GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, TURISMO Y COMERCIO

plan avanza2

INTECO
Instituto Nacional de Tecnologías de la Comunicación

SEGURIDAD EN CLIENTES TWITTER

INTECO-CERT

Noviembre 2009

El presente documento cumple con las condiciones de accesibilidad del formato PDF (Portable Document Format).

Se trata de un documento estructurado y etiquetado, provisto de alternativas a todo elemento no textual, marcado de idioma y orden de lectura adecuado.

Para ampliar información sobre la construcción de documentos PDF accesibles puede consultar la guía disponible en la sección Accesibilidad > Formación > Manuales y Guías de la página http://www.inteco.es.

ÍNDICE

1. INTRODUCCIÓN 4
1.1. Motivación y objetivos 4
1.2. Seguridad en clientes Twitter 5
1.2.1. Autenticación 5
1.2.2. Cifrado 7
1.2.3. Almacenamiento de contraseñas 9
1.3. Metodología 9
2. PRUEBAS DE CLIENTES 12
3. ESTADÍSTICAS 15
4. CONCLUSIÓN Y RECOMENDACIONES 19

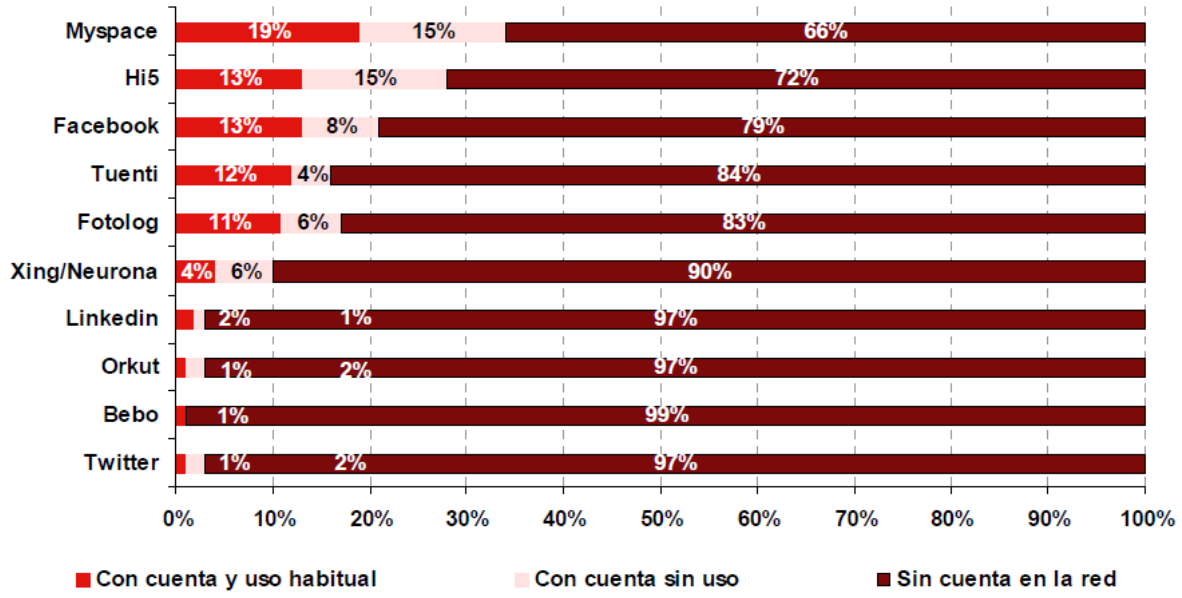
1. INTRODUCCIÓN

1.1. MOTIVACIÓN Y OBJETIVOS

Durante los últimos años, uno de los pilares de lo que se ha venido a denominar *Web2.0* ha sido la generación de contenido por parte de los usuarios. Inicialmente este contenido ha estado presente en forma de blogs, comentarios e información en redes sociales.

Podemos considerar que el *microblogging* es la confluencia de dos de estas formas de difusión de información. Por una parte es una simplificación de un blog, reduciendo el tamaño de sus entradas a la unidad mínima de información: un pensamiento, una imagen, un enlace, una pregunta, etc. Pero también puede verse como la evolución e independización de los mensajes de estado en las redes sociales y de mensajería instantánea.

Gráfico 1: Penetración de las diferentes Redes Sociales en España (Julio 2008)



Fuente: INTECO a partir de Observatorio sobre la Evolución de las redes sociales (The cocktail analysis)

Aunque existen múltiples servicios de *microblogging* (Tumblr, Plurk, Emote.in, Squeelr, Jaiku, identi.ca, etc.), el más popular entre este tipo de servicios es Twitter[1], que ofreciendo un servicio muy simple centrado en la accesibilidad y facilidad por parte de los usuarios para ver y generar contenido, permite una ubicuidad y "frescura" de información sin precedentes.

