



SCALE Features + Beyond: Detail and Demo

Lori Flynn
Senior Software Security Researcher

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Copyright 2018 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® and CERT® are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
DM18-1345

SCALE Static Analysis Alert Auditing Tool

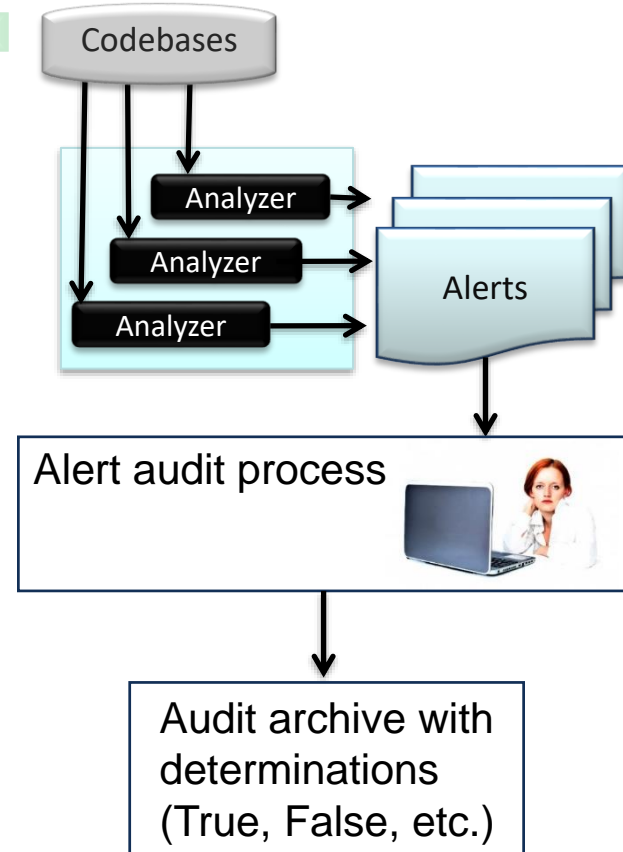
“alert audit” after run SA tool. Can be done at any point in software development lifecycle

Static analysis (SA) tools examine code without executing it

- Flaw-finding SA tools examine syntax, control flow, data flow, and/or type flow for indicators of particular flaws

SEI CERT’s SCALE tool:

- Developed by CERT Secure Coding team since 2010
 - Add new features to enable research
 - Auditors (collaborators & CERT) test new features
- Imports source code plus raw output from SA tools
- Provides GUI to audit alerts and view related code
- Stores audit archive data to exportable database



Static Analysis Tool Output Fully Integrated in SCALe

Any tool output can be integrated into SCALe with standard steps:

- Parse output to required format
- Provide tool checker mappings file (to CERT rules / CWEs)
- Small edits to a few files

Tools with outputs already integrated in SCALe:

(particular versions)

1. GCC (C, C++)
2. G++ (C++)
3. Microsoft Visual C++ compiler (C++)
4. Rosecheckers (C, C++)
5. Coverity Prevent (C, C++, Java)
6. Fortify (C, C++, Java)
7. Cppcheck (C, C++)
8. MSVC's Static Analyzer (C, C++)
9. PC-lint (C, C++)
10. Fortify (C, C++, Java)
11. LDRA (C, C++)
12. Eclipse (Java)
13. FindBugs (Java)
14. Perl Critic (Perl)
15. B Lint (Perl)
16. CCSM (C, C++)
17. Understand (C, C++, C#, Java, Python, ADA, [etc.](#))
18. Lizard (C, C++14, C#, Java, Python, JavaScript, [etc.](#))

Tool Versions

Software	Version	License	C	C++	Java	Perl	Windows	Linux
CERT Rosecheckers*		CMU	Yes	Yes				Yes
PC-lint	9.0	Proprietary	Yes	Yes			Yes	
LDRA	9.4.3	Proprietary	Yes	Yes			Yes	
Coverity	7.6.1	Proprietary	Yes	Yes	Yes		Yes	Yes
Fortify SCA	6.10.0120	Proprietary	Yes	Yes	Yes		Yes	Yes
cppcheck	1.66	Open source	Yes	Yes			Yes	Yes
MS Code Analysis for C/C++ Warnings		Proprietary	Yes	Yes			Yes	
FindBugs™	3.0.1	Open source			Yes		Yes	Yes
Perl::Critic	1.118	Open source				Yes		Yes
B::Lint	1.20	Open source				Yes		Yes

Software	Version	License	C	C++	Java	Perl	Windows	Linux
Microsoft Visual C++	Ultimate 2013 12.0.3110 1.00 Update 4	Proprietary		Yes			Yes	
GCC	4.8.3.20140911	Open source	Yes				Yes	Yes
G++	4.8.3.20140911	Open source		Yes			Yes	Yes
Eclipse	Luna sr2 (4.4.2) Build id: 20150219-0600	Open source			Yes		Yes	Yes
Perl	5.16.3	Open source				Yes		Yes
Lizard		Open source	Yes	Yes	Yes			Yes
CCSM		Open source	Yes	Yes				Yes
Understand		Proprietary	Yes	Yes	Yes		Yes	

What Differentiates SCALe from other Static Analysis Tools? SCALe:

- Has uncommon or unique features that
 - Your org may find useful for audits ← Have confirmed a version with FIPS compliance that is working in a SIPR environment
 - Your org may want to try out, to:
 - ✓ consider adding to your own tools
 - ✓ kindly give us research data
- Is an alert aggregator (*like DHS SWAMP, unlike single static analysis tools*)
- Has an interface for marking audit determinations (*like many tools, unlike SWAMP*)
- Is a research prototype tool (*SWAMP and other open-source tools can be used this way, unlike commercial tools*)
 - CERT adds features for research projects
- Does not compete with other tools
 - Usability, performance, and architecture is typical of small research prototypes
 - Success for us is good research results. (If other tools add the features or take code, great.)

