MISTIC

IMPLEMENTACIÓN DE UN ENTORNO PARA LA CAPTURA DE ATAQUES Y MALWARE EN RED

PABLO CASTAÑO DELGADO

DIRECTOR: JORDI SERRA RUIZ

UNIVERSITAT OBERTA DE CATALUNYA

23-JUNIO-2014

A mis padres, por su apoyo constante y ánimos para seguir adelante día tras día.

A mí hermano, por sacar de mí una sonrisa y hacerme feliz en los momentos más duros.

Y a Sofía, por enseñarme que siempre hay cosas increíbles por vivir y lecciones valiosas que aprender a la vuelta de cada esquina.

Gracias a todos por mantenerme feliz y motivado durante la realización de este proyecto.

La intención principal de este proyecto es proporcionar una visión global del software encargado de desplegar honeypots y de todas las herramientas que dan soporte a sus operaciones. Se utilizará un enfoque a nivel práctico y teórico que permita adquirir una visión global a través de un entendimiento de las características de este software y de las funcionalidades principales, así como las ventajas de su utilización.

Si bien es imposible cubrir en un solo proyecto y en el periodo de tiempo designado los aspectos más avanzados de las honeypots, como los despliegues de honeynets y la utilización de honeypots de alta interacción, al menos se sentarán las bases para un trabajo futuro que permita seguir adquiriendo competencias en la materia.

En primera instancia este documento permite determinar qué modelo de honeypot de entre todas las disponibles es el más adecuado para las necesidades de una organización en base a la información que suministra, pero también en relación a las habilidades del personal de seguridad.

Además, ofrece una visión clara de cómo configurar distintas herramientas que emularán diversos sistemas. La elección de dichos sistemas se ha llevado a cabo atendiendo a las tendencias de ataque actuales y a los avances en tecnologías de la información. Empezando así con honeypots de sistemas convencionales para pasar a otras algo más complejas que emulen sistemas críticos como SCADA o aplicaciones web.

Es por esto que este trabajo no solo aportará información sobre los ataques en sí, sino además sobre las preferencias de los atacantes por los diversos sistemas que en la actualidad se encuentran accesibles desde Internet.

The main purpose of this project is to provide a holistic view on the software responsible for honeypot deploying, along with all the tools that support its operations. It will be used both a theoretical and practical approach, in order to be able to acquire a global view through an understanding of the features of this software and its main capabilities, as well as the advantages associated to its use.

Even if it is impossible to cover the most complex aspects about honeypots in a single project in the designated period for the task, as are the deployment of honeynets and the usage of high interaction honeypots, at least it will establish the base for further work that will allow the acquisition of new skills in this topic.

At first, this document allows to determine the type of honeypot among all of them that better fits for the organization needs according to the provided information, but also regarding the security staff's skills.

Furthermore, this document offers a clear view on how to configure different tools that will emulate several systems. The choice about these systems responds to the current trends on cyber attacks and also to the IT firsts. Thus starting with conventional systems' honeypots and continuing with more complex ones that will emulate critical systems such as SCADA or web applications.

That is why this work will not only provide information about attacks themselves, but also about the attackers' preferences on the different systems that can be found accessible on the Internet nowadays.

ÍNDICE

Capítulo I: INTRODUCCIÓN

Las honeypots se han convertido en una herramienta de gran potencial para los responsables de seguridad de las TI. Eso es así debido a que por primera vez, con la introducción de las honeypots, se puede actuar de forma ofensiva con respecto a un ataque. Identificando las fuentes, analizando la metodología y respondiendo de forma eficaz a una amenaza concreta.

Por otro lado es contrario al resto de medidas de seguridad, pues se despliegan con el único fin de ser atacadas y su seguridad comprometida. Y aun así, son uno de los mecanismos que más información pueden aportar sobre un atacante.

Según Lance Spitzner podemos diferenciar dos tipos de ataques a los sistemas informáticos. El primer tipo (ataques de oportunidad), mucho más común, suele llevarse a cabo utilizando un alto grado de automatización, haciendo uso en su mayoría de scripts y con la única intención de comprometer cuantos más sistemas mejor, siempre que cumplan con unas condiciones deseadas (un gran ancho de banda o capacidad de disco duro suficiente para almacenar malware sin levantar sospechas). Por otro lado, y aquí entran en juego los ataques que suelen sufrir las grandes organizaciones y para los que es muy difícil estar preparado, son aquellos que buscan comprometer un sistema en concreto; buscando las vulnerabilidades necesarias para penetrar en ellos. Estos últimos (ataques dirigidos) son muy peligrosos por el nivel de habilidad que necesita el atacante para penetrar en el sistema. Precisamente por este motivo, los encargados de proteger este sistema deben adquirir una gran habilidad previniéndolos, llegando a conocer a sus atacantes lo mejor posible, teniendo en cuenta además que suelen ser muy cuidadosos con el rastro que dejan en los sistemas. Puede parecer que los ataques dirigidos son más peligrosos, lo que no hay que olvidar es que en el primer caso el número de equipos comprometidos es mucho mayor. Además, hoy en día las técnicas de ingeniería social hacen aún más fácil conseguir un ataque masivo satisfactorio en un gran porcentaje (casos de phishing, gusanos a través de redes sociales, ...). De hecho, en los últimos años, dos de los ataques más importantes Operación Aurora y Stuxnet se han valido de este tipo de ataques para conseguir sus objetivos, combinando en el segundo caso una infección masiva que ampliara la posibilidad de encontrar un sistema objetivo para posteriormente realizar un ataque dirigido una vez encontrada una máquina candidata válida. Muchas de estas infecciones se llevaban a cabo utilizando técnicas de ingeniería social, aprovechando el desconocimiento y la confianza de los usuarios para expandir el gusano de forma rápida y eficiente a través de memorias flash USB.

Riesgo de los equipos en base al tipo de atacante.

Con el paso del tiempo el conocimiento técnico requerido un atacante ha ido disminuyendo. Mientras que en un primer momento era necesario un extenso conocimiento sobre lenguajes de programación, protocolos de red, arquitectura de sistemas... Hoy día hay disponible una gran variedad de herramientas que automatizan estas tareas para hacer el proceso de explotación más sencillo, siendo necesario únicamente introducir una serie de comandos de configuración y lanzar el exploit o en otros casos un simple clic de ratón en un botón una vez seleccionadas las opciones deseadas.

El trabajo de los cibercriminales se ha diversificado, dedicando un gran número de personal a realizar ataques de forma que se buscan profesionales altamente especializados en un campo concreto que se ocuparán

de una tarea relacionada. A día de hoy podemos ver toda una industria en los ataques y la creación de malware. De hecho esto hace que a nivel legislativo sea difícil asignar responsabilidades, puesto que la persona que hace el payload, el encoder o el cpanel ni siquiera forman parte del mismo equipo de desarrollo.

Con esta sofisticación en los ataques y en la preparación para los mismos también se encontró un grave inconveniente asociado para los administradores de sistemas y responsables de seguridad, puesto que a medida que crecía la sofisticación también crecía el hermetismo en cuanto al funcionamiento interno y el diseño del malware. Los 0-days se han vuelto una herramienta muy poderosa, por el desconocimiento de la industria de la seguridad informática sobre su existencia y porque por lo general tienen un mayor grado de efectividad (al no haberse desarrollado ningún parche erradicarlos).

Enviar	**Robar**	**DoS** (denegación de servicio)	**Fraude mediante clics**
Envían - spam - virus - software espía	Roban información privada y personal y se la comunican al usuario malicioso: - números de tarjeta de crédito - credenciales bancarias - otra información personal y confidencial	Lanzan ataques de denegación de servicio (DoS) contra un objetivo específico. Los criminales cibernéticos extorsionan a los propietarios de los sitios web por dinero, a cambio de devolverles el control de los sitios afectados. Sin embargo, los sistemas de los usuarios diarios son el objetivo más frecuente de estos ataques, que sólo buscan molestar.	Los estafadores utilizan bots para aumentar la facturación de la publicidad web al hacer clic en la publicidad de Internet de manera automática.

Figura 1. Causas comunes de ataque.

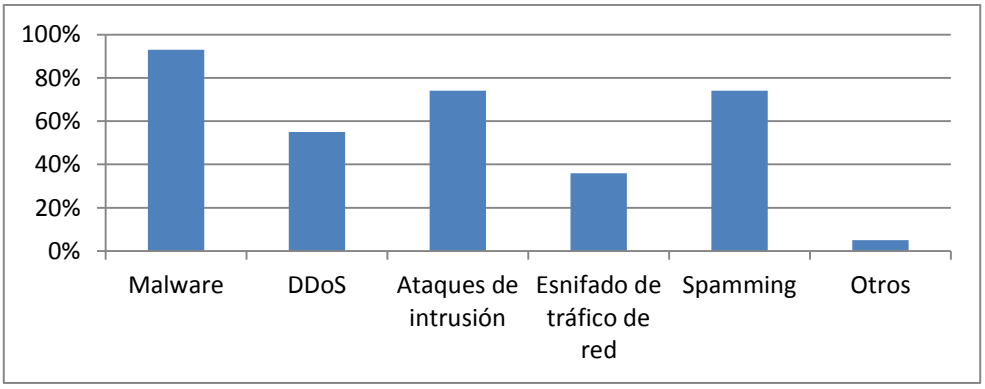


Figura 2. Ataques más detectados por Honeypots.

Historia reciente en el campo de las Honeypots

El inicio en el establecimiento formal e implementación de las Honeypots se remonta a la década de los 90 y es descrito en la obra de Lance Spitzner “Honeypots: Tracking Hackers” por lo que avanzaré hasta el inicio del nuevo milenio, una época mucho menos estudiada en cuanto al desarrollo de honeypots.

Con la llegada del año 2000, el crecimiento de gusanos para sistemas tanto Unix como Windows, incluyendo los derivados del primero. Uno de los hechos más fascinantes sobre este tipo de malware es su increíble capacidad de propagarse a través de Internet y su gran efectividad. La tarea de analizar estos gusanos se volvió complicada, puesto que estaban especialmente programados para ser difíciles de rastrear. CodeRed

por ejemplo, solo residía en memoria, mientras que otros contenían gran cantidad de información basura en su código. Sin embargo todo cambia con la implementación de honeypots.

Destacar también el trabajo de las honeypots contra el troyano Sub7, que comprometía los sistemas Windows y concedía control total gracias a software específico que mandaba peticiones al puerto 27374 de la víctima. Johannes Ullrich del SANS Institute desplegó el 21 de Junio de 2001 una honeypot que emulaba un sistema infectado por Sub7. Su descubrimiento se produjo minutos después, cuando un gusano se estaba haciendo pasar por un cliente Sub7 y pretendía infectar el equipo que ya había sido comprometido. Este estudio permitió identificar el gusano en cuestión y analizarlo en profundidad.

Hasta la fecha solo se habían desplegado honeypots a la espera de capturar ataques o malware conocido que estaba circulando por la red. No obstante, el 8 de enero de 2002, una honeypot emulando un sistema Solaris captó un nuevo exploit dtspcd del que no se tenía conocimiento hasta la fecha. Aunque la vulnerabilidad ya había sido notificada para sistemas Unix el 12 de noviembre del año anterior y se sabía que se podía ganar acceso al sistema, no se conocía ningún exploit que pudiera hacer uso de dicha vulnerabilidad hasta entonces. Una vez más, el valor de las honeypots quedaba probado.

En los últimos años el uso de honeypot ha ido cobrando cada vez mayor relevancia. Los estudios en relación a las mismas abalan constantemente su utilidad y su necesidad. No obstante, hay muy poca concienciación e información en Internet sobre honeypots. Por un lado tenemos herramientas comerciales que en su mayoría llevan mucho tiempo sin actualizarse. Algunas de ellas además hacen su uso a nivel personal inviable debido a los precios prohibitivos, no solo por adquirir el software, sino por su cuota. En este aspecto destaca Specter, una de las herramientas comerciales más completas que actualmente se encuentra en su versión 0.8 (el SO Windows más moderno emulado es Windows XP, un SO que ya ni siquiera es mantenido por Microsft) con un coste de 899 US$ para la versión inicial. Otras herramientas como KFSensor ofrecen una versión de prueba de 30 días lo cual permite comprobar que la herramienta se ajusta a las necesidades del cliente.

Por otro lado tenemos diversos artículos sobre profesionales relativamente recientes que aportan mucha información sobre honeypots y sobre posibles implementaciones, pero que en ningún caso pueden ser tomados como algo más allá de un análisis abstracto.

Como alternativa tenemos a The Honeynet Project, diferentes grupos de trabajo repartidos por todo el mundo que desarrollan proyectos Open Source. Los proyectos son muy variados aunque todos versan sobre las honeypots o sobre el análisis de malware. Entre los proyectos más importantes destacan HoneyD, Dionaea, Kippo, Honeywall o HIHAT. Esta comunidad es la alternativa más fiable a la hora de adquirir software para el despliegue de honeypots, puesto que la mayoría de sus herramientas se mantienen actualizadas y la documentación complementaría suele ser abundante. De hecho, a lo largo del proyecto la mayoría de herramientas utilizadas forman parte de alguno de estos proyectos.

JUSTIFICACIÓN

La justificación principal en el despliegue de una Honeypot es simple: los atacantes. La gran mayoría de equipos accesibles desde Internet, sean sus usuarios conscientes o no, sufren ataques, o al menos son escaneados en busca de vulnerabilidades. Esto no es algo que vaya a detenerse, más aún, está sufriendo un incremento día a día. Nuevos tipos de ataques surgen casi al mismo tiempo que una nueva tecnología o servicio

es puesto a disposición del público. Es imprescindible saber las tendencias dentro de la industria del malware y de los ataques informáticos a través de Internet.

El estudio de los ataques a la seguridad de la información en una organización se ha vuelto una tarea cada vez más completa con la sofisticación de los mismos y la industria de producción de malware, dificultando las labores de estudio de los ataques y las herramientas utilizadas.

Una escalada en la sofisticación de los ataques y el malware requiere por tanto una sofisticación en la forma de estudiar los mismos. Para ello es fundamental la utilización de honeypots, puesto que aportan una visión más completa y detallada de la metodología de ataque.

Las iniciativas en el campo de este tipo de software se limitan en la gran mayoría de los casos a una visión académica y abstracta que se lleva a cabo por distintos expertos en seguridad, pero no es tanta la información relativa a su implementación y funcionamiento. De hecho, la mayoría de enlaces en Internet en páginas de carácter divulgativo y que referencian a soluciones reales se encuentran caídos, lo cual evidencia una falta de actualización en este tipo de herramientas (Bigeye, BackOfficer Friendly, Bait-n-Switch, …).

Por todo lo anterior, este documento pretende aportar una visión nueva, con ejemplos de implementación de diversas herramientas que nos permitirán desplegar honeypots en la red, análisis de las metodologías de ataque y de los pasos que se llevan a cabo durante el proceso.

CONTEXTO

The Honeynet Project

En el mundo de las honeypot es el Honeynet Project, una organización dedicada a la investigación a nivel internacional sobre seguridad. Se dedica a investigar las tendencias en los ataques y a desarrollar herramientas de seguridad.

Es un proyecto que se inició en 1999 como una lista de correo muy modesta. Posteriormente, en 2000, se forma el proyecto Honeynet como tal. Su organización actual se basa en la creación de capítulos que se localizan en un país concreto y que desarrolla productos específicos. Esto le permite al proyecto obtener información a nivel mundial y de forma distribuida.

Desde el año 2009 el Honeynet Project colabora con el Google's Summer of Code (GSoC), dirigido a estudiantes y con la posibilidad de que participen en el desarrollo de soluciones Open Source.

Investigaciones sobre Honeypots.

Fuera de The Honeynet Project hay pocos documentos que traten aspectos relacionados con Honeypots con un nivel de detalle suficiente como para resultar útiles:

"Research in Honeypot Technologies" – Karl Schuttler: describe un escenario de trabajo con dos máquinas que actuarán como servidores, detalla los recursos de esas máquinas y se identifica a él mismo como único participante, sin aportar más datos en su documento público.

"Honeypot reveals mass surveillance of BitTorrent downloaders" – Lisa Vaas 2012: en dicho documento se analiza la forma en la que ciertas organizaciones monitorizan la actividad procedente de IPs previamente identificadas como usuarias de redes de torrent para la descarga de material sujeto a leyes de copyright. Aunque el artículo es relativamente reciente y curioso, lo cierto es que no aporta detalles técnicos sobre el proceso.

"A Virtual Honeypot Framework" – Niels Provos 2004: Este documento entra en detalles técnicos sobre las consideraciones a tener en cuenta para implementar mecanismos de detección. Las dos razones principales a la hora de considerar este trabajo como no apropiado son que por un lado no establece una solución formal en forma de honeypot y en segundo lugar hace 10 años de su publicación y por tanto podríamos considerarlo desactualizado, a pesar de que los planteamientos básicos de la idea son perfectamente válidos.

"Honeytokens: The Other Honeypot" – Lance Spitzner 2003: Este documento, del creador del documento clave en el desarrollo de las honeypots clásicas "Honeypots, tracking hackers", trata aspectos formales en relación a las honeypots. Ayuda enormemente a la aclaración de conceptos pero no aporta información técnica o detallada. En concreto detalla el funcionamiento de los Honey tokens, una honeypot que no es un elemento físico, sino que se despliega como una entidad digital. Una vez más se defiende su utilización como método de estudio de las amenazas internas.

"Lessons Learned from the deployment of a high-interaction honeypot" – Eric Alata, Vincent Nicomette, Mohamed Kaaniche, Marc Dacier, Matthieu Herrb 2007: Este documento está especialmente enfocado en los resultados de la honeypot desplegada. Quizás sea un documento interesante puesto que analiza los resultados de una honeypot de alta interacción. La importancia de este documento radica en que dichas honeypots requieren unos conocimientos más extensos y una configuración más precisa para no ponerlas en peligro. Se pueden distinguir los distintos tipos de ataques y algunos de los pasos llevados a cabo, como los usuarios y contraseñas probados.

"Honeypot-based Forensics" – F. Pouget, M. Dacier: aquí se discuten los usos posibles de honeypots, desde la ralentización en las tareas de los atacantes, la reducción del spam a la mitigación de ataques. Este documento analiza todas estas funciones en una perspectiva temporal más amplia de lo normal. Podemos observar un estudio profundo sobre las características de los ataques recibidos, con un componente técnico muy superior a otras publicaciones. Aunque no es un documento especialmente extenso sí que se caracteriza por ser muy completo en su ámbito de estudio.

"The Social Honeypot Project: Protecting Online Communities from Spammers" – Kyumin Lee, James Caverlee, Seteve Webb 2010: supone un punto de vista distinto al resto de estudios sobre honeypots. En este caso se trata de desplegar perfiles en las distintas redes sociales de forma que se forme una muestra representativa teniendo en cuenta los porcentajes de utilización de cada red social distribuidos por edad y género. Una vez desplegados todos los perfiles se empiezan a analizar los mensajes enviados a los mismos. Para que el estudio pueda ser representativo se ha incluido información básica, con imágenes obtenidas de una base de datos gratuita. En el estudio se hallan evidencias robustas sobre la utilidad de las honeypots sociales para captar ataques de spam.

"Honeypot back-propagation for mitigating spoofing distributed Denial-of-Service attacks" – Sherif Khattab, Rami Melhem, Daniel Mossé, Taieb Znati 2006: se detalla uno de los esquemas propuestos para reducir el impacto de los DoS y rastrear los paquetes hasta su origen. El documento propone un nuevo mecanismo de

rastreo hacia atrás con el uso de honyepots. La recepción de un ataque dispararía la iniciación en cascada de diversas honeypots que permitirían el análisis de un volumen de datos en aumento.

"Experiences With Honeypot Systems: Development, Deployment, and Analysis" – Autores Desconocidos: se trata de una recopilación de investigaciones sobre técnicas de honeypots y recopila los resultados de las mismas. Se estudian honeypots de alta y baja interacción desplegadas en una red universitaria, y los datos obtenidos son analizados.

Tendencias en Honeypots.

Según "A Survey: Recent Advances and Future Trends in Honeypot Research" del 2012 por Matthew L. Bringer, Christoper A. Chelmecki y Hiroshi Fujinoki las principales tendencias en honeypots y su uso son:

Novedades en los tipos

- Adachi – "BitSaucer": una honeypot híbrida con mezcla de alta y baja interacción que permite funcionamiento con requerimientos de recursos típicos de una honeypot de baja interacción y la posibilidad de emular respuestas completas de las honeypots de alta interacción. Se ejecuta un proxy en cada host, como un demonio capaz de generar host virtuales y redirigir tráfico de red. Cada host virtual emula un sistema completa con alta interacción. Al ejecutar los host virtuales bajo demanda reduce considerablemente en nivel de recursos requeridos.
- Alberdi – "Redirection Kit": Esta solución puede monitorizar actividades maliciosas producidas por bots, gusanos y virus impidiendo que dicha actividad salga de la honeypot. Este kit redirige los taaques salientes, como los mensajes de coordinación de bots, hacia otras honeypots. De esta manera se previenen ataques a otros servidores en producción. Con este mecanismo el atacante que utilice la botnet creerá que se está comunicando con equipos fuera de la red.
- Aloefer – "Honeyware": es una honeypot de baja interacción del lado del cliente que detecta servidores web maliciosos. Tiene un alto porcentaje de detección (más del 95%). El problema es que al ser de baja interacción debe realizarse un post-procesamiento aparte. El siguiente paso en esta honeypot es combinar enfoques de alta y baja interacción.
- Anagnostakis – "Shadow honeypot": como un nuevo diseño para mejorar los problemas a la hora de elegir entre precisión de las HI honeypot y la amplitud de la cobertura en la detección de anomalías. Esta herramienta soluciona este problema con un conjunto de aplicaciones de red reales que contienen código de honeypot incrustado en ellas. El tráfico siempre se procesa por el servidor real, pero al mismo tiempo el código de honeypot monitoriza toda la actividad, siendo procesada antes de su tratamiento por un detector de anomalías, que es configurad intencionadamente con un alto porcentaje de falsos positivos, y es la shadow honeypot la que se encargará de redireccionar la petición al servidor en caso de tratarse de un falso positivo.
- Bailey: integra las ventajas de ambos tipos de honeypot (HI y LI) con la amplitud de cobertura (representada con los diferentes tipos de tráfico de red cubiertos) y la fidelidad del comportamiento (el detalle con el que la información es registrada por la honeypot). Se propone el uso de múltiples honeypots de baja interacción como sensores para recopilar información sobre el tráfico de red, si dichos sensores detectan algo inusual que pudiera asemejarse a una amenaza entonces desvían las sesiones sospechosas a honeypots de alta interacción.

