presented by

Vulnerability Management in UEFI

UEFI Fall 2023 Developers Conference & Plugfest

October 11, 2023

Presented by Brian Mullen

Agenda

- Introduction
- Background – SSDLC Primer
- UEFI Vulnerability Management
- Questions

Presenter Bio

- 20+ years developing C/C++ firmware for embedded systems for telecommunications and consumer electronic devices (Past - Fujitsu/Motorola/Google/Arris)
- DRM Security Lead for Motorola's set-top box BU. Managed secure integration of DRM technologies in to products.
  - Cablecard DRM
  - Secure Media HDD DRM
  - DTLA's DTCP-IP streaming DRM
- Project Owner for Comcast's OG-1600 Outdoor WiFi AP Product

- AMI's Head of SSDLC Organization

Before we turn focus to UEFI…

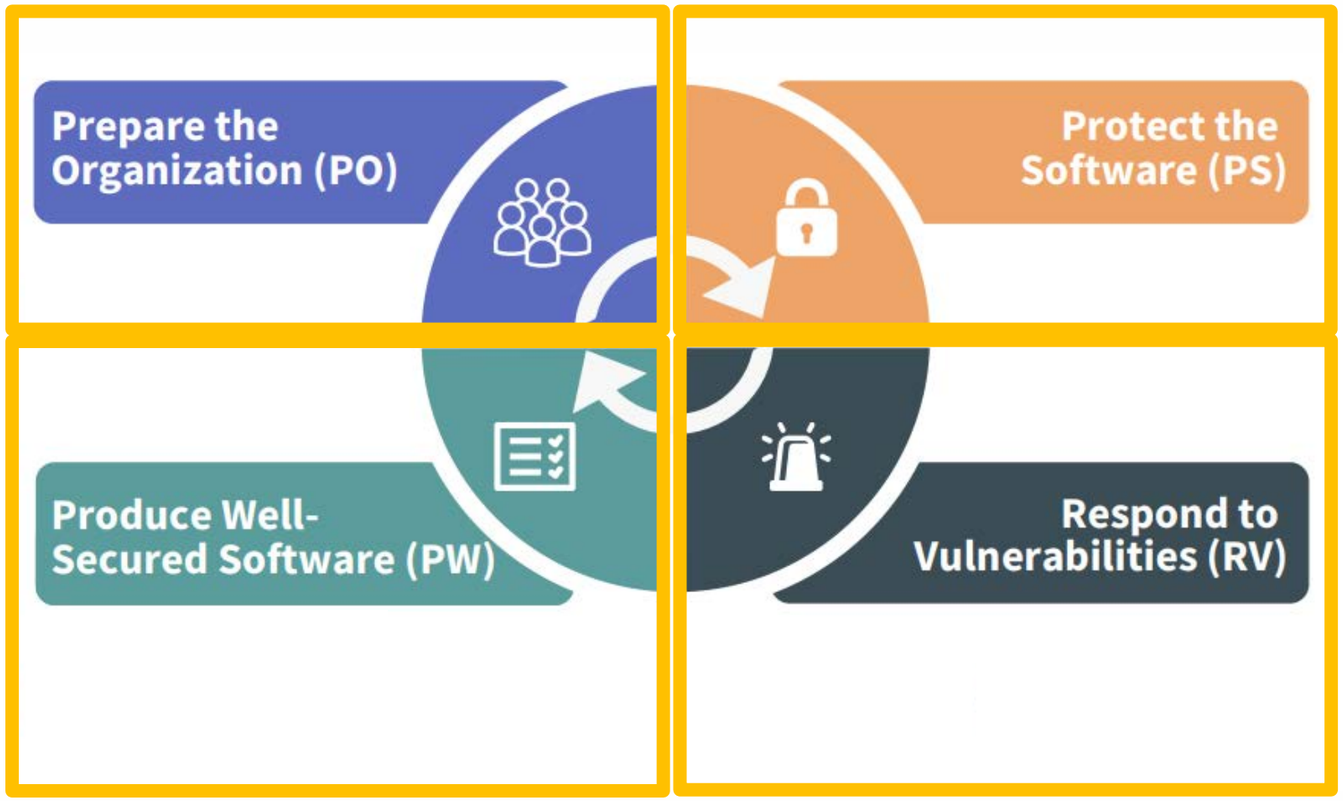
Background – An SSDLC Primer

What is the SSDLC Org?

