AD-Härtungsmaßnahmen jenseits von Group Policies

Mehr ist mehr

Marco Wohler

Viele Einzelmaßnahmen verbessern zusammen die Sicherheit des Active Directory deutlich. Ein Angriff auf die eigene IT zeigt, wo es noch Schwächen gibt.

Der vorliegende Teil der Active-Directory-Reihe setzt die Härtungsmaßnahmen des vorangegangenen Artikels fort. Konnte man dort schon mit regelmäßigen Updates, einer guten Passwort-Policy und den richtigen Gruppenrichtlinien, also mit relativ wenig Aufwand, ein gutes Maß an Sicherheit erzielen, so geben die nun vorgestellten Maßnahmen den letzten Schliff. Das Augenmerk gilt nun dem Schutz der Zugangsdaten und weiteren Maßnahmen wie dem gezielten Einsatz von Firewalls.

Der Artikel zeigt auch, wie man sich mit Angriffswerkzeugen und anderen Tools selbst auditieren kann. Bei der Absicherung eines komplexen Gebildes wie des Active Directory ist es wichtig, die Zusammenhänge zwischen den verschiedenen Funktionen, Protokollen und Maßnahmen zu sehen. Ein Angriff auf die eigene Infrastruktur kann dabei helfen, Lücken aufzuzeigen und die Kenntnisse über die Infrastruktur zu erweitern. Wo nötig, wurden für Interessierte Hinweise zu externen Ressourcen genannt, die zur Vertiefung der Themen genutzt werden können (sie sind über ix.de/zqqa zu finden). Die Kenntnis der vorangegangenen Artikel [1–8] und die Vermeidung der dort geschilderten Fehlkonfigurationen und Nachlässigkeiten wird vorausgesetzt.

Nachdem die Group Policies wie in [8] beschrieben konfiguriert und die dort geschilderten Härtungsmaßnahmen umgesetzt wurden, steht nun der Schutz privilegierter Accounts an. Windows bringt dafür schon einige Funktionen mit, etwa in Form der Gruppe „geschützte Benutzer“ („Protected Users“), zu der alle privilegierten Benutzerkonten gehören sollten. Dadurch werden implizit Härtungsmaßnahmen auf die Konten angewandt. So werden unter anderem veraltete Authentisierungsprotokolle nicht mehr akzeptiert und Kerberos-Einstellungen zu Verschlüsselung und Ticketlebenszeit neu gesetzt. Eine Übersicht über die Gruppe und ihre Funktionen ist bei Microsoft zu finden (siehe ix.de/zqqa).

Schutz von Zugangsdaten

Die Aktivierung von „Konto ist vertraulich und kann nicht delegiert werden“ in den Eigenschaften eines Benutzers stellt sicher, dass die Anmeldedaten eines Kontos nicht von einer vertrauenswürdigen Anwendung aus an andere Computer oder Dienste im Netzwerk weitergeleitet werden können. Bei allen privilegierten Benutzerkonten sollte diese Option angeschaltet sein.

Ein weiteres Feature für die Serverversionen 2016, 2019 sowie für die Windows-10-Versionen Enterprise und Education ist Credential Guard. Es startet einen weiteren, isolierten LSA-Prozess (Local Security Authority, zu Deutsch: lokale Sicherheitsautorität). Im LSA-Prozess werden normalerweise geheime Daten wie auch Kerberos-Tickets abgelegt. Nach Aktivieren der Funktion werden diese Daten in einem durch Virtualisierung geschützten Prozess gespeichert, auf den das restliche Betriebssystem nicht zugreifen kann. Das verhindert insbesondere das einfache Auslesen von Hashes und Kerberos-AES-Schlüsseln [4] und damit nachgelagert auch die Angriffe Pass the Hash, Overpass the Hash und Pass the Key.

Aber Achtung: Credential Guard schützt keine lokal gespeicherten Geheimnisse wie SAM- (Security Account Manager, Sicherheitskontenverwaltung) und Dienstkonten. Die LSA kann per Remote Procedure Call (RPC) mit dem isolierten LSA kommunizieren, um Funktionen wie das Anmelden an einem System zu ermöglichen. Ältere Authentisierungsverfahren wie NTLMv1, MS-CHAPv2, Digest und CredSSP verlieren mit Credential Guard ihre Funktion für einmaliges Anmelden, können aber immer noch verwendet werden. Wie Credential Guard per Gruppenrichtlinienobjekt (Group Policy Object, GPO) eingerichtet wird und was alles beachtet werden muss, beschreibt Microsoft in einem Beitrag (zu finden über ix.de/zqqa).

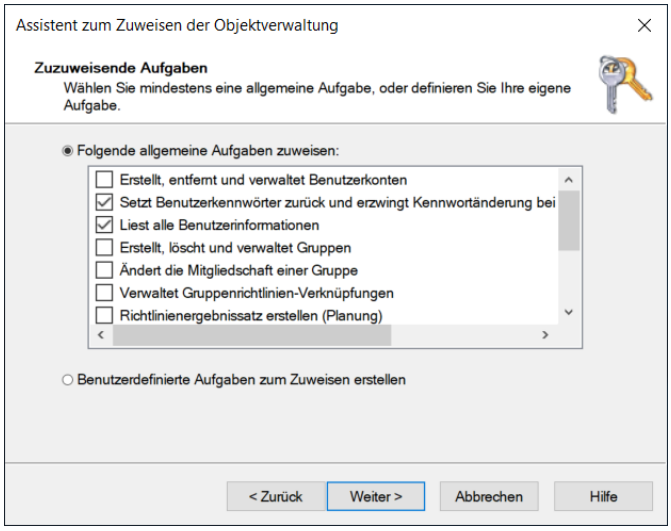
Least Privileges – nur das, was nötig ist

Ein Sicherheitsprinzip in der Informationssicherheit ist das Prinzip der „Least Privileges“. Es bedeutet, dass jeder gerade

nur mit den Berechtigungen ausgestattet wird, die für eine Aufgabe erforderlich sind, lokal wie auf Domänenebene. Brauchen die Benutzer eine Dateifreigabe nicht, wird sie ihnen gar nicht angezeigt. Muss der Zugriff nur auf ein System erfolgen, wird die Anmeldung an anderen Systemen untersagt. Bei Berechtigungen für normale Benutzer ist das in der Regel klarer durchgesetzt als bei anderen Benutzergruppen. So gibt es in Unternehmen häufig Benutzerkonten mit Domänenadmin-Rechten, die für ihre Aufgabe eigentlich gar nicht so viele Berechtigungen benötigen, jedoch oftmals aus Bequemlichkeit in die Gruppe aufgenommen wurden, weil es „dann halt funktioniert". Auch loggen sich Domänenadmins immer mit ihrem privilegierten Benutzer ein, selbst für alltägliche Arbeiten wie Mail und Internet.

