ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

"GEEKS"

DECC-*Report*, Tendencias en Computación

REVISTA TÉCNICA DEL DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN.

ISSN 1390-5236

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA.

VOL. 1, No. 6, 2015

EDITORES
Ing. Walter M. Fuertes D., PhD.
Profesor Titular-Principal del Departamento de Ciencias de la Computación
Sangolquí Ecuador
e-mail: wmfuertes@espe.edu.ec

Ing. Fidel Castro De la Cruz, MSc
Director del Departamento de Ciencias de la Computación. Sangolquí Ecuador
E-mail: flcastro@espe.edu.ec

Portada:	**Ing. Germán Ñacato**
Diagramación:	Varios autores
Impresión:	**Editorial Politécnica de la ESPE**
Año de Edición:	**2015**

Comité Editorial

Nombre	Institución	País
Andrés Epifanía Huertas, PhD Cand.	Universidad Católica Los Ángeles (Chimbote)	Perú
Armando Cabrera, PhD Cand	Universidad Técnica Particular de Loja	Ecuador
Diego Miguel Marcillo, PhD	Universidad de las Fuerzas Armadas ESPE	Ecuador
Diego Pinto, MSc	Universidad de las Fuerzas Armadas ESPE	Ecuador
Daniel Riofrío, PhD	Universidad Politécnica de Madrid	España
Edgar Torres, MSc.	Escuela Politécnica Nacional	Ecuador
Edison G. Espinosa, PhD	Universidad de las Fuerzas Armadas ESPE	Ecuador
Efraín R. Fonseca, PhD	Universidad de las Fuerzas Armadas ESPE	Ecuador
Esteban Gómez, PhD Cand	Universidad Tecnológica Equinoccial UTE	Ecuador
Geovanny Ninahualpa, PhD Cand	Universidad de las Fuerzas Armadas ESPE	Ecuador
Giovanny Ranura, PhD Cand	Universidad de las Fuerzas Armadas ESPE	Ecuador
Jenny Ruiz, MSc	Universidad de las Fuerzas Armadas ESPE	Ecuador
Jenny Torres, PhD	Escuela Politécnica Nacional	Ecuador
Jorge Ramió, PhD	Universidad Politécnica de Madrid	España
John W. Castro, PhD	Universidad Autónoma de Madrid	España
Jonathan Barriga, PhD Cand.	Escuela Politécnica Nacional	Ecuador
Jorge Enrique Otalora, PhD	Universidad Pedagógica Tecnológica de Colombia	Colombia
José Luis García Dorado, PhD	Universidad Autónoma de Madrid	España
Luis Enrique Sánchez Crespo, PhD	SICAMAN	España
Luis Terán, PhD	Universidad de Friburgo	Suiza
Manuel Sánchez Rubio, PhD	Universidad Internacional de la Rioja	España
Marco Molina, PhD Cand.	Universidad Politécnica de Madrid	Ecuador
Mauricio Espinoza, PhD	Universidad de Cuenca	Ecuador
Roberto Andrade, PhD Cand	Escuela Politécnica Nacional	Ecuador
Walter Fuertes, PhD	Universidad de las Fuerzas Armadas ESPE	Ecuador

Presentación

El Departamento de Ciencias de la Computación (DECC) de la Universidad de las Fuerzas Armadas "ESPE", pone en consideración de la comunidad el Volumen 1, No. 6, 2015, de la revista técnica "GEEKS" - **DECC *Report*, Tendencias en Computación**".

En este volumen se recogen los resultados de las investigaciones y tesis de pregrado y postgrado tanto de la carrera de Ingeniería de Sistemas como de los programas de posgrado afines como son Gerencia de Sistemas, Auditoría de Sistemas Tecnológicos, Redes de datos y Conectividad, en los que participan docentes, investigadores y estudiantes locales, nacionales e internacionales. La convocatoria abierta se realizó el 12 de julio de 2015. Se receptaron 63 artículos técnicos, de los cuales se seleccionaron 22 para recibir arbitraje ciego. Dada las exigencias que se demanda a nivel de investigadores con trayectoria, apenas cinco manuscritos fueron aceptados definitivamente por parte del Comité Editorial de la revista. El proceso de revisión fue de arbitraje ciego. El Comité Editorial fue conformado por el 80% de réferis con el grado de PhD o candidatos, tanto ecuatorianos, así como seis españoles, un peruano, un colombiano y un profesor ecuatoriano que trabaja en una universidad Suiza.

Luego del proceso de revisión y arbitraje, fueron reenviados a los autores para su mejoramiento. Estas publicaciones reportan trabajos técnicos-científicos en áreas relacionadas con Ingeniería de Software, Tecnologías de Virtualización, Redes de Computación, y Nube Computacional, específicamente la mayoría de trabajos son trabajos orientados a la Ciberseguridad.

Aprovechamos esta ocasión para agradecer todo el apoyo técnico brindado por varios Coordinadores del DECC, algunos docentes, estudiantes y de manera especial al Comité Editorial de la revista, en la materialización del presente volumen.

Por lo expuesto, "GEEKS" **DECC *Report*, Tendencias en Computación,** constituye un medio de difusión local y nacional, cuya información esperamos resultará de interés para docentes, investigadores y estudiantes, invitándoles a aprovechar su contenido y a continuar enviando sus contribuciones en las siguientes ediciones.

En este mismo contexto, en nuestra calidad de editores de la revista, hacemos propicia la ocasión para invitar a ustedes, docentes, investigadores, estudiantes de grado y pregrado de las diferentes comunidades de investigación, a que participen activamente en los Grupos de Investigación de nuestro Departamento: **Grupo de Investigación de Sistemas Distribuidos, Ciberseguridad y Contenido, y el Grupo de Investigación en Ingeniería de Software Empírica**, con el fin de fortalecer aún más la investigación en estos campos, lo que coadyuvará con el incremento de la producción científica de nuestra universidad y de nuestro querido Ecuador.

Los Editores

Sumario

Volumen 6, No. 1, 2015

Mitigación de Ataques DDoS en Base de Datos Mediante un Balanceador de Carga

Tupac Amaru Cartuche, Henry Lopez G., Oscar Paredes C.
Departamento de Ciencias de la Computación
Universidad de las Fuerzas Armadas, Sangolquí, Ecuador
mtcartuche@espe.edu.ec, hflopez@espe.edu.ec, ogparedes@espe.edu.ec

Resumen—**La incidencia de los ataques de denegación de servicio (DDoS) es alta actualmente en Internet. Varios tipos de ataques comprometen el correcto funcionamiento de los distintos componentes de los sistemas. Uno de estos son las bases de datos, importantes repositorios de información empresarial de alta importancia debido a su naturaleza en cuanto al almacenamiento de datos. Afectar el funcionamiento de un gestor de base de datos puede indisponer en gran forma el funcionamiento de una organización. El presente trabajo examina de forma experimental el uso de un middleware para mitigar este tipo de ataques. Se construye una infraestructura de pruebas en la cual se simula altas tasas de llamadas a un aplicativo Web con el objeto de colapsar una base de datos Postgresql. El alto número de llamadas hace que el gestor de base de datos colapse el servidor. A continuación se realizan pruebas con el middleware PGBouncer que funciona como balanceador de carga y se mide la efectividad de interponerlo entre el aplicativo y el motor de base de datos. Se realizan pruebas con la ayuda de Jmeter simulando varios usuarios concurrentes y validando los efectos en el porcentaje de uso de recursos en el servidor donde se encuentra instalado Postgresql.**

Palabras Clave— Postgresqsl; balanceo de carga; PGBouncer; Ataque de denegación de servicios.

I. INTRODUCCIÓN

Hoy en día empresas y personas dependen en gran medida de la tecnología para realizar actividades que van desde poner en marcha un plan de negocios hasta sencillas actividades para el hogar. Esta dependencia se debe a la facilidad de acceso a la información y a la necesidad cada vez más creciente de contar con servicios tecnológicos que faciliten la ejecución de tareas. Es por esta razón que la tecnología, conformada por hardware y software, debe ser diseñada para soportar acceso concurrente y a gran velocidad hacia los repositorios de información, sin que esta modalidad de uso represente un problema de rendimiento más aún si esta tecnología es utilizada para brindar un servicio a nivel mundial que puede ser consumida por pequeñas y grandes empresas.

Es importante mencionar algunos estudios para entender de mejor manera cómo evoluciona la tecnología. Según estudios realizados por el Instituto Nacional de Estadísticas y Censos del Ecuador (INEC), en el 2012 un censo determinó que el 26.4% de hogares posee computadores y el 13.9% tienen computadores portátiles. Para el año 2013 se destaca que las personas que tienen computador portátil corresponden al 4.2% y solo un 0.9% para aquellas que poseen un computador de escritorio [1].

Por otro lado, si se analiza el uso de Internet se observa que las personas entre 16 y 24 años representan el 64.9% seguido de las personas entre 25 y 34 con el 46.2%, lo que evidencia claramente quienes son los usuarios de Internet según los datos de las últimas estadísticas del INEC para el año 2012. Para el año 2013 el crecimiento es de 3.70% y se pronostica que para el año 2014 será del 5.30%. Como dato adicional importante se debe destacar que el 50.9% de la población usa Internet en el hogar y el 26.6% accede en centros públicos [2].

Conforme avanza la innovación en tecnología, también se observa el uso cada vez más frecuente de teléfonos celulares inteligentes para acceder a diferentes servicios e información desde cualquier parte de mundo. Es así que el 4.70% adquiere líneas celulares, mientras que líneas fijas decrece en un 2.80% según el INEC 2013[1].

Del análisis presentado se observa que cada vez es más fácil acceder a Internet y que las empresas y personas son vulnerables en este espacio, su información privada está expuesta si no se toman las precauciones debidas. Las bases de datos que son los mecanismos implementados para resguardar la información de empresas, clientes y personas, son blanco de ataques especializados con el objetivo de obtener o manipular información. Muchas veces el objetivo del atacante es dejar sin servicio una página o sistema web colapsando su base de datos mediante DDoS.

El presente estudio pretende demostrar cómo las herramientas que controlan o balancean las conexiones a la base de datos ayudan a mitigar uno de los ataques más comunes conocidos, el ataque DDoS. Para las pruebas se utilizó una base de datos PostgreSQL y el balanceador de carga PGBouncer para administrar las conexiones a la base de datos. Adicionalmente se utilizó el software JMeter para ejecutar pruebas de estrés y verificar peticiones al servidor, diagnosticando el comportamiento de conexiones entrantes y aplicando acciones oportunas y proactivas según amerite.

En la sección II se presentan conceptos relacionados con base de datos PostgreSQL, PGBouncer, JMeter y ataques DDoS. En la sección III se presenta el diseño y la implementación del experimento, pasando a analizar y evaluar los resultados obtenidos en la sección IV. En la sección V se mencionan trabajos relacionados y para culminar se presentan la conclusiones y trabajos futuros en la sección VI.

II. MARCO TEÓRICO

A. *Base de datos PostgreSQL*

PostgreSQL es un ORDBMS (Object Relational Database Management System), un sistema de gestión de base de datos relacionales que se encuentra disponible en el mercado desde hace tres décadas y que es distribuido bajo licencia BSD, es decir, una licencia de software libre.

Este sistema de gestión se ha desarrollado a la par con otros gestores de tipo comercial y propietario, de manera que en la actualidad tanto sistemas propietarios como libres comparten gran popularidad a nivel mundial, adjudicándose el 55% las soluciones propietarias y el 45% aquellas que son de libre distribución, según el estudio presentado por la empresa DB-Engines en [3].

Sin embargo, a pesar de que las soluciones propietarias mantienen una hegemonía ligeramente superior sobre las soluciones libres, de mantenerse la tendencia mundial de los últimos años se podría asegurar que en los próximos años las soluciones libres para gestionar bases de datos serán más utilizadas por muchas empresas a nivel mundial (Figura 1).

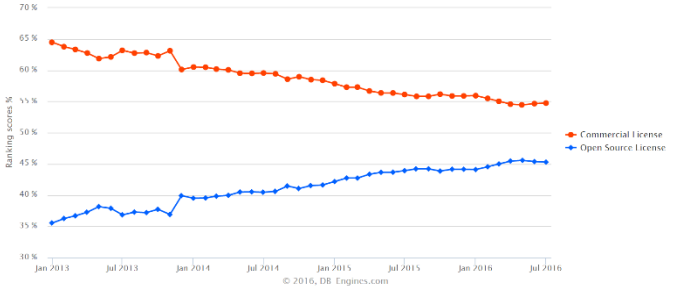


Figura 1: Tendencia de popularidad a Julio 2016. Publicado en [3].

Considerando el panorama presentado en los párrafos anteriores, no es descabellado pensar que los ciber delincuentes orienten sus ataques hacia aquellas empresas que utilicen gestores libres, más aún cuando son ampliamente utilizadas tal como es el caso de PostgreSQL que ocupa el primer lugar en soluciones libres para base de datos relacionales como lo demuestra DB-Engines.

Con esta perspectiva, es necesario buscar mecanismos de protección contra ataques que impidan el normal funcionamiento de los servidores de base de datos, herramientas que ayuden a mejorar y garantizar la seguridad de la información.

B. *Balanceador de carga: PGBouncer*

Un balanceador de carga es un mecanismo de hardware y software utilizado para gestionar una gran cantidad de solicitudes de usuario y dividir de manera equitativa el trabajo entre los recursos disponibles de manera que se eviten cuellos de botella [5].

Para el caso de PostgreSQL, existe un software que administra la cantidad de conexiones entrantes hacia el servidor de base de datos, consiguiendo disminuir y controlar las solicitudes de acceso, reduciendo de esta manera el tiempo de procesamiento y el tiempo de uso de los recursos del servidor [4], gestionando conexiones hacia una o más bases de datos.

PGBouncer soporta tres tipos de pool de conexiones: 1) Pool de sesiones, 2) Pool de transacciones, y; 3) Pool de Sentencias. En el primer caso, se asigna un conexión para cada cliente para todo el tiempo que dure la conexión; en el segundo caso se asigna una conexión por cada transacción que se ejecute, y; en el último caso se restringe la conexión a nivel de sentencias.

Dadas las tres posibilidades o modos de funcionamiento descritos en el párrafo anterior, es posible utilizar el balanceador instalándolo entre el servidor de aplicaciones y el servidor de base de datos para controlar el número de conexiones, autorizando o negando aquellas que puedan ser generadas como parte de un ataque DDoS.

C. *Ataque de denegación de servicio (DDos)*

Un ataque de denegación de servicio distribuido (DDoS) es un procedimiento que atenta contra los recursos de red de un servidor principalmente web y de base de datos, dejando no disponible el servicio para los usuarios de forma temporal o indefinidamente.

Los ataques de denegación de servicios se han desarrollado y ejecutado sobre todas las capas del modelo OSI, empezando por ataques de red hasta ataques a sesiones y en la capa de aplicación hoy en día. Este escalamiento en las diferentes capas del modelo OSI hace que se incremente la capacidad de detectar ataques [5]. Si a lo anterior se complementa con aplicaciones mal diseñadas, que no cumplen buenas prácticas internacionales como OWASP, representan una gran vulnerabilidad que puede ser utilizada por los ciber atacantes.

D. *Generador de carga JMeter*

Software de generación de carga que permite evaluar el desempeño de un servicio o sistema cuando es expuesto a condiciones de gran demanda, principalmente aquellas disponibles en sistemas web.

Inicialmente JMeter fue diseñado solamente para ejecutar pruebas de estrés en aplicaciones web, sin embargo en la actualidad es capaz de realizar desde una petición sencilla hasta una secuencia de peticiones al servidor que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción [6].

III. DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN

A. *Diseño e implementación de la topología de prueba*

Tomando en cuenta que varios de los tipos de ataques DDoS tienen como efecto el consumo desmesurado de recursos de un sistema, se diseñó una topología basada en aplicaciones web para realizar la experimentación.

Se utilizó la arquitectura clásica de las aplicaciones basadas en Internet, configurando un servidor web en el cual se ejecutó un aplicativo constituido de una interface de usuario con varios controles. A continuación se interpuso una capa de lógica de negocio, encargada de transportar los datos desde la interface de usuario hacia la capa de acceso a datos, la que se encargó de manejar la conexión con la base de datos.

En otro servidor se instaló un gestor de base de datos. El motor de base de datos elegido para la experimentación fue

Postgresql, software Open Source que una alternativa de gestor de base de datos altamente recomendable para pequeñas y medianas empresas u organizaciones. Se destaca por su alta capacidad de trabajo y su facilidad de mantenimiento.

Los dos servidores fueron desplegados en un entorno virtualizado ejecutándose en un computador. Los elementos del entorno de prueba se resumen en la siguiente tabla (Tabla I):

TABLA I. EQUIPO USADO EN EL EXPERIMENTO

Cantidad	**Componente**	
	Nombre	***Características***
1	PC 01	Computador Host Intel CORE i5, procesador 2.5Ghz, 2 núcleos, 4 procesadores lógicos, 8 Gb RAM, Windows 10 64 bits.
2	VPC 01	Servidor virtual Intel CORE i5, procesador 2.5Ghz, 1 procesador, 1.5 Gb RAM, Ubuntu 15.10 32 bits
3	VPC 01	Servidor virtual Intel CORE i5, procesador 2.5Ghz, 1 procesador, 1.5 Gb RAM, Fedora 19 (Schrödinger's Cat) 64 bits
4	WiFi	D-Link

Las herramientas de software que fueron utilizadas para el experimento permitieron realizar las tareas de simulación de ataque al aplicativo web desplegado en la infraestructura, medir los efectos de los ataques y mitigar el ataque en base de datos. Se cuenta así con los siguientes componentes (Tabla II):

TABLA II. HERRAMIENTAS DE SOFTWARE USADAS EN EL EXPERIMENTO

Herramienta	
Nº	***Detalles***
1	VMware® Workstation 12 Pro V. 12.0.0 build-2985596
2	Servidor Web Apache 2.4
3	Postgresql 9.3
4	PGBouncer 1.7.2
5	Apache JMeter 2.13
6	Sysstat 11.2.1.1 released (stable version).
7	PHP 5.0
8	Eclipse IDE for PHP Developers Mars .1 Release 4.5.1

Se muestra a continuación la topología de la aplicación web que se ha empleado en el experimento en sus dos fases: 1) sin PGBouncer y, 2) con PGBouncer.

El primer experimento (Figura 2) muestra la configuración de una topología clásica de aplicaciones web en una máquina virtual Ubuntu con un servidor de aplicaciones Apache desarrollada en PHP 5. Este experimento permitió obtener la línea base de medición de resultados al experimentar llamadas hacia la base de datos residente en el servidor Fedora.

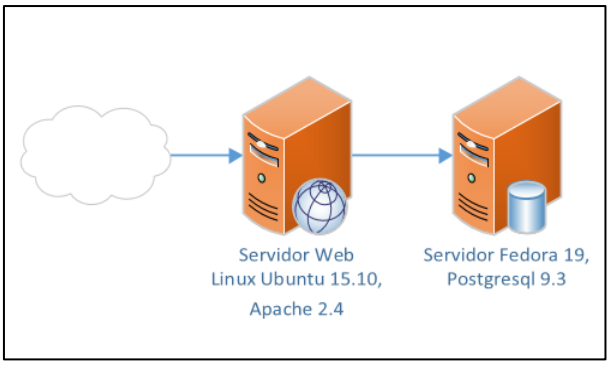


Figura 2. Topología sin PGBouncer

En el segundo experimento, en el cual se planteó la mitigación al problema de ataques DDoS, se mantuvo la misma configuración del primer experimento, agregando a la arquitectura el aplicativo PGBouncer el cual fue desplegado entre el servidor web de aplicación y el servidor de base de datos.

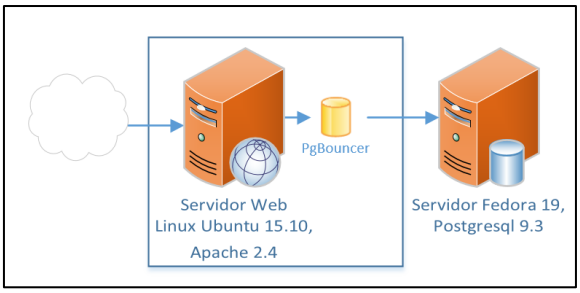


Figura 3. Topología con PGBouncer

PGBouncer funcionó a modo de gestor de conexiones, siendo un middleware entre la aplicación y el servidor de base de datos, administrando el pool de acceso como se aprecia en la Figura 3.

