

Build Week 3 – Esercizio 4: Usare Wireshark per Esaminare il traffico HTTP e HTTPS

Obiettivi:

- **Parte 1:** Catturare e visualizzare il traffico HTTP
- **Parte 2:** Catturare e visualizzare il traffico HTTPS

Contesto/Scenario

HyperText Transfer Protocol (HTTP) è un protocollo a livello di applicazione che presenta i dati tramite un browser web. Con HTTP, non c'è salvaguardia per i dati scambiati tra due dispositivi comunicanti. Con HTTPS, viene utilizzata la crittografia tramite un algoritmo matematico. Questo algoritmo nasconde il vero significato dei dati scambiati. Questo viene fatto attraverso l'uso di certificati che possono essere visualizzati più avanti in questo laboratorio. Indipendentemente da HTTP o HTTPS, si raccomanda di scambiare dati solo con siti web di cui ci si fida. Solo perché un sito usa HTTPS non significa che sia un sito affidabile. Gli attori malevoli usano comunemente HTTPS per nascondere le loro attività.

Nel corso del laboratorio verrà esplorato e catturato il traffico HTTP e HTTPS mediante l'uso di Wireshark.

Risorse Richieste

- **Cyberops Workstation**
- **Connessione Internet**

Parte 1: Catturare e visualizzare il traffico HTTP

In questa fase si utilizza tcpdump per catturare il traffico HTTP e salvarlo in un file di cattura (pcap) tramite opzioni da riga di comando; i file pcap risultanti possono quindi essere analizzati con applicazioni compatibili, come Wireshark.

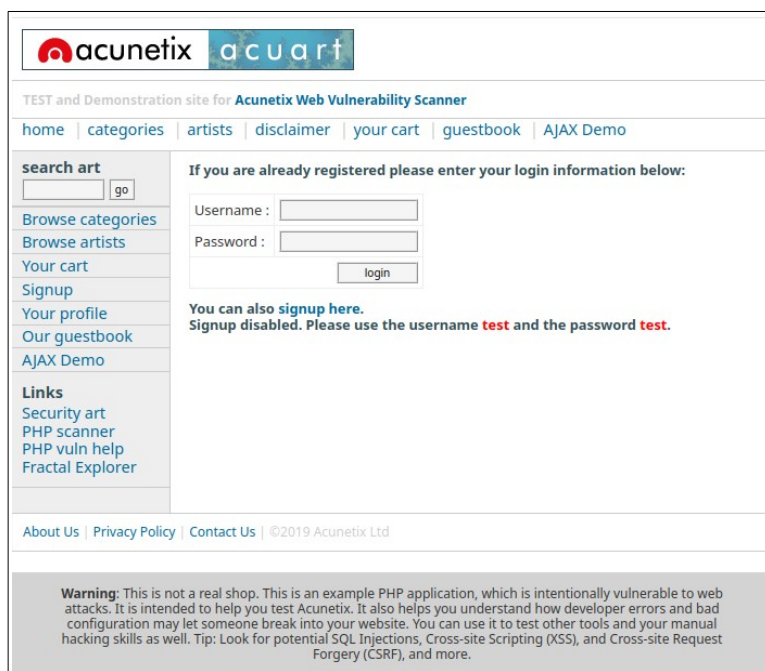
Passo 1: Avviare la macchina virtuale ed effettuare il login Per visualizzare l'interfaccia di rete attiva è sufficiente inserire il comando “ip address” nel terminale di Linux.

```
[analyst@secOps ~]$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:2f:87:a7 brd ff:ff:ff:ff:ff:ff
    altname enx0800272f87a7
    inet 10.0.2.15/24 metric 1024 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86378sec preferred_lft 86378sec
    inet6 fd17:625c:f037:2:a00:27ff:fe2f:87a7/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86381sec preferred_lft 14381sec
    inet6 fe80::a00:27ff:fe2f:87a7/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
```

Il passo successivo è avviare la cattura con tcpdump dal terminale di Kali usando “*sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap*”. Questo comando inoltre salva la cattura in un file pcap chiamato httpdump.pcap.

