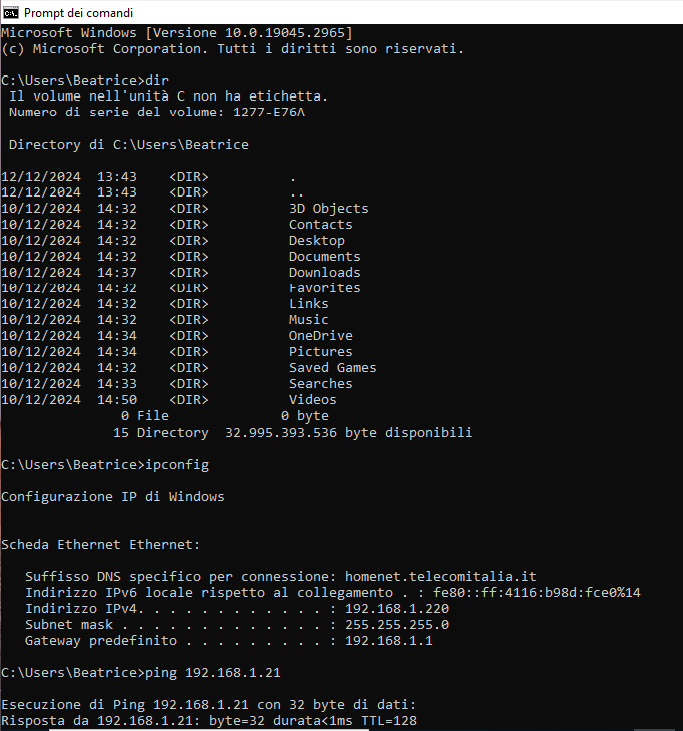
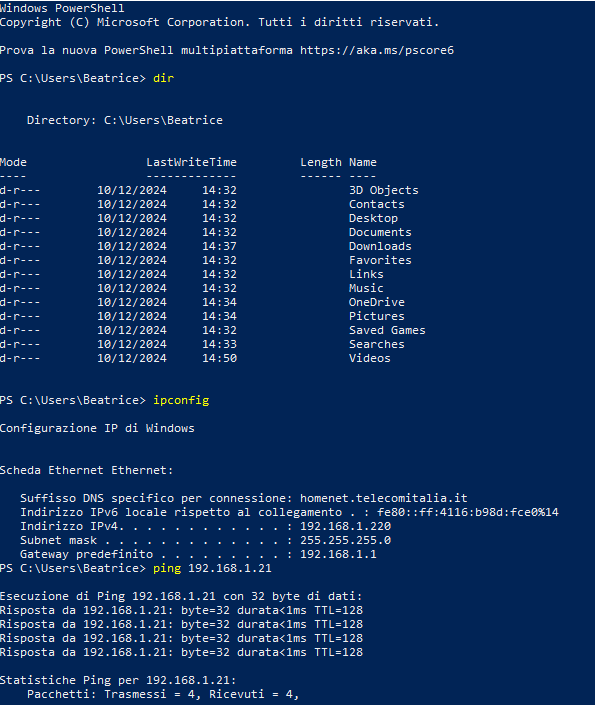
PROGETTO S11E5

Il progetto di oggi comprende più parti separate tra loro, utili per acquisire dimestichezza con i vari programmi che andremo a sfruttare.

**Parte1:** utilizzo di Windows PowerShell

**Windows PowerShell** è uno strumento che permette di **accedere a Windows tramite riga di comando** ma con molte funzionalità avanzate che lo rendono uno strumento potente per amministratori di sistema e utenti avanzati.

Una volta aperto il semplice Prompt dei comandi e il nostro PowerShell andiamo ad inserire il comando dir in entrambe le finestre e possiamo notare come l’elenco delle sottodirectory e dei file è molto simile; entrambi mostrano la data e l’ora dell’ultima modifica, ma in PowerShell possiamo visionare anche i permessi.

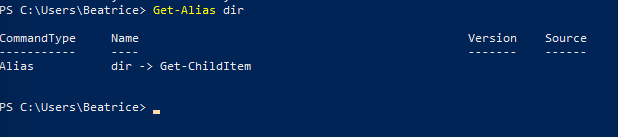


Una delle differenze comuni tra PowerShell e un semplice Prompt dei comandi sono le **cmdlet**: la struttura base dei comandi di PowerShell e seguono una struttura standard formata da Verbo-Nome (verbo: cosa vuoi fare, nome: dove vuoi agire).

La differenza con i comandi normali è che le cmdlet lavorano anche con oggetti; mentre i comandi tradizionali restituiscono solo testo (se si scrive ps quello che uscirà in output saranno tutti i processi attivi, ma per manipolarli servirà usare comandi specifici) in PowerShell invece, con le cmdlet se si scrive il comando get-process quello che uscirà in output saranno tutte le informazioni utili su quel processo.

