SEGURETAT I CONTRASENYES

"Tinc un password que no es pot crackejar", afirma el professor Anton Baka Baka, un respectat docent de l'Universitat de Tombridgetown. Es tracta d'una afirmació simple, encara que atrevida, ja que ens desafia a desmuntar aquesta sentencia.

Per a trobar alguna fissura en aquesta gran paret, començarem parlant de passwords. Al referir-nos a un password, estem donant a entendre que es tracta d'una forma d'autentificació, que utilitza informació secreta per controlar l'accés a alguna altre informació, privilegi o recurs. Així doncs, el principal requisit d'un password és ser secret i només conegut per el seu creador.

Ara que ja hem repassat la funció principal, podem aprofundir en les formes que pot tenir un password. És lògic pensar en una contrasenya com una successió (aleatòria o no) de números i/o lletres, ja que és al que estem més acostumats en el dia a dia, però hem de considerar moltes altres opcions no tant utilitzades, com successions d'imatges o contrasenyes de reconeixement de veu o d'empremtes dactilars.

Indiferentment de com sigui la contrasenya, per a parlar de crackejar, hem d'assumir que la contrasenya s'ha digitalitzat, i per tant, en el cas que debatrem, es tracta d'una seqüència d'uns i zeros.
Així doncs, amb els coneixements que disposem, podem afirmar que el password es pot desxifrar, ja que es tracta simplement d'anar provant combinacions de uns i zeros fins a obtenir el resultat. Tot i això, no sabem la longitud d'aquesta seqüència, i d'aquesta depèn el temps que tardaríem a obtenir el password. Tractant-se d'una combinació binaria de longitud N, sabem que el nombre màxim de combinacions disponibles és de 2^N, i per tant el temps d'espera mitja fins a obtenir el password dependria dels intents per segon que pot fer l'ordinador. Per a fer això, existeixen els algorismes de força bruta, que van provant combinacions fins a obtenir la contrasenya.

Com podem veure en la Il·lustració 1, una contrasenya formada per números és crackejada de forma instantània fins que supera la longitud de 8 caràcters, mentre que al ser de 18 caràcters el temps de crackeig puja fins a 126 anys, una quantitat de temps tant alta que no ens permetria viure per a accedir a l' informació que conté.

number of Characters	Numbers only	Upper or lower case letters	upper or lower case letters mixed	numbers, upper and lower case letters	numbers, upper and lower case letters, symbols
3	Instantly	Instantly	Instantly	Instantly	Instantly
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	3 secs	10 secs
6	Instantly	Instantly	8 secs	3 mins	13 mins
7	Instantly	Instantly	5 mins	3 hours	17 hours
8	Instantly	13 mins	3 hours	10 days	57 days
9	4 secs	6 hours	4 days	1 year	12 years
10	40 secs	6 days	169 days	106 years	928 years
11	6 mins	169 days	16 years	6k years	71k years
12	1 hour	12 years	600 years	108k years	5m years
13	11 hours	314 years	21k years	25m years	423m years
14	4 days	8k years	778k years	1bn years	5bn years
15	46 days	212k years	28m years	97bn years	2tn years
16	1 year	512m years	1bn years	6tn years	193tn years
17	12 years	143m years	36bn years	374tn years	14qd years
18	126 years	3bn years	1tn years	23qd years	1qt years

Il·lustració 1: Temps d'algorismes de força bruta

Per el que fa a seguretats mes avançades, al afegir lletres, majúscules i minúscules, el temps de crackeig per força bruta augmenta, sobretot a partir dels 10 caràcters, on passa a tractar-se de 16 anys amb un augment exponencial.

Finalment, en tractar-se d'una contrasenya amb números, lletres en majúscula i minúscula i símbols,

el temps de crackeig esdevé immens, fins al punt de necessitar 1 quintilló d'anys per a crackejar una contrasenya de 18 caràcters.

