JARWIS

PENETRATION TEST REPORT

TARGET APPLICATION

AWS Account: 123456789012 (us-east-1)

Cloud Infrastructure Security Assessment

REPORT ID

JAR-20260104_194827-692E19

ASSESSMENT DATE

January 04, 2026

ENDPOINTS TESTED

12

ASSESSMENT TYPE

Authenticated & Unauthenticated

Assessment Results

CRITICAL RISK

2 Critical	2 High	2 Medium	1 Low	0 Info

CONFIDENTIAL

Introduction

Jarwis is an AI-powered penetration testing platform that delivers comprehensive security assessments at scale. Leveraging advanced artificial intelligence and machine learning algorithms, Jarwis automates the discovery and verification of security vulnerabilities across web applications, mobile applications, APIs, and cloud infrastructure.

The platform combines the thoroughness of manual penetration testing with the speed and consistency of automated scanning, providing organizations with actionable security insights in hours rather than weeks. Jarwis systematically tests against the OWASP Top 10 and other industry-standard vulnerability classifications, delivering detailed findings with proof-of-concept evidence and remediation guidance.

> **Key Capabilities:** Automated vulnerability discovery • AI-powered verification to minimize false positives • OWASP Top 10 coverage • Detailed proof-of-concept evidence • Contextual remediation recommendations • Multi-format reporting (PDF, HTML, JSON, SARIF)

Executive Summary

Example Corporation engaged Jarwis to conduct a comprehensive penetration test of their cloud infrastructure security assessment environment to evaluate security vulnerabilities and assess potential risks to their digital infrastructure. During the time allotted for this engagement, Jarwis systematically tested the application's defense mechanisms, focusing on industry-standard attack vectors aligned with the OWASP Top Ten vulnerabilities.

Purpose

The penetration test was conducted to provide Example Corporation with a thorough security assessment of their cloud infrastructure security assessment infrastructure. Jarwis evaluated the application's resilience against common attack vectors including injection flaws, cross-site scripting, access control bypasses, authentication weaknesses, and information disclosure vulnerabilities. The assessment aimed to identify

exploitable security weaknesses that could compromise the confidentiality, integrity, and availability of Example Corporation's systems and data.

Findings Overview

During the assessment, Jarwis identified **7 distinct security vulnerabilities** across the target environment. The findings include **2 critical-severity** vulnerabilities, **2 high-severity** vulnerabilities, **2 medium-severity** vulnerabilities, and **1 low-severity** vulnerability.

The **critical-severity finding** involves s3 bucket with public access and sensitive data. An Amazon S3 bucket is configured with public read access and contains sensitive customer data including PII, financial records, and database backups. This data is accessible to anyone on the internet

The **critical-severity finding** involves iam user with excessive permissions and no mfa. An IAM user has AdministratorAccess policy attached, long-term access keys that have never been rotated, and no MFA configured. This account represents a high-value target for credential compromise.

The **high-severity finding** involves security group allows ssh from any ip. A security group attached to production EC2 instances allows SSH (port 22) access from any IP address (0.0.0.0/0). This exposes the instances to brute force attacks and exploitation of SSH vulnerabili

The **high-severity finding** involves rds database publicly accessible without encryption. A production RDS PostgreSQL database is configured as publicly accessible with encryption at rest disabled. The database is exposed to the internet and data is stored in plaintext on disk.

> The critical and high-severity vulnerabilities pose immediate risks to the organization's security posture, as they provide pathways for unauthorized data access and system compromise. Immediate remediation is strongly recommended.

Findings Summary

The following table provides an overview of all vulnerabilities identified during this assessment, organized by severity level.

SEVERITY	VULNERABILITY	CATEGORY	CWE
CRITICAL	S3 Bucket with Public Access and Sensitive Data	Broken Access Control	CWE-284
CRITICAL	IAM User with Excessive Permissions and No MFA	Auth Failures	CWE-287
HIGH	Security Group Allows SSH from Any IP	Security Misconfiguration	CWE-284
HIGH	RDS Database Publicly Accessible Without Encryptio	Cryptographic Failures	CWE-311
MEDIUM	CloudTrail Logging Disabled for S3 Data Events	Logging Failures	CWE-778
MEDIUM	Lambda Functions Using Outdated Runtime	Vulnerable Components	CWE-1104
LOW	Default VPC in Use with Default Security Group	Security Misconfiguration	CWE-1188

Conclusion

The penetration test revealed several significant security vulnerabilities that require immediate attention. The presence of 4 critical and high-severity vulnerabilities represents substantial risks that could lead to unauthorized system access and data compromise. By promptly addressing these identified vulnerabilities, the organization can significantly enhance their application security and reduce exposure to potential cyber threats.

