ESTUDIO DE POLÍTICAS DE SEGURIDAD PARA LA ELABORACIÓN DE SOFTWARE.

CÉSAR ALBERTO MORENO ABAUNZA

UNIVERSITAT OBERTA DE CATALUNYA

MASTER UNIVERSITARIO EN SEGURIDAD DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y DE LAS COMUNICACIONES

ESPAÑA

2018

ESTUDIO DE POLÍTICAS DE SEGURIDAD PARA LA ELABORACIÓN DE SOFTWARE.

Estudiante:
CÉSAR ALBERTO MORENO ABAUNZA

Trabajo final de master para optar al título de
MASTER UNIVERSITARIO EN SEGURIDAD DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y DE LAS COMUNICACIONES

Director
Jorge Chinea LÓPEZ

UNIVERSITAT OBERTA DE CATALUNYA
ESPAÑA
2018

CONTENIDO

1. INTRODUCCIÓN 8
2. OBJETIVOS 9
   2.1. Objetivo General 9
   2.2. Objetivos Específicos 9
   2.3. Solución a los objetivos específicos 9
3. MARCO TEÓRICO 10
   3.1. Política de Seguridad. 10
   3.2. Computación en la nube. 10
4. ESTADO DEL ARTE 12
   4.1. Ataques Cibernéticos. 12
   4.2. Atributos de un modelo de seguridad de Software. 13
      4.2.1. Confidencialidad 13
      4.2.2. Integridad 13
      4.2.3. Disponibilidad 13
5. ANÁLISIS 14
   5.1. Situación actual en las organizaciones. 14
   5.2. Uso de técnicas criptográficas. 14
   5.3. Aplicaciones seguras 15
6. ESTUDIO DE POLÍTICAS DE SEGURIDAD 16
   6.1. POLÍTICAS DE SEGURIDAD PARA LA PROTECCIÓN DEL PUESTO DE TRABAJO. 16
      6.1.1. Seguridad física 16
      6.1.2. Servidor de dominio Windows. Active Directory. 17
      6.1.3. Protección contra Malware 18
   6.2. POLÍTICAS DE SEGURIDAD PARA EL ALMACENAMIENTO EN LA NUBE. 20
      6.2.1. Seguridad de transporte de la información. 20
      6.2.2. Control de acceso a la nube. 22
      6.2.3. Copias de seguridad de la información en la nube. 22
   6.3. POLÍTICAS DE SEGURIDAD EN EL USO DE TÉCNICAS DE CRIPTOGRAFÍA. 23

6.3.1. Certificados electrónicos. 23
6.3.2. Autenticación mediante contraseñas encriptadas. 25
6.3.2.1. Sodium Crypto Library (Libsodium). 26
6.3.3. Criptografía mediante el uso de Llave Pública 29
6.3.3.1. Public-key authenticated encryption 29
6.3.3.2. Public-key signatures 31
6.3.4. Validación de datos de entrada en la aplicación. 32
6.4. POLÍTICAS DE SEGURIDAD PARA LA GESTIÓN DE LOGS. 33
6.4.1. Captura y generación de Logs. 33
6.4.1.1. Logs de accesos de usuarios al sistema. 33
6.4.1.2. Logs de acceso privilegiado o administrativo. 34
6.4.1.3. Logs de accesos remotos. 34
6.4.2. Respaldo, retención y restauración de Logs. 34
6.4.3. Seguridad y resguardo de los LOGS. 35
6.5. POLÍTICAS DE SEGURIDAD PARA EL CONTROL DEL ACCESO EN LOS SISTEMAS DE INFORMACIÓN. 37
6.5.1. Conceptos básicos de un sistema de control de acceso. 37
6.5.2. Control de acceso basado en roles. 38
6.5.3. Control de acceso con Middleware 39
7. CONCLUSIONES 42
8. BIBLIOGRAFÍA 43

LISTA DE FIGURAS

Figura 1 Ejemplo de organización geográfica por zonas de intereses por árbol y dominio. 18
Figura 2 Proceso de validación en la transmisión de un mensaje en protocolo TLS 21
Figura 3 Diagrama de una conexión VPN entre proveedor Cloud y Red Interna..... 21
Figura 4 Infraestructura de Clave Pública – PKI 25
Figura 5 Elementos básicos de un sistema de control de acceso 37
Figura 6 Ejemplo de Control de Acceso basado en Roles 39
Figura 7 Ejemplo de Middleware en Peticiones al Software..... 40

RESUMEN

TÍTULO: ESTUDIO DE LAS POLÍTICAS DE SEGURIDAD PARA LA ELABORACIÓN DE SOFTWARE

AUTOR: MORENO ABAUNZA, César Alberto

PALABRAS CLAVE: Seguridad informática, Políticas de Seguridad, Software.

DESCRIPCIÓN: En la actualidad, existen varios factores que afectan en gran medida la seguridad informática de las empresas que manejan algún tipo de sistema de información y/o redes de informática. Desde organizaciones de TI hasta organizaciones de cualquier actividad económica, las bases de datos y los sistemas de información que se manejan son muy vulnerables a ataques para robo de información para actividades de enriquecimiento personal, o sencillamente para la eliminación de la misma. El problema está en que estas empresas se enfocan en su naturaleza económica y dejan a un lado la seguridad de sus sistemas, ya que se ve una escasez de políticas de seguridad, tanto para los sistemas de información, como para las redes de datos y los servidores donde se encuentran alojados los sistemas de información.

En este trabajo de master se estudiarán políticas de seguridad que se deben tener para la elaboración de Software, partiendo desde la seguridad en la infraestructura donde se alojan los servidores que contiene el Software, hasta las políticas de seguridad de los puestos de trabajo.

ABSTRACT

TITLE: STUDIES OF SECURITY POLICY FOR SOFTWARE DEVELOPMENT

AUTHORS: MORENO ABAUNZA, César Alberto

KEYWORDS: Computer security, security policy, software

DESCRIPTION: Nowadays there are several factors that can generate on a large scale the computer security of enterprises that handles some kind of information systems and/or computer networks. From the organization of IT to any economic activity, data bases and information systems they handle are too vulnerable to attacks of stealing information for personal enrichment activity or just the deletion of it. The problem is that these enterprises are focused in their own economic nature and they put aside their own security systems, because there is a lack of security policy, both for information systems as for data network, and the servers that lodges information systems.

The current master paper, will be studying security policy that should be taking into account for software elaboration, starting from the infrastructure's security that lodges servers that contains the software, to security policy of work places.

1. INTRODUCCIÓN

Hoy en día, la seguridad de la información se ha convertido en un tema que nadie debería ignorar, pero que muchos lo dejan como un tema sin mayor importancia. Vivimos en una sociedad donde todo lo que nos rodea está conectado a internet, lo cual nos facilita hacer tareas del día a día que jamás podríamos imaginar. Pero esto trae consigo muchos peligros, y poner en riesgo la información personal, y hasta nuestra integridad, si no tenemos controles mínimos de seguridad en nuestros equipos personales, de la empresa y de nuestro hogar.

Estamos llegando a una era donde los robos, pérdida de información, secuestros de identidad, no se harán de forma física, sino que se hará de manera virtual. No necesitamos tener al delincuente al frente de nuestros ojos para que esto suceda, simplemente que una vulnerabilidad informática que puedan explotar, tendrá todo lo necesario para tener la información a un clic de distancia.

De acuerdo a lo anterior, las empresas deben garantizar a sus clientes, que su información está protegida, que cuenta con políticas de seguridad en sus redes, servidores, bases de datos, y puestos de trabajo, para evitar, en un porcentaje muy alto, que estos sistemas sean vulnerados por hackers informáticos.

La seguridad en las redes, la seguridad física, la seguridad de los servidores, la seguridad de los datos, la seguridad en el ambiente de trabajo, todos son requisitos básicos, para tener un Software seguro. Implementar políticas de seguridad de Software es uno de los principales requisitos técnicos dentro de una organización para garantizar los sistemas de información seguros. Esto implica principalmente políticas de control de acceso, auditoría de sistemas, copias de seguridad, uso de técnicas criptográficas, entre otros. Pero la seguridad del Software, no depende exclusivamente de las políticas mencionadas anteriormente. También se requiere de políticas de seguridad para los servidores, redes de datos, y políticas de seguridad para la organización y sus empleados, por ejemplo, en la protección del puesto de trabajo, seguridad en el almacenamiento en la nube, entre otros.

2. OBJETIVOS

2.1. Objetivo General

Estudiar las políticas de seguridad para la elaboración de un Software en una organización.

2.2. Objetivos Específicos

- Proteger los puestos de trabajo de los funcionarios una organización evitando la propagación de virus y troyanos que afecten la red interna de la empresa como también el acceso sin autorización a los equipos de cómputo.
- Prevenir la perdida de archivos planos que se manejan dentro de la organización.
- Proporcionar a las organizaciones, sistemas de información con código seguro mediante la validación de datos de entrada y uso de técnicas de programación segura.
- Tener un control de logs de auditoría para monitorear las actividades que se ejecuten en los sistemas de información.
- Blindar los sistemas de información y bases de datos para evitar el acceso de terceros.