Grundsätzlich empfiehlt sich folgendes Vorgehen: Berechtigungen so granular wie möglich vergeben (lokal und in der Domäne), mit delegierten Berechtigungen arbeiten, unnötige Berechtigungen entfernen und privilegierte Benutzer regelmäßig auditieren. Ein Domänenadministrator sollte daher zwei bis n Benutzerkonten besitzen. Ein normales Standardkonto, ein Domänenadministratorkonto und darüber hinaus weitere Konten, beispielsweise ein lokales Benutzerkonto oder spezielle Konten für den Zugriff auf weitere Ressourcen.

Oft sind Dienstkonten mit zu hohen Berechtigungen vorhanden [4]. Sie agieren mit Domänenadmin-Rechten oder sind auf allen Servern per GPO in die lokale Administratorengruppe eingetragen. Ein gutes Beispiel aus der Praxis sind Softwareverteilsysteme. Für das Installieren der Programme sind erhöhte Berechtigungen notwendig. Anstatt diese als LocalSystem laufen zu lassen, werden sie vielerorts unter einem Domänenadmin-Benutzer ausgeführt, was einem Angreifer in die Hände spielt. Dienstkonten sollten daher immer nur die nötigsten Berechtigungen besitzen. Das bedeutet insbesondere, den Zugriff auf die nötigen Systeme zu reduzieren, sodass ein Dienstkonto nicht als Sprungbrett in die ganze Domäne verwendet werden kann.



Der Assistent vereinfacht das Zuweisen delegierter Berechtigungen (Abb. 1).

Dienste sollten daher mit LocalService, NetworkService, LocalSystem oder einem gMSA (gruppenverwaltete Dienstkonten)[8] ausgeführt werden. Die Berechtigungen der in Windows eingebauten Dienstkonten können bei Microsoft nachgelesen werden (siehe ix.de/zqqa). Administrative Berechtigungen in der Domäne sind in den wenigsten Fällen nötig, sie können den meisten Dienstkonten entzogen werden.

Delegierte Berechtigungen sind ein weiterer Baustein, Benutzern und (Teil-) Administratoren nur den Zugriff zu gewähren, der erforderlich ist. Im AD hilft der Dialog „Objektverwaltung zuweisen …". In der „Active-Directory-Benutzer- und -Computer"-Konsole lässt sich per Rechtsklick auf ein Element (zum Beispiel eine Organisationseinheit) der Assistent starten. Er hilft dabei, Benutzern oder einer Gruppe genau die Berechtigungen zuzuteilen, die sie für ihre Arbeit benötigen.

Mehr Verschachtelung, weniger Überblick

Eine weitere Schwierigkeit, auf die man oft stößt, sind verschachtelte Gruppen. Man erkennt in ihnen nur schwer, wer welche Berechtigungen hat oder nicht hat. Eine neue Benutzerin, die der Gruppe „Supporter" hinzugefügt wird, erhält durch die Verschachtelung implizit die Domänenadmin-Rechte (siehe Abbildung 2). Ein Angreifer findet jedoch genau diese Sachverhalte und nutzt sie aus, wie in [4] und [7] zu sehen war. Verschachtelte Gruppen sollte man vermeiden und, wenn das nicht möglich ist, zumindest gut dokumentieren.

Wenn Domänenadministratoren oder andere privilegierte Benutzer nicht genügend geschützt sind, können sie kompromittiert zum Ziel von DCSync-Angriffen wie in [4] werden. Mit erfolgreichem DCSync besitzt der Angreifer die Hashes aller Passwörter aus dem AD. Damit dies gelingt, muss der kompromittierte Benutzer die entsprechenden Berechtigungen haben. Standardmäßig verfügen Mitglieder der Gruppen Domänenadmins, Organisationsadmins sowie Domänencontroller über diese Rechte. Der Systemverantwortliche sollte kontrollieren, wer sie besitzt.

Andere als diese hoch privilegierten Standardbenutzer sollten keine DCSync-Rechte besitzen. Weitere potenziell gefährliche Rechtezuweisungen können über Access Control Lists (ACLs) [5] bestehen. Da niemand weiß, welche Berechtigungen ein spezifischer Benutzer wirklich hat, bleiben sie oft unerkannt. Das in [3] vorgestellte PowerShell-Skript PowerView und dessen Befehl `Find-InterestingDomainACL` [5] sowie der AD ACL Scanner (siehe ix.de/zqqa) können zu weitreichende Berechtigungen finden. Auch andere hoch privilegierte AD-eigene Gruppen wie Konten- oder Serveroperatoren [5] sind nur wenige Kommandozeilenbefehle davon entfernt, Domänenadministratoren zu werden – und sollten im Normalfall keine oder nur wenige Mitglieder haben.

Die Gruppe „Prä-Windows-2000-kompatibler Zugriff" kann es Angreifern un-

iX-TRACT

- Viele einzelne Härtungsmaßnahmen entwickeln ihr volles Potenzial nur in Kombination mit anderen Maßnahmen.
- Besonders privilegierte Benutzer wie Domänenadmins müssen geschützt werden. Das setzt bei den Systemverwaltern und Supportmitarbeitern ein Verständnis für die Maßnahmen voraus.
- Es lohnt sich, sich selbst einmal anzugreifen, da dies zu „Aha-Erlebnissen" führt und man dabei die Sicht eines Angreifers einnehmen kann, der nur eine einzelne Lücke finden muss.

ter Umständen ermöglichen, viele Informationen über das AD anonym abzufragen. Microsoft hat sie ins Leben gerufen, um Windows-NT-Domänen die Interoperabilität mit Active-Directory-Domänen zu ermöglichen, indem sie den nicht authentifizierten Zugriff auf bestimmte Daten erlaubt. Heute findet die Gruppe praktisch keine Verwendung mehr und sollte daher keine Benutzer und Gruppen mehr enthalten. Wenn darin noch die Gruppen „Anonymous-Anmeldung“ oder „Jeder“ enthalten sind, sollte die Systemverantwortliche sie entfernen.

