SEI WEBINAR SERIES | Keeping you informed of the latest solutions

Software Engineering Institute | Carnegie Mellon University

Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0000588

Coordinated Vulnerability Disclosure

Dan Klinedinst

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Vulnerability Response 101

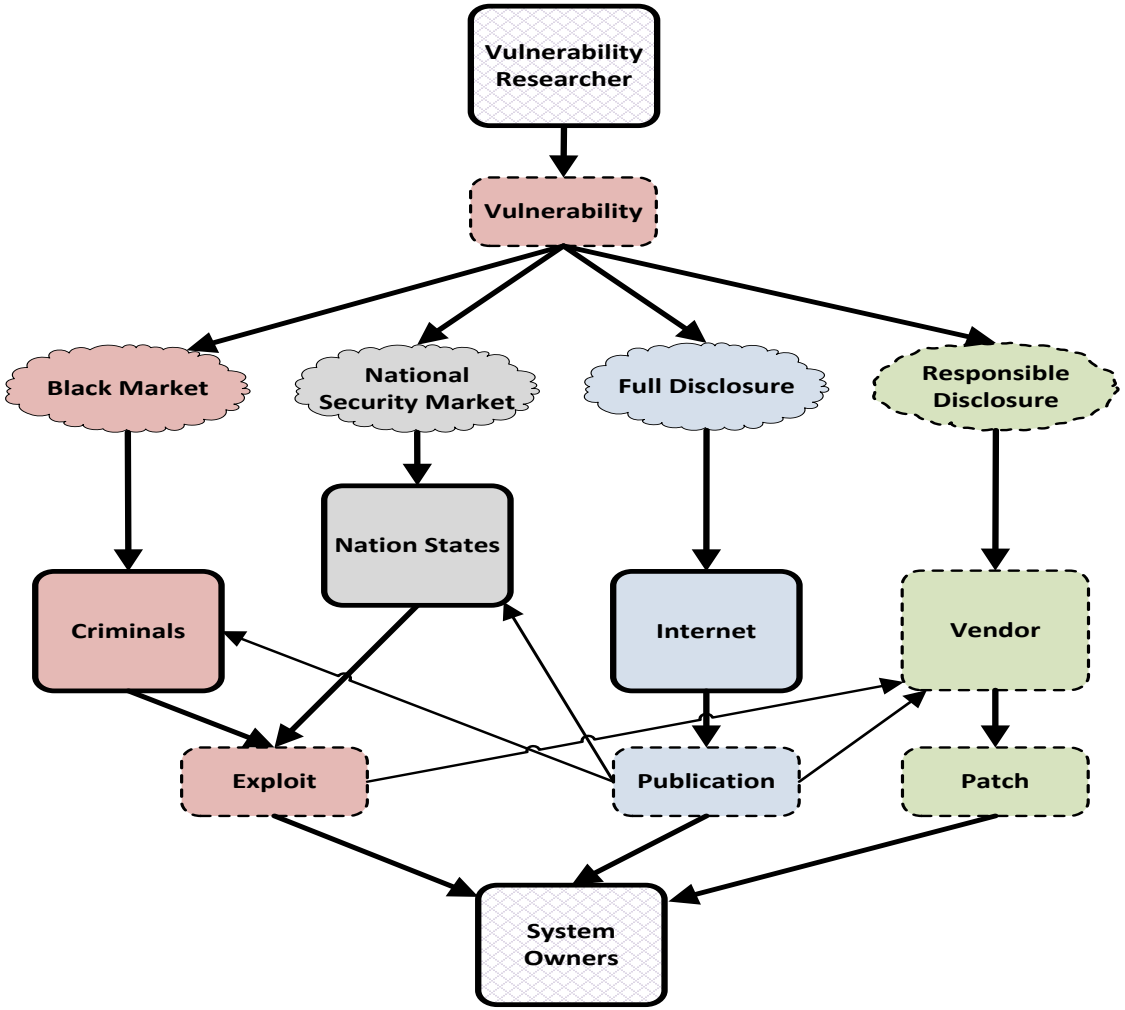
What is Vulnerability Response?

Vulnerability response is the activity of responding to a vulnerability reported **in your organization's** product or service.