- Das – "Active Server (AS)": plantea una solución para mitigar los ataques de tipo DoS ocultando los servidores en producción detrás de un Gateway de acceso (AS). Cada AS autentica a sus clientes y solo entonces se le abre una ruta hasta el servidor. Si el cliente no se autentica entonces el Active Server actúa como una honeypot. Aunque un atacante acceda a diversos AS debe identificarse en todos ellos para que se establezca el path hasta el servidor, mitigando así los ataques DoS, ya que los clientes que solo tengan un AS asignado seguirán enviando tráfico al servidor con normalidad.
- Ghourabi: propone una honeypot del lado del cliente que proteja los routers como respuesta al incremento de ataques y exploits que utilizan protocolos de capa 3 (RIP, OSPF y BGP). Este tipo de honeypot permitiría una detección temprana de los nuevos ataques contra routers. Se obtiene por tanto una honeypot de alta interacción utilizando herramientas disponibles (como Quagga) para enviar mensajes de forma activa a los routers remotos que utilicen un protocolo de enrutamiento concreto, para examinar si han sido comprometidos. Las respuestas se capturan y procesan gracias a Wireshark.
- Jiang: investiga las debilidades de las honeypots de alta interacción implementadas en MVs. El software que permite la virtualización es complejo, y por tanto es susceptible a exploits, bugs y agujeros de seguridad. Por lo tanto, se propone el uso de dos sensores para monitorizar las honeypots de alta interacción. Primero se sitúa un sensor interno dentro de la honeypot que registra las llamadas al sistema producidas y sus respuestas. El sensor externo cubre la debilidad del sensor interno, ya que éste puede ser comprometido por un atacante. El sensor externo por tanto captura los datos de entrada y salida a la honeypot para monitorizar la actividad del primer sensor.
- Khattab: se trata de otra solución que mitiga el impacto de los ataques DoS. La solución propuesta se basa en intercambiar los servidores en producción y las honeypot a lo largo de la red. Los procesos en producción se vuelven automáticamente honeypots cuando se migran a otro host. Esta solución es efectiva cuando la mayoría del tráfico entrante forma parte de ataques DoS. Esta implementación tiene una debilidad, y es que no es útil cuando el volumen de ataques DoS se reduce.
- Kreibichi: propone una honeypot que automáticamente generaría firmas de detección de intrusiones sin demasiada intervención y con la posibilidad de detectar 0-days. El núcleo del prototipo consisten en el registro de conexiones y un nuevo algoritmo de creación de firmas, que se implementan en módulos acoplados al código de una honeypot de baja interacción existente. El registro de conexiones almacena tanto las que se encuentran en fase de establecimiento de conexión como las que ya se encuentran activas. El algoritmo de creación de firmas trabaja a nivel de protocolo y de aplicación para cada conexión sospechosa.
- Lauinger: advierte de la utilización de honeypots por parte de los atacantes para realizar ataques de phishing en aplicaciones de mensajería instantánea (IM). La honeypot invita a dos usuarios legítimos de forma asíncrona a una sesión de IM. La honeypot realiza un ataque de MITM, monitorizando todos los mensajes enviados y modificándolos tras capturarlos.
- Li: utiliza honeypots para implementar funcionalidades de captura de spam en sistemas de banca online. La honeypot propuesta elimina el factor humano en el análisis de los sitios desde los que se lanzan los ataques de phishing utilizando cuentas falsas que monitorizan transacciones bancarias.
- Nazario – "PhoneyC": propone una ampliación de las honeypots tradicionales. En primer lugar las vuelve activas, lo cual supone un despliegue en el lado del cliente. Por otro lado utiliza un analizador dinámico de contenido web que es capaz de interpretar scripts en Javascript, VB Script y Active-X. Este analizador asiste a los administradores en el análisis de los binarios para descubrir servidores web maliciosos. Esta honeypot adquiere el comportamiento de un web crawler que recorre un gran número de servidores web para detectar automáticamente aquellos de tipo malicioso, gracias a la detección de diversos script y actividades de control maliciosas.

Utilizando las salidas de las Honeypots

- Chen – "dynamic forensics": la solución se asegura de que la información recopilada por la honeypot es confiable aunque el malware haya tratado de modificarla. Un sistema forense se activa dinámicamente si el nivel de amenaza detectado se eleva por encima de un punto. Una vez que se activa el sistema forense el tráfico se redirige a una honeypot, llegando a cortar todo el tráfico si se cree que el ataca está progresando más allá de los límites permitidos.
- Cooke: estudia las posibles amenazas que sufren las honeypots por culpa de los bots. La investigación de Cooke demuestra que la sofisticación creciente en los bots supone una amenaza para los honeypots. Cooke experimenta instalando una honeypot para observar si es comprometida repetidamente de ataques procedentes de bots. Los hallazgos demuestran que los bots son cada vez más robustos y las honeypots deben ser protegidas en consecuencia. Motivado por este conexto Cooke propone la creación de honyepots para honeypots, que monitoricen las infecciones intencionadas de honeypots de primer nivel.
- Dantu: propone una nueva arquitectura, que detecte gusanos monitorizando el ratio de sus conexiones salientes. La nueva arquitectura ralentiza los gusanos limitando sus conexiones externas gracias a un control de bucle de retroalimentación cerrado. Se aplican algoritmos proporcionales, integrales y derivativos para reducir la difusión del gusano. Esta medida ralentiza la expansión del gusano y lo hace cinco veces más lento.
- Fairbanks: propone un método para superar la limitación en la recolección de información dentro del sistema de archivos virtual de Linux (VFS) y evitar las posibles modificaciones de ficheros. El problema viene a la hora de identificar el nombre de los archivos a partir de su i-nodo, la solución propuesta se basa en utilizar un enlace bidireccional, desde el nombre del archivo al i-nodo y viceversa gracias a estructuras temporales).
- François: aplica gráficos de intersección para detectar acciones coordinadas en ataques distribuidos. Los gráficos de intersección se usan para identificar los nodos responsables (nodos centrales) en la coordinación oculta de ataques distribuidos mediante la identificación de patrones de comunicaciones en nodos aparentemente no relacionados. La investigación determina que las honeypots distribuidas son mejores, frente a los telescopios de red, a la hora de reconocer las direcciones de origen en los ataques coordinados ocultos, mientras que los telescopios de red son mejores detectando servicios de red objetivo en aquellos casos en los que los ataques no se producen frecuentemente.
- Hoepers: diseñó e implementó un protocolo estándar de intercambio de información de salida proveniente de honeypots heterogéneas. El protocolo promueve la coordinación de actividades en dichas honeypots. Se propuso su integración en honeypots distribuidas para una coordinación posterior que permita una detección mejor y más rápida de amenazas de seguridad en red. Las propiedades que el protocolo satisface son: interoperabilidad, sistemas abiertos, sintaxis que describa la semántica, codificación orientada a objetos y extensión futura de análisis de información.
- Krasser: incide en la importancia de la visualización de la información recogida por las honeypots para ayudar a solucionar los dos problemas principales en el análisis de información. En primer lugar es que la labor de análisis requiere un esfuerzo extenso y duro para encontrar anomalías en grandes volúmenes de información. El segundo problema hace referencia a las grandes aptitudes que deben poseerse para realizar la investigación de la información, así como experiencia y entrenamiento. Entre las variables a tener en cuenta durante el análisis se encuentran las líneas temporales, el tamaño de los paquetes, la distribución de las direcciones IP origen, los tipos de protocolos, la duración de cada conexión y los puertos locales accedidos.
- Mohammed – "Double Honeynet and Principal Component Analysis (PCA)": con su sistema pretende aumentar la precisión en la generación de firmas para gusanos polimórficos. El sistema propuesta captura el

gusano en una primera honeynet, tras lo cual se le permite infectar otros sistemas en la segunda honeynet. El gusano puede moverse como quiera entre esas dos honeynets, de forma que en cada cambio evoluciona su estructura. Todas las versiones generadas se analizan utilizando PCA para producir una firma que se pueda usar en cualquiera de las formas del gusano.

- Narvaez: valora en su estudio la probabilidad de que una honeypot de alta interacción tenga menos éxito en la identificación de páginas web que alojen malware si dicho malware es capaz de detectar la implementación de MV de la honeypot. El autor despliega dos sistemas, una honeypot en una MV y otra en un sistema completo. Ambas honeypots se conectan al mismo conjunto de páginas web maliciosas y se compara su efectividad en la detección de las mismas. El resultado observado sugiere que el uso de MVs en la implementación de honeypots de alta interacción no afectaría negativamente la probabilidad de una detección del atacante, una de las preocupaciones a la hora de utilizar honeypots.
- Newsome – "polygraph": automatiza la generación de firmas para gusanos polimórficos con un bajo porcentaje de falsos negativos y falsos positivos utilizando flujos de red que contienen ruido para generar la firma. La solución extrae la firma que consiste en un conjunto disjunto de sub-cadenas de un gusano polimórfico, en lugar de una sub-cadena particular, lo cual dota al proceso de mayor precisión. Para extraer las firmas de forma eficiente, la solución clasifica el contenido de tres secciones principales de código.
- Raynal: detalla el procedimiento forense para garantizar la fiabilidad de la información utilizando honeypots. Describe procedimientos forenses para investigar incidentes de intrusión en servidores tales como el análisis de actividad de red, análisis de sistema y archivos y recopilación de evidencias. La conclusión que obtiene es que una de las grandes dificultades se produce a la hora de desarrollar metodologías sistemáticas para utilizar la información recopilada por las honeypots para prevenir de forma eficiente ataques a la seguridad de la red conocidos o no.
- Su: aplica minería de datos para identificar nuevos patrones de ataque de forma automática. El autor identifica "episodios" como los nuevos patrones de ataque encontrados. Una vez identificados los episodios el autor propone la aplicación de dos algoritmos WINEPI y MINEPI a la información recopilada por una honeypot comercial llamada KFSensor. Este enfoque se demuestra útil a la hora de identificar gusanos como Korgo, Shelp y Sasser.
- Tang – "double honeypot": permite la extracción efectiva de firmas para detectar gusanos polimórficos que exploten vulnerabilidades de 0-day. También propone la utilización de un nuevo método de análisis para la información capturada llamado "Position Aware Distribution Signatures (PADS)". La honeypot doble aprovecha el hecho de que los gusanos crean conexiones salientes para identificarlos. PADS se diseñó para incrementar la posibilidad de detectar gusanos polimórficos, permitiendo variaciones en las firmas. Para controlar estas variaciones PADS utiliza una distribución de frecuencias de bytes, lo que establece una probabilidad sobre las variaciones en las firmas para cada posición en la cadena de firma.

Configuración de Honeypots

- Briffaut: desarrolla la arquitectura de honeypots de alta interacción distribuidas para la monitorización de anomalías, especialmente abusos y secuestros de honeypots de alta interacción, para automatizar el proceso de reinstalación de las honeypots afectadas. La solución monitoriza el estado de las honeypots desplegadas en la red, gracias a un IDS y otras herramientas, y reinstala automáticamente el sistema completo si se detectan anomalías. El mayor reto de esta solución es encontrar la forma de actualizar imágenes de sistema limpias, ya que los sistemas a veces experimentan cambios de forma legítima. Esto requiere el registro de cada actividad en las honeypots. Además, una reinstalación continua de honeypots ralentizará el sistema y aumentará el tiempo de inactividad de las mismas.

- Carroll: aplica conceptos de teoría de juegos para determinar la distribución óptima de honeypots dentro de una red. Establece cuatro tipos de sistemas: sistemas en producción normales, sistemas normales falsos (honeypots camufladas como sistemas en producción), honeypots (sin camuflar) y honeypots falsas (sistemas en producción que simulan el comportamiento de honeypots). El autor desarrolla modelos para conseguir equilibrio entre atacantes y defensores utilizando esta clasificación. Las técnicas se muestran útiles a la hora de distribuir honeypots.
- Chen: establece una serie de consejos y pautas a la hora de desplegar honeypots, especialmente centradas en el estudio de ataques de inyección SQL, remarcando la utilidad de las honeypots de alta interacción para las actividades relacionadas con el estudio de ataques a bases de datos. Recomienda la monitorización y restricción de ciertos procesos, puesto que se pueden utilizar para alterar el sistema. También aconseja la utilización de proxies entre la web y la BD. Por último aconseja el uso de honey tokens (información no real y rastreable).
- Hecker – "Honeyd Configuration Manager": permite el despliegue dinámico de honeypots tras el análisis de las características de la red en la que se van a desplegar. El gestor utiliza Nmap para escanear la red y recopilar detalles de los sistemas encontrados. Una vez realizado el escaneo el gestor de configuración despliega e inicia honeypots de baja interacción.
- Kohlrausch: describe sus experiencias analizando la información recopilada por una honeypot utilizando Argos para Detectar el gusano W32.Conficker. La honeypot usa DTA que marca cada byte de información recibido de la red y rastrea su actividad en el depurador. DTA se implementa con una combinación de honeypots Argos, el sistema IDS Snort y un depurador. Argos comprueba las amenazas desconocidas, mientras que Snort detecta ataques conocidos. El reto se produce a la hora de identificar acciones puntuales y enmarcarlas dentro de un ataque mucho más amplio y complejo.
- Spitzner: presenta técnicas para detectar amenazas internas utilizando honeypots y honey tokens. Plantea los diferentes retos hallados en las amenazas internas y que no se aprecian en las de tipo externo. Uno de los problemas es que estos atacantes conocen mucho mejor la topología de red y los sistemas, por lo que las honeypots deben ser de alta interacción. Dichos atacantes deben dirigirse hacia las honeypots sin que puedan ser conscientes de que no se encuentran en el sistema deseado. Debe proporcionarse una colección de información falsa que parezca veraz, tal como planes de negocio falsos y especificaciones de diseño incorrectas.

Sistemas anti-honeypot:

- Dornseif: demuestra cómo las honeynets pueden ser comprometidas y controladas por los atacantes sin que eso suponga una actividad inusual en los registros. En concreto, analiza el módulo Sebek y demuestra la facilidad con la que puede ser detectado. La solución a este problema pasaría por implementar Sebek como un parche del kernel y no como un módulo acoplado. También se recomienda el uso de nombres aleatorios para símbolos y variables para ayudar a dificultar la detección por parte de los atacantes.
- Holz: describe técnicas para detectar el despliegue de honeypots, determinando que los mecanismos para evitar dicha detección son uno de los grandes retos para las honeypots de alta interacción. Entre los mecanismos de detección destacan aquellos para determinar el uso de UML, VMWare, Chroot y jaulas de sistema, así como el modo de ejecución del procesador.
- McCarty: estudia el funcionamiento de la herramienta anti-honeypot "Honeypot Hunter", que fue desarrollada para ayudar a los spammers a detectar honeypots que actúen como proxy relay. El funcionamiento es simple, el software crea un servidor de correo local e intenta mandar un correo con el

mismo servidor como destino. Si la acción aparece como satisfactoria pero el servidor no establece la conexión, la honeypot es descubierta. También puede llevar a cabo una operación similar para los sistemas.

- Mukkamala: estudia técnicas de detección para honeypots de baja interacción y honeypots que se ejecuten en MVs mediante el análisis de tiempos y los servicios disponibles, así como TCP/IP fingerprinting. Las respuestas más lentas generalmente se corresponden a aquellos equipos con honeypots. También aprovecha una de las características básicas de las honeypots, la simulación parcial de servicios, para distinguir entre un servicio real y uno simulado.
- Perdisci – "noise injection": permite anular los generadores de firmas automáticos. La técnica de inyección de ruido confunde intencionadamente al generador de firmas para evitar la generación de una firma precisa impidiendo reconocer patrones de bytes válidos. Esta técnica utiliza el envío de flujos de tráfico de red basura que son similares a los flujos que contienen los gusanos polimórficos pero que no contienen ningún byte fijo.
- Wang: estudia diseños de botnets avanzados que permita predecir el comportamiento de las mismas en un futuro. Diseña una botnet peer-to-peer híbrida que es más difícil de desconectar, monitorizar y secuestrar. La clave de la botnet es la creación de dos tipos de bots, los "sirvientes", públicamente accesibles; y los clientes, ocultos gracias al direccionamiento privado o el uso de firewalls. La forma de contrarrestar este diseño es envenenar la botnet mediante el despliegue de una honeynet extensa, con direccionamiento IP estático, para saturar la lista de peers y conseguir el mayor número de información de la botnet.

Técnicas de detección de honeypots

- Propiedades técnicas de las honeypots:
  - Los tiempos de respuesta, banners, entradas en el registro, respuestas inconsistentes o ciertos parámetros.
- Propiedades del sistema en relación a la interacción con los usuarios:
  - Los tiempos de acceso y modificación de los archivos pueden hacer sospechar que el sistema no se usa como un usuario convencional lo haría.
- Captura de datos de red.
- Búsqueda de presencia de MVs:
  - Las soluciones más comunes de virtualización pueden ser detectadas fácilmente en un entorno local.
- Buscar trazas de herramientas típicas usadas en honeypots:
  - Ficheros temporales, volcados del núcleo, backdoors (Sebek)…
  - Buscar en los historiales de archivos/logs en busca de errores de configuración cometidos al desplegar la honeypot.
- Buscar vulnerabilidades y exploits que afecten al software de la honeypot.

ENFOQUE Y MÉTODO SEGUIDO

Para la realización de este proyecto se ha intentado obtener una visión lo más precisa posible de los sistemas que usualmente son objeto de ataques en Internet, respetando siempre las condiciones impuestas como mínimas requeridas.

Partimos con dos sistemas de base, una honeypot que emule un sistema Windows y otra que emule un sistema Linux. Para la máquina Windows utilizaremos un SO virtualizado Windows para probar herramientas para desplegar honeypots en esta plataforma. La honeypot en este caso emulará un sistema desprotegido sin

demasiado filtrado de puertos. En cuanto a la máquina Linux, para un primer acercamiento utilizaremos la distribución Honeydrive; con la que se implementará un equipo sometido a un proceso de hardening que solo acepta peticiones por el puerto 22 (SSH).

Llegados a este punto, en el que los requerimientos establecidos ya han sido cubiertos es el momento de analizar el resto de máquinas a desplegar. Hay dos motivaciones que han resultado importantes a la hora de decantar la elección por los sistemas huésped que alojen la honeypot. Puesto que la necesidad de utilizar una máquina Windows ya estaba cubierta, se ha optado por implementar el resto de las honeypots bajo Linux por dos sencillas razones. La primera es que la mayoría de sistemas que se van a emular son de tipo Linux, la segunda es la variedad de herramientas y documentación disponibles al respecto. Mientras que para Windows muchas de las herramientas son de código propietario y hay que pagar una licencia prohibitiva para el alcance de este estudio, mientras que muchas otras ya no se encuentran disponibles en Internet. En Linux, sin embargo, hay multitud de opciones a la hora de implementar honeypots. Destacan entre todas ellas las herramientas desarrolladas por el ya mencionado Honeynet Project.

Una vez elegido el SO principal sobre el que se implementarán las herramientas conviene analizar el tipo de honeypot a desplegar, de baja, media o alta interacción. Tras un estudio preliminar se descartan las honeypots de alta interacción debido al mayor número de problemas para configurarlas de forma segura y a las habilidades necesarias para hacerlo de forma satisfactoria que permitan obtener resultados fiables y precisos. Puesto que además el equipo que se utiliza para desplegar todas las honeypots no es dedicado y las configuraciones de alta interacción requieren un mayor número de recursos e infraestructura quedan completamente descartadas para este trabajo. No obstante, aunque es uno de los motivos principales, también se optó por no incluirlas debido a que las herramientas para implementarlas son más reducidas y su documentación no es tan abundante y exacta.

De entre las honeypots de baja y media interacción se ha optado por escoger una combinación de ambas. Así por ejemplo, Kippo implementa honeypots de media interacción, mientras que Conpot o Artillery representan el perfil de baja interacción.

Si seguimos subiendo en la abstracción, otra de las decisiones importantes es decidir qué tipo de máquinas van a simular las honeypots además de las dos básicas. Puesto que ya tenemos un sistema Windows y uno Linux considero que habría que cubrir el mayor número posible de sistemas atacados, para aumentar así la ventana de exposición a nivel de sistemas.

Para esto es importante entender las tendencias de ataque en Internet. Hoy en día, la proliferación de los servicios en Internet ha trasladado los objetivos de los ataques, observándose una tendencia cada vez mayor hacia las aplicaciones web y los servidores. Por desgracia, hay ciertos servicios que es difícil emular en la nube con los medios disponibles para la realización de esta investigación. Los sistemas en la nube y las plataformas móviles marcan la tendencia en cuanto a los objetivos de los atacantes en Internet. Debido a que la creación de una honeypot en una arquitectura de nube pública está prohibida por la mayoría de proveedores de servicios en la nube y podría acarrear un incremento desproporcionado en la tarificación debido al aumento intencionado por parte de un atacante del tráfico de red sobre la máquina desplegada, la opción más viable es abandonar esta vía. Con respecto a los dispositivos móviles, es cierto que es otro de los grandes objetivos en alza en la realización de ataques. Aunque mayoritariamente el malware en los dispositivos móviles se propaga mediante la instalación de aplicaciones maliciosas (incluso desde el Play Store) hay muy pocas alternativas para implementar una Honeypot que emule un sistema Android. La mayoría de las investigaciones al respecto se encuentran

todavía en una fase de definición abstracta. Por otro lado, puesto que las honeypots emulan un servicio no tendría mucho sentido emular un dispositivo móvil con puertos a la escucha, puesto que las características de estos sistemas son diferentes. En todo caso, puesto que no es el mayor vector de ataque para esta plataforma tampoco tiene mucha utilidad analizarlo con mayor profundidad.

Con respecto a los sistemas que sí es viable emular se han escogido dos elementos muy interesantes hoy en día. Por un lado, gracias a glastopf se emulará una aplicación web basada en Python. Esta honeypot ayudará a estudiar la relación entre los ataques a los sistemas tradicionales y ataques a nuevas plataformas, potencialmente mucho más valiosas para los atacantes hoy en día por el tipo y cantidad de información que pueden obtener.

Por último, una opción tremendamente valiosa y que representa uno de los sistemas más críticos hoy en día a nivel de seguridad es la honeypot SCADA. Con ella se emularán servicios típicos de este tipo de sistemas para observar las tendencias de ataque. La importancia de esta honeypot radica en la función del sistema que emula. Hoy en día la mayoría de investigaciones sobre honeypots hace referencia a la protección e investigación de ataques en sistemas SCADA, puesto que en muchos casos se encuentran protegidos y el daño que se puede hacer una vez dentro de los sistemas se considera en la mayoría de los casos de tipo catastrófico.

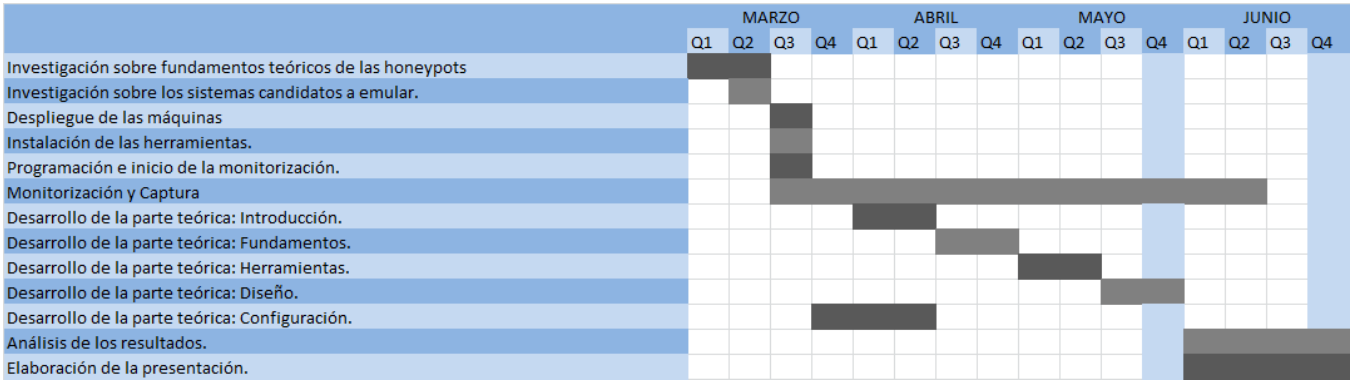
Por tanto, dado el nivel de habilidad y conocimiento actual sobre honeypots y su implementación, y las tendencias y utilidad de los tipos de honeypots para un estudio lo más completo posible de los ataques y distribución de malware hoy día; ésta es una configuración suficientemente estable, precisa y completa para la visión que pretende obtenerse del estudio.

PLANIFICACIÓN DEL TRABAJO

El trabajo a realizar se dividirá en distintas fases que se realizarán de forma secuencial con la posibilidad de solapamiento de alguna de ellas.