Umgebungen mit unterschiedlichen Sicherheitsanforderungen sollten in eigene Forests (Gesamtstrukturen) ausgelagert werden, weil Domänen innerhalb eines Forests keine Sicherheitsgrenzen bilden [6].

Vertrauen gut planen und begrenzen

Bei Sicherheitsvorfällen haben wir häufig entdeckt, dass es auch zwischen unterschiedlichen Forests und den darin enthaltenen Domänen zu weitreichende Vertrauensstellungen (Trusts) gibt [7], die beispielsweise externen Partnerdomänen zu viele Berechtigungen einräumen. So haben wir einen Fall untersucht, bei dem ein Angreifer sich durch erfolgreiches Kompromittieren der Partnerdomäne Zugriff auf die lokale Domäne verschafft hatte, um dort sein Unwesen weiterzutreiben.

Um dem entgegenzuwirken, sollten die Verantwortlichen neue Vertrauensstellungen präzise planen und bestehende überprüfen. Die folgenden Punkte sind dabei wichtig: SID-Filter (Security Identifier) und Quarantäne sollten aktiviert sein (bei der Quarantäne standardmäßig der Fall) und die ausgewählte Authentifizierung sollte konfiguriert werden (standardmäßig ausgeschaltet). SID-Filter und Quarantäne sind quasi dasselbe; es gelten aber andere Notationen für Domain- beziehungsweise Forest-Trusts.

Mit Selective Authentication kann eine Administratorin entscheiden, auf welchen Computerobjekten sie für Benutzer aus der Partnerdomäne die Berechtigung „Allowed to Authenticate“ vergeben möchte. Die Rechte sollten möglichst restriktiv (Least Privileges) vergeben werden. Eine Quelle für weitere Informationen und die entsprechenden Befehle ist das Webportal TechNet von Microsoft (siehe ix.de/zqqa).

Aus Sicherheitssicht außerdem zu beachten sind Dienste, die in zwei verschiedenen Forests laufen und voneinander abhängig sind. Sie bilden eine Kette für einen Angreifer, der sich beispielsweise von einem Webserver auf einen SQL-Server in einem anderen Forest hangeln kann [7]. Um das zu verhindern, sollte das Dienstkonto auf dem SQL-Server nur lokale Berechtigungen besitzen und der Server sollte womöglich in einer DMZ stehen. Der Autor würde einen SQL-Server tendenziell nicht an einer Domäne betreiben oder über einen Read-only Domain Controller (RODC, zu Deutsch: Domänencontroller ohne Schreibberechtigungen) anbinden, ohne direkten Zugriff auf den Domänencontroller.

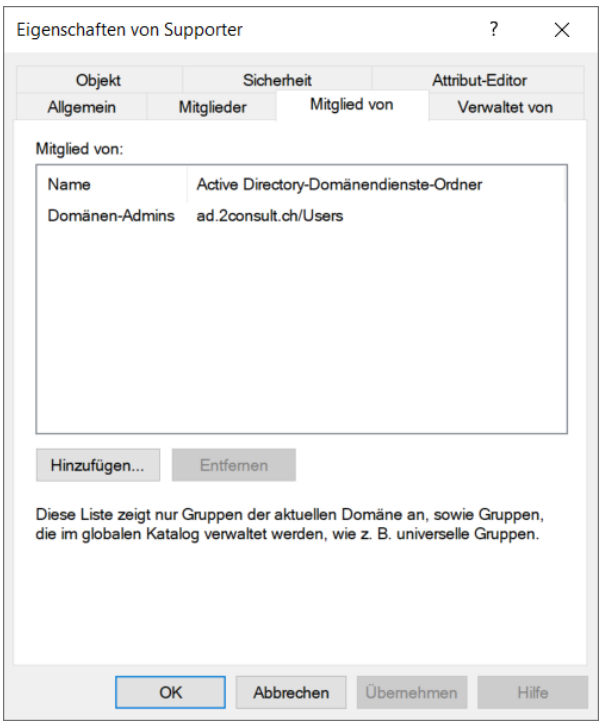
Ein weiteres Problem im lokalen Kontext, aber auch auf Netzwerkfreigaben, können falsch gesetzte NTFS-Berechtigungen sein. Einerseits sollten Dateien mit potenziell sensiblen Inhalten wie Passwörtern nicht ungeschützt und für jeden lesbar auf Netzwerkfreigaben liegen [3]. Andererseits kann ein Angreifer womöglich eine ausführbare Datei ersetzen, wenn normale Benutzer auf dem Pfad, beispielsweise C:\ProgramFiles\, Schreibrechte besitzen, wo sie eigentlich keine haben sollten. Wird das Programm oder Skript mit erhöhten Rechten gestartet, erlangt der Angreifer diese Berechtigungen ebenfalls. Falsch gesetzte NTFS-Berechtigungen können mit dem Werkzeug Sysinternals AccessEnum (siehe ix.de/zqqa) auditiert werden.

Nicht jeden überall schreiben lassen

Grundsätzlich sollten normale Benutzer nur in ihrem Benutzerprofil Schreibberechtigungen besitzen. Es gibt einige wenige Windows-Standardpfade, bei denen Ausnahmen gelten. Auch schon gesichtet: Wenn die Benutzerprofile mit falsch konfigurierten Berechtigungen auf einem Netzlaufwerk liegen, können Benutzer die Profile anderer Benutzer anschauen oder gar verändern. Solche Konfigurationen können schnell auch rechtliche Sanktionen nach sich ziehen, wenn private Daten anderer nicht gegen unbefugte Zugriffe geschützt sind.

Genau wie bei Ordnern und Freigaben gibt es Berechtigungen auch auf OUs (Organisationseinheit im Active Directory). Änderungen daran sind mit Vorsicht zu genießen, können aber unter Umständen die Sicherheit erhöhen, wenn nicht alle Benutzer und Geräte alle Organisationseinheiten einsehen können. Standardmäßig haben alle Teilnehmer Leserechte für die meisten Objekte und Attribute im gesamten AD. So kann ein Standardbenutzer auslesen, welche Domänenadmins es gibt, wie die OU-Baumstruktur aufgebaut ist oder wann sich ein beliebiger Benutzer zuletzt angemeldet hat und vieles mehr.

