ESCUELA SUPERIOR POLITÉCNICA DEL CHIMBORAZO

ANÁLISIS DE VULNERABILIDADES EN REDES ETHERNET UTILIZADAS PARA LA COMPUTACIÓN EN MALLA CON UN MIDDLEWARE FREE SOURCE UTILIZANDO LA METODOLOGÍA PPDIOO

MARCO ANTONIO GAVILANES SAGÑAY

Trabajo de Titulación modalidad: Proyectos de Investigación y Desarrollo presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de:

MAGISTER EN INTERCONECTIVIDAD DE REDES

Riobamba - Ecuador

Enero 2019

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

El trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, titulado "ANÁLISIS DE VULNERABILIDADES EN REDES ETHERNET UTILIZADAS PARA LA COMPUTACIÓN EN MALLA CON UN MIDDLEWARE FREE SOURCE UTILIZANDO LA METODOLOGÍA PPDIOO", de responsabilidad del Ing. Marco Antonio Gavilanes Sagñay, ha sido minuciosamente revisado y se autoriza su presentación.

Ph.D. PROAÑO ORTIZ FREDY BLADIMIR PRESIDENTE	_____ FIRMA
MsC. RAMOS VALENCIA MARCO VINICIO DIRECTOR	_____ FIRMA
MsC. VELOZ REMACHE GERMANIA DEL ROCIO MIEMBRO	_____ FIRMA
MsC. LOZADA YANEZ RAUL MARCELO MIEMBRO	_____ FIRMA

Riobamba, Enero 2019

DERECHOS INTELECTUALES

Yo, Marco Antonio Gavilanes Sagñay, soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del mismo pertenece a la Escuela Superior Politécnica de Chimborazo.

_____

Ing. Marco Antonio Gavilanes Sagñay

CC: 060326479-7

DEDICATORIA

La fortaleza que nos dá la fé, el esfuerzo y optimismo durante estos años de estudio, son el fruto de la gente que cree en mí, es por ello que el presente trabajo de investigación está dedicado a las personas que a lo largo de mi vida me han dado la formación para ser una persona de bien.

Con todo mi amor dedico este trabajo primero a Dios, a mis Padres, mi Hermano, a mi adorada Esposa y a mi más preciado tesoro a mi Hijita Aylin; y especialmente a mi Mamita y todos los seres queridos que me está cuidando desde el cielo.

AGRADECIMIENTO

Los agradecimientos están dirigidos a todas las personas que me apoyaron incondicionalmente en la realización del presente trabajo de investigación, primeramente a Dios y a la Virgen María que siempre guían mi vida; a mis padres, mi hermano y a mi adorada esposa e hijita por su amor, y apoyo incondicional. Al Ing. Vinicio Ramos Valencia Tutor de mi Tesis y a los distinguidos Miembros de la misma por su paciencia y gran colaboración en la realización de este trabajo.

ÍNDICE

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE ANEXOS

RESUMEN

El objetivo fue analizar las vulnerabilidades en redes Ethernet utilizadas para la computación en malla con un middleware. En el desarrollo de la investigación se empleó el método experimental, con la finalidad de aplicar mejores prácticas en un ambiente de pruebas, utilizando la metodología PPDIOO. Se utilizó Globus Toolkit como middleware para la gestión de la malla. Entre las herramientas empleadas están Ubuntu Loiq fue empleado para el ataque de denegación de servicios distributivos, y Ubuntu- NMAP que servicios como scanner y rastreador de puertos. Respecto al análisis de vulnerabilidades y riesgos, se aplicaron los fundamentos de Magerit y OWASP. En el estudio experimental, se emplearon pruebas de intercepción de paquetes y denegación de las mismas, enfocado en el valor de paquetes interactuados e interceptados con respecto a la seguridad que ofrece la red, y los paquetes perdidos respecto al rendimiento. Para el análisis estadístico se aplicó Wilcoxon, luego de cuyo análisis se concluyó que la implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO sí conllevó una mejora en la disponibilidad y seguridad de las redes utilizadas para computación en malla, pero a la vez disminuyó el rendimiento de la misma. En términos porcentuales la disponibilidad incrementó en un 40,31%, la seguridad en un 100%, pero el rendimiento disminuyó en un 30,20%.

Palabras Claves: <TECNOLOGIA Y CIENCIAS DE LA INGENIERÍA>, <REDES>, <ETHERNET>, <COMPUTACIÓN EN MALLA>; <MIDDLEWARE (SOFTWARE)>, <GLOBUS TOOLKIT>

ABSTRACT

The Objective was to analyze the vulnerabilities in Ethernet networks used for computing in mesh with a middleware. In the development of the research, the experimental method was used, with the purposed of applying best practiced in a test environment, using the PPDIOO methodology. Globus Toolkit was used as a middleware for mesh management. Among the tools used are Ubuntu Loiq was employed for the attack of denial of distributive services, and Ubuntu-NMAP that services as scanner and Port tracker. About the analysis of vulnerabilities and risks, the fundamentals of Magerit and OWASP were applied. In the experimental study, packet intercept and denial test were used focused on the value of interacting and intercepted packets with respect to the security offered by the network, and the packets lost with respect to performance. For the statistical analysis, Wilcoxon was applied, after the analysis it was concluded that the implementation of practical improvements in Ethernet networks, in the access layers, distribution and nucleus applying the PPDIOO methodology if it led an improvement in the availability and Security of the networks used for computing in mesh, but at the same time it diminished the performance of the same one. In percentage terms, the availability increase by 40.31 %, security by 100% but the yield decreased by 30.20%.

Key words: TECHNOLOGY AND ENGINEERING SCIENCES, NETWORKS, ETHERNET, MESH COMPUTING, MIDDLEWARE (SOFTWARE), GLOBUS TOOLKIT.

INTRODUCCIÓN

El presente trabajo de investigación, tiene como objetivo analizar las vulnerabilidades en redes Ethernet utilizadas para la computación en malla con un middleware Free Source, el estudio se realizó con la finalidad de evaluar las debilidades que se presentan en las redes de manera que se obtenga información de mismo y conlleve a la realización de indagaciones. Con el pasar del tiempo las computadoras y las comunicaciones ha ido variando de manera profunda y aún más el diseño y la utilización de las redes.

Es importante conocer que existe una variedad de factores que hay que reconocer e identificar las amenazas a las que pueden estar expuestos los sistemas de computación, el trabajo de investigación está enfocado en el análisis de los sistemas ante la utilización y el avance de la tecnología.

La investigación está basada en cinco capítulos las mismas que se detallan a continuación.

Capítulo I Problema de investigación: en el primer capítulo se describe la problemática, en donde se especifican los parámetros que conllevan a que el presente estudio se realice, posteriormente se detalla la formulación del problema y la sistematización del mismo.

Capítulo II Marco teórico: Dentro del marco teórico se estructuraron las definiciones de las variables tanto dependiente como independiente con la finalidad de obtener bases científicas y teóricas.

Capítulo III Marco Metodológico: en este capítulo se desarrolló el tipo y diseño de investigación que se utilizó para el presente estudio, definiendo y mostrando las variables de operacionalización así como los métodos, técnicas e instrumentos aplicados para la recolección de datos criterios éticos y de rigor.

Capitulo IV Resultados: Finalmente en este capítulo se detallan los resultados obtenidos del contraste de indicadores de la investigación, empleando una serie de procesos ilustrados en gráficas y cuadros.

CAPÍTULO I

1. GENERALIDADES DE LA INVESTIGACIÓN

1.1 Problema de Investigación

1.1.1 Planteamiento del Problema

En todo el mundo, el avance de la tecnología, ha provocado que la interacción de los usuarios con los servidores de datos crezca exponencialmente, logrando en la actualidad que, según (Unión Internacional de las Telecomunicaciones, 2017), un 79.6% de la población Europea tenga acceso a Internet, en América un 65.9%, y un promedio del 48% de la población mundial.

Este gran volumen de usuarios interconectados a la red generan un tráfico de información permanente, mismo que tiene que ser gestionado por los equipos de red globales que mantienen la conectividad, así como por los servidores que almacenan y proveen la información que buscan consumir estos usuarios. Esto ha provocado que, para cubrir esta demanda, los servidores y las granjas de servidores hayan tenido que crecer simultáneamente a la demanda, situación que genera costos permanentes para los gestores de información en Internet.

La tecnología, pese a su avance, presenta claras limitaciones en cuanto a puntos como el tamaño, el rendimiento, el almacenamiento, por cuanto la tendencia para empresas que manejan grandes volúmenes de datos y de tráfico en red se ha visto orientada hacia la potencia de procesamiento y almacenamiento por población. Es decir, han decidido incrementar el número de equipos para obtener mayores recursos, pues actualmente no existe un servidor con la suficiente potencia para cubrir la demanda de todas las empresas de internet de la actualidad.

Teniendo en cuenta la limitante actual de la tecnología, los fabricantes han optado por enfocarse en potencializar los equipos de los usuarios, llegando a un punto en el que un computador personal puede tener más recursos que un servidor del año pasado. La inversión en incrementar los servidores para las empresas genera entonces gastos considerables, adicionales a los gastos en equipos personales para los trabajadores dado que, como se mencionó anteriormente, las empresas están ampliamente enlazadas a la gestión de información en sistemas empresariales.

Con el afán de atender la creciente demanda de información por parte de los usuarios, nacen nuevas tendencias para mejorar el rendimiento de las aplicaciones y la información que se comparte en este mundo integrado. Algunos esfuerzos notables en cuanto al aprovechamiento de los recursos de los computadores personales como fuente de procesamiento de información son, entre otros, los lenguajes de programación que se ejecutan al lado del cliente, así como paradigmas de red que distribuyen la gestión de información para mejorar el rendimiento y procesamiento de datos.

En el área de redes, diferentes tecnologías han aparecido tratando de optimizar las granjas de servidores, siendo una de las más exitosas la computación en malla, la cual se define como un paradigma de computación distribuida en la que los computadores de una red comparten sus recursos con cualquier otro sistema (Globus.org, 2017) y están a cargo de un solo equipo gestor (controlador) de todo el ambiente. Al intercambiar información permanentemente, cada uno de los equipos que participan se considera vulnerable a ataques o fallas, a la vez que el medio por el que se realiza la comunicación se vuelve también un factor trascendental a tener en cuenta al momento del diseño y de la implantación de la red, debiendo ser lo más seguro y eficiente posible a fin de que la toma de decisiones y la gestión de recursos sea lo más correcta posible.

A lo largo del tiempo, este factor no ha sido muy estudiado por la limitación que se le ha dado a la computación en malla en cuanto a su implementación en redes locales y al uso que se le ha dado a la misma. Como ejemplo, Disney empleó este paradigma de procesamiento para la generación de la primera película generada totalmente por computadora en la historia misma que, de haberse renderizado con el mejor computador disponible en esa época, hubiese tardado 240000 horas (renderizaba 1 frame en 2 horas, debiendo renderizar 25 frames por segundo en una película de 80 minutos), problema que fue solucionado al utilizar cientos de computadores configurados en malla que se repartieran el trabajo (Fernández, 2016); se evidenció por primera vez desperdicios de tiempo corrigiendo frames mal generados por problemas en la red.

Los grandes volúmenes de información que se deben manejar simultáneamente durante los procesos de minería de datos requieres un amplio rango de paralelismo sobre el cual procesar los grandes lotes de información requerida, tal como lo menciona Ashfaq Husssain, Naser, & Begum (2015) en su Tesis Doctoral “Minería de Datos con Conceptos de Computación en Malla”. Los autores concluyen que una efectiva minería de datos requiere del uso de técnicas predefinidas para la implementación de un modelo de procesamiento que permita resolver los problemas que se presentan en una red, en tiempo real, como son la confidencialidad y aseguramiento de la información, y de esta manera minimizar su vulnerabilidad.

La falta de medidas de seguridad en las redes de datos es un problema que crece día a día considerablemente, y de manera especial, en las redes con conexión a internet; esto se debe al creciente número de atacantes y a la deficiente gestión de las redes por parte de sus administradores. La integridad y confidencialidad de la información de empresas y organizaciones se encuentra entonces en riesgo, al encontrarse claramente expuesta a accesos no autorizados. Espinoza & Montoya (2016), en su análisis a las seguridades de red del Banco Nacional de Fomento en la matriz de la ciudad de Quito, encontraron varias vulnerabilidades entre las que se destacan: gran cantidad de puertos abiertos, programas CGI vulnerables, fugas de información a través de conexiones anónimas y suplantación de direcciones ip.

Considerando la complejidad en la gestión o administración de una red Ethernet para computación en malla (en cuanto al volumen de procesamiento de información requerido), es imperativo que los administradores de red apliquen las técnicas, configuraciones y correctivos necesarios para el aseguramiento de la integridad y confidencialidad de la información que ésta transmite. En este punto, si una red de computación en malla es vulnerable, el rendimiento global podría verse reducido.

La vulnerabilidad de una red de datos afecta a su rendimiento debido a que la calidad de servicio decae, y éste puede ser en algunos casos perceptible al cliente. De esta manera, la vulnerabilidad afecta en múltiples formas al proceso de telecomunicaciones, afectando la eficiencia de una red diseñada para ello.

1.1.2 Formulación del Problema

¿Cómo se puede mejorar el rendimiento de las redes para computación en malla reduciendo sus vulnerabilidades en seguridad?

1.1.3 Sistematización del Problema

- ¿Cómo influye el middleware en la seguridad y el rendimiento de las redes?
- ¿Cómo determinar vulnerabilidades de seguridad en la red de pruebas implementada en el área de investigación?
- ¿En qué capas se puede implementar las mejores prácticas para mejorar las características de la red?
- ¿Cómo mejora el rendimiento de las redes para computación en malla la reducción de sus vulnerabilidades de seguridad?

1.2 Justificación

Dada la creciente tasa de usuarios integrados a internet, produciendo y consumiendo información, existe una creciente necesidad de contar con una mayor cantidad de servidores capaces de satisfacer las necesidades de cada usuario.

En este contexto México, por ejemplo, ha visto un crecimiento del 12.5% de usuarios en internet entre los años 2006 y 2014 (Instituto Nacional de Estadística y Geografía, 2015), lo que representa una mayor demanda hacia los servidores. De esta manera, la infraestructura de la red debió ser mejorada en prestaciones.

En los últimos años se ha visto una constante tendencia por parte de los computadores personales a volverse más accesibles y potentes (Elnikety, Rowstron, Narayanan, Thereska, & Dinnelly, 2009). Actualmente, en el Ecuador, cada Gigabyte de almacenamiento de disco duro SATA cuesta menos de 10 centavos de dólar, tal como lo menciona el catálogo en línea de Serimtec, filial ecuatoriana de Western Digital (SERIMTEC, s.f.). Es así que, bajo este contexto y pretendiendo satisfacer las necesidades cambiantes del cliente, nacen diversos esfuerzos para aprovechar los recursos crecientes de los equipos personales, surgiendo así la idea básica de aprovechar todos o parte de sus recursos en alcanzar un fin común.

La programación o computación en malla, es otro ejemplo de lo antes mencionado, en el cual se busca utilizar los recursos de equipos distribuidos en un área geográfica con el fin de lograr una tarea. Por ejemplo, el proyecto SETI@HOME se encarga de buscar inteligencia extraterrestre con la ayuda de todos los voluntarios que quieran aportar a nivel mundial (setiathome, 2016). Éste funciona con el sistema operativo Debian como base, y emplea foros de ayuda en el cual se mencionan problemas como la pérdida de paquetes, la cual es causada por la topología de la red, y la reducción del rendimiento al obtener información de un cliente con un ancho de banda muy bajo.

Otra de las líneas más utilizadas de computación en malla son los laboratorios de Física de partículas y hasta la misma Organización Europea para la Investigación Nuclear, a partir de ahora CERN, con ayuda del cual se creó el mayor ejemplo de computación en malla actual: el EDG (Grid de Datos Europea) (CERN, 2016). De igual manera que en el ejemplo antes mencionado, éste describió que una gran ventaja para su implementación fue la homogeneidad que tiene la red en la cual fue implementada la topología de malla.

En los ejemplos mencionados a nivel mundial, en cada una de las infraestructuras de red implementadas para la computación en malla todos los autores coinciden en los inconvenientes que han presentado las mismas, siendo el punto de inflexión la reducción del rendimiento por efectos externos detectados en sus redes. Estos efectos se relacionan a paquetes interceptados o perdidos durante el manejo de los mismos, los cuales en su mayoría se deben a las características de la red, más allá del middleware utilizado. Los usuarios de los proyectos mencionados aportan entonces a los proyectos desde su red doméstica sin ajuste alguno y a través de la red mundial de internet, por cuanto se hace necesaria la existencia de un prototipo que mejore el rendimiento y la seguridad de las redes con este fin.

De esta manera, la presente investigación (como lo menciona la segunda pregunta directriz), iniciará con la determinación de las vulnerabilidades más frecuentes en este tipo de redes, para posteriormente implementar una red de pruebas en el ambiente de investigación, la cual servirá para la experimentación y análisis. A continuación se implementará en este ambiente de pruebas todas las sugerencias recolectadas durante la resolución de la tercera pregunta directriz mejorando así las condiciones de la red en el ambiente de pruebas. Con estas nuevas características en la red se realizará un nuevo análisis para definir un prototipo que integre las características finales de la red en su estado de mejor rendimiento y seguridad.

Por la naturaleza cuasi experimental de la investigación, esta no tiene un beneficiario específico personal o empresarial, sin embargo servirá como guía a cualquier proyecto que pretenda utilizar computadores distribuidos en redes domésticas como miembros de una red orientada a la computación en malla, por cuanto la optimización de recursos hardware subutilizados representa un ahorro económico para cualquier proyecto que ya posea una infraestructura de red para sus administrativos. Con los productos obtenidos de la presente investigación, dichos proyectos pueden ser orientados a funcionar como servidor de datos para proveer servicios e información a través de una red sin incurrir en costos adicionales en adquisición de hardware.

1.3 Objetivos

1.3.1 Objetivo General

Analizar las vulnerabilidades en redes Ethernet utilizadas para la computación en malla, con un middleware Free Source.

1.3.2 Objetivos Específicos

- Analizar middlewares Free Source para computación en malla, mediante procesos de revisión bibliográfica, comparación técnica y criterio de expertos, para la selección de uno a emplearse en los ambientes de pruebas.
- Desarrollar un ambiente de pruebas para el uso de computación en malla, el cual permita la determinación de vulnerabilidades en seguridad y rendimiento (línea base), basado en el estudio de casos debidamente descritos en el estado del arte.
- Implementar mejores prácticas en el ambiente de pruebas, en las capas de acceso, distribución y núcleo, mediante la aplicación de la metodología PPDIOO, para su análisis comparativo respecto a la línea base.
- Definir el prototipo de red y configuraciones que garantizan el mejor rendimiento, disponibilidad y seguridad de una red para computación en malla, mediante el análisis técnico de los resultados obtenidos en los procesos anteriores.

CAPÍTULO II

2. MARCO DE REFERENCIA

2.1 Antecedentes

Tras la revisión de trabajos de investigación se han encontrado temas similares que respaldan y sirven de referentes para el desarrollo de la investigación:

La investigación realizada por (Espinoza A. , 2010) con el tema: "Análisis de Vulnerabilidades de la Red LAN de la UTPL", en la que se concluye lo siguiente:

- Este proyecto con tal cualidad permite el accionar de forma positiva anticipando los hechos que se suscitarán, analizando el impacto que existe en los riesgos, el nivel de criticidad, mediante el estudio se demostró que la estrategia actual no es suficiente más bien la combinación de la infraestructura de seguridad en conjunto con las pruebas de ethical hacking. (Espinoza A. , 2010)
- Dentro del estudio se admite que no es posible llegar a tener total seguridad del 100%, pero aplicando ciertas estrategias que sirven de protección se obtendrá un tipo de seguridad aceptable de acuerdo a las necesidades requeridas de seguridad. (Espinoza A. , 2010)

Se recomienda:

- Realizar un análisis completo y profundo de los hallazgos encontrados, sobre todo conocer el efecto que provoca la aplicación de las estrategias ya sean estas a corto, mediano o a largo plazo, la comodidad es tomada en cuenta ante la seguridad. (Espinosa, 2010)
- Es recomendable mantenerse al tanto de las nuevas herramientas disponibles de seguridad, así es posible la actualización del plan de acción, debido a que las vulnerabilidades y amenazas cada día aumentan y cambian velozmente. Debido a la aplicación de la gestión de la seguridad con el plan de (PLAN-DO-CHECK-ACT) para realizar el monitoreo continuo de la seguridad de la información.

Así mismo se muestra el estudio de (Quishpe, 2016), con el tema: "Análisis de Vulnerabilidades en la Red LAN Jerárquica de la Universidad Nacional de Loja, en el Área de la Energía, Industrias y los Recursos Naturales No Renovables", en la que se concluye lo siguiente:

- Emplear la metodología clásica para el análisis de vulnerabilidad en la red del área de energía fue valioso, ya que ayuda a la síntesis del análisis y las transforma en tres frases con lo que se consigue realizar el proyecto en un tiempo corto y de forma organizada. (Quishpe, 2016)
- La utilización de herramientas como OpenVas y Nessus, benefician a la búsqueda de vulnerabilidades lógicas en los equipos considerados valiosos, ya que se complementaron entre amabas y con ello se obtuvo un eficiente resultado. (Quishpe, 2016)

Se recomienda:

- Es importante utilizar por lo menos dos herramientas que ayuden a la evaluación de vulnerabilidades en una red de datos, con el objetivo de lograr que ambas herramientas se complementen entre sí y encontrar la mayor cantidad de debilidades existentes en los equipos, con esto se llega a obtener un resultado muy eficaz para determinar el nivel de seguridad de cada equipo. (CISCO, 2010)
- Realizar la ejecución de un análisis interno y externo de vulnerabilidades de red por lo menos cada tres meses y después de cada cambio significativo en la red, tales como cambio de topología de red modificaciones en las normas de firewall, actualizaciones de protocolos. (CISCO, 2010)

Respecto al tema de vulnerabilidades, actualmente existen y se aplican ampliamente 2 metodologías: OWASP y Magerit, mismas que se dedican a detectar y gestionar vulnerabilidades y riesgos de los sistemas de información; Salgado (2014) lo expresa en su estudio "Análisis de las aplicaciones web de la Superintendencia de Bancos y Seguros, utilizando las recomendaciones Top Ten de OWASP para determinar los riesgos más críticos de seguridad e implementar buenas prácticas de seguridad para el desarrollo de sus aplicativos". En este trabajo el autor menciona que, al compararlo con otros proyectos de Software Libre, OWASP brinda mayor fiabilidad y es el más aceptado a nivel mundial. Dado que el presente estudio pretende la utilización en su mayoría de software libre convierte a OWASP en el proyecto elegido para la detección de vulnerabilidades.

Por otro lado tenemos a Magerit, mismo que Gaona (2013) en su estudio "Aplicación de la metodología Magerit para el análisis y gestión de riesgos de la seguridad de la información aplicado a la empresa Pesquera e Industrial Bravito S.A. en la ciudad de Machala" la compara con otras metodologías de gestión de riesgos. La autora concluye que Magerit es la mejor opción a implementar en el entorno del País, por lo cual se ha considerado su aplicación en el

presente estudio, y se espera obtener resultados aceptables aplicando las sugerencias de la autora antes mencionada.

2.2 Marco Teórico

2.2.1 Computación en malla

Las ciencias de la computación, vienen evolucionando permanentemente gracias a la concreción de las necesidades y servicios humanos que se fundamentan en la optimización de espacio, tiempo y distancia, principales dimensiones en las que académicos e investigadores de la disciplina fijan sus aportes. Los espacios de almacenamiento ya se han reducido desde las formas impresas hasta unidades de alta compresión citando diferentes métodos que preservan la integridad y consistencia de los datos, el tiempo de procesamiento se ha optimizado para dar significado a los conceptos de sincronización y "tiempo real", y finalmente la distancia es minimizada por las redes telemáticas, con las cuales las limitantes de rangos físicos son menguadas por las capacidades y prestaciones de dispositivos y técnicas de disposición de señales; todo esto dispuesto técnicamente para que se constituyan las herramientas esenciales para empresas, academias, familias, y en particular, grupos de investigación. Estas organizaciones, así como los recursos de los que disponen, pueden pertenecer a una misma área en la que realizan sus investigaciones, o en el más común de los casos de la actualidad, encontrarse distribuidos geográficamente (Mendoza, 2007)

Sin embargo, tal dispersión geográfica no podría de forma alguna hacer que se pierda consistencia operativa, razón por la cual se hace imprescindible generar desarrollos disciplinares en cuanto a la organización grid se refiere (Rodriguez, 2013)

Aunque individualmente, los computadores personales son inferiores en muchos aspectos a los supercomputadores, el poder combinado de cientos de computadores personales, unificados en un sistema en grid, representa un recurso computacional importante (Hidalgo, 2009).

Básicamente se trata de volver productivo el tiempo ocioso de los computadores personales o de escritorio que tiene disponibles el usuario.

La computación en malla es uno de los modelos de computación distribuida en la que una red de computadoras comparte sus recursos, entre los cuales se encuentra poder de posicionamiento memoria y almacenamiento con cualquier otro sistema. El objetivo de la computación en malla

es que se pueda procesar mayor cantidad de tareas en menor tiempo, es bastante útil debido a que mediante el mismo se realizan cálculos y procesos digitales de gran escala.

La computación en malla es también reconocida como una forma de red distribuida de procesamiento paralelo, a diferencia de lo que se conoce como computación distribuida debido a que varios equipos de la misma red comparten uno o más recursos y de esta manera se convierte en una supercomputadora.

Entre las principales características se encuentran:

Son distribuidos: Se caracteriza por lo general en dispersar geográficamente los recursos de la red de manera distributiva.

Heterogéneos: Las características de estos procesadores se basan en el poder de procesamiento debido a que un grupo de personas comparten el mismo computador. Un ejemplo de este tipo se puede ver muy claro en las redes sociales debido a que está conformada por muchos tipos de redes, cuyas diferencias se encuentran ocultas, debido a que todas las computadoras que se conectan mediante este procesador utilizan los protocolos de internet para comunicarse una con otra computadora que se encuentre conectada a una red llamada Ethernet y esta a su vez se conecta a otra computadora conectada una red llamada TokenRing, solo basta con que se realice una implementación de los protocolos de internet para cada una de las redes mencionadas.

Otro ejemplo de esta característica se puede observar en los lenguajes de comunicación y en cada una de las aplicaciones escritas por diferentes programadores; en el caso de los lenguajes de programación es muy importante tener como base la representación de caracteres así mismo de las estructuras de los datos de escrituras de los registros, las mismas que pueden cambiar o variar y ocasionar conflictos entre programas que necesiten de la comunicación entre ellos.

Otro ejemplo lo podemos ver en los lenguajes de programación y en las aplicaciones escritas por diferentes programadores; en el caso de los lenguajes de programación es importante tener en cuenta las diferencias que puede haber en la representación de los caracteres y estructuras de datos como cadenas de caracteres y registros, las cuales pueden variar y pueden ocasionar conflictos entre programas que necesitan comunicarse entre ellos. De igual manera dos programas que son desarrollados por programadores diferentes tienen que utilizar estándares comunes para la comunicación en red y para la representación de los datos elementales y las estructuras de datos en los mensajes, ya que, si no se cuenta con dichas similitudes, los

programas no podrán comunicarse entre sí aunque se hayan desarrollado en el mismo lenguaje de programación.

Un ejemplo de esto lo podemos ver muy claro en Internet, ya que es una red que está conformada por muchos tipos de redes (Figura 1) cuyas diferencias se encuentran enmascaradas, puesto que todas las computadoras que se conectan a este utilizan los protocolos de Internet para comunicarse una con otra, así una computadora conectada a una red Ethernet puede comunicarse con otra computadora conectada a una red Token Ring, basta con que se haga una implementación de los protocolos de Internet para cada una de esas redes.

Extensibilidad y Apertura: La extensibilidad y la apertura son dos características de la computación en malla, debido a que están ampliamente ligadas una con la otra. Algunos autores dicen que un sistema abierto debe de ser extensible y otros sostienen que un sistema extensible puede ser etiquetado como un sistema abierto. De cualquier manera, lo que es importante saber y tener en cuenta es que un sistema distribuido debe de contar con ambas características.

Los recursos se asignan y eliminan de forma dinámica: Debe estar preparado para hacer uso de los recursos y por lo mismo dejar de utilizar en el caso de que este se requiera del total procesamiento por parte del usuario que lo comparte.

El procesamiento en grandes volúmenes de datos permite un establecimiento de los modelos matemáticos, la simulación de eventos, y en algunos casos y con reservas, predicciones científicas sobre lo que puede suceder.

Como se dice de manera general si dos cabezas piensan más que una de la misma manera dos procesadores como el CPU y la Unidad de procesamiento podrá producir más en cantidad y en el menor tiempo posible los cálculos y procesos digitales que se requieran. Con varios CPU juntos se puede procesar rápidamente la información y aprovechar el poder que nos brida hoy en día la computación es a ello a lo que se conoce como computación en malla, la misma se ha venido operando desde varios años y obteniendo mejores prácticas, programas de control y varias técnicas para generar el mayor provecho de los elementos que trabajan en operación.

La Computación en malla hace referencia a cualquier evento en el cual se maneja un sistema en una red de computadoras y trata de describir las tendencias hacia la funcionalidad distribuida: sistemas distribuidos, procesamiento distribuido, bases de datos distribuidas y cualquier otro término computacional que sea distribuido.

Se puede decir entonces, que la Computación en malla se refiere a los servicios que provee un Sistema de Computación Distribuido. Una de las primeras caracterizaciones de un Sistema Distribuido fue realizada por Enslow, ya en 1978, que le atribuye las siguientes propiedades: o Está compuesto por varios recursos informáticos de propósito general, tanto físicos como lógicos, que pueden asignarse dinámicamente a tareas concretas. (Enslow, 1978)

Los sistemas se encuentran distribuidos físicamente por lo que funcionan correctamente en un área de comunicaciones, existe un nivel operativo que se integra al control de componentes, el hecho que se basa en la distribución es físicamente transparente, de manera que permiten que los servicios se puedan incorporar especificando simplemente el nombre y no así la localización.

El funcionamiento de los recursos físicos y lógicos está caracterizado por una autonomía coordinada (Aroquipa, 2004)

2.2.2 Recursos

Los computadores modernos incorporan en su hardware múltiples elementos de procesamiento (cores o núcleos) que permiten el desarrollo de algoritmos paralelos para sistemas con memoria compartida. Gracias al surgimiento y al continuo desarrollo de la tecnología CMP (Chip Multiprocesador) grupos de diferentes áreas del conocimiento en todo el mundo han estado diseñando algoritmos paralelos para sacar provecho de las bondades que proveen las arquitecturas multicore.

Dado que no hay uniformidad de diseño en esta tecnología (por ejemplo: número de núcleos, jerarquías de memorias caché y redes de interconexión interna), el diseño de los algoritmos paralelos se hace dependiente de la arquitectura, lo que presenta un gran desafío para los diseñadores y desarrolladores de algoritmos paralelos de refinamiento/des refinamiento de mallas geométricas sobre arquitecturas multicore (Rodriguez, 2013)

2.2.3 Infraestructura

Entre las infraestructuras de software existentes se decide por la más utilizada en open source, terracota, misma que se encuentra escrita en java, uno de los lenguajes más populares actualmente, terracota permite la ejecución de un solo código único que se puede ejecutar independientemente en un clúster convencional. Consta de 2 elementos principales:

Los nodos clientes.- Hacen referencia a un proceso de Java dentro del clúster, los mismos ejecutan una máquina virtual estándar cargando terracota mediante sus librerías cuando esta se inicia.

Terracota Server Array.- Provee una inteligencia tolerante a fallos, además de instalaciones de alto rendimiento, esta 100% compuesta de procesos en JAVA.

Terracota utiliza la tecnología Network-Attached Memory (NAM) para permitir realizar el clustering con Máquinas Virtuales Java, Terracota es la Máquina Virtual Java para la aplicación a ejecutar dentro del clúster, mientras que para la verdadera Máquina Virtual Java, Terracota es la aplicación (Hidalgo, 2009)

2.2.4 Metodología PPDIOO

La metodología PPDIOO permite formalizar el ciclo de vida mediante fases, cada un cumple con una función específica y se relacionan con su antecesora y predecesora mediante la siguiente figura se puede identificar el ciclo de vida de la red según la metodología PPDIOO.

