Phishing for Codes: Russian Threat Actors Target Microsoft 365 OAuth Workflows

By mindgrub

Published: 2025-04-22 · Archived: 2026-04-05 16:16:24 UTC

Since early March 2025, Volexity has observed multiple suspected Russian threat actors conducting highly targeted social engineering operations aimed at gaining access to the Microsoft 365 (M365) accounts of targeted individuals. This activity comes on the heels of attacks Volexity reported on back in February 2025, where Russian threat actors were discovered targeting users and organizations through Device Code Authentication phishing.

Since that reporting, while there has been an observable drop in Russian threat actors leveraging that specific method, Volexity has observed them pivoting to different methods of attack that abuse other legitimate M365 OAuth authentication workflows. These recently observed attacks rely heavily on one-on-one interaction with a target, as the threat actor must both convince them to click a link *and* send back a Microsoft-generated code.

Volexity is currently tracking what is believed to be at least two Russian threat actors, which it tracks as UTA0352 and UTA0355, that are behind these attacks. It is plausible that there are overlaps between these threat actors and those Volexity previously reported as conducting Device Code Authentication phishing campaigns in January and February 2025. This blog post details the different techniques used by these threat actors and the commonalities between their campaigns, which can be summarized as follows:

- The attacker contacts the victim via a messaging application (Signal, WhatsApp) and invites them to join a video call to discuss the conflict in Ukraine.
- Once the victim has responded, the attacker sends an OAuth phishing URL that they claim is required to join the video call.
- The victim is asked to return the Microsoft-generated OAuth code back to the attacker.
- If the victim shares the OAuth code, the attacker is then able to generate an access token that ultimately allows access the victim's M365 account.

Diplomatic Channels to Nowhere

In March 2025, Volexity learned that some of its customers' staff were receiving suspicious messages via Signal and WhatsApp. The targeted staff members worked at NGOs that support human rights and specifically have expertise and experience working on issues related to Ukraine. The messages claimed to be from European political officials and were themed around discussing matters involving Ukraine. In each observed instance, the call to action was to arrange a meeting between the target and a political official, or Ambassador, of the European country of which the sender claimed to represent.

If the target responded to messages, the conversation would quickly progress towards actually scheduling an agreed upon time for the meeting. However, perhaps in an attempt to not arouse suspicion, in some cases the "Ambassador" would not be immediately available, and the meeting would be scheduled for days later.

As the agreed meeting time approached, the purported European political official would make contact again and share instructions on how to join the meeting. These instructions typically came in the form of a document uploaded into the messaging platform. This "official" would then send a link for the target to click on in order to join the meeting. These shared URLs all pointed to the official login portal for M365 via **login.microsoftonline.com**.

