Nuevas Metodologías de Descubrimiento de Contraseñas

ESPECIAL XII - 2017

www.pabloyglesias.com
contacto@pabloyglesias.com

Las contraseñas siguen siendo el sistema de autenticación por defecto de la amplia mayoría de servicios, y de buena parte de los dispositivos tecnológicos que tenemos a nuestro alrededor.

Estado del arte actual

El panorama de nuestros días es conocido por todos nosotros.

Las contraseñas siguen siendo el sistema de autenticación por defecto de la amplia mayoría de servicios, y de buena parte de los dispositivos tecnológicos que tenemos a nuestro alrededor.

Solamente la irrupción de sistemas de autenticación basados en la biometría *(huellas dactilar, iris de los ojos…)*, y la lenta expansión de los sistemas de doble factor, han democratizado un escenario prácticamente dominado por estos **sistemas de identificación basados en el conocimiento.**

Y esto, como seguramente sepa, entraña considerables riesgos.

Una contraseña no es más que un conjunto de caracteres definidos por el usuario *(o por un intermediario)* que debe coincidir con un margen de error mínimo con el patrón definido en su momento. Y digo mínimo porque aunque en teoría la mayoría de sistemas de validación únicamente aceptan la exactitud con el patrón de desbloqueo, en la práctica hay acercamientos que sin aumentar apenas el riesgo de cara a ataques de terceros, sí mejoran considerablemente la usabilidad del servicio. Son, por ejemplo, las estrategias basadas en ofrecer sistemas de autenticación que aceptan mayúsculas o minúsculas, errores puntuales en un único carácter, intercambio de caracteres y una serie de modificaciones sutiles del patrón original que coinciden habitualmente con los clásicos errores que cometemos en teclados físicos, táctiles y móviles.

En todo caso, presuponiendo que la contraseña sea lo suficientemente robusta y que el sistema de validación no contenga errores de diseño o de implementación, es de esperar que un tercero que quiera acceder a los bienes que hay tras esta validación no vaya a poder puesto que desconoce la serie de caracteres que debería incluir. Es decir, que la seguridad propia del sistema, dejando de lado la seguridad del servicio, tecnologías y dispositivos que intervienen en el proceso de autenticación, **depende exclusivamente del conocimiento del usuario**. Algo externo al mundo digital, y por tanto, delegable en el buen quehacer de la persona que está a los mandos. Y algo cuya principal ventaja es que es modificable las veces que el usuario necesite *(podemos cambiar de contraseña cada vez que queramos;*

sin embargo, solo tenemos 5 dedos en cada mano, ergo 10 posibles huellas dactilar, o dos ojos, ergo dos posibles patrones de iris).

¿Cuál es el problema entonces?

Que **el usuario es el eslabón más débil de la cadena.**

La teoría dice que deberíamos tener una contraseña distinta para cada servicio, pero en la práctica, y filtración tras filtración, nos damos cuenta de que o bien por nuestra culpa *(falta de una comunicación adecuada)*, o bien por el propio sistema *(pretender delegar la seguridad del mismo en el eslabón más débil)*, esto falla.

Como cada año desde el 2011 TeamsID libera su estudio titulado ***"Worst** Passwords **List"** (1)*, y edición tras edición ***"123456"*** y ***"password"***son las contraseñas más utilizadas por el universo de usuarios estudiado.

Para colmo, no existe una estandarización aceptada por la industria como la más adecuada para implementar sistemas de autenticación basados en el conocimiento. Lo que en la práctica se traduce en que cada desarrollador diseña esta validación según su propio criterio, normalmente obviando factores tan decisivos en la seguridad del sistema como puede ser un control anti fuerza bruta o el cifrado, la dependencia de terceros, la comunicación y el tratamiento de contraseñas bajo protocolos seguros.

Conocedores de este escenario, los cibercriminales llevan tiempo puliendo las **técnicas de bypaseo de sistemas de autenticación.**

Hay múltiples, y en este estudio quería centrarme en aquellas referentes a la propia seguridad de las contraseñas *(técnicas de descubrimiento de contraseñas)*, dejando para otra ocasión aquellas cuyo fin es atacar a la tecnología que está detrás o encontrar la manera de eludir este tipo de controles.

