DESARROLLO DEL PRODUCTO PARA TEST DE PENETRACION ENFOCADO EN EL FUZZING DE APLICACIONES

Omar López [1], Ingrid Ochoa [2], Angélica Pibaque [3], Alfonso Aranda [4]
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 Vía Perimetral
Apartado 09-01-5863. Guayaquil, Ecuador
laolopez@espol.edu.ec [1], iaochoa@espol.edu.ec [2], anmapiba@espol.edu.ec [3], aaranda@telconet.net [4]

Resumen

Vivimos en plena era tecnológica, donde cada vez más información y datos de carácter personal son informatizados. La seguridad de los sistemas toma mayor importancia, sobre todo ahora que hay más amenazas y vulnerabilidades es necesario proteger uno de los activos más importantes de la organización, la información, garantizando siempre la disponibilidad, la confidencialidad e integridad de la misma.
Nuestro fin es dar a conocer que mediante la implementación y mejora de herramientas que sean capaces de categorizar la veracidad de las alertas y descartar los fallos de las aplicaciones, se podrá garantizar a la organización a que adopte las buenas prácticas sugeridas por la ISO27001:2005 para un correcto tratamiento del riesgo.
El proyecto consiste en la oferta, al mercado nacional, de un servicio de análisis de seguridades de red y aplicaciones basado en técnicas de fuzzing las mismas que consisten no sólo identificar huecos de seguridad a nivel de aplicación sino también identificar la tolerancia a fallas, capacidad de procesamiento, corrupción de memoria y errores en general. El desarrollo de este proyecto se enfocará en 3 puntos. El primero es desarrollar y ejecutar la estrategia de ventas, publicidad y marketing del Servicio (Brochure, Sitios Web, descripción técnica). El segundo punto consiste en empaquetar y customizar el Kit de herramientas que se usarán para lo cual se hará un estudio del arte de los recursos existentes, para plantear y validar mejoras, crear una base de datos con los parámetros necesarios para la respectiva documentación y reporte. Y por ultimo desarrollar la metodología que categorice la veracidad de las alertas para así descartar falsos positivos de todos los reportes que arrojen las herramientas acorde a los datos guardados en la base de datos.
Palabras Claves: *fuzzing, test de penetración, seguridad informática, fuzzers, vulnerabilidades, hackers, bugs.*

Abstract

We live in the era of technology, where more information and personal data are computerized. The system security becomes more important, especially now that there are more threats and vulnerabilities is necessary to protect one of the most important assets of the organization, information, guaranteeing the availability, confidentiality and integrity of it.
Our aim is to show that through implementation and improvement of tools that are able to categorize the veracity of alerts and failures to discard the applications, you can ensure the organization to adopt the best practices recommended by the ISO27001: 2005 for proper treatment of risk.
The project is to supply the domestic market, service analysis and network security applications based on the same fuzzing techniques that consist not only to identify security breaches at the application level but also to identify the fault tolerance capability processing and memory corruption errors in general. The development of this project will focus on 3 points. The first is to develop and execute sales strategy, advertising and marketing of the Service (Brochure, Web sites, technical description). The second point is to package and customize the toolkit to be used for which will study the art of existing resources, to raise and validate improvements, create a database with the necessary parameters for the relevant documentation and reporting . And finally develop the methodology to categorize the veracity of alerts in order to rule out false positive reports of all the tools that yield consistent with the data stored in the database.
Palabras Claves: *fuzzing, penetration testing, computer security, fuzzers, vulnerabilities, hackers, bugs*

1. Introducción

Actualmente, las tecnologías han revolucionado al mundo, mostrando nuevas y diferentes formas de manejar la información, se considera a la información como un activo de la empresa, por lo tanto es primordial proteger este activo, considerado uno de los más importantes de las empresas.

Para proteger este activo las empresas buscan nuevas soluciones que no solo vayan de la mano con el desarrollo de la tecnología sino que también los provea de forma eficiente y eficaz que su información esté disponible en todo momento.

Existen en este momento diferentes implementaciones que realizan esta función pero que por barreras tecnológicas aun se desconocen en nuestro medio ya sea por desinformación o por cuestiones económicas todavía no se han desarrollado.