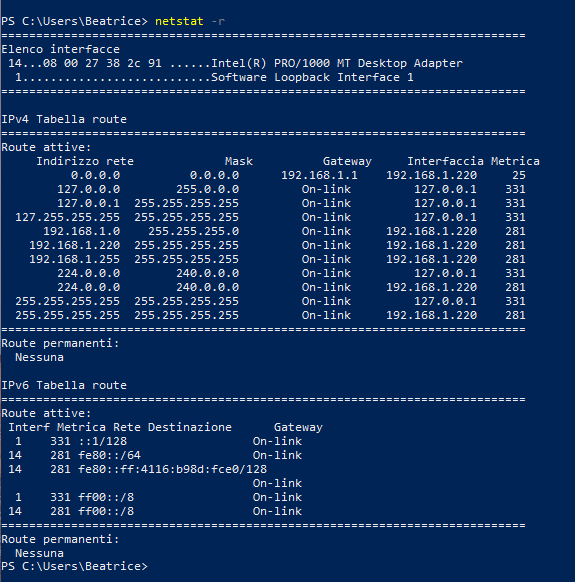
Un'altra caratteristica che rende PowerShell uno strumento avanzato e dal semplice utilizzo, sono gli **alias:** un alias in powershell è come un soprannome o un'abbreviazione per comandi lunghi e complessi che quindi è utile per memorizzarli più facilmente. È molto utile anche perché permette di creare alias personali nel caso in cui c'è un comando che si usa molto spesso.

Sono utili perché veloci, semplici da ricordare e personalizzabili.



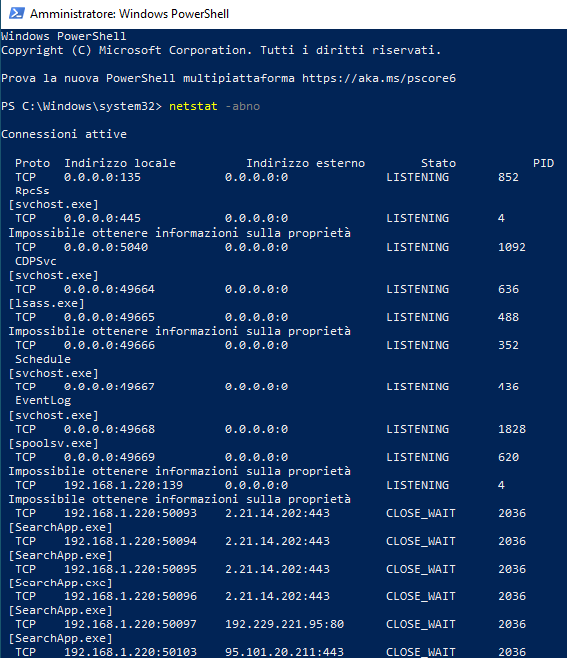
Con il comando **netstat** possiamo **visualizzare le connessioni di rete attive** e altre informazioni relative alla rete sul sistema.

Abbiamo usato il comando netstat –r per visualizzare la tabella di routing

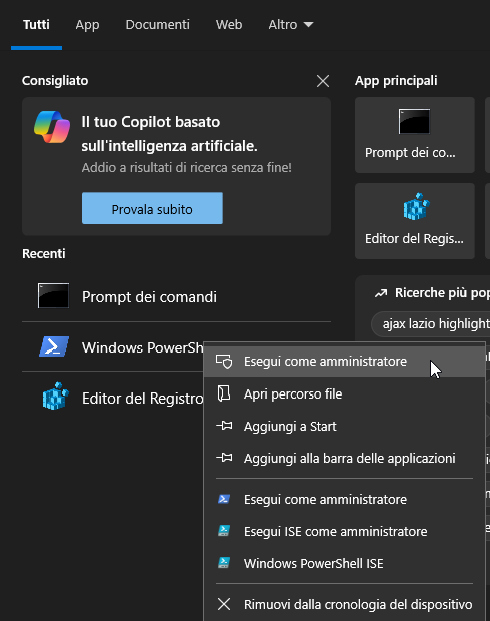


Con la cmdlet Get-Alias dir possiamo vedere che **dir** è l'alias della cmdlet completa Get-Childitem usata per elencare file e cartelle.

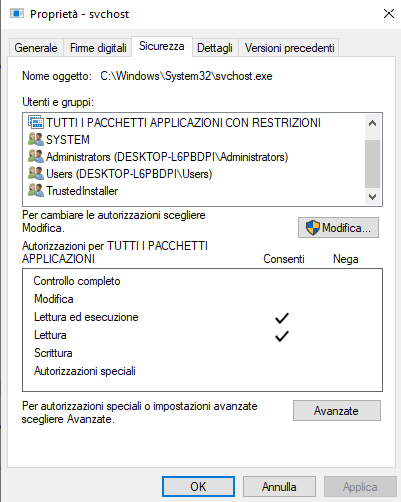
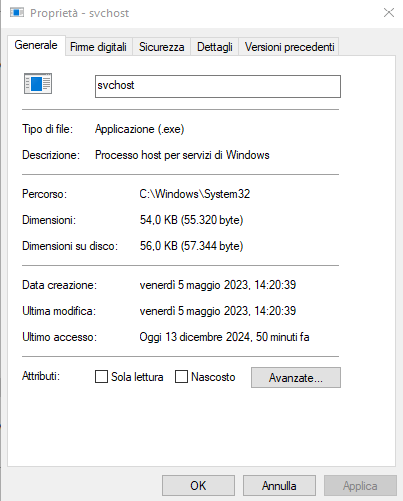
il comando netstat -abno mostra tutte le connessioni di rete attive sul sistema; lo stato di tale connessione; gli indirizzi ip e le porte utilizzate per la connessione; il programma che sta usando quella connessione di rete e il pid del processo che sta usando la connessione



