Computer Networks Project 2

## Shirin Behnaminia 9919863

## Narges Montazeri 9931053

# Part 2:

## 2-1:

زمانی که ما سوکتمون رو ساختیم از دستور زیر استفاده کردیم:

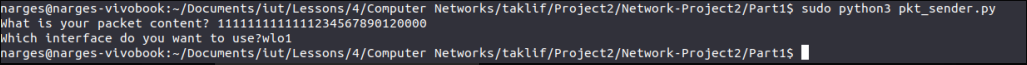
s = socket(AF\_PACKET, SOCK\_RAW)

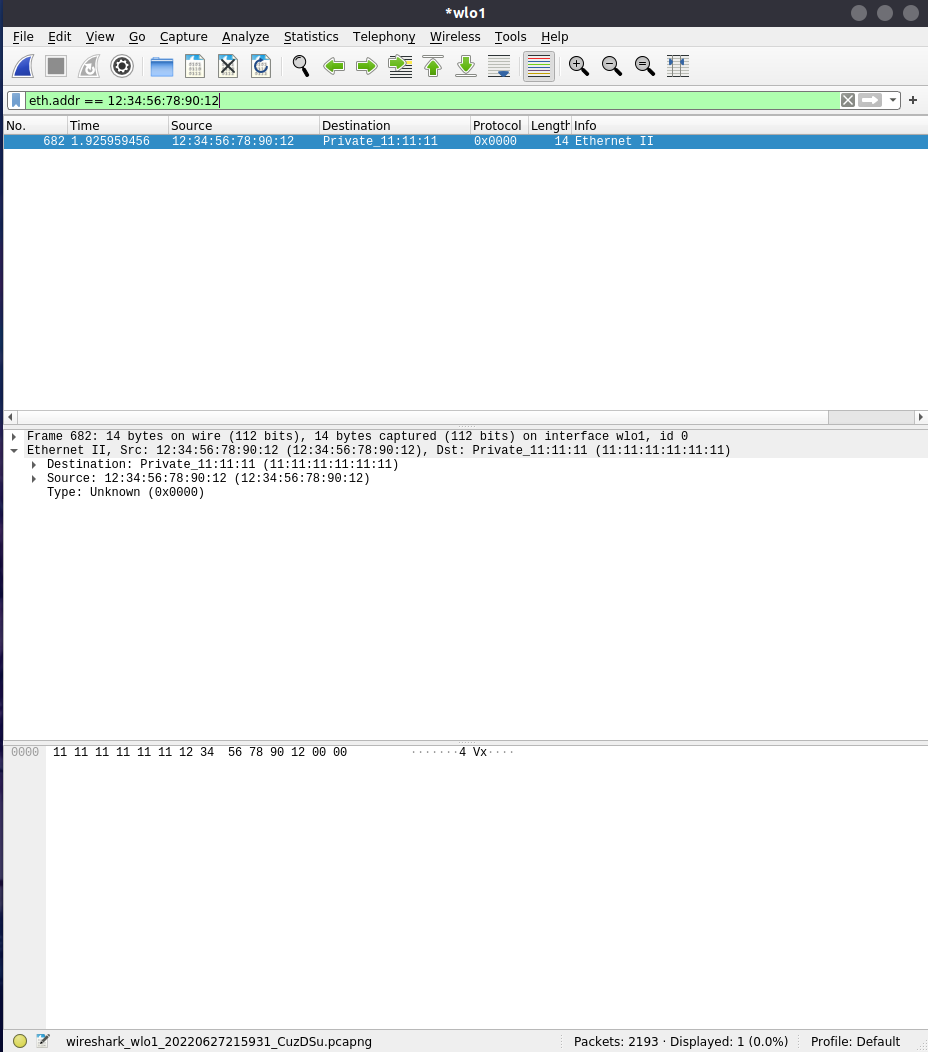
این تایپ از سوکت SOCK\_RAW به ما اجازه میدهد که مستقیما در لایه ی ip یا ethernet قرار بگیریم و بسته ها را ارسال کنیم. در نتیجه حداقل طولی که لازمه یه بسته داشته باشه 14 بایته و کمتر از اون رو ارور میده. چون اگر یه مسیج رندوم به این سوکت بدیم تا منتقل کنه خودش میاد 14 بایت اول رو به عنوان ethernet header در نظر میگیره.(طول هدر اترنت 14 بایته)

## 2-2:

بسته مون باید هدر اترنتش هدر ولیدی باشه. در نتیجه با توجه به نحوه ورودی گرفتن ما که یک رشته هگزادسیمال ورودی میگیره باید به تعداد زوج تا کاراکتر و حداقل 28 تا کاراکتر وارد کنیم که هر کاراکتر عددی بین 0 تا 9 یا حرفی بین a تا f است. اینکه قسمت type عدد ولیدی درنیاد مهم نیس چون وایرشارک میشناسه پکتو فقط type اش unknown در نظر گرفته میشه.

