Archive of SID.ir

مقاله پژوهشی

ارائه یک روش نوین جهت تصدیق صحت ارسال بسته‌ها در شبکه‌های SDN به‌صورت موازی

روزبه بگلری و حاکم بیت‌الهی

چکیده: شبکه‌های کامپیوتری با شکستن فواصل مکانی و زمانی توانسته‌اند کاربران را از سراسر جهان به یکدیگر متصل کنند. از این رو نگهداری و امنیت داده‌ها و اطلاعات، همیشه یکی از چالش‌های اصلی شبکه‌های کامپیوتری بوده است. با پیشرفت تکنولوژی و روش‌های ارتباطات، مکانیسم‌های امنیتی نیز باید مجدداً ارزیابی گردند. با توجه به پیشرفت‌ها، تفاوت‌ها و فرصت‌های جدید در شبکه‌های SDN در مقایسه با شبکه‌های IP، روش‌های موجود برای تأمین امنیت ارسال داده‌ها در شبکه‌های مبتنی بر IP، در شبکه‌های SDN قابل پیاده‌سازی نیستند؛ به همین دلیل با درنظرگرفتن محدودیت‌های SDN برای مقابله با تهدیدهای فرایند ارسال بسته‌ها، روش‌های نوینی ارائه شده‌اند که از مهم‌ترین آنها می‌توان به DYNAPFV اشاره کرد. در این مقاله پس از بررسی روش‌های تصدیق صحت ارسال داده‌ها در شبکه‌های SDN، روشی جدید مبتنی بر DYNAPFV برای تصدیق صحت ارسال بسته‌ها پیشنهاد شده و کلیه مشکلات و نواقص روش‌های موجود، بالاخص DYNAPFV مرتفع گردیده است. نتایج آزمایش‌ها نشان می‌دهند که زمان لازم برای یافتن گره مخرب در الگوریتم پیشنهادی نسبت به الگوریتم DYNAPFV به میزان ۹۲٪ بهبود یافته و نیز با افزایش احتمال تصدیق یکپارچگی بسته از مقدار ۰/۸ به ۰/۹۹، امنیت سیستم بیشتر می‌شود؛ اما در مقابل زمان لازم برای تشخیص سوئیچ‌های مخرب بالاتر می‌رود.

کلیدواژه: **شبکه‌های SDN، امنیت داده، تصدیق صحت ارسال بسته‌ها، Openflow.**

۱- مقدمه

دنیای اطراف ما با شبکه‌های کامپیوتری آمیخته است؛ به‌طوری که در دسترس‌بودن و انتقال دانش به شکل مستقل در گرو استفاده و عملکرد صحیح این شبکه‌هاست؛ لذا باید در امنیت و صحت کارکرد آنها توجه لازم نمود. امروزه پرمصرف‌ترین نوع رسانه، رسانه‌های دیجیتال هستند؛ اما تکنولوژی و محصولی که کاربران زیادی را تحت‌الشعاع قرار دهد، خطرها و تهدیدهای پیشرفته‌ای را نیز به همراه خواهد داشت. از این رو وظیفه متخصصان این حوزه است که علاوه بر ایجاد امکانات لازم برای سهولت استفاده کاربران از این تکنولوژی، امنیت لازم نیز تأمین شود [۱].

در شبکه‌های سنتی که غالباً مبتنی بر IP هستند، با توجه به تاریخچه نسبتاً طولانی خود، افراد زیادی بر روی مسائل حوزه امنیت اطلاعات، تلاش‌های گسترده‌ای انجام داده‌اند. اما با پیشروی تکنولوژی و ظهور انواع بهبودیافته و جدید شبکه‌ها، نیازمندی‌های امنیتی آنها نیز دستخوش تغییر می‌شود [۲]. از این رو شبکه‌های SDN[1] که نوع نسبتاً جدیدی از شبکه‌های کامپیوتری هستند با اصلاحات بنیادی که موجب تغییر نحوه مواجهه با فرایند ارسال داده‌ها از منبع به مقصد شده‌اند، نیازمند روش‌های متفاوت در برخورد با مسائل امنیتی نسبت به شبکه‌های سنتی خواهند بود. بدین صورت که همه دستگاه‌های منتقل‌کننده و تأثیرگذار بر داده‌های شبکه باید تحت اختیار، نظارت و بازبینی موجودیتی به نام کنترل‌کننده باشند. کنترل‌کننده در واقع مجری و تصمیم‌گیرنده سیستم بوده و دستگاه‌های منتقل‌کننده که از لحاظ عملیاتی و پردازشی، دستگاه‌هایی ساده هستند، بر اساس دستوراتی که قبلاً از کنترل‌کننده دریافت کرده‌اند اقدام به تصمیم‌گیری در مورد فعالیت‌های خود می‌کنند. حال با توجه به اینکه داده‌ها بین دستگاه‌های منتقل‌کننده جابه‌جا می‌شوند، صحت یکپارچگی داده‌ها در ارسال و جابه‌جایی باید تأمین گردد [۲].

اگرچه در زمینه تصدیق صحت ارسال و رفتار بسته‌ها در شبکه‌های SDN تلاش‌های ارزشمندی مانند DynaPFV[2] [۲] و Sphinx [۳] انجام شده‌اند، اما همچنان سیستم‌هایی که بر پایه این مکانیسم‌ها عمل می‌کنند می‌توانند تحت حملاتی قرار گیرند که باعث آسیب‌پذیری شبکه و نارضایتی کاربران شوند؛ از جلمه حمله توپولوژی جعلی و حمله عدم سرویس‌دهی ناشی از سیلاب بسته‌ها. علاوه بر این، سرعت این روش‌ها برای یافتن گره مخرب نسبتاً طولانی است و نیاز به حافظه زیادی نیز دارند. هم سرعت پایین و هم حافظه بالا بر تجربه کاربری و نحوه استفاده کاربران تأثیر مستقیم خواهد داشت. در این مقاله، یک روش نوین مبتنی DYNAPFV ارائه می‌گردد که به‌طور مؤثر در مقابل حملات ربایش، تزریق، حذف و دستکاری بسته‌ها مقابله می‌کند. نسبت به روش پایه خود یعنی DYNAPFV حافظه کمتری برای مقابله با حملات می‌خواهد و زمان تشخیص گره‌های مخرب را نیز به‌طور چشمگیری کاهش می‌دهد. به‌طور خلاصه نوآوری‌های مقاله به شرح زیر هستند:

- ارائه یک روش جدید و نوین جهت تصدیق صحت ارسال بسته‌ها در شبکه‌های SDN به‌طوری که با انواع حمله در این زمینه مقابله مؤثر می‌شود.
- روش پیشنهادی این امکان را به شبکه می‌دهد تا ناهنجاری‌های ارسال روی هر مسیر را شناسایی کند.
- روش DYNAPFV در مقابل دو نوع حمله، آسیب‌پذیری جدی دارد: ۱) حمله توپولوژی جعلی که دیدگاه توپولوژی کنترل‌کننده را

این مقاله در تاریخ ۴ اردیبهشت ماه ۱۴۰۱ دریافت و در تاریخ ۲۹ اردیبهشت ماه ۱۴۰۲ بازنگری شد.

روزبه بگلری، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران، (email: rouzbeh.beglari@gmail.com).

حاکم بیت‌الهی (نویسنده مسئول)، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران، (email: beitollahi@iust.ac.ir).

1. Software Defined Network
2. Dynamic Packet Forwarding Verification

Archive of SID.ir

Archive of SID.ir

دستکاری می‌کند و آن را فریب می‌دهد و ۲) هاست‌ها و سوئیچ‌های مخرب می‌توانند با ایجاد سیل ترافیک در شبکه به میزبان‌های دلخواه، باعث ایجاد حملات DoS شوند تا منابع سوئیچ‌های آسیب‌پذیر و یا کنترلر SDN را به‌دست گیرند. روش پیشنهادی در مقابل هر دو نوع حمله ذکرشده مقاوم است و برتری قابل توجهی نسبت به DYNAPFV دارد.

- روش‌های DYNAPFV و NPFV[1] [۲] جهت یافتن سوئیچ‌های مخرب، مستلزم تحمیل زمان و سربار محاسباتی بزرگی بر شبکه می‌گردد. اما روش پیشنهادی به‌صورت چشمگیری این زمان را کاهش میدهد؛ به‌طوری که زمان یافتن سوئیچ‌های مخرب به میزان ۹۲٪ بهبود می‌یابند.
- روش‌های DYNAPFV و NPFV در مقابل حملات کشف ربایش بسته‌ها ناتوان هستند؛ اما در روش پیشنهادی این حمله خنثی شده است.

بخش دوم مقاله به بیان مسئله می‌پردازد و بخش سوم، کارهای پیشین را بیان و ارزیابی می‌کند. بخش چهارم، روش پیشنهادی را شرح می‌دهد و در بخش پنجم به شبیه‌سازی، ارائه نتایج و مقایسه روش پیشنهادی با روش‌های گذشته می‌پردازیم. بخش ششم که بخش پایانی مقاله است به نتیجه‌گیری اختصاص دارد.

۲- ادبیات موضوع و بیان مسئله

در این بخش ابتدا به ادبیات موضوع در این مقاله می‌پردازیم، سپس مسئله و چالش مد نظر را مطرح می‌کنیم و در انتها اهداف و فرضیات پژوهش را بیان خواهیم کرد.

۲-۱ اساس شبکه‌های SDN

SDN یک الگوی شبکه نوظهور است که امیدبخش ایجاد تغییر در محدودیت‌های زیرساختی شبکه‌های فعلی می‌باشد. لایه‌ها در شبکه‌های SDN به‌صورت عمودی یکپارچه شده‌اند و سه نوع سطح یا لایه تعریف شده که موجودیت‌های مختلف شبکه، بازیگران آن هستند: ۱) لایه کاربری که تبیین سیاست‌های سیستم را بر عهده دارد، ۲) لایه کنترل که تصمیم‌گیرنده سیستم بوده و نحوه مواجهه با ترافیک شبکه و مسیر پیش رو را تعیین می‌کند و ۳) لایه داده که ترافیک را طبق سیاست‌های دریافتی از لایه کنترل به رابط بعدی شبکه هدایت می‌کند.

