IBM
if ["$(find . | grep 'IMG[0-9]')"]
rm –rf /
Remove-Partition
PSScriptAnalyzer
IBM Research
socc
ShellCheck
find . -name '*.txt' -exec sh -c 'echo "{}"' \;

Automatically Detecting Risky Scripts in Infrastructure Code

Motivations

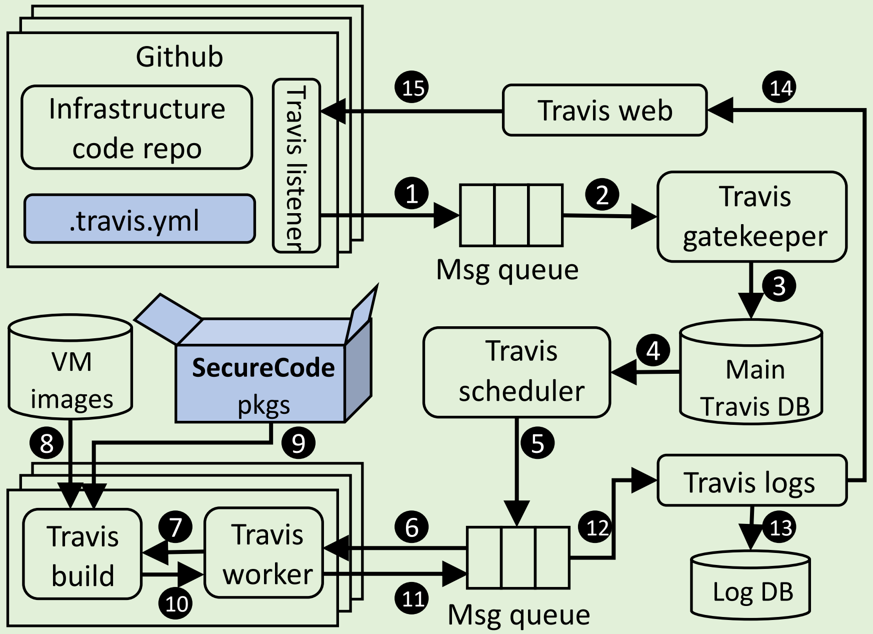
- Risky patterns in **IaC embedded scripts** introduce bugs and expose vulnerabilities.
- **Amazon S3 service outage**: A removal cmd caused 5-hour service disruption with $150 million loss.
- **IaC linters** cannot check IaC embedded scripts.
- Generic **script-analyzers** introduce FPs and FNs.

