Writeup P1 Tarea 1

Writeup P1 Tarea 2

Nombre: Benjamín Aguilar Osorio
Equipo: L05 1NCR31BL35

Persona a cargo de la pregunta **Benjamín Aguilar**

Descripción general del problema

Se trata de un problema de web, donde se debe conseguir informacion oculta en la base de datos que ocupa el servidor.

Metodología de resolución

A continuación se darán los **pasos** que se llevaron a cabo para resolver el problema:

1. Inicialmente, se llevó a cabo una revisión global del inspector de la página web. Primero se revisó el inspector HTML encontrando un comentario oculto que indicaba la supuesta existencia de "gatitos_secretos", pero sin encontrar mayor información.
2. Luego de revisar el HTML de la página, se decidió revisar la red del servidor, donde se pudo notar que uno de los servicios a los que el servidor le hacía una solicitud GET arrojaba un error, sin darle mayor importancia se continuó con la revisión del inspector.
3. Luego, para revisar el código presente en el depurador, se ingresó este en un beautifier para poder leerlo de una mejor manera, pero sin encontrar mayor información.
4. Se revisó el resto de elementos presentes en el inspector, como la memoria, el almacenamiento, etc., pero sin ningún resultado.
   5.Finalmente, se volvió a revisar la red del servidor, indagando más en la página que solicitaba el Favicon, puesto que generaba un error. En la respuesta de esta página se pudo notar que arrojaba un error por una URL incorrecta y daba 2 URLs que supuestamente podrían funcionar, por lo que se probó a utilizar la URL "https://t2p1.cc5325.hackerlab.cl/gatitos/".
5. Al ingresar a la URL, esta simplemente muestra un mensaje donde se informa acerca de una manera de encontrar el año de los gatos. Para supuestamente encontrar el año de estos, se necesitaría enviar el ID del gato("gatito_id") y "date=year", por lo que se decidió investigar en diversos foros cómo sería posible enviar información a un servidor.
6. Finalmente, se llegó a la solución de utilizar "query string" mediante la URL de la página. Para esto, simplemente se le hace una consulta a la página mediante el uso del signo "?". En este caso, se hizo la consulta "https://t2p1.cc5325.hackerlab.cl/gatitos/?gatito_id=1&date=year".

7. Como resultado de la consulta realizada, se consiguió lo que parecía ser una lista con todos los tipos de gatos que podrían salir en el GIF de la página principal, donde cada gato tenía parámetros de tiempo, la URL del GIF y un mensaje secreto. Este último siendo un string vacío para todos los gatos.
8. Con el fin de averiguar más acerca de los posibles resultados que podría arrojar la URL al modificar los valores de la consulta, se decidió intentar hacer una consulta que solicitara el "secret_string", fallando en el intento. Además, se optó por ir variando los valores de "gatito_id" y de "date". Inicialmente, solamente se modificaron los valores de "gatito_id", consiguiendo siempre el mismo resultado, por lo que se optó por la variación del parámetro "date".
9. Al variar el parámetro "date", este siempre entregaba un error, donde si el parámetro entregado no fuese "year/month/day", nos arrojaría un error de tipado, solicitando que el tipo entregado fuese un "timestamp without time zone".
10. Además de arrojar el error de tipado, la página entregada daba información acerca del funcionamiento al recibir la consulta. A partir de esto, se pudo determinar que la página solamente podía recibir "date" y "gatito_id", y, por el contrario, mostraría la misma página correspondiente a la URL "https://t2p1.cc5325.hackerlab.cl/gatitos".
11. Luego de revisar con mayor detalle, se determinó que el valor de "date" solo podía haber sido "year/month/day", puesto que este era simplemente usado en una consulta SQL, donde "date" sería el tipo solicitado de un atributo de tipo "timestamp".
12. A partir de esta información, se decidió hacer un ataque de inyección SQL para poder conseguir más información de la base de datos de donde se sacaban los gatos. Para esto, se hizo la siguiente consulta para comprobar que se podía usar inyección: "https://t2p1.cc5325.hackerlab.cl/gatitos/?gatito_id=0&date=year'+FROM+"gatitos_gatito"."start_datetime"+AT+TIME+ZONE+'Asia/Tokyo'))+AND+1=1+--". Donde la consulta tenía que cumplir con el código SQL que viniese después del parámetro "date", para que así funcionase con normalidad la consulta.
    15.Teniendo ya la certeza de que la inyección era posible y funcionaba, se optó por buscar en la web acerca de consultas que pudiesen entregar el nombre de todas las tablas de la base de datos. Luego de varios intentos y varios fallos, se decidió probar con un nombre particular de tabla.
13. Tomando en consideración el mensaje comentado en el código HTML de la página inicial, se decidió probar con el nombre de tabla "gatitos_secreto", realizando la siguiente consulta: "https://t2p1.cc5325.hackerlab.cl/gatitos/?gatito_id=0&date=year'+FROM+"gatitos_gatito"."start_datetime"+AT+TIME+ZONE+'Asia/Tokyo'))+AND+1=0+UNION+SELECT+*+FROM+gatitos_secreto+--".
14. A partir de esta consulta, se obtuvo una lista con todos los "gatitos secretos" que, a diferencia del resto de los gatitos, estos tenían "secret_message" no vacío. En particular, todos los "secret_message" eran mensajes cifrados en base64.
    18.Luego, se consideró que la flag debía estar entre alguno de los mensajes cifrados, por lo que al saber que todas las flags empiezan con "cc5325", se cifró este mensaje en base64, obteniendo "Y2M1MzI1", y se buscó, usando el buscador del navegador (Ctrl + F), algún mensaje que tuviese el texto solicitado.

15. Finalmente, se encontró el mensaje cifrado, siendo este "Y2M1MzI1XHtINHBQeV9jNHRfMTVfNTRkXH0=", que luego de ser descifrado nos arroja la flag **cc5325{H4pPy_c4t_15_54d}**.