«به نام پروردگار»

گزارش پروژه اول شبکه های کامپیوتری

«Packet sniffer»

استاد: دکتر پویا حجازی

دانشجو: محسن محمدیان

شماره دانشجویی: 9831502

بررسی اجمالی:

این پروژه شامل دو بخش می باشد. یک بخش تئوری که ما را با ساختار پروتکل های لایه ی transport و network شبکه و مفاهیمی چون IPSec بیشتر آشنا خواهد کرد و دیگر بخش عملی آن که در آن به پیاده سازی یک packet sniffer ساده می پردازیم.

در بخش برنامه نویسی پروژه قصد داریم یک شنود کننده ی بسته های مختلف را پیاده سازی کنیم. به این صورت که پس از اجرای برنامه ما قادر باشیم تا مشخصات بسته هایی را که از اینترنت دریافت می کنیم، مشاهده کنیم و در آخر یک فایل متن (txt) که نمایش دهنده یکسری اطلاعات کلی مانند تعداد کل بسته های تبادل شده برای پروتکل های ICMP, TCP, UDP، آدرس IP فرستنده ها به صورت نزولی و تعداد بسته هایی که fragment شده اند به عنوان خروجی برنامه داده خواهد شد.

بخش اول:

- قالب HEADER بسته های پروتکل های زیر را با رسم شکل بیان کنید و بگویید وظیفه ی هر FIELD چیست. (برای پاسخ به این پرسش می توانید از RFCهای 2473, 793, 791, 768 کمک بگیرید.)
  - IPv_4

0 3 4 7 8 15 16 18 31

Version	IHL	Type of service	**Total Length**
Identification		Flags	Fragment offset
Time to live	**Protocol**	Header Checksum	
Source Address			
Destination Address			
Option		**Padding**	
Data			

:Version

این فیلد فرمت هدر اینترنت را در wordهای 32 بیتی مشخص می کند. در اینجا مثلا ورژن 4، IP هست(هدرها در چه فرمتی هستند، IPv_4 یا IPv_6).(4 بیت)

:IHL

طول هدر اینترنت را بر حسب wordهای 32 بیتی مشخص می کند و به نوعی نشاندهنده ی آن هست که data از کجا آغاز می شود در یک datagram. کمترین طول هدر برای یک datagram، برابر 5 می باشد. یعنی پنج word ابتدایی header اجباری هستند.(4 بیت)

:Type of services

برای تشخیص انواع datagramها از یکدیگر بکار می رود. برای مثال میتوان با آن datagram های real-time را از ترافیک های non-real-time تشخیص داد. همچنین دو بیت از آن نشاندهنده ی ازدحام در شبکه خواهد بود.(8 بیت)

این فیلد ساختاری مانند زیر دارد که سه بیت نخست آن اولویت data gram را مشخص می کند و سه بیت دوم type آن را.

```
Bits 0-2:  Precedence.
Bit    3:  0 = Normal Delay,      1 = Low Delay.
Bits   4:  0 = Normal Throughput, 1 = High Throughput.
Bits   5:  0 = Normal Relibility, 1 = High Relibility.
Bit  6-7:  Reserved for Future Use.
```

0 2 3 4 5 6 7

PRECEDENCE	D	T	R	0	0

```
Precedence

          111 - Network Control
          110 - Internetwork Control
          101 - CRITIC/ECP
          100 - Flash Override
          011 - Flash
          010 - Immediate
          001 - Priority
          000 - Routine
```

:Total length

طول datagram را برحسب بایت (واحدها هشت تایی یا octet) نشان می دهد و این طول شامل طول هدر اینترنت (IP) و طول data می باشد. این فیلد نشان می دهد که طول datagram می تواند حداکثر برابر با 65, 535 باشد.(16 بیت)

:Identification

این فیلد به قطعات یک datagram هویت می دهد و مشخص می کند که هر قطعه متعلق به کدام datagram ای می باشد.(16 بیت)

:Flags

وضعیت قطعات یک datagram را مشخص می کند که آیا fragment دیگری برای آن datagram هنوز مانده تا ارسال شود یا خیر.(3 بیت)

```
Bit 0: reserved, must be zero
Bit 1: (DF) 0 = May Fragment,  1 = Don't Fragment.
Bit 2: (MF) 0 = Last Fragment, 1 = More Fragments.
```

اگر DF صفر باشد، یعنی ممکن است هنوز قطعه ای برای ارسال باشد. اگر یک شود، یعنی قطعه ی جاری آخرین قطعه از datagram فعلی است.

اگر MF صفر باشد، یعنی قطعه ی جاری آخرین قطعه از datagram فعلی است؛ و اگر یعنی ممکن است هنوز قطعه ای برای ارسال باشد.

