

# Abusing Azure Active Directory: Who would you like to be today?



@NestoriSyynimaa



# About the author



- Dr. Nestori Syynimaa
  - Owner @ Gerenios Ltd
  - CIO @ Cities surrounding Tampere
- MCITP, MCSA, MCT, MS Certified Expert, CCSK
- Creator of [AADInternals](#)
- [www.linkedin.com/in/nestori](http://www.linkedin.com/in/nestori)
- <http://o365blog.com>

# Contents

- Identity and authentication options in Azure AD / Office 365
- Pass-through authentication (PTA)
- Seamless Single-Sign-On
- Identity federation

# AADInternals

- PowerShell module
- Admin & hacking toolkit for Azure AD & Office 365
- Open source:
  - <https://github.com/gerenios/aadinternals>
  - <http://o365blog.com/aadinternals/>
- Easy to install & use:

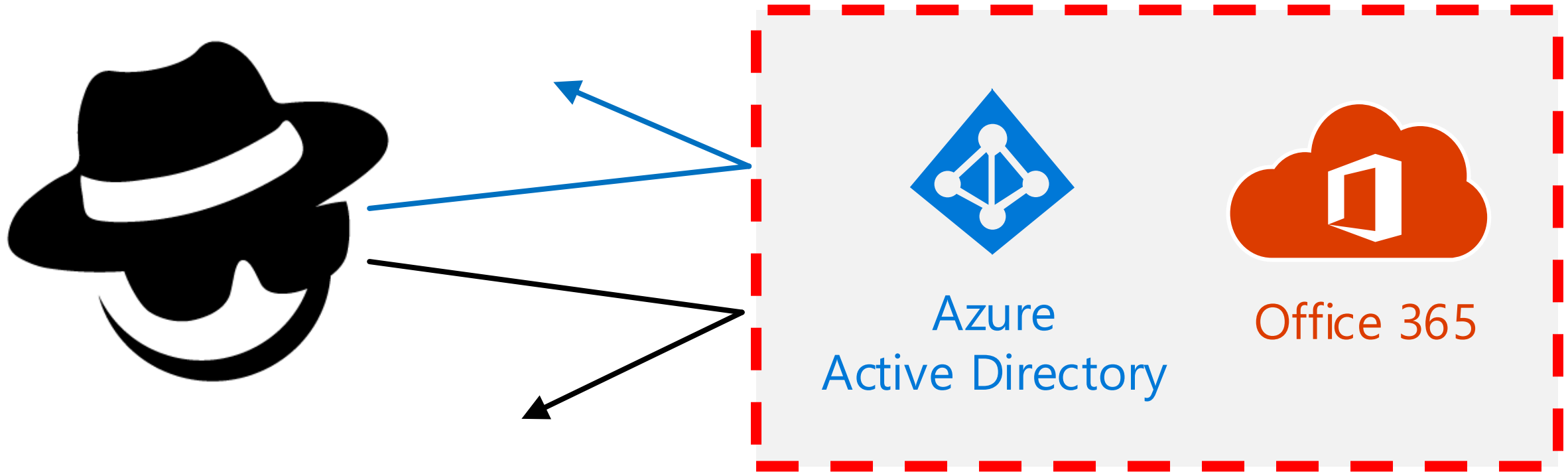
```
C:\PS> Install-Module AADInternals  
C:\PS> Import-Module AADInternals
```

# Identity and authentication options in Azure AD / Office 365

# Identity is the new security perimeter

- One identity can be used with many services
  - Over 2900 apps supports Azure AD identity
- Azure AD identity should be protected in all means!
- Traditional focus on network security doesn't work 😞
  - More sophisticated authentication methods required

The cloud is safe!      How safe is it?



