Hispasec]

Writeup UAM

Mayo 2023

Informe técnico

Documentación

Índice

1. Datos del reto.

1.1. Especificaciones técnicas.

A continuación se detalla el alcance y el conjunto de especificaciones del reto, así como las normativas de aplicación.

Alcance	
Activo/s:	**Archivo .pcapng y 167.235.132.30**
Categoría:	Misc.
Dificultad:	Intermedio-Bajo
Flag 1:	UAM{L3cTur4_D3_U$B_Pc4P!!}
Flag 2:	UAM{SQL1_1NT0_SST1_1NT0_RC3_F3l1C1D4D3S!}
Cumplimiento normativo:	Las contraseñas y otra información sobre los usuarios no son almacenadas en cumplimiento con la ley de protección de datos (Reglamento General de Protección de Datos, RGPD). Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.

2. Proceso de resolución del reto.

A continuación se detalla la manera intencionada en la que se espera que el usuario resuelva el reto. Cada entrada viene encabezada por la descripción detallada del punto en concreto, acompañado de los pasos a seguir para reproducirlo.

2.1. Michelangelo (I) (flag 1).

Recientemente, una empresa ha descubierto que se han filtrado mensajes confidenciales de una conversación en la que se discutían planes estratégicos y se compartía información sensible. Según las investigaciones, se cree que un atacante ha obtenido acceso no autorizado a la conversación y ha robado los datos.

Sería peligroso que esta información acabe en manos de cualquier persona que sepa interpretarla…

Servidor de producción: 167.235.132.30

Formato de la flag: UAM{}

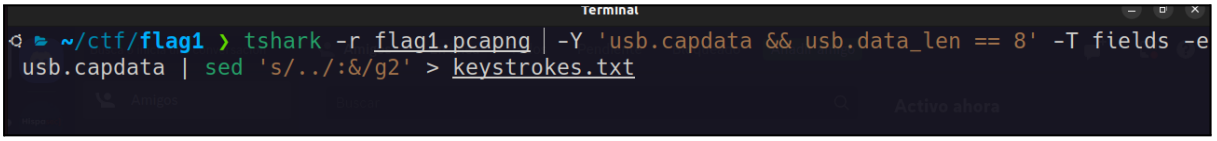
2.1.1. Resolución.

Lectura de un archivo **.pcapng** en el cual se tendrá que sacar texto en formato **ASCII** del propio archivo, el cual será "data" en un principio.

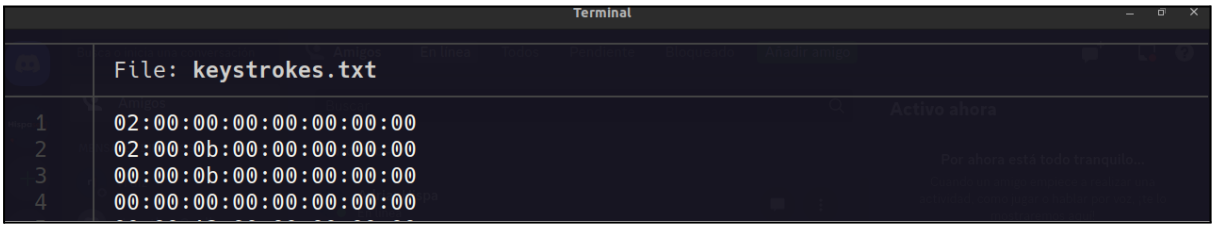
La forma que tendrá el archivo **.pcapng** será la siguiente:

Se deberá sacar la información (o data) válida para la lectura del texto en formato ASCII, para ello podemos emplear el siguiente comando:

- **tshark -r captura.pcapng -Y 'usb.capdata && usb.data_len == 8' -T fields -e usb.capdata | sed 's/../:&/g2' > keystrokes.txt**



Con el comando anterior conseguiremos filtrar por la data del archivo **.pcapng** que tenga 8 caracteres de longitud, dando como resultado un archivo **.txt** con el siguiente formato:



Una vez hemos conseguido esta data, podemos utilizar la herramienta [ctf-usb-keyboard-parser] para convertir la información en formato hexadecimal a formato de texto plano.

De esta forma, y conociendo la dirección IP del servidor de producción, llegaremos a una ruta de una página web donde aparece un texto de confirmación asegurando que hemos obtenido la *flag* del primer reto.

