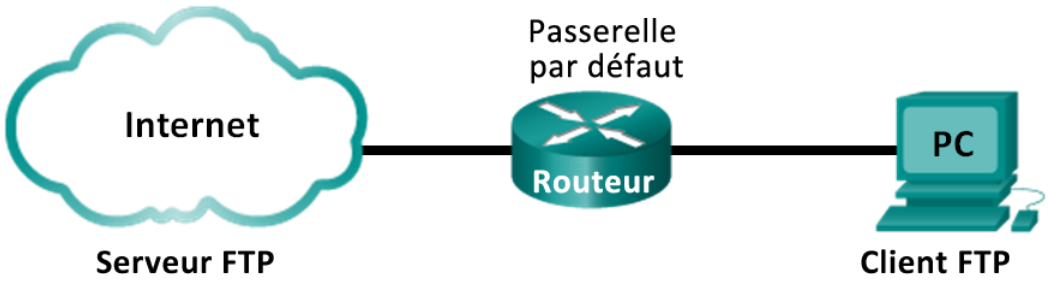
Travaux pratiques – Utilisation de Wireshark pour examiner les captures FTP et TFTP

Topologie – Première partie (FTP)

La première partie mettra l'accent sur une capture TCP d'une session FTP. Cette topologie est composée d'un PC disposant d'un accès à Internet.



Topologie – Deuxième partie (TFTP)

La deuxième partie mettra l'accent sur une capture UDP d'une session TFTP. Le PC doit disposer à la fois d'une connexion Ethernet et d'une connexion console avec le commutateur S1.

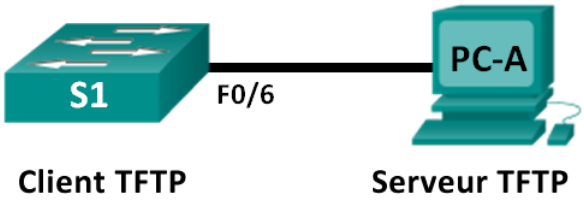


Table d'adressage (deuxième partie)

Périphérique	Interface	Adresse IP	Masque de sous-réseau	Passerelle par défaut
S1	VLAN 1	192.168.1.1	255.255.255.0	NA
PC-A	Carte réseau	192.168.1.3	255.255.255.0	192.168.1.1

Objectifs

1re partie : Identifier les champs d'en-tête TCP et les opérations TCP à l'aide de la capture de session FTP de Wireshark

2e partie : Identifier les champs d'en-tête UDP et les opérations UDP à l'aide de la capture de session TFTP de Wireshark

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 permet d'offrir la prise en charge de 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.

Remarque : la capacité à comprendre les éléments des en-têtes TCP et UDP ainsi que les opérations représentent une compétence cruciale pour les ingénieurs réseau.

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. L'utilitaire de ligne de commande Windows 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 de protocole UDP pour les transferts de fichiers TFTP entre l'hôte et le commutateur S1.

Remarque : le commutateur utilisé est un Cisco Catalyst 2960s équipé de Cisco IOS version 15.0(2) (image lanbasek9). D'autres commutateurs et versions de Cisco IOS peuvent être utilisés. Selon le modèle et la version de Cisco IOS, les commandes disponibles et le résultat produit peuvent varier de ceux figurant dans les travaux pratiques.

Remarque : vérifiez que la mémoire du commutateur a été effacée et qu'aucune configuration initiale n'est présente. En cas de doute, contactez votre instructeur.

Remarque : la première partie suppose que le PC dispose d'un accès à Internet et ne peut pas être effectuée avec Netlab. La deuxième partie est compatible avec Netlab.

Ressources requises – première partie (FTP)

1 PC (Windows 7, Vista ou XP, équipé d'un accès à Internet, d'un accès aux invites de commande et de Wireshark)

Ressources requises – deuxième partie (TFTP)

- 1 commutateur (Cisco 2960 équipé de Cisco IOS version 15.0(2) image lanbasek9 ou similaire)
- 1 PC (Windows 7, Vista ou XP, avec Wireshark et un serveur TFTP, tel que tftpd32, installé)
- Un câble de console permettant de configurer les périphériques Cisco IOS via le port de console
- Un câble Ethernet tel qu'indiqué dans la topologie

1re partie : Identifier les champs d'en-tête TCP et 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.

Étape 1 : Démarrez une capture Wireshark.

a. Interrompez tout le trafic réseau superflu, par exemple fermez le navigateur Web, pour limiter le volume du trafic lors de la capture Wireshark.

b. Lancez la capture Wireshark.

Étape 2 : Téléchargez le fichier Lisezmoi (Readme).

a. À partir de l'invite de commandes, saisissez **ftp ftp.cdc.gov**.

b. 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.

