3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

Recepción: 23 de enero de 2017
Aceptación: 10 de marzo de 2017
Publicación: 14 de marzo de 2017

¿POR QUÉ DESCONFIAR DE LOS DISPOSITIVOS DE ALMACENAMIENTO DIGITAL?

WHY YOU SHOULD NOT TRUST DATA STORAGE DEVICES?

Romel Vera Cadena[1]

1. Consultor de Seguridad Informática. (Ecuador). E-mail: romel.vera.cadena@gmail.com

Citación sugerida:

Vera Cadena, R. (2017). ¿Por qué desconfiar de los dispositivos de almacenamiento digital?. *3C Tecnología: glosas de innovación aplicadas a la pyme*, 6(1), 35-46. DOI: <http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46/>.

3C Tecnología *(Edición 21)* Vol.6 – Nº 1
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

RESUMEN

Los dispositivos de almacenamiento digital pueden no ser tan seguros como se pensaba. Mediante un pendrive se puede pasar *malware* al computador y esto sucede por *software* o *hardware*, siendo éste último abusado por fallas de diseño en la implementación del USB para pasar desapercibido por *software* de seguridad debido a que no existen parches para este tipo de problemas.

ABSTRACT

Data storage devices may not be as safe as previously thought. A pendrive can be used to infect a computer with malware and this can be achieved by software or hardware, the latter being abused by design flaws in the implementation of USB, and cannot be detected by security software because there are no patches for this type of problem.

PALABRAS CLAVE

Malware, Pendrive, USB, Seguridad Informática, badUSB.

KEY WORDS

Malware, Pendrive, USB, Information Security, badUSB.

3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

1. INTRODUCCIÓN

Los usuarios de computadoras intercambian *pendrives* regularmente para poder transmitir información relevante en una empresa o para algo trivial en el hogar. La mayoría conoce que por el uso constante de estos dispositivos se van a encontrar *malwares* y normalmente depende del antivirus evitar que el *malware* infecte al equipo, pero los problemas de seguridad en éstos dispositivos de almacenamiento portátiles no son tan sencillos como se pensaba (Tischer, 2016).

Varios especialistas de seguridad como Karsten Nohl y Jakob Lell han demostrado que los dispositivos USB pueden ser usados para infectar un computador con *malware* sin ser detectado y éste *malware* podría interceptar el uso de internet del usuario o instalar *software* no deseado en el equipo. Esto ocurre porque el problema no radica en la memoria del dispositivo o del computador sino en el *firmware* que controla las funciones básicas del dispositivo USB. Por ejemplo, un dispositivo USB como en el caso de un pendrive contiene un pequeño computador y un *firmware* que es esencialmente un *software* especial o un pequeño sistema operativo que hace que el dispositivo funcione correctamente (Nohl, 2014).

Este *firmware* funciona a bajo nivel y es usualmente programado de fábrica y al momento en que se adquiere un *pendrive* básicamente se estaría haciendo un acto de fe porque se confía que el *firmware* del *pendrive* funciona tal y como fue diseñado por el fabricante (Greenberg, 2014).

En muchos de los dispositivos USB los *firmwares* pueden ser reprogramados al momento de ser insertados en un computador desconocido o de poca confianza. Esto es posible porque algunos de los fabricantes no diseñaron seguridades que eviten la reprogramación del *firmware* (Han, 2016) (Schumilo, 2014).

Como consecuencia, estos dispositivos podrían ser usados para transmitir *malware* o realizar bromas pesadas, como transmitir virus o troyanos, convertir un teclado en un teclado espía haciendo uso de un registro interno de lo que se escribe, enviar imágenes de una webcam a otro computador, etc. (Nasution, 2014) (Stoffregen, 2014).

El presente artículo tiene como propósito concientizar a las personas en cuanto al riesgo que existe cuando se usan dispositivos USB de desconocidos, sean de almacenamiento o no y también demostrar mediante un experimento de laboratorio cómo se construye un dispositivo que pudiese presentarse como un *pendrive,* pero que en realidad funciona como un teclado y que realice acciones en el ordenador.

