Ransomware in the Cloud: Breaking Down the Attack Vectors

By Ofir Balassiano, Ofir Shaty

Published: 2023-11-29 · Archived: 2026-04-05 22:40:59 UTC

The number of ransomware attacks, especially those involving Amazon S3 buckets, Azure Storage accounts and other cloud assets, has increased in recent years. Still, most organizations don't act to reduce these attacks or lower their impact. Our research shows, for example, that only 31% of S3 buckets have versioning enabled, an essential for data recovery, while just two-thirds of sensitive buckets have logging enabled, a prerequisite for detection.

In the battle against ransomware, understanding your enemy is half the victory. By diving into the minds of attackers and evaluating the pros and cons of their techniques, we can anticipate their next moves and fortify our defenses.

In this blog post, we cut through the theory and delve into the practical aspects of ransomware attacks involving cloud environments. To fortify our defense strategies, we draw from real-world data and simulations to explore attack vectors and evaluate both their prevalence and their potential impact.

By examining the modus operandi of ransomware attackers, we aim to understand them, gaining the ability to predict when a technique might occur, as well as the measures to effectively nip it in the bud.

Ransomware Techniques

Ransomware attacks in the cloud are carried out via four main techniques — data deletion, override, re-encryption and disable key, as seen in Figure 1.

