BSP SECURITY MAINTENANCE

Best practices for triaging CVEs in embedded systems

August 2020

SECURE CONNECTIONS FOR A SMARTER WORLD

Agenda

- Importance of security triaging
- Typical triaging process
- CVE assessment
  - Standardized metrics
  - Attack Vectors
  - Exploitability of security issue (complexity, interaction, effort needed)
  - Impact metrics
  - Temporal Metrics
- CVSSv3 vector string
- Calculating your own assessment values
- Demonstration - Triaging process with NXP Vigiles
- Q&A

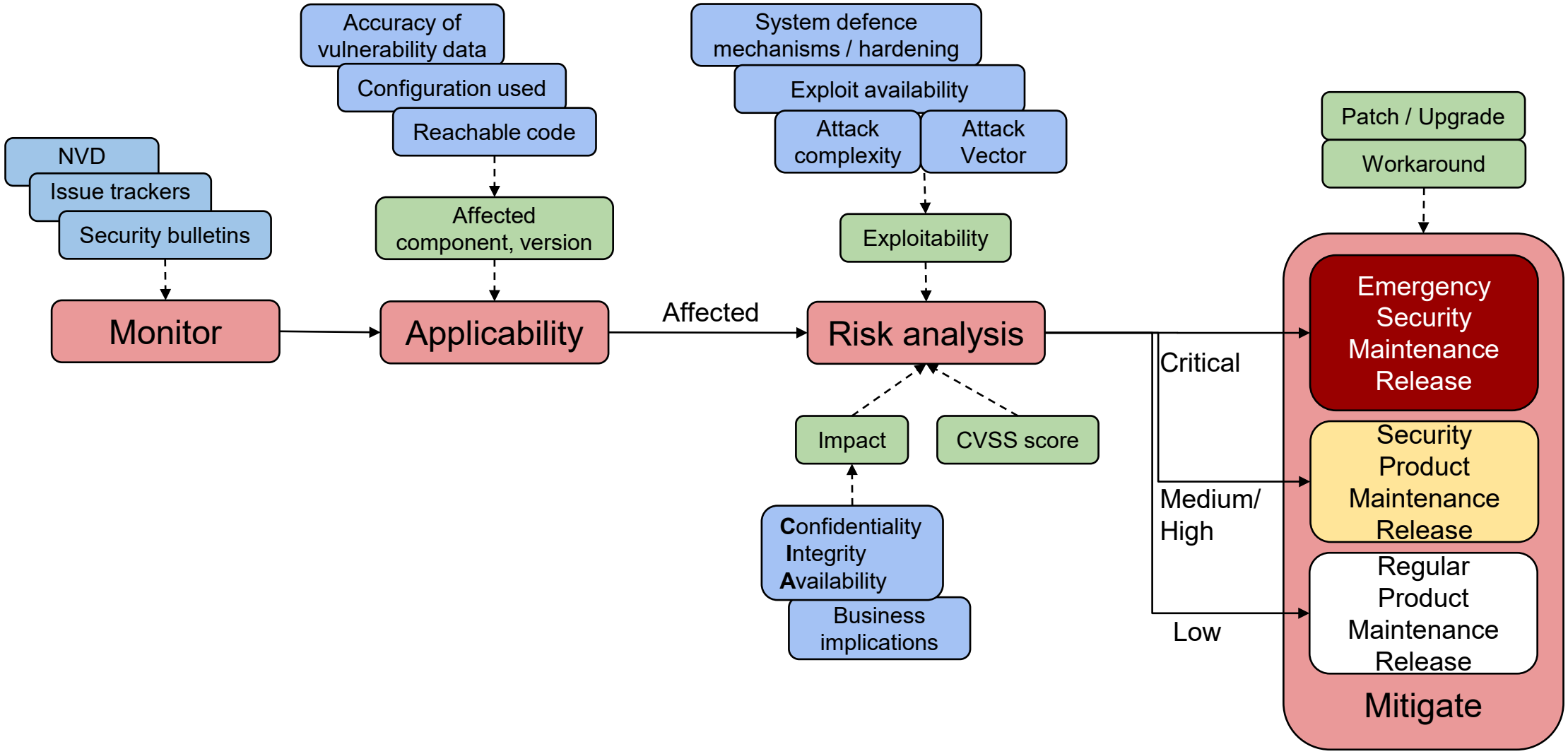
Security Triaging

- Security of a product
  - Implemented features
  - Software components used
  - Policies
  - Product location
  - Desired use
- Sources of security information
  - Audit
  - Security assessment tools
    - SCA
    - Binary
  - Customer reports
  - Vendors announcements/security bulletins
  - Forums
- Need a process that drives a response to security vulnerabilities
- Focus on vulnerabilities that matter most
- Triaging is a process of managing which applicable vulnerabilities to address and when

Triaging Process Considerations

- Is there a single process that applies?
- What does it involve?
- How much time will it take?
- What level of engineering expertise do I require?