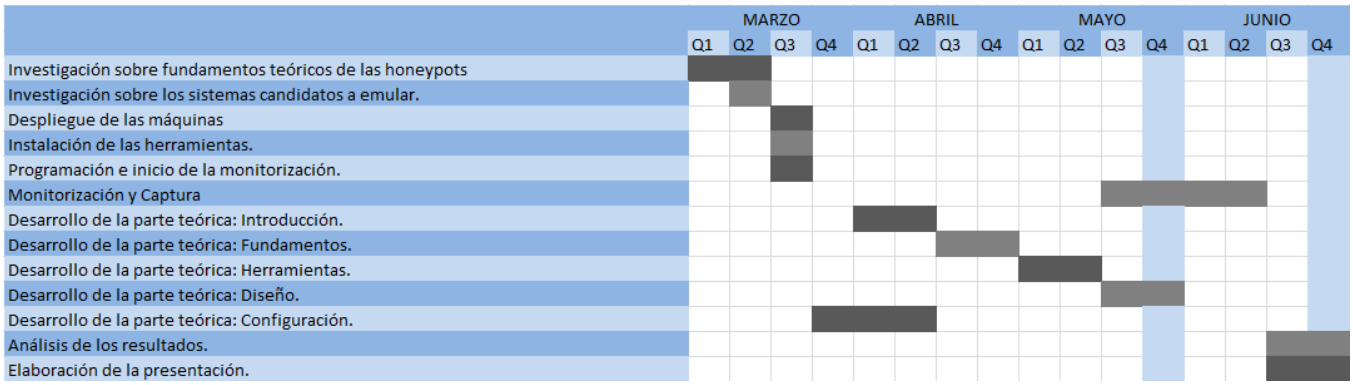
- Fase 1: Investigación sobre fundamentos teóricos de las honeypots.
- Fase 2: Investigación sobre los sistemas candidatos a emular.
- Fase 3: Investigación sobre las herramientas disponibles.
- Fase 4: Despliegue de las máquinas.
- Fase 5: Instalación de las herramientas.
- Fase 6: Programación e inicio de la monitorización.
- Fase 7: Monitorización.
- Fase 8: Desarrollo de la parte teórica del proyecto.
    - Fase 8.1. Introducción.
    - Fase 8.2. Fundamentos teóricos.
    - Fase 8.3. Herramientas.
    - Fase 8.4. Diseño.
    - Fase 8.5. Configuración.
- Fase 9: Captura y análisis de los resultados.
    - Las máquinas funcionarán cinco horas de forma ininterrumpida los días en los que permanezcan activas, esto permite el despliegue de todas las honeypots y al mismo tiempo evita una sobrecarga de trabajo excesiva del equipo anfitrión.
- Fase 10: Elaboración de la presentación.

Esquema de tiempo esperado



Problemas de tiempo hallados: Debido a la falta de ataques el tiempo de instalación de herramientas se ha prolongado un poco más en el tiempo, retrasando ligeramente el inicio del desarrollo de la parte teórica, así como el tiempo disponible para dedicar a esta sección. Debido a que sigo sin obtener ataques en las máquinas tampoco tengo muy claro cómo complementar el trabajo.

Esquema real



Como puede observarse el tiempo de captura se ha reducido considerablemente con respecto al tiempo esperado, cabe decir que aunque realmente las máquinas estuvieron funcionando todo el tiempo descrito en el primer esquema los ataques se produjeron en la franja de tiempo especificada en el segundo esquema. Dejando menos tiempo para el análisis de resultados y la elaboración de la presentación.

Capítulo II: FUNDAMENTOS TEÓRICOS.

DEFINICIONES PREVIAS

- 0-day: ataque informático basado en una vulnerabilidad en un sistema aún no descubierta o sobre la que no se ha informado. Su peligrosidad es extremadamente alta, puesto que no se han podido diseñar contramedidas.
- Ataque: acción mediante la cual un atacante intenta tomar el control, dañar, alterar el funcionamiento o negarlo al resto de usuarios sobre un sistema informático.
- Atacante: un individuo que utiliza una vulnerabilidad para explotar un sistema.
- Amenaza: elemento o acción capaz de atentar contra la seguridad de la información.
- Anfitrión: equipo o sistema que alojará en su interior sistemas adicionales. Referido a virtualización se refiere al sistema que alberga las máquinas virtuales.
- Auto-rooter: rootkits de nueva generación automatizados con capacidades avanzadas de ocultación.
- Blackhat: es la conferencia técnica anual sobre seguridad de la información más relevante a nivel mundial. Durante la Blackhat se llevan a cabo conferencias sobre nuevos hallazgos, sesiones de entrenamiento, charlas con expertos del mundo de la seguridad de la información…
- Bomba temporal: software especialmente diseñado para dejar de funcionar después de una fecha determinada.
- Bug: se trata de un error de software o fallo en un programa que desencadena resultados inesperados y que puede ser explotado para conseguir que dicho software proporcione funcionalidades para las cuales no había sido diseñado originalmente.
- Cibercriminal: persona que lleva a cabo actividades delictivas a través de Internet.
- Código propietario/Software propietario: Cualquier programa informático que no es libre total o parcialmente, debido a la prohibición de uso, redistribución o modificación sin licencia o permiso expreso del titular del software.
- cPanel: herramienta de administración basada en tecnología web para administrar sitios cómodamente a través de una interfaz.
- Debugger/Depurador: herramienta/software que permite la ejecución controlada de un código para poder rastrear el funcionamiento interno del mismo y su interacción con otros elementos del sistema. Su función principal es la de encontrar bugs o códigos de protección.
- DoS (ataque de denegación de servicios): ataque a un sistema que vuelve un servicio inaccesible para los usuarios legítimos. Generalmente logrará su labor consumiendo todo el ancho de banda de la red de la víctima, lo que provocará la pérdida de la conectividad. Utiliza grandes flujos de información para saturar los equipos que deben procesar los paquetes.
- Encoder: es un dispositivo, circuito, software, algoritmo o cualquier otro elemento que convierte información de un formato de codificación a otro.
- Exploit: es un software o una secuencia de comandos que explota un bug, fallo o vulnerabilidad para causar un comportamiento inesperado sobre algún software, hardware o algún componente electrónico. Este comportamiento supone generalmente la explotación del sistema atacado.
- Fingerprinting: se trata de una de las primeras fases en el proceso de explotación, llamada también análisis activo. Sigue a la fase de footprinting o análisis pasivo. En ella se lanzan distintas sondas contra los elementos objetivo para intentar obtener información sobre su arquitectura y funcionamiento.

- Firewall: es un sistema de seguridad de red de tipo perimetral implementado por software o por hardware que controla el tráfico de red saliente y entrante utilizando un conjunto de reglas preestablecidas.
- GSoC: es un programa anual en el que Google premia a todos los alumnos que completen satisfactoriamente un proyecto propuesto sobre software open source durante el verano. Una vez que los mentores designados por las asociaciones participantes elijan un proyecto los integrantes del mismo recibirán 5500$.
- Hacker: persona apasionada por la seguridad informática que intenta extraer todas las funcionalidades posibles de los sistemas/redes/aplicaciones que estudia y que intenta ampliar su conocimiento constantemente investigando sobre temas diversos relacionados con la seguridad informática. Dentro de este grupo de personas se encuentran tanto los que se dedican a asegurar y proteger los sistemas como los que se encargan de romper su seguridad.
- Host: se trata de un equipo conectado a una red de computadores. Un host puede ofrecer recursos, servicios y aplicaciones a usuarios localizados en otros nodos de la red. Un host debe tener asignado una dirección de host a nivel de red.
- Huésped: es un sistema alojado dentro de otro sistema. En entornos de virtualización se trata de las máquinas desplegadas dentro del sistema de virtualización.
- IDS: dispositivo o aplicación software que monitoriza la actividad en redes o sistemas en busca de comportamientos maliciosos o violaciones en las políticas y procedimientos de uso. Hay distintos tipos de IDSs dependiendo del lugar en el que se sitúen y el tipo de información que deban monitorizar.
- Jaula de sistema: es una implementación a nivel de sistema operativo que permite tener sub-divisiones de un sistema y que teóricamente son independientes entre sí.
- Kernel: es un programa que gestiona las peticiones de entrada salida del software y las traduce en instrucciones de procesamiento de información para la unidad central de procesamiento y otros componentes electrónicos.
- Malware (malicious software): es cualquier software utilizado para perturbar operaciones computacionales, recopilar información sensible o ganar acceso a sistemas de computación privados. Dentro del malware existen numerosos subtipos dependiendo de su finalidad, estructura y comportamiento.
- MITM (man-in-the-middle): es una forma de escucha activa en la que el atacante establece conexiones independientes con la víctima y el objetivo del tráfico enviado por dicha víctima para capturar, analizar, modificar en algunos casos y reenviar posteriormente el mensaje a su destinatario.
- MV: es una solución software que emula una computadora. Las máquinas virtuales operan con una arquitectura y funcionalidades similares a los de una máquina real hipotética.
- Open Source: es un modelo de desarrollo que promueve el acceso universal al software a través de las licencias gratuitas y la redistribución universal con el fin de modificar el diseño o la implementación siempre que se mantengan los principios originales.
- Payload: es el código asociado a un exploit que se encarga de llevar a cabo las acciones dañinas.
- Peer: en un modelo descentralizado
- Phishing: el acto de intentar adquirir información sensible sobre los usuarios suplantando la identidad de una entidad de confianza en una comunicación electrónica.
- Sandbox: mecanismo de seguridad para ejecutar programas sospechosos en un entorno controlado y aislado del resto del sistema. Proporciona recursos estrictamente controlados tales como espacio de uso temporal en el disco y la memoria. Generalmente se deshabilita el uso de la red y la posibilidad de leer desde dispositivos de entrada.

- SCADA (supervisory control and data acquisition): es un sistema operativo que codifica señales a través de canales de comunicación para proporcionar control a equipos remotos. Generalmente estos sistemas se utilizan para llevar a cabo tareas de control industrial.
- Torrent: es un tipo de archivo utilizado en redes de intercambio p2p que contiene metadatos sobre archivos y directorios que son distribuidos, así como una lista de localizaciones de red de ordenadores que interconectan la red. El archivo torrent no contiene los archivos a distribuir, sino información sobre los mismos.
- UML: es un tipo específico de kernel de sistema creado principalmente para desarrolladores y usuarios avanzados del kernel tradicional de Linux. UML permite desarrollar nuevas versiones del kernel del sistema de forma segura y experimentar con ellos.
- Vulnerabilidad: puntos débiles del software y los sistemas que pueden ser utilizados por un atacante para comprometer la integridad, disponibilidad o confidencialidad de dicho software o sistemas.

DEFINICIÓN DE HONEYPOT

Una honeypot es una herramienta extremadamente flexible con distintas configuraciones y extensiones que, a pesar de no solucionar ningún problema de seguridad específico como lo haría un firewall o un IDS, puede aportar una gran cantidad de información valiosa que ninguna otra herramienta puede. Su utilidad fundamental es la de detectar ataques por muy sofisticados que éstos sean. Una de sus cualidades principales es la flexibilidad, que además las dota de un gran potencial y que supone uno de los mayores retos a la hora de definirlas y comprender su funcionamiento. Otra diferencia con la mayoría de herramientas de seguridad es que pueden tomar diversas formas.

De manera formal podríamos definir una honeypot como un sistema de TI ficticio y vulnerable establecido para detectar, desviar y en algunos casos contrarrestar intentos de uso no autorizado de los sistemas de información legítimos, mediante el ataque, exploración, explotación y comprometimiento del sistema ficticio por parte de atacantes. Otra de sus características fundamentales es el uso de aislamiento y la monitorización para emular un comportamiento en un sistema que parece contener información o recursos valiosos para los atacantes. En las honeypots se crea por tanto un valor adicional de los recursos una vez que estos son atacados, mientras que apenas poseen valor si nunca son atacadas o detectadas.

Requerimientos clave de una honeypot:

- Control de datos: determinar cómo se contendrán las actividades del atacante para restringirlas únicamente al ámbito de la honeypot.
- Captura de datos: capturar toda la actividad del atacante sin que éste lo sepa.
- Análisis de datos: analizar los datos capturados.
- Recolección de datos: recopilar toda la información de diversas honeypots que hagan referencia a una única fuente de ataque.

Diferencia entre honeypot y honeynet:

A diferencia de las honeypots, que se implementan sobre un único equipo y que pueden configurarse con distintos niveles de interacción, una honeynet se extiende sobre una red entera y el grado de interacción es siempre alto, para que así pueda recopilarse mucha más información.

Las honeynets están especialmente diseñadas para la investigación de nuevas técnicas de ataque y se consideran el concepto de honeypot de alta interacción llevado al extremo. Esto se logra desplegando sistemas convencionales dentro de una red altamente controlada.

El reto no se encuentra en desplegar las honeypots, ya que los sistemas no requieren demasiadas configuraciones más allá de la instalación básica del sistema operativo, sino construir una red controlada que monitorice y capture toda la actividad que ocurra dentro de su alcance. Es por tanto el tipo de honeypot que más riesgo tiene asociado. Como ventaja destacar que aportan el mayor número de información posible.

Honeytokens:

Son un tipo de honeypot no basado en sistemas computacionales. Los honeytokens pueden adoptar infinidad de formas: una cuenta falsa, una dirección de e-mail o una entrada en una base de datos que solo será consultada por consultas maliciosas… Esta definición las hace tremendamente flexibles y versátiles. En general no previenen el acceso a los datos, pero aporta al administrador una medición de la fiabilidad en la integridad de los datos.

Se eligen para que adopten valores únicos y para que aparezca en el tráfico legítimo, pero pueden ser fácilmente detectadas por un IDS, alertando al administrador de un comportamiento que no debería estarse llevando a cabo, por lo tanto además de asegurar la integridad también pueden usarse para detectar actividades maliciosas poco agresivas.

VALOR DE UNA HONEYPOT

Una honeypot puede prevenir un ataque de diversas maneras:

- Por un lado previene ataques automatizados como los gusanos o los auto-rooters. Estos ataques utilizan herramientas que escanean de forma aleatoria porciones de red completas hasta encontrar sistemas vulnerables. Si dichos sistemas se encuentran, las herramientas automatizadas atacarán y comprometerán el sistema. Por consiguiente, si se ralentizan estos escaneos hasta el punto de prácticamente detenerlos, conseguirá frustrarse el ataque. Esto se puede realizar utilizando ciertas particularidades del protocolo TCP.
- Una vez que el ataque es detectado también se puede responder y parar al atacante.

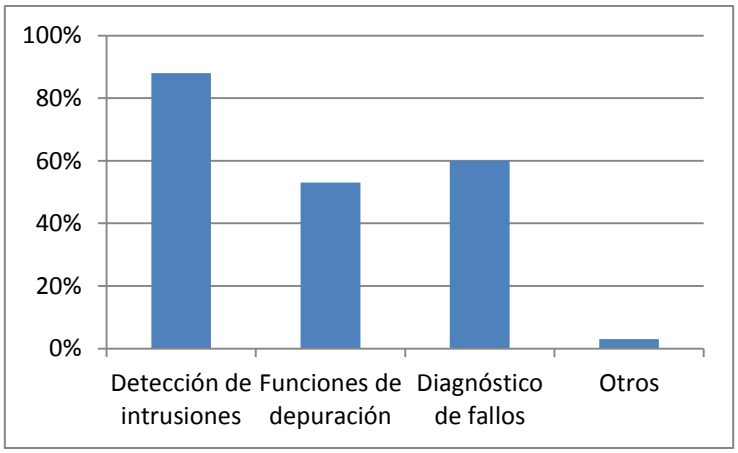


Figura 3. Objetivos generales de los sistemas de detección.

Si un atacante se dispone a atacar un sistema y detecta el uso de honeypot, primero tendrá que averiguar qué sistemas forman parte de la honeynet, y temerá ser registrado por el sistema, por lo que es probable que desista.

Además, una honeypot ayuda a proteger a la organización mediante la detección de ataques. Independientemente de la configuración de la organización, siempre habrá fallos potencialmente explotables. Al detectar a un ataque de forma rápida se aumenta el tiempo de reacción antes de que se produzca el daño para detenerlo o mitigarlo. Las honeypots son una opción mucho mejor que el resto de herramientas de seguridad a la hora de detectar y abordar los problemas de detección, reduciendo además la tasa de falsos positivos gracias al volumen de datos capturados, especialmente en las de baja interacción.

Por último, las honeypots también son efectivas a la hora de responder a los ataques, siendo esta tarea uno de los mayores retos para la seguridad de una organización. Esta tarea es especialmente crítica en los sistemas en producción como los servidores de correo electrónico, puesto que aun habiendo sido comprometidos, no siempre es posible desconectar el sistema para realizar un análisis forense adecuado. Esto dificulta la tarea de análisis de lo sucedido, la forma en la que el atacante ha actuado y el daño causado e incluso saber si el atacante ha atacado otros sistemas. Aun desconectado el sistema para poder analizarlo el volumen de información sigue siendo muy elevado. Puesto que las honeypots pueden desconectarse fácilmente y los datos a analizar suelen ser bastante precisos la tarea de análisis posterior al ataque se vuelve mucho más rápida y sencilla, sabiendo que toda la actividad registrada por estos sistemas es maliciosa y no autorizada. Por lo que las honeypots aportan una manera rápida, efectiva y profunda de responder a un incidente, siendo las honeypots de alta interacción las más efectivas para este propósito, aunque también son las más difíciles de configurar y las que más conocimientos requieren para ser desplegadas de forma efectiva y segura.

TIPSO DE HONEYPOTS

Según el nivel de interacción:

Nivel de interacción	Instalación y configuración	Despliegue y mantenimiento	Recolección de información	Nivel de riesgo
Bajo	Fácil	Fácil	Limitada	Bajo
Medio	Variable	Variable	Variable	Medio
Alto	Difícil	Difícil	Extensiva	Alto

El nivel de interacción determina las actividades que se pueden realizar en el entorno de la honeypot. Al mismo tiempo, también determina la información que aporta sobre la actividad del atacante. El riesgo aumenta por tanto a medida que aumenta el nivel de interacción, ya que el entorno se amplia y las operaciones que se permite realizar al atacante aumenta, existe la posibilidad de que un atacante aproveche una mala configuración de una honeypot de alta interacción para que se salte la protección de la honeypot y acceda al sistema huésped u obtenga gran cantidad de información del mismo.

Baja interacción:

Suelen emplearse en los entornos en producción y ayudan a protegerlos. Es el tipo que menor interacción permite, limitándose en la mayoría de los casos a mostrar determinados puertos abiertos (simulación de servicios) y a permitir intentos de conexión (que serán rechazados, puesto que no hay ningún servicio útil implementado escuchando) así como a mostrar los banners de servicio, por lo que no hay que preocuparse sobre cómo contener las acciones del atacante.

Por su nivel de complejidad son las más fáciles de instalar, configurar, desplegar y mantener, ya que su funcionalidad y su diseño son muy básicos. Cada vez que se observa un intento de conexión de una IP desconocida se intercepta la conexión y se produce la interacción con el atacante. Los datos más comunes de ataque son la información del origen del ataque y las credenciales utilizadas para intentar acceder al servicio simulado. Debido a estas características estas honeypots no detectarán gran cantidad de exploits desconocidos. Puesto que las honeypots también pueden emular sistemas operativos, dependiendo de la configuración de las herramientas el atacante observará diversos elementos al analizar el rastreo de la red, desde equipos convencionales, routers, servidores hasta elementos más complejos como sistemas SCADA de control industrial.

Este tipo de honeypot suele asociarse a las honeypots virtuales, ya que el nivel de interacción requerido permite virtualizar los servicios en lugar de ofrecerlos realmente.

Utilizan análisis estático (coincidencia de firmas y métodos heurísticos) que permite detectar ciertas respuestas maliciosas que pasarían desapercibidas en honeypots de alta interacción, tales como bombas temporales.

- Funciones que una honeypot de baja interacción debe llevar a cabo:
  - Interactuar con el servidor, realizando peticiones y procesando la respuesta.
  - Crear una cola de peticiones de ejecución del servidor hacia la herramienta.
  - Analizar el sistema o la respuesta del servidor en busca de violaciones de las políticas de seguridad de los sistemas.
  - Detectar escaneos o intentos de conexión no autorizados.

- Información obtenida de una honeypot de baja interacción:
  - Tiempo y fecha del ataque.
  - Dirección IP y puerto destino del ataque.
  - Dirección IP y puerto origen del ataque.
  - SO del atacante.

- Ventajas de las honeypots de baja interacción:
  - Buen punto de partida.
  - Fáciles de instalar
  - Riesgo introducido muy limitado.
  - El registro y análisis de la actividad maliciosa es sencillo (en parte porque no hay tanto volumen de información que recoger y ésta no es muy variada).

- Desventajas de las honeypots de baja interacción:
  - Poco funcionales.
  - No existe interacción real con el atacante.
  - La capacidad de registrar información está muy limitada (debido a las actividades permitidas).
  - Solo capturan ataques conocidos.
  - Son muy fáciles de detectar por atacantes experimentados.

Media interacción:

Ofrecen mayores posibilidades de interacción que las de baja interacción pero menos funcionalidades que las de alta interacción. Esto se limita básicamente a emular parcialmente algún servicio vulnerable más allá

de mantener el puerto a la escucha. Otra funcionalidad adicional de este tipo de honeypot es la de capturar el malware que se intente cargar en la honeypot.

Este tipo de honeypot puede desplegarse en un entorno virtualizado que simule uno real, de forma que la actividad del atacante queda enjaulada dentro del entorno virtualizado.

- Funciones que debe llevar a cabo:
  - Similares a las de baja interacción.
  - Emulación parcial de servicios.
- Información obtenida:
  - Información similar a la que se obtiene en las de baja interacción.
  - Información sobre actividad superficial, tal como comandos introducidos en la consola.
- Ventajas:
  - Mayor información que en las de baja interacción sin los riesgos elevados de las de alta interacción.
  - Mayor dificultad de detección que las de baja interacción.
- Desventajas:
  - Muy complejo y con mayor posibilidad de una configuración deficiente.
  - Es muy difícil dotar al entorno virtualizado de funcionalidad completa y operatividad como en un sistema real.
  - Mayor tiempo para instalar y configurar el entorno que las de baja interacción.
  - Mayor número de modificaciones sobre la configuración original que las de baja interacción.
  - Requiere la securización del entorno para poder desplegarse de forma segura.

Alta interacción

Estas honeypots no deben ser entendidas simplemente como un producto que se instala en un sistema. En su lugar, se componen de un conjunto de herramientas, una arquitectura o una red completa de ordenadores que se prepara para recibir los ataques. El entorno generado debe estar estrechamente controlado, junto con todas las actividades que se lleven a cabo dentro de sus límites. Además, todas estas actividades deben ser registradas para su posterior análisis sin que los atacantes lo perciban.

Debido a sus características no suele implementarse en entornos en producción y se relega su uso para fines de investigación.

Las configuraciones sobre las máquinas y las honeypots deben ser profundas, adaptándolas de forma lo más perfecta posible al entorno de despliegue y a las características requeridas.

La única característica que define a estos sistemas como honeypots es que no hay información útil en su interior, aunque debe simularse su existencia para que resulten atractivas a los atacantes. A diferencia de otras honeypots, las de alta interacción deben situarse detrás del firewall para minimizar el riesgo asociado y al mismo tiempo desde las reglas del firewall debe impedirse que la máquina pueda lanzar ataques a otros sistemas.

- Funciones que deben llevar a cabo:
  - Ofrecer un SO completo para que el atacante interactúe con él.
  - Registrar y monitorizar todas las actividades que lleve a cabo el atacante sin que éste se dé cuenta.
  - Asegurar la máquina para que la interacción dentro de la honeypot no afecte a otros sistemas.