:Fragment offset

مشخص می کند هر قطعه، قطعه ی چندم یک datagram هست و در کجای آن قرار دارد.(13 بیت)

:Time to live

این فیلد مشخص می کند که چقدر یک datagram اجازه دارد در سیستم اینترنت بماند و معتبر باشد. اگر مقدار این فیلد صفر شود، datagram باید نابود شود. این فیلد در هدر اینترنت تغییر می کند. زمان را ما برحسب ثانیه حساب می کنیم اما هر ماژول در اینترنت یا روتر که روی datagram پردازش انجام می دهد، مقدار TTL را یک واحد کاهش می دهد؛ حتی اگر زمان پردازش روی آن کمتر از یک ثانیه طول کشیده باشد.(8 بیت)

:Protocol

این فیلد معمولا هنگامی استفاده می شود که یک datagram به مقصد خود رسیده باشد و مشخص می کند که بخش داده ی این datagram به کدام پروتکل لایه ی transport باید منتقل شود. برای مثال مقدار 6 مشخص می کند که data باید به پروتکل TCP تحویل داده شود و مقدار 17 مشخص می کند که به UDP باید تحویل داده شود. مانند شماره پورت در لایه ی network می باشد و لایه ی network و transport را به نوعی به یکدیگر bind می کند.(8 بیت)

:Header CheckSum

یک CheckSum روی wordهای هدر می باشد. این مقدار در هر نقطه ای که هدر اینترنت پردازش می شود، محاسبه و اعتبار سنجی می شود.(16 بیت)

:Source Address

آدرس مبدا را مشخص می کند.(32 بیت)

:Destination Adrees

آدرس مقصد را مشخص می کند.(32 بیت)

:Options

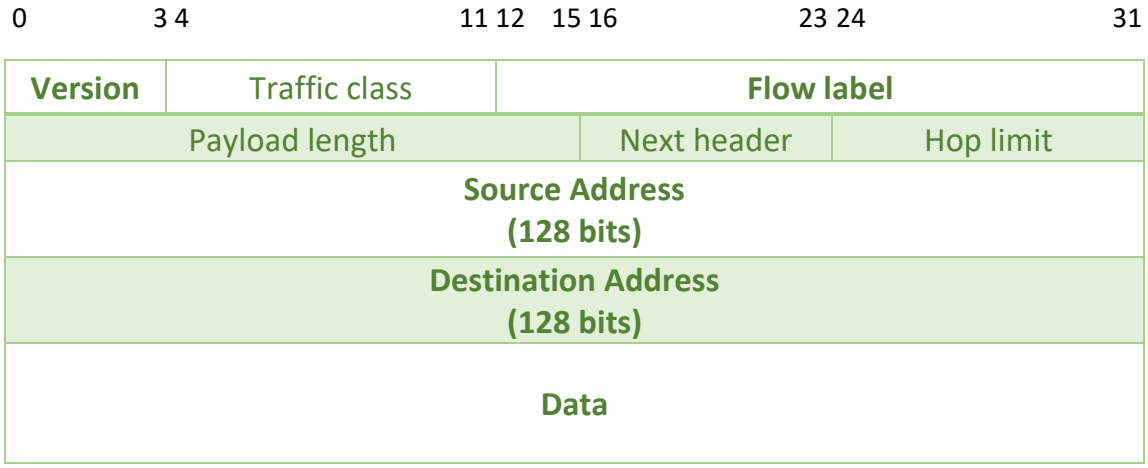
می توانند در datagram ظاهر شوند یا نشوند و باید توسط ماژول های IP پیاده سازی شوند(hostها و gatewayها) چیزی که اختیاری است، انتقال آنها در یک datagram بخصوص است نه پیاده سازی آنها.

در برخی از محیط ها، آپشن های امنیتی، ممکن است در تمامی datagramها نیاز شود.

دو مدل فرمت برای یک option داریم:
Case1: یک تک بایت از نوع option
Case2: یک بایت از نوع option، یک بایت از طول option و درنهایت طول dataهای اصلی option.

Option-lenght هردو بایت option-kind و option-length را هم می شمارد.

- IPv_6



:Version

این فیلد فرمت هدر اینترنت را در wordهای 32 بیتی مشخص می کند. در اینجا مثلا ورژن 6، IP هست(هدرها در چه فرمتی هستند، IPv_4 یا IPv_6).(4 بیت)

:Traffic class

مانند Type of Service در IPv_4 می باشد. می تواند به datagramهای درون یک جریان از داده ها، اولویت دهد؛ یا می تواند برای اولویت دادن به datagramهایی که از applicationهای ویژه (مانند voice-over-IP) نسبت به datagramهای سایر appliacationها استفاده شود.(8 بیت)

:Flow table

در IPv_6 ما مفهومی به نام flow را داریم که یک شبه ارتباط میان مبدا و مقصد برقرار می شود و بسته ها label گذاری می شوند تا مشخص شوند متعلق به کدام flow بخصوصی می باشند که فرستنده درخواست handling آن ها را می دهد؛ مانند non-default quality service یا real-time service. برای مثال ممکن است با audio و video مانند flow رفتار شود ولی برنامه هایی چون e-mail و file transfer با آنها به عنوان flow رفتار نمی شود.

