

How Storm-2949 turned a compromised identity into a cloud-wide breach

By Microsoft Defender Security Research Team

Published: 2026-05-18 · Archived: 2026-06-04 02:00:42 UTC

Microsoft Threat Intelligence recently uncovered a methodical, sophisticated, and multi-layered attack, where a threat actor we track as Storm-2949 launched a relentless campaign with a singular focus: to exfiltrate as much sensitive data from a target organization's high-value assets as possible. The attack exfiltrated data from Microsoft 365 applications, file-hosting services, and Azure-hosted production environments, where the organization's production application ecosystem resides.

What began as a targeted identity compromise rapidly evolved into a full-spectrum assault on the organization's cloud infrastructure. The attack spanned various Azure resources, with emphasis on software-as-a-service (SaaS), platform-as-a-service (PaaS), and infrastructure-as-a-service (IaaS) layers.

Storm-2949 didn't rely on traditional malware and other on-premises tactics, techniques, and procedures (TTPs). Instead, they leveraged legitimate cloud and Azure management features to gain control-plane and data-plane access, which they then used to execute code remotely on VMs, and access sensitive cloud resources such as Key Vaults and storage accounts, among others. These activities allowed them to move laterally across cloud and endpoint environments while blending into expected administrative behavior.

As organizations continue to adopt cloud infrastructure at scale, threat actors are increasingly targeting identity and control plane access rather than individual devices. When cloud identities are compromised, legitimate administrative features can be used to achieve outcomes similar to traditional lateral movement, often with fewer indicators of compromise. Behavior-based detections across endpoints, cloud environments, and identities—such as those provided by Microsoft Defender—can help teams identify and correlate these activities.

In this blog, we unpack the full attack chain from initial access to cloud and endpoint takeover. We then offer actionable insights into how organizations can detect, contain, and prevent similar identity-driven threats in their environments.

Attack chain overview

The campaign that Storm-2949 deployed can be divided into two phases: targeted identity compromise and cloud infrastructure compromise. We discuss each of these phases in detail in the succeeding sections.