- How does the triaging process align with business needs?
- Are there any tools and/or processes I can leverage?

Triaging Process — Security Impact Assessment



Urgency Action Levels

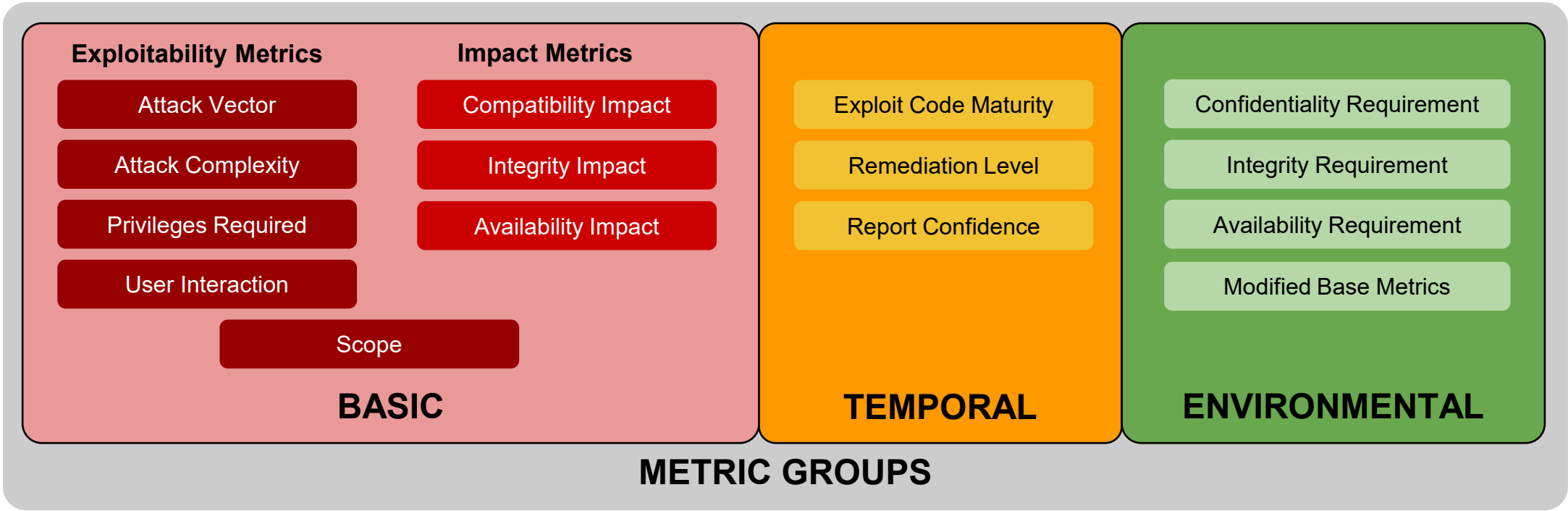
- Typical actions from triaging process

Urgency Action	Description
No action required	The product is not affected by vulnerabilities.
Regular product maintenance release	The vulnerability poses low threat to the product and company liability. The vulnerability can be mitigated as part of a product's standard maintenance cycle.e.g. 1 year.
Security product maintenance release	The vulnerability poses medium or high threat to the product. Such vulnerabilities shall be addressed in the next security product maintenance release. Depending on the product, such releases are scheduled 2 - 4 times a year.
Emergency security product maintenance release	The vulnerability poses a critical threat to the product and company's liability. Confidential information or proper functionality of the product can be jeopardized. Emergency security product maintenance release shall be scheduled immediately. If possible, affected product functionality shall be temporarily disabled. Notice of threat shall be provided to product users.