2.2. Michelangelo (II) (flag 2).

En estos días, los jóvenes aprendices de programación se encuentran ansiosos por responder a su llamada en el mundo de la tecnología. Sin embargo, el sabio consejo del Maestro Splinter resuena en sus mentes, recordándoles que aún tienen mucho por aprender antes de enfrentarse a los desafíos del mundo digital. Aunque están emocionados por poner en práctica sus habilidades y conocimientos, comprenden la importancia de completar su entrenamiento y pulir sus habilidades para enfrentarse con éxito a los obstáculos que se les presenten en su camino. Saben que la paciencia, la dedicación y el esfuerzo son clave para convertirse en verdaderos maestros en el arte de la programación.

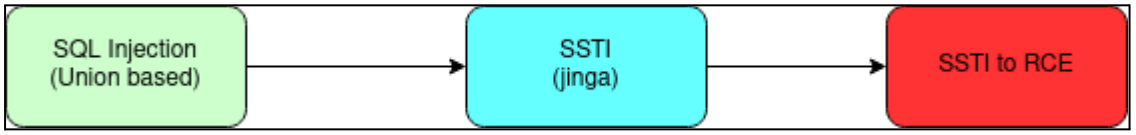
"Sé que están ansiosos de responder a su llamado, pero no han completado su entrenamiento" ~ Maestro Splinter

Formato de la flag: UAM{}

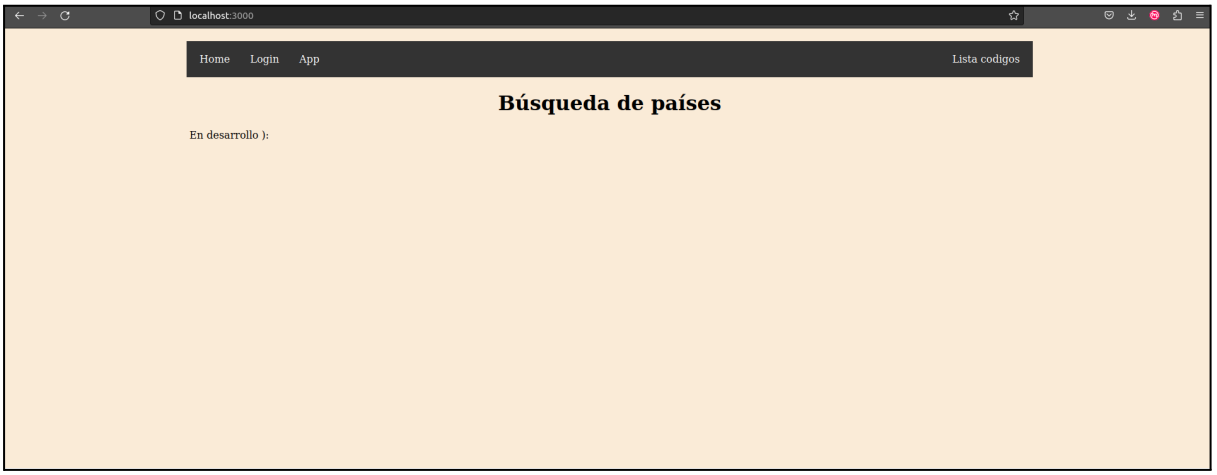
2.2.1. Resolución.

El objetivo de este reto consiste en: Empleo de **SQL** *Injection union based*, ligada a **SSTI (***Server Side Template Injection***)**. Para conseguirlo, el desarrollo de la página web está realizado empleando la librería (o framework) de python Jinja2 para la renderización de las plantillas (o templates), desde el cual podemos ejecutar código en la máquina de forma remota y visualizar el contenido de la contraseña del usuario administrador. Tras esto podemos iniciar sesión como dicho usuario obteniendo automáticamente el resultado del segundo reto o flag 2.

