**S11/L5**

**INDICE  
  
  
- Funzionalità di Powershell  
- Wireshark: Traffico http/HTTPS  
- Esplorazione di Nmap  
- Attacco a un Database MySQL**

**Funzionalità di Powershell**PowerShell è uno strumento avanzato sviluppato da Microsoft per la gestione e l’automazione di sistemi operativi. Rappresenta un’evoluzione del Prompt dei Comandi, offrendo funzionalità molto più potenti e flessibili. La sua struttura si basa su .NET Framework e .NET Core, permettendo di lavorare non solo su Windows ma anche su macOS e Linux nella sua versione Core.

Mentre il Prompt dei Comandi è pensato per eseguire comandi basilari con un output testuale, PowerShell adotta un approccio orientato agli oggetti. Questo significa che ogni comando, o cmdlet, restituisce oggetti strutturati, semplificando la manipolazione e l’elaborazione dei dati. Grazie a questa caratteristica, PowerShell si presta perfettamente alla creazione di script complessi per automatizzare attività ripetitive, come la gestione di utenti, file e applicazioni, nonché alla configurazione e amministrazione di server.

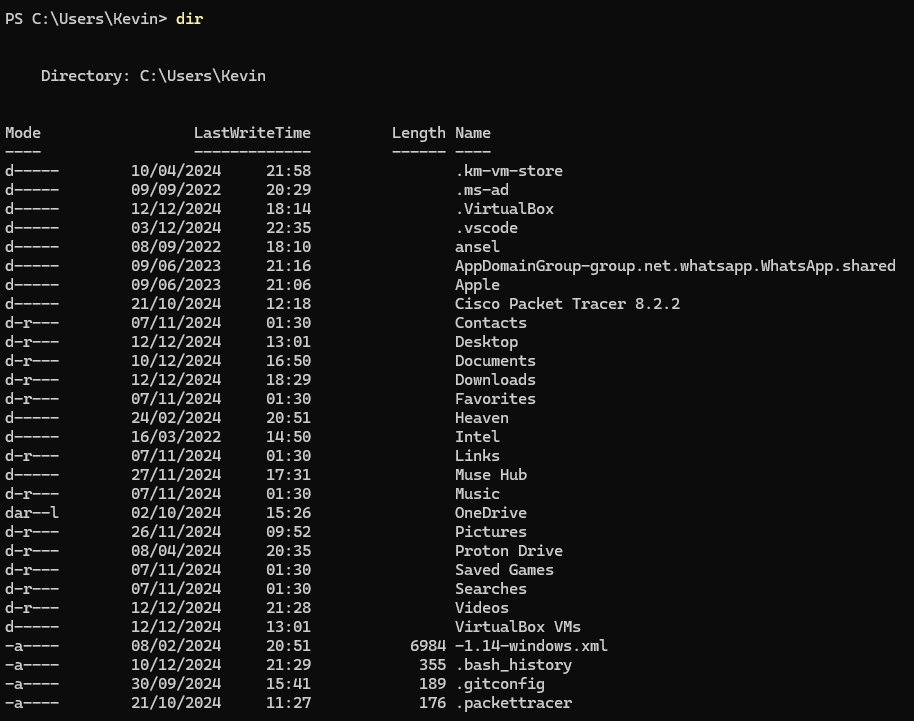
Un’altra differenza fondamentale è la capacità di PowerShell di interagire con API e servizi cloud, come Microsoft Azure, e di lavorare con formati di dati come JSON, XML e CSV in modo nativo. Inoltre, grazie alla modularità del sistema, è possibile estenderne le funzionalità tramite l’aggiunta di nuovi moduli e cmdlet specifici.

