AWS Security Essentials

Aaron Bedra
Chief Scientist, Jemurai
@abedra
keybase.io/abedra

Before we get started

Cloud infrastructure providers offer some incredible benefits

There are loads of reasons to take the leap

But there are a lot of misconceptions about how to approach it

This talk will focus on AWS, but the concepts apply to all providers

It’s easy to get caught up in mirroring your legacy datacenter

Just say no to lift and shift!!!

But you have to realize that this is not a traditional DC

And you shouldn’t
treat it that way

There are aspects
you want to keep

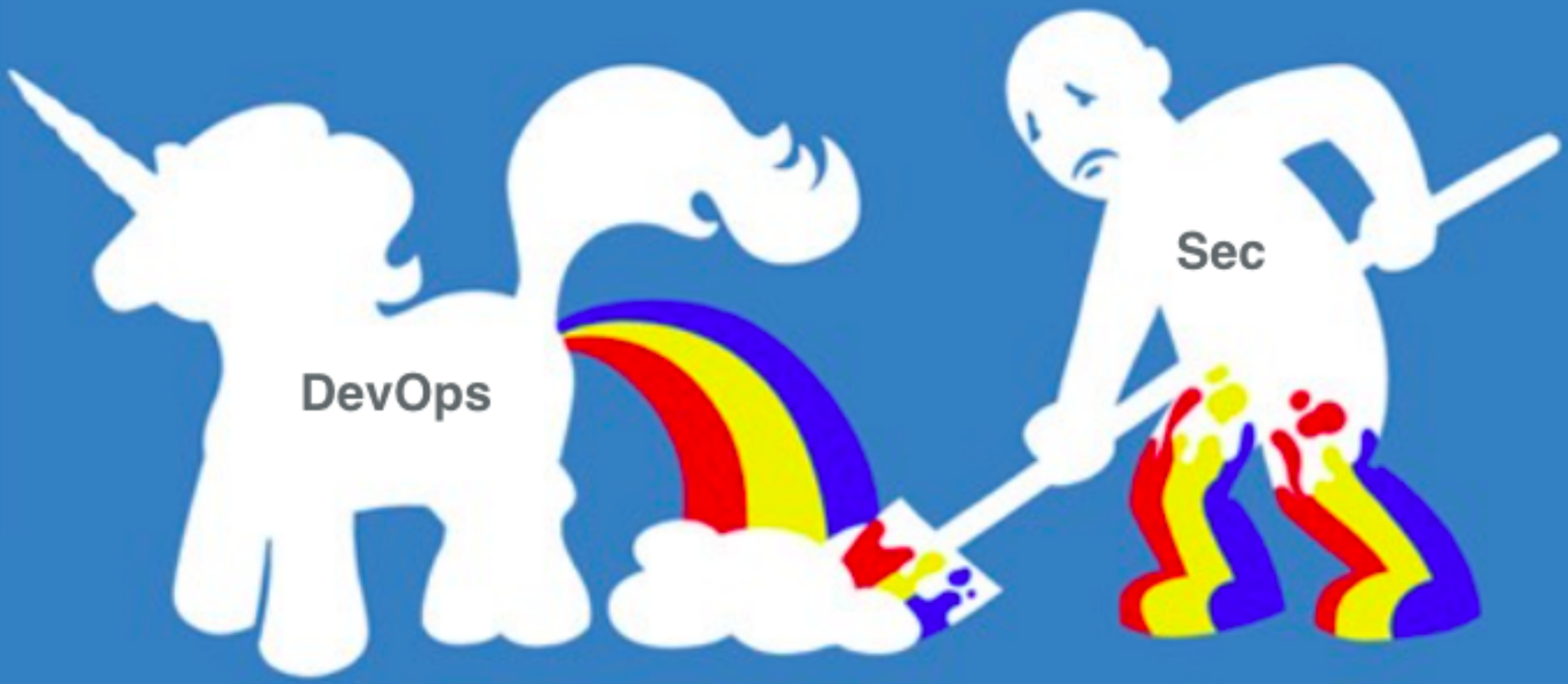
And some you want
to throw away

To get security right in the cloud requires change

Key Areas

- ☑ Automation
- ☑ IAM
- ☑ Network Design
- ☑ Encryption
- ☑ Auditing
- ☑ Continuous Integration

Automation


DevOps
Sec

This is a critical step

There’s no excuse for ignoring automation

The platforms are built for it

Humans clicking buttons is what causes security issues

And what causes blockers, slow down, and frustration

If you desire to move to the cloud and wish to continue clicking buttons, stay where you are

You shouldn’t be logging into the AWS console

In fact, page the security team when it happens

Or just disable console access entirely*

Getting your infrastructure configuration into code is step 1

It allows for review, analysis, and audit

It creates a culture around describing systems as data

The cloud is an abstraction, talk about it as one

It creates a place where simple code review is the only blocker between you and the end result

Automation Checklist

- ☑ All infrastructure is recorded as code
- ☑ All infrastructure changes are made by an automated tool
- ☑ Console logins are restricted to a handful of administrators
- ☑ Teams are educated and empowered to make necessary changes via automation

IAM

Get a grip on users and permissions

I’ve seen some things..