2. METODOLOGÍA

Desarrollando el laboratorio práctico se contestará la siguiente pregunta:

- ¿Es posible burlar software de seguridad utilizando dispositivos de almacenamiento modificados?

Para contestar esta interrogante se realizará un laboratorio práctico en el que se construirá un dispositivo USB modificado y, posteriormente, se ejecutarán pruebas en 10 computadoras diferentes que tengan instalado software de seguridad con la finalidad de poder reflejar los resultados obtenidos en una tabla.

3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

Para corroborar el éxito de la prueba, el dispositivo USB deberá realizar las siguientes acciones visibles en el computador:

1. Esperar 11 segundos después de que él dispositivo haya sido conectado al computador
2. Abrir la ventana "ejecutar" del Windows
3. Abrir el editor de texto del Windows (Notepad)
4. Escribir 2 líneas en el Notepad:
    - "Prueba de dispositivo Teensy."
    - "Completa!"

El sistema operativo seleccionado para realizar éstas pruebas es Windows 10 de Microsoft por tratarse de un sistema operativo en el mercado con un mayor porcentaje de adopción (SpiceWorks, 2016).

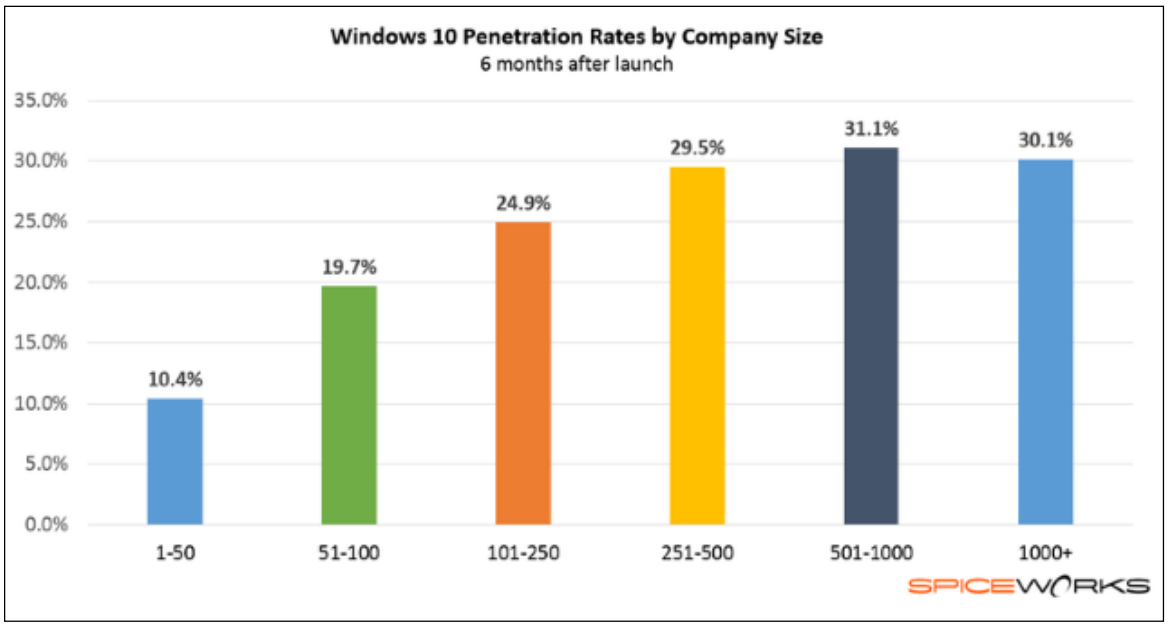

Figura 1. Porcentaje de adopción de Windows 10 en compañías después de 6 meses de su lanzamiento.
Fuente: Spiceworks 2016.

Y para la selección del software de seguridad se han tenido en cuenta las evaluaciones del mejor software de seguridad de AVTEST del mes de diciembre del 2016 (AVTEST, 2017).

