# AWS Security Series

## Part 5: AWS S3 Security: Attack and Defense

Nitin Sharma
CyberSecurity and DevSecOps Engineer
LinkedIn: linkedin.com/in/nitins87
Quora: quora.com/profile/NitinS-1
Blog: 4hathacker.in

# Contents

$ whoami

$ AWS S3: Attack/Breach Scenarios

$ Introduction to Threat Modeling

$ Threat Modeling: AWS S3

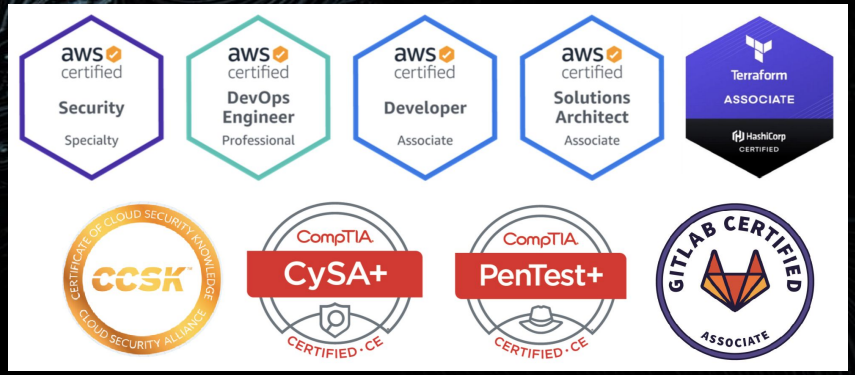
$ AWS S3: Preventing (#Defend) Attacks

$ AWS S3: Detecting (#Defend) Attacks

$ AWS S3: Demo

# $ whoami

- Cybersecurity and DevSecOps professional experienced in Cloud Security, Container Security and DevOps Research
- Certifications:

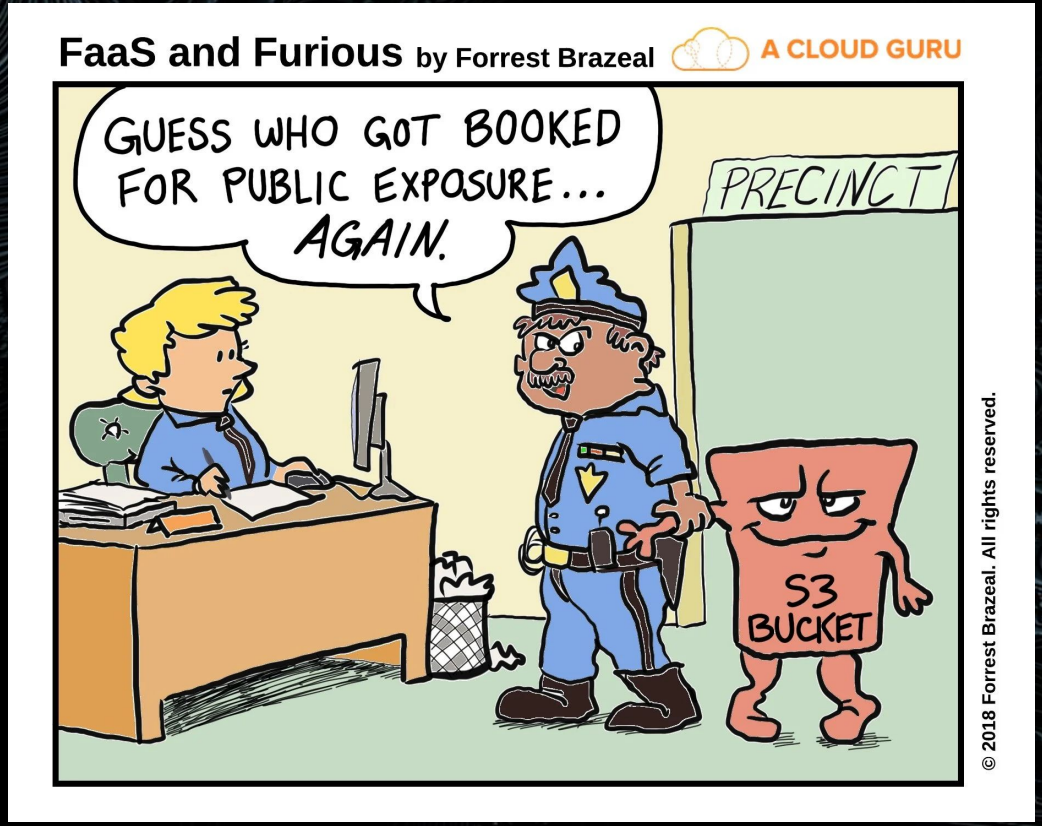




- Published author for "Securing Docker - The Attack & Defense Ways" book under CyberSecrets Publication
- Half Marathon runner, Cyclist and Fitness Enthusiast
- Helping out beginners in Cloud, DevOps and CyberSec at Quora

