Biuletyn WAT
Vol. LXVIII, Nr 1, 2019

Wybór strategii łamania hasła przy nałożonych ograniczeniach czasowych

PRZEMYSŁAW RODWALD

Akademia Marynarki Wojennej, Wydział Nawigacji i Uzbrojenia Okrętowego,
Instytut Uzbrojenia Okrętowego i Informatyki, ul. Śmidowicza 69, 81-103 Gdynia
p.rodwald@amw.gdynia.pl

Streszczenie. Celem artykułu jest przedstawienie metodyki postępowania w przypadku, gdy atakujący (biegły sądowy, technik kryminalistyki, pentester) ma za zadanie złamać hasło do pewnego systemu teleinformatycznego przy nałożonych ograniczeniach czasowych. Sytuacja taka ma często miejsce w toku działań procesowych prowadzonych przez organy ścigania, podczas zatrzymań sprzętu komputerowego. Na wstępie przedstawiono obowiązującą wykładnię prawa wraz z dobrymi praktykami przy zabezpieczaniu materiału dowodowego. Następnie omówiono sposoby przechowywania haseł w systemach informatycznych, po czym dokonano przeglądu różnych klas ataków na hasła. Wyszczególniono także najpopularniejsze narzędzia wspomagające proces łamania haseł. W głównej części pracy przedstawiono autorską strategię przeprowadzania ataku przy nałożonych z góry ograniczeniach czasowych dla dostępnych zasobów sprzętowych. Pokazane zostały również szacunkowe koszty i efektywność ekonomiczna dla wybranych rozwiązań. Obliczenia pokazano na przykładzie dwóch rzeczywistych ataków przeprowadzonych przez autora w trakcie prawdziwych działań procesowych.
Słowa kluczowe: hasła, łamanie haseł, funkcje skrótu, informatyka śledcza
DOI: 10.5604/01.3001.0013.1467

1. Wstęp

W XXI wieku przestępstwa dokonywane za pomocą lub przy wykorzystaniu systemów teleinformatycznych są na porządku dziennym. Komputer lub inne urządzenie (np. tablet, smartfon) może być zarówno narzędziem umożliwiającym popełnienie czynu zabronionego (kradzież, rozpowszechnianie pornografii itp.), jak również może mieć znaczenie jedynie incydentalne w trakcie dokonywania

przestępstwa (baza danych do przechowywania wrażliwych informacji itp.). Systemy komputerowe coraz powszechniej zawierają dowody działalności przestępczej. Przestępcy natomiast, coraz bardziej świadomi ilości „niewygodnych" informacji znajdujących się na urządzeniach, które używają, zabezpieczają dostęp do nich. Robią to najczęściej za pomocą haseł[1]. Statyczne hasła, należące do metod zwanych „coś, co wiesz" (ang. *something you know*), mimo swoich niedoskonałości i istnienia innych mechanizmów uwierzytelniających takich jak: karty magnetyczne/inteligentne — „coś, co masz" (ang. *something you have*) czy techniki biometryczne — „coś, czym jesteś" (ang. *something you are*), pozostają wciąż najczęściej stosowaną techniką uwierzytelniania [1]. Za tą popularnością stoi zarówno łatwość stosowania, jak i duża akceptacja społeczna. Z tego powodu zabezpieczany w trakcie prowadzenia czynności dochodzeniowych sprzęt komputerowy (komputery stacjonarne, laptopy, tablety, smartfony) jest coraz częściej zabezpieczony hasłem. Jak pokazują badania [2], 43% organizacji posiada wdrożone strategie szyfrowania danych i trend ten ciągle rośnie.

Organ procesowy staje więc przed zadaniem „dostania" się do informacji zabezpieczonych hasłem. Zabezpieczone mogą być całe dyski lub partycje (ang. *Full Disk Encryption*); zaszyfrowane najczęściej za pomocą narzędzi takich jak: nierozwijany już, ale wciąż często spotykany TrueCrypt [A], wbudowany w systemach operacyjnych Microsoftu BitLocker, czy komercyjny Symantec Drive Encryption [B]. Zaszyfrowane mogą być również poszczególne pliki (ang. *File Encryption*), w szczególności: wybrane foldery/katalogi (na przykład przy użyciu Windows Encrypted File System), poszczególne programy (różnego rodzaju komunikatory, systemy baz danych), czy też konkretne pliki (na przykład najpopularniejszy pakiet biurowy Microsoft Office posiada wbudowaną funkcję szyfrowania dokumentów: Plik > Chroń dokument > Szyfruj przy użyciu hasła).

Należy zauważyć, że organy ścigania nie dysponują w pełni skutecznymi narzędziami przełamania prawidłowo zastosowanego szyfrowania. Także obowiązujące w Polsce prawo nie ułatwia im tego zadania (szczegóły w punkcie 1.2). Dlatego tak ważne jest odpowiednie przygotowanie do zabezpieczenia sprzętu (punkt 1.3), dzięki któremu czasochłonny i nie zawsze skuteczny proces „łamania hasła"[2] może być w ogóle niepotrzebny lub znacząco ułatwiony.

---

[1] Hasło — pojęcie to używane jest w niniejszej pracy w dwóch kontekstach: a) w przypadku dostępu do systemów teleinformatycznych hasło stanowi pewien „sekret", który łącznie z identyfikatorem umożliwia uwierzytelnienie użytkownika; b) w przypadku systemów kryptograficznych pod pojęciem hasła rozumie się klucz kryptograficzny służący do szyfrowania/deszyfrowania informacji.

[2] Łamanie hasła — kolokwializm ten oznacza znalezienie takiego ciągu znaków, który odpowiada poszukiwanemu (łamanemu) hasłu.

1.1. Analiza literatury przedmiotu

Mimo że sama tematyka łamania haseł jest dość obszerna, to jednak większość autorów skupia się na łamaniu wycieków haseł, a więc najczęściej dużych zbiorów skrótów. Blocki i in. [3] proponują na przykład ekonomiczny model pozwalający oszacować liczbę możliwych do złamania haseł dla wycieków ich skrótów, przy założeniu racjonalności[3] działania atakującego. Picolet [12] proponuje dwunastokrokowy model ataku z powtórzeniami poszczególnych kroków w przypadku łamania kolejnych haseł. Przegląd literatury dotyczący udoskonalenia technik łamania haseł dla dużych zbiorów został zaprezentowany w rozdziale 5.7.

W niniejszej pracy uwaga skupiona jest na poszukiwaniu optymalnej strategii łamania pojedynczego hasła, przy założeniu że nie dysponujemy żadną wiedzą o strukturze samego hasła (jego długości, złożoności).