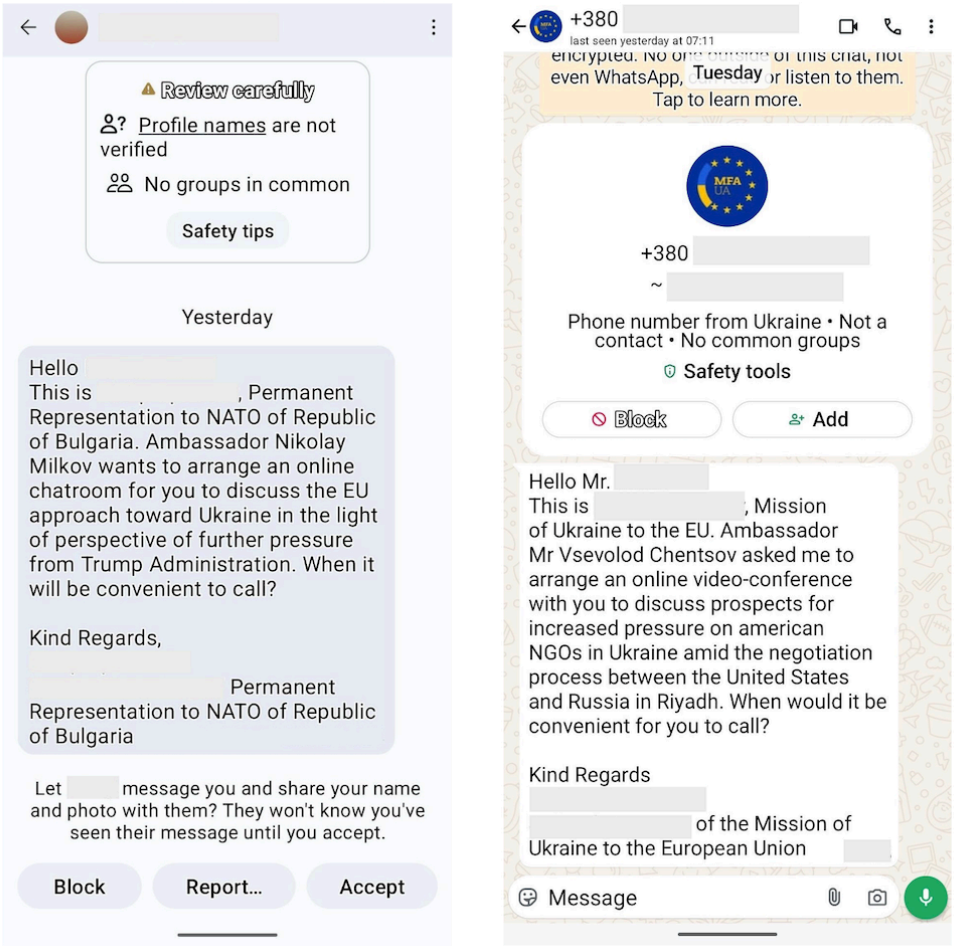
It should come as no surprise that these were, in fact, phishing campaigns, and the supplied instructions involved sending a code back to the attacker. However, unlike the previously observed attacks, these URLs were *not* associated with Device Code Authentication workflows. Instead, these URLs pointed to other Microsoft OAuth 2.0 authentication workflows associated with various legitimate first-party Microsoft applications. Volexity attributes the initial series of attacks observed to a suspect Russian threat actor it tracks as **UTA0352**.

In these campaigns, clicking the link alone would not be enough for the account to be compromised. The code would need to be supplied back to the attacker. The supplied links would redirect to official Microsoft URLs and, in the process, generate a Microsoft Authorization Token that would then appear as part of the URI, and in some cases, within the body of the redirect page. This code could then be used to generate an Access Token, which would grant the holder with access to the account for which it was generated. In multiple observed instances, the attacker would request the code be emailed or sent back via WhatsApp or Signal.

Volexity observed UTA0352 impersonating individuals representing the following countries and affiliations:

- Mission of **Ukraine** to the **European Union**
- Permanent Delegation of the Republic of **Bulgaria** to **NATO**
- Permanent Representation of **Romania** to the **European Union**

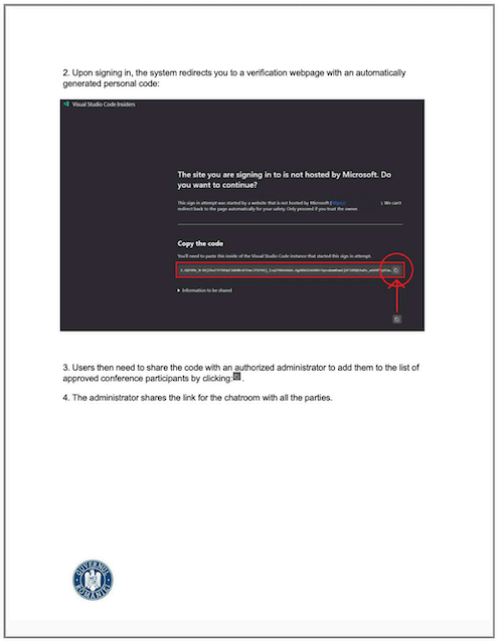
Based on other observations, Volexity believes UTA0352 also likely impersonated officials from **Poland** as well, but Volexity did not observe this directly. The images below show examples of initial outreach messages sent by UTA0352 impersonating various identities on Signal (left) and WhatsApp (right).