A decentralized Product Security Team with a mandate to implement NIST SP 800-218 across the Engineering Organization

What is SP 800-218 About?

Requirements, Processes, Roles



Secure Dev Env
Secure Release
Code Signing

Proactive Security

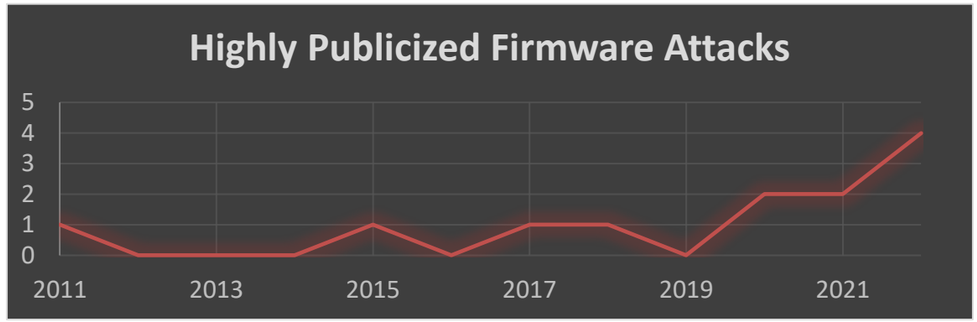
Reactive Security

Managing Vulnerabilities in the UEFI Supply Chain

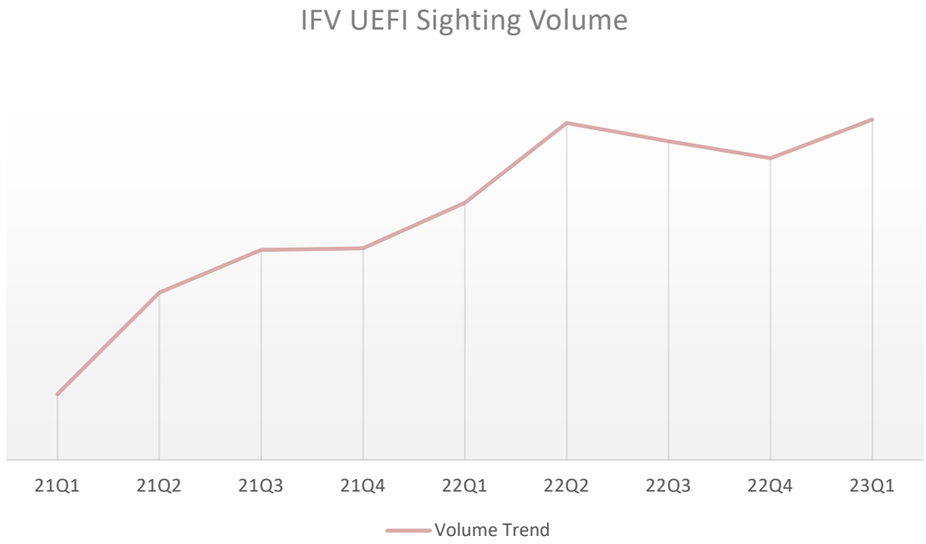
Vulnerability Trends in UEFI

The Rate of Firmware Attacks are Increasing

Exploit	Year
Mebromi	2011
Hacking Team	2015
Equation Group/Vault 7 Leaks	2017
LoJax	2018
MosaicRegressor	2020
Trickboot	2020
FinSpy	2021
ESpecter	2021
MoonBounce	2022
Conti Group Leak	2022
CosmicStrand	2022
BlackLotus	2022



Rogue Actors



Ethical Hackers

Why Invest in a Secure Software Development Lifecycle?

- Reduce Risk to the Supply Chain
- Reacting better/faster when vulnerabilities are identified.
- Gov’t Mandates/Looming Regulations (EO 14028)
  - This was the catalyst for NIST SA 800-218
- Brand Reputation
  - Affects relationships with other supply chain partners
  - Can affect realized value of Organization
- Operating Costs
  - PSIRT is most costly and time-consuming phase of the SDLC to fix an issue

How to Implement an SSDLC

1. Break out of a cycle of reacting
2. Shift-Left into Proactive Security

Breaking the Reactive Cycle

1. Properly staff the team
   - 75% of team should have coding skills
   - 25% for Process, training, project management, auditing/compliance
2. Modernize Vulnerability Management
3. Automate, Automate, Automate

