Sistema de Seguimiento Cardiaco

Juan Sebastián Barragán Jerónimo, Sebastián Sánchez Galiano, Santiago Rojas Herrera

Experimento 2 – Entrega 2

{s.rojas19, js.barragan796, s.sanchez59}@uniandes.edu.co

Fecha de presentación: 6 de abril de 2017

Bogotá, Colombia

Tabla de contenidos

1. Pre-experimentación

1.1. Problemática

Sobre el sistema desarrollado para el manejo de servicios para pacientes del Hospital Cardiológico de Santa fé, se debe implementar las funcionalidades pertinentes a la seguridad del sistema, que permita la autorización, autenticación y el manejo de sesiones en los servicios para los usuarios.

1.2. Objetivo del experimento

Se espera comprobar el impacto puede tener la implementación de técnicas de seguridad en el sistema, específicamente con el uso del framework **Shiro** y los servicios cloud de **Stormpath**, en el desempeño del sistema, llegando a concluir los compromisos necesarios, y más adecuados, para llevar a cabo la operación del hospital cardiológico de Santa fé.

1.3. Descripción del experimento

En el documento se expondrán pruebas de carga para el envío de mediciones desde un dispositivo externo hacia el servidor de la aplicación. Para esto, se hará uso de la herramienta *JMeter*, que permite simular el envío de múltiples peticiones (en este caso http) hacia la aplicación.

Estas pruebas se harán para la aplicación con seguridad desplegada a un nodo (conexión directa) con el fin de comparar el impacto directo de la implementación sobre la aplicación

1.4. Artefactos a construir

- Integración del framework ***Shiro*** en la aplicación
- Modificación de microcontrolador para que pueda autenticarse
- *Configuración de **Stormpath****

1.5. Recursos de la experimentación

- **Nodo de despliegue**
    - Intel Xeon E5-2630 2.6GHz (4 núcleos)
    - 12GB Ram

1.6. Resultados esperados

Se espera que el desempeño de la aplicación sea comprometido por el incremento de overhead producto de la necesidad de envío y recibimiento de solicitudes a los servicios de **Stormpath** en la nube.

1.7. Duración y etapas

Actividad	**Tiempo de desarrollo**
Integración de **Shiro**	~5 horas
Modificación del **microcontrolador**	~3 horas
Configuración de **Stormpath**	~1.5 horas
Pruebas de carga	~1.5 horas
Documentación pruebas	~1 horas

2. Experimentación

- **1 iteración de n peticiones en 1 segundo**

Despliegue sin seguridad

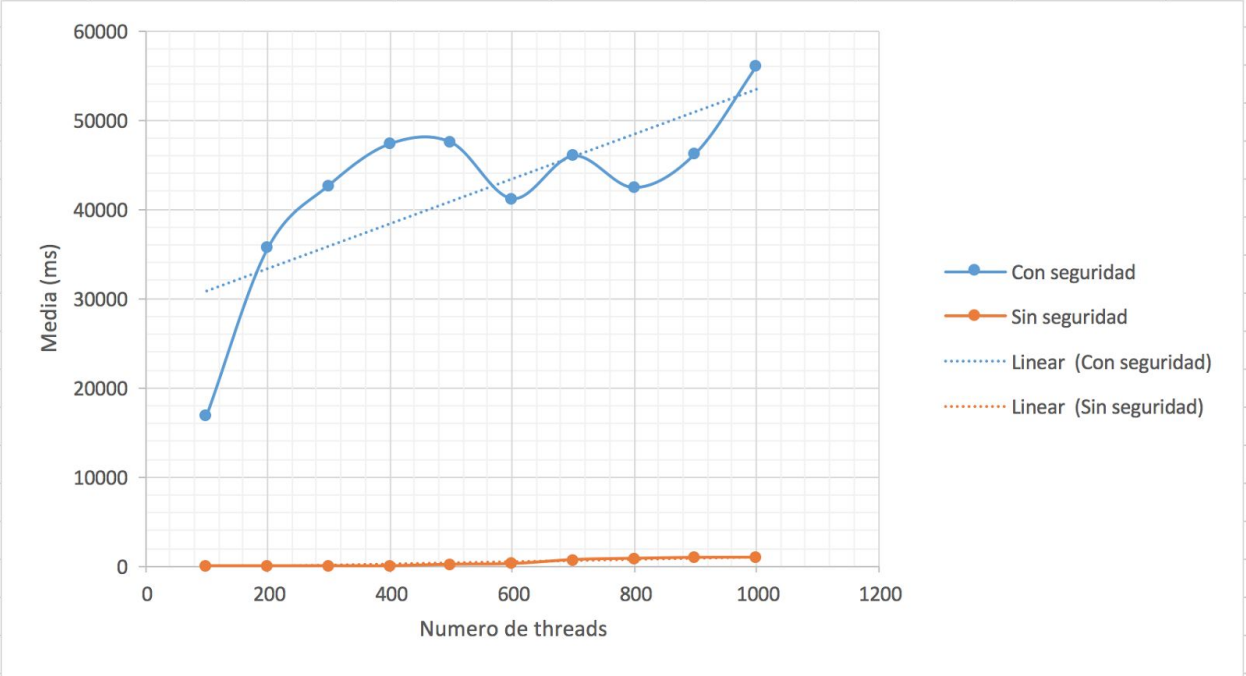
# thread	**Media (ms)**	**Error (%)**	**Throughput (op/s)**
1200	1169	0	305.5
1100	984	0	289.3
1000	955	0	286.6
900	942	0	284.3
800	841	0	290.1
700	696	0	298.0
600	243	0	308.6
500	166	0	318.1