Current SCALe+ Features in this Presentation

- Our determination labels
- Alert fusion
- CWEs and CERT rules
- Advanced prioritization schemes
- Classifier schemes
- Determinations history
- Code metrics
- User field uploads
- Notes
- Hyperlinked checker
- Cascading determinations
- Filters
- (SCALe+) API for classification and advanced prioritization system

Key

Blue: uncommon or unique

Black: common in audit tools

Upcoming SCALe+ Features in this Presentation

- Updated archive sanitizer
- Integrated classification
- Prototype classification and advanced prioritization system:
 1. with SCALe as part of the system
 2. Any other static analysis tool can implement system APIs and be used instead of SCALe in the system

Key

Blue: uncommon or unique

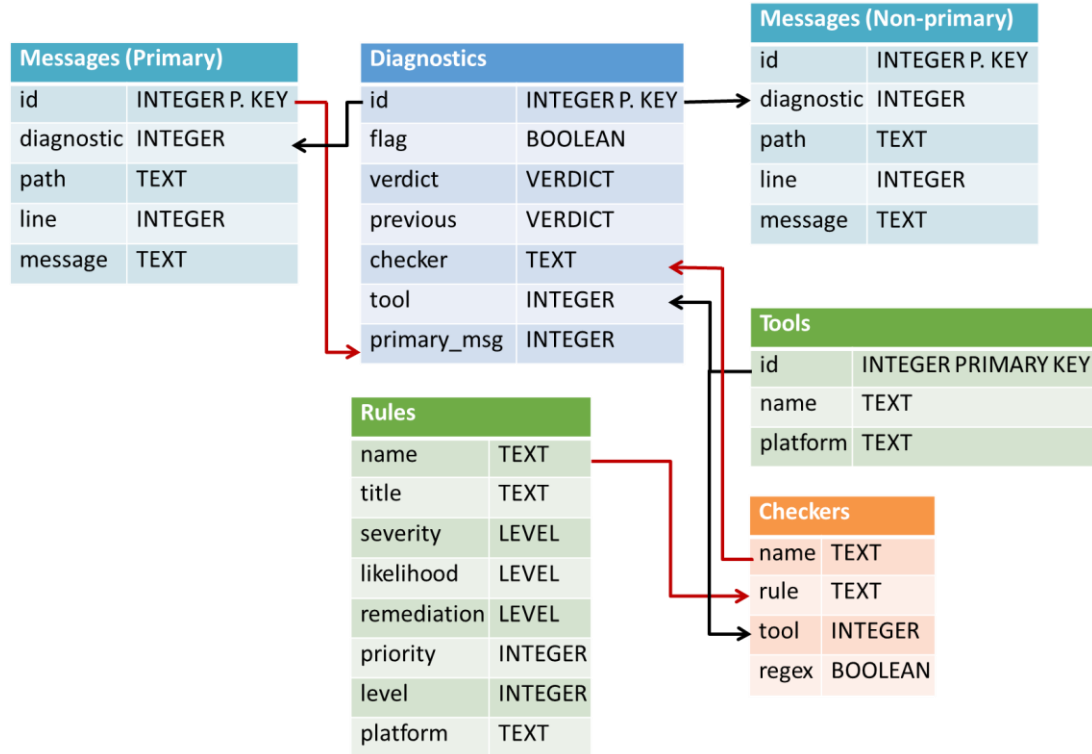
Black: common in audit tools

SCALE v1

Previously-released videos and technical reports only show SCALE v1

- First released outside SEI in 2015
- Enabled imports of 6 flaw-finding static analysis tool outputs
- Alert prioritization according to one metric (e.g., CERT rule 'severity' or 'priority')

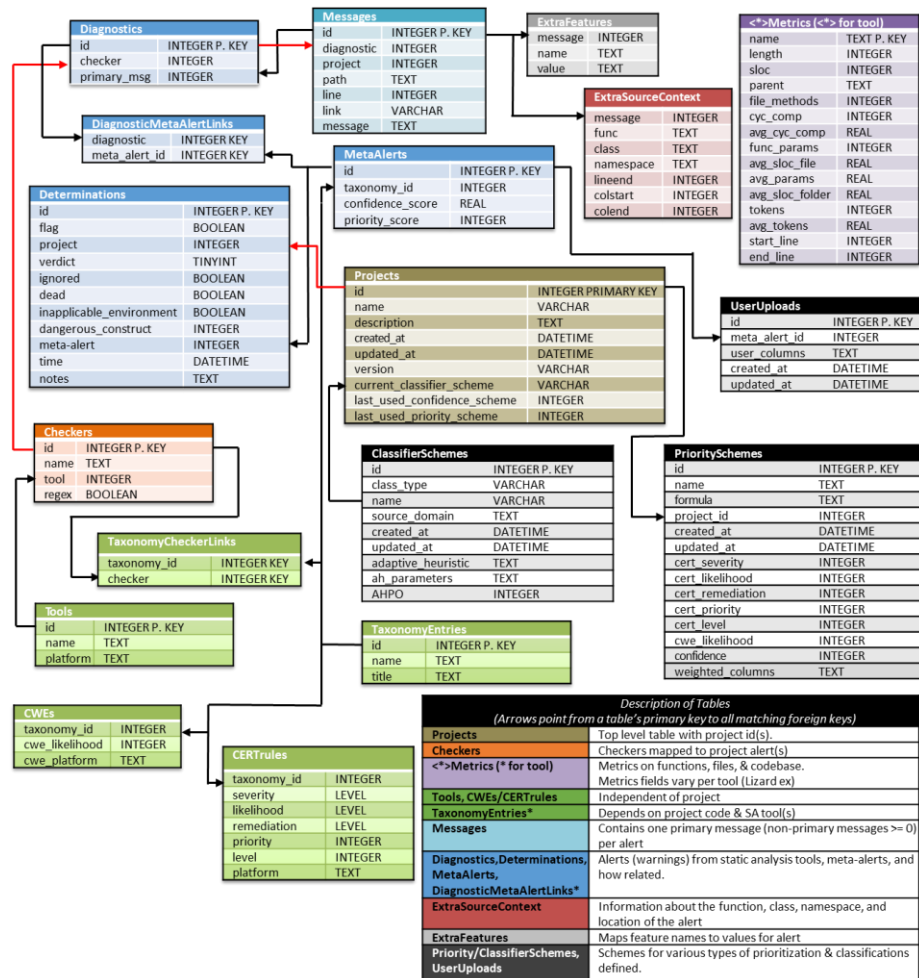
Exported Database Format



SCALE v3 Exported Database Format

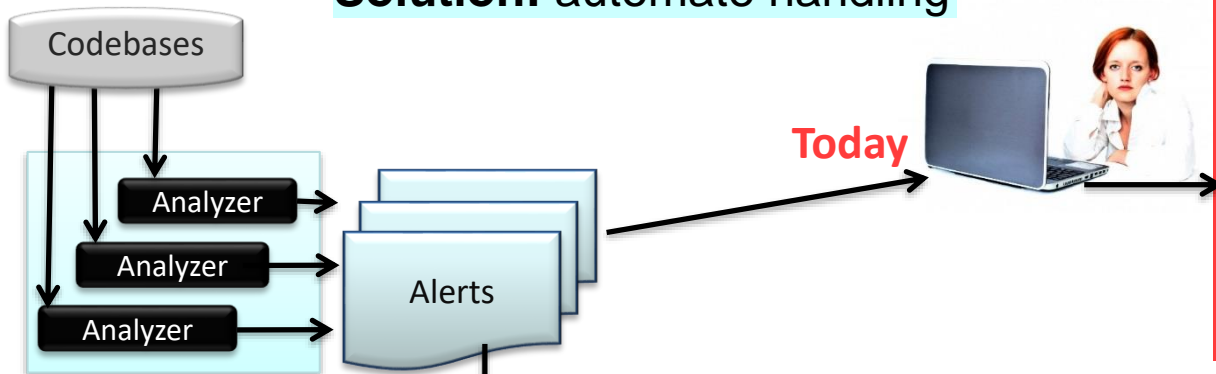
New data for:

- Machine learning classifiers
- Alert prioritization
- Data quality



Classifiers

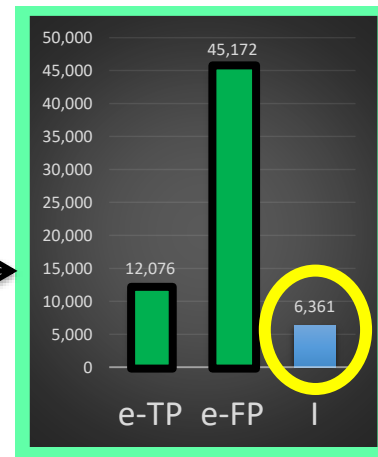
Problem: too many alerts
Solution: automate handling



Project Goal

System that automatically and **accurately** classifies most of the alerts as:

**Expected True Positive (e-TP) or
Expected False Positive (e-FP),
and
the rest as Indeterminate (I)**



SCALE Development

The screenshot shows the SCALE v2/v3 web interface. At the top, the header includes the project name 'Project dos2unix' and a 'New Diagnostic' button. Below this is a complex filter section with dropdowns for 'All Cvs', 'Verdict', 'Previous', 'Path', 'Line', 'Checker', 'Tool', 'Condition', and 'Taxonomy'. A 'Filter' button is located below these filters. The main area displays a table of diagnostics, with the first 10 shown. The table has columns for ID, Flag, Verdict, Supplemental, Notes, Previous, Path, Line, Message, Checker, Tool, Condition, Title, Confidence, Alert P/I, Sev, Lk, Rm, P/I, Lev, CWE, Lk. Below the table is a 'Source Code Viewer' showing the beginning of a C program named 'MAINS'.

ID	Flag	Verdict	Supplemental	Notes	Previous	Path	Line	Message	Checker	Tool	Condition	Title	Confidence	Alert P/I	Sev	Lk	Rm	P/I	Lev	CWE	Lk
1012 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	732	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	unusedAssignmentPAArg	opcheck	CWE-398	N/A	--	--							N/A
1013 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	772	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	unusedAssignmentPAArg	opcheck	CWE-398	N/A	--	--							N/A
1009 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	799	Condition 'TheWid' is always true	knownConditionTrueFalse	opcheck	CWE-570	N/A	--	--							N/A
1010 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	799	Condition 'TheWid' is always true	knownConditionTrueFalse	opcheck	CWE-571	N/A	--	--							N/A
1011 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	1038	Variable 'TheWid' is assigned a value that is never used.	unusedVariable	opcheck	CWE-563	N/A	--	--							N/A
1003 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	141	The scope of the variable 'mstr' can be reduced.	variableScope	opcheck	DCL19-C	Minimize the scope of variables and functions	--	--	1	1	2	2	3		
1005 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	199	The scope of the variable 'mstr' can be reduced.	variableScope	opcheck	DCL19-C	Minimize the scope of variables and functions	--	--	1	1	2	2	3		
1006 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	544	The scope of the variable 'berr' can be reduced.	variableScope	opcheck	DCL19-C	Minimize the scope of variables and functions	--	--	1	1	2	2	3		
1001 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	732	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	unusedAssignmentPAArg	opcheck	MISC12-C	Delete and remove code that has no effect	--	--	1	1	2	2	3		
1002 (6)	[]	[Unknown]	Est	0	0	/usrcommon.c	772	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	unusedAssignmentPAArg	opcheck	MISC12-C	Delete and remove code that has no effect	--	--	1	1	2	2	3		

Used as a research platform

- Extend with new features
- Collaborators give us feedback
- Collaborators generate data required for our classifier research

Over last 3 years, new SCALE features are for classification and prioritization research.

- GitHub public release (SCALE v2), Aug. 2018
- SCALE v3 for research project collaborators

SCALe v2 and v3 Development

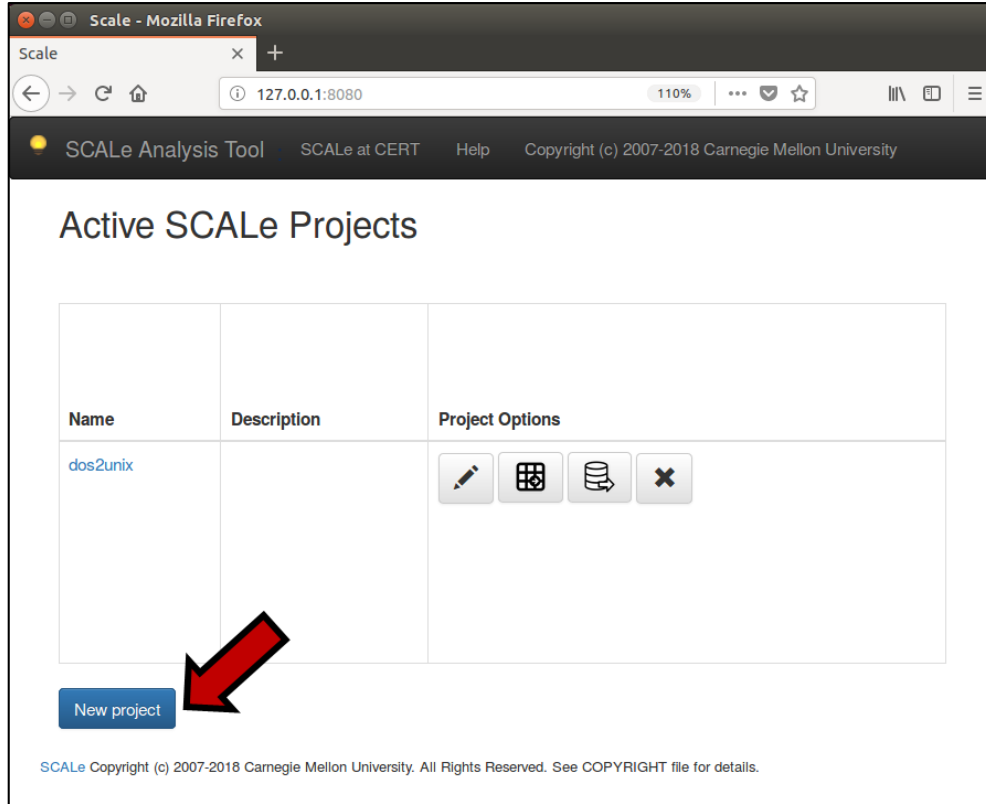
Since late 2015 to now, most SCALe development:

- Added features for classification and prioritization research
 - To provide new types of data for use by classifiers (e.g., as features)
 - To enhance quality of data used to develop classifiers
 - To enable outside organizations to share data with SEI
 - To enable selection of advanced prioritization and classifier schemes
- Done by developers on my research project teams. Including: Ebonie McNeil, David Svoboda, William Snavely, Derek Leung, Jiyeon Lee, Lucas Bengston, Jennifer Burns, Christine Baek, Baptiste Vauthy, Charisse Haruta, Shirley Zhou, Maria Rodriguez De La Cruz, and Elliot Toy.





SCALe v3 Features: Slides and Demos

First demo now

Project Creation



The screenshot shows a web browser window titled "Scale - Mozilla Firefox" with the address bar displaying "127.0.0.1:8080". The page header includes "SCALe Analysis Tool", "SCALe at CERT", "Help", and "Copyright (c) 2007-2018 Carnegie Mellon University". The main content area is titled "Active SCALe Projects" and contains a table with the following structure:

Name	Description	Project Options
dos2unix		   

Below the table, a blue button labeled "New project" is highlighted with a red arrow pointing to it.

SCALe Copyright (c) 2007-2018 Carnegie Mellon University. All Rights Reserved. See COPYRIGHT file for details.

Project Creation

Scale - Mozilla Firefox

Scale x +

127.0.0.1:8080/projects/new 110%

SCALe Analysis Tool SCALe at CERT Help Copyright (c) 2007-2018 Carnegie Mellon University

New project

Name:

Description:

Create Project

SCALe Copyright (c) 2007-2018 Carnegie Mellon University. All Rights Reserved. See COPYRIGHT file for details.

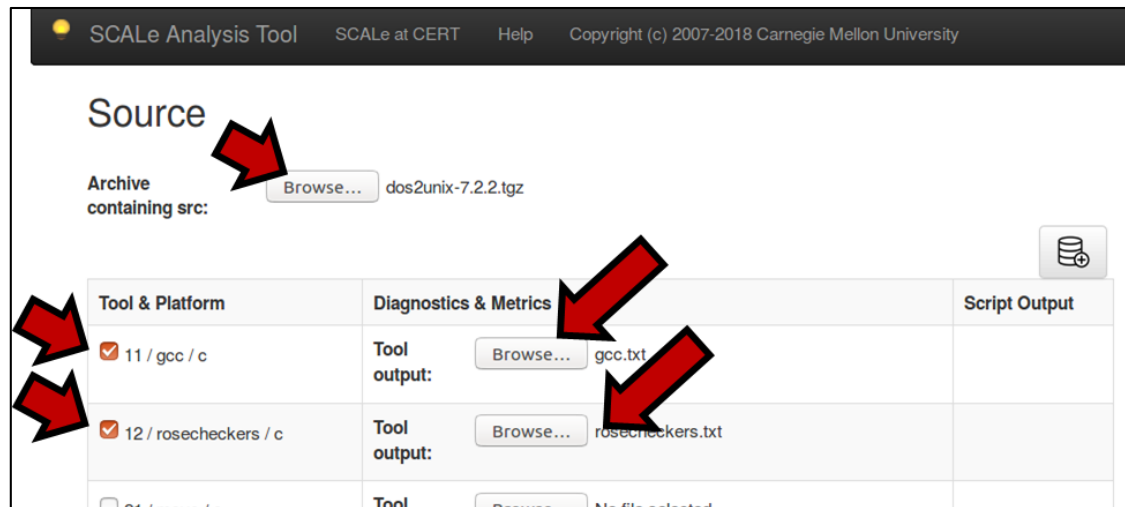
Uploading Source Code and Tool Output

SCALe Analysis Tool SCALe at CERT Help Copyright (c) 2007-2018 Carnegie Mellon University

Source

Archive containing src: dos2unix-7.2.2.tgz

Tool & Platform	Diagnostics & Metrics	Script Output
<input checked="" type="checkbox"/> 11 / gcc / c	Tool output: <input type="button" value="Browse..."/> gcc.txt	
<input checked="" type="checkbox"/> 12 / rosecheckers / c	Tool output: <input type="button" value="Browse..."/> rosecheckers.txt	
<input type="checkbox"/> 01 / gcc / c	Tool output: <input type="button" value="Browse..."/> No file selected	



Uploading Code Metrics Tool Output

<input type="checkbox"/> 91 / lizard / metric	Tool output: <input type="button" value="Browse..."/> No file selected.
<input checked="" type="checkbox"/> 92 / ccsm / metric	Tool output: <input type="button" value="Browse..."/> dos2unix_ccsm.c





Next, Create Project with Two Icon Selections: Icon #1

SCALe Analysis Tool SCALe at CERT Help Copyright (c) 2007-2018 Carnegie Mellon University

Source

Archive containing src: dos2unix-7.2.2.tgz

Tool & Platform	Diagnostics & Metrics	Script Output
<input type="checkbox"/> 11 / gcc / c	Tool output: <input type="button" value="Browse..."/> No file selected.	
<input checked="" type="checkbox"/> 12 / rosecheckers / c	Tool output: <input type="button" value="Browse..."/> rosecheckers.txt	

Next, Create Project with Two Icon Selections: Icon #2

SCALe Analysis Tool SCALe at CERT Help Copyright (c) 2007-2018 Carnegie Mellon University

Source

Source already uploaded

Create Project from Database













Tool & Platform	Diagnostics & Metrics	Script Output
<input type="checkbox"/> 11 / gcc / c	Tool output: <input type="button" value="Browse..."/> No file selected.	
<input checked="" type="checkbox"/> 12 / rosecheckers / c	Tool output already uploaded.	Success!

SCALE Homepage



SCALE Analysis Tool SCALE at CERT Help Copyright (c) 2007-2018 Carnegie Mellon University

Active SCALE Projects

Name	Description	Project Options
dos2unix		   
project2		   
projectCoffee		   

[New project](#)

SCALE Copyright (c) 2007-2018 Carnegie Mellon University. All Rights Reserved. See COPYRIGHT file for details.

