The New ‘Normalized’

Standardizing Security Data using the OASIS Heimdall Data Format (OHDF) ©

Aaron Lippold

Principal Cybersecurity Engineer

Chief Engineer of the MITRE Security Automation Framework (MITRE SAF) ©

The MITRE Corporation

Mike Fraser

VP & Field CTO of DevSecOps & Sophos Factory

Sophos

SOPHOS MITRE | SOLVING PROBLEMS FOR A SAFER WORLD®

Aaron Lippold

Principal Cybersecurity Engineer
Chief Engineer of the MITRE SAF

MITRE Security Automation Framework (SAF) ©

https://saf.mitre.org

alippold@mitre.org

@aaronlippold

https://www.linkedin.com/in/aaronlippold/

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD®

Mike Fraser

VP & Field CTO of DevSecOps

Sophos

https://www.sophos.com

Sophos Factory

https://www.sophos.com/en-us/products/sophos-factory

Mike.Fraser@Sophos.com

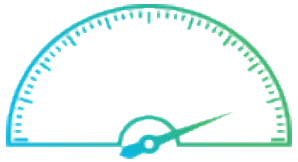
@itascode

https://www.linkedin.com/in/itascode/

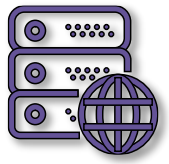
SOPHOS

What is the MITRE Security Automation Framework©?

A suite of open-source security automation tools that facilitate the development, collection, and standardization of content for use by government and industry organizations to



Accelerate ATO



Establish Security Requirements

Build Security In

Assess/ Monitor Vulnerabilities

- **MITRE SAF© VISION**
- *Implement evolving security requirements while deploying apps at speed*

MITRE

Challenge: So Many Formats, So Little Time

- Most security tools do not provide full context to relevant compliance standards for comparison across security tools.

- Security tools typically generate data in unique formats that require multiple dashboards and utilities to process.

- OHDF reduces the time it takes to process security assessments, data in disparate locations and inconsistent semantics of a data element between formats.

Provide a Pathway for Data Alignment

- OHDF is a developing standard format for exchanging normalized security data between cybersecurity tools.

- For Context:

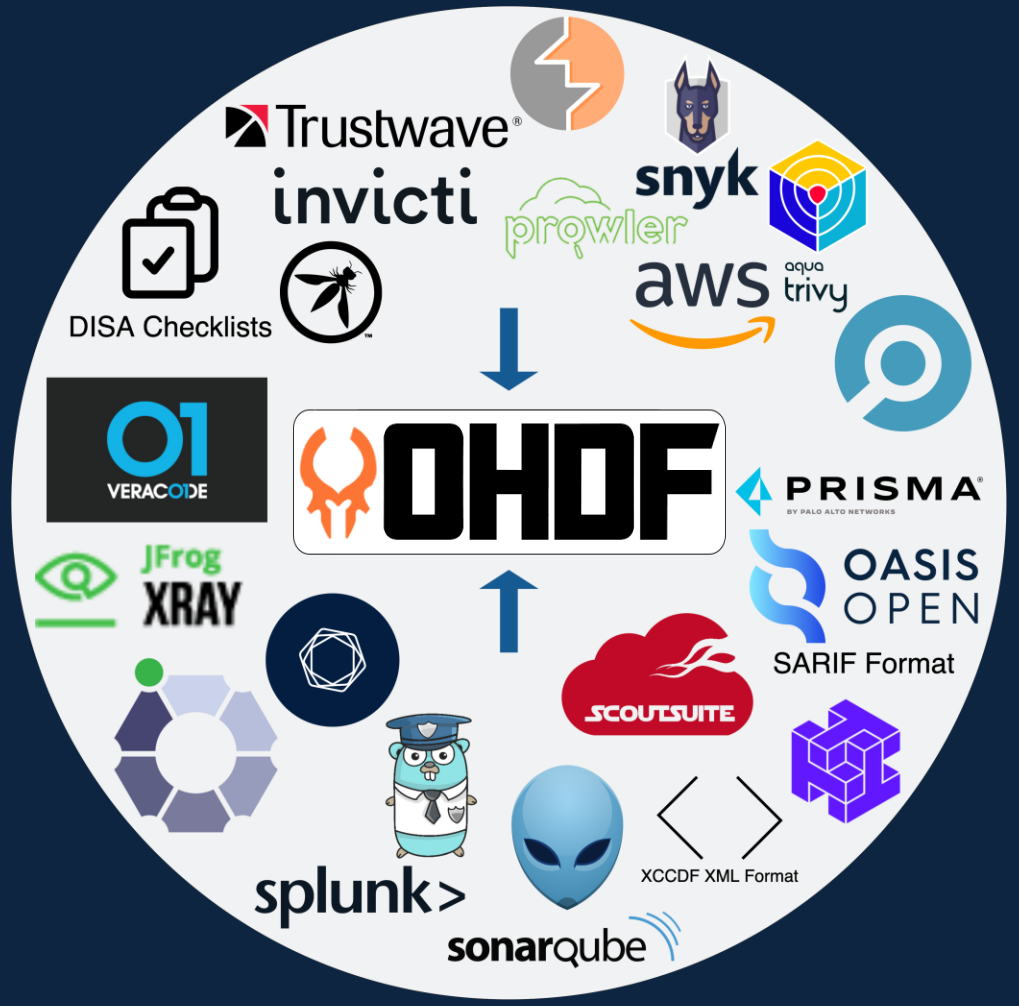
  - ‘Standardization’ is the process of defining data elements in a consistent and contextualized manner.
  - ‘Normalization’ is the process for mapping a format's data elements into another format’s data elements.