Recommended Actions: Remediate 2 critical vulnerabilities within 24-48 hours • Address 2 high-severity findings within 7 days • Plan remediation of 2 medium-severity issues within 30 days • Implement security monitoring and logging for affected components • Conduct follow-up assessment to verify remediation effectiveness

Detailed Findings

The following section provides comprehensive technical details for each vulnerability identified during the assessment, including proof-of-concept evidence, AI-powered analysis, and specific remediation guidance.

JAR-CLD-001

CRITICAL

S3 Bucket with Public Access and Sensitive Data

A01 CWE-284

DESCRIPTION

An Amazon S3 bucket is configured with public read access and contains sensitive customer data including PII, financial records, and database backups. This data is accessible to anyone on the internet.

URL	METHOD	PARAMETER
s3://example-corp-prod-data/	AWS	ACL

EVIDENCE / PROOF OF CONCEPT

```
Bucket ACL: public-read. Found files: customers.csv (45MB), db_backup_2026.sql (2.3GB), financial_reports/*.xlsx
```

AI Analysis

Jarwis AI scanned AWS S3 buckets and found this production data bucket configured with public-read ACL. The bucket contains customer PII, database backups with credentials, and financial documents - all accessible without authentication.

HTTP DETAILS

REQUEST

```
AWS S3 Scan
Region: us-east-1
Target: example-corp-prod-* buckets
```

RESPONSE

```
s3://example-corp-prod-data/
├── customers.csv (45MB) - 2.3M customer records with PII
├── db_backup_2026.sql (2.3GB) - Full production database
├── financial_reports/ (340 files)
└── internal_docs/ (1,200 files)
```

✓ Remediation

Immediately enable S3 Block Public Access at the account level. Remove public ACLs from all buckets. Implement bucket policies with least-privilege access. Enable S3 access logging and CloudTrail for audit.

JAR-CLD-002

CRITICAL

IAM User with Excessive Permissions and No MFA

A07 CWE-287

DESCRIPTION

An IAM user has AdministratorAccess policy attached, long-term access keys that have never been rotated, and no MFA configured. This account represents a high-value target for credential compromise.

URL	METHOD	PARAMETER
arn:aws:iam::123456789012:user/deploy-service	AWS	IAM

EVIDENCE / PROOF OF CONCEPT

```
IAM User 'deploy-service': AdministratorAccess attached, access key age: 847 days, MFA: not enabled, last used: 2 hours ago
```

AI Analysis

Jarwis AI analyzed IAM configurations and found this user with full admin access, extremely old access keys, and no MFA. If these credentials are compromised (leaked in code, logs, or stolen), the entire AWS account is compromised.

HTTP DETAILS

REQUEST

```
IAM Security Audit
Account: 123456789012
Region: Global (IAM)
```

RESPONSE

```
User: deploy-service
Policies: AdministratorAccess (AWS managed)
Access Key 1: AKIA... (created: 847 days ago, last used: 2h ago)
Access Key 2: AKIA... (created: 523 days ago, never used)
MFA: Not configured
Password: Not set (programmatic access only)
```

✓ Remediation

Immediately enable MFA for all IAM users. Rotate access keys to 90-day maximum. Replace AdministratorAccess with least-privilege policies. Consider using IAM roles with temporary credentials instead of long-term access keys.

JAR-CLD-003

HIGH

Security Group Allows SSH from Any IP

A05 CWE-284

DESCRIPTION

A security group attached to production EC2 instances allows SSH (port 22) access from any IP address (0.0.0.0/0). This exposes the instances to brute force attacks and exploitation of SSH vulnerabilities.

URL	METHOD	PARAMETER
sg-0abc123def456	AWS	SecurityGroup

EVIDENCE / PROOF OF CONCEPT

```
Inbound Rule: TCP 22 (SSH) from 0.0.0.0/0. Attached to: 12 EC2 instances including prod-web-*, prod-api-*
```

AI Analysis

Jarwis AI scanned security group configurations and found this production security group allowing SSH from the internet. Combined with weak SSH credentials or key exposure, this enables direct server compromise.