Das Audit- und Sicherheitswerkzeug BloodHound [4] holt sich, um ein Beispiel zu nennen, auf diese Weise wesentliche Informationen aus dem Active Directory. Zu Testzwecken wurden einem Benutzer auf allen OUs, die er nicht benötigt, die Leserechte per Verweigern-Regel (Deny Rule) entzogen. Dabei hatte der Benutzer noch Zugriff auf die Organisationseinheiten der Standardbenutzer sowie auf diejenige, in der sich sein Gerät befindet. Das Resultat war überraschend: BloodHound brach die Informationssuche abrupt und ohne Fehlermeldung ab und hinterließ nur unvollständige Daten. Der Versuch, die trotzdem vorhandenen Daten zu visualisieren, scheiterte. Eine Suche innerhalb der Daten belegte, dass die Server, Domänenadministratoren und andere wichtige



Verschachtelte Gruppen haben nicht nur den Nachteil der Unübersichtlichkeit, sondern bergen auch noch die Gefahr, dass Mitglieder zu viele Rechte erhalten (Abb. 2).

Objekte nicht gefunden wurden und diese Informationen einem Angreifer nicht zur Verfügung stehen würden.

Firewalls: Abschottung ist noch immer Trumpf

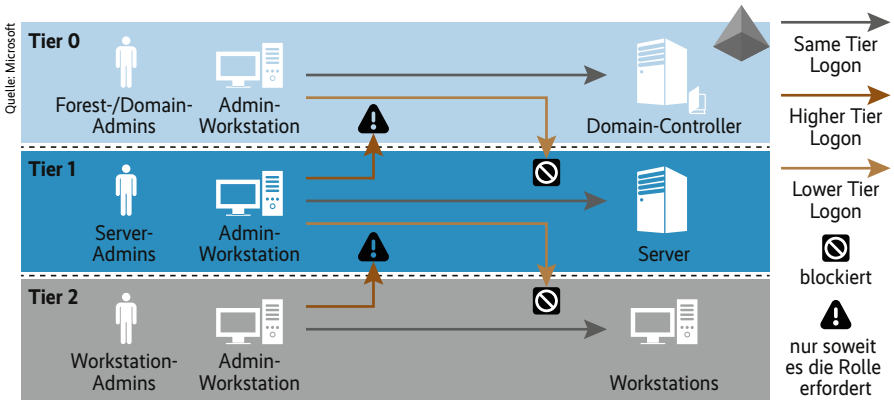
Um es Enumerationswerkzeugen wie BloodHound oder PowerView noch schwerer zu machen, Daten über die Infrastruktur zu sammeln, spielen Firewalls zwischen Netzwerken, aber auch lokale Firewalls eine große Rolle. Auch die weitere Ausbreitung des Angreifers im Netzwerk wird so effektiv erschwert.

Firewalls sind, richtig eingesetzt, eine bewährte und gute Methode, Angreifern und Malware das Eindringen ins oder das Verbreiten im Netzwerk zu erschweren. Eine Netzsegmentierung, die nur die nötigste Kommunikation zwischen den Netzwerkbereichen zulässt, kann einiges verhindern. So müssen Standardbenutzer, die im Büro arbeiten, nicht auf den Remote-Desktop-Port eines Servers gelangen, den SMB-Port der internen Zertifizierungsstelle anfragen oder Maschinen in der Produktion erreichen können. Sind solche Zugriffe nicht unterbunden, können Angreifer sich schneller und besser im Netzwerk ausbreiten und mehr Informationen auslesen – etwa über lohnende Ziele und die Pfade dorthin.

Auch spielt die lokale Firewall eine oft unterschätzte Rolle. Die Windows Defender Firewall ist in jedem Windows Server und Client enthalten und kann per GPO konfiguriert werden. Gute Praktiken und Anleitungen gibt es im Internet (siehe ix.de/zqqa). Können Clients einander dank lokaler Firewall im selben Netzwerk nicht erreichen, können sich Malware und Angreifer nicht von einem Client auf andere Clients ausbreiten. Ein weiterer positiver Effekt: Werkzeuge wie BloodHound erheben viele Informationen durch ein kurzes Verbinden zu anderen Systemen, etwa per SMB-Protokoll. Sind diese Verbindungen nicht möglich, stehen weniger Daten für die Auswertung zur Verfügung. Es wird dadurch schwieriger herauszufinden, welche Benutzer Sitzungen auf wichtigen Zielen haben. Zentral ist dabei das korrekte Einrichten und Aktivieren der Firewalls, sei es lokal oder zwischen Netzwerken.

Tier-Modell und PAW – privilegierte Benutzer schützen

Microsoft hat Konzepte vorgestellt, die beschreiben, wie verschiedene Angriffe, insbesondere Key-Logging, (Over-)Pass the Hash [4] und Pass the Ticket [6], abgewehrt werden können (siehe ix.de/zqqa). Der Kerngedanke dabei ist, privilegierte Benutzer und Systeme in einem Tier-Modell – also auf verschiedenen Ebenen – voneinander zu trennen und speziell zu schützen. Tier 0 ist dabei die am besten geschützte Zone, in der sich auch der Domänencontroller befindet. Verwaltet wird sie über speziell gesicherte Geräte namens Privileged Access Workstations (PAW). Diese Geräte werden ausschließlich für das Management der Tier-0-Umgebung eingesetzt und nicht mit anderen Netzwerken verbunden. Angreifer haben so keine Chance, von anderen Tiers auf Tier 0 zu gelangen (Rechteausweitung), da alle Aktionen von privilegierten Benutzern komplett getrennt stattfinden.



Nach dem Tier-Modell von Microsoft sind Zugriffe nur in definierte Richtungen erlaubt. So sind höhere Privilegien vor Missbrauch von außen besser geschützt (Abb. 3).

