مروری بر راه‌حل‌های دفاعی تشخیص نفوذ مبتنی بر شبکه نرم‌افزارمحور

مسعود محمدعلی‌پور[1]* و سعید شکرالهی[2]

[1] پژوهشکده فضای مجازی، دانشگاه شهید بهشتی، تهران، ایران
m.mohammadalipour@mail.sbu.ac.ir

[2] استادیار پژوهشکده فضای مجازی، دانشگاه شهید بهشتی، تهران، ایران
s_shokrollahi@sbu.ac.ir

چکیده

اغلب شبکه‌های فاقد زیرساخت ثابت مبتنی بر رایانش ابری با چالش‌های امنیتی مختلفی روبه‌رو هستند. در سال‌های اخیر، روش‌های متفاوتی از شبکه نرم‌افزار محور توزیع‌شده جهت مقابله با این چالش‌ها بهره برده‌اند. این فناوری ضمن داشتن قابلیت‌های فراوان، مقابل برخی تهدیدات و عوامل مخرب رایج از قبیل حمله منع سرویس توزیع‌شده با آسیب‌پذیری‌هایی روبه‌رو است. بررسی پژوهش‌های مختلف نشان می‌دهد که به‌منظور رفع آسیب‌پذیری‌ها، نیازمند تلفیق راه‌حل‌های دفاعی مناسب با ساختار شبکه نرم‌افزارمحور توزیع‌شده هستیم؛ بنابراین در این مقاله یک دسته‌بندی کلی از انواع راه‌حل‌های دفاعی در برابر حملات بالا ارائه کردیم. در ادامه ضمن طبقه‌بندی راه‌حل‌های تشخیص نفوذ به دو دسته آستانه‌ای و غیرآستانه‌ای، برخی مثال‌های کاربردی از راه‌حل‌های فوق را بررسی کردیم. به این نتیجه رسیدیم که آستانه‌ای‌بودن روش تشخیص نفوذ، میزان آسیب‌پذیری را تشدید می‌کند و ما ملزم به استفاده از راه‌حل‌های دفاعی غیرآستانه‌ای با معماری شبکه نرم‌افزار محور توزیع‌شده مسطح هستیم.

واژگان کلیدی: امنیت، تشخیص نفوذ، شبکه نرم‌افزار محور، حمله منع سرویس توزیع‌شده

۱- مقدمه

رایانش ابری[1] به‌عنوان یک رابط کاربری مناسب، دسترسی به منابع نرم‌افزاری را برای تعداد زیادی از کاربران از طریق سخت‌افزارهایی که در یک مرکز داده قرار دارند، فراهم می‌سازد. رایانش ابری به‌علت قابلیت‌های فوق‌العاده‌اش سبب جذب مشتریان و سرمایه‌گذاری هنگفتی شده و ویژگی‌های مهمی مانند اطمینان‌پذیری، کاهش هزینه، چندمستأجری، امنیت و غیره را به شبکه‌های مختلف تزریق می‌کند. هم‌اکنون مسائل امنیتی از مهم‌ترین چالش‌های شبکه‌های فاقد زیرساخت ثابت و مدیریت متمرکز مبتنی بر آن است [۱]. در برخی پژوهش‌ها [۲] جهت بهبود امنیت در شبکهٔ مبتنی بر رایانش ابری، استفاده از ویژگی‌های شبکهٔ نرم‌افزارمحور[2] پیشنهاد شده است. این فناوری قابلیت‌هایی مانند برنامه‌ریزی، کنترل متمرکز و تجزیه و تحلیل ترافیک را برای شبکه ارمغان آورده و می‌تواند به‌منظور بهبود امنیت استفاده شود. همچنین مجازی‌سازی، هوشمندی و دیدگاه‌های جدیدی را به شبکه می‌افزاید.

