به نام پروردگار هدایت کننده به راه راست

دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

ترم تحصیلی ۰۲-۰۳

مستند پروژه درس شبکه های کامپیوتری

استاد درس : دکتر احمدرضا منتظرالقائم

طراحان : امیرعلی گلی، محمدحسین دهقانی، مهرشاد جعفری، مهدی قنبرزاده، محمدحسین رنگرز

این پروژه، شامل سه فاز است که هرکدام یک مینی پروژه برای آشنایی با ابزارها و پروتکل‌های کاربردی شبکه است.

در فاز اول شبیه سازی نرم افزار nmap پیاده سازی می‌شود که با مفاهیم پایه شبکه مثل پورت‌ها، پکت‌ها آشنا شده و با استفاده از برنامه نویسی با سوکت پیاده سازی می‌شود.

در فاز دوم پیاده‌سازی ساده‌تر پروتکل FTP انجام شده که این فاز نیست با استفاده از برنامه نویسی سوکت انجام می‌گیرد.

در فاز سوم کار با ابزار wireshark انجام خواهد شد. این فاز نیازی به استفاده از زبان های برنامه‌نویسی نیست .

برای فازهای اول و دوم این پروژه با استفاده از زبان‌های برنامه نویسی java, C++, Python, C#

پروژه اول: ابزار Nmap

Nmap یک ابزار بسیار قدرتمند است که توسط مدیران شبکه (Network Administrator) ، متخصصان امنیت (Security Expert) و حتی هکرها (Hacker) برای کاوش، بررسی و درک بهتر شبکه‌های کامپیوتری مورد استفاده قرار می‌گیرد.

نام این ابزار مخفف شده عبارت "Network Mapper" است. این ابزار به کاربر کمک می‌کند تا دستگاه‌هایی (Device) که در یک شبکه کامپیوتری فعال هستند را پیدا کند سرویس‌ها و برنامه‌هایی که روی آن دستگاه‌ها در حال اجرا هستند را شناسایی کند و حتی مواردی را که از لحاظ امنیتی، آسیب پذیر (Vulnerable) هستند را مشخص کند.

برای کسب اطلاعات بیشتر در مورد این ابزار می‌توانید به این لینک مراجعه کنید.

تعاریف مورد نیاز

ممکن است در حین خواندن این داکیومنت به یک سری تعاریف نیاز پیدا کنید. برای سادگی کار شما برخی از آن تعاریف آورده شده‌اند:

- **هاست (Host):** در مفهوم شبکه‌های کامپیوتری، هاست به دستگاه یا سیستمی اشاره دارد که قادر است به شبکه متصل شود و در شبکه‌ای حضور دارد. هاست می‌تواند یک کامپیوتر، سرور (Server)، روتر (Router)، گیتوی (Gateway) یا ... باشد. برای شناسایی هر هاست در شبکه یک IP منحصر به فرد به آن داده می‌شود.

- **سرویس (Service):** در تعریف شبکه، سرویس به یک نرم‌افزار یا پروتکل خاص اشاره دارد که بر روی یک هاست در شبکه اجرا می‌شود و به دیگر دستگاه‌های حاضر در شبکه خدماتی را ارائه می‌دهند.

- **پورت (Port):** در شبکه‌های کامپیوتری، پورت به یک عدد از ۰ تا ۶۵۵۳۵ اشاره دارد که برای تعیین و شناسایی خدمات و برنامه‌ها مورد استفاده قرار می‌گیرد. هر پورت متانظر با یک

خدمت یا برنامه خاص در یک هاست است و به آن امکان ارتباط و تبادل داده با سایر هاست‌های موجود در شبکه را می‌دهد.

- **پورت باز (Open Port):** اگر در یک هاست پورتی در وضعیت باز قرار داشته باشد یعنی آن دستگاه به درخواست‌های ورودی به این پورت پاسخ می‌دهد و ارتباط با آن دستگاه از طریق آن پورت امکان پذیر است.
- **پورت بسته (Close Port):** در نقطه مقابل پورت باز قرار دارد و اگر در دستگاهی پورتی در این حالت قرار داشته باشد به آن معناست که هاست موردنظر به درخواست‌های ورودی به این پورت پاسخ نخواهد داد و ارتباط با آن دستگاه از طریق پورت ذکرشده امکان پذیر نخواهد بود.

هدف پروژه

در این پروژه قصد داریم تا دانشجویان پس از آشنایی با تعدادی از قابلیت‌های نرم‌افزار Nmap، به پیاده‌سازی برخی از قابلیت‌های ساده این ابزار قدرتمند بپردازند.

