مجله علمی- پژوهشی "پدافند الکترونیکی و سایبری"
سال پنجم، شماره ۴، زمستان ۱۳۹۶، ص ۴۱-۲۹

ارائه الگوریتمی مبتنی بر فاصله هلینگر برای تشخیص و کاهش اثر حملات منع خدمت توزیع شده در شبکه‌های نرم افزار محور

مژگان قصابی[۱]، محمود دی‌پیر[۲*]، ابراهیم مهدی‌پور[۳]

۱-دانشجوی کارشناسی ارشد، دانشگاه آزاد اسلامی، واحد علوم و تحقیقات تهران، گروه کامپیوتر، تهران ، ایران ۲- استادیار، دانشکده رایانه و فناوری اطلاعات، دانشگاه علوم فنون هوایی شهید ستاری، تهران ۳-استادیار، دانشگاه آزاد اسلامی، واحد علوم و تحقیقات تهران، گروه کامپیوتر، تهران
(دریافت: ۹۵/۱۰/۲۹، پذیرش: ۹۶/۰۵/۰۱)

چکیده

شبکه‌های نرم افزار محور برای ایجاد تغییر در معماری شبکه‌های سنتی با عملکرد اختصاصی جهت رسیدن به شبکه‌های هوشمند به وجود آمده‌اند. اخیراً این نوع شبکه‌ها، به‌دلیل انعطاف‌پذیری در مدیریت سرویس‌های شبکه و کاهش هزینه‌های عملیاتی در بین سازمان‌ها محبوبیت خاصی پیدا کرده‌اند. در معماری این شبکه‌ها، سیستم عامل و برنامه‌های کاربردی از سطح سوئیچ‌های شبکه جدا شده و در یک لایه مجازی تحت عنوان کنترل کننده، متمرکز شده است. این معماری به‌دلیل تصمیم‌گیری متمرکز و محدودیت منابع کنترل‌کننده در معرض انواع تهدیدات از جمله حملات منع خدمت توزیع شده قرار دارد. ما در این مقاله، معماری شبکه‌های نرم افزار محور و حملات منع خدمت توزیع شده در این معماری را بررسی کرده و با بهره‌گیری از امکانات منحصر به فرد کنترل‌کننده، الگوریتم جدیدی برای تشخیص و کاهش اثر این حملات ارائه داده‌ایم. ما در این الگوریتم پیشنهادی از رابطه آماری فاصله هلینگر و روش تطبیق متحرک میانگین وزنی به منظور شناسایی حملات منع خدمت توزیع شده در شبکه‌های نرم افزار محور استفاده کرده‌ایم. در این مقاله، حملات منع خدمت توزیع شده در شبکه نرم افزار محور توسط مقلد مینی‌نت به همراه کنترل‌کننده Pox شبیه سازی شده است. آزمایش‌ها و ارزیابی‌های انجام شده در این محیط، کارآیی الگوریتم پیشنهادی و برتری آن نسبت به روش‌های قبلی را نشان می‌دهند.

واژه‌های کلیدی: شبکه‌های نرم افزار محور، حملات منع خدمت توزیع شده، فاصله هلینگر، بخش کنترل، بخش داده.

۱- مقدمه

شبکه‌های سنتی با محدودیت‌هایی مانند پیچیدگی، سیاست‌های متناقض، فقدان مقیاس‌پذیری، وابستگی به تولیدکنندگان و وجود نداشتن هماهنگی بین نیازهای بازار و قابلیت‌های شبکه مواجه هستند. سیر تکامل اینترنت، پدید آمدن سرویس‌های ابری، مجازی‌سازی[1] و تغییرات در الگوهای مصرف داده موجب آشکار شدن ضعف‌ها و محدودیت‌های موجود در شبکه‌های سنتی شده است. با ظهور شبکه‌های نرم افزار محور[2] امیدهای تازه‌ای برای حل مشکلات ساختاری در شبکه‌های سنتی به وجود آمد [۱]. شبکه‌های نرم افزار محور یک معماری نوظهوری است که تا حد زیادی محدودیت‌های شبکه‌های سنتی را برطرف کرده است [۲]. به کارگیری این شبکه‌ها وابستگی به تجهیزات سخت‌افزاری را کاهش داده و موجب صرفه‌جویی در هزینه‌ها می‌شود. زیرساخت شبکه‌های نرم افزار محور، به دلیل دارا بودن دانش متمرکز، مدیریت را تسهیل و انعطاف‌پذیری و پویایی زیادی را به محیط شبکه می‌بخشد [۳].

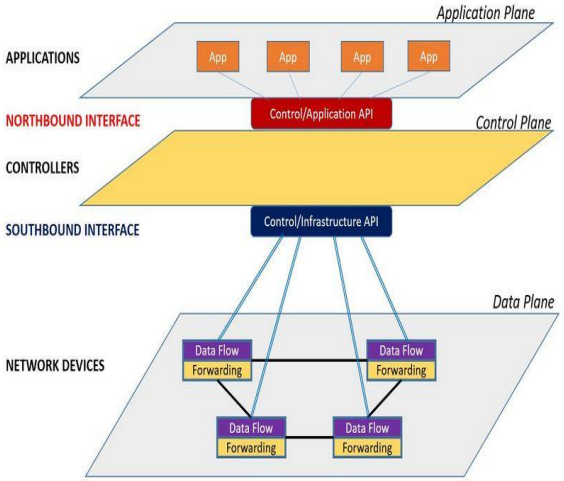
شبکه‌های نرم افزار محور، یک طرح اولیه است که مطابق شکل (۱) کنترل کننده اصلی سیستم به صورت یک واحد متمرکز از بستر انتقال داده، جدا شده و نرم افزارها در لایه بالایی و بستر ارتباطی شبکه در لایه زیرین قرار گرفته است [۴]. لایه کنترل ارتباط مابین برنامه‌های کاربردی بالا و دستگاه‌های قسمت پایین معماری را از طریق کانال امن فراهم می‌کند. در واقع لایه کنترل، کنترل کننده شبکه می‌باشد. کنترل کننده نمای متمرکز شبکه را حفظ می‌کند و از طریق رابط‌های باز به برنامه‌های کاربردی اجازه کنترل اصول زیربنایی شبکه را می‌دهد. در این معماری، جزئیات لایه پایینی از دید نرم افزارهای لایه بالاتر، پنهان شده است. این ویژگی سبب می‌شود که سازمان‌های بزرگ و توسعه‌دهندگان زیرساخت شبکه ضمن افزایش قابلیت

* رایانامه نویسنده مسئول: mdeypir@ssau.ac.ir

1 - Virtualition
2- Software-defined Networks

گسترش‌پذیری و انعطاف پذیری شبکه بتوانند با برنامه‌ریزی، ساختار شبکه را مطابق با نیازمندی‌ها اصلاح و به‌روزرسانی نمایند [۵].

در این معماری برای ایجاد ارتباط امن بین بخش داده و بخش کنترل نیاز به برخی پروتکل‌های خاص است. راهبری توسعه شبکه‌های نرم افزار محور و تدوین استانداردهای مربوطه بر عهده بنیاد شبکه‌های باز[1] است که یک تشکل غیرانتفاعی می‌باشد. از جمله مشهورترین استانداردهای تدوین شده توسط این بنیاد، پروتکل Open Flow است که چگونگی برقراری ارتباط بین بستر کنترلی و بستر ارتباطی تجهیزات مورد نیاز در شبکه‌های نرم افزار محور را تبیین می‌کند [۶].



شکل (۱): ساختار معماری شبکه نرم افزار محور [۵]

در ساختار معماری شبکه‌های نرم‌افزار محور، دستگاه‌های شبکه از جمله سوئیچ‌ها و مسیریاب‌ها فقط مسئولیت ارسال بسته‌ها بر اساس قوانین بخش کنترل کننده متمرکز را برعهده دارند. در این شبکه‌ها، بخش کنترل کننده به‌عنوان مغز این معماری، اطلاعات سوئیچ‌ها و مسیریاب‌ها را پردازش کرده و برای پیکربندی شبکه تصمیم‌گیری می‌کند و قوانین مربوط به نحوه هدایت بسته‌ها را به دستگاه‌های موجود در مسیر ارسال می‌نماید. دستگاه‌های شبکه، دارای چندین جدول جریان می‌باشند که این جداول، اطلاعات و قوانین مربوط به ورودی‌ها را حفظ می‌کنند [۷]. هنگامی‌که اولین بسته از جریان جدید از طرف فرستنده به سوئیچ شبکه نرم افزار محور می‌رسد، سوئیچ سرآیند بسته را با اطلاعات جریان‌های موجود در جداول خود بررسی می‌کند. اگر تطابقی در جدول سوئیچ یافت؛ اعمال به‌روزرسانی شمارنده و فیلدهای تخصیص داده شده به جریان مذکور (بسته/ تطبیق، فراداده) اجرا می‌شود و سپس بسته به مقصد ارسال می‌گردد. درصورتی‌که تطابقی در جدول جریان سوئیچ یافت نشود؛ بسته از طریق کانال امن تحت عنوان بسته packet- in به کنترل کننده ارسال می‌شود. کنترل کننده طبق ماژول‌های خود بسته را پردازش کرده و قوانین لازم که شامل سه عمل اضافه کردن، حذف کردن و به‌روزرسانی می‌باشد را برای این بسته صادر می‌کند. کنترل کننده با اجرای یک الگوریتم مسیریابی، اطلاعات و قوانین مربوط به ورودی جدید را به جداول جریان سوئیچ‌های موجود در مسیر آن جریان ارسال می‌نماید. سپس سوئیچ، بسته را مطابق با قوانین دریافتی از طریق پورت مناسب به سمت مقصد ارسال می‌کند. در صورت ورود جریان مشابه، دستگاه‌های شبکه، بسته را مطابق قوانین ثبت شده در جداول به مقصد هدایت خواهند کرد [۸].