درواقع برای تفکیک datagramهای متعلق به یک flow در روترها بکار می رود تا بتوان آنها را از هم تمییز داد و به آنها اولویت ارسال داد.(20 بیت)

:Payload length

این مقدار یک unsigned integer است که طول بخش data را برحسب بایت مشخص می کند و به دنبال 40 بایت ثابت طول datagram header می آید.(16 بیت)

:Next header

مشخص می کند که داده های این datagram به کدام پروتکل منتقل شود.(برای مثال TCP یا UDP) این فیلد همان مقادیر یکسانی را استفاده می کند که در IPv_4 استفاده می شد. همچنین option headerها از طریق این فیلد قابل دسترسی هستند(به ابتدای هدر پروتکل لایه ی بالاتر اشاره می کند).(16 بیت)

:Hop limit

شبیه TTL در IPv_4 می باشد؛ محتوای این فیلد در هر گام (هربار که datagram توسط یک روتر forward می شود)، یک عدد واحد می یابد و هنگامی که به صفر برسد، datagram نابود می شود.(16 بیت)

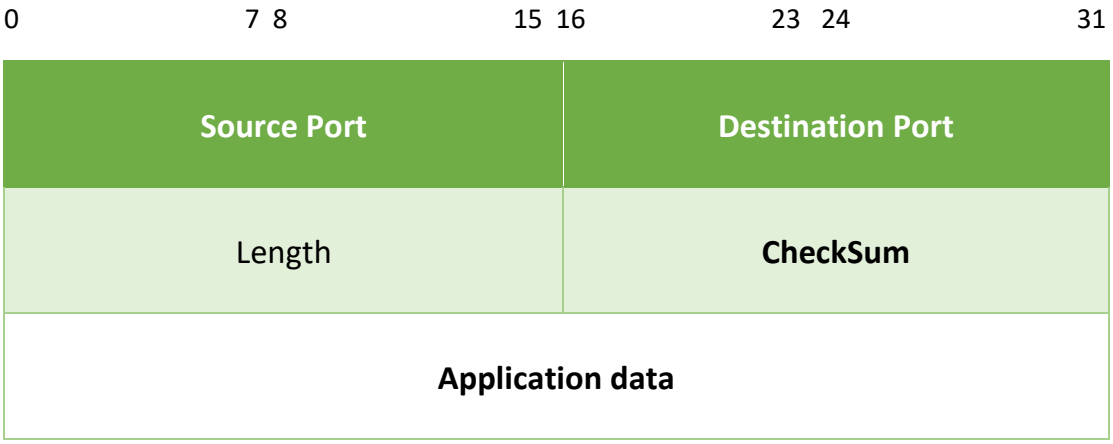
:Source Address

آدرس مبدا را مشخص می کند و 128 بیت ی باشد.

:Destination Address

آدرس مقصد را مشخص می کند و 128 بیت می باشد.

- UDP



:Source Port

این فیلد، یک فیلد اختیاری است و پورت فرآیند (process) ارسال کننده را مشخص می کند. و در غیاب اطلاعات دیگر، گیرنده آن را به عنوان پورتی که باید به آن reply کند درنظر می گیرد. درصورت عدم استفاده، مقدار صفر می گیرد.

:Destination Port

پورت مقصد را مشخص می کند و درون محتوای آدرس مقصد در اینترنت معنا پیدا می کند.

:Length

طول بسته را برحسب بایت، شامل سرآیندها (headers) و داده ها مشخص می کند.

:CheckSum

یک فیلد 16 بیتی حاصل از جمع one's complement بیت های شبه سرآیند (pseudo header) اطلاعات موجود در IP header و UDP header و data می باشد. (اگر حاصل کمتر از 16 بیت برای نمایش شود، 8 بیت صفر در کنار آن concate می شود)

Pseudo header:

به عنوان یک پیشوند (prefix) به سرآیند پروتکل UDP اضافه می شود و شامل آدرس مبدا و مقصد، پروتکل و طول بسته می باشد. این اطلاعات از گم شدن بسته ها جلوگیری می کنند.

0	7 8	15 16	23 24	31

Source Address		
Destination Address		
Zero	Protocol	**UDP Length**

- TCP

0									15 16	31
Source Port									**Destination Port**	
Sequence Number										
Acknowledgement Number										
Data offset	Reserved	C W R	E C E	U R G	A C K	P C H	R S T	S Y N	F I N	**Window**
CheckSum									**Urgent Pointer**	
Options										**Padding**
data										

Source Port:
این فیلد پورت مبدا را مشخص می کند.(16 بیت)

Destination Port:
این فیلد پورت مقصد را مشخص می کند. (16 بیت)

