Embedded Linux Conference
Europe

OpenIoTSummit
Europe

Security Response Tool

David Reyna
Wind River Systems

david.reyna@windriver.com

THE LINUX FOUNDATION

Security Response Management

Risk, Cost, and Best Practices in an Imperfect World

- Keeping our products secure is a requirement for survival
- Security data is available, but can be a flood of data with varying quality and completeness
- Managing security defects can be very inefficient, resulting in high costs
- We need to share best practices, knowledge, awareness, automation, and **tools!**

Agenda (for DevDay 2019)

- **What this presentation is about**
  - Managing the response to the exponentially growing stream of potential security vulnerabilities in our upstream open source content
  - Maintaining the trust between our customers and our products
  - Introducing the new open source **Security Response Tool!**

- **What this presentation is not about**
  - Fixing CVEs
  - Limitations of upstream CVE databases, the changing nature of vulnerabilities (though IoT is trending)
  - CVE Scanners (static, build)

CVEs

- CVE (Common Vulnerability Enumerations)
  - The enumerations of the community tracked security vulnerabilities, separated by the year reported (e.g. CVE-2018-12345)
- Vendors/Sources
  - MITRE: Manages the list of CVEs
  - NIST (National Institute of Standards and Technology): manages the National Vulnerability Database (NVD) of CVEs
  - Hardware Vendors, Software Maintainers, Distros
    - Many vendors track and share CVE's relevant to their product
    - Many CVE aggregators also available (e.g. cvedetails.com)
  - Mailing lists, websites, and forums (public and private)
    - Preview of coming issues, place to discuss issues

General Security Patch Workflow

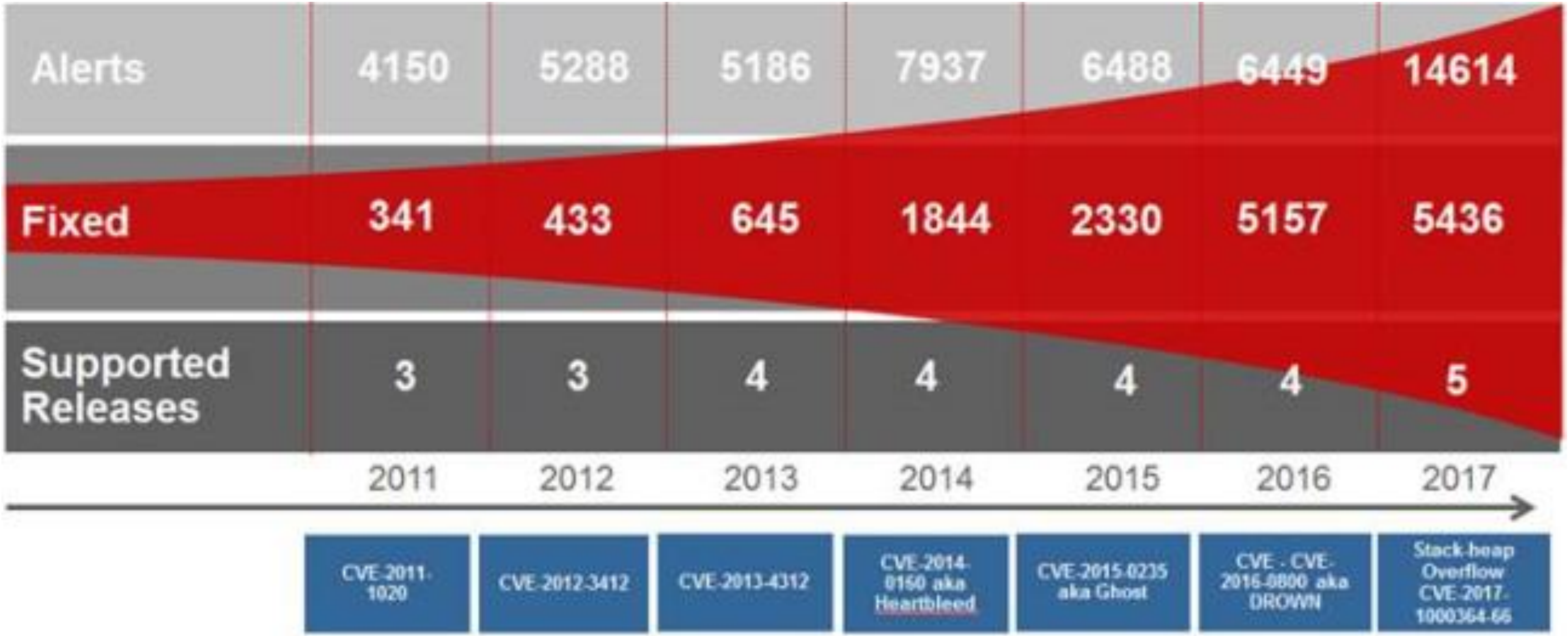
- Upstream CVE Sources
  - Gather data/fixes/info
  - Publish CVE Data
- You (OS Vendor/OEM/etc.)
  - Scan upstream CVEs
  - Manage CVE response
  - Fix CVEs
  - Create patches
- Customer
  - Receive patches
  - Test/deploy

