Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

CC Cryptology II

Zentralstrasse 9, CH-6002 Luzern
www.hslu.ch/wirtschaft

Institut für Wirtschaftsinformatik IWI
Oliver Hirschi
Projektleiter

T direkt +41 41 228 41 75
oliver.hirschi@hslu.ch

Projektmitarbeiter:

- Kurmann Dominic HSLU-W
- Schupp Dominik HSLU-W
- Tuccillo Maurizio HSLU-W

Luzern, 22. Juli 2013

Abschlussbericht

Inhaltsverzeichnis

1 Einleitung / Summary

Zum Schutz von digitalen Daten vor missbräuchlicher Nutzung werden heute vermehrt einfach zu bedienende und dennoch qualitativ hochwertige Verschlüsselungstechniken eingesetzt. Ein Nachteil davon ist aber, dass bei Verlust des Schlüsselmaterials, die in den Daten abgelegten Informationen unwiederbringlich verloren oder dann nur durch aufwändige Datenrettung wiederherstellbar sind. Was für Firmen und Private zu in einem beträchtlichen wirtschaftlichen Schaden führen kann, erweist sich im Bereich der behördlichen Strafuntersuchung, oftmals als grosse Hürde in der Erhebung fallrelevanter Informationen.

In einem ersten Projekt (*Kompetenzaufbau CC Cryptology I*) hat das IWI die Kompetenzen für die Arbeit in dieser Thematik aufgebaut. Im Zentrum standen zum einen die Abklärungen der Bedarfslage nach Leistungen im Bereich der Kryptologie für behördliche Ermittlungsstellen, zum andern aber auch die technischen Möglichkeiten der Umsetzung.

1.1 Motivation / Ziele

Das Folgeprojekt *CC Cryptolog II* wurde zwischen Oktober 2012 und Juli 2013 bearbeitet. Es zielte darauf ab, das bisher erarbeitete Wissen zu konsolidieren und in eine operative Einheit zu überführen. Die aufgebaute Infrastruktur sollte dann einerseits „intern“ für Forschungsprojekte oder auch für die Weiterentwicklung von Studiengängen genutzt werden. Andererseits war das Ziel den schweizerischen Ermittlungsbehörden entsprechende kryptologische Dienstleistungen effizient anzubieten und damit die Dienstleistungen im Rahmen der IT-Forensik-Tätigkeit an der HSLU auszubauen.

1.2 Zielerreichung

Zur Zielerreichung können zwiespältige Aussagen gemacht werden. Auf der einen Seite konnte die Infrastruktur erfolgreich aufgebaut und in Betrieb genommen werden. Darüber hinaus wurden verschiedene Testläufe der Infrastruktur durchgeführt. Zwar nicht wie geplant im Rahmen einer Pilot-Phase für hoheitliche Stellen, sondern mit Benutzerlogin-Passwortanalysen für drei verschiedene Unternehmen (NTLM) und des Internetdienstes LinkedIn (SHA1).

Auf der anderen Seite hatte der strategische Entscheid des Institutes für Wirtschaftsinformatik an der Hochschule Luzern, die Beendigung der IT-Forensik- Tätigkeit[1], selbstsprechend einen grossen Einfluss auf die Zielerreichung „Ausbau Dienstleistungsangebot IT-Forensik-Tätigkeit“, welche damit nämlich hinfällig wurde.

[1] Die Hochschule Luzern – Wirtschaft hat sich in den vergangenen Jahren insbesondere für behördliche Stellen zu einem vertrauenswürdigen und zuverlässigen Ansprechpartner im Bereich der IT-Forensik etabliert. Dennoch wurde es in einem wirtschaftlich angespannten Umfeld zunehmend schwieriger diesen Bereich kostendeckend zu betreiben. IT-Forensik hätte deshalb aus Kostengründen nicht wie bisher weitergeführt werden können. Es wäre ein massiver Umbau nötig gewesen, welcher in der bestehenden Organisation nicht möglich war. Deshalb muss sich die Hochschule Luzern vom Themenfeld IT-Forensik verabschieden.

1.3 Erkenntnisse

Die bisherigen Versuche haben gezeigt, dass mit der neuen Infrastruktur eine beachtliche Performance beim Cracken von diversen Hashwerten erreicht werden kann. Passwörter (z.B. NTLM-Hashes) bis zu einer Länge von 8 Zeichen stellen kein Problem dar. Danach kann es sich lohnen, einen optimierten Zeichensatz zu verwenden, da viele Sonderzeichen mit einer sehr geringen Wahrscheinlichkeit vorkommen.

Durch die Leistungsfähigkeit aktueller Grafikkarten können alle achtstelligen Passwörter (NTLM-Hashes) mit sehr geringem finanziellem Aufwand innerhalb kurzer Zeit wiederhergestellt werden. Daher reichen achtstellige Passwörter nicht mehr aus, um eine ausreichende Sicherheit zu gewährleisten. Es empfiehlt sich eine minimale Passwortlänge von 10 Zeichen festzulegen.

Schwierig wird es, wenn der Hashwert mittels PBKDF2, bcrypt oder SHA3 erzeugt wurde. Selbst eine Steigerung der Performance um den Faktor 100 würde keine spürbare Verbesserung herbeiführen.

1.4 Ausblick

Aufgrund der Aufgabe der IT-Forensik-Tätigkeit am Institut für Wirtschaftsinformatik der Hochschule Luzern – Wirtschaft ist die weitere Verwendung der Resultate und insbesondere auch der aufgebauten Infrastruktur ungewiss.

2 Aufbau der Infrastruktur

Die erste Phase wurde dem Aufbau der Infrastruktur gewidmet. Das heisst insbesondere wurden die aufgrund des Projektes *CC Cryptology I* ausgewählten Grafikkarten beschafft und in die Laborinfrastruktur integriert. Um einen möglichst schnellen Wechsel zwischen der „normalen" Labor-Tätigkeit und der Verwendung der Infrastruktur im Rahmen CC Cryptology zu ermöglichen wurde ein speziell für das *CC Cryptology* vorgesehenes Harddisk-Set konfiguriert und mit den nötigen Software-Komponenten ausgerüstet.

