ANÁLISIS TÉCNICO DE LA POLÍTICA HTTP STRICT TRANSPORT SECURITY (HSTS) COMO ESTÁNDAR DE SEGURIDAD A NIVEL DE SERVICIOS EN CLOUD COMPUTING.

SERGIO ALBERTO GARCIA ALVAREZ

LEYBERTH JOSE RUMBO LUQUEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
ESPECIALIZACIÓN EN TELECOMUNICACIONES
BUCARAMANGA
2016

ANÁLISIS TÉCNICO DE LA POLÍTICA HTTP STRICT TRANSPORT SECURITY (HSTS) COMO ESTÁNDAR DE SEGURIDAD A NIVEL DE SERVICIOS EN CLOUD COMPUTING.

Trabajo de grado presentado como requisito parcial para obtener el título de:
ESPECIALISTA EN TELECOMUNICACIONES

SERGIO ALBERTO GARCIA ALVAREZ

LEYBERTH JOSE RUMBO LUQUEZ

DIRECTOR

RAUL BAREÑO GUTIERREZ
ING. MG. EN TELECOMUNICACIONES

UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
ESPECIALIZACIÓN EN TELECOMUNICACIONES
BUCARAMANGA
2016

DEDICATORIA

A Dios, a mis Padres y todos mis seres queridos por su apoyo.

Sergio Alberto Garcia Alvarez

A Dios por darme la sabiduría de culminar exitosamente este recorrido trazado, a mis padres por su apoyo incondicional, a mis hermanos y sobrino por su ánimo pronto y así hoy poder alcanzar este nuevo logro en mi vida

Leyberth Jose Rumbo Luquez

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

RESUMEN GENERAL DE LA MONOGRAFÍA

TITULO: **ANÁLISIS TÉCNICO DE LA POLÍTICA HTTP STRICT TRANSPORT SECURITY (HSTS) COMO ESTÁNDAR DE SEGURIDAD A NIVEL DE SERVICIOS EN CLOUD COMPUTING.**

AUTORES: **SERGIO ALBERTO GARCIA ALVAREZ**
LEYBERTH JOSE RUMBO LUQUEZ

PALABRAS CLAVE: Computación en la Nube, Seguridad en la Nube, Criptografía, Confidencialidad, SSL/TLS, HSTS, HTTPS

RESUMEN:

La necesidad de seguridad en la nube ha llevado a la búsqueda de diversas estrategias para garantizar la fiabilidad de los datos, la autenticación de los servidores a los que se pretende ingresar es una de estas, debido a que, generalmente, los datos son transportados mediante el protocolo *Transmission Control Protocol* (TCP) y este no garantiza la seguridad del canal o una identificación de equipo seguro. En la actualidad se han desarrollado protocolos de transporte seguro como lo son *Secure Socket Layer* (SSL) y *Transport Layer Security* (TLS) que permiten establecer sus conexiones basados en la generación de certificados que autentican el sitio al que se está accediendo. Sin embargo, no todos los sitios que cuentan con certificados son reales o tienen errores que comprometen la seguridad de los usuarios, si bien los navegadores lo advierten, en algunos casos se ignoran estas advertencias lo que deriva en el ingreso a un sitio potencialmente inseguro aumentando el riesgo de ser víctimas de un ataque informático. Es allí donde HSTS entra a desempeñarse como un complemento de seguridad que permite exclusivamente conexiones seguras entre la fuente y el destino, asegurando al usuario que sus datos, en cuanto a conexión respecta, se encuentran seguros. En esta monografía se presenta una revisión de la computación en la nube al igual que de la política HSTS y la forma en que su implementación permite mejorar la seguridad de los sitios donde se implemente.

---

* Monografía

** Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Especialización en Telecomunicaciones. Director: Raul Bareño Gutierrez.

GENERAL SUMMARY

TITLE **TECHNICAL ANALYSIS OF HTTP STRICT TRANSPORT SECURITY (HSTS) POLICY AS A SECURITY STANDARD IN CLOUD COMPUTING.**

AUTHOR(S): **SERGIO ALBERTO GARCIA ALVAREZ**
LEYBERTH JOSE RUMBO LUQUEZ

KEY WORDS: Cloud Computing, Security in Cloud Computing, Cryptography, Confidentiality, SSL/TLS, HSTS, HTTPS

ABSTRACT:

The continuous effort to improve the security environment in cloud computing has guided I.T designers and developers into different strategies, trying to guarantee data reliability. Server authentication is one of those ways, even more when data is transmitted using the Transmission Control Protocol (TCP) and this one does not guarantee the security into the channel nor a reliable server. Therefore, nowadays protocols as Secure Socket Layer (SSL) and more recently, Transport Layer Security (TLS) have been extremely relevant because they establish connections through authentication certificates, which assure that the server is whose it say it is. However, not all sites that have a certificate are real or most of them have weaknesses that compromise user's security. Despite browsers warn users about not trusted certificates or configuration mistakes, some of them just ignore it increasing the risk to suffer a web attack. Due to HSTS just allow secure connections among source and destination, users can be sure that their data, at least during the connection process, are safe. This document contains a review about cloud computing and HSTS policy and how its implementation allow to improve security on the sites where it is deployed.

---

* Monograph

** School of Electrical Engineering, Electronic and Telecommunication. Specialization in Telecommunications. Director: Raul Bareño Gutierrez

INTRODUCCIÓN

En la actualidad, las empresas y el usuario común han visto en la nube la opción de centralizar sus datos ya que esta, como lo define la NIST SP 800-145, es un modelo que permite compartir un pool de recursos computacionales (redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente configurados y liberados para uso de manera sencilla.

En [1] se expone la formar de como la demanda del servicio en la nube ha ido en aumento y, según sus estimaciones, se espera que para el 2018 el 76% del tráfico global de servicios o aplicaciones se encuentren alojadas en ella. Lo dicho anteriormente se puede ver reflejado con mayor claridad si se tiene en cuenta que en el 2013 el total de tráfico suministrado por los centros de datos fue de 3.1 ZB de este valor un poco más de la mitad (1.6 ZB) corresponden a tráfico proveniente desde los servicios de nube mientras que la proyección a 2018 muestra que de los 8.6 ZB totales, 6.5 ZB serán suministrados por esta.

La expansión de estos servicios tiene asociados unos riesgos, entre los cuales sobresale especialmente todo lo referente a la seguridad de los datos almacenados y las cuentas de los usuarios; de aquí se desprenden una serie de estrategias para mitigar los riesgos presentes entre los cuales se pueden encontrar, fortalecer las políticas de seguridad, manejo de credenciales y permisos, evaluación de controles físicos y el asegurar una conexión segura.

Si bien todas se complementan entre sí, se hará énfasis especialmente en el aseguramiento de la conexión mediante el modelo de conexión *HTTP Strict Transport Security* (HSTS) el cual complementa los protocolos de transporte seguro ya que fuerza a una comunicación constante bajo TLS o SSL para prevenir posibles ataques.

1. OBJETIVOS

1.1. OBJETIVO GENERAL

Analizar las características de seguridad del modelo de conexión segura *HTTP Strict Transport Security* (HSTS) con el fin de determinar sus principales fortalezas y debilidades al ser usado en servicios de computación en la nube

1.2. OBJETIVOS ESPECÍFICOS

- Realizar un análisis de la penetración, crecimiento y utilización de los servicios en la nube en el entorno corporativo actual.
- Desarrollar un estudio de vulnerabilidades que permita identificar posibles debilidades presentes en la prestación de servicios en nube que faciliten posibles ataques.
- Analizar los protocolos SSL y TLS destacando sus características y analizar las vulnerabilidades detectadas en ellos.
- Analizar la política de seguridad HSTS como un mecanismo que permite mejorar autenticación de servicios alojados en la nube.

2. COMPUTACIÓN EN LA NUBE

La computación en la nube hace referencia a las aplicaciones y servicios que se ejecutan en una red distribuida utilizando recursos virtuales, teniendo acceso a estos mediante protocolos de internet comunes y diferentes estándares de red [2]. En un esfuerzo por describir mejor el modelo de nube, diferentes instituciones han definido a qué se hace referencia cuando se habla de computación en la nube.

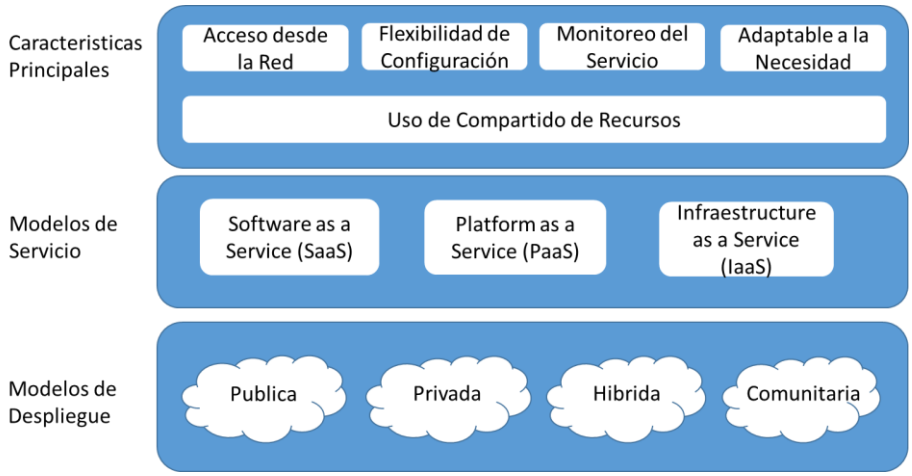
2.1. MODELO NIST

La definición más común es la entregada por científicos del U.S *National Institute of Standards and Technogy* (NIST) la cual define los servicios de nube cómo:

“Un modelo que permite, acceso a un conjunto de recursos informáticos configurables (como lo pueden ser redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente configurados y liberados con un esfuerzo mínimo de administración” [3]

De igual forma, esta definición se encarga de describir 5 características esenciales, 4 modelos de despliegue y 3 modelos de servicio, lo anterior se resume en la figura 1:

FIGURA 1. Modelo NIST para computación en la nube



FUENTE: Diseño basado en [3]

2.2. CARACTERÍSTICAS DE LA COMPUTACIÓN EN LA NUBE

Al revisar detenidamente las características esenciales descritas por el modelo NIST se puede definir cada una como:

- Uso Compartido de los recursos: El *Cloud Service Provider* (CSP) realiza una asignación o redistribución de sus servicios físicos o virtuales a sus múltiples usuarios dependiendo de lo que estos demanden.
- Flexibilidad de Configuración: Las características de configuración pueden ser modificadas con facilidad y rapidez permitiendo al usuario optimizar su tiempo e incluso comprar nuevas características.
- Monitoreo del Servicio: Los servicios de nube tienen la capacidad de controlar y optimizar los recursos disponibles de acuerdo a su uso. De igual forma quien contrata el servicio también puede monitorear, controlar o reportar lo que genera una transparencia en la prestación del servicio.

- Autoadaptable a la necesidad: El usuario puede generar características como una red de almacenamiento de manera autónoma sin requerir de interacción con el CSP.

- Acceso a la Red: Las características se encuentran disponibles en la red y se puede ingresar a estas de manera estándar en las plataformas clientes.

Sumándose a las 5 características, en [4] se reseñan:

- Bajos Costos: Debido a la forma en que es concebido el modelo de operación y a su alto uso, la reducción en los costos siempre es una variable a favor.

- Fácil Utilización: Dependiendo del tipo de servicio ofrecido, en algunos casos el cliente se encontrará con que no existe requerimiento de hardware o licencias de software para poder implementar el servicio.

- Calidad del Servicio: *Quality of Service* (QoS) es una característica que puede ser obtenida durante la negociación con el CSP.

- Confiabilidad: El balanceo de carga y la respuesta que se tiene hacia los fallos generan una gran confiabilidad, difícil de alcanzar por las limitaciones que puede llegar a tener una organización.

2.3. MODELOS DE SERVICIO

Con el pasar del tiempo, se han desarrollado diferentes ofertas de venta de servicios asociados con la computación en la nube. El portafolio de los servicios ofrecidos trae consigo los modelos de servicio. En base a [3] existen 3 modelos que han sido aceptados universalmente:

Infraestructure as a Service (IaaS):este modelo se caracteriza debido a que el CSP entrega la estructura de red como servicio, espacio de almacenamiento y funciones de red, evitando de este modo la compra de servidores, equipos de red y demás equipos que puedan llegar a ser requeridos en determinado momento para el despliegue de una plataforma.

Platform as a Service (PaaS): Se define como la entrega de una plataforma computacional al igual que diferentes soluciones corporativas como un servicio donde se ahorra gran parte del costo asociado al desarrollo y el manejo de estos. Generando un soporte integro durante el ciclo de vida de los servicios.

Software as a Service (SaaS): Comúnmente utilizada por las empresas y los particulares, este servicio es donde se tiene un software que almacena los datos de manera remota y al cual se tiene acceso mediante un usuario cliente proporcionado por la empresa desde un explorador web.

Las diferencias marcadas entre los diferentes modelos de servicio se reflejan de igual forma al momento de hablar de seguridad ya que en un sistema IaaS el manejo y aseguramiento total de las aplicaciones, sistemas operativos y contenidos recaen sobre quien maneja el servicio de nube. Explicándolo de otra forma, entre menos influya el proveedor mayor responsabilidad debe adquirir quien maneja el servicio.

En la figura 2 se aprecian algunos ejemplos de cómo los diferentes servicios son usados por las empresas.

FIGURA 2. Modelos de servicio

FUENTE: Tomado de [11]

2.4. MODELOS DE DESPLIEGUE.

Los modelos de despliegue definen el propósito de la nube y la manera en que esta se localiza. Las 3 clases de modelos de despliegue que hacen parte de la de la definición de la NIST se muestran en la tabla 1:

TABLA 1. MODELOS DE DESPLIEGUE

MODELO	*MANEJADA POR*	*DUEÑO DE LA INFRAESTRUCTURA*	*LOCALIZACIÓN DE LA INFRAESTRUCTURA*	*ACCESIBLE PARA*
Público	Proveedor externo	Proveedor externo	Fuera de la Organización	No Confiables

MODELO	*MANEJADA POR*	*DUEÑO DE LA INFRAESTRUCTURA*	*LOCALIZACIÓN DE LA INFRAESTRUCTURA*	*ACCESIBLE PARA*
Privado/Comunitario	Organización/ Proveedor Externo	Organización/ Proveedor Externo	Dentro o Fuera de la Organización	Confiables
Híbrido	Organización y Proveedor Externo	Organización y Proveedor Externo	Dentro o Fuera de la Organización	Confiables y No Confiables

FUENTE: Tomado de [3].

Se debe aclarar que cuando se hace referencia a:

- Manejada por: se incluye la operación, la seguridad, el mantenimiento, etc.
- La Infraestructura implica: Equipos, Redes, Centros de Almacenamiento, etc.
- La Localización: Hace referencia al espacio físico donde se encuentran los equipos
- Accesible: Hay dos clases de consumidores, los confiables y los no confiables, los primeros hacen referencia a quienes sean parte de la organización y los no confiables, quienes pueden acceder a los recursos, autorizados o no, y que no tienen ningún vínculo con la organización

Los modelos de despliegue deben ser entendidos en un contexto más amplio que el servicio interno contra el externo en lo concerniente a ubicación, recursos e información. También se debe tener en cuenta quién va a hacer uso de estos recursos y quien es el responsable por la administración, seguridad y todo lo concerniente a políticas y estándares.

Por lo anterior, a la hora de realizar un proyecto de migración a la nube, se debe tener en cuenta:

- El tipo de recursos, tareas e información que serán manejadas.

- El responsable de su manejo y el cómo se realiza.

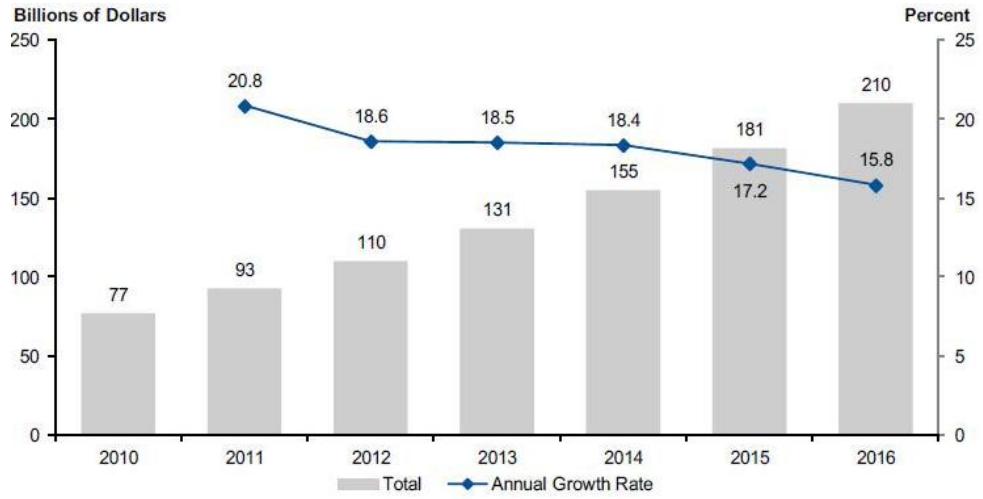
- Qué controles se van a seleccionar y cómo se van a integrar

3. PENETRACIÓN DE LA COMPUTACIÓN EN LA NUBE

Las perspectivas de la computación en la nube en general son muy buenas. Las pequeñas y medianas empresas encuentran en la nube la solución de confiabilidad y ahorro que en muchas ocasiones una infraestructura propia no puede brindar, principalmente por limitaciones técnicas.

Según un estudio realizado por Gartner [5], en figura 3 se muestra el comportamiento del crecimiento esperado para el año 2016 donde las ganancias de los servicios en la nube pública serían mayores a los 200 mil millones US. Lo que indica que en tan solo 6 años las ganancias estuvieron cerca de triplicase, si se tiene en cuenta que en 2010 estas eran cercanas a los 77 mil millones US.

FIGURA 3. Dinero generado por la computación en la nube



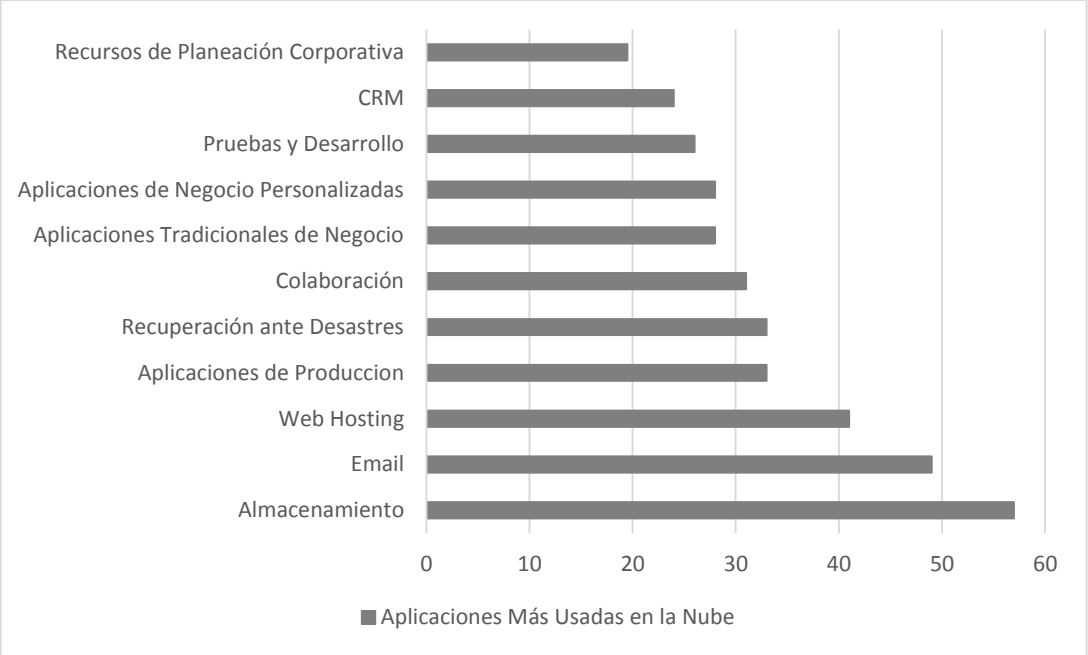
FUENTE: Tomado de [5]

Los servicios más demandados son el SaaS y la publicidad en la nube, esta última se define como un servicio que permite a las organizaciones integrar sus servicios a una experiencia digital personalizada. Estos dos servicios juntos suman cerca del 70% del total de los ingresos. En uno de los estudios adelantados por CISCO

[6], se espera que para 2019 SaaS mantenga su superioridad con cerca del 59% del total de demanda seguido por IaaS con 28% y PaaS con el 11%.

En su estudio Navigating Advanced Topics in Cloud Computing [7], CDW hace referencia a los servicios que han tenido una mayor transición a la nube, almacenamiento, email, y web hosting se destacan especialmente. En la figura 4 se observa el porcentaje de utilización de los diferentes servicios.

