HDC

¿Qué es una cookie?

Cuando hablamos de **cookies** no nos referimos a las galletas que comemos en la cocina. No es un disco marrón y, por cierto, no tiene chispas de chocolate. Una cookie **es un archivo de texto con información que un servidor web almacena en un navegador de un cliente**, para **mantener una sesión** activa (por ejemplo, que nos mantengan logueados en nuestra web de email aunque cerremos la página), recordar ciertos aspectos de **personalización** (como los carritos de compras) o para hacer **seguimiento** y **recolectar información** de aquellos usuarios. Es importante destacar que **las cookies son sólo datos, no código**, así que no pueden tener actos dentro de la computadora de ustedes. Eso sí, los datos que están almacenados pueden ser valiosos y no queremos que cualquiera los obtenga.

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html
Set-Cookie: PREF=ID=5e66ffd215b4c5e6:
TM=1147099841:LM=1147099841:S=Of69MpW
Bs23xeSv0; expires=Sun, 17-Jan-2038 1
9:14:07 GMT; path=/; domain=.google.c
om
```

Repito. **Cada vez que uno navega, deja rastros.**

¿Qué es la ley de cookies?

Cuando hablamos de una ley, significa que está **regulado**. Que a alguien se le ocurrió poner una traba a la desmedida acción de instalarnos cookies, cuando estamos hablando de recolección de datos.

Naturalmente, **las cookies que son necesarias estrictamente para el funcionamiento normal de la página web no necesitan autorización** de parte del usuario que este navegando, como el ejemplo que dije sobre los carritos de compra; o **tampoco avisará cuando sea por un servicio que pidió el cliente expresamente como recordar una contraseña**, aunque sí es necesario que aparezca el aviso legal de parte de la página. Pero la ley de cookies dice que **necesita permiso de parte del**

usuario en tres casos:

- **Cookies sin capacidad de identificación del usuario** (digamos que entonces no recolecta datos del cliente), si no están incluidas como necesarias.
- **Cookies con capacidad de identificación del usuario**, que almacenan información sobre nosotros, intrusas, y que muchas veces se utilizan para medios publicitarios personalizados.
- **Cookie aceptada en la configuración del navegador**, aunque estén pre aceptadas.

NOTA: A veces no aparece un cartel en rojo con un "aceptar" para saber de estas cookies, sino un cartel por debajo, pequeño que dice que **si seguís navegando** en la web, quiere decir que **aceptaste los términos**.

Pero cuando alguien se empeña en hacer algo, lo consigue. Y una de las cosas más valiosas, hoy en día, es la información. Por esto se han creado las **evercookies**. Claro, cuando uno borra las cookies del navegador no podrían recolectar la data del usuario, en cambio estas evercookies **se encargan de replicarse de todas las maneras "legales" posibles** para permanecer dentro. Pero, con maña o navegando en el modo incógnito se pueden eliminar estas cookies. Así que ahora también tenemos las **supercookies** o **cookies por HSTS**.

El **HSTS** es una **propiedad** que tienen los navegadores (a profundizar en otra clase) para **obligar a que ciertas páginas funcionen estrictamente por HTTPS**. Tiene un mecanismo de seguridad interesante en el que se asigna un **ID único**. Es decir, un número de reconocimiento sólo para ti. Si, ya saben. Aunque esté en incógnito, usa el mismo ID. Las super cookies se aprovechan de esto para trackear users y recolectar info.

¿Puedo robar una cookie?

O quizás la primer pregunta sea **¿Para qué quisiera robar una cookie?** Como dijimos antes, las cookies almacenan **datos de sesiones** -con las que, por ejemplo, no necesitamos volver a ingresar usuario y contraseña para seguir logueados en esa web. Se supone que si alguien roba esa cookie, podría acceder a estos datos y loguearse. Si lo pensamos claramente, el servidor web comprueba la cookie con la IP específica y **si no es la misma IP** o el mismo valor de cookie, o si algún parámetro está mal simplemente no nos va a dejar conectarnos con esa sesión. Desde ese punto, tenemos que saber que si vamos a robar las cookies, entonces deberíamos estar conectados en la **misma red que la víctima**. A este ataque se le da el nombre

de **session hijacking (secuestro de sesión)**. Vamos a hacer una demostración de un secuestro de sesión :).

