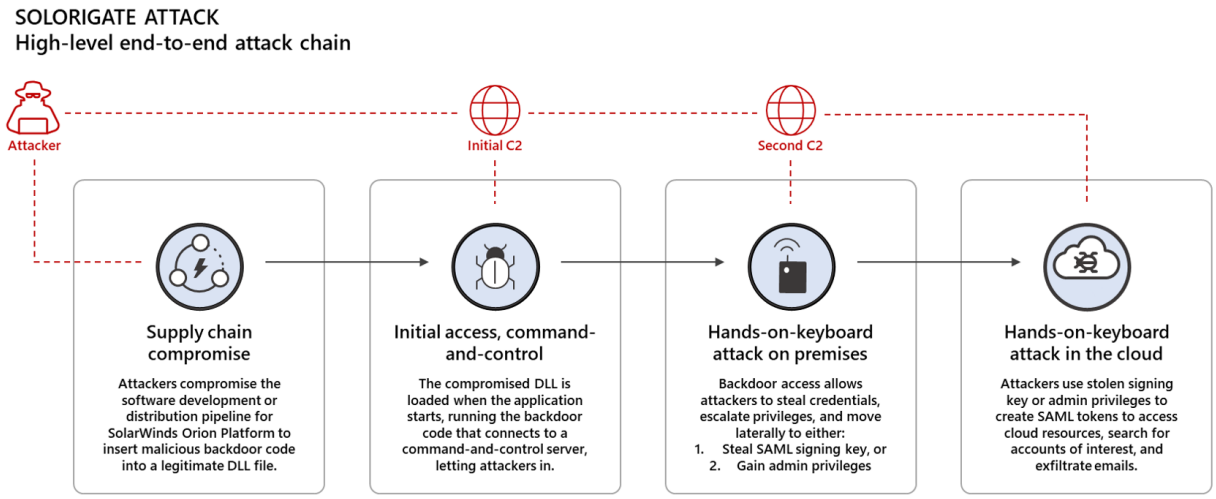
Using Microsoft 365 Defender to protect against Solorigate

microsoft.com/security/blog/2020/12/28/using-microsoft-365-defender-to-coordinate-protection-against-solorigate/

December 28, 2020



> **UPDATE:** Microsoft continues to work with partners and customers to expand our knowledge of the threat actor behind the nation-state cyberattacks that compromised the supply chain of SolarWinds and impacted multiple other organizations. Microsoft previously used 'Solorigate' as the primary designation for the actor, but moving forward, we want to place appropriate focus on the actors behind the sophisticated attacks, rather than one of the examples of malware used by the actors. Microsoft Threat Intelligence Center (MSTIC) has named the actor behind the attack against SolarWinds, the SUNBURST backdoor, TEARDROP malware, and related components as NOBELIUM. As we release new content and analysis, we will use NOBELIUM to refer to the actor and the campaign of attacks.

Microsoft security researchers continue to investigate and respond to the sophisticated cyberattack known as Solorigate (also referred to as Sunburst by FireEye) involving a supply chain compromise and the subsequent compromise of cloud assets. While the related investigations and impact assessments are ongoing, Microsoft is providing visibility into the attack chains and related threat intelligence to the defender community as early as possible so organizations can identify and take action to stop this attack, understand the potential scope of its impact, and begin the recovery process from this active threat. We have established a resource center that is constantly updated as more information becomes available at https://aka.ms/solorigate.

This blog is a comprehensive guide for security operations and incident response teams using Microsoft 365 Defender to identify, investigate, and respond to the Solorigate attack if it's found in your environment. The description of the attack in this blog is based on current analysis and investigations by researchers across Microsoft, our partners, and the intelligence community who are actively collaborating to respond to the attack. This is an active threat that continues to evolve, and the findings included here represent what we know at the time of publishing. We continue to publish and update intelligence, indicators, tactics, techniques, and procedures (TTPs), and related details as we discover them. The report from the Microsoft Security Response Center (MSRC) includes the latest analysis of this threat, known indicators of compromise (IOCs), and initial recommended defenses, and will be updated as new data becomes available.

This blog covers:

Tracking the cross-domain Solorigate attack from endpoint to the cloud

The Solorigate attack is an example of a modern cross-domain compromise. Since these kinds of attacks span multiple domains, having visibility into the entire scope of the attack is key to stopping and preventing its spread.

This attack features a sophisticated technique involving a software supply chain compromise that allowed attackers to introduce malicious code into signed binaries on the SolarWinds Orion Platform, a popular IT management software. The compromised application grants attackers "free" and easy deployment across a wide range of organizations who use and regularly update the application, with little risk of detection because the signed application and binaries are common and are considered trusted. With this initial widespread foothold, the attackers can then pick and choose the specific organizations they want to continue operating within (while others remain an option at any point as long as the backdoor is installed and undetected). Based on our investigations, the next stages of the attack involve on-premises activity with the goal of off-premises access to cloud resources through the following steps:

1. Using the compromised SolarWinds DLL to activate a backdoor that enables attackers to remotely control and operate on a device
2. Using the backdoor access to steal credentials, escalate privileges, and move laterally to gain the ability to create valid SAML tokens using any of two methods:
    1. Stealing the SAML signing certificate (Path 1)
    2. Adding to or modifying existing federation trust (Path 2)
3. Using attacker-created SAML tokens to access cloud resources and perform actions leading to the exfiltration of emails and persistence in the cloud

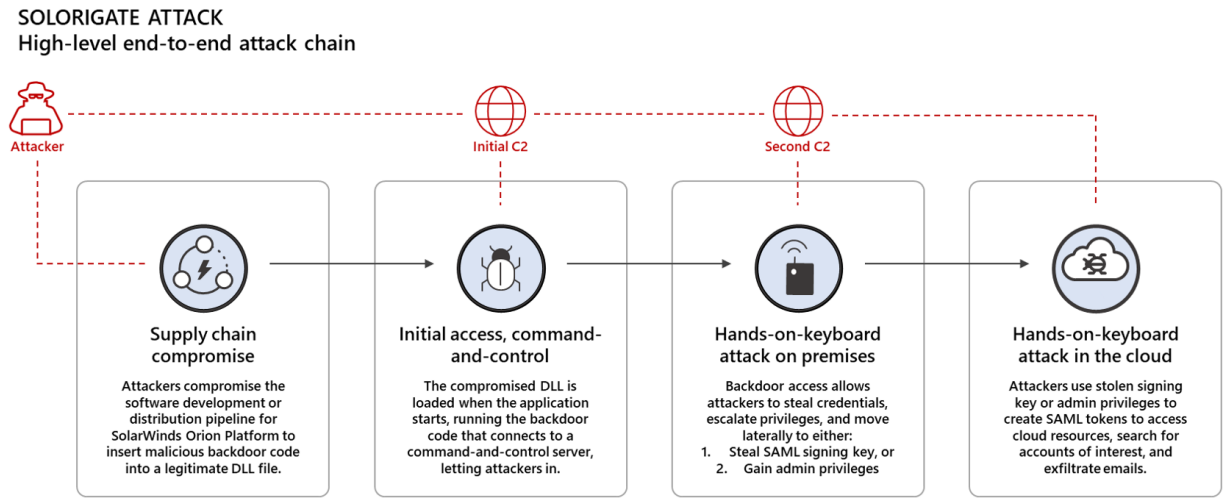


Figure 1. High-level end-to-end Solorigate attack chain

This attack is an advanced and stealthy campaign with the ability to blend in, which could allow attackers to stay under the radar for long periods of time before being detected. The deeply integrated cross-domain security capabilities in Microsoft 365 Defender can empower organizations and their security operations (SOC) teams to uncover this attack, scope out the end-to-end breach from endpoint to the cloud, and take action to block and remediate it. This blog will offer step-by-step guidance to do this by outlining:

- How indicators of attack show up across endpoints, identity, and the cloud
- How Microsoft 365 Defender automatically combines alerts across these different domains into a comprehensive end-to-end story
- How to leverage the powerful toolset available for deep investigation, hunting, and response to enable SOCs to battle the attackers and evict these attackers from both on-premises and cloud environments

Threat analytics: Understanding and responding to active attacks

As soon as this attack was discovered, Microsoft researchers published two threat analytics reports to help organizations determine if they are affected, assess the impact of the attack, and identify actions to contain it.

- Sophisticated actor attacks FireEye provides information about the FireEye breach and compromised red team tools
- Solorigate supply chain attack provides a detailed analysis of the SolarWinds supply chain compromise

The reports are published in Microsoft 365 security center, available to all Microsoft Defender for Endpoint customers and Microsoft 365 Defender early adopters. In addition to detailed descriptions of the attack, TTPs, and indicators of compromise (IoCs), the reports provide real-time data aggregated from signals across Microsoft 365 Defender, indicating the all-up

impact of the threat to the organization, as well as details about relevant incidents and alerts to initiate investigation on. These reports continue to be updated as additional information becomes available.

Given the significance of this threat, we are making similar relevant Microsoft threat intelligence data, including the updated list of IOCs, available to everyone publicly. A comprehensive list of guidance and insights is available at https://aka.ms/solorigate.