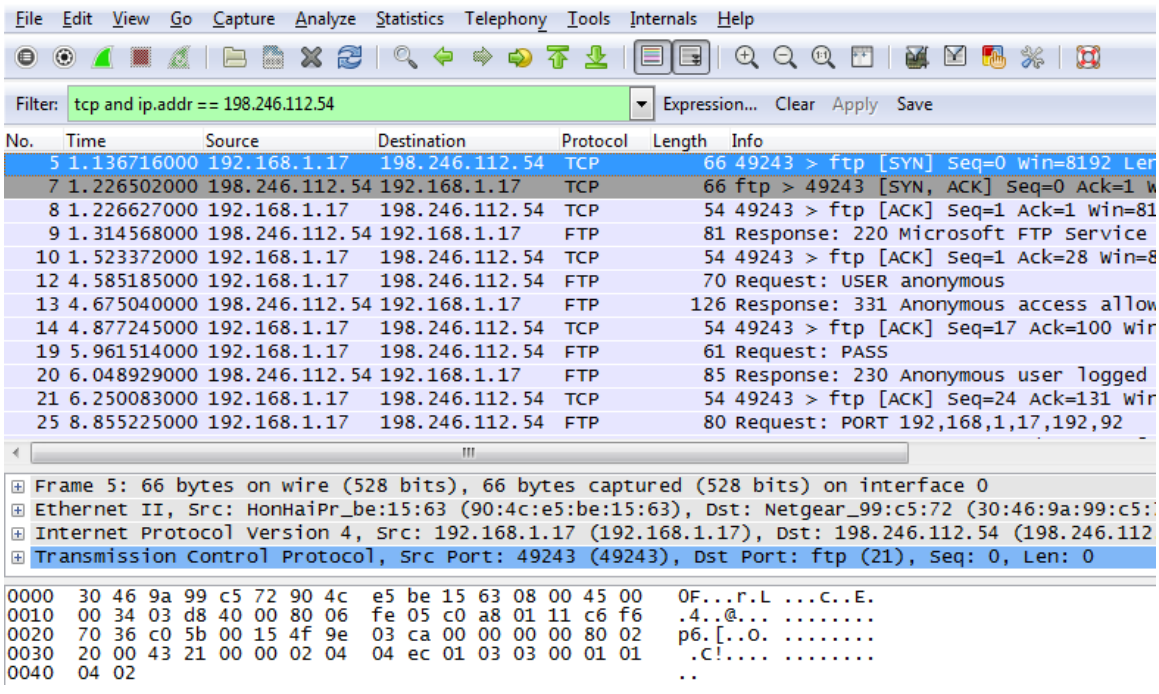
c. Recherchez et téléchargez le fichier Readme.

```
C:\Users\user1>ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
User (ftp.cdc.gov:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 Anonymous user logged in.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
aspnet_client
pub
Readme
Siteinfo
up.htm
w3c
web.config
welcome.msg
226 Transfer complete.
ftp: 76 bytes received in 0.00Seconds 19.00Kbytes/sec.
ftp> get Readme
200 PORT command successful.
150 Opening ASCII mode data connection for Readme(1428 bytes).
226 Transfer complete.
ftp: 1428 bytes received in 0.01Seconds 204.00Kbytes/sec.
ftp> quit
221
```

Étape 3 : Arrêtez la capture Wireshark.

Étape 4 : 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, tapez **tcp and ip.addr == 198.246.112.54** dans la zone **Filter: entry (Filtre : saisie)** et cliquez sur **Apply (Appliquer)**. L'adresse IP, 198.246.112.54, est l'adresse du serveur ftp.cdc.gov.



Étape 5 : Analyse des champs TCP.

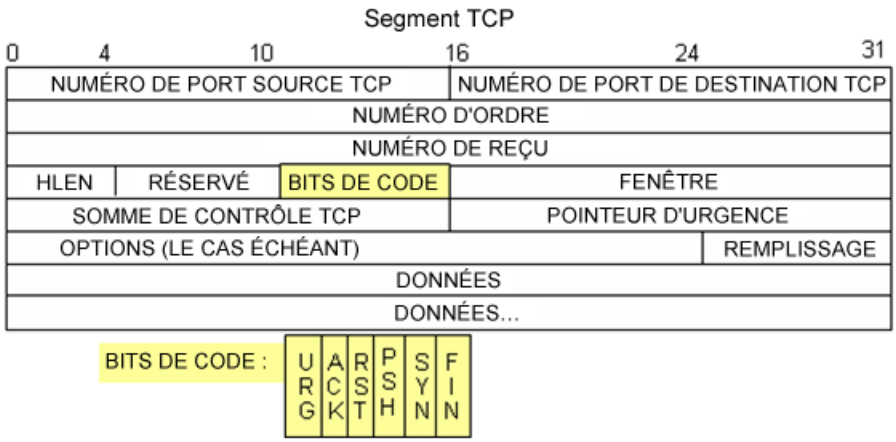
Une fois que le filtre TCP a été appliqué, les trois premières trames dans le volet de la liste des paquets (section supérieure) affiche le protocole TCP de la couche transport, créant ainsi une session fiable. La séquence de [SYN], [SYN, ACK] et [ACK] illustre l'échange en trois étapes.

```
    5 1.136716000  192.168.1.17    198.246.112.54   TCP        66 49243 > ftp [SYN] Seq=0 Win=8192 Le
    7 1.226502000  198.246.112.54  192.168.1.17     TCP        66 ftp > 49243 [SYN, ACK] Seq=0 Ack=1
    8 1.226627000  192.168.1.17    198.246.112.54   TCP        54 49243 > ftp [ACK] Seq=1 Ack=1 Win=8
```