2.1. MATERIALES

1. Teensy++ 2.0

El material principal es un hardware que básicamente es una placa de circuito impreso con un microcontrolador programable y en este caso será un Atmel AVR.

Si posee conocimientos básicos de electrónica puede comprar los componentes individualmente en:

https://www.sparkfun.com

https://www.adafruit.com

3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

O puede comprar la unidad lista para usar en:

https://www.pjrc.com

En el caso de comprar un Teensy es aconsejable adquirir cualquier unidad desde la versión 2.0 en adelante.

Figura 2. Teensy++ 2.0
Fuente: PJRC

Teensy++ 2.0 es una placa de desarrollo USB que viene con un microcontrolador AVR de la firma Atmel AT90USB1286, y se puede programar con el lenguaje Arduino o C.

El Teensy++ 2.0 está diseñado para realizar rápidamente proyectos teniendo poca base en electrónica y programación. Su diseño está pensado para proyectos que se desarrollan en protoboards o en un tablero de pruebas.

2. Cable USB Mini-B

Para conectar el Teensy++ 2.0 al computador es necesario contar con un cable USB Mini-B.

Figura 3. Cable USB Mini-B
Fuente: PJRC

3. Un computador con sistema operativo Windows

2.2. SOFTWARE REQUERIDO

Para poder programar el Teensy++ 2.0 es necesario instalar:

1. Arduino IDE 1.8.1

   https://www.arduino.cc/en/Main/Software

2. Teensyduino 1.35

   https://www.pjrc.com/teensy/td_download.html

3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

Deberá instalarlos en el orden indicado arriba.

El Arduino IDE es un entorno de programación y el Teensyduino contiene librerías para poder usar efectivamente el Teensy.

2.3. LABORATORIO

Para realizar el laboratorio práctico con éxito siga los siguientes 6 pasos:

1. Iniciar el entorno de programación.

   Deberá conectar el Teensy al iniciar el entorno de programación para que el Windows instale los Drivers automáticamente.

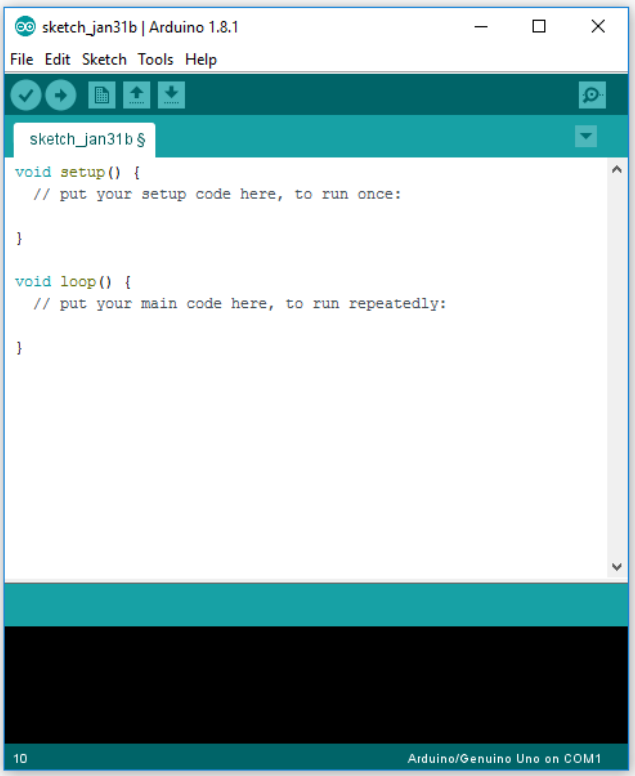

Figura 4. Inicio del software Arduino IDE versión 1.8.1.
Fuente: elaboración propia.

