Cracking Passwords

José de Jesús Angel Angel

noviembre 2014

Definiciones

Passwords: es una cadena secreta de caracteres, que es usada para autenticar la identidad de una persona (generalmente). Es usado en un ambiente de medio riesgo en seguridad.

Definiciones

Los passwords son usados en:

- Acceso a una cuenta de un sistema, como una PC.
- Acceso a una cuenta de correo electrónico, como gmail.
- Acceso a un archivo de datos, como un archivo PDF o una carpeta de archivos.
- Acceso a un servicio, como la conexión de Internet.

Definiciones

Los passwords como parte de la autenticación fuerte:

- El password (llave): algo que yo se.
- Smart Card (Token): algo que yo tengo.
- Huella digital (biometría): algo que yo soy.

Definiciones

Diferentes tipos de passwords:

- PIN, Personal identification number, número de 4 digitos.
- Password, cadena de 8 a 10 caracteres .
- Passphrase, cadena de más de 10 caracteres.

Definiciones

Vida de un password:

- Un usuario (un username) genera su password.
- El password se almacena en una base de datos, asociado al username.
- El password se almacena, en claro, Hash-ado, Salt-Hash-ado.
- El password se renueva.

Definiciones

Cómo funciona un password:

- Al solicitar el servicio, se asocia el username con un password.
- Se aplica el algoritmo para verificar el password, Hash o Salt-Hash del password.
- Se accesa a la base de datos o se transmite el password requerido.
- Si los Hash son iguales, se verifica la identidad.

Definiciones

Almacenamiento de passwords:

- Un password se puede almacenar en el archivo de passwords.
- El archivo de passwords puede estar encriptado, por el SO o una aplicación.
- Es posible configurar el archivo de passwords.
- El archivo de passwords contiene solo los Hash del password.

Definiciones

Transmisión de passwords:

- Un password se puede transmitir usando algún protocolo como FTP, POP3, HTTP.
- La transmisión de información es vulnerable a los sniffers.
- Es posible que un sniffer capture los passwords.

Definiciones

Transmisión de passwords:

- Cifrar los passwords por una capa segura evita la captura por el sniffer.
- Protocolos seguros TLS, VPN.

Definiciones

Política sobre passwords:

- El password debe ser lo más aleatorio posible.
- Debe de cambiarse cada k meses.
- Alerta: casi todo lo que es recordado puede ser violado .
- Mantener el password confidencial .
- Cuando sea posible debe usar un administrados de passwords.

Definiciones

Entropía de un password:

- La entropía de un password mide la incertidumbre de los caracteres del password.
- Si un password tiene k bits de longitud, y es elegido de manera aleatoria, entonces hay 2^k posibilidades de poder elegirlo. Se dice que tiene k bits de entropía.
- Si un password tiene l caracteres que son elegidos de un alfabeto de b caracteres, entonces la entropía del password es b^l.
- Es difícil y relativo calcular la entropía de un password elegido por un usuario, debido a que los usuarios generalmente usan patrones.
- La pregunta es entonces: ¿Cómo los usuarios eligen passwords?.

Práctica I

Entropía de un password:

- Calcular la entropía de algunos passwords.
- Elegir diferentes alfabetos .
- http://rumkin.com/tools/password/passchk.php.
- Encontrar un password de mayor longitud con menor entropía.
- Encontrar un password de mayor entropía con menor longitud.

Bases de datos de Passwords

Como son los passwords:

- https://wiki.skullsecurity.org/Passwords.

Práctica II

Buscar un password en una base de datos:

- Bajar varias bases de datos .
- Buscar un password que haya usado o use.

Longitud de Passwords (rockyou)

Longitud:	5	6	7	8	9	10
Porcentaje:	1.81	13.58	17.47	20.68	15.27	14.04
Longitud:	11	12	13	14	15	16
Porcentaje:	6.04	3.87	2.54	1.73	1.12	0.83

Tipo de caracteres de Passwords (rockyou)

Loweralpha-numeric	43
Loweralpha	26
Numeric	17
loweralpha-special-numeric	2.5
upperalpha-numeric	2.55
mixedalpha-numeric	2.4
Loweralpha-special	2.4
upperalpha	2.0
mixedalpha	1.5
mixedalpha-special-numeric	0.7
mixedalpha-special	0.6
special-numeric	0.5
upperalpha-special-numeric	0.4
upperalpha-special	0.3
special	0.1

