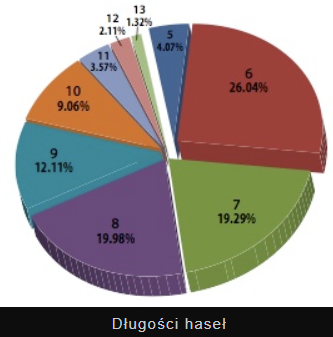
Patryk Gozdera, 277185

BOT – Laboratorium 2

Ataki na hasła

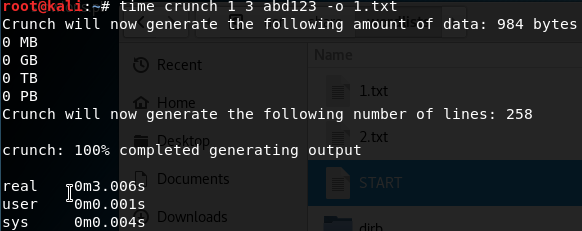
# Słowniki

Plik tekstowy **rockyou** stanowi bazę około 14 mln potencjalnych haseł wykorzystanych przez użytkowników na całym świecie. Słownik ten (a właściwie jego oryginalna wersja – 32 mln haseł), pochodzący z serwisu RockYou.com, jest jednym z najpopularniejszych, możliwych do pobrania. Wnikliwa analiza pliku dostarczyła wiedzy nt. najpopularniejszych haseł, czy najczęściej używanych długości:

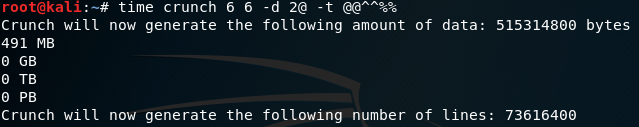


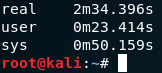
(źródło: <https://niebezpiecznik.pl/post/32-miliony-hasel-wycieklo-jakie-jest-najpopularniejsze/>)

Podczas laboratorium wykorzystywany był program **Crunch** – narzędzie do generowania listy haseł, na podstawie ustalonych przez użytkownika kryteriów. W zależności od wyboru występujących w tworzonym haśle znaków (duże / małe litery, cyfry, znaki specjalne, ich powtórzenia), jak również ich wskazanej ilości, generowanie takiego ciągu może trwać od pojedynczych minut, do godzin. Poniżej znajdują się przykładowe komendy, których użyłem podczas laboratorium:



Minimalna liczba znaków – 1, maksymalna – 3, hasło zbudowane ze znaków „abd123”, zapis słownika do pliku o wskazanej nazwie.



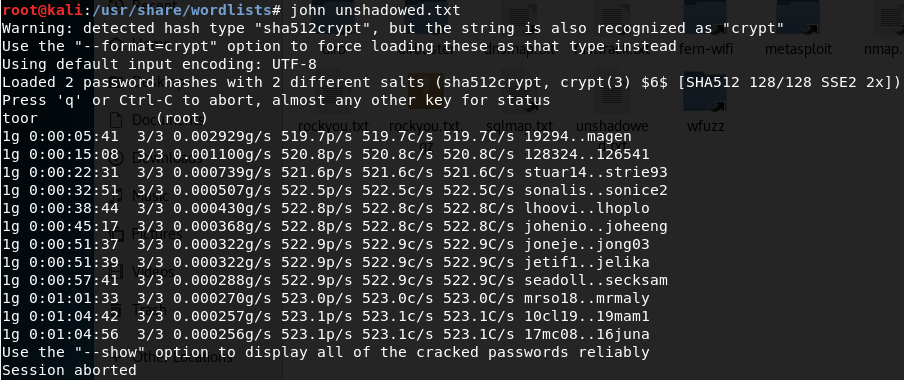
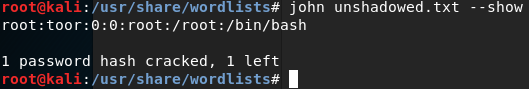
Minimalna liczba znaków – 6, maksymalna – 6, ograniczenie występowania tych samych małych liter do dwóch, -t oznacza użycie wzoru, który kolejno oznacza: 2 małe litery, 2 znaki, 2 cyfry. Jak widać taka operacja trwa już znacznie dłużej, a zapisany słownik zajmuje znacznie większą część przestrzeni dyskowej.