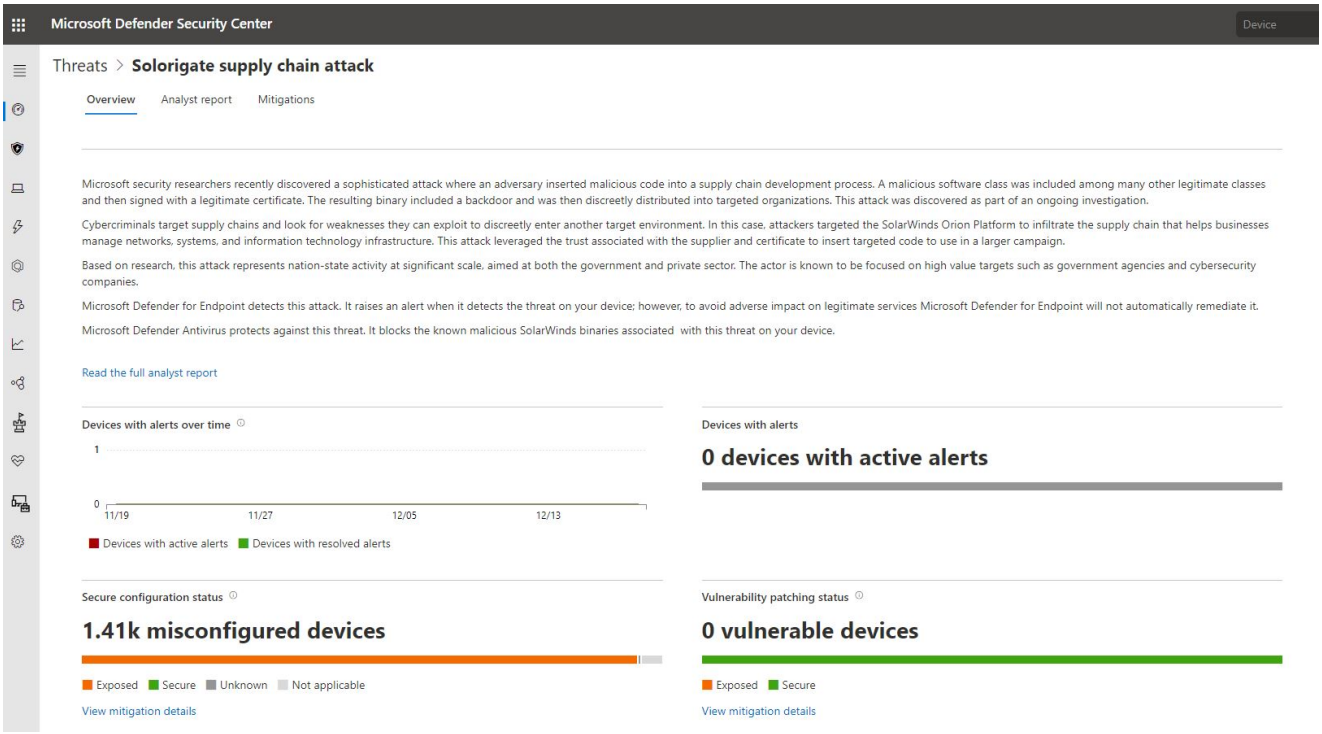


Figure 2. Threat analytics report on Solorigate attack

We recommend Microsoft 365 Defender customers to start their investigations here. After gaining deep understanding of the threat and getting the latest research findings, you can take the following recommended steps:

Find devices with the compromised SolarWinds Orion application

The threat analytics report uses insights from threat and vulnerability management to identify devices that have the compromised SolarWinds Orion Platform binaries or are exposed to the attack due to misconfiguration.

From the *Vulnerability patching status* chart in threat analytics, you can view the mitigation details to see a list of devices with the vulnerability ID TVM-2020-0002, which was added specifically to help with Solorigate investigations:

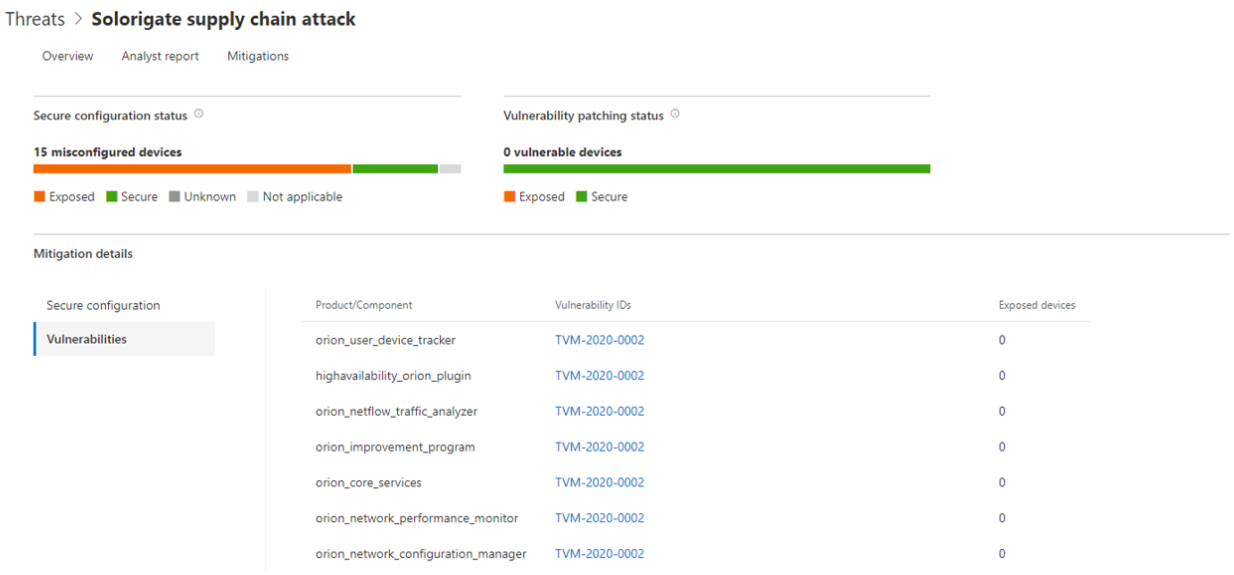


Figure 3. Threat and vulnerability management data shows data on exposed devices

Threat and vulnerability management provides more info about the vulnerability ID TVM-2020-0002, as well as all relevant applications, via the *Software inventory* view. There are also multiple security recommendations to address this specific threat, including instructions to update the software versions installed on exposed devices.

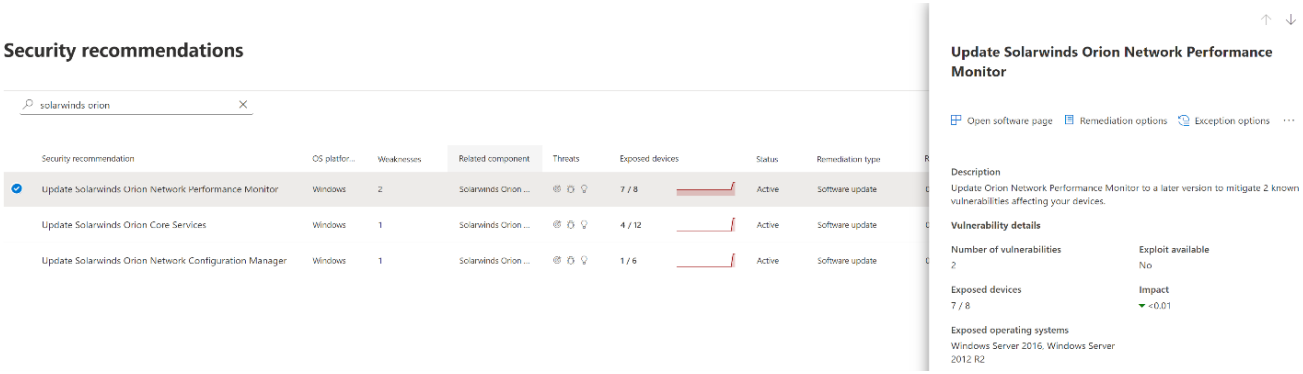


Figure 4. Security recommendations from threat and vulnerability management

Investigate related alerts and incidents

From the threat analytics report, you can quickly locate devices with alerts related to the attack. The *Devices with alerts* chart identifies devices with malicious components or activities known to be directly related to Solorigate. Click through to get the list of alerts and investigate.

Some Solorigate activities may not be directly tied to this specific threat but will trigger alerts due to generally suspicious or malicious behaviors. All alerts in Microsoft 365 Defender provided by different Microsoft 365 products are correlated into incidents. Incidents help you

see the relationship between detected activities, better understand the end-to-end picture of the attack, and investigate, contain, and remediate the threat in a consolidated manner.

Review incidents in the Incidents queue and look for those with alerts relevant to this attacker's TTPs, as described in the threat analytics report (also listed at the end of this blog).

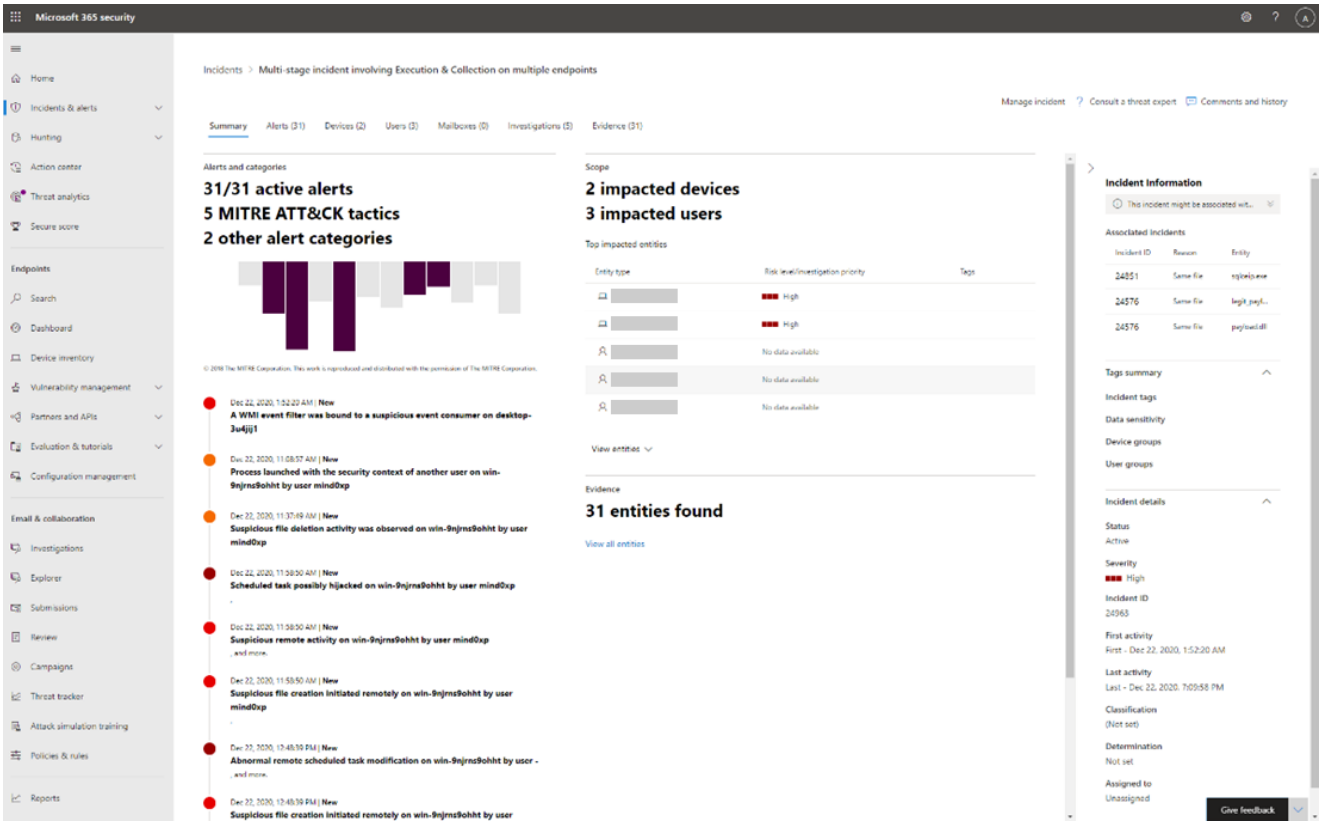


Figure 5. Consolidated Incident view for Solorigate

Some alerts are specially tagged with Microsoft Threat Experts to indicate malicious activities that Microsoft researchers found in customer environments during hunting. As part of the Microsoft Threat Experts service, researchers investigated this attack as it unfolded, hunting for associated attacker behaviors, and sent targeted attack notifications. If you see an alert tagged with Microsoft Threat Experts, we strongly recommend that you give it immediate attention.