Empecemos:

Ingeniería social

Es, de facto, el primer punto donde deberíamos pararnos.

La ingeniería social es una de las herramientas más útiles para descubrir la contraseña del usuario, principalmente porque ataca directamente al mismo, y dependiendo de cómo se implemente, es ajena a la tecnología *(es decir, ajena a los controles de seguridad que esta pudiera haber implementado)*.

Si hablamos de ingeniería social a la mayoría nos vendrá a la mente un ataque dirigido en el que una persona, haciéndose pasar por alguien de confianza de la víctima, consigue mediante el engaño que ésta, argumentando el motivo que sea, le entregue el acceso *(usuario y contraseña)* al servicio objetivo.

Así fue como APT-28, uno de los grupos de ciberdelincuencia-inteligencia *(a veces es difícil saber dónde empiezan unos y dónde acaban otros…)* que parece tener lazos directos con el gobierno de Putin, consiguió exponer toda la corrupción del partido demócrata de Hillary Clinton a varios días vista de las elecciones presidenciales del 2016 *(2)*. Y que, como bien sabe, se saldó con la victoria del republicano Donald Trump.

Al igual que ocurriera meses después con las elecciones francesas, cuyo objetivo era el partido de Macron, y con un resultado totalmente distinto.

Este tipo de ataques son muy costosos y altamente efectivos. Pero sin llegar a este punto, lo cierto es que todavía siguen funcionando muy bien los ataques de ingeniería social masivos, habitualmente apoyados en **campañas de phishing vía email o redes sociales**.

En este tipo de ataques la idea es que la víctima, que es totalmente desconocida para el atacante *(solo se conoce de ella el email o el usuario en redes sociales)* recibirá una comunicación en la que, argumentando la razón que sea, se le insta a loguearse en su cuenta. Desde acceder a un documento de Google Docs, pasando por una alerta de un servicio del cual somos usuario/cliente, o un aviso de una entidad de prestigio *(un banco, Hacienda, un operador de telecomunicaciones, nuestro proveedor de luz o agua…)*.

La página a la que nos lleva, por supuesto, no es la legítima, sino una copia creada ex-profeso para la ocasión, y tan pronto metamos nuestros credenciales, el mal ya estará hecho. Los

Las mecánicas de fuerza bruta se basan en ir probando una a una todas las posibles variaciones de arrays de caracteres junto con el usuario a quien queremos atacar para, con suerte, acabar dando con el par usuario/contraseña adecuado.

malos tendrán el usuario y contraseña en texto plano, y lo mismo nosotros ni nos enteramos, habida cuenta de que puede que una vez lo metamos nos envíe a la página oficial.

Gracias a este tipo de técnicas el atacante puede comprometer la seguridad de nuestra cuenta sin comprometer el sistema. La seguridad de la contraseña, que como ya hemos explicado depende de la confidencialidad del patrón de desbloqueo, se ve rota por parte del usuario. No por la parte tecnológica de toda la operación.

Fuerza bruta

La otra pata de la mesa está compuesta por todas las técnicas utilizadas para comprometer la seguridad de las contraseñas partiendo de su desconocimiento.

Y para ello históricamente se han aplicado **mecánicas de fuerza bruta**. Es decir, ir probando una a una todas las posibles variaciones de arrays de caracteres junto con el usuario a quien queremos atacar para, con suerte, acabar dando con el par usuario/contraseña adecuado.

Afortunadamente cada vez son más los servicios que implementan sistemas anti-fuerza bruta y que almacenan de forma cifrada nuestras contraseñas en su base de datos. Pero esto no significa que el riesgo esté solventado.

Es más, el quid de la cuestión está a día de hoy en cómo trabajar con hashes de contraseña cifrados *(lo que habitualmente se encuentra en bases de datos filtradas)* para obtener su contraseña equivalente.

Puesto que los algoritmos de cifrado son conocidos *(podemos decir que es la parte constante del hash)*, existen herramientas como *Hashcat (3)* que ayudan a los investigadores *(y a la industria del cibercrimen)* a descifrar estas contraseñas probando hashes creados con diferentes algoritmos.

Empezando por lo más básico, como sería ir testando una a una todas las combinaciones posibles partiendo de una ordenación puramente alfabética, a lo sumo comprendida entre los límites que acepta el sistema.