TCP est couramment utilisé au coursd'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. Ainsi, une fois la session FTP terminée, TCP exécute un arrêt et une déconnexion normalement.

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

```
⊞ Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_be:15:63 (90:4c:e5:be:15:63), Dst: Netgear_99:c5:72 (30:46:9a:99:c5:72)
⊞ Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.112.54 (198.246.112.54)
⊟ Transmission Control Protocol, Src Port: 49243 (49243), Dst Port: ftp (21), Seq: 0, Len: 0
    Source port: 49243 (49243)
    Destination port: ftp (21)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Header length: 32 bytes
  ⊟ Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ⊞ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
    Window size value: 8192
    [Calculated window size: 8192]
  ⊞ Checksum: 0x4321 [validation disabled]
  ⊞ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), Nc
```



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

- **TCP source port number (Numéro du port source TCP)** appartient à l'hôte de session TCP qui a ouvert une connexion. Il s'agit généralement d'une valeur aléatoire supérieure à 1 023.
- **TCP destination port number (Numéro du port de destination TCP)** 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 aux applications standard (comme décrit dans le document RFC 1700, tels que Telnet, FTP, HTTP, etc.). 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-dessous, le port de destination est le 21, ce qui correspond au FTP. Les serveurs FTP écoutent sur le port 21 pour les connexions clientes FTP.

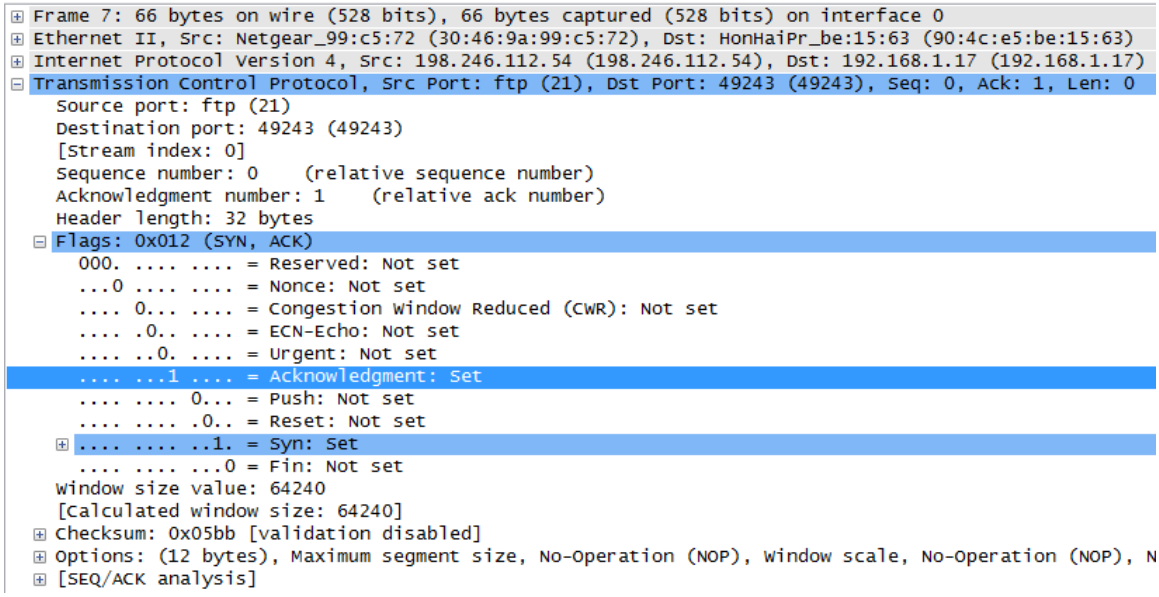
- **Sequence number (Numéro d'ordre)** indique le numéro du dernier octet dans un segment.
- **Acknowledgment number (Numéro de reçu)** indique l'octet suivant attendu par le récepteur.
- **Code bits (Bits de code)** ont une signification spécifique dans la gestion des sessions et dans le traitement des segments. Valeurs intéressantes :
  - ACK :reçu d'un segment
  - SYN : Synchronize, uniquement défini lorsqu'une nouvelle session TCP est négociée au cours de la connexion en trois étapes.
  - FIN : Finish, requête de fermeture de la session TCP.
- **Window size (Taille de la fenêtre)** est la valeur de la fenêtre glissante qui détermine le nombre d'octets pouvant être envoyés avant d'attendre le reçu.
- **Urgent pointer (Pointeur d'urgence)** n'est utilisé qu'avec un indicateur URG (Urgent) : lorsque l'émetteur doit envoyer des données urgentes au récepteur.
- **Options** ne contient actuellement qu'une seule option, et elle est définie comme la taille maximale d'un segment TCP (valeur facultative).