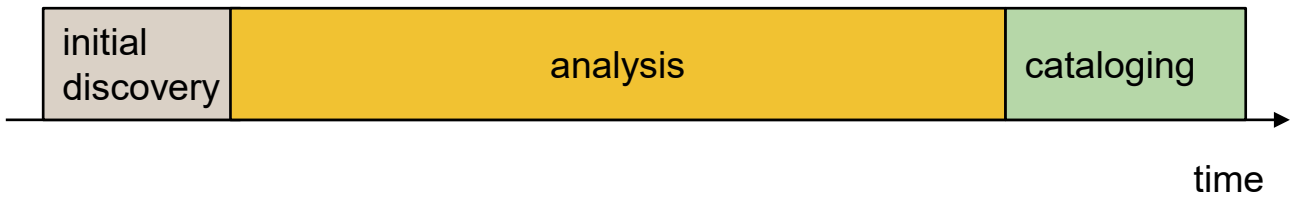
CVE Vulnerability Metrics

- Common Vulnerability Scoring System
  - Score - numerical metric 0-10, that represents severity of security vulnerability information.
  - Vector - Provides scoring information split into different parameter groups

CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:L/E:F/RL:O/RC:U/CR:L/IR:X/AR:L/MAV:L/MAC:L/MPR:N/MUI:R/MS:U/MC:N/MI:N/MA:N



Base Parameters



- NIST goes over the analysis and cataloging exercise. They assign values to Base Metrics
- Base Metrics provide generic information about vulnerability exploitability and impact
  - The Exploitability metrics reflect the technical means and ease by which the vulnerability can be exploited
  - The impact metrics reflect the direct consequence of a successful exploit

Select Base Metrics — Attack Vector (AV)

- Describes the context of the possible exploit
- The more remote an attack can occur, the higher the score
- For more remote attacks, larger group of attackers possible

Metric Value	Description
Network (N)	Remotely exploitable vulnerability at the protocol level. Example - Denial of Service (DoD) attack by sending a specially crafted TCP packet across WAN.
Adjacent (A)	Bound also to a network stack, attack limited to a logically adjacent topology. Attack launched from the same shared physical or logical network.
Local (L)	System can be exploited by read/write/execute capabilities. Exploit by accessing system logically (e.g. console) or remotely (e.g. SSH). Exploit may require user interaction.
Physical (P)	Attack requires physical access to the vulnerable component. Example - Cold boot attack, peripheral attack.

Vector designation: AV:<Value>

Select Base Metrics — Privileges Required (PR)

- Describes the level of privileges attacker must have to successfully exploit given vulnerability

Metric Value	Description
None (N)	Highest risk. Attacker does not require any privileges, access to settings or files to carry out an attack.
Low (L)	Attacker requires privileges for a basic user capabilities. Attacker has limited scope of attack as it can only access files and settings owned by a user, limiting access to the system.
High (H)	Attacker needs user privileges with significant access/control (e.g. administrative) to carry out an attack.

Vector designation: PR:<Value>

Select Base Metrics — User Interaction (UI)

- Need for another human user, other than the attacker, to carry out an attack
- User is not typically aware of an attack when participating

Metric Value	Description
None (N)	System can be exploited without any interaction from another user
Required (R)	Attack requires a user (not attacker) to take certain action (e.g. start a process) for the successful exploit

Vector designation: UI:<Value>

Select Temporal Metrics — Exploit Code Maturity (E)

- Reflects the likelihood of the attack
- Based on state of the expolit techniques, exploit code availability and skill set needed
- The more exploit code is available, the more attackers can leverage it

