Storm-0501's evolving techniques lead to cloud-based ransomware | Microsoft Security Blog

By Microsoft Threat Intelligence

Published: 2025-08-27 · Archived: 2026-04-05 12:47:14 UTC

Microsoft Threat Intelligence has observed financially motivated threat actor Storm-0501 continuously evolving their campaigns to achieve sharpened focus on cloud-based tactics, techniques, and procedures (TTPs). While the threat actor has been known for targeting hybrid cloud environments, their primary objective has shifted from deploying on-premises endpoint ransomware to using cloud-based ransomware tactics.

Unlike traditional on-premises ransomware, where the threat actor typically deploys malware to encrypt critical files across endpoints within the compromised network and then negotiates for a decryption key, cloud-based ransomware introduces a fundamental shift. Leveraging cloud-native capabilities, Storm-0501 rapidly exfiltrates large volumes of data, destroys data and backups within the victim environment, and demands ransom—all without relying on traditional malware deployment.

Storm-0501's targeting is opportunistic. The threat actor initially deployed Sabbath ransomware in an attack against United States school districts in 2021. In November 2023, the actor targeted the healthcare sector. Over the years, the actor switched ransomware payloads multiple times, using Embargo ransomware in 2024 attacks.

In September 2024, we published a blog detailing how Storm-0501 extended its on-premises ransomware operations into hybrid cloud environments. The threat actor gained a foothold by compromising Active Directory environments and then pivoted to Microsoft Entra ID, escalating privileges on hybrid and cloud identities to gain global administrator privileges. The impact phase of these attacks took one of two forms: implanting backdoors in Entra ID tenant configurations using maliciously added federated domains to allow sign-in as nearly any user or deploying on-premises ransomware to encrypt endpoints and servers, eventually demanding ransom for the decryption keys.

Storm-0501 has continued to demonstrate proficiency in moving between on-premises and cloud environments, exemplifying how threat actors adapt as hybrid cloud adoption grows. They hunt for unmanaged devices and security gaps in hybrid cloud environments to evade detection and escalate cloud privileges and, in some cases, traverse tenants in multi-tenant setups to achieve their goals.

In this blog post, we describe the impact of a recent Storm-0501 attack on a compromised cloud environment. We trace how the threat actor achieved cloud-based ransomware impact through cloud privilege escalation, taking advantage of protection and visibility gaps across the compromised environment, and pivoting from on-premises to cloud pivots. Understanding how such attacks are conducted is critical in protecting cloud environments. Below we share protection and mitigation recommendations, including strengthening protections for cloud identities and cloud resources, and detection guidance across Microsoft security solutions to help organizations harden their networks against these attacks.

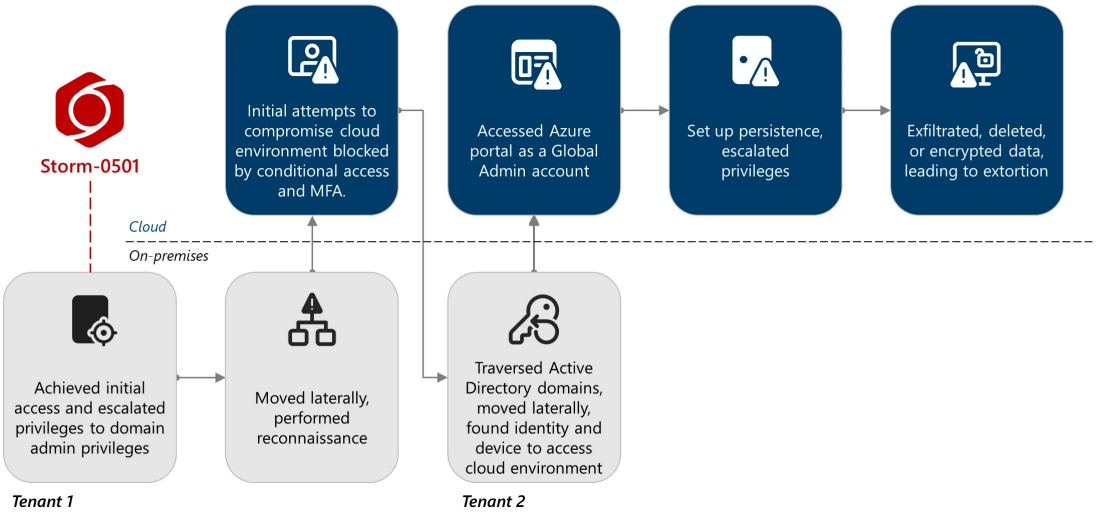

Figure 1. Overview of Storm-0501 cloud-based ransomware attack chain

On-premises compromise and pivot to the cloud

In a recent campaign, Storm-0501 compromised a large enterprise composed of multiple subsidiaries, each operating its own Active Directory domain. These domains are interconnected through domain trust relationships, enabling cross-domain authentication and resource access.

The cloud environment mirrors this complexity. Different subsidiaries maintain separate Microsoft Azure tenants, with varying Microsoft Defender product coverage. Notably, only one tenant had Microsoft Defender for Endpoint deployed, and devices from multiple Active Directory domains were onboarded to this single tenant's license. This fragmented deployment created visibility gaps across the environment.

Active Directory domains were synchronized to several Entra ID tenants using Entra Connect Sync servers. In some cases, a single domain was synced to more than one tenant, further complicating identity management and monitoring. For clarity, this blog focuses on the two tenants impacted by the attack: one where on-premises activity was observed, and another where cloud-based activity occurred.