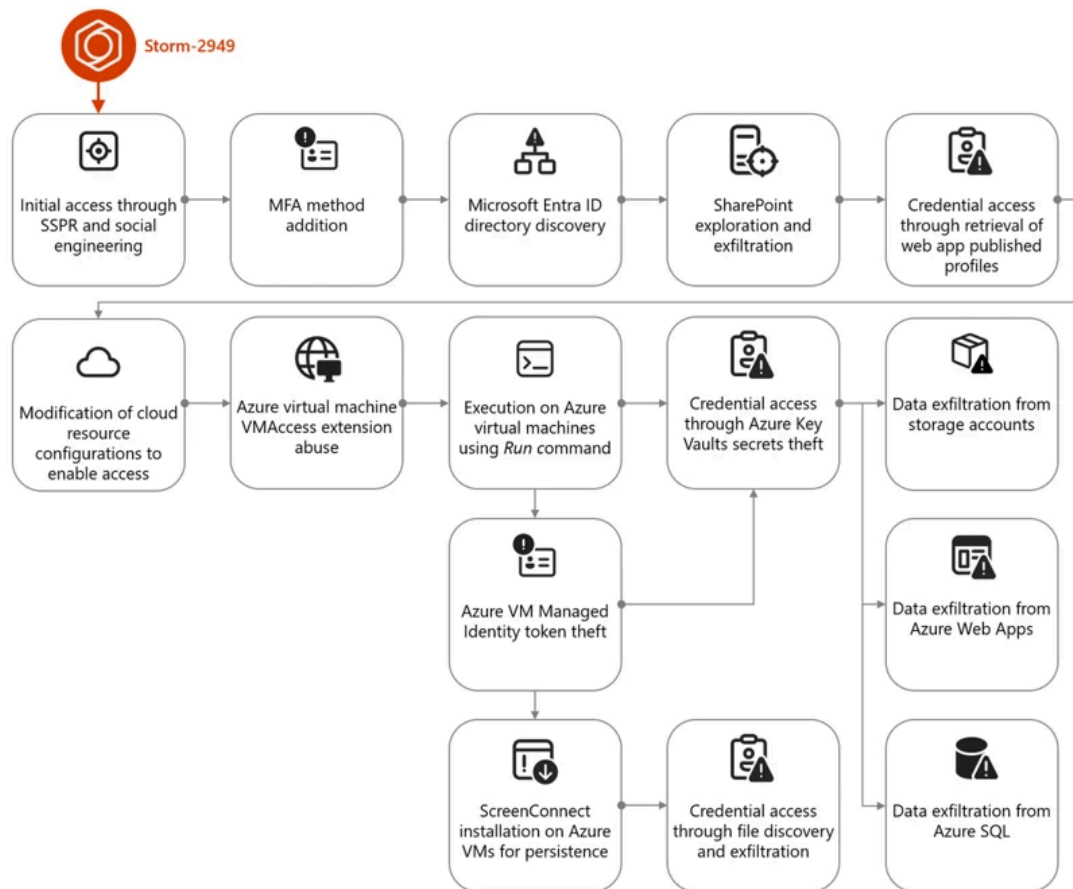


Figure 1. Storm-2949 attack diagram.

Cloud compromise: Microsoft Entra ID and Microsoft 365

In this phase, the threat actor targeted specific users through social engineering to obtain their Microsoft Entra ID credentials. Using these credentials, the threat actor then proceeded to exfiltrate data from Microsoft 365 applications.

Initial access and persistence through targeted social engineering and SSPR abuse

We assess with high confidence that Storm-2949 leveraged a social engineering technique consistent with known abuses of Microsoft’s Self-Service Password Reset (SSPR) process. In such attacks, a threat actor initiates the SSPR process on behalf of a targeted user and subsequently employs social engineering tactics to persuade the user to complete multifactor authentication (MFA) prompts that appear to be legitimate.

For example, the threat actor might impersonate an internal information technology (IT) support representative and contact the user claiming that their account requires urgent verification, instructing them to approve MFA prompts as part of a routine password reset procedure.

Once the user approves these prompts, the threat actor is able to reset the user’s password and remove existing authentication methods, such as phone numbers, email addresses, and Microsoft Authenticator registrations, effectively eliminating MFA as a control and enabling unrestricted account access. Immediately after gaining access to the compromised account, the threat actor is then prompted to re-enable MFA and register a new

authentication method. At this stage, the threat actor enrolls Microsoft Authenticator on their own device, granting themselves persistent access and preventing the legitimate user from signing in.

Storm-2949 used a similar process repeatedly across multiple users within the targeted organization. The selection of victims, which included IT personnel and senior leadership, indicated deliberate targeting. Based on the roles of the compromised users and the investigation findings, we assess that the threat actor likely used an organized and convincing phishing scheme to lure users into completing the fraudulent MFA prompts and thereby compromise their identities.

Directory discovery and persistence

Following the initial identity takeover, the threat actor conducted directory discovery using Microsoft Graph API. Using a custom Python script, they issued automated API requests to enumerate users and applications within the tenant. Through these queries, the threat actor searched Microsoft Entra ID for user accounts based on name patterns and role attributes, likely to identify privileged identities and additional high-value targets.

Figure 2 illustrates the types of Graph API queries observed:

```
...graph.microsoft.com/v1.0/users?$select=displayName,userPrincipalName,jobTitle&$top=50  
  
...graph.microsoft.com/v1.0/users?$filter=startsWith(displayName,  
<REDACTED>)&$select=displayName,userPrincipalName,jobTitle
```

Figure 1. Discovery using cURL.

During this attack phase, the threat actor also attempted to establish persistence by adding credentials to a compromised service principal to enable continued access independent of the compromised user accounts. This attempt failed due to insufficient permissions. Undeterred, the threat actor continued enumerating service principals and known application identifiers, indicating an effort to map application-level access paths and expand long-term footholds within the environment.

Using the same social engineering techniques and SSPR abuse described earlier, the threat actor expanded their foothold by compromising three additional cloud user accounts.

Microsoft 365 discovery and exfiltration

Storm-2949 leveraged their access to the compromised user accounts to explore and exfiltrate files from the victim organizations' cloud file storage services. Shortly after obtaining initial access within the organization, they targeted Microsoft 365 applications, including OneDrive and SharePoint, identifying and accessing the organization's sensitive files, focusing on IT documents concerning virtual private network (VPN) configurations and remote access procedures. We assess that this behavior reflects an attempt to identify opportunities for lateral movement from a compromised cloud identity into the endpoint network.

