Storm-2372 conducts device code phishing campaign | Microsoft Security Blog

By Microsoft Threat Intelligence

Published: 2025-02-14 · Archived: 2026-04-05 12:54:33 UTC

UPDATE (February 14, 2025): Within the past 24 hours, Microsoft has observed Storm-2372 shifting to using the specific client ID for Microsoft Authentication Broker in the device code sign-in flow. More details below.

Executive summary:

Today we're sharing that Microsoft discovered cyberattacks being launched by a group we call Storm-2372, who we assess with moderate confidence aligns with Russia's interests and tradecraft. The attacks appear to have been ongoing since August 2024 and have targeted governments, NGOs, and a wide range of industries in multiple regions. The attacks use a specific phishing technique called "device code phishing" that tricks users to log into productivity apps while Storm-2372 actors capture the information from the log in (tokens) that they can use to then access compromised accounts. These tokens are part of an industry standard and, while these phishing lures used Microsoft and other apps to trick users, they do not reflect an attack unique to Microsoft nor have we found any vulnerabilities in our code base enabling this activity.

Microsoft Threat Intelligence Center discovered an active and successful device code phishing campaign by a threat actor we track as Storm-2372. Our ongoing investigation indicates that this campaign has been active since August 2024 with the actor creating lures that resemble messaging app experiences including WhatsApp, Signal, and Microsoft Teams. Storm-2372's targets during this time have included government, non-governmental organizations (NGOs), information technology (IT) services and technology, defense, telecommunications, health, higher education, and energy/oil and gas in Europe, North America, Africa, and the Middle East. Microsoft assesses with moderate confidence that Storm-2372 aligns with Russian interests, victimology, and tradecraft.

In device code phishing, threat actors exploit the device code authentication flow to capture authentication tokens, which they then use to access target accounts, and further gain access to data and other services that the compromised account has access to. This technique could enable persistent access as long as the tokens remain valid, making this attack technique attractive to threat actors.

The phishing attack identified in this blog masquerades as Microsoft Teams meeting invitations delivered through email. When targets click the meeting invitation, they are prompted to authenticate using a threat actor-generated device code. The actor then receives the valid access token from the user interaction, stealing the authenticated session.

Because of the active threat represented by Storm-2372 and other threat actors exploiting device code phishing techniques, we are sharing our latest research, detections, and mitigation guidance on this campaign to raise

awareness of the observed tactics, techniques, and procedures (TTPs), educate organizations on how to harden their attack surfaces, and disrupt future operations by this threat actor. Microsoft uses Storm designations as a temporary name given to an unknown, emerging, or developing cluster of threat activity, allowing Microsoft to track it as a unique set of information until we reach high confidence about the origin or identity of the threat actor behind the activity.

Microsoft Threat Intelligence Center continues to track campaigns launched by Storm-2372, and, when able, directly notifies customers who have been targeted or compromised, providing them with the necessary information to help secure their environments. Microsoft is also tracking other groups using similar techniques, including those documented by Volexity in their recent publication.

How does device code phishing work?

A device code authentication flow is a numeric or alphanumeric code used to authenticate an account from an input-constrained device that does not have the ability to perform an interactive authentication using a web flow and thus must perform this authentication on another device to sign-in. In device code phishing, threat actors exploit the device code authentication flow.

During the attack, the threat actor generates a legitimate device code request and tricks the target into entering it into a legitimate sign-in page. This grants the actor access and enables them to capture the authentication—access and refresh—tokens that are generated, then use those tokens to access the target's accounts and data. The actor can also use these phished authentication tokens to gain access to other services where the user has permissions, such as email or cloud storage, without needing a password. The threat actor continues to have access so long as the tokens remain valid. The attacker can then use the valid access token to move laterally within the environment.

