Attacking & Defending AWS S3 Bucket

Sapna Singh
@Sapnas1ngh
Agenda

1. Introduction
2. Shared Responsibility Model
3. About AWS S3
4. S3 Breaches & Reasons
5. S3 Customer Responsibility
6. S3 Access Control Mechanism
7. S3 Attack Scenario
8. S3 Security Best Practices
9. S3 Monitoring & Logging
Who is really responsible for cloud security??

**Major Cloud Security Providers**
- Amazon Web Services
- Microsoft Azure
- Google Cloud
- IBM Cloud
- Oracle Cloud

**Service Models in Cloud**
- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)
Shared Responsibility Model

Customer benefits from the Cloud vendors Security and Compliance efforts

Moving to the cloud does NOT make YOU secure by default

There are several areas where security is the customer’s responsibility

Identity and Access Management is almost always the customer’s responsibility

Introduction

About AWS S3

S3 Breaches and Reasons

S3 Customer Responsibility

S3 Access Control Mechanism

S3 Attack Scenario

S3 Security Best Practices

Monitoring and logging for S3

Queries
AWS Shared Responsibility Model

Customer Responsible for security **IN** the cloud

Customer Data

Platform/Application/Identity & Access Management

Operating Systems/Network/Firewall Configuration

Encryption/Network Traffic Monitoring

AWS Responsible for security **OF** the cloud

Compute

Storage

Database

Networking

AWS Global Infrastructure (Regions, Availability Zones, Edge Locations)
Amazon Simple Storage Service

- Simple Storage Service
- Cost effective object storage
- Highly scalable, reliable, durable and fast
- Security compliance and audit
AWS S3 Security Breaches

GoDaddy Leaks ‘Map of the Internet’ via Amazon S3 Cloud Bucket Misconfig

198 million Americans hit by 'largest ever' voter records leak

Insurance startup AgentRun accidentally leaks customers’ personal and health information in cloud configuration error

Accenture exposed by misconfigured AWS storage

Misconfigured Amazon Storage Exposes 14 Million Verizon Customer Records

Verizon Hit by Another Amazon S3 Leak

Security keys and 40,000 plaintext passwords.
Reasons for AWS S3 Breaches

- Misconfiguration
- Human error
- Too many APIs
- Third Party Integration
- Lack of Governance
- No Data classification
AWS S3 Customer Responsibilities

- How to store it
- What data to store
- Whether or not to encrypt data
- Who has access to data
- What security features/tools to be used
S3 Access Control Mechanism

• Identity and Access Management (IAM)
• S3 Bucket Policy
• S3 Access Control List (ACL)
• Pre-signed url
Identity and Access Management (IAM)

- Centralized and powerful tool for authentication and authorization
- Attached to users, groups or roles
- IAM policies specify what actions are allowed or denied on what AWS resources
S3 Bucket Policies

- Bucket Policies are attached to S3 bucket
- Bucket Owner controls the S3 bucket policies
- Access control remains within S3 environment
- Primarily used for Data sharing or web based access

```json
{
    "Version": "2012-10-17",
    "Id": "Policy1542652449582",
    "Statement": [
        {
            "Sid": "Stmt1542652447726",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::myfirstpublicbucketfortest/*"
        }
    ]
}
```
S3 Access Control List

- ACL is legacy control
- ACLs are applied to bucket or object level
- ACLS grant access (can't explicitly deny)
- More fine-grained than bucket policy

```
Everyone

Access to the object
- Read object

Access to this object's ACL
- Read object permissions
- Write object permissions
```
Attack Scenario: AWS S3 Bucket

**S3 ACLS**
- READ
- Write
- READ_ACP
- Write_ACP
- FULL_control

**S3 Bucket Policies**
- s3:DeleteObject
- s3:GetObject
- s3:GetObjectAcl
- s3:PutObject
- s3:PutObjectAcl
- s3:CreateBucket
- s3:DeleteBucket
- s3:ListBucket
- s3:GetBucketAcl
- s3:DeleteBucketPolicy