Session Hijacking

Para lograr el secuestro de la sesión, necesitaremos usar:

- **Wireshark**, que para el que no vio como usarlo, está en la clase anterior.
- **Mozilla Firefox**, navegador.
- **GreaseMonkey**, un complemento para el navegador de Firefox
- **Cookie Injector**, un script que es para GreaseMonkey y que nos va a ayudar a usar la cookie editada

Para instalar **GreaseMonkey** simplemente lo hacemos como cualquier complemento desde esta dirección: **https://addons.mozilla.org/es/firefox/addon/greasemonkey/**

Para instalar el **script**, lo hacemos luego de tener GreaseMonkey desde aquí: **http://userscripts-mirror.org/scripts/show/119798** (arriba a la derecha hay un botón de install).

Como haremos la prueba, usaremos **www.themagicgames.net** como medio para robar la sesión de la víctima. Si ven que con esta técnica no les funciona, intenten con otra página.

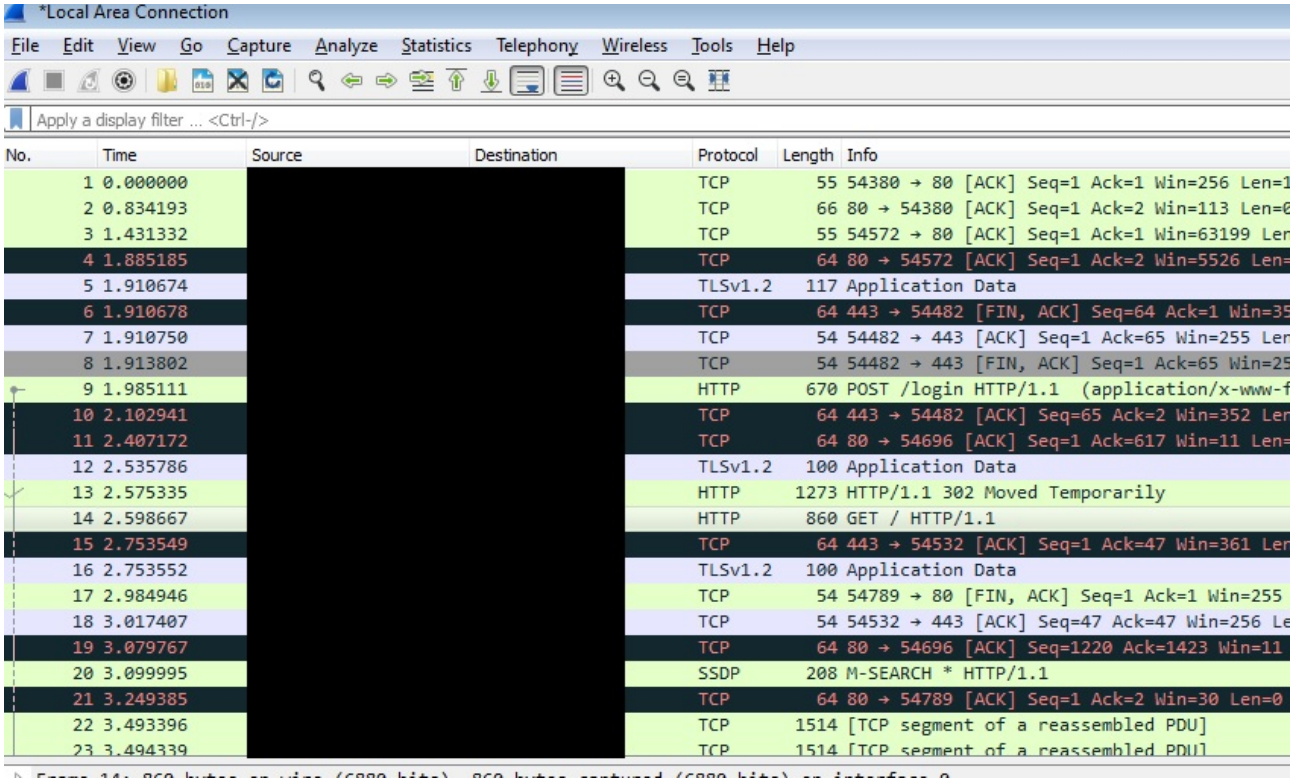
Para testear que funciona, usaré **Google Chrome para identificarme y Mozilla Firefox para autenticarme** desde el nivel del delincuente, porque las cookies son independientes de cada navegador.