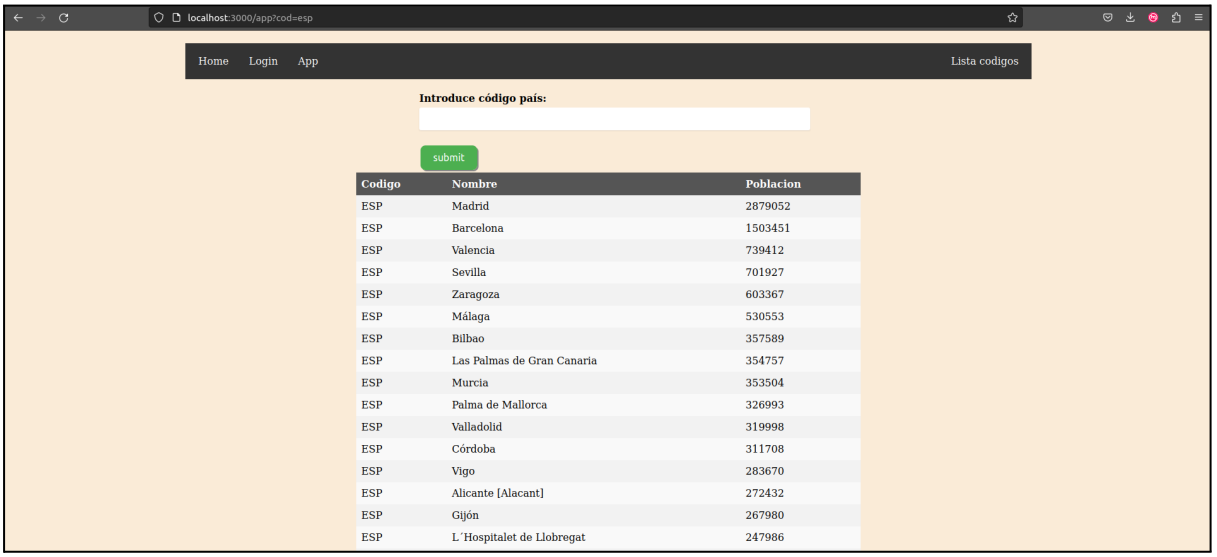
La ejecución se vería de la siguiente manera:



En la página web se visualizará lo siguiente:



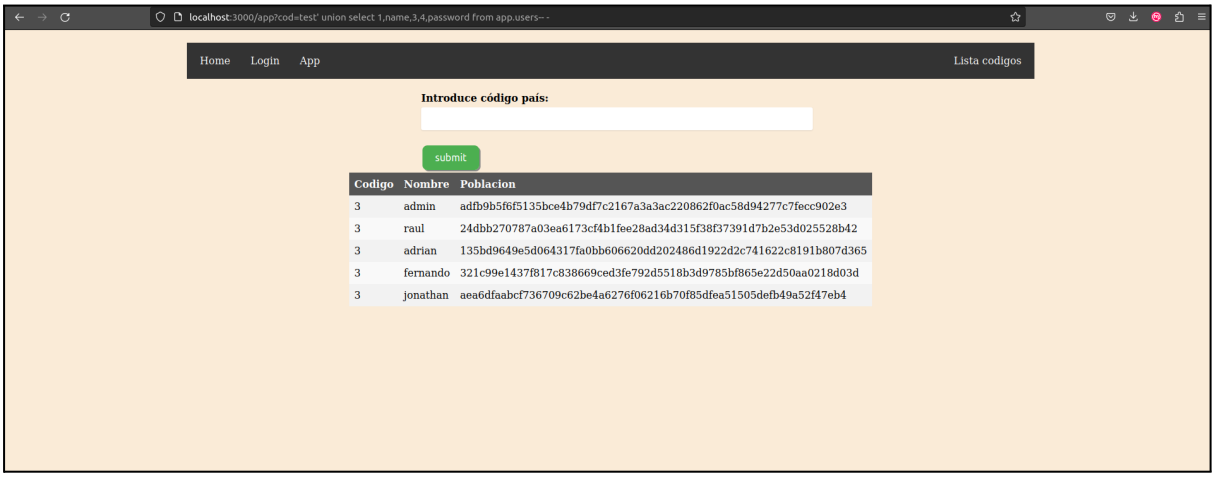
En ella se pueden realizar consultas para recibir la información almacenada en la base de datos en función de un código de país (e.g **ESP: España**).



Codigo	Nombre	Poblacion
ESP	Madrid	2879052
ESP	Barcelona	1503451
ESP	Valencia	739412
ESP	Sevilla	701927
ESP	Zaragoza	603367
ESP	Málaga	530553
ESP	Bilbao	357589
ESP	Las Palmas de Gran Canaria	354757
ESP	Murcia	353504
ESP	Palma de Mallorca	326993
ESP	Valladolid	319998
ESP	Córdoba	311708
ESP	Vigo	283670
ESP	Alicante [Alacant]	272432
ESP	Gijón	267980
ESP	L´Hospitalet de Llobregat	247986

Esta aplicación es vulnerable a inyección SQL, por lo que podremos extraer cualquier tipo de dato de ella.

Con la siguiente consulta obtenemos los usuarios y contraseñas cifradas (o hashes) de la aplicación.