À l'aide de la capture Wireshark du démarrage de la première session TCP (bit SYN défini sur 1), renseignez les informations concernant l'en-tête TCP :

Du PC 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 de reçu :	
Longueur de l'en-tête :	
Taille de fenêtre :	

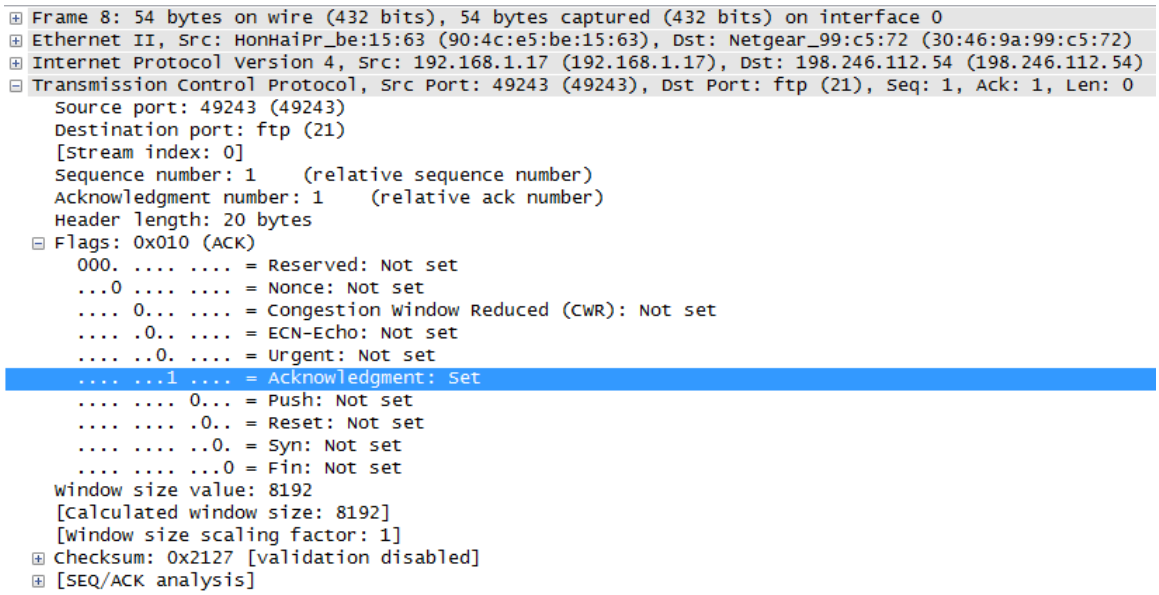
Dans la deuxième capture Wireshark filtrée, le serveur FTP CDC reconnaît la requête de l'ordinateur. Notez les valeurs des bits SYN et ACK.



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 de reçu :	
Longueur de l'en-tête :	
Taille de fenêtre :	

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

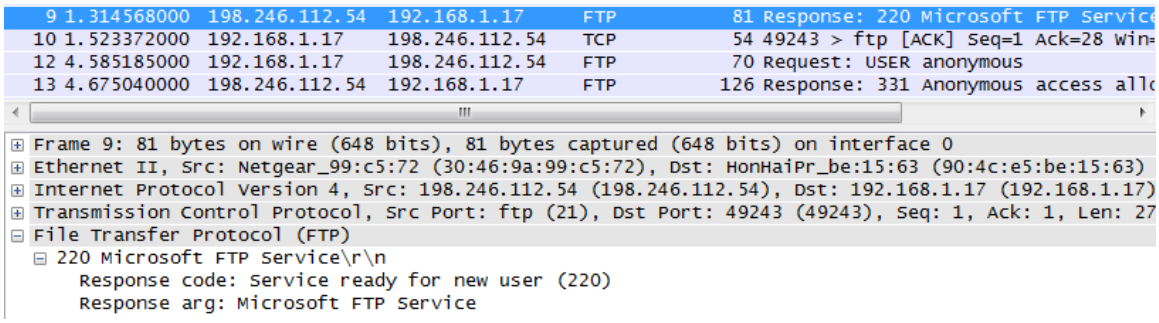


Indiquez les informations suivantes concernant le message ACK.

