Cute Invoice

Challenge information

Who knew invoices could be cute?

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.

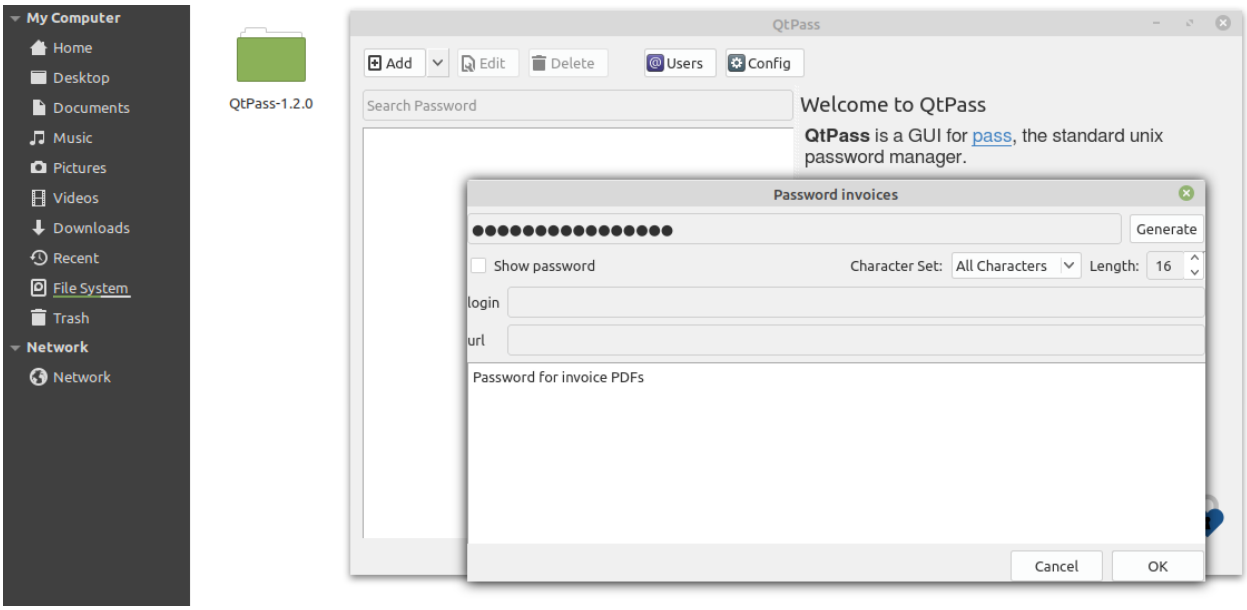
In this challenge we are provided with two files one is an Image and the other is the pdf which is password protected according to the challenge we have to find the password for the same.

The image file gives us information that the pdf is password is protected by program Qtpass and some information about the exact version of the software used

The information given is

Program and version :- QTPass 1.2.0

Password Length :- 16 characters and using All-Characters option



After some research, I found out that there exists a vulnerability for QTPass 1.2.0 in it's a random password generator function

Here is the current password generation function:

```
for (int i = 0; i < length; ++i) {
        int index = Util::rand() % charset.length();
        QChar nextChar = charset.at(index);
        passwd.append(nextChar);
      }
```

The problem here is that modulo will not uniformly distribute that set. The proper way to do things is to just throw away values that are out of bounds. You could try to do the calculation correctly to uniformly stretch or compress, but it's hard to get right, so it's best to just discard numbers outside the set and try again.

