Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Implementácia firewallu založeného na Openflow protokole

Architektúra komunikačných systémov

Beňo Dávid

Harvan Šimon

2017/2018

FIIT-IT

Obsah

1. Úvod

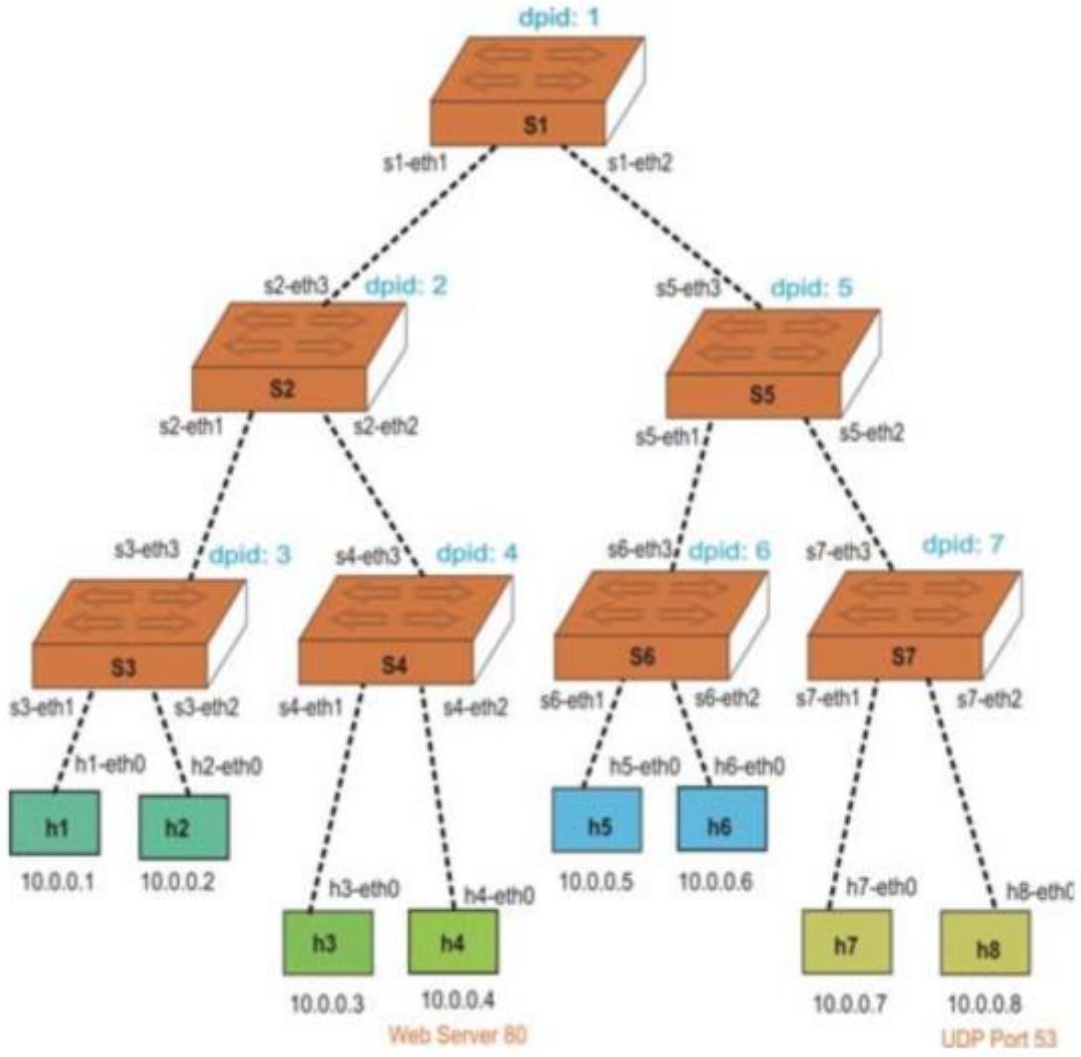
Softvérovo definované siete (SDN) sú technológia, ktorá bude jadrom sietí ďalších generácií. Veľa spoločností a organizácií začalo používať SDN aplikácie. Toto dáva administrátorom flexibilitu pri implementovaní vlastných sieti. Ale zároveň vyvstávajú nové bezpečnostné problémy. Aby sme mohli zabezpečiť SDN siete potrebujeme silný firewall. Aktuálne už existujú firewally, ale majú určite 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 implementuje distribuovaný firewall, kde sa každý OpenFlow prepínač v sieti môže správať ako firewall. Navyše bude tento firewall zvládnuť prevádzku protokolu TCP, UDP a ICMP. Testovali sme tento firewall pomocou emulátora Mininet, ktorý je nainštalovaný v Ubuntu 14.04 nainštalovaným pod VirtualBox. Používame POX riadič založený na Pythone. Táto práca je rozšírením našej predchádzajúcej práce na programovateľných firewall-och.

