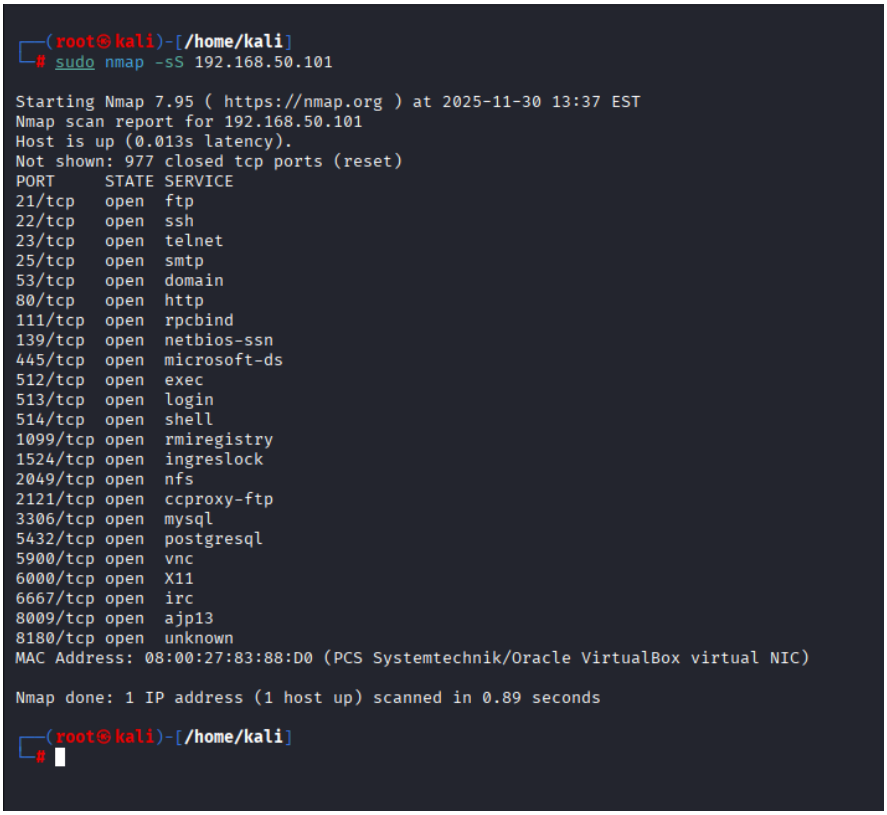
INDICE

INTRODUZIONE ED OBIETTIVO

INTRODUZIONE : il laboratorio è stato dedicato a comprendere in modo pratico come funzionano le scansioni di rete e, soprattutto, come alcuni metodi possano modificare il modo in cui un sistema risponde a queste analisi.
Per farlo è stato utilizzato Nmap, sfruttando le sue funzioni che permettono di cambiare la struttura e il comportamento dei pacchetti inviati verso un host.
Attraverso l'esecuzione graduale delle varie modalità di evasione, l'attività ha permesso di osservare da vicino come un sistema reagisce a traffico non convenzionale, offrendo una visione chiara e concreta del rapporto tra scansioni, servizi esposti e meccanismi di controllo della rete. Questo approccio ha reso l'esperienza più intuitiva, permettendo di collegare immediatamente la teoria ai risultati ottenuti.

OBIETTIVO: testare e documentare i principali metodi di evasione firewall offerti da Nmap, verificandone il comportamento in un ambiente controllato e identificando le differenze tra scansioni convenzionali e tecniche stealth.

