Project 1

شیرین بهنامی نیا – نرگس منتظری

Questions:

1- چرا این روند برعکس نیست یعني کلاینت يك پورت را باز نمي کند و آن را به سرور اطلاع دهد و سرور فايل را روي آن پورت براي کلاينت ارسال کند؟

به طور کلی پروتکل ftp دو مود متفاوت دارد: active mode و passive mode . در passive mode که ما هم در حقیقت همون رو پیاده سازی کردیم دیتاپورت رندوم از سمت سرور تعیین میشود و برای کلاینت فرستاده میشود تا به آن کانکت شود اما در active mode دقیقا برعکس این اتفاق میفتد. اگر چیزی به نام firewall نداشتیم عملا این حالت برامون مشکل خاصی ایجاد نمیکرد اما در حال حاضر شرایط فرق داره. کلاینت ها عموما پشت یک فایروال یا یک NAT قرار دارند و به همین دلیل فقط از طریق پورت های خاصی (که بهش predefined میگن) از بیرون قابل دسترسی اند. پس وقتی سرور تلاش میکنه از طریق پورت رندومی که کلاینت براش فرستاده با اون ارتباط برقرار کنه به احتمال خیلی زیاد اون پورت جزو پورت های predefined نیست و در نتیجه توسط فایروال بلاک میشه و هیچ کانکشنی برقرار نمیشه. به همین دلیل default mode در این پروتکل passive هست.

2- نام این حمله چیست؟

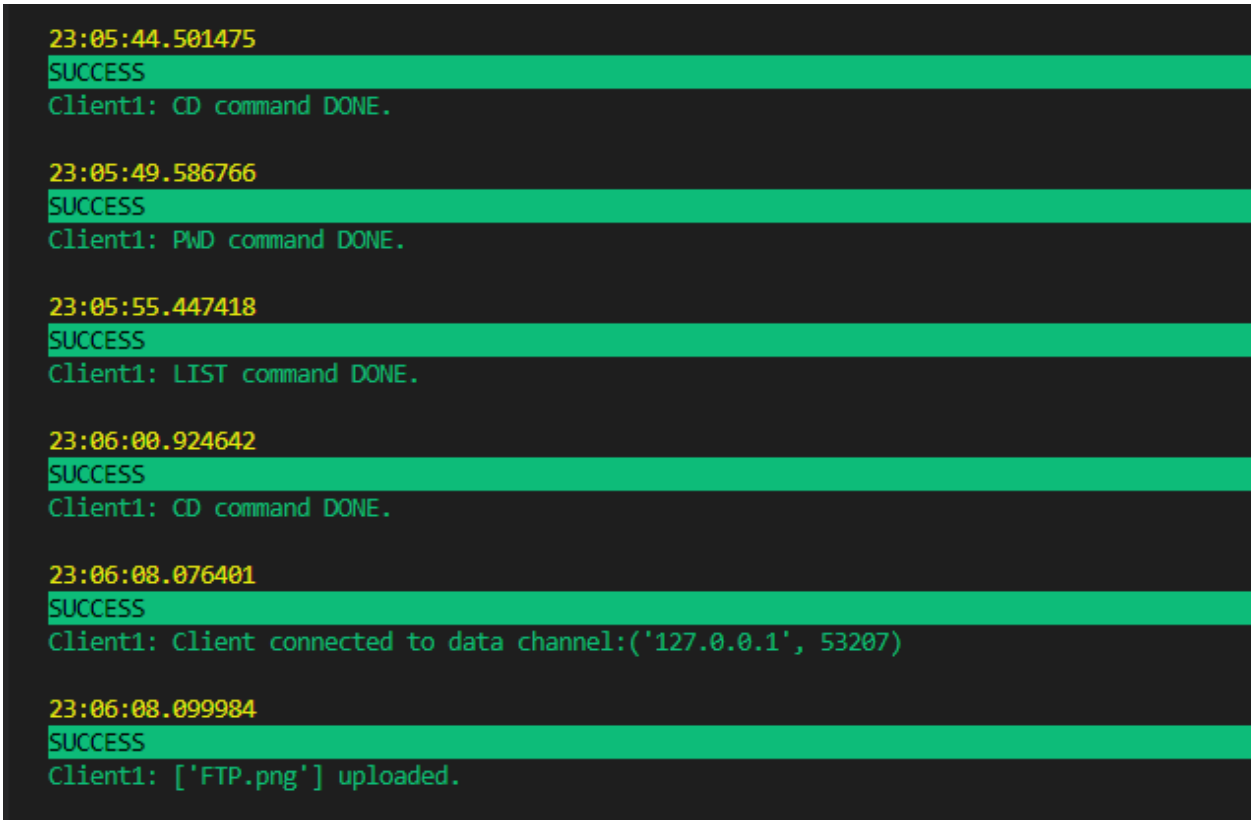
اسم این حمله Directory Traversal Attack است.