The threat actor then launched a large-scale data exfiltration from these storage services. In one instance, Storm-2949 used the OneDrive web interface to download thousands of files in a single action to their own

infrastructure. This pattern of data theft was repeated across all compromised user accounts, likely because different identities had access to different folders and shared directories.

Cloud compromise: Microsoft Azure

Armed with access to multiple compromised identities – which were assigned with privileged custom Azure role-based access control (RBAC) roles on several Azure subscriptions – and a growing understanding of the environment, the threat actor shifted focus toward the victim’s Azure environment. With a clear agenda centered on data exfiltration, Storm-2949 demonstrated a relentless drive to uncover and extract the most sensitive assets within the victim’s Azure environment, specifically from production-based Azure subscriptions.

Their campaign targeted not only core applications but also the broader ecosystem of interconnected resources such as Azure App Services web applications, Azure Key Vaults, Azure Storage accounts, and SQL databases. These resources collectively power the organization’s cloud-hosted services. This phase marked a transition from identity-centric abuse and SaaS data theft to targeting a range of Azure services, with an emphasis on both PaaS and IaaS workloads.

Azure App Service and Key Vault compromise

One of Storm-2949’s main targets was a production Azure App Service web application that contained sensitive data. Following several failed attempts to access this application, likely due to gateway and network restrictions, Storm-2949 shifted focus to other web apps that appeared to be part of the same ecosystem. These auxiliary apps, such as those handling authentication or internal APIs, were individually deployed Azure App Service instances with their own resource identities.

Storm-2949 successfully compromised several of these secondary web apps by taking advantage of the user’s privileged Azure RBAC permissions and invoking the Azure management-plane operation, `microsoft.Web/sites/publishxml/action`, which retrieves the application’s publishing profile. This profile often contains basic authentication credentials for deployment endpoints such as FTP, Web Deploy, and the Kudu management console. Kudu is a built-in administrative interface for Azure App Services that allows authenticated users to browse the file system, inspect environment variables, and execute commands within the app’s context.

Despite successfully compromising several of these auxiliary web apps, Storm-2949 was unable to gain access to the primary production application they were ultimately targeting. It is assessed, that the secondary services, while part of the same broader ecosystem, didn’t contain the level of sensitive data or privileged access the threat actor was seeking. While these footholds provided visibility into application configurations and infrastructure, they didn’t deliver the high-value assets that aligned with the threat actor’s data exfiltration objectives. As a result, the threat actor was forced to pursue alternative paths in their effort to reach the production web app.

Storm-2949 recalibrated their approach and shifted their focus toward backend resources that were part of the sensitive web app ecosystem and could provide stronger leverage. The threat actor pivoted to the organization’s Azure Key Vault estate – an environment more likely to centralize sensitive secrets and offer indirect access to production systems. Part of the compromised user’s Azure RBAC permissions was the privileged Owner role over a specific Key Vault that seemed to contain credentials that would enable the compromise of the production application.

Over the span of four minutes, the threat actor successfully manipulated Key Vault access configurations and accessed dozens of secrets within the said Key Vault. These secrets included database connection strings, identity credentials, and more, dramatically expanding the attack's blast radius.

Among these secrets, we believe the threat actor found credentials that enabled them to access the application they coveted the most, which was the main production web app. After they successfully authenticated into the web app, the threat actor changed its password to retain control. They then began exfiltrating sensitive data from it.

Azure Storage and SQL data exfiltration

In parallel, Storm-2949 expanded access across additional cloud resources inside the ecosystem that contained the web app, including Azure Storage accounts and an Azure SQL server.

To enable access to the server, the threat actor abused their existing Azure RBAC permissions to manipulate the SQL server firewall rules by using the `microsoft.sql/servers/firewallrules/write` operation. They then connected to the SQL server using the credentials they obtained (along with the web app credentials) from the compromised Key Vault.

The threat actor proceeded with data exfiltration and continued to delete the modified SQL firewall rules, which is an activity consistent with defense evasion.

