**Trabajo práctico N° 3**

Informe cuantos usuarios pueden ingresar al <https://timetix.net> y a la <https://app.timetix.net>.

En Jmeter ingreso petición: <https://timetix.net>

**Ejemplo 1**

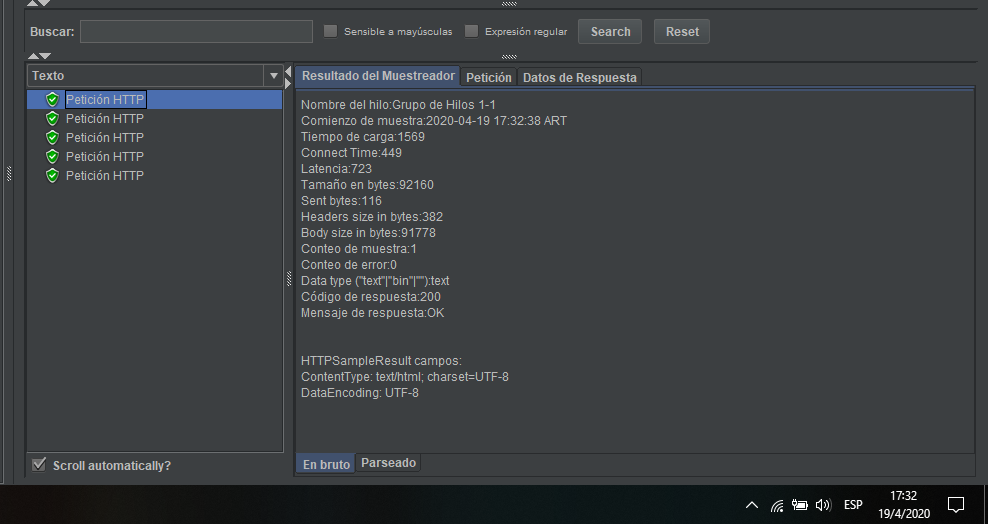
Número de hilos :5

Periodo de subida (en segundos):1

Contador del bucle:1

En “ver árbol de resultados”: Me muestra a los 5 usuarios con un escudo en verde, acepta a todos los usuarios ingresados, entre cada petición varía el nombre de hilos, tiempo de carga, connect time, latencia.

Carga todos los valores rápido.



**Ejemplo 2**

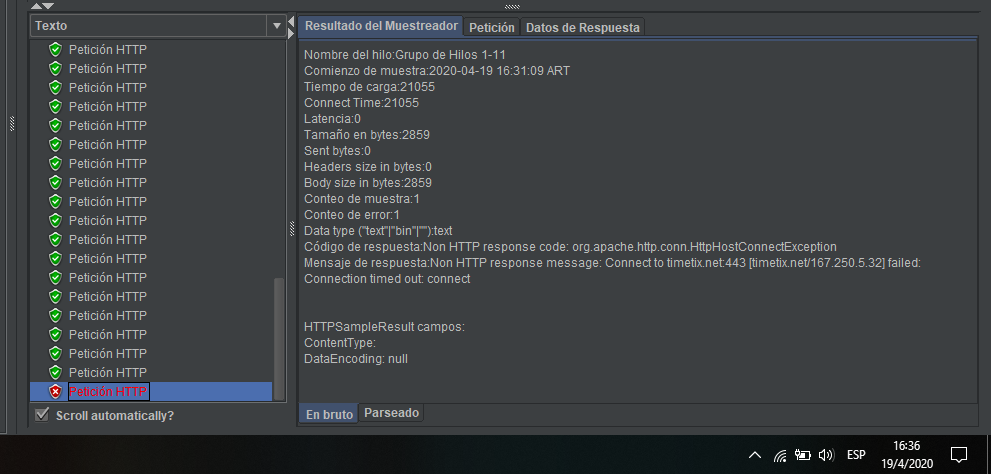
Número de hilos: 55

Periodo de subida (en segundos):3

Contador de bucle:1

Acepta todas las peticiones excepto la última. Varía entre las peticiones el resultado.

Carga todos los valores rápido.



