Obraz zawierający tekst, zrzut ekranu, monitor, komputer

Opis wygenerowany automatycznieKrok 1. Ustalenie własnego adresu IP

Obraz zawierający tekst, zrzut ekranu, computer, monitor

Opis wygenerowany automatycznieKrok 2. Ustalenie maski

Obraz zawierający tekst, zrzut ekranu, computer, komputer

Opis wygenerowany automatycznieKrok 3. Adresy IP aktywnych hostów

Obraz zawierający tekst, zrzut ekranu, monitor, komputer

Opis wygenerowany automatycznieKrok 4. Otwarte porty

Obraz zawierający tekst, zrzut ekranu, monitor, computer

Opis wygenerowany automatycznieKrok 5. Banner grabbing

Obraz zawierający tekst

Opis wygenerowany automatycznieKrok 6\_1 analiza

W konsoli odnaleziono tekst zakodowany formatem Base64

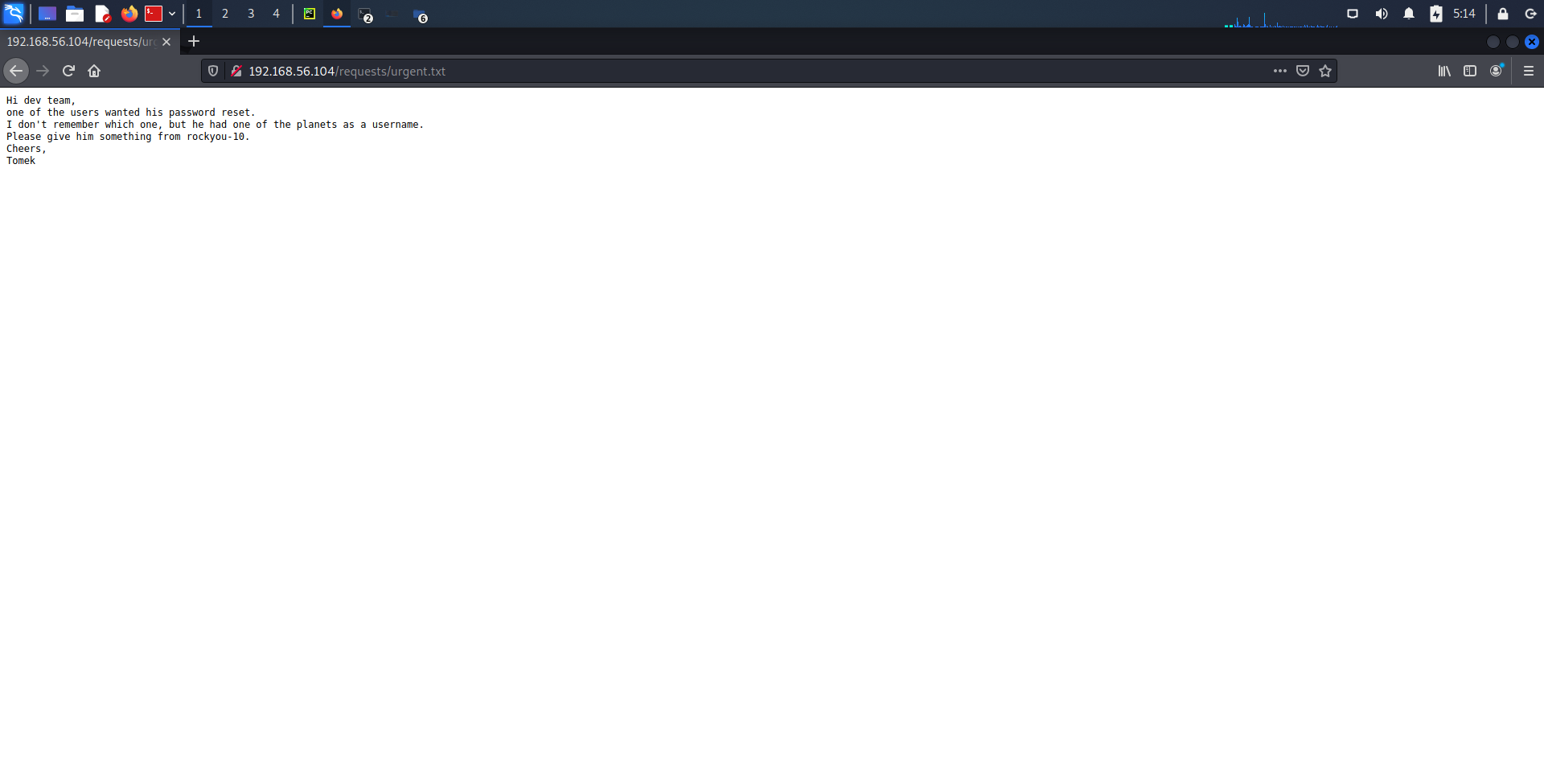
Obraz zawierający tekst

Opis wygenerowany automatycznieKrok 6\_2 odkodowanie teksy za pomocą strony base64decode.org

Obraz zawierający tekst, zrzut ekranu, monitor, komputer

Opis wygenerowany automatycznieKrok 6\_3 enumeracja podstron

Krok 6\_4

Po sprawdzeniu podstron, natrafiliśmy na plik tekstowy urgent.txt

W tym miejscu skończył nam się czas na zadanie 😊

Kody do poszczególnych kroków:

Krok1: ustalenie własnego adresu IP

import os

ip = os.popen('hostname -I').read()

print(ip)

Krok 2: ustalenie maski:

import os

def mask\_search():

device = input("Z jakiego interfejsu sieciowego korzystasz? [np. eth0]: ")

mask = str(os.popen(f"nmcli dev show {device} | grep IP4.ADDRESS").read()).strip().split("/")[1]

print(mask)

return mask

mask\_search()

krok 3. Adresy IP aktywnych hostów:

import os

output = os.popen('hostname -I').read()

lista\_ip = []

from scapy.all import \*

from scapy.layers.inet import \*

for ip in range(100, 256):

adres = output[0:11] + str (ip)

print(adres)

packet = IP(dst=adres, ttl=20)/ICMP()

reply = sr1(packet, timeout=2)

if reply:

print(adres, "is online", end = "\n \n")

lista\_ip.append(adres)

else:

print(adres,"DOWN", end = "\n \n")

print(lista\_ip)

Krok 4. Otwarte porty

from scapy.all import \*

from scapy.layers.inet import \*

otwarte\_porty1 = []

for port in range (1,100):

pakiet = IP(dst="192.168.56.104")/TCP(dport=port, sport=random.randint(20000, 30000), flags="S")

response = sr1(pakiet)

if str(response.getlayer(TCP).flags) =="SA":

otwarte\_porty1.append(port)

print("Otwarte porty VM:", end="")

print(otwarte\_porty1)

Krok 5. Banner grabbing

import socket

targetIP = input("podaj adres:")

while True:

try:

mySocket = socket.socket()

targetPort = input("podaj port:")

if "exit" in targetPort:

pass

else:

mySocket = socket.socket()

mySocket.connect((targetIP, int(targetPort)))

mySocket.send ("Get banner \r \n".encode())

serverRecv = mySocket.recv((2048)).decode()

print(serverRecv)

except:

continue