Arania : Herramienta para colectar código malicioso en ataques a servidores web.

González Hugo
hugo.gonzalez@upslp.edu.mx
Universidad Politécnica de San Luis Potosí

Resumen— En este documento se describe la historia y los motivos que llevaron al desarrollo de "arania", la herramienta para colectar código malicioso en ataques de inyección de código a servidores web en producción. Se explica la importancia de dicha herramienta al ser no-intrusiva, sus diferencias con los honeypots. Además se describe arquitectónicamente y se explica el funcionamiento. Finalmente se concluye con las experiencias obtenidas y con un análisis de la información y código obtenidos a través de arania.

Índice de Términos—Seguridad Web, Honeypots, ataques web.

I. INTRODUCCIÓN

En la actualidad, la propagación de virus, gusanos y ataques automáticos en Internet ha crecido enormemente, en 2001, "Code Red" y "Nimda" infectaron cientos de miles de equipos [1][2] causando pérdidas por millones de dólares [3]. Luego de una relativa calma apareció "SQL Slammer" en enero de 2003, y se esparció rápidamente a través de Internet [4] . Por su rápido mecanismo de escaneo, logró infectar a un 90% de los host vulnerables en los primeros 10 minutos [4], además la enorme cantidad de paquetes de escaneo generados por el slammer ocasionó un ataque de denegación de servicio a nivel global en Internet, muchas redes en Asia, Europa y América estuvieron fuera de servicio durante horas.[5]

Actualmente existen organizaciones como CERT[6], CAIDA[7] y el Instituto SANS [8] que se encargan de monitorear el tráfico en Internet, y ponen especial atención en tráfico anormal, el cual al ser detectado es inmediatamente analizado por expertos.

Los fenómenos recientes, incluyen un nivel de control hacia las máquinas infectadas, formando redes de equipos latentes y disponibles para ser usados en diferentes tipos de actividades, como la propagación del gusano, ataques distribuidos de denegación de servicios, escaneos de equipos vulnerables y más. A este tipo de redes zoombie también se les conoce como botnets. Por lo general estas redes son controladas a través de un servidor de irc.[9]

Aunado a esto, el desarrollo de aplicaciones web cada vez es mayor, presentando nuevas oportunidades y vectores de ataque para gusanos, ataques automáticos, y porque no, también para ataques manuales.[10] Esto generalmente se debe a que el código de estas aplicaciones puede ser de baja calidad [10] o a la numerosa lista de vulnerabilidades reportadas al paso del tiempo..

II. TIPOS DE ATAQUES A APLICACIONES WEB

A. Inyección de SQL

Las entradas de usuario son utilizadas por los atacantes para obtener información o privilegios no autorizados en una base de datos. La mayoría de las aplicaciones web actualmente hacen uso de base de datos para almacenar su contenido. En este ataque se pretende modificar el contenido mostrado por la página web, o bien, conseguir información sobre usuarios y contraseñas del sistema.

B. Inyección de código

Alguna variable o alguna condición de la aplicación web permite al atacante introducir cadenas de código a ejecutarse dentro del servidor, por lo general estos códigos van encaminados a descargar código malicioso en el servidor y luego

ejecutarlo. Con este ataque se pretende modificar contenido mostrado por la aplicación, para luego tomar control completo sobre el sistema, instalando una puerta trasera o un bot.

C. Inclusión de código remoto

Alguna variable o alguna condición de la aplicación web permite al atacante que se ejecute código que está almacenado en otro lugar, por lo general se dá en instrucciones "include" no verificadas. Al igual que en la inyección de código, se pretende tomar control del servidor, instalando una puerta trasera o un bot.

D. Scripting de sitio cruzado (XSS)

Esta técnica sirve para utilizar el servidor web como vía para realizar ataques a usuarios del sitio, robando información sobre sesiones, o redirigiéndoles a otras páginas. Lo que se realiza es incluir código en el sitio, que luego es ejecutado por el navegador web del visitante.

En este documento nos enfocaremos a la inyección de código y a la inclusión de código remoto.