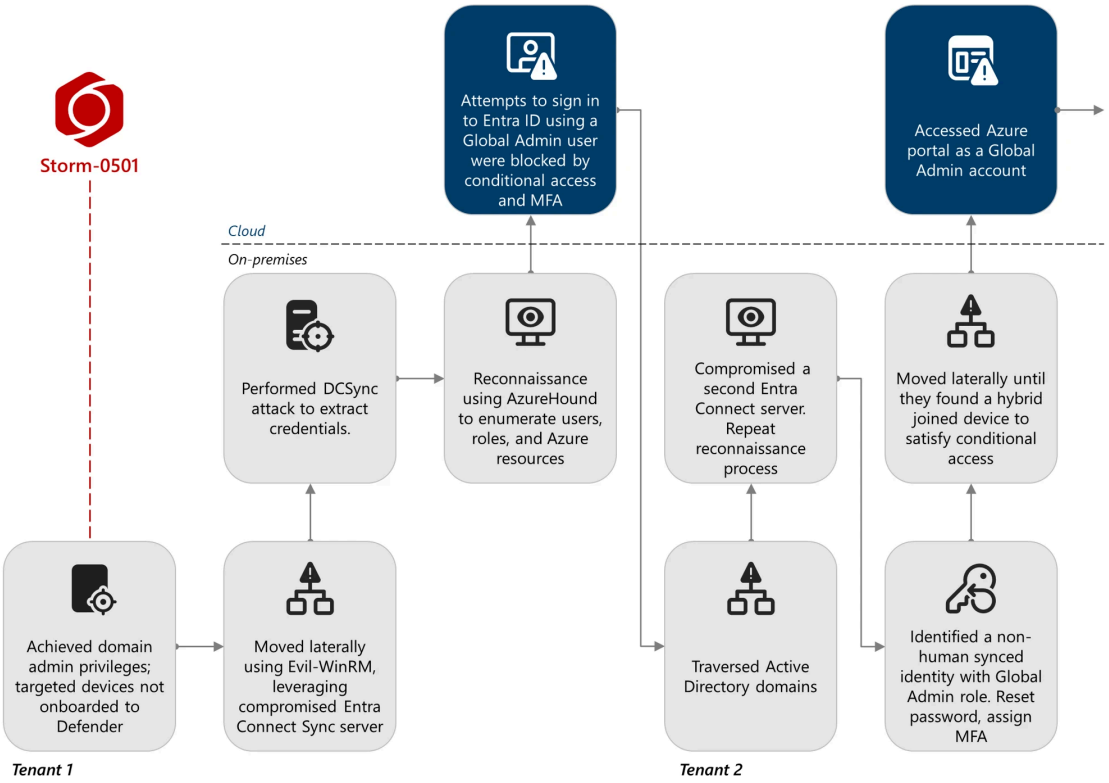

Figure 2. Storm-0501 on-premises attack chain

On-premises activity

For the purposes of this blog, we focus our analysis on the post-compromise phase of the on-premises attack, meaning that the threat actor had already achieved domain administrator privileges in the targeted domain. Read our previous blog for a more comprehensive overview of Storm-0501 tactics in on-premises environments.

The limited deployment of Microsoft Defender for Endpoint across the environment significantly hindered detection. Of the multiple compromised domains, only one domain had significant Defender for Endpoint deployment, leaving portions of the network unmonitored. On the few onboarded devices where Storm-0501 activity was observed, we noted that the threat actor conducted reconnaissance before executing malicious actions. Specifically, the threat actor used the following commands:

sc query sense
sc query windefend

The threat actor checked for the presence of Defender for Endpoint services, suggesting a deliberate effort to avoid detection by targeting non-onboarded systems. This highlights the importance of comprehensive endpoint coverage.

Lateral movement was facilitated using Evil-WinRM, a post-exploitation tool that utilizes PowerShell over Windows Remote Management (WinRM) for remote code execution. The abovementioned commands were executed over sessions initiated with the tool, as well as discovery using other common native Windows tools and commands such as *quser.exe* and *net.exe*. Earlier in the attack, the threat actor had compromised an Entra Connect

Sync server that was not onboarded to Defender for Endpoint. We assess that this server served as a pivot point, with the threat actor establishing a tunnel to move laterally within the network.

The threat actor also performed a DCSync attack, a technique that abuses the Directory Replication Service (DRS) Remote Protocol to simulate the behavior of a domain controller. By impersonating a domain controller, the threat actor could request password hashes for any user in the domain, including privileged accounts. This technique is often used to extract credentials without triggering traditional authentication-based alerts.

Pivot to the cloud

Following the on-premises compromise of the first tenant, the threat actor leveraged the Entra Connect Sync Directory Synchronization Account (DSA) to enumerate users, roles, and Azure resources within the tenant. This reconnaissance was performed using AzureHound, a tool designed to map relationships and permissions in Azure environments and consequently find potential attack paths and escalations.

Shortly thereafter, the threat actor attempted to sign in as several privileged users. These attempts were unsuccessful, blocked by Conditional Access policies and multifactor authentication (MFA) requirements. This suggests that while Storm-0501 had valid credentials, they lacked the necessary second factor or were unable to satisfy policy conditions.

Undeterred, Storm-0501 shifted tactics. Leveraging their foothold in the Active Directory environment, they traversed between Active Directory domains and eventually moved laterally to compromise a second Entra Connect server associated with different Entra ID tenant and Active Directory domain. The threat actor extracted the Directory Synchronization Account to repeat the reconnaissance process, this time targeting identities and resources in the second tenant.

