VECTRA

Vectra AI Cyber Threat Intelligence Guide

Introduction

This brief guide will provide you with a clear view of the threat landscape that's emerging in 2024 by focusing on four key security events and the threat actors behind them:

1. Scattered Spider
2. Midnight Blizzard
3. XZ Utils
4. PAN-OS CVE-2024-3400

As we review these attacks and actors, we'll analyze each and how they have made it easier for attackers to breach your hybrid environment. We're seeing new versions of known techniques and new techniques from unknown threats that lead us, as security professionals, to face the facts as they truly are. Attackers WILL:

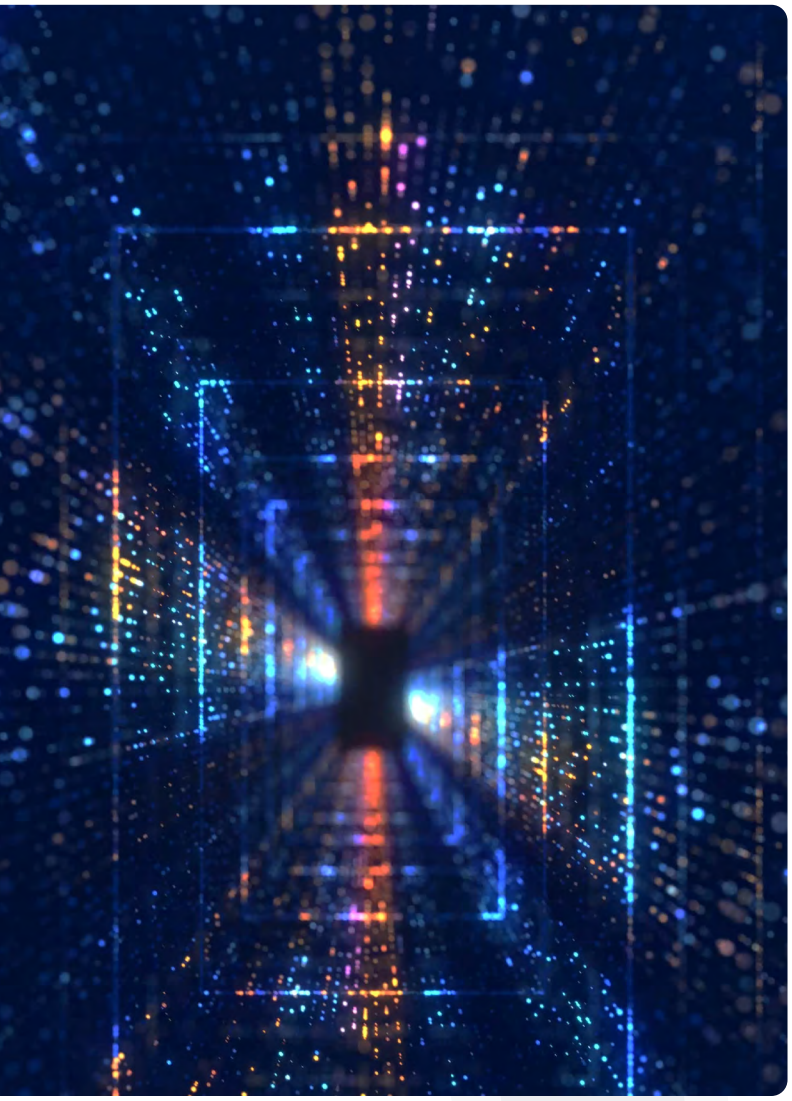
- Gain access to your hybrid network from any source and anyone connected to it
- Quickly access a user account, turning a hybrid attack into an identity attack
- Use multiple seemingly benign steps to gain access to your network without triggering an alert
- Hide within complexity
- Move laterally through your system quickly
- Escalate privileges for data damage, denial, exfiltration, ransom, and more

Table of contents

In some cases, such as **Scattered Spider**, a nasty ransomware group is leveraging creative ways to bypass MFA and get in through cloud identities to then hold your data ransom and encrypt your systems, making it difficult for your organization to function at all.

In the case of **Midnight Blizzard**, we're seeing a real-world example of a no exploit, no Zero-Day, no malware attack that used Microsoft's own tools just as they were designed to function, to breach Microsoft via a series of moves that Microsoft was unable to detect either pre- or post-compromise by mimicking normal behavioral signatures, gaining access within their hybrid environment from any point of contact.

In the **XZ Utils** backdoor attack, unauthorized users with a specific encryption key can inject arbitrary code via an SSH login certificate, which poses a high-risk threat to any organization using that software.



The **PAN-OS CVE – 2024 – 3400** exploit is an unauthenticated remote code execution (RCE) vulnerability, a pure Zero-Day Firewall device running Global Protect VPN with a CVE risk score of 10 out of 10. It enables attackers to execute any kind of code they want on the Palo Alto Firewall without being authenticated, which makes both the probability of more of these attacks and the risks they pose, extremely high.

The events examined in this guide also demonstrate that one way or another, attackers can and will gain access to your hybrid environment, regardless of your defenses, raising both risks and the costs of breaches. Recent statistics confirm this new reality:

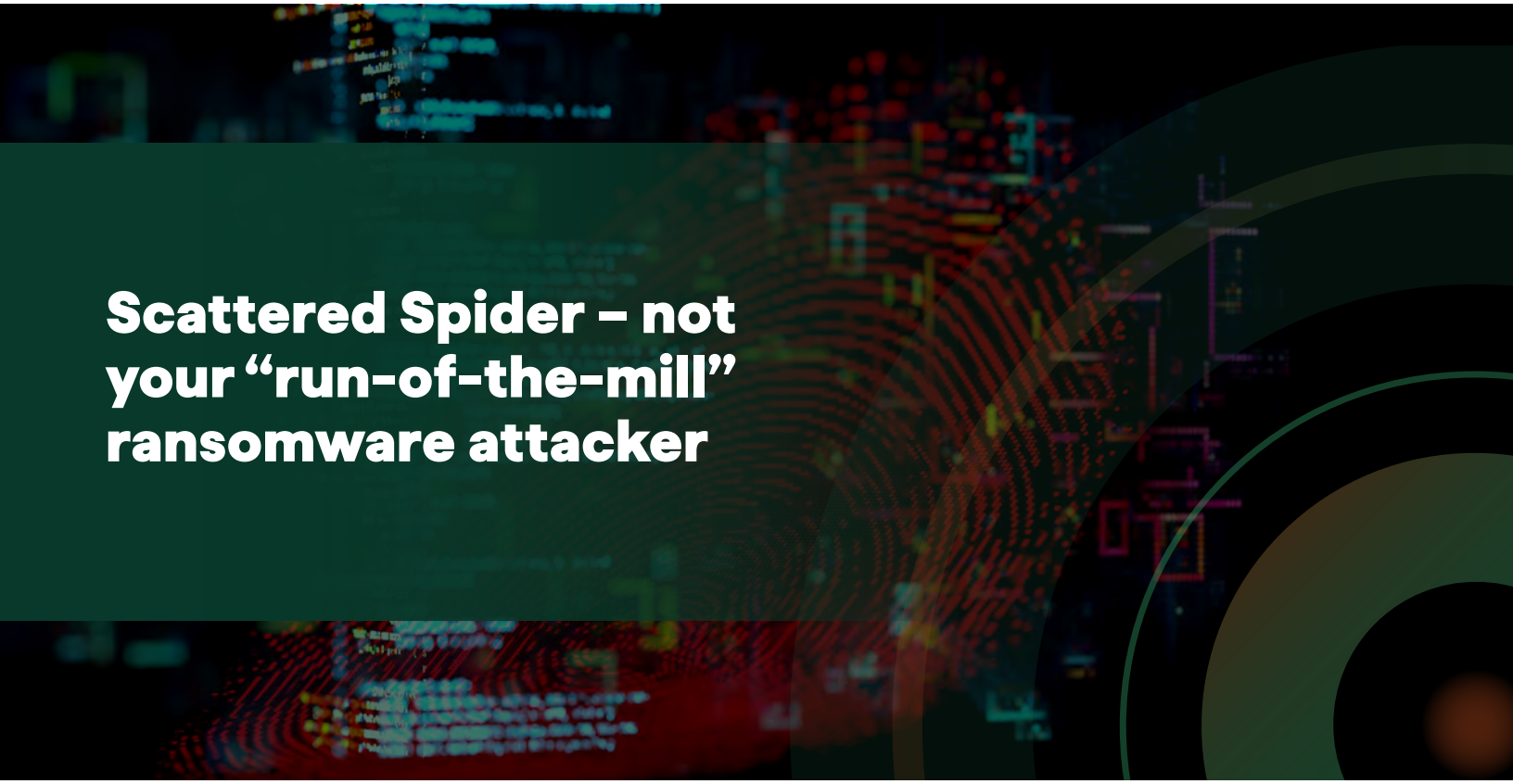
- 90% of breached organizations had MFA in place
- According to IBM, the average cost of a breach associated with an identity breach range from USD 4.55M to USD 4.76M''
- MFA protects 99.99% of accounts, as per Microsoft, but attackers can still bypass it, posing a 0.01% or higher risk
- Attacks are growing in sophistication and risk ratings

