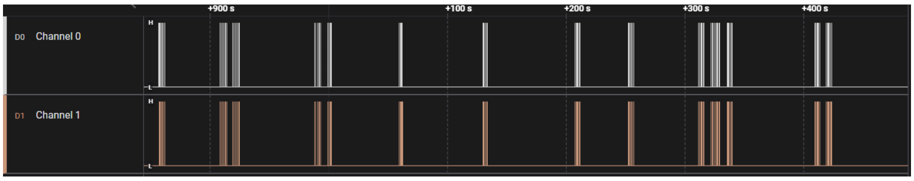
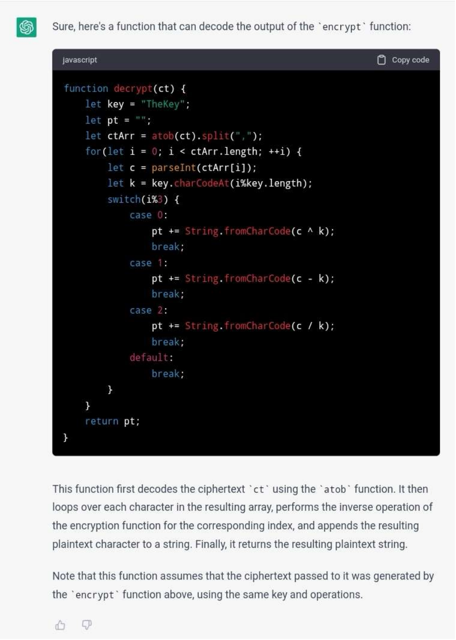
This challenge was a misc / hardware challenge. We needed to get into a ”global company” which was represented by a big wooden box near the admin desk. We were given a website with a page of ”leaks” about the company : https://skillroad.insomnihack.ch, which contained three different leaks that would help us infiltrate this company.

The first page was not accessible directly by a link, but since the second page was indexed as .../items/1338, we easily guessed that the first page was .../items/1337 and we could download the first file !! We could, however, not decode it yet but it looked like this :



On the second page, we could see a commented string in the javascript code. This was a function which encoded a string and the result was given. It was easily visible that the plaintext was easily retrievable so we asked our best friend to help us decode it quickly :)



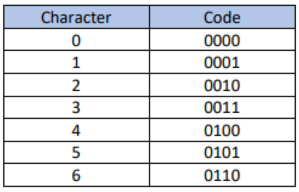
Sure, here's a function that can decode the output of the `encrypt` function:

```javascript
function decrypt(ct) {
    let key = "TheKey";
    let pt = "";
    let ctArr = atob(ct).split(",");
    for(let i = 0; i < ctArr.length; ++i) {
        let c = parseInt(ctArr[i]);
        let k = key.charCodeAt(i%key.length);
        switch(i%3) {
            case 0:
                pt += String.fromCharCode(c ^ k);
                break;
            case 1:
                pt += String.fromCharCode(c - k);
                break;
            case 2:
                pt += String.fromCharCode(c / k);
                break;
            default:
                break;
        }
    }
    return pt;
}
```

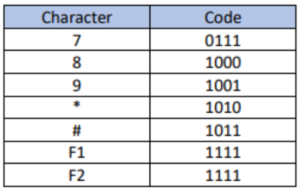
This function first decodes the ciphertext `ct` using the `atob` function. It then loops over each character in the resulting array, performs the inverse operation of the encryption function for the corresponding index, and appends the resulting plaintext character to a string. Finally, it returns the resulting plaintext string.

Note that this function assumes that the ciphertext passed to it was generated by the `encrypt` function above, using the same key and operations.

It was a pdf which explained how to use the first file to get the digicode of the box. The data was encoded in block of four bits according to this table :



Character	Code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110



Character	Code
7	0111
8	1000
9	1001
*	1010
#	1011
F1	1111
F2	1111

After some tries, we figured out the digicode, which was 215968#. We had the digicode to open the first part of the box !! We opened the box and found a NFC tag and a wifi name (QUEST GUEST) + password.

Dear visitors,

Here is the guest wireless access and the visitor badge.

(ONE BADGE PER TEAM!)

There was still one leak to access on the website, so we dug in and found nothing for a while. After many, many tries, we realised that we simply could make a small very basic SQL injection to access the file: ' OR 1=1, – This gave us access to a json file describing an api on the wifi we were connected to. After examining the json file, we figured out that we had to make a request to te api with giving a special id which was the one of the technician while holding the NFC tag on the box, which wrote some data on the NFC tag. When decoded, there was a parameter on the tag which was : admin=0. We changed it to admin=1 and we were able to open the second part of the box !! And a new WIFI appeared : INTERNAL QUEST.

With nmap, we were able to see that the port 435 was open on a certain IP in this new wifi, which corresponds to SMB communications. When connecting to the IP, a login was required which was admin, admin and we could download a kdbx file. After some tries, the master password was found to be rockyou and we could access the flag and the location of the treasure !! *INS*{*W3llDon3!Y0uCompl3tedTh3Expl0itQuetn*(*ovo*)/}