Cheatography

John the Ripper Cheat Sheet

by leon1 via cheatography.com/208551/cs/44706/

Beispiele

$ john --single --format =<Hashtyp> <Passwortdatei>	Angriff mit Single Crack Mode *Nutzt Informationen aus Passwortdatei, um gezielten Angriff durchzuführen.* *Bsp.: User = "john", Password = "john123"*
$ john --wordlist=<Wortliste> --format=<Hashtyp> <Passwortdatei>	Angriff mit spezifischer Wortliste
$ john –incremental[=Zeichensatz] <Passwortdatei>	Angriff mit inkrementellem Modus (Brute Force)

Was ist John the Ripper

Open-Source Tool für Password-Cracking

Standardmäßig CLI, GUI verfügbar

Erkennt oft Hashtypen automatisch

Unterstützt alle gängigen Hashmethoden

Voraussetzungen

Der Hashwert des gesuchten Passworts, z.B. aus /etc/shadow (Linux)

Ggf. der verwendete Hashing-Algorithmus

Funktionsweise von John

1. Hash-Typ (Verschlüsselungstyp) des gegebenen Hashwerts bestimmen

2. Generierung von Hashes für verschiedene Zeichenketten

3. Anhalten, wenn ein generierter Hash mit dem gegebenen Hash übereinstimmt (→ Hashkollision)

Basisbefehl

```
$ john [option] <Pa ssw ort dat -ei>
```

Masken

RegeEx-ähnliche Ausdrücke...

... zur Erstellung von Passwörtern im Brute-Force

> $ john --mask='[Pp][Aa@][Ss5][Ss5]-[Ww][Oo0][Rr][Dd]' password.hash — z.B. P@55w0rD

... zur Anpassung von Wörtern aus Wörterbuch

> $ john --wordlist="wordlist.txt" --mask='?-w?d?d?s' — z.B. password → passwo-rd12!

Umgang mit Salts & speziellen Hashes

Umgang mit Salts & speziellen Hashes

Beispiel für einen verketteten Hash mit Salt

> sha1($s.md5($p)), $s =Salt, $p = Passwort

Lösung: Dynamische Formate

> Entweder eigene oder vorgefertigte Subformate

Auflistung der Subformate

> $ john --list=subformats

Nutzung eines eigenen dynamischen Formats

> --format=dynamic="sha1($s.md5($p))"

Format des gespeicherten Hashwerts

> [userID:][$dynamic_<Subformat_Nr>]$b-ase_16_hash[$salt]

Weitere nützliche Optionen & Befehle

Mehrere CPU-Kerne verwenden

> --fork=<Anzahl>

Session einen Namen geben

> --session=<Name>

Unterbrochene Session wiederherstellen

> --restore[=<Name>]

Gecrackte Hashes mit Passwörtern ausgeben

> cat ~/.john/john.pot

Status einer Session ausgeben

> --status[=<Name>]

Die Option --format

Unterstützte Hashtypen

> z.B. Raw-SHA256, md5crypt, bcrypt

Verfügbare Formate auslesen

```
$ john –list= formats
```

Hashes aus verschlüsselten Dateien erhalten

Nützliche Links

Kompakter "Einsti-egsguide" mit Übungen	https://www.statio-nx.net/how-to-use--john-the-ripper/
Ausführliche themenspezifische Guides	https://miloser-dov.org/?tag=john--the-ripper
Dokumentation	https://github.c-om/openwall/john/-tree/bleeding-jumb-o/doc
CUPP: Tool zur Erstellung maßges-chneiderter Wörter-listen	https://github.c-om/Mebus/cupp

Der Modus --incremental

Unterstützte Zeichensätze

> z.B. digits, upper, alnum, utf8

Verfügbare Zeichensätze auslesen

```
$ john –list= inc –modes
```

Beispiel für einen verketteten Hash mit Salt

> sha1($s.md5($p)), $s =Salt, $p = Passwort

Lösung: Dynamische Formate

> Entweder eigene oder vorgefertigte Subformate

Auflistung der Subformate

> $ john --list=subformats

Nutzung eines eigenen dynamischen Formats

> --format=dynamic="sha1($s.md5($p))"

Format des gespeicherten Hashwerts

> [userID:][$dynamic_<Subformat_Nr>]$base_16_hash[$salt]

Programme, um JtR-geeignete Hashes aus Dateien zu extrahieren

```
xxx2john
```

Extrahieren des Hashwerts der Zip-Datei secret.zip

```
$ zip2john secret.zip > secret.hash
```

Auflistung aller Skripte

```
$ locate *2john
```

Nutzer- und Passwortinfos unter Linux kombinieren

```
$ unshadow /etc/p asswd /etc/s hadow > output.db
```

By **leon1**
cheatography.com/leon1/

Not published yet.
Last updated 19th October, 2024.