Glue_privesc Interim Report

Mentor	Niko
Submisison Date	2024.08.13
Track	디지털포렌식 트랙
Name	이서준

Contents

1. Environment Setup

For this assignment, I used Ubuntu 24.04 LTS (GNU/Linux 6.8.0-40-generic x86_64). As we had already completed the initial setup of Terraform and CloudGoat during the lecture, this report will omit that process and begin with the scenario construction process.

2. Scenario Overview

2.1 Scenario Details

The scenario has the purpose of "Retrieve a secret string ("flag") stored in the SSM Parameter Store". And the scenario consists of the following AWS resources:

1. 1 x VPC (including: 1 S3 bucket, 1 RDS instance, 1 EC2 instance, Glue service)
2. 1 x Lambda function
3. SSM Parameter Store
4. 2 x IAM users

In this environment, it is assumed that a Glue service manager accidentally uploads their access keys through a web page. The manager, upon realizing this, deletes the keys from S3 but overlooks the fact that the keys were also stored in the database.

2.2 Attack Scenario

The attacker's goal is to acquire the manager's keys, access the SSM Parameter Store, and find the parameter value named "flag". The main attack stages are as follows:

1. Exploit SQL injection vulnerability to steal the manager's keys
2. Analyze the permissions of the acquired account and identify vulnerabilities
3. Insert reverse shell code into S3 via the web page
4. Create and execute a Glue job using AWS CLI
5. Access the SSM Parameter Store and extract the target data

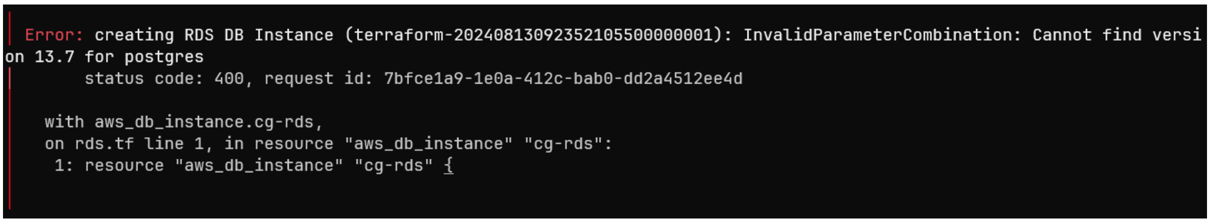
3. Scenario Installation Process

3.1 Initial Attempt and Error Occurrence

To build the scenario environment, we executed the following command:

```
./cloudgoat.py create glue_privesc
```

However, an error occurred as shown in Figure 1. Upon analyzing the error message, we identified that the scenario was designed to use PostgreSQL version 13.7, but this version could not be found.



```
 Error: creating RDS DB Instance (terraform-20240813092352105500000001): InvalidParameterCombination: Cannot find versi
on 13.7 for postgres
       status code: 400, request id: 7bfce1a9-1e0a-412c-bab0-dd2a4512ee4d

   with aws_db_instance.cg-rds,
   on rds.tf line 1, in resource "aws_db_instance" "cg-rds":
    1: resource "aws_db_instance" "cg-rds" {
```

Figure 1. Error of PostgreSQL version

3.2 Problem-Solving Process

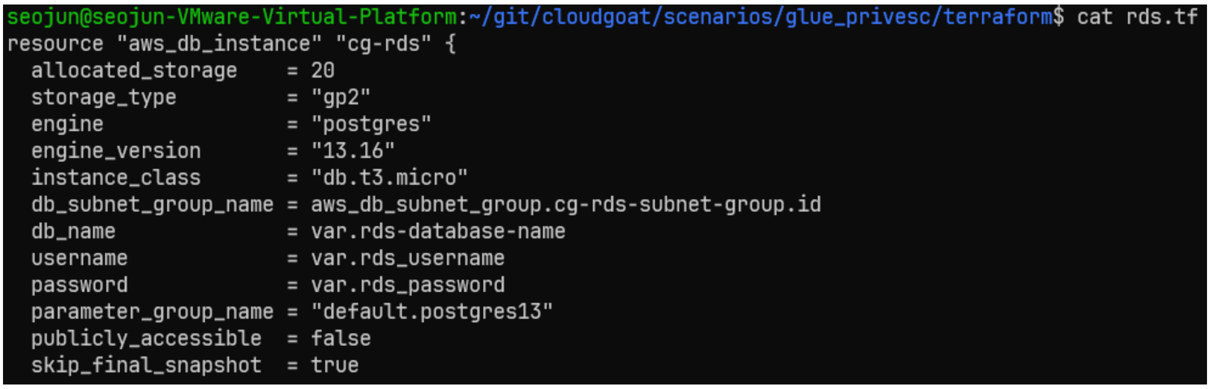
To resolve this issue, I took the following steps:

1. Opened the `terraform/rds.tf` file of the scenario to check the PostgreSQL version settings.

2. Modified the PostgreSQL version from 13.7 to 13.16.

3. Also adjusted the `parameter_group_name` to match the new version.

These modifications can be seen in Figure 2.



```
seojun@seojun-VMware-Virtual-Platform:~/git/cloudgoat/scenarios/glue_privesc/terraform$ cat rds.tf
resource "aws_db_instance" "cg-rds" {
  allocated_storage    = 20
  storage_type         = "gp2"
  engine               = "postgres"
  engine_version       = "13.16"
  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
```

