Forward-Looking Statements

Deploying Detection as Code at Scale

SEC1847A

Nate Zastrow

Senior Detection Engineer | Northwestern Mutual

splunk> .conf23

Nate Zastrow

Senior Detection Engineer
Northwestern Mutual

What is Detection as Code?

What is Detection as Code?

Apply the tools and best practices from software development to detection content development.

What is Detection as Code?

Apply the tools and best practices from software development to detection content development.

Four Principles:

What is Detection as Code?

Apply the tools and best practices from software development to detection content development.

Four Principles:

- Detection Language

What is Detection as Code?

Apply the tools and best practices from software development to detection content development.

Four Principles:

- Detection Language
- Version Control System

What is Detection as Code?

Apply the tools and best practices from software development to detection content development.

Four Principles:

- Detection Language
- Version Control System
- Automated Workflows

splunk> .conf23

What is Detection as Code?

Apply the tools and best practices from software development to detection content development.

Four Principles:

- Detection Language
- Version Control System
- Automated Workflows
- Test-driven Development

Why implement DaC?

Why implement DaC?

- Detections are already code

Why implement DaC?

- Detections are already code
- Ensure thorough detection validation
- Built-in change control
- Increase visibility of changes

- Improve documentation
- Increase development velocity
  – Easier discoverability
  – More confidence with unit tests and required approvals
  – Less time fixing broken alerts
  – Enable broad find/replace-type changes

The Challenges We Faced

The Challenges We Faced

- **700+** correlation searches
- **600+** properties of a correlation search configurable via API
- **20+** content creators across different teams

The Challenges We Faced

- **700+** correlation searches
- **600+** properties of a correlation search configurable via API
- **20+** content creators across different teams
- Upskilling team members in new tools for DaC

The Challenges We Faced

- **700+** correlation searches
- **600+** properties of a correlation search configurable via API
- **20+** content creators across different teams
- Upskilling team members in new tools for DaC
- Already have an extensive in-house detection validation process to integrate with

The Challenges We Faced

- **700+** correlation searches
- **600+** properties of a correlation search configurable via API
- **20+** content creators across different teams
- Upskilling team members in new tools for DaC
- Already have an extensive in-house detection validation process to integrate with
- Mid-project Splunk® Cloud migration

Tool Selection

Tool Selection

YAML - Detection Language

- Human and machine readable
- Syntax is relatively easy to learn
- Supports comments
- Ubiquitous

Docker - Automation

- Already integrated into our git environment
- Very easy to build and iterate on as we mature our use case
- Ubiquitous

Git - Version Control / Automation

- Lots of CI/CD features baked into enterprise versions
- Ubiquitous

Python - Automation

- Splunk maintains a Python SDK so we don't need to write our own middleware
- Ubiquitous

How It Started

How It Started

- First iteration was a simple “pull” sync from Splunk® into git:
  - API call to get current state of all correlation searches
  - Convert results from JSON to YAML
  - Compare with YAML files already in repo
  - Write any changes out
  - Submit merge request for that day’s changes

splunk> .conf23

How It Started

- First iteration was a simple "pull" sync from Splunk® into git:
  - API call to get current state of all correlation searches
  - Convert results from JSON to YAML
  - Compare with YAML files already in repo
  - Write any changes out
  - Submit merge request for that day's changes
- This allowed us to:

How It Started

- First iteration was a simple “pull” sync from Splunk® into git:
  - API call to get current state of all correlation searches
  - Convert results from JSON to YAML
  - Compare with YAML files already in repo
  - Write any changes out
  - Submit merge request for that day’s changes
- This allowed us to:
  - Build and test the scaffolding we needed for the “push” end state

splunk> .conf23

How It Started

- First iteration was a simple “pull” sync from Splunk® into git:
  - API call to get current state of all correlation searches
  - Convert results from JSON to YAML
  - Compare with YAML files already in repo
  - Write any changes out
  - Submit merge request for that day’s changes
- This allowed us to:
  - Build and test the scaffolding we needed for the “push” end state
  - Narrow down which of the hundreds of properties we actually care about

splunk> .conf23

How It Started

- First iteration was a simple "pull" sync from Splunk® into git:
  - API call to get current state of all correlation searches
  - Convert results from JSON to YAML
  - Compare with YAML files already in repo
  - Write any changes out
  - Submit merge request for that day's changes
- This allowed us to:
  - Build and test the scaffolding we needed for the "push" end state
  - Narrow down which of the hundreds of properties we actually care about
  - Provide basic change tracking through daily merge requests

