

Sicherheit in Netzwerken – Praktische Aufgaben 1

Das erste Praktikum besteht aus mehreren Teilen

- A) Betrachtung grundlegender Begriffe zu Charakterisierung von Problemen im Bereich von IT-Sicherheit in Netzwerken,
- B) Gezielte Hacken von Passwörtern in einer isolierten Umgebung (eigene Programmierung) und
- C) Ausprobieren von Software-Werkzeugen, um die Informationsflusssicherheit in Netzwerken zu testen.

Nach diesen praktischen Aufgaben kennen Sie die grundlegenden Begriffe aus dem Bereich IT-Sicherheit, die Ihnen eine Identifikation der unterschiedlichen Problemfelder in der IT-Sicherheit erlauben. Und Sie haben erste Erfahrungen gesammelt, welche Gefahren und praktischen Probleme bei Attacken auf ein IT-System auftreten können, so dass entweder Datensicherheit - am Beispiel von Passwörtern - oder Informationsflusssicherheit - am Beispiel der Überwachung von Kommunikationskanälen in einem IT-System - *nicht mehr gewährleistet* werden können.

Teil A: Betrachtung von grundlegenden Begriffen

Die Studie der grundlegenden Begriffe liefert ein Modell, um Probleme im Bereich der IT-Sicherheit zu analysieren und die Probleme lokalisieren zu können. Aus technischer Sicht lassen sich die Problemfelder im Bereich der IT-Sicherheit aufteilen in: Informationsflusssicherheit (engl. security), Datensicherheit (engl. safety) und technischer Datenschutz (engl. protection).

Beantworten Sie, bitte, die folgenden Fragen:

- i) Was der Unterschied zwischen Informationsflusssicherheit und Datensicherheit? Beschreiben Sie die Begriffe und Unterschiede mit Ihren eigenen Worten.
- ii) Überlegen welche Schutzziele zu garantieren bzw. welche Komponenten in einem Netzwerk zu schützen sind, wenn wir Informationsflusssicherheit gewährleisten wollen? Machen Sie eine Liste und begründen Sie die Ihre jeweilige Auswahl.
- iii) In analoger Weise welche Schutzziele lassen sich formulieren, wenn wir Datensicherheit garantieren wollen? Machen Sie ebenfalls eine Liste mit einer stichwortartigen Begründung.
- iv) Wie ergänzt der Begriff von so genannter Privatheit (engl. privacy) die bisher betrachteten Problemfelder? Was kann nicht durch eine technische Lösung garantiert werden?

Teil B: Hacken von Passwörtern

Der erste Zugang zu einem IT-System erfolgt in der Regel über einen Autorisierungsmechanismus. Die Autorisierung geschieht dabei durch Authentifizierung. Der meist genutzte Mechanismus zur Authentifizierung und damit den Zugriff auf IT-Systeme zu freizugeben, sind Passwörter. Die folgende praktische Aufgabe soll vermitteln, wann Passwörter sicher sind. Weitere Kriterien, wie Sicherheit von Passwörtern zu garantieren ist, finden Sie im Foliensatz zur Veranstaltung auf der Webseite <http://isec.fontysvenlo.org>

Hashed Passwords

In einem Unix Betriebssystem und dessen Derivaten, wie dem Linux Betriebssystem, werden selbst verschlüsselte Passwörter von dem gemeinen Benutzer „versteckt“. In einem System ohne „shadowed passwords“ werden in `/etc/passwd/` die folgenden Informationen über einen Benutzer gespeichert:

- Username
- Hashed password
- User ID (UID)
- Default group ID (GID)
- Vollständiger Name
- Home directory path
- Login shell

Diese Datei ist für alle lesbar, d.h. alle Nutzer können sie einsehen, aber sie darf nur von dem Benutzer `root` geschrieben werden. Somit kann ein Angreifer (hacker or cracker) das „hashed password“ von jedem Benutzer recht einfach erhalten. Das „hashed password“ ist für den Angreifer nützlich, weil dieses Passwort relativ einfach mit einer brute-force Methode ausgetestet und damit ermittelt werden kann. Wenn der Angreifer erstmal die Datei mit einem „hashed password“ erhalten hat, dann kann die angestrebte Entschlüsselung auch lokal auf dem eigenen Rechner erfolgen. Dabei bleibt die Attacke für weitere Sicherheitssysteme des anzugreifenden IT-Systems unbeobachtet. Die meisten Benutzer wählen ein Passwort, in der Regel recht einfach zu knacken ist.