1.2. Wykładnia prawna

W Polsce powszechnie znane jest prawo obywateli do braku konieczności ujawniania swoich haseł w przypadku postępowań karnych. Zgodnie z artykułem 74 Kodeksu postępowania karnego (Obowiązki dowodowe): „Oskarżony nie ma obowiązku dowodzenia swej niewinności ani obowiązku dostarczania dowodów na swoją niekorzyść". Zasada ta jest nie tylko międzynarodowym standardem normatywnym w dziedzinie ochrony prawa człowieka oraz prawem obywatelskim zagwarantowanym przez ustawy zasadnicze niektórych państw, lecz przede wszystkim jedną z istotnych gwarancji procesowych, którą Europejski Trybunał Praw Człowieka wywodzi z zasady „uczciwego procesu" [4]. Zauważyć należy jednak, że prawo do nieujawniania własnych haseł dostępu nie jest obowiązujące we wszystkich systemach prawnych. I tak na przykład w Wielkiej Brytanii obowiązuje Regulation of Investigatory Powers Act [5], na mocy którego od 2007 roku nieujawnienie kluczy szyfrujących (w tym haseł dostępowych) na żądanie organów procesowych grozi karą pozbawienia wolności do lat 2, a w szczególnych przypadkach (np. bezpieczeństwo narodowe) do lat 5. Pierwsze wykorzystanie tej regulacji miało miejsce już w listopadzie 2007 roku przeciwko obrońcom praw zwierząt [6]. Od października 2018 roku służby celne i graniczne Nowej Zelandii mogą żądać od wjeżdżających na terytorium ich kraju podania haseł i odblokowania wwożonych urządzeń elektronicznych, jeżeli mają „uzasadnione podejrzenie", że były one wykorzystywane do działań przestępczych. Odmowa podania haseł może wiązać się z karą grzywny do 3000 dolarów. W Stanach Zjednoczonych natomiast, mimo istnienia tak zwanej piątej poprawki do Konstytucji, gwarantującej prawo do odmówienia zeznań

[3] Racjonalność atakującego rozumiana jest jako zaprzestanie ataku, gdy koszt ataku zaczyna przewyższać wysokość nagrody.

na swoją niekorzyść, w niektórych sprawach procesowych organom procesowym skutecznie udało się nakazać ujawnienie kluczy szyfrujących przez podejrzanego [7].

Innym polskim aktem obejmującym tematykę ujawniania haseł są Wytyczne nr 3 KGP [8], w § 69 punkt 8 widnieje tam zapis: „Przeprowadzający przeszukanie ma prawo zażądać od dysponenta lub użytkownika urządzenia zawierającego dane informatyczne lub systemu informatycznego ujawnienia hasła lub haseł umożliwiających dostęp do urządzenia lub systemu, nawet wówczas, gdy dysponentem lub użytkownikiem jest osoba, o której mowa w art. 74, art. 182 lub art. 183 k.p.k.". W świetle tego zapisu zarówno oskarżony, jak i świadek nie mają obowiązku ujawnienia haseł, jeżeli może to narazić ich bliskich na odpowiedzialność karną.

Innym zagadnieniem prawnym wartym zasygnalizowania jest sam proces przełamywania zabezpieczeń (łamania hasła) przez informatyka śledczego. Zgodnie z Kodeksem karnym osobie, która dokonuje takiego czynu, można by przedstawić zarzut z art. 267 § 1 i 2 k.k. [9]: „§ 1. Kto bez uprawnienia uzyskuje dostęp do informacji dla niego nieprzeznaczonej, otwierając zamknięte pismo, podłączając się do sieci telekomunikacyjnej lub przełamując albo omijając elektroniczne, magnetyczne, informatyczne lub inne szczególne jej zabezpieczenie, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 2. § 2. Tej samej karze podlega, kto bez uprawnienia uzyskuje dostęp do całości lub części systemu informatycznego". Mimo że informacje zawarte w materiale dowodowym nie są przeznaczone dla biegłego, to jednak nie wyczerpuje on znamion czynu zabronionego, ponieważ działa na mocy postanowienia o powołaniu dowodu z opinii biegłego. To właśnie postanowienie wraz z zawartym w nim przedmiotem i zakresem badań uprawniają biegłego do ewentualnego przełamania zabezpieczeń (złamania hasła) w materiale dowodowym.

1.3. Zabezpieczanie materiału dowodowego

Z punktu widzenia zabezpieczania materiału dowodowego przez organy ścigania istotnym faktem powinna być świadomość istnienia i powszechności stosowania narzędzi do szyfrowania. Dawniej „dobre praktyki" informatyki śledczej nakazywały niezwłoczne wyłączenie zabezpieczanego sprzętu, głównie z obawy przed zdalnym usuwaniem danych. Aktualnie, w dobie powszechnego szyfrowania danych, w przypadku gdy urządzenie cyfrowe (komputer, smartfon) jest włączone podczas zatrzymania, należy podjąć wszelkie środki, aby zabezpieczyć istniejące na nim dane, poprzez zrobienie obrazu dysku (kopii binarnej), zrzutu pamięci (ang. *memory dump*) czy nawet zwykłego kopiowania danych. Może to być często jedyny sposób, by zapoznać się z ich treścią.

W przeciwnym wypadku, w sytuacji gdy dane urządzenie ma włączoną funkcję szyfrowania danych, po jego wyłączeniu organ procesowy może bezpowrotnie stracić możliwość dostania się do danych na nim zawartych. W szczególności gdy

zatrzymany nie zdecydowałby się na podanie haseł dostępowych. Przykładem takiego pozyskania danych było zatrzymanie w dniu 1.10.2013 r. Rossa Ulbrichta – założyciela i administratora serwisu SilkRoad [10]. Agenci FBI, podejrzewając, że jego laptop będzie zaszyfrowany, zabezpieczyli go, gdy był włączony w publicznej bibliotece, odwracając uwagę R. Ulbrichta poprzez sfingowaną kłótnię.

Łamanie hasła dla informatyka śledczego powinno być niejako ostatecznością. Najpierw powinien wykorzystać wszelkie możliwości pozyskania hasła. Rozpoczynając od próby nakłonienia zatrzymanego do dobrowolnego podania hasła (na przykład za pomocą różnych socjotechnik), poprzez zabezpieczenie wszelkich dowodów, na których hasła mogą być zapisane (karteczki, notatki, kopie zapasowe itp.), i wreszcie kończąc na przeszukaniu komputera pod kątem wszelkich występujących w nim haseł (hasła zapisane w przeglądarkach internetowych, hasła zapisane w plikach tekstowych itp.). Proces ten ma bardzo duże znaczenie dla tworzenia słownika, który zostanie wykorzystany w procesie łamania hasła. Użytkownicy bardzo często używają tego samego hasła lub jego modyfikacji w różnych miejscach. Coraz większego znaczenia nabiera także zabezpieczanie pamięci ulotnych (RAM). To właśnie w nich mogą znajdować się hasła, które jeśli nie zostaną odpowiednio zabezpieczone (na przykład oprogramowaniem typu: Volatility, Passware Kit Forensic [D], Belkasoft RAM Capturer [D], AccessDdata FTK Imager [E]), będą bezpowrotnie utracone kilka/kilkanaście sekund po odłączeniu od nich zasilania.

2. Przechowywanie haseł

Możliwe sposoby przechowywania haseł w systemach teleinformatycznych zostały szczegółowo opisane w pracy [1]. Zaczynając od najmniej bezpiecznej metody, czyli przechowywania haseł w postaci jawnej, poprzez przechowywanie haseł w postaci zaszyfrowanej (przy wykorzystaniu pewnego algorytmu symetrycznego), a kończąc na przechowywaniu haseł w postaci skrótów wyznaczanych przez kryptograficzne funkcje skrótu. Ten ostatni sposób także znacznie ewoluował na przestrzeni kilkunastu ostatnich lat. Zaczynając od prostych funkcji skrótu (MD5, SHA-1), poprzez ich „solenie" i/lub wielokrotne iteracje, a kończąc na algorytmach adaptacyjnych, w których można zwiększać z czasem liczbę iteracji (bcrypt, PBKDF2) czy też używane przez algorytm zasoby pamięci (SCRYPT, ARGON2). Rzeczywiste systemy informatyczne często nie idą w parze z aktualnymi zaleceniami dotyczącymi bezpiecznego przechowywania haseł [11]. Wynikać to może zarówno z konieczności zachowania kompatybilności wstecznej, jak i wygody deweloperów. W celu ukazania wybranych metod przechowywania haseł przez aplikacje, kilka z nich zostało przedstawionych w tabeli 1.