III. TRABAJO RELACIONADO

A. Honeypot

Un honeypot es un recurso informático, real o ficticio, que su valor reside en el uso malicioso o no autorizado[12]. Existen diferentes tipos de honeypots, entre ellos están los de baja interacción que solo emulan servicios, permitiendo recolectar una cantidad limitada de información, estos honeypots son los mas sencillos de implementar. Ejemplos de ellos on honeyd, nepenthes y kfsensors. También existen honeypots de alta interacción, que no son recursos emulados, sino reales, son sistemas operativos preparados para ser atacados y comprometidos, de los cuales se puede obtener mucha mas información a través de análisis forense, generalmente son configurados en una honeynet, que permite controlar mucho mejor el incidente y facilita el análisis de información [13].

B. php honeypot

Honeypot de baja interacción que emula aplicaciones web, permite que el atacante piense que ha ganado control completo del sistema, indexa por los 10 comandos más utilizados, los ataques recibidos. [14]

C. google hack honeypot (ghh)

Este proyecto surge como reacción al uso de google por los atacantes para buscar vulnerabilidades en aplicaciones web. El ghh utiliza la teoría de honeypots para dar mayor seguridad a tus aplicaciones web, detectando ataques y ofreciendo recursos simulados que son indexados por google. [15]

IV. SERVIDOR DEL ITSLP

A. Antecedentes

El Instituto Tecnológico de San Luis Potosí (ITSLP), tiene su sitio web[16] sobre un Manejador de contenido (CMS por sus siglas en Inglés) llamado Mambo [17] desde hace más de 2 años. Este manejador de contenido está escrito en PHP y utiliza MySQL como gestor de almacenamiento, se ha vuelto muy popular recientemente y existen cerca de 95,900 instalaciones indexadas por google al 22 de Marzo de 2007.

A finales del 2005, se reportaron varias vulnerabilidades relacionadas con Mambo, las cuales permitían ataques de tipo inclusión de código remoto. Los anuncios en formato original se muestran a continuación:

El 17 de noviembre de 2005
- Mambo "mosConfig_absolute_path" Remote File Inclusion Vulnerability

http://www.frsirt.com/english/advisories/2005/2473

- Mambo "mosConfig_absolute_path" Remote Command Execution Exploit
Advisory ID : FrSIRT/ADV-2005-2473
Rated as : High Risk

http://www.frsirt.com/exploits/20051122.mambo452_xpl.php.php

Con estos avisos, se aplicó el parche de seguridad al servidor, pero se comenzó a detectar múltiples intentos de aprovechar esta vulnerabilidad. Y más adelante se reportaron otras tantas vulnerabilidades del mismo estilo sobre el CMS, de las cuales también se siguen recibiendo ataques.

B. *Datos recolectados*

Existen diversos códigos que circulan por la red, la más utilizada en los ataques al servidor web del ITSLP fue un archivo llamado tool.gif, que en realidad es código php, html y javascript para realizar un defacement, o cambio de página principal, descargar y ejecutar archivos.

Se identificaron 42 nombres de herramientas diferentes, hospedadas en 220 sitios. La proliferación de sitios se debe principalmente que cuando identifican un archivo de este tipo es cancelado o borrado. La mayoría de los sitios son de alojamiento gratuito.

Otros códigos modificaban directamente la página principal. Hubo diferentes implementaciones, pero las funciones eran las mismas.

Durante los 3 meses de más intensa actividad en que se monitoreo el servidor a través de los registros, de destacan el total de intentos de explotación *43656*. En total se registraron *973* direcciones ip donde se generaron ataques. Las 5 direcciones con más de 500 ataques son :

Tabla 1 Lista de direcciones con más de 500 ataques

595	212.29.217.4
631	213.193.212.208
682	81.169.182.111
723	81.169.134.50
748	212.87.13.140
761	218.208.21.7
816	66.240.226.25
837	81.208.30.102
2032	193.255.143.5

Las direcciones pertenecen a servicios de webhosting y a 2 universidades.