AWSUGNCR
aws
AWS USER GROUP NCR

# $ whoami (finding mistakes in the past)




(Source: ACloudGuru)

# $ AWS S3 Attack/Breach Scenarios

- Scenario 1: Public S3 Bucket with Customer Data


## Data Breach: Millions of Senior Citizens' Names, Emails, and Phone Numbers Compromised in Senior Care Review Website Database Breach

Published by Cyber Research Team on August 09, 2021

**WizCase's security team led by Ata Hakcil has found a major breach affecting a bucket owned by SeniorAdvisor, "one of the largest consumer ratings and reviews websites for senior care and services across the United States and Canada." This breach compromised users' names, surnames, phone numbers, and more.** Millions of people were left vulnerable in the misconfigured bucket. There was no need for a password or login credentials to access this information, and the data was not **encrypted**.

(Source:WizCase)

# $ AWS S3 Attack/Breach Scenarios

- Scenario 2: GitHub code committed with AWS Access Keys associated with S3 permissions to Customer Bucket (little tricky but bots are everywhere on GitHub...)

**Your AWS Access Key is Exposed for AWS Account** [redacted]

A

Amazon Web Services, Inc. <no-reply-aws@amazon.com>
Wed 11/4/2020 10:05 PM
To: [redacted]

Hello,

We have become aware that the AWS Access Key AKIAWHJ2ODPBUKZJIIOX belonging to IAM User github_deployer along with the corresponding Secret Key is publicly available online at https://github.com/th3g1itch/rabbit-hole/blob/be396ce4328387f08431022cc0b1f437e36bc8aa/enter.py.

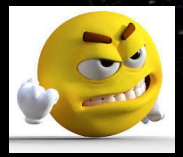
Your security is important to us, and this exposure of your account's IAM credentials poses a security risk to your AWS account and could lead to excessive charges from unauthorized activity or abuse, and violates the AWS Customer Agreement or other agreement with us governing your use of our Services.

To protect your account from excessive charges and unauthorized activity, we have applied the IAM Policy "AWSCompromisedKeyQuarantine" on the IAM User listed above. The IAM Policy applied to the User protects your account by limiting permissions for high risk AWS services.

(Source: Pawel Rzepa Medium Blog)

# $ AWS S3 Attack/Breach Scenarios

- Scenario 3: Attackers encrypting S3 buckets in victim AWS account with their own KMS Key (S3 Ransomware Scenario)
  - Attacker creates a KMS key in their own "personal" AWS account (or another compromised account) and provides "the world" access to use that KMS key for encryption.
  - Attacker identifies a target S3 bucket and gains write level access to it.
  - Attacker checks the configuration of the bucket to determine if it is able to be targeted by ransomware.
  - Attacker uses the AWS API to replace each object in a bucket with a new copy of itself (encrypted with KMS key)
  - Attacker schedules the deletion of the KMS key
  - Attacker uploads a final file such as "ransom-note.txt" without encryption



(Source: Rhino Security Labs)

# $ AWS S3 Attack/Breach Scenarios

- Scenario Nth:

  There could be ‘N’ different such scenarios to tackle the AWS S3 Attackers.

  Hence, an efficient way to address the attacker concerns is needed !

Threat Modeling...

# $ Threat Modeling: Introduction(High Level)

- **Threat**: Any circumstance or event with the potential to harm an information system through unauthorized access, destruction, disclosure, modification of data, and/or denial of service. [Design Time Activity]

- Core steps for threat modelling:

  1. Identify assets, actors, entry points, components, use cases, and trust levels, and include these in a design diagram.

