

Caso - Compliance continuo de servidores

Contexto

El equipo de Seguridad Informática de Mercado Libre, se encarga de realizar controles sobre el compliance continuo de distintos servidores en la nube. Por las cantidades de servidores que se administran, poder asegurar manualmente que estos cumplan el estándar definido por el equipo no es escalable, y se busca alguna forma de poder alivianar esta tarea.

Objetivo

Generar un programa (agente) que pueda ser ejecutado en uno de nuestros servidores, y a partir de distintos comandos del sistema operativo, pueda enviar los siguientes datos a una aplicación (API):

- Información sobre el procesador.
- Listado de procesos corriendo.
- Usuarios con una sesión abierta en el sistema.
- Nombre del sistema operativo.
- Versión del sistema operativo.

Este segundo programa (API) debe consistir en una aplicación con un endpoint que permita al agente enviar la información relevada, para que luego, sea almacenada en un .csv con el formato:

<IP de servidor>_<AAAA-MM-DD>.csv

Por lo tanto, el sistema a construir estaría conformado por:

1. Agente - Que será ejecutado en el servidor para recolectar información
2. API - Para centralizar esta información que proviene de los distintos agentes

Entregables

Se espera:

- Código fuente (en zip o url al repositorio)
- Instrucciones para la ejecución de la aplicación (incluida cualquier aplicación o librería a instalar para el correcto funcionamiento del programa)
- Descripción de la aplicación realizada, supuestos, problemas y soluciones con los que se encontró al realizar la misma.
- Endpoint con detalle de cómo enviarle un request para recuperar la información centralizada por el servidor.
- Evidencia de servidor corriendo en instancia de AWS EC2.

Consideraciones generales

A tener en cuenta:

- Podes usar el lenguaje que te sea más cómodo.

- Deberás instalar una máquina virtual para poder levantar el sistema operativo.
 - Sistema operativo recomendado: <https://www.debian.org/distrib/>
- Se podrán crear todas las funciones/scripts/etc complementarios que se consideren necesarios para un correcto funcionamiento de la aplicación.
- Tener el código en Github
- Tu servidor deberá estar corriendo en una instancia de AWS EC2 (podes generar una cuenta gratuita).

Bonus (opcional):

- Almacenamiento de la información en formato JSON en un archivo de texto plano en lugar de un .csv.
- Documentación.
- Detalle de qué otra funcionalidad crees que podríamos implementar en este sistema a futuro.

Ultra Bonus (opcional)

- Dockerizar la aplicación
- Almacenar toda la información en una DB normalizada