Per tant, si la contrasenya que ha pensat el professor Baka Baka és prou llarga i combina números, lletres i símbols, obtindrem un temps de crackeig molt elevat, i encara que en un futur es podria accedir al recurs que aquesta contrasenya protegeix, seria en un futur tant llunyà que segurament ja no existiríem.
Així doncs, per a obtenir la contrasenya que el professor Baka Baka considera no-crackejable, no podríem utilitzar la força bruta com una eina eficient i útil.

La següent opció a tenir en compte seria accedir al contenidor de la contrasenya i obtenir d'allà la propia contrasenya ja revelada, ja que aquesta ha d'estar en algun lloc, per a comparar-se amb l'entrada que es rep i poder decidir si permet o no l'accés a l' informació. Així doncs, podem suposar que aquesta contrasenya es troba guardada en algun lloc, i si som capaços d'entrar en aquest recipient i veure la contrasenya, podrem donar-la per crackejada.

Així doncs, aquí la nostre suposició es divideix en dos camins: per una banda, una contrasenya que es troba aliena al nostre ordinador; per l'altre una contrasenya que es troba guardada en el nostre ordinador.

Anem a tractar primer el cas aliè. Un servidor conté les contrasenyes dels usuaris que hi tenen accés, i el nostre objectiu és que ens reveli una contrasenya. Per a accedir al propi servidor i visualitzar les contrasenyes guardades, segurament necessitem entrar amb els privilegis de superusuari, i per tant hem de conèixer, o crackejar, la contrasenya del servidor.
Així doncs, per a obtenir una contrasenya d'una forma més ràpida que els atacs de força bruta, ens trobem que hem de realitzar un atac de força bruta igualment per a escalar privilegis i llavors obtenir la contrasenya que ens interessa de forma més fàcil.
Per tant, ens tornem a trobar en el cas d'un atac de força bruta, d'on el temps de desxiframent de la contrasenya dependrà de la seva longitud i complexitat.

Parlem ara del segon cas: un ordinador que conté la contrasenya en qüestió en algun racó de la seva memòria. Sembla més accessible, ja que es troba en el mateix dispositiu en el que estem treballant, però també ens trobem amb un gran problema: hem de superar la capa d'usuari per a tenir accés a tots els arxius del sistema, i això s'aconsegueix amb la contrasenya de root, així que de nou ens trobem amb una contrasenya que desconeixem i hauríem de crackejar d'alguna forma més complicada, com podria ser la força bruta.
Tot i això, suposem que aconseguim entrar al fitxer que conté les contrasenyes. Resulta que en el fitxer estan guardades amb un hash criptogràfic, que es calcula cada cop que l' usuari introdueix la contrasenya i es compara amb el valor guardat per a aquest usuari. Així doncs, no serviria de res introduir el hash com a contrasenya, ja que aquest es recalcularia com a hash durant la comprovació i ens donaria un valor diferent al hash autèntic.

Per a aquest motiu, existeixen les taules Rainbow, que contenen la traducció inversa, de hash a contrasenya, tot i que aquestes fallen al utilitzar la tecnologia Salts, que a dos contrasenyes

iguals(que tindrien el mateix hash) se les hi afegeix fragments aleatoris als hash per a que no siguin idèntics i evitar problemes de seguretat.

Ens adonem doncs, que ho mirem per on ho mirem, si el sistema operatiu implementa una bona seguretat i independència entre l' usuari i el sistema, ens és gairabé impossible accedir a un valor que està guardat en el mateix sistema, i per tant no tenim manera d'obtenir la contrasenya.

Per això, els sistemes de crackeig de contrasenyes més utilitzats actualment no es basen en la força bruta ni en escalar privilegis, ja que això comporta intentar enganyar a la màquina, i en el seu lloc intenten enganyar a l'ésser humà que hi ha al darrere, a l' usuari.

D'aquestes tècniques en podem distingir dues de més utilitzades, el phishing i el keylogger.