SCANSIONE RETE



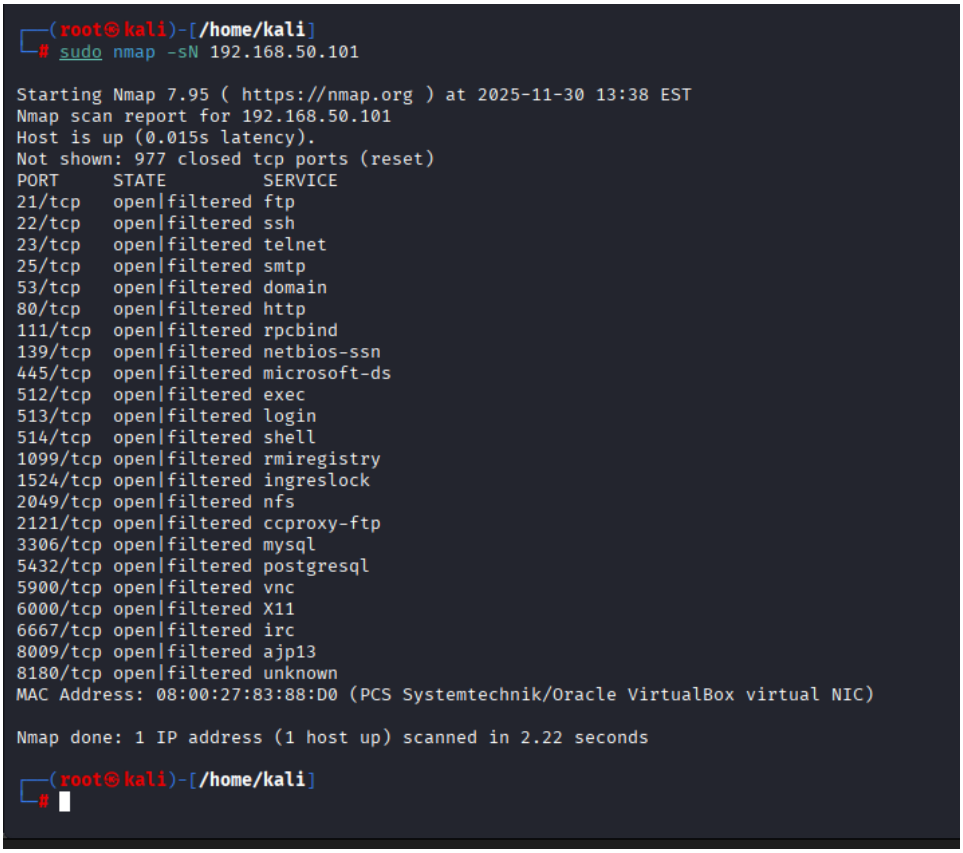
sudo nmap -sS 192.168.50.101 : il comando avvia una scansione di tipo SYN, una tecnica pensata per esplorare in modo rapido e discreto quali servizi siano effettivamente raggiungibili su un sistema remoto.
In pratica, questo metodo invia richieste di connessione “incomplete”, sufficienti però a verificare la presenza di un servizio senza impegnare l’host in una comunicazione completa. È un approccio che consente di ottenere una fotografia chiara dell’esposizione di un sistema, riducendo al minimo l’impatto sul dispositivo analizzato e senza alterarne il comportamento operativo.
Nel caso specifico, l’esecuzione del comando ha mostrato un host pienamente accessibile, con un numero significativo di porte in stato **open**; l’assenza di qualsiasi forma di filtraggio o blocco evidenzia che il sistema risponde in maniera diretta e trasparente a tutte le richieste inviate, permettendo una mappatura completa dei servizi attivi.
Il risultato complessivo conferma quindi una superficie di rete ampia e interamente visibile, elemento che in un contesto reale richiederebbe particolare attenzione dal punto di vista della sicurezza.

```
┌──(root㉿kali)-[/home/kali]
└─# sudo nmap -sF 192.168.50.101

Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-30 13:37 EST
Nmap scan report for 192.168.50.101
Host is up (0.013s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE         SERVICE
21/tcp   open|filtered ftp
22/tcp   open|filtered ssh
23/tcp   open|filtered telnet
25/tcp   open|filtered smtp
53/tcp   open|filtered domain
80/tcp   open|filtered http
111/tcp  open|filtered rpcbind
139/tcp  open|filtered netbios-ssn
445/tcp  open|filtered microsoft-ds
512/tcp  open|filtered exec
513/tcp  open|filtered login
514/tcp  open|filtered shell
1099/tcp open|filtered rmiregistry
1524/tcp open|filtered ingreslock
2049/tcp open|filtered nfs
2121/tcp open|filtered ccproxy-ftp
3306/tcp open|filtered mysql
5432/tcp open|filtered postgresql
5900/tcp open|filtered vnc
6000/tcp open|filtered X11
6667/tcp open|filtered irc
8009/tcp open|filtered ajp13
8180/tcp open|filtered unknown
MAC Address: 08:00:27:83:88:D0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.85 seconds
```

