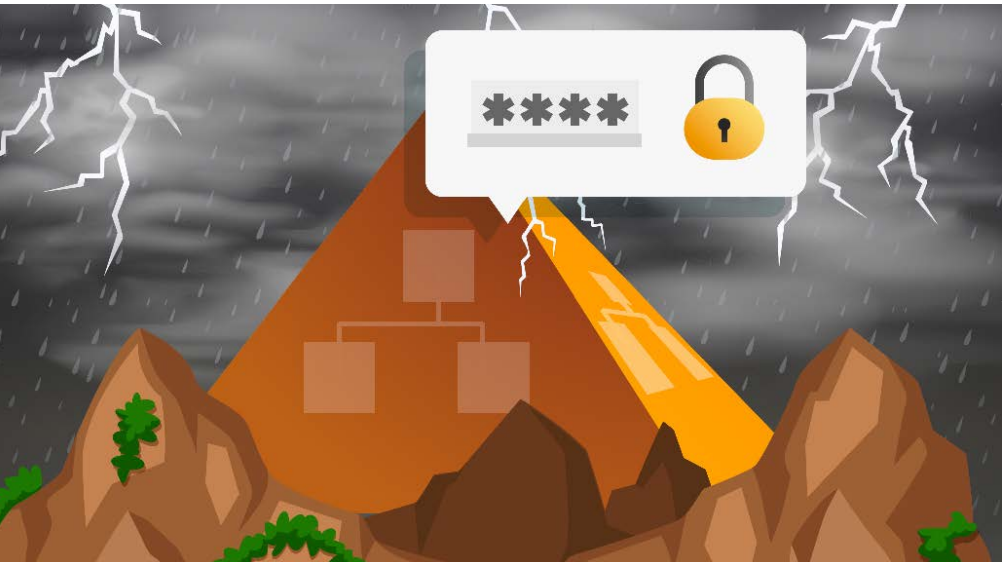


Passwortsicherheit (nicht nur) im Active Directory

Schwierige Wahl

Sandro Affentranger

Schwache Passwörter können von Angreifern leicht erraten und missbraucht werden. Damit können sie Aktionen ausführen, die von denen eines legitimen Benutzers kaum mehr zu unterscheiden sind.

Wie Administratoren ein Active Directory (AD) absichern können, beschreiben bereits zwei Artikel zur AD-Härtung [1, 2]. Darüber hinaus sollte man sich mit der Sicherheit von Passwörtern beschäftigen, denn unter Umständen können sie ein Sicherheitsrisiko darstellen. Selbst mit einer strengen Passwortrichtlinie lässt sich kaum verhindern, dass Benutzer Passwörter wählen, die ein Angreifer leicht erraten kann [3]. Sobald dieser gültige Anmeldedaten in die Finger bekommt, kann er Schutzmaßnahmen umgehen, da er nicht von einem legitimen Benutzer unterscheidbar ist.

Wie gravierend die Kompromittierung eines Passwortes ist, hängt sehr vom Kontotyp ab. Bei einem Account für ein Onlineforum kann der Angreifer sich lediglich als Benutzer ausgeben und in dessen Namen Beiträge posten – der mögliche Schaden hält sich meistens in Grenzen. Handelt es sich jedoch um ein Firmenkonto, erhält der Angreifer unter Umständen Zugang zum internen Firmennetz und kann von dort weitere Attacken gegen das Unternehmen starten. Oftmals existieren innerhalb eines Firmennetzwerks weniger Sicherheitsmechanismen, um solche Angriffe zu erkennen, da die Administratoren davon ausgehen, dass es für einen Angreifer nicht möglich ist, sich Zutritt zum internen Netz zu verschaffen.

Die Passwortrichtlinien, die man heute in Unternehmen antrifft, ähneln sich häufig, denn sie beruhen noch auf den veralteten Empfehlungen des Nationalen Instituts für Standards und Technologie (NIST) aus dem Jahr 2003. Diese fordern eine Mindestlänge von 8 Zeichen. Ziffern und/oder Sonderzeichen müssen verwendet und Passwörter regelmäßig geändert werden.

Benutzer wählen meist kein zufälliges Passwort, weil sie es sich nur schwer merken können – was zu bestimmten Mustern führt, die den Angreifern wiederum bekannt sind und ihnen beim Erraten helfen.

iX-TRACT

- Selbst rigide Richtlinien ermöglichen Passwörter, die ein Angreifer einfach erraten kann.
- Ein Audit gibt Aufschluss darüber, wie es um die Sicherheit der Passwörter im Active Directory steht.
- Im Ergebnis des Audits kann es nötig werden, die Passwortrichtlinie zu ändern oder Multi-Faktor-Authentifizierung einzuführen.
- Die Resultate können auch dazu dienen, Benutzer im Umgang mit Passwörtern zu schulen.