[1] *Estudio sobre la privacidad de los datos personales y la seguridad de la información en las redes sociales online* de INTECO [1] del que se ha extraído el Gráfico 1

Además del interfaz web, desde un principio la propia compañía ofreció acceso mediante SMS para enviar y recibir actualizaciones, con lo que el servicio se hizo accesible a gente que no tiene un teléfono inteligente ni está delante de un ordenador a menudo. Además, este énfasis en la movilidad ha provocado que las actualizaciones de los usuarios se produzcan de forma continua, desde cualquier lugar y en cualquier momento.

Casi desde un principio Twitter puso a disposición de terceros desarrolladores una interfaz de acceso a sus servicios (API, *application programming interface*), favoreciendo la creación de clientes para cualquier plataforma, además de servicios paralelos de búsqueda, seguimiento, estadísticas, etc.

La facilidad para acceder a la API de Twitter junto con la simplicidad funcional de un cliente de este servicio ha posibilitado que existan una infinidad de aplicaciones no oficiales para acceder al mismo. Según las estadísticas de uso en [4], un 78% de usuarios accede a la plataforma a través de aplicaciones diferentes de la interfaz web oficial.

El objetivo de este estudio es hacer una revisión de la seguridad implementada en una gran variedad de esos clientes, para los que se comprobarán los aspectos de autenticación, y cifrado que tienen lugar durante la comunicación entre cliente y servidor, además del almacenamiento de contraseñas para comprobar si el rápido crecimiento de la plataforma ha provocado que la seguridad se haya relegado a un segundo plano, suponiendo un riesgo de suplantación de identidad para los usuarios de estos clientes.

1.2. SEGURIDAD EN CLIENTES TWITTER

Normalmente las consideraciones de seguridad de las redes de *microblogging* se centran en la propia información difundida a través de las mismas, pero este estudio se ha enfocado a un aspecto más tecnológico, la comunicación entre los clientes utilizados por los usuarios y el servicio central de difusión.

Actualmente Twitter proporciona una serie de recomendaciones de seguridad para los usuarios de su API [2], pero esto no fue así desde el principio y como consecuencia muchos de los clientes presentan ciertas debilidades en este aspecto.

En los siguientes apartados se describen los diferentes aspectos relativos a la seguridad de los clientes, las posibles amenazas y las recomendaciones de Twitter.

1.2.1. Autenticación

Como en cualquier aplicación en la que existen perfiles de usuario, es necesario identificarse con un nombre de usuario y certificar que somos los dueños de esa cuenta. Este mecanismo de seguridad se aplica tanto a la interfaz web oficial como a la API que Twitter pone a disposición de los desarrolladores de aplicaciones.

En primer lugar vamos a revisar el proceso de autenticación en la web oficial, en la que podemos encontrar un formulario para introducir nombre de cuenta y contraseña. Tras enviarlos y una vez comprobado en el servidor que son correctos, éste devuelve un testigo,

una cadena aleatoria que identifica unívocamente a nuestra sesión, la famosa *cookie*. El navegador la almacena y la envía en peticiones subsiguientes para no tener que enviar las credenciales en cada una de ellas.

Es importante comprender que tras la comprobación inicial de las credenciales, lo único que identifica una sesión autenticada con el servidor (a priori) es la cookie. Por lo tanto, en caso de que alguien obtenga esa cadena y realice peticiones que la incluyan, el servidor interpretará que vienen del usuario legítimo. Esto es lo que se denomina secuestro de sesión, y permite suplantar la identidad de un usuario durante el tiempo que la *cookie* sea válida y por tanto la sesión se considere activa (en el caso de las aplicaciones web, pueden ser días, semanas, etc.).

Existen mecanismos para evitar este tipo de ataques comprobando parámetros adicionales de las peticiones (IP de origen por ejemplo), pero no entran en el objeto de este estudio.

En el caso de la API, a pesar de que la comunicación se realice mediante http, no se utilizan formularios y *cookies* como mecanismo de autenticación. Twitter ofrece dos formas de autenticación diferentes, detalladas en los apartados siguientes.

1.2.1.1. Autenticación básica

Este es el método de autenticación elegido para la API en sus inicios. Las credenciales del usuario se envían en claro en las cabeceras de la petición http. La ausencia de cualquier tipo de cifrado permite que cualquiera que pueda ver la petición sea capaz de descubrir la contraseña utilizada.

