Hands-on: Metadatenanalyse

Passwortchecker: Sinnvoll oder sinnlos?

Du hast ein neues Passwort und willst wissen, wie sicher es ist. Du gibst es also in einen Online-Checker ein – vielleicht sogar in einen, der es gleich anzeigt und meint: „Super sicher!“ Aber Moment... vertrauen wir diesen Tools zu sehr? Und was genau bewerten sie eigentlich?

Die meisten dieser Tools berechnen eine theoretische Sicherheit basierend auf Länge, Sonderzeichen und Zahlen. Was sie nicht berücksichtigen: dass viele Menschen sehr ähnliche „sichere“ Passwörter verwenden – und diese längst in Wörterbüchern gelandet sind.

Fazit: Passwortchecker können ein grober Hinweis sein, aber echte Sicherheit misst sich daran, wie gut ein Passwort gegen *reale Angriffe* schützt – und das probieren wir heute aus!

Was ihr lernen werdet

- Wie Passwort-Cracking-Tools wie `john the ripper` funktionieren
- Wie Wörterbuch- und Brute-Force-Angriffe funktionieren
- Wie man mit einem echten PDF-Dateihash arbeitet
- Warum sichere Passwörter wichtiger sind als je zuvor!

1. Theorie: John the Ripper und die Angriffsarten

Was ist `John the Ripper`?

`John the Ripper` (kurz: `john`) ist ein beliebtes Open-Source-Tool zum Knacken von Passwörtern. Es analysiert Passwort-Hashes und versucht, die Klartext-Passwörter zu rekonstruieren – entweder durch Wörterbücher oder durch systematisches Durchprobieren.

Wörterbuchangriff

Bei einem Wörterbuchangriff wird eine vorbereitete Liste (das „Wörterbuch“) von häufigen Passwörtern verwendet. Diese Liste kann tausende, manchmal Millionen Einträge umfassen. `john` prüft, ob einer dieser Einträge den Hash knackt.

Brute-Force-Angriff

Brute-Force bedeutet: Alle möglichen Kombinationen von Zeichen werden ausprobiert – systematisch und gnadenlos. Diese Methode garantiert irgendwann Erfolg – dauert aber bei langen Passwörtern mit vielen Zeichentypen u.U. Jahrtausende... außer dein Passwort ist „katze123“

2. Vorbereitung: Die PDFs entpacken

Im Workshop-Repository liegt eine ZIP-Datei mit verschlüsselten PDFs. Diese müsst ihr über das Terminal entpacken. Anschließend wechselt ihr in den Zielordner:

```
mkdir -p ~/Desktop/PDF

cd  ~/IT-Workshop/Hands-on

unzip Passwortsicherheit.zip -d ~/Desktop/PDF
```

3. Praktischer Teil: PDF-Hash extrahieren!

Schritt 1 – Die Python-Umgebung vorbereiten

Navigiere in das Verzeichnis von John the Ripper

```
cd ~/Desktop/john
```

Du brauchst `pdf2john.py`, dafür musst du aber zuerst die Python-Umgebung initialisieren!

```
source ./bin/activate
```

Schritt 2 – Hash aus der PDF extrahieren

Verwende `pdf2john.py`, um den Hash aus deinem geschützten PDF zu extrahieren:

```
python ./run/pdf2john.py ~/Desktop/PDF/<dein PDF> > ~/hash.txt
```

Der extrahierte Hash befindet sich nun in der Datei `hash.txt` in deinem home-Verzeichnis.

4. Praktischer Teil: Den PDF-Hash knacken!

Jetzt kommt der spannende Teil: Du jagst den PDF-Hash durch `John the Ripper`!

Wörterbuchangriff auf die PDFs 1 bis 6

Für die ersten sechs PDFs nutzen wir einen **Wörterbuchangriff**.

```
./run/john --format=pdf --wordlist=wordlist-german.txt ~/hash.txt
```

Schritt 4 – Brute-Force für die restlichen PDFs

Für alle weiteren Dateien (ab PDF Nummer 7) reicht der Wörterbuchangriff nicht mehr aus. Hier kommt der **Brute-Force-Angriff** zum Einsatz.

Verwende den Standardmodus von `john`, um automatisch einen Brute-Force-Versuch zu starten:

```
./run/john --format=pdf --mask=?a?a?a?a?a?a --min-length=1 ~/hash.txt
```

Achtung: Ja nach Hardware kann dieser Prozess mehrere Minuten dauern – Geduld zahlt sich hier aus.

5. Challenge!

Teil 1: Befreit die PDFs von ihren Passwörtern!

Eure Aufgabe:

- Entfernt anschließend mit einem passenden Tool den Passwortschutz vollständig.
- Überprüft, ob die PDF sich danach ohne Passwort öffnen lässt.

Teil 2: Öffnet das geheime ZIP-Archiv!

Im Verzeichnis `/Desktop/PDF` befindet sich ein, verschlüsseltes Archiv:

`challenge.zip`

Es enthält ein verstecktes Meme das die Kommentarsektion bei einer typischen . Ihr kennt das Spiel:

- Findet das Passwort.
- Knackt das Archiv.
- Spielt das Video ab.

Regeln für beide Teile:

- Es gibt **keine Hilfe** von uns!
- Fragt Duck.ai
- Die nötigen Tools sind **nicht vorinstalliert**. Nutzt `apt`, um sie selbstständig zu installieren.
- Denkt an **Multitasking!** Während John the Ripper rechnet, könnt ihr z.B. bereits Duck.AI befragen und z.B. das Entschlüsselungstool für das ZIP-Archiv einrichten
- Dokumentiert eure Vorgehensweise stichpunktartig – was hat funktioniert, was nicht?

Disclaimer

Dieses Hands-On zeigt euch, wie Sicherheitslücken entstehen können – nicht, wie man sie ausnutzt.

Bitte verwendet dieses Wissen ausschließlich für legale und ethische Zwecke, etwa beim Testen der Sicherheit eigener Systeme oder im Rahmen von IT-Security-Ausbildungen.

Der Einsatz von Tools wie `john the ripper` auf fremden Dokumenten ohne Zustimmung ist illegal und kann strafrechtlich verfolgt werden.

Wir setzen darauf, dass ihr verantwortungsvoll mit eurem neuen Wissen umgeht.