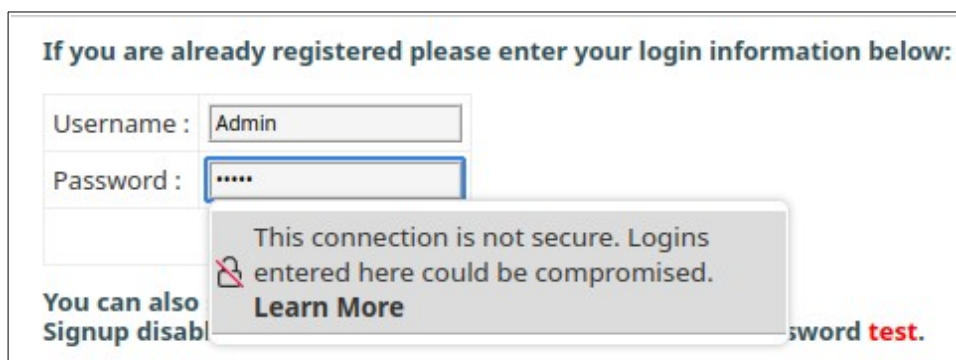
```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
[sudo] password for analyst:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

L'opzione -i indica l'interfaccia di rete, -s 0 imposta la lunghezza massima dei pacchetti catturati e -w salva la cattura in un file leggibile con Wireshark.
Aprendo il sito "http://testphp.vulnweb.com/login.php" via HTTP (non cifrato) si è inviato del traffico in chiaro dalla workstation.



The screenshot shows the Acunetix acuart website. At the top, there's a navigation bar with links: home, categories, artists, disclaimer, your cart, guestbook, and AJAX Demo. Below this is a search bar with the text "search art" and a "go" button. To the right of the search bar is a login section titled "If you are already registered please enter your login information below:". It contains fields for "Username:" and "Password:" with a "login" button. Below the login fields, there's a message: "You can also signup here. Signup disabled. Please use the username test and the password test." At the bottom of the page, there's a footer with links: About Us, Privacy Policy, Contact Us, and ©2019 Acunetix Ltd. A warning message is displayed at the very bottom: "Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more."

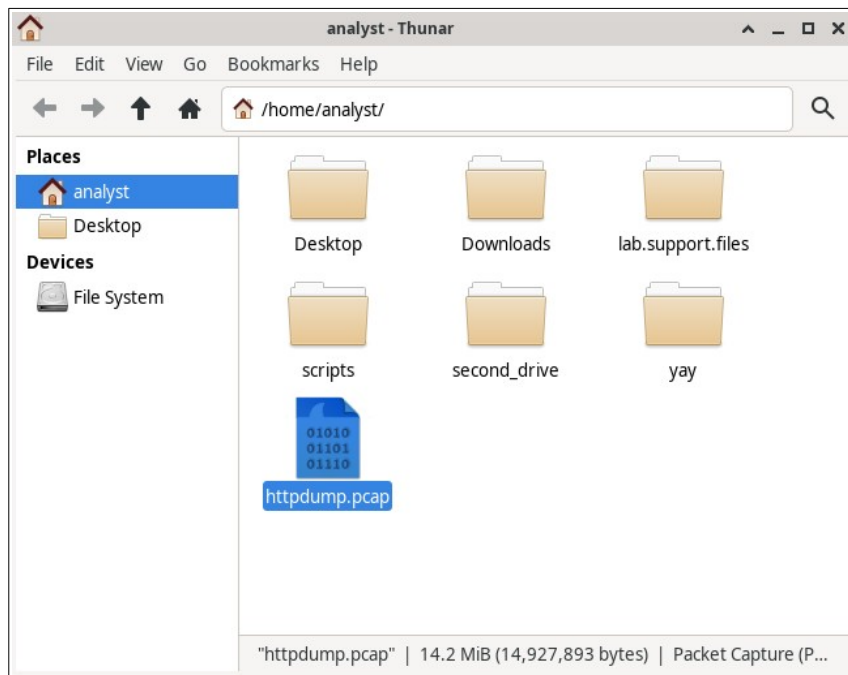
Nella schermata sottostante è mostrata la pagina di login dove sono stati inseriti username e password (Admin/Admin); essendo la connessione non protetta, le credenziali possono essere intercettate da chiunque catturi il traffico di rete (come illustrato dall'analisi pcap successiva).



This is a close-up of the login form from the previous screenshot. The "Username:" field contains the text "Admin". The "Password:" field contains six dots, representing the password "Admin". Below the password field, there's a message: "This connection is not secure. Logins entered here could be compromised. Learn More". To the left of this message, there's a link: "You can also Signup disabled". To the right, there's a link: "password test."

Interrompendo la cattura con "CTRL+C" tcpdump stampa a video una sintesi statistica dei pacchetti acquisiti e chiude il file di cattura.

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
[sudo] password for analyst:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C12590 packets captured
12591 packets received by filter
0 packets dropped by kernel
```