1.2.1.2. OAuth

OAuth es un protocolo abierto que permite la autorización de acceso seguro a una API de forma simple y estándar para aplicaciones web y de escritorio.

El proceso se puede resumir en los siguientes pasos.

El usuario pide a una aplicación cliente de escritorio o web que acceda a su cuenta.

La aplicación solicita a Twitter un token aleatorio que se asociará al par usuario+aplicación.

La aplicación redirige al usuario a la web oficial con el token aleatorio como parámetro.

El usuario se autentica y confirma su permiso para que la aplicación cliente acceda a su cuenta.

Si es un cliente de escritorio, Twitter proporciona un PIN de seguridad. Si es una aplicación web, es redirigido directamente a la aplicación cliente, ya con acceso.

El usuario introduce el PIN de seguridad en la aplicación cliente, que queda autorizada para acceder a su cuenta.

Desde julio de 2009 Twitter ofrece la posibilidad de utilizar este mecanismo de autorización para las aplicaciones que quieren acceder a su API. Actualmente es el mecanismo recomendado, ya que proporciona un nivel de seguridad mayor que la autenticación básica por dos razones: las credenciales no viajan en cada petición, y sobre todo no es necesario confiárselas a aplicaciones de terceros.

Twitter exige que las aplicaciones que quieran hacer uso de OAuth para acceder a su API sean registradas en su página web (http://twitter.com/oauth_clients).

1.2.2. Cifrado

A priori las comunicaciones a través de redes de acceso público como Internet no deben considerarse como confidenciales o privadas, dado que la información viaja por recursos de red que no son controlados por el usuario.

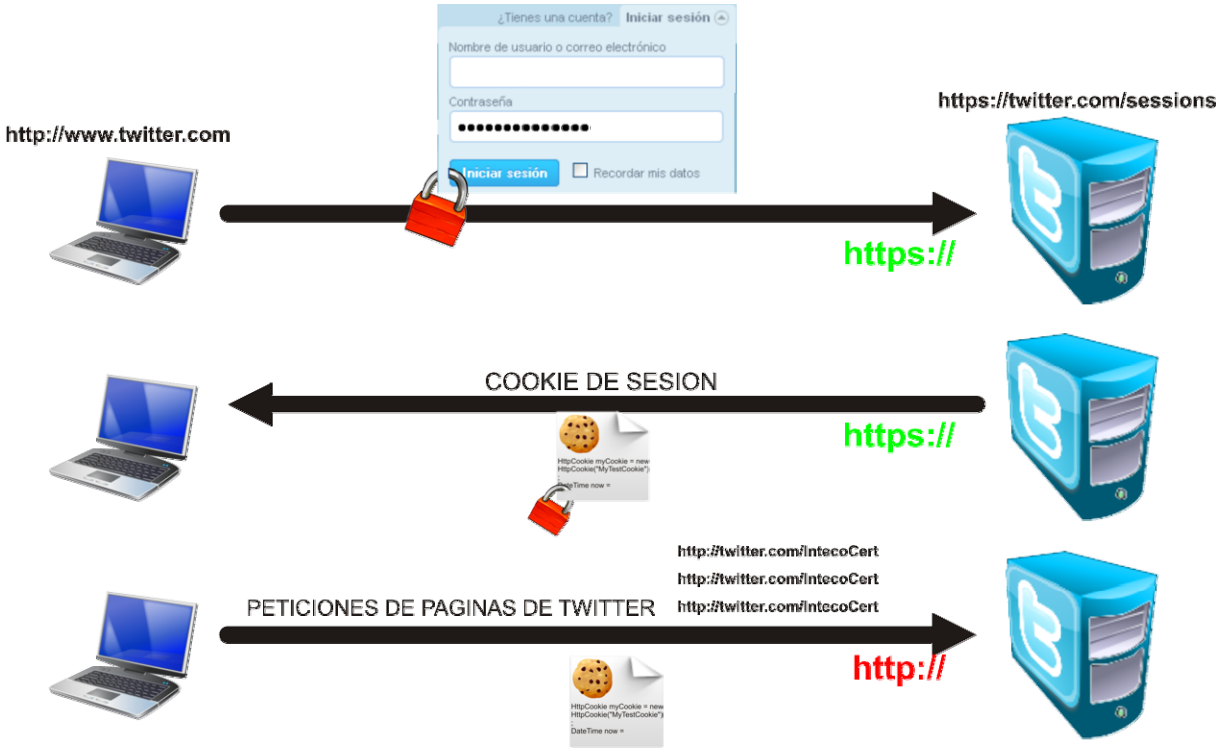
La interceptación y escucha de transmisiones ajenas puede ser realizada por ejemplo en una red local tanto cableada como inalámbrica con técnicas en cierta medida avanzadas,

pero que resultan fácilmente aplicables mediante determinadas herramientas al alcance de cualquiera (*sniffers* de red). Esto permite obtener entre otras cosas credenciales de acceso a sistemas securizados, con la consiguiente amenaza de suplantación de identidad que eso supone.