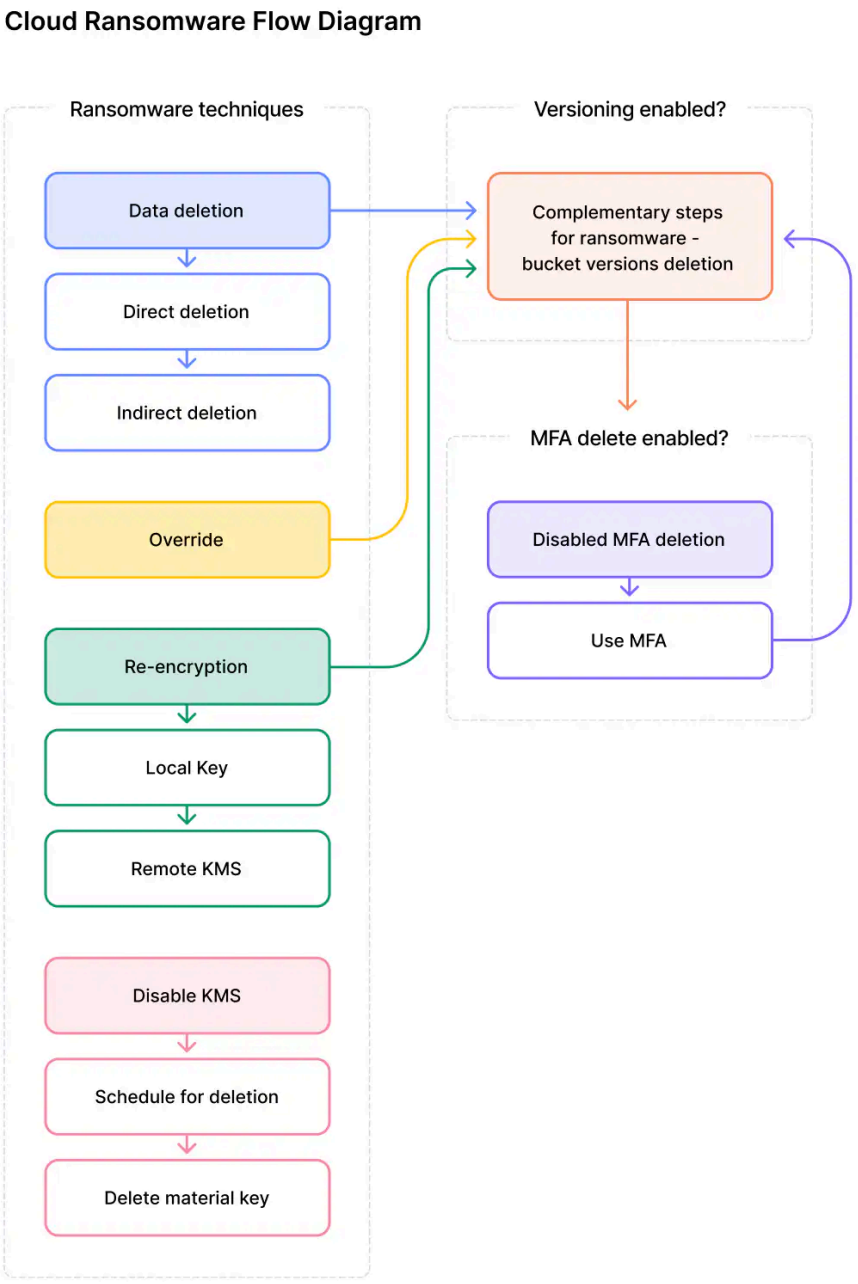

Figure 1: Primary ransomware attack techniques used in the cloud

First, let's dive into each technique and look at the relevant CLI and required permissions, as well as its advantages and limitations from the attacker's perspective.

1. Data Deletion

1a. Direct Data Deletion

During a ransomware attack, the attacker may decide to delete data to gain exclusive control over it. By understanding the mechanisms behind this cyberattack, we can develop more effective counter-strategies.

CLI Command

```
aws s3 rm --recursive s3://your_bucket_name
```

Required Permission

- **'s3:DeleteObject'**

Advantages

- Speed: Deletion of 1,000 objects in 2 seconds
- Doesn't require permission to list objects in the bucket.

Limitation

- Ineffective if versioning is enabled, since only the current version is removed.

1b. Indirect Deletion Using Lifecycle Policy

Alternatively, an attacker can manipulate bucket lifecycle rules by setting them to delete objects automatically after a certain period and by bypassing the need for direct deletion permissions.

CLI Command

```
aws s3api put-bucket-lifecycle-configuration \
--bucket my-bucket \
--lifecycle-configuration '{
"Rules": [
{
"ID": "Delete older versions",
"Status": "Enabled",
"Filter": {
"Prefix": ""
},
"NoncurrentVersionExpiration": {
"NoncurrentDays": 1
```

```
    }
    }
    ]
}'
```

Required Permission

- **‘s3:PutLifecycleConfiguration’**

Advantages

- Stealth: Doesn’t require permissions to access objects directly.
- Helps evade detection, since the attacker doesn’t access the objects themselves.

Limitation

- Time-consuming: Takes at least one day to initiate deletion.

2. Object Override

In this technique, attackers create a blank file and systematically replace every object in the bucket to, in effect, make the data unusable.

CLI Command

```
# Create an empty file locally
touch emptyfile.txt

# Get a list of all objects in the bucket
aws s3 ls s3://ransom-test-with-versioning --recursive | awk '{print $4}' > objects.txt

# Read the file line by line
while IFS= read -r line
do
    # Copy the empty file to each object in the bucket
    aws s3 cp emptyfile.txt s3://ransom-test-with-versioning/$line
done < objects.txt
```

	# Remove the local files
	rm emptyfile.txt objects.txt

Required Permissions

- ‘s3:PutObject’: To overwrite objects within the bucket
- ‘s3:ListBucket’: To list all objects that need to be overwritten

Advantage

- Doesn’t require ‘s3:DeleteObject’.

Limitation

- Ineffective against buckets with enabled versioning, since only the most recent version is modified.
- Requires additional ‘s3:ListBucket’ permission, which is less common.

3. Object Encryption / Re-Encryption

3a. Encrypt / Re-Encrypt Objects with Local Encryption Key

Re-encryption of all objects in the victim’s bucket also involves multiple techniques, including reading every file in the bucket and reuploading it — this time with a local encryption key. In this way, the encryption key is the only way the victim can decrypt the files.

