SCALe Features + Beyond:
Detail and Demo

Lori Flynn
Senior Software Security Researcher

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

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.

SCALe Static Analysis Alert Auditing Tool

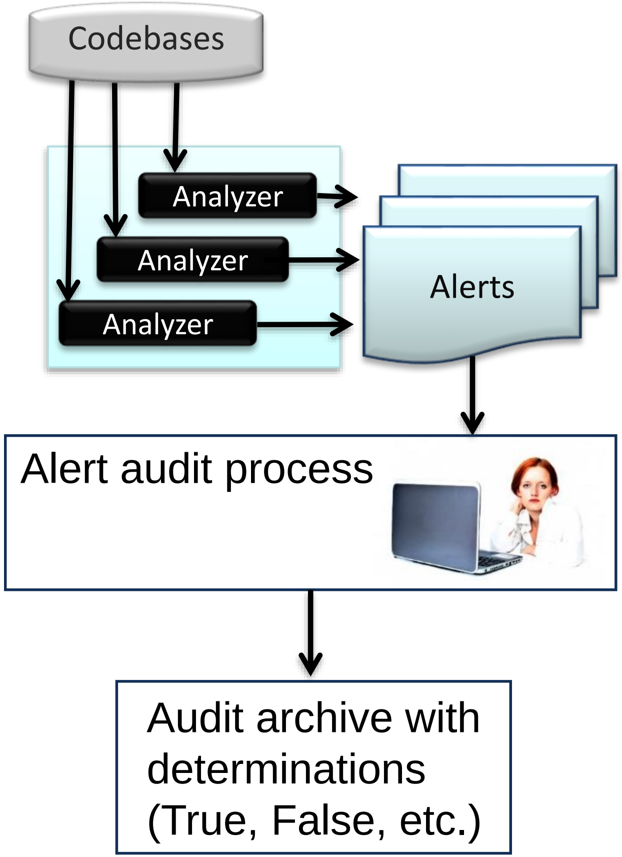
“alert audit” after run SA tool. Can be done at any point in software development lifecycle

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 12.0.31101.00 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
  - 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.)

Have confirmed a version with FIPS compliance that is working in a SIPR environment

Current SCALe+ Features in this Presentation

Key
Blue: uncommon or unique
Black: common in audit tools

- 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

Upcoming SCALe+ Features in this Presentation

Key

Blue: uncommon or unique

Black: common in audit tools

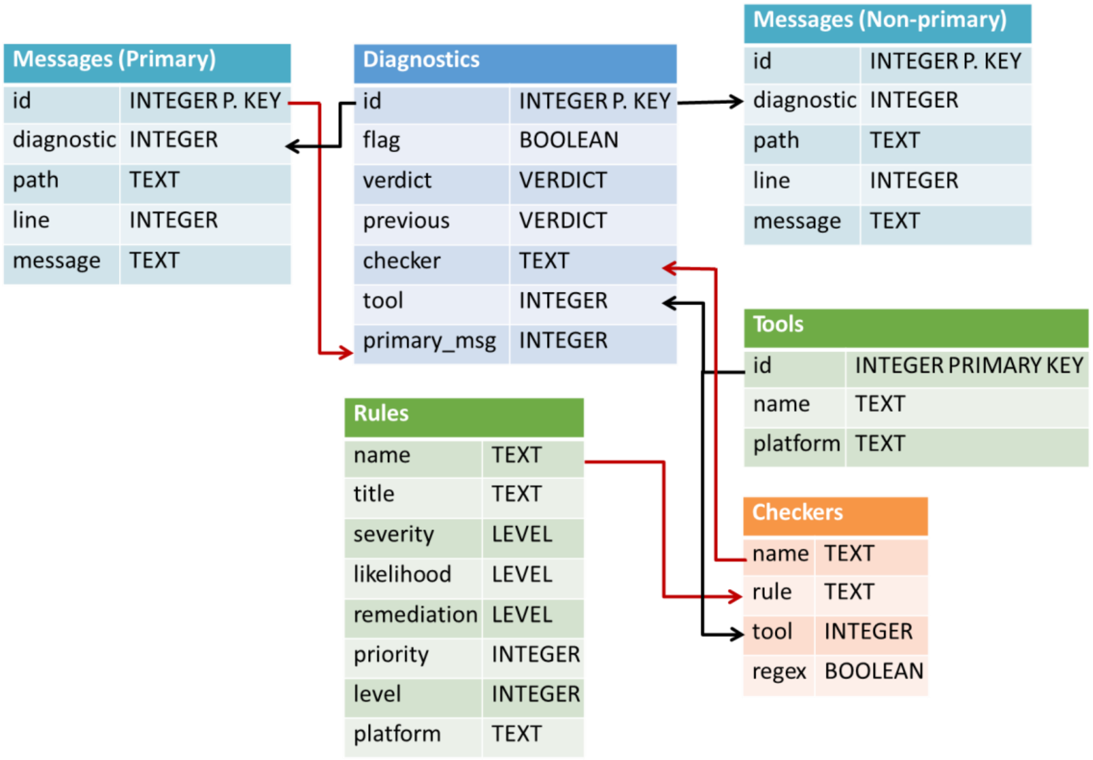
- Updated archive sanitizer
- Integrated classification
- Prototype classification and advanced prioritization system:
  1. with SCALe as part of the system
  2. Any other static analysis tool can implement system APIs and be used instead of SCALe in the system

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



Messages (Primary)	
id	INTEGER P. KEY
diagnostic	INTEGER
path	TEXT
line	INTEGER
message	TEXT

Diagnostics	
id	INTEGER P. KEY
flag	BOOLEAN
verdict	VERDICT
previous	VERDICT
checker	TEXT
tool	INTEGER
primary_msg	INTEGER

Messages (Non-primary)	
id	INTEGER P. KEY
diagnostic	INTEGER
path	TEXT
line	INTEGER
message	TEXT

Tools	
id	INTEGER PRIMARY KEY
name	TEXT
platform	TEXT

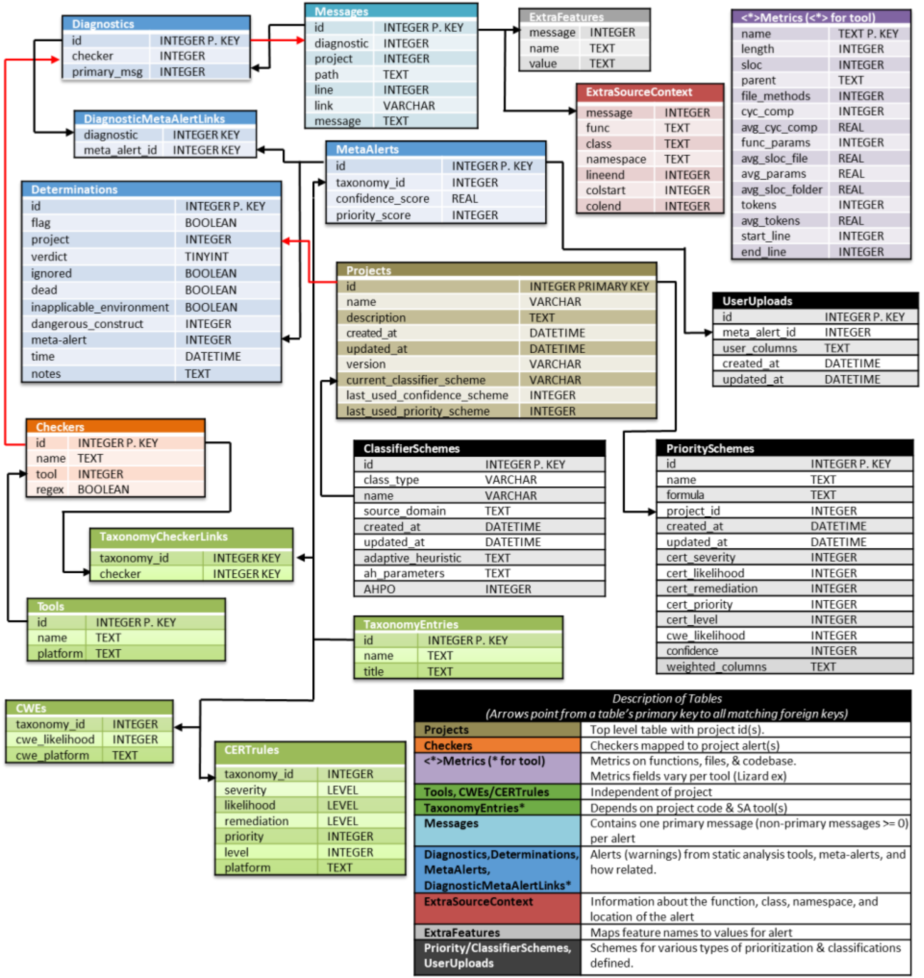
Rules	
name	TEXT
title	TEXT
severity	LEVEL
likelihood	LEVEL
remediation	LEVEL
priority	INTEGER
level	INTEGER
platform	TEXT

Checkers	
name	TEXT
rule	TEXT
tool	INTEGER
regex	BOOLEAN

SCALe v3 Exported Database Format

New data for:

- Machine learning classifiers
- Alert prioritization
- Data quality



Diagnostics	
id	INTEGER P. KEY
checker	INTEGER
primary_msg	INTEGER

Messages	
id	INTEGER P. KEY
diagnostic	INTEGER
project	INTEGER
path	TEXT
line	INTEGER
link	VARCHAR
message	TEXT

ExtraFeatures	
message	INTEGER
name	TEXT
value	TEXT

<*>Metrics (<*> for tool)	
name	TEXT P. KEY
length	INTEGER
sloc	INTEGER
parent	TEXT
file_methods	INTEGER
cyc_comp	INTEGER
avg_cyc_comp	REAL
func_params	INTEGER
avg_sloc_file	REAL
avg_params	REAL
avg_sloc_folder	REAL
tokens	INTEGER
avg_tokens	REAL
start_line	INTEGER
end_line	INTEGER

