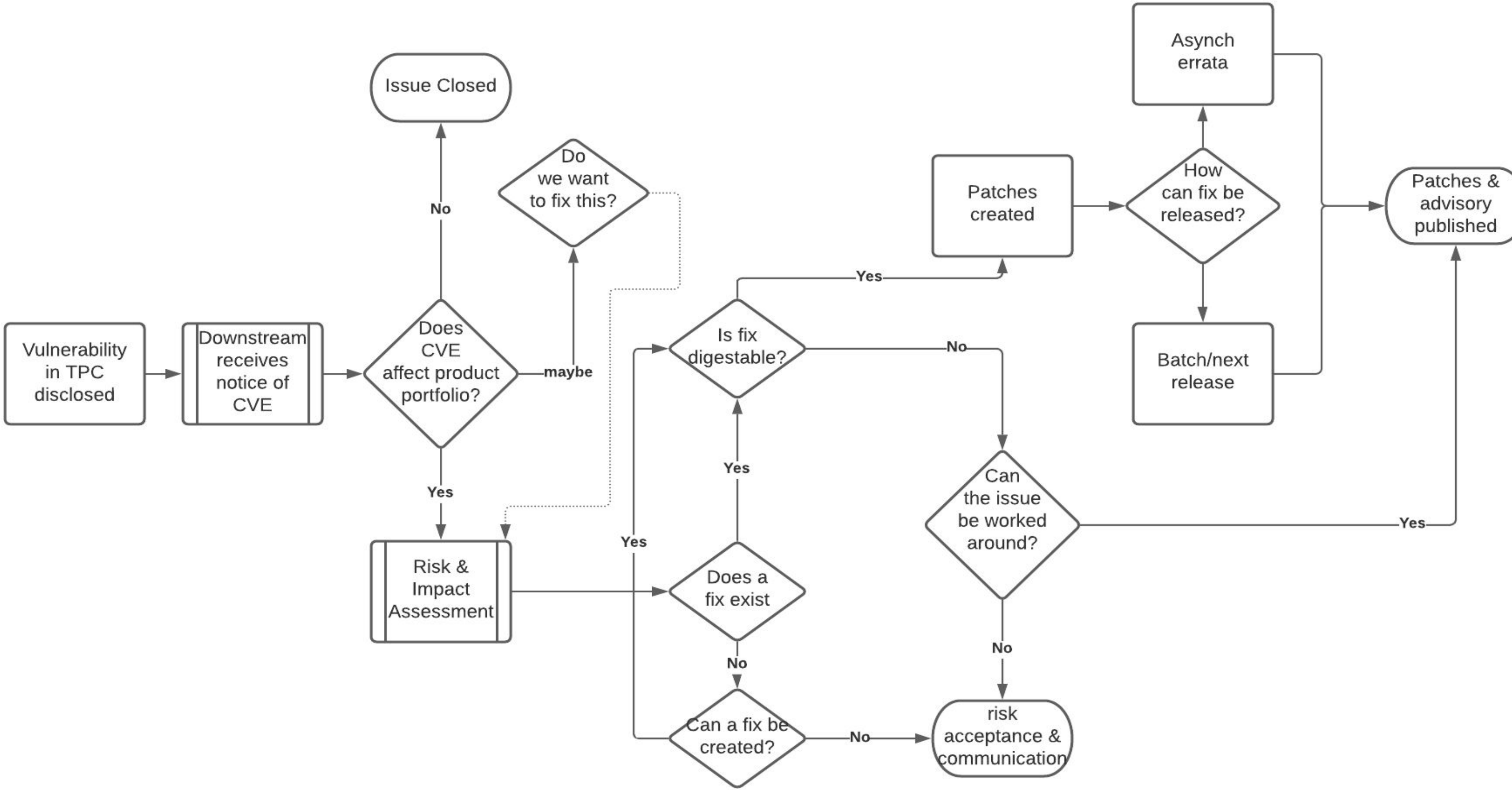
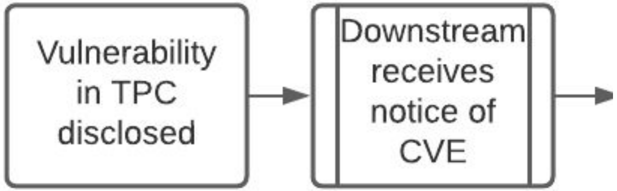
A Day in the Life of a TPC Vuln