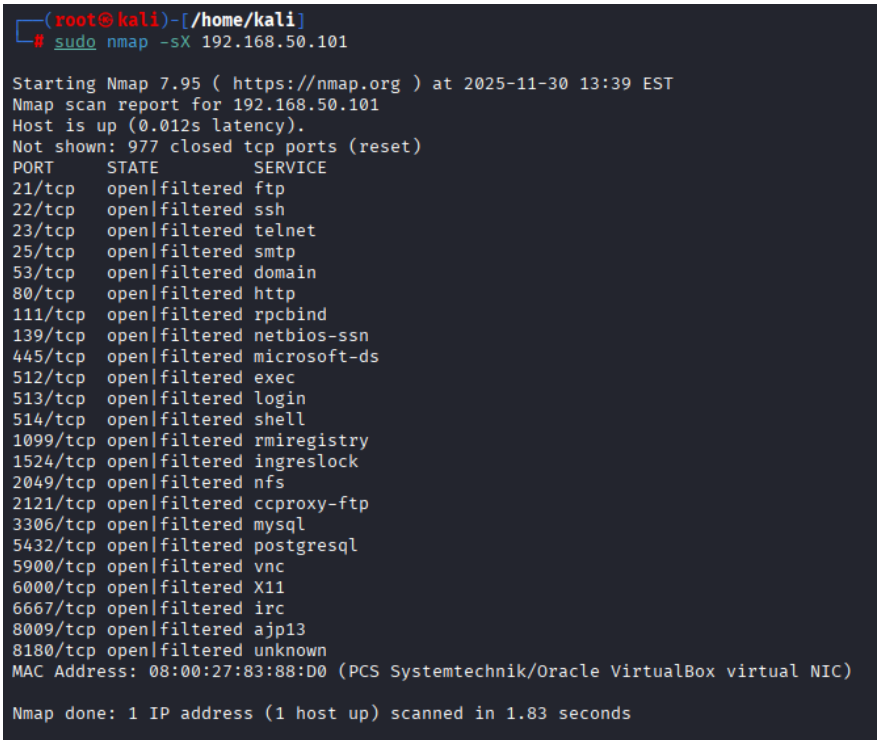
sudo nmap -sF 192.168.50.101 : il comando utilizza la modalità di scansione FIN, che invia pacchetti TCP con il solo flag FIN attivo per valutare la risposta del sistema a richieste non convenzionali.

Nel caso pratico, l'host ha classificato tutte le porte come **open|filtered**, comportamento tipico quando non vengono restituite risposte ai pacchetti FIN di conseguenza, con questa tecnica non è possibile capire se una porta sia realmente aperta oppure se un eventuale filtro stia impedendo la risposta. I

l risultato mostra quindi che lo scan FIN non permette una distinzione precisa sullo stato delle porte, ma evidenzia il modo in cui l'host gestisce sollecitazioni anomale, fornendo un'indicazione utile in un contesto di analisi difensiva.