DiagnosticMetaAlertLinks	
diagnostic	INTEGER KEY
meta_alert_id	INTEGER KEY

ExtraSourceContext	
message	INTEGER
func	TEXT
class	TEXT
namespace	TEXT
lineend	INTEGER
colstart	INTEGER
colend	INTEGER

MetaAlerts	
id	INTEGER P. KEY
taxonomy_id	INTEGER
confidence_score	REAL
priority_score	INTEGER

Determinations	
id	INTEGER P. KEY
flag	BOOLEAN
project	INTEGER
verdict	TINYINT
ignored	BOOLEAN
dead	BOOLEAN
inapplicable_environment	BOOLEAN
dangerous_construct	INTEGER
meta-alert	INTEGER
time	DATETIME
notes	TEXT

Projects	
id	INTEGER PRIMARY KEY
name	VARCHAR
description	TEXT
created_at	DATETIME
updated_at	DATETIME
version	VARCHAR
current_classifier_scheme	VARCHAR
last_used_confidence_scheme	INTEGER
last_used_priority_scheme	INTEGER

UserUploads	
id	INTEGER P. KEY
meta_alert_id	INTEGER
user_columns	TEXT
created_at	DATETIME
updated_at	DATETIME

Checkers	
id	INTEGER P. KEY
name	TEXT
tool	INTEGER
regex	BOOLEAN

ClassifierSchemes	
id	INTEGER P. KEY
class_type	VARCHAR
name	VARCHAR
source_domain	TEXT
created_at	DATETIME
updated_at	DATETIME
adaptive_heuristic	TEXT
ah_parameters	TEXT
AHPO	INTEGER

PrioritySchemes	
id	INTEGER P. KEY
name	TEXT
formula	TEXT
project_id	INTEGER
created_at	DATETIME
updated_at	DATETIME
cert_severity	INTEGER
cert_likelihood	INTEGER
cert_remediation	INTEGER
cert_priority	INTEGER
cert_level	INTEGER
cwe_likelihood	INTEGER
confidence	INTEGER
weighted_columns	TEXT

TaxonomyCheckerLinks	
taxonomy_id	INTEGER KEY
checker	INTEGER KEY

Tools	
id	INTEGER P. KEY
name	TEXT
platform	TEXT

TaxonomyEntries	
id	INTEGER P. KEY
name	TEXT
title	TEXT

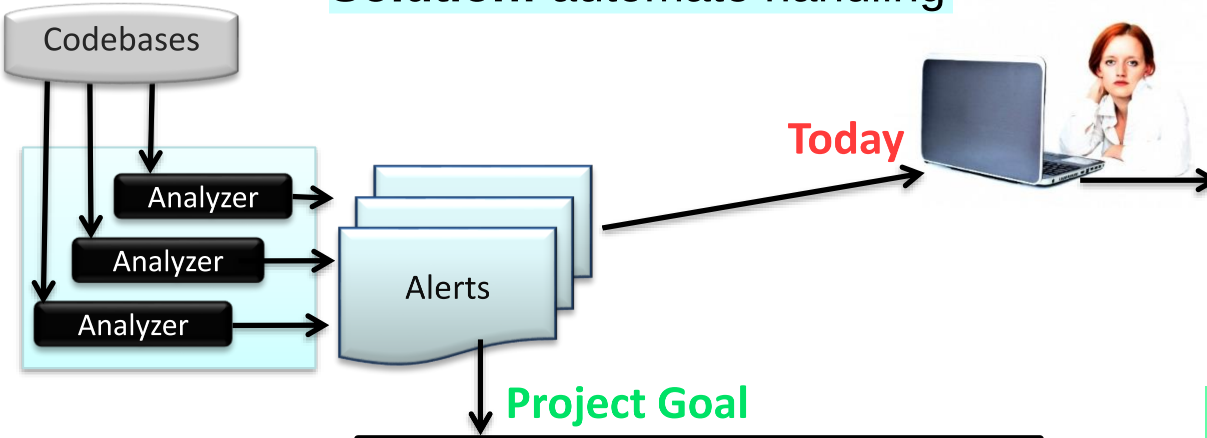
CWEs	
taxonomy_id	INTEGER
cwe_likelihood	INTEGER
cwe_platform	TEXT

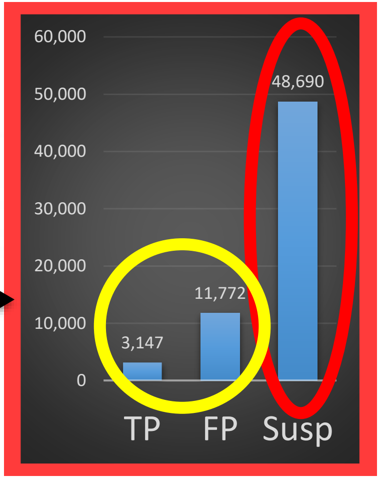
CERTrules	
taxonomy_id	INTEGER
severity	LEVEL
likelihood	LEVEL
remediation	LEVEL
priority	INTEGER
level	INTEGER
platform	TEXT

Description of Tables *(Arrows point from a table's primary key to all matching foreign keys)*	
Projects	Top level table with project id(s).
Checkers	Checkers mapped to project alert(s)
<*>Metrics (* for tool)	Metrics on functions, files, & codebase. Metrics fields vary per tool (Lizard ex)
Tools, CWEs/CERTrules	Independent of project
TaxonomyEntries*	Depends on project code & SA tool(s)
Messages	Contains one primary message (non-primary messages >= 0) per alert
Diagnostics,Determinations, MetaAlerts, DiagnosticMetaAlertLinks*	Alerts (warnings) from static analysis tools, meta-alerts, and how related.
ExtraSourceContext	Information about the function, class, namespace, and location of the alert
ExtraFeatures	Maps feature names to values for alert
Priority/ClassifierSchemes, UserUploads	Schemes for various types of prioritization & classifications defined.

Classifiers

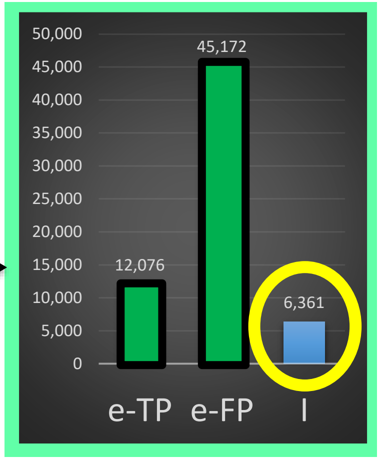
Problem: too many alerts
Solution: automate handling



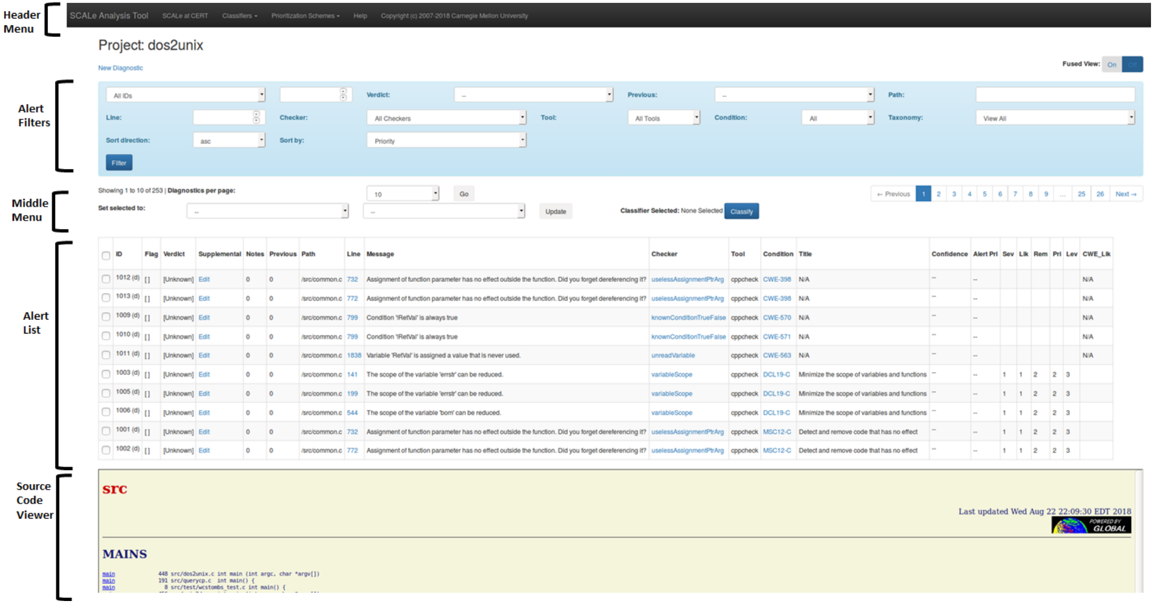


System that automatically and **accurately classifies most of the alerts as:**

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



SCALe Development



Project: dos2unix

New Diagnostic

Fused View: On Off

All IDs

Verdict: --

Previous: --

Path:

Line:

Checker: All Checkers

Tool: All Tools

Condition: All

Taxonomy: View All

Sort direction: asc

Sort by: Priority

Showing 1 to 10 of 253 | **Diagnostics per page:** 10

Set selected to: -- --

Classifier Selected: None Selected

☐	ID	Flag	Verdict	Supplemental	Notes	Previous	Path	Line	Message	Checker	Tool	Condition	Title	Confidence	Alert Pri	Sev	Lik	Rem	Pri	Lev	CWE_Lik
☐	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
☐	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
☐	1009 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	799	Condition '!RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-570	N/A	--	--						N/A
☐	1010 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	799	Condition '!RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-571	N/A	--	--						N/A
☐	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
☐	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	
☐	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	
☐	1006 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	544	The scope of the variable 'bom' can be reduced.	variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions	--	--	1	1	2	2	3	
☐	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	
☐	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

