Esercizio 1 - Privilege escalation

Leggete tutto il testo prima di fare qualsiasi operazione!

Scaricare sulla VM Kali il file change3 e renderlo eseguibile

$ **chmod +x ./change3**

Il comando apporta una modifica a file dentro **/usr/bin** e **/etc**

Fase 1:

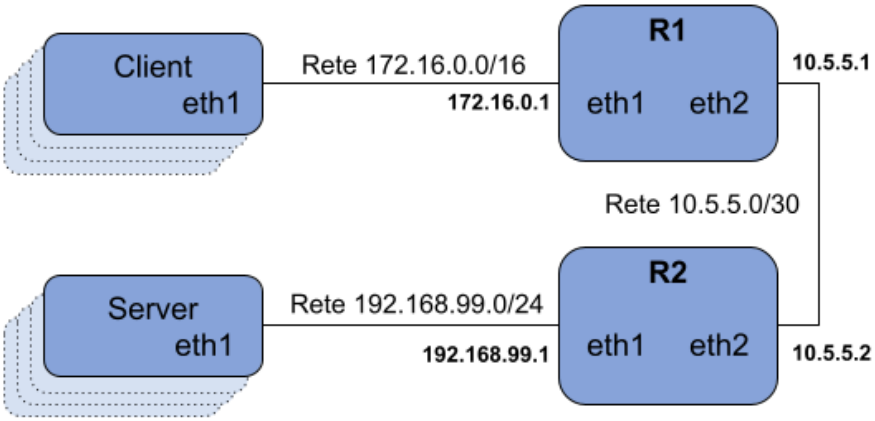
- progettare una strategia per identificare i file modificati e il tipo effettivo di modifiche apportate.
- lanciare **sudo ./change3**
- attuare la strategia ideata al punto 1 per identificare il file modificato e il tipo di modifica apportata.

documentare tutti i passi svolti in modo dettagliato nel file da consegnare **integrity.txt**

Fase 2:
catturate i comandi seguenti e l'output in uno screenshot **privesc.png** (o più di uno, eventualmente, numerandoli)

- iniziate a operare come utente kali senza sudo
- sfruttate le modifiche apportate dal comando change3 per svolgere un'operazione che vi possa permettere di diventare root (si tratta di un procedimento che prevede pochissimi passaggi)
- dimostrate il successo diventando root e lanciando id

Esercizio 2 - Firewall



Facendo riferimento allo schema di rete sopra riportato, si definiscano regole di filtraggio che consentano il traffico come sotto specificato; qualsiasi altro pacchetto deve essere scartato. **NOTA**: le regole devono essere installate su **ogni host** coinvolto nel flusso di traffico specificato.

1. i Client devono poter accedere via HTTPS (porta TCP 443) a qualunque server; il reale indirizzo dei client deve essere nascosto sia a R2 che ai server
2. il Server 192.168.99.2 deve poter accedere via SSH (porta TCP 22) ai due Router
3. il Client 172.16.0.2 deve poter accedere via SSH al server 192.168.99.2 utilizzando R1 e R2 come "Jump Host" nel modello SSH Tunneling

MODALITÀ DI CONSEGNA:

Lo studente dovrà consegnare 1 file **iptables.txt** contenente quattro sezioni chiaramente contrassegnate, una per ogni host su cui è richiesto di installare le regole necessarie, nelle quali elencare i comandi (sintatticamente corretti) che le realizzano.

Esercizio 3 - Intrusion detection

L'obiettivo è quello di creare una serie di regole per l'IDS visto a lezione, Suricata, in modo che permetta di recuperare e ricostruire una flag presente all'interno di **questo tracciato di traffico**.

All'interno del tracciato è presente una flag nel seguente formato:

`SEC{...caratteri...}SEC`
Dove
SEC{ e **}SEC** sono i riferimenti di apertura e chiusura della flag e vanno considerati ALLA LETTERA!
...caratteri... sono una serie di parole di senso compiuto che identificano in modo riconoscibile il pezzo della flag corretto.

La flag è divisa in 2 parti, per cui in **...caratteri...** troverete l'identificazione della PRIMA PARTE della flag e della SECONDA PARTE della flag.

Le due parti però, potrebbero trovarsi su stream o protocolli diversi, ad esempio:
- La prima parte all'interno di uno stream TCP e la seconda su un altro stream TCP separato
- La prima parte all'interno di uno stream TCP e la seconda su un altro protocollo.
- ecc

Il lavoro da svolgere sarà diviso in diverse fasi.

PRIMA FASE - Analizzare MANUALMENTE il tracciato di traffico per

1. identificare tutti i protocolli sia di livello trasporto (livello 4) sia di livello applicativo (livello 5) presenti all'interno del pacchetto
2. identificare con precisione gli IP che generano tutte le connessioni e il relativo "verso" di invio dei contenuti (qualora ce ne fossero)
3. capire dove si trova la flag.

SECONDA FASE

Scrivere una(o più di una se necessario) regola suricata in modalità alert per il traffico del/dei protocollo/i in questione, specificandone gli IP trovati precedentemente.
La specifica degli IP può essere fatta singolarmente (attenzione alle direzioni delle connessioni) oppure utilizzando le variabili di ambiente di suricata per specificare subnet.
Se creata correttamente la regola e configurato correttamente suricata (si ricordi l'esercitazione in classe su MQTT) nei log dovrebbe essere possibile leggere il contenuto degli stream che fanno il match con le regole.

TERZA E ULTIMA FASE

Parsare a piacimento il contenuto degli alert (il file specifico di log?) generati da suricata e ricostruire la flag.

MODALITÀ DI CONSEGNA:

Devono essere consegnati

- Un file **report.txt** dove spieghi il risultato dell'analisi manuale del tracciato di traffico, identificando protocolli, ip e tutto ciò di utile si possa dire riguardo alle informazioni scoperte nella prima fase
- Un file **suricata.yml** con la configurazione di suricata
- Un file **seclab.rules** con le regole create e utilizzate
- Un file **parse.txt** con script o descrizione libera di comandi utilizzati per parsare il log e ricostruire la flag e.g. "cat FILE DI LOG | grep SEC.."
- Un file **screenshot.*estensione-grafica-preferita*** che mostri l'output del parsing con la relativa flag.