A mistake here could provide control over everything

How do we get to a good place?

Use a directory!

If you already have a directory, replicate it into AWS

Avoid keeping multiple systems of record for user accounts

This solves on-boarding and off-boarding issues

And ensures that changes propagate without additional work

If you don’t have a directory, make sure you setup strong account requirements

The root account

Don’t use it

Page security when it is used

There are only a handful of things you should use the root account for

http://docs.aws.amazon.com/general/latest/gr/aws_tasks-that-require-root.html

Use of the root account isn’t acceptable if it’s not on that list

The only permission IAM accounts should have is assume role

Security Token Service should be the gateway to everything

This reduces the direct exposure of credentials

And forces people to think about the roles they need to perform a task

IAM Checklist

- ☑ Root account has MFA enabled
- ☑ Root account has no access keys*
- ☑ Users have no permissions outside of the ability to use STS to assume roles*
- ☑ Directories are replicated into AWS and used as the system of record
- ☑ MFA is enabled for all human users
- ☑ MFA is required to access privileged roles
- ☑ Users are trained and provided tools to make role assumption seamless
- ☑ Users have no inline policies

Network Design

Network design is situation dependent

But there are a few things that matter

Create a boundary

VPC should be that boundary

Isolate environments and scope with VPCs

Monitor what comes in and out of the VPC

Be conscious about entry points!

There should only be one way in

VPN or Bastion hosts

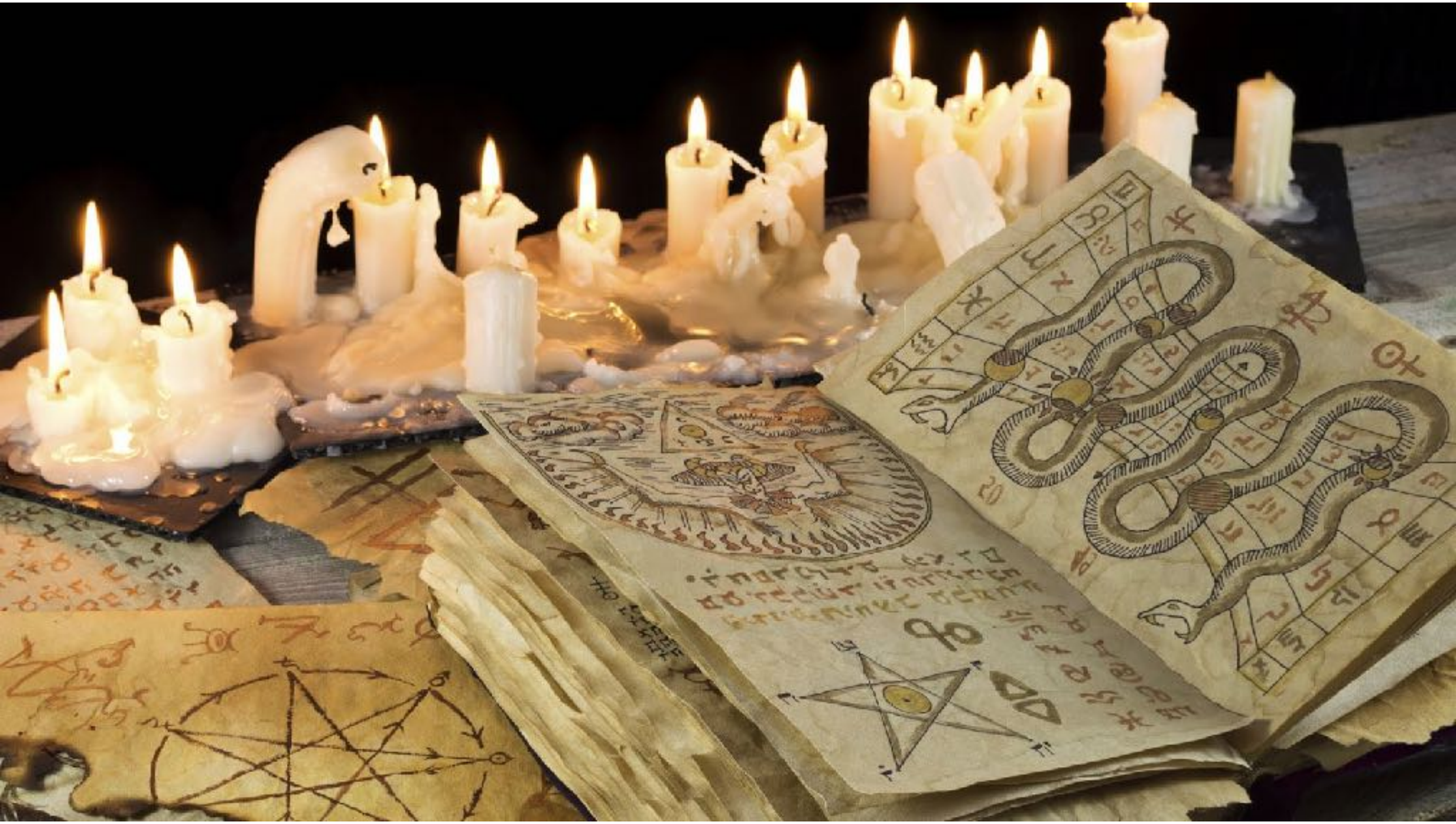
Make sure not to expose management of all machines directly