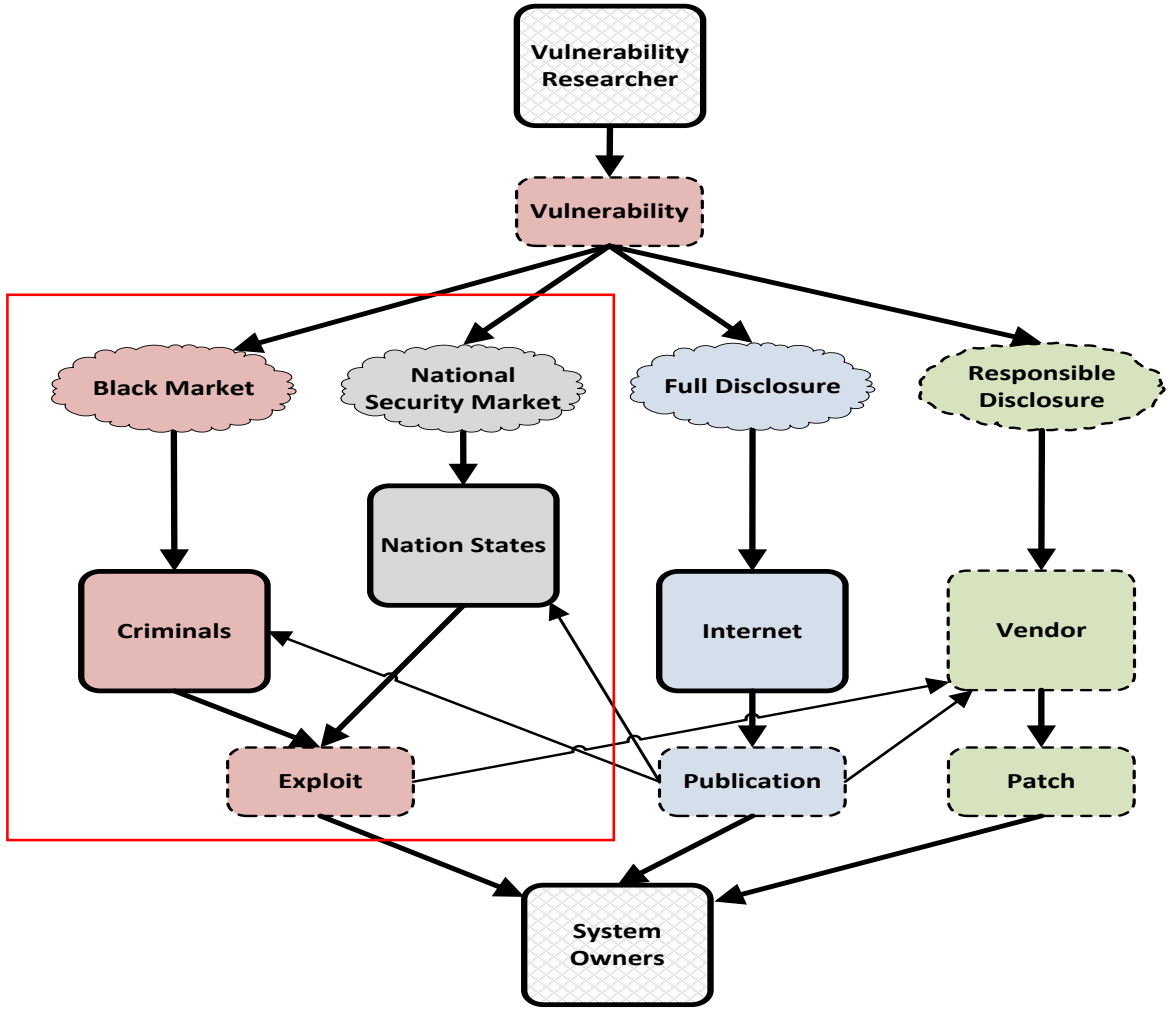
Vulnerability response is part of the larger ecosystem of product security; integrating security processes into the software development lifecycle.

This is not "vulnerability management" (or managing vulnerabilities of products that your organization has purchased or used)

