Angriffe auf das Azure Active Directory und auf Azure-Dienste

Ins Netz gegangen

Frank Ully

Beim Azure Active Directory sind mangelnde Härtung und Fehlkonfigurationen der Dreh- und Angelpunkt, wenn es um die Angreifbarkeit des AAD und darüber abgesicherter Dienste wie Microsoft 365 und die IaaS- und PaaS-Angebote in Azure geht. Wer die typischen Angriffe kennt, weiß, worauf er achten muss.

Microsofts Identitätsdienste sind auf überraschend vielfältige Weise angreifbar. Noch erstaunlicher: Oft sind die Angriffe auch sehr einfach. Das ist die Folge des komplexen Berechtigungssystems, der Exponiertheit von Cloud-Anwendungen und nicht zuletzt der Verknüpfung von On-Premises Active Directory und Azure AD. Dieser Artikel erklärt die wichtigsten Angriffe auf das Azure Active Directory.

Die ATT&CK-Matrix der gemeinnützigen Organisation MITRE Corporation bildet Techniken, Taktiken und Methoden von Angreifern ab und hilft so Verteidigern, das Vorgehen ihrer Widersacher zu verstehen und Gegenmaßnahmen umzusetzen [1]. Die bekannte Enterprise-Matrix kann auf der MITRE-Website passend für Azure AD und Office 365 gefiltert werden, ebenso für IaaS- und PaaS-Dienste. Diese Filterung schränkt die Ergebnisse jedoch zu sehr ein. Angriffe, bei denen Azure AD eine Rolle spielt, die aber auf die Cloud-Umgebungen im Allgemeinen oder Azure selbst zielen, werden dann oft übersehen.

Mehr ins Detail gehen die Azure AD und Microsoft 365 Attack Matrix von Lina Lau und die Azure Attack Matrix von David Okeyode (alle genannten Matrizes siehe ix.de/z9p5). Führt man diese beiden Matrizes, wie in Abbildung 1 gezeigt, zu einer zusammen, wird deutlich, wie verwundbar Microsofts Cloud-Dienste sind.

iX-TRACT

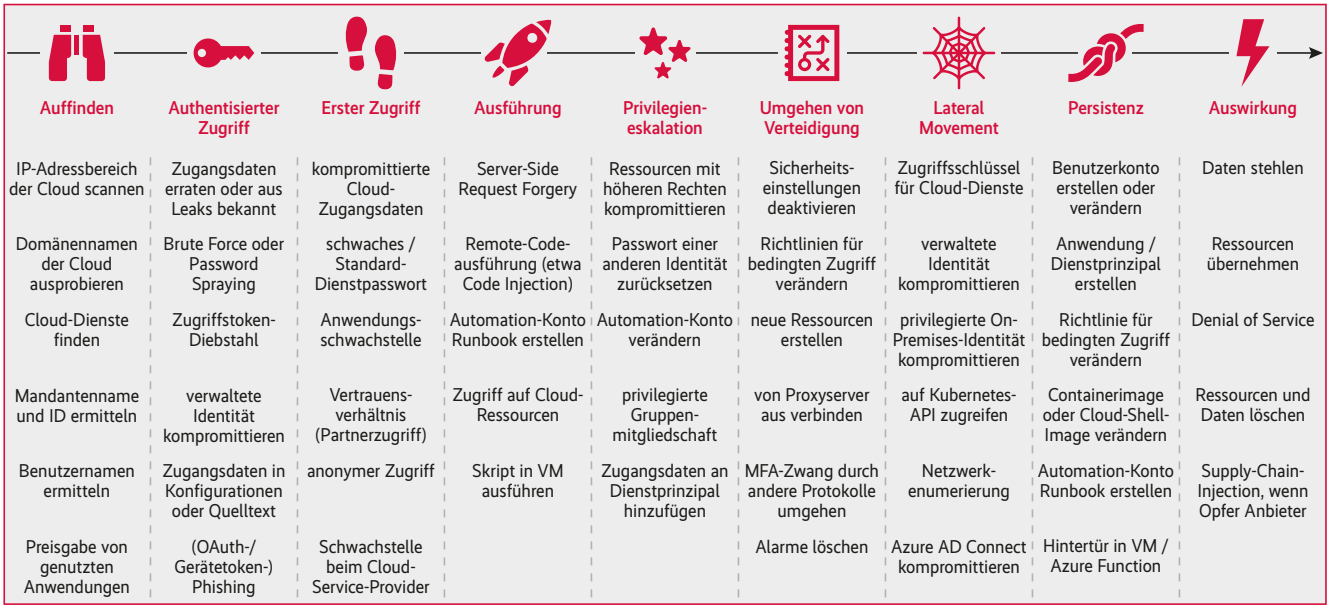
- Wie beim lokalen Active Directory ermöglichen es mangelnde Härtung und Fehlkonfigurationen, dass Angreifer einzelne Identitäten, Azure-Ressourcen oder gar das komplette Azure Active Directory und damit verbundene Abonnements kompromittieren.
- Dabei können sie vom initialen Zugriff ohne Zugangsdaten ihre Rechte unter Umständen immer weiter erhöhen und zwischen Azure-Diensten und dem AAD springen.
- Wenn eine Organisation hybrid arbeitet und eine Verbindung zwischen On-Premises Active Directory und AAD hergestellt hat, ist der Wechsel aus einer kompromittierten Cloud-Instanz zur lokalen Installation möglich – und andersherum.

Unauthentisierte Enumeration durch IP-Adressen und DNS

Informationen über die ins Visier genommene Umgebung zu sammeln, ist zu Beginn und während eines laufenden Angriffs entscheidend für Sicherheitstester sowie für Kriminelle.

Wenn eine Organisation Dienste auf lokalen Servern oder in einem Rechenzentrum aus dem Internet erreichbar macht, sind die öffentlichen IP-Adressen in ihrem Besitz oder angemietet. Im Gegensatz dazu verwendet die Azure-Cloud für alle Kunden einen großen gemeinsamen Adresspool. Zudem ändern sich die öffentlichen Adressen für Dienste wie virtuelle Maschinen bei jedem Neustart, sofern nicht anders konfiguriert.

Sind Kriminelle auf der Suche nach einem beliebigen Opfer, das Azure verwendet, können sie bei Microsoft eine mehrere Megabyte große JSON-Datei mit allen IPv4-Adressbereichen der öffentlichen Cloud herunterladen (siehe ix.de/z9p5). In diesem Adressbestand forschen sie mit einem Portscan [2] nach offenen