**IAM**
- IAM users/groups
- Pre-signed urls

---

**Recon**
- HTML inspection
- HTTP response
- Google dork
- Brute-force
- Permutation

**Scanning**
- S3 ACLs
- S3 Bucket policies
- S3 IAM

**Exploitation**
- List
- Read
- Upload
- Download
- Delete
Target bucket name, domain name, S3 urls, bucket:region

```
root@testlab:~/S3Scanner# python s3scanner.py -l -d myfirstcompany
[not found] : myfirstcompany
root@testlab:~/S3Scanner# cat wordlist.txt
prod
production
dev
development
staging
external
internal
testing
test
data
web
www
backup
archive
uat
```
S3Scanner

```bash
# for name in $(cat wordlist.txt);do python s3scanner.py -l -d myfirstcompany-$name;done
[not found] : myfirstcompany-prod

[found] : myfirstcompany-production | 11.3 KiB | ACLs: {'authUsers': [], 'allUsers': ['READ'],
[found] : myfirstcompany-production - Attempting to dump...this may take a while.
[not found] : myfirstcompany-dev

[found] : myfirstcompany-development | 8.0 KiB | ACLs: AccessDenied
[found] : myfirstcompany-development - Attempting to dump...this may take a while.
[found] : myfirstcompany-staging | AccessDenied | ACLs: AccessDenied

[not found] : myfirstcompany-external
[not found] : myfirstcompany-internal
[not found] : myfirstcompany-testing
[found] : myfirstcompany-test | AccessDenied | ACLs: AccessDenied
[not found] : myfirstcompany-data

[found] : myfirstcompany-web | AccessDenied | ACLs: {'authUsers': [], 'allUsers': ['WRITE', '
[not found] : myfirstcompany-www
[not found] : myfirstcompany-backup
[not found] : myfirstcompany-archive
[not found] : myfirstcompany-uat
```
### AWS-CLI

<table>
<thead>
<tr>
<th>High Level command with AWS CLI</th>
<th>API Level commands with AWS CLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp</td>
<td>copy-object</td>
</tr>
<tr>
<td>ls</td>
<td>create-bucket</td>
</tr>
<tr>
<td>mb</td>
<td>create-multipart-upload</td>
</tr>
<tr>
<td>mv</td>
<td>delete-bucket</td>
</tr>
<tr>
<td>presign</td>
<td>delete-bucket-analytics-configuration</td>
</tr>
<tr>
<td>rb</td>
<td>delete-bucket-encryption</td>
</tr>
<tr>
<td>rm</td>
<td>delete-bucket-lifecycle</td>
</tr>
<tr>
<td>sync</td>
<td>delete-bucket-policy</td>
</tr>
<tr>
<td>website</td>
<td>delete-bucket-replication</td>
</tr>
<tr>
<td></td>
<td>delete-bucket-website</td>
</tr>
<tr>
<td></td>
<td>delete-object</td>
</tr>
<tr>
<td></td>
<td>delete-objects</td>
</tr>
<tr>
<td></td>
<td>delete-public-access-block</td>
</tr>
<tr>
<td></td>
<td>get-bucket-acl</td>
</tr>
<tr>
<td></td>
<td>get-bucket-website</td>
</tr>
<tr>
<td></td>
<td>get-object</td>
</tr>
<tr>
<td></td>
<td>get-object-acl</td>
</tr>
</tbody>
</table>

**Introduction**

- About AWS S3
- S3 Breaches and Reasons
- S3 Access Control Mechanism
- S3 Customer Responsibility
- Monitoring and logging for S3
- Queries

**Shared Responsibility Model**

**Agenda**

- S3 Attack Scenario
- S3 Security Best Practices
root@testlab:~$S3Scanner# aws s3 ls s3://myfirstcompany

An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist.

root@testlab:~$S3Scanner# for name in $(cat wordlist.txt);do echo "Testing bucket: s3://myfirstcompany-$name;aws s3 ls s3://myfirstcompany-$name;echo "***";done

Testing bucket: s3://myfirstcompany-prod

An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist.

***

Testing bucket: s3://myfirstcompany-production

2019-02-15 18:00:59       55 Password.txt
2019-02-16 23:33:19       11499 ServerInfo.docx

***

Testing bucket: s3://myfirstcompany-dev

An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist.

***

Testing bucket: s3://myfirstcompany-development

2019-02-16 23:33:48       8145 NetworkDetails.xlsx
2019-02-15 18:01:49       18 logs.txt

***

Testing bucket: s3://myfirstcompany-staging

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied.

