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

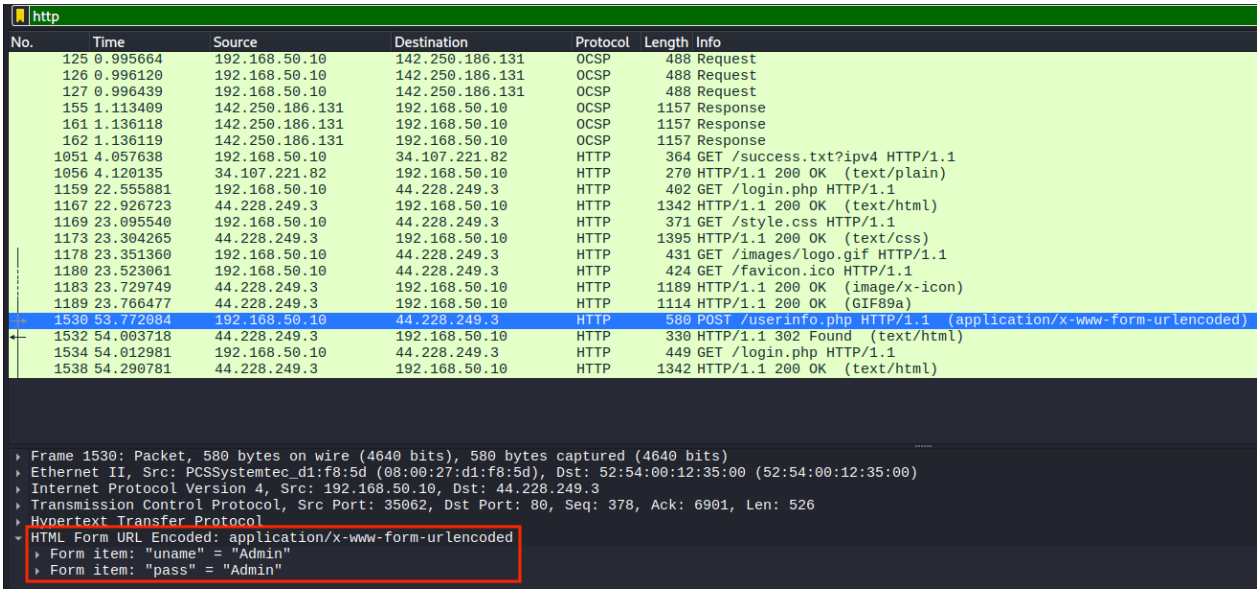
A cura di Iris Canole, Federico Giannini, Daniele Castello, Luca Pani, Rosario Papa, Yari Olmi, Alessandro Salerno

Questo report documenta le procedure e i risultati dell'esercizio di laboratorio incentrato sulla cattura e l'analisi del traffico di rete **HTTP** e **HTTPS** utilizzando `tcpdump` e Wireshark. L'obiettivo primario dell'esercizio è osservare le differenze pratiche nella sicurezza dei dati trasmessi tra i due protocolli, evidenziando le vulnerabilità intrinseche del traffico in chiaro e l'efficacia della crittografia.

Parte 1: Cattura e Analisi del Traffico HTTP

La prima parte dell'esercizio si è concentrata sulla cattura di traffico HTTP **non crittografato**. L'obiettivo era osservare in modo diretto come i dati, incluse informazioni sensibili come le credenziali di accesso, vengono trasmessi in chiaro attraverso la rete. Comprendere questa vulnerabilità è strategicamente importante per apprezzare i meccanismi di sicurezza implementati nei protocolli moderni.

La procedura ha previsto l'uso del comando tcpdump sull'interfaccia di rete `eth0` per catturare tutto il traffico generato durante un tentativo di login al sito di test `http://testphp.vulnweb.com/login.php`. I pacchetti catturati sono stati salvati nel file `httpdump.pcap` per la successiva analisi con Wireshark.



Quali due informazioni vengono visualizzate?

