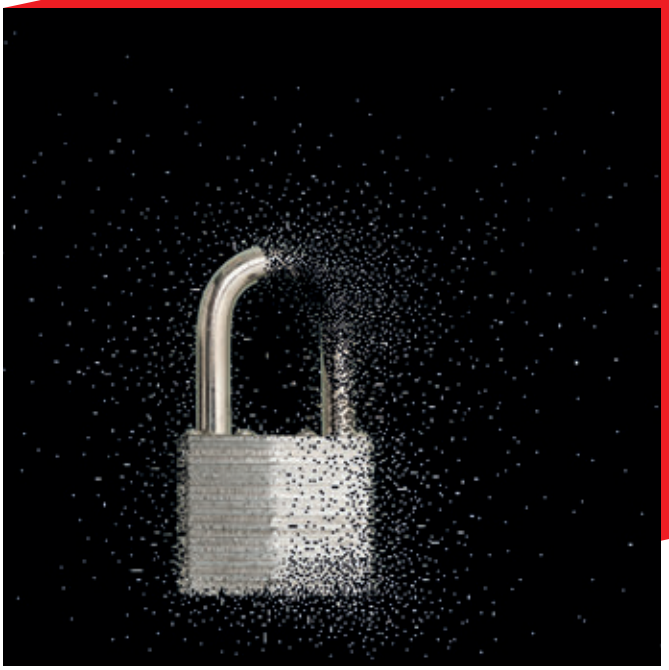
Şifre Güvenliği

Dr. Mahir E. Ocak [*TÜBİTAK Bilim ve Teknik Dergisi*

Şifre kırmaya çalışan hackerlar sadece rastgele şifrelerle şanslarını denemiyor, hem daha önceki şifre kırma olaylarında edinilmiş bilgilerden yararlanıyor hem de karmaşık algoritmalar kullanıyorlar.

Hackerların işlerini zorlaştırmaksa kullanıcılara ve yazılımcılara düşüyor.



Kaba Kuvvetle Şifre Kırma

Şifre kırmayı deneyen bir hackerın başvurabileceği en kaba yöntem tek tek bütün olasılıkları denemektir. Dolayısıyla muhtemel şifrelerin sayısı ne kadar çoksa hackerların işi de o kadar zordur. Pek çoğumuzun başına gelmiştir. İnternette bir siteye üye olmaya çalışırken sistem yazdığınız şifreyi beğenmez, zayıf bulduğunu söyler ve sizi daha güçlü bir şifre belirlemeye zorlar. Örneğin, şifrenin içinde en az bir büyük harf, en az bir rakam ve en az bir noktalama işareti olmasını ister. Her ne kadar sinir bozucu olsa da bu durum kullanıcının güvenliğini sağlamak ve hackerların işini zorlaştırmak içindir. Çünkü şifre belirlerken kullanılan karakterlerin sayısı arttıkça muhtemel şifrelerin sayısı da artar.

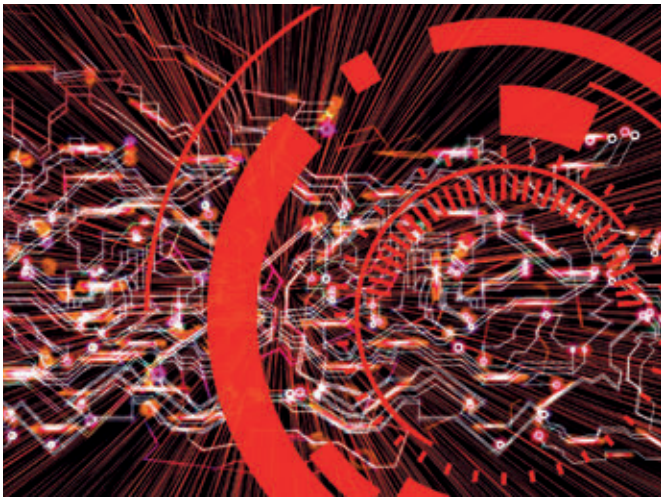
Sadece küçük harflerden oluşan, altı karakter uzunluğundaki şifreleri ele alalım. Şifre oluşturulurken genellikle sadece İngiliz alfabesindeki 26 harf kullanıldığı ve her bir karakter bu 26 harften herhangi biri olabileceği için $26*26*26*26*26*26 = 26^6 \approx 3*10^8$ farklı şifre olabilir. Şimdi de sadece küçük harfler değil, büyük harfler, rakamlar ve on ayrı sembol (örneğin, !, @, #, $, %, ^, &, *, /, ve =) içeren altı karakter uzunluğundaki şifreleri ele alalım. Her bir karakter için 26+26+10+10 = 72 ayrı ihtimal vardır. Dolayısıyla muhtemel tüm şifrelerin sayısı $72^6 \approx 1{,}4*10^{11}$'dir. Yani, kullanılan karakterlerin sayısı 26'dan 72'ye çıktığında muhtemel şifrelerin sayısı yaklaşık 500 katına çıkar. On karakter uzunluğundaki şifreler ele alındığındaysa oran çok daha büyüktür. Sadece 26 küçük harfle oluşturulabilecek şifrelerin sayısı $26^{10} \approx 1{,}4*10^{14}$ iken, 72 karakterle oluşturulabilecek şifrelerin sayısı $72^{10} \approx 3{,}7*10^{18}$'dir. Yani, kullanılan karakterlerin sayısı 26'dan 72'ye çıktığında, muhtemel şifrelerin sayısı yaklaşık 250.000 katına çıkar. Dolayısıyla birinci durumda, bir hacker sadece 1 saniye içinde şifre kırmayı becerebiliyorsa, ikinci durumda neredeyse 3 gün uğraşması gerekecektir.