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ohdf

MITRE

OHDF Data Exchange Format

- Enables the consistent integration, aggregation, and analysis of security data from all available sources
- Preserves data integrity with original source data
- Maximizes interoperability and data sharing
- Facilitates the transformation and transport of data between security/management processes or technologies
- Allows for the mapping and enrichment of security data to relevant compliance standards (GDPR, NIST SP 800-53, PCI-DSS, etc.)



An **open**, **standardized**, and **normalized** format for exchanging security and risk information data between cybersecurity tools.

The OHDF Technical Committee

- Formed in 2023 underneath OASIS Open, the international open source and standards consortium
- Established, proven foundation for the first draft of the specification (v0.9)
- Co-Chairs are Aaron Lippold (MITRE) and Mike Fraser (Sophos)
- Secretary is Stefan Hagen

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ohdf

MITRE

What does OASIS Heimdall Data Format (OHDF) © look like?

JSON-Schema based implementation

- **Current in V0.9 – Initial Base**
- **Structured, normalized, contextualized results**
- **Easy to integrate & add new converters**
- **Transportable across multiple domains**

HDF Core Elements

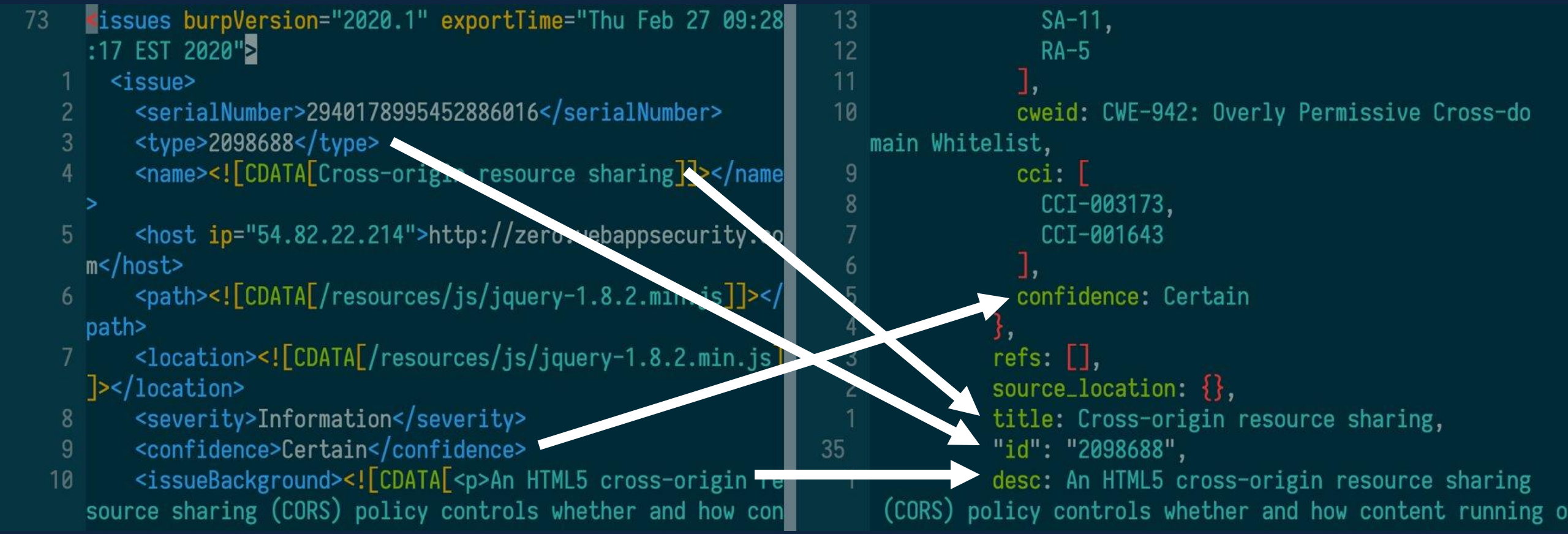
- Test Title – High level overview of the test(s) goal
- Test Description – Details on the intent and possible impact
- Test Audit, aka 'check text' – the validation actions we are asking of the end user
- Test Remediation, aka 'fix text' – the remediation actions we are asking of the end user
- NIST SP 800-53 Control Alignment(s) – the NIST SP 800-53 security control this test(s) relates to
- Test Severity - The static default of the control categorization impact
- Test Impact - The context-specific severity during testing
- Other data tags specific to the source benchmark – other data elements that enhance the context of the test(s)
  - CIS - tags such as the level, version and scoring status of the CIS benchmark
  - DISA STIG - tags such as the DISA Common Correlation Index Identifier (CCI)
- Test Elements – the individual tests that make up the actions in the 'Check Text'

Schema


```
type: "object",
additionalProperties: true,
required: [
  0: "platform",
  1: "profiles",
  2: "statistics",
  3: "version"
],
properties: {
  platform: { ... },
  profiles: { ... },
  statistics: { ... },
  version: { ... }
},
description: "The top level value containing all of the results.",
title: "Exec JSON Output",
definitions: {
  Control_Description: { ... },
  Control_Group: { ... },
  Control_Result: { ... },
  Control_Result_Status: { ... },
  Dependency: { ... },
  Exec_JSON_Control: { ... },
  Exec_JSON_Profile: { ... },
  Platform: { ... },
  Reference: { ... },
  Source_Location: { ... },
  Statistic_Block: { ... },
  Statistic_Hash: { ... },
  Statistics: { ... },
  Supported_Platform: { ... },
  Waiver_Data: { ... },
  Attestation_Status: { ... },
  Attestation_Data: { ... }
```