- Información obtenida:
  - Información basta y completa sobre los ataques y los atacantes.
  - Información sobre ataques y malware desconocidos.
- Ventajas de las honeypots de alta interacción:
  - Altamente funcionales.
  - Información recopilada abundante y precisa sobre las actividades.
  - Mayor oportunidad de aprender sobre el atacante y su metodología.
  - Ayudan a prevenir ataques futuros aún no descubiertos.
  - Permiten comprender mejor las amenazas.

- Desventajas de las honeypots de alta interacción:
  - Es costoso en tiempo construir, configurar, desplegar y mantener una honeypot de este tipo.
  - Hay que saber utilizar distintas herramientas que deben adaptarse a la situación.
  - Requiere un gran esfuerzo analizar una honeypot comprometida por un atacante (una media de 40 horas por cada 30 minutos que el atacante haya actuado sobre el sistema) tanto en tiempo como en dificultad (identificar y analizar todas las modificaciones y el malware introducido en el sistema).
  - Se introduce un nivel elevado de riesgo sobre el sistema y puede comprometer la seguridad de la organización en su totalidad si no se toman las medidas adecuadas.

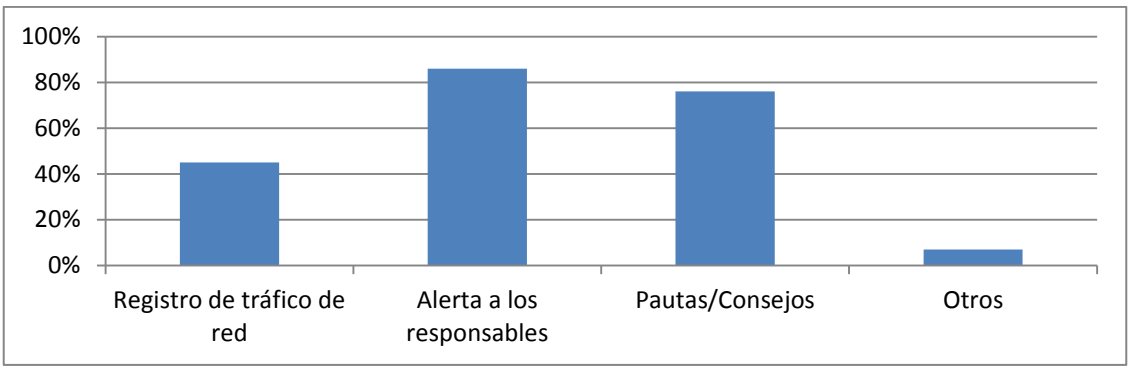


Figura 4. Acciones llevadas a cabo por las honeypots en caso de ataque.

Según su despliegue:

No existe una diferencia significativa de funcionamiento de la honeypot que sea dependiente del despliegue, las diferencias fundamentales radican en el entorno operativo. Las consideraciones que hay que tener en cuenta a la hora de elegir un modelo u otro radican básicamente en el riesgo y la recompensa.

Físicas.

Sistemas que se despliegan con el único propósito de servir como honeypot y que no utiliza ningún mecanismo de virtualización para tal fin.

Aunque son más caras que las honeypots virtuales, son más difíciles de identificar, ya que el atacante tendrá las respuestas usuales de un sistema real. Aunque este hecho por sí solo no identifique de forma precisa la existencia de una honeypot sí puede hacer que el atacante sospeche.

Como desventaja está el hecho de que un sistema honeypot real comprometido puede permitir al atacante utilizar el sistema para actividades maliciosas. En este aspecto hay que diferenciar entre las honeypot Windows y las Linux, siendo las primeras más vulnerables a este hecho que las segundas.

Virtuales.

Son máquinas sobre las que se virtualizan una serie de servicios que serán analizados y registrados.

La ventaja de utilizar una honeypot virtual radica en el hecho de que son más económicas y más seguras. Para aquellas honeypots que se desplieguen en un sistema totalmente virtualizado, además aunque un atacante consiguiera penetrar en el sistema subyacente todavía tendría que superar una capa de virtualización para tomar el control del sistema real. Además, una vez comprometida la máquina virtual, ésta es fácilmente reemplazables sin la necesidad de detener la actividad de la honeypot mientras se analiza la imagen comprometida.

Como desventaja está el hecho de que los servicios virtualizados solo engañarán a los atacantes menos experimentados y habilidosos. Además existe una mayor limitación con respecto a la información que se registrará.

Según el sistema en el que se despliegue:

Honeypot en producción:

- Previene, detecta y ayuda a responder a ataques sufridos por la organización.
- Ayudan a mitigar riesgo en la organización o el entorno.
- Proporcionan un valor específico cuantificable a la seguridad del sistema y las redes.
- A nivel de prevención:
    - Puesto que las honeypots pueden hacer perder mucho tiempo y recursos a un atacante esto indirectamente ayuda a prevenir los ataques sobre la organización.
    - Si el atacante detecta honeypots desplegadas en un entorno quizás se replantee su intención de atacar el entorno.
- A nivel de detección:
    - Reducen el nivel de falsos positivos: debido a que no hay tráfico legítimo que deba alcanzar la honeypot
    - Reduce el nivel de falsos negativos: ya que es difícil que los nuevos ataques pasen inadvertidos.
    - Reduce el volumen de datos capturados: informando solo de información útil sobre ataques reales.
    - Hay que tener en cuenta que en ningún caso deben considerarse a las honeypots como una solución infalible para la seguridad (en seguridad no existen las balas de plata), por lo que deben desplegarse otras herramientas de seguridad en el entorno.
- A nivel de respuesta.
    - A diferencia de otros sistemas en producción, las honeypots pueden ser sometidas a un análisis riguroso sin afectar al funcionamiento de la organización.
    - En las honeypots es más fácil detectar cambios en logs y registros.
    - La información del ataque almacenada en las honeypots no se corromperá con el funcionamiento normal de la misma.

Honeypot para investigación:

- Recopilan información sobre nuevos ataques y herramientas para llevar a cabo los mismos.
- Responde a preguntas sobre quién amenaza a los sistemas, los motivos del ataque o cómo los llevan a cabo, además de la frecuencia de los ataques en caso de producirse de manera repetida.
- Permiten obtener toda esta información de manera relativamente económica si lo comparamos con otros mecanismos de investigación.

- A la hora de comprender al atacante en una honeypot no se depende tanto del establecimiento de hipótesis, puesto que puede verse de forma precisa información que de otro modo habría que suponer.
- Aporta una plataforma de estudio de ciber-amenazas.
- Usos:
    - Captura de amenazas automatizadas analizando los payloads del malware capturado.
    - Prevención de futuros ataques en función del análisis de la información recopilada.
    - Captura de herramientas y técnicas desconocidas.
    - Adquirir un mayor entendimiento de las motivaciones y organización de los atacantes.
    - Obtener información sobre hackers avanzados.
- No reduce el riesgo de forma directa como las honeypots en producción, pero pueden servir como una fuente de información tremendamente valiosa.

VENTAJAS GENERALES

Se recopilan cantidades relativamente pequeñas de información con un gran valor:

La información registrada por las honeypots es relativamente pequeña (del orden de MB por día), por lo que genera un número pequeño de alertas y la cantidad de información que es necesario procesar se reduce. Además, existe la seguridad de que toda la actividad registrada será de tipo malicioso, por lo que no hay que aplicar un proceso de criba para descartar aquella que no sea útil, de hecho, toda la información recopilada en una honeypot es de utilidad. Esto reduce el ruido y vuelve la tarea de análisis de los datos capturados mucho más fácil y barata, así como el valor de los mismos.

Nuevas herramientas y metodologías:

Las honeypots se diseñan para capturar cualquier actividad lanzada contra ellas, por lo tanto a diferencia de otras herramientas de seguridad son capaces de capturar ataques nunca vistos o herramientas que nunca antes habían sido utilizadas.

Número reducido de recursos:

Las honeypots no requieren un número elevado de recursos debido a que solo se encargan de capturar actividad maliciosa. Casi cualquier sistema posee los recursos necesarios para implementar una honeypot en su interior. Además, tampoco tiene un impacto significativo sobre el ancho de banda de la red.

Adaptación a encriptación e IPv6:

A diferencia de muchas otras herramientas de seguridad las honeypots pueden operar en entornos encriptados o con direccionamiento de tipo IPv6. Cualquier ataque seguirá siendo capturado por la honeypot.

Información:

Las honeypots capturan información con un nivel de profundidad que es difícil de alcanzar por otras herramientas.

Simplicidad:

A nivel conceptual no se trata un software muy complejo (especialmente las de baja y media interacción). Las honeypots no utilizan algoritmos complejos, tablas que mantener o firmas que deban ser actualizadas, por todo ello debido a su simpleza es menos probable que ocurran fallos de configuración o errores.

Reducción de falsos positivos y falsos negativos:

Debido a que toda la información registrada es parte de actividad maliciosa y los volúmenes reducidos de información a analizar.

Entrenamiento:

Ya que las honeypots registran ataques y ayudan a detectar nuevos ataques y herramientas son extremadamente útiles entrenando al personal de seguridad de la organización para mantenerlo actualizado en cuanto a la evolución de los ataques y las herramientas.

Retorno de la inversión:

A diferencia de otras herramientas, con las que cuesta evaluar el retorno de inversión, las honeypots demuestran su valor rápidamente y de forma continua. A medida que son atacadas el personal responsable de las inversiones puede comprobar su utilidad mediante la captura de actividad no autorizada. Además, también pueden motivar la implementación de otras medidas de seguridad en base a los ataques recibidos y su utilidad quedaría demostrada con el análisis posterior de las mismas honeypots.

DESVENTAJAS GENERALES

Visión limitada:

Aunque este punto depende en gran medida del tipo de honeypot que se despliegue, en general las honeypots solo capturan ataques que interactúan de forma directa con la honeypot. Por lo tanto no capturarán ataques contra otros sistemas, a menos que el atacante o la amenaza también interactúe con la honeypot.

Riesgo:

Las honeypots tienen riesgos asociados que aumentan con el nivel de interacción con el atacante. Existe la posibilidad de que el atacante consiga anular la honeypot y utilice el sistema para llevar a cabo actividades maliciosas.

Riesgo transferido:

Desplegar una honeypot en un entorno operativo no solo tiene el riesgo inherente asociado a la propia honeypot, también aumenta el riesgo del resto de las máquinas, especialmente cuando la honeypot ha sido mal configurada.

Aumento de los ataques:

Si la implementación de una honeypot no se estudia cuidadosamente y no está justificada puede atraer la atención de ciertos atacantes con poca habilidad que intenten aprovechar un sistema con una vulnerabilidad claramente identificable.

Coste en tiempo:

Una honeypot no aporta información útil a menos que se dedique cierto tiempo a administrarlas, por lo que no pueden utilizarse como una herramienta automatizada, debe tenerse la suficiente habilidad y conocimiento en el tema.

Fuentes de los ataques:

Una honeypot aporta información valiosa sobre los atacantes externos a un sistema, pero poco sobre los atacantes internos; siendo éstos mucho más frecuentes que los primeros. Si una empresa sufre muchos más ataques internos que externos y estos últimos no suponen un riesgo excesivamente alto entonces una honeypot no es la mejor solución de seguridad al problema de los ataques.

Fingerprinting:

En las versiones comerciales usuales es relativamente fácil identificar la identidad real de una honeypot, ya que posee ciertas características y comportamientos fijos. Este riesgo se ve aumentado en las honeypots de investigación, ya que pretenden capturar ataques más avanzados y por tanto los atacantes también poseerán una mayor habilidad.

Capítulo III: HERRAMIENTAS

HERRAMIENTAS DE VIRTUALIZACIÓN.

VirtualBox

Es una herramienta multiplataforma propiedad de Sun Microsystems desde 2008. Permite instalar sistemas operativos de forma virtualizada en un equipo que actúe como huésped, con la posibilidad de asignación de recursos completa e individualizada a través de su consola de configuración.

Los sistemas operativos soportados se encuentran GNU/Linux, Mac OS, OS/2 Warp, Windows y Solaris, pudiendo escoger entre una gran variedad de versiones y distribuciones dentro de las citadas anteriormente.

El software se distribuyó con una licencia de código propietario, aunque actualmente se distribuye como Open Source en su versión empresarial. La licencia de código privado sigue manteniéndose pero se restringe a uso doméstico personal.

Permite montar imágenes de disco como unidad óptica virtual y almacena los discos duros de los sistemas virtualizados en ficheros dentro del anfitrión.

Aunque no goza de todas las funcionalidades de otras herramientas de virutalización ofrece un servicio de calidad y permite, a diferencia de otros soluciones, ejecutar sistemas de forma remota con RDP (Remote Desktop Protocol).

La memoria mínima por SO es de 512 MB de RAM, aunque la mayoría de sistemas requerirán una cantidad mayor.

A nivel de direccionamiento de red tiene opciones muy interesantes, pudiendo establecer interfaces de red tradicionales basadas en NAT (creando una red interna para la máquina que tendrá conectividad con Internet) y conexión puente (se crea una IP adicional en el sistema virtual que se encuentra dentro del rango de IPs de la máquina física), pero además pudiendo crear una conexión para que sólo el host tenga acceso a la interfaz de red virtualizada; lo cual es muy útil para realizar pruebas sobre sistemas vulnerables sin exponerlos ataques externos.

HERRAMIENTAS PARA DESPLEGAR HONEYPOTS.

Artillery

Se trata de una herramienta de respuesta activa avanzada para detectar atacantes antes de que alcancen el resto de recursos de la red. Está escrita en Python bajo licencia Open Source y se ejecuta en sistemas Linux.

Su funcionalidad es compleja, proporcionando servicios de honeypot, monitorización de archivos del sistema, hardening del sistema, información sobre amenazas en tiempo real y un servidor seguro de propósito general sobre el que se montará la honeypot.

Sus objetivos principales son aumentar la seguridad de los sistemas del entorno y dificultar la labor de los atacantes a la hora de penetrar en el sistema.

Artillery monitorizará el sistema de ficheros en busca de cambios en el mismo. Si detecta alguno, enviará un e-mail al propietario del servicio (la configuración por defecto funciona con correos en Gmail). También puede llevar a cabo el envío si detecta ataques de fuerza bruta sobre SSH, bloqueando además la IP del atacante.

Posee un sistema inteligente de gestión de amenazas que bloqueará a atacantes que hayan sido detectados por otro sensor desplegado en Internet (aunque dicha funcionalidad debe configurarse en los ficheros apropiados). Esto ayuda a bloquear ataques conocidos aunque dichos ataques no hayan amenazado un sistema concreto de la red Artillery. Esta tarea se realiza de forma coordinada a nivel mundial por el equipo de TrustedSEc gracias al ATIF (Artillery Threat Intelligence Feed).

Como todas las honeypots, Artillery abrirá una serie ampliable de puertos predefinidos correspondientes a los servicios más comunes, como RPCV/SMB, MS SQL o VNC, buscando posteriormente conexiones a dichos puertos. Al atacante le será devuelto un conjunto aleatorio de información, que hará aparecer un banner irreconocible. Esta sea quizás una de las desventajas de Artillery, ya que el atacante podría sospechar de la existencia de la honeypot. No obstante, los banners que se envían pueden ser modificados para que no contengan caracteres generados aleatoriamente.

En la página web puede encontrarse un histórico de ataques ocurridos organizados geográficamente de forma muy actualizada.

Kippo

Kippo es una solución honeypot de media interacción para sistemas Linux que registra ataques de fuerza bruta sobre SSH (22/tcp). Una de sus cualidades más importante es que una vez logrado el acceso permite cierta interacción con una Shell simulada que posteriormente ofrecerá información al administrador sobre los comandos ejecutados.

Kippo ofrece la visión de un sistema Linux fuertemente fortificado, ya que si solo se instala kippo sobre el sistema orignal, este solo tendrá abiertos los puertos 22/tcp para SSH y 80/tcp para HTTP.

Entre las características de Kippo destacan:

- Simula un sistema de ficheros falsos que permite la eliminación y creación de archivos. El sistema de ficheros emula una arquitectura Debian 5.0.
- Posibilidad de añadir contenido falso a los archivos, de forma que al realizar operaciones como cat /etc/passwd se mostraría información personalizada. Tras la instalación solo se incluye un número muy limitado de información en determinados archivos clave.
- Registro de sesiones almacenado en un formato compatible con ULM, que ofrece información sobre las marcas de tiempo de los accesos.
- Almacena los archivos descargados para una inspección posterior.
- Puesto que el servicio SSH es simulado permite modificar el comportamiento de ciertas sentencias para que no se comporten sobre el sistema como lo harían usualmente.

KFSensor

KFSensor destaca con la facilidad con la que se instala y las pocas exigencias de conocimientos a la hora de configurar la herramienta de forma satisfactoria. No se requiere un hardware específico para su funcionamiento, simplemente un equipo que ejecute un SO basado en Windows, por lo tanto y a diferencia de otras herramientas, no es necesario editar ningún fichero de configuración para establecer una configuración inicial, puesto que viene pre-configurado con los servicios más usuales.

La aplicación simula servicios de sistema en las capas altas de modelo de red OSI (capa de aplicación). Esto permite utilizar los mecanismos de seguridad establecidos por el SO y las librerías de red, reduciendo el riesgo de detección y compromiso de la honeypot gracias a la falta de drives adicionales y cambios sobre la pila IP. Tampoco deben realizarse cambios en los routers o los firewalls para adaptarlos a la máquina que haga las funciones de honeypot.

HoneyBOT

Solución de media interacción que permite la implementación de una honeypot en entornos Windows, de forma totalmente gratuita.

Permite la escucha en un gran número de puertos en los equipos, emulando en cada puerto un servicio vulnerable. Todas las comunicaciones son capturadas por la honeypot de forma segura y registra los resultados de la interacción para futuros análisis.

También almacenará todo el malware que el atacante intente introducir en el sistema, aislándolo de forma segura para que no afecte al sistema.

Entre el malware que la honeypot es capaz de capturar se encuentra: Dabber, Devil, Kuang, MyDoom, Netbus, Sasser, LSAAS, DCOMM, Lithium, Sub7…

Conpot

Se trata de una Honeypot que emula un Sistema de Control Industrial (ICS). A diferencia de otras honeypots dedicadas a emular este tipo de sistemas, conpot apenas requiere configuración adicional una vez instalada la máquina, simplemente la definición de aquellos servicios que se pondrán a la escucha, así como la red sobre la que se montarán y el puerto a la escucha. Esto hace a esta herramienta muy económica en tiempo y en esfuerzo.

Conpot implementa un servidor maestro que emulará un gran conjunto de protocolos de comunicación presentes en los ICSs usuales.

Los dos protocolos principales de control industrial soportados son Modbus y SNMP. Ambos se utilizan como estándares dentro de los ICS.

En su configuración básica, conpot emula un sistema Siemens SIMATIC S7-200 PLC básico, con un modelo de entrada salida y un CP 443-1, necesario en una configuración real para proporcionar conectividad de red.

En la página web de conpot se ofrecen ejemplos de configuración de los servicios por si se necesitara realizar algún cambio en la configuración básica.

Dionaea

Permite capturar malware que trate de explotar vulnerabilidades del sistema en el que se despliega la herramienta, siendo ésta su función principal.

Para evitar que se exploten bugs en el propio software de dionaea la herramienta permite reducir el número de privilegios, de forma que aquellas actividades que requieran privilegios elevados son realizadas por un proceso hijo creado previamente (aunque esta medida por sí misma no aporta seguridad, al menos complica la tarea de explotación por parte de los atacantes.

Dionaea no bloquea los servicios, por lo que pueden ser utilizados al mismo tiempo por otra herramienta. Uno de los mecanismos con los que se logra esto es establecimiento límites por conexión sobre tcp.

El protocolo de trabajo principal de dionaea es SMB. La elección de este protocolo se fundamenta en el número de bugs explotables conocidos y en el porcentaje de ataques sobre dicho protocolo por parte de los gusanos. Para implementar este servicio se utiliza python3 y scapi.

Dionaea también habilita servicios en otros puertos con mayor o menor sofisticación, como http por el puerto 80, un servidor ftp en el 21 y uno tftp en el 69. Un servidor MS SQL que se comunica con el protocolo TDS (Tabular Data Stream) y que escucha por el puerto 1433/tcp, permitiendo que los usuarios ingresen al sistema.

Debido a que dionaea está específicamente diseñado para capturar malware debe ser capaz de identificar la metodología de explotación que se está llevando a cabo, capturar y registrar los payloads de forma segura, para lo cual utiliza la librería libemu, que permite la ejecución de shellcodes en un entorno virtualizado para determinar su naturaleza.

Una vez detectado el payload, dionaea trata de descargarlo a través del protocolo ftp y tftp.

Glastopf

Glastopf es una honeypot de media interacción que emula una aplicación web dentro de la cual hay miles de vulnerabilidades explotables. Esta aplicación permite recopilar información sobre ataques cuyo objetivo sean aplicaciones web.

Lo que hace a glastopf una herramienta superior a otras similares es la capacidad de soportar ataques en múltiples fases, un emulador de vulnerabilidades y una lista de peticiones vulnerables.

La instalación y configuración es muy sencilla, sin la necesidad de una configuración extensa y compleja.

La aplicación funciona como un servidor web normal. Una petición enviada al servidor genera un procesamiento de la misma y una respuesta adecuada, por lo que una de las metas principales es conseguir responder al atacante de una forma lo más precisa posible. Una vez identificado el ataque lanzado contra el servidor, se genera una respuesta que haga creer al atacante que ha tenido éxito. Incluso si el atacante intenta un ataque de Remote File Inclusion, éste tendrá éxito, cargando el archivo malicioso en la aplicación web.

HERRAMIENTAS DE APOYO.

P0f

Herramienta de fingerprinting pasivo de SOs. Identifica los sistemas presentes en las máquinas que se conectan al equipo en el que se ejecuta glastopf, así como las máquinas a las que el equipo se conecta o aquellas que simplemente las que se encuentran dentro del segmento de red.

La información que se muestra con p0f es completa, detectando las conexiones mediante las cuales se accede al equipo, los saltos y su tiempo de actividad. También se pueden detectar puntos de acceso fraudulentos.

A diferencia de otras herramientas de fingerprinting, p0f realiza esta tarea de forma pasiva analizando el tráfico de red que llega la máquina en la que se instala, de modo que no se generan paquetes adicionales. Esto permite que p0f no sature la red con peticiones y que el equipo que se está analizando no sea consciente de dicho análisis ni de que se están capturando ciertos paquetes.

Wireshark

Software libre y multiplataforma que se distribuye bajo licencia GPL. Es un software muy robusto que se vale de la librería pcap para capturar paquetes. La lista de protocolos soportados por Wireshark comprende más de 480 tipos distintos.

Se trata de la evolución de Ethereal, un analizador de protocolos que permite llevar a cabo tareas de análisis y solución de problemas en redes de comunicaciones. Una de las grandes ventajas de Wireshark es que posee una interfaz gráfica (GUI) que gracias a su sistema de tres ventanas (registro de tramas, información por capas de la trama y contenido de la trama en hexadecimal) hace el análisis de paquetes una tarea mucho más sencilla. Otra gran utilidad de esta herramienta es la capacidad de aplicar reglas de filtrado sobre los paquetes y su integración con otras herramientas, como aircrack-ng y otros sistemas que requieran capturas de tráfico de red.

El análisis se puede llevar a cabo tanto en tiempo real como a través de un archivo perteneciente a una captura.

Existe la posibilidad de elegir la interfaz de captura así como si se realizará en modo promiscuo.

Capítulo IV: DISEÑO

DISEÑO GENERAL

Figura 5. Arquitectura del sistema.