Reactive Security

PSIRT

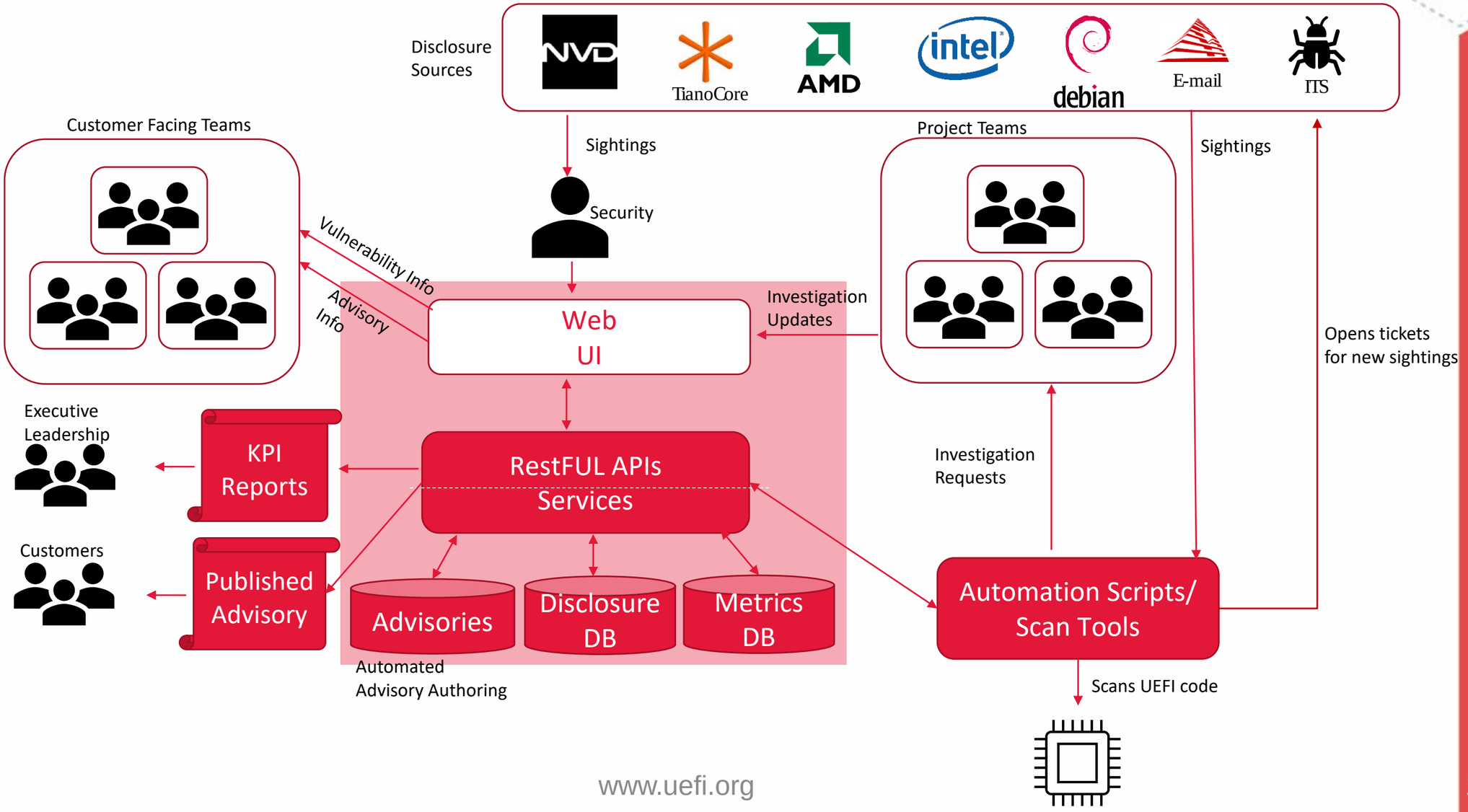
- Create Processes for AMI PSIRT
- Create security remediation processes for Engineering Teams
- Create tools and systems for vulnerability management for PSIRT and Product Engineering and Customer Engineering Teams
- Perform risk assessments (CVSS/CWE) on all vulnerabilities
- Publish Advisories (NDA and Public)
- Work with PO on security related marketing material, security related notices target at customers, web site content
- Operate in a hybrid manner