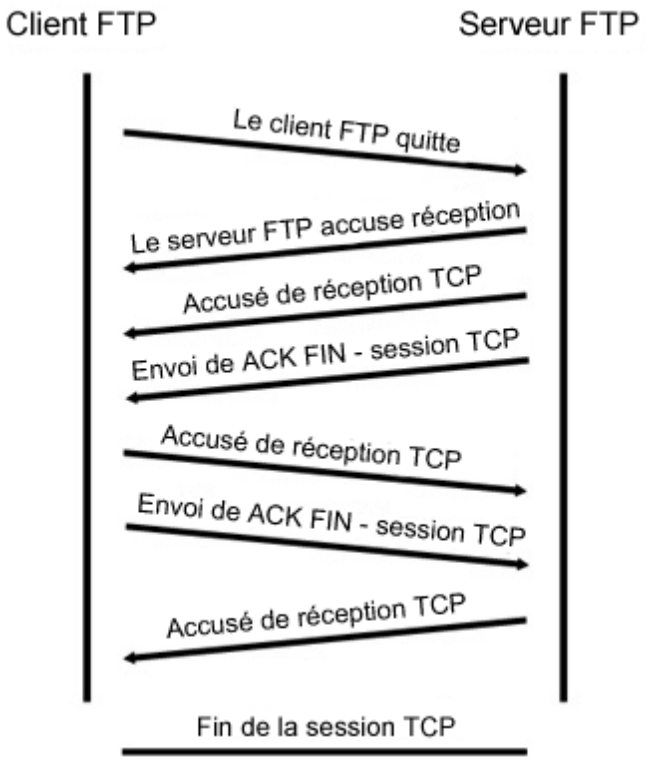
Adresse IP source :	
Adresse IP de destination :	
Numéro du port source :	
Numéro du port de destination :	
Numéro d'ordre :	
Numéro de reçu :	
Longueur de l'en-tête :	
Taille de fenêtre :	

Combiend'autresdatagrammes TCP contenaient un bit SYN ?

Une fois qu'une session TCP est établie, le trafic FTP peut survenir entre le PC et le serveur FTP. Le client FTP et le serveur communiquent entre eux en ignorant que le protocole TCP contrôle et gère la session. 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 le 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 anonyme a été utilisé pour récupérer le fichier Readme. Une fois le fichier transféré, l'utilisateur a mis fin à la session FTP.

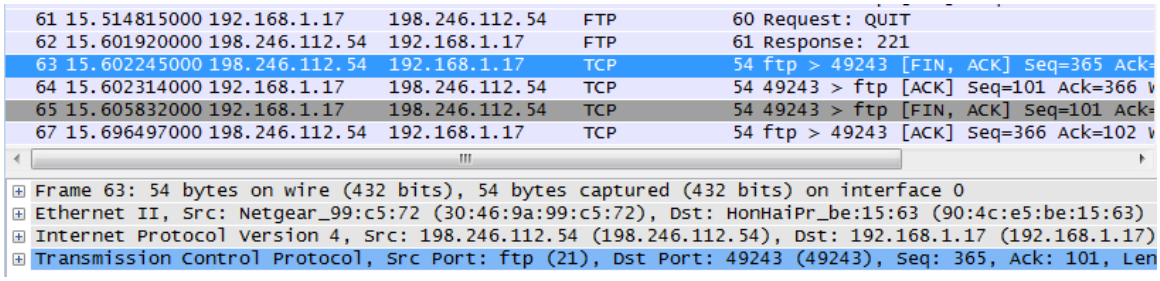
Filter: ftp — Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
9	1.314568000	198.246.112.54	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
12	4.585185000	192.168.1.17	198.246.112.54	FTP	70	Request: USER anonymous
13	4.675040000	198.246.112.54	192.168.1.17	FTP	126	Response: 331 Anonymous access allowe
19	5.961514000	192.168.1.17	198.246.112.54	FTP	61	Request: PASS
20	6.048929000	198.246.112.54	192.168.1.17	FTP	85	Response: 230 Anonymous user logged i
25	8.855225000	192.168.1.17	198.246.112.54	FTP	80	Request: PORT 192,168,1,17,192,92
26	8.945530000	198.246.112.54	192.168.1.17	FTP	84	Response: 200 PORT command successful
27	8.955549000	192.168.1.17	198.246.112.54	FTP	60	Request: NLST
29	9.053034000	198.246.112.54	192.168.1.17	FTP	109	Response: 150 Opening ASCII mode data
39	9.347432000	198.246.112.54	192.168.1.17	FTP	78	Response: 226 Transfer complete.
42	12.621720000	192.168.1.17	198.246.112.54	FTP	80	Request: PORT 192,168,1,17,192,93
43	12.709658000	198.246.112.54	192.168.1.17	FTP	84	Response: 200 PORT command successful
44	12.722592000	192.168.1.17	198.246.112.54	FTP	67	Request: RETR Readme
45	12.811097000	198.246.112.54	192.168.1.17	FTP	118	Response: 150 Opening ASCII mode data
58	13.107294000	198.246.112.54	192.168.1.17	FTP	78	Response: 226 Transfer complete.
61	15.514815000	192.168.1.17	198.246.112.54	FTP	60	Request: QUIT
62	15.601920000	198.246.112.54	192.168.1.17	FTP	61	Response: 221

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 bidirectionnelle, 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 avec le positionnement d'indicateur FIN dans la trame 63. 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 64.