Sequence number:
نشان دهنده ی شماره ترتیب (seq #) یا byte-stream number اولین بیتِ بایتِ اول یک segment می باشد. به غیر از زمانی که SYN flag تنظیم شده باشد؛ اگر SYN، set شده باشد، sequence

number برابر initial sequence number(ISN) خواهد بود و شماره اولین بایت data برابر ISN+1. (32 بیت)

Acknowledgement Number:
اگر بیت ACK تنظیم شده باشد، این فیلد شامل مقدار شماره ترتیب (# seq) بعدی ای خواهد بود که ارسال کننده انتظار دریافت آن را دارد.(32 بیت)

Data offset:
تعداد wordهای 32 بیتی را در هِدِر TCP مشخص می کند. این فیلد درواقع مشخص می کند data ی یک segment از کجا آغاز می شود.(4 بیت)

Control bits:
Urgent Pointer field significant(URG):
نشاندهده ی آن هست که segment حاوی data ای است که در لایه بالاتر transport در سمت فرستنده، به عنوان urgent، mark شده است.

Acknowledgment field significant(AFK):
نشان دهنده ی آن است که مقداری که در فیلد acknowledgement قطعه ی منتقل شده قرار داشته، معتبر بوده است.

Push Function(PSH):
Set شدن این بیت نشان دهنده ی آن است که گیرنده باید به سرعت داده را به لایه ی بالاتر منتقل کند.

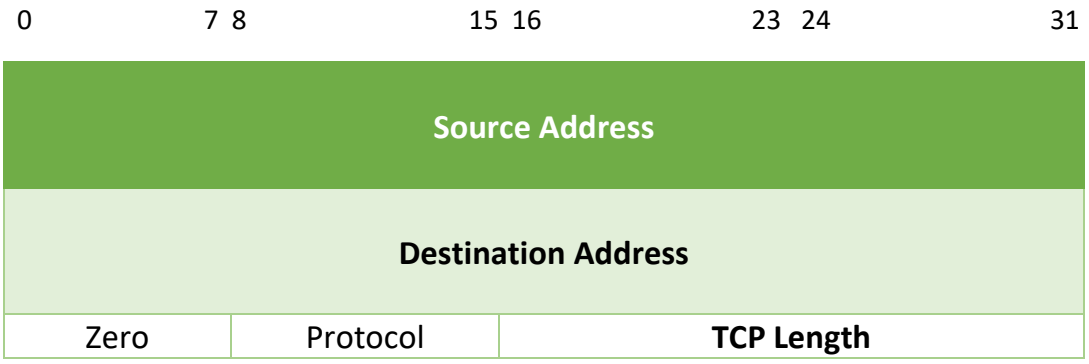
Reset the connection(RST):
برای reset کردن connection در زمان هایی که رفتار فرستنده نامعقول باشد بکار می رود.

Synchronize sequence numbers(SYN):
برای همگام سازی ارتباط و در زمان ایجاد ارتباط و setup connection، مورد استفاده قرار می گیرد.
No more data from sender(FIN):
برای قطع ارتباط در زمانی که فرستنده دیگر داده ای برای ارسال ندارد بکار می رود.

CWR و ECE:
برای هشدار صریح در هنگام ازدحام (congestion) بکار می روند.

Window:
نشان دهنده تعداد بایت هایی است که گیرنده می تواند آنها را دریافت کند.(16 بیت)

:CheckSum
حاصل جمع one’s complement تمام بیت های هدر و متن پیام می باشد. اگر تعداد bit wordهای هدر و متن فرد باشد، یک بایت تمام صفر به سمت راست این فیلد pad می شود تا فرم 16 بیتی آن حفظ شود. توجه شود خود قسمت padding به عنوان یک بخش از segment ارسال نمی شود و تنها در هنگام محاسبه Checksum بایت سمت راست آن با صفر جایگزین خواهد شد.(16 بیت)
همچنین مکانیزم CheckSum شبه هدر 96 بیتی را نیز پوشش می دهد.
مانند شبه هدر UDP هست:



:Urgent pointer
به مکان آخرین بایت urgent data اشاره می کند.(16 بیت)

:Options
Optionها یک سری هدر اختیاری هستند و شامل یک یا چند بایت می شوند و در CheckSum نیز محاسبه می شوند.(تعداد بیت ها متغیر)
دو مدل فرمت برای یک option داریم:
Case1: یک تک بایت از نوع option
Case2: یک بایت از نوع option، یک بایت از طول option و درنهایت طول dataهای اصلی option.
Option-lenght هردو بایت option-kind و option-length را هم می شمارد.

- با جستوجو در اینترنت درباره Transport Layer Security تحقیق کنید. کاربردهای آن را بنویسید و جایگاه آن در مدل لایه ای را تشریح کنید.