NOTA: recuerden que el hijacking de sesiones es importante que se haga desde la misma red de la víctima.

Vamos paso a paso. **Primero nos creamos una cuenta en la web indicada**. En mi caso, la dicha anteriormente.

Completamos todos los pasos para registración para tener la cuenta activa.

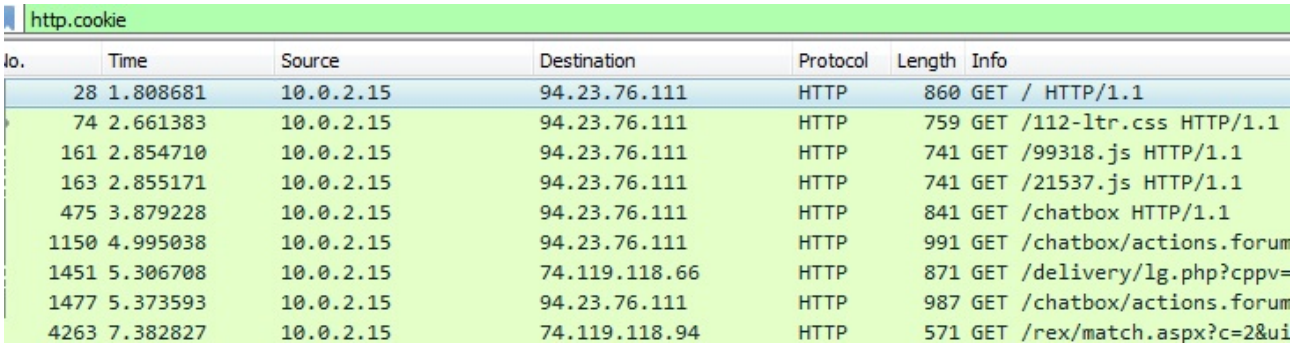
Segundo, ponemos **Wireshark** a capturar paquetes.



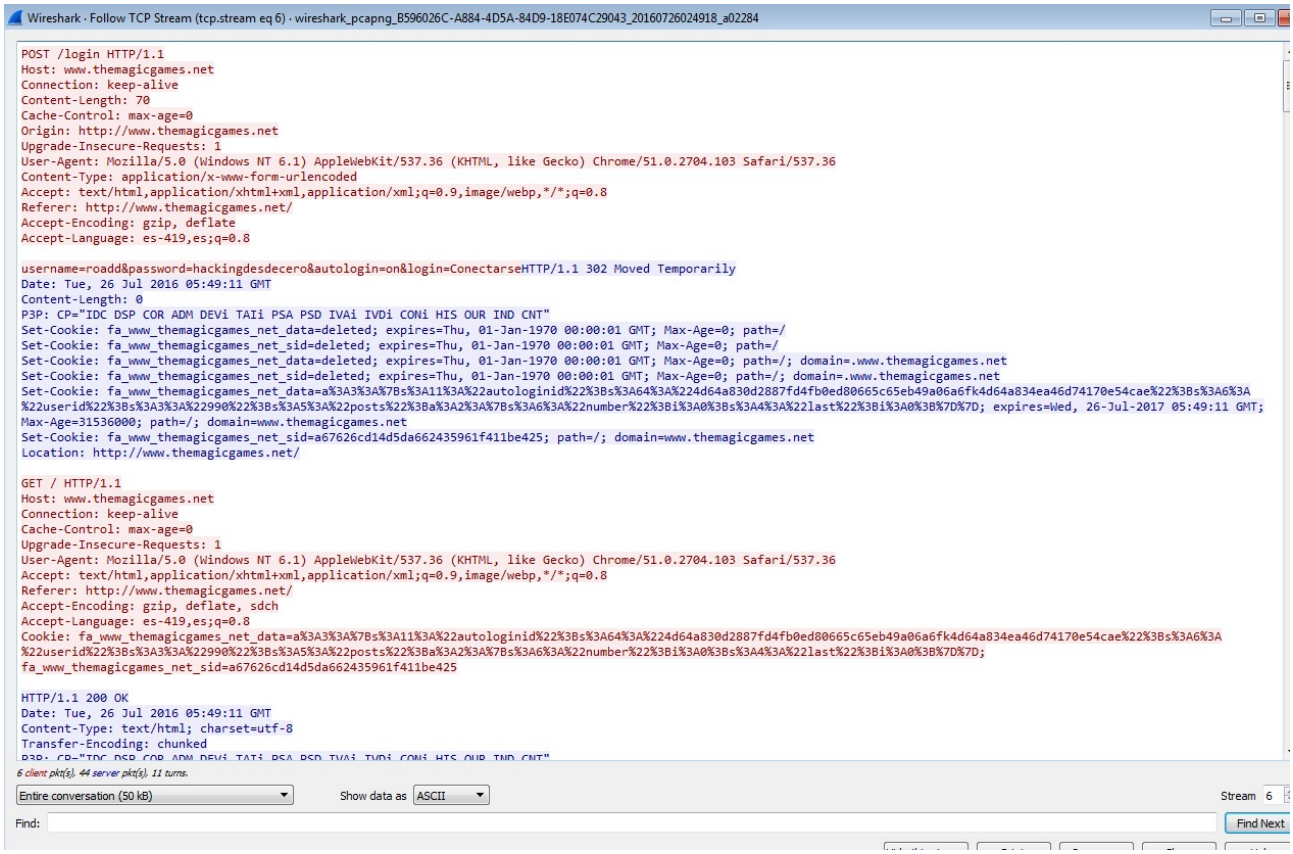
Tercero, desde Chrome, **iniciamos sesión** en la web.

Paramos **Wireshark** y desde el filtro buscamos **http.cookie**.

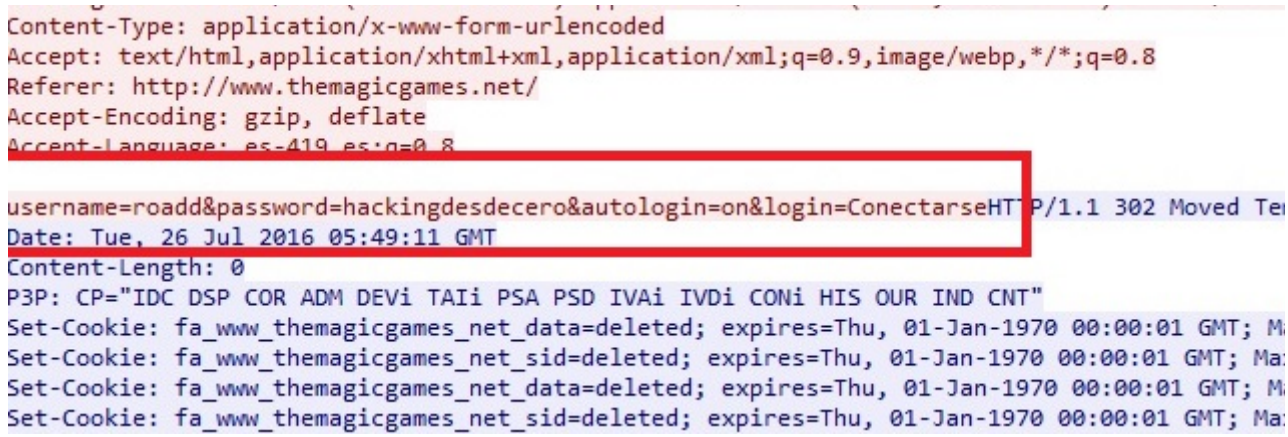
A mi me quedó algo como esto. Aclaro que al tener el Wireshark funcionando durante tan poco tiempo, es fácil de ver dónde está. En el caso de que no lo encontremos, hay algunas otras técnicas que nos pueden ayudar como agregar al filtro “post contains login” y otras chácharas.



El primer paquete de todos dice que es un GET a / (veremos más adelante que es lo que quiere decir exactamente). Por ahora nos quedamos con éste, porque allí será donde pida el **index.html** de la web. Lo que los mortales conocen como la “**home**”. Quizás hubo algún intercambio interesante. Por si no recuerdan, porque no han practicado, **click derecho en el paquete → follow → Tcp Stream.**
Nos saldrá una ventana con la conversación entre cliente y servidor.



Por si no llegan a ver, en la primer petición, el cliente (la víctima) envía el usuario y el **password en texto plano** xD. Es decir que en realidad no nos deberíamos volver locos editando e intentando de que acepten la cookie, esperando que el contrario no se desconecte. Pero vamos a seguir con esto que es lo que quiero que aprendan, ya que en realidad no pasa nunca en texto plano, sino que se le pasa por un algoritmo de hash para que esto no pase.