***
Introduction

About AWS S3

S3 Breaches and Reasons

S3 Access Control Mechanism

Monitoring and logging for S3 Queries

S3 Customer Responsibility

Agenda

S3 Attack Scenario

S3 Security Best Practices

root@testlab:/S3Scanner# for name in $(cat wordlist.txt);do aws s3 cp testcopy.txt s3://myfirstcompany-$name;done

upload failed: ./testcopy.txt to s3://myfirstcompany-prod/testcopy.txt An error occurred (NoSuchBucket) when calling the PutObject operation: The specified bucket does not exist

upload failed: ./testcopy.txt to s3://myfirstcompany-production/testcopy.txt An error occurred (AccessDenied) when calling the PutObject operation: Access Denied

upload failed: ./testcopy.txt to s3://myfirstcompany-dev/testcopy.txt An error occurred (NoSuchBucket) when calling the PutObject operation: The specified bucket does not exist

upload failed: ./testcopy.txt to s3://myfirstcompany-development/testcopy.txt An error occurred (AccessDenied) when calling the PutObject operation: Access Denied

upload failed: ./testcopy.txt to s3://myfirstcompany-staging/testcopy.txt An error occurred (AccessDenied) when calling the PutObject operation: Access Denied

upload failed: ./testcopy.txt to s3://myfirstcompany-external/testcopy.txt An error occurred (NoSuchBucket) when calling the PutObject operation: The specified bucket does not exist

upload failed: ./testcopy.txt to s3://myfirstcompany-internal/testcopy.txt An error occurred (NoSuchBucket) when calling the PutObject operation: The specified bucket does not exist

upload failed: ./testcopy.txt to s3://myfirstcompany-testing/testcopy.txt An error occurred (AccessDenied) when calling the PutObject operation: Access Denied

upload failed: ./testcopy.txt to s3://myfirstcompany-test/testcopy.txt An error occurred (AccessDenied) when calling the PutObject operation: Access Denied

upload failed: ./testcopy.txt to s3://myfirstcompany-data/testcopy.txt An error occurred (NoSuchBucket) when calling the PutObject operation: The specified bucket does not exist

upload failed: ./testcopy.txt to s3://myfirstcompany-www/testcopy.txt An error occurred (NoSuchBucket) when calling the PutObject operation: The specified bucket does not exist

root@testlab:/S3Scanner#
S3 Security Best Practices

- Least Privilege
- IAM
- Bucket Policy
- Bucket ACL
- Pre-signed url
- Amazon S3 Public Block
- Encryption
- Replication
Amazon S3 Block Public Access

- Amazon S3 Block Public access ensures S3 buckets and objects do not have public access.
# Bucket Permissions

## Public access settings for this bucket

Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the Amazon S3 block public access settings at the account level. [Learn more](#)

<table>
<thead>
<tr>
<th>Manage public access control lists (ACLs)</th>
<th>Manage public bucket policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block new public ACLs and uploading public objects <em>(Recommended)</em></td>
<td>Block new public bucket policies <em>(Recommended)</em></td>
</tr>
<tr>
<td><strong>True</strong></td>
<td><strong>True</strong></td>
</tr>
<tr>
<td>Remove public access granted through public ACLs <em>(Recommended)</em></td>
<td>Block public and cross-account access if bucket has public policies <em>(Recommended)</em></td>
</tr>
<tr>
<td><strong>True</strong></td>
<td><strong>True</strong></td>
</tr>
</tbody>
</table>
Bucket Permissions

Amazon S3  >  myfirstcompany-development

Public access settings for this bucket

Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the Amazon S3 block public access settings at the account level. Learn more
S3 Breaches and Reasons

S3 Access Control Mechanism

Monitoring and logging for S3

Shared Responsibility Model

S3 Attack Scenario

S3 Security Best Practices
# Bucket Permissions

## Public access

<table>
<thead>
<tr>
<th>Group</th>
<th>List objects</th>
<th>Write objects</th>
<th>Read bucket permissions</th>
<th>Write bucket permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Everyone</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>-</td>
</tr>
</tbody>
</table>

### S3 buckets

- **myfirstcompany-development**
  - Access: Public
  - Region: US East (N. Virginia)
  - Date created: Feb 15, 2019 2:04:56 PM GMT+0300

- **myfirstcompany-production**
  - Access: Public
  - Region: US East (N. Virginia)
  - Date created: Feb 15, 2019 2:03:54 PM GMT+0300
## Bucket Access Status

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Public</strong></td>
<td>Everyone has access to one or more of the following: List objects, Write objects, Read and write permissions.</td>
</tr>
<tr>
<td><strong>Objects can be public</strong></td>
<td>The bucket is not public, but anyone with the appropriate permissions can grant public access to objects.</td>
</tr>
<tr>
<td><strong>Buckets and objects not public</strong></td>
<td>The bucket and objects do not have any public access.</td>
</tr>
<tr>
<td><strong>Only authorized users of this account</strong></td>
<td>Access is isolated to IAM users and roles in this account and AWS service principals because there is a policy that grants public access.</td>
</tr>
</tbody>
</table>
Encryption

Use Server-Side Encryption

• Amazon S3-Managed Keys (SSE-S3)
• AWS KMS-Managed Keys (SSE-KMS)
• Customer-Provided Keys (SSE-C)

Use Client-Side Encryption

• Use an AWS KMS-managed customer master key
• Use a client-side master key

Default Encryption

• AES-256
• AWS-KMS
Cross Region Replication

• Secure business critical data
• Comply with compliance requirements
• CRR overwrites the access control list
• Provide full access to the owner of the replica
• Maintain object copies under different ownership
• Segregation of rights between the original copy and the replica
What Else?

