Ethan Troy

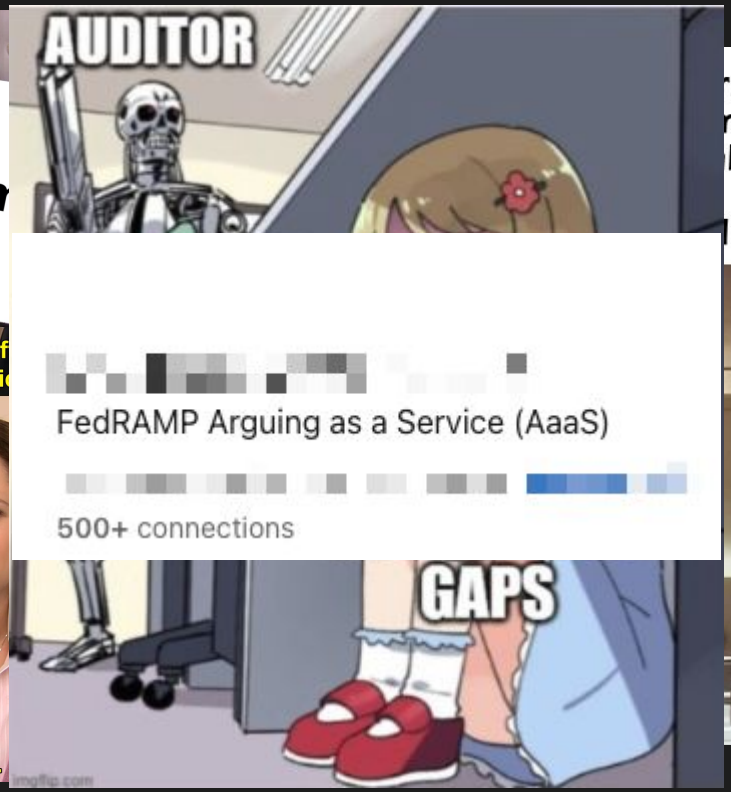
GRC Engineering in the Cloud

The Problems with GRC (Governance, Risk, and Compliance)

“The Players are Not Mature Enough”

The Problem with Auditors

The Problem with GRC Pros

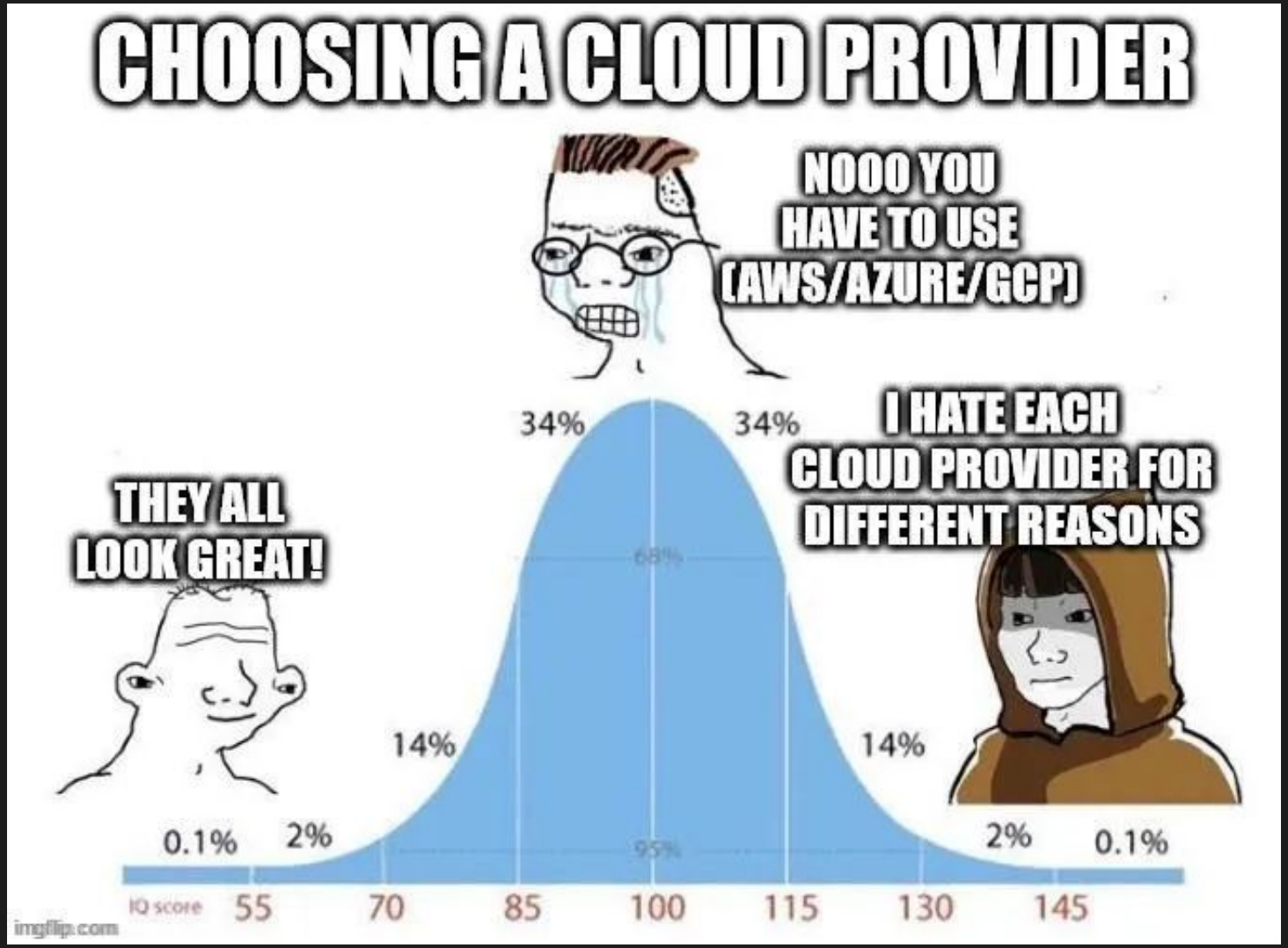


The Problem with Leaders

“You need to be able to translate this to your leadership”

The Solution - Creativity

CHOOSING A CLOUD PROVIDER



Major Clouds

Azure

A favorite among those that come from classical on-prem AD environments

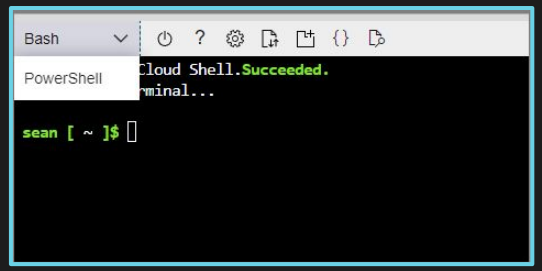
GCP

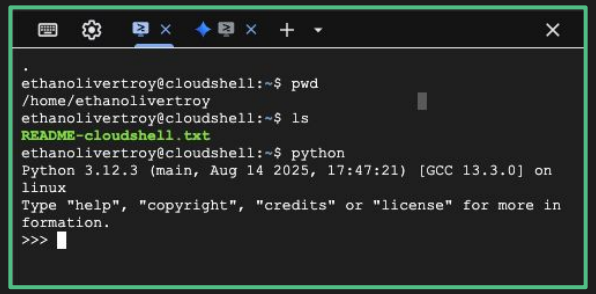
Smallest Market Share but rapidly growing

