07-AWS-IAM-Security-Review-Checklist

AWS IAM Security Review Checklist

For cloud security review engagements covering AWS Identity and Access Management configuration.

---

1. Root Account

- [ ] Root account access keys do not exist (deleted, not just deactivated)
- [ ] MFA enabled on root account (hardware key preferred)
- [ ] Root account not used for day-to-day operations (verify via CloudTrail)
- [ ] Root account email is a monitored group alias, not an individual's inbox
- [ ] Billing/account alerts configured for root account activity

2. IAM Users

- [ ] MFA enforced for all IAM users (especially console access)
- [ ] No shared/generic IAM user accounts
- [ ] Unused IAM users identified and deactivated/removed (check `GetCredentialReport`)
- [ ] Password policy enforced (length, complexity, rotation, reuse prevention)
- [ ] Access keys rotated regularly (≤ 90 days) and unused keys removed
- [ ] No access keys embedded in code, AMIs, or configuration files

3. IAM Policies — Least Privilege

- [ ] No policies with `"Action": "*"` and `"Resource": "*"` (wildcard admin) unless explicitly justified
- [ ] AWS managed policies reviewed for over-permissiveness vs. actual need (prefer custom least-privilege policies)
- [ ] Inline policies minimized in favor of managed/reusable policies
- [ ] Use of IAM Access Analyzer to identify unused permissions and external access
- [ ] Permissions boundaries applied where delegated administration exists
- [ ] Service Control Policies (SCPs) applied at the Organization/OU level for guardrails

4. IAM Roles

- [ ] Cross-account roles use external ID and condition restrictions where applicable
- [ ] EC2/Lambda/ECS instance roles scoped to only the permissions the workload needs
- [ ] No long-lived credentials used where a role/temporary credential would work (e.g., EC2 instance profiles instead of embedded keys)
- [ ] Trust policies reviewed for overly permissive `Principal` fields (avoid `"Principal": "*"` without conditions)
- [ ] Role session duration set appropriately (not unnecessarily long)

5. Groups & Delegation

- [ ] Permissions assigned via groups, not directly to individual users
- [ ] Admin group membership minimized and reviewed regularly
- [ ] Break-glass/emergency access procedure documented and tested

6. Federation & SSO

- [ ] AWS IAM Identity Center (SSO) or federated identity used instead of numerous individual IAM users, where feasible
- [ ] SAML/OIDC trust configuration reviewed for correctness and least privilege
- [ ] Session duration and re-authentication requirements appropriate for sensitive roles

7. Logging & Monitoring

- [ ] CloudTrail enabled in all regions, logs sent to a centralized, access-restricted S3 bucket
- [ ] CloudTrail log file integrity validation enabled
- [ ] IAM-related CloudTrail events monitored via CloudWatch/SIEM (e.g., `CreateUser`, `AttachUserPolicy`, `CreateAccessKey`, root logins)
- [ ] AWS Config rules enabled to detect IAM drift (e.g., `iam-user-mfa-enabled`, `iam-policy-no-statements-with-admin-access`)
- [ ] GuardDuty enabled to detect anomalous IAM/credential activity
- [ ] Alerts configured for root account usage, policy changes, and new access key creation

8. Resource-Based Policies

- [ ] S3 bucket policies reviewed for public access; S3 Block Public Access enabled account-wide unless explicitly required
- [ ] KMS key policies reviewed for overly broad grants
- [ ] Lambda resource policies reviewed for unintended public invoke permissions
- [ ] Secrets Manager / SSM Parameter Store access scoped per application, not broadly shared

9. Third-Party & External Access

- [ ] All third-party/vendor IAM roles reviewed for scope and external ID usage
- [ ] Cross-account trust relationships documented and periodically re-validated
- [ ] No wildcard trust policies allowing any AWS account to assume a role

10. Governance & Review Cadence

- [ ] IAM Access Analyzer findings reviewed and remediated regularly
- [ ] Periodic access recertification process in place (quarterly recommended)
- [ ] IAM changes go through code review / IaC pipeline (Terraform/CloudFormation) rather than manual console changes where possible
- [ ] Least-privilege policy generation tooling used (e.g., IAM Access Analyzer policy generation) when scoping new roles

---

Reference: AWS IAM Best Practices documentation, CIS AWS Foundations Benchmark, AWS Well-Architected Framework (Security Pillar).