Azure AD Security

Attack Surface

- External
- Internal – Resource access
- Internal – API access

Azure Active Directory Objects

- Users
- Groups
- Applications
- Service Principals
- Devices
- Roles

AAD Users and Groups

- Standard Identity for Users
  - Internal <alias>@<tenant>.onmicrosoft.com
  - External <alias>_<HomeTenant>#EXT#@<tenant>.onmicrosoft.com
- Groups
  - Logical collections
  - Can be nested
  - Can have members who are not owners of that group

AAD Applications

- Templates to create Service Principals for authentication
- Applications can be single or multi-tenant
- Multi-tenant apps reside in the tenant they were created in, but service principals are created in target tenant
- Object id – represents application (for AD to recognize application)
- Application id – represents application to service principals

AAD Service Principals

- Instance of an application somewhere
- Service account
  - Password
  - Certificates
- Creating a service principal SPN will also create an application

Devices

- Joined – corporate resources
- Registered – external user owned device BYOD

AAD Roles

- Define permissions for other AAD Objects
- Built-in roles have a predefined permissions set
- Custom roles can be added but permissions should be carefully analyzed as often they are too open
- AAD Role Permissions
  - Namespace
  - Object in namespace
  - Possible properties
  - Action

Azure Resource Manager ARM

- Deployment and management service
- Used to be called Azure Service Manager ASM

- VMs
- Databases
- Storage
- Serverless

ARM Terms

- Tenant
  - org
- Subscription
  - Collection of resource groups
  - Account
- Resource group
  - Logical collection of resources
- Resource
  - Managed items inside resource groups
- Resource Provider
  - A service that provides a type of resource
- RBAC
  - A set of permissions for a user on a resource

Cloud Asset Discovery

- https://login.microsoftonline.com/getusrrealm.srf?login=username@college.edu&xml=1
- https://login.microsoftonline.com/<domain>/v2.0/.well-known/openid-configuration

Control and Data Plane

- Control Plane
  - Manage resource with ARM
  - Create/Delete database or storage account
- Data Plane
  - RDP to virtual machine
  - Read/Write to storage account

ARM RBAC

- Authorization system for Azure resources
- RBAC Role
  - Security principal
    - AAD Object
    - User, Group, Service Principal, Managed Identity
  - Role definition – Collection of permissions

Interacting with AAD

- Portal
- Azure cli (Python)
  - Access tokens saved in ~/.azure/accessTokens.json
- Az Powershell
  - AzureRM deprecated
  - Connect-AzAccount
- AzureAD
  - Currently interacts with Microsoft Graph
  - Can be used with Powershell Core
- MDOnline (older)
  - Does not work in PowerShell Core
- Azure REST APIs
  - https://docs.Microsoft.com/en-us/rest/api/azure

User Enumeration

- Basic user enumeration
  - https://login.Microsoft.com/common/oauth2/token
- Detect invalid users while password spraying with:
  - https://github.com/dafthack/MSOLSpray
- Onedrive user enumeration
  - https://github.com/nyxgeek/onedrive_user_enumeration
- General Enumeration
  - https://github.com/initstring/cloud_enum

Password Spraying

- Try one password to avoid lockout
- Attempt to authenticate to each individual account one time every thirty minutes
- MSOLSpray
  - Valid cred
  - MFA enabled
  - Tenant exists
  - User exists
  - Account is locked
  - Account is disabled
  - Password is expired
- Other Tools
  - O365 Creeper
  - MailSniper

Azure AD Passwords

- Azure AD Banned Password Policy
  - Works on DCs must be 2012 or higher
  - Sysvol needs to be using DFSR
  - Deploy in Audit Mode
- https://aka.ms/PasswordSprayBestPractices
- Nearly 100 Percent of password spray attacks which are successful use legacy auth **pop3, imap, etc**
- Modernize password policy
- MS Stats July 2019 122k accounts compromised due to password spray
- AzureAD/O365 IDP is responsible for auth incl legacy auth
  - Block legacy auth in Exchange at mailbox level
  - Block in Exchange online

Bypass Azure Smart Lockout

- Rotate Ips
- User FireProx with MSOLSpray
  - https://github.com/ustayready/fireprox

Azure Authentication

- Password Hash Synchronization
- Pass Through Authentication
- Active Directory Federation Services (ADFS)
- Certificate-based auth
- Conditional access policies
- Long-term access tokens

Password Hash Synchronization

- Azure AD Connect
- Synchronizes AD hashed credentials to Azure
- User can authenticate to Azure service such as Office365 with domain credentials

Pass Through Authentication

- Credentials stored on-prem
- On-prem agent validates authentication requests to Azure AD
- Allows locally stored creds to be used for SSO authentication to Azure apps

Azure AD Connect Info

- Azure AD Connect service account is granted password hash sync rights
- AAD Connect runs on ""AzureSync" which is in the Servers OU
- The Servers OU has 2 GPOs applied
- -"Server Baseline Policy" GPO adds the Server Admins group (in the Groups OU)
- "Server Config" GPO has 3 Server Tier groups with modify rights