V. ARANIA

A. *Justificación*

Luego de estar monitoreando el servidor, se intentaron descargar el código malicioso que se pretendía incluir, y el código del perlbot. Realizar esta tarea de forma manual fue ardua y no siempre se conseguía obtener la información, ya que algunos códigos ya habían sido eliminados. Y aunque las herramientas usadas que se obtuvieron tenían los mismos nombres, el código algunas veces variaba. Así que se decide desarrollar una herramienta que haga estas labores de forma automática.

B. *Diseño*

El diseño de arania es simple y concreto, se cumplen varios objetivos:

- Monitorea de forma automáticas los registros del servidor web.
- Es un modelo no-intrusivo, no afecta el funcionamiento del servidor web.
- Obtiene el código malicioso que se trato de incluir.
- Lo almacena marcándolo de acuerdo a su contenido.
- Mantiene un registro detallado.

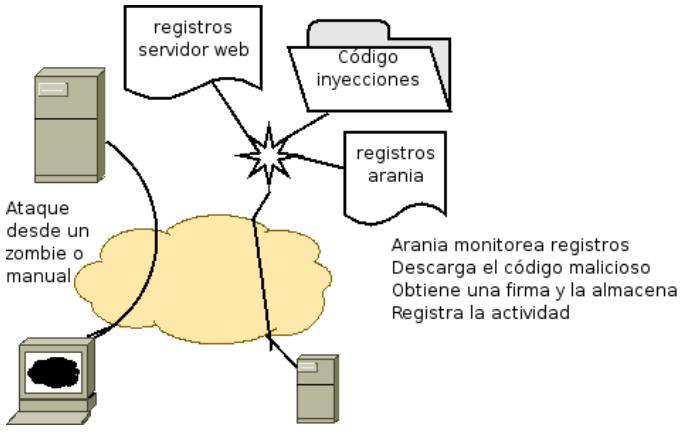

Figura 1 Diseño de arania

C. *Publicación y distribución*

Arania se distribuye bajo licencia GPLv2 y MIT. Está disponible en línea en el sitio de Mexican Honeynet Project [19], y en http://code.google.com/p/arania , hasta la fecha se han liberado 3 versiones.

VI. Resultados

Con el paso del tiempo, el intento de explotación a estas vulnerabilidades ha bajado considerablemente, llegando a tener solo algunos al día. Sin embargo esto demuestra que aún siguen activos equipos comprometidos, y que aún existen sitios vulnerables. Se han capturado 14 especimenes diferentes de inyección de código. Ya no se considera la cantidad de ataques, ni los intentos no exitosos de obtener el código.

Es realmente interesante el poder analizar estos códigos, algunos varían solo un poco y otros tienen funcionalidades totalmente diferentes, por ejemplo los archivos 01f3e68daeda9cb95d6820ea444334d3 y 33d483fe302eab5eb1f0ea74cf929e45 sirven para probar que el sitio es vulnerable. El archivo 0c725f73e1b1ae33d03ce6c75e056571 es un shell que da acceso completo al sistema vulnerable. El archivo ce1bcf10a381bc729d93355fa2976bff es otro shell, conocido como c99shell.

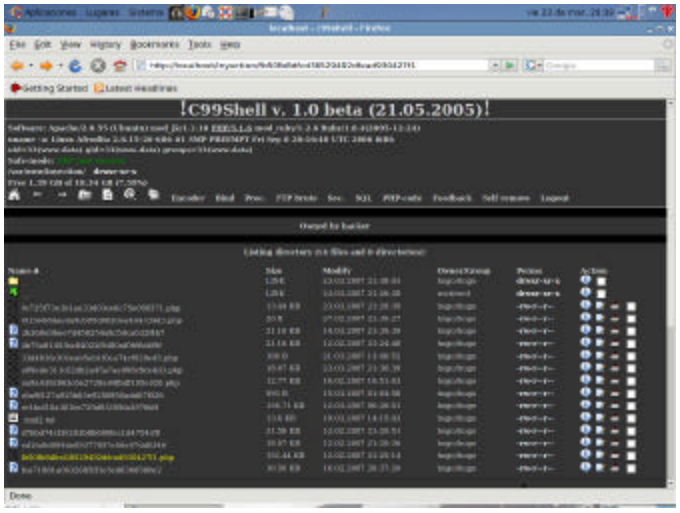

Figura 2 c99shell en acción

La misma herramienta de modificación de página esta en 3b360e9bee7d458254a8c5dca632f4b7 y fea71881a0632685ff1e5cd6366580e2, solo varía el sitio donde se obtiene la segunda parte de la herramienta.

