README.md

CONFidence DS CTF Teaser: Crypto100

Category: Crypto Points: 100 Description:

> We've created a lotto system. Shall we play a game?
>
> ```
> nc 23.253.207.179 10001
> ```
>
> Use the source

Write-up

The lottery is a Python script that uses pycrypto, so run `easy_install pycrypto` if you want to run it locally.

Let's connect to the online service and see how it works:

```
$ nc 23.253.207.179 10001
################################################################################
#                                                                              #
# Welcome to our Lotto!                                                        #
# Bid for $5, win $100!                                                        #
# Our system is provably fair:                                                 #
#    Before each bid you'll receive encrypted result                           #
#    After the whole game we will reveal the key to you                        #
#    Then, you can decrypt results and verify that we haven't cheated on you! #
#     (e.g. by drawing based on your input)                                    #
#                                                                              #
################################################################################

Your money: $100
Round verification: 18167373f0918114f833cdbc184202bf

Your choice:
        1. Buy a coupon for $5
        2. Withdraw your money
        3. Quit
1
Your guess (0-1000): 42
You lost!
The lucky number was: 182
[enter] to continue...

Your money: $95
Round verification: 30b15b44ce3cbfb2b936538c36e41bc5

Your choice:
```

```
        1. Buy a coupon for $5
        2. Withdraw your money
        3. Quit
2
You cannot withdraw your money until you get $1337!
The lucky number was: 292
[enter] to continue...

Your money: $95
Round verification: 1e3f942dfd3f5718789e14ebf570eae7

Your choice:
        1. Buy a coupon for $5
        2. Withdraw your money
        3. Quit
3
The lucky number was: 342
Verification key: 5e230c5a1f2b766309ec28bf2eff01d8
```

Cross-referencing this with the source code, it becomes clear there these are the three options for each round:

1. Bet $5. If you guess the number from `0` to `1000` correctly, you win $100 (i.e. a profit of $95).
2. Withdraw your money. This displays the flag if you have $1337 or more, and fails otherwise. Either way it also reveals the lucky number that you were supposed to guess.
3. Quit the game.

Looking more closely, we learn that the lottery uses the same AES encryption key for each round in the same connection/session:

```
key = Random.new().read(16) # slow, but secure
aes = AES.new(key, AES.MODE_ECB)
```

However, each lucky number is concatenated with a random salt before it's encrypted into the so-called "round verification hash":

```
salted = str(luckyNumber) + '#'
salted = randomExtend(salted)
```

For example, if `luckyNumber` is `84`, then `randomExtend(str(luckyNumber) + '#')` is something like `84#00059d2a12d51`. After patching `lotto.py` to log the salted lucky number for each round, it became clear that all salts started with two zeroes:

```
247#000000000000
84#00059d2a12d51
255#00beca8fe000
364#00ddf93a9d91
538#00552d9ca771
743#003fbc1cb5c9
…
```

Yep, that first one even has only zeroes! This seemed to happen quite commonly; about 10% of all salts ended in `000000000000`. Looks like `randomExtend` isn't as random as it should be :) (Explanation: any number with a `0` as the last digit (i.e. 10% of numbers) raised to a high power will have a bunch of zeroes at the end.)

This flaw enables us to write a fairly simple brute-force solution with the following algorithm:

1. Connect to the lottery service.
2. For each round, make a note of the round verification hash. If the round verification hash has been used before, bet $5 on the lucky number it mapped to before; this gets you $95. If not, choose "Withdraw your money", which shows you the lucky number for this round — remember that the given round verification hash mapped to this lucky number.
3. If you keep repeating this long enough, at some point you'll have $1430. Now you can choose "Withdraw your money" to get the flag.

`brute-force-solution.py` is a Python implementation of this solution. It took the script about 25 minutes to make enough money to get the flag:

```
$ python brute-force-solution.py
Round #1 | Money: $100 | Round verification: 496be14c66a5aac6af1fd841f26102f4
Lucky number: 229
Round #2 | Money: $100 | Round verification: 86244b7964db766497590d826db87c8e
Lucky number: 873
Round #3 | Money: $100 | Round verification: 4f535ef3d98a142cee40ecf6d34a4334
Lucky number: 994
Round #4 | Money: $100 | Round verification: 02996452c102ef13939680754257460f
Lucky number: 178
Round #5 | Money: $100 | Round verification: 88f71ecb1561208166ec112cb6bc13c3
Lucky number: 217
Round #6 | Money: $100 | Round verification: 80510634961a4c985d07ff089b520310
Lucky number: 778
Round #7 | Money: $100 | Round verification: b5475dea706c3c7285568d68092c5a06
Lucky number: 610
Round #8 | Money: $100 | Round verification: a3e0645d43ac069fbc322d5cdbf692bf

[…]

Round #219 | Money: $100 | Round verification: 3d3955b4fee673bbffed2760b7e5f537
You won $100!
The lucky number was: 389
[enter] to continue...

Round #220 | Money: $195 | Round verification: 81aa91772881d915a951e72270f39291
Lucky number: 627
Round #221 | Money: $195 | Round verification: 71d8b1bba071056e5aa510e9b6dbe35d
Lucky number: 743
Round #222 | Money: $195 | Round verification: dfab6036861640d59479444df681ca9e
Lucky number: 165
Round #223 | Money: $195 | Round verification: 0cda534713cbfd4aa218bf2648b879d5
Lucky number: 183
Round #224 | Money: $195 | Round verification: 880091b15457a0a9c0922b9233bbbc52
You won $100!
The lucky number was: 1000
[enter] to continue...

Round #225 | Money: $290 | Round verification: 07f9716235a394eddf246f46ff814280
Lucky number: 245

[…]

Round #1684 | Money: $1335 | Round verification: 54a9309da2f4f6e26b40a6439ae209cf
You won $100!
The lucky number was: 128
[enter] to continue...

Round #1685 | Money: $1430 | Round verification: bcd4bd9ac16e16fe83d76b1831f7da2e
You won! Here's your reward: DSCTF_939da0eec884d9edddbe97b9f9e348dede7211d821a1b56069816d7bad6c0f2e

The lucky number was: 953
[enter] to continue...
```

The flag is `DSCTF_939da0eec884d9edddbe97b9f9e348dede7211d821a1b56069816d7bad6c0f2e`.

Other write-ups and resources

- http://www.pwntester.com/blog/2014/04/27/dragonsector-crypto-100/
- http://blog.dul.ac/2014/04/DSCTF14/