OSADL: Safety Critical Linux Working Group

Project # 1: False positive management infrastructure

Rationale:

For all generic bug detection tools a serious issue are false positives - it is common in the current mainline tools (e.g. sparse, coccicheck, checkpatch.pl, smatch etc.) results to have almost pure false positives or cases that formally are correct positives but do not merit to get ever fixed by some reason (so called "won´t fix" cases [1]).

False positives have serious side effects at the process level:

1. After checking a few positives and finding that all of them are false, developers tend to consider all other findings as false positives and skip their inspection;
2. As the effort of running some tools (e.g. all Coccinelle scripts) is quite high, some developers may simply abandon using these tools at all.

The proposal of this project is to generate a public database of false-positive findings in order to make the current mainline tools easier to use and, thus, more attractive for developers.

Overview of development:

- Phase 1:	Requirements: • Generic tools and interface requirements (no performance requirements initially) • Mapping to standard and tool interfaces • Safety requirements (T2 tool)
- Phase 2:	Design - DB, GUI, I/O-layer (wrappers), inference plugin interface • Selection of adequate technologies • Investigation of trivial plugins (age, diff, file, obsolete) • Security concept for developer access user provided data
- Phase 3:	Implementation - DB setup, GUI prototype, I/O-layer examples plugin examples (blacklist, age, unchanged-pos, unchanged-ctx) validity management (trending - prediction) framework.

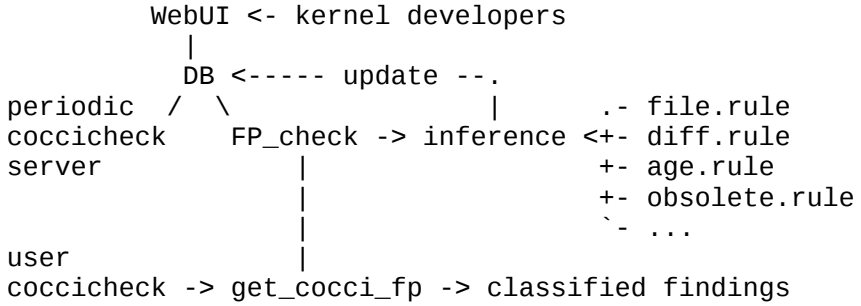

	Figure 1: preliminary community tool tracking architecture
- *Phase 4:*	Implement basic inference engine for trivial plugins based on likelihood functions/ Bayesian statistics: • Initial distribution assumptions (or uninformed distributions) • Criteria for update and review trigger conditions • Criteria for verification of inference engine • Malicious patch design (with internal knowledge) and assessment of risk+threat
- *Phase 5:*	Process specification and verification (records) • Manual verification of classifications (SRS) • Bounded false negatives (basically by SRS) • Periodic re-verification after accumulation of changes (updates and prediction distribution evolution)
- *Phase 6:*	Testing (Coccinelle + sparse) initial candidates for commonly used tools, testing with a full kernel cycle 4.X-> 4.X+1-rc ->4.X+1.{1-7} over a typical non-LTS cycle: • Verification with out-of-tree Coccinelle patches • Submission of test results to kernel developers for review • Selection of further candidate tools to integrate
- *Phase 7:*	Setup publicly accessible server and provide continuous data on findings for common kernel tools: • Feedback and incident reporting facilities • Development and user mailing lists
- *Phase 8:*	Formal acceptance for use in safety related systems: • Formalized Acceptance and Test Criteria (ATC) • Assessment Report (AR) • Presentation at TueV Rheinland • Formal submission for acceptance in the context of Annex A/B. Notably 61508-3 A.8 and A.9 (via Annex-QR) as well as formal selection criteria (61508-1 7.X Selection extension)
- *Phase 9:*	Dissemination and community integration (time-line of this phase would need to go beyond project end, e.g. Linux Janitor etc.) • OSDAL/LF communication channels to industry • Safety community conferences • FLOSS/Linux community events and community channels (LWN)
Possible extensions:	Email notification of commit authors for findings assumed to be true positives (e.g. new)
Participation form:	

	• Project development and data repository • Project mailing lists (development, review) • Dissemination workshops (may be in context of other events)
Responsibilities:	
	• Project legal entity: OSADL • Project resource maintenance OSADL Safety Critical Linux WG • Project management: OSADL • Technical lead: Markus Kreidl, Nicholas Mc Guire • Certification body: TueV Rheinland (proposed)
Effort estimation:	
	TO DO
Timeline:	
Start:	March 1, 2019
Milestone 1	Requirements and Design workshop
Milestone 2	Prototype presentation and discussion
Milestone 3	Initial release version (public/community)
Milestone 4	CA Report and all related artifacts published maintenance phase
Project format:	
	Partner funded project (OSADL members)