- Hardwareseitig wurden 21 Grafikkarten des Typs *nVidia GTX-570* beschafft und in die Laborrechner eingebaut.
- Softwareseitig wurden die beiden kommerziellen Tools *ElcomSoft Distributed Password Recovery* und *Passware Kit Forensic* sowie das frei verfügbare *Hashcat* beschafft und installiert.

2.1 Leistungsausweis im Bereich LM / NTLM Hash

Mit einer einzelnen Grafikkarte (*nVidia GTX-570*) können pro Sekunde 1,8 Mrd. Hashwerte berechnet werden. Diese Leistungsangabe bezieht sich auf den verwendeten Hashing-Algorithmus (NTLM). Bei einem definierten Zeichensatz von 100 Zeichen[2] gibt es bei einer Passwortlänge von 8 Zeichen insgesamt 1e+16 (100^8) verschiedene Passwort-Varianten. Mit einer Infrastruktur bestehend aus 20 Maschinen (Investitionskosten für Grafikkarten < 10kCHF) lassen sich somit alle achtstelligen Passwörter innerhalb von 3.2 Tagen knacken. Mit jeder zusätzlichen Stelle steigt der Aufwand um den Faktor 100 an.

[2] Der vollständige druckbare Zeichensatz umfasst 26 Kleinbuchstaben, 26 Grossbuchstaben, 10 Ziffern, 33 internationale Sonderzeichen, sowie 7 deutschsprachige Sonderzeichen (Umlaute und scharfes S)

3 Werkzeuge

3.1 ElcomSoft / Passware

ElcomSoft Distributed Password Recovery und *Passware Kit Forensic* legen den Fokus vor allem auf die Wiederherstellung von passwortgeschützten Dateiformaten wie z.B. Microsoft Office oder Adobe Acrobat, aber auch System Passwörter von Mac, Windows und Unix.

Im Test zeigten sich gute Performance Werte, unter der Voraussetzung, dass der zu knackende Schlüssel auf der GPU berechnet werden kann. Das Skalieren von CPU's hat im Umfeld der Wiederherstellung von Passwörtern wenig Effekt auf das Endergebnis. Die Performance hängt natürlich auch massgeblich vom Hash-Algorithmus des zu knackenden Dateiformates ab. Bei der aktuellen Office Palette wird das Passwort mehrere Tausend Mal gehasht, was das Cracken enorm erschwert und die Erfolgsaussichten mindert.

Der Konfigurationsaufwand der beiden Tools beschränkt sich auf ein Minimum. Auf den Clients muss lediglich der Agent installiert werden und den Hostnamen des Servers spezifiziert werden. Auf dem Server kann danach die zu crackende Datei geladen werden. Der Rest ist selbsterklärend.

3.2 Hashcat

Hashcat (http://hashcat.net) gilt als eines der schnellsten Tools für das Cracken von Hashwerten wie MD5 oder WPA / WPA2 auf der GPU und unterstützt nebst Brute-Force- & Wörterbuch-Attacken auch die Verwendung von sogenannten Mask-Attacken, regelbasierte Angriffe und weitere Methoden, die es noch zu testen gilt.

3.2.1 Brute-Force- & Mask-Attack

Bei der Mask-Attacke wird für jedes mögliche Zeichen den Zeichensatz definiert. Soll ein Passwort mit der Länge 8 gecrackt werden und alle Sonderzeichen ohne Umlaute miteinbezogen werden, resultiert daraus die folgende Maske: ?a?a?a?a?a?a?a?a. ?a steht für die Zeichensätze ?l, ?u, ?d & ?s.

Bei dieser Maske wird effektiv eine Brute-Force Attacke ausgeführt, da für jedes Zeichen alle Kombinationen im definierten Zeichensatz durchprobiert werden.

Abkürzung	Zeichensatz
?l	abcdefghijklmnopqrstuvwxyz
?u	ABCDEFGHIJKLMNOPQRSTUVWXYZ
?d	0123456789
?s	!"#$%&'()*+,-./:;⇔?@[\]^_`{|}~
?a	?l?u?d?s
?h	8 bit characters from 0xc0 - 0xff
?D	8 bit characters from german alphabet (äöüÄÖÜß)
?F	8 bit characters from french alphabet
?R	8 bit characters from russian alphabet

Tabelle 1: Hashcat Charsets

Die Länge des Zeichensatzes beträgt somit (2 * 26) + 10 + 31 = 93. Die Anzahl Möglichkeiten bei einer Länge von 8 sind 93^8 (5.59581809665e+15). Bei einer durchschnittlichen Leistung von 500 Mio. Hashwerten pro Sekunde dauert der Vorgang max. 129.5 Tage. Bei dieser Berechnung wird die Zeit für die Versuche der Länge 1 – 7 ausser Acht gelassen. Ansonsten lautet die Berechnung wie folgt, wobei n die maximale Länge des Passwortes und Z die Anzahl Zeichen im Zeichensatz sind.

$$\sum_{k=1}^{n} Z^k$$

Wenn die Struktur des Passwortes bekannt ist, kann der Aufwand für die Wiederherstellung des Kennwortes erheblich reduziert werden. Angenommen das zu crackende Passwort lautet Julia1984. Die passende Maske dazu lautet ?lu?l?l?l?l?d?d?d?d. Durch diese Anpassung verkleinert sich der Keyspace auf 52*26*26*26*26*10*10*10*10 (237.627.520.000) Kombinationen. Bei 500 Mio. Hashwerten pro Sekunde würde der Vorgang nun noch 8 Minuten dauern.