- Instalaciones: equipamiento necesario para el funcionamiento de los equipos, como alimentación y conexión de red, así como un ambiente de humedad y temperatura controlado.
- Red: funcionalidad de enrutamiento hacia Internet.
- Almacenamiento: sistema de almacenamiento que contendrá el SO anfitrión y las máquinas huésped virtualizadas. Se necesita espacio suficiente para almacenar todas las máquinas.
- Cómputo: capacidad de procesamiento de datos que den soporte a todos los SO, tanto el real como los virtualizados. Se necesita la capacidad suficiente como para poder ejecutar todas las máquinas concurrentemente junto con el SO anfitrión.
- SO Anfitrión: esta capa se encargará de gestionar el hardware subyacente y proporcionar las funcionalidades necesarias a las capas superiores.
- Software de Virtualización: esta capa permite crear una asignación de recursos suficiente a cada máquina al mismo tiempo que las mantiene aisladas del resto. Por último esta capa se encargará de crear las interfaces de red necesarias para que el router posea visibilidad sobre todos los SOs virtualizados.
- SOs virtualizados: Software que darán soporte a las honeypots.
- Herramientas: estas herramientas se encargarán de montar todas las honeypots sobre los sistemas huéspedes.
- Honeypots: esta es la última capa y serán los sistemas visibles por los atacantes.

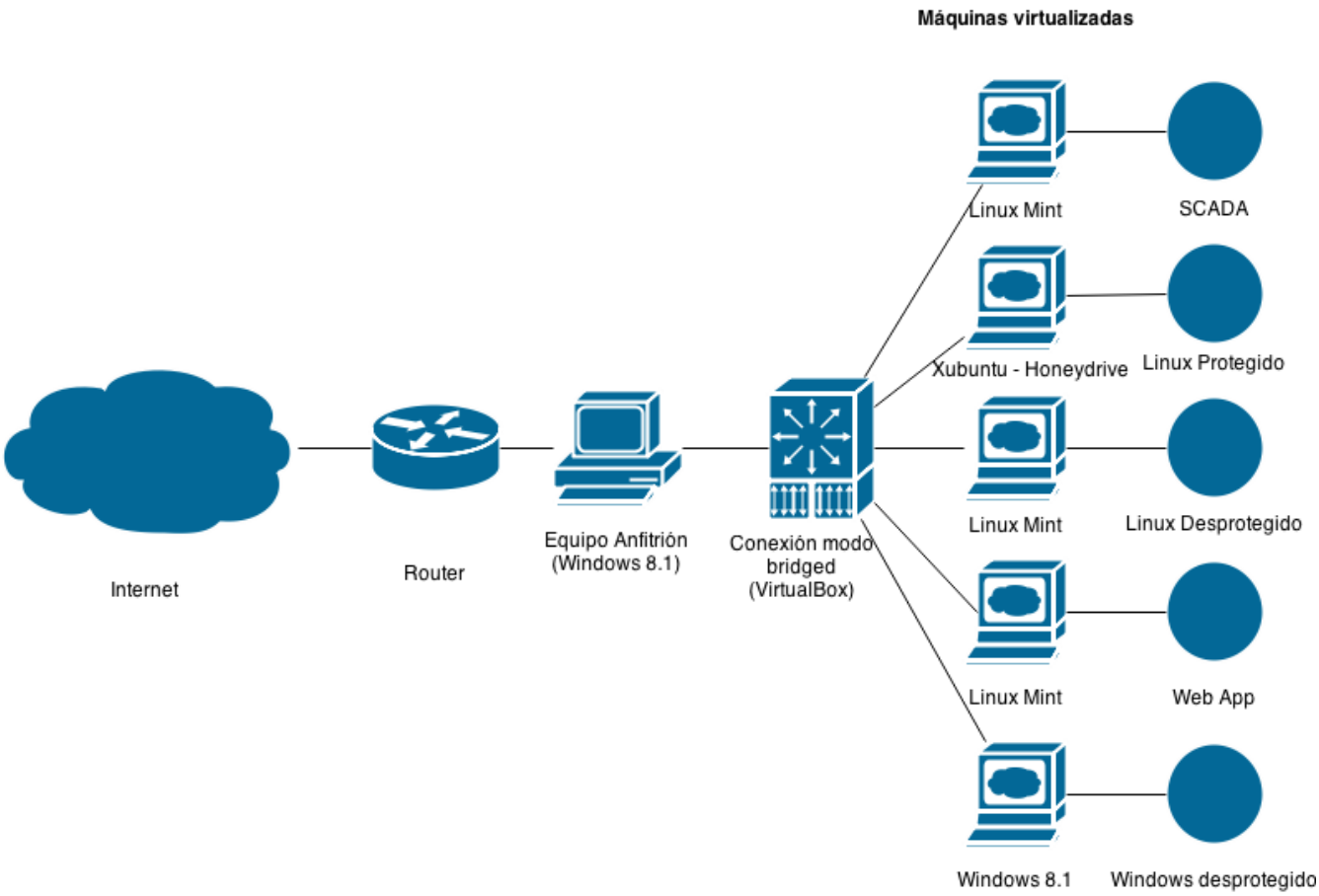
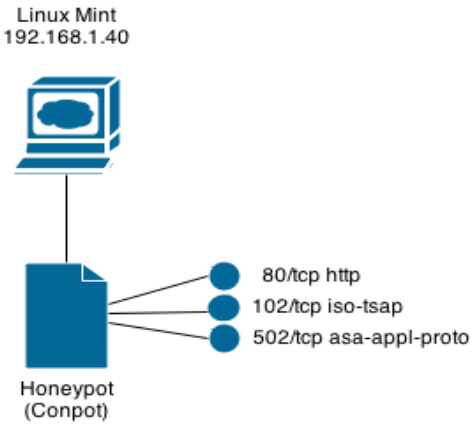


Figura 6. Arquitectura de red.

DISEÑOS ESPECÍFICOS

Máquina Linux - Sistema SCADA

Creará una honeypot que simule un sistema SCADA con los puertos de los servicios de control más usuales activos. Además se incluirá en el directorio de la honeypot una página html que aportará información al atacante una vez que acceda a su código fuente. Aunque en la captura el puerto 161 aparece cerrado lo cierto es que es perfectamente funcional y permite la recepción de ataques. Emulará un sistema Siemens SIMATIC S7-200 PLC básico

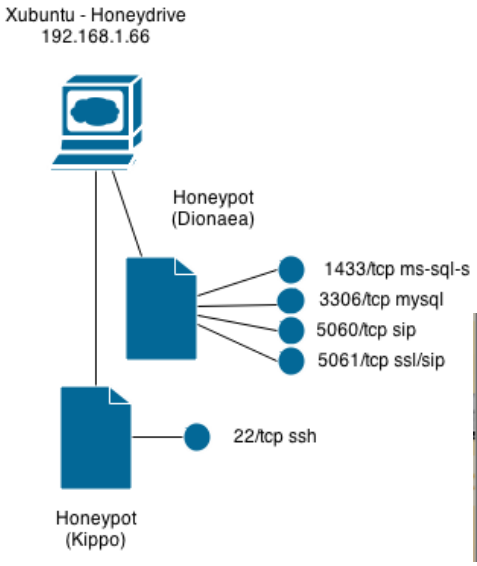



```
Starting Nmap 6.40 ( http://nmap.org )
Nmap scan report for 192.168.1.36
Host is up (0.000054s latency).
PORT    STATE  SERVICE
80/tcp  open   http
102/tcp open   iso-tsap
161/tcp closed snmp
502/tcp open   asa-appl-proto
```

Figura 7 y 8. Asignación de puertos para Conpot y comprobación de puertos con nmap.

Máquina Linux - Simulando máquina sometida a un proceso de hardening.

Simulará un sistema fuertemente protegido, con el puerto 22 habilitado para SSH. También se activará Dionaea para intentar capturar malware.

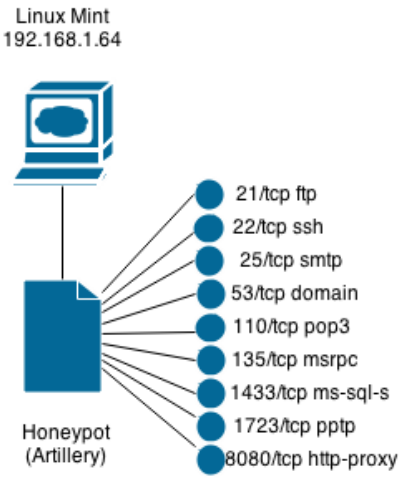



```
PORT     STATE SERVICE   VERSION
22/tcp   open  ssh       OpenSSH 5.1p1
80/tcp   open  http      Apache httpd 2
1433/tcp open  ms-sql-s?
3306/tcp open  mysql     MySQL 5.0.54
5060/tcp open  sip       (SIP end point
5061/tcp open  ssl/sip   (SIP end point
```

Figura 9 y 10. Asignación de puertos para Kippo y Dionaea, y comprobación de puertos con nmap.

Máquina Linux - Funcionamiento genérico con la mayoría de servicios habilitados.

Este sistema emula a una máquina Linux convencional, con un número mayor de puertos abiertos, sin ningún tipo de medida de protección aparente.

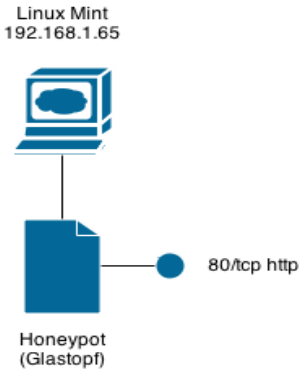



```
PORT      STATE SERVICE
21/tcp    open  ftp?
22/tcp    open  ssh?
25/tcp    open  smtp?
53/tcp    open  domain?
110/tcp   open  pop3?
135/tcp   open  msrpc?
139/tcp   open  netbios-ssn?
445/tcp   open  microsoft-ds?
1433/tcp  open  ms-sql-s?
1723/tcp  open  pptp?
3389/tcp  open  ms-wbt-server?
5800/tcp  open  vnc-http?
5900/tcp  open  vnc?
8080/tcp  open  http-proxy?
10000/tcp open  snet-sensor-mgmt?
44443/tcp open  coldfusion-auth?
```

Figura 11 y 12. Asignación de puertos para Artillery y comprobación de puertos con nmap.

Máquina Linux - Servidor Web Vulnerable.

Esta máquina mostrará un servidor web con una aplicación web vulnerable en su interior con la que los atacantes podrán interactuar. Aquí la información útil no provendrá de los ataques de la máquina sino a la aplicación web.




```
PORT    STATE SERVICE
80/tcp  open  http
139/tcp open  netbios-ssn
445/tcp open  netbios-ssn
```

Figura 13 y 14. Asignación de puertos para Glastopf y comprobación de puertos con nmap.

Máquina Windows - Servicios con mayor índice de ataques activados.

Sistema Windows con un gran número de puertos abiertos. En la práctica y por limitaciones del router se abrirán los puertos más atacados utilizando la información aportada en la web de HoneyBOT.

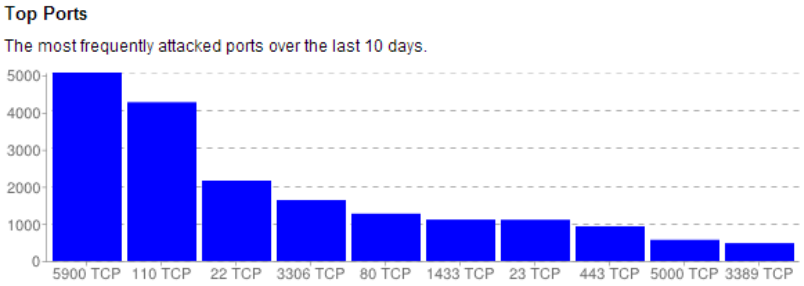


Figura 15. Tendencias de ataques/puerto en sistemas Windows.

Figura 16 y 17. Asignación de puertos para Conpot y comprobación de puertos con nmap.

ASIGNACIÓN DE SERVICIOS A PUERTOS

21/tcp – ftp

Utilizado para File Transfer Protocol (FTP) del lado del servidor. Se utiliza para establecer una sesión FTP, en la que el cliente establece una conexión al servidor FTP a la escucha en el puerto 21. El servidor FTP devuelve entonces un mensaje que solicita al cliente sus credenciales de acceso. Una vez que el servidor FTP autentica la conexión, la información se envía a través de una conexión secundaria por otro puerto.

22/tcp – ssh

Utilizado para conexiones a SSH (Secure Shell), con la finalidad de establecer un sistema seguro de login. En principio se ideó como una alternativa segura a Telnet, un protocolo con las mismas funcionalidades pero claras deficiencias de seguridad.

23/tcp – telnet

Telnet es uno de los protocolos más antiguos y uno de los más populares para acceder a máquinas Unix. El problema de este protocolo son sus diversas vulnerabilidades de seguridad.

25/tcp – smtp

Protocolo para Simple Mail Transfer, utilizado para intercambio de e-mails entre dispositivos. Generalmente se usa en servidores de correo y agentes de transferencia de correo para llevar a cabo tareas de envío y recepción de mensajes, mientras que las aplicaciones de correo a nivel de usuario solo utilizan este protocolo para mandar mensajes, utilizando POP3 para la recepción.

53/tcp – domain

Se utiliza para servicios de DNS.

80/tcp – http

El protocolo HTTP (Hypertext Transfer Protocol) es un protocolo de la capa de aplicación responsable de la comunicación de datos para Internet. Transfiere texto estructurado con hyperlinks entre los nodos. Se trata de un protocolo de petición-respuesta para modelos cliente-servidor.

102/tcp – iso-tsap (s7)

El Transport Service Access Point es utilizado por PLCs Siemens SIMATIC S7-1200. Este protocolo sirve para gestionar la HMI (Human-Machine Interface) que permite controlar ciertos elementos de los sistemas SCADA. Estos sistemas gestionan tareas automatizadas de forma segura, mediante el acceso a través de Internet.

110/tcp – POP3

Post Office Protocol versión 3 trabaja a nivel de aplicación y permite a los clientes locales obtener mensajes de correo electrónico almacenados en un servidor de correo. Su funcionamiento es simple, accede al servidor remoto, descarga los correos del cliente en la máquina desde la que se accede y se desconecta. Es especialmente útil cuando se trabaja con conexiones de disponibilidad inestable.

135/tcp – msrpc

El puerto para RPC (Remote Procedure Call) de Microsoft permite la comunicación de aplicaciones cliente/servidor y suele utilizarse para aquellos clientes que debas acceder mediante VCN de forma remota.

443/tcp – https

Permite la comunicación de tráfico web encriptado mediante SSL/TLS.

502/tcp – asa-appl-proto

Es un protocolo de comunicaciones que se utiliza como controlador lógico programable (PLC). Es el puerto estándar para acceder a dispositivos de control industrial, permitiendo la comunicación entre dispositivos dentro de la misma red. En este esquema el puerto 502 entregaría información de una unidad remota (RTU) a un controlador maestro que supervisa el control y la obtención de datos de los sistemas SCADA.

1433/tcp – ms-sql-s

Este puerto permite a MS SQL Server comunicarse a través de un servidor de seguridad.

1443/tcp – ies-lm

Se utiliza por el Integrated Engineering Software, herramientas de simulación para análisis de diseños de diversa índole.

1723/tcp – pptp

El protocolo PPTP (Point to Point Tunneling Protocol) permite la implementación de redes privadas virtuales (VPN). PPTP se encarga del intercambio seguro de datos en un modelo cliente-servidor que posteriormente desembocarán en la creación de VPNs.

3306/tcp – mysql

Permite la conexión a servidores de bases de datos MySQL a través de Internet.

5060/tcp – sip

SIP (Session Initiation Protocol) es un protocolo de señalización de comunicaciones, utilizado ampliamente para sesiones de comunicación multimedia de uno o varios flujos.

5061/tcp – ssl/sip

Está asignado para encriptar las comunicaciones SIP.

5900/tcp – vnc

VNC (Virtual Network Computing) permite controlar aplicaciones de escritorio de forma remota mediante una interfaz gráfica. Transmite los eventos de teclado y ratón de un ordenador a otro, mandando la señal de la pantalla en la otra dirección. El protocolo funciona bajo cualquier plataforma. Además, permite una conexión múltiple y la descarga y acceso de archivos entre ordenadores.

8080/tcp – http-proxy

Funciona como cualquier otro proxy, solo que en este caso se encuentra alojado en la máquina. Este puerto también se utilizará en determinadas máquinas como puerto HTTP cuando haya varias máquinas con servicio HTTP activo.

Capítulo V: CONFIGURACIÓN

CONFIGURACIÓN DEL ENTORNO

Preparar el entorno operativo

En esta primera fase me ocuparé de iniciar la construcción de los sistemas desde una instalación limpia del SO Windows 8. Una vez eliminados los datos previos mediante una operación de formateo se procede con la instalación del sistema.

Antes de empezar a instalar ningún tipo de software es necesario buscar todas las actualizaciones con ayuda de Windows Update. Llegados a un punto en las actualizaciones el sistema nos informará de que está disponible la versión de Windows 8.1. Procedemos con la descarga y la posterior instalación.

Una vez instalada la versión 8.1 del SO se vuelven a buscar todas las actualizaciones posibles.

Llegados a este punto el SO ya se encuentra en su última versión disponible, por lo que el número de vulnerabilidades queda reducido al mínimo posible.

Como medida preventiva se crea un usuario sin permisos de administrador que será con el que se operará todo el sistema. Tanto la contraseña de este usuario como la del usuario del administrador se elegirán de forma suficientemente robusta, distintas entre sí, y se cambiarán cada 15 días.

Securizar el entorno operativo

En esta fase me ocuparé de securizar la máquina anfitrión para evitar que un ataque a las MVs pudiera desembocar en la amenaza del sistema anfitrión.

El primer paso sería instalar una solución de antivirus. Me he decantado por BitDefender Total Security, por ser una solución bastante completa, que además de antivirus incluye un firewall, sistemas de anti spam y otras funcionalidades interesantes. El software ha sido adquirido de forma legal, de modo que hay disponible una mayor garantía sobre el mismo y existe la posibilidad de contactar con el servicio de asistencia en cualquier momento.

Además de este anti-malware se instala en el sistema un segundo anti-virus: MalwareBytes Anti-Malware, lo que reducirá considerablemente la tasa de falsos negativos, lo que en este caso es preferible aunque se aumenten los falsos positivos.

Estos dos anti-malware se encontrarán en todo momento actualizados a su última versión y se utilizarán para realizar escaneos completos de todo el sistema cada 4 días.

Como software de protección adicional también se instala EMET (Enhaced Mitigation Experience Toolkit), una utilidad de Microsoft que evita la explotación de vulnerabilidades de seguridad, mediante el uso de tecnologías de mitigación. Aunque no evita que las vulnerabilidades sean explotadas, al menos dificulta su explotación al máximo. Una vez instalado se incluirá todo el software instalado en el equipo dentro de la configuración de EMET, para que todos los programas queden protegidos. Además, se establecerá el nivel de seguridad al más alto, para que notifique siempre tras cualquier evento.

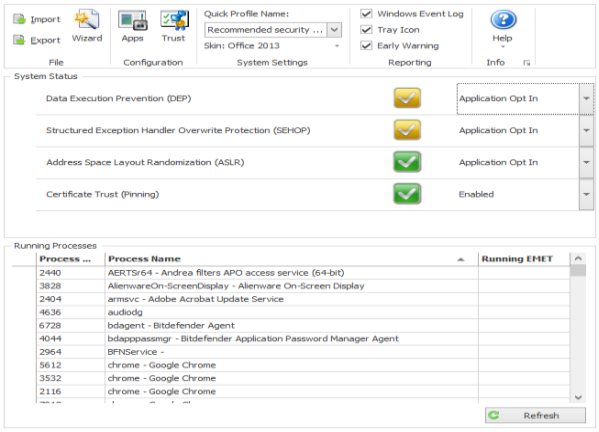


Figura 18. EMET una vez instalado y configurado.

Para garantizar la seguridad de los archivos importantes se instalará el software de backup en la nube Cypherite, y se creará una carpeta con todos los documentos importantes en ella.

Se instalará el software Secunia PSI, que mantendrá todas las aplicaciones actualizadas a su última versión de forma permanente.

Una medida de seguridad efectiva y poco conocida es la imposición de la obligación de utilizar la combinación de teclas Ctrl+Alt+Supr para cambiar entre usuarios. Esto se realiza buscando netplwiz en el cuadro de búsqueda de Windows 8.1 y habilitando “Requiere que los usuarios presionen Ctrl+Alt+Supr”. Aunque esta medida no es infalible sí que evita la escalada de privilegios bajo ciertas condiciones.

Se desactivará la conectividad Bluetooth en el panel de control. Aunque el alcance de la antena Bluetooth es muy reducido se trata de un servicio que no se utilizará y por tanto no tiene sentido mantenerlo activado.

Tras configurar las actualizaciones automáticamente en Windows Update (Panel de Control) e implementar el cifrado de disco con Bitlocker ya solo nos queda instalar un HIDS en el equipo para que registre todas las conexiones que se produzcan y determine aquellas con una finalidad maliciosa.

El HIDS escogido es PatriotNG, creado por uno de los colaboradores de SecurityByDefault. Adicionalmente tendremos que instalar winpcap para que este software pueda funcionar correctamente.

Preparación del entorno de MVs

La opción de virtualización que se utilizará en este proyecto es Oracle VM VirtualBox, por ser una herramienta gratuita y por suponer una alternativa fácil de usar y configurar, tanto a nivel de sistema como sus características de red. Una vez instalado el software y escogido un directorio para almacenar las máquinas virtuales pasamos al siguiente paso.

Instalación de las MVs

Para el despliegue de las honeypots utilizaremos cinco máquinas virtuales.

Primero instalamos el SO Windows 8 en una de las máquinas, con conexión en modo puente, 4096MB de RAM y 30GB de disco duro.

A continuación instalaremos tres máquinas virtuales idénticas con una distribución Linux Mint de 64 Bits en su última versión. A cada máquina le asignaremos 2048MB de RAM y 20 GB de disco duro, todas ellas con una conexión de red en modo puente.

Por último importamos HoneyDrive en VirtualBox.

Todas estas máquinas contarán con una contraseña robusta.

Preparación del router

Nada más acceder al router lo primero será cambiar su contraseña de administración por defecto y se le asignará una con un nivel elevado de robustez.

Para que las máquinas posean visibilidad desde internet y ya que no existe la posibilidad de asignar una IP pública a cada máquina se accederá al router por la puerta de enlace y se realizará una asignación de puertos por IP, de forma que cada IP (que se ha establecido como fija en la configuración de las máquinas) tendrá los puertos abiertos necesarios para que las honeypots sean visibles desde Internet.

El firmware del router no permite especificar más de 9 puertos por máquina, por lo que en el caso de las dos honeypots de emulación de sistemas genéricos (Windows y Linux) se escogerá un subconjunto de los puertos más representativos y con un mayor índice de ataques.

Tras configurar los puertos para las IPs de las honeypots que vayan a estar activas en cada momento se guarda el estado y se sale de la configuración del router (IMPORTANTE: puede haber varias honeypots funcionando al mismo tiempo, pero los puertos utilizados no pueden solaparse).

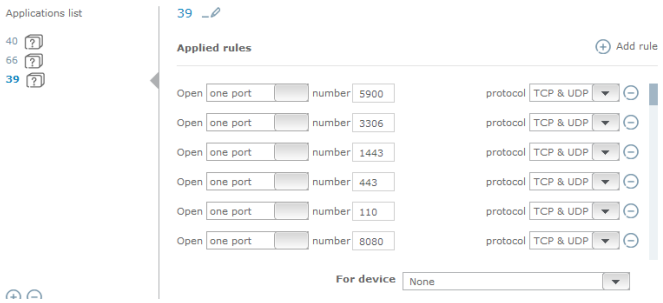


Figura 19. Interfaz de redirección de puertos del router.

