1 Curso POO PHP Secuestro e fixación da sesión

1.1 Secuestro e fixación da sesión

Como xa vimos, PHP emprega sesións para almacenar a información dos usuarios. Cada unha das sesións identifícase mediante un ID que o usuario transmite en cada petición para identificarse. Esta transmisión do ID pode realizarse mediante un parámetro GET, ou mediante unha Cookie.

Existen dous tipos de ataques que se basean neste identificador: o secuestro e a fixación da sesión.

1.1.1 Secuestro da sesión (*session hijacking*)

Chamamos secuestro da sesión a calquera método ou técnica que teña como fin conseguir o identificador da sesión dun usuario. As formas de conseguir este obxectivo son variadas, como por exemplo interceptando o tráfico da rede, ou incluso realizando un ataque de forza bruta (probando moitos IDs) ata conseguir un válido.

Unha das primeiras medidas para evitar o secuestro da sesión dos usuarios é **enviar os identificadores do cliente ao servidor empregando sempre Cookies**. Deste xeito, o identificador non aparece na URL de cada unha das páxinas, o que podería ocasionar o envío ou a publicación accidental do mesmo polo usuario ao querer obter o enlace dunha páxina calquera.

PHP permite aos usuarios empregar o método GET cando as cookies non están habilitadas. Este comportamente pode modificarse empregando a directiva de configuración **session.use_only_cookies**, que a partir da versión 5.3.0 xa se atopa por defecto a '1'.

```
session.use_only_cookies = 1
```

Cando sexa posible tamén é recomendable **empregar conexións seguras con HTTPS** para evitar a interceptación do tráfico.

Outra medida para dificultar o secuestro da sesión é **limitar o seu tempo de vida**. Isto pode facerse de dúas maneiras:

- Empregando a directiva **session.cookie_lifetime** para indicar o tempo de caducidade dunha sesión. Transcurrido ese tempo, a sesión pasa a considerarse inválida e deberá abrirse unha nova. Isto messmo pode facerse coa función **session_set_cookie_params**.
- Empregando a directiva **session.gc_maxlifetime**, que indica o tempo de inactividade dunha sesión. Pasado ese tempo, a sesión pasa a considerarse inválida.

Tamén podemos **cambiar o ID da sesión** de cando en vez empregando a función **session_regenerate_id**. Este método é transparente para os usuarios activos pero provoca a invalidez do anterior identificador, que xa non poderá empregarse con fins maliciosos en caso de ter sido capturado.

Para rematar, outra medida é **comprobar en cada petición os datos relacionados co cliente**, como poden ser o axente de usuario (**$_SERVER['HTTP_USER_AGENT']**) e en casos extremos tamén a dirección orixe da petición (**$_SERVER['REMOTE_ADDR']**). Cando algunha das dúas cambia con respecto á obtida na anterior petición, sobre todo a primeira, debería anularse a sesión e forzar ao usuario a rexistrarse de novo para comezar unha nova sesión.

1.1.2 Fixación da sesión (*session fixation*)

No método de secuestro da sesión se descoñece o identificador do cliente e o que se busca é tratar de descubrir cal é. Nesta ocasión, o que se busca é acadar que o usuario se loguee cun identificador de sesión que xa coñecemos.

Para evitar estes ataques deben terse en conta as mesmas recomendacións mencionadas anteriormente para evitar o secuestro da sesión, e tamén revisar o valor do parámetro **session.use_trans_sid** na configuración de PHP.

```
session.use_trans_sid = 0;
```

Se o valor do parámetro anterior fose '1', PHP lería os identificadores de sesión pasados nun parámetro GET da URL, aínda que o usuario estivese a empregar cookies para a súa transmisión. Isto posibilitaría a creación de URLs fraudulentas que inclúan identificadores de sesión coñecidos por un atacante.

--Víctor Lourido 13:45 26 jul 2013 (CEST)