POWERED BY GLOBAL

MAINS

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

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

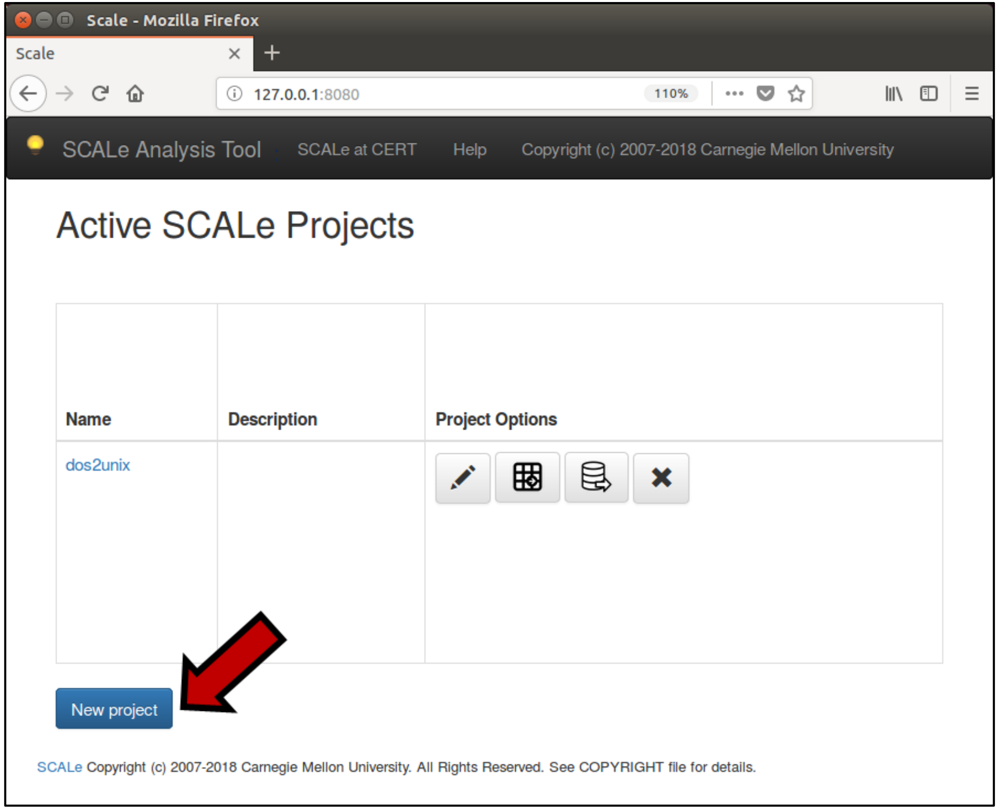
Since late 2015 to now, most SCALe development:

- Added features for classification and prioritization research
  - To provide <u>new types</u> of data for use by classifiers (e.g., as features)
  - To enhance <u>quality</u> 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

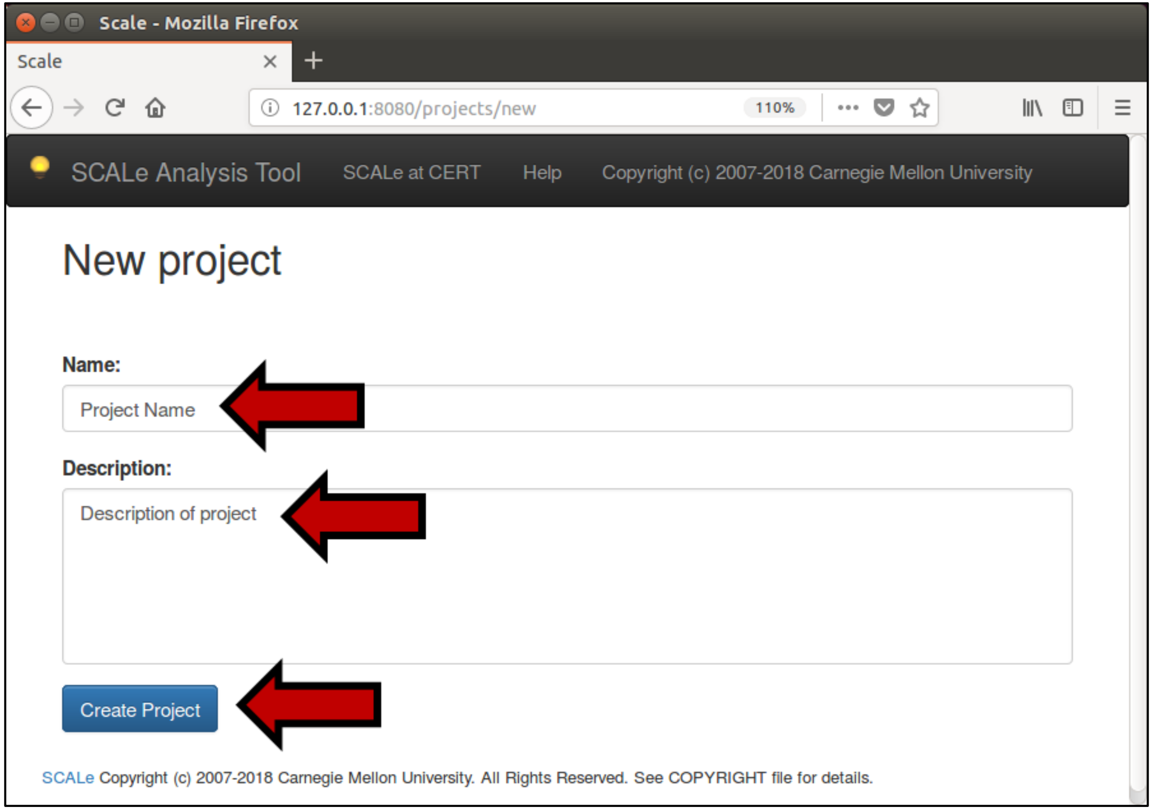


Copyright (c) 2007-2018 Carnegie Mellon University

Active SCALe Projects

Name	Description	Project Options
dos2unix		✎ ▦ ⛁ ✖

Project Creation



Scale - Mozilla Firefox
Scale
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:

Project Name

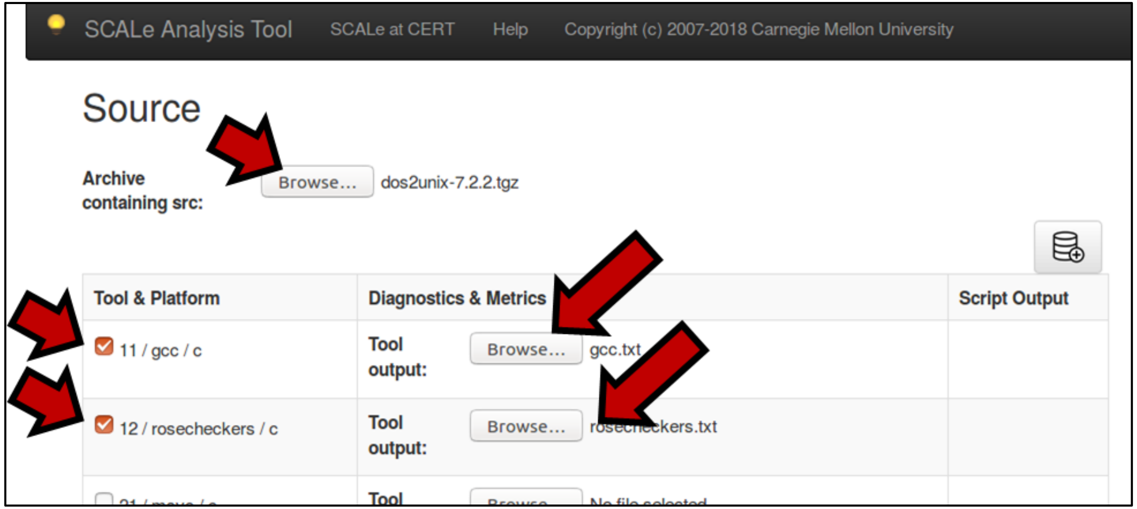
Description:

Description of project

Create Project

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

Uploading Source Code and Tool Output

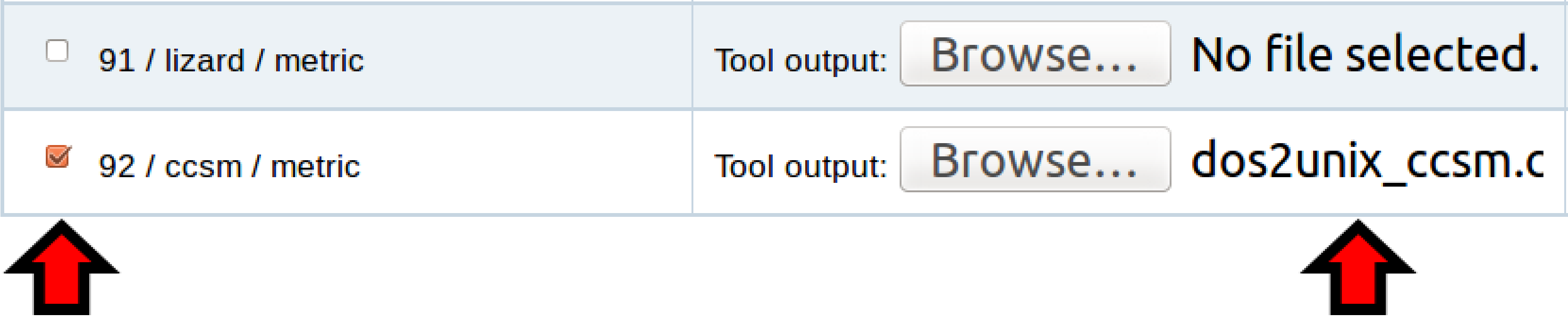


Source

Archive containing src: Browse... dos2unix-7.2.2.tgz

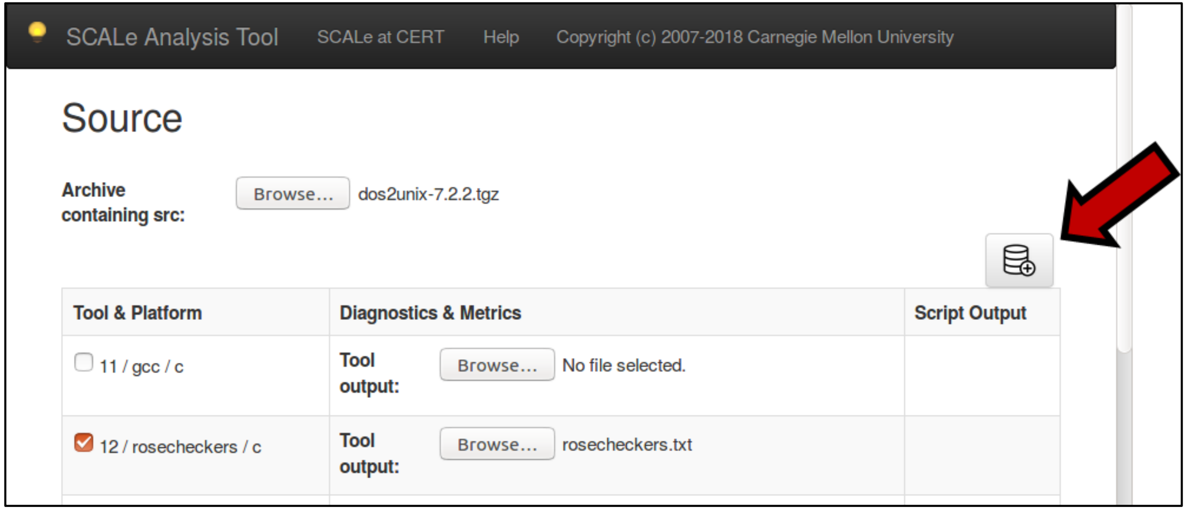
Tool & Platform	Diagnostics & Metrics	Script Output
☑ 11 / gcc / c	Tool output: Browse... gcc.txt	
☑ 12 / rosecheckers / c	Tool output: Browse... rosecheckers.txt	
☐ 21 / [illegible] / c	Tool Browse... No file selected	

Uploading Code Metrics Tool Output



☐	91 / lizard / metric	Tool output: Browse...	No file selected.
☑	92 / ccsm / metric	Tool output: Browse...	dos2unix_ccsm.c

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



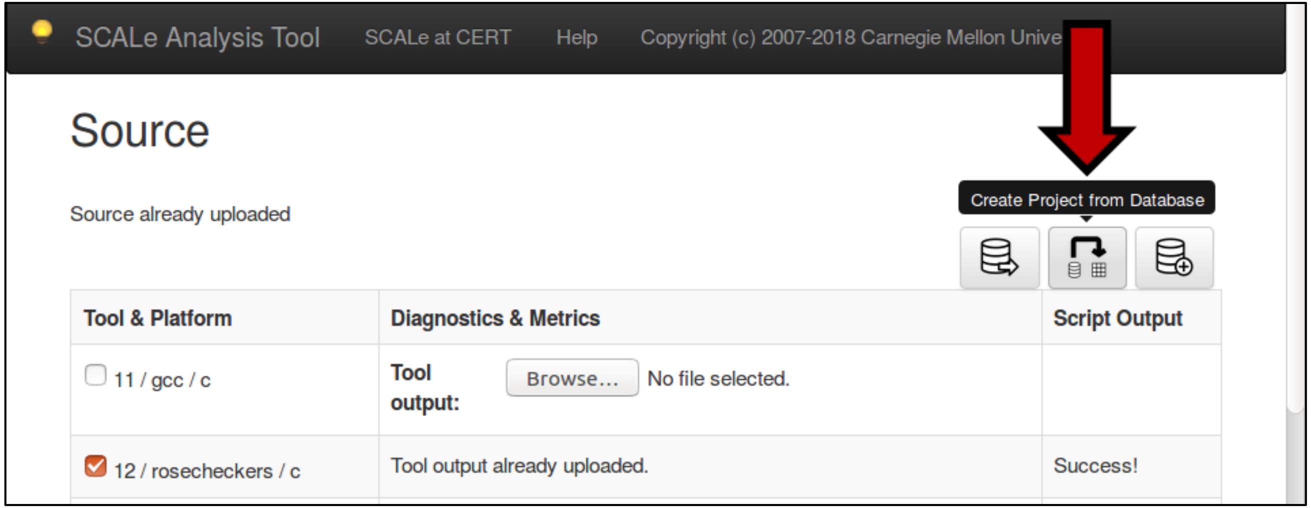
SCALe Analysis Tool   SCALe at CERT   Help   Copyright (c) 2007-2018 Carnegie Mellon University

Source

Archive containing src: Browse... dos2unix-7.2.2.tgz

Tool & Platform	Diagnostics & Metrics	Script Output
☐ 11 / gcc / c	**Tool output:** Browse... No file selected.	
☑ 12 / rosecheckers / c	**Tool output:** Browse... rosecheckers.txt	

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

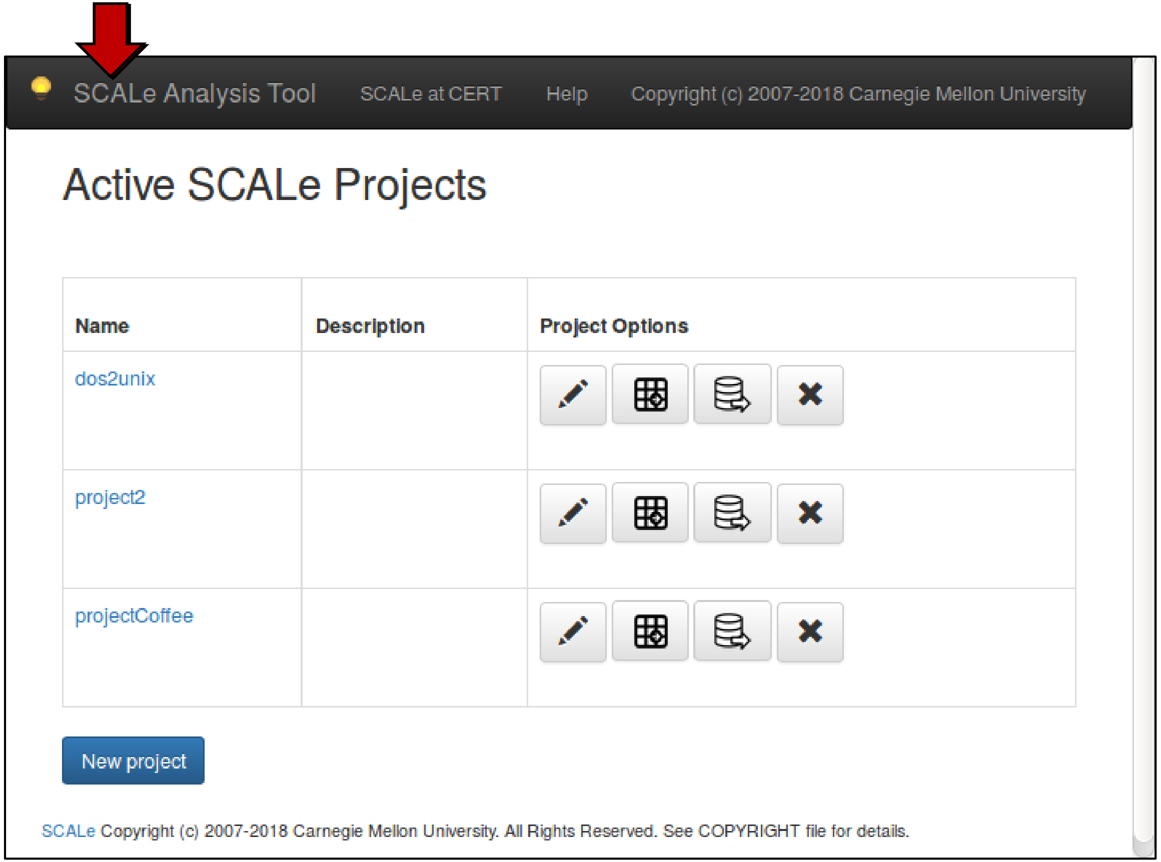

SCALe Analysis Tool
SCALe at CERT
Help
Copyright (c) 2007-2018 Carnegie Mellon Unive
Create Project from Database

Source

Source already uploaded

Tool & Platform	Diagnostics & Metrics	Script Output
□ 11 / gcc / c	**Tool output:** Browse… No file selected.	
☑ 12 / rosecheckers / c	Tool output already uploaded.	Success!