Phishing via Visual Studio Code

In mid-March 2025, one of Volexity's customers was contacted by UTA0352 claiming to be a government official from Romania. They attempted to set up a meeting with their Ambassador. Once the user engaged, the attacker sent a message explaining the need to set up a meeting on their web-based "Extended Verification System (EVS)" hosted on their secure servers. This message was accompanied by a PDF file with instructions on what to expect and how to join, after which a maliciously crafted URL was sent to the target.

The image below shows the two-page PDF document purporting to be from the Romanian Ministry of Foreign Affairs.



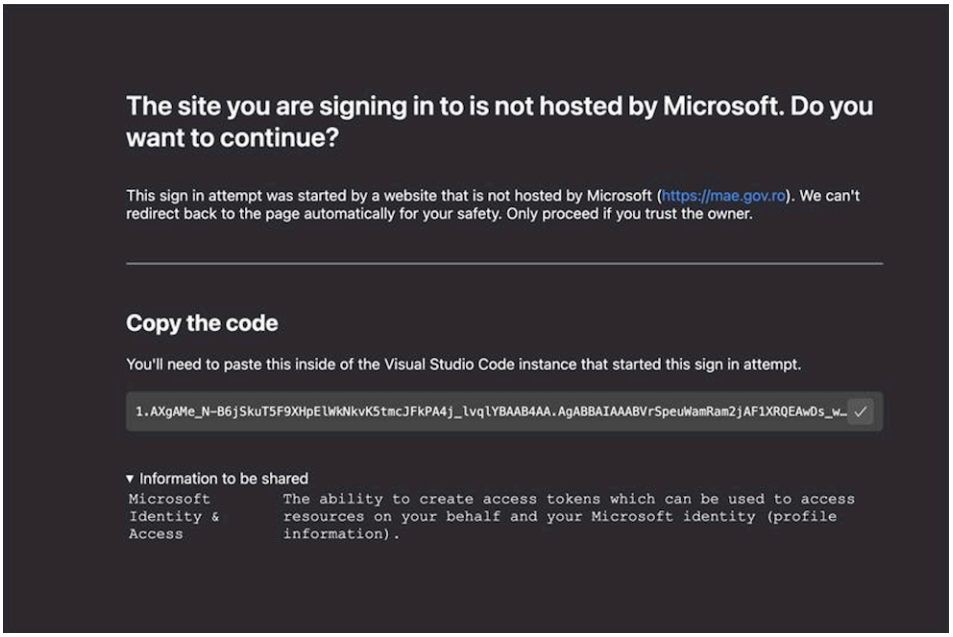
The URL shared by UTA0352 had the following format:

> `https://login.microsoftonline[.]com/organizations/oauth2/v2.0/authorize?state=https://mae.gov[.]ro/[REMOVED]&client_id=aebc6443-996d-45c2-90f0-388ff96faa56&scope=https://graph.microsoft.com/.default&response_type=code&redirect_uri=https://insiders.vscode.dev/redirect&log <EMAIL HERE>`

This URL format is used by M365 to log in to both Microsoft-native/first-party applications and third-party applications using M365's OAuth workflows. The key parameters of the URL are described in the Microsoft OAuth documentation; for convenience, they are briefly described below:

Parameter	Description
`state`	A value to denote the user's state in the application before the request occurred
`client_id`	The application that made the request
`scope`	The access level requested
`response_type`	The method used to send the token back
`redirect_uri`	The handling URI to receive the generated token afterwards