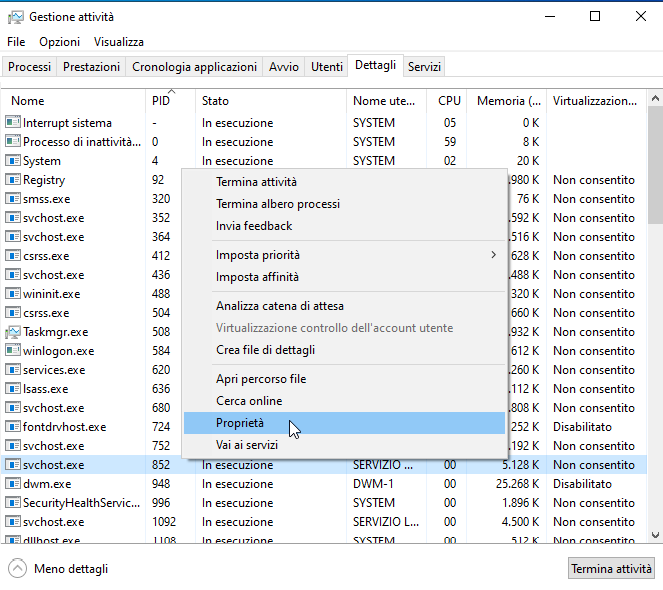
Successivamente abbiamo aperto un’altra finestra PowerShell come amministratori per usare il comando netstat –abno.



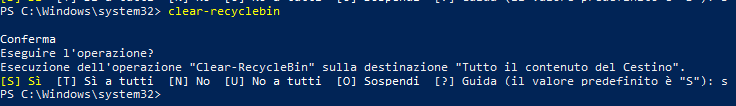
Come i permessi di quel processo, l’utente, quanto spazio di memoria sta utilizzando ecc



Siamo entrati poi nel Task Manager ed abbiamo cercato un PID presente nei risultati del comando precedente (in questo caso l’852) e siamo andati sulle proprietà per ottenere maggiori informazioni



Infine abbiamo visto come con il comando clear-recyclebin possiamo svuotare il cestino da riga di comando



**Parte 2:** Utilizzo di Wireshark per esaminare il Traffico HTTP e HTTPS

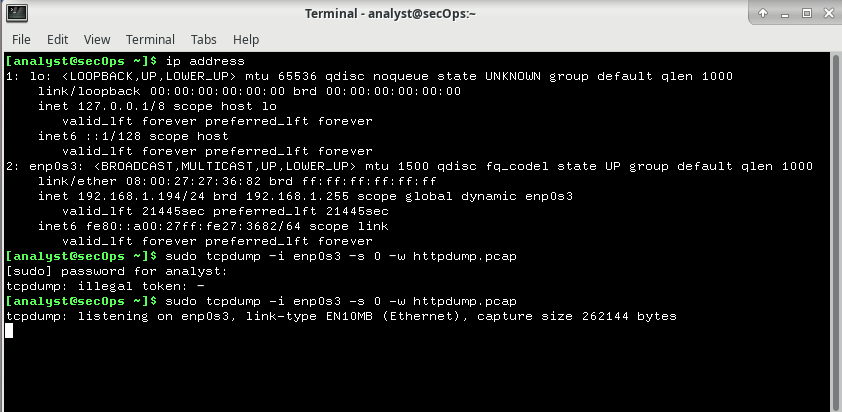
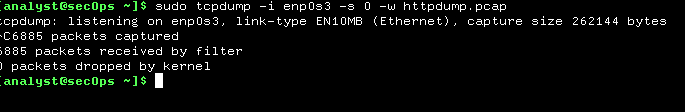
**Wireshark** è uno strumento di monitoraggio e **analisi del traffico di rete** utilizzato per catturare e ispezionare i pacchetti di dati che viaggiano su una rete. Come sappiamo la differenza sostanziale tra il protocollo HTTP e l’HTTPS è l’uso della crittografia di quest’ultimo che rende la trasmissione dei dati più sicura.

In questo esercizio lo useremo per analizzare pacchetti di comunicazione HTTP sulla macchina virtuale Cyberops Workstation e traffico HTTPS sulla macchina Kali Linux.

Useremo tcpdump per catturare il contenuto del traffico e lo salveremo in un file di cattura pacchetti per analizzarlo in seguito con Wireshark.

