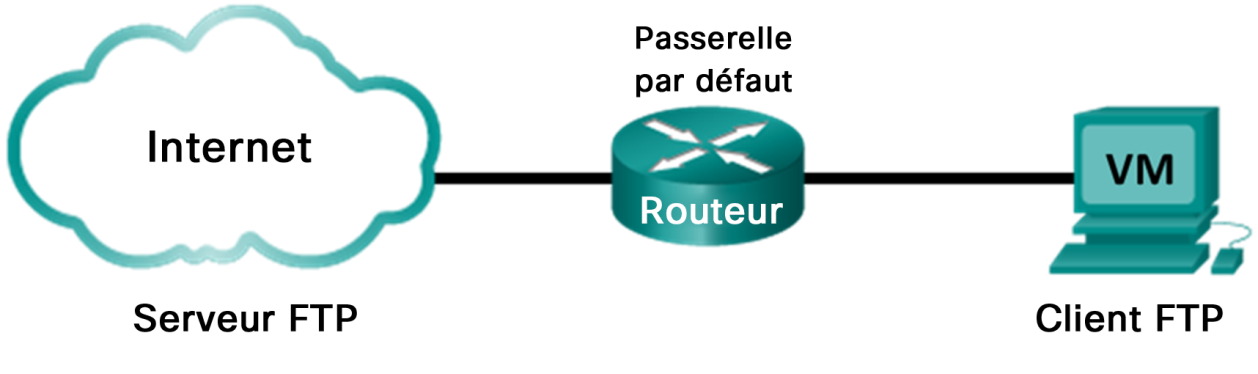
Travaux pratiques - Utilisation de Wireshark pour l'examen de captures TCP et UDP

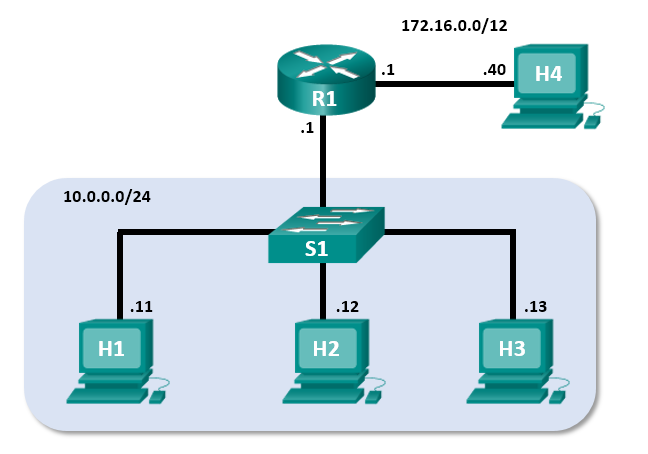
1. Topologie – Partie 1 (FTP)



La première partie mettra l'accent sur une capture TCP d'une session FTP. Cette topologie est composée du poste de travail virtuel CyberOps disposant d'un accès à Internet.

1. Topologie Mininet – Partie 2 (TFTP)

La partie 2 met en évidence une capture UDP d'une session TFTP à l'aide des hôtes dans Mininet.



1. Objectifs

Partie 1 : Identifier les champs d'en-tête TCP ainsi que les opérations TCP à l'aide de la capture de session FTP de Wireshark

Partie 2 : Identifier les champs d'en-tête UDP ainsi que les opérations UDP à l'aide de la capture de session TFTP de Wireshark

1. Contexte/scénario

Les deux protocoles de la couche transport TCP/IP sont le protocole TCP, défini dans le document RFC 761, et le protocole UDP, défini dans le document RFC 768. Les deux protocoles prennent en charge les communications du protocole de couche supérieure. Par exemple, TCP prend en charge la couche transport pour les protocoles HTTP (HyperText Transfer Protocol) et FTP, entre autres. UDP fournit notamment la prise en charge de la couche transport pour le système de noms de domaine (DNS) et TFTP.

Dans la première partie de ces travaux pratiques, vous utiliserez l'outil libre (« open source ») de Wireshark pour capturer et analyser les champs d'en-tête de protocole TCP pour les transferts de fichiers FTP entre l'ordinateur hôte et un serveur FTP anonyme. La ligne de commande du terminal permet de se connecter à un serveur FTP anonyme et de télécharger un fichier. Dans la deuxième partie de ces travaux pratiques, vous utiliserez Wireshark pour capturer et analyser des champs d'en-tête UDP pour les transferts de fichiers TFTP entre deux ordinateurs hôtes Mininet.

