

Laboratorio di Reti A.A 2022/2023
Compito del 20/07/2023
Durata 2h00m

Materiale del corso: <http://public.aries.ing.man/LaboratorioReti/>

Esercizio 1

(editor di testo)

Si definiscano le seguenti regole iptables per la configurazione di un firewall di una workstation dotata di due interfacce di rete (eth0 e eth1). In particolare, si vogliono configurare le seguenti regole.

Catena di INPUT:

- impostare la politica di default per la chain per non consentire traffico in ingresso;
- accettare tutti i pacchetti provenienti dalla sottorete 10.12.10.0/24;
- accettare tutti i pacchetti con porta sorgente UDP 7 e porta destinazione UDP 7;
- accettare pacchetti relativi alle connessioni iniziate dalla workstation;

Catena di OUTPUT

- impostare la politica di default su drop;
- consentire il traffico in uscita per il servizio di SSH;
- consentire il traffico in uscita relativo a HTTP e HTTPS,
- consentire il traffico in uscita relativo a query DNS inviate dall'host;
- consentire il traffico in uscita verso la sottorete 10.14.0.0/16;

Catena di FORWARD

- impostare la politica di default su DROP
- Consentire il traffico proveniente dall'interfaccia eth1 e destinato all'interfaccia eth0 con porta di destinazione TCP 25 con indirizzo di destinazione 10.16.22.15;

Esercizio 2

(editor di testo)

Si implementi una playbook Ansible, pensata per una distribuzione Linux di tipo Debian-based. La playbook deve configurare un framework di “federated learning” su un gruppo di macchine identificato dal nominativo “dev-servers”. La playbook deve includere un file vars.yml (presente nella stessa directory della playbook) che contiene alcune variabili come: *dev_user*, *data_set_url*, *aggregator_address*, *etc....*. Inoltre, la playbook deve copiare un file di configurazione “flower.conf.j2” (quindi un file di template che chiama alcune variabili definite in vars.yml).

La playbook deve implementare i seguenti task:

- effettuare un update dell'indice dei repository;

- effettuare l'upgrade dei pacchetti installati sulla macchina;
- installare i pacchetti "flower", "wget" e "curl";
- creare una directory /etc/flowers, con utente e gruppo il valore definito nella variabile *dev_user*, permission bit 0755;
- copiare il file di configurazione (scritto con templating jinja2) "flower.conf.j2" in "/etc/flower/flower.conf", utente e gruppo quelli definiti nella variabile *dev_user*, permission bit 0755;
- Avviare il servizio di flowers utilizzando il modulo di Ansible opportuno e non utilizzando gli handlers;

Esercizio 3

(editor di testo + interprete python)

Si implementi un'applicazione Client/Server in Python utilizzando le socket TCP che consenta a una macchina client di verificare la shell di default associata a un utente su una macchina server (ad esempio /bin/sh o /usr/bin/bash). A tal fine si ricordi che le informazioni sugli utenti in una macchina di tipo UNIX-like sono memorizzati nel file /etc/passwd. Il processo server deve mettersi in ascolto su una porta, che viene specificata in input dall'utente, e gestire le richieste dei client in modo sequenziale.

Il client deve chiedere in input all'utente l'hostname e la porta del server a cui deve collegarsi. Una volta connesso, il client chiede all'utente il nome dell'utente del quale si vuole controllare la shell di default directory e la invia al server. Al ricevimento del parametro di ricerca, il server esegue il comando corrispondente, ne legge l'output, seleziona la shell di default utilizzando i metodi per la manipolazione delle stringhe in Python, e infine lo invia al client. Al termine della richiesta, il server deve chiudere la connessione con il client, mentre il client stampa a video l'output ricevuto dal server e termina.