And yet, that is just part of the emerging threat picture. On the cyber defense side, the staggering and continually rising number of alerts overwhelms SOC teams. What's more, attackers' specific methods and impacts on security threat detection and response platforms make the following facts crystal clear:

- You're going to be breached
- You're not going to know when, where, or how for several months
- Your team may not detect the breach with your current tools
- You must deploy post-compromise detection

Consequently, the SOC team will continue to pay the price in terms of loss of competence and confidence, as they face an uphill battle of defense.

Most importantly, we'll show you the way forward in defending your environment against these and other sophisticated hybrid attacks to eliminate the threat before these sophisticated attacks impact your hybrid environment.



Scattered Spider – not your "run-of-the-mill" ransomware attacker

Let's start by focusing on a very nasty ransomware group known as Scattered Spider, which is a collection of attackers that are known by several names, such as Starfraud, UNC3944, Scatter Swine, Octo Tempest, and Muddled Libra. For clarity, we'll use just Scattered Spider in this guide.

Scattered Spider is getting a lot of attention for several good reasons. For one, they have developed highly successful and reproducible identity-based attacks that leverage creative ways to bypass MFA and get in through cloud identities. After accessing your environment, they transition into living-off-the-land attacks across the enterprise, extending to the cloud, network, and everything else in your hybrid environment. But fundamentally, the attack starts with identity because whatever identity touches, they can interact with, move through, and ultimately launch their ransomware attacks.

Once it gains access, the Scattered Spider attack mechanism is focused on denial of service and extortion for stolen data. But it's not a typical DDoS type of denial of service, where an attacker breaks things down in your network. Instead, a Scattered Spider attack shuts down your operations from the inside of the network by encrypting systems and blocking access, thereby making it difficult if not impossible to do business. At the same time, they take your data off-site and demand high ransom payments, threaten to put it out into the world, or even use it against you. It's a full extortion attack on top of the denial of service attack.

The Scattered Spider-ALPHV Blackcat ransomware axis

Scattered Spider is often mentioned in the same breath as ALPHV Blackcat. Both fall into the ransomware-as-a-service (RaaS) affiliate category. However, there are several key differences between the two.

For instance, ALPHV Blackcat is notorious in its own space, with a well-known history of launching both traditional and hybrid attacks against cloud enterprise environments, with some of the highest ransomware demands seen in the current landscape. Also, ALPHV Blackcat is a RaaS provider, while Scattered Spider is a RaaS affiliate. In fact, ALPHV Blackcat has been the source of many of Scattered Spider's tools and techniques.

That said, Scattered Spider also uses their tools, third-party identity access brokers, off-the-shelf control points, and technologies from other providers. But Scattered Spider and ALPHV Blackcat have a history of launching traditional and hybrid attacks on cloud enterprise environments, and excel at finding ways to target data wherever it's the most valuable. That means there is no single area of the network or cloud that you can defend less than another. They go after the entire hybrid environment in a highly coordinated way, which includes connected devices and users. Scattered Spider drives the attacks and ALPHV Blackcat drives the negotiations and manages the leaked data. Scattered Spider then pays ALPHV Blackcat a fee to use their ransomware and to its other third-party providers.

Although Scattered Spider and ALPHV Blackcat are separate entities, they're closely connected. This chart gives a clear breakdown between the two.

RAAS Affiliate	RAAS Provider	Initial Access Brokers
Scatter Spider	**ALPHV Blackcat**	
Pays to use the ransomware Agrees on a service fee per collected ransom	Recruits affiliates on forums	
Pays for initial access Targets and executes attacks against victims	Gives affiliates access to a "build your own ransomware package" and "Command and Control" dashboard to track the package	Sells access to target victims
Communicates with the victim via chat portals or other communication channels	Sets up a victim payment portal and "Assists" with victim negotiations	
Manages decryption keys	Manages a dedicated leak site	

Changing relationship between Scattered Spider and ALPHV Blackcat

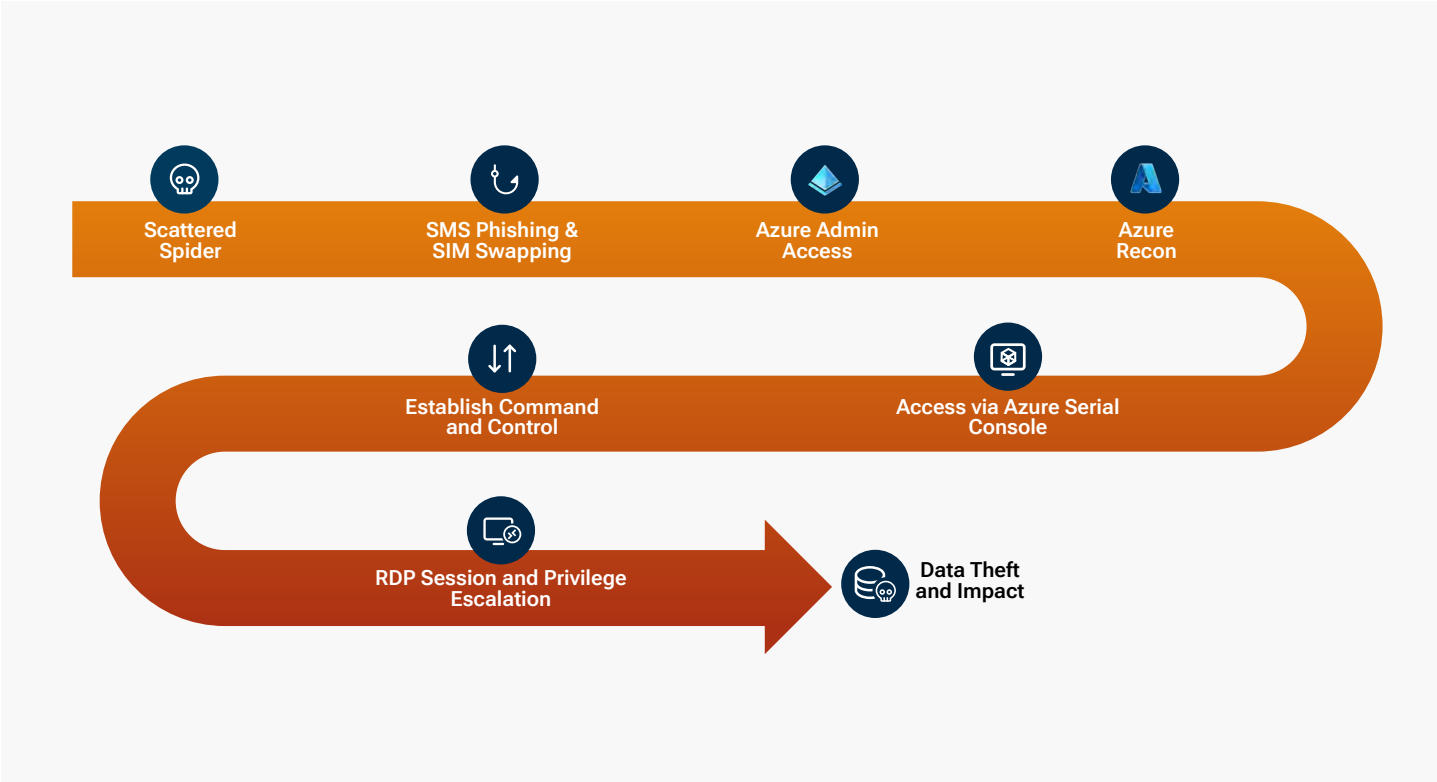
However, recent events may soon change the Scattered Spider-ALPHV Blackcat relationship in ways we don't yet know. With FBI operations taking down the ALPHV Blackcat website, ALPHV Blackcat disappeared from the market, at least in name. Normally, what happens in these cases is the attacker ends up rebranding, and the tools come out a different way. But soon thereafter, ALPHV Blackcat continued to act as a ransomware provider and was immediately involved with the Change Healthcare incident.

