CLOUD SECURITY
ASSESSMENT REPORT

AWS Infrastructure Security Assessment

TechStart Solutions

February 24, 2026 — February 27, 2026
Version 1.0.0

Prepared by
System Administrator

2. Document Control

2.1. Classification & Handling

Classification	Handling Instruction
CONFIDENTIAL	This document must not be shared outside the authorized distribution list. Electronic copies must be stored in encrypted form. Physical copies must be stored in locked containers.

2.2. Document History

Version	Date	Author	Changes
1.0	2026-02-27	System Administrator	Initial release

2.3. Distribution List

Recipient	Role	Access Level
Michael Brown	Cloud Security Lead	Full Report
CISO / Security Director	Executive Oversight	Full Report
DevOps / Platform Engineering	Remediation	Technical Sections
Compliance / GRC Team	Regulatory Alignment	Executive Summary

2. Table of Contents

3. Executive Summary

A comprehensive AWS cloud security assessment was conducted on the production infrastructure of Meridian Financial Group across three AWS accounts (Production, Staging, Shared Services). The assessment evaluated the security posture of the cloud environment against the CIS Amazon Web Services Foundations Benchmark v3.0, the AWS Well-Architected Security Pillar, and NIST Cybersecurity Framework controls.

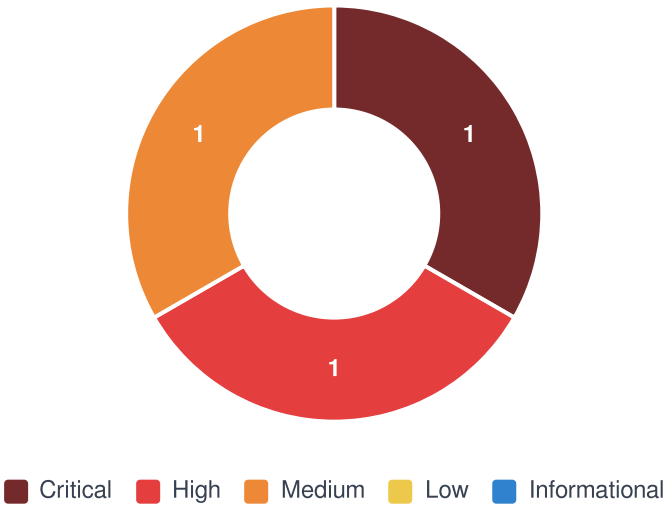
The assessment identified multiple critical and high-severity findings across IAM configuration, network segmentation, data protection, and logging practices. Key concerns include overly permissive IAM roles with wildcard permissions, publicly accessible S3 buckets containing sensitive data, insufficient encryption controls, missing CloudTrail log integrity validation, and unrestricted security group rules allowing lateral movement between production and non-production workloads.

Immediate remediation of Critical and High severity findings is strongly recommended to reduce the risk of unauthorized data access, privilege escalation, and potential regulatory non-compliance (SOC 2, PCI DSS).

3.1. Risk Dashboard

A total of **3** findings were identified across the AWS infrastructure of TechStart Solutions during the assessment period **February 24, 2026** to **February 27, 2026**. The charts below illustrate the severity distribution and affected service domains.

3.1.1. Findings by Severity



Based on the assessment findings, the following strategic recommendations are provided:

- **Implement least-privilege IAM governance** — Establish mandatory permission boundaries, eliminate wildcard policies, and deploy IAM Access Analyzer across all accounts
- **Enforce network micro-segmentation** — Replace permissive security group rules with purpose-specific rules, implement VPC endpoints for AWS service access, and deploy AWS Network Firewall for cross-VPC traffic inspection
- **Centralize logging and detection** — Enable CloudTrail in all regions with log file validation, activate GuardDuty across the organization, and implement automated alerting for high-severity findings
- **Adopt infrastructure-as-code security scanning** — Integrate Terraform/CloudFormation scanning (Checkov, cfn-nag) into CI/CD pipelines to prevent misconfigurations before deployment
- **Establish a cloud security baseline** — Deploy AWS Config conformance packs aligned with CIS Benchmark and schedule quarterly reassessments

