4.2 Reporting

No matter who you are, most of the smartest people work for someone else.
– Bill Joy, Sun Microsystems

Anyone who becomes aware of a vulnerability that does not appear to have been remediated should report the vulnerability to the vendor. One should not assume that a vendor is aware of a specific vulnerability unless it has already been publicly reported, whether by the vendor or elsewhere. The easier it is to report vulnerabilities to a vendor, the less likely that the vendor will be surprised by vulnerability reports disclosed directly to the public.

- Advice for Reporters
  - Finding Vendor Contacts
    - Common Methods
    - Less common, but occasionally useful
    - When all that fails
  - Providing Useful Information
- Advice for Vendors
  - Create Secure Channels for Reporting
  - Reduce Friction in the Reporting Process
- References

Advice for Reporters

Finding Vendor Contacts

Making initial contact with a vendor can sometimes be more difficult than it should be. Here is a list of techniques we've used to figure out how to reach a vendor we've never spoken with before.

Common Methods

- Search the web or the vendor's web site for relevant phrases
  - "report a vulnerability"
  - "security"
  - "report a bug"
  - "bug bounty"
  - "vulnerability disclosure policy"
  - "security@" + company name
  - company name + "PSIRT"
- See if the vendor has a `security.txt` file, often found at `www.example.com/.well_known/security.txt` or sometimes at `www.example.com/security.txt` (securitytxt.org , IETF Draft)
- Check vulnerability disclosure / bug bounty service providers (BugCrowd, Synack, HackerOne, etc.) to find vendor contacts.
- Check the Forum of Incident Response and Security Teams (FIRST) member directory at https://www.first.org/members/teams/
- Check the CVE Numbering Authority list at https://cve.mitre.org/cve/request_id.html#cna_participants
- Search open source code repositories (Github, GitLab, SourceForge, etc.) to find developer contacts.
  - If no direct contact information can be found, posting to the Issues page of a project asking how they'd like to receive vulnerability reports can be useful.
- Submit a bug report through the vendor's online bug tracker
  - If given the option to mark it as security-related, please do so as this often restricts viewing to just the vendor.
- Reach out through social media (Twitter, LinkedIn, etc.) to request the vendor establish a direct communication channel
  - We recommend you avoid posting vulnerability details in public when making initial contact when possible. For example, reporters might instead post an issue to a public bug tracker requesting that the vendor provide a secure method of communication instead of just posting the vulnerability details directly in a publicly visible issue.
- Try emailing commonly used addresses:
  - support@, security@, abuse@, info@, sales@
- Fill out a generic support or "Contact Us" form
- Make a phone call to the vendor

Less common, but occasionally useful

We have rarely had to resort to these techniques, but they have been occasionally useful.

- Send a fax (yes, we've actually done this)
- Send snail mail[1] to an executive
  - If you have access to resources like LexisNexis, you can often find the names of executives in technical roles as a starting point.
  - If message delivery confirmation is desired, in the US you can send certified mail with signature verification. The recipient must sign to receive the mail, and you'll get a signed receipt back.

When all that fails

Some vendors remain unreachable even after a number of reasonable good faith attempts to reach them—and by reasonable we mean considerably less than exhausting the entire list above. Some vendors just do not seem to want to be reached, and that is their prerogative. However, we have found that experience is often the best teacher. When a vendor gets surprised by the publication of a vulnerability in their product and it is clear from the report that attempts to notify them were made but failed, it can prompt the vendor to re-evaluate their vulnerability intake and handling processes to make it easier to reach them in the future.

Providing Useful Information

Reporting a vulnerability requires that the vulnerability is well-documented. This typically means providing the following information:

- The exact product version(s) affected
- A description of how the vulnerability was discovered (including what tools were used) or what you were doing when you encountered the vulnerability
- Proof of concept (PoC) code or reproduction instructions demonstrating how the vulnerability might be exploited
- Ideally, a suggested patch or remediation action if the reporter is aware of how to fix the vulnerability
- Description of the impact of the vulnerability and attack scenario (Kymberlee Price discusses the importance of providing a clear attack scenario in her article [2]).
- Any time constraints (for example, give a date of publication or presentation at a conference if you know)

Reporters that do not provide enough information to a vendor or coordinator may find their reports delayed or even rejected. Using CWE [3] or CAPEC [4] as a reference might be helpful to describe the type of vulnerability you have found and common ways to fix it the problem.

An example of a template for a vulnerability report, based on the CERT/CC's own Vulnerability Reporting Form (VRF) [5], is provided in Appendix D. Vendors that require additional information to validate reports should clearly document their specific requirements in their vulnerability disclosure policy, reporting form, or process description.

Advice for Vendors

Vendors need a mechanism to receive vulnerability reports from others. This reporting mechanism should be easy enough to use that it encourages rather than discourages reports. It can be as simple as a dedicated email address for reporting security issues, a secure web form, or a bug bounty program. Aside from the technical aspects of encouraging reporting, vendors can also provide reporters with other incentives, as discussed in Section 2.4.