**Ejemplo 3**

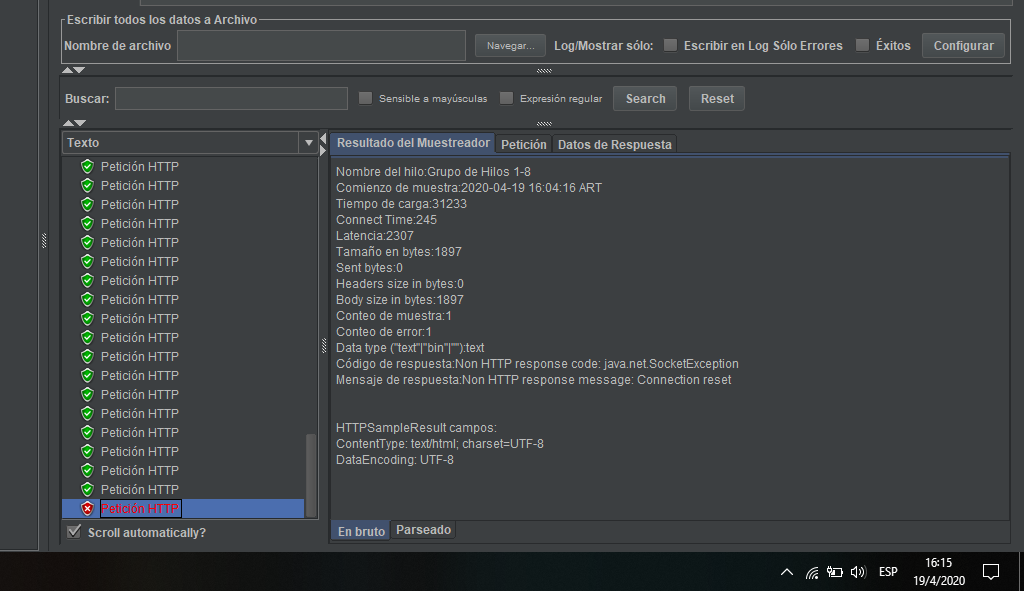
Número de hilos:80

Periodo de subida (en segundo):1

Contador de bucle:1

Acepta todas las peticiones excepto la última. Varía entre las peticiones el resultado.

Carga todas las peticiones.



**Ejemplo 4**

Número de hilos:200

Periodo de subida (en segundo):1

Contador de bucle:1

En “ver árbol de resultado”: Se puede observar que en las primeras peticiones se ven en verde, luego se van alternando entre verdes y rojas.

Tarda en cargar las peticiones.

|  |  |  |
| --- | --- | --- |
| **Resultado** | **Petición verde** | **Petición rojo** |
| Nombre de hilos | Varía el valor | Varía el valor |
| Comienzo de muestra | Mismo valor | Mismo valor |
| Tiempo de carga | Varía el valor | Varía el valor |
| Connect time | Varía el valor | Varía el valor |
| Latencia | Varía el valor | Varía el valor |
| Tamaño de bytes | Varía el valor | Varía el valor |
| Sent bytes | Valor 116 | Valor 0 |
| Headers size in bytes | Valor 382 | Valor 0 |
| Body size in bytes | Valor 91778 | Valor 1897 |
| Conteo de muestra | Mismo valor | Mismo valor |
| Conteo de error | Valor 0 | Valor 1 |
| Data type(“text”|”bin”|””) | Text | Text |
| Código de respuesta | 200 | non http response code:java.net SocketException |
| Mensaje de respuesta | Ok | non http response message:connection reset |