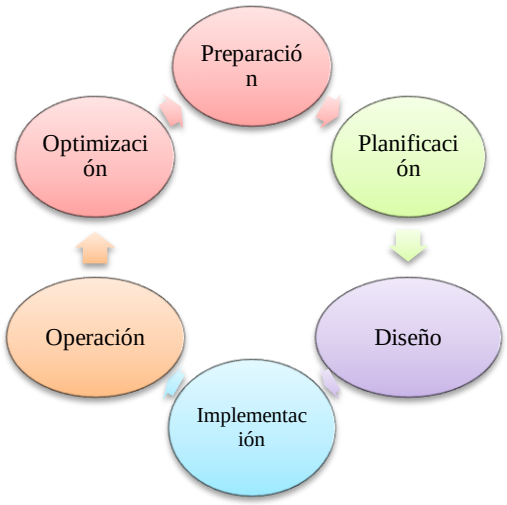


Figura 1-2 Metodología PPDIOO

Elaborado por: Marco Gavilanes

Definida por CISCO y orientada a la creación de redes de datos, se formaliza con 6 fases principales a las que se debe su nombre (CISCO, 2010):

- **Fase de Preparación:** Esta fase crea un caso de negocios para establecer una justificación financiera para la estrategia de red. La identificación de la tecnología que soportará la arquitectura.
- **Fase de Planeación:** Identifica los requerimientos de red realizando una caracterización y evaluación de la red, realizando un análisis de las deficiencias contra las buenas prácticas de arquitectura. Un plan de proyecto es desarrollado para administrar las tareas, parte responsables, hitos y recursos para hacer el diseño y la implementación. Este plan de proyecto es seguido durante todas las fases del ciclo.
- **Fase de Diseño:** El diseño de la red es desarrollado basado sobre los requerimientos técnicos y de negocios, obtenidos desde las fases anteriores. Esta fase incluye diagramas de red y lista de equipos. El plan de proyecto es actualizado con información más granular para la implementación. Después de esta fase aprobada empieza la implementación.
- **Fase de Implementación:** Nuevo equipamiento es instalado y configurado en esta fase. El plan de proyecto es seguido durante esta fase. Los cambios deben ser comunicados en una reunión de control de cambios, con la necesaria aprobación para proceder. Cada paso en la implementación debe incluir una descripción, guía de implementación, detallando tiempo estimado para implementar, pasos para rollback en caso de falla e información de referencia adicional.
- **Fase Operativa:** Esta fase mantiene el estado de la red día a día. Esto incluye administración y monitoreo de los componentes de la red, mantenimiento de ruteo, administración de actualizaciones, administración de performance, e identificación y corrección de errores de red. Esta fase es la prueba final de diseño.
- **Fase de Optimización:** Esta fase envuelve una administración pro-activa, identificando y resolviendo cuestiones antes que afecten a la red. Esta fase puede crear una modificación al diseño si demasiados problemas aparecen, para mejorar cuestiones de performance o resolver cuestiones de aplicaciones.

Entre las principales ventajas que se obtienen por la implementación de esta metodología cabe mencionar (CISCO, 2010):

- Baja el costo total de propiedad por validación de requerimientos de tecnología y planeamiento para cambios de infraestructura y requerimientos de recursos.
- Incrementa la disponibilidad de la red por la producción de un sólido diseño de red y validaciones en las operaciones.
- Mejora la agilidad de negocios estableciendo requerimientos y estrategias tecnológicas.

- Velocidad de acceso para aplicaciones y servicios, mejorando disponibilidad, fiabilidad, seguridad, escalabilidad y performance.

2.2.5 Ataques de Red

Tomando en cuenta el manual OWASP, se detallan a continuación las vulnerabilidades más frecuentes encontradas en sistemas de información.

Basada en los protocolos TCP/IP, la comunicación entre sistemas que conforman una red permite el tráfico de datos en la misma, por lo cual se convierte en un punto de interés para los atacantes informáticos por el volumen de datos que se maneja y la parcialmente simplicidad que conlleva el acceso a la misma.

Entre los ataques más comunes que se dan a infraestructuras de datos tenemos los que tienen como fin obtener nuestra información y aquellos que directamente quieren evitar nuestro funcionamiento, por lo cual mencionaremos 2 tipos de ataque:

DDOS.- Siglas de Distributed Denial of Service, se basa en la saturación de un elemento hardware para que este no pueda atender más peticiones, comúnmente se lo realiza para dejar sin memoria RAM a un equipo aunque también se puede destinar a colapsar una red completa.

Según, (Luz, 2010) el DDOS es:

> "Un servidor por inundación de ancho de banda, el ataque ha de ser distribuido (DDoS) ya que actualmente los servidores tienen un gran ancho de banda. Además, sería muy fácil detectarlo ya que sólo sería desde una IP (en DoS no distribuido). El ancho de banda del ataque se debe acercar al ancho de banda máximo de dicho servidor para colapsarlo."

Entre las herramientas más populares para este tipo de ataques cabe mencionar a la que provee el famoso grupo conocido como anonymous, loic, en especial su versión para Linux loiq, la cual a nivel doméstico y para pruebas de este tipo es sumamente eficiente y didáctica (NewEraCracker, 2016).

Se mencionan también otras herramientas como HPING, HOIC, REFREF las cuales hacen lo mismo, aunque de maneras diferentes (B-Secure, 2015).

Sniffing.- También conocido como captura de paquetes en redes, destaca eavesdropping o husmeo y el análisis de tráfico.

El eavesdropping, al igual que por análisis de tráfico, se hace uso de una tarjeta o adaptador para redes inalámbricas que trabaje sobre el mismo rango de frecuencias y use el mismo método de transmisión que emplea la red inalámbrica objetivo, permitiendo así que el agresor pueda capturar el tráfico transmitido sobre esta red (Mendoza, 2007).

Con la ayuda de alguna aplicación sniffer que tenga soporte para estos tipos de adaptadores de red, en eavesdropping, el agresor puede explorar transmisiones y descubrir redes inalámbricas sin ningún esfuerzo. Algunos sniffers cuentan con herramientas para romper la seguridad que los mecanismos de encriptación ofrecen en las transmisiones inalámbricas. Estas herramientas consisten principalmente en descifrar las claves que utilizan los mecanismos de encriptación a través de la aplicación de algoritmos estadísticos al tráfico de la red objetivo. Con la obtención de estas claves, los datos transmitidos pueden ser descifrados fácilmente permitiendo que la información llegue a ser legible para el agresor (Mendoza Acevedo, 2005).

DNS Spoofing: El DNS pretende provocar un direccionamiento erróneo en los equipos que se encuentran afectados, ello se debe a una traducción errónea de los nombres de dominio a direcciones IP, facilitando de tal manera la redirección de cada uno de los usuarios de los sistemas afectados hacia páginas Web falsas o bien interceptación de los mensajes de correo electrónico.

Para ello, en este tipo de ataque los intrusos consiguen que un servidor DNS legítimo acepte y utilice información incorrecta obtenida de un ordenador que no posee autoridad para ofrecerla. De este modo, se persigue "inyectar" información falsa en el base de datos del servidor de nombres, procedimiento conocido como "envenenamiento de la caché del servidor DNS", ocasionando con ello serios problemas de seguridad, como los que se describen de forma más detallada a continuación.

IP Spoofing: Los ataques de suplantación de la identidad presentan varias posibilidades, siendo una de las más conocidas la denominada "IP Spoofing" ("enmascaramiento de la dirección IP"), mediante la cual un atacante consigue modificar la cabecera de los paquetes enviados a un determinado sistema informático para simular que proceden de un equipo distinto al que verdaderamente los ha originado. Así, por ejemplo, el atacante trataría de seleccionar una dirección IP correspondiente a la de un equipo legítimamente autorizado para acceder al sistema que pretende ser engañado.

2.2.6 Middleware

Es un servicio que se basa en el conjunto de software distribuido que existe entre la aplicación y el sistema operativo y los servicios de la red en un nodo del sistema en la red. Es un Software de conectividad que consiste en un conjunto de servicios que permiten interactuar a múltiples procesos que se ejecutan en distintas máquinas a través de una red. Ocultan la heterogeneidad, abstrae la complejidad subyacente y proveen de un modelo de programación conveniente para los desarrolladores de aplicaciones (Sosa, 2014).

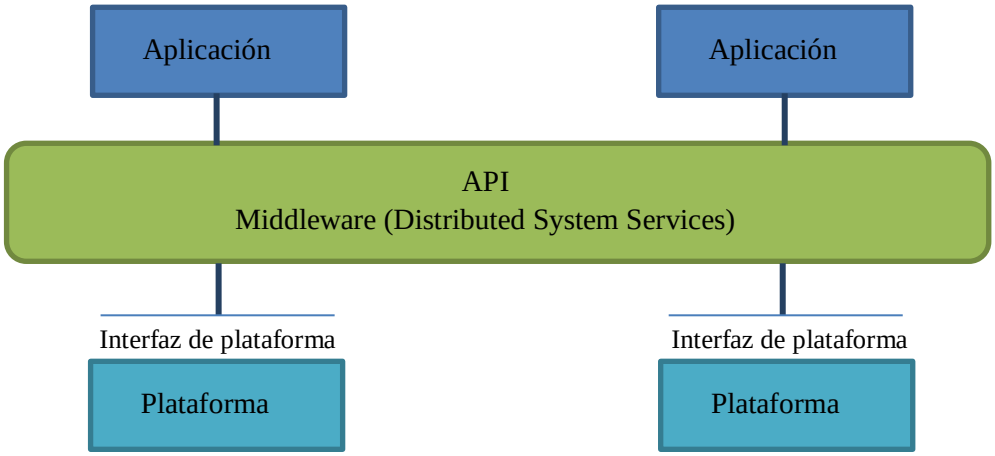


Figura 2-2 Middleware - Utilización

Elaborado por: Marco Gavilanes

(Sosa, 2014), menciona que middleware es una tecnología de apoyo para el desarrollo, despliegue, ejecución e interacción de aplicaciones. Un Middleware se puede definir como una capa intermedia entre dos tecnologías aisladas para lograr la interacción entre ellas. Este modelo de programación consigue la integración puesto que logra hacer transparente las funcionalidades de una tecnología con respecto a la otra.

Las capas intermedias middleware han evolucionado desde ofrecer funcionalidades básicas, hasta ocultar y envolver minuciosas capacidades en sofisticados sistemas, estos sistemas se encuentran equipados para administrar recursos y ofrecer al programador una Interfaz de Programación(API) para el desarrollo de nuevas aplicaciones.

2.2.6.1 Características del Middleware

- Ser confiable y disponible
- Ampliación en la capacidad sin perder su función
- Aplicación y sistema operativo

- Orientado a mensajes

Procesa computación distribuida La taxonomía o clasificación del software middleware puede ser explicada en dos grandes categorías: una de ellas es la integración y otra de aplicación. Estas poseen a su vez diferentes clases a continuación se determina lo siguiente:

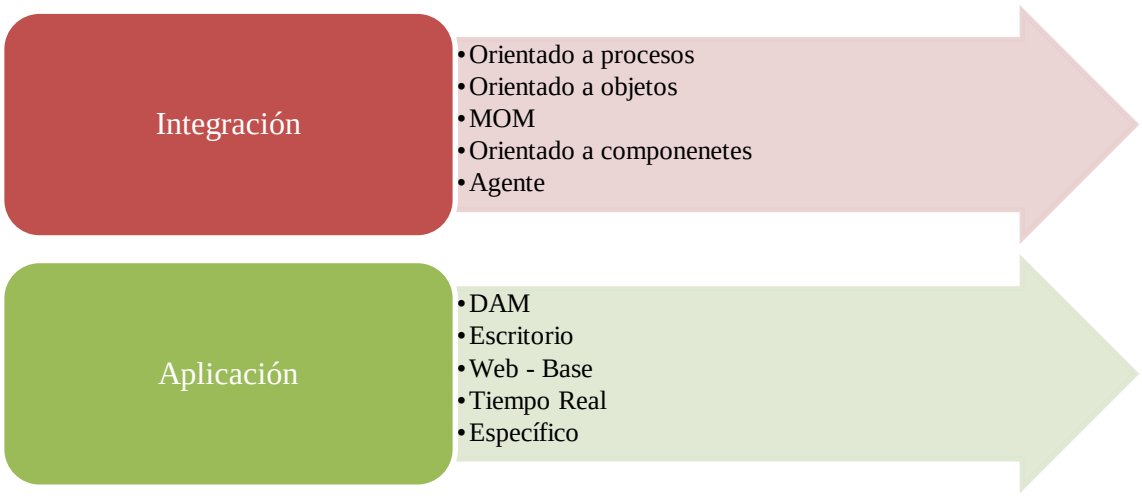


Figura 3-2 Clasificación del Software middleware

Elaborado por: Marco Gavilanes

a) Orientados a procedimientos o procesos

Los middlewares que son orientados a procesos, utilizan una comunicación sincronizada un ejemplo de ellos es el teléfono. Entre las características de estos, es que utilizan el client stub y el server skeleton. El client stub convierte la petición en un mensaje que es mandado al servidor; luego el server skeleton recibe el mensaje, lo convierte en la petición y llama a la aplicación del servidor donde ésta es procesada.

Terminado el procesamiento, ocurre el proceso inverso. El client stub verifica los errores, envía los resultados al software que inició la petición y entonces suspenden el proceso.

Ventajas de Middleware

Entre las ventajas de este middleware es que usan un tipo estándar en nombres de servicios y procesos remotos, pueden retornar respuesta aún con problemas en la red y pueden manejar múltiples tipos de formatos para datos y niveles heterogéneos de sistemas de servicio.

Desventajas de Middleware

Las desventajas son que no poseen escalabilidad, no pueden retornar la información a un programa diferente del que realizó la solicitud (reflexión) y poseen procesos muy rígidos.

b) Orientados a objetos

Soportan pedidos de objetos distribuidos. La comunicación entre los objetos puede ser sincronizada, sincronizada diferida o no sincronizada. Soportan múltiples pedidos similares realizados por múltiples clientes en una transacción. De acuerdo a (CISCO, 2011.) la forma de operar es:

- El objeto cliente llama a un método lógico para obtener un objeto remoto.
- Un ORB Proxy (también conocido como stub) pone en orden la información (marshalling o serialización) y la transmite a través del agente (broker).
- El agente actúa como punto medio y contacta con diversas fuentes de información, obtiene sus referentes IDs, recolecta información y, en ocasiones, la reorganiza.
- El proxy remoto (también conocido como skeleton) desordena (unmarshalling o deserialización) la información que le llega del agente y se la pasa al objeto servidor.
- El objeto servidor procesa la información y genera un resultado que es devuelto al cliente siguiendo los pasos inversos.
- Las ventajas son que permiten generar reflexión y escalabilidad, que opera con múltiples tipos de información y estados y que soporta procesos múltiples. Las desventajas consisten en obtener la existencia de vínculos antes de la ejecución y de un código contenedor para algunos sistemas heredados.

c) Orientados a mensajes

Se pueden dividir en dos tipos, espera y publicación/suscripción. El paso de espera se puede dividir en mensaje y espera. El paso de mensaje inicia con que la aplicación envía un mensaje a uno o más clientes, con el MOM del cliente.

El servidor MOM, recoge las peticiones de la cola (Message Broker) en un orden o sistema de espera predeterminado. Los actos del servidor MOM son como un router y usualmente no interactúan con estas. El MOM de publicación y suscripción actúa de manera ligeramente diferente, es más orientado a eventos. Si un cliente quiere participar por primera vez, se une al bus de información. (Mendoza Acevedo, 2005).

Dependiendo de su función, si es como publicador, suscriptor y ambas, este registra un evento. El publicador envía una noticia de un evento al bus de memoria. El servidor MOM envía un anuncio al subscriptor registrado cuando la información está disponible.

d) Orientados a componentes

Un componente es un «programa que realiza una función específica, diseñada para operar e interactuar fácilmente con otros componentes y aplicaciones». El middleware en este caso en una configuración de componentes. Los puntos fuertes de este middleware es que es configurable y reconfigurable. La reconfiguración se puede realizar en tiempo de ejecución, lo que ofrece una gran flexibilidad para satisfacer las necesidades de un gran número de aplicaciones.

Los agentes son un tipo de middleware que posee varios componentes:

- Entidades. Pueden ser objetos o procesos.
- Medios de comunicación. Pueden ser canales, tuberías, etc.
- Leyes. Identifican la naturaleza interactiva de los agentes. Pueden ser la sincronización o el tipo de esquema.

Las ventajas de los middlewares agentes son que la capacidad de éstos para realizar una gran cantidad de tareas en nombre del usuario y para cubrir una amplia gama de estrategias basadas en el entorno que les rodea. Sin embargo, su implementación es complicada debido a la complejidad y dificultades dadas por las operaciones que manejan.

2.2.6.2 Middleware orientado a mensajes

Los datos se intercambian por el paso de mensajes y o colas de mensajes que apoyan las interacciones sincrónicas entre los procesos de computación distribuida. El sistema MOM asegura la entrega de mensajes de manera inmediata para realizar el uso de colas confiables

proporcionando de esta manera el directorio, la seguridad y los servicios administrativos necesarios para apoyar la mensajería (Mendoza, 2007).

2.2.6.3 Middleware de las tecnologías aplicadas

Oracle es un sistema de gestión de base de datos objeto relacional, es la base de infraestructuras de aplicaciones que tiene mayor aceptación hoy en día, permite a las empresas crear y utilizar aplicaciones empresariales agiles, correctas e inteligentes y al mismo tiempo potenciar al máximo la eficiencia informática aprovechando plenamente la arquitectura moderna de hardware y software.

Middleware basado en la web: En este tipo de middleware se asiste al usuario mediante la navegación web, mediante el uso de interfaces que permiten encontrar las páginas que requieren y necesitan de igual manera detecta cambios de interés del usuario este se basa en el historial de búsquedas.

Provee de un servicio de identificación para un gran número de aplicaciones y comunicación entre procesos independiente del sistema operativo, protocolo de red y plataforma de hardware.

Los middlewares que se encuentran fuertemente unidos a la red se llaman servidores de aplicaciones, ya que mejoraran el rendimiento, disponibilidad, escalabilidad, seguridad, recuperación de información, y soportan la administración colaborativa y su uso.

Los middlewares pueden contactar directamente a la aplicación ganando mejor comunicación entre el servidor y el cliente. Otros servicios importantes dados por este tipo de middleware son servicios de directorios, correos electrónicos, cadenas de suministros de gran tamaño, accesos remotos a información, descarga de archivos, accesos a programas y acceso a aplicaciones remotas.

Middleware a tiempo real: Mediante este software se puede generar información a tiempo real y es caracterizada porque la información que se detalla es la correcta en un instante, pero de igual manera puede no serlo para otro. El middleware en tiempo real soporta cada una de las peticiones sensibles en el tiempo preciso y cada una de las políticas de planificación. Esto se lo realiza mediante servicios que mejoran la eficiencia de las aplicaciones de usuario. Así también los middlewares se dividen en diferentes aplicaciones entre ellas se encuentran:

- Aplicación de base de datos en tiempo real

- Sensor de procesamiento
- Transmisión de información

La información que pasa a través de un middleware en tiempo real se ha incrementado dramáticamente con la introducción de internet, redes inalámbricas, y las nuevas aplicaciones basadas en la difusión.

Las ventajas de este tipo de middleware son que proveen un proceso de decisión que determina el mejor criterio para resolver procesos sensibles al tiempo, y la posibilidad de ayudar a los sistemas operantes en la localización de recursos cuando tienen tiempos límites de operación.

El middleware multimedia es una rama mayor en el middleware en tiempo real. Estos pueden manejar una gran variedad de información. También pueden ser textos, imágenes de todo tipo (GPS, imágenes, etc.), procesadores de lenguajes naturales, música y video. La información debe ser recopilada, integrada y entonces enviada al usuario sensible del tiempo. Los dispositivos multimedia pueden incluir una mezcla de dispositivos tanto físicos (parlantes, cámaras, micrófono) como lógicos.

2.2.6.4 Aplicaciones de Middleware

La arquitectura del sistema está compuesta mediante un middleware que haga de comunicador entre aplicaciones móviles y la instancia de otros aparatos dispositivos. El middleware que conforma esta implementado por la parte del SOAP y el mismo tiene el modelo de datos y la lógica de datos. Los servicios que brinda middleware proporcionan un conjunto más funcional de la API lo que permite que:

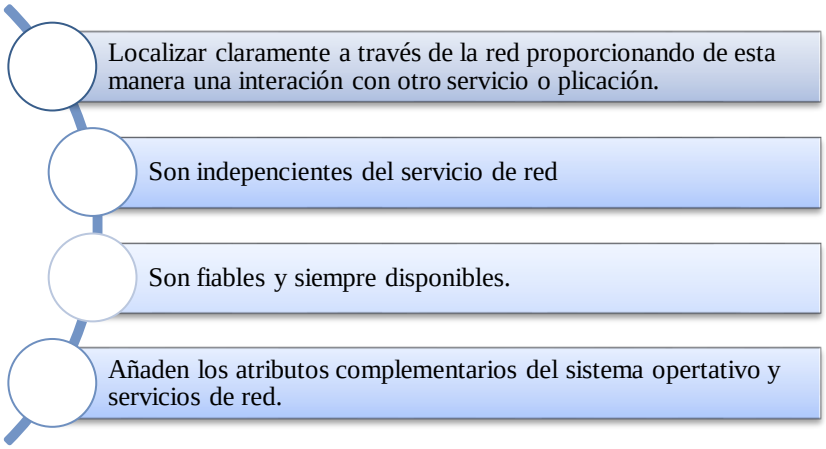


Figura 4-2 Aplicación de Middleware

Elaborado por: Marco Gavilanes

Los modelos de clusters más conocidos por su amplia utilización en función del middleware son: - NUMA (Non-Uniform Memory Access). - PVM (Parallel Virtual Machine). - MPI (Message Pass Interface), que es la que nosotros utilizamos para desarrollar nuestro trabajo.

Las máquinas de tipo NUMA, tienen acceso compartido a la memoria donde pueden ejecutar su código de programa. En el kernel de Linux hay ya implementado NUMA, que hace variar el número de accesos a las diferentes regiones de memoria. 18 MPI y PVM son herramientas ampliamente utilizadas, y son muy conocidas por aquellos que entiende de súper computación basada en GNU/Linux.

MPI es el estándar abierto de bibliotecas de paso de mensajes. MPICH es una de las implementaciones más usadas de MPI; tras MPICH se puede encontrar LAM, otra implementación basada en MPI, que también son bibliotecas de código abierto. PVM es un middleware semejante a MPI, ampliamente utilizado en clusters Beowulf.

PVM habita en el espacio de usuario, y tiene la ventaja de que no hacen falta modificaciones en el kernel de Linux. Básicamente, cada usuario con derechos suficientes puede ejecutar PVM.

2.2.6.5 Globus Toolkit

Globus Toolkit es un middleware empleado, por defecto, en la computación en grid. Es una herramienta open source que provee:

- Gestión de recursos (GRAM)
- Servicio de Monitoreo y Descubrimiento (MDS)
- Servicios de seguridad (GSI)
- Movimiento y gestión de datos GridFTP

Globus Toolkit es utilizada en ambientes de simulación y proyectos de computación en malla o grid, especialmente en redes computacionales pequeñas o medianas. En el entorno local, por ejemplo, en el proyecto "Infraestructura basada en Globus Toolkit para dar soporte a repositorios distribuidos de imágenes médicas" sus autores expresaron lo siguiente:

> Como herramienta de software se utiliza el GSI de Globus como lo hace MEDICUS o mantis GRID, debido a que es una herramienta ampliamente utilizada, catalogada por muchos como el estándar de facto para la construcción de plataformas grid (Guillermo, y otros, 2015, p.185).

Por otra parte, en el proyecto "Desarrollo de una aplicación GRID usando globus toolkit 4" sus autores incluyeron dentro de sus conclusiones:

> El principio detrás de la computación Grid es la interacción entre organizaciones. GT4 da facilidades para la integración de las organizaciones con WS-GRAM (Villacreses & Caiza, 2007, p.78).

Aunque existen varios middlewares para computación en grid, como por ejemplo: gLite, UNICORE, XtreemOS; sin embargo no existe un estudio comparativo concluyente al respecto de ellas y su aplicación en estudios experimentales de mediana complejidad. De manera general varios autores (como los citados en los proyectos anteriores) sugieren el uso de Globus Toolkit debido a que pueden emplearse en este tipo de proyectos, así como al soporte documental y de paquetes (software) existente (escaso, pero mejor y más abundante que el existente para otros middleware open source). En base a este análisis se decidió emplear Globus Toolkit en el presente proyecto.

Tabla 1-2 Revisión comparativa – Sitios Web Oficial

	gLite	**UNICORE**	**XtreemOS**	**Globus Toolkit**
Open source	Sí	Sí	Sí	Sí
Soporte documental	Bajo	Bajo	Bajo	Medio
Soporte hardware	Bajo	Medio	Bajo	Alto
Tamaño implementación	Pequeños y medianos	Supercomputadoras y clusters	Pequeños y medianos	Pequeños y medianos

Realizado por: Marco Gavilanes

Como puede observarse en la **tabla 1** Globus Toolkit es el único middleware (en relación a otros con los cuales se comparó) que cumple los criterios priorizados para la selección de middlware.

2.2.7 Vulnerabilidades y Gestión de Riesgos

2.2.7.1 OWASP

Los atacantes pueden, potencialmente, utilizar diferentes rutas a través de su aplicación para perjudicar un negocio u organización. Cada uno de estos caminos representa un riesgo que puede o no ser suficientemente grave como para merecer atención:

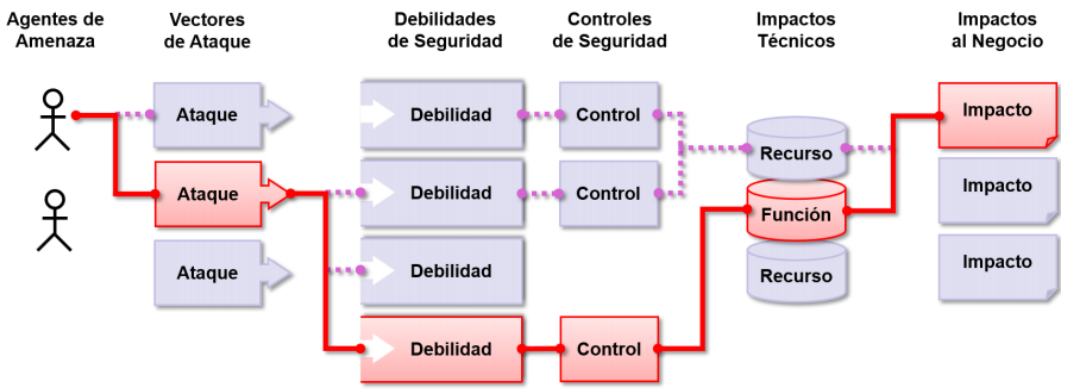


Figura 5-2 Valoración del riesgo OWASP

Fuente: (OWASP, 2017)

Algunas veces, estos caminos son fáciles de encontrar y explotar, mientras que otras son extremadamente difíciles. A fin de determinar el riesgo para una organización, se puede evaluar la probabilidad asociada a cada agente de amenaza, vector de ataque, debilidad de seguridad y combinarlo con una estimación del impacto técnico y de negocio para una organización. Juntos, estos factores determinan su riesgo general.

OWASP, a nivel de aplicación, es una herramienta libre que permite optimizar la seguridad de los sistemas, y está dirigida a todos los interesados en aplicar pruebas para determinar vulnerabilidades.

El OWASP Top 10 se enfoca en identificar los riesgos más críticos para un amplio tipo de organizaciones. Para cada uno de estos riesgos, se proporciona información genérica sobre la probabilidad y el impacto técnico, utilizando el siguiente esquema de evaluación, basado en la Metodología de Evaluación de Riesgos de OWASP.

Agente de Amenaza	Explotabilidad	Prevalencia de Vulnerabilidad	Detección de Vulnerabilidad	Impacto Técnico	Impacto de Negocio
Específico de la Aplicación	Fácil 3	Difundido 3	Fácil 3	Severo 3	Específico del Negocio
	Promedio 2	Común 2	Promedio 2	Moderado 2	
	Difícil 1	Poco Común 1	Difícil 1	Mínimo 1	

Figura 6-2 Probabilidad e impacto del riesgo - OWASP

Fuente: (OWASP, 2017)

Dicha metodología se basa en los siguientes procesos generales:

- Paso 1: Identificar un riesgo
- Paso 2: Identificar factores para estimar la verosimilitud
- Paso 3: Identificar factores para estimar el impacto
- Paso 4: Determinar la gravedad del riesgo
- Paso 5: Decidir qué arreglar
- Paso 6: Personalizar el modelo de calificación de riesgo

2.2.7.2 Magerit

Magerit responde a la normativa ISO 3100 respecto a lo que se denomina "Proceso de Gestión de los Riesgos", sección 4.4 ("Implementación de la Gestión de los Riesgos") dentro del "Marco de Gestión de Riesgos". MAGERIT implementa el Proceso de Gestión de Riesgos dentro de un marco de trabajo para que los órganos de gobierno tomen decisiones teniendo en cuenta los riesgos derivados del uso de tecnologías de la información.

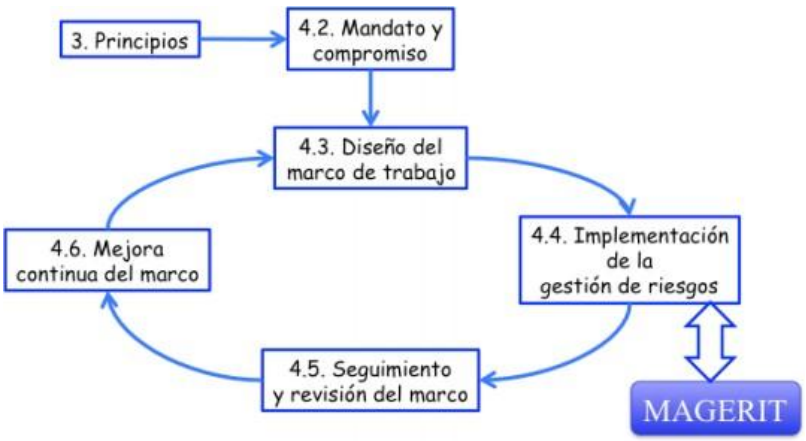


Figura 7-2 ISO 3100 – Marco de trabajo gestión de riesgos

Fuente: (Gobierno de España, 2012)

MAGERIT (Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información) mide la Vulnerabilidad por la frecuencia histórica cuantitativa de la materialización de la

Amenaza, cuando es factible (fiabilidad de un componente hardware, número de fallos de software); o bien por la potencialidad cualitativa de dicha materialización, cuya primera aproximación lleva a emplear una escala vista en las Amenazas potenciales (consideradas ahora reales, o sea agresiones).

En las "Directrices de la OCDE para la seguridad de sistemas y redes de información-Hacia una cultura de la seguridad", en su principio 6 dice:

6) Evaluación del riesgo. Los participantes deben llevar a cabo evaluaciones de riesgo

Existen varias aproximaciones que sirven para analizar los riesgos que pueden sufrir sistemas y las tecnologías de la información y comunicación: guías formales, aproximaciones metódicas y herramientas de soporte. Todas ellas tienen como finalidad el saber cuan seguros o inseguros son los sistemas. Existen muchos elementos que hay que considerar para lograr tener buenos resultados. Es por ello que Magerit está basado sobre una aproximación metódica que no deja lugar a la improvisación, ni dependa de la arbitrariedad del analista.

Proceso de Gestión de Riesgos

Magerit sugiere un proceso metodológico para la gestión de riesgos, mismo que se muestra en el siguiente gráfico:

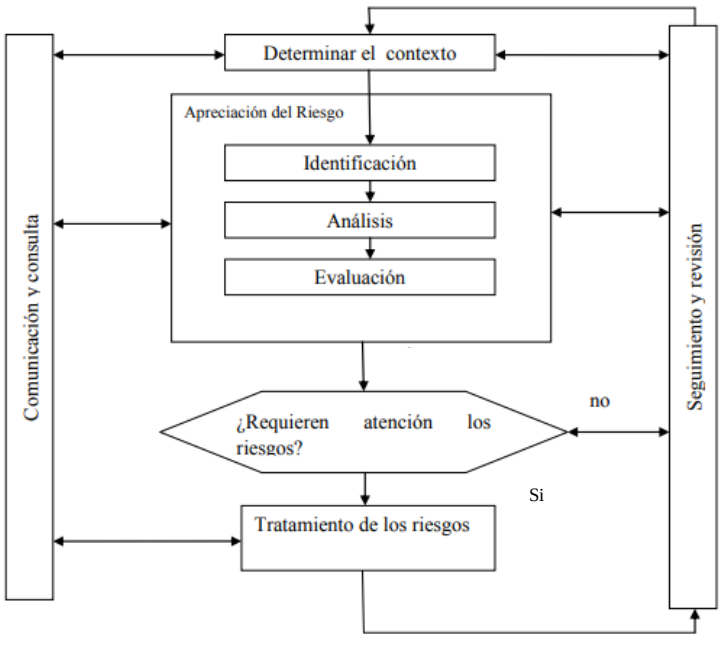


Figura 8-2 Proceso de Gestión de Riesgos

Fuente: (Gaona, 2013)

Determinación del contexto: Lleva a una determinación de los parámetros y condicionantes externos e internos que permiten delimitar una política que se seguirá para gestionar los riesgos.

Identificación de los riesgos: Busca una relación de los posibles puntos de peligro. Lo que se identifique será analizado en la siguiente etapa.

Análisis de riesgos: Busca calificar los riesgos identificados, bien cuantificando sus consecuencias ya sean cuantitativamente o cualitativamente. De una u otra forma, como resultado del análisis tendremos una visión estructurada que nos permita centrarnos en lo más importante.

Evaluación de los riesgos: Aquí entran factores de percepción, de estrategia y de política permitiendo tomar decisiones respecto de que riesgos se aceptan y cuáles no, así como de en qué circunstancias podemos aceptar un riesgo o trabajar en su tratamiento.

