Level Up Your CI/CD: Building a secure pipeline with OSS

Slides

Repo
Fork it and clone it

https://github.com/unicrons/secure-pipeline-workshop

Level Up Your CI/CD: Building a secure pipeline with OSS

SANS

CloudSecNext SUMMIT 2025

sans.org/CloudSecNextSummit

unicrons.cloud

About us

Edu

Software Engineer @ flywire

@eduardoSimon

eduardo-simon

Andoni

Cloud Security Engineer @ PROWLER

@andoniaf

andoniaf

And he's Paco

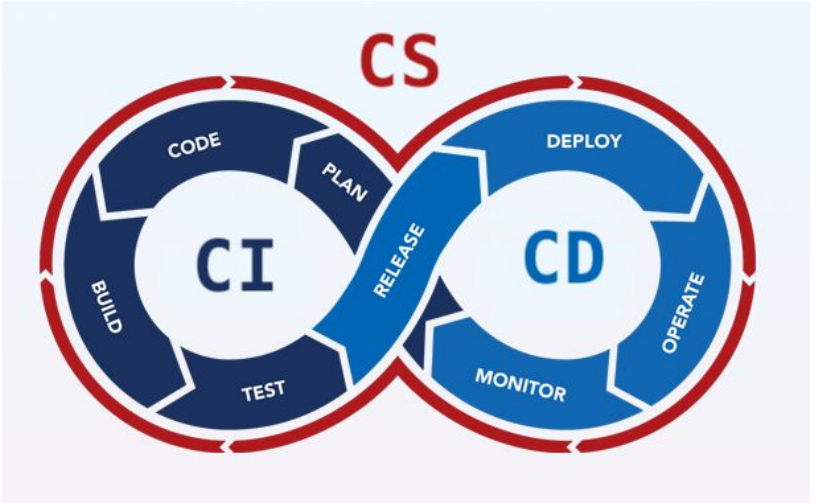
Agenda

1. Introduction
    - What is CI / CD /CS ?
    - The need for secure pipeline

2. The “perfect pipeline” overview

3. Workshop:
    - Brief explanation of the step
    - Integrate OSS tooling
    - Fix the issue

What is CI / CD / CS?



Continuous Integration (CI)

- Developers merge code multiple times a day
- Every merge triggers an automated build and test

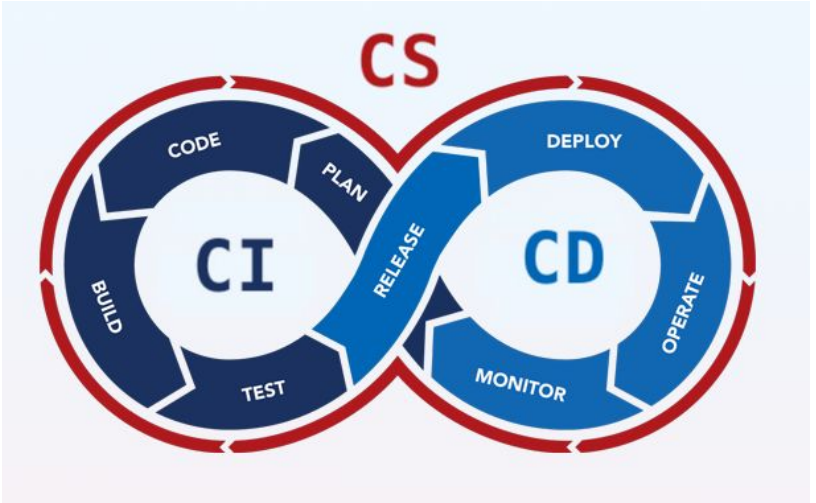
Continuous Delivery (CD)

- Every change can be deployed to production
- “Release day” no longer exists

Continuous Security (CS)

- Embedding security into the process
- Automating security controls and tests throughout the pipeline (like we do for QA)
- Find vulnerabilities when they are cheapest and easiest to fix

What is CI / CD / CS?



Continuous Integration (CI)

- Developers merge code multiple times a day
- Every merge triggers an automated build and test

Continuous Delivery (CD)

- Every change can be deployed to production
- “Release day” no longer exists

Continuous Security (CS)

- Embedding security into the process
- Automating security controls and tests throughout the pipeline (like we do for QA)
- **Find vulnerabilities when they are cheapest and easiest to fix**

Why is pipeline security important?

Why add security steps to the pipeline?

- Prevents supply chain compromises (e.g., malware injection)
- Enables early threat detection (SAST/DAST scans, secret checks)
- Reduces risks in automated environments (leaked keys, backdoors)
- Ensures compliance & builds trust

Why is pipeline security important?

But my repo is private...

...until is not.

Twitch breach leads to leak of source code and streamer earnings data

John Leyden 07 October 2021 at 14:28 UTC

More than 20GB of Intel source code and proprietary data dumped online

"Exconfidential Lake" leak includes docs Intel provided under NDA as recently as May.

DAN GOODIN AND JIM SALTER – AUG 6, 2020 4:59 PM | 150

Toyota source code exposed for five years, impacts 300,000 drivers

Dashveenjit Kaur October 13, 2022

Mercedes-Benz Source Code at Risk: GitHub Token Mishap Sparks Major Security Concerns

Lohit January 29, 2024

Samsung and Nvidia are the latest companies to involuntarily go open-source leaking company secrets

The Microsoft source code breach may be much bigger than we thought

News By Anthony Spadafora published March 22, 2022

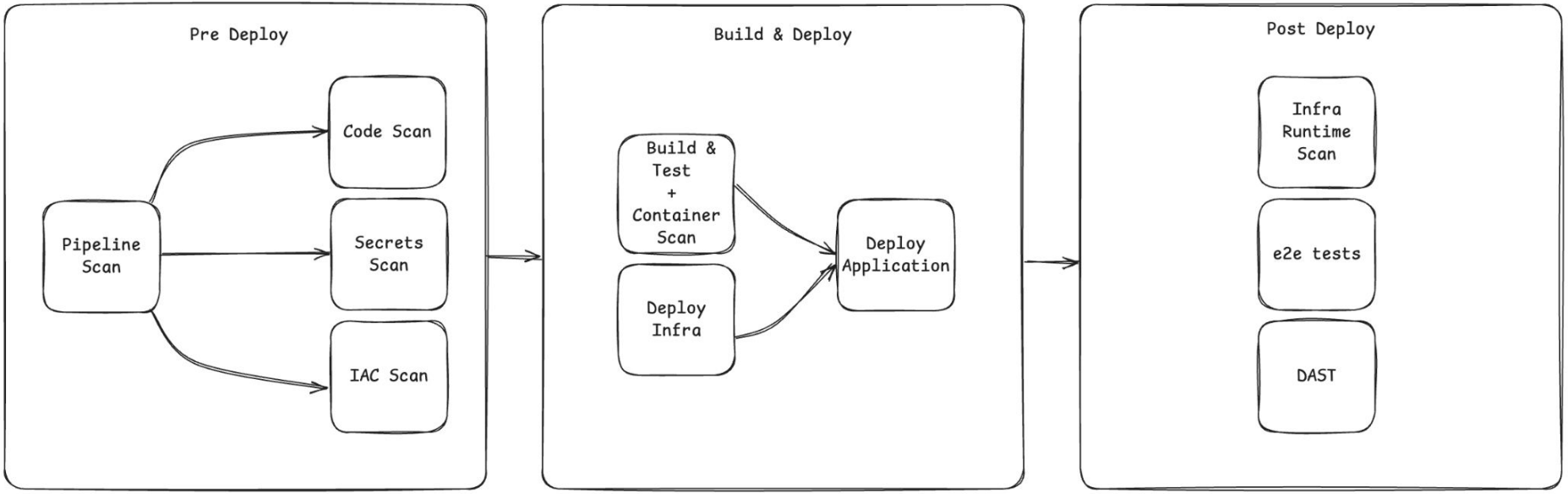
tj-actions changed-files through 45.0.7 allows remote attackers to discover secrets by reading actions logs.

High severity GitHub Reviewed Published on Mar 14 to the GitHub Advisory Database • Updated on Mar 24

NPM BREACH EXPOSES TOPTAL GITHUB REPOSITORIES IN MAJOR JULY 2025 ATTACK

by CertPro Digest | Jul 24, 2025 | Data |

The “perfect pipeline” overview



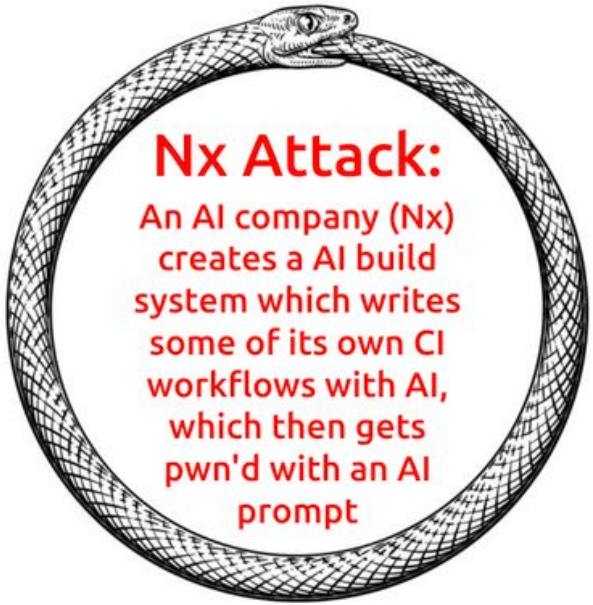
The “perfect pipeline” overview

WHERE IS THE AI?!?