همان طور که مطرح گردید، این الگو در اولین گام با جداکردن منطق کنترل شبکه (سیاست کنترل) از مسیریاب‌ها[2] و سوئیچ‌هایی که ترافیک را هدایت می‌کنند، یکپارچه‌سازی عمودی شبکه‌های IP را می‌شکند. سپس با جداسازی لایه‌های کنترل و دیتا، از یک سو سیاست کنترل را در یک کنترل‌کننده متمرکز منطقی (همان سیستم عامل شبکه) پیاده‌سازی می‌کند و از سوی دیگر سوئیچ‌ها و مسیریاب‌های شبکه را صرفاً به دستگاه‌هایی ساده برای انتقال داده تبدیل می‌کند. این تحول، تغییر سیاست‌ها، پیکربندی (پیکربندی مجدد) و نیز تکامل شبکه را ساده‌تر می‌کند [۴]. تفکیک لایه کنترل و لایه داده را می‌توان با استفاده از یک رابط برنامه‌نویسی مناسب بین سوئیچ‌ها و کنترل‌کننده SDN تحقق بخشید. کنترلر از طریق این رابط برنامه‌نویسی (API) مدیریت مستقیمی را بر عناصر لایه داده اعمال می‌کند [۲]. ظهور شبکه‌های SDN و به‌وجودآمدن هسته‌ای مرکزی برای مدیریت سوئیچ‌ها و قوانین ارسال داده در تمامی بخش‌های شبکه، فرصت مناسبی ایجاد نموده تا در روش‌های Packet Forwarding Verification سنتی، بازنگری عمیقی صورت گیرد. بر اساس تعاریف OpenFlow [۵]، شبکه‌های SDN جهت اجرای دستورات لایه کنترل، یک کنترل‌کننده مرکزی را تحت اختیار دارند. بر روی این کنترل‌کننده مرکزی، یک سیستم عامل شبکه (NOS)[3] تعبیه شده که وظیفه نگهداری از توپولوژی‌های شبکه و پایش بسته‌ها از سوئیچ‌ها را دارد. جهت تبیین قوانین جریان برای سوئیچ‌ها، نرم‌افزارهای متعددی را می‌توان بر بستر NOS توسعه داد تا نحوه برخورد سوئیچ‌ها با بسته‌ها معین گردد. کنترل‌کننده مرکزی، این قوانین جریان را در بین سوئیچ‌های SDN شبکه منتشر می‌کند و نهایتاً سوئیچ‌ها از این قوانین برای پردازش بسته‌ها استفاده خواهند نمود. زمانی که بسته‌ای توسط یک رابط شبکه مانند سوئیچ دریافت می‌شود، آن رابط به جستجو در قوانین جریان خود می‌پردازد تا قانونی درخور مشخصات بسته دریافت‌شده بیابد و در نتیجه نحوه مواجهه با آن بسته را تعیین کند. در صورتی که سوئیچ پس از جستجو، قادر به یافتن قانونی مرتبط با بسته نباشد برای تعیین تکلیف، بسته‌ای جدید را با نام packet_in که حامل آن بسته مورد بحث است، ابتدا تولید و سپس به کنترل‌کننده ارسال می‌کند. در کنترل‌کننده نیز آن بسته به نرم‌افزار مرتبط تحویل داده شده تا قانونی جدید برای آن تبیین گردد. سپس کنترل‌کننده، بسته جدید دیگری را با نام packet_out که حامل قوانین جدید می‌باشد، تولید کرده و به تمام سوئیچ‌هایی که در مسیر تعیین‌شده جهت ارسال قرار دارند توزیع می‌کند. بدین ترتیب، شبکه‌های SDN به‌وسیله توسعه نرم‌افزارهایی که بر روی NOS اجرا می‌شوند، توانایی انعطاف‌پذیری عملیاتی خواهند داشت [۶]. سوئیچ‌های شبکه در شبکه‌های SDN، توانایی پایش آمار انتقال بسته‌های خود (یعنی تعداد بسته‌هایی که هدایت شده‌اند) را دارند. از این طریق، کنترل‌کننده مرکزی با بازیابی این اطلاعات آماری از مجموعه سوئیچ‌ها می‌تواند یک دید کلی از آمار انتقال بسته‌ها در کل شبکه به‌دست آورد. در این حال، انتقال بسته‌ها بین اجزای شبکه SDN مانند هر شبکه دیگری، محیطی مستعد برای انجام حملات و آسیب‌رسانی به شبکه است. یافتن روشی مناسب برای پیشگیری، تشخیص و مواجهه با حملاتی که می‌توانند بسته‌ها را از مسیر مبدأ به مقصد حذف، تکرار، اصلاح و یا در فرایند ارسال به نحوی تغییر ایجاد کنند، حائز اهمیت بسیاری می‌باشد. در شبکه‌های IP برای رسیدن به چنین هدفی از روش‌ها و مکانیسم‌های مبتنی بر رمزنگاری به‌صورت گسترده‌ای استفاده شده است.

۲-۲ حملات مؤثر در انتقال بسته‌ها

از مهم‌ترین بخش‌های آسیب‌پذیر در شبکه‌های سنتی IP و همچنین شبکه‌های SDN، فرایند انتقال بسته‌هاست. در حملات تأثیرگذار بر این فرایند، اساساً می‌توان از طریق ایجاد تغییر در تنظیمات قوانین ارسال در مسیریاب‌ها یا سوئیچ‌ها، در مسیر مبدأ به مقصد بسته‌ها را حذف کرد، در ارسال آنها تأخیر ایجاد کرد، بسته‌های اضافی درج کرد، بسته‌های اصلی را تغییر داد یا بسته‌ها را به تکرار ارسال کرد. بنابراین دستیابی به مکانیسمی منطقی جهت تأیید صحت انتقال بسته در زمان آنی، موضوع مهمی برای اطمینان از میزان امنیت شبکه‌ها در مواجهه با حملات مخرب است [۲].

در این مقاله برای مواجهه با چالش مورد بحث، دو ویژگی امنیتی تأیید یکپارچگی بسته و تأیید مسیر انتقال بسته در نظر گرفته می‌شود و در

1. Novel Packet Forwarding Verification
2. Routers
3. Netowrk Operating System

Archive of SID.ir

Archive of SID.ir

نتیجه تشخیص و مقابله با ناهنجاری‌های مرتبط با این دو ویژگی نیز مورد بررسی قرار خواهند گرفت. از جمله این ناهنجاری‌ها می‌توان به استفاده بیش از حد از منابع سیستم، عملکرد ضعیف در نتیجه سرعت پایین و اعمال بیش از حد تغییرات در پروتکل‌های SDN و بالاخص لایه داده (همان سوئیچ‌های SDN) اشاره کرد. یک مهاجم می‌تواند با تحت اختیارگرفتن لینک یا مسیری مابین رابط‌های شبکه با توجه به شرایط و هدف خود از منابع و امکانات شبکه، سوء استفاده کند. شبکه مورد بحث در این مقاله، متشکل از دسته‌ای از گره‌هاست که در آن، تعداد گره‌های سالم از گره‌های آلوده بیشتر است؛ در نتیجه این اطمینان حاصل می‌شود که یک کنترل‌کننده می‌تواند پیام‌های جمع‌آوری‌شده از سوئیچ‌های مربوط را به‌درستی تأیید کند. همچنین بر اساس مشخصات OpenFlow، فرض می‌کنیم که ارتباطات بین سوئیچ‌ها و کنترل‌کننده ایمن بوده و Security Layer Transport در SDN فعال شده است. تشخیص و مواجهه با حملات زیر به‌صورت اختصاصی در این مقاله مورد بررسی قرار خواهند گرفت:

- حملات توپولوژی[1]: به حملاتی گویند که مهاجم با ایجاد تغییراتی در کنترلر یا سوئیچ‌ها باعث گردد بسته به مسیر صحیح هدایت نشود.
- حملات حذف بسته[2]: به حملاتی گویند که مهاجم در مسیر مبدأ تا مقصد، تعدادی از بسته‌ها را حذف نماید.
- حملات افزودن بسته[3]: به حملاتی گویند که مهاجم در مسیر مبدأ تا مقصد، تعدادی بسته به مجموعه ارسالی اضافه کند.
- حملات ربایش بسته[4]: به حملاتی گویند یک مهاجم سوئیچ‌هایی را روی مسیر مبدأ تا مقصد یک جریان کنترل می‌کند تا بسته‌ها را به مسیری هدایت کند که برای جریان مجاز نیست. بسته‌ها ممکن است به مسیرهای مجاز اصلی بازگردانده یا حذف شوند. بسته‌های ربایش‌شده می‌توانند به‌وسیله مهاجمین مورد سوء استفاده قرار گیرند.
- حملات دستکاری بسته[5]: به حملاتی گویند که یک مهاجم در مسیر مبدأ تا مقصد یک جریان، مستقیماً یک یا چند سوئیچ را کنترل کند تا محتویات بسته‌ها (اعم از هدر بسته یا محموله) یک جریان در مسیر را تغییر دهد. این کار می‌تواند باعث شود مفهوم پیام به کلی از بین رود.
- حمله ترکیبی[6]: مهاجم در این حمله، حملات ربایش و درج بسته را با هم ترکیب و حمله جدیدی را ساماندهی می‌کند. برای مثال یک سوئیچ مخرب می‌تواند تعداد قابل توجهی بسته را دور بریزد و سپس به همان تعداد بسته جدید را که یا ربوده و یا تولید کرده، جایگزین نماید.

در این مقاله، یک شبکه مقیاس بزرگ SDN را به‌عنوان محیط آزمایشی در نظر می‌گیریم که در آن، ترافیک بین یک منبع و مقصد از طریق گره‌های میانی عبور می‌کند. مکانیسم پیشنهادی به شبکه این امکان را می‌دهد تا وجود سه ناهنجاری مورد بحث در هر مسیر تشخیص داده شود. همچنین علاوه بر حملات مطرح‌شده تا حد توان، تلاش برای مقابله با سایر حملات تأثیرگذار در فرایند تصدیق صحت ارسال بسته‌ها در شبکه‌های SDN شده است.

1. Topology Attacks
2. Packet Dropping Attacks
3. Packet Injection Attacks
4. Packet Hijacking Attacks
5. Packet Tampering Attacks
6. Hybrid Attacks

۲-۳ اهداف پژوهش

ابتدا به دو روش مطرح صحت ارسال بسته‌ها در شبکه‌های SDN به نام‌های N-PFV و DYNAPFV می‌پردازیم و نقاط ضعف و قوت آنها را بررسی می‌کنیم.