sudo nmap -sN 192.168.50.101 : il comando utilizza la scansione NULL, una tecnica che invia pacchetti TCP privi di qualsiasi flag per verificare come il sistema gestisce richieste completamente anomale.
Questo tipo di pacchetto non rappresenta un tentativo di connessione reale e viene spesso ignorato dai sistemi operativi, motivo per cui viene usato per analizzare il comportamento del target in situazioni non standard.
Nel caso pratico, l'host ha restituito per tutte le porte lo stato **open|filtered**, indicando che non ha fornito una risposta ai pacchetti NULL di conseguenza, anche con questa tecnica non è possibile distinguere se una porta sia effettivamente aperta o se un eventuale filtro stia bloccando la comunicazione. Il risultato conferma che lo scan NULL non permette di ottenere informazioni precise sullo stato dei servizi, ma consente di osservare come il sistema tratti pacchetti che non seguono il normale flusso di una connessione TCP.



sudo nmap -sX 192.168.50.101 : il comando esegue la scansione XMAS, una tecnica che invia pacchetti TCP contemporaneamente.
Questo tipo di pacchetto non corrisponde a un comportamento normale di una connessione TCP e viene utilizzato per valutare come un sistema gestisce richieste particolarmente anomale.
Lo scopo è osservare se l'host fornisce informazioni aggiuntive quando viene sollecitato in modo non standard o se reagisce in modo simile a quanto accade con altre tecniche stealth.
Nel caso pratico, il sistema ha restituito per tutte le porte lo stato **open|filtered**, esattamente come nelle scansioni FIN e NULL; ciò indica che l'host non ha inviato risposte ai pacchetti XMAS, rendendo impossibile determinare se le porte siano realmente aperte o se un eventuale filtro stia impedendo la comunicazione. Il risultato mostra quindi che anche lo scan XMAS non permette di individuare lo stato reale dei servizi, ma conferma il comportamento coerente dell'host rispetto a pacchetti TCP non convenzionali.

```
┌──(root㉿kali)-[/home/kali]
└─# sudo nmap -sS --source-port 53 192.168.50.101

Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-30 13:40 EST
Nmap scan report for 192.168.50.101
Host is up (0.015s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:83:88:D0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.65 seconds
```

sudo nmap -sS --source-port 53 192.168.50.101 : il comando esegue una scansione SYN utilizzando la porta 53 come porta sorgente, una scelta non casuale perché molte reti considerano il traffico DNS come “fidato” e lo trattano con meno restrizioni. Questa tecnica viene usata per valutare se un sistema modifica il proprio comportamento quando le richieste provengono da una porta tipicamente associata a servizi legittimi.
L’idea è osservare se l’host, o un eventuale filtro lungo il percorso, applica regole diverse in base alla porta utilizzata per inviare i pacchetti.
Nel caso pratico, il risultato ottenuto mostra tutte le porte in stato **open**, esattamente come nella scansione SYN standard. Questo significa che il sistema non applica alcuna distinzione tra richieste provenienti da porte comuni e richieste provenienti dalla porta 53, indicando un comportamento completamente trasparente e l’assenza di qualsiasi meccanismo di filtraggio o controllo basato su questo parametro. Il risultato conferma quindi che, in questo contesto, la manipolazione della porta sorgente non modifica la visibilità dei servizi esposti.

