Nombres: Felipe acevedo, Alvaro Marin, Alejandro Montre

Fecha: 20-06-2024

Módulo: PRE-AUDITORÍA DE SOFTWARE

INTRODUCCIÓN

. Descripción del Sistema Auditado (Fruna)

El sistema auditado es el sitio web de Fruna, una empresa dedicada a la producción y comercialización de productos de confitería y alimentos. Una plataforma que es aparentemente básica o limitada en su estructura visual inicial, cumple con la función de promover y vender productos de confitería y alimentos. Esta página web sirve como vitrina virtual para los productos de la empresa, facilitando a los usuarios la navegación entre categorías de productos, información sobre la empresa y la posibilidad de realizar compras en línea.

. Objetivo de la Auditoría: Propósito y Alcance (Fruna)

Propósito

El objetivo de la auditoría es evaluar la seguridad, el rendimiento y la funcionalidad del sitio web de Fruna para garantizar que ofrece una experiencia segura, eficiente y confiable a sus usuarios. La auditoría identificará vulnerabilidades, medirá la capacidad del sitio para manejar tráfico elevado y verificará que todas las funcionalidades clave operen correctamente.

Alcance

La auditoría abarca los siguientes aspectos del sitio web de Fruna:

1. **Seguridad**: Utilizando OWASP ZAP para detectar vulnerabilidades comunes en la aplicación web.
2. **Rendimiento**: Empleando Apache JMeter para simular usuarios concurrentes y evaluar el tiempo de respuesta y la estabilidad bajo carga.

. Entorno de Prueba (Fruna)

El entorno de prueba es el sitio web de Fruna, donde realizaremos una auditoría exhaustiva utilizando tres herramientas de software especializadas: OWASP ZAP y Apache JMeter . OWASP ZAP se utilizará para identificar y analizar vulnerabilidades de seguridad en el sitio web. Apache JMeter nos permitirá llevar a cabo pruebas de rendimiento y carga.

https://www.fruna.cl

DESCRIPCIÓN DE OWASP ZAP

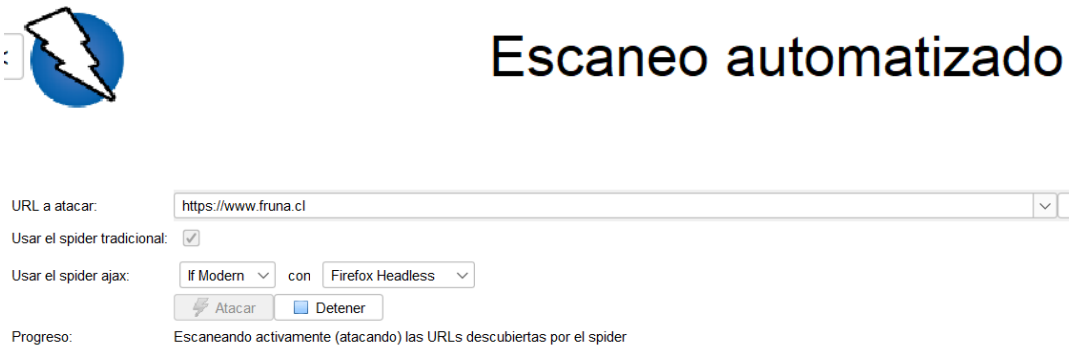
Zap es un escáner de seguridad web multiplataforma de código abierto. Zap tiene la capacidad de funcionar como un proxy intermediario entre el navegador y la aplicación web.Con zap podemos escanear vulnerabilidades, interceptar el tráfico, permite realizar ataques simulados para verificar la seguridad de la aplicación web y genera un informe con toda la información recopilada de las pruebas indicando vulnerabilidades de la aplicación web.

METODOLOGÍA USADA CON ZAP

En el programa de ZAP usamos el modo automático con el modo de ataque con el contexto predeterminado para verificar la seguridad del sitio web en caso de recibir algún ataque informático, se usó el spider ajax en If Modern en el navegador Firefox Headless.
El programa hace un ataque simulado a la página web para averiguar todas las vulnerabilidades con las que cuenta la página.

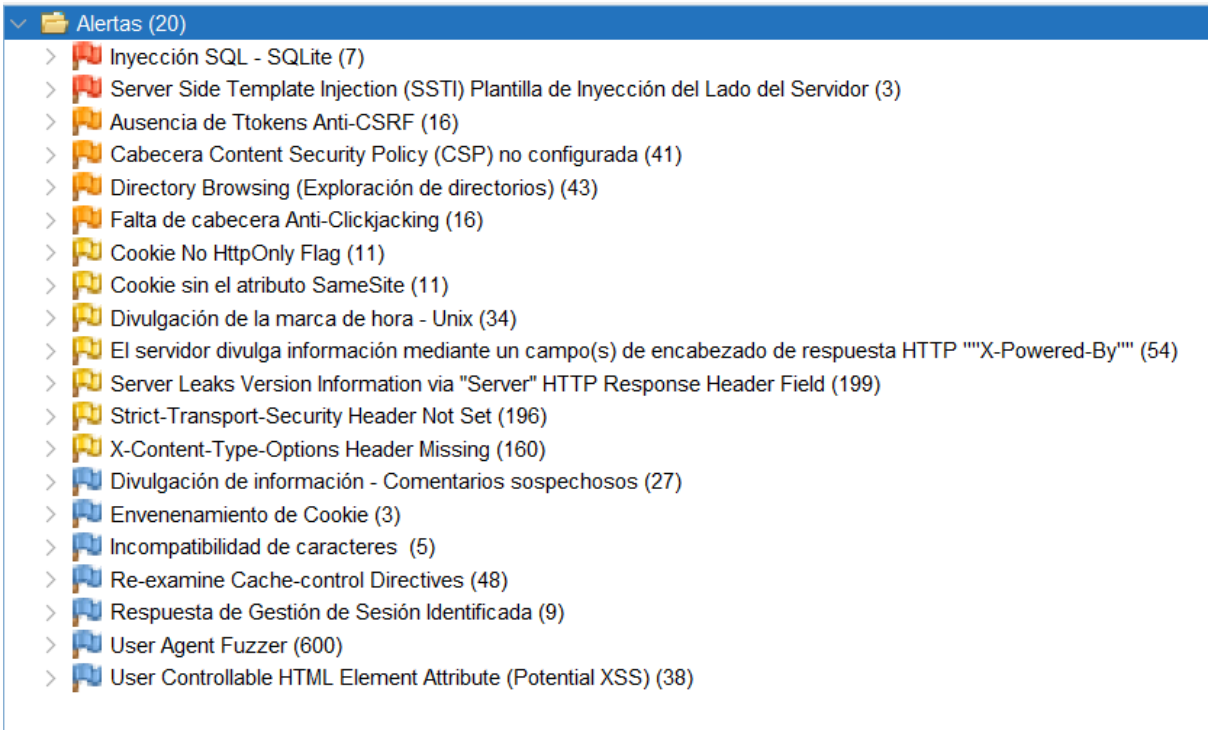
Criterios:

Escaneo automatizado en modo ataque el cual realiza pruebas de penetración agresivas a la pagina https://www.fruna.cl con el spider ajax en el navegador firefox Headless.

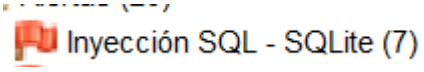


RESULTADOS

ZAP arrojo 20 alertas al momento de realizar el analisis 2 de un riesgo alto, 4 de un riesgo medio, 7 de un riesgo leve y 7 de un riesgo informativo:

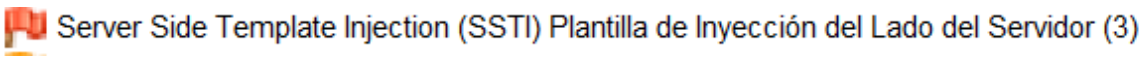


Riesgos altos:



Este riesgo indica que la inyección SQL es posible por ende se pueden ingresar o modificar datos de la base de datos de la página web ya sea para modificar o eliminar los datos o robarlos.