Las técnicas de pruebas de vulnerabilidades son utilizadas en la actualidad en el mercado local pero aun no son completamente explotadas, es por esta razón que se decidió desarrollar el producto Test de Penetración, que no solo permite realizar pruebas de vulnerabilidades, sino que a través del Fuzzing de Aplicaciones podemos hacer un seguimiento a profundidad sobre las vulnerabilidades detectadas

Esta técnica puede estar enfocada en varios instancias desde protocolos de red, formatos de archivos, sistemas de ficheros, etc. Siendo de gran utilidad no solo para los auditores informáticos sino también para los desarrolladores de software que tendrían a esta herramienta como un apoyo.

Es aquí donde nuestro producto encuentra un mercado objetivo que con una campaña de información puede llegar a ser una solución para las empresas, ya que de esta manera protegerían su activo más importante.

2. Marco Teórico

2.1 ¿Qué es Fuzzing?

Se conoce como fuzzing a las diferentes técnicas de pruebas de software capaces de generar y enviar datos secuenciales o aleatorios a una o varias áreas o puntos de una aplicación, con el objeto de detectar defectos o vulnerabilidades existentes en el software auditado.

Es usado por compañías de software y proyectos Open Source para mejorar la calidad del software, por investigadores de seguridad para descubrir y publicar vulnerabilidades, por auditores informáticos para analizar sistemas, y, por delincuentes informáticos (hackers) para encontrar agujeros en sistemas y explotarlos de forma secreta

2.2 ¿Qué es Test de Penetración?

El Test de Penetración, también llamado a veces “hacking ético” es una evaluación activa de las medidas de seguridad de la información.

A través del Test de Penetración es posible detectar el nivel de Seguridad Interna y Externa de los Sistemas de Información de la empresa, determinando el grado de acceso que tendría un atacante con intenciones maliciosas.

2.3 Importancia del Fuzzing

El problema del software inseguro es quizás uno de los retos técnicos más importantes de nuestros tiempos. La seguridad es ahora el factor clave limitante. Cabe recalcar que no se puede tomar el control total de sobre la seguridad de aplicación sin antes realizar pruebas de seguridad en ella. Y aun así, muchas empresas de desarrollo de software no incluyen la comprobación de seguridad como parte de su proceso estándar de desarrollo debido al tiempo que les toma hacer las pruebas y el análisis.

La comprobación de seguridad, por si misma, no es una medida particularmente buena de cuan segura es una aplicación, porque existe un número infinito de modos en que un atacante podría ser capaz de colgar una aplicación, y es simplemente imposible comprobarlas todos. Sin embargo, la comprobación de seguridad tiene la cualidad única de convencer a aquellos que continuamente niegan los hechos de que existe un problema. La comprobación de seguridad ha demostrado ser un elemento clave para cualquier organización que necesita confiar en el software que produce o usa.

El fuzzing cada vez más está tomando suma importancia, logrando así que centros de investigación e investigadores independientes se enfoquen en nuevos métodos e implementaciones de fuzzing.

Cuando se realizan pruebas de seguridad, lo más importante que debe recordarse es revisar continuamente las prioridades.
Hay un número infinito de modos en que una aplicación puede fallar, y uno siempre cuenta con recursos y tiempo limitados, así que hay que asegurarse de que se los gasta sabiamente. Hay que tratar de enfocarse en los agujeros de seguridad que son más fáciles de ser descubiertos y explotados por un atacante, y que conducen a los compromisos de seguridad más serios.

2.4 Etapas del Fuzzing

El funcionamiento de los fuzzers suele componerse de las siguientes etapas:
Obtención de datos: dependiendo del tipo de fuzzing deseado y según la implementación de la herramienta, se obtendrán los datos a enviar de una lista estática almacenada en archivos, o del propio código fuente, o se generará en el momento según las configuraciones efectuadas. Este proceso sólo se puede realizar al inicio de la sesión o justo antes de cada envío.
Envío de datos: una vez que se dispone de la información que se desea enviar a la aplicación, se realizará el proceso.
Análisis: después de realizado el envío, sólo quedará esperar los resultados del fuzzing. Si no se espera ninguna respuesta por parte del objetivo, se deberá estar alerta por si se produce un comportamiento inesperado. Si, en cambio, se recibe una respuesta, entonces en este momento se comprobará si ésta indica un comportamiento normal o si, por el contrario, el ataque ha tenido éxito y la aplicación ha quedado inestable.

