

Glue_privesc(중간)

시나리오: Glue_privesc

Size: Large

Difficulty: Moderate

Command: `$./cloudgoat.py create glue_privesc`

Scenario Resources

- 1 VPC with:
 - S3 × 1
 - RDS x1
 - EC2 ×1
 - Glue service
- Lambda x1
- SSM parameter Store
- IAM Users x 2

Scenario Start(s)

Web address

Scenario Goal(s)

Find a secret string stored in the ssm parameter store

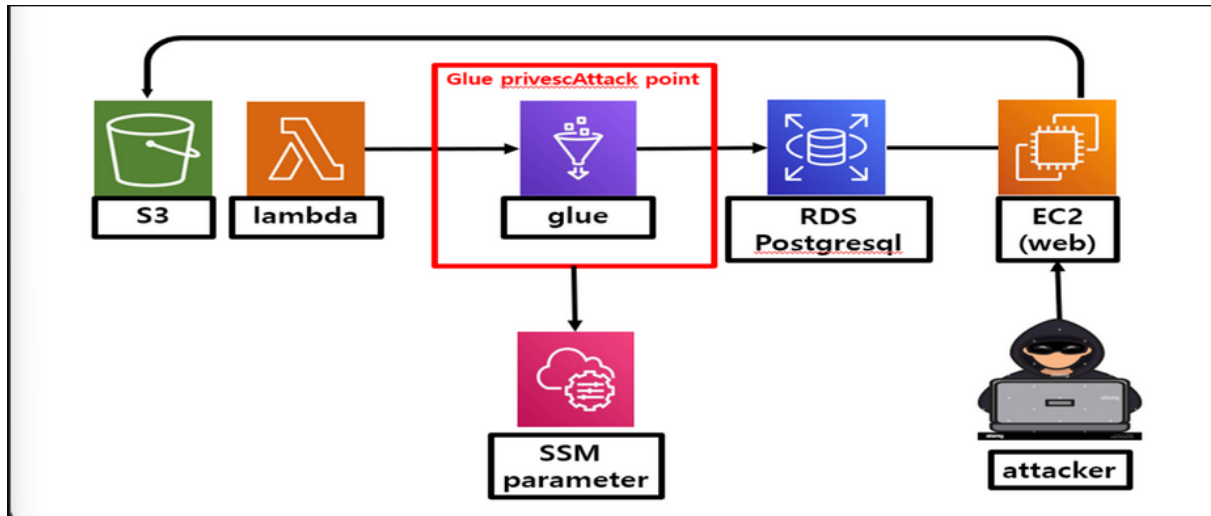
Summary

There is an environment that is implemented as shown in the schematic drawing below. Glue service manager will accidentally upload their access keys through the web page. The manager hurriedly deleted the key from s3, but does not recognize that the key was stored in the DB.

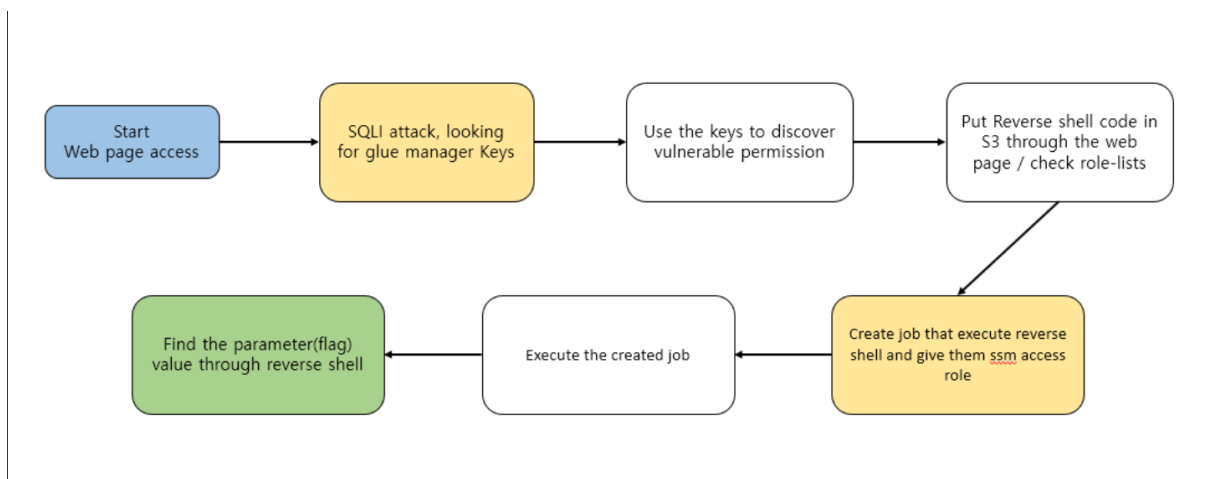
Find the manager's key and access the ssm parameter store with a vulnerable permission to find the parameter value named "flag".