B. Configuración del ataque

Para la realización del ataque, se construyó el aplicativo web dvdrental, con dos páginas PHP. En la primera (/dvdrental/busqueda.php) se incluyó el ingreso de datos para la consulta. Una captura de pantalla se muestra en la Figura 4.

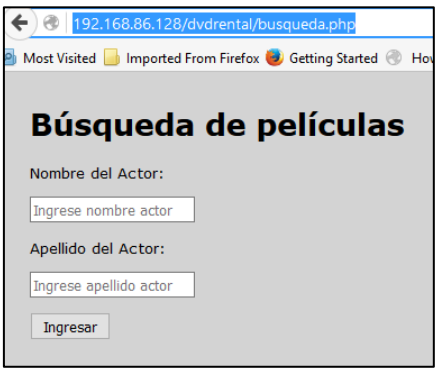


Figura 4. Página busqueda.php

Por medio del botón "ingresar" las variables de búsqueda se dirigen a la segunda página php que realiza la consulta a la base de datos (/dvdrental/busquedaAction.php?) En esta se ejecuta la

consulta mostrada en la Figura 5:

```
select distinct f.title,
                f.description,
                a.first_name ||' - ' || a.last_name as actor
        from    film f, film_actor fa, actor a
        where   f.film_id = fa.film_id
            and fa.actor_id = a.actor_id
            and a.first_name ilike'%a%'
            and a.first_name ilike '%a%'
```

Figura 5. Consulta SQL de prueba

Las tablas utilizadas para construir la consulta de prueba pertenecen a la base de datos "dvdrental" creada en el servidor. Esta es una base de datos Postgresql de prueba descargable con fines académicos [7].

El modelo de consulta incluyó combinaciones de tres tablas (film, film_actor y actor) agregando un filtro por medio del comando "ilike". La ejecución de forma concurrente de esta consulta indispuso los servicios en la infraestructura, provocando demoras y fallas en la ejecución de los comandos en la base de datos y por tanto, fallas en el sistema en general.

La conexión a la base de datos desde el aplicativo desplegado en el servidor Apache se la realizó de forma directa a través de la siguiente cadena de conexión: (Figura 6)

```
host='192.168.86.130' port='5432' dbname='dvdrental'
user='usuario' password='password'
```

Figura 6. Cadena de conexión a base de datos directa

Para realizar el ataque de denegación de servicio, se utilizó la herramienta Apache JMeter, a través de la cual se configuró 5 escenarios de prueba para validar el experimento. Cada escenario estaba representado por 500, 1000, 1500, 2000 y 2500 hilos respectivamente. Cada hilo representó un usuario virtual realizando cada uno de ellos 1 transacción sobre la aplicación web.

C. *Propuesta de mitigación*

En el caso del ataque de denegación de servicio ejecutado en el experimento y su afectación al servidor de base de datos, se propuso el uso del middleware PGBouncer. El mismo se instaló en el servidor de aplicaciones Ubuntu como se describe en la Figura 7.

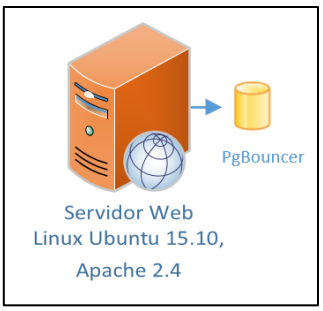


Figura 7. PGBouncer en el servidor de aplicaciones

Los parámetros de configuración se encuentran en el archivo */etc/PGBouncer /PGBouncer .ini* en el cual se definió parámetros como la cadena de conexión a la base de datos remota, la dirección IP y el puerto de escucha para conexiones locales, el tipo de autenticación, el archivo de usuarios, el modo de Pool de conexiones, el máximo número de conexiones desde el aplicativo hacia PGBouncer y desde PGBouncer hacia la base de datos remota, entre otros parámetros (Figura 8).

```
[databases]
dvdrental= host=192.168.86.130 port=5432 dbname=dvdrental
user=[redacted] pool_size=80

[pgbouncer]

logfile = /var/log/postgresql/pgbouncer.log
pidfile = /var/run/postgresql/pgbouncer.pid
listen_addr = 127.0.0.1
listen_port = 6432
auth_type = trust
auth_file = /etc/pgbouncer/userlist.txt
pool_mode = transaction
server_reset_query = DISCARD ALL
max_client_conn = 300
default_pool_size = 20
server_lifetime = 1200
server_idle_timeout = 60
client_idle_timeout = 60
```

Figura 8. Configuraciones PGBouncer

De esta forma, instalado el software propuesto de mitigación para el ataque experimental, se configuró la cadena de conexión del aplicativo de la siguiente manera (Figura 9):

```
host='127.0.0.1' port='6432' dbname='dvdrental'
user='usuario'
```

Figura 9. Configuraciones PGBouncer

Por medio de esta cadena de conexión, la aplicación web de prueba se conectó a la base de datos a través de PGBouncer y no directamente al servidor. Se ejecutó el mismo escenario de pruebas descrito en la Configuración del Ataque.

En la ejecución, PGBouncer creó un pool de conexiones hacia la base de datos remota generando un balanceo de carga en la ejecución de consultas hacia la base de datos; esto ayudó a mitigar la saturación de recursos en el servidor de base de datos Postgresql.

D. *Pruebas y medición*

Realizadas las pruebas de conformidad a lo descrito en la configuración del ataque (B) y la propuesta de mitigación (C), los resultados de las mediciones se colectaron por medio del archivo de salida de Apache JMeter en donde se obtuvo el tiempo de respuesta de las ejecuciones, la latencia, el número de bytes recibidos como respuesta, etc.

Las mediciones se guardaron en archivos .csv para cada una de las ejecuciones y se los procesó para obtener las estadísticas de tiempo de respuesta promedio para cada uno de los experimentos sin el uso de PGBouncer y usando PGBouncer para mitigar el ataque.

Para verificar los parámetros de funcionamiento de las pruebas iniciales en el servidor de aplicaciones y base de datos se utilizó el aplicativo sysstat tanto para medición de uso de CPU como para medir el porcentaje de consumo de memoria. Se usó comandos similares a los de la figura 10, recopilando los resultados para su posterior análisis:

```
sudo sar -r 1 5000 > cpb_100_100_100_mem.txt (Memoria)
sudo sar -u 1 5000 > cpb_100_100_100_cpu.txt (CPU)
```

Figura 10. Configuraciones PGBouncer

IV. Evaluación de resultados y discusión

Como producto de la experimentación se obtuvieron mediciones en parámetros como tiempo promedio de respuesta, porcentaje de éxito de las búsquedas, porcentaje de uso de CPU y de memoria en el servidor de base de datos.

Para tiempo promedio de respuesta (Tabla III), los tiempos de ejecución en el servidor de aplicaciones aumentaron cuando la solución trabajó con PGBouncer con relación a cuándo no se utilizó el mencionado middleware.

TABLA III Tiempos Promedios de Respuesta

	Tiempo Promedio Respuesta (ms)	
Usuarios Concurrentes	**SIN PG BOUNCER (Usarios)**	**CON PG BOUNCER (Usuarios)**
500	1033,32	6331,31
1000	8895,00	9291,74
1500	9227,24	10004,15
2000	28859,21	40259,05
2500	37140,60	47678,43

Esto se debe a que cuando trabajó en su función de balanceador, el programa PGBouncer generó un pool de conexiones por medio del cual se canalizaron las instrucciones SQL (Figura 11).

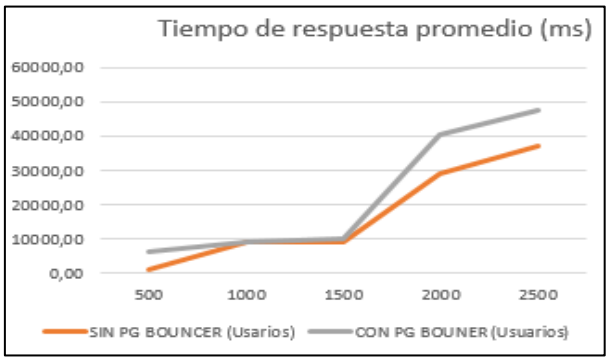


Figura 11 Tiempo de respuesta promedio

Cuando masivamente se enviaron peticiones al servidor de base de datos, PGBouncer detuvo el tráfico a la base de datos y realizó las consultas conforme a la configuración de máximas conexiones configurado en el archivo pgbuncer.ini. Esto repercutió en la respuesta de las páginas.

En cuanto al éxito de las ejecuciones se verificó y registró en la Tabla IV que el porcentaje de éxito es mayor cuando se usó el balanceador de carga, esto debido a que al canalizarse las ejecuciones por el pool de conexiones, este detuvo las transacciones y las realizó ordenadamente aplicando el balanceo de carga.

TABLA IV Tiempos Promedios de Respuesta

	Porcentaje de Éxito (sobre 100%)	
Usuarios Concurrentes	**SIN PG BOUNCER (usuarios)**	**CON PG BOUNCER (Usuarios)**
500	100,00	100,00
1000	100,00	100,00
1500	88,01	88,96
2000	55,68	72,59
2500	50,12	53,24

En la Figura 12 también se aprecia que las ejecuciones exitosas desde el servidor de aplicaciones se vieron comprometidas en este caso para un ataque de denegaciòn de servicios a pesar de que hubo una mejora con el uso de PGBouncer .

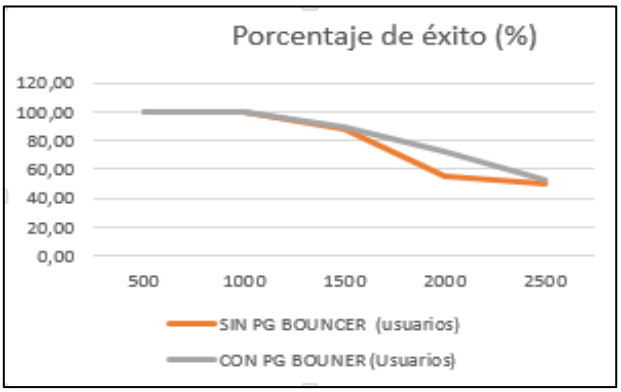


Figura 12.Porcentaje de éxito

En lo referente a los porcentajes de uso de procesador, se obtuvo datos que reflejan que no existió una gran diferencia entre las medidas con el balanceador de carga y sin el mismo en el servidor de base de datos que es de donde se obtuvieron estas medidas (Tabla V).

TABLA V Porcentaje de uso de CPU

	% Uso de CPU	
Usuarios Concurrentes	**SIN PG BOUNCER (usuarios)**	**CON PG BOUNCER (Usuarios)**
500,00	16,30	16,20
1000,00	16,85	16,52
1500,00	17,52	18,20
2000,00	21,30	17,21
2500,00	22,50	20,90

El comportamiento del porcentaje de uso de CPU para el servidor de base de datos cuando estuvo sin ningún tipo de actividad fue de hasta el 2 %, por lo que los porcentajes obtenidos luego del experimento nos indicaron de todas maneras que se aumentó el uso de este recurso (Figura 13).

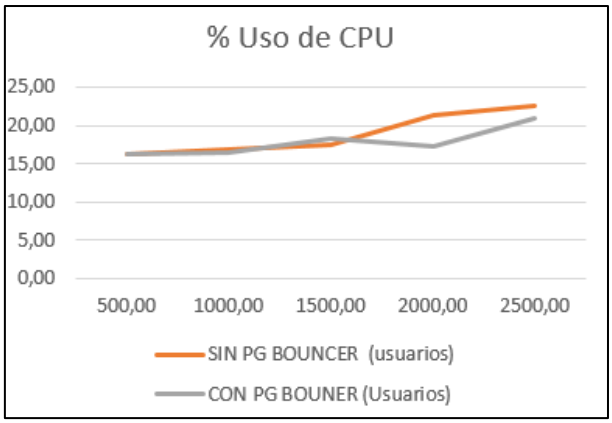


Figura 13.Porcentaje uso de CPU

Al realizarse la experimentación, sin PGBouncer se alcanza porcentajes de uso superiores al 95% de memoria RAM ocasionando mal funcionamiento de la base de datos, indisponiendo el servicio (falta de conexión a la base de datos) o en el mejor de los casos haciendo las consultas mucho más lentas.

Usando PGBouncer se aprecia una disminución en el porcentaje de uso de memoria en el servidor de base de datos, conforme a las mediciones realizadas. (Tabla IV).

TABLA VI PORCENTAJE DE USO DE CPU

	% Uso de Memoria	
Usuarios Concurrentes	**SIN PG BOUNCER (Usarios)**	**CON PG BOUNCER (Usuarios)**
500,00	93,51	87,90
1000,00	94,51	87,80
1500,00	95,21	88,90
2000,00	95,20	84,57
2500,00	95,23	91,10

El funcionamiento de la memoria en el servidor de base de datos registrado durante el experimento se puede apreciar en la Figura 14.

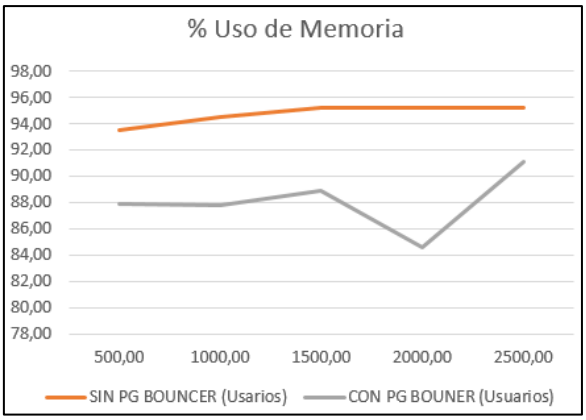


Figura 14.Porcentaje uso de CPU

V. TRABAJOS RELACIONADOS

Ataques DDoS son comunes en el entorno de Internet siendo una de las razones el bajo precio que se puede pagar por realizar un ataque. Por tal motivo constantemente se emplea esfuerzo en la búsqueda de mecanismos de defensa y mitigación de este tipo de ataques, mecanismos necesarios a nivel de aplicación, de red de datos y de base de datos.

Considerando que una base de datos de cualquier organización almacena información valiosa para su propietario, el presente trabajo constituye un aporte hacia la implementación de mecanismos de defensa a nivel de base de datos que sean de bajo costo y que utilicen PostgreSQL como su sistema gestor de base de datos, tomando en cuenta como se describió en la sección II de este artículo, que PostgreSQL se está convirtiendo en una solución libre ampliamente aceptada a nivel mundial.

Existen estudios importantes sobre mitigación de ataques DDoS, Santanna, J. J., Durban, R., Sperotto, A., & Pras, A, exponen en [8] un análisis e identificación de las características de algunos sitios web denominados Booters con el objetivo de facilitar trabajos futuros, en vista que estos Booters están facilitando la ejecución de ataques DDoS en Internet. En [9] se propone un mecanismo de mitigación basado en redundancia de tablas, de manera que el servicio siempre esté activo. Con igual objetivo, Yujie, Z. H. A. O., Bhogavilli, S., & Guimaraes, R, proponen en [10] un sistema implementado por computador para detectar en base a mensajes transmitidos de solicitud y respuesta las direcciones IPs de los atacantes y ejecutar procesos de bloqueo a los atacantes. Y en lo que respecta a mitigación de ataques en redes, en [11] se propone un algoritmo para detectar y mitigar ataques en redes Wireless 3G/4G.

La mayoría de trabajos se han enfocado en soluciones a nivel de red, de servidores web y aplicaciones, por lo que el presente artículo complementa los estudios realizados y aporta a conseguir mayor protección contra ataques de denegación de servicio.

VI. CONCLUSIONES Y TRABAJOS FUTUROS

Las medidas de mitigación frente a ataques DDoS que deben tomar los administradores de sistemas, deben apuntar a todos los componentes de una aplicación. Uno de los componentes críticos en una arquitectura de sistemas es la base de datos, por lo que su correcto funcionamiento y seguridad se vuelve un aspecto crítico. Muchas veces no se contempla implementar soluciones en base de datos, enfocándose más en la seguridad de los servidores Web, en los firewalls y otros tipos de medidas de protección.

Por medio de la experimentación expuesta, se ha demostrado que mediante el uso de administradores de conexiones como PGBouncer, además de optimizar el funcionamiento de las aplicaciones, ayuda a mitigar ataques DDoS contra la base de datos. Los resultados de consumo de recursos en el servidor indicaron una mejora, evitando el colapso de la aplicación en conjunto.

El desempeño de la herramienta de mitigación examinada en el experimento tuvo como contraparte en el servidor de aplicaciones, un aumento del tiempo de respuesta en el

despliegue de las consultas a la base de datos. Esto nos indica que a pesar de que se mitigó el efecto del ataque en el servidor de base de datos, debe de tomarse medidas complementarias en los otros componentes de la arquitectura de un sistema, para que toda la aplicación trabaje en forma adecuada.

Como trabajo futuro se pretende atacar esta última problemática examinando la posibilidad de implementar a parte de la utilidad para realizar balanceo de carga una solución basada en un WAF u otras implementaciones en Firewall, así como mejoras en los métodos de prevención de ataques de SQL Injection entre otras optimizaciones.

REFERENCES

[1] INEC (2013) Disponible en: http://www.ecuadorencifras.gob.ec/tecnologias-de-la-informacion-y-comunicacion-tic/

[2] Disponible en: http://es.slideshare.net/agenciavertice/anlisis-de-estadsticas-internet-y-redes-sociales-de-ecuador-a-junio2014-por-elerick

[3] DB-Engines, "Popularity of open source DBMS versus commercial DBMS" [online], 2016. Disponible en: http://db-engines.com/en/ranking_osvsc.

[4] Squicciarini, A. C., Paloscia, I., & Bertino, E. (2008, April). Protecting databases from query flood attacks. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on* (pp. 1358-1360). IEEE.

[5] EnterpriseDB Corporation, How to Set Up PgBouncer Connection Pooling for Postgres Plus Standard Server, 2010, Disponible en: http://get.enterprisedb.com/docs/Tutorial_All_PPSS_pgBouncer.pdf

[6] Holmes, D. (2013). Mitigating DDoS Attacks with F5 Technology. *F5 Networks, Inc*.

[7] Postgresql Tutorial. Disponible en: http://www.postgresqltutorial.com /postgresql-sample-database/

[8] Mahajan, D., & Sachdeva, M. (2013). DDoS Attack Prevention and Mitigation Techniques-A Review. *International Journal of Computer Applications*, *67*(19).

[9] Santanna, J. J., Durban, R., Sperotto, A., & Pras, A. (2015, May). Inside booters: an analysis on operational databases. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on* (pp. 432-440). IEEE.

[10] Silva, Mauro., Romero, Diego.,Bastidas, Christian., & Fuertes, Walter. (2015, Noviembre). Mitigacion de Ataques DDoS a traves de Redundancia de Tablas en Base de Datos. In Memorias VIII Congreso Iberoamericano de Seguridad Informatica, 2015, on (pp.56-62).

[11] Yujie, Z. H. A. O., Bhogavilli, S., & Guimaraes, R. (2014). *U.S. Patent No. 8,869,275*. Washington, DC: U.S. Patent and Trademark Office.

[12] Gupta, A., Verma, T., Bali, S., & Kaul, S. (2013, January). Detecting MS initiated signaling DDoS attacks in 3G/4G wireless networks. In *Communication Systems and Networks (COMSNETS), 2013 Fifth International Conference on*(pp. 1-60). IEEE.