3.2.2 Job-Distribution

Wie bei Passware oder Elcomsoft kann der Job auch auf mehrere Maschinen verteilt werden. Das Tool funktioniert jedoch nicht auf die gleiche Weise mittels einer Client-/Server-Architektur. Für die einzelnen Maschinen müssen die Parameter spezifisch gesetzt werden.

Im folgenden Beispiel versuchen wir ein Passwort mit einer möglichen Länge von 1 bis 8 Zeichen und einem Zeichensatz, bestehend aus Gross-, Kleinschreibung & Zahlen ohne Sonderzeichen (2*26 + 10 = 62) zu cracken. Es stehen drei Maschinen für den Auftrag zur Verfügung. Der Zeichensatz wird auf die verschiedenen Maschinen verteilt.

Machine	Full Charset	Split Charset	Mask	Example Start	Example End
1	?l?u?d	abcdefghijklmnopqrstu	?2	a	u
			?1?2	aa	9u
			?1?1?2	aaa	99u
			?1?1?1?2	aaaa	999u
			?1?1?1?1?2	aaaaa	9999u
			?1?1?1?1?1?2	aaaaaa	99999u
			?1?1?1?1?1?1?2	aaaaaaa	999999u
			?1?1?1?1?1?1?1?2	aaaaaaaa	9999999u
2	?l?u?d	vwxyzABCDEFGHIJKLMNOP	?2	v	P
			?1?2	av	9P
			?1?1?2	aav	99P
			?1?1?1?2	aaav	999P
			?1?1?1?1?2	aaaav	9999P
			?1?1?1?1?1?2	aaaaav	99999P
			?1?1?1?1?1?1?2	aaaaaav	999999P
			?1?1?1?1?1?1?1?2	aaaaaaav	9999999P
3	?l?u?d	QRSTUVWXYZ0123456789	?2	Q	9
			?1?2	aQ	99
			?1?1?2	aaQ	999
			?1?1?1?2	aaaQ	9999
			?1?1?1?1?2	aaaaQ	99999
			?1?1?1?1?1?2	aaaaaQ	999999
			?1?1?1?1?1?1?2	aaaaaaQ	9999999
			?1?1?1?1?1?1?1?2	aaaaaaaQ	99999999

Tabelle 2: Parameter Jobverteilung

Bei der Maske wird definiert, an welcher Stelle welcher Zeichensatz verwendet wird. ?1 steht hier-bei für den ganzen Zeichensatz (Full Charset), ?2 für den maschinenspezifischen Teil des Zeichen-satzes (Split Charset). Somit wird sichergestellt, dass alle Kombinationen auf allen Maschinen durchprobiert werden.

Damit die Kommandozeilenparameter nicht manuell erzeugt werden müssen, behelfen wir uns mit einem kleinen Tool, welches ein Batchfile für die verschiedenen Maschinen erstellt. Die Ausgabe besteht unter anderem aus den Parametern Charset 1 & 2, Maske und weitere Pfadangaben für die Liste mit den Hashwerten und dem Binary.

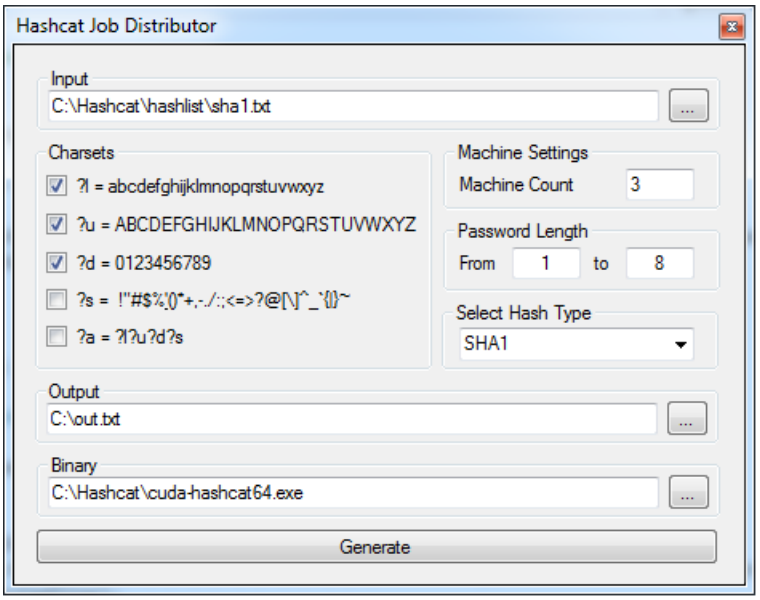


Abbildung 1: Hashcat Job Distributor

Hashcat liegt in unterschiedlichen Versionen vor. Wenn mehrere Hashwerte miteinander gecrackt wer-den sollen, eignet sich oclHashcat-plus. Wird der Auftrag einmal gestartet, kann er auch pausiert werden. Hashcat führt dazu eine Datei mit Statusinformationen namens cudaHashcat-plus.restore welche auch als Parameter übergeben werden kann um einen pausierten Auftrag fortzusetzen.

Bei einem einzelnen Hash wird oclHashcat-lite verwendet. Bei dieser Version besteht zudem die Mög-lichkeit, den Auftrag mit einem bestimmten Offset zu starten.

4 Testlauf der Infrastruktur

Nach der Installation der neuen Infrastruktur für das Cracken von Passwörtern galt es einen ersten Testlauf durchzuführen. Nachfolgend werden die Erkenntnisse und Ergebnisse der ersten Tests vorgestellt.