W powyższych przykładach ‘real’ to czas od początku do końca połączenia, ‘user’ – ilość czasu procesora, spędzona w kodzie trybu użytkownika (poza jądrem) w ramach procesu, a ‘sys’ to czas „pobytu” procesora w jądrze.

# John the Ripper

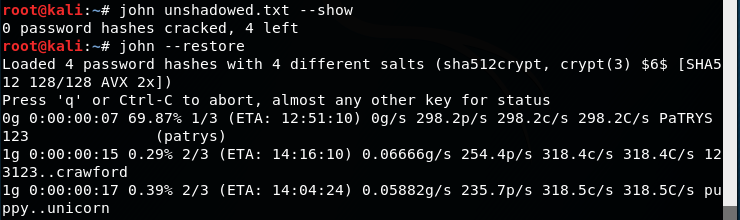
Używane w tej części ćwiczenia narzędzie służy do siłowego łamania haseł (tzw. brute-force).

Po wygenerowaniu pliku z hasłami (komenda: **unshadow /etc/passwd /etc/shadow >unshadowed.txt**), dokonałem próby złamania haseł systemowych z systemu Kali Linux. Owocem tej operacji są poniższe zrzuty:



Ze względu na zbyt długi czas oczekiwania, byłem zmuszony przerwać proces – program poradził sobie z jednym z dwóch haseł.

Następnym etapem zadania była zmiana danych logowania do konta użytkownika. Ustawiłem je w sposób następujący: login: patrys, hasło:123.



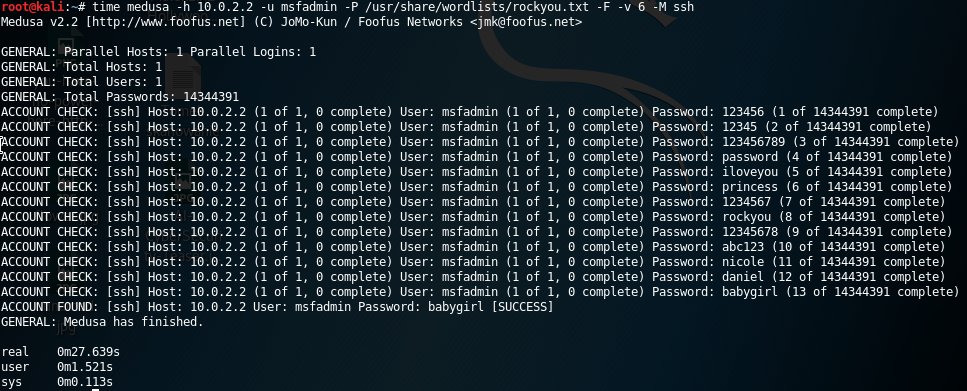
Jak widać program znalazł hasło w mgnieniu oka.

Narzędzie John the Ripper posiada również opcję równoległego łamania haseł, co w znacznym stopniu przyspiesza całą operację. Co więcej, umożliwia on także testować hasła ze wskazanego słownika, używając komendy ‘--wordlist’.