Identity escalation

As a result of the discovery phase where the threat actor leveraged on-premises control to pivot across Active Directory domains and vastly enumerate cloud resources, they gained critical visibility of the organization’s security posture. They then identified a non-human synced identity that was assigned with the Global Administrator role in Microsoft Entra ID on that tenant. Additionally, this account lacked any registered MFA method. This enabled the threat actor to reset the user’s on-premises password, which shortly after was then legitimately synced to the cloud identity of that user using the Entra Connect Sync service. We identified that that password change was conducted by the Entra Connect’s Directory Synchronization Account (DSA), since the Entra Connect Sync service was configured on the most common mode Password-Hash Synchronization (PHS). Consequently, the threat actor was able to authenticate against Entra ID as that user using the new password.

Since no MFA was registered to that user, after successfully authenticating using the newly assigned password, the threat actor was redirected to simply register a new MFA method under their control. From then on, the compromised user had a registered MFA method that enabled the threat actor to meet MFA conditions and comply with the customer’s Conditional Access policies configuration per resource.

To access the Azure portal using the compromised Global Admin account, the threat actor had to bypass one more condition that was enforced by Conditional Access policies for that resource, which require authentication to occur

from a Microsoft Entra hybrid joined device. Hybrid joined devices are devices that are joined to both the Active Directory domain and Entra ID. We observed failed authentication attempts coming from company devices that are either domain-joined or Entra-joined devices that did not meet the Conditional Access condition. The threat actor had to move laterally between different devices in the network, until we observed a successful sign-in to the Azure portal with the Global Admin account coming from a server that was hybrid joined.

From the point that the threat actor was able to successfully meet the Conditional Access policies and sign in to the Azure portal as a Global Admin account, Storm-0501 essentially achieved full control over the cloud domain. The threat actor then utilized the highest possible cloud privileges to obtain their goals in the cloud.

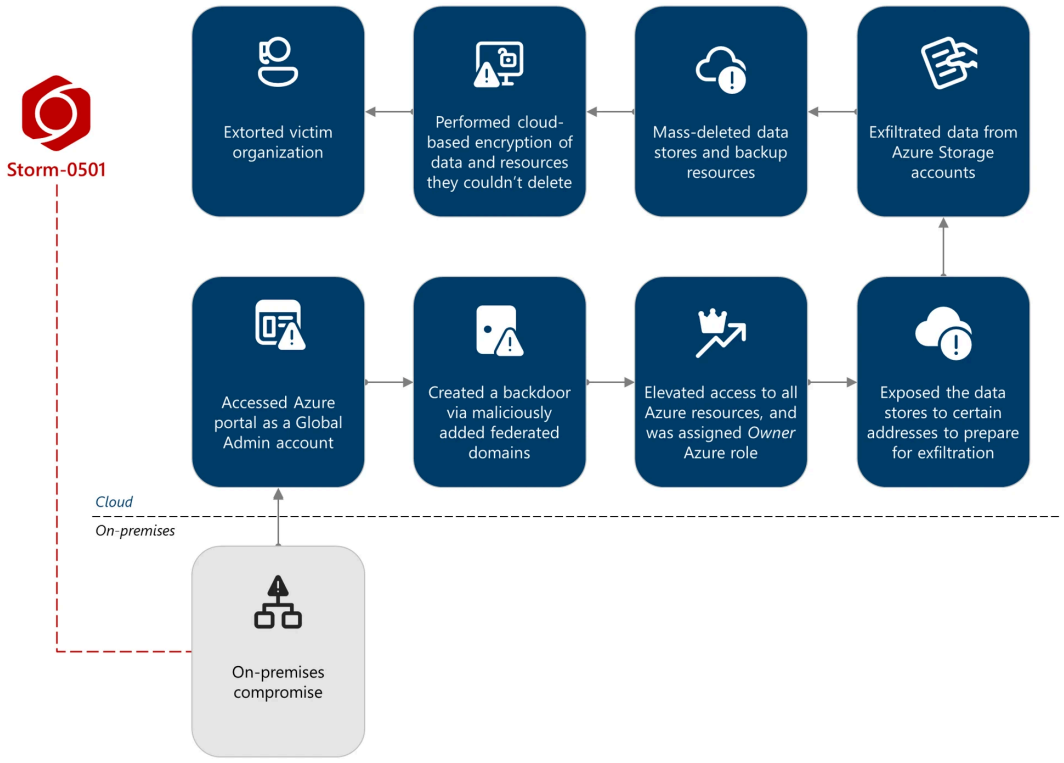

Figure 3. Storm-0501 cloud identity and cloud environment compromise leading to extortion

Cloud identity compromise: Entra ID

Cloud persistence

Following successful authentication as a Global Admin to the tenant, Storm-0501 immediately established a persistence mechanism. As was seen in the threat actor's previous activity, Storm-0501 created a backdoor using a maliciously added federated domain, enabling them to sign in as almost any user, according to the *ImmutableId* user property. The threat actor leveraged the Global Administrator Entra role privileges and the AADInternals tool to register a threat actor-owned Entra ID tenant as a trusted federated domain by the targeted tenant. To establish trust between the two tenants, a threat actor-generated root certificate is provided to the victim tenant, which in turn is used to allow authentication requests coming from the threat actor-owned tenant. The backdoor enabled Storm-0501 to craft security assertion markup language (SAML) tokens applicable to the victim tenant, impersonating users in the victim tenant while assuming the impersonated user's Microsoft Entra roles.