4.1 LinkedIn

Im Sommer 2012 tauchte im Internet ein Textfile mit rund 6.5 Mio. Einträgen auf, die sich als die SHA1 Passwort-Hashes von LinkedIn Benutzern herausstellte. Das Textfile beinhaltet lediglich das gehashte Passwort, alle anderen Daten wurden nicht veröffentlicht. Bei rund 3.5 Mio. Hashwerten wurden die ersten fünf Zeichen durch 00000 ersetzt. Dabei handelt es sich vermutlich um Hashwerte, die der Hacker mit einer Wörterbuch-Attacke in einer ersten Runde gleich wiederherstellen konnte. Übrig blieben 3 Mio. vollwertige Hashwerte, welche die ideale Basis für einen umfangreichen Performance-Test der neuen Infrastruktur darstellten.

Für einen solchen umfangreichen Test eignet sich am besten Hashcat, da es problemlos mit grossen Dateien, welche viele Hashwerte beinhalten, umgehen kann. Zum Zeichensatz (Full Charset) zählten wir nebst den Sonderzeichen auch die deutschen Umlaute und das scharfe S (ß), um möglichst viele Passwörter abzudecken. Pro GPU erreichten wir eine Rate von rund 500 Mio. Hashwerten pro Sekunde. Den Auftrag verteilten wir auf 20 Maschinen, die konstant am Rechnen waren. Nach knapp 2 Wochen hatten wir 889‘409 Passwörter bis und mit der Länge 8 mittels Brute-Force wiederhergestellt. Eine detaillierte Analyse der wiederhergestellten LinkedIn Passwörter befindet sich im Anhang A.

4.2 LM / NTLM Hash

Auf einem Windows System konnten wir den LM & NTLM Hash des Administrators extrahieren. Aufgrund der schwachen Implementierung des LM-Hashes, welche Passwörter länger als 7 Zeichen in zwei Stücke teilt und diese Stücke separat hasht, kann dieser einfach geknackt werden. Jede Hälfte des Passwortes kann daher einzeln angegriffen werden. In diesem Fall konnten wir nach ein paar Sekunden den zweiten Teil des Passwortes, der aus zwei Zeichen bestand, wiederherstellen. Daraus resultierte die Erkenntnis, dass es sich um ein Passwort mit einer Gesamtlänge von 9 Zeichen handeln muss.

In einem nächsten Schritt führten wir mittels ElcomSoft eine Brute-Force Attacke auf den NTLM-Hash aus. Wir wählten den Angriff auf den NTLM-Hash, da dieser im Gegensatz zum LM-Hash auf der GPU berechnet werden kann und daher eine viel bessere Performance bietet. Durch die letzten zwei bekannten Zeichen reduziert sich die Anzahl der möglichen Varianten erheblich. Somit konnten wir auf 4 GPU-Kernen das Passwort innerhalb von 200 Minuten knacken. Hätten wir unsere gesamte Infrastruktur hochgefahren, wären wir in 40 Minuten durch gewesen.

4.2.1 Passwort-Analyse von Benutzer-Accounts drei ausgewählter Unternehmungen

Eine weitere Belastungs-Testreihe der Infrastruktur wurde in Zusammenarbeit mit drei Unternehmen durchgeführt. Dabei ging es darum die Passwörter der Benutzer-Logins zu brechen. Aus den von den drei

Unternehmen zur Verfügung gestellten 35‘846 Passwort-Hashwerten (NTLM) konnten 19‘103 Passwörter wiederhergestellt werden, was einer Quote von rund 53.3% entspricht. Alle Passwörter bis acht Zeichen wurden mittels einer Brute-Force Attacke (bei vollständigem druckbaren Zeichensatz)[3] geknackt. Bei den neun- & zehnstelligen Passwörtern wurde nach Mustern innerhalb des Passwortes gesucht, die durch die Erkenntnisse der Brute-Force Attacke gewonnen wurden.

4.2.1.1 Detaillierte Auswertung

Die nachfolgenden Angaben machen eine Aussage über den Inhalt und die Struktur der gefundenen Passwörter.

Top 10 Grundwörter

Anzahl Passwörter, in welchen die angegebene Zeichenfolge unverändert vorkommt.

******[4] = 77 (0.4%)
sommer = 36 (0.19%)
******[4] = 34 (0.18%)
******[4] = 21 (0.11%)
hallo = 18 (0.09%)
******[4] = 17 (0.09%)
lchx = 16 (0.08%)
******[4] = 16 (0.08%)
winter = 16 (0.08%)
andrea = 14 (0.07%)

Passwortlänge (Sortiert nach Länge)

Anzahl Passwörter, welche die angegebene Länge aufweisen.

5 = 21 (0.11%)
6 = 197 (1.03%)
7 = 213 (1.12%)
8 = 15124 (79.32%)
9 = 2826 (14.82%)
10 = 685 (3.59%)
11 = 2 (0.01%)

1 bis 6 Zeichen = 218 (1.14%)
1 bis 8 Zeichen = 15555 (81.58%)
Mehr als 8 Zeichen = 3513 (18.42%)

[3] Der vollständige druckbare Zeichensatz umfasst 26 Kleinbuchstaben, 26 Grossbuchstaben, 10 Ziffern, 33 internationale Sonderzeichen, sowie 7 deutschsprachige Sonderzeichen (Umlaute und scharfes S)

[4] Aus Datenschutz- und Geheimhaltungsgründen wurden in obiger Liste fünf Einträge (******) maskiert, d.h. auf die Veröffentlichung an dieser Stelle wird verzichtet. Es darf aber angemerkt werden, dass sowohl Unternehmensstandortangaben als auch Unternehmensabkürzungen bevorzugt verwendet werden. Ebenfalls häufigere Verwendung finden Bezeichnungen von Wahrzeichen der Umgebung

Jahreszahlen (Top 10)

Anzahl Passwörter, welche die angegebene Jahreszahl aufweisen.

2013 = 118 (0.62%)
2012 = 89 (0.47%)
2010 = 58 (0.3%)
2011 = 52 (0.27%)
2000 = 44 (0.23%)
2008 = 40 (0.21%)
1988 = 40 (0.21%)
2009 = 39 (0.2%)
1991 = 38 (0.2%)
1990 = 38 (0.2%)

Zeichenfolgen

Anzahl Passwörter, welche das angegebene Zeichen in der letzten Position aufweisen.