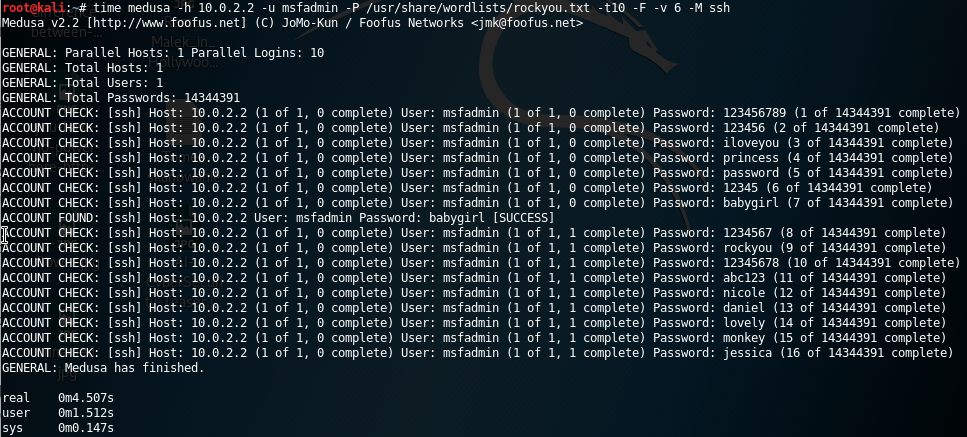
# Medusa i ncrack – porównanie

W tej części laboratorium stosowane są dwa narzędzia do łamania haseł (na przykładzie protokołów SSH oraz FTP) – medusa oraz ncrack. Pierwsze z nich charakteryzuje się okazałą szybkością działania, jest programem masywnie równoległym, wykorzystującym m. in. testy równoległe oparte na wątkach. Z kolei ncrack to program typu open source, używany również do szybkiego równoległego łamania haseł. Wykorzystuje do tego celu dynamiczny silnik, będący uniwersalnym narzędziem, dostosowującym się do konkretnych warunków. Największą różnicą dzielącą oba narzędzia są domyślne ustawienia – ncrack domyślnie działa na wielu wątkach, medusa nie.

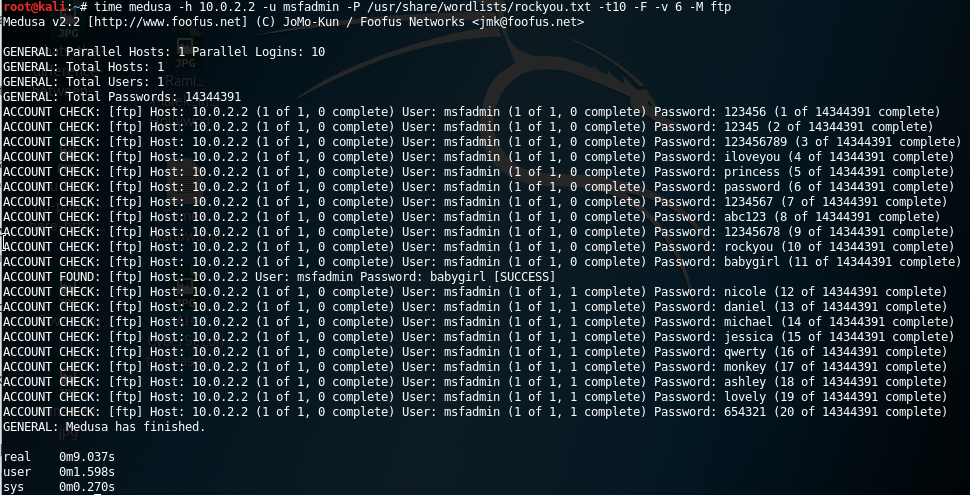
Porównując czas działania obydwu narzędzi w wymienionych wcześniej protokołach, otrzymałem następujące wyniki:



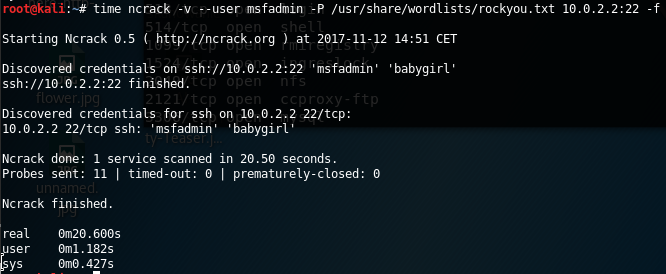
Medusa, domyślne ustawienia, ssh



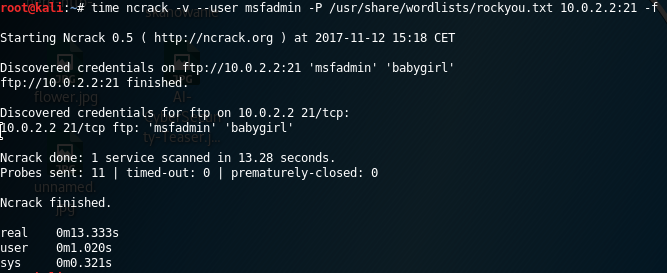
Medusa, 10 wątków, ssh



Medusa, 10 wątków, ftp



Ncrack, domyślne ustawienia, ssh



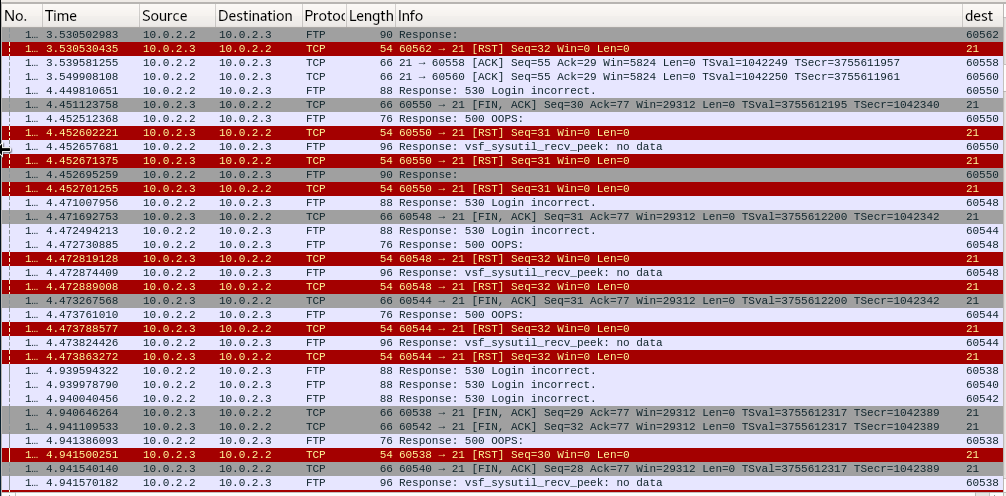
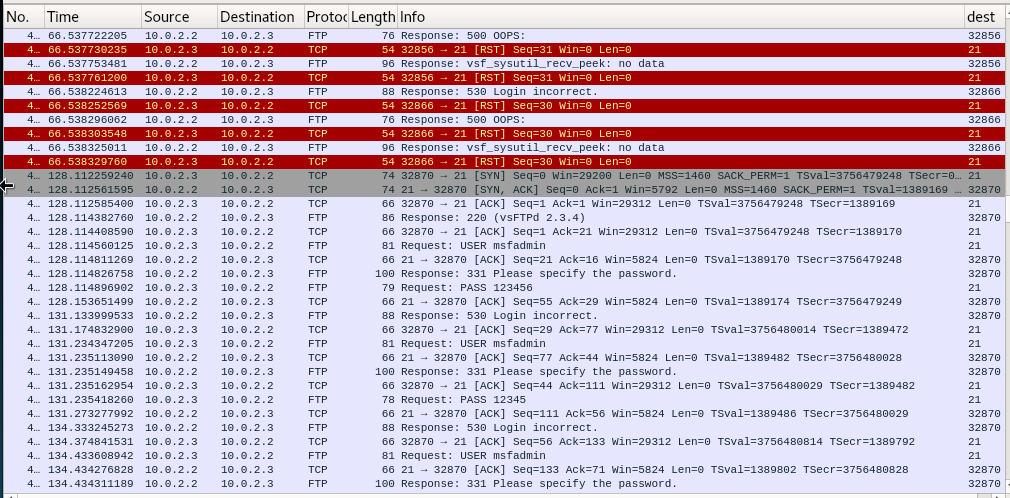
Ncrack, domyślne ustawienia, ftp