1. Ressources requises

* Poste de travail virtuel CyberOps
* Accès Internet

1. Identifier les champs d'en-tête TCP ainsi que les opérations TCP à l'aide de la capture de session FTP de Wireshark

Dans la première partie, vous utiliserez Wireshark pour capturer une session FTP et examiner les champs d'en-tête TCP.

* 1. Démarrez une capture Wireshark.
     1. Démarrez et connectez-vous au poste de travail virtuel CyberOps. Ouvrez une fenêtre du terminal et démarrez Wireshark. Saisissez le mot de passe **cyberops** et cliquez sur **OK** lorsque vous y êtes invité.

[analyst@secOps ~]$ **sudo wireshark-gtk**

* + 1. Démarrez une capture Wireshark pour l'interface **enp0s3**.
    2. Ouvrez une autre fenêtre de terminal pour accéder à un site ftp externe. Saisissez **ftp ftp.cdc.gov** à l'invite. Connectez-vous au site FTP du Centre pour le contrôle et la prévention des maladies (Center for Disease Control and Prevention, CDC) avec l'utilisateur **anonymous** et aucun mot de passe.

[analyst@secOps ~]$ **ftp ftp.cdc.gov**

Connected to ftp.cdc.gov.

220 Microsoft FTP Service

Name (ftp.cdc.gov:analyst): **anonymous**

331 Anonymous access allowed, send identity (e-mail name) as password.

Mot de passe :

230 User logged in.

Remote system type is Windows\_NT.

ftp>

* 1. Téléchargez le fichier Readme (Lisez-moi).
     1. Localisez et téléchargez le fichier Readme (Lisez-moi) en exécutant la commande **ls** afin d'afficher la liste des fichiers.

ftp> **ls**

200 PORT command successful.

125 Data connection already open; Transfer starting.

-rwxrwxrwx 1 owner group 128 May 9 1995 .change.dir

-rwxrwxrwx 1 owner group 107 May 9 1995 .message

drwxrwxrwx 1 owner group 0 Feb 2 11:21 pub

-rwxrwxrwx 1 owner group 1428 May 13 1999 Readme

-rwxrwxrwx 1 owner group 383 May 13 1999 Siteinfo

-rwxrwxrwx 1 owner group 0 May 17 2005 up.htm

Drwxrwxrwx 1 owner group 0 May 20 2010 w3c

-rwxrwxrwx 1 owner group 202 Sep 22 1998 welcome.msg

226 Transfer complete.

**Remarque**: vous pouvez recevoir le message :

421 Service not available, remote server has closed connection

ftp: No control connection for command

Si tel est le cas, le serveur FTP est en panne. Toutefois, vous pouvez poursuivre les travaux pratiques en analysant les paquets que vous avez réussi à capturer et en lisant ceux que vous n'êtes pas parvenu à capturer. Vous pouvez également reprendre les travaux pratiques pour voir si le serveur FTP est de nouveau opérationnel.

* + 1. Exécutez la commande **get Readme** pour télécharger le fichier. Lorsque le téléchargement est terminé, saisissez la commande **quit** pour quitter.

ftp> **get Readme**

200 PORT command successful.

125 Data connection already open; Transfer starting.

ATTENTION !  36 bare linefeeds received in ASCII mode

File may not have transferred correctly.

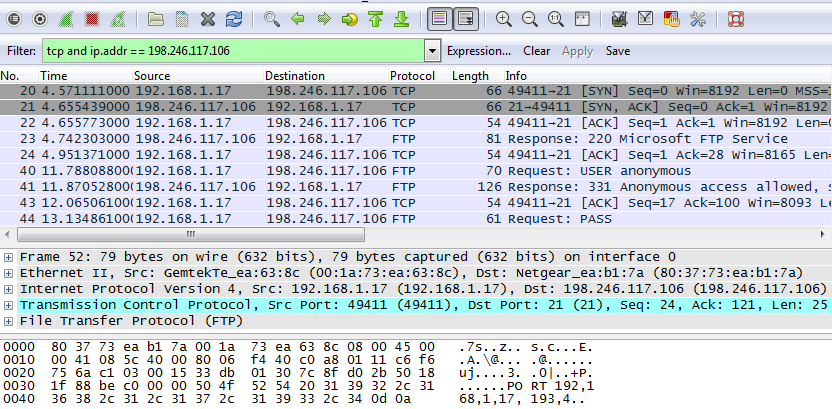
226 Transfer complete.

1428 bytes received in 0.056 seconds (24.9 kbytes/s)

* + 1. Une fois le transfert terminé, saisissez la commande **quit** pour quitter ftp.
  1. Arrêtez la capture Wireshark.
  2. Affichez la fenêtre principale de Wireshark.

