



دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

گزارش پروژه ی اول مهندسی اینترنت

Packet Tunneling

پدیدآورنده:

محمد امین کیانی

۴۰۰۳۶۱۳۰۵۲

یزدان افرا

۴۰۰۳۶۱۳۰۰۵

دانشجویان کارشناسی، دانشکده ی کامپیوتر، دانشگاه اصفهان، اصفهان،

استاد درس: جناب آقای دکتر شاهقلی

نیمسال اول تحصیلی ۱۴۰۳-۰۴

فهرست مطالب

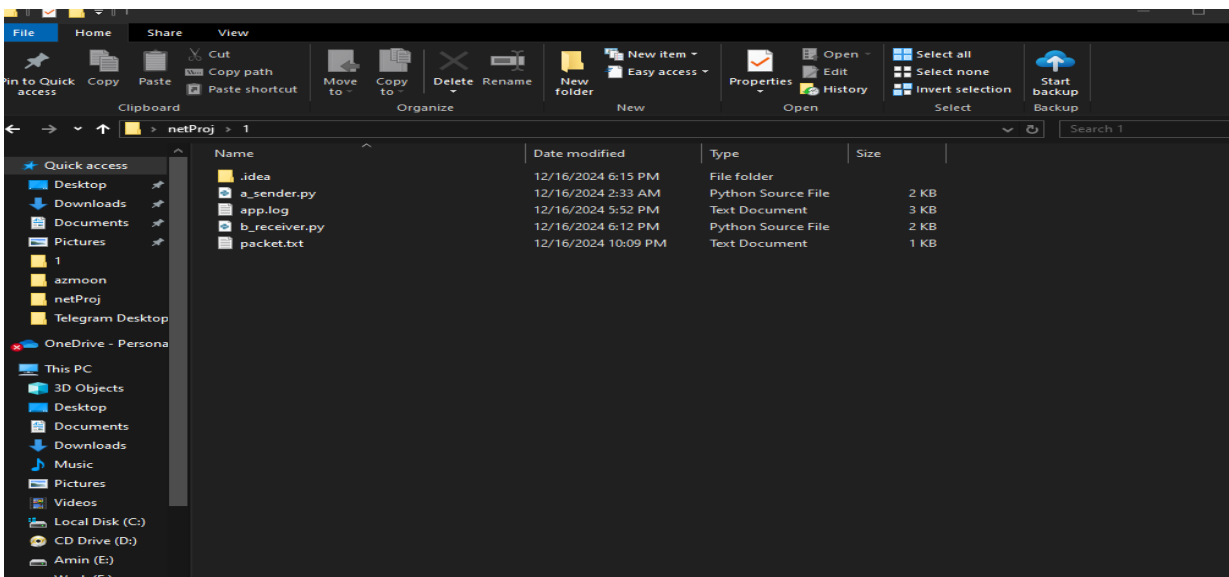
۳	مستندات
۴	سیستم فرستنده
۶	سیستم گیرنده
۶	بخش اول:
۷	بخش دوم:
۷	نتایج

مستندات

در این پروژه دو سیستم که به هم شبکه شده باشند را آماده کنید و بعد روی این دو سیستم دو برنامه بنویسید که:

۱- سیستم اول یک فایل متنی را خوانده و محتوایش را خط به خط داخل بسته های IP قرار دهید. این بسته های IP آدرس مقصد شان آدرس خود همین سیستم هست و مبدا شان سیستم دوم می باشد. بعد بسته را داخل بسته IP دیگری قرار دهید که آدرس مقصدش سیستم دوم هست و بعد هم بسته را برای سیستم دوم ارسال کنید.

۲- سیستم دوم بسته را دریافت و چک می کند که متعلق به خودش باشد. بعد بسته IP درونی که داخل بسته IP بیرونی است را خارج می کند و بدون تغییر برای سیستم اول ارسال می کند و سیستم اول بسته های فاقد تونل را دریافت و چک می کند که متعلق به خودش باشند و بعد محتوای متنی داخل payload بسته ها را پرینت می کند. در واقع سیستم اول فایلی که خودش ارسال کرده بود را خط به خط دریافت و چاپ می کند.



