PRUEBA EXPERIMENTAL DE PENETRACIÓN USANDO UNA SECUENCIA DE COMANDOS CON USB RUBBER DUCKY

Mtro. Flores Montaño Luis Alberto
luisfloresmontano@hotmail.com
M. en C. Esther Viridiana Vázquez Carmona
evazquezc1801@alumno.ipn.mx
M. en C. Rodrigo Vázquez López
Email: rvazquezl1800@alumno.ipn.mX
Dr. Jacobo Sandoval Gutiérrez
j.sandoval@correo.ler.uam.mx

Instituto Politécnico Nacional
Centro de Innovación y Desarrollo Tecnológico en Cómputo
Universidad Autónoma Metropolitana Unidad Lerma
Departamento de Procesos Productivos

Resumen

Este artículo presenta la implementación de un ataque de penetración a un equipo con un sistema operativo Windows, mediante el uso de la herramienta de hackeo ético conocida como “USB Rubber Ducky” desarrollada por la empresa Hak5; esta tiene apariencia de USB para ejecutar el ataque de una forma discreta; cabe resaltar que esta solo funciona insertando a el Rubber Ducky directamente a la máquina huésped; no obstante, esta no es detectable en la barra de tareas del sistema operativo, como una memoria flash; más bien funciona como un teclado HID, lo que permite realizar una secuencia de comandos; los cuales son ejecutados en la máquina huésped, permitiendo generar un ataque físico o de inyección, dicho ataque permite vulnerar un equipo ya que puede recuperar datos confidenciales, como poder ser el caso de una identificación de usuario o contraseñas que pueden ser visualizadas en texto claro. Adicionalmente, Rubber Ducky puede mejorar su empeño si se hace uso de otras herramientas como es el caso de Mimikats, tecnología PHP, powershell, lenguaje de escritura (scripting) y servidores web.

Palabras Clave: Ciberataque, Ciberseguridad, Rubber Ducky, HID, Codificador.

Abstract

This article presents the implementation of a penetration attack on a computer with a Windows operating system, through the use of the ethical hacking tool known as ÜSB Rubber Ducky”developed by the company Hak5; this has the appearance of a USB to execute the attack in a discreet way; It should be noted that this only works by inserting the Rubber Ducky directly into the host machine; however, it is not detectable in the operating system's taskbar, like a flash drive; rather it works like a HID keyboard, allowing scripting; which are executed on the host machine, allowing a physical or injection attack to be generated, this attack allows a computer to be compromised since it can recover confidential data, such as the case of a user identification or passwords that can be displayed in clear text . Additionally, Rubber Ducky can improve its performance if other tools are used, such as Mimikats, PHP technology, powershell, scripting language and web servers.

Keywords: Cyber attack, Cybersecurity,Rubber Ducky, HID, Decode.

I. Introducción

Las computadoras personales, incluyendo las portátiles, las de escritorio, así como dispositivos móviles teniendo en cuenta teléfonos inteligentes y tabletas, tienen en común algo, la entrada de información a través de dispositivos periféricos o en su defecto digital; en otras palabras, reciben información desde teclado. Los teclados estándar USB que forman parte de los dispositivos de interfaz humana (HID), suelen ser aceptados por una amplia gama de dispositivos, así como de sistemas operativos conocidos, es decir, plug and play.

Por otro lado, las interfaces de unidades USB generalmente suelen ser peligrosas, debido al potencial amenaza como una herramienta de piratería. Acorde con los autores (Falliere, 2011) y (Walter, 2012); los usos de almacenamiento USB, pueden servir como mecanismos de programa maligno o "malware"; así recientemente un ciberataque basado en USB, conocido como BadUSB, consta en registrar diversos dispositivos y realizar acciones de encubierta en la computadora personal (máquina huésped). Un ejemplo de esto son las unidades flash USB que pueden registrar un dispositivo o teclado lo que permite la capacidad de inyectar scripts maliciosos.

Dicha funcionalidad puede ser realizada mediante la herramienta desarrollada por la empresa Hak5, esta lleva por nombre como "USB Rubber Ducky", o también conocido como "Patito de hule". Cabe mencionar, que debido al firmware de esta herramienta el dispositivo de ataque USB no puede ser detectado por el antivirus, por lo que lo imposibilita de protegerse ante dicho ataque.

Acorde con (Tian, 2015), los ataques no solo se limitan a las memorias o teclados que hacen uso de USB, este tipo de problemas o ataques se pueden realizar con cualquier dispositivo que utilice un puerto USB, por lo que, cualquier equipo puede llegar a ser susceptible. Adicionalmente los dispositivos estándar USB son demasiados sencillos o simples para funcionar de manera confiable, por lo que autenticar o asegurar estos dispositivos suele ser muy raro en ocasiones, y por lo tanto defenderse de este tipo de ataques deja grandes expectativas de seguridad.

