Università degli Studi di Milano Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea Magistrale in Informatica

Sistemi e Servizi di Telecomunicazione

Prestazioni dei firewall realizzati con appliance dedicate: Dipendenza dal numero di regole e ottimizzazioni

Autore:
Michele Salanti 793091

Anno Accademico 2019–2020

Indice

Capitolo 1

Introduzione

L'attuale infrastruttura di rete si è stabilizzata per oltre un decennio e si è radicata profondamente nel contesto della società umana. Tuttavia, le imprese ed i fornitori di servizi stanno iniziando a realizzare i forti limiti e l'inflessibilità dello stato attuale dell'infrastrutura con le tecnologie di rete in rapida evoluzione e le crescenti richieste da parte degli utenti.

In altre parole si è riconosciuta la necessità di ristrutturare l'attuale architettura con qualcosa di più dinamico e flessibile [1].

1.1 Software-Defined Network

Come risultato, per superare queste limitazioni intorno al 2005 è stata proposta la *Software-Defined Network* (SDN). SDN è un concetto di rete che sta acquistando tendenza negli ultimi anni, il quale *disaccoppia* il *piano di controllo* e il *piano dati*, *centralizzando* così l'intelligenza di rete in un *controller*.

L'idea di separare il controllo e il piano dati è nata principalmente con l'intenzione di permettere di sviluppare ciascuna parte in modo completamente indipendente l'una dall'altra, in modo che il software non sia vincolato dalla limitazione dell'hardware. Ed eventuali amministratori di rete possano orchestrare la rete per mezzo di applicazioni di alto livello, semplificandone la gestione [2][1].

In altre parole si abbandona completamente la concezione verticale e monolitica di apparati di rete "tutto-in-uno", con l'intero stack (sia hardware e software) fornito da un'unica compagnia.

In caso di cambiamenti alla rete, anziché dover configurare manualmente ogni singolo switch in conformità di questi cambiamenti, è possibile utilizzare switch programmabili controllati da un'unica applicazione esterna.

1.2 OpenFlow

Il componente più importante della SDN è un protocollo di comunicazione *tra il piano di controllo ed il piano dati*. Il ***protocollo OpenFlow*** è stato introdotto a tale scopo, supportato dalla *Open*

Networking Foundation (ONF); allo stato attuale, sono disponibili controller SDN open source che implementano OpenFlow, i più noti sono: *NOX*, *POX*, *Ryu* e *Floodlight* [1].

1.3 Obiettivo

Vista la nativa caratteristica delle SDN di effettuare azioni sul flusso del traffico, in particolare azioni come *scartare* pacchetti (drop) o *permetterne il passaggio* ed il diffuso utilizzo del protocollo OpenFlow, in questa relazione andremo a vedere alcune funzioni di firewall implementate tramite questa relativamente *"nuova"* (2005!) architettura e protocollo ed ai possibili vantaggi e risvolte che è possibile avere sia nella sicurezza che nella gestione di reti.