Tabela 1

Metody przechowywania haseł w wybranych aplikacjach

aplikacja	hasło w postaci
Windows	MD4(UTF-16-LE(hasło)) [NT-Hash]
Linux	np. SHA-512(hasło.sól)
MySQL	SHA1(SHA1(hasło))
GaduGadu10	Base64(MD5(hasło))

Poza wiedzą o tym, w jaki sposób dana aplikacja przechowuje hasła użytkowników, najczęściej niezbędna będzie także wiedza, gdzie hasło, a dokładniej jego skrót, jest przechowywane. Przykładowe lokalizacje, w których aplikacje przechowują skróty haseł, zostały przedstawione w tabeli 2.

Tabela 2

Miejsce przechowywania haseł w wybranych aplikacjach z przykładami

aplikacja	Lokalizacja pliku	przykład
Windows	C:\Windows\System32\config\SAM	Anonim:502: B4B9B02E6F09A9BD760F388B67351E2B
Linuks	/etc/shadow	anonim:6ocCCi0SR$YeY6Y5.nR6ZZb JWgcQQhctYsxicMb9CQ9nrDRY8u3F9uM rLSvDDb6Re5Ncpf/PEYcgXzgzQG0GOrn hTEZd2CH0:17802:0:99999:7:::
MySQL	C:\xampp\mysql\data\mysql\user.MYD	*AA81355A7C3902945576 C90825527897F518D7D9
GG10	C:\Users\[username]\AppData\Roaming\ Gadu-Gadu10\[nrGG]\profileBasic.xml	<ProfilePasswordHash> fvVw7Neq3s+h7VemdeGC6A== </ProfilePasswordHash>

W przypadku najpopularniejszych systemów baz danych[4] skróty haseł można wydobyć za pomocą poleceń języka SQL:t

- Oracle 10g: `SELECT username, password FROM dba_users;`
- MySQL5+: `SELECT User, Password FROM mysql.user;`
- MSSQL: `SELECT name, password_hash FROM sys.sql_logins;`
- POSTGRES: `SELECT username, password FROM pg_shadow;.`

[4] Pod warunkiem dostępu na prawach administratora bazy danych.

Warty pokazania jest również zestaw skryptów w języku Python umożliwiający wydobywanie skrótów haseł z popularnych programów i zaszyfrowanych plików. Wybrane nazwy skryptów zostały zaprezentowane w tabeli 3, natomiast pełna lista skryptów dostępna jest na przykład w publikacji [12].

Tabela 3

Skrypty w języku Python wydobywające skróty haseł do wybranych programów

skrypt	program/plik
7z2john.py	7zip
efs2john.py	Windows EFS
pdf2john.py	zaszyfrowane pliki PDF
zip2john.py	zaszyfrowane pliki ZIP

3. Metody łamania haseł

Proces łamania hasła polega na próbie odwrócenia skrótu hasła (ang. *hash*), a więc znalezienia takiego ciągu znaków (hasła), który po skróceniu jest zgodny ze skrótem hasła. Atak wygląda więc następująco: brany jest kolejny ciąg znaków (np. kolejne hasło ze słownika w ataku słownikowym, kolejny ciąg liter w ataku brutalnym itd.), wyliczany zostaje skrót dla danego ciągu i na koniec otrzymany skrót jest porównywany ze skrótem hasła, które jest łamane. Ze względu na wybór kolejnych ciągów ataki można podzielić na kilka typów.

3.1. Atak słownikowy (ang. *dictionary attack*)

Atak słownikowy polega na sprawdzaniu kolejnych haseł z pewnego słownika. Za słowniki mogą służyć: słowniki najpopularniejszych haseł, słowniki haseł, które wyciekły z różnych serwisów internetowych (np. RockYou, MySpace), różne kompilacje haseł pochodzących z różnych wycieków (np. scottlinux [F], exploit.in, berzerk0 [G], wordlists.capsop.com [H]), słowniki słów z danego kraju (np. słownik PWN) itd. Sam pomysł użycia haseł z pewnego słownika wywodzi się z dwóch spostrzeżeń: użytkownicy używają tych samych haseł w różnych miejscach (serwisach internetowych, systemach), hasła oparte na słowach są znacznie łatwiejsze do zapamiętania niż losowo wybrane ciągi znaków [13].

Jeśli mamy do czynienia z atakiem kierowanym na hasło konkretnego użytkownika, to skuteczną metodą jest zbudowanie słownika charakterystycznego dla danego użytkownika zawierającego na przykład: imiona i nazwiska członków jego rodziny, najważniejsze daty (urodzin, ślubu itp.), słowa związane z zainteresowaniami (hobby, marki samochodów, dyscypliny sportowe, nazwy klubów itp.). Dlatego też

tak ważnym elementem jest tutaj sam proces zabezpieczenia materiału dowodowego (punkt 1.3). Po zebraniu zbioru słów charakterystycznych dla danego użytkownika można użyć narzędzi typu *Common User Passwords Profiler* [I] wspomagających tworzenie słowników haseł na podstawie zgromadzonych danych. Atak słownikowy jest skuteczny tylko wówczas, gdy w słowniku znajduje się hasło, które podlega łamaniu. Stąd tak istotnym elementem jest tutaj przygotowanie odpowiednich słowników.

3.2. Atak brutalny (ang. *brute-force attack*)

Atak brutalny, zwany również atakiem wyczerpującym, polega na sprawdzeniu wszystkich możliwych kombinacji ciągów znaków (haseł) o określonej długości należących do określonego zbioru. Najpopularniejsza notacja zapisu grup znaków pochodząca z notacji używanej w najpopularniejszych programach do łamania haseł (hashcat, JohnTheRipper) przedstawia się następująco: ?l — małe litery (ang. *lower-case letters*) [a-z], `?u` – wielkie litery (ang. *upper-case letters*) [A-Z], `?d` — cyfry (ang. *digits*) [0-9], `?s` — znaki specjalne (ang. *special chars*) [`!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~`]. Zakładając przykładowo, że atak ma polegać na przeszukaniu wszystkich możliwych haseł czteroznakowych złożonych tylko z małych liter, wówczas kolejne sprawdzane ciągi znaków wyglądałyby następująco: aaaa, aaab, aaac, …, aaaz, aaba, aabb, aabc, …, zzzz. Głównym problemem w ataku tego typu jest wykładniczo rosnąca liczba haseł wraz ze wzrostem liczby znaków. Przykładowo dla wszystkich ośmioznakowych haseł zbudowanych na 62-znakowym alfabecie [a-zA-Z0-9] (inny zapis: `?l?u?d`) można utworzyć $62^8 = 218340105584896$ haseł. Tabela 4 ukazuje liczby możliwych haseł dla wybranych zbiorów znaków.