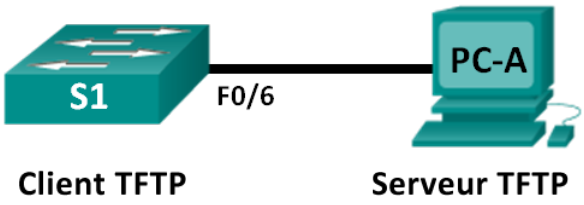
Dans la trame 65, le PC envoie un paquet FIN au serveur FTP pour mettre fin à la session TCP. Le serveur FTP répond par un paquet ACK pour accuser réception du paquet FIN envoyé par le PC dans la trame 67. À présent, la session TCP est interrompue entre le serveur FTP et le PC.



2e partie : Identifier les champs d'en-tête UDP et les opérations UDP à l'aide de la capture de session TFTP de Wireshark

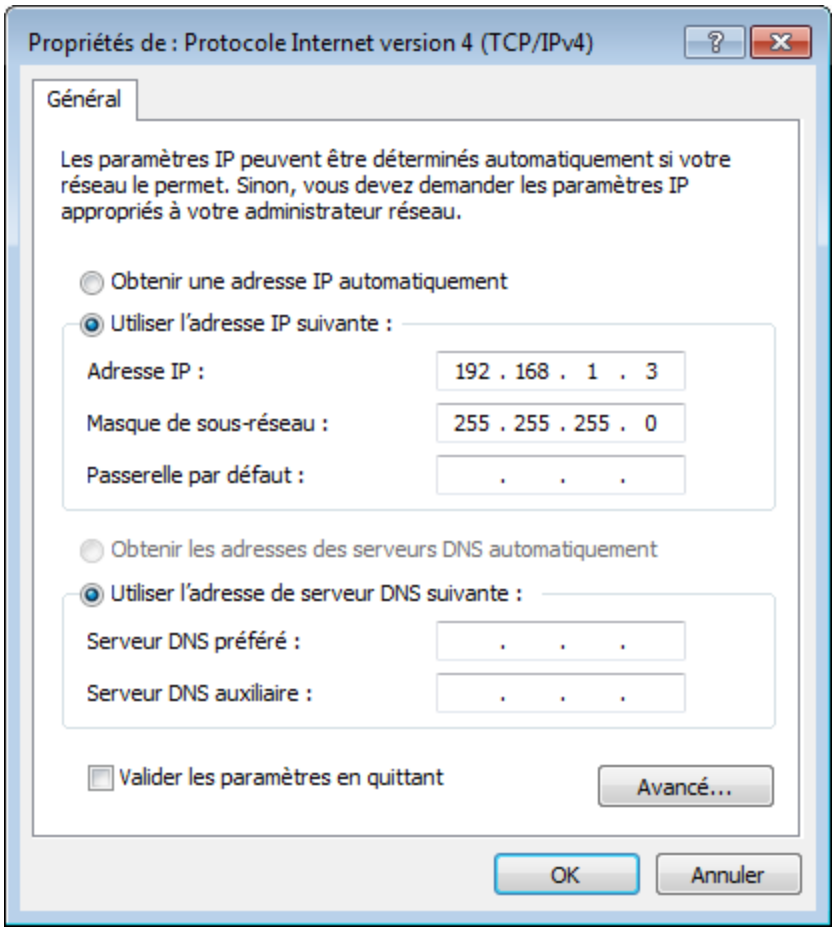
Dans la deuxième partie, vous utiliserez Wireshark pour capturer une session TFTP et examiner les champs d'en-tête UDP.

Étape 1 : Installez cette topologie physique et préparez la capture TFTP.



a. Établissez une connexion console et une connexion Ethernet entre PC-A et le commutateur S1.

b. Si ce n'est déjà fait, configurez manuellement l'adresse IP du PC sur 192.168.1.3. Il n'est pas obligatoire de définir la passerelle par défaut.



c. Configurez le commutateur. Attribuez l'adresse IP 192.168.1.1 à VLAN 1. Vérifiez la connectivité avec le PC en envoyant une requête ping à 192.168.1.3. Le cas échéant, procédez à un dépannage.

```
Switch>enable
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#host S1
S1(config)#interface vlan 1
S1(config-if)#ip address 192.168.1.1 255.255.255.0
S1(config-if)#no shut
*Mar  1 00:37:50.166: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Mar  1 00:37:50.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1,
changed state to up
S1(config-if)# end
S1# ping 192.168.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1007 ms
```