2.3. Solución a los objetivos específicos

- Implementar políticas de seguridad para la protección del puesto de trabajo.
- Implementar políticas de seguridad para el almacenamiento en la nube.
- Implementar políticas de seguridad en el uso de técnicas de criptografía.
- Implementar políticas de seguridad para la gestión de Logs.
- Implementar políticas de seguridad para el control del acceso en los sistemas de información.

3. MARCO TEÓRICO

3.1. Política de Seguridad.

Una política de seguridad es el conjunto de reglas y prácticas que definen y regulan los servicios de seguridad de una organización o sistema con el propósito de proteger sus recursos críticos y sensibles. En otros términos, es la declaración de lo que está permitido y lo que no está permitido hacer. [1]

La política de la seguridad es la base de la seguridad de un sistema. En ella es donde detallamos los servicios de seguridad de los sistemas, en cada uno de los niveles en donde debe intervenir, se determina lo que se puede y no se puede hacer con los recursos del sistema, y quien lo puede hacer, se establecen permisos de acceso a cada uno de los recursos. En algunos casos estas políticas no se tienen de declarar con lenguaje formal, a veces basta con dar unas directrices sobre seguridad con lenguaje informal.

3.2. Computación en la nube.

Una definición cercana es la de permitir al cliente abstraerse del hardware en el cual va a trabajar y centrarse en los servicios que necesita para realizar dicho trabajo. De esta manera para el usuario es totalmente transparente la infraestructura que necesita para trabajar.

Algunas de las características de la computación en la nube son:

- Reducción de costos: Existen empresas dedicadas a rentar hardware a los usuarios interesados en ejecutar sus aplicaciones en la nube, y estos no necesitan centrar su atención en los gastos de compra de equipos ni a gastos en mantenimiento. Es cierto que hay una inversión para la renta de esta infraestructura, pero es menor a los gastos que acarrearía a corto o largo plazo.
- Elasticidad y escalabilidad: Las aplicaciones que están en la nube son totalmente elásticas en cuanto a su sencillez de implementación. También se vuelven escalables, ya que se puede modificar de manera casi inmediata los

[1] R. Shirey. Internet Security Glossary, RFC 2828, IETF. Disponible en línea en: http://www.ietf.org/rfc/rfc2828.txt

recursos que el sistema necesite en cualquier circunstancia que así lo amerite.

- Copias de seguridad: Las empresas que se dedican a la computación en la nube le proporcionan al usuario sistemas de almacenamiento en servidores secundarios, de tal manera que si se produce alguna pérdida de información, estas pueden recuperarla de forma inmediata.
- Seguridad: A pesar de que la información se maneja “en la nube”, y esto permite que la información de los usuarios pueda estar en manos de terceros, las compañías de computación en la nube mantienen estrictos niveles de seguridad para evitar el acceso a la misma.

4. ESTADO DEL ARTE

4.1. Ataques Cibernéticos.

Para nadie es un secreto que internet se ha convertido hoy en un parte de nuestras vidas, sin él es difícil cumplir muchas de nuestras actividades diarias. Lo usamos en el hogar, en la oficina, en la universidad, en el colegio, en el autobús, en el mercado, en casi todos los lugares que visitamos. El internet tiene diferentes usos, mantenernos comunicados con las personas que están en diferentes lugares, en distintos momentos, mantenernos informados con noticias actualizadas, buscar un lugar para ir a almorzar, entre muchas otras funciones. Hasta acá todo es avance en tecnología, pero este avance trae consigo una amenaza en la privacidad, identidad, datos confidenciales, etc. Si antes teníamos la información escrita en un papel en nuestro bolsillo del pantalón, la única forma de que alguien leyera esta información era hurtando el papel. Con la información en internet no tienen que venir hasta nuestra casa para leer la información.

Para poder luchar contra los ciberataques, un tema que causa bastantes dolores de cabezas tanto para las empresas grandes, medianas o pequeñas, como a las personas en general, los gobiernos y las organizaciones hacen lo posible para estar actualizados con parches de seguridad y usar herramientas y técnicas de seguridad informática y así tener protegidos sus datos y mantener en funcionamiento la institución.

Denegación de servicios, DoS, es uno de los ataques más conocidos en internet, el cual ataca una red o un sistema informático, suspendiendo temporalmente la disponibilidad de la red o la función del sistema informático, de tal modo que deniega el acceso a los usuarios al sistema.

Troyano, es código maligno que infecta archivos del sistema, que busca abrir una backdoor, o puerta trasera, para facilitar el acceso de manera remota a equipos dentro de una red informática.

Ransomware, es uno de los ataques más sofisticados y modernos que se utilizan hoy en día y más con la entrada al mercado de las criptomonedas. Consiste en un Malware que secuestra los datos de un equipo (encriptándolos), y pide un monto económico para poder recuperarlos. Hace poco estuvo en circulación el ransomware llamado “WannaCry”, el cual afecto grandes multinacionales en todo el mundo. [1]

4.2. Atributos de un modelo de seguridad de Software.

Cuando se habla de atributos de seguridad que debe tener un Software, nos referimos a 3 atributos principales, los cuales que deben garantizar siempre: La confidencialidad, la Integridad y la disponibilidad.

4.2.1. Confidencialidad

La confidencialidad se refiere a que, en una actividad de la aplicación, el contenido de la actividad no se revela a un agente no autorizado. El término de confidencialidad de acuerdo al contexto en el que se use puede variar. La confidencialidad del remitente se refiere a que al actor que ejecuta el evento debe mantener en secreto el contenido durante la ejecución del evento. La confidencialidad de los datos se refiere a que los datos no pueden ser revelados por ningún tercero mientras se transmite la información, por lo tanto estos deben mantenerse en secreto y con un nivel de seguridad alto (por ejemplo con criptografía).

4.2.2. Integridad

El atributo de Integridad se refiere a la calidad de la ejecución de una actividad, esto quiere decir que la actividad debe desarrollarse de acuerdo a un procedimiento establecido, cumpliendo con los estándares que se definieron por la aplicación. La integridad del actor del sistema de información se refiere a que este debe actuar de acuerdo a las especificaciones propuestas por el arquitecto del software. La integridad de los datos se refiere a que no se debe permitir en ningún momento de la ejecución la manipulación o destrucción de los datos, cuando no se ha previsto en su funcionamiento.

4.2.3. Disponibilidad

El atributo de disponibilidad en una aplicación se refiere a que una actividad se debe completar durante un tiempo determinado. Si nos referimos a la disponibilidad de los datos, esto significa que la información debe estar disponible únicamente por el actor autorizado, pero a su vez esta información no se puede negar en caso de tener que ser usados por terceros.

5. ANÁLISIS

5.1. Situación actual en las organizaciones.

Si nos detenemos que revisar el estado actual de las Organizaciones, por ejemplo en Colombia, se evidencia que en la mayoría de las empresas, en 2016, más del 80% poseen sistemas altamente vulnerables[2], ya que no invierten en el tema de seguridad informática. Y si se entra a evaluar a fondo, el otro 20% restante, pertenece a empresas Gubernamentales, las cuales si invierten en su seguridad, de tal manera que los datos e información personal que alojan no sea robada por hackers.

Los sistemas de información se las organizaciones puede tener un nivel de seguridad en temas de control de acceso y código seguro, pero en la mayoría de los casos el acceso a la información se da por infección de Malware en los equipos de las organizaciones, lo que hace más sencillo el acceso a los servidores internos de la compañía, cuando ya se puede tener acceso a la red interna de la organización. Por eso en este trabajo de Maestría uno de los estudios que se va a realizar es acerca de la implementación de políticas de seguridad para la protección del puesto de trabajo.

La seguridad no solamente es protegernos de los hackers, también es proteger los datos de tal manera que evitemos la pérdida de información por otro tipo de incidentes. Es la mayoría de empresas, la información siempre permanece en servidores de almacenamiento que se encuentran dentro del mismo espacio físico, lo cual es un riesgo cuando no contamos con políticas de seguridad para las redes y los servidores. Para esto se recomienda usar las nuevas tecnologías que nos ofrecen hoy grandes empresas de TI, con soluciones como Google Cloud, o Amazon Drive. Pero no solo es tener nuestra información en la “Nube”, también debemos contar con políticas de seguridad para el almacenamiento en la Nube, otro tema que se estudiará en el presente proyecto.

5.2. Uso de técnicas criptográficas.

Las organizaciones cuentan con información sensible y confidencial como son las claves de acceso a los sistemas de información, información con protección legal, credenciales para correos electrónicos, etc.y esto significa que está información

[2] http://caracol.com.co/radio/2016/06/09/tecnologia/1465469190_389745.html

debe, además de estar en un sitio seguro, estar protegida con técnicas de criptografía, desde el tránsito de la información hasta su almacenamiento.

En este proyecto se estudiaran las políticas de seguridad para el uso de técnicas criptográficas, desde librerías que se pueden usar en lenguajes de programación para cifrar los datos, de tal manera que sean ilegibles por terceros o personas que no tengan la llave de cifrado, garantizando dos atributos fundamentales en es un modelo de seguridad de Software como es la confidencialidad y la integridad.

5.3. Aplicaciones seguras

Cuando hablamos del término “Aplicaciones seguras”, abarca muchas temas, pero en este proyecto tocaremos dos temas muy interesantes que son el Control de Acceso a las aplicaciones y la gestión de Logs.