But in that attack, there was a change in ALPHV Blackcat's behavior with regard to the ransom payment. Once the ransom was paid, instead of taking their fee and then transferring the remaining money to the affiliate as usual, ALPHV Blackcat kept the money it owed to the affiliate and went underground. That's why some expect there to be changes in the Scattered Spider-ALPHV Blackcat relationship regarding the specific tools they use and other IOCs.

Nonetheless, there are plenty of documented examples of Scattered Spider's attack behavior that enable the Vectra AI Platform, which focuses on behavioral patterns, to understand the level of sophistication involved, detect any RaaS attacker behavior, and identify those kinds of attacks very early in their processes post-compromise.

A look at a Scattered Spider cloud-centric identity-based attack

In the example below, Scattered Spider starts with an identity compromise using SMS phishing and SIM swapping. This technique lets Scattered Spider bypass MFA and log into an identity to gain access and begin the attack. There are a few other ways they've done this, but this way has worked well for this group and other active threat actors out there.



SMS phishing can bypass MFA prevention, allowing the attacker access to sign on. They get access to an Azure admin account and immediately pivot and start interacting with Azure and discovering the landscape. Next, they're interacting with different tools in the Azure platform as a Service (PaaS) and start putting tools onto available endpoints, such as VMS, that were deployed in Azure, to effectively do recon from the PaaS angle.

Next, the attacker mapped out a path from the actual IaaS itself. Because there's a functionality in Azure (really in any Microsoft kind of deployment of VMs) called a serial console, they were able to run arbitrary code on the AVM. It's essentially a remote management tool. This enables the attacker to run commands like a reverse SSH from any of the endpoints that are VMs in that Azure cloud.

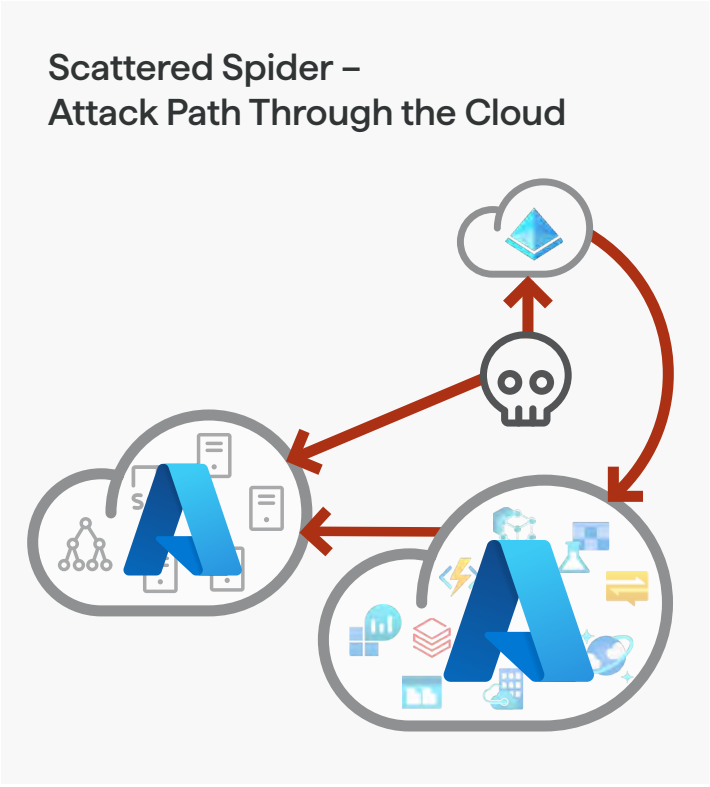
What started as an identity-based compromise has now moved into the cloud, and then moved from the PaaS component into the IaaS piece, from which their traditional ransomware functionality comes into play. C2 is deployed and lateral movement over RDP connections occurs. Then privilege escalation and traditional tactics that we see in ransomware actors are used to move through your network. All of this behavior is now in play after a quite nuanced cloud infiltration. This ultimately leads the attacker to commit data theft and leave an even greater impact as a ransomware attack.

The Vectra AI Platform analysis

There are indications of the sign-on being suspicious and ways to detect this early.

Because of our focus on behavioral coverage, the Vectra AI Platform detects any of the core techniques that any ransomware provider would be using, as well as any that Scattered Spider as a group has developed on their own. This includes other vendors in the market for these ransomware tools, absent the core functionality.

Here's a more zoomed-out view of just one of Scattered Spider's attack techniques. The skull in the middle represents Scattered Spider. They hit the Entra ID identity through SMS phishing, then pivot into Azure PaaS and use that PaaS to connect directly to Azure IaaS, where they can deploy the command and control (C2) that brought Scattered Spider into the IaaS. This is how Scattered Spider can span multiple attack surfaces throughout their attack with minimal prevention that can stop them. This is the type of attack technique–but not the only one–that is used by Scattered Spider.



Scattered Spider is highly effective at accessing and abusing identity

The diagram below shows different views of the documented cloud identity techniques. There is the traditional MITRE view of the identity techniques Scattered Spider has available to them in the cloud: SIM swap, MFA bombing, voice phishing, etc. They register persistence once they've bypassed MFA, which they can do at both device and tenant levels, allowing them to manipulate accounts and start harvesting data. But as we've said, it's not just identity tactics at play, they span the gamut of the attacker's space and the target firm's environment.

