

06

## Sequestro de sessão

### Transcrição

Ao acessar uma URL é feita uma requisição ao servidor e automaticamente é aberta uma sessão, ou seja, um túnel de comunicação. Uma vez que a sessão é aberta é criado um número de identificação, pois o servidor possui uma tabela com chave e valor capaz de identificar o usuário que está enviando a requisição. O complemento da identificação é enviado através de uma resposta, o *Cookie*. Assim, o browser possui um *Cookie* que contém um número de identificação e através dele o servidor identifica quem somos. Resumindo:

The screenshot shows the 'Application' tab in the Google Chrome Developer Tools. On the left, there's a sidebar with sections for 'Application', 'Manifest', 'Service Worker', and 'Clear storage'. The main area is a table with columns: Name, Value, Dom..., Path, Expir..., Size, HTTP, Secure, and Sam... . There are four rows of data:

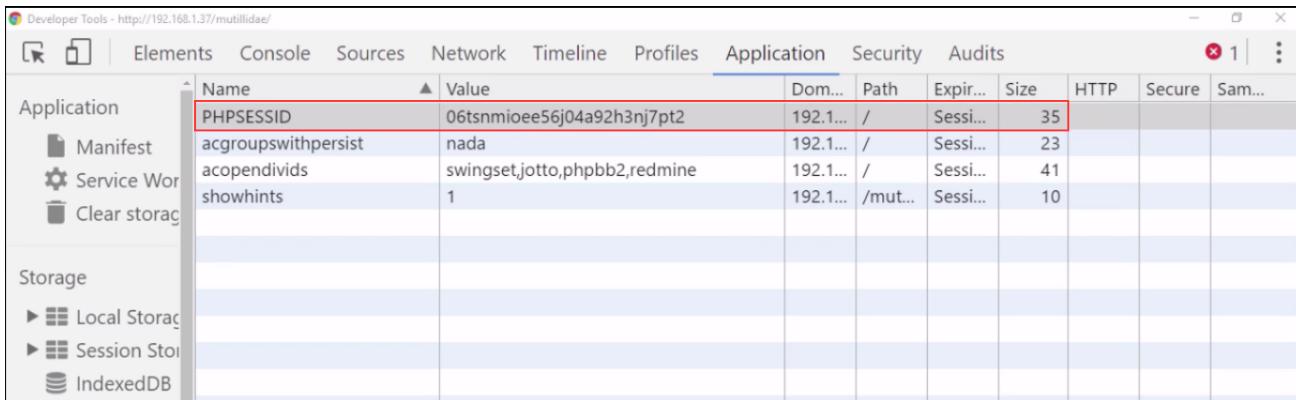
Name	Value	Dom...	Path	Expir...	Size	HTTP	Secure	Sam...
PHPSESSID	06tsnmioee56j04a92h3nj7pt2	192.1...	/	Sessi...	35			
acgroupswithpersist	nada	192.1...	/	Sessi...	23			
acopendifvids	swingset,otto,phpbb2,redmine	192.1...	/	Sessi...	41			
showhints	1	192.1...	/mut...	Sessi...	10			

Acessando o site do *Multillidae* como hackers verificamos que existe um *Cookie* referente a nosso acesso. Podemos verificar isso clicando com o botão direito do mouse e "Inspect Element(Q) > Network". Apertamos "Reload" para recarregar a página e selecionamos a página *Multillidae* e temos o seguinte:

The screenshot shows the Mozilla Firefox Network tab. The request URL is `http://192.168.1.37/mutillidae/`. The Headers section shows the following details:

- Request URL: `http://192.168.1.37/mutillidae/`
- Request method: GET
- Remote address: 192.168.1.37:80
- Status code: 200 OK
- Version: HTTP/1.1
- Keep-Alive: timeout=15, max=100
- Server: Apache/2.2.14 (Ubuntu) mod\_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Pa.../2.2.14 OpenSSL/0.9.8k Phusion\_Passenger/4.0.22
- Vary: Accept-Encoding
- X-Powered-By: PHP/5.3.2-1ubuntu4.30
- Host: 192.168.1.37
- User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101 Firefox/45.0
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8
- Accept-Language: en-US,en;q=0.5
- Accept-Encoding: gzip, deflate
- Cookie: showhints=1; PHPSESSID=pudi1g6rmvokplp0d4cjvld34
- Connection: keep-alive
- Cache-Control: max-age=0

Podemos verificar, nesse usuário, o item *Cookie* e nele aparece o número de identificação da máquina. Esse número é único, diferente de qualquer outro usuário. Por exemplo, se acessarmos o mesmo site através do *Google Chrome* vamos verificar que o *cookie* possui um número distinto:



The screenshot shows the Google Chrome Developer Tools Application tab. It displays a table of cookies. The columns are: Name, Value, Dom..., Path, Expir..., Size, HTTP, Secure, and Sam... (partially visible). One cookie, PHPSESSID, is highlighted with a red border.

	Name	Value	Dom...	Path	Expir...	Size	HTTP	Secure	Sam...
Application	PHPSESSID	06tsnmioee56j04a92h3nj7pt2	192.1...	/	Sessi...	35			
Manifest	acgroupswithpersist	nada	192.1...	/	Sessi...	23			
Service Worker	acopendivids	swingset,jotto,phpbb2,redmine	192.1...	/	Sessi...	41			
Clear storage	showhints	1	192.1...	/mut...	Sessi...	10			
Storage									
▶ Local Storage									
▶ Session Storage									
IndexedDB									

Os números são diferenciados pois o próprio servidor identifica que são dois usuários distintos!

Tendo compreendido isso podemos pensar o seguinte: E se o *hacker* conseguisse de alguma maneira ter acesso ao número *cookie* do usuário que acessou pelo *Google Chrome*? E se o hacker inserisse esse número na página do navegador com o intuito de enganar o servidor?

O que aconteceria é que o servidor seria efetivamente enganado, pois entenderia que o hacker é o mesmo usuário referenciado pelo número do **Cookie**, o usuário do **Google Chrome**.

Este fenômeno que acabamos de descrever chama-se **sequestro de sessão**. Vamos efetuar um ataque deste mesmo gênero no próximo vídeo.