UNSECURED CLOUD OBJECT STORAGE: A PRIMER ON PREVALENCE & MITIGATION TECHNIQUES

BY

MIGUEL RAMIREZ AGUADO

INFORMATION TECHNOLOGY AND MANAGEMENT

Submitted in partial fulfillment of the
requirements for the degree of
Master in Information Technology and Management
in the Graduate College of the
Illinois Institute of Technology
& Master in Telecommunications Engineering
in the Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona of the
Universitat Politècnica de Catalunya

Supervised by Prof. Shawn Davis
Adviser

Chicago, Illinois
11 2018

Acknowledgement

I would like to express my most sincere gratitude to all my family that has supported me through my journey through university that is about to end.

Also, I would like to thank my supervisor Shawn Davis for his support and his help through the development of the project.

Finally, I would like to dedicate this project to my Mom, without her unconditional support and encouragement I would have never been able to finish this journey.

Table of contents

List of tables

List of figures

List of symbols

Symbol	Definition
ACL	Access Control Lists
API	Application Programming Interface
AWS	Amazon Web Services
CLI	Command-Line Interface
GB	GigaByte
SAS	Shared Access Signature
URL	Uniform Resource Locator

Abstract

Cloud storage services are gaining popularity due to their flexibility, elasticity, and support to a growing need of data storage at a very accessible price, around $0.02 per GigaByte (GB) each month. Thus, more and more companies and users are relying on cloud providers to store their files. This project will study the four main cloud storage providers: Amazon Web Services (AWS), Google Cloud, IBM Cloud and Microsoft's Azure. After understanding how each cloud provider works, a scanning tool was developed to search all four providers for public objects. To make sure companies and administrators understand the utility of the tool, privacy settings and permissions are commented, also commenting how and why previous data leakage occurred. The tool has been tested and the results obtained show that there is a lot of sensitive data available to the public. With the help of the developed tool, penetration testers can help companies by making sure that no sensitive data is publicly available.

Chapter 1

Introduction

Advances in networking technology and an increase in the need for computing resources have prompted many organizations to outsource their storage and computing needs. This new economic and computing model is referred to as cloud computing (Kamara & Lauter, 2010).

Public cloud infrastructures like AWS, Google Cloud, IBM Cloud and Microsoft's Azure, help their customers by moving their data to the cloud and saving them the cost of building and maintaining private storage infrastructure (see figure 1). Additionally, cloud storage provides availability and reliability at a really low cost. However, there is concern over the confidentiality and integrity of the data stored publicly, introducing a significant security and privacy risk.

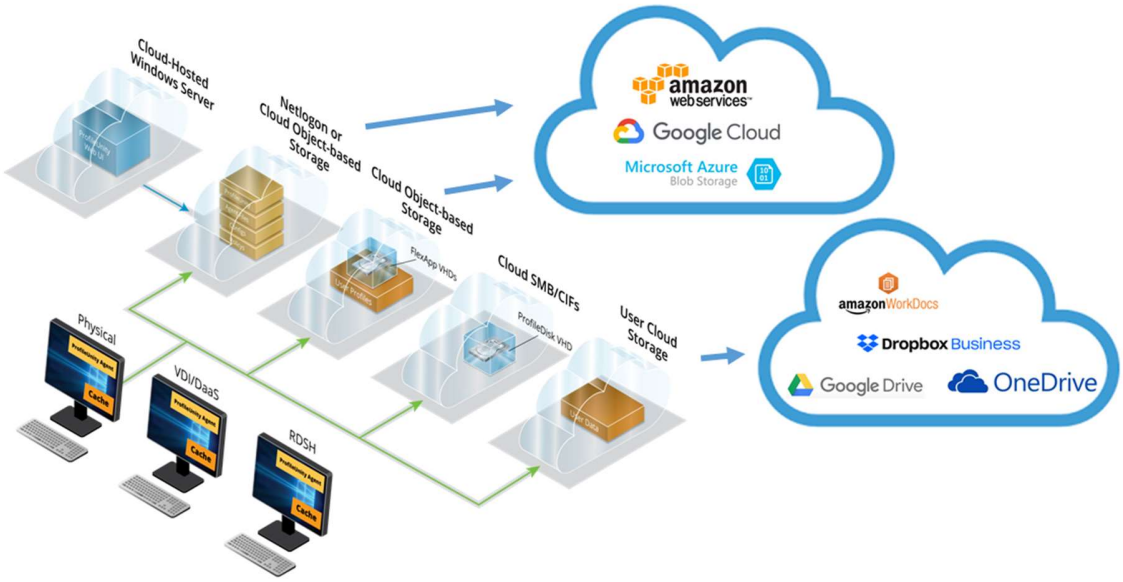

Figure 1 Cloud storage diagram (User Profile & Data Cloud Storage, 2018)

1.1 What is Cloud Storage?

Cloud storage is a cloud computing model that stores data on the Internet through a cloud computing provider who manages and operates data storage as a service (What is cloud storage | AWS, 2018).

Cloud-storage providers offer users clean and simple file-system interfaces, abstracting away the complexities of direct hardware management. At the same time though, such services eliminate the direct oversight of component reliability and security that enterprises and users expect (Bowers, Juels & Oprea, 2009).

Cloud storage is based on a virtualized infrastructure with accessible interfaces. Cloud-based data is stored in logical pools across disparate, commodity servers located on premises or in a data center managed by a third-party cloud provider.

Using the object storage protocol "RESTful API", a file and its metadata are stored as a single object and assigned an ID. In order to retrieve a file, the user needs to present the ID and the system will assemble the metadata and present it to the user for further use (Rouse, 2016).

Data is now stored in different ways depending on the amount and the use it will be given to. The three main storage architectures are (Bradford, 2018):

- File-based storage. Is the classic approach, files are given a name, tagged them with metadata and are then organized in folders under directories and sub-directories. File level storage is simple and easy to organize but finding such files becomes harder the more files are accumulated.
- Block storage. A block is a raw storage volume filled with files that have been split into chunks of data of equal size. An operating system manages

the data and where to store it. Unlike file-based architectures, there are no additional details associated with a block outside of its address. This granular controls makes it higher performance than file-based storage.