TA0001: Initial Access	TA0003: Persistence	TA0004: Privilege Escalation	TA0005: Defense Evasion	TA0006: Credential Access	TA0009: Collection
Phishing: Spearphishing Voice T1566.004	Account Manipulation: Additional Cloud Credential T1098.001	Account Manipulation: Additional Cloud Credentials T1098.001	Impersonation T1656	Multi-Factor Authentication Request Generation T1621	Data from Cloud Storage T1530
Valid Accounts: Cloud Accounts T1078.004	Account Manipulation: Additional Cloud Roles T1098.003	Account Manipulation: Additional Cloud Roles T1098.003	Valid Accounts: Cloud Accounts T1078.004		Data from Information Repositories: SharePoint T1213.002
	Account Manipulation: Device Registration T1098.005	Account Manipulation: Device Registration T1098.005			
		Valid Accounts: Cloud Accounts T1078.004			
		Domain Policy Modification: Domain Trust Modification T1484.002			

Full MITRE ATT&CK view of Scattered Spider

Key arcs to note here center around the live-off-the-land techniques and then using identity to pivot between the areas they're interested in and enabling persistence. Scattered Spider runs the standard ransomware playbook that we're familiar with, like using C2 channels, often with commercial remote access tools or third-party software. It's worth noting that they're not developing crazy payloads; they don't need to. Instead, they're using an off-the-shelf Ransomware-as-a-Service model with tools that have been validated in other places.

TA0001: Initial Access	TA0002: Execution	TA0003: Persistence	TA0004: Privilege Escalation	TA0005: Defense Evasion	TA0006: Credential Access	TA0007: Discovery	TA0008: Lateral Movement	TA0009: Collection	TA0010: Exfiltration	TA0011: Command and Control	TA0040: Impact
Phishing T1566	Service Execution T1648	Create Account T1136	Domain Policy Modification T1484	Modify Cloud Compute Infrastructure T1578	Forge Web Credentials T1606	File and Directory Discovery T1083	Remote Services T1021	Data from Information Repositories T1213	Exfiltration Over Web Services T1567	Remote Access Software T1219	Data Encrypted for Impact T1486
Valid Accounts T1078	Windows Management Instrumentation T11047	Multi-Factor Authentication T1556	Valid Accounts T1078	Impersonation T1656	Multi-Factor Authentication Request Generation T1621	Remote System Discovery T1018		Data Staged T1074		Ingress Tool Transfer T1105	Financial Theft T1657
Exploit Public Facing Application T1190	User Execution T1204	Valid Accounts T1078	Exploitation for Privilege Escalation T1068	Valid Accounts T1078	Unsecured Credentials T1552	Steal Web Session Cookie T1539		Email Collection T1114		Protocol Tunneling T1572	
External Remote Services T1133		Account Manipulation T1098			OS Credential Dumping T1003	Network Service Discovery T1046		Data from Cloud Storage T1530		Proxy T1090	
Trusted Relationship T1199						Permission Groups Discovery T1069				Web Services T1102	
						Cloud Service Dashboard T1538					
						Browser Information Discovery T1217					

They're running things like Mimikatz, using lateral movement via RDP just like any other ransomware actor, and AD, where they're going after Impact/Exfil. That's Scattered Spider's playbook in a nutshell.

Vectra AI's MITRE ATT&CK cloud identity coverage for Scattered Spider

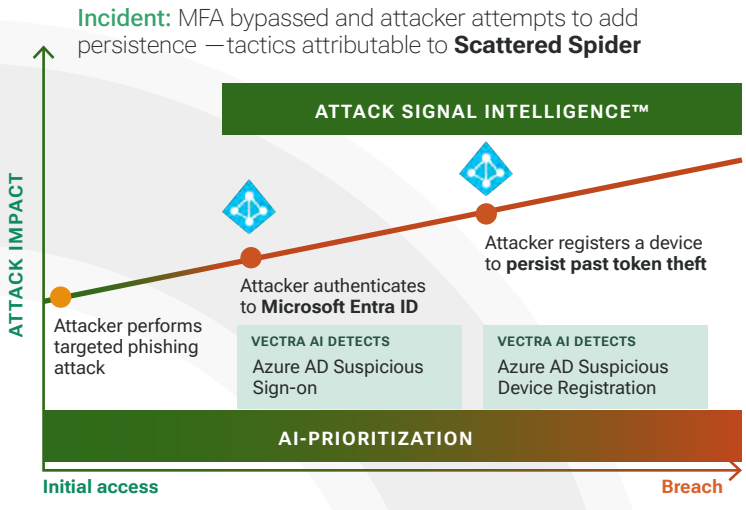
Vectra AI has been monitoring Scattered Spider and other groups in RaaS activity for months and our coverage has been effective against these types of attacks. For perspective, below is the identity matrix that relates to our coverage.

TA0001: Initial Access	TA0003: Persistence	TA0004: Privilege Escalation	TA0005: Defense Evasion	TA0006: Credential Access	TA0009: Collection
Phishing: Spearphishing Voice T1566.004	Account Manipulation: Additional Cloud Credential T1098.001	Account Manipulation: Additional Cloud Credentials T1098.001	Impersonation T1656	Multi-Factor Authentication Request Generation T1621	Data from Cloud Storage T1530
Valid Accounts: Cloud Accounts T1078.004	Account Manipulation: Additional Cloud Roles T1098.003	Account Manipulation: Additional Cloud Roles T1098.003	Valid Accounts: Cloud Accounts T1078.004		Data from Information Repositories: SharePoint T1213.002
	Account Manipulation: Device Registration T1098.005	Account Manipulation: Device Registration T1098.005			
		Valid Accounts: Cloud Accounts T1078.004			
		Domain Policy Modification: Domain Trust Modification T1484.002			

The Vectra AI Platform delivers full coverage from the cloud side for identity. This includes everything from the registration to the domain trust manipulation to the access events, etc. The Vectra AI Platform provides coverage that pinpoints attackers exactly like Scattered Spider and their behavior early in post-compromise. Below are some examples of relevant cases to this attack space to give you an idea of the Vectra AI Platform capabilities.

Scattered Spider attack example 1

This is an anonymized attack that shows the techniques used in the initial events of a Scattered Spider attack.



Attacker stopped in the first 5 minutes

Vectra findings:

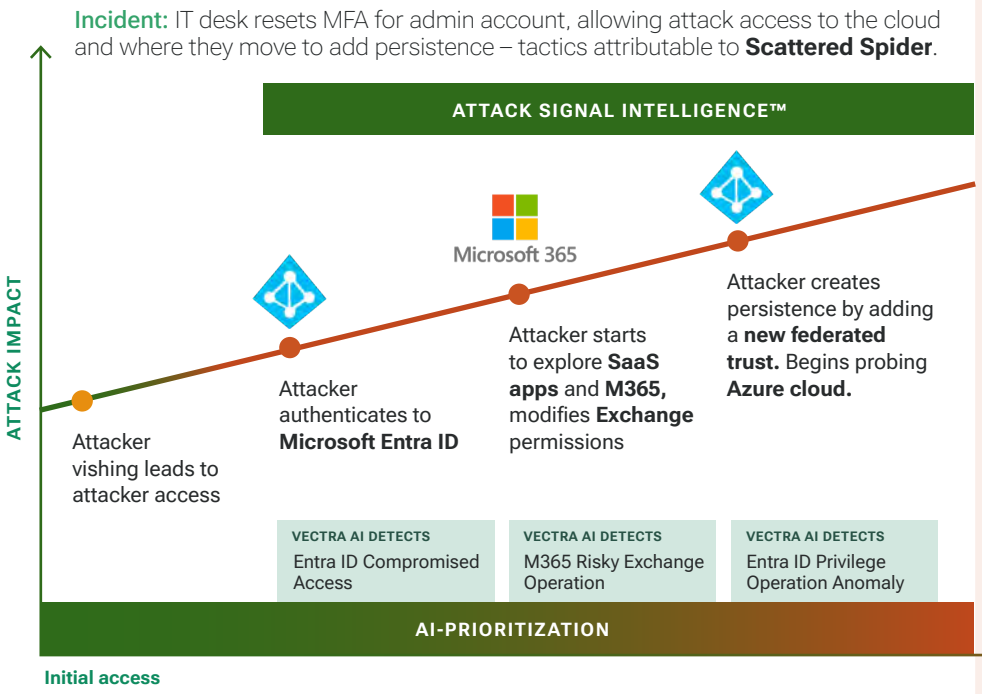
- Attacker sign-in over proxy connection
- Attacker registered a new device in the first 5 minutes of the compromise

The attacker appears to perform targeted phishing. It may have been a phishing kit that enabled them to bypass MFA, which is out in the market and is typical of Scattered Spider. The attacker signed in through the Entra ID account and was moving to add persistence. That means they would register a device because if you steal just a token, you only have temporary access to it.