Use a Modern VMS

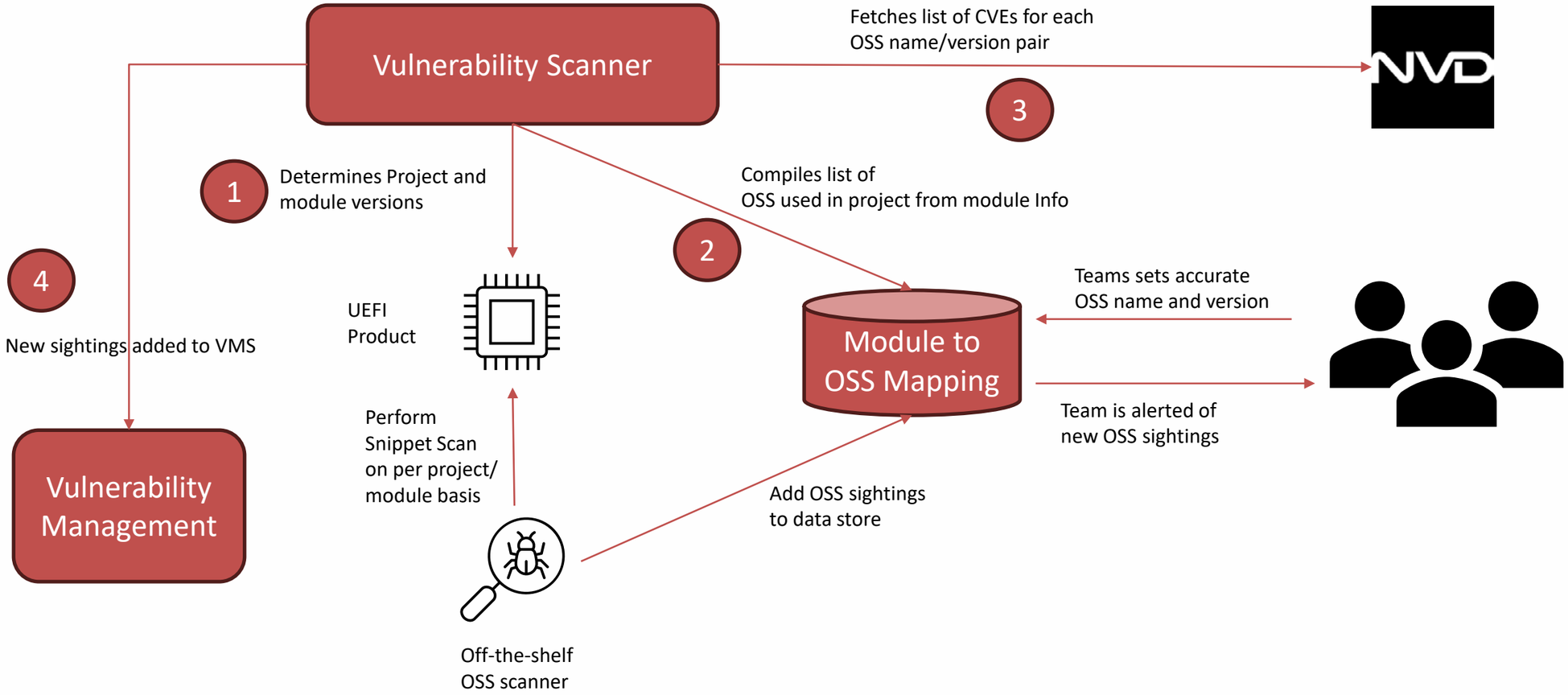
Off-the-shelf or in-house solution

- Off-the-shelf option will require customizations