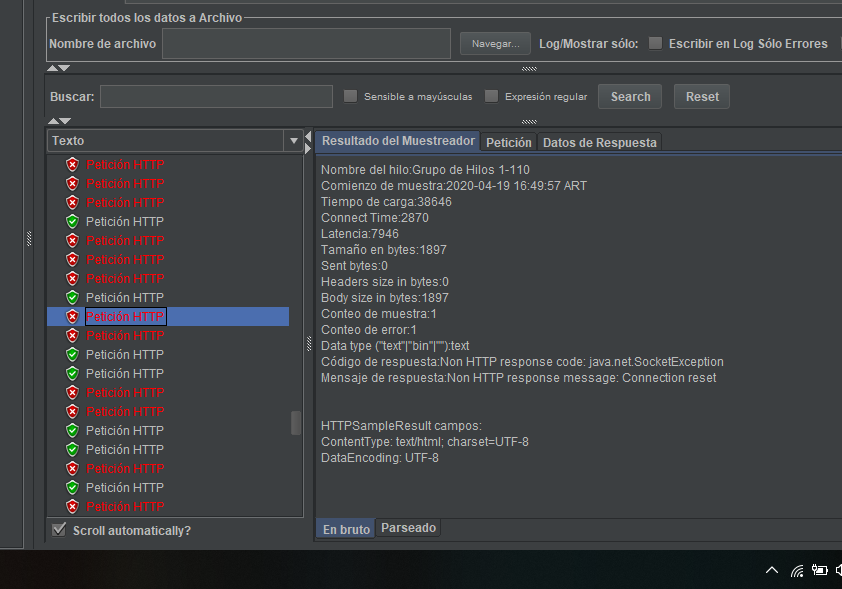
**Ejemplo 5**

Número de hilos: 1000

Periodo de subida (en segundos): 1

Contador de bucle: 1

Tarda un tiempo en dar los resultados, las primeras peticiones las muestra en color rojo luego muestra unas en verde se van alternando, varían los valores.



En Jmeter ingreso petición: <https://app.timetix.net>

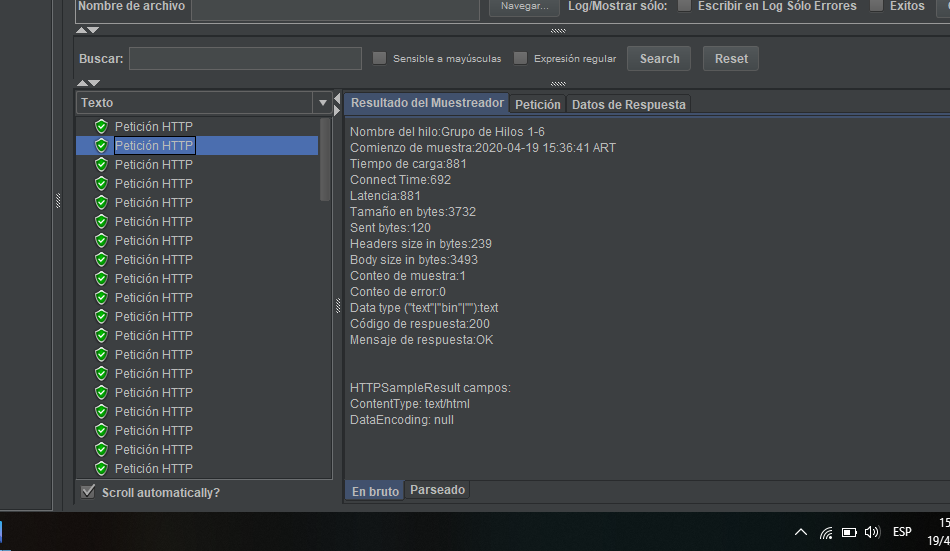
**Ejemplo: 1**

Número de hilos: 80

Periodo de subida (en segundos): 1

Contador de bucle: 1

En “ver árbol de resultados”: muestra las 80 peticiones en verde, entre cada petición varía el nombre de hilos, tiempo de carga, connect time, latencia, carga las peticiones rápido.