CLI Command

	export encryption_key=`openssl rand -hex 16`
	aws s3 cp s3://my-bucket/ s3://my-bucket/ --recursive --sse-c AES256 --sse-c-key $encryption_key

Once a customer key encrypts the object, the victim will receive the following error when trying to access the encrypted files.

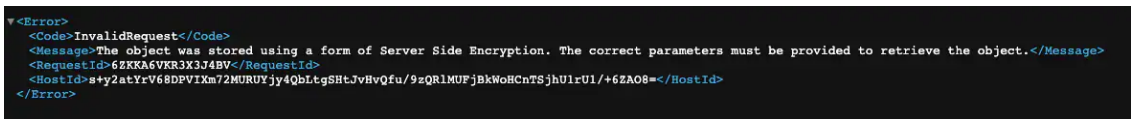

Figure 2: Error message received when attempting to access the encrypted files

Required Permissions

- ‘s3:ListBucket‘: To list and prepare files for re-encryption
- ‘s3:GetObject‘: To retrieve files
- ‘s3:PutObject‘: To upload the re-encrypted files back to the bucket

Advantages

- Speed: Like other methods, re-encrypting a large number of objects can be executed quickly.
- Storage-free attack: The attacker needs to store only the encryption key, not the files.

Limitation

- Ineffective if versioning is enabled, since only the latest version of file is affected by the new encryption.

3b. Encrypt / Re-Encrypt Objects with Remote KMS

Another option is to re-encrypt all the files with a KMS stored in a remote account, which, in this case, is in the attacker's remote account.

CLI Command

```
aws s3 cp s3://my-bucket/ s3://my-bucket/ --recursive --sse aws:kms --sse-kms-key-id <new-key-id>
```

Required Permissions

- ‘s3:ListBucket‘: To list and prepare files for re-encryption
- ‘s3:GetObject‘: To retrieve the files
- ‘s3:PutObject‘: To upload the re-encrypted files back to the bucket
- **‘kms:GenerateDataKey’**

Advantages

- Speed: Deletion of 1,000 objects in 2 seconds
- Storage-free attack: The attacker needs to store only the encryption key, not the files.

Limitation

- Ineffective if versioning is enabled, since only the latest version of files is affected by the new encryption.

4. Disable Key

4a. Schedule Key for Deletion

Data encrypted with KMS can be decrypted only with that KMS. Without the key, the data is inaccessible. Attackers can choose to schedule key deletion for the KMS to make the data inaccessible. It isn’t possible to delete the key immediately, but it must be scheduled at least seven days in advance.

In a prod environment with dozens of keys, it’s possible for such a case to go undetected for 7 days, which will lead to key deletion and data loss.

CLI Command

	aws kms schedule-key-deletion --key-id <new-key-id>

Required Permission

- ‘kms:ScheduleKeyDeletion’

Advantages

- Effective for noncurrent versions if versioning is enabled
- Storage-free attack: The attacker needs to store only the encryption key, not the files.
- Stealth: Doesn’t require permissions to access objects directly.
- Helps evade detection, since the attacker doesn’t access the objects themselves.

Limitation

- Time-consuming: Takes 7 days to complete.

4b. Delete Key Material

A more sophisticated ransomware tactic involves disabling the key material in S3 buckets encrypted with external keys. Unlike AWS-managed keys that have a mandatory delay before deletion, external key material can be deleted immediately, leaving the encrypted data inaccessible.

The attacker’s strategy is to remove access to the encryption key rather than manipulate the data directly.

CLI Command

Required Permission

- **kms:DeleteImportedKeyMaterial**

Advantages

- Data encrypted becomes inaccessible almost instantly as a result of this rapid process
- When deleting key material from versioned buckets, all files encrypted with it are affected
- Stealth: Doesn’t require permissions to access objects directly.
- Helps evade detection, since the attacker doesn’t access the objects themselves.