400	2	0	266.3
300	3	0	294.1
200	2	0	173.9
100	3	0	87.8

Despliegue con seguridad

# thread	Media (ms)	Error (%)	Throughput (op/s)
1200	N/A	N/A	N/A
1100	N/A	N/A	N/A
1000	56012	N/A	N/A
900	46232	N/A	N/A
800	42384	N/A	N/A
700	45934	N/A	N/A
600	41143	N/A	N/A
500	47429	N/A	N/A
400	47283	N/A	N/A
300	42649	N/A	N/A
200	35609	N/A	N/A
100	16911	N/A	N/A

Comparación resultados



3. Post-experimentación

3.1. Duración real

Actividad	**Tiempo de desarrollo**	Δ
Integración de **Shiro**	~4 horas	~1 horas
Modificación del **microcontrolador**	~3 horas	~0 horas
Configuración de **Stormpath**	~0.5 horas	~1 horas
Pruebas de carga	~1.5 horas	~0.5 horas
Documentación pruebas	~1 horas	~0 horas

3.2. Artefactos construidos

- Integración del framework ***Shiro*** en la aplicación
- Modificación de microcontrolador para que pueda autenticarse

- Configuración de **Stormpath no se creó**, pues fue imposible crear una cuenta, y no se tuvo suficiente tiempo para la búsqueda e implementación de otra alternativa.

3.3. Análisis

Los resultados de la experimentación, evidencian un decremento considerable del desempeño de la aplicación; Sin embargo, vale la pena recordar las condiciones en las que se hizo las pruebas de carga con Jmeter:

> Ya que JMeter no recordaba la autorización por *token* de acceso que se tenía al ingresarlo por primera vez, a diferencia de un navegador de internet o el microcontrolador, era necesario enviar el token cada vez para que se hiciera la verificación del usuario usando los servicios; Este caso se aleja de la aplicación en la vida real, pues los microcontroladores (o el dispositivo móvil del usuario) no necesita hacer la verificación del token en **Stormpath** cada vez que se haga una solicitud, siendo que se prioriza el desempeño sobre la seguridad en este área.

Así, las pruebas realizadas no muestran un escenario de uso realista, pero se optó por hacer el análisis de este caso, ya que Jmeter no contaba con las herramientas para recordar la autenticación por token en posteriores peticiones.

También se debe recordar que existe otro compromiso al usar una solución como **Stormpath**, que es la dependencia en el **desempeño y disponibilidad** de este servicio sobre el de los servicios del hospital; Por esto, puede resultar una mejor alternativa el uso de un framework, o servicio que pueda ser desplegado sobre los servidores propios del hospital, aunque se optó por la implementación con Stormpath por el conocimiento que se tenía de esta herramienta por el equipo de desarrollo, y por la complejidad de la solución que se solicitó.

3.4. Conclusiones

En conclusión, la implementación de seguridad con un servicio desplegado en la nube aunque cumple con los requerimientos de seguridad estipulados, atenta críticamente el desempeño de la aplicación, lo que va en contra de las actividades de negocio y los requerimientos no funcionales del sistema.

El uso de una implementación de seguridad desplegada sobre la misma infraestructura usada para la aplicación y servicios del negocio podría solucionar esto.