Cloud Native GRC

Introductions



Jon Zeolla



Brandt Keller

| © 2023 Cloud Native Computing Foundation



Story Time



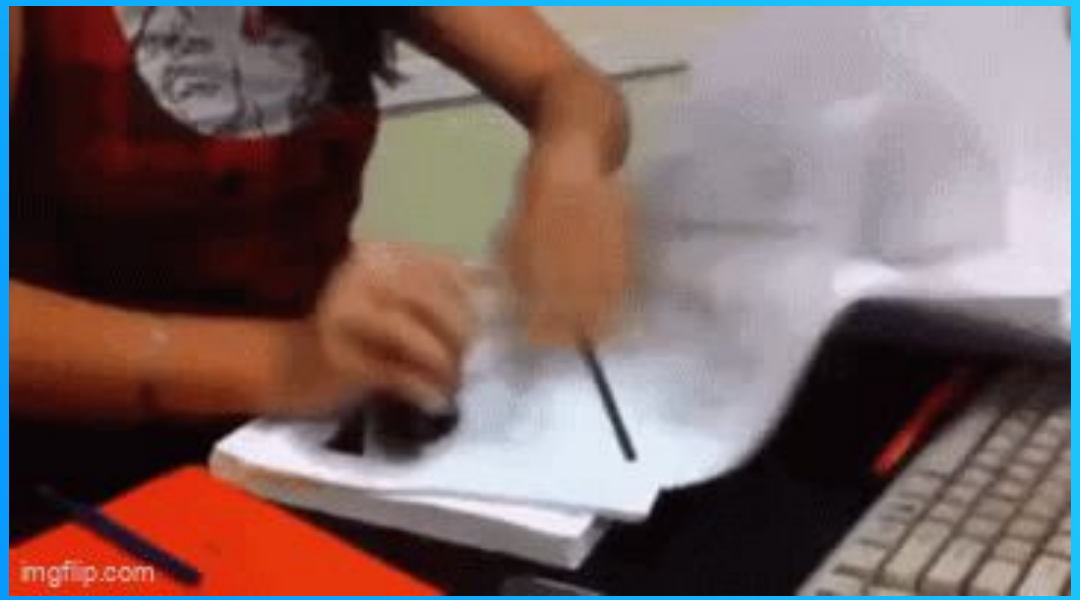
© 2023 Cloud Native Computing Foundation



Your Governance Is Broken

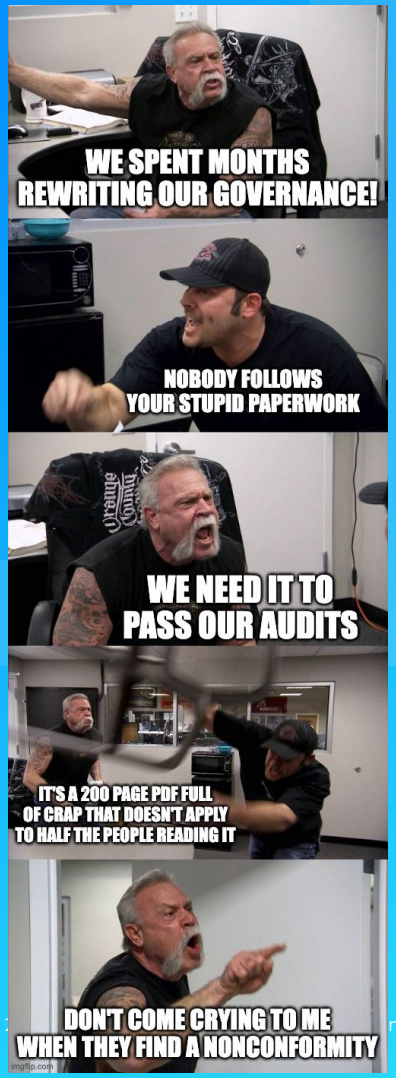
© 2023 Cloud Native Computing Foundation

Employees completing their annual sign-offs on information security policies



Credit: Troy Fine

© 2023 Cloud Native Computing Foundation



Cloud Native GRC

- Assurance via Observability
- Improved Quality
- Automation
- Tight feedback loops

Framing

What is Cloud Native?

“Cloud native practices empower organizations to develop, build, and deploy … to meet their organizational needs at scale in a programmatic and repeatable manner”

https://github.com/cncf/toc/blob/main/DEFINITION.md



© 2023 Cloud Native Computing Foundation

What is GRC? (Wrong answers only)

“GRC stands for Giant Robot Club, which is an exclusive organization dedicated to the hobby of building and battling large, humanoid robots.”

