Security Tools 7: Paßwörter - Ein ewiges Problem?

Security Tools 7: Paßwörter - Ein ewiges Problem?

Bernd Lehle/Oliver Reutter

Unsere bisherigen Artikel beschäftigten sich hauptsächlich mit Dingen, über die sich ein Systembetreuer Gedanken machen muß. Nun kommen wir zu einem The-ma, das wirklich alle Benutzer angeht und bei dem jeder Benutzer durch leichtfertiges Verhalten Sicherheitslücken schaffen kann. Es ist daher nötig, daß diese Informationen an alle Benutzer weitergegeben werden.

Ein alltäglicher Vorgang: Jemand betritt ein Büro, in dem ein Computer steht, auf dem ein Mehrbenutzer-Betriebssystem installiert ist. Der potentielle Benutzer setzt sich an ein Terminal, findet einen Login Prompt vor und tippt seinen Benutzernamen ein. Der Rechner gibt sich damit nicht zufrieden und fragt nach einem Paßwort. Auch das tippt der potentielle Benutzer ein und wenn es stimmt, hat er Zugang zum System und kann damit arbeiten. Jeder, der dies liest, kennt diesen Vorgang und hat ihn sicher schon unzählige Male durchexerziert. Die wenigsten jedoch machen sich Gedanken darüber, was dabei abläuft und was dieser Ablauf im einzelnen bedeutet.

Grundlagen

Was bedeutet nun eigentlich das Paßwort? Wenn wir einen Account mit einem privaten Raum vergleichen, so ist das Paßwort eine Art Schlüssel zu diesem Raum. Genauso wird es von den meisten Benutzern auch verwendet. Leider versagen in der elektronischen Welt der Netzwerke solche einfachen Vergleiche sehr schnell. Übertragen auf die normale Welt ist ein Paßwort nämlich Schlüssel, Kreditkarte und Personalausweis zugleich. Es öffnet den Account wie ein Schlüssel, verschafft in dem Moment aber auch Zugang zu Ressourcen wie Rechenzeit und Plattenplatz und kann somit Kosten verursachen. Vor allem verschafft es aber demjenigen, der das Paßwort kennt, eine Identität innerhalb des Systems bzw. innerhalb des angeschlossenen Netzwerks.

Wer also sein Paßwort an andere Personen weitergibt, sollte sich bewußt sein, was er damit von sich preisgibt: Nicht nur, daß der Mitwisser nun auf alle Daten des Benutzers zugreifen, sondern auch in seinem Namen Ressourcen benutzen (z.B. Drucker oder kostenpflichtige Rechenanlagen), elektronische Post verschicken oder über andere Wege elektronisch kommunizieren kann. Alle positiven/negativen Folgen fallen auf den Benutzer zurück, der sein Paßwort und damit seine Identität

weitergegeben hat. Dies kann besonders unangenehm werden, wenn der Mitwisser andere Netzteilnehmer belästigt oder versucht in andere Rechner einzudringen.

Die Entschuldigung "Ich war`s nicht, ich hab nur mein Paßwort weitergegeben!" wird im Fall von Regreßansprüchen nicht akzeptiert. Eine derartige Regelung, die deshalb explizit die Weitergabe von Paßwörtern verbietet, ist in den meisten Benutzungsordnungen vorgesehen.

Ab und zu passiert es allerdings auch, daß ein Paßwort bekannt wird, ohne daß der Benutzer es freiwillig weitergegeben hat. Um zu verstehen, wie das passiert und wie man sich dagegen schützen kann, begeben wir uns auf einen kleinen Ausflug in die Technik.

Technisches zur Verwaltung von Paßwörtern

Es war nicht immer üblich, daß man sich auf Mehrbenutzersystemen unbedingt mit einem Paßwort ausweisen mußte. UNIX verbrachte einige seiner ersten Jahre ohne Paßwortschutz und selbst nach Einführung der Paßwörter war es noch ein langer Weg bis zu der Paßwortverwaltung wie wir sie heute kennen. Mittlerweile hat jedes ernstzunehmende Workstation-Betriebssystem eine Paßwortverwaltung, die in ähnlicher Form wie bei UNIX funktioniert. Wir wollen uns daher bei den Details exemplarisch auf UNIX beschränken.

