Proof of Concept (PoC) of Cloud Storage Threat Matrix (Microsoft)

Name of Presenter: Akkshat Shah
Intern ID: 261
Presented to: Digisuraksha Parhari Foundation

What is Threat Intelligence?

Threat Intelligence refers to the systematic collection, evaluation, and application of information related to existing and emerging cyber threats. It helps organizations proactively **identify**, **understand**, and **respond** to potential cyberattacks. By offering insights into attacker profiles, tools, techniques, and motives, threat intelligence empowers security teams to make better, data-driven decisions to safeguard critical infrastructure and digital assets.

In simple terms:

- **Tactic** → The *purpose* behind an attacker's action (the "why")
- **Technique** → The *approach* used to carry out that purpose (the "how")
- **Sub-technique** → A more detailed method under a broader technique
- **Procedure** → A specific real-world instance showing how the technique was executed

---

What is the Cloud Storage Threat Matrix?

The **Cloud Storage Threat Matrix** is a security framework introduced by Microsoft that highlights how adversaries can target cloud storage systems (such as **AWS S3**, **Azure Blob Storage**, or **Google Cloud Storage**) using techniques mapped from the **MITRE ATT&CK** framework.

This matrix breaks down an attacker's behavior into structured stages — called **Tactics** — and connects them with real-world techniques and procedures. It enables defenders to understand, detect, and respond to threats specific to cloud file systems.

---

1. Tactic: Reconnaissance

Description: The attacker gathers information about the cloud environment, services, and exposed endpoints before launching an attack.

Technique 1: T1595 – Active Scanning

Description: Scanning for publicly exposed cloud storage buckets.

Procedure 1

- **Objective**: Identify open or misconfigured buckets.
- **Steps**:
    1. Use tools like `s3scanner` or `grayhat warfare`.
    2. Input a list of possible bucket names.
    3. Scan and collect accessible URLs.
- **Outcome**: Lists of buckets accessible without authentication.

Procedure 2

- **Objective**: Find indexed files using search engines.
- **Steps**:

    1. Use Google Dorking like: `site:s3.amazonaws.com filetype:pdf`.
    2. Export results using custom scripts.
- **Outcome**: Reveals publicly listed files in misconfigured buckets.

Technique 2: T1589 – Gather Victim Identity Information

Description: Collect names, emails, and job titles of employees using OSINT.

Procedure 1

- **Objective**: Target storage admins or devops engineers.
- **Steps**:
    1. Use LinkedIn, GitHub, or Hunter.io to find targets.
    2. Extract organization and role-based contacts.
- **Outcome**: Builds a high-value target list.

Procedure 2

- **Objective**: Look for credentials in public repos.
- **Steps**:
    1. Search `.env` files on GitHub with secrets.
    2. Filter by cloud-related keys.
- **Outcome**: Valid credentials discovered for cloud accounts.

Technique 3: T1538 – Cloud Service Discovery

Description: Identify storage types, endpoints, or platforms used.

Procedure 1

- **Objective**: Determine if organization uses AWS, Azure, or GCP.
- **Steps**:
    1. Monitor subdomains and TLS certificates.
    2. Check DNS records and CNAME mappings.
- **Outcome**: Confirms cloud provider.

Procedure 2

- **Objective**: Discover storage resource paths.
- **Steps**:
    1. Look for naming patterns like `*.blob.core.windows.net`.
    2. Use Shodan to scan IPs or ports.
- **Outcome**: Identifies accessible storage endpoints.

---

2. Tactic: Initial Access

Description: Attacker attempts to gain unauthorized entry to the cloud storage environment.

Technique 1: T1078 – Valid Accounts

Description: Use leaked or stolen credentials to access storage systems.

Procedure 1

- **Objective**: Exploit exposed `.env` files with AWS keys.
- **Steps**:
    1. Search GitHub using `filename:.env AWS_SECRET_ACCESS_KEY`.
    2. Test keys using AWS CLI or SDK.
- **Outcome**: Direct access to cloud storage.

Procedure 2

- **Objective**: Use passwords found in data breaches.
- **Steps**:
    1. Search sites like Pastebin, HaveIBeenPwned.
    2. Attempt login to cloud consoles.
- **Outcome**: Unauthorized login to storage.

Technique 2: T1133 – External Remote Services

Description: Access storage using remote management interfaces.

Procedure 1

- **Objective**: Use legitimate tools to access files.
- **Steps**:
    1. Install AWS S3 Browser / Azure Storage Explorer.
    2. Input compromised keys or tokens.
- **Outcome**: Full read/write access to cloud files.

Procedure 2

- **Objective**: Abuse federated logins (SSO).
- **Steps**:
    1. Intercept tokens during OAuth login.
    2. Replay or reuse token in API calls.
- **Outcome**: Remote access without password.

Technique 3: T1190 – Exploit Public-Facing Applications

Description: Exploit web apps that interface with cloud storage.

Procedure 1

- **Objective**: Exploit vulnerable upload endpoint.
- **Steps**:
    1. Find apps that accept file uploads.
    2. Upload script with bypassed file checks.

- **Outcome**: Malicious file reaches cloud storage.

Procedure 2

- **Objective**: Abuse exposed APIs.
- **Steps**:
  1. Use Postman or curl to access API.
  2. Upload, modify, or delete cloud files.
- **Outcome**: Full unauthorized interaction with storage.

---

3. Tactic: Defense Evasion

Description: Techniques to avoid detection or logging while using or modifying cloud storage.

Technique 1: T1027 – Obfuscated Files or Information

Description: Rename malicious files to bypass filters.

Procedure 1

- **Objective**: Hide executable as image.
- **Steps**:
  1. Rename `backdoor.exe` to `invoice.png`.
  2. Upload to public S3 bucket.
- **Outcome**: File looks safe but executes maliciously.

Procedure 2

- **Objective**: Split payload into chunks.

- **Steps**:
    1. Divide ZIP or base64 payload into parts.
    2. Upload separately and rejoin later.
- **Outcome**: Obfuscates payload from detection.

Technique 2: T1070.004 – File Deletion

Description: Delete logs or temporary files.

Procedure 1

- **Objective**: Erase log files after access.
- **Steps**:
    1. Use AWS CLI: `aws s3 rm s3://bucket/logs/ --recursive`.
    2. Confirm deletion with `list` command.
- **Outcome**: No evidence left.

Procedure 2

- **Objective**: Tamper with log retention policy.
- **Steps**:
    1. Change storage policy to 1-day expiry.
    2. Force cleanup before alerting.
- **Outcome**: Logs auto-deleted.

Technique 3: T1562.001 – Disable or Modify Tools

Description: Alter native cloud logging or monitoring.

Procedure 1

- **Objective**: Disable Azure diagnostics logs.
- **Steps**:
    1. Navigate to diagnostics settings.
    2. Toggle off blob log capture.
- **Outcome**: Storage activity no longer logged.

Procedure 2

- **Objective**: Remove S3 bucket policy audit.
- **Steps**:
    1. Edit bucket policy to allow anonymous access.
    2. Prevent updates from triggering alerts.
- **Outcome**: Security bypassed silently.