Para evitarlo, la comunicación entre el usuario y la aplicación ha de cifrarse utilizando algún algoritmo criptográfico que proporcione confidencialidad, junto con un protocolo adecuado para negociar ese cifrado.

En las comunicaciones web el protocolo estándar y probablemente conocido por todos es SSL/TLS sobre el puerto 443. En general, se puede saber que está siendo utilizado cuando la URL a la que accedemos comienza por http**s**.

La web de Twitter no utiliza cifrado de forma predeterminada, ya que al escribir en el navegador la dirección www.twitter.com accede a http://twitter.com, en lugar de redirigir a https://twitter.com (que también está disponible). No obstante, cuando tras rellenar el formulario el nombre de usuario y contraseña realizamos la petición de autenticación, ésta sí se dirige a una URL segura (https://twitter.com/sessions), con lo que las credenciales viajan cifradas por la red .(ver grafico *protocolos cliente-servidor*).



Sin embargo, de nuevo en peticiones subsiguientes la *cookie* viaja en claro por la red, posibilitando los ataques de secuestro de sesión ya mencionados anteriormente.

La situación es mucho más delicada si se utiliza la autenticación básica contra la API, ya que las credenciales viajan en cada petición y bastará con capturar uno de esos paquetes

para averiguar la contraseña del usuario, que puede ser utilizada para suplantar su identidad en cualquier momento que el atacante estime oportuno.

En el caso de la autenticación OAuth el problema es similar al de la autenticación web. Las credenciales nunca viajan por la red entre la aplicación de terceros y los servidores de Twitter (se utiliza el sistema cifrado de autenticación de la web oficial directamente), pero al igual que la *cookie*, el token autorizado sí se envía en cada petición. Si no se utiliza un protocolo de cifrado seguro, pueden producirse los mismos ataques de secuestro de sesión.

1.2.3. Almacenamiento de contraseñas

Como se ha expuesto anteriormente, los clientes que no utilizan el protocolo OAuth para la autenticación/autorización, deben conocer el nombre de usuario y contraseña para acceder a un perfil. Si no queremos tener que introducirlos en cada conexión al servidor (ya sea para enviar un *tweet* o recibir actualizaciones), es necesario que las aplicaciones almacenen esas credenciales.

Independientemente del lugar en el que se guarden (fichero de configuración, registro, base de datos…), los datos de autenticación pueden guardarse en texto claro, o bien cifrados con un algoritmo reversible para evitar que cualquier otro que acceda a esos datos pueda leerlos sin tener la clave correspondiente.

En el caso de utilizar cifrado existen diferentes posibilidades para su implementación. Algunos clientes optan por implementar un cifrado propio, aunque sea con algoritmos estándar, utilizando una clave fija. Un mecanismo en general más seguro es utilizar las funcionalidades de gestión de contraseñas que ofrecen los entornos como KDE (Kwallet), GNOME (GPass) o AIR (*encrypted local store*).

Existe cierta controversia sobre qué mecanismo de almacenamiento es más conveniente. Obviamente el cifrado bien implementado aporta un grado mayor de seguridad, pero existen quienes afirman que la seguridad de los permisos a nivel de sistema de ficheros (o de base de datos) deberían ser suficientes (al menos en sistemas operativos modernos), y en la práctica la mayoría de implementaciones no aportan un grado de seguridad mayor [3].