Lieber lange als komplexe Passwörter

Das NIST hat daher im Jahr 2017 seine Empfehlungen aktualisiert. Jetzt rät das Institut von Komplexitätsregeln ab, da diese die Stärke von Passwörtern nicht so sehr erhöhen, wie es zu erwarten wäre. Wenn Benutzer einen Großbuchstaben in ihrem Passwort verwenden müssen, nehmen sie meistens den ersten. Ziffern und Sonderzeichen hängen sie oft ans Ende des Passworts. Statt der Empfehlung, Passwörter möglichst komplex zu gestalten, lautet die neue Devise deshalb: lang statt komplex!

Auch die Anforderung, Passwörter regelmäßig zu ändern, hat selten den erwünschten Effekt. Damit sich Nutzer nicht ständig ein neues Passwort merken müssen, verändern sie es nach bestimmten Mustern, etwa durch Ergänzen einer Ziffer (Passwort1, Passwort2, Passwort3 und so weiter). Kennt man also das erste Passwort, kann man auch die nachfolgenden erraten. Seit 2017 empfiehlt das NIST daher keine regelmäßigen Passwortwechsel mehr. Etwas später, im Jahr 2020, verabschiedete sich auch das Bundesamt für Sicherheit in der Informationstechnik in seinem IT-Grundschutz-Kompendium davon.

Die Sicherheitsfirma SplashData veröffentlicht seit 2011 jedes Jahr eine Liste mit den beliebtesten Passwörtern, basierend auf geleakten Log-in-Informationen. 2019 wurden über fünf Millionen Passwörter zusammengetragen und analysiert (siehe ix.de/zr2y), hauptsächlich von nordamerikanischen und westeuropäischen Benutzern. Die 10 beliebtesten Passwörter 2019 waren:

1. 123456
2. 123456789
3. qwerty
4. password
5. 1234567
6. 12345678
7. 12345
8. iloveyou
9. 111111
10. 123123

Um zu verhindern, dass solche schwachen Passwörter verwendet werden, kann man eine Richtlinie definieren, die neben der Mindestlänge festlegt, dass Passwörter nicht nur aus Zahlen oder nur aus Buchstaben bestehen dürfen.

Risikoabschätzung durch Audit

Wie kann man das Risiko einer Accountkompromittierung aufgrund schwacher Passwörter im Active Directory genauer abschätzen? Eine Möglichkeit ist ein Passwortaudit. Dessen Ziel ist es, einen Überblick über die Qualität der Passwörter zu erhalten und damit eine Einschätzung, wie leicht oder schwer sie zu erraten sind. Außerdem lassen sich mit den Erkenntnissen des Audits Änderungen an der Richtlinie oder die Einführung von Multi-Faktor-Authentifizierung [4] begründen.

Interessant dabei sind die Unterschiede zwischen Passwörtern normaler Benutzer und solchen mit administrativen Rechten. Benutzerkonten mit Administratorrechten sind ein lukrativeres Ziel für einen Angreifer und daher ist es besonders wichtig, dass sie mit starken Passwörtern gesichert sind. Doch oft wird gerade für Administrator- und Systemdienstkonten die Passwortrichtlinie nicht durchgesetzt – ihre Passwörter genügen dann unter Umständen nicht einmal den Mindestanforderungen oder sie werden nach einer Verschärfung der Passwortrichtlinie nicht überprüft und angepasst [5].

Vorbereitung für ein Passwortaudit

Ein Passwortaudit sollte stets mit der Geschäftsleitung abgesprochen und schriftlich festgehalten werden. Das genaue Vorgehen und die erwünschten Ziele sind im Vorfeld zu dokumentieren. Ebenso muss sichergestellt werden, dass Passwörter und Passworthashes gut geschützt werden. Benutzernamen und Passwörter sollten getrennt aufbewahrt und nach Ende des Audits gelöscht werden. Nach Ende des Audits sollten die Mitarbeiter aufgefordert werden, ihre Passwörter zu wechseln.

Ein solches Audit zeigt jedoch nicht unbedingt einen realen Angriffsvektor. Ein Angreifer ist meist nur in der Lage, wenige Hashes zu extrahieren. Und um alle Hashes aus einem Domänencontroller zu extrahieren, brauchte er Administratorrechte. Sollte ein Angreifer diese besitzen, hat er weitaus bessere Optionen: Denn als Domänenadmin hat er ohnehin die volle Kontrolle über das Active Directory und administrativen Zugriff auf alle Geräte in der Domäne. Aber selbst wenn er nur ein paar Hashes gestohlen hat, reicht schon ein einziges geknacktes Passwort aus, um den Fuß im Firmennetzwerk zu haben.