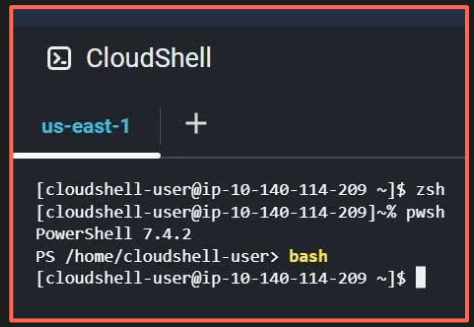
AWS

The Market Leader
Many developer services are actually just AWS Wrappers

Cloud Shell







Azure

Released in September 2017

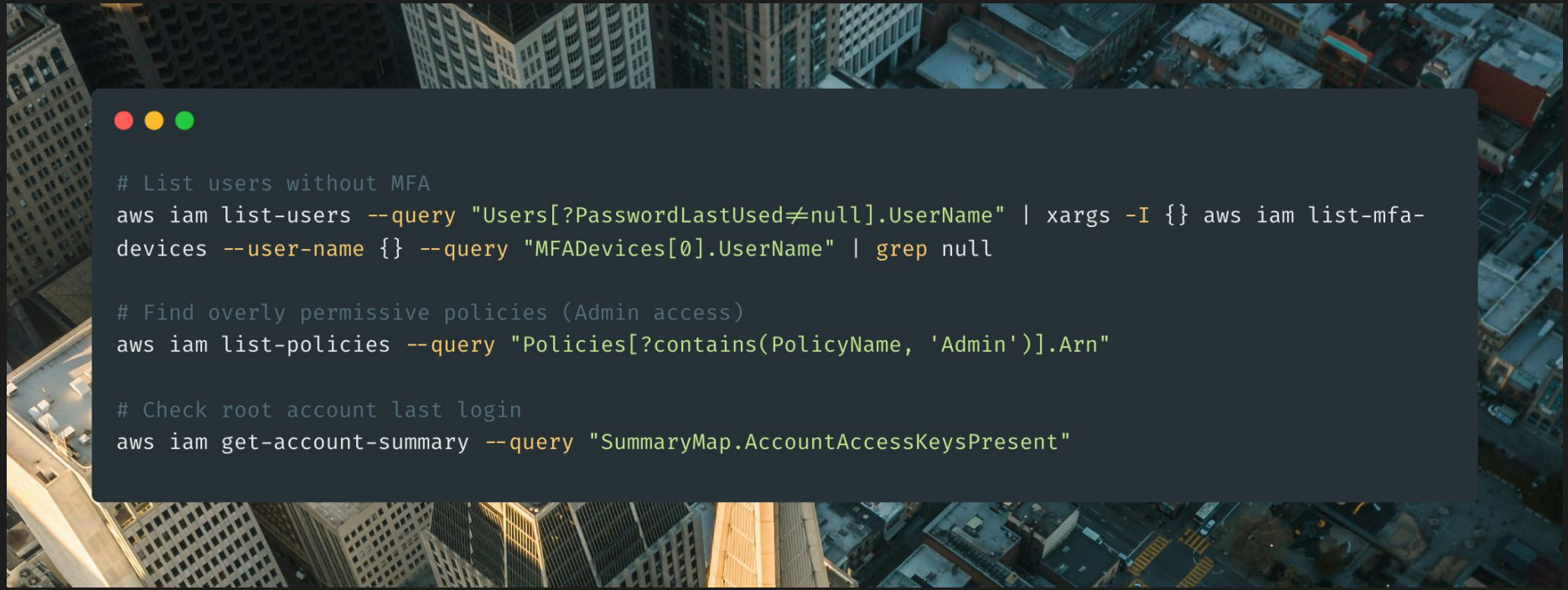
GCP

Released in 2016

AWS

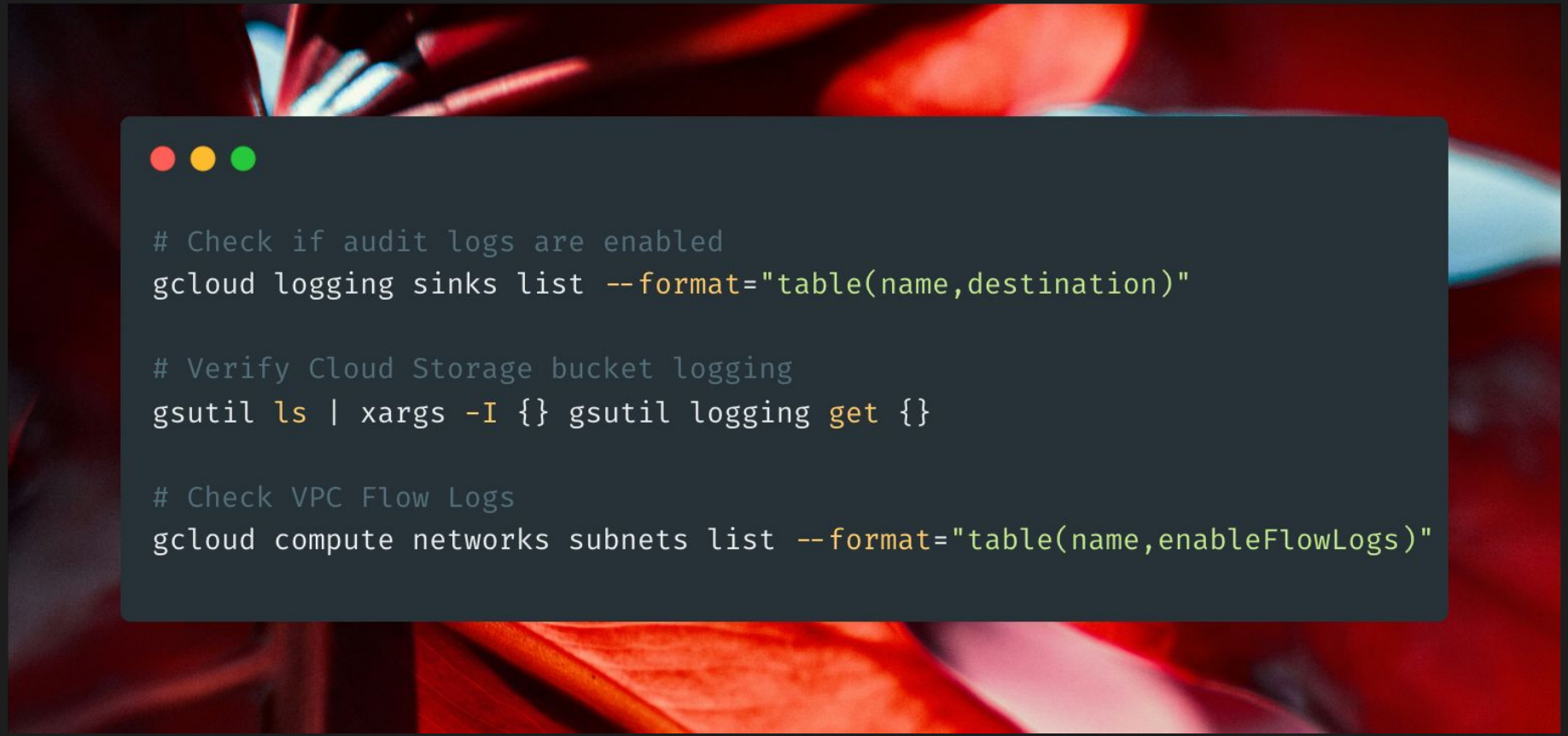
Released in December 2020

There is a CMD for that



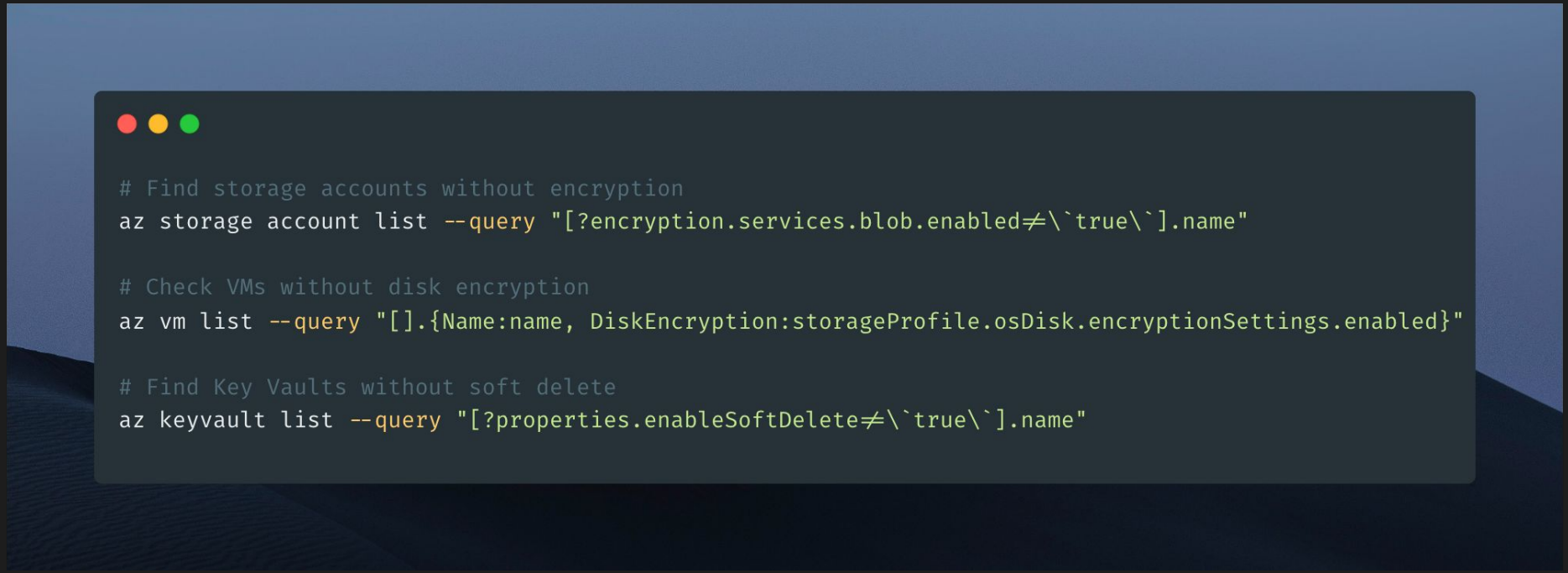
```
# List users without MFA
aws iam list-users --query "Users[?PasswordLastUsed=≠null].UserName" | xargs -I {} aws iam list-mfa-devices --user-name {} --query "MFADevices[0].UserName" | grep null