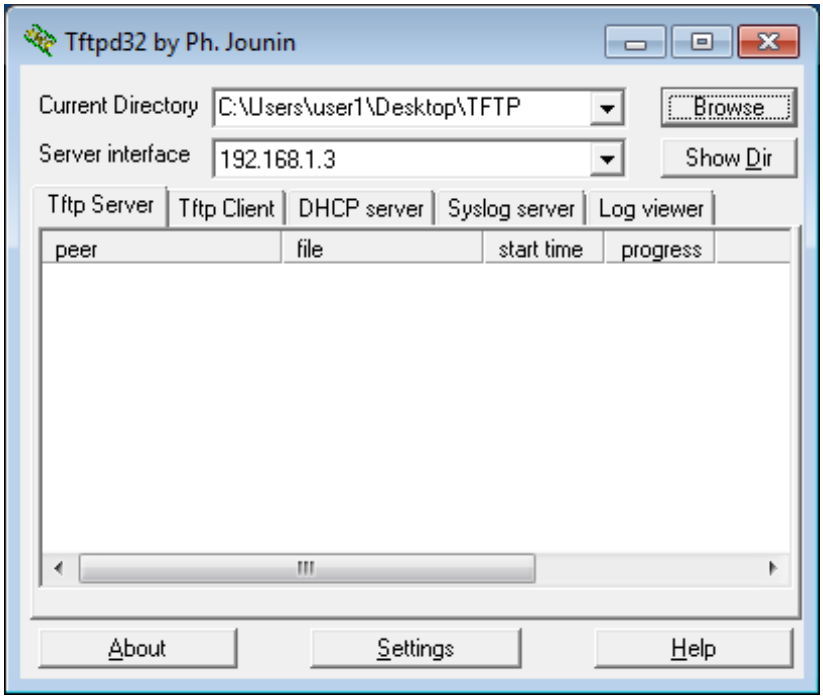
Étape 2 : Préparez le serveur TFTP sur le PC.

a. S'il n'existe pas encore, créez un dossier sur le bureau de l'ordinateur appelé **TFTP**. Les fichiers du commutateur seront copiés à cet emplacement.

b. Démarrez **tftpd32** sur le PC.

c. Cliquez sur **Browse (Parcourir)** et remplacez le répertoire actuel par **C:\Users\user1\Desktop\TFTP** en remplaçant user1 par votre nom d'utilisateur.

Le serveur TFTP doit être similaire à celui-ci :



Notez que Current Directory (Répertoire actuel) indique l'utilisateur et l'interface du serveur (PC-A) avec l'adresse IP **192.168.1.3**.

d. Testez la possibilité de copier un fichier en utilisant TFTP à partir du commutateur vers le PC. Le cas échéant, procédez à un dépannage.

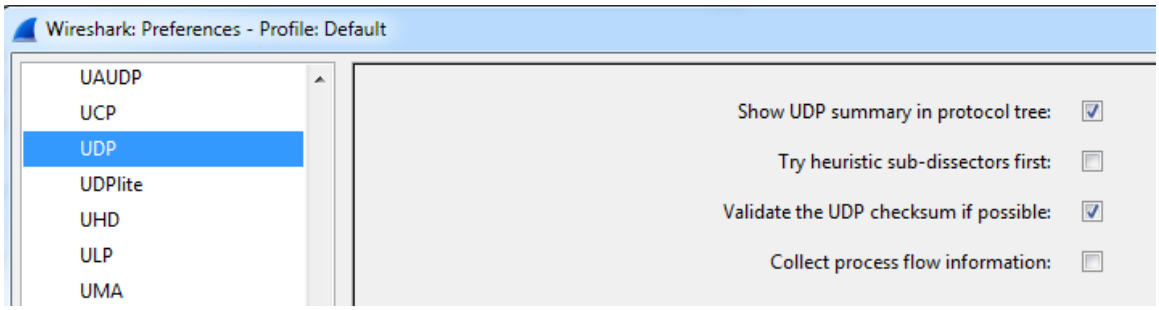
```
S1# copy start tftp
Address or name of remote host []?192.168.1.3
Destination filename [s1-confg]?
!!
1638 bytes copied in 0.026 secs (63000 bytes/sec)
```

Si vous voyez que le fichier a été copié (comme dans les résultats ci-dessus), vous êtes prêt à passer à l'étape suivante. Dans le cas contraire, effectuez un dépannage. Si vous obtenez l'erreur `%Error opening tftp (Permission denied)`, vérifiez d'abord que votre pare-feu ne bloque pas le protocole TFTP et que vous effectuez la copie vers un emplacement pour lequel votre nom d'utilisateur dispose des autorisations appropriées, comme l'ordinateur de bureau.