A lo largo del tiempo, el control de acceso ha sido y siempre será un logro dentro de las compañías en general, ya que es el primer filtro en cualquier sistema de información de cara al usuario final. El control de acceso es el proceso por el cual, dada una petición de recursos, se permite o niega el acceso a los mismos en base a la aplicación de unas políticas de acceso.

Como es el primer filtro al acceso a los sistemas de información, se deben tener políticas de seguridad que garanticen las reglas de alto nivel para regular el acceso a los recursos del sistema de forma abstracta.

Y de la mano al control de acceso se debe contemplar en las aplicaciones un mecanismo que garantice el registro de todas las actividades de los usuarios y sus procesos dentro de la aplicación. Si se desea tener un mayor control de las actividades de los usuarios y poder realizar un informe de auditoría, se debe contar con políticas de gestión de Logs, tema que abarcaremos con mayor profundidad en el presente proyecto.

6. ESTUDIO DE POLÍTICAS DE SEGURIDAD

6.1. POLÍTICAS DE SEGURIDAD PARA LA PROTECCIÓN DEL PUESTO DE TRABAJO.

Antes de hablar de las políticas de seguridad para el desarrollo de Software, debemos entender que, el acceso a la red y a los sistemas de información en una organización, inicia desde el puesto de trabajo de cada funcionario.

Si bien es cierto, la mayoría de los equipos de la compañía, se encuentran en la red interna de la Organización, y por ende, tienen acceso de alguna manera a los servidores donde se alojan los sistemas de información y bases de datos. Esto da la posibilidad de tener una puerta de acceso para los hackers, y poder llegar a tener información privilegiada, o usar nuestros servidores para satisfacer necesidades de atacar a otras compañías.

Para evitar este tipo de inconvenientes que se pueden presentar en cada uno de los puestos de trabajo de los funcionarios de una organización, se deben establecer políticas de seguridad para la protección del puesto de trabajo.

6.1.1. Seguridad física

Uno de los puntos importantes dentro de una política de seguridad de la protección del puesto de trabajo, es la seguridad física. De nada nos sirve tener los sistemas más robustos, los firewalls más potentes, si no contamos con medidas de seguridad física básicas. Con el solo hecho de permitirle a un ladrón, tomar un computador de una compañía, este podría estar robando claves de acceso, información con contenido reservado, lo que generaría un problema de seguridad en la compañía.

Por ejemplo en el caso de los equipos de cómputo de los empleados, estos deben contar con guayas de seguridad que se sujeten a la mesa de trabajo, y cuenten con una clave difícil de descifrar. Las puertas donde se alojen estas oficinas siempre deben permanecer cerradas con llave, y solamente una persona puede ser la autorizada para abrir y cerrar esta puerta.

Para el caso de los servidores, donde se alojan los sistemas de información, y las bases de datos, estos deben contar con una medida de seguridad más

especializadas, por ejemplo debe estar en un cuarto diferente a los que visitan frecuentemente los empleados, deben tener una cerradura con llave, y sistema de refrigeración las 24 horas del día durante los 365 días del año. Adicionalmente, solamente un funcionario, o dependencia, debe ser la autorizada para ingresar a este Data Center.

Estas políticas de seguridad, aunque parecen obvias, no todas las compañías las ponen en práctica, y casi siempre se las cosas más sencillas, es que pueden ocurrir los grandes desastres.

6.1.2. Servidor de dominio Windows. Active Directory.

Una de las políticas de seguridad en los puestos de trabajo, es contar con un método de autenticación al iniciar sesión cuando vamos a usar nuestro equipo después de haber estado un tiempo inactivo. Por ejemplo cuando tenemos un receso en el horario laboral, cuando vamos a almorzar, o cuando culminamos nuestra jornada laboral.

En cualquier de estos casos es necesario cerrar sesión en nuestros equipos, de tal manera que si un tercero intenta acceder a la información que tenemos en este, o intenta eliminarla, no la tendrá tan sencilla, lo digo así porque existen otras maneras más especializadas de saltarnos estos inicios de sesión; pero podemos empezar por lo principal que es siempre dejar nuestra sesión cerrada.

Una herramienta para garantizar esta política de seguridad, es el uso de un servidor de dominios, el cual consiste en un servidor de la red que permite controlar todos los equipos y dispositivos que forman parte de esta red, de tal forma que solamente los usuarios que se encuentren creados en este servidor, puedan estar autorizados para ingresar a los equipos.

Para configurar este servidor de dominios como un controlador de dominio, se debe instalar, en el caso de Servidores con Windows Server, el rol de Active Directory, el cual es nada más que un almacén de datos estructurado como base de una organización lógica y jerárquica de la información del directorio.

El objetivo de tener un servidor de dominios dentro de una compañía es la de garantizar que la autenticación de usuarios a los equipos de la red se realice pasando un filtro de seguridad que se encuentra en el rol del Active Directory del servidor, y así en cualquier momento poder bloquear, cambiar contraseñas de los

usuarios, sin tener que ir hasta el puesto de trabajo. También tenemos la posibilidad de que los administradores puedan administrar datos del directorio con un único inicio de sesión a la red y a su vez, los usuarios autorizados de la red puedan tener acceso a recursos en cualquier lugar de la misma.

Además de tener un control de autenticación, podemos controlar permisos de acceso a unidades de red, de tal manera que se gestionen grupos de usuarios con distintos permisos, y manejar una jerarquía de la organización.

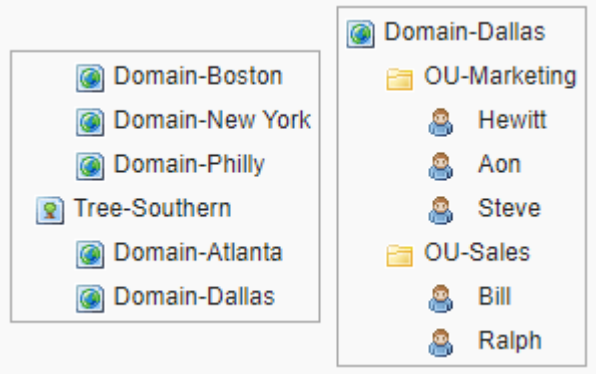

Figura 1 Ejemplo de organización geográfica por zonas de intereses por árbol y dominio.

6.1.3. Protección contra Malware

Uno de los casos más comunes que ponen en riesgo la seguridad de la información en una organización es la propagación de Malware dentro de la compañía. Y se podría empezar por explicar que es un Malware[3]. Malware es un tipo de software intrusivo y hostil que tiene como objetivo infiltrarse o dañar un sistema de información sin la aprobación ni el consentimiento de su propietario.

De acuerdo al daño y a su forma de operar, los Malware tienen 3 grandes categorías. Malware de propagación automática, donde su finalidad es la de extenderse por toda la red de forma automática infectando a todos los equipos que se encuentren dentro de la red. Este tipo de Malware, por lo general infecta los equipos mediante archivos ejecutables, donde muchas veces son archivos PDF enviados como si fueran información referente al trabajo, y los funcionarios por desconocimiento de los mismos, los abren e infectan el equipo.

[3] https://techterms.com/definition/malware

Otra categoría de Malware es el Oculto. Es un tipo de software malicioso que se caracteriza por intentar permanecer desapercibido para el usuario dentro del sistema infectado. En esta categoría están los troyanos, los cuales se camuflan en programas supuestamente legítimos, pero que llevan código malicioso oculto, que permite el control remoto de la máquina afectada, enviando información a un sitio donde el atacante lo haya configurado. Muchas veces los hackers usan técnicas de ingeniería social, para hacer que la víctima de la compañía ejecute el software que contiene este Malware oculto.
La última categoría sería la del Malware lucrativo. Este es uno de los más llamativos en la comunidad hacker, ya que su finalidad no es necesariamente la de dañar los sistemas de información y/o tener acceso a la misma, si no la de proporcionar algún tipo de beneficio al atacante. Uno de los Malware más comunes en esta categoría son los Ransomware. Es un tipo de malware que extorsiona a la víctima de la máquina afectada, exigiendo algún tipo de pago tras cifrar los archivos. La victima una vez genera el pago, comúnmente con BitCoins, el sistema automática desencripta la información. Uno de los Ransomware más común en los últimos años, fue el que infectó a la compañía Telefónica y SONY, y otras más en Europa, el WannaCry[4].

Una vez entendido el concepto de Malware, una de las políticas de seguridad para prevenir este tipo de ataques es mantener actualizados los parches de seguridad de los sistemas operativos que se manejen en la compañía, actualizar de manera automática el antivirus en cada equipo; implementar el uso de herramientas que impidan ejecutar software de fuentes desconocidas, al igual que usar firewall en el servidor de correos para que avise a los empleados cuales contienen malware.

[4] https://www.avast.com/es-es/c-wannacry

6.2. POLÍTICAS DE SEGURIDAD PARA EL ALMACENAMIENTO EN LA NUBE.

Uno de los temas críticos dentro de una organización es la información. Sin la información cualquier empresa se vendría en quiebra, ya sea por perder sus registros contables, inventario de mobiliario, registro de los procesos jurídicos, etc. Para ello la información siempre debe estar almacenada en un sitio seguro, y en la última década, se ha implementado en las organizaciones el almacenamiento de esta información en la nube. Pero no basta con que la tengamos en la nube. Debemos tener unas políticas de seguridad para evitar dolores de cabeza en un futuro.

