

## SSO to domain resources from Azure AD Joined Devices



**Michael Mardahl**

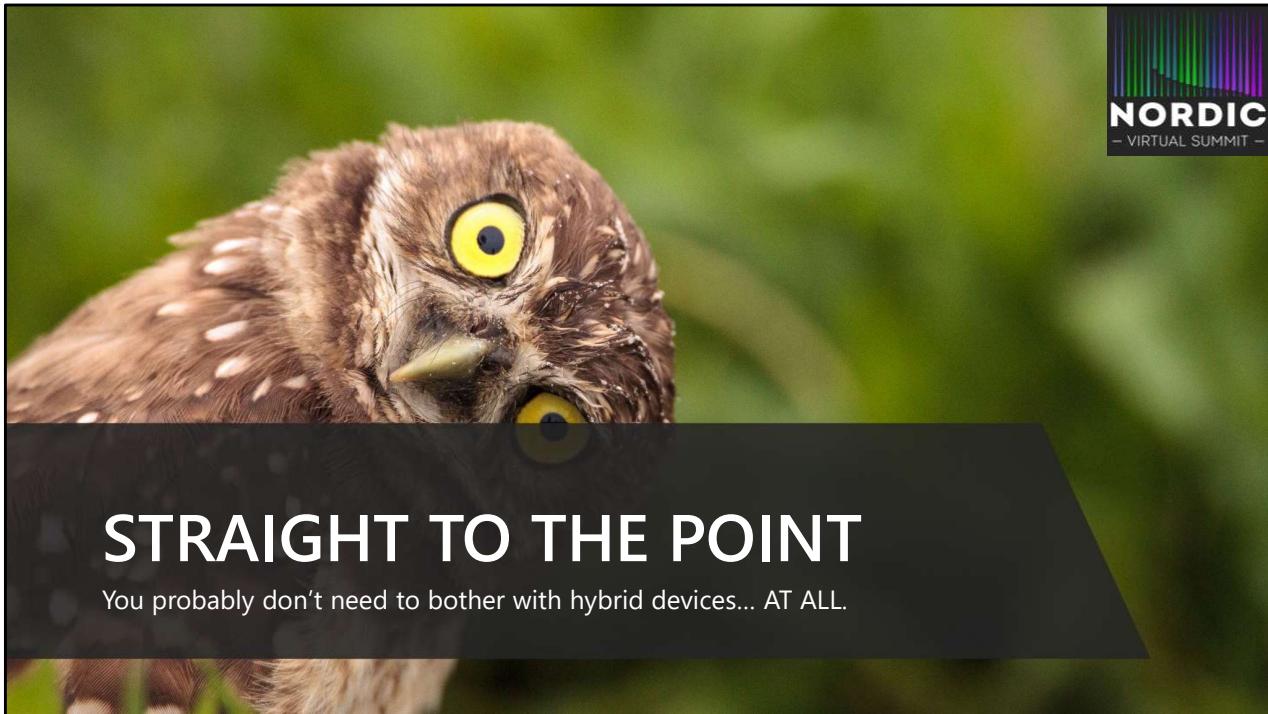
APENTO  
[@michael\\_mardahl](https://twitter.com/michael_mardahl)

**Ben Whitmore**

CloudWay  
[@byteben](https://twitter.com/byteben)

# NORDIC

– VIRTUAL SUMMIT –



## STRAIGHT TO THE POINT

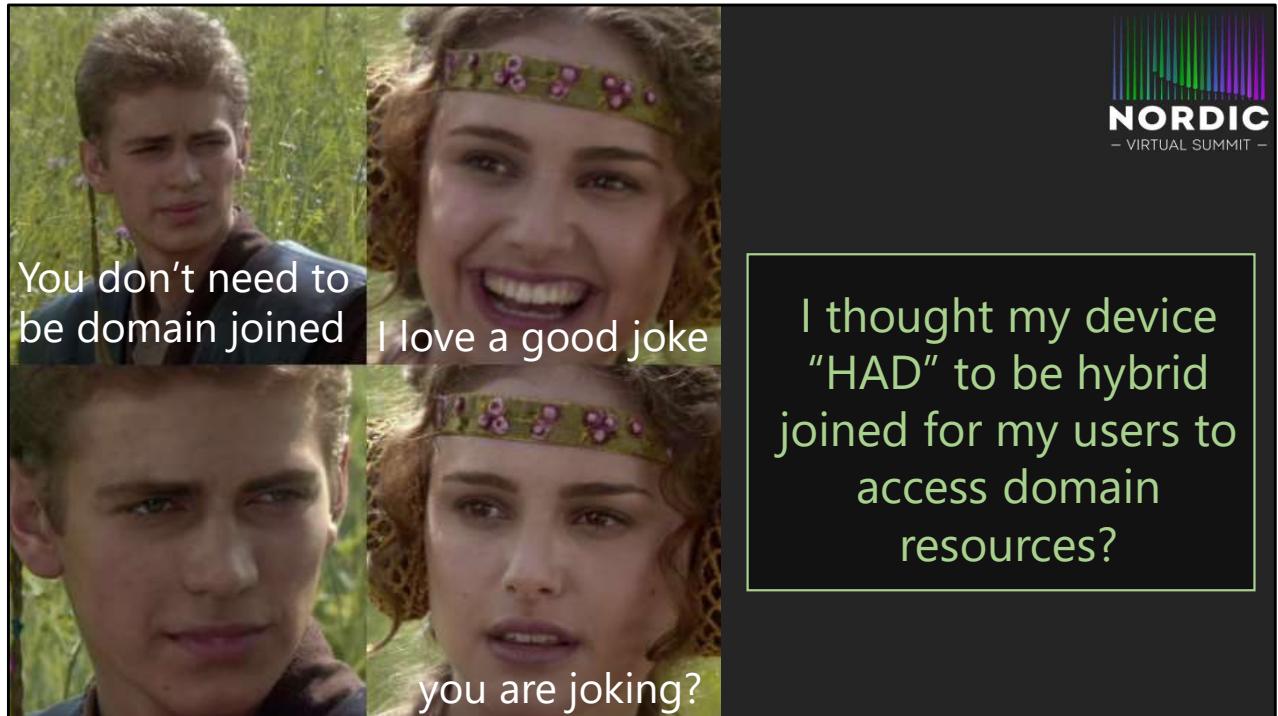
You probably don't need to bother with hybrid devices... AT ALL.

