New Russia-affiliated actor Void Blizzard targets critical sectors for espionage

microsoft.com/en-us/security/blog/2025/05/27/new-russia-affiliated-actor-void-blizzard-targets-critical-sectors-for-espionage/

By Microsoft Threat Intelligence May 27, 2025

Executive summary:

Void Blizzard is a new threat actor Microsoft Threat Intelligence has observed conducting espionage operations primarily targeting organizations that are important to Russian government objectives. These include organizations in government, defense, transportation, media, NGOs, and healthcare, especially in Europe and North America. They often use stolen sign-in details that they likely buy from online marketplaces to gain access to organizations. Once inside, they steal large amounts of emails and files. In April 2025, Microsoft Threat Intelligence observed Void Blizzard begin using more direct methods to steal passwords, such as sending fake emails designed to trick people into giving away their login information.

We thank our partners at Netherlands General Intelligence and Security Service (AIVD) and the Netherlands Defence Intelligence and Security Service (MIVD) for the collaboration on investigating Void Blizzard (also known as LAUNDRY BEAR). You can read their statement here. We also thank our partners at the US Federal Bureau of Investigation for their continued collaboration on investigating Void Blizzard targeting.

Microsoft Threat Intelligence Center has discovered a cluster of worldwide cloud abuse activity conducted by a threat actor we track as Void Blizzard (LAUNDRY BEAR), who we assess with high confidence is Russia-affiliated and has been active since at least April 2024. While Void Blizzard has a global reach, their cyberespionage activity disproportionately targets NATO member states and Ukraine, indicating that the actor is likely collecting intelligence to help support Russian strategic objectives. In particular, the threat actor's prolific activity against networks in critical sectors poses a heightened risk to NATO member states and allies to Ukraine in general.

Void Blizzard's cyberespionage operations tend to be highly targeted at specific organizations of interest to the Russian government, including in government, defense, transportation, media, non-governmental organizations (NGOs), and healthcare sectors primarily in Europe and North America. The threat actor uses stolen credentials—which are likely procured from commodity infostealer ecosystems—and collects a high volume of email and files from compromised organizations.

In April 2025, Microsoft Threat Intelligence Center observed Void Blizzard evolving their initial access techniques to include targeted spear phishing for credential theft. While Void Blizzard's tactics, techniques, and procedures (TTPs) are not unique among advanced persistent threat actors or even Russian nation state-sponsored groups, the widespread success of their operations underscores the enduring threat from even unsophisticated TTPs when leveraged by determined actors seeking to collect sensitive information.

In this report, we share our analysis of Void Blizzard's targeting and TTPs, with the goal of enabling the broader community to apply specific detections and mitigation guidance to disrupt and protect against Void Blizzard's operations. We extend our gratitude to our partners at the Netherlands General Intelligence and Security Service (AIVD), the Netherlands Defence Intelligence and Security Service (MIVD), and the US Federal Bureau of Investigation for their collaboration in investigating and raising awareness on Void Blizzard activity and tooling to help organizations disrupt and defend against this threat actor.

Void Blizzard targets

Void Blizzard primarily targets NATO member states and Ukraine. Many of the compromised organizations overlap with past—or, in some cases, concurrent—targeting by other well-known Russian state actors, including Forest Blizzard, Midnight Blizzard, and Secret Blizzard. This intersection suggests shared espionage and intelligence collection interests assigned to the parent organizations of these threat actors. Since mid-2024, Microsoft Threat Intelligence has observed Void Blizzard targeting the following industry verticals, many resulting in successful compromises:

- Communications/Telecommunications
- Defense Industrial Base
- Healthcare
- Education
- Government agencies and services
- Information technology
- Intergovernmental organizations
- Media
- NGOs
- Transportation

Void Blizzard regularly targets government organizations and law enforcement agencies, particularly in NATO member states and especially in countries that provide direct military or humanitarian support to Ukraine. Within Ukraine, Void Blizzard has successfully compromised organizations in multiple sectors, including education, transportation, and defense. In October 2024, Void Blizzard compromised several user accounts at a Ukrainian aviation organization that had been previously

targeted by Russian General Staff Main Intelligence Directorate (GRU) actor Seashell Blizzard in 2022. This targeting overlap reflects Russia's long-standing interest in this organization and, more broadly, in aviation-related organizations since Russia's invasion of Ukraine in 2022. In 2023, another GRU actor, Forest Blizzard, targeted a prominent aviation organization in Ukraine, and since at least August 2024, it has conducted increasing password spray attacks against several NATO member states' air traffic control providers.