Auffinden	Authentisierter Zugriff	Erster Zugriff	Ausführung	Privilegien-eskalation	Umgehen von Verteidigung	Lateral Movement	Persistenz	Auswirkung
IP-Adressbereich der Cloud scannen	Zugangsdaten erraten oder aus Leaks bekannt	kompromittierte Cloud-Zugangsdaten	Server-Side Request Forgery	Ressourcen mit höheren Rechten kompromittieren	Sicherheits-einstellungen deaktivieren	Zugriffsschlüssel für Cloud-Dienste	Benutzerkonto erstellen oder verändern	Daten stehlen
Domänennamen der Cloud ausprobieren	Brute Force oder Password Spraying	schwaches / Standard-Dienstpasswort	Remote-Code-ausführung (etwa Code Injection)	Passwort einer anderen Identität zurücksetzen	Richtlinien für bedingten Zugriff verändern	verwaltete Identität kompromittieren	Anwendung / Dienstprinzipal erstellen	Ressourcen übernehmen
Cloud-Dienste finden	Zugriffstoken-Diebstahl	Anwendungs-schwachstelle	Automation-Konto Runbook erstellen	Automation-Konto verändern	neue Ressourcen erstellen	privilegierte On-Premises-Identität kompromittieren	Richtlinie für bedingten Zugriff verändern	Denial of Service
Mandantenname und ID ermitteln	verwaltete Identität kompromittieren	Vertrauens-verhältnis (Partnerzugriff)	Zugriff auf Cloud-Ressourcen	privilegierte Gruppen-mitgliedschaft	von Proxyserver aus verbinden	auf Kubernetes-API zugreifen	Containerimage oder Cloud-Shell-Image verändern	Ressourcen und Daten löschen
Benutzernamen ermitteln	Zugangsdaten in Konfigurationen oder Quelltext	anonymer Zugriff	Skript in VM ausführen	Zugangsdaten an Dienstprinzipal hinzufügen	MFA-Zwang durch andere Protokolle umgehen	Netzwerk-enumerierung	Automation-Konto Runbook erstellen	Supply-Chain-Injection, wenn Opfer Anbieter
Preisgabe von genutzten Anwendungen	(OAuth-/ Gerätetoken-) Phishing	Schwachstelle beim Cloud-Service-Provider			Alarme löschen	Azure AD Connect kompromittieren	Hintertür in VM / Azure Function	

Die zusammengeführten Matrizes von Lina Lau und David Okeyode lassen ahnen, wie angreifbar Microsofts Cloud-Identitätsdienste sind (Abb. 1).

Ports mit SSH oder dem Remote-Desktop-Protokoll (RDP) und versuchen es dort mit erratbaren Zugangsdaten. Ein lohnendes Ziel sind Webanwendungen, in denen sich womöglich klassische Schwachstellen wie SQL- oder Command-Injection finden – oder die etwas jüngere Bugklasse der Server-Side Request Forgery (SSRF) [3].

Zielgerichteter können Angreifer im Domain Name System (DNS) nach Subdomänen mit dem Namen der Organisation suchen, die sie angreifen wollen. Dadurch finden sie den Namen des Mandanten des Opfers heraus und welche SaaS-Dienste wie SharePoint Online oder IaaS-/PaaS-Dienste wie App Services oder Schlüsseltresore es verwendet. Diese Dienste nutzen typische Domänennamen wie nameder ressource.vault.azure.net bei Schlüsseltresoren. Allerdings: Nur weil jemand in Azure einen Namen verwendet, bedeutet das nicht unbedingt, dass die Subdomänen wirklich von der gleichnamigen Organisation stammen. Microsoft stellt eine unvollständige Referenzliste mit Azure-Domänen bereit (siehe ix.de/z9p5).

Bei der Suche nach Subdomänen in der Azure-Cloud hilft Sicherheitstestern die PowerShell-Werkzeugsammlung MicroBurst (siehe ix.de/z9p5), wie in Listing 1 gezeigt. Ihr Name leitet sich ab vom Fachbegriff für Fallwinde, die spektakulär aus Wolken herausschießen.

Aufspüren von Speicherkonten

Eine Suche nach Subdomänen findet Speicherkonten, in denen analog zu den bekannten S3-Buckets von AWS sowohl Dateien, als auch Tabellen liegen können [4]. Der Befehl `Invoke-EnumerateAzureBlobs` von MicroBurst spürt durch Kombination von Namensbestandteilen aus einer Wörterliste noch mehr Speicherkonten auf – er fände etwa 2consultit.blob.core.windows.net – und forscht in diesen Konten nach öffentlich einsehbaren Verzeichnislisten sowie nach erratbaren Verzeichnis- und Dateinamen.

cloud_enum ist ein ähnliches Multi-Cloud-Werkzeug, das über Subdomänensuche nicht nur in Azure Ressourcen entdeckt, sondern ebenfalls in AWS und GCP. Findet ein Angreifer ein fehlkonfiguriertes Speicherkonto, das öffentlich zugänglich ist, stößt er darin unter Umständen auf sensible Informationen, beispielsweise AAD-Zugangsdaten. Ebenfalls finden sich

Listing 1: Suche nach möglichen Mandanten einer Organisation nebst genutzten Diensten

```
PS > git clone https://github.com/NetSPI/MicroBurst.git
PS > Import-Module .\MicroBurst\MicroBurst.psm1
PS > Invoke-EnumerateAzureSubDomains -Base 2consult

Subdomain                                Service
---------                                -------
2consult.mail.protection.outlook.com     Email
2consult.onmicrosoft.com                 Microsoft Hosted Domain
2consulttest.onmicrosoft.com             Microsoft Hosted Domain
2consult-public.sharepoint.com           SharePoint
2consult-web.sharepoint.com              SharePoint
2consult.sharepoint.com                  SharePoint
2consult-my.sharepoint.com               SharePoint
2consult.vault.azure.net                 Key Vaults
2consult.queue.core.windows.net          Storage Accounts
2consult.blob.core.windows.net           Storage Accounts
2consult.file.core.windows.net           Storage Accounts
2consult.table.core.windows.net          Storage Accounts
2consult.scm.azurewebsites.net           App Services
2consult.azurewebsites.net               App Services
```

manchmal solche Anmeldeinformationen in Quellcode-Repositorys auf Codeaustauschplattformen wie GitHub.

Einfache Namensfindung

Hat ein Angreifer den Namen des Mandanten seines Opfers herausgefunden, ermittelt er einfach per Webbrowser die Mandanten-ID, indem er https://login.microsoftonline.com/2consult.onmicrosoft.com/v2.0/.well-known/openid-configuration ansurft. Analog findet er über den Endpunkt https://login.microsoftonline.com/getuserrealm.srf?login=benutzeristegal@2consult.onmicrosoft.com&xml=1 heraus, welche Firmenbezeichnung im AAD-Tenant eingetragen ist und ob Federation genutzt wird. Das funktioniert ebenso, wenn das Opfer eine eigene Domäne wie 2consult.ch im Mandanten registriert hat.

Mögliche Benutzernamen kann der Angreifer enumerieren, ohne dass er schon einen gültigen Zugang bräuchte oder dabei ein Alarm ausgelöst würde. Zunächst sammelt er in einer Textdatei die Namen von Mitarbeitern aus öffentlichen Quellen im Internet (Open Source Intelligence, OSINT) [5]. Diese Datei füttert er in ein Skript wie NameMash (siehe ix.de/z9p5), das für einen einzelnen Namen Varianten typischer Benutzernamen ausspuckt, für Peter Pan etwa panpeter, ppan, peter.pan oder pan. In Kombination mit der Mandantendomäne ergeben sich daraus vollständige AAD-Anmeldenamen wie peter.pan@2consult.onmicrosoft.com.

Diesen Kontonamen schickt der Angreifer an eine URL unter `login.microsoftonline.com`. Anhand der zurückkommenden Fehlermeldung lässt sich ermitteln, ob es ein passendes Benutzerkonto gibt oder nicht. Dabei helfen Enumerationswerkzeuge wie das Python-Skript o365creeper (siehe ix.de/z9p5), wie in Listing 2 gezeigt.