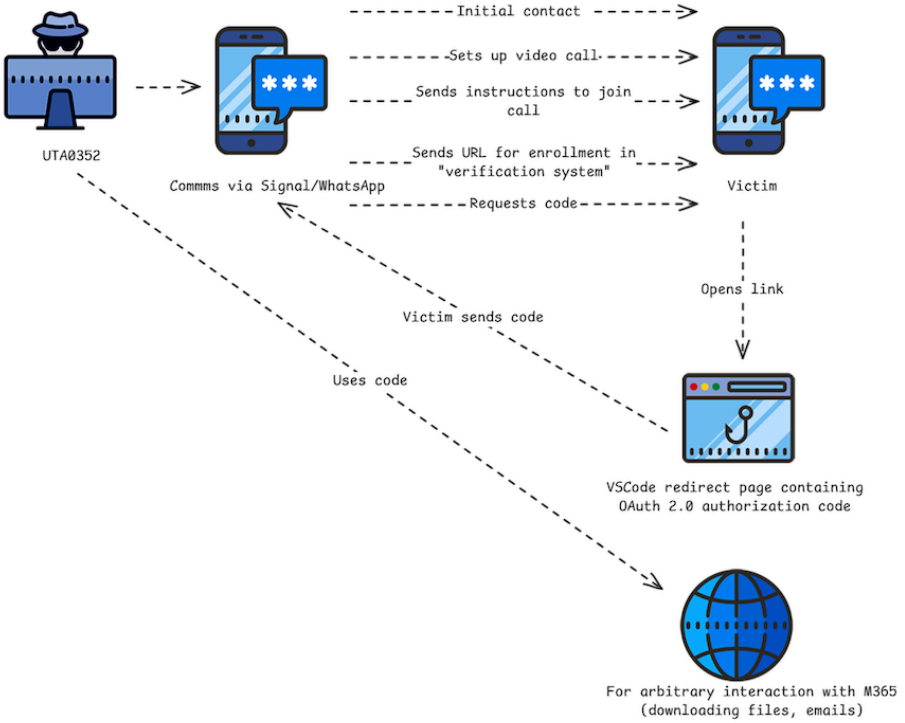
If the user is already logged in with the account specified in the `login_hint` parameter, they will be seamlessly redirected. If the user is not already authenticated, they will be prompted to log in to their M365 account. Once authenticated, the user is redirected to an in-browser version of Visual Studio Code, hosted at **insiders.vscode.dev**. The URL redirects the user to the `/redirect` page, which is designed to receive login parameters from M365, including OAuth. When the user is redirected to this page, they are presented with the following dialog:



The code displayed via the Visual Studio Code dialog is an OAuth 2.0 authorization code that can be used for up to 60 days. This code can be submitted to Microsoft's OAuth workflow for an access token, which can then be used to access the M365 Graph API. Since the original request asked for the user's default access rights, anyone with access to this authorization code also has access to *all resources* normally available to the user. It should be noted that this code also appeared as part of the URI in the address bar. The Visual Studio Code appears to have been set up to make it easier to extract and share this code, whereas most other instances would simply lead to blank pages.

The message shown under the main header uses the `state` value from the prior request, which is commonly used to indicate where the request to authenticate came from. However, as described in Microsoft's documentation, this value is arbitrary. It is up to the handling application (in this case, Visual Studio Code) to decide if and how to use this value. UTA0352 abused this to make it look like an authentication attempt to a Romanian government service. This is a theme repeated in other phishing attacks as an effort to make the links appear legitimate.

The diagram below shows the overall workflow followed by the attacker to target users leveraging the Visual Studio Code first-party application. The workflow varies slightly from other attacks observed later but is fairly similar overall.



Earlier Variations of Visual Studio Code Phishing

In addition to the phishing attack described above, Volexity also identified an older variation of a phishing campaign believed to have been employed by UTA0352. In this earlier campaign, the following URL format was used:

```
hxxps://login.microsoftonline[.]com/common/oauth2/authorize?redirect=https://zoom.us/j/<snip>&client_id=aebc6443-996d-45c2-90f0-388ff96faa56&resource=https://graph.microsoft.com&response_type=code&redirect_uri=https://vscode-redirect.azurewebsites.net&login_hint=<removed>&ui_locales=en-US&mkt=en-US&client-request-id=<removed>
```

The URL differs from the one previously described, in that it employs the format used by the AzureAD v1.0 specification, not v2.0 used in the initially observed campaign. The key differences between the URL parameters used are noted below:

Parameter	Initially Observed Campaign	Earlier Campaign Variation
`redirect_uri`	Uses `insiders.vscode.dev`	Uses `vscode-redirect.azurewebsites.net`.
`resource`	This is the scope parameter in the Oath 2.0 flow	This is the AzureAD v1.0 field used to define the resource for which access is required
`redirect`	Unused in the previously documented campaign	This parameter is included in the request to make it look like the user may be logging into a Zoom call, but it is unused in AzureAD v1.0 authentication workflows

The workflow the older campaign variation initiates is similar to the initially observed campaign. If a user is logged in, they are redirected to **vscode-redirect.azurewebsites.net**, which in turn redirects to a local IP address (127.0.0.1). When this happens, instead of yielding a user interface with the Authorization Code, the code is only available in the URL. The final URL is in the following format:

```
hxxp://127.0.0.1:9217/callback?code=1.ARsAIGLD9ki0FE63WmhS-KbgFENkvK5tmX[snipped]D&session_state=[uuid]
```

This yields a blank page when rendered in the user's browser. The attacker must request that the user share the URL from their browser in order for the attacker to obtain the code.