This was an actual case that was identified by the Vectra AI MDR service. It was seen in the first five minutes of the attack and shut down. The attacker would've had many opportunities to cause havoc if this had been successful, but because of the visibility provided by the Vectra AI Platform, we were able to stop this incident.

Scattered Spider attack example 2

Below is a second, more nuanced example of an attack.



Attacker stopped before impact

Vectra AI findings:

- Malicious **Microsoft Entra ID** access from Azure VM in the user's region
- Adding **Exchange** persistence for delegate access
- Interactions with **Sharepoint** shares
- Adding **Microsoft Entra ID** federated trust for persistence
- Additional **SaaS apps accessed**

This attack involved an attacker driving through voice phishing to prompt a password reset. There are other reports of this being a Scattered Spider technique, but even when that happens, there are aspects of that sign-in that are identifiable as suspicious. There's an opportunity to alert on that axis event, but in all these cases, attackers don't do just one thing but a series of behaviors that are detectable with the right visibility, and Scattered Spider is no different.

You'll notice access into Exchange, moving through SaaS apps as they move into the M365, and then start to put redundant access at the federated trust level with an admin-level account, and then begin probing into Azure. This organization stopped it because it had the right visibility. You can see everything from anonymous access, into an Azure VMA technique to evade detection, then Exchange persistence delegating access to be able to manipulate accounts in that domain, interactions with SharePoint, and adding new federated trust for persistence. In this case, the broader Azure cloud was available to the attacker, but this organization had the right tools in place so the attacker could be stopped before the impact happened.

Full MITRE ATT&CK view of Scattered Spider with the Vectra AI Platform

Vectra AI provides cloud, network, and identity coverage for Scattered Spider (and similar attacks). Highlighted in green are the areas that are covered by the Vectra AI Platform.

TA0001: Initial Access	TA0002: Execution	TA0003: Persistence	TA0004: Privilege Escalation	TA0005: Defense Evasion	TA0006: Credential Access	TA0007: Discovery	TA0008: Lateral Movement	TA0009: Collection	TA0010: Exfiltration	TA0011: Command and Control	TA0040: Impact
Phishing T1566	Service Execution T1648	Create Account T1136	Domain Policy Modification T1484	Modify Cloud Compute Infrastructure T1578	Forge Web Credentials T1606	File and Directory Discovery T1083	Remote Services T1021	Data from Information Repositories T1213	Exfiltration Over Web Services T1567	Remote Access Software T1219	Data Encrypted for Impact T1486
Valid Accounts T1078	Windows Management Instrumentation T11047	Multi-Factor Authentication T1556	Valid Accounts T1078	Impersonation T1656	Multi-Factor Authentication Request Generation T1621	Remote System Discovery T1018		Data Staged T1074		Ingress Tool Transfer T1105	Financial Theft T1657
Exploit Public Facing Application T1190	User Execution T1204	Valid Accounts T1078	Exploitation for Privilege Escalation T1068	Valid Accounts T1078	Unsecured Credentials T1552	Steal Web Session Cookie T1539		Email Collection T1114		Protocol Tunneling T1572	
External Remote Services T1133		Account Manipulation T1098			OS Credential Dumping T1003	Network Service Discovery T1046		Data from Cloud Storage T1530		Proxy T1090	
Trusted Relationship T1199						Permission Groups Discovery T1069				Web Services T1102	
						Cloud Service Dashboard T1538					
						Browser Information Discovery T1217					

Given the diversity of this kind of attacker, it's vital to have both visibility from the identity stage and the cloud stage to the actual kind of network components, as well as clarity, once you have that coverage.

The Vectra AI Platform's signal contextualization and prioritization makes all the difference.

The Vectra AI Platform coordinates disparate signals into one clear signal and gives clarity to any of those pivots happening between the network and the cloud. Then, from a response or control side, you can leverage either the Azure AD integrations to disable the accounts, the AD on the network integration to disable the accounts, EDR integrations to disable the endpoint, or any of the broader vector response functionalities to take immediate action.

The Midnight Blizzard Attack on Microsoft

Midnight Blizzard's identity attack against Microsoft is an extremely interesting attack that provides a real-world example of a no exploit, no Zero-Day, no malware attack. That alone is noteworthy. But the fact that such an attack was successful against the most used, visible, and valuable IT company in the world makes it doubly troubling.

After all, if such an attack was successful against Microsoft and its formidable resources, what organization can defend itself against such attacks? As security professionals, the key question is, "Would your defenses have detected the threat in your environment that Microsoft could not?"

The Microsoft product was used exactly as designed in the attack

The fact that Microsoft's products were used as designed reminds us of live-off-the-land techniques that we see occurring in the network. The attack followed that template, except it was in an M365 and Entra ID environment. Let's take a deeper look to better understand who the attackers were and what they used to execute such a successful and undetected breach.

Who Is Midnight Blizzard?

Like other attack groups, Midnight Blizzard is known by many names you may heard of before, such as APT29, Cozy Bear, NOBELIUM, The Dukes, Dark Halo, and UNC2454, among others. They're part of Russia's Foreign Intelligence Services and target government, technology, and research supply chains. They have been in the news often over the years for their multiple attacks–one of their most well-known was the recent past was Solar Winds, which involved Microsoft O365, with APT29 (Midnight Blizzard) determined to be behind the attack.

Like other attacks, Microsoft wasn't the only target of Midnight Blizzard. Microsoft authorities said that other companies were targeted too, but didn't provide details. However, some of the victims included Denmark Bank and HPE among others. They were compromised by the same group with the same techniques that have proven to be very effective and difficult to defend against.

The timing of the attack

The time to detection was about 1.5 months. On January 19, 2024, Microsoft announced the breach, saying it occurred in late November 2023, but was only detected on January 12. Microsoft analysts were reviewing Exchange Web Services (EWS) logs and found some abnormal activities. The attack exclusively targeted M365 and Entra ID and, as far as we know, there was no pivot from those targets.

There was further Microsoft communication on January 25, which gave us more details about the attack, but there remain a lot of unknowns, so we'll have to connect some of the different dots in this attack.

Microsoft's breach communications and disclosures

In their public statements, Microsoft admitted that "**The attack was not the result of a vulnerability in Microsoft products or services.**" In other words, the attacker didn't misuse the product, but rather, used it as it was designed to be used, yet it allowed the breach in the Microsoft network.

The second quote from Microsoft was also very telling: "**This attack does highlight the continued risk posed to all organizations from well-sourced nation-state threat actors like Midnight Blizzard.**"

Midnight Blizzard is an extremely sophisticated group, well-resourced with good OpSec, making them a highly dangerous threat, even to organizations at the very top of the cybersecurity food chain.