Passwords más usados (rockyou)

password	94911
love	82753
iloveyou	72330
princess	59569
angel	48852
ever	34350
monkey	33849
life	32436
babygirl	32241
Nicole	31998
soccer	31032
rockyou	29130
baby	28622
sexy	27999

Características (rockyou)

Contiene meses del año	459734 (3.2%)
Contiene días de la semana	136398 (0.95%)
Último caracter es 1	1270774 (8.86%)
Último caracter es un número	8316497 (57.98%)
Últimos dos caracteres son números	1539147 (10.73%)
Contienen años	440006 (3.06%)

Longitud de Passwords (Phpbb)

Longitud:	5	6	7	8	9	10
Porcentaje:	4.38	22.46	17.49	29.61	10.43	6.78
Longitud:	11	12	13	14	15	16
Porcentaje:	2.91	1.59	0.69	0.38	0.18	0.08

Tipo de caracteres de Passwords (Phpbb)

Loweralpha-numeric	36.22
Loweralpha	41.44
Numeric	11.24
loweralpha-special-numeric	0.89
upperalpha-numeric	1.22
mixedalpha-numeric	5.21
Loweralpha-special	0.81
upperalpha	0.94
mixedalpha	2.72
mixedalpha-special-numeric	0.89
mixedalpha-special	0.17
special-numeric	0.11
upperalpha-special-numeric	0.03
upperalpha-special	0.02
special	0.03

Características (Phpbb)

Contiene meses del año	3310 (1.8%)
Contiene días de la semana	1743 (0.93%)
Último caracter es 1	13737 (7.73%)
Último caracter es un número	78210 (41.8%)
Últimos dos caracteres son números	1740 (0.93%)
Contienen años	4587 (2.45%)

Longitud de Passwords (Linkedin)

Longitud:	5	6	7	8	9	10
Porcentaje:	0	10.6	11.19	31.45	17.51	13.12
Longitud:	11	12	13	14	15	16
Porcentaje:	6.62	4.19	2.15	1.26	0.63	0.26

Tipo de caracteres de Passwords (Linkedin)

Loweralpha-numeric	43.96
Loweralpha	20.19
Numeric	4.11
loweralpha-special-numeric	4.18
upperalpha-numeric	2.56
mixedalpha-numeric	15.64
Loweralpha-special	1.27
upperalpha	0.58
mixedalpha	3.66
mixedalpha-special-numeric	0.55
mixedalpha-special	1.16
special-numeric	0.24
upperalpha-special-numeric	0.55
upperalpha-special	0.08
special	0.04

Características (Phpbb)

Contiene meses del año	160045 (2.9%)
Contiene días de la semana	57317 (1.04%)
Último caracter es 1	579124 (10.58%)
Último caracter es un número	2833266 (51.35.8%)
Contienen años	138854 (2.52%)

Conclusiones

Con base a los anteriores datos:

- La longitud de los passwords (85%) esta entre 6 y 10.
- Los passwords no tiene mucha entropía.
- Están compuestos de caracteres muy comunes.
- ni siquiera usan todo los caracteres del teclado.
- No es difícil construir reglas que generan patrones en los passwords.

Reglas

Lowercase all letters	l	PAssWOrd	password
Uppercase all letters	u	PAssWOrd	PASSWORD
Toggle all charactersat word	t	PAssWOrd	paSSwoRD
Reverse the entire word	r	PAssWOrd	drOWssAP
Rotates the wordleft	{	PAssWOrd	AssWOrdP
Append character to end	$1	PAssWOrd	PAssWOrd1
Delete character at position 3	D3	PAssWOrd	PAsWOrd
Replace all instances X with Y	ssE	PAssWOrd	PAEEWOrd
Swaps first two characters	k	PAssWOrd	APssWOrd
Duplicates first N characters	y2	PAssWOrd	PAPAssWOrd

Reglas

Con base a los anteriores datos:

- Para crear passwords se crean patrones geométricos en el teclado.
- Por ejemplo tomar la diagonal ”qazwsx”.
- Para crear un diccionario inteligente deben de agregarse estos posibles passwords.

Riesgos de los passwords

Ataques:

- En la práctica el riesgo mayor de un password es la elección del usuario.
- En el peor de los casos, es posible obtener el 60% de los passwords, en el mejor de los casos es posible obtener el 90%.
- En la práctica si un password es elegido por el usuario su entropía es baja.