FIGURA 4. Aplicaciones más usadas en la nube



Fuente: Diseño basado en [7]

3.1. CASO LATINOAMERICA

La mayor parte de las ganancias generadas por los servicios en nube se encuentran centralizados en Europa y en América del norte. Sin embargo, un informe de la *Economic Commission for Latin America and the Caribean* (ECLAC) [8] estima que cerca del 5% provendrá de los servicios prestados en América Latina y que la zona cuenta con una proyección de crecimiento anual del 26%,

indicando que la adopción de la computación en la nube podría ser más rápida que en Europa o Asia. En la tabla 2 se puede observar cómo se comporta la distribución de ganancias por regiones.

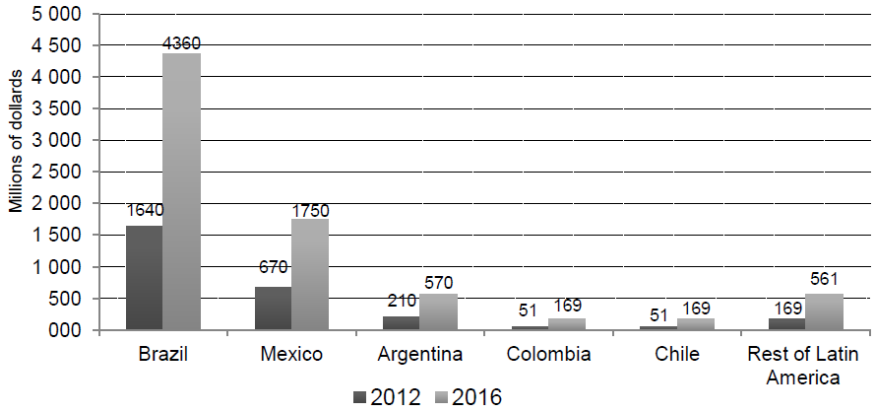
TABLA 2. BENEFICIOS ESTIMADOS EN NUBE PÚBLICA POR REGIONES
(Valores en miles de millones de dólares)

Región	*2011*	*2014*	*2016 (estimado)*
America del Norte	50.8	89.8	125.4
Europa (Oeste)	24.3	34.1	42.5
Asia Pacifico	9.3	14.9	19.0
China Continental	3.0	7.1	11.2
América Latina	2.4	4.7	7.6
Europa (Este)	0.4	0.7	1.1
Norte de Africa	0.3	0.6	0.9
Africa Subsahariana	0.2	0.3	0.5

FUENTE: Diseño adaptado [8]

Del estudio de la ECLAC se debe destacar que Brasil es el país de la zona con mayores entradas por los servicios en nube en el periodo 2012-2016. Sin embargo, los países más dinámicos podrían ser Colombia y Chile, ya que estos esperan triplicar su participación en el mismo periodo de tiempo. La distribución de los beneficios por los servicios prestados muestra que entre los 2 principales países de la región (Brasil y México) se concentra el 58% del dinero entrante por la computación en la nube; 24% corresponden a Argentina, Colombia, Chile y Perú; mientras que el 18% restante corresponde al resto de América Latina. En la figura 5 se observa el comportamiento de la zona frente a los servicios en nube.

FIGURA 5. Dinero generado por servicios en la nube en Latinoamérica



FUENTE: Tomado de [8]

Dentro de otro de los informes de la ECLAC, denominado Broadband in Latin America [9], se observa que la mayor dinámica de la computación en la nube presenta un impacto positivo en el número de empleos generados alrededor de esta actividad. Allí se toman como ejemplos a Brasil y a Argentina donde se reseñan la cantidad de empleos generados 5 y 10 años después de la adopción de la computación en la nube. Lo anterior se refleja en la tabla 3

TABLA 3. PROYECCIÓN DE EMPLEOS GENERADOS POR LA COMPUTACIÓN EN LA NUBE

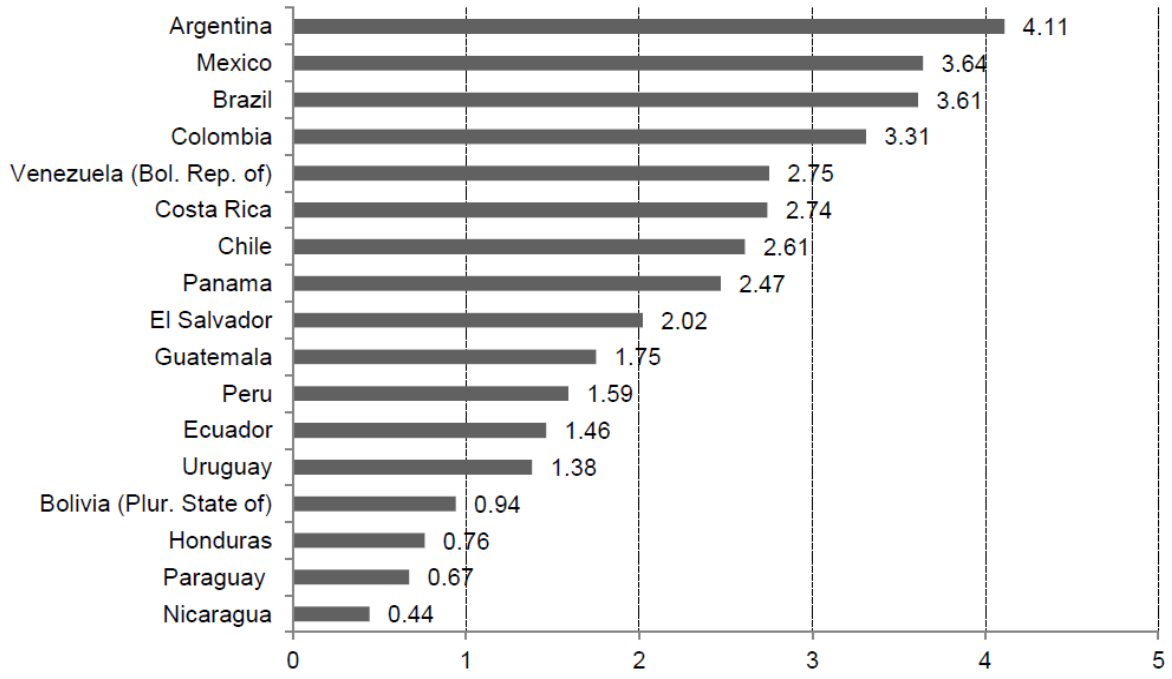
País	*5 años*	*10 años*
Argentina	117.300	128.900
Brasil	861.000	945.000

FUENTE: Tomado de [9]

Si bien es una proyección bastante grande, solo en Estados Unidos, se estima que en los mismos 10 años se generarán cerca de 3’179.000 empleos. Por su parte, la comisión Europea espera que para el 2020 se generen 2.5 millones nuevos puestos de trabajo y un aumento del PBI del 1% [10].

Pese a las ventajas ya discutidas de los servicios en nube, dentro de una investigación realizada por Pyramid Research en el 2013 enunciada en [10], muchas de las empresas en la región no se encuentran preparadas para que sus servicios operen desde la nube. Argentina que es el país mejor ubicado en este aspecto, apenas pasa la mitad de la puntuación máxima (4.1 de 7). En la figura 6 se observa el desempeño de los países Latinoamericanos en este aspecto.

FIGURA 6. Nivel de preparación de las empresas Latinoamericanas a la computación en la nube



FUENTE: Tomado de [10]

En el caso de México como se estima en [11], el gobierno Mexicano puede ahorrar cerca del 0.31% del PIB si todas las entidades estatales trasladan sus operaciones a la nube. Sin embargo los vacíos en la legislación y los desafíos en seguridad deben ser enfrentados de forma directa para que este ahorro se pueda ver materializado

Para el caso particular de Colombia, en [12] se estima que dentro de los siguientes 4 años se presente un fuerte crecimiento en todos los servicios de nube. Al igual que las tendencias mundiales, el servicio que representa mayor interés para la inversión es el SaaS con un 57%, seguido por el IaaS con un 41%, PaaS por su parte se encuentra rezagado ya que muchas compañías aún no comprenden como se puede explotar de forma idónea.

4. DESAFIOS EN LA COMPUTACIÓN EN LA NUBE.

Cada modelo de despliegue o de servicio lleva consigo diferentes riesgos asociados que deben ser clarificados con el CSP y de los cuales los administradores deben ser conscientes. De no tener claro los riesgos a los que se está expuesto se pueden generar decisiones equivocadas por parte de los administradores y por ende un posible deterioro en las funciones de las aplicaciones y/o infraestructura manejadas.

Los controles aplicados a la seguridad en la nube, en su mayoría, no son diferentes a los controles ejercidos en cualquier ambiente de red. Sin embargo, como fue mencionado anteriormente, los niveles de responsabilidad en la seguridad difieren de acuerdo a la clase de servicio contratado. De esta forma, por ejemplo, resulta crítico el manejo de las vulnerabilidades en los protocolos de internet ya que debido a la ubicuidad en el acceso puede exponer la conexión a un ataque *man in the middle* (MITM)

Para clarificar un poco lo anterior se tiene 2 ejemplos, por un lado el servicio AWS EC2 de amazon asegura únicamente la parte física, ambiental y de virtualización, dejando al usuario del servicio encargado de la seguridad de todo lo instalado. Por otro lado Salesforce.com una empresa que funciona bajo el modelo de servicio SaaS ofrece no solo seguridad en la parte física y ambiental, sino que también asegura la infraestructura, las aplicaciones y los datos, aliviando de esta forma mucha de la carga del usuario.

El desarrollo de modelos que permitan entender el papel de la seguridad en el marco de la computación en la nube ha sido plasmado en diversos artículos. En [13] se muestra uno de los escenarios más sencillos de entender se basa en la interacción de 3 participantes: usuarios, servicios y proveedores. Cada uno de estos juega un papel definido dentro del escenario e interactúa con los otros participantes (un usuario puede requerir un servicio o un servicio puede requerir el

uso de más recursos de máquina). De esta misma forma, cada intento de ataque puede ser definido dentro de alguna de las interacciones del modelo anteriormente descrito encontrando 6 diferentes superficies de ataque (usuario a servicio, servicio a usuario, usuario a proveedor, proveedor a usuario, servicio a proveedor y proveedor a servicio).

Otro de los casos estudiados [14], toma los modelos de servicio como base para el análisis de vulnerabilidades en la computación en la nube, allí se describen las diferentes amenazas presentes en cada modelo y la particularidad de cada uno de estos frente a los otros, dando especial énfasis al modelo SaaS ya que al enfocarlo desde la visión del usuario, este depende casi completamente de la seguridad ofrecida por quien oferta el software.

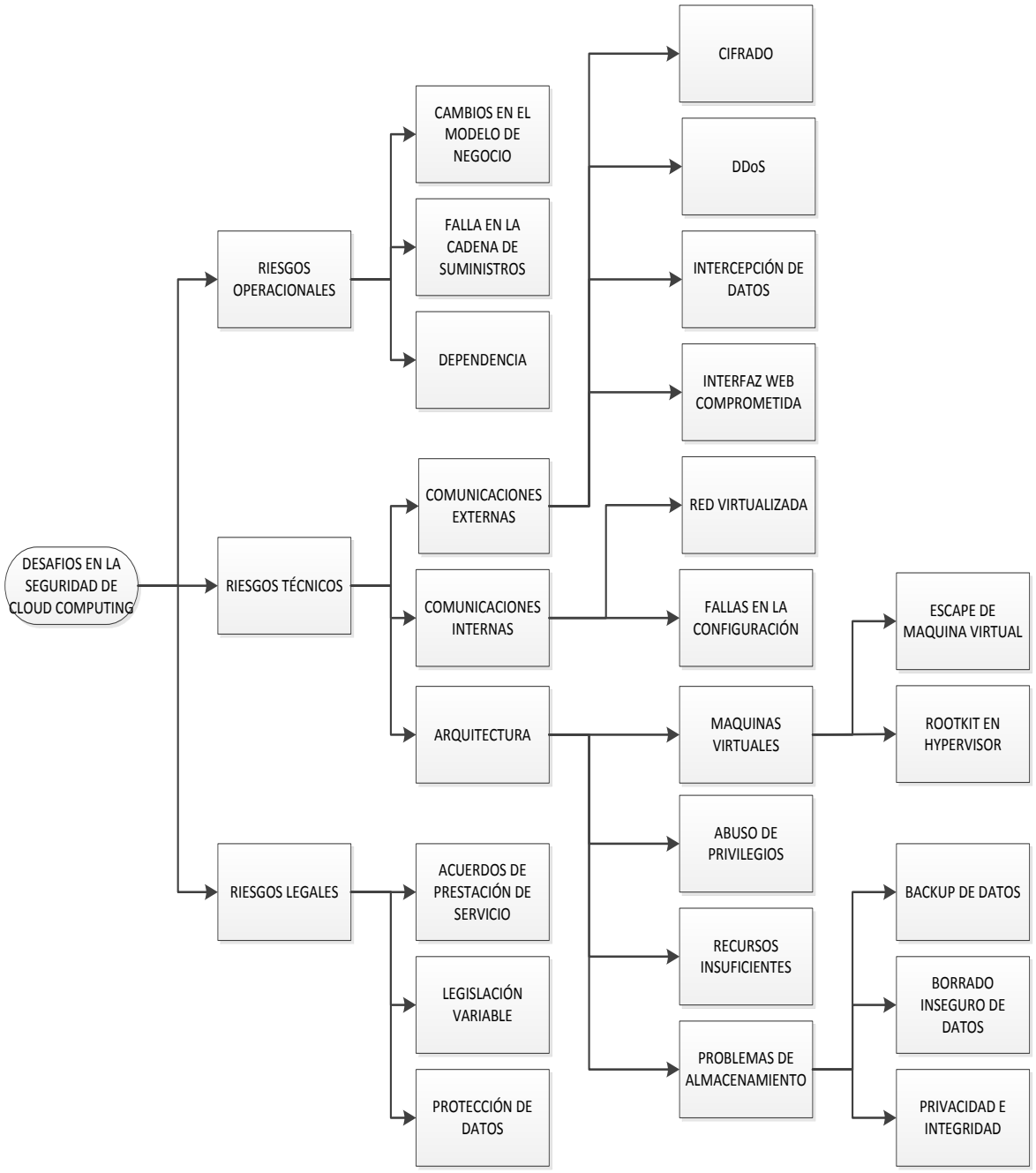
En virtud de lo anteriormente mencionado, se puede considerar que los ambientes de nube son una nueva plataforma computacional donde se puede aplicar la metodología de seguridad tradicional. En general, los sistemas de seguridad para servicios en nube buscan preservar 4 campos: confidencialidad, integridad, disponibilidad y privacidad. Una visión global del manejo de la seguridad en la nube se encuentra descrita en la figura 7:

FIGURA 7. Manejo de la seguridad en la computación en la nube

FUENTE: Diseño adaptado de [14]

Queriendo realizar un análisis de cómo se exponen estos 4 aspectos de forma continua en este tipo de servicios, en la figura 8 se realizó la distinción de 3 grandes grupos que asocian los riesgos, teniendo en cuenta la inseguridad en los protocolos, los problemas de perdida de datos y respaldo de información, el marco legal aplicable, entre otros. Los grupos referenciados son: riesgos operacionales, riesgos técnicos y riesgos legales.

FIGURA 8. Desafíos en la seguridad de la computación en la nube



4.1. RIESGOS OPERACIONALES

4.1.1. Infraestructura compartida: En una arquitectura multiusuario se comparten el conjunto de recursos computacionales y de almacenamiento al igual que los componentes de infraestructura. El compartir recursos puede disminuir la seguridad en estas plataformas ya que uno de los clientes alojados puede ser una máquina virtual maliciosa o encontrarse infectada por alguna clase de virus o malware y, al estar alojado dentro de la misma máquina, generar una ventana para lograr acceso a una maquina diferente debido a posibles fallas de aislamiento en el diseño del servicio contratado. Esto puede afectar toda la arquitectura de la nube en general, destacando que esta clase de ataques es posible especialmente en servicios IaaS. [15][16]

4.1.2. Cambio en el modelo de negocio: Como se ha expresado con anterioridad, la computación en la nube cambia la manera en que los servicios prestados por T.I. son brindados. El hecho que la información y los servicios se encuentren en un sitio externo a la organización, hace que esta necesariamente evalúe los riesgos asociados a la pérdida de control sobre la infraestructura. La organización también debe tener en cuenta las legislaciones aplicables en los países donde se aloje la información, esto será mostrado un poco más adelante, al igual que contar con un manejo apropiado en la administración de los diferentes servicios. [17]

4.1.3. Fallas en la cadena de suministros: El riesgo de no clarificar contractualmente cada uno de los aspectos relevantes de la prestación del servicio, puede llevar al desconocimiento del manejo por parte de terceros de tareas específicas subcontratadas por el proveedor. En estos casos los niveles de seguridad se encuentran compartidos por lo que si la falla se llegase a presentar en la entidad subcontratada, se podría exponer la máquina del usuario a diversas vulnerabilidades. [18]

4.1.4. Dependencia: Al utilizar un servicio en nube, el usuario necesariamente cede el control al proveedor en ciertos aspectos que pueden afectar en algún momento la seguridad. Esta pérdida de autonomía puede llevar a la incapacidad de incumplir metas a nivel organizacional y a una respuesta limitada por parte de la compañía en caso de detectar alguna desviación en la prestación del servicio (riesgos de seguridad, baja confidencialidad, mala calidad en el servicio). De la misma forma, si el proveedor presenta problemas económicos, la migración de los servicios a otra plataforma puede resultar costosa y poco práctica.De igual forma si se llega a detectar actividades maliciosas dentro de alguna maquina vecina, las herramientas de seguridad pueden tomar medidas para mitigar estas amenazas, incluyendo bloqueos de IP y la correspondiente salida de línea de todos los servicios asociados a esta. [19]

4.2. RIESGOS TECNICOS

4.2.1. Comunicaciones externas: Esta clase de riesgos se puede catalogar como los riesgos a los cuales todo servicio se expone a internet.

4.2.2. Debilidad en el cifrado: En la capa de aplicación se encuentra una debilidad frecuente cuando se presta un servicio SaaS o se intenta acceder a la plataforma de administración de algún servicio ya que los datos que viajan quedan expuestos a ser interceptados. Buscando ofrecer una transmisión segura de los datos se utilizan los protocolos SSL/TLS sin embargo las múltiples vulnerabilidades encontradas en el primero de estos protocolos, las malas configuraciones y las opciones de acceso que dan los exploradores web permite a un atacante obtener datos de un usuario incauto. [20]

4.2.3. Distributed Denial Of Service (DDOS): Hace referencia a uno de los ataques más comunes en la actualidad cuando se busca la interrupción de la prestación de un servicio basado en la saturación del objetivo enviado peticiones recurrentes que no corresponden a tráfico legítimo. Como consecuencia de esto, la empresa puede presentar, además del aislamiento en red mientras el problema es solucionado, pérdidas financieras y la afectación de su imagen. [21]

4.2.4. Intercepción de datos: Las fallas en la configuración de seguridad y los altos volúmenes de tráfico que se manejan hacen parte de un escenario donde ataques como sniffing, spoofing o MITM generan una constante preocupación. [22]

4.2.5. Interfaz web comprometida: En esta sección, los problemas a los que está asociada la computación en la nube son los mismos que los presentes en cualquier aplicación web. El grupo *Open Web Application Security Project* (OWASP) en [23] definió las siguientes como las amenazas más críticas:

- Inyección: Ocurre cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. Esto se puede realizar en SQL, OS, y LDAP.

- Pérdida de autenticación y gestión de sesiones: Se presenta debido a las implementaciones erróneas de aplicaciones relacionadas con la autenticación, permitiendo a los atacantes comprometer contraseñas.

- Secuencia de comandos en sitios cruzados (XSS): Se presentan cuando una aplicación toma datos no confiables y los envía a un navegador web sin una validación o codificación apropiada. Lo anterior permite a los atacantes ejecutar secuencias de comandos en el navegador de la

víctima con el objetivo de tomar información de esta o dirigirlo a un sitio malicioso

- Referencia directa insegura a objetos: Se presenta cuando un desarrollador expone una referencia a un objeto de implementación interno sin un control de acceso. Lo anterior deja expuesto el objeto a manipulación por parte de los atacantes.

- Configuración de seguridad incorrecta: Definir herramientas correctas para la aplicación a proteger, mantener el software actualizado y realizar las reconfiguraciones a las que haya lugar son prácticas que deben ser tenidas en cuenta ya que no se puede tratar la seguridad informática como algo estático.

- Exposición de datos sensibles: La falta de protección de datos puede dejar en exposición credenciales de autenticación, bases de datos que pueden ser robadas o modificadas para cometer fraudes u otros delitos.

- Ausencia de control de acceso a las funciones: La falta de verificación de permisos y los accesos que tienen los usuarios influencian permite a los atacantes realizar peticiones o consultas sin la autorización apropiada.

- Falsificación de peticiones en sitios cruzados: Este tipo de ataques obliga a una víctima autenticada a enviar una petición HTTP falsificada, donde se incluyen la sesión del usuario y cualquier otra información de autenticación a una aplicación web vulnerable.