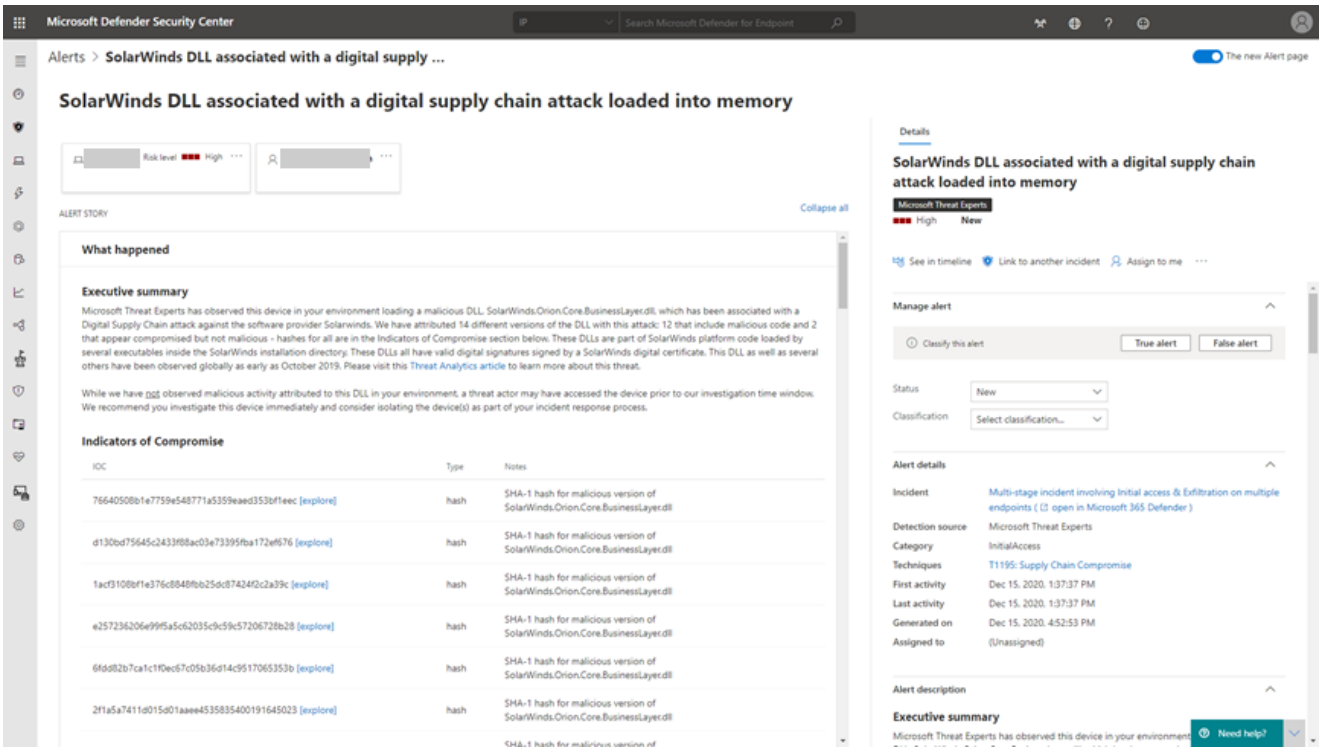


Figure 6. Microsoft Threat Experts targeted attack notification

Additionally, Microsoft Threat Experts customers with Experts on demand subscriptions can reach out directly to our on-demand hunters for additional help in understanding the Solorigate threat and the scope of its impact in their environments.

Hunt for related attacker activity

The threat analytics report also provides advanced hunting queries that can help analysts locate additional related or similar activities across endpoint, identity, and cloud. Advanced hunting uses a rich set of data sources, but in response to Solorigate, Microsoft has enabled streaming of Azure Active Directory (Azure AD) audit logs into advanced hunting, available for all customers in public preview. These logs provide traceability for all changes done by various features within Azure AD. Examples of audit logs include changes made to any resources within Azure AD, such as adding or removing users, apps, groups, roles, and policies. Customers who do not have Microsoft Defender for Endpoint or are not early adopters for Microsoft 365 Defender can see our recommended advanced hunting queries.

Currently, this data is available to customers who have Microsoft Cloud App Security with the Office365 connector. Our intent is to expand availability to more Microsoft 365 Defender customers. The new log data is available in the CloudAppEvents table:

```
CloudAppEvents
| where Application == “Office 365”
```

The log data contains activity logs useful for investigating and finding Azure AD-related activities. This data further enriches the CloudAppEvents table, which also has Exchange Online and Microsoft Teams activities.

As part of making this new data available, we also published a handful of relevant advanced hunting queries, identified by the suffix *[Solorigate]*, to the GitHub repo.

Here's an example query that helps you see when credentials are added to an Azure AD application after 'Admin Consent' permissions were granted:

```
CloudAppEvents
| where Application == "Office 365"
| where ActionType == "Consent to application."
| where RawEventData.ModifiedProperties[0].Name == "ConsentContext.IsAdminConsent"
and RawEventData.ModifiedProperties[0].NewValue == "True"
| extend spnID = tostring(RawEventData.Target[3].ID)
| parse RawEventData.ModifiedProperties[4].NewValue with * "=> [[" dummpy "Scope: " After
"]]" *
| extend PermissionsGranted = split(After, "]",0)
| project ConsentTime = Timestamp , AccountDisplayName , spnID , PermissionsGranted
| join (
CloudAppEvents
| where Application == "Office 365"
| where ActionType == "Add service principal credentials." or ActionType == "Update
application – Certificates and secrets management "
| extend spnID = tostring(RawEventData.Target[3].ID)
| project AddSecretTime = Timestamp, AccountDisplayName , spnID
) on spnID
| where ConsentTime < AddSecretTime and AccountDisplayName <> AccountDisplayName1
```

Microsoft 356 Defender advanced hunting can also assist in many of the recommended incident investigation tasks outlined in the blog, Advice for incident responders on recovery from systemic identity compromises.

In the remaining sections, we will discuss select examples of alerts raised by Microsoft 365 solutions that monitor and detect Solorigate activities across the attack chain on endpoint, identity, and the cloud. These are alerts you may encounter when investigating incidents in Microsoft 365 security center if your organization is affected by this threat. We will also indicate activities which are now blocked by Microsoft 365 Defender. Lastly, each section contains examples of hunting queries you will find useful for hunting for various attacker activities in your environment.

Detecting and blocking malware and malicious behavior on endpoints

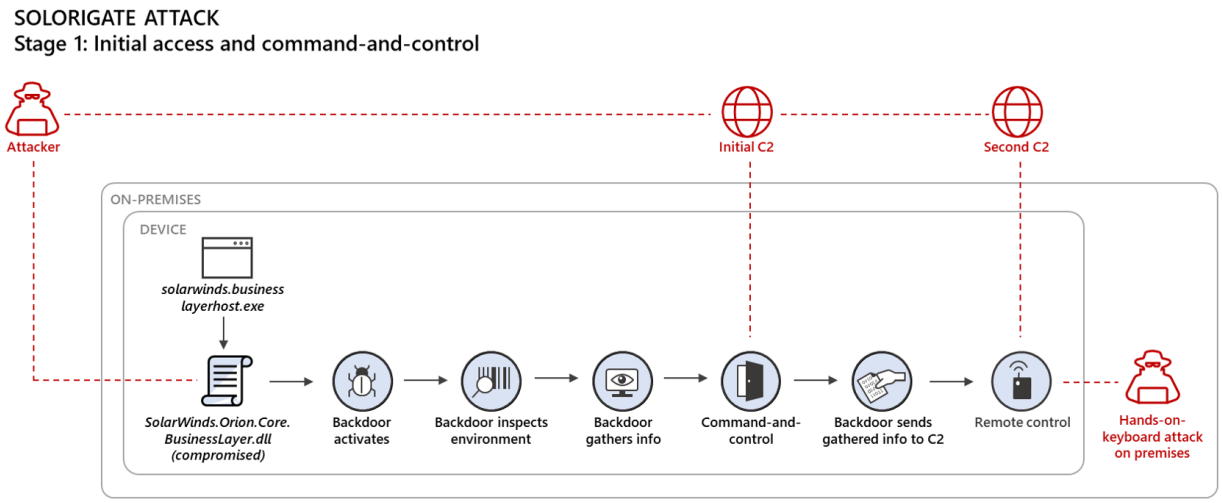


Figure 7. Solorigate attack chain: Initial access and command-and-control

Discovering and blocking backdoor activity

When the compromised SolarWinds binary *SolarWinds.Orion.Core.BusinessLayer.dll* gets loaded on a device through normal update channels, the backdoor goes through an extensive list of checks to ensure it's running in an actual enterprise network and not on an analyst's machine. It then contacts a command-and-control (C2) server using a subdomain that is generated partly with information gathered from the affected device, which means a unique subdomain is generated for each affected domain. The backdoor allows the attackers to remotely run commands on the device and move to the next stages of the attack. For more information, read our in-depth analysis of the Solorigate malware.

Microsoft Defender for Endpoint delivers comprehensive protection against this threat (see full list of detection and protection alerts at the end of this blog). Microsoft Defender Antivirus, the default antimalware solution on Windows 10, detects and blocks the malicious DLL and its behaviors. It quarantines the malware, even if the process is running.

