Eskalation am Vormittag

Wie man allzu oft allzu leicht Domänenadministrator werden kann – ein Erfahrungsbericht

Es ist immer wieder ernüchternd bis erschreckend, wie einfach es sein kann, die eigenen Berechtigungen im beruflich genutzten Windows-Universum unbotmäßig hochzuschrauben – und das betrifft, so unser Autor, alle Unternehmensgrößen, Branchen und Reifegrade in Sachen Sicherheits-Organisation. In seinem Erfahrungsbericht beschreibt er mit einem Augenzwinkern unnötige Arbeitserleichterungen für Penetrationstester und Innentäter – und gibt natürlich auch Tipps, wie man ein härteres Ziel sein könnte.

Von Hagen Molzer, Stuttgart

Hallo, mein Name ist Hagen Molzer und in den letzten zweieinhalb Jahren war ich bei neunzehn verschiedenen Firmen als Active-Directory-Domänenadministrator tätig. So oder so ähnlich könnte ich mich bei einem Stammtisch für IT-Administratoren vorstellen, ohne die Unwahrheit zu sagen. Andere Stammtischteilnehmer würden mich dann vermutlich in eine der drei folgenden Kategorien stecken: Schwätzer/Angeber, Administrator bei einem Managed-Service-Provider (MSP) mit vielen Kunden oder Wolf im Schafspelz.

Die zuletzt genannte Kategorie wäre dabei die zutreffendste. Tatsächlich habe ich mich bei den neunzehn Firmen in kurzer Zeit, manchmal noch vor dem ersten Mittagessen, selbst vom regulären Benutzer zum Domänenadministrator befördert – im Rahmen von beauftragten Innentäteranalysen. Bei einer solchen Analyse haben wir als Penetrationstester das Ziel, im Rahmen eines überschaubaren Projekts (5–7 Tage Aufwand) möglichst viele Schwachstellen im internen Netzwerk einer Firma zu finden und vor allem auch auszunutzen. Das war in meinen letzten zwanzig, Pardon, doch nur neunzehn Projekten so erfolgreich, dass sich die Domäne vollständig kompromittieren ließ.

Bei den „Opfern" handelte es sich sowohl um kleine (unter 50 Mitarbeiter) als auch um sehr große Organisationen (mit über 300 000 Mitarbeitern) aus ganz unterschiedlichen Branchen – etwa um Krankenhäuser, Energieversorger, Bekleidungshersteller, Maschinenbauer, Softwarehersteller oder IT-Dienstleister.

Auch der Reifegrade in puncto IT-Security reichte von „es ist die erste Überprüfung dieser Art" bis zu „wir machen regelmäßig Pentests und setzen die Empfehlungen anschließend auch um". Dass wir als Angreifer in dieser Art von Penetrationstest dennoch in den allermeisten Fällen erfolgreich sind, hat (im Gegensatz zur Wahrnehmung der Konsumenten von Hollywood-Filmen und manchen Medien, dass Hacker zur Ausübung schwarzer Magie befähigt sind) viel mit der an sich wohlbekannten Asymmetrie zu tun, dass die Verteidiger jede einzelne Sicherheitslücke in ihrer Umgebung erkennen und schließen müssen, Angreifer hingegen nur eine oder wenige Schwachstellen finden, ausnutzen und gegebenenfalls mit anderen verketten müssen, um ihr Ziel zu erreichen.

Dass die Admins heutiger Netzwerke häufig nicht einmal jedes System in ihrem Netzwerk – geschweige denn deren Patch-Stände – kennen, macht es nicht besser. Außerdem steht eine große Auswahl kostenloser und in den meisten Fällen auch Open-Source-Werkzeuge der sehr aktiven IT-Security-Community bereit, welche die Aufgabe von Pentestern (und Angreifern) um ein Vielfaches erleichtert.

Das ist auch der Grund, warum man als Pentester in besonders großen und professionell verwalteten Umgebungen global agierender Kunden trotzdem oft erfolgreich bleibt, denn hier besteht eine besonders große Asymmetrie. Auch wenn man in solchen Fällen oft im Vorgespräch während der Projektvorbereitung vereinbart, dass das Security-Team der betreffenden Firma bei verschiedenen Alarmen der Anti-Viren-(AV)- oder SIEM-Lösung mal wohlwollend zur Seite schaut: Ein unauffälligeres Vorgehen wäre meistens ebenfalls möglich, würde aber den zeitlichen Aufwand für das Projekt schnell vervielfachen, ohne zusätzliche Erkenntnisse über vorhandene Schwachstellen zu liefern.