- Uso de componentes con vulnerabilidades conocidas: Las librerías, los frameworks y demás módulos de software funcionan, por lo general, con

todos los privilegios. De esta forma, si se ataca un componente vulnerable esto podría facilitar la intrusión en el servidor o una perdida notable de datos.

- Redirecciones y reenvíos no validados: Los redireccionamientos realizados a otros sitios web, y la utilización de datos no confiables para determinar la página de destino hace posible ataques de phisihing o infección por malware.

4.2.6. Comunicaciones internas.: Los riesgos en comunicaciones internas se encuentran dirigidos a las características específicas generadas en las diferentes arquitecturas de nube presentados.

4.2.6.1. Red virtualizada: La arquitectura en nube no solo comparte recursos de forma física, muchas de las plataformas de virtualización tienen la habilidad de emular el comportamiento de switches y diferentes configuraciones de red basándose en software, permitiendo de esta manera una mejor comunicación más directa y eficiente. Sin embargo el tráfico sobre estas redes no es visible a los dispositivos de seguridad en la red física; por lo anterior se recomienda una doble protección en estos ambientes. [24]

4.2.6.2. Fallas en la configuración: Las fallas en las configuraciones, especialmente las que presentan riesgos de seguridad, tienen una relevancia significativa. Las aplicaciones mal configuradas pueden comprometer radicalmente la seguridad de los usuarios, aplicaciones y todo el sistema en general. [25]

4.2.7. ARQUITECTURA

4.2.7.1. Máquinas Virtuales: Principalmente existen 2 clases de ataques que pueden ser ejecutados en ambientes virtualizados:

- **Escape de Máquina Virtual:** En esta clase de ataque se busca romper la capa de aislamiento en el que se encuentran las máquinas virtuales con el objetivo de poder obtener los permisos de hypervisor en vez de los privilegios por defecto de la máquina. De esta forma el atacante obtiene acceso al equipo y a las otras máquinas virtuales que se encuentran corriendo en esta. [26]
- **Rootkit en Hypervisor:** Se encarga de iniciar un hypervisor falso con el fin de que el sistema asuma que él es quien se encarga de controlar los recursos, sin embargo, en la realidad esta máquina no existe. Este hypervisor también crea un canal para ejecutar código no autorizado en el sistema. En resumen, esta clase de ataque permite a un atacante hacerse con el control de cualquier máquina virtual que se encuentre corriendo y manipular las actividades en el sistema.

Se ha hecho alusión a que las actividades maliciosas dentro de la nube pueden tener un impacto en la integridad, disponibilidad y confidencialidad de cualquier tipo de datos; que las malas configuraciones y los fallos de seguridad permiten que los privilegios no sean otorgados de la forma más adecuada y expongan información que sea confidencial a un usuario estándar. Los problemas anteriores en el abuso de los privilegios o la mala distribución de estos ponen en riesgo no solo la información allí almacenada sino la reputación de la empresa y la confianza del consumidor. [27]

4.2.7.2. Recursos Insuficientes: Un diseño inadecuado de la demanda de recurso por parte de las máquinas virtuales llevará necesariamente a que se vea comprometida la calidad del servicio prestado y todos los servicios asociados.

4.2.7.3. Problemas de Almacenamiento: Con el almacenamiento se pueden presentar dificultades como las mencionadas a continuación:

- **BackUp de Datos:** El correcto manejo de los respaldos, su periodicidad y su custodia, permiten a una organización o proveedor de servicio manejar un correcto plan de gestión de incidentes, mitigando de esta forma los tiempos de respuesta frente a un suceso no esperado. Por todo lo descrito anteriormente, dentro de los acuerdos de prestación de servicio y por iniciativa organizacional, se debe ser consciente de la importancia de estos. [28]
- **Borrado Inseguro de Datos:** Para mantener la confidencialidad y la integridad de la información incluso una vez cumplida su vida útil, se debe asegurar que los datos sean eliminados siguiendo un estándar definido para asegurar que esta no resulte en manos de organizaciones o personas ajenas a los titulares de la misma. [29]
- **.Privacidad e Integridad:** Uno de los puntos más críticos a la hora de realizar un diseño de la nube se centra en garantizar en todo momento la integridad de los datos del cliente. Para lograr esta meta es claro que la implementación de sistemas redundantes es más que necesaria, independientemente de la tecnología utilizada. La privacidad de los datos es otra de las grandes preocupaciones de los usuarios a la hora de contratar servicios en la nube. Para garantizar la privacidad se debe definir dentro de los acuerdos de servicio los términos de privacidad

utilizados por el proveedor y evaluar si estos se ajustan a lo que se requiere. [30]

4.3. RIESGOS LEGALES

La parte legal toma relevancia dentro de los servicios en nube ya que estos rigen la prestación del servicio, que se puede y que no se puede realizar dentro de lo contratado y que implicaciones legales puede generar el mal uso de los recursos.

4.3.1. Acuerdo de Prestación de Servicios: Esta clase de acuerdos especifica las condiciones en las que el servicio será prestado. Los lineamientos presentan entre otras cosas, el desempeño mínimo que se debe proveer, las consecuencias en caso de presentarse una brecha y las acciones que las contrarresten. Con el fin de evitar cualquier inconveniente, los usuarios deben detallar sus necesidades de seguridad para las tareas a desarrollar. [31]

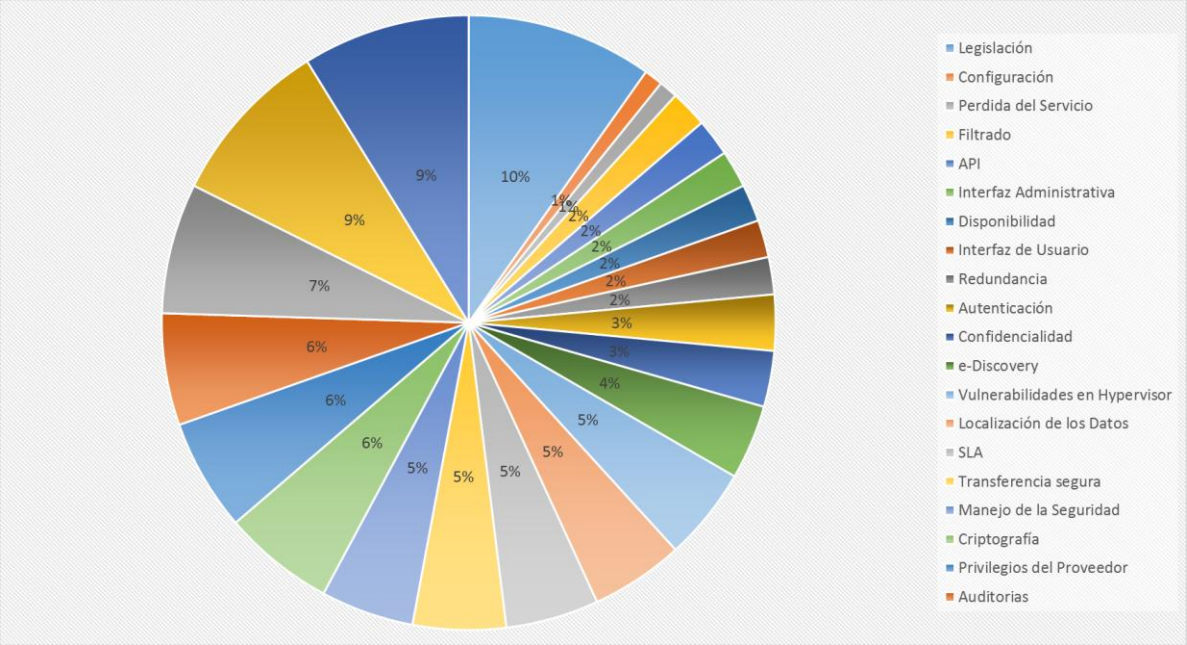
4.3.2. Legislación Variable: A la hora de utilizar un servicio en nube, como se referencia en [17], resulta fundamental conocer donde estará almacenada la información y la clase de datos a almacenar. Lo anterior debido a que algunos países tienen una legislación inestable en cuanto al almacenamiento de datos y las leyes e implicaciones que estas traen se encuentran en constante cambio; de esta forma, por citar un ejemplo, los datos del usuario pueden ser consultados por varias partes sin el consentimiento del usuario

4.3.2.1. Protección de Datos.: Este numeral, además de encontrarse directamente relacionado con el anterior ya que una legislación clara y en pro de la privacidad de los datos favorece su protección, también se basa en la forma como los proveedores suministran

información clara acerca del tratamiento que se le da a sus datos y las medidas de seguridad destinadas para su protección [17]. Esto en función de tener al usuario informado en todo momento y, cuando llegue a ser necesario, este realice cambios necesarios en la configuración buscando mitigar posibles brechas detectadas por el proveedor.

En general, según [32] los problemas identificados que preocupan más a las empresas cuando se hace referencia a computación en la nube son los problemas legales con los 10%, seguidos por la pérdida de control y el aislamiento de los sistemas, lo anterior se ve reflejado en la figura 9.

FIGURA 9. Relevancia de las vulnerabilidades para las empresas



FUENTE: Adaptado de [32]

5. ANALISIS DE SOLUCIONES DE SEGURIDAD PARA LA NUBE.

Hasta el momento se han tenido en cuenta los riesgos a los cuales se encuentran expuestos los servicios y arquitecturas en nube. Sin embargo, las buenas prácticas y el uso de herramientas de seguridad adecuadas permiten la mitigación de muchos de los riesgos anteriormente descritos.

A continuación se hará enfoque en diversos métodos que disminuyen las superficies de ataque en la computación en la nube. Dichas estrategias varían de acuerdo a las necesidades de protección buscadas por el proveedor y por el cliente. Si bien no existe una solución que abarque todos los problemas, el uso en conjunto de estas resulta una manera efectiva de incrementar la seguridad.

5.1. ENTORNO DE RED

Como se describió previamente, la nube tiene problemas muy similares a los de una red común cuando esta sale a internet. Es por lo anterior que se presentan en esta área diversas soluciones dedicadas a evitar el tráfico no autorizado y a filtrar el contenido que debe ingresar a la red.

Un ejemplo de lo anterior se encuentra en un sistema denominado Network Intrusion Detection and Contermeasure Selection (NICE) el cual se define como una herramienta que permite un acercamiento a la red virtual permitiendo un sistema de detección de intrusos reconfigurable. Independientemente donde se encuentre posicionado el atacante este modelo se enfoca en un entorno de detección de ataques en una red virtual buscando aumentar la resistencia de una maquina evitando que sea comprometida por el atacante. Este modelo puede ser usado en la infraestructura de nube tipo IaaS.

El funcionamiento de NICE, según lo descrito en [33], se basa en un software instalado en cada uno de los servidores de la nube asociando estos con un centro

de control mediante un canal de comunicación aislado y seguro; el centro de control es el encargado de desplegar las medidas necesarias para contrarrestar un posible ataque basado en los resultados del analizador de datos.

Otra de las herramientas de monitoreo y prevención de intrusiones desarrolladas para ambientes de sistemas en nubes es SnortFlow [34] el cual se encuentra basado en tecnologías libres como OpenFlow y Snort. La combinación de estas tecnologías da como resultado que el servidor que corre el aplicativo sea capaz de evaluar la seguridad de la red mediante un generador de políticas que analiza el tráfico, generar alarmas en caso de presentarse una desviación en las políticas asignadas y generar una alerta.

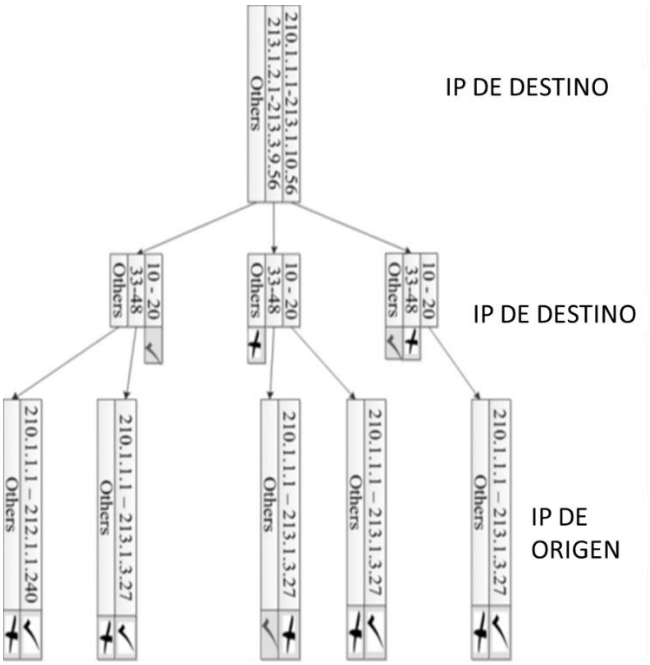
El filtrado de contenido mediante firewalls es uno de los métodos más aplicados en los entornos de red; este tipo de filtrado puede ser realizado en los ambientes virtuales mediante la implementación de software (por ejemplo Iptables) o hardware. Uno de los modelos propuestos para el filtrado de contenidos se basa en lo que los investigadores llamaron tree-rule firewall donde se destaca que el uso de estas soluciones puede presentar limitaciones que pueden derivar en problemas de seguridad, estas limitaciones son:

- Reglas en la sombra
- Cambios de posición entre reglas
- Reglas redundantes
- Dificultan en el uso
- La búsqueda secuencial puede llevar a un problema de velocidad

En [35] con el fin de contrarrestar este tipo de problemas, los investigadores proponen con su sistema que solo la información que llega a ser relevante sea analizada en los diferentes niveles y de esta forma se obtenga en poco tiempo una respuesta de qué hacer con el paquete. Para poderlo explicar de una forma más adecuada, en el momento en que el paquete llega al firewall, este considera la información de las cabeceras donde se encuentre la IP de destino, puerto de

destino y la dirección IP de origen, lo dicho anteriormente se ve reflejado en la figura 10:

FIGURA 10. Tree rule firewall



FUENTE: Adaptado de [35]

En la tabla 4 se condensa la información descrita previamente.

TABLA 4. SOLUCIONES DE SEGURIDAD EN EL ENTORNO DE RED

TRABAJO	***Propósito del Esquema***	***Características de seguridad***
NICE	Aplicación para prevención de intrusos	• Framework para la detección de Intrusos en redes virtuales • Centralizar el control del tráfico virtual.
SnortFlow	Aplicación para prevención de intrusos	• Mezcla entre snort y Openflow • Resguardar la red frente a posibles intrusiones
Tree-rules Firewall	Seguridad en el entorno de red	• Eliminar redundancia de reglas • Estructura basada en 3 reglas • No se ejecutan las reglas de forma secuencial

5.2. ENTORNO VIRTUAL

La virtualización cuenta con múltiples beneficios debido especialmente a una mejor distribución de los recursos del servidor y su capacidad de contener diferentes máquinas en un solo equipo físico, por esto resulta fundamental en los

servicios de nube basados en IaaS al igual que en las partes back-end de los modelos PaaS y SaaS. [36]

Para poder mejorar la seguridad en este campo resulta vital fortalecer todo el ambiente que rodea la máquina virtual, esto se logra aplicando controles como los mencionados en el numeral anterior incluyendo además antivirus, protección a las aplicaciones web y realizando un constante monitoreo del tráfico y las actividades que se presentan en la red.

El manejo del hypervisor toma una relevancia marcada en este punto ya que las buenas prácticas, basadas en la correcta configuración y manejo disminuyen los riesgos asociados a un posible ataque que vaya en detrimento del rendimiento.

El hypervisor debe poder monitorear los componentes fundamentales que puedan verse afectados o ser objetivos de diversos ataques, esto se realiza verificando periódicamente el checksum de archivos ejecutables y librerías. Esta herramienta puede ser configurada para responder cuando se tienen suficientes indicios de que un ataque está siendo realizado. [37]

Con el fin de reducir la superficie de ataque a la cual se encuentra expuesto el hypervisor se propone en [38] la reducción de sus privilegios. Usando *Kernel-based Virtual Machine* (KVM) como hypervisor, este estudio busca obtener dos componentes principales. El primero de ellos es el componente DeHype que se encuentra integrado a cada uno de los huéspedes de la máquina virtual, es el componente con bajos privilegios ejecutado en modo USER y se acerca al 93% del aplicativo KVM, aquí se encuentran los módulos que requieren poca o ninguna interacción con el sistema operativo. Los módulos que no pudieron ser ejecutados como usuario se encuentran en un módulo separado llamado HypeLet. De esta forma cuando se realiza un llamado al sistema KVM, este se procesa de manera local inicialmente y solo de ser necesario se realiza una solicitud al HypeLet.

Un enfoque muy similar se encuentra en el diseño del Hyperlock [39], el cual aísla las máquinas virtuales del hypervisor y reduce el pool de instrucciones para

realizar una ejecución segura del mismo. Adicional a esto también se propone una técnica denominada hypervisor shadowing, que crea un hypervisor fantasma para cada una de los hosts hospedados de manera de que si llegase a ser afectado solo se afecte la máquina a la que se encuentra asignado, disminuyendo posibles pérdidas.

Un poco más radical resulta la eliminación completa del hypervisor, esto se encuentra propuesto bajo la arquitectura NOHYPE, descrita en [40], donde se elimina de esta forma toda la superficie de ataque. Esta arquitectura se compone de 4 características, la primera de ellas es realizar la asignación de memoria y procesador requerido por el cliente con antelación, eliminando de esta manera el uso dinámico que da el hypervisor a los recursos garantizando la disponibilidad de los recursos contratados en todo momento; otra de las claves de esta propuesta es usar únicamente dispositivos de entrada y salida virtualizados, lo anterior haciendo referencia a las conexiones de red, almacenamiento y potencialmente a tarjetas gráficas; el proceso de redescubrir el sistema cuando la máquina virtual arranca mediante el uso de un hypervisor temporal, es la tercera característica de este modelo y se logra almacenando en cache las modificaciones del sistema y características básicas del mismo; la última de las características se basa en un enlace más directo entre las máquinas virtuales y el hardware como lo es el uso dedicado de núcleos y hardware en general.

TABLA 5. SOLUCIONES DE SEGURIDAD EN EL HYPERVISOR

TRABAJO	***Propósito del Esquema***	***Características de seguridad***
DeHype	Técnica para reducir la superficie de ataque en el hypervisor.	• Principio de menor privilegio • Previene el escape de datos entre el kernel y el espacio asignado al usuario

TRABAJO	*Propósito del Esquema*	*Características de seguridad*
HYPERLOCK	Aislar el hypervisor de los hosts	• Hypervisor invisible para cualquier máquina virtual • Acceso controlado al sistema principal.
NOHYPE	Virtualización sin un hypervisor	• Eliminación del hypervisor • Recursos pre asignados

Hasta el momento se ha hablado del hypervisor exclusivamente, sin embargo, en los ambientes virtuales se corre un riesgo durante la ejecución para mitigar estos riesgos en [41] se presenta una solución conocida como CloudVisor la cual busca generar seguridad a los recursos de la máquina virtual. Durante la ejecución, todas las comunicaciones de control entre el hypervisor y la máquina virtual son interceptadas para realizar chequeos de seguridad. Un ejemplo de estos muestra la capacidad del CloudVisor para cifrar los registros y tomar solo los registros que lleguen a ser requeridos para un análisis de la herramienta.

Otro de los modelos que permite brindar seguridad durante la ejecución se denomina HyperCoffer y es descrito en [42]. Este modelo llama la atención en la medida en que solo confía en el procesador y considera todo lo demás como componentes no seguros. HyperCoffer involucra tanto el hardware como el software ya que utiliza una tecnología de procesamiento segura para cifrar los datos de la memoria. Adicionalmente, para evitar ataques de máquinas cruzadas cada línea de cache es marcada con un identificador único de VM. Otra de las características que se da con el fin de mantener integra las comunicaciones entre esta y el hypervisor son cifradas y, al igual que el modelo anterior, se expone solo la información necesaria.

TABLA 6. SOLUCIONES DE SEGURIDAD EN LA MÁQUINA VIRTUAL DURANTE SU EJECUCIÓN

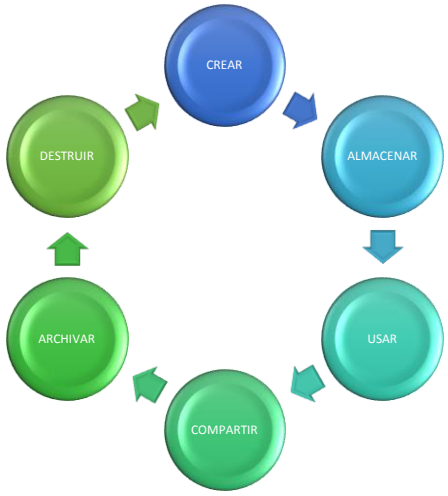
TRABAJO	***Propósito del Esquema***	***Características de seguridad***	***Privacidad***	***Integridad***
CloudVisor	Asegurar la ejecución segura en los ambientes virtuales	• Virtualización anidada • Criptografía • Separación de la seguridad y las tareas de virtualización	SI	SI
HyperCoffer	Asegurar la ejecución segura en los ambientes virtuales	• Virtualización anidada • Criptografía • Separación de la seguridad y las tareas de virtualización • Seguridad frente al rollback en las máquinas virtuales.	SI	SI

5.3. ALMACENAMIENTO DE DATOS

El manejo del ciclo de vida de la información es un campo maduro que expone en 6 pasos las etapas por las que la información pasa desde su creación hasta su

eventual destrucción [43]. Si bien es una progresión lineal, una vez creada la información esta necesariamente no deberá pasar por todas la etapas.