No AI today

- Not deterministic
- Expensive
- Lack of Auditability and Explainability

WHERE IS THE AI?!?



Adan Álvarez Vilchez · 1st
Principal Security Engineer | AWS Community Builder | Building Trust. Re...
3d ·

Did an AI code review miss the injection vulnerability that led to the NX supply chain attack?

Analyzing the root cause of the NX incident, I immediately thought about the topic I've been discussing lately: LLMs for code reviews.

If I'm not mistaken, this PR introduced the injection vulnerability:
- https://lnkd.in/dNgvN59J

The PR seems to have gone through an "AI code review" but no problems were detected.

I did a quick test running this code through Semgrep GitHub Action rules and Semgrep rules do seem to catch it
- https://lnkd.in/dnUQVeU2

Interestingly, in the results from my tests, LLMs seemed to perform worse when analyzing things like GitHub Actions:
- https://lnkd.in/dBm4GNeX

As I've mentioned in previous posts, this is not about dismissing LLMs. They are powerful and improving quickly. What I keep seeing is that combining static analysis with LLMs feels like a stronger approach than relying on just one.

WHERE IS THE AI?!?

No AI today

- Not deterministic
- Expensive
- Lack of Auditability and Explainability

But probably AI tomorrow...

- Fine-tuned small language model can outperform traditional regex-based methods
- Better context for findings (and hopefully less false positives)

AI Example

test: ai review #32

Merged **sanchezpaco** merged 1 commit into `main` from `feat-gemini` 1 hour ago

Conversation 5 | Commits 1 | Checks 10 | Files changed 1

sanchezpaco commented 1 hour ago — Collaborator

No description provided.

sanchezpaco commented 1 hour ago — Collaborator, Author

/ai-review

🚀 1

github-actions bot commented 1 hour ago

🤖 AI Security Review Results

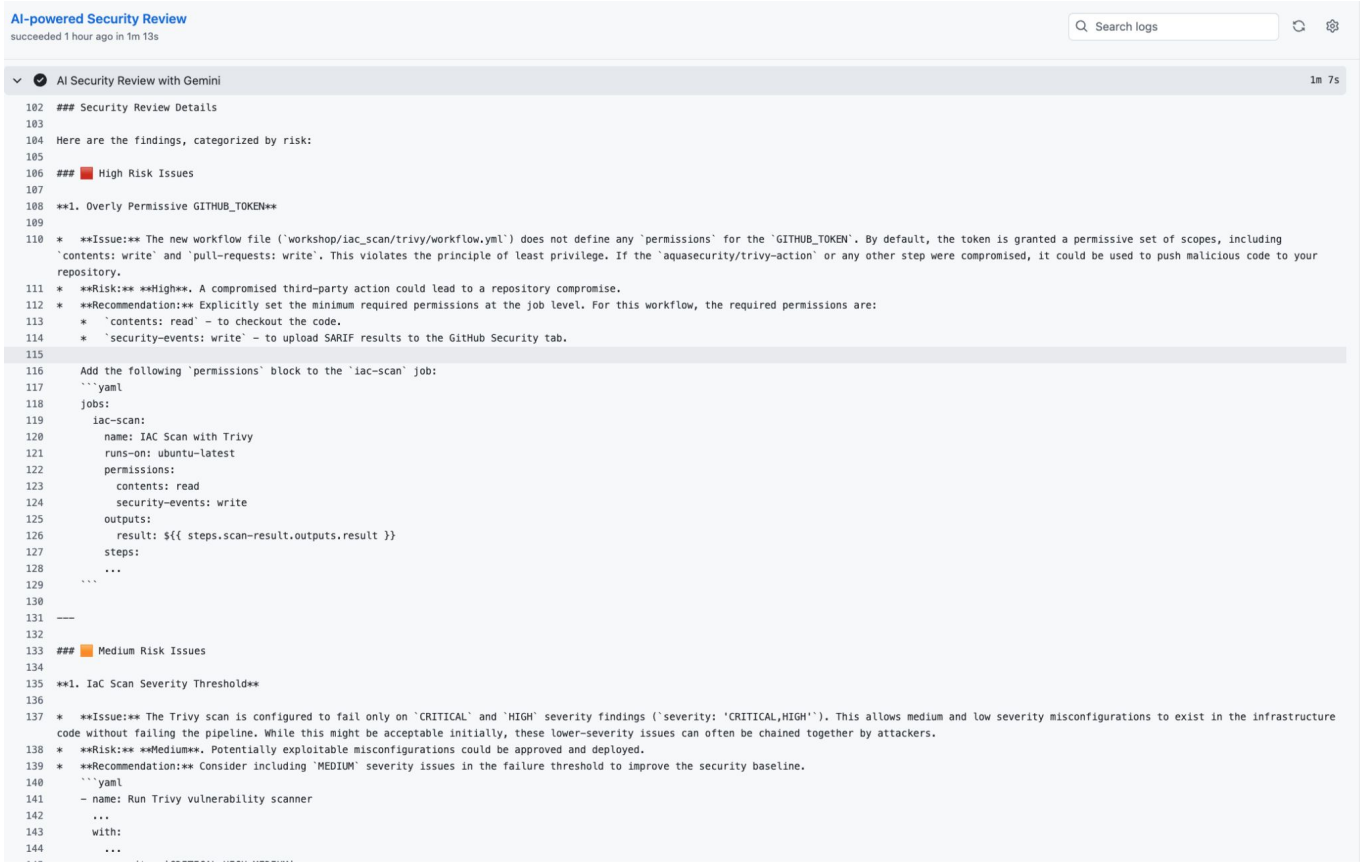
The AI security scan has been completed! 🚀

Triggered by: @sanchezpaco
Command: `/ai-review`

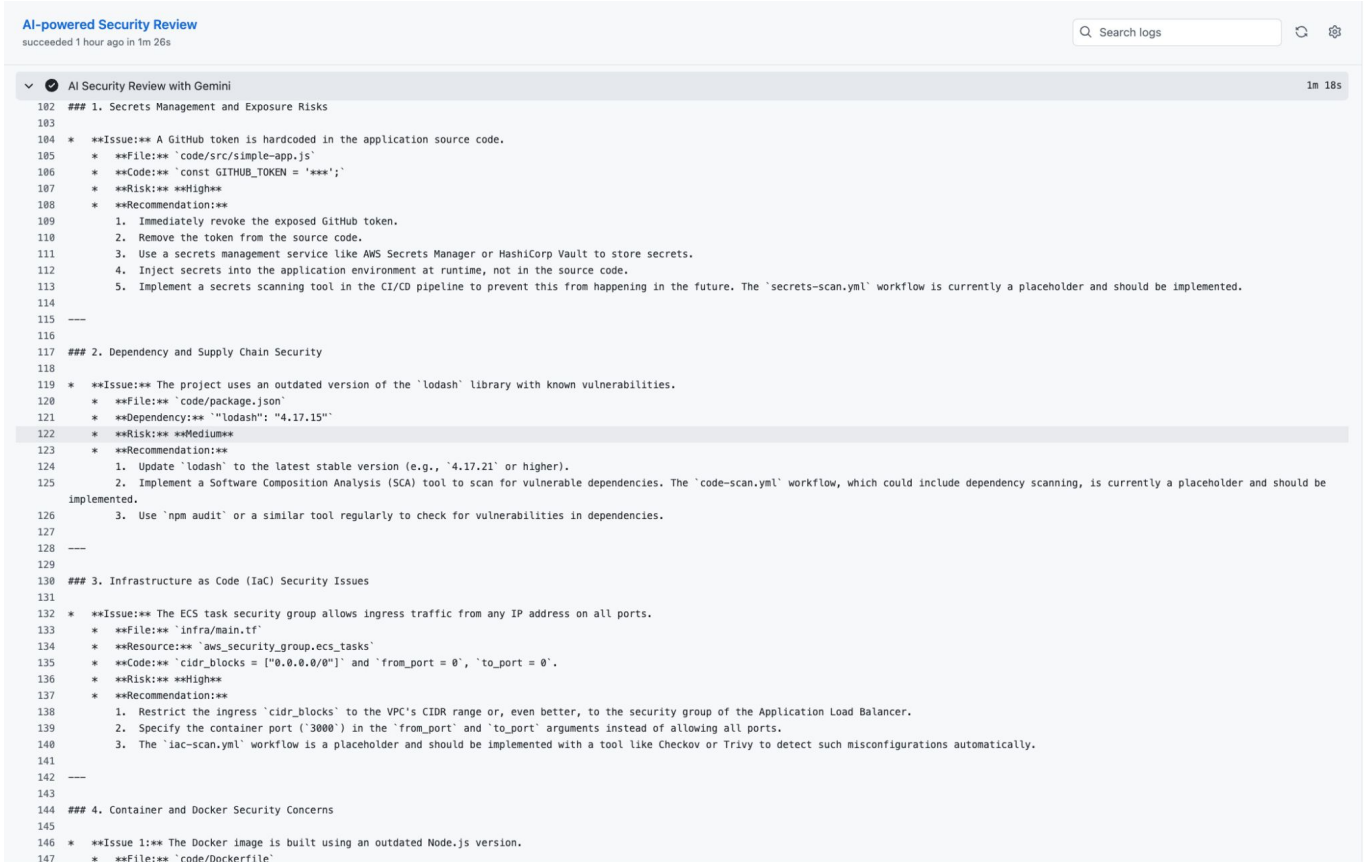
> 💡 **Note:** Detailed analysis is currently logged in the workflow. Soon we'll display the full review here!

