Metody włamań do systemów komputerowych

Robert Michalak

12 stycznia 2003

1 Słabe hasła

Najsłabszym ogniwem w dokonywaniu autoryzacji użytkownika poprzez system haseł jest sam użytkownik. Oto kilka podstawowych błędów popełnianych przez użytkowników przy wyborze hasła:

- Hasło takie samo jak *Username*.
- Hasło jest datą urodzenia, imieniem żony/męża, dziecka, kota, psa itp.
- Czasami ludzie są na tyle leniwi, że hasła nie mają (tylko ENTER).
- Hasło jest słowem występującym w słowniku.
- Hasło jest kombinacją liter występujących blisko siebie na klawiaturze.
- Hasło jest zapisane na kartce. (najczęściej kartka naklejona na monitorze, co przezorniejsi ukrywają ją pod klawiaturą lub za komputerem)

Jeżeli użytkownik wymyśli dobre hasło, to złamać go będzie można jedynie poprzez wypróbowanie wszystkich możliwych haseł. Atak taki nazywany jest metodą „brute force". Jednak większość haseł wybieranych przez użytkowników nie jest hasłami „dobrymi" w powyższym znaczeniu, przez co ich złamanie jest o wiele łatwiejsze.

2 Łamanie haseł

Najczęściej stosowaną metodą w łamaniu haseł jest ich zgadywanie. Chociaż wydaje się, że metoda ta wymaga dużej dozy szczęścia do powodzenia, to jednak często bywa skuteczna.

2.1 Metoda „brute force"

Jak już wspomniałem wcześniej, jedną z metod łamania haseł jest metoda „brute force". Jest to metoda najbardziej skuteczna ze wszystkich, ponieważ mamy pewność, że wcześniej czy później znajdziemy właściwe hasło. Niestety swoją skuteczność okupuje długim czasem działania. Czas ten zależy oczywiście od szybkości komputera, ale także od systemu haseł stosowanego w łamanym systemie. Czasami ten czas jest na tyle długi, że metoda ta jest niemożliwa do zastosowania w praktyce.

2.2 Metoda słownikowa

Metoda ta opiera się na „słabości" haseł. Wykorzystuje ona fakt, że duża część użytkowników nie trudzi się z wymyślaniem skomplikowanego hasła które trudno jest zapamiętać, i wybiera słowo występujące w słowniku. Program łamiący hasła metodą słownikową sprawdza właśnie jako potencjalne hasła słowa ze słownika. Oczywiście słów tych jest o wiele mniej niż wszystkich kombinacji literowych, zatem programy tego typu działają znacznie szybciej od metod typu „brute force", ma jednak także mniejszą skuteczność.

Aby zwiększyć skuteczność metody słownikowej dodaje się do niej drobne poprawki, dzięki którym jest w stanie rozpoznać więcej haseł. Możliwymi ulepszeniami stosowanymi przez użytkowników w celu utrudnienia pracy tego typu programom mogą być:

- dodanie cyfry jako prefiksu czy sufiksu słowa.
- wpisywanie słowa od końca.
- wpisywanie słowa z różnymi kombinacjami wielkich i małych liter.
- różne kombinacje powyższych.

Są to wprawdzie utrudnienia, ale sprytny program potrafi sobie z nimi poradzić.

3 Hasła w systemie Windows 95

Systemy operacyjne Windows 9x przechowują informacje o hasłach w plikach *<username>.pwl*. Każdy użytkownik ma zatem własny plik *PWL*. W momencie, gdy ma nastąpić autoryzacja, system operacyjny tworzy z wprowadzonego przez użytkownika hasła klucz, którym to kluczem odszyfrowywane są informacje zawarte w pliku *PWL*. Po odszyfrowaniu ciąg bajtów porównywany jest z ciągiem "wzorcem", który był wygenerowany w momencie zakładania konta. Jeżeli porównanie wypadnie pomyślnie, to użytkownik uzyskuje dostęp do systemu.

W różnych wersjach systemu wykorzystywane są różne algorytmy szyfrujące. Dla Windows 95 jest to algorytm RC4, dla Windows 95 OSR2 i Windows 98 są to algorytmy RC4 i MD5.

