

Write-Up: Discreet Psychologist (Crypto)

Ομάδα: mousiko_gymsasio_agrinou_lt_1 (Μουσικό Γυμνάσιο
Αγρινίου - Λ.Τ.)
Μαθητές/Μαθήτριες Πέτρος Παπαθανασίου

Επισκόπηση Δοκιμασίας

Η δοκιμασία μας δίνει δύο αρχεία, το source.py και το output.txt. Το source.py ασχολείται με την κρυπτογράφηση ενός μηνύματος και το output.txt μας δίνει τις πληροφορίες της κρυπτογράφησης. Σκοπός της δοκιμασίας είναι να προσπαθήσουμε να αντιστρέψουμε την διαδικασία κωδικοποίησης και να ανακτήσουμε το flag

Επίλυση

Αρχική ανάλυση

Το αρχείο source.py ασχολείται με κρυπτογράφηση με την συνάρτηση `row(generator,ciphertext,prime)`. Από το αρχείο output.txt αμέσως παρατηρούμε ότι γνωρίζουμε το `prime`, τον `generator` και μια σειρά από αριθμούς της μορφής `[101132890306787, [643, 607, 643, 593]]`. Ο πρώτος αριθμός (101132890306787) δηλώνει τον κωδικοποιημένο χαρακτήρα του flag `ciphertext`. Οι επόμενοι τέσσερις αριθμοί αποτελούν τους παράγοντες που χρησιμοποιούνται για την κωδικοποίηση του plaintext.

Η ευπάθεια είναι το γεγονός ότι γνωρίζουμε όλες τις πληροφορίες που χρησιμοποιήθηκαν στην κωδικοποίηση, δηλαδή τα `ciphertext`, `prime`, `generator` και τους παράγοντες.

```
1 {"outputs": [[101132890306787, [643, 607, 643, 593]], [170026252139071, [653, 569, 727, 521]], [183708726761661, [601, 719, 577, 811]], [135124296317819, [929, 967, 1013, 823]], [37347122738542, [809, 569, 977, 641]], [2343296191068, [983, 587, 569, 593]], [157652736731762, [797, 733, 647, 811]], [53767169520066, [787, 761, 617, 769]], [17685333759038, [613, 967, 613, 617]], [127505790185576, [971, 691, 1013, 769]], [173779558301296, [887, 883, 643, 857]], [178011888187584, [757, 733, 547, 673]], [73297665576047, [947, 617, 881, 547]], [74204800950708, [967, 991, 521, 853]], [11514506032657, [811, 523, 631, 599]], [55316439587762, [661, 613, 557, 1009]], [162585381961424, [751, 877, 863, 569]], [118073681149310, [857, 743, 937, 593]], [77515471172195, [967, 829, 937, 733]], [69589561620637, [827, 1013, 719, 739]], [82140133117026, [911, 601, 613, 773]], [128514041384200, [881, 647, 883, 1009]], [78961557837500, [619, 757, 983, 827]], [102125898901740, [743, 739, 907, 863]], [179261777530499, [563, 617, 631, 911]], [38642716285153, [983, 541, 523, 1013]], [105660736603656, [653, 761, 859, 739]], [128336999210691, [617, 683, 787, 911]], [101966327651032, [839, 709, 691, 883]], [74245325241085, [521, 859, 757, 757]], [159671231291082, [1019, 673, 659, 853]], [59601410515, [919, 733, 569, 829]], [161204136331634, [883, 751, 823, 593]], [122275519962291, [653, 863, 751, 757]], [156129139236191, [887, 991, 673, 709]], [42646736284116, [907, 947, 929, 599]], [12114098482578, [829, 971, 827, 599]], [30116411567634, [829, 1021, 911, 907]], [44070940419838, [773, 647, 659, 653]], [70039658209594, [809, 911, 521, 557]], [73849313682655, [787, 563, 881, 953]], [95470950911545, [877, 761, 853, 617]], [32759547555985, [541, 811, 541, 1021]], [96544897170174, [883, 787, 661, 887]], [6561254173275, [911, 661, 733, 809]], [149734745985798, [977, 1021, 769, 743]], [75751888194360, [577, 661, 857, 773]], [25569653524001, [859, 659, 829, 829]], [75713542761046, [587, 919, 773, 743]], {"prime": 187700937902549, "generator": 2}]
```

```
def encrypt(plaintext,prime,generator):  
    a,b,c,d = [getPrime(10) for _ in range(4)]  
  
    ciphertext = plaintext*a*b*c*d  
    for i in range(randint(100,400)):  
        ciphertext = pow(generator,ciphertext,prime)  
  
    return (ciphertext, [a,b,c,d])
```

Τέλος, είναι γνωστό ότι η κωδικοποίηση επαναλαμβάνεται από 100 έως 400 φορές.

Εκμετάλλευση ευπάθειας

Το πρώτο πράγμα που έκανα ήταν να ακολουθήσω την εκφώνηση και να χρησιμοποιήσω διακριτούς λογαριθμούς για να αντιστρέψω την λογική της κωδικοποίησης.

Δοκιμασία 20 Λύσεις

Discreet Psychologist
149

Μέτρια Κρυπτογραφία Cryptography

Η συνταγή του ψυχολόγου μου δεν βγάζει νόημα.
Μακάρι να μπορούσα να βρω τη λογαριθμική πίσω
από αυτήν...