Nun soll betrachtet werden, was im einzelnen abläuft, wenn sich ein Benutzer über das Netz auf einer entfernten Maschine einloggt, wie z.B. auf dem SERVus-Cluster:

Als erstes wird ein Terminalemulator - z.B. `telnet` - gestartet:
`telnet servint1.rus.uni-stuttgart.de`

Der angesprochene Rechner bekommt über den `inet-daemon` das Signal, daß seine Dienste im Netz gewünscht sind und startet daraufhin erst einen `telnet-daemon`, dann einen `login`-Prozeß, der sich mit einem Login Prompt meldet:

```
AIX Version 4
(C) Copyrights by IBM and by others 1982, 1994.
login:
```

Der Rechner verlangt hier den Benutzernamen und sofort darauf das Paßwort. Alles, was getippt wird, geht natürlich in dieser Form als Klartext über das Netz, bzw. befindet sich an beiden Enden kurzfristig in den Puffern des Kernels oder der graphischen Be-nutzeroberfläche:

```
login: ruslehle
ruslehle`s Password:
```

Nach kurzer Zeit kommen dann wie gewohnt die Nachrichten und der Shell Prompt zu Tage, so daß gearbeitet werden kann. Was passiert in dieser Zeit?

Der Rechner muß in irgendeiner Form feststellen, ob das eingegebene Paßwort mit dem übereinstimmt, das von dem Benutzer früher festgelegt wurde. Das könnte einfach dadurch realisiert werden, daß eine Datei existiert, in der alle Benutzernamen mit den zugehörigen Paßwörtern und sonstigen Daten gespeichert sind. Da diese Datei allerdings für alle Benutzer lesbar sein sollte, um Informationen über andere Benutzer zugänglich zu machen, können die Paßwörter dort nicht im Klartext stehen. In der Tat zeigte ein Blick auf die Datei `/etc/passwd` auf einem UNIX-System bis vor kurzem noch folgenden Anblick:

```
zohn:V41B6Zgffh91c:237:302:Hermann Zohn:/home/zohn:/bin/csh
doedel:*:415:303:Doedel-Account:/home/doedel:/bin/sh
kigel:Dgh34KaR4hMq.:419:302:Hans Kigel:/home/kigel:/bin/tcsh
```

```
maier:33qal5Bd0IpF2:368:302:Johann Maier:/home/maier:/bin/bash
```

Bis vor kurzem heißt, daß moderne UNIXe die Paßwortinformationen heute meist an sichereren Orten speichern, aber dazu später mehr.

Hier steht im einzelnen: Der Benutzername, ein verschlüsselter Text, der Paßwortinformation beinhaltet, die Benutzer- und die Gruppennummer, der echte Name des Benutzers und/oder ein Kommentar (Gecos Field), das Home-Verzeichnis und die Login Shell.

Wie wird nun verglichen, ob das Paßwort stimmt? Das Paßwort wird bei dem Prozeß strenggenommen nicht verschlüsselt, sondern als Schlüssel benutzt. Im einzelnen läuft es wie folgt: Die ersten acht Zeichen des Paßwortes werden in einen Schlüssel für das DES (Data Encryption Standard)-Verfahren umgewandelt. Dieses Verfahren benutzt Schlüssel von 56 bit Länge, daher kann ein UNIX-Paßwort nur acht Zeichen lang sein, von denen je 7 bit verwendet werden.

Mit diesem Schlüssel wird nun per DES eine Kette von Nullbits verschlüsselt. Dabei wird nicht exakt das DES-Verfahren aus der Literatur [1] verwendet, sondern ein abgewandeltes, in das während des Verschlüsselungsverfahrens noch 12 bit weitere Information eingestreut werden. Diese eingestreute Information wird als Salt (Salz) bezeichnet und besteht aus zwei alphanumerischen Zeichen.