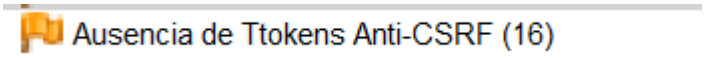
Recomendación:
Se recomienda el uso de un WAF(Web Application Firewall) el cual puede detectar y bloquear intentos de ingreso y de inyecciones SQL, también se puede implementar el uso de consultas parametrizadas lo cual asegura que los datos proporcionados por el usuario no se interpreten como sentencias SQL.



Cuando la entrada del usuario se inserta en la plantilla en lugar de usarse como argumento en el renderizado, el motor de plantilla evalúa. Dependiendo del motor de plantillas, puede producir ejecución remota de código.

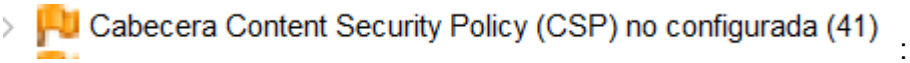
Recomendación:
Utilizar la entrada del usuario como argumento renderizado en lugar de en la plantilla, evitando la posibilidad de ejecución de un código malicioso.

Riesgo medio:



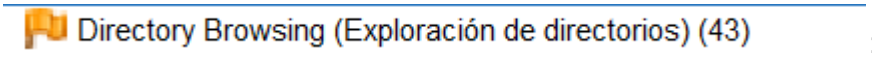
No tiene o no se encontraron TOKENS Anti-CSRF en un formulario de envio HTML.
Una solicitud falsa entre sitios en un ataque que obliga al usuario a enviar su solicitud HTTP a un destino sin su consentimiento para poder realizar una acción como víctima.

Recomendación:
Utilizar una biblioteca o Framework verificado que evite la vulnerabilidad o proporcione elementos que ayuden a evitarla.



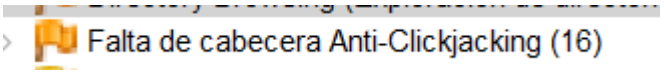
CSP no está configurado por lo que es susceptible a ciertos tipos de ataques como Cross Site Scripting (XSS) y ataques de inyección de datos. Estos tipos de ataques se utilizan para todo, desde robo de datos hasta la desfiguración del sitio o la distribución de malware.

Recomendación:
Asegurar que el servidor web esté configurado para establecer la cabecera Content Security Policy.



Es posible listar el directorio de sistema. El directorio de sistema puede mostrar scripts ocultos, archivos, archivos de copia de seguridad, etc., a los que se puede acceder para leer su información.

Recomendación:
Deshabilita el explorador de archivos. Si es necesario y asegurarse de que los archivos que se puedan mostrar no sean un riesgo.



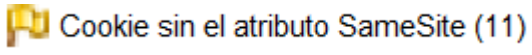
La respuesta no incluye Content-Security-Policy con la directiva 'frame-ancestors' ni X-Frame-Options para proteger contra ataques de 'ClickJacking'.

Recomendación:
Los navegadores web modernos admiten los encabezados HTTP Content-Security-Policy y X-Frame-Options. Hay que asegurarse de que uno de ellos esté configurado en todas las páginas web devueltas por el sitio.
Si queremos que la página esté enmarcada solo por páginas en el servidor, podemos usar SAMEORIGIN; en caso de no esperar que la página esté enmarcada, se puede usar DENY.

Riesgo bajo:

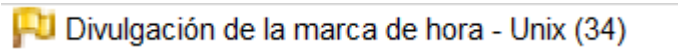
Se ha configurado una cookie sin el indicador HttpOnly, lo que significa que se puede acceder a la cookie mediante JavaScript. Si se puede ejecutar un script malicioso en esta página, se podrá acceder a la cookie y podrá transmitirse a otro sitio. Si se trata de una cookie de sesión, es posible que se produzca un secuestro de sesión.

Recomendación:
Es necesario establecer el HttpOnly Flag para evitar que una cookie pueda ser accedida por javascript, reduciendo el riesgo de que un atacante robe la cookie del usuario a través de scripts.



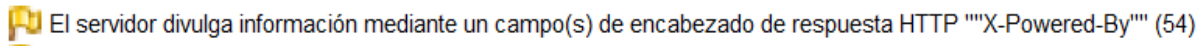
Se ha configurado una cookie sin el atributo SameSite, lo que significa que la cookie se puede enviar como resultado de una solicitud "entre sitios". El atributo SameSite es una contramedida eficaz contra la falsificación de solicitudes entre sitios, la inclusión de secuencias de comandos entre sitios y los ataques de sincronización.

Recomendación:
Asegurarse de que el atributo SameSite esté establecido como “lax” o idealmente “strict” para todas las cookies



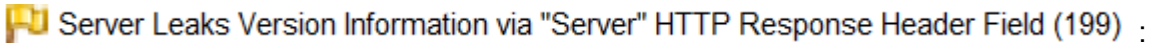
Se divulgó una marca de tiempo por el servidor del navegador.

Recomendación:
Confirmar manualmente que los datos de marca de hora no son sensibles, y que los datos no pueden ser agregados a patrones explotables de divulgación.



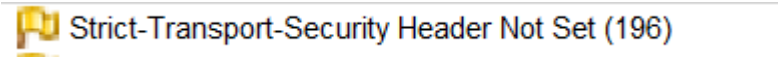
Se descubrió que el servidor de la página web divulga información mediante uno o mas encabezados de respuesta HTTP. El acceso a esta información facilita a los atacantes la identificación de otros marcos/componentes de los que la aplicación web depende.

Recomendación:
Asegurarse de que el servidor esté configurado para suprimir encabezados “X-Powered-By”



El servidor web/de aplicaciones está filtrando información de la versión a través del encabezado de respuesta HTTP "Servidor". El acceso a dicha información puede facilitar que los atacantes identifiquen otras vulnerabilidades a las que está sujeto su servidor web/aplicaciones.

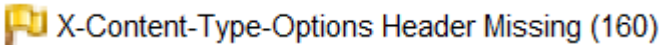
Recomendación:
Asegurarse de que el servidor web esté configurado para suprimir el encabezado “Servidor” o en su defecto que este mismo proporcione detalles irrelevantes y genéricos.



HTTP Strict transport security header es un mecanismo de política de seguridad web en donde el servidor declara que los agentes de usuario conformes deben interactuar con él utilizando únicamente conexiones HTTPS seguras. La página web no cuenta con estas medidas de seguridad.

Recomendación:

Establecer la cabecera HSTS para indicar a los navegadores que solo deben interactuar con el sitio utilizando conexiones HTTPS.



El encabezado Anti-MIME-Sniffing X-Content-Type-Options no estaba configurado en 'nosniff'. Esto permite que las versiones anteriores de Internet Explorer y Chrome realicen un rastreo MIME en el cuerpo de la respuesta, lo que podría provocar que el cuerpo de la respuesta se interprete y se muestre como un tipo de contenido distinto del tipo de contenido declarado. Las versiones actuales (principios de 2014) y heredadas de Firefox utilizarán el tipo de contenido declarado (si está configurado), en lugar de realizar un rastreo MIME.

Recomendación:
Asegurarse de que la aplicación/servidor web configure 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, también hay que asegurarse de que el usuario final utilice un navegador web moderno y compatible con los estándares que no realice ningún rastreo MIME, o que la aplicación web/servidor web pueda indicarle que no realice el rastreo MIME.

Riesgos Informativos:

Divulgación de información - Comentarios sospechosos (27) :

La respuesta parece contener comentarios sospechosos que pueden ayudar a un atacante.
Nota: Las coincidencias realizadas dentro de los scripts o archivos se refieren a todo el contenido, no solo a los comentarios.

