Esercizio 4: Analisi Comparativa del Traffico HTTP e HTTPS con Wireshark

Obiettivo dell'Esercitazione

L'obiettivo di questo laboratorio è dimostrare la differenza fondamentale in termini di sicurezza tra il protocollo HTTP e il protocollo HTTPS. Attraverso l'uso dello strumento di cattura pacchetti **tcpdump** e dell'analizzatore di rete **Wireshark**, è stato esaminato il traffico generato durante l'autenticazione su due siti web differenti. Il motivo principale di questa simulazione è verificare in modo pratico come i dati sensibili (come le password) vengono trasmessi fisicamente sulla rete e quali vulnerabilità comportano le diverse configurazioni.

Fase 1: Analisi del Traffico in Chiaro (HTTP)

1. Identificazione dell'interfaccia di rete

- **Azione:** È stato utilizzato il comando **ip address** nel terminale per identificare l'interfaccia di rete locale (risultata essere **enp0s3** con IP **10.0.2.15**).

- **Motivo:** Per poter "ascoltare" e catturare il traffico di rete, è necessario prima indicare al software di cattura (come tcpdump) su quale specifica "porta" o scheda di rete della macchina virtuale mettersi in ascolto.

2. Cattura del traffico e simulazione di login

- **Azione:** È stata avviata la cattura del traffico salvando l'output in un file denominato **httpdump.pcap**. Successivamente, è stata effettuata una simulazione di accesso inserendo le credenziali "Admin” nel sito vulnerabile

http://testphp.vulnweb.com/login.php.

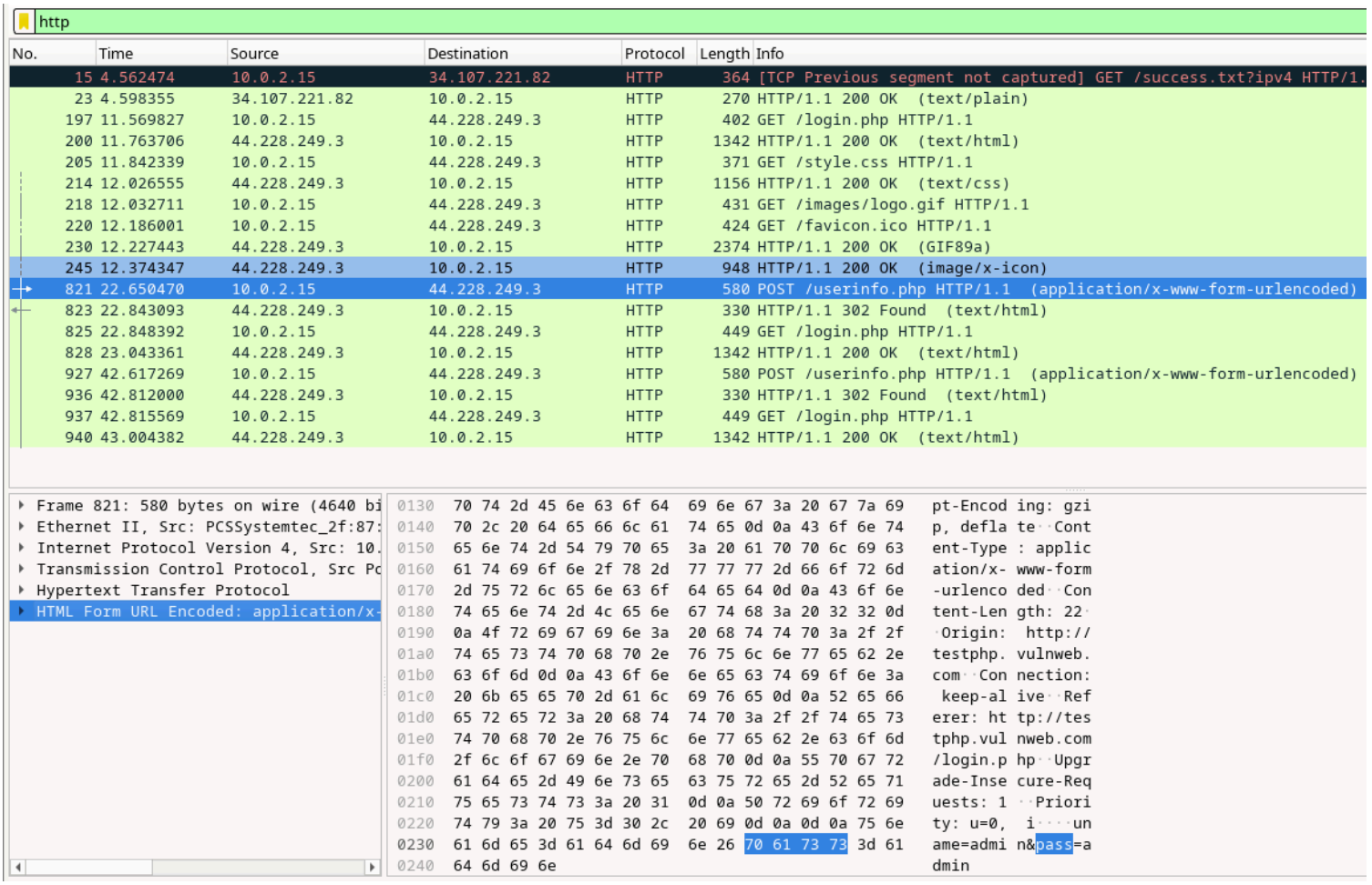
- **Motivo:** Salvare il traffico in un file **.pcap** (Packet Capture) permette di registrare tutti i pacchetti in transito per poi poterli analizzare con calma offline tramite Wireshark. Il sito "testphp" è stato scelto appositamente perché utilizza il vecchio protocollo HTTP, permettendoci di generare traffico non protetto da analizzare.