Zerhackt, nicht im Klartext

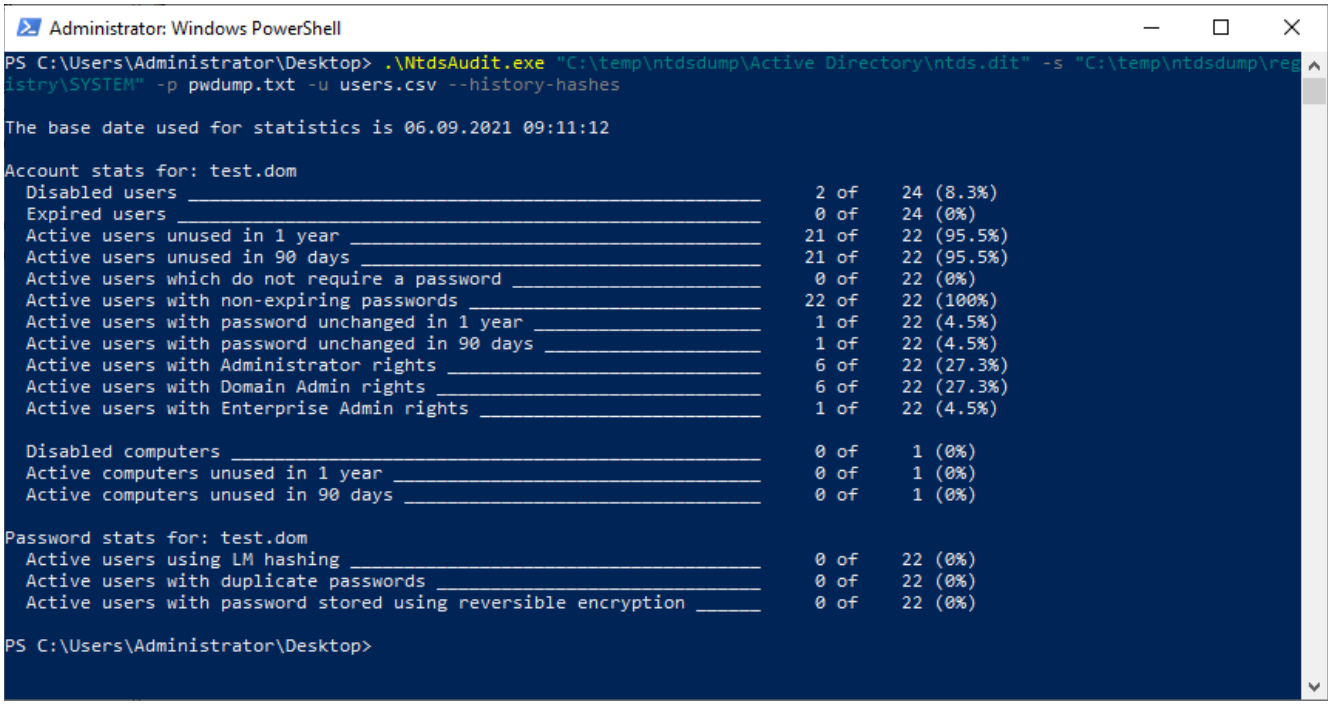
Da eine direkte Analyse der Passwörter nicht möglich ist, muss eine gewisse Vorarbeit geleistet werden. Passwörter werden in der Regel nicht im Klartext gespeichert. Das soll verhindern, dass im Falle der Kompromittierung eines Systems alle Passwörter sichtbar sind, die auf dem System verwaltet werden. Stattdessen werden sie gehasht und nur der Hash wird gespeichert. Daher ist es bei einem Passwortaudit notwendig, zuerst die Passworthashes zu knacken – je mehr, desto besser. Denn je mehr Passwörter im Klartext vorliegen, desto mehr Schlüsse kann man daraus ziehen.

Listing 1: Erstellen eines Snapshots von ntds.dit

```
ntdsutil
  activate instance ntds
     ifm
     "create full C:\Temp\ntds.dit"
  quit
quit
```

Es gibt eine Vielzahl verschiedener Hashfunktionen, aber alle haben zwei wichtige Eigenschaften. Erstens: Der Hash ist immer gleich lang, unabhängig von der Länge des Passworts. Passwort a erzeugt beispielsweise den Hash 186CB09181E2C2ECAAC768C47C729904, das Passwort abcdefghijklmnopqrstuvwxyz den Hash 0BD63185F3484BB000286C85917DC12E.

Zweitens: Auch wenn nur ein Zeichen anders ist, ergibt sich ein völlig anderer Hash. Passwort1 führt zu 045CA02E7001395CF66E002773C2C5C7, Passwort2 zu 12C1D190D4CFEB40F257A357E870B2CF. Beides sind wichtige Eigenschaften – daraus folgt nämlich, dass



Eine Extraktion der Passworthashes mit NtdsAudit liefert bereits erste Erkenntnisse (Abb. 1).

man basierend auf einem Hash nicht auf die Länge des Passwortes schließen kann. Und beim Knacken eines Hashes kann man nie sagen, wie nah man am eigentlichen Passwort ist. Beide Eigenschaften erschweren das Hashcracking.

Hashes extrahieren

Passworthashes aus dem Active Directory eignen sich besonders gut für ein Passwortaudit. Einerseits kann man sie recht einfach aus dem Domänencontroller extrahieren und hat dann die Hashes der gesamten Mitarbeiter. Andererseits sind diese Hashes per Design nicht gut gegen Offlineattacken geschützt; dadurch lassen sie sich einfacher knacken als andere. Das wiederum führt dazu, dass mehr Passwörter geknackt werden können und so mehr Daten für eine Analyse zur Verfügung stehen.