- Object storage. Data is stored in isolated containers known as objects. Objects are given a unique identifier and stored in a flat memory model. Objects can be retrieved presenting its unique ID and they can be stored on a local server or a remote server in the cloud. This architecture gives more flexibility and scalability as it allows the users to manipulate metadata to the users' needs.

1.2 Main Public Cloud Storage Providers

1.2.1 Amazon Web Services

AWS is a subsidiary of Amazon.com that provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis. Subscribers are able to connect to their AWS systems at any time through the Internet.

AWS uses S3 bucket to provide object storage through web services interfaces. Objects are stored into Buckets owned by an AWS account and identified within each bucket by a unique user-assigned key (Muñoz, 2016). Bucket names must be unique across all the platform and cannot contain uppercase characters or underscores.

1.2.2 Google Cloud

Offered by Google, Google Cloud provides cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube.

Google Cloud stores the user's data in a Bucket. A Bucket has three properties that the user specifies when creating it: a globally unique name, a location where the bucket and its contents are stored, and a default storage class for objects added to the bucket (How Google uses Node.js? | Hype.Codes, 2017).

1.2.3 IBM Cloud

IBM Cloud is a suite of cloud computing services from IBM. With IBM Cloud, organizations can deploy and access virtualized IT resources over the Internet.

IBM Cloud Object Storage gives the object an identifying key and is stored in a Bucket with a unique name. This allows for highly scalable storage where the only information needed to retrieve the data is the name of the object and the Bucket where it is stored (Object Storage in the IBM Cloud, 2018).

1.2.4 Microsoft's Azure

Azure is a cloud storage system that provides customers with the ability to store seemingly limitless amounts of data for any duration of time. Customers have access to their data from anywhere at any time and only pay for what they use and store (Calder, Wang, Ogus, Nilakantan, Skjolsvold, Mckelvie & Haridas, 2011).

Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage exposes three resources: the user's storage account, the container in the account and the Blobs in the container.

Blob storage is optimized for storing massive amounts of unstructured data, such as text or binary data (Introduction to Azure Storage – Cloud storage on Azure, 2018).

1.3 Cloud Storage Data Leakage

The increasing use of cloud storage and the amount of data that is being migrated to the cloud, makes cloud services a juicy target for attackers that companies and users have to be aware of.

Attackers have already set their eyes on cloud storage and have been able to get access to the content stored in them and obtain sensitive data containing information of millions of customers.

Big companies that work with a lot of customers and have access to a lot of private information are the targets of attackers. The amount of data that these companies contain and have to store makes them more prone to commit errors and leave information publicly available.

1.3.1 Accenture Cloud Leak

Accenture one of the largest consulting firms left at least four cloud storage buckets unsecured and publicly downloadable, exposing Application Programming Interface (API) data, authentication credentials, certificates, decryption keys, customer information and more sensitive data (O'Sullivan, 2018).

The four AWS S3 buckets were publicly accessible and their content could be downloaded by anyone just by entering the Uniform Resource Locator (URL) with the bucket name into a web browser. The buckets were controlled by a single account responsible for making the content public.

1.3.2 Verizon Wireless Cloud Leak

On July 2017 personal information of about 6 million customers stored on AWS was leaked. In September of the same year corporate information about IT systems and login credentials was also leaked.

The first leak was caused by an employee that placed log information from customers on a publicly accessible S3 server. The second leak a few months later, was caused by an engineer that set up an insecure S3 account that contained technical information about the company (Chickowski, 2018).

In both cases, the leakage was caused by human error, such as incautious users that did not properly understand how to work with permissions and set up the proper privacy settings for the data that was being stored.

1.3.3 Booz Allen Hamilton Cloud Leak

Booz Allen Hamilton, a government contractor was found to have public information stored on the cloud on May 2017. Information that would require a Top Secret-level security clearance from the Department of Defense was accessible to anyone looking in the right place (O'Neill, 2018).

The AWS bucket stored about 28 GB of data including credentials for a senior engineer, passwords to a US government system, and a half dozen unencrypted passwords for government contractors.

In this case not only was information compromised, but further systems could have been as well. The exposed credentials could have been used to gain access to a Pentagon system and other government systems.

1.3.4 WWE Cloud Leak

Back in July 2017, security researchers found a publicly stored database belonging to the WWE. The data was stored again in a AWS S3 bucket without user-name or password protection.

The unsecured data contained customers information including names, physical addresses and personal information. Furthermore, another bucket contained billing address information and usernames for European customers (Sheridan, 2017).

1.4 Understanding Permissions

Most of the cloud data leaks that have occurred are caused by human error, and it is crucial for users that privacy permissions for each provider are understood and studied.

Every Cloud Service provider has different permissions models regarding how content is stored on their systems. In order for organizations to prevent data leakage, first it is necessary to understand how each cloud container works and deals with basic permissions settings.

The following questions have to be answered to understand permissions and how each provider deals with the stored objects:

- Is the bucket and/or objects public or private by default?
- Is anonymous access granted via Command-Line Interface (CLI) and web browser without having to be signed into the provider?
- Is there a setting to allow a logged in user to access the bucket but a logged out anonymous user would be disallowed?

- Is there a setting to allow a specific logged in user access to a private bucket?

If the buckets are public by default, an administrator could forget to change the setting to private when uploading sensitive data. Private buckets by default are more secure but still run the risk of an administrator changing the settings to public.

Anonymous access without being signed into the provider would allow access to every person on the internet that knows the name of the bucket. If this feature is available by default cloud storage would be unreliable as all the data stored in them would be publicly available.

Allowing registered users access to private content, is a key feature of cloud storage as it allows users to share large content between known and authenticated users through the internet without having to setup a physical storage device to a large number of users.

Through Access Control List (ACL) developers can grant registered users read or write permissions for individual buckets or objects, allowing to share files between the users. Another sharing option only for Azure, is through Shared Access Signatures (SAS) which provides a way to grant limited access to objects stored with other Azure clients without exposing the owner account key.