Figura 3 Herramienta de modificación de página

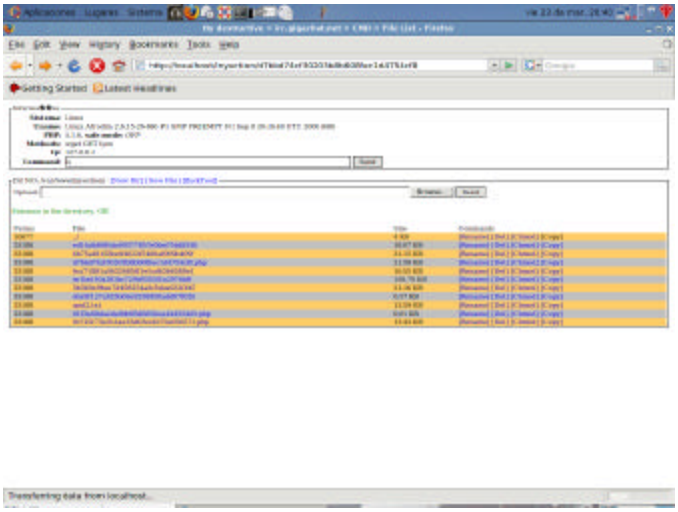

Figura 4 Otra herramienta para controlar sitios atacados

Arania será capaz de analizar también el contenido del archivo en busca de otros códigos que se incluyan, para de esta manera tener armada toda la herramienta.

VII. Discusión

En algunas ocasiones, junto con la inclusión de código remoto, también es capaz de detectar el

perlbot con el que se intenta tomar control completo del sistema atacado. La función principal de este es convertir el equipo atacado en un esclavo de la botnet, y esperar comandos e instrucciones a través del irc.

Este código desarrollado en perl es muy similar, incluye funciones para conectarse a un servidor IRC, y esperar las ordenes del administrador de ese canal. Las ordenes que puede ejecutar son funciones dentro del programa, y están el escaneo de puertos, ataque de denegación de servicio "tcpflood" que es una inundación de peticiones, otra función es obtener la versión del bot, y tiene una función para buscar servidores mambo vulnerables y atacarlos, de echo esta es el mecanismo principal de dispersión.

Existe también una función de inundación de peticiones de http "httpflood", y una de inundación de peticiones udp "udpflood".

Otra función interesante es la de posibilidad de actualización, e incluso algunos perlbots traen una herramienta para crear conexiones de shell invertidas, es decir, que la computadora atacada se conecte a la computadora del atacante y le ofrezca un shell de acceso al sistema.

Los resultados obtenidos en la monitorización indican una red muy amplia de botnets y zoombies, existen trabajos que analizan con mayor detalle este hecho[19], y la herramienta arania ayudará a entender con mayor detalle las herramientas utilizadas para realizar ataques sobre servidores web.

VIII. TRABAJO A FUTURO

Arania soportará el análisis del código descargado, en busca de más código que se encuentre en la web. Se pretende proponer una taxonomía para clasificar este tipo de código incluido, y que arania lo pueda catalogar automáticamente, además en un futuro se propondrá un método de comparación e intercambio de los códigos colectados y una adaptación para que funcione con un repositorio central para almacenar toda la información.

IX. CONCLUSIONES

Arania es una herramienta que nació de la necesidad de contar con mayor información sobre las herramientas utilizadas para atacar aplicaciones web, aunque originalmente se desarrolló para el CMS de Mambo, ahora puede utilizarse de forma general para obtener el código malicioso en ataques de inyección de código y de inclusión de código remoto en una gran variedad de aplicaciones web.