Tools, tactics, and procedures

Initial access

Void Blizzard conducts opportunistic yet targeted high-volume cyberoperations against targets of intelligence value to the Russian government. Their operations predominately leverage unsophisticated techniques for initial access such as password spray and using stolen authentication credentials. Microsoft assesses that Void Blizzard procures cookies and other credentials through criminal ecosystems. These credentials are then used to gain access to Exchange and sometimes SharePoint Online for information collection.

In April 2025, we identified a Void Blizzard adversary-in-the-middle (AitM) spear phishing campaign that targeted over 20 NGO sector organizations in Europe and the United States. The threat actor used a typosquatted domain to spoof the Microsoft Entra authentication portal. Use of a typosquatted domain to spoof Microsoft Entra authentication was a newly observed initial access tactic for this threat actor. This new tactic suggests that Void Blizzard is augmenting their opportunistic but focused access operations with a more targeted approach, increasing the risk for organizations in critical sectors.

In this campaign, the threat actor posed as an organizer from the European Defense and Security Summit and sent emails containing messages with a PDF attachment that lured targets with a fake invitation to the Summit.

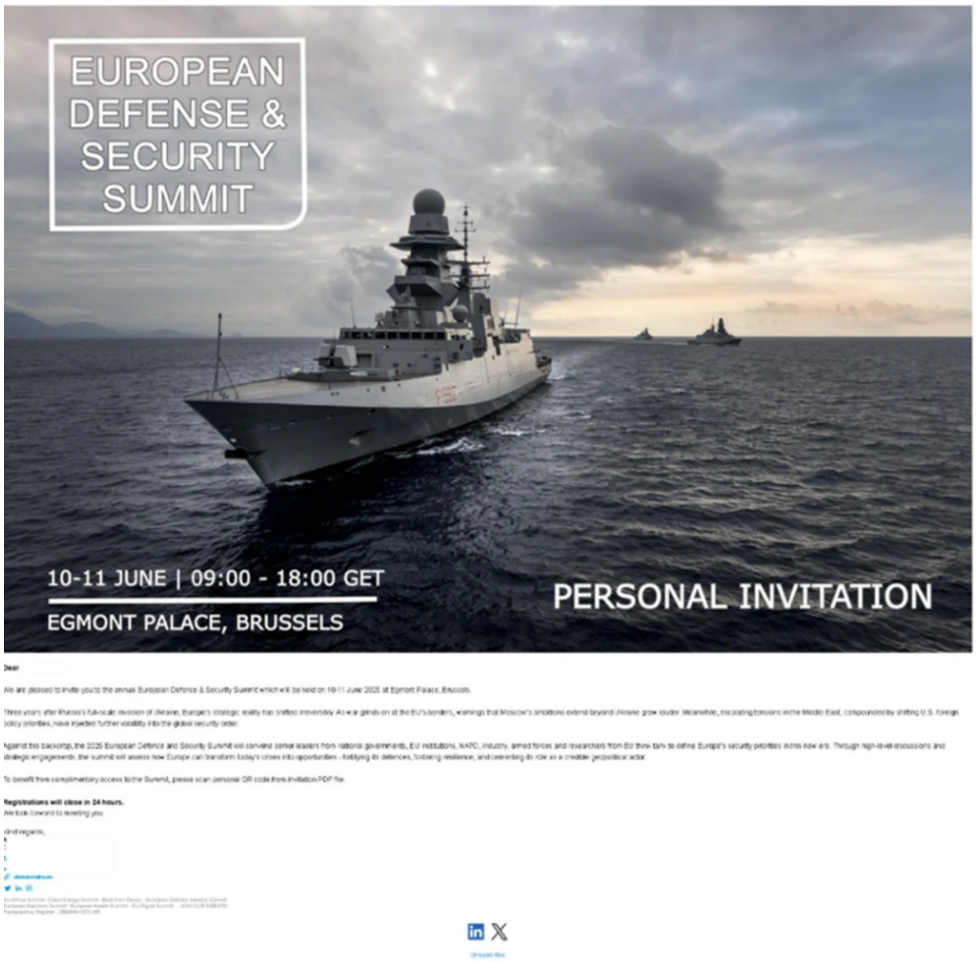

Figure 1. Phishing email body

The attachment contained a malicious QR code that redirected to Void Blizzard infrastructure *micsrosoftonline[.]com*, which hosts a credential phishing page spoofing the Microsoft Entra authentication page. We assess that Void Blizzard is using the open-source attack framework Evilginx to conduct the AitM phishing campaign and steal authentication data, including the input username and password and any cookies generated by the server. Evilginx, publicly released in 2017, was the first widely available phishing kit with AitM capabilities.