Cloud compromise: Azure

Azure initial access and privilege escalation

A tenant's Entra ID and Azure environments are intertwined. And since Storm-0501 gained top-level Entra ID privileges, they could proceed to their final goal, which was to use cloud-based ransomware tactics for monetary gain. To achieve this goal, they had to find the organization's valuable data stores, and these were residing in the cloud: in Azure.

Because they had compromised a user with the Microsoft Entra Global Administrator role, the only operation they had to do to infiltrate the Azure environment was to elevate their access to Azure resources. They elevated their access to Azure resources by invoking the *Microsoft.Authorization/elevateAccess/action* operation. By doing so, they gained the User Access Administrator Azure role over all the organization's Azure subscriptions, including all the valuable data residing inside them.

To freely operate within the environment, the threat actor assigned themselves the *Owner* Azure role over all the Azure subscriptions available by invoking the *Microsoft.Authorization/roleAssignments/write* operation.

Discovery

After taking control over the organization's Azure environment, we assess that the threat actor initiated a comprehensive discovery phase using various techniques, including the usage of the AzureHound tool, where they attempted to locate the organization's critical assets, including data stores that contained sensitive information, and data store resources that are meant to back up on-premises and cloud endpoint devices. The threat actor managed to map out the Azure environment, including the understanding of existing environment protections, such as Azure policies, resource locks, Azure Storage immutability policies, and more.

Defense evasion

The threat actor then targeted the organization's Azure Storage accounts. Using the public access features in Azure Storage, Storm-0501 exposed non-remotely accessible accounts to the internet and to their own infrastructure, paving the way for data exfiltration phase. They did this by utilizing the public access features in Azure Storage. To modify the Azure Storage account resources, the threat actor abused the Azure *Microsoft.Storage/storageAccounts/write* operation.

Credential access

For Azure Storage accounts that have key access enabled, the threat actor abused their Azure *Owner* role to access and steal the access keys for them by abusing the Azure *Microsoft.Storage/storageAccounts/listkeys/action* operation.

Exfiltration

After exposing the Azure Storage accounts, the threat actor exfiltrated the data in these accounts to their own infrastructure by abusing the *AzCopy Command-line tool (CLI)*.

Impact

In on-premises ransomware, the threat actor typically deploys malware that encrypts crucial files on as many endpoints as possible, then negotiates with the victim for the decryption key. In cloud-based ransomware attacks, cloud features and capabilities give the threat actor the capability to quickly exfiltrate and transmit large amounts of data from the victim environment to their own infrastructure, destroy the data and backup cloud resources in the victim cloud environment, and then demand the ransom.

After completing the exfiltration phase, Storm-0501 initiated the mass-deletion of the Azure resources containing the victim organization data, preventing the victim from taking remediation and mitigation action by restoring the data. They do so by abusing the following Azure operations against multiple Azure resource providers:

- *Microsoft.Compute/snapshots/delete* – Deletes Azure Snapshot, a read-only, point-in-time copy of an Azure VM's disk (VHD), capturing its state and data at a specific moment, that exists independently from the source disk and can be used as a backup or clone of that disk.
- *Microsoft.Compute/restorePointCollections/delete* – Deletes the Azure VM Restore Point, which stores virtual machines (VM) configuration and point-in-time application-consistent snapshots of all the managed disks attached to the VM.
- *Microsoft.Storage/storageAccounts/delete* – Deletes the Azure storage account, which contains and organization's Azure Storage data objects: blobs, files, queues, and tables. In all of Storm-0501 Azure campaigns we investigated, this is where they mainly focused, deleting as many Azure Storage account resources as possible in the environment.
- *Microsoft.RecoveryServices/Vaults/backupFabrics/protectionContainers/delete* – Deletes an Azure recovery services vault protection container. A protection container is a logical grouping of resources (like VMs or workloads) that can be backed up together, within the Recovery Services vault.

During the threat actor's attempts to mass-delete the data-stores/housing resources, they faced errors and failed to delete some of the resources due to the existing protections in the environment. These protections include Azure resource locks and Azure Storage immutability policies. They then attempted to delete these protections using the following operations:

- *Microsoft.Authorization/locks/delete* – Deletes Azure resource locks, which are used to prevent accidental user deletion and modification of Azure subscriptions, resource groups, or resources. The lock overrides any user permission.
- *Microsoft.Storage/storageAccounts/blobServices/containers/immutabilityPolicies/delete* – Deletes Azure storage immutability policies, which protect blob data from being overwritten or deleted.

After successfully deleting multiple Azure resource locks and Azure Storage immutability policies, the threat actor continued the mass deletion of the Azure data stores, successfully erasing resources in various Azure subscriptions. For resources that remained protected by immutability policies, the actor resorted to cloud-based encryption.

To perform cloud-based encryption, Storm-0501 created a new Azure Key Vault and a new Customer-managed key inside the Key Vault, which is meant to be used to encrypt the left Azure Storage accounts using the Azure Encryption scopes feature:

- *Microsoft.KeyVault/vaults/write* – Creates or modifies an existing Azure Key Vault. The threat actor creates a new Azure key vault to host the encryption key.
- *Microsoft.Storage/storageAccounts/encryptionScopes/write* – Creates or modifies Azure storage encryption scopes, which manage encryption with a key that is scoped to a container or an individual blob. When you define an encryption scope, you can specify whether the scope is protected with a Microsoft-managed key or with a customer-managed key that is stored in Azure Key Vault.