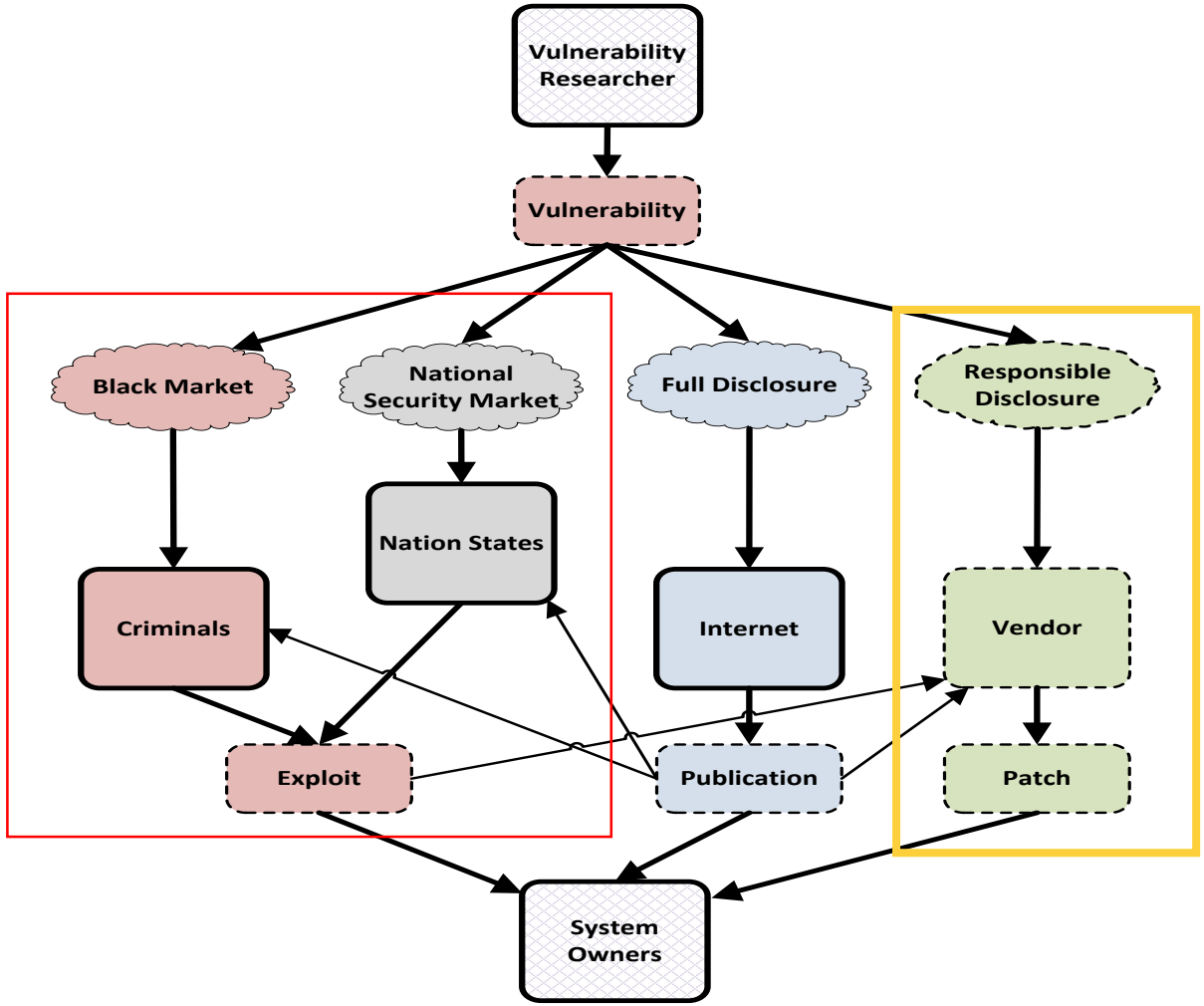
From discovery to the field



From discovery to the field

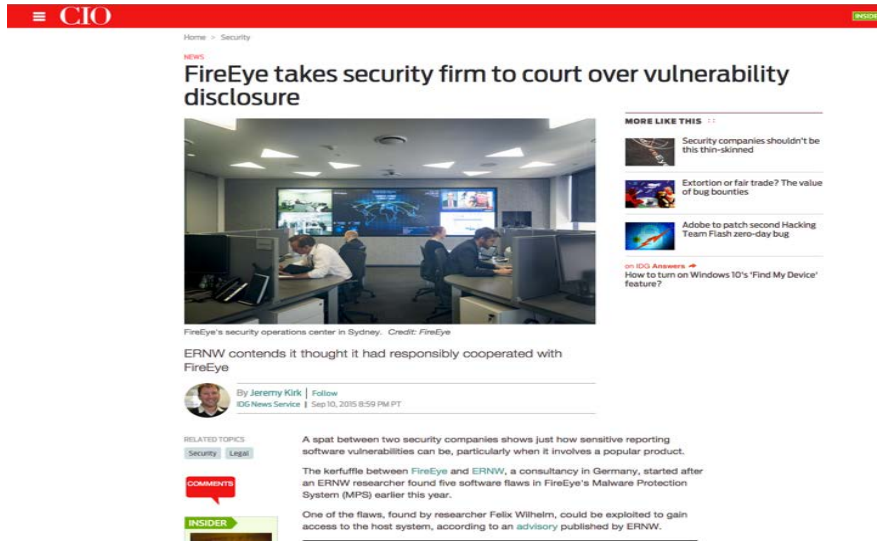


From discovery to the field



Why is vulnerability response important?