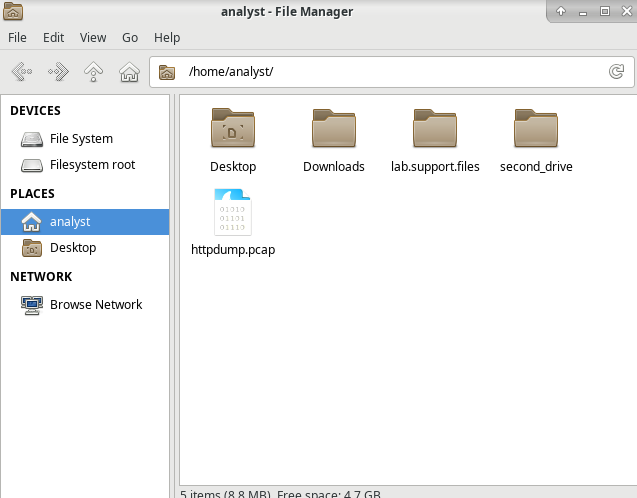
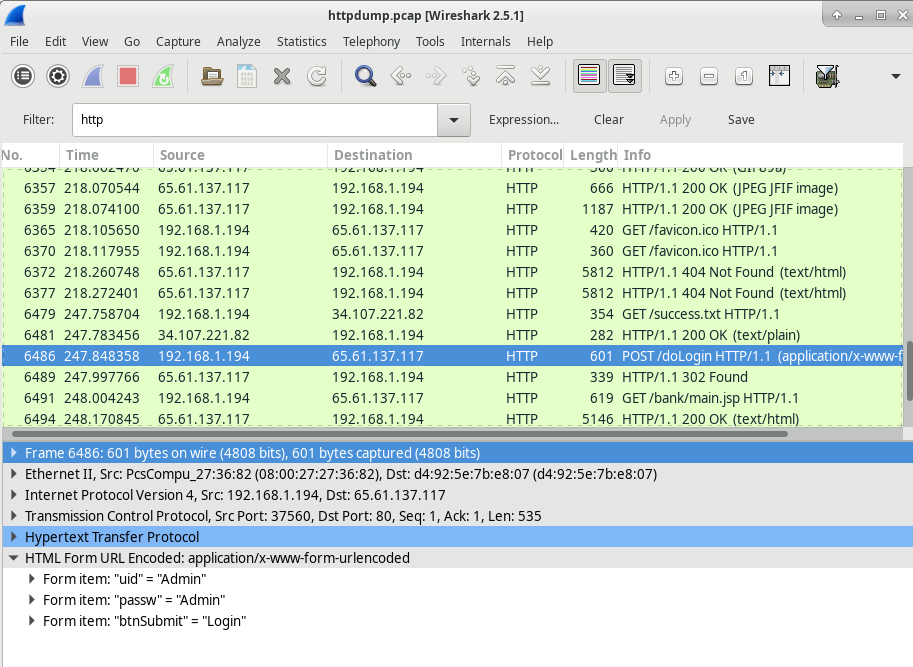
Apriamo il terminale dalla nostra prima macchina e con il comando ip address andiamo a visualizzare l’interfaccia di rete di cui cattureremo il traffico; successivamente con il comando sudo tcpdump –i enp0s3 –s 0 –w httpdump.pcap avviamo tcpdump e registriamo il traffico

Siamo entrati poi su un sito che utilizza il protocollo HTTP, abbiamo fatto l’accesso e abbiamo stoppato la nostra cattura su tcpdump

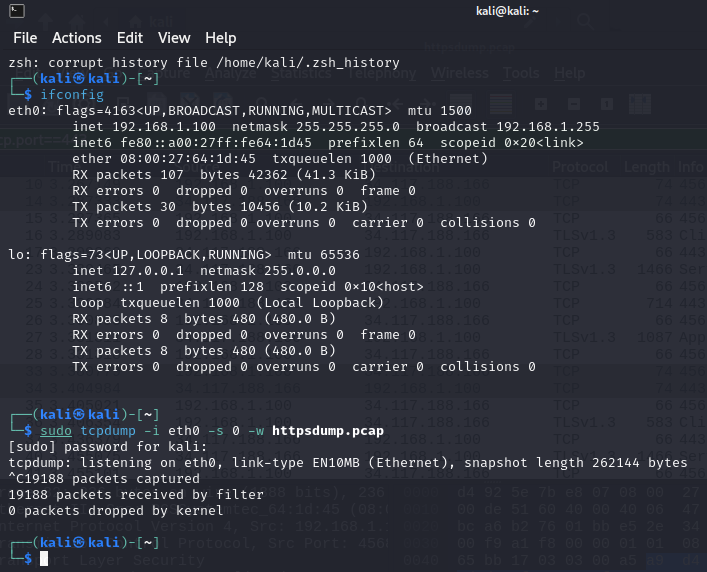
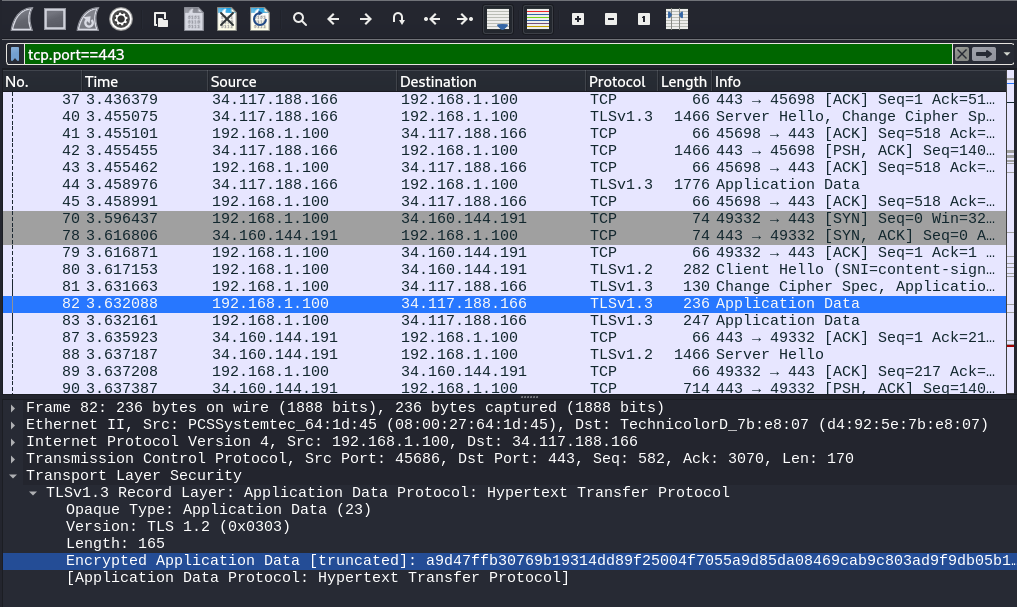