(Managed Workflow)

Volume of CVE Data: Issues

- Volume of CVEs is 1000+ per month and growing
- Every new CVE must be evaluated, even if only a percentage may be applicable
- Costly in sheer numbers and required analysis overhead given the quality limitations
- Incorrectly categorizing a vulnerability can be even more costly in customer escalations and trust

Volume of CVE Data: Example



	2011	2012	2013	2014	2015	2016	2017
Alerts	4150	5288	5186	7937	6488	6449	14614
Fixed	341	433	645	1844	2330	5157	5436
Supported Releases	3	3	4	4	4	4	5
	CVE-2011-1020	CVE-2012-3412	CVE-2013-4312	CVE-2014-0160 aka Heartbleed	CVE-2015-0235 aka Ghost	CVE - CVE-2016-0800 aka DROWN	Stack-heap Overflow CVE-2017-1000364-66

Every CVE Needs to be Triaged

- You need to know what CVEs affect your product and customers
  - *Customer: “Am I affected?”*
- You also need to know what CVEs do not affect your product and customers
  - *Customer: “Are we not vulnerable, or did you miss that one?”*

Issues in CVE Triage

- CVEs may only have a brief or incomplete description
- CVE affected product list (CPEs) may have gaps, errors, unexpected version deviations, even be empty
- CVE content may be misleading, mentioning one package when it actually affects a different package
- CVEs may have few, inaccurate, or missing content links (discussion, reproducers, patches)
- CVE status changes continually as new information is discovered and shared
- Sometimes delays in content updates (dark CVEs)

Why System Analysis is Not Enough

- Can be very valuable in targeting product specific review activities
- Tells you of known vulnerabilities, but not what you are NOT vulnerable to
- Scans almost exclusively in the category of 'needs investigation‘
- Depends on known data
- *Example: Nessus*