Riesgos de los passwords

Tipos de Ataques:

- Activos: estos ataques se llevan a cabo en el momento de la sesión .
- Pasivo: se lleva a cabo fuera de conexión, .

Riesgos de los passwords

Ataques:

- Ataque de fuerza bruta: este ataque recorre todo el espacio de posibles passwords, por ejemplo desde ”aaaaaaaa” hasta ”zzzzzzz”, en el caso del alfabeto de 26 letras minúsculas.
- Ataque de diccionario: este ataque recorre todo el diccionario, donde se encuentran las palabras posibles a ser usadas como passwords, dependen del idioma.
- Ataque de tablas rainbow: este ataque recorre combina los anteriores considerando que lo que se almacena son los hash de los passwords.
- Ataque de diccionarios inteligentes: este ataque construye diccionarios que consideran la maneras más probables que los usuarios construyen passwords, se usan cadenas de Markov.

Riesgos de los passwords

Ataques:

- Ataque de Ingeniería Social: en este caso el atacante hace uso de la inteligencía social para obtener parte total o parcial del pasword de un usuario ogrupos de usuarios.
- Ataque de un canal lateral (side channel attacks): este ataque tiene diferentes facetas y hace uso del sonido, del consumo de poder, del campo electromagnético etc.

Funciones Hash

¿Qué es una función Hash?

- Una función hash es una función de un solo sentido.
- Fácil de evaluar, pero difícil de evaluar su inversa.
- Una función hash asocia una cadena de caracteres de longitud arbitraria una cadena de bits fija.
- Debe ser computacionalmente imposible que dos cadenas tengan un hash igual, es decir, son libres de colisión.
- Un hash asocia de la manera más aleatoria su salida, con una pequeña variación de la cadena original, la salida debe ser completamente diferente.

Funciones Hash

Ejemplos de valores Hash:

- hash(”hello”) = 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
- hash(”hbllo”) = 58756879c05c68dfac9866712fad6a93f8146f337a69afe7dd238f3364946366
- hash(”waltz”) = c0e81794384491161f1777c232bc6bd9ec38f616560b120fda8e90f383853542

Funciones Hash

Standar Hash:

- Del 2007 al 2012 se realizó un concurso para definir un estándar(NIST).
- El ganador fue el algoritmo Keccak.
- El estándar es llamado SHA3.
- Existen las versiones: SHA3-224, SHA3-256, SHA3-384, SHA3-512 .
- Otras: MD4, MD5, SHA1, SHA2.

Práctica III

Encontrar el hash de un password:

- Recuerde un password que haya usado.
- Entonces calcule su hash .
- Buscar en google el hash.
- Qué conclusiones se obtiene si lo encontró o no.

Motivación

Motivaciones:

- En 1948 C.E. Shannon publica las bases de la teoría de las comunicaciones.
- (A Mathematical Theory of Communication,The Bell System Technical Journal, Vol. 27).
- Entre otras cosas aplica modelos de Márkov para estudiar las sucesiones de letras en textos escritos.
- Las cadenas de Márkov han sido usadas en el reconocimiento de voz, de escritura, la compresión de datos, y el filtrado de spam.
- En el algoritmo PageRank de Google, y genemark para estudiar el gen.

Motivación

Modelo de Márkov de orden 0.

- Predecir si ocurre una letra en un texto.
- Primero calcular el número de letras que aparecen en textos, digamos en español.
- Por lo tanto la probabilidad de aparición de la latra ”**a**”, es el cociente del número de letras entre el número de apariciones de **a**.
- En este caso se supone que la aparición de las letras es independiente.
- Aunque la parición de las letras no es independiente, es decir, hay más probabilidad de que aparezca una ”**n**” después de una ”**o**” que una ”**t**”.

Motivación

Modelo de Márkov de orden k.

- Por lo tanto un modelo más aproximdo a la realidad debe incluir que la aparición de una letra depende de otra.
- Un modelo de Markov de orden k, permite que la aparición de una letra tenga una probabilidad fija, pero que depende de las k letras anteriores.

Ejemplo cadena de Márkov de orden 1

Ejemplo:

- Podemos generar una cadena de Márkov iniciando con una letra tomada una un libro elegida aleatoriamente (la pregunta es saber cual es la letra siguiente con más probabilidad).
- Ahora abrimos el libro en otra página aleatoriamente y buscamos la letra, entonces anotamos la letra que le sigue.
- Hacemos lo mismo para la segunda letra, de esta manera estamos construyendo una cadena de Márkov de orden 1.
- Esto nos permite crear un diccionario con palabras (cadenas de caracteres) con mayor probabilidad de ser elegidas como password.

formalización del modelo de Markov

Definición de un diccionario de Markov.