Recomendación:
Eliminar todos los comentarios que devuelvan información que podría ayudar a un atacante y arreglar cualquier problema subyacente al que se refieran.

Envenenamiento de Cookie (3) :

Esta verificación analiza la entrada proporcionada por el usuario en los parámetros de la cadena de consulta y los datos POST para identificar dónde se pueden controlar los parámetros de las cookies. Esto es envenenamiento de cookies y se vuelve explotable cuando un atacante puede manipular la cookie de varias maneras.

Recomendación:
No permitir que la entrada del usuario controle los nombres y valores de las cookies. Si algunos parámetros de cadena de consulta deben establecerse en valores de cookies, también asegurarse de filtrar los puntos y comas que pueden servir como delimitadores de pares de nombre/valor.

Incompatibilidad de caracteres (5) :

Hay una discrepancia en el juego de caracteres entre el encabezado HTTP y el cuerpo del contenido, los navegadores web pueden verse forzados a entrar en un modo de rastreo de contenido no deseado para determinar el juego de caracteres correcto del contenido.

Recomendación:
Forzar UTF-8 para todo el contenido de texto tanto en el encabezado HTPP y etiquetas meta en HTML o declaraciones de codificación en XML.

Re-examine Cache-control Directives (48) :

El encabezado de control de caché no se ha configurado correctamente o falta, lo que permite que el navegador y los servidores proxy almacenen en caché el contenido. Para activos estáticos como css, js o archivos de imagen, esta podría ser la intención; sin embargo, se deben revisar los recursos para garantizar que no se almacene en caché ningún contenido confidencial.

Recomendación:
Para contenido seguro, hay que asegurarse de que el encabezado HTTP de control de caché esté configurado con "sin caché, sin almacenamiento, debe revalidar". Si un activo debe almacenarse en caché, hay que considerar configurar las directivas "públicas, de edad máxima, inmutables".

Respuesta de Gestión de Sesión Identificada (9) :

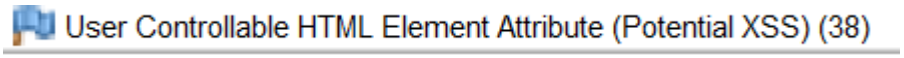
Se identificó que la respuesta dada contiene un token de gestión de sesión. Si la petición se encuentra en un contexto que tiene un método Session Management establecido en "Auto-Detect", esta regla cambiará la gestión de sesión para utilizar los tokens identificados.

Recomendación:
Asegurar las cookies de sesión, configurar expiración de sesiones, regenerar id de sesión y asegurar el almacenamiento de sesiones.

User Agent Fuzzer (600) :

Compare las diferencias en la respuesta según el fuzzeo de User Agent (por ejemplo, sitios móviles, acceso como un motor de búsqueda). Compara el código de estado de la respuesta y el hash del cuerpo de la respuesta con la respuesta original.

Recomendación:
Implementar una lista blanca de User-Agents válidos y rechaza cualquier solicitud que no coincida con esta lista o usar bibliotecas o servicios para analizar User-Agents y detectar patrones sospechosos.



:

Esta verificación analiza la entrada proporcionada por el usuario en los parámetros de la cadena de consulta y los datos POST para identificar dónde se pueden controlar ciertos valores de atributos HTML. Esto proporciona detección de puntos calientes para XSS (cross-site scripting) que requerirá una revisión adicional por parte de un analista de seguridad para determinar la explotabilidad.

Recomendación:
Validar todas las entradas y desinfectar las salidas antes de escribir en cualquier atributo HTML.

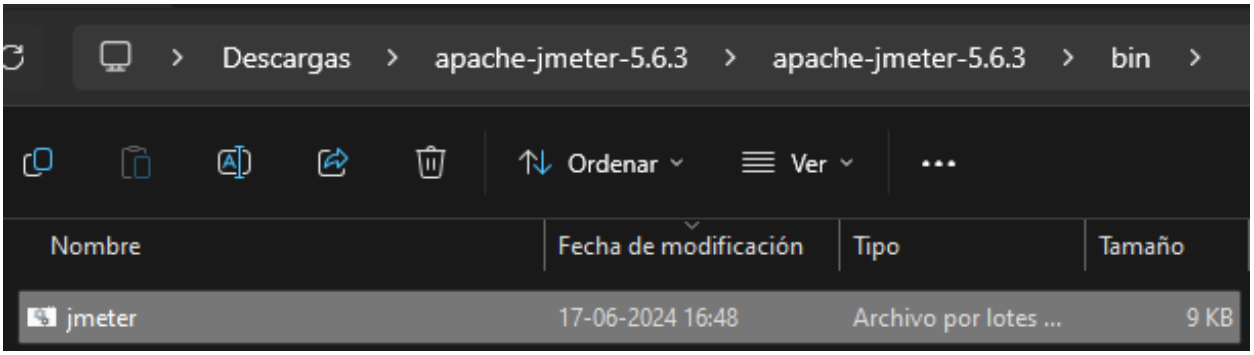
JMETER APACHE:

descripción de la herramienta:

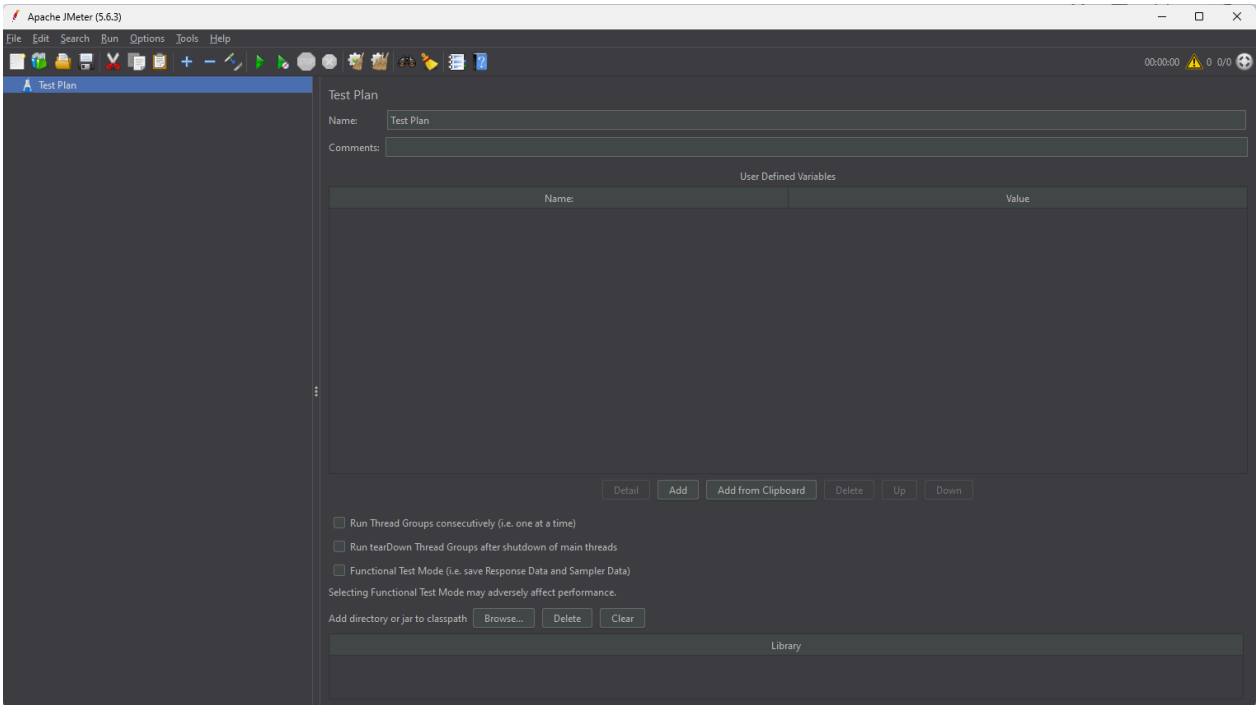
Es una herramienta de software libre diseñada para realizar pruebas de rendimiento en aplicaciones. Permite a los usuarios simular múltiples usuarios simultáneos para evaluar cómo se comporta una aplicación web, servidor, base de datos o red bajo diferentes niveles de carga. Con JMeter, es posible identificar cuellos de botella, medir tiempos de respuesta y asegurar que una aplicación pueda manejar un alto volumen de tráfico sin problemas. Es muy utilizada por desarrolladores y testers para garantizar que sus aplicaciones sean robustas y eficientes antes de ser lanzadas al público.