# Find overly permissive policies (Admin access)
aws iam list-policies --query "Policies[?contains(PolicyName, 'Admin')].Arn"

# Check root account last login
aws iam get-account-summary --query "SummaryMap.AccountAccessKeysPresent"
```

There is a CMD for that



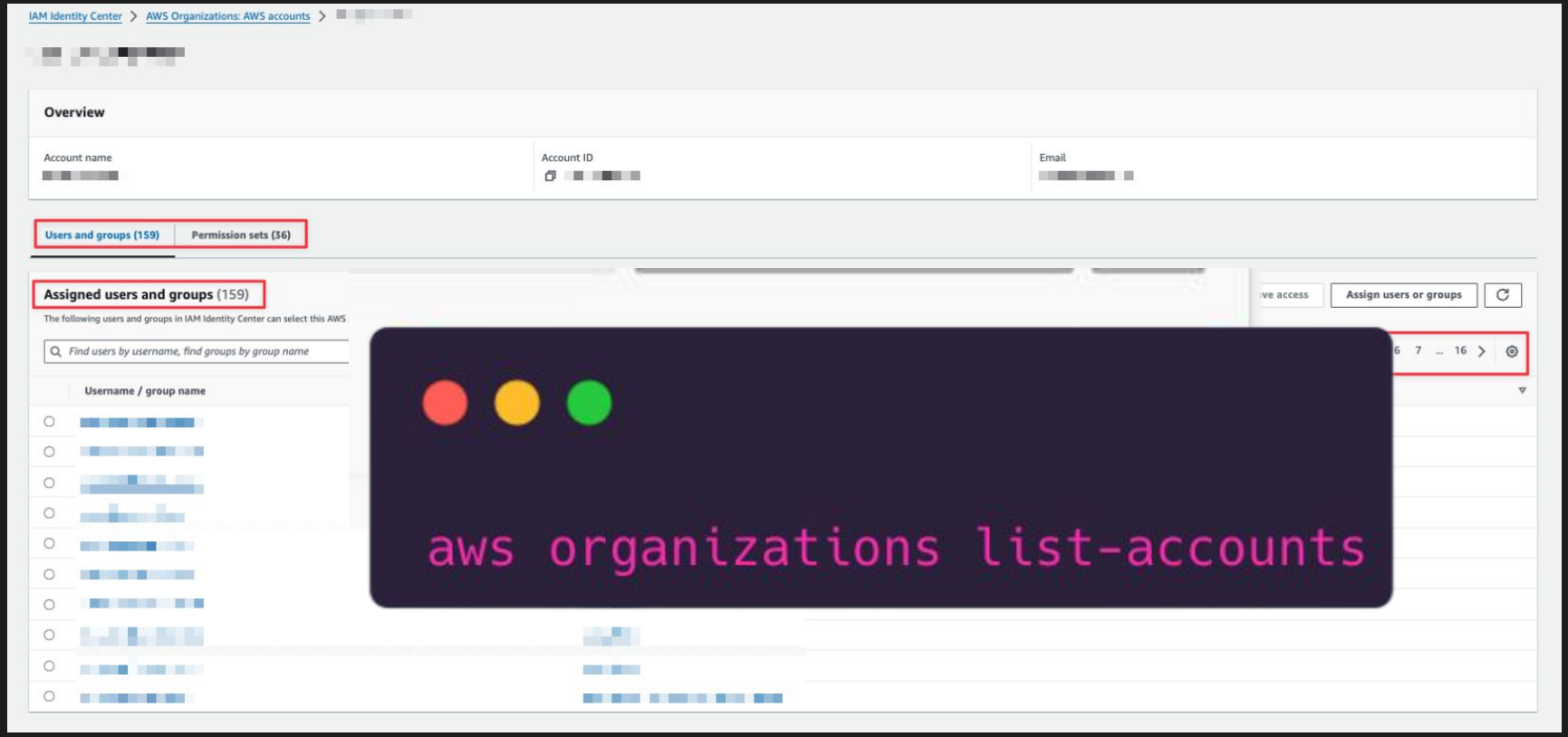
```
# Check if audit logs are enabled
gcloud logging sinks list --format="table(name,destination)"

# Verify Cloud Storage bucket logging
gsutil ls | xargs -I {} gsutil logging get {}

# Check VPC Flow Logs
gcloud compute networks subnets list --format="table(name,enableFlowLogs)"
```

