ارائه راه‌کاری امن جهت تائید تحرک میزبان‌ها در شبکه‌های نرم‌افزار محور (SDN) با استفاده از مکانیزم درهم‌سازی (Hash)

مجتبی قاسم‌زاده[1]، سید امین حسینی سنو[2]

[1] کارشناسی ارشد شبکه‌های کامپیوتری دانشگاه فردوسی مشهد
m.ghasemzadeh@mail.um.ac.ir

[2] دانشیار دانشکده مهندسی دانشگاه فردوسی مشهد
hosseini@um.ac.ir

چکیده

شبکه‌های نرم‌افزار محور (SDN)[1] با جداسازی صفحه کنترل[2] از صفحه داده[3] و متمرکز کردن آن در یک کنترل‌کننده مرکزی، معماری جدیدی در شبکه‌های کامپیوتری ایجاد کرده و امکان ایجاد یک دید کلی از شبکه را فراهم می‌سازد. برهمین اساس، کشف توپولوژی شبکه برای کنترل‌کننده‌های SDN جهت ایجاد یک دید متمرکز از شبکه، ضروری است. با این وجود، در زمان فرآیند شناسایی توپولوژی شبکه، کنترل‌کننده‌های SDN به دلیل عدم تائید هویت میزبان و سوئیچ از حملات مسمومیت توپولوژی و حمله سرقت میزبان رنج می‌برند. راه‌کارهای موجود به دلیل وجود ضعف در آن‌ها و همچنین ارائه راه‌کار پیچیده و سنگین وزن، نتوانسته‌اند این ضعف را به طور مناسب رفع نمایند. در این مقاله تلاش گردیده تا با ارائه راه‌کاری امن، ساده و سبک، یک ساختار تائید هویت مکانی و جابجایی میزبان با استفاده از مکانیزم تابع درهم‌سازی فراهم گردد. راه‌کار ارائه شده با استفاده از نرم‌افزار شبیه‌سازی Mininet و همچنین کنترل‌کننده Pox پیاده‌سازی شده است. نتایج آزمایشات در یک محیط شبکه‌ای مجازی Mininet نشان می‌دهد، راهکار ارائه شده در عین سادگی، تائید هویتی امن و سبک وزن را جهت شناسایی مکان و جابجایی میزبان‌ها فراهم می‌کند.

کلمات کلیدی

شبکه نرم‌افزار محور(SDN)، شناسایی توپولوژی، حملات مسمومیت توپولوژی، امنیت، حمله سرقت میزبان

۱- مقدمه

شبکه‌های نرم‌افزار محور با جداسازی صفحه کنترل از صفحه داده و متمرکز کردن صفحه کنترل در یک واحد مرکزی به نام کنترل‌کننده، معماری جدیدی را در حوزه شبکه‌های کامپیوتری ارائه نموده است. این معماری جدید قابلیت برنامه‌پذیری را به این‌گونه شبکه‌ها افزوده است [2,3]. بر همین اساس، شبکه نرم‌افزار محور زیرساخت شبکه اصلی را برای برنامه‌های کاربردی سطح بالا شفاف نگه می‌دارد. کنترل‌کننده SDN تجهیزات لایه پایین شبکه را جهت ارسال بسته‌ها هدایت و مدیریت می‌نماید و به شبکه‌های پیچیده اجازه می‌دهد تا بدون نیاز به مذاکره با تجهیزات لایه پایین شبکه، این تجهیزات را به صورت مرکزی مدیریت نمایند. در نتیجه، این شبکه‌ها قابلیت برنامه‌ریزی، انعطاف‌پذیری و مقیاس‌پذیری آسان در حوزه مدیریت شبکه را نسبت به روش‌های سنتی به همراه دارد. به طور خاص، شناسایی توپولوژی به عنوان اصلی‌ترین ویژگی کنترل‌کننده‌های SDN عمل می‌کند و به انواع برنامه‌های لایه بالا توانایی مسیریابی بسته‌ها، ردیابی تحرک، مهاجرت زنده، مجازی‌سازی شبکه و بهینه‌سازی شبکه را می‌دهد [4-6]. اطلاعات توپولوژی شبکه نه تنها پایه و اساس تولید و ارسال قوانین جداول جریان است، بلکه پیش‌نیاز مدیریت منابع شبکه در SDN نیز می‌باشد. در نتیجه، برای

کنترل‌کننده‌های SDN بسیار مهم است که از یکپارچگی، دقت و اعتبار اطلاعات توپولوژی شبکه اطمینان حاصل کنند.

به طور کلی، مکانیزم شناسایی توپولوژی شبکه عمدتاً به دو بخش شناسایی مکانی و ردیابی تحرک میزبان و شناسایی ارتباطات و خطوط ارتباطی بین تجهیزات شبکه در کنترل‌کننده SDN تقسیم‌بندی می‌شود. به دلیل عدم وجود مکانیزم تائید هویت توپولوژی، اعتبار این دو بخش نمی تواند در کنترل‌کننده مورد تأیید قرار گیرد. اطلاعات توپولوژی شامل اطلاعات سوئیچ‌ها، میزبان‌ها و پیوندهای بین این واحدها (یعنی پیوندهای سوئیچ با سوئیچ و سوئیچ با میزبان) می‌باشد. در نتیجه، یک مهاجم می‌تواند از ضعف نبود مکانیزم تائید هویت در شناسایی توپولوژی شبکه استفاده کرده و با استفاده از اطلاعات جعلی مربوط به سه بخش فوق، نسبت به مسموم‌سازی و فریب کنترل‌کننده SDN در شناسایی توپولوژی شبکه اقدام نماید. چنین اقداماتی که در بالا عنوان شد توسط مهاجم می‌تواند به راحتی منجر به جداسازی ارسال ترافیک مانند "سیاه چاله" و ربودن ترافیک و شنود ترافیک شود، که در نهایت باعث گمراهی کامل سرویس‌ها/برنامه‌های لایه بالایی می‌شود. بر این اساس، تأیید اطلاعات توپولوژی برای کنترل‌کننده‌های SDN برای محافظت از یک دید کلی در شبکه حیاتی است، و کلید دستیابی به حفاظت امنیتیSDN ، شناسایی سریع ورودی‌های این اطلاعات غیرقانونی می‌باشد.