Das setzt jedoch voraus, dass Benutzer mit entsprechend hohen Berechtigungen nicht auch noch an anderen Orten eingesetzt werden. Wie schon beschrieben sollten Servicebenutzer oder Benutzer für den Support eingeschränkte Rechte haben und nicht als Domänenadmin ihre Arbeit verrichten. Eine einfache Variante einer solchen Umgebung könnte folgendermaßen eingerichtet werden: Ein Administrator arbeitet mit seinem hoch privilegierten Account auf dem Betriebssystem des Rechners, der mit dem Tier-0-Netzwerk verbunden ist. Bei mobilen Geräten könnten ein VPN und die lokale Firewall regeln, dass das Gerät nur Zugriff auf Tier 0 erhält. In einer VM kann der Admin mittels normalem Benutzerkonto auf alle anderen Ressourcen sowie das Internet zugreifen. Fängt er sich in der VM dann Schadsoftware ein, bleibt sie dort und hat keinen Zugriff auf Tier 0.

Wer noch einen Schritt weiter gehen möchte, kann einen weiteren Forest namens Enhanced Security Administrative Environment aufsetzen (ESAE, von Microsoft auch als „Red Forest“ bezeichnet). In dieser separaten Gesamtstruktur werden nur die Administratorenkonten, die PAWs und die dazu nötigen Gruppen eingerichtet. Die administrative Gesamtstruktur hat somit nichts mehr mit der produktiven zu tun, da sie getrennt funktionieren. Zum Verwalten des produktiven AD richtet man eine unidirektionale Vertrauensstellung mit den minimal notwendigen Berechtigungen ein. Normale Benutzerkonten aus dem ESAE können als Domänenadministratoren der produktiven Umgebung eingesetzt werden. Die Administratoren können über ihre PAW aus dem Tier 0 heraus mit Konten aus dem administrativen Forest das AD der Produktion verwalten.

Der Vorteil der beiden Maßnahmen Tier-Modell und ESAE ist der hohe Schutz privilegierter Benutzerkonten. Für Angreifer gibt es keinen direkten Weg mehr, die Domäne einzunehmen, da alle Administratoren vom Netzwerk getrennt über einen definierten Vektor die Umgebung verwalten (siehe Abbildung 3) und sogar – beim Einsatz von ESAE – auch logisch von der Struktur getrennt sind. Die Nachteile liegen beim Aufwand und den Inkompatibilitäten, die auftreten können. Darüber hinaus ist das Thema komplex und erfordert das Lesen weiterer Ressourcen (siehe ix.de/zqqa).

„Veraltet“, aber weiter gültig

Während der Entstehung dieses Artikels stellte Microsoft seine Dokumentation um. Das Tier-Modell und ESAE wurde als „veraltet“ deklariert. Die neuen Empfehlungen anstelle von ESAE (siehe ix.de/zqqa) bauen auf der Microsoft Azure Cloud auf und setzen die moderneren Sicherheitsfeatures von Azure und Azure AD ein. Unternehmen, die mindestens einen Teil ihrer Infrastruktur in der Cloud betreiben, sollten sich die Empfehlungen ansehen. Ein guter Start ist neben der Übersichtsseite der „Security rapid modernization plan (RAMP)“ (siehe ix.de/zqqa), der auf eine schnelle Adaptierung ausgelegt ist.

Das Konzept des Tier-Modells und PAW, aber auch des aufwendigen ESAE ist für On-Premises-Umgebungen aber noch nicht veraltet. Microsoft schreibt, dass sie selbst intern aufgrund der hohen Sicherheitsanforderungen für die Bereitstellung von Cloud-Diensten eine ähnliche Architektur weiterhin betreiben. Wer sich für die Konzepte ohne Cloud interessiert, sollte im Internet nach weiteren Ressourcen suchen, da die alten Artikel von Microsoft größtenteils nicht mehr online sind.

Erfahrungsgemäß nutzen die wenigsten Unternehmen Maßnahmen wie PAW, Tier-Model oder ESAE. Werden privilegierte Benutzer bereits bewusst und gezielt eingesetzt, wäre ein erster Schritt als Zwischenlösung die Einführung von Jump Hosts, also speziell gesicherten und überwachten Systemen für Administrationsaufgaben. So müssen sich Administratoren nicht an ihrem Arbeitsgerät als Domänenadministrator anmelden, was aus Sicherheitssicht ja zu vermeiden ist.

Ein Admin verbindet sich erst mit dem Jump Host und von dort aus zu den Servern. Das hat einige Vorteile: Auf den Jump Hosts ist bekannt, welche Programme und Dienste zu welchem Zweck verwendet werden. Daher kann man sie speziell schützen, indem man, um ein paar Beispiele zu nennen, den Internetzugriff entzieht, mit der Application-Whitelisting-Funktion AppLocker nur Programme zulässt, die auch verwendet werden, Logdaten erhebt und auswertet oder den Zugriff zu und von den Jump Hosts auf das Minimum einschränkt. In einem weiteren Schritt sollten dann PAWs eingeführt werden, da erst sie eine richtige Trennung der privilegierten Benutzer von der restlichen Infrastruktur bieten.

Ausführen, was explizit erlaubt ist

Weitere Möglichkeiten zum Verhindern und Erkennen von Angriffen sind der Einsatz von Application Whitelisting, etwa das bereits erwähnte AppLocker oder Windows Defender Application Control (WDAC), auch bekannt unter der inzwischen veralteten Bezeichnung Device Guard, und die Verwendung eines Next-Generation-Antivirus-Produkts (NGAV). Ersteres wird per Gruppenrichtlinie verteilt und verhindert das Ausführen unerwünschter Software. AppLocker lässt sich zwar in den meisten Fällen umgehen, erschwert es einem Angreifer jedoch, Fuß zu fassen, Informationen zu erbeuten und sich auszubreiten.