UTA0355: Phishing via Compromised Ukrainian Government Account

Then, in early April 2025, Volexity identified another new Microsoft 365 OAuth phishing campaign. This time, the campaign started with an email from a legitimate, compromised Ukrainian Government email account, which was then followed by messages sent via Signal and WhatsApp. The emails and follow-up messages invited targets to join a video conference related to Ukraine's efforts "*to investigate and prosecute atrocity crimes, as well as the country's cooperation with international partners in this field.*"

As with previous OAuth phishing techniques that Volexity has reported, the ultimate intention of this campaign is to use the legitimate Microsoft 365 authentication API to gain access to the victim's email data. However, in this campaign, the attacker uses the stolen OAuth authorization code to register a new device to the victim's Microsoft Entra ID (formerly Azure Active Directory) on a *permanent* basis. After registering the device in Entra ID, the attacker then needs to further socially engineer the target into approving a two-factor authentication request in order to gain access to the victim's email.

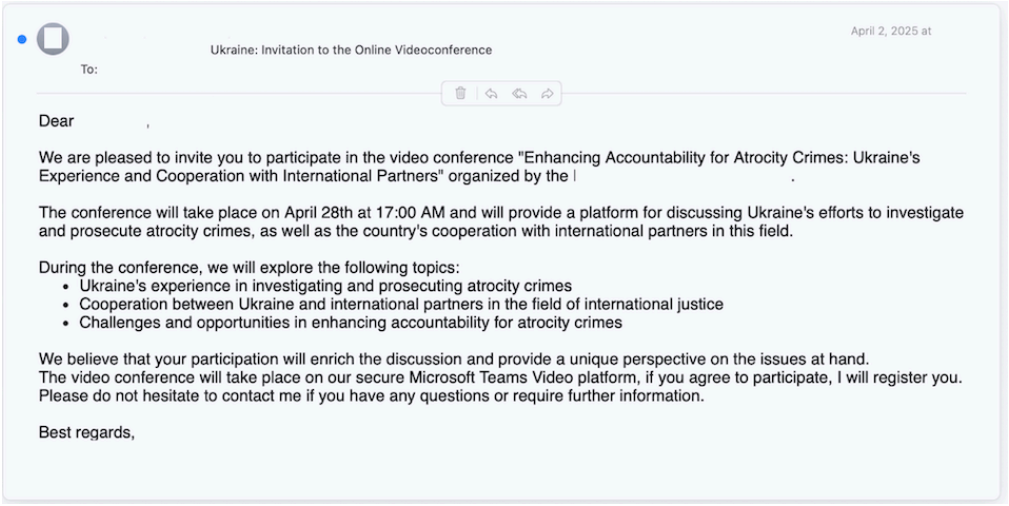
While this campaign uses similar techniques to those employed by UTA0352, Volexity currently tracks these attacks separately and attributes this activity to a threat actor it has labeled **UTA0355**.

Multi-stage Social Engineering

Unlike the attacks that Volexity observed from UTA0352, this new phishing campaign started with an email that was sent to multiple targets. The email invited the targets to a video conference and included the event details. The email did not include any links or instructions, but it did solicit interest from the recipient on if they wished to attend. However, despite the initial outreach coming via email, Volexity found that UTA0355 quickly followed up with each recipient via Signal or WhatsApp, referencing the email that was sent, likely to add legitimacy to their messaging.

Volexity believes that UTA0355 only sent the email from the compromised Ukrainian Government account to individuals for whom it had out-of-band contact information. This was likely to facilitate real-time conversations to assist with social engineering efforts, and to keep information outside of email where it could more easily be discovered or later examined.