There is a CMD for that

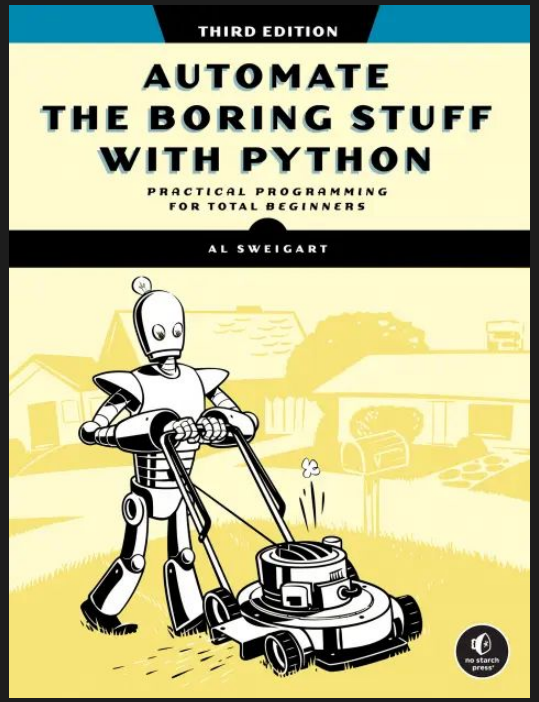


```
# Find storage accounts without encryption
az storage account list --query "[?encryption.services.blob.enabled≠\`true\`].name"

# Check VMs without disk encryption
az vm list --query "[].{Name:name, DiskEncryption:storageProfile.osDisk.encryptionSettings.enabled}"

# Find Key Vaults without soft delete
az keyvault list --query "[?properties.enableSoftDelete≠\`true\`].name"
```

#AbolishScreenshots



Automation




```
~/Gith/w/Fo/2/Fortreum-FedRAMP-20x-GCP-Evaluation on main ?11
❯ python gcp_fedramp_collector.py
```

```
# Only checks bucket ACL - misses critical security settings
aws s3api get-bucket-acl --bucket my-bucket
```

Just because you've programtically pulled something, doesn't mean that is the whole picture

```
# Comprehensive S3 security audit
aws s3api get-bucket-acl --bucket my-bucket
aws s3api get-bucket-policy --bucket my-bucket
aws s3api get-bucket-encryption --bucket my-bucket
aws s3api get-bucket-versioning --bucket my-bucket
aws s3api get-bucket-logging --bucket my-bucket
aws s3api get-public-access-block --bucket my-bucket
aws s3api get-bucket-policy-status --bucket my-bucket

# Check for publicly accessible objects
aws s3api list-objects-v2 --bucket my-bucket \
  --query "Contents[?contains(Grants[].Permission,
'READ')]"
```

```python
import json
import boto3
from datetime import datetime
from pathlib import Path

class ComplianceEvidenceCollector:
    def __init__(self, framework="NIST-800-53"):
        self.framework = framework
        self.s3 = boto3.client('s3')
        self.evidence_bucket = 'compliance-evidence'

    def collect_and_store(self, control_id, evidence_data):
        # Structure evidence
        evidence = {
            'control_id': control_id,
            'framework': self.framework,
            'timestamp': datetime.now().isoformat(),
            'automated_check': True,
            'evidence': evidence_data,
            'status': self.evaluate_compliance(evidence_data)
        }

        # Store to S3 with proper structure
        key = f"{self.framework}/{control_id}/{datetime.now().strftime(

        self.s3.put_object(
            Bucket=self.evidence_bucket,
            Key=key,
            Body=json.dumps(evidence, indent=2),
            ServerSideEncryption='AES256',
            Metadata={
                'control-id': control_id,
                'framework': self.framework,
                'automated': 'true'
            }
        )

        return key

    def evaluate_compliance(self, data):
        # Logic to determine pass/fail
        if data.get('non_compliant_resources'):
            return 'FAIL'
        return 'PASS'
```

```python
# Store to S3 with proper structure
key = f"{self.framework}/{control_id}/{datetime.now().strftime('%Y%m%d_%H%M%S')}.json"

self.s3.put_object(
    Bucket=self.evidence_bucket,
    Key=key,
    Body=json.dumps(evidence, indent=2),
    ServerSideEncryption='AES256',
    Metadata={
        'control-id': control_id,
        'framework': self.framework,
        'automated': 'true'
    }
)
```