پژوهش‌گران [۳] ادعا می‌کنند که برخی شبکه‌های فاقد زیرساخت مانند شبکه موردی سیار می‌توانند جهت برطرف‌کردن مشکلاتی شامل فقدان زیرساخت ثابت و ضعف امنیتی از شبکه نرم‌افزار محور استفاده کنند. همچنین برخی دیگر از پژوهش‌گران [۲] معتقدند که قابلیت‌های این فناوری شامل تجزیه و تحلیل ترافیک، کنترل متمرکز و به‌روزرسانی پویای قوانین سبب تسهیل در تشخیص و مقابله با حملات می‌شوند؛ اما دو ویژگی کنترل نرم‌افزاری و کنترل متمرکز در شبکه نرم‌افزارمحور خود زمینهٔ حملاتی مانند حمله منع سرویس توزیع‌شده[3] را فراهم می‌کند.

با توجه به این‌که دسترسی به سرویس در بین الزامات رایانش ابری مهم است، در شبکه‌های فاقد زیرساخت مبتنی بر رایانش ابری به حملات منع سرویس[4] و منع سرویس توزیع‌شده توجه بیشتری شده است. در این مقاله ضمن بررسی تأثیر به‌کارگیری شبکه نرم‌افزارمحور بر روی تهدیدات منع سرویس توزیع‌شده، مروری بر انواع راه‌کارهای دفاعی حملات بالا داشته و آن‌ها را دسته‌بندی می‌کنیم.

[1] Cloud Computing
[2] Software Defined Network (SDN)
[3] Distributed Denial of Service (DDoS)
[4] Denial of Service (DoS)

* نویسندهٔ عهده‌دار مکاتبات

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

ادامه مقاله به‌صورت زیر سازماندهی شده است: در بخش ۲ کارهای مرتبط بررسی می‌شود. در بخش ۳ ضمن معرفی اجزا، ویژگی‌ها و انواع معماری کنترل‌کننده در شبکه نرم‌افزارمحور، ساختارهای مختلف کنترل‌کننده را مقایسه می‌کنیم. در بخش ۴ حملات منع سرویس توزیع‌شده را طبقه‌بندی کرده، تأثیر آن را در رایانش ابری بررسی کرده و انواع سناریوهای مختلف حمله را در شبکه نرم‌افزارمحور تشریح می‌کنیم. در بخش ۵ علاوه‌بر دسته‌بندی کلی راه‌حل‌های دفاعی، راه‌حل‌های تشخیص نفوذ مبتنی بر شبکه نرم‌افزار محور را طبقه‌بندی می‌کنیم. در بخش‌های ۸،۷،۶ و ۹ برخی مثال‌های کاربردی را در زمینه راه‌حل‌های دفاعی تشخیص نفوذ مبتنی بر شبکه نرم‌افزارمحور ارائه‌شده در پژوهش‌های مختلف را بررسی و در بخش ۱۰ مقاله را جمع‌بندی می‌کنیم.

۲- کارهای مرتبط

تنویر آلام[1] [۴] با ادغام شبکه موردی سیار و رایانش ابری یک مدل متحرک و جدید شبکه موردی سیار - ابر را طراحی کرد. نتایج نشان داد که میان‌افزار مورداستفاده در مدل متحرک بالا، برای ارتباط بین دستگاه‌های هوشمند بدون سیستم متمرکز مناسب است. پولاراکیس[2] [۵] طرح استفاده از بستر شبکه نرم‌افزارمحور در شبکه‌های فاقد زیرساخت را ارائه کرد. طرح بالا نشان می‌دهد که به‌کارگیری شبکه نرم‌افزار محور در شبکه‌های فاقد زیرساخت تاکتیکی (نظامی) که اغلب شامل تیم‌های یکپارچه و مختلف است، باعث تمرکز مدیریت در شبکه فاقد زیرساخت می‌شود. هانگ[3] و همکارانش [۶] از سامانه تشخیص نفوذ با معیار آستانه‌ای و هانی‌پات (تله‌عسل) به‌عنوان روش کاهش‌دهنده حمله، برای محافظت از کنترل‌کننده نرم‌افزارمحور در برابر حملات منع سرویس توزیع‌شده استفاده کرده‌اند. استفاده از این روش، درصد ریسک‌پذیری در تشخیص درست تهدیدات را افزایش می‌دهد. یان[4] و همکارانش [۷] یک الگوریتم برنامه‌ریزی کنترل‌کنندهٔ نرم‌افزارمحور چندصفی بر اساس استراتژی تخصیص برش زمانی را پیشنهاد کردند. نتایج شبیه‌سازی‌ها اثربخشی این طرح را نشان داده است.