Figure 2. PDF attachment with malicious QR code

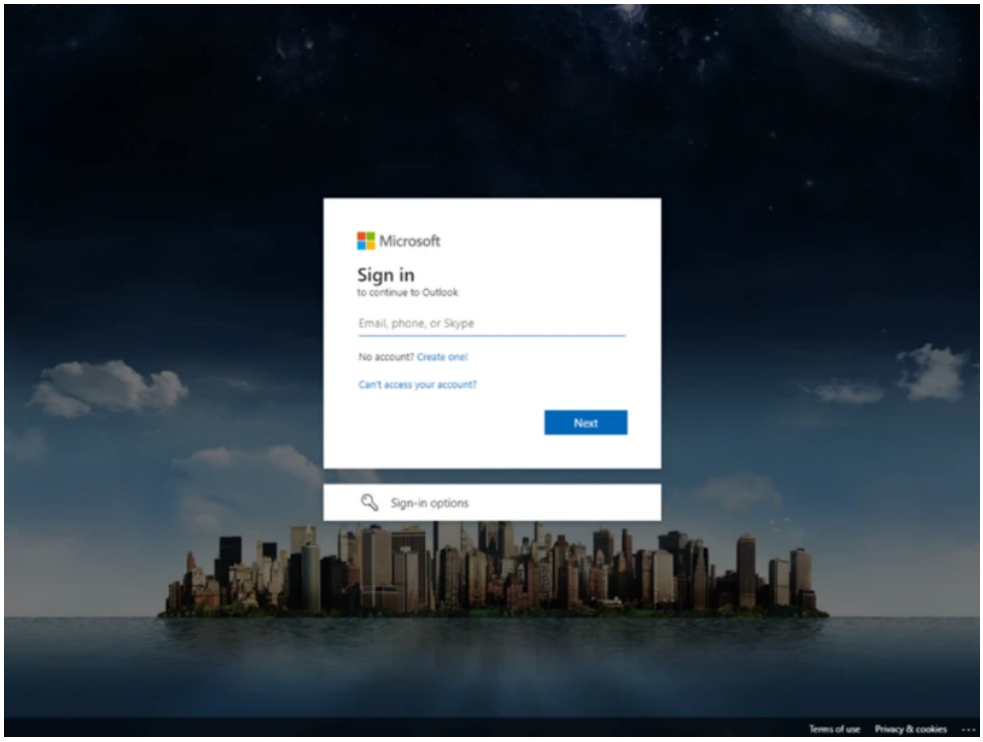

Figure 3. Credential phishing page on actor infrastructure

Post-compromise activity

Despite the lack of sophistication in their initial access methods, Void Blizzard has been effective in gaining access to and collecting information from compromised organizations in critical sectors.

After gaining initial access, Void Blizzard abuses legitimate cloud APIs, such as Exchange Online and Microsoft Graph, to enumerate users' mailboxes, including any shared mailboxes, and cloud-hosted files. Once accounts are successfully compromised, the actor likely automates the bulk collection of cloud-hosted data (primarily email and files) and any mailboxes or file shares that the compromised user can access, which can include mailboxes and folders belonging to other users who have granted other users read permissions.

In a small number of Void Blizzard compromises, Microsoft Threat Intelligence has also observed the threat actor accessing Microsoft Teams conversations and messages via the Microsoft Teams web client application. The threat actor has also in some cases enumerated the compromised organization's Microsoft Entra ID configuration using the publicly available AzureHound tool to gain information about the users, roles, groups, applications, and devices belonging to that tenant.