Check the workflow run for complete details.

AI Example Outputs



AI Example Outputs




```
### 1. Secrets Management and Exposure Risks

*   **Issue:** A GitHub token is hardcoded in the application source code.
    *   **File:** `code/src/simple-app.js`
    *   **Code:** `const GITHUB_TOKEN = '***';`
    *   **Risk:** **High**
    *   **Recommendation:**
        1. Immediately revoke the exposed GitHub token.
        2. Remove the token from the source code.
        3. Use a secrets management service like AWS Secrets Manager or HashiCorp Vault to store secrets.
        4. Inject secrets into the application environment at runtime, not in the source code.
        5. Implement a secrets scanning tool in the CI/CD pipeline to prevent this from happening in the future. The `secrets-scan.yml` workflow is currently a placeholder and should be implemented.

---

### 2. Dependency and Supply Chain Security

*   **Issue:** The project uses an outdated version of the `lodash` library with known vulnerabilities.
    *   **File:** `code/package.json`
    *   **Dependency:** `"lodash": "4.17.15"`
    *   **Risk:** **Medium**
    *   **Recommendation:**
        1. Update `lodash` to the latest stable version (e.g., `4.17.21` or higher).
        2. Implement a Software Composition Analysis (SCA) tool to scan for vulnerable dependencies. The `code-scan.yml` workflow, which could include dependency scanning, is currently a placeholder and should be implemented.
        3. Use `npm audit` or a similar tool regularly to check for vulnerabilities in dependencies.

---

### 3. Infrastructure as Code (IaC) Security Issues

*   **Issue:** The ECS task security group allows ingress traffic from any IP address on all ports.
    *   **File:** `infra/main.tf`
    *   **Resource:** `aws_security_group.ecs_tasks`
    *   **Code:** `cidr_blocks = ["0.0.0.0/0"]` and `from_port = 0`, `to_port = 0`.
    *   **Risk:** **High**
    *   **Recommendation:**
        1. Restrict the ingress `cidr_blocks` to the VPC's CIDR range or, even better, to the security group of the Application Load Balancer.
        2. Specify the container port (`3000`) in the `from_port` and `to_port` arguments instead of allowing all ports.
        3. The `iac-scan.yml` workflow is a placeholder and should be implemented with a tool like Checkov or Trivy to detect such misconfigurations automatically.

---

### 4. Container and Docker Security Concerns

*   **Issue 1:** The Docker image is built using an outdated Node.js version.
    *   **File:** `code/Dockerfile`
```

Important stuff that we've left out...

- pre-commit and other git hooks
- Github deep dive
- A lot of very good tools...

Before the Hands-On...
Questions?

How are we going to run the workshop?

First of all you need to fork this repo:

https://github.com/unicrons/secure-pipeline-workshop

Then, for each step:

1. Brief step explanation
2. Enable the job:
    - Copy the `workflow.yml`
    - Open a PR in your fork to trigger the pipeline
3. Leave you some time to solve the finding
4. Show results to everyone
5. Questions
6. Go to the next

Workshop: Pipeline Security

Why is Pipeline Security Important?

- Baseline security of our SDLC (Software Development Life Cycle)
- Elevated access:
  - to secrets
  - to code
  - to infra

Common Pipeline Security Issues

- Hardcoded Secrets
- Excessive Permissions
- Untrusted Actions
- Insecure Triggers
- Missing Security Controls

Workshop: Pipeline Security

Extra ball: Warden's Ruse

```
…/SOLVED/wardens/Wardens-Ruse-main
 ls -laR
drwxrwxr-x@    - Andoni.Alonso 15 jul 14:04 .github
.rw-rw-r--@ 1,4k Andoni.Alonso 15 jul 14:04 main.tf
.rw-rw-r--@  286 Andoni.Alonso 15 jul 14:04 README.md
.rw-rw-r--@ 174k Andoni.Alonso 15 jul 14:04 repo-visibili
.rw-rw-r--@ 465k Andoni.Alonso 15 jul 14:04 Warden.png

./.github:
drwxrwxr-x@ - Andoni.Alonso 15 jul 14:04 workflows

./.github/workflows:
.rw-rw-r--@ 829 Andoni.Alonso 10 ago 21:56 apply-prod.yaml

…/SOLVED/wardens/Wardens-Ruse-main
 cat README.md | grep TODO
TODO: Lock up once Haisha finishes setting up dc32-wardens-treasure-prod
and <https://d2azf0l1i0s26w.cloudfront.net/>.
```

Challenge | 7 Solves

Warden's Ruse

540

In the enchanted realm of Cybershire, a powerful artifact lies hidden within the mystical Vault of Secrets. This vault, guarded by the vigilant Warden, is fortified with ancient spells, allowing only the chosen ones to pass. However, a flaw in the Warden's enchantment whispers through the winds. Brave adventurers, equipped with their wits and determination, must decipher the cryptic trust sigils, bypass the Warden's watchful eye, and retrieve the secret from Vault. Only the most cunning will succeed in this quest.

Workshop: Pipeline S

Apparently to be used for deploys, using Github Actions

Something suspicious?

```
resource "aws_iam_role" "warden" {
  name               = "warden-production"
  assume_role_policy = data.aws_iam_policy_document.warden-role.json
}

data "aws_iam_policy_document" "warden-role" {
  statement {
    actions = ["sts:AssumeRoleWithWebIdentity"]

    principals {
      identifiers = [aws_iam_openid_connect_provider.github.arn]
      type        = "Federated"
    }

    condition {
      test     = "StringEquals"
      values   = ["sts.amazonaws.com"]
      variable = "token.actions.githubusercontent.com:aud"
    }

    condition {
      test     = "StringLike"
      values   = ["repo:*/Wardens-Ruse:ref:refs/heads/endlessendurance"]
      variable = "token.actions.githubusercontent.com:sub"
    }

    condition {
      test     = "StringEquals"
      values   = ["private"]
      variable = "token.actions.githubusercontent.com:repository_visibility"
    }
  }
}
```

```
    condition {
      test     = "StringEquals"
      values   = ["sts.amazonaws.com"]
      variable = "token.actions.githubusercontent.com:aud"
    }

    condition {
      test     = "StringLike"
      values   = ["repo:*/Wardens-Ruse:ref:refs/heads/endlessendurance"]
      variable = "token.actions.githubusercontent.com:sub"
    }

    condition {
      test     = "StringEquals"
      values   = ["private"]
      variable = "token.actions.githubusercontent.com:repository_visibility"
    }
  }
}
```

Workshop: Pipeline Security

```
condition {
  test     = "StringLike"
  values   = ["repo:*/Wardens-Ruse:ref:refs/heads/endlessendurance"]
  variable = "token.actions.githubusercontent.com:sub"
}
```

A repository

from “*”

named “Wardens-Ruse

using the branch “endlessendurance”

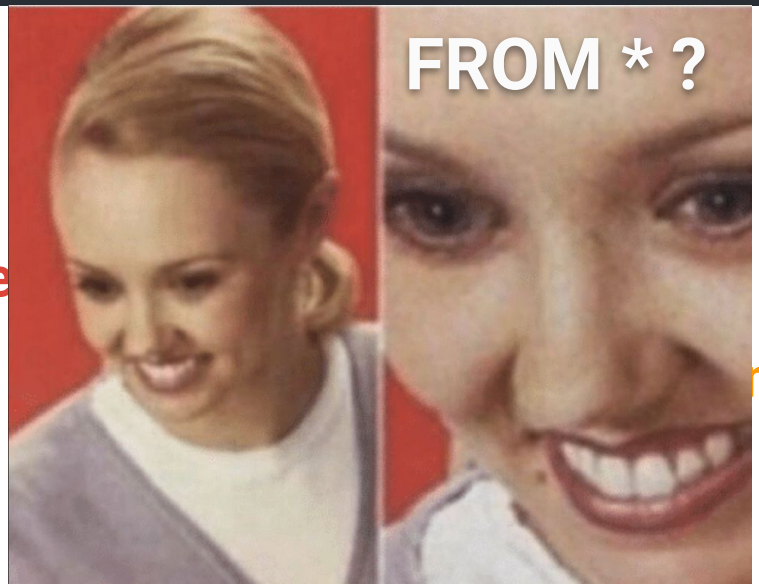
Workshop: Pipeline Security

```
condition {
  test     = "StringLike"
  values   = ["repo:*/Wardens-Ruse:ref:refs/heads/endlessendurance"]
  variable = "token.actions.githubusercontent.com:sub"
}
```

A repository

from “*”

name



ndlessendurance”

Workshop: Pipeline Security

Enable Pipeline Security Step

1. Go to `workshop/pipeline_scan`
2. Choose: `zizmor` or `claws`
3. Go to the tool folder and follow the `workflow.yml` instructions
4. Push the changes and create a PR against your main branch (not Unicrons)
5. Wait for the pipeline results
6. Try to solve them!