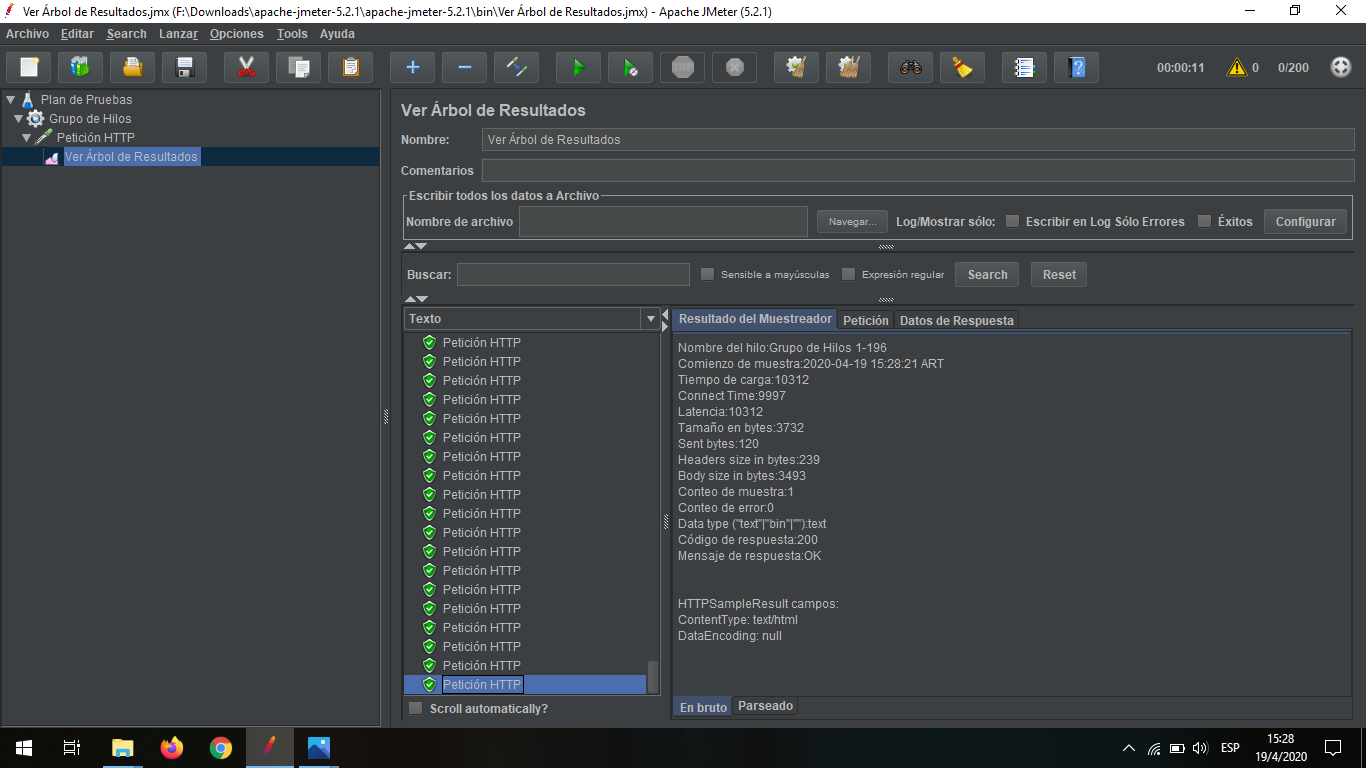
**Ejemplo: 2**

Número de hilos: 200

Periodo de subida (en segundos): 1

Contador de bucle: 1

En “ver árbol de resultados”: muestra las 200 peticiones en verde entre cada petición varia el nombre de hilos, tiempo de carga, connect time, latencia, tarda unos minutos en cargar las peticiones.