Mitigation and protection guidance

Microsoft Threat Intelligence recommends organizations that are most likely at risk, primarily those in critical sectors including government and defense, to implement the following recommendations to mitigate against Void Blizzard activity:

Hardening identity and authentication

- Implement a sign-in risk policy to automate response to risky sign-ins. A sign-in risk represents the probability that a given authentication request isn't authorized by the identity owner. A sign-in risk-based policy can be implemented by adding a sign-in risk condition to Conditional Access policies that evaluate the risk level of a specific user or group. Based on the risk level (high/medium/low), a policy can be configured to block access or force multi-factor authentication.
    - When a user is a high risk and Conditional access evaluation is enabled, the user's access is revoked, and they are forced to re-authenticate.
    - For regular activity monitoring, use Risky sign-in reports, which surface attempted and successful user access activities where the legitimate owner might not have performed the sign-in.
- Require multifactor authentication (MFA). While certain attacks attempt to circumvent MFA, implementation of MFA remains an essential pillar in identity security and is highly effective at stopping a variety of threats.

    Leverage phishing-resistant authentication methods such as FIDO Tokens, or Microsoft Authenticator with passkey. Avoid telephony-based MFA methods to avoid risks associated with SIM-jacking.
- Centralize your organization's identity management into a single platform. If your organization is a hybrid environment, integrate your on-premises directories with your cloud directories. If your organization is using a third-party for identity management, ensure this data is being logged in a SIEM or connected to Microsoft Entra to fully monitor for malicious identity access from a centralized location. The added benefits to centralizing all identity data is to facilitate implementation of Single Sign On (SSO) and provide users with a more seamless authentication process, as well as configure Microsoft Entra ID's machine learning models to operate on all identity data, thus learning the difference between legitimate access and malicious access quicker and easier. It is recommended to synchronize all user accounts except administrative and high privileged ones when doing this to maintain a boundary between the on-premises environment and the cloud environment, in case of a breach.
- Secure accounts with credential hygiene: practice the principle of least privilege and audit privileged account activity in your Entra ID environments to slow and stop attackers.

Hardening email security

- Manage mailbox auditing to ensure actions performed by mailbox owners, delegates, and admins are automatically logged. New mailboxes should already have this feature turned on by default.
- Run a non-owner mailbox access report in the Exchange Admin Center to detect unauthorized access onto a mailbox.

Hardening against post-compromise activity

- If a breach or compromise via commodity info stealer is suspected, ensure that any accounts that may have been accessed by that machine have their credentials rotated in addition to removing the malware. Given the widespread use of infostealers in attacks, organizations should immediately respond to infostealer activity and mitigate the risk of credential theft to prevent follow-on malicious activity.

- Conduct an audit search in the Microsoft Graph API for anomalous activity.
- Create Defender for Cloud Apps anomaly detection policies.
- Prevent, detect or investigate possible token theft activity by reviewing mitigation techniques.
- If you suspect password spray activity against your organization's networks, you can refer to this guide for password spray investigation.

Microsoft Defender XDR detections

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

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

Microsoft Defender for Endpoint

The following alert indicates threat actor activity related to Void Blizzard. Note, however, that this alert can be also triggered by Void Blizzard activity that is not related to the activity covered in this report.

Void Blizzard activity

The following alerts might indicate credential theft activity related to Void Blizzard utilizing commodity information stealers or conducting password spraying techniques. Note, however, that these alerts can be also triggered by unrelated threat activity.

- Information stealing malware activity
- Password spraying

Microsoft Defender for Identity

The following Microsoft Defender for Identity alerts can indicate associated threat activity. Note, however, that these alerts can be also triggered by unrelated threat activity.

- Password Spray
- Unfamiliar Sign-in properties
- Atypical travel
- Suspicious behavior: Impossible travel activity

Microsoft Defender for Cloud Apps

The following Microsoft Defender for Cloud Apps alerts can indicate associated threat activity. Note, however, that these alerts can be also triggered by unrelated threat activity.

- Impossible travel
- Activity from suspicious IP addresses
- Unusual activities (by user)

Microsoft Defender for Cloud

The following alerts might also indicate threat activity associated with this threat. These alerts, however, can be triggered by unrelated threat activity and are not monitored in the status cards provided with this report.

- AzureHound tool invocation detected
- Communication with possible phishing domain
- Communication with suspicious domain identified by threat intelligence

Microsoft Entra ID Protection

The following Microsoft Entra ID Protection risk detections inform Entra ID user risk events and can indicate associated threat activity, including unusual user activity consistent with known attack patterns identified by Microsoft Threat Intelligence research. Note, however, that these alerts can be also triggered by unrelated threat activity.