معماری شبکه‌های نرم افزار محور با به‌روزرسانی پویای جداول جریان دستگاه‌ها، یک شبکه‌ای با انعطاف‌پذیری بالا و مدیریت آسان را ایجاد می‌نماید. اگرچه شبکه‌های نرم افزار محور مزایای زیادی را فراهم می‌کنند ولی همین ویژگی‌های پویا ممکن است برخی مسایل امنیتی را به وجود آورد. مهاجمان در شبکه‌های نرم افزار محور می‌توانند با سوءاستفاده از ویژگی به روزرسانی پویای جداول دستگاه‌ها با ارسال بسته‌های جعلی منابع شبکه را مشغول و موجب سرریز جداول گردند [۷]. کنترل کننده متمرکز این معماری برای مهاجمان مخرب یک هدف بسیار ارزشمند است، زیرا با دراختیار گرفتن این بخش می‌توانند کل شبکه را در دست بگیرند. همچنین، مهاجمان با اشغال منابع محدود کنترل کننده و کانال امن توسط بسته‌های جعلی می‌توانند بهره‌وری شبکه را مختل ساخته و در نهایت موجب از کار افتادن کل شبکه شوند.

یکی از تهدیدات جدی برای شبکه‌های نرم افزار محور حملات منع خدمت توزیع شده[2] است که به‌صورت مستقیم بر کنترل کننده این شبکه‌ها تأثیر می‌گذارد. هدف مهاجم در این نوع حمله غرق کردن کنترل کننده با بسته‌های packet-in است. اغلب حملات منع خدمت توزیع شده از آدرس مبدأ جعلی استفاده می‌کنند. این بسته‌ها به عنوان بسته ورودی جدید در سوئیچ‌ها تفسیر شده و برای پردازش و تعیین وضعیت به کنترل‌کننده ارسال خواهند شد. یکی دیگر از تأثیرات حملات منع خدمت توزیع شده در شبکه‌های نرم افزار محور، پرکردن جدول جریان سوئیچ است. در این حملات به‌دلیل حجم بالای ترافیک ورودی، جداول از جریان‌های جعلی پر می‌گردد [۸].

1- Open Networking Foundotion

2- Distributed Denial of Service Attack

هدف اصلی ما در این مقاله، تشخیص و کاهش اثر حملات منع خدمت توزیع شده در شبکه‌های نرم افزار محور است. برای رسیدن به این هدف، یک روش مؤثر و متناسب با ساختار و ویژگی‌های کنترل کننده شبکه‌های نرم افزار محور پیشنهاد شده است. ادامه این مقاله به‌شرح زیر سازمان‌دهی شده است. در بخش دوم کارهای مرتبط انجام شده در زمینه شبکه‌های نرم‌افزار محور و تشخیص حملات منع خدمت توزیع شده در این شبکه‌ها بررسی می‌شود. در بخش سوم به بیان مسأله پرداخته می‌شود. در بخش چهارم راه حل پیشنهادی و چگونگی استفاده از روش فاصله هلینگر در تشخیص حملات منع خدمت توزیع شده شرح داده می‌شود. بخش پنجم به ارزیابی روش پیشنهادی پرداخته و روش پیشنهادی با سایر روش‌های ارائه شده مقایسه می‌گردد. در بخش ششم جمع‌بندی و نتیجه‌گیری ارائه می‌شود.

۲- مروری بر کارهای انجام شده

مقابله با حملات منع خدمت توزیع شده در شبکه‌های سنتی از چندین دهه پیش مورد مطالعه قرار گرفته است. در [۹]، از طریق تقویت جریان‌های ترافیک محتمل‌تر، کاستن فضای جستجو و تغییر در انتخاب گره پایانی به منظور بهبود الگوریتم فرا ابتکاری مورچگان برای ردیابی حملات منع خدمت پرداخته شده است. در [۱۰]، با استفاده از تحلیل رفتار مرورگری، یک سازوکار دفاعی پویا با قابلیت سفارشی سازی برای تشخیص روبات‌های وب مخرب مشارکت کننده در حملات منع خدمت توزیع شده ارائه شده است. در [۱۱]، یک سازوکار دفاعی با استفاده از روش صف عادلانه وزن‌دار، جهت تشخیص و مقابله با حملات منع خدمت توزیع شده در شبکه‌های مبتنی بر SIP ارائه شده است. اگرچه هدف ما نیز مقابله با این حملات می‌باشد ولی محیط شبکه مورد بررسی ما، کاملاً از لحاظ معماری با شبکه‌های سنتی متفاوت می‌باشد.

در بررسی‌های مربوط به شبکه‌های نرم افزار محور برخی مقالات این معماری را معرفی کرده [۱۲] و به بررسی اجزا و رابط‌های این معماری پرداختند [۱۳] و مشخصات شبکه‌های نرم افزار محور و قابلیت‌های کنترلی این ساختار و تکامل آن را مورد مطالعه قرار داده‌اند [۱۴]. پژوهش‌ها در زمینه امنیت مبتنی بر شبکه‌های نرم افزار محور در آغاز راه است که دلیل این امر می‌تواند جدید بودن این معماری باشد. شین و همکاران [۱۵] یک چارچوب توسعه نرم‌افزار امنیتی به منظور ارتقاء محیط شبکه نرم افزار محور ارائه دادند. پورس و همکاران [۱۶] یک هسته اجرایی امنیتی برای تشخیص تضاد سیاستی در داخل سوئیچ‌های شبکه نرم افزار محور پیشنهاد دادند.

حملات منع خدمت در شبکه‌های نرم افزار محور در برخی از تحقیقات اخیر مورد توجه قرار گرفته است. در [۱۷]، ویژگی‌های معماری شبکه‌های نرم افزار محور در برابر حملات بررسی شده است و اظهار شده که این معماری به‌دلیل ویژگی‌های ساختاری در مقابله با حملات منع خدمت توزیع شده می‌تواند نقش اساسی را ایفا نماید. در [۱۸]، روش ناهمگن براساس همکاری سرویس دهنده سنتی و کنترل کننده مبتنی بر نرم افزار محور برای مقابله با حملات منع خدمت توزیع شده پیشنهاد شده است. در [۱۹]، تهدید اشباع منابع کنترل کننده و در [۲۰] تهدید ازدحام کانال امن و سرریز جداول جریان را مورد بررسی قرار دادند. در این مقالات روشی براساس تغییر و اصلاح ساختار سوئیچ‌ها برای مقابله با این تهدیدات ارائه شده است. در حالی که روش مقابله‌ای بهینه روشی است که در سمت کنترل کننده و بدون نیاز به تغییر ساختار سوئیچ‌ها انجام گیرد. در [۲۱] برای مقابله با حملات منع خدمت، یک شبکه نرم افزار محور مبتنی بر دیوار آتش بر بالای کنترل کننده POX پیشنهاد شده است. در این روش از امکانات شبکه نرم افزار محور استفاده نشده و این شبکه همچون شبکه‌های سنتی درنظر گرفته شده است. در [۲۲] یک سیستم تشخیص نفوذ را در شبکه نرم افزار محور پیشنهاد دادند. این روش به دلیل نصب و استقرار سیستم در بخش کنترل شبکه، بخشی از منابع محدود کنترل کننده را مورد استفاده قرار می‌دهد.

در [۲۳] روشی بر اساس کنترل کننده Nox برای تشخیص حملات منع خدمت توزیع شده پیشنهاد شده است که با استفاده از شبکه‌های عصبی بدون ناظر، براساس ویژگی ترافیک ورودی، حملات را تشخیص می‌دهد. در این طرح روش‌های پاسخ به حملات بررسی نشده است. یکی از ایرادات اصلی این طرح افزایش تأخیر ارسال بسته‌های مجاز به کنترل کننده در اثر احراز صحت همه بسته‌های ورودی است. در [۲۴] برای کاهش اثر حملات منع خدمت توزیع شده در شبکه نرم افزار محور، روش صف چندلایه منصفانه (MLFQ)[1] پیشنهاد شده است. ایراد این روش این است که هنگامی‌که تعداد سوئیچ‌ها و میزبان‌های شبکه زیاد باشد؛ نیاز به حافظه زیادی برای نگهداری صف‌ها خواهد بود.

در [۲۵] برای تشخیص حملات منع خدمت توزیع شده در شبکه‌های نرم افزار محور، روش یادگیری ماشین نقشه‌های خود سازمانده (SOM)[2] پیشنهاد شده است. در این روش، ماشین خود سازمانده رفتارهای شبکه را از طریق جمع‌آوری آمار جریان‌ها از سوئیچ‌های Open Flow آموزش می‌بینند. این روش برای یادگیری

1- Multi-Layer Fair Queueing
2- Self Organization Map

رفتار شبکه به ساعت‌ها آموزش نیاز دارد و بایستی ماتریس محاسباتی در کل دوره، محافظت شود. یکی دیگر از مشکلات این روش این است که این راه حل در کنار کنترل کننده برای تشخیص حملات استفاده می‌شد و از امکانات کنترل کننده بی‌بهره بود. در این مقاله به تأثیرات بسته‌های حمله به کنترل کننده و نحوه ارسال بسته‌های حمله اشاره‌ای نشده است. در [۸] سازوکار تشخیص حملات منع خدمت توزیع شده مبتنی بر روش آماری آنتروپی در شبکه‌های نرم افزار محور ارائه شده است. در این روش برای جمع‌آوری اطلاعات از امکانات کنترل کننده متمرکز استفاده شده و آنتروپی براساس IP آدرس مقصد بسته‌های ورودی، محاسبه می‌شود. اگر مقدار آنتروپی در پنج دوره متوالی از مقدار حد آستانه کمتر باشد؛ حمله تشخیص داده می‌شود. ایراد اصلی این روش ثابت بودن مقدار حد آستانه و عدم توانایی در تشخیص حملاتی با توزیع شدگی یکنواخت بین میزبان‌ها می‌باشد. همچنین، در این روش فقط حمله تشخیص داده می‌شود و هیچ اقدام مقابله‌ای در برابر حملات انجام نمی‌گیرد.