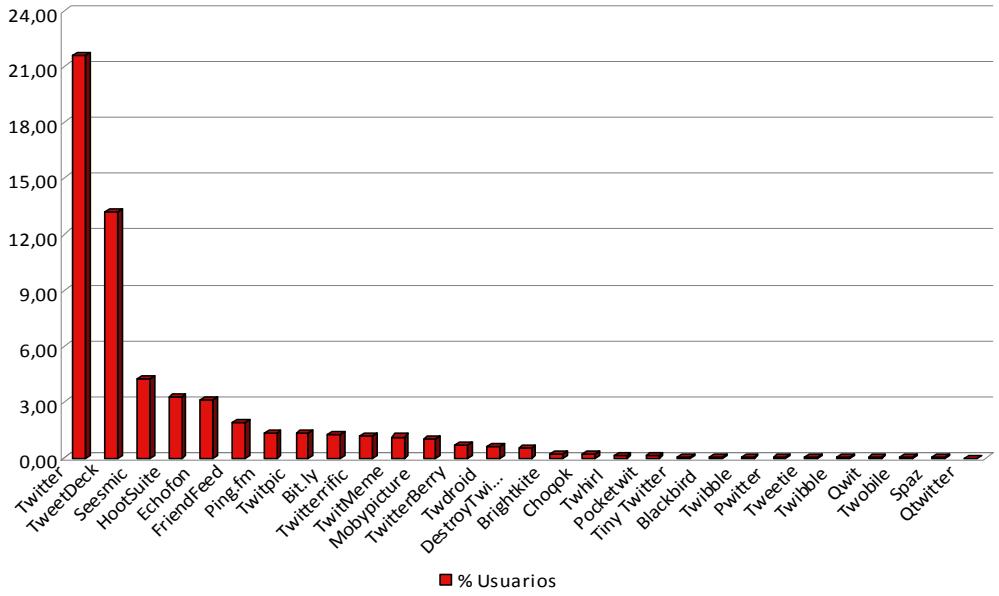
En cualquier caso es indiscutible que el hecho de proporcionar las contraseñas a una aplicación de terceros y tenerlas almacenadas en un sitio adicional (además de los servidores de Twitter) es una opción menos segura que simplemente que el usuario la memorice y la proporcione cuando sea necesario.

1.3. METODOLOGÍA

A la hora de seleccionar los clientes a utilizar, se han utilizado las estadísticas de uso global de clientes disponibles en twitstat.com [4] a fecha de 03 de Noviembre de 2009 (Gráfico 2). A pesar de no tratarse de una fuente oficial, las estadísticas están extraídas de la propia API de Twitter, que lleva un seguimiento del cliente con el que se envía cada *tweet*, por lo que se deben considerar fiables.

Ante la imposibilidad de encontrar estadísticas de uso por plataforma, se ha optado en principio por tomar los clientes más utilizados globalmente. Con el fin de cubrir la totalidad de plataformas consideradas (Windows, Linux, MacOS, iPhone, Android, Blackberry, WindowsMobile), se han añadido algunos de los clientes más populares disponibles para las mismas, aunque su uso no represente un porcentaje significativo del total.

Gráfico 2: Estadísticas de uso de clientes Twitter (03/11/2009)



Fuente: twittstat.com [4]

Para cada cliente se han realizado pruebas con el fin de determinar sus características de seguridad:

- **Mecanismo de autenticación**: se ha comprobado para cada cliente si el mecanismo que utilizan para acceder a la API es la autenticación básica o el protocolo OAuth. Debido a las diferencias en el proceso, es inmediato distinguir entre los dos simplemente añadiendo una cuenta a los clientes.

- **Cifrado de comunicaciones**: la información sobre el uso SSL en los clientes se ha extraído capturando el tráfico de red generado por los clientes, comprobando que en aquellos en que el cifrado es opcional, realmente la opción cumple su función. En el caso especial de los clientes web, es imposible conocer cómo se produce la comunicación con la API de Twitter. En ese caso se tiene en cuenta si se utiliza un protocolo seguro para el acceso del usuario a la aplicación.

- **Almacenamiento de contraseñas**: para los clientes que utilizan autenticación básica y por tanto necesitan almacenar la contraseña del usuario, se ha intentado

determinar si el almacenamiento de la misma se realiza en forma cifrada o de texto plano. La información se ha extraído principalmente de pruebas manuales y en su defecto, de la documentación del cliente, y en algunos casos del propio código fuente.

En el caso de aplicaciones multiplataforma, la mayoría de ellas utilizan un framework o lenguaje portable, por ejemplo AIR, .NET, Java o Python. En esos casos, suponiendo que las características de seguridad son similares en cualquier sistema, se ha elegido la opción más conveniente para realizar las pruebas.

2. PRUEBAS DE CLIENTES

A continuación se resumen en una tabla los resultados de las pruebas realizadas con los clientes de Twitter seleccionados.

El significado de cada columna, aunque resulta bastante autoexplicativo se detalla a continuación:

- **Versión**: la versión de la aplicación utilizada para hacer las pruebas, o en su defecto la fecha en el caso de las plataformas web, ante la ausencia de un número de versión accesible.

- **Porcentaje de usuarios**: porcentaje de usuarios que utilizan el cliente, según las estadísticas tomadas como referencia (ver apartado 1.3 sobre Metodología).

- **Plataforma**: en esta columna se especifica la plataforma utilizada para probar el cliente, independientemente de que la aplicación esté disponible para otros sistemas adicionales.