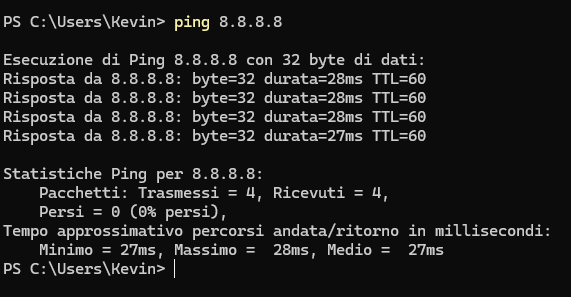
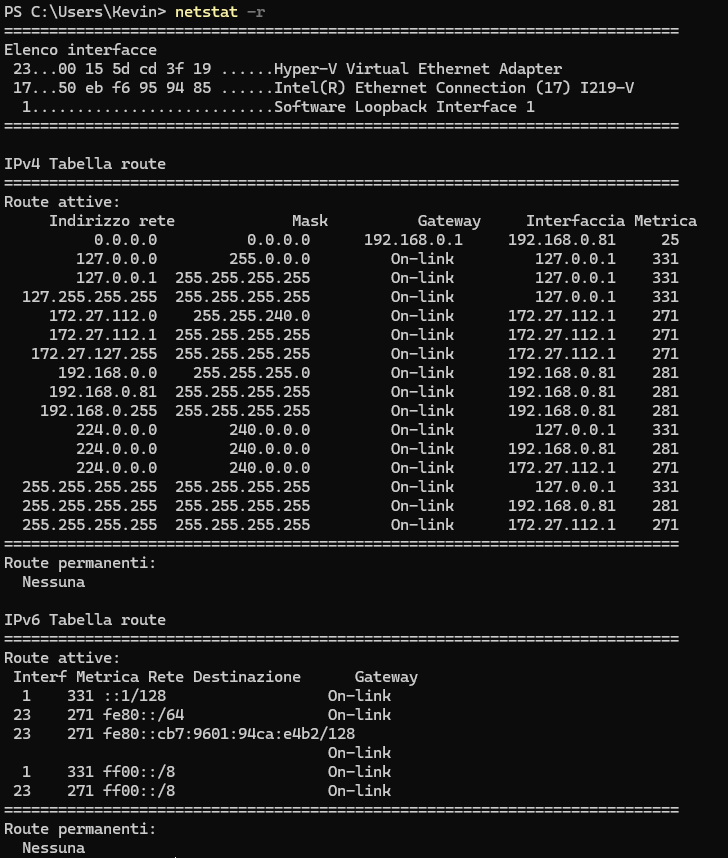
In sintesi, PowerShell non è solo un terminale, ma un ambiente completo per il controllo e l’automazione dei sistemi, indispensabile per professionisti IT e amministratori di rete. Sebbene richieda una curva di apprendimento maggiore rispetto al Prompt dei Comandi, le sue potenzialità lo rendono uno strumento senza pari nel panorama della gestione IT.

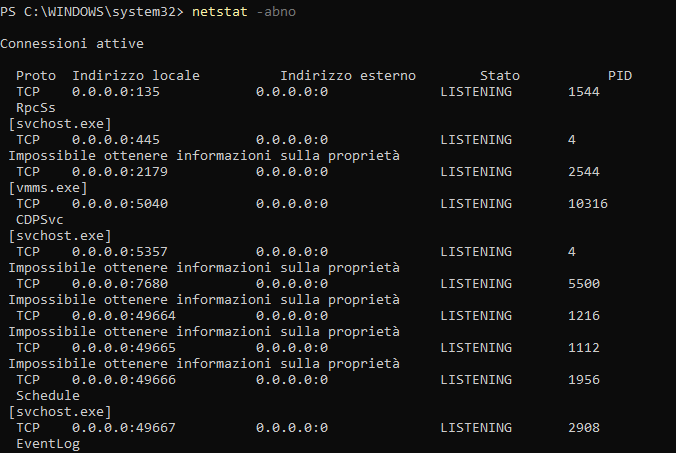
**Alcuni comandi di Powershell**

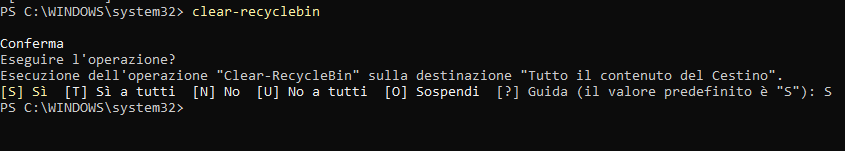
**DIR:** nel Prompt dei Comandi, dir è un comando statico che restituisce un semplice elenco testuale dei file e delle cartelle. In PowerShell, invece, è molto più versatile grazie alla sua natura orientata agli oggetti e al supporto per spazi diversi dal filesystem.  
  
Il comando dir in PowerShell è un alias del cmdlet Get-ChildItem. Questo significa che, anche se deriva dal Prompt dei Comandi, in PowerShell utilizza delle funzionalità avanzate.  
  