MITRE

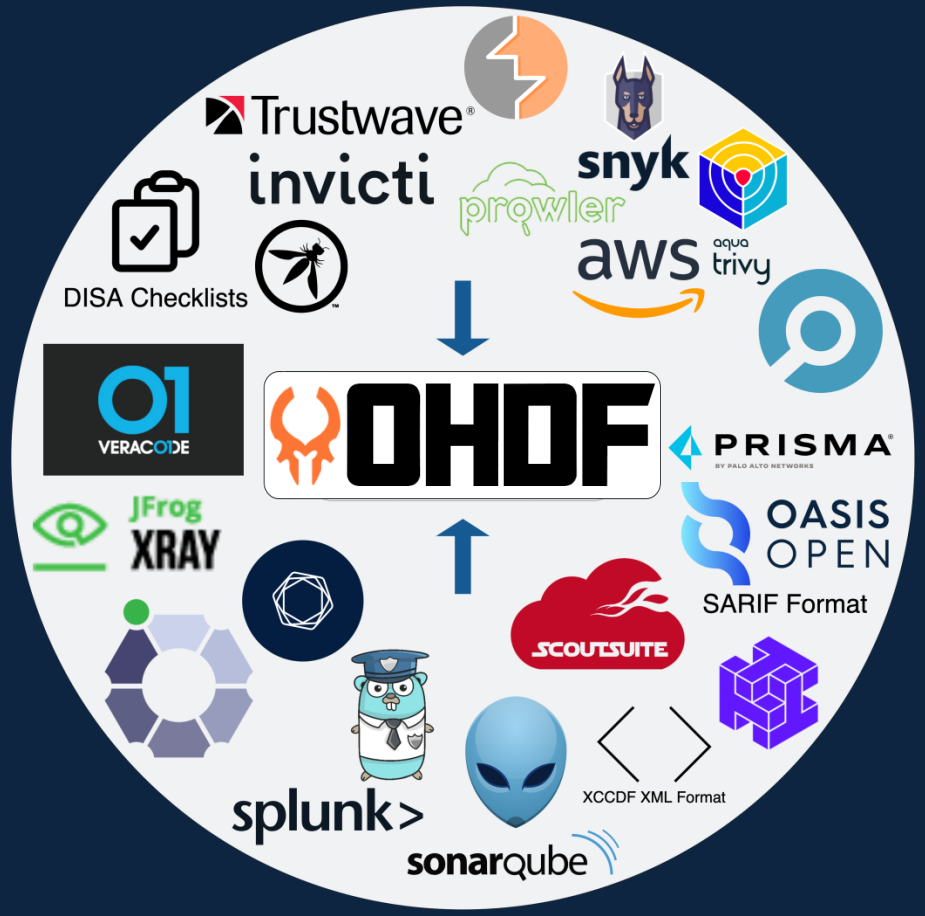
Technical Example: BurpSuite -> OHDF



OHDF Currently Supported Data Formats

- Normalization
- Data Exchange
- Additional context
- CI/CD

Convert from OHDF



Convert to OHDF

OHDF Current Ecosystem, Tooling & Libraries

- ✓ Translate data into a standard format to ensure interoperability
- ✓ Use OHDF Converters as a library in your custom application
- ✓ Add data conversion in your pipeline for automatic normalization in each run

- SAF© CLI (command line interface)

- OHDF Converters

- SAF© GitHub Actions

- Heimdall Lite
- Heimdall Server

- Sophos Factory
- Tools Support with OHDF
- SAF © Solution Catalog

Supported Risk Information Sources

- AWS Security Hub
- Splunk
- AWS Config
- Snyk
- Aqua Security Trivy
- Tenable Nessus
- DBProtect
- CSV / XLSX
- Netsparker / Invicti
- Burp Suite
- GoSec
- Ion Channel
- Prisma
- SonarQube
- OWASP ZAP
- Prowler
- Fortify
- JFrog Xray
- Nikto
- SARIF
- ScoutSuite
- Twistlock
- DISA Checklist
- DISA XCCDF Results
- And more!

https://saf.mitre.org

MITRE

OHDF in Action

- Normalization
- Data Exchange
- Additional context
- CI/CD

SAF CLI & OHDF Libraries allow for users to enhance and enrich their data – attest, waiver, supplement & passthrough

- Manually attest to ”Not Reviewed” controls

```
{
  "control_id": "V-61409",
  "explanation":
    "Audit logs are automatically backed up and preserved as necessary",
  "frequency": "monthly",
  "status": "passed",
  "updated": "2099-05-02",
  "updated_by": "Json Smith, Security"
},
```

- Waive not applicable controls
- Supplement information
  - `$ saf supplement target read -i hdf_with_target.json | jq -rc '.key = "new value”’ | xargs -0 -I{} saf supplement target write -i hdf_with_target.json -d {}`

