Implementácia grafického používateľského rozhrania pre distribuovaný firewall založenom na OpenFlow

Dávid BEŇO, Šimon HARVAN

Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

Ilkovičova 2, 842 16 Bratislava, Slovakia

xbenod@stuba.sk, xharvan@stuba.sk

**Abstract.** Softvérovo definované siete (SDN) sú technológia, ktorá bude jadrom sietí ďalších generácií. Veľa spoločností a organizácií začalo používať SDN aplikácie. Toto dáva administrátorom flexibilitu pri implementovaní vlastných sieti. Ale zároveň vyvstávajú nové bezpečnostné problémy. Aby sme mohli zabezpečiť SDN siete potrebujeme silný firewall. Aktuálne už existujú firewally, ale majú určité nevýhody. Jeden z hlavných nedostatkov existujúcich riešení je, že sú umiestnené na jednom centrálnom zariadení a celý firewall zlyhá ak zlyhá jedno zariadenie. Ďalší nedostatok existujúcich riešení je, že väčšina z nich sú firewall-y druhej vrstvy. V tomto článku popisujeme implementáciu grafického používateľského rozhrania pre distribuovaný firewall, kde sa každý prepínač v sieti môže správať ako firewall. Používame RYU riadič založený na Python-e. RYU už obsahuje firewall, s ktorým sa dá komunikovať pomocou REST funkcii a takto nastavovať jeho pravidlá. Testovali sme tento firewall a k nemu nami vytvorené GUI pomocou emulátora Mininet Emulátor, ktorý je nainštalovaný v Ubuntu 14.04 nainštalovaným pod VirtualBox.

# Úvod

V tradičných sieťach je veľmi ťažké dynamicky upravovať smerovače, prepínače, vyrovnávače zaťaženia, konfigurácie IDS a IPS podľa požiadaviek organizácie. Avšak využitie SDN toto dovoľuje. SDN oddeľuje PRELOZIT:**rovinu dopredu(forwarding plane)** od riadiacej roviny. SDN umožňuje centralizovanému ovládaču dynamicky spravovať všetky zariadenia. V tradičných sieťach musí byť každé zariadenie konfigurované individuálne. Riadiaca rovina a predná rovina komunikujú pomocou protokolu OpenFlow.

Hlavnou funkciou SDN je určiť, čo sa má urobiť s paketom, ktorý bol prijatý prepínačom OpenFlow, a to konzultáciou s tabuľkou vstupov. Keď sa pakety dostanú na prepínač, polia hlavičiek paketov sa porovnajú s položkami toku tabuľky. Ak nastane zhoda, potom sa činnosť vykoná podľa špecifikácie v položke toku. Ak nenastane žiadna zhoda, tak paket bude odoslaný do riadiacej jednotky podľa vstupného prietoku. Toto sa tiež nazýva správa *Packet\_In*. Potom riadič rozhodne, čo sa má urobiť s paketom podľa aplikačnej logiky. Potom môže poveriť prepínač, aby preposielal paket alebo môže pridať tok do tabuľky toku prepínača, aby sa tak s podobnými typmi paketov mohlo v budúcnosti rýchlo pracovať.

# Analýza

## Mininet

Mininet je emulátor siete, ktorý vytvára sieť virtuálnych hostov, prepínačov, ovládačov a odkazov. Mininet hostovia prevádzkujú štandardný sieťový softvér Linux a jeho prepínače podporujú OpenFlow pre vysoko flexibilné vlastné smerovanie a softvérové siete. Spoločnosť Mininet podporuje výskum, vývoj, učenie, prototypovanie, testovanie, ladenie a všetky ostatné úlohy, ktoré by mohli mať prospech z úplnej experimentálnej siete na prenosnom alebo inom počítači.

Mininet poskytuje jednoduchý spôsob, ako dosiahnuť správne správanie systému (a v rozsahu podporovanom vaším hardvérom, výkonom) a experimentovať s topológiami. Mininetové siete pracujú so skutočným kódom vrátane štandardných sieťových aplikácií Unix / Linux, ako aj skutočného jadra Linuxu a sieťového zásobníka.

Z tohto dôvodu sa kód, ktorý vyvíjame a otestujeme v službe Mininet, pre radič OpenFlow, zmenený prepínač alebo host, môže prejsť na skutočný systém s minimálnymi zmenami, pre testovanie v reálnom svete, hodnotenie výkonnosti a nasadenie. Dôležité je to, že návrh, ktorý pracuje v službe Mininet, sa zvyčajne môže presunúť priamo na hardvérové prepínače pre presmerovanie paketov line-rate.

## Firewall

Brána firewall je sieťový bezpečnostný systém založený na hardvéri alebo softvéri, ktorý používa pravidlá na kontrolu prichádzajúcej a odchádzajúcej sieťovej prevádzky.