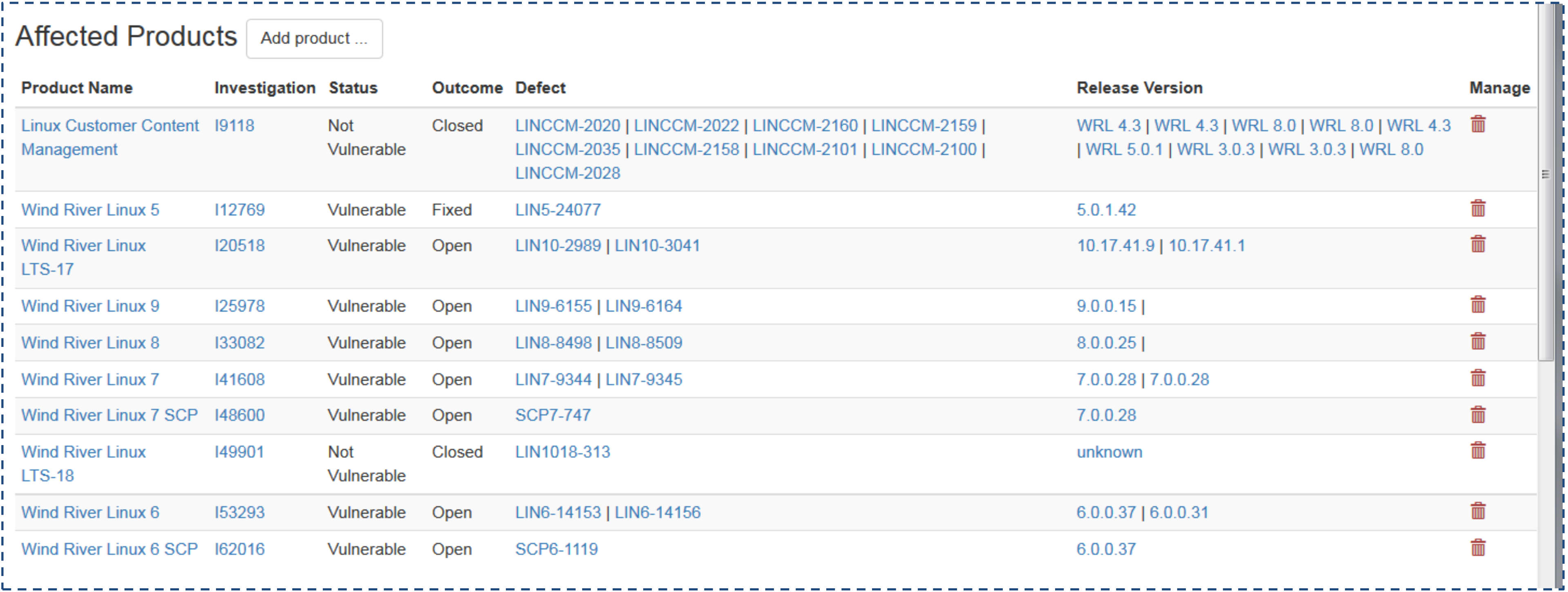
Goal of Security Response

- Automate as much of the process as possible
  - CVE data gathering, updating, change notifications
  - Defect update polling, with filtered change notifications
  - Report tools for management and customers
  - History and audit tracking
- Use multiple sources
  - NIST, MITRE, distros, oss-security, linux-distros (private list), …
- Aggregate the data
  - Central database, central document store

Introducing the SRTool

- Wind River has developed a tool called the “Security Response Tool” based on its cumulative experience
- Its goal is to address the process pain points and inefficiencies, to scale with a limited staff, and to implement best practices
- Wind River has shared this with open source via Yocto Project