چندین مکانیزم تائید هویت، جهت تائید مکانی و تحرک میزبان‌ها وجود دارد که به ارائه راه‌کارهایی جهت ارائه این ویژگی پرداخته‌اند. در اکثر راه‌کارهای عنوان شده، جهت شناسایی و تائید هویت میزبان‌ها از سه پارامتر آدرس فیزیکی میزبان، شماره پورت اتصال میزبان به سوئیچ و شماره سوئیچ متصل به میزبان و در برخی از راه‌کارها از پارامتر آدرس منطقی میزبان نیز استفاده شده است.

با استفاده از سه پارامتر ابتدایی ذکر شده می‌توان مکان میزبان را تعیین و در صورتی که نیاز به تحرک میزبان نباشد، با استفاده از این اطلاعات بسته‌های ارسالی را در شبکه بررسی نمود و اجازه جابجایی بسته‌ها با اطلاعات مکانی متفاوت را نداد. اما با توجه به پیشرفت شبکه‌های کامپیوتری و افزایش شبکه‌ها و تجهیزات بی‌سیم، نیاز به تحرک میزبان‌ها روز به روز افزایش می‌یابد. بر این اساس، نیازمند راه‌حلی برای تائید هویت میزبان‌های متحرک جهت شناسایی و جلوگیری از حملاتی که در این حوزه می‌باشد، هستیم.

برای رفع این نیاز هانگ و همکارانش [7] راه‌حلی به نام TopoGuard را معرفی کردند که اعتبار بروزرسانی‌های توپولوژی را با افزودن ویژگی‌های اضافی (از قبیل نوع دستگاه و لیست میزبان) به هر پورت سوئیچ در کنترل‌کننده OpenFlow تشخیص می‌دهد. سپس اطلاعات توپولوژی را با استفاده از ویژگی‌های اختصاصی و ارتباط آن‌ها به هر کدام از پورت‌های سوئیچ تائید هویت می‌نماید. چنین ویژگی‌هایی نشان می‌دهد که ترافیک پذیرفته شده آیا توسط یک رابط متصل به سوئیچ است یا میزبان. در نتیجه جهت شناسایی مسیر اصلی میزبان و بروزرسانی آن از بسته‌های پویش استفاده می‌گردد. شیانگ و همکارانش [8] با استفاده از یک مدل رسمی به بررسی راه‌کار TopoGuard می‌پردازند و نقاط ضعف این راه‌کار را بررسی و بیان می‌دارند. با توجه به بررسی‌های انجام شده در این مقاله راه‌حل TopoGuard نمی‌تواند حملات سرقت میزبان در زمان مهاجرت میزبان هدف را شناسایی کند.

دهاون و همکارانش [9] راه‌حلی به نام SPHINX را معرفی کرده‌اند که از گراف‌های جریان برای تائید گام به گام تمام بروزرسانی‌ها و محدودیت‌های شبکه استفاده می‌کند. SPHINX با استفاده از این گراف‌های جریانی، رفتارهای مخرب در صفحه داده را شناسایی می‌نمایند. براین اساس، سکویرا و همکارانش [10] به بررسی راه‌کارهای TopoGuard و SPHINX در مقابل حملات Port Probing و Port Amnesia پرداخته و نقاط ضعف این راه‌کارها را در مقابله با این حملات بیان کرده و به طراحی، اجرا و ارزیابی اقدامات مورد نیاز در برابر این حملات می‌پردازند. هانگ و همکارانش [11] با معرفی راه‌کاری به نام TrusTopo تلاش نموده‌اند با استفاده از مکانیزم اعلان وضعیت پورت در شبکه‌های SDN به تائید هویت میزبان جابجا شده بپردازند. گاوو و همکارانش [12] با استفاده از مکانیزم پویش میزبان از طریق بسته‌های ARP و ICMP به تائید هویت میزبان جابجا شده می‌پردازند. بر این اساس، زمانی که میزبان جابجا شد و در مکان جدید نسبت به ارسال بسته اقدام می‌نماید، مکانیزم معرفی شده مکان قبلی میزبان را پویش کرده و در صورت عدم حضور در مکان قبلی، آن را معتبر می‌داند.

بایدیا و همکارانش [13] با استفاده از پروتکل شناسایی لینک LLDP و شناسایی پورت ارتباطی با میزبان و در ادامه با ارسال بسته‌های ARP برای این پورت‌ها نسبت به شناسایی میزبان اقدام نموده و در صورت جابجایی آن، اقدام به ارسال بسته‌های ARP به مکان قبلی میزبان جهت تائید موقعیت جدید آن می‌نماید. موتاهر و همکارانش [14] با استفاده از پروتکل تائید اعتبار Kerberos نسبت به شناسایی میزبان در شبکه SDN اقدام می‌نماید. محمدیفر و همکارانش [15] با معرفی مکانیزم SPV نسبت به تائید هویت میزبان اقدام می‌نمایند. در این روش، جهت تائید هویت میزبان از ارسال بسته‌های پویش به صورت نامحسوس اقدام نموده و در صورتی که میزبان به این بسته‌ها پاسخ صحیح داد، آن را معتبر شناسایی می‌کند.

با توجه به راه‌کارهای عنوان شده در مقالات فوق برخی نقاط ضعف در این راه‌کارها مشاهده می‌گردد. یکی از مهمترین نقاط ضعف در این راه‌کارها عدم تائید هویت مناسب میزبان در زمان تحرک آن است، به گونه‌ای که در صورتی که مهاجم در زمان جابجایی میزبان هدف، خود را به عنوان آن میزبان معرفی نماید، به عنوان میزبان معتبر شناسایی شده و می‌تواند به هدف خود و انجام سایر حملات در شبکه دست یابد. از طرفی، برخی از راهکارهای عنوان شده مدت زمان تائید اعتبار زیادی را در زمان تحرک میزبان صرف خواهد کرد. بر این اساس جهت رفع نقاط ضعف مقالات فوق و همچنین کاهش زمان و ساده سازی مکانیزم تائید هویت میزبان، در این مقاله به معرفی مکانیزم تائید هویت میزبان متحرک با استفاده مکانیزم تابع درهم‌سازی در شبکه‌های SDN می‌پردازیم.

