

Nicolas Diaz Guzmán

Juan Asenjo Segovia

Francisca Sepúlveda Thomassen

Fritzon Brignolle

Boris Sepúlveda González

Programación Segura

Evaluación sumativa 1

Profesora Pamela Acevedo Bonelli

***Índice***

Vulnerabilidades y Soluciones2

Código que puede ser modificado por atacantes6

Modelo de amenazas7

Evidencias del uso de la herramienta GIT en GITHUB8

***Vulnerabilidades y soluciones***

Problema:

Hace falta el encabezado X-Frame-Options para mejorar la seguridad, ya que sin esto algún usuario malicioso podría usar ClickJacking para ingresar algún elemento y otro usuario al hacer clic podría descargar algún malware.

Texto

Descripción generada automáticamente con confianza media

Solución:

Las políticas de seguridad de contenido según el cuadro de accesorios que nos permite identificar a través de este (deny, sameorigin, allow-from) modificando estas opciones podríamos hacer el sitio web mas seguro.

Problema:

No se encontraron tokens Anti-CSRF en un formulario de envío HTML.

Una solicitud falsa entre sitios en un ataque que compromete y obliga a una víctima a enviar su solicitud HTTP a un destino objetivo sin su conocimiento o intención de poder realizar una acción como víctima

El atacante crea una cuenta de host en el dominio de confianza para tener acceso a todos los datos o metadatos de la victima

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Solución:

Utilizar una biblioteca o marco comprobado hace que esta vulnerabilidad sea más fácil de evitar, un ejemplo de esto sería el paquete anti-CSRG.

Problema: El encabezado de control de caché no se ha configurado correctamente o falta, lo que permite que el navegador y los proxies almacenen contenido en caché.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Solución: Siempre que sea posible, asegúrese de que el encabezado HTTP de control de caché esté configurado sin caché, sin almacenamiento, debe revalidar.

Problema:

MIME Confusion Attack permite ataques a través de sitios de contenido generado por el usuario al permitir que los usuarios carguen código malicioso que luego es ejecutado por navegadores que interpretarán los archivos utilizando tipos de contenido alternativos, por ejemplo, implícitoapplication/javascriptfrente a explícitotext/plain. Esto puede resultar en un ataque de "descarga no autorizada", que es un vector de ataque común para el phishing. Los sitios que alojan contenido generado por el usuario deben utilizar este encabezado para proteger a sus usuarios. Esto es mencionado por VeraCode y OWASP.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Solución:

Asegúrese de que la aplicación / servidor web establezca el encabezado Content-Type de manera adecuada y que establezca el encabezado X-Content-Type-Options en 'nosniff' para todas las páginas web.

Si es posible, asegúrese de que el usuario final utilice un navegador web moderno y compatible con los estándares que no realice ningún rastreo de MIME, o que la aplicación web / servidor web pueda indicarle que no realice el rastreo de MIME

***Código que puede ser modificado por atacantes***



***Modelo de amenazas***

Imagen que contiene competencia de atletismo

Descripción generada automáticamente

Modelo de amenazas stride

Una de las primeras amenazas que identificamos es la vulneración de datos ya que con este se puede suplantar identidades de personas obteniendo datos a través de la consulta ya que esta esta mal programada a través de código php