Limitation

- Dependent on obtaining specific permissions (kms:DeleteImportedKeyMaterial), which isn’t common for the attacker

Complementary Ransomware Steps: Deletion of All Noncurrent Versions

If versioning is enabled on the asset, the attacker needs to make all noncurrent versions unavailable for the victim to complete the attack on the victim’s bucket. The techniques outlined above don't handle the remaining versions.

CLI Command (to delete remaining backups)

```
aws s3api list-object-versions \
--bucket my-bucket \
--output json | jq -r '.Versions[] | .Key + " " + .VersionId' | while read key versionId
do
aws s3api delete-object \
--bucket my-bucket \
--key "$key" \
--version-id "$versionId"
done
```

Required Permissions

- ‘s3:DeleteObjectVersion’: To delete noncurrent version of an object
- ‘s3:ListBucketVersions’: To list all noncurrent versions of an object

Advantage

- None

Limitation

- Time to complete depends on the number of noncurrent objects.

Comparative Analysis of Ransomware Techniques

After exploring the individual ransomware attack scenarios, comparing them across similar conditions helps us understand their relative levels of danger and efficacy.

Before doing that, though, let’s point out two key parameters that help us make a proper comparison.

Dimension of Time

The time dimension is important in ransomware attacks because it defines the window of opportunity to detect and respond to the attack. By measuring the speed of the ransomware techniques in handling varied quantities of data, we can rank their relative strength.

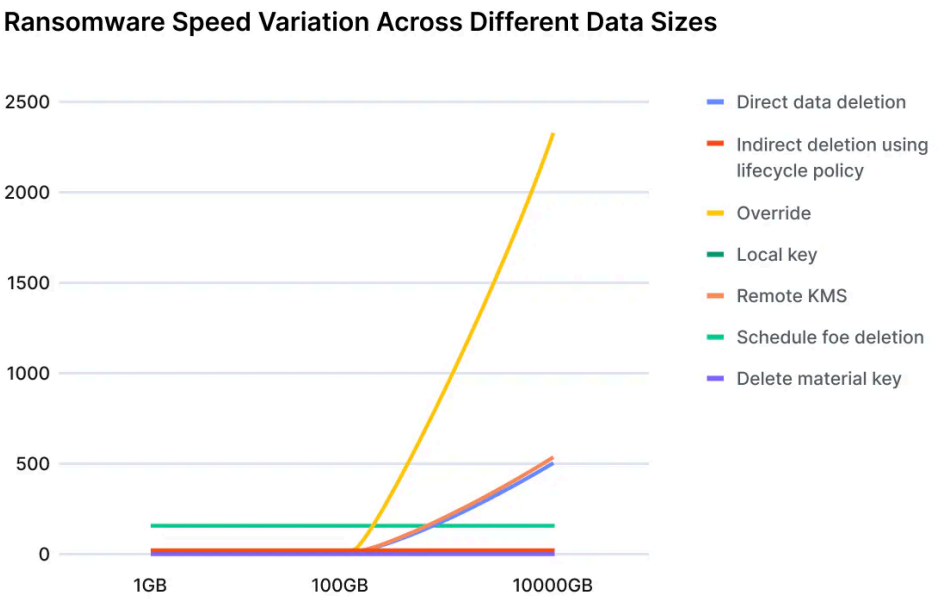

Figure 3: Time dimension of ransomware attack helps identify the attack technique.

The strength of the direct deletion, override, and re-encryption techniques drops as the quantity of data increases. In contrast, techniques that are considered less efficient for small quantities of data, such as schedule for deletion and indirect deletion using life cycle policy, get relatively stronger as the quantity of data increases.

Efficiency of Attack Execution: CLI Vs. Python

Timing is critical in cloud ransomware scenarios. The time required to delete, write or encrypt data can span hours, even days, in the cloud. This window is more than an operational concern. It’s a critical factor regarding the efficacy of attacks and the potential defenses against them. That’s why, when conducting our evaluation, we examined the execution speed of ransomware techniques via CLI and Python scripts on S3 buckets, with and without versioning.

The timing for the various actions are illustrated in the following table.