در ابتدا با در نظر گرفتن خطوط ارتباطی بین سوئیچ‌ها و و میزبان با سوئیچ، جداولی ایجاد می‌گردد که پایه و اساس تصمیم‌گیری‌ها برای ترافیک منتقل شده در آینده می‌باشد. لذا نوع این جداول و داده‌های ذخیره شده در این جداول باعث بهبود زمان تصمیم‌گیری در مورد نحوه برخورد با بسته‌های دریافتی خواهد شد. به گونه‌ای که با افزایش تعداد میزبان‌ها و سوئیچ‌های موجود در شبکه، زمان پاسخگویی و تصمیم‌گیری برای بسته‌ها کاهش خواهد یافت. در ادامه میزبان برای انجام یک تحرک امن و انجام تائید هویت قوی، در ابتدای تحرک خود یک کلید تصادفی ۲۵۶ بیتی هش شده با الگوریتم

SHA256 را برای کنترل‌کننده SDN ارسال می‌کند که مبنای تائید هویت میزبان در آینده خواهد بود.

راه‌کار ارائه شده در این مقاله به شرح زیر خلاصه می‌شود:

- ثبت اولیه اطلاعات مکان میزبان در زمان ورود به شبکه
- ایجاد و استفاده از جداول مناسب جهت انجام تصمیم‌گیری به لحظه و مطمئن در برخورد با بسته‌های متفاوت در شبکه
- ارائه راه‌کاری امن، ساده و سریع جهت تائید هویت میزبان‌های متحرک در شبکه
- بررسی راه‌کار ارائه شده در مقابله با حمله سرقت میزبان

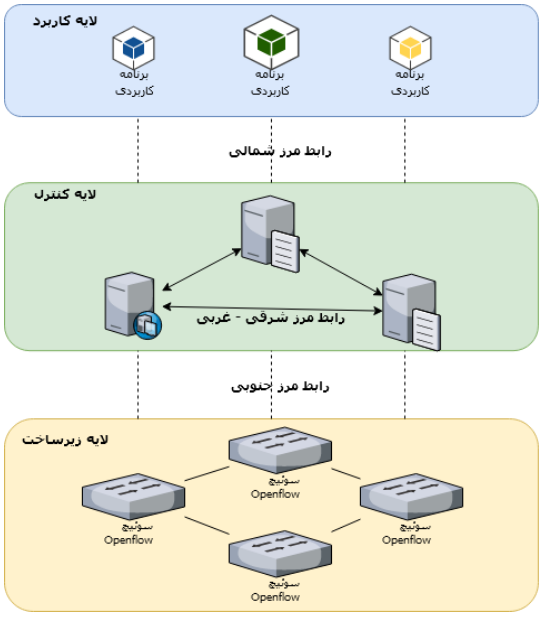
بر همین اساس در ادامه مقاله ما به بخش‌هایی شامل بخش ۲ به عنوان معرفی تعاریف اولیه، بخش ۳ به عنوان تعریف مسئله، بخش ۴ ارائه راه‌حل پیشنهادی، بخش ۵ ارزیابی شبیه‌سازی انجام شده و در نهایت بخش ۶ نتیجه‌گیری تقسیم‌بندی می‌شود.

۲- تعاریف اولیه

این بخش به معرفی اجمالی شبکه نرم‌افزار محور، پروتکل Open Flow و حمله سرقت میزبان می‌پردازد.

۲-۱- شبکه نرم‌افزار محور

شبکه نرم‌افزار محور (SDN) با جداسازی صفحه داده از صفحه کنترل و متمرکز کردن صفحه کنترل در یک کنترل‌کننده مرکزی قابلیت برنامه‌نویسی را به شبکه اضافه کرده است [۱]. کنترل کننده SDN یا صفحه کنترل با بخش سخت‌افزاری (صفحه داده) از طریق رابط‌های برنامه‌نویسی به نام رابط مرز جنوبی ارتباط برقرار می‌کند. رایج‌ترین پروتکل مورد استفاده، با هدف ارتباط در SDN پروتکل OpenFlow است. همانطور که در شکل ۱ مشاهده می‌کنید معماری شبکه SDN از سه لایه اصلی یعنی صفحه داده، صفحه کنترل و صفحه کاربری تشکیل شده است.

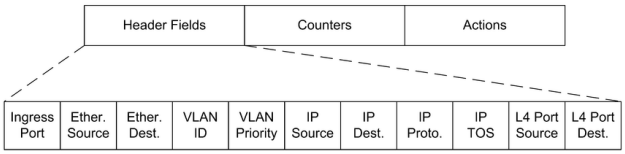


شکل (۱) : معماری شبکه SDN [۱]

بر این اساس رابط مرز جنوبی این امکان را فراهم می‌کند تا یک سوئیچ SDN براساس جداول جریان موجود در خود نسبت به نحوه برخورد با بسته‌های دریافتی اقدام یا اطلاعات بسته‌های دریافتی خود را برای تصمیم‌گیری در مورد آن به کنترل‌کننده ارسال کند. کنترل‌کننده SDN با توجه به ارتباطی که با سایر برنامه‌های کاربردی از طریق رابط مرز شمالی و ارتباطی که با سرویس‌های موجود در شبکه از طریق رابط‌های مرز غربی و شرقی دارد و همچنین عملکرد تعیین شده برای کنترل‌کننده می‌تواند برای بسته‌های دریافتی تصمیم‌گیری نماید.

۲-۲- پروتکل OpenFlow

OpenFlow یکی از پرکاربردترین و مهمترین رابط‌های مرز جنوبی در شبکه SDN می‌باشد. این رابط نحوه ارسال و قالب پیام را برای تعامل بین سوئیچ OpenFlow و کنترل‌کننده SDN تعریف می‌کند. برخلاف سوئیچ‌های سنتی، در OpenFlow تصمیم‌گیری در مورد بسته‌ها براساس ورودی‌های جریان در جدول جریان سوئیچ می‌باشد. هر جدول جریان در سوئیچ شامل مجموعه‌ای از ورودی‌های جریان است. یک ورودی جریان شامل سرآیندهای مطابقت، شمارنده‌ها و مجموعه‌ای از دستورالعمل‌هایی است که برای مطابقت بسته‌ها مورد استفاده قرار می‌گیرند [16]. در شکل ۲ این سرآیندها را مشاهده می‌کنید.