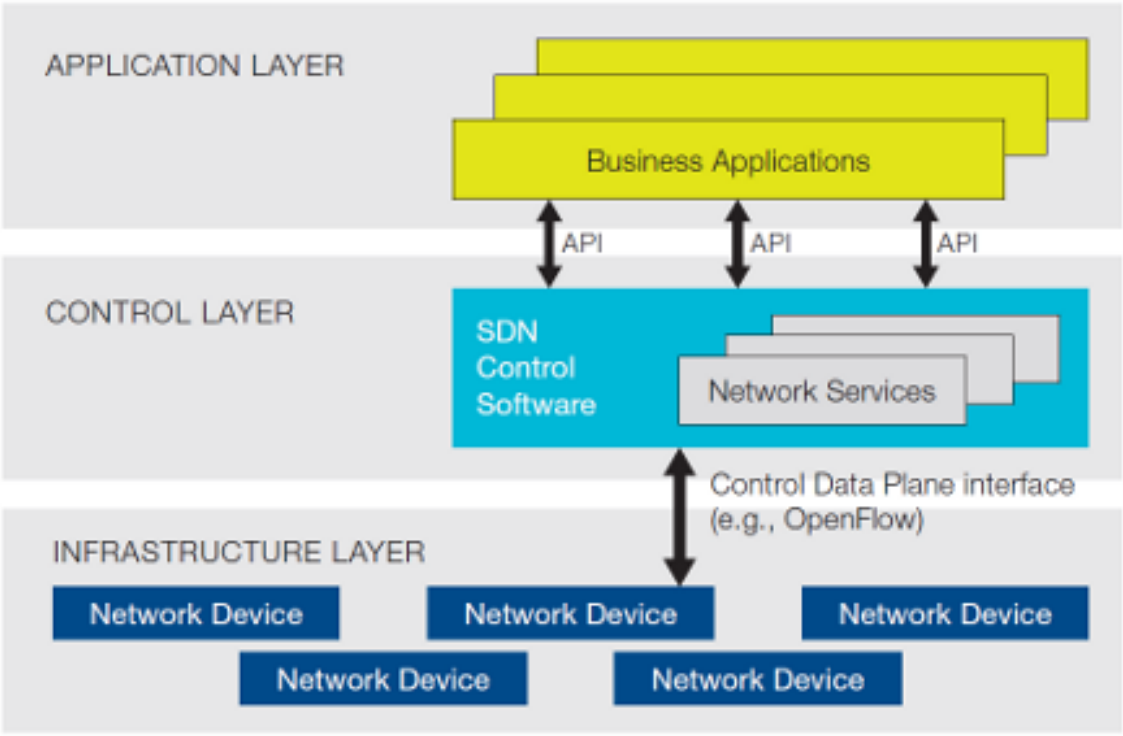


Figura 1.1: Paradigma SDN

Capitolo 2

Implementazione

2.1 Capisaldi di Design

Sono stati fissati ***tre punti chiave*** che devono essere rispettati durante l'implementazione del firewall virtuale [3]:

- Un firewall basato su OpenFlow deve essere in grado di ***filtrare il traffico in modo simile ad un firewall tradizionale***, tramite la definizione di regole di flusso.
- Dovrebbe essere possibile ***distribuire le regole su un'intera rete senza la necessità di configurare gli switch uno alla volta***; ciò affronta il problema di proteggere un'intera infrastruttura di rete da minacce sia interne che esterne.
- Poiché non si può presumere che le reti siano omogenee, dovrebbero essere implementate ***su una rete politiche di sicurezza eterogenee***; ciò richiede l'implementazione di un meccanismo che consente a diverse parti di una rete di filtrare diversi flussi di traffico (vedasi nei paragrafi successivi).

2.2 Componenti principali

Per poter implementre le proposte citate nella sezione **??**, è stato sviluppato `ACLSwitch`[1], un'applicazione sviluppata tramite il controller *framework Ryu* che si propone come firewall virtuale distribuito su tutta una determinata rete.

Altri componenti importanti sono le ***Access Control List*** (ACL) e le ***Policy Domains*** (Politiche di Dominio). L'implementazione effettuata ad-hoc delle ACL eredita tutti i concetti dei firewall tradizionali; mentre l'inclusione delle Politiche di Dominio *estende* le attuali capacità degli

[1] `https://github.com/bakkerjarr/ACLSwitch`

attuali Firewall basati su OpenFlow, fornendo all'amministratore di rete meccanismi di distribuzioni delle regole ACL a gruppi di switch senza la necessità di configurare in maniera separata ognuno di essi. Il metodo di creare regole singole poi distribuirle a dispositivi specifici è tipico sia dei firewall tradizionali che di basati su OpenFlow. Tramite *ACLSwitch* questa funzionalità viene aumentata presentando un meccanismo per raggruppare le regole in *policy domains* che nel pragrafo seguente verranno meglio esplicate.

2.3 ACL Estese

Le capacità delle ACL sono state estese aggiungendo campi extra oltre alle solite 5 tuple (vedasi **??**) risultando come segue [3]:

$$\text{regola ACL } r = \overbrace{ip\_src, ip\_dst, tp\_proto, port\_src, port\_dst}^{\text{Le 5 tuple ACL base}}, \overbrace{policy, action, time\_start, time\_duration}^{\text{Estensioni}}$$

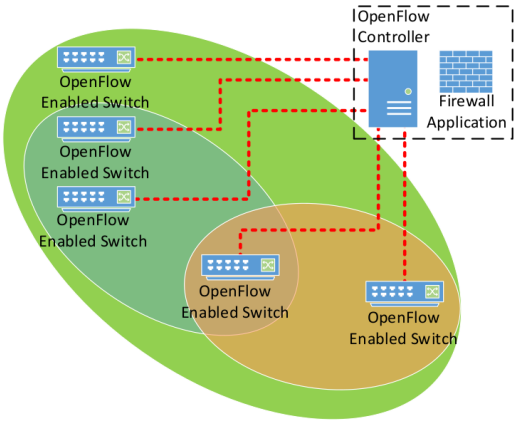
I nuovi campi si identificano come segue:

- ***policy***: è il primo campo ed identifica la politica di dominio di cui fa parte la regola.
- ***action***: indica, come suggerisce il nome, l'azione da intraprendere, per esempio se un certo traffico che matcha debba essere inoltrato o scartato.
- ***time_start*** e ***time_duration***: sono campi facoltativi ed indicano rispettivamente la data di inizio e la durata in cui la regola debba essere in vigore.

