Cracken verkürzter Hashwerte mit Hashcat

Wir verwenden dazu die aktuelle Hashcat-Version, wie man sie auf Github herunterladen kann. https://github.com/hashcat/hashcat.git

Die Treiber für OpenCL und Cuda muss man ebenfalls installieren. Anleitungen gibt es beispielsweise hier:

https://tacticalware.com/installing-and-running-hashcat-on-ubuntu-server-18-04/

https://www.pugetsystems.com/labs/hpc/How-to-install-CUDA-9-2-on-Ubuntu-18-04-1184/#step-2-get-the-right-nvidia-driver-installed

https://developer.nvidia.com/how-to-cuda-python

Zum Vorgehen bei Hashcat:

Unter src/modules gibt es Dateien für die unterstützten Hashverfahren. *module_01400.c* ist für SHA256. Dieses speichert man unter einem neuen Namen ab, z.B. *module_01499.c*

Dann lassen sich in dem neuen Modul bestimmte Parameter einstellen. Geändert werden die Werte zu *DGST_POS0, DGST_POS1 und DGST_POS3*. Damit wird festgelegt, welche Segmente des Hashwertes verglichen werden. Man muss die Werte auf 0,1 und 3 setzen, damit nur die ersten beiden Segmente verglichen werden und man die anderen auf null setzen kann. *token.len_min[0]* und *token.len_max[0]* werden auf 19 gesetzt. Unter *static const char *ST_HASH* wird ein 19-stelliger Beispielwert eingetragen.

Der Wert für *static const u64 KERN_TYPE* wird entsprechend auf 1499 geändert.

const int out_len = 64 wird beibehalten.

Der Kernel für SHA256 im Verzeichnis OpenCL muss ebenfalls modifiziert werden. Das geht aber bislang nur bei dem einfachen Kernel *m01400_a3-pure.cl*. Dort werden die beiden letzten der vier ausgewerteten Segmente auf null gesetzt. Das muss an zwei Stellen gemacht werden.

```
const u32x z = 0;
COMPARE_M_SIMD (r0, r1, z, z);
```

Der Kernel muss dann ebenfalls neu abgespeichert werden unter *m01499_a3-pure.cl*

Dann muss man Hashcat neu bauen, wobei der Kernel erst beim Aufruf kompiliert wird.

Durch das Verfahren werden nur die ersten 16 Zeichen des 19-stelligen Hashwertes ausgewertet. Anhand der von Hashcat gefundenen Werte lässt sich einfach überprüfen, ob die fehlenden drei Stellen übereinstimmen.

Ein Aufruf erfolgt dann beispielsweise unter

./hashcat -m 1499 -a 3 hash.txt 04?H?H?H?H?H?H?H?H?H?H?H?H -o crack.txt

Dabei wird der abgelesene Hashwert in hash.txt eingefügt.

Das schnelle Cracken funktioniert deshalb, weil man bei den 14 Zeichen die beiden ersten durch die Herstellerkennung für NXP ersetzen kann (04). Allerdings haben wir festgestellt, dass auch die beiden letzten Zeichen häufig gleich sind (80). Dadurch geht das Cracken noch einmal deutlich schneller. Selbst mit einer einfachen GeForce GTX 1050 Ti (Neupreis: 150 Euro) lässt sich dann ein Hashwert in maximal 27 Minuten herausfinden.

Der Aufruf wäre dann:
./hashcat -m 1499 -a 3 hash.txt 04?H?H?H?H?H?H?H?H?H?H80 -o crack.txt

Friedhelm Greis/Golem.de