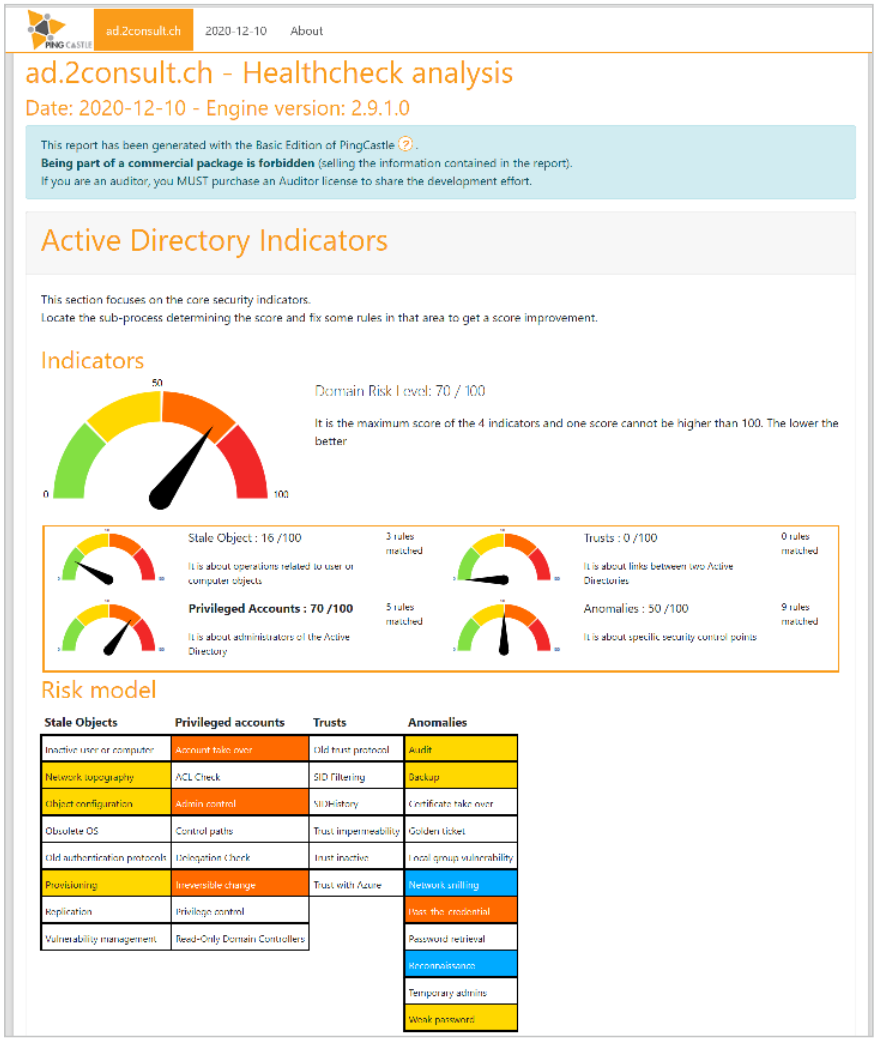
Außerdem schützt es vor normaler Schadsoftware, da diese in der Regel dank strenger Whitelisting-Regeln gar nicht ausgeführt werden darf. Per AppLocker kann man auch die Verwendung von Systemtools unterbinden, solange sie nicht benötigt werden. So können Angreifer nicht auf integrierte Werkzeuge wie PowerShell zurückgreifen. AppLocker ist in der GPO unter Computerkonfiguration\Windows-Einstellungen\Sicherheitseinstellungen\Anwendungssteuerungsrichtlinien\AppLocker zu finden.

Richtig konfiguriert kann ein NGAV ein weiterer Baustein sein, Angriffe abzuwehren oder zu erkennen. Man sollte sich zwar nicht allein auf die Schutzsoftware verlassen, im Zusammenspiel mit anderen Maßnahmen kann sie aber ihren Teil zur Sicherheit beitragen. Erwähnenswert ist zum Beispiel das Antimalware Scan Interface (AMSI) von Microsoft, das es installierter Software erlaubt, mit dem Malwarescanner zu interagieren. So kann das Antivirusprogramm per AMSI unter anderem PowerShell-Befehle prüfen und gegebenenfalls blockieren.

AMSI hilft somit bei der Lösung des Problems, dass Malwarescanner oft nur Skripte erkennen, die auf die Festplatte geschrieben werden, nicht diejenigen, die direkt aus dem Speicher heraus ausgeführt werden. Wird Microsoft Defender eingesetzt, empfiehlt sich darüber hinaus ein Blick auf die zusätzlichen Härtungsmaßnahmen über die „Attack Surface Reduction (ASR)“-Regeln (siehe ix.de/zqqa).

Authentifizierung über gesicherten Tunnel

Kerberos kann ebenfalls gehärtet werden, sodass es schwieriger wird, die damit verbundene Authentisierung anzugreifen. Wird Kerberos Armoring genutzt, die Windows-Implementierung von FAST (Flexible Authentication Secure Tunneling), kann



Ein Beispielreport von PingCastle: Das Tool prüft gängige Risikofaktoren und liefert zahlreiche Hinweise zur Absicherung des Active Directory (Abb. 4).

das Attacken wie Kerberoasting [5] verhindern. Gleichzeitig sollten Benutzer wie in [5] gezeigt die Kerberos-Präauthentifizierung eingeschaltet haben. Wie Kerberos Armoring funktioniert und eingerichtet wird, beschreiben die Referenzen im Internet (siehe ix.de/zqqa).

Weitere gute Quellen zur Absicherung von Active Directory sind der entsprechende Baustein im IT-Grundschutz des BSI sowie die „Active Directory Security Assessment Checklist" des CERT-FR. Die Seite „Active Directory Security" von Sean Metcalf befasst sich immer wieder mit neuen Themen rund um die Sicherheit von Active Directory und bietet viele Links und Informationen zu Angriff und Verteidigung. Ebenfalls einen Blick wert sind alle Seiten unter den „Best Practices for Securing Active Directory" von Microsoft (alle genannten Quellen siehe ix.de/zqqa).

Alles, was bisher zur Prävention beschrieben wurde, verdeutlicht, dass die Maßnahmen am besten in Kombination funktionieren. Nur ein Fehler genügt dem Angreifer, die Lücke zu finden und auszunutzen. Für den Verteidiger bedeutet das einen Mehraufwand gegenüber dem Hacker. Die beste Verteidigung enthüllt daher ein Angriff, da man so ebenfalls die Lücken sucht und die Sicht eines Angreifers einnimmt.

Selbst gehackt ist halb verteidigt

Um sich mit den folgenden Werkzeugen auseinanderzusetzen, sollte eine dedizierte virtuelle Maschine oder ein dediziertes Gerät verwendet werden. Es muss in die Domäne aufgenommen und die Programme müssen in der Regel mit einem Domänenadministrator ausgeführt werden. Ein Domänenadmin ist dazu nicht nötig und es ist nicht empfehlenswert, für das Starten der Programme einen privilegierten Domänenbenutzer zu verwenden. Für die Auswertung mit BloodHound wird zusätzlich ein Kali Linux mit Desktop benötigt. Die Installation wird weiter unten beschrieben.

