SCALe v2 and v3 New Features:
Detail and Demo

Lori Flynn
Senior Software Security Researcher
SCALe Static Analysis Alert Auditing Tool

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
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’)

<table>
<thead>
<tr>
<th>SCALe v2 and v3 New Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTRIBUTION STATEMENT A</td>
</tr>
</tbody>
</table>
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)
- Expected False Positive (e-FP)
- and
- the rest as Indeterminate (I)

Today

Problem: too many alerts
Solution: automate handling

...
SCALe Development

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.
New Features: Slides then Demo

First, we will look at close-ups of the new features in slides.

After that, a demo.
Modified Project Creation
Modified Project Creation

![New project creation interface for Scale](image-url)
Uploading Source Code and Tool Output
### Uploading Code Metrics Tool Output

<table>
<thead>
<tr>
<th></th>
<th>□ 91 / lizard / metric</th>
<th>Tool output: [Browse…] No file selected.</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>92 / ccsm / metric</td>
<td>Tool output: [Browse…] dos2unix_ccsm.c</td>
</tr>
</tbody>
</table>
Next, Create Project with Two Icon Selections: Icon #1
Next, Create Project with Two Icon Selections: Icon #2
## SCALe Homepage

### Active SCALe Projects

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Project Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>dos2unix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>project2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>projectCoffee</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[New project]
New Features: Audit Determinations

**Basic Determinations**
- True
- False
- Complex
- Dependent
- Unknown (default)

**Supplemental Determinations**
- Inapplicable environment
- Dangerous construct
- Dead
- Ignore

Choose ONE per alert!

Choose ANY NUMBER per alert!
Determinations in GUI

Drop-down for primary verdict

Supplemental determination popup:
- select any number

Flag field can have org-defined meaning
New Features: CWE Taxonomy Added

Tool checkers mapped to CWEs and CERT rules.

- 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.

<table>
<thead>
<tr>
<th>dict</th>
<th>Supplemental</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>known</td>
<td>Edit</td>
<td>Variable X may have integer overflow, must investigate 'else' conditional</td>
</tr>
<tr>
<td>known</td>
<td>Edit</td>
<td>Variable Y appears to be handled safely</td>
</tr>
<tr>
<td>known</td>
<td>Edit</td>
<td>0</td>
</tr>
<tr>
<td>known</td>
<td>Edit</td>
<td>0</td>
</tr>
</tbody>
</table>
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.
After Cascaded Import

After cascaded import

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

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

![Image of SCALe Analysis Tool interface]