Zum Extrahieren der Hashes aus dem Active Directory kann das bereits vorinstallierte AD-Verwaltungstool ntdsutil (siehe ix.de/zr2y) verwendet werden. Damit lässt sich ein Snapshot der vorhandenen ntds.dit-Datei erstellen und für die Offlineanalyse und das Extrahieren von Passworthashes an einen neuen Speicherort kopieren. Die Hashes liegen dann im NTLM-Format vor (kurz für NT LAN Manager), auch bekannt als NT-Hash. Es ist das Format, in dem Windows Passworthashes speichert, wie in Listing 1 gezeigt.

Auf Windows-Systemen können die Hashes dann mit dem Tool NtdsAudit (siehe ix.de/zr2y) extrahiert werden. Mit dem Parameter `--history-hashes` lässt sich steuern, ob auch die vorherigen Passwörter ausgelesen werden sollen.

```
.\NtdsAudit.exe "C:\temp\ntds.dit" -p ↵
                        pwdump.txt -u users.csv
```

NtdsAudit liefert bereits erste Statistiken zu den Benutzern und deren Passwörtern. In Abbildung 1 lässt sich beispielsweise sehen, wie viele Domänenadmins es gibt oder ob Passwörter mehrmals vorkommen.

Auf Linux-Systemen können die Hashes mit secretsdump.py (siehe ix.de/zr2y) aus der Impacket-Toolsammlung extrahiert werden, die alten Passwörter liefert `-history`.

```
secretsdump.py -ntds ntds.dit -outputfile ↵
                                    ntds.out LOCAL
```

Passwörter cracken

Für Passwortcracking kann man beispielsweise Hashcat (siehe ix.de/zr2y) einsetzen, ein frei verfügbares Open-Source-Tool zur Passwortwiederherstellung. Es nutzt die Prozessoren von Grafikkarten, die Hashes schneller als normale Computerprozessoren berechnen. Selbst auf älteren Grafikkarten lassen sich damit mehrere Milliarden NTLM-Hashes pro Sekunde ermitteln.

Für das Knacken von Passworthashes gibt es unterschiedliche Vorgehensweisen. Einerseits kann man alle möglichen Kombinationen durchprobieren: aaaa aaaa, aaaaaaab, aaaaaaac und so weiter. Diese Brute Force genannte Methode ist jedoch nicht sonderlich effizient und der Aufwand steigt für längere Passwörter exponentiell.

Je nach Hardware lassen sich mit dieser Methode Angriffe auf Passwörter bis zu einer Länge von acht Zeichen in wenigen Tagen durchführen. Dabei werden lediglich druckbare ASCII-Zeichen berücksichtigt, also alle Groß- und Kleinbuchstaben, Ziffern sowie Sonderzeichen inklusive Umlauten. Mit jedem zusätzlichen Zeichen nimmt die benötigte Rechenzeit exponentiell zu.

Das Cloud-Computing eröffnete auch für das Cracking von Hashes neue Möglichkeiten: Man kann sie verteilt auf mehreren Systemen knacken lassen. Mit dem Tool NPK (siehe ix.de/zr2y) kann man solche Attacken mit Amazon Web Services (AWS) durchführen. Mit AWS wäre es möglich, alle neun Zeichen langen Passwörter in weniger als einer Woche durchzuprobieren. Die Kosten für einen solchen Brute-Force-Angriff würden sich auf ein paar Tausend Euro belaufen, was sich je nach erwartetem finanziellen Gewinn lohnen kann.

Mit dem folgenden Hashcat-Befehl lässt sich ein Brute-Force-Angriff durchführen:

```
hashcat -m1000 hashes.txt -a3 --increment ↵
        --increment-min 1 --increment-max 8 ↵
                                ?a?a?a?a?a?a?a?a
```

Der Parameter `-m` gibt den Hashtyp an, in diesem Fall 1000 für NTLM. Der Parameter `-a` gibt den Angriffsmodus an, der Wert 3 steht für einen Maskenangriff. Maskenangriffe sind eine Unterkategorie der Brute-Force-Attacken, jedoch spezifischer und daher schneller: Statt an jeder Position alle möglichen Zeichen durchzuprobieren, werden nur gewisse Zeichen ausprobiert; dadurch ergeben sich weniger Kombinationen, die berechnet werden.

Listing 2: Verschiedene Masken

```
?l = abcdefghijklmnopqrstuvwxyz
?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ
?d = 0123456789
?s = "space"!"#$%&'()*+,-./↵
                                :;<=>?@[\]^_`{|}~
?a = ?l?u?d?s
?b = 0x00 - 0xff
```

Der letzte Teil des Befehls sagt Hashcat, welche Maske verwendet werden soll. Bei diesem Brute-Force-Angriff werden für alle acht Stellen des Passworts sämtliche Groß- und Kleinbuchstaben, Zahlen sowie Sonderzeichen durchprobiert.

Für längere Passwörter lohnt es sich, genauere Masken zu definieren (siehe Listing 2). Beispielsweise „?u?l?l?l?l?l?l?l?d?s“ für 10-stellige Passwörter, die mit einem Großbuchstaben beginnen, gefolgt von 7 Kleinbuchstaben mit einer Zahl und einem Sonderzeichen am Ende. Damit könnte man zum Beispiel „Passwort1!“ knacken.