در پژوهش‌های [۸]، به‌منظور بهبود رسیدگی به جریان یا انتقال داده، جایگزینی برای پروتکل OpenFlow با عنوان OpFlex توسط سیسکو[5] پیشنهاد شده است. در طرح بالا جهت غلبه بر چالش‌های امنیتی شبکه‌های مبتنی بر شبکه نرم‌افزارمحور و حذف جریان‌های مشکوک از فایروال هوشمند توزیع‌شده استفاده شده است.

بلاویستا[6] و همکارانش [۳] طی پیاده‌سازی‌های انجام‌شده، نشان دادند که شبکه نرم‌افزار محور به چه نحوی می‌تواند مدیریت کیفیت سرویس در شبکه‌ی فاقد زیرساخت را بهبود ببخشد. آن‌ها در این پژوهش‌ها جریان‌های مختلف ارسالی را در شبکه از لحاظ تداخل با توجه به وجود یا عدم وجود شبکه نرم‌افزارمحور، مسیر جایگزین، تأخیر در ارسال و پخش همگانی بررسی کردند.

لی[7] و همکارانش [۹] نوعی معماری شبکه نرم‌افزارمحور سلسله‌مراتبی را پیشنهاد کردند که از کنترل‌کننده‌های چندگانه توزیع‌شده بهره می‌گیرد. الگوی بالا تحت عنوان IRIS، یکی از پروژه‌های کنترل‌کننده در شبکه نرم‌افزارمحور توزیع‌شده است که هم‌زمان مدل سلسله‌مراتبی و توزیع‌شده مسطح را استفاده می‌کند.

قسمی[8] و همکارانش [۱۰] با توجه به قابلیت‌ها و معماری شبکه نرم‌افزارمحور از روش آنتروپی سریع برای بهبود امنیت ابر در برابر حملات منع سرویس توزیع‌شده استفاده کرده‌اند. این راه‌حل نیز مبتنی بر روش آستانه‌ای برای تشخیص نفوذ است. بوانی[9] و همکارانش [۱۱] چارچوب قابل تنظیمی را برای برنامه‌های مختلف در برابر حملات منع سرویس توزیع‌شده با عنوان چارچوب دفاعی پیش‌فعال[10] ارائه کردند. چارچوب پیشنهادی ضمن استفاده از کنترل‌کننده شبکه نرم‌افزارمحور توزیع‌شده، به‌منظور تشخیص حملات بالا و سهولت تنظیم برای انتخاب فیلترهای مختلف از روش تشخیص نفوذ با معیار آستانه‌ای استفاده می‌کند. برخی از پژوهش‌گران [۱۲] انواع حملات منع سرویس توزیع‌شده را بر اساس تخلیه پهنای باند (حملهٔ سیلاب و حملهٔ تقویت) یا حملات تخلیهٔ منبع (حمله بسته ناقص و حمله سوءاستفاده) طبقه‌بندی کرده‌اند. در پژوهش بالا راه‌حل‌های دفاعی این قبیل حملات بر اساس روش‌های مبتنی بر پیشگیری، تشخیص و کاهش نفوذ طبقه‌بندی شده است.

در این مقاله ضمن معرفی و شناسایی شبکه نرم‌افزارمحور و حملات منع سرویس توزیع‌شده و بررسی پژوهش‌های مختلف انجام‌شده، انواع راه‌حل‌های دفاعی مبتنی بر شبکه نرم‌افزارمحور در برابر حملات منع

[1] Tanweer Alam
[2] Poularakis
[3] Huang
[4] Yan
[5] Cisco
[6] Bellavista
[7] Lee
[8] Guesmi
[9] Bawany
[10] Proactive DDoS Defense Framework (Pro Defense)

