باسمه تعالی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و
فناوری اطلاعات

فرم تعریف پروژه

فارغ التحصیلی دوره کارشناسی

تاریخ: .....

شماره: .....

عنوان پروژه: طراحی و پیاده‌سازی vTAP برای مانیتورینگ ماشین های مجازی در شبکه‌های مبتنی بر نرم‌افزار
استاد راهنمای پروژه: دکتر سیاوش خرسندی امضاء:
مشخصات دانشجو: **نام و نام خانوادگی: علیرضا آقایی** گرایش: **نرم‌افزار** **شماره دانشجویی: 9431054** ترم ثبت نام پروژه: **دوم 97-98**
داوران پروژه: **۱-** امضاء داور: **۲-** امضاء داور:
شرح پروژه (در صورت مشترک بودن بخشی از کار که بعهده دانشجو می باشد مشخص شود): **در این پروژه به طراحی و پیاده‌سازی vTAP برای شبکه‌های مبتنی بر نرم‌افزار می‌پردازیم. vTAP یک TAP نرم‌افزاری شده است که قابلیت مشاهده‌پذیری ترافیک ماشین‌های مجازی یا کارکرد شبکه مجازی در محیط‌های محاسبات مجازی را جهت مانیتورینگ شبکه‌ها و برنامه‌ها به دست می‌دهد. بدین ترتیب vTAP می‌تواند در سناریوهایی همچون رؤیت‌پذیری ترافیک محیط‌های مجازی، تعیین گلوگاه‌های سرعت در سطح ماشین مجازی یا کارکرد شبکه مجازی، تشخیص ناهنجاری و Network Forensics به کارگرفته شود.**
وسائل مورد نیاز: **- امکان دسترسی به مقالات و کتب مرتبط** **- یک دستگاه کامپیوتر دارای دسترسی به اینترنت**
محل انجام پروژه: دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر **تاریخ شروع: اسفند ۱۳۹۷**

این قســـمت توســـط دانشـــکده تکمیـــل می‌گـــردد:

تاریخ تصویب در گروه:	**اسم و امضاء:**
تاریخ تصویب در دانشکده:	**اسم و امضاء:**
اصلاحات لازم در تعریف پروژه:	

توجه: پروژه حداکثر یک‌ماه و نیم پس از شروع ترمی که در آن در درس پروژه ثبت نام به عمل آمده است باید به تصویب برسد.

نسخه۱ – دانشکده	نسخه ۲ - استاد راهنما	نسخه ۳ – دانشجو

مقدمه

طی دهه گذشته، رایانش ابری[1] به طرز گسترده‌ای مورد استفاده قرار گرفته‌است تا بتواند گستره زیادی از کاربردها را در بستر اینترنت فراهم کند. عامل مهم و کلیدی در استفاده از رایانش ابری، فناوری مجازی‌سازی سرور[2] می‌باشد که اجازه استفاده کارآمد از منابع محاسباتی را در مراکز داده[3]، آن هم در کنار انعطاف‌پذیری و چابکی در توسعه و مدیریت نرم‌افزار می‌دهد. در چنین محیطی، به منظور پیاده‌سازی سریع و کارآمد یک سرویس، ترافیک زیادی بطور همزمان بین ماشین‌های مجازی در یک میزبان و همچنین ماشین‌های مجازی در میزبان‌های مختلف جابجا می‌شود.

بعنوان نسخه نرم‌افزاری شده دستگاه‌های TAP[4]، مفهوم و همچنین دستگاه vTAP[5] به منظور رؤیت‌پذیری ترافیک بین ماشین‌های مجازی در یک محیط مجازی‌سازی سرور شکل گرفته‌است. یک دستگاه TAP روی یک لینک فیزیکی بین سرور و سوئیچ، یا بین سوئیچ و مسیریاب[6] بسته می‌شود تا با تولید مجدد سیگنال بسته‌های عبوری، امکان فرستادن آن‌ها به یک دستگاه مانیتورینگ مرکزی را جهت بررسی(بعنوان مثال، تحلیل ترافیک جهت تشخیص وقوع حمله در موارد امنیتی و یا برطرف کردن نیازهای کیفیت سرویس) فراهم کند. دستگاه مانیتورینگ مرکزی، بسته‌های کپی‌شده تمامی TAPها را بصورت تجمیع شده[7] در اختیار دارد.