  2. Identify a list of threats.

  3. Per threat, identify mitigations, which may include security control implementations.

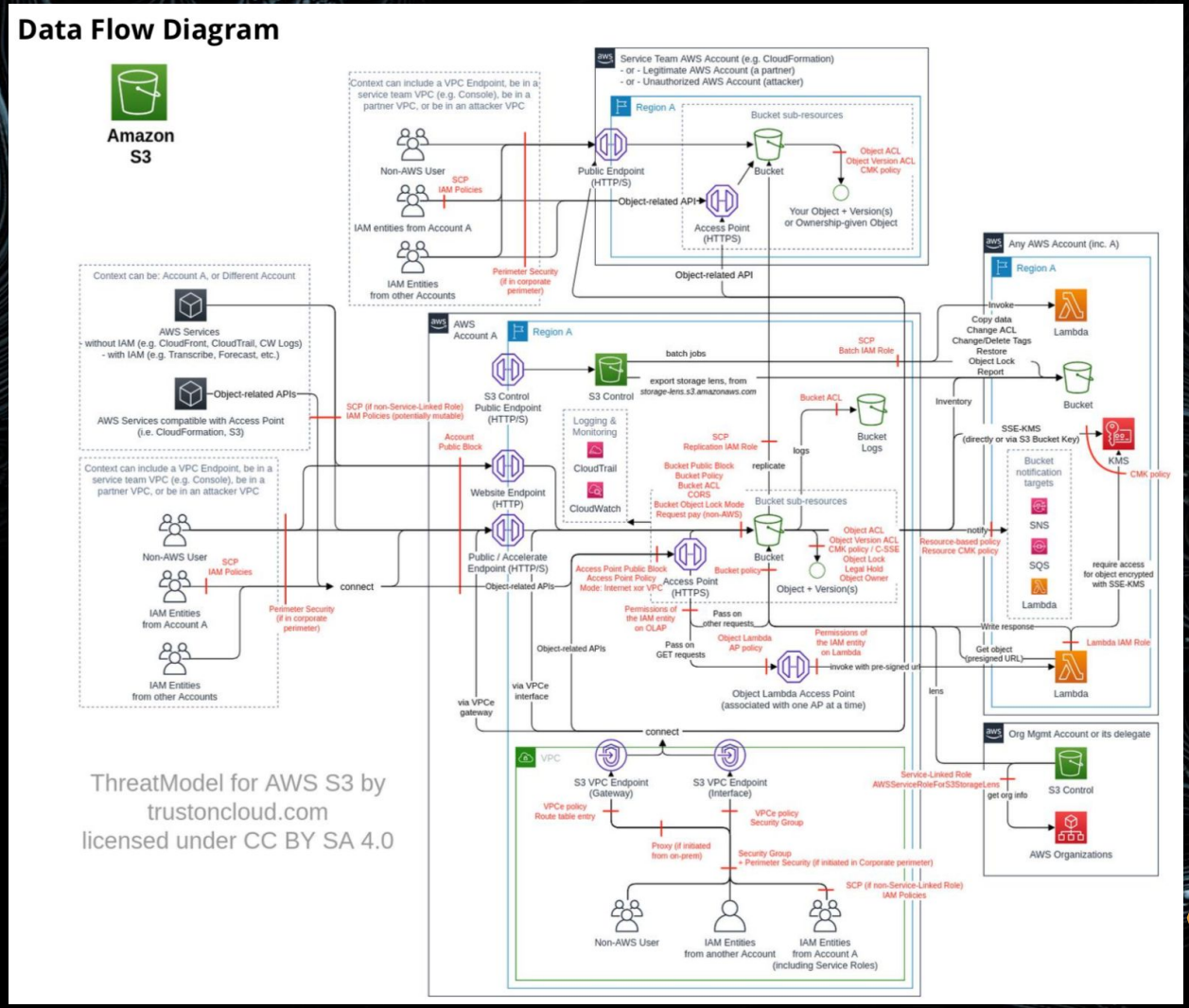
  4. Create and review a risk matrix to determine if the threat is adequately mitigated.