پس از بررسی و مقایسه تحقیقات و روش‌های موجود در بخش پیشینه تحقیق، محصول این مقاله، ارائه الگوریتمی سازگار و بهینه در شبکه‌های SDN خواهد بود که چالش‌های موجود در متدهای پیشین تصدیق صحت ارسال بسته‌ها را در نظر گرفته و عملکردی بهینه از خود ارائه دهد. در الگوریتم پیشنهادی، کمترین تغییر در ساختار بسته‌ها صورت خواهد گرفت تا پیاده‌سازی آن در عمل امکان‌پذیر باشد و با توجه به شرایط شبکه‌های SDN و توان پردازشی محدود سوئیچ‌های موجود در این شبکه‌ها، سربار محاسباتی هر بسته به مقدار متناسبی در نظر گرفته خواهد شد. نهایتاً با شبیه‌سازی محدود این الگوریتم در پروتکل ارتباطی OpenFlow که یک محیط عملیاتی مختص شبکه‌های SDN است، به مقایسه عملکرد و بهینگی این الگوریتم با سایر الگوریتم‌های مرتبط خواهیم پرداخت. در این مقاله، تحقق اهداف زیر در بررسی و ارائه الگوریتم‌های تأیید صحت ارسال بسته‌ها مد نظر خواهد بود.

- **تأیید صحت یکپارچگی بسته:** الگوریتم با تأیید یکپارچگی بسته‌ها از انجام هر گونه تغییر در ساختار بسته‌ها جلوگیری نموده و در نتیجه با حملات تأخیر، تغییر، حذف و اضافه‌کردن بسته‌ها مقابله می‌کند.
- **تأیید صحت انتقال بسته و صحت مسیر:** الگوریتم پس از آنالیز و بررسی مداوم شبکه در هنگام وقوع حمله، ابتدا به تشخیص نوع آن می‌پردازد.
- **عملکرد الگوریتم:** با افزودن مقدار بهینه سربار محاسباتی به شبکه از ایجاد تأخیرهای ارسال و پیچیدگی‌های زمان‌بندی و همگام‌نبودن سوئیچ‌ها جلوگیری می‌کند.

۲-۴ فرضیات پژوهش

- کنترل‌کننده مرکزی SDN، وظیفه هدایت و بازرسی موجودیت‌های شبکه را به عهده دارد.
- دستگاه‌های منتقل‌کننده داده، توان پردازشی و محاسباتی ضعیفی دارند.
- اتصال روترها به کنترل‌کننده با پروتکل TLS ایمن است.
- Openflow به‌عنوان پراستفاده‌ترین و کامل‌ترین پروتکل Southbound است.

۳- پیشینه تحقیق

در شبکه‌های IP، مفاهیم و روش‌های تصدیق صحت ارسال بسته به‌صورت جامعی مورد تحقیق و توسعه قرار گرفته‌اند [۶] تا [۸]. در بعضی از این روش‌ها از تگ‌های رمزنگاری استفاده شده و این تگ‌ها در بسته‌ها جایگذاری می‌شوند تا سوئیچ‌های مابین مبدأ تا مقصد با تطبیق آنها، یکپارچگی و سلامت بسته‌ها را تأیید کنند. اما استفاده از این متدها علاوه بر اینکه موجب ایجاد تغییرات ساختاری در بسته‌ها می‌شوند، میزان قابل توجهی سربار محاسباتی نیز به بسته‌ها اضافه می‌کنند.

سایر روش‌ها مانند Packet Probing و Acknowledgment-based نیز روش‌های مناسبی برای یافتن ناهنجاری‌های ارسال بسته‌ها هستند [۹] و [۱۰]. مثلاً Traceroute با ارسال بسته‌های Probe در طول مسیرهای مشکوک و انتقال این بسته‌ها در هر جهش به همراه بسته‌های packet-in به بخش سیاست کنترل، به تشخیص ناهنجاری‌ها می‌پردازد

Archive of SID.ir

Archive of SID.ir

[۱۱]. در این روش‌ها میزان سربار ارتباطی به میزان قابل توجهی افزایش یافته و همچنین بیشتر این گونه روش‌ها فقط به تشخیص ناهنجاری در اولین و آخرین جهش می‌پردازند و توانایی تشخیص حملات Hijacking را ندارند. در روش‌های Acknowledgment-based سوئیچ‌های همسایه به شکل دوره‌ای با یکدیگر به تعامل پرداخته و هر سوئیچ باید تمامی مسیرها و Acknowledgmentهای دریافتی از سوئیچ‌های دیگر را در خود ذخیره کند [۶] و [۱۰].

روش‌های شمارنده میزان جریان [۱۰] و [۱۲] نیز مؤثر در بررسی صحت ارسال بسته‌ها در شبکه‌ها هستند؛ ولی این روش‌ها مستلزم همگام‌بودن دقیق سوئیچ‌های شبکه می‌باشند که دستیابی به این امر در عمل بسیار پیچیده خواهد بود. روش‌های FloodGuard [۱۳] و AVANT-GUARD [۹] به جهت ایمن‌سازی شبکه‌های SDN معرفی شده‌اند و مکانیسم‌هایی برای مقابله با حملات سیاست کنترل هستند.

دو الگوریتم DYNAPFV [۲] و N-PFV [۲] که از لحاظ روش‌های عملیاتی به یکدیگر نزدیک هستند، هر دو از محاسبات آماری برای تصدیق صحت ارسال بسته استفاده کرده و در نتیجه سربار محاسباتی کمی به شبکه می‌افزایند. سیستم ارزیابی در الگوریتم DYNAPFV پویا و هوشمند است که می‌تواند خود را با شرایط موجود تطبیق دهد؛ ولی در الگوریتم N-PFV روش و الگوریتم ایستا میباشد و قابلیت انعطاف‌پذیری وجود ندارد. الگوریتم DYNAPFV بر پایه N-PFV ایمنی مناسبی را برای تصدیق یکپارچگی و رفتار بسته‌ها در شبکه ارائه می‌کند. اما با توجه به اینکه حملات به توپولوژی شبکه تأثیر مستقیم و بسزایی در تصدیق صحت ارسال بسته‌ها در شبکه دارند و این مکانیسم استراتژی مناسبی برای مقابله با حملات توپولوژی در نظر نگرفته است، از این جهت بی‌دفاع خواهد بود. همچنین در DYNAPFV، داده‌های PACKET_IN ارسالی، شامل سرآیند و بار مفید، همگی به‌صورت خام به کنترل‌کننده ارسال می‌شوند و پس از رسیدن به کنترل‌کننده در توابع MAC قرار می‌گیرند. با توجه به اینکه روش MAC مورد بحث در DYNAPFV به‌صورت قطعی است و حالت تصادفی نیز ندارد، استفاده از MAC در عمل تأثیری در بهبود وضعیت یکپارچگی داده ندارد و صرفاً بار محاسباتی در مجموعه عملکرد شبکه ایجاد می‌کند. به علاوه کلید رمزنگاری MAC صرفاً در کنترل‌کننده موجود است و رمزکردن داده‌های درهم‌ریخته‌شده، تأثیرگذاری در افزایش اطمینان از امنیت شبکه نخواهد داشت؛ زیرا اگر حتی خود کنترل‌کننده تحت حمله قرار گیرد و داده‌ها به سرقت روند، دسترسی به داده‌های درهم‌ریخته و یا رمزشده عملاً تغییری ایجاد نمی‌کند.

در [۳] Sphinx برای شناسایی حملات شناخته‌شده و همچنین ناشناخته به توپولوژی شبکه و فرایند انتقال داده‌ها در سیاست داده SDN پیشنهاد گردیده است. Sphinx به‌طور پویا رفتار جدید شبکه را یاد می‌گیرد و هنگامی که تغییرات مشکوکی را در رفتار سیاست کنترل شبکه مشاهده کند، هشدار می‌دهد. این الگوریتم قادر است تا حملات در SDN را در زمان واقعی و با هزینه بسیار کم تشخیص دهد و برای استقرار، هیچ تغییری در کنترل‌کننده‌ها ایجاد نمی‌کند.

در [۱۴]، یک افزونه امنیتی به نام SDNsec ارائه شده که قدرت مسئولیت‌پذیری را برای ارسال بسته‌ها برای سیاست داده SDN فراهم می‌کند. قوانین ارسال در بسته‌ها رمزگذاری می‌شوند تا از رفتار ثابت شبکه در هنگام پیکربندی مجدد اطمینان حاصل شود و همچنین حملات State-Exhaustion به دلیل جستجوی متعدد در جدول مسیریابی محدود گردد. در [۱۵] طرح Passport ارائه شده است؛ سیستمی که اجازه تأیید آدرس‌های منبع را در شبکه می‌دهد. Passport از رمزنگاری کلید متقارن برای قراردادن نشانه‌ها روی بسته‌ها استفاده می‌کند که به هر سیستم مستقل (AS) در طول مسیر شبکه اجازه می‌دهد تا به‌تنهایی اعتبار منبع آدرس را تأیید کند. این روش بدون نیاز به به‌روزرسانی میزبان با استفاده از سیستم مسیریابی، کلیدهای متقارن را جهت تأیید صحت بسته‌ها توزیع می‌کند.

٤- تصدیق صحت ارسال بسته‌ها به روش پیشنهادی

در این بخش ابتدا ما به اهداف مورد نظر در طراحی یک روش موفق تصدیق صحت ارسال بسته‌ها اشاره کرده و سپس به توصیف روش پیشنهادی می‌پردازیم.

٤-١ اهداف طراحی

ما یک شبکه با مقیاس بزرگ را در مکانیسم تصدیق موازی در نظر می‌گیریم؛ مانند یک شبکه ISP که با SDN پیاده‌سازی شده است، جایی که ترافیک بسته‌ها در میان گره‌های منبع و مقصد بسیاری از طریق گره‌های میانی منتقل می‌شوند [۱۶] و [۱۷]. فرضیه ما بر این است که اکثر گره‌های سالم در شبکه موجود می‌باشد. در ادامه ما اهداف مورد نظر در طراحی یک روش موفق را بیان می‌کنیم:

- **تأیید مسیر ارسال بسته:** تأیید مسیر بسته این اطمینان را می‌دهد که بسته در مسیر درستی ارسال و هدایت می‌شود.
- **تأیید یکپارچگی بسته:** تأیید یکپارچگی این اطمینان را می‌دهد که بسته‌ها به سلامت از مبدأ به مقصد منتقل شده‌اند و محتویات بسته تغییر نکرده است.
- **تأیید رفتار بسته در حرکت:** تأیید رفتار بسته این اطمینان را می‌دهد که تعداد بسته‌ها و زمان‌بندی ارسال آنها، همان گونه که انتظار می‌رود بوده است.
- **مکانیسم ایمن‌سازی شبکه:** مکانیسم ایمن‌سازی این اطمینان را می‌دهد که حتی در صورت ربایش بسته‌ها، محتویات آنها مورد سوء استفاده قرار نمی‌گیرد.
- **مصرف زمان بهینه:** مکانیسم تبیین‌شده باید در زمان بهینه، وظایف خود را انجام داده و به نتیجه مورد نظر دست یابد.
- **نیاز به منابع محاسبات و ذخیره‌سازی کم:** نیاز به منابع مورد استفاده در سیستم، بالاخص سوئیچ‌ها که توان محدودی دارند باید مدیریت شده و کم باشد.
- **عدم اعمال تغییرات زیاد در شبکه:** تغییرات زیاد در لایه کنترل و هدایت داده‌ها باعث می‌شود پیاده‌سازی دشوار گردد؛ در نتیجه باید حداقل تغییرات اعمال گردد.