Use tools to report on external footprint

Network Design Checklist

- ☑ Everything is deployed inside a VPC
- ☑ Flow logs are enabled and monitored
- ☑ Everything that can has a security group attached
- ☑ Any security group that allows access from 0.0.0.0/0 has a detailed description and justification
- ☑ Remote access is only allowed via a bastion host or internal interface

Encryption



We can all acknowledge that this is difficult

But we can make choices that reduce effort and the chance for mistakes

KMS

This is your new default

All your keys should originate from KMS

Any AWS service you use that stores data should have a kms key attached

```
resource "aws_db_instance" "default" {
  allocated_storage     = 10
  storage_type          = "gp2"
  engine                = "mysql"
  engine_version        = "5.6.17"
  instance_class        = "db.t1.micro"
  name                  = "mydb"
  username              = "foo"
  password              = "bar"
  db_subnet_group_name = "my_database_subnet_group"
  parameter_group_name = “default.mysql5.6"
}
```

```
resource "aws_db_instance" "default" {
  allocated_storage     = 10
  storage_type          = "gp2"
  engine                = "mysql"
  engine_version        = "5.6.17"
  instance_class        = "db.t1.micro"
  name                  = "mydb"
  username              = "foo"
  password              = "bar"
  db_subnet_group_name = "my_database_subnet_group"
  parameter_group_name = “default.mysql5.6"
  kms_key_id            = “${aws_kms_key.foo.key_id}”
}
```

There are limitations

When called directly, KMS has a limit of 4 kilobytes per message

But it does allow you to generate data encryption keys

Going this path does push you to write crypto into your code

But it does provide you with strong randomness during key generation

Let’s take a complete look

Start with AWS services encrypted using KMS

If you have to move away use KMS for key encryption keys

Encryption Checklist

- ☑ All Master or Key Encryption Keys are stored using KMS
- ☑ All KMS keys have the rotation option enabled
- ☑ All AWS services utilized that store data should use KMS
- ☑ Data encryption keys are generated using kms master keys and stored encrypted

Auditing

How do you know things are configured correctly?

Scout2

Scout2 audits all configurations across all regions

It produces a report of dangerous issues

Scout2 Analytics Compute Database Management Messaging Network Security Storage Regions Filters Help

Inline group policy allows sts:AssumeRole *

- Policies checked: 0
- Policies flagged: 0

Inline role policy allows sts:AssumeRole *

- Policies checked: 0
- Policies flagged: 0

Inline user policy allows sts:AssumeRole *

- Policies checked: 0
- Policies flagged: 0

Managed policy allows sts:AssumeRole *

- Policies checked: 4
- Policies flagged: 0

Password reuse enabled

- Password policy checked: 1
- Password policy flagged: 0

Root account used recently

- Root account checked: 1
- Root account flagged: 1

Lack of key rotation (Inactive)

- Access keys checked: 2
- Access keys flagged: 0

User without MFA

- Users checked: 1
- Users flagged: 0

Inline role policy allows NotActions

- Policies checked: 0
- Policies flagged: 0

Inline user policy allows NotActions

- Policies checked: 0
- Policies flagged: 0

Managed policy allows NotActions

- Policies checked: 4
- Policies flagged: 0

Minimum password length too short

- Password policy checked: 1
- Password policy flagged: 1

Role with inline policies

- Roles checked: 2
- Roles flagged: 0

Root account has active keys

- Root account checked: 1
- Root account flagged: 0

User with inline policies

- Users checked: 1
- Users flagged: 0

Inline role policy allows iam:PassRole *

- Policies checked: 0
- Policies flagged: 0

Inline user policy allows iam:PassRole *

- Policies checked: 0
- Policies flagged: 0

Managed policy allows iam:PassRole *

- Policies checked: 4
- Policies flagged: 0

Password expiration disabled

- Password policy checked: 1
- Password policy flagged: 1

Lack of MFA (root account)