۳- بیان مسأله

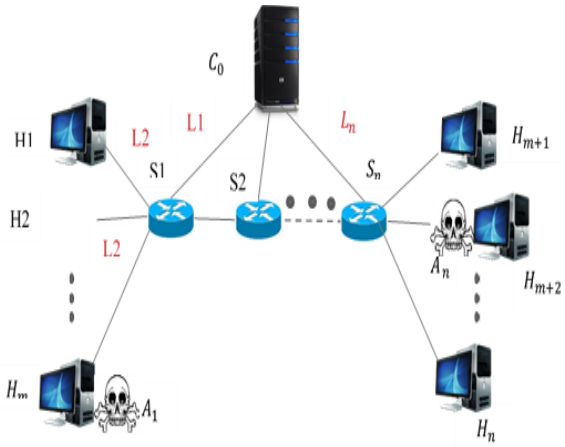
شبکه‌های نرم افزار محور یک فن‌آوری جدیدی هستند که قبل پیاده سازی زیرساخت‌های این معماری بایستی تدابیر لازم جهت امنیت و مقابله با تهدیدات و حملات در آن اندیشیده شود. همان‌طورکه در شکل (۲) نشان داده شده است، یک شبکه نرم-افزار محور متشکل از مجموعه‌ای از سوئیچ‌های نرم‌افزاری $S=\{s_1, s_2, \dots, s_n\}$ مجموعه‌ای از میزبان‌های شبکه $H=\{H_1, H_2, \dots, H_n\}$ مجموعه‌ای از لینک مجازی $L=\{L_1, L_2, \dots, L_n\}$ و کنترل کننده متمرکز C_0 می‌باشد. در این معماری به‌دلیل کنترل متمرکز، حملات منع خدمت تهدیدی جدی محسوب می‌شوند، زیرا بر روی کارآیی شبکه، افزایش تأخیر و دور ریختن بسته‌های مجاز تأثیر می‌گذارند. اگرچه این حملات در شبکه‌ها از مبدأ اینترنت شناخته شده بود، اما تعداد این حملات در شبکه‌های امروزی بسیار زیادتر شده است و بدون شک به عنوان یک تهدید برای شبکه‌های آینده نیز خواهد بود.

در حمله به یک شبکه نرم افزار محور، قدم اول شناسایی SDN بودن شبکه است. اغلب شبکه‌های سنتی، جداول انتقال پیش تنظیم شده‌ای دارند، بنابراین، در این شبکه‌ها نیازی به زمان اضافی برای پردازش و ایجاد یک جریان برای بسته‌های ورودی جدید نیست. ولی در شبکه‌های نرم افزار محور، کنترل کننده بایستی یک زمان کوتاهی را برای جریان‌های ورودی بسته‌های جدید اختصاص دهد، بنابراین زمان بیشتری را برای پردازش اولین بسته در مقایسه با سایر بسته‌ها صرف می‌کند. براساس این دانش، مهاجم A با تسخیر تعدادی میزبان H و بررسی زمان پاسخ به اولین بسته نسبت به سایر بسته‌ها اقدام به شناسایی نوع شبکه می‌کند. اگر زمان پاسخ به اولین بسته بیشتر از زمان پاسخ سایر بسته‌ها باشد؛ مهاجم نرم افزار محور بودن شبکه را تشخیص می‌دهد. در مرحله بعد، مهاجمA با استفاده از میزبان‌های H تسخیرشده ($H_m, H_{m+2} \dots$،) حجم انبوهی از بسته‌های جعلی را به سمت کنترل کننده C_0 ارسال می‌کند.

نتیجه حمله مهاجم A موجب اشغال منابع محدود کنترل کننده C_0 شده و پردازش بسته‌ها به واسطه غیرقابل دسترس - شدن کنترل کننده مختل می‌گردد. در نهایت، مجموعه سوئیچ-های S قادر به ارتباط با کنترل کننده C_0 نخواهند بود. در این شبکه‌ها اگر ارتباط بین سوئیچ‌ها و کنترل کننده از بین برود؛ شبکه توان پردازشی خود را از دست خواهد داد. به بیان دیگر، اگر پردازش بسته‌ها در کنترل کننده به واسطه غیرقابل دسترس شدن کنترل کننده C_0 مختل شود؛ کل معماری شبکه نرم افزار محور از کار خواهد افتاد، بنابراین، حملات منع خدمت توزیع - شده در این معماری بسیار مخرب‌تر خواهد بود [۱۸]. در نتیجه، حفاظت از کنترل کننده در معماری‌های شبکه‌های نرم افزار محور امری ضروری است. اگر تشخیص در مراحل اولیه حملات و قبل از آسیب رسیدن به کنترل کننده انجام گیرد؛ می‌توان اقدامات مقابله‌ای را در شبکه‌های نرم افزار محور اعمال کرد.

اگرچه تاکنون پیشنهادات زیادی برای تشخیص یا مقابله با این حملات در شبکه‌های سنتی وجود دارد، اما طبق بررسی‌های ما در زمینه تشخیص و مقابله با حملات منع خدمت توزیع شده در شبکه‌های نرم افزار محور مطالعات کافی انجام نگرفته و روش‌های کاربردی لازم در این حوزه ارائه نشده است. در شبکه-های نرم افزار محور به‌دلیل محدودیت‌های منابع کنترل کننده، روش‌های سنتی مقابله‌ای از جمله دیوار آتش و سیستم‌های تشخیص نفوذ در این شبکه‌ها ایده مناسبی نمی‌باشد. همچنین، خصوصیات انعطاف پذیری بالا و مدیریت آسان شبکه‌های نرم افزار محور، ابزار قدرتمندی را برای تشخیص و مقابله با حملات منع خدمت توزیع شده فراهم آورده است. بنابراین، استفاده از روش‌های مقابله‌ای سنتی موجب عدم بهره‌گیری درست از مزایای این معماری می‌گردد. مسأله ما در این‌جا ارائه روشی مؤثر به منظور تشخیص و مقابله با حملات منع خدمت در شبکه‌های نرم افزار محور است که ضمن سادگی کارآیی بهتری در مقایسه با

روش‌های قبلی داشته باشند. با توجه به این که شبکه‌های نرم افزار محور هنوز به بلوغ لازم نرسیده‌اند، ارائه روش‌های کارا در این زمینه و تجمیع این روش‌ها با معماری کنونی این نوع شبکه‌ها می‌تواند در حفظ امنیت آن‌ها مؤثر باشد. از طرف دیگر، زمینه را برای اعتماد بیشتر به این شبکه‌ها و به کارگیری فراگیرتر آن‌ها فراهم کند.



شکل (۲): حملات منع خدمت توزیع‌شده در شبکه نرم افزارمحور

۴- راه حل پیشنهادی

در روش‌های ارائه شده در زمینه تشخیص و مقابله با حملات منع خدمت توزیع شده در شبکه‌های نرم افزار محور، مشکلاتی از جمله نیاز به تغییرات در ساختار سوئیچ‌ها، نیاز نصب تجهیزات، استفاده از منابع محدود کنترل کننده، ایجاد سربار اضافی در کنترل کننده و عدم توانایی تشخیص حملاتی با توزیع شدگی یکنواخت مطرح است. ما در این مقاله با بهره‌گیری از امکانات معماری شبکه نرم افزار محور، سازوکار مقابله‌ای مختص این معماری را ارائه می‌دهیم. کنترل متمرکز شبکه‌های نرم افزار محور، امکانات مناسبی را برای ارزیابی میزان بسته‌های ورودی جدید به شبکه فراهم می‌کند. ما از این فرصت فراهم شده توسط کنترل کننده، برای جمع‌آوری آمار بسته‌های ورودی جدید استفاده می‌کنیم. در الگوریتم پیشنهادی با استفاده از روش فاصله هلینگر، حمله منع خدمت توزیع شده قبل از آسیب رسیدن به کنترل کننده تشخیص داده می‌شود.

پس از تشخیص حمله بایستی اقدامات لازم جهت کاهش اثر حملات در شبکه اعمال گردد. کاهش اثر حملات منع خدمت توزیع شده، مجموعه‌ای از روش‌ها برای حفاظت از شبکه مورد بررسی در برابر حملات منع خدمت توزیع شده است. یکی از عملکردهای اختصاصی کنترل کننده در شبکه‌های نرم افزار محور، جمع‌آوری آمار از همه سوئیچ‌های مبتنی بر پروتکل Open Flow برای تشخیص جریان‌های غیرفعال است. مقدار idle-time outموجود در ماژول کنترل کننده، بیانگر مدت زمانی است که اگر طی این مدت، بسته یا ترافیکی که با جریان مربوطه مطابقت داشته باشد، وارد نشود؛ جریان مذکور حذف می‌گردد. مقدار hard-time out نیز به کل مدت زمانی که یک جریان (حتی در صورت تداوم ارسال ترافیک از جریان مذکور) می‌تواند در جدول بماند اطلاق می‌شود. پس از طی این مدت، جریان مذکور از جداول حذف می‌شود. مقادیر پیش‌فرض این ماژول‌ها در کنترل‌کننده Pox، ۶۰s می‌باشد. در این الگوریتم پیشنهادی، با تنظیم مقادیر hard- time out و idle- time out می‌توان جریان‌های بسته‌های جعلی را کنترل نمود. در این سازوکار پس از تشخیص حمله، با تغییر مقادیر time out بسته‌های حمله از جداول حذف شده و IP آدرس مبدأ مربوط به بسته‌های جعلی مسدود می‌شوند. به این ترتیب ما می‌توانیم تأثیرات حملات منع خدمت توزیع شده در شبکه نرم افزار محور را کاهش دهیم. این الگوریتم پیشنهادی دارای مزایای زیر می‌باشد:

- شفافیت: جمع‌آوری آماری با سایر عملکردهای کنترل کننده تداخل ندارد.
- انعطاف‌پذیری: این روش با تغییرات پیکربندی سیستم سازگار است.
- توانایی تشخیص حملات با توزیع شدگی یکنواخت بین میزبان‌ها
- محاسبات کم و عدم ایجاد سربار در کنترل کننده
- بدون نیاز به تغییرات در ساختار دستگاه‌های شبکه
- بدون نیاز به نصب تجهیزات اضافی

۴-۱- روش فاصله هلینگر

در نظریه اطلاعات و آمار، فاصله آماری برای تعیین کمیت فاصله بین دو موجودیت آماری به کار گرفته می‌شود که این موجودیت آماری ممکن است متغیر احتمالی یا توزیع احتمال یک نمونه باشد. در واقع، اندازه‌گیری فاصله آماری، تفاوت بین متغیرهای تصادفی را بیان می‌کند [۲۶]. در آمار و احتمالات از فاصله هلینگر برای تعیین کمیت شباهت بین دو مجموعه توزیع احتمال استفاده می‌شود. روش فاصله هلینگر در سال ۱۹۰۹ توسط ارنست هلینگر ارائه شد [۲۷]. اگر دو مجموعه توزیع احتمال P و Q در فضای متناهی، دارای احتمالات p_i و q_i مطابق رابطه (۱)

باشند [۲۸]:

$$P = \{p_1, p_2, \dots, p_n\} \quad و \quad Q = \{q_1, q_2, \dots, q_n\} \qquad (۱)$$

با توجه به مثبت بودن مقادیر احتمالات روابط زیر را داریم [۲۸]:

$$p_i \gg 0\,,\ q_i \gg 0 \quad \sum_i^n q_i = 1,\ \sum_i^n p_i = 1 \qquad (۲)$$

فاصله هلینگر طبق رابطه (۳)، به صورت زیر محاسبه می‌شود [۲۷]:

$$HD^2_{(p,q)} = 1/2 \sum_{i=1}^{n} (\sqrt{p_i} - \sqrt{q_i})^2 \qquad (۳)$$

اندازه فاصله هلینگر همیشه بین مقادیر صفر و یک است. اگر توزیع احتمالات بین دو مجموعه یکسان باشند؛ مقدار فاصله هلینگر صفر می‌گردد. زمانی‌که دو توزیع احتمال کاملاً متفاوت باشند؛ مقدار فاصله هلینگر برابر یک خواهد بود. به بیان دیگر، اگر انحراف قابل توجهی در طول زمان روند بررسی، صورت نگرفته باشد؛ مقادیر فاصله هلینگر، اعداد نزدیک صفر محاسبه می‌شوند. زمانی‌که مقدار فاصله هلینگر افزایش یابد؛ بیانگر رخداد ناهنجاری و تغییر توزیع احتمالات می‌باشد. این ویژگی، فاصله هلینگر را روشی مناسب برای تعیین کمیت شباهت بین دو مجموعه داده در وضعیت عادی و وضعیت حمله تبدیل کرده است [۲۹].

ما در این مقاله قصد داریم با به‌کارگیری امکانات شبکه نرم افزار محور و با استفاده از روش فاصله هلینگر، یک سازوکار تشخیص و کاهش اثر حملات منع خدمت توزیع شده در شبکه‌های نرم‌افزار محور ارائه دهیم. هر رفتار عادی ممکن است دچار تغییر شده و نوسانات فاصله هلینگر را در طول زمان به‌وجود آورد. بنابراین حد آستانه ثابت برای سازوکار تشخیصی راه حل مناسب و عملی نمی‌باشد. ما در الگوریتم پیشنهادی از روش تطبیق متحرک میانگین وزنی (EWMA)[1] برای ایجاد حد آستانه تطبیقی [۳۰] در شبکه نرم افزار محور استفاده می‌کنیم. برای ایجاد حد آستانه تطبیقی نیاز به یک شاخص واقعی ناهنجاری داریم که حد آستانه تطبیقی برای روش فاصله هلینگر طبق روابط زیر قابل محاسبه می‌باشد [۲۹]:

$$H_{n+1} = (1-\alpha).H_n + \alpha.HD_n \qquad (۴)$$

$$\sigma_n = |H_n - HD_n| \qquad (۵)$$

$$S_{n+1} = (1-\beta).S_n + \beta.\sigma_n \qquad (۶)$$

$$H^{thre}_{n+1} = \lambda.H_{n+1} + \mu.S_{n+1} \qquad (۷)$$

1- Exponential Weighted Moving Average

اساس ایده حدآستانه تطبیقی، پیش‌بینی مقادیر بعدی براساس مقادیر فعلی است. در رابطه‌های بالا HD_n مقدار فعلی فاصله هلینگر و H_n و H_{n+1} به ترتیب برآورد میانگین فاصله‌های هلینگر فعلی و بعدی می‌باشند. σ_n میزان انحراف H_{n+1} از HD_n است. S_n و S_{n+1} نیز بیانگر میانگین انحراف فاصله هلینگر فعلی از فاصله هلینگر بعدی هستند. با استفاده از مقادیر H_{n+1} و S_{n+1} مقدار حد آستانه تطبیقی براساس رابطه (۷) محاسبه می‌شود. به منظور ایجاد حاشیه امن در حد آستانه ضرایب λ و μ در این رابطه اعمال شده است. در روابط بالا، همه پارامترهای λ ، μ، α و β قابل تغییر هستند. ما در این پژوهش مقدار پارامترهای λ، μ، αو β را به ترتیب با اعداد ۰/۵، ۱، ۰/۱۲۵، ۰/۲۵ مقداردهی می‌کنیم.

۴-۲- شبه کد الگوریتم پیشنهادی

در الگوریتم پیشنهادی، برای تشخیص حملات منع خدمت توزیع شده، احتمالات تکرار بسته‌هایی با IP آدرس مقصد یکسان هر بازه زمانی، در کنترل کننده مورد بررسی قرار می‌گیرد. ما در این مقاله برای دستیابی به دقت تشخیص بالا و هزینه محاسباتی پایین، طول بازه زمانی را 5s درنظر می‌گیریم. ابتدا جدولی را در کنترل کننده برای ثبت و ذخیره اطلاعات مربوط به IP آدرس مبدأ و مقصد بسته‌های انتقالی و زمان ورود بسته از سمت سوئیچ به کنترل کننده تعریف می‌کنیم. همچنین، در این جدول یک شمارنده M برای شمارش تعداد تکرار IP آدرس مقصد هر بسته انتقالی و یک شمارنده L برای شمارش تعداد بسته‌ها در هر بازه زمانی تعریف می‌شود. در این سازوکار، بسته‌های ورودی به کنترل کننده در هر بازه زمانی ۵ ثانیه‌ای برطبق IP آدرس مقصد، مورد تجزیه و تحلیل قرار می‌گیرند.

در شکل (۳) شبه کد الگوریتم پیشنهادی آمده است. همان‌طوری که در این شبه کد مشخص است، مقادیر اولیه ماژول‌های hard-time out و idle-time out را 60s درنظر می‌گیریم. مقادیر متغیرهای زمانی (t_0 ، c) و شمارنده‌ها (L، M) با عدد صفر مقداردهی اولیه می‌شوند. سپس IP آدرس مقصد اولین بسته ورودی مربوط به بازه زمانی 5s بررسی می‌شود. اگر مشخصات این بسته در جدول موجود بود، به تعداد شمارنده M مربوط به تعداد تکرار بسته‌هایی با IP آدرس مقصد بسته مورد نظر، یک واحد اضافه می‌گردد (خط ۹-۸). در غیر این‌صورت در جدول یک جریان جدیدی با مشخصات بسته مورد بررسی، ایجاد می‌شود (خط۱۱). در مرحله بعدی زمان ورود بسته در متغیر زمانی c ذخیره می‌شود و همچنین به تعداد شمارنده L که مربوط به تعداد بسته‌های هر بازه زمانی می‌باشد، یک واحد اضافه می‌گردد (خطوط ۱۴-۱۲).