زمانی رخ می‌دهد که حمله‌کننده بتواند به فراتر از فهرست‌های اصلی سرور و فایل‌هایی که اجازه دارد، دسترسی پیدا کند. اگر او چنین دسترسی را داشته باشد می‌تواند هم فایل‌هایی که مجاز نیست را دریافت کند و به اطلاعات آن‌ها دسترسی پیدا کند و هم ممکن است دستورات سیستم عامل را اجرا کند که باعث ایجاد مشکل در سرور می شود.

3- خروجی های سرور و کلاینت:

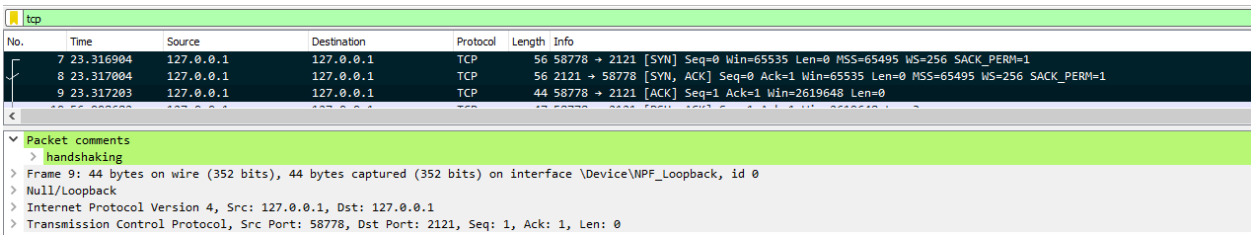
از خروجی های کلاینت جلوتر اسکرین شات قرار دادیم.

سرور:



Wireshark:

1- handshaking packets

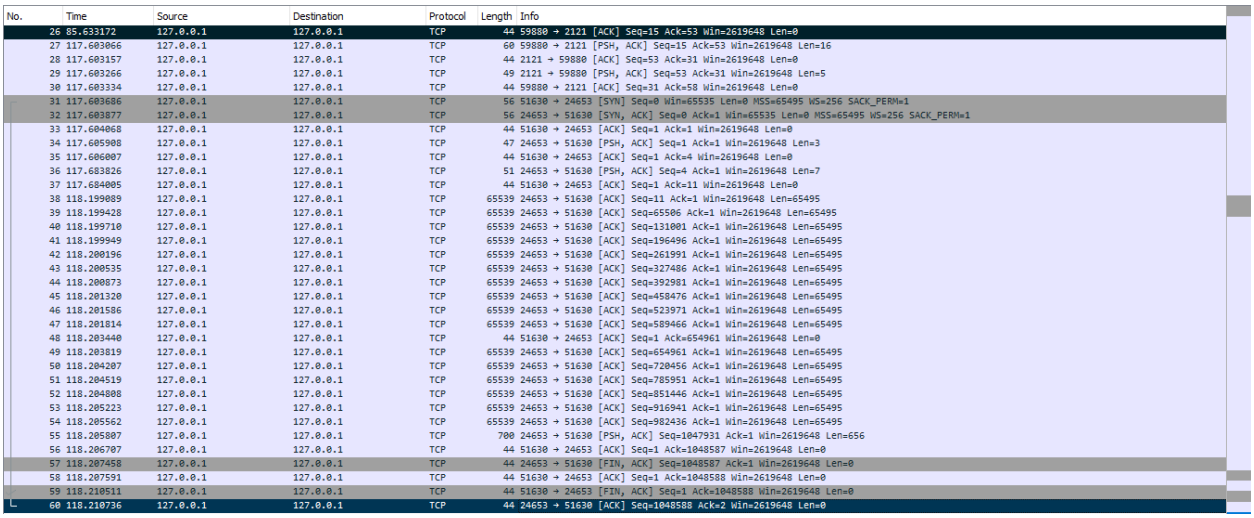


-Step1(SYN): Client to Server (informs the server that the client is likely to start communicate.)