Wörterbuchangriff inklusive Varianten

Es ist allerdings meist nicht sinnvoll, alle möglichen Passwörter durchzugehen. Auch Maskenangriffe werden mit zunehmender Länge schnell ineffizient. Ziel sollte stattdessen sein, möglichst nur diejenigen Passwörter zu probieren, die Benutzer am wahrscheinlichsten wählen. Bei einer Wörterbuchattacke wird genau dies getan: Das entsprechende Werkzeug geht eine Liste von Wörtern durch, die meistens noch mit gewissen Regeln verändert werden, damit aus einem einzelnen Wort mehrere Kandidaten entstehen.

Mögliche Regeln wären beispielsweise „Schreibe den ersten Buchstaben groß“ oder „Ersetze den Buchstaben e durch die Ziffer 3 “. Aus „passwort“ wird so zum Beispiel „PASSWORT“, „Passwort“, „passwort1“, „p@sswort“ usw. Als Eingabewörterbuch kommen mehrere Möglichkeiten infrage, beispielsweise die häufigsten Wörter im Deutschen oder Englischen, Listen mit Namen von Personen oder mit geleakten Passwörtern.

Bei einem Audit der Passwörter eines Active Directory lohnt es sich, eine auf das Unternehmen zugeschnittene, spezifische Wörterliste zusammenzustellen. Dafür kann man etwa die Website des Unternehmens durchsuchen und alle dortigen Wörter in die Liste aufnehmen. Weitere Informationen, zum Beispiel die Standorte aller Niederlassungen, können ebenfalls in die Liste eingetragen werden.

Abbildung 2 zeigt einen Wörterbuchangriff, bei dem die RockYou-Passwortliste – eine bekannte Liste mit über 14 Millionen geleakten Passwörtern – zum Einsatz kommt. Kombiniert wird sie mit dem Regelset dive, das mit Hashcat mitgeliefert

Ein Wörterbuchangriff mit Hashcat knackt bereits erste Passwörter (Abb. 2).

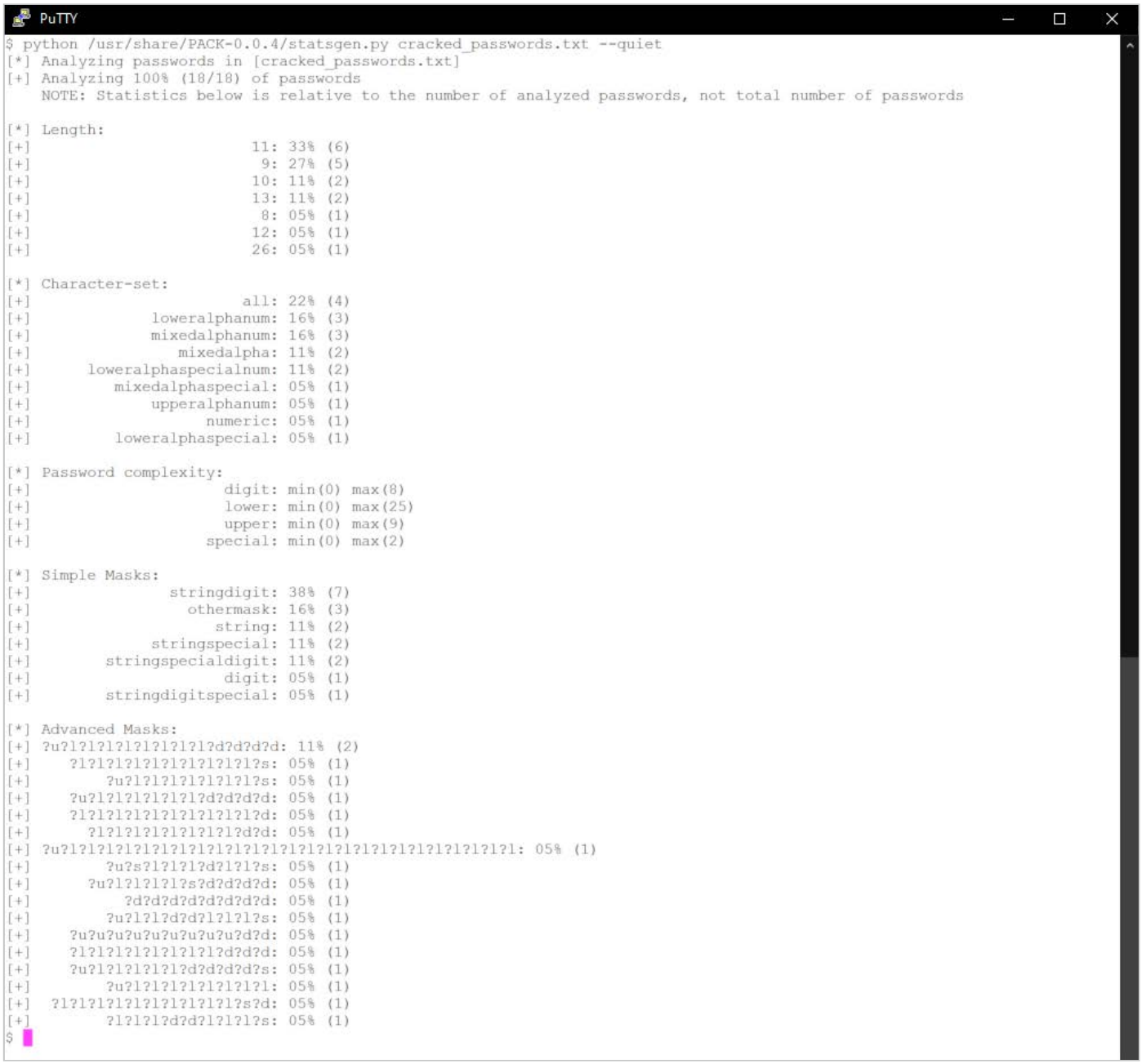
wird und fast 100 000 Regeln enthält. Der Angriff dauert nur wenige Minuten und findet bereits erste Passwörter.