2. Analýza

V tejto kapitole je zhrnutá analýza článku a popísané technológie, s ktorými budeme v tomto projekte pracovať.

2.1 Analýza článku

Článok hovorí o firewalle v SDN sieti v ktorej by nepostačilo jedno zariadenie s firewallom, ako je možné vidieť na obrázku 1. Toto je jedna z nevýhod existujúcich firewall riešení, kde zabezpečenie môže byť prelomené ak zlyhá jedno zariadenie.



Obrázok 1.: Topológia siete

Autori v článku navrhli riešenie, kde sa každý OpenFlow prepínač môže správať ako firewall, môže mať vlastné pravidlá a môže sám kontrolovať prevádzku.

Experiment navrhli na emulátore Mininet na ktorom je možné emulovať akýkoľvek druh siete. Využili POX riadič, ktorý bol rovnako ako ich riadič napísaný v Python-e. V ich experimente vytvorili stromovú topológiu s 2 koncovými zariadeniami pripojenými na 1

prepínač a každé 2 prepínače do ďalšieho prepínača rovnako ako na obrázku. Všetky koncové zariadenia mali IP adresu od 10.0.0.1 po 10.0.0.8

Testovanie pozostávalo zo 4 testovacích scénarov, ktoré mal byť firewall schopný zvládnuť.

- h1 by nemalo byť schopné pingnúť h2
- h1 by nemalo byť schopné pingnúť h7
- h1 by nemalo byť schopné pristúpiť na web server h4
- h7 by nemalo byť schopné pristúpiť na h8 cez UDP port 53

Z článku je jasné, že centralizované riešenie (firewall na jednom zariadení) nie je možné použiť, lebo by nepokryl všetky scenáre. Toto zabezpečili rozmiestnením firewallu na rôzne zariadenia. Konkrétne použili na každom prepínači prislúchajúce pravidlo a teda nepreťažovali prepínače pravidlami.

Testovanie prebehlo pomocou príkazu pingall, curl a nc, ktorými zistili, že ich riešenie funguje na všetkých zvolených scenároch.

V článku úspešne implementovali a otestovali ich navrhnuté riešenie. Ďalšou prácou, môže byť navrhnutie grafického rozhrania.

2.2 Technológie

V tejto kapitole sú popísané technológie, ktorými budeme v našom projekte pracovať.

2.2.1 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.

2.2.2 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.

2.2.3 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.

2.2.4 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.

2.2.5 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.

2.2.6 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.

3. Návrh riešenia

V našom zadaní sa budeme držať postupov v článku. Autori článku, nezverejnili zdrojové súbory firewallu, ktorý použili, takže miesto nich použijeme Ryu firewall. Experiment postavíme na nasledovných komponentoch:

- Mininet network emulator (verzia 2.2.1)

- Ryu - Firewall
- Ryu - Controller
- REST API - Ryu
- Web rozhranie

3.2 Príprava experimentu

Experiment budeme robiť na emulovanej sieti v emulátore Mininet. Vytvoríme sieť so stromovou topológiou. Na vytvorené zariadenia použijeme Ryu controller a firewall.