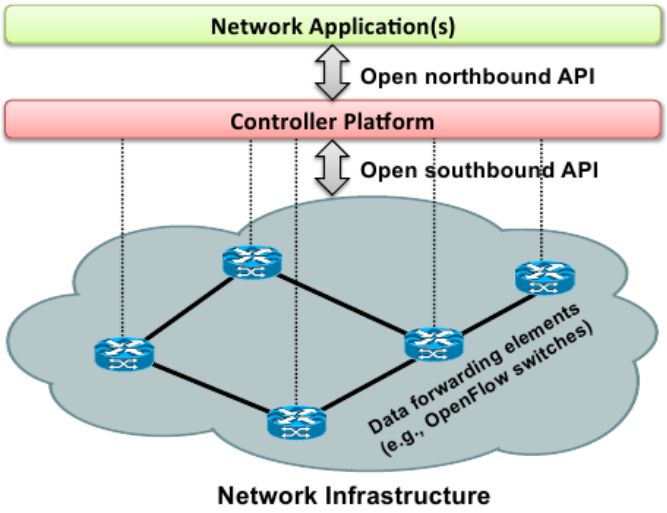
مشکل اینجاست که در شبکه مراکز داده با امکان مجازی‌سازی سرور، TAPهای سخت افزاری نمی‌توانند جهت مانیتورینگ ترافیک بین ماشین‌های مجازی در یک میزبان به کار بروند، از طرفی نبود TAP نیز سرویس‌های به نسبت مهمی را با اختلال مواجه می‌سازد. بدین منظور و برای حل این مشکل، ما طراحی و پیاده‌سازی یک vTAP را مبتنی بر سوئیچ مجازی[8] انجام خواهیم داد. در قسمت‌های بعد توضیحات بیشتری درباره ابزارهای پیاده‌سازی vTAP و همچنین نحوه تست و ارزیابی سامانه داده شده است.

---

[1] Cloud Computing
[2] Server Virtualization
[3] Data Centers
[4] Test Access Point
[5] Virtual Test Access Point
[6] Router
[7] Aggregated
[8] Virtual Switch

شرح پروژه

با تکامل شبکه‌های مبتنی بر نرم‌افزار[9] و همچنین مجازی‌سازی کارکردهای شبکه[10]، شرکت‌های ارتباطات راه دور و تأمین‌کنندگان سرویس‌های ابری سعی بر آن دارند که عملیات شبکه را که به حالت عادی و سنتی، صفحه داده[11] و کنترل[12]شان هر دو در یک دستگاه جمع می‌شد را بصورت مجازی شده پیاده‌سازی کنند. منظور از صفحه کنترل بخشی از شبکه است که تصمیم می‌گیرد ترافیک بخش‌های مختلف شبکه چگونه هندل شوند و صفحه داده، قسمتی از شبکه است که ترافیک را مطابق تصمیم‌گیری که صفحه کنترل برایش انجام داده ارسال می‌کند. در یک شبکه سنتی، مثال صفحه داده میتواند بخش‌های فیزیکی یک مسیریاب باشد که از صفحه کنترل یعنی پیکربندی های دستگاه فرمان می‌گیرد. در تصویر 1 این بخش های فیزیکی مسیریاب با عنوان Data forwarding elements مشخص شده است. ضمنا در شبکه‌های مبتنی بر نرم‌افزار Northbound API واسطی است که ارتباط بین کنترلر و برنامه‌های کاربردی (نرم‌افزار) در سطح شبکه را برقرار میکند. در نهایت Southbound API هم واسطی است که ارتباط بین کنترلر را با المان‌های صفحه داده در زیرساخت شبکه (سخت‌افزار) برقرار می سازد. کلمه Open برای API به این معناست که این API ها قابلیت دسترسی عمومی[13] دارند. مطالب گفته شده، در تصویر 1 قابل مشاهده است:



تصویر 1- معماری کلی یک شبکه مبتنی بر نرم‌افزار [1]