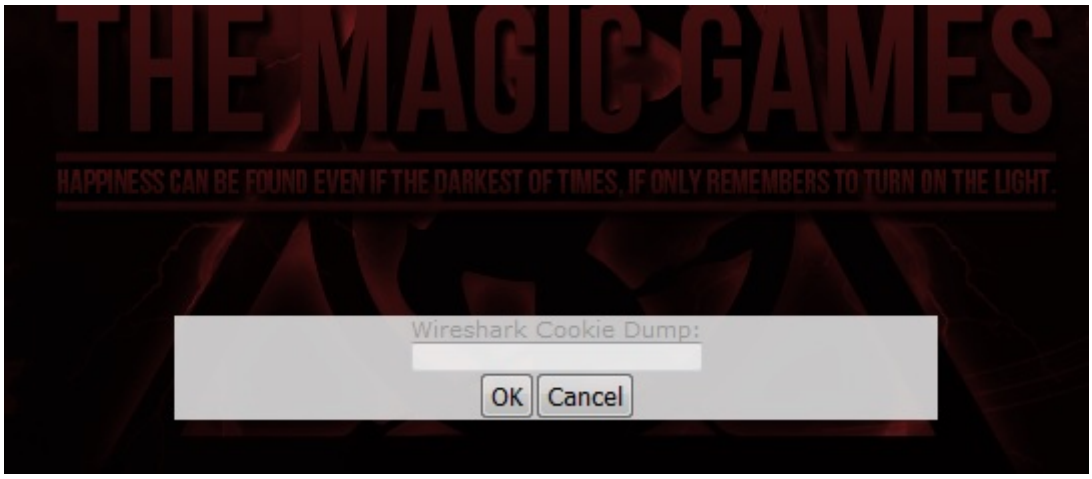


Muy lindo todo. En el próximo turno del cliente, vemos como envía la **cookie** con “Cookie: blablabla”.

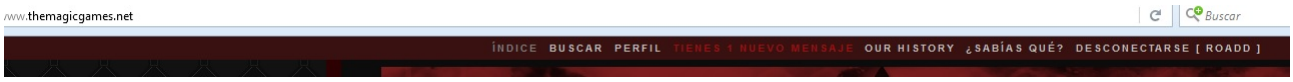
```
GET / HTTP/1.1
Host: www.themagicgames.net
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://www.themagicgames.net/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: es-419,es;q=0.8
Cookie: fa_www_themagicgames_net_data=a%3A3%3A%7Bs%3A11%3A%22autologinid%22%3Bs%3A64%3A%224d64a830d2887fd4fb0ed80665c65eb49a06a6fk4d64a834ea46d74170e54cae%22%3Bs%3A6%3A
%22userid%22%3Bs%3A3%3A%22990%22%3Bs%3A5%3A%22posts%22%3Ba%3A2%3A%7Bs%3A6%3A%22number%22%3Bi%3A0%3Bs%3A4%3A%22last%22%3Bi%3A0%3B%7D%7D;
fa_www_themagicgames_net_sid=a67626cd14d5da662435961f411be425
```

Vamos a copiar todo el blablabla, desde fa_www hasta e425. Son como 3 líneas a copiar.

Ahora vamos a Firefox, entramos a la misma web y presionamos **alt + c** (que sería el comando para llamar al cookie injector de GreaseMonkey). Les saldrá una ventanita así:



Allí dentro, copiamos toda la cookie que copiamos desde Wireshark, le damos a OK y recargamos la página.



Todo perfecto. Tenemos la sesión en nuestras manos. Por hoy es todo lo que veremos porque esta clase está trabándome de empezar otras. :D Pronto seguiremos con más clases, pequeños aprendices.

Pueden seguirme en Twitter: @RoaddHDC

Contactarse por cualquier duda a: r0add@hotmail.com

Para donaciones, pueden hacerlo en bitcoin en la dirección siguiente:

1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw

También recomiendo que se unan al foro: underc0de.org/foro

Este tutorial puede ser copiado y/o compartido en cualquier medio siempre aclarando que es de mi autoría y de mis propios conocimientos.

Roadd.