- Modelo de Markov de orden 0.
- todos los caracteres aparecen de manera independiente.
- $P(\alpha) = \prod_{x \in \alpha} \nu(x)$.
- Un diccionario de Markov de orden 0 es $D_\theta = \{\alpha | \prod_{x \in \alpha} \nu(x) \geq \theta\}$

formalización del modelo de Markov

Definición de un diccionario de Markov.

- Modelo de Markov de orden 1.
- Se considera la proabilidad de la acurrencia de dos caracteres juntos.
- $P(x_1 x_2 .. x_n) = \nu(x_1) \prod_{i=1}^{n-1} \nu(x_{i+1}|x_i)$.
- Un diccionario de Markov de orden 1 es $D_\theta = \{x_i x_2 .. x_n | \nu(x_1) \prod_{i=1}^{n-1} \nu(x_{i+1}|x_i) \geq \theta\}$

Objetivo del diccionario de Markov

Calculos.

- El objeto de un diccionario (inteligente) de Markov es: reducir el espacio de posibles password que se pueden generar, por ejmplo con un alfabeto de 26 letras y passwords de 8.
- En el caso general serían
$$P(26,8) = \frac{26!}{(26-8)!} = 62,990,928,000.0.$$
- En el caso del modelo de orden 0.
- Un diccionario con $1/7 (= 14\%)$, del total, un password tendría el 90% de probabilidad de pertenecer a el.
- Es decir el espacio de palabras de logitud 8 sería de $8,818,729,920.0$.

Objetivo del diccionario de Markov

Calculos.

- Un diccionario con $1/11 (= 9\%)$, del total, un password tendría el 80% de probabilidad de pertenecer a el.
- Un diccionario con $1/40 (= 2.5\%)$, del total, un password tendría el 50% de probabilidad de pertenecer a el.

Objetivo del diccionario de Markov

Calculos modelo orden 1.

- Un diccionario con $1/10 (= 10\%)$, del total, un password tendría el 98% de probabilidad de pertenecer a el.
- Un diccionario con $1/20 (= 5\%)$, del total, un password tendría el 95% de probabilidad de pertenecer a el.
- Un diccionario con $1/60 (= 1.6\%)$, del total, un password tendría el 90% de probabilidad de pertenecer a el.
- Un diccionario con $1/250 (= 0.4\%)$, del total, un password tendría el 80% de probabilidad de pertenecer a el.
- A. Narayanan V. Shmatikiv, Fast Dictionary attacks on Passwords using time-space Tradeoff.

Creación de un cadena de Markov a partir de un password, de la base de datos Rockyou.

Ejemplo de 2-gram

password	abc123	monkey	123455	assurance
pa	ab	mo	12	as
as	bc	on	23	ss
ss	c1	nk	34	su
sw	12	ke	45	ur
wo	23	ey	55	ra
or				an
rd				nc
				ce

Creación de un cadena de Markov a partir de un password, de la base de datos Rockyou.

Ejemplo de 3-gram

password	abc123	monkey	123455	assurance
pas	abc	mon	123	ass
ass	bc1	onk	234	ssu
ssw	c12	nke	345	sur
swo	123	key	455	ura
wor				ran
ord				anc
				nce

Ocurrencias de 2-grams en la base de datos Rockyou

an	1216714
ar	834574
er	795998
ma	745282
in	741061
12	735937
19	650542
el	588617
on	581316
al	575873
00	570997

Ocurrencias de 3-grams en la base de datos Rockyou y su posición

mar	88621	0
lov	62729	1
ove	62930	2
ilo	54507	0
ari	50539	1
sha	48340	0
085	45572	0
cha	43118	0
and	44381	1

Rainbow Tables

Como se guardan los passwords:

- Los passwords, en la actualidad se guardan en hash-ados
- Es decir, no se guarda el password en claro, sino que se guarda su valor hash.
- Esto dificulta el ataque de fuerza bruta y diccionario para encontrar en password.
- Sin ambargo existen técnicas en la actualidad para optimizar la búsqueda aún hash-eado el password.
- Una de estas técnicas es llamada ”Tablas Rainbow”.