The threat actor abused the Azure Storage encryption scopes feature and encrypted the Storage blobs in the Azure Storage accounts. This wasn't sufficient, as the organization could still access the data with the appropriate Azure permissions. In attempt to make the data inaccessible, the actor deletes the key that is used for the encryption. However, it's important to note that Azure Key vaults and keys that are used for encryption purposes are protected by the Azure Key Vault soft-delete feature, with a default period of 90 days, which allows the user to retrieve the deleted key/vault from deletion, preventing cloud-based encryption for ransomware purposes.

After successfully exfiltrating and destroying the data within the Azure environment, the threat actor initiated the extortion phase, where they contacted the victims using Microsoft Teams using one of the previously compromised users, demanding ransom.

Mitigation and protection guidance

Microsoft recently implemented a change in Microsoft Entra ID that restricts permissions on the Directory Synchronization Accounts (DSA) role in Microsoft Entra Connect Sync and Microsoft Entra Cloud Sync. This change helps prevent threat actors from abusing Directory Synchronization Accounts in attacks to escalate privileges. Additionally, a new version released in May 2025 introduces modern authentication, allowing customers to configure application-based authentication for enhanced security (currently in public preview). It is also important to enable Trusted Platform Module (TPM) on the Entra Connect Sync server to securely store sensitive credentials and cryptographic keys, mitigating Storm-0501's credential extraction techniques.

The techniques used by threat actors and described in this blog can be mitigated by adopting the following security measures:

Protecting on-premises

- Turn on tamper protection features to prevent threat actors from stopping security services such as Microsoft Defender for Endpoint, which can help prevent hybrid cloud environment attacks such as Microsoft Entra Connect abuse.
- Run endpoint detection and response (EDR) in block mode so that Defender for Endpoint can block malicious artifacts, even when your non-Microsoft antivirus does not detect the threat or when Microsoft Defender Antivirus is running in passive mode. EDR in block mode works behind the scenes to remediate malicious artifacts detected post-breach.
- Turn on investigation and remediation in full automated mode to allow Defender for Endpoint to take immediate action on alerts to help remediate alerts, significantly reducing alert volume.

Protecting cloud identities

- Secure accounts with credential hygiene: practice the principle of least privilege and audit privileged account activity in your Microsoft Entra ID and Azure environments to slow or stop threat actors.
- Enable Conditional Access policies – Conditional Access policies are evaluated and enforced every time the user attempts to sign in. Organizations can protect themselves from attacks that leverage stolen credentials by enabling policies such as device compliance or trusted IP address requirements.
    - Set a Conditional Access policy to limit the access of Microsoft Entra ID Directory Synchronization Accounts (DSA) from untrusted IP addresses to all cloud apps. Please refer to the advanced hunting section and check the relevant query to get those IP addresses.
    - For Entra Connect Sync servers using application-based authentication, use Conditional Access for workload identities to restrict the application's service principal from similar unauthorized access.
- Ensure multifactor authentication (MFA) requirement for all users. Adding more authentication methods, such as the Microsoft Authenticator app or a phone number, increases the level of protection if one factor is compromised.
    - Ensure phishing-resistant multifactor authentication strength is required for Administrators.
    - Ensure Microsoft Azure overprovisioned identities should have only the necessary permissions.
- Ensure separate user accounts and mail forwarding for Global Administrator accounts. Global Administrator (and other privileged groups) accounts should be cloud-native accounts with no ties to on-premises Active Directory. See other best practices for using Privileged roles here.
- Ensure all existing privileged users have an already registered MFA method to protect against malicious MFA registrations
- Implement Conditional Access authentication strength to require phishing-resistant authentication for employees and external users for critical apps.
- Refer to Azure Identity Management and access control security best practices for further steps and recommendations to manage, design, and secure your Entra ID environment.
- Ensure Microsoft Defender for Cloud Apps connectors are turned on for your organization to receive alerts on the Microsoft Entra ID Directory Synchronization Account and all other users.
- Enable protection to prevent by-passing of cloud Microsoft Entra MFA when federated with Microsoft Entra ID. This enhances protection against federated domains attacks.
- Set the *validatingDomains* property of *federatedTokenValidationPolicy* to "all" to block attempts to sign-in to any non-federated domain (like .onmicrosoft.com) with SAML tokens.
- If only Microsoft Entra ID performs MFA for a federated domain, set *federatedIdpMfaBehavior* to *rejectMfaByFederatedIdp* to prevent bypassing MFA CAPs.
- Turn on Microsoft Entra ID protection to monitor identity-based risks and create risk-based Conditional Access policies to remediate risky sign-ins.

Protecting cloud resources

- Use solutions like Microsoft Defender for Cloud to protect your cloud resources and assets from malicious activity, both in posture management, and threat detection capabilities.
- Enable Microsoft Defender for Resource Manager as part of Defender for Cloud to automatically monitor the resource management operations in your organization. Defender for Resource Manager runs advanced security analytics to detect threats and alerts you about suspicious activity.