Project: dos2unix

New Diagnostic

Fused View: On

Showing 1 to 10 of 253 | Diagnostics per page:


Set selected to:

Classifier Selected: None Selected

<input type="checkbox"/>	ID	Flag	Verdict	Supplemental	Notes	Previous	Path	Line	Message	Checker	Tool	Condition	Title	Confidence	Alert Pri	Sev	Lik	Rem	Pri	Lev	CWE_Lik		
<input type="checkbox"/>	1012 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	732	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	CWE-398	N/A	--	--							N/A	
<input type="checkbox"/>	1013 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	772	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	CWE-398	N/A	--	--								N/A
<input type="checkbox"/>	1009 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	799	Condition 'RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-570	N/A	--	--								N/A
<input type="checkbox"/>	1010 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	799	Condition 'RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-571	N/A	--	--								N/A
<input type="checkbox"/>	1011 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	1838	Variable 'RetVal' is assigned a value that is never used.	unreadVariable	cppcheck	CWE-563	N/A	--	--								N/A
<input type="checkbox"/>	1003 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	141	The scope of the variable 'errstr' can be reduced.	variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions	--	--	1	1	2	2	3			
<input type="checkbox"/>	1005 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	199	The scope of the variable 'errstr' can be reduced.	variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions	--	--	1	1	2	2	3			
<input type="checkbox"/>	1006 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	544	The scope of the variable 'bomr' can be reduced.	variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions	--	--	1	1	2	2	3			
<input type="checkbox"/>	1001 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	732	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	MSC12-C	Detect and remove code that has no effect	--	--	1	1	2	2	3			
<input type="checkbox"/>	1002 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	772	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	MSC12-C	Detect and remove code that has no effect	--	--	1	1	2	2	3			

SRC

Last updated Wed Aug 22 22:09:30 EDT 2018



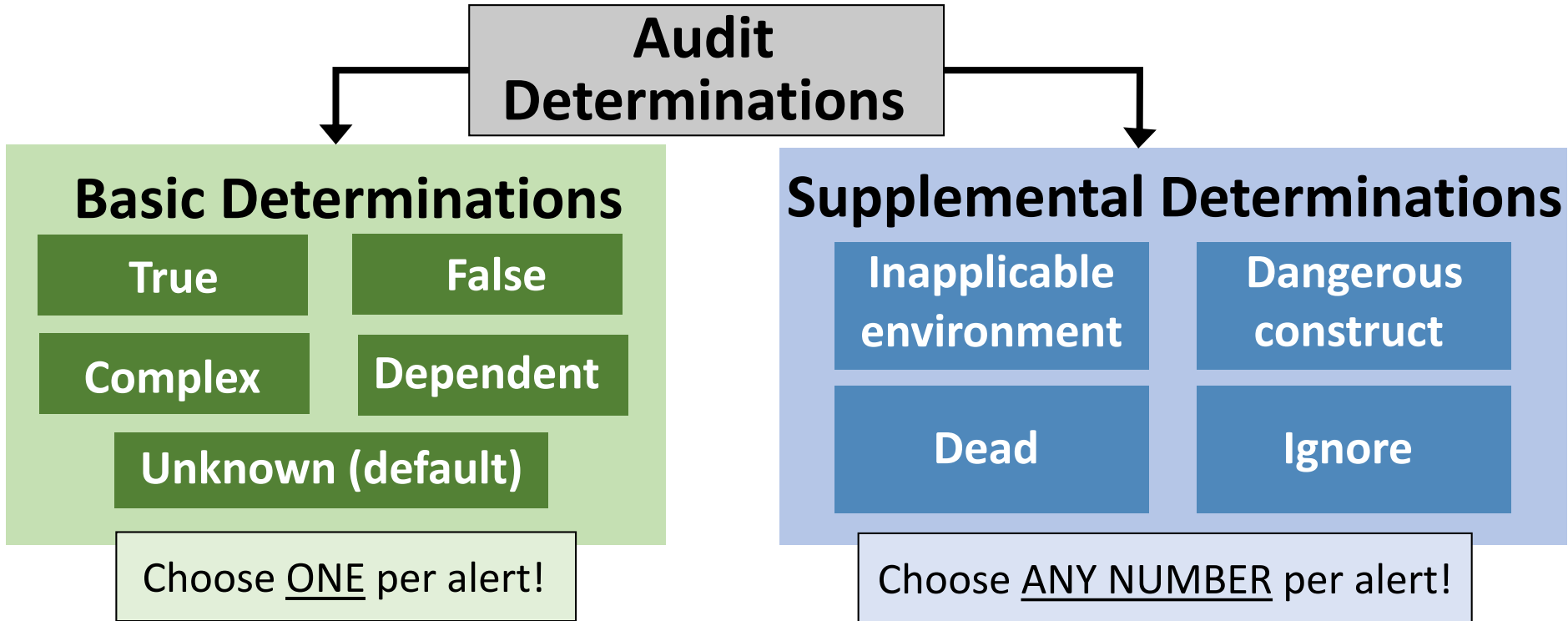
MAINS

```

448 src/dos2unix.c int main (int argc, char *argv[])
191 src/qlerycp.c int main() {
8 src/test/acstombs_test.c int main() {

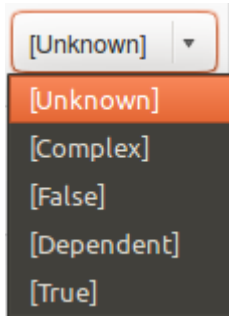
```

New Features: Audit Determinations



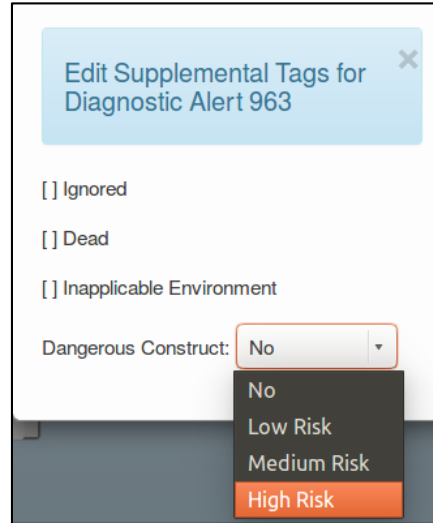
Determinations in GUI

Drop-down for primary verdict



Supplemental determination popup:

- select any number



Flag field can have org-defined meaning

<input type="checkbox"/>	ID	Flag	Verdict	Supplemental	Notes
<input type="checkbox"/>	963 (d)	[x]	[True]	Ignored Dangerous Construct - Med Edit	0
<input type="checkbox"/>	964 (d)	[]	[False]	Ignored Edit	var Y possible integer overflow
<input type="checkbox"/>	953 (d)	[]	[Unknown]	Edit	0

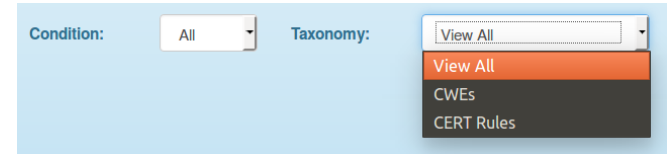
New Features: CWE Taxonomy Added

Tool checkers mapped to CWEs and CERT rules.