3. Analisis de Requerimientos y Diseño

3.1 Requerimientos Funcionales

Desde el punto de vista de los requisitos funcionales de seguridad, las normas aplicables, las políticas y reglamentos conducen ambas a la necesidad de un tipo de control de seguridad, así como el control de la funcionalidad.

3.2 Requerimientos No Funcionales

Los requerimientos no funcionales influyen en la operatividad del software. Para el desarrollo de la herramienta consideramos los siguientes:

Reproducibilidad: Un requerimiento necesario para una herramienta de fuzzing es que ésta reproduzca los resultados de ambas pruebas, las secuenciales y las individuales. Al crear este tipo de herramientas fuzzer, se debe de proveer este con una lista de números de casos de pruebas maliciosas. Se deben de generar reportes con la finalidad de informar al usuario los fallos de la aplicación

Documentación: La Documentación de los varios resultados de las pruebas son también de gran utilidad ya que es difícil reunirse con el creador del fuzzing, es más factible leer la documentación en caso de alguna duda.

Reusabilidad: Si queremos usar la herramienta fuzzer para un tipo de formato de archivo diferente, no deberíamos tener que reescribir de nuevo el código más bien se debería de crear un componente reusable en el cual podamos cambiar lo que queremos a nuestras necesidades y así minimizaríamos el tiempo para crear nuestra propia herramienta.

3.3 Diseño

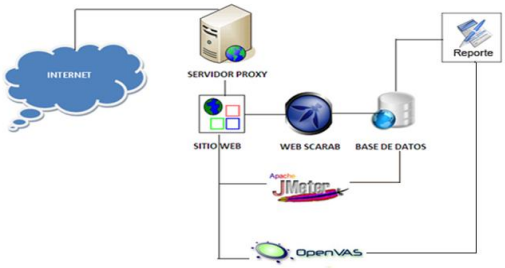

Figura 3.1 Diseño del Producto

3.3.1. Aquitectura

El Web Fuzzing se centra específicamente en los paquetes que se ajustan a la especificación de HTTP. Se registra todas las comunicaciones entre el usuario y el servidor web, esto incluye todas las

imágenes, archivos CSS, Javascript, parámetros, etc. De acuerdo al diseño y a las herramientas usadas, se podrá ofrecer un amplio servicio en el cual se podrá actuar como proxy HTTP, se interceptarán las peticiones HTTP request y HTTP response para poder estar informados de lo que hace el sitio web y observar los tipos de fallos o atentados que se podrían producir.

3.3.2. Características

Se provee de un número de plugins, cuyo objetivo principal, por el momento, es agregar funcionalidad de seguridad, reportar las vulnerabilidades y explotarlas. Estos plugins incluyen:

- Proxy - Observa el trafico entre el navegador y el servidor Web. Es capaz de observar tanto HTTP como trafico HTTPS cifrado. Permite al operador controlar las peticiones y respuestas que pasan por el proxy.
- Intercepción Manual - Permite al usuario modificar peticiones y respuestas HTTP y HTTPS antes de que ellas alcancen el servidor o el navegador.
- Simulador de ancho de banda - permite al usuario emular una red más lenta, de manera que observe como se desempeña su sitio cuando es accedido, se observa el retraso y la cantidad en milisegundos en que se demora cada usuario en acceder a dicho sitio.
- Peticiones manuales - permite editar y reenviar peticiones anteriores la creación de peticiones nuevas completas.
- Análisis de identificadores de sesión - recolecta y analiza un número de cookies (y eventualmente parámetros en el URL también) para determinar visualmente el grado de aleatoriedad y predictibilidad.
- XSS/CRLF – se buscan datos controlados por el usuario en los encabezados y cuerpo de las respuestas HTTP para identificar posibles inyecciones CRLF (partición de respuesta HTTP) y vulnerabilidades de secuencia de comandos en sitios cruzados (XSS).

3.4 Herramientas de Desarrollo

Las herramientas que han sido seleccionadas para el desarrollo del proyecto fueron elegidas por diferentes criterios como la experiencia en el manejo de la herramienta, la curva de aprendizaje de nuevas herramientas adoptadas y la facilidad de implementación.