Rainbow Tables

Como funcionan:

- P. Oechslin, Making a Faster Cryptanalytic Time-Memory trade off.
- Se tiene el Hash de un Password (que llamaremos texto claro).
- Queremos encontrar el texto claro, que llamaremos T_n.
- La primera idea fue, crear tablas de posibles password con su hash.
- Entonces si nuestro Hash esta en la tabla, pues sabemos cual T_n

Rainbow Tables

Como funcionan:

- Sin embargo esto causa mucha perdida de tiempo y memoria.
- Por lo que se craron las rainbow tables.
- Para la creación de una Rainbow Table considere al conjunto P de posibles passwords, que puede ser un diccionario.
- Llamanos al conjunto H el conjunto de los Hashes.
- Para generar la primera fila de la tabla Rainbow se procede de la siguiente manera.

Rainbow Tables

Como funcionan:

- Se parte de un $a_{11} \in P$, y se le aplica la función Hash, obteniendo $a_{12} = Hash(a_{11})$.
- La parte mágica de las tablas rainbow se tiene al considerar una función g, que se le llama función reducción (en la literatura).
- $g : H \rightarrow P$, es decir comvierte un valor hash a un texto (password).
- Como funciones reducción pueden tomarse el truncamiento, o códigos que convierten bits a textos.
- Así: $a_{13} = g(a_{12})$, $a_{14} = Hash(a_{13})$, $a_{15} = g(a_{14})$.

Rainbow Tables

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	...	$a_{1(2k)}$
texto	hash	texto	hash	texto	...	hash

- Tenemos entonces, varias cadenas como parte de las filas de la rainbow table.
- Por otra parte tenemos un Hash del que no sabemos su texto.
- (1) Chequeamos primero si el Hash coicide con algún Hash final de todas las filas, es decir, si hash$= a_{i(2k)}$ para algún i.

Rainbow Tables

Como funcionan:

- (2) Si encontramos un i termina la búsqueda.
- (3) Si no es así, le aplicamos una función reducción g, y le sacamos su hash, vamos al paso (1).
- (4) Si el hash coincide con algún hash final, entonces el original está en esa cadena.
- (5) Ahora, de la cadena encontrada iniciamos desde el texto inicial y recorremos la cadena, hasta encontrar el hash inicial. Por lo tanto hemos hallado su texto .

Rainbow Tables

- Se llaman Rainbow, debido a que pueden usarse diferentes funciones reducción g.
- Las funciones reducción deben de diseñarse de acuerdo al espacio de password a atacar.

Práctica IV

Encontrar el hash en una Rainbow table:

- http://project-rainbowcrack.com/.

Hash con Salt

- Para evitar este anterior ataque, las bases de datos que guardan passwords, pueden agregar el aditamento llamado Salt.
- Al password (64 bits), se le agregan tantos bytes aleatorios del tamaño del hash (128 bits) y se obtiene el hash = hash(password—Salt), se guarda enseguida del hash el salt.

- http://hashcat.net/oclhashcat/
- https://www.thc.org/thc-hydra/
- https://www.freerainbowtables.com/

Historia

- En 1965, Peter Wright publica su libro Spycatcher.
- En 1956, la inteligencia británica logra colocar micrófonos cerca de la máquina de cifrado Hagelin, en la embajada de Egipto en Londres.
- Se obtiene la clave de cifrado con las grabaciones.
- Puede ser el primer Side Channel Attack conocido de la historia.

Side Channel Attacks

- Usando un ataque del tipo SCA, es posible recuperar todos los caracteres de un password.
- Los experimentos hechos dicen que entre un 80 % y un 90 % de texto aleatorio se puede recuperar.
- Si se combina este ataque a uno de fuerza bruta o similar, es posible recuperar todos los passwords.
- L. Zhuang, F. Zhou, J. Tygar, Keyboard Acoustic Emanation Revisted,In Proceedings of the 12th ACM Conference on Computer and Communications Security.

Bibliografía

- C. Yiannis, modern Password Cracking: A hands-on approach to creating an optimised and versatile attack.
- M. Zviran W. J. Haga, Password Security: An Empirical Study.
- R. Morris, K. Thompson, Password Security: a case history.
- M. Bakker, R. van der Jagt, GPU based password cracking.
- D. Schweitzer J. Boleng, C. hudges, L. Murphy, Visualizing Keyboard Pattern.
- W. Ma, J. Campbell, D. Tran, D. Kleeman, Password Entropy and Password Quality.

Contacto
José de Jesús Ángel Ángel
jjaa@math.com.mx
www.math.com.mx