Es importante señalar que junto al avance de las tecnologías de almacenamiento en la nube, han aparecido nuevas vulnerabilidades y amenazas de seguridad. Entre estas vulnerabilidades se encuentran las siguientes: violación de datos, pérdida de datos, denegación de servicios. Para ello, las organizaciones pueden implementar las siguientes políticas de seguridad para el almacenamiento en la nube.

6.2.1. Seguridad de transporte de la información.

En todos los servicios en la nube, la comunicación de nuestros sistemas de información con los del proveedor de servicio en la nube, se hace a través de la red. Desde el registro de un nuevo campo en la base de datos, hasta el guardado de un archivo PDF. Esto significa, que esta comunicación al proveedor de servicios en la nube es susceptible a ataques informáticos, para obtener credenciales de acceso de usuarios, manipular contenido de la información, entre otros.

Una de las políticas de seguridad que se debe tener en cuenta de la transmisión de datos en nuestros sistemas de información, hacía el almacenamiento en la nube es la de implementar protocolos criptográficos que proporcionen autenticación y cifrado de la información entre los servidores. Son más comunes son SSL y TLS, pero con el tiempo el protocolo SSL, ha quedado obsoleto por la cantidad de vulnerabilidades que se han encontrado. Por eso la mejor opción es TLS.

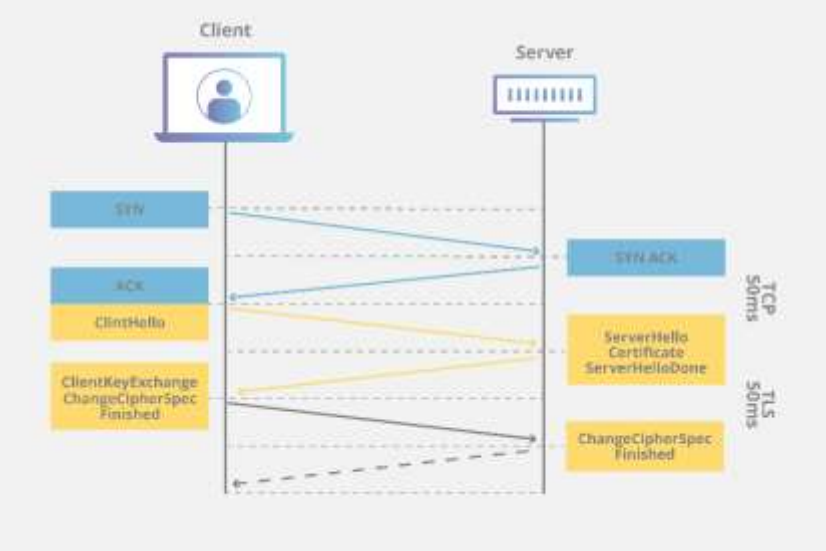

Figura 2 Proceso de validación en la transmisión de un mensaje en protocolo TLS

Otra política de seguridad muy importante en el transporte de la información hacia la nube, es el uso de una VPN[5]. Un servicio de VPN Cloud de acuerdo al servicio de Google Cloud VPN, consiste en conectar de manera segura la red local de la compañía donde se tienen los sistemas de información y red de oficina, con la red de nube del proveedor que tengamos contratado para el almacenamiento, a través de una conexión VPN[6]. Esto garantiza que toda la transmisión de datos que se haga desde la red interna a la nube este encriptada por una puerta de enlace VPN y luego descifrada de la misma manera.

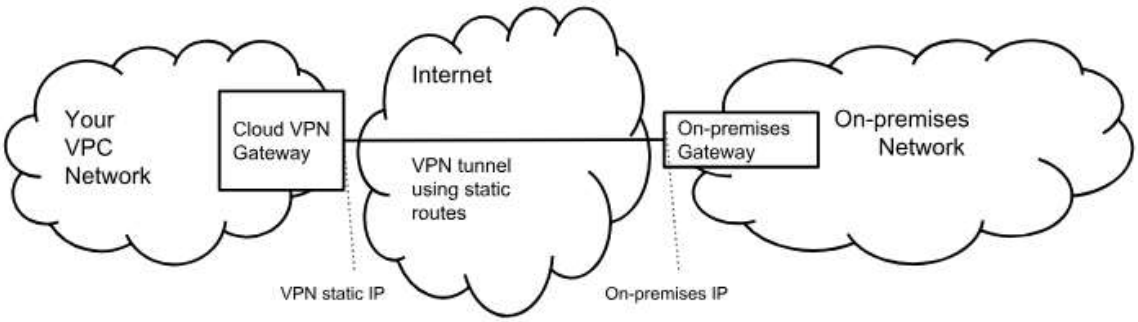

Figura 3 Diagrama de una conexión VPN entre proveedor Cloud y Red Interna

[5] https://en.wikipedia.org/wiki/Virtual_private_network
[6] https://cloud.google.com/vpn/docs/concepts/overview

6.2.2. Control de acceso a la nube.

Al momento de crear una instancia en la nube con un proveedor, es necesario crear unas credenciales de acceso remoto, la más común por acceso SSH, para la administración e intercambio de archivos. Estas credenciales de acceso son un punto vulnerable para un sistema en la nube, y es punto vital que debemos tener en cuenta con una política de seguridad.

Si bien es cierto que podemos tener una contraseña robusta, a las cuentas de usuarios que tengamos para el acceso a la nube, no siempre es la mejor opción, ya que mediante ataques de fuerza bruta, un análisis del propietario con ingeniería social y un buen diccionario podrían romper esta contraseña para tener el control de nuestra máquina en la nube.

Para tener un control de acceso robusto, lo podríamos hacer mediante la creación de claves públicas y privadas SSH.

La mayoría de proveedores de computación en la nube, ofrecen a sus clientes un firewall perimetral, el cual mediante reglas el administrador de la máquina virtual puede proteger su aplicación. Por defecto este firewall bloquea todos los puertos para evitar ataques por puertos vulnerables. Una buena política de seguridad para el acceso a la nube, es crear reglas en este firewall donde se indique que grupo de IP’s pueden tener acceso mediante algún puerto en específico, Ej Puerto 22 para SSH.

6.2.3. Copias de seguridad de la información en la nube.

Una de las razones por las que tenemos la información en la nube, es para garantizar la alta disponibilidad de la información, además que tener una escalabilidad para nuestra empresa. Pero como todo, hasta las mejores infraestructuras en el mundo puede fallar. Por lo tanto una política de seguridad que se debe manejar dentro de las compañías, es la de realizar copias periódicas automatizadas, mediante protocolos seguros, como los visto anteriormente, a un servidor que destinemos en nuestra organización, para tener un respaldo de nuestros datos, en caso de tener que realizar algún tipo de contingencia o recuperación de la misma.

6.3. POLÍTICAS DE SEGURIDAD EN EL USO DE TÉCNICAS DE CRIPTOGRAFÍA.

La identificación digital forma parte de la mayoría de servicios que hoy en día se manejan dentro de cualquier organización que tenga acceso a internet. Para acceder al sistema de información de la compañía, o para realizar una llamada por VozIP, para realizar un pago con nuestra tarjeta de crédito, usando una pasarela de pagos, entre otros muchos otros procesos.

Pero suena bastante sencillo, hablando en términos de interfaz gráfica. Y es aquí donde se da inicio al uso de técnicas que garanticen que el servicio lo está ejecutando el usuario que dice ser, y esto se logra mediante la "autenticación de la identidad".

A partir de este capítulo de este proyecto, se empiezan a tocar temas propios de las fases de desarrollo de un Software seguro que cumpla con políticas de seguridad robustas, empezando con el uso de técnicas de criptografía. En cada una de las técnicas que se van a ver a continuación, se va a demostrar cómo deben cumplir y garantizar la confidencialidad, integridad, autenticidad y no repudio de la información que se considere sensible dentro de la compañía.

6.3.1. Certificados electrónicos.

La firma electrónica o certificados electrónicos, son utilidades que se deben implementar como una política de seguridad dentro de las organizaciones en el uso de sus sistemas, ya que garantizar la autenticación de identidad con el uso de sistemas de criptografía de clave pública.

Estos sistemas se basan en el uso de dos claves: la clave privada o "secret key", que solo conoce el propietario, encargado de los sistemas de información en la organización, y la clave pública o "public key", la cual como su nombre lo indica es la llave que pueden conocer cualquier agente sin tener alguna consecuencia en la seguridad del sistema. La clave privada o secreta, debe estar muy bien resguardada, ya que esta se utiliza para poder invertir la llave publica, y enviar el mensaje.

El uso de estas claves, depende del proceso que se quiera implementar. Por ejemplo que queremos dar **confidencialidad** a un mensaje que A envìo a B, sin

que un tercero pueda leer el mensaje, A usará la clave pública de B, y cuando B reciba su mensaje, B podrá utilizar su correspondiente clave privada para acceder a su contenido. Pero si lo que se quiere es dar **autenticidad** a un mensaje que A envía a B, es lo que se conoce como firma electrónica. A usa su clave privada para firmar el mensaje, cuando B recibe el mensaje, comprobará la validez de la firma, ya que conocerá la llave pública de A para poder realizar la inversa a la firma y verificarla.

