Report esercizio

Argomento: Attacchi DoS (Denial of Service) - Simulazione di un UDP Flood

Requisiti del Programma:

Input dell'IP Target:

Il programma deve richiedere all'utente di inserire l'IP della macchina target.

Input della Porta Target:

Il programma deve richiedere all'utente di inserire la porta UDP della macchina target.

Costruzione del Pacchetto:

La grandezza dei pacchetti da inviare deve essere di 1 KB per pacchetto.

Suggerimento: per costruire il pacchetto da 1 KB, potete utilizzare il modulo random per la generazione di byte casuali.

Numero di Pacchetti da Inviare:

Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare.

Flusso Operativo del codice

L'utente specifica:

IP del target: Indirizzo IPv4 della macchina bersaglio.

Porta UDP: Numero di porta compreso tra 0 e 65535.

Numero di pacchetti: Quantità totale di pacchetti UDP da inviare.

Lo script:

Crea un socket UDP utilizzando socket.AF_INET e socket.SOCK_DGRAM.

Genera pacchetti di 1 KB pieni di dati casuali tramite random._urandom().

Invia ogni pacchetto utilizzando sendto().

Stampa il numero del pacchetto inviato per il monitoraggio.

Gestione degli Errori:

Se si verifica un errore durante la creazione del socket o l'invio dei pacchetti, lo script stampa un messaggio di errore specifico.

Alla fine, il socket viene chiuso per liberare risorse.

Ripetizione o Uscita:

Alla fine dell'attacco, l'utente può scegliere se ripetere l'attacco o terminare l'esecuzione.

Limitazioni Tecniche:

Invio Sequenziale: I pacchetti vengono inviati uno alla volta, riducendo la velocità massima.

Mancanza di Multi-threading: L'aggiunta di più thread migliorerebbe notevolmente le prestazioni.

Controllo delle Prestazioni Assente: Lo script non misura il tempo impiegato o la velocità di invio.

Effettuato scan di rete per identificare le porte aperte sulla macchina WindowsXP

Comando: sudo nmap –sU –p 1-1024 192.168.50.102

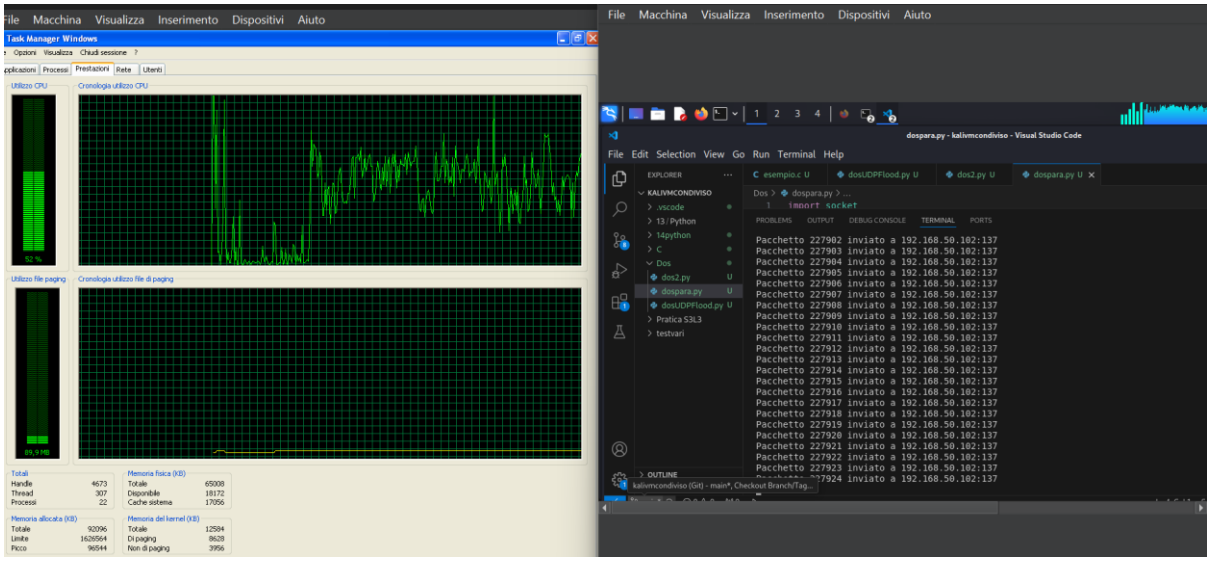
Porta UDP 137: Utilizzata da NetBIOS Name Service (NBNS) per la risoluzione dei nomi su reti basate su Windows.

