# IKEV2 TESTING

DAVIDE DE ZUANE & RAHMI EL MECHRI

# CONTENTS



# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

# 1 INTRODUCTION

La sicurezza negli ultimi anni sta diventando sempre più importante per le comunicazioni,

c'è sempre più bisogno di realizzare collegamenti sicuri, uno strumento molto utile a questo fine sono le VPN.

## 1.1 IPsec

Introduzione a quello che è ipsec e suo funzionamento

## 1.2 IKE

Importanza di IKE per effettuare lo scambio di chiavi per stabilire la SA.

Concetti su cui si basa IKE

## 1.3 Strongswan

Parlare rapidamente di quello che è quello che è anche l'architettura di strongswan charon.

# 2 SETUP

Descrivere come si è deciso di effettuare il testing. Vediamo anche quelle che sono le convenzioni che abbiamo utilizzato nella nostra sperimentazione.

## 2.1 Environment

Confiugrazione delle macchine virtuali utilizzate con KVM. quantità di ram e cpu assegnata, sistema opeartivo su cui si è effettutato il testing e in particolare la rete in modalità network bridge.

## 2.2 Configuration

```
/etc
├── ipsec.conf
├── ipsec.secret
└── ipsec.d
    ├── cacerts
    ├── certs
    └── private
```

### *Certificati*

IKEv2 supporta l'autenticazione tramite certificati, per poter avere dei certificati con delle firme valide occorre generare dei certificati da Root Autority. Per i test che andremo ad eseguire ci servono due certificati Root CA:

- un certificato RSA
- un certificato ECDSA

La caratteristica dei certificati Root CA è che sono autofirmati. Spiegare il perchè

Dunque procediamo con il generare la chiave privata della Root CA:

```
$ pki --gen --type rsa --size 2048 --outform pem > 'ca.rsa.key.pem'
$ pki --gen --type ecdsa --size 256 --outform pem > 'ca.ecdsa.key.pem'
```

Andiamo a creare i certificati di chiave pubblica andando ad applicare la chiave private appena generata:

```
$ pki --self --ca --lifetime 3650 --in 'ca.<type>.key.pem' --type <type> --
    dn "CN=Root CA" --outform pem > ca.<type>.cert.pem
```

Generazione dei certificati, non si possono utilizzare i certificati presenti nel repository perchè sono legati all'IP del responder e dell'initiator.

```
$ pki --gen --type ecdsa --size 256 --outform pem > '/etc/ipsec.d/private'
```

#### 2.2.1 *Mschap*

#### 2.2.2 *RSA*

#### 2.2.3 *ECDSA*

## 3 TESTING

### 3.1 Automating Test

### 3.2 Results

## 4 CONSLUSIONI

## REFERENCES