٤-٢ تصدیق صحت ارسال بسته‌ها به‌صورت موازی (PaPFV)

جهت طراحی یک مکانیسم جامع با استفاده از فرضیات DYNAPFV و با الهام از Sphinx، اقدام به توسعه روشی به نام مکانیسم تصدیق موازی یکپارچگی داده‌ها کرده‌ایم تا ضعف‌های تحلیل‌شده مرتفع گردند و در نتیجه حاصل کار، الگوریتمی با بهبود عملکرد، سرعت و اطمینان خواهد بود.

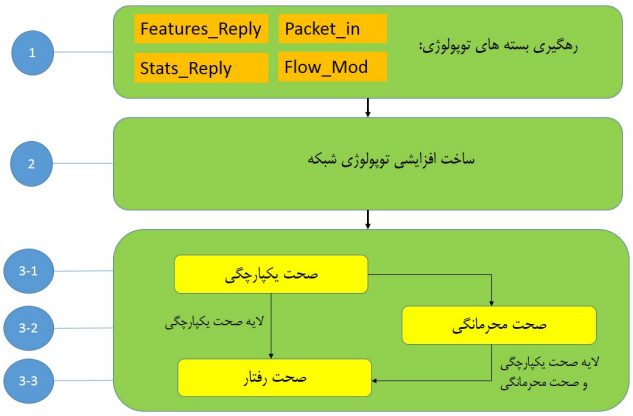
٤-٣ معماری PaPFV

چرخه عملیات این مکانیسم دارای بخش‌های زیر است:

١) رهگیری بسته‌های مؤثر در توپولوژی شبکه

Archive of SID.ir

Archive of SID.ir



شکل ۱: فرایندهای PaPFV.

۲) ایجاد و به‌روزرسانی گراف جریان در شبکه به‌صورت تدریجی
۳) مکانیسم ایمن‌سازی شبکه
۴) مکانیسم اعتبارسنجی شبکه

در روش پیشنهادی یا همان روش تصدیق موازی، کنترلر همواره تمام بسته‌های مرتبط با انتقال و توپولوژی شبکه را که از سوئیچ‌های شبکه زیرمجموعه به سمت کنترلر و برعکس جابه‌جا می‌شوند، پایش و بررسی می‌کند تا تصویری صحیح از جای‌گیری گره‌ها و نحوه انتقال بسته‌ها داشته باشد. کنترلر با پایش این بسته‌ها، یک گراف از وضعیت گره‌های موجود در شبکه را برای خود ترسیم می‌کند تا تغییرات غیرمنتظره مشخص شوند و در نتیجه، کنترلر توانایی تشخیص اقدام خرابکارانه را خواهد یافت. در این روش، اطلاعات بازگشتی از سوئیچ‌ها به جهت تغییر توپولوژی اعتبارسنجی می‌شوند و همچنین تغییرات کلی به‌صورت ناگهانی رخ نخواهد داد و مسیرها در مواقع نیاز به‌صورت تدریجی تغییر می‌یابند. سپس با الهام از مکانیسم DynaPFV در ابتدا با احتمال بدبینانه نسبت به امنیت شبکه، اقدام به تصدیق صحت و ایمن‌سازی پیام‌ها در شبکه می‌کنیم. در ادامه و به‌صورت تدریجی با بازرسی‌های امنیتی، احتمال وجود عنصر مخرب در شبکه را تغییر خواهیم داد و متناسب با آن عمل می‌کنیم.

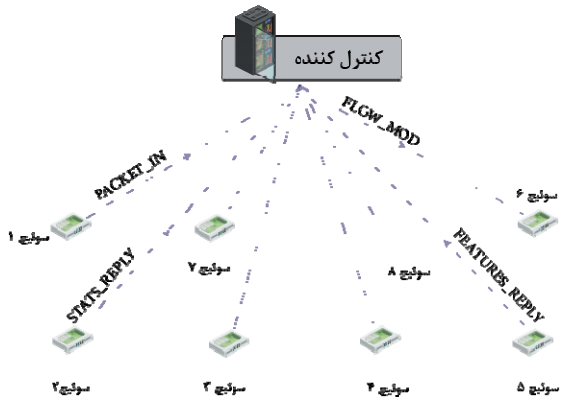
شکل ۱ معماری سطح بالای روش پیشنهادی (PaPFV) را نشان می‌دهد که از سه بخش اصلی رهگیری بسته‌های توپولوژی، ایجاد افزایشی توپولوژی شبکه و سرویس‌های امنیتی (صحت یکپارچگی، صحت محرمانگی و صحت رفتار) تشکیل شده است.

٤-۳-۱ رهگیری بسته‌های مؤثر در توپولوژی شبکه

همان طور که مطرح شد از اصول اولیه تصدیق صحت ارسال بسته‌ها در شبکه، سلامت توپولوژی و دید کنترلر از مکان سوئیچ‌ها و میزبان‌ها است. در مکانیسم تصدیق موازی، تمامی بسته‌هایی که در توپولوژی شبکه مؤثر هستند توسط PaPFV بازرسی می‌شوند. در این راستا مطابق با شکل ۲، کنترلر اقدام به بازرسی اطلاعات مورد نیاز خود (فراداده‌ها) از packet-inها، packet-outها و جریان‌های داده بین گره‌های شبکه خواهد نمود.

رهگیری بسته‌های مؤثر در توپولوژی شبکه در مکانیسم تصدیق موازی به چهار نوع بسته وابسته است:

- **PACKET_IN:** در مکانیسم PaPFV، بسته PACKET_IN که توسط یک سوئیچ ارسال می‌گردد، به‌عنوان نمایانگر شروع جریان و حامل اطلاعات توپولوژیکی (IP-MAC میزبان، اتصالات MAC-Port یا Switch-Port) استفاده می‌شود.



شکل ۲: نحوه ارتباط کنترلر با سوئیچ‌ها.

- **FLOW_MOD:** صدور این نوع بسته توسط کنترلر نشان‌گر مسیرهای مورد نظری است که باید توسط جریان‌ها طی گردد و هر به‌روزرسانی بعدی نیز در مسیر ارسال توسط FLOW_MOD تبیین می‌شود.
- **STATS_REPLY:** جهت استخراج آمار سطح جریان در صفحه داده‌ای از این بسته استفاده می‌شود. این بسته به‌صورت دوره‌ای از سوئیچ‌ها توسط کنترلر درخواست و دریافت می‌شود. در پاسخ به کنترلر، آمار کل سطح جریان در صفحه داده از جمله بسته‌ها/بایت‌های منتقل‌شده به‌صورت کامل ارسال خواهد شد. مکانیسم PaPFV علاوه بر ساخت توپولوژی از این بسته در اعتبارسنجی شبکه نیز استفاده می‌کند.
- **FEATURES_REPLY:** زمانی که اولین بار یک سوئیچ به کنترلر متصل می‌گردد، برای معرفی خود اطلاعاتی مانند وضعیت پورت‌ها و ... را ارسال می‌کند که توسط PaPFV رهگیری می‌شود.

با توجه به مباحث مطرح در OpenFlow، بیشترین تأثیر در شکل‌گیری توپولوژی در شبکه به عهده این چهار نوع بسته می‌باشد. بسته‌های FEATURES_REPLY، PACKET_IN، STATS_REPLY از سمت سوئیچ‌ها و بسته FLOW_MOD از سمت کنترلر ارسال می‌گردد.

در مکانیسم تصدیق موازی، گرافی از توپولوژی و نحوه قرارگیری گره‌ها در شبکه با توجه به این چهار نوع بسته و به‌صورت تدریجی ترسیم می‌شود؛ الگوریتم PaPFV در مورد تغییر توپولوژی و مسیر کنترلر تصمیمات ناگهانی نخواهد گرفت. این تصمیمات ناگهانی می‌توانند در صورتی باشند که سوئیچی که تحت کنترل یک مهاجم درآمده، به‌صورت اشتباه یا عامدانه تصویر غلطی با پیام‌های ارسالی خود به کنترلر ارائه کند و بسته‌ها به مسیر نامشخصی هدایت شوند.

٤-۳-۲ ایجاد و به‌روزرسانی گراف جریان در شبکه به‌صورت تدریجی

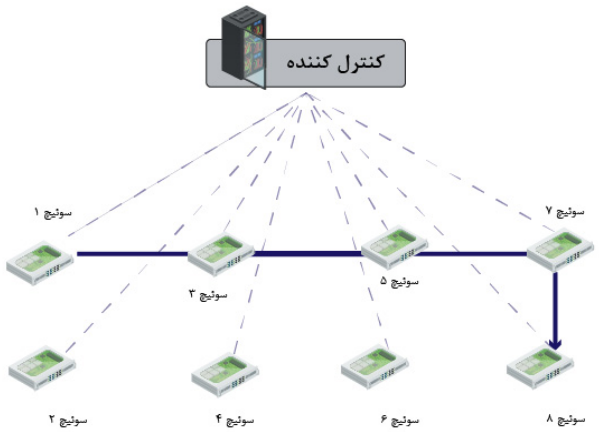
در محیط‌های SDN سه موجودیت مؤثر در تشخیص مسیر و ارسال داده وجود دارد: میزبان، سوئیچ و جریان [۱۸] و [۱۹]. PaPFV فراداده‌های مرتبط با این سه موجودیت را ذخیره کرده و هر تغییر اساسی را در توپولوژی با آنچه از قبل می‌داند بررسی می‌کند. این فراداده‌ها شامل موارد زیر هستند:

۱) Source MAC/IP/port
۲) Switch and in/out-port
۳) Flow match and statistics

اتصالات IP و Mac در هر گره منبع و مقصد در شبکه، نشان‌دهنده مکان هر گره نسبت به دیگر نقطه‌ها، اتصالات Mac/Port مشخص‌کننده یک

Archive of SID.ir

Archive of SID.ir



شکل ۳: تعیین مسیر حرکت جریان توسط کنترلر.