Initial access gained via password spraying

We know that password spraying is more effective against accounts without MFA protections. An interesting side note is that at an industry event earlier in 2023, Microsoft announced that only 38% of monthly active users (MAUs) have MFA turned on, which meant that 62% of MAUs were not protected by MFA. It's surprising how many users still don't have it turned on, but also, how willing Microsoft was to disclose that fact.

The first step in password spraying is to find a valid account in a M365 tenant. Enumerating a valid account is easy and doesn't leave any trace. For instance, if you build a list to test toward a tenant, you will not have any log generated on the Microsoft side. That means attackers can enumerate as much as they want and build a valid account that will enable them to perform a password-spraying attack. And yes, the account compromised in this attack had no MFA.

Of course, password spraying tools are easily obtained on GitHub, such as O365 Spray, Spray365, or MAAD-AF. They use a low and slow type of attack to avoid the lockdown mechanisms designed to block brute force attacks, which typically default to a 1-minute lockdown after 10 failed password attempts within a 5-minute window. This helps attackers get around this lockdown configuration and avoid any type of detection. It's how the user account, that test account, got compromised in the first place.

Defense evasion via residential proxy network

In terms of defense evasion, Microsoft mentioned the use of a residential proxy network as part of the attack, which allows threat attackers to be as close as possible to actual users. The attackers chose a localization that prevented certain detection triggers from Azure AD Identity Protection that a defender would have operational, such as suspicion sign-in or atypical travel. It also prevented any type of IOC detection, because the attacker kept changing IP addresses, actually using residential IP to keep changing locations, making traditional detection ineffective.

Plus, all the password spraying tools mentioned above support proxy, so it's easy to set up. Among them are IP Royal, OxyLabs, BrightData, Infatica, and others, that attackers can use. Below is a screenshot of an example of IPRoyal that we use for testing internally to replicate the attack.

Replicating the attack

The configuration takes only a couple of minutes to setup. The attacker specifies the country, the region, or regions (all over the world) cities, and states, and specifies the type of proxy wanted. Then the attacker provides a username, and password, and then directly, the proxy comes online to use in the tool. That's it.

In terms of pricing, money is much less of a problem for nation-state actors. But for only $1.75, you can do 1GB worth of brute force! It's also interesting to point out just how easy and cheap those residential proxy networks are.

Exploring the demo setup of the proxy in IPRoyal Residence

This proxy setup, which will be similar to the attacker setup, takes just a few minutes to set up as well. Because it's Microsoft, the country to specify is the United States and the city is Seattle. Then the attacker just copies and pastes the username and password, or just that proxy command which combines the user name, password, and the IP configuration of the proxy. Again, very simple.

In this example, the way Spray365 works, the attacker has a list of valid usernames and a list of passwords that it wants to test with them. The attacker then executes the plan with the Proxy Configuration. With the proxy, again, it's slow going through the list but can be accelerated. The attacker tries all the combinations, not just different username-password combinations, but user agent, device, and things like that.

A few clarifications of terms

Before we move to the next step of the attack, there are some definition clarifications needed. This is a complex environment so it's important to understand before going to the second step of the attack.

Entra ID App Registration (formerly 0Auth App) is an identity you can create within your tenant and it's going to be an app registration, a global registration, of the app across all tenants. When you create an app registration in your app on your tenant, it automatically creates an enterprise application, also known as a service principal. A service principal is a local representation of this app, so when you are in the same tenant, you will end up with one app registration and automatically have an enterprise application created for that app, too, which is the service principal.

Enterprise applications can be owned and foreign, so imagine you publish consent for an app into your tenant, which is the case for tenant two in the slide. As the attacker, you only have the service principal created, so you have a local representation of this app that is foreign, owned by a different tenant, in your tenant. With all the permissions that have been defined on the app registration, you get them consented to by an administrator on a different tenant. Permission can be a predefined role, or it can be API permissions.

API permissions are either dedicated (have to be authenticated to the app as a user and then you can have access to the permissions) or, they're an application type, which means this is the app itself that authenticates, using a clarion secret or a certificate, so the attacker can use those permissions, and it's more like a programmed application.

How the attack was set up

The attack started with two tenants – the test tenant and the target tenant. As the attacker, we'd create a legacy app in the test tenant, that can be multitenant, according to Microsoft. Once we create the app, we define permission for the app. According to Microsoft, the permission was well elevated, so application permission and what we see in this application is `Read.Write.All` as well as `Application.Role.Assignment`.

Continuing the permission process as the attacker, this app initially had elevated permission in the target tenant, so we granted the access. We then get the URL to share this application with a different tenant. We copy that and then move to the target tenant, which is the production tenant in the Microsoft environment in that example, and just copy-paste the consent URL and the attacker will have consented to this app.

That was a simplified version of the initial configuration, but that configuration was already in place between that test tenant and that production tenant for who-knows-how-long before Microsoft was breached. Checking the demo legacy app, it's an enterprise application that's a service principal, and the permission has been granted to that service principal on the production tenant, so the permission is exactly the same as the test tenant.

Now let's use some representation to make this attack process easier to understand. In the beginning, there is a test tenant, a user account that has been compromised, and a legacy test app that the user has access to, although it's not clear from Microsoft what permission the user/attacker has. But he has access to that application – maybe he owns it, maybe he has some application administrator role within the tenant, but essentially, he was able to control that app, which means he was able to generate a secret for that app.

The app was already consented to within the production tenant–it could be a question of months or even years that this app was there–and it is a service principal with elevated permission that we've seen in the production tenant. We don't know exactly what the permission level was. It could be different combinations, but we can assume that the main role that was available to that app was AppRole Assignment `Read.Write.All` from the graphic PI. We'll see exactly why that's the best assumption in terms of permission that this app has. The other thing we can assume is that the app had an application `Read.Write.All`, but it could also be other roles, it could be Tier 2 super roles, probably not global administrator, but at least elevated.

Can there be a suspicious sign-on event for an application authenticating itself (certificate or secret)? *Not without additional diagnostic settings!*

You might wonder why, when the user/attacker was able to generate a secret and you authenticate with that secret as the app to the target tenant, there is no suspicious sign-in event for this authentication.

If there is a certification or a secret, usually there is a sign-in event, just like you have for users. An interesting finding is that there is no such an event without additional diagnostic settings turned on. But if you have not enabled to log this type of sign-in event in your tenant, you won't have anything. You will have a user sign-in event, but you won't have a service principal sign-in event. The reason for that is probably because you would perform a lot of authentications per day since there are a lot of logs, which is a lot of work. So even if you wanted to enable them, you would have to send them to Loginatic workspace, HeavenHub, or a storage account. You won't be able to send them directly to your SIEM, for example, and is not even something you can collect in-house.

Next step? *Gaining access to Microsoft*

The threat actors used the app that was privileged on the production/target tenant to create a new user.

We can assume that of those permissions listed here, none allow the service principal to create new users. So, we imagine/assume that the service principal was using existing permission to elevate user access. The attacker elevates himself into a new role that will allow him to create new users. In that case, the most likely permission that would be granted to create new users would be the `Directory.ReadWrite.All` on the graphic PI. So, that's how a privilege escalation works – you can just assign yourself additional permission.

Normally, when you assign a new permission to any principals or users–it could be a service principal–you have to assign it and then you have to grant it. The grant must be performed by an administrator, so it's a two-step process. You have to make sure that everything that is assigned is granted by an administrator, which is the second step. *The key takeaway from this is that the consent process can be bypassed. There are very dangerous permissions that could be assigned and used that can bypass the consent process.*

In that case, the `AppRoleAssignment.ReadWrite.All` will allow you to do just that. A user could grant new permissions to any principals and they will be automatically consented, in a one-step process to bypass the whole thing. That makes this extremely dangerous and is also why it's likely permission that this app had to have to make this attack possible.