Metric Value	Description
Not Defined (X)	Insufficient information to choose any of the other values
High (H)	Exploit code exists or is not required (manual trigger).Exploit works in every situation. Exploit is mature, widely available and easy to use.
Functional (F)	Functional exploit code is available. It works in most situations where the vulnerability exists.
Proof-of-Concept (P)	Only PoC code is available. The exploit iis not functional in all situations and may require further modifications and skills.
Unproven (U)	Exploit code is unavailable or its theoretical

Vector designation: E:<Value>

Select Temporal Metrics — Remediation Level (RL)

- Important for vulnerability prioritization
- Fix for the vulnerability may not exist, or a workaround can be used ahead of an official fix release
- Urgency of the vulnerability decreases as the fix becomes final.

Metric Value	Description
Not Defined (X)	Insufficient information to select any other value. No impact on temporal score
Unavailable (U)	Solution is not available or it can not be applied
Workaround (W)	Unofficial solution is available. Typically originating from users of affected component and not the vendor. Developers typically provide workaround steps
Temporary Fix (T)	Official (still temporary) fix is available. Example - vendor provided temporary hotfix
Official Fix (O)	Verified, complete solution is available from a vendor. Solution can come in a form of a patch or an upgrade of the vulnerable component

Vector designation: RL:<Value>

Metric Group	Metric Name (and Abbreviated Form)	Possible Values	Mandatory?
Base	Attack Vector (AV)	[N,A,L,P]	Yes
	Attack Complexity (AC)	[L,H]	Yes
	Privileges Required (PR)	[N,L,H]	Yes
	User Interaction (UI)	[N,R]	Yes
	Scope (S)	[U,C]	Yes
	Confidentiality (C)	[H,L,N]	Yes
	Integrity (I)	[H,L,N]	Yes
	Availability (A)	[H,L,N]	Yes
Temporal	Exploit Code Maturity (E)	[X,H,F,P,U]	No
	Remediation Level (RL)	[X,U,W,T,O]	No
	Report Confidence (RC)	[X,C,R,U]	No
Environmental	Confidentiality Requirement (CR)	[X,H,M,L]	No
	Integrity Requirement (IR)	[X,H,M,L]	No
	Availability Requirement (AR)	[X,H,M,L]	No
	Modified Attack Vector (MAV)	[X,N,A,L,P]	No
	Modified Attack Complexity (MAC)	[X,L,H]	No
	Modified Privileges Required (MPR)	[X,N,L,H]	No
	Modified User Interaction (MUI)	[X,N,R]	No
	Modified Scope (MS)	[X,U,C]	No
	Modified Confidentiality (MC)	[X,N,L,H]	No
	Modified Integrity (MI)	[X,N,L,H]	No
	Modified Availability (MA)	[X,N,L,H]	No

Source: first.org

Calculating Own Assessment Values

- Why
  - Allows for more fine tuned assessment of the CVE impact on specific design and product
  - Can take into account product's own risks
  - In-house security teams can define company own metrics
  - Helpful when supporting other business units with a single platform
- Business Implications
  - Accurate assessment is important to minimize collateral damage
  - Lack of ongoing assessment can lead to revocation of certification
- Compliance
  - Example — HIPAA: Requirement in environmental metrics - importance of confidentiality rating is high

CVSS Score Calculator https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.

CVSS Base Score: 8.1
Impact Subscore: 6.0
Exploitability Subscore: 1.4
CVSS Temporal Score: 8.1
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 8.1

Show Equations

CVSS v3.1 Vector
AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H/E:H/RL:X/RC:C

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
Network (AV:N) | Adjacent Network (AV:A) | **Local (AV:L)** | Physical (AV:P)

Attack Complexity (AC)*
Low (AC:L) | **High (AC:H)**

Privileges Required (PR)*
None (PR:N) | Low (PR:L) | High (PR:H)

User Interaction (UI)*
None (UI:N) | Required (UI:R)