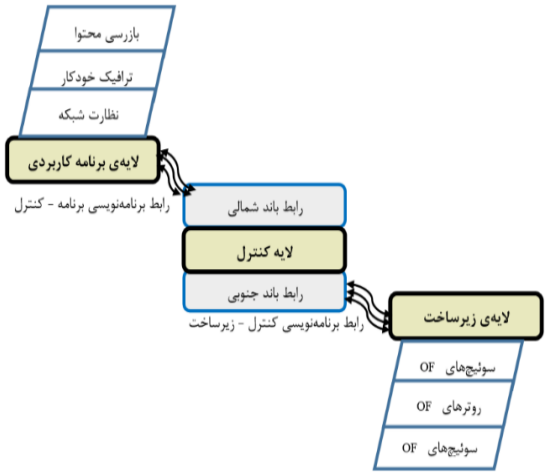
[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

سرویس توزیع‌شده را دسته‌بندی و در ادامه راه‌حل‌های دفاعی تشخیص نفوذ را به دو دسته تشخیص نفوذ با معیارهای آستانه‌ای و غیرآستانه‌ای طبقه‌بندی کردیم. راه‌حل‌های آستانه‌ای شناخته‌شده شامل روش‌های آنتروپی و یادگیری ماشین و راه‌حل‌های غیرآستانه‌ای شناخته‌شده شامل تجزیه و تحلیل الگوی ترافیک، نرخ اتصال و ادغام Snort با OpenFlow هستند؛ سپس برخی مثال‌های کاربردی از راه‌حل‌های دفاعی با قابلیت تشخیص نفوذ آستانه‌ای یا غیرآستانه‌ای مبتنی بر شبکه نرم‌افزارمحور که در پژوهش‌های مختلف ارائه شده را بررسی و در پایان ضمن جمع‌بندی مطالب، پیشنهادهایی را مطرح کردیم.

۳- استفاده از شبکه نرم‌افزارمحور

به علّت رشد فناوری بی‌سیم ناهمگن، شبکه‌های فاقد زیرساخت مبتنی بر شبکه نرم‌افزار محور نیز انعطاف‌پذیری، پویایی، مقیاس‌پذیری و فرصت‌هایی را برای حل مسائل امنیتی فراهم می‌کنند. شبکه نرم‌افزارمحور ضمن ارائه و معرفی برنامه‌ریزی، کنترل متمرکز، تجزیه و تحلیل ترافیک و بهبود امنیت، مجازی‌سازی و هوشمندی را در شبکه ایجاد می‌کند. با توجه به این‌که به‌کارگیری یک راه‌حل کنترل متمرکز با سطح تمرکززدایی، خراب‌کاری و تأخیر در محیط‌های فاقد زیرساخت ناسازگار است، کاربرد آن با چالش‌هایی روبه‌رو است [۱۳].

اساس کار شبکه نرم‌افزارمحور بر جداسازی هوش شبکه (کنترل) از پوسته (انتقال اطلاعات) است. این قبیل شبکه‌ها پس از انتشار جاوا توسط سان مایکروسیستمز[1] در سال ۱۹۹۵ رایج شدند. در شبکه‌های بالا کنترل‌کنندهٔ متمرکز کل شبکه را هدایت کرده و برنامه‌ریزی سبب پویایی شبکه، شخصی‌سازی و تعریف کاربری‌های جدید بر اساس نیاز هر سازمان می‌شود [۱۴].



(شکل-۱): اجزاء و معماری سه‌لایه‌ای SDN [۱۶]

۳-۱- اجزاء و معماری شبکه نرم‌افزار محور

شبکه نرم‌افزار محور شامل سه سطح (لایه) است. این سطوح مانند شکل (۱) شامل سطح داده، سطح کنترل و سطح برنامه است. معماری شبکه نرم‌افزارمحور بر اساس جداسازی سطح کنترل از سطح داده و صدور مجوز کنترل داده از یک کنترل‌کننده متمرکز با استفاده از پروتکلی امن مانند OpenFlow که برقرارکننده ارتباط بین سطوح داده و کنترل است، پایه‌ریزی می‌شود [۱۵]. استفاده از شبکه نرم‌افزارمحور رویکردی جدید با درنظرداشتن ایدهٔ «برنامه‌ریزی» به‌منظور مدیریت سطوح داده و کنترل است [۱۶].