Manipulación de datos

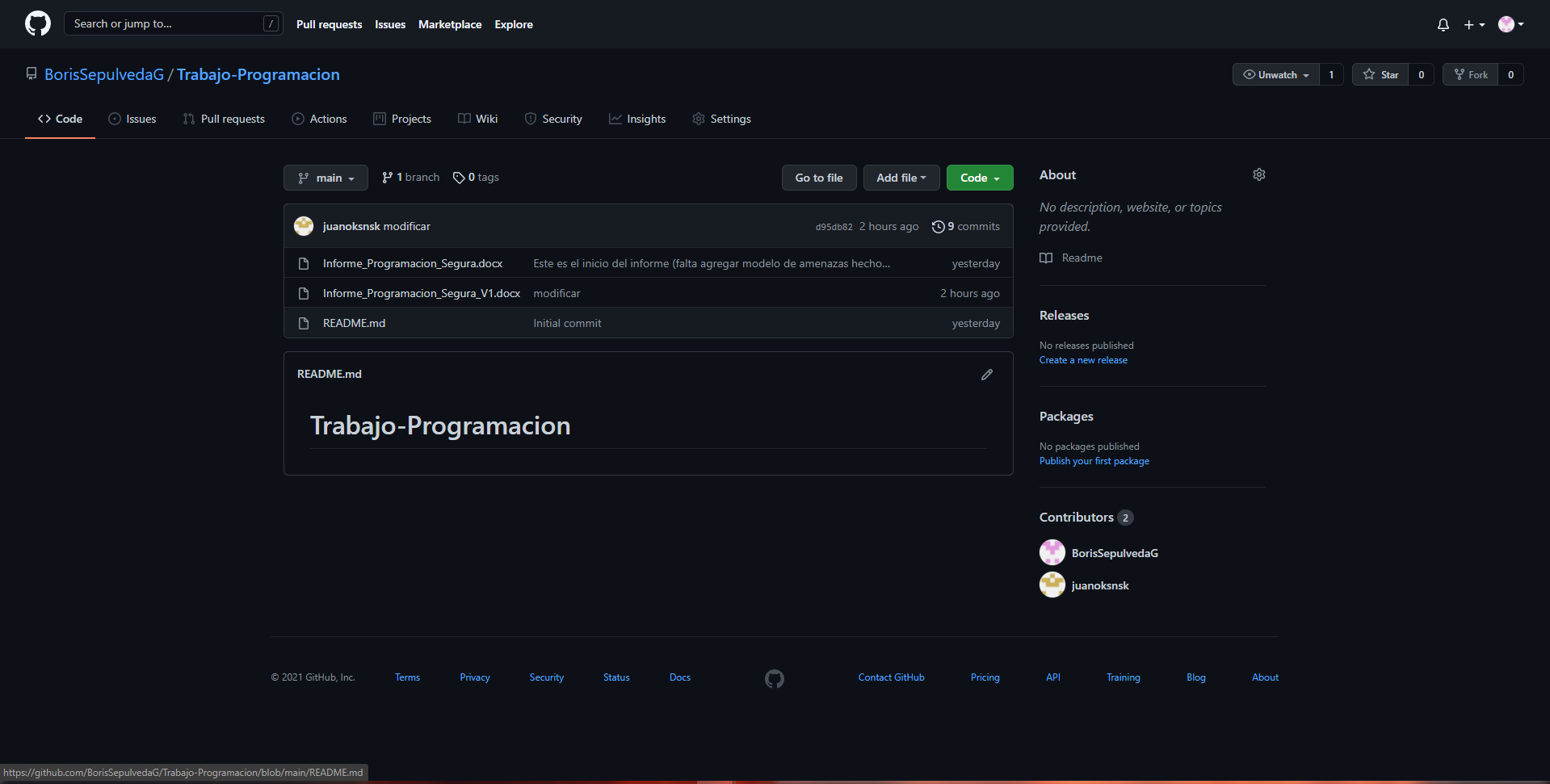
Dadas las circunstancias en esta pagina se pueden manipular datos de forma maliciosa ya que al consultar en la página nos muestra información primordial de cada usuario registrado.