1 = 1269 (6.66%)
3 = 1113 (5.84%)
2 = 1038 (5.44%)
9 = 768 (4.03%)
8 = 761 (3.99%)
7 = 759 (3.98%)
4 = 725 (3.8%)
5 = 717 (3.76%)
6 = 690 (3.62%)
0 = 657 (3.45%)

Anzahl Passwörter, welche die angegebenen Zeichen in den letzten zwei Positionen aufweisen. (Top 10)

12 = 290 (1.52%)
13 = 275 (1.44%)
23 = 245 (1.28%)
11 = 230 (1.21%)
10 = 178 (0.93%)
88 = 170 (0.89%)
00 = 133 (0.7%)
89 = 117 (0.61%)
01 = 114 (0.6%)
90 = 113 (0.59%)

Anzahl Passwörter, welche die angegebenen Zeichen in den letzten drei Positionen aufweisen. (Top 10)

123 = 183 (0.96%)
013 = 109 (0.57%)
012 = 77 (0.4%)
000 = 73 (0.38%)

010 = 63 (0.33%)
234 = 62 (0.33%)
011 = 57 (0.3%)
988 = 39 (0.2%)
009 = 38 (0.2%)
008 = 38 (0.2%)

Anzahl Passwörter, welche die angegebenen Zeichen in den letzten vier Positionen aufweisen. (Top 10)

2013 = 106 (0.56%)
2012 = 75 (0.39%)
1234 = 57 (0.3%)
2010 = 53 (0.28%)
2011 = 50 (0.26%)
1988 = 39 (0.2%)
2000 = 39 (0.2%)
2008 = 36 (0.19%)
1990 = 35 (0.18%)
1991 = 35 (0.18%)

Anzahl Passwörter, welche die angegebenen Zeichen in den letzten fünf Positionen aufweisen. (Top 10)

23456 = 12 (0.06%)
12345 = 7 (0.04%)
05055 = 2 (0.01%)
31990 = 2 (0.01%)
62010 = 2 (0.01%)
56789 = 2 (0.01%)
66666 = 2 (0.01%)
21991 = 2 (0.01%)
14477 = 1 (0.01%)
06002 = 1 (0.01%)

Einzelne Ziffer am Ende = 2702 (14.17%)
Zwei Ziffern am Ende = 2903 (15.22%)
Drei Ziffern am Ende = 658 (3.45%)

Zeichensätze

Hier werden die in den Passwörtern verwendeten Zeichensätze aufgeschlüsselt.

Charset	Beschreibung	Resultat
mixedalphanum	Gross- & Kleinschreibung, Ziffern	9270 (48.62%)
loweralphanum	Kleinschreibung, Ziffern	2212 (11.6%)
upperalphaspecialnum	Grossschreibung, Sonderzeichen, Ziffern	1952 (10.24%)
mixedalphaspecialnum	Gross- & Kleinschreibung, Sonderzeichen, Ziffern	1881 (9.86%)
mixedalpha	Gross- & Kleinschreibung	1654 (8.67%)
mixedalphaspecial	Gross- & Kleinschreibung, Sonderzeichen	599 (3.14%)
loweralpha	Kleinschreibung	524 (2.75%)
loweralphaspecialnum	Kleinschreibung, Sonderzeichen, Ziffern	412 (2.16%)
upperalphaspecial	Grossschreibung, Sonderzeichen	293 (1.54%)

upperalphanum	Grossschreibung, Ziffern	112 (0.59%)
Numeric	Ziffern	73 (0.38%)
loweralphaspecial	Kleinschreibung, Sonderzeichen	53 (0.28%)
upperalpha	Grossbuchstaben	13 (0.07%)
specialnum	Sonderzeichen, ZIffern	5 (0.03%)

Tabelle 3: Verwendete Zeichensätze

Nur Kleinbuchstaben = 524 (2.75%)
Nur Grossbuchstaben = 13 (0.07%)
Nur Alphanumerische Zeichen = 537 (2.82%)
Nur Numerische Zeichen = 73 (0.38%)

Zeichensätze sortiert

Hier wird die Reihenfolge der verwendeten Zeichensätze dargestellt. „stringdigit“ bedeutet z.B., dass das Passwort aus einer beliebigen Anzahl von Gross- & Kleinbuchstaben besteht, gefolgt von einer beliebigen Anzahl von Zahlen. Es wurde versucht, alle Passwörter einer Kategorie zuzuordnen. Alle Passwörter, die keiner Kategorie zugeordnet werden konnten, wurden als „othermask“ kategorisiert.

othermask: 6222 (32.63%)
stringdigit: 6001 (31.47%)
stringdigitstring: 2635 (13.82%)
allstring: 2191 (11.49%)
digitstring: 732 (3.84%)
stringspecialstring: 486 (2.55%)
stringspecialdigit: 333 (1.75%)
digitstringdigit: 276 (1.45%)
alldigit: 73 (0.38%)
specialstring: 60 (0.31%)
stringspecial: 44 (0.23%)
specialstringspecial: 15 (0.08%)