FIRST TPC WG


Vulnerability in TPC disclosed
Downstream receives notice of CVE
Does CVE affect product portfolio?
No
Issue Closed
maybe
Do we want to fix this?
Yes
Risk & Impact Assessment
Does a fix exist
Yes
Is fix digestable?
Yes
Patches created
How can fix be released?
Asynch errata
Batch/next release
Patches & advisory published
No
Can the issue be worked around?
Yes
No
risk acceptance & communication
No
Can a fix be created?
Yes
No


Vulnerability in TPC disclosed
Downstream receives notice of CVE

How do you find out about upstream vulnerabilities? (Box 1)

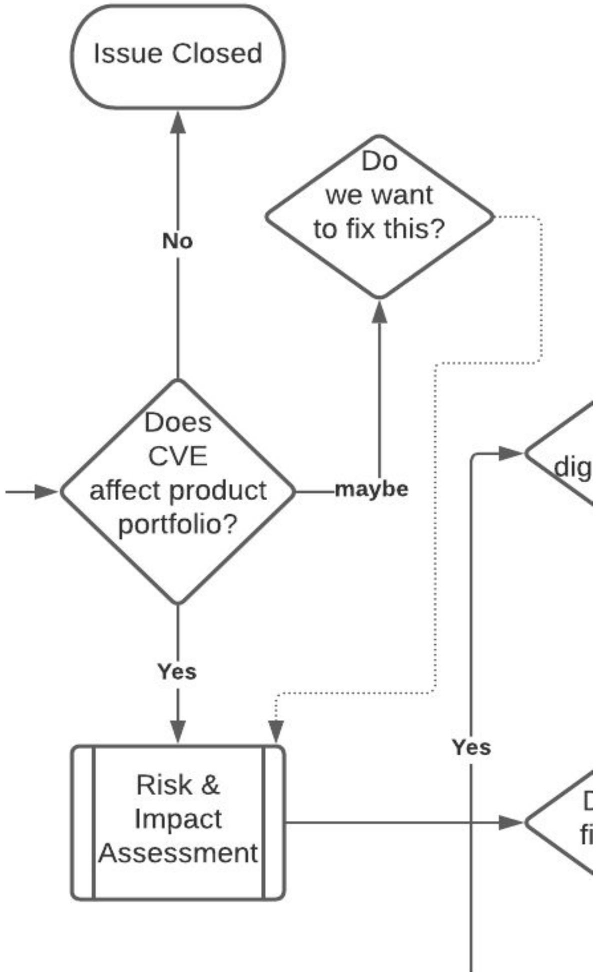
- Inventory of existing components (manifest, SBOM, etc)
    - Require products to deliver product manifest that tracks upstream sources
- NVD pull/threat intel/tooling
    - OWASP, Blackduck, Whitesource, etc
- Researcher notification
- Your customers
- External coordinators (CERT-CC, et. al.) at public disclosure
- CSIRT notice to PSIRT
- VDP/Bug Bounty program
    - BugCrowd, HackerOne
- News sites, Twitter
- Follow specific researchers relevant to your tech
- Monitor for upstream advisories
    - Commercial supplier (SUSE, Red Hat, etc)
- Track upstream repository commits
- Monitoring conference abstracts

Manifesting

- Part of sdl processes
- Internal authorized software repo
- Run scanners on dependency list to track version deltas
- Record upstream component support lifecycle
- Record upstream component community maturity
- Will vary highly by organization

You MUST know what you are consuming

Recommendations - complete component manifests & top-level dependencies - connection to SDLC

