



Mission Name

The Doors

Historical Context

Ethan is tasked with disabling the security apparatus and surveillance by navigating the locks of each door within the Skytech network bridge room.

Technical Synopsis

Ethan's mission involves remote infiltration into the Skytech network bridge, a secured area guarded by multi-tiered security doors. He must locate and utilize the appropriate keys to bypass each barrier, facing a series of challenges designed to test his resolve and skill.

Mission Brief

Ethan, your objective lies within the Skytech Network Bridge, a vault secured by numerous cryptographic locks. Each door presents a unique puzzle, requiring the correct decryption key for access. Your greeting message upon entry will be: "Welcome to the Skytech Network Bridge Security System." Best of luck!

Detailed Assignment

Ethan's goal is to dismantle the surveillance and security mechanisms by systematically unlocking the doors within the Skytech network bridge room.

Operational Venue

SYLVARCON | SKYTECH Headquarters

Tools

- `nc -nv IP 7777`

Questions

What famous equation is based first door?

- Quadratic

What famous number is based second door?

- Pi

What type of number is based third door?

- Primes

Items

1. In first door resolve quadratic equation with c or $a = 1$, $b^2 - 4ac = 0$
2. In second door automatize the number of pi decimal and calculate the next one
3. In third door automatize the next prime number

Categories

- Programming
- Mathematics

Write Up

Connect to the specified port using Netcat or a similar tool:

- nc IP 7777

Upon connection, the security system will present a unique challenge for each door you encounter.

First Door Challenge:

You'll be tasked with solving quadratic equations where either the coefficient 'c' or 'a' equals 1. Given the equation $b^2 - 4ac = 0$, you can simplify to find that $b^2/4a = c$ or $b^2/4c = a$. Notably, 'b' is often a multiple of 4, simplifying the solution to a straightforward calculation.

Second Door Challenge:

This challenge involves recognizing a sequence of numbers as the digits of π to a certain number of decimal places. Your task is to automate the identification and continuation of this sequence.

Third Door Challenge:

Initially appearing unsolvable, this challenge presents a sequence where one number requires predicting the next. However, if you've successfully automated solutions for the previous doors, you'll notice that all presented numbers are prime. The task then becomes automating the identification of the subsequent prime number.

Automating Solutions:

To efficiently tackle these challenges, especially the third, utilize a script named `writeup.py`. This script will facilitate the automatic submission of correct answers for each door's challenge.

By executing `writeup.py`, you can streamline the process of passing through all the doors to ultimately obtain the flag.

```
#!/bin/python
# -*- coding: utf-8 -*-

import random
import socket
import sys
import time
import os
import re
```

```
import string
import math
import cmath
import decimal

def is_prime(num, test_count):
    if num == 1:
        return False
    if test_count >= num:
        test_count = num - 1
    for x in range(test_count):
        val = random.randint(1, num - 1)
        if pow(val, num - 1, num) != 1:
            return False
    return True

def generate_big_prime(n):
    found_prime = False
    while not found_prime:
        p = random.randint(2**(n - 1), 2**n)
        if is_prime(p, 1000):
            return p

def pi_decimal(n):
    """Calculate PI to the Nth digit.
    This one uses the native python Decimal type that allows arbitrary
    precision. With n = 1000, it took less than a second on my laptop.
    """
    D = decimal.Decimal
    decimal.getcontext().prec = n + 10
    pi = D(0)
    k = 0
    tail_prev = ""
    while True:
        pi += (D(4) / D(8 * k + 1) - D(2) / D(8 * k + 4) - D(1) / D(8 * k + 5) - D(1) / D(8 * k + 6)) / D(16)**D(k)
        tail = str(pi)[n + 1:-1]
        if tail == tail_prev:
            break
        tail_prev = tail
        k += 1
    decimal.getcontext().prec = n
    return str(pi * D(1))
```

```
def door1(ciphertext):
    # ax^2+bx+c
    # full format Got 1x^2+532x+70756, resolv the problem (format: 0.00000)
    # format 1x^2+936x+219024
    a = int(ciphertext.split("x^2+")[0].split("Got ")[1])
    b = int(ciphertext.split("x^2+")[1].split("x+")[0])
    c = int(ciphertext.split("x^2+")[1].split("x+")[1].split(",")[0])
    # -b + RAIZ[b^2-4ac / 2a]
    sol = ((-b) + ((cmath.sqrt((b * b) - 4 * a * c)))) / (2 * a)
    # 5 decimales maximo
    plaintext = "%.5f" % sol.real
    return plaintext

def door2(ciphertext):
    # format Got 3.14159265358979323846264338328, return pi with one more decimal
    decimals = len(ciphertext.split("3.")[1].split(",")[0])
    plaintext = pi_decimal(decimals+2)
    return plaintext

def door3(ciphertext):
    # Got 32717, next prime is?
    prime_n = int(ciphertext.split("Got ")[1].split(",")[0])
    cond = True
    while cond:
        prime_n = prime_n + 2
        if is_prime(prime_n, 1000):
            cond = False
    plaintext = prime_n
    return str(plaintext)

finish = True

s = socket.socket()
s.connect(('18.130.152.9', 7777))

while finish:

    # s.settimeout(1)
```

```
# obtener ciphertext
ciphertext = s.recv(2048)
#print ciphertext.split("\n")
print ciphertext
# obtener ciphertext
time.sleep(1)
ciphertext = s.recv(2048) #delete if local
print ciphertext
print "[+]First Door: " + ciphertext.split("\n")[5]
ciphertext = ciphertext.split("\n")[5]
print ciphertext
puerta1 = door1(ciphertext)
# envio quadratic
print "[+]First key: " + puerta1
s.send(puerta1)

time.sleep(1)
# obtener ciphertext
ciphertext = s.recv(2048)
print ciphertext
print "[+]Second Door: " + ciphertext.split("\n")[5]
ciphertext = ciphertext.split("\n")[5]

puerta2 = door2(ciphertext)
# envio pi
print "[+]Second key: " + puerta2
s.send(puerta2)

time.sleep(1)
ciphertext = s.recv(2048)
print ciphertext
print "[+]Third Door: " + ciphertext.split("\n")[5]
ciphertext = ciphertext.split("\n")[5]

puerta3 = door3(ciphertext)
# envio prime
print "[+]Third key: " + puerta3
s.send(puerta3)

time.sleep(1)

# flag
ciphertext = s.recv(2048)
print ciphertext
```

```
finish = False  
s.close()
```

```
python writeup.py  
Welcome Skytech Security Doors Management  
  
First Door  
Please answer with the correct key to continue:  
Got 64x^2+16x+1, resolv the problem (format: 0.00000)  
[+]First Door: Got 64x^2+16x+1, resolv the problem (format: 0.00000)  
Got 64x^2+16x+1, resolv the problem (format: 0.00000)  
[+]First key: -0.12500  
Go to Next Door  
  
Second door  
Please answer with the correct key to continue:  
Got 3.1415926535897932384626433832795, please add one more decimal  
[+]Second Door: Got 3.1415926535897932384626433832795, please add one more decimal  
[+]Second key: 3.14159265358979323846264338327950  
Go to Next Door  
  
Third door  
Please answer with the correct key to continue:  
Got 41, next number is?  
[+]Third Doord: Got 41, next number is?  
[+]Third key: 43  
Go to Next Door  
  
You have deactivate the security system  
flag{d00rs_are_really_stucks}
```

Figure 1

Flag Information

flag{d00rs_are_really_stucks}