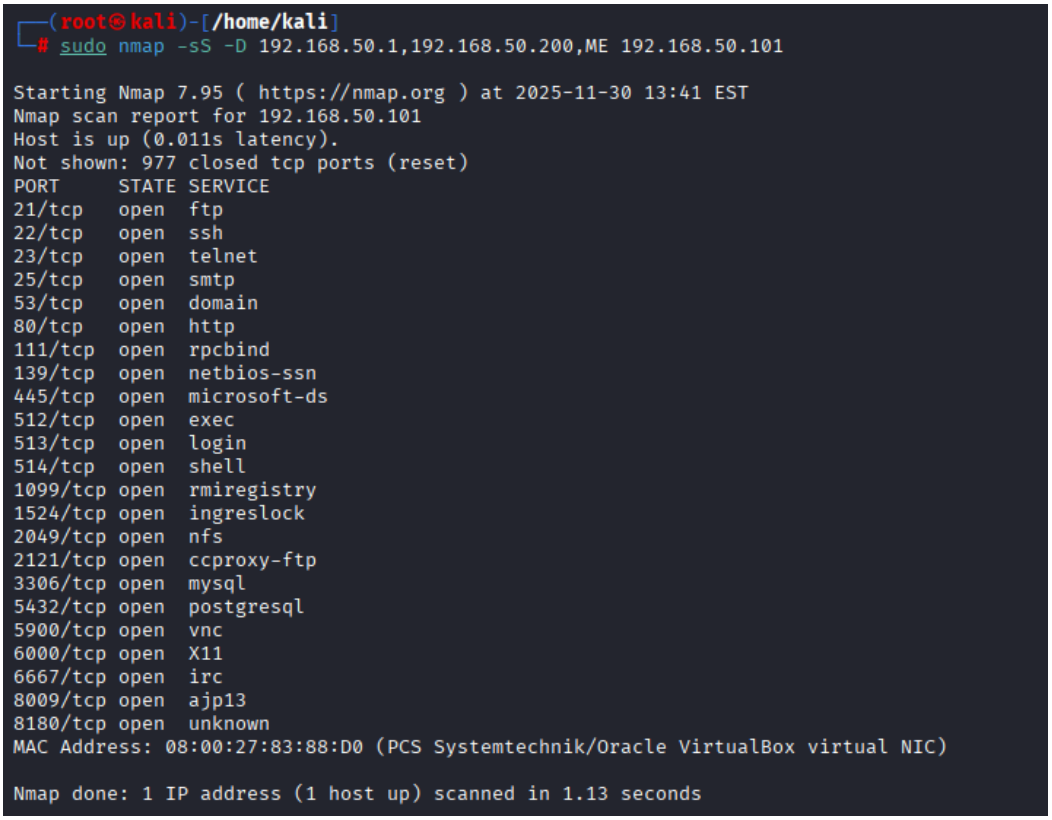
```
┌──(root㉿kali)-[/home/kali]
└─# sudo nmap -sS -f 192.168.50.101

Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-30 13:40 EST
Nmap scan report for 192.168.50.101
Host is up (0.012s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:83:88:D0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.14 seconds
```

sudo nmap -sS -f 192.168.50.101 : il comando esegue una scansione SYN applicando la frammentazione dei pacchetti, una tecnica che suddivide i dati della richiesta in segmenti molto piccoli.

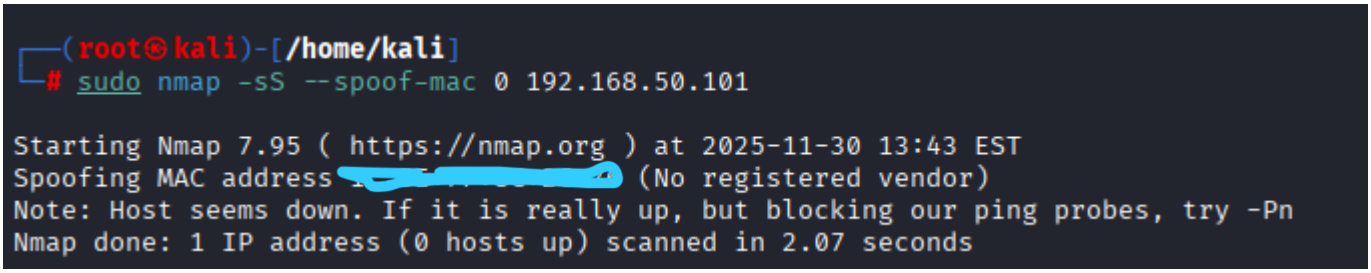
Questo metodo viene utilizzato per osservare come un sistema gestisce pacchetti frammentati, che possono talvolta aggirare controlli superficiali o filtri che analizzano soltanto la prima parte della comunicazione. L'obiettivo è verificare se, modificando la struttura dei pacchetti, il sistema risponde in maniera diversa rispetto a una scansione tradizionale.