SRTool: Vulnerability Page Example

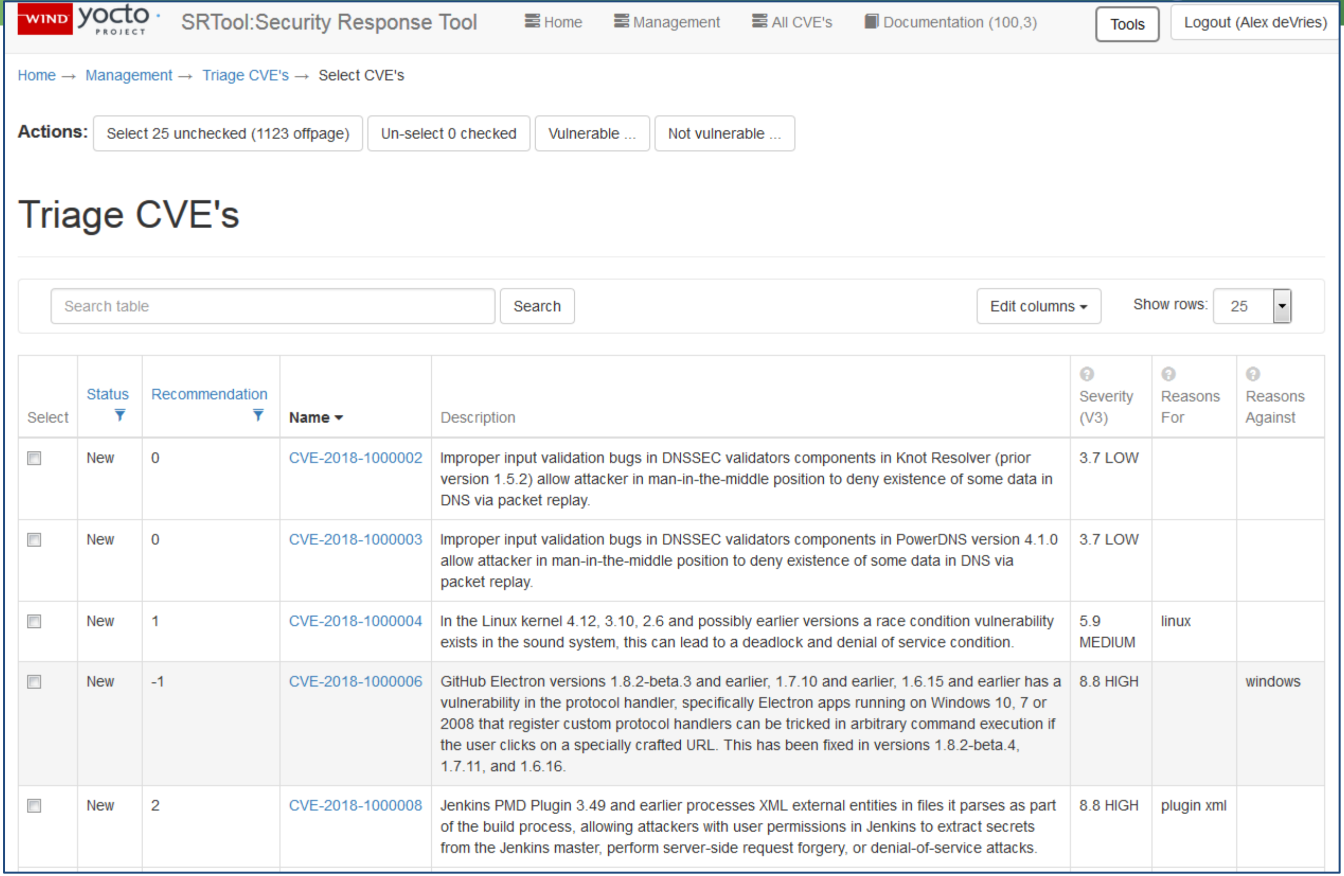


Affected Products

Add product ...