پیشنهاد:

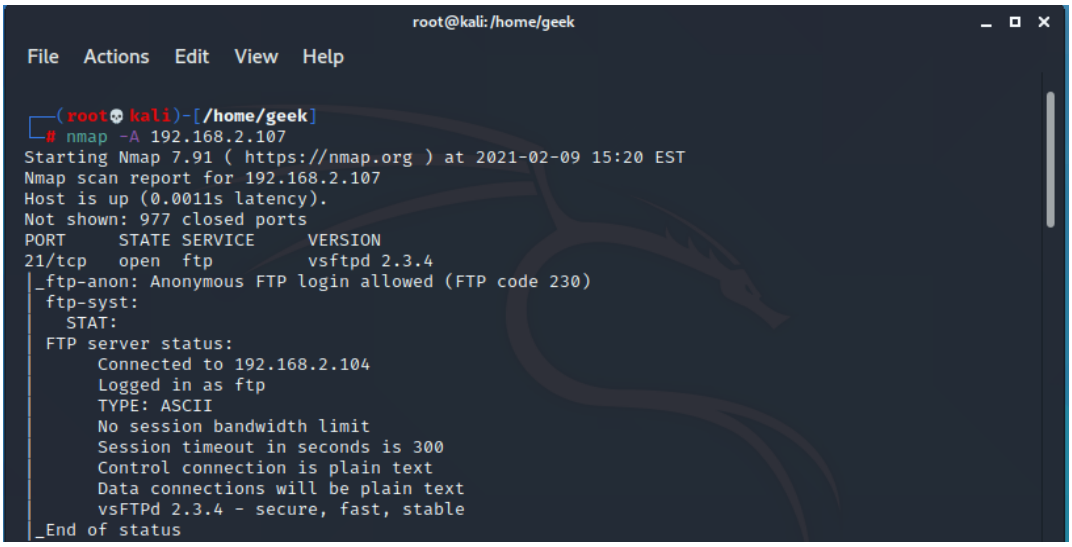
توصیه می‌شود برای آشنایی بیشتر با این نرم‌افزار، برنامه را دانلود کرده و پس از نصب، تعدادی قابلیت‌های ساده آن را امتحان کنید. همچنین برای مشاهده نحوه کار این ابزار می‌توانید از این لینک به صورت آنلاین، برخی از قابلیت‌های آن را امتحان کرده و نتیجه را مشاهده کنید.

در ادامه تصاویری از محیط ابزار و همچنین وبسایت معرفی شده قرار داده شده است.

```
pentester@TryHackMe$ sudo nmap -sV 10.10.76.34

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 05:03 BST
Nmap scan report for 10.10.76.34
Host is up (0.0040s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)
25/tcp  open  smtp    Postfix smtpd
80/tcp  open  http    nginx 1.6.2
110/tcp open  pop3    Dovecot pop3d
111/tcp open  rpcbind 2-4 (RPC #100000)
MAC Address: 02:A0:E7:B5:B6:C5 (Unknown)
Service Info: Host:  debra2.thm.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

برای مثال در تصویر بالا کاربر پس از دادن IP هدف خود به ابزار Nmap و استفاده از sV- نتایج اسکن را که شامل شماره پورت، وضعیت هر پورت، سرویسی که روی آن پورت در حال اجراست و همچنین ورژن آن سرویس را به عنوان گزارش دریافت کرده است.



یا در این تصویر کاربر با دادن آپشن A- به ابزار درخواست گزارش کاملی از اسکن هدف را دارد.

تصویر زیر نیز از وبسایت معرفی شده و اسکن دامنه Google.com گرفته شده است.

قابلیت‌های مدنظر جهت پیاده سازی

برنامه پیاده‌سازی شده توسط شما باید بتواند پس از دریافت آدرس IP هدف و یک بازه از پورت‌هایی که قصد بررسی آن‌ها را داریم عملیات‌های زیر را انجام دهد:

۱. بررسی وضعیت آنلاین بودن یا نبودن یک Host
۲. بررسی محدوده‌ای از پورت‌های یک Host و گزارش پورت‌هایی که در حالت Open قرار دارند و همچنین سرویس‌هایی که روی آن پورت‌ها در حالت اجرا قرار دارند.
۳. شبیه‌سازی متدهای GET و POST پروتکل HTTP

تمامی قابلیت‌های خواسته شده به وسیله Socket Programming قابل پیاده سازی هستند. در ادامه به بررسی هرکدام از موارد گفته شده می‌پردازیم.

۱. بررسی وضعیت آنلاین بودن یا نبودن یک Host

برای پیاده‌سازی این قابلیت برنامه باید تلاش کند یک ارتباط با هاست خواسته شده برقرار کند. در صورتی که این ارتباط با موفقیت برقرار شد می‌توان دریافت که هاست موردنظر آنلاین است و در غیر این صورت هاست آفلاین شناخته شده و نتیجه گزارش داده خواهد شد.

۲. بررسی پورت‌ها

برای پیاده‌سازی این قابلیت برنامه باید پس از دریافت IP یک هاست و یک رنج از پورت‌های مدنظر جهت اسکن شدن، تک تک پورت‌ها را مورد بررسی قرار دهد و در صورتی که پورت در وضعیت باز قرار داشت؛ شماره آن پورت و سرویسی که روی آن پورت درحال اجرا است را برگرداند.

در مثال زیر نمونه ای از ورودی و خروجی مدنظر برای قابلیت‌های شماره ۱ و ۲ را مشاهده می‌کنید.