Tratamiento de los riesgos: Recopila las actividades encaminadas a modificar la situación de riesgo.

Comunicación y consulta: Lo que se desea alcanzar un equilibrio entre la seguridad y productividad.

Seguimiento y revisión: Cuando se ha terminado de realizar el Análisis de Riesgos los resultados que arroje esta investigación lo más conveniente, será de poner en práctica para evitar incidentes dentro de su entorno.

2.2.8 Prueba de pares relacionados en Wilcoxon

Es una prueba no paramétrica que comprara las muestras relacionadas, debe cumplir las siguientes características:

- Es libre de curva, no necesita una distribución especifica
- Nivel ordinal de la variable dependiente.

Se utiliza para comparar dos mediciones de rangos estos son medianos y de igual manera determinan que la diferencia no se deba al azar que la diferencia sea estadísticamente significativa.

Aplicaciones de la prueba de Wilcoxon

- Trabaja con datos de tipo ordinal
- Establece diferencias de magnitudes (+y-)
- Dos muestras aparentadas
- Establece las diferencias
- Con muestras grandes (>25) se intenta lograr la distribución normal se utiliza la prueba Z.

La prueba es posible utilizarla en lugar de la prueba paramétrica t de Student para una muestra en la cual se necesita probar una hipótesis en relación con un parámetro que refleja una tendencia central. Tal prueba no paramétrica es conocida como prueba de rangos con signos de Wilcoxon, la cual se maneja cuando se tienen datos medidos a un nivel más alto que una escala ordinal.

Cuando se violan las suposiciones de la prueba t, la prueba de Wilcoxon, que hace menos suposiciones y menos estrictas, es pertinente usarla para detectar las diferencias significativas. Las suposiciones necesarias para efectuar esta prueba de Wilcoxon son las siguientes:

- Los datos obtenidos deben ser medidos a un nivel más alto que el de escala ordinal.
- El fenómeno aleatorio de interés genere una variable continua.
- Los datos deben ser seleccionados en forma aleatoria e independiente.
- La distribución de las diferencias entre los datos observados y la mediana hipotética debe ser aproximadamente simétrica.

La prueba de Wilcoxon, de rangos con signo de una muestra, considera no sólo si un valor observado es o no mayor (menor) que la mediana hipotética, sino también qué tan grande (pequeña) es.

La prueba de Wilcoxon fue diseñada para detectar cualquier clase de diferencia entre dos grupos; algunas de ellas son: ubicación, dispersión, forma, o las tres. Es posible usarla cuando se ha logrado una medición de la información en cuando menos escala ordinal y se desea probar que dos muestras mutuamente independientes se han tomado o no de la misma población o de poblaciones idénticas.

Los únicos supuestos necesarios para aplicar la prueba son los siguientes:

- Para evitar empates que la variable aleatoria de interés sea continua.

- Que los datos a recopilar cuando menos tengan una escala ordinal de medición, tanto entre como dentro de las dos muestras.
- Que ambas muestras sean elegidas en forma aleatoria e independiente de sus respectivas poblaciones.

2.2.9 Benchmarking

El Benchmarking puede definirse como un estudio comparativo de las diversas áreas o sectores de la empresa con el fin de mejorar su funcionamiento. En el área de sistemas puede considerarse como una herramienta a ser empleada para analizar el grado de funcionalidad de los sistemas, o las ventajas y desventajas comparativas entre varias tecnologías, con el fin de mantenerse a la vanguardia tecnológica (Boxwell, 2008)

Respecto a la evaluación comparativa, se inicia estableciendo parámetros comunes y/o comparativos sobre los cuales establecer cualitativa y cuantitativamente una calificación o ponderación. Para la evaluación, es importante tener un conocimiento intermedio – alto de las empresas / tecnologías a comparar. Es así que el benchmarking es una herramienta válida para llevar a cabo procesos de autoevaluación. (Gurut & Velasco, 2010)

Por lo que en nuestro estudio aplicamos el Benchmarking para realizar un análisis comparativo de las herramientas a ser utilizadas en nuestro estudio.

CAPÍTULO III

3. METODOLOGÍA DE INVESTIGACIÓN

3.1 Diseño de la Investigación

Dada la naturaleza del estudio, este se define como Cuasi-Experimental, dado que el ambiente de pruebas, así como el tráfico a analizar, no son generados de manera aleatoria, sino previamente definidos por el investigador, además se manipula una variable independiente y evaluación de su correspondiente efecto en la variable dependiente.

Su validez se alcanza a medida que se logren capturar paquetes entendibles, así como estresar la red o sus miembros a niveles que los dejen inoperables, esto para medir la seguridad y el rendimiento correspondientemente.

En cuanto a Software, todo el ambiente es definido utilizando las herramientas open source que hayan demostrado ser las más pertinentes para nuestro caso y considerando que el estudio se enfoca a redes domésticas, en cuanto a hardware, NO se utilizan dispositivos de red costosos y especializados en seguridad, se utilizan dispositivos domésticos de marcas y modelos genéricos.

3.2 Tipo de Investigación

En la investigación se considera que el tipo de estudio que se va a realizar es una **investigación descriptiva y aplicada**, ya que se utilizara el conocimiento para realizar un estudio comparativo de los ataques informáticos realizados hacia la red utilizada para la computación en malla frente a las herramientas diseñadas para monitorear, detectar, analizar y prevenir dichos ataques.

3.3 Métodos

Para este proyecto se utilizarán los siguientes métodos de investigación:

Método Científico: Conocido como el método experimental de prueba y error, será de utilidad para la selección de herramientas como de configuraciones a implementar en el ambiente de pruebas, así como para la selección y uso de las herramientas de prueba de seguridad de la red,

ya que las ideas, conceptos, y teorías expuestas en este proyecto de tesis son verificables como válidos. Se ha realizado las siguientes consideraciones para esta investigación:

- Se plantea la investigación en base a las vulnerabilidades existentes en las redes domésticas y las aplicaciones que las utilizan para computación en malla.
- Se trazan los objetivos de la investigación que permitirán resolver el problema de riesgos existentes en el entorno de estudio.
- Se justifican los motivos por los cuales se proponen realizar la siguiente investigación.
- Se elabora un marco teórico que ayude a forjar una idea general para la realización del trabajo de tesis, y justifique el uso de herramientas específicas seleccionadas entre otras disponibles.
- Se plantea una hipótesis la cual es una posible respuesta al problema planteado y posee una directa relación entre el problema y el objetivo.
- Se propone la operacionalización de las variables en base a la hipótesis planteada.
- Se observa el comportamiento del ambiente de pruebas y se realiza la recolección de datos tanto en el estado vulnerable como en el seguro para su posterior contrastación.
- Se realiza la prueba de la hipótesis con los resultados obtenidos utilizando un método de comprobación adecuado a la característica de los datos obtenidos en el levantamiento de la información.
- Se elaboran las conclusiones en base a los resultados y recomendaciones producto de las experiencias obtenidas durante la investigación realizada.

Método Deductivo: Debido a que se partirá del conocimiento general y los casos de prueba existentes para trabajar con nuestro caso de pruebas particular y específico, considerando que existe material suficiente sobre temas principales como la seguridad en redes y limitada información sobre las mallas de datos.

3.4 Técnicas

Considerando las diferentes etapas del proyecto, se utilizan técnicas como:

Observación Indirecta: Utilizada principalmente para la selección de herramientas disponibles a utilizar, basados en estudios preliminares ya existentes.

Observación Directa: Utilizada para el análisis de los diversos momentos del proyecto, así como para el levantamiento información.

Fichaje: Utilizada para la gestión y organización de la información obtenida.

3.5 Fuentes de Información

Primaria:

Información original obtenida por el investigador en el ambiente de pruebas implantado, con el fin de contrastar la hipótesis.

Secundaria:

- Artículos publicados en revistas científicas.
- Trabajos de investigación publicados a nivel nacional e internacional con temas afines al investigado.
- Páginas de internet que brinden información confiable y especializada.
- Libros especializados en la biblioteca y electrónicos.
- Revistas electrónicas.

3.6 Recursos

a) Recursos humanos

Dentro la parte humana intervienen:

- Ejecutor del Proyecto
- Tutor del Proyecto
- Miembros del Tribunal Calificador
- Asesores Expertos

b) Recursos materiales

Considerando que el presente proyecto se planteará utilizando equipos físicos disponibles, estos recursos materiales serán detallados en los apartados correspondientes, para evitar la duplicidad de información en este apartado únicamente se detallarán los recursos materiales que no formen parte del experimento directamente.

- Memorias USB

- Impresora
- Hojas Papel Bond

c) Recursos técnicos

Tabla 2-3 Recursos Técnicos

RECURSO	CARACTERÍSTICA	DESCRIPCIÓN
Computador de Escritorio	Procesador Intel Dual Core, 1 gb de RAM	Computador dedicado a ser un cliente de la grid de datos
Computador de Escritorio	Procesador Intel Dual Core, 1 gb de RAM	Computador dedicado a ser un cliente de la grid de datos
Computador de Escritorio	Procesador Intel Dual Core, 1 gb de RAM	Computador dedicado a ser un cliente de la grid de datos
Computador de Escritorio	Procesador Intel Dual Core, 1 gb de RAM	Computador dedicado a ser un cliente de la grid de datos
Router Huawei	Modelo hg8245	Router ADSL / Frontera
Computador de Escritorio	Procesador Intel Core 2 duo con 2 gb de Ram	Computador dedicado a ser el servidor de la grid de datos
Switch Genérico	8 Puertos	Switch correspondiente a la Red LAN
Laptop 1	Procesador Intel core I7 con 16 gb de Ram	Equipo destinado a ser el atacante, mediante el uso de máquinas virtuales se pretende incrementar la concurrencia
Linux Ubuntu 12.04	Versión LTS 32 bits	Sistema Operativo Libre utilizado en todos los miembros de la grid de datos, así como en los atacantes de red.
Wireshark	Versión 2.4.2 para Linux Ubuntu 64 bits	Analizador de Protocolos de Red
Linux Ubuntu 12.04	Versión LTS 64 bits	Sistema Operativo Libre utilizado en el servidor
SNORT	IDPS para Ubuntu 12.04	Sistema de Detención y Prevención de Intrusiones
VirtualBox	Versión 5.2	Software de Virtualización de Sistemas Operativos
Globus Toolkit	Versión 5	Middleware seleccionado para computación en malla, se encuentra tanto

en el servidor como en los clientes

Realizado por: Marco Gavilanes

3.7 Planteamiento de la Hipótesis

La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO conlleva una mejora en la disponibilidad, seguridad y el rendimiento de las redes utilizadas para computación en malla.

3.8 Determinación de las Variables

Variable Independiente: Implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo según la metodología PPDIOO.

Variable Dependiente: Disponibilidad, seguridad y rendimiento de las redes utilizadas en computación en malla.

3.9 Operacionalización Conceptual de Variables

Tabla 3-3 Operacionalización de Variables

VARIABLE	CONCEPTUALIZA-CIÓN	DIMENSIONES	INDICES	ITEMS	CRITERIO DE MEDICIÓN	TÉCNICA	INSTRU-MENTO	ESCALA
V. I: IMPLEMENTACIÓN DE MEJORES PRÁCTICAS EN REDES	Estrategias de optimización y organización de redes que pretenden que se ajusten a los requerimientos empresariales.	Gestión de Servicios Informáticos reconocidos a nivel mundial	Frecuencia de incidentes atendidos	¿Cuántos paquetes perdidos se detectan en un trabajo de computación en malla?	Cantidad	Observación	Ambiente de Pruebas	Numeración real positiva a partir del 0
			Tiempo invertido en el manejo de incidentes	¿Cuánto tiempo se desperdicia en el proceso de computación en malla a causa de la corrección de incidentes?	Cantidad	Observación	Ambiente de Pruebas	Numeración real positiva a partir del 0
V. D: SEGURIDAD, DISPONIBLIDAD Y RENDIMIENTO DE LAS REDES UTILIZADAS EN COMPUTACIÓN EN MALLA	Métricas esenciales en cualquier red para la garantía de la integridad de los datos y la optimización de recursos.	Seguridad	% de paquetes interceptados	¿Cantidad de paquetes que fueron interceptados durante su paso por la red de datos?	Cantidad	Observación	Wireshark/ SNORT	Numeración real positiva a partir del 0
		Disponibilidad	% recursos de procesamiento	¿Porcentaje de recursos de procesamiento (CPU y Memoria)?	MTBF	Observación	Loiq/ SNORT	Escala MTBF
		Rendimiento	% de paquetes perdidos	¿Cantidad de paquetes que se perdieron durante su viaje en la red?	Cantidad	Observación	Wireshark /Netcat/ SNORT	Numeración real positiva a partir del 0

Realizado por: Marco Gavilanes

3.10 Población y Muestra

Dado que en el Ecuador no existe ningún entorno de red para computación en malla, las mediciones no se pueden realizar en un entorno real. Por ello, la presente investigación se desarrollará un ambiente de pruebas. Para la determinación del número de equipos a ser utilizados, se han revisado otros estudios en los cuales se han planteado ambientes de pruebas o simulación de redes, como es el caso del estudio de Castillo (2012) denominado "Estudio comparativo del rendimiento de servidores web de virtualización sobre la plataforma windows server 2008". En este proyecto, la autora planteo el siguiente ambiente de pruebas:

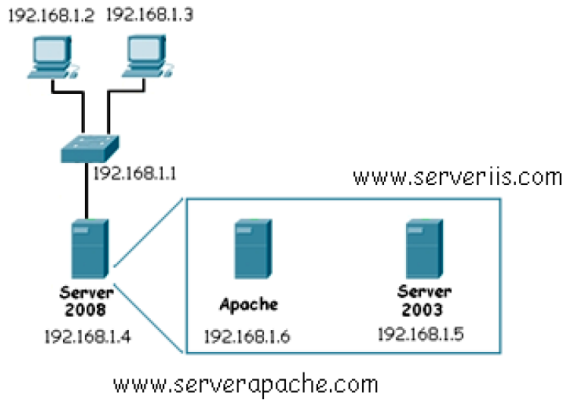


Figura 9-3 Estructura de red – prueba de rendimiento – caso 1

Fuente: (Castillo, 2012, p. 82)

Por otra parte, Córdova & Merino (2017) en su proyecto "Diseño e implementación de un emulador de redes", plantea las siguientes estructuras de red:

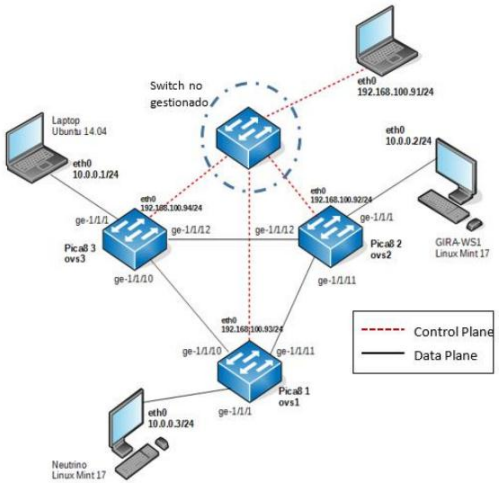


Figura 10-3 Estructura de red – pruebas de fidelidad – caso 2

Fuente: (Córdova & Merino, 2017, p. 96)

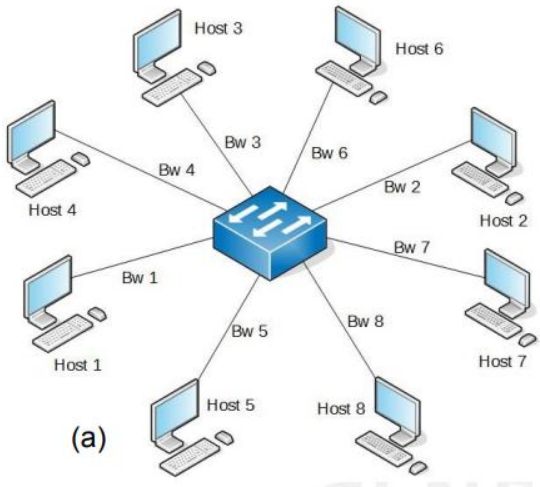


Figura 11-3 Estructura de red – pruebas de escalabilidad – caso 2

Fuente: (Córdova & Merino, 2017, p. 102)

Considerando los escenarios anteriormente expuestos, y que éstos emplean en promedio 5 computadores intercomunicados en red, se plantea el uso de este mismo número de equipos en una red Ethernet doméstica similar a las estudiadas durante el estado del arte, constituyéndose en la población de investigación. En base a esta red se demostrará la hipótesis, y en cuanto al tráfico, este será generado por un script creado por el investigador, compatible con Globus Toolkit y gestionado en el grid mediante GRAM.

3.11 Amenazas existentes en las Redes de Computadores

Actualmente, considerando el gran avance de la tecnología y las políticas ambientales que tratan de disminuir el uso de elementos como el papel y el plástico, han impulsado a que la mayoría de empresas mantengan su información de manera digital prioritariamente, así como también un gran porcentaje de personas prefieren mantener sus archivos digitalizados y en la nube así como prefieren tener álbumes digitales para sus recuerdos en lugar de álbumes físicos como se hacía hace unos años.

Considerando esta necesidad de conexión y de digitalización, como se expresó en el capítulo 2, el acceso a internet se elevó considerablemente en los últimos años, lo que consecuentemente generó que en cada hogar con acceso a internet existiera una red doméstica para proveer el mismo y en consecuencia volviendo propenso a ataques a este entorno en particular.

Se establecen 2 elementos fundamentales para considerar un riesgo de seguridad, simultáneamente debe existir vulnerabilidades y amenazas, es decir, debe existir un problema de

seguridad y una amenaza o atacante hacia ese problema, dado que la simple existencia de 1 de los 2 elementos no será suficiente para calificarlo como riesgo, por cuanto en el presente estudio se determinarán teóricamente las vulnerabilidades conocidas y se forzarán amenazas para poder obtener los riesgos de seguridad.

3.11.1 Determinación de amenazas a analizar

Considerando las sugerencias descritas por OWASP en su apartado de top 10 rc2 (OWASP, 2017), que trata sobre los 10 riesgos de seguridad más críticos en la seguridad de las aplicaciones web, se escoge de los 10 sugeridos, los aplicables al entorno del presente estudio.

Tabla 4-3 OWASP Top 10 2017

Código	Descripción
A1	Inyección
A2	Pérdida de Autentificación y Gestión de Sesiones
A3	Exposición de Datos Sensibles
A4	Entidad externa de XML (XXE)
A5	Pérdida de Control de Acceso
A6	Configuración de Seguridad Incorrecta
A7	Secuencia de Comandos en Sitios Cruzados (XSS)
A8	Des-serialización Insegura
A9	Uso de Componentes con Vulnerabilidades Conocidas
A10	Registro y Monitoreo Insuficientes

Fuente: (OWASP, 2017, p. 4)
Realizado por: Marco Gavilanes

Dado que formalmente no se considera como una aplicación web, sino un entorno de red, cuyo tráfico no se genera por navegadores sino por aplicaciones a nivel de sistema operativo, se eliminan temas como la autenticación (A2) y la inyección de scripts en la página web (A7).

Por otro lado, las configuraciones propias de las herramientas se mantienen bajo su propio estándar de seguridad bajo el sistema operativo, por cuanto el acceso a las mismas no se realiza a través de la red, eliminando así el uso de archivos XML (A4), por último se utiliza la versión del software más actualizado, al igual que las herramientas a las que la empresa da soporte, por cuanto la gestión de componentes queda fuera de nuestro control y bajo la responsabilidad de la empresa proveedora del middleware (A9).

Por lo anterior mencionado, se escoge de la lista de vulnerabilidades las siguientes para su testing.

- A1 Inyecciones
- A3 Exposición de Datos sensibles
- A5 Control de Acceso Defectuoso
- A6 Errores en la configuración de Seguridad
- A8 Des-serialización insegura
- A10 Insuficiente monitoreo

El entorno propuesto estará compuesto por 8 equipos dentro de la red de pruebas 192.168.1.0/24 en las direcciones:

Tabla 5-3 Equipos, red de pruebas

Dirección IP	**Funcionalidad**
192.168.1.10	servidor
192.168.1.11	atacante 1
192.168.1.12	atacante 2
192.168.1.13	equipo misceláneo
192.168.1.14	cliente 1
192.168.1.15	cliente 2
192.168.1.16	cliente 3
192.168.1.17	cliente 4

Realizado por: Marco Gavilanes

Basados en lo anteriormente mencionado, destacan 3 elementos principales dentro de las redes domésticas orientadas a la computación en malla, cada una con sus amenazas diferentes y cada una propensa a ataques, en la siguiente tabla se especifican los riesgos a los que está propenso cada elemento de la red.

Tabla 6-3 Riesgos en la Red

Amenaza	Sistema Operativo	Globus Toolkit	Red Física
Denegación de Servicio	Ubuntu 12.04, de manera nativa no presenta configuraciones de seguridad para prevenir un ataque.	No presenta un módulo interno orientado a la protección contra los ataques.	Al tratarse de una red doméstica, los equipos existentes no proveen una solución para prevenir los ataques de Denegación de Servicios, además que sus recursos son limitados.
Análisis de Tráfico / Rastreo de Puertos	De manera nativa no presenta herramientas que protejan el tráfico de datos.	Internamente protege los datos que el servidor envía a sus clientes, en cambio la información que los clientes envían al servidor no presenta ninguna protección.	Al tratarse de equipos domésticos, no contamos con la seguridad que brindan equipos como CISCO o Microtik contra este tipo de intrusiones.
Suplantación de Identidad	De manera nativa únicamente existe para la comunicación ssh, dado que utiliza un sistema de doble validación de llaves.	La herramienta a través de **myproxy** realiza la autenticación de cada miembro que pretenda comunicarse como cliente para obtener información utilizando un certificado SSL.	Considerando que el ataque se realiza desde una red externa, la suplantación de identidad no puede ser monitoreado ni protegido desde los elementos físicos de la red doméstica
Ataques de Fuerza Bruta	Todos los elementos involucrados están expuestos a un ataque de fuerza bruta y de diccionario para obtener cadenas de autenticación, de cifrado, etc dado que no presentan implementaciones de control del tipo desafío-respuesta para este control.		

Realizado por: Marco Gavilanes

3.12 Vulnerabilidades existentes

Como se menciona en el marco teórico, existen 3 capas dentro del modelo Jerárquico, las mismas que pueden ser cumplidas por varios equipos en cada capa o por otro lado, por un mismo dispositivo en varias capas, esto en dependencia de los recursos disponibles para la red.

Como se visualiza en el Gráfico 12, la red existente dispone de 1 router y 1 switch, lo cual según el modelo jerárquico representarían las capas de Acceso (Switching) y de Distribución (Routing), con lo que restaría la ubicación de la capa Núcleo (Backbone) misma que se define como el segmento de la red dedicado al switch de alta velocidad, al tener una red doméstica, esta diferenciación no se ve implementada resultando en la disposición de las capas de Acceso y Núcleo juntas en el Switch interno de la red y los procesos del middleware.

Tabla 7-3 Capas de la Red

Capa	Dispositivo	Detalle
Acceso	Switch	Provee la comunicación, el acceso de grupo de trabajo y controla el tráfico generado en este proceso dentro de la red interna
Distribución	Router	Provee las funciones de Listado de Acceso, firewalls, VLans, Ruteo estático entre otras relacionadas a la comunicación externa de la red.
Núcleo	Switch	Gestiona la comunicación de alta velocidad y baja latencia, en este caso no se dispone de un dispositivo orientado a esta funcionalidad, por cuanto el tráfico en su totalidad será gestionado por el mismo dispositivo de la Capa de Acceso

Realizado por: Marco Gavilanes

Considerando estos detalles, se complementa la decisión con lo mencionado por NIST (National Institute of Standards and Technology) ((NIST, 2007) (NIST, 2015), mismo que especifica que los principios de seguridad de una red son Confidencialidad, Integridad y Disponibilidad, por cuanto estos campos se asocian a las capas mencionadas anteriormente para definir el tipo de ataque a realizar a cada uno de los dispositivos de la red.

3.12.1 Selección de herramientas de análisis

La **Tabla 8** muestra la relación entre los principios de seguridad y las capas a atacar, información con la cual se definen las herramientas a utilizar para los ataques y aplicando la metodología del Benchmarking se obtienen los siguientes resultados, mismos que se encuentra representada en la **Tabla 9.**

Como se aprecia en la **Tabla 8**, se requieren ataques de SNIFFING y DDOS a diferentes equipos para obtener la información que nos permita evidenciar el estado de la red en cuanto a seguridad, por cuanto se revisan estudios relacionados a las herramientas disponibles para dichos ataques obteniendo los siguientes candidatos.

Tabla 8-3 Relación entre Principios y Capas

Principio de Seguridad	Definición	Capa Vulnerable	Ataque a Realizar
Disponibilidad	Se establece como el porcentaje de tiempo utilizable que presenta la red durante un lapso determinado. (Mejor mientras más alto)	Capa de Distribución	Se establece un ataque de denegación de servicios (DDOS a partir de ahora) como herramienta para reducir la disponibilidad a través de la saturación del Router de Frontera desde fuera de la red.
		Capa de Acceso y Núcleo	Se establece un ataque DDOS teniendo como objetivo el SWITCH y los miembros de la red, saturándolos con peticiones para que no puedan atender las necesarias.
Integridad	Se establece como la correspondencia entre la información almacenada con la generada, la integridad de la información se guarda cuando los datos almacenados corresponden a los generados y representan con fidelidad la información real. (Mejor mientras más alto)	Capa de Acceso y Núcleo	Considerando que la información es procesada dentro de la red, el riesgo a la integridad de datos se presenta en la posible alteración de un lote a procesar por uno de los clientes, lo que resultaría en la corrupción de la respuesta entregada al middleware, proceso que sería posible si se logra acceder a la estructura del lote que entrega el middleware a sus clientes, por cuanto se establece un ataque de SNIFFING a los miembros de la red para capturar y descifrar la estructura de un lote entregado por el middleware

Confidencialidad	Se establece como el proceso mediante el cual la información puede ser compartida únicamente con los entes que se considere necesario, la confidencialidad de la información en la red se mantiene mientras un elemento no pueda revisar información a la que según la lógica organizacional no tiene permitido acceder. (Mejor mientras más baja cantidad de información se pueda obtener)	Capa de Distribución	Como se mencionó anteriormente las redes domésticas a estudiar tienen como finalidad proveer del servicio de internet a los usuarios de las mismas, por cuanto los datos están expuestos a ser atacados desde equipos fuera de la red interna, por cuanto se establece un ataque de SNIFFING al Router de Frontera para verificar que no exista información que se exponga a la red pública.
		Capa de Acceso y Núcleo	Considerando que se tiene como fin principal integrar los miembros de una red en un trabajo compartido, la información viaja principalmente entre los miembros haciendo uso de la red interna, por cuanto se establece un ataque de SNIFFING hacia el tráfico generado en la red interna.

Realizado por: Marco Gavilanes

Tabla 9-3 Comparación – analizadores de tráfico y monitoreo

	ANALIZADORES DE TRÁFICO DE RED		MONITOREO DE REDES	
	WIRESHARK	**SNNIFER PRO**	**CENTREON**	**NAGIOS**
REQUERIMIENTOS	• Esta aplicación puede llegar a consumir bastantes recursos por lo que se puede saturar la memoria y espacio en disco, es recomendable un equipo con las características adecuadas. • Los requerimientos tanto para Linux como Windows son: • Arquitectura para 32 y 64 bits • Memoria RAM desde 128 (depende el número de paquetes que se vallan a capturar) • Espacio disponible 75 MB (dependiendo el número de paquetes) • Resolución de pantalla de 1280x2040 • Tarjetas de red soportadas , en Ethernet cualquiera y en inalámbricas las 802.11	• Es una herramienta analizadora de red que consiste en un conjunto de funciones, la cual no tiene muchos requerimientos. • Está diseñado para aprovechar las características y ventajas de Windows 32-bits como multitarea.	• Es una herramienta que está destinada a la monitorización y administración de redes y de dispositivos. • Como requerimientos necesita gran cantidad de programas ajenos y bastantes utilidades. • Deben de instalase: un servidor web, php5, servidor de MySQL, Perl, librerías, herramientas etc... • Se instalan ciertos plugins, etc…	Es un software de monitorización de equipos y servicios de red, creado para ayudar a los administradores a tener siempre el control de qué está pasando en la red y conocer los problemas que ocurren en la infraestructura antes de que los usuarios de la misma los perciban. Los requerimientos mínimos para su funcionamiento son: • Procesador: P4 1.8 GHz • Memoria RAM: 1 GB • Sistema Operativo: Linux • Tarjeta de Red:10/100/1000
SOPORTE DE PLATAF.	• Wireshark tiene un soporte multiplataforma es decir se puede utilizar en plataforma Unix y Windows el cual puede analizar el tráfico en la red: • Proporciona una interfaz para la adquisición de datos bruto (raw data) y el análisis de tráfico de la red. • Similar a tcpdump, pero con presentación gráfica. • Varias opciones para ordenar y filtrar la	• Tiene soporte solo en plataforma Windows para 32 bits. Donde se puede usar para: • Capturar el tráfico de red con análisis detallado • Diagnosticar problemas • Monitorear la actividad en tiempo real	• Centreon es una herramienta de monitorización y supervisión de redes y sistemas de trabajo y servidores para las plataformas: ○ GNU/Linux ○ Windows	• La plataforma Nagios se ejecuta sobre sistema operativo Linux, además proporciona los servicios de: ○ HTTP ○ POP3 ○ IMAP ○ FTP

	información capturada. Ver todo el tráfico de red. • Capturar y analizar los paquetes de datos. • Compatible con diferentes protocolos de red. Crear un plug-in para la lectura de nuevos protocolos. • Captura el tráfico USB. • La detección de llamadas VoIP de tráfico capturado. Si está utilizando una codificación compatible, también será capaz de reproducir el flujo multimedia.	• Generar alarmas • Prueba de red simulando tráfico. • Notifican cuando se detectan problemas. • Recolecta la utilización detallada y estadística de error para estaciones individuales.	○ Mac OS.	○ SSH
NODOS DE RED	El soporte de nodos de red puede ser en redes de cualquier tipo: • Misma máquina. • Misma red física (hubs). • Mismo switch con puerto espejo. • Mismo switch con cache poisoning. • En ambas maquinas (origen y destino).	Su soporte puede capturar y analizar el Ethernet, paquetes demasiado grandes, errores de alineación, token ring, WAN, paquetes U-frame, etc…	Centreon tiene gran escalabilidad y potencial para monitorizar una gran red de dispositivos.	• Soporta monitorear redes tanto locales como redes por internet • Sistemas de soporte a la red • Elementos de red • Aplicaciones de red • Soporte para implementar hosts de monitores redundantes
RENDIMIENTO	Proporciona alto rendimiento dentro del análisis de red ya que implementa una amplia gama de filtros que facilitan la definición de criterios de búsqueda para los más de 1100 protocolos soportados actualmente	Brinda gran rendimiento ya que puede realizar diversos tipos de análisis así como el filtro de paquetes y la generación de alarmas por si llega a existir algún fallo en la red.	Brinda opciones acerca del rendimiento, muestra información de los procesos y el estado en el que se encuentra el servidor.	• Mejora de productividad • Antelación de problemas • Reporte y aviso de incidentes • Agilidad en su tratamiento • Mejor y mayor relación e integración de sectores adjuntos

FACILIDAD DE USO	• Tiene una fácil instalación. • Además a la hora de comparar los paquetes es muy fácil por tener múltiples funciones.	Sniffer pro es una aplicación fácil de aprender y por tanto tiene un simple uso.	Centreon Permite un fácil y rápido acceso hasta los procedimientos y soluciones de incidencias	Es fácil de usar además La verificación de disponibilidad se delega en plugins y hace chequeos en paralelo (usando forking) así como la programación de chequeos inteligente.
COSTO – BENEFICIO	Wireshark es una herramienta principal de apoyo para ayudar a detectar, o al menos acotar en gran medida, los problemas generados por ataques a la red además es una aplicación de bajo coste y grandes beneficios por ser libre.	Sniffer pro es una herramienta para la visibilidad de redes de tipo comercial pero que brinda al analizador un gran beneficio por lo que invertir en el ofrece grandes ventajas	Centreon ofrece grandes beneficios ya que es una herramienta que nos brinda una gestión más cómoda	Nagios es un Software Libre, sin soporte, por lo que no tiene costo.

Realizado por: Marco Gavilanes

Para el proceso de Sniffing, se toma como base el estudio realizado por Judith Jazmin para la Universidad de Madrid, en donde analiza diferentes herramientas de Sniffing, e incluyendo la metodología de Benchmarking se realiza un análisis comparativo entre las mismas.

Al final de su estudio, concluye mencionando que entre la herramienta seleccionada Wireshark presenta las mejores características para procesos de SNIFFING mediante la aplicación del Benchmarking, por cuanto se establece esta herramienta para las pruebas de Sniffing en su versión más actual para Ubuntu 2.4.2.

De la misma manera, para los ataques DDOS y aplicando el Benchmarking, se tiene en consideración las 2 tendencias principales como son los ataques de órbita alta y de órbita baja, mismos que presentan herramientas específicas como son HOIC y LOIC respectivamente, cuyas características se describen a continuación.

