HORIZONSHIELD

Securing Your Digital Future, Today.

CLOUD SECURITY SERIES · BOOK 05 OF 16

Cloud Security Hardening Guide

AWS · Azure · GCP — Complete Configuration Reference 2026

Comprehensive hardening for all three major cloud platforms. IAM, network security, storage, compute, logging, and CSPM — with actionable CLI commands, CIS Benchmark mappings, and misconfigurations to avoid.

PLATFORMS	CONTROLS	STANDARD	LEVEL
3	200+	CIS Benchmarks	Intermediate

Contents

CLOUD SECURITY HARDENING 2026

CHAPTER 01

Shared Responsibility Model

Misunderstanding this boundary causes the majority of cloud data breaches. The cloud provider secures the infrastructure. You secure everything you put on it.

AWS AZURE GCP SHARED RESPONSIBILITY

The Shared Responsibility Model

The cloud provider's security does not protect you from your own misconfiguration. A public S3 bucket, overpermissioned IAM role, or unpatched EC2 instance are all customer responsibility failures — the provider's security is irrelevant to them.

Domain	IaaS (EC2/VM/GCE)	PaaS (RDS/App Service)	SaaS (M365/GWorkspace)	Key Implication
Physical infrastructure	Provider	Provider	Provider	Never your responsibility
Hypervisor	Provider	Provider	Provider	Cannot access hypervisor layer
OS patching	YOUR RESPONSIBILITY	Provider	Provider	CRITICAL — patch EC2/VM yourself
Application code	YOUR RESPONSIBILITY	YOUR RESPONSIBILITY	Provider	Your code security is always yours
Data encryption	YOUR RESPONSIBILITY	YOUR RESPONSIBILITY	YOUR RESPONSIBILITY	You always own your data

Domain	IaaS (EC2/VM/GCE)	PaaS (RDS/App Service)	SaaS (M365/GWorkspace)	Key Implication
IAM configuration	YOUR RESPONSIBILITY	YOUR RESPONSIBILITY	YOUR RESPONSIBILITY	IAM is always your configuration choice
Network security (VPC)	YOUR RESPONSIBILITY	YOUR RESPONSIBILITY	N/A	Security groups, NACLs, flow logs — all yours

#1
PUBLIC STORAGE BUCKETS
39% of cloud breaches

#2
OVERPRIVILEGED IAM — WILDCARD * PERMISSIONS
34% of cloud breaches

#3
NO MFA ON ROOT/GLOBAL ADMIN
28% of cloud breaches

#4
SECURITY GROUPS OPEN 0.0.0.0/0 ON SSH/RDP
22% of cloud breaches

CSPM IS NON-NEGOTIABLE

Deploy a Cloud Security Posture Management tool (AWS Security Hub, Microsoft Defender for Cloud, Wiz, Prisma Cloud) BEFORE deploying workloads. CSPM continuously scans your cloud configuration for misconfigurations — catching the human errors that cause 95% of cloud breaches.

CHAPTER 02

IAM Hardening — All Three Platforms

Cloud IAM misconfiguration is the leading cause of cloud breaches. Least privilege and MFA for all privileged access.

LEAST PRIVILEGE | MFA | SERVICE ACCOUNTS | ROOT ACCOUNT

AWS IAM Hardening

```
# Root account — NEVER use for daily operations
# Check if root has MFA (must be enabled):
aws iam get-account-summary --query 'SummaryMap.AccountMFAEnabled'
# 1 = enabled, 0 = NOT ENABLED (CRITICAL FINDING)

# Delete root access keys if any exist:
aws iam list-access-keys --user-name root
# Any output = CRITICAL — delete immediately
aws iam delete-access-key --user-name root --access-key-id AKIAXXXXXXXX

# Generate credential report for all IAM users:
aws iam generate-credential-report
aws iam get-credential-report --query 'Content' \
  --output text | base64 --decode | \
  awk -F',' 'NR>1 && $8=="false" {print "NO MFA:", $1}'

# Find overpermissive policies (wildcard actions or resources):
aws iam list-policies --scope Local --query 'Policies[*].Arn' --output text | \
  tr '\t' '\n' | while read ARN; do
    POLICY=$(aws iam get-policy-version \
      --policy-arn "$ARN" \
      --version-id $(aws iam get-policy --policy-arn "$ARN" \
        --query 'Policy.DefaultVersionId' --output text) \
      --query 'PolicyVersion.Document')
    echo "$POLICY" | grep -q '"\*"' && echo "OVERPRIVILEGED: $ARN"
  done

# Enable IAM Access Analyzer in all regions:
```

```
for region in us-east-1 us-west-2 eu-west-1 eu-central-1; do
  aws accessanalyzer create-analyzer \
    --analyzer-name hs-security-analyzer --type ACCOUNT \
    --region $region 2>/dev/null && echo "Enabled: $region"
done
```

CIS Control	AWS CLI Verification	Pass Criteria
Root MFA enabled	aws iam get-account-summary | grep AccountMFAEnabled	Value = 1
No root access keys	aws iam list-access-keys --user-name root	No keys returned
All IAM users have MFA	From credential report: mfa_active column	All active users = true
Password min 14 chars	aws iam get-account-password-policy | jq .PasswordPolicy.MinimumPasswordLength	Value >= 14
Access Analyzer enabled	aws accessanalyzer list-analyzers	Analyzer with status ACTIVE
CloudTrail all regions	aws cloudtrail describe-trails --include-shadow-trails false	IsMultiRegionTrail = true
S3 public access block	aws s3control get-public-access-block --account-id	All four = true