New permission, elevated access to user accounts

Once you have that new permission, the service principal will be able to create a user account and maybe elevate the access of that user account.

According to Microsoft, the next step the attacker took was to create a new app within the test tenant. This new app got consented by the user account that was created by the attacker in the production tenant Microsoft, allowing a different app within the production tenant to be completely controlled by the attacker.

The last step of the attack was to `Grant.Role_full_access_as_app` from O365 Exchange Online API and this was `full.access.as.app` permission granted to the new service principal that was created by the threat actors. With the permission that was granted to that service principal, you can access any mailbox. That's what led to the breach.

Hiding behind complexity

Why did the attacker take so many hops between users to service principals? If that service principal was able to assign any permission to itself, it could have easily granted itself permission required to access those mailboxes. The assumption is that the attacker was trying to avoid raising an alert by directly escalating privilege to that service principal. To do that, the attacker covers his path with layers of complexity.

Now that we've seen how the attack rolled out, the different phases, and filled in some gaps on what the attacker may well have done, how could you have prevented or even detected it?

What can you do for prevention?

The first preventative step is to audit. It's very time-consuming, so you might want to automate the process with available tools. However, the breach would not have been possible if that configuration was not in place. Someone consented to this app with those permissions in the production tenant. Normally, it would have fired detections right away.

Prevention is important but often fails. Plus it's time-consuming, expensive, difficult, and requires the right skillset. That's why you should rely on strong post-compromise detection and response capabilities. It's faster and will detect this type of compromise before it becomes a breach.

Detect the breach as early as possible. Post-compromise detection capabilities are extremely valuable in this case. You would have known when it was consented and determined why the application had so much permission, so this breach could have been prevented.

Visibility is key. Accessing the logs from Microsoft, which are complex, and collecting them on different APIs, requires visibility, which is key to having strong detection program capabilities.

You can configure Microsoft Entra ID to not allow consent to new App, but only administrators. And for known MFA users, you can still use a conditional access policy. You'll need a P1 or P2 license, but then you can rely on things like trusted IP and trusted device to still limit who can access and authenticate

Microsoft–and the rest of the world–now knows that it can be breached through its own products, which highlights just how difficult it is to detect such an attack. Layers of complexity make such attacks possible and difficult to detect.

Would your tools be able to detect such an attack? It's a valid question to ask yourself.

There are multiple attack paths. These attackers chose a complex path to avoid any type of detection from Microsoft, making it hard to follow their path and the origin of the breach.

Overall, complexity plays a big part in the vulnerability aspect. The Microsoft product is complex, the environment is complex, and the size of Microsoft and the number of tenants they have adds complexity. There's additional complexity not only in principals but with adding Apps, you start opening things to your environment and trust boundaries between the different tenants that you need to control through those permissions.

Defending it all becomes extremely complex quite quickly. Microsoft was blind to the attack. It focused on initial access anomalies such as suspicious sign-on and atypical travel. But the residential proxy didn't detect them because they weren't suspicious–they were coming from within the U.S. and using different IPs, but even with a change of IP, if they're similar and within the U.S. or the same state, they won't trigger anything, so Microsoft had no awareness in this attack sequence.

What the Vectra AI Platform would have detected at every step

The Vectra AI Platform delivers visibility that Microsoft does not. For example, password spraying, even if it's using a residential proxy network, would have triggered the brute force detection on the Vectra AI Platform. Also, when a new secret was created for a legacy test app, on the test tenant, that would have triggered the Redundant Access Creation Detection on the Vectra AI Platform.

Furthermore, the app in the production tenant or target tenant would have triggered two detections on the Vectra AI Platform. The first would have been a privilege operation anomaly when it was attempting the first privilege escalation to be able to create new users. The second would have been Suspicious 0Auth Application, because those additional elevated permissions are assigned to 0Auth App, so that would have triggered detection as well.

Plus, if the user is in the target tenant and assigning high permission privilege elevating him, that would have triggered three different detections on the Vectra AI Platform that would have covered this specific behavior: Privilege Operation Anomaly, Admin Account Creation, and Redundant Access Creation. Note that the new user will not trigger any detection because there is no high-privilege permission assigned to that user yet.

When the app gets high privilege permission, O365 full access as an app, this will trigger the Privilege Operation Anomaly and Suspicious 0Auth Application detections. In fact, all those privileged escalations trigger the same sets of detections.

The Generate Secret for the Malicious App that the threat actor created would have triggered the Redundant Access detection on the Vectra AI Platform. Otherwise, simply creating a new app doesn't fire a new detection.

Want to detect your environment from the threat of such an attack?

The Vectra AI Platform provides the necessary, multiple detections to cover different phases of the attack where Microsoft is completely blind to detect such a threat. Vectra AI delivers the coverage, clarity and control your team needs to confidently operate in a complex threat environment and help secure your organization against breaches from Midnight Blizzard and other sophisticated threat actors.

For more about Midnight Blizzard, watch the Vectra AI Threat Brief video.

The XZ Utils Backdoor

On March 29, 2024, a malicious commit was discovered in the XZ Utils repository. It introduced a backdoor that would compromise systems running the software. In this case, it was a supply chain. XZ Utils is an open-source data compression utility widely used across Linux and Unix-like operating systems. The backdoor allowed unauthorized users with a specific encryption key to inject arbitrary code via an SSH login certificate. The potential impact of the compromised supply chain could be even greater than the SolarWinds breach.

How can I find out if I was exposed to the XZ Utils vulnerability?

In response to the XZ Utils exploit, a project named xzbot has been introduced by the security community. It offers tools for organizations to assess their exposure to this vulnerability, including:

- **honeypot**: fake vulnerable server to detect exploit attempts
- **ed448 patch**: patch liblzma.so to use our own ED448 public key
- **backdoor format**: format of the backdoor payload
- **backdoor demo**: click to trigger the RCE assuming knowledge of the ED448 private key

What can we learn from an enterprise risk security management perspective?

The motives behind this backdoor are still under investigation, but from an enterprise risk security management (ERSM) perspective, it has the sort of long-term strategic planning profile that's typical of a nation-state threat actor. Unknown threats don't just happen; they're carefully thought out and developed. Fortunately, a Microsoft engineer named Andres Freund was troubleshooting what may have initially seemed like a benign performance issue and ended up thwarting a malicious actor.

Whether by luck, curiosity, or both, Freund's actions saved a lot of people a lot of trouble, which led to the need to explore the cultural aspects of our teams.

Strengthen your defenses with tooling and culture

Beyond tools and solutions, you can enhance your safety by rethinking standard decision-making processes. The reality is that XZ Utils Backdoor was an unknown risk, and in enterprise security risk management (ESRM), it's the unknown risks that will sink you.

Unknown risks typically take one of two forms. In the first scenario, they mimic identified risks but mostly lurk under the waterline of routine discovery measures, like icebergs. In the second, unknown risks are like *uncharted reefs*. They're entirely underwater and unique to some aspect of the business — but not entirely foreign to the collective and distributed domain knowledge, expertise, and experience of the workforce.

That's why you need to cultivate a business culture based as much if not more on curiosity than conventional thinking. Encourage your team to continuously evaluate risk management decisions and challenge assumptions. Promote continuous improvement. Seek a holistic risk management approach that combines tooling and human intuition to test and validate across the full chain of protective, detective, response, and recovery activities. Also, encourage collaboration and incentivize people to discover the unknown risks by inquiring, investigating, and hunting beyond otherwise mundane constraints.

Cultural values, *curiosity*, and *collaboration* give your workforce a real shot at identifying and prioritizing risks through their collective domain expertise, knowledge, and experience. This is how successful organizations assemble, identify, and mitigate risks long before any damage is done.