جریان یکتای داده بین دو گره و بررسی تطابق جریان‌ها نمایان‌گر نقاط میانی و مسیرهای طی‌شده توسط جریان است. علاوه بر این، PaPFV این اتصالات توپولوژیکی فیزیکی و منطقی خاص جریان را برای تمامی گره‌ها و نیز وضعیت انتقال بسته‌ها توسط هر سوئیچ میانی در مسیر جریان را که با FLOW_MOD مشخص می‌شود، به‌خاطر می‌سپارد تا در صورت به‌روزرسانی‌های مخرب، توانایی شناسایی آنها را داشته باشد. شکل ۳ تعیین مسیر حرکت جریان توسط کنترلر را نشان می‌دهد.

همزمان با پایش گراف جریان و رهگیری بسته‌های مؤثر در تغییر توپولوژی شبکه، PaPFV به‌عنوان یک نرم‌افزار لایه کاربری اقدام به ایجاد لیست سیاهی از مسیرهایی می‌کند که هنوز تأییدشده نیستند. این لیست سیاه در اختیار ماژول‌های توپولوژی کنترل‌کننده قرار می‌گیرد تا از مسیریابی از طریق لینک‌های اعلام‌شده جلوگیری گردد و PaPFV به‌طور مداوم و تدریجی اقدام به به‌روزرسانی این لیست می‌کند. در این پژوهش که به‌طور اختصاصی از کنترل‌کننده Floodlight استفاده شده است، با توجه به آنکه این کنترل‌کننده متن باز است، تغییراتی اولیه و سطحی در کلاس TopologyManager.java پیاده‌سازی کردیم تا توانایی اعمال لیست تولیدشده را داشته باشیم.

٤-۳-۳ مکانیسم ایمن‌سازی شبکه

وقتی یک جریان جدید با درخواستی جدید، وارد یک سوئیچ ورودی در شبکه شود، سوئیچ ورودی برای تعیین تکلیف، یک بسته PACKET_IN را تولید و به‌عنوان یک درخواست به کنترلر ارسال می‌کند. علاوه بر فعالیت‌های طبیعی شبکه‌های SDN بر اساس میزان اطمینان از شبکه، کنترلر به وسیله یکی از روش‌های زیر به تناسب با بسته مواجه می‌شود:

۱) لایه امنیتی یک: تصدیق صحت پیام

۲) لایه امنیتی دو: تصدیق صحت پیام و محرمانگی داده‌ها

در الگوریتم شماره ۱ (شکل ۴)، کلیات مکانیسم پیشنهادی بیان گردیده و همان طور که قابل مشاهده است، مهم‌ترین اصل در این مکانیسم، احتمال سلامت شبکه (عدم حضور مهاجمین) خواهد بود. در جریان فعالیت‌های مکانیسم، محرک‌هایی موجود است که احتمال اطمینان و سلامت را کاهش یا افزایش می‌دهند. در ابتدا فرض مکانیسم بر عدم سلامت شبکه می‌باشد ($\lambda = 1$) و λ بیانگر میزان سلامت شبکه است. اگر این پارامتر به سمت یک میل کند، بیانگر عدم سلامت شبکه و اگر به سمت صفر میل کند، بیانگر سلامت مطلق شبکه است.

سطح امنیتی درجه یک

در صورت دریافت بسته در کنترلر از یک سوئیچ ورودی، بسته به کلاس Integrity_verification ارسال می‌شود؛ اما قبل از ارسال بر

Parallel_Verification
Input: an incoming packet_in packet F for flow f, a set of hash values of packet_in packets P for flow f received from the ingress switch, the counter of packets received from egress switch m, a set of statistics Si for fi that retrieved from corresponding switches in the path, the sum of counters of packets received from ingress switch u, and a probability of a packet under integrity verification;
Output: verification succeeds, otherwise fails;

1. if packet F of flow fi received then
2. if F is from an ingress switch then
3. sample pakcet_in packets (with λ probability)
4. if (λ>h)
5. Confidentiality_Verification()
6. Distribute F back to Ingress Switch
7. Distribute egress rule
8. end if
9. set the lifetime of rules according to t
10. distribute rules to switches
11. Integrity_Verification (F, λ, m)
12. end if
13. else if F is from an egress switch
14. Integrity_Verification (F, λ, m)

شکل ۴: الگوریتم شماره ۱.

Integrity_Verification (F, λ, table)
Input: an incoming packet_in packet F for flow f, a set of hash values of packet_in packets P for flow f received from the ingress switch, the counter of packets received from egress switch m, a set of statistics Si for fi that retrieved from corresponding switches in the path, the sum of counters of packets received from ingress switch u, and a probability of a packet under integrity verification;
Output: verification succeeds, otherwise fails;

1. if (F is from an ingress switch)
2. feed PACKET_IN to SHA-256
3. store it with a random timer bigger than round trip time
4. else if (F is from an egress Switch and hash result exists in table)
5. remove hash from table
6. β++
7. λ = α*λ
8. else if (F is from an egress Switch and hash result doesn't exist in table
9. or a record in table expires)
10. λ = λ+1/2
11. if (size of(table)+β-unmatched packets/size of(table)+β) <θ)
12. return false
13. end if
14. end if
15. if (flag =1) then
16. Confidentiality_Verification (F, Key);
17. Distribute F back to egress Switch
18. end if
19. else if a record in hash table expires then
20. Integrity_Verification()
21. else if rules installed in switches expire then
22. Beahaviour_Verification()
23. end if

شکل ۵: الگوریتم شماره ۲.

اساس میزان λ، تصمیم‌گیری می‌شود که آیا خود بسته بدون تغییر ارسال گردد یا به‌صورت رمزنگاری‌شده (الگوریتم شماره ۲ بیان‌شده در شکل ۵). در ابتدا زمان حیات مشخصی- برابر با مسیر رفت و برگشت محاسباتی بسته به مقصد از مبدأ- برای جریان تنظیم می‌گردد. در صورتی که نیاز به رمزنگاری بسته نباشد، بسته ابتدا طی یک الگوریتم درهم‌نگاری به هش تبدیل گردیده و در یک جدول به همراه یک زمان تصادفی ذخیره خواهد شد. سپس کنترلر با توجه به مقصد بسته و گرافی که از شبکه برای خود ترسیم کرده است، اقدام به انتشار FLOW_MOD در قالب بسته‌های PACKET-OUT می‌کند. سپس بسته در امتداد مسیر جریان در شبکه با توجه به گره‌هایی که کنترلر در آنها قوانین مورد نیاز را نصب کرده، پیش می‌رود تا به مقصد برسد. در این هنگام گره مقصد یک

Archive of SID.ir

Archive of SID.ir

جدول ۱: مقایسه روش‌های مختلف صحت ارسال بسته‌ها در شبکه‌های SDN.

مدل کنترل دسترسی	مزایا	معایب
روش NPFV	– سادگی مدل – بررسی تمام بسته‌ها – ایمنی بالا – مقابله با حملات (حذف، تأخیر، افزودن و دستکاری بسته‌ها)	– زمان بسیار زیاد اجرا – نیاز به حافظه زیاد – عدم بررسی توپولوژی – عدم کشف حملات رباش
روش DYNAPFV	– بررسی بسته‌ها به‌صورت پویا – ایمنی مناسب – مقابله با حملات (حذف، تأخیر، افزودن و دستکاری بسته‌ها)	– زمان اجرای بالا – نیاز به حافظه زیاد – عدم بررسی توپولوژی – عدم کشف حملات رباش
روش پیشنهادی	– بررسی بسته‌ها به‌صورت پویا – ایمنی بالا – زمان اجرای مناسب – بررسی توپولوژی – مقابله با تمامی حملات ذکرشده برای دو روش بالا به اضافه مقابله با حمله رباش بسته‌ها	احتمال کم کشف گره مخرب در بررسی رفتاری بسته‌ها هنگام استفاده از α با مقادیر پایین‌تر از ۷۰

PACKET_IN حاوی سربار و بار بسته دریافتی را به کنترلر ارسال می‌کند و کنترلر نیز پس از درهم‌ریختن بسته به دنبال همان عبارت درهم‌ریخته در جدول می‌گردد. یک جدول PaPFV حداقل باید دارای عناصر مشخص‌شده باشد.

در صورتی که بسته یافت شد الگوریتم PaPFV به میزان بسته‌های تأییدشده، یک عدد اضافه کرده، هش را از جدول حذف می‌کند و میزان اطمینان به سلامت شبکه را افزایش می‌دهد (λ متمایل به صفر می‌شود). اما در صورتی که بسته یافت نشد، میزان λ دو برابر شده و اطمینان ما از سلامت شبکه کاهش خواهد یافت. پس از آن به دنبال یافتن گره(های) مخرب در شبکه می‌گردیم.

این جستجو به‌راحتی با استفاده از یکی از الگوریتم‌های نمونه‌برداری مانند درخت جستجوی باینری یا روش‌های پیچیده‌تر مثل زنجیره مارکوف مونت کارلو انجام می‌شود. همچنین در انتخاب الگوریتم درهم‌نگاری، الگوریتمی مانند SHA-۲۵۶ به‌دلیل سرعت عملیاتی و کوتاه‌بودن عبارت درهم‌ریخته آن در مقایسه با سایر الگوریتم‌ها توصیه می‌شود. همچنین با توجه به آنکه ممکن است نمونه‌گیری از بسته‌ها در شبکه به‌صورت تصاعدی زیاد گردد، استفاده از SHA-۲۵۶ باعث می‌شود که احتمال به‌وجودآمدن مشکل لانه کبوتری بسیار کم باشد. با این حال استفاده از سایر الگوریتم‌های درهم‌نگاری با توجه به مشخصات و نیازمندی‌های شبکه امکان‌پذیر است.

برخلاف DynaPFV و به‌جای استفاده از MAC برای ذخیره‌سازی PACKET_IN، همان طور که مطرح گردید، فرایند رمزنگاری داده‌ها پس از درهم‌ریختن عملاً تأثیری بر امنیت نداشته و صرفاً باعث کاهش سرعت عملیاتی خواهد شد. در نتیجه صرفاً از یک الگوریتم درهم‌نگاری مانند SHA-۲۵۶ استفاده نموده و حاصل را در جدول ذخیره می‌کنیم.

سطح امنیتی درجه دو

اگر میزان λ از یک میزان مشخص (N) بیشتر باشد، در این صورت ابتدا پرچم رمزنگاری در جدول ۱ برای بسته در حال بررسی ۱ می‌شود. سپس فقط بخش بار بسته بدون سربار، رمزنگاری و همراه با کلید مختص آن در جدول ذخیره می‌گردد. برخلاف سطح امنیتی درجه یک، دیگر خود PACKET_IN درهم‌نگاری نمی‌شود؛ بلکه نسخه رمزشده آن در الگوریتم درهم‌نگاری قرار می‌گیرد. سپس کنترلر، بسته رمزشده را به گره ورودی شبکه برمی‌گرداند؛ به‌طوری که از این پس فقط سربار بسته توسط گره‌ها قابل مشاهده است. در ادامه کنترلر با توجه به مقصد بسته و گرافی که از شبکه برای خود ترسیم کرده، اقدام به انتشار FLOW_MOD در قالب بسته‌های PACKET-OUT می‌کند.