Nell'analisi del file di cattura con Wireshark, il traffico HTTP viene filtrato e si individua una richiesta di tipo POST inviata alla pagina di login. Selezionando il pacchetto corrispondente ed espandendo nella finestra inferiore la sezione HTML Form URL Encoded: application/x-www-form-urlencoded, vengono mostrati i campi del modulo trasmessi dal client al server.

http					
No.	Time	Source	Destination	Protocol	Length Info
9879	28.752762	10.0.2.15	44.228.249.3	HTTP	424 GET /favicon.ico HTTP/1.1
9887	28.808589	44.228.249.3	10.0.2.15	HTTP	2394 HTTP/1.1 200 OK (GIF89a)
9901	28.932281	44.228.249.3	10.0.2.15	HTTP	1189 HTTP/1.1 200 OK (image/x-icon)
12452	60.487931	10.0.2.15	34.107.221.82	HTTP	347 GET /canonical.html HTTP/1.1
12454	60.536499	34.107.221.82	10.0.2.15	HTTP	352 HTTP/1.1 200 OK (text/html)
12458	60.537243	10.0.2.15	34.107.221.82	HTTP	364 GET /success.txt?ipv4 HTTP/1.1
12469	60.559435	34.107.221.82	10.0.2.15	HTTP	270 HTTP/1.1 200 OK (text/plain)
12529	68.611035	10.0.2.15	44.228.249.3	HTTP	580 POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
12531	68.797572	44.228.249.3	10.0.2.15	HTTP	330 HTTP/1.1 302 Found (text/html)
12533	68.799360	10.0.2.15	44.228.249.3	HTTP	449 GET /login.php HTTP/1.1
12535	69.010143	44.228.249.3	10.0.2.15	HTTP	2802 HTTP/1.1 200 OK (text/html)
▶ Frame 12529: 580 bytes on wire (4640 bits), 580 bytes captured (4640 bits) ▶ Ethernet II, Src: PCSSystemtec_2f:87:a7 (08:00:27:2f:87:a7), Dst: 52:55:0a:00:02:02 (52:55:0a:00:02:02) ▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 44.228.249.3 ▶ Transmission Control Protocol, Src Port: 39274, Dst Port: 80, Seq: 726, Ack: 9649, Len: 526 ▶ Hypertext Transfer Protocol ▼ HTML Form URL Encoded: application/x-www-form-urlencoded ▶ Form item: "uname" = "Admin" ▶ Form item: "pass" = "Admin"					
					0000 52 55 0a 00 02 02 08 00 27 0010 02 36 f3 84 40 00 00 06 13 0020 f9 03 99 6a 00 50 3f a2 74 0030 ff ff 34 1f 00 00 50 4f 53 0040 69 6e 66 6f 2e 70 68 70 20 0050 31 0d 0a 48 6f 73 74 3a 20 0060 2e 76 75 6c 6e 77 65 62 2e 0070 65 72 2d 41 67 65 6e 74 3a

Quali due informazioni vengono visualizzate?

In questo caso Wireshark evidenzia i parametri username e password con i valori in chiaro, confermando che le credenziali sono state trasmesse senza alcuna cifratura sul canale HTTP.

Parte 2: Catturare e visualizzare il traffico HTTPS

Per l'analisi del traffico HTTPS viene avviato tcpdump da terminale su una workstation Linux. Durante l'acquisizione vengono generate connessioni HTTPS verso un sito web, così che i pacchetti scambiati vengano registrati in un file di cattura. Successivamente il file prodotto viene aperto in Wireshark per esaminare il contenuto delle comunicazioni cifrate.