- Anomalous Token (sign-in) (RiskEventType: anomalousToken)
- Password spray (RiskEventType: passwordSpray)
- Anomalous Token (user) (RiskEventType: anomalousToken)
- Attacker in the Middle (RiskEventType: attackerinTheMiddle)
- Activity from Anonymous IP address (RiskEventType: anonymizedIPAddress)
- Microsoft Entra threat intelligence (sign-in): (RiskEventType: investigationsThreatIntelligence)
- Suspicious API Traffic (RiskEventType: suspiciousAPITraffic)

Microsoft Security Copilot

Security Copilot customers can use the standalone experience to create their own prompts or run the following pre-built promptbooks to automate incident response or investigation tasks related to this threat:

- Incident investigation
- Microsoft User analysis
- Threat actor profile
- Threat Intelligence 360 report based on MDTI article
- Vulnerability impact assessment

Note that some promptbooks require access to plugins for Microsoft products such as Microsoft Defender XDR or Microsoft Sentinel.

Threat intelligence reports

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

Microsoft Defender Threat Intelligence

Void Blizzard

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

Hunting queries

Microsoft Defender XDR

Microsoft Defender XDR customers can find related Void Blizzard spear phishing activity related to this threat in their networks by running the following queries.

Possible phishing email targets

The following query can help identify possible email targets of Void Blizzard's spear phishing attempts

```
EmailEvents
| where SenderFromDomain in~ ("ebsumrnit.eu")
| project SenderFromDomain, SenderFromAddress, RecipientEmailAddress, Subject, Timestamp
```

Communication with Void Blizzard domain

The following query can help surface devices that might have communicated with Void Blizzard's spear phishing domain:

```
let domainList = dynamic(["micsrosoftonline.com", "outlook-office.micsrosoftonline.com"]);
union
(
    DnsEvents
    | where QueryType has_any(domainList) or Name has_any(domainList)
    | project TimeGenerated, Domain = QueryType, SourceTable = "DnsEvents"
),
(
    IdentityQueryEvents
    | where QueryTarget has_any(domainList)
    | project Timestamp, Domain = QueryTarget, SourceTable = "IdentityQueryEvents"
),
(
    DeviceNetworkEvents
    | where RemoteUrl has_any(domainList)
    | project Timestamp, Domain = RemoteUrl, SourceTable = "DeviceNetworkEvents"
),
(
    DeviceNetworkInfo
    | extend DnsAddresses = parse_json(DnsAddresses), ConnectedNetworks =
parse_json(ConnectedNetworks)
    | mv-expand DnsAddresses, ConnectedNetworks
    | where DnsAddresses has_any(domainList) or ConnectedNetworks.Name has_any(domainList)
    | project Timestamp, Domain = coalesce(DnsAddresses, ConnectedNetworks.Name), SourceTable
= "DeviceNetworkInfo"
),
(
    VMConnection
    | extend RemoteDnsQuestions = parse_json(RemoteDnsQuestions), RemoteDnsCanonicalNames =
parse_json(RemoteDnsCanonicalNames)
    | mv-expand RemoteDnsQuestions, RemoteDnsCanonicalNames
    | where RemoteDnsQuestions has_any(domainList) or RemoteDnsCanonicalNames
has_any(domainList)
    | project TimeGenerated, Domain = coalesce(RemoteDnsQuestions, RemoteDnsCanonicalNames),
SourceTable = "VMConnection"
),
(
    W3CIISLog
    | where csHost has_any(domainList) or csReferer has_any(domainList)
    | project TimeGenerated, Domain = coalesce(csHost, csReferer), SourceTable = "W3CIISLog"
),
(
    EmailUrlInfo
    | where UrlDomain has_any(domainList)
    | project Timestamp, Domain = UrlDomain, SourceTable = "EmailUrlInfo"
),
(
    UrlClickEvents
    | where Url has_any(domainList)
    | project Timestamp, Domain = Url, SourceTable = "UrlClickEvents"
)
| order by TimeGenerated desc
```

Microsoft Sentinel

The Microsoft blog Web Shell Threat Hunting with Azure Sentinel provides hunting queries and techniques for Sentinel-specific threat hunting. Several hunting queries are also available below.

NOTE: Microsoft Sentinel customers can use the following queries to detect phishing attempts and email exfiltration attempts via Graph API. While these queries are not specific to threat actors, they can help you stay vigilant and safeguard your organization from phishing attacks. These queries search for a week's worth of events. To explore up to 30 days' worth of raw data to inspect events in your network and locate potentially related indicators for more than a week, go to the **Advanced hunting** page > **Query** tab, select the calendar dropdown menu to update your query to hunt for the **Last 30 days**.