بسته پس از طی مسیر مشخص‌شده برای جریان توسط کنترلر به سوئیچ خروجی (مقصد) می‌رسد. این گره موظف است بر اساس میزان اطمینان از شبکه و دستوری که کنترلر به گره صادر کرده، تمام محتویات بسته را در قالب یک PACKET_IN به کنترلر ارسال نماید. پس از دریافت بسته توسط کنترلر، ابتدا بسته در فرایند مقایسه هش‌ها قرار می‌گیرد و در صورت یافت‌شدن با کلید مختص خود، اقدام به رمزگشایی بسته خواهد شد. سپس بسته رمزگشایی‌شده به گره انتهایی باز می‌گردد. در صورتی که معادل بسته درهم‌ریخته، بسته یافت نشد همانند قبل به سطح امنیتی درجه یک عمل خواهد شد.

سطح امنیتی درجه دو به جهت مقاومت در برابر حملات رباش بسته تعبیه شده است؛ بدین معنی که اگر در مسیر بسته‌ها گرهی در حال رباش و شنود بسته وجود داشته باشد، این سطح امنیتی از وقوع این حملات جلوگیری خواهد کرد. نکته لازم به توجه آن است که میزان N بر حسب نیازمندی مدیران شبکه به ایمن‌بودن قابل تغییر می‌باشد.

۴-۳-۴ اعتبارسنجی رفتار شبکه

آغازگر محرک اعتبارسنجی در شبکه، پایان زمان یک جریان است. در این هنگام ما به بررسی رفتار گره‌ها می‌پردازیم و میزان اطمینان از شبکه را کاهش یا افزایش می‌دهیم. بررسی رفتار گره‌ها، صداقت گره‌ها در انتقال بسته و عدم وجود گره مخرب جهت انجام حمله حذف بسته، تزریق بسته و رباش بسته را مشخص خواهد کرد. در الگوریتم‌های شماره ۳ و ۴ (شکل‌های ۶ و ۷)، مکانیسم عملیاتی با الهام از DynaPFV شرح داده شده است.

برخلاف DynaPFV که از تمام گره‌های در مسیر جریان، درخواست ارسال آمار یا flow_stats می‌کند، در ابتدا PaPFV فقط از سه گره مبدأ و ماقبل مقصد و مقصد درخواست آمار جریان می‌دهد و سپس تعداد بسته‌های مرتبط را مقایسه می‌کند. با توجه به مکانیسم ایمن‌سازی شبکه در مراحل قبل و سنجش آمار مبدأ و مقصد مشخص است در صورتی که گرهی در مسیر، مقصود مخربی داشته باشد، حداقل در یکی از دو گره انتهایی مشخص خواهد شد. دلیل انتخاب دو گره این است که در صورتی که یکی از آنها آمار غیرواقعی و دستکاری‌شده ارائه دهد، در دیگری

Archive of SID.ir

Archive of SID.ir

Confidentiality_Verification
Input: Encrypted packet F of flow F, Hash table
Output: verification succeeds, otherwise fails;

1. if F is from an ingress switch, then
2. encrypt payload
3. store the encryption key in database
4. set encryption flag in data row to 1
5. return encrypted payload
6. else if F is from an egress switch, then
7. decrypt payload
8. return decrypted payload
9. End if

شکل ۶: الگوریتم شماره ۳.

Behaviour_Verification
Input: Reliability_Index (λ), a set of switches in flow path
Output: Verification Succeeds, otherwise fails and updated Reliability_Index (λ)

1. Retrieve stats from switches [0, n-1, n]
2. If (switches [0] == switches[n-1] && switches [0] == switch[n]) then
3. β = λ
4. samplingSpace = β*(size of n)
5. for (int i=0; i<samplingSpace; i++)}
6. randomSwitch = Random (min=2, max=size of(n)-2, samplingSpace)
7. if (switches[randomSwitch] == switches [0]) then
8. β = α*β
9. samplingSpace = β*(size of n)
10. remove switches[randomSwitch] from samplingSpace
11. If (β<y) then
12. λ = α*λ
13. Return true
14. end if
15. else if (switches[randomSwitch] =! switches [0]) then
16. λ = λ+1/2
17. samplingSpace = (size of n)
18. Return false
19. end for
20. else if (switches [0] =! switches[n]) then
21. λ = λ+1/2
22. samplingSpace = (size of n)
23. return false

شکل ۷: الگوریتم شماره ۴.

مشخص خواهد شد. با توجه به آنکه جامعه آماری مشخص است، اقدام به تعیین تعداد فضای نمونه می‌کنیم. سپس به‌صورت تصادفی و با حذف گزینه از کل جامعه آماری، گره‌های کاندیدا را برای بررسی انتخاب می‌کنیم. اگر مکانیسم از این مرحله موفقیت‌آمیز عبور نکرد اقدام به جستجوی تمامی گره‌ها خواهیم کرد؛ اما اگر همه شرطها درست رعایت شده باشند، با احتمال λ (که از لایه‌های قبلی مکانیسم برای کل شبکه تولید شده) برای هر سوئیچ انتخابی از سمت کنترلر آمار مورد نظر دریافت خواهد شد و با تعداد بسته‌های جریان در گره اول مقایسه می‌گردد. در صورتی که مقایسه موفقیت‌آمیز بود، β (که در ابتدا برابر با λ است) به مقدار ثابتی کاهش می‌یابد. این فرایند تا زمانی ادامه پیدا می‌کند که β از میزان مشخصی کوچک‌تر باشد و نهایتاً ما میزان اعتماد به شبکه را افزایش خواهیم داد و از مکانیسم خارج می‌شویم. اما در صورتی که طرفین یک مقایسه مساوی نباشند، ما میزان λ را به نصف تغییر داده و در ادامه مکانیسم، آمار تمامی گره‌ها و نه فقط یک جامعه نمونه را درخواست و بررسی خواهیم کرد. نکته قابل توجه آن است که در حلقه for، تعداد جستجوها به‌صورت پویا در نظر گرفته شده است، بدین معنا که هر مقایسه صحیح در تعداد گره‌های بعدی مورد نیاز به سنجش، تأثیرگذار خواهد بود.

۵- نتایج و تحلیل آزمایش‌ها

۵-۱ بهبود DYNAPFV

با توجه به فرضیات DYNAPFV، این مکانیسم از کد اصالت‌سنجی پیام استفاده می‌کند. این استفاده بدین صورت است که سوئیچی که قصد ارسال Packet-in به کنترلر را دارد، تمامی بخش‌های بسته را بدون رمزنگاری و به‌صورت خام ارسال می‌کند. باید توجه داشت که بر اساس فرضیات Openflow ارتباط مابین سوئیچ‌ها و کنترلر به‌وسیله پروتکل TLS ایمن شده‌اند [۲۰]. ما نیز در مدل پیشنهادی فرض می‌کنیم که این ارتباط ایمن است. همچنین همان طور که در بخش گذشته نیز مطرح گردید، استفاده از MAC با کلیدی متقارن که چرخه حیات آن صرفاً در کنترلر خواهد بود و تصمیم استفاده از آن، چه در packet-inهای سوئیچ‌های ورودی و چه خروجی، وابسته به خود کنترلر است؛- یا به مفهوم دیگر کلید مابین گیرنده و فرستنده‌ای توزیع نشده است و از کلید در هیچ زمانی برای رمزگشایی استفاده نخواهد شد- ویژگی امنیتی احراز هویت به‌صورت صحیح پیاده‌سازی نشده و رمزنگاری داده درهم‌شده عملاً تأثیری بر امنیت نخواهد داشت. به‌طوری که در صورت حمله بر یک کنترلر، کلید و پیام به‌سادگی به دست مهاجم خواهد رسید و حتی قابل تغییر خواهد بود.

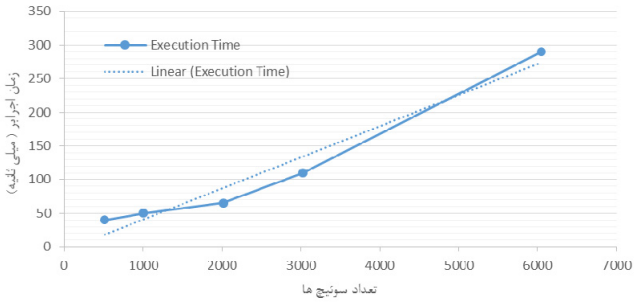
علاوه بر این با توجه به حجم بالای داده‌های در صف تصمیم‌گیری در شبکه، رمزنگاری داده‌های درهم‌شده در این حجم بالا، حتی با یک کلید متقارن (که زمان کمتری نسبت به کلیدهای نامتقارن مصرف خواهد کرد)، عملاً تأثیری برای امنیت ندارد و صرفاً باعث تحمیل سربار عملیاتی خواهد شد. در صورتی که صرف درهم‌شدن داده‌ها می‌توانست کافی باشد. اما مهم‌ترین مسئله در مکانیسم DynaPFV، عدم توجه به دو نوع حمله اساسی است که تأثیر مستقیمی بر صحت انتقال داده‌ها دارند:

۱) اولین نوع حمله همان طور که در مکانیسم Sphinx در فصل قبل مشاهده گردید، توپولوژی جعلی است. در این حمله که بسته‌های ARP، IGMP، LLDP و غیره در قالب Packet-in از سمت سوئیچ‌ها ارسال می‌شود، مهاجم نمای توپولوژی شبکه را ایجاد می‌کند. کنترل‌کننده‌ها پیام‌های LLDP را برای کشف توپولوژی و پیام‌های IGMP را برای نگهداری گروه‌های چندبخشی پردازش می‌کنند؛ در حالی که درخواست‌ها و پاسخ‌های ARP را می‌فرستند تا میزبان‌های نهایی را قادر به ایجاد کش‌های ARP سازند که ارتباطات شبکه را تسهیل می‌کند. مهاجم می‌تواند در قالب میزبان، پیام‌های مذکور را جعل نماید تا دیدگاه توپولوژی کنترل‌کننده را دستکاری کند و آن را فریب دهد تا قوانین جریان را برای انجام انواع حملات در شبکه بگذارد.