What is hybrid anyway? Why are people doing it?

Talk about moving stuff to cloud but people still need to access legacy on prem stuff

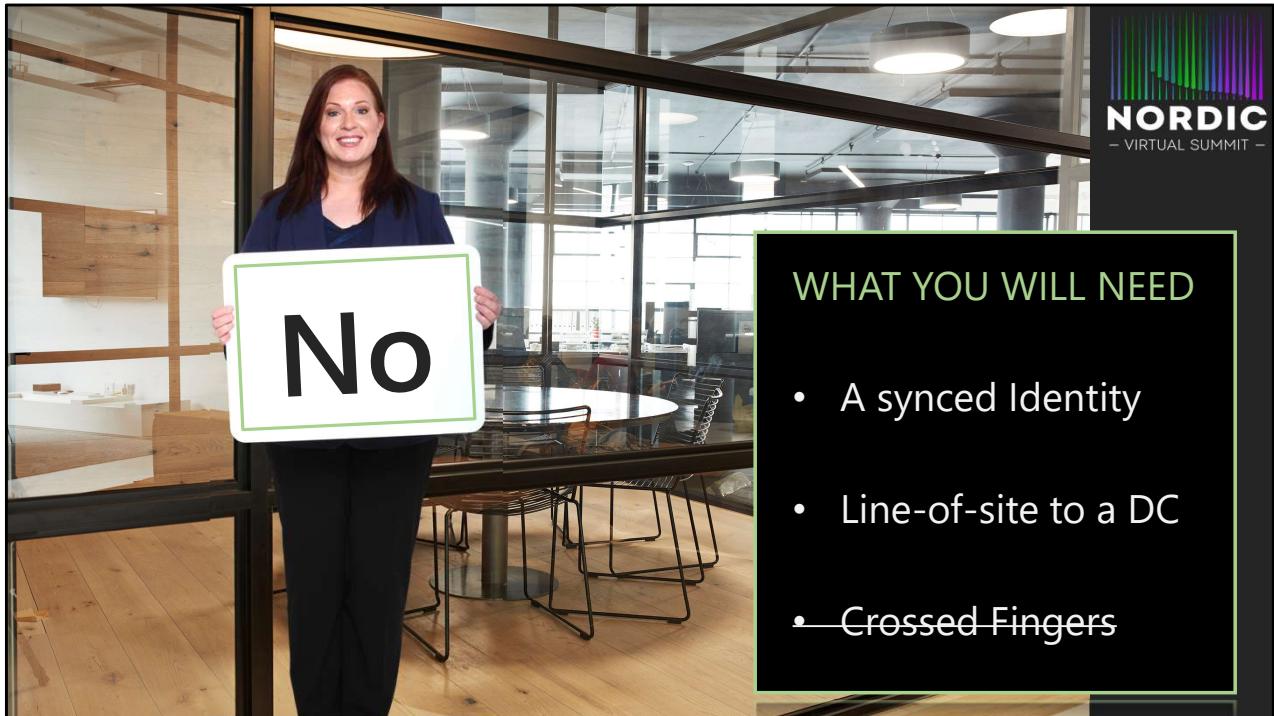
Talk about option during AutoPilot to join devices to domain. Needs VPN, can be messy, lots of failure points. DNS, Certificates, VPN, AADConnect timing

NEXT SLIDE: Your device does not need to be Hybrid Joined



I thought my device  
“HAD” to be hybrid  
joined for my users to  
access domain  
resources?

Talk about device doesn't need to be domain joined.  
Don't go into too much detail – quick slide



## WHAT YOU WILL NEED

- A synced Identity
- Line-of-sight to a DC
- ~~Crossed Fingers~~

No, device does not need to be joined to your domain for the user to access domain resources

Typically, the user requests domain resources – not the device.

There are some scenarios where you “might need” to hybrid join devices. e.g.

- Device needs to exist in AD for legacy app
- Accessing resources across different forests

The only caveat is you need a synced identity, line of sight to a Domain Controller. You don't even need luck (unless your DNS is really bad)

# Why a synced Identity?

Your synced identity knows about your Domain



Our Domain Controller needs to know who we are before it will let us access on-premise resources. How does it know who we are?

The device is not domain joined, and we typically sign in with an Azure AD account

The primary source of Identity needs to be the on-premise Domain – which we kinda know already.

We need to get attributes into Azure AD so when we log on, those attributes come down with the PRT.

We can then use those attributes to begin our SSO journey to domain resources.