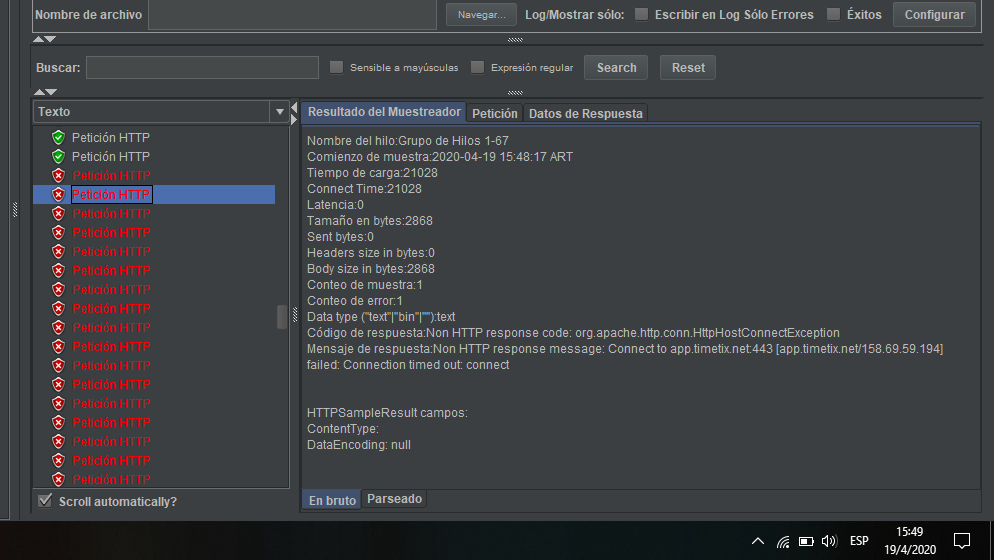
**Ejemplo: 3**

Número de hilos: 478

Periodo de subida (en segundos): 1

Contador de bucle: 1

Tarda en cargar las peticiones, las primeras peticiones salen en color verde luego todas rojas.



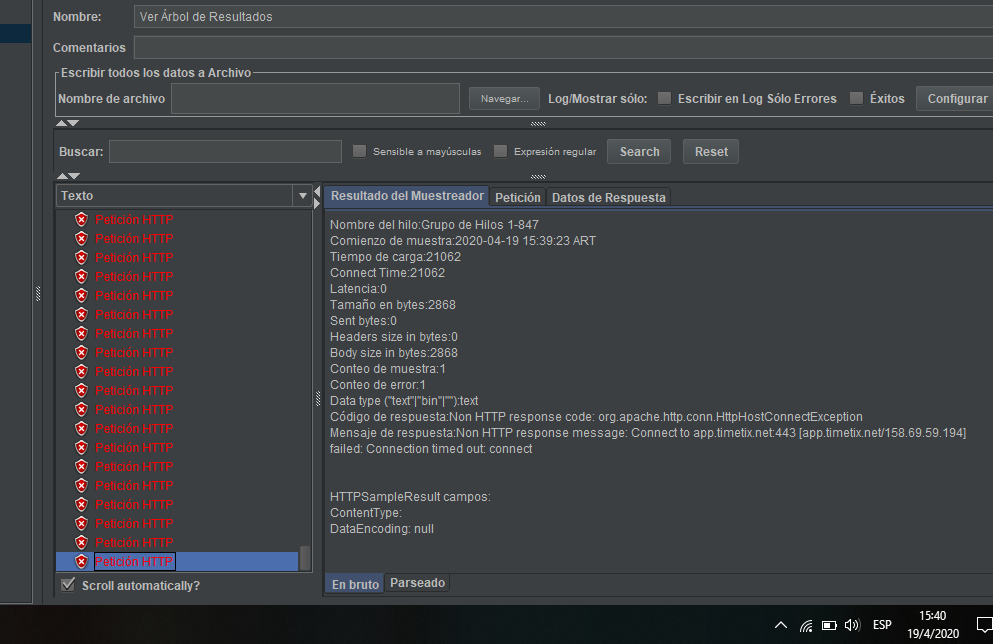
**Ejemplo: 4**

Número de hilos: 1000

Periodo de subida (en segundos): 1

Contador de bucle: 1

Tarda en mostrar las peticiones no se cargan todas se ven en rojo.