Note: The web page and the glue ETL job used in this scenario require some latency. The web page requires 1 minute after applying, and Glue requires 3 minutes after uploading the file. If the data file is not applied properly, please wait a little longer!

Schematic drawing



Exploitation Route(s)



Route Walkthrough

※ The attacker identifies the web page functionality first. When you upload a file, it is stored in a specific s3, and you can see that the data in that file is applied to the monitoring page.

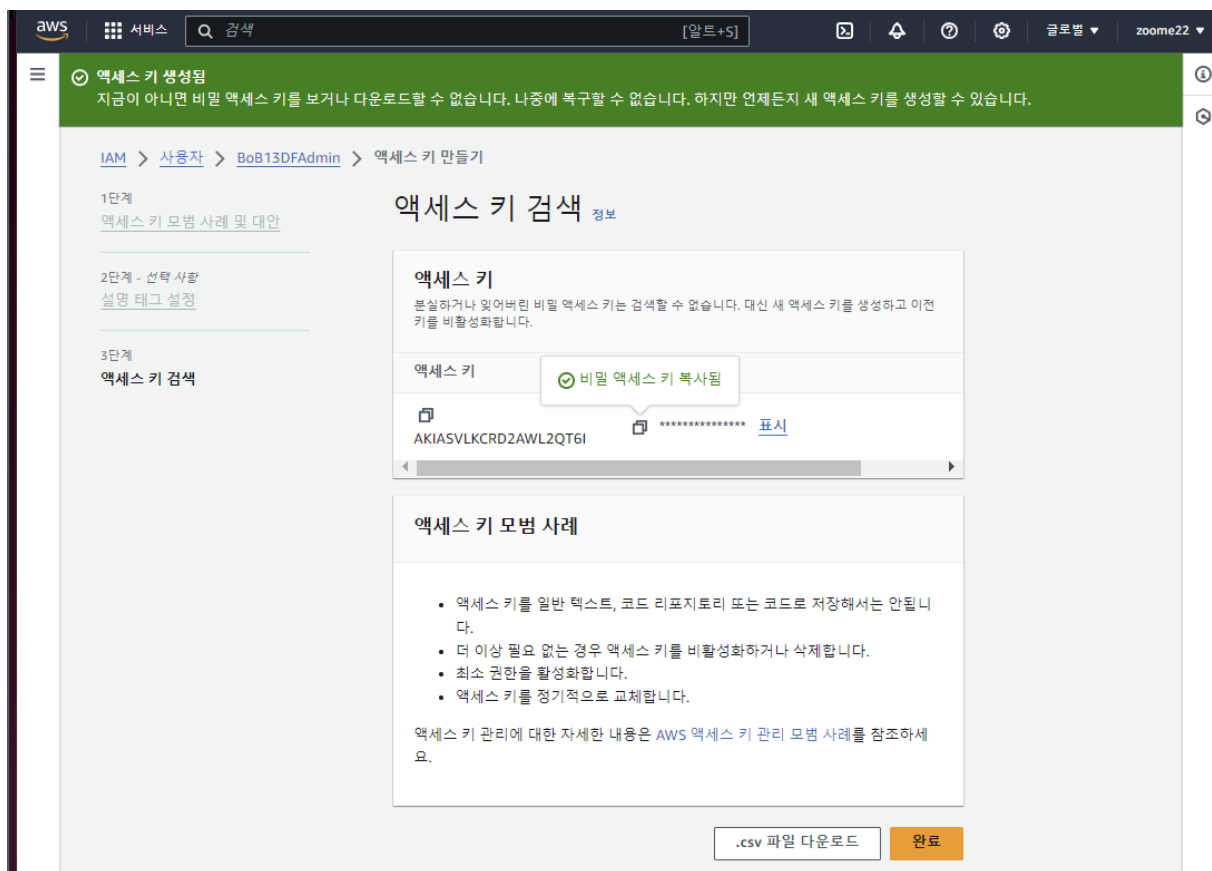
1. The attacker steals the Glue manager's access key and secret key through a SQL Injection attack on the web page.
2. The attacker checks the policies and permissions of the exposed account to identify any vulnerable privileges. Through these privileges, the attacker discovers the ability to create and execute a job that can perform a reverse shell attack, enabling them to obtain the desired role simultaneously.