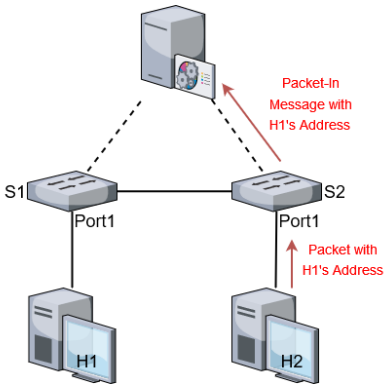
شکل (۲) : سرآیندهای OpenFlow [16]

هنگامی که یک سوئیچ یک بسته را دریافت می‌کند، شروع به مطابقت جدول جریان با توجه به سرآیندهای انطباق با سرآیندهای بسته دریافتی می‌کند. در صورت تطبیق، بسته براساس دستورالعملی که در ورودی جریان تطبیق داده شده است پردازش می‌شود. در غیر این صورت، بسته مطابق با ورودی عدم تطبیق در جدول جریان ارسال می‌شود. به طور پیش فرض، این‌گونه بسته‌ها از طریق پیام‌های Packet-In به کنترل کننده ارسال می‌شوند. پیام OpenFlow نقش مهمی در ارتباط بین سوئیچ و کنترل‌کننده برعهده دارد. سه نوع پیام برای پیام‌های OpenFlow وجود دارد: ۱- پیام کنترل‌کننده به سوئیچ، ۲- پیام‌های غیر همزمان، ۳- پیام‌های متقارن. پیام‌های کنترل‌کننده به سوئیچ مانند پیام‌های Packet-Out و Features توسط کنترل‌کننده آغاز می‌شوند و برای مدیریت یا کنترل سوئیچ استفاده می‌شوند. پیام‌های غیر همزمان توسط سوئیچ آغاز می‌شوند تا رویداد تغییر وضعیت سوئیچ را به کنترل‌کننده اطلاع دهند مانند پیام‌های Packet-In و پیام‌های Port-Status. پیام‌های متقارن می‌توانند توسط یک سوئیچ یا یک کنترل‌کننده برای تکمیل اتصالات ارسال و یا دریافت شوند. همچنین از این پیام‌ها برای پردازش برخی مشکلات بین سوئیچ‌ها و کنترل‌کننده استفاده می‌شود. هنگامی که کنترل‌کننده پیام‌های Packet-In را دریافت می‌کند که حاوی بسته‌های عدم تطابق هستند، سرویس ارسال‌کننده مسیری را برای این بسته‌ها محاسبه می‌کند. اما اگر مسیری برای این بسته‌ها وجود نداشته باشد، کنترل‌کننده به

سوئیچ دستور می‌دهد که این بسته‌ها را از طریق پیام‌های Packet-Out به پورت‌های دیگر منتقل کند.

۳-۲- حمله سرقت میزبان

در شبکه‌های SDN با میزبان‌های متحرک، جهت شناسایی و ردیابی تحرک میزبان‌ها از سرویس ردیابی میزبان استفاده می‌شود. این سرویس با نظارت بر بسته‌های Packet-IN ارسالی توسط سوئیچ‌ها که حاوی اطلاعات مکانی میزبان است، تحرک میزبان‌ها را نظارت می‌نماید. اما به دلیل وجود برخی ضعف‌های امنیتی در این سرویس، این سرویس به طور دقیق نمی‌تواند اطلاعات مکانی را تائید و میزبان را تائید هویت نماید. لذا هرگونه اطلاعات مکان میزبان ممکن است برروی سرویس ردیابی میزبان تاثیر گذاشته و تحرک میزبان را غیرقابل اعتماد نماید. یکی از حملاتی که از این ضعف امنیتی استفاده می‌کند، حمله سرقت میزبان می‌باشد. در شکل ۳ نحوه انجام این حمله به تصویر کشیده شده است.



شکل (۳) : حمله سرقت میزبان

اطلاعات مکانی میزبان‌های H1 و H2 براساس آدرس منطقی و شماره پورت سوئیچ و شماره سوئیچ برروی سرویس ردیابی میزبان ذخیره شده است. قبل از شروع حمله، سرویس ردیابی میزبان اطلاعات مکان میزبان H1 را به عنوان پورت ۱ در سوئیچ S1 ثبت می‌کند. اما با توجه به عدم وجود مکانیزم تائید هویت میزبان در این سرویس، میزبان H2 با دستکاری مقادیر بسته‌های ارسالی خود و تغییر آدرس فیزیکی خود به آدرس فیزیکی میزبان H1 خود را به جای این میزبان با ارسال این بسته‌ها معرفی می‌کند. سپس این بسته از میزبان H2 به سوئیچ S2 ارسال می‌شود. در نتیجه، کنترل‌کننده SDN در نظر می‌گیرد که میزبان H1 به پورت ۱ در سوئیچ S2 مهاجرت کرده و سپس کنترل‌کننده جداول مربوط به این میزبان را به روز می‌کند. در نتیجه کنترل‌کننده تمام بسته‌های ارسالی به سمت میزبان H1 را برای مهاجم یعنی میزبان H2 ارسال می‌کند.

برای مقابله با حمله ربودن میزبان، راه‌حل‌های موجود عمدتاً به دو نوع زیر طبقه‌بندی می‌شوند، ۱- تائید هویت موجودیت میزبان، ۲- بررسی قانونی بودن مهاجرت میزبان