Was steht nun in der Paßwortdatei? Man erkennt eine Kette aus13 Zeichen. Die ersten beiden bilden den Salt, die restlichen elf sind das Ergebnis der Verschlüsselung der Nullen, so gestaltet, daß keine Doppelpunkte oder nicht druckbare Zeichen darin vorkommen.

Beim Login-Vorgang wird also mit dem eingegebenen Paßwort unter Verwendung der ersten beiden Zeichen der 13er-Kette eine Kette von Nullbits verschlüsselt. Stimmt das Er-gebnis der Verschlüsselung mit den restlichen elf Zeichen überein, ist das Paßwort richtig und der Benutzer wird zugelassen. Um aus den 13 Zeichen das Paßwort zu rekonstruieren, müßte man das modifizierte DES-Verfahren brechen, was selbst beim Einsatz von schneller Hardware heutzutage sehr aufwendig ist. Normale DES-Verschlüsselungs-Chips können hier nicht eingesetzt werden, da sie den Salt-Trick nicht beherrschen.

Bei so viel Verschlüsselungstechnik mag sich der Benutzer nun fragen, warum die Sa-che nicht absolut sicher ist.

Angriffstechniken auf Paßwörter

Die Sicherheit der Paßwörter läßt sich auf verschiedene Weisen umgehen, dazu wird am besten nochmals der bereits beschriebene Login-Prozeß betrachtet.

Das erste Problem stellt der Benutzer dar. Um vernünftig arbeiten zu können, muß er sein Paßwort irgendwo außerhalb des Rechners speichern, so daß er es immer wieder findet, wenn er es braucht. Idealerweise weiß er seine Paßwörter auswendig. Viele Benutzer vertrauen aber ihrem Gedächtnis nicht und benutzen andere Hilfsmittel. Eines der schlechtesten Hilfsmittel ist das Aufschreiben des Paßwortes. Selbst wenn dieser Zettel im Portemonnai oder verschlossen im Schrank liegt, ist diese Lösung nicht ak-zeptabel. Ein neugieriger Blick im falschen Moment reicht aus, um das Paßwort herauszufinden. Wesentlich besser ist es, sich für seine Paßwörter ein System zu überlegen, mit dem man anhand einfacher Hilfsmittel (z.B. Telefonbuch oder andere öffent-liche Quellen großer Informationsmengen) ein vergessenes Paßwort schnell wieder re-konstruieren kann, obwohl das Paßwort selbst kompliziert und schlecht zu merken ist.

Der nächste Angriffspunkt ist das Eintippen des Paßwortes. Die meisten Menschen, die häufig am Computer arbeiten, können relativ schnell tippen, so daß es für einen normalen Beobachter nicht

mö glich ist, durch Verfolgen der Finger den getippten Text zu lesen. Probleme tauchen dann auf, wenn jemand sehr langsam tippt oder das Paßwort so kompliziert ist, daß man sich auf der Tastatur stark verrenken muß und so wiederum nur sehr langsam tippen kann. Es zählt zum guten Ton, wenn man gemeinsam an einem Computer arbeitet, wegzuschauen, wenn jemand sein Paßwort eintippt. Was nämlich trotz schnellen Tippens immer wieder passiert ist das Folgende:

```
login: ruslehle
ruslehle`s Password:
login incorrect
login: sf%hsgt
sf%hsgt`s Password:
```

Was ist hier passiert? Der Benutzer hat sich beim ersten Eingabeversuch des Paßwortes vertippt. Er weiß dies und da er in Eile ist, kann er es nicht erwarten, das Paßwort richtig einzugeben, wobei er dann vergißt, daß auch der Login-Name erneut gefragt wird. Statt des Login-Namens gibt er nun sein Paßwort ein, was nun jeder in Sichtweite des Bildschirms mühelos lesen kann. Nun heißt es sich schnell richtig einzuloggen, den Screen zu löschen und das Paßwort zu ändern, denn normalerweise steht der zweite, nun auch ungültige, Login-Versuch wie folgt in den System-Logdateien:

```
Jan 7 09:54:45 login failed from servint1 as sf%hsgt
```

Da der Rechner das Paßwort als Login-Name interpretiert, den es natürlich nicht gibt, protokolliert er den Vorgang als ungültiges Login durch den Benutzer `sf%hgst`. Auf manchen Systemen sind die Logdateien für alle Benutzer lesbar und somit kann jeder die Paßwörter bekommen.