The following table provides information as to how the four main object storage providers operate (see table 1).

	Public or Private by default?	*Anonymous access granted?*	*Logged in user access to private bucket?*
AWS S3 Bucket	Private	• When Public • With pre-signed URL	Through ACL permissions granted
Google Cloud Bucket	Private	• When Public • With pre-signed URL	Through ACL permissions granted
IBM Cloud Bucket	Private	• When Public • With pre-signed URL	Through ACL permissions granted
Azure Blobs	Private	When Public	With Shared Access Signature (SAS) grant access

Table 1 Permissions for each provider (Managing Access Permissions to Your Amazon S3 Resources - Amazon Simple Storage Service, 2018) (Access Control Options | Cloud Storage | Google Cloud, 2018) (Bucket permissions, 2017) (Manage anonymous read access to container and blobs, 2017)

Even though the four main bucket providers set their object storage to private by default, administrators may still improperly change a bucket to public that contains sensitive data. A popular bucket scanning tool has been modified in order to allow companies and penetration testers the ability to search for these insecure public buckets.

Chapter 2

Bucket Scanner

The Bucket Scanner developed for this project is based on the Bucket Finder 1.1 developed by Robin Wood (Wood, 2018). Wood's Bucket Finder 1.1 has a creative commons attribution-share alike license which encourages modification.

Bucket Scanner has adapted the code provided by Robin Wood by adding new functionality. Initially, Bucket Finder 1.1 could scan AWS S3 buckets and provide information about them. The new tool has added the options to additionally scan Google Cloud and IBM Cloud.

The tool takes bucket names from a wordlist that the user must provide. The program then checks every name in the wordlist to see if the bucket exists in AWS S3, Google Cloud and IBM Cloud. Companies and authorized penetration testers can use custom wordlists made of the company's names or product names to run through the cloud providers and make sure the objects stored are set to private or search for any buckets employees may have created without authorization.

If there exists a bucket with a name found on the list, the tool will check if such bucket is public or private. Furthermore, the tool will list all files that are stored in public buckets and check if the files are public or private.

Finally, if the download option is enabled, public files found inside a public bucket will be downloaded into a new directory keeping the name of the bucket and files[1] (see figure 2).

[1] The download option should not be used to download files in a bucket without permission form the owner or company of the bucket

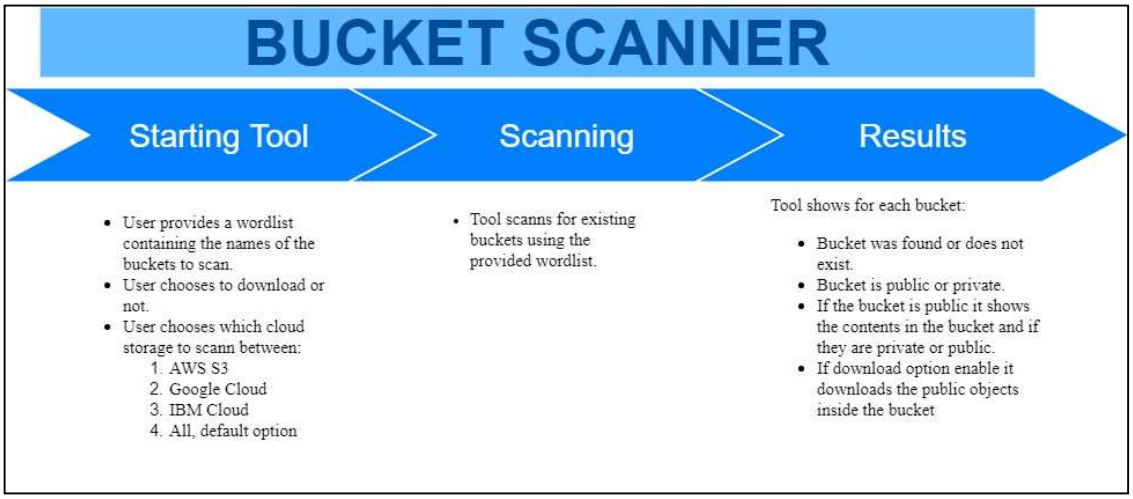

Figure 2 Execution of Bucket Scanner

2.1 How Does the Bucket Scanner Work?

All of the cloud providers specify that the name identifier of each bucket has to be unique across their cloud service. Therefore, there will be no bucket with the same name as another bucket in the same cloud platform. Even that the owner or the region are different, no bucket name will be used more than once.

The unique identifier feature allows performing a dictionary scan through all the cloud infrastructure of the provider. The Bucket Scanner that has been developed performs a dictionary scan on each cloud provider to search for confidential data.

To start the scanning tool, it is necessary that the user provides a dictionary containing a wordlist with names. These names will be used as the name of the buckets the tool will search for.

Additionally, the user must choose between which of the three storage providers will be scanned; if no Cloud Storage is specified, the scanner will scan all three providers

by default. Moreover, the user can choose whether to download the public content that the scanner finds or just list the content inside the bucket.

Once, the user has input the command with the necessary information for the system to work, the tool will start to scan for buckets. In the odd case that the user inputs wrong information or any of the necessary commands are not in place, the tool will notify the user what is wrong and display what information is required.

To see if the bucket exists, the scanner combines the name given on the wordlist and combines it with the host URL of the service provider:

http://<end-point-url>/<bucket-name>

The endpoints being:

- http://s3.amazonaws.com for AWS S3
- http://storage.googleapis.com for Google Cloud
- http://s3-api.us-geo.objectstorage.softlayer.net for IBM Cloud

In the case that the bucket is private or it does not exist, a browser would show an Error Code showing the problem (see figure 3 and 4):

Figure 3 Error Code for Private Bucket

Figure 4 Error Code for non-existent Bucket

If the tool encounters private or inexistent buckets, error codes are displayed to the user regarding if the bucket is private or if the bucket does not exist.