Tabela 4

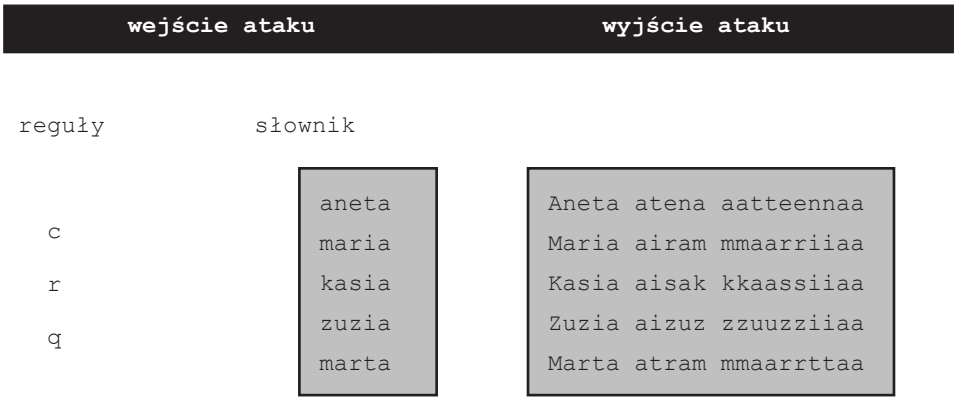
Liczby możliwych haseł dla wybranych zbiorów znaków

długość hasła	zbiór znaków	liczba znaków	liczba haseł
8	?l?u?d	62	218 340 105 584 896
8	?l?u?d?s	95	6 634 204 312 890 625
9	?l?u?d	62	13 537 086 546 263 552
9	?l?u?d?s	95	630 249 409 724 609 375
10	?l?u?d	62	839 299 365 868 340 224
10	?l?u?d?s	95	59 873 693 923 837 890 625

Atak brutalny jest skuteczny, jeśli atakujący dysponuje odpowiednio dużą mocą obliczeniową i czasem. Sama idea skuteczności ataku brutalnego bazuje na spostrzeżeniu, że krótkie hasła są łatwiejsze do zapamiętania [13] i dlatego są chętniej stosowane.

3.3. Atak oparty na regułach (ang. *rule attack*)

Atak oparty na regułach polega na sporządzeniu na podstawie pewnego słownika listy słów utworzonych poprzez: modyfikacje, wstawienia, powtórzenia, pomijanie pewnych znaków do haseł znajdujących się w słowniku. Przykładowymi, niejedynymi, operacjami modyfikującymi słowa mogą być: `c` — pierwszy znak słowa wielką literą, pozostałe małymi literami, `r` — odwrócenie całego słowa, `{` — rotacja słowa w lewo, `q` — zdublowanie każdego znaku w słowie, `[` — usunięcie pierwszego znaku w słowie, `iNX` — wstawienie znaku `X` na pozycji `N`, `oNX` — zastąpienie znaku na pozycji `N` znakiem `X`. Powyższa nomenklatura jest także spójna dla najpopularniejszych programów do łamania haseł. Przykład tworzenia haseł dla ataku opartego na regułach został przedstawiony na schemacie 1.

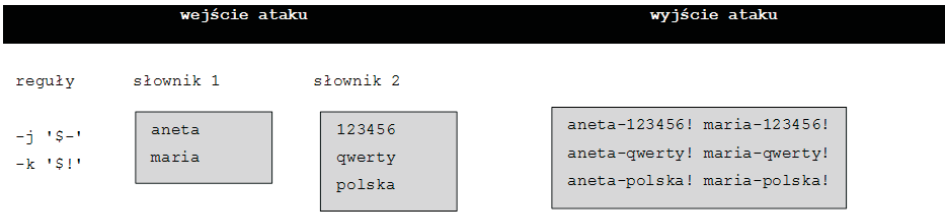


Schemat 1. Przykład tworzenia haseł dla ataku opartego na regułach

Programy do łamania haseł (np. hashcat) udostępniają najpopularniejsze reguły. Pośród nich znajdują się na przykład gotowe reguły dla tak zwanej LeetSpeak [J] czy też zamieniające poszczególne litery w słowniku na wielkie (toggle-attack [K]).

3.4. Atak kombinacyjny (ang. *combinator attack*)

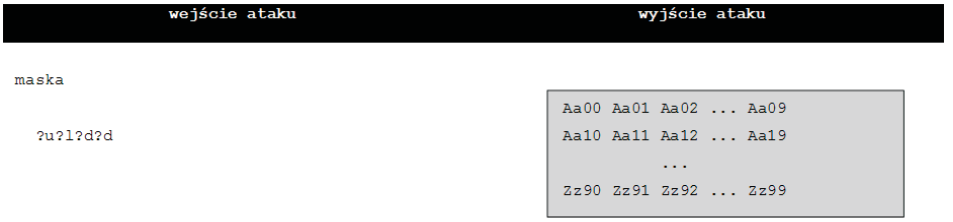
Atak kombinacyjny jest oparty na dwóch słownikach (mogą to być te same słowniki) — łączy wszystkie słowa z jednego słownika ze wszystkimi słowami z drugiego słownika. Dodatkowo do każdego słownika można dodać pewną regułę dołączania. Atak ten jest szczególnie skuteczny, gdy łamane hasło złożone jest z dwóch słów słownikowych. Przykład tworzenia haseł dla ataku kombinacyjnego został przedstawiony na schemacie 2.



Schemat 2. Przykład tworzenia haseł dla ataku kombinacyjnego

3.5. Atak oparty na maskach (ang. *mask attack*)

Atak oparty na maskach jest pewną odmianą ataku brutalnego, umożliwiającą jego zawężenie. Głównym powodem jego stosowania jest redukcja przestrzeni potencjalnych haseł, co czyni ten atak bardziej wydajnym od ataku brutalnego. Przykładowo, jeśli atakujący chce złamać hasło „Zuzia1978", wówczas przy użyciu ataku brutalnego musiałby przeszukać 62^9 (13 537 086 546 263 552) haseł. Wiedząc natomiast, że pierwsza litera jest wielka, potem następują cztery małe litery zakończone czterema cyframi, złożoność ataku może zostać znacząco zredukowana (dzięki zastosowanej masce `?u?l?l?l?l?d?d?d?d`) do przeszukania 26 x 26^4 x 10^4 (118 813 760 000) haseł. Przykład tworzenia haseł dla ataku opartego na maskach został przedstawiony na schemacie 3.



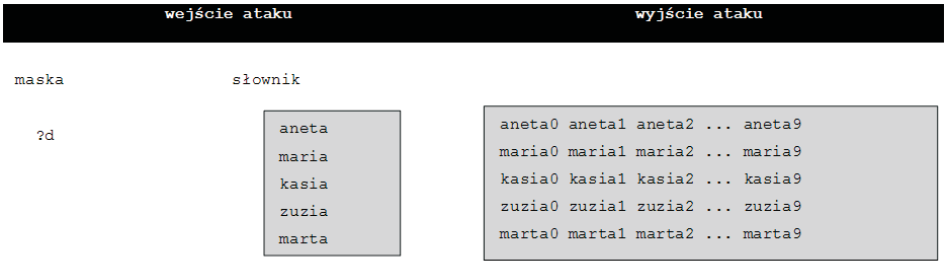
Schemat 3. Przykład tworzenia haseł dla ataku opartego na maskach

Programy do łamania haseł (np. hashcat) udostępniają najpopularniejsze maski pochodzące z analiz statystycznych dostępnych wycieków haseł.

3.6. Atak hybrydowy (ang. *hybrid attack*)

Atak hybrydowy łączy w sobie atak słownikowy z atakiem brutalnym. Dla każdego słowa znajdującego się w słowniku zostaje dodana pewna część łamania brutalnego, na przykład kolejne liczby, dowolne dwie litery itp. Dołączenie może

następować zarówno przed słowem ze słownika, jak i po danym słowie. Przykład tworzenia haseł dla ataku hybrydowego został przedstawiony na schemacie 4.