- Being unable to be reached by a researcher (or CERT) will result in full disclosure of the vulnerability in your product.
- Slow or improper handling of a vulnerability may result in bad press, legal action, or harm to your customers.
- A good response process can
  - reduce the cost of fixes (by integrating it with your development cycle)
  - encourage researchers to test your software responsibly
  - produce good press about your company



What is a vulnerability?

A vulnerability is:

"a set of conditions that allows an attacker to violate an explicit or implicit security policy. Vulnerabilities can be software defects, configuration or design decisions, unexpected interactions between systems, or environmental changes."

- from CERT definition

All software has vulnerabilities (and they increase with complexity)

Table 1 – Vulnerability density versus defect density measured for some software systems

Systems	Msloc	Known defects	Known defect density (per Ksloc)	Known vulnerabilities	V_{KD} (per Ksloc)	V_{KD}/D_{KD} ratio (%)	Release date
Windows 95	15	5000	0.3333	50	0.0033	1.00	Aug 1995
Windows 98	18	10000	0.5556	84	0.0047	0.84	Jun 1998
Windows XP	40	106500	2.6625	125	0.0031	0.12	Oct 2001
Windows NT 4.0	16	10000	0.625	180	0.0113	1.80	Jul 1996
Win 2000	35	63000	1.80	204	0.0058	0.32	Feb 2000

Alhazmi OH et al., Measuring, analyzing and predicting security vulnerabilities in software systems, Computers & Security (2006), doi:10.1016/j.cose.2006.10.002

Exploits

An exploit is:

"a piece of software or technique that takes advantage of a security vulnerability to violate an explicit or implicit security policy."

- from CERT definition

0-day vulnerability

- A vulnerability that is known by a 3rd party by the time it is publicly discovered (and no patch is available).

Vulnerability Disclosure

Non-Disclosure

A vulnerability that may be reported to the vendor (or not), but is not disclosed to the public regardless if the vendor decides to fix the vulnerability.

Full Disclosure

A vulnerability that is disclosed to the public, so both the vendor and the public find out about it at the same time.

Coordinated Disclosure

Also called "responsible disclosure"

A vulnerability that is disclosed to the vendor, and after a reasonable period of time is disclosed to the public.

Other Terms

Common Vulnerability Enumeration (CVE)

- A unique identification number for public vulnerabilities. Maintained by MITRE at cve.mitre.org
- Not a complete list of all vulnerabilities

Common Vulnerability Scoring System (CVSS)

- A method of determining the severity of a particular vulnerability. Maintained by FIRST at first.org/cvss

The CVE ecosystem

Why people care about CVEs:

- Most security tools use the National Vulnerability Database as a feed to update their vulnerability scanners
- These scanners are widespread and used by security companies, internally at organizations, and by governments.
- When a vulnerability appears on a scanner, the client usually wants to fix it. If there is no fix, they will complain. For some clients (government, military) a vulnerability is a show stopper for implementation.