On the contrary if the bucket is public, there will not be any error code from a web browser and it will additionally show the file names stored inside (see figure5). The tool will display this same information to the user displaying the public bucket and the respective file names.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>448test</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <Delimiter/>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>test.txt</Key>
    <LastModified>2018-08-25T16:15:38.338Z</LastModified>
    <ETag>"098f6bcd4621d373cade4e832627b4f6"</ETag>
    <Size>4</Size>
    <Owner>
      <ID>a1705ff8-d58c-4f3b-93c4-e95bca3db6fa</ID>
      <DisplayName>a1705ff8-d58c-4f3b-93c4-e95bca3db6fa</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>test3.docx</Key>
    <LastModified>2018-11-02T18:22:55.801Z</LastModified>
    <ETag>"50e6d94813487d4b669281db3d1b7ed9"</ETag>
    <Size>11930</Size>
    <Owner>
      <ID>a1705ff8-d58c-4f3b-93c4-e95bca3db6fa</ID>
      <DisplayName>a1705ff8-d58c-4f3b-93c4-e95bca3db6fa</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```


Figure 5 Browser output when Public Bucket

The tool uses the HTTP HEAD method to determine if bucket objects are public or private.

HEAD requests allows meta-information in the response headers to be retrieved without having to download the actual content; this allows a faster way to check if a file is publicly available without actually downloading, or viewing the content inside the files.

Using the HEAD request on each file in the directory list, the scanner will get a response either: “200 OK” or “403 Forbidden” (see figure 6). With these two responses, the program can enumerate if the files stored are public or private.

Figure 6 Error 403 trying to access Private Object

Finally, if the user has enabled the option to download the content inside the bucket, the scanner will download all the public objects found and placed them in a folder with the name of the bucket followed by the name of the cloud provider.

To summarize, a penetration tester hired by company X could use a wordlist that contains the bucket names that the company may use (company name, company product names, etc.) to see if the buckets are public or private. The tool will then display if the buckets are private or public and the content stored inside the public buckets. If the penetration tester is authorized by the company; they could additionally enable the download option and study the files that the tool retrieves from the cloud. Finally, with the downloaded content, the tester will determine if it should be stored publicly or need to be changed to private. See figure 7 for a flow diagram of the tool's operations.

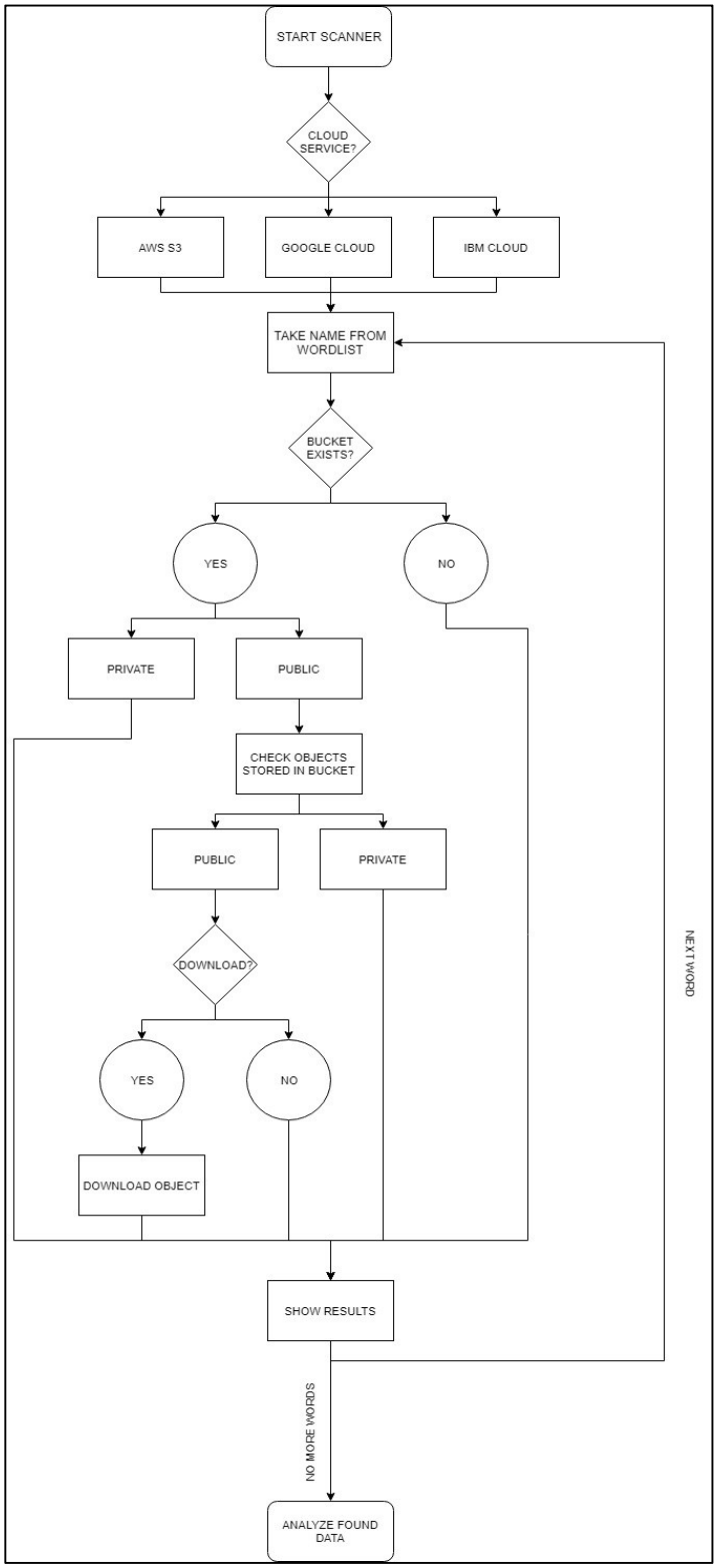

Figure 7 Flow diagram for Bucket Scanner

The following figure is an example of the output displayed to the user after running the tool through a short wordlist in which buckets have different privacy settings and the download option has been enabled (see figure8).