بنابراین در شبکه‌های سنتی صفحات داده و کنترل، هر دو در یک دستگاه سخت‌افزاری(بعنوان مثال دستگاه مسیریاب) قرار دارند. در شبکه های مبتنی بر نرم‌افزار، این دو صفحه از هم جدا شده‌اند بصورتی که صفحه کنترل به جای این که در همان دستگاه پیاده‌سازی شود، بعنوان یک واحد منطقی و نرم‌افزاری در بیرون از صفحه داده پیاده‌سازی می‌گردد. این مطلب، به ما امکان برنامه‌نویسی برای شبکه و اعمال سیاست‌های ارسال را به گونه ای انعطاف‌پذیرتر از آن چه در شبکه‌های سنتی با آن مواجه بودیم خواهد داد.

---

[9] Software Defined Networks
[10] Network Function Virtualization
[11] Data Plane
[12] Control Plane
[13] Public Availability

بطور کلی vTAP یک راه‌حل مبتنی بر نرم‌افزار برای گرفتن و یا کپی‌کردن داده جاری بین ماشین‌های مجازی می‌باشد. vTAP به رؤیت‌پذیری واضح ترافیک درون یک ماشین مجازی یا ترافیک بین دو ماشین مجازی مختلف کمک می‌کند. این امر آن‌ها را قادر می‌سازد تا داده‌های مربوط به یک ماشین مجازی را کپی یا فیلتر کنند یا حتی داده‌های کپی‌شده را به ابزارهای مانیتورینگ فیزیکی یا مجازی ارسال نمایند.

شرکت‌های IXIA و Gigamon دو مورد از شرکت‌هایی هستند که تاکنون نسخه‌ای تجاری از محصول vTAP خود عرضه کرده‌اند. این نسخه‌های تجاری از طرفی با کارایی بالا و پیاده‌سازی اختصاصی برای شرکت تولیدکننده، توانسته‌اند تاکنون مزایای خود را در حوزه صنعت اثبات کنند. اما از سویی دیگر در سمت مشتریان، وابستگی پیاده‌سازی به شرکت تولیدکننده، دشواری در نگهداری[۱۴] و همچنین سخت تر بودن قابلیت ارتقاء[۱۵] بعلت انحصاری بودن محصول، از مشکلات عمده محسوب می‌شوند.

برای همین، در این پروژه سعی بر آن شده تا با استفاده از بسترهای متن باز حال حاضر از قبیل Open vSwitch، کنترلر ONOS و همچنین پلتفرم OpenStack یک پیاده‌سازی قابل ارتقاء، برنامه‌ریزی شونده[۱۶] و سازگار[۱۷] از محصول vTAP صورت گیرد.

بسیاری از راه حل‌های vTAP منابع زیادی از سیستم را برای مانیتورینگ کارکردهای شبکه مجازی مصرف می‌کنند. راهکارهایی نظیر port mirroring مصداق راه‌حل هایی هستند که هرچند ساده، ولی در عمل مصرف زیاد منابع، ما را از استفاده از آنها بازمی‌دارد. استفاده از DPDK[18] در صفحه داده موجب تسریع برخی از کارکردهای شبکه مجازی می‌شود. در این پروژه رویکرد اصلی استفاده از کنترلر ONOS[19] به همراه پروتکل OpenFlow برای صفحه کنترل می‌باشد و محدودیتی روی صفحه داده اعمال نکرده‌ایم. هر چند پیاده‌سازی مجدد پروژه با استفاده از DPDK برای ارتقای صفحه داده می‌تواند به عنوان قدم بعدی پروژه تلقی شود که به کاربردهای واقعی نیز نزدیکتر است.