# Identity Options

- Managed

- Authentication performed by Azure AD
  - Against Azure AD
    - Cloud-only
    - Password-Hash Synchronization (PHS) (+ optional Seamless SSO)
  - Against on-prem AD
    - Pass-Through Authentication (PTA) (+ optional Seamless SSO)

- Federated

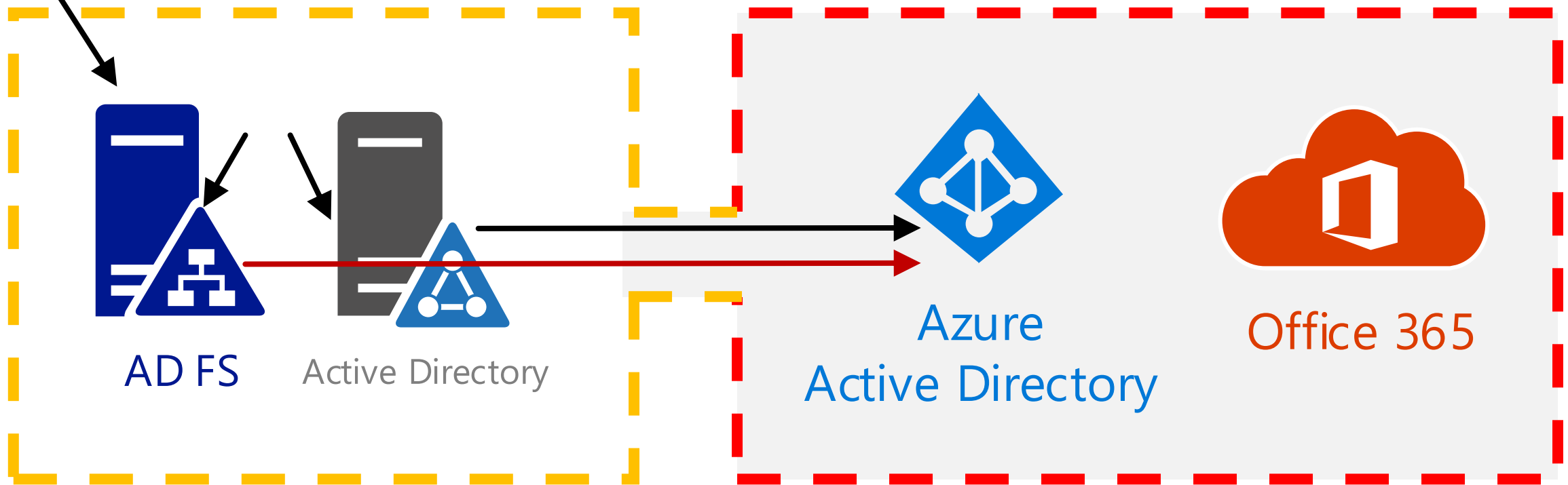
- Authentication performed by Identity Provider (i.e. on-prem AD)

# Azure AD best practices

- Integrate on-premises directory to Azure AD
- Turn on password hash synchronization
- Enable SSO
- Use Azure AD for authenticating apps
- Configure conditional access
- Enable MFA
- Limit the number of admin users