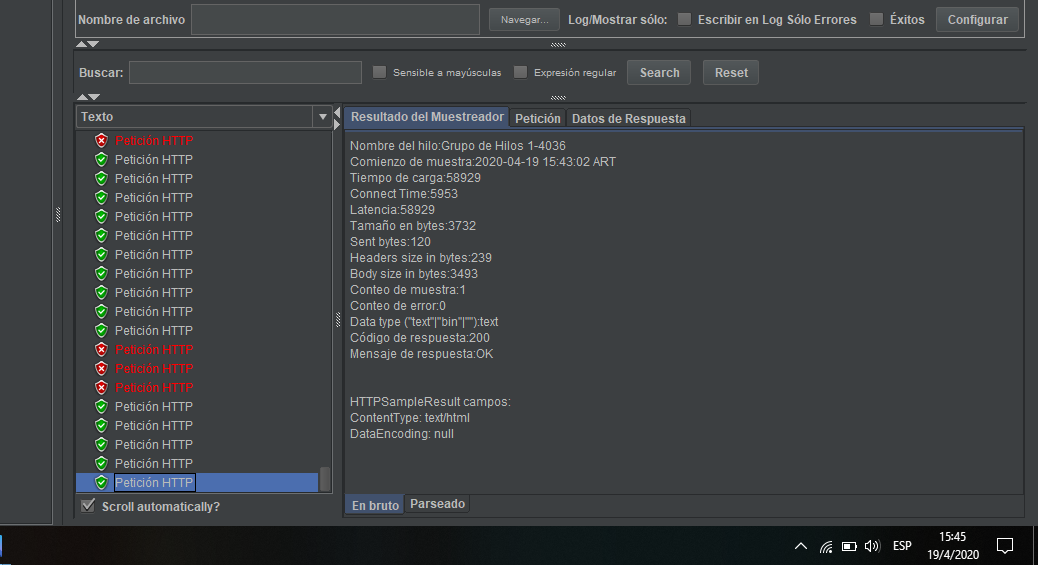
**Ejemplo: 5**

Número de hilos: 10000

Periodo de subida (en segundos): 1

Contador de bucle: 1

Tarda un tiempo en dar los resultados, las primeras peticiones las muestra en color rojo luego muestra unas en verde se van alternando, varían los valores.



**Los 10 principales riesgos de seguridad de OWASP son:**

1.Inyección: Las fallas de inyección, como SQL, NoSQL, OS o LDAP ocurren cuando se envían datos no confiables a un intérprete, como parte de un comando o consulta. Los datos dañinos del atacante pueden engañar al intérprete para que ejecute comandos involuntarios o acceda a los datos sin la debida autorización.

2.Pérdida de autenticación: Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son implementadas incorrectamente, permitiendo a los atacantes comprometer usuarios y contraseñas, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios (temporal o permanentemente).

3.Exposición de datos sensibles: Muchas aplicaciones web y APIs no protegen adecuadamente datos sensibles, tales como información financiera, de salud o Información Personalmente Identificable (PII). Los atacantes pueden robar o modificar estos datos protegidos inadecuadamente para llevar a cabo fraudes

con tarjetas de crédito, robos de identidad u otros delitos. Los datos sensibles requieren métodos de protección adicionales, como el cifrado en almacenamiento y tránsito.

4.Entidades externas XML(XXE): Muchos procesadores XML antiguos o mal configurados evalúan referencias a entidades externas en documentos XML. Las entidades externas pueden utilizarse para revelar archivos internos mediante la URI o archivos internos en servidores no actualizados, escanear puertos de

la LAN, ejecutar código de forma remota y realizar ataques de denegación de

servicio (DoS).

5.Pérdida de control de acceso: Las restricciones sobre lo que los usuarios autenticados pueden hacer no se aplican correctamente. Los atacantes pueden explotar estos defectos para acceder, de forma no autorizada, a funcionalidades y/o datos, cuentas de otros usuarios, ver archivos sensibles, modificar datos, cambiar derechos de acceso y permisos, etc.

6.Configuraciones de seguridad incorrecta: La configuración de seguridad incorrecta es un problema muy común y se debe en parte a establecer la configuración de forma manual, ad hoc o por omisión (o directamente por la falta de configuración). Son ejemplos: S3 buckets abiertos, cabeceras HTTP mal configuradas, mensajes de error con contenido sensible, falta de parches y actualizaciones, frameworks, dependencias y componentes desactualizados, etc.

7. Secuencia de comandos en sitios cruzados (XSS): Los XSS ocurren cuando una aplicación toma datos no confiables y los envía al navegador web

sin una validación y codificación apropiada; o actualiza una página web existente con datos suministrados por el usuario utilizando una API que ejecuta JavaScript en el navegador. Permiten ejecutar comandos en el navegador de la víctima y el atacante puede secuestrar una sesión, modificar (defacement) los sitios web, o re direccionar al usuario hacia un sitio malicioso.