Muhtemel şifrelerin sayısını ifade ederken "şifre uzayının" büyüklüğünden ve entropisinden bahsedilir. *A* tane farklı karakterle oluşturulabilecek *N* uzunluktaki şifrelerin sayısı $T = A^N$'dir. Şifre uzayının entropisi $1+[\log_2 T]$ olarak tanımlanır. Bu ifadede köşeli parantezler, içerisindeki sayının kendisinden küçük ve kendisine en yakın tam sayıya yuvarlanması gerektiği anlamına gelir. Başka bir deyişle entropi, uzayın büyüklüğünü 2 tabanında ifade etmek için gerekli bitlerin sayısıdır. Örneğin, 26 karakterle oluşturulabilecek altı karakterli şifre uzayının entropisi $1+[\log_2 26^6] = 1+[28,20] = 29$ bittir. 72 karakterle oluşturulabilecek on karakterli şifre uzayının entropisi $1+[\log_2 72^{12}] = 1+[61,69] = 62$ bittir.

Günümüzde 64 bitlik ve daha küçük uzaylar çok küçük, 64-80 bitlik uzaylar küçük, 80-100 bitlik uzaylarsa orta derecede büyük kabul ediliyor. Bir şifrenin en azından birkaç yıl güvenli olabilmesi içinse şifre uzayının entropisinin en azından 128 bit olması gerektiği tavsiye ediliyor. Tabloda görüldüğü gibi 26 karakterle oluşturulan 6 karakter uzunluğundaki şifreleri kırmak hiç de zor değildir. Bugünün teknolojisiyle 100 bitlik bir şifre uzayı için gereken zamansa evrenin yaşından bile daha büyüktür. Bu büyüklükte bir uzaydaki şifreleri sadece bir saat içinde kırabilecek bir bilgisayarın geliştirilmesi için de yaklaşık 115 yıl beklemek gerekecektir. Tablodaki değerlere bakıldığında "birkaç yıllık güvenlik" için önerilen 128 bitlik uzayların abartılı derecede büyük olduğu düşünülebilir. Ancak tablodaki değerlerin tamamı tek bir bilgisayar ile kaba kuvvet kullanarak şifre kırmaya çalışan bir hackerın ihtiyacı olan zamandır. Ancak hackerın elinde 1000 bilgisayar varsa gerekli zaman binde birine düşecektir. Oysa daha da önemlisi hackerların genellikle kaba kuvvetle şifre kırmaya çalışmadığıdır. Hackerlar daha kısa süre içinde şifre kırmalarına imkân verecek daha karmaşık yöntemler kullanırlar. Ayrıca kullanıcılar ve sistem yöneticileri de bazen tutum ve davranışlarıyla hackerların işini kolaylaştırır.