در صورت استفاده از DPDK، Open vSwitch در user mode کار می‌کند. این امر نسبت به حالت عادی که Open vSwitch در kernel mode است، باعث می‌شود لایه‌های سنگین پشته شبکه در kernel دور زده شود و ارتباط مستقیم با سخت‌افزار مربوط به networking برقرار گردد. همچنین بعلت استفاده از Hugepage ها در DPDK که اندازه‌شان از 2MB تا 1GB متغیر است، تعداد memory page های کمتری نسبت به حالت استاندارد (با اندازه معمولا 4KB) مورد نیاز خواهد بود. این امر، موجب کاهش TLB[20] miss و افزایش سرعت عملکرد Open vSwitch خواهد شد. [2]

مهم ترین مورد از موارد استفاده[۲۱] vTAP تقویت دفاع در برابر حملات امنیتی است. vTAP ها، بهترین سد دفاعی در برابر تهدیدات محیط‌های مجازی به حساب می‌آیند. چرا که تشخیص مخاطرات امنیتی بدون بررسی ترافیک بین دو ماشین مجازی جهت بررسی حملات امکان‌پذیر نمی‌باشد. استفاده از سناریوهای ایجاد حملات ساختگی و بررسی تشخیص آنها توسط سیستم مانیتورینگ[۲۲] با استفاده از Framework های آماده جهت انجام این کار از قبیل pytbull و Suricata می‌تواند سامانه vTAP پیاده‌سازی شده را در محیطی نزدیکتر به محیط واقعی مورد ارزیابی و تست قرار دهد و در کنار استفاده از DPDK گامی فراتر تلقی شود.

---

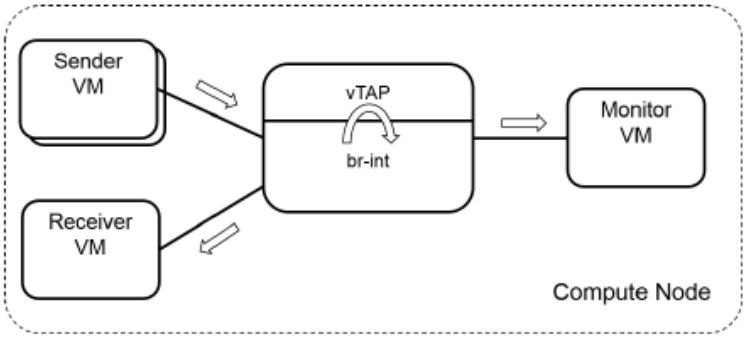
[14] Maintenance
[15] Expandability
[16] Programmable
[17] Compatible
[18] Data Plane Development Kit
[19] Open Network Operating System
[20] Translation Lookaside Buffer
[21] Use Cases

[22] سیستم ماتیتورینگ می‌تواند یک سیستم فیزیکی یا یک ماشین مجازی باشد.

روش تست و ارزیابی سامانه

برای تست سامانه، آنرا بین دو ماشین مجازی قرار میدهیم که یکی از آنها فرستنده و دیگری گیرنده است. سامانه باید بتواند سیاست‌های TAP را بدرستی روی جریان داده از فرستنده به گیرنده پیاده کند. دو سیاست اصلی TAP شامل فرستادن خود ترافیک به گیرنده و فرستادن ترافیک کپی شده به سیستم مانیتورینگ است. جمع‌آوری ترافیک در سیستم مانیتورینگ اقدام اصلی این پروژه میباشد. همانطور که در بخش قابلیت‌ها و همچنین موارد استفاده vTAP نیز ذکر شد این امر میتواند توسط DPDK تسریع شده و همچنین توسط بسترهای ایجاد و تشخیص حمله در محیط‌های نزدیک‌تر به محیط‌های واقعی تحلیل شده و مورد تست و ارزیابی قرار گیرد.

پیاده‌سازی و استقرار vTAP میتواند در host و یا در ماشین مجازی انجام گیرد. محل استقرار vTAP و این که در host باشد یا در ماشین مجازی برای این پروژه اهمیتی ندارد و تنها تفاوتی که ایجاد میکند در نحوه اتصال سوئیچ مجازی vTAP به ماشین‌های مجازی مختلف است. این اتصال مستلزم استفاده از انواع مختلفی از interface ها و port های از پیش تعبیه شده سوئیچ مجازی و نیز ایجاد تنظیمات مربوطه در پیکربندی شبکه ماشین‌های مجازی در hypervisor میباشد.