Listing 2: Herausfinden realer Benutzernamen für einen AAD-Mandanten

```
PS > git clone https://github.com/LMGsec/o365creeper.git
PS > python.exe .\o365creeper\o365creeper.py -f ↵
                moegliche-benutzerkonten.txt -o gueltige-benutzerkonten.txt

peterpan@2consult.onmicrosoft.com - INVALID
panpeter@2consult.onmicrosoft.com - INVALID
peter.pan@2consult.onmicrosoft.com - VALID
ppan@2consult.onmicrosoft.com – INVALID
[...]
```

Listing 3: Ausprobieren desselben Passworts für alle gefundenen AAD-Benutzer

```
PS > git clone https://github.com/dafthack/MSOLSpray.git
PS > Import-Module .\MSOLSpray\MSOLSpray.ps1
PS > Invoke-MSOLSpray -UserList .\gueltige-benutzerkonten.txt ↵
                                                    -Password Sommer2022!
[*] There are 25 total users to spray.
[*] Now spraying Microsoft Online.
[*] SUCCESS! peter.pan@2consult.onmicrosoft.com : Sommer2022!
```

Passwörter sprühen und Zugangsdaten reinstopfen

Eine gängige Methode von Angreifern ist das sogenannte Password Spraying. Dabei wird ein wahrscheinlich verwendetes Passwort wie „Winter2021“ oder „Unternehmensname2022!“ mit bekannten Konten durchprobiert. Da die Anmeldeversuche mit einem bestimmten Passwort verteilt auf alle Benutzer stattfinden, wird kein Konto gesperrt, im Gegensatz zu Dutzenden oder Hunderten fehlerhaften Versuchen bei einem einzelnen Benutzer. Ähnlich zielgerichtet funktioniert Credential Stuffing, bei dem Angreifer durch Leaks bekannt gewordene Kombinationen aus Benutzernamen und Passwort ausprobieren.

Wenn Konten oder Passwörter für Cloud und on Premises zwar nicht synchronisiert werden, ein Benutzer in beiden Umgebungen jedoch dasselbe Kennwort verwendet, erhält ein Angreifer mit einem erfolgreich ausgespähten Azure-Kennwort womöglich Zugang zum eigentlich getrennten Unternehmensnetzwerk – oder andersherum.

Ein Werkzeug zum Sprühen von Passwörtern ist MSOLSpray von Beau Bullock, gezeigt in Listing 3. In Kombination mit anderen Tools wie FireProx können Angreifer bei Authentifizierungsanfragen zudem ihre Quell-IP-Adressen rotieren, um die Smart-Lockout-Funktion von Azure AD zu umgehen (beide Tools siehe ix.de/z9p5).

Ein zweiter Faktor könnte den Angreifer davon abhalten, mit den gefundenen Zugangsdaten Schaden anzurichten. MFA-Sweep (siehe ix.de/z9p5), ebenfalls von Bullock, findet jedoch mögliche Lücken in der Mehr-Faktoren-Abdeckung: Es probiert Anmeldungen an sechs APIs wie Microsoft Graph oder Azure Management sowie an Legacy-Schnittstellen wie Microsoft 365 Exchange Web Services oder Active Sync. Wurde eine dieser Schnittstellen bei der Konfiguration des zweiten Faktors vergessen, kann sich der Angreifer dort anmelden.

OAuth- und anderes Phishing

Beim sogenannten OAuth-Phishing überträgt ein AAD-Benutzer über einen Klick auf einen per E-Mail erhaltenen Link

Berechtigungen für das Zurücksetzen der Kennwörter

Kennwort für ↓ kann zurückgesetzt werden von →	Kennwort-administrator	Helpdesk-administrator	Authentifizierungs-administrator	Benutzer-administrator	privilegierter Authentifizierungs-administrator	globaler Administrator
Authentifizierungsadministrator			✓		✓	✓
Rolle „Verzeichnis lesen“	✓	✓	✓	✓	✓	✓
globaler Administrator					✓	✓
Gruppenadministrator				✓	✓	✓
Gasteinladender	✓	✓	✓	✓	✓	✓
Helpdeskadministrator		✓		✓	✓	✓
Nachrichtencenter-Leser		✓	✓	✓	✓	✓
Kennwortadministrator	✓	✓	✓	✓	✓	✓
privilegierter Authentifizierungs-administrator					✓	✓
Administrator für privilegierte Rollen					✓	✓
Meldet Reader		✓	✓	✓	✓	✓
Benutzer (keine Administratorrolle)	✓	✓	✓	✓	✓	✓
Benutzer (keine Administratorrolle, aber Mitglied einer Gruppe, der Rollen zugewiesen werden können)					✓	✓
Benutzeradministrator				✓	✓	✓
Leseberechtigter für Berichte mit Nutzungszusammenfassung		✓	✓	✓	✓	✓

Rechte an eine vom Angreifer registrierte bösartige Anwendung beziehungsweise ihren Dienstprinzipal. Der Angriff wird auch als unrechtmäßige Erlaubniserteilung (Illicit Consent Grant) bezeichnet. Zur Attacke stehen der 365-Stealer und das Office 365 Attack Toolkit bereit (beide siehe ix.de/z9p5). Ohne Bestätigung eines Admins erhält der Angreifer jedoch nur niedrige Berechtigungen und Lesezugriff auf den betroffenen Mandanten.

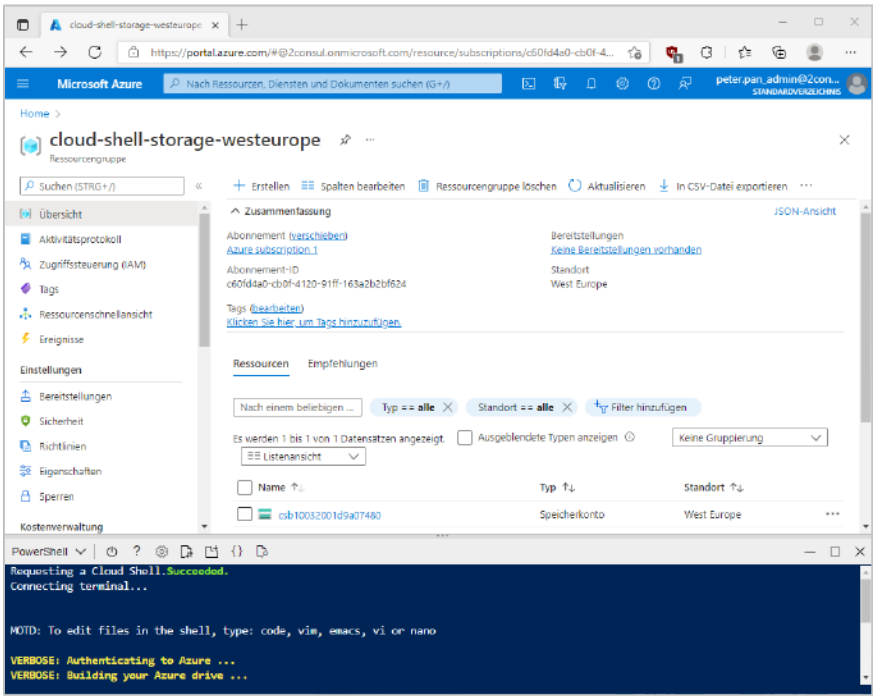
Eine neuere Technik ist das Fischen nach Gerätecodes, die eigentlich dazu gedacht sind, sich als Benutzer an Geräten wie Druckern oder smarten Fernsehern zu authentisieren. Ein Blogartikel von Bobby Cooke schildert ausführlich, wie Angreifer damit Zugriffs- und Aktualisierungstoken erbeuten (siehe ix.de/z9p5).