4. Assessment Scope

The assessment covered the Meridian Financial Group AWS Organization consisting of three accounts: **Production** (743XXXXXXXX), **Staging** (891XXXXXXXX), and **Shared Services** (562XXXXXXXX). The scope included all deployed AWS services, cross-account trust relationships, network connectivity, data storage configurations, and security monitoring posture. The assessment was performed using authenticated access with read-only SecurityAudit permissions across all accounts.

4.1. AWS Accounts in Scope

Account ID	Account Name	Environment

4.2. AWS Services Evaluated

The following AWS services were evaluated during the assessment:

- **Identity & Access:** IAM, AWS SSO (Identity Center), AWS Organizations, STS, Cognito
- **Compute:** EC2, ECS/Fargate, Lambda, EKS, Auto Scaling
- **Networking:** VPC, Security Groups, NACLs, ALB/NLB, CloudFront, Route 53, Transit Gateway, WAF
- **Storage & Database:** S3, EBS, RDS (PostgreSQL, Aurora), DynamoDB, ElastiCache
- **Encryption & Secrets:** KMS, Secrets Manager, ACM
- **Monitoring & Logging:** CloudTrail, CloudWatch, GuardDuty, AWS Config, Security Hub, VPC Flow Logs
- **Deployment:** CodePipeline, CodeBuild, ECR, CloudFormation

5. AWS Environment Overview

Meridian Financial Group operates a multi-account AWS Organization managed through AWS Control Tower. The environment hosts customer-facing banking applications, internal tools, and data analytics workloads. Production workloads process approximately 2.4 million transactions daily and store sensitive financial data subject to SOC 2 Type II and PCI DSS v4.0 compliance requirements.

Organization ID	o-abcd1234ef
Primary Region	us-east-1 (N. Virginia)
Secondary Region	us-west-2 (Oregon) — DR
AWS Accounts	3 (Production, Staging, Shared Services)
VPCs	7 (connected via Transit Gateway)
EC2 Instances	142 (Production: 89, Staging: 38, Shared: 15)
ECS Services	23 Fargate services across 4 ECS clusters
Lambda Functions	67 (12 public-facing via API Gateway)
S3 Buckets	94 (including 3 identified as publicly accessible)
RDS Instances	8 (Aurora PostgreSQL clusters + standalone RDS)
IAM Users	34 (22 with console access, 12 programmatic only)
IAM Roles	187 (including 14 cross-account trust roles)

5.1. Architecture Summary

The architecture follows a hub-and-spoke model with the Shared Services account acting as the central hub. All VPCs are connected via AWS Transit Gateway with route tables controlling traffic flow between environments. Internet-facing workloads are fronted by Application Load Balancers with AWS WAF protection. Internal services communicate via VPC endpoints and private DNS zones managed through Route 53 Hosted Zones.

Container workloads run on ECS Fargate with images stored in private ECR repositories. CI/CD is managed through CodePipeline with CodeBuild stages for testing and deployment. Infrastructure is provisioned via Terraform with state files stored in a dedicated S3 backend with DynamoDB locking.

6. Assessment Methodology

The assessment was performed using a structured methodology aligned with the CIS Amazon Web Services Foundations Benchmark, AWS Well-Architected Security Pillar, and NIST CSF. Each phase targets a specific layer of the cloud infrastructure.