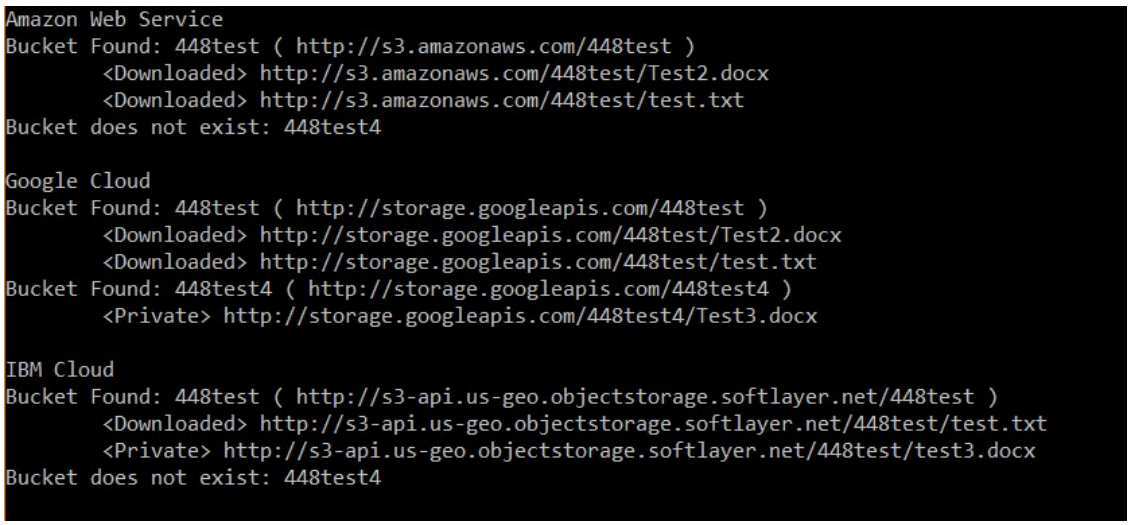

Figure 8 Display example from Bucket Scanner

2.2 Results

One goal of this project was to perform scans against two wordlists which contained typical names for buckets. The results were then analyzed and broken down to see what types of buckets and files organizations generally explicitly set to be publicly available (since all three providers configure buckets to be private by default).The download option was disabled in both scans as we only wanted to collect the metadata of files that organizations chose to make public and not view or collect the content of the files themselves.

The findings have been analyzed based on three steps: 1) the number of public, private, and not found buckets; 2) the public files and private files stored inside public buckets; and 3) a breakdown by the type of files that are public.

The first list that the tool has run through contains 1281 words with common bucket names that contain company names, and other random words. For the first step, the obtained results are listed below (see table 2):

	Total	*Percentage*
Scanned Buckets	3458	100%
Public Buckets	160	5%
Private Buckets	1573	45%
Not Found Buckets	1725	50%

Table 2 First scan Bucket result

Before analyzing the buckets found, comment that not all the names on the list are valid bucket names therefore not obtaining 3843 buckets scanned[2],[3].

As it can be seen from the table above, most of the buckets that the tool has found are set to private so the tool cannot access the metadata inside. However, 5% of the buckets are set to public meaning there are files stored inside which may contain sensitive information.

Following the next step shows the amount of public and private files that have been found inside the previously public buckets (see table 3):

	Total	*Percentage*
Private Files	7909	20%
Public Files	30921	80%
Total Files	38830	100%

Table 3 First scan files result

[2] https://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html
[3] https://cloud.google.com/storage/docs/naming

Table 3 shows that only one-fifth of the bucket's files are set to private and that the majority are set to be publicly accessible.

Finally, the breakdown of the type of public files found is shown below (see table 4):

Type	*Extensions*	*Total*
Images	jpg|png|gif|svg|bmp	18931
Music and Video	mp4|mp3|flv|wmv|swf	3309
Documents	txt|docx|pdf|pptx|xlsx|csv	1700
Web	html|xml|js|css|json	1756
Archives	zip|rar|tar|gz|7z	1392
Database	sql|dbf|dd|DDD	53
Runnables	jar|exe	29
Backups	bak	0
Other		3751

Table 4 First scan type of public files result

Most of the content that is stored inside the public buckets are multimedia files that may be innocuous but potentially could be displaying sensitive images of customers or even pictures of invoices. The tool has found a large number of publicly available written documents, databases and archives which could potentially contain sensitive data (see figure 9).

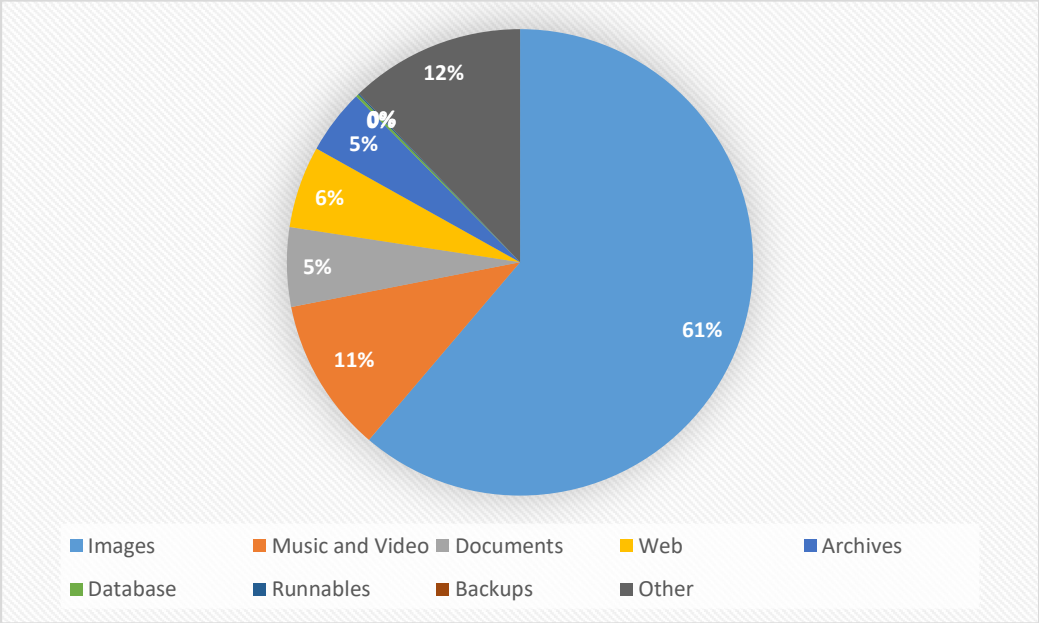

Figure 9 First scan pie chart for file type

The next list that has run through the tool, contains 1907 words that also consist of common names. However, this time we will break down the obtain results by their provider giving us a more detailed analysis on how organizations utilize the different cloud services.