Nel caso pratico, l'host ha restituito nuovamente tutte le porte in stato **open**, esattamente come nello scan SYN standard; questo indica che la frammentazione non ha influenzato in alcun modo la visibilità dei servizi e che il sistema gestisce correttamente i pacchetti frammentati senza applicare restrizioni o comportamenti particolari.

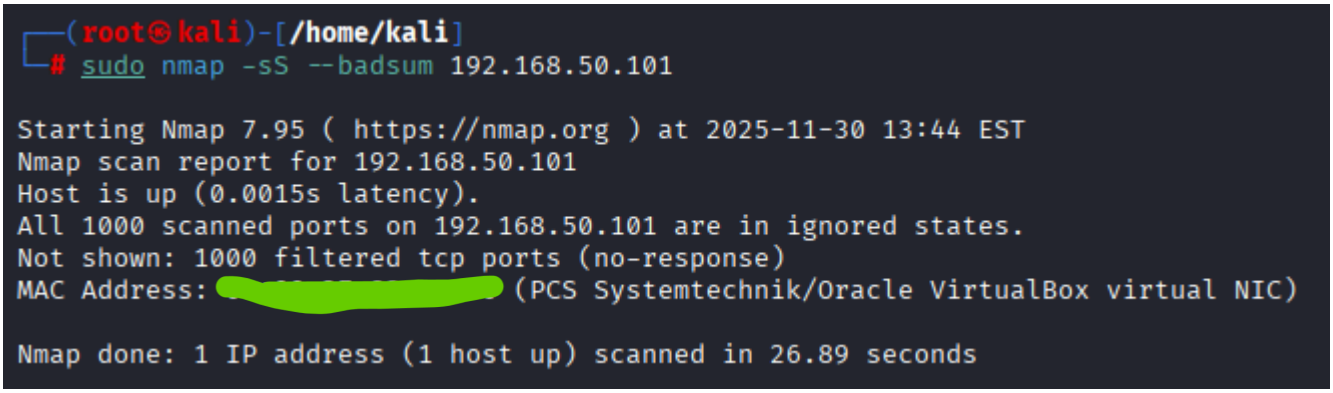
Il risultato conferma che, in questo ambiente, la frammentazione non altera l'esito della scansione e non modifica l'accesso ai servizi esposti.