The initial email that was sent to various targets is shown in the image below.



April 2, 2025 at

Ukraine: Invitation to the Online Videoconference

To:

Dear ,

We are pleased to invite you to participate in the video conference "Enhancing Accountability for Atrocity Crimes: Ukraine's Experience and Cooperation with International Partners" organized by the .

The conference will take place on April 28th at 17:00 AM and will provide a platform for discussing Ukraine's efforts to investigate and prosecute atrocity crimes, as well as the country's cooperation with international partners in this field.

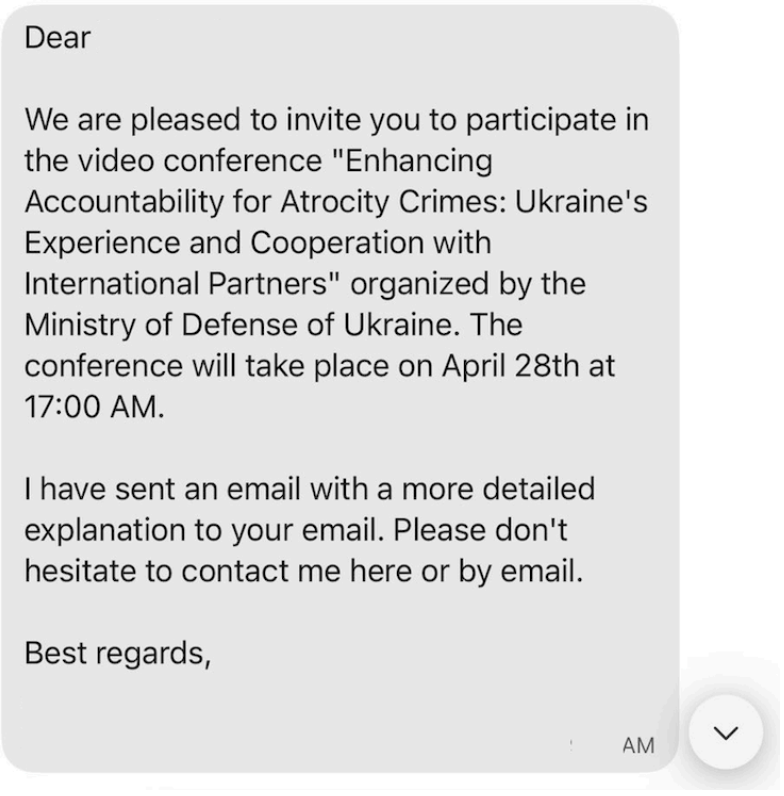
During the conference, we will explore the following topics:

- Ukraine's experience in investigating and prosecuting atrocity crimes
- Cooperation between Ukraine and international partners in the field of international justice
- Challenges and opportunities in enhancing accountability for atrocity crimes

We believe that your participation will enrich the discussion and provide a unique perspective on the issues at hand.
The video conference will take place on our secure Microsoft Teams Video platform, if you agree to participate, I will register you.
Please do not hesitate to contact me if you have any questions or require further information.

Best regards,

Not long after this email was sent to various targets, the follow-up message from UTA0355 referencing the email was sent via Signal or WhatsApp; an example of which is shown below.



Dear

We are pleased to invite you to participate in the video conference "Enhancing Accountability for Atrocity Crimes: Ukraine's Experience and Cooperation with International Partners" organized by the Ministry of Defense of Ukraine. The conference will take place on April 28th at 17:00 AM.

I have sent an email with a more detailed explanation to your email. Please don't hesitate to contact me here or by email.

Best regards,

AM

OAuth Phishing and the Device Registration Service

Like other OAuth phishing techniques, the one used in this campaign involved direct interaction with the victim to have them click a link and supply a code back to the attacker. This code is then sought by the attacker and used to obtain illicit access to M365 resources.