As before, the first step of the analysis shows the number of public and private buckets and the buckets that do not exist (see table 5).

	Amazon	*Google*	*IBM*	*Total*	*Percentage*
Scanned Buckets	1170	1541	1907	4618	100%
Public Buckets	75	18	3	96	2%
Private Buckets	749	977	176	1902	41%
Not Found Buckets	346	546	1728	2620	57%

Table 5 Breakdown scan Bucket results

Like the list before, this new list also contains names that are not valid for Amazon and Google.

As the results show on table 5, Amazon buckets contain the most public buckets from the three providers. This result could be due to the fact that Amazon makes it easier for users to make the bucket public, as the user can choose that option during setup, even though by default the buckets are set to private.

Additionally, IBM contains fewer buckets with the names on the list, which could be due to being the one that is less used or just because the names used by companies are not as common as with the other two providers.

Next, the breakdown of the public and private files found inside of the public buckets are displayed (see table 6):

	Amazon	*Google*	*IBM*	*Total*	*Percentage*
Private Files	2688	6150	5	8843	37%
Public Files	12785	1422	1003	15210	63%
Total Files	15473	7572	1008	24053	100%

Table 6 Breakdown scan files result

On the following results shown in Figure 10. Amazon is also the one with most public files stored. Based on the results, the users that rely on Google Cloud may be more careful with their privacy settings as less than 20% of files are stored publicly.

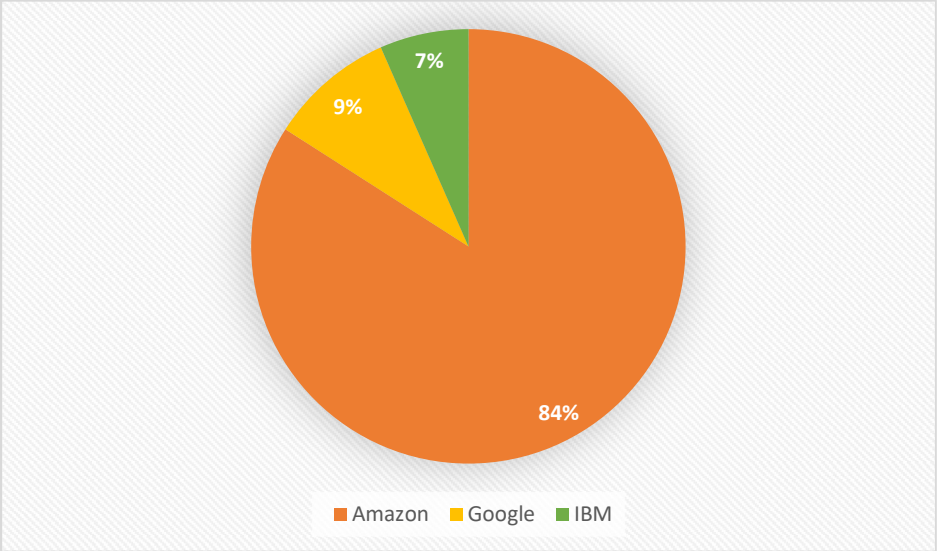

Figure 10 Breakdown scan pie chart distribution of public files by provider

Also, the public files found in IBM Cloud are mainly stored inside the same public bucket which could indicate that a particular user was trying to explicitly share a large amount public of content.

Lastly, the breakdown of the public files found by their type are displayed (see table 7):

Type	*Extensions*	*Amazon*	*Google*	*IBM*	*Total*
Images	jpg|png|gif|svg|bmp	7114	1143	0	8257
Music and Video	mp4|mp3|flv|wmv|swf	877	0	1000	1877
Documents	txt|docx|pdf|pptx|xlsx|csv	1027	80	0	1107
Web	html|xml|js|css|json	1478	2	0	1480
Archives	zip|rar|tar|gz|7z	376	163	0	539
Database	sql|dbf|dd|DDD	41	5	0	46
Runnables	jar|exe	584	0	0	584
Backups	bak	172	0	0	172
Other		1116	29	3	1148

Table 7 Breakdown scan type of public files result

Breaking down the public files by type into a graph (see figure 11), makes it even more clear that AWS S3 is the service that stores most public data of the three providers. Not only does AWS contain public documents and databases, but it also contains a large number of backups that may contain a lot of information about companies and their users.