Shadowed passwords

Beim Verstecken von Passwörtern („shadowed passwords“) werden die „hashed passwords“ in einer anderen Datei gespeichert, in der Regel in `/etc/shadow` in einem Linux-System. Dieses Verzeichnis kann in der Regel nur über einen Administrator-Account (`root`) gelesen werden. Neuere Linux-Systeme verwenden solche „shadowed passwords“. Damit ist der „Klau“ von solchen Passwörtern weitaus schwieriger, weil nur mit einem Zugriff über `root` die „hashed passwords“ gefunden werden können. Dieses wurde in der Regel als ausreichender Schutz aufgefasst, weil nur Benutzer mit `root`-Rechten, die dazugehörigen Dateien und entsprechenden Daten überhaupt sehen können.

Wenn Passwörter als „shadowed passwords“ abgelegt sind, dann findet sich in der `/etc/passwd`-Datei ein `x` (oder ein anderer Buchstabe) anstelle des Passworts. In System mit „shadowed passwords“ werden unter `/etc/shadow` die folgenden Benutzerinformationen gehalten:

- User login name
- Hashed password, beginning with **salt** (s. u. eigener Abschnitt)
- Anzahl Tage, seit dem 1. Januar 1970 das Passwort verändert wurde
- Anzahl Tage, seit die letzte Änderung stattgefunden hat
- Anzahl Tage, bis zu dem Tag eine erneute Änderung verlangt wird
- Anzahl Tage, an denen gewarnt wird, dass das Passwort ausläuft
- Anzahl Tage, bevor der Account inaktiv gesetzt wird
- Anzahl Tage, seit dem 1. Januar 1970 die der Account inaktiviert war
- Vorbehaltliche Angaben

Leider werden bei einigen Netzwerk-Authentifizierungsmechanismen die verschlüsselten Passwörter im Netzwerk übertragen. Damit werden diese Daten wieder

angreifbar, indem ein Angreifer versucht genau diese Daten abzuhören. Zusätzlich kann es passieren, dass diese Passwörter per Backup-Strategien auf andere Speichermedien, wie portable Festplatten, kopiert werden. Offensichtlich werden damit „hashed passwords“ wieder angreifbar.

Salz in Passwörtern (salt)

salt ist eine zufällig generierte Zeichenkette, die zu einem Passwort hinzugefügt wird und bevor das Passwort „gehashed“ wird. **salt** (Salz) kann tatsächlich aufgefasst werden, in dem man sich vorstellt, dass die eigentliche Zeichenkette des Passworts „gewürzt“ wird. Ein „Einstreuen“ zusätzlicher Buchstaben soll das Entschlüsseln des tatsächlichen Passworts erschweren. Diese zufällig generierte Zeichenkette **salt** wird dann mit dem Passwort gespeichert. MD5 (Message-Digest algorithm 5) ist ein Verschlüsselungsalgorithmus. Bei der Anwendung von MD5 beginnt in diesem Fall die **salt**-Zeichenkette mit ‚\$1\$‘ und endet mit einem Zeichen ‚\$‘. Lese die **salt**-Zeichenkette für den spezifischen Benutzer aus der shadow-Datei und benutze diese Zeichenkette in der crypt()-Funktion. Die Signatur der crypt()-Funktion hat folgende Signatur

```
/**
```

```
* Linux/BSD MD5Crypt function  
* @return The encoded password an MD5 hash  
* @param salt 8 byte string  
* @param password user password  
*/
```

```
public static final String crypt( String password, String salt);
```

Ihre Aufgabe

Ausnahmsweise dürfen Sie hier jetzt einmal als Hacker betätigen. Bitte, werden Sie nicht zum Cracker! Was ist der Unterschied zwischen einem Hacker und einem so genannten Cracker? Erklären Sie die Begriffe, bitte, mit eigenen Worten. Überlegen Sie sich bitte auch, welche Rolle Sie beim Lösen der Aufgabe eingenommen haben!

Nehmen Sie an, dass Sie durch einen glücklichen Zufall an eine /etc/shadow-Datei eines Studenten-Administrators bei Fontys gekommen sind. Unglücklicherweise sind alle Passwörter mit einem MD5-Hash-Algorithmus verschlüsselt. Stellen Sie sich aber einmal vor was, Sie alles mit dem entschlüsselten Passworts eines Studenten-Administrators anfangen könnten.

