



**Politecnico  
di Torino**

Master of Science in Computer Engineering

Master Degree Thesis

**Prova**

**Supervisors**

prof. Fulvio Valenza  
prof. Name Surname  
dott. Name Surname  
dott. Name Surname

**Candidate**

Benito MARRA

ACADEMIC YEAR 2023-2024



# Summary

Text of the summary

# Acknowledgements

Acknowledgement (optional)

# Contents

List of Figures	6
List of Tables	7
Listings	8
<b>1 Introduzione</b>	<b>10</b>
1.1 Obiettivo della Tesi . . . . .	10
1.2 Descrizione della tesi . . . . .	11
<b>2 Background</b>	<b>13</b>
2.1 Introduzione a Verefoo . . . . .	13
2.2 Definizione dei Security Requirements . . . . .	13
2.3 Ambiente di test . . . . .	13
<b>3 Docker</b>	<b>14</b>
3.1 Introduzione a Docker . . . . .	14
3.2 Docker Compose . . . . .	14
<b>4 Conclusions</b>	<b>15</b>
Bibliography	16

# List of Figures

# List of Tables

# Listings





# Chapter 1

## Introduzione

### 1.1 Obiettivo della Tesi

Nell'ultimo decennio diverse tecnologie di rete si sono sviluppate, creando reti sempre più robuste ed efficienti. In questo scenario, una tecnologia in particolare, la "*Network Function Virtualization*"(NFV) ha reso possibile creare delle reti che svolgono le funzioni di sicurezza e trasporto dei dati tramite la virtualizzazione di quest'ultime. Ciò ha permesso di definire le "*Software Defined Network*"(SDN) che introducono la possibilità di controllare le operazioni di rete tramite software.

Sfruttando queste tecnologie è stato sviluppato Verefoo(VERified REfinement and Optimized Orchestration) cioè un framework in grado di riuscire ad implementare nei nodi della rete i vari "*Network Security Requirements*"(NSR) in una topologia predefinita e fornita come input al framework.

Gli obiettivi di questa tesi possono essere riassunti nei 3 punti:

1. Sistemare e migliorare una demo precedentemente sviluppata per mostrare le potenzialità del framework. All'interno di questa demo il framework può accettare solo un determinato requisito di sicurezza di rete ovvero la *Protection Property* cioè la possibilità di far passare il traffico crittografato da un nodo ad un altro della topologia in maniera sicura. Per fare ciò verranno allocate nella topologia dei VPN Gateway in grado di crittare e decrittare il traffico in ingresso ed in uscita.
2. Cercare una soluzione per poter integrare le varie versioni di Verefoo. Prima del lavoro che verrà spiegato più avanti il framework era in grado, in un'unica esecuzione, di allocare solo VPN Gateway oppure di allocare solo FireWall per garantire la *Isolation Property* e la *Reachability Property*, mentre l'obiettivo che si cerca di raggiungere è di riuscire ad allocare contemporaneamente in un'unica esecuzione del framework sia i VPN Gateway che i Firewall necessari al corretto funzionamento della rete.
3. Produrre una nuova demo, diversa da quella del punto 1 in grado di mostrare le nuove capacità del framework. La demo prodotta dovrà essere in grado di poter allocare, in un'unica esecuzione tutti i Firewall e i VPN Gateway definiti dalle proprietà di rete che sono stati forniti come input al framework.

Da questo momento in poi per tutto il documento mi riferirò alla demo da modificare come Demo-A mentre alla demo da implementare come Demo-B.

## 1.2 Descrizione della tesi

Dopo aver spiegato nel Capitolo [1] gli obiettivi ed il lavoro prodotto per raggiungerli, il resto della tesi è definito nel seguente modo:

- Nella prima parte del Capitolo [2] si descrive il framework di Verefoo, ponendo particolare attenzione sul suo funzionamento ad alto e basso livello. Nella seconda parte sono descritte le definizioni delle Proprietà di sicurezza da passare come input al framework, con una spiegazione dettagliata di come queste intervengono nella definizione della topologia finale che verrà fornita come output dal framework. Infine verranno introdotti i grafi che verefoo richiede ed utilizza nella computazione dei vari *NSF*.
- Il Capitolo [3] definisce l'architettura di docker, specificando la differenza tra usare docker per la virtualizzazione e delle semplici macchine virtuali. Successivamente viene fatto un approfondimento sul docker-compose, un tool in grado di poter istanziare più container velocemente tramite script. Nella parte finale viene spiegato come effettuare il networking sui container istanziati, come definirlo tramite docker-compose e come testare le comunicazioni in modo efficiente.
- Il Capitolo [4] descrive i lavori svolti nella Demo-A. Inizialmente viene descritto tramite pezzi di codice lo sviluppo dell'installer prodotto affinché un qualsiasi utente possa utilizzare la demo in maniera pratica ed agile. Nei paragrafi successivi vengono evidenziati i punti critici incontrati, elencando le modifiche apportate affinché essa possa funzionare correttamente. Nell'ultimo paragrafo verranno specificati ulteriori upgrade che si possono inserire nella demo per mettere in mostra in maniera ancora più evidente il lavoro svolto da Verefoo.
- Il Capitolo [5] contiene un breve approfondimento degli obiettivi 2 e 3 della tesi con una descrizione accurata dei vari passi che sono stati svolti prima della soluzione definitiva. In questo capitolo si evidenziano anche le difficoltà che sono emerse lavorando al framework, e verranno proposte alcune soluzioni per poter evitare simili problematiche in futuro.
- Il Capitolo [6] descrive la soluzione finale scelta ed implementata su Verefoo. In questo capitolo viene quindi spiegato, anche tramite frammenti di codice, gli step che il framework eseguirà per produrre in output una rete che soddisfi contemporaneamente tutti i requisiti di sicurezza passati come input.
- Il Capitolo [7] descrive lo sviluppo della Demo-B. In un primo momento si mostra la topologia di rete scelta da virtualizzare, con la finalità di indicare

le nuove funzionalità di verefoo sviluppate al completamento del secondo obiettivo della tesi. Successivamente vengono descritti tutti i passi svolti per implementare la demo, con un commento per il codice che è stato utilizzato. Infine si evidenziano anche i limiti della demo prodotta con alcuni futuri aggiornamenti possibili.

- Il Capitolo [\[8\]](#) elenca i lavori futuri da svolgere all'interno del framework, la necessità di poter implementare soluzioni alternative a quella proposta in questo documento, e i limiti che devono essere superati affinché il framework possa essere utile in un ambiente reale e non solo di testing virtualizzato. Infine vengono descritte le conclusioni del lavoro, con un riassunto generale di tutto ciò che è stato prodotto.

# Chapter 2

## Background

### 2.1 Introduzione a Verefoo

Descrivo il framework, cosa fa e quale è il suo scopo.

### 2.2 Definizione dei Security Requirements

Spiego in maniera breve come sono definiti i requisiti di sicurezza su verefoo

### 2.3 Ambiente di test

Spiego perchè è necessario avere un ambiente dove mostrare il framework in funzione  
description [\[1\]](#)

# Chapter 3

## Docker

### 3.1 Introduzione a Docker

Parte di docker.

### 3.2 Docker Compose

Parte di Docker compose.

description [\[1\]](#)

# Chapter 4

## Conclusions

Conclusion and future works

# Bibliography

- [1] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, “Automated optimal firewall orchestration and configuration in virtualized networks,” in *NOMS 2020 - IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, April 20-24, 2020*. IEEE, 2020, pp. 1–7. [Online]. Available: <https://doi.org/10.1109/NOMS47738.2020.9110402>