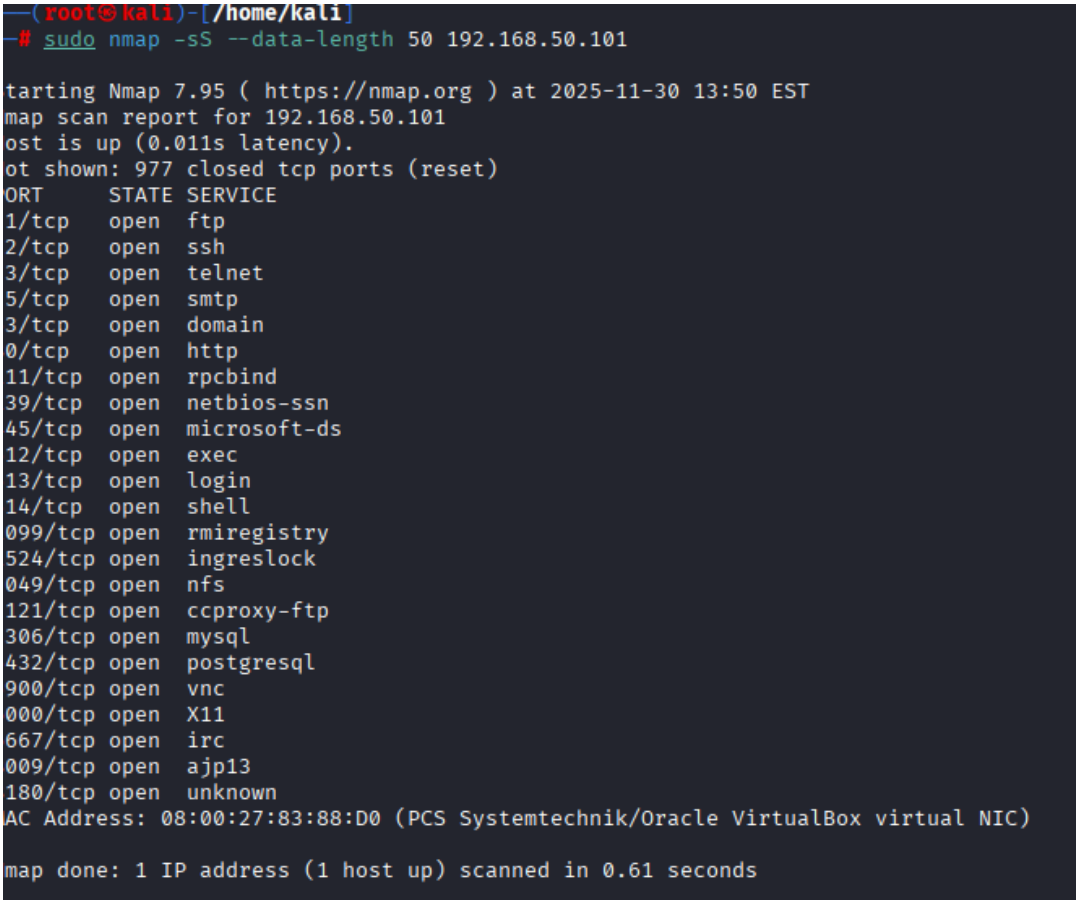
sudo nmap -sS -D 192.168.50.1,192.168.50.200,ME 192.168.50.101 : questo comando utilizza la modalità decoy, una tecnica che permette di mascherare l'origine reale della scansione generando traffico proveniente da più indirizzi IP diversi.
L'idea è simulare la presenza di più sorgenti contemporanee, inserendo il proprio indirizzo reale tra indirizzi fittizi o non rilevanti, così da rendere meno immediata l'identificazione dell'host che sta effettivamente conducendo l'analisi.
Questa strategia viene impiegata non per modificare la visibilità dei servizi, ma per confondere eventuali sistemi di logging o monitoraggio che registrano attività sospette.
Nel caso pratico, la scansione ha restituito tutte le porte in stato **open**, esattamente come una normale scansione SYN, confermando che il comportamento dell'host non cambia in base al numero o alla natura degli indirizzi sorgente dichiarati. L'operazione, quindi, non influisce sul rilevamento dei servizi esposti, ma offre una rappresentazione utile del modo in cui un sistema registrerebbe traffico proveniente da sorgenti apparentemente multiple.
Questo rende la tecnica significativa in un contesto di analisi, poiché permette di valutare l'efficacia di eventuali meccanismi di monitoraggio nel distinguere tra traffico legittimo e tentativi di mascheramento.



sudo nmap -sS --spoof-mac 0 192.168.50.101 : questo comando forza Nmap a inviare i propri pacchetti utilizzando un indirizzo MAC falsificato, generato casualmente.
Lo scopo della tecnica è osservare come un sistema o un eventuale meccanismo di rete reagisca quando il pacchetto sembra provenire da un dispositivo completamente diverso rispetto a quello reale. L'alterazione dell'indirizzo MAC è una forma di mascheramento che può essere impiegata per eludere controlli superficiali basati sulla verifica dell'origine del traffico o per rendere meno immediata l'associazione tra l'attività di scansione e l'host effettivo che la sta eseguendo.
Nel caso pratico, l'host non ha risposto alla scansione e Nmap ha riportato lo stato "host seems down", nonostante la macchina fosse effettivamente raggiungibile. Questo comportamento dipende dal fatto che l'indirizzo MAC generato non corrisponde a nessun dispositivo noto nella rete locale e, di conseguenza, il sistema non ha aggiornato la propria tabella ARP, ignorando completamente i pacchetti ricevuti. Il risultato mostra chiaramente che, in questo contesto, la falsificazione del MAC interrompe la comunicazione a livello locale e impedisce a Nmap di ottenere qualsiasi informazione sulle porte, evidenziando come la manipolazione del livello di collegamento influisca in modo diretto sulla visibilità dei servizi.