Enumeration als authentisierter Benutzer

Im klassischen AD kann ein Benutzer mit Netzwerkzugang zu einem Domänencontroller und gültigen Zugangsdaten viele hilfreiche Daten abrufen, darunter Benutzer- und Computerkonten sowie Gruppenmitgliedschaften [2]. In der Cloud können sich Benutzer mit Zugriff auf Microsoft-365-Dienste im Azure-Portal anmelden und aus dem Azure-AD-Mandanten Benutzerkonten, Geräte und Gruppenmitgliedschaften auflisten. Das ist vom gesamten Internet aus möglich.

Falls beispielsweise im Kommentarfeld zu einem Benutzer ein Hinweis auf sein Passwort steht, kann ein authentisierter Benutzer das analog zum On-Premises AD sehen.

Hat der Angreifer initial eine Identität kompromittiert, kann er sich je nach Art des Kontos am Azure-Portal anmelden, eines der Befehlszeilenwerkzeuge verwenden oder direkt auf die REST-APIs zugreifen. Zahlreiche Angriffswerkzeuge stehen zur Verfügung. MicroBurst etwa bringt



Dateien für die Cloud-Shell werden in einem Speicherkonto abgelegt, das ein Angreifer mit Mitwirkenden-Rechten kompromittieren könnte (Abb. 2).

eine Funktion mit, die viele Inhalte aus dem AAD wie Benutzer, Dienstprinzipale und Gruppen und womöglich sichtbare ARM-Informationen über RBAC-Rollen und Azure-Ressourcen in Textdateien ausspielt:

```
Connect-AzAccount
Get-AzDomainInfo -Folder ↵
        mandantenenumeration -Verbose
```

Das Enumerationswerkzeug ROADrecon aus der Python-basierten ROADtools-Sammlung von Dirk-jan Mollema (siehe ix.de/z9p5) liest ebenfalls viele Informationen aus dem AAD aus und stellt sie sogar dynamisch in einem Webbrowser dar:

```
pip install roadrecon
roadrecon auth -u peter.↵
       pan@2consult.onmicrosoft.com ↵
                        -p Sommer2022!
roadrecon gather
roadrecon gui
```

Selbst der MFA-Status einzelner Benutzer kann abgefragt werden, weil sie Mitglied der Rolle „Active Authentication Administrator“ sind (das Entfernen der Rolle beseitigt nicht den Zwang zum zweiten Faktor). ROADrecon enumeriert über den Befehl `roadrecon plugin policies` mit Standardzugangsdaten über Microsoft-interne APIs sogar Richtlinien für bedingten Zugriff. Im Azure-Portal werden nicht alle verfügbaren Informationen angezeigt, zum Beispiel nicht alle privilegierten AAD-Rollen (siehe ix.de/z9p5). Es gibt auch grafische Enumerationswerkzeuge wie Stormspotter (siehe ix.de/z9p5), das Angriffspfade ähnlich wie BloodHound [6] im On-Prem AD aufzeigt. BloodHound selbst kann ebenfalls in Azure nach Angriffswegen suchen, im Vergleich zu seinen Fähigkeiten in einem lokalen Verzeichnis allerdings nur rudimentär.

Nicht nur der allmächtige globale Administrator ist eine hoch privilegierte Rolle, die volle Kontrolle über das Azure-Verzeichnis hat. AAD-Rollen mit interessanten Rechten sind vielfältig: Der Administrator für privilegierte Rollen kann die globale Adminrolle zuweisen – und damit selbst volle Kontrolle übernehmen. Ein Gruppenadministrator kann Gruppen für AAD, aber auch für Teams oder SharePoint verwalten. Beim lokalen Administrator für in Azure AD eingebundene Geräte spricht der Name für sich, ähnlich hoch berechtigt auf verwalteten Geräten ist der Intune-Administrator.

Für einen Angreifer ebenso interessant sind AAD-Rollen, die bei Sicherheitsprinzipalen anderer Rollen das Passwort zurücksetzen dürfen: Zum Beispiel kann der privilegierte Authentifizierungsadministrator das Passwort globaler Admins ändern. Eine vollständige Dokumentation der Berechtigungen zum Kennwortzurücksetzen findet sich in der Microsoft-Doku über eingebaute AAD-Rollen (ix.de/z9p5), einen Auszug zeigt die Tabelle „Berechtigungen für das Zurücksetzen der Kennwörter“.

Unerwünschte Gäste

Manch ein Sicherheitsverantwortlicher mag denken, seine Firma verwalte in ihrem Azure AD ja nur eigene Mitarbeiter. Das kann stimmen, muss es aber nicht: In der Standardeinstellung darf jeder Benutzer Gäste einladen – und ein Gast weitere Gäste. Das Ergebnis kann eine von unerwünschten Besuchern überlaufene Party im eigenen Mandanten sein.

Gastbenutzer können sich mit den Anmeldedaten des eigenen Kontos im Azure AD anmelden und Informationen über reguläre Benutzer abfragen, wenn sie deren Anmeldenamen (User Principal Name, UPN) kennen – in der Regel die E-Mail-Adresse. Sie können auch Informationen über eine Gruppe auslesen, wenn sie deren ID kennen.

Beide Ansätze lassen sich mit der Werkzeugsammlung DCToolbox von Daniel Chronlund (siehe ix.de/z9p5) miteinander verknüpfen, wenn mindestens ein Benutzer bekannt ist. Das Skript `Get-DCAzureADUsersAndGroupsAsGuest` ermittelt alle Gruppen, in denen die Zielbenutzer Mitglied sind, dann alle anderen Mitglieder dieser Gruppen und wiederholt das bis zu fünfmal.

Gäste lesen nicht nur Informationen aus, sondern erhöhen unter Umständen ihre Privilegien, wenn Cloud-Admins dynamische Gruppen unsicher einsetzen. In diesen Gruppen werden Mitglieder nicht fest zugeordnet, sondern über Attribute wie die Niederlassung automatisch eingruppiert.

Angenommen, die Organisation hat die Verwaltung von Cloud-VMs an einen Dienstleister ausgelagert und dafür eine dynamische Gruppe „VM-Mitwirkende“ erstellt, ihr die Azure-Rolle „Mitwirkender von virtuellen Computern“ für das ge-

Listing 4: Abrufen von Zugriffsdaten einer verwalteten Identität

```
curl -H Metadata:true 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=↵
                                                                    https%3A%2F%2Fmanagement.azure.com%2F'
curl -H Metadata:true 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=↵
                                                        2018-02-01&resource=https%3A%2F%2Fgraph.microsoft.com%2F'
```

Listing 5: Zugriff auf REST-APIs und Anmeldung an Azure mit Zugriffstoken für eine verwaltete Identität

```
PS > $token = 'eyJ0eX[...]'; $graph = 'eyJ0eX[...]'
PS > $RequestParams = @{
        Method = 'GET'
        Uri = 'https://management.azure.com/subscriptions?api-version=2020-01-01'
        Headers = @{
                'Authorization' = "Bearer $token"
        }
}
PS > (Invoke-RestMethod @RequestParams).value
PS > Connect-AzAccount -AccountId 164aaf57-30af-41f0-840a-0e21ed149947 -AccessToken $token -GraphAccessToken $graph
```

samte Abo zugewiesen und macht nun Benutzer mit „vm-klempner“, also dem Domänennamen des Dienstleisters, im UPN zu Mitgliedern.