Método (pasos):

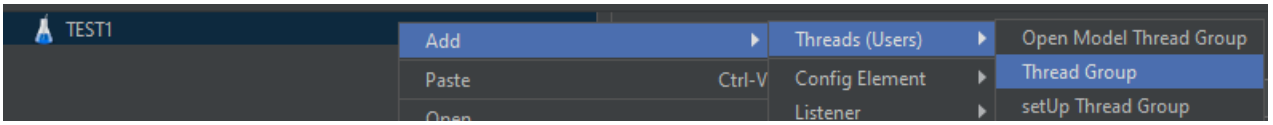
Revisar si tiene **java** instalado sino no se abrirá el programa principal. Abrir la carpeta descomprimida de apache jmeter y abrir **jmeter.bat**, se abrirá un cmd con un script automático para abrir el programa y su UI (Interfaz de usuario)



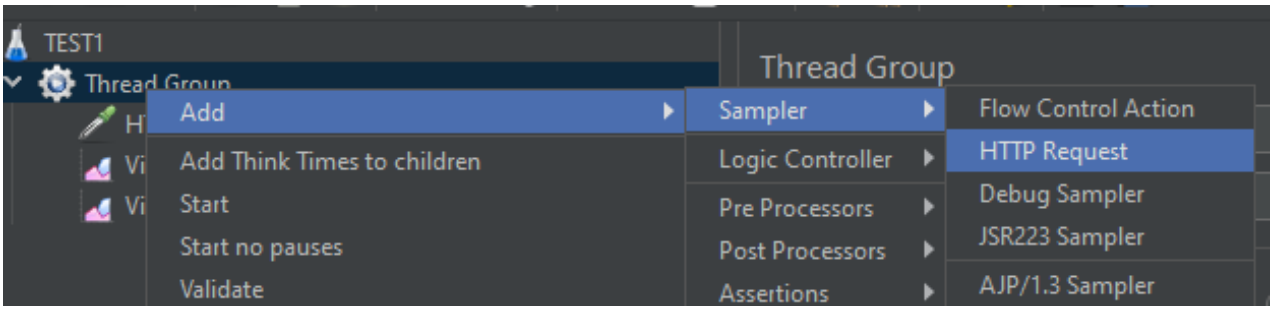
UI general



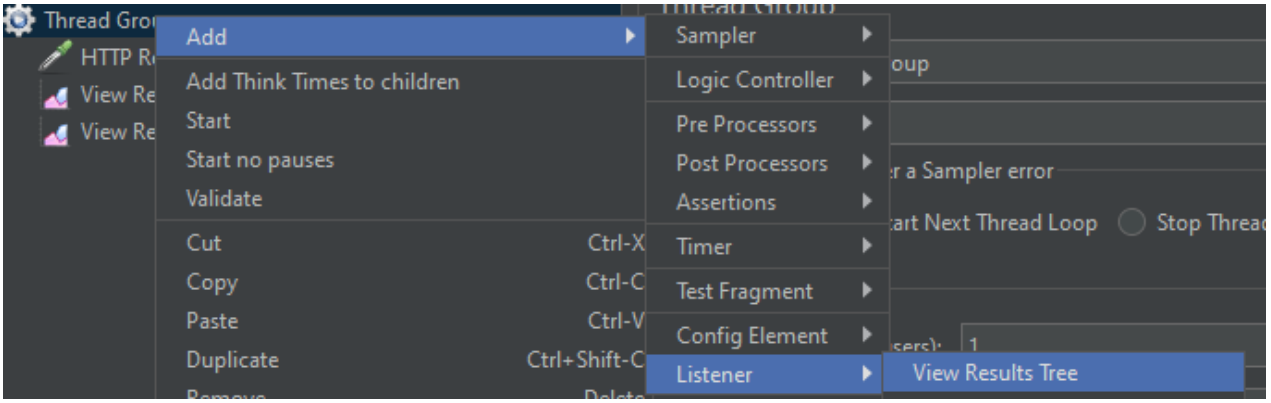
Ajustes para hacer peticiones request https con usuarios.
Creamos un grupo en el que se ajustaran nuestros usuarios luego



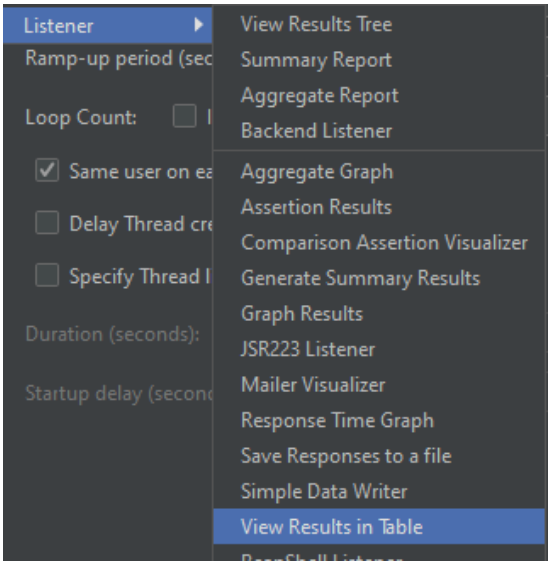
Dentro del grupo agregamos un **sampler** con la opción de **HTTP Request**



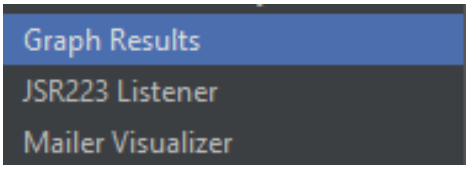
Dentro del grupo agregamos un listener para ver **resultados en forma de árbol**



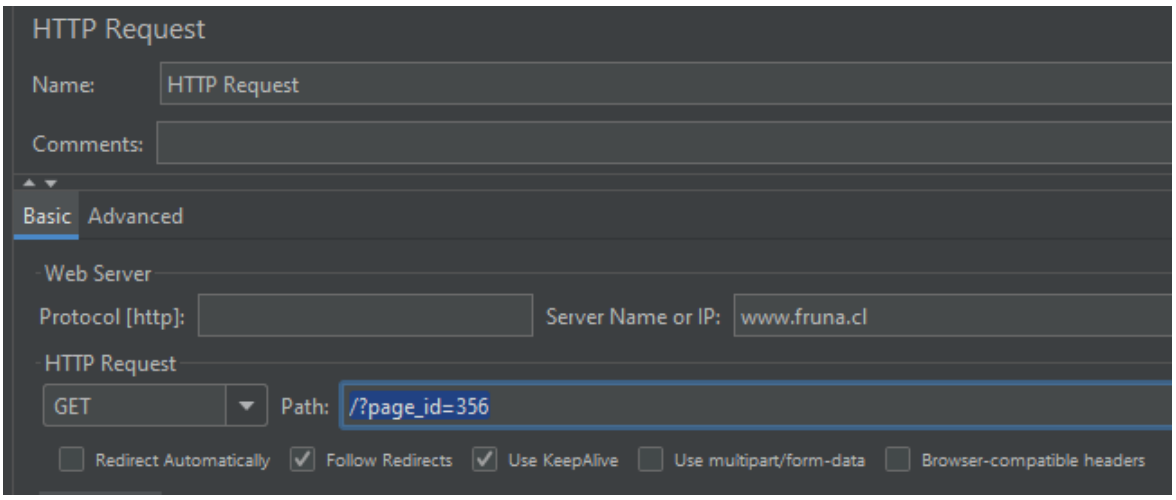
Agregamos otro listener pero para ver los **resultados en forma de tablas.**