Evaluación del Ataque ShellShock

William Sani, Roger Jaimes, y Jessenia Ramón

williamwrsa@gmail.com, rogerj007@gmail.com, jessenia.ramonc@gmail.com

Abstract— **Actualmente las aplicaciones Web son cada vez más utilizadas por los usuarios en el ciberespacio, lo que ha provocado que cada vez se vayan incrementando y descubriendo las vulnerabilidades de los servidores y aplicaciones a fin de realizar ataques informáticas con un fin determinado. Una vulnerabilidad presente en algunos servidores Web, es el conocido ShellShock, que es una vulnerabilidad en el Shell Bash de los sistemas operativos Linux/Unix, el cual permite ejecutar comandos por atacantes de manera remota, por lo que se le conoce también con el nombre de Bashdoor. Considerando lo indicado, se realizó un análisis y evaluación del vector de ataque usado por ShellShock utilizando herramientas open source bajo un ambiente virtualizado de experimentación, el cual permitió de manera práctica realizar un ataque backdoor a un servidor Web vulnerable. Adicionalmente se instaló un WAF, como un mecanismo de mitigación para este tipo de ataques. De la experimentación realizada se pudo determinar la criticidad de este tipo de ataques y la importancia de realizar una actualización de dispositivos que utilizan Bash Shell o implementar WAF a fin de mitigar ese tipo de ataques.**

Palabras clave— *Exploit, CVS, ShellShock, pruebas de penetración, WAF.*

I. INTRODUCCIÓN

Los ataques "ShellShock" o Bashdoor podrían permitir a un atacante tomar control remoto de millones de servidores y computadores en todo el mundo. El nombre oficial de esta vulnerabilidad es GNU Bash Remote Code Execution Vulnerability (CVE-2014-6271). Esta vulnerabilidad permite a hackers realizar ataques con inyección de código remoto y tomar el control del sistema objetivo Linux, Unix o Mac. En este momento más de la mitad de los servidores de Internet y teléfonos Android se encuentran afectados, dado que el 51% de los servidores web de todo el mundo funcionan con Linux.[1] El alcance contemplado por esta falla teniendo en cuenta que no se requiere autenticación para explotar esta vulnerabilidad sobre el código abarca principalmente la divulgación no autorizada de información; modificación sobre la configuración del sistema operativo y la interrupción del servicio derivada de los dos puntos anteriores.[2]

La comunidad científica ha investigado e implementado mecanismos para disminuir y mitigar estos ataques. La mayoría de distribuciones de GNU/Linux han lanzado actualizaciones tanto para sistemas de escritorio como para servidores Linux; pero el verdadero problema radica en aquellos sistemas que no se actualizan (parches), bien porque no hay nadie que los mantenga o porque se trata de sistemas incrustados en dispositivos que no están preparados para recibir actualizaciones o el fabricante ha dejado de publicarlas para ciertos modelos[3]. También se debe considerar que esta vulnerabilidad puede realizarse utilizando el protocolo SSH y con DHCP, esto en determinados equipos, como por ejemplo: routers y switch, que pueden estar sin soporte por parte de los fabricantes por lo que algunos de estos seguirán con la vulnerabilidad.

Esta investigación se enfoca en el análisis y evaluación del ataque ShellShock, utilizando como plataforma de experimentación un ambiente virtual de red para identificar como actúa dicho ataque, utilizando VMware como hypervisor. Para llevarlo a cabo se ha implementó una red LAN y WAN con la finalidad de inhabilitar los accesos internos y eternos. Como contribución se ha implementado un Web Application Firewall para mitigar este tipo de ataques. Se instaló el WAF Open Source ModSecurity, el cual se integra con el servidor Web Apache.

Finalizando la experimentación se pudo determinar la criticidad de este tipo de ataques y la importancia de realizar una actualización de dispositivos que utilizan Bash Shell o implementar otros mecanismos de seguridad como WAF, a fin de mitigar ese tipo de ataques.

Las principales contribuciones del presente trabajo son: i) análisis y evaluación del ataque ShellShock; y, ii) implementación de un WAF que permita detectar y mitigar estos ataques.

El resto del artículo ha sido organizado de la siguiente manera: la sección 2 describe el fundamento teórico. La sección 3 muestra los componentes del experimento, así como el proceso de mitigación. La sección 4 ilustra los resultados obtenidos. La sección 5 muestra trabajos relacionados con el tema, Finalmente en la sección 6 se exponen las conclusiones y trabajo futuro.

II. FUNDAMENTO TEÓRICO

La vulnerabilidad en el Shell bash, de sistemas operativos Unix/Linux, tiene algunos vectores de ataques. Una vulnerabilidad es un punto abierto en un o más sistemas informáticos que podría afectar los objetivos de confidencialidad, integridad, disponibilidad, no repudio y autenticación de la información. Un vector de ataque, por su parte, es el método que utiliza una amenaza para atacar un sistema. Entre los principales se tienen los siguientes:

A. *Servidores HTTP*

El error ShellShock está atacando principalmente los servidores Web HTTP. Aquellos servidores que se ejecutan en FastCGI o CGI son capaces de exponer el bash al vector de petición de HTTP. Las peticiones HTTP maliciosas permiten a los ciberdelincuentes integrar comandos en el servidor y el Bash puede ejecutarlas.

El atacante puede utilizar esta conexión para realizar diferentes tipos de ataques, como DDOS o para obtener información.

Existen lenguajes de programación, como Perl, PHP y scripts de Python que no son utilizados a través de los sistemas de CGI anteriormente mencionados por lo que probablemente no se verán afectados por esta vulnerabilidad.[4]

B. *Clientes DHCP*

Los clientes de DHCP también podrían ser vulnerables debido a del error ShellShock. Esto es válido para UNIX y el sistema Linux, pero no está afectando al sistema OSX.

Los clientes DHCP pueden pasar comandos de Bash, un sistema vulnerable puede ser atacado cuando se conecta a una red abierta Wi-Fi. Un cliente DHCP típicamente solicita y recibe direcciones IP de un servidor DHCP, pero también puede proveer una serie de opciones adicionales. Un servidor DHCP infectado puede proveer, en una de estas opciones un string hecho de tal forma que ejecute un código malicioso en una computadora de trabajo

Durante el ataque, el criminal cibernético también puede utilizar el vector CGI con el fin de poner en peligro el servicio DHCP en un servidor que es legítimo.[4]

C. *SSH*

La mayoría de los sistemas de SSH están configurados de tal manera que restringe los comandos que el usuario puede aplicar. Los atacantes utilizan el bug Bash en las sesiones SSH con el fin de ir vulnerar las restricciones aplicadas. Sin embargo, esto requiere autenticación y es por eso es que este vector ofrece una escalada de privilegios.

Los sistemas que utilizan SSH, incluyendo rsync, rlogin, subversión, y otros también pueden verse afectados.[4]

D. *Sistema Unix Printing Común (CUPS)*

Un servidor de impresión, el Common UNIX Printing System está disponible en muchos UNIX, los sistemas BSD y Linux. Trabaja con variables que son controlados por el usuario y basado en esto se establecen las variables de entorno en el tratamiento de los filtros. Puede actuar como un vector para la vulnerabilidad Sheshock, en el caso de que el Bash sea inicializado por el sistema de impresión común de UNIX. Actualmente, este vector es teórico.

E. *Shellshock Attack*

El concepto de ShellShock attack consiste en el uso de la vulnerabilidad en el Shell bash. El Shell se utiliza para ejecutar comandos en Unix / Linux; es decir, actúa como un intérprete de lenguaje de comandos. ShellShock puede incluso afectar a las versiones más recientes del Shell bash. También se conoce como el error bash. Permite a un atacante obtener el control sobre el equipo. Un tipo de variables en bash son las variables ambientales. Una vulnerabilidad en el bash se gana a través de esta variable ambiental. Aunque ciertas condiciones tienen que cumplirse para la explotación de la vulnerabilidad, una vez que su atacante tiene éxito puede hacerse con el control del servidor remoto.

En la figura 1 se esquematiza la vulnerabilidad ShellShock. En general, el código dentro de la función tendrá problemas en su ejecución, mientras que el código fuera de las llaves se ejecutará, esto es la vulnerabilidad CVE-2014-6271. El atacante puede acceder remotamente usando un payload, de ahí el nombre bashdoor.

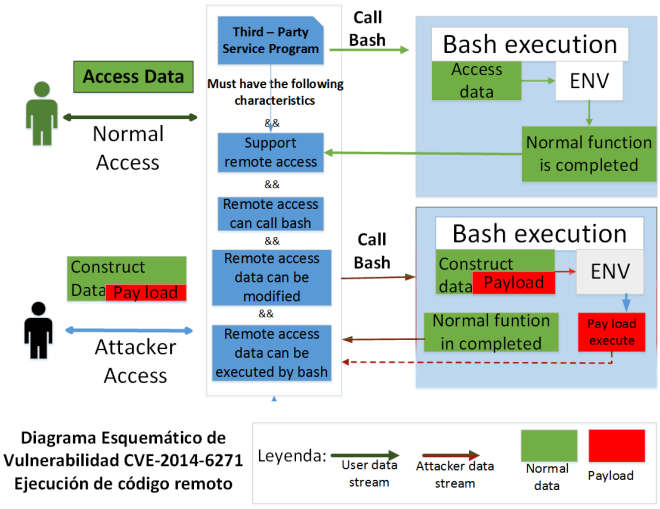


FIGURA 1. Vulnerabilidad ShellShock

Hoy en día existen algunas herramientas para realizar pruebas de penetración, una de ellas es Metasploit, la cual será utilizada para realizar la explotación de la vulnerabilidad en el Shell hacia un servidor Web, utilizando payload, para establecer una conexión remoto como la indicada en la figura 1.

III. CONFIGURACIÓN DEL EXPERIMENTO

A. *Diseño e Implementación topología de prueba*

Para realizar el experimento, se implementó un ambiente virtual experimental, utilizando VMware como hypervisor. A continuación, en la figura 2, se presenta la topología empleada:

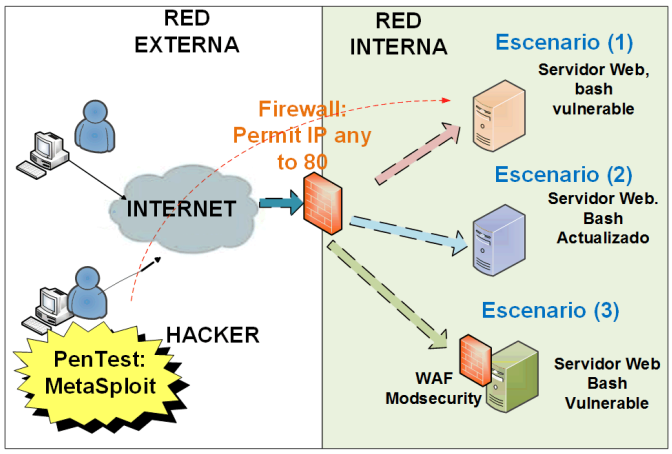


Figura 2. Ambiente Virtualizado de Experimentación

Como se aprecia en la figura 1, el ambiente virtualizado consta de los siguientes elementos: En la red interna o LAN: Un servidor Web, con Ubuntu 12, en el perímetro un firewall de capa 3 y en la red externa un equipo con la distribución de Linux Back Box.

El firewall de capa 3, está configurado de tal modo que permite el acceso desde usuarios de la red externa (PC-ATAQUE) a través del puerto 80 hacia el servidor Web (PC-VÍCTIMA) y también al puerto 22, para realizar actividades de administración.

Para el servidor Web se tienen tres escenarios, en todos se tiene el servidor web Apache, en el cual se habilitó CGI (Common Gateway Interface), para la ejecución de script CGI. En el primer escenario, se verificó que la versión del sistema operativo Ubuntu, tenga la vulnerabilidad en el Shell bash, *CVE*-2014-6271, para realizar la prueba de explotación. Para el segundo escenario, se tiene un servidor de similares características, pero con el parche para el Bash. El tercer escenario, es el mismo servidor que el primer escenario; no obstante, en este servidor se tiene instalado un servidor WAF.

En lo que respecta al equipo que realiza el ataque se utilizó la distribución de Linux Back Box, ya que cuenta con herramientas de ethical hacking preinstaladas; y, para la experimentación se utilizará la herramienta Metasploit.

B. Configuración de los componentes de experimentación

Como se indicó anteriormente, el firewall de capa 3, permite el acceso hacia el servidor Web por el puerto 80, esto se verifica accediendo al servidor desde un cliente utilizando un navegador web. Adicionalmente se habilita el CGI en el servidor Web, lo cual se comprueba ejecutando un script tipo CGI.

Seguido de esto, para el primer escenario, se realiza un escaneo de manera general a fin de identificar o corroborar que el servidor Web tenga la vulnerabilidad ShellSock, esto se lo realiza con el siguiente comando.

curl -A "() { :; }; echo; /bin/cat /etc/passwd" servidor-web/cgi-bin/script.sh > dat2.txt

Si el servidor tiene la vulnerabilidad, se ejecutará el comando "cat /etc/passwd" y dicha información se almacenará en el archivo dat2.txt.

Una vez verificada la vulnerabilidad, se procede a realizar la explotación, para lo cual se configura el Metasploit que tiene precargado un exploit para la vulnerabilidad *CVE*-2014-6271. En el exploit, se colocan parámetros tales como la dirección del servidor remoto - víctima, el directorio CGI con el acceso al script existente en servidor; adicionalmente, se configura el Payload, el mismo que permite establecer una conexión remota inversa entre el servidor víctima y el atacante; es decir, si se realiza el ataque satisfactoriamente, se inicia un túnel entre el PC atacante, desde donde se pueden ejecutar comandos de manera remota en el servidor Web.

C. Propuesta de Mitigación

A continuación se explican dos propuestas de mitigación para los ataques tipo ShellShock:

i) Actualización del bash del sistema operativo - Escenario 2. Los sistemas operativos tienen diferentes parches los mismos que sirven para corregir errores o agregar funcionalidades. En este sentido, existen parches para el bash, que permiten corregir la vulnerabilidad del bash evitando de esta manera los ataques del tipo ShellShock. Para verificar la mitigación propuesta, en el escenario 2, se realizará la actualización del bash del sistema operativo en el servidor víctima, después se procede a realizar un ataque tipo ShellShock con la herramienta MetaSploit, hacia el servidor Web.

ii) Implementación de un Web Application Firewall WAF – Escenario 3. Para mitigar el ataque tipo ShellShock hacia servidores web, se puede utilizar un Firewall a nivel de capa aplicación, a fin de bloquear los ataques de exploit. Para la experimentación se instaló un WAF basado en software libre denominado ModSecurity, el cual se integra con el servidor web vulnerable.

El resultado de estas propuestas de mitigación se expone a continuación en la sección IV

IV. Evaluación De Resultados Y Discusión

A. Escenarios y resultados de Experimentación.

Como se indicó anteriormente el objetivo de esta investigación es realizar la evaluación del ataque tipo ShellShock con un ambiente de experimentación virtualizado, considerando para esto la realización de una prueba de penetración aprovechando la vulnerabilidad del bash, para analizar la problemática de este tipo de ataque. A continuación se detallan los escenarios de experimentación:

Escenario (1): Como se indicó en el apartado 3.1, en el primer escenario, se verificó que el servidor víctima tiene la vulnerabilidad en el bash, con el comando:

env x='() { :;}; echo vulnerable' bash -c "echo Esta es una prueba"

El resultado de ese comando retorna:

vulnerable
Esta es un Prueba

Adicionalmente, se enviaron comandos básicos de Linux hacia el servidor web víctima, como por ejemplo: la

sentencia *curl -A "() { :; }; echo; /bin/ls" pc-víctima/cgi-bin/script.sh > dat.txt*

Con estas pruebas se pudo constatar que se puede obtener cualquier tipo de información, e incluso ejecutar comandos que puedan afectar a otros servidores tanto en la red interna como en la externa, utilizando como medio el servidor víctima, todo esto sin necesidad de utilizar herramientas avanzadas, lo cual demuestra la criticidad de esta vulnerabilidad.

Después se procedió a realizar la prueba de penetración, explotando la vulnerabilidad bash del servidor, utilizando Metasploit, obteniéndose como resultado el control del Shell remoto del servidor a través del puerto 80, utilizando el payload meterpreter. Cabe indicar que una vez que se tiene acceso al servidor, se puede ejecutar y tener acceso a determinados comandos y directorios, en función de los permisos con que se ejecute el CGI vulnerable, tal y como se indica en la figura 1.

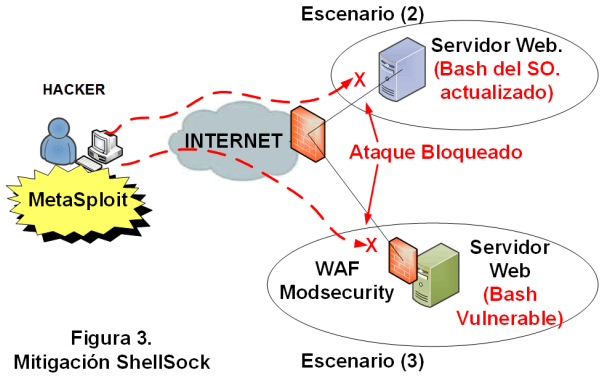


Figura 3.
Mitigación ShellSock

Escenario (2) Mitigación: En este escenario, se realizó la actualización del bash del sistema operativo Ubuntu, con los siguientes comandos "sudo apt-get update && sudo apt-get install bash"

Seguido de esto se verifica que el servidor víctima ya no tiene la vulnerabilidad en el bash, utilizando nuevamente el comando: *env x='() { :;}; echo vulnerable' bash -c "echo Esta es una Prueba"*

Ahora, cuando se ejecuta el comando, se obtiene como respuesta: *"Esta es una Prueba"*, sin la palabra *"vulnerable"*.

Para finalizar, se procede a realizar la prueba de penetración con Metasploit, constatando que ya no se tiene el acceso remoto con el exploit. Figura 3.

Escenario (3). Para este escenario se instaló un WAF en el servidor Web que tiene la vulnerabilidad en el Bash. El WAF utilizado es el Modsecurity, el mismo que tiene un conjunto de reglas que se pueden descargar desde su sitio web. Para verificar que el WAF esté habilitado en el servidor Ubuntu, se utilizó el siguiente comando: "sudo a2enmod mod-security"

Nuevamente se procedió a realizar la prueba de penetración para la vulnerabilidad ShellShock y se constató que no se puede establecer una conexión remota, usando el exploit de Metasploit, Figura 3. Adicionalmente, se revisaron los logs del WAF, en el cual se evidenció el reconocimiento del ataque (Apache-Handler: CGI-script), en el archivo modsec_audit.log

B. Discusión

Como se pudo apreciar en la prueba de penetración realizada, en el escenario (1) se tiene que la vulnerabilidad Shell Shock, es muy crítica, ya que básicamente se tiene acceso al Shell del servidor atacado, pudiendo acceder a información privilegiada éste, realizar cambios en las configuraciones de los archivos y principalmente convertirse en una maquina zombie, para producir ataques de DoS hacia otros servidores tanto internos como externos.

Se presentaron también dos formas de mitigación: una de ellas el mantener actualizado el sistema operativo Linux y el Shell bash; sin embargo, en la práctica existen empresas que tienen sistemas antiguos en los cuales no se pueda realizar dicha actualización. También se debe considerar que esta vulnerabilidad puede realizarse utilizando el protocolo SSH y con DHCP, esto en determinados equipos, como por ejemplo: routers y switch, que pueden estar sin soporte por parte de los fabricantes por lo que algunos de estos seguirán con la vulnerabilidad.

Como segunda alternativa de mitigación, un poco más compleja que la anterior, es la instalación de un servidor WAF, que para nuestro caso se integró al servidor Web, pero también se podría considerar el uso de WAF dedicados, los cuales servirían no solo para mitigar los ataques ShellShock, sino que también ayudaría a proteger al servidor de otro tipo de ataques.

V. TRABAJO RELACIONADO

La vulnerabilidad, hecha pública por el experto en seguridad Unix Stephane Chazelas, estuvo presente por más de 20 años y afecta a todas las versiones de bash hasta la 4.3

Robert Graham, experto en seguridad, considera que ShellShock pone en riesgo no sólo a los ordenadores y muchos de los servidores de la Web sino también en dispositivos o equipos que usan versiones modificadas de Linux como sistema operativo, como por ejemplo: las cámaras de vídeo IP podrían ser vulnerables ya que rara vez se actualizan.

En el trabajo realizado en [10]**¡Error! No se encuentra el rigen de la referencia.**, la evaluación del riesgo de vulnerabilidades en servidores Web, tanto para técnicos en seguridad como para usuarios no técnicos para escanear sus servidores Web y encontrar las implicaciones de las vulnerabilidades en sus sistemas. Esto mediante la construcción de una solución que realiza pruebas de concepto hacia las vulnerabilidades más críticas existentes, incluida la de ShellShock; no obstante, no se especifica el ambiente de experimentación.