Ein Angreifer, der initialen Zugriff hat und diese Regel kennt – dazu genügen die AAD-Rechte des globalen Lesers –, kann diese Attribute normalerweise nicht an seinem eigenen Konto bearbeiten. Aber er kann sich selbst als Gast einladen mit einer E-Mail-Adresse, bei der irgendwo die Zeichenkette vorkommt, etwa kein-vm-klempner@angreifer.cn. Mit diesem neuen Konto ist er Mitglied in der VM-Mitwirkenden-Gruppe und kann PowerShell-Schadcode mit SYSTEM-Rechten auf allen virtuellen Maschinen ausführen.

Azure-Dienste kompromittieren

Oft haben Entwickler die weitreichende Mitwirkenden-Rolle. Mit dieser RBAC-Rolle in einem Abo, einer Ressourcengruppe oder einzelnen Ressourcen kann ein Angreifer in VMs nicht nur Malware ausführen, sondern auch ihre Festplatteninhalte herunterladen. Bei Schlüsseltresoren kann er sich Zugriff auf alle darin verwahrten Geheimnisse verschaffen und in Speicherkonten alle Container und Dateien lesen – und schreiben. Besonders kritisch wird das bei Speicherkonten für die Cloud-Shell, die mit „cs“ beginnen (siehe Abbildung 2). Ein Mitwirkender auf dem Cloud-Shell-Speicherkonto kann das mehrere Gigabyte große Dateisystem herunterladen, in einem Linux-System einbinden, Hintertüren für die Bash- und die PowerShell-Variante einbauen und die Cloud-Shell-Datei wieder hochladen. Nutzt nun ein globaler Admin diese Browser-Shell im Azure-Portal, ist sein Konto kompromittiert.

Aber nicht nur Mitwirkenden-Rechte sind gefährlich. Benutzer mit Leserechten können in der Historie der Bereitstellungsvorlagen für den Azure Resource Manager nach Ausgaben suchen, die als einfache Zeichenkette ausgespielt werden und nicht als SecureString geschützt sind, oder nach Zugangsdaten in den Vorlagen selbst. Ein Leser kann die Inhalte von Automation-Runbooks und den Quelltext und Dateien von Function Apps anzeigen oder Containerimages aus der Azure Container Registry (ACR) herunterladen, einer privaten Docker-Registry. Finden sich dort Zugangsdaten für eine höher privilegierte Identität oder interessanter Quelltext, ist der Angreifer einen Schritt weiter.

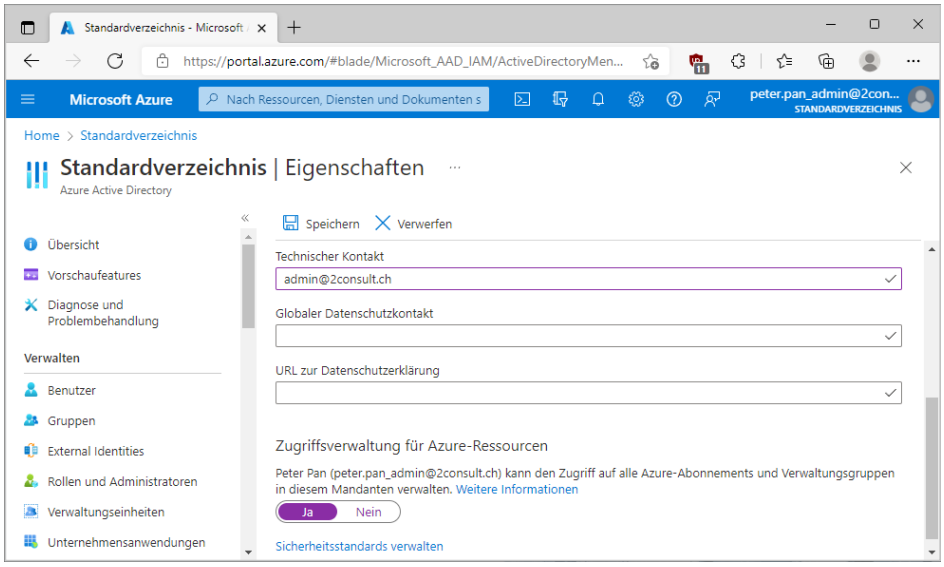
Das Toolkit MicroBurst hilft einem Angreifer, schnell die meisten derart verfügbaren Passwörter und Zertifikate auszulesen:

```
Get-AzPasswords -Verbose ↵
   -ModifyPolicies Y -ExportCerts Y ↵
                        | Out-GridView
```

Ein Angreifer kann selbst ohne vorherigen Zugriff Schwachstellen ausnutzen. Kann er etwa in einer in die Cloud gehobenen Webanwendung, die in einer virtuellen Maschine läuft, durch einen Bug Code ausführen, hat er Glück, wenn die virtuelle Maschine mit einer verwalteten Identität in Azure AD verknüpft ist. Dann kann er aus der VM heraus über einen privaten Metadatenendpunkt unter 169.254.169.254 Zugriffstoken für ARM und den Microsoft Graph abrufen, wie in Listing 4 dargestellt.

Auf ähnliche Weise kann ein Hacker, ausgehend von Umgebungsvariablen, in einer verwundbaren serverlosen App Zugriffstoken beispielsweise für Schlüsseltresore abfragen.

Auf seinem eigenen Rechner nutzt der Angreifer nun die erbeuteten Token, um im Namen und mit den Rechten der verwalteten Identität Daten von REST-APIs abzufragen oder dorthin Befehle zu schicken, wie in Listing 5 gezeigt. Er kann sich



Rechteerhöhung vom globalen Admin zum Benutzerzugriffsadministrator: Diese AAD-Rolle kann Azure-Abos übernehmen (Abb. 3).

ebenfalls direkt mit einem der Standard-PowerShell-Module bei Azure anmelden.

Falls die verwaltete Identität über höhere Rechte in AAD oder in Azure verfügt – eine AAD-Adminrolle hat oder eine Mitwirkenden-Rolle in einem Azure-Abo, wie das oft der Fall ist –, kann er davon ausgehend die Cloud-Umgebung seines Opfers weiter angreifen.

Der Wechsel von einem übernommenen Azure-Abo zum anderen ist in der Praxis einfach, weil oft dieselben Benutzer Besitzer mehrerer Abos sind.

Vom Azure Active Directory zu Azure-Diensten

Ein globaler Administrator kann alle Einstellungen im Azure-AD-Mandanten lesen und ändern. Falls ein Angreifer ein solches Konto kompromittiert, hat das Opfer nicht nur die Kontrolle über seinen Tenant, sondern über alle damit verbundenen Azure-Abos verloren. Das liegt trotz der Rollentrennung zwischen AAD und ARM daran, dass sich auf der Verzeichnisebene angewendete Rollen auf Azure-Ressourcen auswirken, die mit dem jeweiligen Tenant verbunden sind: nämlich wenn mit den AAD-Adminrechten ein Benutzer zum Mitglied einer vorhandenen Gruppe gemacht wird, die Mitwirkender eines Azure-Abonnements ist. Gruppenmitgliedschaften können zudem nicht nur von globalen Admins geändert werden; die Rollen Benutzer-, Gruppenadministrator und Verzeichnisschreibberechtigte haben ebenfalls dieses Recht.

Darüber hinaus werden Azure-Abos oft mit dem Konto eines globalen Admins abgeschlossen. Dadurch ist dieses Konto von Anfang an Eigentümer des Abos und hat darin volle Rechte. Durch die beschriebene Fähigkeit mancher hoch privilegierter Rollen, Passwörter zurückzusetzen, ist ebenfalls eine Bewegung von AAD zu Azure möglich, wenn das Passwort für einen Benutzer geändert wird, dem ein Azure-Abo gehört.