Mediante el uso de firmas electrónicas, se garantizan 3 propiedades relacionadas con la seguridad informática: Se asegura que nadie ha modificado el mensaje firmado (**integridad del mensaje**), la que la modificación de modificación de un solo bit del mensaje, ocasionaría que no se pudiera verificar la misma; también se asegura que el mensaje efectivamente fue firmado por A y no por otro agente, lo que quiere decir que asegura la (**identidad del firmante**), ya que la operación la puede realizar únicamente el propietario de la clave privada; y por último como A es el único que conoce la llave privada, solo se puede demostrar que A fue quién firmo el mensaje, por lo tanto asegura la tercera propiedad de **no repudio** de la firma electrónica.

El objetivo de una política de seguridad en el uso de un "**Certificado electrónico",** es la dar fe de la relación de una clave pública a una identidad, como un usuario o un servicio. Actualmente existen varias compañías que emiten este tipo de certificados electrónicos que se les llama "**Autoridad de Certificación**", los cuales pueden expedir certificados de claves públicas. Estas autoridades disponen de una plataforma telemática (infraestructura de clave pública PKI por sus siglas en inglés **Public Key Infrastructure,** que permite generar y gestionar claves y certificados.

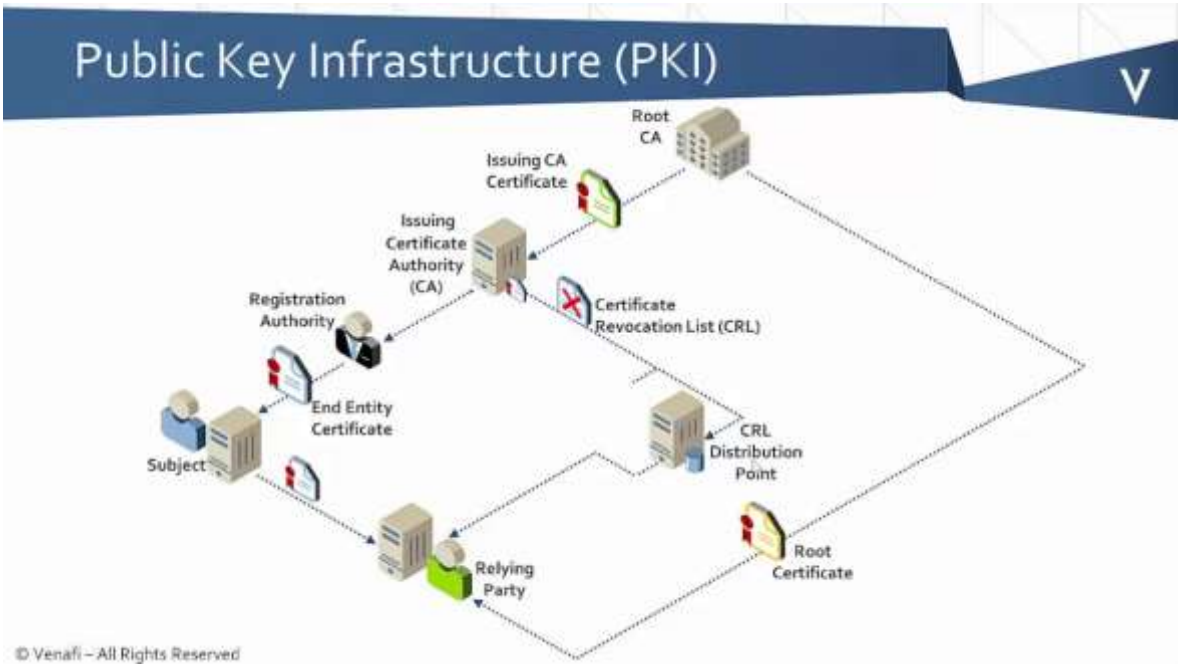

Figura 4 Infraestructura de Clave Pública – PKI

6.3.2. Autenticación mediante contraseñas encriptadas.

En los sistemas de información existen distintos métodos para el acceso al mismo, entre ellos encontramos métodos biométricos, como dispositivos de lectura de huellas dactilares, identificación a través del análisis de iris, rasgos faciales, voz, entre otros, pero en este caso en particular, hablaremos sobre la autenticación mediante el uso de una contraseña.

Si se va a dar de alta a un usuario que va a tener ciertos privilegios dentro de nuestra organización, debemos entender que existen varias estrategias que podrían utilizar los delincuentes informáticos para suplantar a un usuario, entre ellos ataques de fuerza bruta mediante combinaciones de caracteres aleatorios, o mediante la búsqueda de patrones (conociendo a la víctima) para adivinar su contraseña. Para ello siempre se recomienda que las contraseñas tengan cierto “nivel de seguridad”, combinando mayúsculas, con minúsculas, números y caracteres especiales, y sin tener un patrón de repetición o palabras que existan en un diccionario. También se puede evitar este tipo de ataques mediante políticas de seguridad en el código del Software, de las cuales hablaremos en el siguiente capítulo.

Independientemente de la rigurosidad de la contraseña, muchos ataques que se hacen a los sistemas de información no siempre son al usuario final del sistema de información, sino a la base de datos donde se encuentran alojadas estas contraseñas. Por lo tanto el administrador del sistema no debe guardar estas contraseñas tal cual como las ingresa el usuario final, sino la función criptográfica de resumen (hash), lo cual no es sino una secuencia de bits o clave secreta de longitud definida a partir de una secuencia de entrada y el uso de un "salt"[7].

Al momento de decidir utilizar técnicas de criptografía es bueno utilizar librerías criptográficas que nos permitan crear estos hash de nuestras contraseñas, pero es importante que no sean tan comunes, ya que existen funciones como SHA-1 o MD5 que por el tiempo que llevan en el mercado, ya existen varios diccionarios y exploits que ayudan a descifrarlas.

6.3.2.1. Sodium Crypto Library (Libsodium).

Sodium es una librería moderna y fácil de usar para el cifrado, descifrado, firmas digitales y funciones de criptografía de resumen de contraseñas. Su objetivo es la de proporcionar todas las funciones básicas necesarias para construir aplicaciones con criptografía de alto nivel.

Sodium es multiplataforma y multilenguaje lo cual permite que se ejecute en una gran variedad de sistemas operativos y compiladores Windows, Linux, iOS o Android[8].

Una de las funciones que tiene esta librería es la de "Password Hashing". Las características de esta función en particular, son las siguientes:

- La clave generada tiene un tamaño definido por la aplicación, sin importar la longitud de la contraseña del usuario.
- La misma contraseña con los mismos parámetros, siempre tendrá la misma salida.
- La misma contraseña, con distinto "salt", producirá salidas diferentes.
- La función que genera el hash a partir de la contraseña y el hash ingresado, requiere el uso de bastante CPU y buena cantidad de memoria, lo que mitiga ataques de fuerza bruta al intentar verificar cada contraseña.

[7] https://en.wikipedia.org/wiki/Salt_(cryptography)
[8] https://download.libsodium.org/doc/

Argon [2], es una familia de esquemas de funciones de resumen de contraseñas, ganadora del Password Hashing Competition (PHC) [2] en 2015, está optimizado para arquitecturas x86 y explota el uso de cache y memoria en los procesadores actuales de AMD e Intel.

Para entender cómo sería el proceso de la creación del hash de una contraseña en el desarrollo de un Software, colocare el siguiente ejemplo de una función que usa Argon2i para crear el hash, en lenguaje PHP.

```
string sodium_crypto_pwhash_str(string $password,int $opslimit,int $memlimit)
```

La anterior función crea un hash que podamos guardar en la base de datos del Software. Se recomienda que para el caso de $optlimit y $memlimit, se usen las constantes por defecto que tiene la librería.

Para un ejemplo práctico, una vez el usuario ingresa la contraseña, al momento de darse de alta en el sistema, el servidor debe realizar el siguiente procedimiento.

```
// Data from front-end
$password = filter_input(INPUT_POST,$_POST[‘passowd’] , FILTER_SANITIZE_MAGIC_QUOTES);

// hash the password and return an ASCII string
$hash = sodium_crypto_pwhash_str(
       $password,
       SODIUM_CRYPTO_PWHASH_OPSLIMIT_INTERACTIVE,
       SODIUM_CRYPTO_PWHASH_MEMLIMIT_INTERACTIVE
);
```

Creado el hash, este se almacena en la respectiva base de datos que se esté manejando y al momento en que el usuario desee iniciar sesión, basta con implementar el siguiente código:

```
// Data from front-end
$password = filter_input(INPUT_POST,$_POST[‘passowd’] , FILTER_SANITIZE_MAGIC_QUOTES);

if (sodium_crypto_pwhash_str_verify($hash, $password)) {
    // Se recomienda usar la function sodium_memzero, cada vez que se verifique.
    sodium_memzero($password);
    // Password was valid
} else {
    // Se recomienda usar la function sodium_memzero, cada vez que se verifique.
    sodium_memzero($password);
    // Password was invalid.
}
```

6.3.3. Criptografía mediante el uso de Llave Pública

Como se mencionó en el capítulo 6.3.1 existen certificados electrónicos para garantizar la autenticidad del mensaje, las llaves públicas como política de seguridad de la información es muy importante aplicarla también en la elaboración de Software, cuando se desea enviar mensajes de manera cifradas entre dos agentes dentro del mismo entorno. Esto quiere decir que los únicos que podrán leer un mensaje determinado, serán los que tienen la llave pública del emisor, y solo lo podrán hacer si el emisor cuenta con su llave privada.