    - Enabling Defender for Resource Manager allows users to investigate Azure management operations within the Defender XDR, using the advanced hunting experience.
- Utilize the Azure Monitor activity log to investigate and monitor Azure management events.
- Utilize Azure policies for Azure Storage to prevent network and security misconfigurations and maximize the protection of business data stored in your storage accounts.
- Implement Azure Blog Storage security recommendations for enhanced data protection.
- Utilize the options available for data protection in Azure Storage.
- Enable immutable storage for Azure Blob Storage to protect from accidental or malicious modification or deletion of blobs or storage accounts.
- Apply Azure Resource Manager locks to protect from accidental or malicious modifications or deletions of storage accounts.
- Enable Azure Monitor for Azure Blob Storage to collect, aggregate, and log data to enable recreation of activity trails for investigation purposes when a security incident occurs or network is compromised.
- Enabled Microsoft Defender for Storage using a built-in Azure policy.
- After enabling Microsoft Defender for Storage as part of Defender for Cloud, utilize the *CloudStorageAggregatedEvents* (preview) table in advanced hunting to proactively hunt for storage malicious activity.
- Enable Azure blob backup to protect from accidental or malicious deletions of blobs or storage accounts.
- Apply the principle of least privilege when authorizing access to blob data in Azure Storage using Microsoft Entra and RBAC and configure fine-grained Azure Blob Storage access for sensitive data access through Azure ABAC.
- Use private endpoints for Azure Storage account access to disable public network access for increased security.
- Avoid using anonymous read access for blob data.
- Enable purge protection in Azure Key Vaults to prevent immediate, irreversible deletion of vaults and secrets. Use the default retention interval of 90 days.
- Enable logs in Azure Key Vault and retain them for up to a year to enable recreation of activity trails for investigation purposes when a security incident occurs or network is compromised.
- Enable Microsoft Azure Backup for virtual machines to protect the data on your Microsoft Azure virtual machines, and to create recovery points that are stored in geo-redundant recovery vaults.

General hygiene recommendations

- Utilize Microsoft Security Exposure Management, available in the Microsoft Defender portal, with capabilities such as critical asset protection and attack path analysis that enable security teams to proactively reduce exposure and mitigate the impact of Storm-0501 hybrid attack tactics. In this case, each of the critical assets involved – Entra Connect server, users with DCSync permissions, Global Administrators – can be identified by relevant alerts and recommendations.
- Investigate on-premises and hybrid Microsoft Security Exposure Management attack paths. Security teams can use attack path analysis to trace cross-domain threats that exploit the critical Entra Connect server to pivot into cloud workloads, escalate privileges, and expand their reach. Teams can use the ‘Chokepoint’ view in the attack path dashboard in Microsoft Security Exposure Management to highlight entities appearing in multiple paths.

- Utilize the Critical asset management capability in Microsoft Security Exposure Management by configuring your own custom queries to pinpoint your organization's business-critical assets according to your needs, such as business-critical Azure Storage accounts.

Microsoft Defender XDR detections

Microsoft Defender XDR customers can refer to the list of applicable detections below. Microsoft Defender XDR coordinates detection, prevention, investigation, and response across endpoints, identities, email, apps to provide integrated protection against attacks like the threat discussed in this blog.

Customers with provisioned access can also use Microsoft Security Copilot in Microsoft Defender to investigate and respond to incidents, hunt for threats, and protect their organization with relevant threat intelligence.

Tactic	**Observed activity**	**Microsoft Defender coverage**
Initial access	– Suspicious sign-ins	**Microsoft Defender XDR** – Authentication with compromised credentials – Compromised user account in a recognized attack pattern – Malicious sign in from a risky IP address – Malicious sign in from an IP address associated with recognized attacker infrastructure – Malicious sign in from recognized attacker infrastructure -Malicious sign-in from an unusual user agent – Malicious sign-in from known threat actor IP address – Successful authentication from a malicious IP – Successful authentication from a suspicious IP – Successful authentication using compromised credentials – User compromised through session cookie hijack – User signed in from a known malicious IP Address – Suspicious Azure sign-in by user with active session on a device involved in a credential theft attempt **Microsoft Defender for Identity** – Possibly compromised user account signed in – Possibly compromised service principal account signed in **Microsoft Defender for Cloud Apps** – Suspicious login from AADInternals tool **Microsoft Defender for Cloud** ***Defender for Resource Manager*** – Suspicious invocation of a high-risk 'Initial Access' operation detected (Preview)

		Defender for Storage – Access from an unusual location to a storage account – Access from an unusual location to a sensitive blob container – Access from a known suspicious IP address to a sensitive blob container – Access from a suspicious IP address – Unusual unauthenticated public access to a sensitive blob container
Execution	– Various types of execution-related suspicious activity by an attacker were observed – Crafting access tokens and executing actions against the cloud	**Microsoft Defender for Endpoint** – Compromised account conducting hands-on-keyboard attack – Potential human-operated malicious activity – Suspicious cmdlets launch using AADInternals
Persistence	– Federated domain backdoor was added	**Microsoft Defender for Cloud Apps** – Backdoor creation using AADInternals tool
Privilege escalation	– Elevated access to Azure resources – Assignment of *Owner* Azure role	**Microsoft Defender XDR** – Suspicious Azure elevate access operation by a user with an active session on a device involved in a credential theft attempt – Possibly compromised Microsoft Entra Connect Sync account elevated its access to Azure resources – Possibly compromised user elevated access to Azure resources **Microsoft Defender for Cloud** ***Defender for Resource Manager*** – Suspicious elevate access operation – Suspicious invocation of a high-risk ‘Privilege Escalation’ operation detected (Preview) – Suspicious Azure role assignment detected (Preview)