**Ping:** Il comando ping su PowerShell è un’utility utilizzata per verificare la connettività di rete tra il computer locale e un dispositivo remoto (come un server, un router o un altro computer). Questo comando invia richieste ICMP al dispositivo specificato, per determinare se è raggiungibile e quanto tempo impiega a rispondere.

**DIR**



**PING**  
  
  
  
  
  
  
  
**Netstat**: Il comando **netstat** su PowerShell è un'utilità di rete che consente di monitorare le connessioni attive sul computer, fornendo informazioni dettagliate su porte, protocolli e stati delle connessioni. Questo strumento è utile per analizzare il traffico di rete, diagnosticare problemi di connettività e identificare eventuali attività sospette.  
  
Con **netstat -r** verrà mostrata la **tabella di instradamento** del sistema. Questa tabella è fondamentale per comprendere come il computer decide il percorso dei pacchetti di rete verso le loro destinazioni.  
  


Il comando **netstat -abno** fornisce un'analisi dettagliata di tutte le connessioni di rete attive sul sistema, includendo informazioni sui processi che le gestiscono, le porte utilizzate e lo stato delle connessioni. Questo comando è particolarmente utile per diagnosticare problemi di rete, monitorare attività sospette e analizzare l'uso delle risorse di rete da parte delle applicazioni.  
  


Il comando **Clear-RecycleBin** in PowerShell viene utilizzato per svuotare il Cestino del sistema operativo, eliminando in modo definitivo tutti i file e le cartelle presenti in esso. È particolarmente utile per automatizzare la gestione dello spazio su disco o per eseguire operazioni di manutenzione.  
  


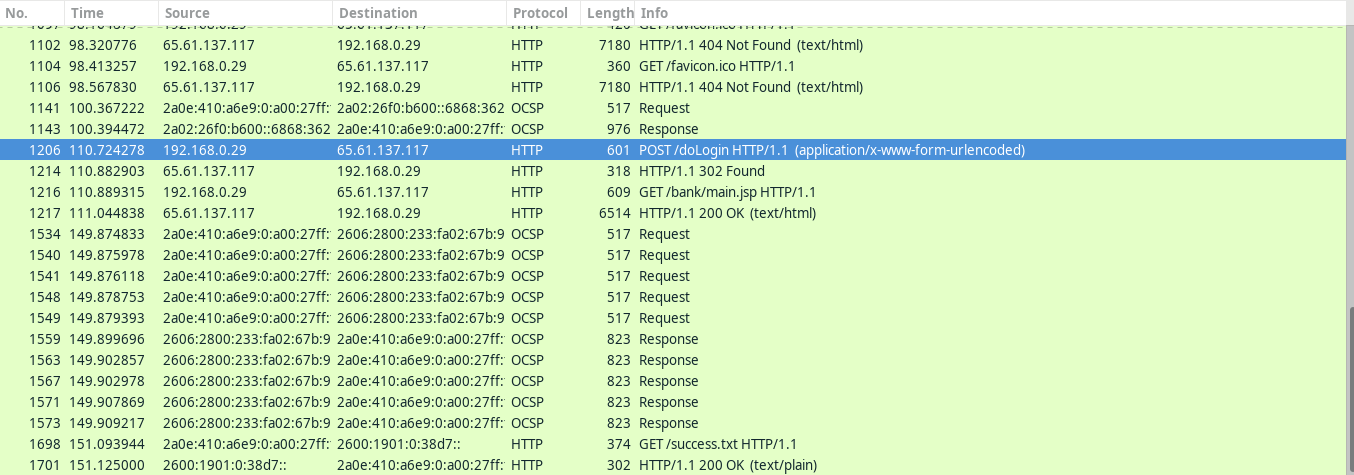
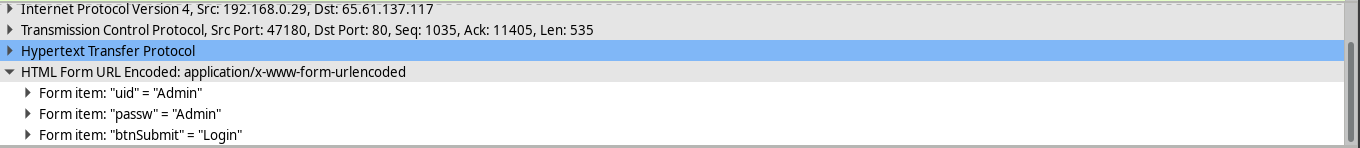
**Wireshark: Traffico HTTP/HTTPS**