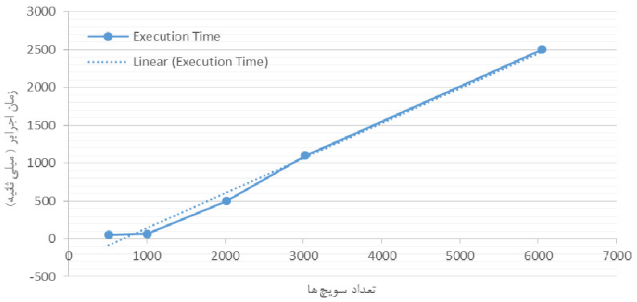
۲) دومین نوع حمله، از حملات ارسال در صفحه داده‌هاست. هاست‌ها و سوئیچ‌های مخرب می‌توانند با ایجاد سیل ترافیک در شبکه به میزبان‌های دلخواه، باعث ایجاد حملات DoS شوند تا منابع، سوئیچ‌های آسیب‌پذیر یا کنترلر SDN را به‌دست گیرند. این حمله‌ها در شبکه‌های SDN می‌توانند به‌صورت حمله به حافظه TCAM در سوئیچ‌ها باشند که یک حافظه فوق سریع برای نگهداری قوانین ارسال است. میزبان‌های مهاجم، سوئیچ TCAM را برای انجام حملات DoS مستقیم علیه میزبان‌های دیگر هدف قرار می‌دهند. همچنین می‌توانند با ایجاد سیلی از ترافیک در شبکه، کنترل‌کننده را مجبور به نصب تعداد زیادی از قوانین جریان کنند و در نتیجه TCAM سوئیچ را از پای درآورند. متعاقباً هیچ قانون جریان دیگری

Archive of SID.ir

Archive of SID.ir



شکل ۸: ارزیابی سرعت‌یافتن موجودیت مخرب در الگوریتم PaPFV.



شکل ۹: ارزیابی سرعت‌یافتن موجودیت مخرب در الگوریتم DYNAPFV.

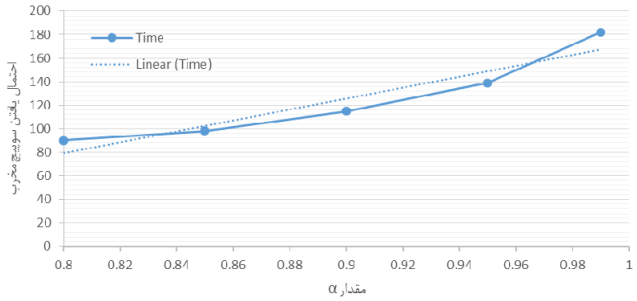
را نمی‌توان روی این سوئیچ نصب کرد؛ تا زمانی که جریان‌های نصب‌شده منقضی گردند. اگر این سوئیچ در یک مسیر بحرانی در شبکه باشد، ممکن است منجر به تأخیر قابل توجه یا افت بسته شود. این دو، حملاتی هستند که در ارتباط مستقیم با صحت ارسال بسته‌ها در شبکه می‌باشند و باید در یک مکانیسم جامع برای تأیید صحت ارسال بسته‌ها در نظر گرفته شوند.

۵-۲ نتایج آزمایش‌ها

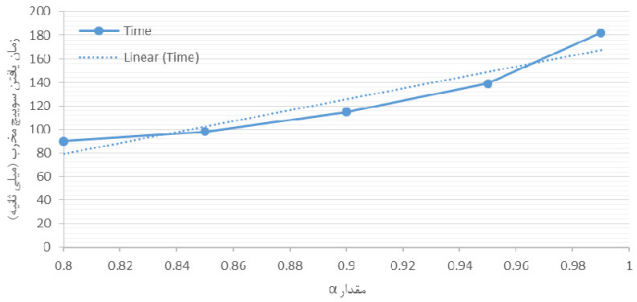
مدل پیشنهادی را پس از پیاده‌سازی کدها با زبان جاوا بررسی نمودیم و به‌دلیل پیچیدگی و چندوجهی‌بودن اجرای کدها در کنترلری مانند Floodlight از اجرای تمام بخش‌ها در کنترلر و Mininet خودداری کردیم. علاوه بر کدهای الگوریتم پیشنهادی، جهت مقایسه و به‌دلیل در دسترس‌نبودن کدهای DYNAPFV، بخش‌های مورد نیاز مکانیسم مذکور را نیز با زبان جاوا پیاده‌سازی نمودیم. ارزیابی سرعت و صحت عملکرد یافتن سوئیچ‌های مخرب هدف ارزیابی‌ها بود. محیط عملیاتی جهت اجرا نیز یک دستگاه کامپیوتر شخصی با پردازنده Intel با فرکانس ۲٫۹۰ گیگاهرتز و حافظه RAM ۱۲ گیگابایت بوده است.

نتایج ارزیابی برای سرعت یافتن موجودیت مخرب در شکل‌های ۸ و ۹ نشان داده شده است. در الگوریتم PaPFV، $y=۰٫۱$ $\alpha=۰٫۹۹$ هستند و به‌ترتیب y شاخص سنجش β برای خروج از حلقه جستجو و α ضریب کاهش β و λ است. این مقادیر بدین جهت انتخاب شده‌اند که شرایط هر مکانیسم مشابه و در زمان آغاز فعالیت باشند.

همان طور که در شکل‌های ۸ و ۹ مشخص است با افزایش سوئیچ‌ها، میزان زمان لازم برای یافتن سوئیچ مخرب در الگوریتم DYNAPFV به‌صورت تصاعدی افزایش می‌یابد. اما در الگوریتم پیشنهادی، زمان جستجو با شیبی منطقی و با فاصله بسیار زیاد از DYNAPFV است؛ به‌طوری که برای جستجو در بین ۶۰۰۰ گره، DYNAPF به ۲۵۰۰ میلی‌ثانیه زمان نیاز دارد؛ اما PaPFV در کمتر از ۲۰۰ میلی‌ثانیه به جواب خواهد رسید. نتیجه این آزمایش نشان می‌دهد که روش پیشنهادی نسبت به روش DYNAPF، زمان یافتن سوئیچ‌های مخرب را به میزان ۹۲٪ کاهش می‌دهد.



شکل ۱۰: احتمال یافتن سوئیچ مخرب با مقادیر α.



شکل ۱۱: زمان لازم برای یافتن سوئیچ مخرب با مقادیر α.

در ادامه، نتایج ارزیابی احتمال سنجش سوئیچ مخرب با تغییر α در شکل‌های ۱۰ و ۱۱ آمده و α از بازه بین ۰٫۸ تا ۰٫۹۹ متغیر انتخاب شده است. با افزایش α، امنیت سیستم افزایش یافته و احتمال کشف‌نشدن گره مخرب پایین می‌آید. در تمام آزمایش‌ها تعداد کل گره‌ها ۱۰۰ عدد است که از این تعداد، ۵ گره به‌صورت نامتوالی در فضای نمونه پخش شده‌اند. نتایج آزمایش بر اساس میانگین ۱۰۰ بار اجرا آمده‌اند.

طبق برآیند حاصل از نتایج آزمایش، پارامتر α نقشی تعیین‌کننده در احتمال کشف عنصر مخرب خواهد داشت؛ به‌طوری که افزایش α به ۰٫۹۹ که مقدار ماکسیمم است، ایمنی ۹۸ درصدی ارائه می‌کند.

در شکل ۱۰ سرعت اجرای الگوریتم پیشنهادی با مقدار متغیر α از بازه‌های بین ۰٫۸ تا ۰٫۹۹ نمایش داده شده است. در این آزمایش همچنان در جامعه نمونه، ۱۰۰ سوئیچ که از بین آنها ۹۵ عدد سالم و ۵ سوئیچ مخرب هستند، وجود دارد. نتایج بر اساس میانگین ۱۰۰ بار آزمایش حاصل آمده و نشان می‌دهند که با افزایش α، زمان لازم برای کشف گره مخرب افزایش می‌یابد. در نتیجه هرچه میزان α بیشتر باشد، امنیت سیستم و زمان لازم نیز بیشتر خواهد شد. بدین ترتیب مدیران شبکه باید برحسب نیاز شبکه، تعادلی را بین دو متغیر امنیت و زمان برقرار کنند.

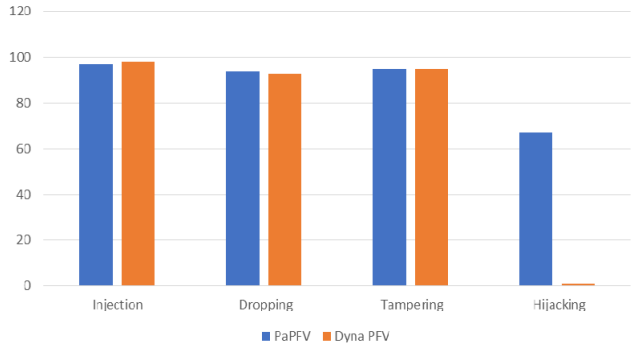
۵-۳ مقایسه مدل پیشنهادی با سایر روش‌های پیشین

روش پیشنهادی PaPFV در مقایسه با DYNAPFV قدرت مقابله بهتری در مقابل حملات دارد و از سربار منابع کمتری استفاده می‌کند. شکل ۸ نشانگر مقایسه عملکرد الگوریتم‌های PaPFV و DYNAPFV تحت صد بار حمله از انواع تزریق، حذف، دستکاری و ربایش بسته می‌باشد. لازم به ذکر است که داده‌های مرتبط با DYNAPFV از ادبیات خود مکانیسم و PaPFV به‌صورت عملیاتی برگرفته شده است.

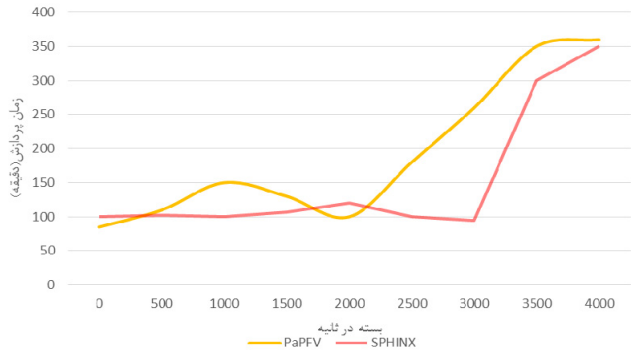
در شکل ۱۲ همان گونه که مشاهده می‌شود، عملکرد PaPFV در سه حمله تزریق، حذف و دستکاری بسته اندکی بهتر است؛ اما با توجه به آنکه DYNAPFV راه حلی برای مقابله با حملات ربایش بسته ارائه نکرده و قادر به تشخیص این نوع حملات نیست، در نمودار به‌صورت صفر نمایش داده شده است؛ اما PaPFV قادر به مقابله با ۶۰٪ حملات ربایش بسته به‌صورت موفقیت‌آمیز بوده است.