Reference link:- https://github.com/IJHack/QtPass/issues/338

According to the above issue, I thought that we can use this generate function with some modification and get the 1000 passwords and use them as the wordlist which I will use to brute force the

First, we have to download and install QT IDE

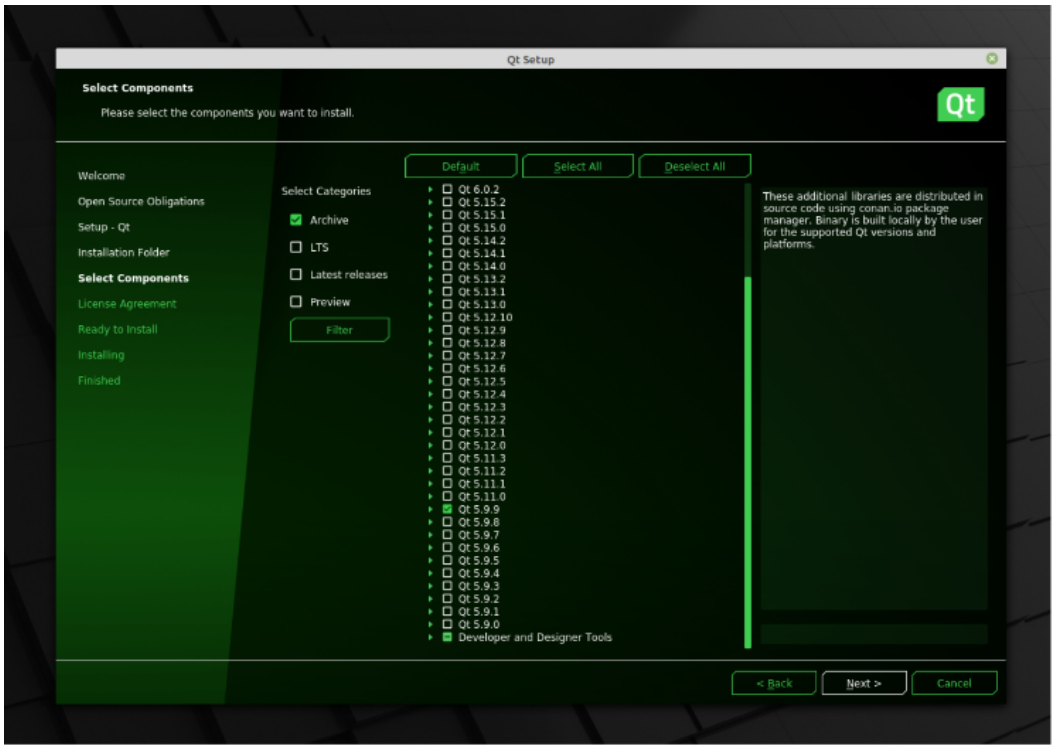
Visit https://www.qt.io and download the software from Try QT

Once downloaded you then need to go through the install process

Code for installing the program

```
chmod +x qt-unified-linux-x64-4.1.1-online.run
./qt-unified-linux-x64-4.1.1-online.run
```

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



Code for password generation is

```
#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;
}
```

We have to use the Qt create with the application console to get main.cpp file where we can use this code with qt libraries else we are unable to execute the code successfully.

The code will give as the output of 1000 password guess which we can then use as the wordlist for cracking the password of the pdf.

To crack passwords we can use the john the ripper, we just have to use the pdf2john.pl file for converting the password hash of the encrypted invoice pdf into hash