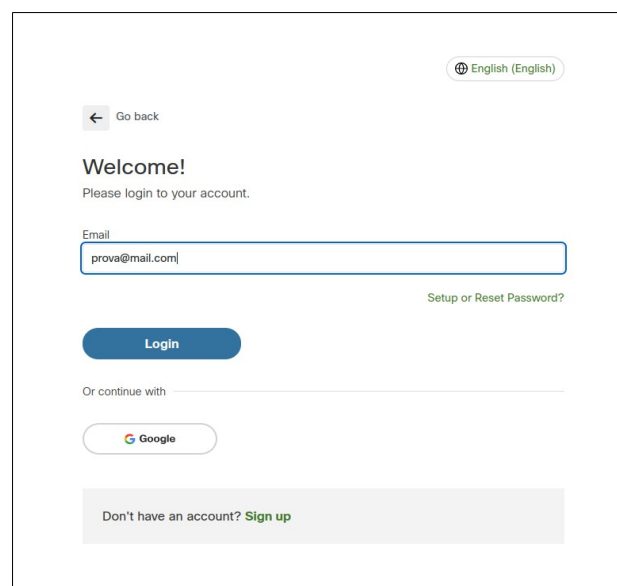
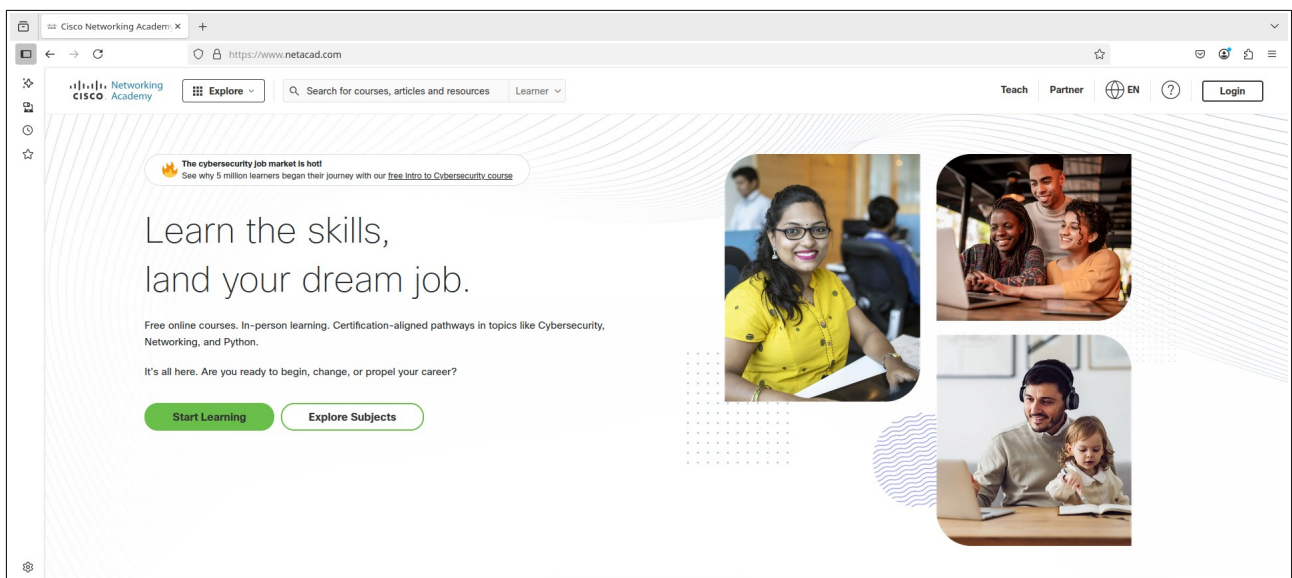
```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpsdump.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

Dal browser di Kali è stata avviata la navigazione verso il sito “www.netacad.com”



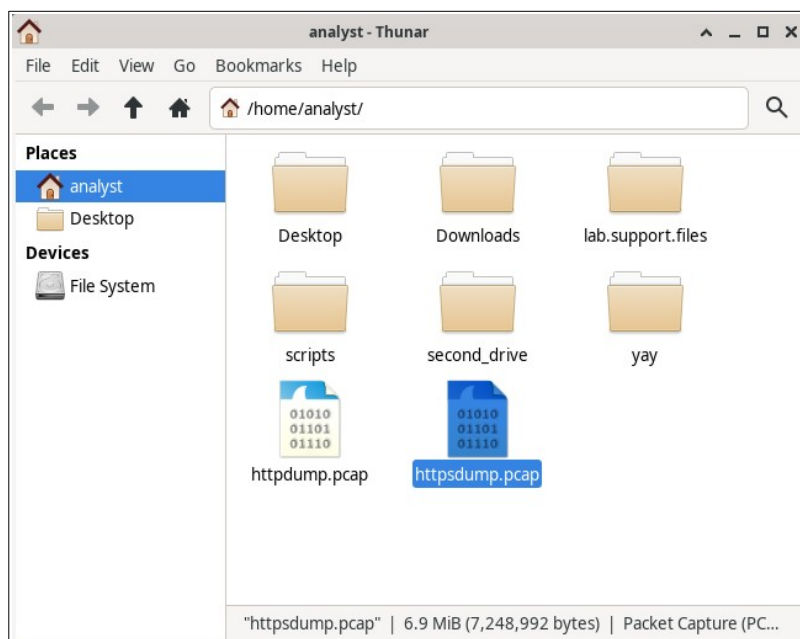
Cosa noti riguardo all'URL del sito web?

L'URL del sito web inizia con “https://”, indicando che la connessione utilizza il protocollo HTTPS. Ciò significa che i dati trasmessi sono cifrati tramite TLS, a differenza di HTTP in cui le informazioni viaggiano in chiaro.



Dopo il tentativo di login, la cattura dei pacchetti è stata interrotta dal terminale.

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpsdump.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C2050 packets captured
2052 packets received by filter
0 packets dropped by kernel
```