۳- تعریف مسئله

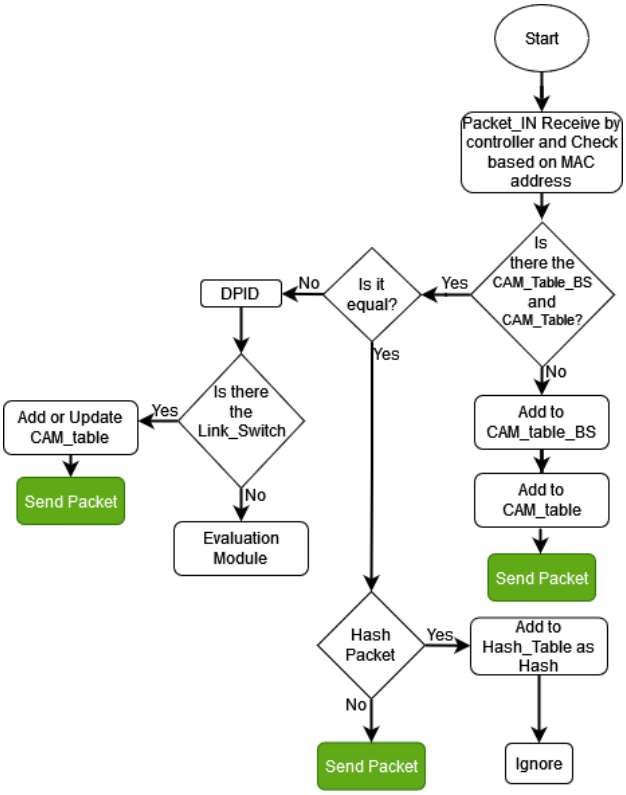
با توجه به مطالب عنوان شده در قسمت‌های قبل، گسترش روز افزون شبکه‌های پیشرفته و میزبان‌های متحرک نیاز به تائید هویت میزبان‌های متحرک را افزایش داده است. از این رو، مقابله با حملاتی که تحرک میزبان‌ها را به چالش کشیده و از نقاط ضعف این تحرک سوء استفاده می‌کنند، مهم و حیاتی است. در شبکه‌های SDN به دلیل افزوده شدن قابلیت برنامه‌ریزی در شبکه و تمرکز واحد مدیریت در یک کنترل‌کننده مرکزی، این شبکه‌ها بسیار مورد توجه و استفاده قرار گرفته و با توجه به ویژگی‌های عنوان شده برای این شبکه‌ها، توانایی مقابله با حملات به میزبان‌های متحرک را افزایش داده است. در ساده‌ترین راه در این شبکه‌ها برای مقابله با این‌گونه حملات، جلوگیری از تحرک میزبان‌ها است، که این امر با ذات پویا و متحرک شبکه در تضاد می‌باشد. از طرفی با استفاده از اطلاعاتی که شبکه‌های SDN در اختیار مدیر قرار می‌دهد، بویژه پروتکل OpenFlow امکان پایش و تائید هویت میزبان را فراهم می‌کند. بر همین اساس راه‌حل‌های مطرح گردید که در بخش تعاریف به معرفی برخی از این راه‌حل‌ها پرداختیم [7-15]. از مهمترین نقاط ضعفی که در این راه‌حل‌ها مشاهده می‌شود، عدم تائید هویت دقیق میزبان‌ها می‌باشد. به گونه‌ای که در زمان تحرک میزبان و قبل از اتصال میزبان به شبکه، در صورتی که مهاجم از قطع اتصال میزبان هدف با شبکه اطلاع پیدا کند، می‌تواند خود را به عنوان آن میزبان معرفی نماید و تائید اعتبار گردد [8,10]. علاوه بر آن برخی از راه‌حل‌های عنوان شده، در مقابل برخی از حملات که آن‌ها را ذکر کردیم آسیب‌پذیر می‌باشند [10]. از طرفی، در برخی از راه‌حل‌های عنوان شده با افزایش زمان پردازش تائید هویت و بار پردازشی برروی کنترل‌کننده همراه هستیم [12-14]. از این رو نیاز است جهت مقابله با حملات این حوزه، راه‌حلی معرفی گردد تا علاوه بر دقت بالا در تائید هویت میزبان متحرک، از مکانیزم ساده و سبک وزنی برخوردار باشد.

۴- راه‌حل پیشنهادی

در این بخش به معرفی راه‌کار ارائه شده در این مقاله می‌پردازیم. در این راه‌حل به دلیل اینکه خطوط ارتباطی بین سوئیچ‌ها برروی شناسایی تحرک میزبان‌ها تاثیرگذار هستند، مورد توجه قرار گرفته و با استفاده از مکانیزم‌های امن نظیر [11] یا سایر راه‌حل‌های معرفی شده در جهت شناسایی و بروزرسانی این خطوط ارتباطی، این خطوط شناسایی گردیده و مورد استفاده قرار می‌گیرد. در این راه‌کار ۴ جدول جهت شناسایی خطوط و وضعیت میزبان مورد استفاده قرار می‌گیرد. در این راه‌کار کنترل‌کننده، سوئیچ و خطوط بین کنترل‌کننده و سوئیچ امن در نظر گرفته می‌شود. همانطور که در شکل ۴ مشاهده می‌کنید، نمودار جریان، نحوه استفاده از جداول و نحوه تصمیم‌گیری در مورد بسته‌های مختلف نشان داده شده است.

خطوط ارتباطی بین سوئیچ‌ها در جدولی به نام Link_Switch ذخیره می‌گردد. در ابتدا وقتی میزبان برای اولین بار وارد شبکه می‌شود، اطلاعات مکانی میزبان براساس آدرس فیزیکی، شماره پورت سوئیچ متصل به میزبان و شماره سوئیچ مذکور در جدول CAM_Table_BS و در جدول CAM_Table براساس شماره سوئیچ، آدرس فیزیکی و شماره پورتی که این بسته از آن دریافت شده است، ذخیره می‌گردد. در ادامه با دریافت بسته‌هایی از این میزبان، اطلاعات این بسته‌ها در صورتی که از روی یک خط متصل به میزبان دریافت گردد با این جداول بررسی می‌گردد و براساس آن نحوه برخورد با بسته تصمیم‌گیری می‌شود. در صورتی که بسته از یک خط ارتباطی بین دو سوئیچ با توجه به بررسی جدول Link_Switch دریافت گردد، نادیده گرفته شده و فقط در جدول CAM_Table براساس اطلاعات

ذکر شده ذخیره یا بروزرسانی می‌گردد. در ادامه در صورتی کـه میزبـان قصـد تحرک و جابجایی در شبکه را داشته باشد، مکانیزم تائید هویت جهـت انجـام این تحرک را اجرا می‌کند.



شکل (۴) : نمودار جریان روش ارائه شده

میزبان متحرک قبل از انجام تحرک و جابجایی یک کلمـه عبـور ۲۵۶ بیـت تصادفی تولید کرده و آن را توسط تابع درهم‌سـازی بـا الگـوریتم SHA256 هش می‌نماید. در ادامه میزبان نام کاربری تصادفی ایجاد شده را نزد خود نگه می‌دارد و کلمه عبور هش شده را قبل از جدا شدن از شبکه برای کنترل‌کننده ارسال می‌کند. کنترل‌کننده با دریافت بسته کلمه عبور هش شده از سـوئیچی که مستقیماً بـه آن میزبـان متصـل اسـت، آن را در جـدول Hash_Table متناسب بـا آدرس فیزیکـی میزبـان ذخیـره می‌کنـد. میزبـان پـس از انجـام جابجایی و اتصال مجدد به شبکه، نسبت به ارسال کلمه عبور تصادفی بـدون انجام عملیات هش برای کنترل‌کننده اقدام می‌نماید.