Product Name	Investigation	Status	Outcome	Defect	Release Version	Manage
Linux Customer Content Management	I9118	Not Vulnerable	Closed	LINCCM-2020 | LINCCM-2022 | LINCCM-2160 | LINCCM-2159 | LINCCM-2035 | LINCCM-2158 | LINCCM-2101 | LINCCM-2100 | LINCCM-2028	WRL 4.3 | WRL 4.3 | WRL 8.0 | WRL 8.0 | WRL 4.3 | WRL 5.0.1 | WRL 3.0.3 | WRL 3.0.3 | WRL 8.0	🗑
Wind River Linux 5	I12769	Vulnerable	Fixed	LIN5-24077	5.0.1.42	🗑
Wind River Linux LTS-17	I20518	Vulnerable	Open	LIN10-2989 | LIN10-3041	10.17.41.9 | 10.17.41.1	🗑
Wind River Linux 9	I25978	Vulnerable	Open	LIN9-6155 | LIN9-6164	9.0.0.15 |	🗑
Wind River Linux 8	I33082	Vulnerable	Open	LIN8-8498 | LIN8-8509	8.0.0.25 |	🗑
Wind River Linux 7	I41608	Vulnerable	Open	LIN7-9344 | LIN7-9345	7.0.0.28 | 7.0.0.28	🗑
Wind River Linux 7 SCP	I48600	Vulnerable	Open	SCP7-747	7.0.0.28	🗑
Wind River Linux LTS-18	I49901	Not Vulnerable	Closed	LIN1018-313	unknown	🗑
Wind River Linux 6	I53293	Vulnerable	Open	LIN6-14153 | LIN6-14156	6.0.0.37 | 6.0.0.31	🗑
Wind River Linux 6 SCP	I62016	Vulnerable	Open	SCP6-1119	6.0.0.37	🗑

SRTool: Guided Incoming CVE Triage



Home → Management → Triage CVE's → Select CVE's

Actions: Select 25 unchecked (1123 offpage) | Un-select 0 checked | Vulnerable ... | Not vulnerable ...

Triage CVE's

Select	Status	Recommendation	Name	Description	Severity (V3)	Reasons For	Reasons Against
☐	New	0	CVE-2018-1000002	Improper input validation bugs in DNSSEC validators components in Knot Resolver (prior version 1.5.2) allow attacker in man-in-the-middle position to deny existence of some data in DNS via packet replay.	3.7 LOW		
☐	New	0	CVE-2018-1000003	Improper input validation bugs in DNSSEC validators components in PowerDNS version 4.1.0 allow attacker in man-in-the-middle position to deny existence of some data in DNS via packet replay.	3.7 LOW		
☐	New	1	CVE-2018-1000004	In the Linux kernel 4.12, 3.10, 2.6 and possibly earlier versions a race condition vulnerability exists in the sound system, this can lead to a deadlock and denial of service condition.	5.9 MEDIUM	linux	
☐	New	-1	CVE-2018-1000006	GitHub Electron versions 1.8.2-beta.3 and earlier, 1.7.10 and earlier, 1.6.15 and earlier has a vulnerability in the protocol handler, specifically Electron apps running on Windows 10, 7 or 2008 that register custom protocol handlers can be tricked in arbitrary command execution if the user clicks on a specially crafted URL. This has been fixed in versions 1.8.2-beta.4, 1.7.11, and 1.6.16.	8.8 HIGH		windows
☐	New	2	CVE-2018-1000008	Jenkins PMD Plugin 3.49 and earlier processes XML external entities in files it parses as part of the build process, allowing attackers with user permissions in Jenkins to extract secrets from the Jenkins master, perform server-side request forgery, or denial-of-service attacks.	8.8 HIGH	plugin xml	