Mapping to Frameworks

GRC Automation Software

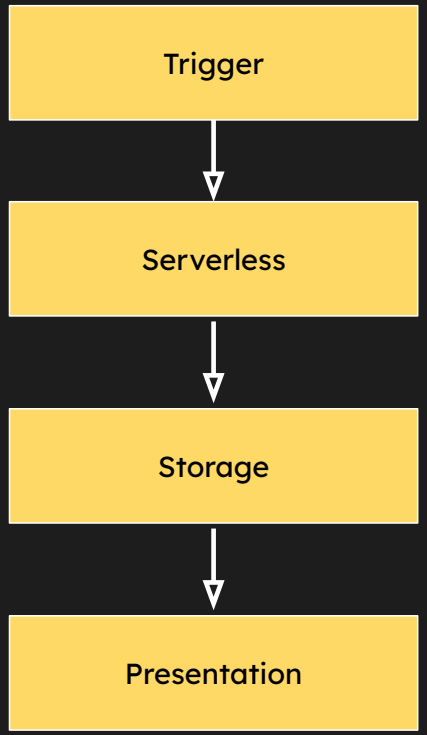
DRATA

ARCHER

PARAMIFY

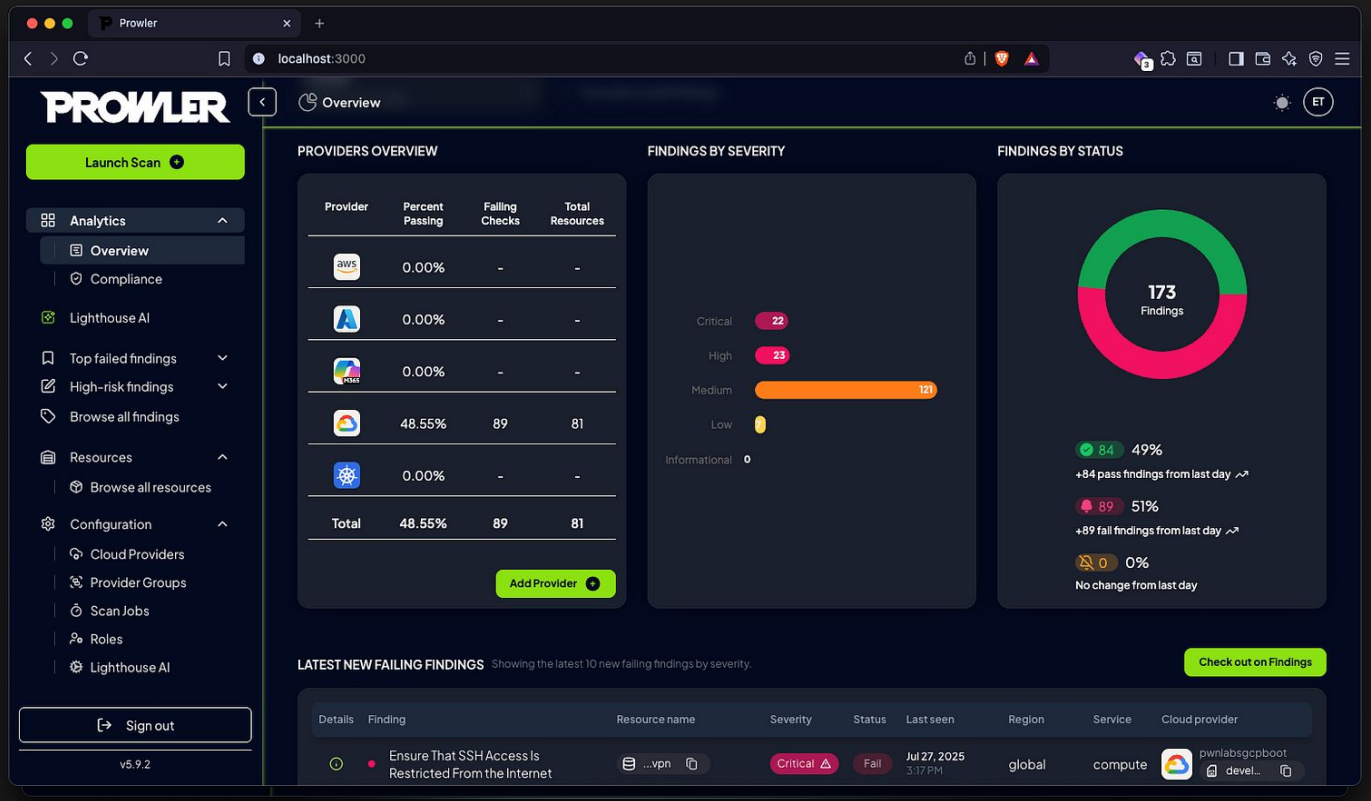
RegScale

Scaling This Yourself



It does not matter what it is called.
They all do the same thing.
Don’t get caught up on names.
Get caught up on the function and result you want to achieve.
Be creative.