-Step2(SYN-ACK): Server to Client (signifies the response of the segment it received and SYN signifies with what sequence number it is likely to start the segments with)

-Step3(ACK): Client to Server (client acknowledges the response of the server)

2-TCP Packet size limitation



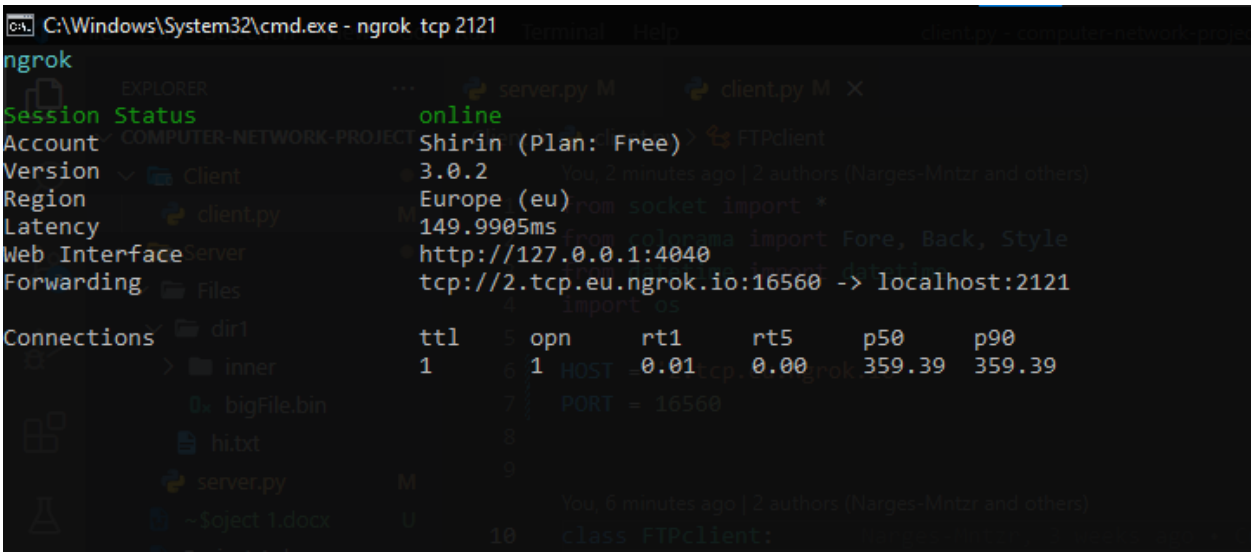
-ماکزیمم اندازه پکت در این پروتکل تقریبا برابر 64 کیلوبایت است. (با توجه به وایرشارک 65539 بایت به طور حدودی) این مقدار با توجه به سخت افزار مورد استفاده در شبکه تعیین میشود.(mtu)

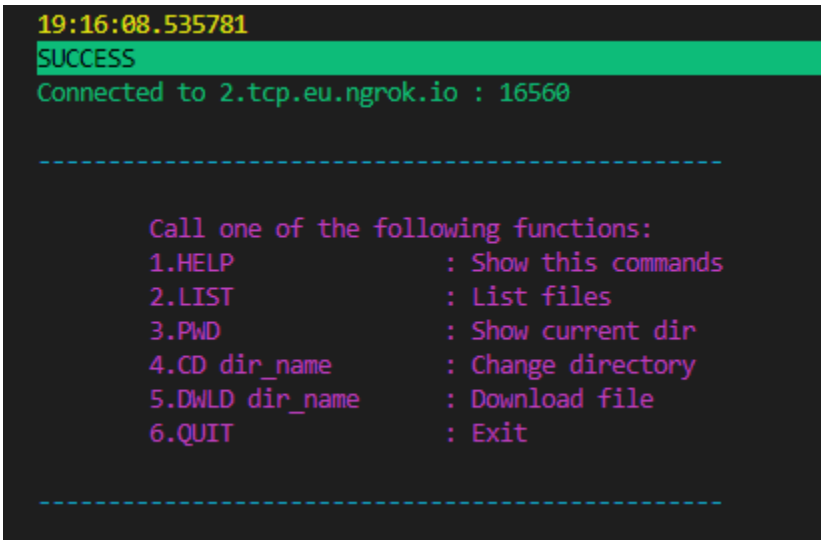
MTU : MTU is the maximum size of the data transfer limit set by hardware in a network. The packet size should never exceed MTU.