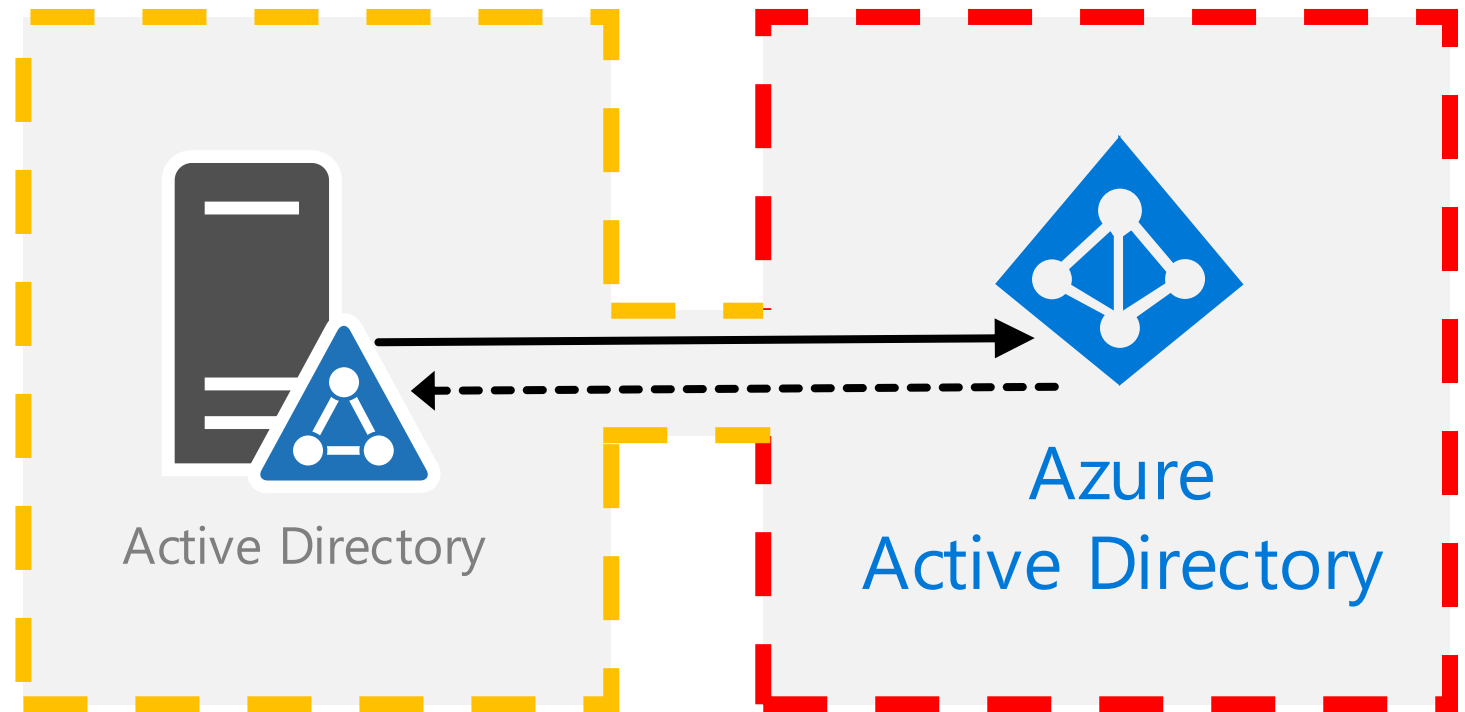
<https://docs.microsoft.com/en-us/azure/security/fundamentals/identity-management-best-practices>

