REPORTE DE LA VULNERABILIDAD ENCONTRADA
EN EL SITIO WEB

[redacted]

22 DE MAYO DE 2018

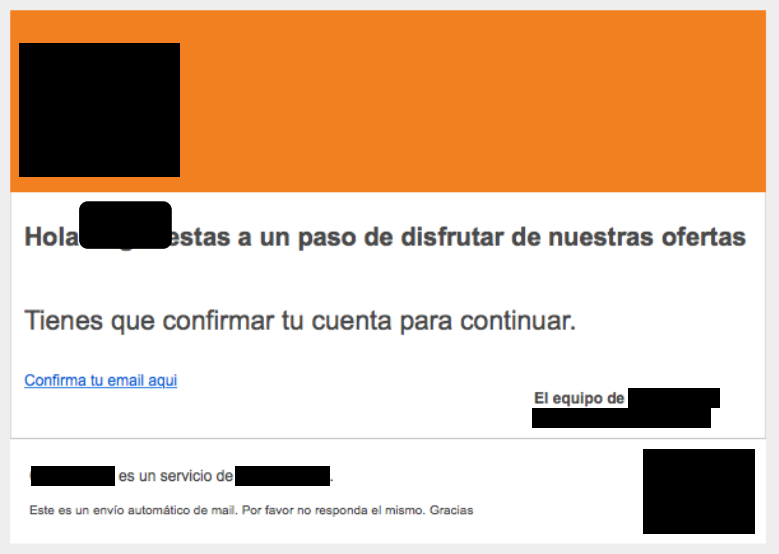
POR EDGAR OSORNIO VELÁZQUEZ
[redacted]

DESCRIPCIÓN DE LA VULNERABILIDAD

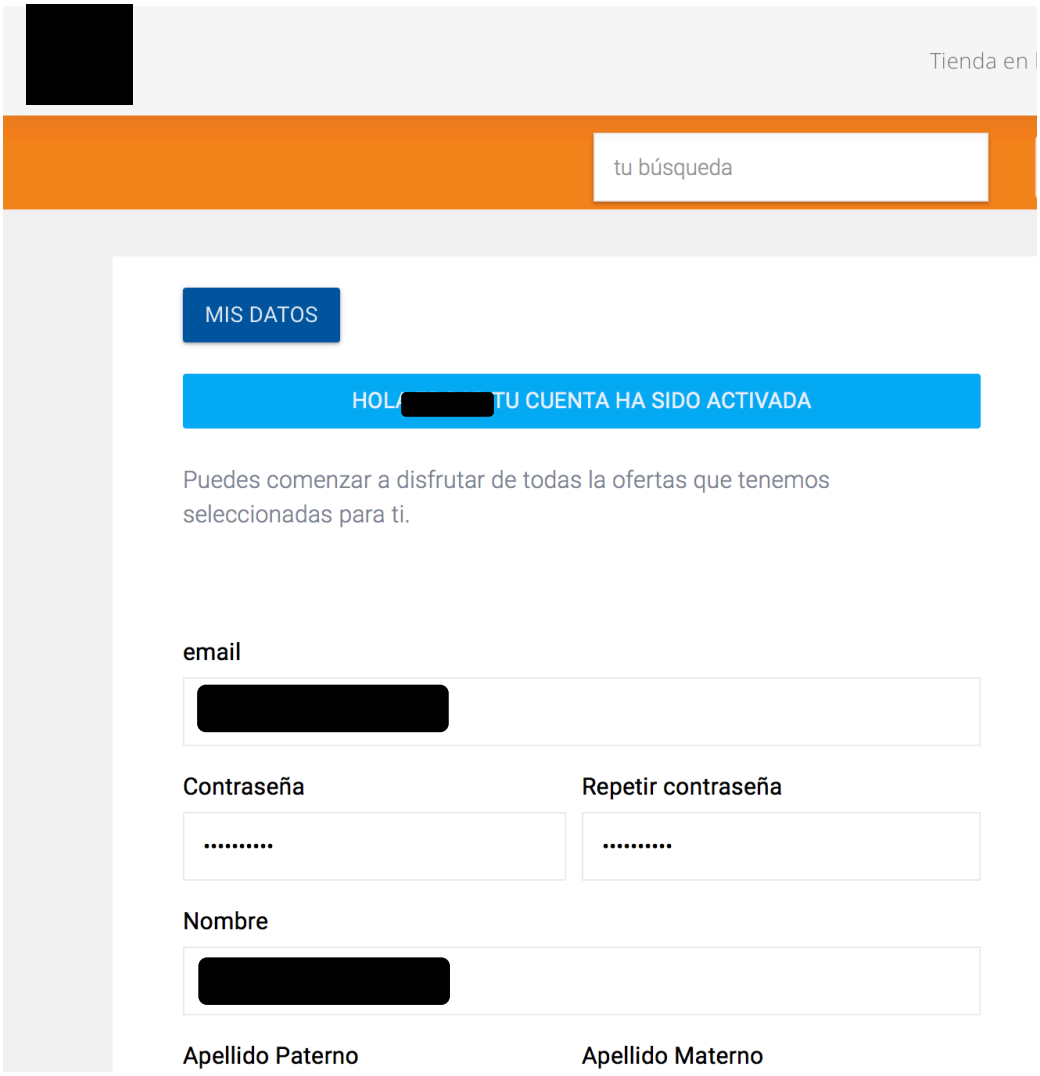
En el mes de mayo de 2018, una vulnerabilidad muy delicada fue descubierta en la página [redacted], de la empresa [redacted]. Específicamente, fue encontrada en la variante del sistema del centro comercial [redacted]. Aún así, al parecer el fallo se encuentra tanto en el sistema principal, como en la de [redacted].