CONFIGURACIÓN DE LAS HERRAMIENTAS

Herramienta para Windows.

KFSensor:

- Abrimos la MV Windows.
- Descargamos KFSensor de: http://www.keyfocus.net/kfsensor/download/

  - Nos pide un registro para poder descargar, por lo que introducimos un nombre y una dirección de correo.
- La instalación es simple, basta con ir pulsando siguiente durante todo el proceso.
- Una vez instalada abrimos la aplicación.
- Al pulsar sobre "Start Server" en la parte superior izquierda arranca la Honeypot.

HoneyBOT honeynet:

- Descargar de la página oficial.
- Doble click sobre el instalador de HoneyBOT.
- Lanzar el software HoneyBOT desde el icono del escritorio una vez que haya sido instalado.
- Para iniciar la honeybot pulsar sobre el botón "Play" en la parte superior.
- Seleccionaremos en "Options" la opción "Capture Binaries" para almacenar todo el malware que se cargue dentro del sistema.

Herramienta para Linux (Honeydrive)

- Descargamos honeydrive de la página web oficial.
- Importamos una nueva máquina virtual en VirtualBox con el archivo .OVA descargado.
- Abrimos la distribución y nos logeamos: honeydrive/honeydrive.
- Kippo:
  - Ir al directorio de kippo en HoneyDrive: cd /opt/kippo
  - Ejecutar la aplicación: ./start.sh
  - Para consultar los logs ir a: localhost/kippo-graph/index.php
- Dionaea:
  - Para arrancar dionaea:
    - Nos movemos al directorio: cd /opt/dionaea/bin
    - Ejecutamos dionaea: sudo ./dionaea –l all
- Para ver los resultados:
  - Nos movemos a dionaeaFR: cd /opt/dionaeaFR
  - Ejecutamos los scripts de inicio:
    - Sudo python manage.py collecstatic
    - Sudo Python manage.py runserver 192.168.1.100:8000
  - Accedemos a la dirección y comprobamos los resultados.

Herramientas para Linux

Artillery:

  - Necesitamos instalar git para poder descargar artillery.
    - Sudo apt-get update && apt-get install git
  - Descargamos artillery (el directorio de descarga es trivial, ya que la instalación se realizará en /var/artillery):
    - Sudo git clone https://github.com/trustedsec/artillery/ artillery/
  - Nos movemos al directorio que acabamos de descargar y lo instalamos:
    - Cd /artillery
    - Sudo ./setup.py install
  - Ahora debemos configurar artillery:

- Sudo gedit /var/artillery/config
- Se nos abrirá un archivo en el que tendremos que ir configurando lo siguiente:
  - Ficheros a monitorizar:
    - MONITOR_FOLDERS="/var/www","/etc/","/root/", "/bin","/usr"
  - Hosts excluidos de análisis:
    - WHITELIST_IP=127.0.0.1,localhost,192.168.1.0/24
  - Puertos sobre los que se montarán servicios simulados:
    - PORTS="135,445,22,1433,3389,8080,21,5900,25,53,110,1723,1337,10000,5800,44443"
- El resto de opciones las dejamos por defecto (salvo los intentos de fuerza bruta sobre servicios, dejándolos todos activados).

- En este punto Artillery ya está listo para funcionar por lo que ejecutamos:
  - Sudo service artillery start

Glastopf:

- Instalamos las dependencias:
  - Sudo apt-get update
  - Sudo apt-get install python2.7 python-openssl Python-gevent libevent-dev python2.7-dev build-essential make
  - Sudo apt-get install Python-chardet Python-request Python-sqlalchemy Python-lxml
  - Sudo apt-get install Python-beautifulsoup mongodb Python-pip Python-dev Python-setuptools
  - Sudo apt-get install g++ git php5 php5-dev liblapack-dev gfortran libmysqlclient-dev
  - Sudo apt-get install libxml2-dev libxslt-dev
  - Sudo pip install –update distribute
- Instalamos y configuramos el sandbox seguro PHP
  - Cd /opt
  - Sudo git clone git://github.com/glastopf/BFR.git
  - Cd BFR
  - Sudo phpize
  - Sudo ./configure –enable-bfr
  - Sudo make && sudo make install
  - Abrimos el archivo php.ini y añadimos la siguiente línea:
    - Zend_extension = /usr/lib/php5/20121212/bfr.so
- Instalamos glastopf:
  - Sudo pip install glastopf
  - Cd /opt
  - Sudo git clone https://github.com/glastopf/glastopf.git
  - Cd glastopf
  - Sudo Python ./setup.py install
- Arrancamos la honeypot.
  - Sudo apachectl –k stop (esto para Apache y elimina el servicio a la escucha por el puerto 80 que posteriormente usará glastopf).
  - Sudo glastopf-runner

Herramienta para sistema SCADA

Conpot

- Instalamos las dependencias de conpot:
  - Sudo apt-get install git libsmi2ldbl smistrip libxslt1-dev Python-dev libevent-dev Python-pip
- Nos movemos al directorio /opt:
  - Cd /opt
- Descargamos conpot
  - Sudo git clone https://github.com/glastopf/conpot
- Instalamos los requerimientos:
  - Pip install –r requirements.txt
- Aunque aquí debería haberse instalado conpot sin problemas debemos solucionar dos problemas.
  - Una de las librerías que utiliza conpot, modbut-tk es necesaria para ejecutar el servicio [modbus], pero en su versión estándar genera un problema debido a la falta de un atributo, por lo que debemos sustituir dicha librería por la versión de glastopf
    - Nos movemos al directorio donde está modbus-tk instalado:
      - Cd /usr/local/lib/python2.7/dist-packages/Conpot-0.2.2-py2.7.egg/
    - Borramos el original:
      - Sudo rm –dfr Modbus_tk
    - Descargamos la versión de glastopf:
      - Sudo git clone https://github.com/glastopf/modbus-tk /modbus_tk
    - Nos movemos al directorio:
      - Cd /modbus_tk
    - Instalamos el paquete:
      - Sudo Python ./setup.py install
  - Una vez hecho esto debemos configurar conpot a nuestro gusto
    - Nos movemos al directorio de conpot
      - Cd /opt/conpot
    - Abrimos el archivo de configuración:
      - Sudo gedit /conpot/conpot.cfg
      - Para activar los servicios modbus, s7, http y snmp debemos introducir las siguientes líneas (es importante introducirlo a mano y no copiarlo para no introducir caracteres no soportados por el parser):

```
[modbus]
Enabled = True
Host = 0.0.0.0 (esto creará el servicio en nuestra máquina, como si indicásemos localhost como dirección).
Port = 502
[s7]
Enabled = True
Host = 0.0.0.0
Port = 102
[snmp]
Enabled = True
Host = 0.0.0.0
Port = 161
[http]
Enabled = True
```

Host = 0.0.0.0
Port = 80

- Estamos listos para arrancar conpot. Primero hay que ir al directorio apropiado:
  - Cd ../bin
- Ejecutamos la honeypot:
  - Sudo conpot

CONFIGURACIÓN DE LAS HERRAMIENTAS DE APOYO.

p0f

- Para instalar p0f basta con ejecutar el siguiente comando:
  - Sudo apt-get install p0f.
- Una vez instalado y antes de iniciar cada honeypot se ejecutará p0f en la máquina huésped correspondiente con el siguiente comando.
  - Sudo p0f –vt –i eth0 > /home/<usuario>/Desktop/p0frepor_<fecha>
    - La opción –vt permitirá que cada entrada cuente con una marca de tiempo y la salida sea más verbosa, con lo cual obtendremos más información sobre el atacante.

Wireshark

- Wireshark puede ser descargado desde: http://www.wireshark.org/download.html
- Puesto que será utilizado en la máquina anfitrión para monitorizar toda la actividad de las MVs en tiempo real bastará con elegir la versión para Windows y seguir los pasos (es necesario tener la librería winpcap instalada previamente).
- Para arrancarla hacemos clic sobre el icono que se habrá generado, seleccionamos la interfaz que se desea escuchar, en mi caso la interfaz virtual creada por VirtualBox y ya solo queda pulsar sobre el botón “Start” para empezar a esnifar paquetes.

Capítulo VI: DATOS CAPTURADOS.

MECANISMOS DE OBTENCIÓN DE INFORMACIÓN.

Una vez que las honeypots se despliegan y empiezan a emular servicios es importante poder procesar los ataques que reciben.

Las herramientas Kippo, Dionaea y Glastopf ofrecen información útil sobre los ataques, como los comandos ejecutados, el malware cargado y las interacciones con la aplicación web respectivamente.

Pero aun así todavía habría gran cantidad de información que pasaría desapercibida, y es aquí donde entra en juego la herramienta p0f, con ella no solo sabemos el instante exacto en el que se producen los ataques sino además información relevante como el SO utilizado, el tiempo que el host atacante está activo, información sobre la conexión que utiliza o las redes que hay entre el atacante y la honeypot.

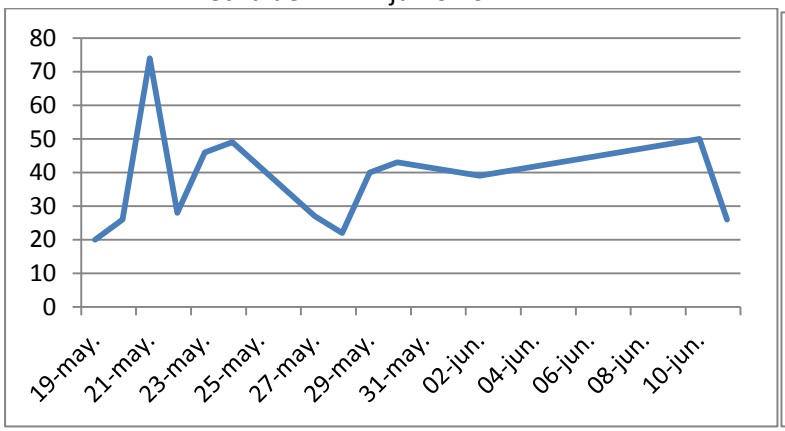
Una vez que tenemos toda esta información ya solo falta analizar la localización del origen de los ataques, para lo cual utilizamos una herramienta en línea de whois sobre IP: http://www.ip-adress.com/whois/
Con esto ya tenemos la capacidad de analizar toda la información relevante que puede extraerse de los ataques a las honeypots.

RESUMEN DE LAS HONEYPOTS.

Honeypot SCADA

Conpot:

Ataques totales: 490
Promedio de ataques/día: 40.83
IPs distintas:230
Periodo de actividad:
Fecha de inicio: 19 mayo 2014
Fecha de fin: 11 junio 2014



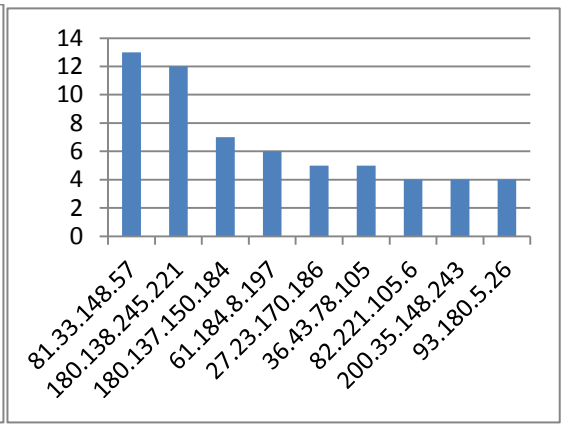


Figura 19 y 20. Actividad por días y Top 10 IPs.

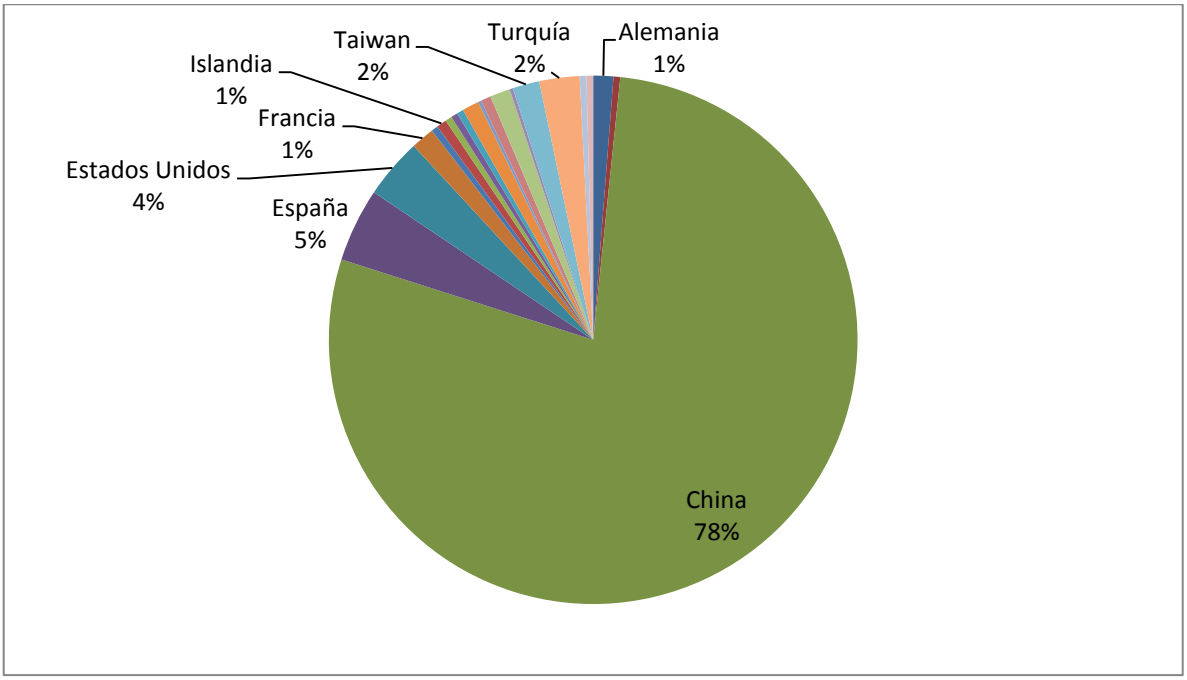
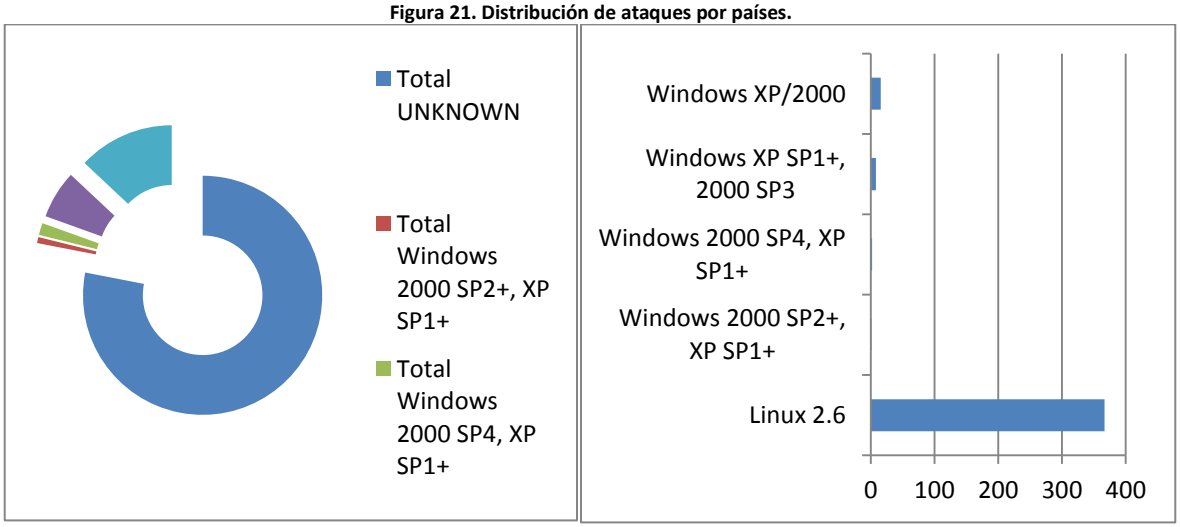


Figura 21. Distribución de ataques por países.



Figuras 22 y 23. Distribución de ataques por Sistemas Operativos.

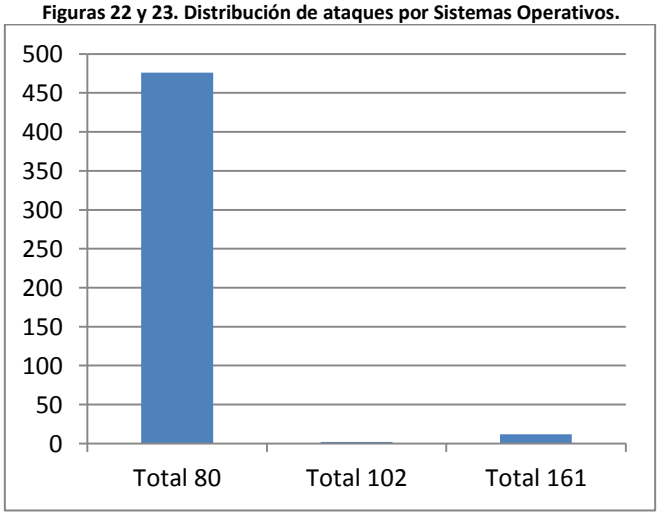


Figura 24. Distribución de ataques por puerto.

Observaciones:

La actividad en esta Honeypot se mantiene hasta cierto punto uniforme a lo largo del tiempo, produciéndose un número de ataques más o menos similar en su periodo de actividad salvo en los primeros días, en los que se observan picos de actividad. Por detrás de la honeypot creada con Kippo es la que más ataques recibió, con un total de 490.

Así mismo las IPs asociadas a los ataques también suelen distribuirse en un rango uniforme de ataques, destacando por encima de la media dos IPs con un número más elevado de ataques aunque tampoco se forma especialmente destacable.

El país de origen que predomina en los ataques es China, especialmente la región de Nanjing, en Jiangsu. Seguida muy de lejos por EEUU y España. Lo curioso del caso de España es que los ataques se producen de una localización geográfica relativamente próxima a la de la Honeypot, por ese mismo motivo se comprobó que no se hubiera producido un falso positivo con la propia máquina honeypot de alguna manera, tras lo cual se observó que aunque proceden de una localización próxima ni siquiera pertenecen al mismo núcleo urbano.

La mayoría de estos ataques fueron lanzados por máquinas con sistema operativo Linux 2.6. la mayoría de ellos de las máquinas localizadas en Nanjing. La predilección por máquinas con SO Windows para la realización de ataques a este tipo de sistemas es muy escasa.

En relación a los puertos de ataque la mayoría se limitan a explorar el puerto 80, en el que se mostraba una página cuyo código se asemejaba al de un sistema SCADA por defecto, mostrando una serie de elementos de configuración. Accediendo al código fuente podían verse los puertos que se utilizaban para el sistema S7 y el Modbus, sin embargo el siguiente puerto más accedido es el encargado del SNMP, teniendo el puerto destinado a S7 un solo acceso y ninguno el dedicado al Modbus.

En relación a esta honeypot cabe destacar un problema bastante importante y es que suelen ser habituales los fallos a la hora de establecer comunicaciones con los puertos abiertos, especialmente el de SNMP. Debido a la falta de documentación al respecto es complicado determinar la causa del error y puesto que igualmente se indica la IP asociada a la conexión en la que se produjo el error dichos accesos se contabilizaros aunque produjeran un error en la honeypot.

Honeypot Linux sistema seguro.

Kippo:

Ataques totales: 6720
Promedio de ataques/día: 480
IPs de origen distintas: 68
Periodo de actividad:
Fecha de inicio: 7 Mayo 2014
Fecha de fin: 11 Junio 2014

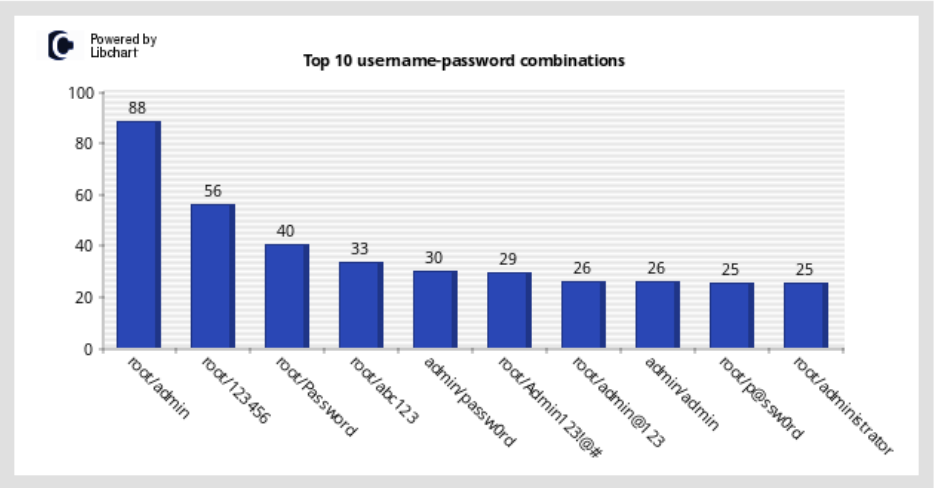


Figura 25. Top 10 combinaciones de Usuario/Contraseña.

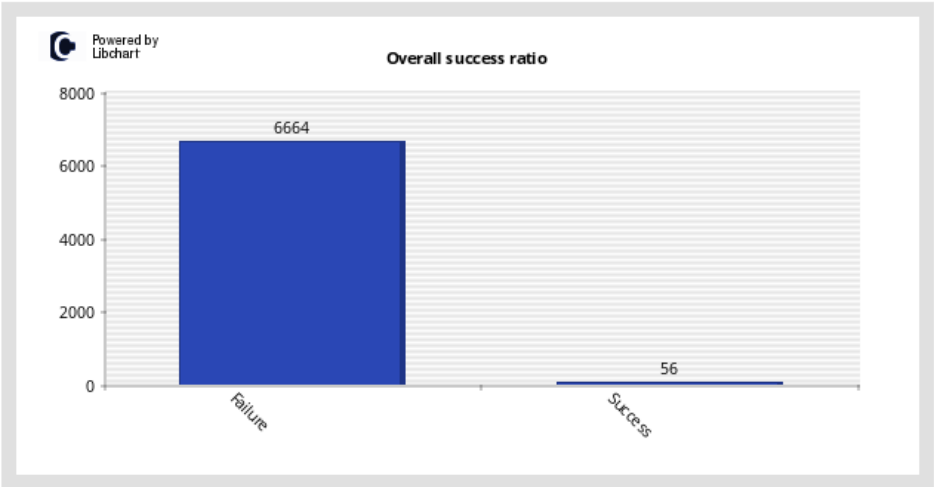
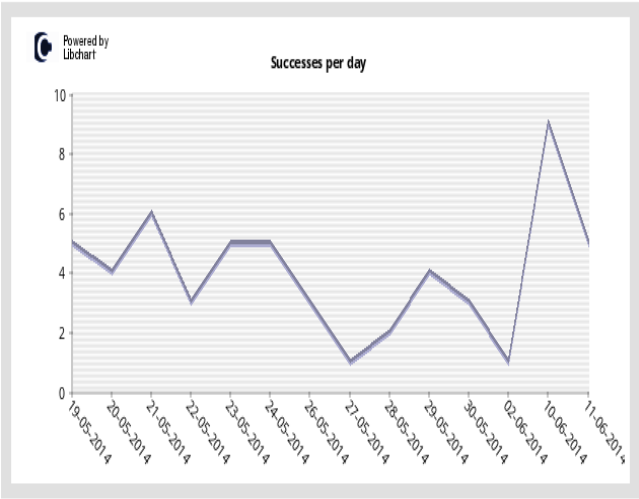
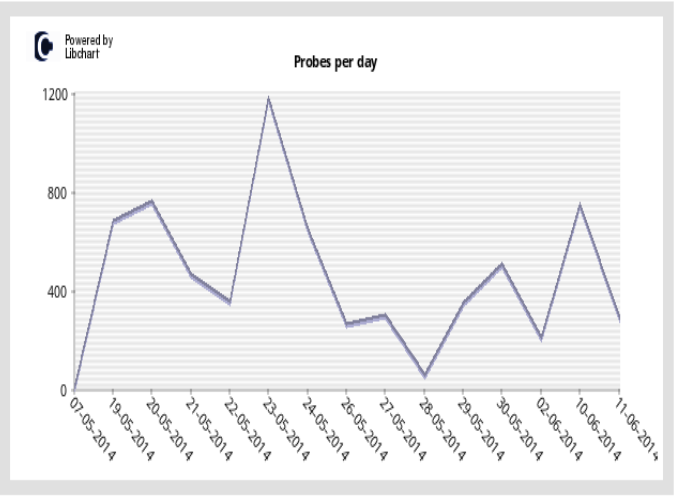


Figura 26. Ratio de éxito.