Häufigkeitsverteilung & Rechenaufwand

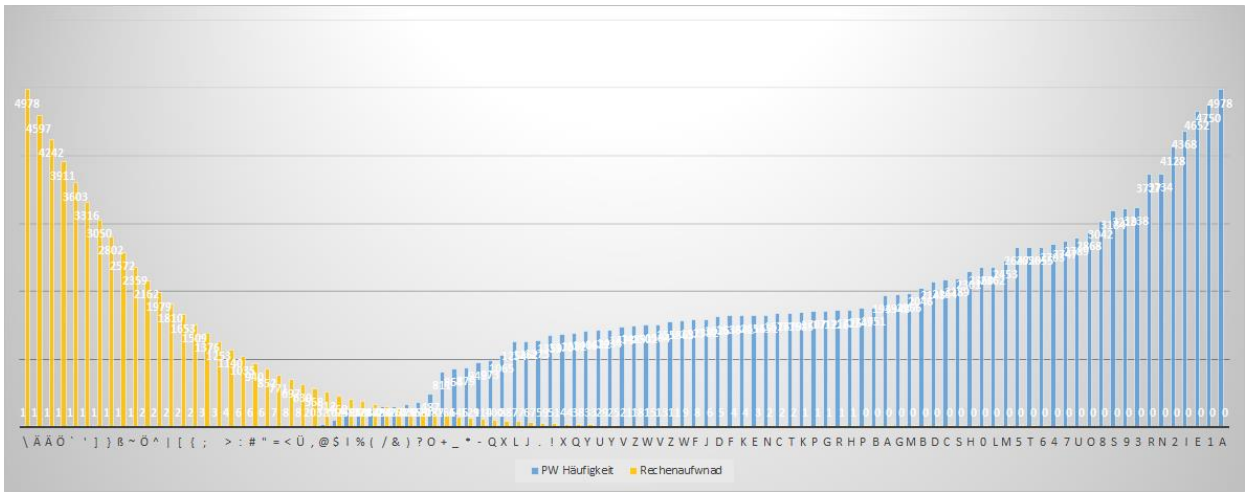


Abbildung 2: Häufigkeitsverteilung & Rechenaufwand

Das Diagramm zeigt mittels der blauen Balken, in wie vielen Passwörtern das bestimmte Zeichen auf der Achse verwendet wird. Es werden also die verwendeten Zeichen nach zunehmender Häufigkeit dargestellt. Die gelben Balken zeigen, wie sich der Rechenaufwand mit Auslassen eines Zeichens reduziert. Die Reihenfolge der Zeichen auf der Achse ist dabei nicht wichtig, sondern können beliebig angeordnet werden. Ausschlaggebend ist, dass sich der Aufwand mit jedem weggelassenen Zeichen massiv reduziert.

4.3 OS X – SHA512 PBKDF2

Bei einem sichergestellten Macbook mit OS X 10.8.2 (Mountain Lion) haben wir eine Datei mit dem Login des Benutzers unter /private/var/db/dslocal/nodes/Default/users/[username].plist extrahiert. Diese Datei enthält nebst der Bezeichnung des verwendeten Algorithmus auch der Hash und das dazugehörigen Salt.

Aktuell ist Passware die einzige Applikation, die mit diesem Hash umgehen kann. Leider besteht keine Unterstützung für die Berechnung auf der GPU.

Aufgrund der speziellen Beschaffenheit der verwendeten PBKDF2 Funktion, die extra darauf ausgelegt ist langsam zu sein, und der Verzicht der GPU, erreichten wir ca. 20-30 Berechnungen pro Sekunde. Selbst bei einer möglichen GPU-Unterstützung wäre die Performance immer noch viel zu niedrig, um eine effektive Brute-Force Attacke auszuführen.

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang

A Analyse der wiederhergestellten LinkedIn Passwörter mittels Pipal

Total entries = 889408
Total unique entries = 889407

Top 10 passwords
SciNet = 2 (0.0%)
EScislow = 1 (0.0%)
zaleski1 = 1 (0.0%)
Eamonn = 1 (0.0%)
07271947 = 1 (0.0%)
71pontiac = 1 (0.0%)
alpine1951 = 1 (0.0%)
ferrari2 = 1 (0.0%)
jackass21 = 1 (0.0%)
loyola2324 = 1 (0.0%)

Top 10 base words
link = 1332 (0.15%)
linked = 1304 (0.15%)
pass = 460 (0.05%)
golf = 315 (0.04%)
tiger = 290 (0.03%)
soccer = 248 (0.03%)
welcome = 243 (0.03%)
success = 238 (0.03%)
abcd = 235 (0.03%)
mustang = 233 (0.03%)

Password length (length ordered)
2 = 8 (0.0%)
6 = 36825 (4.14%)
7 = 73166 (8.23%)
8 = 722300 (81.21%)
9 = 28938 (3.25%)
10 = 18141 (2.04%)
11 = 6543 (0.74%)
12 = 2455 (0.28%)
13 = 713 (0.08%)
14 = 244 (0.03%)
15 = 58 (0.01%)
16 = 17 (0.0%)

Password length (count ordered)
8 = 722300 (81.21%)
7 = 73166 (8.23%)

6 = 36825 (4.14%)
9 = 28938 (3.25%)
10 = 18141 (2.04%)
11 = 6543 (0.74%)
12 = 2455 (0.28%)
13 = 713 (0.08%)
14 = 244 (0.03%)
15 = 58 (0.01%)
16 = 17 (0.0%)
2 = 8 (0.0%)

000000000011111111
012345678901234567

One to six characters = 36833 (4.14%)
One to eight characters = 832299 (93.58%)
More than eight characters = 57109 (6.42%)

Only lowercase alpha = 47268 (5.31%)
Only uppercase alpha = 3106 (0.35%)
Only alpha = 50374 (5.66%)
Only numeric = 13126 (1.48%)

First capital last symbol = 22311 (2.51%)
First capital last number = 141174 (15.87%)

Months
january = 46 (0.01%)
february = 14 (0.0%)
march = 88 (0.01%)
april = 20 (0.0%)
may = 1114 (0.13%)
june = 215 (0.02%)
july = 152 (0.02%)
august = 26 (0.0%)
september = 127 (0.01%)
october = 23 (0.0%)

november = 46 (0.01%)
december = 165 (0.02%)