Wege zum Erfolg

Ein paar anschauliche Beispiele zeigen schnell, warum es in internen Netzwerken so einfach ist, ein erfolgreicher „Hacker" zu sein – und skizzieren im Umkehrschluss, wie man Innentätern, Eindringlingen und Pentestern das Leben etwas schwerer machen könnte.

Passable Password-Hashes

Bei einer Innentäteranalyse für eine Bank konnten wir auf unserem Ausgangspunkt (regulärer Windows-10-Client eines Benutzers) über eine schwach konfigurierte Pfad-Variable unsere lokalen Privilegien erweitern und uns lokale Administratorberechtigungen auf dem System verschaffen. Mit diesen erweiterten Rechten war es mit dem Werkzeug *Mimikatz* möglich, den NTLM-Hash (der in Windows-Netzwerken weitgehend zum Passwort äquivalent ist) des Benutzerkontos der Softwareverteilung aus dem Arbeitsspeicher auszulesen. Dieser Account verfügte, wie es sich für eine Softwareverteilung eben gehört, über administrative Rechte auf einem Großteil der internen Infrastruktur. Damit war es ein Leichtes, einen Pass-the-Hash-Angriff mit dem Softwareverteilungsbenutzer auf ein anderes Windows-System im Netzwerk durchzuführen, an dem zu diesem Zeitpunkt ein Domänenadministrator angemeldet war. Dieser Account (bzw. dessen NTLM-Hash) ließ sich dort, wieder per Mimikatz, kompromittieren – somit waren wir in der Lage, die Identität eines Domänenadministrators im internen Netzwerk anzunehmen.

Ein weiteres, noch simpleres Beispiel trug sich erst kürzlich zu: Wir durchsuchten mithilfe eines Open-Source-Werkzeugs und mit einem Active-Directory-(AD)-Konto mit regulären Benutzerrechten das Netzwerk nach Dateifreigaben, auf die wir Leserechte hatten. Das Werkzeug sucht in den Freigaben dann nach Dateien, in denen sich oft sicherheitskritische Inhalte befinden: zum Beispiel PowerShell-Skripte mit einkodierten Zugangsdaten im Klartext oder ungeschützte SSH-Private-Keys. Dabei fiel in einer mehr als fünf Jahre alten Skriptdatei die Verwendung des Passworts „Pa$$w0rd" auf. Nach einiger Zeit in der Branche weiß man als Pentester, dass das Thema Passworthygiene (also das Verwenden eines einzigartigen, ausreichend komplexen und nicht vorhersehbaren Passworts für jedes einzelne Benutzerkonto beziehungsweise System) ein relativ neuer Trend ist – genauso wie die regelmäßige Änderung der Passwörter technischer Benutzer (z. B. von Dienste-Konten).

Also: Warum nicht einfach mal „Pa$$w0rd“ in einem sogenannten Passwort-Spraying-Angriff gegen alle in der Domäne vorhandenen Benutzer prüfen? Eine solche Liste der Anmeldenamen aller Domänenbenutzer lässt sich schließlich mit regulären Benutzerrechten vom Domänencontroller erfragen. Und siehe da: Sechs Konten nutzten dieses vermeintlich „komplexe“ Passwort (es umfasst ja schließlich Zeichen aus allen vier Kategorien: Großbuchstaben, Kleinbuchstaben, Zahlen und Sonderzeichen). Man mag es jetzt vielleicht nicht glauben, aber eines der sechs Konten war tatsächlich Mitglied der Domänenadministratorengruppe – damit war die Domäne noch vor dem ersten Kantinengang kompromittiert.