© 2023 Cloud Native Computing Foundation

What is GRC?

“GRC stands for Governance, Risk Management, and Compliance. It refers to a framework or strategy used by organizations to manage their overall governance, enterprise risk management, and compliance with regulations”



© 2023 Cloud Native Computing Foundation

“Is our GRC programmatic and repeatable?”

Barriers to Entry

- Enterprise Platforms
  - Great Option - not universally accessible
  - Underlying Data format (IE Vendor Lock)
- Free / Open Source Software
  - Disparate feature sets
  - Disparate data formats

© 2023 Cloud Native Computing Foundation



Where are we?

- Guard Rails
  - Admission Control
  - Policy
- Tooling
  - Tools that perform GRC functions
  - Tools that meet Controls
- Open Standards
  - OSCAL

© 2023 Cloud Native Computing Foundation



What did we learn?



Use Case: **Lula**

Problems:

- Human Processes & engrained tooling
- Data Exchange
- Expectations & Resistance
- Not Repeatable
- Not Programmatic

Solutions:

- Focus on what matters - iterate - feedback

© 2023 Cloud Native Computing Foundation



Demo



GRC processes, now with more 🤖

CLOUD NATIVE
COMPUTING FOUNDATION



https://landscape.cncf.io/?group=projects-and-products&view-mode=grid&project=cncf&tag=security

Conftest

```
package main

denylist := ["nginx"]

deny[msg] {
  some i
  input[i].Cmd == "from"
  val := input[i].Value
  contains(val[i], denylist[_])
  msg = sprintf("unallowed image found %s", [val])
}
```

https://www.conftest.dev/

© 2023 Cloud Native Computing Foundation



goss/dgoss/kgoss/dcgoss

```
port:
  tcp:22:
    listening: true
    ip:
    - 0.0.0.0
  tcp6:22:
    listening: true
    ip:
    - '::'
service:
  sshd:
    enabled: true
    running: true
user:
  sshd:
    exists: true
    uid: 74
    gid: 74
    groups:
    - sshd
    home: /var/empty/sshd
    shell: /sbin/nologin
group:
  sshd:
    exists: true
    gid: 74
process:
  sshd:
    running: true
```

https://github.com/goss-org/goss/

© 2023 Cloud Native Computing Foundation



Checkov

```
resource "aws_s3_bucket" "foo-bucket" {
#same resource configuration as previous example, but acl set for public access.

  acl           = "public-read"
}
data "aws_caller_identity" "current" {}
```


© 2023 Cloud Native Computing Foundation

https://www.checkov.io/1.Welcome/Quick%20Start.html



Checkov

```
Passed checks: 3, Failed checks: 1, Skipped checks: 0

Check: "Ensure all data stored in the S3 bucket is securely encrypted at rest"
 PASSED for resource: aws_s3_bucket.foo-bucket
 File: /example.tf:1-25


Check: "Ensure the S3 bucket has access logging enabled"
 PASSED for resource: aws_s3_bucket.foo-bucket
 File: /example.tf:1-25


Check: "Ensure all data stored in the S3 bucket have versioning enabled"
 PASSED for resource: aws_s3_bucket.foo-bucket
 File: /example.tf:1-25


Check: "S3 Bucket has an ACL defined which allows public access."
 FAILED for resource: aws_s3_bucket.foo-bucket
 File: /example.tf:1-25

   1 | resource "aws_s3_bucket" "foo-bucket" {
   2 |   region        = var.region
   3 |   bucket        = local.bucket_name
   4 |   force_destroy = true
   5 |
   6 |   tags = {
   7 |     Name = "foo-${data.aws_caller_identity.current.account_id}"
   8 |   }
   9 |   versioning {
  10 |     enabled = true
  11 |   }
```

https://www.checkov.io/1.Welcome/Quick%20Start.html

© 2023 Cloud Native Computing Foundation



InSpec Policies

```
control 'sshd-05' do
  impact 1.0
  title 'Server: Check sshd_config owner, group and permissions.'
  desc 'The sshd_config should owned by root or a defined user, only be writable/readable by owner and not be executable.'

  describe file(sshd_custom_path + '/sshd_config') do
    it { should exist }
    it { should be_file }
    it { should be_owned_by sshd_custom_user }
    it { should be_grouped_into os.darwin? ? 'wheel' : sshd_custom_user }
    it { should_not be_executable }
    it { should be_readable.by('owner') }
    it { should_not be_readable.by('group') }
    it { should_not be_readable.by('other') }
    it { should be_writable.by('owner') }
    it { should_not be_writable.by('group') }
    it { should_not be_writable.by('other') }
  end
end
```