Keylogger:

Suposant que disposem de la direcció de correu del professor Baka Baka, podríem enviar-li un malware que s'activés i registrés totes les contrasenyes que escriu en el seu ordinador, i en el cas que accedís amb la contrasenya en qüestió, es guardaria i seria enviada, facilitant així l'obtenció de la contrasenya, per més extensa que fos.
Un cas molt simple i eficient, però no hem d'oblidar que la majoria dels usuaris d'internet utilitzen antivirus, i que segurament el nostre programa malware seria detectat i eliminat abans de posar-se en execució, per el que difícilment obtindríem alguna contrasenya.

Phishing:

Aquesta tècnica, molt més utilitzada actualment, permet obtenir les contrasenyes sense cap atac de força bruta ni malware, aprofitant-se només de l' ingenuïtat i la desconeixença dels usuaris.
Suposant, en aquest cas també, que disposem de la direcció de correu del professor Baka, podríem enviar-li un correu fent-nos passar per l'empresa o web en la que ell ha utilitzat la contrasenya que considera in-crackejable, i explicar-li que hi ha hagut algun accés no autoritzat a la seva compta.

Llavors, com que desitjaria saber que ha passat o qui ha intentat accedir, li facilitaríem un enllaç a una pàgina web pràcticament idèntica a la que està registrat, i li demanaríem les seves credencials. Si ell no en sospités res, accediria amb el seu usuari i contrasenya a la nostre web, i al fer-ho ens estaria donant la contrasenya sense ni adonar-se'n.
Aquesta tècnica doncs, ens permet obtenir la contrasenya directament de l'usuari, sense haver d'utilitzar cap traducció ni força bruta, i per això es tracta de la més efectiva i la mes denunciada actualment.

Com podem veure en les il·lustracions 2 i 3, la diferència és mínima, només és pot distingir per la direcció de la pàgina web, i tot i així a vegades són capaços de encobrir-la i presentar una pàgina calcada a l'original.

Il·lustració 2: Pàgina de Facebook original

Il·lustració 3: Pàgina de Facebook amb phishing

Així doncs, després d'estudiar els mecanismes possibles per a crackejar una contrasenya, podríem afirmar que la frase del professor Anton Baka manca de matisos i i no és del tot certa, tot i que és cert que pot tenir un password molt difícil de crackejar.
En primer lloc, com hem vist, amb algorismes de força bruta es pot desxifrar, encara que depenent de la llargada de la contrasenya podríem estar parlant de milions d'anys de temps mitjà de crackeig.
Per altre banda, hem vist que la seguretat de hash criptogràfic queda ofuscada per les taules Rainbow, així que podria ser que es pogués obtenir la contrasenya amb aquest mètode.

Finalment, hem de tenir en compte que més enllà de la seguretat que aporta l'ordinador i que com hem vist, no és 100% fiable, hi ha el factor humà, i per tant el professor Anton Baka hauria d'estar preparat per a tot tipus d'enganyifes com el phishing.

Per tant, no hi ha password que no es pugui crackejar, però si prou fort com per a mantenir el recurs que es vol protegir de forma segura durant una quantitat de temps força elevada.

BIBLIOGRAFIA:

«Complex Passwords Harder To Crack, But It May Not Matter» per Jason Sherril.
15 de Juny de 2012. Article de blog: https://www.inetsolution.com/blog/june-2012/complex-passwords-harder-to-crack,-but-it-may-not

«Criptoanálisis: Las tablas rainbow» per José Román.
3 d' Abril de 2008. Article del blog: https://www.emezeta.com/articulos/criptoanalisis-las-tablas-rainbow

«Keylogger» de Wikipedia.
Ultima edició del 1 de Novembre de 2017. Article de wikipedia:
https://es.wikipedia.org/wiki/Keylogger

«Phising:no muerdas el anzuelo» per INCIBE.
20 de Gener de 2017. Article de blog: https://www.incibe.es/protege-tu-empresa/blog/phishing-no-muerdas-el-anzuelo

«Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks» per W. Melicher, Blase Ur, S.M. Segreti, S. Komanduri, L. Bauer, N. Christin i Lorrie F. Cranor.
15 d' Agost de 2016. Article de blog: https://www.usenix.org/conference/atc17/technical-sessions/presentation/melicher