Tabla 10-3 Comparación - Herramientas de ataque DDOS

	Multi-thread	**Multi-target**	**Acepta Scripts**	**Nativo Para Linux**	**Dificultad de Uso**	**Protección Legal**
HOIC	Si	Si	Si	Bugs Conocidos	Alta	Ninguna, el usuario se vuelve propietario de la herramienta y responsable del ataque al utilizarla
LOIC	Si	No	Si	Si (LOIQ versión basada en Qt, mientras LOIC se basa en C)	Media	El ataque se encarga a un Botnet que se vuelve responsable del ataque, legalmente se consideraría como un ataque involuntario

Realizado por: Marco Gavilanes

Como se muestra en la **tabla 10**, las ventajas de usar LOIC, específicamente LOIQ que es la versión basada en las librerías de Qt, por tanto es totalmente compatible con Ubuntu, superan a las de HOIC en cuanto a la seguridad, la curva de aprendizaje y el tipo de ataque que pretendemos realizar, por cuanto para DDOS se establece LOIQ como la herramienta a utilizar.

Finalmente para el monitoreo de los miembros de la red durante el ataque se requiere una herramienta de control de amenazas, como se mencionó en el capítulo 2, estudios nacionales relacionados al tema de seguridad establecen a SNORT como la herramienta idónea para

nuestro caso, por cuanto se establece la implementación de la misma en las máquinas miembros para poder obtener la información requerida sobre los ataques a implementar.

3.13 Ambiente de Simulación y pruebas

Como se mencionó al inicio del proyecto, la metodología a seguir para la implementación del prototipo final será PPDIOO, la misma que incluye 6 fases, un resumen del avance metodológico se presenta en la siguiente tabla.

Tabla 11-3 Avance PPDIOO

Fase	Descripción
Preparación	Considerando el tema del proyecto, la preparación consta del análisis previo a la definición del tema a estudiar dado que el proyecto no requería financiamiento
Planeación	Contempla todo el marco teórico relacionado al tema que sirve como base para establecer las fases a seguir en el proyecto y la orientación que tomará el mismo
Diseño	Contempla las decisiones tomadas basadas en el análisis realizado en el capítulo 3, mismo que deja planteadas las necesidades de la red y de las herramientas para la medición del desempeño de la misma
Implementación	Contempla el proceso de construcción física de la red así como de su preparación vía software para obtener el funcionamiento deseado de la red
Operativa	Durante esta fase se realiza el levantamiento de la información requerida para la solución de las vulnerabilidades encontradas en la red implementada
Optimización	Durante esta fase se implementan las mejores prácticas necesarias en las redes de acuerdo a lo detectado durante la fase operativa

Realizado por: Marco Gavilanes

Por cuanto el siguiente proceso es la implementación de la red y la instalación y configuración del middleware Globus Toolkit, mismo que, como se mencionó en el marco teórico es el que nos brinda las mejores prestaciones para el caso actual.

3.13.1 Ambiente de pruebas 1: Infraestructura Inicial

Configuración Hardware

Dada la disponibilidad física de los equipos requeridos para la implementación de la red como de los recursos necesarios para el ataque a la misma, no se utilizan herramientas de virtualización de entornos, en su lugar se tiene un entorno real especificado en la siguiente ilustración y basada en la tabla expresada en Recursos Técnicos. La infraestructura además posee un acceso de red igualitaria.

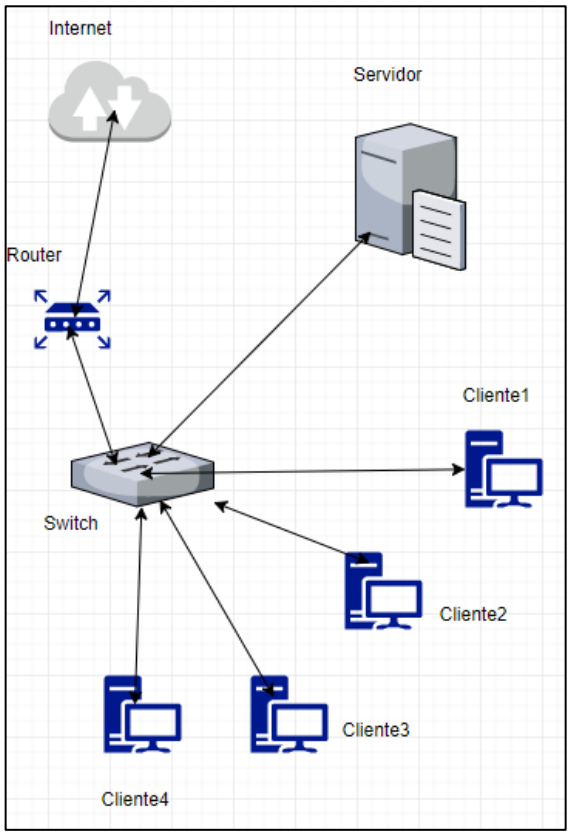


Figura 12-3 Infraestructura Inicial

Realizado por: Marco Gavilanes

Descripción

En el entorno y siguiendo las recomendaciones de Globus Toolkit, se utiliza una versión LTS del sistema operativo Ubuntu, cuyo soporte está garantizado por globus.org, por cuanto se destina el equipo con mayor hardware a ser el servidor mientras que el resto de equipos serán utilizados como clientes de la grid de datos, considerando que las redes a estudiar son

domésticas, estas no poseen un servidor dedicado a ser un firewall o un proxy, siendo esta la razón de centralizar los servicios necesarios en un único servidor principal.

A cada una de las máquinas se asigna las direcciones IP definidas en el punto **3.11.1.1**, generando así un ambiente en que todas las máquinas tienen conexión de datos.

Configuración Software

Ya con la red genérica definida con las propiedades más comunes entre redes domésticas se procede a la instalación del middleware para computación en malla en el computador servidor, mismo que se basa en la información disponible en su página oficial, misma que se menciona en el capítulo 2 del presente documento.

El procedimiento completo de la instalación se encuentra en los documentos digitales adjuntos al presente documento, aquí se mencionan los puntos principales a seguir durante la instalación del servidor.

- Instalación del Sistema Operativo Ubuntu 12.04 LTS
- Instalación y configuración de pre requisitos para Globus Toolkit 5
- Instalación de los servicios de Globus Toolkit (incluído simple_ca)
- Instalación del servidor de gftp
- Instalación del servidor myproxy
- Configuración del acceso a myproxy utilizando el certificado SSL creado con simple_ca (la clave del certificado es **globus**)
- Configuración del nivel de acceso de usuarios a las herramientas Globus Toolkit

Al final de este proceso se obtendrá una instancia independiente de Ubuntu 12.04 capaz de utilizar los servicios de Globus Toolkit para gestionar trabajos con gram entre todos los miembros de la red de computación en malla (actualmente solo el servidor), por cuanto para comunicar este servidor con sus clientes, se requiere un paso adicional que es la configuración del firewall haciendo uso de firewall-cmd para permitir la comunicación por los puertos requeridos por los diferentes servicios, mismos que son detectados utilizando netstat (herramienta nativa de Ubuntu para la verificación de tráfico en un puerto).

- Detección de puertos utilizados por los servicios.
- Configuración del firewall del Sistema Operativo.

Con esto obtendremos un servidor a la espera de la integración de más miembros a la red de datos para computación en malla (clientes Globus Toolkit).

Es indispensable tener la clave del certificado que utiliza el servidor para la identificación de miembros (**globus** en nuestro caso).

De igual manera a continuación se especifican las instrucciones generales para la creación de un cliente Globus Toolkit y su integración a la malla.

- Instalación del Sistema Operativo Ubuntu 12.04 LTS
- Instalación y configuración de pre requisitos para Globus Toolkit 5
- Instalación de los servicios de Globus Toolkit (incluído simple_ca)
- Instalación del servidor de gftp
- Instalación del servicio myproxy
- Importación del certificado SSL en uso por el servidor a través de myproxy
- Autorización del certificado en el Servidor
- Configuración del firewall del Sistema Operativo abriendo los mismos puertos que en el servidor.

Este proceso nos generará un cliente de la malla capaz de procesar trabajos bajo la gestión del servidor utilizando los servicios de gram.

Este proceso es repetido 3 veces más para generar los clientes especificados en el punto **3.11.1**

Finalizado este proceso, se establece como **Cumplido** el punto de **Implementación** que se describe en la **Tabla 11,** por cuanto se puede avanzar al punto Operativo de la implementación.

3.13.2 Ejecución del Experimento

Tomando las mismas consideraciones que en la sección anterior, se tienen los puntos A1, A3, A5, A6, A8 y A10 del documento OWASP, por cuanto se considerarán las mejores prácticas en redes orientadas a corregir estos puntos, dejando de lado, dada la naturaleza del tema los ajustes a nivel de red dado que se especifica el uso en redes domésticas.

Por otro lado, para la prueba de SNIFFING marcada como requerida en la **Tabla 8,** se vuelve necesaria la existencia de tráfico entre los miembros de la red, para lo cual se siguen las

instrucciones de Globus Toolkit para la generación de un script que gestione la creación de trabajos en gram con myproxy que se procesen en la malla. Este tráfico se generará para TCP como para UDP, cubriendo así los tipos de tráfico necesarios para tener un análisis completo.

La herramienta generadora de tráfico se encuentra adjunta en los archivos digitales entregados con el proyecto, a continuación se adjunta el flujo de proceso que sigue la herramienta, representado en UML.

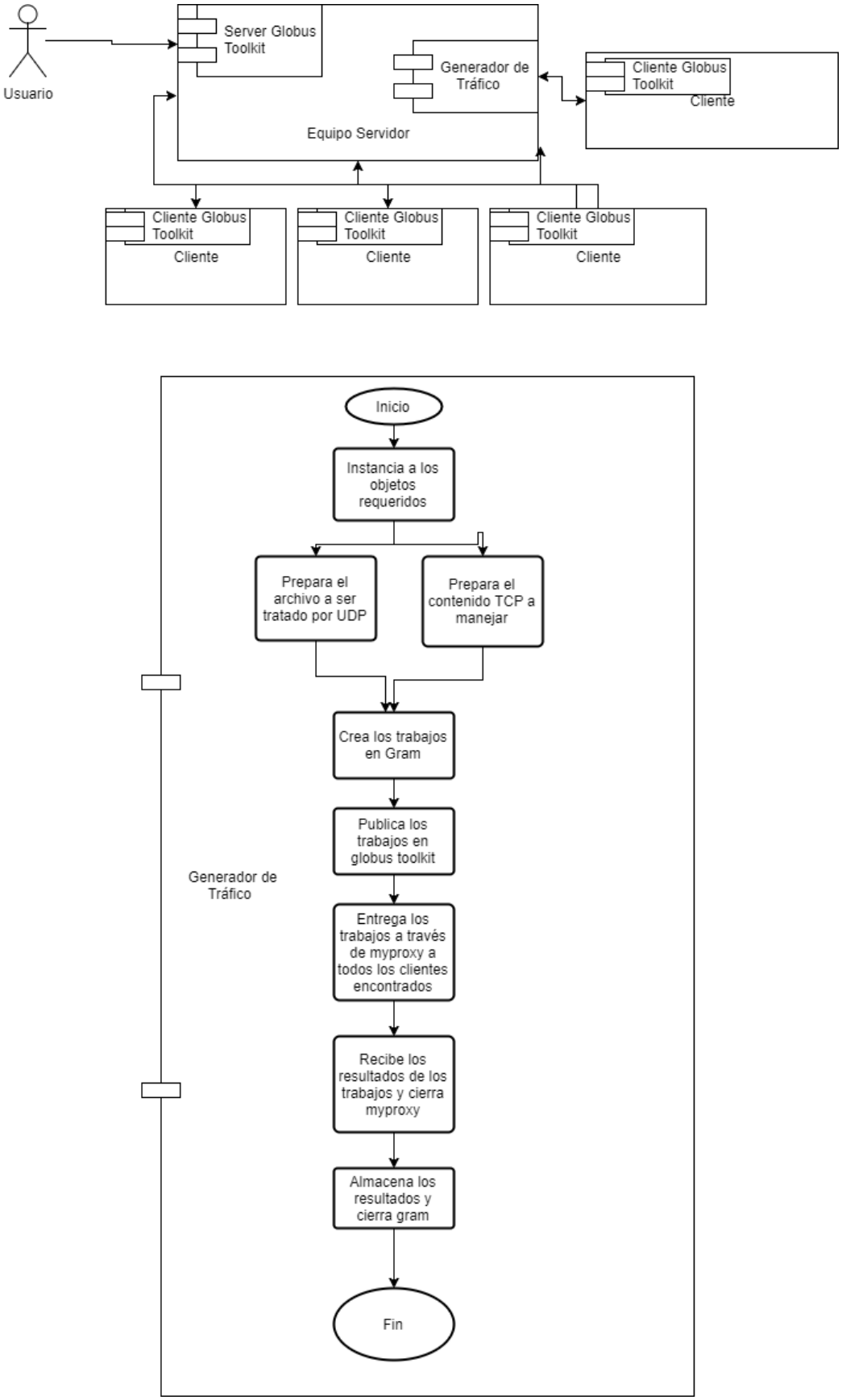


Figura 13-3 Línea de Flujo

Elaborado por: Marco Gavilanes

El proceso anterior se mantiene ejecutándose en un loop mientras no se la suspenda manualmente, generando así tráfico permanente en la red del tipo UDP y TCP, la cantidad de paquetes enviados se obtendrá de SNORT en el servidor para poder contrastar esta información con la cantidad de paquetes que se pudieron capturar con Wireshark, obteniendo así el porcentaje de paquetes interceptados, los valores exactos serán mostrados en el capítulo 4 y serán utilizados para la comprobación de la hipótesis planteada.

Por otro lado, la **Tabla 8** especifica la necesidad de un Ataque DDOS, mismo que se planifica con la herramienta LOIQ. La siguiente tabla establece los parámetros a utilizar en el ataque con sus respectivos objetivos.

Tabla 12-3 Ataque DDOS

Objetivo	Configuraciones	Detalles
Router	Se establece la IP en la herramienta y se apunta a ella (Lock ON), se establece el ataque por UDP, el puerto a utilizar y se coloca un mensaje OPCIONAL	El ataque se realiza desde un segmento externo de red al router de frontera para comprobar el principio de disponibilidad en la capa de distribución
Switch	Considerando la limitante de ataque exclusivo a direcciones IP o dominios, el ataque al SWITCH no está disponible, por cuanto para comprobar la red se atacarán a los miembros directamente.	No disponible por la limitante del atacante.
Servidor	Se establece la IP en la herramienta y se apunta a ella (Lock ON), se establece el ataque por UDP, el puerto a utilizar y se coloca un mensaje OPCIONAL	Se estresará al servidor con peticiones con un paralelismo de 100 que recomienda la herramienta para comprobar la disponibilidad en la capa de acceso y núcleo
Clientes (1, 2, 3 y 4)	Se establece la IP en la herramienta y se apunta a ella (Lock ON), se establece el ataque por UDP, el puerto a utilizar y se coloca un mensaje OPCIONAL	Se estresará a los clientes de manera individual con peticiones con un paralelismo de 100 que recomienda la herramienta para comprobar la disponibilidad en la capa de acceso y núcleo

Realizado por: Marco Gavilanes

Finalizadas estas pruebas, se consideró **Concluida** la fase de **Operación** de la metodología PPDIOO, por cuanto se pasó a la fase de Optimización.

3.13.3 Optimización

La fase final de la metodología contempla las correcciones requeridas en la red para lograr un funcionamiento óptimo, las correcciones requeridas son consideradas las mejores prácticas a implementar en la red para lograr un equilibrio entre la seguridad y el rendimiento de la red orientada a la computación en malla, la seguridad se validará por la reducción de la cantidad de paquetes entendibles capturados durante la generación de tráfico mientras que el rendimiento se medirá en base al tiempo que la red cumpla el principio de Disponibilidad especificado anteriormente.

A continuación, integrando el modelo OWASP con las condiciones planteadas durante todo el capítulo, se presenta un resumen de la observación y decisiones tomadas en cuanto al funcionamiento de cada uno de los puntos mencionados en OWASP. Los valores numéricos correspondientes se utilizarán en el capítulo 4 para el análisis de resultados, mientras que en el presente conglomerado se explicarán de manera general.

El proceso completo de funcionamiento y mediciones se encuentra adjunto en los archivos digitales entregados con el presente proyecto.

A1 Inyecciones

Considerando que el acceso durante la fase de pruebas se realiza a través de la interfaz directa dentro del sistema operativo, no existe una interfaz web a la que hacer inyecciones MIME (Extensión Multipropósito de Correo de Internet) directamente como lo menciona OWASP, sin embargo las inyecciones son un problema relacionadas a los Ataques DDOS, a los que en el apartado anterior se determinó que el prototipo es vulnerable, por cuanto considerando el sistema operativo Ubuntu 12.04 que tenemos en uso y sus herramientas nativas para no generar mayor consumo de recursos, se decide enfrentar al ataque DDOS utilizando la herramienta route de Ubuntu, complementada con SNORT que nos entregará la información requerida sobre el origen del ataque.

Experimento

Utilizando el log de SNORT, mismo que se adjunta en el Anexo 6, se detecta la IP de la que proviene el ataque, por cuanto utilizando la herramienta route, se procede a denegar el acceso a las peticiones provenientes del atacante utilizando:

route add -host 192.168.1.11 reject

Con lo que bloquearemos todo el tráfico proveniente del atacante, se considera una solución válida, basados en la naturaleza del entorno, al ser todas las máquinas partes de una grid de datos, únicamente se necesita que exista comunicación entre ellas, volviendo así factible la solución.

Considerando el uso de memoria ram en el Sistema operativo, el uso de memoria ram no se vio afectado, dado que la herramienta route ya pertenece y se encuentra en permanente ejecución en el Sistema Operativo.

A3 Exposición de Datos sensibles

Dado que en el experimento anterior se logró obtener de manera completa la información compartida entre el middleware y los miembros de la malla de computadores y tomando en consideración la sugerencia de OWASP, se implementa en el generador de tráfico la encriptación SSL sobre el mismo socket de comunicación establecido y basados en el certificado ubicado en /tmp/x509up_u1002, mismo que se genera y utiliza para la autenticación entre el middleware y sus clientes, la información del certificado se adjunta en el Anexo 8, obtenida por la propia herramienta de globus-toolkit "grid-cert-info".

Al establecer la comunicación encriptada sobre SSL, al igual que en el caso anterior sigue siendo capturada por wireshark en su retorno al middleware, pero en este caso la información es indescifrable dada la utilización del certificado.

A5 Control de Acceso Defectuoso

Al utilizar la autenticación por defecto de globus-toolkit mediante SSL, este proceso no puede ser mejorado.

A6 Errores en la configuración de seguridad

Como lo menciona la guía de pasos iniciales para Globus Toolkit 5, se deshabilita el firewall de las máquinas para permitir una comunicación correcta en un ambiente de pruebas, configuración que en un ambiente real se considera extremadamente peligrosa, por cuanto haciendo uso de las herramientas propias del sistema operativo, se detectan los puertos que son utilizados para la comunicación durante el correcto funcionamiento de la malla de computadores, utilizando netstat se logra detectar el listado de puertos que deben permanecer abiertos: 7512, 2811, 2119 y se gestiona el firewall propio del sistema operativo a través de su utilidad nativa firewall-cmd, utilizando:

firewall-cmd --zone=public –add-port=”número de puerto”/”protocolo de comunicación” –permanent

Con lo que se restringe el acceso a puertos innecesarios, de la misma manera se logra realizar esta acción sin la intervención de software externo, impidiendo así el uso de recursos.

A8 Des-serialización Insegura

Considerando la naturaleza del prototipo, se definen 2 etapas de comunicación entre los miembros de la malla de computadores:

1. **Autenticación.-** Proceso del que se encarga el middleware haciendo uso de la tecnología SSL y cuyo manejo no se puede personalizar, proceso inicial indispensable que intercambia y almacena la información de encriptación relacionada a su certificado, el Anexo 1 presenta un ejemplo de la autenticación realizada de forma manual desde uno de los clientes al middleware.
2. **Intercambio de Información.-** Proceso a cargo del generador de tráfico personalizado mismo que utilizando la información de encriptación relacionada al certificado almacenada durante el proceso de autenticación, envía y recibe información encriptada sin enviar información adicional relacionada a la serialización, dado que ese intercambio lo realizó el middleware previamente, por cuanto el generador de tráfico no maneja ni transporta los datos de encriptación únicamente la información ya encriptada.

A10 Insuficiente Monitoreo

Tomando en consideración que todos los ataques realizados lograron ser documentados por la herramienta SNORT y que el fin de esta investigación es lograr una seguridad aceptable sin afectar el rendimiento del prototipo, se establece que la herramienta SNORT utilizada para monitoreo presenta la suficiente información requerida para la identificación de vulnerabilidades.

Por otro lado, teniendo en cuenta el tipo de redes a utilizar y el ambiente doméstico que se trata de simular, la instalación independiente de snort se hace necesaria para el correcto monitoreo.

A continuación se presenta una tabla de resumen sobre el experimento contrastándolo con el prototipo anterior, en el que se denotará las características principales que debe presentar el prototipo para ser considerado seguro y con un rendimiento aceptable.

Tabla 13-3 Resumen del experimento en contraste con el prototipo anterior

	Solución	**Recursos Adicionales**	**Observación**
Inyecciones	SI	Ninguno	Se soluciona utilizando una herramienta propia del Sistema Operativo ya en ejecución
Exposición de Datos sensibles	SI	Carga adicional relacionada al proceso criptográfico	Se utiliza el certificado SSL utilizado por el middleware para encriptar y desencriptar los datos
Control de Acceso Defectuoso	No Aplica	No Aplica	El control de acceso lo gestiona directamente el middleware
Errores en la configuración de seguridad	Si	Ninguno	Se soluciona utilizando una herramienta propia del Sistema Operativo ya en ejecución por cuanto no será invasiva
Des-serialización Insegura	No Aplica	No Aplica	Se utiliza para el proceso criptográfico la información generada por el proceso de Autenticación, evitando así el manejo del transporte de la llave criptográfica
Insuficiente Monitoreo	SI	Coste de procesamiento propio de la	La herramienta SNORT fue capaz de detectar todos los ataques realizados y sirvió para detenerlos

herramienta SNORT

Realizado por: Marco Gavilanes

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

4.1 Resultados de Disponibilidad

Dado que los resultados aquí expuestos pueden ser verificados como válidos, el análisis de los mismos se debe realizar con una metodología confiable, por lo cual siguiendo la tendencia del Gobierno Español en su portal de administración electrónica, se hará uso de la metodología MAGERIT para análisis de riesgos de los Sistemas de Información, dado que está orientada (al igual que el presente estudio) a la detección de riesgos de seguridad y a la corrección de los mismos.

Como lo menciona la metodología, el tener dispositivos especializados reducirá el valor de riesgo. Dado que el estudio se basa en redes domésticas, el riesgo se establece en los siguientes valores.

Tabla 14-4 Probabilidades de Riesgo según Magerit

Riesgo	Probabilidad
Penetración	66%
Plagio de Información	33%
Posible Inhabilitación	99%

Realizado por: Marco Gavilanes

De la tabla 14, se realiza un ajuste considerando que el punto 2 hace referencia a un ambiente general sin la integración de una herramienta software, por cuanto en nuestro caso tenemos al servidor myproxy gestionando este apartado, lo cual lo vuelve poco confiable según los lineamientos de la metodología, por cuanto esta sugiere ignorar este literal, lo cual coincide con el planteamiento obtenido por OWASP en fases anteriores del proyecto.

Magerit define el impacto de una vulnerabilidad como el costo de recursos que se invierte en solventar un problema suscitado y lo categoriza en 4 rangos (Solarte, 2016):

Tabla 15-4 Rangos de Riesgo según Magerit

Bajo	0%-25%
Medio	25.01%-50%
Alto	50.01%-75%
Muy Alto	>75%

Realizado por: Marco Gavilanes

Para el cálculo del porcentaje de vulnerabilidad se establece como recursos: CPU (procesador) y Memoria requeridos para solventar el incidente. Durante el experimento, y utilizando la herramienta **free** del sistema operativo así como los log de snort, se obtuvieron los siguientes resultados en promedio en el Servidor. La "Vulnerabilidad" se calculó como la suma de los porcentajes de los recursos empleados en cada tipo de ataque.

Tabla 16-4 Magerit: vulnerabilidades identificadas – recursos Servidor, sin optimización

Ataque	CPU	Memoria	Vulnerabilidad
SNIFFING	26%	16%	**42%**
DDOS	99%	87%	**186%**

Realizado por: Marco Gavilanes

Los valores de la tabla 16 evidencian que la principal vulnerabilidad de la red está en el principio de Disponibilidad. En cuanto al Sniffing, este se ubica en el rango de medio, también teniendo en consideración que la herramienta en ejecución genera un consumo en recursos adicional (el valor se desprecia al tratarse por el momento sobre un estudio de Vulnerabilidad).

A continuación se analizará el riesgo que generan estas vulnerabilidades, utilizando la siguiente fórmula:

Fórmula 1-4 Fórmula del cálculo del riesgo

$$Riesgo = Amenaza * Vulnerabilidad$$

Donde:

- **Amenaza:** se basa en un catálogo que mantiene la metodología Magerit.
- **Vulnerabilidad:** calculado previamente

Tabla 17-4 Magerit: cálculo del riesgo- servidor, sin optimización

Amenaza	Vulnerabilidad	Impacto	Riesgo
SNIFFING	42%	Medio	42%*0.66=27.72%
DDOS	186%	Muy Alto	183%*0.99=181.17%

Realizado por: Marco Gavilanes

Evidenciado en los resultados, se prioriza la solución para el ataque DDOS en el servidor, mismo que se solucionó en todas las máquinas según lo especifica el punto A1 en el capítulo anterior utilizando una herramienta propia del sistema operativo, con lo cual el riesgo se redujo de la siguiente manera en el servidor:

Tabla 18-4 Magerit: vulnerabilidades identificadas – recursos Servidor, con optimización

Ataque	CPU	Memoria	Vulnerabilidad
SNIFFING	26%	16%	**42%**
DDOS	32%	19%	**51%**

Realizado por: Marco Gavilanes

Tabla 19-4 Magerit: cálculo del riesgo- servidor, con optimización

Amenaza	Vulnerabilidad	Impacto	Riesgo
SNIFFING	42%	Medio	42%*0.66=27.72%
DDOS	51%	Alto	51%*0.99=50.49%

Realizado por: Marco Gavilanes

Por otra parte, los clientes se comportaron de la siguiente manera (siguiendo las mismas especificaciones matemáticas):

Tabla 20-4 Magerit: vulnerabilidades identificadas – recursos Cliente, sin optimización

Ataque	CPU	Memoria	Vulnerabilidad
SNIFFING	64%	54%	118%
DDOS	100%	100%	200%

Realizado por: Marco Gavilanes

Tabla 21-4 Magerit: cálculo del riesgo- clientes (promedio), sin optimización

Amenaza	Vulnerabilidad	Impacto	Riesgo
SNIFFING	118%	Muy Alto	118%*0.66=77.88%
DDOS	200%	Muy Alto	200%*0.99=198%

Realizado por: Marco Gavilanes

Posterior a la optimización, se registraron los siguientes valores:

Tabla 22-4 Magerit: vulnerabilidades identificadas – recursos Cliente, con optimización

Ataque	CPU	Memoria	Vulnerabilidad
SNIFFING	64%	54%	118%
DDOS	73%	61%	134%

Realizado por: Marco Gavilanes

Tabla 23-4 Magerit: cálculo del riesgo- clientes (promedio), con optimización

Amenaza	Vulnerabilidad	Impacto	Riesgo
SNIFFING	118%	Muy Alto	118%*0.66=77.88%
DDOS	134%	Muy Alto	134%*0.99=132.66%

Realizado por: Marco Gavilanes

Estos valores representan al promedio de los valores obtenidos en los experimentos en los clientes, dado que, al tener todos el mismo hardware y las mismas funciones pueden ser agrupados.

Las consideraciones sobre los datos vienen especificadas por la Metodología, siendo estas:

- El riesgo es mayor a medida que el equipo tenga menos recursos, lo que se puede apreciar claramente en los datos obtenidos, confirmando los mismos.
- El valor neto obtenido durante los cálculos no es relevante, lo considerable es la diferencia entre estos valores.

La siguiente tabla representa el resumen de los valores obtenidos:

Tabla 24-4 Resumen – resultados de disponibilidad

Elemento	Sin optimización	Con optimización
Servidor	181.17	50.49
Clientes	198.00	132.66

Realizado por: Marco Gavilanes

4.2 Resultados de Seguridad y Rendimiento

Durante los mismos experimentos anteriores, se obtiene la siguiente tabla de resultados sobre la cantidad de paquetes capturados antes y después de la corrección del ambiente de pruebas, durante cada uno de los 10 experimentos realizados. Los experimentos se realizaron como se especifica en el capítulo anterior.

Tabla 25-4 Experimentos de seguridad – paquetes capturados antes de la corrección

Experimento	Cantidad Generada	Capturados en Envío	Capturados en Retorno	Detalle	Descifrable
01	386	8	146	Los paquetes contienen metadatos que informa sobre el tipo de tráfico (TCP o UDP), además que contienen la información	1
02	415	8	234		1
03	391	8	316		1
04	289	8	197		1
05	419	8	243		1

06	446	8	145	de la trama y de	1
07	375	8	92	autenticación, siendo solo	1
08	296	8	184	esta última encriptada.	1
09	125	8	46		1
10	384	8	163		1

Realizado por: Marco Gavilanes

Durante los 10 experimentos realizados se lograron capturar paquetes entendibles siguiendo las siguientes consideraciones:

- *Cantidad generada:* hace referencia a los paquetes salientes detectados por el SNORT existente en el servidor.
- *Capturados en envío:* son los paquetes con información relacionada a Globus Toolkit que tienen como Origen al Servidor mientras que Capturados en retorno son aquellos que tienen como destino al Servidor.
- *Descifrable*: especifica con el valor 1 si el contenido de la trama NO se encuentra encriptado y con un valor de 0 si el contenido de la trama referente a información así como el correspondiente a autenticación se encuentran encriptados.

De la **Tabla 25-4** se obtiene datos trascendentales como la protección unidireccional que provee Globus Toolkit cuando se comunica con los clientes, dado que ningún paquete es capturable mientras que también notamos que envía información de uso a un servidor externo, esto se soluciona desactivando el envío de estadísticas de uso.

El valor de Riesgo de esta matriz se calcula siguiendo las instrucciones de Magerit de la siguiente manera.

Fórmula 2-4: Fórmula del cálculo del riesgo - Magerit

$$Riesgo = Capturados\ en\ envío * Descifrable + Capturados\ en\ Retorno * Descifrable$$

Los cálculos efectuados dieron los siguientes resultados:

Tabla 26-4 Riesgos Magerit calculados sobre seguridad, experimentación sin correcciones

Experimento	Valor de Riesgo
01	154
02	242
03	324
04	205
05	251
06	153
07	100
08	192
09	54
10	171
Promedio	**186,40**

Realizado por: Marco Gavilanes

Implementando las soluciones mencionadas se realizan nuevamente los 10 experimentos, utilizando el mismo tráfico (script generador de tráfico) y se obtienen los siguientes resultados.

Tabla 27-4 Experimentos de seguridad – paquetes capturados post corrección

Experimento	Cantidad Generada	Capturados en Envío	Capturados en Retorno	Detalle	Descifrable
01	343	0	203	Los paquetes contienen metadata que informa sobre el tipo de tráfico (TCP o UDP), pero tanto la trama como los parámetros de autenticación se encuentran encriptados sobre el certificado compartido por myproxy	0
02	137	0	76		0
03	415	0	241		0
04	341	0	146		0
05	462	0	213		0
06	348	0	245		0
07	192	0	16		0
08	431	0	304		0
09	417	0	261		0
10	319	0	109		0

Realizado por: Marco Gavilanes

El tráfico se convierte en indescifrable dado que la llave de encriptación nunca viaja por el canal de comunicación, sino que al momento de configurar la red para la malla, el certificado es instalado en el cliente por myproxy, por cuanto la encriptación se realiza en el servidor

utilizando esta llave existente y la desencriptación se realiza en el cliente utilizando la llave que ya posee, requiriéndose la captura de la llave en el momento de la instalación de la red para volver esos datos descifrables para un atacante.

Utilizando la misma fórmula del riesgo obtenemos que el riesgo se vuelve 0 en todos los experimentos. Sin embargo, la experimentación demostró que el consumo de recursos incrementó por los procesos de encriptación y desencriptación implementados, por cuanto, en base a las tablas del apartado anterior se tienen los siguientes incrementos en el uso de recursos tanto en el servidor como en los clientes.