Schemat 4. Przykład tworzenia haseł dla ataku hybrydowego

3.7. Inne typy ataków

Poza omówionymi wyżej atakami warto wymienić bardziej zaawansowane techniki wykorzystujące pewien zbiór danych testowych w celu posegregowania haseł według prawdopodobieństwa ich występowania. Podejście to w literaturze doczekało się różnych rozwiązań, które można przydzielić do czterech kategorii: metody oparte na modelu Markowa, PCFG (ang. *Probabilistic Context-Free Grammars*), metody oparte na uczeniu maszynowym oraz metody oparte na personalizacji. Pierwsza z wymienionych kategorii używana jest w celu zmniejszenia liczby przeszukiwanych haseł przy ataku brutalnym. Łańcuchy Markowa pierwszego rzędu zostały po raz pierwszy wykorzystane do łamania haseł przez Narayanana i Shmatikova, którzy obliczali prawdopodobieństwo wystąpienia kolejnego znaku w haśle w zależności od znaku poprzedniego, bazując na słownikach konkretnego języka i wyciekach haseł [14]. Metoda ta została później udoskonalona przez Castelluccia i in. [15] poprzez wykorzystanie łańcuchów wyższych rzędów. Druga kategoria metod bazuje na badaniu prawdopodobieństwa występowania haseł o określonej masce, użyta została po raz pierwszy przez Weir i in. [16]. Trzecia kategoria obejmuje próby wykorzystania sieci neuronowych [17]. Natomiast w ostatniej grupie metod próbuje się udoskonalić inne techniki poprzez wykorzystanie spersonalizowanych danych. Do przykładowych podejść zaliczyć można: Personal-PCFG [18], TarGuess [19] czy OMEN+ [20].

4. Narzędzia do łamania haseł

4.1. Oprogramowanie (Software)

Jedną z pierwszych decyzji, przed którą staje informatyk śledczy, rozpoczynając proces łamania hasła, jest wybór odpowiedniego oprogramowania. Wybór ten zależy: od algorytmu, który został użyty do przechowania hasła (program musi wspierać ten typ hasła), dostępności danego oprogramowania (programy darmowe i płatne), od możliwości wykorzystania środowiska rozproszonego przez daną aplikację, czy też od samych preferencji atakującego. Do najpopularniejszych narzędzi ogólnego przeznaczenia, przydatnych w procesie łamania haseł, należą między innymi: hashcat [L], JohnTheRipper [M], Cain and Abel [N]. Natomiast do programów wspomagających proces przełamywania zabezpieczeń w postaci haseł do popularnych programów (m.in.: TrueCrypt, BitLocker, RAR, ZIP, MS Office) zaliczyć można między innymi: Passware Kit Basic [O] estońskiej firmy Passware wydobywający hasła z różnych zrzutów pamięci; BitCracker [P] do łamania haseł wolumenów zaszyfrowanych BitLockerem; Password Recovery Toolkit [Q] firmy AccessData lub jego wersję umożliwiającą atak w środowisku rozproszonym Distributed Network Attack [R].

W dzisiejszych czasach, gdy proces wyliczania skrótów (szczególnie dla popularnych funkcji takich jak MD5, SHA-1) może być znacząco przyspieszony dzięki wykorzystaniu mocy obliczeniowej kart graficznych, istotne jest wybranie oprogramowania wspierającego sprzętową akceleracje.

4.2. Sprzęt (Hardware)

Najważniejszym czynnikiem przy łamaniu hasła jest wybór odpowiedniej platformy sprzętowej. Oczywiście jest on w dużej mierze zależny od możliwości atakującego. Może on dysponować jedynie komputerem klasy PC, a może mieć dostęp do wyspecjalizowanej platformy opartej na kilku, kilkunastu kartach graficznych. Inną wartą rozważenia opcją może być użycie coraz popularniejszych chmur obliczeniowych. Wymienić warto tutaj: Azure Microsoftu [S], Amazon Web Services [T], chmurę obliczeniową NVidii [U] opartej na kartach z rodziny Tesla, czy serwis vast.ai [V] umożliwiający dzielenie się mocą obliczeniową kart graficznych użytkowników.

5. Wybór strategii łamania hasła

Wybór strategii łamania hasła zostanie przedstawiony dla scenariusza, gdy atakujący dysponuje skrótem hasła, wie, za pomocą jakiego algorytmu skrót ten został utworzony, jednak nie ma żadnej wiedzy na temat samego hasła (jego długości, złożoności itp.). Z doświadczenia autora sytuacja taka jest częstym przypadkiem napotykanym w rzeczywistych sprawach procesowych. Strategia zostanie opisana na przykładzie programu hashcat dla dwóch rzeczywistych przypadków, dla łamania hasła do: a) komunikatora Gadu-Gadu 10, oznaczonego w dalszej części artykułu [MD5], b) bazy danych MySQL [SHA-1**][5]. Dodatkowym ograniczeniem będzie dysponowanie czasem jednego miesiąca, co także jest często praktykowane przy typowych sprawach procesowych.

Rozważania zostaną przedstawione dla czterech popularnych wariantów sprzętowych: laptopa z procesorem i wbudowaną kartą graficzną [LAPTOP], komputera PC z kartą graficzną klasy GeForce GTX1070 (popularny zestaw dla „gracza") [GAMER], wyspecjalizowanego urządzenia złożonego z sześciu kart graficznych klasy GeForce GTX1080 [HASHKILLER], wykorzystania wynajętej mocy obliczeniowej (na przykład z serwisu vast.ai) [SHARING]. W tabeli 5 pokazany został szacunkowy koszt zakupu oraz godzina pracy platformy przy założeniach, że średni koszt 1 kWh energii elektrycznej w Polsce wynosi 60 groszy i rzeczywistych pomiarach zużywanej energii.

Tabela 5

Szacunkowy koszt zakupu i roboczo-doby pracy dla wybranych platform

platforma	koszt zakupu [pln]	moc [kW]	koszt roboczo-doby [pln]
[LAPTOP]	5000	100	1,4
[GAMER]	6000	250	3,6
[HASHKILLER]	30000	1250	18,0
[SHARING][1]	0	n/a	48,9

[1] Przykładowa oferta pochodząca z serwisu vast.ai o następujących parametrach: identyfikator oferty — 164861, cena wynajmu — 1,148 dol./godz., GPU - 4x GTX 1080 Ti, obliczenia roboczodoby dla kursu dolara 1 dol. = 1,75 zł.

[5] Ze względu na fakt, że baza danych MySQL przechowuje hasła w postaci zaprezentowanej w tabeli 1, a więc w postaci dwukrotnie użytego algorytmu SHA-1, dla przejrzystości zapisu wprowadzono oznaczenie SHA-1**.

5.1. Przeszukiwanie bazy skrótów

Mając dany skrót hasła, przed rozpoczęciem samego procesu jego łamania warto przeszukać istniejące bazy skrótów. W sieci Internet znajduje się wiele stron internetowych, zarówno bezpłatnych, jak i płatnych, zawierających całkiem pokaźną liczbę par w postaci: hasło – skrót. Aktualnie do stron zawierających największą liczbę zdekodowanych skrótów, najczęściej dla funkcji MD5, zaliczyć można między innymi: [W], [X], [Y]. Czasami także samo podanie skrótu hasła w najpopularniejszych wyszukiwarkach potrafi zwrócić w wynikach wyszukiwania łamane hasło.