این پروتکل برای ارسال دیتا وقتی با حجمی بیشتر از این مقدار مواجه میشود خودش دیتا را به بسته هایی با ماکزیمم سایز 64k میشکند و در پکت های جداگانه ارسال میکند که در مقصد(کلاینت) دوباره به هم متصل میشن و دیتای اصلی رو تشکیل میدن. (هر پکت با توجه به اینکه چه مسیری بهینه تره براش، مسیر متفاوتی رو طی میکنه.)
به طور کلی استفاده از پکت های کوچکتر احتمال loss of dataرو کم میکنه و network latency رو کم میکنه.(همون lag)
-در انتقال bigfile.bin باتوجه به حجم 1024 کیلوبایتی تقریبا همونطور که انتظار داشتیم دیتا در 16 تا پکت ارسال شده است.

Ngrok:

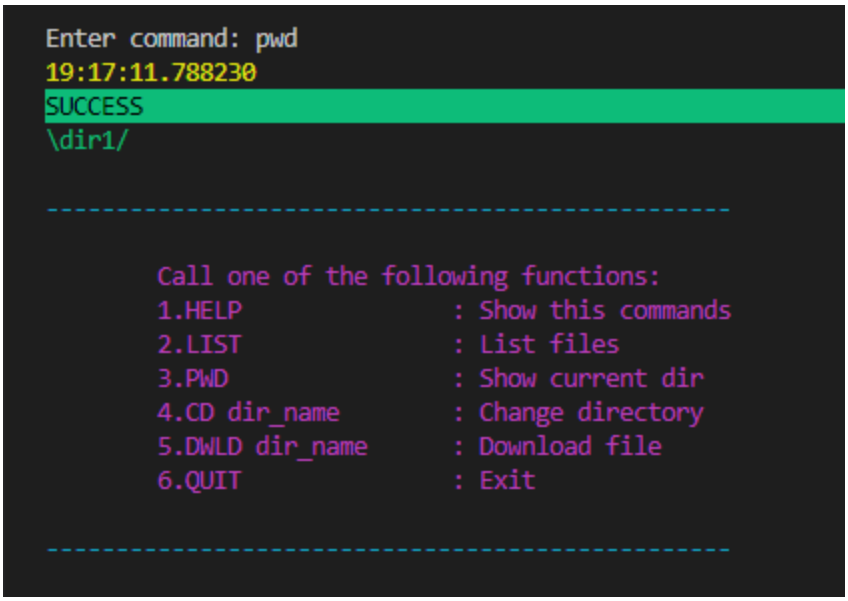
Part1:

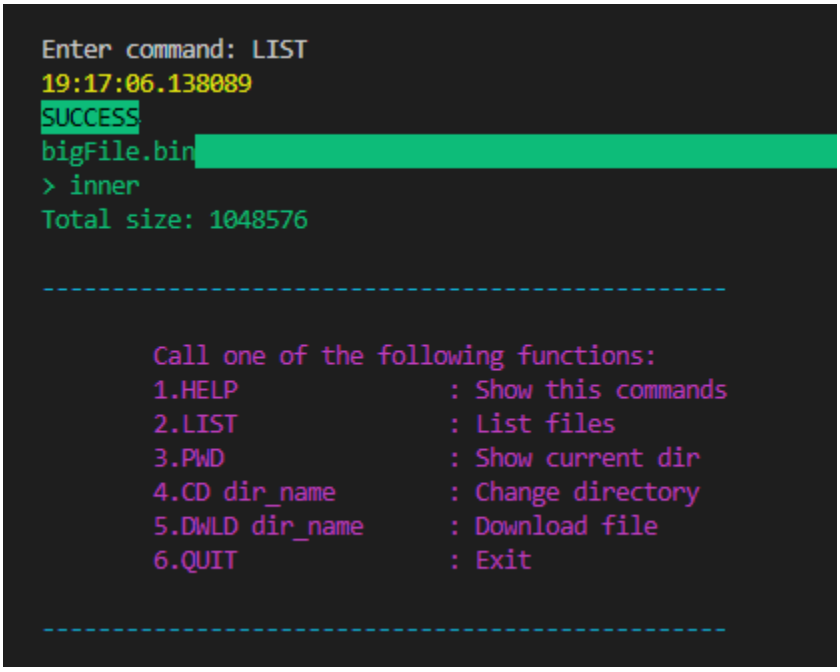

C:\Windows\System32\cmd.exe - ngrok tcp 2121
ngrok
Session Status online
Account Shirin (Plan: Free)
Version 3.0.2
Region Europe (eu)
Latency 149.9905ms
Web Interface http://127.0.0.1:4040
Forwarding tcp://2.tcp.eu.ngrok.io:16560 -> localhost:2121
Connections ttl opn rt1 rt5 p50 p90
1 1 0.01 0.00 359.39 359.39