3.4.1. Herramientas de Software

NetBeans: Es una plataforma que permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos.

WebScarab: Es una herramienta multiplataforma (Java) que sirve para realizar pruebas de aplicaciones web funcionando como proxy interceptor:

- Registra todos los accesos (documentación).
- Modificación arbitraria de peticiones y respuestas.
- Extensible a través de complementos: análisis de identificadores de sesión, pruebas automáticas de parámetros (*fuzzer*).

JMeter : Es un proyecto de Apache Jakarta que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web.
JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas. JMeter puede también ser configurado como un monitor, aunque es comúnmente considerado una solución ad-hoc respecto de soluciones avanzadas de monitoreo.

OpenVas: Es el acrónimo de *Open Vulnerability Assessment System*, un completo escanner de vulnerabilidades que

permite evaluar los riesgos de seguridad en los equipos de una red y cerrar sus vulnerabilidades.

4. PLAN DE NEGOCIOS

4.1 Plan Estratégico

Visión
El Test de Penetración se convertirá en un producto con un enfoque en el mercado ecuatoriano, necesario para la validación y prueba de software en empresas medianas y grandes, entregando la más alta calidad y un servicio único.

Misión
Ser pioneros en las herramientas de seguridad informática basadas en aplicaciones fuzzing.

4.1.2. Análisis Externo de la Empresa

Oportunidades

- Pocas empresas en desarrollo de productos enfocados en Test de Penetración.
- Empresas enfocadas en seguridad informática no tienen paquetes definidos para Test de Penetración.
- Empresas en el mercado local necesitan productos que validen sus aplicaciones.

Amenazas

- Consumidores no interesados en el producto.
- Empresas locales no entiendan el porqué del producto.
- Poca demanda del producto debido a desinformación.
- Poco conocimiento de las herramientas para los test de penetración.
- Barreras comerciales

4.2.2. Análisis Interno de la Empresa

Fortalezas

- Nos encontramos actualmente desarrollando el producto para lanzar una versión beta, las personas involucradas en esto tienen total disponibilidad.
- Nuestra visión hacia el futuro es posicionar el producto como una herramienta necesaria e indispensable para los consumidores.
- El mercado el cual vamos a llegar es nuevo por lo tanto debemos crear la necesidad a los consumidores, ser innovadores.

Debilidades

- Poco conocimiento de las herramientas para el desarrollo del software.
- Falta de experiencia en elaboración de plan de negocios.
- Falta de experiencia en marketing especializado.
- Falta de documentación en proyectos a utilizarse.

4.2. Delimitaciones del Proyecto

Se contará con un capital inicial aportado por nosotros como creadores y distribuidores del servicio, el cual servirá para los gastos de publicidad y comercialización.
La inversión inicial del proyecto comprende solo los gastos de publicidad y comercialización del servicio, de manera particular se asumirán los gastos por el empaquetamiento del software, ya que estamos usando herramientas Open Source, lo cual nos reduce el costo de inversión.

4.2.1. Competencia

En la actualidad en el mercado ecuatoriano hemos investigado a posibles competidores pero estos se encuentran en la región Sierra, en la Costa no se han encontrado competidores semejantes.
Hemos encontrado productos sustitutos, empresas que ofrecen servicios de auditoría informática, las cuales se encargan de la instalación y configuración de sistemas, mantenimiento, migración y seguridad de datos, permisos, todo esto ofrecido como un paquete de servicios.

4.2.2. Mercado Objetivo

Gobierno, instituciones públicas y privadas.

4.2.3. Factores Claves de Éxito

Los factores que se perciben como claves para el éxito en el sector son:

- Integrar los servicios que se presten dentro de la cultura y funcionamiento diario de la empresa-cliente. Estos no deben sentirse como un elemento lejano o sin importancia, sino como una parte funcional e importante de la empresa.
- Servicios complementarios. Servicio de atención de incidencias, asesorar a los trabajadores de la empresa-cliente en el uso de las nuevas tecnologías implementadas en su empresa.