5.2. Oszacowanie mocy obliczeniowej

Pierwszą czynnością, jaką należy wykonać, rozpoczynając łamanie hasła, jest oszacowanie mocy obliczeniowej, którą dysponuje atakujący, i wydajności dostępnych zasobów sprzętowych dla algorytmu, jakim został wyznaczony skrót łamanego hasła, tak zwany benchmark. Przykładowe polecenia dla programu hashcat, na dwóch różnych platformach sprzętowych i dla różnych funkcji skrótu, zostały przedstawione w kodzie 1.

```
// platforma [GAMER], algorytm bazy MySQL [300]
D:\hashcat-4.2.1>hashcat64.exe -b -D 0 -m 300
  ...
* Device #1: GeForce GTX 1060 6GB, 1536/6144 MB allocatable, 10MCU
Hashmode: 300 - MySQL4.1/MySQL5
Speed.Dev.#1:  2109.6 MH/s (78.86ms) @ Accel:256 Loops:128 Thr:512 Vec:1

// platforma [LAPTOP], algorytm MD5 [1]
D:\hashcat-4.2.1>hashcat64.exe -b -D 1 -m 0
  ...
* Device #2: Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz, 4MCU
Hashmode: 0 - MD5
Speed.Dev.#2:  231.8 MH/s (17.73ms) @ Accel:1024 Loops:1024 Thr:1 Vec:8
```

Kod 1. Wiersz poleceń dla programu hashcat ukazujący benchmark dla wybranych algorytmów

Dla wyszczególnionych w tabeli 5 rozwiązań sprzętowych oszacowano ich wydajności dla dwóch algorytmów (funkcji skrótu MD5 i algorytmu bazy danych MySQL), uzyskując wyniki zaprezentowane w tabeli 6.

Tabela 6

Wydajność dla wybranych platform sprzętowych

platforma	[MD5]	[SHA-1**]
[LAPTOP]	231,8 MH/s[1]	48,9 MH/s
[GAMER]	11 726,3 MH/s	2 109,6 MH/s
[HASHKILLER]	153,9 GH/s[2]	20,0 GH/s
[SHARING]	124,6 GH/s	20,8 GH/s

[1] MH/s – 10^6 skrótów (hashy) na sekundę.
[2] GH/s – 10^9 skrótów (hashy) na sekundę.

5.3. Przygotowanie słowników

W sieci Internet znajduje się duża liczba różnych gotowych słowników ułatwiających proces łamania haseł. Należą do nich różnorodne bazy pochodzące ze źródeł wyszczególnionych w punkcie 3.1 niniejszego artykułu. Autor w swojej praktyce używa dwóch typów słowników. Jednego będącego kompilacją haseł pochodzących z różnych wycieków zawierającego ponad 1,5 mld haseł [dict_breaches]. Drugiego tworzonego na potrzeby konkretnej sprawy [dict_private], zawierającego: wszelkie zidentyfikowane dane osobowe, informacje znalezione na profilach społecznościowych, a także hasła pozyskane w trakcie analizy materiału dowodowego („zapamiętane" hasła przeglądarek internetowych, hasła zapisane w postaci jawnej w plikach tekstowych itp.).

5.4. Atak słownikowy

Pierwszym atakiem, od którego zaleca się rozpoczęcie procesu łamania haseł, jest atak słownikowy, gdzie słowniki stanowią zbiory: [dict_breaches] oraz [dict_private]. Atak ten nawet dla największych słowników haseł spotykanych w Internecie (na przykład słownik [dict_breaches] o rozmiarze 95 GB) jest stosunkowo szybki do wykonania. Czas jego wykonania na platformie sprzętowej [gamer] wynosi około 1 godziny.

5.5. Atak brutalny

Kolejnym atakiem zalecanym do wykonania po nieskutecznym ataku słownikowym jest atak brutalny. Atakujący najpierw musi odpowiedzieć sobie na pytanie, ile czasu może poświęcić na ten atak przy posiadanych zasobach sprzętowych. Tabela 7 przedstawia czas trwania ataku brutalnego dla trzech długości haseł (do 7, do 8 i do 9 znaków) przy pełnej złożoności (litery małe i wielkie alfabetu angielskiego, cyfry i znaki specjalne — 95 znaków) na różnych platformach sprzętowych.

Tabela 7

Czas łamania brutalnego dla wybranych platform sprzętowych i algorytmu

platforma	algorytm	wydajność [GH/s]	czas trwania ataku [hh:mm:ss] do 7 znaków[1]	do 8 znaków[2]	do 9 znaków[3]
[LAPTOP]	[SHA-1**]	0,05	392:05:33	37248:46:59	3538634:23:34
	[MD5]	0,23	85:14:15	8097:33:42	769268:20:46
[GAMER]	[SHA-1**]	2,06	9:31:01	904:05:48	85889:11:03
	[MD5]	11,45	1:42:44	162:39:30	15452:33:07
[HASHKILLER]	[SHA-1**]	20,00	0:58:49	93:07:19	8846:35:10
	[MD5]	153,90	0:07:39	12:06:06	1149:39:14
[SHARING]	[SHA-1**]	20,80	0:56:33	89:32:25	8506:19:58
	[MD5]	124,60	0:09:26	14:56:50	1419:59:52

[1] Liczba wszystkich haseł o długości do 7 znaków wynosi 70576641626495.
[2] Liczba wszystkich haseł o długości do 8 znaków wynosi 6704780954517110.
[3] Liczba wszystkich haseł o długości do 9 znaków wynosi 636954190679126000.

Wnioski płynące z tabeli 7 są następujące:

— atakujący dysponujący platformą opartą na procesorze ([laptop]) może przeprowadzić atak brutalny na hasła o długości do 7 znaków dla mniej złożonych obliczeniowo[6] funkcji skrótu (czas: ponad 3 dni); wykonanie ataku na hasła o większych długościach czy też bardziej złożonych obliczeniowo[7] funkcji skrótu jest niepraktyczne;
— atakujący dysponujący platformą opartą na GPU ([gamer]) może przeprowadzić atak brutalny na hasła o długości do 8 znaków dla mniej złożonych obliczeniowo funkcji skrótu (czas: ponad 6 dni), dla bardziej złożonych funkcji czy też dłuższych haseł atak ten jest zbyt czasochłonny;
— atakujący dysponujący specjalizowaną platformą ([hashkiller], [sharing]) może przeprowadzić atak brutalny na hasła o długości do 8 znaków (czas: kilka dni dla mniej złożonych obliczeniowo funkcji skrótu); wykonanie ataku na hasła o długości do 9 znaków (oraz dłuższych) jest niepraktyczne (czas: około roku).

5.6. Atak oparty na regułach

W przypadku niepowodzenia ataku brutalnego w trzeciej kolejności zaleca się przeprowadzenie ataku opartego na regułach dla słownika [dict_private]. Popularne programy do łamania haseł (np. hashcat) mają już przygotowane zestawy najpopularniejszych reguł. Z doświadczeń autora wynika, że atak ten jest szczególnie

[6] Do funkcji mniej złożonych obliczeniowo zaliczyć można na przykład algorytmy MD4, MD5.
[7] Do funkcji bardziej złożonych obliczeniowo zaliczyć można na przykład algorytmy SHA-1**, SHA-256.

skuteczny, gdy dysponujemy „bogatym" słownikiem ukierunkowanym na konkretnego użytkownika.

Dla ukierunkowanego słownika zawierającego kilkaset haseł czas trwania tego ataku, nawet przy wyborze kilkudziesięciu zestawów reguł na platformie sprzętowej [gamer], wynosi około 1-2 godzin. Czas ataku zmienia się diametralnie przy zastosowaniu dużych, nieukierunkowanych słowników[8].