# AD Connect



Synchronization Service Manager on BB-APP1

File Tools Actions Help

Operations Connectors Metaverse Designer Metaverse Search

Metaverse Search

Scope by Object Type: All Collation: <default>

Attribute Operator Value

Metaverse Object Properties

Unique identifier (GUID): {A85F0D79-E520-EC11-A291-0015D004105}

Display Name: Barry Badger

Object type: person

Attributes Connectors

Attribute Name	Value	Contributing MA
accountEnabled	true	byteben.com
accountName	barry	byteben.com
cloudAnchor	User_ddc3cf6e-3b3b-461c-a823-5319...	bytebenlab.onmicrosoft.com
cloudSourceAnc...	6C+nyt+XDka+r1ZdNY3Azg==	bytebenlab.onmicrosoft.com
cn	Barry Badger	byteben.com
contributingConn...	{5169c5e4-bf49-45bb-885f-34e4c27be...	byteben.com
countryCode	0	byteben.com
deviceKey	00 02 00 00 20 00 01 A2 51 E2 90 87 ...	bytebenlab.onmicrosoft.com
displayName	Barry Badger	byteben.com
distinguishedName	CN=Barry Badger,OU=Users,OU=LAB,...	byteben.com
domainFQDN	byteben.com	byteben.com
domainNetBIOS	BYTEBEN	byteben.com
forestFQDN	byteben.com	byteben.com
forestNetBIOS	BYTEBEN	byteben.com
givenName	Barry	byteben.com
mail	barry@byteben.com	byteben.com
objectSid	01 05 00 00 00 00 05 15 00 00 00 ...	byteben.com
objectSidString	S-1-5-21-652495413-2945131906-116...	byteben.com
pwdLastSet	20190911074813.0Z	byteben.com

Talk about which attributes are synced up.

# Attributes in AAD



[https://graph.microsoft.com/v1.0/users?\\$filter=DisplayName eq 'Barry Badger'&\\$select=displayName, onPremisesDomainName, onPremisesSamAccountName](https://graph.microsoft.com/v1.0/users?$filter=DisplayName eq 'Barry Badger'&$select=displayName, onPremisesDomainName, onPremisesSamAccountName)

Response preview   Response headers   Code snippets   Toolkit component   Adaptive cards

```
{ "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users(displayName,onPremisesDomainName,onPremisesSamAccountName)", "@odata.count": 1, "value": [ { "displayName": "Barry Badger", "onPremisesDomainName": "byteben.com", "onPremisesSamAccountName": "barry" } ] }
```

These attributes get pulled down with out Primary Refresh Token at logon

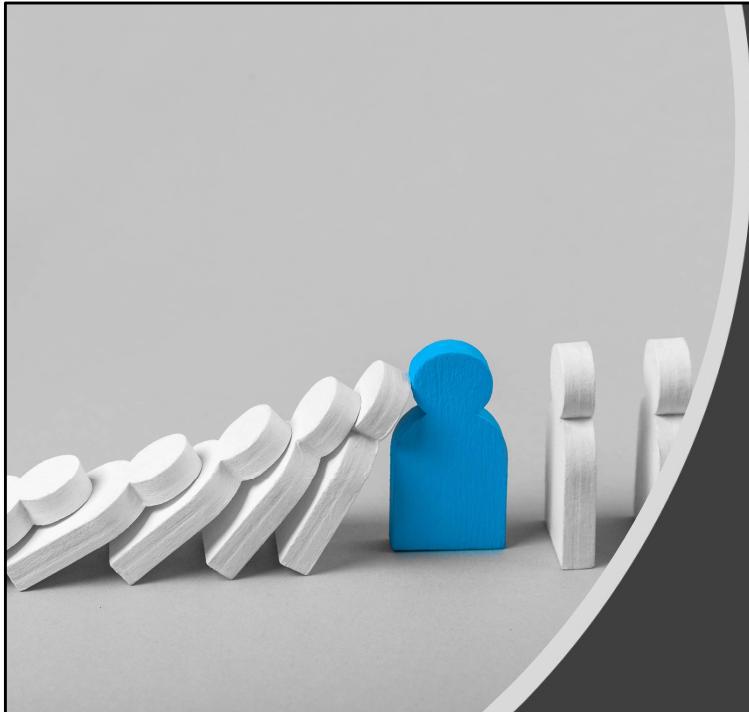


## LAB TIME

show attributes sync in AD Connect and GE



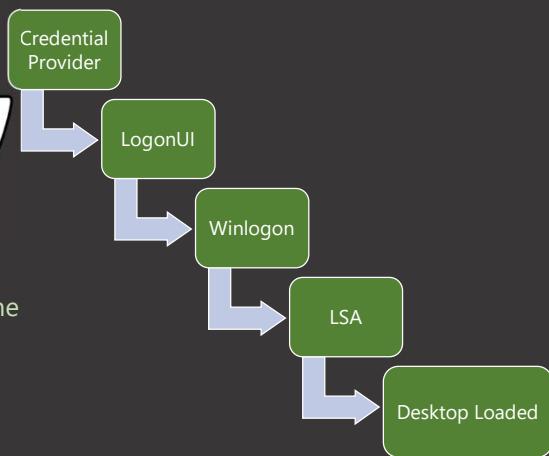
# Now What?