در گام بعدی اندازه بازه زمانی برای تشخیص اتمام عملیات مربوط به یک بازه زمانی ۵ ثانیه‌ای بررسی می‌شود (خط ۱۵). اگر اندازه بازه زمانی کمتر از 5s بود (خطوط ۳۶-۳۵)، بسته ورودی بعدی وارد سیستم شده و مورد بررسی قرار می‌گیرد (خط ۷). در صورت اتمام بازه زمانی عملیاتی، با توجه به مقادیر شمارنده‌های M و L توزیع احتمالات هر بازه زمانی طبق رابطه (۸) محاسبه می‌شود. به این ترتیب، مجموعه توزیع احتمالات مجموعه q به‌دست می‌آید (خط ۱۷). از آن‌جایی‌که برای محاسبه فاصله هلینگر نیاز به دو مجموعه توزیع احتمال می‌باشد، برای مدت زمان معینی ترافیک‌های وضعیت عادی شبکه را بررسی کرده و طبق مراحل فوق، مجموعه توزیع احتمالات وضعیت عادی را به‌دست می‌آوریم. این مجموعه به عنوان مجموعه p در الگوریتم پیشنهادی بیان شده است. سپس فاصله هلینگر دو مجموعه توزیع احتمال و حد آستانه تطبیقی مربوطه طبق روابط (۳ و ۷) محاسبه می‌گردد (خطوط ۱۹-۱۸).

$$q = \mathrm{M}/L \qquad (۸)$$

در گام بعدی، مقادیر فاصله هلینگر و حد آستانه مقایسه می‌شوند (خط ۲۰). اگر مقدار فاصله هلینگر کمتر از حد آستانه باشد، بیانگر وضعیت عادی در بازه زمانی مورد نظر است. در این صورت، مقدار متغیر زمانی c در متغیر t_0 قرار گرفته و مقدار شمارنده M و L نیز برای بررسی بازه زمانی بعدی صفر می‌گردد. سپس مراحل ذکر شده برای بازه زمانی 5s بعدی ادامه می‌یابد (خطوط ۳۳-۲۸). اگر مقدار فاصله هلینگر از مقدار حد آستانه بیشتر باشد؛ بیانگر وقوع حمله در شبکه است. در این صورت، IP آدرس‌های مبدأ مربوط به جریان بسته ورودی با IP آدرس مقصد مورد نظر، مسدود شده و مقادیر hard-time out و idle-time out برای حذف جریان‌های حمله از جداول سوئیچ‌ها با مقادیر صفر تنظیم می‌شوند. سپس مقدار متغیر c در متغیر t_0 قرار گرفته و مقادیر شمارنده‌های M و L نیز برای ادامه روند بررسی بازه‌های زمانی بعدی صفر می‌گردند (خطوط ۲۷-۲۱).

۵- ارزیابی و مقایسه

ما در این مقاله برای ارزیابی الگوریتم پیشنهادی، شبکه نرم افزار محور و حملات منع خدمت توزیع شده را با استفاده از محیط مینی‌نت[۳۱] در بستر سیستم عامل لینوکس شبیه سازی کرده‌ایم. شبیه ساز مینی‌نت در واقع یک مقلد شبکه است که با استفاده از آن می‌توان یک هم‌بندی متشکل از تعدادی میزبان مجازی، لینک مجازی و سوئیچ مجازی شبکه‌های نرم افزار محور را اجرا نمود. همچنین، می‌توان سوئیچ‌های این شبکه را به یک کنترل کننده خارجی متصل کرد. این شبیه ساز برنامه‌ای است که می‌تواند شبکه مجازی را ایجاد کرده و هسته‌های واقعی، کدهای برنامه و سوئیچ‌های شبکه را بر روی یک ماشین (ماشین مجازی، ابر یا سیستم واقعی) به اجرا درآورد [۳۲]. از آن‌جایی‌که در این شبیه سازی، هم‌بندی مورد بررسی در مقیاس کوچک تست خواهد شد، ما یک کنترل کننده کم حجم و سبک را نیاز داریم. کنترل کننده POX را برای انجام این پروژه انتخاب نمودیم و این کنترل کننده را به همراه برخی نرم افزارهای مورد نیاز از جمله اسکاپی و وایرشارک به همراه محیط مینی‌نت نصب کرده‌ایم. در این‌جا نرم افزار اسکاپی برای تولید ترافیک و وایرشارک برای پایش وضعیت ترافیکی مورد استفاده قرار گرفته‌اند.

```
1) {
2) c=0;  t_0 = 0; M=0; L=0;
3) L_1:
4) Hard − time out = 60; Idle – time out = 60;
5) L_2:
6)  t_1 = t_0;
7) L_3: Input (packet − in)_i
8)    If ip address exists then
9)     M++;
10)   Else
11)      Add to hash table;
12)  c = t_i;
13)  t_2 = c;
14)  L++;
15)  If t_2 − t_1 ≥ 5  then
16)   {
17)      q_i = M / L;
18)      HD^2_(p,q) = 1/2 Σ_{i=1}^{n} (√p_i − √q_i)^2;
19)      H^thre_{n+1} = λ.H_{n+1} + μ.S_{n+1};
20)     If HD_n >  H^thre_{n+1}  then
21)       {
22)     Detect attack; Blocked IP address attacks;
23)        Hard- time out= 0;
24)        Idle- time out = 0;
25)        t_0=c; L=0; M=0;
26)        Go to L_1;
27)        }
28)      Else
29)        {
30)        No attack;
31)        t_0=c; L=0; M=0;
32)        Go to  L_2;
33)         }
34)  }
35)  Else
36) Go to  L_3;
37) }
```

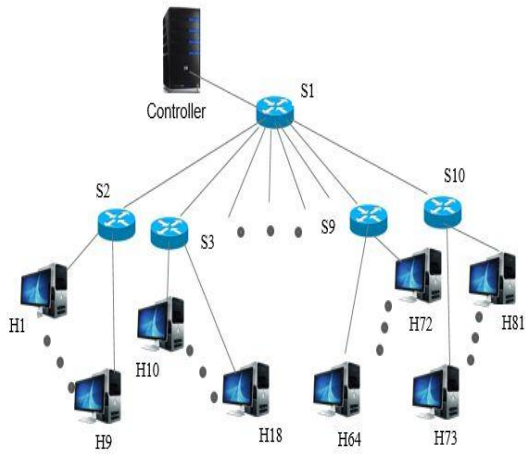
شکل (۳): شبه کد الگوریتم پیشنهادی

هدف ما از این شبیه سازی ارزیابی سازوکار ارائه شده می‌باشد. برای این منظور، ابتدا حملات منع خدمت توزیع شده در شبکه نرم‌افزار محور را شبیه‌سازی می‌کنیم. سپس ترافیک‌های عادی و ترافیک‌های حمله را بررسی کرده و با

محاسبه مقادیر فاصله هلینگر در هر دو وضعیت، الگوریتم پیشنهادی را مورد تجزیه و تحلیل قرار می‌دهد. در نهایت، این سازوکار را با سایر روش‌های ارائه شده مورد مقایسه قرار خواهیم داد.

۵-۱- راه‌اندازی شبکه نرم افزار محور در محیط مقلد مینی نت

برای راه‌اندازی شبکه نرم افزار محور و تولید ترافیک‌های حمله منع خدمت توزیع شده در محیط مینی‌نت، همبندی درختی متشکل از ۸۱ میزبان و ۱۰ سوئیچ را به همراه لینک‌های مجازی ایجاد می‌کنیم. سوئیچ‌های این شبکه به نحوی تنظیم می‌شوند که با یک کنترل کننده در حال اجرای خارجی با آدرس مشخص متصل باشند. در این همبندی برای ایجاد سوئیچ‌ها، از سوئیچ مجازی OVS[1] که یک سوئیچ نرم افزاری با قابلیت اجرا بر روی سخت افزار و نرم افزار می‌باشد، استفاده می‌کنیم. بعد از ایجاد همبندی، کنترل کننده POX را برای شناسایی آدرس‌های مک در لایه دو و انجام عمل سوئیچینگ، تنظیم و اجرا می‌کنیم. با اجرای کنترل کننده، سوئیچ‌ها به کنترل متصل می‌شوند. همبندی ایجاد شده را در شکل (۴) می‌بینید. همان‌طور که در این شکل نشان داده شده است، سیستم‌های میزبان به سوئیچ‌های مجازی براساس پروتکل Open Flow متصل شده‌اند و خود این سوئیچ‌ها نیز به یک سوئیچ سطح بالاتر متصل هستند. این مجموعه تشکیل دهنده بخش داده‌ای شبکه است. بخش داده‌ای شبکه از طریق سوئیچ‌ها به کنترل کننده POX که خارج از شبکه مجازی قرار دارد، متصل می‌شود و دستورات مربوطه را در صورت لزوم از آن دریافت می‌کند.



شکل (۴): همبندی شبکه نرم‌افزار محور ایجاد شده در محیط مینی‌نت

پس از ایجاد همبندی، با استفاده از برنامه اسکاپی دو نوع ترافیک حمله و ترافیک عادی را در شبکه تولید می‌کنیم. در تنظیمات برنامه اسکاپی برای تولید ترافیک حمله، از بسته‌هایی با IP آدرس مقصد تصادفی استفاده می‌کنیم. همچنین، برای توزیع شدگی حمله نیز چندین میزبان را به‌عنوان میزبان‌های تسخیر شده در نظر گرفته و برنامه اسکاپی را در پایانه خارجی میزبان‌های مربوطه اجرا می‌کنیم. برای تعیین این‌که یک بسته اطلاعاتی در اینترنت یا سایر شبکه‌ها به چه برنامه‌ای در میزبان مقصد تعلق بگیرد، از شماره درگاه استفاده می‌شود. در پروتکل‌های ارتباطی TCP این عدد در سرآیند بسته اطلاعاتی ارسالی قرار می‌گیرد و به میزبان مقصد ارسال می‌گردد. شماره درگاهی که برنامه کاربردی به وسیله آن، اطلاعات را ارسال می‌کند، به‌عنوان شماره درگاه مبدأ و شماره درگاهی که برنامه کاربردی سیستم میزبان به وسیله آن اطلاعات را دریافت می‌کند، به‌عنوان شماره درگاه مقصد نامیده می‌شوند. با توجه به این‌که در حالت پیش‌فرض Open Flow تنها سرآیند بسته‌ها به کنترل کننده ارسال می‌گردد، بسته‌های تولیدی در این شبیه سازی فاقد محتوی خواهند بود. مشخصات بسته‌های ترافیک تولیدی را در جدول (۱) می‌بینید.