Scope (S)*
Unchanged (S:U) | **Changed (S:C)**

Impact Metrics

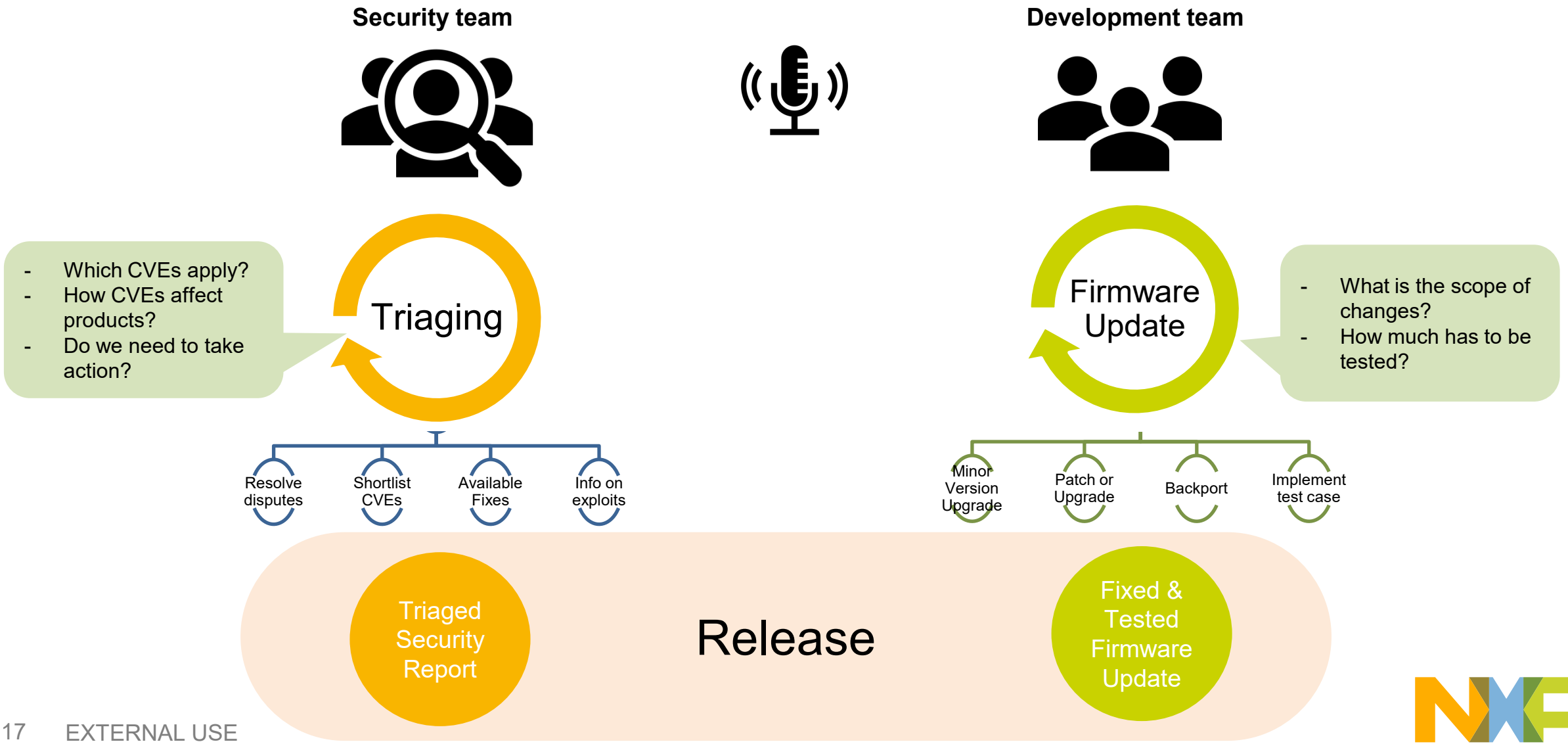
Confidentiality Impact (C)*
None (C:N) | Low (C:L) | **High (C:H)**

Integrity Impact (I)*
None (I:N) | Low (I:L) | **High (I:H)**

Availability Impact (A)*
None (A:N) | Low (A:L) | **High (A:H)**

* - All base metrics are required to generate a base score.