3. List the roles to use "iam:passrole," write the reverse shell code, and insert this code file (.py) into S3 through the web page.
4. In order to gain SSM access, Perform the creation of a Glue service job via AWS CLI, which also executes the reverse shell code.
5. Execute the created job.
6. Extract the value of "flag"(parameter name) from the ssm parameter store.

A cheat sheet for this route is available [here](#)

https://github.com/RhinoSecurityLabs/cloudgoat/tree/master/scenarios/glue_privesc

환경 설정



BoB13DFAdmin이라는 IAM user를 추가하고 Ubuntu 24.04 환경에서 CLI로 aws를 연결하였다.

```
user@user-VMware-Virtual-Platform: ~/Desktop/cloudgoat
is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed argcomplete-3.2.3 boto3-1.34.159 botocore-1.34.159 click
default-group-1.2.4 jmespath-1.0.1 pluggy-1.5.0 s3transfer-0.10.2 sqlite-fts4-1
0.3 sqlite-utils-3.37 tabulate-0.9.0
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ chmod +x cloudgoat.py
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ ./cloudgoat.py config pr
file
No configuration file was found at /home/user/Desktop/cloudgoat/config.yml
Would you like to create this file with a default profile name now? [y/n]: y
Enter the name of your default AWS profile: BoB13DFAdmin
A default profile name of "BoB13DFAdmin" has been saved.
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ ./cloudgoat.py config wh
itelist --auto
No whitelist.txt file was found at /home/user/Desktop/cloudgoat/whitelist.txt

CloudGoat can automatically make a network request, using https://ifconfig.co to
find your IP address, and then overwrite the contents of the whitelist file wi
th the result.
Would you like to continue? [y/n]: y

whitelist.txt created with IP address 218.146.20.61/32
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$
```

cloud_breach_s3으로 리소스를 생성하여 설정을 완료하였다.

화이트리스트된 ip : 218.146.20.61/32

```
./cloudgoat.py destroy glue_privesc
```

요금이 나오지 않도록 사용 후에 리소스를 제거해주어야 한다.

시나리오 실습

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

시나리오를 시작하기 위해 다음 명령어를 입력한다.

```

Error: creating RDS DB Instance (terraform-2024081313204886180000000f): InvalidParameterCombination: Cannot find version 13.7 for postgres
    status code: 400, request id: fbe8b066-0587-45ef-b522-5e63721cb281

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" {

[cloudgoat] Error while running `terraform apply`.
  exit code: 1
  stdout: None
  stderr: None

```

glue_privesc 설치 중 RDS 버전을 읽지 못하는 오류가 발생한다.

```

user@user-VMware-Virtual-Platform: ~/Desktop/cloudgoat/glue_privesc_...
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

  port = "5432"

  storage_encrypted = true

  vpc_security_group_ids = [
    aws_security_group.cg-rds-security-group.id,
  ]

  depends_on = [local_file.sql_file]
-- INSERT --
5,32 Top

```

cg-rds 파일을 수정하여 버전 정보를 최신으로 업데이트 해준다.

- engine_version = 13.16

```
Apply complete! Resources: 8 added, 1 changed, 1 destroyed.

Outputs:

cg_web_site_ip = "54.209.212.221"
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 = 54.209.212.221
cg_web_site_port = 5000

[cloudgoat] Output file written to:

/home/user/Desktop/cloudgoat/glue_privesc_cgida033ch2f1a/start.txt
```

이후 다시 create 명령어를 입력하면 정상적으로 수행된다.