If a query provides high value insights into possible malicious or otherwise anomalous behavior, you can create a custom detection rule based on that query and surface those insights as custom alerts. To do this in the Defender XDR portal, run the query in the Advanced hunting page and select **Create detection rule**. To do this in the Sentinel portal, use hunting capabilities to run and view the query's results, then select New alert rule > **Create Microsoft Sentinel alert**.

Campaign with suspicious keywords

In this detection, we track emails with suspicious keywords in subjects.

```
let PhishingKeywords = ()
  {pack_array("account", "alert", "bank", "billing", "card", "change",
"confirmation","login", "password", "mfa", "authorize", "authenticate", "payment", "urgent",
"verify", "blocked");};
  EmailEvents
  | where Timestamp > ago(1d)
  | where EmailDirection == "Inbound"
  | where DeliveryAction == "Delivered"
  | where isempty(SenderObjectId)
  | where Subject has_any (PhishingKeywords())
```

Determine successfully delivered phishing emails to Inbox/Junk folder

This query identifies threats which got successfully delivered to Inbox/Junk folder.

```
EmailEvents
  | where isnotempty(ThreatTypes) and DeliveryLocation in~ ("Inbox/folder","Junk folder")
  | extend Name = tostring(split(SenderFromAddress, '@', 0)[0]), UPNSuffix =
tostring(split(SenderFromAddress, '@', 1)[0])
  | extend Account_0_Name = Name
  | extend Account_0_UPNSuffix = UPNSuffix
  | extend IP_0_Address = SenderIPv4
  | extend MailBox_0_MailboxPrimaryAddress = RecipientEmailAddress
```

Successful sign-in from phishing link

This content is employed to correlate with Microsoft Defender XDR phishing-related alerts. It focuses on instances where a user successfully connects to a phishing URL from a non-Microsoft network device and subsequently makes successful sign-in attempts from the phishing IP address.

```
let Alert_List= dynamic([
    "Phishing link click observed in Network Traffic",
    "Phish delivered due to an IP allow policy",
    "A potentially malicious URL click was detected",
    "High Risk Sign-in Observed in Network Traffic",
    "A user clicked through to a potentially malicious URL",
    "Suspicious network connection to AitM phishing site",
    "Messages containing malicious entity not removed after delivery",
    "Email messages containing malicious URL removed after delivery",
    "Email reported by user as malware or phish",
    "Phish delivered due to an ETR override",
    "Phish not zapped because ZAP is disabled"]);
    SecurityAlert
    | where AlertName in~ (Alert_List)
    //Findling Alerts which has the URL
    | where Entities has "url"
    //extracting Entities
    | extend Entities = parse_json(Entities)
    | mv-apply Entity = Entities on
        (
        where Entity.Type == 'url'
        | extend EntityUrl = tostring(Entity.Url)
        )
    | summarize
        Url=tostring(tolower(take_any(EntityUrl))),
        AlertTime= min(TimeGenerated),
        make_set(SystemAlertId, 100)
        by ProductName, AlertName
    // matching with 3rd party network logs and 3p Alerts
    | join kind= inner (CommonSecurityLog
        | where DeviceVendor has_any  ("Palo Alto Networks", "Fortinet", "Check Point",
"Zscaler")
        | where DeviceProduct startswith "FortiGate" or DeviceProduct startswith  "PAN" or
DeviceProduct startswith  "VPN" or DeviceProduct startswith "FireWall" or DeviceProduct
startswith  "NSSWeblog" or DeviceProduct startswith "URL"
        | where DeviceAction != "Block"
        | where isnotempty(RequestURL)
        | project
            3plogTime=TimeGenerated,
            DeviceVendor,
            DeviceProduct,
            Activity,
            DestinationHostName,
            DestinationIP,
            RequestURL=tostring(tolower(RequestURL)),
            MaliciousIP,
            SourceUserName=tostring(tolower(SourceUserName)),
            IndicatorThreatType,
            ThreatSeverity,
            ThreatConfidence,
            SourceUserID,
            SourceHostName)
        on $left.Url == $right.RequestURL
    // matching successful Login from suspicious IP
    | join kind=inner (SigninLogs
        //filtering the Successful Login
        | where ResultType == 0
```

```
        | project
            IPAddress,
            SourceSystem,
            SigniningTime= TimeGenerated,
            OperationName,
            ResultType,
            ResultDescription,
            AlternateSignInName,
            AppDisplayName,
            AuthenticationRequirement,
            ClientAppUsed,
            RiskState,
            RiskLevelDuringSignIn,
            UserPrincipalName=tostring(tolower(UserPrincipalName)),
            Name = tostring(split(UserPrincipalName, "@")[0]),
            UPNSuffix =tostring(split(UserPrincipalName, "@")[1]))
        on $left.DestinationIP == $right.IPAddress and $left.SourceUserName ==
$right.UserPrincipalName
    | where SigniningTime between ((AlertTime - 6h) .. (AlertTime + 6h)) and 3plogTime
between ((AlertTime - 6h) .. (AlertTime + 6h))
```