HTTP DETAILS

REQUEST

```
Security Group Audit
Account: 123456789012
Region: us-east-1
```

RESPONSE

```
Security Group: sg-0abc123def456 (prod-web-sg)
Inbound Rules:
- TCP 22 (SSH): 0.0.0.0/0 ⚠️
- TCP 443 (HTTPS): 0.0.0.0/0
- TCP 80 (HTTP): 0.0.0.0/0
Attached to: 12 instances
```

✓ Remediation

Restrict SSH access to specific IP ranges (corporate VPN/bastion). Use AWS Systems Manager Session Manager for shell access without opening SSH. Implement AWS Network Firewall for additional protection.

JAR-CLD-004

HIGH

RDS Database Publicly Accessible Without Encryption

A02 CWE-311

DESCRIPTION

A production RDS PostgreSQL database is configured as publicly accessible with encryption at rest disabled. The database is exposed to the internet and data is stored in plaintext on disk.

URL	METHOD	PARAMETER
prod-database.abc123.us-east-1.rds.amazonaws.com	AWS	RDS

EVIDENCE / PROOF OF CONCEPT

```
RDS Instance: PubliclyAccessible=true, StorageEncrypted=false, Engine=PostgreSQL 13.4, MultiAZ=false
```

AI Analysis

Jarwis AI audited RDS configurations and found this production database exposed to the internet without encryption. Anyone who obtains credentials can access the database from anywhere, and compromised storage would expose plaintext data.

HTTP DETAILS

REQUEST

```
RDS Security Audit
Account: 123456789012
Region: us-east-1
```

RESPONSE

```
RDS Instance: prod-database
Engine: PostgreSQL 13.4
PubliclyAccessible: true ⚠️
StorageEncrypted: false ⚠️
MultiAZ: false
BackupRetention: 7 days
VPC: vpc-prod
Subnets: public-subnet-1a, public-subnet-1b
```

✓ Remediation

Disable public accessibility immediately. Enable encryption at rest (requires snapshot and restore for existing DBs). Enable encryption in transit. Move RDS to private subnets with VPC endpoints.

JAR-CLD-005

MEDIUM

CloudTrail Logging Disabled for S3 Data Events

A09 CWE-778

DESCRIPTION

CloudTrail is not configured to log S3 data events (GetObject, PutObject, DeleteObject). This means object-level access to sensitive data cannot be audited or investigated.

URL	METHOD	PARAMETER
arn:aws:cloudtrail:us-east-1:123456789012:trail/main-trail	AWS	CloudTrail

EVIDENCE / PROOF OF CONCEPT

```
Trail 'main-trail': ManagementEvents=All, DataEvents=None configured. S3 data events not logged.
```

AI Analysis

Jarwis AI analyzed CloudTrail configuration and found S3 data events are not being logged. If sensitive data in S3 is accessed or exfiltrated, there will be no audit trail to investigate the incident.

HTTP DETAILS

REQUEST

```
CloudTrail Configuration Audit
Account: 123456789012
Region: us-east-1
```

RESPONSE

```
Trail: main-trail
MultiRegion: true
ManagementEvents: ReadWriteType=All
DataEvents: None configured
InsightsSelectors: None
LogFileValidation: enabled
```

✓ Remediation

Enable S3 data event logging for all sensitive buckets. Consider enabling for all buckets with appropriate log filtering. Implement log analysis with CloudWatch Logs Insights or a SIEM solution.

JAR-CLD-006

MEDIUM

Lambda Functions Using Outdated Runtime

A06 CWE-1104

DESCRIPTION

Multiple Lambda functions are running on deprecated or EOL (End of Life) runtimes that no longer receive security updates. These functions are vulnerable to known runtime exploits.

URL	METHOD	PARAMETER
arn:aws:lambda:us-east-1:123456789012:function:*	AWS	Lambda

EVIDENCE / PROOF OF CONCEPT

```
Functions using deprecated runtimes: python3.6 (5 functions), nodejs12.x (3 functions), nodejs10.x (2 functions - EOL)
```

AI Analysis

Jarwis AI inventoried Lambda functions and found several using deprecated Python 3.6 and Node.js 10.x/12.x runtimes. These runtimes no longer receive security patches, exposing the functions to known vulnerabilities.

HTTP DETAILS

REQUEST

```
Lambda Runtime Audit
Account: 123456789012
Region: us-east-1
```

RESPONSE

```
Deprecated Runtimes Found:
- python3.6: data-processor, api-handler, report-gen, auth-service, email-sender
- nodejs12.x: image-resizer, pdf-generator, webhook-handler
- nodejs10.x: legacy-api, cron-job (EOL - no longer invocable after deprecation date)
```

✓ Remediation

Upgrade all Lambda functions to supported runtimes (Python 3.11+, Node.js 18.x+). Implement automated runtime upgrade testing in CI/CD. Enable AWS Lambda runtime deprecation

notifications in AWS Health.

