

# 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.

http						
No.	Time	Source	Destination	Protocol	Length	Info
15	4.562474	10.0.2.15	34.107.221.82	HTTP	364	[TCP Previous segment not captured] GET /success.txt?ip=10.0.2.15 HTTP/1.1
23	4.598355	34.107.221.82	10.0.2.15	HTTP	270	HTTP/1.1 200 OK (text/plain)
197	11.569827	10.0.2.15	44.228.249.3	HTTP	402	GET /login.php HTTP/1.1
200	11.763706	44.228.249.3	10.0.2.15	HTTP	1342	HTTP/1.1 200 OK (text/html)
205	11.842339	10.0.2.15	44.228.249.3	HTTP	371	GET /style.css HTTP/1.1
214	12.026555	44.228.249.3	10.0.2.15	HTTP	1156	HTTP/1.1 200 OK (text/css)
218	12.032711	10.0.2.15	44.228.249.3	HTTP	431	GET /images/logo.gif HTTP/1.1
220	12.186001	10.0.2.15	44.228.249.3	HTTP	424	GET /favicon.ico HTTP/1.1
230	12.227443	44.228.249.3	10.0.2.15	HTTP	2374	HTTP/1.1 200 OK (GIF89a)
245	12.374347	44.228.249.3	10.0.2.15	HTTP	948	HTTP/1.1 200 OK (image/x-icon)
821	22.650470	10.0.2.15	44.228.249.3	HTTP	580	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
823	22.843093	44.228.249.3	10.0.2.15	HTTP	330	HTTP/1.1 302 Found (text/html)
825	22.848392	10.0.2.15	44.228.249.3	HTTP	449	GET /login.php HTTP/1.1
828	23.043361	44.228.249.3	10.0.2.15	HTTP	1342	HTTP/1.1 200 OK (text/html)
927	42.617269	10.0.2.15	44.228.249.3	HTTP	580	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
936	42.812000	44.228.249.3	10.0.2.15	HTTP	330	HTTP/1.1 302 Found (text/html)
937	42.815569	10.0.2.15	44.228.249.3	HTTP	449	GET /login.php HTTP/1.1
940	43.004382	44.228.249.3	10.0.2.15	HTTP	1342	HTTP/1.1 200 OK (text/html)

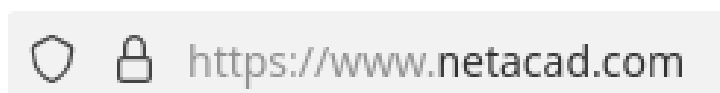
  

▶ Frame 821: 580 bytes on wire (4640 bits captured) capture length 580 bytes ▶ Ethernet II, Src: PCSysntec_2f:87:61, Dst: 10.0.2.15 ▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 44.228.249.3 ▶ Transmission Control Protocol, Src Port: 54432, Dst Port: 80, Seq: 305212800, Win: 65535, Len: 580 ▶ Hypertext Transfer Protocol ▶ HTML Form URL Encoded: application/x-www-form-urlencoded	0130 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 0140 70 2c 20 64 65 66 6c 61 74 65 0d 0a 43 6f 6e 74 0150 65 6e 74 2d 54 79 70 65 3a 20 61 70 70 6c 69 63 0160 61 74 69 6f 6e 2f 78 2d 77 77 77 2d 66 6f 72 6d 0170 2d 75 72 6c 65 6e 63 6f 64 65 64 0d 0a 43 6f 6e 0180 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 32 32 0d 0190 0a 4f 72 69 67 69 6e 3a 20 68 74 74 70 3a 2f 2f 01a0 74 65 73 74 70 68 70 2e 76 75 6c 6e 77 65 62 2e 01b0 63 6f 6d 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 01c0 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 52 65 66 01d0 65 72 65 72 3a 20 68 74 74 70 3a 2f 2f 74 65 73 01e0 74 70 68 70 2e 76 75 6c 6e 77 65 62 2e 63 6f 6d 01f0 2f 6c 6f 67 69 6e 2e 70 68 70 0d 0a 55 70 67 72 0200 61 64 65 2d 49 6e 73 65 63 75 72 65 2d 52 65 71 0210 75 65 73 74 73 3a 20 31 0d 0a 50 72 69 6f 72 69 0220 74 79 3a 20 75 3d 30 2c 20 69 0d 0a 0d 0a 75 6e 0230 61 6d 65 3d 61 64 6d 69 6e 26 70 61 73 73 3d 61 0240 64 6d 69 6e	Content-Type: application/x-www-form-urlencoded Content-Length: 22 Origin: http://testphp.vulnweb.com Connection: keep-alive Referer: http://testphp.vulnweb.com/login.php Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 6.0; rv:2.0) Gecko/20100101 Firefox/4.0.1 Name=admin&pass=admin
---	--	--

## 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.

No.	Time	Source	Destination	Protocol	Length	Info
7	0.033276	10.0.2.15	151.101.65.91	TCP	74	52722 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S
14	0.070821	151.101.65.91	10.0.2.15	TCP	60	443 → 52722 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
15	0.070886	10.0.2.15	151.101.65.91	TCP	54	52722 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
16	0.072207	10.0.2.15	151.101.65.91	TLSv1.3	2401	Client Hello (SNI=ads.mozilla.org)
17	0.072691	10.0.2.15	151.101.65.91	TLSv1.3	60	Change Cipher Spec
18	0.072919	151.101.65.91	10.0.2.15	TCP	60	443 → 52722 [ACK] Seq=1 Ack=1461 Win=65535 Len=0
19	0.072919	151.101.65.91	10.0.2.15	TCP	60	443 → 52722 [ACK] Seq=1 Ack=2348 Win=65535 Len=0
20	0.072919	151.101.65.91	10.0.2.15	TCP	60	443 → 52722 [ACK] Seq=1 Ack=2354 Win=65535 Len=0
21	0.077338	10.0.2.15	151.101.65.91	TLSv1.3	854	Application Data

▶ Frame 21: 854 bytes on wire (6832 bits) captured on interface eth0 (0.077338 seconds)

Ethernet II	Src: PCSSystemtec 2f:87:0010	03 48 08 c8 40 00	4a 19 0a 00 02 0f	97 65	...
...	...	...	...	...	...

## 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.