 *Response*
351   •   Vendor should review the incident to understand the causes and reduce future occurrences.
352   •   Vendor should implement and demonstrate new policies and controls for handling sensitive
353   information.
354   •   Vendor should implement sufficient auditing and logging of vulnerability information to
355   enable quick and clear identification of the root causes of the leak.
356   •   Vendor should understand why and where the vulnerability been leaked while attempting
357   to prevent further damage.
358   •   Vendor should analyze the situation and establish a priority remediation timeline.
359   •   For transparency and damage control, the vendor should publish a statement to the public
360   and to affected customers.

361 **Variant 5: Vendor does not remediate a reported vulnerability**

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

369 *Causes*
370   •   Finder and vendor fail to set clear expectations for remediation and disclosure.
371   •   Vendor no longer exists.
372   •   Vendor chooses not to fix. There could be several reasons for the vendor not fixing and
373   identifying a vulnerability including:
374   o   Vendor no longer supports the product.
375   o   There are compatibility issues impacting the fix.
376   o   Vendor does not have the resources to fix the vulnerability.
377   o   Vulnerability remediation is prohibitively expensive.
378   o   The vulnerability is a low priority for the vendor.
379   •   Vendor is unable to verify vulnerability.
380   •   Vendor does not consider the report to be a vulnerability.

381 *Prevention*
382   •   Vendor should clearly document product support timelines and limitations including end-
383   of-life, end-of-support, and end-of-security-support dates.

384	• Finder should provide clear documentation and artifacts to support vulnerability
385	verification.
386	• Both parties (vendor and finder), should clearly communicate and negotiate expectations
387	and timelines, and acknowledge receipt of each communication.

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

394 **Variant 6: Missing communication between peer vendors impedes coordination**

395 *Description*

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

401 *Example 1.* A vulnerability named 'httpoxy' affected many CGI or CGI like environments.

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

406 *Example 2.* CVE-2008-1447

407 CVE-2008-1447 is a vulnerability in DNS protocol that was first mitigated by UDP source port
408 randomization idea implemented in djbdns in 1999. While importance of this mitigation was
409 emphasized on public mailing lists, many other DNS implementations lacked this mitigation until
410 2008. When a practical exploit for this vulnerability was demonstrated in 2008, the source port
411 randomization mitigation was widely implemented.

412 *Causes*
- 413 • Vendor may not be aware that peers use the same component or technology, or may
- 414 not be aware of all potentially affected peers.
- 415 • Vendor may find it difficult to identify or coordinate with affected peers.
- 416 • Vendor may intentionally withhold information for perceived competitive
- 417 advantage.
- 418 • Vendor may fail to recognize an issue as a vulnerability (e.g., lack of CVE ID).

Prevention

420   • Vendor should develop and maintain awareness of peers (e.g., utilize FIRST
421   directory to identify peers).
422   • Vendor should develop and maintain awareness of coordinators.
423   • Vendor should cooperate with peers on security measures to protect common
424   customers.
425   • Vendor should recognize vulnerabilities and publish accordingly (e.g., assign CVE
426   ID).

427 *Responses*

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

431 **Variant 7: Coordinator makes vulnerability details public prior to remediation**

432 *Description*

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

437 *Causes*

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

444 *Prevention*

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

456 *Responses*

457   •   Vendor can increase priority of response process.

458   •   Vendor can release interim advisory.