Nastavenie parametrov bude možné cez webové rozhranie pripojené k sieti, ktoré vytvoríme my. Nastavenia firewallu, ktoré budú napĺňať naše testovacie scenáre, sa budú dať vykonať cez naše webové rozhranie. Naša aplikácia bude s firewallom komunikovať cez REST API.

V tejto sieti sme si pripravili, podobne ako v článku, testovacie scenáre, ktoré by mal byť náš firewall schopný obslúžiť.

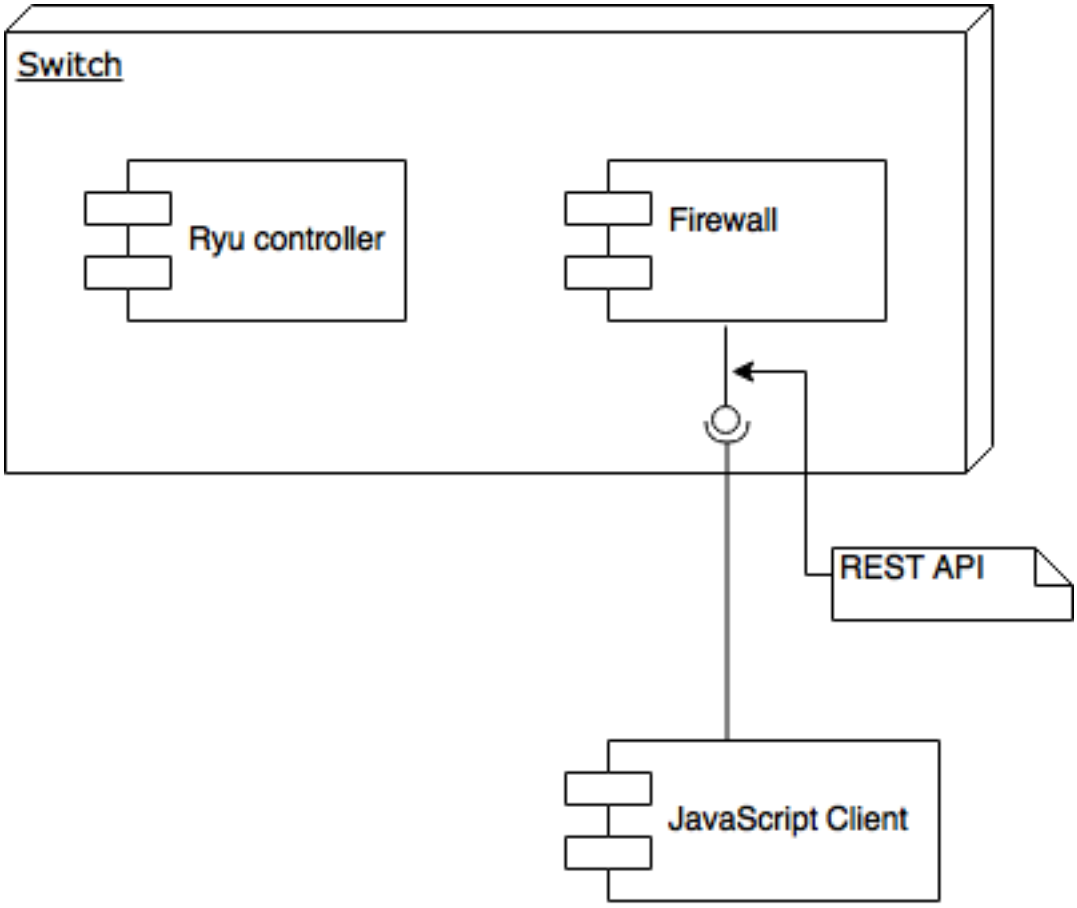
Náš testovací scenár je:

- h1 by nemalo byť schopné pingnúť h2.
- h1 by nemalo byť schopné pingnúť h7.
- h1 by nemalo byť schopné pristúpiť na web servér h4.
- h7 by nemalo byť schopné pristúpiť na udp port 53 na stroji h8.

3.3 Funkcie firewallu

Nami pripravené web rozhranie bude mať nasledovné funkcie:

- 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



Obrazok 2.: Diagram komponentov