Tabla 28-4 Magerit, cálculo del riesgo – servidor, con optimización

Ataque	**CPU**	**Memoria**	**Vulnerabilidad**
SNIFFING	32%	18%	50%
Amenaza	**Vulnerabilidad**	**Impacto**	**Riesgo**
SNIFFING	50%	Medio	50%*0.66=33%

Realizado por: Marco Gavilanes

Tabla 29-4 Magerit, cálculo del riesgo – clientes, con optimización

Ataque	**CPU**	**Memoria**	**Vulnerabilidad**
SNIFFING	73%	81%	154%
Amenaza	**Vulnerabilidad**	**Impacto**	**Riesgo**
SNIFFING	154%	Muy Alto	154%*0.66=101.64%

Realizado por: Marco Gavilanes

Como resumen, en cuanto a SNIFFING, el promedio de los riesgos de los 10 experimentos basados en la captura de paquetes, se muestra a continuación:

Tabla 30-4 Resumen – resultados de seguridad

Elemento	**Sin optimización**	**Con optimización**
Servidor/Cliente	184.6	0

Realizado por: Marco Gavilanes

Respecto al Rendimiento tenemos un **impacto negativo** resumido a continuación:

Tabla 31-4 Resumen – resultados de rendimiento

Elemento	Sin optimización	Con optimización
Servidor	27.72	33.00
Clientes	77.88	101.64

Realizado por: Marco Gavilanes

A continuación se muestran los resultados obtenidos por cada elemento muestral de tipo cliente, considerando que estos serán ingresados el software de análisis estadístico:

Tabla 32-4 Resultados de disponibilidad – clientes de red en malla

	CPU	MEMORIA	Sin optimización	riesgo (*0,99)	CPU	MEMORIA	Con optimización	riesgo (*0,99)
cliente 1	99	100	199	197,01	72	62	134	132,66
cliente 2	100	100	200	198,00	74	61	135	133,65
cliente 3	100	98	198	196,02	73	58	131	129,69
cliente 4	100	100	200	198,00	73	61	134	132,66
PROMEDIOS	**100**	**100**	**200**	**198,00**	**73**	**61**	**134**	**132,66**

Realizado por: Marco Gavilanes

Tabla 33-4 Resultados de rendimiento – clientes de red en malla

	CPU	MEMORIA	Sin optimización	riesgo (*0,66)	CPU	MEMORIA	Con optimización	riesgo (*0,66)
cliente 1	63	53	116	76,56	70	80	150	99,00
cliente 2	64	54	118	77,88	75	81	156	102,96
cliente 3	63	52	115	75,90	71	80	151	99,66
cliente 4	64	55	119	78,54	74	83	157	103,62
PROMEDIOS	**64**	**54**	**118**	**77,88**	**73**	**81**	**154**	**101,64**

Realizado por: Marco Gavilanes

4.3 Comprobación de la Hipótesis

Para la comprobación de la hipótesis se ha considerado la prueba estadística de Wilcoxon en base a las siguientes razones:

- *Los datos obtenidos se encuentran organizados en parejas*: Wilcoxon compara el rango medio de dos muestras relacionadas.
- *La presente hipótesis plantea una prueba de dependencia entre una variable dependiente y una independiente*: Wilcoxon establece si la diferencia entre el rango medio de dos muestras relacionadas se debe al azar o no (la diferencia es significativa desde el punto de vista estadístico).
- *No se puede suponer la normalidad de los datos obtenidos:* No existen los suficientes datos experimentales para efectuar una prueba de normalidad. Wilcoxon se aplica en el caso de que no se pueda suponer la normalidad de la muestra, siendo una técnica de comprobación de hipótesis para datos NO PARAMÉTRICOS.

Para la presente comprobación se tomó en consideración la tabla de resumen del punto 4.1 y las 2 tablas de resumen del punto 4.2. Los valores se resumen a continuación:

Tabla 34-4 Resultados de disponibilidad, seguridad y rendimiento

		Disponibilidad		**Seguridad**		**Rendimiento**	
		Sin Opt.	**Con Opt.**	**Sin Opt.**	**Con Opt.**	**Sin Opt.**	**Con Opt.**
Servidores	Servidor	181,17	50,49	184,6	0	27,72	33,00
Clientes	Cliente 1	197,01	132,66	184,6	0	76,56	99,00
	Cliente 2	198,00	133,65	184,6	0	77,88	102,96
	Cliente 3	196,02	129,69	184,6	0	75,90	99,66
	Cliente 4	198,00	132,66	184,6	0	78,54	103,62

Realizado por: Marco Gavilanes

El análisis estadístico se basó en los siguientes parámetros:

Nivel de significancia: 5% = 0,05%

Toma de decisiones: p (Significancia) <0,05 rechaza la hipótesis nula

Sistema de Hipótesis:

Ho (Homogeneidad): La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO **NO** conllevó cambios significativos en la disponibilidad, seguridad y el rendimiento de las redes utilizadas para computación en malla.

Ha (Diferencias): La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO **SÍ** conllevó una mejora o disminución (cambio significativo) en la disponibilidad, seguridad y el rendimiento de las redes utilizadas para computación en malla.

Para la prueba se tienen 3 conjuntos de datos que se resumen en la tabla 34. Las pruebas se realizaron de manera independiente, generando por lo tanto 3 resultados. Con ellos se establecieron los casos de mejora y retroceso, sobre los que se evaluó el costo beneficio real acumulado de la implementación.

4.3.1 Aplicación de pruebas estadísticas – Proceso matemático

El estadístico de la prueba de los signos de Wilcoxon es:

Fórmula 3-4 Estadístico de la prueba de los signos de Wilcoxon

$$W^+ = \sum_{zi>0} R_i$$

Donde

z = diferencia absoluta entre los valores pareados

R = Rango calculado de cada par

Por lo cual W es igual a la suma de los rangos R correspondientes a los valores positivos de z.

Debido a que el tamaño de la muestra n < 25 no se puede sugerir una distribución normal de los datos, razón por la cual se comparará el valor obtenido (W) con los valores de la tabla de rangos críticos y signos de Wilcoxon a un nivel de significancia de 0,05 (ver gráfico 14).

Considerando que “una prueba de una cola (unilateral) normalmente está asociada a una hipótesis alternativa para la cual se conoce el signo de la potencial diferencia antes de ejecutar el

experimento y la prueba” , la presente prueba adoptará los valores de la tabla de rangos y signos de Wilcoxon para prueba de 1 cola, ya que se pretende demostrar la “mejora en la disponibilidad, seguridad y el rendimiento de las redes utilizadas para computación en malla”, luego de efectuados los procesos de optimización.

En el gráfico 14 puede observarse que, con una muestra n=5, y considerando que es una prueba de 1 cola, el punto crítico para una significancia de 0,05 es cero (0). Por lo tanto, los criterios de decisión estadística serían los siguientes:

- Si W <= 0 se acepta la hipótesis nula (no conllevó cambios significativos en la disponibilidad, seguridad y el rendimiento de las redes utilizadas para computación en malla)

- Si W > 0 se rechaza la hipótesis nula y se acepta la alternativa (si conllevó una mejora o disminución en la disponibilidad, seguridad y el rendimiento de las redes utilizadas para computación en malla). Si la diferencia de rangos es positiva, se demostrará una mejora, si es negativa una disminución.

Tabla 15. Prueba de rangos y signos de Wilcoxon.

VALORES CRÍTICOS (p=0.05)

n	Prueba 1 cola	Prueba 2 colas
5	0	
6	2	0
7	3	2
8	5	3
9	8	5
10	10	8
11	13	10
12	17	13
13	21	17
14	25	21
15	30	25

Tabla tomada de HR Neave. Elementary Statistics Tables. George Allen & Unwin Ltd.

Figura 14-4 Prueba de rangos y signos de Wilcoxon, valores críticos con p=0,05

Fuente: (Gómez, Vivó, & Soria, 2001)

Elaborado por: Marco Gavilanes

Considerando la significancia y punto crítico del análisis del presente estudio estadístico, las regiones de rechazo y aceptación de la hipótesis nula se presentan en el gráfico 15.

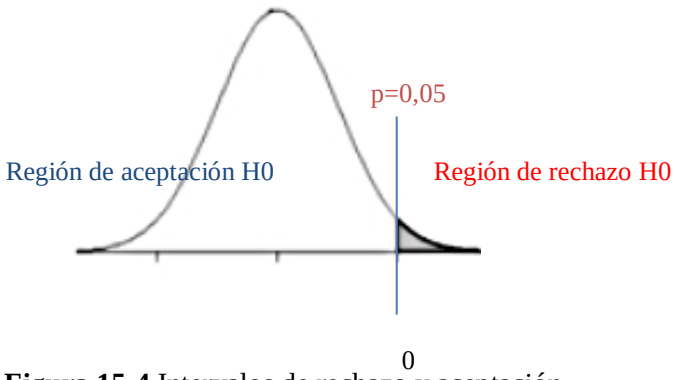


Figura 15-4 Intervalos de rechazo y aceptación

Elaborado por: Marco Gavilanes

Cálculo de W

Para el cálculo de W en cada sub variable se realizaron los siguientes procesos (resumidos numéricamente en las tablas 35 a la 37):

1) Ordenar las diferencias de menor a mayor valor, prescindiendo de los signos.

2) Promediar las posiciones de los números iguales y descartar las diferencias nulas, y asignar dicho valor al "rango" del par analizado.

3) Sumar los rangos (positivos y/o negativos por otro).

Tabla 35-4 Cálculo de W, sub variable disponibilidad

Sin Opt	Con Opt	Diferencia	Posición	Rango
181,17	50,49	130,68	5	5
197,01	132,66	64,35	2	1,5
198,00	133,65	64,35	1	1,5
196,02	129,69	66,33	4	4
198,00	132,66	65,34	3	3
			W (positivo)	15

Realizado por: Marco Gavilanes

Para la sub variable "disponibilidad" se obtuvo un W = 15 (positivo), el cual es mayor que cero, a partir del cual se rechazó la hipótesis nula y se concluyó que:

*La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO **SÍ** conllevó una **mejora** en la **disponibilidad** de las redes utilizadas para computación en malla.*

Tabla 36-4 Cálculo de W, sub variable seguridad

Sin Opt	Con Opt	Diferencia	Posición	Rango
184,60	0	184,6	1	3
184,60	0	184,6	2	3
184,60	0	184,6	3	3
184,60	0	184,6	4	3
184,60	0	184,6	5	3
			W (positivo)	15

Realizado por: Marco Gavilanes

Para la sub variable "seguridad" también se obtuvo un W = 15 (positivo), el cual es mayor que cero, a partir del cual se rechazó la hipótesis nula y se concluyó que:

*La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO **SÍ** conllevó una **mejora** en la **seguridad** de las redes utilizadas para computación en malla.*

Tabla 37-4 Cálculo de W, sub variable rendimiento

Sin Opt	Con Opt	Diferencia	Posición	Rango
27,72	33	-5,28	1	1
76,56	99	-22,44	2	2
77,88	102,96	-25,08	4	4,5
75,9	99,66	-23,76	3	3
78,54	103,62	-25,08	5	4,5
			W (negativo)	15

Realizado por: Marco Gavilanes

Para la sub variable "rendimiento" también se obtuvo un W = 15 (negativo), el cual es mayor que cero, a partir del cual se rechazó la hipótesis nula y se concluyó que:

*La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO **SÍ** conllevó una **disminución** en el **rendimiento** de las redes utilizadas para computación en malla.*

Finalmente, se pudo concluir que:

> *La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO* ***SÍ*** *conllevó una* ***mejora en la disponibilidad y seguridad*** *de las redes utilizadas para computación en malla, pero a la vez* ***disminuyó*** *el* ***rendimiento*** *de la misma.*

La disminución del Rendimiento, se debe a que en la Red Ethernet tanto en el Servidor como en los Clientes de la Red se realizaron las configuraciones necesarias para solventar las diferentes vulnerabilidades detectadas, por lo que el rendimiento en cada uno de los integrantes de la red disminuyó y por lo tanto disminuye el rendimiento total de la Red Ethernet.

4.3.2 Aplicación de pruebas estadísticas – SPSS

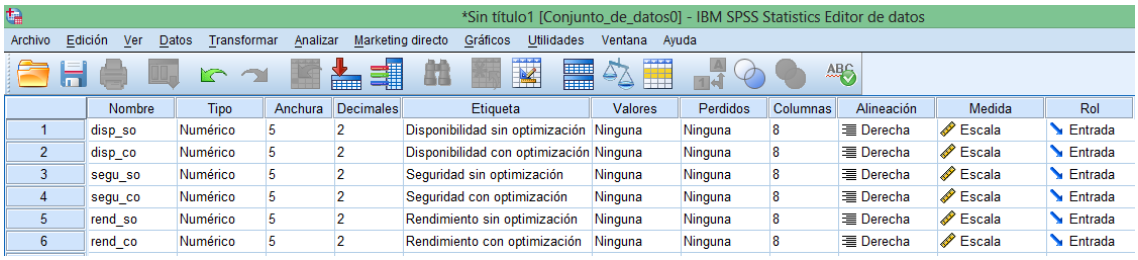
Con la finalidad de validar los resultados obtenidos matemáticamente, se empleó el software estadístico SPSS para confrontar dichos resultados. Cabe recalcar que este análisis es bilateral, a diferencia del anterior que fue unilateral, ya que el software analiza las variables sin conocer el signo de la diferencia que se busca comprobar.

En el análisis estadístico se ha empleado el software SPSS vs 19. Considerando la existencia de 3 subvariables o conjuntos de datos (disponibilidad, seguridad y rendimiento), se efectuaron 3 pruebas estadísticas según la siguiente distribución de variables SPSS:

Tabla 38-4 Variables SPSS

Sub variable	Nombre	Descripción
Disponibilidad	disp_so	Disponibilidad sin optimización
	disp_co	Disponibilidad con optimización
Seguridad	segu_so	Seguridad sin optimización
	segu_co	Seguridad con optimización
Rendimiento	rend_so	Rendimiento con optimización
	rend_co	Rendimiento sin optimización

Realizado por: Marco Gavilanes



	Nombre	Tipo	Anchura	Decimales	Etiqueta	Valores	Perdidos	Columnas	Alineación	Medida	Rol
1	disp_so	Numérico	5	2	Disponibilidad sin optimización	Ninguna	Ninguna	8	Derecha	Escala	Entrada
2	disp_co	Numérico	5	2	Disponibilidad con optimización	Ninguna	Ninguna	8	Derecha	Escala	Entrada
3	segu_so	Numérico	5	2	Seguridad sin optimización	Ninguna	Ninguna	8	Derecha	Escala	Entrada
4	segu_co	Numérico	5	2	Seguridad con optimización	Ninguna	Ninguna	8	Derecha	Escala	Entrada
5	rend_so	Numérico	5	2	Rendimiento sin optimización	Ninguna	Ninguna	8	Derecha	Escala	Entrada
6	rend_co	Numérico	5	2	Rendimiento con optimización	Ninguna	Ninguna	8	Derecha	Escala	Entrada

Figura 16-4 SPSS – vista de variables

Elaborado por: Marco Gavilanes

En cuanto a los valores ingresados (Tabla 34), se incluye a continuación la vista de datos del software SPSS:

	disp_so	disp_co	segu_so	segu_co	rend_so	rend_co
1	181,17	50,49	184,60	,00	27,72	33,00
2	197,01	132,66	184,60	,00	76,56	99,00
3	198,00	133,65	184,60	,00	77,88	102,96
4	196,02	129,69	184,60	,00	75,90	99,66
5	198,00	132,66	184,60	,00	78,54	103,62

Figura 17-4 SPSS – vista de datos

Elaborado por: Marco Gavilanes

Para en análisis estadístico, se seleccionaron las opciones Analizar > Pruebas no paramétricas > Cuadros de diálogo antiguos > 2 muestras relacionadas, luego de lo cual se ingresaron las variables, por pares, tal como se muestra en el siguiente gráfico:

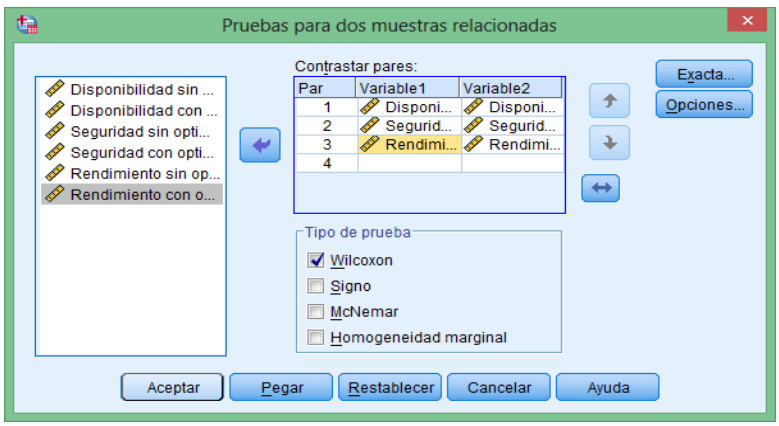


Figura 18-4 SPSS – Configuración de pruebas para dos muestras relacionadas

Elaborado por: Marco Gavilanes

Una vez procesados los datos, el visor estadístico de SPSS nos devuelve los siguientes resultados:

Estadísticos de contraste[c]

	Disponibilidad con optimización - Disponibilidad sin optimización	Seguridad con optimización - Seguridad sin optimización	Rendimiento con optimización - Rendimiento sin optimización
Z	-2,032[a]	-2,236[a]	-2,032[b]
Sig. asintót. (bilateral)	,042	,025	,042

a. Basado en los rangos positivos.
b. Basado en los rangos negativos.
c. Prueba de los rangos con signo de Wilcoxon

Figura 19-4 SPSS – Resultados de pruebas estadísticas Wilcoxon

Fuente: SPSS v. 19

Disponibilidad

La prueba de disponibilidad está basada en rangos positivos, lo cual supone que la diferencia entre ambas variables (riesgo sin optimización – riesgo con optimización) es positivo, lo cual demostraría inicialmente un mejoramiento de la disponibilidad al haberse reducido positivamente el riesgo.

Al analizarse el valor de p (Significancia asintólica):

0,042 < 0,05 = **se rechaza la hipótesis nula**, respecto a la subvariable disponibilidad**.**

De esta manera se ha demostrado estadísticamente que:

> **Ha (Diferencias):** La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO **SÍ** conllevó una **mejora en la disponibilidad** de las redes utilizadas para computación en malla.

Seguridad

De igual manera, la prueba de seguridad está basada en rangos positivos, lo cual supone que la diferencia entre ambas variables (riesgo sin optimización – riesgo con optimización) es positivo,

lo cual demostraría inicialmente un mejoramiento de la seguridad al haberse reducido positivamente el riesgo.

Al analizarse el valor de p (Significancia asintólica):

0,025 < 0,05 = **se rechaza la hipótesis nula**, respecto a la subvariable seguridad**.**

De esta manera se ha demostrado estadísticamente que:

> **Ha (Diferencias):** La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO **SÍ** conllevó una **mejora en la seguridad** de las redes utilizadas para computación en malla.

Rendimiento

La prueba de rendimiento, al contrario de las anteriores, está basada en rangos negativos, lo cual supone que la diferencia entre ambas variables (riesgo sin optimización – riesgo con optimización) es negativo, lo cual demostraría efectivamente una disminución del rendimiento al haberse aumentado el riesgo, al contrario de las anteriores donde la diferencia es positiva.

Al analizarse el valor de p (Significancia asintólica):

0,042 < 0,05 = **se rechaza la hipótesis nula**, respecto a la subvariable rendimiento**.**

Considerando que el estudio está basado en rangos negativos, se ha demostrado estadísticamente que:

> **Ha (Diferencias):** La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO **SÍ** conllevó una **disminución del rendimiento** de las redes utilizadas para computación en malla.

Conclusión

En base a la aplicación de la prueba de Wilcoxon para 2 muestras relacionadas, se pudo concluir que:

La implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO **SÍ** conllevó una **mejora en la disponibilidad y seguridad** de las redes utilizadas para computación en malla, pero a la vez **disminuyó** el **rendimiento** de la misma.

Los resultados de ambos procesos, tanto el matemático como el estadístico, fueron iguales, comprobándose la validez estadística de los mismos.

4.4 Análisis porcentual

Con la finalidad de establecer la medida porcentual de la afectación de la optimización de la red en malla experimental, se promediaron los valores cuantitativos obtenidos de disponibilidad, seguridad y rendimiento, y se calcularon los porcentajes y diferencias del estado anterior y posterior a la optimización.

Tabla 39-4 Análisis Porcentual

		Disponibilidad		**Seguridad**		**Rendimiento**	
		Sin Opt	**Con Opt**	**Sin Opt**	**Con Opt**	**Sin Opt**	**Con Opt**
Servidores	Servidor	181,17	50,49	184,6	0	27,72	33
Clientes	Cliente 1	197,01	132,66	184,6	0	76,56	99
	Cliente 2	198	133,65	184,6	0	77,88	102,96
	Cliente 3	196,02	129,69	184,6	0	75,9	99,66
	Cliente 4	198	132,66	184,6	0	78,54	103,62
Promedios		194,04	115,83	184,60	0,00	67,32	87,65
Diferencia		78,21		184,60		-20,33	
Porcentajes		**40,31%**	Aumentó	**100%**	Aumentó	**-30,20%**	Disminu-ción

Realizado por: Marco Gavilanes

Como puede observarse en la tabla 39, la disponibilidad incrementó en un 40,31%, la seguridad en un 100%, pero el rendimiento disminuyó en un 30,20%. Estos valores referenciales permitirán, a los administradores de red, tomar decisiones basadas en una perspectiva cuantitativa.

CAPÍTULO V

5. PROPUESTA

Como pudo observarse en el capítulo anterior, el rendimiento se ve afectado al aplicar mejores prácticas en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO y el modelo OWASP. De esta manera, las configuraciones planteadas en el presente capítulo responderán a la necesidad de los administradores de redes malla de mejorar la seguridad y disponibilidad de la información, cuando el rendimiento de los procesos computacionales de la red no es un parámetro decisivo en la gestión de datos.

Esto puede resultar contradictorio considerando que las redes en malla están diseñadas para incrementar el rendimiento de los procesos. Debe considerarse sin embargo que aunque el rendimiento de los nodos disminuye al aplicar correctivos en vulnerabilidades como: Inyecciones, Exposición de Datos sensibles, Errores en la configuración de seguridad, e Insuficiente Monitoreo; el procesamiento distribuido siempre será más eficiente que los sistemas centralizados.

5.1 Configuraciones propuestas

El Anexo 1 contiene un manual de implementación de Globus Toolkit a nivel de clientes y servidor.

A más de las actividades propuestas en dicho manual, se sugiere la aplicación de las siguientes configuraciones:

5.1.1 Inyecciones

Para la detección de posibles “inyecciones”, debe considerarse la realización de los siguientes pasos:

a) Instalar y Configurar SNORT

a.1. Ingresar el comando de instalación:

apt-get install snort

a.2. Ingresar la dirección de red local que será monitoreada y hacer clic en aceptar:

Un solo equipo: *192.168.1.10/32*

Toda la red local: *192.168.1.0/24*

Figura 20-5 Snort, configuración de intervalo de direcciones para la red local

Elaborado por: Marco Gavilanes

a.3. Luego de instalado, se realiza la configuración completa con el comando:

dpkg-reconfigure snort

a.4. Seleccionar "manual" en la ventana de opciones

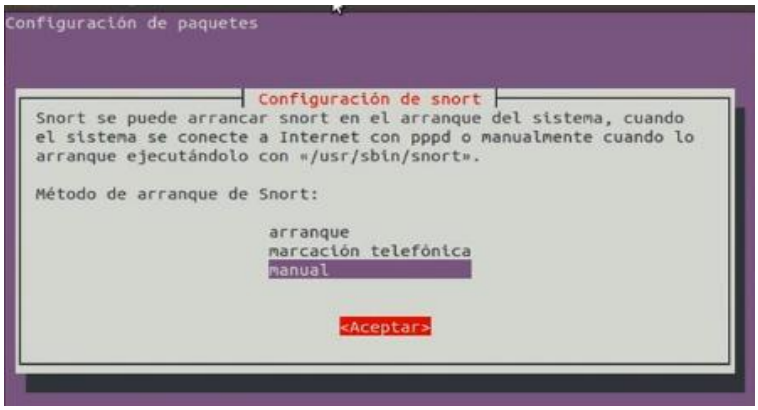


Figura 21-5 Snort, configuración método de arranque

Elaborado por: Marco Gavilanes

a.5. Ingresar las interfaces que escuchará SNORT durante el monitoreo:

Figura 22-5 Snort, configuración de interfaces para monitoreo

Elaborado por: Marco Gavilanes

a.6. Volverá a solicitar la configuración del intervalo de direcciones para la red local, seleccionar "aceptar".

a.7. Habilitar el modo promiscuo, para a revisión de todos los paquetes que pasen por el segmento de Ethernet.

a.8. Activar tarea en cron, para envío de resúmenes diarios de los registros de Snort a un correo electrónico e ingresar su correo electrónico.

Figura 23-5 Snort, configuración tarea cron

Elaborado por: Marco Gavilanes

a.9. Configurar el número mínimo de ocurrencias para que SNORT incluya una alerta en los informes. Inicialmente se colocará en 1, pero a medida que se vaya monitorizando la red deberá cambiar en función a los eventos de vulnerabilidad que se vayan presentado.

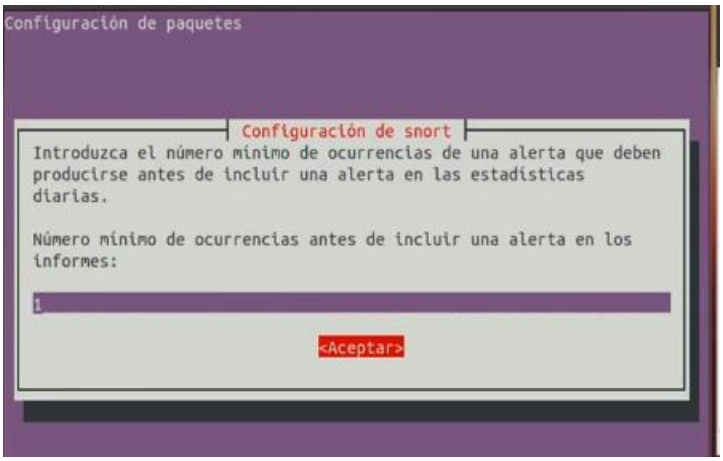


Figura 24-5 Snort, configuración número de ocurrencias

Elaborado por: Marco Gavilanes

a.10. Reiniciar SNORT

Figura 25-5 Snort, reiniciar proceso

Elaborado por: Marco Gavilanes

a.11. Crear/Editar un archivo de reglas de SNORT mediante el comando

nano /etc/snort/rules/site.rules

a.12. Se ingresaran todas las reglas que se requieran en dicho archivo, por ejemplo, las siguientes reglas alertan sobre el ingreso a "Facebook", o si "alguien intenta leer los puertos", o si se "hace ping" desde algún equipo. Una vez ingresadas se guardan los cambios en el archivo.

```
alert tcp any any -> any any (msg:"alguien entró a facebook";
content:"facebook";sid:19910314;rev:1;)
alert tcp any any -> any any (msg:"alguien esta intentando leer los
puertos";content:"nessus";sid:19910315;rev:1;)
alert icmp any any -> any any (msg:"ping";sid:19919316;rev:1;)
```

a.13. Editar el archivo de configuración de snort para incluir el archivo de las reglas que recientemente se creó

nano /etc/snort/snort.conf

Incluir en el contenido la siguiente linea:

include $RULE_PATH/site.rules

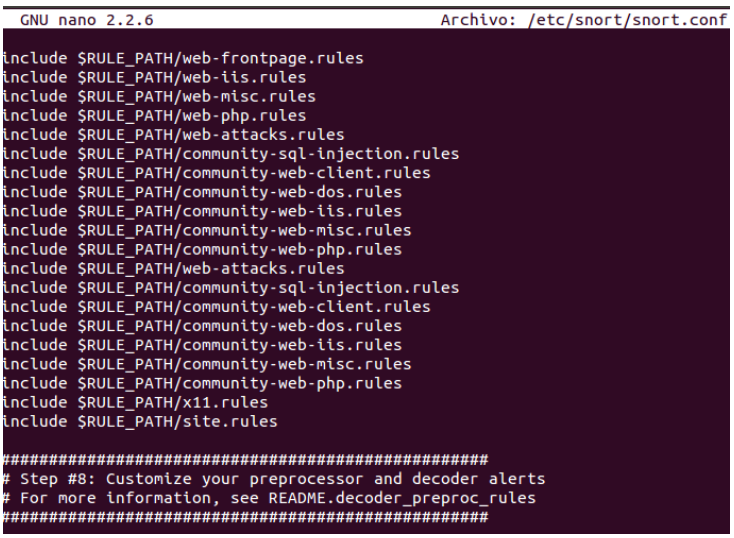


Figura 26-5 Snort, agregar regla a archivo de configuración

Elaborado por: Marco Gavilanes

b) Monitorear e identificar IPs de equipos atacantes

b.1. Ingresar a la carpeta de SNORT mediante el comando

cd /etc/snort

b.2. Iniciar el proceso "snort" de monitoreo mediante el comando

snort –A console –c snort.conf –i wlan0

Si se efectúan pruebas, podrá identificarse que las reglas se ejecutan correctamente, y que snort detecta las ips de los equipos que infringen dichas reglas

Figura 27-5 Snort, resultados de monitoreo

Elaborado por: Marco Gavilanes

Respecto al prototipo de red planteado en el presente proyecto, la monitorización debe efectuarse sobre los puertos a través de los cuales se receptará y enviará información, desde y hacia el servidor.

Por otra parte, el número de ocurrencias para iniciar una alerta debe ser mínima (por ejemplo, 1 o 2), para la detección temprana de equipos atacantes.

c) Denegar acceso a equipo atacante

Una vez identificado el equipo atacante (ip), denegar su acceso mediante el siguiente comando:

route add -host <ip del atacante> reject

5.1.2 Errores en la configuración de seguridad

Mediante el comando netstat se detectar el listado de puertos que deben estar abiertos para el adecuado funcionamiento de la malla: 7512, 2811, 2119.

Posteriormente, se gestiona el firewall propio del sistema operativo a través de su utilidad nativa firewall-cmd, utilizando:

firewall-cmd --zone=public –add-port=”número de puerto”/”protocolo de comunicación” –permanent

5.1.3 Insuficiente monitoreo

Se sugiere la instalación de software de monitoreo, como lo es SNORT, con la finalidad de detectar todos los ataques realizados y detenerlos.

CONCLUSIONES

- En base procesos de revisión bibliográfica, se seleccionó el middleware Globus Toolkit para la etapa de experimentación, considerando que es ampliamente aplicado en entornos experimentales, y que mantiene un gran soporte documental y de paquetes de instalación (para ambientes linux/Ubuntu) en relación a otros como gLite, UNICORE, XtreemOS.
- El ambiente de pruebas fue desarrollado en base a una red de acceso igualitario compuesta por 4 computadores clientes, 1 servidor y 2 atacantes red. La selección del número de equipos participantes en el experimento respondió al análisis de otros proyectos experimentales similares.
- Mediante la metodología PPDIOO se aplicaron mejores prácticas en el ambiente de pruebas, en las capas de acceso, distribución y núcleo, integrando el modelo OWASP respecto a los siguientes riesgos de seguridad: A1 Inyecciones, A3 Exposición de Datos sensibles, A5 Control de Acceso Defectuoso, A6 Errores en la configuración de Seguridad, A8 Des-serialización insegura, A10 Insuficiente monitoreo.
- La metodología PPDIOO, como un ciclo que orienta al que lo aplica claramente al mejoramiento de los procesos, sentó la base procedimental sobre la cual pudo desarrollarse el experimento exitosamente, es decir según la metodología al realizar la fase de operación se pudo identificar claramente las vulnerabilidades existentes y en la fase de optimización se logró minimizar dichas vulnerabilidades.
- Mediante la aplicación de la prueba estadística de Wilcoxon se pudo concluir que la implementación de mejores prácticas en redes Ethernet, en las capas de acceso, distribución y núcleo aplicando la metodología PPDIOO sí conllevó una mejora en la disponibilidad y seguridad de las redes utilizadas para computación en malla, pero a la vez disminuyó el rendimiento de la misma. En términos porcentuales, la disponibilidad incrementó en un 40,31%, la seguridad en un 100%, pero el rendimiento disminuyó en un 30,20%.
- Como resultado del proceso investigativo, se especificaron las configuraciones mínimas a aplicar en una red en malla de acceso igualitario, a fin de mejorar su disponibilidad y seguridad.

RECOMENDACIONES

- Utilizar la presente investigación como guía para la elaboración de un proyecto en donde se pretenda utilizar computadores distribuidos en redes domésticas como miembros de una red orientada a la computación en malla, de manera que se optimice los recursos hardware subutilizados, para cualquier proyecto que posea una infraestructura de red para sus administrativos, los mismos que estén orientados a funcionar como servidor de datos para proveer servicios e información a través de una red sin incurrir en costos adicionales en adquisición de hardware.
- Implementar un ambiente de pruebas de características similares de acuerdo con el estudio del arte para hacer uso de la computación en malla de manera que se determine las vulnerabilidades en seguridad y rendimiento.
- Mejorar la seguridad del tráfico a través de la implementación de serialización en los datos antes de ser enviados y basar la des-serialización en un eficiente control de acceso, para que los paquetes que llegasen a ser interceptados no puedan ser interpretados tan fácilmente. Además, se recomienda deshabilitar el envío de estadísticas al sitio del fabricante para que esa información guía no pueda ser interceptada, complicando así aún más la reconstrucción de los datos interceptados.