Attack Options - Azure AD Connect

- Compromise account that is a member of the Server Admins group or any of the Server Tier groups
- Compromise account delegated right to modify groups in the Groups OU
- On-Prem AD
  - AD user can enumerate all user accounts & admin group memebership with network access toa Domain Controller
- Azure AD
  - Azure AD user can enumerate all user accounts & admin group memberships with access to Office 365 Services (Internet)
  - User enumeration of id’s possible with no account access

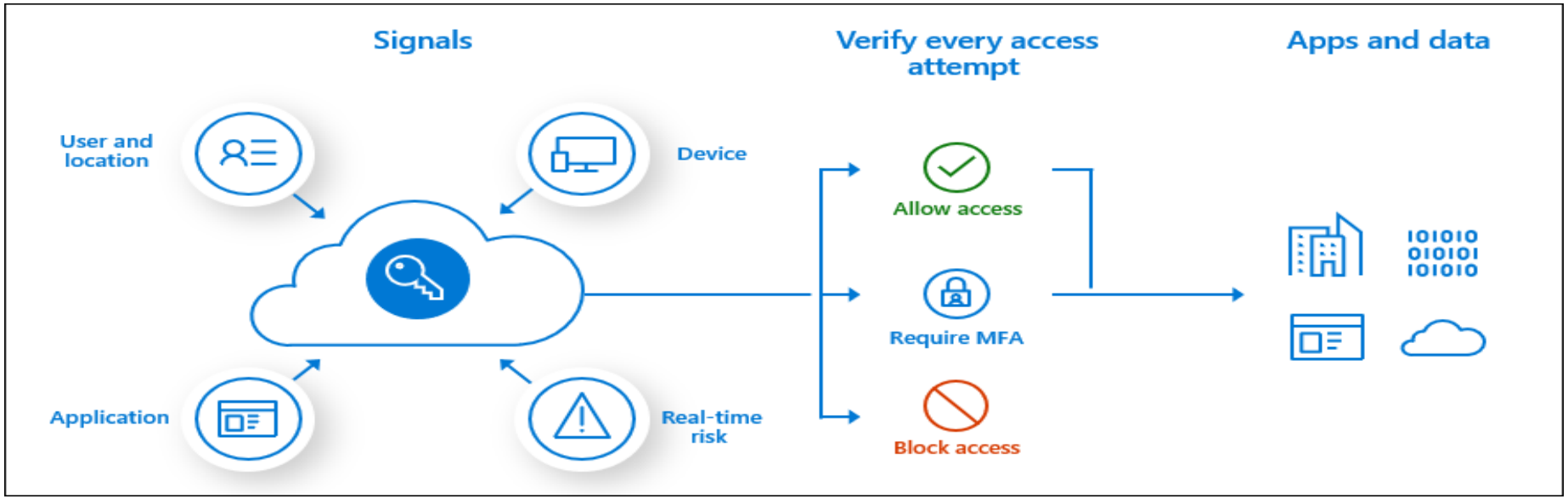
ADFS

- Cannot authenticate directly to MS Online portal
- Credentials stored on-premises
- Federated trust is between Azure and On-prem AD to authenticate access to the cloud
- Attacks require authenticating to on-prem ADFS portal rather than Azure endpoints
- Capture requests with Proxy

Azure AD Connect Health with ADFS

- Alerts about common ADFS issues
- cert expiring, missing updates, performance etc)
- bad Password Attempts Risky IPs
- ADFS 2016 2019 Smart Lockout

Conditional Access Policies



- MFA
- Security defaults
  - Requires MFA
  - Block legacy protocol
  - Protects privileged activities → Azure Portal
  - **Disabled for conditional access policies**
- Fine grained controls for access to resources
  - When/where MFA is applied
  - Device platforms use User Agent to apply policies

MFA Sweep (Policy Validation)

- MFASweep is a PowerShell script that attempts to log in to various Microsoft services using a provided set of credentials and will attempt to identify if MFA is enabled
  - Azure Service Management API
  - Microsoft Graph API
  - O365 Exchange Services
  - O365 Web Portal
  - O365 Active Sync
  - ADFS
  - https://github.com/dafthack/MFASweep
- It also has an additional check for ADFS configurations and can attempt to log in to the on-prem ADFS server if detected

Azure Portal

- Authenticated users
  - portal.azure.com
  - View Azure Active Directory
- Can get AD info with Powershell cmdlets or az cli
- Lock down with
  - Set-MsolCompanySetting –UserPermissionToReadOtherUsersEnabled $false

Powershell Modules

- AzureAD and MSOnline
- Allows full management and interaction with Azure AD and Office 365 Services
- Be careful with access tokens used by scripts and checked in to Github etc.