Days
monday = 19 (0.0%)
tuesday = 7 (0.0%)
thursday = 30 (0.0%)
friday = 4 (0.0%)
saturday = 23 (0.0%)
sunday = 63 (0.01%)

Months (Abreviated)
jan = 1467 (0.16%)
feb = 390 (0.04%)
mar = 4244 (0.48%)
apr = 569 (0.06%)
may = 1114 (0.13%)
jun = 1153 (0.13%)
jul = 898 (0.1%)
aug = 676 (0.08%)
sept = 222 (0.02%)
oct = 584 (0.07%)
nov = 916 (0.1%)
dec = 789 (0.09%)

Days (Abreviated)
mon = 2073 (0.23%)
tues = 25 (0.0%)
wed = 440 (0.05%)
thurs = 31 (0.0%)
fri = 572 (0.06%)
sat = 626 (0.07%)
sun = 1472 (0.17%)

Includes years
1975 = 652 (0.07%)
1976 = 627 (0.07%)
1977 = 636 (0.07%)
1978 = 684 (0.08%)
1979 = 669 (0.08%)
1980 = 769 (0.09%)
1981 = 685 (0.08%)
1982 = 670 (0.08%)
1983 = 618 (0.07%)
1984 = 640 (0.07%)
1985 = 495 (0.06%)
1986 = 391 (0.04%)
1987 = 316 (0.04%)
1988 = 240 (0.03%)
1989 = 229 (0.03%)
1990 = 204 (0.02%)
1991 = 197 (0.02%)
1992 = 157 (0.02%)

1993 = 173 (0.02%)
1994 = 195 (0.02%)
1995 = 220 (0.02%)
1996 = 257 (0.03%)
1997 = 263 (0.03%)
1998 = 317 (0.04%)
1999 = 348 (0.04%)
2000 = 1010 (0.11%)
2001 = 658 (0.07%)
2002 = 567 (0.06%)
2003 = 574 (0.06%)
2004 = 662 (0.07%)
2005 = 698 (0.08%)
2006 = 790 (0.09%)
2007 = 898 (0.1%)
2008 = 1167 (0.13%)
2009 = 755 (0.08%)
2010 = 952 (0.11%)
2011 = 1256 (0.14%)
2012 = 374 (0.04%)
2013 = 49 (0.01%)
2014 = 28 (0.0%)
2015 = 57 (0.01%)
2016 = 35 (0.0%)
2017 = 34 (0.0%)
2018 = 25 (0.0%)
2019 = 136 (0.02%)
2020 = 175 (0.02%)

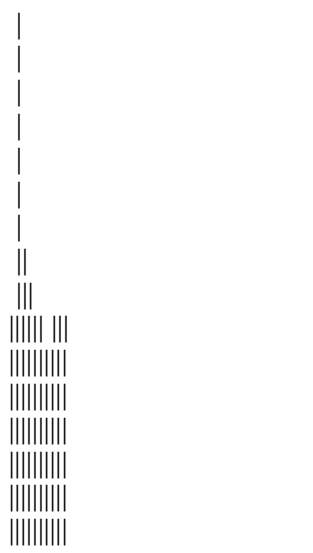
Years (Top 10)
2011 = 1256 (0.14%)
2008 = 1167 (0.13%)
2000 = 1010 (0.11%)
2010 = 952 (0.11%)
2007 = 898 (0.1%)
2006 = 790 (0.09%)
1980 = 769 (0.09%)
2009 = 755 (0.08%)
2005 = 698 (0.08%)
1981 = 685 (0.08%)

Colours
black = 119 (0.01%)
blue = 603 (0.07%)
brown = 173 (0.02%)
gray = 133 (0.01%)
green = 158 (0.02%)
orange = 7 (0.0%)
pink = 275 (0.03%)
purple = 8 (0.0%)
red = 2109 (0.24%)
white = 65 (0.01%)
yellow = 32 (0.0%)

violet = 20 (0.0%)
indigo = 95 (0.01%)

Single digit on the end = 149369 (16.79%)
Two digits on the end = 165080 (18.56%)
Three digits on the end = 43805 (4.93%)

Last number
0 = 40205 (4.52%)
1 = 92297 (10.38%)
2 = 51361 (5.77%)
3 = 45847 (5.15%)
4 = 36663 (4.12%)
5 = 35228 (3.96%)
6 = 31016 (3.49%)
7 = 37785 (4.25%)
8 = 35944 (4.04%)
9 = 36247 (4.08%)


0123456789

Last digit
1 = 92297 (10.38%)
2 = 51361 (5.77%)
3 = 45847 (5.15%)
0 = 40205 (4.52%)
7 = 37785 (4.25%)
4 = 36663 (4.12%)
9 = 36247 (4.08%)
8 = 35944 (4.04%)
5 = 35228 (3.96%)
6 = 31016 (3.49%)

Last 2 digits (Top 10)
01 = 13962 (1.57%)
11 = 10868 (1.22%)
12 = 7954 (0.89%)

00 = 7558 (0.85%)
23 = 7220 (0.81%)
10 = 7135 (0.8%)
08 = 5980 (0.67%)
07 = 5735 (0.64%)
99 = 5270 (0.59%)
22 = 5106 (0.57%)

Last 3 digits (Top 10)
123 = 3192 (0.36%)
001 = 2017 (0.23%)
007 = 1882 (0.21%)
000 = 1708 (0.19%)
234 = 1652 (0.19%)
008 = 1234 (0.14%)
011 = 1221 (0.14%)
010 = 1103 (0.12%)
111 = 1044 (0.12%)
100 = 946 (0.11%)

Last 4 digits (Top 10)
1234 = 1351 (0.15%)
2008 = 1003 (0.11%)
2011 = 969 (0.11%)
2000 = 868 (0.1%)
2010 = 804 (0.09%)
2007 = 770 (0.09%)
1980 = 699 (0.08%)
2006 = 688 (0.08%)
2009 = 641 (0.07%)
1978 = 629 (0.07%)