FIGURA 11. Ciclo de vida de la información.



FUENTE: Adaptado de [43]

De lo descrito en la figura 11, a continuación se presenta una breve descripción de cada una de las fases:

Crear: Es la generación de nuevo contenido digital o la alteración de contenido existente.

Almacenar: Ocurre comúnmente con la creación y consiste en ubicar la nueva información en un repositorio de almacenamiento.

Uso: Los datos son vistos, procesados, o se encuentran involucrados en una actividad que no incluye la modificación de estos.

Compartir: La información es accesible a otros y puede ser consultada según lo estipulado por el propietario de esta.

Archivar: Los datos dejan de estar en uso y entran en un almacenamiento de largo plazo.

Destrucción: Los datos son destruidos de forma permanente ya sea de manera física o digital.

De la misma forma como se tiene un ciclo de vida, también es importante definir políticas y procedimientos sobre el uso de la información generada. Para lo anterior se debe tener en cuenta:

- Clasificación de la Información
- Políticas de Manejo de Información
- Localización y Jurisdicción aplicable.
- Autorizaciones, Pertenencia y Custodia.

El valor de los datos para una empresa varía según su importancia y criticidad para el negocio, independientemente de su ubicación mantener segura esta información es prioritario para las organizaciones.

La seguridad de los datos incluye los controles y las tecnologías usadas para el manejo de la información. Esto puede ser realizado en 3 momentos para detectar y prevenir la migración de datos en la nube, proteger los que se encuentran en tránsito hacia esta y asegurarlos una vez se encuentra en la nube. [36]

La integridad de los datos es, para el usuario, una de las preocupaciones más grandes cuando almacena información en la nube. Uno de los puntos clave para los prestadores del servicio debe ser entonces el poder garantizar la detección de modificación no autorizada y la corrupción de los datos. Para lograr tal fin, se han desarrollado estudios para poder verificar que los datos se encuentran correctos sin conocer explícitamente todos los datos mediante el uso de códigos correctores de borrado y tokens homomorfos que al ser integrados derivan en un protocolo challenge-response con el fin de verificar la concordancia de los datos e identificar cualquier comportamiento erróneo de los servidores. [44]

Además de buscar la integridad de los datos, la confidencialidad de los mismos al ser almacenados es de gran importancia, es por esto que se debe asegurar que los sistemas del proveedor no revelarán la información transmitida a nadie excepto a los procesos designados cuya ejecución es bien conocida [45]. En otras palabras realizar un proceso de comunicación “ciega” dentro de los servidores mediante un canal seguro entre los procesos dedicados y ocultos para el resto, incluso procesos de administrador o root.

Otro de los métodos para asegurar la información almacenada se presenta bajo el nombre de SecCloud, que es un protocolo seguro de almacenamiento que utiliza procesos de cifrado para su fin [46]. Usando criptografía basada en emparejamientos se generan las llaves de usuario, nube y un ente de confianza. Su funcionamiento hace que la información del usuario se divida en un número m de mensajes que es firmado por la entidad certificadora, estos mensajes son enviados a la nube de manera cifrada con la clave de sesión (calculado mediante el protocolo Diffie-Hellman). Una vez se obtiene la información en la nube, se descifra la información, se verifica la firma y se almacena la información en el espacio asignado al usuario.

El uso de los analizadores de contenido, normalmente usados para prevenir la pérdida de información en las bases de datos, permite identificar información sensible en el contenido almacenado por lo que, cuando se trata de asegurar los datos, resulta bastante útil. Logrando que la organización pueda definir políticas basadas en el tipo de información, estructura o clasificación. Una vez definidas las políticas, se procede a revisar la existencia de violaciones a estas. [36]

TABLA 7. SOLUCIONES DE SEGURIDAD EN EL ALMACENAMIENTO DE DATOS

TRABAJO	*Propósito del Esquema*	*Características de seguridad*	*Privacidad*	*Integridad*
Secure and Dependable Storage Services	Metodología de seguridad para datos almacenados	• Redundancia de datos	NO	SI
A comined approach to ensure data security in cloud computing	Esquema de seguridad para los datos almacenados	• Cifrado simétrico SSL • Control de datos	SI	SI
SecCloud	Protocolo para almacenamiento seguro y privacidad	• Autoridad de confianza • Verificación de firma	SI	SI

5.4. API Y APLICACIONES DE NUBE.

Dentro de la arquitectura SaaS, es común encontrarse con aplicaciones desarrolladas para diversos fines, por esto, es importante mantener el acceso seguro a las aplicaciones de accesos no autorizados. En [47] se propone el uso de un protocolo denominado *Authentication, Authorization, Accounting* (AAA) el cual

filtra las solicitudes de acceso ilegitimas empleando control de acceso a la red. Cada requerimiento es recibido inicialmente por una aplicación denominada AAA cloud y reenviada a un Diameter Server, encargado de revisar si los parámetros de autenticación y decide si se entrega o se deniega la solicitud de acceso a la aplicación. Adicionalmente, esta aplicación, al igual que muchas otras que basan el acceso mediante HTTP, cifra y asegura las credenciales del servidor mediante SSL/TLS.

Un entorno seguro multiarquitectura denominado como *Security as a Service* (SECaaS) desarrollado en [48], recomienda diferentes controles de seguridad enfocados hacia las necesidades del usuario. Una vez conocidos los requerimientos de seguridad por parte del usuario la herramienta identifica los proveedores que ofrecen estos servicios. Una vez filtrados los proveedores la herramienta registra la aplicación del usuario con estos proveedores.

Proveer una plataforma segura para las aplicaciones presenta un desafío constante para los CSP, Buscando generar la confianza suficiente para el usuario, una de las alternativas que surgen es el uso de la criptografía de curva elíptica que puede ser usada para crear de manera más eficiente y rápida las llaves públicas. El uso del modelo ECC junto con el *Trusted Platform Module* (TPM) asegura la integridad de la plataforma antes de mover cualquier aplicación allí. [49]

TABLA 8. SOLUCIONES DE SEGURIDAD EN LAS API Y APLICACIONES DE NUBE

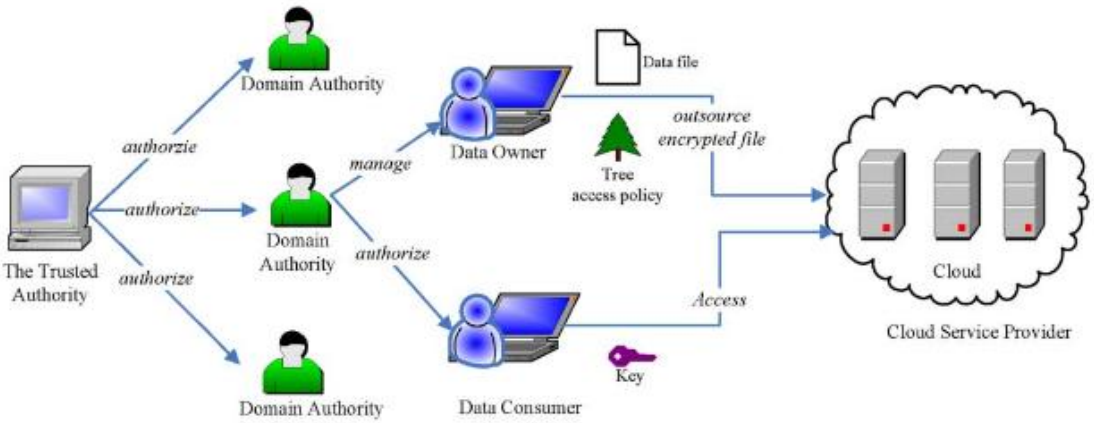
TRABAJO	***Propósito del Esquema***	***Características de seguridad***
Diameter-AAA	Control de acceso a las aplicaciones en la nube	• Autenticación • Autorización

TRABAJO	*Propósito del Esquema*	*Características de seguridad*
SECaaS	Seguridad como servicio en aplicaciones en la nube	• Mantener monitoreados los servicios
TPM y Criptografía de Curva Elíptica	Esquema de confiabilidad en la nube	• Integridad en la aplicación • Integridad en la plataforma •

5.5. CONTROL DE ACCESO.

Buscando mejorar el control de acceso a los servicios contratados con el CSP se han adelantado diferentes estudios que permiten garantizar al usuario que solo quien él decida puede acceder a la información bajo los permisos otorgados. Muchos de estos esquemas manejan la criptografía basada en atributos, sin embargo, en muchos casos la implementación de este modelo carece de flexibilidad cuando se implementan políticas de acceso complejas. Una de las mejoras plantea el uso de jerarquías en la estructura de usuarios, bajo el nombre *Hierarchical Attribute-set-based* encryption (HASBE) [50]. Para entender más claramente su funcionamiento en la figura 12 se muestra el modelo. En él interactúan 5 actores: el CSP, los dueños de los datos, quienes consultan los datos, autoridades de dominio y una autoridad de confianza.

FIGURA 12. HASBE como solución de control de acceso



FUENTE: Adaptado de [50]

La estructura jerárquica cuenta con la autoridad de confianza como la autoridad principal y la responsable de manejar los dominios de alto nivel; a su vez, cada uno de estos dominios se puede dividir en subdominios dependiendo del cliente. Quienes revisan o suben la información, pueden ser trabajadores de la organización cliente por lo que cada dominio de autoridad es responsable de manejar los accesos. En este modelo, ni quienes suben de la información ni quienes la consultan mantendrán una conexión continua con la nube.

Otra de las funciones de la autoridad de confianza es la generación y distribución de los parámetros del sistema y de la clave maestra para cada uno de los dominios de autoridad. Las llaves privas y publicas emitidas a los usuarios por el dominio de autoridad es de la forma de árbol jerárquico, donde cada elemento es un atributo o un conjunto de estos. El control de acceso se encuentra definido de igual forma por la estructura de árbol jerárquico. La llave de cifrado de los datos es protegida mediante HASBE usando la estructura de clave de acceso que especifica las políticas de control de acceso y los atributos.

TABLA 9. SOLUCIONES DE SEGURIDAD EN LAS API Y APLICACIONES DE NUBE

TRABAJO	***Propósito del Esquema***	***Características de seguridad***
HASBE	Esquema de control de acceso para la nube	• Jerarquía Confiable • Re-cifrado • Autenticación de usuario • Privacidad

6. CRIPTOGRAFIA

Dentro del análisis que se ha llevado a cabo sobre las soluciones de seguridad planteadas para los servicios en nube, resalta especialmente el uso de la criptografía como mecanismo de seguridad.

Basado en [51], en el contexto puramente histórico la criptografía inicia con la historia, con el origen del lenguaje escrito. Sin embargo, la criptografía moderna inicia con la comunicación eléctrica. En la literatura anglosajona, la criptografía es ilustrada por personajes reconocidos: Alice y Bob. Por razones bíblicas, la representación maligna usualmente toma el nombre de Eva. De acuerdo con lo expresado popularmente, Alice y Bob desean comunicarse de forma segura buscando la manera de protegerse de Eva. Es en este punto, donde la criptografía se concentra en 3 paradigmas fundamentales:

- Confidencialidad: La información no se debe filtrar fuera de las partes que la mantienen.
- Autenticación: La información debe ser clara acerca de quién es el autor (firmas de autenticación, control de acceso)
- Integridad: La información debe ser protegida contra cualquier intento malicioso de modificación.

Adicional a lo anterior, la criptografía se caracteriza a lo largo de tres dimensiones independientes:

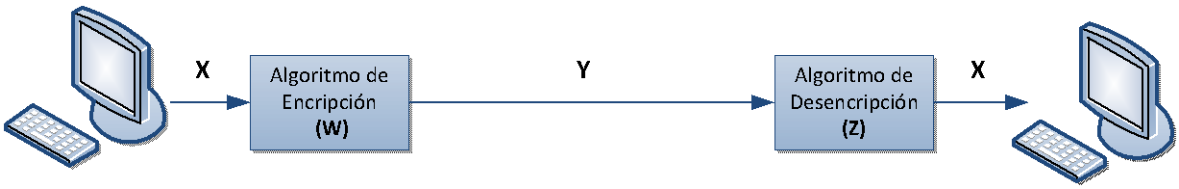
- El tipo de operaciones usadas para transformar un texto plano en uno cifrado, todos los algoritmos de cifrado se encuentran basados en dos principios generales: sustitución, donde cada elemento en el texto plano (letra o bit) es transformada en otro elemento y la transposición, donde los elementos del texto son reordenados.
- El número de llaves usadas, Si tanto quien envía la información como quien la recibe usan la misma llave, el sistema es denominado como simétrico, de

llave simple o llave secreta. Por el contrario, si quien envía y quien recibe usan claves diferentes, el sistema es denominado asimétrico, dos llaves o cifrado de llave pública.

- La manera en que el texto plano es procesado, Un cifrado por bloques procesa un bloque de entrada cada vez, produciendo un bloque de salida por uno de entrada. Un cifrado tipo stream procesa los elementos de entrada de forma continua, produciendo cada vez un elemento a la salida que va de la mano con la entrada.

Un modelo de cómo es un sistema de cifrado convencional se representa en la figura 13, Inicialmente se cuenta con un texto plano (X) que pasa por un algoritmo de cifrado (w) obteniendo un texto cifrado (W(X)=Y), el cual solo puede ser descifrado por la contraparte (Z (Y)=X).

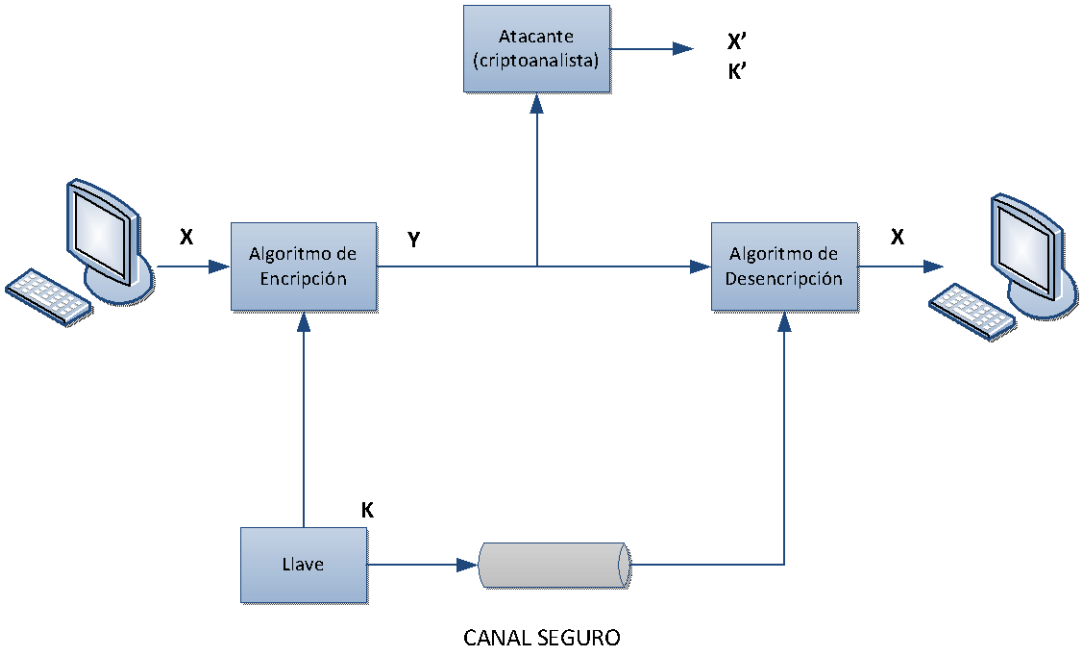
FIGURA 13. Modelo de cifrado convencional



6.1. CRIPTOGRAFIA SIMETRICA [51]

Como se definió previamente, la criptografía simétrica es aquella donde quien genera y quien recibe la información comparten la misma llave, en la figura 14 se observa un modelo convencional de este tipo de criptografía. Adicionalmente, se agrega un atacante (criptoanalista) quien obtiene el mensaje cifrado e intenta obtener el mensaje en texto plano o la clave para su posterior uso.

FIGURA 14. Modelo de cifrado simétrico



Existen diversos algoritmos bajo este modelo, algunos de los cuales ya no son seguros, como ejemplo se citarán 2, DES y AES.

6.2. CRIPTOGRAFIA ASIMETRICA O DE CLAVE PÚBLICA

En [52] se muestra como esta clase de criptografía emerge como un intento de mejora a los problemas que se tienen con el cifrado simétrico. El primer problema es la distribución de la clave, como es conocido esta distribución se realiza de dos maneras: que las partes que se comunican ya hayan compartido la clave, que alguien les distribuyó o mediante el uso de un centro de distribución de claves. Es precisamente este segundo aspecto que de alguna forma niega la esencia de la criptografía: la capacidad de mantener en total secreto la comunicación.

El segundo problema y que aparentemente no tiene relación con el primero son las firmas digitales. Las cuales se hicieron necesarias en la medida que más gente veía la necesidad de firmar los documentos de forma que no existiera duda que es quien en realidad dice ser.

Los algoritmos asimétricos se basan en una llave para el cifrado y en una llave diferente, pero relacionada, para realizar el descifrado. Estos algoritmos tienen una característica singular:

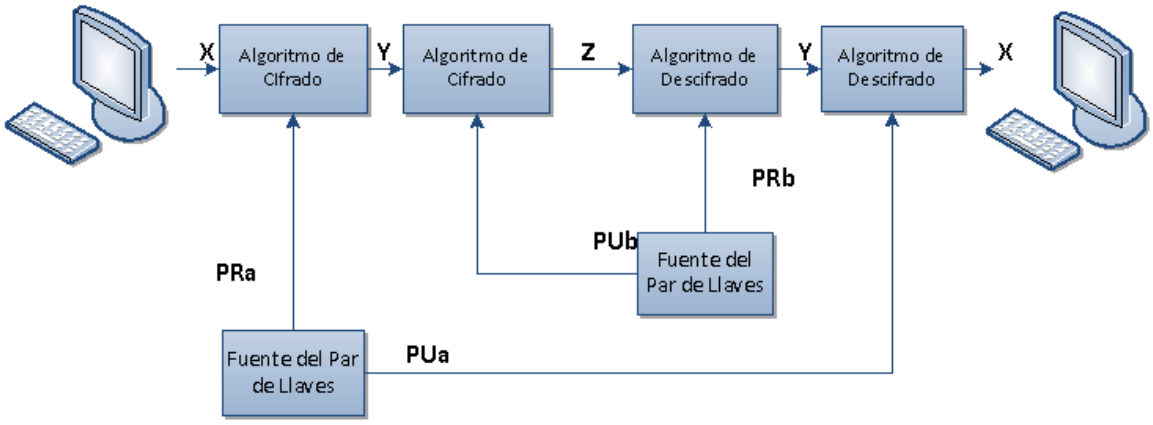
- Es computacionalmente improbable determinar la llave de descifrado teniendo solo el conocimiento del algoritmo criptográfico y la llave de cifrado.

Adicionalmente, en algunos algoritmos como el RSA se puede ver la siguiente característica:

- Cualquiera de las dos llaves involucradas puede ser usada para el cifrado, dejando la otra para el descifrado

En la figura 19 se describe un sistema de cifrado de llave pública del modo autenticación y privacidad. En este modelo se inicia cifrando el mensaje usando la llave privada de la fuente, de esta forma se genera una firma digital (Y) y genera el no repudio; el siguiente paso es realizar un nuevo cifrado (Z), esta vez con la llave publica del destino. Para realizar el descifrado de Z quien posee la clave privada con la que fue cifrado el mensaje Y y conoce a su vez la clave pública del origen

FIGURA 15. Modelo de cifrado asimétrico



La aplicación de este algoritmo, como se mencionó anteriormente, se puede realizar de 3 formas:

- Cifrado/Descifrado
- Firma Digital
- Intercambio de Llaves

De lo mostrado en la figura 19 es valioso destacar qué:

- No existe dificultad computacional para un destino generar un par de llaves.
- Para el origen resulta sencillo conocer la llave pública del destino y procesarla con el mensaje a ser cifrado.
- Es imposible que un atacante, conociendo la llave pública del destino, determine la llave privada del mismo o recobre el mensaje original.
- Las llaves pueden ser utilizadas en cualquier orden.

Tal como se mencionó, dentro de los métodos que existen para cifrado de clave se utilizan los métodos de RSA y Diffie-Hellman que se describen a continuación [53]:

6.2.1. RSA : El cliente genera la clave simétrica, lo cifra con la clave pública del servidor y lo envía, el servidor descifra con su clave privada la clave simétrica y así poder establecer la conexión.