Checker	Tool	Condition	Title	Confidence	Alert Pri	Sev	Lik	Rem	Pri	Lev	CWE_Lik
uselessAssignmentPtrArg	cppcheck	CWE-398	N/A								N/A
uselessAssignmentPtrArg	cppcheck	CWE-398	N/A								N/A
knownConditionTrueFalse	cppcheck	CWE-570	N/A								N/A
knownConditionTrueFalse	cppcheck	CWE-571	N/A								N/A
unreadVariable	cppcheck	CWE-563	N/A								N/A
variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions			1	1	2	2	3	
variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions			1	1	2	2	3	
variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions			1	1	2	2	3	
uselessAssignmentPtrArg	cppcheck	MSC12-C	Detect and remove code that has no effect			1	1	2	2	3	
uselessAssignmentPtrArg	cppcheck	MSC12-C	Detect and remove code that has no effect			1	1	2	2	3	

- Some CWEs have CWE Likelihood.
- Can filter by CWE or CERT Rules taxonomy
- Can filter for single rule/CWE



New Feature: Notes

- Notes by auditor about determinations, alert, meta-alert, checker, condition, or language.
- The text can help later auditors reviewing same or similar issues.

dict	Supplemental	Notes
known]	Edit	Variable X may have integer overflow, must investigate 'else' conditional
known]	Edit	Variable Y appears to be handled safely. 
known]	Edit	0
known]	Edit	0

Prioritization Schemes

Prioritization schemes with mathematical formulas user can create and/or use



Create New Scheme

Name:

Instructions CWES CERT_RULES

cert_severity

cert_likelihood

cert_remediation

cert_priority

cert_level

confidence

Formula for CERT_RULES

Generate The Formula

Prioritization Formula:

Save Priority Priority Scheme Saved

Cancel Run Priority

User Field Uploads

User field uploads

- For advanced users that can work with SQLite databases and generate values
- Uploaded fields can be used in priority scheme
- CSV uploaded file
 - One line per project meta-alert ID
 - Left-most field has meta-alert ID
 - Top row holds field labels

```
meta_alert_id,safeguard_countermeasure,  
vulnerability,residual_risk,impact,  
threat,risk,complexity,severity,coupling  
112,5,1,4,9,1,1,5,5,1  
2,9,3,3,3,1,1,1,9,3  
3,3,1,1,1,8,1,5,5,1  
4,6,1,1,5,2,1,8,8,1  
5,2,1,1,2,3,1,7,7,5  
6,5,1,4,4,1,2,4,5,1  
7,8,5,3,4,8,2,4,9,9  
8,2,1,3,2,8,3,8,8,1  
9,6,4,3,6,9,1,4,4,4  
10,3,2,2,5,7,1,4,5,9  
11,6,1,1,9,6,1,7,7,1  
12,2,8,4,1,6,1,4,4,8
```

Demo

New Features: Cascade Determinations

Edit project

- Upload determinations from same tool on previous version of code
- Uses `diff` for line matches
- Match alert and line, then auto-cascade determination
- Caution: Data, control, and type flow changes may cause a previously-correct determination to change.

The screenshot shows the 'Edit project' interface. At the top, there is a 'Name:' field containing 'dos2unix_v3' and a 'Description:' text area. Below these is an 'Update Project' button. The interface is divided into three sections for uploading data:


- Upload SCALe Database:** Includes a 'Browse...' button, the text 'No file selected.', and an 'Upload SCALe database' button.
- Upload GNU Global Pages Archive (.zip or .tgz):** Includes a 'Browse...' button, the text 'No file selected.', and an 'Upload pages' button.
- Upload Determinations from Project:** Includes a dropdown menu currently showing 'dos2unix_v1' and an 'Upload determinations' button.

Three red arrows highlight specific features: one points to the 'Name' field, one points to the 'Upload Determinations from Project' section, and one points to the 'Upload determinations' button.

After Cascaded Import

After cascaded import

- Notes field show determination was cascaded
- Database records note about cascaded determination



<input type="checkbox"/>	ID	Flag	Verdict	Supplemental	Notes	Pre
<input type="checkbox"/>	721 (d)	[]	[False]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
<input type="checkbox"/>	719 (d)	[]	[True]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
<input type="checkbox"/>	720 (d)	[]	[True]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
<input type="checkbox"/>	734 (d)	[]	[Complex]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
<input type="checkbox"/>	735 (d)	[]	[Complex]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
<input type="checkbox"/>	736 (d)	[]	[Unknown]	Edit	0	0
<input type="checkbox"/>	737 (d)	[]	[Unknown]	Edit	0	0
<input type="checkbox"/>	738 (d)	[]	[Unknown]	Edit	0	0

Alert Fusion

- Alert fusion for {filepath, line, condition} reduces auditor effort
 - Multiple tools may indicate the same flaw
 - Make determination one time
 - See messages and insight about the flaw from all the tools at once

Screenshot shows fused (yellow) and unfused alerts.

- Fused alerts not expanded here (proprietary tools).