6.1. Assessment Phases

Phase	Domain	Focus Areas	Frameworks
1	**Identity & Access** IAM, SSO, Organizations	IAM policies, role trust, cross-account access, MFA enforcement, privilege escalation paths, permission boundaries	CIS 1.x, NIST AC
2	**Network & Infrastructure** VPC, Security Groups, WAF	VPC segmentation, SG rules, NACLs, VPC peering, Transit Gateway, public exposure, WAF bypass	CIS 5.x, NIST SC
3	**Data Protection** S3, RDS, KMS, Secrets Manager	S3 bucket policies, encryption at rest/transit, KMS key rotation, public bucket exposure, database exposure	CIS 2.x, NIST SC
4	**Logging & Monitoring** CloudTrail, GuardDuty, Config	CloudTrail coverage, log integrity, GuardDuty findings, Config rules, alarm coverage, incident response readiness	CIS 3.x-4.x, NIST AU
5	**Compute & Containers** EC2, ECS, Lambda, EKS	Instance metadata hardening (IMDSv2), container image scanning, Lambda permissions, EKS RBAC, patch compliance	CIS 2.x, NIST CM

6.2. Tools & Techniques

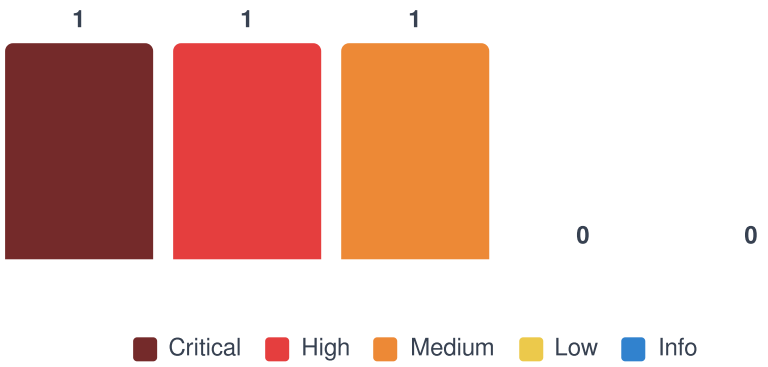
The following tools and techniques were utilized during the assessment:

- **Prowler** — Comprehensive AWS security assessment tool aligned with CIS Benchmark, PCI DSS, and NIST 800-53
- **ScoutSuite** — Multi-cloud security auditing tool for cross-service configuration analysis
- **Pacu** — AWS exploitation framework for simulating attacker techniques and privilege escalation paths
- **CloudMapper** — AWS environment visualization and attack surface mapping
- **IAM Access Analyzer** — Native AWS tool for identifying resource policies that grant external access
- **Steampipe** — SQL-based cloud resource querying for custom compliance checks
- **Cloudsplaining** — IAM policy analysis for identifying privilege escalation and data exfiltration risks
- **S3Scanner** — S3 bucket discovery and permission enumeration
- **AWS CLI** / **Boto3** — Manual verification and custom enumeration scripts
- **Checkov** — Infrastructure-as-code static analysis for Terraform and CloudFormation misconfigurations

7. Findings Summary

The following dashboard provides a consolidated view of all identified findings organized by severity. The table below includes CIS Benchmark mapping and affected AWS service for each finding.

7.0.1. Severity Distribution



7.1. All Findings

#	Finding	AWS Service	CIS Control	Severity	Score
1	Public S3 Bucket with Sensitive Data	S3	CIS 2.1.1 - Ensure S3 Bucket Policy is set to deny HTTP requests	Critical	9.1
2	IAM Overly Permissive Policies	IAM	CIS 1.16 - Ensure IAM policies that allow full administrative privileges are not attached	High	8.8
3	EC2 Instance Metadata Service v1 Enabled	EC2	CIS 5.6 - Ensure IMDSv2 is enabled on all EC2 instances	Medium	6.8

8. Public S3 Bucket with Sensitive Data

Critical	**CVSS 9.1** — Critical	S3

CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
CIS Control	CIS 2.1.1 - Ensure S3 Bucket Policy is set to deny HTTP requests
Affected Resource	arn:aws:s3:::company-backup-prod

8.1. Overview

The S3 bucket `company-backup-prod` was found to have a bucket policy granting public read access via the `s3:GetObject` action to the principal `"*"`. The bucket contains database backups, application configuration files, and exported customer data in CSV format. The Block Public Access settings at both the account and bucket level were not enabled.
Additionally, the bucket does not enforce encryption at rest using SSE-S3 or SSE-KMS, and the bucket policy does not require TLS for data in transit (no `aws:SecureTransport` condition).