Similar to the SQL server compromise, to set up and prepare for massive data exfiltration from Azure Storage, the threat actor also manipulated storage account network access configurations using the `microsoft.storage/storageaccounts/write` operation. This manipulation enabled public access to the storage accounts from a closed set of threat actor-owned IP addresses. In addition, the threat actor abused the Azure management-plane operation `microsoft.Storage/storageAccounts/listkeys/action` to access multiple storage account Shared Access Signature (SAS) tokens and account keys, enabling the use of static, non-interactive authentication to retrieve data.

Using these keys, the threat actor downloaded large volumes of data from several Azure Storage accounts using a custom Python script that leveraged the Azure SDK for Storage. The script allowed them to programmatically enumerate and download blobs directly to their own endpoint device. This storage-based exfiltration continued over multiple days since the initial access, with the threat actor alternating between secret- and OAuth-based authentication as access conditions and controls evolved.

Azure Virtual Machines compromise

Apart from the web app and data-store resource compromise, the abuse of [Azure Virtual Machine \(VM\) extensions](#) and administrative features – specifically [Run Command](#) and the [VMAccess extension](#) – were also prominent elements of this attack. These activities appear to have been primarily intended to expand operational access within the victim environment by leveraging compromised VMs as intermediary footholds. Observed actions across these systems focused on credential harvesting and environment discovery, as well as attempts to access resources that weren't directly reachable through previously compromised identities. These efforts included domain reconnaissance and the collection of authentication material that could facilitate movement between cloud and on-premises environments, as well as enable access to additional high-value assets.

Shortly after the initial access, the threat actor operated in parallel, trying to compromise the organization's virtual machines. Using the compromised users assigned with privileged Azure RBAC permissions, the threat actor deployed the VMAccess extension to create a new local administrator account on a targeted VM. VMAccess is an Azure VM extension intended to help administrators restore access to a VM when credentials get lost or misconfigured by allowing password resets or the addition of privileged local users through the Azure management plane. In this case, the threat actor abused the extension to gain backdoor access to an administrator user on the VM.

Using the Run Command feature, the threat actor deployed a script attempting to abuse the VM's managed identity by requesting an access token from the Azure Instance Metadata Service (IMDS) and using it to authenticate to – and retrieve secrets from – the production web app-related Key Vault. However, the threat actor wasn't able to retrieve the secrets because the managed identity lacked the required permissions. Yet, this attempt shows the threat actor using guest-level execution as a bridge to additional Azure resource access through workload identity.

```
$token = (Invoke-RestMethod -Headers @{"Metadata"="true"} -Method Get -Uri
"http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01
&resource=https://vault.azure.net").access_token; $url =
"https://<REDACTED>.vault.azure.net/secrets/<REDACTED>?api-version=7.4";
$secret = (Invoke-RestMethod -Headers @{"Authorization"="Bearer $token"} -
Method Get -Uri $url).value; Write-Output $secret
```

Figure 2. Token theft and Key Vault access script.

ScreenConnect installation and defense evasion

Storm-2949 further abused the Run Command by running a PowerShell script intended to deploy persistent remote access while reducing host-based security visibility on multiple VMs.

The script attempted to weaken Microsoft Defender Antivirus by disabling several protections, including real-time protection and behavior monitoring, and by interfering with its associated service. These changes lowered the likelihood that subsequent activity would be blocked or generate actionable alerts on the device.

The script then installed the ScreenConnect remote monitoring and management (RMM) tool obtained from threat actor-controlled infrastructure. The installation process included several steps intended to masquerade the tool's presence, such as making the network request appear consistent with trusted software updates and placing files in locations intended to resemble legitimate system content.

To further obscure the tool's presence, the script attempted to rename or configure the installed service to resemble legitimate Windows components, providing a simple form of local masquerading.

Finally, the script attempted cleanup actions to remove local forensic artifacts that could be attributed to the threat actor. These included clearing Windows event logs, removing execution artifacts, and deleting command history and temporary files. Such steps are commonly observed in post-compromise activity and are generally intended to complicate investigation rather than provide durable evasion.