# The cloud is as safe as your on-prem!



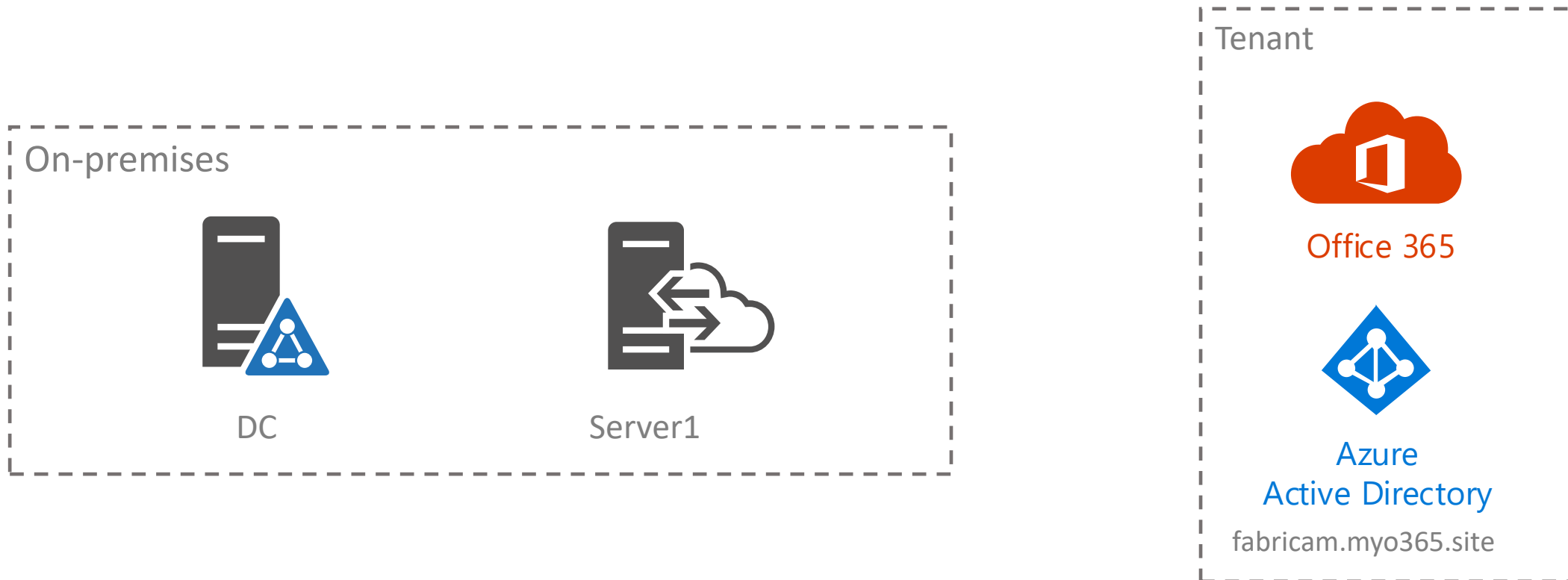
# Azure AD Connect

- Synchronizes objects from on-prem to Azure AD
  - Users & Contacts
  - Groups
  - Devices
  - Password hashes\*
- Writeback\*
  - Groups
  - Passwords
  - Devices
- Configures auth.



\* ) optional

# Demo setup



# Pass-through authentication (PTA)

# Purpose

- To allow users to use on-prem passwords in the cloud
  - No need for extra hardware (cf. Federated Identity)

# How the authentication is performed?

- New authentication request is added to the queue by AAD
- Authentication Agent fetches the request
- User name and password are passed to WIN32 API LogonUserW
  - Returns *true* or *false*
- Azure AD does NOT check:
  - Whether the authentication agent is running on the correct domain

Adam Chester (2019): <https://blog.xpnsec.com/azuread-connect-for-redteam>

# What needed to exploit?

- A compiled DLL (C/C++)
  - Custom implementation of LogonUserW
    - Save the credentials to a log file
    - Let everyone in (returns always true)
  - A “trampoline” to hook LogonUserW to our implementation
- Inject the DLL to Authentication Agent process

Adam Chester (2019): <https://blog.xpnsec.com/azuread-connect-for-redteam>

# Demo



DEEPSEC

# Seamless Single-Sign-On

**DEEP**SEC

# Purpose

- To provide single-sign-on (SSO) to the cloud using Kerberos

# How the authentication is performed?

- Azure AD checks:
  - Server checksum is valid
  - Timestamps are valid
  - User with matching SID exists
- Azure AD does NOT check:
  - User name
  - User display name
  - User principal name
  - Server name
  - Domain name
  - Realm

# What needed to exploit?

- Seamless SSO enabled in Azure AD
- AZUREADSSOACC computer account password
- Target user's SID

# Demo



DEEP SEC

# Identity federation

# Purpose

- To enable using on-prem identities in cloud
- To provide single-sign-on (SSO) using Windows Integrated Authentication (WIA)

# AAD Connect configuration

- Creates an AD FS farm
  - Self-signed certificates for token signing and encryption
    - Encrypted and stored to a configuration database
    - Encryption key stored to an AD object
  - Protects https with the given certificate
  - Adds Azure AD as trusted party and configures claim rules
- Configures Azure AD
  - Converts selected domain to federated
  - Configures domain with AD FS information
    - Login and logout urls
    - Issuer url
    - Public key of token signing certificate

# How the authentication is performed?

- Azure AD checks:
  - Issuer matches the federated domain (AAD wide unique!)
  - Public key matches the federated domain
  - Signature is valid
  - User with matching ImmutableId (aka CloudAnchor) exists
- Azure AD does NOT check:
  - Whether the user's domain matches the federated domain
  - Is the federated domain verified

# How to exploit?

- At least one federated domain in the Azure AD
- The certificate with private key of the federated domain
- The issuer of the federated domain
- Target user's ImmutableId (or ms-DS-ConsistencyGuid)

# Demo



DEEPSEC

TL; DR

DEEPSEC

# TL; DR

- Pass-through authentication (PTA)
  - Authentication agent can be installed on any server
  - Can be used to harvest credentials and create backdoors
- Seamless Single-Sign-On (a.k.a. DesktopSSO)
  - Any domain using Seamless SSO can issue Kerberos tickets for any tenant user
  - Can be used to create backdoors
- Identity federation
  - Any registered IdP can issue SAML tokens for any tenant user
  - Can be used to create backdoors (also using unregistered domains) and bypass MFA

# Thank you!

@NestoriSyynimaa

[linkedin.com/in/nestori](https://www.linkedin.com/in/nestori)

<http://o365blog.com>

**DEEP**SEC