En [7], Se realiza una categorización y análisis de los ataques de inyección de código, como por ejemplo SQL Injections, Cross Site Scritping y ShellSock, y se plantea una aplicación o herramienta experimental para detectar y explotar este tipo de ataques; no obstante, hoy en día ya se tienen herramientas similares que permiten mitigar este tipo de

ataques, aunque se debe tener en cuenta que estas tienen reglas que deben estar actualizándose constantemente a fin de mitigar nuevas vulnerabilidades.

También en [8], se realiza pruebas experimentales para indicar el impacto de los ataques tipo Heartbleed Bugs y los ataques tipo Bash Bug; sin embargo, en toda la experimentación se realizan pruebas únicamente de los ataques Heartbleed, quedando un vació en lo que se refiere a las pruebas con ShellShock.

En [9], se tiene que en el año 2015 los ataques a aplicaciones Web basados en ShellShock crecieron teniéndose cerca de 173 millones de ataques contra los clientes de Akamai. ShellShock también cambió significativamente el equilibrio de los ataques a través de http vs. https, en gran parte debido a que estos ataques se llevaron a cabo sobre todo a través de HTTPS. El error ShellShock fue anunciado por primera vez en septiembre de 2014 y recibió atención de los medios; como resultado de esto se espera que la mayoría de sistemas sean actualizados, por lo que el número de intentos de explotar esta vulnerabilidad vaya decreciendo. Por otro lado La proliferación de las redes robot construido a partir de dispositivos, como router, está causando un aumento de ShellShock mediante la explotación de las credenciales de inicio de sesión predeterminadas.

El National Vulnerability Database, recomienda como buena práctica, el tener un plan de mantenimiento, para actualizar parches para la vulnerabilidad Bash que están en constante actualización. Algunos creen que la solución todavía permite que ciertos caracteres que se inyecta en las versiones del bash vulnerables a través de variables de entorno especialmente diseñados. Los atacantes pueden crear nuevos métodos para eludir las restricciones de entorno y ejecutar comandos de Shell, métodos de derivación identificadas en los siguientes trabajos todavía funcionan tal como [5]: CVE-2014-6271, CVE-2014-6277, CVE-2014-6278, CVE-2014-7169, CVE-2014-7186, CVE-2014-7187.

En lo que se refiere a pruebas de penetración, en [11] se indica de manera generalizada los métodos para realizar estas pruebas para servidores Web utilizando Kali Linux. Aquí explican las formas de hacer reconocimiento de vulnerabilidades y diferentes ataques, incluidos SSL, XSS y otras técnicas de inyección de código, sin embargo, no se indican de manera específica las pruebas de explotación con ShellShock.

A diferencia de los trabajos antes mencionados, éste se centra principalmente en el ataque tipo ShellShock y se implementan dos mecanismos de mitigación, uno actualizando el bash del sistema operativo y otro mediante la utilización de un WAF.

VI. CONCLUSIONES

En el presente trabajo se pudo simular ataques hacia un servidor Web, utilizando herramientas de penetración, para explotar la vulnerabilidad ShellShock, de esta manera se constató la problemática que tiene este tipo de ataques, ya que se puede tomar el control de un equipo remotamente. A pesar de esto, se pudo ver que en la práctica existen principalmente dos opciones para mitigar esta vulnerabilidad en servidores Web, la una realizando actualizaciones al bash del sistema y otra instalando firewall a nivel de aplicación como los WAF, muchos de los cuales tienen reglas para detectar y bloquear esos ataques.

En lo que respecta a la experimentación futura, se podría utilizar los otros vectores de ataque que tiene ShellShock, SSH y DHCP, en dispositivos móviles o equipos de comunicación, para realizar ataques de DDoS. Una explotación de un servidor DHCP pueden configurar opciones y parámetros maliciosos, especialmente diseñados para que los clientes ejecuten ataques sin que se den cuenta afectando a otros clientes conectados en el mismo entorno de red.

REFERENCIAS

[1] «“ShellShock”, un peligro que se siente a distancia.,» 26 Noviembre 2014. [En línea]. Available: http://www.benditaess.com/projects/customer/etek/website/?p=45.

[2] J. Alborts, «#Shellshock, la grave vulnerabilidad en Bash – y todo lo que debes saber,» 26 Septiembre 2014. [En línea]. Available: http://www.welivesecurity.com/la-es/2014/09/26/shellshock-grave-vulnerabilidad-bash/.

[3] L. Lopresti, «Shellshock Bash Vulnerability.,» 25 Septiembre 2014. [En línea]. Available: http://www.securetia.com/shellshock-bash-vulnerability/.

[4] B. Bilbao, «Shellshock las cosas que debemos saber.,» 2 Octubre 2014. [En línea]. Available: http://sensorstechforum.com/es/shellshock-the-things-we-must-know/.

[5] NIST, «National Vulnerability Database. “Vulnerability Summary for CVE-2014-6271.” Last accessed,» 27 Septiembre 2014. [En línea]. Available: http://web.nvd.nist.gov/ view/vuln/detail?vulnId=CVE-2014-6271.2..

[6] Trend Micro Incorporated, «TrendLabs Security Intelligence Blog. “Heartbleed.”,» 27 Septiembre 2014. [En línea]. Available: http://blog.trendmicro.com/trendlabs-security-intelligence/?s=heartbleed&Submit=+Go+.

[7] Anastasios Stasinopoulos, «Commix: Detecting and exploiting command injection flaws. Department of Digital Systems, University of Piraeus,» [En línea].

[8] J. I. M. Secaira, «Niveles de Impacto: Heartbleed Bugs vs. Bash Bugs.,» Revista publicando, nº ISSN 1390-9304, pp. 65-77, 2(5). 2015.

[9] «Akamai’s [state of the internet] / security / Q3 2015 /,» 2015. [En línea]. Available: http://www.stateoftheinternet.com.

[10] B. Delamore, An Extensible Web Application Vulnerability Assessment and Testing Framework. Thesis University of Waikato,, 2015.

[11] J. A. Ansari, Web Penetration Testing with Kali Linux Second Edition.

Determinación de niveles de agresividad en comentarios de la red social Facebook por medio de Minería de Texto

Martel Wilfredo, Carranco Diego, Cevallos Daniel
Universidad de las Fuerzas Armadas – ESPE
Quito, Ecuador
{wmartel, dcarranco, dcevallos}@espe.edu.ec

***Resumen*—La presente investigación está basada en la utilización de técnicas de Text Mining para análisis de comentarios realizados por los usuarios de la red social Facebook. Además, se utilizan diccionarios, conjunto de palabras ofensivas con pesos asignados, y algoritmos como el de Levenshtein que permiten encontrar la similitud entre dos palabras. Posteriormente se procede a la clasificación de los niveles de agresividad que van desde bajo, medio y alto, con el fin de clasificar las personas o entes cibernéticos que son potencialmente una amenaza al resto de la población virtual. Finalmente, como resultado de la investigación se obtiene un listado de personas que deberían ser investigados, expulsados o bloqueados de las redes sociales para prevenir futuros incidentes.**

Palabras Clave*—*Text Mining, Distancia de Levenshtein, API de consultas de Facebook.

I. INTRODUCCIÓN

En la actualidad debido al auge de las tecnologías de la información y las comunicaciones, se genera una auto dependencia de la personas a estas técnicas y herramientas, por tal motivo es inminente el quedar expuestos a grandes amenazas[12] entre las que se puede mencionar, el hurto y divulgación de datos confidenciales, ataques a sitios web para denegación de servicios, hurto de contraseñas e información, suplantación de identidad, y la intimidación o el acoso en sitios de opinión en la web, siendo los más comunes la redes sociales, blogs y foros.

Refiriéndonos específicamente a la red social Facebook® con un poco más de 1390 millones de usuarios agrupados en su mayoría entre adolescentes y adultos jóvenes, los cuales se encuentran expuestos y vulnerables constantemente a los bajos escrúpulos de entes cibernéticos los cuales pueden o no representar personas reales, con el fin de ofender, acosar, discriminar, maltratar, estafar, persuadir, o reclutar gente con fines delictivos[13], utilizando el foro o set de intercambio de notas de las llamadas páginas de fan de Facebook.

En respuesta a este peligro, es imprescindible el reconocimiento de los niveles de agresividad de los comentantes, de esta manera, basándose en algoritmos de distancia de Levenshtein [14] para reconocer las frases y palabras clave de acuerdo a un diccionario, conjunto de palabras ofensivas, establecido a nuestra zona geográfica, correspondiente en este caso a los ubicados en Ecuador y que realizan aportes al Fan Page de las instituciones gubernamentales elegidos como muestra, de esta parte significativa de la población de comentarios se pudo determinar patrones de comportamiento y así clasificar en niveles de agresividad [29][28] a los individuos cibernéticos que muchas veces se esconden en falsos perfiles para hacer daño [13].

El trabajo de investigación por lo antes expuesto se basa en agrupar a los individuos que exponen sus comentarios en niveles de agresividad para poder clasificarlos, a diferencia del Webcrowling y Minería de Opinión, que busca determinar agrupaciones sociales con metodología de clustering para fines de publicidad. Esto significa que al identificar posibles o potenciales amenazas se puede separar a estos grupos con el fin de darle el uso adecuado a las redes sociales que son parte de la web 2.0, enfocándolo incluso con fines educativos [17].

El resto del artículo ha sido organizado por secciones en las que se expone un marco teórico con los conceptos básicos y su uso en la investigación, posteriormente en el numeral tres y cuatro nos adentramos en la metodología y la topología de investigación para la obtención de resultados confiables y consistentes, a continuación se expone los trabajos relacionados de otros autores como referencia en el numeral cinco y finalmente las conclusiones y trabajos futuros que se pretenden por los investigadores de esta publicación.

II. MARCO TEÓRICO

A continuación, se detalla una serie de conceptos acerca de técnicas, algoritmos y métodos utilizados para dar marcha, ejecución y obtención de resultados del proyecto de investigación.

A. Minería de Texto

Técnica utilizada para extraer desde un texto plano datos que puedan generar información relevante, en nuestro caso se utiliza para limpieza, reconstrucción de datos y procesamiento de los comentarios que posteriormente serán analizados por el algoritmo de Levenshtein, con la finalidad de encontrar relación entre las palabras.

B. Distancias de Levenshtein

Es una técnica matemática[14] desarrollada para determinar el número de operaciones en que una cadena puede transformarse en otra. Su campo de aplicación va desde aplicativos de correctores ortográficos, sistemas de reconocimiento de voz hasta sistemas de detección de plagios.

Con esta herramienta se puede determinar la probabilidad de similitud entre dos palabras o frases. En nuestra investigación se utiliza esta técnica para comparar cada palabra de un comentario con un diccionario definido en una base de datos.

A continuación, se exponemos un ejemplo del algoritmo de Levenshtein:

La distancia entre “hola” y “brola” es 2, porque hay que hacer 2 operaciones sobre la palabra “hola” para obtener” brola”:

substituir la “h” por una “r” e insertar una “b”.

Consideraciones:

- Cuanto más corta es la distancia entre las dos cadenas, más parecidas son. Si la distancia es 0, las dos palabras son iguales.
- El algoritmo de Levenshtein no tiene en cuenta consideraciones fonéticas. Por ejemplo, en español, dos apellidos como Hernández y Fernández.

TABLA I. ALGORITMO DE LEVENSHTEIN

```
static int Levenshtein(string s1, string s2) {
  int coste = 0;
  int n1 = s1.Length;
  int n2 = s2.Length;
  int[,] m=new int[n1+1,n2+1];

  for (int i = 0; i <= n1; i++) {
    m[i,0] = i;
  }
  for (int i = 1; i <= n2; i++) {
    m[0,i] = i;
  }
  for (int i1 = 1; i1 <= n1; i1++) {
    for (int i2 = 1; i2 <= n2; i2++) {
      coste = (s1[i1 - 1] == s2[i2 - 1]) ? 0 : 1;
      m[i1, i2] = Math.Min(
        Math.Min(
          m[i1 - 1, i2] + 1,
          m[i1, i2 - 1] + 1
        ),
        m[i1 - 1, i2 - 1] + coste
      );
    }
  }
  return m[n1, n2];
}
```

C. *Red Social Facebook*

Es un aplicativo web creado por David Zuckerberg, destinado a la interrelación virtual de personas, y representadas por usuarios o avatares en la que se permite el intercambio de información multimedia, comentarios y opiniones. Además, esta red social cuenta con una gran concurrencia de usuarios la cual es perfecta para realizar la investigación. Se debe hacer énfasis que la red social cuenta con una herramienta Graph API [14] que permite extraer información de las publicaciones realizadas.

D. *Fan page*

Es la sección o funcionalidad de la red social Facebook en donde se permite la creación de un perfil público en el que se permite publicar contenido multimedia y de opinión y los usuarios que gusten (Like) de la misma puedan dar sus comentarios e interactuar con otros internautas.

E. *Población*

En estadística corresponde al universo de datos o anotaciones posibles para el caso de la investigación sería todo el universo de comentarios públicos expuestos en las fans pages de las instituciones gubernamentales.

F. *Muestra*

Es la representación significativa de la población en este caso se ha escogido a los comentarios de los perfiles de las paginas gubernamentales del Ecuador.

G. *Patrón de comportamiento de agresividad*

En psicología estos patrones de conducta violenta varían según la frecuencia y asociación entre conductas, los tipos de conducta agresiva que manifiestan y la situación en la que se presentan dichas conductas, apareciendo los gestos de ira y la agresión verbal como conducta más frecuente [4]. A continuación se expone brevemente una clasificación conceptual de la agresividad basada en [5]:

- Agresión física: Ataque a un organismo mediante armas o elementos corporales, con conductas motoras y acciones físicas, el cual implica daños corporales.
- Agresión verbal: Respuesta oral que resuelta nociva para el otro, a través de insultos o comentarios de amenaza o rechazo.
- Agresión social: Acción dirigida a dañar la autoestima de los otros, su estatus social u ambos, a través de expresiones faciales, desdén, rumores sobre otros o la manipulación de las relaciones interpersonales.

Existen muchos más tipos de agresiones, pero de acuerdo a la clasificación clínica se ha establecido diferencias para poder distinguirlas, entre ellas:

- Agresividad reactiva: Motivada por la emoción de la ira.
- Agresividad instrumental: Busca evitar un obstáculo o lograr una meta sin que exista una motivación de la emoción de la ira.

Esta investigación, se centra en la agresividad reactiva porque es la explosión de ira del individuo bajo ciertas condiciones. Es decir, que una persona con una conducta agresiva, en su momento puede agredir físicamente o insultar a otro solo por el hecho de tener ideologías distintas. Justamente las agresiones verbales son de interés para la investigación porque se persigue obtener perfiles de agresividad de acuerdo al vocabulario o forma de expresarse de una persona.

H. *Nivel de agresividad*

De acuerdo a lo expuesto en el punto g, para poder determinar perfiles que nos indiquen el nivel de agresividad de una persona en base a su vocabulario o forma de expresarse, se debe contar con una clasificación de acuerdo a la expresión utilizada, esto significa que se debe tener un diccionario de palabras nocivas [12] y clasificarlas de acuerdo a su ofensa. En esta investigación se determina tres niveles:

- Nivel bajo: se refiere a un perfil pasivo, muy poco sociable, negativo, siempre llevando la contraria y con mucha ansiedad de ser tomado en cuenta. Las palabras utilizadas por este perfil no son tan ofensivas porque trata de encontrar un equilibrio. El tipo de individuos de este perfil, después de ofender casi siempre terminan con gesto amable.
- Nivel medio: se refiere a un comportamiento medio explosivo, negativo y solo en ciertos casos dependiendo del tipo de interacción. Se consideran accesible, y pueden manejar la situación.
- Nivel alto: se refiere a un comportamiento amenazante, dominante en todos los sentidos, quiere siempre tener la razón, aunque carezca de ella, utiliza palabras ofensivas e hirientes para tratar de que su idea sea aceptada. Además, se debe presentar mucha atención a las ofensas porque puede tornarse en realidad.

I. *Amenaza*

Se refiere a la intención o al peligro inminente de ocurrir algún tipo de actividad que ponga en riesgo la integridad física, psicológica o social de una persona o grupo de personas.

J. Zonificación

Constituye a dar un espacio geográfico, agruparlo socialmente o por ciertas características que permita el tratamiento de la información generada de forma homogénea y sin influencia de datos aislados.

K. Variabilidad Lingüística

Corresponde a los diferentes matices que tiene el lenguaje incluso en el mismo idioma que hace de una zona o grupo social diferente. Es decir, no es lo mismo decir rata en un tema de biología que rata en un entorno político. Por tal motivo es importante tomar en cuenta este aspecto al momento de realizar la limpieza de los datos y su análisis.

L. R Studio

Herramienta informática, incorpora el lenguaje R, para extracción y procesamiento de información para darle un tratamiento, matemático, estadístico basado en modelos previamente establecidos. Esta herramienta cuenta con varios paquetes de análisis de texto tal como el de Levenshtein que a su vez tiene un compendio completo de algoritmos para encontrar similitudes entre cadenas, las cuales son útiles para la investigación.

III. MATERIALES Y MÉTODOS

Para llevar a cabo la investigación planteada, se tuvo que pasar por un proceso riguroso de extracción de información, depuración y posteriormente a su análisis en las herramientas de minería de datos. A continuación, se expone la topología (ver Fig., 1) utilizada que describe de manera general, el proceso para proseguir a la explicación del paso a paso. La experimentación se centra en tener un muestreo de comentario de la red social utilizando el Graph API de Facebook, para posteriormente realizar una limpieza de los datos, que consisten en eliminar tildes palabras no deseadas, emoticones y reconstrucciones de palabras por conflictos de codificación [15]. Una vez, que la información se encuentre limpia para el análisis, se procede a extraerla utilizando la herramienta R, la cual se conecta a la base de datos de PostgreSQL y procede a clasificarla, ordenarla y guardarla por medio de un algoritmo escrito en R [13], el cual, a su vez, incorpora otros algoritmos embebidos. Hay que mencionar, que los algoritmos interactúan directamente con el diccionario de la base para identificar si el contenido del comentario es agresivo y si es, identifica la probabilidad de similitud del contenido con la del diccionario. De acuerdo a las pruebas realizadas, la probabilidad de aceptación para encontrar la similitud más aproximada entre dos palabras es de 90% [13]. Finalmente, se obtiene un listado de resultados de todo el proceso realizado, los mismos que se exponen en una hoja de cálculo para su comprobación y presentación [14]. A continuación, se describe el proceso de extracción de información y análisis de datos.

Fig. 1. Topología de Extracción y Análisis de datos

A. Perfil de agresividad:

Determinar el perfil de agresividad de una persona en base a sus publicaciones, no es un trabajo sencillo, porque involucra analizar aspectos psicológicos del individuo. En esta investigación se utilizó un diccionario [12], conjunto de palabras ofensivas, las cuales fueron ponderadas de acuerdo a su agresividad. A continuación, se muestra un ejemplo.

TABLA II. DICCIONARIO DE PALABRAS CON PONDERACIÓN

Palabras	Peso	Palabras
Maldito	1	Maldito
Lárgate	0.5	Lárgate
Estúpido	0.4	Estúpido

La Tabla I, es solo una muestra de archivo original, esto se debe a su gran magnitud de palabras que son involucradas, pero parte de esta sección esta explicada en el video [16] donde se explica pasó a paso el proceso de la experimentación.

B. Niveles de agresividad:

Permite clasificar el nivel en que el usuario se encuentra. Por tal motivo se ha definido una matriz, tal y como se muestra en la Tabla 3.

TABLA III. NIVELES DE AGRESIVIDAD POR RANGO

Nivel de agresividad	Identificador	Rango
BAJO	B	0 .1- 0.4
MEDIO	M	0.5 – 0.7
ALTO	A	0.8 - 1

- Bajo: indica todas aquellas palabras que contengan peso entre 0.1 y 0.4.
- Medio: indica todas aquellas palabras que contengan peso entre 0.5 y 0.7

- Alto: indica todas aquellas palabras cuyo peso este entre los 0.8 y 1.

