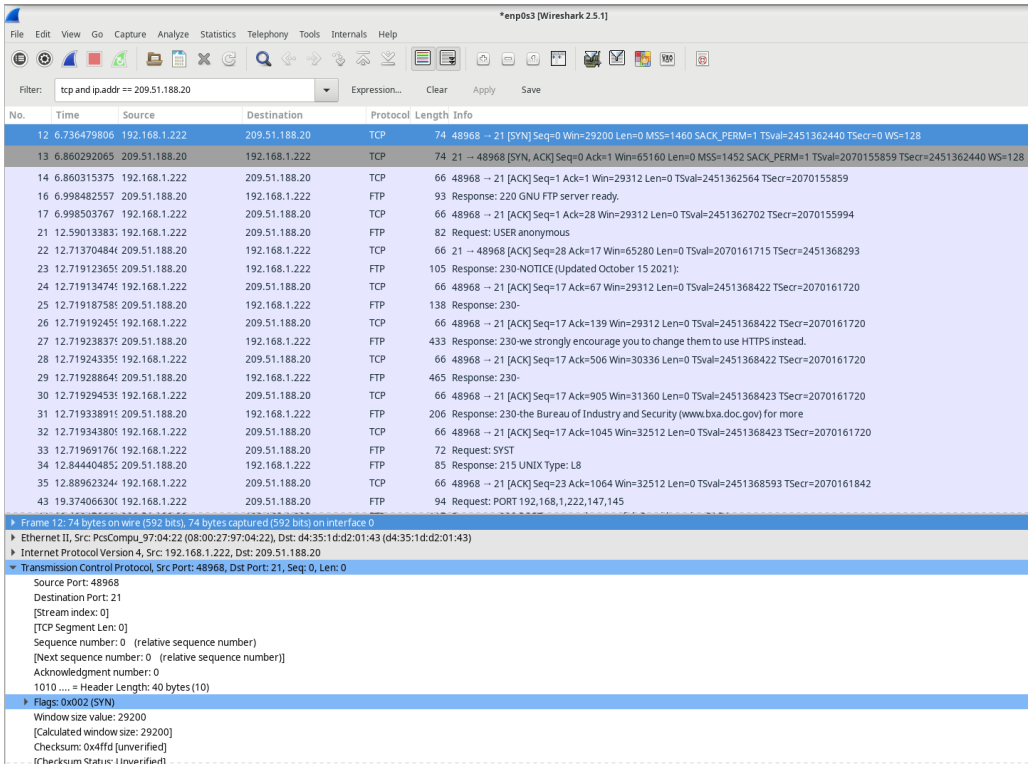
Utilizzo di Wireshark per Esaminare le Catture TCP e UDP

Identificare i campi dell'intestazione TCP e il funzionamento utilizzando una cattura di sessione FTP in Wireshark.

Per esaminare TCP, ho avviato una sessione FTP verso un server pubblico. A causa di restrizioni riscontrate nell'accesso anonimo al server ftp.cdc.gov indicato nella guida, ho utilizzato il server alternativo ftp.gnu.org.

Ho avviato Wireshark per catturare il traffico sull'interfaccia di rete principale della VM. Successivamente, ho utilizzato il client ftp da terminale per connettermi al server, autenticarmi come utente anonymous e eseguire i comandi ls e get README. Al

termine delle operazioni, ho interrotto la cattura Wireshark e applicato filtri per isolare i pacchetti rilevanti.



L'analisi dei segmenti TCP catturati ha permesso le seguenti osservazioni:

Instaurazione della Connessione: Sebbene non fosse l'obiettivo primario, la cattura ha mostrato l'handshake a tre vie (SYN, SYN-ACK, ACK) per stabilire la connessione di controllo sulla porta TCP 21, confermando la natura orientata alla connessione di TCP. Connessioni TCP separate sono state stabilite per i trasferimenti dati (es. ls, get).