کنترل‌کننده با توجه به عدم انطباق اطلاعات بسته دریافـت شـده بـا جـداول CAM_Table_BS و CAM_Table وارد ماژول ارزیـابی تائیـد هویـت میزبان می‌شود. با توجه به اینکه اولین بسته دریافتی از سـمت میزبـان معتبـر پس از جابجایی بسته کلمه عبور تصادفی می‌باشد که برای کنترل‌کننده ارسال شده است، لذا کنترل‌کننده کلمه عبور تصادفی دریافـت شـده را بـا الگـوریتم SHA256 هش کرده و با مقدار هش دریافت شده قبلی مقایسه می‌نماید. در صورت برابر بودن مقادیر مربوطـه، میزبـان معتبـر شناسـایی شـده و جـداول CAM_Table_BS و CAM_Table براساس اطلاعـات مکـانی جدیـد کاربر بروزرسانی شده و مقادیر هش و کلمه عبور دریافتی از میزبان از جـدول Hash_Table حذف می‌گردد. در صورتی میزبان دوبـاره قصـد جابجـایی و تحرک را داشته باشد، نسبت به ایجاد مجدد کلمه عبـور تصـادفی ۲۵۶ بیتـی جدید اقدام می‌نماید و فرآیند ذکر شده را مجددا طی خواهد کرد.

۵- ارزیابی

در این بخش به ارزیابی راه‌کار ارائه شـده در ایـن مقالـه می‌پـردازیم. جهـت ارزیابی راه‌کار ارائه شده در این مقاله ما از شبیه‌ساز شبکه‌های SDN بـه نـام Mininet استفاده کرده‌ایم [17]. راه‌کار ارائه شده در این مقالـه بـا اسـتفاده کنترل‌کننده Pox که بر پایه برنامه‌نویسی با زبان پایتون می‌باشد، اجرا گردیده است [18]. جهت محاسبه زمان پاسخ راه‌کار ارائه شده، ما از توپولوژی ۳ لایه (Access-Distributed-Core) استفاده کرده‌ایم.

در این راه‌کار به دلیل تاثیرگذاری خطوط ارتبـاطی بـین سـوئیچ‌ها در تحـرک میزبان مورد توجه قرار می‌گیرد. در نتیجه راه‌کار به گونـه‌ای طراحـی گردیـده است تا با انواع مختلف روش‌های ارائه شده در زمینه شناسایی خطوط ارتباطی بین سوئیچ‌ها سازگار بوده و بتواند از این روش‌ها استفاده کند. خطوط ارتباطی بین سوئیچ‌ها در یک جدول به نام Link_Switch ذخیره می‌گردد و راه‌کـار ارائه شده در این مقاله به دلیل در نظر گرفتن پایش و تصـمیم‌گیری در مـورد بسته‌ها در ابتدای دریافـت بسـته از سـوئیچ‌های مـرزی متصـل بـه میزبـان، بسته‌هایی که از روی این خطوط دریافت می‌کند را نادیده گرفته و بدون تائید هویت آن‌ها در جدول CAM_Table ذخیره یا بروزرسانی می‌کند. لـذا ایـن امر باعث افزایش سرعت تصـمیم‌گیری در شـبکه‌ها بـه ویـژه در شـبکه‌های بزرگتر و پیچیده‌تر می‌گردد. از طرفی در نظـر گـرفتن ایـن خطـوط ارتبـاطی قابلیت بکارگیری راه‌کـار ارائـه شـده در ایـن مقالـه را بـا انـواع سـاختارهای شبکه‌های مختلف فراهم می‌کند.

در راه‌کـار ارائـه شـده در ایـن مقالـه از جـداول CAM_Table_BS و CAM_Table اسـتفاده می‌گـردد. در جـدول CAM_Table_BS تنهـا اطلاعات مکانی میزبان (خط ارتبـاطی بـین میزبـان و سـوئیچ) شـامل آدرس فیزیکی میزبان، شماره پورت اتصالی میزبان به سوئیچ و شماره سوئیچ متصـل به میزبان ثبت و سایر اطلاعات خطوط ارتباطی نظیـر خطـوط ارتبـاطی بـین سوئیچ‌ها در این جدول ذخیره نمی‌گردد. لذا با اسـتفاده از ایـن جـدول امکـان حمله سرقت میزبان از روی سایر سوئیچ‌ها به حداقل مقدار خود می‌رسد و بـه شدت کاهش می‌یابد. با استفاده از جدول CAM_Table و ذخیره اطلاعات مکانی میزبان براساس شماره سوئیچ، آدرس فیزیکی و شماره پورت متصل به آن سوئیچ، مسیر حرکت بسـته مشـخص گردیـده و باعـث افـزایش سـرعت تصمیم‌گیری و ارسال بسته خواهد شد. با توجه به راه‌کـار ارائـه شـده در ایـن مقاله استفاده از یک کلمه عبور تصادفی ۲۵۶ بیتی امکان یـافتن آن را تقریبـا غیرممکن می‌نماید.

علاوه بر این، ارسال کلمه عبور به صورت هش شده با در نظر گرفتن امکـان شنود در شبکه، دستیابی به کلمه عبور را کاهش داده و ارسـال مسـتقیم ایـن مقدار و دریافت آن توسط کنترل‌کننده از اولین سوئیچ متصل به میزبان، امکان شنود توسط مهاجم را از بین خواهد برد. استفاده کلمه تصادفی جدیـد در هـر تحرک نیز توسط میزبان بر امن‌تر شدن راه‌کار ارائه شده اضافه می‌کند. بـرای بررسی راه‌کار ارائه شده در نحوه برخورد با میزبان معتبر و میزبان مهاجم ما از توپولوژی نشان داده شده در شکل ۳ استفاده کرده‌ایم.