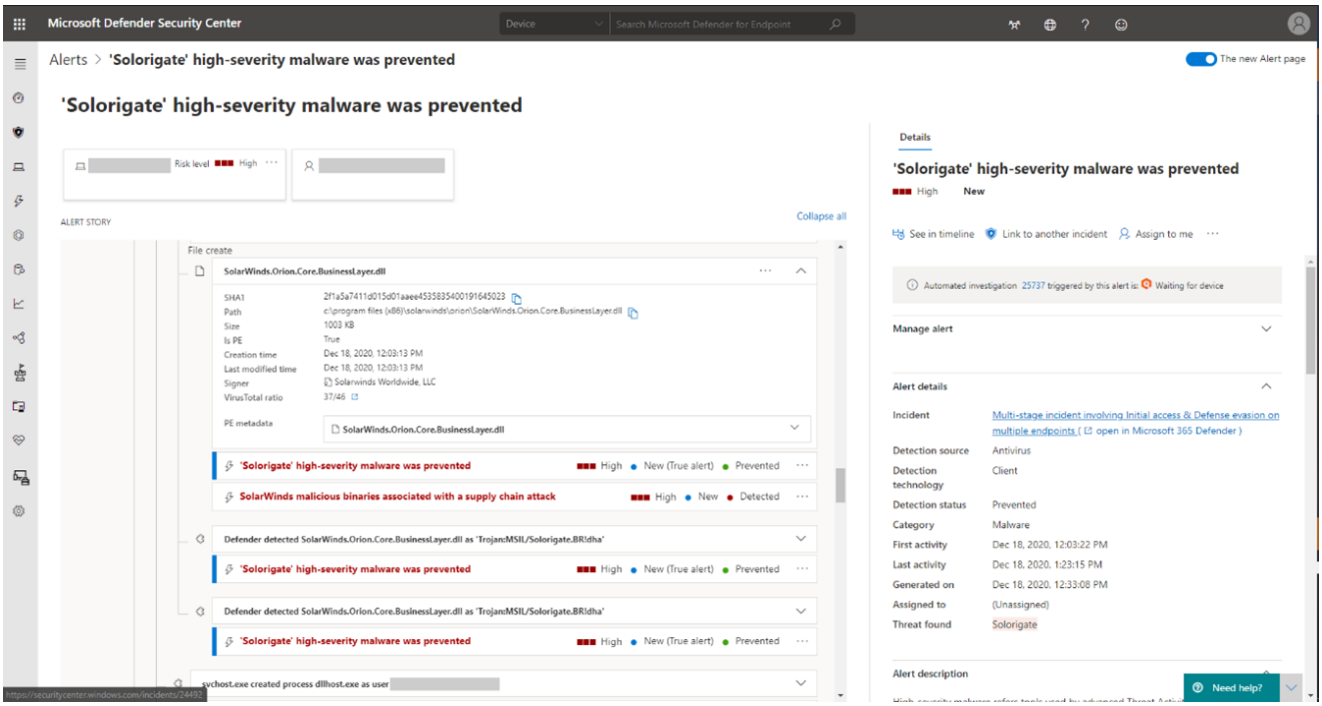


Figure 8. Microsoft Defender for Endpoint blocks malicious binaries

If the malicious code is successfully deployed, the backdoor lies dormant for up to two weeks. It then attempts to contact numerous C2 domains, with the primary domain being **.avsvmcloud[.]com*. The backdoor uses a domain generation algorithm to evade detection. Microsoft 365 Defender detects and blocks this behavior.

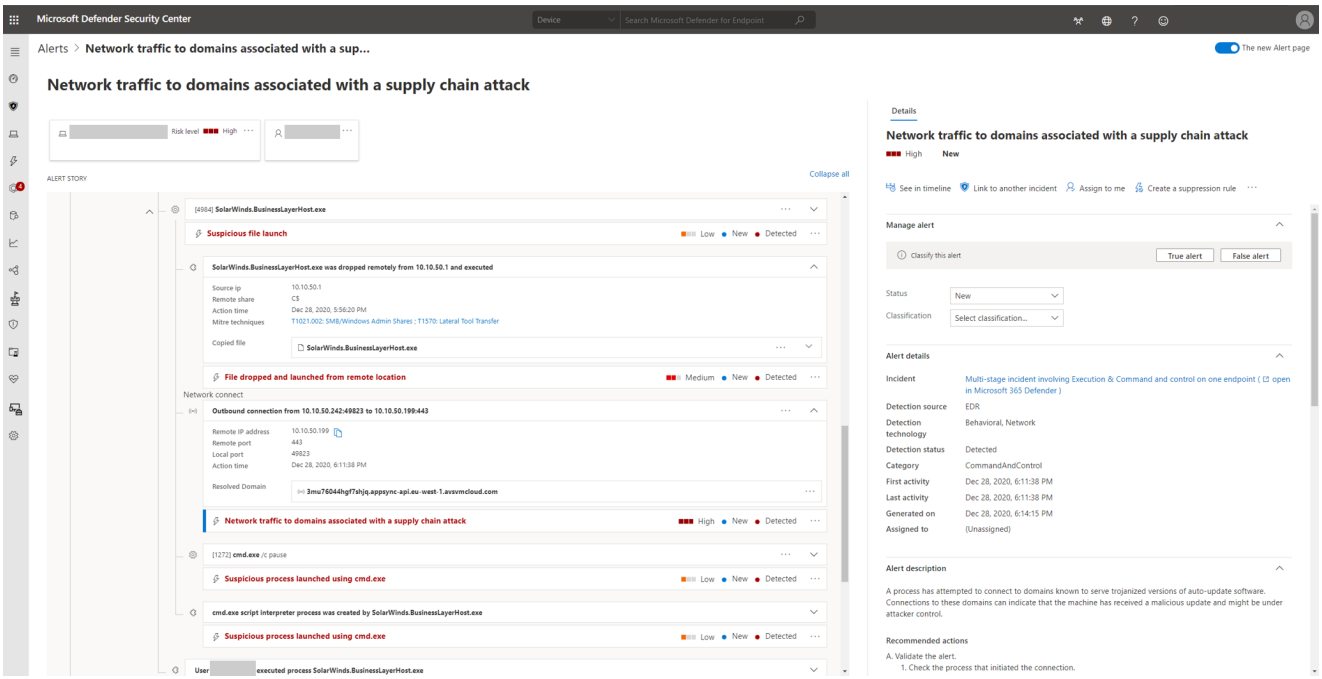


Figure 9. Microsoft Defender for Endpoint prevented malicious C2 callback

Discovering potentially tampered devices

To evade security software and analyst tools, the Solorigate malware enumerates the target system looking for certain running processes, loaded drivers, and registry keys, with the goal of disabling them.

The Microsoft Defender for Endpoint sensor is one of the processes the malware attempts to disable. Microsoft Defender for Endpoint has built-in protections against many techniques attackers use to disable endpoint sensors ranging from hardened OS protection, anti-tampering policies, and detections for a variety of tampering attempts, including “Attempt to stop Microsoft Defender for Endpoint sensor”, “Tampering with Microsoft Defender for Endpoint sensor settings”, or “Possible sensor tampering in memory”.

Successfully disabling Microsoft Defender for Endpoint can prevent the system from reporting observed activities. However, the multitude of signals reported into Microsoft 365 Defender provides a unique opportunity to hunt for systems where the tampering technique used might have been successful. The following advanced hunting query can be used to locate devices that should be reporting but aren’t:

```
// Times to be modified as appropriate
let timeAgo=1d;
let silenceTime=8h;
// Get all silent devices and IPs from network events
let allNetwork=materialize(DeviceNetworkEvents
| where Timestamp > ago(timeAgo)
and isnotempty(LocalIP)
and isnotempty(RemoteIP)
and ActionType in (“ConnectionSuccess”, “InboundConnectionAccepted”)
and LocalIP !in (“127.0.0.1”, “::1”)
| project DeviceId, Timestamp, LocalIP, RemoteIP, ReportId);
let nonSilentDevices=allNetwork
| where Timestamp > ago(silenceTime)
| union (DeviceProcessEvents | where Timestamp > ago(silenceTime))
| summarize by DeviceId;
let nonSilentIPs=allNetwork
| where Timestamp > ago(silenceTime)
| summarize by LocalIP;
let silentDevices=allNetwork
| where DeviceId !in (nonSilentDevices)
and LocalIP !in (nonSilentIPs)
| project DeviceId, LocalIP, Timestamp, ReportId;
// Get all remote IPs that were recently active
let addressesDuringSilence=allNetwork
| where Timestamp > ago(silenceTime)
| summarize by RemoteIP;
```

```
// Potentially disconnected devices were connected but are silent
silentDevices
| where LocalIP in (addressesDuringSilence)
| summarize ReportId=arg_max(Timestamp, ReportId), Timestamp=max(Timestamp),
LocalIP=arg_max(Timestamp, LocalIP) by DeviceId
| project DeviceId, ReportId=ReportId1, Timestamp, LocalIP=LocalIP1
```

Microsoft is continuously developing additional measures to both block and alert on these types of tampering activities.

Detecting hands-on-keyboard activity within an on-premises environment

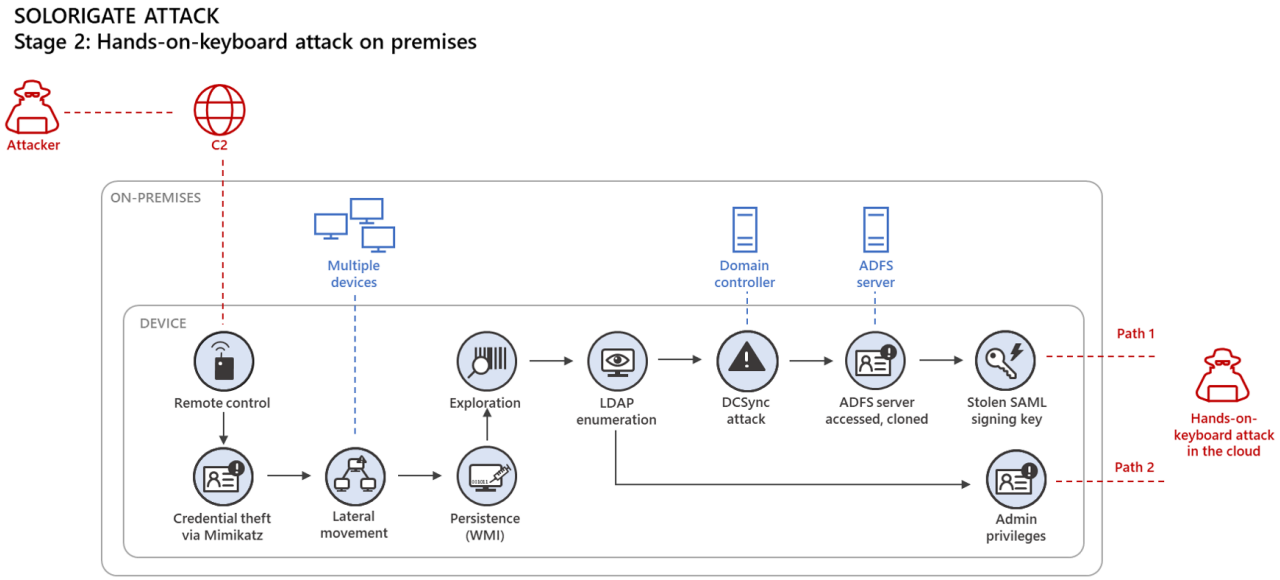


Figure 10. Solorigate attack chain: Hands-on-keyboard attack on premises

After establishing a backdoor connection on an affected device, the attacker's next goal is to achieve off-premises access to the organization's cloud services. To do this, they must find a way to gain permissions to those services. One technique we have seen the attackers use is to go after the organization's Active Directory Federation Services (AD FS) server to obtain the proverbial "keys" to the identity kingdom. AD FS enables federated identity and access management by securely sharing digital identity and entitlement rights across security and enterprise boundaries; effectively, it is the "LSASS for the cloud." Among other things, AD FS stores the Security Assertion Markup Language (SAML) token signing certificate, which is used to create authorization tokens for users or services in the organization so they can access cloud applications and resources after authentication.

