

# Challenge: Walkie Hackie

## Challenge Description :

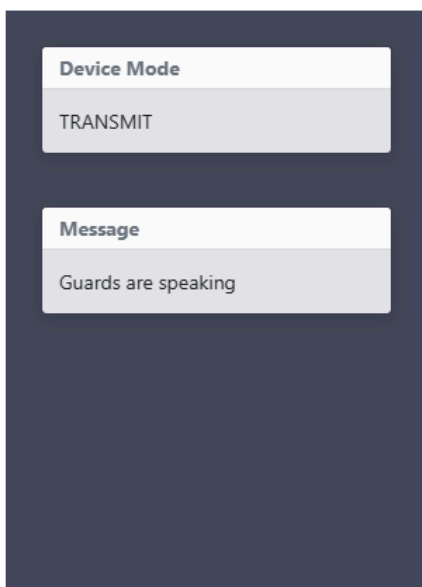
Our agents got caught during a mission and found that the guards are using old walkie-talkies for their communication. The field team captured their transmissions. Can you interrupt their communication to help our agents escape from the guards?

## Context :

- Just Analyze a web-application, You are also given four radiofrequency signal capture files. You must decode these radio transmissions and craft the correct payload to send to the transmit endpoint to get the Flag.

## Flag :

- First Install the software and install the files. Drag the file into Logic2 to start analyzing it and to view the contents of it .



The screenshot shows a web interface for the 'Walkie-Hackie Transceiver'. It has two main sections: 'Device Mode' and 'Message'. The 'Device Mode' section has a 'TRANSMIT' button. The 'Message' section displays the text 'Guards are speaking'.

## Walkie-Hackie Transceiver

### Send RF Signals Over HTTP

Preamble:	<input type="text" value="Ex: AABBCDD"/>
Sync Word:	<input type="text" value="Ex: AABBCDD"/>
Payload:	<input type="text" value="Ex: AABBC"/>
<input type="button" value="Transmit"/>	

- Filtering the radiofrequency signal file to get a form of data we extract the signal pulses as bits. Basically Converting Pulses to a bunch of Ones and Zeros.
- After getting the bits from the radiofrequency signal file we then convert the bits to a hexadecimal format using python.
- I did all this in Kali linux using pre-installed tools to filter out the radio transmissions like Gqrx and inspectrum.

- Each radiofrequency signal capture file gave out a different result after extracting the bits to hexadecimal.
- The format for each radiofrequency signal is made up of three parts:
  - 0xaaaaaaaa 73214693 a1ff14
  - 0xaaaaaaaa 73214693 b1ff57
  - 0xaaaaaaaa 73214693 a2ff84
  - 0xaaaaaaaa 73214693 b2ff24
- To get the flag we will need to enter this data into the website as its similar to its current placeholders,
- Acknowledging this we need to generate the last part of the signal, the Payload.
- Looking at the results we already got we can justify that the middle part will be the same as [ FF ] making this a lot easier. we will only need to brute force the first and last section of the payload.
- Getting a quick one-liner to generate the characters into a function to use with curl, this is also easier to setup with burp-suite.
- Finally getting the Right payload should end up in a 200 Status response, you can also just make a request to get the flag with the right payload if the page doesn't render it.

[ 94.237.57.229:42877/transmit -d 'pa=aaaaaaaa&sw=73214693&pl=00fff9 ]

- The Flag after finding the right payload is : HTB{B4s1c\_r4d10\_fund4s}

## Walkie-Hackie Transceiver

Congrats. HTB{B4s1c\_r4d10\_fund4s}

### Send RF Signals Over HTTP

Preamble:	<input type="text" value="aaaaaaaa"/>
Sync Word:	<input type="text" value="73214693"/>
Payload:	<input type="text" value="00fff9"/>
<input type="button" value="Transmit"/>	

Congrats. HTB{B4s1c\_r4d10\_fund4s}</p></br></p>

RF Signals Over HTTP</h4><br>

form method="post" action="/transmit">

Preamble:</td><td><input type="text" name="pa" placeholder="Ex: AABBCDD" /></td></tr>  
 style="color:white;">Sync Word:</td><td><input type="text" name="sw" placeholder="Ex:

payload:</td><td><input type="text" name="pl" placeholder="Ex: AABBCD" /></td></tr>

colspan="3"><button class="btn btn-outline-success">Transmit</button></td></tr></form>

GENQDDA:~\$ curl 94.237.57.229:42877/transmit -d 'pa=aaaaaaaa&sw=73214693&pl=00fff9'