# $ MITRE ATT&CK matrix for AWS S3

| 1 Initial Access | 2 Execution | 3 Persistence | 4 Privilege Escalation | 5 Defense Evasion | 6 Credential Access |
|---|---|---|---|---|---|
| | | | Gain access by modifying or deleting important object tags [S3.T33] | Exfiltrate data via ungoverned S3 endpoint [S3.T45] | |
| | | | Reduce bucket security by deleting the bucket policy [S3.T38] | Evade detection by disabling S3 access logs [S3.T51] | |

| 7 Discovery | 8 Lateral Movement | 9 Collection | 10 Exfiltration | 11 Impact |
|---|---|---|---|---|
| Recon of AWS root account emails using email ACL grantee feature [S3.T19] | Reduce bucket security by modify the bucket public access block [S3.T52] | Move prod data in non-prod environment [S3.T11] | Bucket takeover to gather data [S3.T1] | Grant unauthorized access to a private bucket by changing bucket ACL [S3.T4] |
| | Reduce bucket security by modify the account public access block [S3.T53] | Unauthorized collection of data by swapping access point [S3.T28] | Unauthorized access to data via bucket replication [S3.T2] | Use bucket to upload a malware or modify an object to include a malware [S3.T14] |

# $ Threat Modeling: AWS S3



# $ Threat Modeling: AWS S3

→ Pick one Feature of S3 (Involved in your DFD)

→ Identify Associated IAM/Bucket Actions

→ Create Threat List & Map it to MITRE ATT&CK

→ Set Security Control Objectives and Priorities

→ Testing & Validation (#Defend)

# $ Threat Modeling: AWS S3

→ **Pick one Feature of S3 (Involved in your DFD)**

- **Object Upload/Download**

  You can upload and download virtually any number of objects to an external S3 bucket you authorized to.

# $ Threat Modeling: AWS S3

→ **Identify Associated IAM/Bucket Actions**

| Action | IAM Permission |
|---|---|
| Retrieves an object from Amazon S3. | s3:GetObject |
| Adds an object to a bucket. | s3:PutObject |
| Sets the access control list (ACL) permissions for an object. You must have WRITE_ACP permission to set the ACL of an object. | s3:PutObjectAcl |

# $ Threat Modeling: AWS S3

## → Create Threat List & Threat Map with MITRE

| Name | CVSS |
|---|---|
| Exfiltrate your data hosted on an external bucket, by using of a compromised IAM access from Internet | Medium (6.7) |
| Exfiltrate data by uploading it to an attacker bucket using a non-authenticated user or an unauthorized external IAM entity via one of your S3 VPC endpoints. | Medium (6.2) |
| Exfiltrate data stored on S3 via AWS services | Medium (5.8) |
| Exfiltrate data to an attacker bucket via public endpoint | Medium (5.7) |
| Unauthorized upload of a private object in an accessible bucket (e.g. public) you do not own. | Medium (5.7) |
| Exfiltrate data by using a S3 VPC endpoint to upload data to an attacker bucket using an internal IAM entity. | Medium (5.5) |
| Unauthorized modification of an object to become public or accessible in a private bucket you do not own by changing object ACL | Medium (5.2) |
| Intercept data in transit to an external bucket | Medium (4.6) |
| Unauthorized object restore into an unauthorized bucket | Medium (4.5) |
| Upload in an authorized external bucket, but in an incorrect AWS account | Medium (4.0) |
| Loss of ownership of an object | Low (2.6) |
| Exfiltrate data via ungoverned S3 endpoint | Low (1.9) |
| Use of less secure or old S3 features | Low (1.9) |

| | |
|---|---|
| **Threat Id** | S3.T3 |
| **Name** | Exfiltrate your data hosted on an external bucket, by using of a compromised IAM access from Internet |
| **Description** | IAM credentials can be compromised. An attacker can use a compromised but authorized credential to download your object from an external bucket via the public endpoint (using or not their own VPC endpoint). |
| **Goal** | Data theft |
| **Mitre ATT&CK** | TA0010 |
| **CVSS** | Medium (6.7) |
| **IAM Access** | { "UNIQUE": "s3:GetObject" } |

# $ Threat Modeling: AWS S3

## → Set Security Control Objectives and Priorities

| Control Objectives | Priority | # of associated Controls | | |
|---|---|---|---|---|
| | | Directive | Preventative | Detective |
| **Identify and ensure the protection all external buckets hosting your objects**<br>Identify all buckets you don't control hosting your objects, define their authorized data classification, identify their respective owners (and AWS account ID), their ObjectACL requirements (including S3 Object Ownership), and get assured for the protection (e.g. through contractual agreement, verified by assurance programs, or using this ThreatModel). | High | 1 | - | - |
| **Enforce good coding practice**<br>Ensure all S3 buckets interacted with are in the correct AWS account (e.g. using the condition in all compatible S3 requests: x-amz-expected-bucket-owner and x-amz-source-expected-bucket-owner) | Medium | 1 | - | - |
| **Monitor S3 with Amazon GuardDuty and Macie**<br>Enable and monitor S3 protection in Amazon GuardDuty in all AWS accounts in all Regions, and protect it using GuardDuty ThreatModel | Low | 1 | - | - |
| **Encrypt or tokenize critical data**<br>Aligned with your data governance, encrypt on the client side - or tokenize - appropriate data | Very Low | 1 | - | - |