Last 5 digits (Top 10)
12345 = 102 (0.01%)
11982 = 56 (0.01%)
11980 = 55 (0.01%)
11978 = 54 (0.01%)
11981 = 54 (0.01%)
21981 = 53 (0.01%)
20000 = 52 (0.01%)
21982 = 52 (0.01%)
21980 = 52 (0.01%)
11979 = 51 (0.01%)

US Area Codes
234 = NE Ohio: Canton, Akron (OH)

Character sets
loweralphanum: 358440 (40.3%)
mixedalphanum: 255940 (28.78%)
mixedalphaspecialnum: 73154 (8.23%)
loweralphaspecialnum: 59580 (6.7%)
loweralpha: 47268 (5.31%)

mixedalpha: 26888 (3.02%)
upperalphanum: 16015 (1.8%)
loweralphaspecial: 14318 (1.61%)
numeric: 13126 (1.48%)
mixedalphaspecial: 11262 (1.27%)
upperalphaspecialnum: 6272 (0.71%)
upperalpha: 3106 (0.35%)
specialnum: 1311 (0.15%)
upperalphaspecial: 1046 (0.12%)
special: 168 (0.02%)

Character set ordering
stringdigit: 264533 (29.74%)
othermask: 251291 (28.25%)
stringdigitstring: 175154 (19.69%)
allstring: 77262 (8.69%)
stringspecialdigit: 33446 (3.76%)
digitstring: 32519 (3.66%)
digitstringdigit: 19706 (2.22%)
stringspecialstring: 14042 (1.58%)
alldigit: 13126 (1.48%)
stringspecial: 6304 (0.71%)
specialstring: 1115 (0.13%)
specialstringspecial: 742 (0.08%)
allspecial: 168 (0.02%)

Hashcat masks (Top 10)
?l?l?l?l?l?l?d?d: 54997 (6.18%)
?l?l?l?l?l?l?l?l: 39285 (4.42%)
?l?l?l?l?l?l?l?d: 31072 (3.49%)
?l?l?l?d?d?l?l?l: 20224 (2.27%)
?u?l?l?l?l?l?d?d: 14931 (1.68%)
?u?l?l?l?l?l?l?d: 14529 (1.63%)
?l?l?l?l?d?d?d?d: 13136 (1.48%)
?l?l?l?l?l?d?d?d: 13128 (1.48%)
?d?d?d?d?d?d?d?d: 13071 (1.47%)
?l?l?l?l?d?l?l?l: 9443 (1.06%)

Haufigkeitsanalyse von allen Zeichen über alle Positionen

Char	Hex	Total	%	v	0x76	74999	1.06	-	0x2D	12805	0.18
1	0x31	361864	5.12	z	0x7A	55172	0.78	Z	0x5A	12718	0.18
a	0x61	357417	5.06	x	0x78	52821	0.75	X	0x58	11636	0.16
e	0x65	296783	4.20	A	0x41	48514	0.69	_	0x5F	9718	0.14
0	0x30	261485	3.70	M	0x4D	46303	0.66	Q	0x51	9714	0.14
i	0x69	249078	3.52	S	0x53	44777	0.63	&	0x26	8081	0.11
2	0x32	247263	3.50	L	0x4C	43122	0.61	%	0x25	5672	0.08
r	0x72	241378	3.41	!	0x21	38849	0.55	+	0x2B	5074	0.07
n	0x6E	235943	3.34	B	0x42	37954	0.54	,	0x2C	3887	0.05
s	0x73	235909	3.34	T	0x54	36502	0.52	=	0x3D	3629	0.05
o	0x6F	229178	3.24	R	0x52	36166	0.51	/	0x2F	3428	0.05
l	0x6C	208511	2.95	@	0x40	34913	0.49	)	0x29	3309	0.05
t	0x74	204086	2.89	C	0x43	34205	0.48	?	0x3F	2581	0.04
m	0x6D	192585	2.72	D	0x44	33375	0.47	(	0x28	2580	0.04
3	0x33	188479	2.67	P	0x50	32531	0.46	^	0x5E	2304	0.03
4	0x34	162275	2.30	q	0x71	31172	0.44	;	0x3B	2174	0.03
9	0x39	160594	2.27	N	0x4E	30080	0.43		0x20	1627	0.02
d	0x64	154546	2.19	J	0x4A	29418	0.42	:	0x3A	1551	0.02
c	0x63	154455	2.19	E	0x45	29262	0.41	'	0x27	970	0.01
7	0x37	153767	2.18	K	0x4B	29176	0.41	~	0x7E	783	0.01
u	0x75	149485	2.11	I	0x49	27710	0.39	[	0x5B	699	0.01
8	0x38	148083	2.09	G	0x47	27430	0.39	>	0x3E	699	0.01
5	0x35	144450	2.04	H	0x48	26481	0.37	<	0x3C	690	0.01
k	0x6B	137869	1.95	F	0x46	22289	0.32	"	0x22	552	0.01
p	0x70	134197	1.90	$	0x24	19936	0.28	]	0x5D	522	0.01
h	0x68	133782	1.89	W	0x57	19350	0.27	\	0x5C	349	0.00
6	0x36	132891	1.88	O	0x4F	19139	0.27	|	0x7C	337	0.00
b	0x62	132225	1.87	#	0x23	17695	0.25	`	0x60	262	0.00
g	0x67	122438	1.73	U	0x55	17327	0.25	{	0x7B	227	0.00
y	0x79	104350	1.48	V	0x56	16403	0.23	}	0x7D	182	0.00
j	0x6A	95318	1.35	*	0x2A	16240	0.23	Ã	0xC3	2	0.00
w	0x77	85826	1.21	.	0x2E	14342	0.20	©	0xA9	2	0.00
f	0x66	85477	1.21	Y	0x59	14236	0.20				