- **Autenticación**: protocolo de autenticación utilizado para acceder a la API, Básico o mediante OAuth, tal y como se especifican en el apartado 1.2.1.

- **SSL**: determina si se usan conexiones cifradas o no, o si es una opción que hay que activar.

- Clientes de escritorio/móviles: si el cliente utiliza comunicaciones cifradas para acceder a la API.

- Clientes web: si la comunicación entre el usuario y la plataforma web usa conexiones cifradas (HTTPS).

- **Cifrado de contraseñas**: determina si el almacenamiento de las contraseñas en el dispositivo que ejecuta el cliente es cifrado o no. No es aplicable en las plataformas web por la imposibilidad de conocer la infraestructura que se utilice en sus servidores.

Tabla 1. Resultados de pruebas con clientes Twitter

Cliente	Versión	% de Usuarios	Plata-forma	Autenticación	SSL	Cifrado de contraseñas
Twitter	03/11/2009	21,6	WEB	N/A	Opcional	N/A
TweetDeck (Desktop)	v0,31,4	13,23	Windows	Básica	Sí	Si
Seesmic (Desktop)	v0,66	4,22	Windows	Básica	Sí	Si
HootSuite	03/11/2009	3,32	WEB	Básica	No	N/A
Echofon	v2.1.1	3,11	Iphone	Básica	Sí	Si
FriendFeed	03/11/2009	1,9	WEB	Básica	Sí	N/A
Ping.fm	03/11/2009	1,35	WEB	Básica	Opcional	N/A
Twitpic	03/11/2009	1,35	WEB	Básica	Opcional	N/A
Bit.ly	03/11/2009	1,3	WEB	Básica	No	N/A
Twitterrific (MacOS)	v3.2.1	1,2	Mac	Básica	Sí	
Twitterrific (iPhone)	v2.1	1,2	Iphone	Básica	Sí	Si
TwitMeme	03/11/2009	1,15	WEB	OAuth	No	N/A
Mobypicture	03/11/2009	1	WEB	Básica	No	N/A
Mobypicture (Android)	v20091910	1	Android	Básica	Sí	Si
TwitterBerry	v0.9.1.1	0,7	Blackberry	Básica	Sí	Si
Twdroid	v2.7.1	0,65	Android	Básica	Sí	Si
DestroyTwitter	v1.7.2 Beta	0,58	Windows	Básica	Sí	Si
Brightkite	v1.2.2	0,25	Android	Básica	Sí	Si
Choqok		0,2	Linux	Básica	Opcional	Si
Twhirl	v.0..9.4	0,17	Linux	Básica	Sí	Si
Pocketwit	v75b	0,14	Wmobile	Básica	No	Si
Tiny Twitter	v1.8.4	0,1	Wmobile	Básica	No	No
Blackbird	v0.5.23	0,05	Blackberry	Básica	Sí	
Twibble	V0.8.3	0,05	Blackberry	Básica	Sí	Si
Pwitter	V1.1.6	0,05	Mac	Básica	Sí	Si
Tweetie	v1.2.4	0,05	Mac	Básica	Sí	Si
Tweetie	v2	0,05	Iphone	Básica	Sí	Si

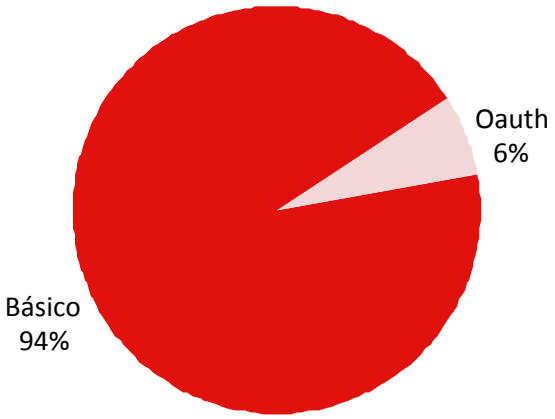
Cliente	Versión	% de Usuarios	Plataforma	Autenticación	SSL	Cifrado de contraseñas
Twibble	v0.5.6	0,05	Linux	Básica	Sí	Si
Qwit	v1.0-Beta	0,05	Linux	Básica	Opcional	Si
Twobile	v1.7.5	0,05	Wmobile	Básica	Opcional	Si
Spaz	v0.8.2	0,03	Linux	Básica	Opcional	Si
Qtwitter	V0.6.6	N/A	Linux	OAuth	Sí	N/A
Twitter	03/11/2009	21,6	WEB	N/A	Opcional	N/A
TweetDeck (Desktop)	v0,31,4	13,23	Windows	Básica	Sí	Si
Seesmic (Desktop)	v0,66	4,22	Windows	Básica	Sí	Si
HootSuite	03/11/2009	3,32	WEB	Básica	No	N/A

Fuente: INTECO y twittstat.com(% de uso)

3. ESTADÍSTICAS

A continuación se muestran diferentes gráficas que recogen la distribución de los clientes para cada una de las características de seguridad analizadas.