Brána firewall funguje ako bariéra medzi dôveryhodnou sieťou a nedôveryhodnou sieťou. Firewall kontroluje prístup k zdrojom siete prostredníctvom pozitívneho modelu kontroly. To znamená, že jediná prevádzka povolená na sieť je definovaná v pravidlách brány firewall - všetka ďalšia prevádzka bola zamietnutá.

## SDN

Softvérovo definovaná sieť (SDN) je zastrešujúcim výrazom, ktorý zahŕňa niekoľko druhov sieťových technológií zameraných na to, aby bola sieť agilná a flexibilná ako virtualizovaná serverová a ukladacia infraštruktúra moderného dátového centra. Cieľom SDN je umožniť sieťovým inžinierom a administrátorom rýchlo reagovať na meniace sa požiadavky.

V sieťach definovaných softvérom môže správca siete tvarovať prevádzku z centralizovanej ovládacej konzoly bez toho, aby sa musel dotýkať jednotlivých prepínačov a môže poskytovať služby všade tam, kde sú potrebné v sieti, bez ohľadu na to, aké konkrétne zariadenia server alebo iné hardvérové súčasti sú pripojené k sieti. Kľúčovými technológiami implementácie SDN sú funkčné oddelenie, sieťová virtualizácia a automatizácia prostredníctvom programovateľnosti.

## Ryu

Ryu kontrolór je otvorený kontrolór pre SDN siete, ktorý je navrhnutý tak, aby zvyšoval agilitu siete tým, že umožňuje jednoduchú správu a spôsob kontrolovania prevádzky. Vo všeobecnosti je kontrolór „mozog“ prostredia SDN, oznamuje informácie smerom k prepínačom a smerovačom ale aj aplikáciám a API na severi.

Ryu poskytuje softvérové komponenty s dobre definovanými aplikačnými programovými rozhraniami (API), ktoré uľahčujú vývojárom vytváranie nových aplikácií pre správu a riadenie siete. Tento prístup komponentov pomáha organizáciám prispôsobiť nasadenia tak, aby vyhovovali ich špecifickým potrebám prevádzky. Vývojári môžu rýchlo a ľahko upraviť existujúce komponenty alebo implementovať svoje vlastné, aby zabezpečili, že základná sieť môže spĺňať meniace sa požiadavky svojich aplikácií.

Zdrojové kódy ryu sú dostupné na serveri github a sú spravované Ryu komunitou. Ryu je napísaný v jazyku Python a je pod Apache 2.0 licenciou, takže je dostupný pre každého. Ryu plne podporuje OpenFlow, čo je jeden z prvých a častou používaných komunikačných štandardov. OpenFlow sa využíva práve na komunikáciu s prepínačmi a smerovačmi – pre nastavenie správy prevádzky siete.

## OpenFlow

OpenFlow je považovaný za jeden z prvých SDN štandardov. Definuje komunikačný protokol v prostredí SDN, ktorý umožňuje SDN kontroléru priamo komunikovať s prepínačmi a smerovačmi, fyzickými aj virtuálnymi, aby sa mohli lepšie prispôsobiť meniacim sa obchodným požiadavkám.

## REST API

REST (Representational state transfer) je webový štandard založený na architektúre a používaní protokolu HTTP pre komunikáciu. REST navrhol v roku 2000 Roy Fielding (spoluautor protokolu HTTP) v rámci svojej dizertačnej práce. Rozhranie REST je použiteľné pre jednotný a jednoduchý prístup k zdrojom. To môžu byť dáta rovnako ako napríklad stavy aplikácie. Na rozdiel od iných prístupov ako napríklad SOAP nie je REST orientovaný procedurálne ale dátovo. Každý zdroj má vlastný identifikátor URI a REST definuje štyri základné metódy na prístup k nim – CRUD (Create, Read, Update, Delete). Konkrétne sa jedná o HTTP metódy GET, PUT, POST, DELETE.

# Grafické používateľské rozhranie

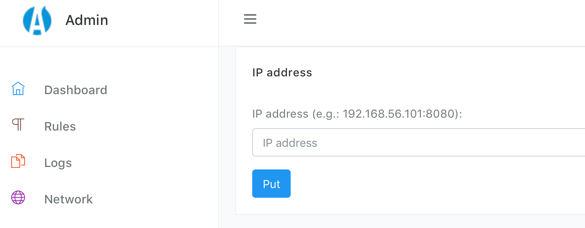
Pre lepšiu vizualizáciu aktuálneho stavu firewallu na prepínačoch, bola vytvorená grafická aplikácia. Je určená pre akýkoľvek počítač pripojený k stroju, na ktorom je spustený kontrolér siete. Aplikácia je založená na farmeworku Bootstrap a všetky dependencies sú manažované balíčkovým manažérom NPM. Primárnym cieľom frameworku Bootstrap je responzívnosť a použiteľnosť aplikácií na rôznych platformách.