(Don’t worry, we will show the results later.)

Workshop: Pipeline Security (Results)

Zizmor

unpinned action reference

In pull request in `refs/pull/2/merge` 5 days ago

`.github/workflows/pipeline-scan.yml:25`

```
22            persist-credentials: false
23
24        - name: Run zizmor 🌈
25          uses: zizmorcore/zizmor-action@main
```

unpinned action reference

zizmor

```
26          with:
27            min-severity: "high"
```

```yaml
steps:
  - name: Checkout repository
    uses: actions/checkout@11bd71901bbe5b1630ceea73d27597364c9af683 # v4.2.2
    with:
      persist-credentials: false

  - name: Run zizmor 🌈
    uses: zizmorcore/zizmor-action@f52a838cfabf134edcbaa7c8b3677dde20045018 # v0.1.1
    with:
      min-severity: "high"
```

Claws

```
🚀 Running Claws security analysis...
Violation: UnpinnedAction on .github/workflows/pipeline-scan.yml:23
All reusable actions must be pinned to a full sha1 commit hash.
For more information:
https://github.com/betterment/claws/blob/main/README.md#unpinnedaction
      - name: Set Up Ruby
>>>         uses: ruby/setup-ruby@master
        with:
          ruby-version: '3.3'
      - name: Set Up Claws Config
```

```yaml
  - name: Set Up Ruby
    uses: ruby/setup-ruby@2a7b30092b0caf9c046252510f9273b4875f3db9 #v1.254.0
    with:
      ruby-version: '3.3'
```

Workshop: Pipeline Security

Extra ball: Warden’s Ruse

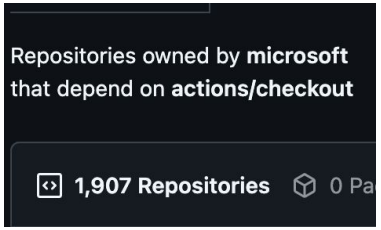

```
./workshop/pipeline_scan/extra/WardenRuseReturns
```

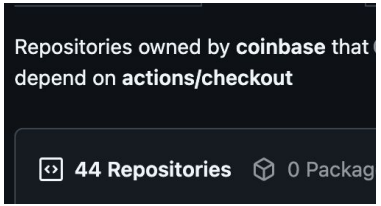
This time the Admin has been studying the Github Actions OIDC token and has added some new conditions to the policy to make it more secure...

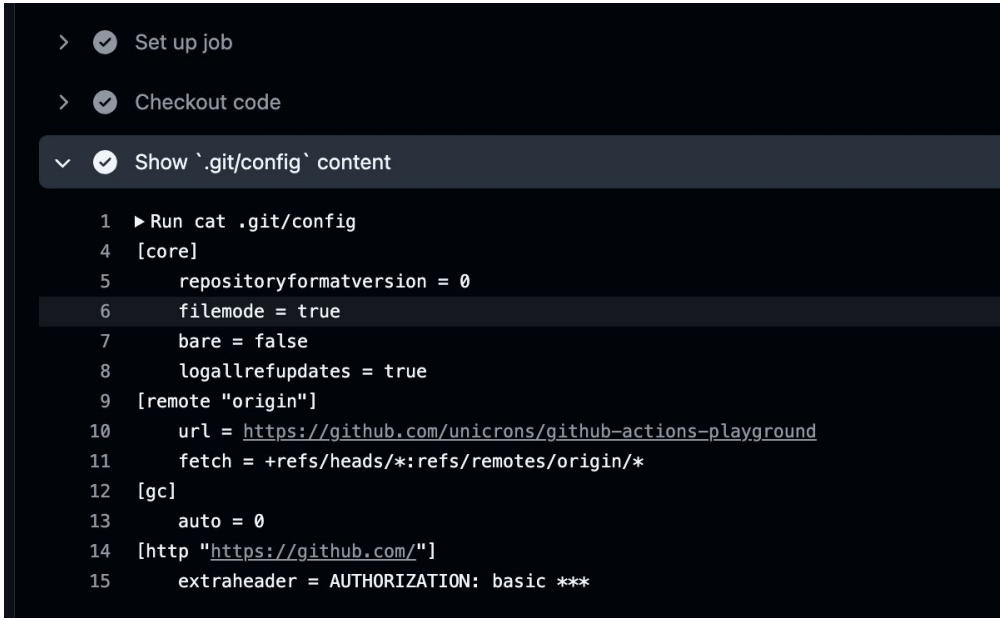
Workshop: Pipeline Security

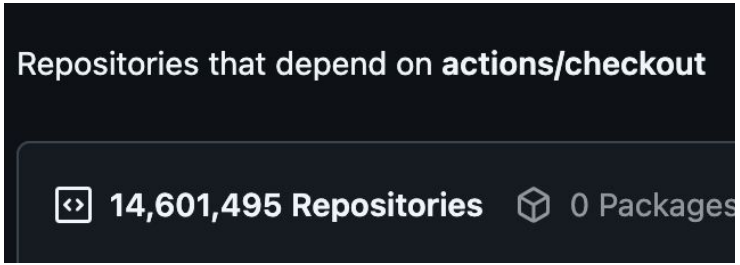
Extra ball: `actions/checkout` can leak Github Credentials

`./workshop/pipeline_scan/extra/checkoutLeak`







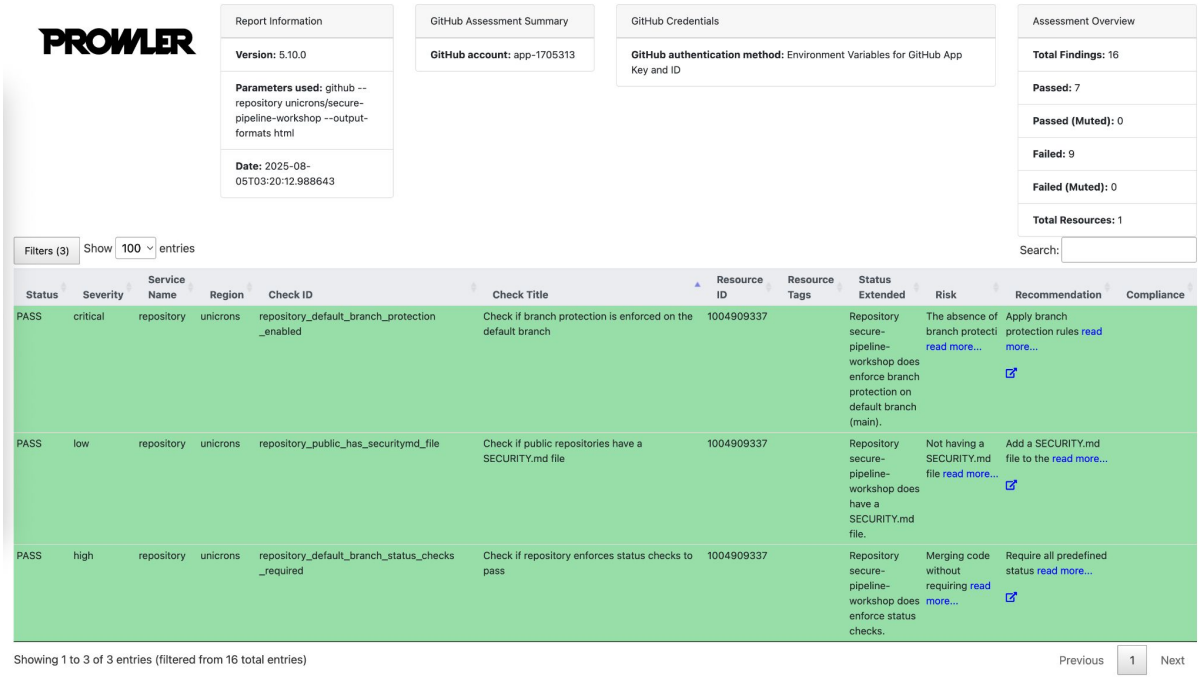


https://yossarian.net/til/post/actions-checkout-can-leak-github-credentials/

Workshop: Pipeline Security

Extra ball: Prowler Github Scanning

```
./workshop/pipeline_scan/extra/prowler
```



PROWLER

Report Information	GitHub Assessment Summary	GitHub Credentials	Assessment Overview
Version: 5.10.0	**GitHub account:** app-1705313	**GitHub authentication method:** Environment Variables for GitHub App Key and ID	**Total Findings:** 16
Parameters used: github --repository unicrons/secure-pipeline-workshop --output-formats html			**Passed:** 7
Date: 2025-08-05T03:20:12.988643			**Passed (Muted):** 0
			Failed: 9
			Failed (Muted): 0
			Total Resources: 1

Filters (3) Show 100 entries Search:

Status	Severity	Service Name	Region	Check ID	Check Title	Resource ID	Resource Tags	Status Extended	Risk	Recommendation	Compliance
PASS	critical	repository	unicrons	repository_default_branch_protection_enabled	Check if branch protection is enforced on the default branch	1004909337		Repository secure-pipeline-workshop does enforce branch protection on default branch (main).	The absence of branch protecti read more...	Apply branch protection rules read more...	
PASS	low	repository	unicrons	repository_public_has_securitymd_file	Check if public repositories have a SECURITY.md file	1004909337		Repository secure-pipeline-workshop does have a SECURITY.md file.	Not having a SECURITY.md file read more...	Add a SECURITY.md file to the read more...	
PASS	high	repository	unicrons	repository_default_branch_status_checks_required	Check if repository enforces status checks to pass	1004909337		Repository secure-pipeline-workshop does enforce status checks.	Merging code without requiring read more...	Require all predefined status read more...	