<input type="checkbox"/>	968 (d)	[]	[Unknown]	Edit	0	0	/src/dos2unix.c	358	Guarantee that array indices are within the valid range	ARR30-C	rosecheckers	ARR30-C	Do not form or use out-of-bounds pointers or array subscripts						3	3	1	9	2
<input type="checkbox"/>	277 (m)	[]	[Unknown]	Edit	0	0	/src/dos2unix.c	358				INT32-C	Ensure that operations on signed integers do not result in overflow						3	3	1	9	2
<input type="checkbox"/>	969 (d)	[]	[Unknown]	Edit	0	0	/src/dos2unix.c	393	Guarantee that array indices are within the valid range	ARR30-C	rosecheckers	ARR30-C	Do not form or use out-of-bounds pointers or array subscripts						3	3	1	9	2
<input type="checkbox"/>	281 (m)	[]	[Unknown]	Edit	0	0	/src/dos2unix.c	393				INT32-C	Ensure that operations on signed integers do not result in overflow						3	3	1	9	2
<input type="checkbox"/>	970 (d)	[]	[Unknown]	Edit	0	0	/src/unix2dos.c	357	Guarantee that array indices are within the valid range	ARR30-C	rosecheckers	ARR30-C	Do not form or use out-of-bounds pointers or array subscripts						3	3	1	9	2
<input type="checkbox"/>	285 (m)	[]	[Unknown]	Edit	0	0	/src/unix2dos.c	357				INT32-C	Ensure that operations on signed integers do not result in overflow						3	3	1	9	2
<input type="checkbox"/>	971 (d)	[]	[Unknown]	Edit	0	0	/src/unix2dos.c	390	Guarantee that array indices are within the valid range	ARR30-C	rosecheckers	ARR30-C	Do not form or use out-of-bounds pointers or array subscripts						3	3	1	9	2
<input type="checkbox"/>	289 (m)	[]	[Unknown]	Edit	0	0	/src/unix2dos.c	390				INT32-C	Ensure that operations on signed integers do not result in overflow						3	3	1	9	2

Demo


- Determination history
- Upload new tool output
- Hyperlinked checker

New Feature: Determination History

History kept of primary and supplemental determinations, notes, and flag



Flag	Verdict	Supplemental	Notes	Previous	Path	Line
[]	[True]	Dangerous - Med Edit	0	2	/src/common.c	809
[]	[Unknown]	Edit	0	0	/src/common.c	1090
[]	[Unknown]	Edit	0	0	/src/common.c	1606
[]	[Unknown]	Edit	0	0	/src/common.c	264
[]	[Unknown]	Edit	0	0	/src/common.c	289
[]	[Unknown]	Edit	0	0	/src/common.c	479




Additional Information for alert 483

Supplemental Messages

Message	Line	Path
Use typedefs to improve code readability	809	/src/common.c

Determination Log



Time	Flag	Verdict	Supplemental	Notes
2018-11-06 05:30:29 UTC	[]	[Unknown]		0
2018-11-06 05:57:29 UTC	[]	[True]		0
2018-11-06 05:57:43 UTC	[]	[True]	Dangerous Construct - Med	0


Hyperlinked Checker

Link to meta-alerts for that line, file, and checker

- May be multiple conditions (e.g, a CWE and a CERT rule)
- Helps auditor see related information, including related determinations


Select hyperlink to see list

Line	Message	Checker	Tool	Condition
732	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	CWE-398
772	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	CWE-398
799	Condition 'RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-570



All meta-alerts for checker + location

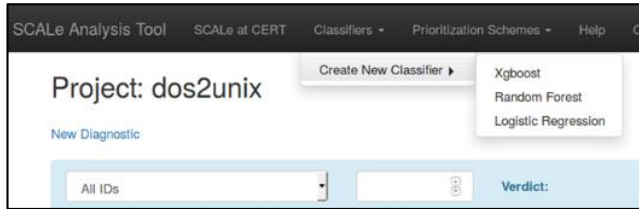
Path	Line	Message	Checker	Tool	Condition
/src/common.c	799	Condition 'RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-570
/src/common.c	799	Condition 'RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-571
/src/common.c	799	Condition 'RetVal' is always true	knownConditionTrueFalse	cppcheck	MSC07-C



Demo

Classification scheme

Classification Scheme

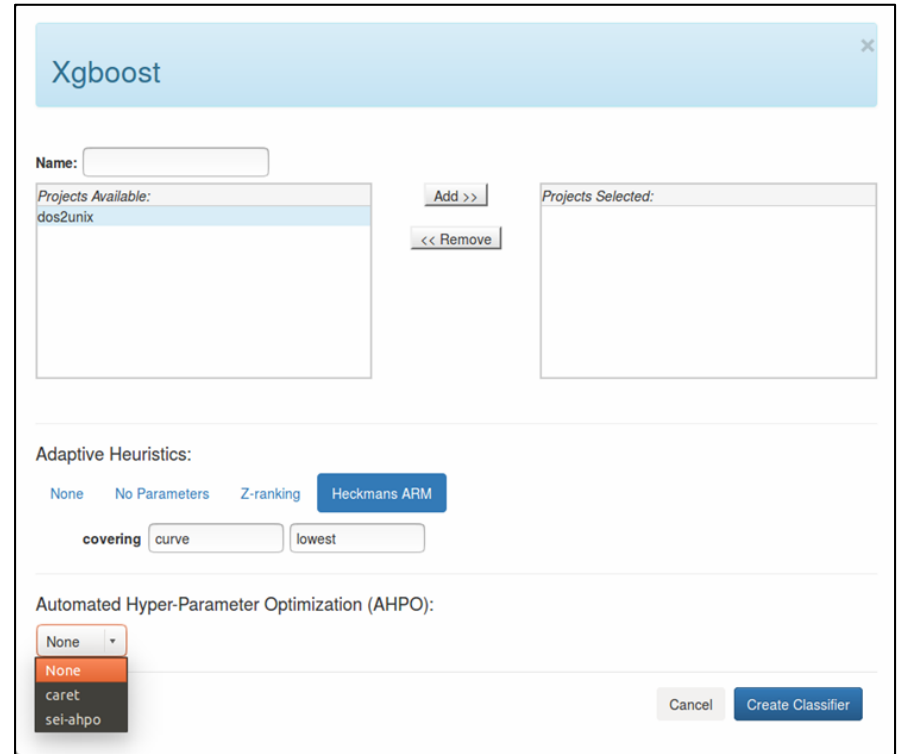


Select projects with audited alerts to develop classifier with

Select

- Type of classifier
- Type of adaptive heuristic
- Type automated hyper-parameter classification

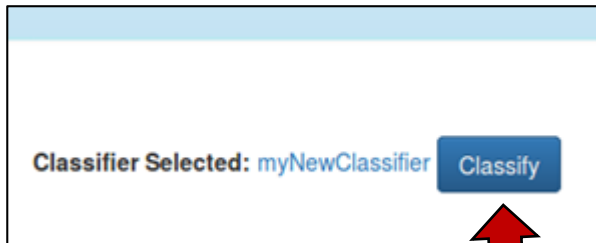
Then create the classifier



Run the Classifier on a Project

Select 'Classify' button to run the classifier on a project

- Classifier predicts alert determinations
- When fully functional, this will cause meta-alerts to be classified
- Currently, example metrics are loaded for the 'Confidence' field
 - Usability demonstration only
 - Values not currently from classifier