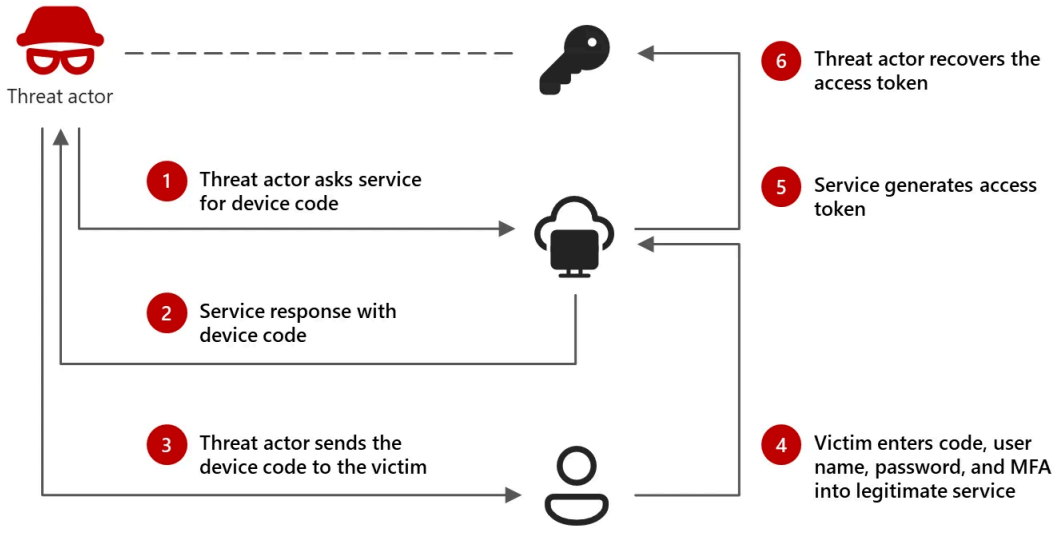

Figure 1. Device code phishing attack cycle

Storm-2372 phishing lure and access

Storm-2372's device code phishing campaign has been active since August 2024. Observed early activity indicates that Storm-2372 likely targeted potential victims using third-party messaging services including WhatsApp,

Signal, and Microsoft Teams, falsely posing as a prominent person relevant to the target to develop rapport before sending subsequent invitations to online events or meetings via phishing emails.

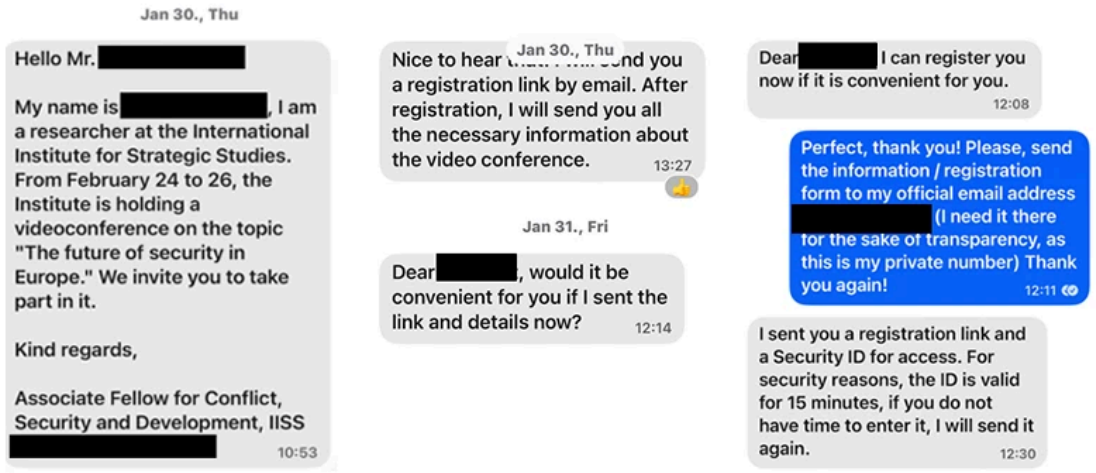

Figure 2. Sample messages from the threat actor posing as a prominent person and building rapport on Signal

The invitations lure the user into completing a device code authentication request emulating the experience of the messaging service, which provides Storm-2372 initial access to victim accounts and enables Graph API data collection activities, such as email harvesting.

Figure 3. Example of lure used in phishing campaign

On the device code authentication page, the user is tricked into entering the code that the threat actor included as the ID for the fake Teams meeting invitation.

Post-compromise activity

Once the victim uses the device code to authenticate, the threat actor receives the valid access token. The threat actor then uses this valid session to move laterally within the newly compromised network by sending additional phishing messages containing links for device code authentication to other users through intra-organizational emails originating from the victim's account.