Phishing link click observed in network traffic

The purpose of this content is to identify successful phishing links accessed by users. Once a user clicks on a phishing link, we observe successful network activity originating from non-Microsoft network devices.

```
//Finding MDO Security alerts and extracting the Entities user, Domain, Ip, and URL.
    let Alert_List= dynamic([
    "Phishing link click observed in Network Traffic",
    "Phish delivered due to an IP allow policy",
    "A potentially malicious URL click was detected",
    "High Risk Sign-in Observed in Network Traffic",
    "A user clicked through to a potentially malicious URL",
    "Suspicious network connection to AitM phishing site",
    "Messages containing malicious entity not removed after delivery",
    "Email messages containing malicious URL removed after delivery",
    "Email reported by user as malware or phish",
    "Phish delivered due to an ETR override",
    "Phish not zapped because ZAP is disabled"]);
    SecurityAlert
    |where ProviderName in~ ("Office 365 Advanced Threat Protection", "OATP")
    | where AlertName in~ (Alert_List)
    //extracting Alert Entities
     | extend Entities = parse_json(Entities)
    | mv-apply Entity = Entities on
    (
    where Entity.Type == 'account'
    | extend EntityUPN = iff(isempty(Entity.UserPrincipalName), tostring(strcat(Entity.Name,
"@", tostring (Entity.UPNSuffix))), tostring(Entity.UserPrincipalName))
    )
    | mv-apply Entity = Entities on
    (
    where Entity.Type == 'url'
    | extend EntityUrl = tostring(Entity.Url)
    )
    | summarize
AccountUpn=tolower(tostring(take_any(EntityUPN))),Url=tostring(tolower(take_any(EntityUrl))),
 min(TimeGenerated)by SystemAlertId, ProductName
    // filtering 3pnetwork devices
    | join kind= inner (CommonSecurityLog
    | where DeviceVendor has_any  ("Palo Alto Networks", "Fortinet", "Check Point",
"Zscaler")
    | where DeviceAction != "Block"
    | where DeviceProduct startswith "FortiGate" or DeviceProduct startswith  "PAN" or
DeviceProduct startswith  "VPN" or DeviceProduct startswith "FireWall" or DeviceProduct
startswith  "NSSWeblog" or DeviceProduct startswith "URL"
    | where isnotempty(RequestURL)
    | where isnotempty(SourceUserName)
    | extend SourceUserName = tolower(SourceUserName)
    | project
    3plogTime=TimeGenerated,
    DeviceVendor,
    DeviceProduct,
    Activity,
    DestinationHostName,
    DestinationIP,
    RequestURL=tostring(tolower(RequestURL)),
    MaliciousIP,
    Name = tostring(split(SourceUserName,"@")[0]),
    UPNSuffix =tostring(split(SourceUserName,"@")[1]),
    SourceUserName,
    IndicatorThreatType,
    ThreatSeverity,AdditionalExtensions,
```

```
    ThreatConfidence)on $left.Url == $right.RequestURL and $left.AccountUpn ==
$right.SourceUserName
    // Applied the condition where alert trigger 1st and then the 3p Network activity
execution
    | where AlertTime between ((3plogTime - 1h) .. (3plogTime + 1h))
```