Divulgación de información

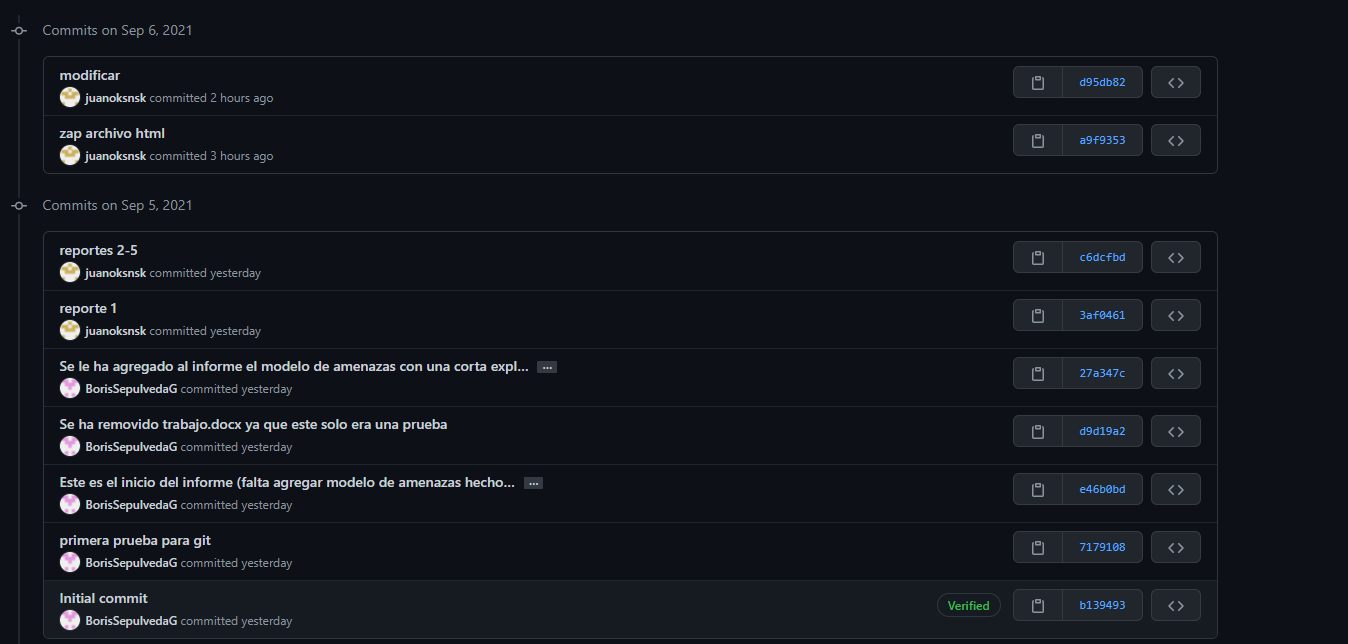
Es la exposición de datos sensibles sobre esta página, en la cual podemos ver la fuga de datos

***Evidencia del uso de la herramienta GIT en GITHUB***

Para el desarrollo en conjunto de este informe se utilizo la herramienta GIT en conjunto con la pagina web GITHUB ya que de esta manera los cambios que se realizaran en el informe quedarían guardados en versiones posteriores en caso de cometer alguna equivocación

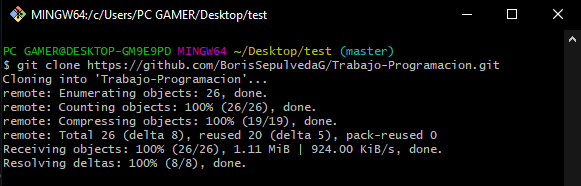


En total se registraron 9 commits en el historial de GITHUB



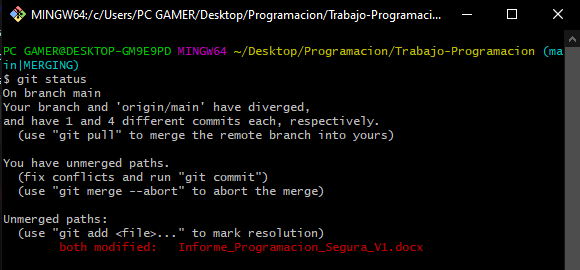
Todas estas versiones eran subidas mediante la herramienta GIT con los siguientes comandos

1. git clone <https://github.com/BorisSepulvedaG/Trabajo-Programacion.git>



Este comando era específicamente utilizado para copiar el repositorio de GITHUB en archivos locales

1. git status



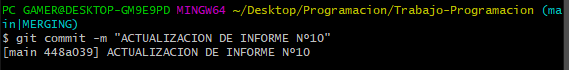
Este comando era utilizado para identificar los archivos modificados, creados o eliminados a los cuales no se les realizaba seguimiento o bien si se les realizaba pero estos habían sido modificados.

1. git add Informe\_Programacion\_Segura.docx o git add .



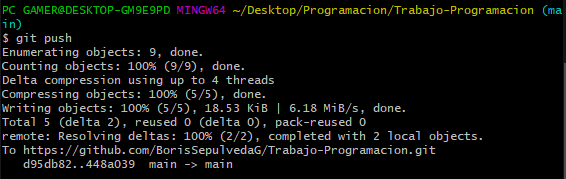
Con este comando seleccionábamos los archivos que serian actualizados en el proyecto de GITHUB

1. git commit -m “ACTUALIZACION DE INFORME Nº10”



Este comando era utilizado para realizar la actualización de los archivos de manera local

1. git push



Este comando es con el cual los cambios que realizábamos de manera local se veían reflejados dentro del proyecto en GITHUB