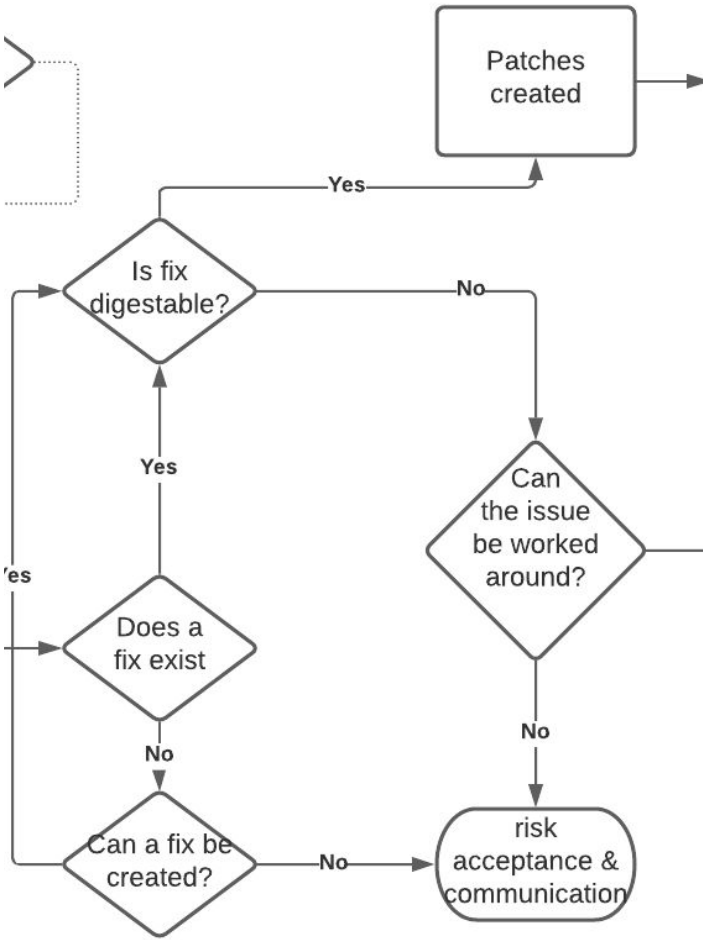


Does CVE affect product portfolio?

- Sharing manifests internally and making them discoverable will possibly help ease corp/customer concerns/visibility
    - Using tooling like blackduck to view the BOM to see where the component is deployed
- What is support state of component? Where its lifecycle is it?
- Trade-off of costs of detailed analysis vs. just fixing it
    - Visibility of the component affects decision
    - Credibility of reporting group (do they have a reputation/track record of good reports/analysis?)
    - What is End Of Life/Support obligations? What is corp policy
- CI/CD tooling helps automate a lot of this testing
    - Github dependabot
- NO-path - talk about things PSIRT could do (issue advisory/note about unaffected state) - false-positive documentation for internal use for support

Risk & Impact Assessment

- How do you KNOW you are vulnerable?
    - Reproducer
    - Bug bar
    - Scanning tools/code analysis
    - Manifest grep
    - Trusted partner/researcher collab
- Measuring Impact
    - How bad is it?
- Learn org risk appetite/tolerances/legal-regulatory obligations
- Think about how scanners reporting this will reflect upon the organization. How does that influence if this needs/does get fixed?
- Sources of risk that are not immediately apparent - what are the hidden costs of “fighting” publicly with a researcher?
- Less mature orgs will be more qualitative in their assessments. As orgs grown and mature, they can move to quantitative facts/numbers based review of actual impacts to org
- Understanding gorg’s risk tolerance/appetites. Understanding revenue/renewals impacts
- Understanding where/how your product is used (regulated industries, health & safety concerns, global privacy concerns)



Does a Fix exist?

- Who is the supplier? What/when will they react to this CVE
  - Will fix ever be created?
  - What is timeline for fix?
- Can the organization create their own fix if the supplier will not craft one?
  - What is product delivery schedule?
  - Does org have expertise to “roll their own” patch?
    - Support/legal requirements
    - Resource availability/schedule

Is fix digestible?

