Commands in the Air (Marzo 2018)

Chávez G. Alejandro

***Resumen*—Commands in the Air o COMAIR, es el nombre que recibe esta investigación, la cual pretende demostrar e identificar técnicas de ataques a sistemas Air Gapping, con el objetivo de protegerlos contra posibles intrusiones de este tipo en el futuro.**

***Índice de Términos*—Red de computadoras, Air Gap Malware, Ciberseguridad.**

I. Introducción

ESTE documento expone la primer etapa de una investigación que trata sobre la seguridad de los sistemas Air Gapping. En esta fase se expone una técnica ofensiva a este tipo de sistemas utilizando como vector de ataque los SSID (Service Set Identifier) que son identificados por las interfaces de red inalámbricas. En fases posteriores a la investigación, se espera aprovechar otros vectores de ataque, como por ejemplo, las ondas electromagnéticas que emiten algunos componentes de los computadores, con el fin de transmitir la información recabada a un receptor, o bien, utilizar señales de luz, señales acústicas, entre otros medios.

II. Metodología

Lo siguiente son los pasos que se utilizaron para demostrar dicha técnica de ataque, basada en los SSIDs capturados por las tarjetas de red inalámbricas:

A. Definir la idea

Se pretende desarrollar un programa como prueba de concepto a lo largo de toda la investigación, que atienda a las peticiones que le son solicitadas a través de los SSIDs que son captados por la interfaz de red inalámbrica del equipo comprometido. Estas solicitudes son instrucciones que el programa reconoce y ejecuta en el sistema operativo huésped.

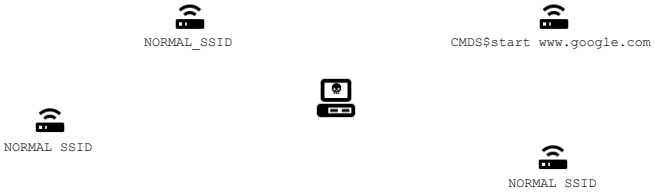

Fig. 1. Esquema del funcionamiento

Según el contexto en el que se esté corriendo el programa, serán los privilegios que tendrán las instrucciones al ser ejecutadas por el sistema operativo.

Las instrucciones pueden ser mostrar un mensaje en pantalla o abrir alguna página en el navegador, hasta grabar audio, video o ambos y subir las grabaciones a un servidor FTP, entre otras muchas posibilidades.

En el esquema anterior, se muestra el ejemplo de un equipo infectado con el malware y un SSID malicioso que contiene una instrucción, que al ser debidamente procesada por el malware, será ejecutada por el sistema operativo, abriendo la página de www.google.com en el navegador por defecto.

B. Análisis del programa

Al ser la primera fase, el programa tiene ciertas limitantes, por ejemplo:

- Solo funciona bajo sistemas operativos Windows y no permanece oculto para el usuario.
- Solo ejecuta comandos que sean entendidos por la terminal de Windows. Es decir, no incorpora la funcionalidad de mostrar mensajes en pantalla, grabar audio, vídeo o alguna otra característica.
- No cuenta con la capacidad de ejecutar instrucciones que usen más de una línea.

A continuación se describen los módulos y funciones con los que cuenta el programa en esta primer etapa:

Main

Este es el archivo principal. Desde este archivo se hacen las llamadas a los demás módulos.

mod_wifi

Este módulo es responsable de manejar los datos capturados por la interfaz de red inalámbrica. En la siguiente tabla se muestran y describen las funciones contenidas en este módulo.

TABLA I
Módulo de Wi-Fi

Nombre	Argumentos	Descripción
scan	Ninguno	Muestra una lista de los SSIDs de las redes inalámbricas que se encuentran dentro del rango de cobertura de la tarjeta de red inalámbrica.
executeCommand	int, char[]	Limpia la instrucción y ejecuta el comando capturado con la función scan(). Recibe como parámetros un número (INT) que determina el tipo de instrucción y un arreglo (CHAR) con dicha instrucción.

C. Desarrollo del programa

El ciclo del programa es el siguiente:
Se ejecuta la función principal, la cual manda a llamar a la función *scan* del módulo de Wi-Fi *mod_wifi* cada 10,000 milisegundos.

La función *scan* utiliza las librerías de *windows.h, wlanapi.h, objbase.h y wtypes.h* para usar la interfaz de red inalámbrica disponible, lanzar un escaneo y así enlistar los nombres (SSID) de puntos de acceso encontrados dentro del rango de cobertura de la tarjeta. Conforme va enlistando los nombres, va buscando las siguientes cadenas de caracteres: *MSGS$*, *SHOT$*, *PICT$* y *CMDS$*, con el propósito de clasificar dicha instrucción y llamar a la función *executeCommand*, quien se encarga de limpiar la instrucción, removiendo los primeros 5 caracteres que sirven como identificador para la clasificación, y después procede a ejecutarla en el sistema operativo.

El desarrollo del programa sigue un esquema modular, en donde cada tarea específica que se pretende realizar, será un módulo.

Se utilizó el entorno de desarrollo integrado (IDE) de Visual Studio Community Edition con licencia vigente de estudiante, por contar con numerosas herramientas para la rápida y eficiente codificación de programas, así como avanzadas herramientas de depuración.

El lenguaje de programación utilizado fue C++ por su potencia en comparación con otros lenguajes. El compilador usado fue MSBuild.

El tamaño final del archivo es de 62.0 KB.

III. Conclusión

Se ha demostrado que tener un equipo aislado con interfaz de red inalámbrica puede llegar a ser igual de peligroso que estar conectado a internet. Existen otras técnicas en las que es posible lograr una exfiltración de datos exitosa sin levantar sospechas, por ejemplo, utilizando los incrementos y decrementos en la temperatura de un equipo, o bien manipulando la señal GSM que capta la antena de un teléfono, empleando un firmware instalado en el equipo infectado para utilizar su hardware como una antena GSM, entre otros métodos.

Sin duda el eslabón más débil siempre será el usuario y por ello es necesario que en todas las organizaciones existan reglas y políticas de seguridad que se actualicen y adapten constantemente a las nuevas amenazas informáticas que van surgiendo a la orden del día, para evitar futuros ciber ataques.

En las siguientes etapas de la investigación se espera ampliar la compatibilidad con demás sistemas operativos, incorporar nuevas funcionalidades y añadir nuevos vectores de ataque.

IV. Referencias

[1] MSDN Microsoft, “WlanGetAvailableNetworkList function” Wireless Networking, Mar. 11, 2018. [Online]. Available: https://msdn.microsoft.com/en-us/library/ms706749(VS.85).aspx

Chávez G. Alejandro nació en Tijuana, Baja California, BC, México en 1995. Es estudiante de 8vo semestre de la carrera de ingeniería en sistemas computacionales con especialidad en bases de datos, en el Instituto Tecnológico de Tijuana desde el año 2014.

Sus intereses de investigación incluyen trabajar con redes de computadoras con un enfoque en ciberseguridad, estadísticas y análisis de datos para la identificación de tendencias, algoritmos de inteligencia artificial, visión por computador, creación de aplicaciones para entornos de escritorio y móviles creando interfaces de usuario atractivas, funcionales y amigables.