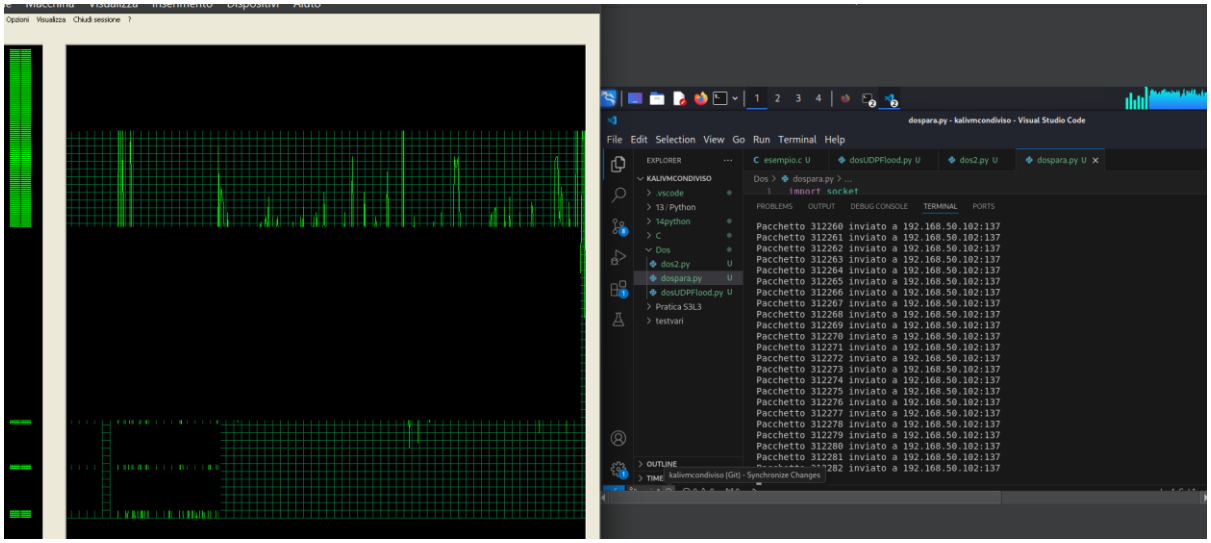
Windows XP usa NBNS per convertire i nomi NetBIOS in indirizzi IP (una funzione simile al DNS).