459

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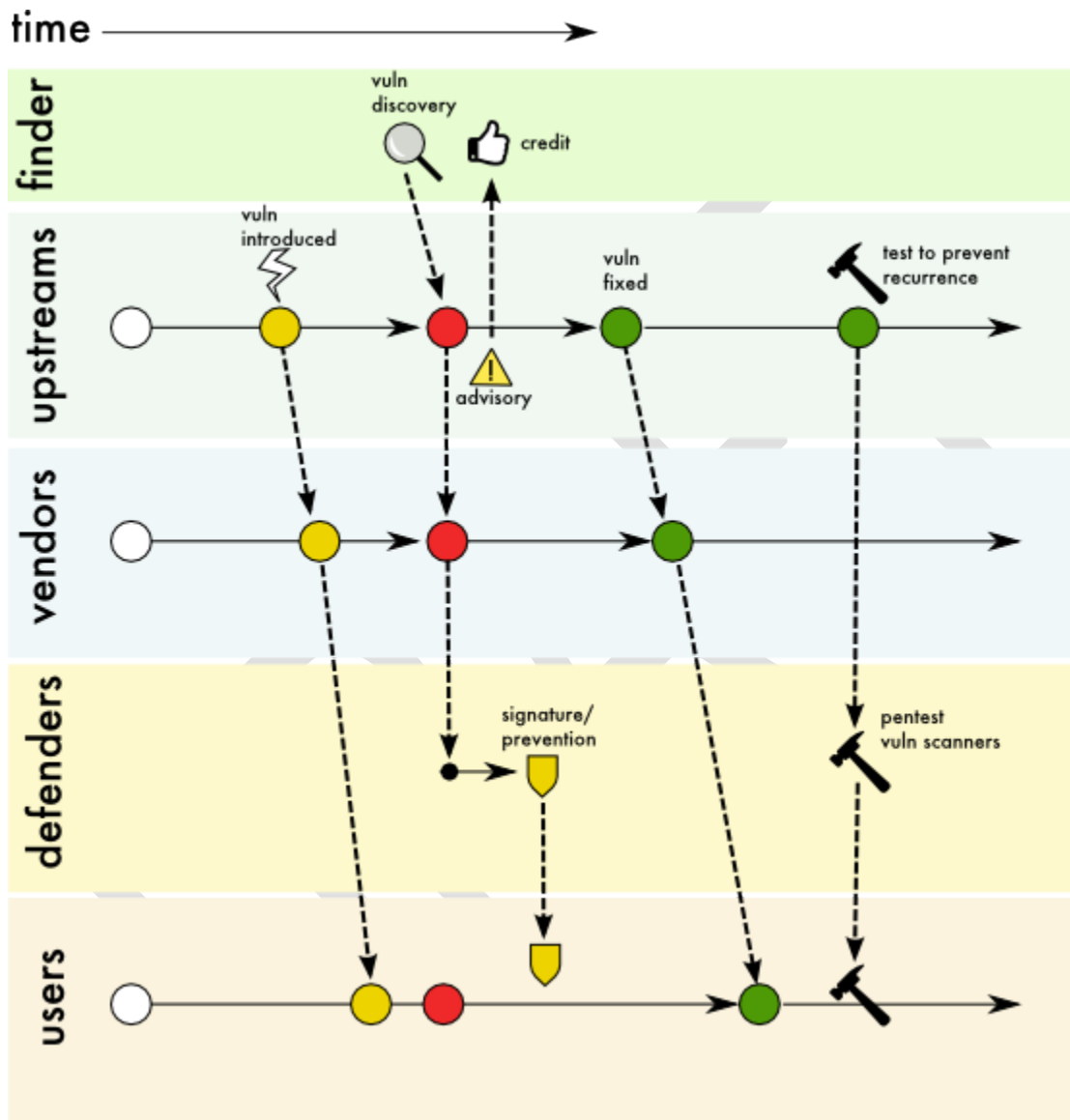461 remediation

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

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

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

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

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

475 The variations, including causes, preventions, and responses from Use Case 2 also apply to Use Case
476 3.

477 *Example 1.* Vendor advance warning:

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

482 *Example 2*. Vendor expected cadence:

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

487 *Example 3*. Researcher advance warning:

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

492

---

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

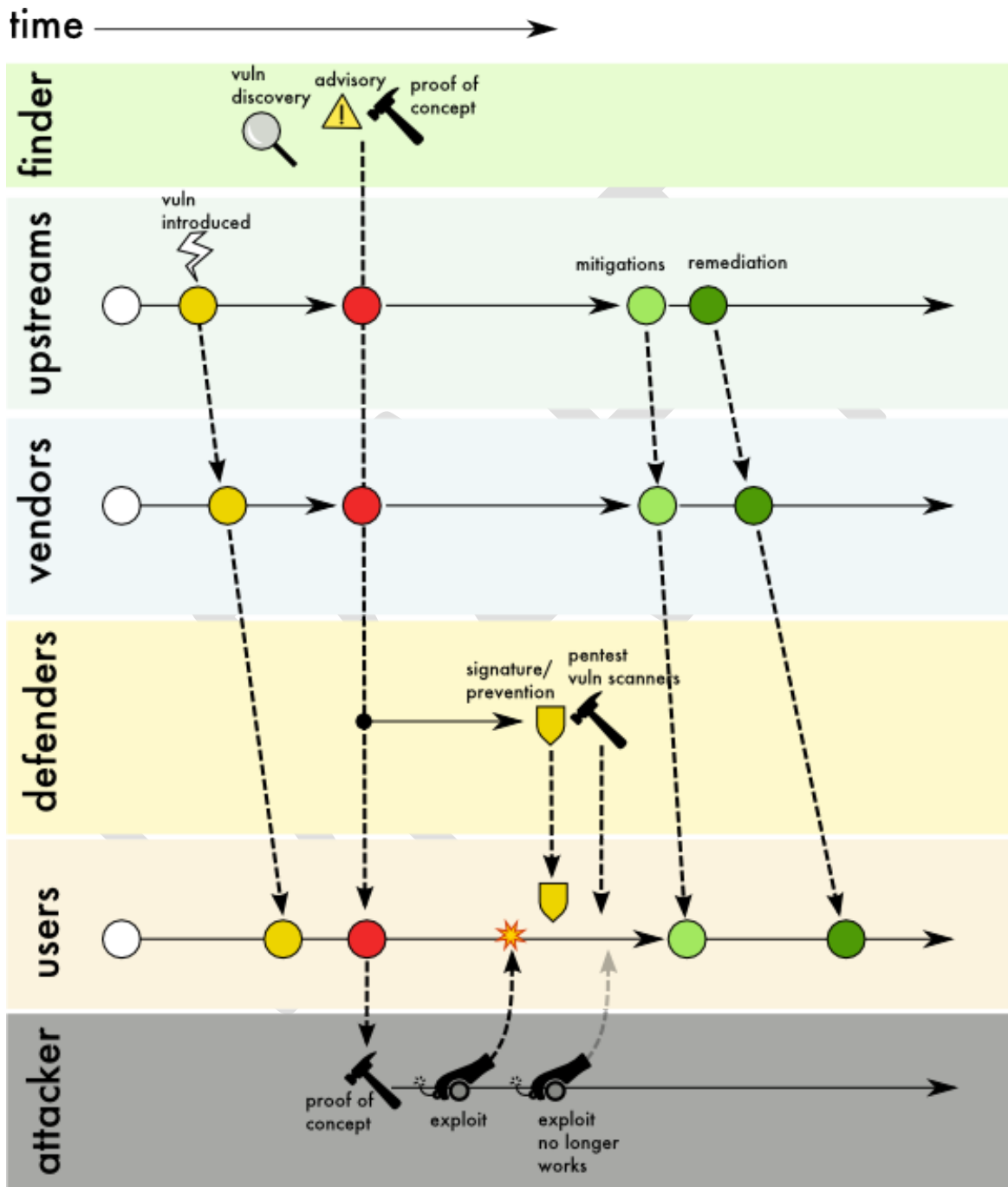Use Case 4: Public disclosure or exploitation of vulnerability prior to vendor
awareness

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

495

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

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

534 *Response*

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

543 **Variant 2: Previously undisclosed vulnerability used in attacks**

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

551 *Causes*
552 • Incentives available for non-disclosure or exploitation are greater than incentives provided
553   for disclosure.
554 • The vulnerability could be in a malware or a botnet in which case a disclosure is likely to
555   make the nefarious software more secure.
556 • Incomplete vendor fixes may lure attackers to find closely related vulnerabilities.
557

558 *Prevention*
559 • Vendors should generally take steps to improve software security and reduce
560   vulnerabilities. Such activity, generally referred to as Secure Software Development

| 561 | Lifecycle (SSDL) or Security Development Lifecycle (SDL), is beyond the scope of this |
| 562 | document.[13] |

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

579 *Response*
- 580 • Vendor and defender should analyze exploits to determine the vulnerability.
- 581 • Where appropriate, the vendor should consider providing a security advisory that can
- 582   contain:
- 583   o acknowledgement of the problem
- 584   o development status of the remediation
- 585   o possible mitigations and workarounds
- 586 • Vendor can accelerate patch testing and release.
- 587 • Users can apply vendor fixes when available.
- 588 • Users can apply workarounds provided by the vendor.
- 589 • Users can apply workarounds for prevention or defenses recommended by the internal or
- 590   external security community.
- 591 • Users can utilize security best practices to limit potential impacts.
- 592 • When prioritizing vulnerabilities or weaknesses found by any assessment (internal or by
- 593   customers), vendors should consider that attackers can find the same or similar
- 594   vulnerabilities.
- 595 • If defenders find incident indicators, then those should be reported to appropriate vendors
- 596   or stakeholders for investigation.
- 597

---

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

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.

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

675

Supporting Resources

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

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

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

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

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