Sobald die ersten Passwörter gecrackt wurden, lohnt es sich, diese zu analysieren. So kann man gezieltere Angriffe auf Passwörter mit demselben Muster starten. Insbesondere wenn nicht nur die aktuellen, sondern auch vorhergehende Passwörter aus dem Active Directory extrahiert wurden, lassen sich die derzeit genutzten Passwörter auf der Basis ihrer Vorgänger knacken. Wenn beispielsweise ein Benutzer bei jedem Passwortwechsel einfach ein weiteres Zeichen anhängt, kann das mit einem Wörterbuchangriff erkannt werden.

Zusätzlich zu den genannten Methoden gibt es zahlreiche weitere Vorgehensweisen und Werkzeuge. Grenzen beim Herausfinden der Passwörter setzen die verfügbare Rechenleistung und die zur Verfügung stehende Zeit.

Have I Been Pwned?

Der australische Sicherheitsexperte Troy Hunt betreibt die Dienste „Have I Been Pwned“ und „Pwned Passwords“ (beide siehe ix.de/zr2y). Mit „Have I Been Pwned“ kann man überprüfen, ob man von bekannten Datenlecks betroffen ist. Wenn man dort die eigene E-Mail-Adresse eingibt, werden alle Datenlecks aufgelistet, in denen diese Adresse vorkommt. Außerdem



Die Analyse der gecrackten Passwörter mit PACK liefert unter anderem Erkenntnisse zu ihrer Länge und Komplexität (Abb. 3).

kann man sich registrieren, um eine Benachrichtigung zu erhalten, sobald ein neues Datenleck mit der E-Mail-Adresse öffentlich gemacht wurde. „Pwned Passwords" hingegen ist, wie der Name schon vermuten lässt, eine Sammlung von bekannten kompromittierten Passwörtern. Man kann sein Passwort eingeben und erfährt dann nicht nur, ob es in Datenlecks vorgekommen ist, sondern auch wie oft.

Allerdings sollte man sehr vorsichtig sein, wenn es darum geht, ein Passwort auf der Website eines Drittanbieters einzugeben. „Pwned Passwords" implementiert ein k-Anonymitätsmodell (siehe ix.de/zr2y), um zu verhindern, dass ein Passwort durch einen solchen Check selbst bekannt wird. Beim Prüfen wird daher das Passwort zuerst gehasht und nur die ersten fünf Zeichen des Hashes werden an den Server geschickt. Dieser antwortet dann mit allen Hashes, die mit der mitgelieferten Teilzeichenkette anfangen. Danach kann lokal im Browser überprüft werden, ob das Passwort in einem Leak vorkam oder nicht. Der Server weiß also nicht, welches Passwort überprüft wurde.

Passwörter analysieren

Angreifer nutzen solche Datenlecks ebenfalls. Zum einen werden beim Versuch, einen Account zu kompromittieren, zuerst die am häufigsten auftretenden Passwörter ausprobiert, um vielleicht einen möglichen „Quick Win" zu erlangen. Zum anderen analysieren auch Angreifer geleakte Passwörter, um Muster zu erkennen und ihre Attacken entsprechend zu optimieren.

Noch bevor mit dem Cracken der Passworthashes begonnen wurde, kann man bereits erste Aussagen über die verwendeten Passwörter treffen. Bei NTLM-Hashes handelt es sich um sogenannte „ungesalzene" (unsalted) Hashes. In der Kryptografie versteht man unter „Salz" eine zufällige Zeichenfolge, die der Hashfunktion zusätzlich zum Passwort übergeben wird. So wird für zwei gleiche Passwörter ein unterschiedlicher Hash berechnet. Da

NTLM-Hashes ungesalzen sind, haben gleiche Passwörter einen identischen Hash. Das heißt, man kann die Hashes auf ihre Einzigartigkeit hin untersuchen. So lässt sich schon vor dem Cracken der Passwörter sagen, wie viele Passwörter mehrmals verwendet werden. Es ist anzunehmen, dass es sich dabei um schwache Passwörter handelt.