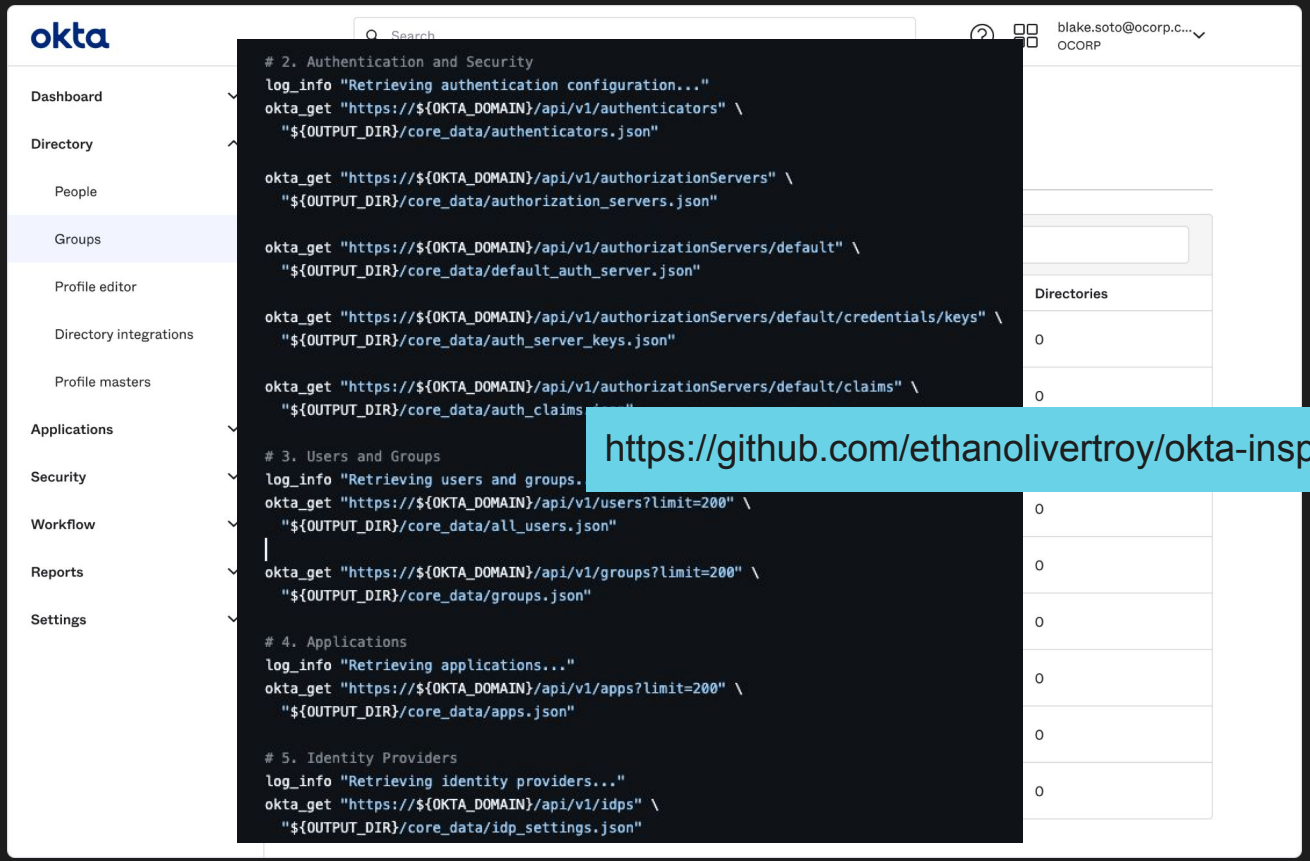
Open Source Tools



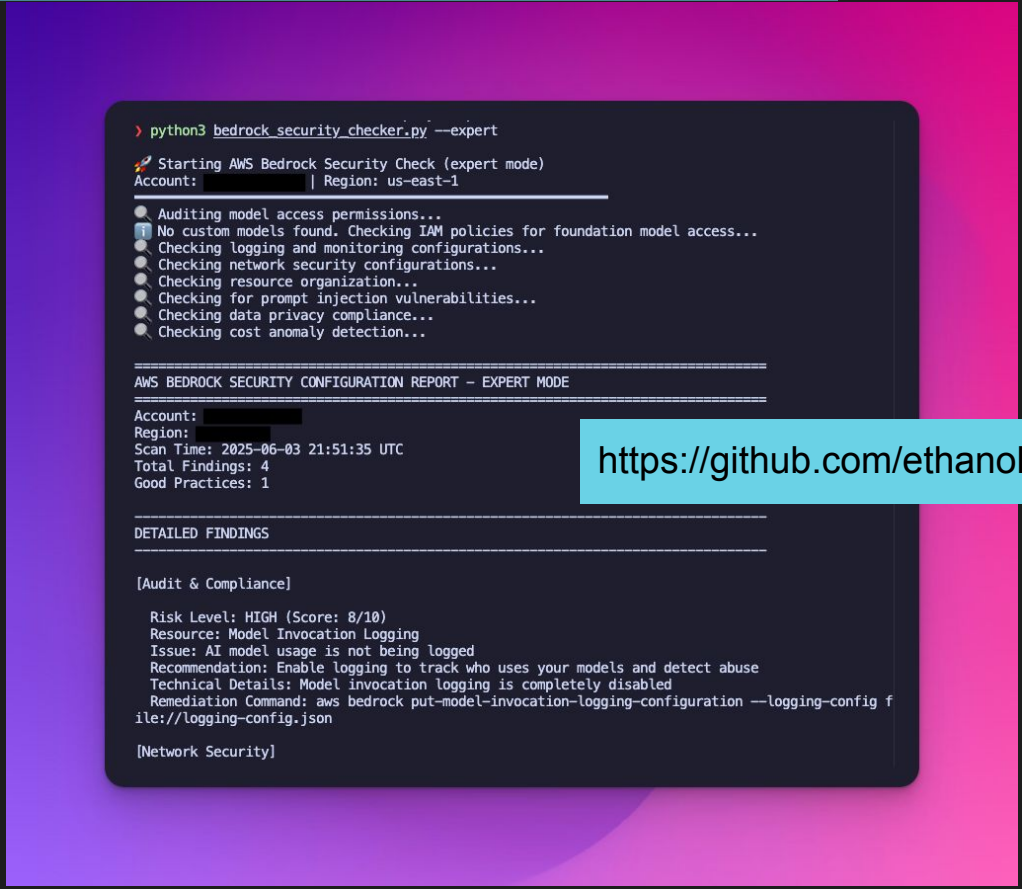
Open Source Tools

Offensive Security has some of the best “GRC” Tools
They simply have to be repurposed

Building Tools with APIs

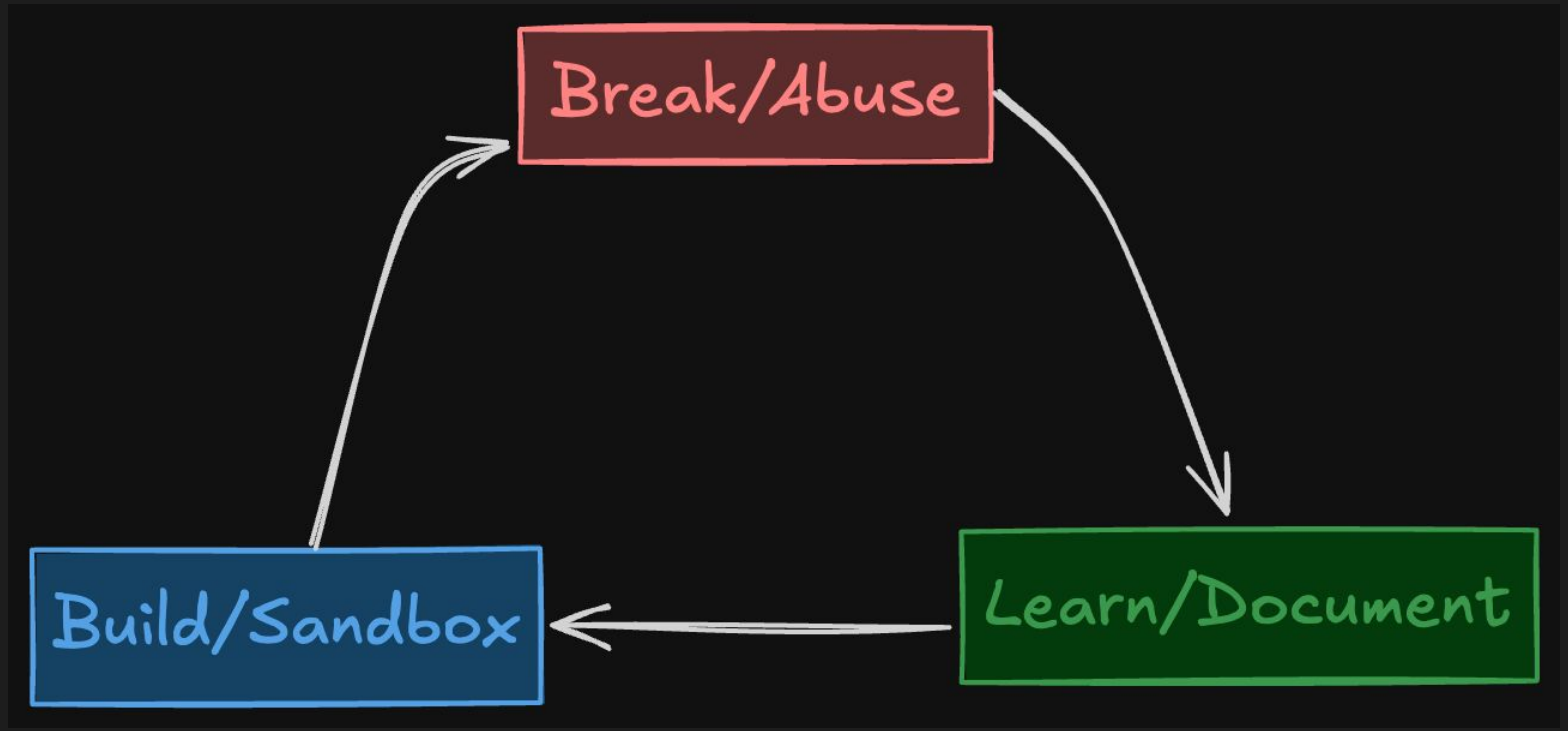


Building Tools with APIs



https://github.com/ethanolivertroy/wilma

Anyone Can Do This



Think Like A . . .