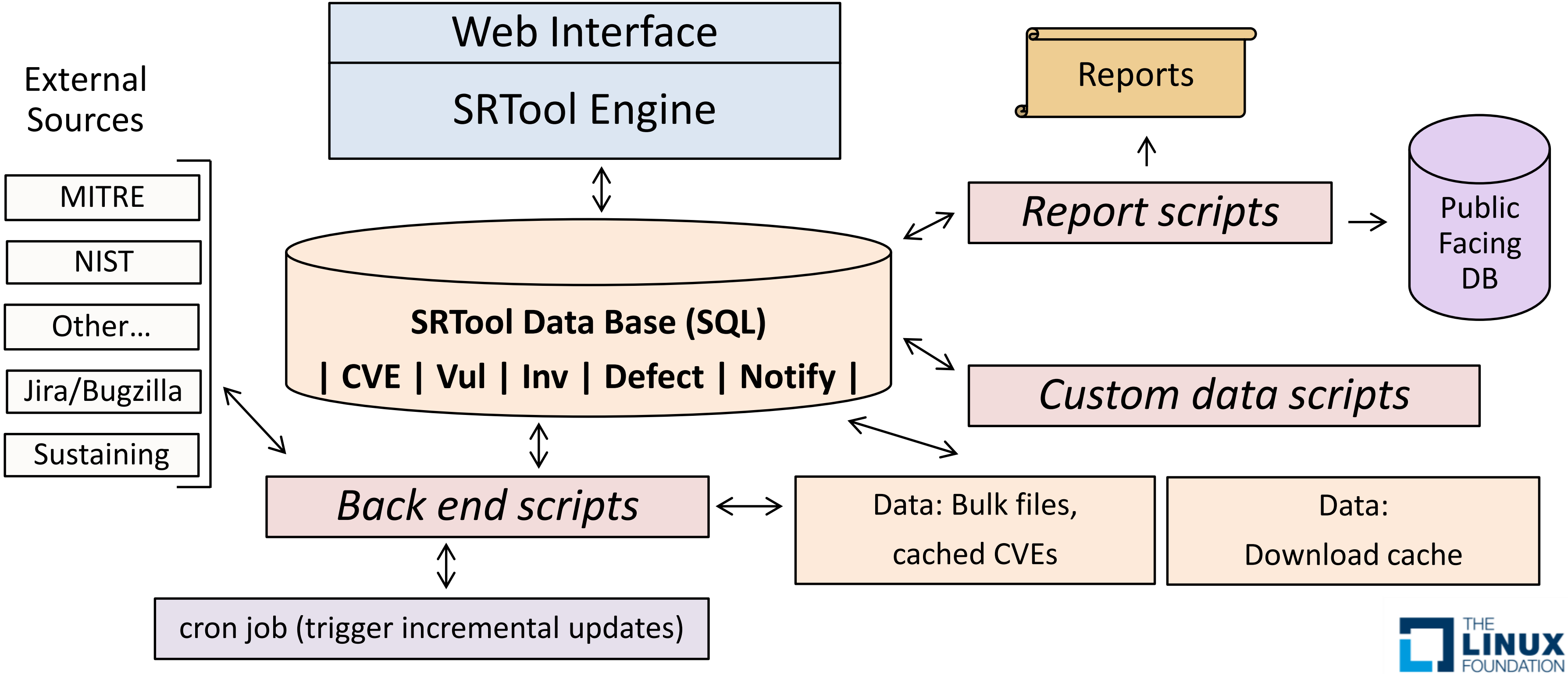
- CVE incoming rate 1000+ a month
- View for fast review and triage
- Heuristics from the previous defects to help guide the filtering process

Why not just use defect system

Defect systems are often poor security management systems

- Defects are per product, CVE's are across products
- An issue may need to be tracked before a CVE is created or published
- Hard to manage embargoed data in defect systems
  - Projects are normally public to entire product groups
  - Would require shadow projects
  - Would require a shadow project per authorized access list
- Awkward promoting private issues to public defects

SRTool: Functional Layout



How You Can Adopt The SRTool

- Clone the SRTool code base
- Automatically receive the upstream CVE data
- Use simple modular extensions to instantiate:
  - Your products
  - Your defect system integration (sample Jira integration available)
  - Your custom reports
  - Your business rules (e.g. public CVE publishing)
- See this link for details:
  - https://wiki.yoctoproject.org/wiki/Contribute_to_SRTool#Adapting_SRTool_to_your_Organization

Conclusion

- There is quite a wealth of vulnerability information available.
- With knowledge, awareness, adaptability, and automation, we can manage this struggle.
- We need to spend people’s time on the actual problems, not the process
- The SRTool community page is hosted here:
  - https://wiki.yoctoproject.org/wiki/Contribute_to_SRTool
- Use these links to learn more:
    - https://lists.yoctoproject.org/listinfo/yocto-security
    - david.reyna@windriver.com (SRTool maintainer)