C. *Facebook fan page*

Para poder analizar los comentarios de las publicaciones, se necesitaba que sean necesariamente públicos por dos cosas: la primera, porque Facebook no presenta restricciones; la segunda, porque las personas no se limitan a decir lo que piensan, y eso es justamente lo que se buscaba en la investigación, que las personas sean libres de expresarse, para poder determinar su nivel de agresividad a través de sus comentarios. Así que, se escogió las fans page del gobierno por las cantidades enormes de comentarios realizados.

D. *Facebook Graph API*

Es una API de Facebook [14] que permite interactuar a los desarrolladores con Facebook y de esta formar tener acceso a los datos, pero solo funciona con páginas públicas de Facebook. Cabe mencionar que requieren de token, la URL y un identificador del post. Los identificadores del post, es una numeración que facebook asigna a cada comentario que un usuario realiza. A continuación, se describe un ejemplo de lo expuesto anteriormente.

Se establece la URL del a fan page de la cual se desea extraer todos los comentarios. URL a ser analizada: https://www.facebook.com/MashiRafael/photos/a.339953076034199.95492.248984428464398/1080747815288051/?type=1&theater

De esta URL lo que interesa es el identificador [id_post]:1080747815288051. Este identificador permite obtener todos los comentarios de dicha publicación, además devuelven datos de las personas que comentaron incluido la fecha de publicación.

Ahora se hace uso del API de Facebook para obtener los comentarios.

http://graph.facebook.com/[id_post]:/comments?limit=numComentarios&token=

- id_post: es el identificador
- numComentarios: cantidad de comentarios a mostrar.
- token: código generado por facebook, la cual te permite tener acceso a consultas via query por un tiempo determinado.

Al reemplazar los valores, usted obtendrá todos los comentarios realizados en el post.

E. *Transformación de JSON a filas y columnas*

Los datos obtenidos del API de Facebook, retornan datos en json y estos necesitan ser transformados a una estructura estática definida por nosotros en la base de datos y para posteriormente ser leídos en la base de datos postgreSQL y analizarlos con la herramienta R.

Para el proceso de transformación se utilizó la propia herramienta de postgreSql, es decir la base misma soporta almacenar datos en json y así mismo permite recorrerlos como si fuera una Tabla normal, ver Tabla 4.

F. *Inserción de Json a Base de Datos postgreSQL:*

Para este proceso se utilizaron sentencias SQL que permiten el trasformado de los nodos Json a registros de texto plano. A continuación, se muestra un ejemplo de la conversión de Json a texto plano.

TABLA IV. CONVERSIÓN DE JSON A TEXTO PLANO

```
SELECT (o->'from'->>'name') as Nombre,
       (o->'from'->>'id') as Id,
       (o->>'message') as Mensaje,
to timestamp((o->>'created time'),'YYYY-MM-DD"T"HH24:MI:SS"Z"') as FechaPublicacion,
       (o->'comments'->'data') as Comentario,
  FROM sitio s
  , json_array_elements(data->'data') o
```

Lo expresado en la Tabla 4, es la forma de transformar json a texto plano, sin necesidad de requerir a herramientas de terceros.

G. *Depuración de datos la base PostgreSQL*

Para poder analizar la información en R, es necesario que haya palabras completas y estén casi correctamente escritas, de caso contrario habrá problemas al momento de analizarlas. Por tal razón es necesario aplicar algoritmos [22] de corrección como los que fueron utilizados en esta investigación. A continuación se expone un ejemplo de la limpieza de datos.

Ejemplo: "Lucy es corrupt@ y tiene ….siempre….será…asi..:) ".

Comentarios como el que se acaba de ejemplificar son muy comunes en los usuarios. Debido a esto, se deben hacer limpieza de datos. En este caso, para el análisis del comentario es necesario quitar los puntos, paréntesis, y reemplazar las tildes para que los algoritmos puedan trabajar correctamente.

H. *Análisis de los datos en R*

Esta herramienta integra algunos algoritmos para análisis de texto, tales como el de Levenshtein cuyo nombre de paquete es "stringdist". Este paquete tiene muchos algoritmos entre ellos el "stringsim" de la misma familia. Este último algoritmo, retorna la probabilidad de similitud entre dos palabras. Los parámetros de "stringdist" se describen a continuación. La Tabla 4 muestra su forma de uso:

- Palabra objetivo: es la palabra ideal que se desea encontrar, en nuestro caso proviene de nuestro diccionario.
- Palabra origen: son los comentarios localizados en nuestra base de datos.
- Método: el método utilizado es el de "jw", el cual brinda un mejor rendimiento en comparación respecto a los otros.
- Probabilidad de acierto: es la probabilidad de acierto de similitud entre dos palabras. Entre más cercano este a cero mucho mejor será la comparación pero para eso deben ser extremadamente idénticas por tal razón, se mantiene un valor prudente para nuestro propósito.

TABLA V. USO DE LA FUNCIÓN "STRINGSIM"

```
prob <- stringsim ('ALTO','ALTAR' ,method = 'jw',p=0.08);
```

palabra character vary	peso numeric(4,2)	nivel charact	mensaje_analisis text
corrupto	0.50	M	vivimos tiempo donde
ignorante	0.50	M	periodismo pais e
miserable	0.70	M	creer estas almas
estupido	0.70	M	jajajajaja causa risa
parido	0.90	A	presidente difer
borrego	0.30	B	presidente difer
perro	0.50	M	jorge guaman has borra
ladron	0.50	M	igualitos lado lado
ladron	0.50	M	igualitos lado lado
corrupto	0.50	M	hablando feriados ban
meco	0.40	M	presidente rafael
culiar	0.80	A	presidente rafael
conche	0.80	A	conoce historia rec
ano	0.50	M	conoce historia rec
corrupto	0.50	M	cerrara esa corru
huevada	0.50	M	pana tantas huevadas
pedo	0.20	B	cuente asambleita
ano	0.50	M	sale gran interce
cojo	0.30	B	sale gran interce
corrupto	0.50	M	gente mayoria
mediocre	0.60	M	xq aca
pene	0.30	B	ecorae traduce nekas
pedo	0.20	B	suena perdio
tonto	0.30	B	justamente gente us
payaso	0.30	B	cierra ojete payaso

Fig. 2. Resultados del Proceso de Análisis de Agresividad al de las palabras más utilizadas por los usuarios

En la Fig.2, se observa las palabras del diccionario que están inmersas dentro del comentario. Además, se observar el peso de cada palabra y el nivel de agresividad.

En la Fig.3, se muestra el resultado del procesado de texto, que duró aproximadamente 15 minutos para procesar 3762 comentarios, teniendo en cuenta que cada comentario tiene aproximadamente entre 20 a 1200 palabras. El resultado con una probabilidad del 90% de similitud entre las palabras, se encontraron 1318 coincidencias con el diccionario.

IV. Evaluación De Resultados

La muestra utilizada para esta investigación fue de 2791 usuarios en trece publicaciones distintas. Antes de empezar a encontrar la relación de los niveles de agresividad de los usuarios, se realizará una exploración de la información que se ha obtenido, producto del resultado del algoritmo de clasificación de palabras. La Tabla 6 muestra las diez palabras más utilizadas en los comentarios.

TABLA VI. LAS DIEZ PALABRAS MÁS UTILIZADAS

Palabras	Peso	Nivel de agresividad	Frecuencia	Probabilidad
Ano	0.50	M	91	0,069
Asco	0.50	M	85	0,064
Basura	0.50	M	77	0,058
Borrego	0.30	B	68	0,052
Burro	0.50	M	52	0,039
corrupto	0.50	M	42	0,032
Ladrón	0.50	M	39	0,030
Mierda	0.80	A	37	0,028
Puta	1.00	A	35	0,027
Rata	0.70	M	33	0,025

Al realizar la primera consulta a la base de datos, sobre las 1318 incidencias que hubieron, se encontró que 91 de las 1318 se refieren a la palabra “ano”, lo que significa que el 0,069% de las personas que han participado en esas publicaciones tienen esa palabra en su vocabulario.

En segundo lugar, le sigue la palabra “asco” sinónimo de repugnancia hacia algo, la probabilidad de estar presentes en sus vocabularios es del 0,064%. Además, se observa que el nivel de agresividad de las top 10, en su mayoría es media y le sigue la alta, lo que indica que los usuarios tienen un comportamiento explosivo inicial y durante el intercambio de opiniones.

Si se realiza una sumatoria de la probabilidad se tiene que el 42,41% de los usuarios que han publicado utilizan cualquiera de estas palabras durante sus conversaciones. La Fig.3, muestra la división porcentual de las palabras más utilizadas por los usuarios.

Fig. 3. División porcentual de las palabras más utilizadas por los usuarios

- Frecuencia del nivel de agresividad

TABLA VII. Frecuencia del nivel de agresividad en los individuos de investigación

Nivel de agresividad	Frecuencia	Porcentaje
Mediano	860	0,65
Alta	243	0,18
Baja	215	0,16

Al observar la Tabla 7, se verifica que el nivel de agresividad más frecuente en los individuos investigados es el mediano. Es decir, tienen un comportamiento explosivo, al inicio y durante el intercambio de ideas y en sus vocabularios están inmersas las palabras de la Tabla 6.

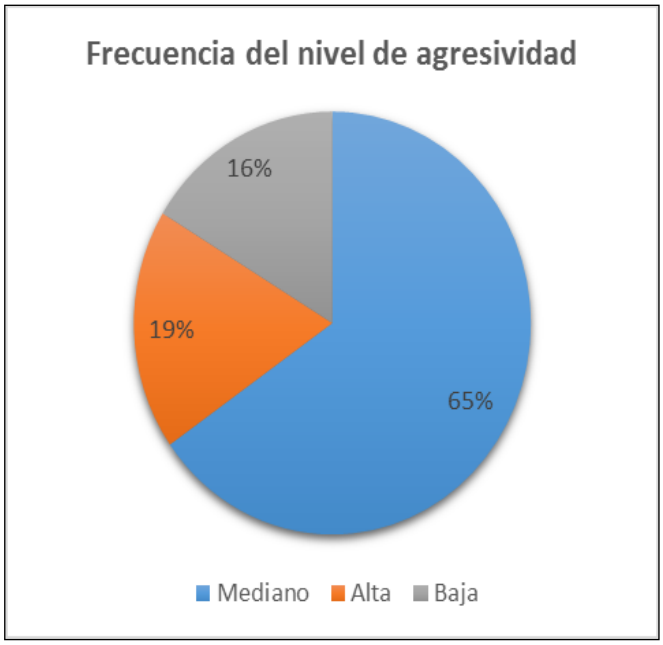


Fig. 4. Frecuencia de niveles de agresividad

Sin embargo, lo más preocupante, es el segundo lugar de los niveles de agresividad, porque en su vocabulario hay palabras agresivas consideradas las más ofensivas de todas. Por lo tanto, se puede inferir que el 18% de los individuos investigados son altamente ofensivos.

Además, en la Fig.4, se observa que un 16% de los individuos investigados poco o casi nada agresivos con otras personas.

- Top 10 de los individuos con nivel de agresividad alta.

La Tabla 8, lista los individuos con nivel de agresividad alta y también, lista los números de insultos realizados hacia otra persona. Para poder identificar los individuos, se realizó una consulta a la base de datos de los 10 primeros individuos con insultos graves ordenados descendentemente y como resultado tenemos ese listado. Pero si se desea saber más, sobre las palabras más utilizadas por estas personas, ver la Fig.5.

TABLA VIII. TOP TEN DE INDIVIDUOS CON NIVEL DE AGRESIVIDAD ALTA

Nombre	N° Insultos	Nivel
Alfredo Martínez	10	A
David Villacrés	4	A
Derek Córdova	3	A
Diego Galarraga Villalba	6	A
Edgar Ayovi	5	A
Fernando Juan Guamán	4	A
Jorge Luis Charvet Castro	6	A
Kira Ryūzaki	6	A
Macías Vélez Andrius	4	A
Tardo Estebas	4	A

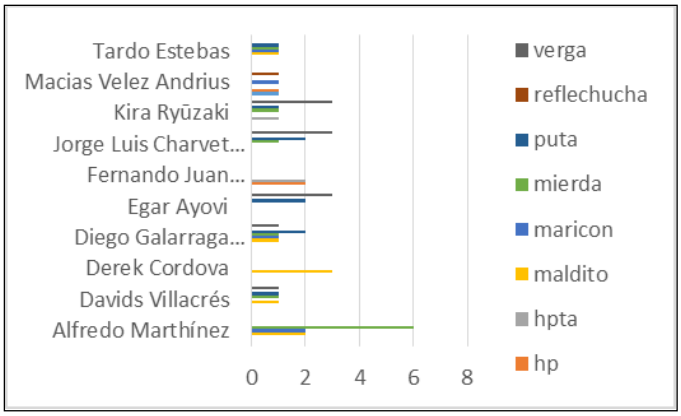


Fig. 5. Individuos agresivos vs palabras comunes

V. TRABAJOS RELACIONADOS

En la investigación de WebCrowling [28] se evidencias técnicas y modelos matemáticos para extracción de datos en sitios web con el objetivo de analizar la información y tener una percepción del contenido. A esto se le conoce como patrones de opinión que se utilizan en el marketing para ofrecer productos a consumidores. También, existen otras investigaciones como análisis sentimental [29], que consiste en analizar los comentarios de las personas para determinar la parte subjetiva de los individuos, cuyos fines son la obtención de perfiles de consumidores.

VI. CONCLUSIONES Y TRABAJOS FUTUROS

Como conclusiones generales cabe establecer que al explorar la información, después del proceso de clasificación, se encontró con niveles altísimos de insultos en gran mayoría de los comentarios, tal es el caso de la Tabla 6, donde se demuestra que existen un 42,41% de personas que utilizan palabras ofensivas para expresarse. Además, la gran mayoría de las palabras son agresivas, para ser específicos un 65% usa palabras de nivel de agresividad mediano mientras que otra parte conformando por el 18% utiliza frase con nivel de agresividad alto y tan solo el 16% de los individuos no son tan violentos.

Este trabajo de investigación demuestra que de 2791 usuarios, el 65% de ellos tiene un vocabulario medio agresivo, el 18% un vocabulario agresivo y el 16% poco agresivos. Por lo tanto, se puede inferir que el 65% de las personas que publica un mensaje en fan pages de política, son medianamente agresivos y tiene un comportamiento medio violento.

Como trabajo futuro se planea implementar una herramienta en tiempo real, que interactúe directamente con la red social Facebook y mediante una "app" embebida en Facebook, determine si es adecuado o no aceptar la solicitud de un desconocido. Además, las aplicaciones de este tipo de minería de texto tienen una gran utilidad a nivel de marketing porque se podría obtener perfiles de consumidores.

REFERENCIAS

[12] C. Domínguez,. "Las Redes Sociales. Tipología, uso y consumo de las redes 2.0 en la sociedad digital actual." Documentación de las Ciencias de la Información 33 (2010): 45-68 [online] http://revistas.ucm.es/index.php/DCIN/article/view/DCIN1010110045A/18656.

[13] Phillips, Whitney. "LOLing at tragedy: Facebook trolls, memorial pages and resistance to grief online." First Monday 16.12 (2011).

[14] Levenshtein VI (1966). "Binary codes capable of correcting deletions, insertions, and reversals".

[15] Fernando Juárez, Alba Dueñas & Yamilé Méndez, 17-05-2005, ISSN 1697-2600, [online] http://www.redalyc.org/pdf/337/33760108.pdf

[16] Mauricio Batallas Bustamante, "Agresividad, Hostilidad e Ira en adolescentes que juegan video juegos." ,2014:19-24 [online] http://dspace.udla.edu.ec/bitstream/33000/3441/1/UDLA-EC-TPC-2014-04(S).pdf

[17] K. Cela, W. Fuertes, C. Alonso, F. Sánchez, , Revista Estilos de Aprendizaje, nº5, Vol. 3, 04-2010, [online] http://learningstyles.uvu.edu/index.php/jls/article/view/123/86.

[18] M. Gonzalez, "Patrones de Comportamiento", 28-09-2010, ISBN 978-84-9717-323-0, [online] http://www.elcampamentodedios.com/28sep10b.pdf .

[19] M. Montes, M. Gómez, A. Gelbukh, and A. López López. "Minería de texto empleando la semejanza entre estructuras semánticas." Computación y Sistemas 9.1 (2005): 63-81.

[20] Aggarwal, Charu C., and ChengXiang Zhai. Mining text data. Springer Science & Business Media, 2012.

[21] Elder IV, John, and Thomas Hill. Practical text mining and statistical analysis for non-structured text data applications. Academic Press, 2012.

[22] Buneman, Peter, and Robert E. Frankel. "FQL: a functional query language." Proceedings of the 1979 ACM SIGMOD international conference on Management of data. ACM, 1979.

[23] "Diccionario de palabras ofensivas",2016. [online] https://1drv.ms/f/s!AvddUrAdljaYgmMi5eCx34HydZh0

[24] Wilfredo Martel, "Algoritmo de Ordenación y Clasificación de palabras ofensivas", 2016. [online] https://1drv.ms/f/s!AvddUrAdljaYgmbDEokfXJZmr4hq

[25] "Resultados de la investigación", 2016. [online] https://1drv.ms/x/s!AvddUrAdljaYgmKnrf_ZNcGS1e76

[26] "Algoritmo de limpieza de datos", 2016. [online] https://1drv.ms/f/s!AvddUrAdljaYgmh9ie20r4RPjL34

[27] Wilfredo Martel, "Video del proceso de extracción y análisis de datos", 2015.[online] https://1drv.ms/f/s!AvddUrAdljaYgmowCe7RzxynL2vp

[28] http://www.palermo.edu/ingenieria/pdf2013/12/12CyT_01webcrawling.pdf.

[29] J,Akaichi, Z. Dhouioui, M. J. Lopez-Huertas Perez. "Text mining Facebook status updates for sentiment classification,", System Theory, Control and Computing (ICSTCC), 2013 17th International Conference, 10- 2013 ,ISBN 978-1-4799-2227-7.

Prácticas de Ingeniería de Requisitos en las Empresas de Desarrollo de Software, en la Ciudad de Quito - Ecuador

Javier Simbaña Saransig, Gabriel Simbaña Quinsasamin, Cecilia Hinojosa Raza, Mario Ron Egas
Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas- ESPE, Sangolquí, Ecuador

E-mail: simbana.javier@gmail.com, jgabriels_87@hotmail.com, {cmhinojosa, mbron}@espe.edu.ec

***Resumen*—Antecedentes: Uno de los elementos clave para mejorar la industria de software en el Ecuador es contar con un diagnóstico de las prácticas que utilizan las empresas en el proceso de desarrollo y específicamente, en la Ingeniería de Requisitos (IR). Este diagnóstico permitirá orientar acciones concretas de mejora, en cuanto a la calidad del software. Objetivo: El presente trabajo se planteó como objetivo realizar un estudio exploratorio de las prácticas de la IRs que aplican las empresas de desarrollo de software de la ciudad de Quito - Ecuador. Metodología: El estudio se realizó en 51 empresas elegidas de forma aleatoria. Mediante una encuesta guiada, diseñada para utilizar estadística descriptiva. Resultados: Los datos ponen de manifiesto el estado actual de la IR en un sector representativo de la industria del software de Quito. Conclusiones: En la presente investigación se detectaron debilidades en la aplicación de la IR en las empresas de desarrollo de software, radicadas en la ciudad de Quito, ante lo cual se propone como alternativa crear sinergia entre la universidad y la empresa para emprender proyectos orientados a la solución de los problemas detectados**

Palabras Clave— Prácticas de Ingeniería de Requisitos, Investigación exploratoria, Empresas de desarrollo de software, Quito – Ecuador

I. INTRODUCCIÓN

El sector de desarrollo de software ha sido reconocido por el gobierno ecuatoriano como un eje estratégico, el cual aporta al cambio de la matriz productiva del país [1]. A través del Ministerio de Industrias y Productividad, se busca fortalecer la cadena de valor del sector software, promocionar e insertar los productos ecuatorianos en mercados internacionales. Para fortalecer el sector, es preciso contar con un conocimiento más profundo sobre las prácticas que aplican las empresas de desarrollo de software, en general; y sobre las prácticas de la Ingeniería de Requisitos (IR), en particular.