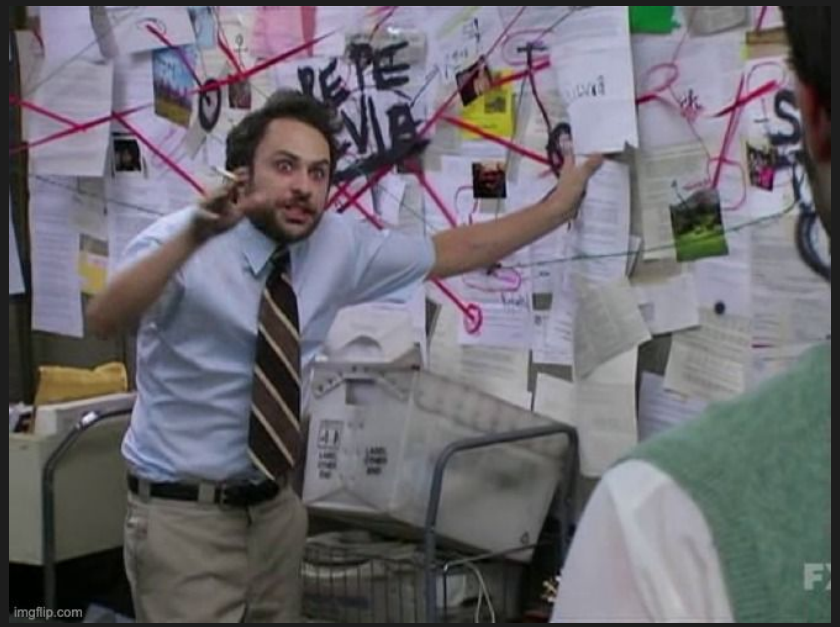

I JUST INSTALLED KALI LINUX
I'M MR.ROBOT

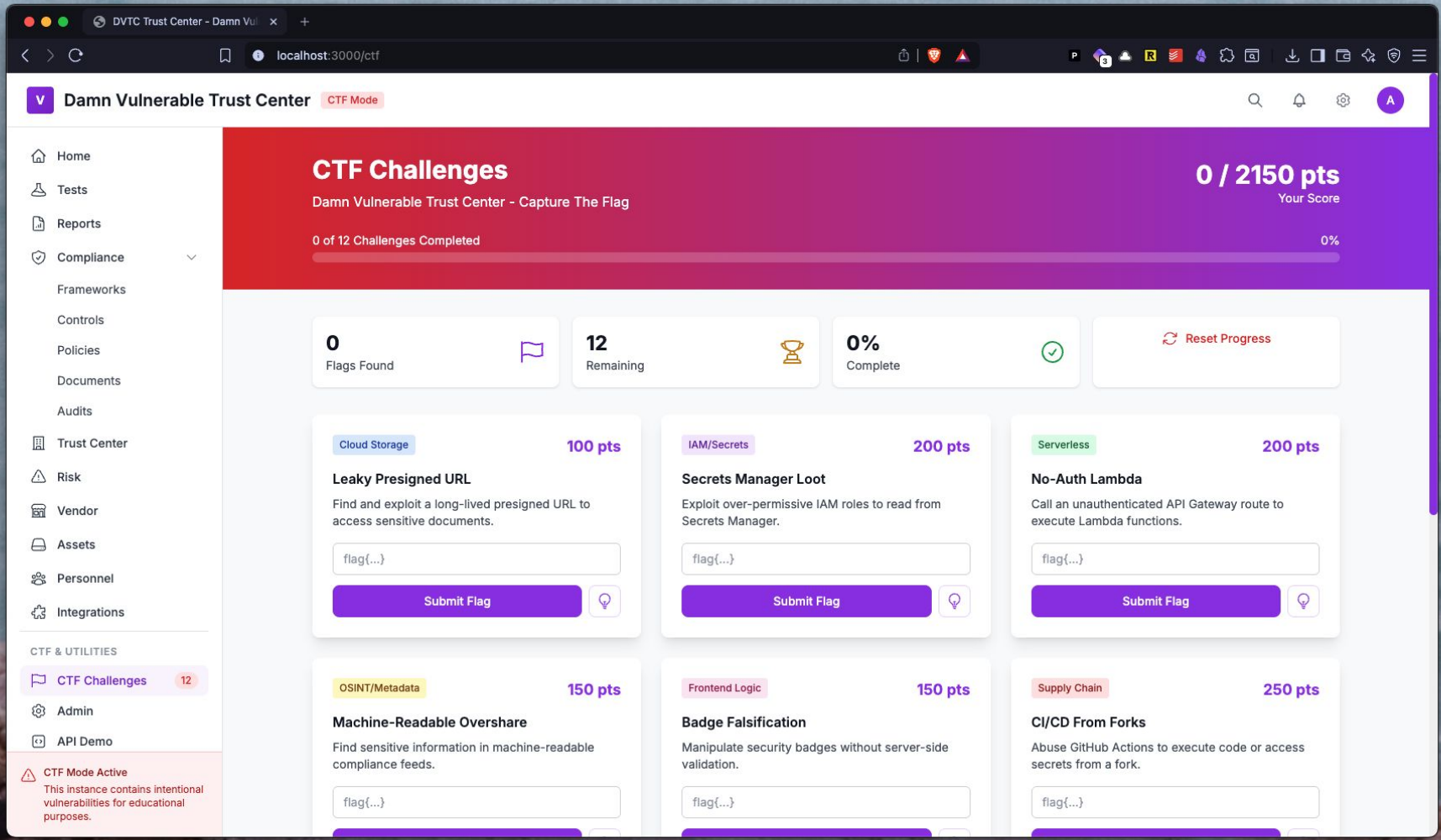

WHEN YOU FIND A COMPLIANCE VIOLATION
BEFORE THE AUDITOR DOES.


THE TWO STATES OF EVERY PROGRAMMER
I AM A GOD.
I HAVE NO IDEA WHAT I'M DOING.

Simple But Not Easy

Developers/Builders Don’t Always Think in the Same Way as Security Engineers




DVTC Trust Center - Damn Vul
localhost:3000/ctf
Damn Vulnerable Trust Center
CTF Mode
A
Home
Tests
Reports
Compliance
Frameworks
Controls
Policies
Documents
Audits
Trust Center
Risk
Vendor
Assets
Personnel
Integrations
CTF & UTILITIES
CTF Challenges 12
Admin
API Demo
CTF Mode Active
This instance contains intentional vulnerabilities for educational purposes.
CTF Challenges
Damn Vulnerable Trust Center - Capture The Flag
0 / 2150 pts
Your Score
0 of 12 Challenges Completed
0%
0
Flags Found
12
Remaining
0%
Complete
Reset Progress
Cloud Storage
100 pts
Leaky Presigned URL
Find and exploit a long-lived presigned URL to access sensitive documents.
flag{...}
Submit Flag
IAM/Secrets
200 pts
Secrets Manager Loot
Exploit over-permissive IAM roles to read from Secrets Manager.
flag{...}
Submit Flag
Serverless
200 pts
No-Auth Lambda
Call an unauthenticated API Gateway route to execute Lambda functions.
flag{...}
Submit Flag
OSINT/Metadata
150 pts
Machine-Readable Overshare
Find sensitive information in machine-readable compliance feeds.
flag{...}
Frontend Logic
150 pts
Badge Falsification
Manipulate security badges without server-side validation.
flag{...}
Supply Chain
250 pts
CI/CD From Forks
Abuse GitHub Actions to execute code or access secrets from a fork.
flag{...}

Next Steps


I fear no man.
But that thing...
Setting Up
Git and GitHub
it scares me.


So programming is 15% coding,
25% debugging and 60% googling?
You just insulted my entire race of people.
But yes.