2. Seleccionar el board que se está usando.

   Click en Tools -> Board -> Teensy++ 2.0

3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

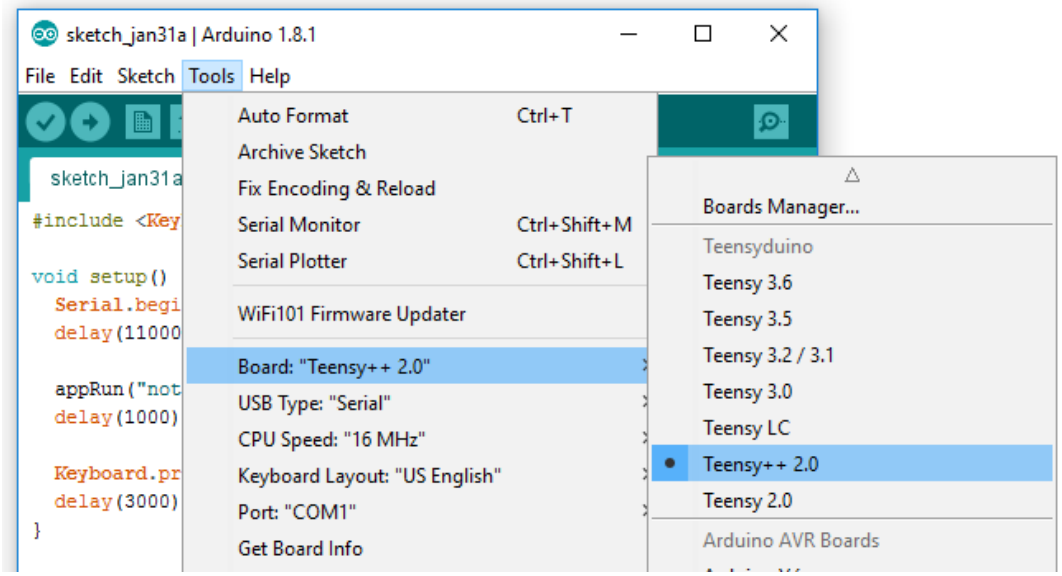

Figura 5. Selección del board Teensy++ 2.0.
Fuente: elaboración propia.