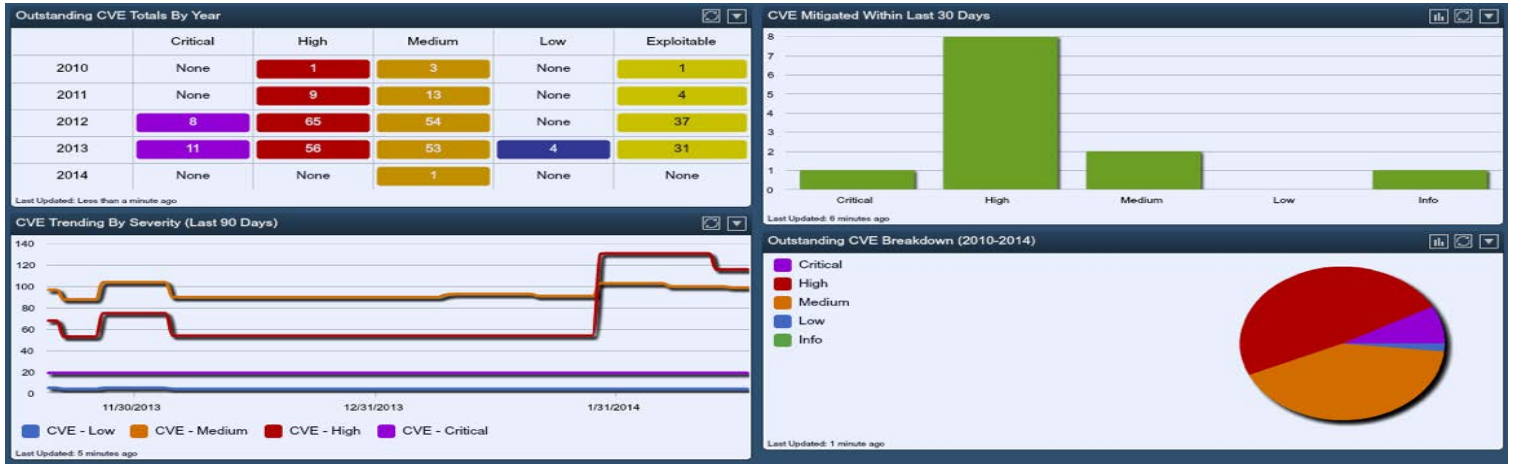


Image courtesy of Tenable.com

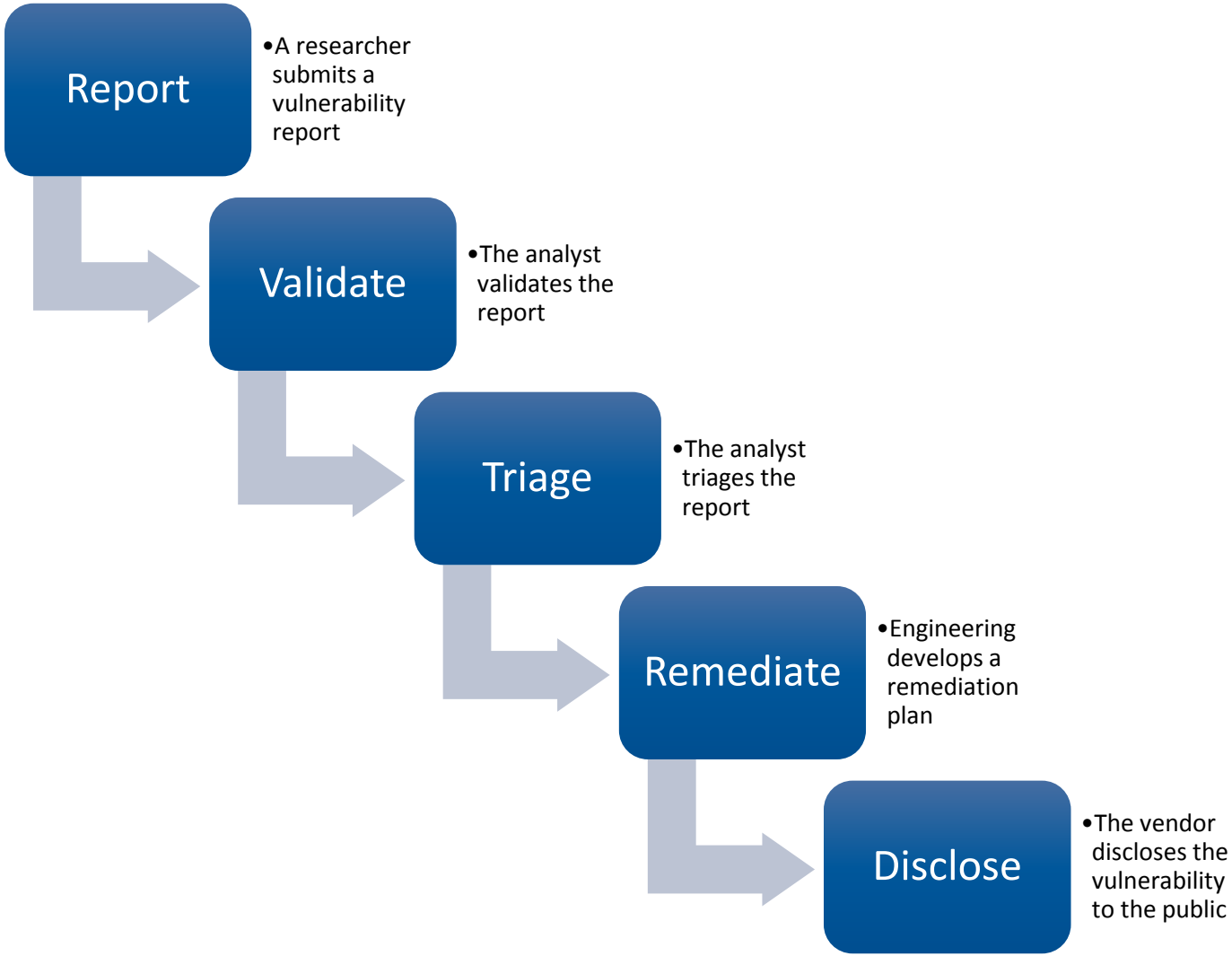
Polling Question

Does your organization have a vulnerability response program or Product Security Incident Response Center (PSIRT)?

A. Yes, fully operational
B. In the process of building one
C. These functions are handled ad hoc.
D. Not currently.

Building Vulnerability Response Processes

Vulnerability Response Process Overview



Vulnerability Response Program

One of the main reasons people contact CERT is to reach out to the vendor for them after failed attempts to contact the vendor.

Usually these vendors end up not having a vulnerability response program.

In many cases, we have to publish a 0-day after 45 days of non-contact.

Building a Program – Initial Steps

Having an easily findable security point of contact on the corporate website is arguably the most important step to forming a Vulnerability Response team.

Recommended Points of Contact:

- security@vendor.com
- Vulnerability Reporting Form
- vendor.com/security page

- Stay active (or at least watch) activity on social media
  - Twitter, Facebook, Linkedin

Building a Program – Initial Steps

- Identify a team member that can answer emails and has reach back into the engineering team
- Identify email address and advertise on public site
- Assign an internal vulnerability identifier
- Add workflow to Product support and PR groups for vulnerability handling
  - Product Support should forward vulnerability related communication to the designed POC.
  - PR should have a process for handling a vulnerability in your product

Polling Question

Which vulnerability analysis function is most crucial to your organization?

A. Processes – Receiving vulnerabilities, triaging them, internal coordination, addressing in a timely manner
B. Maintenance – developing, testing and deploying patches or security fixes to customers / products
C. Culture / Business model – Getting organizational leaders to recognize the importance of vulnerability response
D. Development – Threat modeling, security testing, secure DevOps, secure coding
E. Discovery – Technical expertise and/or tools to identify and validate vulnerabilities