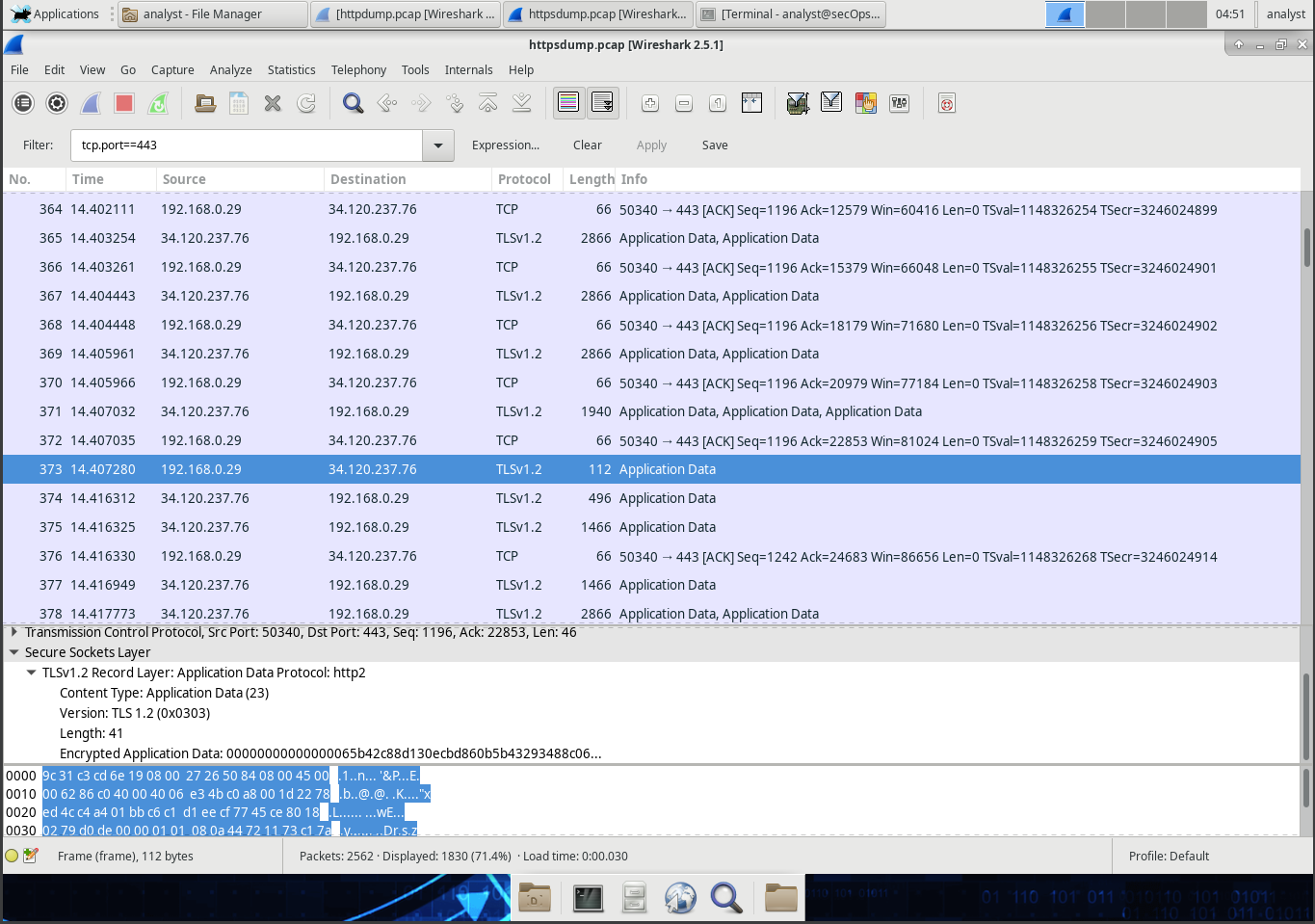
Wireshark è uno strumento di analisi del traffico di rete che permette di catturare e ispezionare i pacchetti che transitano su una rete. È ampiamente utilizzato in ambito IT per diagnosticare problemi di rete, analizzare prestazioni, monitorare la sicurezza e capire meglio il funzionamento dei protocolli di rete. Grazie alla sua capacità di decodificare migliaia di protocolli, Wireshark fornisce una visione dettagliata dei dati in transito, inclusi i pacchetti a livello di applicazione, trasporto e rete.