L'analisi del file di cattura in Wireshark, e in particolare del pacchetto `1530`, rivela la trasmissione delle credenziali di accesso in formato **plaintext (testo in chiaro)**. Espandendo i dettagli del protocollo HTTP, nella sezione "HTML Form URL Encoded" sono chiaramente visibili le due informazioni inviate dal client al server, associate ai rispettivi nomi dei campi del form:

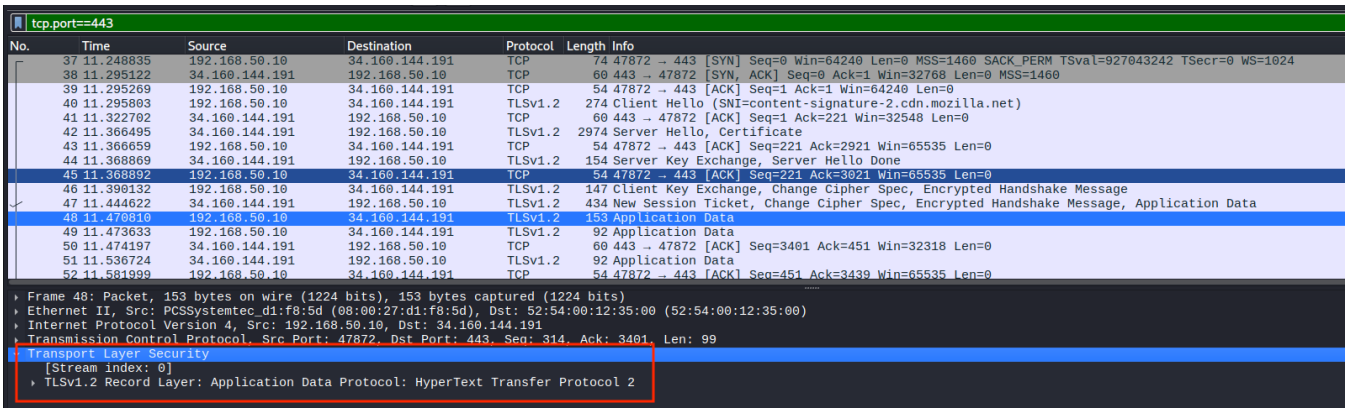
- `uname`: `Admin`
- `pass`: `Admin`

Questa osservazione dimostra in modo inequivocabile come **chiunque** sia in grado di **intercettare** il traffico di rete (*eavesdropping*) possa leggere le credenziali senza alcuna difficoltà. Il chiaro rischio per la sicurezza evidenziato da questa cattura sottolinea la necessità di meccanismi di protezione, che verranno analizzati nella parte successiva dedicata al traffico crittografato.

Parte 2: Cattura e Analisi del Traffico HTTPS

Questa seconda parte dell'esercizio si è focalizzata sull'analisi del traffico HTTPS per dimostrare come l'implementazione della crittografia protegga efficacemente i dati in transito. L'obiettivo era confrontare i risultati ottenuti con quelli della precedente analisi HTTP, evidenziando il ruolo del livello di **sicurezza aggiuntivo**.

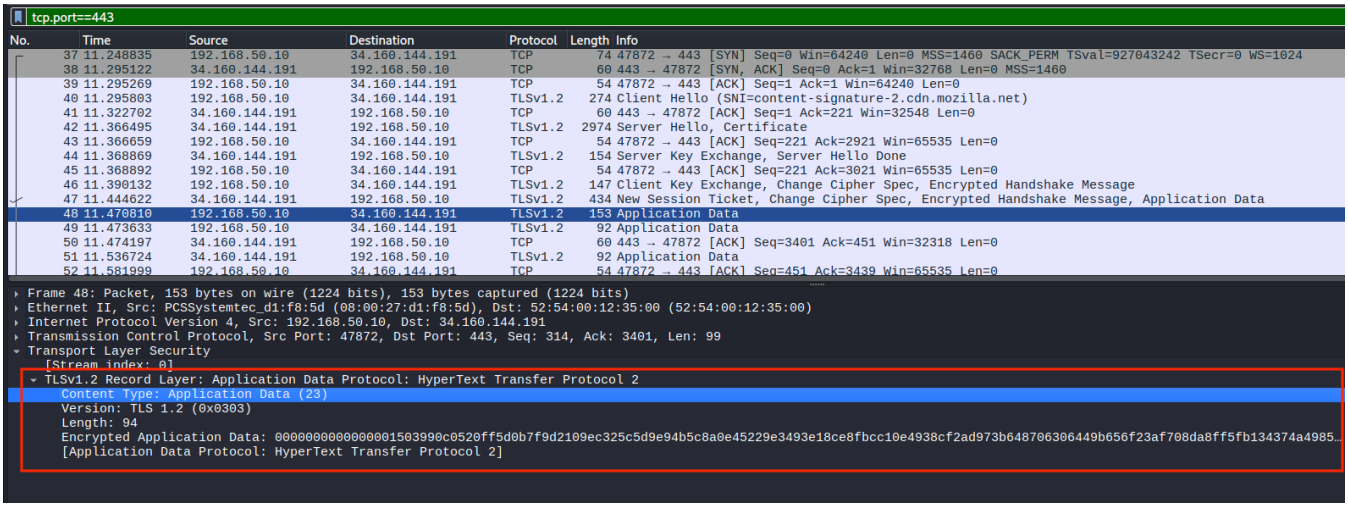
La procedura è stata analoga alla precedente, catturando il traffico generato durante l'accesso a un sito che utilizza HTTPS. Il traffico è stato salvato nel file `httpsdump.pcap`. All'interno di Wireshark, è stato applicato il filtro `tcp.port==443` per isolare e analizzare esclusivamente i pacchetti relativi alla sessione HTTPS.