Allerdings kann sich ein globaler Administrator direkt aller Azure-Abos und der darin verwalteten Ressourcen bemächtigen, selbst wenn er keine Rechte darüber besitzt: Über die Funktion „Zugriffsverwaltung für Azure-Ressourcen“ in den AAD-Eigenschaften im Portal kann er seine Rechte auf die eines Benutzerzugriffsadministrators erweitern (siehe Abbildung 3). Diese Rolle ermöglicht das Verwalten von Azure-Ressourcen. Ein Angreifer kann sich damit zum Eigentümer aller Azure-Abos machen und diese Privilegieneskalation über das Azure-Kommandozeilenwerkzeug oder das Portal durchführen.

Unternehmen ist es in der Regel nicht bewusst, dass ein globaler Microsoft-365-Administrator die Möglichkeit hat, Azure-Rollenmitgliedschaften zu kontrollieren. Selbst wenn ein solcher Benutzer aus den globalen Admins entfernt wird, bleibt er weiter Benutzerzugriffsadmin, bis die Einstellung auf ursprünglichem Weg rückgängig gemacht wird.

Aus der Cloud ins Lokale

Falls in einem der Azure-Abos, über das sich ein Angreifer mit Adminrechten Kontrolle verschaffen konnte, Domänencontroller für ein On-Premises AD laufen, kann der Eindringling auf dem virtualisierten Windows-Server mit SYSTEM-Rechten Schadcode oder Hackertools ausführen: etwa Mimikatz, das die Passworthashes aller On-Premises-Konten vom Controller ausliest [6]. So schafft ein Eindringling auf direktem Weg den Sprung von der Cloud in eine „lokale“ Domäne.

Genauso wie in der alten On-Premises-Welt Benutzerrechner und Server Teil einer Domäne sind, kann ein Client mit einem AAD-Mandanten verbunden werden. In einer klassischen Domäne ist ein Benutzerrechner über Gruppenrichtlinien [7] verwaltbar – im AAD über den Microsoft Endpoint Manager und Intune. Intune ist eine AAD-Ressource, der Zugriff darauf wird mit Rollen in Azure AD gesteuert.

Ein Client kann entweder mit einer On-Premises-Domäne oder einem Azure-AD-Mandanten verbunden sein, aber auch mit beiden: als in Azure AD eingebundenes Hybridgerät. (Daneben gibt es in Azure AD registrierte Geräte, die in Bring-Your-Own-Device-Szenarien agieren und denen man von vornherein weniger vertraut als Geräten, die vom Unternehmen verwaltet werden.)

Mit dem Befehl `dsregcmd /status` kann man sich auf seinem Gerät anzeigen lassen, ob es in eine klassische Domäne oder in Azure AD eingebunden ist – oder in beide und damit hybrid. In AAD lässt

Listing 6: Herausfinden, ob Geräte von Azure aus verwaltet werden

```
Get-AzureADDevice -All $true | where {$_.iscompliant -eq $true} | select DisplayName,DeviceOSType,↵
                                                                    DeviceTrustType,IsManaged,IsCompliant

DisplayName  DeviceOSType  DeviceTrustType  IsManaged  IsCompliant
-----------  ------------  ---------------  ---------  -----------
PC-ADMIN01   Windows       ServerAd             True         True
PC-ADMIN02   Windows       ServerAd             True         True
```

Listing 7: Auslesen von Passwörtern von einem kompromittierten Azure-AD-Connect-Server und DCSync-Angriff auf die lokale Domäne

```
PS > Install-Module AADInternals
PS > Get-AADIntSyncCredentials

ADDomain : AD.2CONSULT.CH
ADUser : MSOL_882bef6ee0a1
ADUserPassword : vl]&O\MXK)@C#B"&Zgqdda[Sfl[...]
AADUser : Sync_AADCONNECT_882bef6ee0a1@2consult.onmicrosoft.com
AADUserPassword : @,CeH|o-nMfEyCpg

PS > runas /noprofile /user:ad.2consult.ch\MSOL_882bef6ee0a1 powershell
PS > Invoke-Mimikatz -Command '"lsadump::dcsync /domain:ad.2consult.ch /all /csv"'
```

sich über die Azure-AD-PowerShell herausfinden, ob ein Gerät von Azure AD verwaltet wird, wie in Listing 6 gezeigt. Dazu genügen einfache Benutzerrechte. Die Eigenschaft `DeviceTrustType` mit dem Wert `ServerAd` bedeutet, dass das Gerät in ein On-Premises AD und ein AAD eingebunden ist.

Wenn nun ein Domänenadmin ein Hybridgerät verwendet, führt ein Angreifer mit Kontrolle über einen globalen oder Intune-Administrator-Account beliebige PowerShell-Skripte mit Schadcode auf diesem Gerät aus, das ebenso eine Verbindung mit einem lokalen Verzeichnis hat. Dies geht komfortabel über das Azure-Portal oder mit den Standard-PowerShell-Modulen. Dadurch hat ein Eindringling den Sprung von der Cloud in eine lokale Domäne geschafft und einen Domänenadmin kompromittiert.

Selbst wenn keine Benutzersynchronisierung stattfindet, wie sie gleich beschrieben wird, hat ein Angreifer so die Grenzen zwischen den verschiedenen Identitätsverwaltungssystemen On-Premises AD und AAD überschritten. Besonders kritisch ist, wenn Geräte aus verschiedenen lokalen Domänen mit demselben AAD-Mandanten hybrid verbunden werden. Dann sind Angriffspfade von einer On-Premises-Gesamtstruktur über die Cloud in eine andere On-Premises-Gesamtstruktur möglich, selbst wenn keine AD-Vertrauensstellungen bestehen (siehe ix.de/z9p5).

Hybrididentitäten und On-Premises-Kompromittierung

Eine einzige Benutzeridentität für die Authentifizierung für alle Unternehmensressourcen, unabhängig vom Standort, hat einigen Charme. Falls AD-Benutzer dieselben Anmeldeinformationen in der lokalen Domäne und in der Cloud verwenden sollen, gibt es drei Möglichkeiten: Password-Hash-Synchronisierung (PHS), Pass-Through-Authentifizierung (PTA) und Active Directory Federation Services (ADFS). Ein Vergleich der Techniken findet sich in [8].

Ein Angreifer kann alle drei Arten von Hybrididentität kompromittieren. Besonders häufig nutzen Organisationen PHS, bei der Passwörter aus dem lokalen AD mit AAD synchronisiert werden. Genauer gesagt überträgt ein im AD installierter Azure-AD-Connect-Server die abermals gehashten Passworthashes [6] der synchronisierten Domänenbenutzerkonten in die Cloud. Damit das funktioniert, erstellt AAD Connect bei der Installation zwei wesentliche Konten: Im lokalen AD legt es das ADDS-Connector-Konto an, dessen Name mit `MSOL_` beginnt, und versieht es mit DCSync-Rechten [6], damit es On-Premises-Daten wie die Passworthashes lesen und schreiben kann. In Azure AD wird ein Connector-Konto erstellt, dessen

Name mit Sync_ beginnt. Es erhält eine besondere Rolle für Verzeichnissynchronisierungsaufgaben, mit der es Daten wie Benutzerattribute, darunter die Passworthashes, und Gruppenmitgliedschaften in AAD schreiben darf.