Wireshark a capturé de nombreux paquets pendant la session FTP sur ftp.cdc.gov. Pour limiter la quantité de données à analyser, appliquez le filtre **tcp and ip.addr == 198.246.117.106** et cliquez sur **Apply**.

**Remarque** : l'adresse IP 198.246.117.106 correspond à l'adresse de [ftp.cdc.gov](ftp://ftp.cdc.gov) lors de la création de ces travaux pratiques. L'adresse IP peut être différente pour vous. Si c'est le cas, recherchez le premier paquet TCP qui a commencé la connexion en 3 étapes avec [ftp.cdc.gov](ftp://ftp.cdc.gov). L'adresse IP de destination est l'adresse IP que vous devez utiliser pour votre filtre.



**Remarque** : votre interface Wireshark peut être différente de l'image ci-dessus.

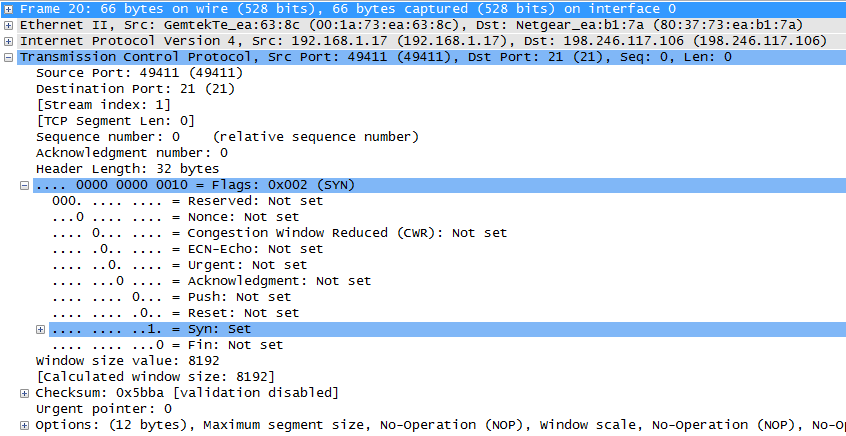
* 1. Analysez les champs TCP.

Une fois le filtre TCP appliqué, les trois premiers paquets (partie supérieure) affichent la séquence [SYN], [SYN, ACK] et [ACK] qui correspond à la connexion TCP en trois étapes.

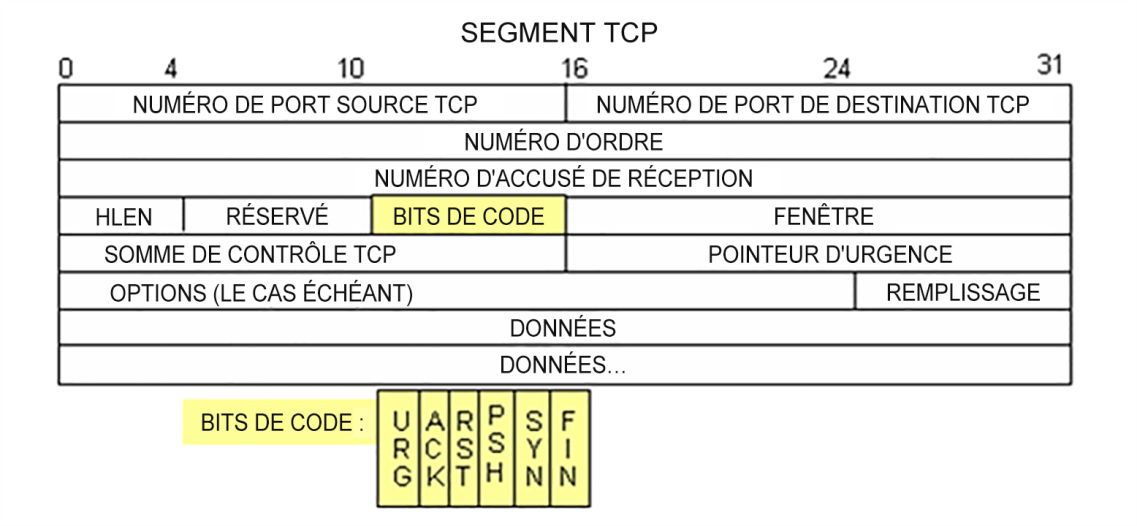
Screen shot of the three packets displaying the TCP three-way handshake

TCP est couramment utilisé au cours d'une session pour contrôler la transmission et l'arrivée des datagrammes ainsi que pour gérer la taille des fenêtres. Pour chaque échange de données entre le client FTP et le serveur FTP, une nouvelle session TCP est démarrée. Au terme du transfert de données, la session TCP est fermée. Une fois la session FTP terminée, TCP exécute un arrêt, puis une déconnexion.