Come nell'analisi precedente, anche in questo caso è stato generato un file .pcap da esaminare. L'analisi in Wireshark è stata condotta espandendo l'area dei pacchetti e applicando il filtro "tcp.port==443" per isolare il traffico HTTPS. Tra i pacchetti risultanti è stato selezionato un messaggio di tipo Application Data, il cui contenuto è visibile nella finestra inferiore.

tcp.port == 443					
No.	Time	Source	Destination	Protocol	Length Info
25	0.184158	34.160.144.191	10.0.2.15	TLSv1.2	195 Server Key Exchange, Seq=...
26	0.184163	10.0.2.15	34.160.144.191	TCP	54 42104 → 443 [ACK] Seq=...
27	0.186245	10.0.2.15	34.160.144.191	TLSv1.2	147 Client Key Exchange, Ch...
28	0.186376	34.160.144.191	10.0.2.15	TCP	60 443 → 42104 [ACK] Seq=...
29	0.291615	34.160.144.191	10.0.2.15	TLSv1.2	434 New Session Ticket, Cha...
31	0.292408	10.0.2.15	34.160.144.191	TLSv1.2	153 Application Data
33	0.292668	10.0.2.15	34.160.144.191	TLSv1.2	92 Application Data
34	0.292752	34.160.144.191	10.0.2.15	TCP	60 443 → 42108 [ACK] Seq=...
36	0.297072	34.160.144.191	10.0.2.15	TLSv1.2	434 New Session Ticket, Cha...
37	0.297558	10.0.2.15	34.160.144.191	TLSv1.2	170 Application Data

▶ Frame 33: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)
▶ Ethernet II, Src: PCSSystemtec_2f:87:a7 (08:00:27:2f:87:a7), Dst: 52:55:0a:00:02:02 (52:55:0a:00:02:02)
▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 34.160.144.191
▶ Transmission Control Protocol, Src Port: 42108, Dst Port: 443, Seq: 413, Ack: 3402, Len: 38
▼ Transport Layer Security
▼ TLSv1.2 Record Layer: Application Data Protocol: HyperText Transfer Protocol 2
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 33
Encrypted Application Data: 0000000000000022a2458c2fa5c9587f6fc8cd8383146e3e79e3dea73f6c95e47
[Application Data Protocol: HyperText Transfer Protocol 2]

I dati dell'applicazione sono in formato plaintext o leggibile?

Nel traffico HTTPS i dati dell'applicazione non sono in formato plaintext: risultano cifrati e quindi non leggibili. Wireshark li mostra come Application Data all'interno del record TLS, senza rivelare username, password o altri contenuti sensibili, a differenza di quanto accadeva con l'analisi HTTP.

Domande di Riflessione

1. Quali sono i vantaggi dell'uso di HTTPS invece di HTTP?

HTTPS protegge i dati scambiati tra client e server attraverso la cifratura TLS. Questo garantisce riservatezza (le credenziali e le informazioni sensibili non viaggiano in chiaro), integrità (i dati non possono essere alterati senza essere rilevati) e autenticazione (il certificato digitale del server permette di verificare l'identità del sito). In pratica, rispetto a HTTP, HTTPS difende l'utente da intercettazioni e attacchi man-in-the-middle.

2. Tutti i siti web che usano HTTPS sono considerati affidabili?

No. HTTPS assicura solo che la connessione sia cifrata e che i dati non siano visibili a terzi durante il transito. Non garantisce che il sito sia sicuro o legittimo: anche i siti malevoli possono usare certificati validi e apparire "sicuri" nel browser. Per questo è importante valutare sempre la reputazione e l'affidabilità del sito oltre alla presenza di HTTPS.

Conclusione

L'analisi condotta con tcpdump e Wireshark ha evidenziato la differenza sostanziale tra traffico HTTP e traffico HTTPS. Nel primo caso le informazioni, incluse le credenziali di accesso, risultano visibili in chiaro e facilmente intercettabili, dimostrando l'assenza di qualsiasi protezione. Nel secondo caso, invece, i dati applicativi appaiono come pacchetti TLS cifrati, non interpretabili senza le chiavi di sessione, a conferma dell'efficacia della crittografia nel proteggere la comunicazione. Tuttavia, è importante ricordare che l'uso di HTTPS non implica automaticamente l'affidabilità di un sito web: la crittografia assicura la riservatezza dei dati in transito, ma non garantisce la legittimità del servizio.