Documentación - Proyecto Final

Eric Fernando Torres Rodriguez A01700249
Martín Adrian Noboa Monar A01704052
Samuel Octavio González Azpeitia A01704696

Seguridad Informática
20/11/2021

Índice:

Introducción

La seguridad _____ , es por ello que los sistemas de doble autenticación se han vuelto más comunes en todos los sistemas que utilizamos actualmente, es por esto que para el proyecto final de la materia de seguridad informática hemos desarrollado un proyecto el cual consiste en la creación de una página web la cual cuenta con inicio de sesión con doble autenticación. En el siguiente documento se abordará con detalle el desarrollo de este proyecto al igual que las guías de usuario para el funcionamiento de este.

Solución

La solución implementada tiene dos partes , el desarrollo de la página web se realizó en php con el framework de laravel donde para el primer paso de autenticación se utiliza una contraseña cifrada con hash.

Bcrypt

Laravel cuenta con la función Hash incluida la cual utiliza el algoritmo de cifrado bcrypt, desarrollado por Niels Provos and David Mazières, este algoritmo de hash está basado sobre el cifrado Blowfish que utiliza “sal” y es computacionalmente adaptable, es decir que se puede configurar el algoritmo para compensar hardware más rapido, haciendo que la función corra más lento.

Bcrypt fue diseñado para el hash de contraseñas, por lo tanto, es un algoritmo lento. Esto es bueno para el hash de contraseñas, ya que reduce el número de contraseñas por segundo que un atacante podría usar al crear un ataque de diccionario.

Durante su diseño los desarrolladores de esta algoritmo utilizaron la costosa fase de configuración de claves del cifrado Blowfish para desarrollar un nuevo algoritmo de configuración de claves para Blowfish llamado "eksblowfish", que significa "programación clave costosa Blowfish".

Una cadena hash bcrypt tiene la forma:

[algoritmo] [costo] $ [22 caracteres sal] [31 caracteres hash]

Por ejemplo:

$ 2a $ 10 $ N9qo8uLOickgx2ZMRZoMyeIjZAgcfl7p92ldGxad68LJZdL17lhWy

- **$ 2a $:** el identificador del algoritmo hash (bcrypt)
- **10:** factor de costo (210, es decir, 1024 rondas)
- **N9qo8uLOickgx2ZMRZoMye:** salt de 16 bytes (128 bits), Radix-64 codificado como 22 caracteres
- **IjZAgcfl7p92ldGxad68LJZdL17lhWy:** hash de 24 bytes (192 bits), Radix-64 codificado como 31 caracteres

El algoritmo de Bcrypt funciona en dos principales fases:

- Fase 1:

Una función llamada EksBlowfishSetup se configura usando el “costo computacional” deseado, la sal (caracteres extras) y la contraseña para inicializar el estado de eksblowfish. Luego, bcrypt ejecuta una programación de claves costosa que consiste en realizar una derivación de clave principal, donde obtenemos un conjunto de subclaves. Aquí, la contraseña se utiliza como clave principal. En caso de que el usuario haya seleccionado una contraseña incorrecta o corta, convertimos esa contraseña / clave en una contraseña / clave más larga. La práctica antes mencionada también se conoce como estiramiento de teclas.

Lo que estamos atravesando en esta primera fase es promover el fortalecimiento de claves para ralentizar los cálculos, lo que a su vez también ralentiza a los atacantes.

- Fase 2:

El valor mágico es el valor “OrpheanBeholderScryDoubt” de 192 bits. Este valor se cifra 64 veces utilizando eksblowfish en modo ECB>) con el estado de la fase anterior. El resultado de esta fase es el costo y el valor de sal de 128 bits concatenados con el resultado del ciclo de cifrado.

Fortify

Para poder hacer el inicio de sesión con dos factores nos apoyamos de la herramienta Fortify, es una implementación de backend de autenticación agnóstica de frontend para Laravel. Fortify registra las rutas y los controladores necesarios para implementar todas las funciones de autenticación de Laravel, incluido el inicio de sesión, el registro, el restablecimiento de contraseña, la verificación de correo electrónico y más.

Elegimos hacerlo por medio de códigos QR, para esto se debe crear una pantalla dentro de su aplicación donde los usuarios puedan administrar sus configuraciones de autenticación de dos factores. Esta pantalla debe permitir al usuario habilitar y deshabilitar la autenticación de dos factores, así como volver a generar sus códigos de recuperación de autenticación de dos factores.

Para habilitar la autenticación de dos factores, la aplicación realiza una solicitud POST al /user/two-factor-authentication punto final definido por Fortify. Cuando tiene éxito la solicitud, el usuario será redirigido a la URL anterior con esto el status variable de sesión se establece en two-factor-authentication-enabled. Puede detectar este status variable de sesión dentro de sus plantillas para mostrar el mensaje de éxito apropiado.

A continuación, debe mostrar el código QR de autenticación de dos factores para que el usuario lo escanee en su aplicación de autenticación. Si está utilizando Blade para renderizar la interfaz de su aplicación, puede recuperar el código QR SVG usando el `two Factor QrCode Svg` método disponible en la instancia del usuario.

Para comenzar a implementar la funcionalidad de autenticación de dos factores, debemos indicarle a Fortify cómo devolver nuestra vista del desafío de autenticación de dos factores. Toda la lógica de representación de la vista de autenticación de Fortify se puede personalizar utilizando los métodos apropiados disponibles a través de la `Laravel\Fortify\Fortify` Clase. Por lo general, debe llamar a este método desde el `boot` método de la `App\Providers\Fortify ServiceProvider` clase de su aplicación.

Instalación

La guía de instalación se encuentra en el repositorio del proyecto:

https://github.com/MartinNoboa/ProyectoCiberseguridad

Referencias:

Arias D (25-Feb-2021) Auth0.Hashing in Action: Understanding bcrypt. Recuperado de: https://auth0.com/blog/hashing-in-action-understanding-bcrypt/

Laravel(S.F) Laravel.Hashing.REcuperado de: https://laravel.com/docs/8.x/hashing

Sanchez O. (2-Aug-2016). Medium. Programador: ¿cómo guardas tus contraseñas?. Recuperado de: https://medium.com/universo-mutante/si-guardas-contrase%C3%B1as-as%C3%AD-pones-en-peligro-a-tus-usuarios-e567cc2e1ec5

Anonimo (S.F).hmong. Bcryp. Recuperado de: thttps://hmong.es/wiki/Bcrypt