این لایه بیان می کند که چگونه رمزنگاری، سبب بهبود پروتکل TCP از لحاظ سرویس های امنیتی مانند: محرمانگی (confidentiality)، تمامیت داده(data-integrity) و احراز هویت مقصد می شود. چون پروتکل TCP خود به تنهایی امنیت بسته ها را پشتیبانی نمی کند. این ورژن بهبودیافته ی TCP تحت عنوان Secure Sockets Layer (SSL) شناخته می شود؛ و ورژن اندک تغییر یافته ی آن نیز Transport Layer Security(TLS) نام دارد.

SSL توسط تمامی وب سرورها و web browserهای محبوب پشتیبانی می شود و Gmail و سایت هایی تجاری بزرگی نظیر Amazon و eBay از آن استفاده می کنند. درواقع هنگامی که ما یک خرید از اینترنت با کارت اعتباری خود انجام می دهیم، تعامل browser ما با مقصد روی SSL انجام می شود.

هنگامی که در browser پروتکل http بجای https در URL نمایش داده می شود، نشاندهنده ی آن است که از SSL دارد استفاده می شود.

وظایف و کاربردها:

Confidentiality:

از شنود اطلاعات و بسته ها توسط متجاوزین جلوگیری می کند.(برای مثال از شنود کردن اطلاعات کارت اعتباری یک شخص درهنگام خرید توسط یک هکر جلوگیری میکند)

Data integrity:

تمامیت داده ها را حفظ می کند و از تغییر و تکثیر آنها جلوگیری می کند.

server authentication:

از دزدیدن هویت یک وب سایت، توسط یک وب سرور جعلی، جلوگیری می کند.(مثلا یک سایت خودش را جای دیجی کالا جا بزند و از خریدهای مردم پول دریافت کند)

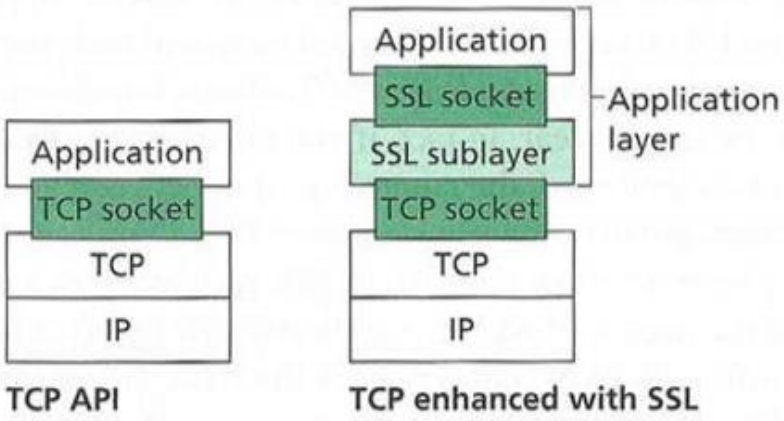
تمامی برنامه هایی که از TCP استفاده می کنند، میتوانند از SSL هم استفاده کنند. SSL یک API ساده مانند TCP API دارد که socketها را دربر می گیرد. وقتی یک application می خواهد از SSL استفاده کند، باید کلاس ها و کتابخانه های آن را include یا import کند.

- ✓ برای امنیت تراکنش های آنلاین توسط کارت های اعتباری بکار می رود.
- ✓ برای ایمن کردن loginها و هر اطلاعات حساس دیگری که در اینترنت رد و بدل می شود.
- ✓ برای ایمنی انتقال فایل ها، روی سوریس های https یا FTPs؛ مانند مالک های وب سایت ها که صفحات جدیدی را در وب سایت های خود اضافه می کنند یا فایل های حجیم را منتقل می کنند.
- ✓ برای ایمنی workflow applicationها و برنامه های مجازی سازی بکار می رود.
- ✓ برای ایمن سازی ارتباط میان یک کلاینت ایمیل مانند Microsoft Outlook و سرور ایمیل مانند Microsoft exchange نیز بکار می رود.

جایگاه آن در مدل لایه ای:

هنگامی که یک application از SSL استفاده می کند، process ارسال کننده، داده ی clear text خود را به سوکت SSL می دهد و SSL در host مبدا داده را رمز می کند و تحویل TCP میدهد.

در مقصد هم داده ی رمز شده را TCP می گیرد و تحویل SSL می دهد و SSL آن را decrypt یا رمزگشایی می کند و از طریق سوکت خود، آن را تحویل process مدنظر می دهد.



- در مورد IPSec تحقیق کنید.

1) IPSec چیست؟

یک پروتکل است که security یا امنیت را در لایه ی network مهیا می کند. IPsec، datagramها را میان هر دو ماژول(hostها و routerها) لایه ی شبکه، ایمن می کند.

2) کاربردهای آن را بنویسید.

کاربردهای آن در private network ها می باشد؛ هنگامی که یک سازمان می خواهد یک شبکه ی منفرد اختصاصی داشته باشد به دلیل مسائل امنیتی، باید تمامی زیرساخت های فیزیکی شبکه را پیاده سازی کند که این هزینه ی زیادی دارد. بجای آن از Virtual Private Network(VPN) استفاده می شود. که به کمک IPsec بخش payload یک datagram رمزگذاری می شود و یک IPsec header هم می گیرد و روی IPv_4 ارسال می شود.

