UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR IT-SICHERHEIT

Praktikum 2

Passwörter

Zu bearbeiten bis zum 02. Mai 2023

Cybersecurity im Sommersemester 2023

Jan Wichelmann, Anja Köhl

Einführung

In diesem Praktikum werden Sie sich mit Angriffen auf die Passwort-Infrastruktur eines typischen Linux-Systems befassen. Dabei sollen sie versuchen, eine Anzahl nachlässig gewählter Passwörter zu knacken. Dazu werden wir Ihnen Dateien zur Verfügung stellen, in denen verschlüsselte Passwörter gespeichert sind, wie sie in so gut wie jedem Linux-System zum Einsatz kommen. Im echten Leben ist es (zum Glück) nicht ganz so einfach, an diese Dateien heranzukommen; wir werden jedoch in den späteren Praktika darauf eingehen, wie man an solche Daten gelangen kann.

Aufgabe 1 Vorbereitung: Eine Shadow-Datei angreifen

In dieser Aufgabe sollen Sie einen Wörterbuch-Angriff auf eine `shadow`-Datei implementieren. Diese Datei wird auf den gängigen Linux-Systemen genutzt, um die Passwörter der Benutzer des Systems verschlüsselt abzuspeichern. Lesen Sie sich die Man-Page zu dieser Datei durch, um ihren Aufbau zu verstehen:

```
man shadow
```

Die in der Datei gespeicherten Passwörter sind folgendermaßen aufgebaut:

```
$ID$Salt$Hash
```

Hierbei gibt `ID` das verwendete Verfahren (siehe Man-Page der `crypt`-Funktion) und `Salt` den verwendeten Salt an, mit denen das verschlüsselte Passwort `Hash` generiert wurde. Im Praktikum werden wir anstelle der unter Linux eingesetzten crypt-Funktion exemplarisch einen einfachen SHA-512 Hash einsetzen, der erheblich effizienter zu berechnen ist und daher keinesfalls einfach so in der Praxis eingesetzt werden sollte[1].

Die Hashfunktion ist Ihnen durch `PasswordHash.java` gegeben und sollte nicht modifiziert werden. Ein Beispiel für die Verwendung gibt Ihnen die Datei `Main.java`, die Sie um Ihren Angriffscode ergänzen sollen. Schreiben Sie nun ein Java-Programm[2] zur Durchführung eines Brute-Force-Angriffs auf eine `shadow`-Datei unter Beachtung der folgenden Schritte:

1. Es gibt grundsätzlich zwei Ansätze, dieses Problem algorithmisch anzugehen:
   - Sie testen für jedes ungelöste Passwort alle Möglichkeiten
   - Sie testen für jede Möglichkeit alle ungelösten Passwörter

   Überlegen Sie sich die Vor- und Nachteile der beiden Ansätze, und entscheiden Sie anschließend, welcher hier eher geeignet sein könnte.

[1] Eine gute weiterführende Erklärung hierzu findet sich hier: `https://security.stackexchange.com/q/211`

[2] Sie können auch eine andere Programmiersprache Ihrer Wahl benutzen; im Moodle finden Sie bereits einige fertige, getestete Versionen der Hashfunktion für andere Sprachen. Falls Sie sie selbst neu implementieren sollten: Stellen Sie sicher, dass sich die Hashfunktion identisch zur vorgegebenen verhält! Schicken Sie anschließend gern die Implementierung der Hashfunktion an die Betreuer, damit sie in die Sammlung aufgenommen wird.

2. Schreiben Sie eine Methode `bruteForce()`, die zu einer `shadow`-Datei versucht die Klartextpasswörter zu ermitteln. Dazu müssen aus jeder Zeile der Salt und der Hash extrahiert werden. Um einen Wörterbuch-Angriff durchzuführen, stellen wir Ihnen ein von uns erstelltes Wörterbuch `dict.txt` zur Verfügung, welches Sie im Moodle herunterladen können.

3. Erweitern Sie die Methode `bruteForce` in der Art, dass Sie die in der Wörterbuchdatei enthaltenen Worte mit einzelnen Ziffern, Zahlen oder Sonderzeichen verketten, oder auf andere Art und Weise verändern.

   Weiterhin sollte die Methode auch Passwörter finden können, die ohne Nutzung des Wörterbuchs erzeugt wurden.

   Um einen Brute-Force-Angriff durchzuführen, ist es hilfreich ein Gefühl dafür zu haben wie "komplex" ein Passwort ist. Ein einfaches Maß dafür ist die Shannon-Entropie auf Basis der Anzahl aller möglichen Passwörter: Je höher die Entropie des jeweiligen Passwortschemas, desto aufwändiger der Angriff.

   Bearbeiten Sie hierzu die Verständnisfragen auf dem Praktikumsserver.

Hinweis: Seien Sie kreativ bei Ihrem Brute-Force-Angriff und überlegen Sie, wie sich Passwörter zusammensetzen könnten. Prüfen Sie beispielsweise:

- Kurze Ketten beliebiger Buchstaben oder Ziffern.
- Wörter des Wörterbuchs, die mit zusätzlichen Zeichen beginnen oder enden.
- Groß- und Kleinschreibung.
- Wörter des Wörterbuchs, deren Buchstabenreihenfolge invertiert wurde.

Sie können davon ausgehen, dass für Passwörter nur druckbare Zeichen aus dem 7-Bit-ASCII-Zeichensatz[3] zum Einsatz kommen. Umlaute oder gar Unicode-Zeichen kommen nicht vor, Sie müssen sich also keine Gedanken um Zeichencodierung oder ähnliches machen.

Achten Sie gegebenenfalls auf die Zeilenenden im Wörterbuch und in der Shadow-Datei. Windows markiert neue Zeilen mit der Steuerzeichen-Sequenz `CRLF` (Schreibweise: `\r\n`), während Linux nur das Steuerzeichen `LF` (`\n`) verwendet[4]. Die im Moodle verfügbaren Dateien sind allesamt mit Linux-Zeilenenden versehen, aber bei Bearbeitung unter Windows könnten sich diese versehentlich ändern. Manche Klassen zum zeilenweisen Lesen von Dateien neigen dazu, das `CR`-Steuerzeichen (`\r`) als Teil des Texts in der Zeile zu interpretieren, was entsprechend zu falschen Hashwerten führt.

Aufgabe 2 Password-Challenge

Laden Sie die Passwortdatei `shadow.txt` aus dem Moodle herunter.

Ihr Ziel ist es nun, möglichst viele Passwörter aus dieser Datei zu ermitteln. Erweitern und modifizieren Sie Ihren Quelltext dazu gegebenenfalls. Das Wörterbuch `dict.txt` wird Ihnen ebenfalls zur Verfügung gestellt.

Die gefundenen Passwörter können Sie auf dem entsprechenden Praktikumsserver eingeben und auf Korrektheit prüfen lassen.

Es gibt hierbei drei Kategorien:

- Pflicht-Passwörter (`user1` bis `user7`): Diese Passwörter müssen geknackt werden, um das Praktikum zu bestehen.
- Freiwillige Passwörter (`user8` bis `user30`): Diese Passwörter sind freiwillig und bringen Aufgabenpunkte, werden also unabhängig von der Anzahl der Funde bewertet.
- Flag-Passwörter (`user31` bis `user42`): Diese Passwörter sind freiwillig und generieren Flags. Die Anzahl der Punkte skaliert abhängig davon, wie oft das jeweilige Passwort geknackt wurde.

[3] Auf deutsch: Zeichen, die Sie ohne Weiteres auf einer amerikanischen Tastatur eintippen können.
[4] `https://stackoverflow.com/a/1552775`