Zebrane powyżej dane zawarłem w tabeli:

|  |  |  |  |
| --- | --- | --- | --- |
| Badany protokół | Medusa (domyślnie) | Medusa (10 wątków) | Ncrack |
| SSH | 27,639 s | 4,507 s | 20,600 s |
| FTP | 17,231 s | 9,037 s | 13,333 s |

Jak widać, dla domyślnych ustawień obu narzędzi, ncrack radzi sobie lepiej, niezależnie od zastosowanego protokołu. Gdy jednak ustawimy liczbę wykonywanych wątków w programie medusa na 10, obserwujemy znaczną poprawę tempa wykonywanego procesu – medusa radzi sobie lepiej niż ncrack. Zmiana ustawień ncracka w kwestii wątków nie miałaby sensu, gdyż tak jak zostało wspomniane wcześniej – używa on tej opcji domyślnie.

W tej części łamane było hasło konta msfadmin na maszynie zdalnej z systemem Metasploitable. Szukanym hasłem było ‘babygirl’, a wykorzystywanym słownikiem ‘rockyou.txt’.

Poniżej znajdują się zrzuty ruchu sieciowego z programu Wireshark, generowanego przez używane narzędzia podczas wykonywanych ataków:

Ncrack, ftp

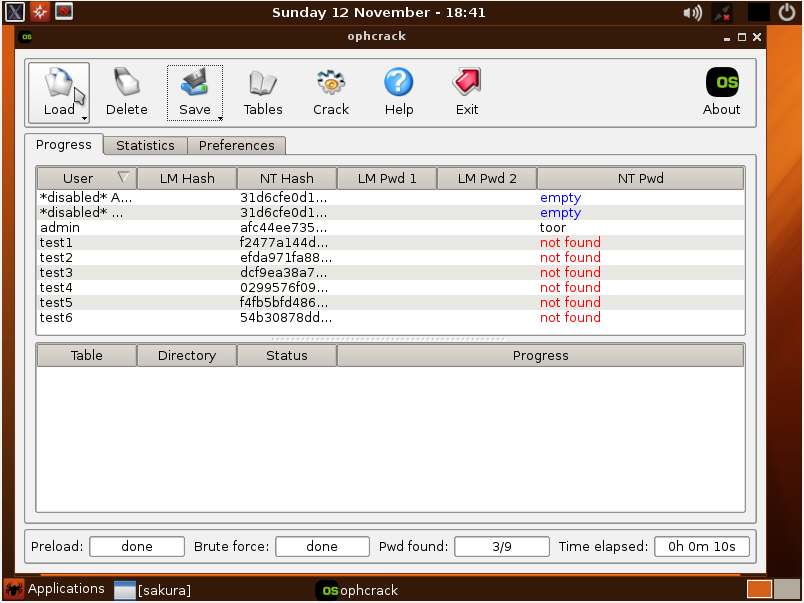
Medusa, ftp

Obserwujemy otwarcie sesji FTP (dla obydwu narzędzi). Wysyłane są kolejne żądania, zawierające badane hasła. W przypadku nieprawidłowego dopasowania, wysyłana jest odpowiedź postaci ‘530 Login incorrect’, a następnie żądanie o ponowne podanie hasła – ‘Please specify the password’. W przypadku dopasowania, wysyłana jest z kolei odpowiedź ‘230 Login successful’ i sesja może zostać zamknięta.