A	N	T	S	D	X
26	6	$3,08 \times 10^8$	29 bit	0,31 saniye	0 yıl
26	10	$1,41 \times 10^{14}$	48 bit	39,21 saat	10 yıl
100	10	$10,0 \times 10^{20}$	67 bit	3171 yıl	49 yıl
100	15	$10,0 \times 10^{30}$	100 bit	$3,17 \times 10^{13}$ yıl	115 yıl
200	20	$1,05 \times 10^{46}$	153 bit	$3,33 \times 10^{29}$ yıl	222 yıl

Yukarıdaki tabloda saniyede bir milyar şifreyi test edebilme kapasitesine sahip bir bilgisayarı olan, kaba kuvvetle şifre kırmaya çalışan bir hackerın farklı büyüklükteki şifre uzaylarındaki olası tüm şifreleri denemesi için gerekli süreler veriliyor. Tabloda *A* şifre oluştururken kullanılabilecek farklı karakterlerin sayısını, *N* şifredeki karakterlerin sayısını, *T* olası tüm şifrelerin sayısını, *S* şifre uzayının entropisini, *D* olası tüm şifreleri denemek için gerekli zamanı gösteriyor. Tablonun en son sütununda verilen *X* değerleriyse, bilgisayarların kapasitesinin her iki yılda bir iki katına çıktığını söyleyen Moore yasasının doğru olduğu var sayıldığında, kaç yıl sonra üretilecek bir bilgisayarın sadece bir saat içinde olası tüm şifreleri test edebileceğini gösteriyor.



Şifre Sözlükleri

Birkaç yıllık güvenlik için tavsiye edilen 128 bitlik şifre uzayları için nasıl bir şifreye ihtiyacımız olduğunu hesaplamaya çalışalım. Eğer 26 karakter kullanılarak şifre oluşturuluyorsa, şifrenin uzunluğu en azından 28 karakter olmalıdır. Bu sayı, kullanılan karakterlerin sayısı 72'ye çıktığında 21'e, 200'e çıktığındaysa 17'ye düşer. Şifrenin uzunluğunu 10 ile sınırlamak istediğindeyse yaklaşık 7150 karakter arasından seçim yapılması gerekir. Görüldüğü gibi güvenli bir şifre hem uzun olmalı hem de çok çeşitli karakterler içermelidir.

Kendiniz için bir şifre oluşturduğunuzu düşünün. Sistem sizden 20 karakter uzunluğunda bir şifre seçmenizi isterse ne yaparsınız? Eğer şifrenizin güvenli olmasını istiyorsanız yaptığınız karakter tercihlerinin tamamen rastgele olması gerekir. Ancak böyle bir şifreyi akılda tutmak da pek kolay değildir. Bu yüzden insanların büyük çoğunluğu kolayca akılda tutabilecekleri bir karakter dizisi belirlerler. İsimler, soy isimler, doğum tarihleri, anlamlı kelimeler ve tümceler... Bu durum hackerların işini hayli kolaylaştırır.

Hackerlar, geçmişte yaşanmış büyük çaplı şifre kırma olayları sırasında ele geçirilen şifreleri bir araya getirir, kullanılma sıklıklarına göre tasnif eder ve rastgele şanslarını denemek yerine genellikle "şifre sözlükleri" olarak adlandırılan bu listeleri kullanırlar.

2017 yılında ele geçirilmiş,
beş milyon kullanıcının şifrelerinin yer aldığı
bir veri tabanında en sık rastlanan
25 şifre şunlardır:

1.	123456	16.	starwars
2.	password	17.	123123
3.	12345678	18.	dragon
4.	qwerty	19.	passw0rd
5.	12345	20.	master
6.	123456789	21.	hello
7.	letmein	22.	freedom
8.	1234567	23.	whatever
9.	football	24.	qazwsx
10.	iloveyou	25.	trustno1
11.	admin		
12.	welcome		
13.	monkey		
14.	login		
15.	abc123		

Bu listedeki şifrelerin tamamı ya basit kelimeler (password, login) ya basit söz dizileri (letmein, iloveyou) ya basit sayı dizileri (123456, 12345678) ya da klavyedeki ardışık harflerle oluşturulmuş basit karakter dizileridir (qwerty, qazwsx). Bu listelerin içeriği ülkeye, zamana ya da nereden ele geçirildiğine bağlı olarak değişir. Ancak insanların şifre seçerken ne kadar özensiz davrandıklarını göstermeleri açısından önemlidirler.

Söz konusu olan cep telefonu pinleri gibi dört rakamlı şifreler olduğunda da durum benzerdir. DataGenetics Web sayfasının 2013 yılında yayımladığı bir rapora göre insanların %11'inin şifresi 1234'tür, %6'sının şifresi ise 1111'dir.

Rastgele bir şifre belirlemek yerine kolay akılda tutulabilecek tercihler yapılması, taranması gereken şifre uzayını çok küçültür ve saldırganların işini hayli kolaylaştırır.