![Image of Create New Scheme interface]
User Field Uploads

User field uploads

- For advanced users that can work with SQL 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
```
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
## 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).

<table>
<thead>
<tr>
<th>[908 (s)]</th>
<th>[Unknown]</th>
<th>Edit</th>
<th>0</th>
<th>0</th>
<th>arc/dos2unix.c</th>
<th>368</th>
<th>Guarantee that array indices are within the valid range</th>
<th>ARR30-C</th>
<th>rosecheckers</th>
<th>ARR30-C</th>
<th>Do not form or use out-of-bounds pointers or array subscripts</th>
<th>3</th>
<th>3</th>
<th>1</th>
<th>9</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>[277 (m)]</td>
<td>[Unknown]</td>
<td>Edit</td>
<td>0</td>
<td>0</td>
<td>arc/dos2unix.c</td>
<td>368</td>
<td>Guarantee that array indices are within the valid range</td>
<td>ARR30-C</td>
<td>rosecheckers</td>
<td>ARR30-C</td>
<td>Ensure that operations on signed integers do not result in overflow</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>[909 (s)]</td>
<td>[Unknown]</td>
<td>Edit</td>
<td>0</td>
<td>0</td>
<td>arc/dos2unix.c</td>
<td>368</td>
<td>Guarantee that array indices are within the valid range</td>
<td>ARR30-C</td>
<td>rosecheckers</td>
<td>ARR30-C</td>
<td>Do not form or use out-of-bounds pointers or array subscripts</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>[281 (m)]</td>
<td>[Unknown]</td>
<td>Edit</td>
<td>0</td>
<td>0</td>
<td>arc/dos2unix.c</td>
<td>368</td>
<td>Guarantee that array indices are within the valid range</td>
<td>ARR30-C</td>
<td>rosecheckers</td>
<td>ARR30-C</td>
<td>Ensure that operations on signed integers do not result in overflow</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>[970 (s)]</td>
<td>[Unknown]</td>
<td>Edit</td>
<td>0</td>
<td>0</td>
<td>arc/unix2dos.c</td>
<td>357</td>
<td>Guarantee that array indices are within the valid range</td>
<td>ARR30-C</td>
<td>rosecheckers</td>
<td>ARR30-C</td>
<td>Do not form or use out-of-bounds pointers or array subscripts</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>[285 (m)]</td>
<td>[Unknown]</td>
<td>Edit</td>
<td>0</td>
<td>0</td>
<td>arc/unix2dos.c</td>
<td>357</td>
<td>Guarantee that array indices are within the valid range</td>
<td>ARR30-C</td>
<td>rosecheckers</td>
<td>ARR30-C</td>
<td>Ensure that operations on signed integers do not result in overflow</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>[971 (s)]</td>
<td>[Unknown]</td>
<td>Edit</td>
<td>0</td>
<td>0</td>
<td>arc/unix2dos.c</td>
<td>357</td>
<td>Guarantee that array indices are within the valid range</td>
<td>ARR30-C</td>
<td>rosecheckers</td>
<td>ARR30-C</td>
<td>Do not form or use out-of-bounds pointers or array subscripts</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>[289 (m)]</td>
<td>[Unknown]</td>
<td>Edit</td>
<td>0</td>
<td>0</td>
<td>arc/unix2dos.c</td>
<td>357</td>
<td>Guarantee that array indices are within the valid range</td>
<td>ARR30-C</td>
<td>rosecheckers</td>
<td>ARR30-C</td>
<td>Ensure that operations on signed integers do not result in overflow</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>2</td>
</tr>
</tbody>
</table>
New Feature: Archive Sanitizer

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.
New Feature: Determination History

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

<table>
<thead>
<tr>
<th>Line</th>
<th>Message</th>
<th>Checker</th>
<th>Tool</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>752</td>
<td>Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?</td>
<td>uselessAssignmentPtrArg</td>
<td>c��check</td>
<td>CWE-398</td>
</tr>
<tr>
<td>772</td>
<td>Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?</td>
<td>uselessAssignmentPtrArg</td>
<td>c操check</td>
<td>CWE-398</td>
</tr>
<tr>
<td>799</td>
<td>Condition &quot;RetVal&quot; is always true</td>
<td>knownConditionTrueFalse</td>
<td>c操check</td>
<td>CWE-570</td>
</tr>
</tbody>
</table>

All meta-alerts for checker + location

<table>
<thead>
<tr>
<th>Path</th>
<th>Line</th>
<th>Message</th>
<th>Checker</th>
<th>Tool</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>src/common.c</td>
<td>799</td>
<td>Condition &quot;RetVal&quot; is always true</td>
<td>knownConditionTrueFalse</td>
<td>c操check</td>
<td>CWE-570</td>
</tr>
<tr>
<td>src/common.c</td>
<td>799</td>
<td>Condition &quot;RetVal&quot; is always true</td>
<td>knownConditionTrueFalse</td>
<td>c操check</td>
<td>CWE-571</td>
</tr>
<tr>
<td>src/common.c</td>
<td>799</td>
<td>Condition &quot;RetVal&quot; is always true</td>
<td>knownConditionTrueFalse</td>
<td>c操check</td>
<td>MSC07-C</td>
</tr>
</tbody>
</table>
Demo
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

• GitHub SCALe v2 publication Aug. 2018
• Paper “Prioritizing Alerts from Multiple Static Analysis Tools, using Classification Models,” SQUADE (ICSE workshop)
• SEI blog post: “Test Suites as a Source of Training Data for Static Analysis Alert Classifiers” (Apr. 2018)
• SEI Podcast (video): “Static Analysis Alert Classification with Test Suites” (Sep. 2018)
• SEI blog post: “SCALe: A Tool for Managing Output from Static Code Analyzers” (Sep. 2018)
• SEI Technical Report “Integration of Automated Static Analysis Alert Classification and Prioritization with Auditing Tools” (Publication expected November 2018)
Contact Information

Lori Flynn
Senior Software Security Researcher
Telephone:  +1 412.268.7886
Email:  lflynn@sei.cmu.edu