Showing 1 to 3 of 3 entries (filtered from 16 total entries) Previous 1 Next

Questions?

Workshop: Secrets Scan

Why is Secrets Scan Important?

- Data breaches
- Access to our systems
- Supply Chain Attacks

→

- Compliance Issues
- Financial Loss
- Reputation Damage

Common Leaked Secrets

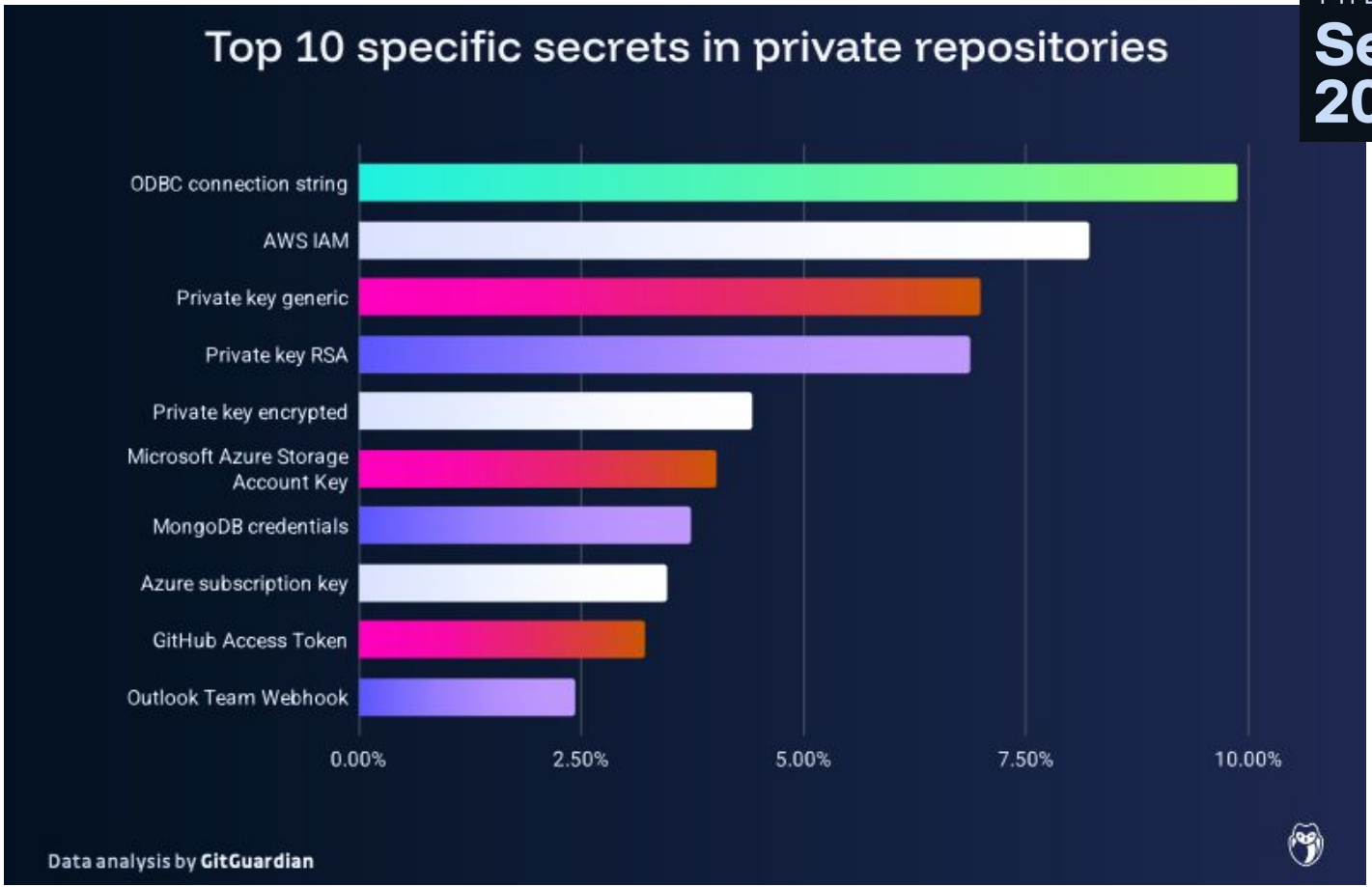
- Hardcoded passwords
- Database Connection Strings
- Cloud Provider Keys
- 3rd Party API Keys
- Private keys