Hash Fonksiyonları

Bir saldırganın işini kolaylaştıracak en önemli şeylerin başında sisteme erişim gelir. Çünkü kullanıcıların şifreleri ile ilgili bilgiler eninde sonunda sistemde kayıtlı olmalıdır. Eğer kullanıcıların şifreleri sistemde "çıplak" olarak kayıtlıysa saldırgan sisteme erişerek edindiği bilgileri kullanıp sorunsuzca istediğini yapabilir. Ancak kullanıcı şifrelerini çıplak olarak sistemde tutmak, her ne kadar nadir bir durum olduğu söylenemezse de, kötü bir sistem yöneticiliği örneğidir. Güvenli sistemlerde, kullanıcı şifrelerinin kaydı tutulurken genellikle Hash fonksiyonlarından yararlanılır.

Hash fonksiyonları, girdi olarak bir karakter dizisi alıp çıktı olarak da yine bir karakter dizisi verir. Bu fonksiyonların en önemli özelliği, "parmak izi" olarak adlandırılan çıktılara bakarak girdinin ne olduğunu belirlemenin pratikte neredeyse imkânsız olmasıdır.

Günümüzde güvenli olarak kabul edilen ve yaygın olarak kullanılan Hash fonksiyonlarından biri, SHA256 olarak adlandırılır. Bu fonksiyon herhangi bir karakter dizisini A, B, C, D, E ve F harfleri ile 0'dan 9'a kadar rakamlardan oluşan 64 karakter uzunluğunda bir diziye dönüştürür. Örneğin, fonksiyona girdi olarak "sifre" dizisini verdiğimizde elde ettiğimiz parmak izi şudur:

8F37491F3A3539F0B9A6306EFB9EFE520251A15B4D6317B741DE73AE05F3A4F5.

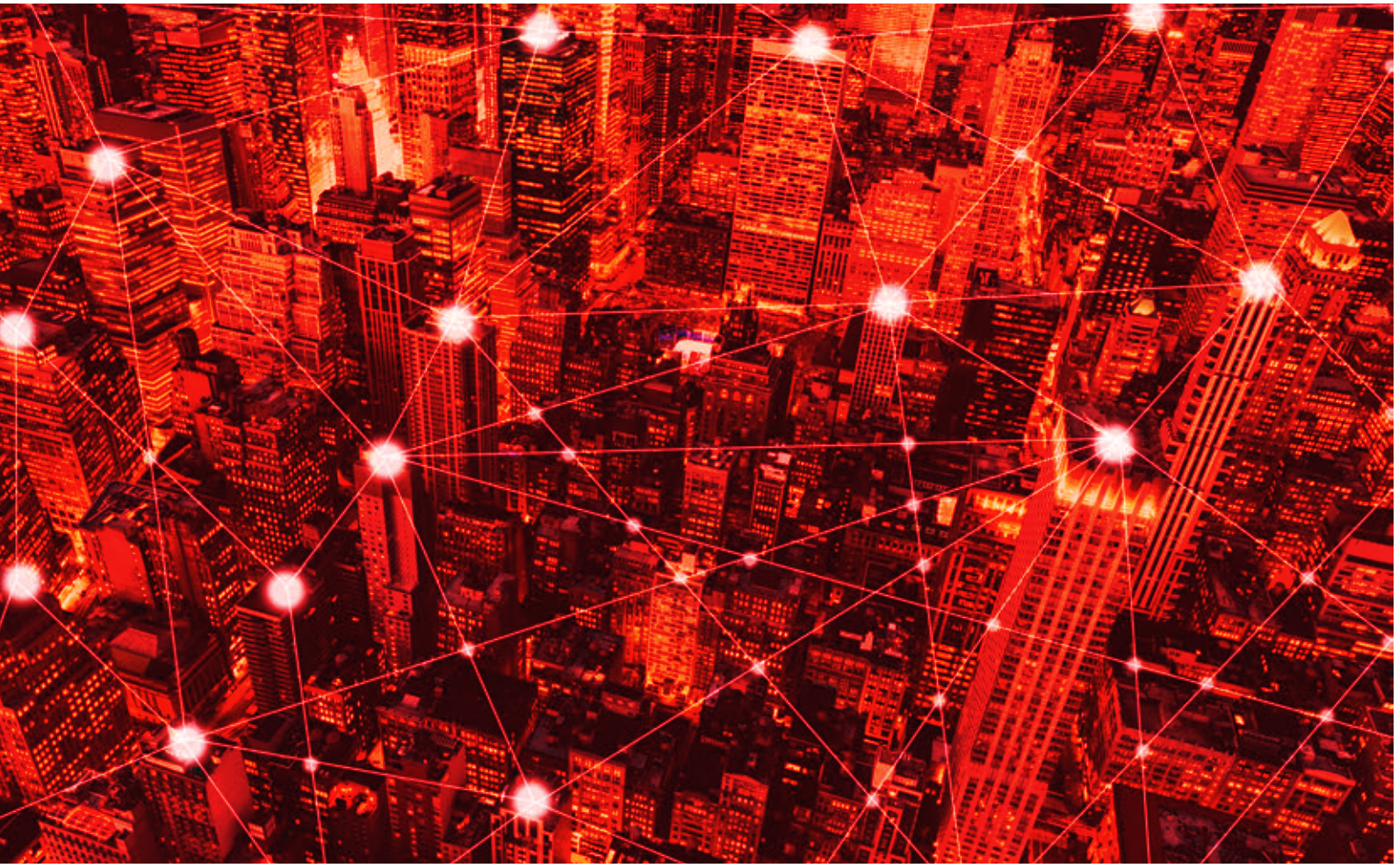
SHA256 fonksiyonuna girdi olarak "Sifre" dizisini verdiğimizdeyse şu parmak izini elde ederiz:

5FD6A283EEFEF9CE6704F8B5775D42C84CF7E8FDE344AF3C10602C7116556C33.

Kullandığımız girdiler arasında çok küçük bir fark var: sadece küçük s harfini büyük S harfi ile değiştirdik. Ancak parmak izlerine baktığımızda aralarında çok büyük farklar olduğunu görürüz. Hash fonksiyonlarının en önemli özelliği de tam olarak budur. Girdiler birbirine ne kadar benzerse benzesin çıktılar birbirinden çok farklıdır. Bu yüzden parmak izlerine bakarak fonksiyona verilen girdileri tahmin etmek imkânsız denilebilir. Siz de şu adresteki Hash üreticisini kullanarak bu durumu test edebilirsiniz:

https://passwordsgenerator.net/sha256-hash-generator/.

Güvenli bir sistemde, kullanıcı hesabı oluştururken bir Hash fonksiyonu kullanılarak girdiği şifrenin parmak izi üretilir ve kaydedilir. Çıplak şifreninse ne sistemde ne de başka bir yerde kaydı tutulmaz. Kullanıcı sisteme ileri bir tarihte tekrar giriş yapmak istediğinde, yine aynı Hash fonksiyonu kullanılarak girdiği şifrenin parmak izi üretilir ve bu parmak izi sistemde kayıtlı olan parmak izi ile karşılaştırılır. Eğer parmak izleri uyuşuyorsa girilen şifrenin doğru olduğuna karar verilir ve sisteme giriş yapılmasına izin verilir. Aksi durumdaysa sisteme giriş yapılmasına izin verilmez.



Tuzlama

İnsanların şifre belirlerken özensiz tercihler yapmasına ve sistemde kayıtlı bilgilerin ele geçirilmesine karşı geliştirilmiş bir önlem "tuzlama"dır. Sistem, kullanıcının hesap oluştururken belirlediği şifrenin sonuna "tuz" olarak adlandırılan rastgele bir karakter dizisi ekler. Kullanıcı tuzun ne olduğunu bilmez. Kullanıcı sisteme giriş yapmak istediğinde, önce kullanıcı tarafından yazılan şifre otomatik olarak "tuzlanır" daha sonra da tuzlu şifrenin parmak izi hesaplanır. Tuzlamanın kullanıldığı bir sistemde her kullanıcı ile ilgili üç bilgi kayıt altında tutulur: kullanıcı adı, tuz ve tuzlanmış şifrenin parmak izi. Farklı kullanıcılar için farklı tuzlar kullanıldığından iki kullanıcının çıplak şifreleri aynı olsa bile tuzlanmış şifrelerinin parmak izi farklı olur.