How It Started

- First iteration was a simple “pull” sync from Splunk® into git:
  - API call to get current state of all correlation searches
  - Convert results from JSON to YAML
  - Compare with YAML files already in repo
  - Write any changes out
  - Submit merge request for that day’s changes
- This allowed us to:
  - Build and test the scaffolding we needed for the “push” end state
  - Narrow down which of the hundreds of properties we actually care about
  - Provide basic change tracking through daily merge requests
  - Prove the concept to leadership

How It’s Going

How It's Going

- All production detections managed via GitOps
- Mandatory review required for all changes to prod
- MR template checklist ensures consistent and thorough review of changes

How It's Going

- All production detections managed via GitOps
- Mandatory review required for all changes to prod
- MR template checklist ensures consistent and thorough review of changes
- Full CI/CD pipeline with:
  - Linting
  - SPL safety checks
  - Unit tests
  - Deploy to Splunk® ES via Splunk® SDK
  - Automated documentation via git pages

splunk> .conf23

How It's Going

- All production detections managed via GitOps
- Mandatory review required for all changes to prod
- MR template checklist ensures consistent and thorough review of changes
- Full CI/CD pipeline with:
  - Linting
  - SPL safety checks
  - Unit tests
  - Deploy to Splunk® ES via Splunk® SDK
  - Automated documentation via git pages

100+ merges per month / 1,000+ total

splunk> .conf23

“I love peer review.”

- Anonymous Engineer

Demo

Where We're Going From Here

Where We're Going From Here

- Fully integrate CI/CD with our existing detection validation process
- Automate more of the MR checklist (SPL linting, improved dynamic analysis)
- Bring in content creators outside of the security team
- Expand the scope to other Splunk® content (macros, lookups, dashboards)
- Bring other tools under the detection as code umbrella (endpoint sensor logic, suricata rules, etc.)

What We Learned Along the Way

Thorns

Roses

What We Learned Along the Way

Thorns

- Upskilling is a team-wide lift
- It’s a challenge to keep tooling simple and onboarding easy
- Schema updates can be a chore
- Running “push” and “pull” syncs simultaneously can get interesting

Roses

What We Learned Along the Way

Thorns

- Upskilling is a team-wide lift
- It’s a challenge to keep tooling simple and onboarding easy
- Schema updates can be a chore
- Running “push” and “pull” syncs simultaneously can get interesting

Roses

- Enforced and automated documentation
- Enforced alert output standards
- Custom tagging
- Automated testing
- Catching all sorts of issues before they hit prod
- Team-wide visibility of changes
- Large scale updates

Parting Advice

Parting Advice

- Start small

Parting Advice

- Start small
- Get your content creators on board up front

Parting Advice

- Start small
- Get your content creators on board up front
- Bring in outside stakeholders early:
  - Incident Response
  - Red Team
  - Threat Intel
  - Upper Management

Parting Advice

- Start small
- Get your content creators on board up front
- Bring in outside stakeholders early:
  - Incident Response
  - Red Team
  - Threat Intel
  - Upper Management
- Put a lot of thought into your definition language schema

Parting Advice

- Start small
- Get your content creators on board up front
- Bring in outside stakeholders early:
  - Incident Response
  - Red Team
  - Threat Intel
  - Upper Management
- Put a lot of thought into your definition language schema
- Make it *(as)* easy *(as possible)*

Additional Resources

- SEC1197C - Build Detection as Code Like the Splunk Threat Research Team (slides)
- Detection-as-code: Why it works and where to start
- Detectionengineering.io
- Sigma
- Splunk Security Content Repo
- Me! (email | LinkedIn | splunk-usergroups: @Nate Zastrow)

Questions?

Thank You

splunk> .conf23