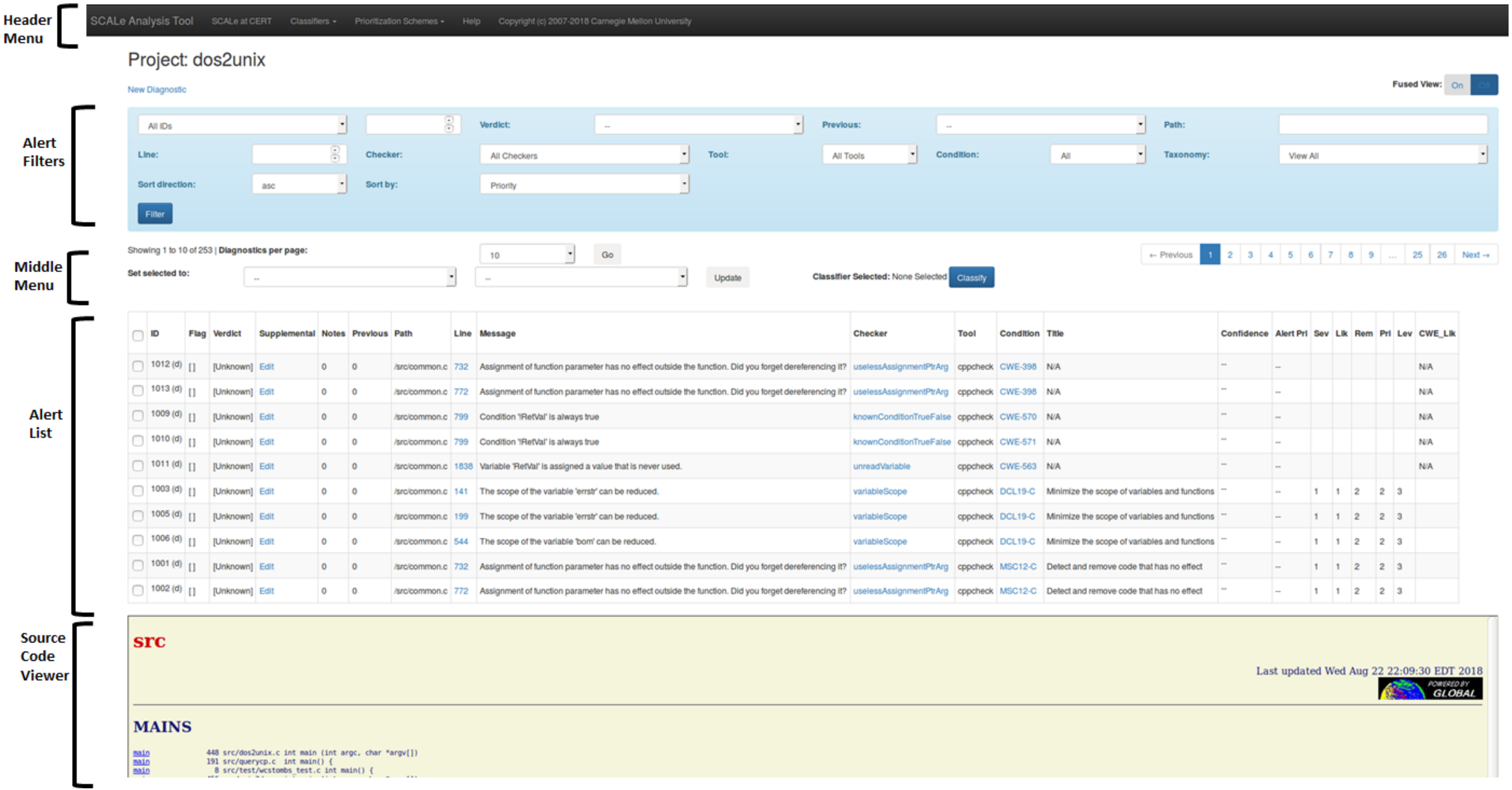
SCALe Homepage



Active SCALe Projects

Name	Description	Project Options
dos2unix		✎ ▦ ⛁ ✖
project2		✎ ▦ ⛁ ✖
projectCoffee		✎ ▦ ⛁ ✖

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



Header Menu

Project: dos2unix

New Diagnostic

Fused View: On Off

Alert Filters

All IDs ▾

Verdict: -- ▾

Previous: -- ▾

Path:

Line:

Checker: All Checkers ▾

Tool: All Tools ▾

Condition: All ▾

Taxonomy: View All ▾

Sort direction: asc ▾

Sort by: Priority ▾

Filter

Middle Menu

Showing 1 to 10 of 253 | **Diagnostics per page:** 10 ▾ Go

← Previous 1 2 3 4 5 6 7 8 9 ... 25 26 Next →

Set selected to: -- ▾ -- ▾ Update

Classifier Selected: None Selected Classify

Alert List

□	ID	Flag	Verdict	Supplemental	Notes	Previous	Path	Line	Message	Checker	Tool	Condition	Title	Confidence	Alert Pri	Sev	Lik	Rem	Pri	Lev	CWE_Lik
□	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
□	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
□	1009 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	799	Condition '!RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-570	N/A	--	--						N/A
□	1010 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	799	Condition '!RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-571	N/A	--	--						N/A
□	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
□	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	
□	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	
□	1006 (d)	[]	[Unknown]	Edit	0	0	/src/common.c	544	The scope of the variable 'bom' can be reduced.	variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions	--	--	1	1	2	2	3	
□	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	
□	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	

Source Code Viewer

src

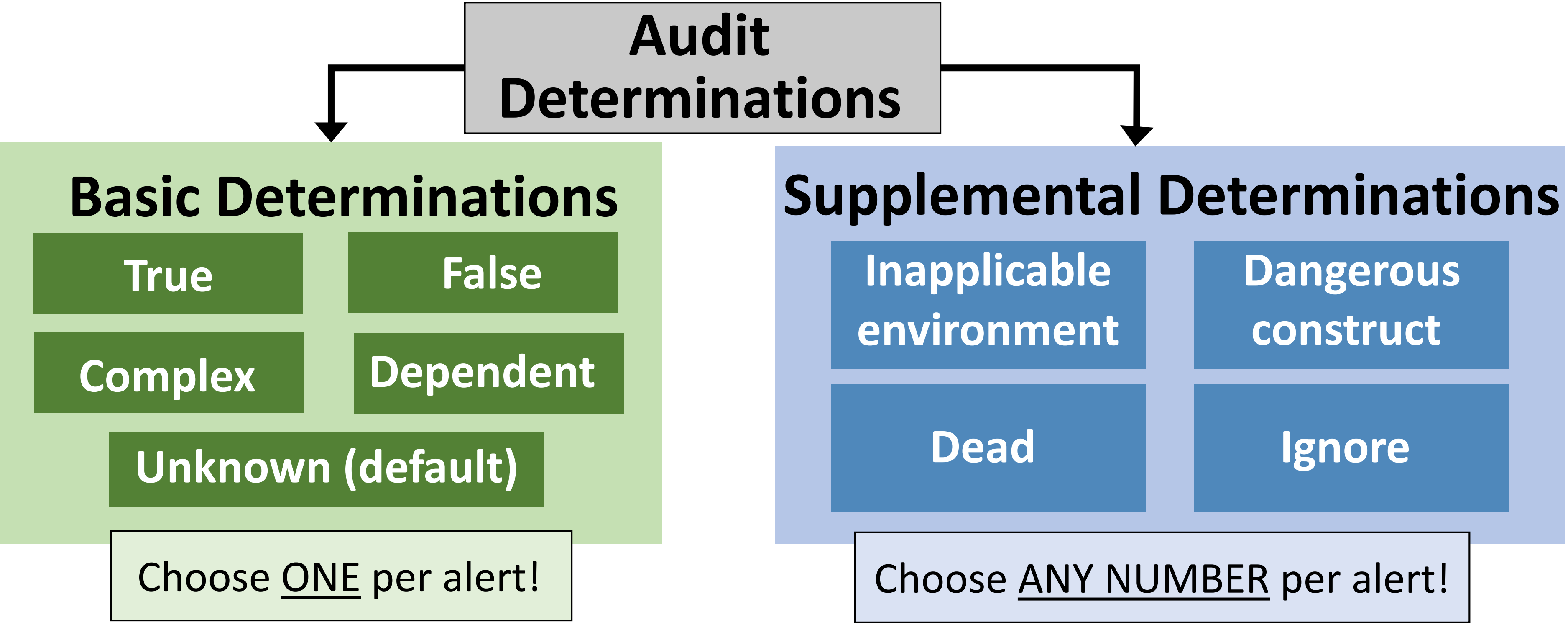
Last updated Wed Aug 22 22:09:30 EDT 2018

POWERED BY GLOBAL

MAINS

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

New Features: Audit Determinations



Audit Determinations

Basic Determinations

True

False

Complex

Dependent

Unknown (default)

Choose ONE per alert!

Supplemental Determinations

Inapplicable environment

Dangerous construct

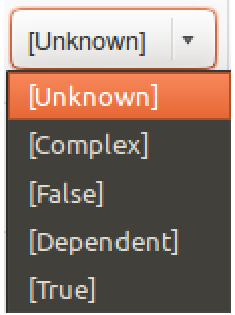
Dead

Ignore

Choose ANY NUMBER per alert!

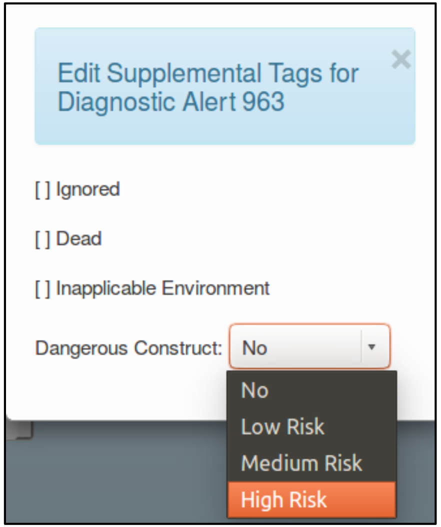
Determinations in GUI

Drop-down for primary verdict



Supplemental determination popup:

- select any number


Edit Supplemental Tags for Diagnostic Alert 963

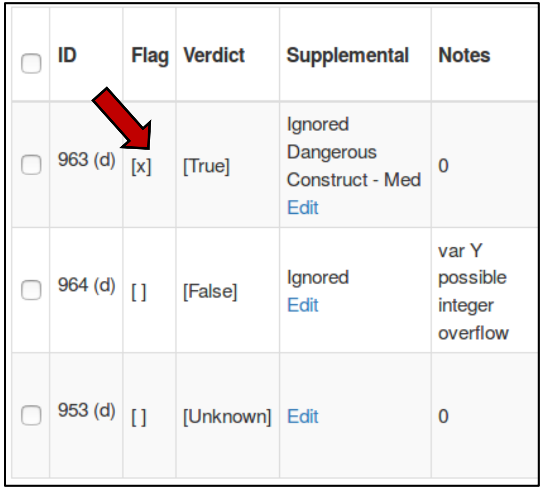
[ ] Ignored

[ ] Dead

[ ] Inapplicable Environment

Dangerous Construct: No

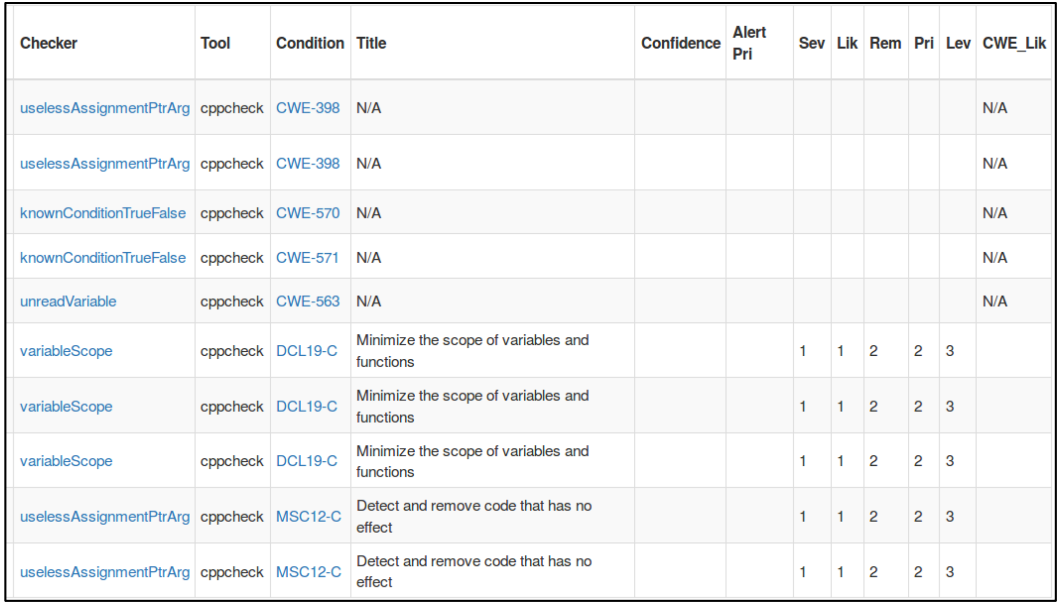
Flag field can have org-defined meaning



□	ID	Flag	Verdict	Supplemental	Notes
□	963 (d)	[x]	[True]	Ignored Dangerous Construct - Med Edit	0
□	964 (d)	[ ]	[False]	Ignored Edit	var Y possible integer overflow
□	953 (d)	[ ]	[Unknown]	Edit	0

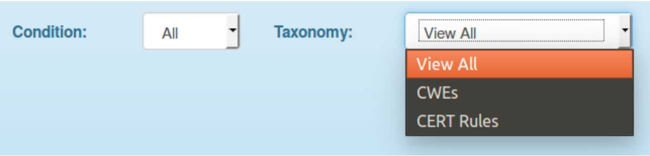
New Features: CWE Taxonomy Added

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

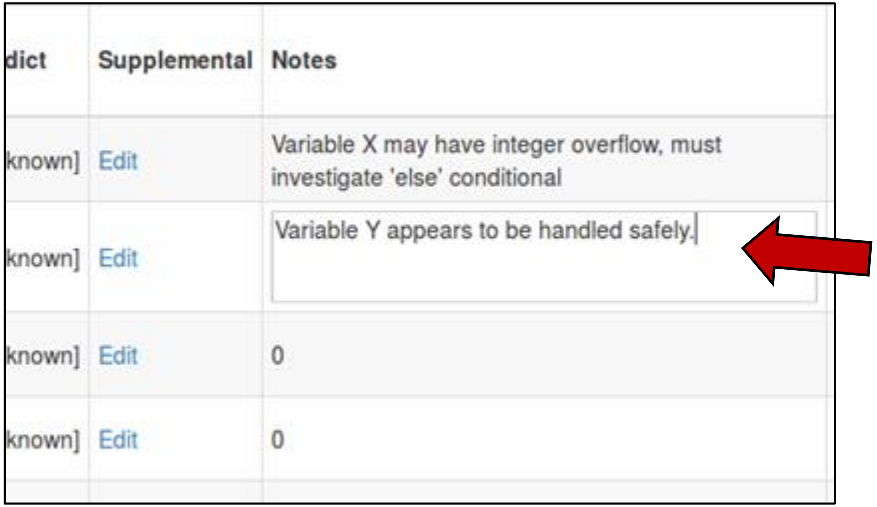


Condition: All

Taxonomy: View All

New Feature: Notes

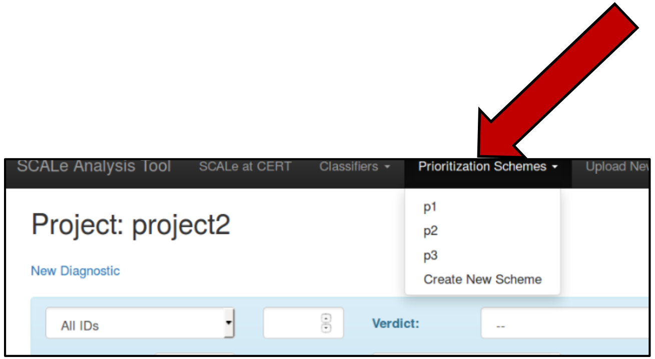
- Notes by auditor about determinations, alert, meta-alert, checker, condition, or language.
- The text can help later auditors reviewing same or similar issues.



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

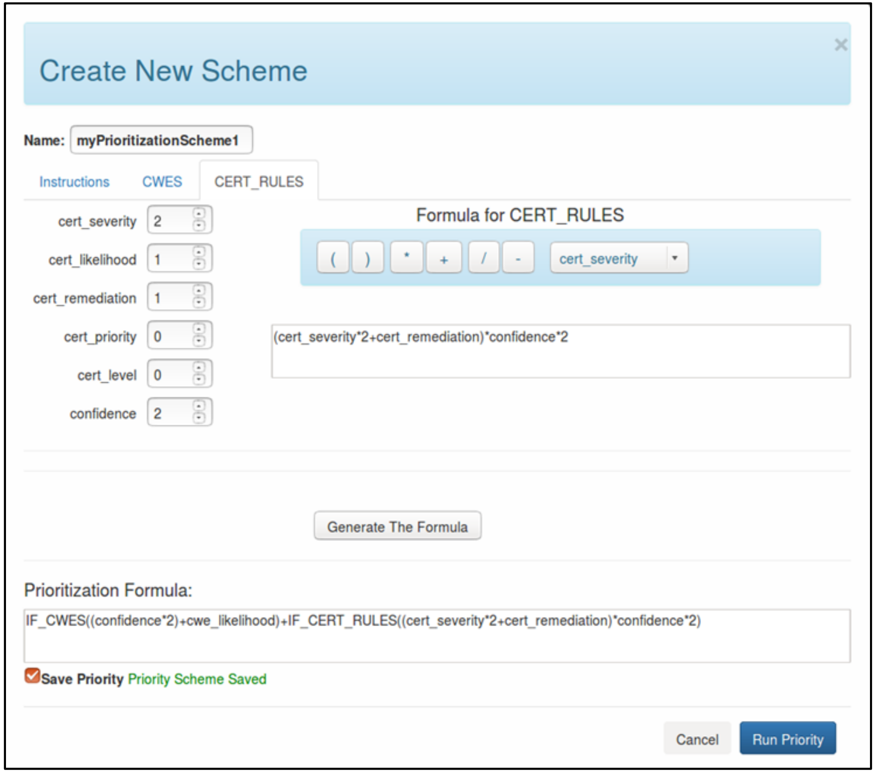
Prioritization schemes with mathematical formulas user can create and/or use



Project: project2

All IDs

Verdict: --



Create New Scheme

Name: myPrioritizationScheme1

Formula for CERT_RULES

cert_severity 2

cert_likelihood 1

cert_remediation 1

cert_priority 0

cert_level 0

confidence 2

cert_severity

(cert_severity*2+cert_remediation)*confidence*2

Prioritization Formula:

IF_CWES((confidence*2)+cwe_likelihood)+IF_CERT_RULES((cert_severity*2+cert_remediation)*confidence*2)

