Write-Up: My Rainbow Sorcerer (Crypto)

Ομάδα:	mousiko_gymsasio_agriniou_lt_1 (Μουσικό Γυμνάσιο Αγρινίου - Λ.Τ.)
Μαθητές/Μαθήτριες	Πέτρος Παπαθανασίου

Επισκόπηση Δοκιμασίας

Η δοκιμασία μας δίνει έναν server και ένα αρχείο σε python. Από τον όνομα καταλαβαίνουμε ότι το αρχείο είναι αυτό που τρέχει όταν συνδεόμαστε στον server με netcat.

Επίλυση

Αρχική ανάλυση

Ανοίγοντας το με έναν editor είδα ότι ασχολείται με την κρυπτογράφηση. Πιο αναλυτικά, χρησιμοποιεί κάποιους συγκεκριμένους χαρακτήρες και δημιουργεί τέσσερεις διαφορετικές κρυπτογραφήσεις. Έπειτα μας δίνει ως output ένα hash, το οποίο είναι κρυπτογραφημένο χρησιμποιόντας έναν από αυτούς τους τέσσερεις τρόπους. Όταν τσεκάρει το input που του δίνουμε, αν είναι σωστό διαβάζει από ένα αρχείο “flag.txt”.

Για αυτόν τον λόγο το πρώτο πράγμα που σκέφτηκα ήταν να κάνω command execution μέσω του input. Αφού δοκίμασα να τρέξω εντολές συστήματος, όπως “import('os').system('ls')” ή “import('os').system('dir')” παρατήρησα ότι δεν δίνει κάποιο αποτέλεσμα, τσεκάροντας ταυτόχρονα αν ο server είναι windows ή linux. Το ίδιο έκανα και με εντολές python σε περίπτωση που μπορεί να λάβει ως input κώδικα σε python. Για παράδειγμα εντολές όπως “open('flag.txt').read()” ή “print(open('flag.txt').read())”. Όμως και πάλι δεν υπήρχε αποτέλεσμα.

Εκμετάλλευση ευπάθειας

Έτσι σκέφτηκα να δημιουργήσω ένα script, το οποίο να ελέγχει την ύπαρξη του κάθε hash μέσα σε ένα αρχείο λίστας όλων των πιθανόν hash. Πρώτα όμως θα έπρεπε να δημιουργήσουμε την λίστα. Για αυτό δημιούργησα ένα πρόγραμμα που να δημιουργεί τις πιθανές απαντήσεις ανάλογα με το δωσμένο hash.

```python
from hashlib import md5, sha256, sha1
from itertools import product
from Crypto.Util.number import bytes_to_long as b2l, long_to_bytes as l2b
import json

pool = "0123456789abcdef"

def bxor(a, b, c):
    return bytes(x ^ y ^ z for x, y, z in zip(a, b, c))

hash_dict = {}

for chars in product(pool, repeat=5):
    input_str = "".join(chars)

    grade4 = sha1(input_str.encode()).hexdigest()
    grade2 = sha1(bxor(sha256(input_str.encode()).digest(), md5(input_str.encode()).digest(), sha1(input_str.encode()).digest())).hexdigest()
    grade1 = sha1(sha256(md5(input_str.encode()).digest()).digest()).hexdigest()
    special_grade = sha1(l2b(b2l(md5(input_str.encode()).digest()[:8]) << 32) + l2b(b2l(md5(input_str.encode()).digest()[8:]) << 52)).hexdigest()

    hash_dict[grade4] = input_str
    hash_dict[grade2] = input_str
    hash_dict[grade1] = input_str
    hash_dict[special_grade] = input_str

# Η λίστα που δημιουργείται
with open("list.json", "w") as f:
    json.dump(hash_dict, f)

print("Τέλος Διαδικασίας")
```

Τρέχοντας αυτό το python script πήρα το αρχείο json με την λίστα. Μετά από αυτό έπρεπε να φτάξω ένα script, που να κάνει το decryption.

```python
import json

with open("list.json", "r") as p:
    hash_dict = json.load(p)

get_hash = input("Δώσε το hash: ").strip()
if get_hash in hash_dict:
    print(f"Hash: {hash_dict[hash_to_find]}")
else:
    print("Έγινε λάθος!")
```

Έκανα κάποιες δοκιμές και λειτουργούσε.

```
└─$ python3 lookup.py
Δώσε το hash: ed03f0d9d84576533700c737f7ebba3b122cd1bf
Hash: 5d8a1
```

Με αυτά τα στοιχεία έτρεξα τον server και ταυτόχρονα τα προγράμματα για να παίρνω τα σωστά αποτελέσματα. Όμως μου έδινε πολύ λίγο χρόνο (5 δευτερόλεπτα) και ήθελε πολλά

διαφορετικά checks (500). Για αυτόν τον λόγο σκέφτηκα ότι πρέπει να γίνει με έναν αυτοματοποιημένο τρόπο. Η μοναδική λύση που μπορούσα να σκεφτώ ήταν να δημιουργήσω ένα script το οποίο να συνδέεται στον server με την χρήση των sockets. Αυτό φαίνεται στον παρακάτω κώδικα.

```
1   import socket
2
3   HOST = "challenges.pmdk.gr"
4   PORT = 58569
5
6   s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7   s.connect((HOST, PORT))
```

Ως τελικό κώδικα χρησιμοποιήσα έναν, ο οποίος να κάνει print τα δεδομένα που εμφανίζει ο server με συγκεκριμένα βήματα και δίνει στο input την σωστή απάντηση. Έβαλα και έναν μετρητή για να γρωρίζω τις φορές που εκτελέστηκε η while, την σημαία μου την έδωσε την φορά 725, αφού το πρόγραμμα εκτελούνταν κάποιες φορές πιο γρήγορα από όσο ήθελε το input για να εμφανιστεί.

```
└─$ python3 final.py
Do you have the Hash because you know the Input?
Or you have the Input because you know the Hash ~ Suguru Geto


0


Hash: 'd4653e433f1a5c64adec9c377ba2574d6379ac11'
Answer: '91df6'
1


Hash: '73253047a3ee82d125947e378d0c4ea9e71129c3'
Answer: '5a95a'
2
```

Το αποτέλεσμα μας δίνει ένα βίντεο στο youtube. Όταν το ανοίγω βλέπω ότι είναι ένα τραγούδι το Bad Habot από το κανάλι 7clouds. Έτσι, σκέφτομαι ότι δεν γίνεται να είναι κάτι κρυμμένο μέσα στο βίντεο. Για αυτό τσεκάρω μήπως το flag είναι το link για το βίντεο και τελικά είναι αυτό.

Σημαία

Η σημαία που βρέθηκε:

```
FLAG{https://www.youtube.com/watch?v=bU2EvRBUmxc}
```