8.2. Evidence & Exploitation

1. Identified the bucket during passive reconnaissance using common naming patterns:

```
aws s3 ls s3://company-backup-prod --no-sign-request
```

2. Listed and downloaded objects without any authentication:

```
aws s3 cp s3://company-backup-prod/exports/customers_2024.csv . --no-sign-request
```

3. The exported file contained 15,000+ customer records including names, email addresses, phone numbers, and billing information.

8.3. Business Impact

The publicly accessible bucket exposes sensitive business data and customer PII to any internet user. Database backups may contain credentials, API keys, and application secrets. Customer data exposure creates regulatory liability under GDPR, CCPA, and other privacy frameworks. The lack of encryption and TLS enforcement compounds the data protection failures.

8.4. Remediation

Immediately restrict bucket access and enable protective measures:

```
# Block all public access
aws s3api put-public-access-block --bucket company-backup-prod   --public-access-block-
configuration
"BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=t
rue"

# Enable default encryption
aws s3api put-bucket-encryption --bucket company-backup-prod   --server-side-encryption-
configuration   '{"Rules":[{"ApplyServerSideEncryptionByDefault":
{"SSEAlgorithm":"aws:kms"}}]}'
```

Enable S3 Block Public Access at the account level to prevent future misconfigurations. Enable CloudTrail S3 data events to audit access and detect unauthorized downloads of sensitive objects.

8.4.1. AWS CLI / IaC Fix

```
# Terraform - Secure S3 bucket configuration
resource "aws_s3_bucket_public_access_block" "backup" {
  bucket = aws_s3_bucket.backup.id

  block_public_acls       = true
  block_public_policy     = true
  ignore_public_acls      = true
  restrict_public_buckets = true
}

resource "aws_s3_bucket_server_side_encryption_configuration" "backup" {
  bucket = aws_s3_bucket.backup.id

  rule {
    apply_server_side_encryption_by_default {
      sse_algorithm     = "aws:kms"
      kms_master_key_id = aws_kms_key.s3.arn
    }
  }
}

resource "aws_s3_bucket_policy" "enforce_tls" {
  bucket = aws_s3_bucket.backup.id
  policy = jsonencode({
    Version = "2012-10-17"
    Statement = [{
      Sid       = "EnforceTLS"
      Effect    = "Deny"
      Principal = "*"
      Action    = "s3:*"
      Resource  = [
        aws_s3_bucket.backup.arn,
        "${aws_s3_bucket.backup.arn}/*"
      ]
      Condition = {
        Bool = { "aws:SecureTransport" = "false" }
      }
    }]
  })
}
```

8.5. References

AWS - S3 Block Public Access
CIS AWS Foundations Benchmark

9. IAM Overly Permissive Policies

High	**CVSS 8.8** — High	IAM

CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
CIS Control	CIS 1.16 - Ensure IAM policies that allow full administrative privileges are not attached
Affected Resource	arn:aws:iam::123456789012:policy/DeveloperFullAccess

9.1. Overview

Multiple IAM policies were identified granting overly broad permissions, including `"Action": "*"` on `"Resource": "*"`. The custom policy `DeveloperFullAccess` is attached to 12 IAM users and 3 IAM roles used by development teams. Several IAM users also have both console access and long-lived access keys, with no MFA enforcement.

The AWS account does not use AWS Organizations SCPs to restrict available actions, and IAM Access Analyzer is not enabled to detect externally shared resources.

