Honeypots
(parte II)

Honeypots (parte II)

- **Agenda**
  - Honeynet virtuales (alto nivel de interacción)
  - Honeypots de bajo nivel de interacción
  - Los 3 principios que gobiernan a los Honeypot
  - Recolección, Análisis y Control de Datos
  - Lineas de trabajo actuales y futuras.
  - Client Honeypot
  - Honeypot Farm usando Honeymole (Agosto 2007)

Grupo de Seguridad Informática

Honeypots (parte II)

- Virtual Honeypots ("alto nivel de interacción")
  - Motivación, Implementaciones
  - ¿Como proteger a un Honeypot/Honeynet?
- Honeypots de bajo nivel de interacción
  - Objetivo, Ventajas / Desventajas
  - Implementaciones
  - ¿Como protegernos de vulnerabilidades del Honeypot?
- Soluciones Híbridas (Honeypot Farms)
- Client Honeypots

Honeypot virtuales

- Alto nivel de Interacción (Honeynet)
  - Físicos
  - Virtuales
- En general se caracterizan por:
  - Hacer uso intensivo de recursos
  - Difíciles de construir
  - Complejos de mantener
- Solución: Virtual Honeynets

Honeypot Virtuales

- Solución que permite “ejecutar” todo lo necesario en una misma computadora
- !! Posible gracias al software de virtualización de sistemas operativos !!
- Se clasifican en dos tipos
  - Self-Contained Virtual Honeynet
  - Hybrid Virtual Honeynet

Honeypot virtuales

- Desventajas
  - Los Sistemas Operativos que podemos desplegar están limitados (hardware y soft de virtualización)
  - El atacante puede comprometer el software de virtualización
  - Fingerprinting (ser detectado)

Honeypot Virtuales Implementaciones

- Las dos implementaciones mas difundidas estan basadas en:
  - VMware
    http://www.vmware.com
  - User-Mode Linux (UML)
    http://www.user-mode-linux.sourceforge.net
  - ARGOS (Vrije Universiteit Amsterdam)
    http://www.few.vu.nl/argos
  - ***¿ XEN ?***

Honeynet Virtuales VMware

- Hay disponibles versiones gratuitas (Player / Server).
- Soporta multiples sistemas Operativos para los sistemas **guest** (FreeBSD, Linux, Windows, Solaris, etc)
- Para el monitoreo del sistema guest disponemos de SEBEK ***http://www.honeynet.org/tools/sebek***

Honeynet Virtuales UML

- Solo podemos emular sistemas Linux
- Mecanismo para monitoreo sistema ***guest*** (TTY logging)
- HPPFS. Habilidad para modificar el /proc del sistema guest y parecer un S.O. real.
- SKAS Mode. Binarios del kernel UML y datos son invisibles a sus procesos
- multiples sistemas virtuales compartiendo el root File System (COW, Copy On Write)

Honeynet Virtuales Argos

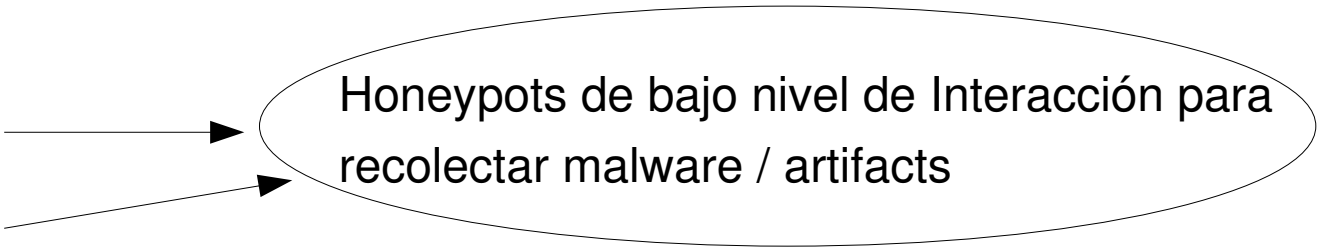
- Objetivo: Detectar zero-day attacks
- Basado en: Análisis dinámico de datos corruptos (*"Dynamic Taint Analisis"*)
  Los atacantes envían datos de entrada mal formados a un programa de forma que esos datos influyan el flujo de ejecución
- Idea: Monitorear todos los datos de entrada externos (tainted o corruptos) de los programas
- Usa QEMU como mecanismo de emulación/virtualización

Honeynet Virtuales honeywall

- ¿Como protegernos de un Honeypot en caso de haber sido comprometido?
- Evitar ser usado en una Botnet o para enviar spam.
- Solución: !! **Honeywall !!**
- Desarrollado como parte de la arquitectura de la Gen III y Gen IV del proyecto honeynet
- Bridge “transparente”