Dans Wireshark, des informations TCP détaillées sont disponibles dans le volet de détails des paquets (section centrale). Sélectionnez le premier datagramme TCP à partir de l'ordinateur hôte et développez le datagramme TCP, comme illustré ci-dessous.



Le datagramme TCP développé ressemble au volet de détails des paquets indiqué ci-dessous.



L'illustration ci-dessus est un schéma de datagramme TCP. Une explication de chaque champ est fournie pour référence :

* Le **numéro de port source TCP** (TCP source port number) appartient à l'hôte de session TCP qui a ouvert une connexion. Il s'agit généralement d'une valeur aléatoire supérieure à 1 023.
* Le **numéro de port de destination TCP** (TCP destination port number) permet d'identifier le protocole de couche supérieure ou l'application sur le site distant. Les valeurs comprises entre 0 et 1 023 représentent les « ports réservés » et sont associées aux services et applications standard (comme décrit dans le document RFC 1700, par exemple Telnet, FTP et HTTP). La combinaison de l'adresse IP source, du port source, de l'adresse IP de destination et du port de destination identifie de façon unique la session à la fois vis-à-vis de l'émetteur et du récepteur.

**Remarque** : dans la capture Wireshark ci-dessus, le port de destination est le port 21, ce qui correspond au FTP. Les serveurs FTP écoutent sur le port 21 pour les connexions clientes FTP.

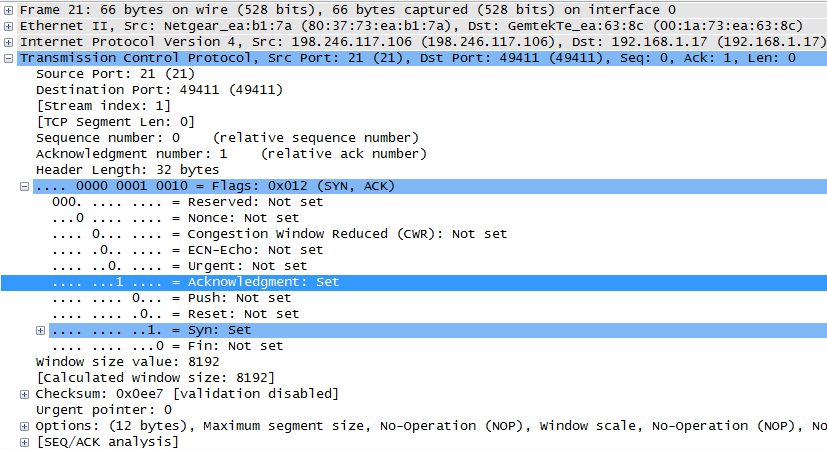
* Le **numéro d'ordre** (Sequence number) indique le numéro du dernier octet dans un segment.
* Le **numéro d'accusé de réception** (Acknowledgment number) indique l'octet suivant prévu par le récepteur.
* Les **bits de code** ont une signification spécifique dans la gestion des sessions et dans le traitement des segments. Valeurs intéressantes :
  1. **ACK** : (Acknowledgment ) accusé de réception d'un segment.
  2. **SYN** : (Synchronize) uniquement défini lorsqu'une nouvelle session TCP est négociée au cours de la connexion en trois étapes.
  3. **FIN** : (Finish) requête pour fermer la session TCP.
* La **taille de la fenêtre** (Window size) est la valeur de la fenêtre glissante. Elle détermine le nombre d'octets pouvant être envoyés avant d'attendre l'accusé de réception.
* Le **pointeur d'urgence** (Urgent pointer) n'est utilisé qu'avec un indicateur URG (Urgent), lorsque l'émetteur doit envoyer des données urgentes au récepteur.
* Les **options** ne contiennent actuellement qu'une seule valeur, définie comme étant la taille maximale d'un segment TCP (valeur facultative).

À l'aide de la capture Wireshark du démarrage de la première session TCP (bit SYNC défini sur 1), renseignez les informations concernant l'en-tête TCP. Il est possible que certains champs ne concernent pas ce paquet.