در ابتدا همانطور که در شکل ۵ مشاهده می‌کنید میزبـان H1، بـرای میزبـان H2 بسته پویش ارسال می‌کند و میزبان H2 به آن پاسخ می‌دهـد. در ادامـه

میزبان H1 از روی سوئیچ S2 با استفاده از راه‌کار ارائه شده در این مقاله جدا شده و به سوئیچ S1 متصل می‌گردد و بسته پویش برای H2 ارسال می‌کند. با توجه به تائید هویت صحیح H1، بسته‌های ارسالی مورد تائید قرار گرفته و برای H2 ارسال شده و میزبان H2 به آن پاسخ می‌دهد.

```
INFO:forwarding.Switch:PacketIn: Port:1 DPID:2 ETH
 MAC.src:00:00:00:00:00:01 => MAC.dst:00:00:00:00:
00:02 and IP.src:10.0.0.1 => IP.dst:10.0.0.2
INFO:forwarding.Switch:PacketIn: Port:2 DPID:2 ETH
 MAC.src:00:00:00:00:00:02 => MAC.dst:00:00:00:00:
00:01 and IP.src:10.0.0.2 => IP.dst:10.0.0.1
Port 1 on Switch 2 has been removed.
Port 21 on Switch 1 has been added.
INFO:forwarding.Switch:PacketIn: Port:21 DPID:1 ET
H MAC.src:00:00:00:00:00:01 => MAC.dst:00:00:00:00
:00:02 and IP.src:10.0.0.1 => IP.dst:10.0.0.2
INFO:forwarding.Switch:PacketIn: Port:2 DPID:2 ETH
 MAC.src:00:00:00:00:00:02 => MAC.dst:00:00:00:00:
00:01 and IP.src:10.0.0.2 => IP.dst:10.0.0.1
```

شکل (۵) : جابجایی میزبان H1 و پویش میزبان H2 توسط آن

در ادامه به بررسی مقابله راه‌کار ارائه شده در مقاله در مقابل حمله سرقت میزبان می‌پردازیم. همانطور که در شکل ۶ مشاهده می‌کنید، میزبان H1 برروی پورت شماره ۲ در سوئیچ S2 قصد دارد خود را به عنوان میزبان H2 در همان سوئیچ و متصل به پورت شماره ۳ معرفی کرده و بسته پویش برای یکی از میزبان‌ها در شبکه ارسال کند. با توجه به اینکه میزبان H2 در شبکه هیچ‌گونه تحرکی انجام نداده و فرآیند تائید هویت صورت نگرفته است. در نتیجه میزبان H1 به عنوان مهاجم شناسایی شده و اجازه جابجایی بسته در شبکه به آن داده نمی‌شود.

```
INFO:forwarding.Switch:PacketIn: Port:2 DPID:3001
ETH MAC.src:00:00:00:00:00:02 => MAC.dst:00:00:00:
00:00:10 and IP.src:10.0.0.2 => IP.dst:10.0.0.10
The host 00:00:00:00:00:02 in port 2 in dpid 3001
is not allowed to move or is invalid.
```

شکل (۶) : مقابله با حمله سرقت میزبان

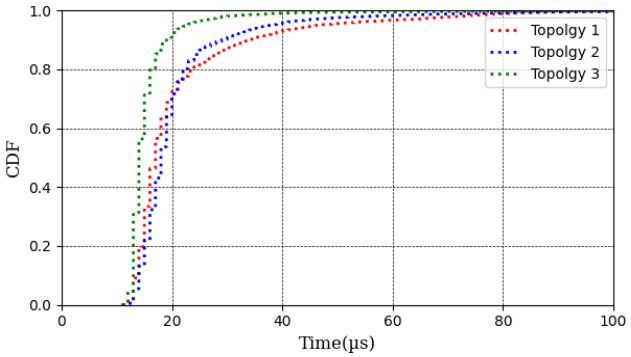
در ادامه ما به بررسی تاخیر پردازش در راه‌کار ارائه شده در این مقاله می‌پردازیم. نتایج تجربی با استفاده از تابع توزیع تجمعی CDF مقایسه می‌شوند، که نشان‌دهنده احتمال کمتر یا مساوی بودن یک متغیر تصادفی با یک مقدار معین است [19].

مقیاس توپولوژی در این آزمایش در جدول ۱ نشان داده شده است. هر سوئیچ لبه به ۲۵ میزبان متصل است. جهت بررسی زمان تاخیر پردازش در زمان حمله یک مهاجم، ما میزبان H1 به عنوان مهاجم در نظر می‌گیریم. این میزبان مهاجم شروع به ارسال ۱۰۰۰۰ بسته به فاصله زمانی ۰/۰۱ ثانیه می‌کند.

جدول (۱) : مقیاس توپولوژی

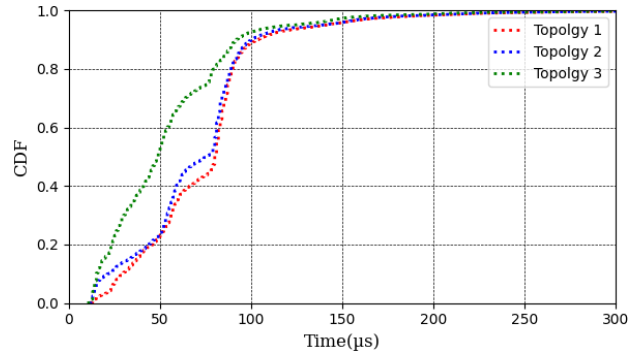
توپولوژی	سوئیچ	میزبان
توپولوژی ۱	۷	۱۰۰
توپولوژی ۲	۱۳	۲۰۰
توپولوژی ۳	۲۵	۴۰۰

نتایج تجربی براساس تأخیر پردازش راه‌کار ارائه شده در مواجه با بسته‌های معتبر و بسته‌های نامعتبر در شکل‌های ۷ و ۸ به ترتیب نشان داده شده است. در شکل ۷ مدت زمان تاخیر پردازش بسته‌های معتبر در شبکه نشان داده شده است. همانطور که مشاهده می‌کنید ۹۰ درصد زمان تاخیر پردازش کمتر از ۴۰ میکروثانیه بوده و با افزایش مقیاس شبکه به کمتر از ۲۰ میکروثانیه کاهش می‌یابد که این مدت زمان در شبکه‌های کامپیوتری زمان بسیار ناچیزی می‌باشد.



شکل (۷) : CDF تأخیرهای اندازه‌گیری شده در بسته‌های معتبر

در شکل ۸ مدت زمان تاخیر پردازش بسته‌های نامعتبر و حذف آن‌ها نشان داده شده است. با توجه به راه‌کار ارائه شده در این مقاله در ۹۰ درصد، مدت زمان تاخیر پردازش کمتر از ۱۰۰ میکروثانیه خواهد بود. علاوه بر این با افزایش مقیاس شبکه، مدت زمان تاخیر کاهش پیدا خواهد کرد.