3) مزایا و معایب استفاده از آن را مورد بررسی قرار دهید.

مزایا:

i. IPsec مستقل از برنامه ها است، پس تمام پروتکل های لایه ی network را پشتیبانی می کند.وقتی Ipsec روی یک روتر یا firewall نصب می شود، نیازی به تغییر تنظیمات نرم افزار در سیستم سرور یا کاربر نداریم.

ii. در تکنولوژی IPsec، تکنولوژی های مورد استفاده ی ارتباط client-to-site، site-to-client و client-to-client یکسان هستند.

iii. سومین مزیت آن ایمن سازی و رمز کردن تمامی کانال ها و داده های ارسالی می باشد.

معایب:

i. Communication performance آن پایین می باشد. به دلیل آنکه IPsec وابستگی زیادی به امنیت دارد و همین مورد روی performanceارتباطی آن تاثیر می گذارد.

ii. IPsec VPN نیازمند نرم افزار کلاینت هست. باید برای هر کلاینت یک نرم افزار به خصوص را نصب کنیم تا بتوان TCP/IP stack سیستم شخصی خودمان را پیاده کنیم. که این امر ممکن است باعث ناسازگاری سیستم با سایر نرم افزارها شود.

iii. مشکلات NAT و firewall traversal به راحتی حل نمی شوند. محصولات IPsec مشکلات دسترسی از راه دور، firewall traversal و ترجمه آدرس های شبکه را حل نمی کنند. برای مثال اگر شخصی IPsec client را نصب کرده باشد؛ ولی درون شبکه شرکت دیگری به اینترنت دسترسی نداشته باشد، IPsec توسط firewall شرکت مسدود خواهد ماند تا زمانی که شخص به ادمین شبکه اطلاع دهد تا یک پورت دیگر را روی firewall باز کند.

- در بسته های فیلد IP_{v4} مقدار فیلد Protocol چیست؟ و چه مقادیری می تواند داشته باشد؟

این فیلد معمولا هنگامی استفاده می شود که یک datagram به مقصد خود رسیده باشد و مشخص می کند که بخش داده ی این datagram به کدام پروتکل لایه ی transport باید منتقل شود. برای مثال مقدار 6 مشخص می کند که data باید به پروتکل TCP تحویل داده شود و مقدار 17 مشخص می کند که به UDP باید تحویل داده شود. مانند شماره پورت در لایه ی network می باشد و لایه ی network و transport را به نوعی به یکدیگر bind می کند.

برخی از مقادیر دیگر در جدول زیر آمده است:

protocol	value
LARP	**91**
GMTP	**100**
SMP	**121**
VRRP	**112**
ICMP	**1**
IGMP	**2**
RDP	**27**
ST	**5**
PRM	**21**
UDP	**17**
TCP	**6**

بخش دوم:

برنامه نویسی:

برای پیاده سازی packet sniffer در این پروژه، از لایه ی data link شروع کرده و بسته ها را دریافت می کنیم و در هر لایه بسته ها unpack شده و بایت های آن برای پیدا کردن فیلدهای موردنظر خوانده می شوند و لایه به لایه بالا می رویم تا به لایه ی application برسیم.

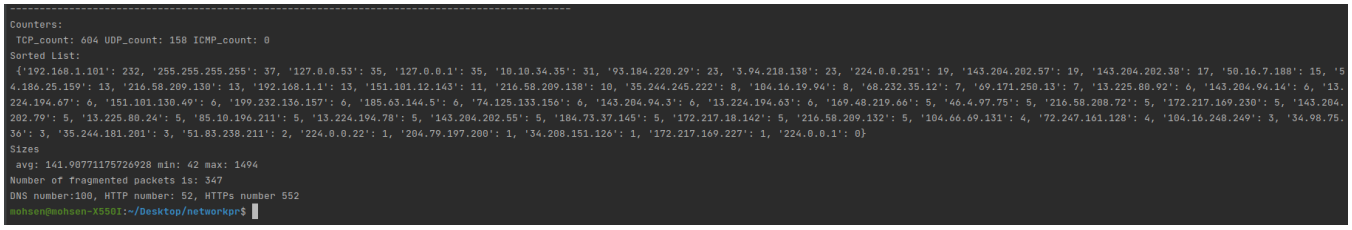
درواقع ما در هرلایه هدرهای آن لایه را از بسته برداشته و بخش payload یا dataی بسته را به لایه ی بالاتر پاس می دهیم.