4.3. Estudio de Mercado

4.3.1. Análisis de la Demanda

Nuestra demanda se basa en servicios públicos o servicios prestados con esto nos referimos a que el servicio es prestado hacia la comunidad en general, aunque no necesariamente por parte del estado, en nuestro caso nuestro sector está delimitado al área informática, actualmente conocidos como TIC (*Nuevas Tecnologías de la Información y de la Comunicación* o **IT** para «*Information Technology*»)

En el mercado local no tenemos un competidor directo, ya que los que existen son variaciones de Hacking Ético pero ninguno ofrece este servicio con herramientas fuzzing.
El mercado objetivo está en crecimiento con lo cual nos aseguramos que tendremos clientes que necesitaran nuestros servicios, lo cual será a través de un contrato, por servicios prestados con nuestros potenciales clientes.

4.3.2. Análisis de la Oferta

En el Ecuador no tenemos datos oficiales sobre uso de nuevas tecnologías ni sobre herramientas enfocadas en Seguridad Informática.

4.4. Proceso de Comercialización

El proceso de comercialización inicia desde la propuesta del servicio, la cual se la hará de manera escrita, ya que es un servicio completo.
Con el fin de prestar un servicio de calidad y personalizado, se analizará las características de los clientes (objetivos, imagen, posicionamiento, clientes, etc.) con el objetivo de lograr de una integración de los aspectos tecnológicos dentro del funcionamiento y de la imagen de la empresa.
El elemento principal del servicio será la comunicación permanente con el cliente, búsqueda de soluciones en el menor tiempo posible, buscando siempre la optimización de recursos, procurando que el cliente se sienta parte del proceso de prestación del servicio.

4.5. Penetración en el mercado

4.5.1. Canales de Distribución Red Comercial

Nos basaremos en la distribución del servicio en forma personal, se contactaran posibles clientes, con los cuales se distribuirá la información detallada del servicio, es fundamental conocer las necesidades de los clientes, para ofrecer una solución efectiva y eficaz.
Es importante que en la presentación de los servicios se acceda tanto a la persona con autoridad en la decisión final como a los trabajadores de la empresa vinculados a las nuevas tecnologías, que facilitan el proceso de toma de decisión.
Siguiendo esta referencia, para introducirnos en el mercado seguiremos las siguientes estrategias:

- Mailing: Previa selección de los clientes objetivo
- Visitas personalizadas: Previo contacto telefónico. En estas presentaciones se mostrarán trabajos desarrollados por la empresa. El éxito comercial previsto está entre el 10% y 15%.
- Acuerdos de colaboración con agencias de publicidad y marketing
- Acuerdos con asociaciones empresariales para ofrecer nuestros servicios a sus asociados en condiciones ventajosas

La forma de cobro será mediante una suscripción anual en la cual el cliente tendrá un respectivo usuario con el cual podrá acceder al servicio. Aunque también se podría tener membrecías adicionales si

el cliente así lo estima.

4.5.2. Acciones de promoción

Las acciones de promoción tendrán los siguientes objetivos.

- Presentar las ventajas de los servicios de la manera más sencilla y atractiva posible.
- Distinguir y poner en valor las diferencias respeto a los competidores y alternativas no especializadas.
- Crear prestigio e imagen de calidad.

Para apoyar estas acciones realizaremos:

- Mailing destinado a las empresas que cumplan el perfil de cliente potencial.
- Presentación on-line de nuestros servicios mediante invitaciones a conocer nuestra Web.
- Catálogos publicitarios en los que se recogerán el producto y servicio que ofrecemos.
- Presencia en directorios comerciales
- Campaña en Internet

El coste aproximado de estas acciones será de ($1442,22) el primer año, inversión que se mantendrá en sucesivos ejercicios.

4.6. Rentabilidad

VAN	147,08
TIR	85%

Podemos concluir que el valor actual netoes mayor a 0 lo cual significa que el proyecto es rentable y puede aceptarse.

Sobre la tasa interna de retorno tenemos que el valor es mayor a la tasa nominal propuesta (8%) por lo tanto el proyecto es rentable.