Existen diversos métodos para penetrar una máquina como el caso de un probador de penetración, explotando vulnerabilidades del sistema o incluso por ingeniería social. Sin embargo, la estrategia de usar un dispositivo USB, y ejecutar un código malicioso sin el conocimiento de la víctima, llega a ser ventajoso. Teniendo como ejemplo que la víctima tiene un descuido y deja su equipo de trabajo por unos momentos, el pirata informático o atacante, podría conectar USB, y ejecutar códigos maliciosos para la máquina huésped. Es importante mencionar que hay autores como (Voutema, 2015), donde explica que pueden realizar este tipo de ataques con las herramientas del tipo "BadUSB" y "RubberDucky", haciendo uso de otros dispositivos complementarios como un Micro Arduino, utilizado como reemplazo de estos últimos.

II Rubber Ducky una USB para el registro de teclas

El USB Rubber Ducky es una herramienta desarrollada por Hak5 (Hak5, 2013). Este dispositivo incluye un microcontrolador, el cual puede ser programable. Otra característica del dispositivo es el comportamiento de teclado y la apariencia de una unidad flash USB; por lo que es fácil de ocultar en un puerto de un equipo de cómputo, por otro lado, también puede estar oculto en el administrador de tareas.

No obstante, para realizar un ataque usando el Rubber Ducky se necesita tener acceso físico a la máquina víctima, así como la programación de un "script" del malware a ser inyectado en el dispositivo.

Como se mencionó anteriormente los equipos de cómputo confían inherentemente en los dispositivos que forman parte del HID (como puede ser un teclado), ya que estos logran la interacción y comunicación máquina-humano, para las tareas diarias. Partiendo de lo anterior, el USB Rubber Ducky funciona como un emulador de teclado disfrazado dentro de un dispositivo muy parecido al de una memoria USB. Es importante resaltar que esta herramienta ha sido utilizada por diversos profesionales de TI, probadores de penetración (en inglés pentesters), y piratas informáticos desde hace más de 10 años, por lo que dicha herramienta se ha convertido en una plataforma comercial muy utilizada, para los ataques de inyección, haciendo uso de pulsaciones de teclas en el sistema. Esto combinado con lenguaje de secuencias de comandos, cargas útiles (payloads) las cuales pueden ser escritas y desarrolladas como el autor del ataque lo deseé. En la figura 1 se muestra la herramienta del Rubber

Ducky de Hak5, la cual como ya se ha mencionado tiene una imagen muy parecida a una unidad flash USB.

Figura 1. USB Rubber Ducky desarrollada por Hak5.

Adicionalmente, es común que los usuarios dejen desatendidos sus equipos de cómputo, teniendo la oportunidad de colocar a esta herramienta e implementar comandos maliciosos de manera automático, haciendo uso únicamente de pocos minutos para este tipo de acción.

En este caso la investigación es realizada, usando la herramienta de Rubber Ducky y un equipo de cómputo víctima, el cual tendrá un sistema operativo Windows 11.

III. Tecnologías y herramientas utilizadas

Para desarrollar este tipo de ataque de inyección es necesario el uso de diversas herramientas, en este caso puede ser de software y de hardware, por lo que durante esta sección se analizan diversas herramientas y tecnologías que se utilizan para el ataque.

Intrusión a la máquina huésped

Para la realización del ataque por inyección, es necesario tener un enfoque de la máquina objetivo, en la cual se hará el ataque por la herramienta. Dicha máquina es una computadora Huawei con Windows 11, 64 bits, y teniendo como antivirus Bitdefender.

Hardware Rubber Ducky

Se hace uso de la herramienta USB Rubber Ducky, la cual es conectada a la máquina huésped o máquina víctima. Cabe destacar que dicha herramienta contiene un microcontrolador programable de 60 MHz, así como una ranura SD y como ya se ha mencionado una característica importante de esta es el comportamiento que tiene, muy parecido al de un teclado común, y, por otro lado, esta no es mostrada en el administrador de tareas, por lo que llega ser no detectable para la máquina víctima.

Lenguaje de secuencias de comandos

El lenguaje de comando se utiliza principalmente para escribir las cargas útiles de programa malicioso, que también son conocidos como “malware payload”, para realizar esto se usa algunas secuencias de comandos del lenguaje de Rubber Ducky. Este tipo de escritura puede hacerse desde cualquier editor de textos, como puede ser el caso de un bloc de notas. Cabe mencionar que al escribirse cada comando debe ir en mayúscula y en una nueva línea, adicionalmente se pueden invocar pulsaciones de teclas, combinaciones de estas o cadenas de texto para ofrecer retrasos o pausas.