# User logs in to Windows

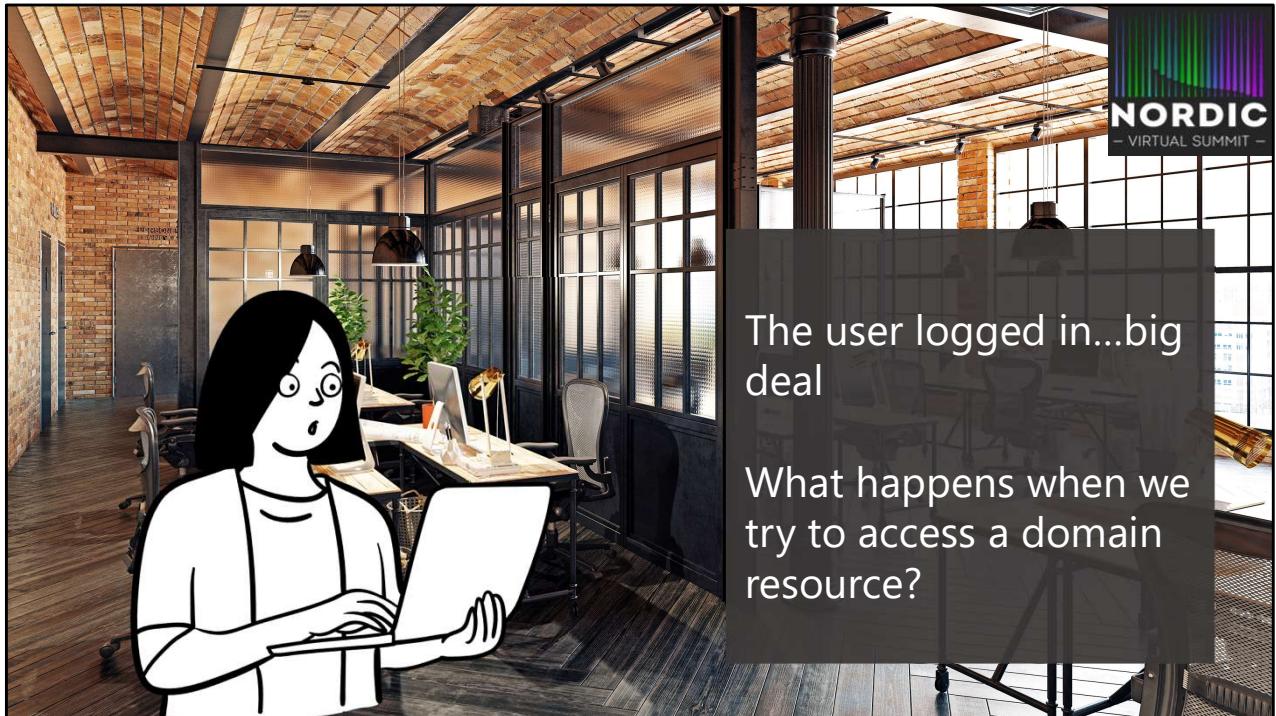


Kim just started something awesome by logging on



Kim, unknowingly, chooses a Credential Provider by entering their username and password and a call is made to the LSA.

Credentials are authenticated and the desktop loads.

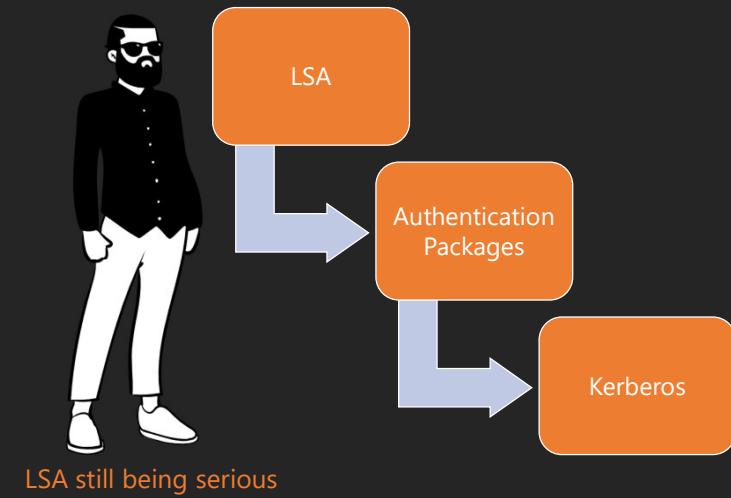


The user logged in...big deal

What happens when we try to access a domain resource?



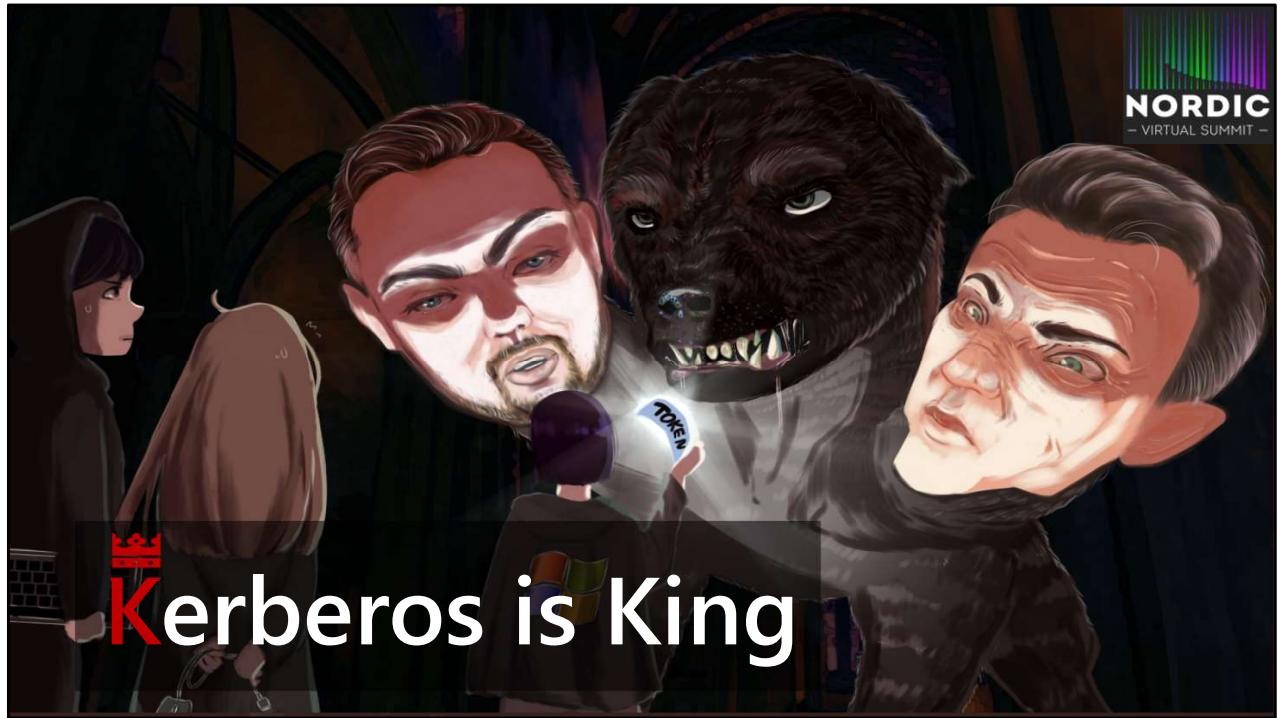
## User accesses a domain resource



The LSA is asking which other authentication packages can use those credentials. The Kerberos Authentication Package puts its hand up.