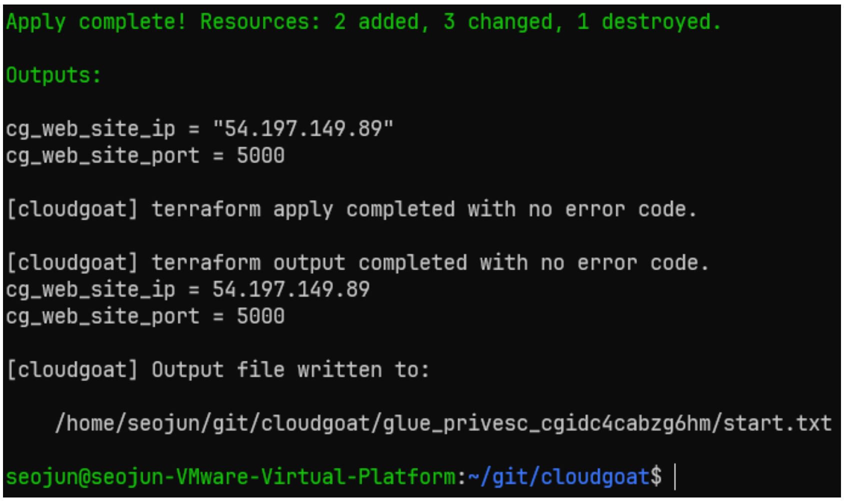
Figure 2. Modified the PostgreSQL vresion in rds.tf

3.3 Retry After Modification

After making these changes, I re-ran the same command:

./cloudgoat.py create glue_privesc

This time, as shown in Figure 3, the scenario build was successfully completed.

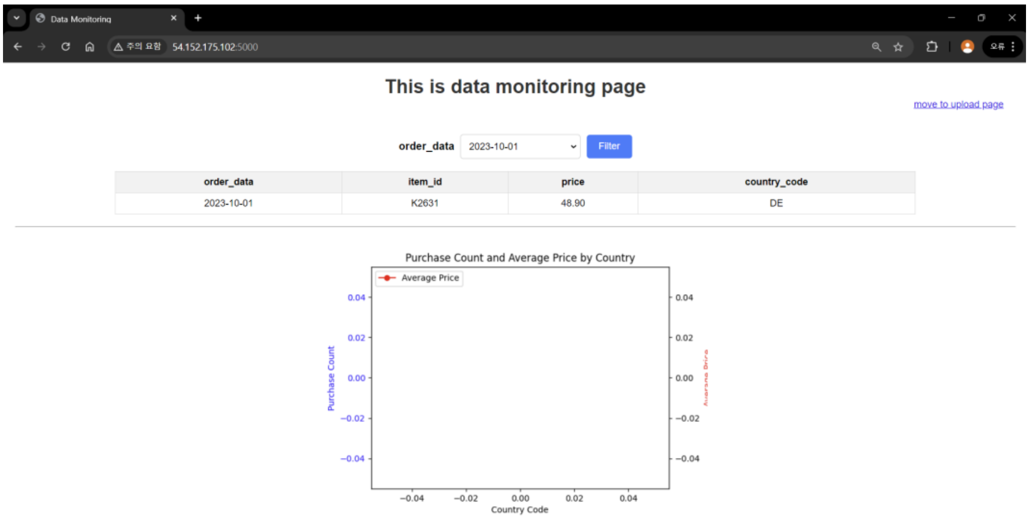


```
Apply complete! Resources: 2 added, 3 changed, 1 destroyed.

Outputs:

cg_web_site_ip = "54.197.149.89"
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 = 54.197.149.89
cg_web_site_port = 5000

[cloudgoat] Output file written to:

    /home/seojun/git/cloudgoat/glue_privesc_cgidc4cabzg6hm/start.txt

seojun@seojun-VMware-Virtual-Platform:~/git/cloudgoat$ 
```

Figure 3. Successfully completion of the scenario building

4. Web Interface Confirmation

After the scenario was built, I accessed the URL provided in the output. Figure 4 shows the web page loaded normally.



This is data monitoring page

move to upload page

order_data 2023-10-01 Filter

order_data	item_id	price	country_code
2023-10-01	K2631	48.90	DE

Figure 4. The web page view of the scenario

5. Interim Evaluation and Future Plans

5.1 Current Progress

I have successfully completed the environment configuration to run the "glue_privesc" scenario. Although I encountered unexpected issues during this process, I was able to resolve them by referring to AWS official documentation and Terraform configurations.

5.2 Learning Experience

Through this environment configuration process, I learned the following:

1. Version compatibility issues that can occur when configuring cloud environments
2. Structure and modification methods of Terraform files
3. Understanding of AWS RDS settings

5.3 Future Plans

1. Scenario Resolution: My next step will be to proceed with the actual problem-solving process of the "Glue_privesc" scenario.

2. Script Writing: For the final report, I try to write a script that can automate the solution to this problem.

3. In-depth Analysis: I will perform a detailed analysis of the vulnerabilities discovered and the attack paths.

This interim report has documented in detail the problems encountered during the environment configuration process and how I solved them. I expect this experience to be greatly helpful in solving problems that may occur in real cloud environments. In the next final report, I will present the scenario resolution process along with the results of writing an automation script.