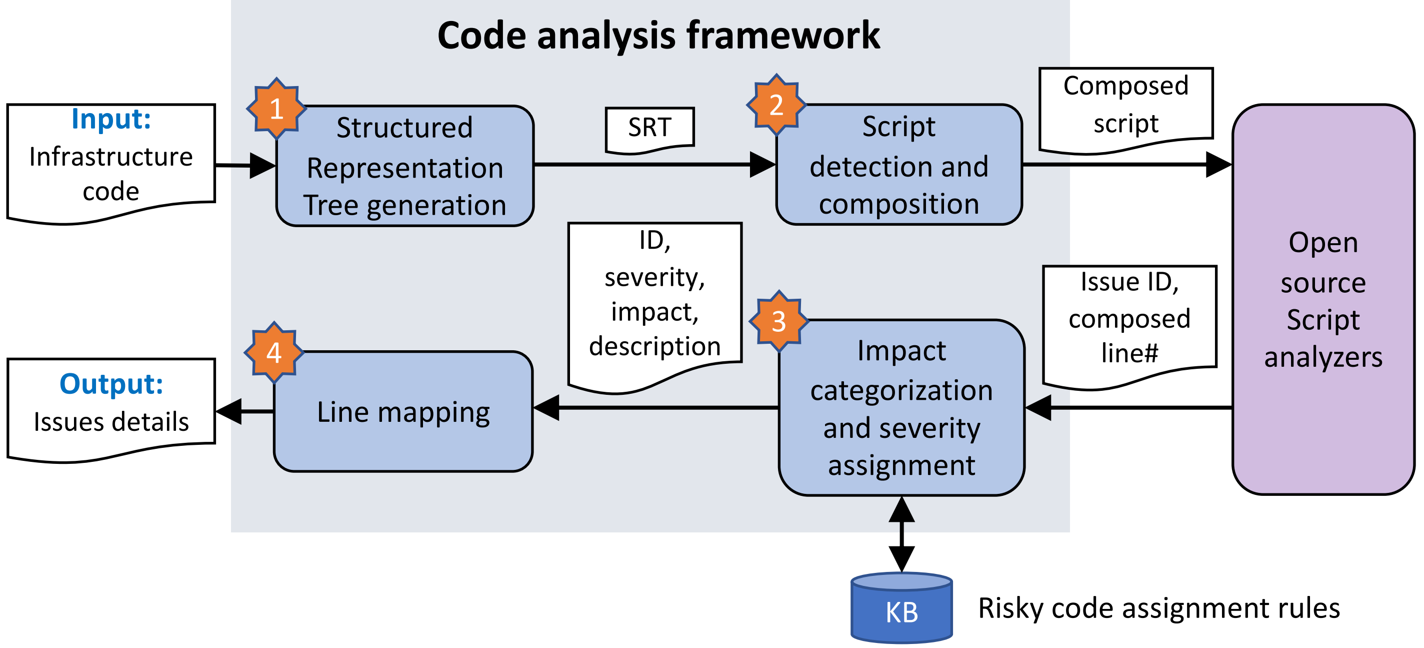
Opportunity & Contributions

- Bridge the gap between generic script-analyzers and business consequence to deliver an accurate **checking framework**.
- Generate risky code **knowledge-base** with severity levels and business impact categories.
- Implement a real-world solution, i.e., **SecureCode**, on the proposed framework.

SecureCode Implementation & Setup

- SecureCode checks risky scripts in Ansible playbooks.
- Integrate with IBM CI/CD pipeline.
- Test 45 IBM Services community repos.




Code analysis framework

Input:
Infrastructure code
1
Structured Representation Tree generation
SRT
2
Script detection and composition
Composed script
Open source Script analyzers
Issue ID, composed line#
3
Impact categorization and severity assignment
ID, severity, impact, description
4
Line mapping
Output:
Issues details
KB

Risky code assignment rules

Output Format

ID: SC2154 **Type:** Warning **Impact:** Security **Severity:** High

Description: unassigned **ansible_node** is vulnerable to injection attacks.

Detailed description: file:///localpath/SecureCode/rules/SC2154.md
https://remotepath/SecureCode/rules/SC2154.md

Location: roles/backup_missed_unix/tasks/main.yml:24

Original: shell {{ tsm_command }} "select count(*) from sessions where client={{ ansible_node }}"

Composed: shell dsmadmc -se=${tsm_servername} -id=${param_tsmuser} -pass=${param_tsmpass} -tabdelimited -dataonly=yes -noconfirm "select count(*) from sessions where client_name=${**ansible_node**}”

Explanation: Unassigned ansible_node variable allows a user to pass any value from a cmd line; ansible_node is used in a SQL command, which is vulnerable to SQL injection attacks.

User Experience

- **Throughput Improvement**: LOCs reviewed per person per day
  - **5x** vs manual, **2-5x** vs ShellCheck, **2-3x** vs PSScriptAnalyzer
- **Efficiency Gain**: the number of issues to be identified
  - **5x** vs manual, **2-3x** vs ShellCheck, **2-3x** vs PSScriptAnalyzer

Detection Accuracy & Statistics

- SecureCode detects **3535** issues from the 45 repos with 1492 automation files.
- **116** issues are FPs.
- Stats of **3419** true bugs are shown in the right table.

Impact	High	Medium	Low	Total
Non-risk	0	0	862	862
Availability	2	0	0	2
Performance	0	51	0	51
Security	1204	0	0	1204
Reliability	485	247	568	1300
Total	1691	298	1430	3419

Ting Dai (IBM Research), Alexei Karve (IBM Research), Grzegorz Koper (IBM GTS), Sai Zeng (IBM Research), “Automatically Detecting Risky Scripts in Infrastructure Code”, SoCC’20.