Ein einfach zu verwendendes Werkzeug, das gut interpretierbare Ratschläge zur weiteren Absicherung des AD ausgibt, ist PingCastle. Es kann von der Website (siehe ix.de/zqqa) heruntergeladen, entpackt und per Doppelklick ausgeführt werden. Nach erfolgreichem Sammeln von Informationen gibt das Prüftool einen Bericht in HTML-Form im gleichen Verzeichnis aus, aus dem das Programm gestartet wurde. Der Bericht liefert einen Zahlenwert sowie einen Überblick über die Funde, zu denen es im Einzelnen noch weitere Informationen gibt (siehe Abbildung 4). Zu jedem Eintrag wird beschrieben, um was es geht, was das technisch bedeutet und was man dagegen tun kann, außerdem findet man weiterführende Links. PingCastle ist ein wertvolles Tool, das für die eigene Verwendung gratis zur Verfügung steht. Für Einsätze bei Kunden oder andere Szenarien sollte man sich mit der Lizenzierung vertraut machen.

Ein weiteres Werkzeug ist ADExplorer von Sysinternals (siehe ix.de/zqqa). Der ADExplorer stellt eine Verbindung zu einem Active Directory her und zeigt die Objekte und deren Attribute an. Viele, die das zum ersten Mal ausprobieren, sind erstaunt, wie viele Informationen ein normaler Domänenbenutzer aus dem Active Directory auslesen kann [3]. Fragen wie „Wann hat sich der Chef zum letzten Mal eingeloggt, wie oft und wann hat der Admin sein Passwort falsch eingegeben, welche OUs gibt es oder wer gehört einer bestimmten Gruppe an?" können alle beantwortet werden. Der obige Abschnitt zu Least Privileges beschreibt, wie man den Lesezugriff einschränken kann.

Bei Angreifern sehr beliebt ist das Werkzeug BloodHound [4]. Es liest Daten aus dem Active Directory aus und verbindet sich nach Möglichkeit kurz mit weiteren Systemen im Netzwerk, um an zusätzliche Daten zu kommen. Das PowerShell-Skript oder C#-Programm, mit dem man die Daten sammelt, heißt SharpHound. Die erhobenen Daten werden danach in BloodHound importiert, das in der Regel auf einem Linux-System läuft. Hinter BloodHound werkelt eine Neo4j-Datenbank. Das Tool stellt die Daten in Graphen dar, die sich durchsuchen lassen. Man kann einen vorgegebenen Suchvorschlag ausführen, etwa „Suche den kürzesten Weg zum Domänenadmin", oder eigene Suchanfragen definieren.

Die Hundemeute nimmt Witterung auf

Um BloodHound zu verwenden, muss als Erstes das SharpHound Skript heruntergeladen werden. Da das Skript von Malwarescannern erkannt wird, sollte man es nur in der dedizierten VM herunterladen und in den Malwarescanner-Einstellungen als Ausnahme einrichten. Es kann von GitHub heruntergeladen werden (siehe ix.de/zqqa) und wird danach per PowerShell ausgeführt. Dabei passiert noch nicht viel, denn erst jetzt kann der eigentliche Befehl ausgeführt werden: Invoke-BloodHound. Das Scriptlet sammelt die Informationen und speichert sie in dem Ordner als ZIP-Archiv, in dessen Kontext sich die PowerShell gerade befindet.

Um die Daten mit BloodHound auswerten zu können, muss zuerst ein Kali Linux inklusive BloodHound installiert werden. ISO-Dateien oder fertige virtuelle Maschinen (VMs) zum Herunterladen sind auf der Kali-Website verfügbar (siehe ix.de/zqqa). Benutzername sowie Passwort lauten standardmäßig „kali". Nach erfolgreichem Starten einer Standardinstallation muss BloodHound aufgespielt werden. Dies geschieht über folgende Befehle in der Shell:

```
sudo apt-get update && apt-get install ↗
                                  bloodhound
```

Um BloodHound zu starten, muss zuerst die Datenbank Neo4j gestartet werden. Zuvor ist ein Verzeichnis für die Logs anzulegen:

```
sudo mkdir /usr/share/neo4j/logs
sudo neo4j start
```

Anschließend muss die Benutzerin das Passwort für Neo4J ändern. Dies wird im Browser unter der URL http://localhost:7474/browser/ erledigt. Standardbenutzername sowie -passwort lauten „neo4j". Nach dem Ändern des Passwortes kann der Browser wieder geschlossen und BloodHound endlich gestartet werden. Dazu in der Shell `bloodhound` eintippen und Enter drücken. Für das Log-in an der Neo4J-Datenbank werden der Benutzer „neo4j" und das neu erstellte Passwort benötigt.

Ist BloodHound installiert und gestartet, kann man rechts mit dem Symbol „Upload Data" das von SharpHound erstellte ZIP auswählen. Nach dem Import können die Objekte durchsucht und angezeigt werden. Über das Hamburger-Menü gelangt man in den Tab „Analysis". Dort werden vorgegebene Suchaufträge gelistet. Interessante Suchanfragen sind zum Beispiel „Find Shortest Paths to Domain Admins" oder „Shortest Paths to High Value Targets". Man kann auch nach bekannten schützenswerten Objekten und dann per Rechtsklick nach dem kürzesten Pfad zu dem entsprechenden Objekt suchen. So wird gezeigt, wo es noch Lücken, zu viele Berechtigungen oder ungeahnte Pfade zu Objekten über unbekannte Verbindungen gibt, die beseitigt oder verhindert werden müssen.

Doch aufgepasst: Nur weil es eine Verbindung über mehrere Objekte zu einem Ziel gibt, heißt das nicht, dass der Weg auch ausgenutzt werden kann. Die verschiedenen Verbindungen müssen interpretiert und eingeordnet werden können.

Ohne richtiges Wissen über Windows und Active Directory wird BloodHound keine oder nur vermeintliche Geheimnisse preisgeben. Richtig eingesetzt kann es aber den Verteidiger dazu befähigen, selbst nach dem Angriffsweg zu suchen, bevor es jemand anderes tut.