MITRE

Example: OHDF via GitHub Actions

- Normalization
- Data Exchange
- Additional context
- CI/CD

```yaml
name: Code analysis

on:
  pull_request:
    types: [opened, synchronize, reopened]
  workflow_call:
  workflow_dispatch:

env:
  SONARQUBE_PROJECT_KEY: "saf-example-pipeline"

jobs:
  code_analysis:
    name: Code analysis
    runs-on: ubuntu-20.04
    steps:
      - name: Check out
        uses: actions/checkout@v3
        with:
          fetch-depth: 0
      - name: Sonarqube scan
        uses: kitabisa/sonarqube-action@v1.2.0
        with:
          host: ${{ secrets.SONARQUBE_HOST }}
          login: ${{ secrets.SONARQUBE_TOKEN }}
          projectKey: ${{ env.SONARQUBE_PROJECT_KEY }}
      - name: Convert sonarqube scan to hdf
        uses: mitre/saf_action@v1
        with:
          command_string: "convert sonarqube2hdf -n ${{ env.SONARQUBE_PROJECT_KEY }} -u ${{ secr
ets.SONARQUBE_HOST }} --auth ${{ secrets.SONARQUBE_TOKEN }} -o sonarqube-hdf.json"
      - name: Upload sonarqube hdf to heimdall
        run: 'curl --show-error --fail --insecure -F "data=@sonarqube-hdf.json" -F "filename=${{
 github.ref_name }}-sonarqube-hdf.json" -F "public=true" -H "Authorization: Api-Key ${{ secrets.
HEIMDALL_API_KEY }}" "${{ secrets.HEIMDALL_HOST }}/evaluations"'
```