5. Implementación del Servicio

Las herramientas que se han sido utilizadas para este proyecto son:

- WebScarab, proyecto OpenSource, alojado en http://www.owasp.org/development/webscarab.
- Jmeter, proyecto de Apache, alojado en http://jakarta.apache.org/jmeter
- OpenVas: http://www.openvas.org/

El objetivo de esta herramienta es que pueda utilizarse de forma automática o interactiva para evaluar la seguridad de aplicaciones web.
Nuestro producto llamado, FWEB - TOOL está diseñado para pruebas de estrés y de vulnerabilidades en aplicaciones web, las cuales permiten examinar las cabeceras en las peticiones enviadas y recibidas, se lo puede operar como un proxy de intercepción, que permite al operador revisar y modificar las peticiones creadas por el navegador antes de que sean enviados al servidor, y para revisar y modificar respuestas enviadas por el servidor antes de que sean recibidas por el navegador de tal manera que simula todas las funcionalidades de un Navegador ("Browser"), siendo capaz de manipular resultados en determinada requisición y reutilizarlos para ser empleados en una nueva secuencia.

Además, posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de requisiciones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción.
FWeb - Tool proporciona un número de plugins los cuales son destinados para generar peticiones (basado en la interacción del usuario, y basados en la información obtenida mediante el análisis de páginas recuperadas por otros plugins).
Entre las características del FWeb – Tool tenemos:
Funcionalidades que provee WebScarab
El plugin de Manual Request se encarga de realizar un análisis pasivo buscando datos controlados por el usuario en los encabezados y cuerpo de las respuestas HTTP para identificar posibles.

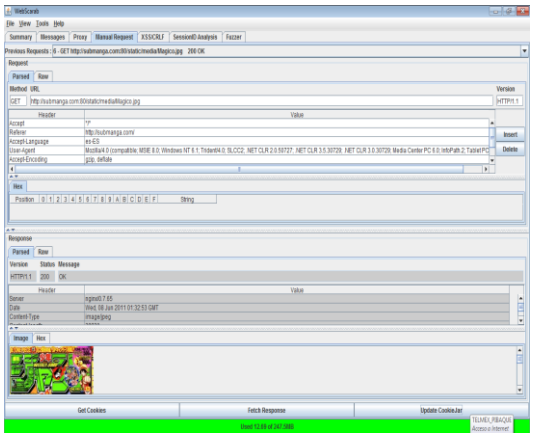

Figura 5.1 Plugin Manual Request

El plugin de XSS/CRLF se encarga de Inyecciones CRLF (partición de respuesta HTTP) y vulnerabilidades de secuencia de comandos en sitios cruzados (XSS).
El plugin de SessionID Analysis que recolecta y analiza un número de cookies para poder determinar mediante un grafico el grado de aleatoriedad y predictibilidad.
El plugin de Fuzzer nos permite un análisis con parámetros definidos previamente, ya sea desde un archivo plano o una variable ingresada.

Funcionalidades que provee Jmeter

Los plugins de PeticionHTTP y Resultados en Árbol permiten al usuario emular una red más lenta, simulando un gran tráfico de usuarios que acceden a dicha pagina y así se puede observar cómo se desempeña el sitio cuando es accedido.

Funcionalidades que provee OpenVAS

OpenVAS es un sistema de análisis de vulnerabilidades y un scanner de seguridad de red el cual nos provee pruebas generales de vulnerabilidades, dándonos un panorama general, ingresando solo la dirección a analizar, tiene una interfaz grafica a modo de cliente, los resultados son analizados en el servidor de OpenVAS.

5.1. Pruebas

Mediante las pruebas se intentará reunir la mayor cantidad de información posible sobre el sitio web seleccionado.
Para dar fe del buen rendimiento del servicio web, éste ha sido sometido a pruebas de stress. Se van a realizar peticiones HTTP emulando a usuarios realizando búsquedas de productos.
Su funcionamiento es muy sencillo. Primero hay que indicarle los siguientes parámetros:

- Número de hilos: Equivale al número de usuarios que se desean emular.
- Periodo de subida: Es el lapso de tiempo en segundos que se desea tener entre cada grupo de usuarios.
- Contador del bucle: Utilizado para indicar el número de veces que se va a llevar a cabo la emulación.