6.3.3.1. Public-key authenticated encryption

Siguiendo con la teoría de que la llave privada solo la puede tener el agente emisor, de tal manera que no la puedan robar para crear mensajes auténticos, y basándonos en la librería Libsodium explicada en el capítulo 6.3.2.1, se explicarán otras funciones que permite en lenguaje PHP, enviar mensajes encriptados entre dos personas, garantizando la autenticidad, confiabilidad e identidad de la misma.

Para explicar las funciones, se hará un algoritmo práctico, en donde “Cesar” va a enviar un Mensaje a “Maria”. Antes que nada, para que Cesar y Maria logren comunicarse en privado y garantizar la confianza del mensaje, tienen que intercambiar sus claves públicas.

```
//Computador de Cesar

//Generamos un par de llaves aleatorias para que sea asignada como llave pública
y privada.

$cesar_box_keypair = sodium_crypto_box_keypair();

//Separamos la llave pública y privada con la función crypto_box

$cesar_box_secretkey = sodium_crypto_box_secretkey($cesar_box_kp);
$cesar_box_publickey = sodium_crypto_box_publickey($cesar_box_kp);
```

El mismo procedimiento se debe generar en el computador de Maria.

```
//Computador de Maria

//Generamos un par de llaves aleatorias para que sea asignada como llave pública
y privada.

$maria_box_kp = sodium_crypto_box_keypair();

//Separamos la llave pública y privada con la función crypto_box
```

```
$ maria_box_secretkey = sodium_crypto_box_secretkey($maria_box_kp);
$ maria_box_publickey = sodium_crypto_box_publickey($maria_box_kp);
```

Una vez fueron creadas en cada usuario las llaves públicas y privadas, la privada la debe almacenar cada usuario, sin que cualquier otra parte la pueda tener, y se intercambian las llaves públicas.

Una vez ambos tienen la llave pública de la contraparte, se procede a crear el mensaje y se envía. Cesar que va a hacer el emisor cifra el mensaje con la llave pública de Maria, y agrega una etiqueta de autenticación con su llave privada, de tal manera que cuando Marìa lo reciba, primero comprobará la integridad del mensaje, y luego si lo descifrará usando su propia llave privada.

```
// Computador de Cesar
$message = 'Hola Maria, este es mi mensaje. Soy Cesar';

// Creamos la llave conjunta para el envío del mensaje
$cesar_to_maria_kp = sodium_crypto_box_keypair_from_secretkey_and_publickey(
    $cesar_box_secretkey,
    $maria_box_publickey
);

//Creamos el valor nonce[9]
$nonce = random_bytes(SODIUM_CRYPTO_BOX_NONCEBYTES);

//Creamos el texto cifrado para enviar
$ciphertext = sodium_crypto_box(
    $message,
    $nonce,
    $cesar_to_maria_kp
);

//Computador de María
$maria_to_cesar_kp = sodium_crypto_box_keypair_from_secretkey_and_publickey(
    $maria_box_secretkey,
    $cesar_box_publickey
);

//Desciframos el mensaje de Cesar
$original_msg = sodium_crypto_box_open(
    $ciphertext,
    $nonce,
    $maria_to_cesar_kp
);
if ($original_msg === false) {
    // Malformed message
}else{
```

[9] https://en.wikipedia.org/wiki/Cryptographic_nonce

```
    echo $original_msg;
}
```

6.3.3.2. Public-key signatures

En algunas ocasiones los mensajes no necesariamente son end-to-end [10], pero si pueden requerir en una firma con llave pública, de tal manera que vayan firmados con la llave privada del emisor, y que cualquiera que tenga la llave pública pueda verificar la identidad del emisor y su autenticidad.

Para explicarlo de una menor manera, se hará un ejemplo práctico como el capítulo anterior.

Cesar desea crear un mensaje y ponerle su firma y enviarlo a todo el equipo de trabajo, y para ello crea un par de llaves y comparte la llave pública.

```
//Computador de Cesar

//Generamos un par de llaves aleatorias para que sea asignada como llave pública
y privada.

$cesar_sign_kp = sodium_crypto_sign_keypair();

//Separamos la llave pública y privada con la función crypto_sign

$ cesar_sign_secretkey = sodium_crypto_sign_secretkey($cesar_sign_kp);
$ cesar_sign_publickey = sodium_crypto_sign_publickey($cesar_sign_kp);
```

Una vez creada las llaves y compartida la llave pública, Cesar crea el mensaje y se incluye la firma.

```
$message = 'Hola. Este mensaje fue creado por Cesar';
$signed_msg = sodium_crypto_sign(
    $message,
    $cesar_sign_secretkey
);
```

Para que cada receptor pueda validar el mensaje y remover la firma:

```
$original_msg = sodium_crypto_sign_open(
    $signed_msg,
    $cesar_sign_publickey
);
```

[10] https://en.wikipedia.org/wiki/End-to-end_encryption

```
if ($original_msg === false) {
    // Error de firma
} else {
    echo $original_msg;
}
```

6.3.4. Validación de datos de entrada en la aplicación.

Una de las vulnerabilidades más comunes de las aplicaciones web, son las inyecciones SQL y se encuentra en el TOP 10 de Riesgos en Seguridad de Aplicaciones en la OWASP[11]. Una vulnerabilidad de Inyección SQL ocurre cuando se envían datos serializados como parte de una consulta, datos que el atacante puede utilizar para engañar al intérprete de tal manera que ejecute un comando involuntario y así pueda acceder a datos que no tiene autorización, o saltar controles de accesos de la aplicación.

Este tipo de vulnerabilidades se presenta cuando los datos enviados por el usuario no son filtrados o validados por la aplicación, o cuando se invocan consultas no parametrizadas, y sin codificar los parámetros de entrada conforme al contexto a las políticas que se implementen en el proceso de análisis y diseño.

Como políticas de seguridad en las empresas al momento de desarrollar software, se debe tener en cuenta las siguientes:

- Implementar comprobaciones de integridad para cualquier objeto serializado enviado por el usuario, por ejemplo mediante el envío de token de usuario, y validando en el lado del servidor, que este token efectivamente sea del usuario que está activo en la aplicación y tiene permisos para ejecutar la acción.
- Aplicar técnicas de validación de entrada de datos al servidor, lo cual se puede hacer con los diferentes lenguajes en el que se esté desarrollando, como técnicas de saneamiento y filtros, garantizando que el dato ingresado corresponda al tipo de dato definido por el sistema. También es importante tener la política de escapar caracteres especiales.
- Creación de logs de monitoreo, al momento de que el servidor de aplicaciones, arroje algún error por sintaxis o alguna excepción, cuando se ejecute la consulta SQL, de esta manera podemos solucionar de manera inmediata el bug de programación.

[11] https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf

6.4. POLÍTICAS DE SEGURIDAD PARA LA GESTIÓN DE LOGS.

En todos los sistemas de información, aplicativos, bases de datos, servidores y Software en general, debe existir una política de seguridad para la creación de Logs o rastros de auditoría que registren todas las actividades de los usuarios, las excepciones, fallas y eventos de seguridad del sistema.

Es responsabilidad de la Oficina de TI, monitorear la actividad de cada uno de los Logs que se implementen como política de seguridad en la compañía.

Para realizar un correcto uso de estos Logs, se deben establecer roles y responsabilidades para la gestión de estos, además de contar con entrenamiento de las personas que se involucrarán.

6.4.1. Captura y generación de Logs.

Para cumplir con el objetivo general de la generación de Logs, los registros que se recopilen en estos archivos, deben ser datos que sirvan de interés para realizar análisis, revisión, examen y reconstrucción de los datos y eventos mediante el historial de estos registros capturados, además de servir como soporte para investigaciones de auditorías externas.

En este proyecto vamos a ampliar la gestión de Logs para el caso de desarrollo de Software, y dejar a un lado los dos del sistema, y servidores. Para ello nos vamos a concentrar en Logs de accesos de usuarios al sistema, acceso privilegiado o administrativo, uso de mecanismos de autenticación e identificación y accesos remotos.

6.4.1.1. Logs de accesos de usuarios al sistema.

Uno de los puntos claves al momento de implementar una política de seguridad para la gestión de Logs, es el primero evento que realiza un usuario al ingresar al sistema. Si bien es cierto que el usuario solo ingresa su usuario y contraseña, por debajo debemos capturar la mayor información posible para garantizar la seguridad del mismo y de la compañía.