Azure Entra ID Hardening

```
# List all Global Administrators (should be 2-4 — no more):
az ad directory-role show --display-name 'Global Administrator' \
  --query 'id' -o tsv | \
  xargs az ad directory-role member list --id \
  --query '[].{Name:displayName,UPN:userPrincipalName}' -o table

# List all Conditional Access policies:
az rest --method GET \
  --url 'https://graph.microsoft.com/v1.0/identity/conditionalAccess/policies' \
  --query 'value[].{Name:displayName,State:state}' -o table

# Check for legacy authentication block policy (MUST exist):
# Look for policy blocking: Exchange ActiveSync + Other clients
# If missing: any attacker with credentials can bypass MFA via legacy protocols

# Privileged Identity Management (PIM) — no standing privileged access:
# All admin roles should be eligible (activated on demand) not permanent
# Check: Entra ID → Identity Governance → Privileged Identity Management
```

GCP IAM Hardening

```
# List all users with Owner/Editor/Viewer (avoid these broad roles):
gcloud projects get-iam-policy PROJECT_ID \
  --flatten='bindings[].members' \
  --format='table(bindings.role,bindings.members)' | \
  grep -E 'roles/owner|roles/editor'

# Check for service account keys (prefer Workload Identity instead):
gcloud iam service-accounts list --format='value(email)' | \
  while read SA; do
    KEYS=$(gcloud iam service-accounts keys list --iam-account="$SA" \
      --managed-by=user --format='value(name)' 2>/dev/null | wc -l)
    [ $KEYS -gt 0 ] && echo "SERVICE ACCOUNT KEYS: $SA ($KEYS keys)"
  done

# Enable Organization Policy constraints:
gcloud resource-manager org-policies set-policy \
  --organization=ORG_ID constraint-file.yaml
# Key constraints to enforce:
# constraints/compute.requireOsLogin
# constraints/compute.restrictCloudNATUsage
# constraints/iam.disableServiceAccountKeyCreation
```

SERVICE ACCOUNT KEYS ARE A CRITICAL RISK

Exported GCP service account keys are long-lived credentials frequently leaked in code repositories. Prefer Workload Identity Federation (short-lived tokens) or VM-attached service accounts. If you must use keys: rotate every 90 days, store in Secret Manager, and alert on any key usage from unexpected IPs.

CHAPTER 04

Storage Security

Public cloud storage is the #1 cause of cloud data breaches. Every bucket must be explicitly hardened.

S3 | AZURE BLOB | GCS | ENCRYPTION | IMMUTABILITY

Cloud Storage Hardening — All Platforms

```
# AWS S3 — Find all public buckets immediately:
aws s3api list-buckets --query 'Buckets[*].Name' --output text | \
  tr '\t' '\n' | while read bucket; do
    BLOCK=$(aws s3api get-bucket-public-access-block \
      --bucket "$bucket" 2>/dev/null | jq '.PublicAccessBlockConfiguration | \
      all(. == true)' 2>/dev/null || echo 'false')
    [ "$BLOCK" != 'true' ] && echo "⚠  PUBLIC: $bucket"
  done

# Enable account-level public access block (blocks ALL buckets):
aws s3control put-public-access-block \
  --account-id $(aws sts get-caller-identity --query Account --output text) \
  --public-access-block-configuration \

BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=tru
e

# Enable Object Lock (IMMUTABLE — ransomware cannot delete):
aws s3api put-object-lock-configuration \
  --bucket YOUR-BACKUP-BUCKET \
  --object-lock-configuration '{
    "ObjectLockEnabled": "Enabled",
    "Rule": {"DefaultRetention": {"Mode": "COMPLIANCE", "Days": 30}}
  }'
# COMPLIANCE mode: Cannot be deleted even by root account during retention

# Enable server-side encryption with KMS:
aws s3api put-bucket-encryption \
  --bucket YOUR-BUCKET \
  --server-side-encryption-configuration '{
```

```
    "Rules": [{"ApplyServerSideEncryptionByDefault": {
      "SSEAlgorithm": "aws:kms",
      "KMSMasterKeyID": "alias/your-key"
    }, "BucketKeyEnabled": true}]
  }'
```

Control	AWS S3	Azure Blob	GCP Cloud Storage	Risk if Missing
Block public access	Account-level public access block	Disable "Allow Blob public access"	Enable UBLA (uniform bucket-level access)	PUBLIC DATA EXPOSURE
Encryption at rest	SSE-KMS with CMK	Customer-managed key in Key Vault	Cloud KMS with customer-managed key	Regulatory non-compliance
Enforce HTTPS only	Bucket policy deny HTTP	Secure transfer required = Enabled	Enforce HTTPS in bucket policy	Data interceptable in transit
Versioning/Soft delete	Versioning + MFA delete	Blob soft delete 30+ days	Object versioning enabled	Ransomware/accidental deletion loss
Immutable backup	Object Lock COMPLIANCE mode	Immutable storage policy	Object retention policy	Backup deletion by ransomware
Access logging	Server access logging enabled	Storage analytics logging	Data Access audit logs	Cannot investigate unauthorized access

HS

HORIZONSHIELD

Securing Your Digital Future, Today.

Cloud Security Hardening Guide 2026

Part of the HorizonShield Security Series — 16 comprehensive professional cybersecurity manuals covering every domain of modern enterprise security.