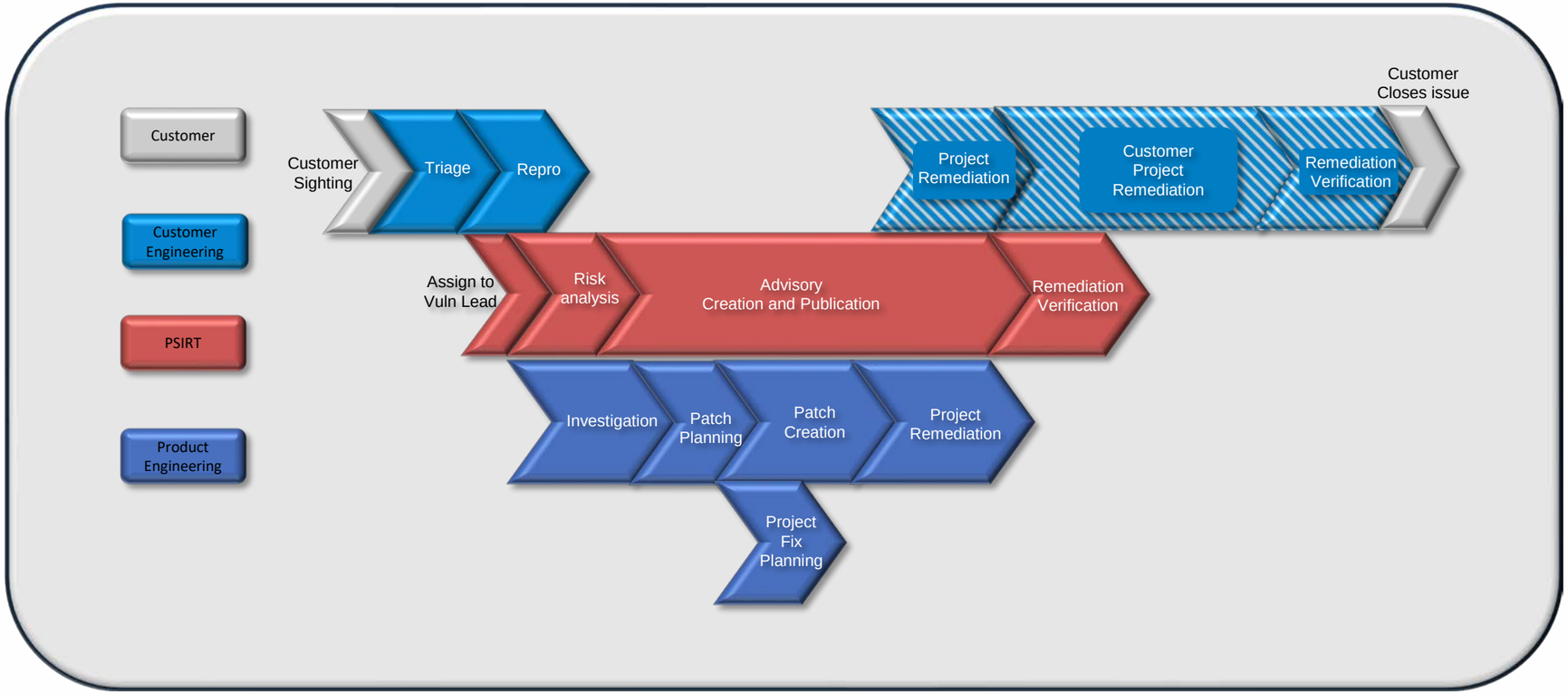
Custom Vulnerability Management



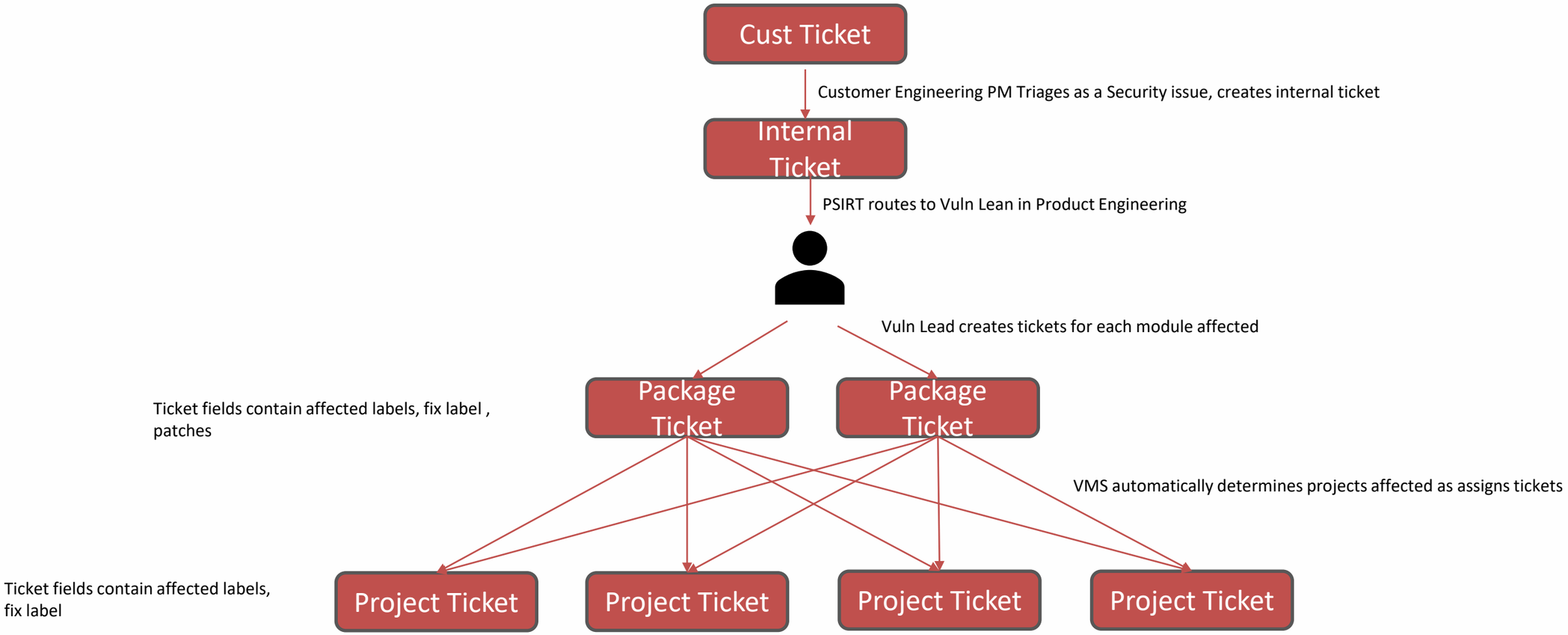
UEFI Vulnerability Scanner



Automate the Vulnerability Lifecycle