Defense evasion	– Attempts to tamper with Microsoft Defender Antivirus – Manipulation of Azure Storage account configurations	**Microsoft Defender for Endpoint** – Attempt to turn off Microsoft Defender Antivirus protection **Microsoft Defender for Cloud** ***Defender for Resource Manager*** – Suspicious invocation of a high-risk ‘Defense Evasion’ operation detected (Preview)
Credential access	– Entra Connect Sync server compromise and sync accounts extraction – Extracting credentials from remote machines – Executing DCSync operation against a domain controller – Access Azure Storage accounts access keys – Creation of a key inside an Azure Key Vault for encryption of Azure Storage data	**Microsoft Defender Antivirus** – Trojan:Win32/SuspAdSyncAccess.A!EntraConnect – Backdoor:Win32/AdSyncDump!EntraConnect – Behavior:Win32/DumpADConnectCreds.A!EntraConnect – Trojan:Win32/SuspAdSyncAccess.A!EntraConnect – Behavior:Win32/SuspAdsyncBin.A!EntraConnect **Microsoft Defender for Endpoint** – Entra Connect Sync credentials extraction attempt – Indication of local security authority secrets theft – Potential Entra Connect Tampering – Ongoing hands-on-keyboard attack using Impacket toolkit – Possible source of DCSync attack **Microsoft Defender for Identity** – Suspected DCSync attack (replication of directory services) **Microsoft Defender for Cloud Apps** – Compromised Microsoft Entra ID Cloud Sync account – AADInternals tool used by a Microsoft Entra Sync account – Entra Connect Sync account suspicious activity following a suspicious login – Suspicious sign-in to Microsoft Entra Connect Sync account **Microsoft Defender for Cloud** ***Defender for Resource Manager*** – Suspicious invocation of a high-risk ‘Credential Access’ operation detected (Preview) ***Defender for Key Vault***

		– Suspicious key vault recovery detected – Unusual application accessed a key vault – Unusual operation pattern in a key vault – Unusual user accessed a key vault
Discovery	– Verifying whether Microsoft Defender for Endpoint is onboarded on a machine – Reconnaissance activity against Active Directory/Entra ID/Azure – AzureHound tool invocation in the cloud environment	**Microsoft Defender for Endpoint** – Suspicious sequence of exploration activities **Microsoft Defender for Cloud Apps** – Suspicious use of AzureHound **Microsoft Defender for Identity** – Reconnaissance tool was observed **Microsoft Defender for Cloud** ***Defender for Resource Manager*** – AzureHound tool invocation detected
Lateral movement	– Lateral movement between endpoints in the network – Lateral movement using Evil-WinRM – Cloud sign-in attempts using stolen credentials or access tokens extracted from compromised endpoints	**Microsoft Defender for Endpoint** – Possibly malicious use of proxy or tunneling tool – Suspicious remote PowerShell execution **Microsoft Defender for Cloud Apps** – Suspicious login from AADInternals tool
Exfiltration	– Data collection and theft from Azure Storage accounts	**Microsoft Defender for Cloud** ***Defender for Resource Manager*** – Suspicious invocation of a high-risk ‘Data Collection’ operation detected (Preview)

		Defender for Storage – The access level of a potentially sensitive storage blob container was changed to allow unauthenticated public access – Publicly accessible storage containers successfully discovered – Publicly accessible storage containers unsuccessfully scanned – Unusual amount of data extracted from a storage account – Unusual deletion in a storage account – Unusual amount of data extracted from a sensitive blob container – Unusual number of blobs extracted from a sensitive blob container – Unusual SAS token was used to access an Azure storage account from a public IP address – Suspicious external access to an Azure storage account with overly permissive SAS token – Suspicious external operation to an Azure storage account with overly permissive SAS token – Access from a suspicious IP address
Impact	– Mass Azure data store resources deletion and encryption	**Microsoft Defender XDR** – Suspicious Azure data store resources deletion attempt by a user with an active session on a device involved in a credential theft attempt **Microsoft Defender for Cloud** ***Defender for Resource Manager*** – Suspicious backup resource deletion (Preview) – Suspicious invocation of a high-risk ‘Impact’ operation detected (Preview) ***Defender for Storage*** – Unusual deletion in a storage account

Threat intelligence reports

Microsoft customers can use the following reports in Microsoft products to get the most up-to-date information about the threat actor, malicious activity, and techniques discussed in this blog. These reports provide the intelligence, protection information, and recommended actions to prevent, mitigate, or respond to associated threats found in customer environments.

Microsoft Defender XDR threat analytics

- Actor Profile: Storm-0501
- Activity Profile: Ransomware actor Storm-0501 expands to hybrid cloud environments

Microsoft Security Copilot customers can also use the Microsoft Security Copilot integration in Microsoft Defender Threat Intelligence, either in the Security Copilot standalone portal or in the embedded experience in the Microsoft Defender portal to get more information about this threat actor.

Hunting queries

Microsoft Defender XDR

Microsoft Defender XDR customers can run the following query to find related activity in their networks:

Sign-in activity

Explore sign-in activity from I*dentityLogonEvents*, look for uncommon behavior, such as sign-ins from newly seen IP addresses or sign-ins to new applications that are non-sync related:

```
IdentityLogonEvents
| where Timestamp > ago(30d)
| where AccountDisplayName contains "On-Premises Directory Synchronization Service Account"
| extend ApplicationName = tostring(RawEventData.ApplicationName)
| project-reorder Timestamp, AccountDisplayName, AccountObjectId, IPAddress, ActionType, ApplicationName, OSPlatform, DeviceType
```

The activity of the sync account is typically repetitive, coming from the same IP address to the same application. Any deviation from the natural flow is worth investigating. Cloud applications that are usually accessed by the Microsoft Entra ID sync account are Microsoft Azure Active Directory Connect, Windows Azure Active Directory, and Microsoft Online Syndication Partner Portal.