☑ Save Priority Priority Scheme Saved

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

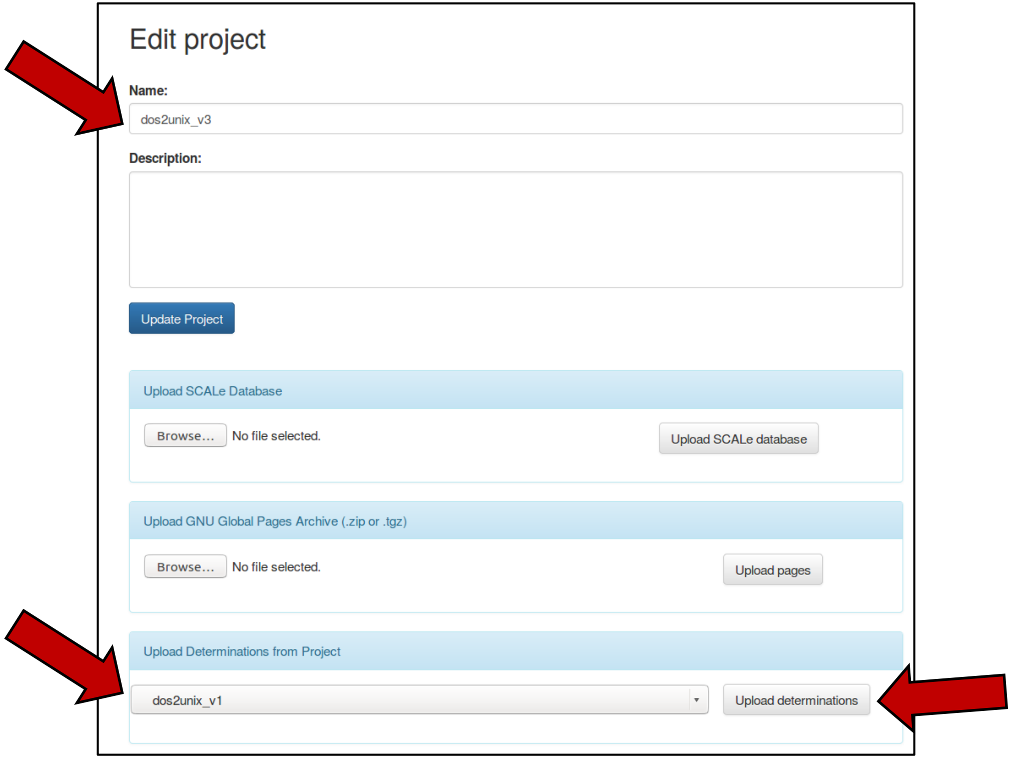
```
meta_alert_id,safeguard_countermeasure,
vulnerability,residual_risk,impact,
threat,risk,complexity,severity,coupling
112,5,1,4,9,1,1,5,5,1
2,9,3,3,3,1,1,1,9,3
3,3,1,1,1,8,1,5,5,1
4,6,1,1,5,2,1,8,8,1
5,2,1,1,2,3,1,7,7,5
6,5,1,4,4,1,2,4,5,1
7,8,5,3,4,8,2,4,9,9
8,2,1,3,2,8,3,8,8,1
9,6,4,3,6,9,1,4,4,4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



Edit project

Name: dos2unix_v3

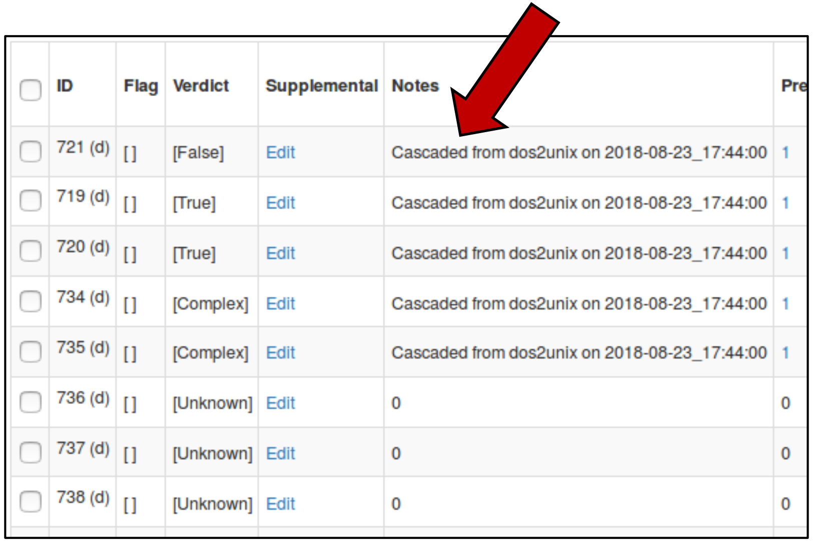
Description:

Update Project
Upload SCALe Database
Browse… No file selected.
Upload SCALe database
Upload GNU Global Pages Archive (.zip or .tgz)
Browse… No file selected.
Upload pages
Upload Determinations from Project
dos2unix_v1
Upload determinations

After Cascaded Import

After cascaded import

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



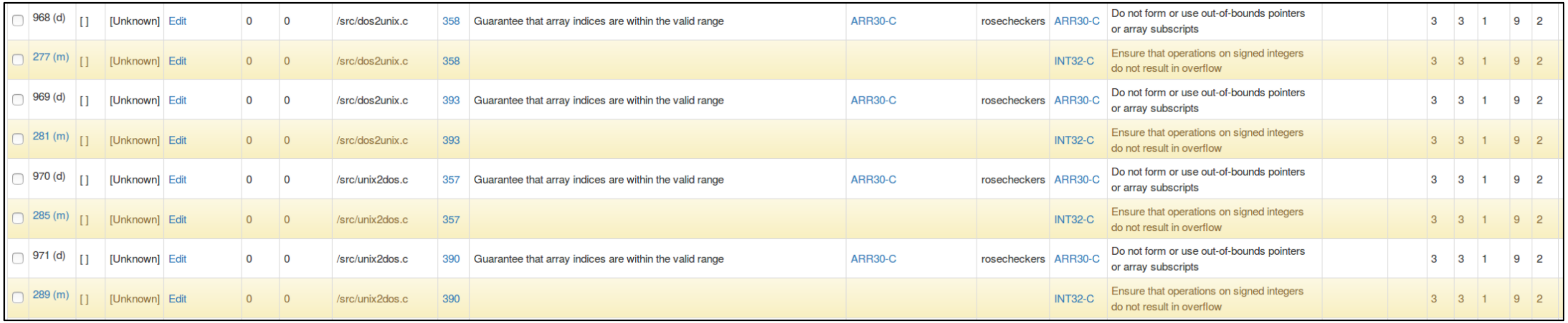
□	ID	Flag	Verdict	Supplemental	Notes	Pre
□	721 (d)	[]	[False]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
□	719 (d)	[]	[True]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
□	720 (d)	[]	[True]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
□	734 (d)	[]	[Complex]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
□	735 (d)	[]	[Complex]	Edit	Cascaded from dos2unix on 2018-08-23_17:44:00	1
□	736 (d)	[]	[Unknown]	Edit	0	0
□	737 (d)	[]	[Unknown]	Edit	0	0
□	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).



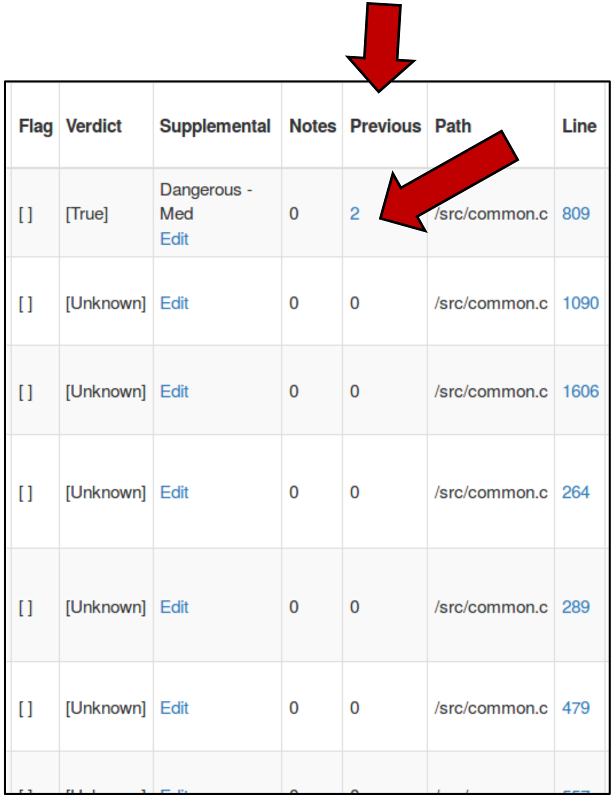
☐	968 (d)	[]	[Unknown]	Edit	0	0	/src/dos2unix.c	358	Guarantee that array indices are within the valid range	ARR30-C	rosecheckers	ARR30-C	Do not form or use out-of-bounds pointers or array subscripts			3	3	1	9	2
☐	277 (m)	[]	[Unknown]	Edit	0	0	/src/dos2unix.c	358				INT32-C	Ensure that operations on signed integers do not result in overflow			3	3	1	9	2
☐	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
☐	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
☐	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
☐	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
☐	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
☐	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

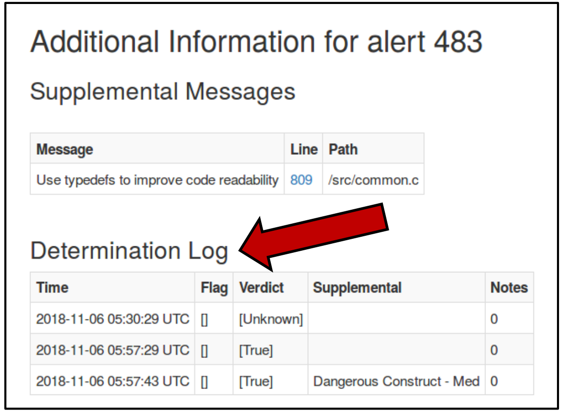
- Determination history
- Upload new tool output
- Hyperlinked checker

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

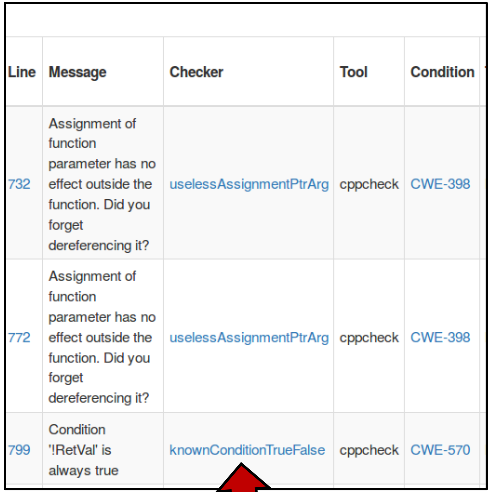
Time	Flag	Verdict	Supplemental	Notes
2018-11-06 05:30:29 UTC	[]	[Unknown]		0
2018-11-06 05:57:29 UTC	[]	[True]		0
2018-11-06 05:57:43 UTC	[]	[True]	Dangerous Construct - Med	0

Hyperlinked Checker

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

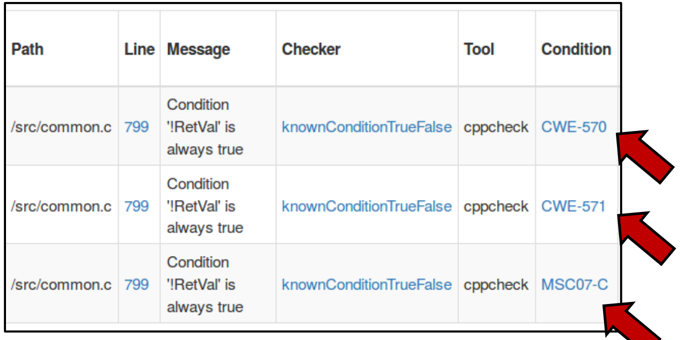
- May be multiple conditions (e.g., a CWE and a CERT rule)
- Helps auditor see related information, including related determinations

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

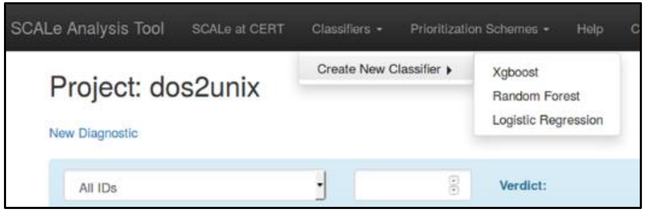


Path	Line	Message	Checker	Tool	Condition
/src/common.c	799	Condition '!RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-570
/src/common.c	799	Condition '!RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-571
/src/common.c	799	Condition '!RetVal' is always true	knownConditionTrueFalse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

Demo

Classification scheme

Classification Scheme



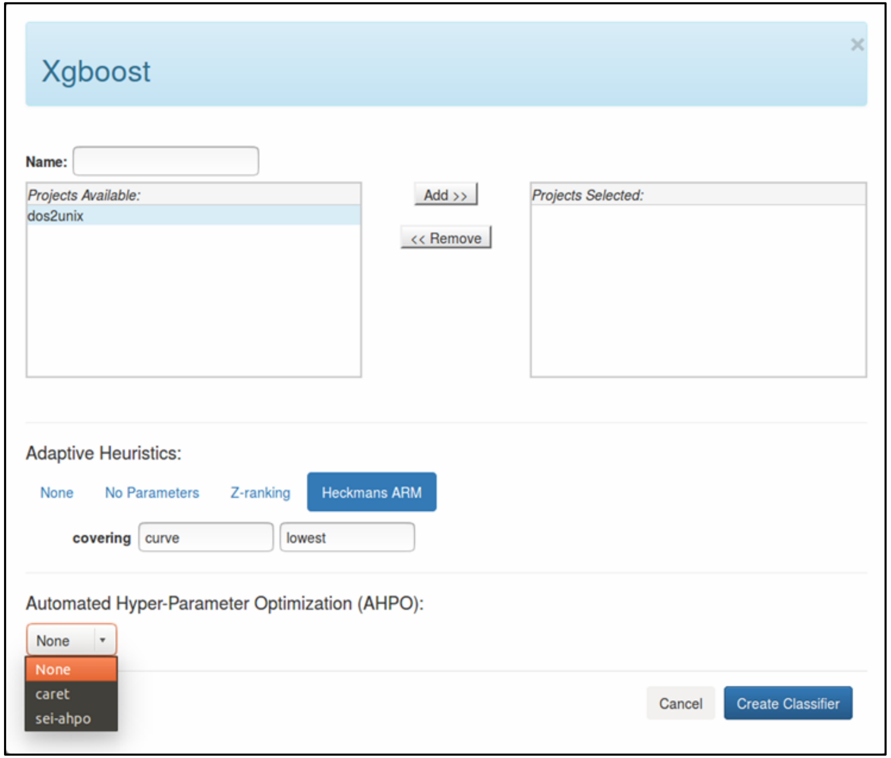
Project: dos2unix

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



Name:

Projects Available: dos2unix

Projects Selected:

Adaptive Heuristics: Heckmans ARM

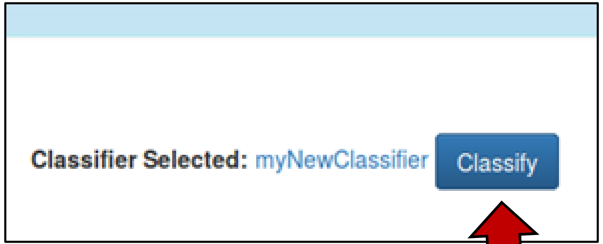
covering curve lowest

Automated Hyper-Parameter Optimization (AHPO): None

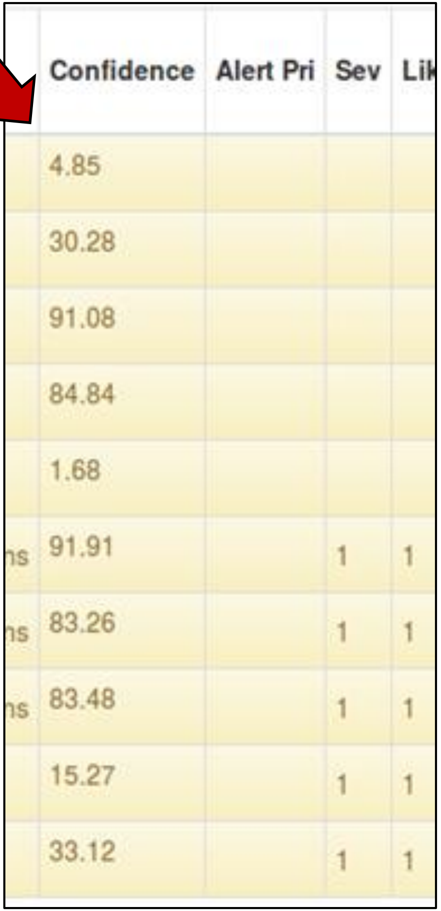
Run the Classifier on a Project

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

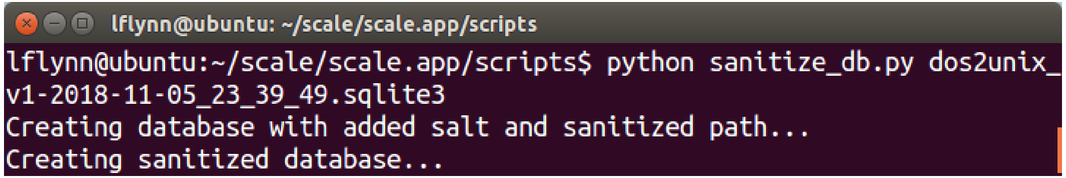


Classifier Selected: myNewClassifier



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

New Feature: Archive Sanitizer