Hat sich ein Angreifer in der lokalen Domäne breitgemacht und gelingt es ihm, dort seinen Schadcode auszuführen, etwa über eine der zahlreichen Möglichkeiten zur klassischen Privilegieneskalation wie bearbeitbare Gruppenrichtlinien [7], die auf den Azure-AD-Connect-Server angewendet werden, dann kann er die Passwörter für die beiden Connector-Konten auslesen (siehe Listing 7). Eine weitere Möglichkeit hat er, wenn er einen AD-Benutzer kompromittiert hat, der Kontenoperator ist: Der Operator kann das Kennwort für das oben erwähnte MSOL-Konto zurücksetzen. Mit diesem On-Premises-Konto und dessen DCSync-Rechten kann der Angreifer die Passwörter aller Benutzer im lokalen AD auslesen.

Vom Lokalen in die Cloud

Mit dem ebenfalls erbeuteten Passwort für das Sync-Konto in Azure AD kann sich der Angreifer beim Cloud-Identitätsdienst anmelden und dort die Zugangsdaten eines AAD-Benutzers ändern – wie in Listing 8 gezeigt – und sich anschließend als dieser Benutzer anmelden.

Bei Hybrididentitäten mit PTA sieht ein Angreifer, der den AAD-Connect-Server unter seine Kontrolle gebracht hat, Klartextpasswörter von Benutzern, sobald sie sich bei einem Azure-Dienst anmelden. Bei ADFS führt die Kompromittierung eines Verbundservers dazu, dass Angreifer sich nicht nur im lokalen AD als beliebiges Konto ausgeben können, sondern ebenso gegenüber einem damit synchronisierten AAD-Mandanten; damit sind ebenfalls beide Umgebungen gefallen.

Um die Arbeit mit einer hybriden Identität bequemer zu machen, können Admins darüber hinaus bei PHS und PTA Seamless Single Sign-on (SSO) einrichten, nahtloses einmaliges Anmelden mit Azure AD. Dabei wird im lokalen Verzeichnis das Computerkonto AZUREADSSOACC$ erstellt. Ein Angreifer, der das On-Premises AD kompromittiert hat, kann mit einer Technik zum Auslesen von Passworthashes dessen Hash auslesen, ein silbernes Ticket [9] für einen bestimmten SSO-Cloud-Endpunkt von Microsoft erstellen und sich damit gegenüber AAD als beliebiger synchronisierter Benutzer ausgeben.

Faule Angreifer müssen sich gar nicht die Mühe machen, ein lokales Verzeichnis weitgehend zu übernehmen, um davon ausgehend als Cloud-Admin die Azure-Umgebung des Opfers zu kapern. Es genügt, wenn sie zum Beispiel über eine Phishingkampagne zielgerichtet Malware auf dem Arbeitsrechner eines Admins ausführen, der darüber Microsoft-365- oder Azure-Dienste verwaltet und über eine der genannten Adminrollen verfügt. Nutzt der Systemverwalter zur Administration das Azure-Portal im Edge-Browser, klaut der Eindringling mit einem Werkzeug wie Chlonium (siehe ix.de/z9p5) dessen Sitzungscookies für das Portal und importiert sie in seinen eigenen Browser. Solange die Cookies gültig sind, kann der Angreifer nun selbst im Azure-Portal administrieren.

Ist der Admin auf der Kommandozeile zu Hause und nutzt die Azure-Befehlszeilenschnittstelle oder eines der PowerShell-Module, kopiert sich der Eindringling aus dem Verzeichnis %USERPROFILE%/.azure dessen Konfigurationsdateien einschließlich Zugriffstoken. Bei beiden Techniken braucht der Angreifer nicht einmal lokaler Administrator auf dem Rechner des Cloud-Admins zu werden und, falls eingesetzt, keinen zweiten Faktor auszuspähen.

Bedingter Zugriff

Bedingter Zugriff (Conditional Access) ist eine Premiumfunktion und in der Voreinstellung nicht aktiv. Die Richtlinien werden beim Zugriff auf Ressourcen ausgewertet und bilden in ihrer einfachsten Form Wenn-dann-Anweisungen: Wenn ein Benutzer auf eine Ressource zugreifen möchte, muss er eine Aktion ausführen oder eine Bedingung erfüllen. Damit können Admins Richtlinien erstellen, die für Benutzer einen zweiten Faktor erfordern oder die Anmeldung aus bestimmten Ländern – erkannt anhand der IP-Adresse – verbieten. Es ist alles erlaubt, was nicht ausdrücklich verboten ist.

Eine häufig anzutreffende Fehlkonfiguration des bedingten Zugriffs ist, dass in Richtlinien für Mehr-Faktor-Authentisierung (MFA) neben wenigen Notfallkonten, die davon ausgenommen sein sollten, Ausnahmen für normale Benutzer eingefügt werden. Dadurch kann ein Angreifer, der ein Passwort eines solchen Kontos erbeutet hat, sich ohne zweiten Faktor anmelden. Harmlos erscheinende MFA-Ausschlüsse für alle Identitäten, die sich von vertrauenswürdigen IP-Adressbereichen anmelden, können unvorhergesehene Schwachstellen aufreißen, wenn darunter das Gäste-WLAN ist.

Bei Richtlinien, die sich auf die Geräteplattform beziehen, gibt es einerseits Fehler im Aufbau. Aktiviert der Admin einzeln alle derzeit verfügbaren Plattformen – Windows, macOS, iOS und Android – und geht davon aus, dass die Richtlinie nun für alle Anmeldungen gilt, loggt sich ein Angreifer via Linux-Rechner ein. Andererseits kann ein Angreifer die Geräteplattform einfach fälschen, da sie aus dem User-Agent-String abgeleitet wird, den Browser bei jeder HTTP-Anfrage übertragen. Falls eine Richtlinie die Anmeldung eigentlich auf Mobilgeräte beschränkt, aktiviert ein Angreifer in den Entwicklerwerkzeugen seines Browsers im Handumdrehen den User-Agent-String eines iPads.

Ist die AAD-Standardeinstellung unverändert, kann ein Angreifer sogar Richtlinien umgehen, die ein konformes Gerät erfordern, ohne dass er unmittelbar ein solches etwa durch Malware-Phishing kompromittiert hat: Er meldet einen eigenen Rechner an und macht ihn konform – oder er fälscht dessen Rückmeldungen an den Endpoint Manager über die angebliche Konformität (siehe ix.de/z9p5).

Persistenz: sich dauerhaft und unbemerkt einnisten

Ein nicht nur für Sicherheitstester, sondern ebenso für Admins erhellender Blogartikel zum bedingten Zugriff ist „The Attackers Guide to Azure AD Conditional Access“ von Daniel Chronlund (siehe ix.de/z9p5).