Quando si tratta di monitorare il traffico HTTP e HTTPS, Wireshark offre diverse funzionalità. Per il traffico HTTP, che è un protocollo di comunicazione non criptato, Wireshark può facilmente intercettare e visualizzare il contenuto delle richieste e risposte HTTP. Durante una cattura, è possibile applicare un filtro come http per concentrarsi esclusivamente sui pacchetti HTTP. In questo modo, si possono analizzare dettagli come gli URL richiesti, le intestazioni HTTP, i metodi (GET, POST, ecc.), i dati inviati e ricevuti, e il comportamento generale della comunicazione tra il client e il server.

Tuttavia, il traffico HTTPS, essendo criptato tramite SSL/TLS, è molto più complesso da analizzare. Wireshark non può decriptare automaticamente il traffico HTTPS, quindi i dati sono visibili solo fino al punto in cui viene stabilita la connessione crittografata. Per poter monitorare il traffico HTTPS in modo dettagliato, è necessario disporre delle chiavi di sessione SSL/TLS o configurare un *man-in-the-middle* proxy (come un proxy SSL) che possa intercettare e decriptare le comunicazioni. In alternativa, Wireshark può mostrare metadati relativi alla connessione, come gli handshake SSL/TLS, ma il contenuto effettivo della comunicazione rimarrà criptato senza le informazioni necessarie per decriptarlo.

In generale, Wireshark è uno strumento potente per analizzare il traffico di rete, ma per HTTPS richiede configurazioni aggiuntive per ottenere una visibilità completa sulle comunicazioni sicure.

**Esempio di traffico HTTP  
  
**Provando ad effettuare il log-in su un sito che utilizza solamente il protocollo http sarà possibile intercettare le credenziali d’accesso tramite il monitoraggio del traffico  
  


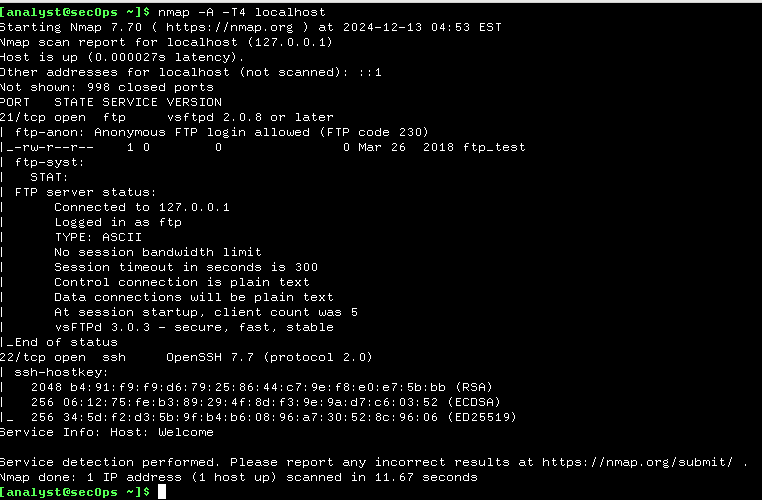
Esempio di traffico HTTPS  
  


È possibile notare la differenza con il sito web in http. Anche se Wireshark è in grado di monitorare i movimenti i dati saranno criptati, dunque sarà impossibile vedere le credenziali in chiaro.

**Esplorazione di Nmap**Nmap (Network Mapper) è uno strumento di scansione delle reti utilizzato per esplorare e analizzare le reti di computer. È particolarmente utile per amministratori di sistema e professionisti della sicurezza, poiché consente di ottenere informazioni dettagliate su una rete, come quali dispositivi sono connessi, quali porte sono aperte su questi dispositivi e quali servizi o applicazioni stanno ascoltando su quelle porte. Nmap è ampiamente utilizzato per scoprire vulnerabilità nelle reti e per mappare l'infrastruttura di rete, sia in ambienti aziendali che in situazioni di test di penetrazione.