Vuln Lifecycle Management



Responsiveness

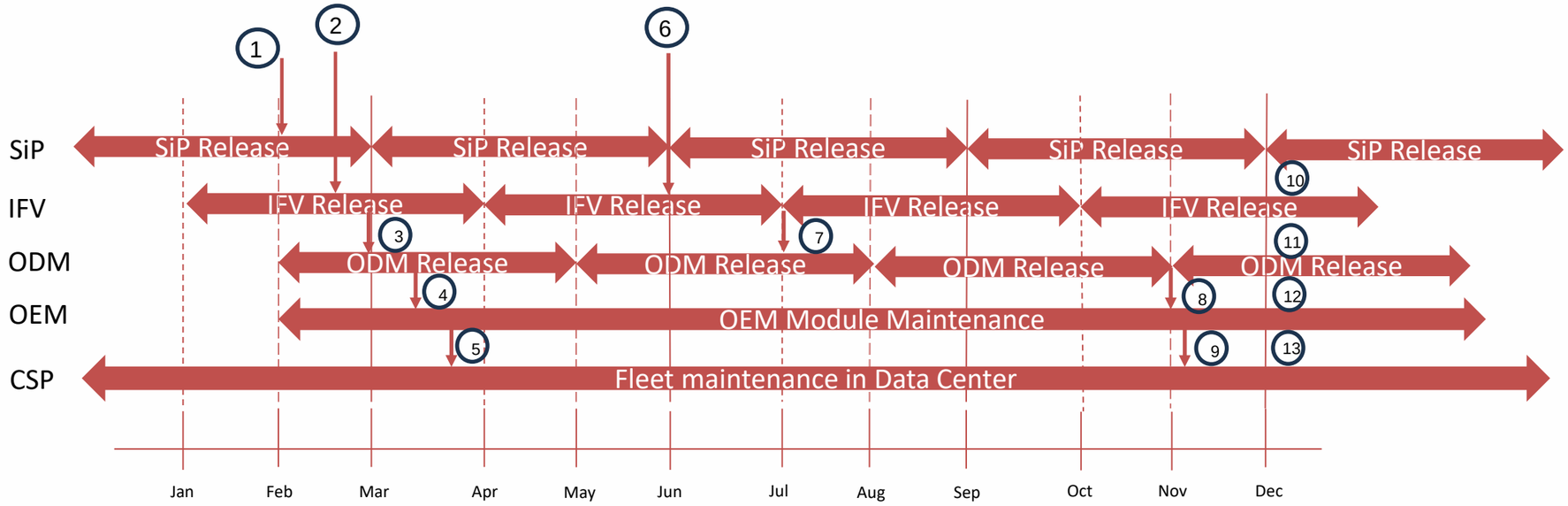
Key Metric for Product Security Incident Response Team (PSIRT)