BIBLIOGRAFÍA

Arancegui, M. (17 de Julio de 2001). *Analisis y politicas de clusters* . Obtenido de http://eprints.ucm.es/6760/1/27-01.pdf

Aroquipa, E. (12 de Agosto de 2004). *http://www.unap.edu.pe/cidiomas/licing/pdf/sd.pdf.* Obtenido de http://www.unap.edu.pe/cidiomas/licing/pdf/sd.pdf

Castillo, J. (2012). *Estudio comparativo del rendimiento de servidores web de virtualización sobre la plataforma windows server 2008.* Obtenido de Dspace ESPOCH: http://dspace.espoch.edu.ec/handle/123456789/1946

CISCO. (15 de Febrero de 2010). *Diseño de redes* . Obtenido de https://wmagliano.wordpress.com/2008/09/27/diseno-de-redes-capitulo-1-ppdioo/

Córdova, R., & Merino, A. (2017). *Diseño e implementación de un emulador de redes.* Obtenido de Repositorio Digital - Pontificia Universidad Católica del Perú: http://tesis.pucp.edu.pe/repositorio/handle/123456789/8688

Enslow, P. (1978). *What is a "distributed" data processing system Computer.*

Espinoza, A. (Abril de 14 de 2010). *http://dspace.utpl.edu.ec/bitstream/123456789/1352/3/Espinosa_Otavalo_Ang%C3%A9lica%20del%20Cisne.pdf.* Obtenido de http://dspace.utpl.edu.ec/bitstream/123456789/1352/3/Espinosa_Otavalo_Ang%C3%A9lica%20del%20Cisne.pdf

Espinoza, A., & Montoya, D. (2016). *Diagnóstico de Seguridad Informática utilizando la metodología de Análisis de Vulnerabilidades en la red del Banco Nacional de Fomento - Casa Matriz Quito - Ecuador.* Obtenido de Dspace ESPE: http://repositorio.espe.edu.ec/handle/21000/11951

Gaona, K. (2013). *Aplicación de la metodología Magerit para el análisis y gestión de riesgos de la seguridad de la información aplicado a la empresa Pesquera e Industrial Bravito S.A. en la ciudad de Machala.* Obtenido de DSPACE UPS: https://dspace.ups.edu.ec/bitstream/123456789/5272/1/UPS-CT002759.pdf

Gobierno de España. (2012). *MAGERIT – versión 3.0 Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información.* Obtenido de https://www.ccn-cert.cni.es/documentos-publicos/1791-magerit-libro-ii-catalogo/file.html

Gómez, S., Vivó, M., & Soria, E. (2001). Pruebas de significación en Bioestadística . *Revista de Diagnóstico Biológico SciELO*. Obtenido de http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S0034-79732001000400008

Guillermo, J., Gualán, R., Solano, L., Collaguazo, D., Ramírez, W., & Cruz, A. (2015). *Infraestructura basada en Globus Toolkit para dar soporte a repositorios distribuidos*

de imágenes médicas. Obtenido de Tesis Universidad de Cuenca: https://publicaciones.ucuenca.edu.ec/ojs/index.php/maskana/article/viewFile/713/626

Hidalgo, M. I. (14 de Junio de 2009). *Administrador de proyectos de Grid* . Obtenido de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/330/IBERICO_MART%C3%8DN_ADMINISTRADOR_DE_PROYECTOS_DE_GRID_COMPUTING_QUE_HACEN_USO_DE_LA_CAPACIDAD_DE_COMPUTO.pdf?sequence=1

Luz, S. d. (3 de Noviembre de 2010). *Listado de diferentes ataques a las redes de los ordenadores* . Obtenido de https://www.redeszone.net/2010/11/03/ataques-a-las-redes-listado-de-diferentes-ataques-a-las-redes-de-ordenadores/

Mendoza, M. A. (Mayo de 2015 de 2007). *Apendizaje colaborativo sobre mallas computacionales* . Obtenido de http://cytig.udistrital.edu.co/elearning/pdfs/miguel_mendoza.pdf

NIST. (2007). *Border Gateway Protocol* . Obtenido de https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-54.pdf

NIST. (2015). *Guide to Industrial Control Systems (ICS) Security*. Obtenido de https://csrc.nist.gov/publications/detail/sp/800-82/rev-2/final

OWASP. (2017). *OWASP Top 10 - 2017 Los diez riesgos más críticos en Aplicaciones Web*. Obtenido de https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf

Quishpe, H. D. (16 de Marzo de 2016). *Análisis de Vulnerabilidades en la RED LAN Jerarquica de la Universidad Nacional de Loja* . Obtenido de http://dspace.unl.edu.ec/jspui/bitstream/123456789/16039/1/Quishpe%20Malla%2C%20Henry%20David.pdf

Rodriguez, P. (18 de Agosto de 2013). *Comunicación y admisión* . Obtenido de https://www.dcc.uchile.cl/node/833

Salgado, A. (2014). *Análisis de las aplicaciones web de la Superintendencia de Bancos y Seguros, utilizando las recomendaciones Top Ten de OWASP para determinar los riesgos más críticos de seguridad e implementar buenas prácticas de seguridad para el desarrollo de sus aplica.* Obtenido de DSPACE ESPE: http://repositorio.espe.edu.ec/xmlui/handle/21000/8245

Sánchez, J. (2017). *Análisis de vulnerabilidades y diseño de procesos correctivos de la página web de la Dirección de Educación a Distancia y Virtual de la Universidad Técnica de Ambato.* Obtenido de Dspace UTA: http://repositorio.uta.edu.ec/jspui/handle/123456789/25531

SN. (24 de Marzo de 2009). *Algoritmos* . Obtenido de http://informaticafrida.blogspot.com/2009/03/algoritmo.html

Solarte, F. (2016). *Estándar MAGERIT de análisis, evaluación y gestión de riesgos*. Obtenido de http://blogsgsi.blogspot.com/2016/07/eatandar-magerit-de-analisis-evaluacion.html

Sosa, V. (16 de Febrero de 2014). *Middleware* . Obtenido de http://www.tamps.cinvestav.mx/~vjsosa/clases/sd/Middleware_Recorrido.pdf

Unión Internacional de las Telecomunicaciones. (2017). *ITU World Telecommunication Indicators Database.* Rusia.

Villacreses, J., & Caiza, V. (2007). *Desarrollo de una aplicación GRID usando globus toolkit 4*. Obtenido de Tesis Escuela Politécnica Nacional: http://bibdigital.epn.edu.ec/handle/15000/157

GLOSARIO

A

Algoritmo: se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas. O bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema. Por lo tanto, podemos decir que es un conjunto ordenado y finito de pasos que nos permite solucionar un problema. (SN, 2009)

B

Benchmarking:

C

Clúster: En el ámbito de la informática, clúster se emplea con diversos sentidos. Se llama clúster al conjunto de computadoras (ordenadores) que se relacionan entre sí a través de una red de alta velocidad, actuando como una unidad (es decir, como una sola computadora). (Arancegui, 2001).

Computación: es también un área de conocimiento constituida por disciplinas relativas a las ciencias y la tecnología, para el estudio, desde el punto de vista teórico y práctico, de los fundamentos del procesamiento automático de datos, y su desarrollo, implementación y aplicación en sistemas informáticos. (Significados.com, s.f.)

H

Hardware: Es la parte física de un ordenador o sistema informático, está formado por los componentes eléctricos, electrónicos, electromecánicos y mecánicos, tales como circuitos de cables y circuitos de luz, placas, utensilios, cadenas y cualquier otro material, en estado físico, que sea necesario para hacer que el equipo funcione. El término viene del inglés, significa partes duras. (Significados.com, s.f.)

L

Lenguaje de programación: es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación. (Cartago, 2012)

M

Memoria Caché: Cuando en informática se habla de memoria caché o cache se está hablando de aquella cantidad de información que permanece de manera temporal en la computadora y que ayuda a la adquisición de velocidad y eficiencia cuando es necesario recurrir a determinado tipo de datos. El nombre de memoria cache proviene del francés, que significa "escondido" u "oculto". (Definición ABC, s.f)

Multicore: Es el término que describe al día de hoy los procesadores que tienen dos o más fichas de trabajo del procesador (más comúnmente conocida como núcleos) que trabajan simultáneamente como un solo sistema. Dual núcleos o chips con dos procesadores que funcionan como un sistema único son el primer tipo de núcleos múltiples aplicaciones de la tecnología. (AMD blogsot, 2008)

P

Programa: Un programa es una serie de instrucciones ordenadas, codificadas en lenguaje de programación que expresa un algoritmo y que puede ser ejecutado en un computador. (Capouya, 2009)

R

Red telemática: La Telemática cubre un campo científico y tecnológico de una considerable amplitud, englobando el estudio, diseño, gestión y aplicación de las redes y servicios de comunicaciones, para el transporte, almacenamiento y procesado de cualquier tipo de información (datos, voz, vídeo, etc.), incluyendo el análisis y diseño de tecnologías y sistemas de conmutación. (Tobar, 2010)

T

Tecnología: La tecnología es la aplicación coordinada de un conjunto de conocimientos (ciencia) y habilidades (técnica) con el fin de crear una solución (tecnológica) que permita al ser humano satisfacer sus necesidades o resolver sus problemas. (Definición, 2012)

ANEXOS

Anexo A: Configuración Globus Toolkit en red experimental

El sistema operativo empleado en el escenario de pruebas es **ubuntu-12.04.5-desktop-i386**. Luego de instalado en los equipos clientes y servidor, se procedió a realizar la siguiente configuración:

a. Configuración Servidor

a.1. Descargar pre-requisitos Globus toolkit:

wget http://www.globus.org/ftppub/gt5/5.2/5.2.3/installers/repo/globus-repository-5.2-stable-precise_0.0.3_all.deb

```
root@mservidor:/home/mservidor# wget http://www.globus.org/ftppub/gt5/5.2/5.2.3/installers/repo/globus-repositor
y-5.2-stable-precise_0.0.3_all.deb
--2018-05-04 11:58:08--  http://www.globus.org/ftppub/gt5/5.2/5.2.3/installers/repo/globus-repository-5.2-stable
-precise_0.0.3_all.deb
Resolviendo www.globus.org (www.globus.org)... 52.72.174.199, 54.85.236.50
Conectando con www.globus.org (www.globus.org)[52.72.174.199]:80... conectado.
Petición HTTP enviada, esperando respuesta... 301 Moved Permanently
Ubicación: http://toolkit.globus.org/ftppub/gt5/5.2/5.2.3/installers/repo/globus-repository-5.2-stable-precise_0
.0.3_all.deb [siguiente]
```

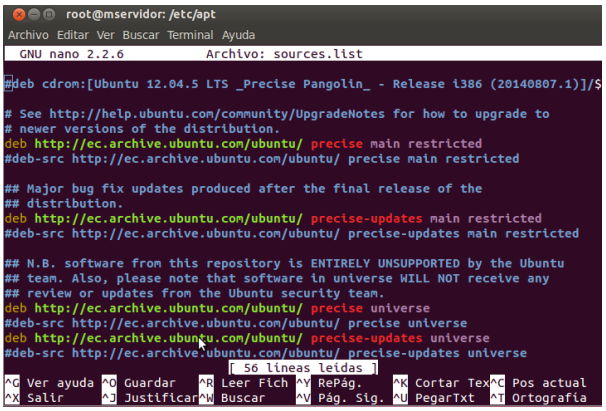
a.2. Instalar software de Globus toolkit:

dpkg -i globus-repository-5.2-stable-precise_0.0.3_all.deb

```
root@mservidor:/home/mservidor# dpkg -i globus-repository-5.2-stable-precise_0.0.3_all.deb
Seleccionando el paquete globus-repository-5.2-stable-precise previamente no seleccionado.
(Leyendo la base de datos ... 147048 ficheros o directorios instalados actualmente.)
Desempaquetando globus-repository-5.2-stable-precise (de globus-repository-5.2-stable-precise_0.0.3_all.deb) ...
Configurando globus-repository-5.2-stable-precise (0.0.3) ...
OK
```

a.3. Comentar (con el signo #) los source que inician con deb-src del archivo /etc/apt/sources.list, previo a actualizar el sistema operativo:

nano /etc/apt/sources.list

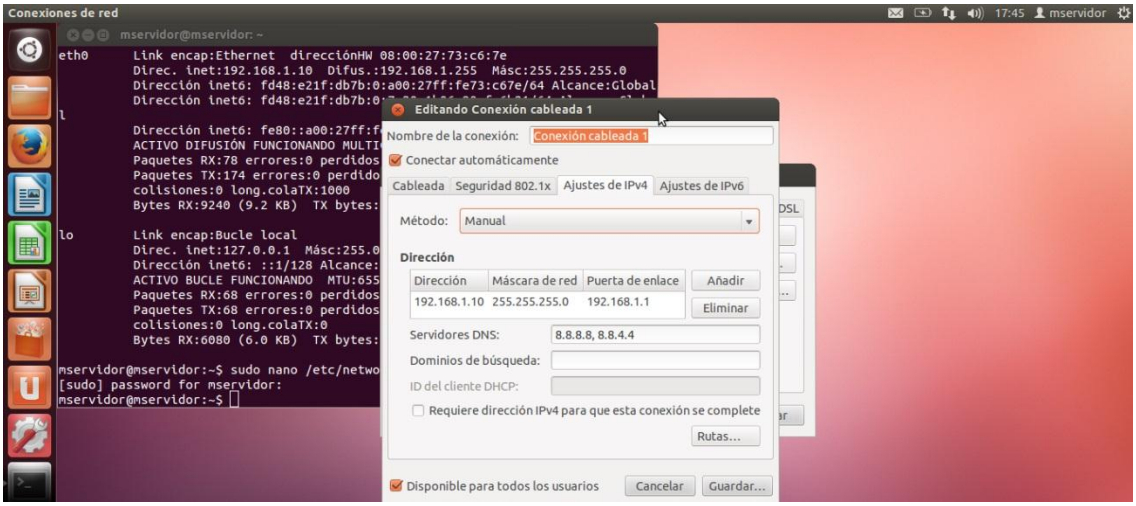


a.4. Actualizar el sistema operativo:

apt-get update

```
root@mservidor:/etc/apt# apt-get update
Obj http://ec.archive.ubuntu.com precise Release.gpg
Obj http://ec.archive.ubuntu.com precise-updates Release.gpg
Obj http://ec.archive.ubuntu.com precise-backports Release.gpg
Obj http://security.ubuntu.com precise-security Release.gpg
Des:1 http://extras.ubuntu.com precise Release.gpg [72 B]
Obj http://ec.archive.ubuntu.com precise Release
Obj http://ec.archive.ubuntu.com precise-updates Release
Obj http://security.ubuntu.com precise-security Release
Obj http://extras.ubuntu.com precise Release
```

a.5. Configurar una ip fija para los equipos:



Reiniciar Red e Interfaz Ethernet

sudo /etc/init.d/networking restart (reiniciar red)

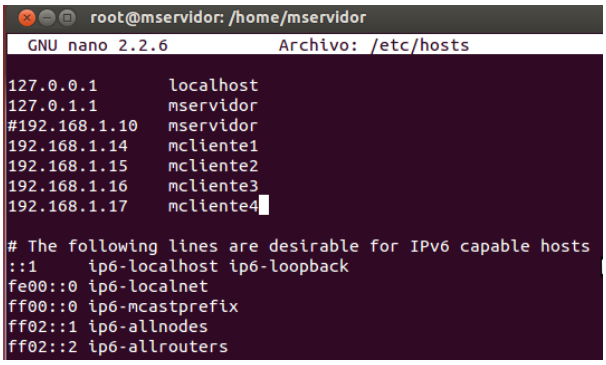
sudo ifconfig eth0 down

sudo ifconfig eth0 up

a.6. Agregar las direcciones ip de los equipos que se conectarán en la malla

nano /etc/hosts

```
root@mservidor:/home/mservidor# nano /etc/hosts
```



a.7. Instalar los paquetes requeridos para el manejo de los servicio Myproxy, GSI y GridFTP

apt-get install -y globus-data-management-client globus-gridftp globus-gram5 globus-gsi myproxy myproxy-server myproxy-admin libperl4-corelibs-perl

El comando anterior instala además la entidad certificadora de Globus Toolkit denominada SimpleCA. Con ella, se crea automáticamente una nueva entidad certificadora, un certificado público en el directorio de confianza de Globus, y una llave a ser empleada por los usuarios para ejecutar los servicios de Globus Toolkit.

```
root@mservidor:/home/mservidor# apt-get install -y globus-data-management-clie
nt globus-gridftp globus-gram5 globus-gsi myproxy myproxy-server myproxy-admin
 libperl4-corelibs-perl
You may use a command similar to the following:

 cat /etc/grid-security/hostcert_request.pem | mail root@mservidor

Only use the above if this machine can send AND receive e-mail. if not,
please
mail using some other method.

Your certificate will be mailed to you within two working days.
If you receive no response, contact Globus Simple CA at root@mservidor

The new signed certificate is at: /var/lib/globus/simple_ca/newcerts/01.
pem

/
Configurando libmyproxy5 (5.9-10) ...
Configurando libperl4-corelibs-perl (0.003-1) ...
Configurando myproxy (5.9-10) ...
Configurando myproxy-admin (5.9-10) ...
Configurando myproxy-server (5.9-10) ...
Configurando globus-gridftp (0.0.0) ...
Configurando globus-gsi (0.0.0) ...
Configurando globus-gram-job-manager-fork (1.5-5) ...
Configurando globus-gram-job-manager-fork-doc (1.5-5) ...
Configurando globus-gram-job-manager-fork-setup-poll (1.5-5) ...
Configurando globus-gram5 (0.0.0) ...
Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
```

a.8. Instalar los certificados y llave del servidor en Myproxy, para que éste servicio pueda emplearlos posteriormente.

Generación de llaves públicas 644

install -o myproxy -m 644 /etc/grid-security/hostcert.pem /etc/grid-security/myproxy/hostcert.pem

Generación de llaves públicas 600

install -o myproxy -m 600 /etc/grid-security/hostkey.pem /etc/grid-security/myproxy/hostkey.pem

```
root@mservidor:/home/mservidor# install -o myproxy -m 600 /etc/grid-security/hos
tkey.pem /etc/grid-security/myproxy/hostkey.pem
root@mservidor:/home/mservidor# install -o myproxy -m 644 /etc/grid-security/hos
tcert.pem /etc/grid-security/myproxy/hostcert.pem
```

Considerar que:

644 (Rw-r - r -) Propietario puede leer y escribir; los demás sólo pueden leer.

600 (Rw -------) Propietario puede leer y escribir en un archivo; los demás no tienen derechos.

a.9. Crear el servidor Myproxy

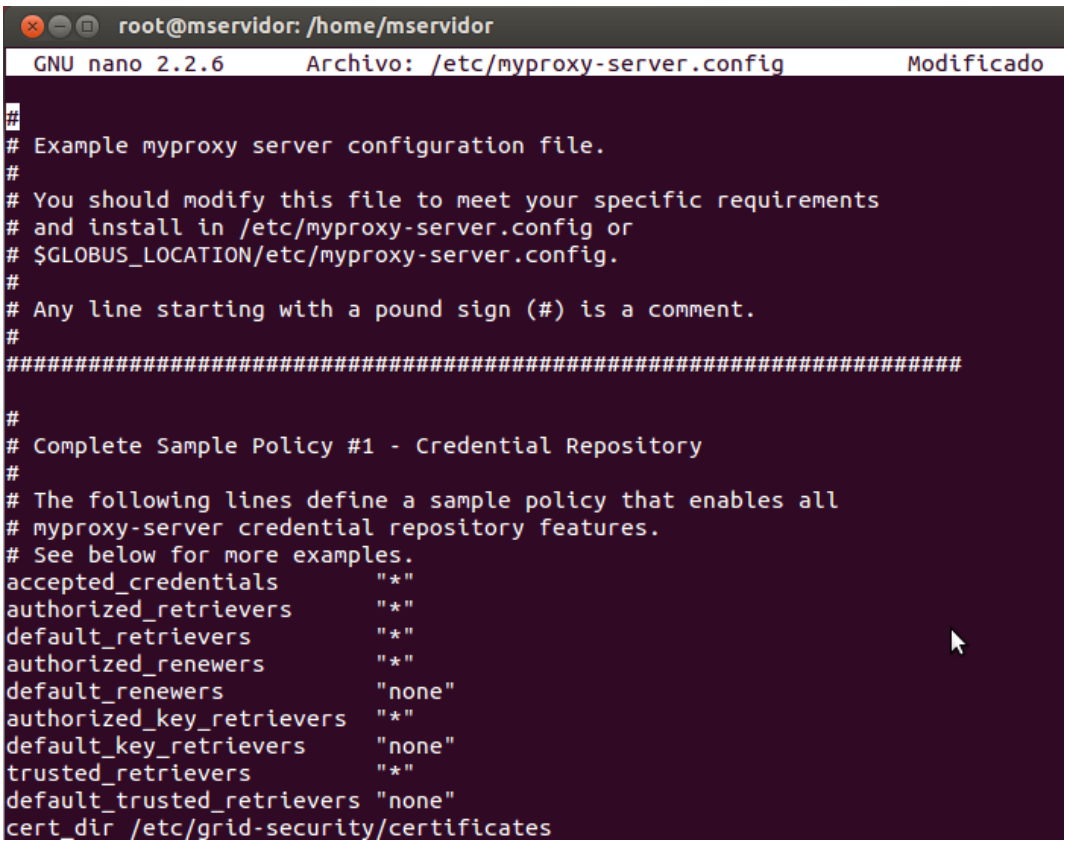
apt-get install myproxy-admin

```
root@mservidor:/home/mservidor# apt-get install myproxy-admin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
myproxy-admin ya está en su versión más reciente.
0 actualizados, 0 se instalarán, 0 para eliminar y 332 no actualizados.
root@mservidor:/home/mservidor#
```

a.10. Asociar Myproxy a SimpleCA

MyProxy almacena los certificados de los usuarios. Se modificará su configuración en el archivo /etc/myproxy-sever.config con la finalidad de que éste servicio emplee Simple CA, retirando el símbolo # en las líneas de la sección "Complete Sample Policy #1"

nano /etc/myproxy-server.config



a.11. Agregar el usuario myproxy al grupo de Simple CA, para que el servidor myproxy pueda generar certificados registrado en la red en malla.

usermod -a -G simpleca myproxy

```
root@mservidor:/home/mservidor# usermod -a -G simpleca myproxy
```

a.12. Iniciar el servicio myproxy

service myproxy-server start

```
root@mservidor:/home/mservidor# service myproxy-server start
 * myproxy-server already running
```

Configurar servicio myproxy para que éste arranque al igual que el equipo

chkconfig myproxy-server on

a.13. Verificar los puertos que se están utilizando

netstat -an | grep 7512

```
root@mservidor:/home/mservidor# netstat -an | grep 7512
tcp        0      0 0.0.0.0:7512            0.0.0.0:*               ESCUCHAR
```

a.14. Crear las credenciales

a.14.1. Cambiar al prompt del usuario myproxy

su - -s /bin/sh myproxy

a.14.2. Añadir la variable del PATH

PATH=$PATH:/usr/sbin

a.14.2. Agregar el usuario

En este caso, se agregará al cliente1, el cual dentro de la red se denomina mcliente1.

myproxy-admin-adduser -c "mcliente1" -l mservidor

```
root@mservidor:/home/mservidor# su - -s /bin/sh myproxy
$ PATH=$PATH:/usr/sbin
$ myproxy-admin-adduser -c "mcliente1" -l mservidor
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

The new signed certificate is at: /var/lib/globus/simple_ca/newcerts/04.pem

using storage directory /var/lib/myproxy
Credential stored successfully
Certificate subject is:
/O=Grid/OU=GlobusTest/OU=simpleCA-mservidor/OU=local/CN=mcliente1
```

La clave ingresada será ingresada posteriormente, cuanto el usuario requiera autentificarse ante el servidor proxy.

a.15. Crear mapa de credenciales

En este mapa se registrarán las credenciales de los usuarios, mismas que servirán para acceder a los servicios de Globus Toolkit

grid-mapfile-add-entry –dn "/O=Grid/OU=GlobusTest/OU=simpleCA-mservidor/OU=local/CN=mcliente1" -ln mservidor

```
root@mservidor:/home/mservidor# grid-mapfile-add-entry -dn "/O=
Grid/OU=GlobusTest/OU=simpleCA-mservidor/OU=local/CN=mcliente1"
 -ln mservidor
Modifying /etc/grid-security/grid-mapfile ...
/etc/grid-security/grid-mapfile does not exist... Attempting to create /
etc/grid-security/grid-mapfile
New entry:
```

a.16. Iniciar servicio GridFTP

A fin de probar GSI, la cual es la Insfraestructura de Seguridad para Grid que posee Globus Toolkit, se empleará el servicio GridFTP del mismo Globus Toolkit. Dicho servicio permite la compartición de archivos mediante un canal seguro establecido entre dos nodos certificados y autentificados en la malla o Grid.

service globus-gridftp-server start

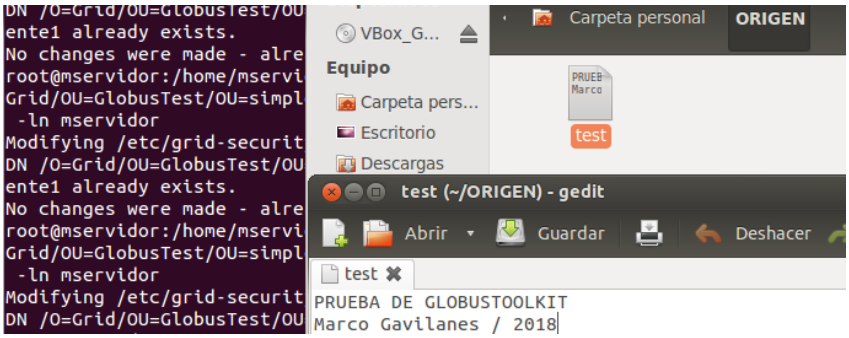
```
root@mservidor:/home/mservidor# service globus-gridftp-server start
 * Started GridFTP Server
root@mservidor:/home/mservidor# service globus-gridftp-server status
GridFTP Server Running (pid=1048)
root@mservidor:/home/mservidor#
```

Se verifica el estado del servicio, así como se conoce el puerto a través del cual escucha peticiones, en este caso el 1048

service globus-gridftp-server status

```
root@mservidor:/home/mservidor# service globus-gridftp-server status
GridFTP Server Running (pid=1048)
```

a.17. Generar archivo en el servidor, para su transferencia via gridftp desde el cliente1



b. Configuración Nodos Cliente

Como ejemplo, se ha expuesto la configuración del cliente1.

b.1. Descargar pre-requisitos Globus toolkit:

wget http://www.globus.org/ftppub/gt5/5.2/5.2.3/installers/repo/globus-repository-5.2-stable-precise_0.0.3_all.deb

```
root@mcliente1:/home/mcliente1# wget http://www.globus.org/ftppub/gt5/5.2/5.2.3/
installers/repo/globus-repository-5.2-stable-precise_0.0.3_all.deb
--2018-05-05 19:15:50--  http://www.globus.org/ftppub/gt5/5.2/5.2.3/installers/r
epo/globus-repository-5.2-stable-precise_0.0.3_all.deb
Resolviendo www.globus.org (www.globus.org)... 52.72.174.199, 54.85.236.50
Conectando con www.globus.org (www.globus.org)[52.72.174.199]:80... conectado.
Petición HTTP enviada, esperando respuesta... 301 Moved Permanently
Ubicación: http://toolkit.globus.org/ftppub/gt5/5.2/5.2.3/installers/repo/globus
-repository-5.2-stable-precise_0.0.3_all.deb [siguiente]
--2018-05-05 19:15:51--  http://toolkit.globus.org/ftppub/gt5/5.2/5.2.3/installe
rs/repo/globus-repository-5.2-stable-precise_0.0.3_all.deb
Resolviendo toolkit.globus.org (toolkit.globus.org)... 192.5.186.47
Conectando con toolkit.globus.org (toolkit.globus.org)[192.5.186.47]:80... conec
tado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 4586 (4,5K) [application/x-debian-package]
Grabando a: “globus-repository-5.2-stable-precise_0.0.3_all.deb”

100%[======================================>] 4.586       --.-K/s   en 0,1s

2018-05-05 19:15:51 (34,0 KB/s) - “globus-repository-5.2-stable-precise_0.0.3_al
l.deb” guardado [4586/4586]
```

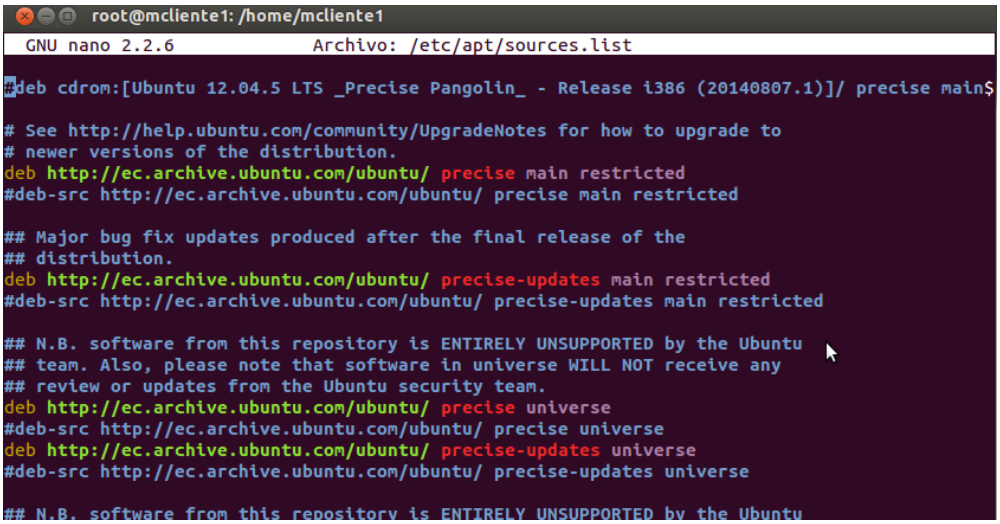
b.2. Instalar software de Globus toolkit:

dpkg -i globus-repository-5.2-stable-precise_0.0.3_all.deb

```
root@mcliente1:/home/mcliente1# dpkg -i globus-repository-5.2-stable-precise_0.0
.3_all.deb
Seleccionando el paquete globus-repository-5.2-stable-precise previamente no sel
eccionado.
(Leyendo la base de datos ... 147048 ficheros o directorios instalados actualmente.)
Desempaquetando globus-repository-5.2-stable-precise (de globus-repository-5.2-stable-precise
_0.0.3_all.deb) ...
Configurando globus-repository-5.2-stable-precise (0.0.3) ...
OK
```

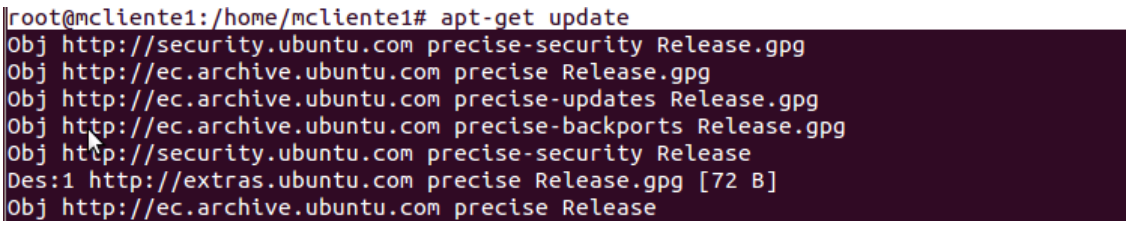
b.3. Comentar (con el signo #) los source que inician con deb-src del archivo /etc/apt/sources.list, previo a actualizar el sistema operativo:

nano /etc/apt/sources.list

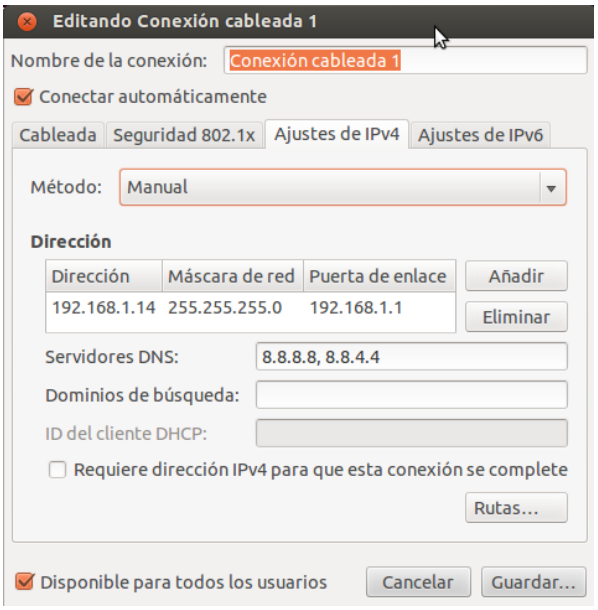


b.4. Actualizar el sistema operativo:

apt-get update



b.5. Configurar una ip fija para los equipos:



Reiniciar Red e Interfaz Ethernet

sudo /etc/init.d/networking restart (reiniciar red)