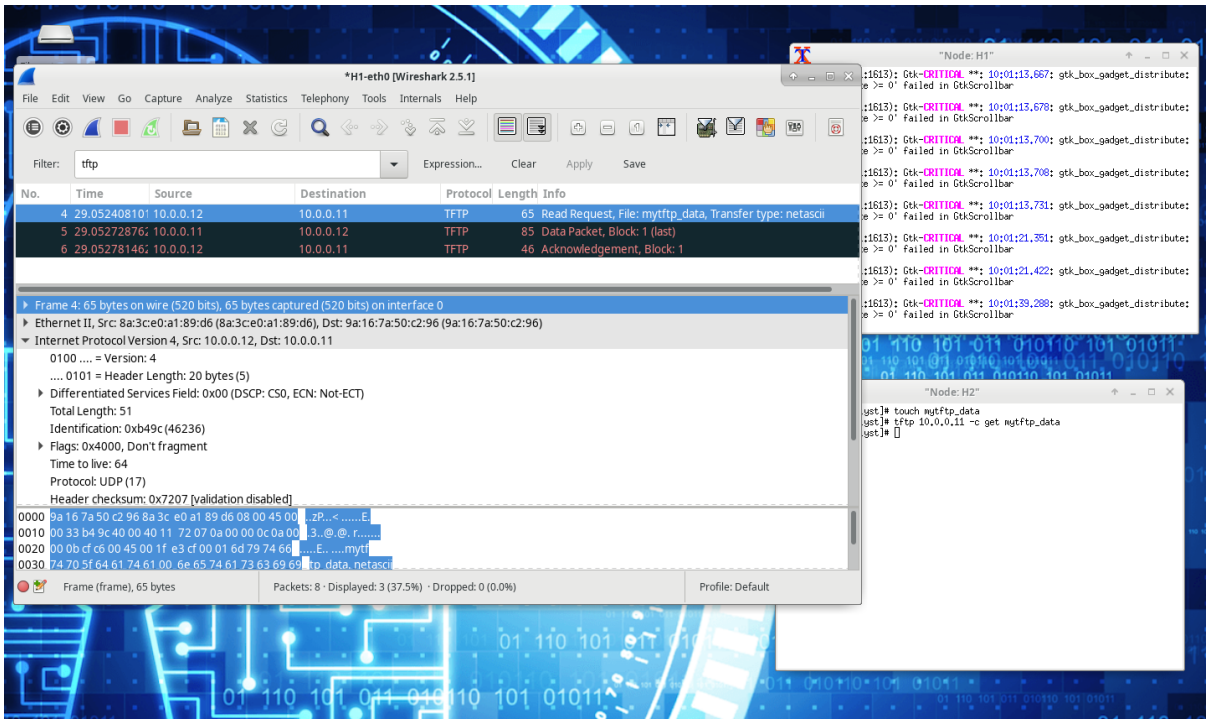
Affidabilità e Ordinamento: La progressione dei valori nei campi Sequence Number e Acknowledgment Number è stata chiaramente osservata. Questi meccanismi sono risultati fondamentali per garantire la consegna ordinata e affidabile dei dati relativi ai comandi FTP, alle risposte del server e al contenuto del file trasferito. Ogni segmento di dati trasmesso richiedeva un corrispondente ACK.

Gestione della Connessione e Flusso: I flag TCP sono stati ispezionati: SYN e ACK per la gestione della connessione; ACK per la conferma dei dati ricevuti; PSH per indicare la necessità di consegnare immediatamente i dati all'applicazione; FIN per la chiusura ordinata delle connessioni. Il campo Window Size mostrava valori dinamici, indicativi del meccanismo di controllo di flusso implementato da TCP.

Struttura dell'Intestazione: È stata verificata la presenza e la funzione dei campi standard dell'header TCP (Porte Sorgente/Destinazione, Numeri di Sequenza/Ack, Flag, Window Size, Checksum, Urgent Pointer - sebbene non utilizzato in questo caso).

Identificare i campi dell'intestazione UDP e il funzionamento utilizzando una cattura di sessione TFTP in Wireshark.

Per l'analisi di UDP, ho configurato un ambiente TFTP utilizzando Mininet. Ho avviato un server tftpd sull'host H1 e ho utilizzato l'host H2 come client. Ho creato un file di test (my_tftp_data) su H1. Ho avviato Wireshark su H1 per catturare il traffico sull'interfaccia H1-eth0. Da H2, ho eseguito il comando tftp 10.0.0.11 -c get my_tftp_data per scaricare il file. Completato il trasferimento, ho fermato la cattura e applicato filtri (tftp) in Wireshark.



L'esame dei datagrammi UDP ha rivelato un comportamento distinto rispetto a TCP:

Natura Connectionless: Non è stato osservato alcun handshake preliminare. La comunicazione è iniziata direttamente con l'invio del pacchetto TFTP RRQ (Read Request) da H2 alla porta UDP 69 di H1.

Mancanza di Affidabilità Intrinseca: UDP non implementa meccanismi intrinseci per garantire la consegna, l'ordine o la duplicazione dei pacchetti.

Affidabilità a Livello Applicativo: La necessaria affidabilità per il trasferimento del file è stata gestita interamente dal protocollo TFTP stesso. Questo è stato evidente dall'osservazione dello scambio di pacchetti DATA (contenenti blocchi di file numerati progressivamente) e pacchetti ACK specifici di TFTP, utilizzati per confermare la ricezione di ciascun blocco. La gestione di ritrasmissioni in caso di perdita di pacchetti DATA o ACK sarebbe a carico di TFTP, non di UDP.