De la machine virtuelle au serveur CDC (seul le bit SYN est défini sur 1) :

|  |  |
| --- | --- |
| Adresse IP source |  |
| Adresse IP de destination |  |
| Numéro du port source |  |
| Numéro du port de destination |  |
| Numéro d'ordre |  |
| Numéro d'accusé de réception |  |
| Longueur de l'en-tête |  |
| Taille de fenêtre |  |

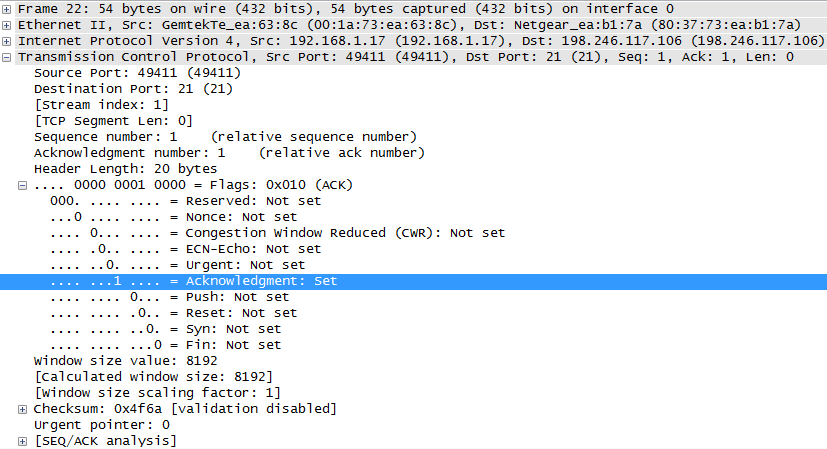
Dans la deuxième capture Wireshark filtrée, le serveur FTP CDC reconnaît la requête de la machine virtuelle. Notez les valeurs des bits SYN et ACK.



Indiquez les informations suivantes concernant le message SYN-ACK.

|  |  |
| --- | --- |
| Adresse IP source |  |
| Adresse IP de destination |  |
| Numéro du port source |  |
| Numéro du port de destination |  |
| Numéro d'ordre |  |
| Numéro d'accusé de réception |  |
| Longueur de l'en-tête |  |
| Taille de fenêtre |  |

Lors de l'étape finale de la négociation visant à établir une communication, la machine virtuelle envoie un accusé de réception au serveur. Notez que seul le bit ACK est défini sur 1 et que le numéro d'ordre a été incrémenté à 1.



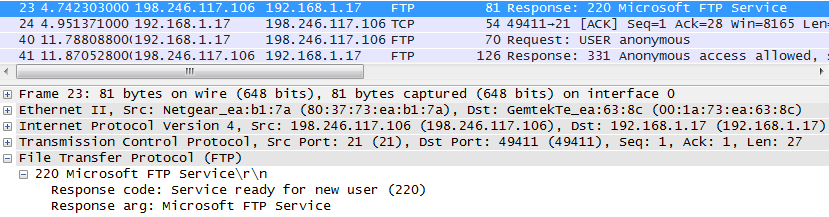
Indiquez les informations suivantes concernant le message ACK.

|  |  |
| --- | --- |
| Adresse IP source |  |
| Adresse IP de destination |  |
| Numéro du port source |  |
| Numéro du port de destination |  |
| Numéro d'ordre |  |
| Numéro d'accusé de réception |  |
| Longueur de l'en-tête |  |
| Taille de fenêtre |  |

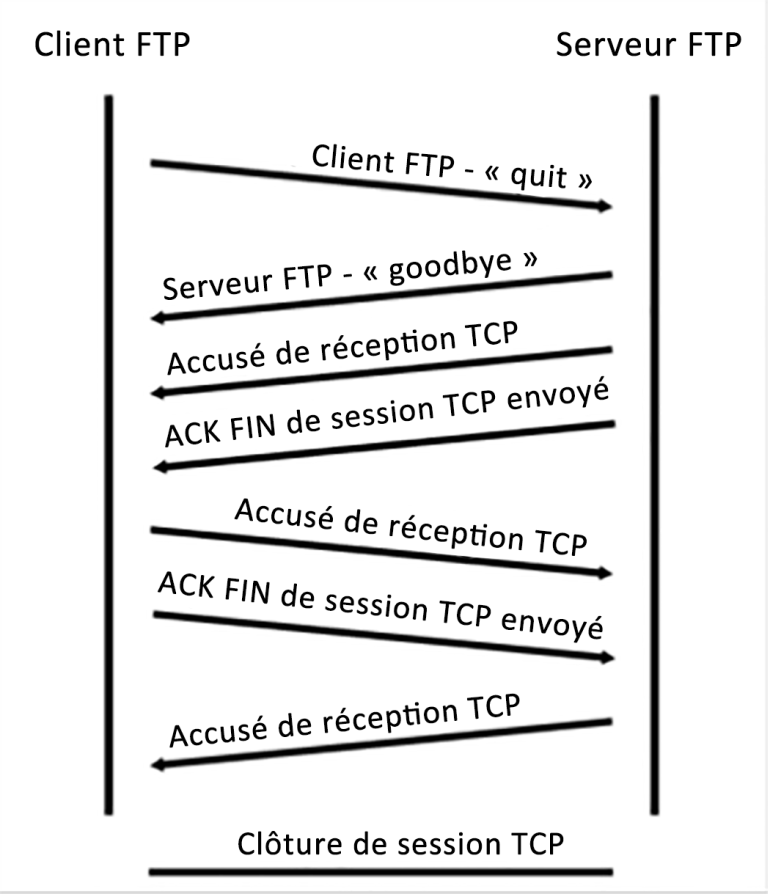
Combien d'autres datagrammes TCP contenaient un bit SYN ?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