Hat der Benutzer nun auch diese Klippen umschifft, droht ihm neue Gefahr durch die interne Verarbeitung seines Paßwortes. Gibt der Benutzer einen Text auf der Tastatur ein, wird er vom Tastaturtreiber des Kernels entgegengenommen und verbleibt dann kurz in einem Puffer, bis er an die Anwendung weitergeleitet wird. Diese Puffer können von einem Benutzer mit entsprechenden Privilegien gelesen und verändert werden. So ist es leicht mö glich Tastatureingaben abzuhören oder zu verändern.

Noch leichter, nämlich ohne Privilegien funktioniert das ganze, wenn die Anwendung, an die die Eingabe weitergeht eine mangelhaft gesicherte graphische Benutzeroberfläche ist. Wird z.B. X-Windows ohne die Sicherheitsmechanismen `xhost` oder `Magic Cookies` betrieben, kann jeder beliebige Internet-Teilnehmer die Tastatureingaben abhören oder verändern. Der Schutz durch die `Secure Keyboard`-Funktion, den einige Versionen des `xterm`-Programmes bieten, ist dabei völlig wirkungslos.

Wie man X-Windows sicher betreibt, wurde in dem Artikel von Markus Müller (BI. 3 94) ausführlich dargelegt.

Auch wenn das Paßwort sicher durch alle Oberflächen gedrungen ist, lauern neue Ge-fahren, sobald es sich auf ein Netzwerk begibt. Ist das Paßwort nicht für den Rechner bestimmt, an dem der Benutzer sitzt, sondern für einen, der entfernt im Netzwerk sitzt, muß es erst dorthin gelangen.

Leider ist eine Verschlüsselung des Paßwortes auf diesem Weg nicht vorgesehen und jemand, der Zugang zu einem Netzwerk-Segment hat, an dem das Paßwort vorbeikommt, kann es leicht mitlesen. Abhilfe schaffen hier nur kryptographische Login-Verfahren wie ssh oder Kerberos, die im rpool, im SERVus-Cluster oder im HWW ver-wendet werden.

Neben diesen Abhör- und Ablese-Angriffen gibt es aber noch eine ganz andere Klasse von Methoden, an Paßwörter heranzukommen. Man bezeichnet sie als Wörterbuch-Angriffe (Dictionary Attacks).

Zum Ausführen dieser Angriffe benötigt man folgende Dinge:

1. Die Paßwortdatei mit den verschlüsselten Paßwortinformationen
2. Ein Wörterbuch mit möglichst vielen Worten, die als Paßwörter in Frage kommen könnten
3. Ein Programm, das die beschriebene Paßwort- Verschlüsselung möglichst schnell und einfach durchführt
4. Viel Rechenzeit oder einen sehr schnellen Rechner.

2. und 3. sind im Internet sehr leicht zu beschaffen. Ein Repertoire von über einer Million Wörtern aus vielen europäischen Sprachen sowie wichtige Fachwörter, Sagengestalten, Vornamen, Nachnamen, Popgruppen, Geburtsdaten, etc. steht auf ftp-Servern zur Verfügung. Ein komfortables Programm, um die Wörter durchzuprobieren findet man mit Alec Muffet`s `Crack` , das mittlerweile als Version 5.0 frei verfügbar ist. `Crack` testet nicht nur alle ihm zur Verfügung gestellten Wörter, sondern berechnet von jedem Wort 280 Abarten (vorwärts, rückwärts, groß, klein, mit angehängten Zahlen und Sonderzeichen, etc.). Weitere Abarten von Wörtern sind frei konfigurierbar. Auch alle verfügbare Information über den Benutzer (Login-Name, echter Name, Rechnername, etc.) wird als Paßwort in siebzig Abarten durchgetestet. Ebenso ist das Programm darauf angelegt auf Workstation Clustern oder Parallelrechnern die volle Performance herauszuholen; Wörter durchzutesten ist trivial parallelisierbar.