5.7. Atak oparty na maskach

Ze względu na fakt, że przeprowadzony atak brutalny wyczerpał wszystkie możliwe hasła o określonej długości (na przykład do 7 lub do 8 znaków), kolejny atak, a więc atak oparty na maskach, nie powinien już powielać przeszukanych haseł. Popularne programy do łamania haseł mają już przygotowane zestawy najpopularniejszych masek pochodzących z analizy haseł występujących w wyciekach (na przykład w programie hashcat dostępne są maski haseł z wycieku RockYou). Atakującemu pozostaje tylko odrzucenie masek zbędnych. W celu odrzucenia wszystkich masek o określonej długości (`$mask_size`) i pozostawienia tylko masek większych od niej można wykorzystać skrypt zaprezentowany w kodzie 2.

```
<?php
  $mask_size = 8;
    $file_in = file('rockyou-7-2592000_cleaned.hcmask');
    $file_ou = fopen('rockyou-'.$mask_size.'+.hcmask', 'w');
    for ($i = 0; $i < sizeof($file_in); $i++)
       if (strlen($file_in[$i]) > 2*$size) fwrite($file_ou, $file_in[$i]);
    fclose ($file_ou);
?>
```

Kod 2. Skrypt w języku PHP wybierający maski powyżej zadanej długości

Atak ten jest ostatnim atakiem zalecanym do wykonania przez autora. Ze względu na jego złożoność, liczba przeszukanych masek uzależniona jest głównie od czasu, jakim dysponuje jeszcze atakujący. Zakładając, że posiada listę masek o długości co najmniej (`$mask_size`), za pomocą skryptu zaprezentowanego w kodzie 3 można obliczyć liczbę przeszukanych haseł przy założonej mocy obliczeniowej w określonym czasie.

[8] Przykładowo dla słownika [`dict_breaches`], platformy [`gamer`], zestawu reguł o nazwie `best64.rule` pochodzących z programu hashcat czas takiego ataku wynosi około 16 dni.

```
<?php
....$time_max  = 10*24*60*60; //maksymalny czas ataku w sekundach (np.10 dni)
    $file_mask = file('rockyou-8+.hcmask');
....$benchmark = 20 * 10^9;  //wydajność [H/s] (np.20 GH/s)

    $total_time = 0; $total_hash = 0;
    for ($i = 0; $i < sizeof($file_mask); $i++) {
        $mask = str_replace("?", "", $file_mask[$i]);      $count = 1;
        for($j=0; $j < strlen($mask); $j++) {
            switch ($mask[$j]) {
                case "l": $count *= 26;  break;
                case "u": $count *= 26;  break;
                case "d": $count *= 10;  break;
                case "s": $count *= 33;  break;
                case "p": $count *= 62;  break;
                case "a": $count *= 95;  break;
                }
            }
            $total_hash += $count;
            $total_time += ($count / $benchmark);
        if ($time_max < $total_time) {
          echo "masks: $i, total_hash: $total_hash, total time: $total_time";
          break;
        }
    }
?>
```

Kod 3. Skrypt w języku PHP obliczający liczbę przeszukanych haseł

5.8. Szacowanie efektywności

Na koniec pokazano oszacowanie efektywności zaproponowanych ataków dla czterech przedstawionych platform sprzętowych przy założeniu, że atakujący dysponuje czasem 30 dni na przeprowadzenie ataku i postępuje zgodnie z wyżej zaprezentowanym scenariuszem. Najpierw wykona atak słownikowy dla słowników: [dict_breaches] oraz [dict_private], następnie atak brutalny (do 7 lub 8 znaków, w zależności od platformy), po czym atak oparty na regułach dla słownika [dict_private], a pozostały czas wykorzystany zostanie na atak oparty na maskach[9].

[9] Za pozostały czas (przeznaczony na atak oparty na maskach) przyjmuje się tutaj czas dwudziestu pięciu dni pomniejszony o czas ataku brutalnego. Pozostałe 5 dni z dostępnych 30 szacunkowo przeznacza się na same przygotowania ataków, przygotowanie słowników, wykonanie opinii końcowej itp.

Ze względu na relatywnie krótki czas poświęcony na atak słownikowy i oparty na regułach, ataki te nie zostały uwzględnione w szacowaniu efektywności[10]. Dla każdej platformy pokazano: liczbę wygenerowanych skrótów (przeszukanych haseł), czas wykonania ataku oraz koszt przeszukania PH[11] (efektywność ekonomiczna), przy czym na koszt przeprowadzenia ataku składa się tylko koszt energii elektrycznej lub koszt wynajęcia mocy obliczeniowej (zgodnie z wartościami z tabeli 6).

Tabela 8

Efektywność ekonomiczna ataków dla algorytmu [MD5]

platforma	koszt roboczo-doby [pln]	wydaj -ność [GH/s]	atak brutalny		atak na maskach		efektywność ekonomiczna [pln / PH]
			skróty [PH]	czas [hh:mm]	skróty [PH]	czas [hh:mm]	
[LAPTOP]	1,4	0,23	0,07	85:15	0,43	514:45	70,45
[GAMER]	3,6	11,45	6,70	162:40	18,03	437:20	3,64
[HASHKILLER]	18,0	153,90	6,70	12:07	325,72	587:53	1,35
[SHARING]	48,9	124,60	6,70	14:57	262,43	585:03	4,54

Tabela 9

Efektywność ekonomiczna ataków dla algorytmu [SHA-1**]

platforma	koszt roboczo-doby [pln]	wydaj -ność [GH/s]	atak brutalny		atak na maskach		efektywność ekonomiczna [pln / PH]
			skróty [PH]	czas [hh:mm]	skróty [PH]	czas [hh:mm]	
[LAPTOP]	1,4	0,05	0,0007	4:08	0,11	595:52	324,07
[GAMER]	3,6	2,06	0,07	9:32	4,38	590:28	20,23
[HASHKILLER]	18,0	20,00	6,70	93:08	36,49	506:52	10,42
[SHARING]	48,9	20,80	6,70	89:33	38,22	510:27	27,21

Z tabel 8 i 9 wynika że: a) największą efektywność ekonomiczną uzyskuje się przy zastosowaniu specjalizowanego rozwiązania sprzętowego [HASHKILLER]; b) używanie mocy obliczeniowej procesora [LAPTOP] jest nieefektywne (zarówno ze względu na liczbę przeszukanych skrótów, jak i koszt jednostkowy); c) warte rozważenia jest wypożyczanie mocy obliczeniowej [SHARING] — daje to zadowalającą efektywność ekonomiczną przy braku kosztów początkowych (zakupu platformy).

[10] Czasy ich wykonania są podobne dla wszystkich platform sprzętowych i przeszukana zostaje jednakowa liczba słów znajdujących się w słownikach.

[11] 1 PH = 10^15 skrótów (hashy), w programie hashcat używana jest notacja układu SI (notacja dziesiętna, w której 1 P = 10^15), a nie notacja powszechnie używana w informatyce (notacja dwójkowa, w której 1 P = 2^50).

6. Podsumowanie

Mimo że coraz częściej słyszy się o tym, że era haseł statycznych powoli wygasa — są „niewygodne, niepewne i drogie" [21], więc czas je zastąpić nowymi technikami uwierzytelniania — to jednak jeszcze długo będą tematem zainteresowania i badań informatyków śledczych. Należy jednak liczyć się z tym, że hasła będą coraz trudniejsze. Wpływ mają na to między innymi: coraz większa świadomość użytkowników, zalecenia i wytyczne odnośnie do polityki haseł [11], a nawet proponowane rozwiązania systemowe (na przykład w stanie Kalifornia od 2020 roku każde nowo wyprodukowane urządzenie z zaprogramowanym hasłem dostępu będzie musiało posiadać unikalne hasło [22]).

Przedstawiona w artykule ścieżka postępowania w przypadku próby złamania pojedynczego hasła jest wynikiem autorskich doświadczeń nabytych w trakcie przeprowadzonych rzeczywistych ataków. Ma ona pomóc informatykom śledczym w: przygotowaniu scenariusza ataku, wyborze optymalnego rozwiązania sprzętowego, oszacowaniu kosztów i liczby możliwych do przeszukania haseł.