Yes, managing unknown risks is about more than just cultural efforts — *people, processes*, and *technology* all play a role. But cultural enablement not only increases your foundation and agility in the face of modern threats but also empowers you to tap into the full potential of your people and your tooling. That's why culture is critical — your organization's security practice depends on it. Set everyone up for success by prioritizing it.

How does the Vectra AI Platform protect customers from exploits like the XZ Utils Backdoor?

These kinds of incidents should not deter organizations from using open-source software. Supply chain risks are not exclusive to open-source projects; they can also affect commercial software, as seen in the SolarWinds breach. The key to mitigating these risks lies in adopting detection and response technologies, like the Vectra AI Platform, which can identify threats regardless of the exploits used by attackers.

The Vectra AI Platform can quickly identify attackers exploiting these types of backdoors. In incidents where backdoors like the one found in XZ Utils are exploited, Vectra AI's detection capabilities would identify the core sequence of the attacker's progression from remote access, and discovery, to lateral movement, well before the attack could achieve its objectives. The relevant detections related specifically to XZ Utils being exploited include reverse SSH tunnels that would be triggered by Suspicious Remote Access and lateral movement over the SSH protocol detected by Suspicious Admin.

Not sure where to start?

Thousands of SOC analysts and architects use The Vectra AI Platform to detect, prioritize, investigate, and respond to unknown threats across multiple attack surfaces. Take a self-guided tour to see how it works.

PAN-OS CVE-2024-3400 – Detection and Response

On April 10, 2024, a Zero-Day exploit within the Global Protect feature of Palo Alto Networks PAN-OS was discovered by Volexity on one of their customer's firewalls. The exploit, which is known as PAN-OS CVE – 2024– 3400, was an unauthenticated remote code execution (RCE) vulnerability, a pure Zero-Day Firewall device running Global Protect VPN.

A high-risk Zero-Day exploit

This Zero-Day attack has been given a CVE score of 10 out of 10 because it poses critical security risks to organizations of all kinds. It enables attackers to execute any kind of code they want on the Palo Alto Firewall without being authenticated, which makes both the probability of more of these attacks and the risks they pose, extremely high.

We know that the attack impacted multiple PAN-OS versions and has already been actively exploited. Palo Alto has been releasing multiple patches since April 14, but not all have been released nor have all versions of PAN-OS patched yet, but you must patch and deploy solutions as soon as possible. You can find out which patch has been released, which may apply to your PAN-OS version, and the expected release dates for patches at the Palo Alto Networks website.

Many options to identify the exploit and your vulnerability

Fortunately, there are many tools you can use to identify vulnerabilities in Palo Alto devices that have been used by attackers. Showdown is one example, where you can query to every kind of Palo Alto firewall that has Global Protect enabled and that may be vulnerable to the exploit.

You can also use a Palo Alto scanner to determine which version of the OS a specific device is running, so you can know if it's vulnerable. You can also use these tools to figure out if you're in the database, which can tell you if you're vulnerable to the exploit.

Zero-Day and other severe threats are here to stay

This new Zero-Day exploit shouldn't surprise anyone because there have always been Zero-Day attacks and probably always will be. That said, we're seeing a lot of them lately and you need to be prepared to respond quickly to protect your network and data and stay out of the headlines.

For example, just a few weeks ago, there was CVE-2024-3094, which exploited a backdoor into XZ-Utils, which posed a severe threat and was deemed highly critical with a risk rating of 10 out of 10. It gained lots of publicity, including attention in a couple of recent Vectra AI blog posts.

Earlier this year, there was also CVE-2024-1709, the ConnectWise and ScreenConnect Vulnerability. Both had a severity score of 10 as well. The ConnectWise vulnerability allowed attackers to completely bypass authentication and gain unlimited access.

Before that was the CVE-2023-35802 vulnerability, with a CVE score of 9.8, which involved Remote Unauthenticated API Access Vulnerability to Evanti Solutions. This was an extremely critical, high-risk attack as well.

Finally, in July of last year, there was CVE-2023-33308, Critical Fortinet FortiOS, and FortiProxy RCE Vulnerability. Again, given the fact that Fortinet has millions of users in the U.S., it makes it one more highly critical exploit among many.

Expect to be compromised

The takeaway from these and other publicized high-risk attacks in the past few months is that sooner or later, attackers are going to get in. In short, you will be compromised.

At Vectra AI, we assume that will happen and, as we're reminded with this latest exploit, it's unrealistic to think otherwise.

Therefore, it's not enough to be able to identify your exposure to attacks on your perimeter, that is, north-south traffic. You must also have exposure to east-west traffic within your perimeter, in your network, the cloud, and so forth.

Don't let Zero-Day attacks become identity-based attacks

In other words, Zero-Day attacks, including this latest one, quickly turn into identity-based attacks. They can then rapidly escalate into a massive breach incident, which is why avoiding latency in your post-compromise detection and response is such a critical factor.

Fortunately, the Vectra AI Platform is not only able to recognize and track anomalous post-compromise movement, but it automatically identifies, prioritizes, and mitigates these attacks in real time. That means stopping attacks in minutes, not months, with rapid post-compromise detection and response. That means zero latency and next-to-zero false positives.

That's the kind of detection that keeps your environment safe and your organization out of the headlines.

To learn more about how Vectra AI defends against the PAN-OS vulnerability and explain how it can be quickly dispatched in real-time, watch the Vectra AI podcast with host Mark, "Woj" Wojtasiak, VP of Product Marketing at Vectra AI and Fabien Guillot, Director of Technical Marketing.

Conclusion

A new assessment of the threat landscape is needed. In the modern hybrid enterprise, hybrid attacks are rendering traditional security approaches inefficient and ineffective. Hybrid attacks can start with anyone or anything, and move anywhere are any time at speed to disrupt business operations at scale, despite having every preventative measure in place. What's more, since all enterprises are hybrid, all attacks are hybrid attacks.

Thus, the trend toward more sophisticated and dangerous hybrid attacks is undeniably clear. But getting down to brass tacks, so is the reality that intrusion prevention isn't doing its job. But how can it be when your threat surface expands to any connected devices and people? The inevitability, therefore, is that your hybrid environment will be breached in any number of ways by a variety of techniques, both known and unknown, and may well remain undiscovered by your team for months on end.

The answer isn't adding more conventional intrusion prevention tools that fail or low visibility network solutions that have blind spots in both east-west and north-south traffic, so you remain unaware when a breach has occurred. The solution is an integrated signal with threat surface coverage, clarity, and control that identifies, prioritizes and mitigates the most urgent threats across your hybrid surface with AI-driven behavioral analytics for rapid post-compromise detection and response within minutes, not months.

This AI-driven performance not only enhances and adds resilience to your security posture, but also unlocks the value of your SOC team, raising both confidence and competence in your team for better outcomes and morale.

Learn more about the Vectra AI Platform can enhance your security—and your team's confidence and competence—visit www.vectra.ai

About Vectra AI

Vectra AI is the leader in hybrid attack detection, investigation and response. The Vectra AI Platform delivers integrated signal across public cloud, SaaS, identity, and data center networks in a single platform. Vectra AI's patented Attack Signal Intelligence empowers security teams to rapidly detect, prioritize, investigate and stop the most advanced hybrid cyber-attacks. With 35 patents in AI-driven detection and the most vendor references in MITRE D3FEND, organizations worldwide rely on the Vectra AI Platform and MXDR services to move at the speed and scale of hybrid attackers. For more information, visit www.vectra.ai.

For more information please contact us:
Email: info@vectra.ai | vectra.ai