



**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* ha reso possibile creare delle reti che svolge le sue funzioni di sicurezza e trasporto tramite la virtualizzazione di quest'ultime. Ciò ha permesso di definire le *Software Defined Network* che introducono la possibilità di creare dei forwarding path tramite software.

Sfruttando queste tecnologie viene concepito Verefoo(VERified REfinement and Optimized Orchestration) un framework in grado di riuscire ad allocare le varie NFV in una topologia predefinita e fornita come input al framework.

Il lavoro svolto in questa tesi è stato svolto su ed attraverso questo framework, cercando di raggiungere 3 obiettivi ben precisi:

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 prima demo come Demo-A mentre alla seconda come Demo-B.

## 1.2 Descrizione della tesi

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

- Il Capitolo [2] descrive il framework di Verefoo, ponendo particolare attenzione sul suo funzionamento ad alto e basso livello. Nella seconda parte del capitolo si potranno trovare le definizioni delle Proprietà di sicurezza da passare come input al framework, con una spiegazione dettagliata di come queste intervengano 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 fra usare docker e delle semplici macchine virtuali. Nella seconda parte del capitolo verrà fatto un approfondimento su docker-compose, un tool in grado di poter istanziare più container velocemente tramite script. Nella parte finale questo capitolo tratterà il networking sui vari container, come effettuarlo tramite docker-compose e come far comunicare i container in modo efficiente.
- Il Capitolo [4] descriverà i lavori svolti nella Demo-A. Nella prima parte verrà 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 verranno evidenziati i punti critici riscontrati nella demo precedentemente scritta, 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 evidenzieranno anche le difficoltà che sono emerse lavorando al framework, e verranno proposte alcune soluzioni per poter evitare ulteriori problematiche in futuro.
- Il Capitolo [6] descrive la soluzione finale adottata nell'implementazione su Verefoo. In questo capitolo viene quindi spiegato, anche tramite frammenti di codice di Verefoo, gli step che il framework eseguirà per soddisfare contemporaneamente i requisiti di sicurezza che la rete deve avere.
- Il Capitolo [7] approfondirà lo sviluppo della seconda demo. Inizialmente verrà descritto l'obiettivo da raggiungere con lo sviluppo della demo, mostrando e motivando la topologia di rete scelta da virtualizzare per poter raggiungere l'obiettivo. All'interno di questo capitolo si trova anche la parte implementativa della demo con il codice utilizzato e spiegato.

- Il Capitolo [\[8\]](#) elencherà i lavori futuri da svolgere all'interno del framework, la necessità di poter implementare soluzioni alternative a quella proposta in questa tesi, e i limiti che devono essere superati affinché il framework possa essere utile in un ambiente reale e non solo di testing virtualizzato. Infine verranno 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>