Ante la importancia que reviste la IR en la competitividad de las empresas de desarrollo de software, y la falta de estudios referentes a este ámbito en el Ecuador, se pone a consideración esta investigación, la cual presenta una aproximación del estado de la IR en el sector de desarrollo de software en Quito. Se espera que los datos resultantes de este estudio contribuyan al planteamiento de acciones que favorezcan el desarrollo eficiente de este sector estratégico del país.

El artículo está estructurado de la siguiente forma: en la sección 2 se detallan los antecedentes sobre la industria de software en el Ecuador, la importancia de la Ingeniería de Requisitos y el proceso que sirvió de base para esta investigación. La sección 3 describe el diseño de la investigación. En la sección 4 se presentan los resultados obtenidos. Y finalmente, en la sección 5 se presentan las conclusiones y trabajos futuros.

II. ANTECEDENTES

A. La industria de software en el Ecuador

En la ciudad de Quito - Ecuador, ochenta empresas se dedican exclusivamente al desarrollo de software [2] y corresponden al 49% de la industria ecuatoriana de software. En Guayaquil se encuentran radicadas el 37%. Se trata de una industria relativamente nueva, conformada en su mayor parte por microempresas. Empezó su crecimiento en la década de los noventa y ha presentado una importante evolución en los últimos años, con un tasa de crecimiento anual del 22.4% [3]. Según el Reporte Global de Competitividad del Foro Económico Mundial, el país ha ascendido del puesto 106 al 101 en cuanto al indicador Absorción Empresarial de la Tecnología, entre el 2010 y el 2013 [4]. La mayoría de estas firmas incursionan en el desarrollo de soluciones para pequeñas y medianas empresas. Su producción se enfoca de manera importante en software de gestión, tales como: ERP, CRM, ERM, gestión de logística y aplicaciones móviles.

B. La Ingeniería de Requisitos

La Ingeniería de Requisitos es un proceso cooperativo, iterativo e incremental, [5] en el cual se descubren, analizan, documentan, comunican, validan y gestionan [6] las características o restricciones operativas y funcionales que se esperan del sistema, las cuales deben ser: documentadas, completas y acordadas entre los involucrados [7]; de tal manera que sean la base para las posteriores fases del desarrollo del

sistema.

La Ingeniería de Requisitos es la fase más importante y difícil del proceso de desarrollo de software [8] [9] [10], es la base para la planificación, diseño, implementación y pruebas del software. Los requisitos deficientes son la principal causa del fracaso de proyectos, al respecto, la consultora internacional Standish Group [11], cita que tan solo el 32% de los proyectos de desarrollo de software, se pueden considerar exitosos, el 44% se entregaron fuera de plazo, excedieron su presupuesto y no cubrieron la totalidad de las características y funcionalidad pactada, y el 24% de los proyectos fueron cancelados. En este mismo estudio se puntualiza que el principal factor para el fracaso de un proyecto de desarrollo de software radica en la mala calidad de los requerimientos.

Jones [12] luego de analizar cientos de organizaciones, llegó a determinar que el proceso de ingeniería de requisitos es deficiente en más del 75 por ciento de las mismas. Si bien este estudio fue realizado hace dos décadas, es importante analizar cuánto han avanzado las organizaciones en este sentido.

C. Proceso de la Ingeniería de Requisitos

Para obtener requisitos de calidad, es preciso seguir un proceso bien definido, así también utilizar en cada fase las técnicas que aporten a la calidad de los resultados y a la eficiencia del proceso; al respecto, el "Software Engineering Body of Knowledge – SWEBOK" [13] determina las fases genéricas del proceso de IR, las cuales son: Elicitación, Análisis, Especificación y Validación. Estas fases siguen una secuencia lógica y se ejecutan de manera iterativa.

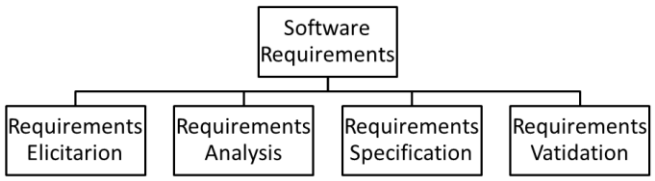


Figure 1. Fases del proceso de Requisitos del Software (IEEE Computer Society, 2014)

III. DISEÑO DE LA INVESTIGACIÓN

A. Método de investigación

El presente estudio es de carácter exploratorio, su objetivo es determinar las prácticas de la IR que aplican las empresas de desarrollo de software, en la ciudad de Quito. Las unidades de análisis para el presente estudio fueron las empresas de desarrollo de software, representadas por el líder de proyectos, el gerente de desarrollo o el desarrollador encargado de las actividades de la ingeniería de requisitos. En todos los casos hubo un solo representante por organización.

Hipótesis: Las empresas de desarrollo de software de la ciudad de Quito, no aplican sistemáticamente los lineamientos de la Ingeniería de Requisitos.

Población: Para el estudio la población estuvo conformada por las empresas y organizaciones dedicadas al desarrollo de software en la ciudad de Quito, que ascendieron a ochenta [2].

Muestra: La muestra se determinó aplicando el método de muestreo aleatorio simple. En la investigación participaron 51 empresas de desarrollo de software, radicadas en la ciudad de Quito; considerando que en esta ciudad se concentra aproximadamente el 49% de las empresas de este estratégico sector [15]. El tamaño de la muestra se estableció utilizando la fórmula (1) del método estadístico para una población finita.

$$n = \frac{Z^2 * P * Q * N}{(N-1) * E^2 + Z^2 * P * Q}$$

Dónde:

N = número de la población = 80

E = margen de error = 5%

Z = nivel de confianza = 1.65 (90%)

P = probabilidad de éxito = 15%

Q = probabilidad de fracaso = 85%

n = tamaño de la muestra = 51.

B. Elaboración del instrumento de investigación

Para la extracción de la información, se estructuró un cuestionario con preguntas abiertas, cerradas y de opción múltiple. Las preguntas estuvieron enfocadas a los siguientes tópicos:

- Tiempo de vida de la organización.
- Certificaciones que ha obtenido.
- Tamaño de los equipos de desarrollo.
- Número de proyectos de desarrollo.
- El proceso de ingeniería de requisitos aplicado por la empresa.
- Las técnicas utilizadas en cada una de las fases de la IR.
- Problemas detectados en su organización en el ámbito de la IR.

C. Recolección y análisis de datos

Los datos fueron recolectados con métodos directos o de primer nivel, los investigadores estuvieron en contacto directo con los líderes de proyectos de las empresas que conformaron la muestra. Se aplicó una entrevista semiestructurada, la misma que tuvo una duración aproximada de 45 minutos, en cada empresa; el objetivo fue conocer las técnicas que aplican en el proceso de la IR. Con el fin de identificar patrones o relaciones entre los datos, se clasificaron las empresas en: pequeñas, medianas y grandes y el análisis de datos fue mayoritariamente cuantitativo.

IV. Evaluación De Resultados

A. Empresas encuestadas

La actividad principal de las empresas que conformaron la muestra, es el desarrollo de software. De las 51 organizaciones, 10 tienen una antigüedad menor a 5 años, 28 se encuentran en el rango de 5 - 15 años y 13 organizaciones están en el mercado por más de 15 años. El tamaño de las áreas de desarrollo y mantenimiento de software de las organizaciones, se midió en base al número de personas que conformaban los equipos de desarrollo. De las 51 organizaciones, 17 tenían equipos con menos de 6 personas, los equipos de 21 organizaciones estaban en el rango de 6 – 20 personas y 13 organizaciones con más 20 personas en el área.

B. Proyectos de software desarrollados en los 2 últimos años

La investigación arrojó que el 68.98% de proyectos desarrollados tuvo una duración menor a un mes, estos proyectos mayoritariamente correspondieron a mantenimiento de software (corrección de errores), el 14.38% proyectos tuvieron una duración en el rango de 1 - 6 meses y el 16.64% corresponde a proyectos cuya duración fue mayor a 6 meses. Los datos obtenidos dan indicios de que los altos niveles de mantenimiento correctivo probablemente se deben a que el proceso de Ingeniería de Requisitos no es lo suficiente robusto.

C. Personal especializado en ingeniería de requisitos

Según lo respondido por las empresas encuestadas se obtuvo que el 70.6% no tiene personal especializado en IR, tan solo el 29.4% cuenta con personal capacitado específicamente en esta área. Al analizar los resultados en relación al tamaño y el personal especializado, se encontró que el 82,35% de las organizaciones pequeñas tienen un mayor porcentaje de personal no especializado.

Esta situación podría considerarse una debilidad de la industria de desarrollo de software de la ciudad de Quito, acentuándose en las empresas pequeñas. El detalle de los resultados, se puede apreciar en la Tabla I.

TABLA III. Personal especializado en ingeniería de requisitos

	Posee personal especializado en IR			
Tamaño empresa	SI	Porcentaje	NO	Porcentaje
De 1 a 5 Empleados	3	17,65%	14	82,35%
De 5 a 6 empleados	8	38,10%	13	61,90%
> 20 empleados	4	30,77%	9	69,23%
Total	15	29,40%	36	70,60%

D. Certificación CMMI

A fin de obtener información orientadora sobre el grado de madurez del proceso de desarrollo de software, se consultó a las empresas si contaban con certificación CMMI. Los resultados arrojaron que el 98.03% no cuenta con esta certificación, tan solo el 1.97% tiene certificación, como se aprecia en la Tabla II. Cabe indicar que la única empresa que cuenta con este tipo de certificación, es una empresa internacional.

TABLA IV. Número de empresas con certificación CMMI

Tamaño empresa	**SI**	**Porcentaje**	**NO**	**Porcentaje**
De 1 a 5 Empleados	0	0.00%	17	33,33%
De 5 a 6 empleados	0	0.00%	21	41,18%
Más de 20 empleados	1	1.97%	12	23,53%
Total	1	1.97%	50	98.03%

Este aspecto reviste importancia debido a que la calidad es fundamental para el sector, especialmente en el Ecuador cuya industria está conformada principalmente por pequeñas y medianas empresas. A nivel mundial, las organizaciones del ramo orientan sus esfuerzos a mejorar sus procesos y realizar productos de calidad, que les garanticen la preferencia y permanencia en el mercado. La ausencia de este tipo de certificaciones podría ser otro elemento que dificulta el fortalecimiento del sector software en el país.

E. Certificación ISO para el proceso de desarrollo de software

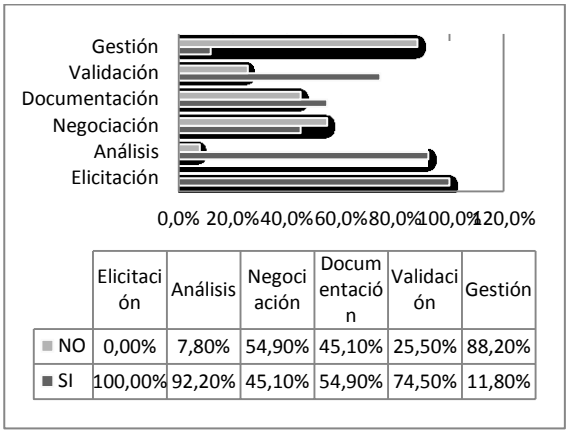
Según lo respondido por las empresas encuestadas se obtuvo que el 9.8% de las organizaciones tienen certificación ISO para el proceso de desarrollo de software, el detalle se presenta en la Tabla III.

TABLA V. Número de empresas con certificación ISO

Tamaño empresa	**NO**	**Porcentaje**	**SI**	**Porcentaje**
De 1 a 5 Empleados	17	33,33%	0	0.00%
De 5 a 6 empleados	17	33,33%	4	7,84%
Más de 20 empleados	12	23,53%	1	1,96%
Total	46	90.20%	5	9.80%

F. Fases del proceso de ingeniería de requisitos utilizadas en la industria ecuatoriana

Para obtener información sobre las fases del proceso que aplican las empresas, los encuestados describieron su proceso de IR, mientras los investigadores documentaron mediante diagramas y notas. Luego se contrastó con una pregunta cerrada, en la que se les solicitó indiquen las fases que aplican, la cual constaba de 6 ítems correspondientes a cada fase del proceso de IR, mismos que permitían respuestas múltiples. Los resultados obtenidos en la pregunta cerrada se presentan en la Fig. 2.



	Elicitación	Análisis	Negociación	Documentación	Validación	Gestión
NO	0,00%	7,80%	54,90%	45,10%	25,50%	88,20%
SI	100,00%	92,20%	45,10%	54,90%	74,50%	11,80%

Figure 2. Respuestas sobre la aplicación del proceso de ingeniería de requisitos

De los resultados obtenidos se desprende que la mayoría de empresas no siguen un proceso sistemático de la IR que garantice la calidad de los requisitos. Llama la atención el bajo porcentaje de empresas que cumplen con las fases de gestión y documentación. En el 88,20% de las organizaciones los requisitos no se gestionan. Así también llama la atención que el 45,10% de los encuestados, manifestaron que no atienden la fase de documentación, factor que impactaría negativamente en la planificación y gestión del proyecto. Finalmente, 45,10% de los encuestados indicaron que aplican la fase de negociación, lo que puede ser un indicador de que los requisitos, mayoritariamente no son consensuados entre los involucrados. Este constituye otro elemento, que no favorece a la calidad de los requisitos obtenidos.

G. *Técnicas más utilizadas en cada fase del proceso de ingeniería de requisitos, en la industria local*

Las técnicas que utilizan las empresas en las diferentes fases del proceso de la IR, se resume en la Tabla IV.

TABLA VI. TÉCNICAS MÁS UTILIZADAS PARA CADA FASE DE REQUISITOS EN LA INDUSTRIA LOCAL

Fase del proceso	**Nombre de la técnica**	**%**
Elicitación	Entrevista	32,30%
	Grupos de trabajo	24,20%
	Estudio de sistemas existentes	22,60%
Análisis	Casos de uso	40,50%
	Escenarios	26,20%
	Modelo de clases	21,40%
Negociación	Negociación win - win	47,10%
	Matriz de interacción	41,20%
	Otras	11,80%
Documentación	Lenguaje natural	71,40%
	Otros	28,26%
Validación	Prototipos	35,20%
	Control documental	24,10%
	Inspecciones	20,40%
Gestión	Un solo criterio	63.60%
	Clasificación y top-ten	22,70%

V. VALIDEZ Y LIMITACIONES DEL ESTUDIO

A. *Validez interna*

En la presente investigación se ha utilizado una serie de estrategias, para fomentar la validez interna de la investigación. Se aplicó un muestreo aleatorio simple, a un listado de las empresas desarrolladoras de software. Se empleó una encuesta guiada, con el objeto de clarificar cualquier duda que pudiera surgir en el encuestado y que el estudio proporcione respuestas significativas y coherentes.

La elección de las empresas fue aleatoria, sin embargo, algunas de las empresas seleccionadas, en primera instancia, se negaron a participar en el estudio, por lo que se procedió a un nuevo sorteo para completar la muestra, para no restarle validez al estudio. Esta situación puede introducir un sesgo, ya que una de las posibilidades de no haber aceptado participar en el estudio, puede ser la inseguridad que tienen sobre la forma como ejecutan su proceso de IR.

Por otro lado, la validez interna de la investigación, pudo haberse fortalecido si se hubiera considerado una categorización específica de las empresas y de los proyectos, lo cual hubiera permitido obtener información más detallada para el diagnóstico. Este tema puede ser abordado en trabajos futuros.

B. *Validez externa*

El presente estudio es de carácter exploratorio, por lo tanto, las conclusiones no deberían ser comprendidas como generalizaciones más allá del ámbito de estudio. Los resultados obtenidos deberían ser considerados como hipótesis que deben ser validadas más a fondo.

VI. TRABAJOS RELACIONADOS

En cuanto a estudios relacionados a la investigación de las prácticas de ingeniería de requisitos se encontró un trabajo realizado por miembros de la IEEE, en Malasia llamado "Investigation into Requirements Management Practices in the Malaysian Software Industry" [14] este estudio se realizó para tener un conocimiento de las prácticas de ingeniería de requisitos en dicho país, a nivel de CMM 2. Fue un estudio realizado a diferentes empresas por medio de un cuestionario para la obtención de los datos. Con los resultados obtenidos en la investigación se indica que la industria de ese país carece de buenas prácticas en la gestión de requisitos.

A demás se encontró otro trabajo realizado por la Facultad de Ingeniería y Ciencias Exactas, UADE, Ciudad Autónoma de Buenos Aires, llamado "Prácticas de Ingeniería de Requerimientos en el desarrollo de aplicaciones Web" este estudio se realizó en diferentes empresas de Argentina, específicamente en IR para aplicaciones web, dicho estudio expresa que "los resultados en términos de cronograma y presupuesto, permiten deducir una cierta debilidad en las metodologías de estimación. Esta debilidad es previsible pues los procesos de estimación no han logrado estabilizarse en las aplicaciones convencionales (el arte "negro"), menos aún en las aplicaciones Web que agregan grados de complejidad al problema.". [15]

VII. CONCLUSIONES Y TRABAJO FUTURO

La presente investigación provee un diagnóstico inicial sobre las prácticas que aplican las organizaciones radicadas en la ciudad de Quito – Ecuador, en el proceso de la IR. Para esta primera aproximación, se optó por el método basado en encuestas guiadas, lo cual permitió obtener información relevante del sector, mas queda abierta la posibilidad de profundizar el estudio de los tópicos abordados, utilizando otras herramientas de investigación. Por ejemplo, es necesario determinar si el alto porcentaje de proyectos orientados al mantenimiento correctivo se debe a la mala calidad de los requisitos.

La industria ecuatoriana de desarrollo de software, es una industria joven, conformada, en un alto porcentaje, por micro y pequeñas empresas, el 74% cuenta con equipos menores a 20 personas. Solo el 30% de organizaciones tiene personal con formación específica en IR. Al parecer, la falta de personal capacitado explicaría el bajo porcentaje de empresas que cumplen con el ciclo del proceso de IR.

Ante el escenario descrito, surge la necesidad de crear sinergia entre la universidad y la empresa para el desarrollo de proyectos orientados a la solución de los problemas detectados. Es necesario generar proyectos de capacitación para el personal que es responsable de realizar las actividades de: recolección, análisis, especificación, validación y gestión de requisitos. Así también es necesario, diseñar soluciones que permitan la implementación de las buenas prácticas de la IR en las empresas del sector, acorde a sus características particulares.

REFERENCIAS

[1] MIPRO. (2013, Noviembre) MIPRO suscribió convenio con AESOFT. Video.

[2] Superintendencia de Compañías del Ecuador. (2014, Mar.) Consulta de Compañías. [En línea]. http://www.supercias.gob.ec/: http://www.supercias.gob.ec/portalinformacion/index.php?archive=portaldeinformacion/consultadirectorioparametro.zul

[3] AESOFT. (2012, Junio) AESOFT. [En línea]. http://www.aesoft.com.ec/www/index.php/118-slideshow/154-http-www-slideshare-net-aesoft-ot-20489-microsoftfolleto

[4] Conferencia de las Naciones Unidas sobre Comercio y Desarrollo. (2012, Noviembre) unctad. [En línea]. http://unctad.org/es/paginas/PressRelease.aspx?OriginalVersionID=109

[5] Ortas, Aproximacion a la Ingenieria de Requerimientos. Uruguay: Universidad ORT, 2001.

[6] IEEE Std 610, "Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries," New York, 1990.

[7] Klaus Phol, Requirements Engineering Fundamentals, Principles, and Techniques. Berlin: Springer, 2010.

[8] Carlos Zapata, "Una propuesta de metaontología para la educción de requisitos," vol. 18, no. 1, 2009.

[9] D. Pandey, U. Suman, and A.K. Ramani, "An Effective Requirement Engineering Process Model for Software Development and Requirements Management," vol. 1, no. 978-1-4244-8093-7, 2010.

[10] Hubert Hofman and Franz Lehner, "Requirements Engineering as a Success Factor in Software Projects," IEEE Software, p. 58.66, 2001.