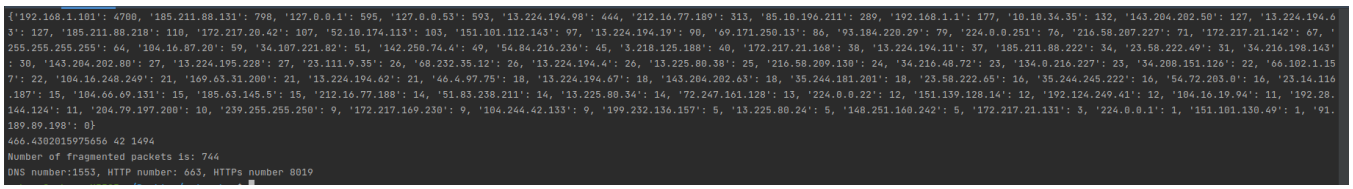
این برنامه اطلاعات مختلفی را به ما می دهد از قبیل:

1. تعداد کل بسته های تبادل شده ICMP, TCP, UDP.
2. آدرس IP فرستنده های بسته ها.
3. تعداد بسته هایی که قطعه بندی شده اند به همراه آنکه هر بسته به چند قطعه تبدیل شده است و ID آن چه می تواند باشد.
4. بیشترین، کمترین و میانگین سایز بسته ها.
5. تعداد بسته هایی که پروتکل آنها HTTP, HTTPs, DNS بوده چندتا است.

توجه: برای خروج از برنامه کافیست دستور "stop" در terminal linux به صورت lower case، تایپ شده و سپس Enter زده شود.(توجه شود که حتما دقت شود دستور stop به درستی وارد شود و از لحاظ املایی نیز غلط نداشته باشد)