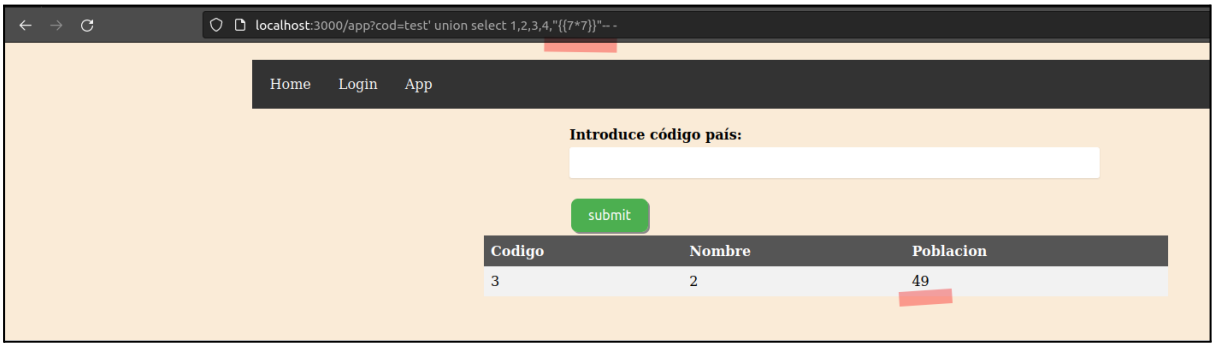


Codigo	Nombre	Poblacion
3	admin	adfb9b5f6f5135bce4b79df7c2167a3a3ac220862f0ac58d94277c7fecc902e3
3	raul	24dbb270787a03ea6173cf4b1fee28ad34d315f38f37391d7b2e53d025528b42
3	adrian	135bd9649e5d064317fa0bb606620dd202486d1922d2c741622c8191b807d365
3	fernando	321c99e1437f817c838669ced3fe792d5518b3d9785bf865e22d50aa0218d03d
3	jonathan	aea6dfaabcf736709c62be4a6276f06216b70f85dfea51505defb49a52f47eb4

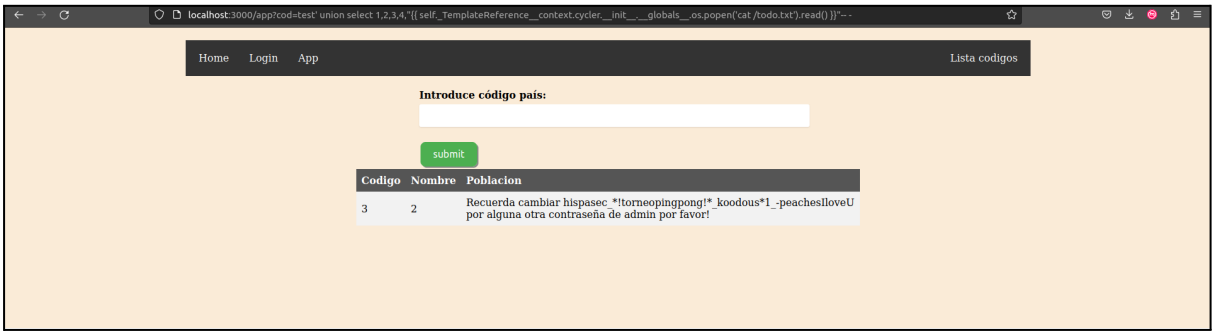
Al decodificar los hashes de los usuarios que no son "**admin**", obtendremos los siguientes resultados:

adfb9b5f6f5135bce4b79df7c2167a3a3ac220862f0ac58d94277c7fecc902e3	Unknown	Not found.
24dbb270787a03ea6173cf4b1fee28ad34d315f38f37391d7b2e53d025528b42	sha256	sstime
135bd9649e5d064317fa0bb606620dd202486d1922d2c741622c8191b807d365	sha256	sstiffanyss
321c99e1437f817c838669ced3fe792d5518b3d9785bf865e22d50aa0218d03d	sha256	ghinassti
aea6dfaabcf736709c62be4a6276f06216b70f85dfea51505defb49a52f47eb4	sha256	lamissti

Dada esta pista, comenzamos los intentos de inyección de código para vulnerar **Jinja2** en la inyección SQL.



Como podemos ver, la aplicación también es vulnerable a **SSTI**, por lo que tendremos ejecución remota de comandos en la aplicación web.



Con esto podremos leer el archivo **todo.txt**, situado en la raíz del sistema, del cual conseguimos la contraseña del usuario "**admin**" y podremos iniciar sesión.

Una vez llegados a este punto, el segundo reto es resuelto obteniendo el mensaje que nos lo confirma, es decir, la segunda *flag*.

Hispasec]

Hispasec Sistemas S.L.
C/ Severo Ochoa, 10 - 29590, Málaga
Telf: (+34) 952 020 494
info@hispasec.com