Post-compromise activity using ScreenConnect

The threat actor used the deployed ScreenConnect to launch commands across multiple compromised devices, performing basic discovery. This included collecting host level details (for example, operating system and configuration information) and enumerating domain context such as user accounts and group memberships.

Across a subset of those hosts, the threat actor focused on credential harvesting techniques. They discovered and exfiltrated .pfx certificate files – artifacts that might contain private keys and could be valuable for follow-on access if imported or reused elsewhere. In parallel, they searched for remote file shares for likely credential exposure by scanning files for password related strings. Not every collection effort occurred on every host; rather, it was distributed across systems based on what data and access each host provided.

These actions show ScreenConnect being used as a practical execution channel to run discovery, collect credentials, and attempt to operationalize access across different devices.

While the threat actor ultimately established execution on several endpoints, these systems didn't appear to yield high value data aligned with their objectives. The endpoint activity primarily served as a secondary capability for discovery and credential harvesting, rather than a core exfiltration channel.

Throughout this incident, Microsoft Defender generated multiple alerts that helped analysts piece together activity across endpoints and cloud. Defender correlated these signals into unified incidents, surfacing high-fidelity alerts and a coherent view of threat actor activity. This kind of cross-domain correlation – collecting and normalizing telemetry and linking related alerts – illustrates the value of an integrated detection and response approach for improving signal-to-noise clarity and end-to-end visibility.

Mitigation and protection guidance

The visibility provided by correlated alerts across identities, cloud, and endpoints can help organizations investigate and understand attacks end-to-end. Building on this visibility, organizations can reduce risk and limit the impact of similar attacks by deploying appropriately scoped detection and response capabilities (including Microsoft Defender where applicable) and by applying targeted hardening practices.

Ensure adequate security coverage across attack surfaces

To effectively detect and respond to attacks that span identity, cloud, and endpoint environments, organizations should ensure they have monitoring, detection, and response capabilities deployed and properly configured across those surfaces. The following examples describe how Microsoft Defender capabilities can be used to help with this; equivalent controls might be available in other security solutions.

Use Microsoft Defender for Endpoint for:

- [Tamper protection](#) enabled to prevent threat actors from stopping security services such as Defender for Endpoint, which can help prevent hybrid cloud environment attacks.
- [Endpoint detection and response \(EDR\)](#) in block mode so that Defender for Endpoint can block malicious artifacts, even when your non-Microsoft antivirus doesn't 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.

- [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.

Use [Microsoft Defender for Cloud](#) to protect your cloud resources and assets from malicious activity, both in posture management ([Microsoft Defender Cloud Security Posture Management](#)), and threat detection capabilities. Enable workload protection capabilities across cloud resources, including:

- [Microsoft Defender for Resource Manager](#)
- [Microsoft Defender for App Service](#)
- [Microsoft Defender for Key Vault](#)
- [Microsoft Defender for Storage](#)
- [Microsoft Defender for Databases](#)
- [Microsoft Defender for Servers](#)

In addition, leverage the Microsoft Defender XDR to hunt for threats across cloud environments and resource with [advanced hunting](#). Security teams can proactively investigate threat actor activity by querying telemetry across multiple domains using tables such as [CloudAuditEvents](#), [CloudStorageAggregatedEvents](#), and others, enabling deep visibility into control-plane and data-plane operations, authentication events, and cross-service attack patterns.

Use [Microsoft Defender for Cloud Apps](#) and enable connectors to monitor SaaS activity.

Security hardening and best practices

In addition to deploying the appropriate Defender capabilities, organizations should apply the following security controls and practices to mitigate similar attack paths:

Identity protection

- 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.
- [Ensure MFA](#) is required 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 MFA strength](#) is required for Administrators and privileged user accounts.
- 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 cloud environment.
- Turn on [Microsoft Entra ID protection](#) to monitor identity-based risks and create risk-based Conditional Access policies to remediate risky sign-ins.