Filtriamo per visualizzare solo i pacchetti HTTP e analizzando la richiesta POST, vedremo che aprendo la sezione **HTML Form URL Encoded: application/x-www-form-urlencoded** ci appariranno in output le credenziali in chiaro che abbiamo usato per accedere al sito.

Apriamo il nostro file in cui è stata salvata la cattura fatta con tcpdump, su Wireshark per analizzarne il contenuto

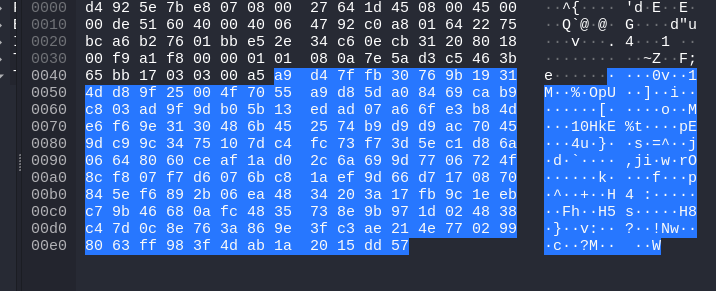


La stessa cosa l’abbiamo fatta su Kali analizzando il traffico tra noi ed un sito che utilizza un protocollo HTTPS che è quindi crittografato



Abbiamo aperto tcpdump e abbiamo catturato il traffico sull’interfaccia di rete eth0

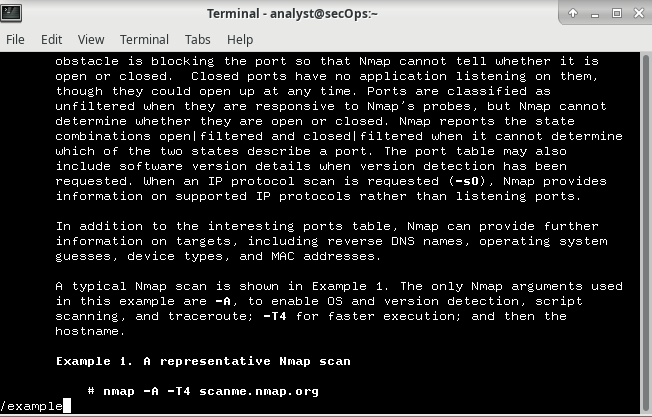
Abbiamo aperto il nostro file con Wireshark e analizzando il pacchetto contenente Dati applicazione notiamo che appare una sezione diversa da prima che non ci permette di visualizzare le credenziali in chiaro usate per accedere al sito, ma vedremo questi dati crittografati e quindi non leggibili



**Parte 3:** Esplorazione di Nmap

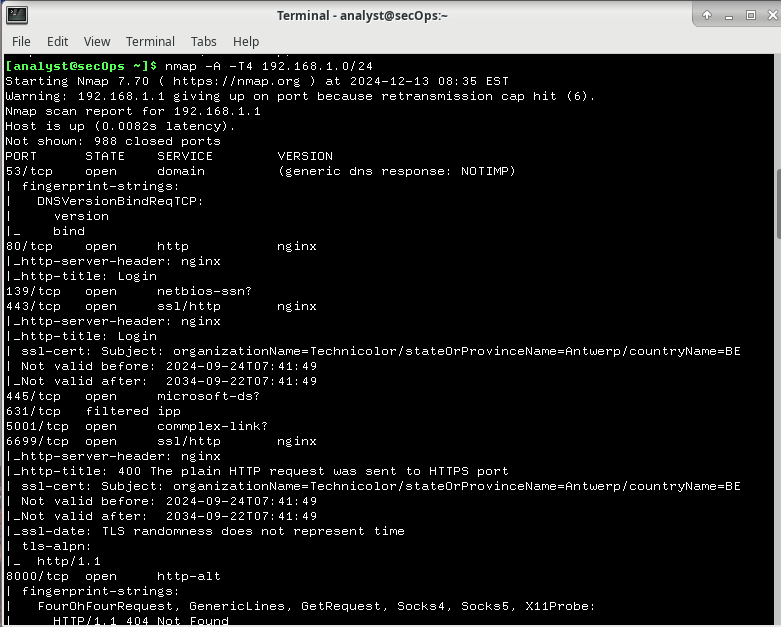
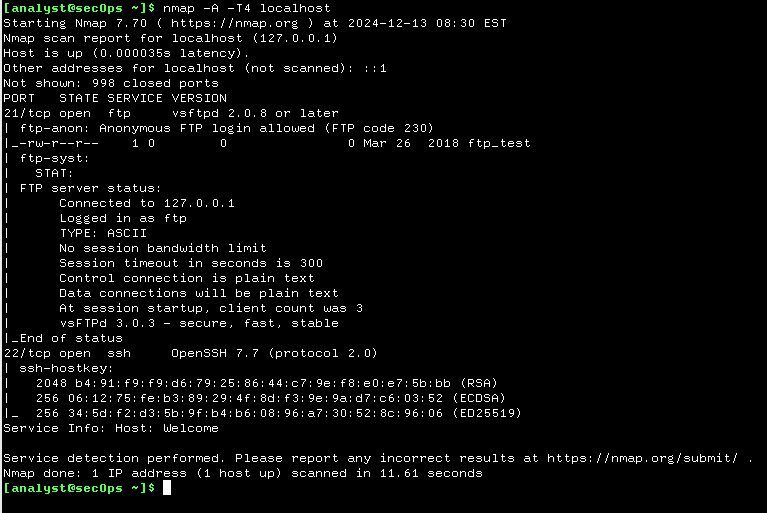
**Nmap** è uno strumento fondamentale per analizzare la sicurezza e la configurazione di una rete; alcune delle sue funzionalità sono quelle di scansione delle porte; rilevamento dei dispositivi presenti su una rete e individuazione di eventuali vulnerabilità