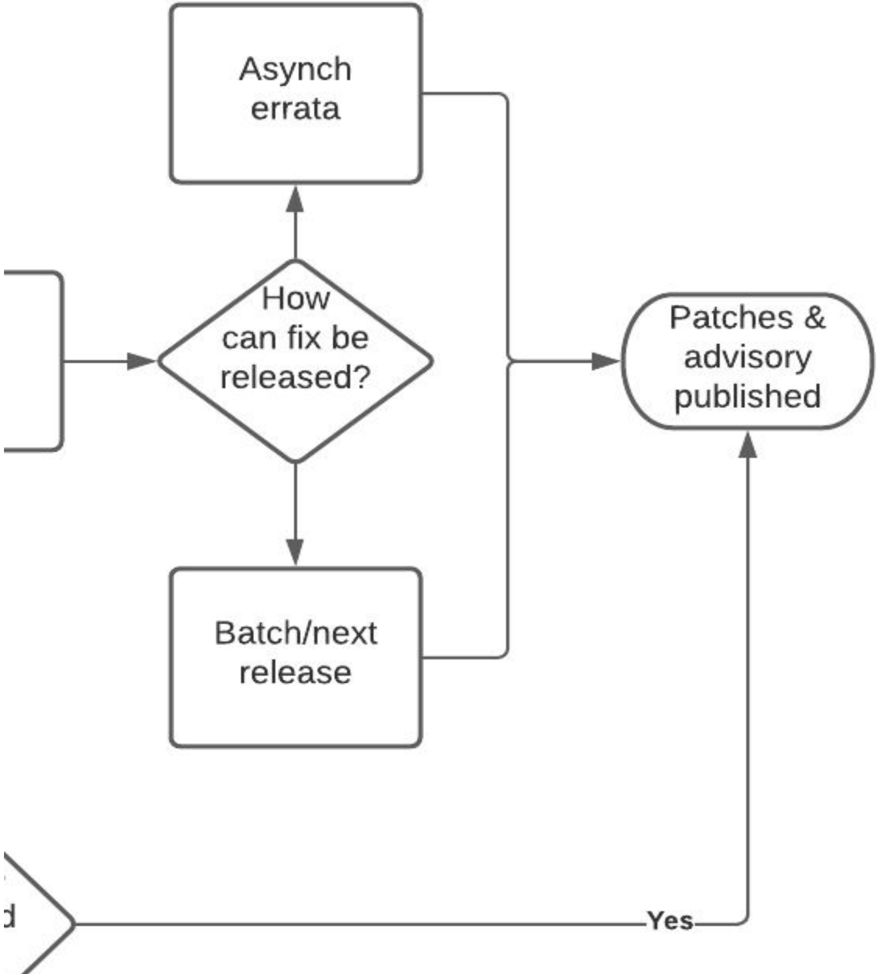
- Is fix technically compatible with offering?
- Dependencies
  - Is there a blocker with dependencies/versions?
  - Can they be resolved?

Issue Can’t be fixed - Can’t be worked around

- Deeper analysis of impacts of issue on product
  - Does this issue REALLY impact you?
- Talking about exploits/detections
- Cost/benefit/risk analysis of repercussions of not fixing issue
- What alternatives do end-users have?
- Is product EOL? How close?
- Can supplier be influenced to create fix if they initially are unwilling/unable?
  - Does an alternate supplier exist?

Can the issue be worked around or mitigated?

- Explore alternate versions as fix
- Document exploit scenario
- Can feature be disabled? Is it disabled by default? What is default deploy config
- Risk assessment on Likelihood v. Impact
- “Fix” may be a patch, alternative mitigation, or other work-around
- Would it cost more to fix the issue than the org is possibly liable for?



Patches Release/Advisory Published

- Advisory delivery timing/position may vary
- Pre-emptive notification on the issue prior to analysis
- Communication types (might not merit a formal advisory, could be a blog,a note, a bulletin, etc). What are your org priorities?