Y una vez definidas los parámetros anteriores, hay que indicarle la dirección del servidor al cual van a ir dirigidas las peticiones HTTP.
Para un mejor resultado ha decidido realizar varias pruebas de stress en donde la variable que diferencia es el tiempo entre ciclos que los usuarios tienen para realizar su petición. Así se podrán llegar a saber los límites que puede tener el servidor.
Se le agrega peticiones HTTP para poder ver lo que se recibe y se envía como información, es conveniente agregar Manejadores de Cabeceras para poder grabar valores de las sesiones en caso de que las haya.
Con ayuda de los diferentes tipos de gráficos, se puede hacer un mejor conteo de las muestras sacando un promedio, porcentaje de error, máximo y mínimo. Para así poder hacer un análisis más detallado al momento de mostrar el reporte al usuario.

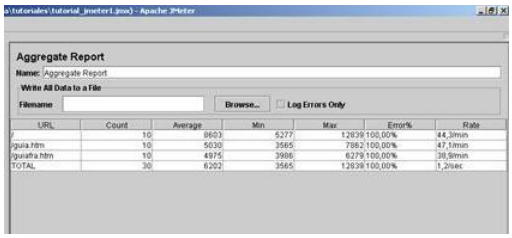

Figura 5.2 Ejemplo Peticiones

5.2. Análisis

El análisis se lo efectúa mediante los resultados de las muestras, se tabula la información y se obtiene un grafico de muestras vs tiempo, el promedio y la desviación estándar. Mediante el uso de todos los escáneres de vulnerabilidades descubiertas se determina si hay cualquier otra vulnerabilidad que podría ser

explotada para obtener acceso a un host de destino en una red.

CONCLUSIONES

El interés por diferentes tipos de ataques se ha incrementando con rapidez. Muchas empresas están buscando aumentar la eficiencia de sus operaciones y reducir las vulnerabilidades. La oferta de este tipo de soluciones cada vez es mayor, el fuzzing, es una alternativa muy eficaz a la hora de encontrar vulnerabilidades en aplicaciones, sobre todo nuevas vulnerabilidades, pero es preciso tener en cuenta que no existe ninguna herramienta que pueda garantizar la seguridad de un software al 100 %, por lo que el uso de este tipo de soluciones debe ser una parte más de la auditoría de un sistema.
El poder del FWEB – TOOL radica en la capacidad de leer peticiones, las cuales pueden ser clasificadas para un control más exhaustivo, identificar las vulnerabilidades y simular una red traficada por muchos usuarios y así calificar el rendimiento de los sitios analizados.

RECOMENDACIONES

Debido a las consecuencias que puede ocasionar el tener un sitio web vulnerable, se recomienda tomar medidas de seguridad exigibles a los ficheros y tratamientos de datos informativos. El hecho de que una aplicación deje de responder puede ser el indicio de que existe alguna forma de controlar el flujo de instrucciones. Si es así, el problema es muy grave y es necesario hacer uso de las respectivas herramientas para controlar la situación.

Al ejecutar cualquier herramienta, nos debemos de concentrar en remover las vulnerabilidades detectadas más críticas. Inicialmente será imposible atacar todas las disconformidades en forma simultánea (excepto que se empiece de cero siguiendo las mejores prácticas de testeo).

BIBLIOGRAFIA

[1] Michael Sutton, Adam Greene, Pedram Amini, Bruce Force Vulnerability Discovery, Pearson Eduacation, 2008.

[2] Owasp, Guia de Pruebas OWASP Version 3, Fundacion OWASP, 2008

[3] Jose Miguel Esparza Muñoz, Security Research S21sec labs, 2008.

[4] Verizon Business, 15 ataques de seguridad más comunes en las empresas, http://mgluaces.wordpress.com/2009/12/30/los-15-ataques-de-seguridad-mas-comunes-en-empresas/, fecha de consulta agosto 2010.

[5]Owasp, WebScarab, https://www.owasp.org/index.php/Proyecto_WebScarab_OWASP , fecha de consulta agosto 2010.

[6] Osmosis Latina, Jmeter, http://www.osmosislatina.com/jmeter/basico.htm, fecha de consulta enero 2011.

[7] Apache Jakarta Project, Jmeter, http://jakarta.apache.org/jmeter/, fecha de consulta enero 2011.

[8] OpenVas, OpenVas, http://www.openvas.org/, fecha de consulta enero 2011.

[9]SRI, Instructivo Depreciaciones, http://descargas.sri.gov.ec/download/pdf/instructivo_101.pdf, fecha de consulta marzo 2011.