19:16:08.535781
SUCCESS
Connected to 2.tcp.eu.ngrok.io : 16560
Call one of the following functions:
1.HELP : Show this commands
2.LIST : List files
3.PWD : Show current dir
4.CD dir_name : Change directory
5.DWLD dir_name : Download file
6.QUIT : Exit

```
------------------------------------------------

Enter command: cd dir1
19:16:24.119584
SUCCESS
\dir1/

------------------------------------------------

        Call one of the following functions:
        1.HELP              : Show this commands
        2.LIST              : List files
        3.PWD               : Show current dir
        4.CD dir_name       : Change directory
        5.DWLD dir_name     : Download file
        6.QUIT              : Exit

------------------------------------------------
```

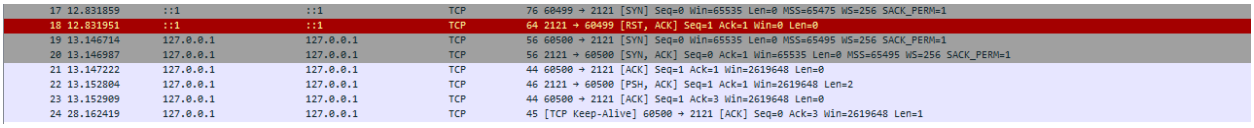
```
Enter command: cd fff
19:18:13.680403
ERROR
fff not found.

------------------------------------------------

        Call one of the following functions:
        1.HELP              : Show this commands
        2.LIST              : List files
        3.PWD               : Show current dir
        4.CD dir_name       : Change directory
        5.DWLD dir_name     : Download file
        6.QUIT              : Exit

------------------------------------------------
```


Enter command: pwd
19:17:11.788230
SUCCESS
\dir1/
Call one of the following functions:
1.HELP : Show this commands
2.LIST : List files
3.PWD : Show current dir
4.CD dir_name : Change directory
5.DWLD dir_name : Download file
6.QUIT : Exit


Enter command: LIST
19:17:06.138089
SUCCESS
bigFile.bin
> inner
Total size: 1048576
Call one of the following functions:
1.HELP : Show this commands
2.LIST : List files
3.PWD : Show current dir
4.CD dir_name : Change directory
5.DWLD dir_name : Download file
6.QUIT : Exit

```
Enter command: help

------------------------------------------------

        Call one of the following functions:
        1.HELP              : Show this commands
        2.LIST              : List files
        3.PWD               : Show current dir
        4.CD dir_name       : Change directory
        5.DWLD dir_name     : Download file
        6.QUIT              : Exit

------------------------------------------------
```

Part2:



سه تا بسته ی اول مربوط به handshaking است.(دو تا بسته ی پایین تر که فلگ psh دارند در حالت قبلی هم مشاهده میشد.) اینجا حالت loopback است و تفاوت قابل ملاحظه ای با حالت قبلی ندارد. اما به طور کلی (هرچند من فقط موفق شدم در حالت loopbackپکت ها را ببینم) اگر بتوانیم پکت ها را بیرون از حالت loopback ببینیم، پورتی که سرور مشخص کرده هم در قسمت infoمشخص است و منطقا ادرس مبدا و مقصد هم متفاوت است.

یه تفاوتی که اینجا به چشم میاد نسبت به حالت قبلی اینه که اینجا تعداد زیادی بسته TCP-Keep-Alive داریم که وقتی دستوری وارد نمی کنیم فرستاده میشن که مطمئن بشن یک کانکشن هنوز فعاله.

Part3:

مشکلی که وجود دارد این است که پورت دیتا چنل به صورت رندوم تولید میشه و ما نمیتونیم حین اجرا به شیوه فعلی برای اون ip عمومی بسازیم.

ولی این کار ممکن است. به کمک کتابخانه pyngrok در پایتون می‌توان ابتدا شماره پورت رندومی را ایجاد کرد و سپس یک کانکشن با ngrok برقرار کرد و آدرس هاست و شماره پورت آن را برای کلاینت ارسال کرد. سپس کلاینت به آدرسی که دریافت می‌کند وصل و فایل مورد نظر، از طریق آن کانال از سرور ارسال می‌شود .

البته نسخه رایگان ngrok مشکلاتی مانند محدودیت زمان و پهنای باند دارد که برای ارسال فایل‌های بزرگ از سرور به کلاینت مشکل ساز است و نمی توان فایل را یکجا ارسال کرد بلکه باید فایل را به تکه‌های کوچک‌تری تقسیم‌بندی کرد.

(ما این کار را در فایل جداگانه‌ای به اسم use-ngrok انجام داده‌ایم و آن را در کنار فایل اصلی قرار دادیم)