Command to do is

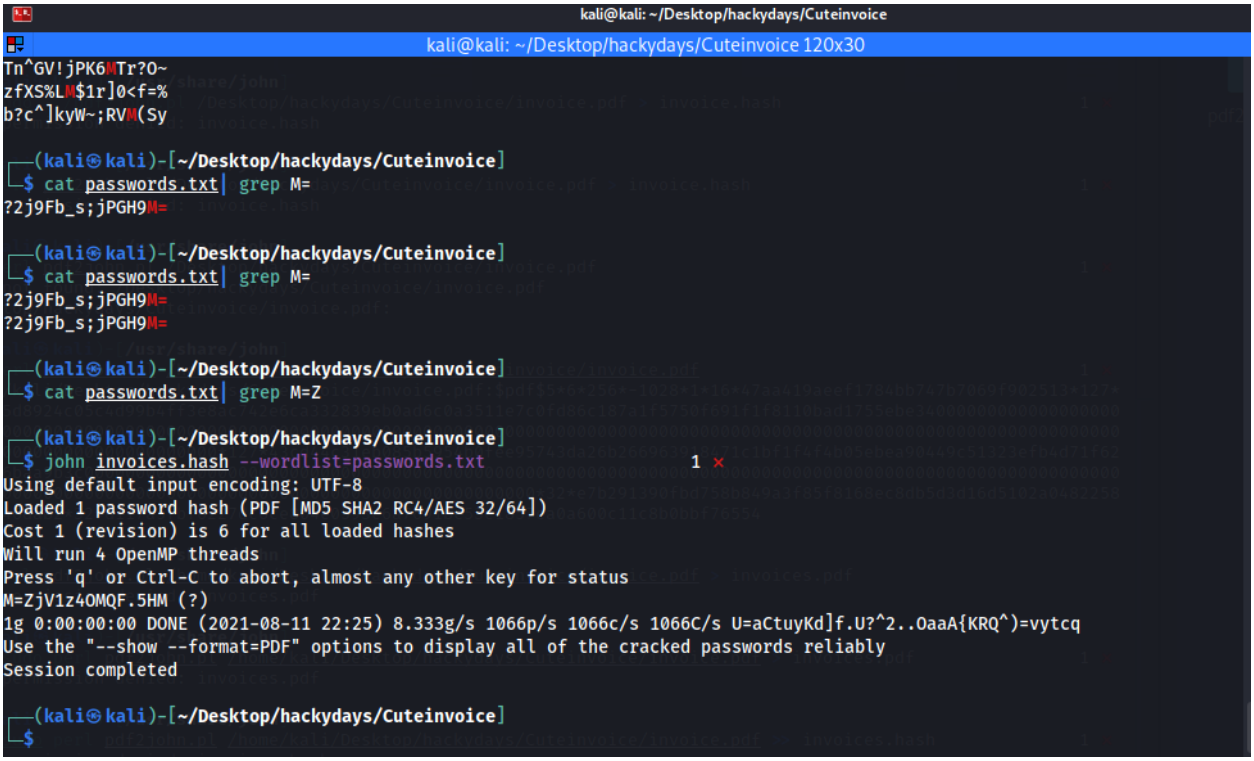
```
sudo perl pdf2john.pl /home/kali/Desktop/hackydays/Cuteinvoice/invoice.pdf > /home/kali/Desktop/hackydays/Cuteinvoice invoices.hash
```