Por ejemplo:

Contraseña formada por entre 4 y 6 caracteres alfabéticos.

Posibles combinaciones: aaaa, aaab, aaac,..., aaba, aabb,..., xzzz, yzzz, zzz,...

Hashcat se encarga de para cada una de estas combinaciones ir probando diferentes algoritmos hasta que el hash resultante sea semejante a alguno de la lista

Como esto puede llevar horas, días y hasta años para probar todas las posibles opciones, se ha optado por utilizar bibliotecas que parten de esos estudios como el de TeamsID que mencionábamos al principio del whitepaper para intentar probar suerte con las contraseñas más habituales, o también existen herramientas que permiten generar bibliotecas específicas en base al conocimiento que quizás tengamos de la víctima, como es el caso de CeWL *(4)*, capaz de crear bibliotecas a partir de los datos obtenidos de perfiles en redes sociales o páginas web *(OSINT)*.

Un acercamiento más sensato pasaría por testar todas las combinaciones posibles con nombres comunes más fechas de nacimiento, o nombre del trabajador más nombre de la empresa, o inicial del nombre más apellido…

Según un estudio de la compañía *Praetorian Security (5)*, de 34 millones de contraseñas analizadas obtuvieron 13 estructuras diferentes de contraseñas que pueden perfectamente servir como base de conocimiento previo para realizar ataques de fuerza bruta más efectivos.

Parece que aunque sea inconscientemente la mayoría tendemos a crear contraseñas que empiezan con una letra en mayúscula seguida de varias en minúscula, con algún dígito o carácter no alfanumérico final. En total, entre cinco y ocho caracteres, con entre dos y cuatro dígitos. También se tiende a intercambiar vocales por dígitos *(la "a" por "4", la "e" por "3"…)*, o también por símbolos *(la "a" por "@", la "e" por "€",…)*. Y además, los caracteres no alfanuméricos más utilizados suelen ser el símbolo de punto ***"."*** y el del cierre de la exclamación ***"!"***.

Como crear este tipo de bibliotecas es muy pesado, algunas de estas herramientas han optado por ofrecer un conjunto de reglas, de tal manera que según las peticiones que hagamos la herramienta es capaz de crear para una contraseña *"password"* diferentes alternativas tales como *"Password"*, *"Password01"*, *"Pasword1!"*, *"P4ssw0rd"*…, cifrarla con diferentes algoritmos, y probar el hash obtenido en cada caso con un listado de pares usuario/contraseña, abarcando así un buen número extra de conjuntos. Tan pronto haya una

coincidencia, sabremos qué algoritmo se ha utilizado, simplificando el proceso para el resto de cuentas.

Otra opción, por cierto, es aplicar fuerza bruta no a la contraseña, sino al usuario, de tal manera que probaremos por ejemplo para la contraseña *"123456"*, que ya hemos visto que era de las más utilizadas, un listado de cuentas de correo o usuarios que previamente hayamos obtenido, quizás de una filtración anterior.

Con la ventaja, por cierto, que los sistemas de control de fuerza bruta, si es que han sido implementados, no suelen cubrir este escenario *(suelen estar enfocados a intentos con la contraseña, no con diferentes usuarios y misma contraseña)*.

El último campo de estudio en referencia a las metodologías de descubrimiento de contraseñas pasa por intentar ofrecer un ecosistema de algoritmos de cifrado inteligentes que compliquen todavía más el trabajo a los cibercriminales.

Un reciente estudio llevado a cabo por *Google Brain* explora el uso que podrían tener las redes neuronales en el ya clásico problema de comunicación entre Alice, Bob y Eve *(Alice y Bob tienen que encontrar la manera de decirse algo sin que Eve sea capaz de entenderles)*.

La idea pasa porque el sistema sea capaz de **generar un algoritmo de cifrado distinto para cada caso**, haciendo por tanto que el hash esté formado por la contraseña *(una variable dependiente de cada usuario)* y el algoritmo *(en vez de ser constante, también sería variable y único para cada usuario)*.