© 2023 Cloud Native Computing Foundation

https://github.com/inspec/inspec



InSpec Policies

```
control 'ssh-08' do
  impact 1.0
  title 'Client: Check for secure ssh ciphers'
  desc 'Configure a list of ciphers to the best secure ciphers (avoid older and weaker ciphers)'
  describe ssh_config(ssh_custom_path + '/ssh_config') do
    its('Ciphers') { should eq(ssh_crypto.valid_ciphers) }
  end
end
```


© 2023 Cloud Native Computing Foundation

https://github.com/inspec/inspec



InSpec Policies

```
control 'ssh-16' do
  impact 1.0
  title 'Client: Disable password-based authentication'
  desc 'Avoid password-based authentications.'
  describe ssh_config(ssh_custom_path + '/ssh_config') do
    its('PasswordAuthentication') { should eq('no') }
  end
end
```


© 2023 Cloud Native Computing Foundation

https://github.com/inspec/inspec



BLOG / COMMUNITY POST

Introduction to the Cloud Native Security Controls Catalog

Posted on June 7, 2022

Community post by Jon Zeolla, CTO and Co-Founder of Seiso

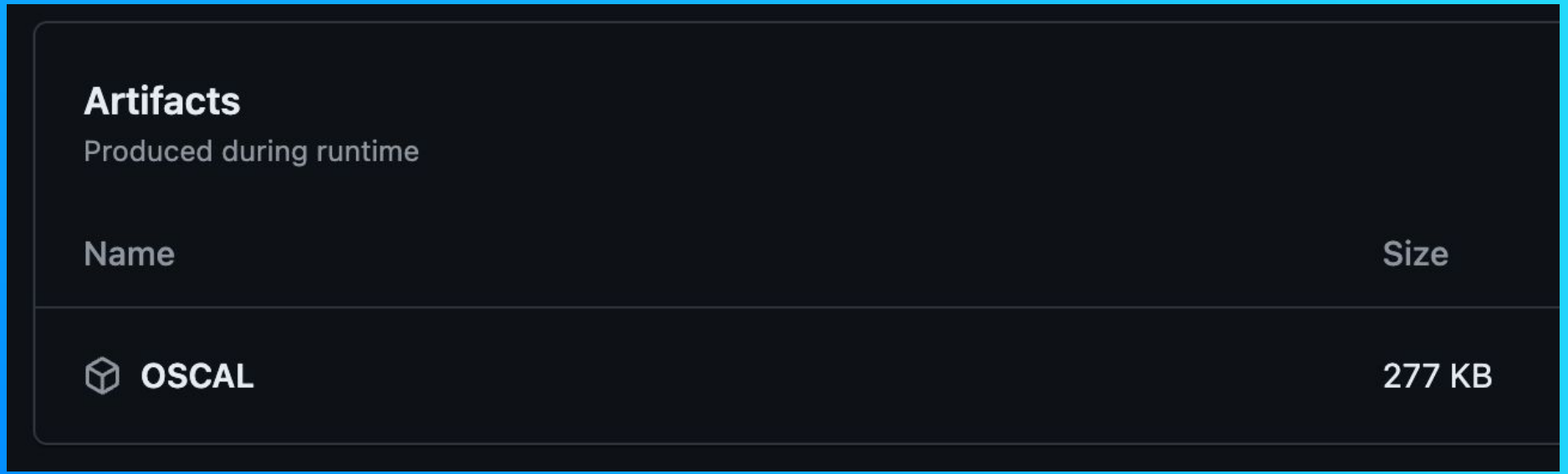
The CNCF Security Technical Advisory Group (“Security TAG”) has provided a wealth of information to assist organizations in the planning and design of secure cloud native systems, including the **Cloud Native Security Whitepaper** (CNSWP) and the **Software Supply Chain Best Practices Paper** (SSCP). Organizations would like to self-assess their company’s alignment with these materials in a clear, concrete way, such as via checklists or automation.