Şifrelerin tuzlanması saldırganların işini hayli zorlaştırır. Örneğin, sistemin her şifrenin sonuna 200 karakter arasından seçilmiş, üç karakter uzunluğunda bir tuz eklediğini düşünelim. Böyle bir sistemde bir saldırganın şifre sözlüklerindeki şifreleri tek tek deneyerek başarılı olma şansı hiç yoktur. Yapması gereken şey sözlükteki her bir şifrenin sonuna olası tüm tuzları ekleyip öyle denemeler yapmaktır. Olası tüm tuzların sayısı 200^3=8.000.000 olduğu için bu durum şifreyi kırmak için harcanacak zamanın sekiz milyon katına çıkacağı anlamına gelir. Eğer saldırganın elinde şifre sözlüklerindeki tüm şifreleri sadece bir saat içinde denemesine imkân veren bir teknoloji varsa bile tüm tuzlanmış şifreleri denemesi yaklaşık 900 yıl sürecektir.

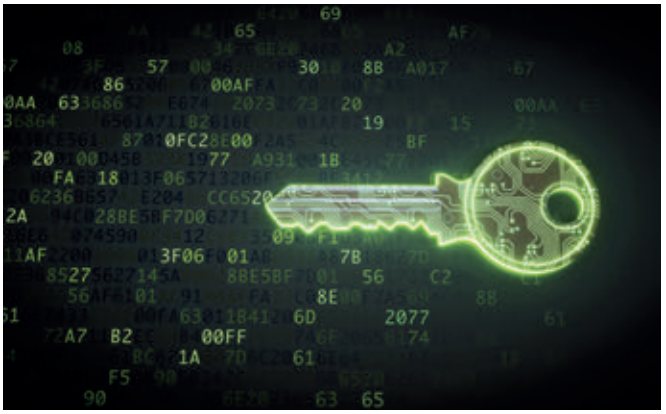
Gökkuşağı Tabloları

Bir sistemdeki kullanıcı bilgilerinin bir saldırganın eline geçtiğini düşünelim.

Saldırganın elinde kullanıcı hesaplarıyla ilgili üç şey vardır: kullanıcı isimleri, *belki* tuzlar ve parmak izleri.
Bu bilgiler tek başına sisteme giriş yapmak için yeterli değildir.

Peki, saldırganın edindiği bilgileri kullanarak kullanıcı şifrelerini bulması mümkün müdür?

Sistemde tuzlama yapılmadığını varsayalım.
Saldırganın takip edebileceği iki "kaba" yöntem vardır.



1. Yöntem: Saldırgan tüm muhtemel şifreler için tek tek parmak izlerini hesaplar. Çalınmış bilgilerdeki parmak izleriyle uyumlu şifreler bulunduğunda doğru şifreler de bulunmuş olur. Bu yöntem fazla bilgisayar hafızası istemez. Çünkü işlemler sırasında hiçbir şeyin kaydı tutulmaz. Ancak muhtemel tüm şifreler için parmak izlerini hesaplamak aşırı derecede uzun sürecektir. Örneğin kullanıcı şifreleri 12 karakter uzunluğundaysa ve 26 ayrı karakter kullanılıyorsa, saniyede bir milyar test yapabilen bir bilgisayarla muhtemel tüm şifrelerin parmak izini hesaplamak $26^{12}/(10^9 x 3600 x 24)=1104$ gün sürecektir. Eğer saldırganın elinde aynı güçte 1000 makine varsa gerekli süre yaklaşık bir güne düşer. Ancak bu yöntemin yine de uygulanabilir olduğu söylenemez. Çünkü 1000 makineyle bir günde yapılacak hesap sadece tek bir veri tabanındaki bilgiler içindir. Hiçbir şeyin kaydı tutulmadığı için, ileri bir tarihte başka bir veri tabanı ele geçirildiğinde tüm hesapları baştan yapmak gerekecektir.

2. Yöntem: Muhtemel tüm şifrelerin parmak izleri önceden hesaplanır ve kaydı tutulur. Bu durumda yapılması gereken tek şey ele geçen bilgilerle önceden hesaplanmış şifre-parmak izi çiftlerini karşılaştırmaktır. Ele geçirilen parmak izleri tablodaki hangi şifrenin parmak iziyle uyuşuyorsa doğru şifre odur. Bilgiler ele geçirildikten sonra bu yöntemle şifre kırmak için, fazla zaman gerekmez. Sadece kayıtlı verilerin taranması yeterlidir. Ancak bu yöntemde bilgileri saklamak için çok fazla hafıza gerekir. Tablolardaki her bir şifrenin 12 byte, her bir parmak izinin de (SHA256 Hash fonksiyonları kullanılıyorsa) 32 byte yer kaplayacağı düşünülürse, 26 karakterle oluşturulan 12 karakter uzunluğundaki şifreler için gerekli hafıza yaklaşık $26^{12} x (12+32)=4{,}2 x 10^{18}$ byte olacaktır. Başka bir deyişle, saldırganın 1 terabayt kapasiteli 4,2 milyon tane sabit diske ihtiyacı vardır. Kısacası bu yöntem de, birincisi gibi, pratikte uygulanabilir değildir.

