Análisis y entendimiento del problema

Daniel Galindo Ruiz 201813517 Violeta Luna Rodríguez Moreno 201512346

**Caso 2 - Infraestructura computacional**

Departamento de Ingeniería de Sistemas Computación

Universidad de Los Andes

En este documento se muestra el acercamiento y análisis inicial a la construcción del *Sistema Time & Attendance.* Este es un sistema de información que tiene como objetivo automatizar y agilizar los procesos de registro de la llegada y la salida de los empleados de a campo a sus turnos.

# Datos que maneja el sistema

A continuación, se presenta una lista de los datos que maneja el sistema y que deben ser protegidos.

1. Información (ubicación y hora) de llegada y salida de los trabajadores de campo.
2. Información de usuarios y permisos dentro de la ampliación.

# Requerimientos de seguridad

En esta sección se presentan los requerimientos de seguridad para cada uno de los datos que se listaron en la sección anterior.

1. Para la información de llega y salida de los trabajadores de campo, es importante que se cumpla el siguiente criterio:

* **Integridad:** los datos deben permanecer de acuerdo con la realidad. Esto es importante, dado que es con base en los datos de ingresos y salidas, que se supervisará evaluará una parte del desempeño de cada empleado. De no conservar la integridad los datos, podría dejarse pasar llegadas tarde reiterativas o llamar la atención por llegadas a destiempo, a personas que cumplen a cabalidad con los horarios.
* **Confidencialidad:** es necesario que, para esta información, se maneje confidencialidad ya que tiene datos sensibles como la ubicación de un trabajador en un momento del día. Así que es necesario que sólo los supervisores sepan esa información y se maneje con su debida confidencialidad. Si no se maneja de esta forma, los datos de ubicación de los trabajadores podrían ser usados con malas intenciones por cualquier persona al saber dónde se moviliza cada trabajador e incluso poder conocer su dirección de residencia.

1. Para la información de los usuarios y permisos dentro de la aplicación:

* **Autenticación:** debe ser necesario autenticar que los usuarios de la aplicación sí sean supervisores o trabajadores. Esto porque puede que sus dispositivos personales hayan sido robados o usados por personas ajenas a la empresa, así que, desde la aplicación en el smartphone de los trabajadores, debería existir una forma de autenticar que la persona a reportarse en efecto sea el trabajador. De igual forma con los supervisores; si van a registrar un trabajador desde su computador o tablet, debe haber manera de asegurarse que el que usa la aplicación es el supervisor. Si no se autentica que el usuario es realmente un trabajador o un supervisor, se corre el riesgo de registrar un trabajador en cualquier momento desde cualquier lugar y se puede usar como excusa para generar problemas si no era el turno real del trabajador o incluso se puede llenar y bloquear la base de datos con peticiones falsas de registro de trabajadores.
* **Integridad:** se espera que los datos de usuarios, autenticación y permisos no puedan ser modificados por personal no autorizado. Esto es importante, dado que se quiere restringir el acceso a cierta información y que solo los usuarios puedan acceder a su información o a la que tengan permiso. De no garantizar este requerimiento, se podrían llegar al caso en que la información de autenticación de los usuarios sea cambiada, lo que llevaría a suplantaciones. De igual manera, no garantizar que lo permisos son inalterables por personal no autorizado, llevaría a que los usuarios no permitidos pudieran ver información otros. En particular, podría llevar al caso en el que trabajadores que no son supervisores, pudieran ver información de los demás trabajadores.

# Vulnerabilidades

A continuación, se listas las vulnerabilidades que se considera, tiene el sistema. De igual manera se argumenta su importancia dentro del sistema.

1. Ataque de a los servidores con exceso de peticiones, lo que desencadena en una negación de servicio (DoS). Es decir, que los servidores donde se encuentre desplegada la aplicación *Time & Attendance,* sean atacados y no puedan prestar sus servicios, como autenticación de usuarios y registro de llegadas y salidas de trabajadores. Es importante considerar este ataque dado podría desencadenar en que el sistema deje de funcionar por un periodo de tiempo, contrario a lo que se espere, que este siempre disponible.
2. Otra vulnerabilidad es que dependiendo del algoritmo escogido para generar llaves (ya que de este depende el tamaño de la llave), la información cifrada con esas llaves puede ser más susceptible a modificarse si se intercepta la comunicación (man in the middle). Lo ideal es que sólo se manejen llaves de gran tamaño, pero esto ya depende del tiempo esperado para realizar operaciones en la aplicación; un algoritmo asimétrico para descifrar y cifrar que usa llaves grandes tarda más tiempo que si se usan algoritmos simétricos.
3. Puede suceder que, si el sistema está interceptado por un man in the middle, el cliente envíe código malicioso al servidor o algún tipo de SQLInjection a través del reto. Esto pues para obtener el reto, se descifra con una llave simétrica que se obtiene con la llave pública del cliente; es decir, el reto se puede conocer fácilmente si se logra interceptar la comunicación entre el cliente y el servidor. Siguiendo el protocolo, el reto que envía el cliente al servidor es cifrado con la llave pública del servidor y de nuevo, si el man in the middle pudo interceptar el envío del certificado del servidor al cliente, pudo extraer la llave pública del servidor.
4. Similar a la vulnerabilidad anterior, si un atacante toma control del servidor, puede utilizarlo para robar información que reciba de los clientes, como sus identificadores de usuario, llave pública o incluso enviar malware a través del reto. Todo esto se traduce en que el servidor podría convertirse en una herramienta útil para el atacante, mediante la cual puede llegar a afectar a muchos usuarios dependiendo del uso que le dé al servidor secuestrado. Por ejemplo, puede hacer criptojacking con el servidor o pedir dinero a cambio de la devolución del servidor.