Honeypots de bajo nivel de Interacción

- Objetivo
  - Detectar exploits conocidos
  - Medir que tan atacada es una red
- Implementaciones
  - Honeyd
  - GHH, Google Hack Honeypot
  - PHP.HoP: PHP HoneyPot
  - Nepenthes
  - Honeytrap



Implementaciones

- Honeyd, Honeytrap, Nepenthes ....
- GHH
  - Simula aplicaciones web vulnerables o configuradas incorrectamente
  - Una vez que el sitio GHH es indexado, va a ser devuelto como resultado de búsquedas de motores
- PHP.HoP
  - Provee un mecanismo para identificar y observar “ataques” realizados por herramientas automáticas

Honeypots de bajo nivel de Interacción

- ¿Hay que asegurarlo?, ¿Contra que?
- Respuesta:
  Vulnerabilidades en el soft. del Honeypot
- ¿Como?
  - chroot Jail
  - systrace, ***http://www.citi.umich.edu/u/provos/systrace***
  - Otras: AppArmour, Solaris Containers, Dominios SeLinux, Virtualización (?)

los 3 principios que gobiernan ..

- **Seguridad**
  El atacante o adversario debe ser bien aislado o contenido.
- ***Performance (*)***
  Indicador de cuanto tráfico puede manejar el Honeypot o con cuantos adversarios interactuar
- ***Fidelity (*)***
  Grado de realismo provisto al atacante
- Es difícil balancear los 3 objetivos

Recolección y Análisis de datos

- Tema medular a los efectos de poder hacer un análisis de los datos recogidos de:
  - honeypot,
  - tráfico de red,
  - sistema guest y sistema host donde se ejecuta las virtualizaciones, etc.
- Independiente del nivel de interacción.
- Concepto “formalmente” incorporado en Gen IV de Honeynet y sobre el que se está trabajando.

Grupo de Seguridad Informática

Lineas de trabajo del GSI

- Obtener datos de los que esta pasando a nivel nacional.
- Primer etapa: Honeypots de bajo nivel de interacción (honeyd y honeytrap)
- Desarrollo de framework para la recolección de datos y posterior análisis así como también la generación de alertas

Grupo de Seguridad Informática

Lineas de trabajo GSI

- Virtualización (*)
- Sandboxing de aplicaciones (systrace, AppArmour, Solaris Containers, SeLinux)
- Segunda etapa: Evolucionar a honeypots de alto nivel de interacción (honeynet)
- Otras lineas posibles:
  - **Client Honeypots (*)**
  - **Honeypot Farms**

Client Honeypot

- Motivación: existe una gran variedad de ataques sobre aplicaciones cliente.
  - Microsoft Internet Explorer, Outlook/Outlook Express, Media Player, Office Suite.
  - Real Player, AOL Instant Messenger, Winamp Media Player, Mozilla Firefox, etc
- ¿Como diseñar un honeypot para aprender sobre este tipo de ataques?

Client Honeypot

- Observar:
  - Los clientes dependen del servidor con el que están trabajando.
  - deben seguir un protocolo y son estos quienes deben iniciar la conexión !!!
- No vamos a esperar a los atacantes en forma pasiva, vamos a buscar en forma activa el código malicioso (web based client honeypot)
- También podemos tener client-side honeypots pasivos (p.e. Clientes de correo)

Client Honeypot: Tipos

- Bajo nivel de interacción
  - Usa un cliente simulado (honeyC o wget)
  - Usualmente utiliza análisis estático y *signature matching*
- Alto nivel de interacción
  - Usa un sistema operativo dedicado
  - Conduce al cliente para interactuar con el servidor malicioso potencial
  - Luego de cada interacción chequea el sistema operativo por cambios de estado sin autorización

Web Client Honeypot Implementaciones

- Bajo nivel de interacción
  - HoneyC
    http://honeyc.sourceforge.net
- Alto nivel de interacción
  - HoneyClient o MITRE HoneyClient
    http://www.honeyclient.org/trac
  - Capture-HPC
    http://capture-hpc.sourceforge.net
  - Strider HoneyMonkey Exploit Detection
    http://research.microsoft.com/HoneyMonkey

Grupo de Seguridad Informática

Bibliografía (parte II)

- N. Provos, T. Holtz, Virtual Honeypots From Botnet Tracking to Intrusion Detection,
  ISBN-10 0-321-33632-1, Julio 2007
- Know Your Enemy: Malicious Web Servers, Honeynet Project, Agosto 2007
- Know Your Enemy: Defining Virtual Honeynets, Honeynet Project, Enero 2003
- Know Your Enemy: Learning with VMware, Honeynet Project, Enero 2003
- Know Your Enemy: Learning with User-Mode Linux, Honeynet Project, Diciembre 2002
- Know Your Enemy: Honeynets, Honeynet Project, Mayo 2006
- [Seif07] C. Seifert, R. Steenson, T. Holz, R. Yuan, M. A. Davies, Know Your Enemy: Malicious Web Servers, The Honeynet Project, Agosto 2007.