Algunos de los comandos más comunes son DELAY y STRING. DELAY es seguido de un número que representa milisegundos. Un ejemplo de esto es que, si escribimos “DELAY 3000”, significa que en la secuencia de comandos tendrá una pausa de 3 segundos, ante de que pase a la siguiente secuencia. Esto es necesario para asegurarse que la secuencia de comandos funciona correctamente y sin problemas, adicionalmente funciona para llevar la secuencia de comandos del Rubber Ducky, a la computadora huésped, ya que algunas no están actualizadas o son más lentas.

Posteriormente, el comando STRING, indica el procesamiento del texto, en este caso puede aceptar a uno o más caracteres. No obstante, el comando WINDOWS (o GUI) puede emular el botón de Windows. En la figura 2, se muestra un ejemplo de un código sencillo que muestra en un bloc de notas un mensaje

de “Hola estoy en tu máquina”, esta secuencia de comando se realiza en cuestión de milisegundos un momento que se conecta el Rubber Ducky.

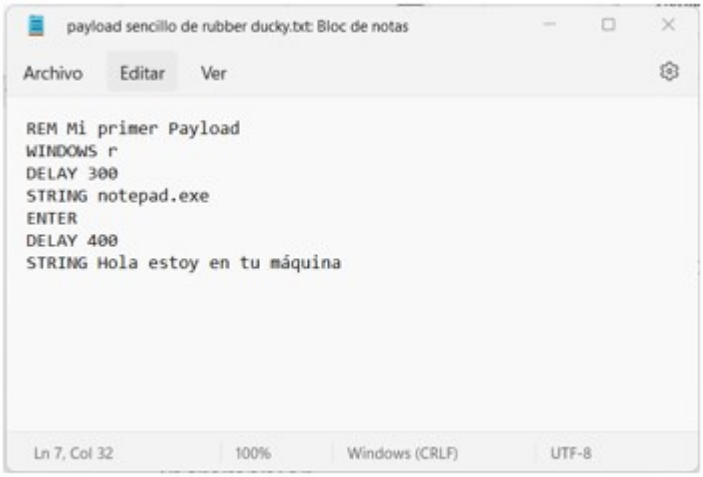

Figura 2. Ejemplo de una secuencia de comandos para Rubber Ducky

Kit de herramientas Duck NG

También conocida como Duck Toolkit NG, es una prueba de penetración la cual es una plataforma de código abierto, que se puede encontrar en línea en la página https://www.ducktoolkit.com/; esta permite a los desarrolladores generar carga útiles o payloads para Rubber Ducky, esto para usarse en Windows, Mac OSX, Linux entre otros sistemas operativos. Por lo que esta herramienta puede ayudar a preconstruir cargas útiles o payloads y decodificar algunas existentes. Es importante mencionar que este kit necesita de acceso administrativo, acceso a internet y powershell, un lenguaje de programación orientado a objetos con líneas de comando interactivas para Microsoft Windows.

Codificador

Hay que tomar en cuenta que la secuencia de comando del Rubber Ducky es un formato simple que puede ser legible por humanos por lo que se puede compartir y modificar fácilmente; sin embargo, esto no puede ser procesado por el USB Rubber Ducky, por lo que para su entendimiento se debe derivar de un archivo inject.bin a través de un codificador. Cabe mencionar que existen diversos codificadores de código abierto para la secuencia de comandos de Rubber Ducky (Ducky Script); acorde con la empresa de Hak5, recomienda hacer el uso del codificador oficial que lleva por nombre JavaScript Ducky de Hak5, el cual puede codificar en línea desde cualquier navegador moderno o sin conexión, descargando su software.

Como ya se mencionó la herramienta JS Ducky Encoder, puede generar un archivo ejecutable en Rubber Ducky, para poder realizar esto se descarga dicha herramienta de downloads.hak5.org/ducky; posteriormente se abre el jseconder.html en un navegador compatible con Javascript; posteriormente que se ingresa en el Ducky Script en el área de texto principal, y se hace clic en “Generate Payload” y finalmente se da clic a “Download Payload”. En la figura 3 se muestra un ejemplo del codificador “Ducky Encoder”.