A pesar de que es comercialmente muy usado este método, tiene una vulnerabilidad, y es precisamente el hecho de que las claves pública y privada se usan tanto para el cliente enviar la clave simétrica, como el servidor para descifrar ésta, hace que si un atacante logra acceder y tiene la clave privada del servidor, podrá descifrar las sesiones establecidas entre servidor y usuario.

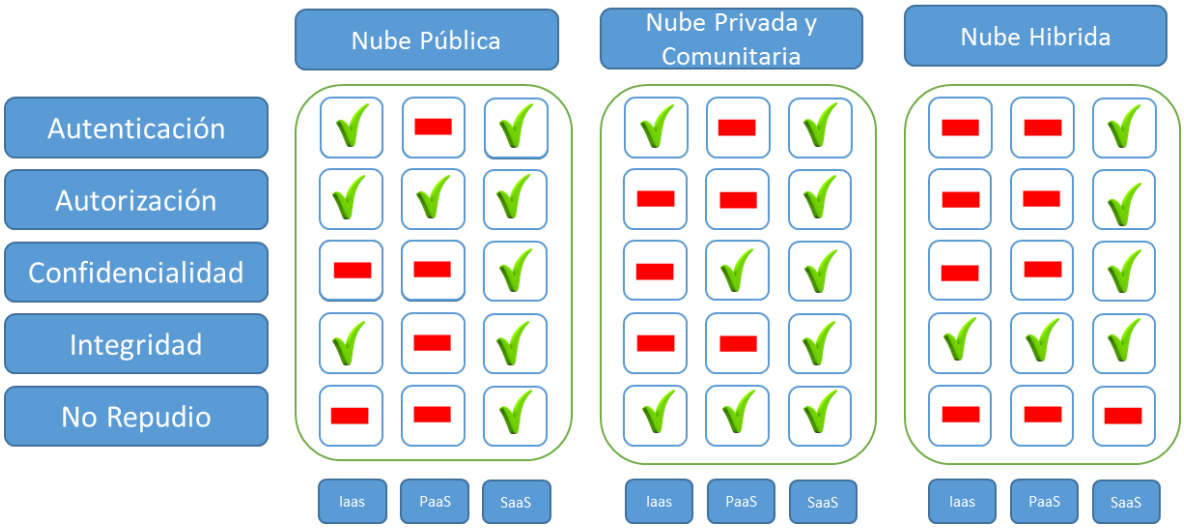
6.2.2. DIFFIE-HELLMAN : A diferencia de RSA, en este método el cliente y servidor acuerdan una clave simétrica secreta compartida, sin necesidad de publicarla en el Handshake, la clave privada se utiliza para verificar el Handshake, pero la clave simétrica nunca es comunicada ni por el cliente, ni por el servidor, esto hará que el atacante pierda cualquier tipo de acceso a la sesión, aunque tenga la clave privada.

A su vez con este método se pueden crear claves simétricas temporales, con esto buscando mitigar riesgos de seguridad de la sesión, básicamente lo que se hace es usar una clave durante una sesión, al momento de iniciar una nueva sesión se borra la anterior y se crea una nueva.

7. SEGURIDAD EN CAPA DE TRANSPORTE.

La conexión juega un papel fundamental en la computación en la nube, si se observa la figura 20 se genera una noción sobre qué tan importante resulta asegurarla y como el cifrado juega un rol fundamental en cada uno de los modelos de servicio. Se debe aclarar que los check son un requerimiento, mientras el menos es opcional.

FIGURA 16. Requerimientos de seguridad para los diferentes modelos de nube



FUENTE: Adaptado de [54]

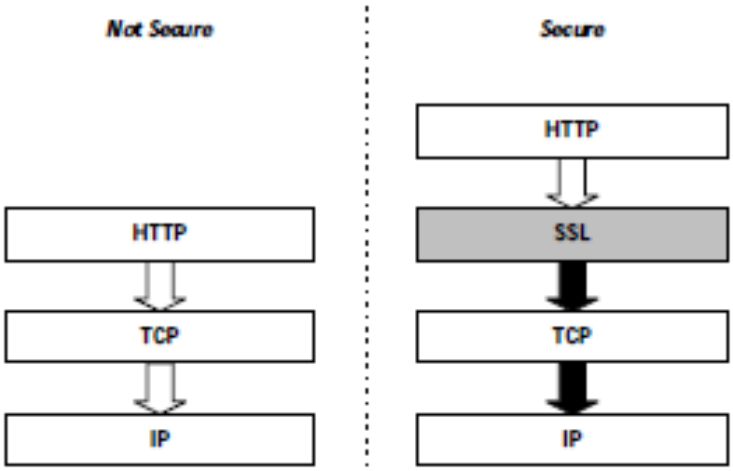
Es por esto que quienes ofrezcan servicios en la nube, ya sea un CSP o los administradores de una nube privada, deben velar por una conexión segura. Este tipo de conexiones se logra mediante protocolos de transporte seguros, los cuales se encargan de crear un canal cifrado entre el cliente y un servidor remoto.

7.1. SSL

El protocolo Secure Socket Layer (SSL) desarrollado por Netscape Corporation con el fin de proveer seguridad y fiabilidad en la conexión entre usuario y servidor web, de manera que el usuario pueda estar seguro que el sitio web que visita es seguro y válido y no uno donde su información pueda ser vulnerada.

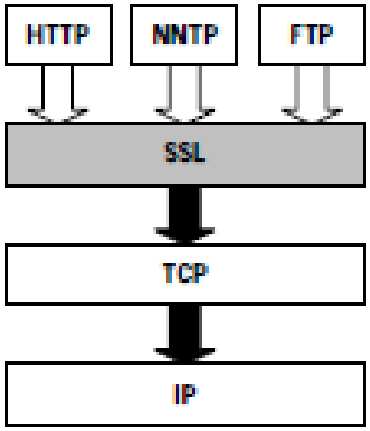
SSL fue diseñado como un protocolo separado solo por seguridad, la figura 21 muestra cómo SSL añade seguridad, ubicándose entre HTTP y TCP respectivamente, sin necesitar cambios significativos en dichos protocolos, pero si añadiendo como ventaja, el hecho de que mejora notablemente la seguridad permitiendo que aplicaciones que no sean HTTP, como NNTP (Net News Transfer Protocol) y FTP (File Transfer Protocol) tal como se ve en la figura 22. [55]

FIGURA 17. Modelo de comunicación segura con SSL



FUENTE: Tomado de [56]

FIGURA 18. Protocolos con seguridad SSL



FUENTE: Tomado de [56]

SSL al ser un protocolo de capas, toma los datos a enviar dividiendo en partes uniformes, aplica un MAC (Message authentication code) de forma opcional comprime los datos, utiliza método de cifrado, para luego transmitir el resultado. Al recibirse el mensaje es descifrado, previamente verificado, descomprime si es necesario y vuelve a ser reagrupado para ser entregado al cliente de manera confidencial. [56]

Los principales objetivos que se planteó el *Internet Engineering Task Force* (ietf) con la versión SSL 3.0 fueron:

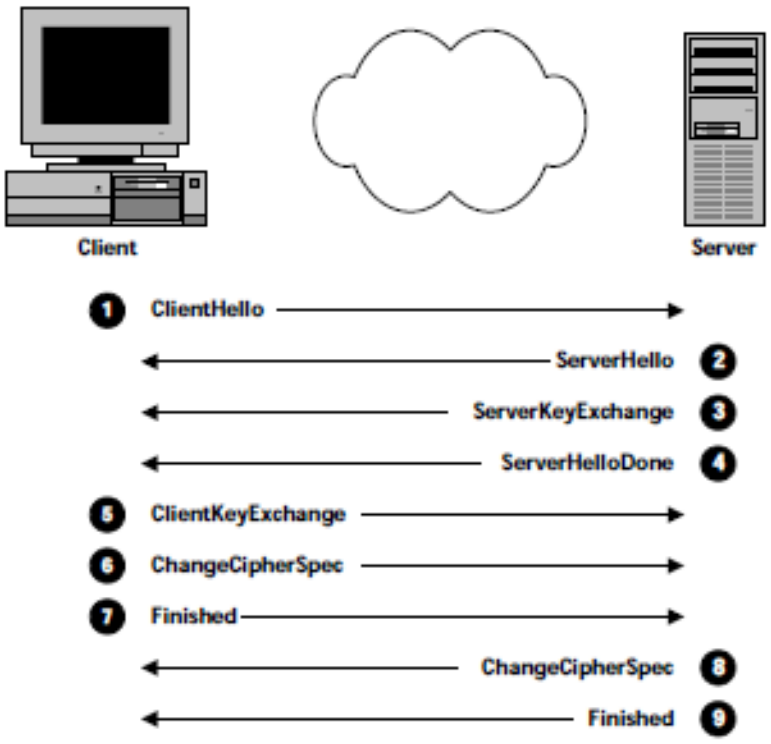
- Seguridad de conexión mediante criptografía
- Interoperabilidad
- Extensibilidad para ser aprovisionado con llaves públicas y métodos de cifrado de ser requerido.
- Eficiencia en los recursos.

7.2. FUNCIONAMIENTO DE SSL

En [57] se describen los dos roles que determina SSL, uno es el cliente quien inicia la comunicación y el servidor que responde y actúa de acuerdo a las opciones que le envíe el cliente, cabe notar que la diferencia entre éstos son las labores que desempeñan al momento de decidir las políticas de seguridad y el tipo de sistema a implementar para la conexión segura.

Para entablar la conexión entre cliente y servidor se determina un canal cifrado de comunicaciones, pues así se hará el intercambio de mensajes, tal como lo muestra la figura 23 y se explica en detalle a continuación.

FIGURA 19. Funcionamiento SSL



FUENTE: Tomado de [57]

- **ClientHello:** Este es el punto inicial, donde se envía un saludo al servidor, revisando estado de conexión y acordar parámetros de seguridad en la transmisión de los datos, en este se incluye también la versión SSL que el cliente actualmente usa y puede soportar.

- **ServerHello:** Es la forma con la que el servidor responde a la solicitud del cliente, se confirman los parámetros expuestos anteriormente.

- **ServerKeyExchange:** Adicional a que con ServerHello, se incluyen los algoritmos criptográficos y tamaños de clave, con ServerKeyExchange se indica la clave pública.

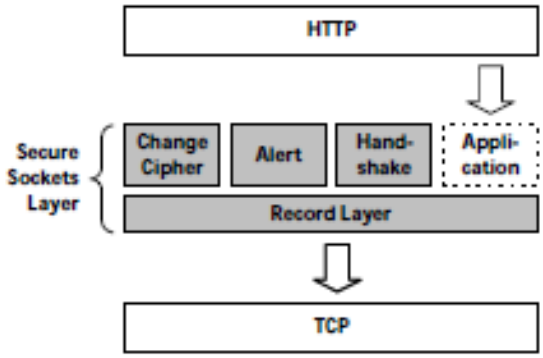
- **ServerHelloDone:** No contiene información relevante, pero es importante para el cliente, ya que el servidor le confirma que puede avanzar a la siguiente fase de negociación.
- **ClientKeyExchange:** Usando la clave pública que ya el servidor ha provisto al cliente, se envía la información privada del cliente, para el algoritmo de cifrado simétrico que ambos van a usar durante la sesión, tal información el cliente la cifra con la misma clave del servidor, para que de esta manera se pueda evitar ataques y/o interceptaciones.

- **ChangeCipherSpec:** Terminada la etapa inicial de negociación, el protocolo SSL puntualiza este mensaje con el fin de confirmar que los servicios de seguridad pueden iniciar.

- **Finished:** Tanto el servidor como cliente acuerdan finalizar el proceso, este mensaje permite corroborar que la etapa de negociación ha sido cumplida satisfactoriamente, y además no ha sido vulnerable la información.

7.2.1. Composición de SSL: Hacen parte de este protocolo diferentes componentes tales como: el protocolo ChangeCipherSpec, protocolo Alert, protocolo Handshake, y aplicaciones como HTTP y Record Layer protocol. En la figura 24 se observa la composición de la capa segura.

FIGURA 20. Composición SSL

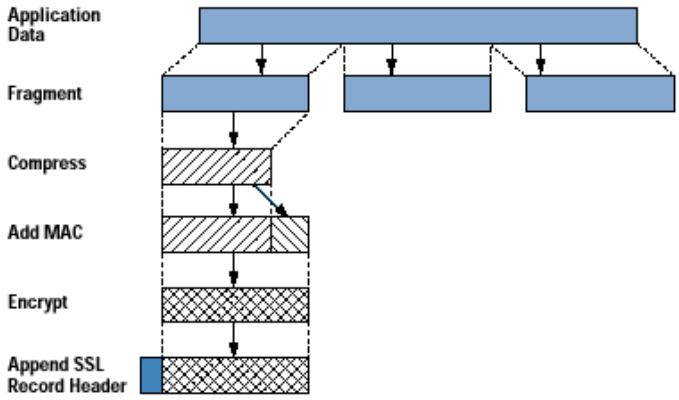


FUENTE: Tomado de [57]

- **Record Layer Protocol :** Recibe los mensajes de ChangeCipherSpec, Alert, Handshake, y aplicaciones, los encapsula, les da el formato apropiado, para luego transmitir a la capa de transporte TCP. A su vez es responsable del servicio de cifrado dado que se encuentre activo, brindando confidencialidad.

  Cuando los mensajes transmitidos entre cliente y servidor, superan la restricción de SSL de que su valor no exceda 214 bytes (16384), SSL Record se encarga de concatenar y fragmentar en grupos de mensajes de 214 bytes o menor tamaño, y de ser necesario son comprimidos, sin perder datos, aplica una MAC sobre los datos usando una clave secreta, y se hace el proceso de cifrado simétrica. En la figura 25 se observa la operación del Record Protocol.

FIGURA 21. Operación del record protocol



FUENTE: Tomado de [58]

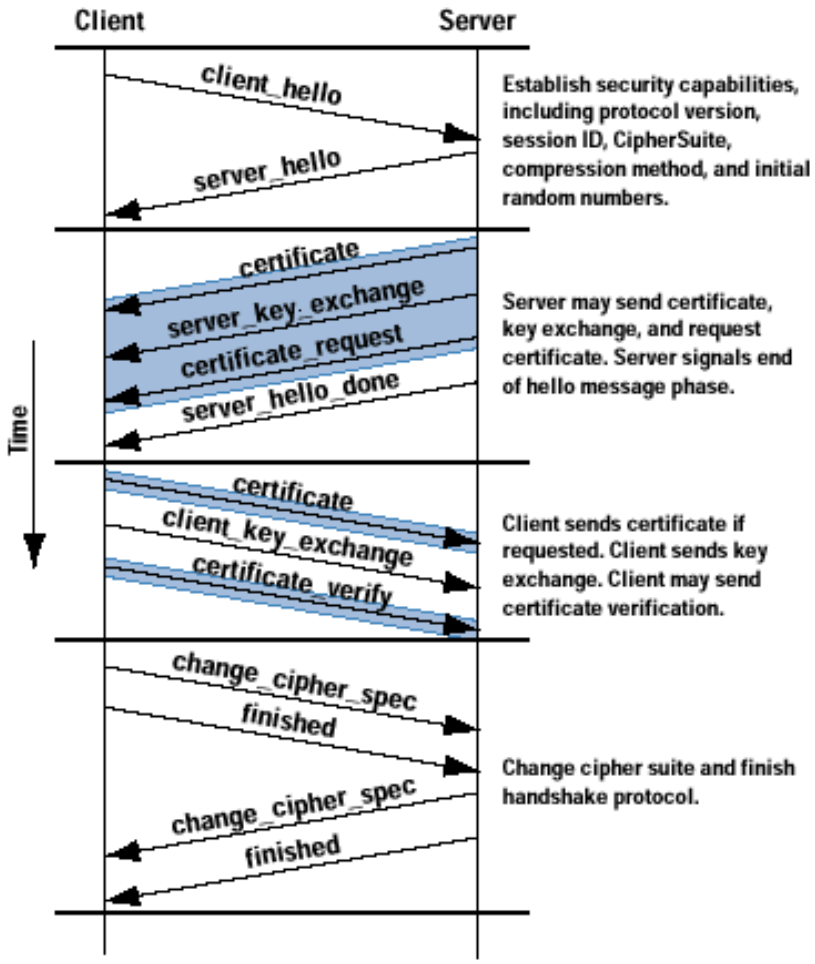
- **ChangeCipherSpec Protocol:** Es considerado como un protocolo simple, y podría pensarse que debería hacer parte o introducirse dentro del Handshake o Alert, pero de acuerdo al diseño de SSL no podría funcionar debido a su arquitectura. Puntualmente el aporte que este protocolo al tener un solo byte con valor de 1 hace es activar el estado pendiente para ser copiado en el estado actual.

- **Alert Protocol:** Este protocolo es usado como su nombre lo indica "alerta", tiene dos campos, uno es advertencia y otro de notificación fatal, el cual puede hacer que se termine de forma inmediata la conexión y mostrar la especificación del error encontrado.

- **Handshake Protocol:** Es un protocolo muy importante dentro de SSL, ya que se encarga de la negociación de parámetros para llevar a cabo la sesión entre el usuario y servidor, realiza el seguimiento durante la conexión hasta finalizar la misma.

El protocolo handshake encapsula en la SSL Record frecuentemente varios mensajes en uno solo, para realizar la etapa de negociación lo hace a través de un primer byte para definir el tipo de mensaje handshake a utilizar, luego 3 bytes que determina el tamaño del mensaje, y de allí el contenido de la información.

FIGURA 22. Pasos del handshake protocol



FUENTE: Tomado de [58]

En la figura 26 se observan los pasos que hace handshake para llevar a cabo la conexión entre el servidor y cliente.

Inicialmente el cliente hace la solicitud al servidor, para establecer la conexión haciendo referencia a los lineamientos que se tendrán en cuenta durante la comunicación como es: versión más reciente del protocolo SSL, identificador de sesión de longitud variable, método de compresión, lista de algoritmo criptográfico soportado por el cliente.

En la siguiente fase el servidor entra en escena haciendo el envío del certificado digital de tipo X.509 o infraestructura de clave pública si es necesario o el cliente lo solicita, con esto culmina la etapa del saludo entre cliente y servidor, para luego el cliente revisar que el certificado enviado por el servidor es auténtico, verificado este paso el cliente puede enviar mensajes de confirmación al servidor, también envía certificado de ser necesario, si éste no cuenta con el certificado envía un mensaje de alerta para reemplazarlo; luego el cliente hace intercambio de claves y certificado de verificación.

En la última fase, teniendo presente que la conexión se ha verificado y es segura, la comunicación entre cliente y servidor ha sido exitosa y se han configurado de manera correcta todos los parámetros a utilizar, se envía un mensaje de finalización y con esto termina le ejecución el protocolo handshake.

7.2.2. Certificados digitales SSL: Tal como se ha definido anteriormente el concepto del protocolo SSL, es brindar una conexión segura entre el usuario y servidor, para ello hace uso de los llamados certificados SSL, el cual se encarga de brindar la confianza requerida, cifrando la información que se envía, y dando un parte de autenticidad, donde la persona que accede al sitio web (correo electrónico, pagos electrónicos, entidades bancarias y/o financieras, entidades de gobierno, etc.), se sienta segura de que realmente accede al sitio que desea. Con esto se evita el phishing o suplantación de identidad.

Estos certificados digitales contienen información del dominio para el cual se expidió, propietario y su ubicación, fecha y tiempo de expiración del mismo. [59]

Para corroborar que el sitio al que accede es seguro, es importante revisar en la barra de direcciones, donde encuentra un símbolo en forma de candado, aunque esto no es suficiente, es necesario hacer el procedimiento de verificación certificado SSL de acuerdo al navegador que utilice.

Existe variedad de certificados digitales [60], los cuales se pueden clasificar según los dominios que se tengan, los cuales pueden ser únicos, comodín, y multidominio; y a su vez del nivel de validación que se requiera, dentro de los que se encuentran:

- Validación de dominios: es de bajo costo, cifrado elemental, incluye la autenticación del registro del nombre del dominio.

- Validación de organización: adicional a lo que permite el de dominios, valida nombre y dirección del propietario.

- Validación extendida: es el más completo en cuanto a seguridad, ya que incluye dentro de la verificación que la entidad esté legal y físicamente constituida.

7.3. TLS

Tal como se mencionó anteriormente, el protocolo SSL, fue diseñado originalmente por la compañía Netscape, pero debido a ciertas vulnerabilidades encontradas en operaciones de internet, la IETF (Internet Engineering Task Force), implementa el protocolo TLS (Transport Layer Security), en su primer versión TLS 1.0 en el año 1999

Las diferencias entre este protocolo en su versión TLS 1.0 y SSL 3.0 no son tan marcadas o dramáticas, pero suficientemente significativas, ya que busca que entre ambos no exista interoperabilidad, aunque se incorpora un mecanismo de retorno a la versión SSL 3.0. [61]

Después de la primera versión, en búsqueda de brindar una mayor seguridad en la conexión, y debido a que se han encontrado algunas deficiencias, el grupo de trabajo IETF, ha producido las versiones TLS1.1 en Abril 2006, TLS1.2, en 2008 y la versión TLS1.3 que se encuentra en fase de desarrollo.