8.Deserialización insegura: Estos defectos ocurren cuando una aplicación recibe objetos serializados dañinos y estos objetos pueden ser manipulados o borrados por el atacante para realizar ataques de repetición, inyecciones o elevar sus privilegios de ejecución. En el peor de los casos, la deserialización insegura puede conducir a la ejecución remota de código en el servidor.

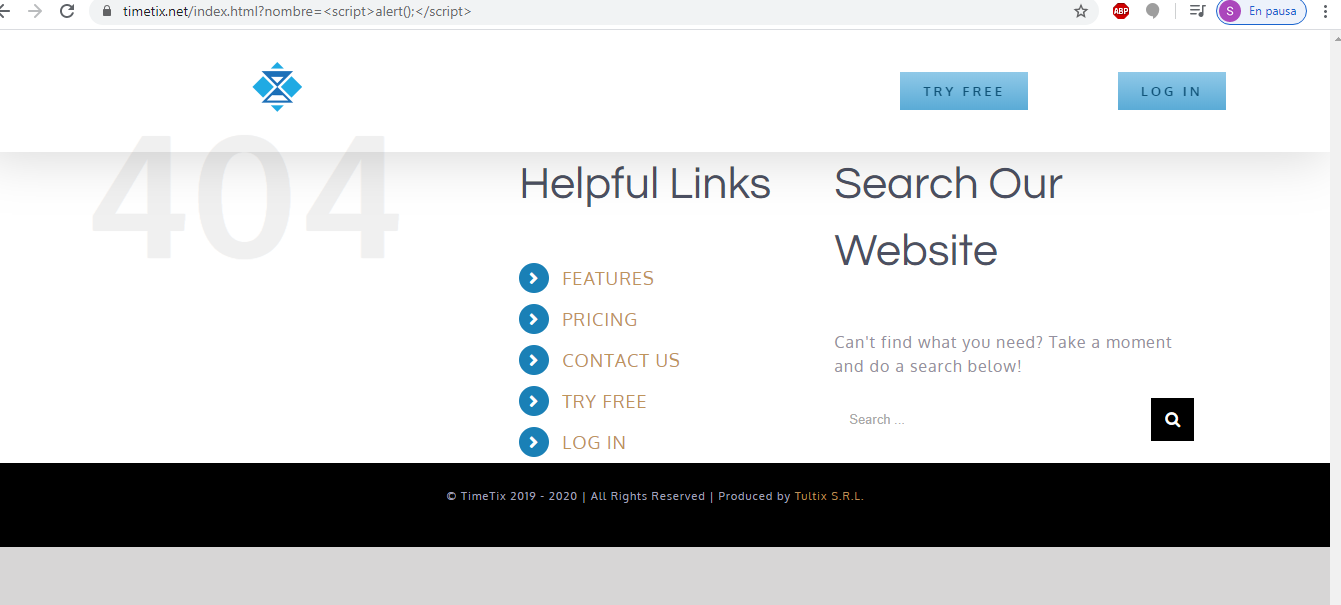
9.Componentes con vulnerabilidad conocidas: Los componentes como bibliotecas, frameworks y otros módulos se ejecutan con los mismos privilegios que la aplicación. Si se explota un componente vulnerable, el ataque puede provocar

una pérdida de datos o tomar el control del servidor. Las aplicaciones y API que utilizan componentes con vulnerabilidades conocidas pueden debilitar las defensas de las aplicaciones y permitir diversos ataques e impactos.

10. Registro y monitoreo insuficientes: El registro y monitoreo insuficiente, junto a la falta de respuesta ante incidentes permiten a los

atacantes mantener el ataque en el tiempo, pivotear a otros sistemas y manipular, extraer o destruir datos. Los estudios muestran que el tiempo de detección de una brecha de seguridad es mayor a 200 días, siendo típicamente detectado por terceros en lugar de por procesos internos.

Ejemplo 1: Ingrese la url https://timetix.net/index.html?nombre=<script>alert();</script> luego pulse enter cambio la web timetix.



Ejemplo 2: ingrese la url https://timetix.net/select\*from users where username=’pepe’ or ‘1’=´1’; luego pulse enter cambio la web.