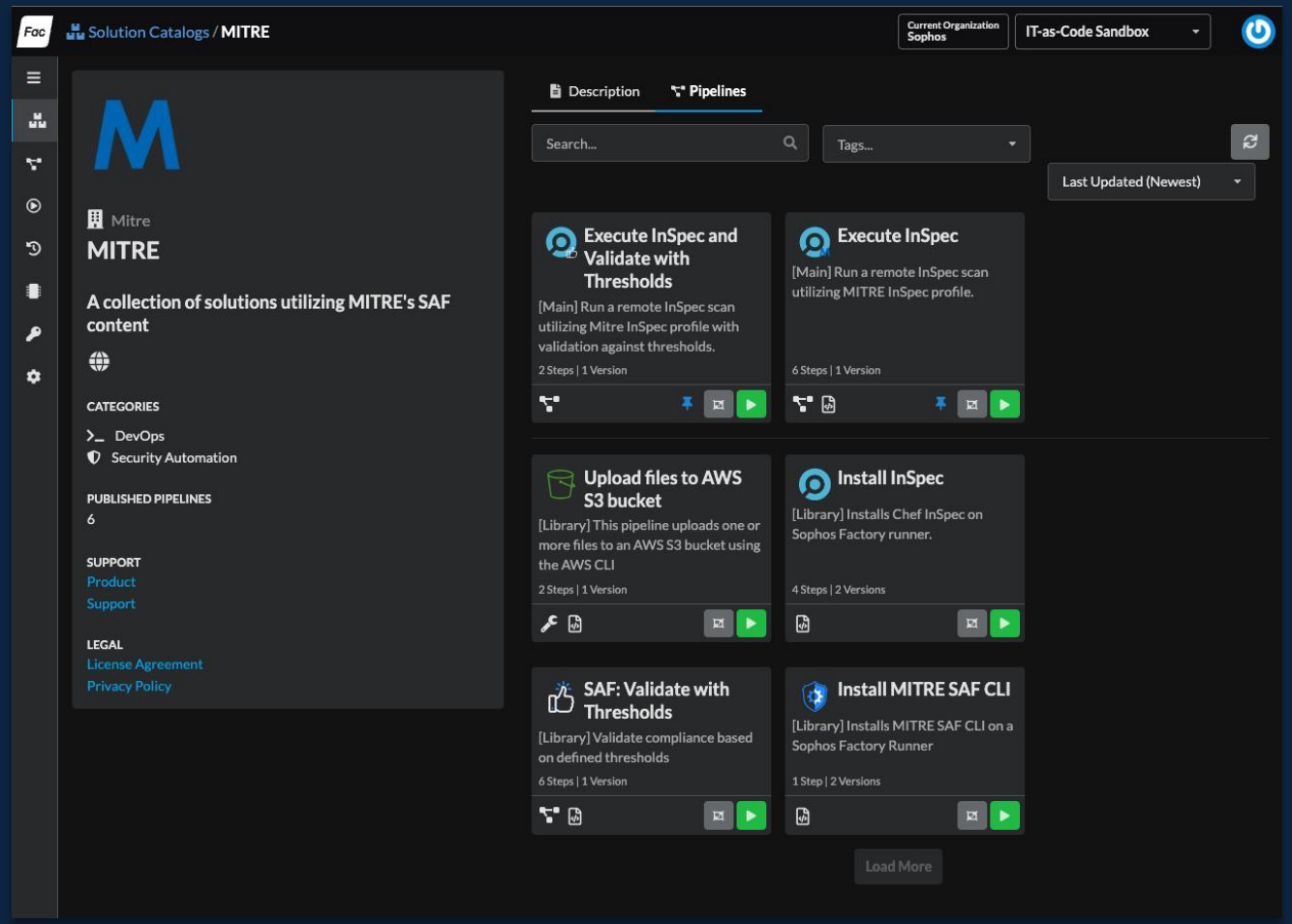
MITRE

SOPHOS: OHDF Integration in Sophos Factory

Sophos Factory

DevSecOps Automation Platform

- Tool support of existing modules
  - SonarQube, Trivy, Twistlock, & OpenSCAP
- Normalization of data across tools
- SAF CLI support in pipeline
- Add additional tools via pipelines
- Published via solution catalog

