Challenge Information

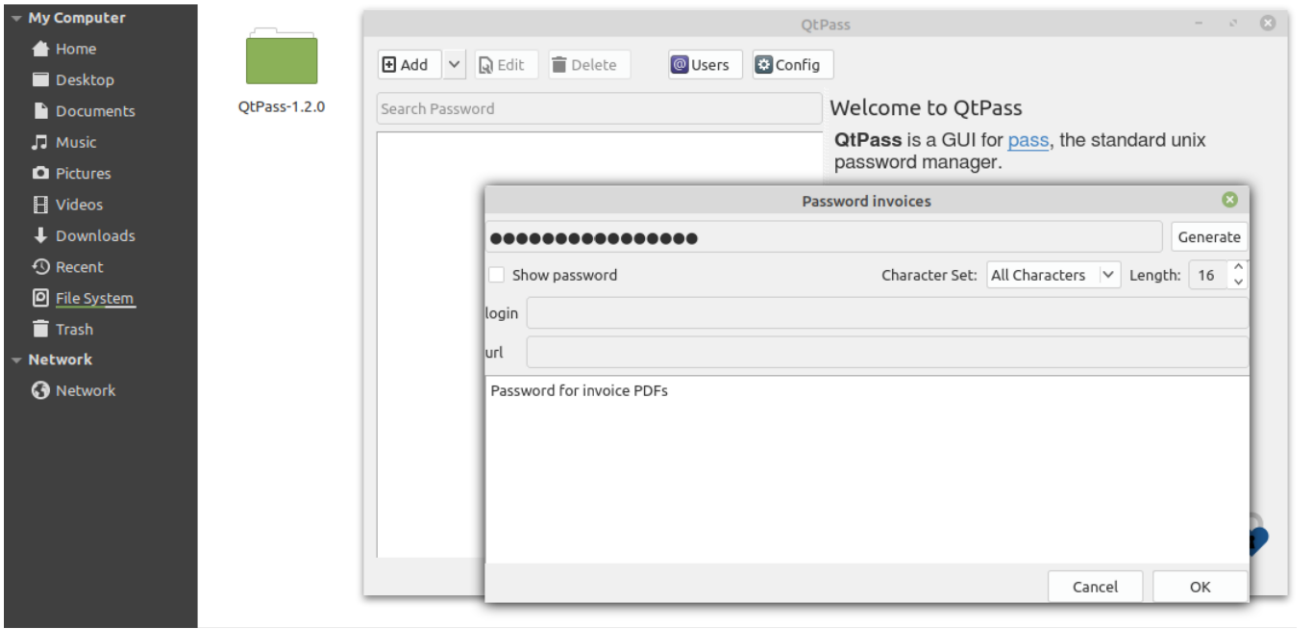
Who knew invoices could be cute?

Files Supplied:

https://portal.hackazon.org/files/0bd27e34beb174089ff0aade9af1119445bac3bc/invoice.pdf

https://portal.hackazon.org/files/1a8158704962f9633af4fdf59af0b27f57834313/invoice.png

Information Gathering from Invoice.png



Obeserved the follow:

Linux OS version is **Linux Mint**
QTPass is being used
QTPass is version **1.2.0**
Password is **16 Characters** and using **All-Characters** option

Flag 1:

Description: Cute Invoice

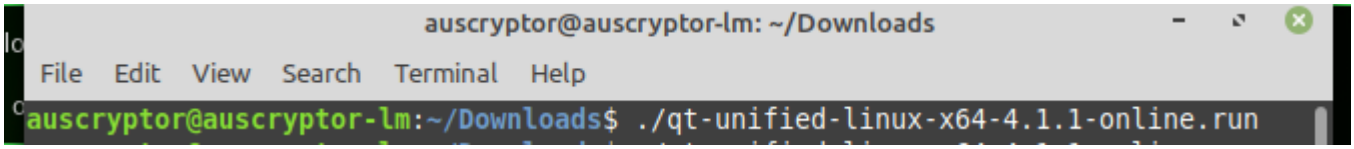
[200 points] Cute Invoice

Who knew invoices could be secure AND cute? Our third-party contractor for space shuttle parts is using the best tooling for sending us secure invoices.