sudo nmap -sS --badsum 192.168.50.101 : questo comando forza Nmap a generare pacchetti TCP con un checksum deliberatamente errato.
Il checksum è un valore di controllo utilizzato dai sistemi di rete per verificare l'integrità dei pacchetti ricevuti; quando risulta invalido, il pacchetto viene scartato automaticamente dal sistema senza alcuna risposta. L'obiettivo della tecnica è osservare come l'host gestisce pacchetti corrotti e verificare se restituisce comunque informazioni utili o se interrompe completamente la comunicazione.
Nel caso pratico, la scansione ha riportato tutte le porte nello stato "ignored" o "filtered", senza alcuna risposta significativa da parte dell'host. Questo comportamento è previsto: il sistema ha scartato ogni pacchetto non conforme, impedendo a Nmap di ottenere dati sulle porte aperte. Il risultato dimostra che, in questo contesto, pacchetti con checksum errato non permettono di rilevare correttamente i servizi esposti e rendono la scansione inefficace, confermando che l'integrità del pacchetto è un requisito imprescindibile per qualunque tipo di comunicazione di rete.



sudo nmap -sS --data-length 50 192.168.50.101 : questo comando esegue una scansione SYN aggiungendo 50 byte di dati extra all'interno di ogni pacchetto inviato.
L'obiettivo della tecnica è modificare artificialmente le dimensioni del traffico generato, così da osservare se un sistema di rete o un eventuale filtro reagisca in modo diverso quando i pacchetti presentano un formato meno prevedibile rispetto al normale. L'inserimento di dati aggiuntivi non altera la logica della connessione, ma permette di valutare se la variazione nella lunghezza del pacchetto possa influenzare l'identificazione dei servizi o l'elaborazione da parte dell'host.
Nel caso pratico, l'host ha risposto mostrando tutte le porte rilevate come **open**, esattamente come avviene con una scansione SYN standard. Questo risultato indica che l'aggiunta di dati extra non ha modificato il comportamento del sistema né alterato la visibilità dei servizi esposti. Il sistema ha gestito correttamente i pacchetti più lunghi senza introdurre restrizioni o anomalie, confermando che, in questo ambiente, la variazione della lunghezza non ha alcun impatto sulla rilevazione delle porte aperte.

CONCLUSIONE

Il laboratorio ha permesso di osservare in modo semplice e diretto come un sistema risponde a diversi tipi di richieste di rete generate con Nmap.
Utilizzando metodi più o meno “creativi”, come modificare la forma dei pacchetti, simulare indirizzi diversi o cambiare alcuni parametri della comunicazione, è stato possibile vedere quali informazioni il sistema rende disponibili e quali invece non vengono restituite.
Nel nostro ambiente di test, privo di protezioni o filtri, la maggior parte delle scansioni ha mostrato chiaramente tutti i servizi attivi, confermando che l’host risponde senza particolari limitazioni. Alcune tecniche non hanno prodotto risultati utilizzabili, mostrando invece che il sistema ignora completamente quel tipo di richieste.
Nel complesso, l’attività ha offerto una panoramica chiara sul comportamento dell’host e su come cambia la visibilità dei servizi in base al tipo di richiesta inviata. Questo permette di comprendere meglio quali informazioni sono effettivamente esposte e quanto sia importante, in un contesto reale, affiancare a questi sistemi adeguate misure di protezione per evitare che un attaccante possa ottenere una mappa completa dei servizi disponibili.