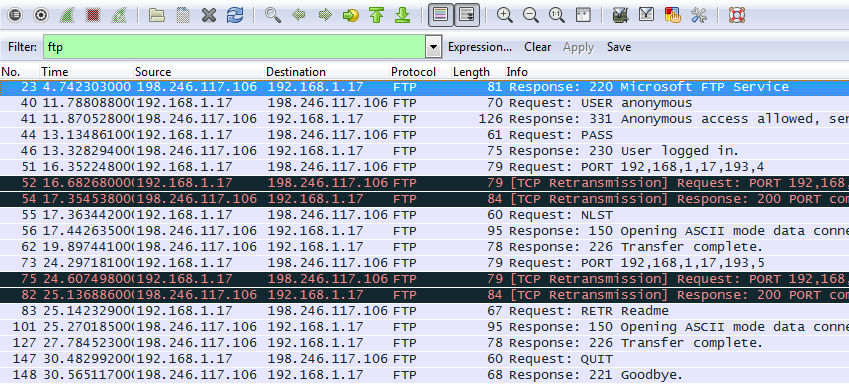
Une fois qu'une session TCP est établie, le trafic FTP peut circuler entre le PC et le serveur FTP. Le client FTP et le serveur communiquent entre eux sans tenir compte du contrôle et de la gestion de la session par TCP. Lorsque le serveur FTP envoie *Response: 220* au client FTP, la session TCP sur le client FTP envoie un accusé de réception à la session TCP sur le serveur. Cette séquence est visible dans la capture Wireshark ci-dessous.



Une fois la session FTP terminée, le client FTP envoie une commande pour « quitter ». Le serveur FTP accuse réception de la fin de la session FTP avec un message *Response: 221 Goodbye*. À ce stade, la session TCP du serveur FTP envoie un datagramme TCP au client FTP, et annonce ainsi la fin de la session TCP. La session TCP du client FTP accuse réception du datagramme de fin, puis envoie la fin de sa propre session TCP. Lorsque l'émetteur de la fin de la session TCP, le serveur FTP, reçoit une fin en double, un datagramme ACK est envoyé pour accuser réception de la fin et la session TCP est fermée. Cette séquence est visible dans le schéma et la capture ci-dessous.



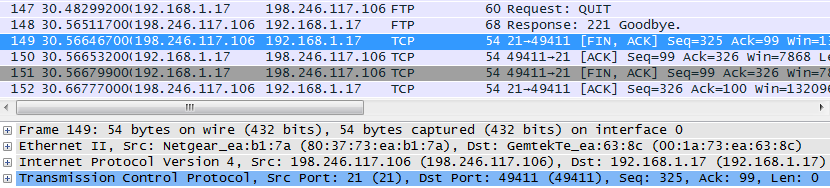
En appliquant un filtre **ftp**, la séquence entière du trafic FTP peut être examinée dans Wireshark. Notez la séquence des événements au cours de cette session FTP. Le nom d'utilisateur **anonymous** a été utilisé pour récupérer le fichier Readme (Lisez-moi). Une fois le fichier transféré, l'utilisateur a mis fin à la session FTP.



Appliquez le filtre TCP à nouveau dans Wireshark pour examiner la fin de la session TCP. Quatre paquets sont transmis à la fin de la session TCP. Comme la connexion TCP est en mode duplex intégral, chaque sens doit se terminer indépendamment. Examinez les adresses source et de destination.

Dans cet exemple, le serveur FTP n'a plus de données à envoyer dans le flux. Il envoie un segment dont l'indicateur FIN est défini dans la trame 149. Le PC envoie un paquet ACK pour accuser réception du paquet FIN mettant fin à la session du serveur vers le client dans la trame 150.