> *Paradigma clásico de cifrado:*
>
> *HASH = ALGORITMO DE CIFRADO (constante) + CONTRASEÑA (variable)*
>
> *Paradigma inteligente de cifrado:*
>
> *HASH = ALGORITMO DE CIFRADO (variable) + CONTRASEÑA (variable)*

Bajo el nombre de *Adversarial Neural Cryptography (6)* quizás estemos ante el futuro de unos sistemas de autenticación basados en el conocimiento que ofrezcan mayores garantías.

¿Qué podemos hacer como usuarios?

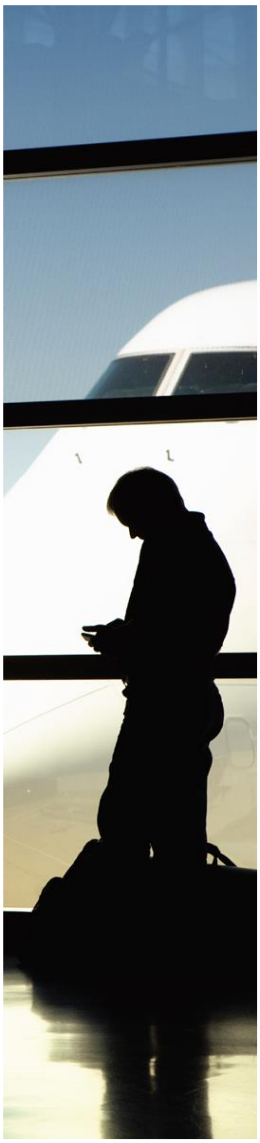
En base a todo lo expuesto, creo que es oportuno señalar algunos tips sencillos de aplicar que mejoran en su conjunto considerablemente la seguridad de las cuentas protegidas bajo este tipo de sistemas de autenticación. A saber:

- **El doble factor de autenticación:** Es lo más rápido y útil si lo que buscamos es mayor seguridad. De esta manera el servicio o producto estará protegido no solo por una contraseña, sino presumiblemente por un sistema basado en la posesión, que además utilizará un canal distinto para identificarse *(un token enviado a una aplicación, un SMS…)*.
- **Utilizar contraseñas menos habituales:** Ya conocemos los patrones habituales de construcción de contraseñas. Lo suyo entonces es evitar utilizarlos, poniendo mayúsculas por en medio de la contraseña y no al principio de la misma, intercalando números y letras arbitrariamente, no solo cuando hay vocales, utilizando otros caracteres no alfanuméricos distintos al punto y la exclamación…
- **La seguridad básica sigue siendo crítica:** El utilizar diferentes contraseñas para cada servicio, el ser consciente y conocer las tipologías de phishing más habituales y el evitar guardar contraseñas de forma insegura *(recuerde que hablamos de un paradigma de seguridad basado en la confidencialidad)* siguen y previsiblemente seguirán siendo elementos fundamentales de la seguridad de nuestras contraseñas en años venideros.

Nos dirigimos hacia un campo de batalla que en los próximos años se decidirá entre guerras de algoritmos: Unos dedicados a buscar sistemas cada vez más complejos de descifrar, y otros empeñados en explotar las limitaciones de estos sistemas para bypasear su cifrado.

Pero el problema de los ataques de ingeniería social va a seguir siendo un camino alternativo cada vez más cómodo e igual de efectista. Ese arma capaz de aprovecharse del sistema atacando al eslabón más débil, donde estamos todos nosotros.

De ahí la importancia de ser conscientes del funcionamiento de estas mecánicas. Porque si bien a cada paso estamos generando un ecosistema tecnológico más seguro, seguimos siendo débiles en todo aquello que depende única y exclusivamente de la parte humanista del sistema.



Puede acceder a los últimos informes en la sección archivo de la página

Referencias

1. "123456" and "password" again top SplashData's annual "Worst Passwords List" (TeamsID).
2. Russian Hackers Targeted Hillary Clinton Campaign Google Accounts (Forbes).
3. Hashcat: Advanced password recovery (Hashcat).
4. CeWL: Custom Word List Generator (DigiNinja).
5. Statistics will crack your password (Praetorian).
6. Learning to protect communications with adversarial neural cryptography (Martín Abadi and David G. Andersen, Google Brain, PDF).

Para realizar comentarios sobre este estudio, por favor, diríjase a la página habilitada para tal fin.

Información de contacto

Pablo F. Iglesias
Analista de Información
@PYDotCom
contacto@pabloyglesias.com