Peki, şifre kırmaya çalışan bir saldırganın kullanabileceği, birinci yöntemden daha az işlem kapasitesi ve ikinci yöntemden daha az hafıza gerektiren bir yöntem var mıdır? Cevap maalesef evet. Saldırganlar, gökkuşağı tabloları olarak adlandırılan tabloları önceden hazırlayarak bir sabit diske kaydedebilir, daha sonra da bu tabloları kullanarak görece kısa bir süre içinde çalınmış verilerdeki parmak izlerinin hangi şifrelere ait olduğunu tespit edebilir.

Gökkuşağı tablolarının nasıl çalıştığını anlamaya çalışalım. Şifreleri *P* harfiyle, şifrelerden parmak izi üreten Hash fonksiyonlarını da *h* harfiyle gösterelim. Ayrıca parmak izlerinden yeni şifreler üreten bir fonksiyon olsun ve onu da *R* harfiyle gösterelim. Kısacası *h(P)* işlemi *P* şifresinden bir parmak izi, *R(h(P))* işlemiyse bu parmak izinden yeni bir şifre üretir. Hash fonksiyonlarının tersi olmadığı için, *R(h(P))* işlemiyle üretilen şifre, doğal olarak, başlangıçtaki *P* şifresinden farklıdır. Ancak yine de bir kullanıcı tarafından belirlenebilecek muhtemel şifrelerden biridir.

Gökkuşağı tabloları oluşturulurken önce bir P_0 şifresi alınır ve yeni bir şifre hesaplanır: $P_1=R(h(P_0))$. Daha sonra aynı işlem elde edilen yeni şifrelere tekrar tekrar uygulanır: $P_2=R(h(P_1))$, $P_3=R(h(P_2))$, $P_4=R(h(P_3))$, ... Hesaplama, parmak izinin bit gösterimi 20 tane 0 ile başlayan bir şifre bulunduğunda, *n* sayıda adım sonra sonlandırılır. Başlangıçtaki şifre P_0 ve *n*'inci adım sonunda elde edilen, 20 tane sıfır ile başlayan parmak izi, $h(P_n)$ gökkuşağı tablosuna eklenir. İşlemler sırasında elde dilen diğer şifrelerin ve parmak izlerininse kaydı tutulmaz. Aynı işlem muhtemel tüm şifreler için tek tek yapılır. Böylece muhtemel *bazı* şifrelerden ve 20 tane sıfırla başlayan parmak izlerinden oluşan bir tablo ortaya çıkar.

Ele geçirilmiş bir veri tabanındaki parmak izlerinin hangi şifrelere ait olduğunu bulmak için, önce bir parmak izine, f_0, *R* fonksiyonu uygulanarak bir şifre, $P_1=R(f_0)$, üretilir. Sonra bu şifreden yeni bir parmak izi üretilir: $f_1=h(R(f_0))$. Daha sonra aynı işlem defalarca tekrar edilir: $f_2=h(R(f_1))$, $f_3=h(R(f_2))$, $f_4=h(R(f_3))$ ... ta ki bir parmak izi f_n 20 tane sıfırla başlayıncaya kadar. Daha sonra bulunan parmak izinin gökkuşağı tablosundaki hangi şifreye karşılık geldiğine bakılır. Tablodaki şifre arzu edilen şifre değildir. Ancak bu şifreye *h* ve *R* fonksiyonları tekrar tekrar uygulandığında eninde sonunda veri tabanındaki parmak izine karşılık gelen şifre elde edilecektir.

Bu yöntem yukarıda bahsedilen yöntemlerin ikisinden de çok daha etkindir. Birinci yönteme göre çok daha kısa süre içinde sonuç alınır. Çünkü muhtemel tüm şifreler tek tek denenmez; hesaplara, gökkuşağı tablosundaki belirli bir şifreden başlanır ve eninde sonunda doğru şifrenin bulunacağı bilinir. Ayrıca bu yöntem ikinci yönteme göre de çok daha az depolama kapasitesi ister. Çünkü gök kuşağı tablosunda sadece 20 tane sıfır ile başlayan parmak izleri yer alır. Bir bitin alabileceği 2 değer (0 ve 1) olduğu için 20 tane 0 ile başlayan parmak izlerinin sayısı muhtemel tüm parmak izlerinin sayısının 2^{20}'de biri (yaklaşık milyonda biri) kadardır. İkinci yöntemde muhtemel tüm şifreleri ve parmak izlerini kaydetmek için 4,2 milyon tane 1 terabaytlık hard disk gerekiyordu. Dolayısıyla bir gökkuşağı tablosunu kaydetmek için toplam kapasitesi 4,2 terabaytın üzerinde olan birkaç hard disk yeterli olacaktır.