جدول (۱): مشخصات بسته‌های ترافیک حمله

محتوی	شماره درگاه مقصد	شماره درگاه مبدأ	نام پروتکل
فاقد محتوی	۶۶۳۳	۶۸	TCP

۵-۲- ارزیابی الگوریتم پیشنهادی

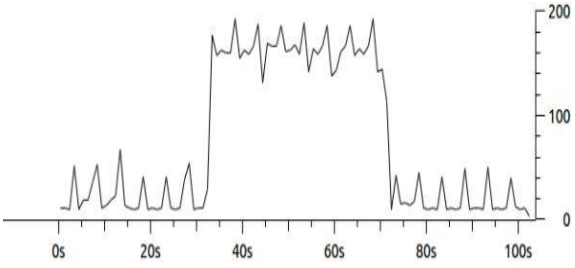
در حملات منع خدمت توزیع شده تولید بسته‌های حمله، سریع‌تر از تولید بسته‌های ترافیک عادی انجام می‌گیرد. ما برای پیاده سازی این ویژگی حملات، فاصله زمانی تولید بسته‌ها را در حالت عادی به ۰/۱s و فاصله زمانی تولید بسته‌های ترافیک حملات را به ۰/۱ s ، ۰/۰۱۵ s و ۰/۰۲۱s در برنامه اسکاپی تنظیم کردیم. طبق رابطه (۹) با اعمال این تنظیمات نرخ حملات ایجاد شده ۱۰٪، ۱۵٪ و ۲۱ ٪ محاسبه می‌شود. در این رابطه T_A و T_N به ترتیب فاصله زمانی ارسال بسته‌ها در شرایط حمله و شرایط عادی می‌باشد [۸].

$$R = \frac{T_A}{T_N} \times 100 \qquad (۹)$$

در شبکه‌های نرم افزار محور هر بسته جدیدی که وارد سیستم می‌شود، اگر از منظر سوییچ‌ها جدید به نظر برسد برای

1- Open virtual switch

تعیین تکلیف به سمت کنترل کننده ارسال می‌شود. در نتیجه، در کنترل کننده این شبکه، مجموعه‌ای از بسته‌هایی ارسالی از سمت سوییچ‌ها خواهیم داشت. ما برای بررسی حملات منع خدمت توزیع شده در شبکه نرم افزار محور، حملاتی با نرخ‌های ۱۰٪، ۱۵٪ و ۲۱٪ را به مدت ۴۳s ایجاد کردیم. نمودار (۱) گراف حاصل از شبیه سازی حمله با نرخ ۲۱ ٪ را نشان می‌دهد که محور افقی بیانگر زمان و محور عمودی بیانگر تعداد بسته‌های رد و بدل شده بین سوئیچ و کنترل کننده می‌باشند. همان‌طور که در این نمودار قابل مشاهده است، حمله از زمان ۳۲s تا ۷۵s صورت گرفته و نرخ بسته‌های ورودی از ۴۰ بسته در ثانیه به ۲۰۰ بسته در ثانیه افزایش یافته است.



نمودار (۱): گراف حاصل از شبیه سازی حمله با نرخ ۲۱ درصد

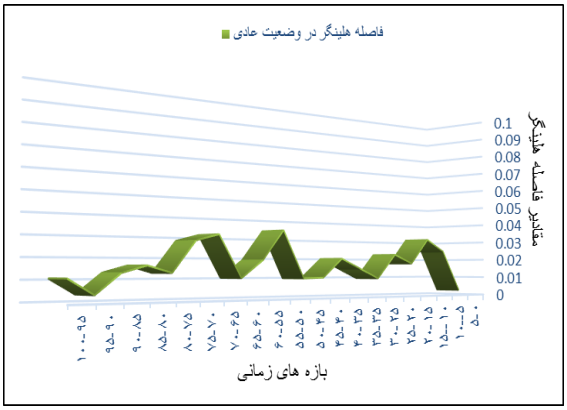
۵-۲-۱- تجزیه و تحلیل اثرات حمله

برای ارزیابی الگوریتم پیشنهادی، ابتدا به مدت زمان معینی جریان عبوری وضعیت عادی شبکه را نمونه‌گیری می‌کنیم. این دوره زمانی را فاصله زمانی آموزشی می‌نامیم. سپس به نمونه‌گیری وضعیت حمله می‌پردازیم. این دوره زمانی را نیز فاصله زمانی تست، نام‌گذاری می‌کنیم که طول این دوره، با طول دوره فاصله زمانی آموزش برابر خواهد بود. در این مقاله، IP آدرس مقصد را به عنوان متغیر احتمالی درنظر می‌گیریم. با استفاده از آمارهای به‌دست آمده در بازه‌های زمانی مشخص شده، مجموعه توزیع احتمال در فاصله زمانی آموزشی (p_i ها) و توزیع احتمال در فاصله زمانی تست (q_i ها) از طریق رابطه (۱۰) محاسبه می‌شوند:

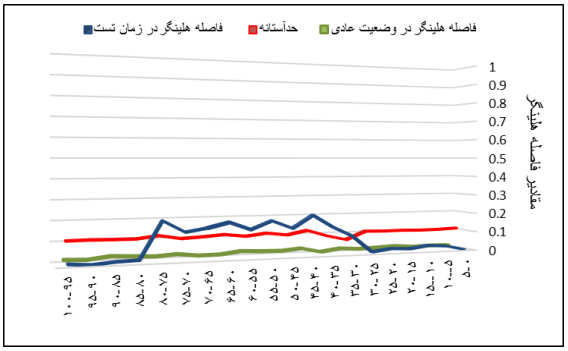
$$P_i = x_i/n \qquad (۱۰)$$

در این رابطه، X_i تعداد تکرار بسته‌ها با IP آدرس مقصد یکسان می‌باشد و n تعداد بسته‌های موجود در بازه زمانی مشخص شده است. بدین ترتیب، دو مجموعه توزیع احتمال شامل توزیع احتمالات دوره زمانی آموزشی و توزیع احتمالات دوره زمانی تست را در بازه‌های زمانی ۵ ثانیه‌ای به‌دست می‌آوریم. طبق روابط (۳ و ۷) فاصله هلینگر و حد آستانه تطبیقی بین اعضای متناظر دو مجموعه محاسبه می‌گردد.

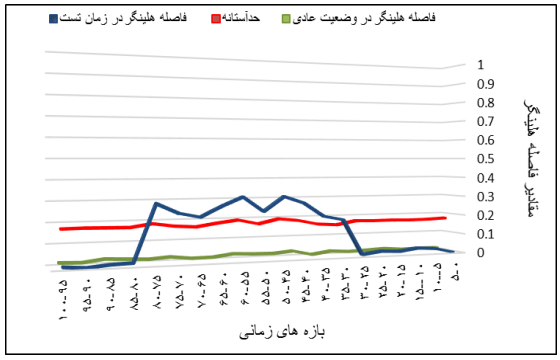
نمودار (۲) تغییرات مقادیر فاصله هلینگر را در وضعیت عادی شبکه نشان می‌دهد. در این نمودار، محور افقی بیانگر بازه‌های زمانی مورد بررسی و محور عمودی بیانگر مقادیر فاصله‌های هلینگر محاسبه شده می‌باشد. هنگامی‌که شبکه در وضعیت عادی قرار دارد، میزان تغییرات فاصله هلینگر بسیار ناچیز و در حدود ۰/۰۲۸۹ می‌باشد. در وضعیت عادی مقدار فاصله هلینگر کمتر از مقدار حد آستانه مربوط به بازه زمانی مورد نظر است. مطابق نمودارهای (۵-۳)، در بازه‌های زمانی ۳۲s تا ۷۵s به‌دلیل وقوع حمله، توزیع احتمالات دچار نوسان شده و مقادیر فاصله هلینگر متناسب با میزان بسته‌های حمله دچار تغییر می‌شوند. میزان تغییرات مقادیر فاصله هلینگر در وضعیت حملاتی با نرخ‌های ۱۰٪، ۱۵٪ و ۲۱٪ به ترتیب در حدود ۰/۲۰۶۷، ۰/۳۱۱۷ و ۰/۴۵۳۹ می‌باشد که این میزان تغییرات در مقایسه با وضعیت عادی افزایش چشم‌گیری را نشان می‌دهند.



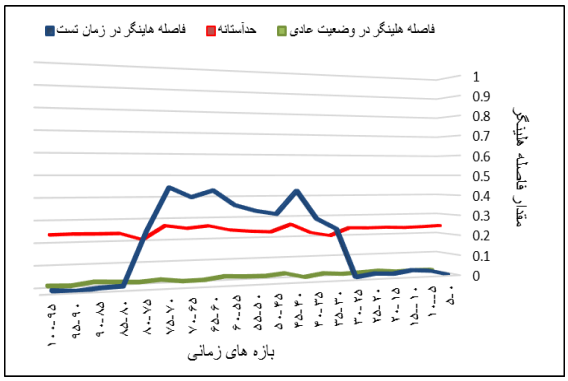
نمودار (۲): میزان تغییرات فاصله هلینگر در وضعیت عادی شبکه



نمودار (۳): میزان تغییرات فاصله هلینگر در حمله با نرخ ۱۰ ٪



نمودار (۴): میزان تغییرات فاصله هلینگر در حمله با نرخ ۱۵ ٪



نمودار (۵): میزان تغییرات فاصله هلینگر در حمله با نرخ ۲۱ ٪

همان‌طور که در این نمودارها قابل مشاهده است، در زمان وقوع حمله، مقدار فاصله هلینگر از حد آستانه مربوطه فراتر می‌رود.

