

Laboratorio di Reti  
A.A 2021/2022  
Compito del 12/01/2023  
Durata 2h30m

**Slide 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 macchina Linux. In particolare, si vogliono configurare alcune regole per la catena di INPUT e per quella di OUTPUT.

Catena di INPUT:

- impostare la politica di default per la chain di input per non accettare pacchetti;
- abilitare il traffico SSH verso la macchina (si noti che la porta del servizio è la 8022 e non quella di default);
- abilitare tutto il traffico proveniente dalla sottorete 10.16.0.0/16;
- abilitare il traffico ICMP;

Catena di OUTPUT

- non consentire traffico verso la sottorete 10.16.10.0/24;

Catena di FORWARD

- consentire il passaggio del traffico tra l'interfaccia eth0 e l'interfaccia eth1;

### **Esercizio 2**

(editor di testo)

Si supponga di dover decodificare una parte di messaggio contenuto in un protocollo applicativo per applicazioni IoT. La parte di header da decodificare ha dimensione 8 bit ed è codificata come big endian. Gli 8 bit comprendono 1 bit che indica se il messaggio è una query o una risposta (1=query, 0=response), 2 bit per identificare il tipo di query, 1 bit per identificare se la query è urgente o meno, e i gli ultimi 4 bit per identificare il formato in cui si desidera ricevere la risposta.

| Q/R (1 bit) | Q\_TYPE (2 bit) | Q\_URG (1 bit) | R\_FORMAT (4 bit) |

Si scrivano le operazioni di maschera e bit shifting necessarie per reperire il valore delle diverse componenti. Inoltre, si supponga che dopo gli 8 bit letti, se il bit Q/R è a 1, i successivi 16 bit presenti nel messaggio codificano un numero intero senza segno. Si implementi la lettura di tale numero dal buffer.

### Esercizio 3

(editor di testo)

Si implementi una playbook Ansible, pensata per macchine Linux Debian-based. La playbook deve installare e configurare un web server “apache2” sul quale configurare un blog statico.

Per prima cosa, la playbook deve aggiornare l’indice dei repository e installare il pacchetto “apache2”. Poi, deve copiare il file di configurazione “apache2.conf” in /etc/apache2 con owner “root”, group “root” e permessi ‘0644’. Inoltre, la playbook deve creare una nuova directory “/var/www/blog” e associarla all’utente “root” e al gruppo “root”, permission mask 0744. Infine, si deve copiare il contenuto di una cartella locale blog all’interno della cartella /var/www/blog appena creata sulla macchina remota.

Si realizzino i seguenti task:

- aggiornamento dell’indice dei repository;
- installazione del pacchetto “apache2”;
- copia del file di configurazione “apache2.conf”, presente nella directory corrente del controller in “/etc/apache2/”
- creazione della directory “/var/www/blog”;
- copia del contenuto della cartella blog presente sul controller in “/var/www/blog”;
- enable e start di “apache2”;

### Esercizio 4

(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 i processi associati a un determinato utente in esecuzione su una macchina remota. 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 invia al server una stringa contenente il nome dell’utente (letta in input da terminale) di cui si vuole ottenere la lista dei processi, ad esempio “root”. Al ricevimento del nome utente, il server esegue il comando corrispondente, ne legge l’output e lo invia al client. Al termine della richiesta il server chiude la connessione con il client, mentre il client stampa a video l’output ricevuto dal server e termina.