© 2023 Cloud Native Computing Foundation

https://github.com/cncf/tag-security/blob/5053f4b0dcb48f14da73cd3909f0ed53f560bfb7/cloud-native-controls/phase-one-announcement.md

Cloud Native Security Controls

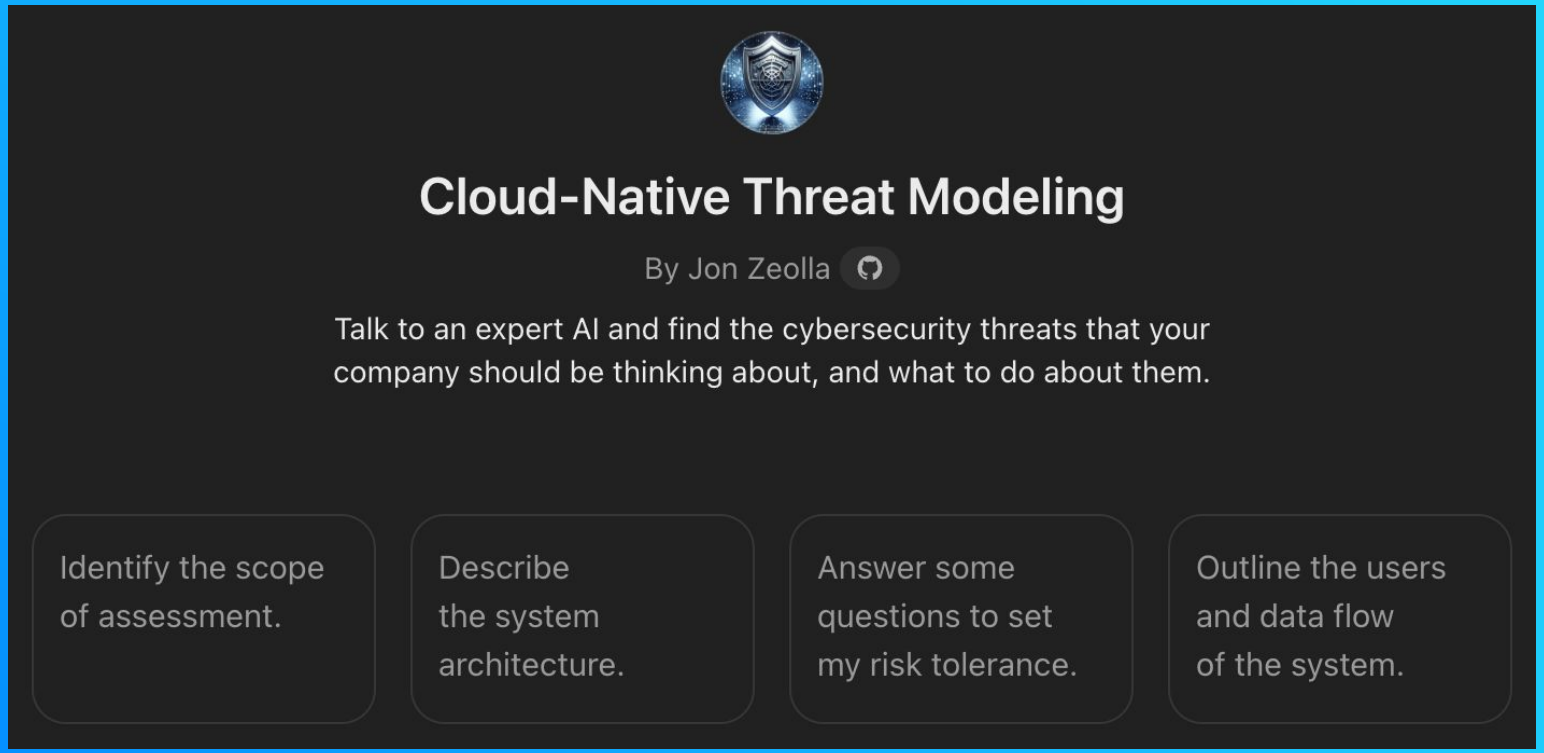
Machine Readable Formats (OSCAL)



© 2023 Cloud Native Computing Foundation

https://github.com/cloud-native-security-controls/controls-catalog/actions/runs/9097151258

Risk and Threat Modeling



https://zenable.io/threats
https://github.com/JonZeolla/threat-modeling-custom-gpt

© 2023 Cloud Native Computing Foundation



Now what?

Questions?

© 2023 Cloud Native Computing Foundation