Cloud resource protection

- Use the [Azure Monitor activity](#) log to investigate and monitor Azure management events.
- Configure and harden resources firewall rules and access controls to allow access only from trusted IP ranges and virtual networks to prevent unauthorized access.
- Use Azure policies to continuously enforce the hardened configurations.
- Practice and apply [Azure Storage security](#) best practices:
- Use 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 Blob Storage security recommendations](#) for enhanced data protection.
- Use 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.
- 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.
- [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 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 attribute-based access control \(ABAC\)](#).
- Practice and apply [Azure Key Vault security](#) best practices:
- 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.
- [Restrict public network access](#) to Azure Key Vault by enabling private endpoints and disabling public access to reduce exposure to unauthorized access attempts.
- Regularly audit Azure RBAC role assignments and Key Vault access policies, depending on the Key Vault permission model, to ensure least privilege and detect over-permissioned identities. Microsoft explicitly recommends Azure RBAC over Key Vault access policies.
- Configure SQL server firewall rules to restrict access to known IP addresses and monitor for unauthorized changes to firewall configurations.
- Enforce [authentication through Microsoft Entra ID](#) for SQL instances to reduce reliance on static credentials and improve access control
- Practice and apply Azure App Service security [best practices](#):
- [Disable legacy authentication](#) methods and enforce managed identity usage for Azure App Services to prevent credential theft through publishing profiles.
- Monitor and restrict access to Azure App Service publishing credentials by limiting RBAC permissions and auditing usage of the publish profile API.
- [Enable diagnostic logging](#) in App Service logs to detect suspicious deployment or configuration changes.

- 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.
- Audit and restrict the use of Azure VM features and extensions such as Run Command and VMAccess by limiting RBAC permissions and monitoring for suspicious invocation patterns.
- Use [Azure Policy to restrict or audit](#) the deployment of Azure VM extensions across your subscriptions.

General hygiene recommendations

- Investigate [Microsoft Security Exposure Management attack paths](#). Security teams can use attack path analysis to trace cross-domain threats that pivot to cloud workloads, escalate privileges, and expand their reach.
- Use Azure Policy to enforce organizational standards and prevent the deployment of risky configurations, such as public access to sensitive resources.
- Implement consistent [cloud security recommendations](#) hygiene.

Indicators of compromise (IOCs)

IOCs reflect observations at the time of analysis and may not be exhaustive or persistent.

Indicator	Type	Description
176.123.4[.]44	IP address	Attacker egressed from this address
91.208.197[.]87	IP address	Attacker egressed from this address
185.241.208[.]243	IP address	ScreenConnect instance used by Attacker

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, and 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.

Note that the following detections only covers the threat activities we've observed at the time of analysis.

Tactic	Observed activity	Microsoft Defender coverage
Initial access	– Sign-in activity from attacker infrastructure to compromised identities – Sign-in and authentication activity to	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

	Azure resources	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 – Impossible Travel Microsoft Defender for Identity – Possibly compromised user account signed in – Possibly compromised service principal account signed in Microsoft Defender for Cloud <i>Defender for Resource Manager</i> Suspicious invocation of a high-risk ‘Initial Access’ operation detected (Preview) <i>Defender for Databases</i> Login from an unusual location <i>Defender for Storage</i> – Access from an unusual location to a storage account – Access from an unusual location to a storage blob container – 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	Microsoft Defender XDR – Possibly compromised user ran a malicious script using an Azure VM extension – Potential hybrid ransomware or hands-on-keyboard attack originating from Azure VM extensions – Hybrid ransomware or hands-on-keyboard attack originating from Azure VM extensions – Azure VM extension activity followed by ransomware or hands-on-keyboard attack Microsoft Defender for Cloud <i>Defender for Resource Manager</i> – Suspicious invocation of a high-risk ‘Execution’ operation detected (Preview) – Azure Resource Manager operation from suspicious IP address – Suspicious Run Command invocation detected (Preview) <i>Defender for Servers P2</i> – Run Command with a suspicious script was detected on your virtual machine – Suspicious Run Command usage was detected on your virtual machine (Preview) – Suspicious unauthorized Run Command usage was detected on your virtual machine (Preview) Microsoft Defender for Endpoint – Compromised account conducting hands-on-keyboard attack – Potential human-operated malicious activity – Suspicious process execution – Suspicious command execution via ScreenConnect – Suspicious activity through Azure VM extension process
Persistence	– Attacker device registered as MFA method	Microsoft Defender for Identity – Suspicious addition of default third-party MFA method to user account – Suspicious Entra device join or registration