Gráfico 3: Mecanismos de Autenticación por porcentaje de clientes



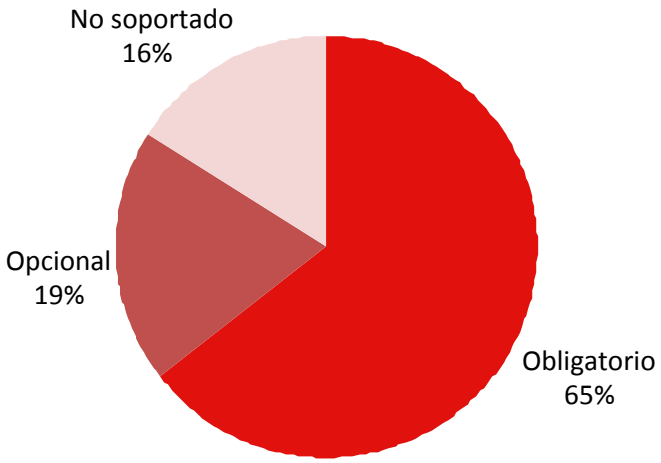
Fuente: INTECO

Relativo al mecanismo de autenticación, se puede ver en el Gráfico 3 que el uso de OAuth frente a la autenticación básica es mínimo. Esto seguramente tenga dos motivos:

- El proceso de autenticación básico era el único disponible inicialmente y OAuth sólo ha estado disponible desde julio de 2009. Es difícil persuadir a los desarrolladores de aplicaciones anteriores a esa fecha para que cambien algo que está funcionando.

- Aunque los desarrolladores lo conozcan, algunos argumentan que el proceso de autorización inicial de la cuenta para el usuario es demasiado engorroso [5].

Gráfico 4: Uso de cifrado en las comunicaciones

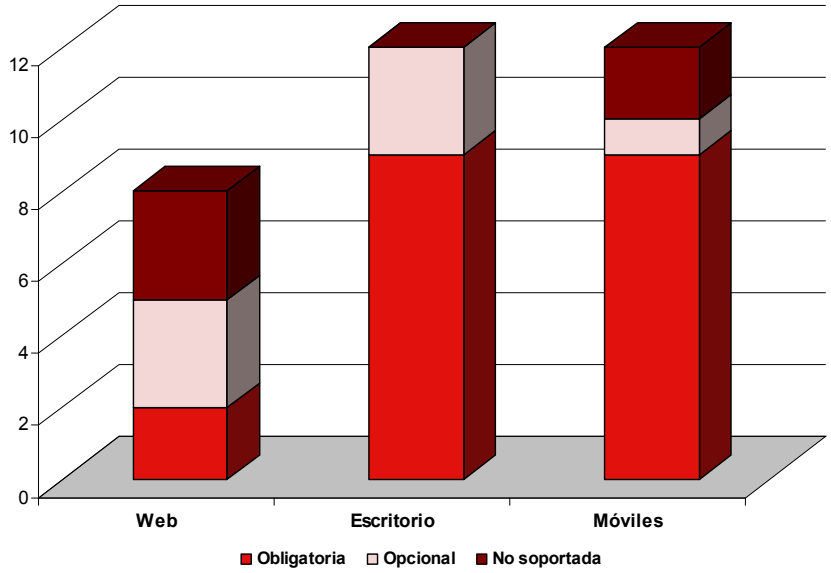


Fuente: INTECO

En el caso del cifrado de las comunicaciones (Gráfico 4) el uso es mayoritario con un 84%, aunque está activado por defecto en un porcentaje bastante menor (65%).

Es importante resaltar que solamente en uno de los casos en los que el cifrado no está soportado, se utiliza autenticación mediante OAuth. Eso significa que en el resto, el robo de credenciales sería inmediato con la captura de una sola petición del usuario.