We have everything we need to contact a Domain Controller – that information came down with our PRT:-

- SamAccountName
- onPremisesDomainName



Kerberos is king here.  
Michael and I and fluffy the dog  
Guards the hades, gates of hell, watchdog of the underworld etc..bit like guarding Windows Vista.  
Came out in the 80's

Kerberos has 3 parts/heads (Michael = User, Fluffy = KDC, Ben = service)

So what is Kerberos and how does it help us?



# Kerberos is King



## Some Kerberos basics

1. "Kerberos" is just a cryptographic ticketing system
2. In order to request tickets to domain resources, the Kerberos protocol needs to know where the "Ticket Master" is
3. The "Ticket Master" is called the Key Distribution Centre (KDC)
4. The KDC in a Windows Domain is a Domain Controller

here are some basics of Kerberos. Basically I need to get a ticket from the domain controller, or KDC, to prove who I am to the service e.g. file server

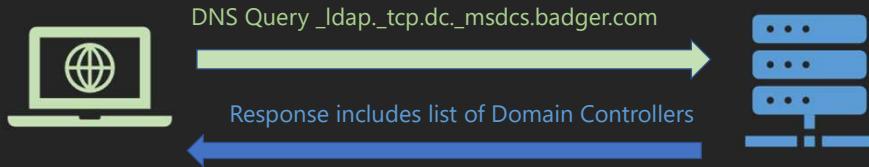


# Kerberos – Finding a KDC



Azure AD joined devices are not aware of your domain, so how do they find the KDC to get tickets?

**DCLOCATOR** is a process used by Windows systems to locate the closest available Domain Controller



First my client needs to find the KDC, how does it do this when the client knows nothing about the domain?

Remember we have some synced attributes. “OnPremise Domain Name”

DCLocator needs a domain hint, which it gets from the `onpremisedomainname` that came down with the PRT.

A Domain Controller (KDC) is then returned to the client

NEXT SLIDE: We can show this manually to prove DC locator works

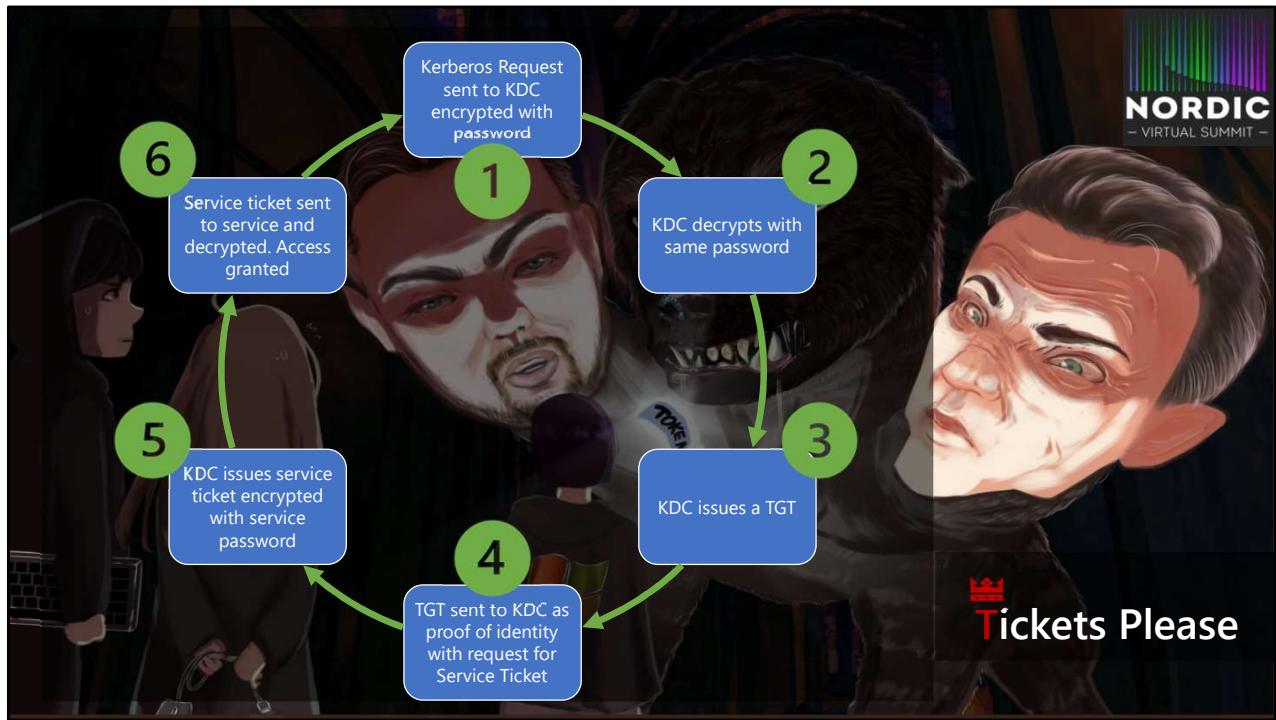
```
PS C:\Users\barry> nltest /dsgetdc:byteben.com
DC: \\bb-dc2.byteben.com
Address: \\192.168.0.3
Dom Guid: bca09d2d-d6fc-45e2-820d-4f10f38985cb
Dom Name: byteben.com
Forest Name: byteben.com
Dc Site Name: Default-First-Site-Name
Our Site Name: Default-First-Site-Name
Flags: PDC GC DS LDAP KDC TIMESERV WRITABLE DNS_DC
SECRET WS DS_8 DS_9 DS_10 KEYLIST
The command completed successfully
PS C:\Users\barry>
```