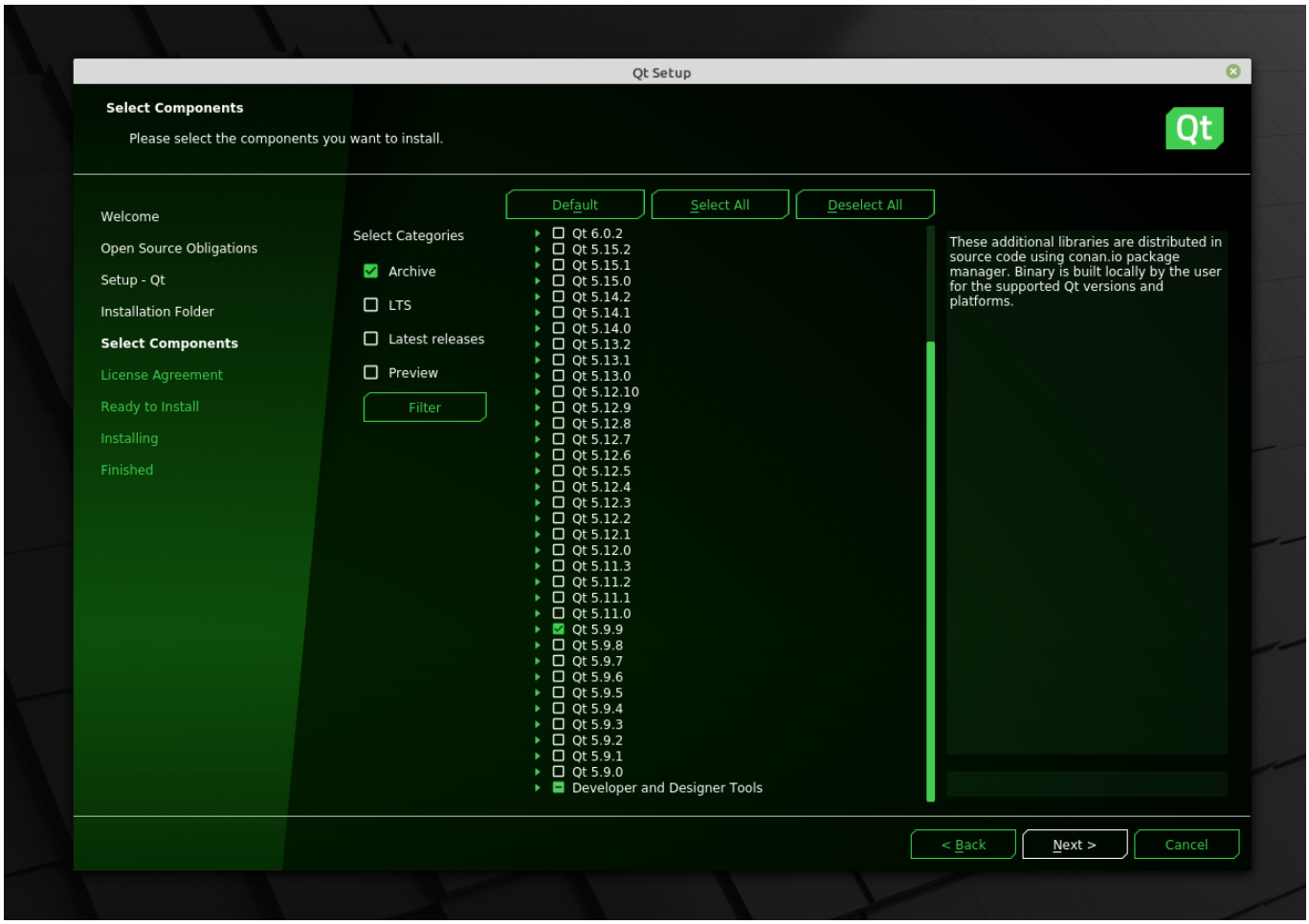
Download and Install QT IDE

Visit https://www.qt.io/ and Download QT https://www.qt.io/download (I downloaded Try QT)

Once downloaded you then need to go through the install process.



You will need to select version 5.9.9 (You will need to used the filter option on the left with archive ticked)



installation will take a while

While qt ide was installed I did some googling to find if there was any known vunerabiliities with QTPass 1.2.0 , as it turns out there is an issue with the random password generator function.

https://github.com/IJHack/QtPass/issues/338

Based on this above info, my next thought was to replicate the generate password function with setting the rand seed manually from 0 to 999

Here is the code i came up with:

```
#include <QCoreApplication>
#include <QLocale>
#include <QTranslator>
#include <QtGlobal>
#include <iostream>
using namespace std;
static const char
charset[]="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890~!@#$%&*()_-+={}
[]|:;<>,.?^";

int charsetLen=sizeof(charset)-1;
int main()
{
    for(int i=0; i<=999; i++)
    {
        qsrand(i);
        string password = "";
        for(int l=0; l<=15; l++)
        {
            password += (char) charset[qrand() % charsetLen];
        }

        cout<<password<<endl;
    }

    return 0;
}
```

I then used the output from the above code and copied and pasted it into a passwords.txt

Once i had my list of possible passwords it was time to see if i could crack the pdf password.