1. Schreiben Sie einen **Brute Force Password Cracker** in *Java*, so dass die Passwörter in der gespeicherten shadow-Datei entschlüsselt werden können. Das Passwort besteht aus 3, 4 und 6 Zeichen aus dem Alphabet mit ausschließlich Kleinbuchstaben [a-z]. Um Ihr Ziel zu erreichen, müssen Sie Folgendes tun:
 - a. Parsen Sie die shadow-Datei und generieren Sie ein Abbild das Benutzernames (user name) und des „hashed password“.
 - b. Schreiben Sie eine Java-Klasse, die systematisch Worte über dem gegebenen Alphabet [a-z] generiert.
 - c. Benutzen Sie die crypt()-Funktion aus der nl.fontys.linux.utils.MD5Crypt-Klasse, um Hash-Werte für diese generierte Worte zu erzeugen. Benutzen Sie dazu auch die **salt**-Zeichenkette aus der shadow-Datei.
 - d. Vergleichen Sie die erzeugten hash-Werte mit den des ursprünglichen „hashed password“ aus der shadow-Datei.
 - e. Testen Sie Ihre Implementierung mit dem Account ‚threeletterman‘, der ein Passwort mit drei Buchstaben hat.

2. Sie sollten das Passwort von ‚theelletterman‘ ermitteln können! Wie lange brauchen Sie, um das Passwort zu entschlüsseln? Erweitern Sie die Funktionalität Ihres Brute Force Password Crackers, so dass Sie die Zeit messen können, die Sie brauchen, um ein Passwort zu knacken? Wieviele Passwörter sind möglich, wenn sie genau aus drei Buchstaben über dem Alphabet [a-z] bestehen? Bestimmen Sie ein gemessenes Ergebnis und geben Sie eine entsprechende Formel an.
3. Nach dem Sie Ihre Implementierung um eine Möglichkeit zur Zeitmessung erweitert haben, testen Sie Ihr Programm, indem Sie das ‚forthletterman‘-Passwort knacken. Wie lange brauchen Sie dafür? Wie lange würde es dauern, wenn Sie Passwort bestehend aus sechs Buchstaben entschlüsseln wollen?
4. Die Methode hat im Namen die Bezeichnung „brute force“; d.h. es ist eine aus der Sicht der Implementierung sehr einfache und auch ineffiziente Methode. Was können Sie machen, um den Entschlüsselungsmechanismus zu verbessern? Geben Sie mindestens drei Vorschläge an.

Teil C: Ausprobieren von Software-Werkzeugen

Verschiedene Anbieter werben mit Software-Werkzeugen, die dazubeitragen sollen ein Informationssystem in einem Netzwerk „sicher“ zu machen.

Betrachten Sie die Liste der folgenden Anbieter solcher Software-Werkzeuge. Die Liste ist sicherlich unvollständig.

- Nessus - <http://www.nessus.org>
- Kismet - <http://www.kismetwireless.net>
- Ethereal - <http://www.ethereal.com>
- Tcpdump - <http://www.tcpdump.org>
- Windump - <http://winpcap.polito.it>

Gehen Sie auf die Web-Seiten der angegebenen Anbieter und untersuchen Sie die Software, indem Sie die folgenden Fragen beantworten:

1. Bestimmen Sie die Funktionalitäten der Software-Werkzeuge im Allgemeinen.
2. Für welches Betriebssystem ist die Software jeweils geeignet?
3. Können Sie aus den Angaben 1 und 2 ableiten, worin besondere Schwachstellen in den jeweiligen IT-Systemen (u.a. im Abhängigkeit des gewählten Betriebssystems) liegen könnten?
4. Erinnern Sie sich an das ISO/OSI 7 Schichtenmodell und erinnern Sie sich auch an die TCP/IP-Implementierung. Welche Schichten werden unter TCP/IP in Bezug auf das ISO/OSI 7 Schichtenmodell nicht implementiert?
5. Wenn Sie sich jetzt nochmals die angebotenen Software ansehen, welche Ebenen einer (hybriden) TCP/IP-OSI Architektur werden auf Ihre Informationsflusssicherheit getestet und welche jeweils nicht?
6. Unterscheiden Sie zwischen den ermittelten Funktionalitäten in Bezug auf „gute“ und „schlechte“ Software-Werkzeuge. Welche können neben dem Schutz der eigenen IT-Infrastruktur auch zu Attacken missbraucht werden?
7. Denken Sie über (software-technische Arbeits-)Umgebungen nach, in denen manche der angegebenen Software-Werkzeuge zu einer Gefahr werden können. Geben Sie eine entsprechende Liste von Problemen an. Haben Sie Ideen, was zu tun ist, um ggf. die Software-Werkzeuge dennoch einsetzen zu können?