Dans la trame 151, le PC envoie un paquet FIN au serveur FTP pour mettre fin à la session TCP. Le serveur FTP répond par un paquet ACK pour accuser réception du paquet FIN envoyé par le PC dans la trame 152. À présent, la session TCP est interrompue entre le serveur FTP et le PC.



1. Identifier les champs d'en-tête UDP ainsi que les opérations UDP à l'aide de la capture de session TFTP de Wireshark

Dans la deuxième partie, vous utiliserez Wireshark pour capturer une session TFTP et examiner les champs d'en-tête UDP.

* 1. Démarrer Mininet et le service tftpd.
     1. Démarrez Mininet. Saisissez **cyberops** comme mot de passe lorsque vous y êtes invité.

[analyst@secOps ~]$ **sudo lab.support.files/scripts/cyberops\_topo.py**

[sudo] password for analyst:

* + 1. Saisissez H1 et H2 à l'invite de commande **mininet >**.

\*\*\* À partir de l'interface de ligne de commande :

mininet> **xterm H1 H2**

* + 1. Dans la fenêtre du terminal **H1**, démarrez le serveur tftpd en utilisant le script fourni.

[root@secOps analyst]# **/home/analyst/lab.support.files/scripts/start\_tftpd.sh**

[root@secOps analyst]#

* 1. Créez un fichier de transfert tftp.
     1. Créez un fichier texte à l'invite du terminal **H1** dans le dossier /srv/tftp/.

[root@secOps analyst]# **echo "Ce fichier contient mes données tftp." > /srv/tftp/my\_tftp\_data**

* + 1. Vérifiez que le fichier a été créé avec les données souhaitées dans le dossier.

[root@secOps analyst]# **cat /srv/tftp/my\_tftp\_data**

Ce fichier contient mes données tftp.

* + 1. En raison des mesures de sécurité mises en place pour ce serveur tftp, le nom du fichier récepteur doit déjà exister. Sur **H2**, créez un fichier nommé **my\_tftp\_data**.

[root@secOps analyst]# **touch my\_tftp\_data**

* 1. Capturer une session TFTP dans Wireshark
     1. Lancez Wireshark dans **H1**.

[root@secOps analyst]# **wireshark-gtk &**

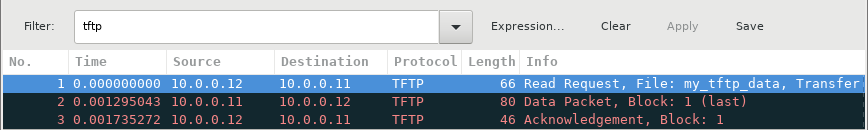
* + 1. À partir du menu **Edit**, choisissez **Preferences** et cliquez sur la flèche pour développer **Protocols**. Faites défiler vers le bas, puis sélectionnez **UDP**. Activez la case à cocher **Validate the UDP checksum if possible** et cliquez sur **Apply**. Cliquez ensuite sur **OK**.



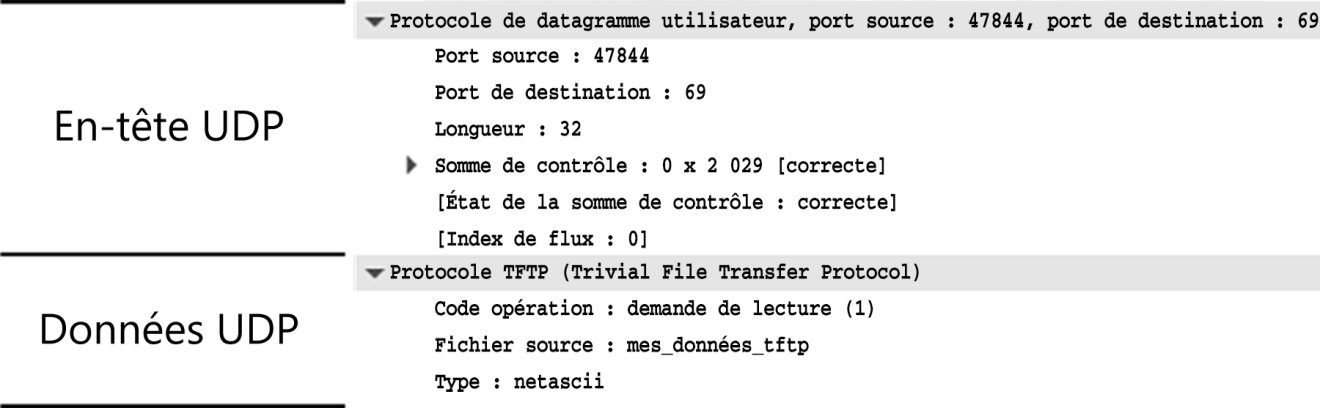
* + 1. Démarrez une capture Wireshark sur l'interface **H1-eth0**.
    2. Démarrez une session tftp de **H2** au serveur tftp sur **H1** et obtenez le fichier **my\_tftp\_data**.