Nami pripravené web rozhranie bude mať nasledovné funkcie:

* Získanie IP stroja, na ktorom je spustený kontrolér siete.
* Prezeranie statusu všetkých firewall prepínačov
* Zapnutie/Vypnutie firewall prepínaču
* Prezeranie statusu logovania všetkých firewall prepínačov
* Zapnutie/Vypnutie logovania firewall prepínaču
* Prezeranie firewall pravidiel všetkých prepínačov
* Pridanie pravidla pre špecifický prepínač
* Odobranie pravidla pre špecifický prepínač

**Získanie IP stroja, na ktorom je spustený kontrolér siete**

Pri prvom spustení aplikácie, používateľ zadá IP adresu stroja, na ktorom beží controlér siete a spravuje prepínače a routre pomocou protokolu OpenFlow.

****

**Prezeranie statusu všetkých firewall prepínačov**

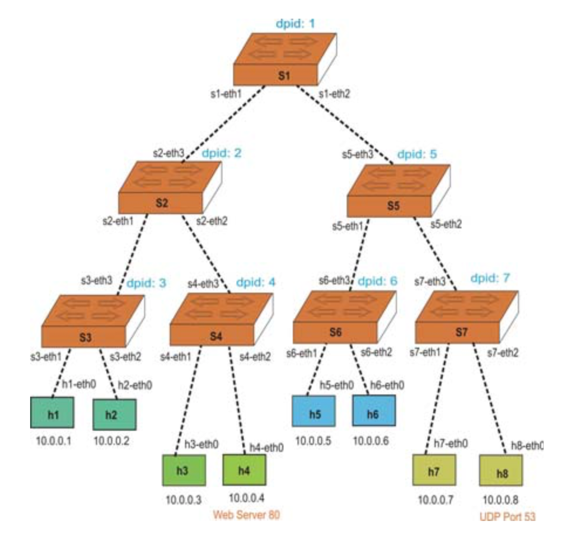
Po štarte aplikácie sa zobrazí zoznam všetkých prepínačov v sieti a aktuálny stav firewalu na nich. Firewall na každom prepínači sa dá zapnúť alebo vypnúť. Tiež sa dajú zapnúť/vypnúť firewally na všetkých prepínačoch naraz.

**Display test results and statistics**

When the testing is finished, the GUI-based application provides results and statistics of the testing. These statistics show a number of requests for allocation (in detail), a number of successfully allocated memory blocks (this means a number of successful requests), a number of unsuccessful requests, relative fragmentation and utilization of the memory.

# Experiment

Ako testovacie prostredie sme použili Ubuntu 16.04 spustené pomocou programu VirtualBox. Pre simulovanie SDN siete sme použili simulovací nástroj Mininet. Ako riadič pre SDN sieť sme si vybrali RYU a to z dôvodu, že už obsahuje firewall, pre ktorý sme dorobili grafické používateľské rozhranie. Pre experiment sme vytvorili stromovú topológiu príkazom:

sudo mn --topo tree,3 --mac --switch ovsk protocols=OpenFlow13 --controller remote -x

Vytvorená sieť pozostáva zo siedmich prepínačov a ôsmych hostiteľských počítačov (staníc?). Každá stanica má pridelenú IP v rozsahu od 10.0.0.1 po 10.0.0.8. Ďalej na stanici Host 4 je spustený web server na port 80 pomocou programu „SimpleHTTPServer“ a na stanici Host 8 je spustený UDP server pomocou programu „nc“.

- size of requested block

- success rate of the request

- request satisfying time

- real size of the allocated block

- the beginning address of the block

# Testovanie

An advanced functionality of our platform is comparing several allocation algorithms to each other. Each of these algorithms will be tested by several tests. These tests are divided to several levels, where each level has a specific scenario.

1. level – testing basic functionality

2. level – testing fragmentation with requests for small blocks

3. level – testing fragmentation with requests for large blocks

4. level – stress testing

The first level tests are testing if all functions of the memory allocation algorithm are working properly and are ready for next tests.

The second and third level tests are testing memory fragmentation. These tests are trying to find out how long it takes to fail any request due to the fragmentation. The difference between these two levels is that the second level contains requests for small size blocks. On the other hand, third level contains requests for large blocks.

The fourth level is stress testing and shows how the allocator handles many various requests. There are many requests for different size blocks of memory. The aim of this test is to ballast system with the most fragmentation as it is possible, so it is hard to allocate free block of memory and satisfy request. Result of this test shows how much time does it take to fail any request for due to the fragmentation.

# Záver

Distribuovaný firewall funguje správne a je schopný zvládnuť prevádzku protokolu ICMP, TCP a UDP. Do všetkých prepínačov alebo konkrétnych prepínačov môžeme vložiť pravidlá na základe našich požiadaviek. To znamená, že GUI funguje tiež správne a správne komunikuje s jednotlivými prepínačmi. Budúca práca môže zahŕňať testovanie tohto firewallu na skutočnom hardvéri.

References

1. Hu, Fei, Qi Hao, and Ke Bao, "A survey on software-defined network and openflow: from concept to implementation," IEEE Communications Surveys & Tutorials 16, no. 4 (2014): 2181-2206.