3. Analisi dei pacchetti HTTP con Wireshark

- **Azione:**Aprendo il file in Wireshark, è stato applicato il filtro **http** per isolare il pacchetto **POST** inviato al server durante il login.

- **Motivo:** In una rete transitano migliaia di pacchetti al secondo. Il filtro **http** elimina il "rumore di fondo". È stato cercato specificamente il pacchetto **POST** perché, nel protocollo HTTP, questo è il metodo standard utilizzato dai browser per "inviare" i dati compilati in un modulo (come username e password) verso il server.

Risultati della Fase 1: Espandendo la sezione "HTML Form URL Encoded" del pacchetto POST, è emerso chiaramente che il protocollo HTTP non offre alcuna salvaguardia per i dati scambiati. Le informazioni critiche inserite (nome utente e password) vengono trasmesse e visualizzate completamente in chiaro. Questo rende i dati estremamente vulnerabili a qualsiasi intercettazione da parte di un utente malintenzionato in ascolto sulla stessa rete.

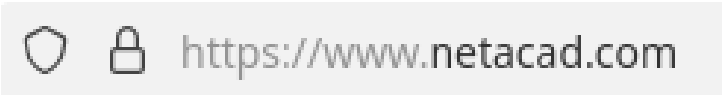


Fase 2: Analisi del Traffico Cifrato (HTTPS)

1. Navigazione su sito protetto

- **Azione:** È stata avviata una nuova cattura di rete (**httpsdump.pcap**). La navigazione è stata poi effettuata verso la piattaforma **www.netacad.com**. Dal browser si nota l'utilizzo del protocollo **https://** e la presenza di un lucchetto chiuso.

- **Motivo:** Questo passaggio serve a generare un set di dati di confronto. L'icona del lucchetto è un indicatore visivo progettato per segnalare all'utente che il client e il server hanno stabilito con successo una connessione che utilizza la crittografia.



2. Analisi del traffico sulla porta 443

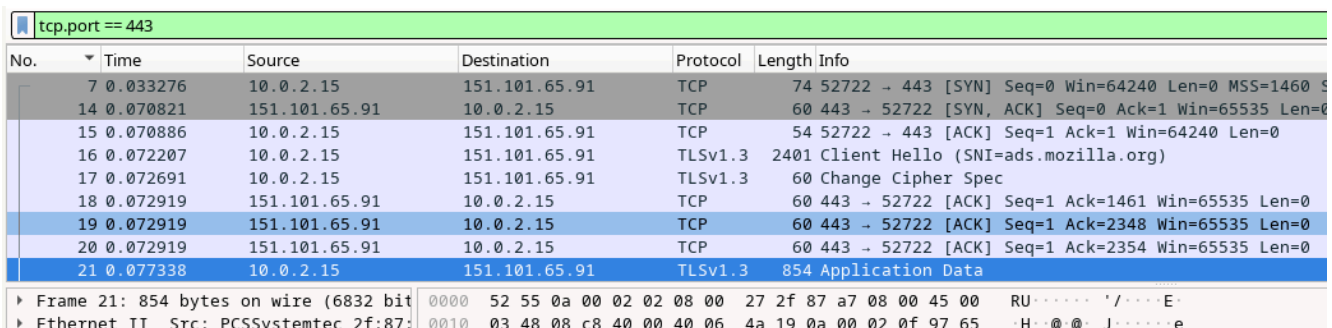
- **Azione:** Su Wireshark, la nuova cattura è stata filtrata utilizzando la stringa **tcp.port == 443**.

- **Motivo:** Poiché il traffico è cifrato, Wireshark non può più riconoscere i dati semplicemente cercando la parola "http". È stato quindi necessario filtrare per la porta logica numero 443, che è lo standard universale su cui viaggia il traffico sicuro HTTPS.

3. Verifica dell'efficacia della cifratura

- **Azione:** Analizzando l'elenco dei pacchetti, la dicitura HTTP è scomparsa ed è stata sostituita da protocolli di sicurezza come "Transport Layer Security" (TLS). Espandendo un pacchetto etichettato come "Application Data", si nota che i dati non sono leggibili o in formato plain text.

- **Motivo:** L'obiettivo finale era verificare cosa vedrebbe un hacker intercettando questa connessione. Al posto delle password, appare esclusivamente una stringa esadecimale incomprensibile sotto la voce "Encrypted Application Data". Questo dimostra l'efficacia dell'algoritmo matematico, che ha nascosto con successo il vero significato dei dati scambiati.



Conclusioni

Dalla comparazione pratica dei due protocolli emergono due riflessioni fondamentali:

1. **Vantaggi dell'HTTPS:** L'analisi ha dimostrato che il vantaggio principale di HTTPS risiede nella riservatezza dei dati. Mentre HTTP trasmette informazioni sensibili in chiaro, rendendole leggibili a chiunque le intercetti, HTTPS utilizza algoritmi di crittografia e certificati per rendere il traffico incomprensibile a terze parti. Un attore malevolo otterrebbe solo stringhe di dati cifrati inutilizzabili.

2. **Affidabilità e percezione della sicurezza:** È cruciale comprendere che la presenza di HTTPS e del "lucchetto" non rende automaticamente un sito web "affidabile" o intrinsecamente

sicuro. Il lucchetto garantisce esclusivamente che il canale di trasmissione sia cifrato, ma non certifica in alcun modo l'onestà di chi possiede il server dall'altra parte. Poiché gli attori malevoli usano sempre più spesso l'HTTPS per dare una falsa percezione di sicurezza ai loro siti di phishing, si raccomanda di scambiare dati solo con entità di cui si possiede una fiducia verificata.