3. Seleccionar el modo de operación.

   Click en Tools -> USB Type -> Keyboard + Mouse + Joystick

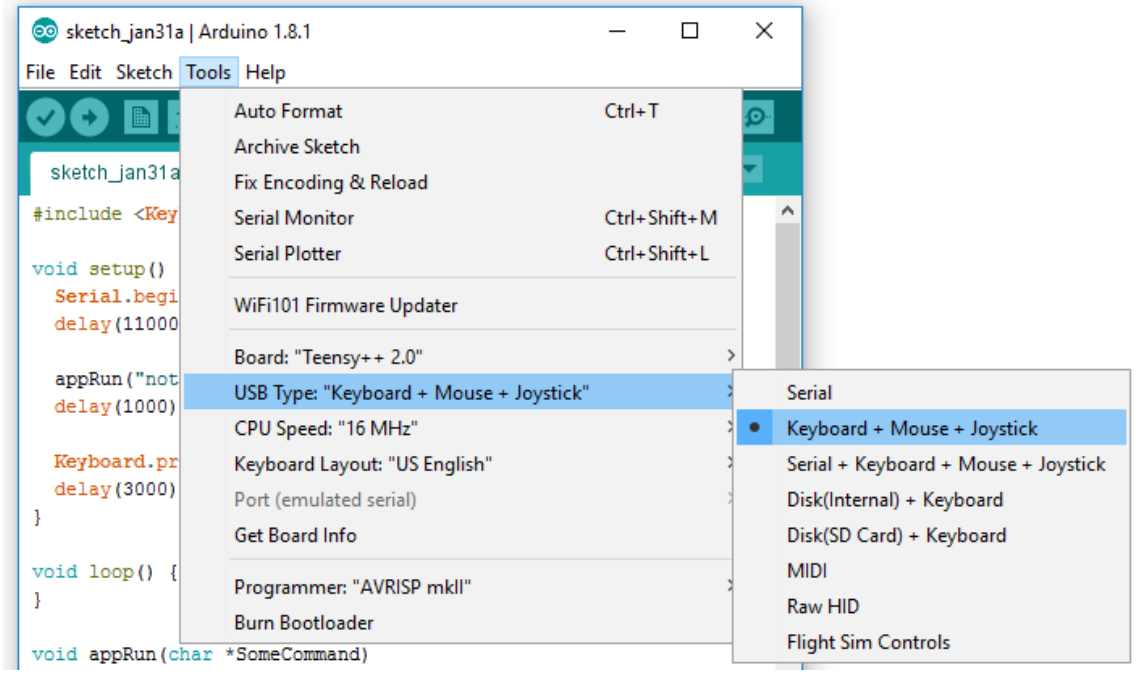

Figura 6. Selección del modo de operación.
Fuente: elaboración propia.

3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

4. Escribir el código en la ventana de texto del entorno de programación:

```
void setup() {
  Serial.begin(9600);
  delay(11000);

  appRun("notepad");
  delay(1000);

  Keyboard.println("Prueba de dispositivo Teensy.");
  delay(1000);
  Keyboard.println("Completa!");
}
void loop() {
}
void appRun(char *SomeCommand)
{
  Keyboard.set_modifier(128);
  Keyboard.set_key1(KEY_R);
  Keyboard.send_now();
  Keyboard.set_modifier(0);
  Keyboard.set_key1(0);
  Keyboard.send_now();
  delay(1500);
  Keyboard.print(SomeCommand);
  Keyboard.set_key1(KEY_ENTER);
  Keyboard.send_now();
  Keyboard.set_key1(0);
  Keyboard.send_now();
}
```

Código 1. Código fuente de las funciones que ejecutará el Teensy++ 2.0.
Fuente: elaboración propia.

5. Compilar el código.

   Click en Sketch -> Verify/Compile

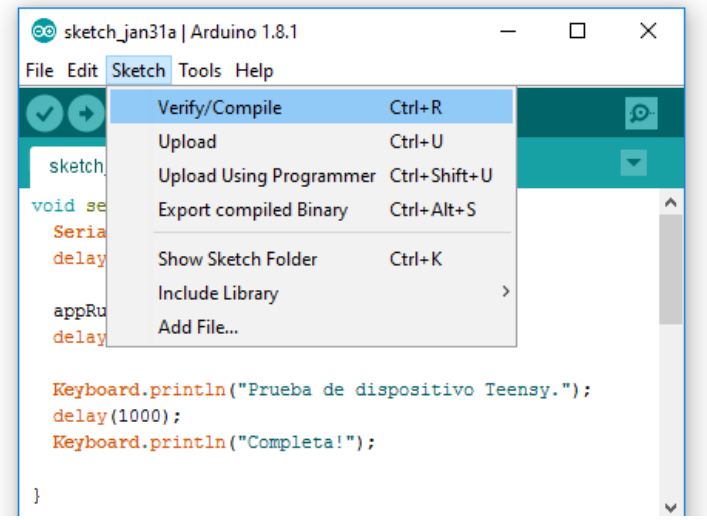

Figura 7. Compilar código fuente.
Fuente: elaboración propia.

3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

6. Subir el código compilado al Teensy.

   Si ha desconectado el Teensy en este paso deberá conectarlo nuevamente para poder continuar.