تصویر 2- بستر تست پروژه به کمک Open vSwitch متصل به ماشین‌های مجازی [3]

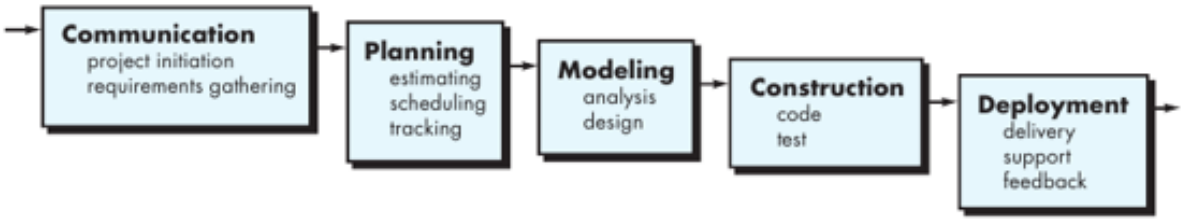
در صورت استفاده از DPDK جهت ارتقاء صفحه داده، ارزیابی سامانه vTAP تسریع شده بدینصورت است که بسته‌هایی با سایزهای مختلف، به vTAP یک بار در حالت عادی (kernel mode) و یک بار در حال استفاده از DPDK (user mode) فرستاده میشود. ارتقاء و تقویت گذردهی[۲۳] در حالت استفاده از DPDK، هم در Receiver VM و هم در Monitor VM بیانگر استقرار درست و هدفمند DPDK بعنوان صفحه داده در این سامانه خواهد بود.

[23] Throughput

رویکرد انجام پروژه

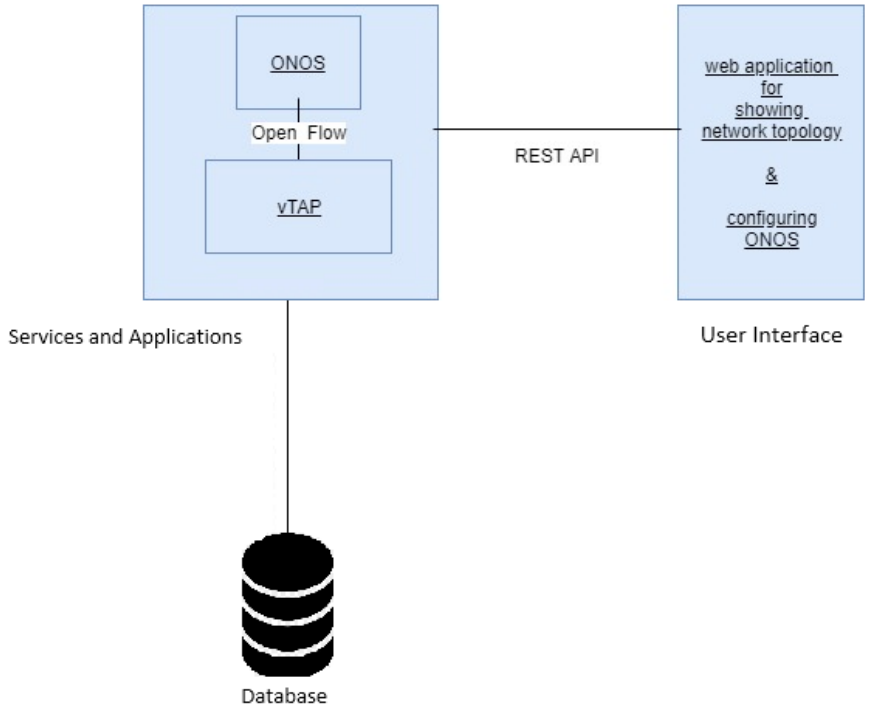
برای انجام این پروژه از مدل فرآیند waterfall استفاده خواهد شد.
دلایل استفاده از این مدل فرآیند :

- نیازمندی‌های سیستم از ابتدا مشخص می‌باشند و در اصل، این مدل waterfall است که در آن ابتدا تمامی فعالیت های پروژه قبل از شروع به کار توسعه نرم افزار برنامه‌ریزی و زمان‌بندی میشوند. [4]
- پروژه به فازهای مختلفی تقسیم شده و تا انجام یک فاز تمام نشود، فاز بعدی از پروژه شروع نمی‌گردد.