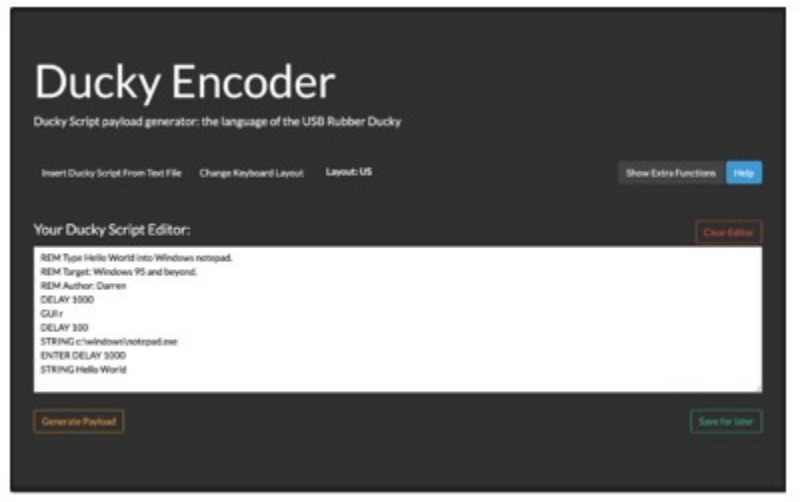

Figura 3. Interfaz del codificador Ducky Encoder

Prueba

Una vez que el Ducky Script se codifica en un archivo inject.bin, se prueba el “Payload”. Se copia el archivo inject.bin en la raíz de la tarjeta Micro SD, donde posteriormente se va a insertar en el USB Rubber Ducky.

Una vez haciendo esto, se tiene que insertar a una máquina huésped/víctima, una vez conectada el Rubber Ducky, toma un momento en enumerar los comando como un teclado HID y cargar los controladores genéricos, por lo que es recomendable tener un retraso de al menos de un segundo en el código; en dado caso que no tenga éxito es posiblemente porque no se ha cargado correctamente los controladores de teclados genéricos.

VI. Conclusiones

En este proyecto se ha mostrado un procedimiento para utilizar el Rubber Ducky, así como un ejemplo sencillo para los comando de secuencias; por otro lado, se conoció la vulnerabilidad que puede tener Windows o en su defecto cualquier otro sistema operativo, debido a que la herramienta entra a la máquina huésped como un teclado HID, con la imposibilidad de ser detectado como una amenaza. Por último, cabe mencionar que se mostró un ejemplo sencillo de “payload” para que pueda ser ejecutado; sin embargo, existen diversos repositorios donde se puede encontrar más “payloads”; adicionalmente, se pueden utilizar otras herramientas junto con Rubber Ducky, como es el caso de Powershell, Mimikatz, un servidor web o PHP; en conjunto con esto se puede explotar más las vulnerabilidades de sistemas operativos. Teniendo en cuenta que solo se necesitan unos segundos para robar de manera discreta y confidencial la información de un equipo.

Referencias

1. Al-Zarouni, M. (**2006**). *The reality of risks from consented use of USB devices.*

2. Arduino Micro (**2015**). http://arduino.cc/en/ Main/ArduinoBoardMicro

3. Bhakte, R., Zavarsky, P., & Butakov, S. (**2016, June**). *Security Controls for Monitored Use of USB Devices Based on the NIST Risk Management Framework In 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 461-466). IEEE.

4. Griscioli, F., Pizzonia, M., & Sacchetti, M. (**2016, December).**). *USBCheckIn: Preventing BadUSB attacks by forcing human-device interaction. In 2016 14th Annual Conference on Privacy, Security and Trust (PST)* (pp. 493-496). IEEE.

5. Falliere, N., Murchu, L. O., & Chien, E. (**2011**). *W32. stuxnet dossier. White paper, symantec corp., security response,* 5(6), 29.

6. Hak5. (**2013**). *Episode 709: USB Rubber Ducky Part 1* http://hak5.org/episodes/episode-709,

7. Hak5. *USB Rubber Ducky* https://hakshop.myshopify.com/products/usb-rubber-ducky-deluxe

8. Hak5. *Hak5/usbrubberducky-payloads* https://github.com/hak5/usbrubberducky-payloads

9. *Payloads Rubber Ducky.* https://www.jesusninoc.com/07/03/payloads-rubber-ducky/

10. *Payload Generator.* https://www.ducktoolkit.com/payload/windows

11. Tian, D. J., Bates, A., & Butler, K. (**2015, December**). *Defending against malicious USB firmware with GoodUSB. In Proceedings of the 31st Annual Computer Security Applications Conference* (pp. 261-270).

12. Vouteva, S., Verbij, R., & Roos, J. (**2015**). *Feasibility and Deployment of Bad USB. University of Amsterdam, System and Network Engineering Master Research Project.*