```
PS C:\Users\     \Desktop> python nmap.py 1.1.1.1 80 81
1.1.1.1 is online
open port detected: 1.1.1.1     -- Port: 80     -- Service: http
```

۳. شبیه‌سازی متدهای GET و POST

POST و GET از متدهای درخواست پروتکل HTTP (HTTP Request Methods) هستند. GET برای فراخوانی داده مورد استفاده قرار می‌گیرد و متد پست برای ثبت کردن یک مقدار جدید. برای پیاده سازی این قابلیت، یک فایل server.py در اختیار شما قرار خواهد گرفت.

این فایل یک سرور را شبیه‌سازی می‌کند که اطلاعات تعدادی از کاربران را نگهداری می‌کند. این اطلاعات در تصویر زیر قابل مشاهده هستند.

```python
users = {
    'user1': {'name': 'Alice', 'age': 30},
    'user2': {'name': 'Bob', 'age': 25},
    'user3': {'name': 'Charlie', 'age': 35},
}
```

شما باید در برنامه پیاده‌سازی شده خودتان قابلیتی را به وجود بیاورید که ابزار بتواند با متد GET اطلاعات کاربر خواسته شده را که با ID آن کاربر (ستون اول که شامل مقادیر user1, user2, user3 می‌باشد ID کاربران را مشخص می‌کند) داده می‌شود پیدا کرده و مقادیر آن را گزارش دهد. فرمت قابل قبول برای برنامه سرور به شرح زیر است:

GET user_id

که شما با وارد کردن ID کاربر مدنظر می‌توانید اطلاعات آن را مشاهده کنید. به عنوان مثال به تصویر زیر دقت کنید.

```
Enter 'GET user_id' or 'POST user_name user_age' to simulate a request: GET user1
Response from the server:
HTTP/1.1 200 OK
Content-Type: application/json

{'name': 'Alice', 'age': 30}
Enter 'GET user_id' or 'POST user_name user_age' to simulate a request:
```

همچنین ابزار باید این قابلیت را داشته باشد که بتواند با استفاده از متد POST و دریافت نام و سن کاربر، اطلاعات آن کاربر را به مجموعه اطلاعات کاربرها اضافه کند. فرمت قابل قبول برای برنامه سرور به شرح زیر است:

POST user_name user_age

دستور POST پس از ساخت هر کاربر جدید یک ID منحصربه فرد برای او می‌سازد که به فرمت زیر است:

user + {شماره آخرین یوزر ساخته شده + ۱}

به عنوان مثال ID اولین یوزر ساخته شده برابر خواهد بود با user4.

نکته: لازم به ذکر است که در هر دو دستور مقادیر باید با کاراکتر space از هم جدا شده باشند.

به عنوان مثالی برای دستور POST به تصویر زیر دقت کنید:

```
Enter 'GET user_id' or 'POST user_name user_age' to simulate a request: POST Arthur 43
Response from the server:
HTTP/1.1 200 OK

User data updated
Enter 'GET user_id' or 'POST user_name user_age' to simulate a request: GET user4
Response from the server:
HTTP/1.1 200 OK
Content-Type: application/json