Dopodiché siamo passati alla scansione delle porte; prima sul nostro localhost notando che le porte attive sono la 21 su cui gira il protocollo FTP e la 22 su cui gira il protocollo SSH



La prima cosa che abbiamo fatto dopo aver aperto la nostra macchina virtuale è stata quella di eseguire il comando man nmap da terminale.

Il comando man è usato su sistemi Linux per visualizzare le pagine di manuale relative ad un programma o ad un comando, in questo caso relative ad nmap.



La stessa cosa l’abbiamo fatta per scannerizzare la rete di cui fa parte la nostra macchina, abbiamo quindi inserito l’IP di network per visualizzare tutti gli host connessi alla nostra rete e le porte aperte per ogni dispositivo connesso

Da questa scansione di rete abbiamo visto che gli host connessi alla nostra rete sono 4:

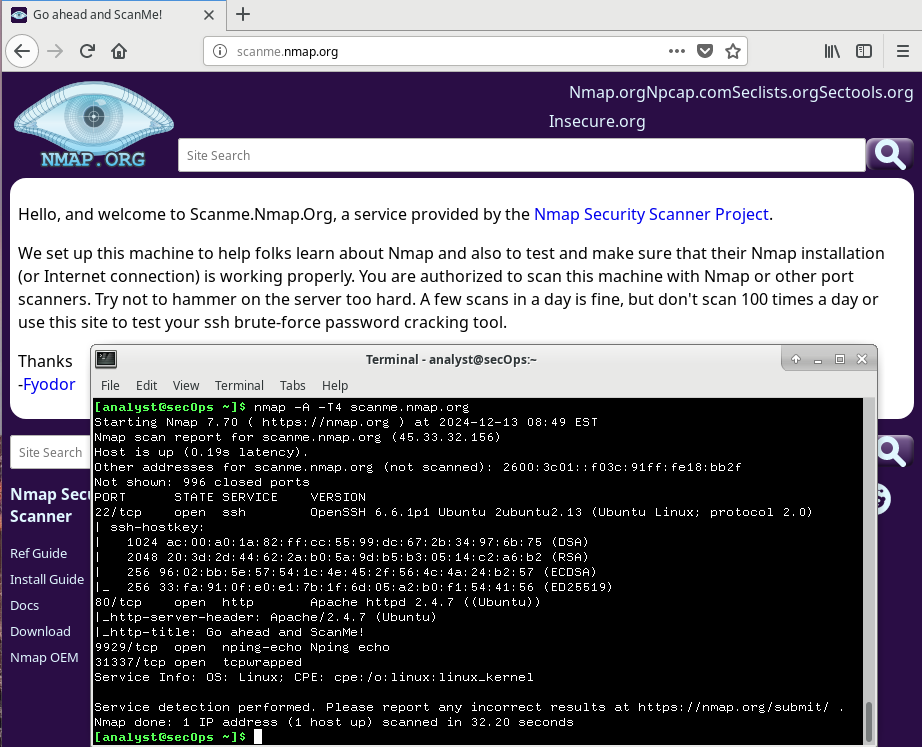
il **192.168.1.1** usato per il gateway dove sono aperte la porta diverse porte tra cui la 53 su cui gira il DNS; la porta 80 per l’http; la 443 per l’HTTPS; la 139 per il netbios

il **192.168.1.9** su cui non ci sono porte aperte

il **192.168.1.176** su cui è aperta la porta 8080 per il servizio http

e il **192.168.1.194** ovvero la nostra macchina su cui sono aperte la porta 21 per il protocollo FTP e la 22 per l’SSH