To attack the AD FS infrastructure, the attackers must first obtain appropriate domain permissions through on-premises intelligence gathering, lateral movement, and credential theft. Building from the backdoor described above, the attackers leverage fileless techniques for privilege escalation, persistence, and lateral movement, including evading analysis by using system binaries and exploration tools that masquerade as other benign binaries. The attackers also carefully chose organization-specific command-and-control (C2) domains and use custom organization-specific tool naming and locations.

Microsoft Defender for Endpoint detects a wide array of these attack techniques, allowing SOC teams to track the attacker's actions in the environment and take actions to contain the attack. The following section covers detections for the techniques used by the attackers to compromise the AD FS infrastructure.

Identifying attacker reconnaissance

Attackers collect data from Active Directory using a renamed version of the utility ADFind, running queries against Domain Controllers as part of the reconnaissance stage of the attack. Microsoft Defender for Endpoint detects this behavior and allows the SOC analyst to track compromised devices at this stage to gain visibility into the information the attacker is looking for.

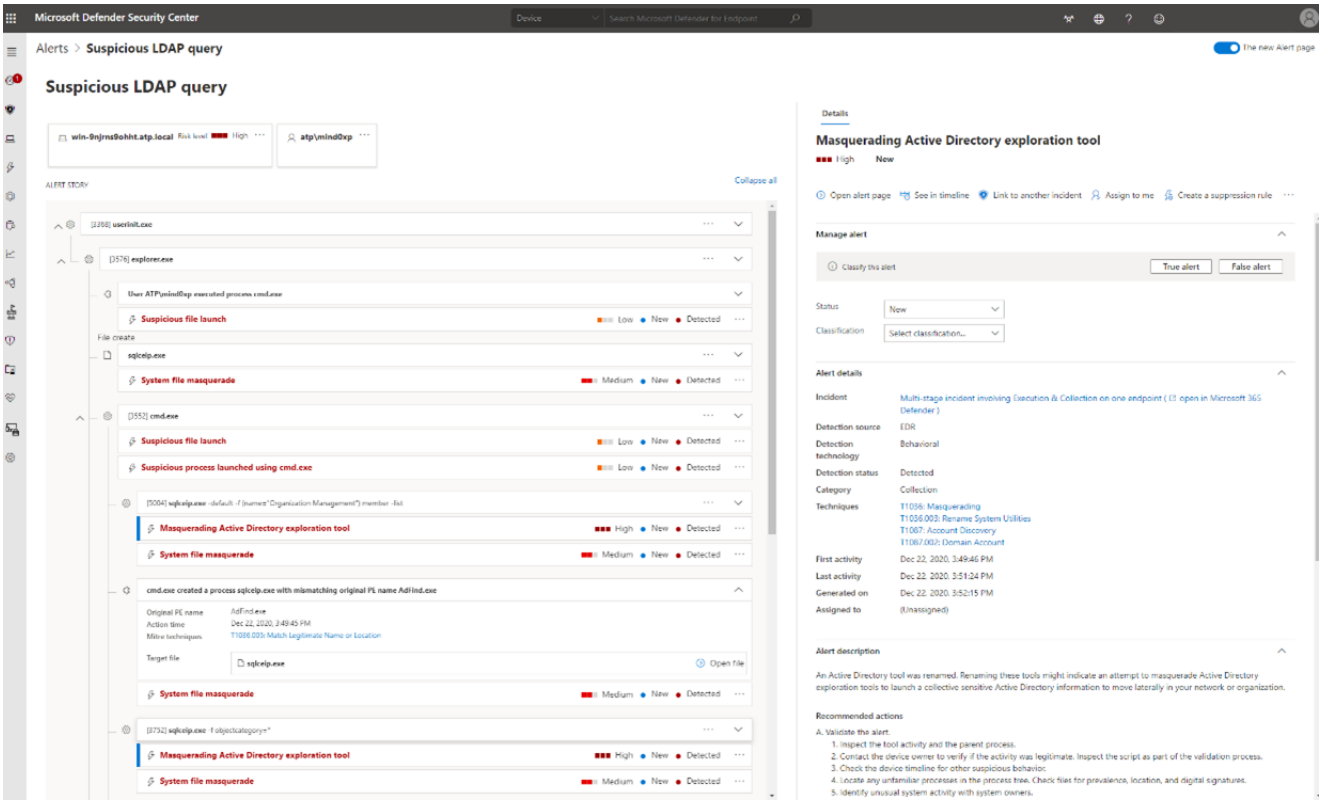


Figure 11. Microsoft Defender for Endpoint detects usage of masquerading exploration tools

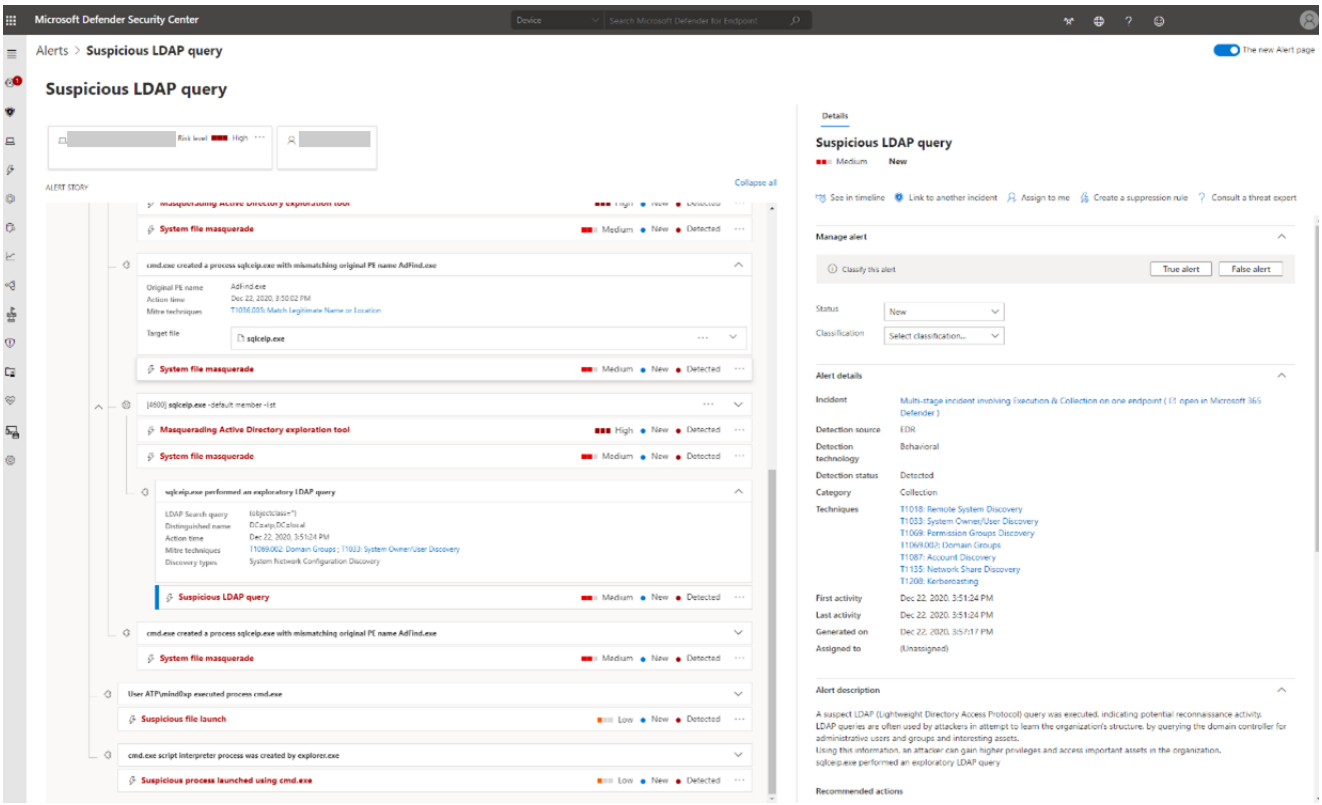


Figure 12. Microsoft Defender for Endpoint detects usage LDAP query for reconnaissance.

Stopping lateral movement and credential theft

To gain access to a highly privileged account needed for later steps in the kill chain, the attackers move laterally between devices and dump credentials until an account with the needed privileges is compromised, all while remaining as stealthy as possible.

A variety of credential theft methods, such as dumping LSASS memory, are detected and blocked by Microsoft Defender for Endpoint. The example below shows the detection of lateral movement using Windows Management Instrumentation (WMI) to run the attacker's payload using the *Rundll32.exe* process.