```
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...
```

Added data sanitizer script

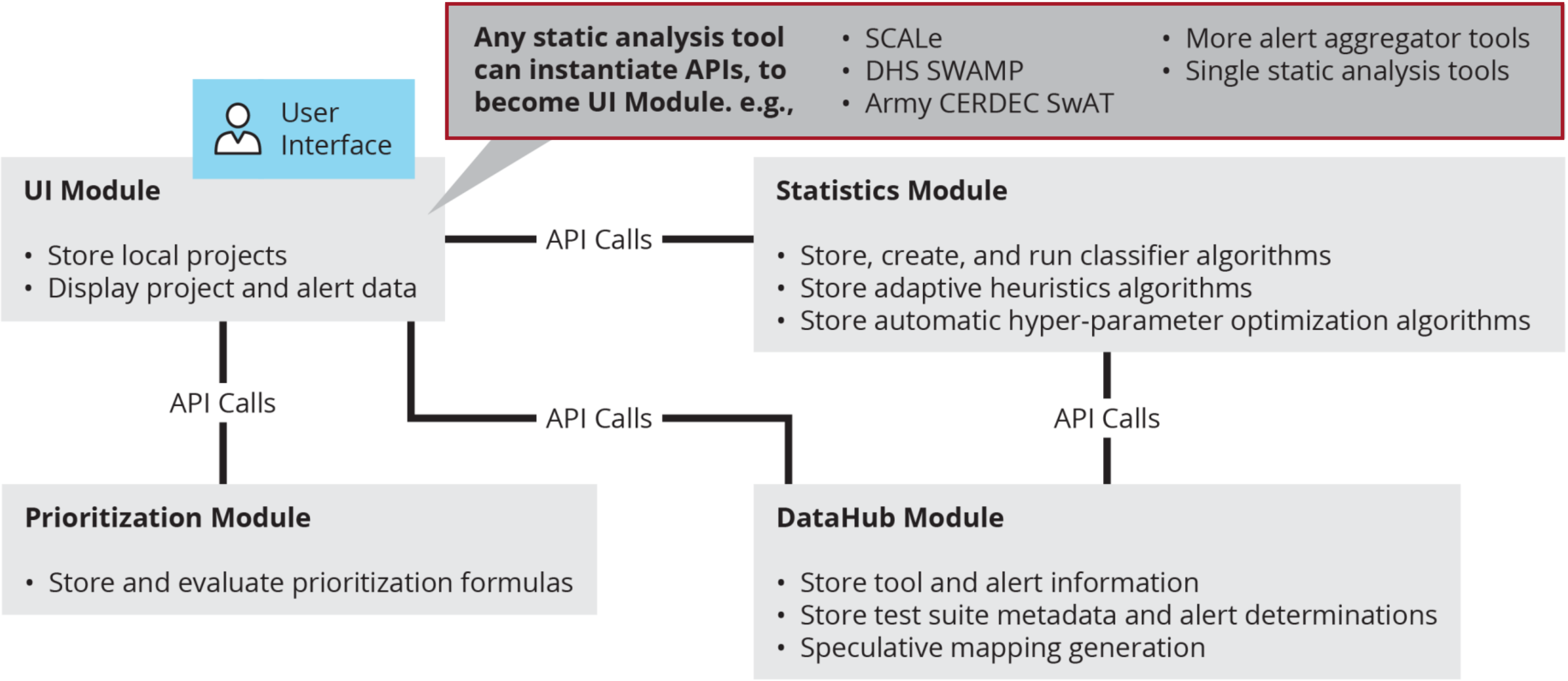
- Anonymizes sensitive fields
- SHA-256 hash with salt
- Enables analysis of features correlated with alert confidence

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



Any static analysis tool can instantiate APIs, to become UI Module. e.g.,

- SCALe
- DHS SWAMP
- Army CERDEC SwAT

- More alert aggregator tools
- Single static analysis tools

User Interface

UI Module

- Store local projects
- Display project and alert data

Statistics Module

- Store, create, and run classifier algorithms
- Store adaptive heuristics algorithms
- Store automatic hyper-parameter optimization algorithms

Prioritization Module

- Store and evaluate prioritization formulas

DataHub Module

- Store tool and alert information
- Store test suite metadata and alert determinations
- Speculative mapping generation

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