Cosa ha sostituito la sezione HTTP che era nel file di cattura precedente?

L'analisi del pacchetto rivela che i dati dell'applicazione non sono più direttamente visibili sotto il

livello TCP. La sezione HTTP in chiaro, presente nella cattura precedente, è stata sostituita da un nuovo livello protocollare: **Transport Layer Security (TLS)**. Più specificamente, come mostrato nei dettagli del pacchetto, i dati sono incapsulati all'interno del **TLSv1.2 Record Layer**, che gestisce la crittografia dei dati dell'applicazione prima della loro trasmissione.



I dati dell'applicazione sono in formato plaintext o leggibile?
Esaminando il contenuto del livello TLS, si osserva un campo denominato **"Encrypted Application Data"**. Il contenuto di questo campo è una stringa di byte esadecimali non interpretabile da un essere umano. Si può quindi concludere in modo definitivo che i dati dell'applicazione **non sono in plaintext e non sono leggibili**.

Tuttavia, un'analisi più approfondita rivela un dettaglio significativo: lo strumento di analisi di Wireshark è in grado di identificare il tipo di protocollo incapsulato nel flusso crittografato, segnalandolo come `[Application Data Protocol: HyperText Transfer Protocol 2]`. Questo dimostra che, sebbene il contenuto dei dati sia protetto e illeggibile (*ciphertext*), i moderni analizzatori di protocollo possono comunque derivare metadati sul tipo di traffico che transita all'interno del tunnel TLS.

Questo risultato dimostra come HTTPS riesca a nascondere i dati, garantendo la confidenzialità della comunicazione e mitigando i rischi di intercettazione visti nell'analisi del traffico HTTP.

Domande di Riflessione

Questa sezione finale passa dall'osservazione pratica dei pacchetti alla riflessione sulle implicazioni concettuali e di sicurezza più ampie. Sulla base delle prove raccolte durante l'esercizio, vengono analizzati i vantaggi di HTTPS e i limiti della sua garanzia di affidabilità.

Vantaggi dell'Uso di HTTPS

Quali sono i vantaggi dell'uso di HTTPS invece di HTTP?
Il vantaggio principale e fondamentale dell'uso di HTTPS rispetto a HTTP è la **crittografia**. Come dimostrato empiricamente in questo laboratorio, HTTPS utilizza un protocollo di sicurezza (TLS) per applicare un algoritmo matematico che nasconde il vero significato dei dati scambiati.

Questo si traduce in un beneficio cruciale:

- **Confidenzialità:** Impedisce a terzi di intercettare e leggere informazioni sensibili in transito (*eavesdropping*). La cattura HTTPS ha mostrato dati illeggibili ("Encrypted Application Data"), mentre la cattura HTTP ha esposto le credenziali di login in chiaro.

Affidabilità dei Siti HTTPS

Tutti i siti web che usano HTTPS sono considerati affidabili?
No, non tutti i siti che usano HTTPS sono necessariamente affidabili. Sebbene HTTPS garantisca che la *comunicazione* tra il browser e il server del sito sia crittografata e sicura da intercettazioni, **non fornisce alcuna garanzia sull'affidabilità o sulle intenzioni del sito stesso**.

Come indicato nel materiale dell'esercizio, "Solo perché un sito usa HTTPS non significa che sia un sito affidabile". Gli attori malevoli utilizzano comunemente HTTPS proprio per nascondere le loro attività illecite, il che significa che la crittografia protegge la comunicazione dannosa tanto quanto quella legittima.

Conclusione

Questo esercizio di laboratorio ha dimostrato con successo il ruolo essenziale di HTTPS nel proteggere la confidenzialità dei dati in transito. L'analisi comparativa dei pacchetti ha evidenziato la vulnerabilità critica di HTTP, che espone dati sensibili in chiaro, e la robusta protezione offerta dalla crittografia di HTTPS. In sintesi, l'attività ha confermato il ruolo indispensabile di HTTPS nel garantire la confidenzialità dei dati, un principio fondamentale della sicurezza delle comunicazioni di rete.