- Responsiveness is the measure of time it takes to reach each phase of the vulnerability Lifecycle (and ultimately remediate the issue)
- Supply chain partners ask their upstream partners for commitments on responsiveness they can in turn offer commits to their downstream partners
- Researchers want to go public ASAP and usually start with a 60- or 90-day embargo period after the initial reporting of the sighting

Reasonable Expectations

- Supply chain partners should be able to provide history responsiveness KPIs
- Supply chain partners should have upstream partner commit to responsiveness guidelines and record this in SLAs

UEFI Supply Chain Remediation

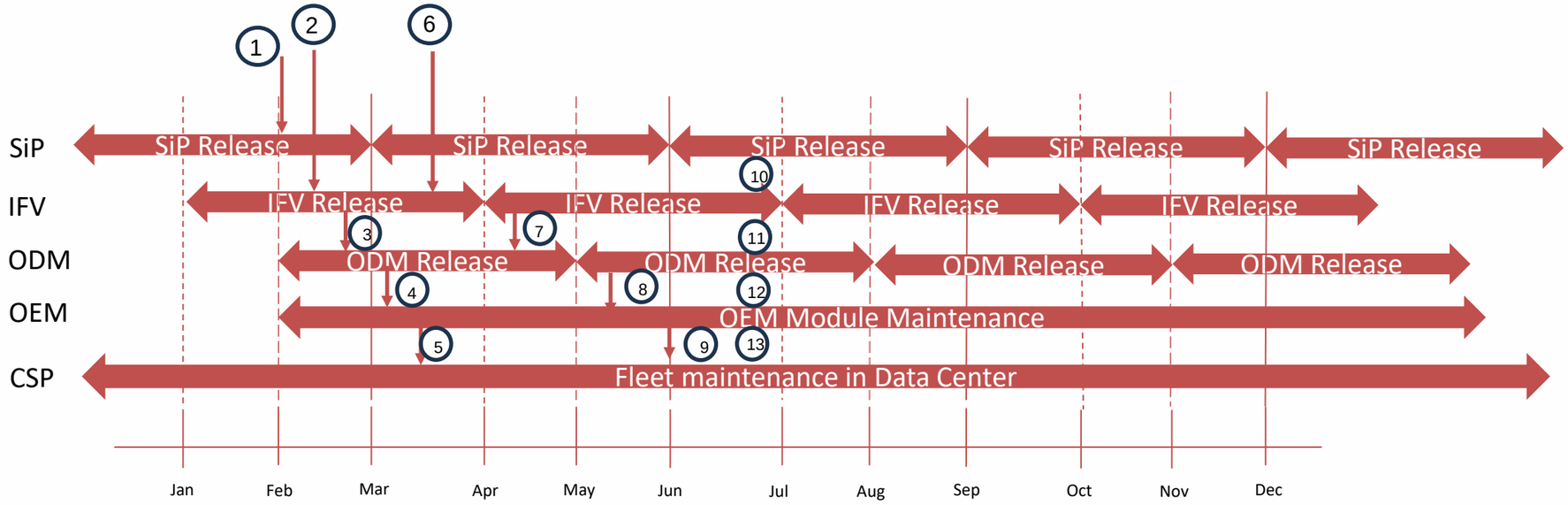


We need to work together to remediation issues faster

Remediation Steps Detailed

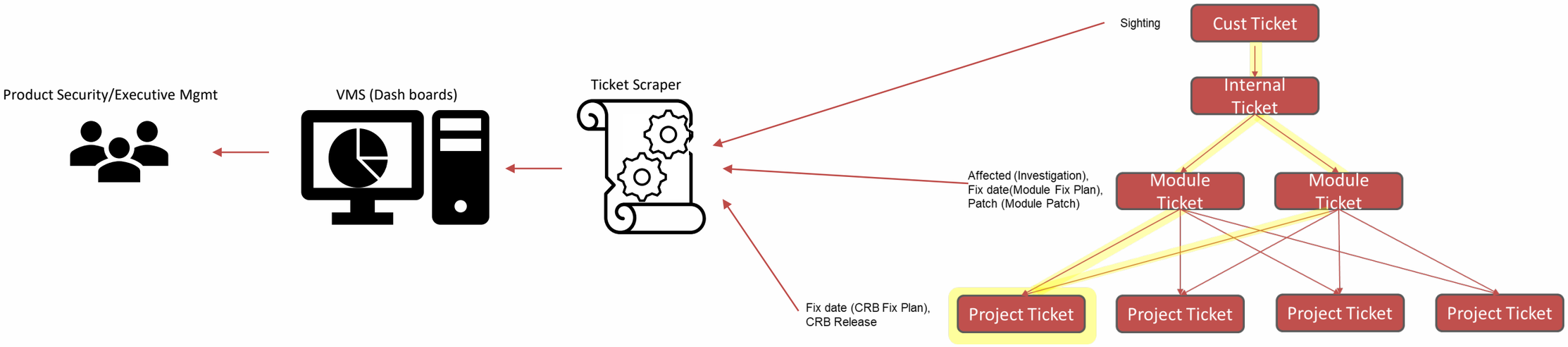
1. It immediately begins an investigation into which product is determined to be affected. The SiP issues an advisory to the IFV within a week (which is a reasonable response time) of the sighting.
2. The IFV performs an investigation and determines its product is also affected so an IFV advisory is issued to the ODM.
3. The ODM follows suit by issuing an advisory to the OEM after it determines that it has inherited the same vulnerability.
4. And finally, the OEM issues an advisory to all its CSP customers that the issue exists, and a fix is being worked on.
5. The SiP addresses the issue in the first release the patch is available in.
6. The SiP may have had the patch earlier but typically SiPs only provide remediations in releases which are made available on a quarterly basis. After the SiP issues the patch, the IFV integrates the fix. Typically, this integration takes a minimum of 3 weeks. In this illustration, it can be observed that the IFV is able to integrate the changes just in time to have the fix available in the release issued at point 7.
7. Unfortunately, the remediation was made available to the ODM at a bad time as there is not enough time to include the remediation in the next release so it must be targeted for the subsequent release which is almost four months away and at that time it is made available to the OEM (to put up on their website so customers can download it).
8. After the OEM makes the UEFI FW release available, the CSP will download the image and update all the servers in the data center.