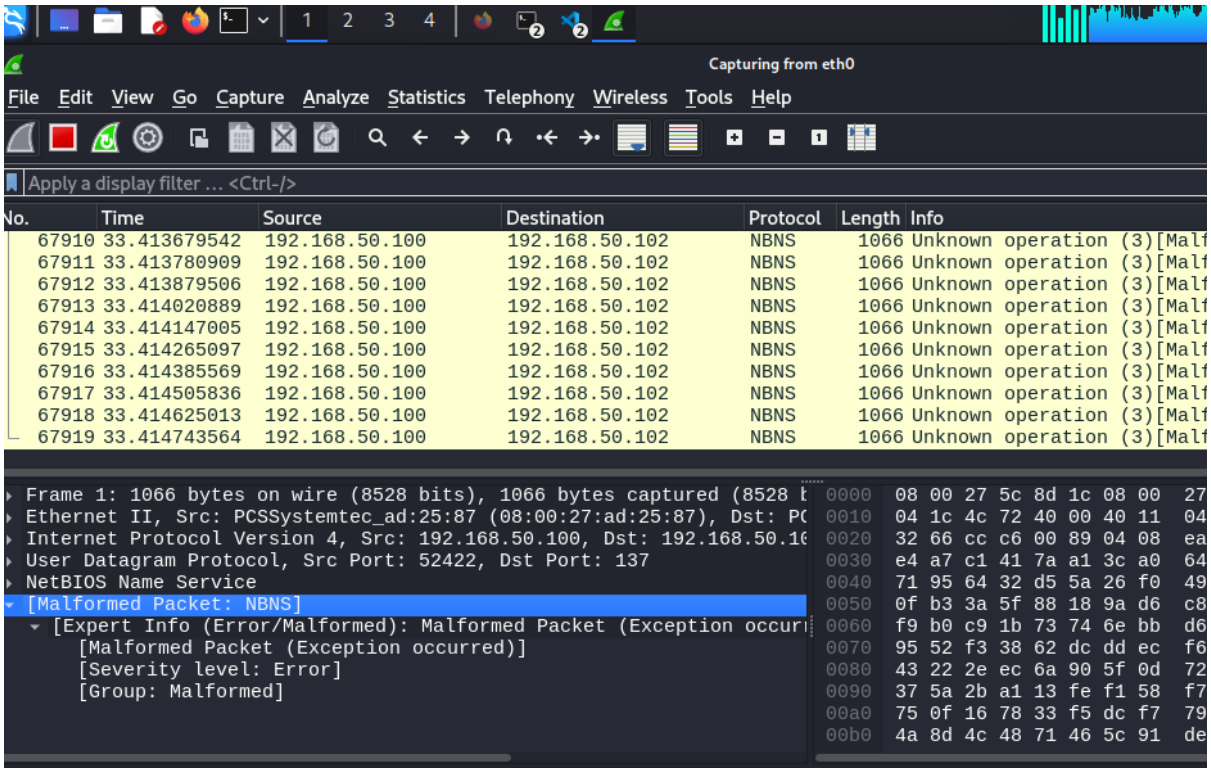
Un attacco UDP Flood verso questa porta tenta di saturare il servizio NetBIOS, rendendo difficoltosa la risoluzione dei nomi e, in alcuni casi, causando instabilità al sistema.

Osservazioni durante il Test

Utilizzo delle risorse:







Dettagli del Protocollo NBNS

NBNS è utilizzato per la risoluzione dei nomi NetBIOS in reti basate su Windows.

Porta tipica: 137 (UDP).

Quando un pacchetto è etichettato come "Malformed," significa che i dati catturati non rispettano le specifiche del protocollo.

Pacchetti Randomici: Lo script udp_flood sta inviando pacchetti con contenuti casuali generati da random._urandom. Questi pacchetti non contengono dati validi per il protocollo NBNS.

Risultato osservato: Wireshark rileva pacchetti NBNS "malformati," confermando che il traffico generato è casuale e non conforme al protocollo.

Conclusioni per il Report sull'Esercizio UDP Flood

Risultati Principali

L'esercizio ha dimostrato come un attacco UDP Flood possa saturare un servizio vulnerabile come NetBIOS Name Service (NBNS) sulla porta UDP 137 di Windows XP. La simulazione ha evidenziato l'efficacia dell'invio massivo di pacchetti casuali in ambienti di test controllati. Le seguenti osservazioni sono emerse durante il test:

Traffico Generato:
Pacchetti UDP di 1 KB generati casualmente sono stati rilevati tramite Wireshark come "malformati," indicando che il traffico non segue uno standard specifico del protocollo NBNS.

Comportamento del Target:
Windows XP ha risposto gestendo alcuni pacchetti NBNS, ma il sistema è diventato meno reattivo con l'aumentare del carico, confermando l'efficacia del test.

Limitazioni Riscontrate

Pacchetti Malformati:
I pacchetti inviati sono casuali e non conformi agli standard NBNS, riducendo l'efficacia dell'attacco per test specifici.

Invio Sequenziale:
L'invio di pacchetti è sequenziale, limitando le prestazioni. L'introduzione di multi-threading migliorerebbe l'efficienza.

Assenza di Feedback Avanzato:
Sebbene il programma mostri il rendimento finale, mancano metriche più dettagliate come i tempi medi di risposta del target.

Ottimizzazione delle Prestazioni: Aggiungere multi-threading o multiprocessing per migliorare la velocità di invio dei pacchetti.

Protocollo Simulato Realistico: Includere pacchetti conformi agli standard NBNS per test più precisi.

Miglioramento dell'Interfaccia: Integrare ulteriori metriche come il tempo medio di risposta e la percentuale di pacchetti ricevuti con successo.