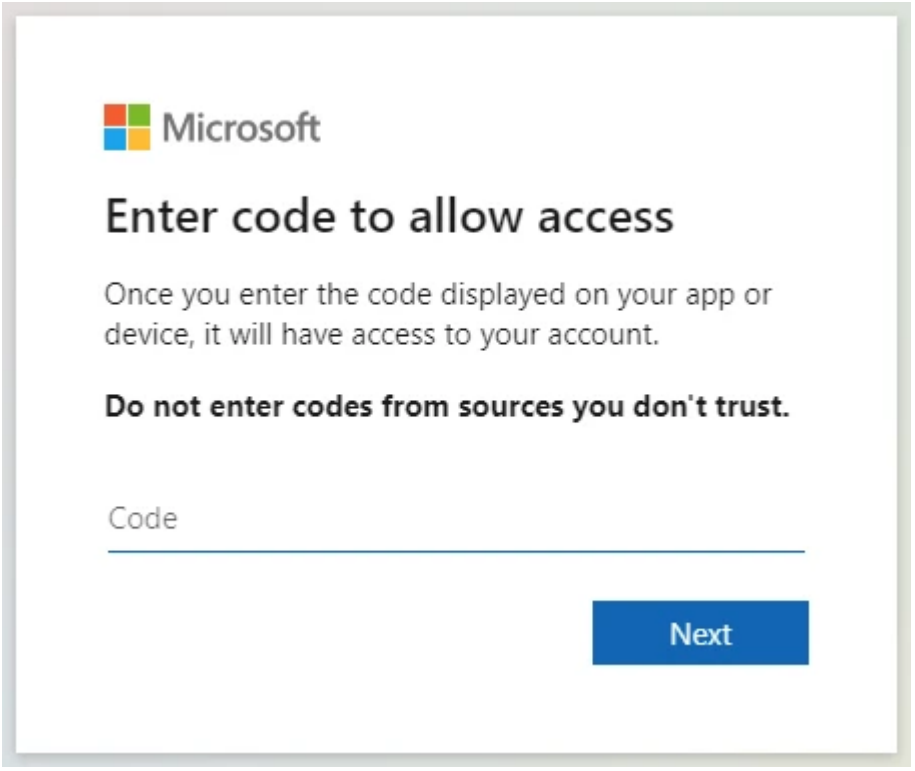

Figure 4. Legitimate device code authentication page

Additionally, Microsoft observed Storm-2372 using Microsoft Graph to search through messages of the account they've compromised. The threat actor was using keyword searching to view messages containing words such as username, password, admin, teamviewer, anydesk, credentials, secret, ministry, and gov. Microsoft then observed email exfiltration via Microsoft Graph of the emails found from these searches.

February 14, 2025 update:

Within the past 24 hours, Microsoft has observed Storm-2372 shifting to using the specific client ID for Microsoft Authentication Broker in the device code sign-in flow. Using this client ID enables Storm-2372 to receive a refresh token that can be used to request another token for the device registration service, and then register an actor-controlled device within Entra ID. With the same refresh token and the new device identity, Storm-2372 is able to obtain a Primary Refresh Token (PRT) and access an organization's resources. We have observed Storm-2372 using the connected device to collect emails.

The actor has also been observed to use proxies that are regionally appropriate for the targets, likely in an attempt to further conceal the suspicious sign in activity.

While many of the mitigations and queries listed below still apply in this scenario, alerts involving anomalous token or PRT activity surrounding close-in-time device registrations may also be a useful method for identifying this shift in technique. Additionally, enrollment restrictions – limiting the user permissions that can enroll devices into your Microsoft Entra ID environment – can also help to address this attack behavior.

Attribution

The actor that Microsoft tracks as Storm-2372 is a suspected nation-state actor working toward Russian state interests. It notably has used device code phishing to compromise targets of interest. Storm-2372 likely initially approaches targets through third-party messaging services, posing as a prominent individual relevant to the target to develop rapport before sending invites to online events or meetings. These invites lure the user into device code authentication that grants initial access to Storm-2372 and enables Graph API data collection activities such as email harvesting.

Storm-2372 targets include government, NGOs, IT services and technology, defense, telecommunications, health, higher education, and energy/oil and gas in Europe, North America, Africa, and the Middle East.

Mitigation and protection guidance

To harden networks against the Storm-2372 activity described above, defenders can implement the following:

- Only allow device code flow where necessary. Microsoft recommends blocking device code flow wherever possible. Where necessary, configure Microsoft Entra ID's device code flow in your Conditional Access policies.
- Educate users about common phishing techniques. Sign-in prompts should clearly identify the application being authenticated to. As of 2021, Microsoft Azure interactions prompt the user to confirm ("Cancel" or "Continue") that they are signing in to the app they expect, which is an option frequently missing from phishing sign-ins.
- If suspected Storm-2372 or other device code phishing activity is identified, revoke the user's refresh tokens by calling *revokeSignInSessions*. Consider setting a Conditional Access Policy to force re-authentication for users.
- 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 evaluates 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.

The following best practices further help improve organizational defenses against phishing and other credential theft attacks:

- Require multifactor authentication (MFA). While certain attacks such as device code phishing attempt to evade 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.