سیستم فرستنده

ارسال بسته ی تونل شده(تو در تو):

```
# Mohammad Amin Kiani 4003613052

# 192.168.1.106

from scapy.all import *
from scapy.layers.inet import IP, ICMPError
import logging

# log config file
log_file = "app.log"
logging.basicConfig(
    filename=log_file,
    level=logging.DEBUG,
    format="%(asctime)s - %(levelname)s - %(message)s"
)

# each line is a new packet
def read_file(file_path):

    with open(file_path, "r") as file:
        lines = file.readlines()

    lines = [line.strip() for line in lines]

    return lines

def create_send_tunnel_packet(dest_ip, src_ip):
    lines = read_file("packet.txt")
    for payload in lines: # proto 4 6 ==> 10 nadarim
        tunnel_packet = IP(src=dest_ip, dst=src_ip, proto=10) / payload # inner
        # tunnel from 102 to me(106)
        packet = IP(src=src_ip, dst=dest_ip) / tunnel_packet # outter tunnel
        # (my packet) from 106 to me(102)
        send(packet, verbose=False) #verbos => Log
        logging.info(f"send one packet to {dest_ip}")

# end packet to finish code
tunnel_packet = IP(src=dest_ip, dst=src_ip, proto=10) / "EOF"
packet = IP(src=src_ip, dst=dest_ip) / tunnel_packet
```

```

send(packet, verbose=False)
logging.info(f"send one packet to {dest_ip}")

def print_packet(packet): # oon be man
    if packet.haslayer(IP) and packet[IP].dst == '192.168.1.106' and
packet[IP].src == '192.168.1.102'\
        and packet[IP].proto == 10:
        logging.info(f"receive one packet from {packet[IP].src}")
        chunk_of_file = packet[IP].payload
        try:
            printable_file_content = bytes(chunk_of_file).decode('utf-8') #
convert byte to string
            print(printable_file_content)

        except:
            print("I can't ")

def start_receiving():
    sniff(prn=print_packet, filter="ip", store=0)

create_send_tunnel_packet(dest_ip='192.168.1.102', src_ip='192.168.1.106')
#-----
start_receiving()

```

تذکر: پروتکل رزرو نشده‌ی ۱۰ را برای بسته های شخصی خود که بصورت محلی در لایه ی سوم میفرستیم، انتخاب می کنیم تا از ورود دیگر بسته ها مثل ping و ... در کنسول ارسال خود جلوگیری کنیم و صرفا بسته های ما بدون هیچ مداخلگر icmp یی تبادل شود. برای ارسال هر بسته ای در این لایه باید پروتکل ۱۰ را از کد ها حذف کرده تا اجازه ی عبور همگی داده شود.

```

app.log - Notepad
File Edit Format View Help
1024-12-16 17:43:00,221 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:20,950 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:20,956 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:20,962 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:20,967 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:20,975 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:20,980 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:20,985 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:20,992 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:20,996 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:21,004 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:21,008 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:21,014 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:21,020 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:21,025 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:21,030 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:21,036 - INFO - send one packet to 192.168.1.102
1024-12-16 17:52:21,153 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,154 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,156 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,167 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,159 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,160 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,166 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,167 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,168 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,168 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,169 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,172 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,176 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,179 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,180 - INFO - receive one packet from 192.168.1.102
1024-12-16 17:52:21,181 - INFO - receive one packet from 192.168.1.102

```

سیستم گیرنده

بخش اول:

دریافت محتوای داخل بسته ی تونل شده:

```
# 192.168.1.102

from scapy.all import *
from scapy.layers.inet import IP, ICMPError
import logging

#log config
packets = []
file_transmission_complete = False
log_file = "app.log"
logging.basicConfig(
    filename=log_file,
    level=logging.DEBUG,
    format="%(asctime)s - %(levelname)s - %(message)s"
)

def handler_return_tunnel_packet(packet): # male man
    if packet.haslayer(IP) and packet[IP].dst == '192.168.1.102':
        print("ok")
        inner_packet = packet[IP].payload
        print(inner_packet.haslayer(IP))
        if inner_packet.haslayer(IP):
            print(inner_packet.show())
            logging.info(f"receive one packet from {packet[IP].src}")
            print(bytes(inner_packet.payload))
            packets.append(inner_packet)

def stop_sniff(packet):
    if packet.haslayer(IP) and packet[IP].dst == '192.168.1.102':
        inner_packet = packet[IP].payload
        if b'EOF' in bytes(inner_packet.payload):
            return True
```

```

# just give ip packet
def start_receiving():
    sniff(prn=handler_return_tunnel_packet, filter="ip", store=0,
stop_filter=stop_sniff)

#scapy sniffer
start_receiving()

#-----

for packet in packets:
    send(packet)
    logging.info(f"send one packet to {packet[IP].src}")
    print(packet)

```

بخش دوم:

برگشت بسته ی داخل بسته ی تونل شده و چاپ آن در فرستنده:

نتایج