Triaging and Fixing



Effort Involved

- Assumptions:
  - Single product
  - One BSP
  - 100 RFS packages
  - LTS Linux kernel
- Effort includes:
  - Monitoring of vulnerabilities
  - Linux kernel triaging
  - RFS triaging
  - Team collaboration and sharing
- Effort: 12-15 man-months/year

Tools

What to look for in a triaging tool?

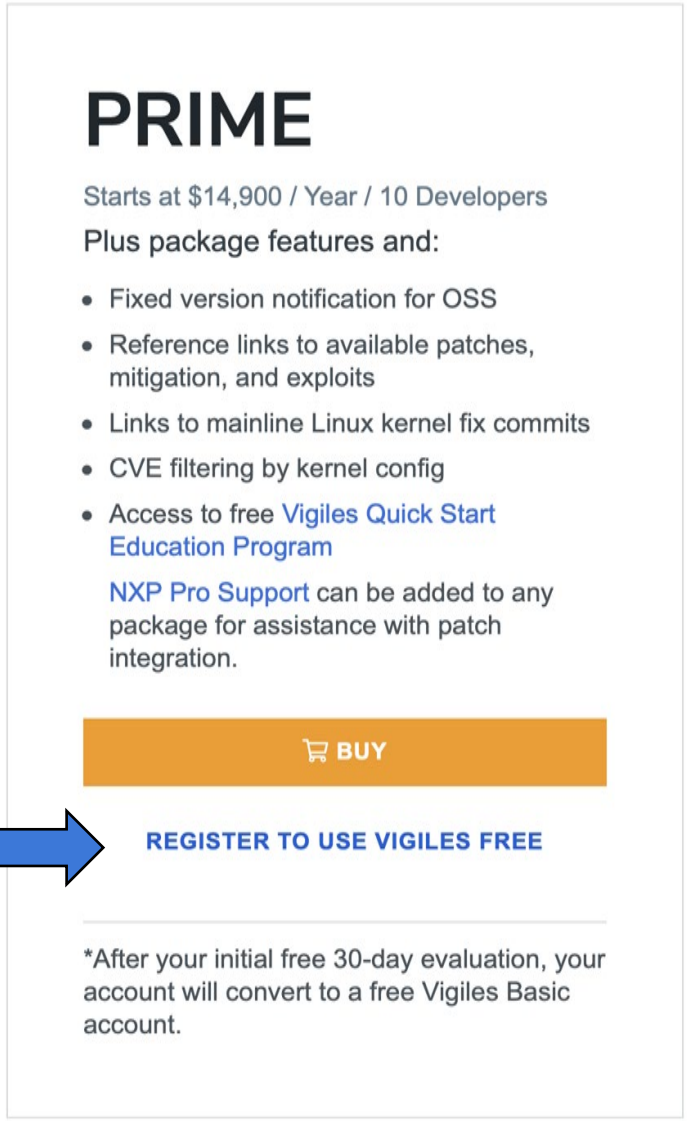
- Accuracy
- Configuration specific vulnerability information (Linux kernel, U-Boot)
- Filters (Attack vector, CVSS, Package, Status, Platform)
- Ability to specify and filter based on Custom Score
- Ability to triage with traceability over time
  - per package
  - per product release
- Ability to whitelist CVEs and complete packages
- Per triage reports
- Build system integration
- API for integration into company processes

Triage Demo

Takeaways

- Establishing a process of monitoring and triaging vulnerabilities is of utter importance
- Automation is your friend = less effort and less time
- Benefits of using NXP Vigiles:
  - Superior vulnerability data
  - Optimized for embedded systems
    - Kernel config triage option reduces triage effort by 4x
  - Intuitive prioritization and powerful filtering mechanisms
  - Collaboration

www.nxp.com/vigiles



More Information

- Visit www.NXP.com/Vigiles
- Sign up for a free trial
- Review your BSP to see how well you are *(not)* covered!

Have questions or need help? Write us at prosupport@nxp.com

Thank You!

Q & A

SECURE CONNECTIONS
FOR A SMARTER WORLD