[11] The Standish Group. (2009, Apr.) Standish Group report. [En línea]. http://www1.standishgroup.com/newsroom/chaos_2009.php

[12] Capers Jones, Applied software measurement: assuring productivity and quality. New York: McGraw Hill, 1996.

[13] IEEE Computer Society, Guide to the Software Engineering Body of Knowledge. Québec: IEEE, 2014.

[14] A. Zainol. (2008, octubre) IEEE. [En línea]. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4722056&searchWithin%3Dp_Authors%3A.QT.Zainol%2C+%2FA%2F.QT.%26searchWithin%3Dp_Author_Ids%3A37658732400

[15] Alejandro Oliveros, Fernando J. Danyans, and Matías L. Mastropietro. (2013, octubre) Facultad de Ingeniería y Ciencias Exactas, UADE, Ciudad Autónoma de Buenos Aires. [En línea]. http://wer.inf.puc-rio.br/WERpapers/pdf_counter.lua?wer=WER14&file_name=paper9.pdf

Detección y mitigación de ataques ARP Spoof empleando entornos virtualizados

Marcia Cordero, Myriam Viñamagua y Carlos Garzón
Departamento de Ciencias de la Computación, Programa de Maestría en Gerencia de Sistemas, Universidad de las Fuerzas Armadas –ESPE, Sangolquí, Ecuador
marciveth@gmail.com, myriamv83@gmail.com, carloscfga@hotmail.com

***Abstract*—Los ataques ARP Spoof tienen como finalidad infiltrarse en una red mediante el envío de ARP (Protocolo de Resolución de Direcciones) falsos al bus Ethernet para explorar paquetes de datos, alterar el tráfico o incluso detenerlo. La presente investigación se enfoca en la evaluación del ataque ARP Spoof (sniffer), utilizando como plataforma de experimentación un entorno virtual de red que permite identificar cómo actúa dicho ataque cuando un cliente accede a páginas Web publicadas en un servidor; para llevar a cabo la investigación se diseñó e implementó una red híbrida con delimitación WAN, LAN y DMZ (Zona Desmilitarizada). La herramienta evaluada fue BetterCAP la cual se especializa en este tipo de ataques. Para validar está investigación se desarrolló un mecanismo de detección y mitigación de los ataques a nivel de Shell script. Finalmente, se evaluó el número de paquetes de entrada al atacante cuando éste actúa en modo sniffer.**

Palabras clave: Arp Spoof, BetterCAP, GNS3, VMWare, Virtualización

I. INTRODUCCIÓN

Un ataque de ARP Spoof tiene como objetivo infiltrarse en una red mediante el envió de ARP falsos al bus Ethernet con la finalidad de asociar la MAC del atacante con la dirección IP de otro nodo de confianza como por ejemplo el Gateway para explorar el paquete de datos, alterar o detener el tráfico. En la presente investigación se seleccionó el ataque ARP Spoof, que envía ataques ARP falsos al bus Ethernet y provoca que el atacante husmee el tráfico de red de la máquina atacada con el propósito de obtener información sensible, como por ejemplo nombres de usuario, contraseñas, cookies, mensajes de correo, mensajería instantánea, conversaciones VoIP, etc. El uso de registros ARP estáticos permite mitigar los ataques ARP Spoof.

En este contexto, la comunidad científica ha realizado investigaciones para mitigar los ataques a redes utilizando las tecnologías de virtualización de acuerdo con el monitoreo e identificación de ataques de redes [1]. Bajo esta guía, el modelo propuesto por F. J. Díaz Jiménez y J.G. Palacio Velásquez [2], indica claramente la disección de un ataque MITM (Man in the Middle) mediante ARPSpoofing y técnicas de protección. Adicionalmente los investigadores Melgar Jara [3] y Cazar Jácome, D. A. [4] desarrollan aplicaciones para detección de ataques en redes IPv4/IPv6 y en específico análisis de IP Spoofing. En [5] Herrera Figueroa y Helmuth Lenin los investigadores realizan un ataque a redes IP en un entorno corporativo real. En este mismo ámbito [6], [7], han utilizado sistemas virtualizados y estudios comparativos de sistemas de virtualización y de seguridad. Otros investigadores Echeverry Parada, J. S. [8], [9], [10] utilizaron metodologías para el diagnóstico continuo de las redes informáticas, análisis forenses a paquetes de datos todo esto en redes LAN de instituciones públicas y privadas.

El presente trabajo se enfoca en la evaluación del ataque ARP Spoof, utilizando como plataforma de experimentación un entorno virtual de red que permita identificar cómo actúa dicho ataque y cuál sería su impacto. Para llevarlo a cabo se diseñó e implementó una red con delimitación WAN, LAN y DMZ, con el fin de que el experimento sea lo más cercano a un entorno real en la topología se utilizó el Firewall Cisco 5520 sobre GNS3 el mismo que además de realizar funciones propias de un cortafuegos actúa como enrutador de los segmentos de red con ello los equipos clientes tienen la posibilidad de acceder tanto a servidores como salir hacia una red externa como Internet. La herramienta evaluada fue BetterCAP instalada sobre un ambiente Linux. Para validar está investigación se desarrolló un mecanismo de detección y mitigación de los ataques utilizando Shell scripts

Entre las principales contribuciones de esta investigación se tiene: i) la evaluación de ataques ARP Spoof mediante la herramienta BetterCAP para determinar el impacto en la red y ii) creación de scripts de detección y mitigación de este ataque.

El documento ha sido organizado como sigue: el capítulo 2 presenta el fundamento teórico; en el 3 se describe el diseño y configuración del experimento, topología de la red y los scripts de detección y mitigación; en el 4 se presenta la evaluación de resultados y discusión; en el 5 se presentan los trabajos relacionados y en el 6 se establecen las conclusiones y se señala el trabajo futuro.

II. MARCO TEÓRICO

A. GNS3

Es un simulador gráfico de red para diseño de topologías complejas y realización de simulaciones sobre ellas. En esta herramienta libre de simulación se puede cargar cualquier sistema operativo de enrutadores y se puede ejecutar cualquier tipo de configuración simple o compleja como si se estuviese trabajando en un equipo real. [16]

Para permitir completar simulaciones, GNS3 está estrechamente vinculada con: i) Dynamips, un emulador de IOS que permite a los usuarios ejecutar binarios imágenes IOS de Cisco Systems; ii) Dynagen, un front-end basado en texto para Dynamips; iii) Qemu y VirtualBox, para permitir utilizar máquinas virtuales como un firewall PIX; iv) VPCS, un emulador de PC con funciones básicas de networking; v) IOU (IOS on Unix), compilaciones especiales de IOS provistas por Cisco para correr directamente en sistemas UNIX y derivados. [17]

B. ARP Spoof

ARP SPOOF es un método usado para infiltrarse en una red Ethernet conmutada, que permite al atacante explorar paquetes de datos, alterar el tráfico, o incluso detenerlo. Esta

técnica tiene como principio enviar mensajes ARP falsos al bus Ethernet con la finalidad de asociar la MAC del atacante con la dirección IP de otro nodo de confianza como por ejemplo el Gateway. El principio del ARP Spoofing es enviar mensajes ARP falsos (falsificados, o spoofed) a la Ethernet. [18]

El ataque de ARP Spoofing puede ser ejecutado desde una máquina controlada (el atacante ha conseguido previamente hacerse con el control de la misma: intrusión), o bien la máquina del atacante está conectada directamente a la LAN Ethernet.

C. BetterCAP

BetterCAP es una poderosa herramienta para realizar diversos tipos de ataques "Hombre en el medio" contra la red, manipular tráfico HTTP y HTTPS en tiempo real y mucho más. [15]

Analizando las características principales de esta herramienta se puede encontrar con un sniffer desarrollado especialmente para centrarse en el siguiente tipo de tráfico: páginas web visitadas capturando las direcciones URL, páginas web seguras HTTPS que se visitan, los datos POST de las conexiones HTTP, autenticaciones HTTP, recopila los credenciales de las conexiones FTP, recopila los credenciales de las conexiones IRC, captura y recopila los credenciales de las conexiones de correo electrónico POP, IMAP y SMTP, detecta y recopila los credenciales de las conexiones NTLM como HTTP, SMB, LDAP, etc.

III. CONFIGURACIÓN DEL EXPERIMENTO

A. Herramientas

Para la implementación del experimento se ha planteado una arquitectura basada en herramientas de virtualización y de simulación para los equipos de comunicaciones, además de herramientas de software libre las cuales se detallan a continuación:

1) *Sistema de Virtualización:* Como sistema de virtualización se utilizó VMware Workstation 12 Pro [11] sobre Windows 10, con el fin de instalar y configurar tanto clientes como servidores.

2) *Simulador de equipos de comunicaciones:* A fin de disponer de equipos de comunicaciones lo más cercanos a los físicos se utilizó GNS3 [12] que es un software de simulación de red que permite diseñar topologías de red complejas y poner en marcha simulaciones sobre ellos.

3) *Firewall: S*e implementó sobre GNS3 un firewall de marca Cisco, modelo 5520 [13], el cual es diseñado para empresas pequeñas y posee características de: alta disponibilidad, IPSec, SSL VPN y la posibilidad de añadir funciones de prevención de intrusos (IPS) y anti-X. En el mismo se establecieron tres zonas de seguridad (inside, outside y dmz). La conexión SSH fue habilitada para permitir la gestión del equipo.

4) *Web Server:* Como servidor Web se utilizó Apache [14] sobre Centos 6 a fin de brindar páginas Web solicitadas por clientes que utilizan navegadores Web.

5) *Herramienta de inyección de ataques ARP Spoof:* Como herramienta para realizar el ataque se utilizó BetterCAP [15] funcionando sobre Linux. Esta misma herramienta permite capturar el tráfico obtenido cuando se efectúa el ataque con opciones de exportar el tráfico a archivos pcap.

B. Diseño de la topología experimental

A fin que el experimento sea lo más cercano a un entorno real se requirió la creación de una infraestructura de red con los elementos y esquemas base que se puede encontrar en cualquier red de un entorno de producción. Por lo que, para la implementación de la topología se utilizó el Firewall Cisco 5520 sobre GNS3 el mismo que además de realizar funciones propias de un cortafuegos actúa como enrutador de los segmentos de red (inside, outside y dmz), con ello los equipos clientes tienen la posibilidad de acceder tanto a servidores como salir hacia una red externa como Internet, un computador con Windows 10 con el correspondiente hipervisor que permitió la creación de las máquinas virtuales tanto cliente, cliente-atacante y servidor Web. El esquema de la arquitectura implementada se muestra en la Fig. 1.

C. Configuración de Servidores y Clientes

El esquema planteado se implementó y ejecutó en el equipo anfitrión con VMware y GNS3, el mismo dispone de procesador Core i7, memoria RAM de 8 GB y almacenamiento de 500 GB. Para el servidor Web se utilizó una máquina virtual de un procesador y 512 MB de memoria RAM con Centos 6 y Apache 2 con dos páginas Web de prueba publicadas, para el cliente (víctima) se configuró una máquina virtual con un procesador y 512 MB de memoria RAM con Centos 6 y para el cliente (atacante) se configuró una máquina virtual con un procesador y 512 MB de memoria RAM con Kali Linux y el software BetterCAP para realizar los ataques.

D. Configuración del Firewall Cisco ASA 5520

Para la puesta en funcionamiento del Firewall Cisco ASA 5520 es necesario: i) Disponer de la imagen del IOS; ii) Cargar la imagen al software GNS3; iii) Configurar los parámetros de número de procesadores, memoria RAM, disco duro y iv) La configuración del equipo el cual permita cumplir su cometido. En la Fig. 2 se muestra la ventana de configuración de GNS3 para el Firewall Cisco ASA 5520.

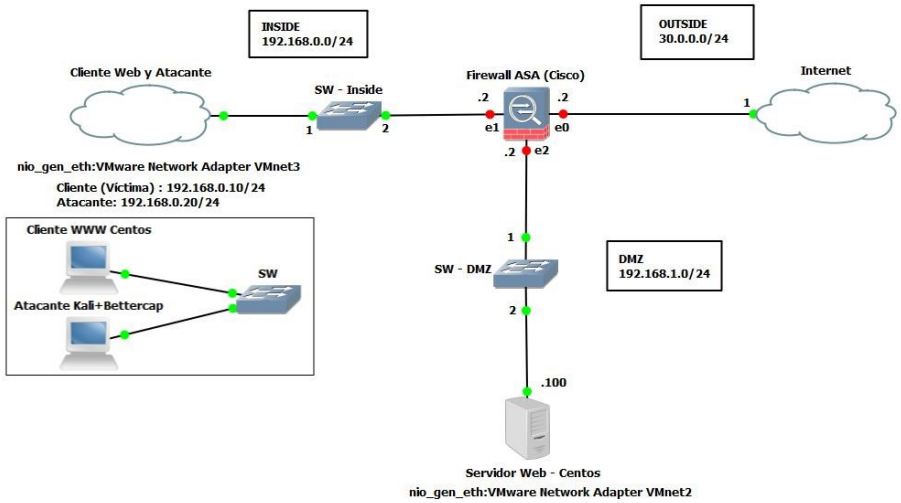


Figure 1. Configuración de Firewall en GNS3

Como parte de la configuración lógica del equipo se crearon los segmentos de la red y reglas que permiten dirigir los paquetes de acuerdo a las solicitudes, por ejemplo: un cliente en la LAN que solicita acceder al servidor Web, sus solicitudes son redirigidas por el Firewall hacia la DMZ donde se encuentra dicho servidor y viceversa, de igual manera existen reglas que permiten únicamente las peticiones hacia el puerto tcp/80 (http) en el cual se ejecuta el servicio Web en conjunto con la respectiva regla NAT que permite redirigir los paquetes de este servidor en función de donde provienen las solicitudes. A continuación, se muestra la interface, puerto y descripción de lo configurado en el dispositivo tal como se muestra en la Fig. 1:

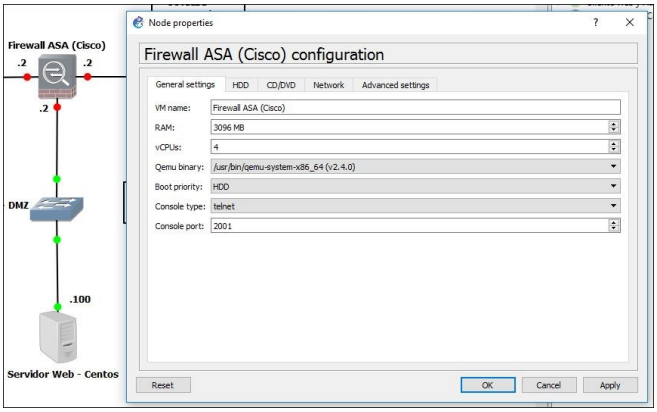


Figure 2. Configuración de Firewall en GNS3

- Interface "GigabitEthernet 0/0": Interface conectada directamente a la red externa (outside) WAN, con red 30.0.0.0/24.
- Interface "GigabitEthernet 0/1": Interface conectada directamente a la red interna (inside) LAN, con red 192.168.0.0/24.
- Interface "GigabirEthernet 0/2": Interface conectada directamente a la red DMZ, con red 192.168.1.0/24.

E. Implementación de la plataforma experimental

Para la implementación de este experimento se ha utilizado el siguiente procedimiento: i) En primer lugar, se ha creado en VMware el servidor Web para lo cual se instaló Centos 6 y el servidor Apache2; ii) Se creó la segunda máquina virtual, el cliente, con Centos 6 habilitado el entorno gráfico con el fin de poder utilizar el navegador Web integrado para acceder a las páginas publicadas en el servidor Web; iii) Posteriormente, se creó la tercera máquina virtual, el atacante, con Kali Linux y se instaló la herramienta BetterCAP para generar los ataques; iv) Cada una de las máquinas virtuales han sido configuradas en redes virtuales distintas para separar el tráfico de cada uno de los segmentos de red; v) En GNS3, se ha creado un nuevo proyecto y se ha importado cada una de las máquinas virtuales de VMware, además se habilitó el Firewall con las respetivas configuraciones descritas anteriormente, por medio de switches se ha interconectado todos los clientes hacia el Firewall.

De acuerdo al procedimiento descrito anteriormente se puede evidenciar que el software GNS3 se convierte en la base para este experimento debido a que controla tanto los equipos de conectividad como las máquinas virtuales, obteniendo como beneficio la posibilidad de iniciar, pausar o detener el experimento y evitar la saturación de los recursos del computador anfitrión.

F. Generación de Ataques

La generación de ataques ARP Spoof se realizó ejecutando la aplicación BetterCAP en la máquina atacante con Kali Linux desde la LAN hacia la víctima que solicita páginas Web al servidor ubicado en la DMZ. Para generar un ataque, la herramienta posee varias opciones, sin embargo, se utilizó la opción que permite capturar el tráfico http que se genere cuando la víctima solicite las páginas Web al servidor. En la Fig. 3 se muestra la pantalla donde se muestra el funcionamiento del software al realizar un ataque.

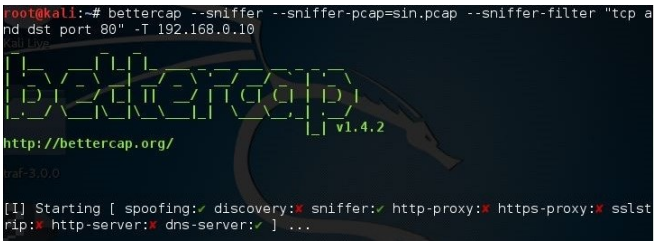


Figure 3. Funcionamiento de BetterCAP

Este tipo de ataques se ejecutan añadiendo un registro extra en la tabla ARP de la víctima a fin que todo el tráfico generado aparte de ir por la puerta de enlace al destino, lo envía al equipo atacante. Este tráfico capturado se evidenció comprobando la cantidad de información recibida a través de la consola de la propia aplicación la cual permite esta funcionalidad o enviando estos paquetes a un archivo pcap para su posterior análisis en aplicaciones como Wireshark y Ethereal.

G. Algoritmo que detecta el ataque ARP Spoof

A diferencia de otros tipos de ataques como los de denegación de servicio en los cuales al momento de iniciarse en el equipo víctima se puede observar claramente saturación de los recursos como procesamiento, memoria RAM, número de paquetes recibidos, etc., lo que a simple vista da la impresión al usuario que algo está sucediendo en su computador y genera una alerta, en los ataques de ARP Spoof no se produce saturación, si se conociera a ciencia cierta que es víctima de ataque, la única forma de detectar es observando los registros de la tabla ARP e identificar que existe un registro de direcciones MAC duplicados con distintas direcciones IP. Es decir, que el tráfico generado por el cliente irá hacia la puerta de enlace y hacia la máquina atacante.

De acuerdo a lo anteriormente mencionado, para poder realizar la detección oportuna de este tipo de ataques se ha diseñado un script de Linux el cual hace un análisis de la tabla ARP a fin de detectar la intrusión. En la Fig. 4 se muestra el diagrama de flujo de este algoritmo.

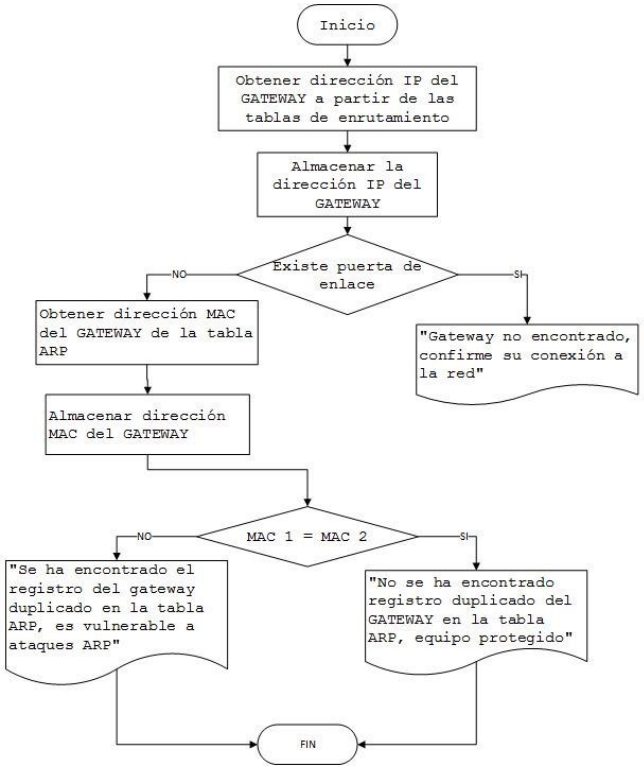


Figure 4. Flujo del proceso de detección de ataque

H. Algoritmo que mitiga el ataque ARP Spoof

Ante una inminente amenaza de un ataque ARP Spoof es necesario tomar acciones inmediatas debido a que el atacante es capaz de ver todo el tráfico de la víctima, es decir podría ser capaz en el caso de un ataque hacia el tráfico http, identificar claves, páginas accedidas, palabras más buscadas, etc. Para ello, se ha diseñado un algoritmo que mitiga la amenaza basándose en el principio básico de la creación de registros estáticos en la tabla ARP, es decir que en el cliente se registrara la dirección IP y la MAC de la puerta de enlace, con ello a pesar que el atacante intente realizar la duplicación del registro del Gateway en la tabla ARP, la víctima siempre enviará el tráfico hacia la dirección IP del registro estático creado, con ello se bloquea todo envío de tráfico hacia otro destino.