La vulnerabilidad expone la información de los usuarios del sistema, incluido su correo electrónico y la contraseña. Esto es delicado, ya que lamentablemente hay mucha gente que suele utilizar una misma contraseña para diversas páginas web. Si la contraseña que registran en el sistema es la misma a la de su correo, las posibilidades son infinitas, pues los correos electrónicos son frecuentemente utilizados para realizar "reseteos" de contraseñas de muchas servicios de internet.

La vulnerabilidad se encuentra en la verificación de las cuentas de usuarios. Una vez que un usuario se registra en la plataforma, se le es enviado un correo electrónico, en el que, mediante un enlace, se le pide verificar su email.



Una vez que el usuario hace clic en el enlace, es redireccionado directamente a su perfil, en dónde se despliega toda su información. Aquí se encuentra el primer error. El estándar en un sistema donde se maneje una confirmación de correo electrónico es que, cuando se de clic en el enlace, simplemente se despliegue una vista en la que se le informe al usuario que su cuenta fue verificada correctamente, y que ya le es posible iniciar sesión, sin accesar inmediatamente a su perfil (OWASP, 2017).



El enlace utilizado para la verificación del correo es como el siguiente:

http://[redacted].com/registration.php?token=708be71b9ab6e0a84252760579ade9f1 (Enlace ficticio).

A simple vista, cualquier informático se puede dar cuenta que el parámetro "token" enviado se trata de una cadena de caracteres encriptada con la función MD5. Solo basta escribir un programa de desencriptación, o hasta utilizar una página en internet para desencriptar dicho token, y darse cuenta que se trata de un número, siendo este el ID del usuario en la base de datos.

Con esta información, y teniendo en cuenta que lo más usual en un sistema es tener los IDs de los usuarios de manera autoincrementable, solo basta

con encriptar una serie de números, desde el 0 hasta algún otro número deseado, para obtener el token del usuario con dicho ID. En adición, se puede escribir un programa para realizar scrapping en la página. Como cada enlace redirigiría al perfil de cada usuario en donde se despliega su información, se pueden obtener todos los datos de cada persona registrada.

Para solucionar este problema, bastaría con cambiar el algoritmo de encriptación a uno que permita integrar un "salt", para que así no sea tan fácil generar los token de validación de los usuarios (Van der Stock & Cuthbert, 2015). Mejor aún, se podría cambiar el algoritmo de encriptado a uno de hasheado. La diferencia entre un hasheo y una encriptación es que la encriptación sí permite regresar la cadena de caracteres generada al texto original, mientras que un hasheo no lo permite (Security Innovation Europe, 2016).

Por último, es posible la obtención de las contraseñas de los usuarios, porque están escritas en texto plano dentro del HTML. Esto me hace pensar en dos cosas. Una de ellas es que, las contraseñas las guarden en su base de datos encriptadas, y no hasheadas. O peor aún, que se estén guardando en texto plano. Los estándares más actuales dictan que lo más seguro es utilizar funciones de hasheo para las constraseñas (Steven, Manico, & Righetto, 2018). De esta forma, si un atacante obtiene acceso a su base de datos, le será imposible obtener los valores originales de las contraseñas de los usuarios.

En este caso, mi recomendación sería cambiar a un algoritmo de hasheado fuerte que trabaje con un "salt", como lo podría ser un SHA512. Así, cada vez que se inicie sesión, la contraseña se tendrá que hashear del lado del cliente, para que cuando llegue al servidor, este la compare con el hash que esté registrado como contraseña en la base de datos.

AVISO LEGAL

La propuesta aquí descrita es meramente una sugerencia. Yo, Edgar Osornio Velázquez me deslindo de cualquier resultado que pueda tener la implementación de la misma, pues aún con las mejoras propuestas el sistema puede seguir estando vulnerable.

BIBLIOGRAFÍA

OWASP. (2017). OWASP Top 10 - The Ten Most Critical Web Application Security Risks. *Owasp*, 22. Recuperado a partir de https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf%0Ahttps://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf%0Ahttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:OWASP+Top+10+-2010#1

Security Innovation Europe. (2016). WHAT IS THE DIFFERENCE BETWEEN HASHING AND ENCRYPTING.

Steven, J., Manico, J., & Righetto, D. (2018). Password Storage Cheat Sheet. Recuperado a partir de https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet

Van der Stock, A., & Cuthbert, D. (2015). Application Security Verification Standard. *Owasp*, (3.0), 70. https://doi.org/https://www.owasp.org/images/3/33/OWASP_Application_Security_Verification_Standard_3.0.1.pdf