9.2. Evidence & Exploitation

1. Enumerated IAM policies and found wildcard permissions:

```
aws iam list-policies --scope Local --query 'Policies[].PolicyName'
aws iam get-policy-version --policy-arn
arn:aws:iam::123456789012:policy/DeveloperFullAccess   --version-id v1
```

2. Using compromised developer credentials (from a leaked .env file in a public repository), escalated privileges by creating a new IAM user with admin access:

```
aws iam create-user --user-name backdoor-admin
aws iam attach-user-policy --user-name backdoor-admin   --policy-arn
arn:aws:iam::aws:policy/AdministratorAccess
```

3. The new admin user had full control over all AWS resources in the account.

9.3. Business Impact

Overly permissive IAM policies violate the principle of least privilege. If any user or role with these permissions is compromised, the attacker gains full administrative control over all AWS resources. This includes the ability to access and exfiltrate data from all services, create backdoor accounts, modify security configurations, launch resources for cryptocurrency mining, and delete critical infrastructure.

9.4. Remediation

Replace wildcard policies with least-privilege policies scoped to specific services, actions, and resources. Use IAM Access Analyzer to generate policies based on actual usage patterns.

9.4.1. AWS CLI / IaC Fix

```
# Generate a least-privilege policy from CloudTrail activity
aws accessanalyzer generate-policy --policy-generation-details '{
  "principalArn": "arn:aws:iam::123456789012:role/DevRole",
  "trailProperties": [{
    "cloudTrailArn": "arn:aws:cloudtrail:us-east-
1:123456789012:trail/main",
    "regions": ["us-east-1"]
  }]
}'

# Example scoped policy for a developer role
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject", "s3:PutObject", "s3:ListBucket",
        "lambda:InvokeFunction", "lambda:UpdateFunctionCode",
        "logs:GetLogEvents", "logs:FilterLogEvents"
      ],
      "Resource": [
        "arn:aws:s3:::dev-*",
        "arn:aws:lambda:us-east-1:123456789012:function:dev-*",
        "arn:aws:logs:us-east-1:123456789012:log-group:/aws/lambda/dev-*"
      ]
    }
  ]
}
```

9.5. References

AWS IAM Best Practices
AWS IAM Access Analyzer

10. EC2 Instance Metadata Service v1 Enabled

Medium	**CVSS 6.8** — Medium	EC2

CVSS Vector	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:N/A:N
CIS Control	CIS 5.6 - Ensure IMDSv2 is enabled on all EC2 instances
Affected Resource	i-0a1b2c3d4e5f67890 (web-prod-01), i-0f9e8d7c6b5a43210 (api-prod-01)

10.1. Overview

Multiple EC2 instances in the production environment have the Instance Metadata Service version 1 (IMDSv1) enabled. IMDSv1 responds to simple HTTP GET requests to the link-local address `169.254.169.254` without requiring a session token. This makes it vulnerable to Server-Side Request Forgery (SSRF) attacks, where an attacker can trick the application into making requests to the metadata endpoint.

The instances use IAM instance roles with permissions to access S3 buckets and DynamoDB tables. If an attacker exploits an SSRF vulnerability in any application running on these instances, they can retrieve temporary IAM credentials from the metadata service and use them to access AWS resources from outside the instance.

10.2. Evidence & Exploitation

1. Identified an SSRF vulnerability in the web application running on the EC2 instance.
2. Used the SSRF to query the IMDSv1 endpoint:

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/
# Response: WebAppRole

curl http://169.254.169.254/latest/meta-data/iam/security-credentials/WebAppRole
```

3. Retrieved temporary AWS credentials (AccessKeyId, SecretAccessKey, Token) valid for 6 hours.
4. Used the credentials externally to access S3 production data:

```
AWS_ACCESS_KEY_ID=ASIA... AWS_SECRET_ACCESS_KEY=... AWS_SESSION_TOKEN=...   aws s3 ls
s3://company-prod-data/
```

10.3. Business Impact

IMDSv1 combined with SSRF vulnerabilities provides a direct path to AWS credential theft. The temporary credentials inherit all permissions of the instance's IAM role, which typically includes access to S3, DynamoDB, SQS, and other services critical to the application. An attacker can use these credentials from any location until they expire, accessing, modifying, or deleting cloud resources.

10.4. Remediation

Enforce IMDSv2 on all EC2 instances. IMDSv2 requires a session token obtained via a PUT request with a TTL header, which cannot be triggered via typical SSRF vectors:

10.4.1. AWS CLI / IaC Fix

```
# Enforce IMDSv2 on existing instances
aws ec2 modify-instance-metadata-options   --instance-id i-
0a1b2c3d4e5f67890   --http-tokens required   --http-endpoint enabled   --
http-put-response-hop-limit 1

# Terraform - Enforce IMDSv2 for all new instances
resource "aws_instance" "web" {
  ami           = "ami-0123456789abcdef0"
  instance_type = "t3.medium"

  metadata_options {
    http_tokens                 = "required"  # Enforce IMDSv2
    http_endpoint               = "enabled"
    http_put_response_hop_limit = 1
  }
}

# AWS Config rule to detect non-compliant instances
resource "aws_config_config_rule" "imdsv2" {
  name = "ec2-imdsv2-check"
  source {
    owner             = "AWS"
    source_identifier = "EC2_IMDSV2_CHECK"
  }
}
```

10.5. References

AWS - Instance Metadata Service v2
AWS Security Blog - SSRF and IMDS

11. Compliance Gap Analysis

Each finding has been mapped to the applicable CIS Amazon Web Services Foundations Benchmark v3.0 control. This mapping enables prioritization of remediation efforts that simultaneously address security vulnerabilities and compliance gaps. Findings that affect PCI DSS or SOC 2 controls are additionally flagged in the detailed finding sections with the relevant control reference.

11.1. Finding-to-Control Mapping

CIS Control	Section	Finding	Gap Status	Severity
CIS 1.16 - Ensure IAM policies that allow full administrative privileges are not attached	IAM	IAM Overly Permissive Policies	Non-Compliant	High
CIS 2.1.1 - Ensure S3 Bucket Policy is set to deny HTTP requests	S3	Public S3 Bucket with Sensitive Data	Non-Compliant	Critical
CIS 5.6 - Ensure IMDSv2 is enabled on all EC2 instances	EC2	EC2 Instance Metadata Service v1 Enabled	Non-Compliant	Medium

12. Appendix

12.1. Assessment Tool Reference

The following tools were used during the assessment. Each tool targets a specific layer of AWS infrastructure security.

Tool	Purpose	Coverage
Prowler	CIS Benchmark, PCI DSS, NIST 800-53 automated compliance scanning	IAM, S3, CloudTrail, VPC, EC2
ScoutSuite	Multi-service configuration audit with risk-scored findings	All AWS services
Pacu	AWS post-exploitation: privilege escalation, data exfiltration simulation	IAM, Lambda, EC2, S3
Steampipe	SQL-based cloud resource querying for custom compliance checks	All AWS services
Cloudsplaining	IAM policy analysis for privilege escalation and data exfiltration paths	IAM
CloudMapper	AWS environment visualization and network attack surface mapping	VPC, EC2, ELB

12.2. Prowler Severity Mapping

Prowler findings were mapped to the CVSS-based severity scale used throughout this report:

Prowler Level	CVSS Range	Report Severity	Example
critical	9.0 – 10.0	Critical	Root account without MFA, publicly accessible RDS with default creds
high	7.0 – 8.9	High	IAM user with wildcard admin policy, S3 bucket with public read
medium	4.0 – 6.9	Medium	CloudTrail not enabled in all regions, SG allowing 0.0.0.0/0 on non-critical port
low	0.1 – 3.9	Low	Access keys older than 90 days, unused IAM roles
informational	N/A	Info	Best practice: enable AWS Config, use SCPs for guardrails