Figuras 27 y 28. Éxitos y pruebas totales por día.

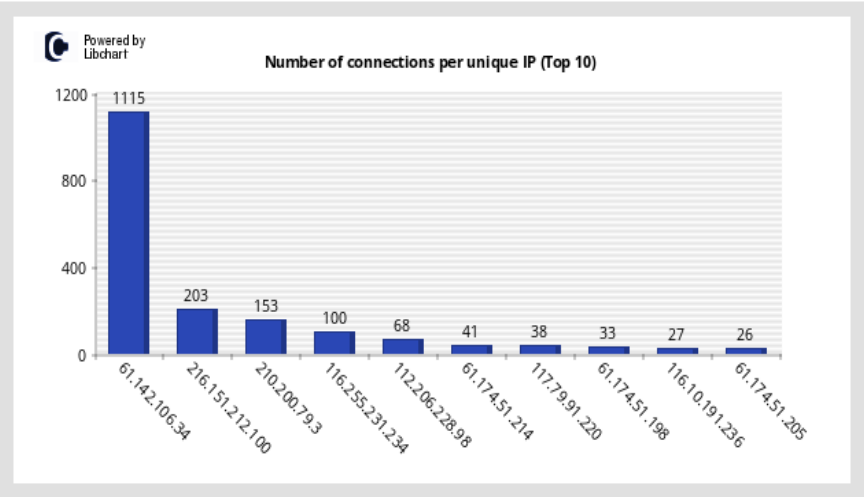


Figura 29. Conexiones por IP.

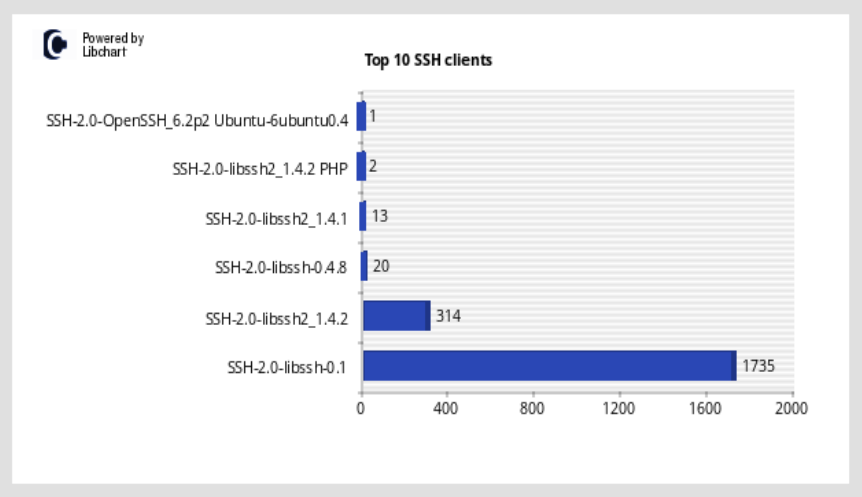


Figura 30. Clientes SSH más utilizados.

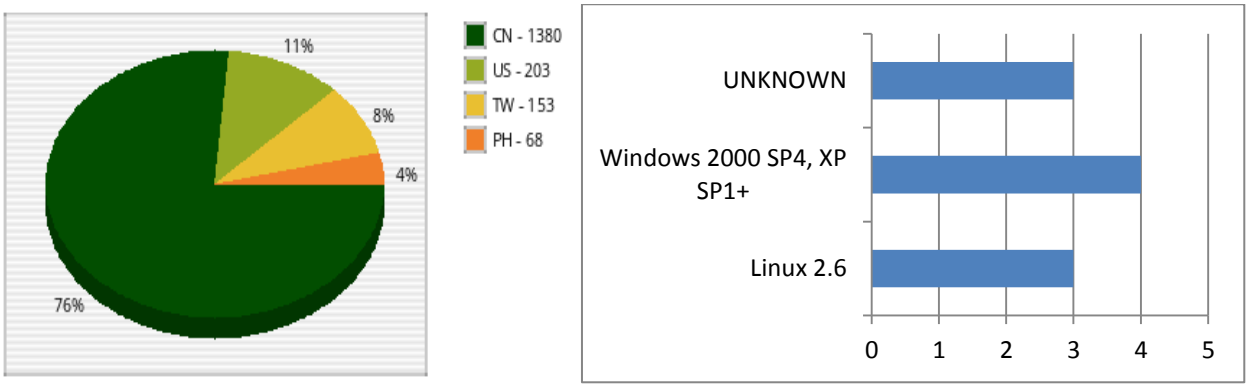
Actividad post-comprometimiento:

Comandos ejecutados: 0

Archivos descargados: 0

ID	IP Address	Probes	City	Region	Country Name	Code	Latitude	Longitude	Hostname
1	61.142.106.34	1115	Guangzhou	Guangdong	China	CN	23.116699	113.25	61.142.106.34
2	216.151.212.100	203	San Antonio	TX	United States	US	29.488899	-98.398697	216.151.212.100
3	210.200.79.3	153	Taipei	T'ai-pei	Taiwan	TW	25.0392	121.525002	210.200.79.3
4	116.255.231.234	100	Henan	Zhejiang	China	CN	27.8456	120.492798	116.255.231.234
5	112.206.228.98	68			Philippines	PH	13	122	112.206.228.98.pldt.net
6	61.174.51.214	41	Huzhou	Zhejiang	China	CN	30.8703	120.0933	214.51.174.61.dial.wz.zj.dynamic.163data.com.cn
7	117.79.91.220	38	Beijing	Beijing	China	CN	39.928902	116.388298	117.79.91.220
8	61.174.51.198	33	Huzhou	Zhejiang	China	CN	30.8703	120.0933	198.51.174.61.dial.wz.zj.dynamic.163data.com.cn
9	116.10.191.236	27	Nanning	Guangxi	China	CN	22.8167	108.316704	116.10.191.236
10	61.174.51.205	26	Huzhou	Zhejiang	China	CN	30.8703	120.0933	205.51.174.61.dial.wz.zj.dynamic.163data.com.cn

Figura 31. Detalle de las IPs origen de más ataques.



Figuras 32 y 33. Ataques por país y por sistemas Operativos

Dionaea:
-No se ha capturado malware-

Observaciones:

Esta máquina ha sido quizás la más sencilla de poner en funcionamiento y por tanto hay un mayor periodo de datos capturados. Igualmente, el tiempo adicional funcionando no justifica el número de ataques con respecto a las demás. Con mucha diferencia es la máquina más atacada, con 6720 ataques y 56 accesos exitosos. Además, a diferencia del resto de honeypots, en ésta se aprecian muchos más intentos de una misma IP, lo cual quizás sea revelador a la hora de elegir entre niveles de interacción para las honeypots desplegadas.
Quizás el hecho más sorprendente sea que a pesar de haber recibido tantos ataques con un buen número de éxitos, no se ha producido ninguna interacción con el sistema. El registro de Dionaea no muestra ninguna infección por malware, del mismo modo que Kippo no registró ningún comando ejecutado en el sistema ni ningún archivo descargado.

A la hora de acceder al sistema la mayoría de pruebas se limitaban a las contraseñas y usuarios por defecto como admin, 12345 o root desde sistemas tantos Linux como Windows en una proporción muy similar.

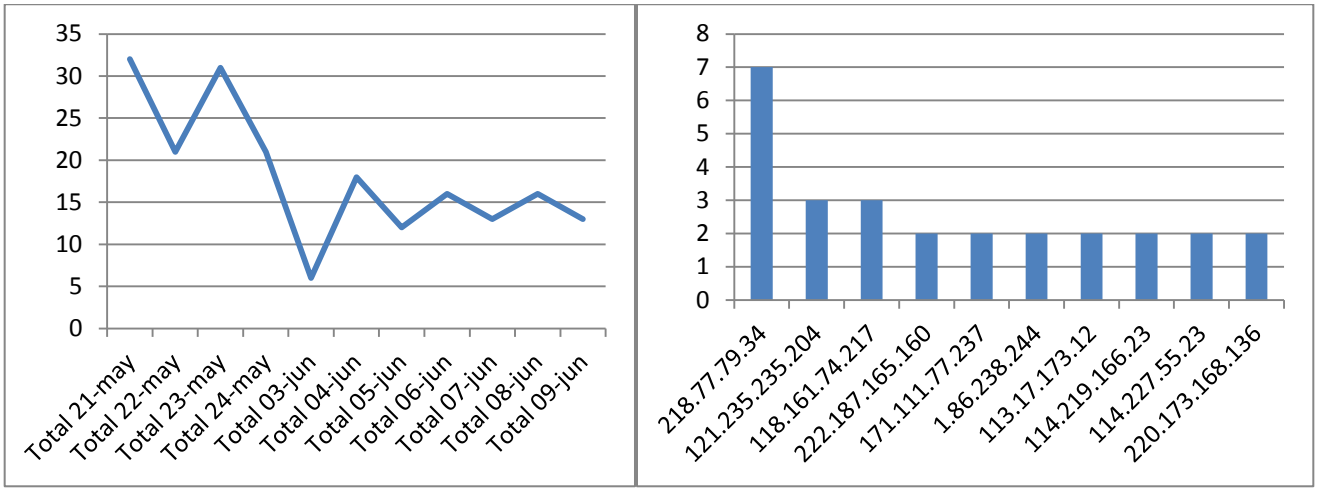
Una vez más las máquinas más utilizadas para lanzar los ataques se basan en sistemas Linux 2.6, y el país desde el que más ataques se producen es China, con un 76%, seguida de EEUU con un 11%.

En cuanto la forma de acceder al servicio SSH el software preferido es SSH-2.0_libssh-0.1 y SSH-2.0_libssh2_1.4.2, siendo predominante el primero.

Honeypot Linux sistema inseguro.

Artillery:

Ataques totales: 199
Promedio de ataques/día: 18.09
IPs distintas: 177
Periodo de actividad:
Fecha de inicio: 21 mayo 2014
Fecha de fin: 9 junio 2014



Figuras 34 y 35. Actividad por días y Top 10 de IPs.

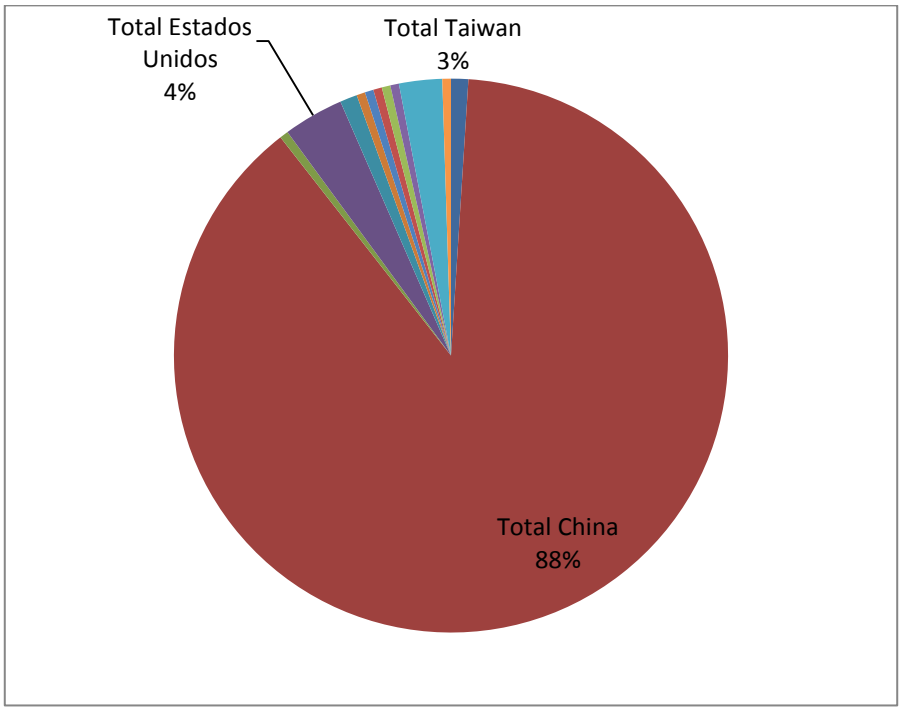
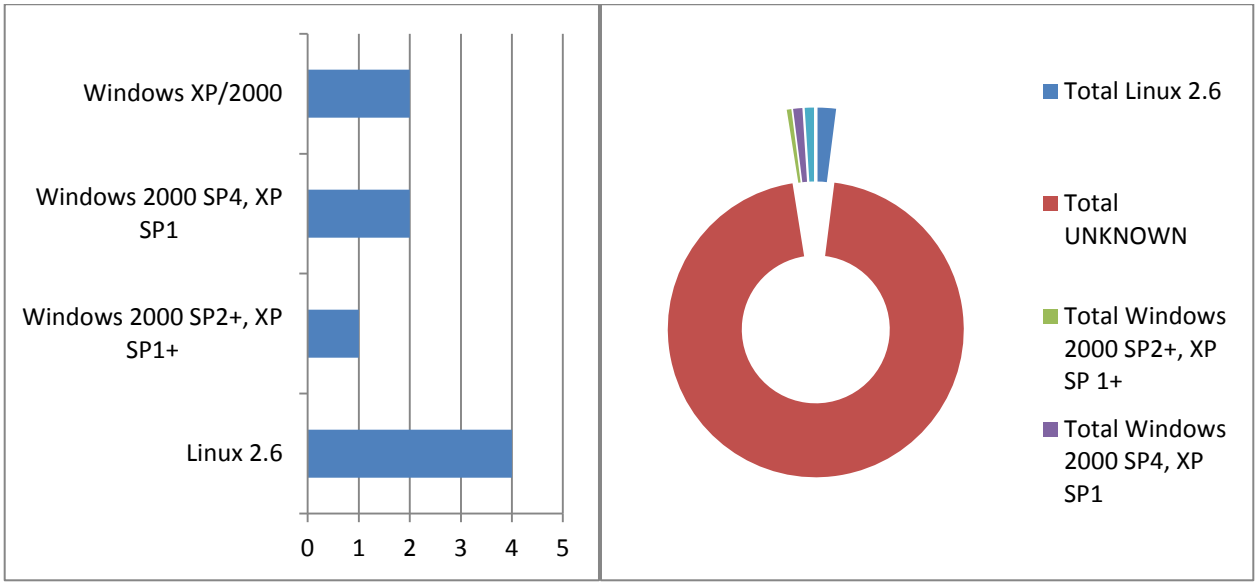


Figura 36. Distribución de ataques por país.



Figuras 37 y 38. Distribución de ataques por Sistemas Operativos.

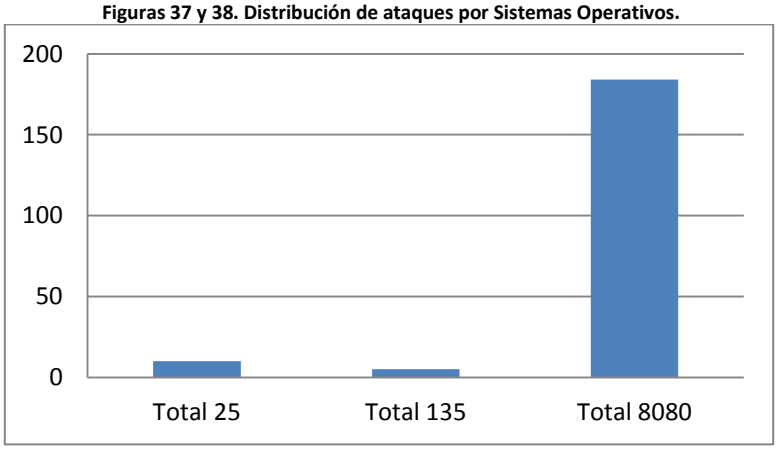


Figura 39. Distribución de ataques por puerto.

Observaciones:

En este caso el número de ataques decrece con respecto a los primeros días y consigue estabilizarse posteriormente sin llegar a producirse un número de ataques considerables.

En cuanto a las IPs atacantes no hay un despunte considerable de ninguna de ellas, aunque la IP 218.77.79.34 realizó más del doble de ataques que la siguiente IP con más ataques lo cierto es que este número es muy bajo si lo comparamos con los realizados contra Kippo o Conpot. El número de ataques promedio se mantiene muy bajo con 1 o 2 en su mayoría, en esta honeypot en particular este hecho es debido es que una vez que se realiza un ataque sobre un puerto éste devuelve información basura al atacante, y por tanto no motiva la realización de más interacciones por parte de los atacantes.

China vuelve a situarse como el país desde el que más ataques se lanzaron, aumentando en un 10% su presencia sobre el total de países en relación al número de ataques. Una vez más, Nanjing es la fuente de un gran número de ataques procedentes de este país.

Un gran inconveniente derivado del bajo número de accesos por IP es que p0f no pudo determinar en la mayoría de los casos el SO desde el que se lanzaban. Con respecto a los que sí, en este caso Linux deja de ser el predominante y con una ventaja muy ajustada Windows pasa a ser el SO desde el que más se ataca la honeypot.

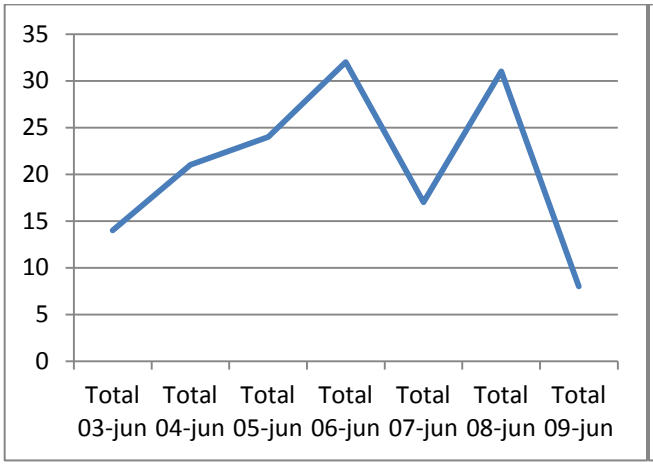
En relación a los puertos como ha pasado en la gran mayoría de las honeypots el servicio más atacado es HTTP, en este caso en el puerto 8080, con unos pocos accesos al puerto 25 y correspondiente al protocolo SMTP; y un número de accesos aún más bajo al puerto 135.

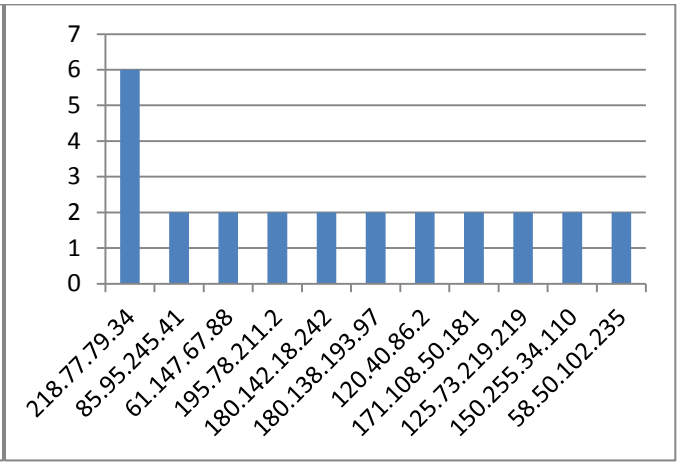
No se produjo ningún acceso por SSH (abierto aquellos días de Kippo no estaba en funcionamiento) o Telnet por lo que no se registró ninguna IP en la lista de IPs bloqueadas.

Honeypot de aplicación web.

Glastopf:

Ataques totales: 147
Promedio de ataques/día: 21
IPs distintas: 76
Periodo de actividad:
Fecha de inicio: 3 junio 2014
Fecha de fin: 9 junio 2014





Figuras 40 y 41. Actividad por días y Top 10 de IPs.

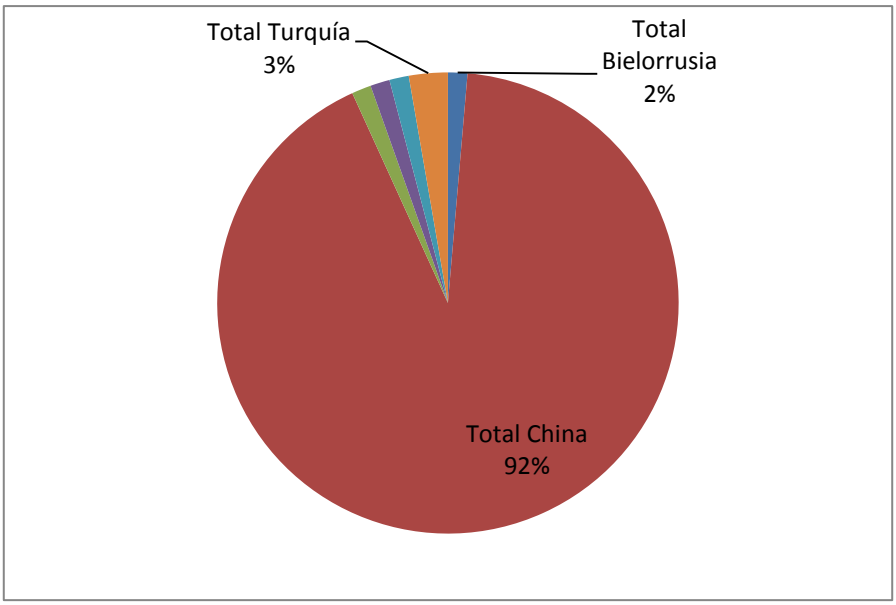
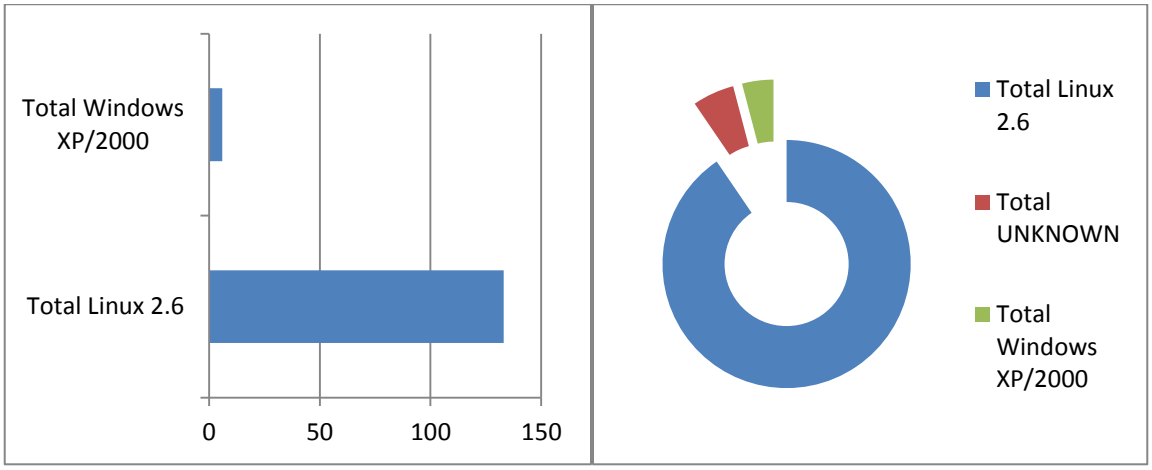


Figura 42. Distribución de ataques por país.