نتایج حاصل از بررسی مقادیر فاصله هلینگر در وضعیت عادی و وضعیت حملاتی با نرخ‌های مختلف در جدول (۲) نشان داده شده است. در این جدول، همچنین میزان تغییرات حد آستانه نیز با توجه به متغیر بودن آن آمده است. طبق نتایج به‌دست آمده، اختلاف بین مقادیر کمترین فاصله هلینگر در ترافیک حملاتی با نرخ‌های ۱۰٪، ۱۵٪ و ۲۱٪ و بیشترین فاصله هلینگر در ترافیک وضعیت عادی به ترتیب ۰/۰۷۸۸، ۰/۱۴۷۸ و ۰/۲۱۲۸ محاسبه می‌شود. این مقادیر، افزایش ۷۱/۶۳، ۸۲/۵۶ و ۸۷/۲۱ درصدی فاصله هلینگر در ترافیک حمله را نشان می‌دهند. بنابراین، افزایش فاصله هلینگر وجود حمله را بسیار برجسته می‌کند. طبق این جدول، اختلاف بین بیشترین مقدار فاصله هلینگر موجود در حد آستانه و کمترین مقدار فاصله هلینگر در حملاتی با نرخ‌های ۱۰٪، ۱۵٪ و ۲۱٪ به ترتیب برابر ۰/۰۱۵۵، ۰/۰۱۸۴ و ۰/۰۲۸۱ است که این مقادیر را حاشیه امن حالت حمله می‌نامیم. میزان اختلاف بین کمترین مقدار فاصله هلینگر موجود در حد آستانه و بیشترین فاصله هلینگر در وضعیت عادی نیز به ترتیب برابر با ۰/۰۴۶۸، ۰/۰۱۲۷۸ و ۰/۱۷۶۴ می‌باشد که این مقادیر را حاشیه امن حالت عادی می‌نامیم. به بیان دیگر، در این حملات حاشیه امن حالت عادی به ترتیب ۳/۰۱، ۶/۹۴ و ۶/۲۷ برابر حاشیه امن حالت حمله است. طبق این نتایج، این الگوریتم پیشنهادی به‌دلیل دارا بودن حاشیه امن متناسب، قادر است حملات منع خدمت توزیع شده با هر نرخی را به درستی تشخیص داده و اقدامات مقابله‌ای را در شبکه‌های نرم افزار محور اعمال نماید.

جدول (۲): نتایج محاسبات انجام شده برای محاسبه فاصله هلینگر

	حمله با نرخ ۲۱٪	حمله با نرخ ۱۵٪	حمله با نرخ ۱۰٪	وضعیت عادی
بیشترین مقدار فاصله هلینگر	۰/۴۵۶۲	۰/۳۱۴	۰/۲۰۹	۰/۰۳۱۲
کمترین مقدار فاصله هلینگر	۰/۲۴۴	۰/۱۷۹	۰/۱۱۰	۰/۰۰۲۳
بیشترین حد آستانه	۰/۲۷۲۱	۰/۱۹۷۴	۰/۱۲۵۵	
کمترین حد آستانه	۰/۲۰۷۶	۰/۱۵۹	۰/۰۷۸	

۵-۳- مقایسه الگوریتم پیشنهادی با سازوکارهای قبلی

در این بخش به مقایسه الگوریتم پیشنهادی با سایر سازوکارهای ارائه شده در شبکه نرم افزار محور می‌پردازیم. بر طبق بررسی‌های انجام شده، الگوریتم پیشنهادی ما برخلاف سازوکار ارائه شده در مقالات [۲۰،۱۹] نیاز به تغییرات ساختار سخت افزاری سوئیچ‌ها ندارد و با ایجاد تغییرات نرم افزاری جزیی در کنترل کننده قابل پیاده سازی است. در مقالات [۲۵،۲۱] معماری شبکه نرم افزار محور را همانند معماری سنتی درنظر گرفته شده و از امکانات منحصر به فرد این معماری بی‌بهره بودند. راه حل ارائه شده در این مقالات نیاز به نصب تجهیزات اضافی داشت تا بتوانند در کنار کنترل کننده اقدامات مقابله‌ای را انجام دهند. الگوریتم پیشنهادی ما، با بهره‌گیری از امکانات کنترل کننده و در درون کنترل کننده انجام می‌گیرد و نیازی به نصب تجهیزات در بخش‌های مختلف شبکه ندارد.

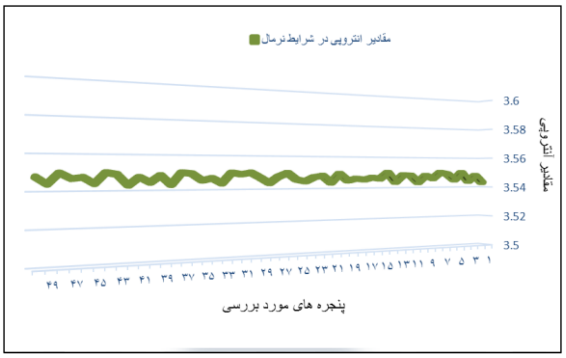
در ادامه به منظور مقایسه دقیق روش پیشنهادی با روش‌های گذشته، روش ارائه شده در [۸] را بر روی داده‌های حاصل از

شبیه سازی انجام شده، اعمال کرده و نتایج حاصل را مورد تجزیه و تحلیل قرار می‌دهیم. در [۸] برای تشخیص حمله، روش آنتروپی در هم‌بندی متشکل از ۶۴ میزبان و یک کنترل کننده بررسی شده است. در این روش، هر ۵۰ بسته ورودی به‌عنوان یک پنجره در نظر گرفته می‌شود. سپس تعداد بسته‌هایی با IP آدرس مقصد یکسان در هر پنجره مورد بررسی قرار می‌گیرد. در این پژوهش در ۵۰ پنجره متوالی مقادیر آنتروپی بررسی شد. نتایج حاصل از بررسی مقادیر آنتروپی در شرایط حمله و شرایط عادی را در جدول (۳) می‌بینید.

جدول (۳): نتایج حاصل از بررسی روش آنتروپی

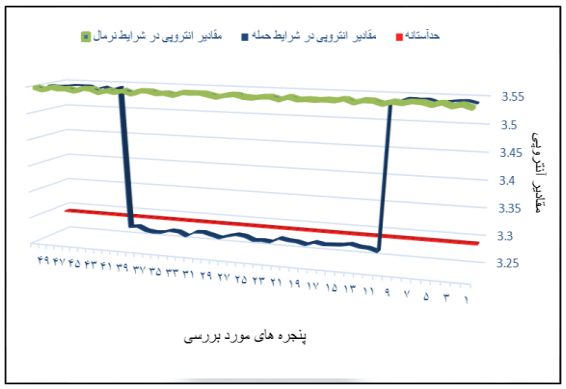
	بیشترین مقدار آنتروپی	کمترین مقدار آنتروپی
وضعیت عادی	۳/۵۵۰	۳/۵۴۳
وضعیت حمله	۳/۲۸۳	۳/۲۷۳

نمودار (۶)، میزان تغییرات مقادیر آنتروپی در شرایط عادی را نشان می‌دهد. در این نمودار، محور افقی بیانگر پنجره‌های مورد بررسی و محور عمودی بیانگر مقادیر آنتروپی است. در شرایط عادی شبکه، میزان تغییرات آنتروپی بسیار ناچیز و در حدود ۰/۰۰۷ است.



نمودار (۶): میزان تغییرات آنتروپی در وضعیت عادی شبکه

هنگامی‌که در شبکه حمله اتفاق می‌افتد، به‌دلیل افزایش تعداد بسته‌هایی با IP آدرس مقصد یکسان، مقادیر آنتروپی کاهش می‌یابد. در این بررسی میزان کاهش آنتروپی حدود ۰/۲۷۷ بوده است. طبق روش ارائه شده در این مقاله، مقدار حد آستانه برای این حمله ۳/۲۹ محاسبه شد. میزان افت آنتروپی در شرایط حمله و مقدار حد آستانه در نمودار (۷) نشان داده شده است.



نمودار (۷): میزان تغییرات آنتروپی در وضعیت حمله

نتایج حاصل از بررسی مشاهدات انجام شده در جدول (۴) آمده است. طبق این جدول، در روش آنتروپی حاشیه امن حالت حمله بسیار کمتر از حاشیه امن حالت عادی می‌باشد، بنابراین، ممکن است برخی حملات با تغییرات آنتروپی جزیی قابل تشخیص نباشند. درنتیجه، در این روش احتمال خطای تشخیصی افزایش می‌یابد. ولی در روش پیشنهادی ما، هر دو وضعیت عادی و حمله، حاشیه امن مناسب‌تری برخوردارند. در روش آنتروپی، به‌دلیل حد آستانه ثابت، توانایی تشخیص حمله‌ای با نرخ‌های مختلف را ندارد و در تشخیص حملاتی که ترافیک حمله به صورت یکنواخت بین میزبان‌ها توزیع شود، با مشکل مواجه خواهد شد. همچنین، این روش فاقد سازوکار کاهش اثر حمله می‌باشد. ولی روش پیشنهادی ما، دارای حد آستانه تطبیقی است و این روش قادر است حتی این‌گونه حملات را نیز شناسایی کرده و اقدامات لازم جهت کاهش اثر حملات را در شبکه نرم افزار محور اعمال نماید. نتایج مقایسه در جدول (۴) نشان داده شده است.

جدول (۴): نتایج حاصل از ارزیابی روش فاصله هلینگر و روش آنتروپی

	الگوریتم پیشنهادی	سازوکار مقاله [۸]
مقدار حاشیه امن حالت عادی	۰/۱۷۶۴	۰/۲۵۳
مقدار حاشیه امن حالت حمله	۰/۰۲۸۱	۰/۰۰۷
نسبت حاشیه‌های امن	۶/۲۷	۳۶
نوع حدآستانه	تطبیقی	ثابت

۶- نتیجه‌گیری

در این مقاله به بررسی شبکه‌های نرم افزار محور و راه حل‌های ارائه شده برای مقابله با حملات منع خدمت توزیع شده در این معماری پرداخته و با بهره‌گیری از فرصت‌های ایجاد شده توسط

کنترل کننده، راه حل مؤثر مبتنی بر روش‌های آماری را برای تشخیص و کاهش اثر حملات منع خدمت توزیع شده در شبکه‌های نرم افزار محور ارائه نمودیم. طبق بررسی‌های ما، تاکنون روش فاصله هلینگر در شبکه‌های نرم افزار محور به منظور تشخیص حملات منع خدمت استفاده نشده و این اولین راه حل در نوع خود برای شبکه‌های نرم افزار محور است. با اعمال این روش به‌عنوان روش تشخیصی، قادر به تشخیص حملات با نرخ‌های مختلف خواهیم بود. در الگوریتم پیشنهادی، پس از تشخیص حملات با بهره‌گیری از امکانات کنترل کننده اقدامات مقابله‌ای اعمال می‌گردد. این الگوریتم، برای جمع‌آوری آمار و محاسبات لازم، حافظه و منابع زیادی را مورد استفاده قرار نمی‌دهد و با اعمال تغییرات جزیی در کنترل کننده قابل پیاده سازی می‌باشد. به عبارت دیگر با این روش، کنترل کننده بدون نیاز به نصب تجهیزات، با وارد کردن دستورات و تغییرات جزیی توانایی تشخیص و کاهش اثر حملات را پیدا می‌کند.