# Tęczowe tablice

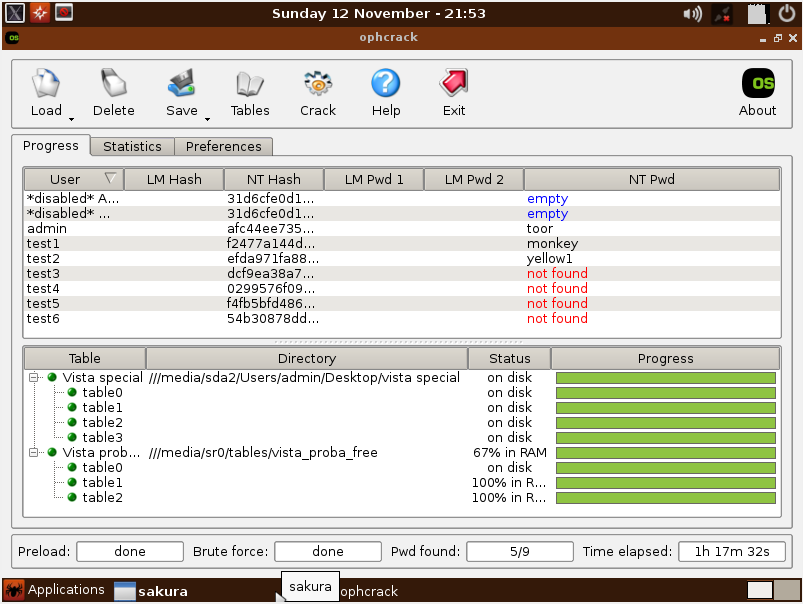
Tęczowe tablice to baza skrótów wykorzystywana w łamaniu haseł zakodowanych jednokierunkową funkcją skrótu. Pozwala ona na zaoszczędzenie mocy obliczeniowej koniecznej do złamania hasła metodą brute force.

Po właściwym skonfigurowaniu obrazu systemu, przystąpiłem do użycia udostępnionych tablic (vista special oraz vista probabilistic free) do łamania haseł kont użytkowników systemu windows 7. Efektem tych działań są poniższe zrzuty ekranu:



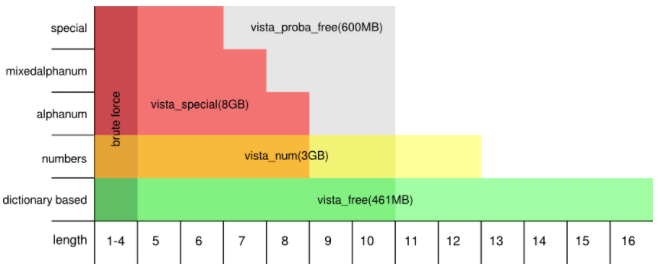
Efekt działania brute force

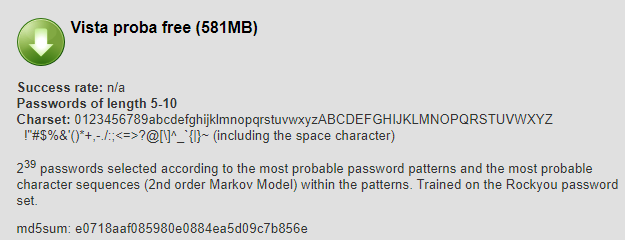
Jak widać brute force zdołał złamać jedynie hasło do konta admin.