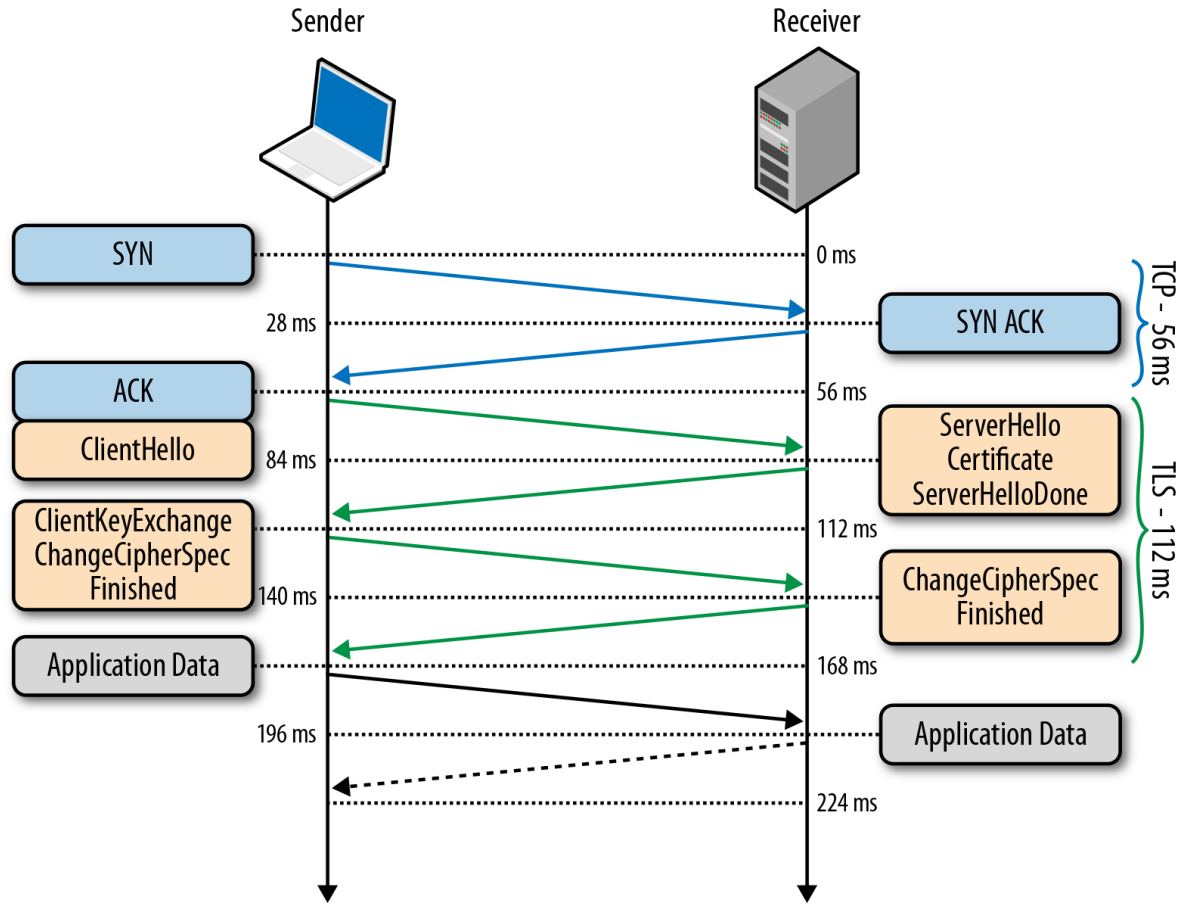
El protocolo TLS ha sido diseñado para proveer una conexión segura, que se basa en tres servicios esenciales [53]:

- Cifrar la información enviada de un ordenador a otro.
- Autenticación para verificar la validez del certificado del servidor, e identidad del cliente.
- Integridad para detectar si el mensaje ha sido manipulado o es falso.

Estos servicios se usan bajo el criterio de cada usuario, es decir, idealmente se deben usar los tres en su totalidad ya que si se decide no usar alguno, queda expuesto a altos riesgos de inseguridad.

7.3.1. TLS HANDSHAKE: Con el propósito de que el intercambio de información entre en el servidor y cliente, se de en los términos y condiciones más seguras, es necesario revisar ciertas condiciones, como lo es la versión TLS a usar, verificar certificados digitales ocasionalmente y seleccionar el ciphersuite (conjunto de cifrado).

FIGURA 23. Handshake protocol en TLS



FUENTE: Tomado de [53]

En la figura 27 se observa el procedimiento de este protocolo, el cual se describe a continuación:

TLS se ejecuta sobre la capa de transporte (TCP), por tanto lo primero es activar esta conexión, de allí, el cliente da a conocer al servidor el tipo de TLS que actualmente está usando, lista de ciphersuites compatibles, y las demás condiciones que sean necesarias para la conexión.

El servidor toma decisiones con base a lo que el cliente le ha propuesto, y envía su certificado digital al cliente. El cliente verifica el certificado y si ambos han negociado exitosamente, se avanza al paso siguiente, que es establecer la clave simétrica, el cual lo hace con RSA o Diffie-Hellman. De allí entonces el servidor procesa la información y planteamientos realizados por el cliente, verifica la autenticidad del mensaje y envía respuesta al cliente con un mensaje cifrado de finalizado.

El cliente haciendo uso de la clave simétrica, descifra el mensaje, realiza el proceso de autenticación, si todo está acorde a las condiciones acordadas, se establece la conexión para el envío de datos.

7.3.2. TLS RECORD PROTOCOL : El protocolo TLS Record se encarga de identificar los mensajes de Handshake y Alert, verificando su integridad y seguridad.

Cuando el protocolo recibe los datos los separa en grupos máximo de 214 bytes, o 16 KB por registro, eventualmente son comprimidos de ser necesario, aplica un código de autenticación MAC o HMAC, y luego los datos son cifrados de acuerdo al parámetro de negociación; a partir de allí el protocolo envía estos datos a la capa de transporte TCP, aquí se

aplica el procedimiento de manera inversa, descifra, verifica MAC y extrae los datos, para enviar a protocolos de niveles superiores

Se deben tener en cuenta unas consideraciones importantes al momento de hacer uso del protocolo de registro:

- Tamaño máximo del TLS record es de 16KB

- Cada registro contiene una cabecera de 5 bytes, un MAC (20 bytes para SSL3.0, TLS 1.0, TLS1.1 y hasta 32 bytes para la versión TLS 1.2)

- Para realizar el proceso de descifrar y verificar el registro, toda la TLS record debe estar disponible. [53]

7.3.3. ESTADO DE CONEXIÓN DEL TLS PROTOCOL: Un estado de conexión TLS es el modo de funcionamiento del protocolo TLS Record, especifica tres algoritmos como son: algoritmo de cifrado, algoritmo de compresión y algoritmo de MAC. [62]

Los parámetros de seguridad del protocolo TLS para una conexión en su estado de lectura y de escritura, son establecidos por los siguientes valores:

- **Fin de conexión:** Lo determinará el cliente o servidor en la conexión, al momento de culminar la sesión.

- **Algoritmo PRF:** Es usado con el fin de crear las claves del master secret.
- **Algoritmo de cifrado masivo:** Determina el tamaño de la clave, y del bloque de cifrado.

- **Algoritmo de MAC:** Autentica los mensajes, y especifica el tamaño del valor retornado por el algoritmo.

- **Algoritmo de compresión:** Comprime los datos.

- **Master secret:** Clave secreta de 48 byte compartida entre los dos actores involucrados en la conexión.

- **Valor aleatorio del cliente:** Valor de 32 bytes proporcionado por el cliente.

- **Valor aleatorio del servidor:** Valor de 32 bytes proporcionado por el servidor.

- **Estado de compresión:** Muestra el estado actual del algoritmo de compresión.

- **Estado de cifrado:** Muestra el estado en el que se encuentra el algoritmo de cifrado, ya sea para cifrar o descifrar.

- **Clave MAC:** Es la clave que se determinó para la conexión.

- **Número de secuencia:** Cada estado de conexión contiene un número de secuencia, que separa el estado de escritura del estado de lectura. Este debe reiniciarse con un valor de cero (0), cada vez que se hace un nuevo estado de conexión activo.

8. VULNERABILIDADES DE LOS PROTOCOLOS SSL Y TLS

A pesar de que el protocolo SSL/TLS ha tenido varias actualizaciones, en pro de brindar un alto nivel de seguridad en la conexión, no ha sido exento de ataques, que dejan entrever ciertas vulnerabilidades, de las cuales se han tomado acciones para mitigar el riesgo. A continuación se describen algunos de los ataques más conocidos.

TABLA 10. VULNERABILIDADES Y FORMAS DE MITIGACIÓN EN SSL Y TLS

ATAQUE	***VULNERABILIDAD***	***FORMA DE MITIGACIÓN***
BEAST	Es un ataque al protocolo SSL, el cual a través del acceso a la red local engañando al navegador, puede capturar tráfico del usuario, y con solo con tener un fragmento de texto sin formato y una carga cifrada, el atacante podrá conseguir una cookie de sesión a través de la puesta en marcha de herramientas de criptoanálisis. [63]	Para neutralizar este ataque se hace necesario migrar a las versiones 1.1 y 1.2 del protocolo TLS, así como también la implementación del método de cifrado RC4, el cual es inmune a este ataque. También se recomienda cerrar la sesión, una vez sea finalizada, ya que se evita la filtración de cookies de sesión por parte del atacante.

ATAQUE	*VULNERABILIDAD*	*FORMA DE MITIGACIÓN*
CRIME	Compression Ratio Info-leak Made Easy (CRIME) Este ataque reemplazó a BEAST, actuando de forma similar, como un agente que busca engañar al usuario remitiéndolo a sitios web falsos, y con base al nivel de acceso que tenga el atacante sobre la red del usuario podría agregar el código CRIME en una conexión HTTP. [64]	Como una forma de mitigación, además de las medidas aplicadas al ataque BEAST, se deshabilitó por completo la compresión SSL/TLS de forma predeterminada en las versiones de los navegadores.

ATAQUE	*VULNERABILIDAD*	*FORMA DE MITIGACIÓN*
LUCKY THIRTEEN	Utilizando la técnica llamada Padding, el ataque LUCKY logra acceder al cifrado principal de TLS, haciendo uso de la debilidad del modo de cifrado de bloques CBC en el protocolo, el atacante captura datos y cambia bloques de la información que se está enviando para detectar el tiempo que transcurre en dar respuesta el servidor, para así después de analizar estos tiempos de respuesta y repetir esta acción durante un tiempo determinado, pueda dar con el contenido cifrado real.	Como se ha notificado anteriormente la importancia de migrar a la versión 1.2 del protocolo TLS, no es ajeno para este ataque, así como también la utilización del cifrado RC4. Como medida correctiva implementaciones como CyaSSL, GnuTLS, NSS, OpenSSL, Opera entre otras actualizaron parches de seguridad.

ATAQUE	*VULNERABILIDAD*	*FORMA DE MITIGACIÓN*
HEARTBLEED	Heartbleed es una falla de seguridad en OpenSSL, que afecta a servidores de correo como Yahoo, el banco de imágenes Flickr, sitios como Google, Facebook entre otros, debido a una vulnerabilidad al hacer uso de la aplicación. El atacante actúa de forma incógnita logrando leer parte de la información de la memoria del servidor, como es la clave de cifrado que hayan acordado el cliente y servidor para la conexión sin que estos se den cuenta. A su vez si el cliente ingresa a un sitio web inseguro, de poca credibilidad, el atacante tendría acceso a cookies de sesión del usuario. [65]	Respecto a este ataque se tomaron medidas importantes debido al alto número de afectaciones, una de ellas fue actualizar a la última versión estable de OpenSSL, como lo es la 1.0.1g. A su vez se cambiaron las credenciales y certificados de los sitios web y como medida preventiva cambios de contraseñas a los usuarios.

ATAQUE	*VULNERABILIDAD*	*FORMA DE MITIGACIÓN*
POODLE	En el año 2014 después de un estudio realizado por ingenieros de Google se descubrió Poodle, vulnerabilidad cuyo objetivo se centra en degradar la conexión segura, para poder hacerse con los datos del usuario. [66]	En vista de que el ataque Poodle solo afecta a las versiones de SSL, la solución viable fue dejar de utilizar este protocolo y usar solo TLS1.0 o posteriores. [64]
	Poodle hace que cuando el cliente accede a una conexión segura, esta falle, y lo obliga a utilizar un protocolo de comunicación más antiguo, como es el caso SSL3.0, aprovechando la vulnerabilidad de este protocolo. [67]	Al encontrarse el protocolo SSL comprometido, los navegadores web, como es el caso de Mozilla, decidieron bloquear las conexiones con SSL 3.0. [68]
		Sin embargo se ha descubierto que, en el protocolo TLS, en su versión más reciente 1.2 puede existir una vulnerabilidad que permitiría a Poodle funcionar también, aunque no en todas las implementaciones de TLS. [69]

9. HTTP Strict Transport Security (HSTS) [70]

Si bien HTTPS se encuentra diseñado para ofrecer seguridad sobre ataques activos en la red y quienes en algún punto se encuentren interceptando paquetes, en muchas ocasiones los *User agent* (UA), que es básicamente una aplicación cliente HTTP que puede ser manipulada por el usuario, comprometen la seguridad de este con el fin de permitir las conexiones con sitios que no configuran HTTPS de forma correcta o generan certificados autofirmados. Este último caso se ve ejemplificado cuando se despliega al usuario un mensaje de advertencia pero se le da al usuario la opción de continuar su camino hacia el sitio mediante un simple clic. Esto es permitido por dos razones:

- Compatibilidad: Muchos sitios web se encuentran con certificados configurados de forma incorrecta.

- Intención Desconocida: Algunos sitios utilizan certificados autofirmados y tienen partes de su sitio corriendo sobre HTTP porque este mecanismo genera protección frente a ataques pasivos y creen que el costo de implementar completamente HTTPS sobrepasa el riesgo de ocurrencia y la relevancia de la información contenida en el servidor.

En él se propone un mecanismo de seguridad que entrega al UA una guía de la manera en que deben ser tratados los errores en la conexión, básicamente, todo error en la comunicación mediante HTTPS se considera como un ataque y no como un simple error en la configuración. Habilitar esta herramienta permite qué:

- Las conexiones que no se abran bajo HTTPS se re direccionen automáticamente a conexiones seguras lo que genera que no exista interacción sin el uso de TLS.

- Todo error en TLS, incluidos los certificados autofirmados y los errores de nombres comunes, terminan la sesión TLS.

- Intentos de incluir contenido inseguro dentro del sitio, genera fallas en la red

9.1. AMENAZAS

Los desarrolladores de este complemento de seguridad describen dos clases de amenazas, las dirigidas y las no dirigidas.

9.1.1. Amenazas Dirigidas [51]

9.1.1.1. Ataques de Red Pasivos

Son ataques que buscan obtener información que es transmitida en una red, con el fin de:

- Copiar información

- Analizar el tráfico (intercepción de identidad): lectura de cabeceras permitiendo la identificación de origen y destinatario.

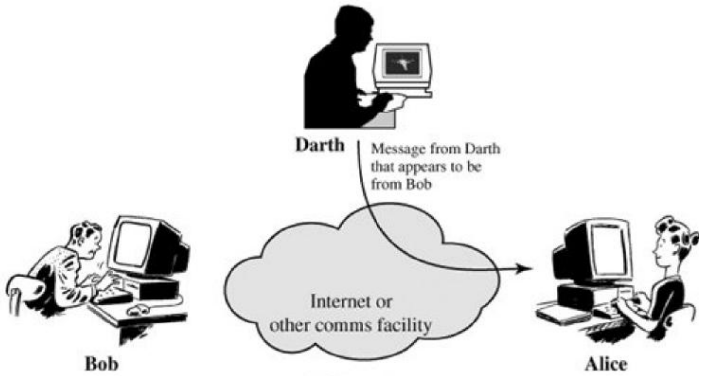
Un ejemplo de este tipo de ataques se encuentra cuando un usuario se encuentra navegando dentro de una red Wireless y un atacante cercano puede estar registrando el tráfico de las conexiones sin cifrado, como lo son las peticiones HTTP.

9.1.1.2. Ataques de Red Activos

Este tipo de ataque implica la modificación del flujo de datos transmitidos o la creación de un falso flujo de datos. A su vez, los ataques activos se dividen en:

- Suplantación de identidad (falsificación de identidad): fingir ser otra identidad o clonar una identidad. Como se observa en la Figura 28 en esta clase de ataques Darth (el atacante) se hace pasar por Bob para enviar un mensaje a Alice.

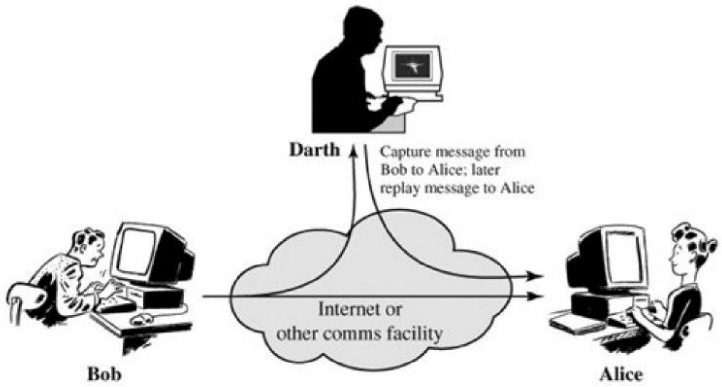
FIGURA 24. Suplantación



FUENTE: Tomado de [51].

- Repetición: Retransmitir mensajes con el fin de provocar fallas en el sistema. En la Figura 29 se puede observar la manera en que funciona este ataque; Bob envía un mensaje a Alice (victima) sin saber que Darth (atacante) captura los mensajes para retransmitirlos posteriormente.

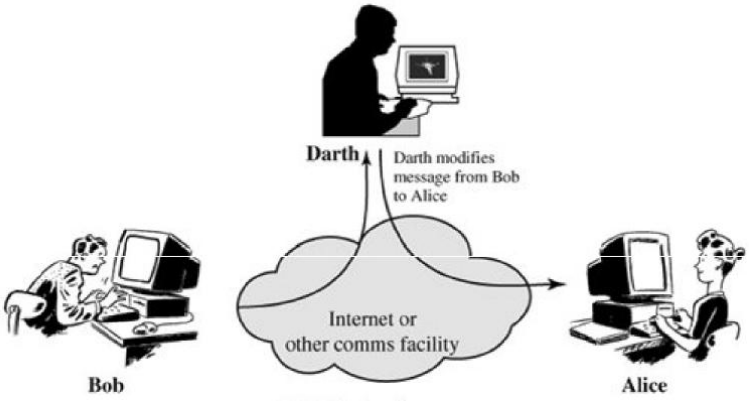
FIGURA 25. Repetición



FUENTE: Tomado de [51].

- Modificación de mensajes (alteración de mensajes): alterar, retrasar, reordenar la información. En el ataque descrito en la Figura 30 se observa la manera en que funciona este ataque; Bob envía un mensaje a Alice (victima) sin saber que Darth (atacante) captura los mensajes con el fin de modificarlos y enviarlos posteriormente para causar fallas en el sistema.

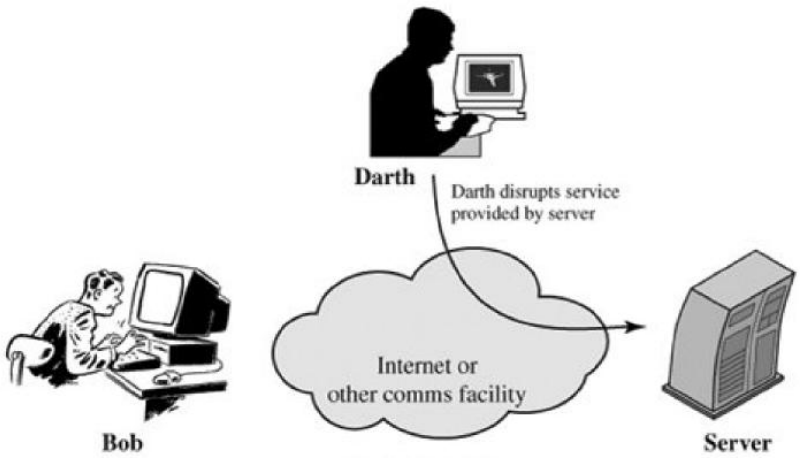
FIGURA 26. Modificación de paquetes



FUENTE: Tomado de [51].

- Interrupción de servicio :dejar fuera de servicio algún recurso del sistema (denegación de servicios) (ver figura 31)

FIGURA 27. DoS



FUENTE: Tomado de [51].

9.1.1.3. Honestos pero Imperfectos Desarrolladores Web

Un gran número de sitios web fueron elaborados por diversos desarrolladores, quienes ocasionalmente cometen errores y no son conscientes de las brechas de seguridad que se pueden generar. Incluso si se realizó una revisión detallada, un complemento inseguro embebido puede comprometer la seguridad de una plataforma de acceso mediante la inyección de un script.