Agregamos un listener más de para ver **resultados en gráficos** y así terminamos con la configuración general que mostrará los tres resultados de la forma requerida**.**

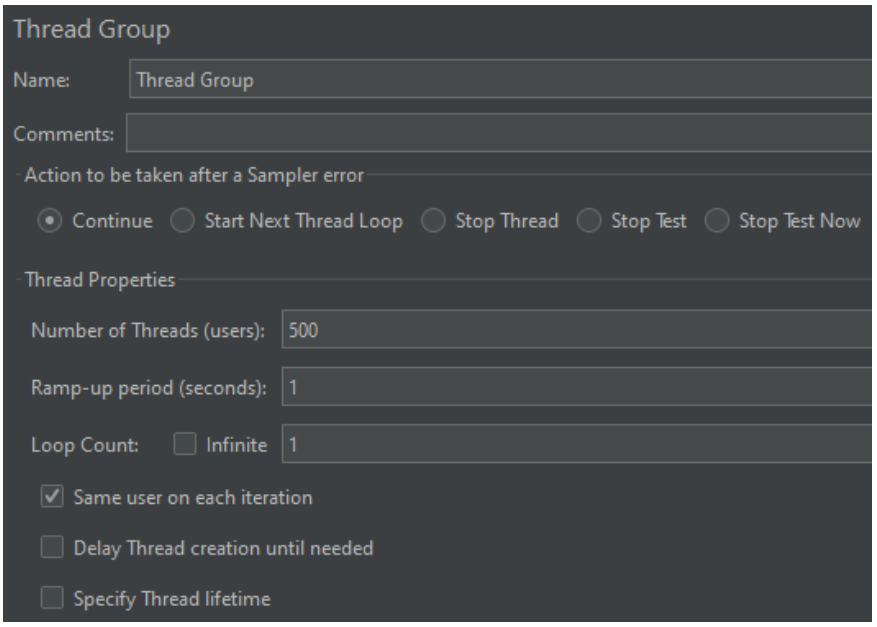


En el sampler “**HTTP Request**” Configuraremos el server (“www.fruna.cl”) en el espacio “**Server Name:**”, configuraremos el método en “**GET**” y el **path** con las dulcerías de fruna en este caso “**/?page_id=356**”, tal como se muestra en la imagen de abajo.

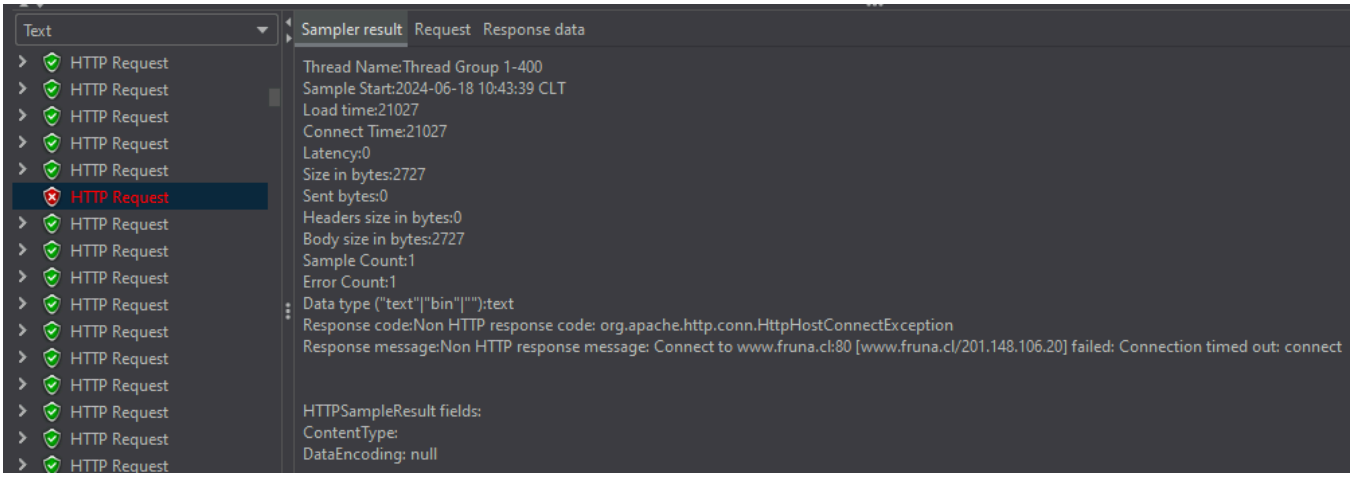


Criterios

Empezaremos con una configuración leve, en **Thread group** insertamos la cantidad de **500 usuarios**, tiempo de espera **(Ramp-up period)** de 1 segundo y 1 ciclo **(Loop count)**, hasta que resulte con mostrar un error o varios, luego aumentaremos gradualmente para ver si el error persiste o muestra diferencias y sacar conclusiones.



Resultado de árbol



Como se puede observar en la columna aparece un dato en rojo y en los resultados del sampler se ve que hay un solo 1 error con descripción Response message: “Connection time out”

Resultado en tabla:

	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Late...	Connect Time(ms)
	10:43:39.181	Thread Group 1-400	HTTP Request	21027		2727	0	0	21027
3	10:43:38.364	Thread Group 1-15	HTTP Request	7480		176643	258	19	9
10	10:43:38.344	Thread Group 1-6	HTTP Request	8734		176643	258	19	7
32	10:43:38.346	Thread Group 1-7	HTTP Request	16029		176643	258	20	11
6	10:43:38.350	Thread Group 1-8	HTTP Request	7947		176643	258	21	11
15	10:43:38.336	Thread Group 1-2	HTTP Request	9516		176643	258	21	13
16	10:43:38.358	Thread Group 1-12	HTTP Request	10237		176643	258	21	8