    - Block legacy authentication with Microsoft Entra by using Conditional Access. Legacy authentication protocols do not have the ability to enforce MFA, as legacy MFA (per-user MFA prompts) is susceptible to abuse.
- 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 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.

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 Office 365

Microsoft Defender for Office 365 detects malicious emails, HTML files, and other threat components associated with this attack.

-

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:

- Activity from Anonymous IP address (RiskEventType: anonymizedIPAddress)
- Microsoft Entra threat intelligence (sign-in): (RiskEventType: investigationsThreatIntelligence)

Hunting queries

Microsoft Defender XDR

The following query can help identify possible device code phishing attempts:

```
let suspiciousUserClicks = materialize(UrlClickEvents
    | where ActionType in ("ClickAllowed", "UrlScanInProgress", "UrlErrorPage") or IsClickedThrough != "0"
    | where UrlChain has_any ("microsoft.com/devicelogin", "login.microsoftonline.com/common/oauth2/deviceauth")
    | extend AccountUpn = tolower(AccountUpn)
    | project ClickTime = Timestamp, ActionType, UrlChain, NetworkMessageId, Url, AccountUpn);
//Check for Risky Sign-In in the short time window
let interestedUsersUpn = suspiciousUserClicks
    | where isnotempty(AccountUpn)
    | distinct AccountUpn;
let suspiciousSignIns = materialize(AADSignInEventsBeta
    | where ErrorCode == 0
    | where AccountUpn in~ (interestedUsersUpn)
    | where RiskLevelDuringSignIn in (10, 50, 100)
    | extend AccountUpn = tolower(AccountUpn)
    | join kind=inner suspiciousUserClicks on AccountUpn
    | where (Timestamp - ClickTime) between (-2min .. 7min)
    | project Timestamp, ReportId, ClickTime, AccountUpn, RiskLevelDuringSignIn, SessionId, IPAddress, Url
);
//Validate errorCode 50199 followed by success in 5 minute time interval for the interested user, which suggests a pause to input the code from the phishing email
let interestedSessionUsers = suspiciousSignIns
    | where isnotempty(AccountUpn)
    | distinct AccountUpn;
let shortIntervalSignInAttemptUsers = materialize(AADSignInEventsBeta
```

```
    | where AccountUpn in~ (interestedSessionUsers)
    | where ErrorCode in (0, 50199)
    | summarize ErrorCodes = make_set(ErrorCode) by AccountUpn, CorrelationId, SessionId
    | where ErrorCodes has_all (0, 50199)
    | distinct AccountUpn);
suspiciousSignIns
| where AccountUpn in (shortIntervalSignInAttemptUsers)
```

This following query from public research surfaces newly registered devices, and can be a useful in conjunction with anomalous or suspicious user or token activity:

```
CloudAppEvents
| where AccountDisplayName == "Device Registration Service"
| extend ApplicationId_ = tostring(ActivityObjects[0].ApplicationId)
| extend ServiceName_ = tostring(ActivityObjects[0].Name)
| extend DeviceName = tostring(parse_json(tostring(RawEventData.ModifiedProperties))[1].NewValue)
| extend DeviceId =
tostring(parse_json(tostring(parse_json(tostring(RawEventData.ModifiedProperties))[6].NewValue))[0])
| extend DeviceObjectId_ = tostring(parse_json(tostring(RawEventData.ModifiedProperties))
[0].NewValue)
| extend UserPrincipalName = tostring(RawEventData.ObjectId)
| project TimeGenerated, ServiceName_, DeviceName, DeviceId, DeviceObjectId_, UserPrincipalName
```

Microsoft Sentinel

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:

- Campaign with suspicious keywords
- Determine Successfully Delivered Phishing Emails to Inbox/Junk folder.
- Successful Signin from Phishing Link
- Phishing link click observed in Network Traffic
- Suspicious URL clicked Anomaly of MailItemAccess by GraphAPI
- OAuth Apps accessing user mail via GraphAPI

- OAuth Apps reading mail both via GraphAPI and directly
- OAuth Apps reading mail via GraphAPI anomaly

References

- https://www.blackhillsinfosec.com/dynamic-device-code-phishing/
- https://www.huntress.com/blog/oh-auth-2-0-device-code-phishing-in-google-cloud-and-azure
- https://github.com/secureworks/family-of-client-ids-research?tab=readme-ov-file#which-client-applications-are-compatible-with-each-other

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, and on X (formerly Twitter) at https://x.com/MsftSecIntel.

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.