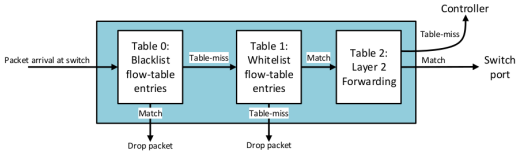
2.3.1 Policy Domains

Un policy domain rappresenta una relazione tra le regole ACL e gli switch OpenFlow. Un policy domain può contenere molte regole ACL ma ogni singola regola ACL può essere membro di un solo policy domain. Un policy domain può essere assegnato a molti switch e ad ogni switch possono essere assegnati più policy domain (Fig. 2.1a).

L'astrazione fornita dai policy domain consente a un'intera rete di essere protetta da una singola politica e, per estensione, da un singolo meccanismo. Tuttavia, offrono anche la flessibilità di assegnare diversi gruppi di regole a switch separati e, per estensione, a punti diverse della stessa rete.



(a) I policy domains (ellissi colorate) possono essere assegnati a switch per proteggere simultaneamente l'intera rete e mettere in vigore altre politiche [3]



(b) Implementazione della pipeline [3]

2.4 Deploy delle Flow Table Entries

OpenFlow supporta due stili di distribuzione delle FTE: *deploy reattivo* e *deploy proattivo*.

Il ***deploy reattivo*** viene utilizzato dal controller come risposta a un messaggio `packet-in` da uno switch. Quando si creano firewall utilizzando OpenFlow, questa modalità non è adatta a causa del non trascurabile overhead che impone al controller per elaborare decisioni sul traffico.

Questo problema viene risolto *deployando in modalità proattiva* agli switch le FTE nel momento in cui vengono create le regole ACL.

2.5 Pipeline

Come mostrato in Fig. 2.1b, la pipeline è la seguente [3]:

- `Table 0` è la prima tabella della pipeline e contiene FTE associate a bloccare flussi di traffico facenti parte di una lista nera; in altre parole i pacchetti che matchano vengono scartati. I pacchetti che corrispondono alla voce `table-miss` vengono inoltrati alla `Table 1` tramite l'azione `Goto-Table`. Pertanto, questa tabella viene utilizzata per fare una prima scrematura, bloccando flussi di traffico ***indesiderati***.

- `Table 1` è la tabella successiva nella pipeline, contiene FTE associate a far passare flussi di traffico facenti parte di una lista bianca; in altre parole i pacchetti che matchano vengono direttamente inoltrati alla `Table 2`, il resto, ossia quelli che corrispondono alla `Table-miss` vengono scartati.

- `Table 2` esegue il ruolo di un normale switch ethernet.

Capitolo 3

Note Finali

OpenFlow fornisce tutti i mezzi principali per configurare switch compatibili ed applicare le principali politiche di sicurezza. Come già accennato nel capitolo **??**, gli articoli diversi da quello di *Bakker* che l'autore di questo testo ha consultato sembrano non sfruttare appieno la flessibilità offerta da questa tecnologia; consentendo solo di configurare un solo switch o di farlo singolarmente per ognuno.

La soluzione proposta dal gruppo di *Bakker* presenta un meccanismo che combina questi due approcci in modo che gli host all'interno di una rete possano essere protetti dalle minacce ***sia interne che esterne***, facilitando al contempo il settaggio di diverse operazioni di filtraggio tra gruppi di switch diversi.

Oltretutto, ***come nota personale***, l'autore di questo testo si sente di aggiungere la concatenzione nella pipeline (cap. 2.5) di una tabella con politiche permissive succeduta da una con politiche restrittive permette di semplificare sia la logica che il numero di regole necessarie; questo, oltre a conferire un ovvio vantaggio sulla mantenibilità da parte degli amministratori di rete, permette anche di garantire prestazioni ottimali, visto che come mostra il lavoro del gruppo di *Collings* [4], sono molto legate (specialmente per quanto riguarda la latenza) al numero di regole applicate.

Bibliografia

[1] Michelle Suh, Sae Hyong Park, Byungjoon Lee, and Sunhee Yang. Building firewall over the software-defined network controller. In *16th International Conference on Advanced Communication Technology*, pages 744–748. IEEE, 2014.

[2] Open Networking Fundation. Software-defined networking: The new norm for networks. *ONF White Paper*, 2(2-6):11, 2012.

[3] Jarrod N Bakker, Ian Welch, and Winston KG Seah. Network-wide virtual firewall using sdn/openflow. In *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 62–68. IEEE, 2016.

[4] Jake Collings and Jun Liu. An openflow-based prototype of sdn-oriented stateful hardware firewalls. In *2014 IEEE 22nd International Conference on Network Protocols*, pages 525–528. IEEE, 2014.