Figuras 43 y 44. Distribución de ataques por Sistemas Operativos.

Ataques por Puerto:

Esta Honeypot solo recibe ataques al puerto 80.

Observaciones:

La actividad en esta honeypot experimenta un crecimiento en los primeros días y tras un periodo de inestabilidad desciende en el último, logrando un número relativamente bajo de ataques.

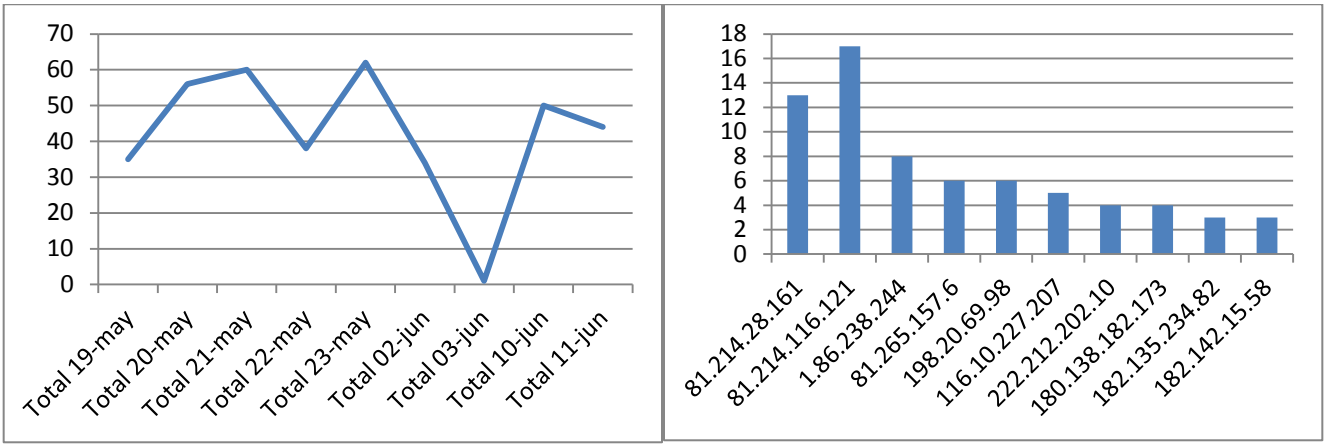
La IP con más ataques es 218.77.79.36, que llega a triplicar a las siguientes, observando una tendencia de 1 o 2 ataques por IP. Esto es en parte debido al mal funcionamiento de la Honeypot, produciendo constantes errores de conexión, tal y como se aprecia en los logs, debido a fallos en las conexiones difícilmente solucionables sin documentación disponible sobre la herramienta. De hecho nos encontramos a la honeypot con un número más bajo de ataques por IP, quizás debido a este inconveniente.

La presencia de ataques desde China vuelve a incrementarse con respecto a las honeypots anteriores y llega a su máximo con un 92% de origen del total de ataques. Como es habitual en el conjunto de honeypots, Nanjing es el lugar desde el que se origina la mayoría de ataques, con equipos Linux 2.6, generalmente con un tiempo de actividad estable salvo en algunos casos puntuales en los que los ataques se producen desde máquinas con un número de horas de actividad muy superior.

Honeypot Windows.

HoneyBOT:

Ataques totales: 380
Promedio de ataques/día: 25.55
IPs distintas: 230
Periodo de actividad:
Fecha de inicio: 19 mayo 2014
Fecha de fin: 11 junio 2014



Figuras 45 y 46. Actividad por días y Top 10 de IPs.

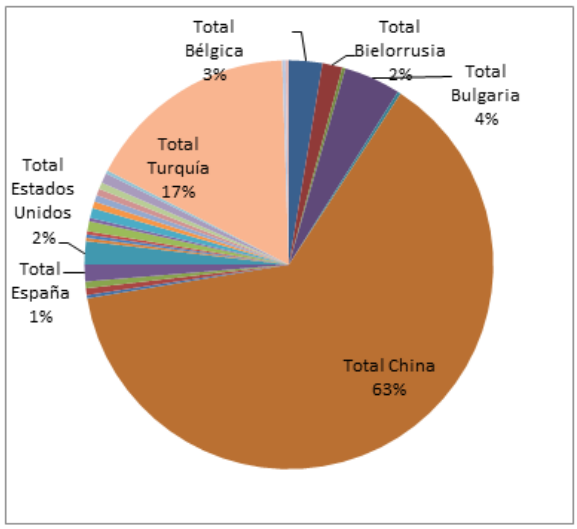


Figura 47. Distribución de ataques por país.

Ataques por SO:

En esta honeypot no es posible determinar el SO origen.

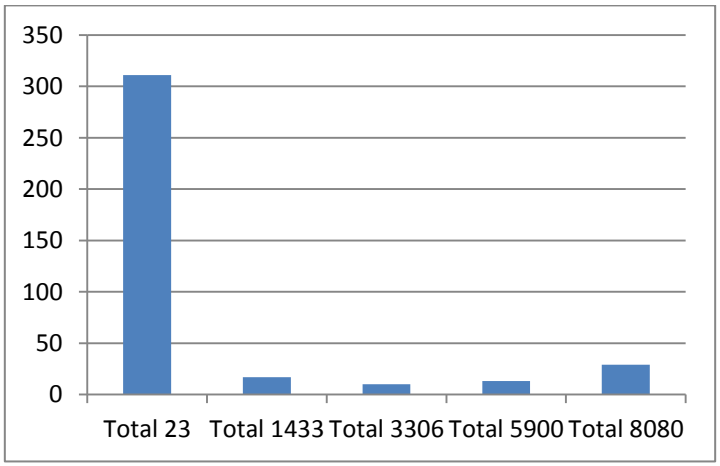


Figura 48. Distribución de ataques por puertos.

Observaciones:

En este caso probablemente estemos ante una de las honeypots más fáciles de configurar y con una presentación de resultados más amigable, aunque no exenta de problemas como la identificación del SO atacante.

En general los ataques no experimentan variaciones demasiado pronunciadas a lo largo del tiempo, si bien el día 03 de junio, a pesar de mantener la máquina funcionando el mismo número de horas que el resto de días solo se recibe un ataque.

El número de ataques por IP es mayor que en las máquinas basadas en Linux Mint, aunque no se puede equiparar al volumen de la honeypot implementada con Kippo.

China pierde representación en el total de ataques llegando al 63%, mientras que países con una presencia prácticamente nula en los ataques a otras honeypots, como Turquía, cobran relevancia, llegando al 17% en este caso. Así mismo se produce un aumento en los orígenes de los ataques, siendo la máquina que ha recibido ataques de un mayor número de países.

En general los ataques se han producido al servicio Telnet, sabidamente inseguro. Aunque también se aprecia en los últimos días una combinación de ataques al puerto 8080 y 23 desde una misma IP. Los puertos 3306 (MySQL), 1433 (MS SQL) y 5900 (VNC) también han sido fuente de ataques, aunque en menor medida. Es curioso como en esta honeypot el puerto atacado predominante pasa de ser el 80 al 23.

Cabe señalar dos particularidades bastante extrañas ocurridas dentro de esta Honeypot, puesto que en varias ocasiones los ataques estaban identificados con una IP de tipo privado que no se corresponde a las asignadas en el segmento de red en el que se encontraban las máquinas. Ni siquiera se trata del mismo tipo de IP, debido a que éstas correspondían a un Tipo B mientras que el segmento de las honeypots implementaba una de Tipo C.

Otro problema fue el sistema de exportación de logs, HoneyBOT permite exportar todas las capturas, pero luego no se puede volver a visualizar en la herramienta, lo cual dificulta enormemente el proceso de análisis, por esta razón, y a pesar de que la honeypot estuvo funcionando más días de los que se analizan, solo se ha podido realizar el análisis de los días presentes en el documento complementario.

Capítulo VII: CONCLUSIONES

CONCLUSIONES

En relación a las características de los ataques:

La mayoría de los ataques no parecen seguir un patrón preestablecido o fijo de actuación por lo que es de suponer que se realizan a pequeña escala desde equipos particulares o pequeñas instalaciones. Este no es el caso de la gran mayoría de los ataques procedentes de China. Las características similares de las máquinas atacantes junto al hecho de que gran parte de ellas proceden de Nanning o Chengdu hacen suponer que se trata de un complejo especializado en el rastreo de Internet en busca de algún tipo de elemento o simplemente una lanzadera de ataques.

Lo cierto es que en Internet aparecen varias referencias a este complejo, desde CyberSecurityBlog hablan de cómo su herramienta de seguridad "Shadow Security Suite" han detectado el mismo tipo de ataques que las honeypots de este proyecto, solo que en mi caso el rango de localizaciones de IPs se ve incrementado considerablemente. Se trata de ataques y sondas lanzadas de forma coordinada desde múltiples localizaciones y siempre en territorio chino. Muchos de estos equipos son utilizados como redes zombies para lanzar ataques en masa. A diferencia de la información aportada por CyberSecurityBlog, los puertos atacados no corresponden a servicios CAP (Calendar Access Protocol) o NetBIOS, sin embargo sí que se aprecian ataques a puertos dedicados para SQL. Una vez que se logra el acceso el sistema intenta explotar vulnerabilidades en Java ejecutando scripts en Javascript.

El motivo por el que las honeypots se han visto afectadas es que el sistema de escaneo se basa en rangos de IPs, por lo que no es necesario que la IP pertenezca a ningún tipo de organización o goce de especial relevancia.

En relación al uso de honeypots:

Como se ha comprobado en el caso anterior, pero también gracias al resto de ataques sufridos, las honeypots son una herramienta de gran ayuda para estudiar los ataques a sistemas informáticos, las tendencias en los ataques y las metodologías. Lo cierto es que los barridos de IPs vuelve objetivo de ataques no solo puntos estratégicos de la red, sino a cualquier equipo desde el que se pueda lanzar un ataque posterior, lo cual es más que palpable debido a que las máquinas desplegadas no contenían ningún tipo de información relevante, no contaban con una capacidad de procesamiento elevado ni se encontraban dentro de ninguna red corporativa. En la mayoría de los casos ni siquiera se suministraba información una vez que se accedía a los puertos, y sin embargo en un corto periodo de tiempo el número de ataques ha sido bastante elevado.

Sin estas herramientas, estudiar los ataques y el malware sería mucho más complejo, puesto que se pasaría de ser una parte activa con la ayuda de las honeypots, a un mero espectador que recibe ataques sin una gran capacidad de análisis previo de los mismos.

Lo cierto es que para poder obtener niveles de información útiles, más allá de un mero recuento estadístico, se hace necesaria la utilización de honeypots de media y alta interacción, pero al mismo tiempo y especialmente en el caso de las últimas, su utilización es muy compleja y en cierto modo peligrosa; pudiendo suponer el comprometimiento del sistema general.

Estamos por tanto ante herramientas de enorme poder pero que conllevan riesgos asociados que han de ser considerados cuidadosamente antes de desplegarlas.

En relación a la efectividad de las honeypots desplegadas:

En la definición del alcance del proyecto se optó por la utilización de honeypots que fueran abordables debido al poco tiempo para realizar el proyecto y al elevado conocimiento sobre las mismas que hay que poseer para desplegar satisfactoriamente otras más avanzadas. A lo largo del proceso he podido analizar las formas en la que los ataques se realizan sobre los puertos habilitados, no obstante debido a que en su mayoría se trata de honeypots de baja interacción la información no es muy abundante.

Había tres honeypots de media interacción, las desplegadas gracias a las herramientas Kippo, Dionaea, HoneyBOT y Glastopf, pero por desgracia en los tres primeros casos no se llegó a producir acceso más allá del puerto de servicio y en el último el gran número de errores y excepciones producidas han imposibilitado el análisis de las interacciones. Esto ha supuesto el aprovisionamiento de una menor cantidad de información, no solo a nivel cuantitativo sino especialmente a nivel cualitativo, por lo que en este sentido el proyecto ha resultado insuficiente.

Con respecto a la comparación entre las dos honeypots para Windows, lo cierto es que tanto la versión de código libre HoneyBOT como KFSensor ofrecen resultados similares, por lo que a menos que se busque una funcionalidad específica de un software concreto la honeypot de software libre es igual de buena que la de software propietario.

En relación al software de emulación escogido:

Desde un primer momento una de las principales dificultades del proyecto se basó en el software de despliegue de honeypots a emplear. Por un lado las herramientas Windows que constan en la mayoría de estudios sobre honeypots ya no se encuentran disponibles en internet, por lo que su uso, incluso en el caso de encontrarlas en algún repositorio, no se antojaba aconsejable.

Las herramientas que sí eran fácilmente adquiribles eran todas de código propietario y requerían del pago de tasas desproporcionadas para su utilización en un proyecto de final de máster.

Con este problema en mente al final se optó por una solución de código libre HoneyBOT que ofrecía funcionalidades aceptables. Por otro lado también se probó una versión comercial en su versión de evaluación, pero dado que su periodo de prueba había expirado cuando se empezó a volcar la información extraer y analizar los informes resultó excesivamente complejo.

En relación a las herramientas implementadas bajo sistema Linux el problema es distinto. Aquí, y gracias al proyecto Honeynet, hay bastantes herramientas disponibles, pero en su mayoría carecen completamente de documentación, incluso para los procesos de instalación y despliegue. Por otro lado la mayoría presentan errores en tiempo de ejecución, como en el caso de Conpot o Glastopf que dificultan en gran medida la obtención de resultados y posterior análisis.

En este aspecto se ha optado por ofrecer el mayor número de información posible con las herramientas disponibles, a pesar de que el funcionamiento de alguna entrañaba algunas complejidades.

En relación al proceso general de este proyecto:

Durante la realización de este proyecto me he encontrado con diversas dificultades que he ido sorteando con mayor o menor suerte. La primera de ellas es la falta de información y herramientas actualizadas, lo cual ha supuesto una menor posibilidad de innovación en la parte teórica del proyecto.

Por otro lado problemas con el router y la redirección y traducción de puertos ha ocasionado que no haya podido realizar capturas durante tanto tiempo como me hubiera gustado y que haya tenido menos tiempo para realizar el análisis posterior. Esto en sí tampoco es un problema excesivamente importante, ya que a lo largo de todos los días se aprecia cierto patrón en los ataques, por lo que aunque un número mayor de días a la escucha hubiera supuesto un mayor volumen de información confío en que los resultados hubieran sido muy similares.

OBJETIVOS CONSEGUIDOS

- Comprensión a nivel teórica de los fundamentos y principios de las honeypots.
- Comprensión práctica de diversos tipos de honeypots de baja y media interacción.
- Adquisición de habilidades para la configuración de un gran número de software de despliegue de honeypot.
- Estudio de las fuentes de ataque.
- Estudio de los servicios más atacados a través de Internet.
- Descubrimiento a nivel personal de la red china de distribución de malware y ataques a equipos.
- Comprensión de la distribución a nivel mundial de ataques informáticos.
- Análisis de los sistemas más atacados en base a la naturaleza de los servicios.

OBJETIVOS NO CONSEGUIDOS

- Obtención de información de las intrusiones en las honeypots de media interacción (aunque sí se hayan registrado los ataques en sí).
- Despliegue de una Honeypot que emule un entorno IaaS.
- Despliegue de una Honeypot que emule un sistema Android.
- Depuración completa de las honeypots utilizadas para que no produzcan fallos en su ejecución.
- Utilización de herramientas que inicialmente se consideraron para el despliegue como HoneyD: en este caso particular se optó por no utilizar este software debido a que en los escaneos con nmap todos los puertos aparecían filtrados, en su lugar se optó por la utilización de Conpot que a pesar de producir errores ha cumplido completamente con las expectativas iniciales.

POSIBLES AMPLIACIONES DEL TRABAJO

Mayor precisión en la implementación de Honeypots.

Por un lado me gustaría poder llegar a comprender mejor el funcionamiento de todas las honeypots empeladas. Depurar los errores encontrados y conseguir un nivel de conocimiento suficiente como para poder llegar a implementar honeypots de alta interacción de forma segura.

Por otro lado, y debido a la falta de tiempo, hay varias honeypots que no he podido implementar y que habría sido interesante desplegar. En concreto me encuentro implementando una honeypot de alta interacción basada en una infraestructura de nube privada para un entorno IaaS de Cloud Computing con la ayuda de OpenStack. En dicha honeypots se analizaría tanto las interacciones con el SO físico y el hipervisor como con el SO virtualizado.

Otro sistema que estuve considerando en las fases iniciales y que acabé descartando fue la implementación de una honeypot que emulara un sistema móvil como Android. Aunque a nivel de comunicaciones los dispositivos móviles no se comportan de la misma manera y utilizan tecnologías de comunicación adicionales habría sido interesante implementar un sistema de este tipo, que si bien no emulara servicios escuchando en puertos como el resto de honeypots, pudiera analizar ataques. En relación a este punto hay software como APKInspector que puede resultar interesante, aunque en cierto modo se trata más de un sandbox que de una honeypot en sí, por lo que descarté su utilización.

Mejor entendimiento de los ataques

Debido al tipo de honeypot implementada la información aportada por los ataques es muy reducida, si bien ayuda a empezar a entender ciertos aspectos de los ataques como los ataques coordinados distribuidos o los sistemas operativos más utilizados, aún queda mucho por estudiar al respecto. Una vez analizadas las fuentes de los ataques y las metodologías hasta el momento en el que acceden al puerto es de vital importancia y tremendamente interesante observar y analizar qué pasa después, las formas en las que los atacantes toman el control de sistema o empiezan a distribuir malware. Hasta cierto punto la falta de conocimiento en esta segunda etapa ha sido debida a un problema de cantidad de ataques, ya que en la máquina con Kippo y Dionaea no se ha llegado a ejecutar ningún comando, lo cual hubiera podido aportar información útil.

Mayor tiempo de captura.

Aunque es posible que los patrones de ataque se siguieran repitiendo a lo largo del tiempo como lo han estado haciendo durante el tiempo que las honeypots han estado accesibles, una mayor cantidad de información quizás aportara robustez a los resultados obtenidos para cada honeypot.

Mejora de la apariencia de las Honeypots (nivel de detalle y camuflaje)

Volver las Honeypots más atractivas mejoraría el promedio de ataques y el grado de interacción y permanencia en la Honeypot de los mismos, debido a que con el diseño actual los atacantes pierden el interés en la Honeypot tras unas pocas interacciones, probablemente debido a que no hay nada que les motive a permanecer en ellas.

BIBLIOGRAFÍA

[1] Repson "Dionaea Honeypot + DionareaFR", 2013 [versión online]: http://www.repson.org/2013/09/dionaea-honeypot-dionaeafr/

[2] 1aNormus "HONEYDRIVE – REVIEW", 2012 [versión online]: http://www.tekdefense.com/news/2012/12/27/honeydrive-review.html

[3] Seifreed "Conpot (ICS/SCADA Honeypot)", 2013 [versión online]: http://www.dragonjar.org/conpot-icsscada-honeypot.xhtml?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+dragonjar%2FpKru+%28La+Comunidad+DragonJAR%29

[4] K. Schuttler "Research in Honeypot Technologies", Eastern Michigan University College of Technology [versión online]: http://www.emich.edu/ia/pdf/research/Honeypotresearch.pdf

[5] Niels Provos "A Virtual Honeypot Framework", Google Inc. 2004 [versión online]:http://static.usenix.org/event/sec04/tech/full_papers/provos/provos_html/

[6] Lance Spitzner "Honeytokens: The Other Honeypot", 2003 [versión online]: http://bandwidthco.com/sf_whitepapers/honeypots/Honeytokens%20-%20The%20Other%20Honeypot.pdf

[7] F. Pouget, M. Dacier, V. H. Pham "On the advantages of deploying a large scale distributed honeypot platform" ECCE, 2005 [versión online]: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.194.1780

[8] E. Alata, V. Nicomette, M. Ka{â}niche, M. Dacier, M. Herrb "Lessons Learned from the deployment of a high-interaction honeypot" EDCC, 2007 [versión online]: http://arxiv.org/abs/0704.0858

[9] F. Pouget, M. Dacier "Honeypot-based Forensics", Institut Eurécom, 2004 [versión online]: ftp://ftp.mirror.ac.za/www.honeynet.org/papers/individual/AusCERT_fullpaper_BIS.pdf

[10] K. Lee, J. Caverlee, S. Webb "The Social Honeypot Project: Protecting Online Communities from Spammers", 2010 [versión online]: http://faculty.cse.tamu.edu/caverlee/pubs/lee10www.pdf

[11] S. Khattab, R. Melhem, D. Mossé, T. Znati, "Honeypot back-propagation for mitigating spoofing distributed Denial-of-Service attacks", Journal of Parallel and Distributed Computing, 2006 [versión online]: http://dl.acm.org/citation.cfm?id=1232117

[12] R. McGrew "Experiences With Honeypot Systems: Development, Deployment, and Analysis", IEEE, 2006 [versión online]: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.80.9889

[13] M. L. Bringer, C. A. Chelmecki, H. Fujinoki "A Survey: Recent Advances and Future Trends in Honeypot Research" MECS, 2012 [versión online]: http://www.mecs-press.org/ijcnis/ijcnis-v4-n10/IJCNIS-V4-N10-7.pdf

[14] L. Spitzner "Honeypots: Definitions and Value of Honeypots" 2003 [versión online]: http://www.tracking-hackers.com/papers/honeypots.html

[15] Autor Desconocido "HONEYPOTS", The Study Materials [versión online]: http://www.thestudymaterial.com/presentation-seminar/electronics-presentation/256-honeypots.html?start=6

[16] A. Stolarski "Virtual Honeypots" Infosec Institute, 2013 [versión online]: http://resources.infosecinstitute.com/virtual-honeypots/

[17] B. Posey "Strategies for real and virtual honeypots" TechRepublic, 2004 [versión online]: http://www.techrepublic.com/article/strategies-for-real-and-virtual-honeypots/#

[18] B. Posey "Real vs. Virtual honeypots" SearchEnterpriseDesktop 2005 [versión online]: http://searchenterprisedesktop.techtarget.com/tip/Real-vs-virtual-honeypots

[19] C. Seifert, I. Welch, P. Komisarczuk "HoneyC – The Low-Interaction Client Honeypot" 2006 [versión online]: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.6882&rep=rep1&type=pdf

[20] Autor Desconocido "Offshore CyberProbes/Cyber Attacks: New Sophisticated Coordination" Black Lab Security Lab, 2007[versión online]: http://cybersecurityblog.blogspot.com.es/2007/10/offshore-cyberprobescyberattacks-new.html

[21] Autor Desconocido "Requirements Collection and Analysis" European Network of Affined Honeypots, 2006 [versión online]: http://www.fp6-noah.org/publications/deliverables/D0.2.pdf

[22] L. Spitzner "Honeypots: Tracking Hackers." Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

About.com: TCP/IP Transmission Control Protocol / Internet Protocol [online]: http://compnetworking.about.com/od/tcpip/

Speedguide.net [online]: http://www.speedguide.net/

IP WHOIS Search [versión online]: http://www.ip-adress.com/whois/