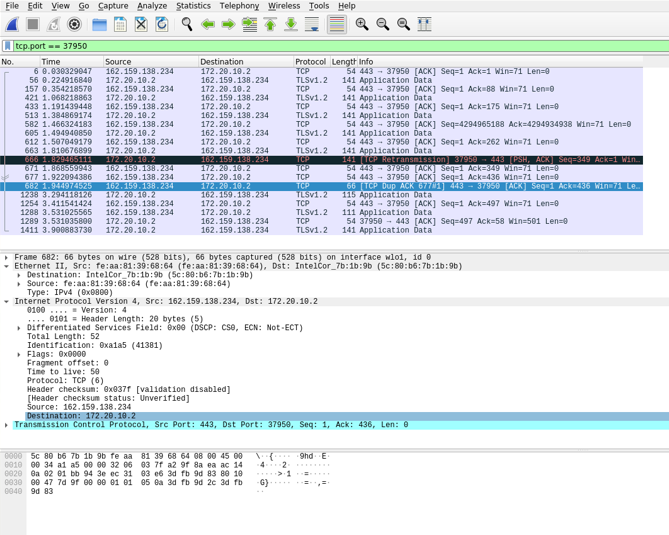
## 2-3:





## 2-4:





پکتی که به عنوان duplicated ack شناخته شده پکتیه که ما مجددا با همون seq num و ack num فرستادیم به همین دلیل dup ack در نظر گرفته شده.

## 2-5:

در این حمله، فرد هکر با شیوه های مختلف ip spoofing و sniff کردن پکت ها به محتوای ریکوستی که فرضا A برای web server میفرستد دست پیدا میکند و بدون اینکه نیاز باشد محتوای ان را بخواند یا رمزگشایی کند ان را برای سرور retransmit(replay) میکند. این موضوع باعث میشور سرور به اشتباه بیفتد و تصور کند پیام از سمت شخص A است که مثلا قصد داره دوباره لاگین کند. در نتیجه دیتا را برای فرد هکر میفرستد.

کاری که ما در سوال قبل انجام دادیم به نوعی یجور replay attack بود.

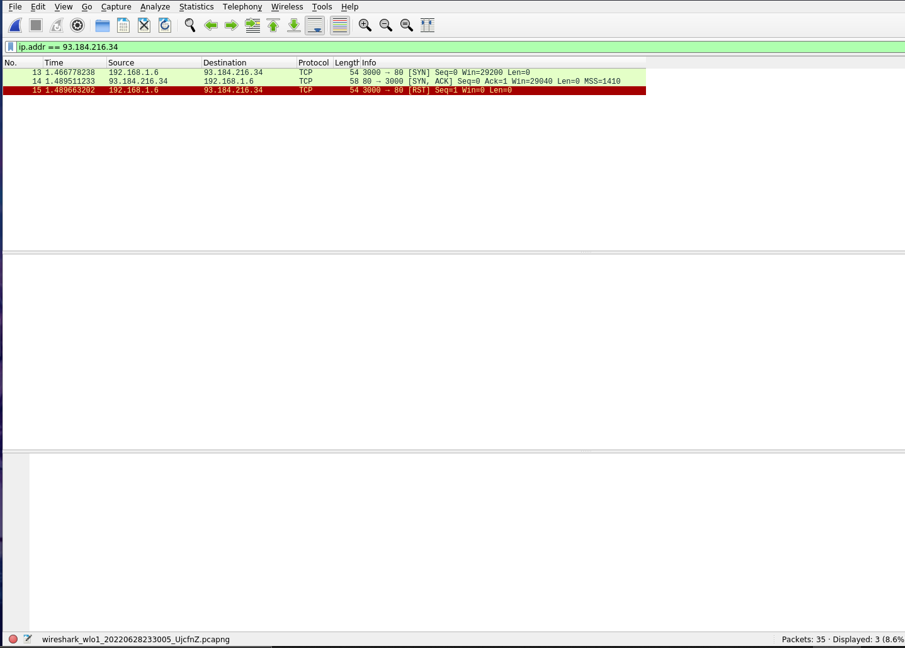
# Part 3:

## 3-1:

مقادیری که نمیتونیم دلخواه مقدار دهی کنیم یکی مقادیر مربوط به مبدا هست, مقادیری که به tcp\_syn بودن بسته مربوط است و مقادیری مثل چک سام که وابسته به بقیه فیلد ها مفدار میگیرند. لذا مقادیر قابل تغییر توی شکل عبارتند از:

Dest\_mac, proto3, ver, diff , t\_len, id, ttl , dest\_ip , dest\_port, seq\_num, w\_size, up,

## 3-2:



3-3:

این بسته ها قسمتی از فرایند triple handshaking را نشان میدهند. بسته ی اول همون بسته ایه که ما ارسال کرده ایم. بسته دوم بسته syn-ack است که از سمت سرور ارسال شده است. بسته ی سوم قاعدتا باید بسته ack باشد که ما برای سرور ارسال کرده ایم ولی فلگ rst دارد. دلیلش احتمالا این است که ما در سمت خودمون دیگه اک رو دریافت نمیکنیم چون قصد برقراری ارتباط tcp را نداریم. لذا کانکشن ایجاد شده داون میشود و به همین دلیل فلگ rst برابر 1 مقدار گرفته است.

## 3-4:

فایل ifinfo.sh ضمیمه شده است.

# Part 4:

## 4-1:

14 Bytes

## 4-2:

14 Bytes 😐

اگر منظور هدر tcp باشه 20 بایته.

## 4-3:

با چک کردن فلک های SYN , ACK در هدر tcp بسته ی موردنظر.



## 4-4:

Tcp\_syn\_sender sends a packet:



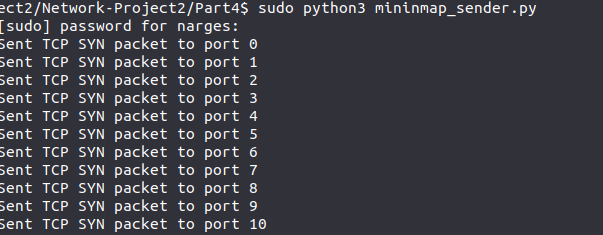
Mini-wireshark sniffes it:



# Part 5:

## 5-1:

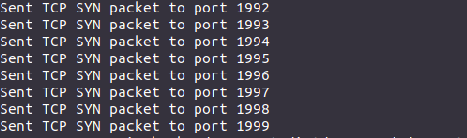
Mininmap :



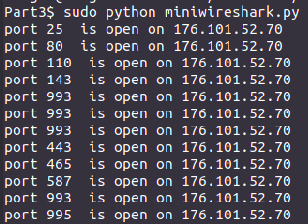
.

.

.



miniwireshark:



## 5-2:

باتوجه به نتیجه قسمت قبل پورت‌های ۲۵، ۸۰، ۱۱۰، ۱۴۳، ۴۴۳، ۴۶۵، ۵۸۷، ۹۹۳، ۹۹۵ روی آدرس ۱۷۲.۱۰۱.۵۲.۷۰ متعلق به دانشگاه صنعتی اصفهان باز هستند که متعلق به سرویس‌های زیر هستند.

۲۵ : Simple Mail Transfer Protocol (SMTP)

۸۰ : Hypertext Transfer Protocol (HTTP)

۱۱۰ : Post Office Protocol, version 3 (POP3)

۱۴۳ : Internet Message Access Protocol (IMAP)

۴۴۳ : HTTP Secure (HTTPS)

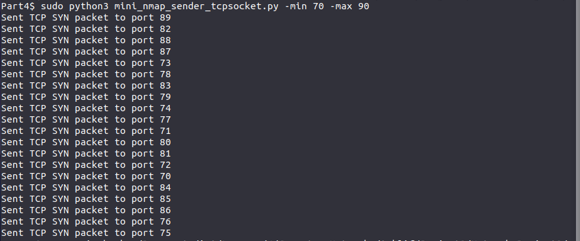
۴۶۵ : for email client to email server communication

۵۸۷ : email message submission

۹۹۳ : Internet Message Access Protocol (IMAPS)

۹۹۵ : Post Office Protocol 3 (POP3S)

## 5-3:



این بار برای هر کانکشن یک سوکت جدا ساختیم که به شکل پارالل در چند ترد جدا دیتا ارسال میکنند. به همین خاطر همونطور که در شکل بالا پیداست بسته ها لزوما به ترتیب پورت ارسال نشده اند. از طرف دیگه با توجه به تصویر وایرشارک که در صفحه بعد قرار داده شده است، اختلاف زمان بین بسته های متوالی بسیار کم است و این به همان همزمان ارسال شدن بسته ها برمیگردد.

این مثال نشون میده که از پورت های رنج 90 – 70 فقط پورت 80 دانشکاه باز است که منطقیه.

به طور کلی هرچند در زمان خیلی صرفه جویی شد ولی هزینه ساختن هر ترد برای هر سوکت ممکنه یکم محدودیت نرم اقزاری ایجاد کنه.