Archive of SID.ir

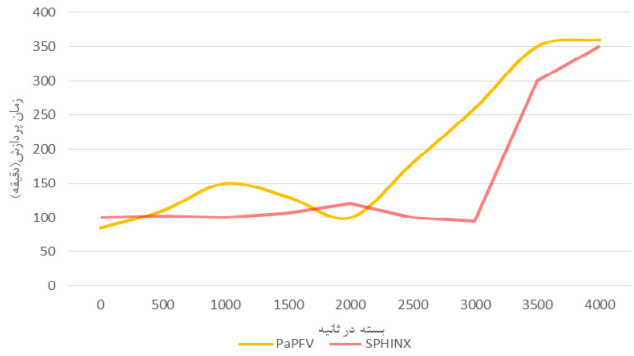
Archive of SID.ir



شکل ۱۲: مقایسه عملکرد الگوریتم‌های PaPFV و DYNAPFV تحت حملات مختلف.



شکل ۱۳: زمان پردازش PACKET_IN در PaPFFV و Sphinx.



شکل ۱۴: زمان پردازش FLOW_MOD در دو الگوریتم PaPFV و Sphinx.

حال شکل ۱۳ نمایانگر مقایسه دو مکانیسم Sphinx و PaPFV در بررسی زمان پردازش PACKET_IN در مقادیر صد و پانصدتایی است. در پردازش صد بسته، هر دو مکانیسم نتایج مشابه از خود به نمایش می‌گذارند؛ اما در پردازش پانصد بسته، عملکرد PaPFV بهتر است.

شکل ۱۴ نشان‌دهنده مقایسه زمان پردازش FLOW_MOD در دو الگوریتم PaPFV و Sphinx می‌باشد. طبق نتایج، میانگین عملکرد Sphinx در تعداد زیاد بسته کمی بهتره بوده؛ ولی نهایتاً دو مکانیسم به یکدیگر نزدیک شده‌اند.

هر یک از روش‌های تصدیق صحت ارسال بسته‌ها در شبکه‌های SDN دارای معایب و مزایایی هستند. جدول ۱ به‌طور خلاصه، مقایسه اجمالی بین روش‌های ذکرشده و روش پیشنهادی را نشان می‌دهد.

٦- نتیجه‌گیری

در این مقاله یک روش جدید برای صحت ارسال بسته‌ها در شبکه‌های SDN معرفی گردید. اگرچه روش‌های زیادی برای صحت ارسال بسته‌ها در شبکه‌های سنتی ارائه شده‌اند، اما در زمینه شبکه‌های جدید SDN، پژوهش‌های کمتری انجام شده است. دو روش مطرح DYNAPFV و NPFV که برای صحت ارسال بسته‌ها در شبکه‌های SDN معرفی شده‌اند از چندین چالش رنج می‌برند. زمان اجرای زیاد، حافظه مورد نیاز بالا، ضعف در مقابل حمله توپولوژی جعلی و ضعف در مقابل حملات ربایش بسته‌ها از نقاط ضعفی هستند که این روش‌ها از آنها رنج می‌برند. روش پیشنهادی نسبت به هر دو روش مطرح قبلی، سربار بسیار کمتری دارد و در زمان بسیار کمتری می‌تواند صحت ارسال بسته‌ها را بررسی کند. نتایج آزمایش‌ها نشان می‌دهند که این زمان به میزان بیشتر از ۹۰٪ کاهش می‌یابد. همچنین روش پیشنهادی، حمله توپولوژی جعلی را خنثی می‌کند و در مقابل حملات ربایش بسته‌ها مقاوم است.

در شبکه‌های SDN که از روش پیشنهادی می‌خواهند استفاده کنند، تعدادی از پارامترها در الگوریتم، قابلیت تغییرپذیری برای ایجاد تعادل بین سرعت، ایمنی و عملکرد شبکه را دارند. بدین صورت مدیران شبکه برحسب نیاز می‌توانند از قابلیت انعطاف‌پذیری سیستم بهره ببرند.

مراجع

[1] D. Kreutz, *et al.*, "Software-defined networking: a comprehensive survey," *Proceeding of the IEEE*, vol. 103, no. 1, pp. 14-76, Jan. 2015.

[2] Q. Li, X. Zou, Q. Huang, J. Zheng, and P. P. C. Lee, "Dynamic packet forwarding verification in SDN," *IEEE Trans. on Dependable and Secure Computing*, vol. 16, no. 6, pp. 915-929, Dec. 2019.

[3] M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "Sphinx: detecting security attacks in software-defined networks," in *Proc. of Network and Distributed System Security Symp., NDSS'15,* 15 pp., San Diego, CA, USA, 7-7 Feb. 2015.

[4] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114-119, Feb. 2013.

[5] M. Al Ahmad, M. Diab, and S. S. Patra, "Analysis and performance evaluation of openflow controller in SDN using N-policy," in *Proc. of Int. Conf. on Recent Advances in Science and Engineering Technology, ICRASET'23,* 5 pp., B G NAGARA, India, 23-24 Nov. 2023.

[6] X. Zhang, A. Jain, and A. Perrig, "Packet-dropping adversary identification for data plane security," in *Proc. of the ACM CoNEXT Conf.*, Article Id.: 24, 12 pp., Madrid, Spain, 9-12 Dec. 2008.

[7] H. J. Kim, C. Basescu, L. Jia, S. B. Lee, Y. C. Hu, and A. Perrig, "Lightweight source authentication and path validation," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 271-282, Aug. 2014.

[8] H. Beitollahi, D. M. Sharif, and M. Fazeli, "Application layer DDoS attack detection using cuckoo search algorithm-trained radial basis function," *IEEE Access*, vol. 10, pp. 63844-638542022.

[9] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks," in *Proc. of the ACM SIGSAC Conf. on Computer & Communications Security*, pp. 413-424, Berlin, Germany, 4-8 Nov. 2013.

[10] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Sustaining cooperation in multi-hop wireless networks," in *Proc. of the 2nd Conf. on Symp. on Networked Systems Design & Implementation*, vol. 2, pp. 231-244, 2-4 May 2005.

[11] R. Aryan, A. Yazidi, F. Brattensborg, O. Kure, and P. E. Engelstad, "SDN spotlight: a real-time openflow troubleshooting framework," *J. of Future Generation Computer Systems*, vol. 133, pp. 364-377, Aug. 2022.

[12] H. Yu, K. Li, and H. Qi, "An active controller selection scheme for minimizing packet-in processing latency in SDN," *J. of Security and Communication Networks*, vol. 2019, Article ID: 1949343, Oct. 2019.

[13] H. Wang, L. Xu, and G. Gu, "FloodGuard: A DoS attack prevention extension in software-defined networks," in *Proc. of 45th Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks,* pp. 239-250, Rio de Janeiro, Brazil, 22-25 Jun. 2015.

[14] T. Sasaki, C. Pappas, T. Lee, T. Hoefler, and A. Perrig, "SDNsec: forwarding accountability for the SDN data plane," in *Proc. of 25th Int. Conf. on Computer Communication and Networks, ICCCN'16,* 10 pp., Waikoloa, HI, USA, 1-4 Aug. 2016.

[15] X. Liu, A. Li, X. Yang, and D. Wetherall, "Passport: secure and adoptable source authentication," in *Proc. of the 5th USENIX Sympo.*

Archive of SID.ir

Archive of SID.ir

on Networked Systems Design and Implementation, pp. 365-378, San Francisco, CA, USA 16-18 Apr. 2008.

[16] Y. Chen, Y. Yang, X. Zou, Q. Li, and Y. Jiang, "Adaptive distributed software defined networking," *J. of Computer Communications*, vol. 102, pp. 120-129, Apr. 2017.

[17] S. Hong, R. Baykov, L. Xu, S. Nadimpalli, and G. Gu, "Towards SDN-defined programmable byod (bring your own device) security," in *Proc. of NDSS'16,* 15 pp., San Diego, CA, USA, 21-24 Feb. 2016.

[18] H. Hu, W. Han, G. J. Ahn, and Z. Zhao, "Flowguard: building robust firewalls for software-defined networks," in *Proc. of 3rd Workshop on Hot Topics in Software Defined Networking*, pp. 97-102, Chicago, IL, USA, 22-22 Aug. 2014.

[19] O. Blial, M. Ben Mamoun, and R. Benaini, "An overview on SDN architectures with multiple controllers," *J. of Computer Networks and Communications*, vol. 2016, Article ID: 9396525, Apr. 2016.

[20] D. Kreutz, F. M. V. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proc. of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, pp. 55-60, Hong Kong, China, 16-16 Aug. 2013.

روزبه بگلری تحصیلات خود را در مقطع کارشناسی در رشته مهندسی کامپیوتر در دانشگاه زنجان در سال ۱۳۹۸ به پایان رساند. سپس در سال ۱۴۰۱ در مقطع کارشناسی ارشد گرایش شبکه‌های کامپیوتری از دانشگاه علم و صنعت ایران فارغ التحصیل شد. زمینه‌های تحقیقاتی مورد علاقه ایشان، شبکه‌های SDN، امنیت شبکه، ارتقای پروتکل‌های شبکه و افزایش کارایی شبکه‌های می‌باشد.

حاکم بیت‌الهی در سال ۱۳۸۱ در مقطع کارشناسی در رشته مهندسی کامپیوتر دانشگاه تهران فارغ التحصیل شد. سپس در سال ۱۳۸۴ در مقطع کارشناسی ارشد گرایش معماری سیستم‌های کامپیوتری دانشگاه صنعتی شریف فارغ التحصیل گردید. برای تحصیلات دکترا به دانشگاه لوون بلژیک رفت و در سال ۱۳۹۲ در مقطع دکترا در این دانشگاه فارغ التحصیل شد. پس از آن یک سال به عنوان محقق پسا دکترا در دانشگاه لوون باقی ماند. در سال ۱۳۹۴ به عنوان عضو هیأت علمی در دانشگاه سوران، اقلیم کردستان، مشغول کار شد. سپس در سال ۱۳۹۷ به دانشگاه علم و صنعت ایران رفت و از آن تاریخ به عنوان عضو هیأت علمی و مدیرگروه معماری سیستم‌های کامپیوتری مشغول فعالیت هستند. نام‌برده تا کنون هدایت ۴ دانشجوی دکترا و حدود ۳۰ دانشجوی ارشد را بر عهده داشته است. زمینه‌های پژوهشی ایشان عبارتند از: طراحی شتابدهنده-های سخت‌افزاری برای هوش مصنوعی، امنیت شبکه‌های کامپیوتری و SDN، امنیت سخت‌افزار و سیستم‌های بی‌درنگ.

Archive of SID.ir