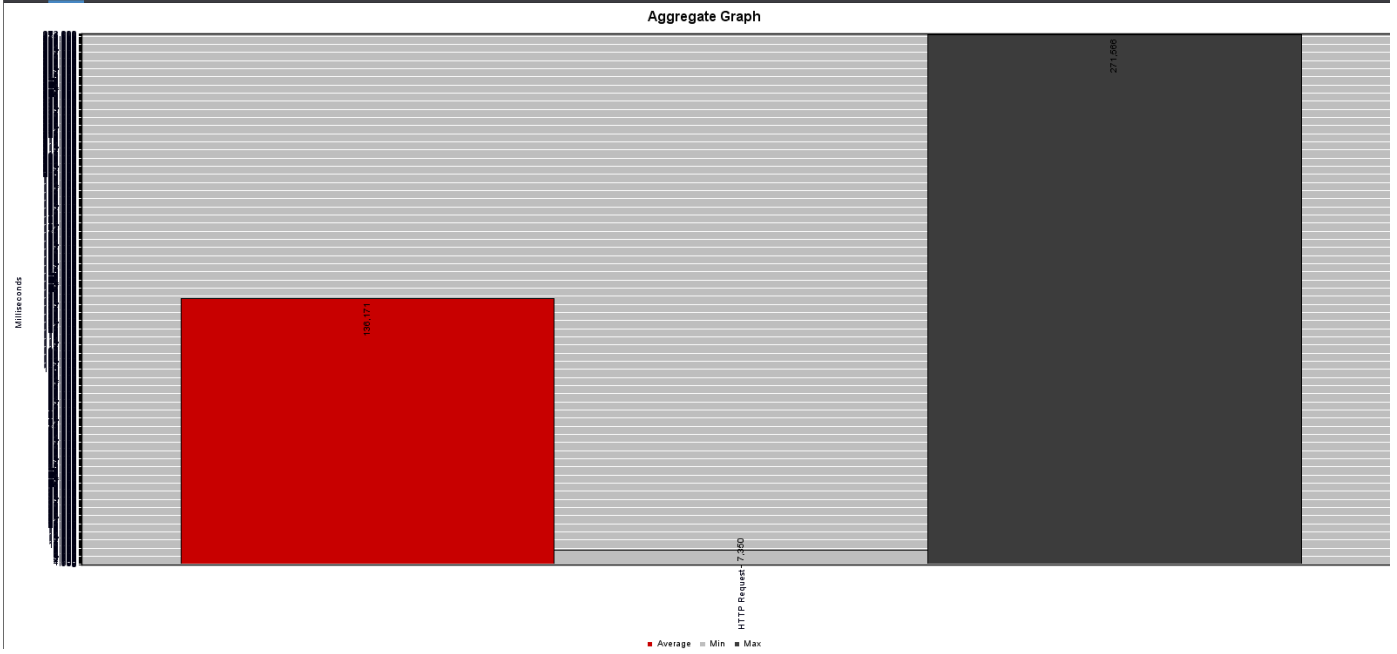
En los resultados por sampler de tabla se ordena la latencia de forma ascendente, para observar de mejor forma el error. Se analiza que tiene el mayor connect time o ping con 21027 ms, el menor envío de bytes y latencia con 0 en ambos

Resultado en gráfico:

Ajustando la selección para que de esta forma general, muestra la media, máxima y mínima de datos con éxito con los que se hizo la request.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
HTTP Request	500	136171	135605	252609	265224	270524	7350	271566	0.20%	1.8/sec	315.84	0.
TOTAL	500	136171	135605	252609	265224	270524	7350	271566	0.20%	1.8/sec	315.84	0.

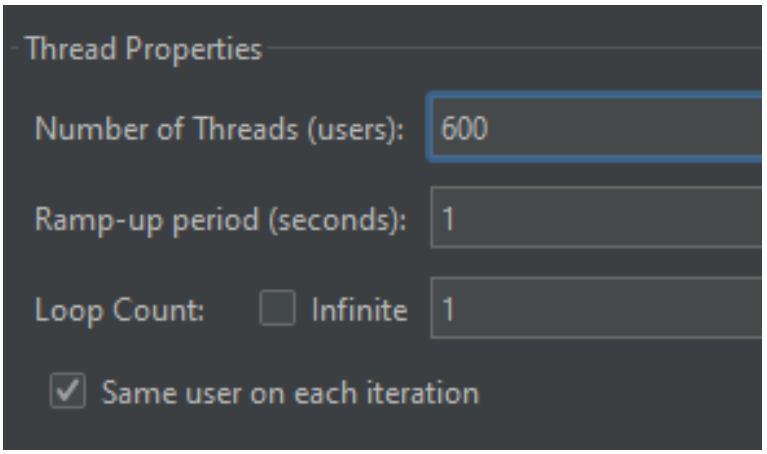
Settings | Graph



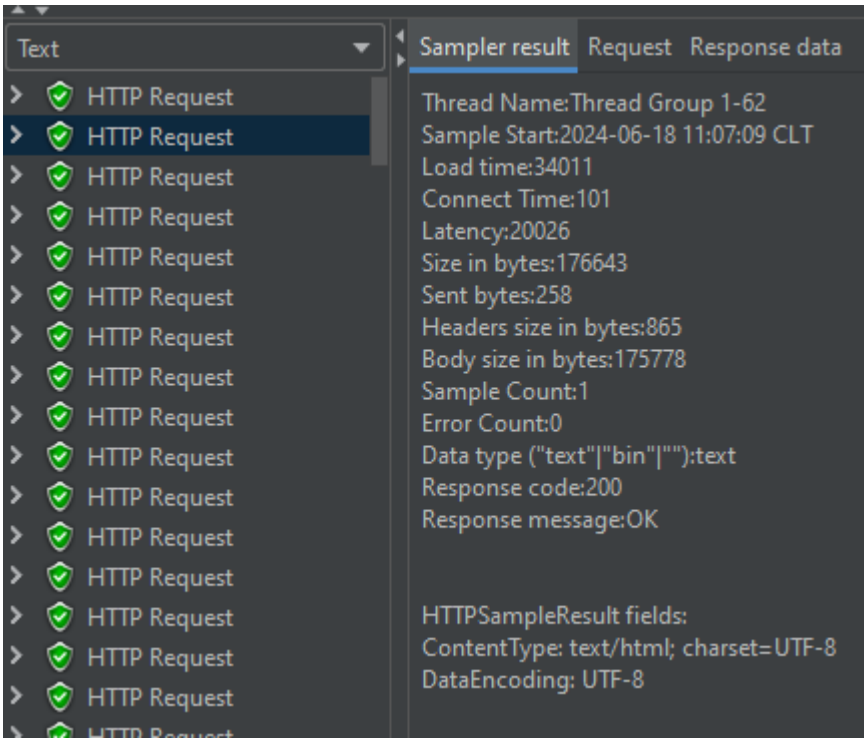
Datos de request con error del 0.20%, **Media** de 135605 (rojo), **Minima** de 7350 (gris) y **máxima** 271566 (gris oscuro)

Recomendación: es crucial optimizar el rendimiento del servidor y la base de datos para prevenir errores de "connection timeout", implementando herramientas de monitoreo y mejoras en la infraestructura de hosting para asegurar la escalabilidad y fiabilidad del sitio, especialmente bajo condiciones de alto tráfico.

Aumentamos el número de usuarios a 600 y 700 para ver si persiste el mismo error, es diferente o aumenta.



Resultado de árbol



A simple vista no se puede apreciar ningún error en las columnas especialmente cuando se lee Error Count:0

Resultado en tabla:

	Start Time	Thread Name	Label	Sample Time(...	Status	Byt...	Sent bytes	Latency	Connect Time(ms)
1	11:07:10.152	Thread Group 1-573	HTTP Request	2078		2723	0	0	2078
2	11:07:10.152	Thread Group 1-574	HTTP Request	2094		2723	0	0	2094
3	11:07:10.153	Thread Group 1-576	HTTP Request	2097		2723	0	0	2097
4	11:07:10.167	Thread Group 1-582	HTTP Request	2094		2723	0	0	2094
5	11:07:10.152	Thread Group 1-568	HTTP Request	2111		2723	0	0	2111
6	11:07:10.153	Thread Group 1-571	HTTP Request	2110		2723	0	0	2110
7	11:07:10.152	Thread Group 1-570	HTTP Request	2111		2723	0	0	2111
8	11:07:10.167	Thread Group 1-579	HTTP Request	2096		2723	0	0	2096
9	11:07:10.153	Thread Group 1-577	HTTP Request	2110		2723	0	0	2110
10	11:07:10.152	Thread Group 1-569	HTTP Request	2114		2723	0	0	2114
11	11:07:10.182	Thread Group 1-593	HTTP Request	2090		2723	0	0	2090
12	11:07:10.153	Thread Group 1-575	HTTP Request	2119		2723	0	0	2119
13	11:07:10.167	Thread Group 1-583	HTTP Request	2107		2723	0	0	2107
14	11:07:10.182	Thread Group 1-586	HTTP Request	2092		2723	0	0	2092
15	11:07:10.168	Thread Group 1-585	HTTP Request	2106		2723	0	0	2106
16	11:07:10.168	Thread Group 1-581	HTTP Request	2112		2723	0	0	2112
17	11:07:10.182	Thread Group 1-587	HTTP Request	2098		2723	0	0	2098
18	11:07:10.152	Thread Group 1-572	HTTP Request	2134		2723	0	0	2134
19	11:07:10.183	Thread Group 1-592	HTTP Request	2108		2723	0	0	2108
20	11:07:10.197	Thread Group 1-598	HTTP Request	2094		2723	0	0	2094
21	11:07:10.197	Thread Group 1-599	HTTP Request	2094		2723	0	0	2094
22	11:07:10.167	Thread Group 1-580	HTTP Request	2124		2723	0	0	2124
23	11:07:10.182	Thread Group 1-590	HTTP Request	2115		2723	0	0	2115
24	11:07:10.183	Thread Group 1-588	HTTP Request	2118		2723	0	0	2118
25	11:07:10.167	Thread Group 1-578	HTTP Request	2134		2723	0	0	2134
26	11:07:10.197	Thread Group 1-597	HTTP Request	2094		2723	0	0	2094
27	11:07:10.168	Thread Group 1-584	HTTP Request	2123		2723	0	0	2123
28	11:07:10.183	Thread Group 1-589	HTTP Request	2120		2723	0	0	2119
29	11:07:10.197	Thread Group 1-596	HTTP Request	2119		2723	0	0	2119
30	11:07:10.197	Thread Group 1-600	HTTP Request	2119		2723	0	0	2119
31	11:07:10.197	Thread Group 1-594	HTTP Request	2135		2723	0	0	2135
32	11:07:10.182	Thread Group 1-591	HTTP Request	2150		2723	0	0	2150
36	11:07:10.183	Thread Group 1-595	HTTP Request	9586		2723	0	0	9586
33	11:07:08.948	Thread Group 1-10	HTTP Request	10106		176643	258	27	16
34	11:07:08.946	Thread Group 1-9	HTTP Request	10607		176643	258	29	18

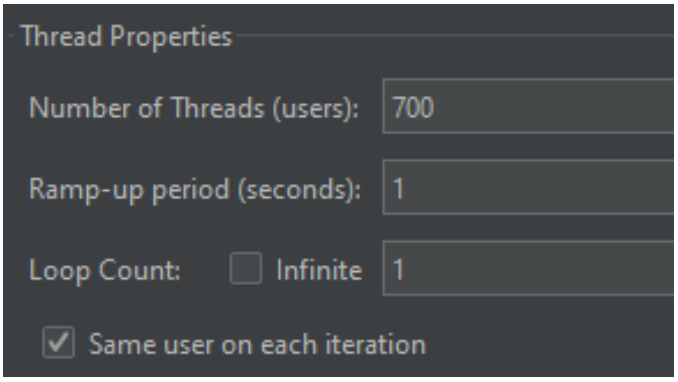
Pero aparentemente luego aparecen un montón de errores al principio de las request y se arreglan sin ningún otro error mostrado.

Resultado en gráfico:

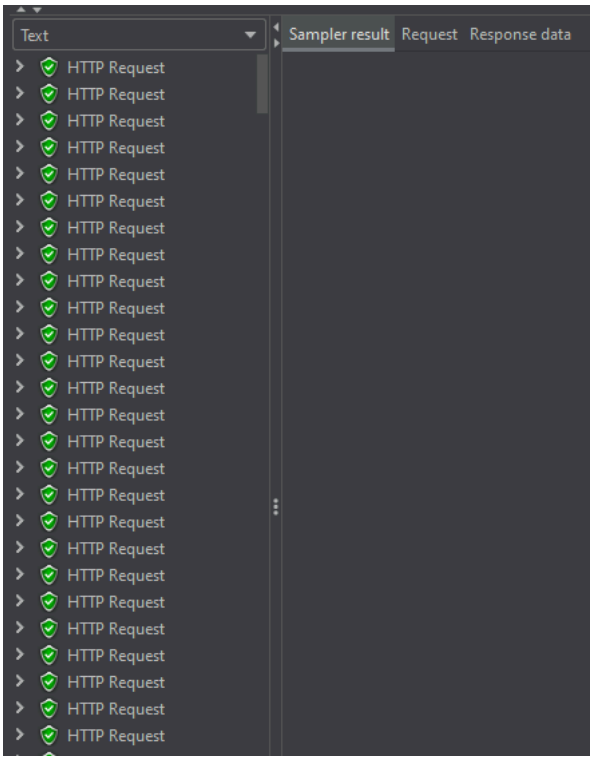
Datos de request con error del 5.50%, **Media** de 128351 (rojo), **Minima** de 2078 (gris) y **máxima** 263970 (gris oscuro)

Recomendación: Implementar soluciones de escalabilidad, como el balanceo de carga y el uso de Content Delivery Networks (CDN), puede ayudar a distribuir la carga de manera más eficiente. Además, se sugiere realizar pruebas de estrés periódicas y ajustar los parámetros de timeout del servidor para mejorar la respuesta inicial bajo condiciones de alto tráfico.

Aumentamos un poco más la cantidad pero mostró cantidades similares



Resultado de árbol



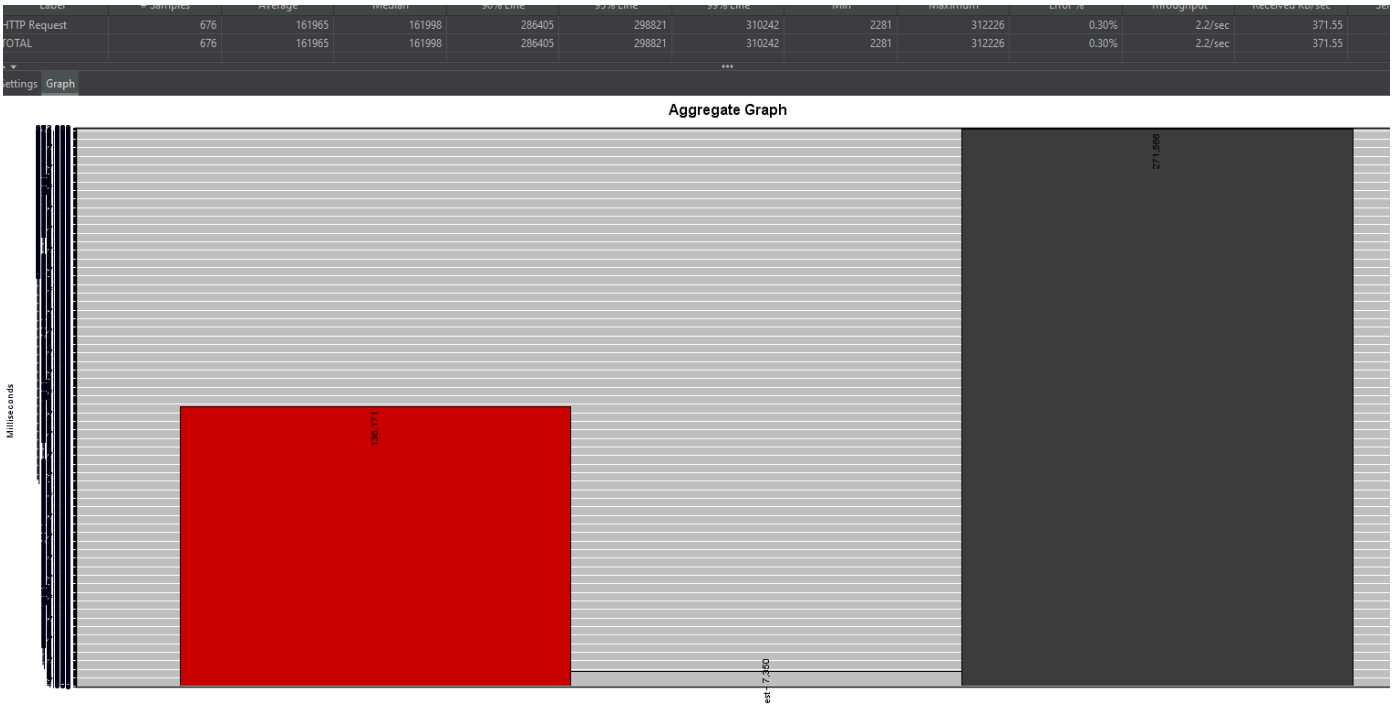
Ningún error en ambas columnas

Resultado en tabla:

	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Byt...	Latency	Connect Time(ms)
1	11:14:55.161	Thread Group 1-382	HTTP Request	2281		2723	0	0	2281
2	11:14:55.160	Thread Group 1-379	HTTP Request	2287		2723	0	0	2287
3	11:14:54.518	Thread Group 1-6	HTTP Request	4150		176643	258	30	11
4	11:14:54.522	Thread Group 1-7	HTTP Request	4409		176643	258	27	13
5	11:14:54.531	Thread Group 1-13	HTTP Request	5283		176643	258	32	18
6	11:14:54.534	Thread Group 1-18	HTTP Request	5556		176643	258	29	14

Igual que anteriormente solo mostró errores del mismo tipo “Connection time out” al principio pero luego ningún otro más.

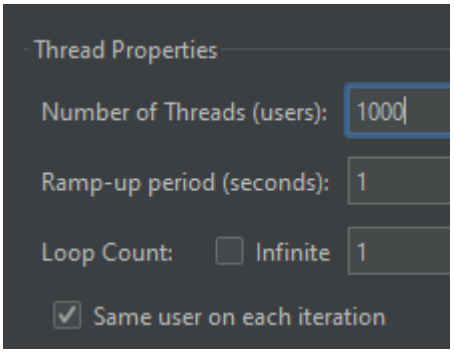
Resultado en gráfico:



Datos de request con error del 0.30%, **Media** de 161998 (rojo), **Minima** de 2281 (gris) y **máxima** 312226 (gris oscuro)

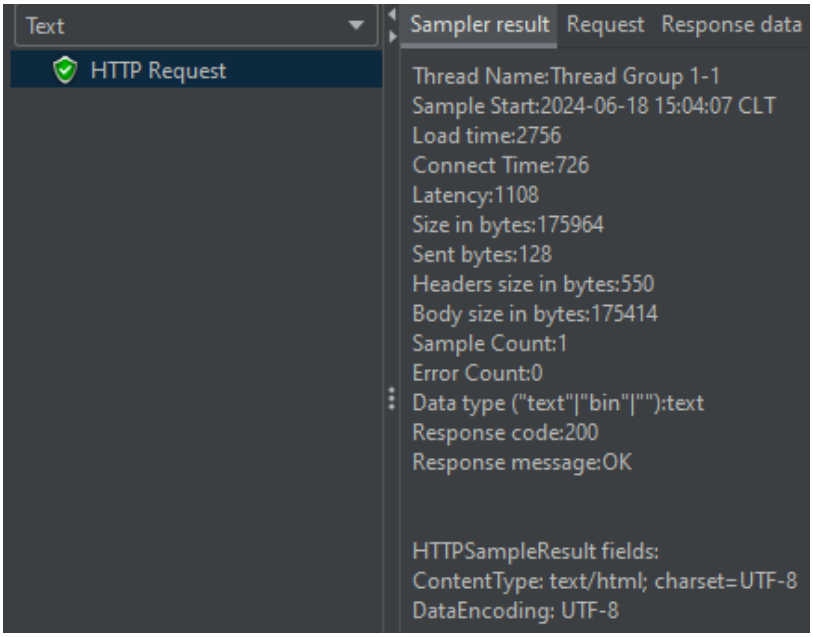
Recomendación: Mejorar la capacidad de su infraestructura técnica para manejar mejor el aumento de tráfico, ya que con 700 usuarios se presentaron errores al inicio. Esto incluye actualizar a un plan de hosting más robusto, optimizar el código del backend y las consultas a la base de datos, e implementar un sistema de caché eficiente.

Se sube la cantidad de usuarios a 1000 una cantidad moderada para hacer pruebas en páginas web.



Intente correr el programa varias veces pero se **cerraba inesperadamente** y se abre el cmd con esto:

Baje la cantidad de a poco para ver si funcionaba empezando con 700 hasta llegar a 1 pero siguió pasando el mismo error y mis sospechas son que la página web habrá bloqueado mi IP para que no sobrecargue con request y que colapse. Finalmente hice una prueba con un solo usuario en otra computadora y esta funcionaba sin problemas.



Recomendación: Como es una prueba no autorizada a la compañía pero aun asi no me bloquearon la conexión a partir de las 700 request se le recomienda reforzar la seguridad de su página web para prevenir pruebas no autorizadas y posibles ataques de denegación de servicio (DoS). Implementar un firewall de aplicaciones web para filtrar y monitorear el tráfico HTTP, establecer límites de tasa de conexión para evitar sobrecargas, y configurar sistemas de detección y prevención de intrusiones (IDS/IPS) si es que no los tiene.

Conclusiones

Resumen de los Hallazgos
Resumir los principales hallazgos de la auditoría.

ZAP:
Los resultados que nos otorgo OWASP ZAP despues del escaneo automático en la pagina de Fruna, fueron bastante preocupantes, los resultados arrojaron múltiples vulnerabilidades bastante graves como la posibilidad de hacer inyecciones SQL o el poder redirigir al usuario a otras paginas e incluso robar las cookies, esto nos demuestra el pobre nivel de seguridad de la pagina web y la urgencia de realizar una mejora en este apartado.

Apache JMeter:
Las pruebas de auditoría con Apache JMeter revelaron que la página web de la empresa de confitería experimenta errores de "connection timeout" y problemas de estabilidad bajo cargas de tráfico elevado, especialmente con 700 usuarios o más. También se identificó que la infraestructura actual no está optimizada para manejar grandes volúmenes de usuarios simultáneos, sugiriendo la necesidad de mejoras en la capacidad del servidor y en las configuraciones de seguridad para prevenir accesos no autorizados y asegurar un rendimiento consistente.

Efectividad de las Herramientas Utilizadas
Evaluar la efectividad de las herramientas seleccionadas en la auditoría.

ZAP:
OWASP ZAP nos ha resultado una herramienta sumamente efectiva y útil, nos demostró el peligro que corre la empresa al mantener la página web en el estado actual. Zap nos entregó bastante información sobre qué tipo de vulnerabilidades encontró y como poder solucionar estas vulnerabilidades, sumado a la interfaz sencilla y a la posibilidad de crear un informe de manera automática con la información recopilada hace que Zap sea una herramienta muy recomendable tanto por su funcionalidad y rendimiento como por su fácil uso y baja barrera de entrada.

Apache JMeter:
Apache JMeter demostró ser una herramienta eficaz para evaluar el rendimiento y la capacidad de la página web de la empresa Fruna, identificando claramente los puntos de fallo bajo distintas cargas de usuarios. A través de estas pruebas, se pudieron detectar errores críticos de "connection timeout" y problemas de estabilidad, proporcionando datos valiosos para guiar mejoras en la infraestructura del servidor.

Reflexión Final
Reflexiones sobre el proceso de auditoría y cualquier aprendizaje significativo.

Después de terminar esta auditoría y de analizar los resultados otorgados de la investigación y los resultados de los programas, nos damos cuenta de lo vulnerable que es el sitio web de Fruna a pesar de ser el sitio de una empresa grande en el país, las múltiples vulnerabilidades que presenta esta web son indicativo del pobre manejo que se ha tenido en la creación de la misma además de demostrar la necesidad de un replanteamiento sobre cómo mejorar la seguridad de la empresa en todo ámbito y evitar robos de información o manipulaciones en datos importantes para una empresa.