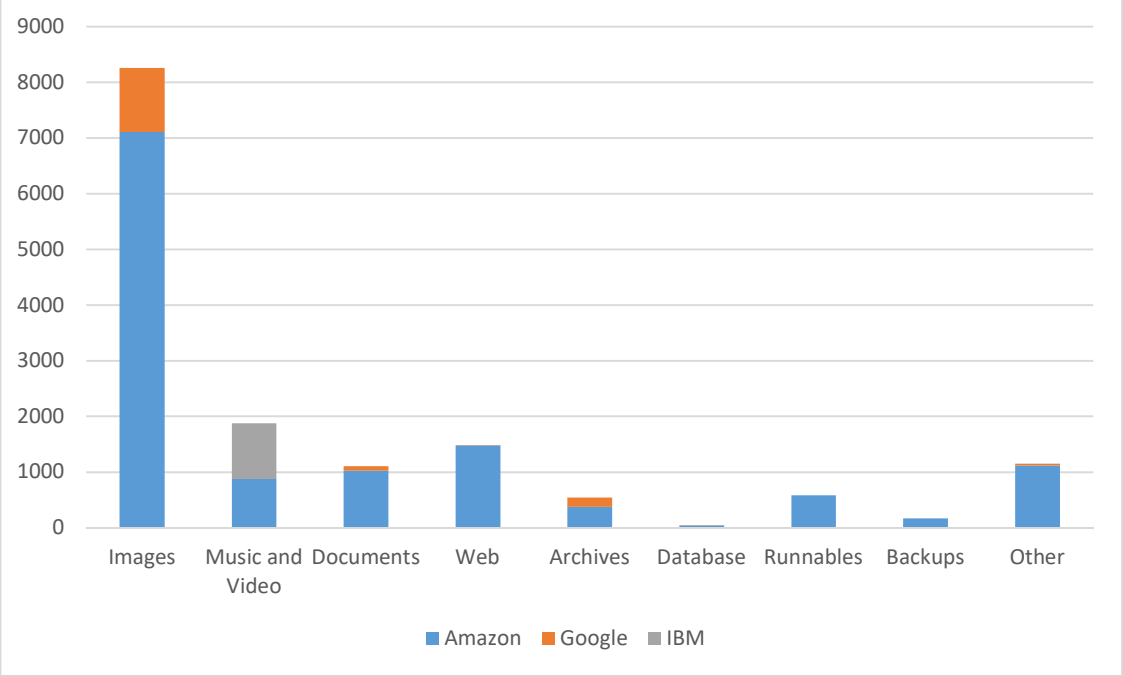

Figure 11 Stacked bar graph for file type and provider

Google Cloud and IBM Cloud clearly stored fewer public objects on their cloud servers. For Google and IBM allowing public access to buckets and files is more complex that in the case of Amazon, this reduces the probability that an inexpert administrator changes the permissions settings to allow public access by mistake.

2.3 Securing Unsecured Buckets & Other Sharing Methods

As results have shown, there is a lot of data that is already accessible to the public. This constitutes a major security threat which most companies are unaware of. To prevent future data leakage, users and developers have to be familiar on how to configure the buckets to be private again and use other methods to share the content.

The easiest way to make all the stored content private again, would be by setting the permissions of the bucket to be private, thus making all the content inside private. This option is available through web browser and CLI for AWS and Google Cloud, while only through CLI for IBM.

In the web browser, for both providers AWS and Google Cloud, the permissions that are currently set for "allUsers" have to be removed. On the other hand, when using the CLI the owners have to first log in and after introduce the necessary commands, provided by the User's Guide of each provider, that makes the bucket private again[4],[5],[6].

Additionally, there is another option that allows administrators to only make individual objects inside a bucket private again, leaving the rest available to the public. This option, requires going through each object and deleting the public permission. Like

[4] https://docs.aws.amazon.com/AmazonS3/latest/dev/access-control-block-public-access.html
[5] https://cloud.google.com/storage/docs/access-control/create-manage-lists#changing-acls
[6] https://console.bluemix.net/docs/services/cloud-object-storage/iam/public-access.html#allowing-public-access

the option before this option is allowed through web browser and CLI for AWS and Google Cloud and only on CLI for IBM Cloud.

As mentioned before, for web browser, the permissions for “allUsers” has to be removed and for CLI the necessary command has to be introduced. This option can be tedious and time consuming and for these reasons cloud storage should contemplate other options that allow administrators to share objects but only with other known and authenticated users.

The second option requires more work as developers need to go object per object deleting the public access from them. To simplify this process companies should create public and private buckets and separate files depending on the information they stored. Organizations should periodically audit their public buckets to ensure no private data has been placed in public buckets.

2.3.1 Other sharing methods

Even though making the content public is the easiest way to share objects with others, sensitive data should only be available for trusted users. Allowing access to only a group of users is much more secure than giving access to everybody and prevents the use of scan attacks in the search of public information.

All of the providers allow developers to give access to their contents through two methods: ACL, and pre-signed URL.

Through ACL the owner of the bucket can grant access to a group of trusted users to its content. This way only allows logged in users to access the content. As the owner of the bucket has to give access to a known and registered user, it verifies that only that user can access the content.

Additionally, ACL allows administrators to grant different levels of permissions so that the trusted users can not only view the content and also modify it and delete it (Access Control List (ACL) Overview - Amazon Simple Storage Service, 2018) (Access Control Lists (ACLs) | Cloud Storage | Google Cloud, 2018) (Creating Access Control Lists, 2018).

On the other hand, a pre-signed URL allows to give access to the content to anyone that has the URL, it is not as secure as ACL because non-logged in users would also be able to access the object if they somehow are able to obtain the URL. These URL are a time limited way to share an object with others even if they are not authenticated (Create a pre-signed URL, 2018) (Share an Object with Others - Amazon Simple Storage Service, 2018) (Signed URLs | Cloud Storage | Google Cloud, 2018).

Pre-signed URL allow to easily share content between users and even that is not as secure as ACL, the URLs configuration is more complex than the URLs of public buckets, and the time limit ensures that it is only share for a short period of time.

2.4 Problems Scanning Microsoft's Azure

The Bucket Scanner was first thought to also be able to scan Azure's Blobs making able to scan all four of the main Cloud Storage providers.

The same permissions study was carried out for Azure to get to know its permissions and how to access public content that is stored inside. Turns out that objects are stored in a different manner than the other providers.

The URL configuration for Azure differs from the other providers the following way:

https://<resource_name>.blob.core.windows.net/<blob_name>

Users have to first create a resource and then are allowed to make Blobs inside of such resource, making it harder to use a dictionary scan as Blobs with the same name can be stored inside resources that have different names. A scanner for Azure would have to run through to wordlist; one with names for resources and a second list with Blobs names. For each resource name that the scanner would find that exists, the scanner would then send a request for each name on the Blob list, increasing exponentially the number of requests the scanner would have to do.

Also, when making a request the browser display does not show if the Blob exists or not. It just returns the Error Code "Resource Not Found" for both existent and non-existent Blobs (see figure 12).

Figure 12 Error Code for Blobs

The Error Code, makes the Bucket Scanner unable to determine if a Blob exists for a given name on the list.

Microsoft's Azure has a security advantage over the other providers as it cannot be dictionary scanned as easily in search of sensitive data.

Chapter 3

Conclusion

As it has been proved three of the major public cloud storage providers can be dictionary force scanned with the help of this tool and a word list containing typical names that companies would use to name their buckets.