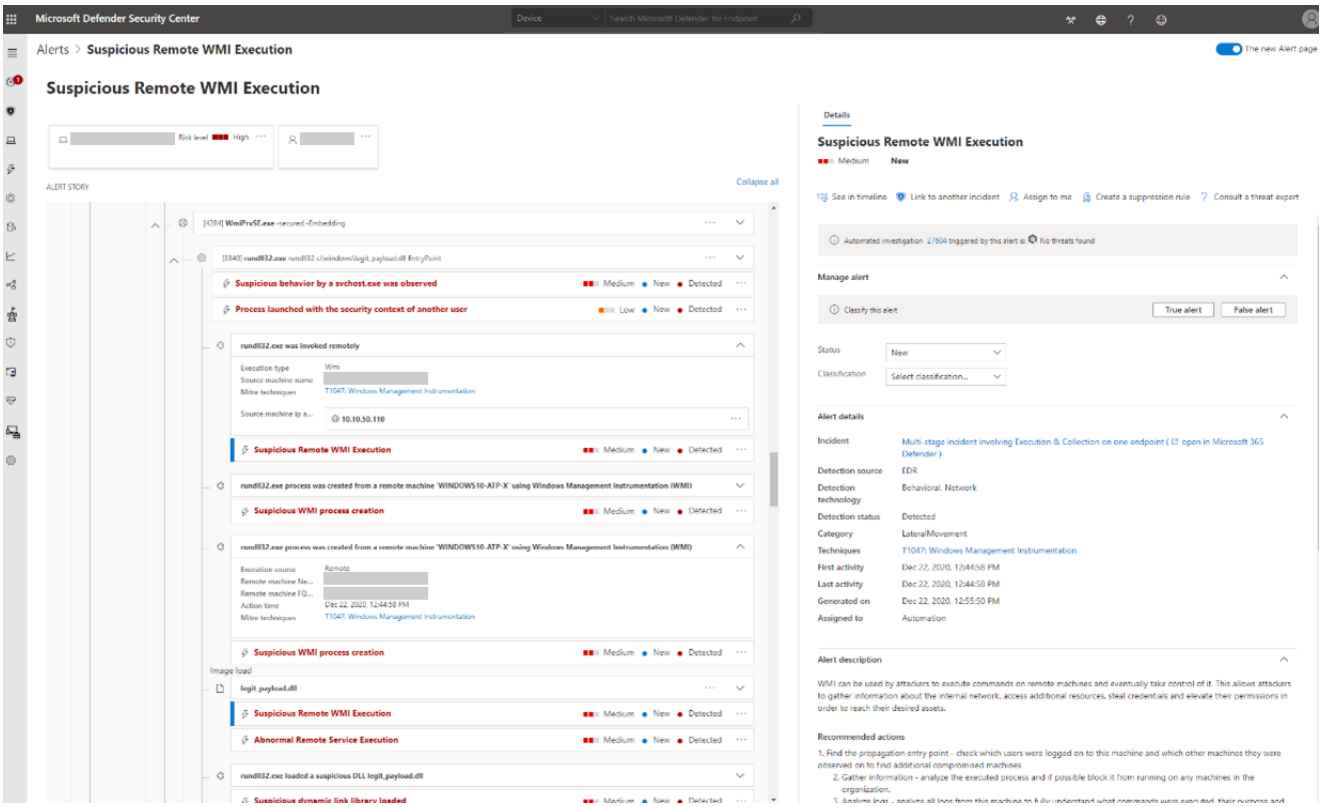


Figure 13. Microsoft Defender for Endpoint alert for suspicious remote WMI execution highlighting the attacker's device and payload

Microsoft Defender for Identity also detects and raises alerts on a variety of credential theft techniques. In addition to watching for alerts, security analysts can hunt across identity data in Microsoft 365 Defender for signs of identity compromise. Here are a couple of example Microsoft Defender for Identity queries looking for such patterns:

Enumeration of high-value DC assets followed by logon attempts to validate stolen credentials in time proximity

```
let MaxTime = 1d;
let MinNumberLogon = 5;
//devices attempting enumeration of high-value DC
IdentityQueryEvents
| where Timestamp > ago(30d)
| where Application == "Active Directory"
| where QueryTarget in ("Read-only Domain Controllers")
//high-value RODC assets
| project Timestamp, Protocol, Query, DeviceName, AccountUpn
| join kind = innerunique (
//devices trying to logon {MaxTime} after enumeration
IdentityLogonEvents
| where Timestamp > ago(30d)
```

| where ActionType == “LogonSuccess”
| project LogonTime = Timestamp, DeviceName, DestinationDeviceName) on DeviceName
| where LogonTime between (Timestamp .. (Timestamp + MaxTime))
| summarize n=dcount(DestinationDeviceName), TargetedDC =
makeset(DestinationDeviceName) by Timestamp, Protocol, DeviceName
| where n >= MinNumberLogon

High-volume of LDAP queries in short time filtering for non-DC devices

let Threshold = 12;
let BinTime = 1m;
//approximate list of DC
let listDC=IdentityDirectoryEvents
| where Application == “Active Directory”
| where ActionType == “Directory Services replication”
| summarize by DestinationDeviceName;
IdentityQueryEvents
| where Timestamp > ago(30d)
//filter out LDAP traffic across DC
| where DeviceName !in (listDC)
| where ActionType == “LDAP query”
*| parse Query with * “Search Scope: ” SearchScope “, Base Object:” BaseObject “, Search Filter: ” SearchFilter*
| summarize NumberOfDistinctLdapQueries = dcount(SearchFilter) by DeviceName, bin(Timestamp, BinTime)
| where NumberOfDistinctLdapQueries > Threshold

At this point, SOC teams can take containment measures within the Microsoft 365 security center, for example, using indicators to isolate the devices involved and block the remotely executed payload across the environment, as well as mark suspect users as compromised.

Detecting and remediating persistence

Microsoft Defender for Endpoint also detects the advanced defense evasion and masquerading techniques used by the attackers to make their actions as close to normal as possible, such as binding a WMI event filter with a logical consumer to remain persistent. Follow the recommended actions in the alert to remove persistence and prevent the attacker’s payload from loading after reboot.

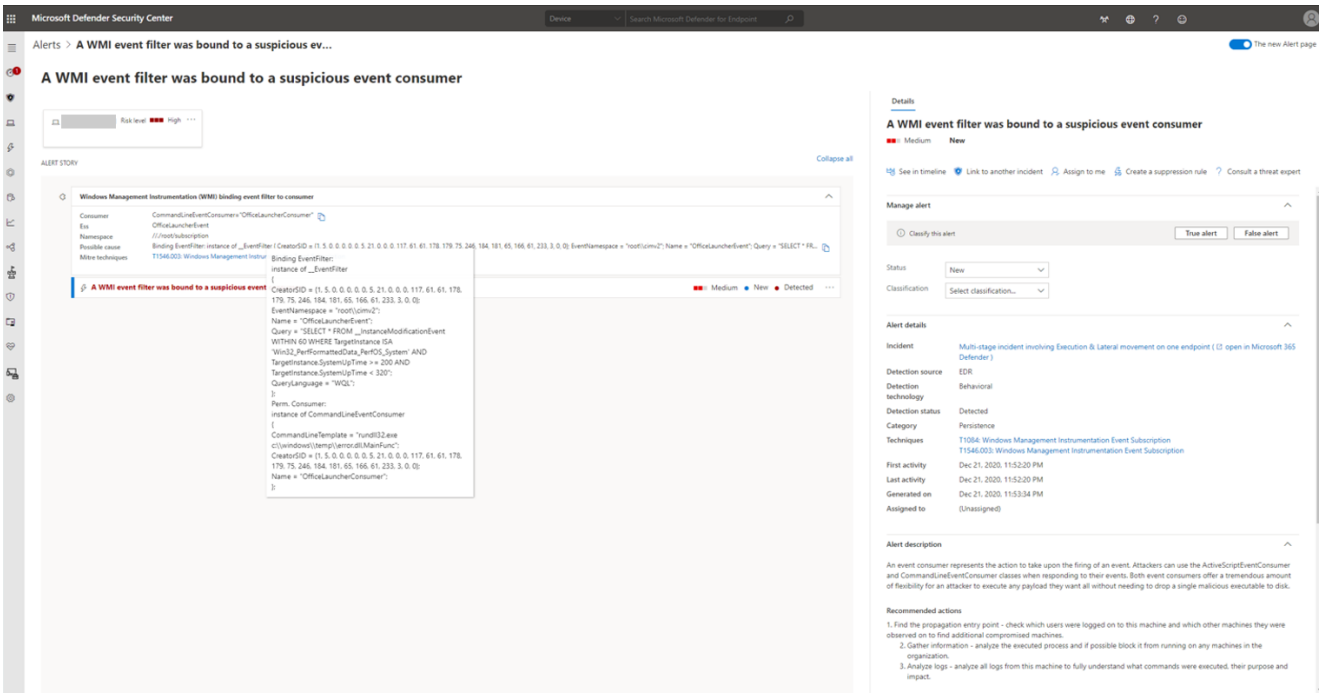


Figure 14. Microsoft Defender for Endpoint alert for WMI event filter bound to a suspicious consumer showing the persistence and the scheduled command line

Catching AD FS compromise and the attacker's ability to impersonate users in the cloud

The next step in the attack focuses on the AD FS infrastructure and can unfold in two separate paths that lead to the same outcome—the ability to create valid SAML tokens allowing impersonation of users in the cloud:

- **Path 1 – Stealing the SAML signing certificate**: After gaining administrative privileges in the organization's on-premises network, and with access to the AD FS server itself, the attackers access and extract the SAML signing certificate. With this signing certificate, the attackers create valid SAML tokens to access various desired cloud resources as the identity of their choosing.
- **Path 2 – Adding to or modifying existing federation trust**: After gaining administrative Azure Active Directory (Azure AD) privileges using compromised credentials, the attackers add their own certificate as a trusted entity in the domain either by adding a new federation trust to an existing tenant or modifying the properties of an existing federation trust. As a result, any SAML token they create and sign will be valid for the identity of their choosing.

In the first path, obtaining the SAML signing certificate normally entails first querying the private encryption key that resides on the AD FS container and then using that key to decrypt the signing certificate. The certificate can then be used to create illicit but valid SAML tokens that allow the actor to impersonate users, enabling them to access enterprise cloud applications and services.

Microsoft Defender for Endpoint and Microsoft Defender for Identity detect the actions that attackers take to steal the encryption key needed to decrypt the SAML signing certificate. Both solutions leverage unique LDAP telemetry to raise high-severity alerts highlighting the attacker's progress towards creating illicit SAML tokens.