Por lo general este tipo de Logs se pueden almacenar en una base de datos, donde se recomienda guardar fecha y hora del inicio de la sesión, dirección MAC, dirección IP desde donde se inició sesión, identificación del equipo donde se inició sesión (en el caso de dispositivos móviles, podemos capturar marca del equipo, identificador

único, de tal manera que podamos gestionar equipos de confianza para futuros inicios de sesión).

6.4.1.2. Logs de acceso privilegiado o administrativo.

Como política de seguridad en la gestión de Logs, es muy importante contar con un Logs que guarde un token al momento de iniciar sesión, para que cualquier actividad que se quiera realizar se haga enviando este token al front-end, para evitar que otro usuario lo suplante. Estos token deben ser únicos en el sistema y deben enviarse cada vez que se quiera hacer una solicitud que involucre privilegios o cambios en los datos.

Como medida adicional, cualquier intento de ejecutar acciones al sistema, donde no se cuente con estos privilegios, debe ser notificada al administrador del sistema, para poder identificar como está tendiendo acceso a la ruta donde se desea ejecutar la acción ilícita, para así poder tomar cartas en el asunto.

6.4.1.3. Logs de accesos remotos.

Tener la seguridad de que los que estamos iniciando sesión realmente somos nosotros, nos lleva a tener una tranquilidad sobre cualquier sistema de información, por eso se debe tener una política de seguridad en la gestión de Logs cuando realizamos accesos remotos.

Cuando hablamos de accesos remotos, es cada vez que se inicia desde una dirección IP que no es de nuestro uso cotidiano, en este Log, debe contener información de fecha y hora exacta del inicio de sesión, país y ciudad desde donde se inició sesión, información del navegador y dispositivo que inició, ya que con esta información podemos llevar un control de si fue o no suplantado un usuario.

6.4.2. Respaldo, retención y restauración de Logs.

Una de las políticas de seguridad para la gestión de Logs es la de establecer un plan de respaldo de logs de auditoría mediante la implementación en el código fuente de la aplicación con la que se cuente, estableciendo directrices de detección, respaldo, y recuperación de los mismos. Esto es muy importante ya que es estos logs constituyen una evidencia para la identificación de incidentes de seguridad de cualquier naturaleza.

Los logs de auditoría deben contar con una política de guardado, ya que debe contener fecha, hora y actividad exacta que esté cumpliendo, para que podamos garantizar que no se vayan a sobrescribir con los demás Logs, y sean fácilmente encontrados.

Se debe tener una política clara de cuánto debe durar cada archivo de Log en nuestro sistema, ya que muy posiblemente debamos realizar un algún momento auditorías internas al Software, para encontrar irregularidades en el funcionamiento.

Para el caso de la retención y almacenamiento, en una política de seguridad de gestión de Logs se debe tener en cuenta las siguientes etapas:

- Datos de producción: son los datos que se les debe hacer un análisis y monitoreo en tiempo real, evaluación o auditorías periódicas.
- Datos de respaldo: Son los datos que son una copia idéntica de los datos que se encuentran en producción, pero que fueron modificados en algún momento por el usuario, como sus datos personales, preferencias, etc.
- Datos de archivo activo: Son datos de producción que deben ser almacenados por un mayor tiempo por cuestiones regulatorias y/o requerimientos legales, como son los Logs de contabilidad, controles de acceso, entre otros.

Después de tener políticas sobre qué información se va a guardar y qué no, se deben implementar las políticas de seguridad en el almacenamiento de estos Logs. Para ello se debe tener en cuenta las siguientes características:

- Almacenamiento Online: Los datos son almacenados en servidores de alto rendimiento en la nube, donde se garantice el ágil acceso y ofrezca alta disponibilidad.
- Almacenamiento Near-Line: Los datos se almacenan en sistema removibles y que estén disponibles para usuarios limitados por largos periodos de tiempo.
- Almacenamiento Offline: Los datos son alojados en discos duros o cintas magnéticas en algún sitio donde se garantice el acceso después de ser montados.

6.4.3. Seguridad y resguardo de los LOGS.

Otra política de seguridad que se debe tener en cuenta es asegurar la disponibilidad, confidencialidad e integridad de los logs a través de todo su ciclo de vida, y esto va desde el periodo de generación, tratamiento, almacenamiento y eliminación. Esto

porque los Logs son susceptibles a ser alterados y eliminados, si no se tienen un control estricto de seguridad para su almacenamiento y transmisión.

La manipulación de estos datos dentro de los logs, tiene un impacto negativo y crítico para la organización, como por ejemplo ocultar algún incidente, robo de información, suplantación, elevación de privilegios sin permisos.

Para evitar este tipo de inconvenientes, es necesario que dentro de las políticas de seguridad que se tengan, se contemple la protección de estos archivos y el lugar donde se encuentren almacenados, con controles físicos y lógicos de acceso, definir una capacidad adecuada de recursos para el almacenamiento y procesamiento de los logs, y tener los permisos necesarios para que estos archivos no puedan ser modificados y se pueda garantizar la integridad de los mismos.

6.5. POLÍTICAS DE SEGURIDAD PARA EL CONTROL DEL ACCESO EN LOS SISTEMAS DE INFORMACIÓN.

El control de acceso en los sistemas de información es uno de los procesos vitales dentro de la arquitectura del Software, donde dada una petición de recursos, se permite o se niega el acceso al sistema en base a la aplicación de las políticas de acceso.

El control de acceso comprende mecanismos de autenticación, autorización y auditoría. Sus principales son proteger los datos y recursos frente al acceso no autorizado, o frente a una modificación no autorizada de los mismos, y su vez garantizar el acceso a usuarios legítimos con sus respectivos roles y permisos.

La primera fase del sistema de control de acceso es la definición de políticas de seguridad de acceso, donde se establecen el conjunto de reglas que regulan el acceso a los recursos del sistema de manera abstracta.

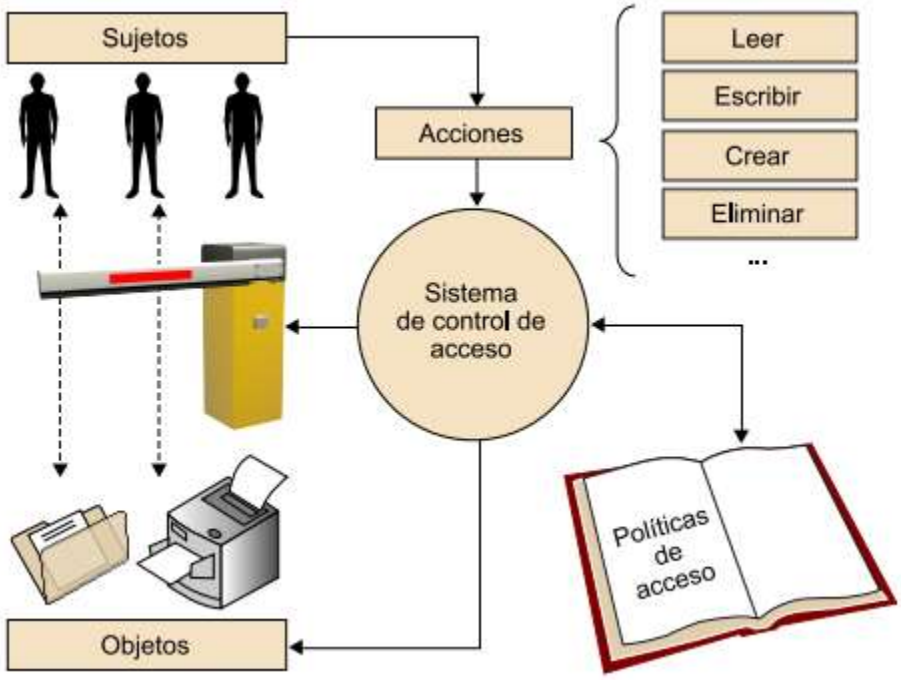

Figura 5 Elementos básicos de un sistema de control de acceso

6.5.1. Conceptos básicos de un sistema de control de acceso.

Todo sistema de control de acceso tiene algunos elementos básicos:

- Objetos: Son todas las entidades de un sistema, susceptibles a ser protegidas. En el caso específicos de un Software, tenemos los directorios, archivos, y módulos dentro del aplicativo.

- Acciones: Es todo lo que se puede realizar sobre un objeto. Son aquellas acciones que se puede realizar sobre algún módulo o archivo como lectura, escritura, creación y/o eliminación.

- Sujetos: Se refiere a la entidad dentro de un sistema de información con capacidad para requerir el acceso a objetos del sistema. En un Software típico estos sujetos son los usuarios del sistema.

Al momento de elaborar un Software, estos conceptos se deben tener muy claros, y antes de crear las políticas de seguridad de control de acceso, sabemos que todo sujeto puede realizar acciones sobre los objetos del sistema. El sistema de control de acceso regula si determinado sujeto tiene permisos para ejecutar determinada acción sobre un determinado objeto.

6.5.2. Control de acceso basado en roles.

En función de cómo se aplican y gestionan las políticas de acceso de distingue un tipo de control que actualmente se implementa en la mayoría de los Software, este se denomina Control de Acceso basado en Roles. En este tipo, las políticas son definidas el sistema mediante permisos de clases o roles, donde cada rol tiene ciertos privilegios y cada sujeto tiene asignado un rol, de tal manera que cada sujeto del sistema, tiene privilegios del rol o los roles que adquiera.