Zielführende Zertifikate

Eine weitere Quelle für häufig vorhandene und sicherheitskritische Schwachstellen ist eine in die Domäne integrierte Microsoft Public-Key-Infrastructure (PKI). Solch eine PKI ist mit nur wenigen Klicks installiert – sie stellt dann schnell und zuverlässig verschiedene Zertifikate für unterschiedliche Verwendungszwecke aus und alle Systeme in der Domäne vertrauen ihren Inhalten. Nur wenige Administratoren beschäftigen sich aber nach der Installation tiefer gehend mit den Detailkonfigurationen, Fallstricken und einer etwaigen Härtung der angebotenen Zertifikat-Templates oder der PKI selbst.

Daher ist eine der „Low-hanging Fruits“ zu Beginn eines jeden Pentests die Überprüfung, ob ein regulärer Domänenbenutzer oder ein Computerkonto (wie unser regulärer Windows-10-Client, von dem wir starten) berechtigt ist, ein Zertifikat auf Basis einer Zertifikatvorlage zu beziehen, die eine Benutzerauthentifizierung erlaubt – und ob es gleichzeitig möglich ist, im „Subject-Alternative-Name“-(SAN)-Feld des Zertifikats ein weiteres Principal anzugeben. Ist das der Fall, schreibt man einfach ein Domänenadministratorenkonto (zum Beispiel administrator@kundendomäne.local) in den SAN, bezieht das Zertifikat und nutzt anschließend das Angriffswerkzeug *Rubeus*, um auf Basis des ausgestellten Zertifikats ein Kerberos-Ticket-Granting-Ticket (TGT) im Kontext des Domänenadministrators zu erhalten. Vorausgesetzt, diese Schwachstelle ist vorhanden, dauert diese Attacke circa fünf Minuten und die Domäne ist vollständig kompromittiert. Ein solcher Angriffsvektor war in dieser oder einer abgewandelten Form bei fünf meiner letzten neunzehn Projekte erfolgreich.

Multi-Pass – und die rühmliche Ausnahme

Da stellt sich schon fast die Frage, was dem Pentester eigentlich noch zu tun bleibt, wenn er sein Ziel bereits am Montagvormittag erreicht hat? Bei einem typischen sechstägigen Pentest hat man schließlich noch rund drei Tage Zeit, bevor an den letzten beiden Projekttagen die Dokumentation droht ... Doch die Suche nach weiteren Schwachstellen wird selten langweilig – schließlich versucht man, diese zu anderen, von den ersten Angriffen abweichenden Pfaden zu verketten, um einen möglichst umfassenden und breiten Abschlussbericht zu liefern, der möglichst viele der vorhandenen Schwachstellen erfasst – die der Auftraggeber anschließend beheben kann. Im Durchschnitt finden wir pro Innentäterprojekt tatsächlich zwei bis drei voneinander unabhängige Angriffspfade zur Kompromittierung der Domäne.

Aber zum Glück – oder aus der Perspektive des Pentesters „leider“ – gibt es, wie bei den meisten Regeln, auch Ausnahmen. Letztes Jahr habe ich nichts ahnend eine Innentäteranalyse bei einem bayerischen Mittelständler begonnen, die meinen bis dahin siebzehn Domänen-Kompromittierungen in Folge umfassende Killstreak jäh beendete: Dieser Kunde hatte alle Low-hanging Fruits im Vorfeld bereits selbst abgeerntet, diverse typische Sicherheitslücken im Active Directory geschlossen und das Tiering-Modell von Microsoft sowie eine strikte Mikrosegmentierung im internen Netzwerk eingeführt. Nach viel Frustration und noch mehr freiwilligen Überstunden musste ich mich am Ende des Projekts mit der Kompromittierung eines Tier-1-Serveradministrators zufriedengeben, der zwar weitreichende administrative Rechte auf vielen Servern hatte, wovon ich aber fast alle über das Netzwerk nicht erreichen konnte. Im gesamten Projektverlauf war ich nicht in der Lage, auch nur eine einzige erfolgreiche RDP-Verbindung aufzubauen :-(

Härter ist besser!

Das bringt uns zu den Empfehlungen zur besseren Absicherung der IT-Infrastruktur. Open-Source-Hacking-Werkzeuge tun einerseits, was ihr Name schon besagt. Andererseits können Administratoren sie aber auch nutzen, um selbst nach Schwachstellen zu suchen und diese anschließend zu beheben. Zum Beispiel lässt sich die Active-Directory-Domäne mit den Tools *BloodHound* (https://github.com/BloodHoundAD/BloodHound) und *PingCastle* (https://github.com/vletoux/pingcastle) sehr effizient auf Angriffspfade innerhalb der Domäne und generell auf Schwachstellen und typische Fehlkonfigurationen überprüfen. Als weiteres Werkzeug prüft *Certify* wichtige Aspekte der Konfiguration von Zertifikatvorlagen einer Microsoft PKI (https://github.com/GhostPack/Certify).