	Confidence	Alert Pri	Sev	Lik
	4.85			
	30.28			
	91.08			
	84.84			
	1.68			
hs	91.91		1	1
hs	83.26		1	1
hs	83.48		1	1
	15.27		1	1
	33.12		1	1

New Feature: Archive Sanitizer

Added data sanitizer script

- Anonymizes sensitive fields
- SHA-256 hash with salt
- Enables analysis of features correlated with alert confidence

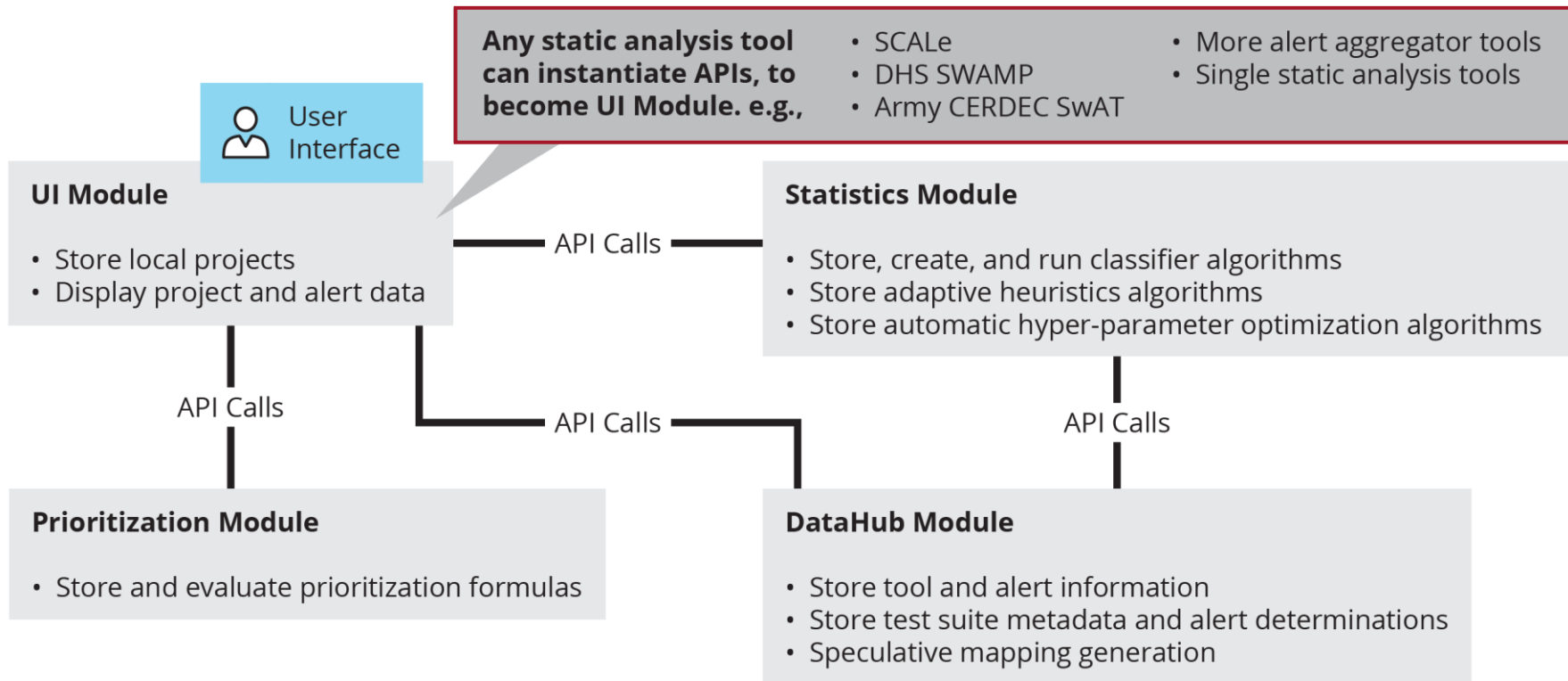
```
lflynn@ubuntu: ~/scale/scale.app/scripts
lflynn@ubuntu:~/scale/scale.app/scripts$ python sanitize_db.py dos2unix_
v1-2018-11-05_23_39_49.sqlite3
Creating database with added salt and sanitized path...
Creating sanitized database...
```

Audit archive for project is in a database

- DB fields may contain sensitive information
- Sanitizing script anonymizes or discards fields
 - Diagnostic message
 - Path, including directories and filename
 - Function name
 - Class name
 - Namespace/package
 - Project filename

Caution: GitHub sanitizer not fully updated for SCALE v2 database – don't count on it.

Architecture: API (β) Released, Prototype in Development



Architecture Development

Representational State Transfer (REST)

- Architectural style that defines a set of constraints and properties based on HTTP
- RESTful web services provide interoperability between systems
- Client-server

We chose to develop a RESTful API

- Swagger/OpenAPI open-source development toolset
 - Develop APIs
 - Auto-generate code for server stubs and clients
 - Test server controllers with GUI
 - Wide use (10,000 downloads/day)

SCALe Development for Architecture Integration

SCALe will make UI Module API calls in prototype system.

- Other alert auditing tools (e.g., DHS SWAMP) also can instantiate UI Module API.

Next Steps and Collaboration Opportunities

Code development to complete 4-server system instantiation with SCALe as UI Module

- Collaboration opportunities:
 - Implementation of API by collaborators to extend their own alert auditing tools
 - Feedback on API, code system, and adaptive heuristics
 - Alert audit data needed (sanitized fine)
 - **Additional ideas welcome!**

References

- SEI Technical Report “Integration of Automated Static Analysis Alert Classification and Prioritization with Auditing Tools: Focus on SCALe” (Publication expected November 2018)
- Presentation [Automating Static Analysis Alert Handling with Machine Learning: 2016-2018](#) (Oct. 2018)
- SEI blog post: “[SCALe: A Tool for Managing Output from Static Code Analyzers](#)” (Sep. 2018)
- SEI Podcast (video): “[Static Analysis Alert Classification with Test Suites](#)” (Sep. 2018)
- [GitHub SCALe v2 publication](#) (Aug. 2018)
- Paper “[Prioritizing Alerts from Multiple Static Analysis Tools, using Classification Models](#),” SQUADE ICSE workshop (June 2018)
- SEI blog post: “[Test Suites as a Source of Training Data for Static Analysis Alert Classifiers](#)” (Apr. 2018)
- Paper “[Static Analysis Alert Audits: Lexicon & Rules](#)”, IEEE Cybersecurity Development Conference (Nov 2016)

Contact Information

Lori Flynn

Senior Software Security Researcher

Telephone: +1 412.268.7886

Email: lflynn@sei.cmu.edu