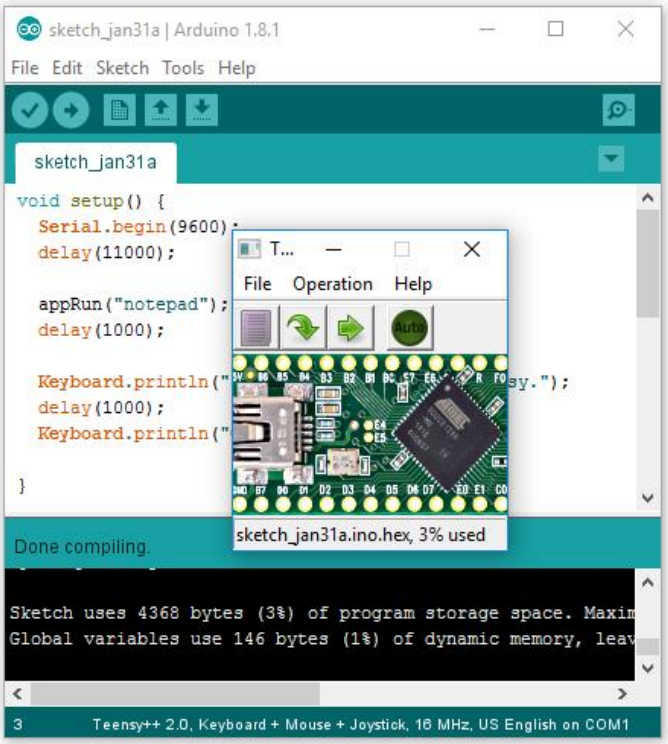

Figura 8. Código compilado y listo para subir al Teensy.
Fuente: elaboración propia.

Para poder subir el código al Teensy es necesario presionar el botón de programación del Teensy.

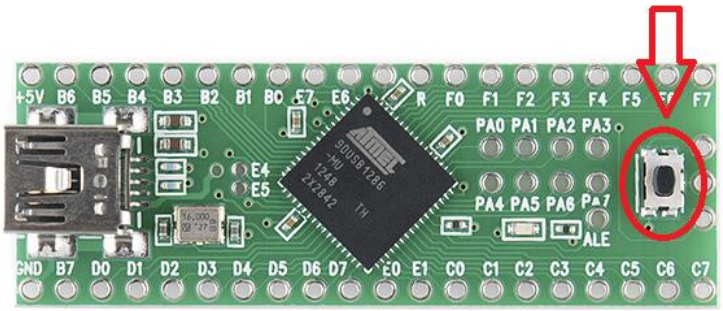

Figura 9. Ubicación del botón de programación del Teensy++ 2.0.
Fuente: elaboración propia, basada en gráfico de PJRC.

Finalmente, el código hará que el Teensy se comporte como un teclado y enviará los comandos programados a la computadora.

Lo puede probar desconectándolo y volviéndolo a conectar al mismo PC, o a otro PC.

3C Tecnología *(Edición 21)* Vol.6 – Nº 1
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

3. RESULTADOS

Se realizaron pruebas a 10 computadoras diferentes que contaban con software de seguridad.

Los *softwares* de seguridad fueron descargados de sus sitios web oficiales y sus versiones corresponden a versiones de pruebas (trial) descargados el día sábado 7 de enero del 2017.

Tabla 1. Resultado de pruebas usando el Teensy++ 2.0.

Prueba	Sistema Operativo	Software de Seguridad	Teensy ejecución de código	Detectado
1	Windows 10 Pro v1607	Windows Defender	100%	No
2	Windows 10 Home v1607	Windows Defender	100%	No
3	Windows 10 Home v1511	ESET Smart Security	100%	No
4	Windows 10 Pro v1607	Kaspersky Total Security	100%	No
5	Windows 10 Pro v1607	Avast Premier	100%	No
6	Windows 10 Home v1511	AVG Ultimate	100%	No
7	Windows 10 Pro v1607	Norton Security Premium	100%	No
8	Windows 10 Pro v1607	Avira Antivirus	100%	No
9	Windows 10 Pro v1607	Panda Global Protection	100%	No
10	Windows 10 Home v1607	McAfee LiveSafe	100%	No

Fuente: Elaboración propia.

En la tabla de resultados se observa que ningún software de seguridad pudo detectar las acciones programadas que realizó el Teensy++ 2.0 en el computador.

El Teensy++ 2.0 realizó todas las acciones visuales que se programaron usando el código fuente del laboratorio.

3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

4. CONCLUSIONES Y RECOMENDACIONES

Al comienzo del presente artículo se indagó en el tema del uso de dispositivos USB tomando en consideración especial los de almacenamiento al tratarse de dispositivos de uso diario para transferir información, y posteriormente se planteó la interrogante "¿Es posible burlar software de seguridad utilizando dispositivos de almacenamiento modificados?".