Mehrere Ebenen für mehr Sicherheit

Und nicht zuletzt das Beispiel des bayerischen Mittelständlers hat eindrücklich gezeigt, dass die Einführung des Tiering-Modells (aka Enterprise-Access-Modell) von Microsoft [1,2], idealerweise gepaart mit einer strikten

Mikrosegmentierung des Netzwerks, Angreifern das Leben viel schwerer macht.

Beim Tiering-Modell von Microsoft teilt man die IT-Infrastruktur in mehrere Ebenen ein – typischerweise in:

- Tier 2: Clients und Terminalserver
- Tier 1: Server
- Tier 0: Domänencontroller und weitere besonders sicherheitskritische Systeme

Anschließend erhalten administrative Benutzerkonten nur noch Berechtigungen innerhalb der jeweiligen Ebene, um es einem Angreifer zu erschweren, seine Privilegien auf andere Ebenen auszuweiten. Außerdem werden in dem Zuge verschiedene Maßnahmen zur Absicherung der neuen hochprivilegierten Benutzerkonten implementiert, zum Beispiel sogenannte Privileged-Access-Workstations (PAWs) eingeführt: separate, speziell gehärtete Systeme, die man ausschließlich für die Administration verwendet.

Minimierte Konnektivität für weniger Angriffspfade

Eine weitere wirkungsvolle Maßnahme, die das Tiering-Modell sehr gut (nach dem Zwiebelschalenprinzip oder Defense-in-Depth-Ansatz) ergänzt, ist die Einführung einer Mikrosegmentierung, das heißt das strikte Filtern des Datenverkehrs zwischen jedem einzelnen Netzwerk-Host. Flache Netzwerk-Topologien, in denen jedes Gerät jedes andere Gerät im Netzwerk auf allen Ports erreichen konnte, sind damit (endlich) Geschichte – und auch Geräte, die sich in demselben VLAN befinden, können sich nicht mehr gegenseitig erreichen, wenn sie das nicht müssen.

Das erklärt auch, warum meine RDP-Verbindungen und Lateral-Movement-Versuche als Serveradministrator im zuletzt genannten Beispiel durchgängig gescheitert sind: Zum Ende des Projekts hat mich der Kunde, in einem sympathischen bayerisch angehauchten Dialekt, aufgeklärt, dass RDP – genauso wie fast alle anderen Ports und Protokolle, die administrativen Zugriff ermöglichen – bei ihm im ganzen Netzwerk ausschließlich von einer Handvoll PAWs erlaubt sind ...

Von Bären und anderen Menschen

Zweck der aufgeführten Maßnahmen ist es, entweder dafür zu sorgen, dass ein realer Angreifer aufgibt und sich ein einfacheres Opfer sucht, oder als Verteidiger so viel Zeit zu gewinnen, dass man auf den Angreifer aufmerksam wird, bevor er sein Ziel erreicht hat – und ihn so wieder aus dem Netzwerk werfen kann, bevor großer Schaden entsteht. Außerdem bleibt man als gutes Beispiel im Gedächtnis eines Pentesters und wird womöglich löblich, wenn auch anonym, in einem Fachartikel in der <kes> erwähnt ;-) ■

Hagen Molzer ist Leitender Berater bei der cirosec GmbH – in seinen Schwerpunktbereichen Active-Directory- und Windows-Betriebssystem-Sicherheit führt er Penetrationstests, Beratungsprojekte sowie Red-Team-Assessments durch.

Literatur

[1] Microsoft, Ebenenmodell zur Partitionierung von Administratorrechten, Januar 2023, https://learn.microsoft.com/de-de/microsoft-identity-manager/pam/tier-model-for-partitioning-administrative-privileges

[2] Microsoft, Enterprise-Zugriffsmodell, Januar 2023, https://learn.microsoft.com/de-de/security/compass/privileged-access-access-model