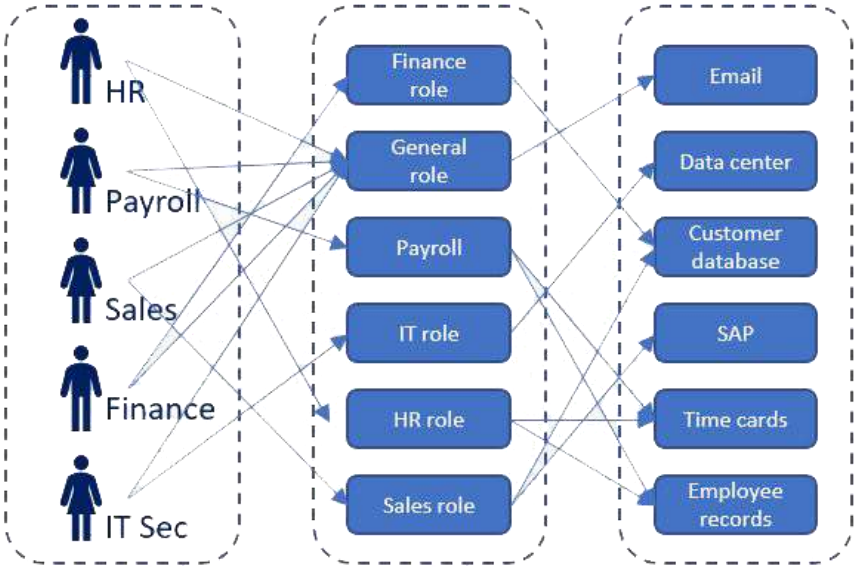

Figura 6 Ejemplo de Control de Acceso basado en Roles

Cuando se trabaja en una empresa grande, y existen bastantes usuarios que harán parte del sistema de información, resulta poco práctico definir privilegios de acceso para cada usuario de forma individual.

El control de acceso basado en roles (RBAC [4] por sus siglas en inglés role-bassed access control), se fundamenta en la idea de asignar privilegios para realizar acciones a roles en vez de a sujetos del sistema. De tal manera que, si al sujeto se le asignan diferentes roles, este hereda todos los privilegios del rol (como se evidencia en la figura 6).

El rol se basa en las actividades o funciones que cumple alguien dentro de la organización.

6.5.3. Control de acceso con Middleware[12]

En la mayoría de sistemas de información, existen módulo que son visibles por cualquier sujeto, sin importar el rol que tenga dentro de la plataforma, como es la vista de registro, o de inicio de sesión, o la parte del front-end donde ven los servicios que ofrece una organización.

El uso de Middleware en el desarrollo de Software proporciona un mecanismo para filtrar solicitudes que ingresan a la aplicación de la organización.

[12] https://www.redhat.com/es/topics/middleware/what-is-middleware

Para profundizar en el tema, una de las alternativas que tienen los arquitectos de Software, es implementar por medio de un Framework, sus aplicaciones o nuevos desarrollos dentro de una organización, en PHP, existe uno que ha sido catalogado como uno de los Frameworks creados para ayudar a los desarrolladores en crear aplicaciones web simples y flexibles, llamado Laravel [5].

Laravel dentro de sus componentes incluye un Middleware que verifica que el usuario de su aplicación este autenticado para poder ejecutar cualquier petición HTTP. Si el usuario no está autenticado en el sistema, el Middleware redirigirá al usuario a la pantalla de inicio de sesión. Este es un ejemplo de Middleware que se pueden implementar en las organizaciones.

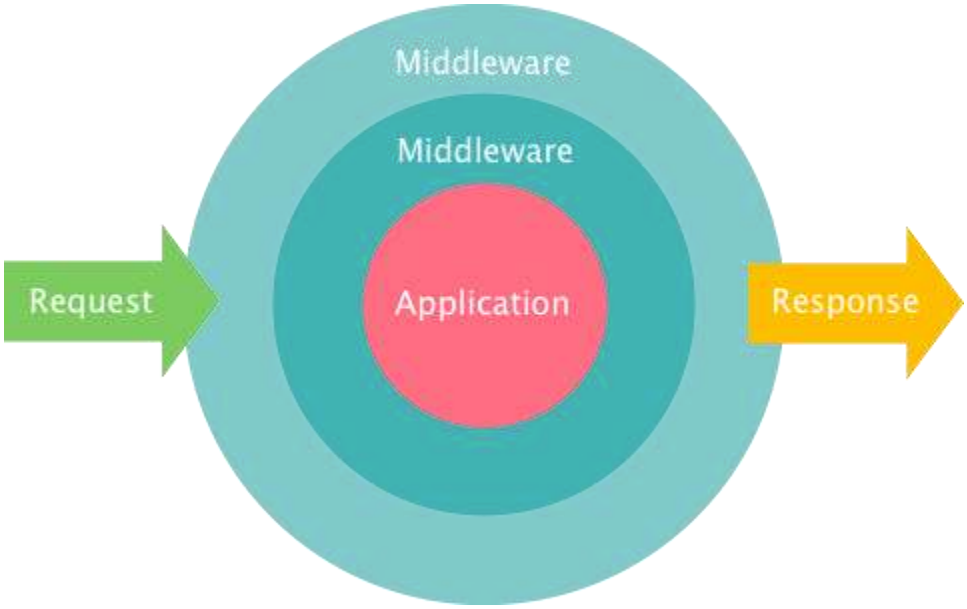

Figura 7 Ejemplo de Middleware en Peticiones al Software

Para el caso del control de acceso de las aplicaciones, con Laravel se puede implementar un Middleware, que cada vez que el usuario ejecute una petición HTTP, este primero pase por unas reglas, que se implementan dentro del Middleware, donde verifica si el usuario que está intentando ejecutar la acción, tiene los privilegios correspondientes. Esta consulta la puede realizar a la base de datos donde se relacione el modelo RBAC, y si el Middleware detecta que no tiene el rol para ejecutar la acción en el objeto, este lo redirecciona y lo saca de la sesión actual.

Tal como se evidencia en la figura 7, una aplicación puede contar con “n” cantidad de Middleware, los cuales pueden manejar diferentes tipos de controles de seguridad para el acceso o peticiones a la aplicación. Otro ejemplo de Middleware, y que actualmente se utiliza en el desarrollo de Software, es la protección frente a ataques CSRF[13] (Falsificación de petición en sitios cruzados, o por sus siglas en inglés cross-site request forgery).

Laravel para prevenir este tipo de ataques, genera automáticamente un “token” de CSRF para cada sesión activa administrada por la aplicación. Este token se utiliza para verificar que el usuario autenticado es el que realmente está haciendo las solicitudes a la aplicación.

[13] https://es.wikipedia.org/wiki/Cross-site_request_forgery

7. CONCLUSIONES

Durante el desarrollo del presente proyecto se incursionó en el estudio de las políticas de seguridad que se deben tener en cuenta al momento de desarrollar un Software partiendo de la base de toda organización que son sus funcionarios, estudiando técnicas que eviten la propagación de virus y troyanos que puedan afectar directa e indirectamente a la red donde se encuentra alojado los servidores que contienen el Software.

También se estudiaron las políticas de seguridad que se deben tener en cuenta al momento de tener alternativas de guardado de la información, en este caso el almacenamiento en la nube, la cual es una tecnología que se puede adaptar en las organizaciones, siempre y cuando te tengan las suficientes políticas de seguridad para evitar que te pierda la información.

Todo lo anterior no cumpliría con un sistema seguro, si no cuenta con el uso de técnicas de criptografía. Según lo estudiado, estas políticas se seguridad garantizan la autenticidad, confidencialidad, integridad, asegurar la identidad y el no repudio de los mensajes y sus emisores. Para entender un poco de estas propiedades se profundizo en el uso de contraseñas encriptadas para la autenticación, certificados electrónicos, firmas electrónicas y el concepto de llave pública.

Los LOGS de todo el sistema, para procesos de monitoreo y auditoría, no dejan de ser un tema ajeno a la seguridad de la información, ya que dependemos de estos para poder identificar eventos dentro del sistemas que afecten los datos de alguna manera negativa, o poder evidenciar el robo de información, o sencillamente para poder tener un historial de todas las actividades de los usuarios.

Y por último se estudiaron las políticas de seguridad para el control de acceso al sistema, lo cual es el talón de Aquiles que permite en muchos de los sistemas de información, saltarse todos las anteriores políticas, implementadas para tener un sistema seguro, logrando suplantar a un usuario si no se tienen las medidas necesarias para evitar este tipo de intrusiones.

8. BIBLIOGRAFÍA

[1] Turaev, H, **"**Prevention of Ransomware Execution in Enterprise Environment on Windows OS: Assessment of Application Whitelisting Solutions" 2018.

[2] A. Biryukov, Argon2: the memory-hard function for password hashing and other applications, https://password-hashing.net/argon2-specs.pdf

[3] Password Hashing Competition (PHC) - https://password-hashing.net/

[4] Ferraiolo, D.F. & Kuhn, D.R. (October 1992). "Role-Based Access Control"

[5] Muhammad Anif "Designing Internship Monitoring System Web Based With Laravel Framework", 2017.