Workshop: Secrets Scan

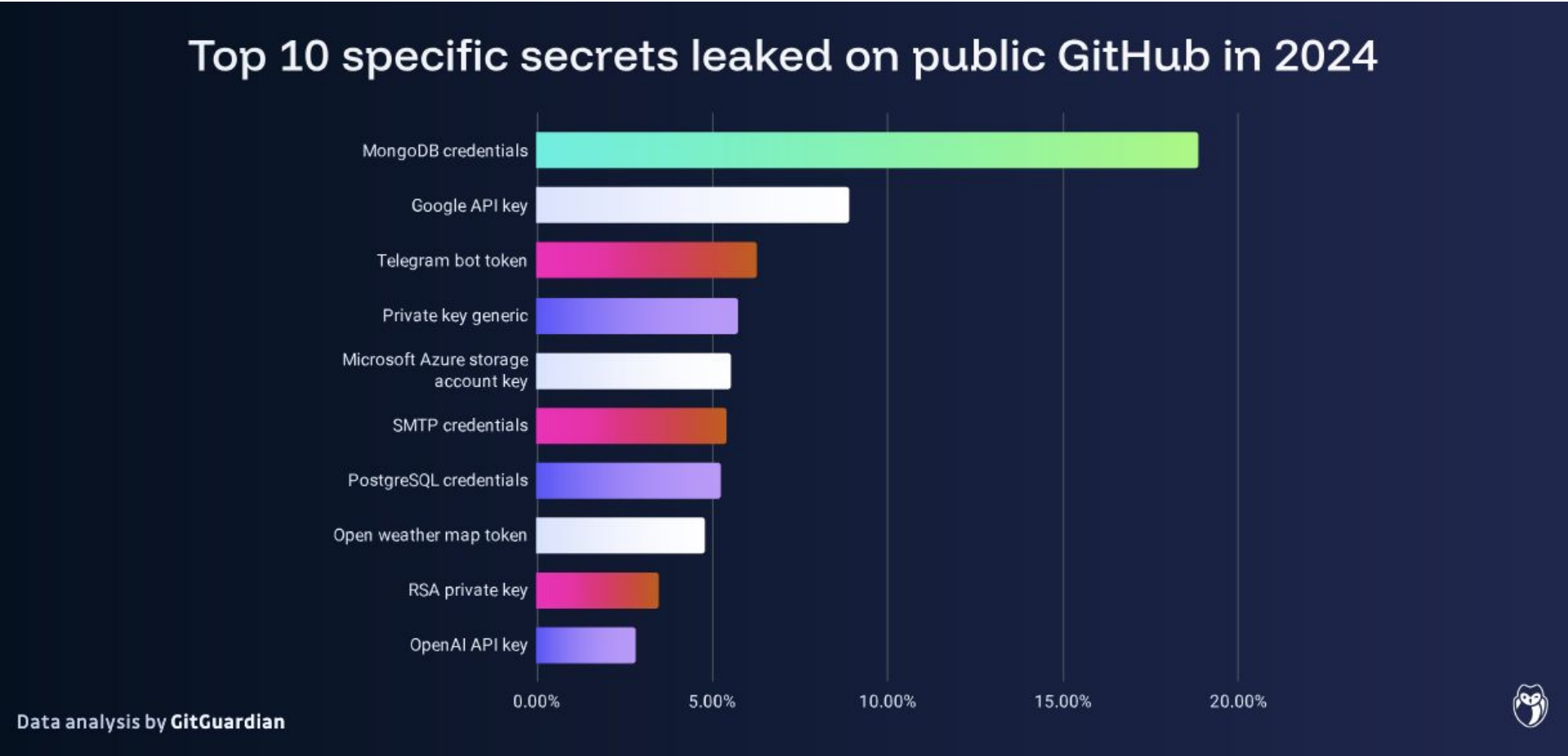
GitGuardian

THE STATE OF

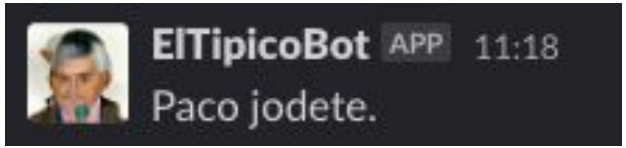
Secrets Sprawl 2025



Workshop: Secrets Scan

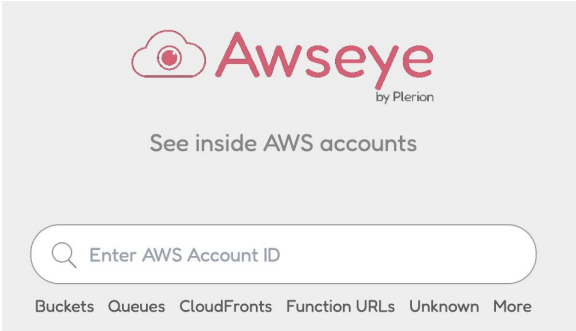


Workshop: Secrets Scan



Other Types of Sensitive Data

- Webhooks
  - Expose internal domains
  - Trigger actions (like Slack Webhooks)
- Account IDs
  - Easier enumeration
- Internal Documents
- PII



Awseye is an open-source intelligence (OSINT) and reconnaissance service tracking publicly accessible AWS data.

Workshop: Secrets Scan

Enable Secrets Scan step

1. Go to `workshop/secrets_scan`
2. Choose: `trufflehog` or `gitleaks`
3. Go to the tool folder and follow the `workflow.yml` instructions
4. Push the changes and create a PR against your main branch (not Unicrons)
5. Wait for the pipeline results
6. Try to solve them!

(Don’t worry, we will show the results later.)

Workshop: Secrets Detection (Results)

Trufflehog

```
✅ Found verified result 🐷🔑
Detector Type: AWS
Decoder Type: PLAIN
Raw result: AKIA2T2SJH6MS337PDWL
Resource_type: Access key
Account: 729780141977
Arn: arn:aws:iam::729780141977:user/canarytokens.com@@xom68iew4t07umwp9nj01uy4s
Rotation_guide: https://howtorotate.com/docs/tutorials/aws/
User_id: AIDA2T2SJH6M5OCWXJQ6T
File: code/src/simple-app.js
Line: 3
```

Gitleaks

```
Finding:     const AWS_SECRET_ACCESS_KEY = 'oMKFrMwcYIJB/PU7l2EOG8wg9KOfQapwVKGP4HaD'
Secret:      oMKFrMwcYIJB/PU7l2EOG8wg9KOfQapwVKGP4HaD
RuleID:      generic-api-key
Entropy:     4.834184
File:        code/src/simple-app.js
Line:        4
Fingerprint: code/src/simple-app.js:generic-api-key:4

Finding:     ...WS_ACCESS_KEY_ID = 'AKIA2T2SJH6MS337PDWL
Secret:      AKIA2T2SJH6MS337PDWL
RuleID:      aws-access-token
Entropy:     3.921928
File:        code/src/simple-app.js
Line:        3
Fingerprint: code/src/simple-app.js:aws-access-token:3
```

Questions?

Workshop: Code Scan

Why is **SAST** Code Scan Important?

- Vulnerability detection in your source code logic
- Faster feedback loop compared to DAST
- Understands vulnerabilities with code context, not only matching regexp.

Common code security issues

- RCE (Remote code execution)
- Secrets leakage
- DoS (Denial of Service)
- SQL Injection

Workshop: Code Scan

Why is **SCA** Code Scan Important?

- License Risk Assessment
- Outdated/Vulnerable Component Detection
- Helps you build a Software Bill of Materials (SBOM)
- Mitigate (detect) supply change attacks

Common SCA findings can cause:

- License issues
- Vulnerabilities in your dependencies
- Executing untrusted code

Workshop: Code Scan

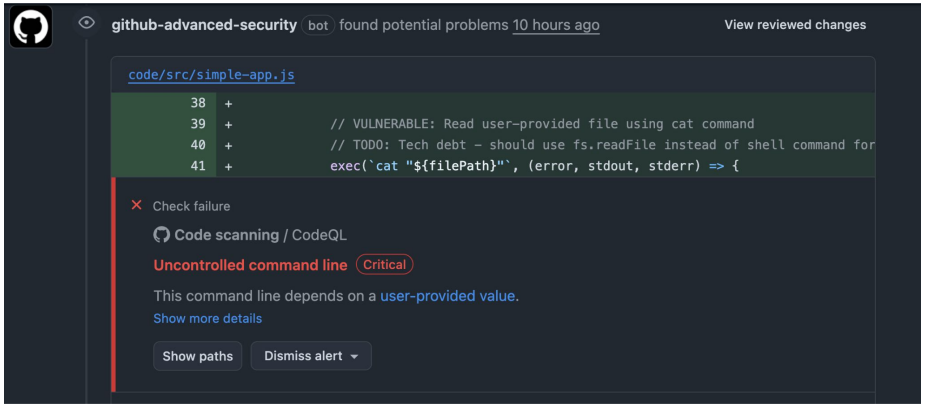
Enable Secrets Scan step

1. Go to `workshop/code_scan`
2. Choose one **SAST** tool as a job: `CodeQL` or `semgrep`
3. Add the **SCA** tool as a job: `Dependency Check`
4. Go to the tool folder and follow the `workflow.yml` instructions.
5. Push the changes and create a PR against your main branch (not Unicrons)
6. Wait for the pipeline results
7. Try to solve them!

(Don’t worry, we will show the results later.)

Workshop: Code Scan (Results SAST)

CodeQL



semgrep

Workshop: Code Scan (Results SCA)

Dependency Check

```
52 Error:
53
54 One or more dependencies were identified with vulnerabilities that have a CVSS score greater than or equal to '7.0':
55
56 package-lock.json?lodash (pkg:npm/lodash@4.17.15): CVE-2020-8203(7.5), GHSA-p6mc-m468-83gw(7.400000095367432), GHSA-35j
57
58 See the dependency-check report for more details.
59
```

Workshop: Code Scan (Results SCA)

Dependency Check

DEPENDENCY-CHECK

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the re

How to read the report | Suppressing false positives | Getting Help: github issues

Sponsor

Project: code

Scan Information (show all):

- *dependency-check version*: 12.1.3
- *Report Generated On*: Mon, 8 Sep 2025 05:22:25 GMT
- *Dependencies Scanned*: 2 (2 unique)
- *Vulnerable Dependencies*: 1
- *Vulnerabilities Found*: 6
- *Vulnerabilities Suppressed*: 0
- ...

Summary

Summary of Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
lodash:4.17.15		pkg:npm/lodash@4.17.15	HIGH	6		3

Check the **Upload Scan Report** step in the SCA Job to check the HTML report with deeper info on the vulnerabilities.

Questions?

Workshop: Infra as Code Scan

Why is IaC Scan Important?

- Identify misconfigurations before deployments
- Treat IaC as any other piece of software
- Scalable security approach for the Cloud

This is supposed to be us creating “infra as code”.

Workshop: Infra as Code Scan

Common IaC Issues

- Access Control:
  - Overly Permissive IAM Policies, Publicly Accessible Resources, Missing Authentication Controls, Default or Weak Credentials
- Encryption:
  - Unencrypted Storage at Rest, Unencrypted Backups & Snapshots, Missing TLS/In-Transit Encryption, Weak or Outdated Cipher Suites
- Network Security:
  - Open Security Groups (0.0.0.0/0), Public-Subnet Exposure, Mis-scoped Load Balancers/Endpoints
- Compliance & Governance:
  - Insufficient Logging & Audit Trails, Lack of Continuous Monitoring/Alerts, Missing Resource Tagging, Improper Backup Retention/Encryption

Workshop: Infra as Code Scan

Specific IaC Recommendations

- Use encrypted remote backends
- Use state locking to prevent concurrent modifications
- Mark sensitive data appropriately
- Pin provider versions
- Vet third-party modules

We recommend running IaC scans against `terraform plan` and equivalents when possible for more accurate results.

Workshop: Infra as Code Scan

Extra ball: Terraform Plan RCE

https://alex.kaskaso.li/post/terraform-plan-rce

Terraform and external programs. You can use the `external` data source to run arbitrary code during a `plan`.

```
data "external" "example" {
    program = ["python", "${path.module}/example-data-source.py"]

        query = {
# arbitrary map from strings to strings, passed
# to the external program as the data query.
            id = "abc123"
        }
}
```

The `query` will be passed as a JSON string on `stdin` to the `program`; you could use this to grab variables from Terraform.