شکل (۸) : CDF تأخیرهای اندازه‌گیری شده در بسته‌های نامعتبر

۶- نتیجه‌گیری

گسترش روز افزون شبکه‌های کامپیوتری و افزایش تحرک میزبان‌ها در این شبکه‌ها، نیاز به تائید هویت میزبان‌ها را در زمان انجام این تحرک از اهمیت ویژه‌ای برخوردار کرده است. راه‌کارهای ارائه شده، با توجه به فرآیند انجام تائید هویت و نواقصی که در آن‌ها وجود دارد و از طرفی پیچیدگی برخی از این راه‌کارها و زمان پردازش بالا آن‌ها، اجرای این راه‌کارها را با چالش مواجه کرده است. لذا در این مقاله تلاش گردید راه‌کاری ارائه گردد تا مکانیزم تائید هویت با امنیت بالا انجام و این مکانیزم ساده و سبک وزن باشد. بر همین اساس راه‌کار ارائه شده به دلیل استفاده از یک کلمه تصادفی ۲۵۶ بیتی، استفاده از الگوریتم SHA256، دریافت بسته کلمه هش شده و کلمه تصادفی از اولین سوئیچ متصل به میزبان و عدم جابجایی این بسته‌ها در شبکه، در نظر

گرفتن اهمیت خطوط ارتباطی بین سوئیچ‌ها و ایجاد تصمیم‌گیری متفـاوت در نحوه برخورد با بسته‌های که از روی این خطوط دریافت می‌گردد و همچنـین پردازش فرآیند تائید هویت در اولین سوئیچی که با میزبان در ارتباط بوده، که باعث کاهش عبور ترافیک نامعتبر در شبکه می‌شود، می‌تواند امنیت مورد نیاز را در این زمینه تا حد قابل قبولی تامین نماید. علاوه بر این، ویژگی‌های فـوق امکان استفاده از راه‌کار ارائه شده در شبکه‌های مختلف با ویژگی‌های مختلف فراهم نموده است. برای کارهای آینده، افزودن روش‌های مختلف تائید هویت در شبکه را می‌توان در نظر گرفت تا علاوه بر بهبود کارایی بیشـتر و افـزایش سطح امنیت، بتوان قابلیت انعطاف‌پذیری بیشتر را در اختیار مدیران شبکه قرار داد.

پیوست ها

واژه میزبان ذکـر شـده در عنـوان و مـتن مقالـه عبـارت اسـت از تجهیـزات سخت‌افزاری که در شبکه توسط کاربران مورد استفاده قرار گرفته و از طریـق آدرس فیزیکـی، شناسـایی و از یکـدیگر متمـایز می‌گردنـد. ماننـد: کـامپیوتر رومیزی، کامپیوتر قابل حمـل، تلفن‌هـای هوشـمند، تلفن‌هـای تحـت شـبکه، سرورها و .....

مراجع

[۱] قاسم زاده، حسینی سنو، "حفاظت از شبکه‌های نرم‌افزار محور (SDN) در مقابل حملات ARP با استفاده از سرور DHCP، بیست و ششمین کنفرانس انجمن کامپیوتر ایران، ۱۳۹۹

[2] B. Nunes, M. Mendonca, X. Nguyen, K. Obraczka and T. Turletti, "*A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks*", IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1617-1634, 2014.

[3] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky and S. Uhlig, "*Software-Defined Networking: A Comprehensive Survey*," Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, 2015.

[4] Z. Shu et al., "*Traffic engineering in software-defined networking: Measurement and management*," IEEE Access, vol. 4, pp. 3246-3256, 2016.

[5] A. Alsaeedy and E. Chong, "*A review of mobility management entity in LTE networks: Power consumption and signaling overhead*", International Journal of Network Management, vol. 30, no. 1, 2019.

[6] QIN, Yiling, et al. "*Interference and topology-aware VM live migrations in software-defined networks*." In: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2019.

[7] S. Hong, L. Xu, H. Wang and G. Gu, "*Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures*", Proceedings 2015 Network and Distributed System Security Symposium, 2015.

[8] S. Xiang, H. Zhu, L. Xiao and W. Xie, "*Modeling and Verifying TopoGuard in OpenFlow-Based Software Defined Networks*", 2018 International Symposium on Theoretical Aspects of Software Engineering (TASE), 2018.

[9] M. Dhawan, R. Poddar, K. Mahajan and V. Mann, "*SPHINX: Detecting Security Attacks in Software-Defined Networks*", Proceedings 2015 Network and Distributed System Security Symposium, 2015.

[10] SKOWYRA, Richard, et al. "*Effective topology tampering attacks and defenses in software-defined networks*". In: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 2018.

[11] X. Huang, P. Shi, Y. Liu and F. Xu, "*Towards trusted and efficient SDN topology discovery: A lightweight topology verification scheme*", Computer Networks, vol. 170, p. 107119, 2020.

[12] S. Gao, Z. Li, B. Xiao and G. Wei, "*Security Threats in the Data Plane of Software-Defined Networks*", IEEE Network, vol. 32, no. 4, pp. 108-113, 2018.

[13] S. Baidya and R. Hewett, "*Detecting host location attacks in SDN-based networks*", 2020 29th Wireless and Optical Communications Conference (WOCC), 2020.

[14] H. Mutaher and P. Kumar, "*Security-Enhanced SDN Controller Based Kerberos Authentication Protocol*", 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021.

[15] A. Alimohammadifar et al., "*Stealthy Probing-Based Verification (SPV): An Active Approach to Defending Software Defined Networks Against Topology Poisoning Attacks*", Computer Security, pp. 463-484, 2018.

[16] OpenFlow. Accessed: Nov. 2021. [Online]. Available: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[17] Mininet. Accessed: Nov. 2021. [Online]. Available: https://github.com/mininet/mininet/wiki/Documentation

[18] Pox Controller. Accessed: Nov. 2021. [Online]. Available: http://openflow.stanford.edu/display/ONL/POX+Wiki

[19] Cumulative Distribution Function, Accessed: Nov. 2021. [Online]. Available: https://en.wikipedia.org/wiki/Cumulative_distribution_function.

پانویس ها

[1] Software Defined Network

[2] Control Plane

[3] Data Plane