Cloud activity

Explore the cloud activity (*ActionType*) of the sync account. Similar to sign-in activity, this account by nature performs a certain set of actions including *update User*., *update Device*., and so on. New and uncommon activity from this user might indicate an interactive use of the account, which could legitimate action from someone in the organization or malicious action by the threat actor.

```
CloudAppEvents
| where Timestamp > ago(30d)
| where AccountDisplayName has "On-Premises Directory Synchronization Service Account"
| extend Workload = RawEventData.Workload
```

```
| project-reorder Timestamp, IPAddress, AccountObjectId, ActionType, Application, Workload, DeviceType, OSPlatform, UserAgent, ISP
```

Pay close attention to action from different DeviceTypes or OSPlatforms, this account automated service is performed from one specific machine, so there shouldn't be any variety in these fields.

Azure management events

Explore Azure management events by querying the new *CloudAuditEvents* table in advanced hunting in the Defender portal. The *OperationName* column indicates the type of control-plane event executed by the user.

```
let Storm0501Operations = dynamic([
//Microsoft.Authorization
"Microsoft.Authorization/elevateAccess/action",
"Microsoft.Authorization/roleAssignments/write",
"Microsoft.Authorization/locks/delete",
//Microsoft.Storage
"Microsoft.Storage/storageAccounts/write",
"Microsoft.Storage/storageAccounts/listkeys/action",
"Microsoft.Storage/storageAccounts/delete",
"Microsoft.Storage/storageAccounts/blobServices/containers/immutabilityPolicies/delete",
"Microsoft.Storage/storageAccounts/encryptionScopes/write",
//Microsoft.Compute
"Microsoft.Compute/snapshots/delete",
"Microsoft.Compute/restorePointCollections/delete",
//Microsoft.RecoveryServices
"Microsoft.RecoveryServices/Vaults/backupFabrics/protectionContainers/delete",
//Microsoft.KeyVault
"Microsoft.KeyVault/vaults/write"
]);
CloudAuditEvents
```

```
| where Timestamp > ago(30d)
| where AuditSource == "Azure" and DataSource == "Azure Logs"
| where OperationName in~ (Storm0501Operations)
| extend EventName = RawEventData.eventName
| extend UserId = RawEventData.principalOid, ApplicationId = RawEventData.applicationId
| extend Status = RawEventData.status, SubStatus = RawEventData.subStatus
| extend Claims = parse_json(tostring(RawEventData.claims))
| extend UPN = Claims["http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"]
| extend AuthMethods = Claims["http://schemas.microsoft.com/claims/authnmethodsreferences"]
| project-reorder ReportId, EventName, Timestamp, UPN, UserId, AuthMethods, IPAddress, OperationName, AzureResourceId, Status, SubStatus, ResourceId, Claims, ApplicationId
```

Exposure of resources and users

Explore Microsoft Security Exposure Management capabilities by querying the *ExposureGraphNodes* and *ExposureGraphEdges* tables in the advanced hunting in the Defender portal. By utilizing these tables, you can identify critical assets, including Azure Storage accounts that contain sensitive data or protected by an immutable storage policy. All predefined criticality rules can be found here: Predefined classifications

```
ExposureGraphNodes
| where NodeLabel =~ "microsoft.storage/storageaccounts"
// Criticality check
| extend CriticalityInfo = NodeProperties["rawData"]["criticalityLevel"]
| where isnotempty( CriticalityInfo)
| extend CriticalityLevel = CriticalityInfo["criticalityLevel"]
| extend CriticalityLevel = case(
              CriticalityLevel == 0, "Critical",
              CriticalityLevel == 1, "High",
              CriticalityLevel == 2, "Medium",
              CriticalityLevel == 3, "Low", "")
| extend CriticalityRules = CriticalityInfo["ruleNames"]
```

```
| extend StorageContainsSensitiveData = CriticalityRules has "Databases with Sensitive Data"

| extend ImmutableStorageLocked = CriticalityRules has "Immutable and Locked Azure Storage"

// Exposure check

| extend ExposureInfo = NodeProperties["rawData"]["exposedToInternet"]

| project-reorder NodeName, NodeId, CriticalityLevel, CriticalityRules, StorageContainsSensitiveData, ImmutableStorageLocked, ExposureInfo
```

The following query can identify critical users who are mainly assigned with privileged Microsoft Entra roles, including *Global Administrator*:

```
ExposureGraphNodes

| where NodeLabel =~ "user"

| extend UserId = NodeProperties["rawData"]["accountObjectId"]

| extend IsActive = NodeProperties["rawData"]["isActive"]

// Criticality check

| extend CriticalityInfo = NodeProperties["rawData"]["criticalityLevel"]

| where isnotempty(CriticalityInfo)

| extend CriticalityLevel = CriticalityInfo["criticalityLevel"]

| extend CriticalityLevel = case(

              CriticalityLevel == 0, "Critical",

              CriticalityLevel == 1, "High",

              CriticalityLevel == 2, "Medium",

              CriticalityLevel == 3, "Low", "")

| extend CriticalityRules = CriticalityInfo["ruleNames"]

| extend GlobalAdministrator = CriticalityRules has "Global Administrator"

| project-reorder NodeName, NodeId, UserId, IsActive, CriticalityLevel, CriticalityRules, GlobalAdministrator
```

Omri Refaeli, Karam Abu Hanna, and Alon Marom

Learn more

For the latest security research from the Microsoft Threat Intelligence community, check out the Microsoft Threat Intelligence Blog.

To get notified about new publications and to join discussions on social media, follow us on LinkedIn, X (formerly Twitter), and Bluesky.

To hear stories and insights from the Microsoft Threat Intelligence community about the ever-evolving threat landscape, listen to the Microsoft Threat Intelligence podcast.