Étape 3 : Capture d'une session TFTP dans Wireshark

a. Ouvrez Wireshark. À partir du menu **Edit (Edition)**, choisissez **Preferences** etcliquez sur le signe (+) pour développer **Protocols**. Faites défiler vers le bas, puis sélectionnez **UDP**. Activez la case à cocher

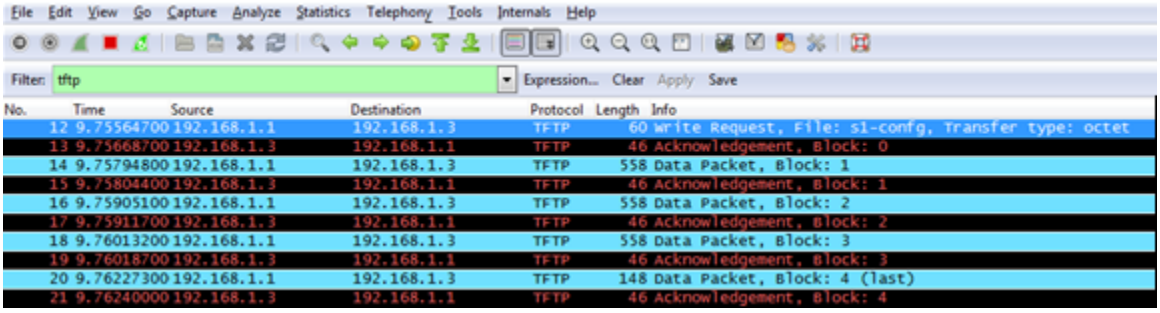
Validate the UDP checksum if possible (Valider la somme de contrôle UDP si possible) et cliquez sur **Apply (Appliquer)**. Cliquez ensuite sur **OK**.



b. Démarrez une capture Wireshark.

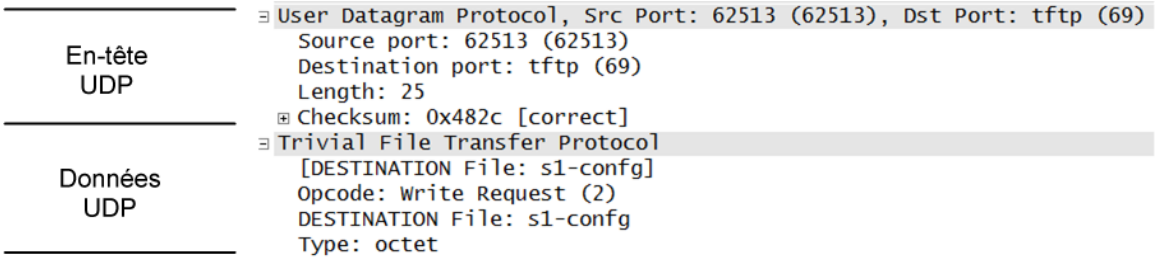
c. Exécutez la commande **`copy start tftp`** sur le commutateur.

d. Arrêtez la capture Wireshark.



e. Définissez le filtre sur **tftp**. Les informations affichées doivent être similaires à celles figurant ci-dessus : Ce transfert TFTP permet d'analyser les opérations UDP de la couche transport.

Dans Wireshark, des informations détaillées UDP 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.

SEGMENT UDP

0	16	31
PORT SOURCE UDP	PORT DE DESTINATION UDP	
LONGUEUR DU MESSAGE UDP	SOMME DE CONTRÔLE UDP	
DONNÉES		
DONNÉES...		

À l'aide de la capture Wireshark du premier datagramme IDP, 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. Renseignez les informations relatives à 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 :	

```
⊟ User Datagram Protocol, Src Port: 58565 (58565), Dst Port: 62513 (62513)
    Source port: 58565 (58565)
    Destination port: 62513 (62513)
    Length: 12
  ⊞ Checksum: 0x8372 [incorrect, should be 0xa385 (maybe caused by "UDP checksum offload"?)]
⊟ Trivial File Transfer Protocol
    [DESTINATION File: s1-confg]
    Opcode: Acknowledgement (4)
    Block: 0
```

Remarque : le datagramme UDP de retour possède un port de source UDP différent. Toutefois, ce dernier sert au transfert TFTP restant. Comme 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 somme de contrôle UDP. Pour plus d'informations sur la raison de cet événement, effectuez une recherche sur « UDP checksum offload » (déchargement de somme de contrôle).

Remarques générales

Cestravauxpratiquesontpermisd'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 ?

Défi

Étant donné que les protocoles FTP et TFTP ne sont pas sécurisés, toutes les données transférées sont envoyées en texte clair. Ceci comprend les ID d'utilisateurs, les mots de passe ou le contenu des fichiers en texte clair. L'analyse de la session FTP de couche supérieure permet d'identifier rapidement l'ID d'utilisateur, le mot de passe ainsi que les mots de passe du fichier de configuration. L'analyse des données TFTP de couche supérieure est un peu plus complexe. Toutefois, le champ de données peut être examiné et les informations d'ID d'utilisateur et de mot de passe pour la configuration peuvent être extraites.

Nettoyage

Sauf indication contraire de votre instructeur :

1) Supprimez les fichiers qui ont été copiés sur votre ordinateur.
2) Supprimez les configurations sur le commutateur **S1**.
3) Supprimez l'adresse IP manuelle du PC et restaurez la connectivité Internet.