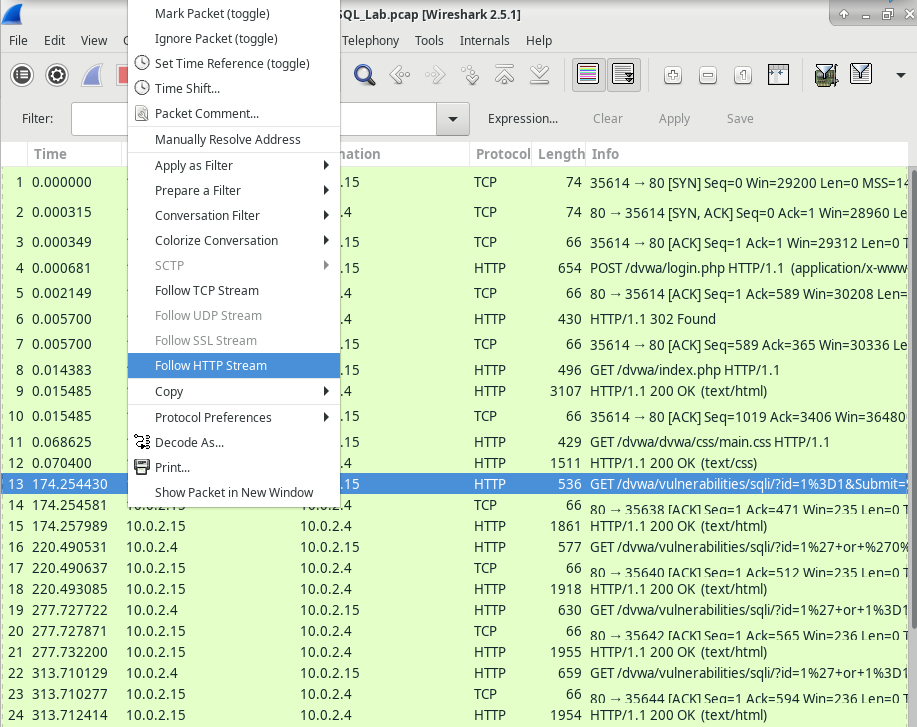
La terza e ultima scansione l’abbiamo fatta su un sito web che consente agli utenti di scansionarlo per fare pratica e abbiamo visto che le porte note aperte per questo sito sono la 22 per l’SSH e l’80 per l’HTTP



**Parte 4:** visualizzare un file PCAP relativo ad un attacco contro un database SQL

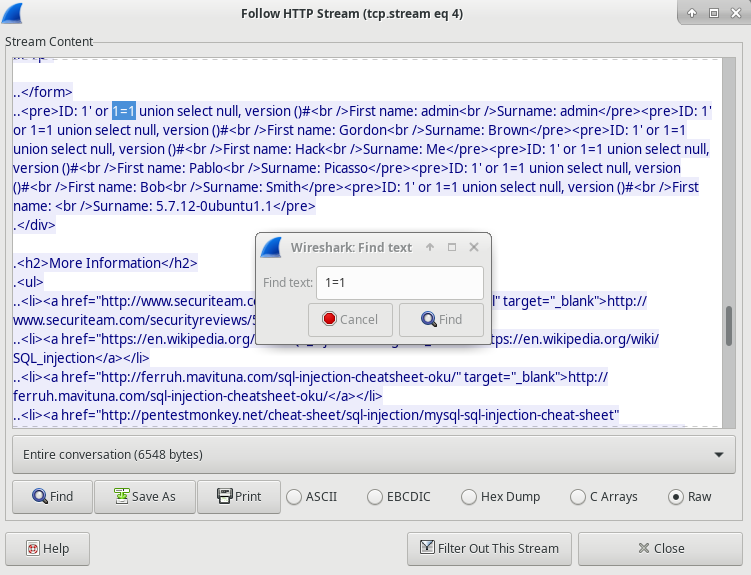
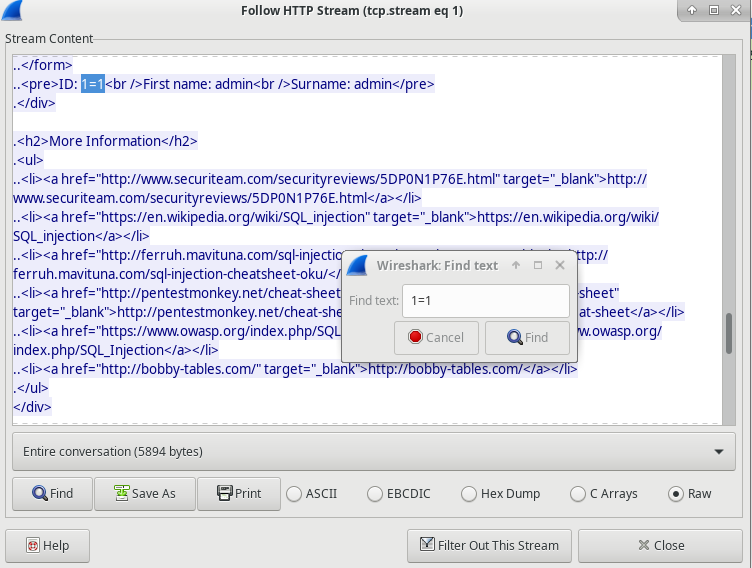
Anche per questa ultima parte abbiamo analizzato un attacco SQL injection usando Wireshark sulla nostra macchina Cyberops Workstation.