Gráfico 5: Uso de cifrado en las comunicaciones por plataforma



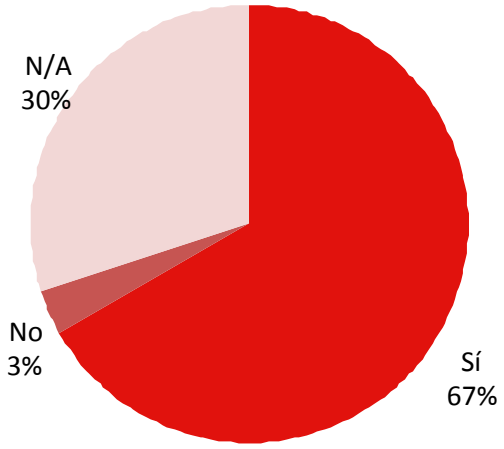
Fuente: INTECO

Como se puede ver en el Gráfico 5, el cifrado es mucho más utilizado en plataformas distintas de escritorio y móviles. Es sorprendente que en una plataforma en la que los mecanismos de conexión segura como http/ssl están tan extendidos para otras aplicaciones, éstos no se utilicen en más ocasiones. Hay que recordar que para las aplicaciones web estos datos se refieren a la comunicación entre el usuario y el cliente, sería deseable poder comprobar cómo es la comunicación entre la aplicación y la API de Twitter, para asegurar que la ruta de comunicación es confidencial en todos los tramos, con lo que la situación podría ser aún peor de lo que muestran las estadísticas.

Es destacable que entre las plataformas móviles, los clientes que no soportan cifrado son todos para Windows Mobile, que dispone solamente de un cliente con cifrado opcional. En el resto de sistemas operativos móviles se usa de forma obligatoria.

En sistemas Linux el cifrado es mayoritariamente opcional, algo que puede verse en consonancia con las posibilidades de personalización y un perfil de usuario avanzado que prefiere controlar la configuración de sus aplicaciones. En todos los clientes Mac y Windows, el uso de comunicaciones seguras es obligatorio.

Gráfico 6: Almacenamiento cifrado de contraseñas



Fuente: INTECO

En el caso de que las aplicaciones tienen que almacenar las credenciales del usuario por utilizar la autenticación básica, vemos que casi el total de aquellas en las que se ha podido comprobar, aplican las prácticas recomendadas. No obstante, sería necesario comprobar si los mecanismos de cifrado utilizados aportan realmente un grado de confidencialidad suficiente para confiar en que nuestras contraseñas están seguras.

En la mayoría de los casos se utilizan mecanismos de almacenamiento de contraseñas estándar que podemos considerar seguros, pero al menos en una de las aplicaciones se ha comprobado que se utiliza una clave fija con un cifrado XOR fácilmente reversible (Qwit).

4. CONCLUSIÓN Y RECOMENDACIONES

Tras analizar las características de seguridad de los clientes de Twitter más utilizados se pueden extraer las siguientes conclusiones:

- El uso de OAuth no está extendido, pero es de esperar que se adopte tarde o temprano, dada la intención de Twitter de eliminar el soporte para la autenticación básica [2]. Este mecanismo tiene ventajas importantes en cuanto a seguridad, y creemos que a pesar de su aparente mayor complejidad, una vez que su uso esté popularizado y existan librerías funcionales para la mayoría de lenguajes, todos los clientes terminarán por migrar a esta solución, para el beneficio de los usuarios.

- Aunque se utilice autenticación básica, la mayoría de clientes utilizan conexiones cifradas de tal forma que el usuario está protegido frente a robo de credenciales y secuestros de sesión.

- No obstante, aunque exista protección ante ataques de terceros, se siguen confiando las credenciales de acceso a una aplicación de la que muchas veces no se puede determinar su nivel de seguridad o su legitimidad.

Teniendo estos puntos en cuenta, desde INTECO sugerimos las siguientes recomendaciones:

- Los usuarios deben comprobar las características de seguridad de los clientes, buscando un compromiso entre funcionalidad y nivel de seguridad deseado, dando preferencia a aquellos clientes que utilicen OAuth y cifren las comunicaciones.

- Los desarrolladores de aplicaciones deben conocer y seguir en la medida de lo posible las recomendaciones de Twitter [2], que se pueden resumir en utilizar OAuth para la autenticación y sobre todo cifrar las comunicaciones para cualquier acceso que requiera autorización.

- Por parte de Twitter existen posibles medidas a tomar para mejorar la seguridad: ofrecer a los usuarios utilizar SSL para todas las peticiones como una opción de configuración de la cuenta (tal y como hace Google en Gmail), de tal forma que si se activa, automáticamente redirige tus peticiones al sitio HTTPS.

5. REFERENCIAS

1. Estudio sobre la privacidad de los datos personales y la seguridad de la información en las redes sociales online, *INTECO, 2009.*

2. http://apiwiki.twitter.com/Security-Best-Practices

3. http://developer.pidgin.im/wiki/PlainTextPasswords.

4. http://www.twitstat.com/twitterclientusers.html.

5. http://blog.atebits.com/2009/02/fixing-oauth/