The command completed successfully

```
PS C:\Users\barry> nltest /dsgetdc:byteben.com
DC: \\bb-dc2.byteben.com
Address: \\192.168.0.3
Dom Guid: bca09d2d-d6fc-45e2-820d-4f10f38985cb
Dom Name: byteben.com
Forest Name: byteben.com
Dc Site Name: Default-First-Site-Name
Our Site Name: Default-First-Site-Name
Flags: PDC GC DS LDAP KDC TIMESERV WRITABLE DNS
SECRET WS DS_8 DS_9 DS_10 KEYLIST
The command completed successfully
PS C:\Users\barry>
```

nltest /dsgetdc:byteben.com

NLTEST – show on Windows 11 device



Now we know where the KDC is, lets go into a bit more detail on how Kerberos tickets work and help us access domain resources

high level

1. Kerberos request sent to the KDC. Request is encrypted with user password.
2. The KDC knows the user password, it's in AD, so it uses it to try and decrypt the request.
3. If it successfully decrypts it, the passwords match so the user is good. The KDC then issues a "Golden" ticket to the user. The golden ticket, or TGT is proof the user authenticated. Authentication has a lot of overhead so we don't want to keep doing it.
4. The user then sends that TGT (proof of they can come into the club) along with another request to access a service, like CIFS (fileshare)
5. The KDC sees the user has a valid TGT (BTW, only the KDC can decrypt the TGT) and issues a "Service Ticket". The service ticket contains things like group membership and service requested.
- 6.



# Kerberos – Tickets please



Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

kerberos

No.	Time	Source	Destination	Protocol	Length	Info
49	3.278658	192.168.0.83	192.168.0.3	KRB5	282	AS-REQ
50	3.280924	192.168.0.3	192.168.0.83	KRB5	255	KRB Error: KRB5KDC_ERR_PREAMUTH_REQUIRED
57	3.281582	192.168.0.83	192.168.0.3	KRB5	362	AS-REQ
60	3.353328	192.168.0.3	192.168.0.83	KRB5	93	AS-REP
67	3.354132	192.168.0.83	192.168.0.3	KRB5	1621	TGS-REQ
69	3.356265	192.168.0.3	192.168.0.83	KRB5	1625	TGS-REP
77	3.356912	192.168.0.83	192.168.0.3	KRB5	1423	TGS-REQ
78	3.357358	192.168.0.3	192.168.0.83	KRB5	1452	TGS-REP
82	3.357681	192.168.0.83	192.168.0.3	SMB2	3225	Session Setup Request
84	3.358441	192.168.0.3	192.168.0.83	SMB2	314	Session Setup Response

The client sends a request for a ticket to the KDC

# Kerberos – Tickets please



# klist tgt

klist tgt allows you to view the specifics of the issued TGT

This TGT will now be used to prove identity for future AS requests in this session



# Kerberos – Tickets please



When we try to access a file share, we see the client send a request to the KDC.

The request contains our TGT which validates who we are.

No.	Time	Source	Destination	Protocol	Length	Info
90	8.658598	192.168.0.83	192.168.0.3	KRB5	1622	TGS-REQ
92	8.660437	192.168.0.3	192.168.0.83	KRB5	1627	TGS-REP
97	8.662352	192.168.0.83	192.168.0.71	SMB2	1890	Session Setup Request
99	8.663902	192.168.0.71	192.168.0.83	SMB2	314	Session Setup Response

```
ticket
  tkt-vno: 5
  realm: BYTEBEN.COM
  sname
    name-type: kRB5-NT-SRV-INST (2)
    sname-string: 2 items
      SNameString: krbtgt
      SNameString: BYTEBEN.COM
  enc-part
```



# Kerberos – Tickets please



kerberos

No.	Time	Source	Destination	Protocol	Length	Info
90	8.658598	192.168.0.83	192.168.0.3	KRBS	1622	TGS-REQ
92	8.660437	192.168.0.3	192.168.0.83	KRBS	1627	TGS-REP
97	8.662352	192.168.0.83	192.168.0.71	SMB2	1890	Session Setup Request
99	8.663902	192.168.0.71	192.168.0.83	SMB2	314	Session Setup Response

```
realm: BYTEBEN.COM
└ sname
  ↘ name-type: kRB5-NT-SRV-INST (2)
    ↘ sname-string: 2 items
      SNameString: cifs
      SNameString: bb-app1.byteben.com
  ↘ enc-part
    etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
    kvno: 16
      ↘ cipher-type: AES256-CTS-HMAC-SHA1-96
      ↘ key-size: 32
      ↘ etype: 18
      ↘ kvno: 16
```

The KDC then issues us a service ticket for the resource we just tried to access.

