Uses for CVE root cause mapping

You have the data, now what?

Jeremy West
Sr Manager, PSIRT

Alexander Bushkin
Sr Engineer, PSIRT

Red Hat
Product Security

What we'll discuss today

- What problem are we solving?
- Methodologies and challenges
- Benefits of CWE usage and prioritization
- What is everyone else doing?

Source:
Insert source data here
Insert source data here
Insert source data here

Goal: Use data driven analysis to prevent vulnerabilities

Software Vendor

- Vulnerability counts increase and you feel you're always fixing CVEs
- How do I prioritize with an overload of SAST/DAST results, threat models, and more
- Why are we collecting data for the sake of collecting data
- We understand root cause … now what?

Software Users

- Is the software secure?
- Constant vuln remediation is costly

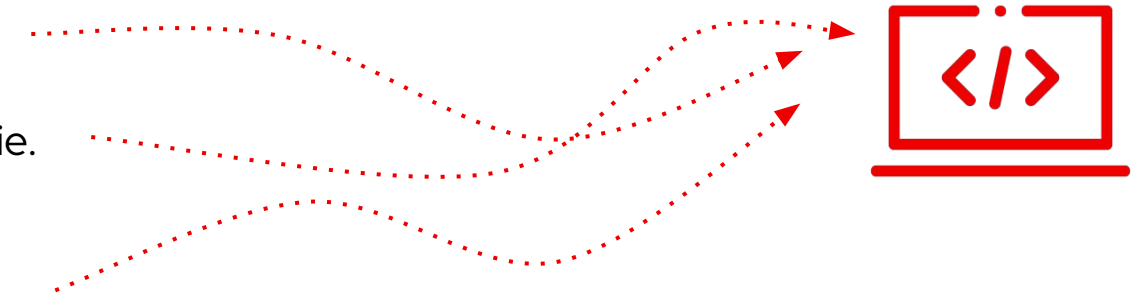
Lost in the land of weaknesses

Beyond the world of CWE

Weakness: a condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities (mitre)

Weakness Origins

- Threat models (bad design)
- SAST/DAST results (bad code)
- Additional Threat intelligence research (ie. external risks)
- CVE root cause mapping (something happened … why)

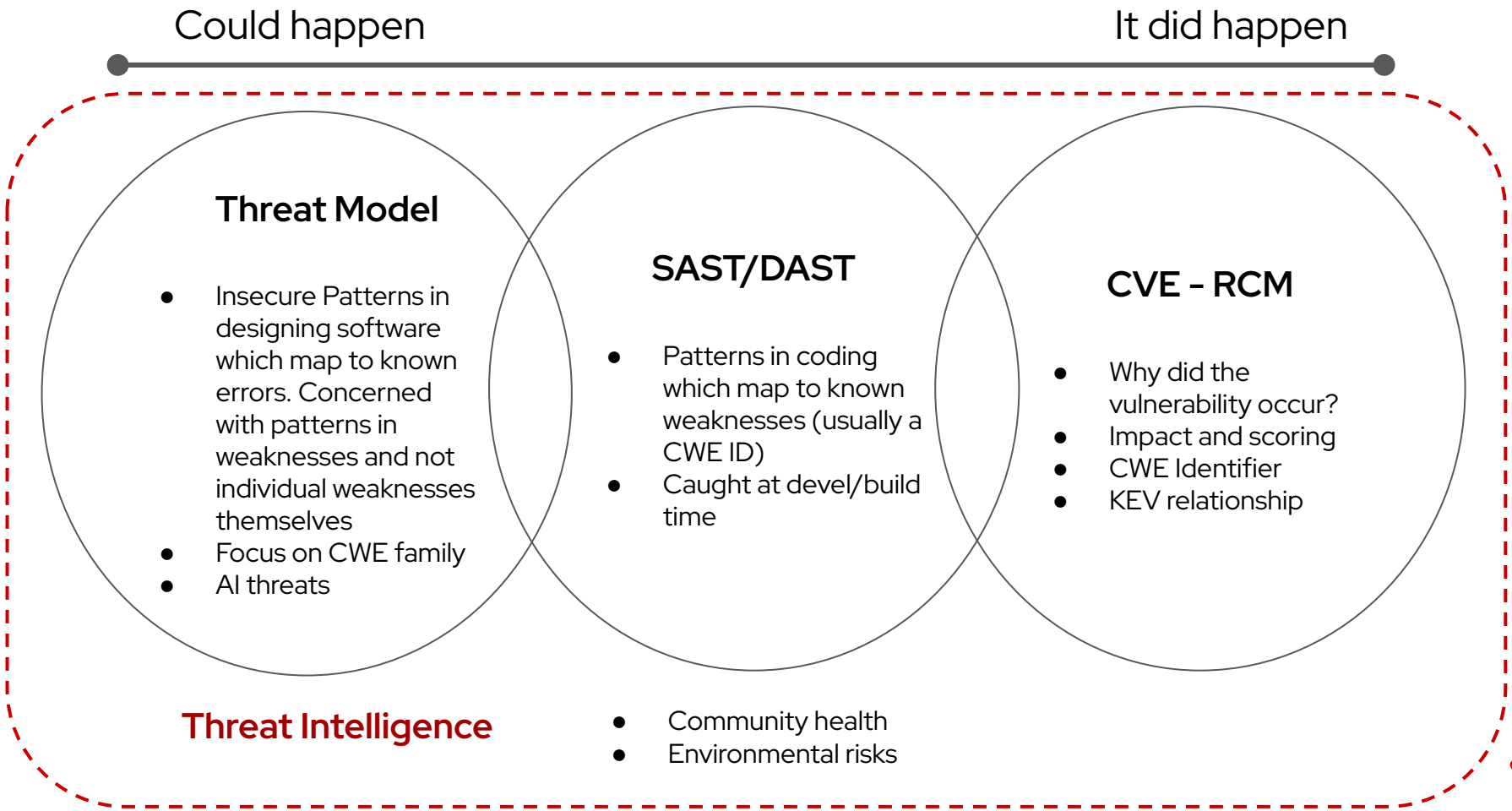


Solution:

Finding commonality in the data

Commonality

What are the options for aligning and prioritizing



Red Hat
Product Security

Standardizing with CWE helps with prioritization

Step 1:

In order to prioritize multiple sets of data, you need to blend that data together somehow. Is <weakness> the same thing as <weakness>. CWE was built for this purpose and is literally a topology for common weakness enumeration.

Recommendation: Classify all weaknesses using CWE IDs (the taxonomy is alive and well!)

Step 2:

You now have lots of CWE data to filter through. How do you do this effectively without creating noise?

Recommendation: <refer to the next slide!>

Option 1: Presenting high level categorization

Intent: Help software developers improve the security posture of code by prioritizing top level weakness themes which contribute to high vulnerability count

1. Narrow to a subset of components where high numbers of vulnerabilities were found.
2. Utilize CVE RC mapping data and focus on CWE categories

1. **webkitgtk**:
   a. **CWE-664**: Improper Control of a Resource Through its Lifetime
   b. **CWE-1218**: Memory Buffer Errors
2. **ghostscript**:
   a. **CWE-707**: Improper Neutralization
   b. **CWE-664**: Improper Control of a Resource Through its Lifetime
3. **thunderbird**:
   a. **CWE-664**: Improper Control of a Resource Through its Lifetime
   b. **CWE-355**: User Interface Security Issues
4. **firefox**
   a. **CWE-664**: Improper Control of a Resource Through its Lifetime
   b. **CWE-355**: User Interface Security Issues
5. **gstreamer1-plugins-good**
   a. **CWE-1218**: Memory Buffer Errors
   b. **CWE-465**: Pointer Issues
   c. **CWE-189**: Numeric Issues

Option 2: Presenting low level weaknesses

Intent: Help software developers improve the security posture of code by prioritizing top level weakness themes which contribute to high vulnerability count

1. Narrow to a subset of components where high numbers of vulnerabilities were found.
2. Utilize CVE RC mapping data and focus on repeat low level CWE IDs

1. **webkitgtk**:
   a. **CWE-119**: Improper Restriction of Operations within the Bounds of a Memory Buffer
   b. **CWE-125**: Out-of-bounds Read
   c. **CWE-200**: Exposure of Sensitive Information to an Unauthorized Actor
2. **ghostscript**:
   a. **CWE-20**: Improper Input Validation
   b. **CWE-23**: Relative Path Traversal Weakness ID: 23
   c. **CWE-121**: Stack-based Buffer Overflow
3. **thunderbird**:
   a. **CWE-120**: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
   b. **CWE-356**: Product UI does not Warn User of Unsafe Actions
   c. **CWE-451**: User Interface (UI) Misrepresentation of Critical Information
4. **firefox**
   a. **CWE-120**: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
   b. **CWE-356**: Product UI does not Warn User of Unsafe Actions
   c. **CWE-451**: User Interface (UI) Misrepresentation of Critical Information
5. **gstreamer1-plugins-good**
   a. **CWE-125**: Out-of-bounds Read
   b. **CWE-476**: NULL Pointer Dereference
   c. **CWE-191**: Integer Underflow (Wrap or Wraparound)

Using CWE data for focused change

CWE as a Connecting Thread

Flexibility

CWE creates a flexible yet precise mapping for security concepts from high-level to low-level
- Ex. Memory Buffer Errors → Out-of-Bounds Reads

Can generalize or specify based on the audience or the framework you are utilizing
- Ex. Threat Modeling requires higher-level concepts
- Ex. SAST/DAST output prioritization benefits from lower-level, specific weaknesses

Standardized & Common Language

CWE creates a shared vocabulary that can be used interchangeably by different stakeholders (ex. developers, researchers, management) to discuss a security concept, concern, etc.

Provides standardization in an industry which is famous for its fragmentation and tendency towards “every vendor has their own terminology”

Additional Solutions

Make CWE data actionable

Additional Recommendations

- Tailor the results based on the audience. Use higher-level CWEs to communicate security risks to higher-level audiences (leadership) and more technical, lower-level CWEs to communicate security risks to lower-level audiences (engineering)
- Remember upstream contributors. Communicate results to upstream contributors; help guide the security focus in OSS communities by tying them back to real-world (tangible) issues
- Integrate CWE metrics into development pipelines; use it as a cross-check with SAST/DAST output to prioritize the weaknesses to focus on. (We have another presentation specifically on this topic!)
- Weakness resolution is not limited to just a patch in code; includes guardrails, security practices, controls, hardening, etc

Audience Question

How are you leveraging CWE data?

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat
Product Security