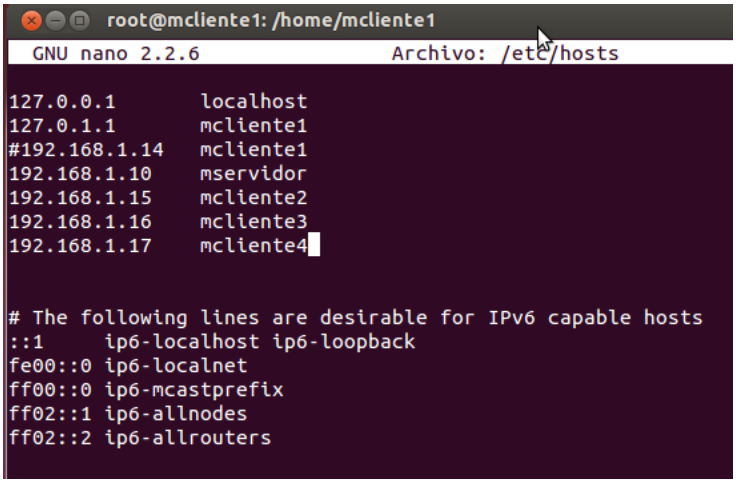
sudo ifconfig eth0 down

sudo ifconfig eth0 up

b.6. Agregar las direcciones ip de los equipos que se conectarán en la malla

nano /etc/hosts

```
root@mcliente1:/home/mcliente1# nano /etc/hosts
```



b.7. Instalar los paquetes requeridos para el manejo de los servicio Myproxy y GridFTP

apt-get install –y globus-gridftp myproxy globus-gram5

```
root@mcliente1:/home/mcliente1# apt-get install -y globus-gridftp myproxy globus-gram5
```

```
   /etc/rc5.d/S20globus-gridftp-server -> ../init.d/globus-gridftp-server
 Adding system startup for /etc/init.d/globus-gridftp-sshftp ...
   /etc/rc0.d/K20globus-gridftp-sshftp -> ../init.d/globus-gridftp-sshftp
   /etc/rc1.d/K20globus-gridftp-sshftp -> ../init.d/globus-gridftp-sshftp
   /etc/rc6.d/K20globus-gridftp-sshftp -> ../init.d/globus-gridftp-sshftp
   /etc/rc2.d/S20globus-gridftp-sshftp -> ../init.d/globus-gridftp-sshftp
   /etc/rc3.d/S20globus-gridftp-sshftp -> ../init.d/globus-gridftp-sshftp
   /etc/rc4.d/S20globus-gridftp-sshftp -> ../init.d/globus-gridftp-sshftp
   /etc/rc5.d/S20globus-gridftp-sshftp -> ../init.d/globus-gridftp-sshftp
 * Started GridFTP Server
Enabling sshftp access to globus-gridftp-server: Successfully created /etc/gridftp-sshftp.
OK
Configurando globus-gsi-cert-utils-progs (8.6-1) ...
Configurando globus-gss-assist-progs (9.0-2) ...
Configurando libmyproxy5 (5.9-10) ...
Configurando myproxy (5.9-10) ...
Configurando globus-gridftp (0.0.0) ...
Configurando globus-gram-job-manager-fork (1.5-5) ...
Configurando globus-gram-job-manager-fork-doc (1.5-5) ...
Configurando globus-gram-job-manager-fork-setup-poll (1.5-5) ...
Configurando globus-gram5 (0.0.0) ...
Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
```

b.8. Configurar seguridad

Aceptación de las certificaciones Simple CA

Se pide al servidor que autentifique al cliente1, devolviéndole un certificado el cual es instalado en dicho cliente.

myproxy-get-trustroots -b -s mservidor

```
root@mcliente1:/home/mcliente1# myproxy-get-trustroots -b -s mservidor
Bootstrapping MyProxy server root of trust.
New trusted MyProxy server: /O=Grid/OU=GlobusTest/OU=simpleCA-mservidor/CN=mservidor
New trusted CA (8a90862e.0): /O=Grid/OU=GlobusTest/OU=simpleCA-mservidor/CN=Globus Simple
CA
Trust roots have been installed in /etc/grid-security/certificates/.
```

b.9. Iniciar servicio de gridftp de globus toolkit

service globus-gridftp-server start

```
root@mcliente1:/home/mcliente1# service globus-gridftp-server start
 * Started GridFTP Server
root@mcliente1:/home/mcliente1#
```

Configurar gridftp para que éste arranque al igual que el cliente

chkconfig globus-gridftp-server on

```
root@mcliente1:~# chkconfig globus-gridftp-server on
root@mcliente1:~#
```

Verificar que las conexiones activos en el cliente (tcp), y cual se encuentra asociado a globus-gridftp

netstat –antupl | grep 2811

```
root@mcliente1:~# netstat -antupl | grep 2811
tcp        0      0 0.0.0.0:2811            0.0.0.0:*               ESCUCHAR    1058/globus-
gridftp
```

b.10. Si el firewall está activo, debe agregarse el puerto por el cual se escucha a globus-gridftp en su configuración

firewall-cmd –zone=public –add-port=1058/tcp –permanent

firewall-cmd –zone=public –add-port=1058/tcp –permanent

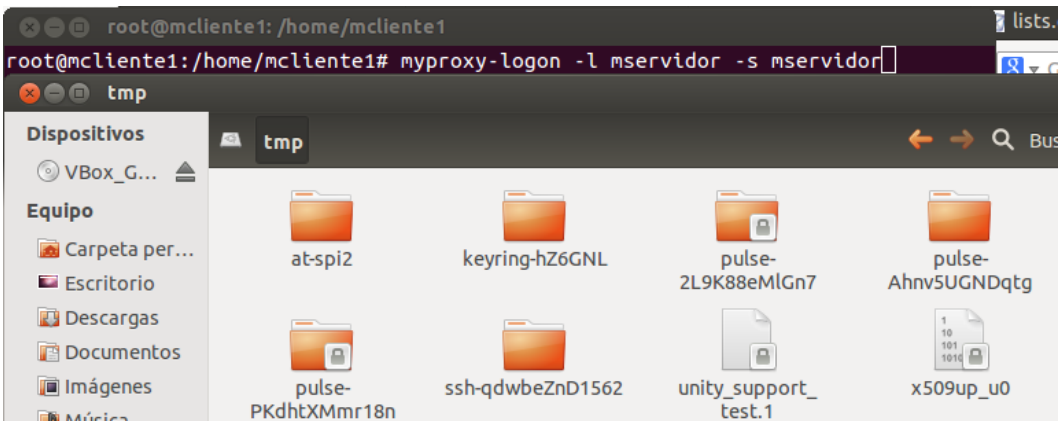
firewall-cmd --reload

b.11. Iniciar las autentificaciones, a través del servidor myproxy

su - -s /bin/sh mcliente1 [opcional]

myproxy-logon -l mservidor -s mservidor

```
root@mcliente1:/home/mcliente1# myproxy-logon -l mservidor -s mservidor
Enter MyProxy pass phrase:
A credential has been received for user mservidor in /tmp/x509up_u0.
```

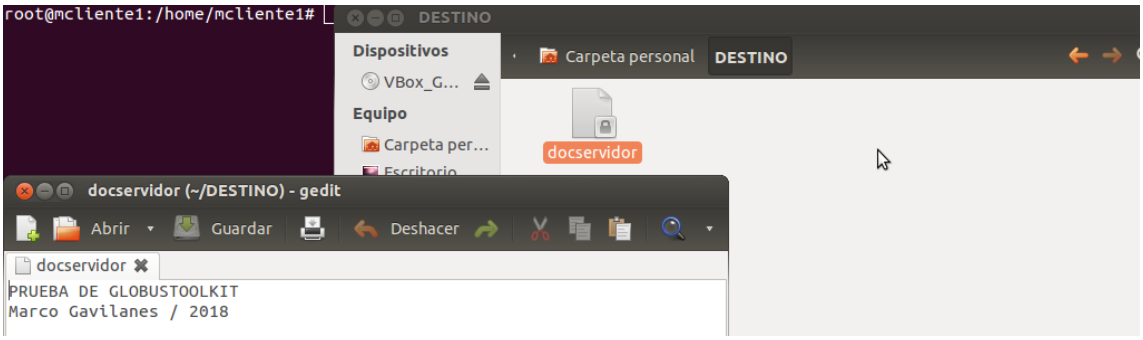


b.12. Enviar archivo a servidor, mediante gsiftp de globus toolkit

globus-url-copy gsiftp://mservidor/home/mservidor/ORIGEN/test file:///home/mcliente1/DESTINO/docservidor

```
root@mcliente1:/home/mcliente1# globus-url-copy gsiftp://mservidor/home/mservido
r/ORIGEN/test file:///home/mcliente1/DESTINO/docservidor
root@mcliente1:/home/mcliente1#
```

b.13. Verificar recepción del archivo



A fin de examinar el proceso de comunicación, se emplea la opción –dbg

```
root@mcliente1:/home/mcliente1# globus-url-copy -dbg gsiftp://mservidor/home/mse
rvidor/ORIGEN/test file:///home/mcliente1/DESTINO/docservidor
debug: starting to get gsiftp://mservidor/home/mservidor/ORIGEN/test
debug: connecting to gsiftp://mservidor/home/mservidor/ORIGEN/test
debug: response from gsiftp://mservidor/home/mservidor/ORIGEN/test:
220 mservidor GridFTP Server 6.43 (gcc32, 1396993756-83) [Globus Toolkit 5.2.6rc
0] ready.

debug: authenticating with gsiftp://mservidor/home/mservidor/ORIGEN/test
debug: response from gsiftp://mservidor/home/mservidor/ORIGEN/test:
230 User mservidor logged in.

debug: sending command to gsiftp://mservidor/home/mservidor/ORIGEN/test:
SITE HELP

debug: response from gsiftp://mservidor/home/mservidor/ORIGEN/test:
214-The following commands are recognized:
    ALLO    APPE    REST    CWD     CDUP    DCAU    EPSV    FEAT
    ERET    MDTM    STAT    ESTO    HELP    LIST    MODE    NLST
    MLSC    MLSD    PASV    RNFR    MLSR    MLST    NOOP    OPTS
    STOR    PASS    PBSZ    PORT    PROT    SITE    EPRT    RETR
    SPOR    MFMT    SCKS    TREV    PWD     QUIT    SBUF    SIZE
    SPAS    STRU    SYST    RNTO    TYPE    USER    LANG    MKD
    RMD     DELE    CKSM    DCSC
214 End
```

b.14. Ejecutar procesos en el servidor

globus-job-run mservidor/jobmanager-fork-poll /bin/hostname

```
$ globus-job-run mservidor/jobmanager-fork-poll /bin/hostname
mservidor
$
```

c. Verificación usuario no autorizado

Se mostrará el escenario en el que un usuario no autorizado intenta obtener un archivo desde el servidor.

En este caso, se usará el cliente2. En este cliente se aplicarán todos los pasos de la sección B, con la consideración que la clave de paso es desconocida para él, y en el paso b10 se obtendrá el siguiente resultado:

```
root@mcliente2:/home/mcliente2# myproxy-logon -l mservidor -s mservidor
Enter MyProxy pass phrase:
Failed to receive credentials.
ERROR from myproxy-server:
invalid credential passphrase
```

El cliente tiene iniciado el servicio gridftp:

```
root@mcliente2:/home/mcliente2# service globus-gridftp-server status
GridFTP Server Running (pid=944)
```

Existe conexión entre cliente y servidor, tal como puede observarse a continuación:

```
root@mcliente2:/home/mcliente2# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_req=1 ttl=64 time=6.92 ms
64 bytes from 192.168.1.10: icmp_req=2 ttl=64 time=15.8 ms
64 bytes from 192.168.1.10: icmp_req=3 ttl=64 time=10.2 ms
64 bytes from 192.168.1.10: icmp_req=4 ttl=64 time=7.73 ms
64 bytes from 192.168.1.10: icmp_req=5 ttl=64 time=9.91 ms
64 bytes from 192.168.1.10: icmp_req=6 ttl=64 time=6.06 ms
```

Al intentar el envío del archivo mediante gsiftp de globus toolkit, dentro de la misma red, aparecerá el siguiente error:

```
root@mcliente2:/home/mcliente2# globus-url-copy gsiftp://mservidor/home/mservido
r/ORIGEN/test file:///home/mcliente2/DESTINO/docservidor

error: globus_ftp_control: gss_init_sec_context failed
globus_gsi_gssapi: Error with gss credential handle
globus_credential: Valid credentials could not be found in any of the possible l
ocations specified by the credential search order.
Valid credentials could not be found in any of the possible locations specified
by the credential search order.
Attempt 1
globus_credential: Error reading host credential
globus_sysconfig: Could not find a valid certificate file: The host cert could n
ot be found in:
1) env. var. X509_USER_CERT
2) /etc/grid-security/hostcert.pem
3) $GLOBUS_LOCATION/etc/hostcert.pem
4) $HOME/.globus/hostcert.pem

The host key could not be found in:
1) env. var. X509_USER_KEY
2) /etc/grid-security/hostkey.pem
3) $GLOBUS_LOCATION/etc/hostkey.pem
4) $HOME/.globus/hostkey.pem


Attempt 2
globus_credential: Error reading proxy credential
globus_sysconfig: Could not find a valid proxy certificate file location
globus_sysconfig: Error with key filename
globus_sysconfig: File does not exist: /tmp/x509up_u0 is not a valid file
Attempt 3
globus_credential: Error reading user credential
globus_sysconfig: Error with certificate filename: The user cert could not be fo
und in:
1) env. var. X509_USER_CERT
2) $HOME/.globus/usercert.pem
3) $HOME/.globus/usercred.p12
```

Anexo B: Actividades Generales – Ejecución del Experimento

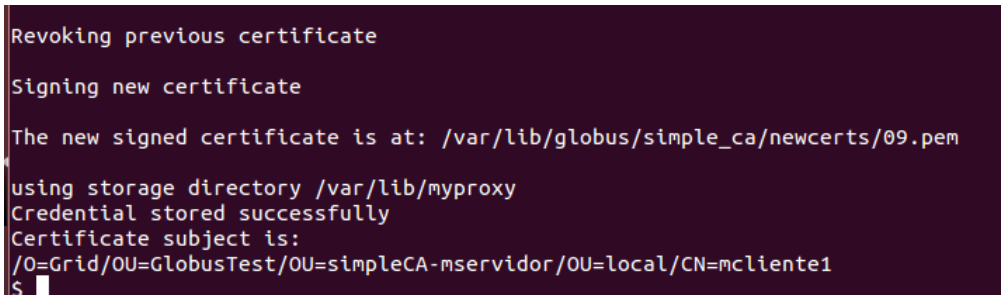
CONEXIÓN DE CLIENTES A RED GRID - GLOBUS TOOLKIT

1. Repetir los pasos del punto a.14 al a.16 del anexo I: "Configuración de Globus Toolkit en red experimental" para cada uno de los clientes en grid de la red (total 4 clientes)

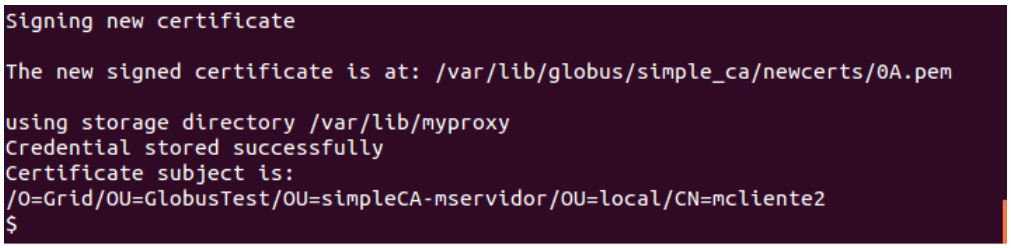
2. Las credenciales generadas durante el proceso se muestran a continuación

Cliente1

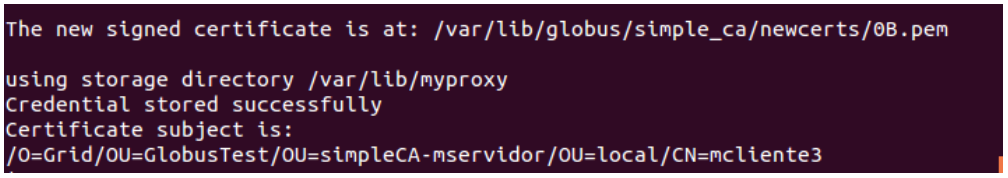
Las pruebas de configuración de Globus Toolkit fueron realizadas con el Cliente 1, razón por la cual al solicitar nuevamente la credencial globus revoca el certificado anterior y genera uno nuevo.



Cliente2



Cliente3



Cliente4

```
The new signed certificate is at: /var/lib/globus/simple_ca/newcerts/0C.pem

using storage directory /var/lib/myproxy
Credential stored successfully
Certificate subject is:
/O=Grid/OU=GlobusTest/OU=simpleCA-mservidor/OU=local/CN=mcliente4
```

3. En el servidor, se identificó el puerto por cual se comunica glogus-gridftp. En este caso, se está escuchando a través del puerto tcp 3835.

```
root@mservidor:/# service globus-gridftp-server status
GridFTP Server Running (pid=3835)
root@mservidor:/# netstat -antupl | grep 3835
tcp        0      0 0.0.0.0:2811            0.0.0.0:*               ESCUCHAR    3835/globus-gridftp
udp        0      0 0.0.0.0:40289           0.0.0.0:*                           3835/globus-gridftp
```

ENVIÓ DE PROCESOS EN GRID MEDIANTE WSGRAM

Obtener certificado de usuario

4. Obtener certificado que identifique el usuario mservidor

grid-cert-request

```
root@mservidor:/etc/grid-security# grid-cert-request

    /root/.globus/usercert_request.pem already exists
    /root/.globus/usercert.pem already exists
    /root/.globus/userkey.pem already exists
```

Inmediatamente se ingresó como nombre “Servidor” y una clave.

5. Firmar el certificado mediante el comando (ubicándose previamente en la ruta de los certificados de globus: /root/.globus

grid-ca-sign –in usercert_request.pem –out usercert.pem

```
root@mservidor:~/.globus# ls
nusercert.pem  usercert.pem  usercert_request.pem  userkey.pem
root@mservidor:~/.globus# ls
nusercert.pem  usercert.pem  usercert_request.pem  userkey.pem
root@mservidor:~/.globus# grid-ca-sign -in usercert_request.pem -out usercert.pem
```

```
Signing new certificate

The new signed certificate is at: /var/lib/globus/simple_ca/newcerts/12.pem
```

6. Copiar el certificado firmado en la ruta de globus

cp /var/lib/globus/simple_ca/newcerts/12.pem /root/.globys/usercer.pem

```
root@mservidor:~/.globus# cp /var/lib/globus/simple_ca/newcerts/12.pem /root/.globus/usercert.pem
```

7. Verificar el contenido del certificado y el estado del proxy, mediante los comandos

grid-cert-info –all

```
root@mservidor:~/.globus# grid-cert-info -all
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 18 (0x12)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: O=Grid, OU=GlobusTest, OU=simpleCA-mservidor, CN=Globus Simple CA
        Validity
            Not Before: May 23 01:38:45 2018 GMT
            Not After : May 23 01:38:45 2019 GMT
        Subject: O=Grid, OU=GlobusTest, OU=simpleCA-mservidor, OU=local, CN=Servidor
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                    00:de:09:1b:25:3d:d5:cc:ff:35:5a:0e:fb:7c:32:
                    12:f8:54:ae:cf:18:79:ea:7f:44:71:b3:8e:c3:96:
                    e2:5d:1a:c5:be:03:d1:dc:f4:17:67:25:1d:a5:9b:
                    99:19:77:75:ae:eb:e7:12:b3:8c:c9:74:0b:b2:2a:
                    63:2d:2f:9d:12:76:64:ec:bc:25:a1:60:06:61:07:
                    1f:df:f6:c4:01:2b:41:e7:53:8b:9c:12:ae:dc:99:
                    5c:d9:c5:fc:23:2c:52:38:1e:2a:16:b4:8c:46:73:
                    71:cb:03:f8:1d:d0:2a:67:40:aa:4c:f1:17:1a:41:
                    28:0e:8b:4f:ed:6e:0e:a4:27
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Netscape Cert Type:
                SSL Client, SSL Server, S/MIME, Object Signing
    Signature Algorithm: sha1WithRSAEncryption
         0c:25:48:d5:7b:b1:2d:3c:47:f4:f7:fb:3c:14:1f:22:67:8b:
```

grid-proxy-info

```
root@mservidor:~/.globus# grid-proxy-info
subject  : /O=Grid/OU=GlobusTest/OU=simpleCA-mservidor/OU=local/CN=Servidor/CN=546773281
issuer   : /O=Grid/OU=GlobusTest/OU=simpleCA-mservidor/OU=local/CN=Servidor
identity : /O=Grid/OU=GlobusTest/OU=simpleCA-mservidor/OU=local/CN=Servidor
type     : RFC 3820 compliant impersonation proxy
strength : 1024 bits
path     : /tmp/x509up_u0
timeleft : 9:58:31
```

8. Iniciar el grid-proxy mediante el siguiente comando

grid-proxy-init

```
root@mservidor:~/.globus# grid-proxy-init
Your identity: /O=Grid/OU=GlobusTest/OU=simpleCA-mservidor/OU=local/CN=Servidor
Enter GRID pass phrase for this identity:
Creating proxy ..................................................... Done
Your proxy is valid until: Wed May 23 08:49:05 2018
root@mservidor:~/.globus#
```

Obtener certificado de host

9. Obtener certificado que identifique al host

grid-cert-request –host mservidor

```
root@mservidor:~/.globus# grid-cert-request -host mservidor

    /etc/grid-security/hostcert_request.pem already exists
    /etc/grid-security/hostcert.pem already exists
    /etc/grid-security/hostkey.pem already exists
```

10. Firmar el certificado mediante el comando (ubicándose previamente en la ruta de los certificados de globus: /etc/grid-security

grid-ca-sign –in hostcert_request.pem –out hostcert.pem

```
root@mservidor:~/.globus# cd /etc/grid-security
root@mservidor:/etc/grid-security# grid-ca-sign -in hostcert_request.
pem -out hostcert.pem

Revoking previous certificate

Signing new certificate

The new signed certificate is at: /var/lib/globus/simple_ca/newcerts/
13.pem
```

11. Copiar el certificado firmado en la ruta de globus

cp /var/lib/globus/simple_ca/newcerts/13.pem /etc/grid-security/hostcert.pem

```
root@mservidor:/etc/grid-security# cp /var/lib/globus/simple_ca/newce
rts/13.pem /etc/grid-security/hostcert.pem
root@mservidor:/etc/grid-security#
```

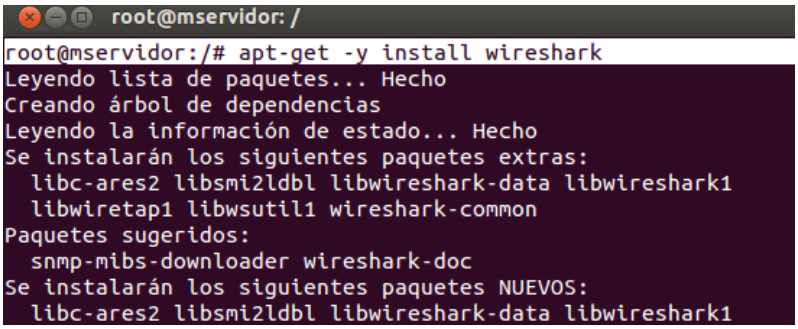
MONITOREO DE RED

Wireshark

12. Instalar y configurar Wireshark

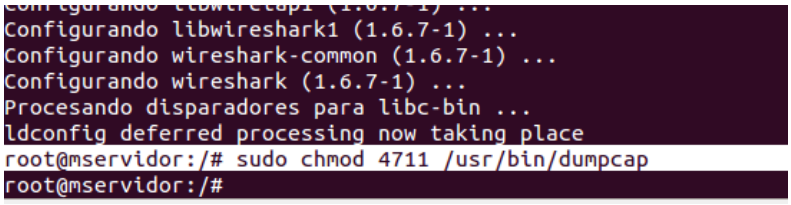
    a. Ejecutar el comando

    sudo apt-get –y install wireshark

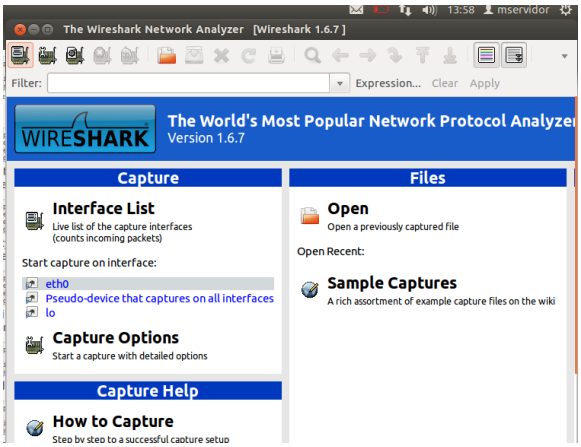


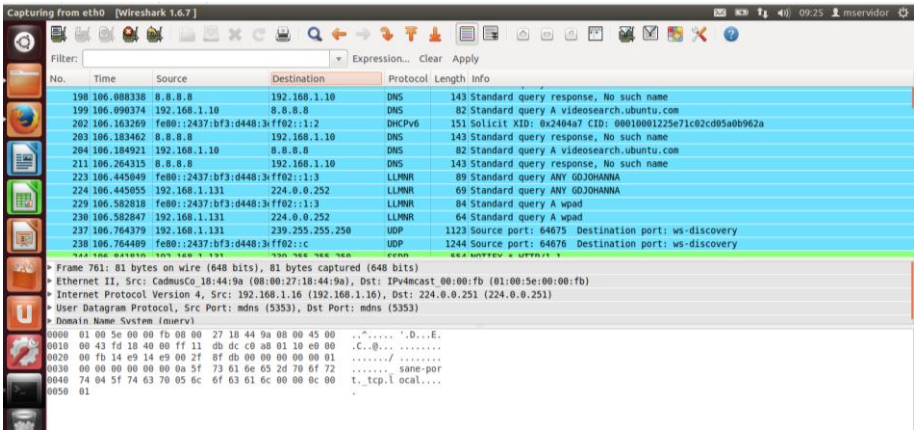
    b. Activar permisos al ejecutado dumpcap

    sudo chmod 4711 /usr/bin/dumpcap



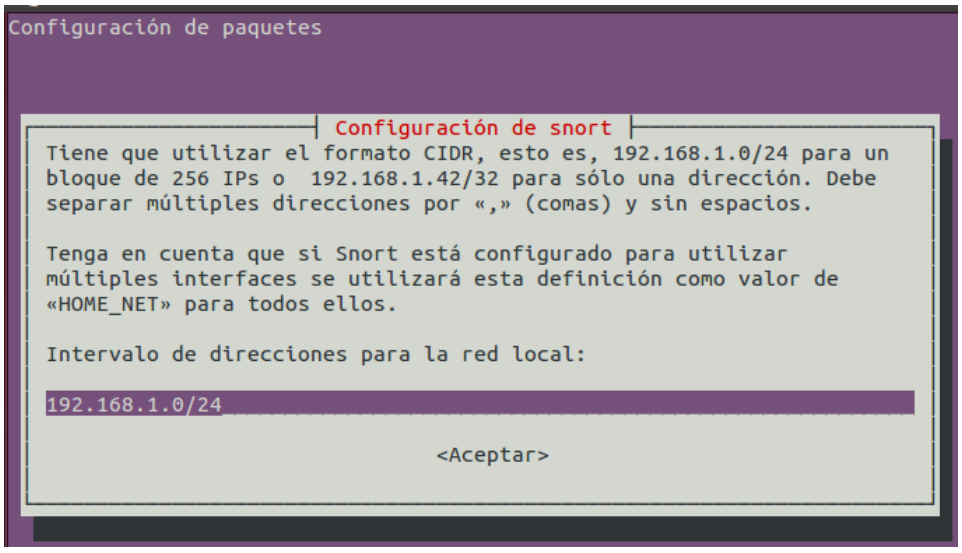
    c. Ejecutar aplicación y seleccionar la interfaz eth0 para la monitorización de la red

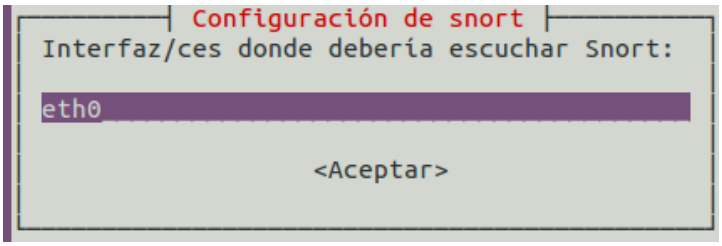




Snort

13. Instalar y Configurar snort para control de la interfaz eth0, en la red 192.168.1.0/24 (ver apartado 4.4.1 para la descripción completa de los pasos de instalación)



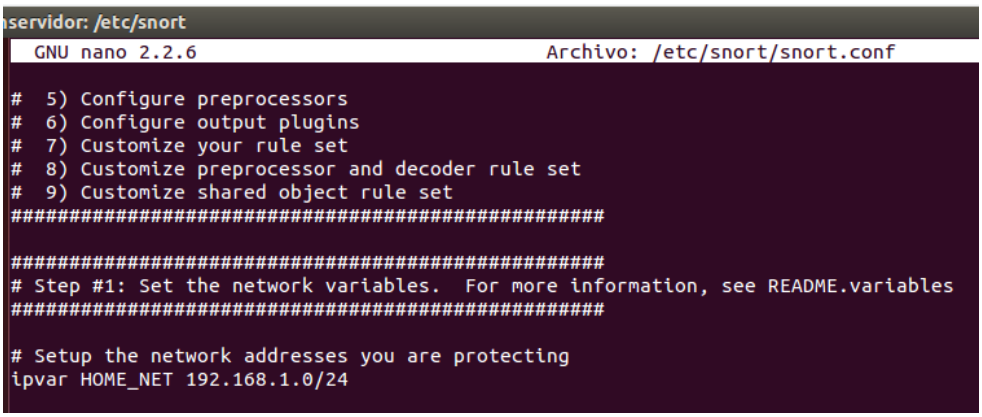


14. Registrar regla de monitorización

En la fase de configuración del ambiente de pruebas, se consideró la inclusión de la siguiente regla de monitoreo del puerto 7512, el cual es uno de los puertos empleados por globus para sus conexiones tcp:

alert tcp any 7512 -> any any (msg:"Alguien intenta leer el Puerto 7512";content:"globus puerto tcp";sid:1000;rev:1;)

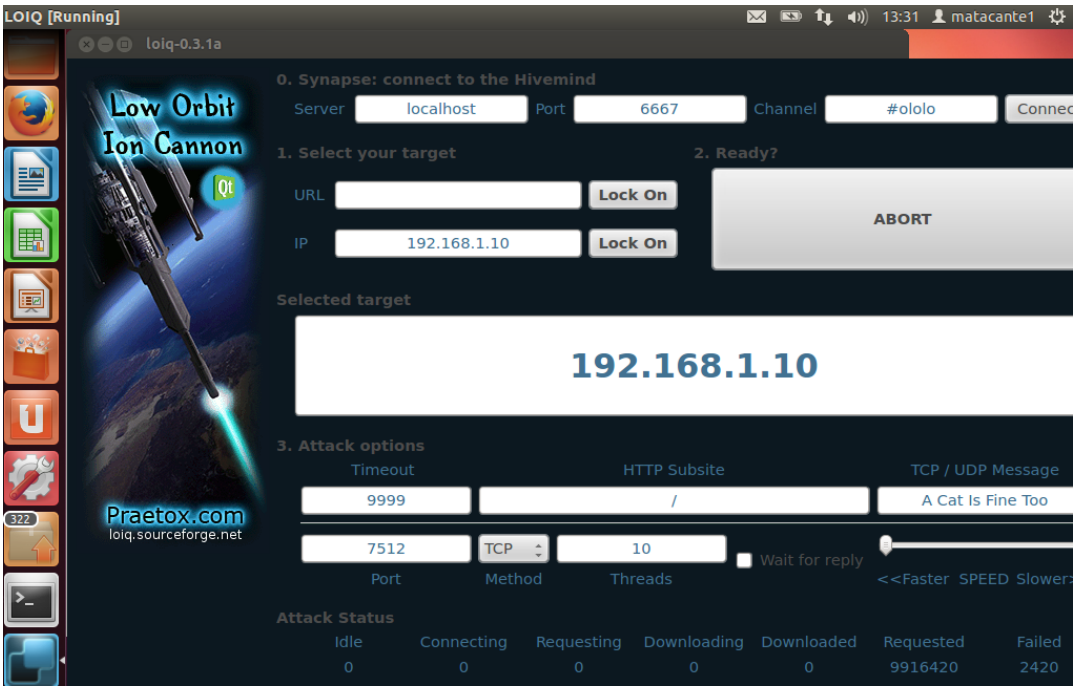
15. Configurar HOME NET de snort_conf con la dirección de red que se está monitorizando



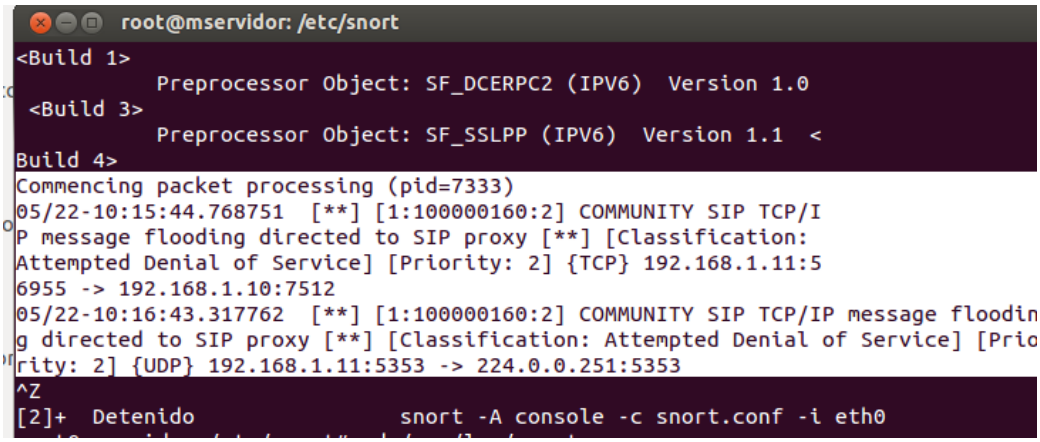
16. Iniciar monitorización

```
root@mservidor:/etc/snort# snort -A console -c snort.conf -i eth0
```

17. En los equipos atacantes se ejecuta la aplicación LOIQ, y se generó un ataque de denegación de servicio al equipo servidor (192.168.1.10), a los puertos TCP 2811, 7512, 2119; 10 paquetes enviados.