Workshop: Infra as Code Scan

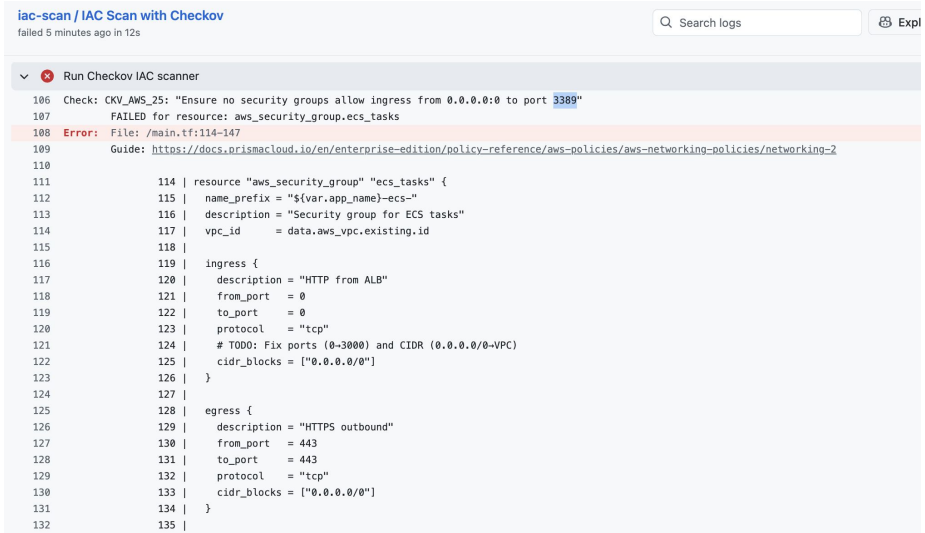
Enable IaC Scan step

1. Go to `workshop/iac_scan`
2. Choose: `trivy` or `checkov`
3. Go to the tool folder and follow the `workflow.yml` instructions
4. Push the changes to a new branch and create a PR against your main branch (not Unicrons)
5. Wait for the pipeline results
6. Try to solve them!

(Don’t worry, we will show the results later.)

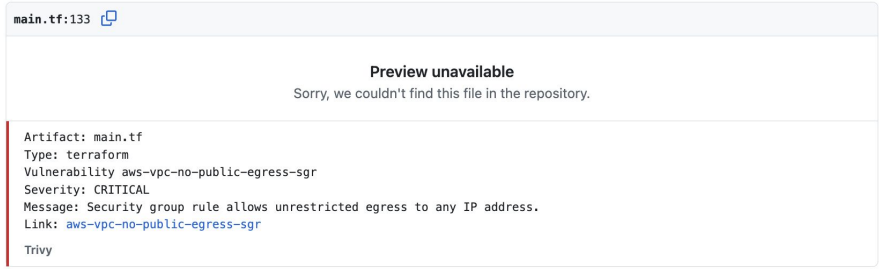
Workshop: Infra as Code Scan (Results)

Checkov




```
ingress {
  description = "HTTP from ALB"
  from_port   = 3000
  to_port     = 3000
  protocol    = "tcp"
  cidr_blocks = [data.aws_vpc.existing.cidr_block]
}
```

Trivy




```
# trivy:ignore:AVD-AWS-0104
egress {
  description = "HTTPS outbound"
  from_port   = 443
  to_port     = 443
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}

# trivy:ignore:AVD-AWS-0104
egress {
  description = "HTTP outbound"
  from_port   = 80
  to_port     = 80
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}
```

Questions?

Workshop: Container Scan

Why is Container Scan Important?

- Common way to deploy software → Common target for attackers
- Access to underlying infra/secrets

Common Container Security Issues

- Vulnerable dependencies
- Hardcoded Secrets
- Excessive Privileges (root, network, filesystem)

Workshop: Container Scan

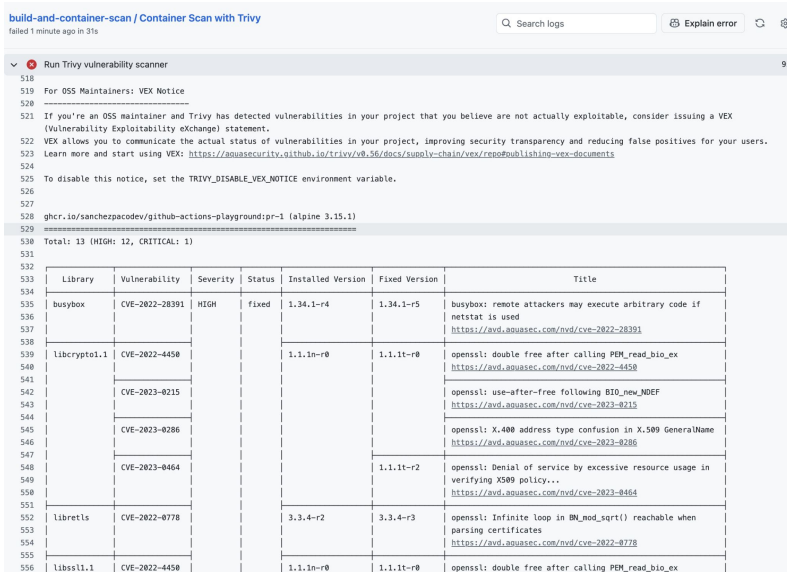
Enable Pipeline Security Step

1. Go to `workshop/container_scan`
2. Choose: `trivy` or `grype`
3. Go to the tool folder and follow the `workflow.yml` instructions
4. Push the changes to a new branch and create a PR against your main branch (not Unicrons)
5. Wait for the pipeline results
6. Try to solve them!

(Don’t worry, we will show the results later.)

Workshop: Container Scan (Results)

Trivy



build-and-container-scan / Container Scan with Trivy
failed 1 minute ago in 31s

Search logs | Explain error

Run Trivy vulnerability scanner

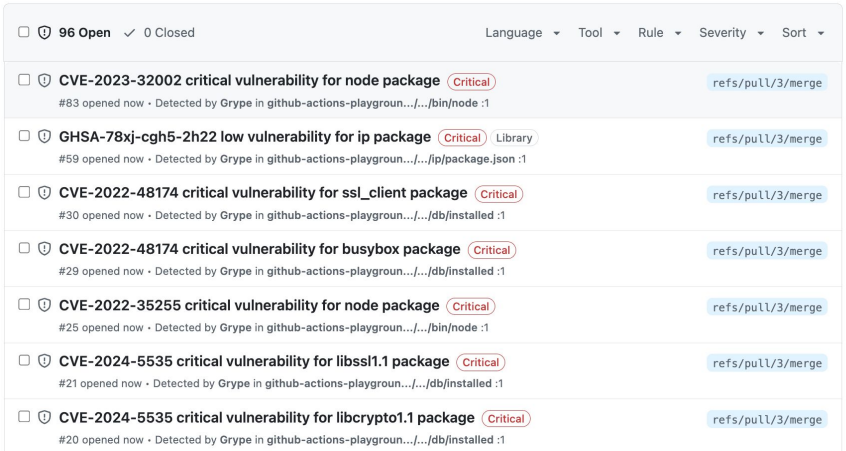
```
For OSS Maintainers: VEX Notice
--------------------------------
If you're an OSS maintainer and Trivy has detected vulnerabilities in your project that you believe are not actually exploitable, consider issuing a VEX
(Vulnerability Exploitability eXchange) statement.
VEX allows you to communicate the actual status of vulnerabilities in your project, improving security transparency and reducing false positives for your users.
Learn more and start using VEX: https://aquasecurity.github.io/trivy/v0.56/docs/supply-chain/vex/repo#publishing-vex-documents

To disable this notice, set the TRIVY_DISABLE_VEX_NOTICE environment variable.


ghcr.io/sanchezpacodev/github-actions-playground:pr-1 (alpine 3.15.1)
==========================================================================
Total: 13 (HIGH: 12, CRITICAL: 1)

┌─────────────┬────────────────┬──────────┬────────┬───────────────────┬───────────────┬──────────────────────────────────────────────────────────────┐
│   Library   │ Vulnerability  │ Severity │ Status │ Installed Version │ Fixed Version │                            Title                             │
├─────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ busybox     │ CVE-2022-28391 │ HIGH     │ fixed  │ 1.34.1-r4         │ 1.34.1-r5     │ busybox: remote attackers may execute arbitrary code if      │
│             │                │          │        │                   │               │ netstat is used                                              │
│             │                │          │        │                   │               │ https://avd.aquasec.com/nvd/cve-2022-28391                   │
├─────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libcrypto1.1 │ CVE-2022-4450 │          │        │ 1.1.1n-r0         │ 1.1.1t-r0     │ openssl: double free after calling PEM_read_bio_ex           │
│             │                │          │        │                   │               │ https://avd.aquasec.com/nvd/cve-2022-4450                    │
│             ├────────────────┤          │        │                   │               ├──────────────────────────────────────────────────────────────┤
│             │ CVE-2023-0215  │          │        │                   │               │ openssl: use-after-free following BIO_new_NDEF               │
│             │                │          │        │                   │               │ https://avd.aquasec.com/nvd/cve-2023-0215                    │
│             ├────────────────┤          │        │                   │               ├──────────────────────────────────────────────────────────────┤
│             │ CVE-2023-0286  │          │        │                   │               │ openssl: X.400 address type confusion in X.509 GeneralName   │
│             │                │          │        │                   │               │ https://avd.aquasec.com/nvd/cve-2023-0286                    │
│             ├────────────────┤          │        │                   │               ├──────────────────────────────────────────────────────────────┤
│             │ CVE-2023-0464  │          │        │                   │ 1.1.1t-r2     │ openssl: Denial of service by excessive resource usage in    │
│             │                │          │        │                   │               │ verifying X509 policy...                                     │
│             │                │          │        │                   │               │ https://avd.aquasec.com/nvd/cve-2023-0464                    │
├─────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libretls    │ CVE-2022-0778  │          │        │ 3.3.4-r2          │ 3.3.4-r3      │ openssl: Infinite loop in BN_mod_sqrt() reachable when        │
│             │                │          │        │                   │               │ parsing certificates                                         │
│             │                │          │        │                   │               │ https://avd.aquasec.com/nvd/cve-2022-0778                    │
├─────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libssl1.1   │ CVE-2022-4450  │          │        │ 1.1.1n-r0         │ 1.1.1t-r0     │ openssl: double free after calling PEM_read_bio_ex           │
```