Victim Interaction

If the target responded to the message UTA0355 sent via Signal or WhatsApp, they would be sent an M365 login URL to click; the URL format is shown below:

```
https://login.microsoftonline.com/common/oauth2/authorize?
url=https://teams.microsoft.com/[redacted]&client_id=29d9ed98-a469-4536-ade2-
```

> `f981bc1d605e&resource=01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9&response_type=code&redirect_uri=https%3A%2F%2Flogin.microsoftonline.com%2FWebApp%2FCloudDomainJoin%2F8&amr_values=n<email@address.here>`

After the victim logged in via the shared Microsoft URL, they would be redirected to a new URL that contained an OAuth authorization code within the URL. The attacker included additional instructions indicating the victim should share the URL they see in their address bar after the redirect occurs. The URL generated by the victim's browser, and subsequently returned to UTA0355, follows the format below:

> `https://login.microsoftonline.com/WebApp/CloudDomainJoin/8?code=[redacted]&session_state=[redacted]`

By sharing this URL with the attacker, the victim would unknowingly hand over all the information required to authenticate as themselves. This is similar to other observed and suspected attack campaigns. And again, this does require the target to both click a link and send back a code or URL. However, the victim is only ever asked to interact with legitimate Microsoft 365 services, which users may inherently see as trustworthy. Additionally, it may not immediately appear obvious to a user that sharing data from their address bar, or from text displayed on a legitimate Microsoft webpage, would facilitate granting an attacker access to their M365 account.

UTA0355 OAuth Abuse

The URL the victim would share with the attacker includes a code parameter containing an OAuth authorization code, which would then be used to grant access tokens. Unlike similar previous campaigns, the resource requested in the initial login is not access to the Microsoft Graph API; instead, it is for the Device Registration Service. This service is used by Windows to join new devices to Entra ID. The attacker would use this access to join a new device named `DESKTOP-[redacted]` to the victim's Entra ID. Volexity was able to use the ROADTools project to replicate these steps, and followed this guide to create a new token with full permissions for Microsoft Graph API access. This technique uses a flaw in the Entra ID API design to grant an access token with a greater level of access than that initially granted.

After the initial interaction had taken place, and UTA0355 had registered their device with the victim's Microsoft Entra ID (Azure AD), Volexity observed an additional interaction with the victim. In this interaction, UTA0355 requested that the victim approve a two-factor authentication (2FA) request to "*gain access to a SharePoint instance associated with the conference*". This was required to bypass additional security requirements, which were put in place by the victim's organization, in order to gain access to their email.

Post-compromise Activity

Volexity assesses with high confidence that the attacker required the victim to approve a 2FA request to access email items. In logs reviewed by Volexity, initial device registration was successful shortly after interacting with the attacker. Access to email data occurring the following day, which was when UTA0355 had engineered a situation where their 2FA request would be approved. Once access was granted, logs showed the attacker downloaded the target's email using a session tied to the newly registered device.

The login activity, email access, and device registration all took place using a client IP address belonging to a proxy network that was geolocated to where the victim was located.

Detecting Related Activity

To generally prevent or detect these attacks, Volexity recommends the following:

- Consider alerting on M365 login activity where the Visual Studio Code `client_id` value `aebc6443-996d-45c2-90f0-388ff96faa5` is used in combination with a `resourceDisplayName` containing "Microsoft Graph". Depending on your environment and the usage of Visual Studio Code, this may or may not be feasible, as legitimate users will also login using this `client_id`.
- Consider alerting on the following URL format (either embedded in email or proxy logs). Note that the parameters in the URL can appear in any order, and that the `redirect_uri` values must be set to `insiders.vscode.dev/redirect` or `vscode-redirect.azurewebsites.net`:

> *`https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize?state=[any_url]&client_id=aebc6443-996d-45c2-90f0-388ff96faa56&scope=https://graph.microsoft.com/.default&response_type=code&login_hint=[email]&redirect_uri=https://insiders.vscode.dev/redirect`*

> `https://login.microsoftonline[.]com/common/oauth2/authorize?redirect=[any_url]&client_id=aebc6443-996d-45c2-90f0-388ff96faa56&resource=https://graph.microsoft.com&response_type=code&redirect_uri=https://vscode-redirect.azurewebsites.net&login_hint=[email]&ui_locales=en-US&mkt=en-US&client-request-id=[removed]`