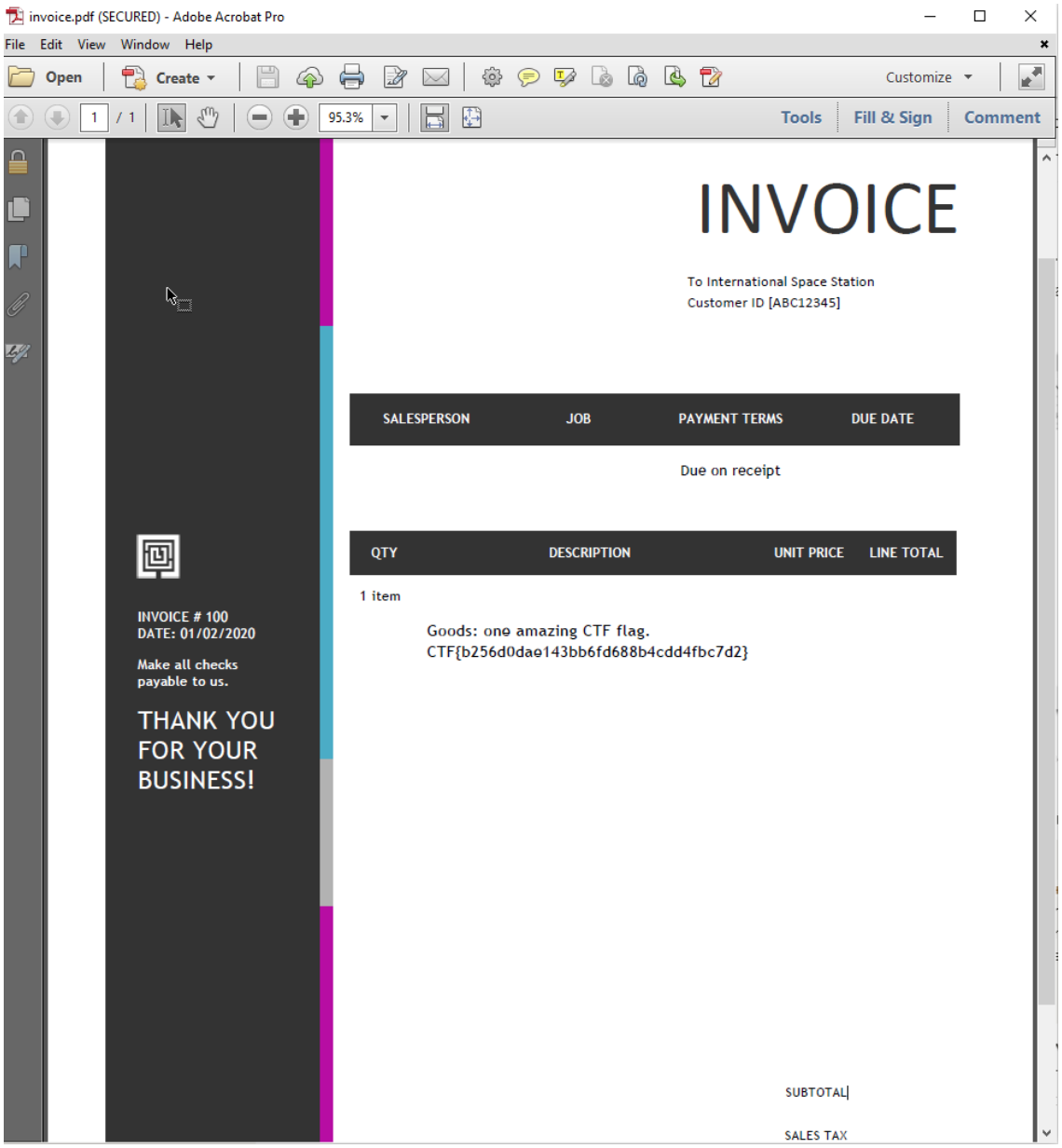
The we can use the john to crack this hash

Command to do so is

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

After this we get the cracked password as M=ZjV1z4OMQF.5HM which will unlock the pdf and we get the hash


kali@kali: ~/Desktop/hackydays/Cuteinvoice
kali@kali: ~/Desktop/hackydays/Cuteinvoice 120x30
Tn^GV!jPK6MTr?O~
zfXS%LM$1r]0<f=%
b?c^]kyW~;RVM(Sy
(kali㉿kali)-[~/Desktop/hackydays/Cuteinvoice]
$ cat passwords.txt| grep M=
?2j9Fb_s;jPGH9M=
(kali㉿kali)-[~/Desktop/hackydays/Cuteinvoice]
$ cat passwords.txt| grep M=
?2j9Fb_s;jPGH9M=
?2j9Fb_s;jPGH9M=
(kali㉿kali)-[~/Desktop/hackydays/Cuteinvoice]
$ cat passwords.txt| grep M=Z
(kali㉿kali)-[~/Desktop/hackydays/Cuteinvoice]
$ john invoices.hash --wordlist=passwords.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])
Cost 1 (revision) is 6 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
M=ZjV1z4OMQF.5HM (?)
1g 0:00:00:00 DONE (2021-08-11 22:25) 8.333g/s 1066p/s 1066c/s 1066C/s U=aCtuyKd]f.U?^2..OaaA{KRQ^)=vytcq
Use the "--show --format=PDF" options to display all of the cracked passwords reliably
Session completed
(kali㉿kali)-[~/Desktop/hackydays/Cuteinvoice]
$



Flag :- CTF{b256d0dae143bb6fd688b4cdd4fbc7d2}