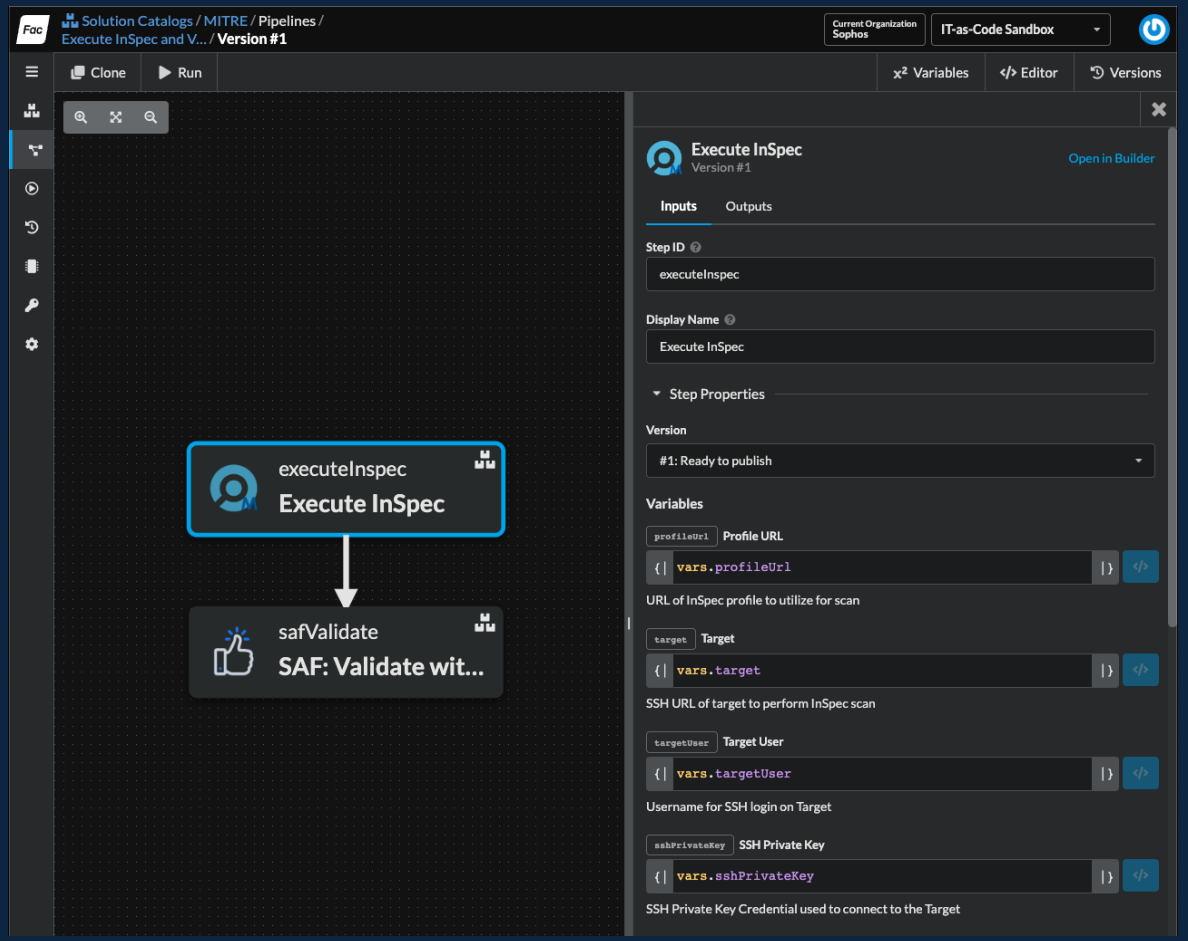


SOPHOS: OHDF Integration with Sophos Factory in Action

- Start with one tool, e.g., InSpec
- Add additional tools for use case
- View output in Run History

Future Work

- Push to Heimdall Server
- Push to other systems



Integration Paths for OHDF

- Natively generate/process OHDF
  - InSpec, SAF CLI, Heimdall Application
- Pipeline
  - SAF CLI subcommand (`saf convert`)
- Libraries
  - HDF Converters
  - InSpecJS

- https://github.com/mitre/saf/wiki/How-to-recommend-development-of-a-mapper

```
control 'SV-230367' do
  title "RHEL 8 user account passwords must be configured so that existing
passwords are restricted to a 60-day maximum lifetime."
  desc  "Any password, no matter how complex, can eventually be cracked.
Therefore, passwords need to be changed periodically. If RHEL 8 does not limit
the lifetime of passwords and force users to change their passwords, there is
the risk that RHEL 8 passwords could be compromised."
  desc  'rationale', ''
  desc  'check', "
    Check whether the maximum time period for existing passwords is restricted
to 60 days with the following commands:

    $ sudo awk -F: '$5 > 60 {print $1 \" \" $5}' /etc/shadow

    $ sudo awk -F: '$5 <= 0 {print $1 \" \" $5}' /etc/shadow

    If any results are returned that are not associated with a system account,
this is a finding.
  "
  desc 'fix', "
    Configure non-compliant accounts to enforce a 60-day maximum password