Nachdem die ersten Passwörter gecrackt wurden, können diese mit PACK (Password Analysis and Cracking Toolkit, siehe ix.de/zr2y) analysiert werden. PACK erstellt eine Liste von Passwortmasken und deren Häufigkeit, mit denen man Maskenangriffe starten kann.

Abbildung 3 zeigt das Ergebnis der mit PACK gecrackten Passwörter. Das Tool liefert Statistiken über die Länge und Komplexität der Passwörter sowie häufige Masken. Im gezeigten Beispiel entspricht die häufigste Maske einer Reihe von Groß- und Kleinbuchstaben, gefolgt von ein oder mehreren Ziffern. Das Beispiel demonstriert auch, dass die Länge allein noch kein starkes Passwort ausmacht. Wenn das Passwort auf einem langen Wort oder einer bekannten Wortfolge basiert, kann es möglicherweise durch einen Wörterbuchangriff herausgefunden werden. Bei einem AD-Passwortaudit ist häufig die Mehrzahl der Passwörter innerhalb kurzer Zeit gecrackt. Daraufhin kann man, je nach Zeit und Ziel des Audits, verschiedene Aspekte untersuchen, naheliegend sind etwa Länge und Komplexität. Außerdem kann untersucht werden, wie viele der Passwörter mehr als nur einmal verwendet werden – oder wie viele der Passwörter in bekannten Datenlecks vorkommen.

Falls die Passwortrichtlinie vorschreibt, dass Passwörter regelmäßig geändert werden müssen und die vorhergehenden nicht wiederverwendet werden dürfen, existiert für jeden Benutzer ein Passwortverlauf, der ebenfalls von einem Domänencontroller extrahiert und analysiert werden kann. Beim Betrachten von Passwortverläufen lässt sich untersuchen, ob Benutzer jeweils ein völlig neues Passwort wählen oder lediglich das alte verändern.

Klassiker von Haustiername bis Geburtstag

Benutzer setzen ihre Passwörter oft aus verschiedenen Bausteinen zusammen. So kommen unter anderem Wörter, Jahreszahlen, Buchstabensequenzen wie „abc“, wiederholende Zeichen und Tastaturmuster zum Einsatz. Die Ergebnisse einer von Google durchgeführten Studie (siehe ix.de/zr2y) zeigten, dass die meisten Benutzer persönliche Informationen in Passwörter einbauen: Beliebt sind Namen von Haustieren, Daten wie Geburtstage, Namen von Familienmitgliedern et cetera. Auf diese Art entstehen Passwörter, die sich einfacher merken lassen als komplett zufällig generierte. Angreifer nutzen die Muster aus, die durch derartiges „Konstruieren“ von Passwörtern entstehen.

Auch der Name des Unternehmens ist ein häufig verwendeter Baustein. Zwar werden zusätzlich beliebte Zeichensubstitutionen wie das erwähnte Ersetzen des Buchstabens „e“ durch die Ziffer „3“ oder „a“ durch „@“ angewendet (Unt3rn3hm3n statt Unternehmen), aber solche Passwörter lassen sich durch einen Wörterbuchangriff leicht herausfinden.

Gängige Substitutionen, bei denen Buchstaben durch ähnlich aussehende Ziffern oder Sonderzeichen ersetzt werden, erhöhen die Stärke des Passwortes kaum, weil sie bekannt und daher in Passwort-

cracking-Regeln enthalten sind. Mit einem Passwortaudit lässt sich auch zeigen, dass Komplexitätsanforderungen bei Passwortrichtlinien die Sicherheit von Passwörtern leider nicht besonders erhöhen. Wenn ein Passwort vom System nicht akzeptiert wird, modifiziert der Benutzer es so lange, bis es akzeptiert wird. Diese Anpassungen sind meist vorhersehbar, wie das folgende Beispiel zeigt:

Initiales Passwort:	passwort
Es muss ein Großbuchstabe verwendet werden:	Passwort
Es muss eine Ziffer verwendet werden:	Passwort1
Es muss ein Sonderzeichen verwendet werden:	Passwort1!

Die Idee von Komplexitätsanforderungen bei Passwortrichtlinien ist, dass Benutzer keine Passwörter nur aus Groß- oder nur aus Kleinbuchstaben wählen und damit den Zeichenraum verkleinern. Die Anforderungen führen aber häufig zu den gezeigten Mustern, die beim Knacken von Passworthashes ausgenutzt werden können. Deshalb ist es wichtig, dass diese Richtlinien ständig neu evaluiert und gegebenenfalls angepasst werden. Mindestens genauso wichtig ist es jedoch, dass Benutzer verstehen, warum ihre Passwörter abgelehnt wurden.

Selbst gute Passwortrichtlinien führen aber nicht automatisch dazu, dass keine einfach zu erratenden Passwörter mehr verwendet werden. Es wird wohl immer Benutzer geben, die versuchen, die Richtlinien zu „umgehen“, indem sie ein einfach zu merkendes Passwort wählen, das aber gerade noch zugelassen wird. Mindestens genauso wichtig wie eine gute Passwortrichtlinie ist es, die Mitarbeiter im Umgang mit Passwörtern zu schulen. Diese Schulung sollte den Mitarbeitern vermitteln, wie starke Passwörter gewählt werden und dass Passwörter nie für mehrere Konten oder Zugänge verwendet werden sollten, weder privat noch innerhalb der Firma.