[root@secOps analyst]# **tftp 10.0.0.11 -c get my\_tftp\_data**

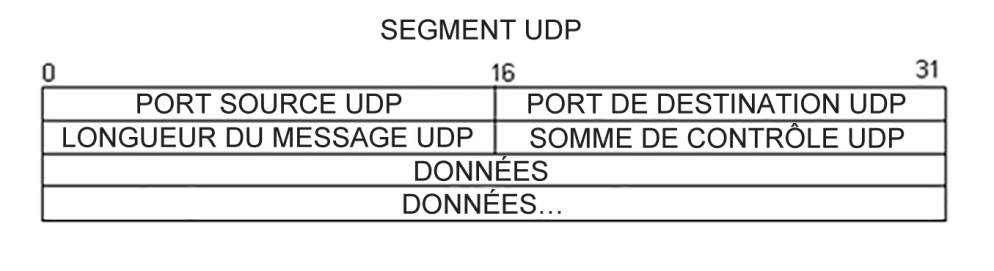
* + 1. Arrêtez la capture Wireshark. Définissez le filtre sur **tftp** et cliquez sur **Apply**. Utilisez les trois paquets TFTP pour remplir le tableau et répondre aux questions dans le reste de ces travaux pratiques.



Des informations UDP détaillées sont disponibles dans le volet de détails des paquets Wireshark. Sélectionnez le premier datagramme UDP à partir de l'ordinateur hôte, et déplacez le pointeur de la souris vers le volet de détails des paquets. Il peut s'avérer nécessaire de modifier le volet de détails des paquets et de développer l'enregistrement UDP en cliquant sur la zone de développement du protocole. Le datagramme UDP développé doit être semblable au schéma ci-dessous.



La figure ci-dessous représente un schéma de datagramme UDP. Les informations d'en-tête sont peu nombreuses par rapport au datagramme TCP. De même que pour le protocole TCP, chaque datagramme UDP est identifié par les ports source et de destination UDP.



À l'aide de la capture Wireshark du premier datagramme UDP, renseignez les informations concernant l'en-tête UDP. La somme de contrôle est une valeur hexadécimale (base 16), identifiée par le code 0x précédent :

|  |  |
| --- | --- |
| Adresse IP source |  |
| Adresse IP de destination |  |
| Numéro du port source |  |
| Numéro du port de destination |  |
| Longueur du message UDP |  |
| Somme de contrôle UDP |  |

De quelle manière UDP vérifie-t-il l'intégrité du datagramme ?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examinez la première trame renvoyée par le serveur tftpd. Complétez les informations sur l'en-tête UDP :

|  |  |
| --- | --- |
| Adresse IP source |  |
| Adresse IP de destination |  |
| Numéro du port source |  |
| Numéro du port de destination |  |
| Longueur du message UDP |  |
| Somme de contrôle UDP |  |

Remarque : le datagramme UDP de retour possède un port de source UDP différent. Toutefois, ce dernier sert au transfert TFTP restant. Étant donné que la connexion n'est pas fiable, seul le port source d'origine utilisé pour commencer la session TFTP sert à gérer le transfert TFTP.

Notez également que la somme de contrôle UDP est incorrecte. Ceci provient très probablement du déchargement de la somme de contrôle UDP. Pour plus d'informations sur la raison de cet événement, effectuez une recherche sur « UDP checksum offload » (Déchargement de la somme de contrôle).

* 1. Nettoyage

Au cours de cette étape, vous allez arrêter et nettoyer Mininet.

* + 1. Dans le terminal qui a lancé Mininet, saisissez **quit** à l'invite de commande.

mininet> **quit**

* + 1. À l'invite, saisissez **sudo mn – c** pour nettoyer les processus démarrés par Mininet.

[analyst@secOps ~]$ **sudo mn -c**

1. Remarques générales

Ces travaux pratiques ont permis aux participants d'analyser les opérations des protocoles TCP et UDP à partir de sessions FTP et TFTP capturées. En quoi le protocole TCP gère-t-il la communication différemment du protocole UDP ?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_