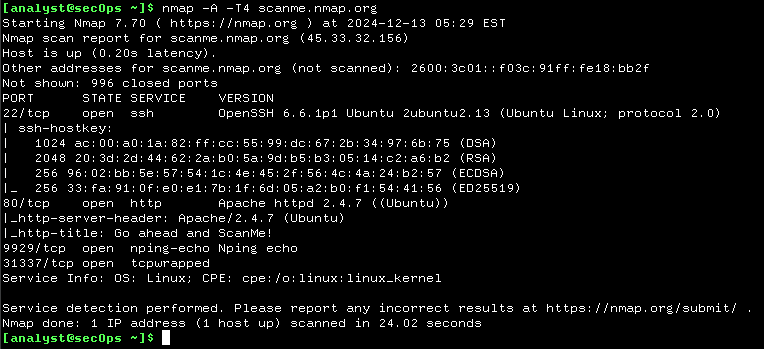
Nmap è in grado di rilevare una varietà di informazioni, tra cui l'indirizzo IP di un dispositivo, il sistema operativo in uso, i firewall o altri dispositivi di protezione presenti e le applicazioni in esecuzione su un determinato host. Viene spesso utilizzato per fare una scansione di rete in modo da determinare quali dispositivi sono attivi, quali porte sono aperte e se ci sono potenziali vulnerabilità che potrebbero essere sfruttate da un attaccante.

Il software offre diverse opzioni avanzate, come la scansione delle versioni dei servizi, il rilevamento del sistema operativo e la possibilità di eseguire scansioni stealth, che nascondono l'origine della scansione, rendendo più difficile rilevare l'attività. Nmap è anche uno strumento di audit di sicurezza essenziale, poiché consente di identificare porte aperte che potrebbero essere utilizzate per attacchi, monitorare l'integrità della rete e gestire meglio la configurazione della rete per proteggere i dispositivi da accessi non autorizzati.

**Esempi di comandi Nmap**

Il parametro -A attiva una serie di funzionalità avanzate di Nmap che includono:

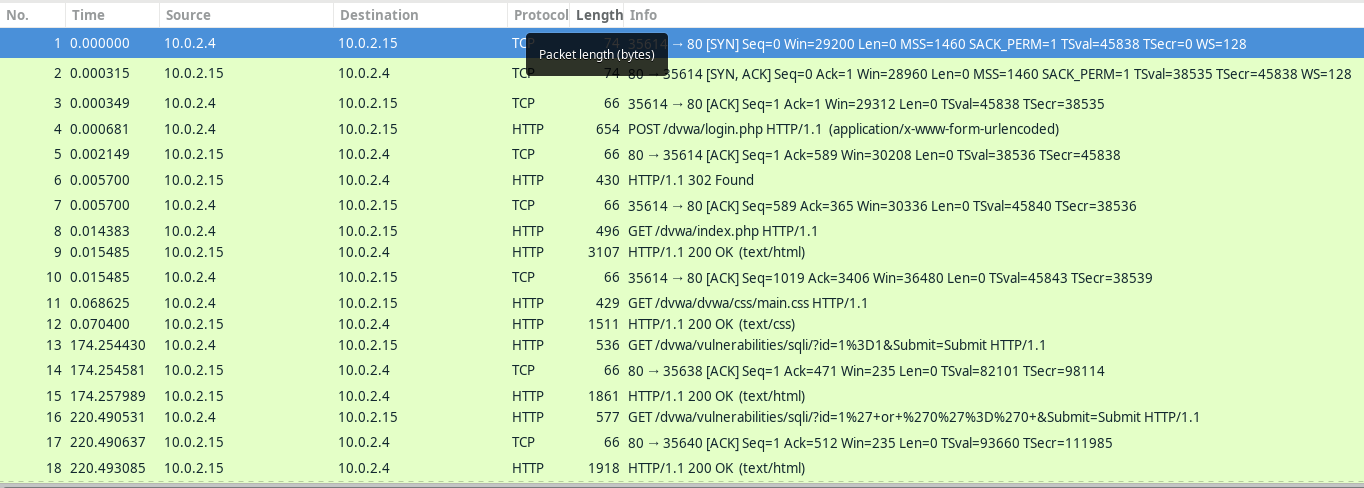
* Rilevamento del sistema operativo: Nmap tenta di identificare il sistema operativo in uso sul dispositivo di destinazione, analizzando le risposte alle richieste di rete e i comportamenti specifici del sistema.
* Rilevamento dei servizi e delle versioni: Oltre a scansionare le porte aperte, Nmap prova a determinare quale servizio è in esecuzione su ciascuna porta (ad esempio, Apache, MySQL, FTP, ecc.), e anche quale versione di quel servizio è presente.
* Rilevamento degli script Nmap: Nmap esegue anche script di rilevamento di vulnerabilità e di informazioni, che possono essere usati per testare la sicurezza del sistema e identificare vulnerabilità comuni.  
    