9.1.2. Amenazas No Dirigidas

Phishing

Esta clase de ataques ocurre cuando un atacante solicita las credenciales de autenticación de un usuario bajo la fachada de una página confiable y conocida por el usuario. Esta clase de ataques resulta ser altamente efectiva ya que muchos usuarios no notan la diferencia entre un sitio falso y uno real. Si bien HSTS no se encuentra diseñado para contrarrestar resulta muy útil como complemento contra el phishing.

Malware y Vulnerabilidades en el Buscador

Debido a que HSTS es pensado como un mecanismo de seguridad del buscador, se parte de creencia que el usuario cuenta con la protección necesaria y con que los buscadores se encuentren actualizados para contrarrestar ataques que puedan comprometer la sesión, incluso si se está utilizando HSTS.

9.2. REQUERIMIENTOS DE SEGURIDAD

Buscando mitigar los riesgos mencionados anteriormente la publicación RFC 6797 ha dispuesto unos requerimientos base para el correcto despliegue de HSTS.

- Los sitios web deben poder declarar a los UAs que ellos deben ser contactados usando la política strict security.

- Los sitios web deben ser capaces de re direccionar de forma segura a aquellos UAs que inicien una conexión de forma insegura.

- Los UAs deben mantener los datos acerca de los sitios web que tienen habilitada la política strict security por el tiempo que declaren estos sitios.

Adicionalmente, los UAs necesitan mantener en el cache la información más reciente acerca de la política strict security.

- Los UAs deben reescribir todas las conexiones inseguras http para utilizar el esquema seguro https para aquellos sitios donde la política de seguridad se encuentre habilitada.

- Los administradores deben ser capaces de transferir la política strict security a los subdominios de aquellos dominios donde esta política se encuentra habilitada.

9.2.1. Defensas controladas por el usuario [71]

9.2.1.1. Fortalecer el uso de HTTPS

Muchos sitios funcionan con una mezcla de contenidos sobre HTTP y HTTPS, teniendo especial cuidado con usar este último en donde se requiera autenticación o se desarrollen los pagos. Esto protege los datos sensibles de usuario de cualquier atacante que se encuentre realizando un escaneo de los paquetes de red. Sin embargo, muchos de estos sitios manejan cookies no seguras, las cuales contienen el identificador de sesión del usuario, este tipo de cookies es enviado de forma plana y permite a quien se encuentre interceptando la red, obtener las credenciales del usuario.

Esta clase de vulnerabilidades se puede mitigar visitando el sitio web de forma web, el usuario escribe HTTPS en la url y revisa el estado de la conexión antes de continuar adelante.

9.2.1.2. Errores en los certificados

La configuración incorrecta de los servidores web puede causar un número de errores en los certificados HTTPS:

- Nombres comunes desiguales: HTTPS requiere que el servidor presente un certificado donde el nombre concuerde con el nombre de host del servidor.
- Autofirmados: Muchos dueños de sitios desean utilizar HTTPS pero no cuentan con la capacidad o simplemente no desean comprar el certificado a una autoridad certificadora. Esto, como se expresó anteriormente, genera protección ante ataques pasivos.
- Expirados: Los certificados son válidos por un periodo de tiempo. Muchos servidores web presentan certificados que aún no son válidos o cuyo periodo ya expiró.

Cuando los usuarios se topan con la advertencia generada por el navegador web, muchos no comprenden estas advertencias y automáticamente las ignoran. HSTS fuerza a que estos errores de certificados sean tratados como fatales.

9.2.2. Defensas controladas por el sitio [71]

9.2.2.1. Cookies Seguras

Un sitio que se preocupe por la seguridad puede marcar una cookie como segura lo que permite al navegador abstenerse de enviar está bajo una conexión insegura. Para utilizar este tipo de cookies, el sitio debe asegurar que todo el tráfico se maneja sobre HTTPS. Muchos sitios, incluidos aquellos que manejan defensas anti-phishing usan estas secure cookies para almacenar un segundo factor de autenticación. Si bien estas cookies evitan fácilmente los ataques pasivos, un atacante puede utilizar certificados inválidos para intentar robarlas si el usuario pasa las advertencias generadas por el navegador. Por lo anterior, HSTS termina las conexiones y no revela las cookies seguras del sitio cuando se presenta un certificado no valido.

9.2.2.2. Acceso Restringido Basado en Clave Pública

Con el objetivo de no revelar las cookies, se busca restringir el acceso a estas basándose en la llave pública del servidor. El objetivo es evitar un ataque tipo pharming que pueda comprometer las cookies generadas por el servidor real.

9.3. CONFIGURACIÓN DE HSTS [70]

Un host HTTP se declara a sí mismo un host HSTS mediante la emisión de la política HSTS a los UAs, que se encuentra representado y es transportado a través de la cabecera Strict Transport Security HTTP sobre una conexión segura como lo puede llegar a ser TLS. Una vez el UA reciba sin errores y procese la cabecera, este reconoce al host como un host HSTS.

La política HSTS busca guiar al UA a una comunicación con un host HSTS conocido sobre una comunicación segura y especifica el tiempo de duración de esta directriz.

La configuración de HSTS resulta sencilla dentro del servidor web y se realiza de la siguiente manera:

Strict-Transport-Security: max-age=NUMBER; includeSubDomains

El parámetro max-age hace referencia al tiempo (en segundos) en los que el UA del usuario deberá comunicarse mediante HTTPS con el servidor de destino. Para el ejemplo en cuestión se tiene un tiempo de 1 año y este conteo se reinicia cada vez que el usuario ingresa al sitio.

El parámetro includeSubDomains pretende forzar a que el UA del usuario valide que la comunicación sea realizada de forma segura en todos los subdominios del sitio.

Un parámetro adicional es la condición “preloaded” la cual permite a los administradores tener sitios preconfigurados con la política HSTS lo que fortalece la seguridad de los navegadores evitando ataques MITM

Dentro de las consideraciones de la configuración se debe tener en cuenta qué:

- El orden de los parámetros no es relevante.
- Todos los parámetros deben aparecer una única vez en la cabecera STS
- Los UAs deben ignorar cualquier cabecera STS que contenga directivas u otra información que no se encuentre dentro de la sintaxis definida en la RFC 6797

9.4. PROCESAMIENTO DEL MODELO EN SERVIDORES Y UAS

Cuando se responde a una solicitud HTTP que se encuentra transportada de forma segura, un host HSTS debería incluir en la respuesta del mensaje la cabecera STS que debe cumplir gramáticamente con lo especificado anteriormente.

Si un host HSTS recibe una solicitud HTTP sobre una conexión no segura, este debe enviar un mensaje de respuesta que contenga un código indicando una redirección personal y una cabecera que contenga la solicitud HTTP original alterada o una dirección generada de acuerdo a la política local sobre un esquema HTTPS.

Si una respuesta generada cumple con la estructura de la cabecera STS el UA puede tomar dos decisiones:

- Reconocer el host como un host HSTS si esto no se ha realizado con anterioridad.

- Actualizar el cache propio en caso de que el parámetro max-age y/o includeSubdomains de la cabecera STS contenga información diferente a la que se encuentra almacenada previamente.

  Se debe considerar que si el parámetro max-age tiene un valor de cero (0) el UA debe remover la información almacenada en cache.

De igual forma:

- Si una respuesta HTTP es recibida sobre conexión insegura, el UA debe ignorar cualquier presencia de cabecera STS que se encuentre.

9.4.1. Coincidencia en dominio para un host HSTS conocido

Un dominio, cualquiera que sea, puede coincidir con un dominio de host HSTS de dos formas: una congruencia total o una coincidencia de súper dominio. Si no llega a existir coincidencia entonces el dominio no corresponde a un host HSTS conocido.

- Congruencia: Se presenta si todos los caracteres concuerdan entre el dominio dado y el host HSTS conocido

- Súper Dominio: Se presenta cuando el host HSTS conocido concuerda con la parte derecha del dominio dado

Se debe destacar que si un UA recibe una respuesta HTTP de un host HSTS conocido sobre un canal seguro pero la respuesta no cuenta con la cabecera STS,

el servidor debe continuar tratando el host como conocido hasta que el parámetro max-age se alcance.

9.5. CONSIDERACIONES EN HSTS DURANTE SU IMPLEMENTACIÓN

Si bien no es normativa, las buenas prácticas siempre hacen parte del aseguramiento de la seguridad.

9.5.1. Tiempo de vencimiento de la política

Los diseñadores deben considerar el tiempo que va a estar activa la política, se puede trabajar con un tiempo constante en el futuro o con un tiempo deseado por parte del desarrollador. Si bien resultan ser muy parecidos se debe tener en cuenta que cada vez que un usuario le pida a su navegador ingresar a un host HSTS conocido, este actualiza los campos por lo que en algunas ocasiones resulta inoficioso considerar un tiempo extremadamente largo.

9.5.2. Usar HSTS con certificados auto-firmados

Si las siguientes 4 condiciones se cumplen:

- Un sitio web u organización se encuentran generando su propio certificado
- La autoridad que presenta el certificado típicamente no se encuentra dentro de los conocidos por los navegadores.
- La política HSTS se encuentra activa dentro de un host firmado por la organización
- El certificado no concuerda con un certificado TLS útil

Entonces las conexiones seguras fallaran por el diseño de HSTS. Esto con el fin de proteger frente a varios ataques activos.

Si una compañía desea ser su propia autoridad certificadora y utilizar certificados auto-firmados, en lo que concierne a HSTS, esto se puede desarrollar entregando su autoridad certificadora a los navegadores de los usuarios.

9.6. CONSIDERACIONES DE SEGURIDAD EN LA IMPLEMENTACIÓN

Adicional a la configuración de los parámetros se deben tener algunas consideraciones de seguridad importantes. Algunas de las consideraciones a tener en cuenta son:

- La necesidad de incluir el parámetro includeSubDomains, sin este un atacante podría adquirir las cookies de dominio, incluso si son seguras, ya que estas cookies son entregadas a cualquier subdominio de la aplicación web. Así pues, si un atacante agrega un registro DNS que apunte a un subdominio falso, incluso si se tiene HSTS configurado, las cookies serán enviadas a quien realizó la solicitud cuando el usuario ingrese a ese subdominio.

- HSTS podría ser usado como plataforma para ciertas formas de ataque DoS contra los sitios web. Esto sucede debido a que un atacante puede habilitar la política HSTS en el UA para una aplicación web determinada, permitiendo solo la comunicación segura e impidiendo, si el sitio no cuenta con la configuración adecuada, la conexión.

- Los ataques MITM es una de las situaciones que a las que se enfrentan aquellos que siguen un enlace a un host HSTS desconocido usando HTTP en vez de HTTPS lo que deja la conexión vulnerable a esta clase de ataques.

- Los ataques de red activos pueden corromper los protocolos de tiempo de red como lo es NTP (Network Time Protocol) haciendo menos efectivo a HSTS contra los clientes que confían en este protocolo o carecen de un reloj de tiempo real. Delorean es uno de los ejemplos de esta clase de vulnerabilidades, donde manipulando el sistema se puede forzar a HSTS a expirar y conectarse bajo HTTP.

9.7. USO ACTUAL DE HSTS

Tomando la seguridad como prioridad los desarrolladores de los navegadores han implementado HSTS en estos con el fin de mejorar la experiencia del usuario. El último de los navegadores en adoptar esta característica de seguridad fue Internet Explorer (IE), el cual lo soporta desde junio de 2015. En la tabla 11 se observan los navegadores y las versiones desde las cuales soportan HSTS. [72]

TABLA 11
NAVEGADORES QUE SOPORTAN HSTS

Característica	*Chrome*	*Firefox*	*IE*	*OPERA*	*SAFARI*
Soporte HSTS	4.0	4.0	11	12	7

FUENTE: Tomado de [72]

Pese a la mejora notable que representa HSTS como herramienta para reforzar la seguridad de la conexión, al 4 de noviembre del 2015 según estadísticas de Trustworthy Internet [73], de los sitios más populares solo el 5% de un total de 142,267 sitios tiene habilitada esta política.

10. CONCLUSIONES

- Dentro del análisis realizado a la penetración en el mercado de la computación en la nube, se encuentra en Colombia gran oportunidad de crecimiento, según [9] en tan solo 4 años se logró triplicar las ganancias de los servicios prestados en la nube por lo que el país se puede posicionar como uno de los referentes de América latina en los siguientes años, especialmente en cuando a arquitectura SaaS se refiere. Resulta de vital importancia para la ampliación del mercado la inclusión de las pymes, para lo cual se debe generar concientización en las personas de cuan beneficioso puede llegar a ser invertir en los servicios en nube.

- El aumento de confianza de los servicios en la nube permitirá mayores beneficios a los CSP. Para lograrlo se debe generar en estos últimos, la responsabilidad de mantener la información del cliente segura en todo su ciclo de vida. En esta parte juegan un papel muy importante los SLA que se firmen y la conciencia que tenga el cliente de cuán importante es conocer los límites del mismo para evitar pérdidas o filtraciones indeseables de información sensible.

- La legislación de los países debe ser fortalecida para evitar brechas que impacten negativamente en la confidencialidad de los usuarios sin que esto implique poner en riesgo las leyes de una nación. Resulta fundamental de esta manera, implementar políticas claras que se ajusten a la situación actual de crecimiento de las nuevas tecnologías y conocer sus impactos en las políticas públicas de una nación.

- Todo sistema de comunicación es vulnerable y puede ser objetivo de ataques, por esta razón establecer políticas de seguridad que permitan

reducir los riesgos potenciales es de suma importancia, también es cierto que día a día se producen nuevos software maliciosos que afectan a los usuarios y sus datos, por esta razón es necesario estar a la vanguardia de estos temas y establecer criterios pertinentes para brindar seguridad. El auge de la computación en nube representa una nueva área que debe ser comprendida para poder realizar una correcta configuración de las defensas que se van a desplegar.

- Adicional a lo anteriormente mencionado, la diversidad de amenazas a los que se enfrenta un sistema de computación en la nube hace necesaria la implementación de diversas soluciones que individualmente serían insuficientes para garantizar los datos de los usuarios y la integridad de los sistemas en una red convencional. Hypervisor es un elemento a tener en cuenta ya que sus brechas de seguridad pueden comprometer gran cantidad de datos e incluso el hardware mismo de la máquina.

- Las múltiples vulnerabilidades encontradas en SSL han llevado a declararlo inseguro y a que los navegadores planteen la necesidad de impedir las conexiones con sitios que puedan llegar a ser potencialmente inseguros lo que ha llevado a las entidades certificadoras y a los administradores de I.T. al uso del protocolo TLS en su versión más reciente o como mínimo en su versión 1.

- Dada la fácil implementación de HSTS esta política puede ser ampliamente desplegada por aquellos administradores que deseen fortalecer el sistema de seguridad que se les ofrece a los usuarios y el aseguramiento de que en todo momento las comunicaciones serán transmitidas bajo una conexión segura.

- Como complemento a las estrategias de seguridad HSTS permitiría fortalecer los esquemas de acceso y transmisión, mitigando los riesgos a los que se encuentran expuestos los usuarios, especialmente a lo referente con la confidencialidad e integridad de los datos. Esto último va de la mano con un fuerte protocolo de cifrado que no permita que se degraden los datos a un texto plano en caso de que estos estén siendo capturados durante el camino.

- Para el mejoramiento de la seguridad en la nube, HSTS puede llegar a desempeñar un papel fundamental en la autenticación de los servicios SaaS y PaaS ya que en la interacción entre la plataforma del CSP y el usuario se transmiten los inicios de sesión, que de ser detectados por un atacante, podrían dejar expuesta la información a la que tiene acceso esa cuenta en específico.

REFERENCIAS BIBLIOGRAFICAS

[1] «Growth in the Cloud [en línea] <http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/growth_cloud_infographic.pdf> [Citado 10 de enero 2015]».

[2] «ROUNTREE, Derrick; CASTRILLLO, Ilena. The Basics of Cloud Computing: Undestanding the Fundamentals of Cloud Computing in Theory and Practice, Syngress, 2013, p.155».

[3] «Mell, Peter.The NIST Definition of Cloud Compting. National Institute of Standards and Technology, 2011, p7.».

[4] «SOSINSKY, Barrie. Cloud Computing Bible, WILEY, 2011, p. 492.».

[5] «Gartner Predicts Infraestructure Services Will Accelerate Cloud Computing Growth [en línea] < http://www.forbes.com/sites/louiscolumbus/2013/02/19/gartner-predicts-infrastructure-services-will-accelerate-cloud-computing-growth/> [citado 22 de Octubre de 2».

[6] «Cisco Global Cloud Index: Forecast and Methodology, 2014-2019 [en línea] < http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf > [citado 26 de Octubre de 2015]».

[7] «CLOUD 401: NAVIGATING ADVANCED TOPICS IN CLOUD COMPUTING [en línea] < http://www.cdwnewsroom.com/wp-content/uploads/2015/02/CDW_Cloud-401_Report_FINAL_022315.pdf > [citado 15 de Octubre de 2015]».

[8] «Cloud computing in Latin America Current situation and policy proposals [en línea] < http://www.cepal.org/publicaciones/xml/7/52947/CloudcomputinginLA.pdf > [citado 12 de Octubre de 2015]».

[9] «Jordán, Valeria; Galerin, Hernán. Broadband in Latin America: Beyond Connectivity. Santiago de Chile: Libros CELAC, 2013, 340 p».

[10] «Cloud computing como generador de empleo [en línea] < news.sap.com/latinamerica/2013/03/26/cloud-computing-generador-empleo/ > [citado 10 de Noviembre de 2015]».

[11] «“Cómputo en la nube”: nuevo detonador para la competitividad de México [en línea] <http://imco.org.mx/wp-content/uploads/2012/6/computo_en_la_nube_detonador_de_competitividad_doc.pdf> [citado 22 de Octubre 2015]».

[12] «Analysis of the Colombian Cloud Computing Market [en línea] < http://es.slideshare.net/FrostandSullivan/frost-sullivan-analysis-of-the-colombian-cloud-computing-market > [citado 15 de Octubre de 2015]».

[13] «GRUSCHKA Nils, JENSEN, Meiko, “Attack Surfaces: A Taxonomy for Attacks on Cloud Services,” Cloud Computing, EN: IEEE International Conference, 2010, p. 276-279.».

[14] «XIAO, Zhifeng; XIAO, Yang. Security and Pricacy in Cloud Compting. IEEE COMMUNICATIONS SURVEYS & TUTORIALS. IEEE, 2013, p. 843-859».

[15] «FERNANDEZ, Diogo; Soares Liliana. Security issues in cloud environments: a survey. Int. J. Inf. Secur, 2014, p 113-170».

[16] «FERNANDEZ, Diogo; Soares Liliana. An analysis of security issues for cloud computing. Int. J. Inf. Secure, 2013, p 1-13».

[17] «MODI, Chirag; PATEL, Dhiren. A survey on security issues and solutions at different layers of Cloud computing. J. Supercomput, 2013, p 561-692».

[18] «VACCA, John R. Network and System Security. Syngress, 2013, p 86».

[19] «RAO, Celumadhava. Data Security Challenges and Its Solutions in Cloud Computing. EN: ICCC, 2015, p 204-209».

[20] «ZISSIZ, Dimitrios; LEKKAS, Dimitrios. Addressing cloud computing security issues. Future Generation Computer Systems, 2012, p 583-592».

[21] «Distributed denial- of-service attack (DDOS) definition [en línea] < http://searchsecurity.techtarget.com/definition/distributed-denial-of-service-attack> [citado 18 de Octubre de 2015]».

[22] «Distributed denial- of-service attack (DDOS) definition [en línea] < http://searchsecurity.techtarget.com/definition/distributed-denial-of-service-attack> [citado 18 de Octubre de 2015]».

[23] «Los diez riesgos más críticos en Aplicaciones Web [en línea] < https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf > [citado 25 de Septiembre de 2015]».

[24] «WU, Hanqian ; DING, Yi. Network Security for Virtual Machine in Cloud Computing, En: 5ta conferencia en ciencias computacionales y Convergencia de las tecnologías de la Información, 2010, p 18-21».

[25] «EGHTESADI, Arash, Preservation of security configuration in the cloud. En: Conferencia Internacional IEEE en Ingenieria de Nube; 2014, p 17-26».

[26] «SONG, Meng Hua. Analysis of risk for virtualization technology. Applied Mechanics and Materials, vol. 539, 2014, p 374-377.».

[27] «Eliminating the hypervisor attack surface for a more secure cloud. [en línea]

< https://www.cs.princeton.edu/~jrex/papers/ccs11.pdf >[citado 12 de Noviembre de 2015]».

[28] «Subashini, S. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, vol 34. 2011, p. 1-11».

[29] «DEYAN, Chen; ZHAO, Hong. Data Security and Privacy Protection Issues in Cloud Computing. En: Conferencia Internacional de Ciencias computacionales e Ingeniería Electrónica, 2012, p. 647-651».

[30] «REN, Kui. Toeard Secure and Dependable Storage Services in Cloud Computing. IEEE Transactions on Services Computing, VOL 5. 2012, p. 220-232».