Geringfügige Probleme gibt es heute mit Punkt 1, da die meisten modernen Betriebssysteme die Paßwortinformation mittlerweile getrennt von der Benutzerinformation aufbewahren. In der eigentlichen Paßwortdatei stehen nur Name, Verzeichnis und Login Shell, während eine weitere Datei existiert, die dann die Paßwortinformation trägt, aber nur vom Systembetreuer lesbar ist.

Es gibt allerdings noch eine Vielzahl von Systemen, die dieses sogenannte Shadowing zwar beherrschen, aber nur auf Anforderung auch anwenden. Eine verbreitete Anwendung, die das Shadowing nicht beherrscht, ist NIS (Yellow Pages). Dort kann man die komplette Paßwortinformation jederzeit von allen beteiligten Rechnern mit `ypcat passwd` anfordern. Wenn der Portmapper oder eine andere Sicherungsinstanz nicht ausreichend gegen Zugriffe von außen geschützt ist, kann jeder Internet-Teilnehmer die Paßwortdatei anfordern. Das Shadowing kann aber auch noch anders umgangen werden. Ein Eindringling, der es geschafft hat auf einem Rechner Systemverwalterrechte zu be-kommen, kopiert als erstes die geschützte Datei. Diese kann er an anderer Stelle dann in aller Ruhe knacken, um Zugriff auf das System zu haben, auch wenn sein erster Zugang entdeckt und gesperrt wurde. Shadowing ist also auch hier nicht die ultimative Lösung.

Auf Punkt 4 werde ich gleich in einem Beispiel eingehen.

Beispiel für einen Angriff gegen eine Paßwortdatei

Alle Daten und Fakten des folgenden Beispiels wurden von Mitarbeitern unserer Ar-beitsgruppe mit Unterstützung einiger Systemverwalter am RUS bestätigt.

Als erstes besorgten wir uns Paßwortdateien. Das kann quasi legal über NIS oder Ko-pieren realisiert werden oder man sucht gezielt nach schlecht gesicherten NIS-Systemen und holt sich dort die Paßwörter. Unsere Dateien enthielten jeweils etwa 200 Paßwörter. Dann wurde auf zwei Rechnerarchitekturen das Programm `Crack` installiert. Als Wörterbuch nahmen wir 1,2 Millionen Wörter, die wir uns im Internet zusammenkopierten. Die verwendeten Rechner waren ein PC 486DX4/100 mit 16 MB RAM unter Linux und die intel Paragon des Rechenzentrums mit 105 intel 860 XP-Prozessoren mit je 32 MB RAM unter OSF/1. Das Programm wurde leicht verändert, so daß jeder Prozessor nur noch etwa 10 000 Wörter zu probieren hatte.

Auf der Paragon wurde nur die idle-Zeit genutzt, d.h. Rechenzeit, die kein anderer Be-nutzer anforderte. Der Linux-PC widmete sich ausschließlich dem Paßwortknacken.

Die Ergebnisse waren typisch und könnten sich jederzeit überall wiederholen. Die gefundenen Paßwörter verteilen sich wie folgt:

a. Nach wenigen Sekunden bis wenigen Minuten kamen 2-5 Prozent der Paßwörter heraus, die einfach dem Login-Namen entsprachen
b. Innerhalb einer Stunde (Paragon), bzw. eines halben Tages (PC) kamen weitere 15 Prozent der Paßwörter heraus, die die Struktur <Wort><Ziffer> hatten (z.B. herbert4) oder unverändert im Wörterbuch standen
c. In den folgenden 55 Stunden (Paragon), bzw. 14 Tagen (PC) kamen weitere 3-5 Prozent der Wörter heraus, die etwas komplizierter waren (groß/klein, mit Jahreszahlen oder Sonderzeichen).