9. The next step is public disclosure. Given the organic sequence of events detailed in the illustration above, October 1st, ~300 days after the initial sighting, seems like an appropriate time to go public. In this case, a 300-day embargo period seems reasonable for the UEFI supply chain as opposed to the arbitrary 90-day embargo period.
10. Steps (10), (11), (12), and (13) represent all supply chain partners issuing public advisories after the 300-day embargo period expires

Patches Enable Acceleration



Supply chain remediation time can be halved

Monitor Responsiveness KPIs



Mine the data to find bottlenecks and optimize teams and processes

Responsiveness Guidelines

Phase	CVSS version 3.1 Risk Service Level			
	Critical (CVSS 9-10)	High (CVSS 7-8.9)	Medium (CVSS 4-6.9)	Low (CVSS 0.1 -3.9)
Investigation				
Module Fix Plan				
Module Patch/Release				
CRB Fix Plan				
CRB Patch/Release				
Public Disclosure Date				

- Ask your upstream supplier to provide this table populated with the number they can achieve
- Provide this table to your customers to give them an expectation of how fast you can provide remediations

Detail Security Offering in SLAs

Once the organization's responsiveness is understood, you are in a position to

1. Offer Responsiveness Commitments
2. Offer Transparency – Access to KPIs

Downstream partners need this so they can generate their own responsiveness commitments for their downstream partners

What is Next?

- Proactive security – prevent vulnerabilities before they get into the released product

Everything Else…

- Craft processes that Product Engineering and Customer Engineering Teams can implement
- Provide and Support Tools engineering can use to implement processes
- Facilitate Compliance
  - Perform Process GAP analysis on engineering teams
  - Assist engineering teams in becoming compliant
- Train Engineering on process, tools, secure coding practices
- Train CISO org on how to determine compliance
- Operate in a decentralized manner

Questions?

References

1.) SSDF graphic - https://www.chainguard.dev/unchained/goodbye-sdlc-hello-ssdf-what-is-the-secure-software-development-framework

2.) EO 14028 - https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity

3.) SSDF to EO 14028 spreadsheet - https://csrc.nist.gov/files/pubs/sp/800/218/final/docs/nist.sp.800-218.ssdf-table.xlsx

4.) NIST SP 800-218 - https://doi.org/10.6028/NIST.SP.800-218

Thanks for attending the UEFI Fall 2023 Developers Conference & Plugfest

For more information on UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by