Author: eld3t1c

crypto-disc...

Σημεία Submit

Αυτό δεν έχει αποτέλεσμα, καθώς διαπίστωσα ότι η επανάληψη από 100-400 φορές μου έδινε ως αποτέλεσμα μία ακολουθία χαρακτήρων και δεν ήξερα ποιος είναι ο σωστός.

Οπότε, προσανατολίστηκα στο να κάνω την διαδικασία της κωδικοποίησης και να ελέγχω αν τα αποτέλεσμα ciphertext που μου δίνει κάθε φορά θα ισούται με το ciphertext στο output.txt. Ως text για την κωδικοποίηση έπαιρνα χαρακτήρες από μία λίστα που είχε όλους τους εκτυπούμενους, και πιθανούς για flag, ascii χαρακτήρες. Άρα το πρόγραμμα κάθε φορά που έφτανε στην ισότητα, σταμάταγε και μου έκανε print στην κονσόλα τον χαρακτήρα που βρήκε. Με αυτήν την τεχνική, μπορούσα να τσεκάρω και την εγκυρότητα του προγράμματός μου, αφού γνώριζα ότι οι πρώτοι χαρακτήρες θα ήταν FLAG{. Κάθε φορά που έβρισκα έναν χαρακτήρα τον σημείωνα, ώστε που βρήκα όλο το flag.

```
1 whitelist = " !#$%&'()*+,-./123456789;:;<=>?@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~"
2 prime = 187700937902549
3 generator = 2
4 a,b,c,d = 587, 919, 773, 743 # Εδώ βάζουμε τους παράγοντες
5
6
7 for ch in whitelist:
8     plaintext = ord(ch)
9     ciphertext = plaintext * a * b * c * d
10    for i in range(400):
11        ciphertext = pow(generator,ciphertext,prime)
12        if ciphertext == 75713542761046: # Έλεγχος των ciphertext
13            print(ch)
14            break
15
16
17
18
```

Το πρόγραμμα το έτρεξα για όλα τα ciphertext.

```
{ "outputs": [  
  [101132890306787, [643, 607, 643, 593]],  
  [170026252139071, [653, 569, 727, 521]],  
  [183708726761661, [601, 719, 577, 811]],  
  [135124296317819, [929, 967, 1013, 823]],  
  [37347122738542, [809, 569, 977, 641]],  
  [2343296191068, [983, 587, 569, 593]],  
  [157652736731762, [797, 733, 647, 811]],  
  [53767169520066, [787, 761, 617, 769]],  
  [17685333759038, [613, 967, 613, 617]],  
  [127505790185576, [971, 691, 1013, 769]],  
  [173779558301296, [887, 883, 643, 857]],  
  [178011888187584, [757, 733, 547, 673]],  
  [73297665576047, [947, 617, 881, 547]],  
  [74204800950708, [967, 991, 521, 853]],  
  [11514506032657, [811, 523, 631, 599]],  
  [55316439587762, [661, 613, 557, 1009]],  
  [162585381961424, [751, 877, 863, 569]],  
  [118073681149310, [857, 743, 937, 593]],  
  [77515471172195, [967, 829, 937, 733]],  
  [69589561620637, [827, 1013, 719, 739]],  
  [82140133117026, [911, 601, 613, 773]],  
  [128514041384200, [881, 647, 883, 1009]],  
  [78961557837500, [619, 757, 983, 827]],  
  [102125898901740, [743, 739, 907, 863]],  
  [179261777530499, [563, 617, 631, 911]],  
  [38642716285153, [983, 541, 523, 1013]],  
  [105660736603656, [653, 761, 859, 739]],  
  [128336999210691, [617, 683, 787, 911]],  
  [101966327651032, [839, 709, 691, 883]],  
  [74245325241085, [521, 859, 757, 757]],  
  [159671231291082, [1019, 673, 659, 853]],  
  [59601410515, [919, 733, 569, 829]],  
  [161204136331634, [883, 751, 823, 593]],  
  [122275519962291, [653, 863, 751, 757]],  
  [156129139236191, [887, 991, 673, 709]],  
  [42646736284116, [907, 947, 929, 599]],  
  [12114098482578, [829, 971, 827, 599]],  
  [30116411567634, [829, 1021, 911, 907]],  
  [44070940419038, [773, 647, 659, 653]],  
  [70039658209594, [809, 911, 521, 557]],  
  [73849313682655, [787, 563, 881, 953]],  
  [95470950911545, [877, 761, 853, 617]],  
  [32759547555985, [541, 811, 541, 1021]],  
  [96544897170174, [883, 787, 661, 887]],  
  [6561254173275, [911, 661, 733, 809]],  
  [149734745985798, [977, 1021, 769, 743]],  
  [75751888194360, [577, 661, 857, 773]],  
  [25569653524001, [859, 659, 829, 829]],  
  [75713542761046, [587, 919, 773, 743]]],  
}
```

F
L
A
G
{
h
t
t
p
s
:
/
/
w
w
w
.
y
o
u
t
u
b
e
.
c
o
m
/
w
a
t
c
h
?
v
=
R
B
t
l
P
T
2
3
P
T
M
}

Σημαία

Η σημαία που βρέθηκε:

FLAG{ https://www.youtube.com/watch?v=RBt1PT23PTM }