{'name': 'Arthur', 'age': 43}
Enter 'GET user_id' or 'POST user_name user_age' to simulate a request:
```

همانطور که مشاهده می‌کنید سرور پس از دریافت اطلاعات کاربر جدید آن اطلاعات را تحت ID جدید user4 ذخیره کرده که از طریق آن نیز قابل دسترسی و فراخوانی هستند.

پروژه دوم: پروتکل FTP

FTP یا File Transfer Protocol، یک پروتکل استاندارد برای انتقال فایل‌ها از یک سیستم به سیستم دیگر در شبکه‌های کامپیوتری است. FTP امکان انتقال فایل‌ها به صورت دوطرفه بین یک کلاینت (معمولاً یک کامپیوتر شخصی) و یک سرور (یک سیستم کامپیوتری که فایل‌ها را در اختیار دیگر کامپیوترها قرار می‌دهد) را فراهم می‌کند.

هدف پروژه

هدف اصلی این پروژه در درس شبکه، ایجاد یک سیستم انتقال فایل است که به کاربران امکان می‌دهد فایل‌ها را بین کلاینت و سرور با استفاده از پروتکل FTP انتقال دهند. دانشجویان در این پروژه با مفاهیم مهمی مانند ارتباط شبکه، برنامه‌نویسی سوکت، پروتکل انتقال فایل، امنیت شبکه، احراز هویت و مدیریت دسترسی آشنا می‌شوند و همچنین تجربه کار با ابزارهای شبکه و امنیت را به دست آورند.

شرح پروژه

در این پروژه، شما باید یک سیستم انتقال فایل طراحی کنید که به کاربران اجازه می‌دهد فایل‌ها را از کلاینت به سرور انتقال دهند. برای آشنایی کامل با این پروتکل، RFC 959 را مطالعه کنید. برخی از فرمان‌های کاربردی این پروتکل در ادامه توضیح داده می‌شوند.

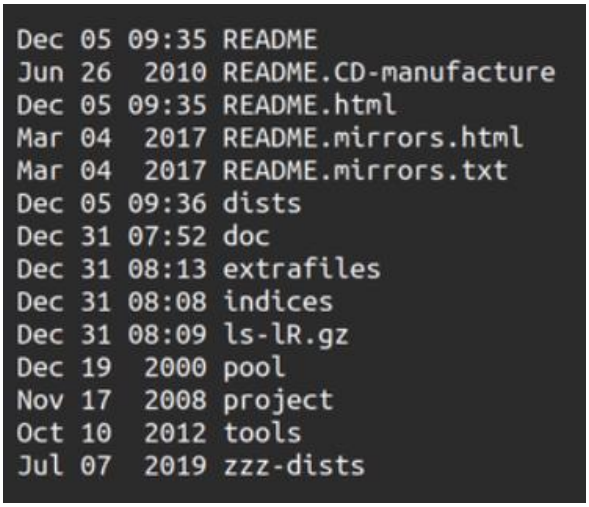
- USER: این فرمان، نام کاربری را برای اتصال به سرور ارسال می‌کند. برای مثال:

```
USER mahdi
```

- PASS: پس از وارد کردن نام کاربری، باید رمز عبور نیز ارسال شود. برای مثال:

```
PASS 1234
```

- LIST: با استفاده از این فرمان، فایل‌ها و دایرکتوری‌ها به کاربر نشان داده می‌شود. این اطلاعات به صورت یک فهرست به همراه اطلاعاتی مانند نام، اندازه، سطح دسترسی و تاریخ ایجاد است. در شکل ۱، نمونه‌ی آن نمایش داده شده است.



شکل ۱: اطلاعات فرمان LIST

همچنین کاربر می‌تواند یک پارامتر pathname مشخص کند. در این صورت، اطلاعات مسیر خواسته شده به آن نشان می‌دهد. اگر این مسیر یک دایرکتوری یا گروهی از فایل‌ها باشد، سرور باید فهرست این اطلاعات را به کاربر نمایش دهد. در صورتی که مسیر مشخص شده یک فایل باشد، باید اطلاعات داخل فایل به کاربر نمایش داده شود. به عنوان مثال:

LIST /path/directory

- **RETR: برای انتقال یک فایل از سرور به کلاینت، از این فرمان استفاده می‌شود. برای فراخوانی این فرمان نیاز است که مسیر فایل را نیز ارسال کنیم:**

RETR /path/file.txt

- **STOR: با استفاده از این فرمان، یک فایل را از کلاینت به سرور منتقل می‌کنیم. در صورتی که قبلا این فایل در سمت سرور وجود داشته باشد (نام یکسانی داشته باشند)، فایل جدید جایگزین فایل قبلی می‌شود.**

STOR /client-path /server-path

- **DELE: یک فایل را می‌توان با مشخص کردن مسیر آن در سمت سرور حذف نمود. زمانی که این فرمان اجرا می‌شود، از سمت سرور یک پیام (Do you really wish to delete? Y/N) برای تایید فرستاده می‌شود و در صورتی که کاربر Y را فشار دهد، فایل حذف خواهد شد.**
- **MKD: می‌توان یک دایرکتوری جدید در سمت سرور ایجاد کرد. مسیر می‌تواند به دو صورت absolute و relative مشخص شود.**

در صورتی که مسیر absolute باشد (نسبت به روت مشخص می‌شود):

MKD /home/user

زمانی که مسیر relative است (نسبت به مسیر فعلی مشخص می‌شود):

MKD ../folder

- RMD: می‌توان یک دایرکتوری را در سمت سرور حذف نمود. مطابق فرمان KMD، می‌توان مسیر را به دو صورت absolute و relative مشخص کرد.
- PWD: این فرمان مسیر فعلی در سمت سرور را نمایش می‌دهد. زمانی که PWD اجرا می‌شود، مسیر فعلی به کاربر (کلاینت) نمایش داده می‌شود:

/home/user/public

- CWD: می‌توان مسیر را در سمت سرور عوض کرد. این کار نیز به دو صورت absolute و relative انجام می‌شود. برای مثال:

CWD /home