Insgesamt wurden 20-25 Prozent der Paßwörter gefunden, was gutem Durchschnitt entspricht.

Man kann also davon ausgehen, daß ein beliebiger Angreifer, wenn er schlecht ausgestattet ist in zwei Wochen, wenn er gut ausgestattet ist an einem Wochenende oder schneller zum selben Ergebnis kommen kann.

Schutzmaßnahmen

Sie werden sich zurecht fragen, was man nun dagegen tun kann. Im folgenden werden wir einige Tips zum besseren Umgang mit Paßwörtern geben, die man als Systembetreuer kennen muß und seinen Benutzern weitergeben sollte. Diese Regeln gelten für alle paßwortgeschützten Vorgänge wie Accounts auf UNIX, VMS, Windows NT, o.ä. aber auch für Telebanking oder weniger offensichtliche Paßwörter:

1. Ein Paßwort wird niemals und an niemanden weitergegeben. Wenn mehrere Personen an einem Projekt arbeiten, bekommt jeder ein eigenes und der Rest wird über Dateiberechtigungen oder `set uid`-Programme gemacht
2. Ein Paßwort darf kein Wort sein, das in einem Wörterbuch irgendeiner Sprache zu finden ist oder in irgendeiner Form Bezug zum Benutzer hat
3. Ein Paßwort sollte Groß- und Kleinbuchstaben sowie Zahlen oder Sonderzeichen enthalten
4. Ein Paßwort wird nirgends aufgeschrieben
5. Ein Paßwort muß leicht und schnell zu tippen sein
6. Ein Paßwort sollte die maximale Länge nutzen (8 Zeichen unter UNIX, 14 unter Windows NT, Pass Phrases bei PGP oder anderen Anwendungen).

Nach so vielen Regeln hier auch ein paar konstruktive Vorschläge, das Paßwortproblem in den Griff zu bekommen:

1. Verwenden Sie Paßwortsysteme. Paßwörter kann man beispielsweise aus einzelnen Elementen zusammensetzen, die zusammen keinen Sinn machen und durch Sonderzeichen getrennt sind
2. Sichern Sie Ihr System weitestmöglich durch die Verwendung von Shadowing und abgesichertem NIS bzw. X-Windows gegen Paßwortdiebstahl ab
3. Nutzen Sie Programme wie `npasswd` oder `passwd+`, die es dem Benutzer nicht erlauben schlechte Paßwörter zu wählen, da sie diese sofort beim Ändern mit einem Wörterbuch vergleichen
4. Bevorzugen Sie wenn möglich moderne, sichere Authentisierungsmechanismen wie kerberos und ssh
5. Greifen Sie in kritischen Umgebungen zu Einweg-Paßwörter.

Der Punkt 4 ist leider etwas heikel, da er nicht leicht zu implementieren ist und meistens das Mit-sich-herumtragen einer längeren Liste von Einweg-Paßwörtern bedingt. Interessenten, die damit

experimentieren wollen, sei das Programmpaket S/Key empfohlen. In kritischen Umgebungen im Industriebereich kommen Smart Cards zum Einsatz, die aus einer Vorgabe beim Login und einer PIN das Einwegpaßwort errechnen und so die Listen überflüssig machen.

Ansprechpartner in sicherheitsrelevanten Fragen

- sneakers@rus.uni-stuttgart.de
- dfncert-request@cert.dfn.de

Literatur

- [1] Bruce Schneider, Applied Cryptography, Wiley & Sons, 1996
- [2] manpage zu crypt (3)

Bernd Lehle, NA-5531
E-Mail: lehle@rus.uni-stuttgart.de

Oliver Reutter, NA-4513
E-Mail: Oliver.Reutter@rus.uni-stuttgart.de