Pomimo tego, że szyfr RC4 uważany jest za bezpieczny, to włamanie do systemu Windows jest prostsze niż wskazywałoby na to bezpieczeństwo RC4. RC4 generuje swój klucz, który wykorzystywany jest przy szyfrowaniu, na podstawie ciągu bajtów dostarczanego mu przez system Windows. Ciąg ten powstaje poprzez operacje na haśle użytkownika. Problem w tym, że Windows ogranicza długość tego ciągu do czterch bajtów.

Wystarczy zatem sprawdzić 2^{32} możliwości kluczy, aby odnaleźć ten właściwy. Nie rozwiązuje to wprawdzie do końca problemu znalezienia hasła, ale teraz wystarczy metodą słownikową lub „brute force" przeanalizować różne hasła i porównać ich czterobajtowy kod z kodem znalezionym w poprzednim kroku. Metoda ta jest bardzo szybka ze względu na to, że bardzo szybko działa kodowanie hasła do długości czterech bajtów.

Microsoft zauważył tą usterkę, więc w kolejnych wersjach systemu wprowadził poprawki mające na celu zwiększenie bezpieczeństwa:

- Przestrzeń kluczy zwiększyła się do 2^{128}
- wykorzystano dodatkowo jednokierunkową funkcję haszującą MD5.

W chwili obecnej w internecie można spotkać programy analizujące do 200 tysięcy haseł w ciągu sekundy. Przy takiej szybkości niemożliwe staje się przeprowadzenie ataku metodą „brute force", nadal jednak metoda słownikowa sprawdzi się w wielu przypadkach ze względu na czynnik ludzki.

4 Hasła w systemie Windows NT

Do przechowywania haseł w systemie NT służy rejestr. Hasła przechowywane są tam w postaci zaszyfrowanej jednokierunkową funkcją haszującą MD4, której wynik był dodatkowo szyfrowany algorytmem, którego zadaniem było „zaciemnić" działanie MD4. Wszystko było w porządku do czasu, gdy została znaleziona funkcja odwrotna do funkcji „zaciemniającej". Po tym fakcie cały system haseł w Windows NT stał się tak bezpieczny, jak bezpieczna jest funkcja MD4. Jednokierunkowość tej funkcji oznacza, że znając jej wartość nie jesteśmy w stanie podać arumentu, dla którego ta wartość jest przyjmowana.

W praktyce okazało się, że założenia o jednokierunkowości funkcji nie do końca się sprawdziły. Wprawdzie nikomu jeszcze nie udało się podać jej odwrotności[1], jednak są już znane odwrotności dla niektórych jej wartości (np. dla 0). Dlatego nie można uważać tej funkcji za bezpieczną.

[1] a przynajmniej nie została ona podana publicznie

5 Hasła w systemach UNIX

Jak wszyscy wiemy hasła w systemach UNIX przechowywane są w pliku */etc/passwd* bądź */etc/shadow*. Są tam jednak przechowywane w postaci zaszyfrowanej. Ponieważ algorytm szyfrujący jest jednokierunkowy, więc nie można na podstawie zaszyfrowanego hasła znaleźć jego niezaszyfrowanego odpowiednika. Pozostają więc jedynie programy do łamania haseł, a razem z nimi metody słownikowe i „brute force". Te jednak mają różną skuteczność w zależności od konkretnego systemu. Na przykład w starszych systemach operacyjnych było wprowadzone ogranicznie na długość hasła - maksymalnie 8 znaków. Znajomość takiego ograniczenia znacznie ułatwia atakującemu odgadnięcie hasła - w tym wypadku wystarczy zwykłe sprawdzenie wszystkich możliwości - nie jest ich wcale tak dużo.

Ogólnie systemy Unix'owe są uważane za najbardziej bezpieczne systemy operacyjne. Nie istnieją (a przynajmniej nie są opisane) bardziej wyrafinowane metody łamania haseł niż kolejne wariacje metody słownikowej i ..brute force".