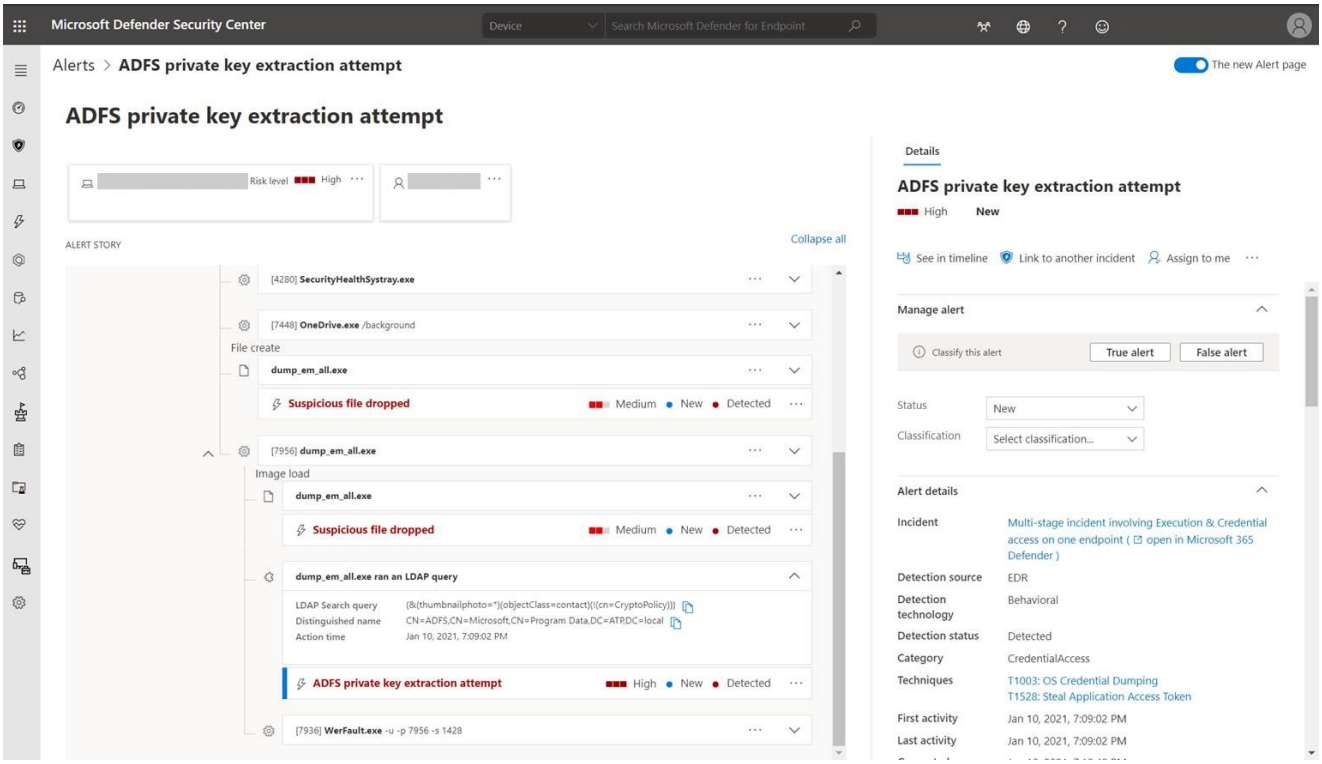


Figure 15. Microsoft Defender for Endpoint detects a suspicious LDAP query being launched and an attempted AD FS private key extraction

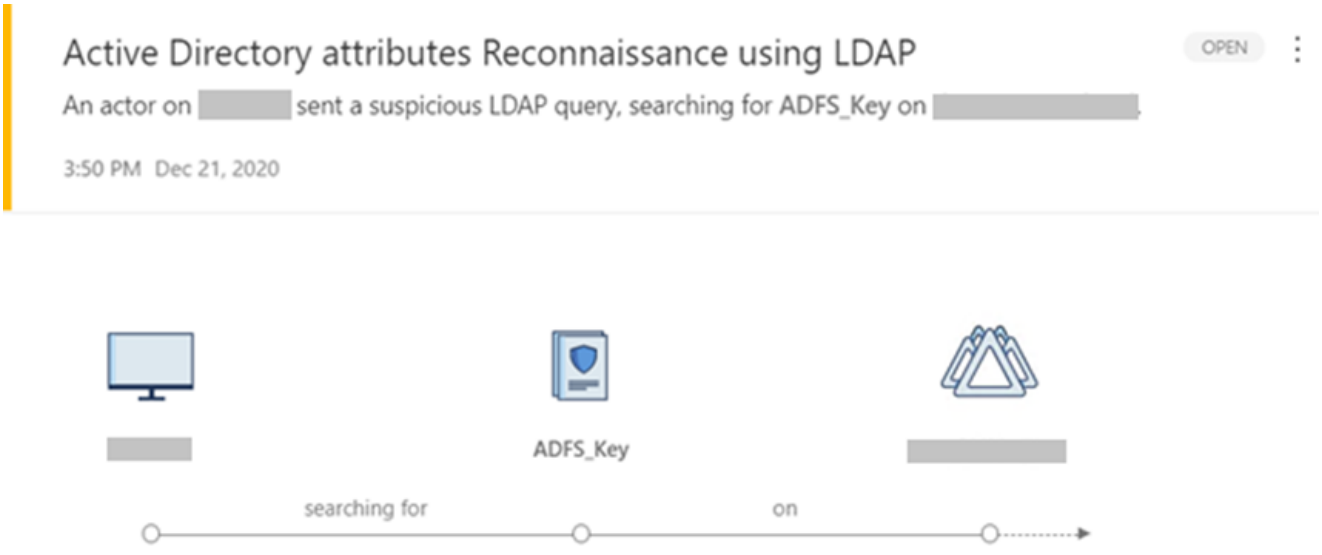


Figure 16. Microsoft Defender for Identity detects private key extraction via malicious LDAP requests

For the second path, the attackers create their own SAML signing certificate outside of the organization's environment. With Azure AD administrative permissions, they then add the new certificate as a trusted object. The following advanced hunting query over Azure AD audit logs shows when domain federation settings are changed, helping to discover where the attackers configured the domain to accept authorization tokens signed by their own signing certificate. As these are rare actions, we advise verifying that any instances identified are the result of legitimate administrative activity.

ADFSDomainTrustMods

```
let auditLookback = 1d; CloudAppEvents
| where Timestamp > ago(auditLookback)
| where ActionType =~ "Set federation settings on domain."
| extend targetDetails = parse_json(ActivityObjects[1])
| extend targetDisplayName = targetDetails.Name
| extend resultStatus = extractjson("$.ResultStatus", tostring(RawEventData), typeof(string))
| project Timestamp, ActionType, InitiatingUserOrApp=AccountDisplayName,
targetDisplayName, resultStatus, InitiatingIPAddress=IPAddress, UserAgent
```

If the SAML signing certificate is confirmed to be compromised or the attacker has added a new one, follow the best practices for invalidating through certificate rotation to prevent further use and creation of SAML tokens by the attacker. Additionally, affected AD FS servers may need to be isolated and remediated to ensure no remaining attacker control or persistence.

If the attackers accomplish either path, they gain the ability to create illicit SAML tokens for the identities of their choosing and bypass multifactor authentication (MFA), since the service or application accepting the token assumes MFA is a necessary previous step in creating a properly signed token. To prevent attackers from progressing to the next stage, which is to access cloud resources, the attack should be discovered and remediated at this stage.

Detecting the hands-on-keyboard activity in the cloud environment

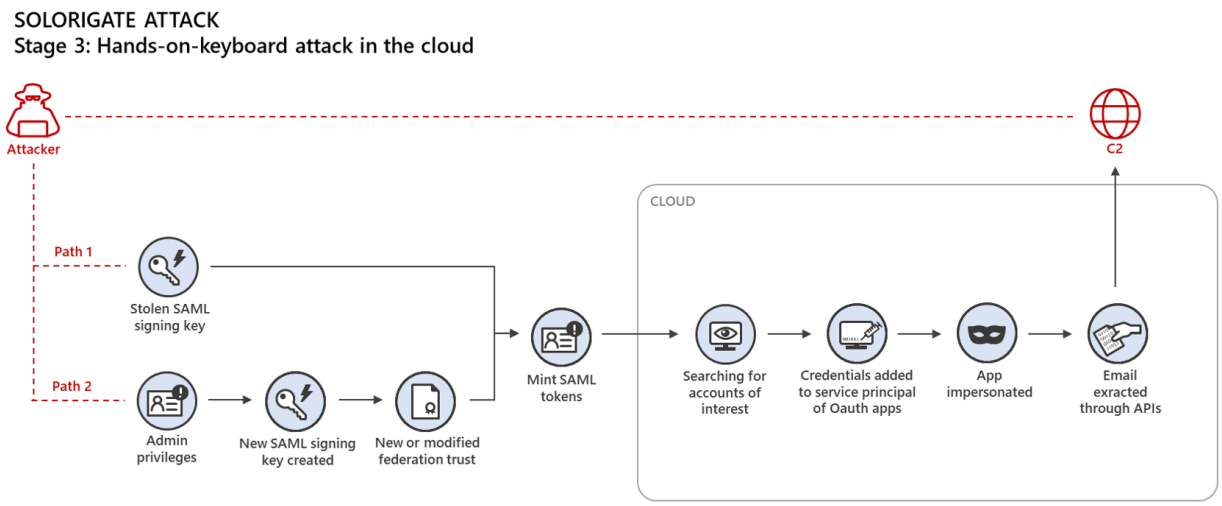


Figure 17. Solorigate attack chain: Hands-on-keyboard attack in the cloud

With the ability to create illicit SAML tokens, the attackers can access sensitive data without having to originate from a compromised device or be confined to on-premises persistence. By abusing API access via existing OAuth applications or service principals, they can attempt to blend into the normal pattern of activity, most notably apps or service principals with existing *Mail.Read* or *Mail.ReadWrite* permissions to read email content via Microsoft Graph from Exchange Online. If the application does not already have read permissions for emails, then the app may be modified to grant those permissions.

Identifying unusual addition of credentials to an OAuth app

Microsoft Cloud App Security (MCAS) has added new automatic detection of unusual credential additions to an OAuth application to alert SOCs about apps that have been compromised to extract data from the organization. This detection logic is built on an anomaly detection engine that learns from each user in the environment, filtering out normal usage patterns to ensure alerts highlight real attacks and not false positives. If you see this alert in your environment and confirm malicious activity, you should take immediate action to suspend the user, mark the user as compromised, reset the user's password, and remove the credential additions. You may consider disabling the application during investigation and remediation.