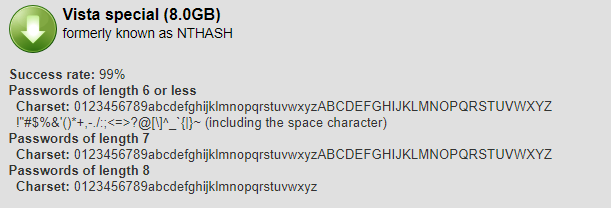


Zakończenie procesu użycia tęczowych tablic

Jak możemy zaobserwować, tablice tęczowe poradziły sobie jedynie z dwoma hasłami. Złamane zostały dane jedynie dla użytkowników test1 oraz test2. Wyjaśnieniem takiego wyniku może być charakterystyka działania tego typu tablic (żródło: <http://ophcrack.sourceforge.net/tables.php>):

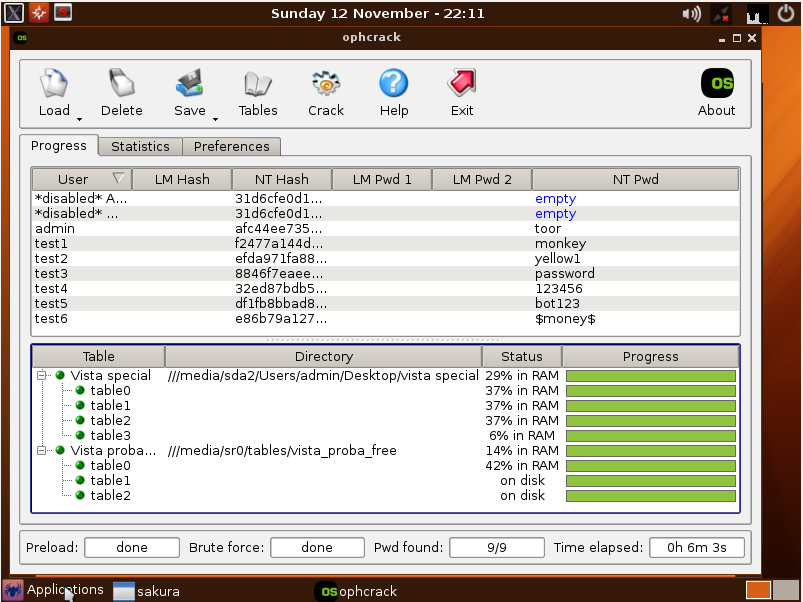






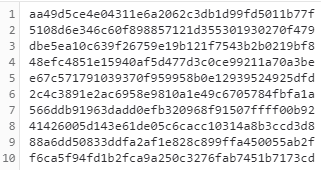
Jak widać, wykorzystywane tablice potrafią z bardzo wysoką skutecznością łamać tylko określone typy haseł. Nie mogą być one bowiem zbyt długie, składać się z polskich znaków, niektórych znaków specjalnych, czy znaków występujących w innych alfabetach niż łaciński. Najprawdopodobniej nieodgadnięte hasła składały się właśnie z takich znaków lub po prostu były zbyt długie.

W kolejnej części tego zadania, dokonałem zmiany haseł użytkowników systemowych, których wcześniej nie udało się złamać. Wybór haseł był ukierunkowany możliwościami tablic, tak aby można było złamać wszystkie hasła, w możliwie szybki sposób. Poniższy zrzut ilustruje wszystkie właściwie dopasowane hasła dla każdego z dostępnych użytkowników:

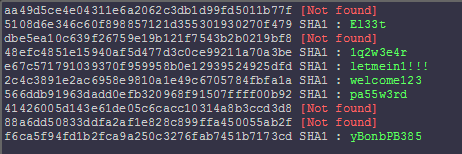


# Narzędzia dostępne w sieci

W ostatniej części tego ćwiczenia laboratoryjnego należało posłużyć się znalezionymi w Internecie serwisami do złamania haseł przechowywanych w bazie danych. Wykorzystywany był w tym celu zrzut bazy danych, prezentujący się następująco:



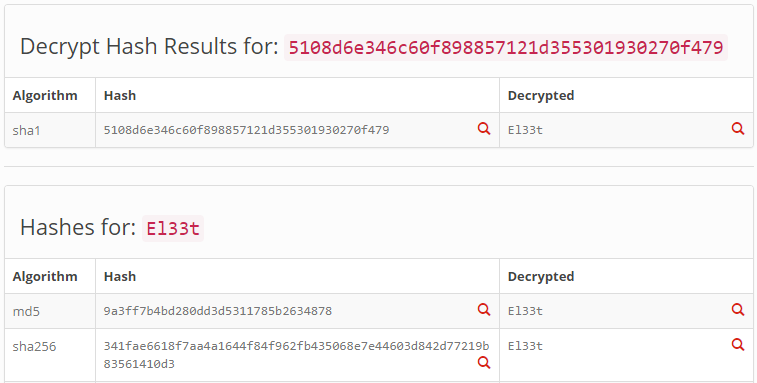
Jest to nic innego jak tablica 10-ciu hashów, o nieznanym sposobie hashowania. Do złamania zawartych w niej haseł użyte zostały 3 przykładowe narzędzia online: <https://hashkiller.co.uk>(który okazał się najefektywniejszy), [www.crackstation.net](http://www.crackstation.net) oraz [www.hashtoolkit.com](http://www.hashtoolkit.com).



Efekt działania hashkiller.co.uk



Efekt działania crackstation.net



Efekt działania hashtoolkit.com

Wszystkie uzyskane za pomocą internetowych narzędzi hasła, zostały otrzymane w wyniku porównywania skrótów, stworzonych przez metodę SHA1 (metoda hashująca, która wytwarza skrót o długości 160 bitów i liczebności 40 znaków; obecnie nieużywana – zastąpiona wersją SHA2).

Hasła, których nie udało się uzyskać, prawdopodobnie posiadały w swoich ciągach różnego typu znaki specjalne, które były niemożliwe do przejścia dla wykorzystywanych narzędzi.

W drugiej części ćwiczenia należało posłużyć się ogólnodostępnym narzędziem do generowania skrótów MD5 i SHA256, a następnie sprawdzić jak wybrane serwisy radzą sobie z hasłami o różnym stopniu skomplikowania. Poniższa tabela ilustruje wyniki przeprowadzonych testów na różnego typu hasłach:

|  |  |  |
| --- | --- | --- |
| Badane hasło | MD5 (czas) | SHA256 (czas) |
| admin | 57 ms | 41 ms |
| AdmIN | 123 ms | 103 ms |
| Adminuś | NOT FOUND | NOT FOUND |
| Adminek12345678 | NOT FOUND | NOT FOUND |
| Admin123 | 797 ms | 612 ms |
| ГгДдДд | NOT FOUND | NOT FOUND |
| aDmi$%n | NOT FOUND | NOT FOUND |

Jak widać, sprawdzane w tej części serwisy radzą sobie tylko z niektórymi hasłami. Mianowicie, gdy hasło składa się z cyfr, małych lub dużych liter i nie jest zbyt długie, możliwe jest jego złamania w bardzo szybkim czasie (poniżej jednej sekundy). Gdy natomiast wzbogacimy je o polskie znaki, wydłużymy jego długość w znacznym stopniu lub zastosujemy elementy z alfabetu innego niż łaciński (w tym przypadku cyrylica), badane serwisu nie radzą sobie ze złamaniem takiego hasła. Dzieje się tak w głównej mierze ze względu na ograniczone bazy znanych skrótów, które fizycznie mogą powstać. Znacząca większość takich haseł pochodzi z języka angielskiego, stąd ograniczone możliwości tych programów. Być może należałoby skorzystać ze specjalnie stworzonej bazy dla znaków ukierunkowanych na konkretną grupę (np. właśnie cyrylicę). Niemniej jednak, język ten jest na tyle popularny na świecie, że narzędzia te stwarzają naprawdę duże możliwości łamania danych logowania.

Powyższe wyniki pokazują, że warto w swoim haśle zawierać jakieś znaki specjalne i tworzyć je dostatecznie długie, aby zwiększyć bezpieczeństwo swojego konta.