6 Podsumowanie

Niezależnie od tego, jaki system kryptograficzny został zastosowany w systemie, zawsze najsłabszym ogniwem w całym procesie autoryzacji jest człowiek. W większości przypadków właśnie dzięki „słabym" hasłom można uzyskać dostęp do systemu. Aby temu zaradzić, administratorzy stosują coraz bardzie wyrafinowane systemy sprawdzania „siły" haseł. Jedną z metod przy ustalaniu nowego hasła jest odrzucanie haseł opartych na słowach ze słownika, a także takich, które mają zbyt mało różnych liter bądź są za krótkie. Nie jest to jednak całkowite rozwiązanie problemu, gdyż zmuszanie użytkownika do wpisywania trudnych do zapamiętania haseł może spowodować to, że będzie on takie hasło zapisywał. A nawet najlepsze hasło, gdy zostanie gdzieś zapisane, można uznać za złamane.

Cały system jest tak bezpieczny, jak nasłabszy jego element. Sprawdza się to także w przypadku włamań do systemów komputerowych. Co nam po najlepszym nawet systemie zabezpieczeń, jeżeli klienci łączą się z serwerem poprzez program *telnet*, w którym hasło przesyłane jest tekstem jawnym.

Wydawać by się mogło, że jeżeli zastosujemy dobry, sprawdzony algorytm kryptograficzny, to tworzony przez nas system będzie bezpieczny. Jest to często wrażenie mylne, gdyż oprócz samego algorytmu liczy się także jakość jego implementacji. W chwili obecnej najczęstszymi atakami kryptograficznymi są właśnie ataki na konkretną implementację, w której zauważono lukę, choć sam zastosowany algorytm jest bezpieczny.

6.1 Przyszłość

Powstają coraz szybsze komputery, tak więc to, co dzisiaj zajmuje 10 lat niedługo być może będzie zajmować kilka sekund. Należy o tym pamiętać przy tworzeniu systemów zabezpieczeń. Z czasem ataki typu „brute force" mogą stać się łatwiejsze do przeprowadzenia.

Coraz więcej mówi się o komputerach kwantowych. Choć na razie nie widać żadnego prototypu, to znane są już algorytmy na ten typ komputera. Do ciekawszych niewątpliwie należy algorytm faktoryzacji liczb pierwszych Shor'a. Gdyby udało się zbudować komputer kwantowy dzisiejsza kryptografia oparta na faktoryzacji dużych liczb pierwszych (m.in. RSA) ległaby w gruzach.

6.1.1 timing i power attack

Obiecujące podejście w kwestii łamania szyfrów opracował Kocher. Wymaga ona od atakującego dokładnego mierzenia czasu obliczeń. Przyjrzyjmy się tej metodzie na przykładzie deszyfrowania RSA.

Zakładamy, że szyfrujący zna mnóstwo kryptogramów i może obesrwować proces deszyfrowania przez program (lub chip), tzn. zmierzyć czasy wykonywania określonych zadań. Deszyfrowanie RSA wymaga obliczenia wartości wyrażenia $R = c^d \mathrm{mod} n$ Potęgę tę można przykładowo wyliczyć w następujący sposób:

1. Podstawiamy $R = 1$ i przebiegamy wszystkie bity liczby d, począwszy od najniższego.

2. Jeśli przetwarzany bit d ma wartość 1, wówczas mnożymy rezultat częściowy R przez c; wpp. R pozostaje niezmienione.

3. Zamieniamy c przez jego kwadrat i przechodzimy do kolejnego bitu liczby d

Jak widać, czas obliczenia drugiego kroku zmienia się w zależności od wartości bitu d. Jeżeli był on równy 1 to wykonywane jest mnożenie, jeżeli 0 - nic nie jest wykonywane. Na tej podstawie można wnioskować o postaci liczby d.

Na podobnej zasadzie działa *power attack*. Wykorzystuje się w nim wahania poboru mocy przez kartę chipową. Na tej podstawie można np. odróżnić operację mnożenia od podnoszenia do kwadratu na kartach RSA.

Oczywiste jest, jaki to stanowi problem dla systemów, do których dostęp uzyskuje się po włożeniu w czytnik odpowiedniej karty.