Azure AD User Attributes

- User attributes and sensitive information
- Credentials can be found
  - Description
  - Comment fields
- Module MSOnline
- Connect MsolService
- $users= Get-MsolUser; foreach($user in $users) { $props = @();

Other Attack Considerations and Tools

- Service Principal Hijacking
  - 200 default service principals in O365 tenant
- Key vaults
  - Stores passwords and other secrets
  - Azure Key Vault
  - Secrets Management - Azure Key Vault _> store and tightly control access to tokens, passwords, certificates, API keys, and other secrets

Tools

- ScoutSuite
  - Good info baseline
- ROADTools
- PowerZure
- MicroBurst
- AzureHound

Scoutsuite

https://github.com/nccgroup/ScoutSuite/wiki/Azure

- Persistent monitoring - so you know about changes or issues as they arise
- One tool - all configuration checks in one place for speed and simplicity
- Multi-vendor support - AWS, Azure and GCP public cloud accounts
- Agnostic platform - a trusted third-party tool

  - **Run Scout using --user-account-browser**
  - **Through a browser, pick your azure account**

RoadTools

https://github.com/dirkjanm/ROADtools

- ROADtools is a framework to interact with Azure AD. It currently consists of a library (roadlib) and the ROADrecon Azure AD exploration tool.
- ROADrecon is a tool for exploring information in Azure AD from both a Red Team and Blue Team perspective. In short, this is what it does:
  - Uses an automatically generated metadata model to create an SQLAlchemy backed database on disk.
  - Use asynchronous HTTP calls in Python to dump all available information in the Azure AD graph to this database.
  - Provide plugins to query this database and output it to a useful format.
  - Provide an extensive interface built in Angular that queries the offline database directly for its analysis

PowerZure

- https://powerzure.readthedocs.io/en/latest/Functions/operational.html
- https://github.com/hausec/PowerZure
  - Enumeration
  - Operational
- What if user has logged into Azure CLI before (not unusual for System Admins) and they have an accessToken in their .Azure file.
- The tester could then take that token and impersonate the user in Azure, where they now have Contributor access to several different VMs.
- Access all Key Vaults

MicroBurst

- MicroBurst: A PowerShell Toolkit for Attacking Azure

- Managed Identities on Azure VMs can be given excessive Azure permissions. Access to these VMs could lead to privilege escalation
- https://www.netspi.com/blog/technical/cloud-penetration-testing/azure-privilege-escalation-using-managed-identities/
- https://www.netspi.com/blog/technical/cloud-penetration-testing/azure-automation-accounts-key-stores/

AzureHound

- AzureHound uses the “Az” Azure PowerShell module and “Azure AD” PowerShell module for gathering data within Azure and Azure AD. Visualize Azure attack surface and pivots
- Can be used to self audit
- Once the Az module is installed, you can import AzureHound by using the command:

AzureHound Cont.

- It is also possible to steal the access tokens from a compromised machine if that machine has been used to login to Azure PowerShell before. Copy the existing files
  - C:\users\[Username]\.azure\AzureRmContextSettings.json
  - C:\users\[Username]\.azure\TokenCache.dat
- For stealing AzureAD tokens, the tokens are cached in one of the module's DLL files and requires the PowerShell process context in order to access the tokens. They can be stolen using the command:
  - $token = [Microsoft.Open.Azure.AD.CommonLibrary.AzureSession]::AccessTokens['AccessToken']
  - $token.AccessToken
- You can then decode this JWT token to gather the UserPrincipalName and TenantID by copy and pasting it into the JWT decoder.

Using AzureHound

- To use AzureHound, you can invoke it using the command “Invoke-AzureHound”
- By default, AzureHound will output the results to a file called “[timestamp]-azurecollection.zip” in current directory
- This can be changed using the “-OutputDirectory” switch, e.g. “Invoke-AzureHound -OutputDirectory “C:tempresults””
- -TenantId xxxx-xxxx-xxxx-xxxx | Gather using a specific tenant Id instead of using the current one

Protection Advice

Phishing Protections

- Monitor Azure AD Logs
- Pull Logs from Azure AD Graph API
- Azure Event Hub
- Splunk
- Syslog

Audit Consented Permissions for All Apps

- Got to Enterprise Apps...click on Permissions
  - List Admin consented permission and permission rating
  - Review consents that are of ConsentType 'AllPrincipals’
  - Discrete permissions that each delegated permission or application has
  - ClientDisplayName suspicious
  - Specific users that have consents granted...check if high
- Get-AzureADPSPermissions.ps1

Other Important Items

- Treat ADFS or Azure AD connect as a Tier 0 resource
  - MFA for Admins
- Conditional Access or Baseline Policy for Admins
  - https://aka.ms/aadbaseline
- Azure AD Privilege Identity Management PIM P2 license
  - https://aka.ms/deploymentplans
- FIDO2 Global Administrator
  - http://aka.mis/fido2docs
- Leaked Credential Reporting
- Flip authentication to Azure AD only (Internal breach)

Key Take Aways

- Audit your Azure AD Environment!
- Lots of good tools available
- We will be releasing a lab on Azure AD in the next few months