If you dont have the source code of JohnTheRipper (git clone https://github.com/magnumripper/JohnTheRipper.git)

Use pdf2john.pl to extract the hash of the provided pdf

Command→

```
./pdf2john.pl /home/kali/cute_invoice/invoice.pdf > /home/kali/cute_invoice/invoice.hash
```

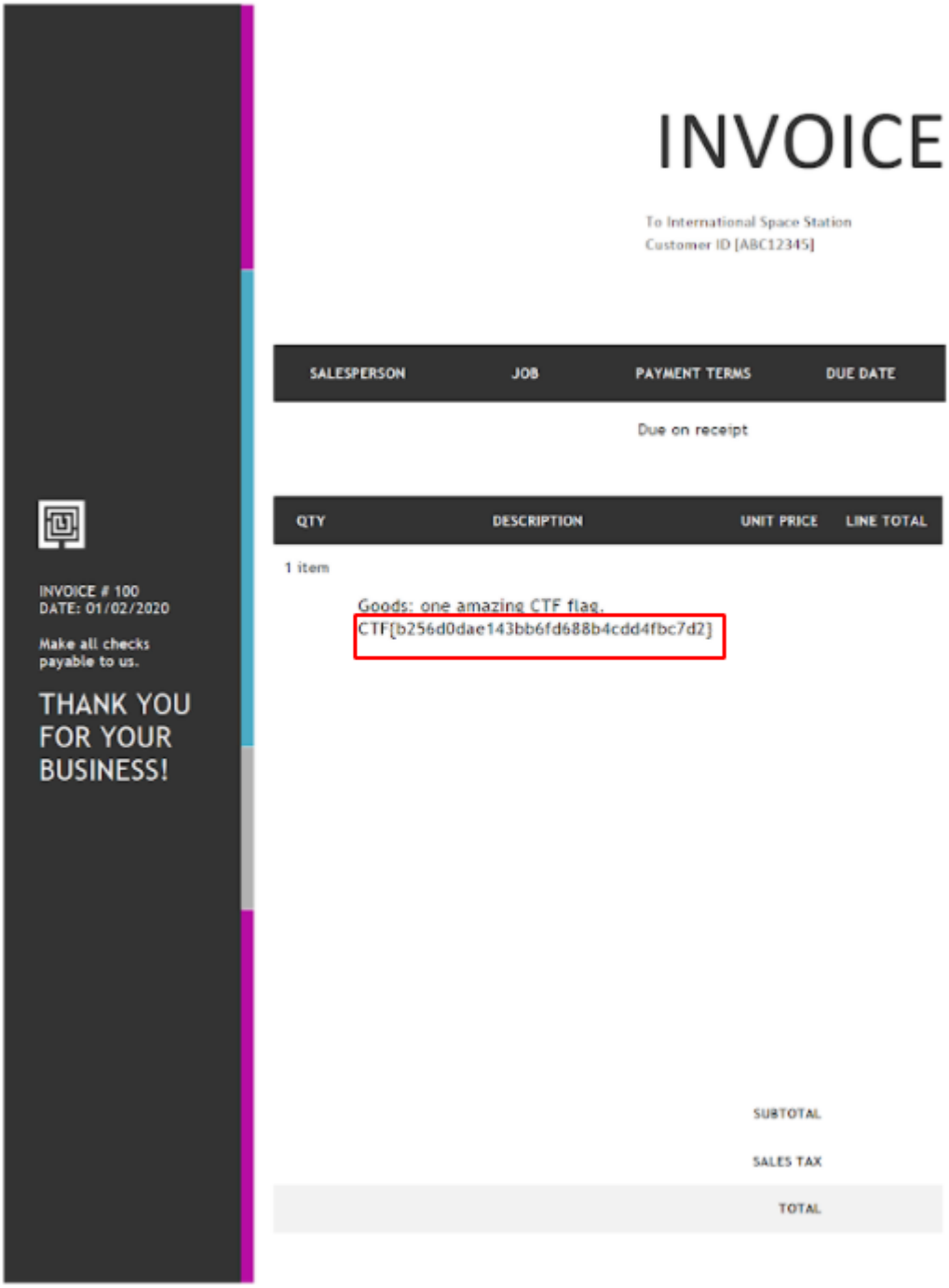
i then used the pdfhash with john to crack

Command→

```
john invoice.hash --wordlist=passwords.txt
```

The Correct PDF Password was: **M=ZjV1z4OMQF.5HM**

Using the above password we get the following Invoice Info



Flag: **CTF{b256d0dae143bb6fd688b4cdd4fbc7d2}**