Dopo aver aperto il file PCAP che si trovava all’interno della macchina e che conteneva la cattura di un traffico dati durante un attacco SQL injection sui database della DVWA, siamo andati ad analizzare le mosse dell’attaccante passo passo

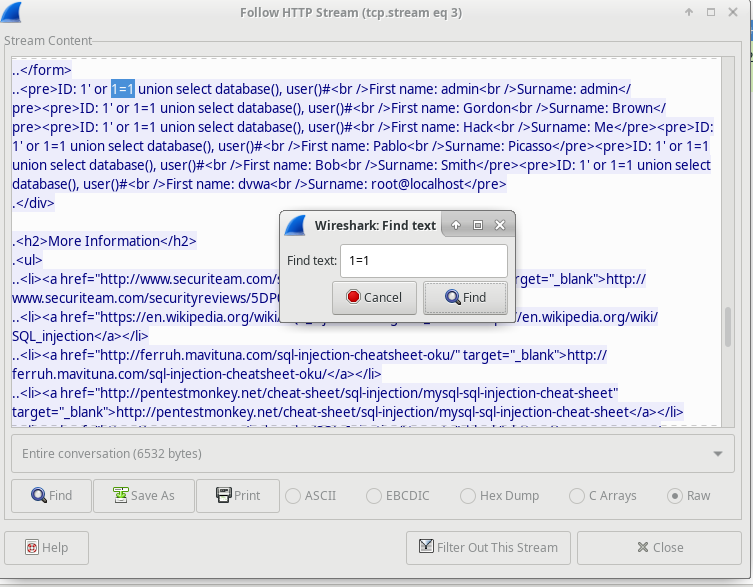


Siamo andati ad analizzare tutte le richieste HTTP GET che ci permetteranno di seguire il flusso dei dati per capire al meglio le query usate. Abbiamo quindi aperto l’HTTP stream della riga 13

Andiamo ad analizzare il traffico di sorgente ed inseriamo 1=1 nel campo find per cercare una corrispondenza e vediamo che l’attaccante ha inserito la query nella barra di ricerca e il sito invece di mandare in output il messaggio di errore ha salvato la query. L’attaccante così si è accertato che il sito è vulnerabile e che non filtra l’input utente; può quindi avviare l’SQL injection.



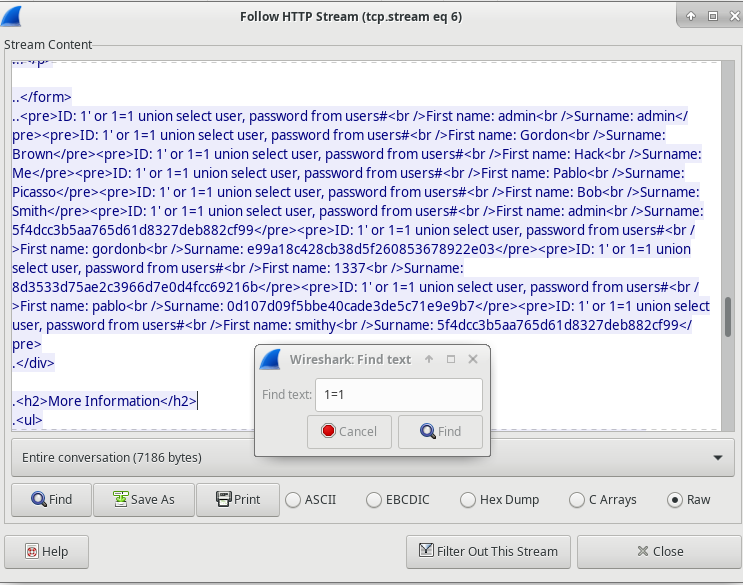
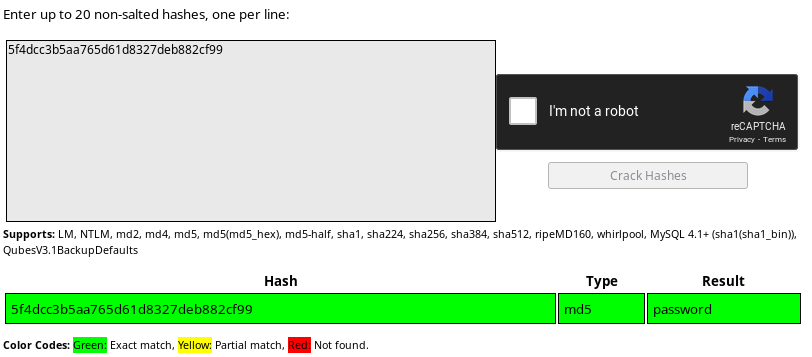
Continuiamo a seguire il flusso analizzando il secondo pacchetto http GET ovvero la riga 19, inseriamo nuovamente il testo 1=1 e vediamo che questa volta l’utente ha inserito la query 1' o 1=1 union select database(), user()# e anche questa volta il sito non ha dato in output errore permettendo all’attaccante di entrare a conoscenza degli account utente registrati in quel database



Andiamo sulla riga 22, inseriamo nuovamente 1=1 e notiamo che l’attaccante ha inserito una nuova query per visualizzare l’identificativo della versione; quindi più si va avanti più l’attaccante entra in possesso di informazioni che può sfruttare a suo favore

Quello che siamo andati a fare poi è usare l’hash ricavato per entrare in possesso della psw in chiaro di un utente usando il sito di cracking <https://crackstation.net/> e abbiamo trovato la password dell’utente admin che in chiaro è password.

Analizziamo l’ultima riga ovvero la 28 in cui reinseriamo nuovamente 1=1 e notiamo che l’attaccante ha inserito la query 1'or 1=1 union select user, password from users# entrando in possesso di tutti i codici hash degli utenti registrati in quel database. Finisce così l’SQL injection



Analizziamo la riga 25 e inseriamo questa volta la parola users e notiamo una quarta query che permette all’attaccante di entrare in possesso di tutte le tabelle nel database