- Root account checked: 1
- Root account flagged: 0

Lack of key rotation (Active)

- Access keys checked: 2
- Access keys flagged: 0

User with multiple API keys

- Users checked: 1
- Users flagged: 1

Run this tool on your infrastructure and see what you find

You will likely be surprised

Take some time to discuss and correct these issues

This helps with audit of configuration

But what about user activity?

CloudTrail/
CloudWatch

These tools are invaluable

They are an absolute must for anyone taking security seriously

Enable CloudTrail for all active regions

Use CloudWatch to establish alerts on big ticket concerns

Or better yet, use a third party that can do this for you

You don’t have to manage everything on your own

There are services provided by others that do this well

threat stack

Make sure to enable activity logging and create actionable responses to bad actions

Alert Event Examples

- ☑ Root account login
- ☑ Root account key usage
- ☑ New user created
- ☑ User added to administrative roles
- ☑ Too many KMS decrypt events

Amazon GuardDuty

Amazon GuardDuty is a continuous security monitoring and analysis service that detects potential threats to your AWS environment.

Get started

Getting started guide

Continuous

Continuously monitor your AWS environment for suspicious activity and generate findings.

Learn more

Comprehensive

Analyze multiple data sources, including AWS CloudTrail events and VPC Flow Logs.

Learn more

Customizable

Customize GuardDuty by adding your own threat lists and trusted IP lists.

Learn more

Auditing Checklist

- ☑ Configuration analysis is performed at least daily, if not for every change
- ☑ CloudTrail is enabled for all active regions
- ☑ CloudWatch metrics and alarms are implemented for major violation cases
- ☑ Guard Duty?

Continuous Integration

How do I automate it?

Because your infrastructure is in code, you can hook audit into the CI pipeline

Now you can block or rollback changes based on audit findings

You can also institute a seamless sign-off policy that lives in code hosting

alex commented 6 days ago security member

LGTM

compliance-patrol commented 6 days ago security member

✅ **Congratulations, peer review completed** ✅

Please note that any further code change will reset the review status and will require an additional reviews to move forward.

Note: This is an automated message from Eligible pull request validation system.

himanshu-eligible merged commit **803978c** into `master` 6 days ago Hide details Revert

3 checks passed

✓	**ci/circleci_enterprise** Your tests passed on CircleCI Enterprise!	Details
✓	**codeclimate** Code Climate didn't find any new or fixed issues.	Details
✓	**eligible/pull-request** All pull-request checks succeeded!	Details

compliance-patrol removed the **ready for review** label 6 days ago

Pull request successfully merged and closed

You're all set—the `himanshu/SEC-1098` branch can be safely deleted. Delete branch

SEC-1098: Add new endpoints for ssl labs test

Refresh

Repository	security/security_tests
Pull Request	SEC-1098: Add new endpoints for ssl labs test
PR State	**Merged**
Author	himanshu-eligible
Eligible PR requirements	Compliant
Validation results	

Check	Result	Description
Pull-request title format	JIRA found: SEC-1098	*Pull-request titles need to have a JIRA ticket name as a prefix*
Pull-request default title	Custom PR title detected	*Pull-request title should not be the default one from the template*
JIRA link in description	Link to SEC-1098 is present within the description	*Pull-request description needs to have a link to JIRA matching the JIRA id in the title*
SSL-Validation	SSL validation is not ignored in the pull request	*SSL validation should not be ignored*
Peer-reviewed	PR has been reviewed by alex	*Pull requests need to be reviewed by at least one senior engineer before merging*
Migration-reviewed	Pull request does not include any migrations	*Migrations should be reviewed by Technical Operations Team*
Security-reviewed	Pull request does not require security review.	*Changes should be reviewed by Security Team*
PR-link-on-jira	PR link found in comments of all attached JIRA tickets	*PR link should be on jira task*

CI Checklist

- ☑ All infrastructure changes trigger configuration audits
- ☑ Critical issues found in CI trigger immediate response
- ☑ Use the CI pipeline to create a sign-off process that allows teams to move faster

The benefits of being on provider like AWS are massive

If there is still fear of shifting, please look again!

Healthcare, Finance, Government, etc. are there already

Questions?