

HackThisCode / CTF-Writeups

Watch5

★ Star15

🍴 Fork9




<> Code ⓘ Issues 0 🔄 Pull requests 2 📁 Projects 0 Insights ▼

Branch: master ▼ CTF-Writeups / 2016 / SCTF / Lengthy-Lingo /

Create new file Find file History

bryde Added writeups

Latest commit b6f227a on Apr 14 2016

..		
 README.md	Added writeups	a year ago
 decode.pl	Added writeups	a year ago
 encrypted.dat	Added writeups	a year ago

📖 README.md

#Lengthy Lingo

Can you crack the code? We intercepted this flag but can't seem to figure out how it was encrypted.

Hint: *The numbers don't seem to follow a specific pattern...*

🔗 Solution overview

We get a file **encrypted.dat** which contain the cipher text that we need to decode, it contain a lot what seems to be huge integers separated by commas.

My first idea was to convert each of these large integers to their equivalent hex representation and see if some pattern would emerge, that however didn't seem to give me anything solid.

Reading the challenge title and the hint over and over and then sleeping on it, proved to be a good approach and yeah i got it.

"Lengthy Lingo" : It has something to do with length

What has length in this encrypted.dat data??

Its filled with integers seperated by commas!

Does the integers have length?

They have a value.

Wait they also have a length in how many digits they contain!!!

Finally it had dawned on me, that we could try to go through each integer counting the length of each one, the hint says **The numbers don't seem to follow a specific pattern...** which also told me the numbers wasn't important themselves.

We need a program for this

Program

The task at hand is to write a program that take a filename as input in the terminal and splits the content of the file on ","

1. Handle command line arguments
2. Split the content of the file on ","
3. For each element in the list we split out, count the number of characters.
4. Convert the length value for each integer to their ascii equivalent.
5. Print the result!

```
#!/usr/bin/perl

my $filename = $ARGV[0];
unless (-e $filename) {usage();exit(0);}

open my $fh,"<$filename" or die "Error opening file $filename!\n";
my $content = join "",<$fh>;
chomp $content;

print decrypt($content);

sub decrypt {
    my $content = shift;

    my @parts = split /,/,$content;
    my $result = "";
    foreach my $p (@parts) {
        $p =~ s/\\s+//g;
        $result .= chr(length($p));
    }
    return $result;
}

sub usage() {
    print "\n";
    print "Usage: \n";
    print "\t$0 <encrypted-file>\n";
    print "\n";
}
```

Solution

Run the program

```
./decode.pl encrypted.dat
```

and you get the flag!!

```
sctf{101_th3_numb3r5_d1dn'7_3v3n_m4tt3r}
```