Pamiętać jednak należy, że samo łamanie hasła powinno być niejako ostatecznością, bo jak uważa jeden z klasyków zajmujących się przełamywaniem zabezpieczeń, łatwiej jest „złamać człowieka niż hasło" [23].

Źródło finansowania pracy — środki własne autora.

Artykuł wpłynął do redakcji 2.11.2018 r. Zweryfikowaną wersję po recenzjach otrzymano 10.12.2018 r.

Przemysław Rodwald https://orcid.org/0000-0003-4261-8688

LITERATURA

[1] Rodwald P., Biernacik B., *Zabezpieczanie haseł w systemach informatycznych*, Biuletyn WAT/ Bulletin MUT, 2018, vol. 67, s. 73-92, DOI: 10.5604/01.3001.0011.8036.

[2] Ponemon Institute, *Global Encryption Encryption Trends Study*, April 2018, http://go.thalesesecurity.com/rs/480-LWA-970/images/2018-Ponemon-Global-Encryption-Trends-Study-ar.pdf, [dostęp: 31.10.2018].

[3] Blocki J., Harsha B., Zhou S., *On the economics of offline password cracking*, 2018 IEEE Symposium on Security and Privacy, 2018, s. 853-871.

[4] Adamski A., *Prawo karne komputerowe*, CH Beck, 2000, s. 213.

[5] Regulation of Investigatory Powers Act, 2000, http://www.legislation.gov.uk/ ukpga/2000/23/ contents, [dostęp: 31.10.2018].

[6] Ward M., *Campaigners hit by decryption law*, BBCNews.com, 20.11.2007, http://news.bbc.co.uk/2/hi/technology/7102180.stm, [dostęp: 31.10.2018].

[7] Denning D.E., Baugh W.E., *Hiding Crimes in Cyberspace*, 11.08.1999, http://cryptome.org/hiding-db.htm, [dostęp: 31.10.2018].

[8] Wytyczne Nr 3 Komendanta Głównego Policji w sprawie wykonywania niektórych czynności dochodzeniowo-śledczych przez policjantów, z dnia 30 sierpnia 2017 r. (Dz.Urz. KGP z 2017 r., poz. 59).

[9] *Kodeks karny*, Ustawa z dnia 20 lipca 2018 r. (Dz.U. z 2018 r. poz. 1600).

[10] Bertrand N., *The FBI staged a lovers' fight to catch the kingpin of the web's biggest illegal drug marketplace*, BusinessInsider.com, 22.01.2015, https://www.businessinsider.com/the-arrest-of-silk-road-mastermind-ross-ulbricht-2015-1, [dostęp: 31.10.2018].

[11] Grassi P.A., Fenton J.L., Newton E.M., Perlner R.A., Regenscheid A.R., Burr W.E., Justin P.R., *NIST Special Publication 800-63B. Digital Identity Guidelines: Authentication and Lifecycle Management*, Bericht, NIST, 2017, https://pages.nist.gov/800-63-3/sp800-63b.html, [dostęp: 31.10.2018].

[12] Picolet J., *Hash Crack: Password Cracking Manual v 2.0*, ISBN-10: 9781975924584.

[13] Morris R., Thomson K., *Password security: A case history*, Communications of the ACM, 1979, vol. 22, no. 11, s. 594-597.

[14] Narayanan A., Shmatikov V., *Fast dictionary attacks on passwords using time-space tradeoff*, Proc. CCS 2005. ACM, 2005, s. 364-372.

[15] Castelluccia, C., Durmuth M., Perito D., *Adaptive password-strength meters from markov models*, in NDSS, 2012.

[16] Weir M., Aggarwal S., de Medeiros B., Glodek B., *Password cracking using probabilistic context-free grammars*, Security and Privacy, 30th IEEE Symposium on. IEEE, 2009, s. 391-405.

[17] Melicher W., Ur B., Segreti S.M., Komanduri S., Bauer L., Christin N., Cranor L.F., *Fast, lean and accurate: Modeling password guessability using neural networks*, Proc. USENIX Security 2016, 2016.

[18] Li Y., Wang H., Sun K., *A study of personal information in human chosen passwords and its security implications*, INFOCOM 2016, The 35th Annual IEEE International Conference on Computer Communications, 2016, s. 1-9.

[19] Wang D., Zhang Z., Wang P., Yan J., Huang X., *Targeted online password guessing: An underestimated threat*, Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, s. 1242-1254.

[20] Castelluccia C., Chaabane A., Durmuth M., Perito D., *When privacy meets security: Leveraging personal information for password cracking*, arXiv preprint arXiv:1304.6584, 2013.

[21] Tung L., *Microsoft: Here's why we're declaring end of password era*, zdnet.com, 25.009.2018, https://www.zdnet.com/article/microsoft-heres-why-were-declaring-end-of-password-era/, [dostęp 31.10.2018].

[22] Senate Bill No. 327 (SB-327) Information privacy: connected devices, 28.09.2018, https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB327, [dostęp: 31.10.2018].

[23] Mitnick K., *Sztuka podstępu. Łamałem ludzi, nie hasła*, Helion, 2003.

Spis odnośników internetowych [dostęp na dzień 31.10.2018.]:

[A] http://truecrypt.sourceforge.net/

[B] https://symantec.com/smb/drive-encryption

[C] https://www.passware.com/kit-forensic/

[D] https://belkasoft.com/ram-capturer

[E] https://accessdata.com/product-download/ftk-imager-version-4.2.0

[F] https://gist.github.com/scottlinux/9a3b11257ac575e4f71de811322ce6b3
[G] https://github.com/berzerk0/Probable-Wordlists/tree/master/Real-Passwords
[H] https://wordlists.capsop.com
[I] https://github.com/Mebus/cupp
[J] https://pl.wikipedia.org/wiki/Leet_speak
[K] https://hashcat.net/wiki/doku.php?id=toggle_case_attack
[L] https://hashcat.net/hashcat
[M] https://www.openwall.com/john/
[N] https://www.oxid.it/cain.html
[O] https://www.passware.com/kit-forensic/
[P] https://github.com/e-ago/bitcracker
[Q] https://accessdata.com/product-download/password-recovery-toolkit-prtk-version-7.6.0
[R] https://accessdata.com/product-download/distributed-network-attack-dna-version-7.3.0
[S] https://azure.microsoft.com/
[T] https://aws.amazon.com/free/
[U] https://www.nvidia.com/en-gb/data-center/tesla/
[V] https://vast.ai/
[W] https://md5online.org
[X] https://crackstation.net
[Y] https://hashkiller.co.uk

PRZEMYSŁAW RODWALD

Choosing a password breaking strategy with imposed time restrictions

Abstract. The aim of the article is to present the password breaking methodology in case when an attacker (forensic investigator, court expert, pen tester) has imposed time restrictions. This is a typical situation during many legal investigations where computers are seized by legal authorities but they are protected by passwords. At the beginning, the current state of law in that matter is presented, along with good practices in seizing the evidence. Then, the ways of storing static passwords in information systems are showed, after which various classes of password breaking methods are reviewed (dictionary, brute-force, rule, combinator, mask, hybrid, etc.). The most popular tools supporting this process are listed as well. The main part of the paper presents the original strategy of conducting an attack on a single hashed password with time constraints. Costs as well as economic efficiency for four different hardware solutions (laptop, gaming computer, rig with 6 GPU's, cloud computing) are discussed. The calculations are shown on the example of two real attacks carried out by the author in the real legal cases.
Keywords: passwords, breaking passwords, hash functions, computer forensics
DOI: 10.5604/01.3001.0013.1467