Wie man gute Passwörter erzeugt

Lange, zufällige Passwörter sind starke Passwörter. Jedoch ist es schwierig, sie sich zu merken. Eine Möglichkeit besteht darin, eine Passphrase statt eines Passwortes zu wählen. Aber auch bei Passphrasen ist Vorsicht geboten – sie sollten ebenfalls nur schwierig zu erraten sein. Von gängigen Phrasen wie „Sesam öffne dich!“ wird abgeraten.

Beispiel 1: „möwe telefon reisen geisterhaus streng“

Beispiel 2: „fische mögen keinen blauen Gurkensalat!“

Die beiden Beispiele illustrieren den Unterschied zwischen zufällig generierten und selbst kreierten Passphrasen. Für die erste Passphrase wurden zufällig fünf Wörter des deutschen Wortschatzes ausgewählt, der laut Duden aus 300 000 bis 350 000 Wörtern besteht (siehe ix.de/zr2y).

Die zweite Passphrase wurde selbst kreiert. Ein durchschnittlicher Sprecher der deutschen Sprache hat geschätzt jedoch nur 12 000 bis 16 000 Wörter im Repertoire. Alleine deswegen ist sie weniger stark, da es weniger mögliche Kombinationen aus fünf Wörtern gibt. Außerdem bildet die Passphrase einen grammatikalisch korrekten Satz, was die Anzahl der potenziellen Kombinationen nochmals verringert. Beide Passphrasen sollten dennoch leichter zu merken sein als ein zufällig generiertes Passwort – und sind sicherer als ein Passwort aus zehn Klein- und Großbuchstaben, Zahlen und Sonderzeichen.

Doppelt hält besser

Passwörter und auch Passphrasen, und seien sie noch so stark, können mit einer einzigen Phishingattacke gestohlen werden und den dazugehörigen Account kompromittieren.

Generell ist daher das Implementieren von Multi-Faktor-Authentifizierung (MFA) empfehlenswert. Mit einem zusätzlichen Faktor steht und fällt die Authentifizierung nicht mehr nur mit dem Passwort – und um einen Account zu übernehmen, müssen zwei Faktoren kompromittiert werden, was deutlich schwieriger ist. Multi-Faktor-Authentifizierung lässt sich im Windows-Umfeld implementieren, allerdings nicht für alle Protokolle. So lässt beispielsweise SMB (Server Message Block) keinen zweiten Faktor zu.

Bei einer Multi-Faktor-Authentifizierung kommen mehrere unabhängige Faktoren zum Einsatz. Alle Faktoren müssen dabei korrekt sein, sonst schlägt die Überprüfung fehl. Mögliche Faktoren sind etwas, was der Benutzer weiß, zum Beispiel ein Passwort, etwas, was er besitzt, etwa eine Smartcard, oder ein biometrisches Merkmal des Benutzers, beispielsweise ein Fingerabdruck.

Eine der wichtigsten Eigenschaften biometrischer Faktoren darf man nicht vergessen: Auch wenn diese die Sicherheit der Authentifizierung zunächst erhöhen, ist zu bedenken, dass ein biometrischer Faktor im Falle einer Kompromittierung nicht geändert werden kann. Als primäre oder einzige Faktoren sind sie daher nicht empfehlenswert.

Technik schlägt Mensch

Das beste Passwort ist eines, das man sich nicht merken kann – und auch nicht muss. Ein Passwortmanager kann einem die Arbeit abnehmen: Er erstellt zufällig generierte, lange Passwörter und merkt sie sich mitsamt dem zugehörigen Account oder Dienst. Man benötigt nur noch ein einziges Passwort, das für den Passwortmanager. Das sollte aus naheliegenden Gründen allerdings besonders stark sein. Wenn möglich, sollte für den Passwortmanager selbst sowie für die verwalteten Accounts Multi-Faktor-Authentifizierung eingesetzt werden. Jeder einzelne Account wäre auch im Falle einer Kompromittierung des Passwortmanagers noch immer geschützt.

Der Einsatz eines Passwortmanagers ist trotz des Risikos der Kompromittierung aller Passwörter auf einen Schlag empfehlenswert. Er führt bei korrekter Benutzung dazu, dass für jeden Account ein starkes, einzigartiges Passwort verwendet wird. Setzt man dazu Mehr-Faktor-Authentifizierung ein, sind die Zugangsdaten gut geschützt. (ur@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] Jürgen Seeger; Intelligenter Zugriff; Multi-Faktor-Authentisierung: Methoden und Dienstleister; *iX* 5/2019, S. 84

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

[6] Die im Text angesprochenen Werkzeuge und Artikel sind über ix.de/zr2y zu finden.

Sandro Affentranger

ist Senior Penetration Tester bei der Oneconsult AG. Sein Spezialgebiet sind Penetrationtests auf Infrastruktur- und Applikationsebene.