Para ejecutar lo anteriormente descrito, se ha diseñado un script de Linux el cual confirma tanto la dirección IP y MAC real del Gateway y con esos datos crea el registro estático en la tabla ARP. En la Fig. 5 se muestra el diagrama de flujo de este algoritmo.

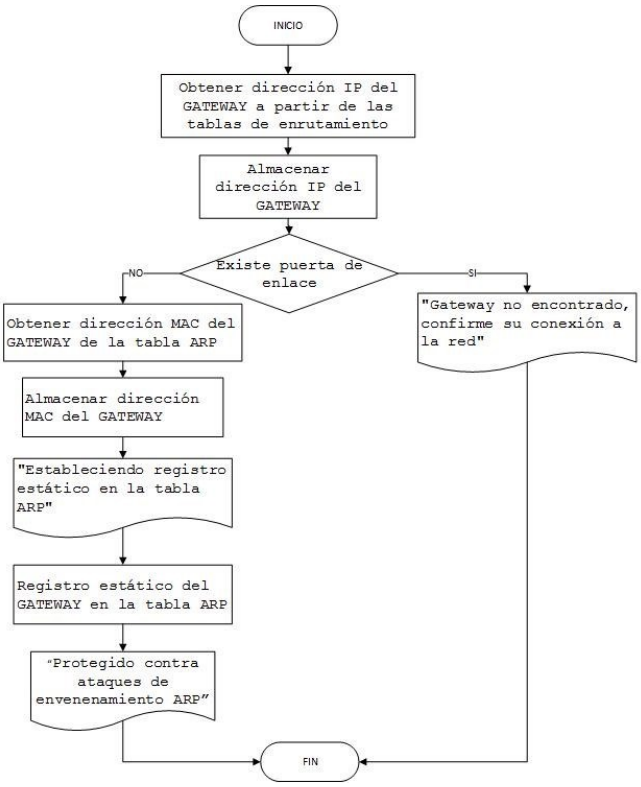


Figure 5. Flujo del proceso de mitigación de ataque

IV. EVALUACIÓN DE RESULTADOS Y DISCUSIÓN

A. Evaluación de resultados, línea base (ataque)

Tal como se explicó en secciones anteriores, este experimento se enfoca en la detección y mitigación de ataques ARP Spoof, para ello, a fin de evaluar el resultado de la mitigación diseñada se planteó una prueba en la cual la máquina atacante intercepta los paquetes de un cliente que solicita páginas Web a un servidor. La efectividad del algoritmo diseñado se midió en términos del número de paquetes capturados en el atacante, para lo cual se configuró la herramienta para que envíe todo el tráfico capturado a un archivo pcap y como segunda forma de verificación se observó la salida de la consola de la aplicación.

El procedimiento inicia detectado que equipos se encuentran activos en la red, luego se inicia el ataque hacia la máquina víctima, en esta etapa BetterCAP muestra en su consola las direcciones IP y MAC del Gateway de la víctima para posterior con esta información proceder a agregar el registro en la tabla MAC de la víctima la cual permita que el tráfico generado además ir por el Gateway predeterminado vaya a la máquina atacante. Luego, en la máquina cliente se evidencia en la tabla MAC que existe un registro duplicado, es decir, existe dos direccione IP con una misma dirección MAC. En la Fig. 6 se muestra la duplicidad existente luego de efectuado el ataque.

```
[root@centosCG Escritorio]# arp -n
Address                  HWtype  HWaddress           Flags Mask
192.168.0.2              ether   00:0c:29:71:d9:c0   C
192.168.0.20             ether   00:0c:29:71:d9:c0   C
```

Figure 6. Duplicidad de dirección MAC

A este punto del experimento el atacante es capaz de aplicar todas las opciones de ataque derivados del ARP Spoof, para este experimento en particular se realizó la prueba capturando el tráfico http de la víctima y enviado el mismo a un archivo pcap, para ello desde el cliente se realizó la descarga de un archivo de 35 MB desde el servidor Web y se midió el número de paquetes en el archivo pcap. En la Fig. 7 se muestra el número de paquetes capturados en el intervalo de tiempo que duró la descarga.

Figure 7. Número de paquetes capturados durante el ataque

Además, la consola de la aplicación BetterCAP muestra que se ha solicitado una descarga y lo muestra en pantalla. En la Fig. 8 se muestra la salida de la consola.

```
root@kali:~# bettercap --sniffer --sniffer-pcap=traficoc.pcap --sniffer-filter "
tcp and dst port 80" -T 192.168.0.10
                                              |_| v1.4.2
http://bettercap.org/

[I] Starting [ spoofing:✔ discovery:✘ sniffer:✔ http-proxy:✘ https-proxy:✘ sslst
rip:✘ http-server:✘ dns-server:✔ ] ...
[I] [GATEWAY] 192.168.0.2 : 00:00:AB:2D:01:01 ( Logic Modeling )
[I] [SNIFFER] Saving packets to /root/traficoc.pcap .
[I] [TARGET] 192.168.0.10 : 00:0C:29:B2:03:41 ( VMware )
[192.168.0.10 > 192.168.1.100:http] [GET] http://192.168.1.100/archivo.bin
[192.168.0.10 > 192.168.1.100:http] [GET] http://192.168.1.100/archivo.bin
```

Figure 8. Salida de la consola de BetterCAP durante el ataque

Respecto al consumo de recursos en la maquina atacada no se observó incremento significativo en términos de procesamiento y memoria RAM. En la Fig. 9 se muestra el consumo de recursos durante el ataque en la máquina víctima.

Por otro lado, en la máquina atacante se evidenció que existe un incremento del 60% del consumo de procesamiento mientras que en la memoria RAM no se observó incremento significativo. En la Fig. 10 se muestra el consumo de recursos en la máquina atacante.

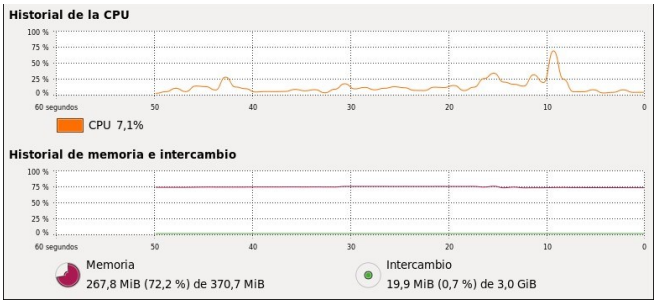


Figure 9. Consumo de recursos en la víctima durante el ataque

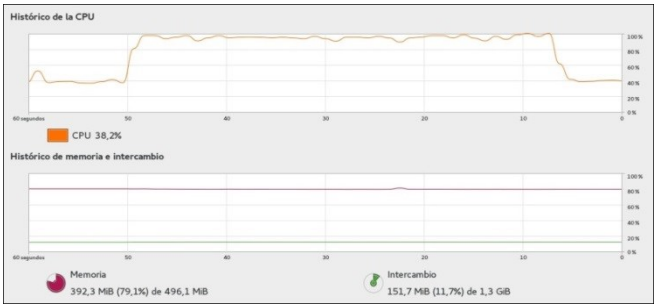


Figure 10. Consumo de recursos en el atacante durante el ataque

B. Evaluación de resultados aplicando mitigación

Para mitigar los ataques ARP Spoof se aplicó el algoritmo mencionado en secciones anteriores, mismo que establece un registro ARP estático en la máquina atacante. En la Fig. 11 se muestra la salida luego de aplicada esta mitigación con el detalle del registro creado en la tabla ARP.

```
Interrupción de la interfaz eth0:  Estado de dispositivo: 3 (desconectado)
                                                                [  OK  ]
Interrupción de la interfaz de loopback:                        [  OK  ]
Activación de la interfaz de loopback:                          [  OK  ]
Activando interfaz eth0:  Estado de conexión activa: activada
Ruta de conexión activa: /org/freedesktop/NetworkManager/ActiveConnection/3
                                                                [  OK  ]
Obteniendo la direccion IP y MAC del Gateway ....
IP del Gateway: 192.168.0.2 MAC: 00:00:AB:2D:01:01
Estableciendo registro estatico en la tabla ARP
192.168.0.2              ether   00:00:ab:2d:01:01   CM                    eth0
Protegido contra ataques de envenenamiento ARP
[root@centosCG Escritorio]#
```

Figure 11. Aplicación de mitigación

Esta acción es transparente para el atacante por lo que el procedimiento para realizar el ataque es el mismo mencionado en la sección anterior. Luego de efectuado este ataque se pudo observar que se vuelve a crear un nuevo registro en la tabla arp con la diferencia que ya no existe la duplicidad de direcciones MAC. En la Fig. 12 se muestra la tabla ARP aplicado la mitigación.

```
[root@centosCG Escritorio]# arp -n
Address                  HWtype  HWaddress           Flags Mask
192.168.0.20             ether   00:0c:29:71:d9:c0   C
192.168.0.2              ether   00:00:ab:2d:01:01   CM
```

Figure 12. Tabla MAC aplicado mitigación

De la misma forma para medir el efecto de la mitigación se realizó la misma prueba de descarga de un archivo desde el servidor Web y se midió el número de paquetes del archivo pcap creado por BetterCAP para transferir los paquetes capturados. El primer efecto que se evidenció es que luego de finalizado la descarga del archivo BetterCAP no pudo crear el archivo pcap debido a no capturó ningún paquete. En las Fig. 13 y 14 se muestra el número de paquetes capturados por

BetterCAP y la salida de la consola de la aplicación la misma que no mostró información alguna.

Figure 13. Número de paquetes capturados durante el ataque aplicado la mitigación

Figure 14. Salida de la consola de BetterCAP durante el ataque aplicada la mitigación

Respecto al consumo de recursos en la maquina atacada no se observó de igual manera incremento significativo en términos de procesamiento y memoria RAM. Por otro lado, en la máquina atacante tampoco se evidencio incremento de recursos esto es debido a que ya no debió procesar paquetes ya que no logró capturar a ninguno. En la Fig. 15 se muestra el consumo de recursos en la máquina atacante.

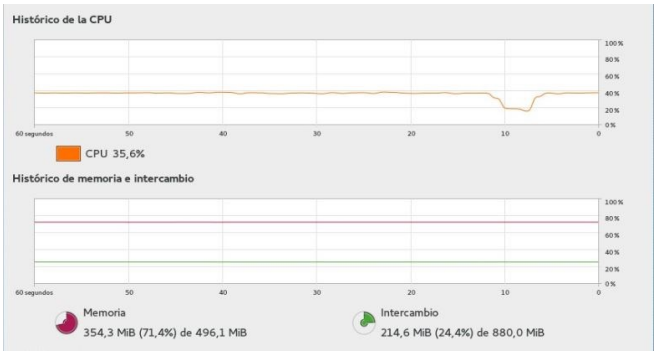


Figure 15. Consumo de recursos en el atacante durante el ataque aplicada la mitigación

V. Trabajos Relacionados.

En esta sección se han incluido los Trabajos más relevantes, que se han encontrado durante toda la investigación. Con respecto al ámbito educativo los trabajos desarrollados por Muñoz [1] y Zapata [10] indica el monitoreo e identificación de la red, define los ataques que puede sufrir, quien o quienes son los atacantes se analiza los tipos de ataques y las vulnerabilidades que tiene el sistema operativo, se muestra como defenderse de ataques y las medidas de seguridad. Zapata diseña e implementa varias topologías de experimentación usando entornos virtuales de red, dentro de las cuales se prueba el escaneo de puertos, denegación de servicios entre otros, los resultados de esta investigación proponen reducir las amenazas y vulnerabilidades mediante un demonio en Shell script el que detecta, controla y mitiga los ataques mencionados en el estudio.

La investigación de F. J. Díaz Jiménez y J.G. Palacio Velásquez [2] presenta una serie de técnicas que pueden ser utilizadas para vulnerar la seguridad en una red desde adentro, a través de la técnica de ARP Spoofing, realizando ataques de tipo Man In The Middle, al final presenta una serie de técnicas que pueden ser aplicadas para proteger a las redes de dichos ataques y minimizar los riesgos de robo de información. En este mismo ámbito Melgar Jara, E. S. [3] y Cazar Jácome, D. A. [4] han utilizado conceptos de auditoria de red que permitan detectar los múltiples ataques producidos por personas ajenas, presenta una herramienta que podrá detectar ataques comunes que se encuentran en las redes IPv4/IPv6 como por ejemplo "Man In The Middle" o la Denegación de Servicios.

Adicionalmente Gutiérrez Benito, F y Nicolalde Rodríguez, D. A. utilizan conceptos de virtualización y se realiza un estudio comparativo de sistemas de virtualización. Todos estos estudios investigativos han sido utilizados para la presente investigación. En relación a la utilización de entornos virtualizados los estudios de [6], [7] y [10] demuestran el uso de esta herramienta como una opción, ya que se puede simular servidores en operación reduciendo significativamente los costos y la administración.

VI. Conclusiones y Trabajos Futuros.

El presente trabajo investigativo se enfocó en la evaluación del ataque ARP Spoof utilizando plataformas de virtualización y para la simulación de la red se utilizó GNS3, el cuál emula firmware de los routers y dispositivos de Cisco Systems (IOS) además de establecer escenarios para conectar cada uno de los equipos. Para producir el ataque se utilizó el software libre BetterCAP, para la evaluación del ataque se midió en términos del número de paquetes capturados en el atacante, para lo cual se configuró la herramienta para que envíe todo el tráfico capturado a un archivo pcap y como segunda forma de verificación se observó la salida de la consola de la aplicación. Para contrarrestar dichos ataques, se desarrolló un demonio en Shell script que detectó, controló y mitigó el ataque ARP Spoof.

Como trabajo futuro se plantea evaluar los ataques ARP Replay y ARP Poisoning con el algoritmo desarrollado a fin de validar su efectividad, además se plantea evaluar ataques ARP Spoof en redes IPv6, utilizando otros mecanismos de mitigación como la encriptación, sistemas de detección de intrusos.

Referencias

[1] Muñoz Cedeño, D. E. (2014). "Monitoreo e identificación de ataques a redes" (Doctoral dissertation).

[2] J. Díaz Jiménez y J.G. Palacio Velásquez. Disección de un ataque MITM mediante ARP Spoofing y Técnicas de Protección Existentes, Barranquilla, Ed. Coruniamericana, Vol. I, 2012. 9-24

[3] Melgar Jara, E. S. (2015). Desarrollo de un conjunto de aplicaciones para detección de ataques en redes IPv4/IPv6 utilizando Python.

[4] Cazar Jácome, D. A. (2015). Análisis de IP Spoofing en redes IPv6 (Doctoral dissertation, 2015.).

[5] Herrera Figueroa, Helmuth Lenin. "Simulación de ataques a redes IP en un entorno corporativo real." (2015).

[6] Gutiérrez Benito, F. (2014). Laboratorio Virtualizado de Seguridad Informática con Kali Linux.

[7] Nicolalde Rodríguez, D. A. (2015). Estudio comparativo de sistemas de virtualización y de seguridad, caso de estudio Museo QCAZ de la PUCE.

[8] Echeverry Parada, J. S. (2013). Metodología para el diagnóstico continúo de la seguridad informática de la red de datos de la Universidad Militar Nueva Granada.

[9] Chumi Sarmiento, W., & Flores Escobar, D. (2014). Análisis forense a paquetes de datos en la red LAN de la Universidad Tecnológica Equinoccial como aporte al cumplimiento de las Normas PCI-DSS (Doctoral dissertation, Universidad de las Fuerzas Armadas ESPE. Maestría en Evaluación y Auditoría de Sistemas Tecnológicos.).

[10] Zapata Molina, L. P. (2012). Evaluación y mitigación de ataques reales a redes IP utilizando tecnologías de virtualización de libre distribución

[11] VMWare: https://www.vmware.com/products/workstation/. Última comprobación, marzo 2016.

[12] GNS3: https://www.gns3.com/. Última comprobación, marzo 2016.

[13] Cisco ASA 5500: https://www.cisco.com/web/ES/publicaciones/07-08-cisco-dispositivos-serie-ASA5500.pdf. Última comprobación, marzo 2016.

[14] Apache: https://httpd.apache.org/. Última comprobación, marzo 2016.

[15] BetterCAP: https://www.BetterCAP.org/. Última comprobación, marzo 2016.

[16] Tamayo Domínguez, M. F. (2013). Estudio, diseño y simulación en gns3 de guías de laboratorio para redes de datos ii y networking de la facultad de electrónica de la Universidad Israel. Quito.

[17] Díaz Cervantes, L. (2010). Evaluación de la herramienta GNS3 con conectividad a enrutadores reales.

[18] Fiallos Noboa, J. G. (2012). Análisis de tráfico" IP" para medianas empresas basado en software libre como parte de una política de seguridad informática

Como prueba de concepto de este proyecto favor ver el Video URL: https://youtu.be/OFfH6JN8tg0

GEEKS DECC-REPORTS

VOL 1 No 6, 2015

TENDENCIAS EN COMPUTACIÓN

CALL FOR PAPERS

ISSN: 1390-5236

MOTIVACIÓN

La Universidad de las Fuerzas Armadas ESPE y el Departamento de Ciencias de la Computación (DECC) conscientes de su responsabilidad social y buscando impulsar iniciativas tendientes a promover el desarrollo integral de la Ciencia y Tecnología en el Ecuador, invita a investigadores a participar en el próximo ejemplar de la revista GEEKS-DECC Report Tendencias en Computación.

TEMAS DE INTERES

El Comité Editorial de la revista pone a disposición de la comunidad científica el portal para envío de trabajos de investigación, desarrollo e innovación en las siguientes áreas, no exclusivas ni totalmente restrictas:

- Ingeniería de Software
- Sistemas Distribuidos
- Tecnologías de la Información y Comunicación
- Computación Orientada a Servicios
- Auditoria Informática y Evaluación de sistemas
- Algoritmia y Programación
- Tecnologías de Virtualización
- Cloud Computing
- Seguridad Informática
- Minería de datos estructurados y no estructurados
- Inteligencia Artificial
- Entornos Virtuales y Realidad Aumentada.

INFORMACIÓN PARA LA PRESENTACIÓN DE ARTÍCULOS TÉCNICOS

Los artículos pueden ser escritos en español, inglés o portugués, y deben ser remitidos al e-mail wmfuertes@espe.edu.ec hasta las 23:59:59 (-5 GMT) del día 24 de noviembre del 2015. Los trabajos enviados deben presentar resultados de investigación originales o reportes con experiencias relevantes y no deben estar publicados ni en proceso de evaluación en otras revistas o conferencias técnicas. Todos los artículos aceptados después de la correspondiente revisión por pares serán publicados en la revista GEEKS, DECC Report Tendencias en Computación cuyo ISSN es 1390-5236 y que cuenta con un tiraje mínimo de 700 ejemplares.

La extensión máxima de los artículos es 8 páginas formato IEEE doble columna. El formato para la escritura del artículo está disponible en:
https://www.ieee.org/conferences_events/conferences/publishing/templates.html

FECHAS IMPORTANTES:

- Envío de artículo completo: 24/11/2015
- Notificación de aceptación: 15/12/2015
- Envío de la versión final: 23/12/2015