• Review open buckets regularly
• Use S3 inventory
• Enable MFA delete
• Enable bucket versioning
• Enable bucket logging
Monitoring & Logging

- Log all API calls for S3 bucket
- Log all configuration changes
- Monitor all changes to S3 policy changes
- Track all application accessing S3
- Set up incident use cases for S3 bucket
- Monitor malicious behavior
- Event-driven security
Monitoring and Logging

- Cloud Trail
- Guard Duty
- Cloud Watch
- Cloud Config
- Trusted Advisor
- AWS Macie
AWS Cloud Trail

AWS CloudTrail increases visibility into user and resource activity by recording AWS Management Console actions and API calls.
Amazon GuardDuty is a threat detection service that continuously monitors for malicious or unauthorized behavior to help protect AWS accounts and workloads.
AWS Trusted Advisor analyzes your AWS environment and provides best practice recommendations in five categories:

- IAM Use
- Amazon S3 Bucket Permissions
- MFA on Root Account
- Amazon EBS Public Snapshots
- Amazon RDS Public Snapshots
- Security Groups-Specific Ports Unrestricted
Description: Checks buckets in Amazon Simple Storage Service (Amazon S3) that have open access permissions or allow access to any authenticated AWS user. Bucket permissions that grant List access can result in higher than expected charges if objects in the bucket are listed by unintended users at a high frequency. Bucket permissions that grant Upload/Delete access create potential security vulnerabilities by allowing users that to add, modify, or remove items in a bucket.

<table>
<thead>
<tr>
<th>Region Name</th>
<th>Region API Parameter</th>
<th>Bucket Name</th>
<th>ACL Allows List</th>
<th>ACL Allows Upload/Delete</th>
<th>Status</th>
<th>Policy Allows Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>us-east-1</td>
<td>us-east-1</td>
<td>myfirstcompany-development</td>
<td>No</td>
<td>No</td>
<td>Yellow</td>
<td>Yes</td>
</tr>
<tr>
<td>us-east-1</td>
<td>us-east-1</td>
<td>myfirstcompany-production</td>
<td>Yes</td>
<td>No</td>
<td>Yellow</td>
<td>No bucket policy</td>
</tr>
<tr>
<td>us-east-1</td>
<td>us-east-1</td>
<td>myfirstcompany-staging</td>
<td>No</td>
<td>No</td>
<td>Green</td>
<td>No bucket policy</td>
</tr>
<tr>
<td>us-east-1</td>
<td>us-east-1</td>
<td>myfirstcompany-test</td>
<td>No</td>
<td>No</td>
<td>Green</td>
<td>No bucket policy</td>
</tr>
<tr>
<td>us-east-1</td>
<td>us-east-1</td>
<td>myfirstcompany-web</td>
<td>No</td>
<td>Yes</td>
<td>Red</td>
<td>No bucket policy</td>
</tr>
<tr>
<td>us-east-2</td>
<td>us-east-2</td>
<td>myfirstprivatebucketforitest</td>
<td>No</td>
<td>No</td>
<td>Green</td>
<td>No bucket policy</td>
</tr>
<tr>
<td>us-east-2</td>
<td>us-east-2</td>
<td>myfirstpublicbucketforitest</td>
<td>No</td>
<td>No</td>
<td>Yellow</td>
<td>Yes</td>
</tr>
<tr>
<td>us-east-2</td>
<td>us-east-2</td>
<td>myfirstbucketforwebhosting</td>
<td>No</td>
<td>No</td>
<td>Yellow</td>
<td>Yes</td>
</tr>
<tr>
<td>us-east-2</td>
<td>us-east-2</td>
<td>testbucketpolicypermissions</td>
<td>Yes</td>
<td>Yes</td>
<td>Red</td>
<td>No bucket policy</td>
</tr>
</tbody>
</table>
Macie analyzes and processes data stored in Amazon S3 buckets.

- Analyze and Process data in S3 bucket
- Classify sensitive and business-critical data stored
- Uses ability of CloudTrail for capturing API activity for S3 object
- Continuous monitor and discover new data
Q&A!!