Firespring®

Lincoln Devops Meetup
April 3, 2019

Securing your AWS environment

Why is it important?

So much of our security mindset focuses on our code and applications. From things like the OWASP Top 10 to every scanning tool on the market, much attention is paid to the front door.

AWS account security is a foundational aspect of securing your company.

- Protect the keys to the kingdom.
- Protect against intrusion.
- Protect against accidents by internal users.

Foundational:

AWS Security Best Practices (highlights)

- **Manage AWS Accounts, IAM Users, Groups and Roles**
- **Manage OS-level access to Amazon EC2 Instances**

Foundational:

AWS Security Best Practices (highlights)

- Secure your data
  - Resource access
  - Managing encryption keys
  - Protecting data at rest
  - Protect data in transit
  - Secure your OS
  - Manage security monitoring, alerting, audit trail, etc.

Foundational:

<Continue long list of items to do>

OR….

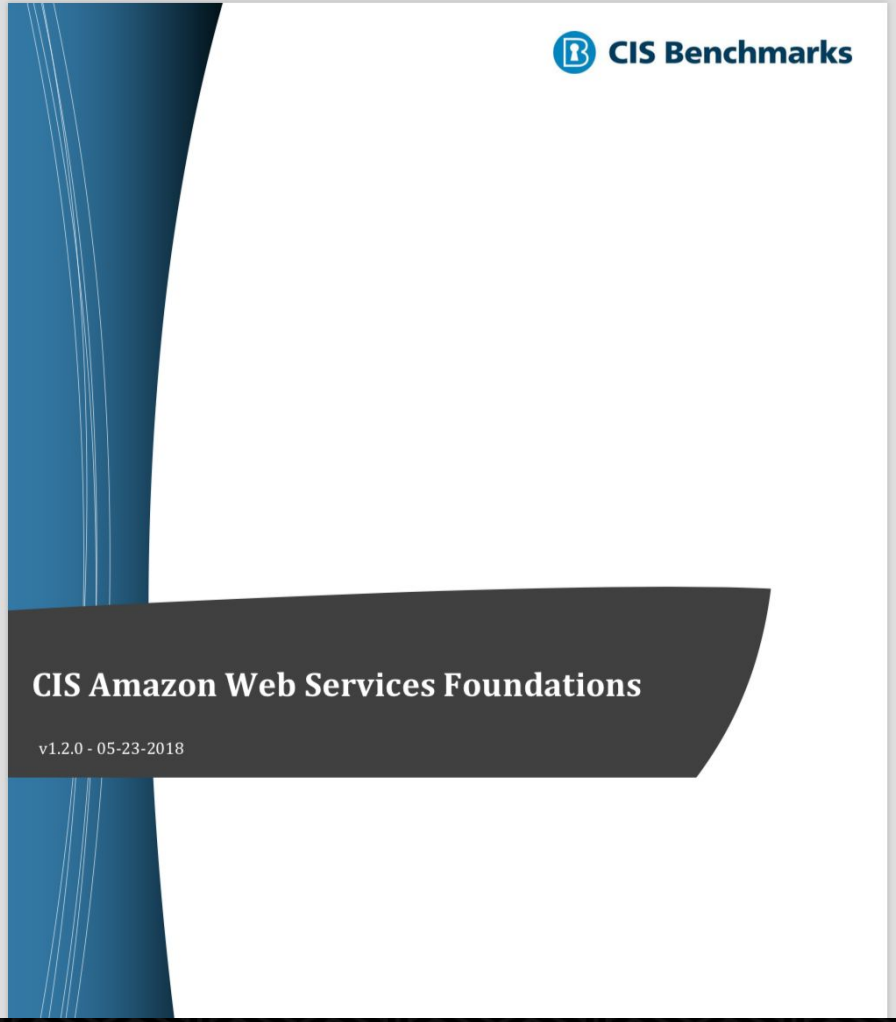
Firespring

CIS Benchmarks™

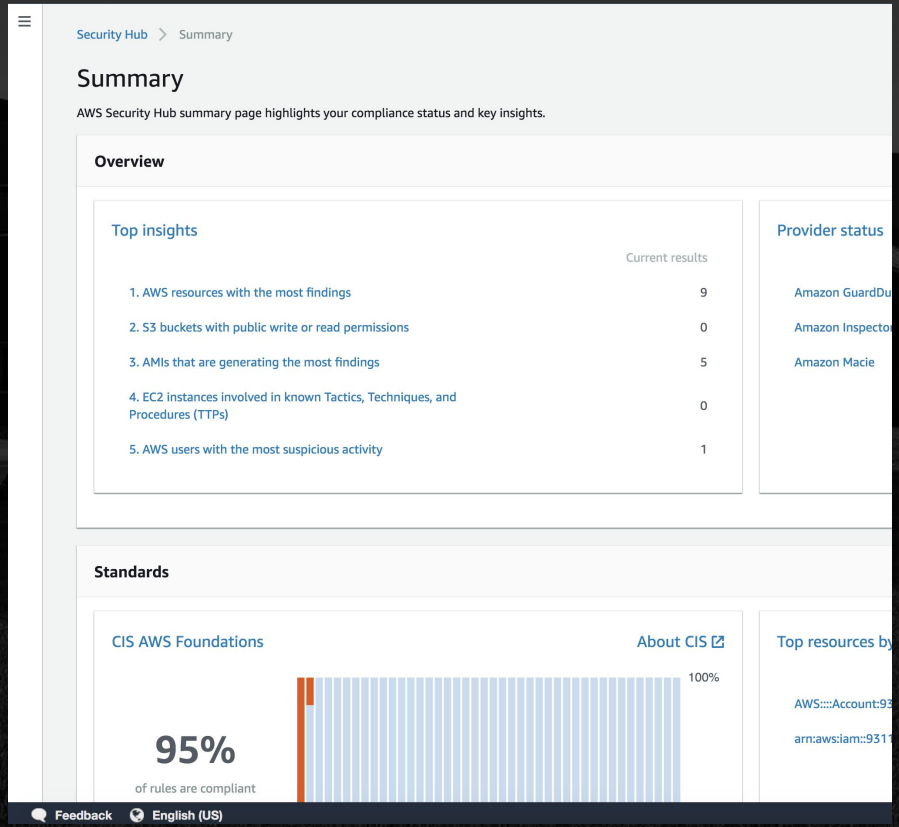
CIS Amazon Web Services Foundations

(The best starting point that we've seen)

Firespring


CIS Benchmarks
CIS Amazon Web Services Foundations
v1.2.0 - 05-23-2018

AWS Security Hub



Firespring

Continuing with other rapid fire best practices...

Firespring

Encrypt all the things.

(in transit and at rest)

Don't reinvent encryption.

(Just use KMS)

Automate all SSL certificate renewal.

Use Inspector and patch religiously.

Log all the things (Cloudwatch, Cloudtrail, etc.).

Consider using AWS services (RDS, Fargate, etc.) when running smaller scale apps.

Limit egress traffic by using VPC Endpoints.

Consider using SSM Session Manager instead of bootstrapping or distributing SSH keys.

Use instance roles for all instance access to AWS services.

Use separate accounts for Prod, Dev, etc.

Join multiple accounts inside of an Organization.