	– ScreenConnect installed on Azure VMs	Microsoft Defender for Cloud Apps – Suspicious addition of device with strong MFA – Suspicious addition of strong authentication device – Malicious device with strong MFA was registered Microsoft Defender for Endpoint Uncommon remote access software
Defense evasion	– Attempts to tamper with Microsoft Defender Antivirus – Manipulation of Azure Storage account, Key Vault, and SQL database configurations	Microsoft Defender for Endpoint – Attempt to turn off Microsoft Defender Antivirus protection – Attempt to clear event log – Event log was cleared Microsoft Defender for Cloud Defender for Resource Manager Suspicious invocation of a high-risk ‘Defense Evasion’ operation detected (Preview) Defender for Key Vault Suspicious policy change and secret query in a key vault
Credential access	– Secret extraction from Azure Key Vault – Attempted theft of workload identity tokens using Azure VM Run Command – Credential harvesting from endpoints through ScreenConnect – Publishing Azure App Service web app profile for credential access – Listing Azure storage account access keys for access	Microsoft Defender Antivirus – Trojan:Win32/SuspAdSyncAccess – Backdoor:Win32/AdSyncDump – Behavior:Win32/DumpADConnectCreds – Trojan:Win32/SuspAdSyncAccess – Behavior:Win32/SuspAdsyncBin Microsoft Defender for Endpoint – Indication of local security authority secrets theft – Password stealing from files Microsoft Defender for Cloud Defender for Resource Manager Suspicious invocation of a high-risk ‘Credential Access’ operation detected (Preview) Defender for Servers P2 Run Command with a suspicious script was detected on your virtual machine

		<i>Defender for Key Vault</i> – Suspicious policy change and secret query in a key vault – High volume of operations in a key vault – Unusual application accessed a key vault – Unusual operation pattern in a key vault – Unusual user accessed a key vault – Access from a suspicious IP address to a key vault
Discovery	– Domain and system discovery commands run on virtual machines	Microsoft Defender for Endpoint Suspicious sequence of exploration activities Microsoft Defender for Cloud Apps Suspicious file access
Lateral movement	– Traversal between cloud resources and applications	Microsoft Defender for Identity Suspicious sign-in to a web app following MFA phone number tampering activity Microsoft Defender for Cloud Apps Compromised user accessed a SaaS application Microsoft Defender for Cloud <i>Defender for Resource Manager</i> Suspicious invocation of a high-risk ‘Data Collection’ operation detected (Preview)
Exfiltration	– Data exfiltration from Azure Storage accounts and other resources – Data exfiltration from file storage services	Microsoft Defender XDR Suspicious behavior: Mass download Microsoft Defender for Cloud Apps – Suspicious massive data read – Suspicious mass download from risky or unusual session – Suspicious mass download from risky or unusual session – Suspicious mass download from risky or unusual session – Possible exfiltration of data archive – Possible data exfiltration from a suspicious IP address – Suspicious quantity of downloaded archive files

		Microsoft Defender for Cloud <i>Defender for Resource Manager</i> Suspicious invocation of a high-risk ‘Data Collection’ operation detected (Preview) <i>Defender for Storage</i> – 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 data access activity – Unusual amount of data extracted from a sensitive blob container – Unusual number of blobs extracted from a sensitive blob container – Potential data exfiltration detected – Access from a suspicious IP address
--	--	--

This research is provided by Microsoft Defender Security Research with contributions from Adi Segal, Karam Abu Hanna, Alon Marom, and members of Microsoft Threat Intelligence.

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](#).

Review our documentation to learn more about our real-time protection capabilities and see how to enable them within your organization.

[How Microsoft discovers and mitigates evolving attacks against AI guardrails](#)

Learn more about [securing Copilot Studio agents with Microsoft Defender](#)

Evaluate your AI readiness with our latest [Zero Trust for AI workshop](#).

Learn more about [Protect your agents in real-time during runtime \(Preview\)](#)

Explore [how to build and customize agents with Copilot Studio Agent Builder](#)

[Microsoft 365 Copilot AI security documentation](#)

Source: <https://www.microsoft.com/en-us/security/blog/2026/05/18/storm-2949-turned-compromised-identity-into-cloud-wide-breach/>