Listing 8: Anmelden an Azure AD mit dem Azure-AD-Connector-Konto und Ändern des Passworts eines Benutzers in AAD

```
PS > $pass = ConvertTo-SecureString '@,CeH|o-nMfEyCpg' -AsPlainText -Force
PS > $cred = New-Object System.Management.Automation.PSCredential("Sync_AADCONNECT_↵
                                                    882bef6ee0a1@2consult.onmicrosoft.com", $pass)
PS > Get-AADIntAccessTokenForAADGraph -Credentials $cred -SaveToCache
PS > Get-AADIntUser -UserPrincipalName clouduser@2consult.ch | select
DirSyncEnabled, ObjectID, UserPrincipalname
PS > Set-AADIntUserPassword -CloudAnchor "User_70f87269-f258-4473-8cca-267b50110e7b" -Password ↵
                                                          "Von1Angreifer!gesetztes.Passwort" -Verbose
```

Listing 9: Einem Dienstprinzipal Zugangsdaten hinzufügen

```
PS > $ServicePrincipal = Get-AzureADServicePrincipal -Filter "DisplayName eq 'backdoor-app'"
PS > $StartDate = Get-Date
PS > $EndDate = $StartDate.AddYears(10)
PS > New-AzureADServicePrincipalPasswordCredential -ObjectId $ServicePrincipal.ObjectId -StartDate ↵
                                                   $StartDate -EndDate $EndDate -Value "PasswortfürdenDienstprinzipal"
```

Hat ein Angreifer einmal die Cloud-Dienste einer Organisation kompromittiert, kann er auf vielen Wegen in der „Persistenz" genannten Angriffsphase seinen Zugriff auf AAD oder Azure dauerhaft und vom Opfer mehr oder weniger unbemerkt sichern. Das funktioniert analog zum Vorgehen im On-Premises AD [10].

Durch die Hintertür

Naheliegend, aber auffällig ist das Erstellen eines neuen globalen Admins oder eines Benutzers mit Mitwirkenden-Rechten für Abonnements; das kann ein regulärer Benutzer oder ein Gast sein. Wenn legitime Admins ein solches Hintertürkonto aufdecken und löschen, kann eine vom Angreifer zuvor eingerichtete Watcher-Aufgabe, die regelmäßig ausgeführt wird, oder Logic App das Konto wiederherstellen.

Eine weitere Möglichkeit ist, Anmeldeinformationen einem Dienstprinzipal hinzuzufügen, der bereits durch die ursprüngliche (Fehl-)Konfiguration der Organisation über hohe Rechte verfügt oder sie vom Angreifer mit Adminrechten zugewiesen bekommt. Die hohen Rechte können

- die AAD-Rolle eines privilegierten Authentifizierungsadministrators sein, der Authentifizierungsmethoden für jeden Benutzer, einschließlich globaler Administratoren, zurücksetzen kann;
- die RBAC-Zuweisung als Mitwirkender an einem Azure-Abonnement
- oder einzelne kritische Berechtigungen für die Microsoft-Graph-API wie `Role Management.ReadWrite.Directory`.

Um einem Dienstprinzipal Zugangsdaten hinzuzufügen, sind die Rechte des globalen, des Anwendungs- oder des Hybrididentitätsadministrators notwendig. In PowerShell funktioniert das mit einem Passwort wie in Listing 9 gezeigt; eine Anmeldeinformation kann aber auch ein Zertifikat sein.

Nun meldet der Angreifer sich mit der Identität des Dienstprinzipals und dessen Rechten an. MFA-Regeln greifen dort nicht, eine Anmeldung ist direkt ohne zweiten Faktor möglich:

```
PS > $tenantId = (Get-AzContext).↵
                                Tenant.Id
PS > $credentials = Get-Credential
PS > Connect-AzAccount ↵
      -ServicePrincipal -Credential ↵
       $credentials -Tenant $tenantId
```

Es gibt viele weitere Optionen für Eindringlinge, sich unauffällig in einer Azure-Umgebung festzukrallen. Etwa kann ein Automation-Konto zur Hintertür werden. Dem ausführenden Konto solcher automatisierter Aufgaben können RBAC-Rollen im Azure-Abo sowie AAD-Rollen zugewiesen werden. Mit diesen Rechten kann ein neuer Benutzer angelegt und privilegierten Gruppen hinzugefügt werden. Angreifer können die Hintertür jederzeit über einen sogenannten Webhook-Link aus der Ferne unauthentisiert öffnen.

Auch in einzelnen Azure-Diensten kann sich ein Eindringling einnisten: Er kann in einem virtuellen Rechner Schadsoftware oder ein lokales Administratorkonto einrichten, in einer SQL-Datenbank Firewall-Ausnahmen konfigurieren oder für ein Backup-Speicherkonto einen Schlüssel anfertigen.

Ähnlich wie bei der Kompromittierung einer Domäne oder Gesamtstruktur in einem klassischen Active Directory kann man der eigenen Cloud-Umgebung streng genommen erst wieder vertrauen, wenn man den Mandanten und die Abos neu aufsetzt.

Fazit

Da Microsofts Cloud und durch unzureichenden Schutz viele von Organisationen verwaltete Dienste wie virtuelle Maschinen, Webanwendungen oder Speicherkonten aus dem Internet erreichbar sind und weil Benutzer schwache Passwörter wählen oder auf eine Phishingmail hereinfallen, ist es für Angreifer einfach, initial Zugriff erlangen.

Durch die enge Verknüpfung von AAD und Azure-Diensten, unsichere Standardeinstellungen und Fehlkonfigurationen kann eine einzelne kompromittierte Identität oder eine Schwachstelle in einer selbst entwickelten Webanwendung dazu führen, dass Angreifer ihre Privilegien innerhalb Azure AD oder Azure erhöhen, zwischen beiden Ebenen wechseln und von der Cloud ausgehend On-Premises-Umgebungen kompromittieren.

Wie sich Organisationen dagegen schützen können, beschreibt der nachfolgende Artikel „Das Netz verstärken" ab Seite 60. (ur@ix.de)

Quellen

[1] Marko Klaus; Modellierte Kriegsführung; Realistische Vorhersage von Cyberattacken; *iX* 3/2020, S. 120

[2] Frank Ully; Nach oben gehangelt; Informationsbeschaffung – was jeder Domänenbenutzer alles sieht; *iX* 10/2020, S. 58

[3] Tobias Glemser; Zwo, eins, Risiko …; OWASP Top 10 in neuer Version; *iX* 2/2022, S. 86

[4] Christoph Puppe; Krähennest in der Wolke; Cloud Security Operations Center im Vergleich; *iX* 7/2021, S. 88

[5] Sascha Herzog; G0ne Phishing …; Red Teaming: Gezielte Fallen stellen; *iX* 9/2018, S. 106

[6] Frank Ully; Fette Beute; Passwörter und Hashes – wie Angreifer die Domäne kompromittieren; *iX* 11/2020, S. 94

[7] Frank Ully; Frisch geröstet; Roasting, Rechte, Richtlinien: Wie Angreifer sich im Active Directory Zugriff verschaffen; *iX* 12/2020, S. 92

[8] Inés Atug; Anforderungspuzzle; Microsoft 365 in Unternehmen sicher nutzen; *iX* 7/2021, S. 46

[9] Frank Ully; Vertrauensfragen; Active Directory: Wie Angreifer Tickets, Delegierung und Trusts missbrauchen; *iX* 2/2021, S. 116

[10] Yves Kraft, Frank Ully; Zwischen den Wäldern; Inter-Forest und Persistenz: Wie Angreifer sich über einen AD-Forest hinaus ausbreiten und festsetzen; *iX* 4/2021, S. 102

[11] Frank Ully; Das Netz verstärken; Azure Active Directory und Azure-Dienste absichern; *iX* 4/2022, S. 60

[12] Die im Text erwähnten Werkzeuge, Blogartikel und weiteren Materialien sind über ix.de/z9p5 zu finden.

Frank Ully

ist Head of Research der Oneconsult Deutschland AG in München. Er beschäftigt sich mit aktuellen Themen der offensiven IT-Sicherheit.