  L'opzione **-T4** definisce la velocità e l'intensità con cui la scansione viene eseguita. Nmap offre 5 livelli di timing, numerati da **T0** a **T5**, dove:
* **T0** è il più lento e discreto, pensato per evitare di essere rilevato da sistemi di sicurezza.
* **T5** è il più veloce, ma può risultare più facilmente rilevabile dai sistemi di difesa.
* L'opzione **-T4** è una scelta equilibrata, che consente di eseguire una scansione più rapida rispetto a **T3** (il livello predefinito) senza compromettere troppo la precisione. È spesso utilizzato quando si desidera completare la scansione in tempi relativamente brevi, mantenendo un buon livello di affidabilità. Utilizzare **-T4** è ideale in ambienti dove la discrezione non è una priorità assoluta e si desidera ottenere risultati rapidamente.

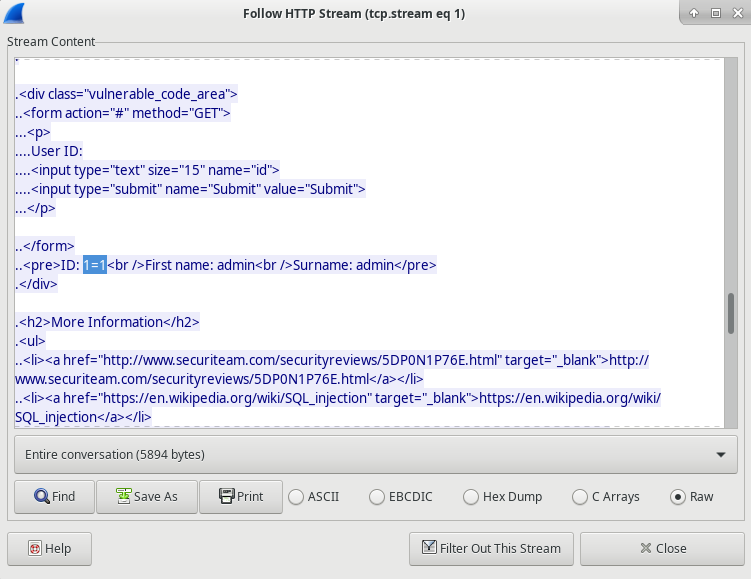
**Esempio di scansione  
  
**Utilizzando i comandi precedentemente descritti si può notare la presenza di due protocolli aperti sul sito scanme.nmap.org  
  
La porta 22 (ssh) utilizzata per il controllo remoto di dispositivi in chiave cifrata e la porta 80 (http) utilizzata per la navigazione sul web.

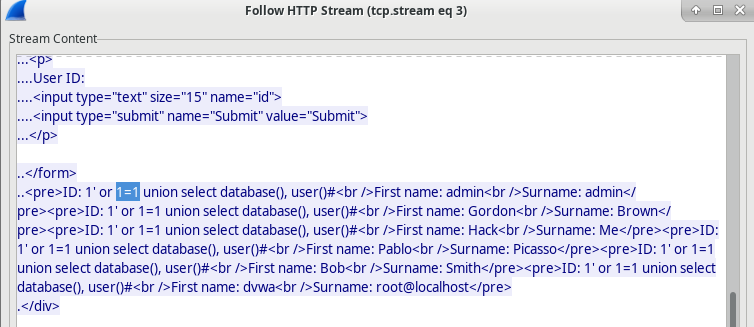
**Attacco a un Database MySQL**  
  
Wireshark è uno strumento potente per l'analisi del traffico di rete, che può essere utilizzato per monitorare vari tipi di attacchi, inclusi gli **attacchi SQL injection**. Un attacco SQL injection si verifica quando un attaccante inserisce comandi SQL maligni in un'applicazione web che interagisce con un database, con l'intento di manipolare o estrarre dati sensibili. Wireshark può essere utilizzato per identificare e analizzare queste vulnerabilità attraverso l'ispezione dei pacchetti di rete che viaggiano tra client e server.

