☁️ Threat Intelligence PoC – Cloud Storage Threat Matrix (Microsoft)

Intern Name: Anoop Shivadas
Intern ID: 129
Organization: DigiSuraksha Foundation
Project Title: Cloud Storage Threat Matrix (Microsoft)

🎯 Objective

The purpose of this Proof of Concept (PoC) is to study how attackers exploit cloud storage services by mapping their behavior to **MITRE ATT&CK-style tactics, techniques, and procedures (TTPs)**. Using Microsoft's Cloud Storage Threat Matrix, this project explains real-world attack paths and provides defensive strategies to strengthen security.

🧭 Part 1: Tactics

Tactics represent the **adversary's overall goals** when targeting cloud storage. The following tactics are relevant:

- **Reconnaissance** → Attackers search for exposed or misconfigured buckets.
- **Initial Access** → Entry via misconfigurations or exposed tokens.
- **Execution** → Running malicious scripts through storage-integrated services.
- **Persistence** → Maintaining long-term access in storage.
- **Privilege Escalation** → Gaining higher permissions by abusing leaked keys.
- **Defense Evasion** → Hiding activity using temporary tokens.
- **Credential Access** → Harvesting secrets from uploaded files.
- **Discovery** → Mapping out accounts, containers, and policies.
- **Lateral Movement** → Using storage as a pivot to other services.
- **Collection** → Gathering sensitive files.
- **Exfiltration** → Exporting staged data.
- **Impact** → Deleting/encrypting data for disruption.

🧩 Part 2: Techniques

🌐 Technique 1: Unsecured Cloud Storage Buckets

- **Tactics:** Reconnaissance, Initial Access
- **Description:** Buckets configured with public access allow attackers to read/download data without authentication. Using scanners or brute-forced names, attackers may uncover backups, source code, or customer data.
- **Real Context:** Data leaks at Tesla and Accenture occurred due to public buckets.
- **Defense:** Default private access, posture management tools, and logging.

🔑 Technique 2: Exploiting Shared Access Tokens (SAS / Pre-Signed URLs)

- **Tactics:** Defense Evasion, Collection
- **Description:** Time-bound URLs (SAS, pre-signed links) can bypass normal IAM if leaked. Attackers silently download/upload data with little trace.
- **Real Context:** Developers committing SAS tokens in GitHub or Slack.
- **Defense:** Use short expiry, restrict IPs, monitor logs, prefer user-delegated tokens.

🦹 Technique 3: Cloud Credential Harvesting

- **Tactics:** Credential Access, Privilege Escalation
- **Description:** Attackers search storage for .env files, API keys, OAuth tokens, or SSH keys. Harvested secrets allow escalation into databases, CI/CD, or cloud accounts.
- **Real Context:** Groups like LAPSUS$ exploited hardcoded cloud credentials.
- **Defense:** Use secret managers (AWS Secrets Manager, Azure Key Vault), enable DLP/secret scanning.

🗺️ Technique 4: Enumeration of Storage Accounts

- **Tactic:** Discovery
- **Description:** After entry, adversaries enumerate containers, IAM policies, and naming conventions to identify weak links.
- **Tools:** Azure CLI (az storage account list), AWS CLI, MicroBurst scripts.
- **Defense:** Tight RBAC, monitor excessive list/enumeration activity.

💣 Technique 5: Data Destruction / Ransomware

- **Tactic:** Impact
- **Description:** Attackers may delete/encrypt storage data directly through APIs, leading to outages.
- **Real Context:** Ransomware variants have targeted S3 buckets.
- **Defense:** Enable soft-delete, versioning, WORM policies, and regular restore testing.

⚙️ Part 3: Procedures

✅ Procedure 1: Discovering & Accessing Misconfigured Storage

Goal: Simulate how attackers find public buckets.
Steps:

1. Open CLI or PowerShell.
2. Run a scanner (e.g., MicroBurst, AZScanner) with a wordlist.
3. Identify open containers.
4. Use az storage blob list --account-name <acct> --container-name <container> to view contents.
   Result: If public, files can be listed/downloaded without login.

✅ Procedure 2: Exploiting a Leaked SAS Token

Goal: Show how exposed tokens grant unauthorized access.
Steps:

1. A developer commits a SAS token URL:
2. https://myblob.blob.core.windows.net/container/file.txt?<sas-token>
3. Attacker copies URL and runs:
4. curl -O "https://myblob.blob.core.windows.net/container/file.txt?<sas-token>"
5. File is downloaded without authentication.
   Result: Sensitive files accessed using token abuse.

📊 Mapping of Tactics → Techniques → Procedures

Tactic	Technique	Procedure
Reconnaissance	Unsecured Buckets	Public container discovery
Initial Access	Unsecured Buckets	Anonymous object listing
Defense Evasion	Leaked SAS Tokens	Accessing blob via token
Credential Access	Credential Harvesting	Searching for .env / API keys
Discovery	Enumeration of Storage	CLI listing / policy review
Impact	Data Destruction	Delete/encrypt + restore testing

🛡️ Suggested Mitigations

Threat	Mitigation
Public bucket access	Enforce private by default, CSPM scanning
SAS token misuse	Short expiry, IP restrictions, HTTPS only
Credential leaks	Use Key Vault/Secret Manager, DLP scanning
Enumeration abuse	Monitor excessive list requests, tight RBAC
Data destruction	Enable versioning, immutability, backup tests

---

📚 References

- MITRE ATT&CK Framework → https://attack.mitre.org/
- Microsoft Cloud Storage Threat Matrix → Microsoft Security Blog
- Tools: MicroBurst, AZScanner

✅ Conclusion

This PoC demonstrates how cloud storage can be exploited through **misconfiguration, token abuse, secret leaks, and destructive actions**. By following the structured TTP format, defenders can better anticipate threats and deploy effective mitigations. The use of Microsoft's Cloud Storage Threat Matrix ensures that this research is aligned with industry frameworks.