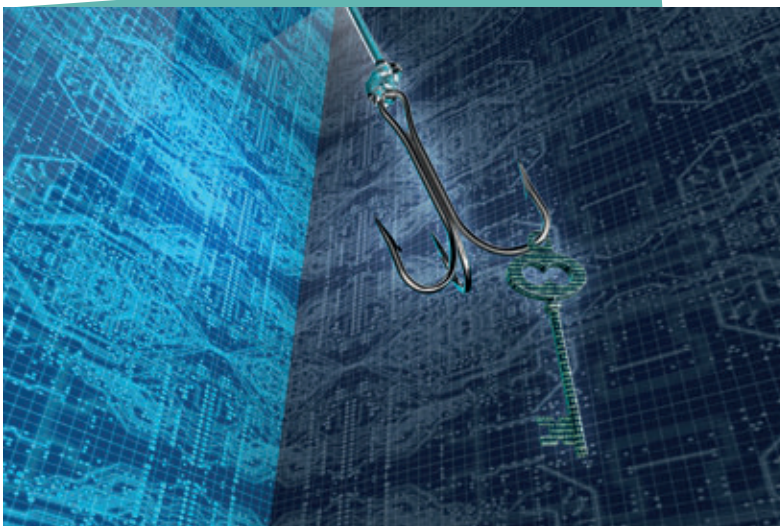
Çünkü sisteme sızmayı başaran bir saldırganın çıplak şifreleri ele geçirmesi durumunda istediği her şeyi yapabilmesinin önünde hiçbir engel kalmaz. Her ne kadar algoritmalar kullanarak parmak izlerinden şifreleri çözmek mümkün olsa da bunun için hem zaman hem güçlü bilgisayarlar hem de büyük depolama alanları gerekir. Bununla birlikte, saldırganlar parmak izlerini çözmek için uğraşırken sisteme sızıldığının farkına varılabilir ve gerekli önemler alınabilir.

Kullanıcıların yapması gereken en temel şeyse şifrelerini rastgele karakterler seçerek belirlemeleri. Akılda tutması kolay şifreler kullanmak saldırganların işini hayli kolaylaştırıyor. Özellikle de kullanılan şifreler daha önceleri hacklenmiş şifrelerden biriyse ve şifre sözlüklerinde yer alıyorsa. Kullandığınız şifrelerin sözlüklerde yer alıp almadığını https://haveibeenpwned.com/Passwords adresindeki tarayıcıyı kullanarak öğrenebilirsiniz. Eğer şifreleriniz daha önceleri hacklenmiş şifreler arasında yer alıyorsa değiştirmenizde fayda var. ■

Sonuç

Günümüzde pek çok web sayfası, kullanıcıların hesap açarken önerdikleri şifreleri test ediyor, zayıf bulmaları durumunda kullanıcıları daha güvenli şifreler belirlemeye zorluyor. Bu yöntem, her ne kadar kullanıcıların sinirlerini bozsa da güvenliğin artırılması açısından çok yararlı. Yaygın olarak kullanılan başka yöntemler de var. Örneğin, birkaç kez yanlış şifre girildikten sonra hesabın kilitlenmesi ya da her başarısız denemeden sonra bekleme süresinin iki katına çıkarılması gibi. Ancak bu yöntemler de, bilinmeyen bir güvenlik açığı sebebiyle, saldırganın varlığını belli etmeden sisteme erişmeyi başarması durumunda etkisiz kalıyor.

Kullanıcıların güvenliğini sağlamak için sistem tasarımcılarına düşen en temel görev, kullanıcıların şifreleri ile ilgili bilgileri çıplak hâlde sistemde tutmamak ve Hash fonksiyonları kullanarak şifrelerin parmak izini üretmek.



Kaynak

Delahaye, Jean-Paul, "The Mathematics of (Hacking) Passwords", *Scientific American*, https://www.scientificamerican.com/article/the-mathematics-of-hacking-passwords/?print=true, 12 Nisan 2019.