Los resultados del laboratorio práctico reflejan que sí es posible burlar software de seguridad mediante la programación o reprogramación de dispositivos USB.

Por lo tanto, es posible que un delincuente informático o una persona maliciosa pueda infectar equipos informáticos, interceptar contraseñas que se registren en el teclado, exfiltrar información, etc. logrando causar daños y pérdidas financieras a una empresa o individuo.

Como medidas cautelares se sugiere que las personas se aseguraren de que todos los dispositivos USB que se vayan a conectar en un computador provengan de fuentes fiables.

De sospechar de un dispositivo USB lo recomendable es avisar a un técnico de confianza para que le realice pruebas al dispositivo en un ambiente seguro.

En las empresas es posible realizar pruebas de seguridad usando estos dispositivos que aparentan ser *pendrives* normales con la finalidad de demostrar lo inseguro que es confiar en dispositivos USB desconocidos.

En el hogar se debe educar a los miembros de la familia para evitar ser víctimas de delincuentes informáticos ya que estos dispositivos son cada vez más asequibles, fáciles de usar y fáciles de programar.

5. FUTURAS INVESTIGACIONES

Es posible crear dispositivos más elaborados, por ejemplo:

- Disfrazar el Teensy como un pendrive mediante el uso de una impresora 3D o desmantelando otro dispositivo y adaptando el conector USB tipo A.

Figura 10. Teensy 3.2 con adaptación USB del tipo A.
Fuente: Elie.net.

3C Tecnología ***(Edición 21)*** *Vol.6 – Nº 1*
Marzo – junio'17, 35– 46
Área de Innovación y Desarrollo, S.L.
ISSN: 2254 – 4143
DOI: http://dx.doi.org/10.17993/3ctecno.2017.v6n1e21.35-46

- Se puede analizar otros dispositivos de similar función como el USB Rubber Ducky de Hak5 el cual viene armado, listo para usar y es fácil de programar.

  El USB Rubber Ducky se puede conseguir en:

  https://hakshop.com/products/usb-rubber-ducky-deluxe

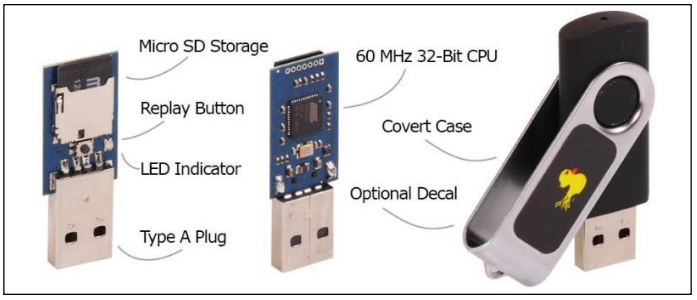

Figura 11. USB Rubber Ducky de la empresa Hak5.
Fuente: Hak5 HakShop.

6. REFERENCIAS BIBLIOGRÁFICAS

AVTEST. (2017). *The best antivirus software*. Retrieved from AVTEST: <https://www.av-test.org/>.

Greenberg, A. (2014). *Why the Security of USB Is Fundamentally Broken*. Retrieved from Wired: <https://www.wired.com/2014/07/usb-security/>.

Han, S. S. (2016). *IRON-HID: Create your own bad USB.*

Nasution, S. M. (2014). Integration of kleptoware as keyboard keylogger for input recorder using teensy USB development board. *IEEE*, 1-5.

Nohl, K. (2014). *BadUSB–On accessories that turn evil.* Black Hat USA.

Schumilo, S. S. (2014). *Don't trust your USB! How to find bugs in USB device drivers.* Blackhat Europe.

SpiceWorks. (2016). *Windows 10 Adoption: Sprinting out the Gate*. Retrieved from SpiceWorks: <https://www.spiceworks.com/it-articles/windows-10-adoption/>.

Stoffregen, P. J. (2014). *Teensy usb development board.*

Tischer, M. D. (2016). Users really do plug in USB drives they find. *IEEE Symposium* (pp. 306-319). IEEE.