- CDUP: مشابه دستور CWD، برای عوض کردن مسیر فعلی در سرور استفاده می‌شود. با این تفاوت که با اجرا شدن این دستور، به دایرکتوری والد منتقل می‌شویم. برای مثال، اگر دستور CDUP در مسیر /home/user/public اجرا شود، مسیر جدید /home/user خواهد بود.
- QUIT: این دستور باعث می‌شود که ارتباط کلاینت یا سرور قطع شود. در صورتی که یک جابه‌جایی فایل در حال انجام باشد، ابتدا عملیات کامل شده و سپس ارتباط بسته می‌شود.
- در صورتی که این عملیات با موفقیت انجام شود و دایرکتوری وجود داشته باشد، پیغام 200 directory changed to /home نمایش داده می‌شود. در غیر این صورت خطای 400 directory doesn’t exist نشان داده می‌شود. برای بقیه فرمان‌ها می‌توان از این پاسخ ایده گرفت. برای مثال، زمانی که نام کاربری و رمز عبور کاربر درست است، پیغام 200 (عملیات موفق)، در غیر این صورت 400 (خطا) به همراه یک متن مناسب نمایش داده شود.

این پروژه شامل قابلیت‌های زیر است:

- **سرور:** شما باید یک سرور FTP ایجاد کنید که فایل‌ها و دایرکتوری‌ها را نگهداری می‌کند و همچنین باید کلاینت‌ها به آن متصل شوند و فرمان‌های مشخص شده را اجرا کنند. زمانی که اتصال کلاینت برقرار می‌شود، یک لیست از دستورات مثل LIST، RETR و... به همراه نحوه استفاده به کلاینت فرستاده می‌شود.

- **کلاینت:** چند برنامه کلاینت که به کاربران امکان می‌دهند وارد سرور شوند، فایل‌ها را مشاهده کنند و همچنین فایل‌ها را دریافت کنند.
- **فرمان‌های موجود:** کلاینت باید بتواند فایل‌های خود را با فرمان STOR به سرور بفرستد، با فرمان RETR یک فایل را دریافت کند و همچین از دیگر فرمان‌های توضیح داده شده مانند LIST، DELE، MKD، RMD، PWD، CDUP و QUIT پشتیبانی کند.
- **مدیریت دسترسی‌ها:** دسترسی‌های کاربران باید مشخص شود. برای مثال، برخی از فایل‌ها و دایرکتوری‌ها به صورت خصوصی (private) در سمت سرور تعریف شده‌اند و همه‌ی کاربران دسترسی به این فایل‌ها و دایرکتوری‌ها را ندارند و فقط کاربران مشخص شده قابلیت خواندن و نوشتن را خواهند داشت.
- **مدیریت خطا:** برنامه‌ی شما باید خطا را به درستی مدیریت کند و پیغام مناسب به کاربر نشان دهد. برای مثال، زمانی که فرمان CWD /home اجرا می‌شود، در صورتی که این عملیات با موفقیت انجام شود و دایرکتوری وجود داشته باشد، پیغام 200 directory changed to /home نمایش داده می‌شود. در غیر این صورت خطای 400 directory doesn’t exist نشان داده می‌شود. برای دیگر فرمان‌ها نیز می‌توان از این پاسخ ایده گرفت. برای مثال، زمانی که نام کاربری و رمز عبور درست است، پیغام 200 (عملیات موفق)، در غیر این صورت 400 (خطا) به همراه یک متن مناسب نمایش داده شود.