Ein Projekt namens PlumHound geht noch einen Schritt weiter und macht die Daten aus BloodHound für Verteidiger besser zugänglich. Mit dem Werkzeug lassen sich Berichte oder Task-Listen auf Basis der BloodHound-Daten erstellen. Das für Security-Teams konzipierte Werkzeug weiter zu beschreiben würde den Rahmen dieses Artikels sprengen, detaillierte Informationen dazu liefert aber die Projektseite auf GitHub (siehe ix.de/zqqa).

Auf Nummer sicher gehen

Trotz aller genannten Maßnahmen kann es immer passieren, dass ein Angreifer eine Lücke findet – vor allem dann, wenn er nicht allein agiert, genug Ressourcen zur Verfügung hat und einem Unternehmen gezielt Schaden zufügen möchte. Auch ein Angriff von innen ist nicht auszuschließen, wenn ein Admin mit Wut im Bauch die Malware direkt auf dem Domänencontroller platziert, Daten löscht oder GPOs böswillig verändert hinterlässt. Es kann letztlich immer zu einem Sicherheitsvorfall kommen. Maßnahmen für den Worst Case vorzubereiten, ist daher sinnvoll und zwingend notwendig, denn es kann im schlimmsten Fall um das Überleben der Firma gehen.

Das Backup sollte funktionieren und eine gewisse Resilienz aufweisen, damit es Löschversuchen von Angreifern und intelligenter Schadsoftware standhalten kann. Grundlegend für Backup sind folgende Punkte: Wie auch in anderen Zusammenhängen sollte beim Backup das Principle of Least Privileges, also das Prinzip der geringsten Privilegien, gelten. Konkret sollen nur Benutzer mit der entsprechenden Rolle Backup-Daten löschen oder verändern dürfen. Die Backup-Agents sollten ebenfalls nur die nötigsten Berechtigungen erhalten. Sind Systeme, die Backup-Daten speichern, nicht in der Domäne, können Angreifer trotz erlangter Domänenadmin-Berechtigungen das Backup unter Umständen nicht löschen.

Ein Medienbruch – in diesem Rahmen ist das Offlinenehmen eines Backup-Mediums gemeint – sollte in regelmäßigen Abständen stattfinden. Sollte alles andere vernichtet werden, bleibt nur noch dieses Backup übrig [1]. Daher sollte der Zeitabstand so gewählt werden, dass das Unternehmen auch im schlimmsten Fall überleben kann. Des Weiteren wollen Backups getestet werden. Manch ein Admin kam schon ins Schwitzen, weil die Sicherungskopien nicht richtig erstellt wurden oder die Wiederherstellung scheiterte. Ein guter Nebeneffekt solcher Tests: Man verschafft sich Know-how. Außerdem schützt die räumliche Trennung zweier oder mehrerer Backup-Kopien vor dem Ausfall eines Speichergerätes oder eines kompletten Standorts.

Wie man Backups erstellt und Systeme und Daten wiederherstellt, sollte nicht nur eine Person im Unternehmen wissen. Im besten Fall ist alles sauber dokumentiert und auch offline verfügbar. Durch das regelmäßige Testen der Backups inklusive Recovery kann Wissen auf mehrere Personen verteilt werden, wenn man es richtig organisiert.

Aufstehen, Krönchen richten …

Business Continuity, das Weiterführen des Geschäfts auch bei Sicherheitsvorfällen (Incidents), obliegt dem Management. Unternehmer sollten sich Gedanken dazu machen, wie man auch in widrigen Situationen das Tagesgeschäft aufrechterhält oder in angemessener Frist wiederherstellen kann. Für die IT bedeutet das, Systeme und Services gemäß sinnvoller Reihenfolge und innerhalb einer bestimmten Zeit wiederherstellen zu können. Dafür muss der entsprechende Notfallplan jedoch schon ausgearbeitet und verfügbar sein.

Eine gute Strategie zum Erstellen eines solchen Notfallplans können zu Beginn Tabletop-Exercises sein, bei denen man verschiedene Szenarien definiert und deren Ablauf mit den richtigen Personen zusammen bespricht und durchspielt. Wichtig ist die Verfügbarkeit eines solchen Plans und der relevanten Daten auch im Worst Case. Dazu gehören Passwörter, Netzpläne, wichtige Teile der Dokumentation, Schlüssel zum Entschlüsseln der Backups, Zugänge zu Räumlichkeiten, Zugang zum Backup, Verzeichnis wichtiger Rufnummern und dergleichen mehr. Die genannten Punkte sollen dazu anregen, sich weiter mit dem Thema auseinanderzusetzen und das Unternehmen auch auf Krisen vorzubereiten.

Fazit

Härten ist nicht trivial und setzt Wissen und Zeit voraus. Viele der Maßnahmen wirken erst im Zusammenspiel mit anderen und nur bei korrekter Konfiguration. Am besten wird Sicherheit umgesetzt, wenn das Thema bereits unternehmensweit in den Prozessen seinen Platz hat und nicht erst im Nachhinein „übergestülpt“ werden muss. Der Schutz des AD bedeutet eigentlich den Schutz der gesamten Infrastruktur. Es reicht nicht, nur den Domänencontroller zu härten. Und doch ist er das Herz der Domäne, von dem aus alles gesteuert werden kann. Hat ein Angreifer das Active Directory beziehungsweise den Domänencontroller unterwandert, müssen die Systemverantwortlichen alles wiederherstellen oder neu aufbauen. Der Schutz des Active Directory und seiner privilegierten Benutzer sollte daher jedem am Herzen liegen. (ur@ix.de)

Quellen

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

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

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

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

[5] Frank Ully; Frisch geröstet; Roasting, Rechte, Richtlinien: Wie Angreifer sich im Active Directory Zugriff verschaffen; *iX* 12/2020, S. 92

[6] Frank Ully; Vertrauensfrage; Active Directory: Wie Angreifer Tickets, Delegierung und Trusts missbrauchen; *iX* 2/2021, S. 116

[7] Yves Kraft, Frank Ully; Zwischen den Wäldern; Inter-Forest und Persistenz: Wie Angreifer sich über einen AD-Forest hinaus ausbreiten und festsetzen; *iX* 4/2021, S. 102

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

[9] Alle im Artikel angesprochenen Werkzeuge, Blogbeiträge, technischen Dokumentationen et cetera sind über ix.de/zqqa zu finden.

Marco Wohler

ist System Engineer bei der Oneconsult AG. Als Gegenpart zu den Penetration-Testing-Teams ist er spezialisiert auf die Absicherung von IT-Systemen.