In this example we accessed a file share on bb-app1.byteben.com



# Kerberos – Tickets please



```
Windows PowerShell
```

```
Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
Start Time: 11/6/2021 7:52:06 (local)
End Time: 11/6/2021 17:52:06 (local)
Renew Time: 11/12/2021 22:05:03 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
Cache Flags: 0x1 -> PRIMARY
Kdc Called: bb-dc2.byteben.com

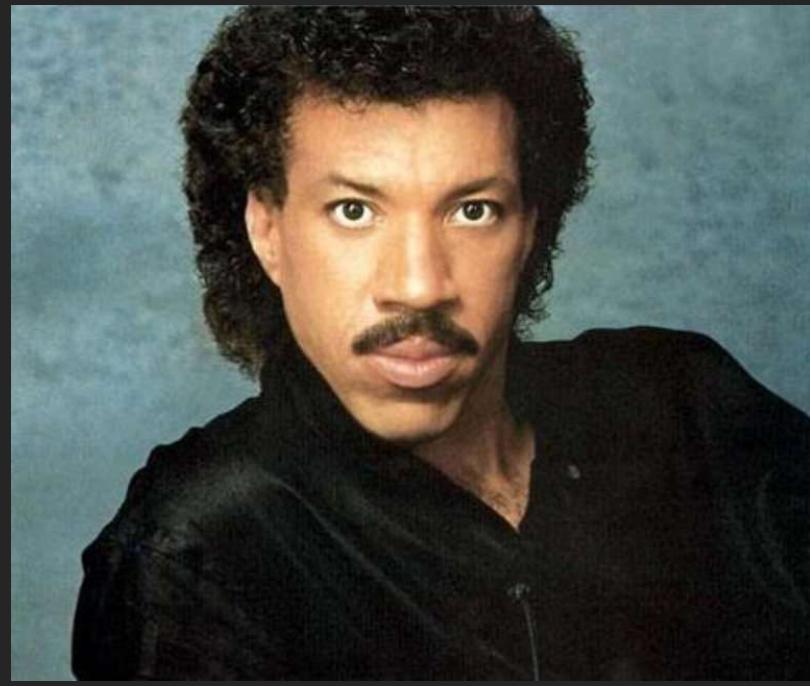
#2>
Client: barry @ BYTEBEN.COM
Server: cifs/bb-app1.byteben.com @ BYTEBEN.COM
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
Start Time: 11/6/2021 14:54:56 (local)
End Time: 11/6/2021 17:52:06 (local)
Renew Time: 11/12/2021 22:05:03 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
Cache Flags: 0
Kdc Called: bb-dc2.byteben.com

#3>
Client: barry @ BYTEBEN.COM
Server: cifs/bb-dc2.byteben.com @ BYTEBEN.COM
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize
Start Time: 11/5/2021 22:05:03 (local)
End Time: 11/6/2021 8:05:03 (local)
Renew Time: 11/12/2021 22:05:03 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
```

klist

klist displays all the service tickets we have been granted

Here is the service ticket we requested to access the file share

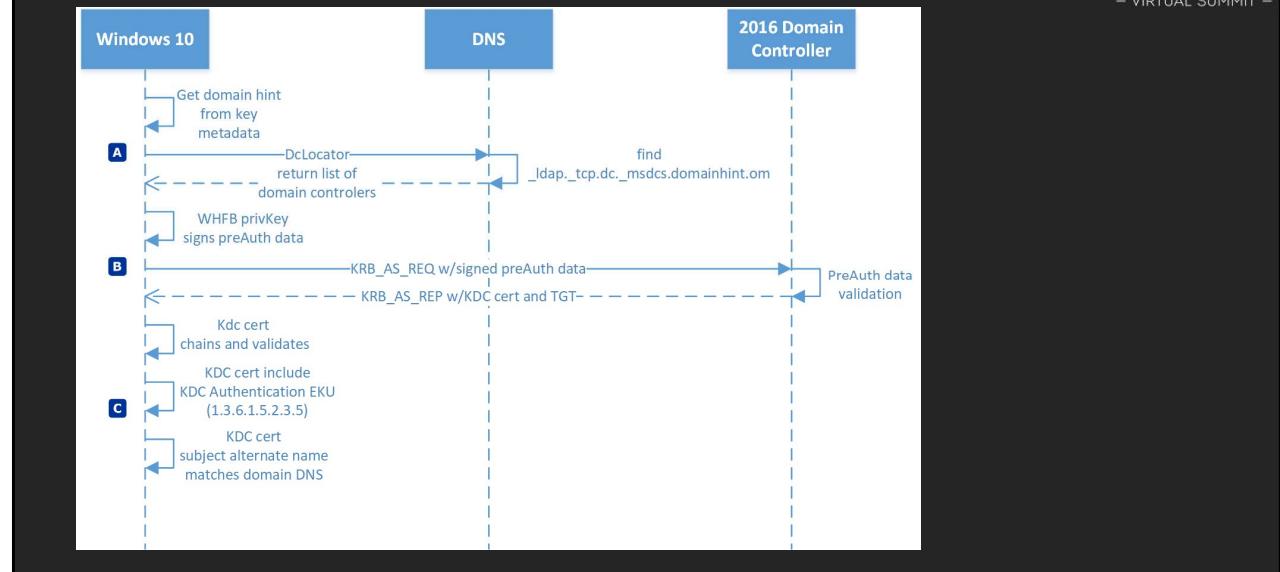


## Windows Hello (is it me you're looking for) for Business

Passwordless SSO to on-prem?

- PREREQUISITES
- <https://docs.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/hello-hybrid-aadj-sso-base>
- Azure Active Directory Connect synchronization (not cloud sync!)
- Device registered in azure ad/preferably also intune managed
- Certificate Revocation List (CRL) Distribution Point (CDP)
- 2016 Domain Controllers
- Domain Controller certificate
- Network infrastructure in place to reach your on-premises domain controller. If the machines are external, this can be achieved using any VPN solution.  
DNS resolution!
- Demo on teamviewer

# Kerberos – Windows Hello



Deepdive on WHfB auth:

<https://docs.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/hello-how-it-works-authentication>

# WHfB - The missing link



The screenshot shows two terminal windows side-by-side. The left window displays 'Device Details' with a JSON object containing fields like DeviceId, Thumbprint, DeviceCertificateValidity, KeyContainerId, KeyProvider, IpmpProtected, and DeviceAuthStatus. The right window displays the 'msDS-KeyCredentialLink' attribute from an LDAP entry, which contains 'deviceKeys' and 'identities' arrays. The 'deviceKeys' array includes a device ID and a long key material string. The 'identities' array includes a sign-in type ('userPrincipalName'), issuer ('auxiliator.onmicrosoft.com'), and issuer-assigned ID ('ler@iphase.dk'). Both windows have several lines of output truncated at the bottom.