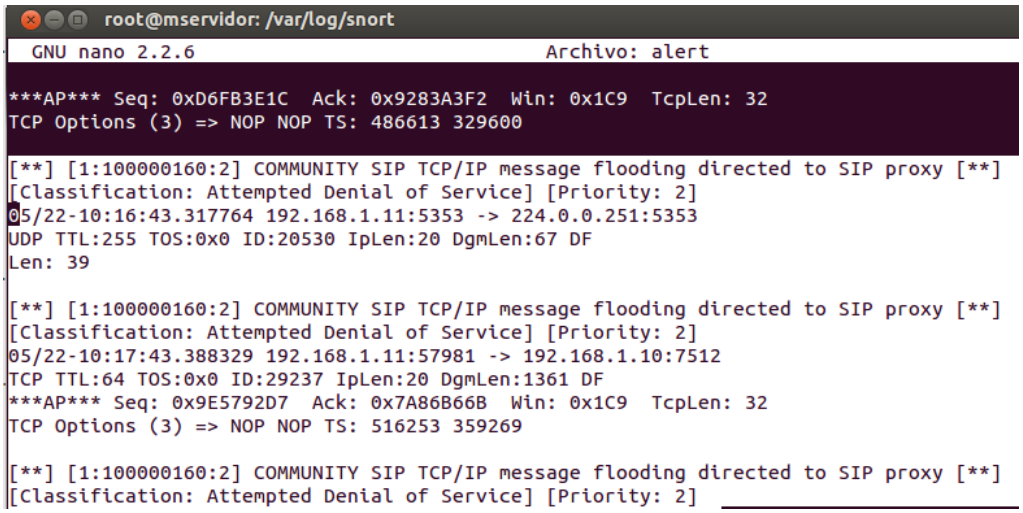
18. Al detener la monitorización, pudo observarse el monitoreo correspondiente al procesamiento de paquetes



El mensaje, al ser de tipo COMMUNITY, permitió establecer la existencia de una regla de carácter superior para la el análisis de ataques de denegación de servicio.

Al revisar el log de alertas se obtuvieron los siguientes resultados:

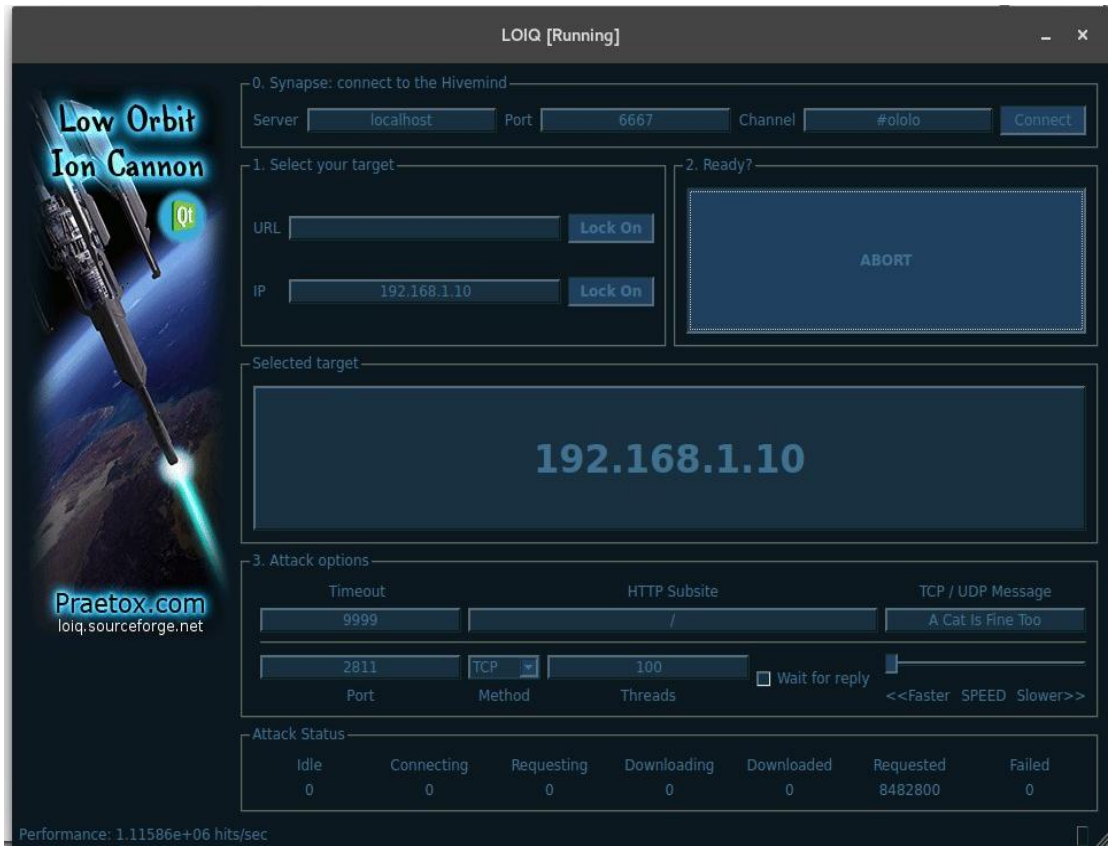
Ruta: */var/log/snort/alert*

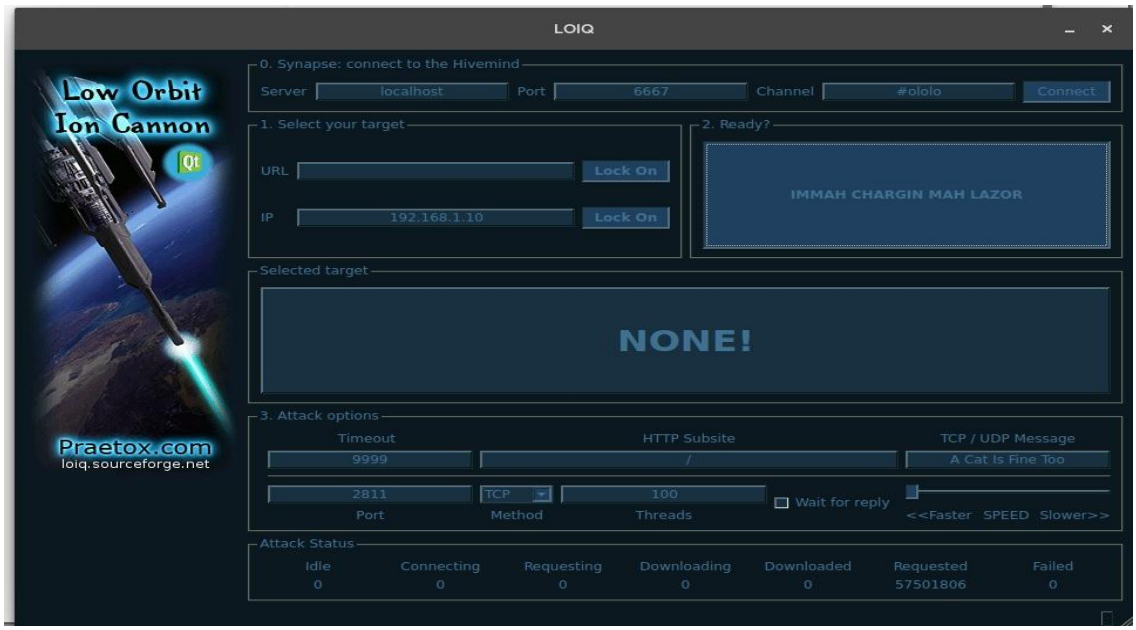

root@mservidor: /var/log/snort
GNU nano 2.2.6 Archivo: alert
AP Seq: 0xD6FB3E1C Ack: 0x9283A3F2 Win: 0x1C9 TcpLen: 32
TCP Options (3) => NOP NOP TS: 486613 329600
[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**]
[Classification: Attempted Denial of Service] [Priority: 2]
05/22-10:16:43.317764 192.168.1.11:5353 -> 224.0.0.251:5353
UDP TTL:255 TOS:0x0 ID:20530 IpLen:20 DgmLen:67 DF
Len: 39
[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**]
[Classification: Attempted Denial of Service] [Priority: 2]
05/22-10:17:43.388329 192.168.1.11:57981 -> 192.168.1.10:7512
TCP TTL:64 TOS:0x0 ID:29237 IpLen:20 DgmLen:1361 DF
AP Seq: 0x9E5792D7 Ack: 0x7A86B66B Win: 0x1C9 TcpLen: 32
TCP Options (3) => NOP NOP TS: 516253 359269
[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**]
[Classification: Attempted Denial of Service] [Priority: 2]

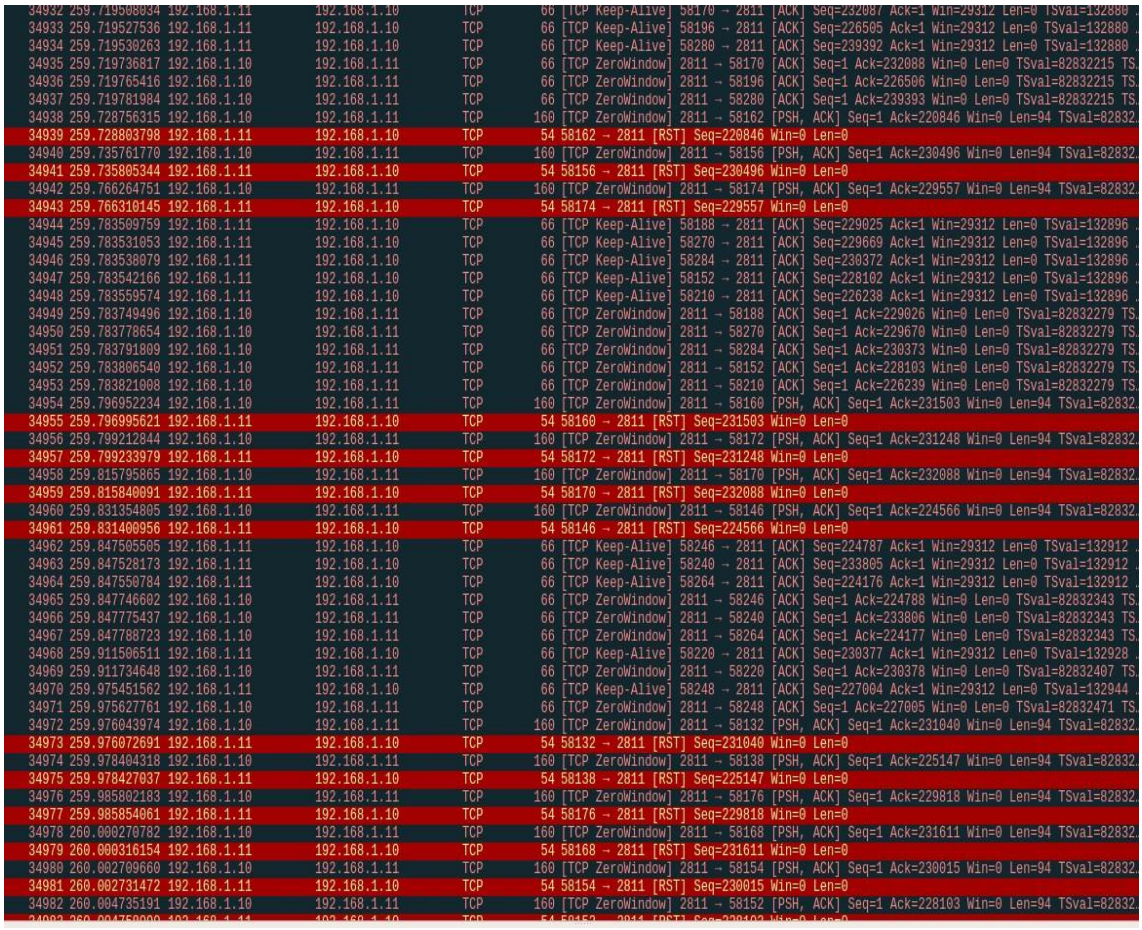
Anexo C: Puertos abiertos en el servidor vulnerable

```
root@mservidor:/# nmap 192.168.1.10

Nmap scan report for 192.168.1.10
Host is up (0.00093s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
2119/tcp open  gsigatekeeper
2811/tcp open  gsiftp
7512/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned
```

Anexo D: Ataque de denegación de servicios con Loiq






34932 259.719508034 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58170 → 2811 [ACK] Seq=232087 Ack=1 Win=29312 Len=0 TSval=132880
34933 259.719527536 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58196 → 2811 [ACK] Seq=226505 Ack=1 Win=29312 Len=0 TSval=132880
34934 259.719530263 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58280 → 2811 [ACK] Seq=239392 Ack=1 Win=29312 Len=0 TSval=132880
34935 259.719736817 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58170 [ACK] Seq=1 Ack=232088 Win=0 Len=0 TSval=82832215 TS.
34936 259.719765416 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58196 [ACK] Seq=1 Ack=226506 Win=0 Len=0 TSval=82832215 TS.
34937 259.719781984 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58280 [ACK] Seq=1 Ack=239393 Win=0 Len=0 TSval=82832215 TS.
34938 259.728756315 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58162 [PSH, ACK] Seq=1 Ack=220846 Win=0 Len=94 TSval=82832.
34939 259.728803798 192.168.1.11 192.168.1.10 TCP 54 58162 → 2811 [RST] Seq=220846 Win=0 Len=0
34940 259.735761770 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58156 [PSH, ACK] Seq=1 Ack=230496 Win=0 Len=94 TSval=82832.
34941 259.735805344 192.168.1.11 192.168.1.10 TCP 54 58156 → 2811 [RST] Seq=230496 Win=0 Len=0
34942 259.766264751 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58174 [PSH, ACK] Seq=1 Ack=229557 Win=0 Len=94 TSval=82832.
34943 259.766310145 192.168.1.11 192.168.1.10 TCP 54 58174 → 2811 [RST] Seq=229557 Win=0 Len=0
34944 259.783509759 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58188 → 2811 [ACK] Seq=229025 Ack=1 Win=29312 Len=0 TSval=132896
34945 259.783531053 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58270 → 2811 [ACK] Seq=229669 Ack=1 Win=29312 Len=0 TSval=132896
34946 259.783538079 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58284 → 2811 [ACK] Seq=230372 Ack=1 Win=29312 Len=0 TSval=132896
34947 259.783542166 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58152 → 2811 [ACK] Seq=228102 Ack=1 Win=29312 Len=0 TSval=132896
34948 259.783559574 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58210 → 2811 [ACK] Seq=226238 Ack=1 Win=29312 Len=0 TSval=132896
34949 259.783749496 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58188 [ACK] Seq=1 Ack=229026 Win=0 Len=0 TSval=82832279 TS.
34950 259.783778654 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58270 [ACK] Seq=1 Ack=229670 Win=0 Len=0 TSval=82832279 TS.
34951 259.783791809 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58284 [ACK] Seq=1 Ack=230373 Win=0 Len=0 TSval=82832279 TS.
34952 259.783806540 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58152 [ACK] Seq=1 Ack=228103 Win=0 Len=0 TSval=82832279 TS.
34953 259.783821008 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58210 [ACK] Seq=1 Ack=226239 Win=0 Len=0 TSval=82832279 TS.
34954 259.796952234 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58160 [PSH, ACK] Seq=1 Ack=231503 Win=0 Len=94 TSval=82832.
34955 259.796995621 192.168.1.11 192.168.1.10 TCP 54 58160 → 2811 [RST] Seq=231503 Win=0 Len=0
34956 259.799212844 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58172 [PSH, ACK] Seq=1 Ack=231248 Win=0 Len=94 TSval=82832.
34957 259.799233979 192.168.1.11 192.168.1.10 TCP 54 58172 → 2811 [RST] Seq=231248 Win=0 Len=0
34958 259.815795865 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58170 [PSH, ACK] Seq=1 Ack=232088 Win=0 Len=94 TSval=82832.
34959 259.815840091 192.168.1.11 192.168.1.10 TCP 54 58170 → 2811 [RST] Seq=232088 Win=0 Len=0
34960 259.831354805 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58146 [PSH, ACK] Seq=1 Ack=224566 Win=0 Len=94 TSval=82832.
34961 259.831400956 192.168.1.11 192.168.1.10 TCP 54 58146 → 2811 [RST] Seq=224566 Win=0 Len=0
34962 259.847505505 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58246 → 2811 [ACK] Seq=224787 Ack=1 Win=29312 Len=0 TSval=132912
34963 259.847528173 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58240 → 2811 [ACK] Seq=233805 Ack=1 Win=29312 Len=0 TSval=132912
34964 259.847550784 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58264 → 2811 [ACK] Seq=224176 Ack=1 Win=29312 Len=0 TSval=132912
34965 259.847746602 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58246 [ACK] Seq=1 Ack=224788 Win=0 Len=0 TSval=82832343 TS.
34966 259.847775437 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58240 [ACK] Seq=1 Ack=233806 Win=0 Len=0 TSval=82832343 TS.
34967 259.847788723 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58264 [ACK] Seq=1 Ack=224177 Win=0 Len=0 TSval=82832343 TS.
34968 259.911506511 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58220 → 2811 [ACK] Seq=230377 Ack=1 Win=29312 Len=0 TSval=132928
34969 259.911734648 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58220 [ACK] Seq=1 Ack=230378 Win=0 Len=0 TSval=82832407 TS.
34970 259.975451562 192.168.1.11 192.168.1.10 TCP 66 [TCP Keep-Alive] 58248 → 2811 [ACK] Seq=227004 Ack=1 Win=29312 Len=0 TSval=132944
34971 259.975627761 192.168.1.10 192.168.1.11 TCP 66 [TCP ZeroWindow] 2811 → 58248 [ACK] Seq=1 Ack=227005 Win=0 Len=0 TSval=82832471 TS.
34972 259.976043974 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58132 [PSH, ACK] Seq=1 Ack=231040 Win=0 Len=94 TSval=82832.
34973 259.976072691 192.168.1.11 192.168.1.10 TCP 54 58132 → 2811 [RST] Seq=231040 Win=0 Len=0
34974 259.978404318 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58138 [PSH, ACK] Seq=1 Ack=225147 Win=0 Len=94 TSval=82832.
34975 259.978427037 192.168.1.11 192.168.1.10 TCP 54 58138 → 2811 [RST] Seq=225147 Win=0 Len=0
34976 259.985802183 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58176 [PSH, ACK] Seq=1 Ack=229818 Win=0 Len=94 TSval=82832.
34977 259.985854061 192.168.1.11 192.168.1.10 TCP 54 58176 → 2811 [RST] Seq=229818 Win=0 Len=0
34978 260.000270782 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58168 [PSH, ACK] Seq=1 Ack=231611 Win=0 Len=94 TSval=82832.
34979 260.000316154 192.168.1.11 192.168.1.10 TCP 54 58168 → 2811 [RST] Seq=231611 Win=0 Len=0
34980 260.002709660 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58154 [PSH, ACK] Seq=1 Ack=230015 Win=0 Len=94 TSval=82832.
34981 260.002731472 192.168.1.11 192.168.1.10 TCP 54 58154 → 2811 [RST] Seq=230015 Win=0 Len=0
34982 260.004735191 192.168.1.10 192.168.1.11 TCP 160 [TCP ZeroWindow] 2811 → 58152 [PSH, ACK] Seq=1 Ack=228103 Win=0 Len=94 TSval=82832.

Anexo E: Intercepción de paquetes en la red

```
[root@cliente1 ~]# su - -s /bin/sh guser
Último inicio de sesión:jue oct 19 09:48:32 -05 2017en tty1
-sh-4.2$ globus-job-run server.globus.com/jobmanager-fork-poll /bin/python /bin/ex1.py
]
```

4194..	172.694944580	192.168.1.10	192.168.1.11	TCP	66 22 → 53600 [FIN, ACK] Seq=141146877 Ack=68198 Win=312 Len=0 TSval=83706314 TSecr=3..
4194..	172.694971725	192.168.1.11	192.168.1.10	TCP	66 53600 → 22 [ACK] Seq=68198 Ack=141146878 Win=17695 Len=0 TSval=366461 TSecr=837063..
4194..	188.067235956	192.168.1.10	8.8.8.8	DNS	82 Standard query 0xf0bc A usage-stats.globus.org
4194..	188.067549265	192.168.1.10	8.8.8.8	DNS	82 Standard query 0x4e70 AAAA usage-stats.globus.org
4194..	188.710274442	HuaweiTe_a8:39:01	Broadcast	ARP	42 Who has 192.168.1.10? Tell 192.168.1.1

Destination	Protocol	Length	Info
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3205 Ack=95533 Win=1444 Len=0 TSval=346465 TSecr=83686353
192.168.1.10	SSH	110	Client: Encrypted packet (len=44)
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3249 Ack=95649 Win=1444 Len=0 TSval=346569 TSecr=83686768
192.168.1.10	SSH	102	Client: Encrypted packet (len=36)
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=95685 Win=1444 Len=0 TSval=346973 TSecr=83688385
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=95825 Win=1444 Len=0 TSval=346976 TSecr=83688396
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=95893 Win=1444 Len=0 TSval=346991 TSecr=83688458
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=96537 Win=1444 Len=0 TSval=347006 TSecr=83688514
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=96629 Win=1444 Len=0 TSval=347006 TSecr=83688516
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=96721 Win=1444 Len=0 TSval=347006 TSecr=83688516
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=96813 Win=1444 Len=0 TSval=347006 TSecr=83688516
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=96905 Win=1444 Len=0 TSval=347006 TSecr=83688516
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=96997 Win=1444 Len=0 TSval=347006 TSecr=83688516
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97089 Win=1444 Len=0 TSval=347006 TSecr=83688516
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97181 Win=1444 Len=0 TSval=347006 TSecr=83688516
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97273 Win=1444 Len=0 TSval=347006 TSecr=83688516
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97365 Win=1444 Len=0 TSval=347006 TSecr=83688517
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97457 Win=1444 Len=0 TSval=347006 TSecr=83688517
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97549 Win=1444 Len=0 TSval=347006 TSecr=83688517
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97641 Win=1444 Len=0 TSval=347006 TSecr=83688517
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97733 Win=1444 Len=0 TSval=347006 TSecr=83688517
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97825 Win=1444 Len=0 TSval=347006 TSecr=83688517
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=97917 Win=1444 Len=0 TSval=347006 TSecr=83688517
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98009 Win=1444 Len=0 TSval=347006 TSecr=83688517
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98101 Win=1444 Len=0 TSval=347006 TSecr=83688518
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98193 Win=1444 Len=0 TSval=347006 TSecr=83688518
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98285 Win=1444 Len=0 TSval=347006 TSecr=83688518
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98377 Win=1444 Len=0 TSval=347006 TSecr=83688518
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98469 Win=1444 Len=0 TSval=347006 TSecr=83688518
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98561 Win=1444 Len=0 TSval=347006 TSecr=83688518
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98653 Win=1444 Len=0 TSval=347006 TSecr=83688519
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98745 Win=1444 Len=0 TSval=347006 TSecr=83688519
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98837 Win=1444 Len=0 TSval=347007 TSecr=83688519
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=98929 Win=1444 Len=0 TSval=347007 TSecr=83688519
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99021 Win=1444 Len=0 TSval=347007 TSecr=83688519
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99113 Win=1444 Len=0 TSval=347007 TSecr=83688520
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99205 Win=1444 Len=0 TSval=347007 TSecr=83688520
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99297 Win=1444 Len=0 TSval=347007 TSecr=83688520
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99389 Win=1444 Len=0 TSval=347007 TSecr=83688520
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99481 Win=1444 Len=0 TSval=347007 TSecr=83688520
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99573 Win=1444 Len=0 TSval=347007 TSecr=83688521
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99665 Win=1444 Len=0 TSval=347007 TSecr=83688521
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99757 Win=1444 Len=0 TSval=347007 TSecr=83688521
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99849 Win=1444 Len=0 TSval=347007 TSecr=83688521
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=99941 Win=1444 Len=0 TSval=347007 TSecr=83688521
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=100033 Win=1444 Len=0 TSval=347007 TSecr=83688522
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=100125 Win=1444 Len=0 TSval=347007 TSecr=83688522
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=100217 Win=1444 Len=0 TSval=347007 TSecr=83688522
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=100309 Win=1444 Len=0 TSval=347007 TSecr=83688522
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=100401 Win=1444 Len=0 TSval=347007 TSecr=83688522
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=100493 Win=1444 Len=0 TSval=347007 TSecr=83688523
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=100585 Win=1444 Len=0 TSval=347007 TSecr=83688523
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=100677 Win=1444 Len=0 TSval=347008 TSecr=83688523
192.168.1.10	TCP	66	53600 → 22 [ACK] Seq=3285 Ack=100769 Win=1444 Len=0 TSval=347008 TSecr=83688523

Anexo F: Log Snort

```
10/19-09:44:39.322345 192.168.1.10:22 -> 192.168.1.11:53600
TCP TTL:64 TOS:0x10 ID:12965 IpLen:20 DgmLen:144 DF
***AP*** Seq: 0x71319B15  Ack: 0x80039132  Win: 0x138  TcpLen: 32
TCP Options (3) => NOP NOP TS: 83717242 354189
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/19-09:44:39.322375 192.168.1.11:53600 -> 192.168.1.10:22
TCP TTL:64 TOS:0x10 ID:32921 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x80039132  Ack: 0x71319B71  Win: 0x5A4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 354189 83717242
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/19-09:44:39.322440 192.168.1.10:22 -> 192.168.1.11:53600
TCP TTL:64 TOS:0x10 ID:12966 IpLen:20 DgmLen:144 DF
***AP*** Seq: 0x71319B71  Ack: 0x80039132  Win: 0x138  TcpLen: 32
TCP Options (3) => NOP NOP TS: 83717242 354189
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/19-09:44:39.322471 192.168.1.11:53600 -> 192.168.1.10:22
TCP TTL:64 TOS:0x10 ID:32922 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x80039132  Ack: 0x71319BCD  Win: 0x5A4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 354189 83717242
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/19-09:44:39.322534 192.168.1.10:22 -> 192.168.1.11:53600
TCP TTL:64 TOS:0x10 ID:12967 IpLen:20 DgmLen:144 DF
***AP*** Seq: 0x71319BCD  Ack: 0x80039132  Win: 0x138  TcpLen: 32
TCP Options (3) => NOP NOP TS: 83717242 354189
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/19-09:44:39.322565 192.168.1.11:53600 -> 192.168.1.10:22
TCP TTL:64 TOS:0x10 ID:32923 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x80039132  Ack: 0x71319C29  Win: 0x5A4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 354189 83717242
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/19-09:44:39.322629 192.168.1.10:22 -> 192.168.1.11:53600
TCP TTL:64 TOS:0x10 ID:12968 IpLen:20 DgmLen:144 DF
***AP*** Seq: 0x71319C29  Ack: 0x80039132  Win: 0x138  TcpLen: 32
TCP Options (3) => NOP NOP TS: 83717242 354189
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/19-09:44:39.322659 192.168.1.11:53600 -> 192.168.1.10:22
TCP TTL:64 TOS:0x10 ID:32924 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x80039132  Ack: 0x71319C85  Win: 0x5A4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 354189 83717242
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/19-09:12:20.512408 192.168.1.10:38106 -> 192.168.1.14:51552
TCP TTL:64 TOS:0x0 ID:1715 IpLen:20 DgmLen:397 DF
***AP*** Seq: 0xBC59E59B  Ack: 0x15E84E33  Win: 0x134  TcpLen: 32
TCP Options (3) => NOP NOP TS: 81778432 48726297
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/19-09:12:20.512776 192.168.1.14:51552 -> 192.168.1.10:38106
TCP TTL:64 TOS:0x0 ID:47634 IpLen:20 DgmLen:127 DF
***AP*** Seq: 0x15E84E33  Ack: 0xBC59E6F4  Win: 0x152  TcpLen: 32
TCP Options (3) => NOP NOP TS: 48726300 81778432
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/19-09:12:20.512910 192.168.1.14:51552 -> 192.168.1.10:38106
TCP TTL:64 TOS:0x0 ID:47635 IpLen:20 DgmLen:52 DF
***A***F Seq: 0x15E84E7E  Ack: 0xBC59E6F4  Win: 0x152  TcpLen: 32
TCP Options (3) => NOP NOP TS: 48726300 81778432
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/19-09:12:20.514879 192.168.1.10:38106 -> 192.168.1.14:51552
TCP TTL:64 TOS:0x0 ID:1716 IpLen:20 DgmLen:52 DF
***A***F Seq: 0xBC59E6F4  Ack: 0x15E84E7F  Win: 0x134  TcpLen: 32
TCP Options (3) => NOP NOP TS: 81778434 48726300
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/19-09:12:20.515317 192.168.1.14:51552 -> 192.168.1.10:38106
TCP TTL:64 TOS:0x0 ID:47636 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x15E84E7F  Ack: 0xBC59E6F5  Win: 0x152  TcpLen: 32
TCP Options (3) => NOP NOP TS: 48726303 81778434
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/19-09:12:24.463655 192.168.1.3:41451 -> 192.168.1.63:15600
UDP TTL:64 TOS:0x0 ID:26295 IpLen:20 DgmLen:63 DF
Len: 35
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

Anexo G: Detección del Atacante con snort

```
root@mservidor: /var/log/snort
 GNU nano 2.2.6                                  Archivo: alert

***AP*** Seq: 0xD6FB3E1C  Ack: 0x9283A3F2  Win: 0x1C9  TcpLen: 32
TCP Options (3) => NOP NOP TS: 486613 329600

[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**]
[Classification: Attempted Denial of Service] [Priority: 2]
05/22-10:16:43.317764 192.168.1.11:5353 -> 224.0.0.251:5353
UDP TTL:255 TOS:0x0 ID:20530 IpLen:20 DgmLen:67 DF
Len: 39

[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**]
[Classification: Attempted Denial of Service] [Priority: 2]
05/22-10:17:43.388329 192.168.1.11:57981 -> 192.168.1.10:7512
TCP TTL:64 TOS:0x0 ID:29237 IpLen:20 DgmLen:1361 DF
***AP*** Seq: 0x9E5792D7  Ack: 0x7A86B66B  Win: 0x1C9  TcpLen: 32
TCP Options (3) => NOP NOP TS: 516253 359269

[**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**]
[Classification: Attempted Denial of Service] [Priority: 2]
```

Anexo H: Información sobre el certificado SSL

```
root@mservidor:~/.globus# grid-cert-info -all
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 18 (0x12)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: O=Grid, OU=GlobusTest, OU=simpleCA-mservidor, CN=Globus Simple CA
        Validity
            Not Before: May 23 01:38:45 2018 GMT
            Not After : May 23 01:38:45 2019 GMT
        Subject: O=Grid, OU=GlobusTest, OU=simpleCA-mservidor, OU=local, CN=Servidor
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                    00:de:09:1b:25:3d:d5:cc:ff:35:5a:0e:fb:7c:32:
                    12:f8:54:ae:cf:18:79:ea:7f:44:71:b3:8e:c3:96:
                    e2:5d:1a:c5:be:03:d1:dc:f4:17:67:25:1d:a5:9b:
                    99:19:77:75:ae:eb:e7:12:b3:8c:c9:74:0b:b2:2a:
                    63:2d:2f:9d:12:76:64:ec:bc:25:a1:60:06:61:07:
                    1f:df:f6:c4:01:2b:41:e7:53:8b:9c:12:ae:dc:99:
                    5c:d9:c5:fc:23:2c:52:38:1e:2a:16:b4:8c:46:73:
                    71:cb:03:f8:1d:d0:2a:67:40:aa:4c:f1:17:1a:41:
                    28:0e:8b:4f:ed:6e:0e:a4:27
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Netscape Cert Type:
                SSL Client, SSL Server, S/MIME, Object Signing
    Signature Algorithm: sha1WithRSAEncryption
         0c:25:48:d5:7b:b1:2d:3c:47:f4:f7:fb:3c:14:1f:22:67:8b:
```