Create Secure Channels for Reporting

Whether you are a vendor or a coordinator, you need to have open channels for communication with vulnerability finders and reporters. In our experience, the most common means of communication is email. For this reason, the CERT/CC recommends that vendors establish a specific and well-publicized email alias such as <security@example.com> solely for receipt of vulnerability reports.

However, since email is an insecure communications channel by default, many vendors, reporters, and coordinators prefer to use encrypted mail instead. Although x.509 encrypted mail exists, we have found PGP-compatible tools such as GnuPG to be more widely used by CVD participants. Vendors are encouraged to create and publish a PGP key affiliated with the security email alias to allow the confidentiality of sensitive reports to be maintained in transit.

Alternatively, some vendors choose to offer a web form specifically for receiving reports of security-related issues. Such forms can then deliver the report directly to your security or engineering team. The CERT/CC discourages reliance on general "Contact Us" web forms that pass through an organization's communications or customer support teams. Many finders will balk at having to get past these nontechnical interfaces into the vendor. In addition, security messages often must be triaged and processed differently than other incoming contacts.

Another possibility is to make use of a third-party bug bounty or coordination platform. For more information on common CVD tools, see Section 7.

Reduce Friction in the Reporting Process

As a vendor, it is important to not treat reporters with suspicion or hostility. It's likely they have important information about your product, and they want to share it with you.
Furthermore, vendors should take care not to frustrate reporters' attempts to make contact by building too many assumptions into their CVD process.

In the course of our own vulnerability coordination efforts, we have observed all of the following erroneous assumptions built into vendor's CVD process:

- *The vendor and the reporter are engaged in a binding two-way negotiation.* – This is false for at least two reasons: First, the reporter noticed an observable fact about a product flaw, and they're free to tell whomever they please. Similarly, the vendor is not obliged to keep the reporter in the loop once the report is received. And while it might be possible to eventually constrain the reporter or vendor contractually, that's not a foregone conclusion at the outset. Secondly, there exist adversaries who are entirely indifferent to whatever reporter-vendor agreements about disclosure might be reached, so even the most strictly binding agreement between vendor and reporter can be rendered irrelevant by events beyond either party's control.
- *The reporter is always a customer or has a customer ID.* – At the CERT/CC, we have hit walls in our communication attempts when a vendor's technical support function refuses to help us without a customer ID number. Be sure your tech support staff understand how to forward vulnerability reports to the appropriate individuals or groups within the organization. Vulnerability reports can arrive from anyone.
- *The reporter is willing and/or able to fill out a web form.* – Some reporters prefer to use anonymous email; be sure you have more communication lines open than just a web form.
- *The reporter is a human.* – Sometimes reports can be auto-generated by tools. Include a clearly defined reporting format for tools if at all possible.
- *The reporter can send or receive encrypted mail.* – The CERT/CC encourages encrypted mail when possible, but it is not appropriate to presume all reporters can or must use encrypted mail. If the reporter declines to use encrypted mail, offer other options. These may include encrypted zip files or a company upload service such as FTP.

- *The reporter has an account on your private portal.* – The reporter may not be a customer with a portal account; furthermore, the reporter may wish to remain anonymous and will be unwilling to register for a portal account. Again, be sure it is easy for reporters to find more than one communication channel.
- *The reporter will wait indefinitely for your reply before communicating with others about what they know.* – Technology sometimes fails, and we wonder if a message was received. It is helpful to let the reporter know as soon as possible that the report was received. Give regular updates on the process so that the reporter is involved and there is mutual understanding. If reporters are kept out of the loop, they may seek out a third-party coordinator or even publish their report without notice.
- *The reporter will keep your correspondence private.* – Lack of response or rudeness on the part of a vendor may result in the reporter choosing to post the correspondence publicly. In addition to the negative attention this draws to the vendor and reporter alike, such negative experiences may discourage finders and reporters from reporting vulnerabilities to the vendor in the future.

References

1. Wassermann, Garrett. *Reach Out and Mail Someone.* 6 August 2015. https://insights.sei.cmu.edu/cert/2015/08/reach-out-and-mail-someone.html
2. K. Price, "Writing a bug report - Attack Scenario and Impact are key!" 2 August 2015. [Online]. Available: https://forum.bugcrowd.com/t/writing-a-bug-report-attack-scenario-and-impact-are-key/640. [Accessed 17 May 2017].
3. MITRE, "Common Weakness Enumeration (CWE)," [Online]. Available: https://cwe.mitre.org/. [Accessed 17 May 2017].
4. MITRE, "Common Attack Pattern Enumeration and Classification," [Online]. Available: https://capec.mitre.org/. [Accessed 17 May 2017].
5. CERT/CC, "Vulnerability Reporting Form," [Online]. Available: https://vulcoord.cert.org/VulReport/. [Accessed 17 May 2017].

< 4.1 Discovery | 4.3 Validation and Triage >