Even if most of the public files do not contain sensitive information, there are files that do contain important information that could harm users if sensitive data falls into the unethical hands.

All of the cloud providers set the buckets to private by default, so it is in hands of the developers to change the buckets to public and then also change the content stored inside to also be public.

Being able to download sensitive public data from the companies constitutes a major security threat that most companies are probably not aware of, as it is the cloud administrators that have to be careful with the files they store publicly and be sure that none of it could be utilized to harm the users or the company.

Companies and administrators using public cloud providers should perform periodical scans and audits on their buckets to make sure that there is no bucket public that stores sensitive information. With the help of this scanner, companies and authorized penetration testers would be able to perform scans on their system and know if they are safe or not. Public buckets that contain sensitive data, should be changed to private access in order to prevent possible data leaks.

Additionally, cloud administrators should be trained about the different information they will be uploading to the cloud and making sure that they understand how to properly set up permissions accordingly.

Furthermore, as all the providers have options to share the buckets with known and authenticated users, developers should learn and utilize those options to share the content when needed. Even if this option is more tedious for administrators, and requires more knowledge from part of all the employees, companies should encourage administrators to use these methods to share the content only with a few trusted users instead of making the content public to everyone.

Finally, as cloud object storage becomes more and more popular, written policies should take care on how the data is stored and shared through the cloud to minimize the possible human error of making sensitive data public instead of keeping it private.

In conclusion, cloud storage will continue to grow during the upcoming years, becoming a point of attack for hackers. Companies should make sure that their data is properly stored and not available to the public. The Bucket Scanner could be utilized as a tool for companies and penetration testers to search for and audit potentially sensitive data that may have uploaded and improperly configured to be publicly available.

References

Bowers, K. D., Juels, A., & Oprea, A. (2009). HAIL: A high-availability and integrity layer for cloud storage. In *Proceedings of the 16th ACM conference on Computer and communications security* (pp. 187-198). ACM.

Calder, B., Wang, J., Ogus, A., Nilakantan, N., Skjolsvold, A., McKelvie, S., ... & Haridas, J. (2011). Windows Azure Storage: a highly available cloud storage service with strong consistency. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles* (pp. 143-157). ACM.

Chickowski, E. (2018). Leaky Buckets: 10 Worst Amazon S3 Breaches. Retrieved from https://businessinsights.bitdefender.com/worst-amazon-breaches.

Introduction to Azure Storage - Cloud storage on Azure. (2018). Retrieved from https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction

Kamara, S., & Lauter, K. (2010). Cryptographic cloud storage. In *International Conference on Financial Cryptography and Data Security* (pp. 136-149). Springer, Berlin, Heidelberg.

Object Storage in the IBM Cloud. (2018). Retrieved from https://console.bluemix.net/docs/services/ibm-cos/index.html

O'Neill, P. (2018). Booz Allen Hamilton leaves 60,000 unsecured DOD files on AWS server - CyberScoop. Retrieved from https://www.cyberscoop.com/booz-allen-hamilion-amazon-s3-chris-vickery/

O'Sullivan, D. (2018). System Shock: How A Cloud Leak Exposed Accenture's Business. Retrieved from https://www.upguard.com/breaches/cloud-leak-accenture.

Rouse, M. (2016). Cloud Storage. Retrieved from https://searchstorage.techtarget.com/definition/cloud-storage.

Sheridan, K. (2017). 10 Major Cloud Storage Security Slip-Ups (So Far) this Year. Retrieved from https://www.darkreading.com/cloud/10-major-cloud-storage-security-slip-ups-(so-far)-this-year/d/d-id/1330122?image_number=6

User Profile & Data Cloud Storage - Liquidware. (2018). Retrieved from https://www.liquidware.com/products/profileunity/cloud-storage

What is Cloud Storage? | AWS. (2018). Retrieved from https://aws.amazon.com/what-is-cloud-storage/?nc1=f_ls

Bibliography

Access Control List (ACL) Overview - Amazon Simple Storage Service. (2018). Retrieved from https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html

Access Control Lists (ACLs) | Cloud Storage | Google Cloud. (2018). Retrieved from https://cloud.google.com/storage/docs/access-control/lists

Access Control Options | Cloud Storage | Google Cloud. (2018). Retrieved from https://cloud.google.com/storage/docs/access-control/

Bradford, C. (2018). Storage Wars: File vs Block vs Object Storage Systems | Object Based Storage Systems | StorageCraft. Retrieved from https://blog.storagecraft.com/object-storage-systems/

Bucket permissions. (2017). Retrieved from https://console.bluemix.net/docs/services/cloud-object-storage/iam/buckets.html#bucket-permissions

Create a pre-signed URL. (2018). Retrieved from https://console.bluemix.net/docs/services/cloud-object-storage/hmac/presigned-urls.html#create-a-presigned-url

Creating Access Control Lists. (2018). Retrieved from https://www.ibm.com/support/knowledgecenter/en/SSP4XS_2.1.0/com.ibm.icip.doc/web_access/t_add_ACL.html

How Google uses Node.js? | Hype.Codes. (2017). Retrieved from https://hype.codes/how-google-uses-nodejs

Manage anonymous read access to container and blobs. (2017). Retrieved from https://docs.microsoft.com/en-us/azure/storage/blobs/storage-manage-access-to-resources

Managing Access Permissions to Your Amazon S3 Resources - Amazon Simple Storage Service. (2018). Retrieved from https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html

Muñoz, J. (2016). Old Habits Die Hard - On S3, endpoints, regions, signatures and Boto 3. Retrieved from http://javiermunhoz.com/blog/2016/02/01/on-s3-endpoints-regions-signatures-and-boto-3

Share an Object with Others - Amazon Simple Storage Service. (2018). Retrieved from https://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html

Wood, R. (2018). Bucket Finder - DigiNinja. Retrieved from https://digi.ninja/projects/bucket_finder.php

Signed URLs | Cloud Storage | Google Cloud. (2018). Retrieved from https://cloud.google.com/storage/docs/access-control/signed-urls