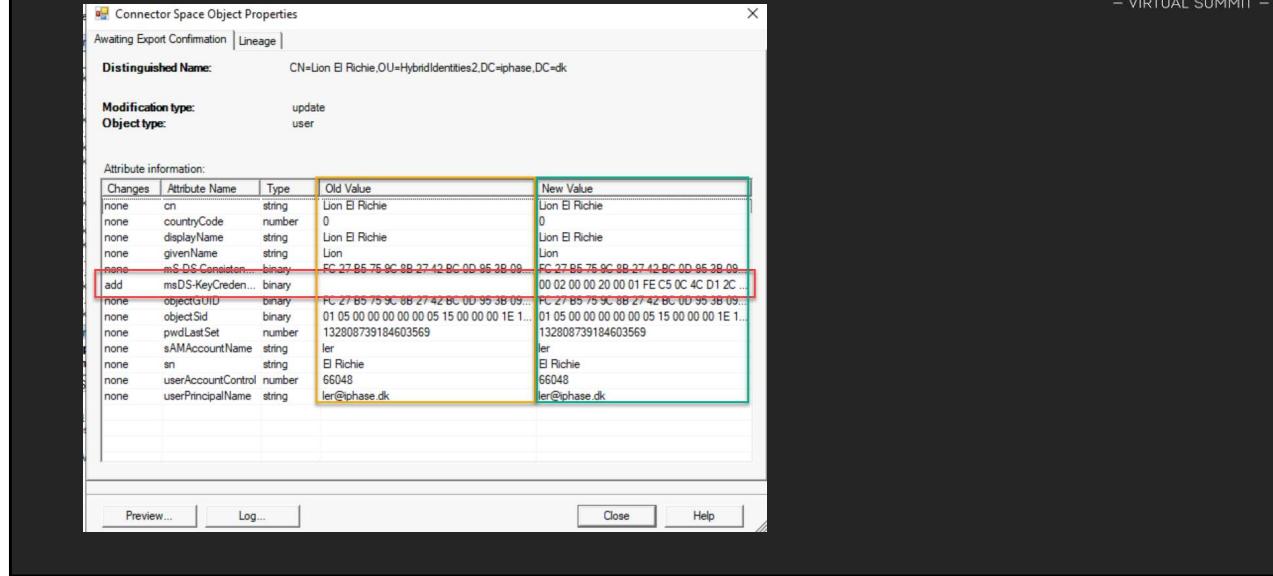
```
Device Details
{
    "DeviceId": "ce37eef7-9791-4608-bc8a-b2580052fc9",
    "Thumbprint": "2C19CD0060FADC0DE70E582D029E43C8BFAC094",
    "DeviceCertificateValidity": [ "2021-11-08 20:28:03.000 UTC -- 2031-11-08 20:50:03.000 UTC" ],
    "KeyContainerId": "99958628-cc22-4874-be9c-6888ad978a0c",
    "KeyProvider": "Microsoft Platform Crypto Provider",
    "IpmpProtected": "YES",
    "DeviceAuthStatus": "SUCCESS"
}

[{"deviceKeys": [
        {
            "deviceId": "ce37eef7-9791-4608-bc8a-b2580052fc9",
            "keyMaterial": "U1NBMDQTAIAADAAAAAFAAAAAAAAQABTSYgAAABXTVZ1VRck+peGr1kk09MhnaeO1G3D77kndD7aP3LmbD9Vnud001u40mC9V70eu0001=61cad+U/Hu6VNC73mYQO7UD1KvXULM4lono/OCnF82ccGdTt5SnMHFI6lgPSymRifj+j6+LwLMr+R0a90Yr2WN//mCD1GnBQ/UU8K2Fymiu51R+090u4puJkUXNFOjEgIna0Q==",
            "keyType": "NGC"
        }
    ],
    "identities": [
        {
            "signInType": "userPrincipalName",
            "issuer": "auxiliator.onmicrosoft.com",
            "issuerAssignedId": "ler@iphase.dk"
        }
    ]
}
```

The msDS-KeyCredentialLink contains the WHfB key used complete the circle of trust.

Not available with “cloud sync” aka. “cloud provisioning”

# WHfB - The missing link



The msDS-KeyCredentialLink contains the WHfB key used complete the circle of trust.

Not available with “cloud sync” aka. “cloud provisioning”



## A moment of silence

For all the hybrid joined devices that are no longer with us after this presentation...

# You have been listening to:



## Ben «007» Whitmore

- Senior Cloud Consultant @ Cloudway
- 20+ years of ITPro experience
- MVP – Enterprise Mobility
- GOV IT experience
- BLOGGER/Writer
  - [MSEndpointMgr.com](http://MSEndpointMgr.com)
  - [byteben.com](http://byteben.com)
- Proud father of 3 girls
- Avid DOCS reader



## Michael Mardahl

- Cloud Architect @ APENTO
- 20+ MS Certifications
- 20+ years of ITPro experience
- MVP – Enterprise Mobility
- ISO27001-LI Certified
- BLOGGER/Writer
  - [MSEndpointMgr.com](http://MSEndpointMgr.com)
  - [MeasureUp!](http://MeasureUp!)
- Proud father to an autistic girl
- A bit bonkers