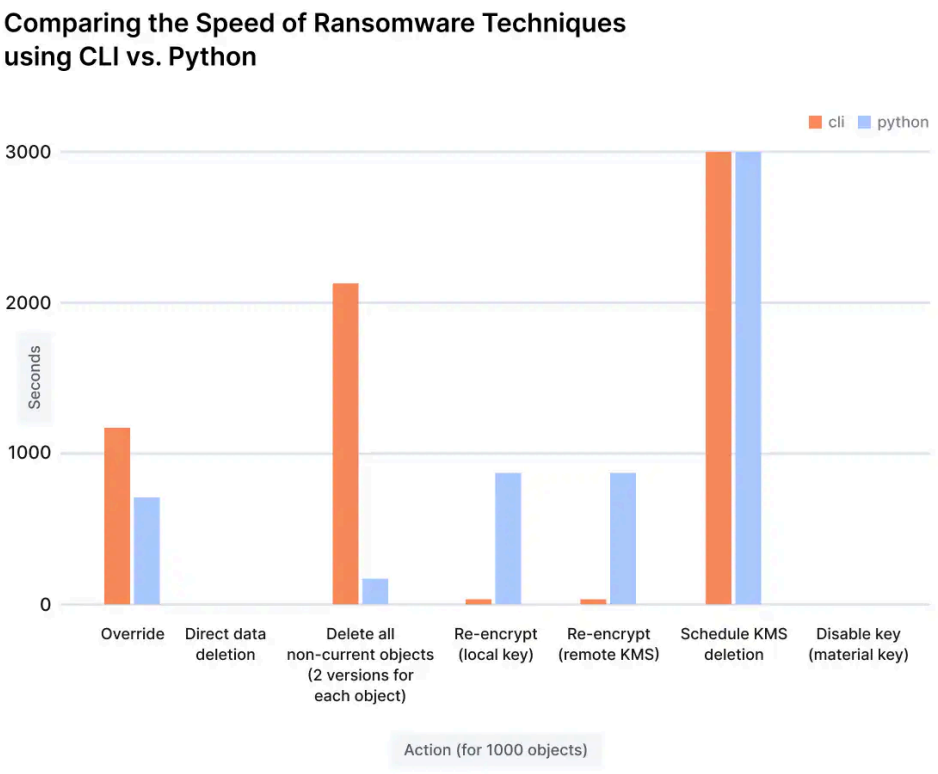

Figure 4: The execution speed of ransomware techniques via CLI and Python scripts on S3 buckets

Our findings again highlight the need for security teams to reduce the mean time to detect (MTTD) and mean time to respond (MTTR) for attacks on their cloud infrastructure.

At-a-Glance Comparison

The table below summarizes our findings based on a comparison of the speed, impact and reversibility of each method. It not only highlights the most risky techniques, but also underscores the importance of timely detection and response strategies in mitigating potential damage.

	Direct data deletion	Indirect deletion using life cycle policy	Override	Re-encrypt (local key)	Disable key (material key)
Speed of execution for 10TB	Moderate	Low	High	Moderate	Low
Required permissions	s3:DeleteObject	s3:PutLifecycleConfiguration	s3:PutObject, s3:ListBucket	s3:ListBucket, s3:GetObject, s3:PutObject	kms:DeleteImportedKeyMaterial
Impact on data	Total loss	Total loss	Data becomes unusable	Encrypted	Encrypted
Reversibility	requires versioning	Irreversible	Requires versioning	Requires key	Irreversible
Effectiveness against versioning	Ineffective	Effective	Ineffective	Ineffective	Effective
Detection difficulty (stealth)	Easy (due to mass deletion)	High	Moderate (multiple writes are not noticeable)	Moderate (multiple writes are not noticeable)	Low
Complexity	Low	High	Moderate	High	High
Popularity	High	Low	Moderate	Low	Low

When examining the table, it becomes evident that quick and irreversible techniques pose the greatest risk. Some techniques are stealth and affect noncurrent versions, while others aren't and don't. Every technique requires reading the data. As a result, encryption at rest is one of the best protection mechanisms \against ransomware in the cloud.