# $ Threat Modeling: AWS S3

## → Testing & Validation

**Identify and ensure the protection all external buckets hosting your objects**

| Type | Control | Testing | Effort | Feature Class(es) | Threat(s) and Impact | CVSS-weighted Priority |
|---|---|---|---|---|---|---|
| Directive (COSO)<br>Protect (NIST CSF) | [S3.C11]<br>Identify all buckets you don't control hosting your objects, define their authorized data classification, identify their respective owners (and AWS account ID), their ObjectACL requirements (including S3 Object Ownership), and get assured for the protection (e.g. through contractual agreement, verified by assurance programs, or using this ThreatModel). | Request the list of all authorized external buckets authorized to host your objects, their respective owners (and AWS account ID), their ObjectACL requirements (including S3 Object Ownership), their data classification and the mechanism used to ensure the security of those buckets | Medium | S3.FC1<br>S3.FC16<br>S3.FC5 | S3.T3 (High)<br>S3.T5 (Very Low)<br>S3.T6 (Low)<br>S3.T7 (Very Low)<br>S3.T8 (Very Low)<br>S3.T9 (Very Low)<br>S3.T11 (Low)<br>S3.T14 (Very Low)<br>S3.T15 (Very Low)<br>S3.T21 (Very Low)<br>S3.T31 (High)<br>S3.T43 (Low) | High |
| Preventative (COSO)<br>Protect (NIST CSF) | [S3.C12, depends on S3.C11]<br>Allow only authorized ACL on objects for bucket you don't control (e.g. using IAM and VPC endpoint policy with the ACL conditions) | Put an object with an unauthorized ACL, it should be denied. | Medium | S3.FC1 | S3.T5 (Medium)<br>S3.T6 (High) | Medium |
| Detective (COSO)<br>Detect (NIST CSF) | [S3.C13, depends on S3.C11]<br>Monitor that only authorized external buckets are used (e.g. via CloudTrail S3 data events in resources[].accountId and resources[].ARN). Both account ID and bucket name must be verified. | Make a call to an unauthorized bucket, it should be detected | Low | S3.FC1<br>S3.FC5 | S3.T1 (Low)<br>S3.T7 (Low)<br>S3.T11 (Low)<br>S3.T21 (Low)<br>S3.T31 (Medium) | Medium |
| Directive (COSO)<br>Protect (NIST CSF) | [S3.C14, depends on S3.C11]<br>Scan all data before uploading to an external bucket to ensure the classification of the data is aligned with the bucket classification (e.g. using Macie). | Request 1) the mechanism ensuring all data are scanned for proper data classification before upload to an external bucket are configured, 2) its records of execution for all object upload flows, and 3) plan to move any older object upload flows | High | S3.FC1<br>S3.FC16<br>S3.FC5 | S3.T5 (High)<br>S3.T14 (High)<br>S3.T15 (Medium) | Medium |

# $ AWS S3 Security Best Practices

S3 Security Best Practices

## Preventative

- Correct Policies and no Public Access
- Least Privilege Access
- Use of IAM Roles to access S3 Buckets
- Enable MFA Delete
- Encryption at Rest
- Encryption at Transit
- Use S3 Object Lock
- Use S3 Cross Region Replication for backups
- Use VPC endpoints for S3 Access

**Demo Time...**

## Monitoring & Auditing (Detective)

- Identify and Audit all your S3 buckets
- Monitoring S3 Bucket Activity (S3 API calls)
- Enable S3 Server Access Logging
- Use AWS Cloudtrail to record S3 Data Events
- Enable AWS Config to simplify auditing for misconfigurations
- Use AWS Macie to protect sensitive info.
- Check Trusted Advisor


# Thank You...

**For queries feel free to connect with AWS Delhi User Group at:**

1. **AWS Delhi User Group (MeetUp) - Learn Together, Grow Together**
2. **AWS User Group Delhi NCR (Follow us on LinkedIn Page)**