Grype



96 Open | 0 Closed | Language | Tool | Rule | Severity | Sort

- CVE-2023-32002 critical vulnerability for node package — Critical — refs/pull/3/merge
  #83 opened now • Detected by Grype in github-actions-playgroun.../.../bin/node :1
- GHSA-78xj-cgh5-2h22 low vulnerability for ip package — Critical — Library — refs/pull/3/merge
  #59 opened now • Detected by Grype in github-actions-playgroun.../.../ip/package.json :1
- CVE-2022-48174 critical vulnerability for ssl_client package — Critical — refs/pull/3/merge
  #30 opened now • Detected by Grype in github-actions-playgroun.../.../db/installed :1
- CVE-2022-48174 critical vulnerability for busybox package — Critical — refs/pull/3/merge
  #29 opened now • Detected by Grype in github-actions-playgroun.../.../db/installed :1
- CVE-2022-35255 critical vulnerability for node package — Critical — refs/pull/3/merge
  #25 opened now • Detected by Grype in github-actions-playgroun.../.../bin/node :1
- CVE-2024-5535 critical vulnerability for libssl1.1 package — Critical — refs/pull/3/merge
  #21 opened now • Detected by Grype in github-actions-playgroun.../.../db/installed :1
- CVE-2024-5535 critical vulnerability for libcrypto1.1 package — Critical — refs/pull/3/merge
  #20 opened now • Detected by Grype in github-actions-playgroun.../.../db/installed :1

Dockerfile: FROM node:16.14.0-alpine -> FROM node:24-alpine

package.json: "lodash": "4.17.15" -> "lodash": "4.17.21" (run again `npm install`

Questions?

Workshop: Runtime Infra Scan

Why is Runtime Infra Scan Important?

- We can’t catch everything in previous steps
- Manual changes exist
- It’s the final source of truth

Common Runtime Infra Security Issues

- Access Control
- Encryption
- Network Security
- Compliance & Governance
- Drift

Workshop: Runtime Infra Scan

Enable Pipeline Security Step

1. Go to `workshop/runtime_infra_scan`
2. Choose: `prowler` or `steampipe`
3. Go to the tool folder and follow the `workflow.yml` instructions
4. Push the changes to a new branch and create a PR against your main branch (not Unicrons)
5. Wait for the pipeline results
6. Try to solve them! (if you are using your own account 😅)

Workshop: Runtime Infra Scan (Results)

Prowler

PROWLER

Report Information	AWS Assessment Summary	AWS Credentials	Assessment Overview
Version: 5.10.0	**AWS Account:** REDACTED	**User Id:** AROARQ426VA5WG4VSAANC:ProwlerSession	**Total Findings:** 99
Parameters used: aws --service iam s3 --output-formats html -z	**AWS-CLI Profile:** default	**Caller Identity ARN:** arn:aws:sts::REDACTED:assumed-role/ProwlerRole/ProwlerSession	**Passed:** 62
Date: 2025-08-08T00:38:23.134464	**Audited Regions:** All Regions		**Passed (Muted):** 0
			Failed: 37
			Failed (Muted): 0
			Total Resources: 21

Filters (2) Show 100 entries Search:

Status	Severity	Service Name	Region	Check ID	Check Title	Resource ID	Resource Tags	Status Extended	Risk	Recommendation	Compliance
PASS	high	s3	us-east-1	s3_bucket_shadow_resource_vulnerability	Check for S3 buckets vulnerable to Shadow Resource Hijacking (Bucket Monopoly)	arn:aws:s3:::warden-easter-egg-c4bak76z		S3 bucket warden-easter-egg-c4bak76z is not a known shadow resource.	An attacker can pre-create S3 read more...	Ensure that all S3 buckets ass read more...	
FAIL	medium	s3	us-east-1	s3_bucket_no_mfa_delete	Check if S3 bucket MFA Delete is not enabled.	arn:aws:s3:::warden-easter-egg-c4bak76z		S3 Bucket warden-easter-egg-c4bak76z has MFA Delete disabled.	Your security credentials are read more...	Adding MFA delete to an S3 buc read more...	•MITRE-ATTACK: T1485 •Prowle read more...
FAIL	low	s3	us-east-1	s3_bucket_lifecycle_enabled	Check if S3 buckets have a Lifecycle configuration enabled	arn:aws:s3:::warden-easter-egg-c4bak76z		S3 Bucket warden-easter-egg-c4bak76z does not have a lifecycle configuration enabled.	The risks of not having lifecy read more...	Enable lifecycle policies on y read more...	•ISO27001-2022: A.8.10 •PCI- read more...
PASS	medium	s3	us-east-1	s3_bucket_acl_prohibited	Check if S3 buckets have ACLs enabled	arn:aws:s3:::warden-easter-egg-c4bak76z		S3 Bucket warden-easter-egg-c4bak76z has bucket ACLs disabled.	S3 ACLs are a legacy access co read more...	Ensure that S3 ACLs are disabl read more...	•PCI-4.0: 7.2.1.24, 7.2.2.24, read more...
PASS	medium	s3	us-east-1	s3_bucket_default_encryption	Check if S3 buckets have default encryption (SSE) enabled or use a bucket policy to enforce it.	arn:aws:s3:::warden-easter-egg-c4bak76z		S3 Bucket warden-easter-egg-c4bak76z has Server Side Encryption with AES256.	Amazon S3 default encryption p read more...	Ensure that S3 buckets have en read more...	•ISO27001-2022: A.8.11, A.8.2 read more...
FAIL	medium	s3	us-east-1	s3_bucket_event_notifications_enabled	Check if S3 buckets have event notifications enabled.	arn:aws:s3:::warden-easter-egg-c4bak76z		S3 Bucket warden-easter-egg-c4bak76z does not have event notifications enabled.	Without event notifications, i read more...	Enable event notifications for read more...	•PCI-4.0: 11.5.2.5, 11.6.1.5, read more...
FAIL	medium	s3	us-east-1	s3_bucket_kms_encryption	Check if S3 buckets have KMS encryption enabled.	arn:aws:s3:::warden-easter-egg-c4bak76z		Server Side Encryption is not configured with kms for S3 Bucket warden-easter-egg-c4bak76z.	Amazon S3 KMS encryption provi read more...	Ensure that S3 buckets have en read more...	•ISO27001-2022: A.8.11, A.8.2 read more...

Steampipe

1.5 Ensure MFA is enabled for the 'root' user account

The 'root' user account is the most privileged user in an AWS account. Multi-factor Authentication (MFA) adds an extra layer of protection on top of a username and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their username and password as well as for an authentication code from their AWS MFA device.

OK	Skip	Info	Alarm	Error	Total
1	0	0	0	0	1

	Reason	Dimensions
✅	MFA enabled for root account.	REDACTED

1.6 Ensure hardware MFA is enabled for the 'root' user account

The 'root' user account is the most privileged user in an AWS account. MFA adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password as well as for an authentication code from their AWS MFA device. For Level 2, it is recommended that the root user account be protected with a hardware MFA.

OK	Skip	Info	Alarm	Error	Total
0	0	0	1	0	1

	Reason	Dimensions
❌	MFA enabled for root account, but the MFA associated is a virtual device.	REDACTED

1.7 Eliminate use of the 'root' user for administrative and daily tasks

With the creation of an AWS account, a 'root user' is created that cannot be disabled or deleted. That user has unrestricted access to and control over all resources in the AWS account. It is highly recommended that the use of this account be avoided for everyday tasks.

OK	Skip	Info	Alarm	Error	Total
0	0	0	1	0	1

	Reason	Dimensions
❌	Root password used 05-Aug-2025 (2 days). Access Key 1 never used. Access Key 2 never used.	REDACTED

Questions?

Thanks for coming!

eduardoSimon

eduardo-simon

@andoniaf_

andoniaf

andoniaf.unicrons.cloud

unicrons

@unicrons_cloud

unicrons.cloud

Feel free to contact us, or open an issue in the repo.
Any feedback would be greatly appreciated.

https://github.com/unicrons/secure-pipeline-workshop