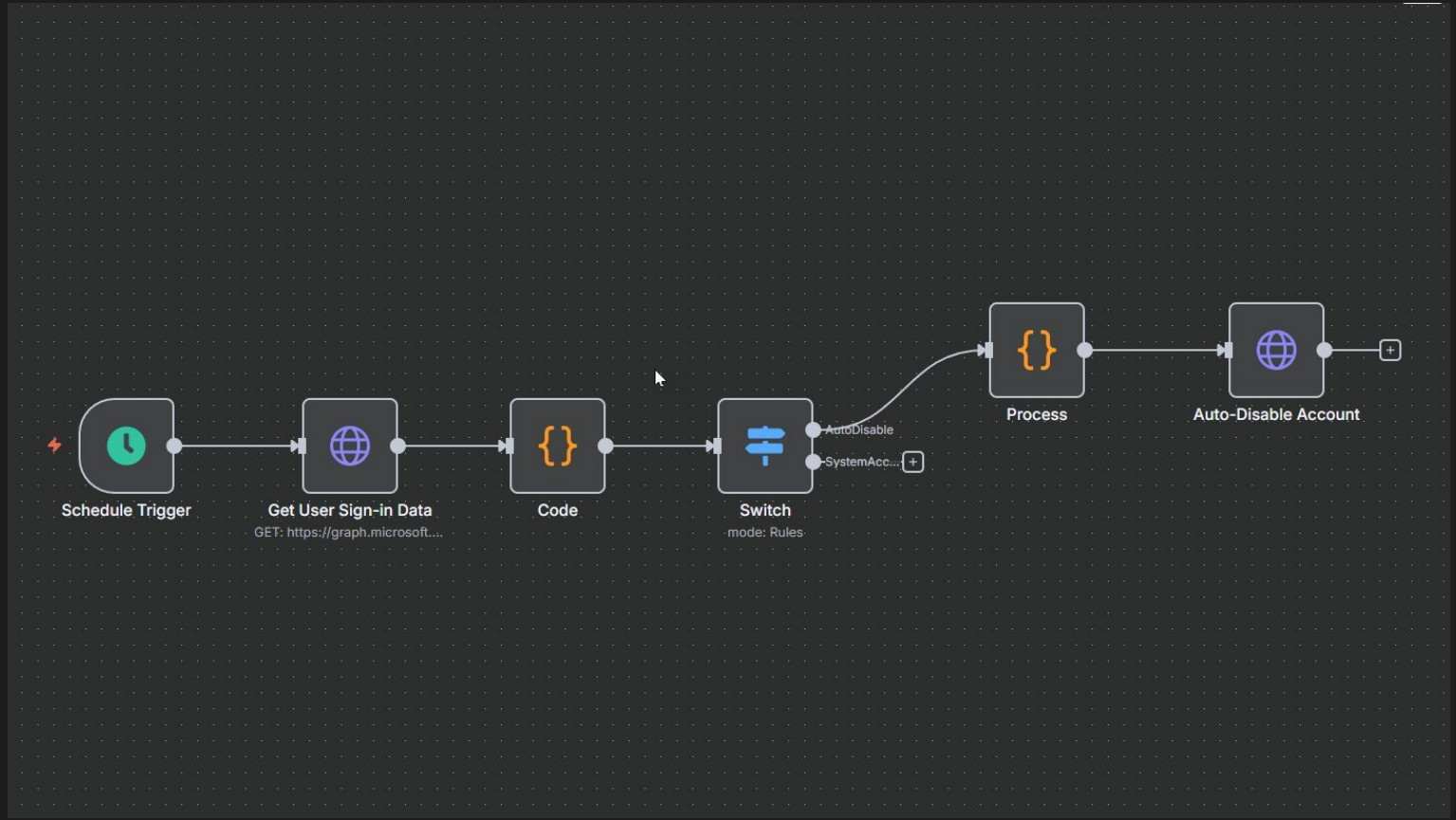
Using AI

Gemini

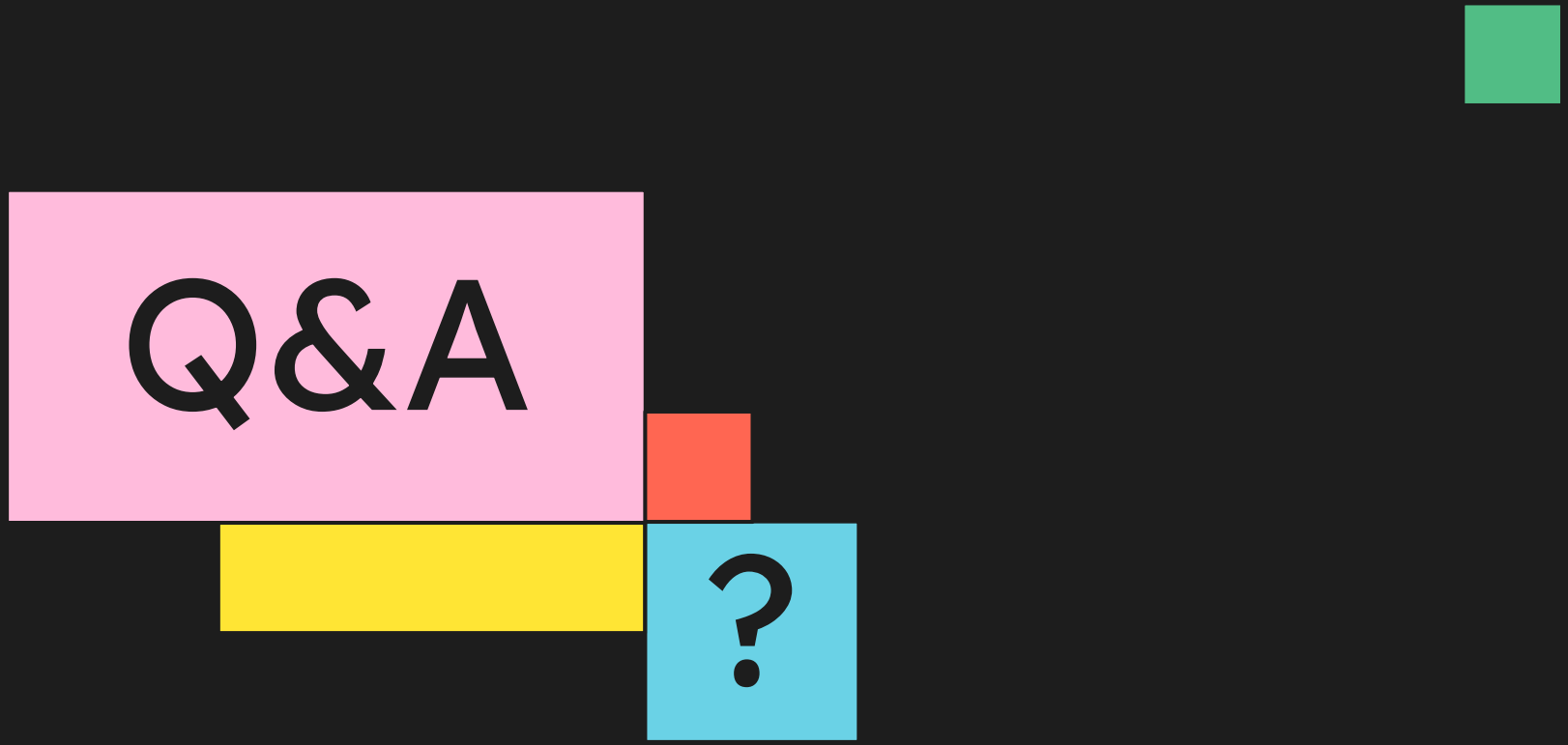
ChatGPT

Claude

Workflow Automation Ideas




PWNED LABS
Sign in
Sign up
Your security training ground
START FOR FREE!
Experience, real-world, byte sized cloud security labs for training cyber warriors. From beginners to pros, our engaging platform allows you to secure your defenses, ignite your career and stay ahead of threats.
Join us at any stage of your journey
PWNED LABS
UNLOCK YOUR CYBER POWER
Subscribe to hackIDLE.com - 3 Memberships for PWNED Labs


Q&A
?