Maturing the Program

After you've built the initial outreach infrastructure, the first vulnerability report will probably be a trial, of sorts, until your team and processes are built up properly.

To mature your program, you will eventually need to:

- Have full-time or part-time support for your security@ email address, twitter, etc.
- Have tools to support the workflow, such as a testing environment, ticketing system, secure communication.
- Public Relations support for dealing with media fallout.
- Development processes to handle vulnerabilities in the disclosure timeframe.
- Triage of new vulnerability reports.
- Skilled vulnerability analysts to review incoming reports.

Communication Plan

The vulnerability response communications plan should clearly identify points of contact, email aliases, and stakeholders.

The plan should identify how to handle situations such as:

- Active exploitation of the reported vulnerability is identified in the wild.
- The vulnerability is privately disclosed by two separate parties.
- The vulnerability has been made public before the agreed upon date.
- The vulnerability is present in more than just your own products.
- The vulnerability is more severe than the researcher suggests.

External communications

- Researchers – A quick email update goes a long way
- CERT – Keep contact info and encryption keys current
- Customers – Communicate both technical information and honest information about potential impact / risk.
- Media – Don’t downplay risks, but correct erroneous information.
  - Back to Jeep hack

Resources

ISO Standards

29147 Vulnerability Disclosure

- ISO/IEC 29147:2014 gives guidelines for the disclosure of potential vulnerabilities in products and online services. It details the methods a vendor should use to address issues related to vulnerability disclosure.

30111 Vulnerability Handling Processes

- ISO/IEC 30111:2013 gives guidelines for how to process and resolve potential vulnerability information in a product or online service.

Contact Information:
Dan Klinedinst
djklinedinst@cert.org

Public Vulnerability Information:
www.kb.cert.org/vuls

Software Engineering Institute | Carnegie Mellon