JAR-CLD-007

LOW

Default VPC in Use with Default Security Group

A05 CWE-1188

DESCRIPTION

The default VPC is being used for production resources, and the default security group has been modified with overly permissive rules. This increases the attack surface and complicates network security management.

URL	METHOD	PARAMETER
vpc-abc123 (default)	AWS	VPC

EVIDENCE / PROOF OF CONCEPT

```
Default VPC in use in us-east-1. Default security group modified with inbound rules allo
wing 0.0.0.0/0 on multiple ports.
```

AI Analysis

Jarwis AI found the default VPC is being used for production workloads. Default VPCs have a predictable CIDR range and the default security group cannot be deleted, making security hardening more difficult.

HTTP DETAILS

REQUEST

```
VPC Configuration Audit
Account: 123456789012
Region: us-east-1
```

RESPONSE

```
Default VPC: vpc-abc123
CIDR: 172.31.0.0/16
Resources in Default VPC: 23 EC2 instances, 4 RDS instances, 2 ELBs
Default Security Group Rules Modified: Yes
- Inbound: TCP 22, 80, 443, 3306, 5432 from 0.0.0.0/0
```

✓ Remediation

Create custom VPCs with planned CIDR ranges for production workloads. Implement VPC Flow Logs for network monitoring. Use Network ACLs as an additional layer of defense. Delete unused default VPCs.

Methodology

Jarwis employs a systematic, multi-phase approach to penetration testing that combines automated scanning with AI-powered analysis to deliver comprehensive security assessments.

1 Reconnaissance

Automated discovery of endpoints, forms, APIs, and application structure using headless browser technology and intelligent crawling algorithms.

2 Pre-Auth Testing

Security testing of publicly accessible surfaces including login forms, registration flows, and unauthenticated API endpoints.

3 Authentication Analysis

Examination of authentication mechanisms, session management, token handling, and credential storage practices.

4 Post-Auth Testing

Authenticated testing including IDOR, CSRF, privilege escalation, and access control bypass attempts.

5 AI Verification

AI-powered analysis of all findings to eliminate false positives and provide contextual remediation recommendations.

6 Reporting

Generation of comprehensive reports with detailed findings, proof-of-concept evidence, and prioritized remediation guidance.

OWASP Top 10 Coverage

This assessment covers the OWASP Top 10 (2021) security risks, the industry-standard framework for web application security.

CODE	CATEGORY	STATUS
A01	Broken Access Control	✓ Tested

CODE	CATEGORY	STATUS
A02	Cryptographic Failures	✓ Tested
A03	Injection (SQL, XSS, Command)	✓ Tested
A04	Insecure Design	✓ Tested
A05	Security Misconfiguration	✓ Tested
A06	Vulnerable and Outdated Components	✓ Tested
A07	Identification and Authentication Failures	✓ Tested
A08	Software and Data Integrity Failures	✓ Tested
A09	Security Logging and Monitoring Failures	✓ Tested
A10	Server-Side Request Forgery (SSRF)	✓ Tested

Appendix

Discovered Endpoints

The following endpoints and components were identified during the reconnaissance phase of the assessment.

```
1. s3://example-corp-prod-data/
2. s3://example-corp-logs/
3. s3://example-corp-backups/
4. arn:aws:iam::123456789012:user/deploy-service
5. arn:aws:iam::123456789012:role/lambda-execution
6. sg-0abc123def456 (prod-web-sg)
7. sg-0def456abc789 (prod-db-sg)
8. prod-database.abc123.us-east-1.rds.amazonaws.com
9. arn:aws:lambda:us-east-1:123456789012:function:data-processor
10. arn:aws:lambda:us-east-1:123456789012:function:api-handler
11. vpc-abc123 (default VPC)
12. vpc-prod123 (production VPC)
```

Disclaimer

⚠️ Important Notice

This security assessment report is provided for informational purposes only. The findings contained herein represent the security posture of the target application at the time of testing. Jarwis AI Security Platform does not guarantee the completeness or accuracy of this assessment. Security is a continuous process, and new vulnerabilities may be

discovered after this report is generated. The recipient of this report is responsible for implementing the recommended remediation measures and verifying their effectiveness.

Terms of Use