Las herramientas utilizadas para el ataque a CMS Mambo, también pueden ser utilizadas para atacar otras aplicaciones web que permitan la inclusión de código remoto, como se comprobó con una aplicación diseñada específicamente para permitir este tipo de ataques.

El problema de la inseguridad informática crece cada día, y los esfuerzos por mantenerse a salvo exigen cada vez más de las aplicaciones y del administrador de las mismas, es importante mantener contacto con anuncios para poder preparar nuestros sistemas y evitar ser víctimas de estos atacantes.

Al día de hoy, a más de 2 años del anuncio de la vulnerabilidad de Mambo, siguen existiendo ataques, como aún siguen existiendo ataques de gusanos de hace 5 años. El crecimiento de Internet y la gran cantidad de equipos conectados se pueden volver un paraíso para los atacantes..

REFERENCIAS

[1] D. Moore, C. Shannon, and J. Brown. Code-Red: a case study on the spread and victims of an Internet Worm. In Proc. ACM/USENIX Internet Measurement Workshop, France, November, 2002.

[2] CAIDA. Dynamic Graphs of the Nimda worm. Disponible en línea: http://www.caida.org/dynamic/analysis/security/nimda/, consultado el 2 Febrero 2007.

[3] USA Today News. The cost of Code Red: $1.2 billion. Disponible en línea: http://www.usatoday.com/tech/news/2001-08-01-code-red-costs.htm, consultado el 6 de Febrero de 2207.

[4] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. IEEE Security and Privacy, 1(4):33-39, July 2003
[5] CNN News. Computer worm grounds ?ights, blocks ATMs.Sitio web: http://europe.cnn.com/2003/TECH/internet/01/25/internet.attack/, consultado el 4 de Febrero de 2007.
[6] CERT Coordination Center. Sitio web: http://www.cert.org, consultado el 3 de Febrero de 2007.
[7] Cooperative Association for Internet Data Analysis. Sitio web: http://www.caida.org, consultado el 3 de Febrero de 2007.
[8] SANS Institute. Sitio web: http://www.sans.org, consultado el 3 de Febrero de 2007.
[9] Puri, Ramneek. Bot & Botnet: An Overview. GSEC Practical Assignment. August 2003.
[10] Riden, Jaime, McGeehan Ryan, Engert Brian, Mueter Michael. Know Your Enemy: Web Applications Threats. Febrero 2007. Disponible en línea : http://www.honeyner.org/papers/webapps, consultado el 10 de Marzo de 2007.
[11] D.J. Daley and J. Gani. Epidemic Modelling: An Introduction. Cambridge University Press, 1999.
[12] Spitzner, Lance, "Know Your Enemy: Learning about security threats". 2005.
[13] The honeynet project. Know Your Enemy: Gen II honeynets, 2003. Disponible en línea: http://www.honeynet.org/papers/gen2/, consultado el 20 de Marzo 2007.
[14] Sitio web del proyecto : http://www.rstack.org/phphop/, consultado el 20 de Marzo de 2007.
[15] Sitio web del proyecto : http://ghh.sourceforge.net/, consultado el 21 de Marzo de 2007.
[16] Sitio web del ITSLP: http://www.itslp.edu.mx, consultado el 22 de Marzo de 2007.
[17] Sitio web del CMS : http://www.mamboserver.com, consultado el 21 de Marzo de 2007.
[18] Bächer Paul, Holz Thorsten, Kötter Markis, Wicherski Georg. Know Your Enemy: Tracking botnets, 2005. disponible en línea: http://www.honeyent.org/papers/bots/, consultado el 23 de Marzo 2007.
[19] Sitio web del MHP: http://www.honeynet.org.mx, consultado el 23 de Marzo de 2007.

Autores

Hugo Francisco González Robledo, recibió el grado de Maestro en Ciencias por el Instituto Tecnológico de San Luis Potosí en 2005. Miembro fundador del Mexican Honeynet Project. Miembro profesional de la ACM. Tiene certificación LPI nivel 1. Su experiencia gira en torno al Software Libre y Seguridad Informática. Ha sido ponente en diversos foros internacionales.