Protecting Against Ransomware

While ransomware can devastate, you can take actions to neutralize the techniques and reduce the attack's effectiveness.

1. Enable versioning to ensure that when one object is re-encrypted with a new KMS, the older version encrypted with the old KMS won't be lost forever. This will also increase the time it takes to complete the attack.
2. Enable MFA deletion when enabling versioning on the bucket. When a bucket has MFA delete enabled in its versioning configuration, the bucket owner is required to include the x-amz-mfa request header in any request to delete a version of an object permanently or modify the bucket's versioning state.

3. Use delete protection. As a secondary protection mechanism, we can use the S3 Object Lock on top of S3 versioning to prevent data from being deleted or overwritten.
4. Use KMS with key material managed by AWS rather than that managed externally or located in a key store like CloudHSM. KMS managed by AWS can't be deleted immediately. It requires scheduling 7 days in advance, which gives you enough time to respond to an attack.
5. Use KMS with access policy to allow specific principals to read the data. Attackers who can't read the data can't carry out a ransomware attack. Put appropriate permissions in the policy as a second layer of security to access the objects in the S3 bucket.
6. Remove powerful permissions on S3 buckets like 's3:PutLifecycleConfiguration,' 's3:PutEncryptionConfiguration,' and 's3:DeleteObjectVersion' to reduce the attack surface.
7. Remove powerful permissions on KMS like 'kms:DeleteImportedKeyMaterial' and 'kms:DeleteCustomKeyStore' to reduce the attack surface.

Enforce separation of duties. Our 2023 State of Data Security Research shows that all principals with admin permissions also have consumer privileges. Use the divide-and-conquer approach to separate the permissions required to carry out a full attack by one principal.

The Stats: How Vulnerable Are You?

Encryption

- 10% of encrypted buckets utilize CMK for enhanced security.
- 4% of the 10% above are remotely managed.
- 72% of remote CMK buckets aren't actively monitored.

Data Protection Measures

- 31% of buckets have versioning enabled, which is essential for data recovery.
- 67% of sensitive buckets have logging enabled.
- 1% of the buckets have object lock, which is crucial for preventing data tampering.

These figures reveal significant gaps in current data security practices, highlighting areas where immediate improvements are necessary to bolster defenses against ransomware attacks.

Early Detection and Response Tactics

Knowing how much time it takes to perform a ransomware attack on your data is the first step to choosing a prevention and detection strategy. But you'll also need to enable logging of data events on your S3 buckets.

Scenario 1: Versioning and MFA Deletion Are Disabled

In a bucket whose versioning and MFA deletion aren't turned on, the first sign of a ransomware attack will be multiple attempts to read / write files from the bucket. The detection requires critical mass before it's deemed as an attack, and by nature, detection will raise an alert only after the attack has begun and the data has been exfiltrated or deleted.

Scenario 2: Versioning and MFA Deletion are Enabled

If the bucket is protected with versioning and MFA delete, preliminary signs of an attack could include either removing the configuration of MFA deletion, versioning or changing bucket encryption configuration. These signs help us become aware of the attack before it impacts the data. And as long as it's advancing, we'll receive additional signs that something is happening that requires our attention.

As previously shown, the presence of noncurrent versions in a versioned S3 bucket adds complexity to a ransomware attack. This means that attackers have to navigate through additional steps to compromise the data, which in turn provides you with a longer time frame to detect and counteract the attack before any significant data loss occurs.

Learn More

For insights into cloud security and a better understanding of how your data is exposed in the cloud, read our comprehensive State of Cloud Data Security 2023 report. This research sheds light on crucial aspects of cloud data security and provides actionable steps to defend your valuable data effectively.

If you haven't tried Prisma Cloud and would like to test drive best-in-class Code to Cloud™ security, we'd love for you to experience a free 30-day trial.