- **گزارش‌گیری:** تمامی عملیات‌ها و فعالیت‌ها در سمت کلاینت و سرور برای افزایش امنیت باید در سمت سرور ثبت شوند و کلاینت ها با دستور REPORT بتوانند آن را مشاهده کنند. در واقع این فرمان تاریخچه فرمان‌های اجرا شده در سمت سرور را به همه نمایش می‌دهد. (می‌توان صرفا یک کاربر مشخص به عنوان ادمین به این اطلاعات دسترسی داشته باشد).
- **امنیت:** در مورد روش‌های تقویت امنیت اتصال FTP مانند FTPS و SFTP مطالعه کنید (اجباری) و در صورت امکان برخی از آن‌ها را در برنامه خود پیاده‌سازی کنید (امتیازی).

پروژه سوم: ابزار WireShark

WireShark یک برنامه نرم افزاری آنالیز پروتکل شبکه منبع باز است که توسط Gerald Combs در سال ۱۹۹۸ معرفی شده است. یک سازمان جهانی متشکل از متخصصان شبکه و توسعه دهندگان نرم افزار، از WireShark پشتیبانی میکنند.متخصصان شبکه و امنیت و حتی هکر ها برای بررسی پکت هایی که در شبکه در حال انتقال است از این نرم افزار قدرتمند استفاده میکنند.

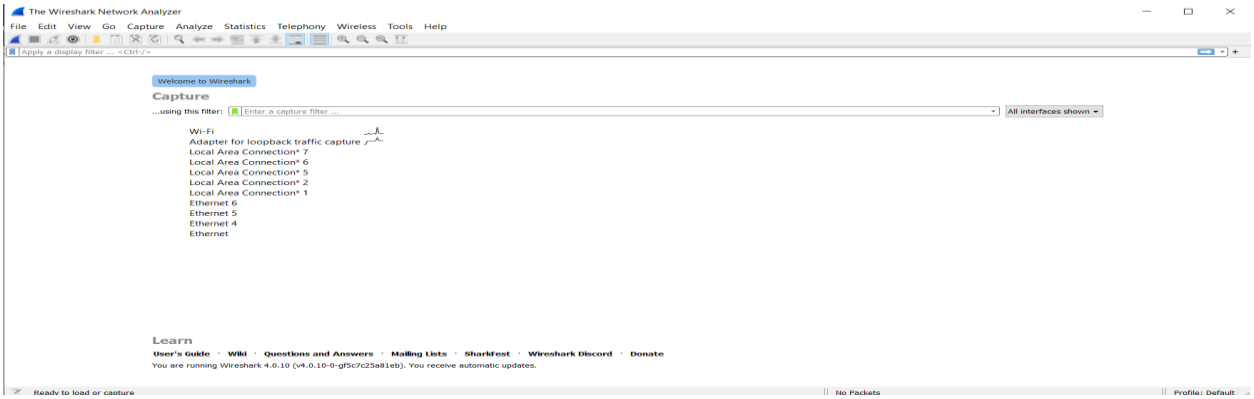
برای اطلاعات بیشتر میتوانید این فیلم کوتاه را مشاهده کنید.

هدف پروژه

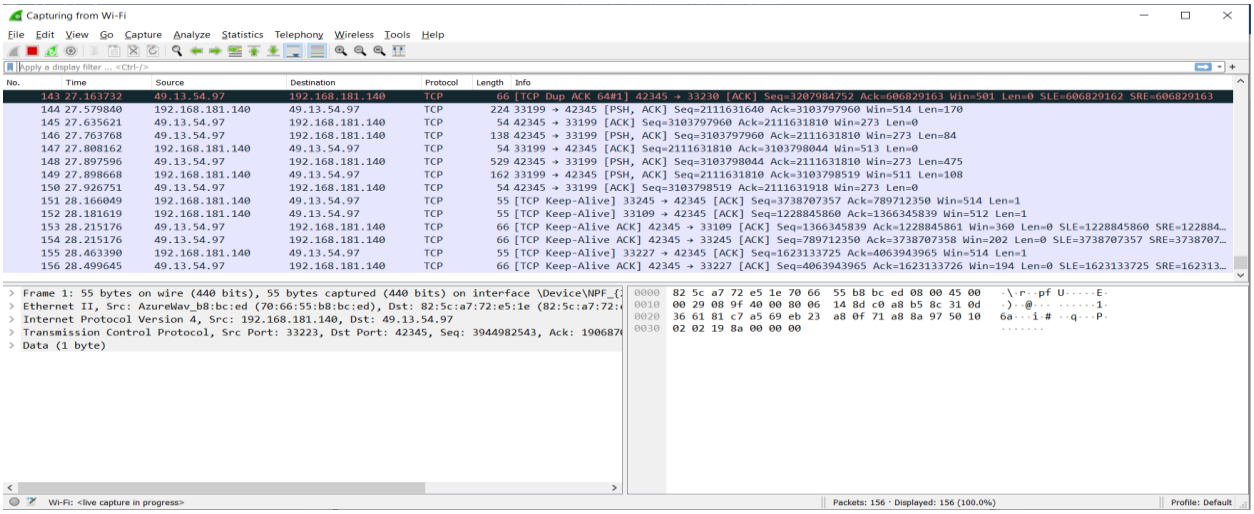
در این پروژه قصد داریم تا دانشجویان پس از آشنایی با تعدادی از قابلیت‌های نرم‌افزار WireShark، بتوانند با این نرم افزار قدرتمند کار کنند.

راهنمای کار با نرم افزار

در ادامه راهنمای کار با WireShark را به صورت تصویری یاد خواهیم داد. پس از نصب و اجرای WireShark با تصویری به صورت زیر رو به رو خواهیم شد.

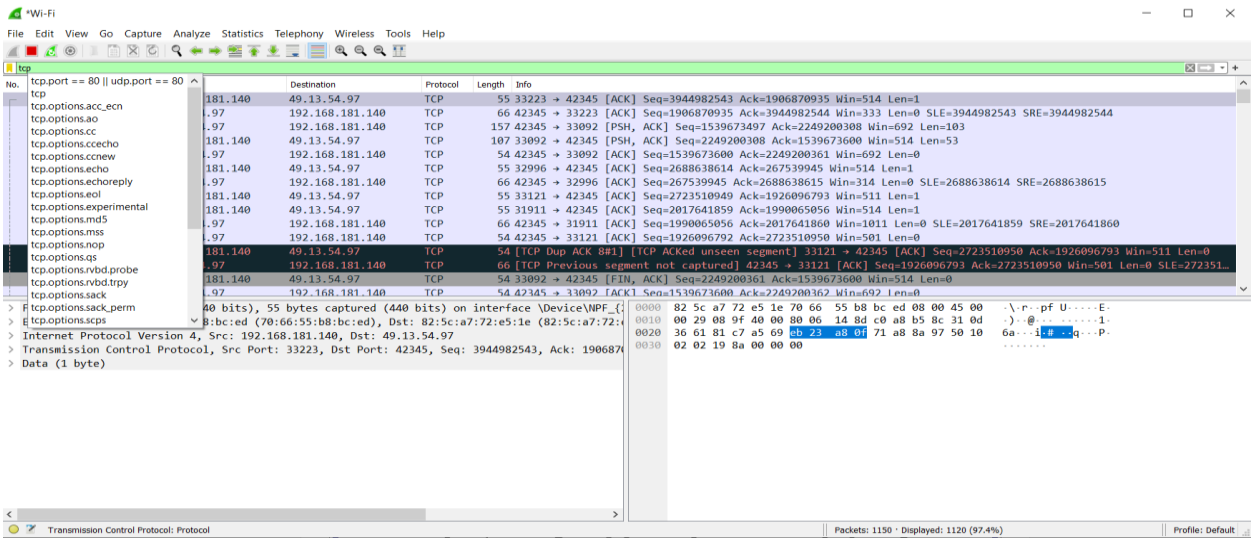


در صورت استفاده از کارت شبکه wireless برروی گزینه WI-FI کلیک کنید تا WireShark شروع به ضبط پکت ها کند. پس از کلیک برروی WI-FI با همچین صفحه ای مواجه خواهیم شد.



در این قسمت میتوانید برروی پکت های مختلف کلیک کرده و محتوای آنها را تماشا کنید.در مثال زیر تلاش میکنیم با راهنمای مرحله به مرحله یک پکت TCP را دریافت کرده و محتوای header آن را تماشا کنید.

ابتدا در قسمت فیلتر پروتکل TCP را وارد میکنیم تا تمام پروتکل های TCP دریافت شده را نمایش بدهد.



سپس به قسمت نمایش محتوای پکت ها میرویم و Transmission Control Protocol را انتخاب میکنیم.