- Evaluate the business impact associated with blocking access to the `insiders.vscode[.]dev` and `vscode-redirect.azurewebsites.net` hostnames and consider implementing such blocks.
- Educate users about the risks associated with new and unknown contacts established through secure messaging platforms. It is crucial that users understand the importance of verifying the identities of contacts that reach out via Signal, WhatsApp, or other secure messaging platforms. Verification should be done out-of-band instead of trusting information provided via unsolicited or unexpected outreach.
- Consider looking for newly registered devices, and correlate with registering IP addresses to look for low-reputation or proxy IP addresses appearing in the `ClientIPAddress`.
    - Be aware that ongoing access to the user's email via the Microsoft Graph API will not contain an attacker IP address in the `ClientIPAddress` field; instead, it contains a Microsoft IP address. This behavior does not appear to be documented and is counterintuitive to security analysis.
    - While attacker activity cannot easily be tracked via the `ClientIPAddress` field, Volexity was able to track attacker activity based on the unique deviceId value associated with the device that was registered by the attacker.
    - The `ClientAppID` field for ongoing access may differ from the user's typical email client, as it will use an AppID corresponding to one created by the attacker.
- Consider implementing conditional access policies that restrict access to organizational resources to only approved or managed devices. This can be effective at preventing device registration and unauthorized access to other resources such as email.
    - At the time of publication, Volexity is not aware of a way to block specific first-party Microsoft apps via conditional access policies. Conditional access can be used to block access to all services for non-compliant devices; however, this may prove challenging for organizations to implement in a short timeframe.
    - It should also be noted that Microsoft Teams was one of the resources Volexity also observed UTA0352 targeting. It is not likely reasonable or feasible for most organizations to block this, if it were even possible.

Conclusion

Motivated threat actors will continuously look for new ways to circumvent security controls and gain access to resources using new methods that are not well known to users or cyber defense teams. This latest series of attacks marks the second time since January 2025 that Russian threat actors have blitzed little-known techniques to obtain access to M365 resources. Volexity surmises that these attacks targeting NGOs, Think Tanks, and human rights defenders may be ramping up in order to capitalize on the current situation these individuals and organizations are facing in the form of budget cuts and reduced staffing.

Similar to the Device Code Authentication phishing campaigns that Volexity reported in February 2025, these recent campaigns benefit from all user interactions taking place on Microsoft's official infrastructure; there is no attacker-hosted infrastructure used in these attacks. Similarly, these attacks do not involve malicious or attacker-controlled OAuth applications for which the user must explicitly grant access (and thus could easily be blocked by organizations). The use of Microsoft first-party applications that already have consent granted has proven to make prevention and detection of this technique rather difficult.

Organizations should train users to be highly vigilant when it comes to unsolicited contact, especially if it arrives via secure messaging apps and request that users click links or open attachments. Further, for this specific type of attack to be successful, the attacker must request that the user send back a URL or code from the link they clicked on. This tactic is not something users are typically trained to be aware of and should be added to an organization's users' security awareness training.

At this time, Volexity notes that NGOs, organizations related to human rights and providing aid and humanitarian assistance, and those with ties to Ukraine are likely the most at risk and potential targets of these campaigns. And while this threat activity is ongoing in limited targeted attacks, Volexity expects that this method of attack will continue and may become more widespread. Therefore, organizations should broadly warn users about this type of attack.

Finally, all messages attributed to UTA0352 and UTA0355 were themed around Ukraine, and targeted numerous individuals and organizations that have historically been targeted by Russian threat actors. Based on this, and the use of similar tactics

observed in February 2025, Volexity assesses with medium confidence that both UTA0352 and UTA0355 are Russian threat actors. It is unclear if they are working in coordination or have overlaps with Volexity's previous reporting.

> **INVESTIGATIVE ASSISTANCE**
>
> If any organization or individual believes they may have been targeted by UTA0352, UTA0355, or in a similar attack, please feel free to reach out to Volexity via our contact form. We would be glad to assess any potential targeting and assist in determining if such an attack may have succeeded.

Acknowledgements

Volexity would like to thank its customers for their vigilance, cooperation, hard work, and dedication to defending human rights. Volexity would also like to thank the MIL.CERT-UA of the Ministry of Defence of Ukraine for their ongoing assistance and cooperation.