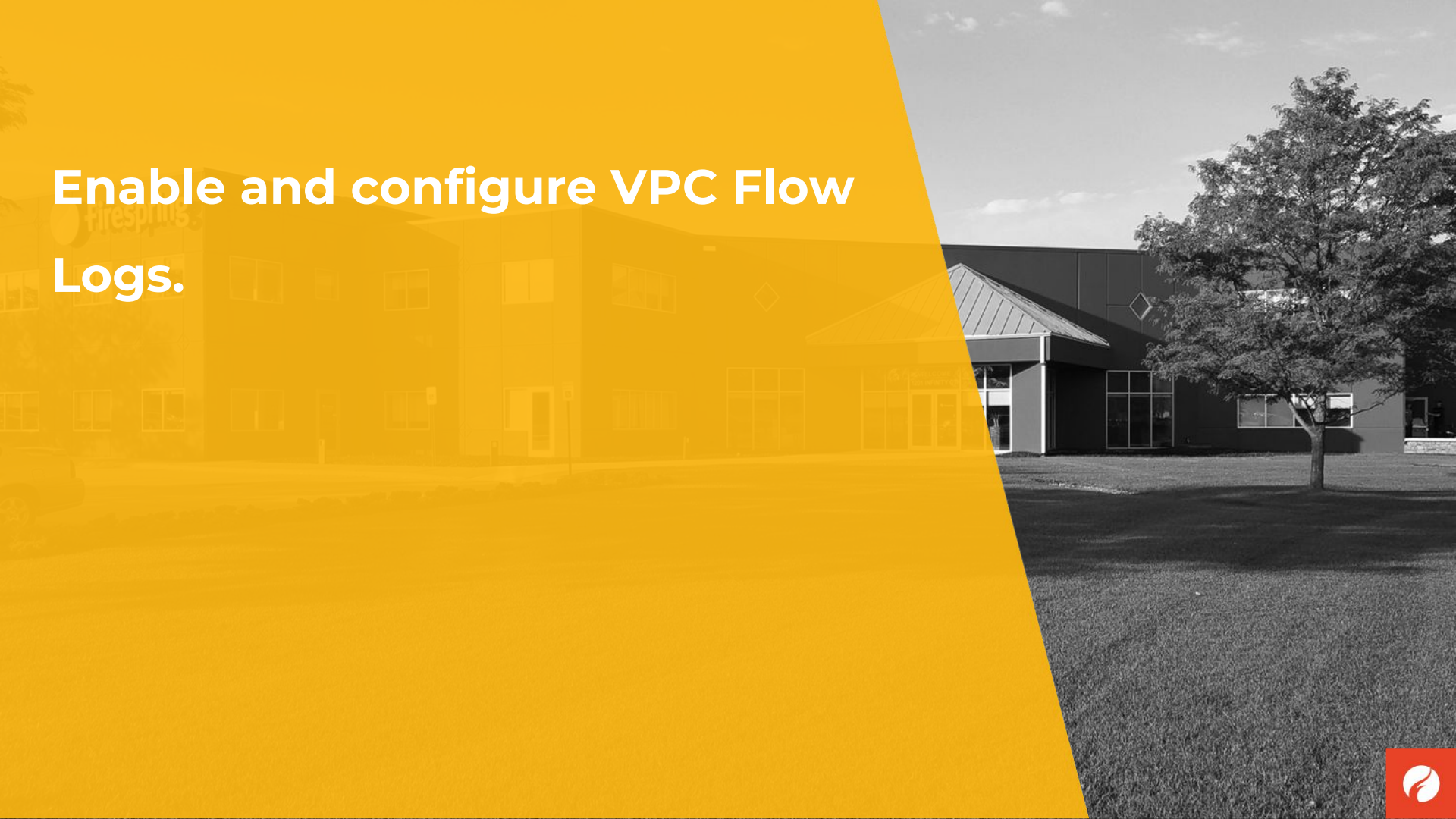
Ensure no S3 buckets are set to publicly available unless there is a *very* good reason.

Implement infrastructure as code whenever possible (Cloudformation or Terraform).

Start small and grow it over time if you've already built it all by hand.

Regularly rotate all credentials.

Enable and configure VPC Flow Logs.



Not everyone needs to be an admin.

(no, really...)

Strongly consider attending re:Invent (or a similar AWS conference).

Enforce 2FA on all user interactions.

Overview:

Cloudformation

- Necessary groups, policies, and users are added
- All groups have the policy to require login to be 2FA
    - Allow login and change password
    - Allow management of credentials and 2FA
    - Deny everything else if not logged in with 2FA
- 2fa_login_iam.yml
- Additional AWS 2FA Documentation

Process:

New AWS Web Console User

- Create the new user in CF
- Provision temporary login credentials
- User logs in, updates password, configures 2FA
- User logs out and logs back in using 2FA
    - User now has access to policies defined for their group

Process:

New AWS CLI User

- Create Access Key in IAM Dashboard
- Replace placeholders in CLI login script
    - 2fa_login_cli.sh
    - Store the updated script in a secure place
        - We store it as a secure note in lastpass

Process:

AWS CLI User Login

- Execute the login script
  - Turns off command history
  - Turns off console output
  - Sets your Access Key in the environment
  - Turns on console output
  - Collects the current 2FA code
  - Requests a session token from STS
  - Sets new credentials in the environment
  - Turns on command history
- Credentials will be valid for 8 hours