تصویر 3- مراحل مدل فرآیند waterfall [5]

شمای کلی پروژه



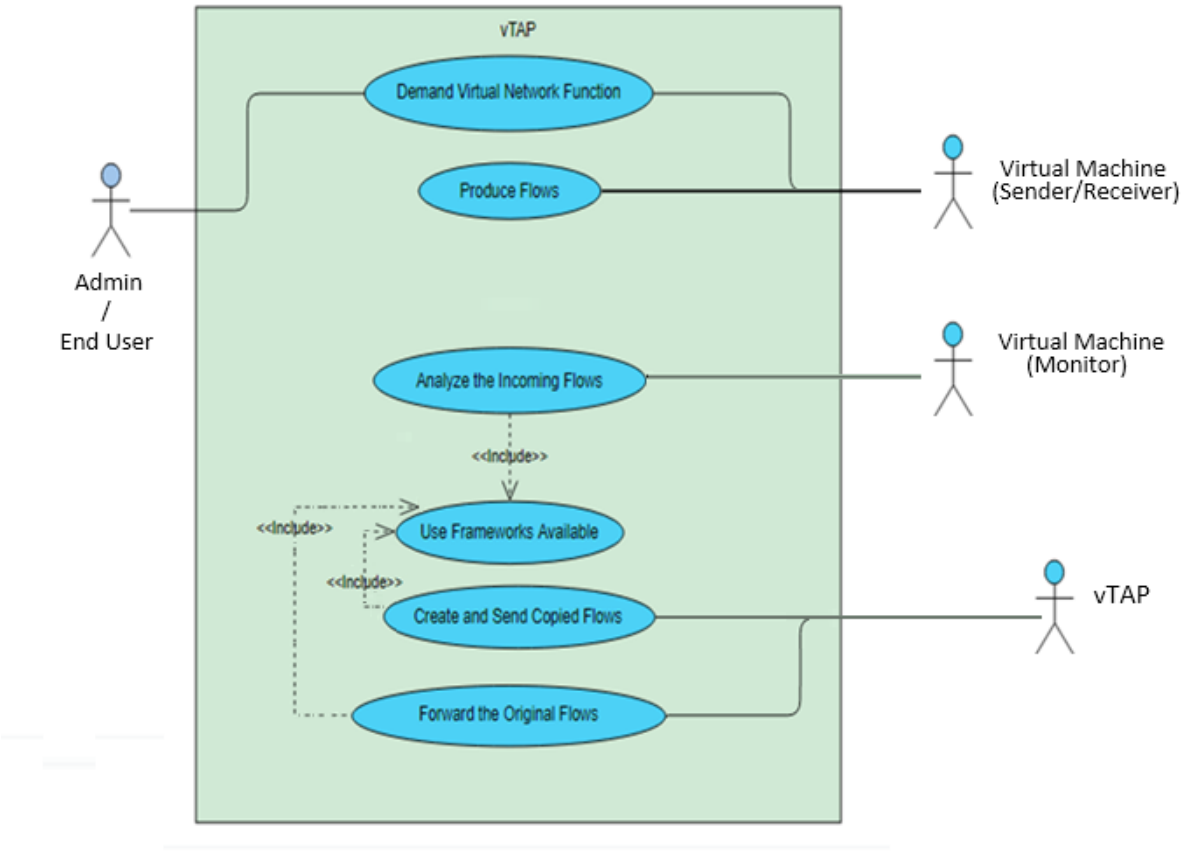
تصویر 4- شمای کلی پروژه

معماری سامانه از دو بخش اصلی User Interface و Services and Applications تشکیل شده است. بخش User Interface یک نرم‌افزار وب است که اطلاعات لازم از توپولوژی شبکه را به همراه اطلاعات پیکربندی ONOS در اختیار کاربر قرار میدهد.
همانطور که از نامش پیداست، بخش Services and Applications تمامی برنامه‌های کاربردی و سرویس هایی که برای ساخت و همچنین تست مانیتورینگ این سیستم vTAP مورد نیازند را در خود جای میدهد. این بخش، خود دو بخش عمده کوچکتر دارد که یکی کنترلر ONOS و دیگری خود vTAP میباشد. لازم بذکر است این، Open vSwitch است که پایه اصلی ساخت vTAP را تشکیل میدهد ولی بعلت کلی تر بودن vTAP نامی از Open vSwitch در این شکل آورده نشده است.