در کارهای آتی می‌توان روش پیشنهادی ارائه شده در این مقاله را بهبود داده و در شبکه‌هایی با بیش از دو کنترل کننده مورد بررسی قرار داد. از آن‌جایی‌که روش پیشنهادی در این مقاله، راه حل تشخیص و کاهش اثر بعد از وقوع حمله است. بنابراین، می‌توان در بررسی‌های آینده بر روی چگونگی پیشگیری حملات منع خدمت توزیع شده در شبکه‌های نرم افزار محور متمرکز شد.

۷- منابع

[1] N.McKeown, T.Anderson, H.Balakrishnan, G.Parulkar, L.Peterson, J.Rexford and J.Turner, "OpenFlow: enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol.38, no.2, pp.69-74, 2008.

[2] ONF Market Education Committee, "Software-defined networking: The new norm for networks," ONF White Paper, 2012.

[3] S.Luo, J.Wu, J.Li and B.Pei, "A Defense Mechanism for Distributed Denial of Service Attack in Software-Defined Networks," In 2015 Ninth International Conference on Frontier of Computer Science and Technology, IEEE, pp. 325-329, August, 2015.

[4] D.Kreutz, F.M.Ramos, P.E.Verissimo, C.E.Rothenberg, S.Azodolmolky and S.Uhlig, "Software-defined networking: A comprehensive survey," Proceedings of the IEEE, vol.103, no.1, pp.14-76, 2015.

[5] M.Pham, D.B.Hoang, "SDN applications-The intent-based Northbound Interface realisation for extended applications," In NetSoft Conference and Workshops (NetSoft), 2016 IEEE, pp. 372-377, June, 2016.

[6] N.N.Dao, J.Park, M. Park and S.Cho, "A feasible method to combat against DDoS attack in SDN network," In 2015 International Conference on Information Networking (ICOIN), IEEE, pp. 309-311, January, 2015.

[7] H.T.N.Tri, K. Kim, "Assessing the impact of resource attack in Software Defined Network," In 2015 International Conference on Information Networking (ICOIN), IEEE, pp.420-425, January, 2015.

[8] S.M.Mousavi, M.St-Hilaire, "Early detection of DDoS attacks against SDN controllers," In Computing, Networking and Communications (ICNC), 2015 International Conference on IEEE, pp. 77-81, February, 2015.

[9] M.Hamedi, M.R.Shamani, M.J.Shamani, "Optimize the ant colony algorithm to track DoS attacks," Journal of Electronical & Cyber Defence, vol. 1, no. 4, pp.77-86, 2012. (in Persian)

[10] M. Fathian, M.Abdollahi Azgomi, H. Dehghani, "Modeling Browsing Behavior Analysis for Malicious Robot Detection in Distributed Denial of Service Attacks," Journal Of Electronical & Cyber Defence, vol. 4, no. 2, pp.1-13, 2016. (in Persian)

[11] M.Abassi, S.A.Hosseini, S.A.Vaezie, "Effective defense mechanism based a fair queue weighted against flood denial of service attacks in SIP networks," In 16th National Innovation Conference Computer Engineering and Information Technology, 2012. (in Persian)

[12] A.Salarvand, "Software defiend networks," In 1th symposium computer networks, Iran, Qom unit Sama Vocational School, 2012. (in Persian)

[13] E.Haleplidis, K.Pentikousis, S.Denazis, J.H.Salim, D.Meyer, O.Koufopavlou, "Software-defined networking (sdn): Layers and architecture terminology," No. RFC 7426, 2015.

[14] C.B.L.Contreras, D.Lopez, "Cooperating layered architecture for SDN," Draft-contrerassdnrg-layered-sdn-01 (work in progress), 2014.

[15] S.Shin, P.A.Porras, V.Yegneswaran, M.W.Fong, G.Gu and M.Tyson, "FRESCO: Modular Composable Security Services for Software-Defined Networks," In NDSS, February, 2013.

[16] P.Porras, S.Shin, V.Yegneswaran, M.Fong, M.Tyson, and G.Gu, "A security enforcement kernel for OpenFlow networks," In Proceedings of the first workshop on Hot topics in software defined networks, ACM, pp. 121-126, August, 2012.

[17] B.Wang, Y.Zheng, W.Lou and Y.T.Hou, "DDoS attack protection in the era of cloud computing and software-defined networking," Computer Networks, 81, pp.308-319, 2015.

[18] R. Mohammadifar, A. A. Rezaei, "Protection Against Flooding Attacks In Traditional Networks in Heterogeneous Partnership With Service Provider And Software Define Network (SDN) Controller," Journal of Electronical & Cyber Defence, vol. 4, no. 4, pp.63-78, 2017. (in Persian)

[19] S.Shin, V.Yegneswaran, P.Porras and G.Gu, "AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks," In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security ACM, pp. 413-424 , November, 2013.

[20] R.Kandoi, M.Antikainen, "Denial-of-service attacks in OpenFlow SDN networks," In 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, pp. 1322-1326, May, 2015.

[21] M.Suh, S.H.Park, B.Lee and S.Yang, "Building firewall over the software-defined network controller," In 16th International Conference on Advanced Communication Technology, IEEE, pp. 744-748, February, 2014.

[22] Y.L.Hu, W.B.Su, L.Y.Wu, Y.Huang and S.Y.Kuo, "Design of event-based intrusion detection system on OpenFlow network," In 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE, pp. 1-2, June, 2013.

[23] R.Braga, E.Mota, A.Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," In Local Computer Networks (LCN), 2010 IEEE 35th Conference on IEEE, pp. 408-415, October, 2010.

[24] P.Zhang, H.Wang, C.Hu and C.Lin, "On Denial of Service Attacks in Software Defined Networks," IEEE Network, vol.30, no.6, pp.28-33, 2016.

[25] M.Ramadas, S.Ostermann and B.Tjaden, "Detecting anomalous network traffic with self-organizing maps," In International Workshop on Recent Advances in Intrusion Detection, Springer Berlin Heidelberg, pp. 36-54, September, 2003.

[26] L.Le Cam, G.L.Yang, "Asymptotics in statistics: some basic concepts," Springer Science & Business Media, 2012.

[27] E.Hellinger, "Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen," Journal für die reine und angewandte Mathematik, vol.136, pp.210-271, 1909.

[28] A.R.Khajoeinezhad, H.R.Dalili and S.R.Chogan, "Study the impacts of INVITE flooding attack in VOIP and offering a new approach to detect attack," Electronics Industries Quarterly, vol. 6, no. 2, pp. 29-37, 2015. (in Persian)

[29] J.Tang, Y.Cheng and C.Zhou, "Sketch-based SIP flooding detection using Hellinger distance," In Global Telecommunications Conference, GLOBECOM 2009, IEEE, pp. 1-6, November, 2009.

[30] J.F.Kurose, K.W.Ross, "Computer networking: a top-down approach," Addison Wesley, 2007.

[31] B.Lantz, B.Heller and N.McKeown, "A network in a laptop: rapid prototyping for software-defined networks," In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, ACM, p.19, October, 2010.

[32] S.Oshima, T.Nakashima and T.Sueyoshi, "Early DoS/DDoS detection method using short-term statistics," In Complex, Intelligent and Software Intensive Systems (CISIS), International Conference on IEEE, pp. 168-173, February , 2010.

Journal of Electronical & Cyber Defence
Vol. 5, No. 4, 2017, Serial No. 20

A New Algorithm Based on Hellinger Distance for Mitigation of DDoS Attacks in Software Defined Networks

M. Ghasabi, M. Deypir*, E. Mahdipour

*Shahid Sattari Aeronautical University of Science and Technology

(Received: 18/01/2017 , Accepted: 23/07/2017)

ABSTRACT

Software defined network (SDN) was born to make changes to existing network architectures and devices with specific function to reach an intelligent network. Recently, this networks have gained popularity in enterprise networks because of the flexibility in network service management and reduced operational cost. In this architecture, operating system and applications from the network switch are decoupled. They centralized in a virtual layer that called the controller. In the SDN, due to the centralized decision-making and resources controller limitations are exposed to all kinds of threats such as Distributed Denial of Service (DDoS) attacks. In this paper we will review SDN architecture and DDOS attacks in SDN. We proposed a novel detection and mitigation algorithm that takes advantage of unique features of the SDN architecture. In the proposed algorithm, for detecting DDOS attacks in SDN, a statistical method based on Hellinger distance and Exponential Weighted Moving Average (EWMA) technique are used. In this paper, DDOS attacks in SDN is simulated by MiniNet emulator with Pox controller. Our experiments performed in the simulator, showed the efficiency of the proposed method and its superiority compared to previous approaches.

Keywords: Software Defined Networks, Distributed Denial of Service, Hellinger distance, Control plane, Data plane .

* Corresponding Author Email: mdeypir@ssau.ac.ir