[31] «SeDaSC: Secure Data Sharing in Clouds. [en línea] < http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7008450 > [citado 13 de Noviembre de 2015]».

[32] «GONZALEZ Nelson; MIERS, Charles. A quantitative analysis of current security concerns and solutions for cloud computing, Journal of Cloud Computing, 2012, p. 1-11.».

[33] «CHUNG, Chun-Jen. NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems. IEEE Transactions on Dependable and Secure Computing, vol 10, 2013, p. 198-211».

[34] «XING, Tianyi; HUANG, Dijiang. SnortFlow: A openflow-based intrusion prevention system in cloud environment, En: 2nd GENI Research and Educational Experiment Workshop, 2013, P. 89-92».

[35] «HE, Xiangjian; CHOMSIRI, Thawatchai. Improving cloud network security using the Tree-Rule firewall. Future Generation Computer Systems, vol 30,

2014, p. 116-126.».

[36] «Security Gidance for Critical Areas of Focus in Cloud Computing v3.0 < https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf > [citado 10 de Noviembre de 2015]».

[37] «LOMBARDI, Flavio; DI PIETRO, Roberto. Secure virtualization for cloud computing. Journal of Network and Computer Applications, vol 34, 2011, p. 1113-1122».

[38] «Taming Hosted Hypervisors with (Mostly) Deprivileged Execution [en línea] < http://www4.ncsu.edu/~cwu10/files/NDSS13_DEHYPE.pdf > [citado 8 de Noviembre de 2015]».

[39] «Isolating Commodity Hosted Hypervisors wit HyperLock [en línea] < http://www.cs.fsu.edu/~zwang/files/EuroSys12.pdf > [citado 8 de Noviembre de 2015]».

[40] «Eliminating the Hypervisor Attack Surface for a More Secure Cloud [en línea] < https://www.cs.princeton.edu/~jrex/papers/ccs11.pdf > [citado 8 de Noviembre de 2015]».

[41] «CloudVisor: Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization [en línea] < http://sigops.org/sosp/sosp11/current/2011-Cascais/printable/15-zhang.pdf > [citado 10 de Noviembre de 2015]».

[42] «XIA, Yubin; LIU Yutao. Architecture Support for Guest-Transparent VM Protection from Untrusted Hypervisor and Physical Attacks. En: IEEE 19th International Symposium on High Performance Computer Architecture, 2013, p. 246-257.».

[43] «Create & Manage Data [en línea] < http://www.data-archive.ac.uk/create-

manage/life-cycle > [citado 15 de Noviembre de 2015]».

[44] «SOOD, Sandeep. A combined approach to ensure data security in cloud computing. Journal of Network and Computer Applications, vol 35. 2012, p. 1831-18».

[45] «WANG, Cong; LOU, Wenjing. Toward Secure and Dependable Storage Services in Cloud Computing. IEEE Transactions on Services Computing, vol 5, 2012, p. 220-232».

[46] «WEI, Lifei; ZHU, Haojin. Security and privacy for storage and computation in cloud. Information Sciences, vol 258, 2014. P. 371-386».

[47] «SAH, Sushil; SHAKYA,Saroj. A security Management for Cloud Based Applications and Services with Diameter-AAA, En: IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques, 2014, p. 6-11».

[48] «An analysis of security issues for cloud computing [en línea] < http://www.jisajournal.com/content/4/1/5 > [citado 12 de Noviembre de 2015]».

[49] «O.D, Alowolodu; A.O, Adetunmbi. Ellitic Curve Cryptography for Securing Cloud Computing Applications, International Journal of Computer Applications, vol 66, 2013, p. 10-17».

[50] «Wan Zhiguo. HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing, vol. 7,2012, p. 743-754».

[51] STALLINGS, William. Cryptography and Network Security. Prentice Hall Press, 2005. 592 p.

[52] «ANDRESS, Jason. The Basics Of Information Security: Undestanding the

Fundamentals of InfoSec in Theory and Practice, Syngress, 2011, p. 63-80».

[53] «High Performance Browser Networking, Ilya Grigorik [en linea] < http://chimera.labs.oreilly.com/books/1230000000545/ch04.html > [citado el 11 de noviembre de 2015]».

[54] RAMGOVIND, Sumant. The Management of Security in Cloud Computing. Information Security for South Africa (ISSA), 2010, p 1-7

[55] «The Secure Sockets Layer (SSL) [en línea] < http://www.facweb.iitkgp.ernet.in/~sourav/SSL.pdf > [citado 2 noviembre de 2015]».

[56] «The Secure Sockets Layer (SSL) Protocol Version 3.0, Freier, Kocher, Agosto 2011 [en línea] < https://tools.ietf.org/html/rfc6101 > [citado 3 noviembre de 2015]».

[57] «SSL & TLS Essentials: Securing the Web Stephen A. Thomas [en línea] < http://imcs.dvfu.ru/lib.int/docs/Web/SSL%20&%20TLS%20Essentials.%20Securing%20the%20Web.pdf > [citado 25 octubre de 2015]».

[58] CISCO, SSL: Foundation for Web Security [en línea] < http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_1-1/ssl.html > [citado 26 de Octubre de 2015]

[59] «Marco teórico sobre los protocolos ssl/tls y su aplicación en el sistema de comercio electrónico, Carlos Fernando Estévez Marín, Yirish Arturo Martínez Ortega, 2013 [en línea] < http://tangara.uis.edu.co/biblioweb/tesis/2013/149814.pdf > [citado 1 novie».

[60] «Todo lo que debe saber sobre certificados SSL [en línea] < http://www.verisign.com/es_LA/domain-names/web-presence/website-optimization/ssl-certificates/index.xhtml > [citado 9 de noviembre de 2015]».

[61] «The Transport Layer Security (TLS) Protocol Version 1.0 Dierks & Allen RFC 2246, Enero 1999 [en linea] < https://www.ietf.org/rfc/rfc2246.txt > [citado el 10 de noviembre de 2015]».

[62] «The Transport Layer Security (TLS) Protocol Version 1.2 Dierks & E. Rescorla, RFC 5246, August 2008 [en linea] < https://tools.ietf.org/html/rfc5246 > [citado el 12 de noviembre de 2015]».

[63] «SSL ATTACKS [en línea] < http://resources.infosecinstitute.com/ssl-attacks/ > [citado 14 de Noviembre de 2015]».

[64] «CRIME SSL/TLS attack [en línea] < https://acunetix.com/vulnerabilities/web/crime-ssl-tls-attack > [citado 14 de Noviembre de 2015]».

[65] «5 cosas que debes saber sobre Heartbleed [en línea]<http://www.welivesecurity.com/la-es/2014/04/09/5-cosas-debes-saber-sobre-heartbleed/ > [citado 13 de noviembre de 2015]».

[66] «Poodle, la vulnerabilidad en SSL 3.0 y cómo te puede afectar [en línea] < http://www.welivesecurity.com/la-es/2014/10/15/poodle-vulnerabilidad-ssl-3/ > [citado 13 de noviembre de 2015]».

[67] «This POODLE Bites: Exploiting The SSL 3.0 Fallback [en línea] < https://www.openssl.org/~bodo/ssl-poodle.pdf > [citado 14 de Noviembre de 2015]».

[68] «mozilla support [en línea] <https://support.mozilla.org/es/kb/Firefox%20no%20se%20puede%20conectar%20de%20forma%20segura%20porque%20el%20protocolo%20SSL%20est%C3%A1%20desactivado > [citado 13 de noviembre de 2015]».

[69] «Poodle contraataca, con TLS en la mira [en línea] <

http://www.welivesecurity.com/la-es/2014/12/10/poodle-contraataca-tls/ > [citado 13 de noviembre de 2015]».

[70] «HTTP Strict Transport Security (HSTS) [en línea] < https://tools.ietf.org/html/rfc6797 > [citado 8 de Septiembre de 2015]».

[71] «ForceHTTPS_Protecting High-Security Web Sites from Network Attacks [en línea] < https://crypto.stanford.edu/forcehttps/forcehttps.pdf > [citado 15 de Septiembre de 2015]».

[72] «HTTP Strict Transport Security [en línea] < https://developer.mozilla.org/en-US/docs/Web/Security/HTTP_strict_transport_security > [citado 31 de Octubre de 2015]».

[73] «SSL PULSE [en línea] < https://www.trustworthyinternet.org/ssl-pulse/ > [citado 20 de Noviembre de 2015]».

BIBLIOGRAFIA

5 cosas que debes saber sobre Heartbleed [en línea]<http://www.welivesecurity.com/la-es/2014/04/09/5-cosas-debes-saber-sobre-heartbleed/ > [citado 13 de noviembre de 2015].

An analysis of security issues for cloud computing [en línea] < http://www.jisajournal.com/content/4/1/5 > [citado 12 de Noviembre de 2015].

Analysis of the Colombian Cloud Computing Market [en línea] < http://es.slideshare.net/FrostandSullivan/frost-sullivan-analysis-of-the-colombian-cloud-computing-market > [citado 15 de Octubre de 2015].

ANDRESS, Jason. The Basics Of Information Security: Undestanding the Fundamentals of InfoSec in Theory and Practice, Syngress, 2011, p. 63-80 .

CHUNG, Chun-Jen. NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems. IEEE Transactions on Dependable and Secure Computing, vol 10, 2013, p. 198-211.

Cisco Global Cloud Index: Forecast and Methodology, 2014-2019 [en línea] < http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf > [citado 26 de Octubre de 2015].

CISCO, SSL: Foundation for Web Security [en línea] < http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_1-1/ssl.html > [citado 26 de Octubre de 2015].

CLOUD 401: NAVIGATING ADVANCED TOPICS IN CLOUD COMPUTING [en línea] < http://www.cdwnewsroom.com/wp-content/uploads/2015/02/CDW_Cloud-401_Report_FINAL_022315.pdf > [citado 15 de Octubre de 2015].

Cloud computing como generador de empleo [en línea] < news.sap.com/latinamerica/2013/03/26/cloud-computing-generador-empleo/ > [citado 10 de Noviembre de 2015].

Cloud computing in Latin America Current situation and policy proposals [en línea] < http://www.cepal.org/publicaciones/xml/7/52947/CloudcomputinginLA.pdf > [citado 12 de Octubre de 2015].

CloudVisor: Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization [en línea] < http://sigops.org/sosp/sosp11/current/2011-Cascais/printable/15-zhang.pdf > [citado 10 de Noviembre de 2015].

Cómputo en la nube: nuevo detonador para la competitividad de México [en línea] <http://imco.org.mx/wp-

content/uploads/2012/6/computo_en_la_nube_detonador_de_competitividad_doc.pdf> [citado 22 de Octubre 2015].

Create & Manage Data [en línea] < http://www.data-archive.ac.uk/create-manage/life-cycle > [citado 15 de Noviembre de 2015].

CRIME SSL/TLS attack [en línea] < https://acunetix.com/vulnerabilities/web/crime-ssl-tls-attack > [citado 14 de Noviembre de 2015].

DEYAN, Chen; ZHAO, Hong. Data Security and Privacy Protection Issues in Cloud Computing. En: Conferencia Internacional de Ciencias computacionales e Ingeniería Electrónica, 2012, p. 647-651.

Distributed denial- of-service attack (DDOS) definition [en línea] < http://searchsecurity.techtarget.com/definition/distributed-denial-of-service-attack> [citado 18 de Octubre de 2015].

Distributed denial- of-service attack (DDOS) definition [en línea] < http://searchsecurity.techtarget.com/definition/distributed-denial-of-service-attack> [citado 18 de Octubre de 2015].

EGHTESADI, Arash, Preservation of security configuration in the cloud. En: Conferencia Internacional IEEE en Ingenieria de Nube; 2014, p 17-26.

Eliminating the Hypervisor Attack Surface for a More Secure Cloud [en línea] < https://www.cs.princeton.edu/~jrex/papers/ccs11.pdf > [citado 8 de Noviembre de 2015].

Eliminating the hypervisor attack surface for a more secure cloud. [en línea] < https://www.cs.princeton.edu/~jrex/papers/ccs11.pdf >[citado 12 de Noviembre de 2015].

FERNANDEZ, Diogo; Soares Liliana. An analysis of security issues for cloud computing. Int. J. Inf. Secure, 2013, p 1-13.

FERNANDEZ, Diogo; Soares Liliana. Security issues in cloud environments: a survey. Int. J. Inf. Secur, 2014, p 113-170.

ForceHTTPS_Protecting High-Security Web Sites from Network Attacks [en línea] < https://crypto.stanford.edu/forcehttps/forcehttps.pdf > [citado 15 de Septiembre de 2015].

Gartner Predicts Infraestructure Services Will Accelerate Cloud Computing Growth [en línea] < http://www.forbes.com/sites/louiscolumbus/2013/02/19/gartner-predicts-infrastructure-services-will-accelerate-cloud-computing-growth/> [citado 22 de Octubre de 2.

GONZALEZ Nelson; MIERS, Charles. A quantitative analysis of current security concerns and solutions for cloud computing, Journal of Cloud Computing, 2012, p. 1-11.

Growth in the Cloud [en línea] <http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/growth_cloud_infographic.pdf> [Citado 10 de enero 2015].

GRUSCHKA Nils, JENSEN, Meiko, “Attack Surfaces: A Taxonomy for Attacks on Cloud Services,” Cloud Computing, EN: IEEE International Conference, 2010, p. 276-279.

HE, Xiangjian; CHOMSIRI, Thawatchai. Improving cloud network security using the Tree-Rule firewall. Future Generation Computer Systems, vol 30, 2014, p. 116-126.

High Performance Browser Networking, Ilya Grigorik [en linea] < http://chimera.labs.oreilly.com/books/1230000000545/ch04.html > [citado el 11 de noviembre de 2015].

HTTP Strict Transport Security (HSTS) [en línea] < https://tools.ietf.org/html/rfc6797 > [citado 8 de Septiembre de 2015].

HTTP Strict Transport Security [en línea] < https://developer.mozilla.org/en-US/docs/Web/Security/HTTP_strict_transport_security > [citado 31 de Octubre de 2015].

Isolating Commodity Hosted Hypervisors wit HyperLock [en línea] < http://www.cs.fsu.edu/~zwang/files/EuroSys12.pdf > [citado 8 de Noviembre de 2015] .

Jordán, Valeria; Galerin, Hernán. Broadband in Latin America: Beyond Connectivity. Santiago de Chile: Libros CELAC, 2013, 340 p .

LOMBARDI, Flavio; DI PIETRO, Roberto. Secure virtualization for cloud computing. Journal of Network and Computer Applications, vol 34, 2011, p. 1113-1122.

Los diez riesgos más críticos en Aplicaciones Web [en línea] < https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf > [citado 25 de Septiembre de 2015].

Marco teórico sobre los protocolos ssl/tls y su aplicación en el sistema de comercio electrónico, Carlos Fernando Estévez Marín, Yirish Arturo Martínez Ortega, 2013 [en línea] < http://tangara.uis.edu.co/biblioweb/tesis/2013/149814.pdf > [citado 1 novie.

Mell, Peter.The NIST Definition of Cloud Compting. National Institute of Standards and Technology, 2011, p7.

MODI, Chirag; PATEL, Dhiren. A survey on security issues and solutions at different layers of Cloud computing. J. Supercomput, 2013, p 561-692.

mozilla support [en línea] <https://support.mozilla.org/es/kb/Firefox%20no%20se%20puede%20conectar%20de%20forma%20segura%20porque%20el%20protocolo%20SSL%20est%C3%A1%20desactivado > [citado 13 de noviembre de 2015].

O.D, Alowolodu; A.O, Adetunmbi. Ellitic Curve Cryptography for Securing Cloud Computing Applications, International Journal of Computer Applications, vol 66, 2013, p. 10-17.

Poodle contraataca, con TLS en la mira [en línea] < http://www.welivesecurity.com/la-es/2014/12/10/poodle-contraataca-tls/ > [citado 13 de noviembre de 2015].

Poodle, la vulnerabilidad en SSL 3.0 y cómo te puede afectar [en línea] < http://www.welivesecurity.com/la-es/2014/10/15/poodle-vulnerabilidad-ssl-3/ > [citado 13 de noviembre de 2015].

RAMGOVIND, Sumant. The Management of Security in Cloud Computing. Information Security for South Africa (ISSA), 2010, p 1-7.

RAO, Celumadhava. Data Security Challenges and Its Solutions in Cloud Computing. EN: ICCC, 2015, p 204-209 .

REN, Kui. Toeard Secure and Dependable Storage Services in Cloud Computing. IEEE Transactions on Services Computing, VOL 5. 2012, p. 220-232.

ROUNTREE, Derrick; CASTRILLLO, Ilena. The Basics of Cloud Computing: Undestanding the Fundamentals of Cloud Computing in Theory and Practice, Syngress, 2013, p.155.

SAH, Sushil; SHAKYA,Saroj. A security Management for Cloud Based Applications and Services with Diameter-AAA, En: IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques, 2014, p. 6-11.

Security Gidance for Critical Areas of Focus in Cloud Computing v3.0 < https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf > [citado 10 de Noviembre de 2015].

SeDaSC: Secure Data Sharing in Clouds. [en línea] < http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7008450 > [citado 13 de Noviembre de 2015].

SONG, Meng Hua. Analysis of risk for virtualization technology. Applied Mechanics and Materials, vol. 539, 2014, p 374-377.

SOOD, Sandeep. A combined approach to ensure data security in cloud computing. Journal of Network and Computer Applications, vol 35. 2012, p. 1831-18.

SOSINSKY, Barrie. Cloud Computing Bible, WILEY, 2011, p. 492.

SSL & TLS Essentials: Securing the Web Stephen A. Thomas [en línea] < http://imcs.dvfu.ru/lib.int/docs/Web/SSL%20&%20TLS%20Essentials.%20Securing%20the%20Web.pdf > [citado 25 octubre de 2015].

SSL ATTACKS [en línea] < http://resources.infosecinstitute.com/ssl-attacks/ > [citado 14 de Noviembre de 2015].

SSL PULSE [en línea] < https://www.trustworthyinternet.org/ssl-pulse/ > [citado 20 de Noviembre de 2015].

STALLINGS, W. C. (s.f.).

Subashini, S. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, vol 34. 2011, p. 1-11.

Taming Hosted Hypervisors with (Mostly) Deprivileged Execution [en línea] < http://www4.ncsu.edu/~cwu10/files/NDSS13_DEHYPE.pdf > [citado 8 de Noviembre de 2015].

The Secure Sockets Layer (SSL) [en línea] < http://www.facweb.iitkgp.ernet.in/~sourav/SSL.pdf > [citado 2 noviembre de 2015].

The Secure Sockets Layer (SSL) Protocol Version 3.0, Freier, Kocher, Agosto 2011 [en línea] < https://tools.ietf.org/html/rfc6101 > [citado 3 noviembre de 2015].

The Transport Layer Security (TLS) Protocol Version 1.0 Dierks & Allen RFC 2246, Enero 1999 [en linea] < https://www.ietf.org/rfc/rfc2246.txt > [citado el 10 de noviembre de 2015].

The Transport Layer Security (TLS) Protocol Version 1.2 Dierks & E. Rescorla, RFC 5246, August 2008 [en linea] < https://tools.ietf.org/html/rfc5246 > [citado el 12 de noviembre de 2015].

This POODLE Bites: Exploiting The SSL 3.0 Fallback [en línea] < https://www.openssl.org/~bodo/ssl-poodle.pdf > [citado 14 de Noviembre de 2015].

Todo lo que debe saber sobre certificados SSL [en línea] < http://www.verisign.com/es_LA/domain-names/web-presence/website-optimization/ssl-certificates/index.xhtml > [citado 9 de noviembre de 2015].

VACCA, John R. Network and System Security. Syngress, 2013, p 86.

Wan Zhiguo. HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing, vol. 7,2012, p. 743-754.

WANG, Cong; LOU, Wenjing. Toward Secure and Dependable Storage Services in Cloud Computing. IEEE Transactions on Services Computing, vol 5, 2012, p. 220-232.

WEI, Lifei; ZHU, Haojin. Security and privacy for storage and computation in cloud. Information Sciences, vol 258, 2014. P. 371-386.

WU, Hanqian ; DING, Yi. Network Security for Virtual Machine in Cloud Computing, En: 5ta conferencia en ciencias computacionales y Convergencia de las tecnologías de la Información, 2010, p 18-21.

XIA, Yubin; LIU Yutao. Architecture Support for Guest-Transparent VM Protection from Untrusted Hypervisor and Physical Attacks. En: IEEE 19th International Symposium on High Performance Computer Architecture, 2013, p. 246-257.

XIAO, Zhifeng; XIAO, Yang. Security and Pricacy in Cloud Compting. IEEE COMMUNICATIONS SURVEYS & TUTORIALS. IEEE, 2013, p. 843-859 .

XING, Tianyi; HUANG, Dijiang. SnortFlow: A openflow-based intrusion prevention system in cloud environment, En: 2nd GENI Research and Educational Experiment Workshop, 2013, P. 89-92.

ZISSIZ, Dimitrios; LEKKAS, Dimitrios. Addressing cloud computing security issues. Future Generation Computer Systems, 2012, p 583-592.