Suspicious URL clicked

This query correlates Microsoft Defender for Office 365 signals and Microsoft Entra ID identity data to find the relevant endpoint event *BrowerLaunchedToOpen* in Microsoft Defender ATP. This event reflects relevant clicks on the malicious URL in the spear phishing email recognized by Microsoft Defender for Office 365.

```
// Some URLs are wrapped with SafeLinks
// Let's get the unwrapped URL and clicks
  AlertInfo
  | where ServiceSource =~ "Microsoft Defender for Office 365"
  | join (
          AlertEvidence
          | where EntityType =="Url"
          | project AlertId, RemoteUrl
      )
      on AlertId
  | join (
          AlertEvidence
          | where EntityType =="MailMessage"
          | project AlertId, NetworkMessageId
      )
      on AlertId
  // Get the unique NetworkMessageId for the email containing the Url
  | distinct RemoteUrl, NetworkMessageId
  | join EmailEvents on NetworkMessageId
  // Get the email RecipientEmailAddress and ObjectId from the email
  | distinct RemoteUrl, NetworkMessageId, RecipientEmailAddress , RecipientObjectId
  | join kind = inner IdentityInfo on $left.RecipientObjectId  == $right.AccountObjectId
  // get the UserSid of the Recipient
  | extend OnPremSid = AccountSID
  | distinct RemoteUrl, NetworkMessageId, RecipientEmailAddress , RecipientObjectId,
OnPremSid
  // Get the Url click event on the recipient device.
  | join kind = inner
      (DeviceEvents
      | where ActionType == "BrowserLaunchedToOpenUrl"| where isnotempty(RemoteUrl)
      | project UrlDeviceClickTime = Timestamp , UrlClickedByUserSid = RemoteUrl,
                  InitiatingProcessAccountSid, DeviceName, DeviceId,
InitiatingProcessFileName
      )
     on $left.OnPremSid == $right.InitiatingProcessAccountSid and $left.RemoteUrl ==
$right.UrlClickedByUserSid
  | distinct UrlDeviceClickTime, RemoteUrl, NetworkMessageId, RecipientEmailAddress,
RecipientObjectId,
      OnPremSid, UrlClickedByUserSid, DeviceName, DeviceId, InitiatingProcessFileName
  | sort by UrlDeviceClickTime desc
```

Anomalies in MailItemAccess by GraphAPI

This query looks for anomalies in mail item access events made by Graph API. It uses standard deviation to determine if the number of events is anomalous.

```
let starttime = 30d;
  let STDThreshold = 2.5;
  let allMailAccsessByGraphAPI = CloudAppEvents
  | where   ActionType == "MailItemsAccessed"
  | where Timestamp between (startofday(ago(starttime))..now())
  | where isnotempty(RawEventData['ClientAppId'] ) and RawEventData['AppId'] has "00000003-
0000-0000-c000-000000000000"
  | extend ClientAppId = tostring(RawEventData['ClientAppId'])
  | extend OperationCount = toint(RawEventData['OperationCount'])
  | project Timestamp,OperationCount , ClientAppId;
  let calculateNumberOfMailPerDay = allMailAccsessByGraphAPI
  | summarize NumberOfMailPerDay =sum(toint(OperationCount)) by
ClientAppId,format_datetime(Timestamp, 'y-M-d');
  let calculteAvgAndStdev=calculateNumberOfMailPerDay
  | summarize avg=avg(NumberOfMailPerDay),stev=stdev(NumberOfMailPerDay) by ClientAppId;
  calculteAvgAndStdev  | join calculateNumberOfMailPerDay on ClientAppId
  | sort by ClientAppId
  |  where NumberOfMailPerDay > avg + STDThreshold * stev
  | project ClientAppId,Timestamp,NumberOfMailPerDay,avg,stev
```

Indicators of compromise

Indicator	Type	Description
micsrosoftonline[.]com	Domain	Actor-controlled spear-phishing domain (Evilginx)
ebsumrnit[.]eu	Domain	Actor-controlled spear-phishing domain (malicious sender)
outlook-office[.]micsrosoftonline[.]com	Domain	Actor controlled spear-phishing domain
06a5bd9cb3038e3eec1c68cb34fc3f64933dba2983e39a0b1125af8af32c8ddb	SHA-256	Malicious email attachment

Learn more

For the latest security research from the Microsoft Threat Intelligence community, check out the Microsoft Threat Intelligence Blog: https://aka.ms/threatintelblog.

To get notified about new publications and to join discussions on social media, follow us on LinkedIn at https://www.linkedin.com/showcase/microsoft-threat-intelligence, on X (formerly Twitter) at https://x.com/MsftSecIntel, and on Bluesky at https://bsky.app/profile/threatintel.microsoft.com.

To hear stories and insights from the Microsoft Threat Intelligence community about the ever-evolving threat landscape, listen to the Microsoft Threat Intelligence podcast: https://thecyberwire.com/podcasts/microsoft-threat-intelligence.