Wireshark consente di catturare e analizzare il traffico HTTP, in cui si possono trovare le richieste contenenti query SQL. Gli attacchi SQL injection di solito utilizzano richieste HTTP, come quelle **GET** o **POST**, per inviare dati maligni al server web.

Con l'uso di filtri, come http o tcp.port==80, è possibile concentrarsi solo sul traffico HTTP, facilitando l'individuazione di query SQL potenzialmente manipolate. Wireshark permette di seguire lo **stream HTTP**, visualizzando il flusso di dati tra client e server, e consentendo di analizzare eventuali query SQL inviate. Se un attaccante invia un input come 1=1 in un campo di ricerca, questo potrebbe generare una query SQL che restituisce informazioni sensibili, come il nome del database o gli account utente, rendendo evidente una vulnerabilità.

Infine, Wireshark può essere utilizzato per monitorare la risposta del server, aiutando a determinare se l'applicazione è vulnerabile alla **SQL injection**, ad esempio se restituisce dati non autorizzati anziché un messaggio di errore. Attraverso questa analisi, è possibile rilevare attacchi in corso e adottare le misure correttive necessarie.  
  
  
**Esempio di SQL Injection  
  
**

L'immagine rappresenta una cattura di traffico di rete in cui avviene una sequenza di scambio dati tra due macchine, presumibilmente un client (10.0.2.4) e un server (10.0.2.15). Questa analisi evidenzia una tipica interazione web basata su HTTP e TCP, che include autenticazione e richieste verso risorse web.  
La sequenza mostra che potrebbe essere associata a un'attività di test su un'applicazione vulnerabile (ad esempio DVWA). Il client sembra autenticarsi e successivamente testare la sicurezza tramite richieste mirate (ad esempio una potenziale SQL injection). Questo tipo di traffico è tipico di un ambiente di testing controllato.  
  
Si può notare che dopo l'autenticazione, il client richiede altre risorse web, inclusi file statici come css/main.css e contenuti dinamici. Una delle richieste si riferisce a una vulnerabilità SQL con parametri come id=1%27D1. Questo potrebbe essere parte di un test di penetrazione o un attacco SQL injection simulato.  
  
  
Possiamo notare nella riga 11 del traffico catturato è presente un tentativo di query malevola: 1=1 comunemente utilizzata negli attacchi di **SQL Injection**. Può essere sfruttata per manipolare una query SQL e ottenere un comportamento non previsto dall'applicazione.   
  




La riga 19 rivela il risultato di un attacco SQL Injection. L'attaccante ha sfruttato questa vulnerabilità inserendo un payload SQL, nello specifico 1' or 1=1 union select database(), user()#, che consente di eseguire comandi non autorizzati sul database.

Il risultato di questa iniezione restituisce informazioni sensibili, come i nomi e cognomi di utenti, il nome del database attualmente in uso ("dvwa") e l'utente del database ("root@localhost"). Questo scenario evidenzia chiaramente una vulnerabilità che consente l'accesso non autorizzato a dati riservati attraverso un errore di validazione dell'input.  
  
  
È possibile notare la ripetizione di queste query malevole in svariate altre righe monitorate precendetemente da Wireshark come la 22, 25 e la riga 28.  
  
  
**Conclusione**  
  
Wireshark è uno strumento fondamentale per il monitoraggio della rete perché permette di analizzare il traffico in tempo reale, offrendo una visione dettagliata di ogni pacchetto trasmesso. Questa capacità è essenziale per identificare vulnerabilità, diagnosticare problemi di rete e rilevare attività sospette, come attacchi informatici o fughe di dati. Inoltre, consente agli amministratori e agli esperti di sicurezza di comprendere a fondo i comportamenti della rete, individuare anomalie e verificare il corretto funzionamento di applicazioni e protocolli. Grazie alla sua interfaccia intuitiva e alle potenti funzioni di filtraggio e decodifica, Wireshark è uno strumento imprescindibile per chiunque voglia garantire la sicurezza e l'efficienza delle proprie infrastrutture di rete.