lifetime restriction.

    $ sudo chage -M 60 [user]
  "
  impact 0.5
  tag severity: 'medium'
  tag gtitle: 'SRG-OS-000076-GPOS-00044'
  tag gid: 'V-230367'
  tag rid: 'SV-230367r627750_rule'
  tag stig_id: 'RHEL-08-020210'
  tag fix_id: 'F-33011r567848_fix'
  tag cci: ['CCI-000199']
  tag nist: ['IA-5 (1) (d)']

  shadow.users.each do |user|
    # filtering on non-system accounts (uid >= 1000)
    next unless user(user).uid >= 1000
    describe shadow.users(user) do
      its('max_days.first.to_i') { should cmp <= 60 }
      its('max_days.first.to_i') { should cmp > 0 }
    end
  end
end
```

Future Work

- **Post Conference Activities**
  - Merge current base PRs
  - Stabilize schedule of the OHDF TC meetings
  - Iterate to OHDF v1.0 (additional data elements)

- **Community & Contribution Goals**
  - Handle more types of cybersecurity data (e.g., SBOM, event-log data)
  - Specification additions & community engagement

Issues, Discussions, and Pull Requests Welcome!

MITRE

Questions?

MITRE SAF ©	https://saf.mitre.org/#/
MITRE Heimdall © (and other libraries)	https://github.com/mitre/heimdall2
MITRE SAF CLI	https://github.com/mitre/saf
OHDF Technical Committee	https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ohdf
Sophos	https://www.sophos.com/en-us/products/sophos-factory
MITRE SAF © Email	saf@groups.mitre.org
Aaron Lippold	alippold@mitre.org
Mike Fraser	Mike.Fraser@Sophos.com

MITRE