```
Transmission Control Protocol, Src Port: 33223, Dst Port: 42345, Seq: 3944982543, Ack: 1906
    Source Port: 33223
    Destination Port: 42345
    [Stream index: 0]
    [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 1]
    Sequence Number: 3944982543
    [Next Sequence Number: 3944982544]
    Acknowledgment Number: 1906870935
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window: 514
    [Calculated window size: 514]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x198a [unverified]
```

همانطور که مشاهده میکنید توانستیم به Header این پکت دسترسی پیدا کنیم.

بخش اول: راهنمای گرفتن یک Response پروتکل HTTP

ابتدا WireShark را باز کرده سپس پکت های منتقل شده را ضبط کنید پس از آن این لینک را در پنجره مرورگر خود باز کنید. سپس ضبط پکت هارا متوقف کنید.سپس به سوالات زیر پاسخ دهید.

سوالات:

۱. **مرورگر شما از کدام ورژن HTTP استفاده میکند ؟ کدام یک از ورژن HTTP برای اجرای سرور است ؟**
۲. **مرورگر شما چه زبان هایی را به عنوان زبان های قابل قبول سرور نشان میدهد ؟**
۳. **IP سیستم شما چیست ؟ IP سیستم مقصد چیست ؟**

۴. کد وضعیت برگشتی از سمت مقصد به مرورگر شما چیست ؟
۵. چند بایت محتوا به مرورگر شما برگردانده میشود ؟
۶. با بررسی داده های خام در پنجره محتوای بسته، آیا سرصفحه هایی در داخل داده ها مشاهده می کنید که در پنجره فهرست بسته نمایش داده نمی شوند؟ اگر چنین است، یکی را نام ببرید.

بخش دوم: بازیابی (Retrieve) اسناد طولانی

در مثال قبل داکیومنت بازیابی شده ساده و فایل کم حجمی بوده است. بیایید ببینیم وقتی یک فایل طولانی HTML را دانلود می کنیم چه اتفاقی می افتد. موارد زیر را انجام بدهید.

برنامه وایرشارک را باز کرده سپس پکت ها را ضبط کنید، پس از آن این لینک را در مرورگر خود وارد کنید، ضبط پکت هارا متوقف کنید سپس پکت هارا برای پروتکل HTTP فیلتر کنید.

مشاهدات خود را توضیح دهید و بگویید تفاوت این مسئله با مسئله قبلی چگونه است ؟

سوالات:

۱. چند پیام ارسالی(request) توسط مرورگر شما ارسال شده است ؟
۲. کد وضعیت و عبارت (phrase) در پیام دریافتی(response) چیست ؟
۳. چند بخش TCP حاوی داده برای حمل پاسخ HTTP واحد فایل HTML دانلود شده نیاز است ؟

بخش سوم: احراز هویت HTTP

در نهایت، بیایید از وب سایتی بازدید کنیم که با رمز عبور محافظت می شود و دنباله پیام HTTP رد و بدل شده برای چنین سایتی را بررسی کنیم. این لینک با رمز عبور محافظت میشود. نام کاربری برابر با dm557 است و رمز عبور برابر با network است. بنابراین بیایید به این سایت "ایمن" محافظت شده با رمز عبور دسترسی پیدا کنیم.مراحل زیر را انجام دهید.

ابتدا وایرشارک را باز کنید و پکت هارا ضبط کنید، سپس این لینک را در مرورگر خود وارد کنید و نام کاربری و رمز عبور داده شده را وارد کنید.پس از انجام مراحل قبل ضبط پکت هارا قطع کنید.

سوالات:

۱. پاسخ سرور (کد وضعیت و عبارت) در پاسخ به پیام اولیه HTTP GET از مرورگر شما چیست؟
۲. وقتی مرورگر شما برای بار دوم پیام HTTP GET را ارسال می کند، چه فیلد جدیدی در پیام HTTP GET گنجانده شده است؟

نکته:

نام کاربری (dm557) و رمز عبور (شبکه) که وارد کردید در رشته کاراکترها (ZG01NTc6bmV0d29yaw==) به دنبال هدر "Authorization: Basic" در پیام HTTP GET مشتری کدگذاری می شوند. در حالی که ممکن است به نظر برسد که نام کاربری و رمز عبور شما رمزگذاری شده است، آنها به سادگی در قالبی به نام Base64 کدگذاری می شوند. نام کاربری و رمز عبور رمزگذاری نشده است! برای مشاهده این موضوع، به این سایت بروید و رشته رمزگذاری شده با ZG01NTc6bmV0d29yaw== base64 را وارد کنید و رمزگشایی کنید. شما از کدگذاری Base64 به رمزگذاری ASCII ترجمه کرده اید، و بنابراین باید نام کاربری و رمز عبور خود را ببینید.