پروتکل OpenFlow پروتکل متداولی است که در Southbound API و بین کنترلر و سویچ مجازی مورد استفاده قرار میگیرد.

در پیاده‌سازی این سامانه یک پایگاه داده نیز مورد نیاز است که حاوی پایگاه داده‌های مربوط به Open vSwitch، کنترلر ONOS و همچنین برای نگهداری هرگونه اطلاعات دیگری است که در سامانه، تولید شده نیاز به ذخیره سازی دارد.

نمودار Use Case



تصویر 5- نمودار Use Case

در شبکه‌های مبتنی بر نرم‌افزار به توالی از بسته‌ها بین یک مبدأ و یک مقصد، flow گفته می‌شود. در شبکه‌های میتنی بر نرم‌افزار، برای این flow ها می‌توان برنامه نوشت (البته در حالت کلی تر فلسفه اصلی شبکه های مبتنی بر نرم‌افزار این است که می‌توان کل شبکه را برنامه‌ریزی کرد) و یا برایشان سیاست تعریف نمود.

در ابتدا Admin یا End User از سامانه vTAP یک کارکرد شبکه مجازی درخواست میکند. برای Admin این کار میتواند تعریف یک سیاست جدید برای flow بین دو ماشین مجازی مبدأ و مقصد بوده و برای End User نیز این مطلب میتواند به سادگی ping کردن یک ماشین مجازی دیگر یا درخواست یک سرویس مجازی از آن ماشین مجازی دیگر باشد.

در هر دوی این حالات، در نهایت ماشین مجازی (Sender یا Receiver) ترافیکی را ایجاد میکند که از آن با Produce Flows یاد شده است.

از اینجا به بعد، نقش vTAP مطرح می‌شود که متشکل از انجام دو عمل است:

1- Forward کردن flow اصلی به ماشین مجازی مقصد، مثل حالت ارتباط مستقیم بین ماشین مجازی مبدأ و مقصد که vTAP بین شان وجود نداشته باشد. سناریوی Forward the Original Flows به همین امر می‌پردازد.

2- ایجاد و فرستادن یک کپی از flow اصلی به ماشین مجازی Monitor جهت تحلیل ترافیک عبوری. سناریوی Create and Send Copied Flows برای این منظور تدارک دیده شده‌است.

این دو عمل، میتوانند با استفاده از DPDK در صفحه داده (که Framework آماده محسوب میشود) تسریع گردند. استفاده از DPDK میتواند بعنوان گام بعدی این پروژه تلقی شود.

حال که vTAP یک کپی از flow اصلی را به ماشین مجازی Monitor میفرستد، بعنوان قدم نهایی و در آخرین سناریو، Monitor روی flow کپی‌شده عملیاتی انجام میدهد. در ساده‌ترین حالت این کار ثبت اطلاعات درباره ترافیک عبوری خواهد بود. در حالات پیچیده‌تر استفاده از سناریوهای ایجاد حمله ساختگی و بررسی تشخیص آن توسط سیستم مانیتورینگ مستقر در ماشین مجازی Monitor با استفاده از Framework های آماده جهت انجام این کار از قبیل pytbull و Suricata میتواند بعنوان گام بعدی این پروژه تلقی شود. از این سناریو، چه در ساده‌ترین حالت و چه در حالت‌های پیچیده‌تر و بعنوان گام بعدی پروژه تحت عنوان Analyze the Incoming Flows یاد شده و استفاده از تمامی Framework های آماده در سناریوهای ممکن فوق الذکر، در سناریو Use Frameworks Available قرار میگرد.

مراجع و منابع

[1] Kreutz, D., Ramos, F. M. V., Esteves Veríssimo, P., Esteve Rothenberg, C., Azodolmolky, S., & Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. Proceedings of the IEEE, 103(1), 14-76.

[2] “DPDK Documentation,” [online]. Available: core.dpdk.org/doc/

[3] Jeong, S., You, J. H., & Hong, J. W. (2019). Design and Implementation of Virtual TAP for SDN-based OpenStack Networking, 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, USA: IEEE

[4] Sommerville, I. (2015). Software engineering (10th ed.). Boston: Pearson.

[5] Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach. New York, USA: McGraw-Hill.