نمونه هایی از نتیجه ی اجرای برنامه:





```
        00\x00\x00\x90\x85\x07\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18
        \x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x
        2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37
--------------------------------------------------------------------------------------------
p
Ethernet Frame:
        - Packet Size: 201
        - Destination: FF:FF:FF:FF:FF:FF, Source: D4:6E:0E:44:A1:A2, Protocol: 8
        - IPv4 Packet:
                - Version: 4, Header Length: 20, TTL: 64
                - Protocol: 17, Source Address: 192.168.1.1, Destination Address: 255.255.255.255
        - UDP Segment:
                - Source Port: 39779, Destination Port: 7437
                - Data:
                        \x4b\x41\x4e\x4e\x4f\x55\x25\x4e\x00\x00\x00\x00\x00\xd4\x6e\x0e\x44\xa1\xa2
                        \x54\x44\x2d\x57\x38\x39\x36\x38\x76\x35\x00\x00\x00\x00\x00\x00\x54\x44\x2d
                        \x57\x38\x39\x36\x38\x76\x35\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
                        \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
                        \x00\x00\x00\x00\x00\x00\x08\x00\x00\x00\x00\x31\x2e\x30\x32\x2e\x38\x33\x2e
                        \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00\x01
                        \x00\x00\x00\x02\x00\x00\x00\x02\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
                        \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
                        \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
                        \x00\x00
--------------------------------------------------------------------------------------------


Ethernet Frame:
        - Packet Size: 32
        - Destination: D4:6E:0E:44:A1:A2, Source: 40:9F:38:18:21:1D, Protocol: 8
        - IPv4 Packet:
                - Version: 4, Header Length: 20, TTL: 64
                - Protocol: 47, Source Address: 192.168.1.102, Destination Address: 46.105.134.229
        - Data:
        \x20\x81\x88\x0b\x00\x00\x9f\x0e\x00\x00\xa6\xb2
--------------------------------------------------------------------------------------------
20 98 0
{'192.168.1.102': 3287, '46.105.134.229': 2718, '255.255.255.255': 51, '127.0.0.53': 13, '127.0.0.1': 13, '224.0.0.251': 12, '46.4.97.75': 5, '239.255.255.250': 4, '224.0.0.22': 4, '224.0.0.1': 0}
436.2861215036522 40 1454
15
mohsen@mohsen-X550I:~/Desktop/networkpr$
```

```
Ethernet Frame:
        - Packet Size: 120
        - Destination: D4:6E:0E:44:A1:A2, Source: 40:9F:38:18:21:1D, Protocol: 8
        - IPv4 Packet:
                - Version: 4, Header Length: 20, TTL: 64
                - Protocol: 47, Source Address: 192.168.1.102, Destination Address: 46.105.134.229
        - Data:
        \x30\x01\x88\x0b\x00\x58\x9f\x0e\x00\x00\x7f\xdf\xff\x03\x00\x21\x45\x00\x00\x
        54\x23\x96\x40\x00\x40\x01\xc2\xbb\xac\x1e\x00\xb0\xd8\x3a\xcf\x4e\x08\x00\xc9
        \xd2\x26\xc2\x00\x5f\xc1\x53\xef\x5f\x00\x00\x00\x00\x90\x85\x07\x00\x00\x00\x
        00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21
        \x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x
        35\x36\x37
--------------------------------------------------------------------------------------------
o
Ethernet Frame:
        - Packet Size: 124
        - Destination: 40:9F:38:18:21:1D, Source: D4:6E:0E:44:A1:A2, Protocol: 8
        - IPv4 Packet:
                - Version: 4, Header Length: 20, TTL: 46
                - Protocol: 47, Source Address: 46.105.134.229, Destination Address: 192.168.1.102
        - Data:
        \x30\x01\x88\x0b\x00\x58\x17\xf1\x00\x00\xa6\xb2\x00\x00\x7f\xdf\xff\x03\x00\x
        21\x45\x00\x00\x54\x00\x00\x00\x00\x00\x72\x01\xf4\x51\xd8\x3a\xcf\x4e\xac\x1e\x00
        \xb0\x00\x00\xd1\xd2\x26\xc2\x00\x5f\xc1\x53\xef\x5f\x00\x00\x00\x00\x90\x85\x
        07\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d
        \x1e\x1f\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x
        31\x32\x33\x34\x35\x36\x37
--------------------------------------------------------------------------------------------

Ethernet Frame:
        - Packet Size: 70
        - Destination: 45:00:00:54:00:00, Source: 00:00:72:01:F4:51, Protocol: 15064
Data:
        \xcf\x4e\xac\x1e\x00\xb0\x00\x00\xd1\xd2\x26\xc2\x00\x5f\xc1\x53\xef\x5f\x00\x
        00\x00\x00\x90\x85\x07\x00\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18
        \x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x
        2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37
--------------------------------------------------------------------------------------------
```

```
        - Data:
        \x30\x81\x88\x0b\x00\x38\x17\xf1\x00\x01\x1e\xcb\x00\x00\xe0\x55\xff\x03\x00\x
        21\x45\x00\x00\x34\xa4\x6a\x40\x00\xef\x06\x81\xd3\xd1\xa7\xe7\x0f\xac\x1e\x00
        \xb0\x01\xbb\xad\x46\x99\x74\x34\xab\xf8\x6d\x50\xad\x80\x10\x3f\x7d\x64\x03\x
        00\x00\x01\x01\x08\x0a\xbe\xf5\xaf\x0c\x6d\xba\xcb\xbd
--------------------------------------------------------------------------------------------
21 123 0
{'192.168.1.102': 2320, '46.105.134.229': 2137, '255.255.255.255': 78, '224.0.0.251': 19, '127.0.0.53': 11, '127.0.0.1': 11, '46.4.97.75': 5, '224.0.0.22': 3, '224.0.0.1': 1}
362.49219875180003 40 1454
Number of fragmented packets is: 12
DNS number:24, HTTP number: 11, HTTPs number 0
```

```
{'192.168.1.101': 4700, '185.211.88.131': 798, '127.0.0.1': 595, '127.0.0.53': 593, '13.224.194.98': 444, '212.16.77.189': 313, '85.10.196.211': 289, '192.168.1.1': 177, '10.10.34.35': 132, '143.204.202.50': 127, '13.224.194.6
3': 127, '185.211.88.218': 110, '172.217.20.42': 107, '52.10.174.113': 103, '151.101.112.143': 97, '13.224.194.19': 90, '69.171.250.13': 86, '93.184.220.29': 79, '224.0.0.251': 76, '216.58.207.227': 71, '172.217.21.142': 67,
255.255.255.255': 64, '104.16.87.20': 59, '34.107.221.82': 51, '142.250.74.4': 49, '54.84.216.236': 45, '3.218.125.188': 40, '172.217.21.168': 38, '13.224.194.11': 37, '185.211.88.222': 34, '23.58.222.49': 31, '34.216.198.143'
: 30, '143.204.202.80': 27, '13.224.195.228': 27, '23.111.9.35': 26, '68.232.35.12': 26, '13.224.194.4': 26, '13.225.80.38': 25, '216.58.209.130': 24, '34.216.48.72': 23, '134.0.216.227': 23, '34.208.151.126': 22, '66.102.1.15
7': 22, '104.16.248.249': 21, '169.63.31.200': 21, '13.224.194.62': 21, '46.4.97.75': 18, '13.224.194.67': 18, '143.204.202.63': 18, '35.244.181.201': 18, '23.58.222.65': 16, '35.244.245.222': 16, '54.72.203.0': 16, '23.14.116
.187': 15, '104.66.69.131': 15, '185.63.145.5': 15, '212.16.77.188': 14, '51.83.238.211': 14, '13.225.80.34': 14, '72.247.161.128': 13, '224.0.0.22': 12, '151.139.128.14': 12, '192.124.249.41': 12, '104.16.19.94': 11, '192.28.
144.124': 11, '204.79.197.200': 10, '239.255.255.250': 9, '172.217.169.230': 9, '104.244.42.133': 9, '199.232.136.157': 5, '13.225.80.24': 5, '148.251.160.242': 5, '172.217.21.131': 3, '224.0.0.1': 1, '151.101.130.49': 1, '91.
189.89.198': 0}
466.43020159756560 42 1494
Number of fragmented packets is: 744
DNS number:1553, HTTP number: 663, HTTPs number 8019
```