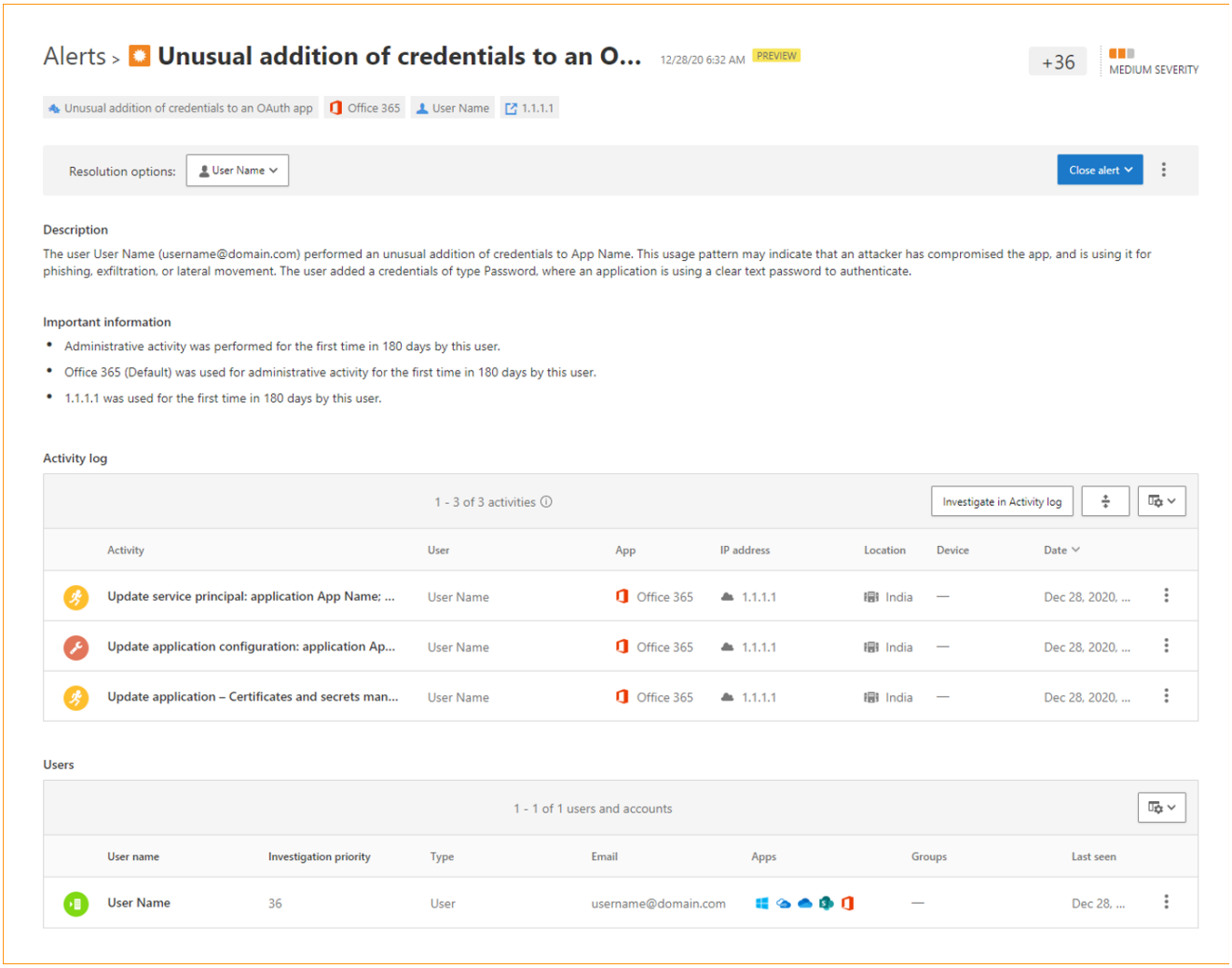


Figure 18. Microsoft Defender Cloud App Security alert for unusual addition of credentials to an OAuth app

SOCs can use the following Microsoft 365 Defender advanced hunting query over Azure AD audit logs to examine when new credentials have been added to a service principle or application. In general, credential changes may be rare depending on the type and use of the service principal or application. SOCs should verify unusual changes with their respective owners to ensure they are the result of legitimate administrative actions.

NewAppOrServicePrincipalCredential

```
let auditLookback = 1d; CloudAppEvents
| where Timestamp > ago(auditLookback)
| where ActionType in (“Add service principal.”, “Add service principal credentials.”, “Update application – Certificates and secrets management “)
| extend RawEventData = parse_json(RawEventData)
| where RawEventData.ResultStatus =~ “success”
| where AccountDisplayName has “@”
| extend targetDetails = parse_json(ActivityObjects[1])
```

```
| extend targetId = targetDetails.Id
| extend targetType = targetDetails.Type
| extend targetDisplayName = targetDetails.Name
| extend keyEvents = RawEventData.ModifiedProperties
| where keyEvents has “KeyIdentifier=” and keyEvents has “KeyUsage=Verify”
| mvexpand keyEvents
| where keyEvents.Name =~ “KeyDescription”
| parse keyEvents.NewValue with * “KeyIdentifier=” keyIdentifier:string “,KeyType=”
keyType:string “,KeyUsage=” keyUsage:string “,DisplayName=” keyDisplayName:string “]” *
| parse keyEvents.OldValue with * “KeyIdentifier=” keyIdentifierOld:string “,KeyType” *
| where keyEvents.OldValue == “[]” or keyIdentifier != keyIdentifierOld
| where keyUsage == “Verify”
| project-away keyEvents
| project Timestamp, ActionType, InitiatingUserOrApp=AccountDisplayName,
InitiatingIPAddress=IPAddress, UserAgent, targetDisplayName, targetId, targetType,
keyDisplayName, keyType, keyUsage, keyIdentifier
```

Discovering malicious access to mail items

OAuth applications or service principals with *Mail.Read* or *Mail.ReadWrite* permissions can read email content from Exchange Online via the Microsoft Graph. To help increase visibility on these behaviors, the *MailItemsAccessed* action is now available via the new Exchange mailbox advanced audit functionality. See if this feature is enabled by default for you. Important note for customers: If you have customized the list of audit events you are collecting, you may need to manually enable this telemetry.

If more than 1,000 *MailItemsAccessed* audit records are generated in less than 24 hours, Exchange Online stops generating auditing records for *MailItemsAccessed* activity for 24 hours and then resumes logging after this period. This throttling behavior is a good starting point for SOCs to discover potentially compromised mailboxes.

MailItemsAccessedThrottling

```
let starttime = 2d;
let endtime = 1d;
CloudAppEvents
| where Timestamp between (startofday(ago(starttime))..startofday(ago(endtime)))
| where ActionType == “MailItemsAccessed”
| where isnotempty(RawEventData[‘ClientAppId’]) and RawEventData[‘OperationProperties’]
[1] has “True”
| project Timestamp, RawEventData[‘OrganizationId’],AccountObjectId,UserAgent
```

In addition to looking for throttled telemetry, you can also hunt for OAuth applications reading mail via the Microsoft Graph API whose behavior has changed prior to a baseline period.

OAuthGraphAPIAnomalies

```
//Look for OAuth App reading mail via GraphAPI — that did not read mail via graph API in prior week
let appMailReadActivity = (timeframeStart:datetime, timeframeEnd:datetime) {
CloudAppEvents
| where Timestamp between (timeframeStart .. timeframeEnd)
| where ActionType == “MailItemsAccessed”
| where RawEventData has “00000003-0000-0000-c000-000000000000” // performance check
| extend rawData = parse_json(RawEventData)
| extend AppId = tostring(parse_json(rawData.AppId))
| extend OAuthAppId = tostring(parse_json(rawData.ClientAppId)) // extract OAuthAppId
| summarize by OAuthAppId
};
appMailReadActivity(ago(1d),now()) // detection period
| join kind = leftanti appMailReadActivity(ago(7d),ago(2d)) // baseline period
on OAuthAppId
```

Microsoft 365 Defender’s cross-domain XDR correlation enables stronger response to critical security incidents

Like the rest of the security industry, Microsoft continues to track the Solorigate attack, an active threat that continues to unfold as well as evolve. As part of empowering our customers and the larger security community to respond to this attack through sharing intelligence and providing advice, this blog serves to guide Microsoft 365 customers to take full advantage of the comprehensive visibility and the rich investigation tools available in Microsoft 365 Defender. This blog shows that many of the existing capabilities in Microsoft 365 Defender help address this attack, but the unique scenarios created by the threat resulted in some Solorigate-specific detections and other innovative protections, including ones that are made possible by deeply integrated cross-domain threat defense.

For additional information and further guidance, refer to these Microsoft resources:

Microsoft will continue to provide public information about the patterns and techniques of this attack and related intelligence for customers to defend themselves, in addition to enhancing the protection capabilities of Microsoft security solutions.

Appendix: Additional details for detection and hunting

Detection details

Attack stage	Microsoft 365 Defender detection or alert
Initial access	Microsoft Defender for Endpoint: • 'Solorigate' high-severity malware was detected/blocked/prevented (Trojan:MSIL/Solorigate.BR!dha) • SolarWinds Malicious binaries associated with a supply chain attack
Execution and persistence	Microsoft Defender for Endpoint: • 'Solorigate' high-severity malware was detected/blocked/prevented (Trojan:Win64/Cobaltstrike.RN!dha, Trojan:PowerShell/Solorigate.H!dha) • Suspicious process launch by Rundll32.exe • Use of living-off-the-land binary to run malicious code • A WMI event filter was bound to a suspicious event consumer
Command and Control	Microsoft Defender for Endpoint: An active 'Solorigate' high-severity malware was detected/ blocked/prevented (Trojan:Win64/Cobaltstrike.RN!dha)
Defense evasion	Microsoft Defender for Endpoint: Suspicious audit policy tampering
Reconnaissance	Microsoft Defender for Endpoint: • Masquerading Active Directory exploration tool • Suspicious sequence of exploration activities • Execution of suspicious known LDAP query fragments
Credential access	Microsoft Defender for Endpoint: • Suspicious access to LSASS (credential access) • AD FS private key extraction attempt • Possible attempt to access ADFS key material • Suspicious ADFS adapter process created Microsoft Defender for Identity: • Unusual addition of permissions to an OAuth app • Active Directory attributes Reconnaissance using LDAP Microsoft Cloud App Security: Unusual addition of credentials to an OAuth app
Lateral movement	Microsoft Defender for Endpoint • Suspicious file creation initiated remotely (lateral movement) • Suspicious Remote WMI Execution (lateral movement)

Exfiltration	Microsoft Defender for Endpoint • Suspicious mailbox export or access modification • Suspicious archive creation

Advanced hunting queries

Attack stage	Query link in GitHub repo
General	Microsoft Defender for Endpoint Threat and Vulnerability Management: SolarWinds Orion software in your org
Initial access	Microsoft Defender for Endpoint: • Malicious DLLs loaded in memory • Malicious DLLs created in the system or locally • Compromised SolarWinds certificate
Execution	Microsoft Defender for Endpoint: • SolarWinds processes launching PowerShell with Base64 • SolarWinds processes launching CMD with echo • ADFS adapter process spawning: *DeviceProcessEvents* *| where InitiatingProcessFileName =~"Microsoft.IdentityServer.ServiceHost.exe"* *| where FileName in~("werfault.exe", "csc.exe")* *| where ProcessCommandLine !contains ("nameId")*
Command and Control	Microsoft Defender for Endpoint • C2 communications • C2 lookup
Credential access	Azure Active Directory (Microsoft Cloud App Security): • Credentials added to AAD app after admin consent • New access credential added to application or service principal • Domain federation trust settings modified • Add uncommon credential type to application • Service Principal Added To Role
Exfiltration	Exchange Online (Microsoft Cloud App Security): • Mail Items Accessed Throttling Analytic • Mail Items Accessed Anomaly Analytic • OAuth Apps reading mail via GraphAPI anomaly • OAuth Apps reading mail both via GraphAPI and directly