وظایف سطوح مختلف شبکه نرم‌افزارمحور به‌شرح زیر است:

۱- سطح کاربرد[2]

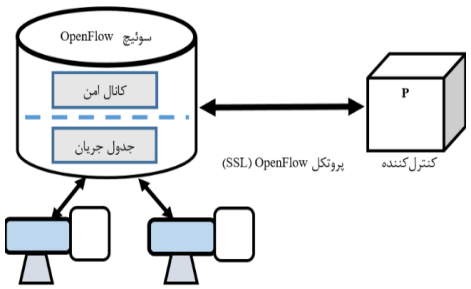
در سطح کاربرد به‌عنوان نخستین لایه، مدیریت خط‌مشی، مدیریت کاربر، مدیریت بهینه‌سازی و کیفیت سرویس انجام می‌شود. همهٔ توابع شبکه و ابزارهای نظارتی قسمتی از این لایه هستند [۱۶].

۲- سطح کنترل

در این معماری کنترل‌کننده متمرکز، مسئول ترجمه دستورهای لایهٔ کاربرد به مسیر داده‌های شبکه نرم‌افزارمحور است [۱۵،۱۶].

۳- سطح داده

از یک رابط بین سطوح داده و کنترل و مجموعه موتورهایی که انتقال داده‌ها را از طریق پردازش خارجی یا داخلی تراکنش‌های مسیریابی پیش می‌برند، ساخته شده است؛ درضمن ممکن است یک یا چند مسیر جهت ارسال داده وجود داشته باشد [۱۵،۱۶].



(شکل-۲): اجزاء سوئیچ OpenFlow [۱۶]

۳-۲- نحوهٔ عملکرد پروتکل OpenFlow

این پروتکل ارتباط امن بین سوئیچ و کنترل‌کننده را برقرار می‌کند. شبکه OpenFlow می‌تواند متشکل از

[1] Sun Microsystems

[2] Application

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

چندین شبکه فیزیکی مجهز به دستگاه‌های حمل و نقل مانند روتر و سوئیچ باشد. سوئیچ OpenFlow شامل سه بخش جداول جریان، کانال ارتباطی امن و پروتکل OpenFlow است که در شکل (۲) نشان داده شده است [۱۵]. سطح کنترل به‌عنوان کنترل‌کنندهٔ پروتکل فوق عمل و کنترل‌کننده عملیات به‌روزرسانی، افزودن یا حذف ورودی‌های جریان را به‌وسیله قوانین ازپیش‌تعریف‌شده اجرا می‌کند [۱۶].

۳-۳- ویژگی‌ها و مزایای شبکه نرم‌افزارمحور

شبکه نرم‌افزارمحور به‌عنوان روشی انعطاف‌پذیر جهت کنترل شبکه به‌صورت نظام‌مند ظاهر شده است. بستر بالا از پروتکل OpenFlow جهت برقراری ارتباط امن بین سطوح داده و کنترل بهره می‌برد. وجود برخی ویژگی‌های فوق‌العاده در شبکه نرم‌افزارمحور باعث حل بسیاری از مشکلات امنیتی فعلی شده است. شبکه نرم‌افزارمحور دارای ویژگی‌های مهمی از قبیل قابلیت برنامه‌ریزی، چابکی و پویایی، کنترل و مدیریت متمرکز، تجزیه و تحلیل ترافیک، به‌روزرسانی پویای قوانین حمل و نقل، بهینه‌سازی منابع و حفظ امنیت است [۱۷].

همچنین شبکه بالا دارای مزایایی نسبت به شبکه‌های مرسوم مانند استقرار آسان‌تر، ارائه خدمات جدید، کاهش هزینه مدیریتی و عملیاتی فناوری‌های ناهمگن، عملیات مؤثر زیرساخت‌های چندفروشنده، افزایش مداوم و شفاف عملیات و سرویس‌دهی، بهبود امنیت در شبکه، افزایش قابلیت برنامه‌ریزی عناصر شبکه در پلتفرم بی‌سیم توزیع‌شده و ممانعت از درگیرکردن تجهیزات است.

برقراری امنیت یک چالش بزرگ در شبکه‌های مبتنی بر شبکه نرم‌افزارمحور و به‌طورکلی در شبکه‌های موردی سیار با قابلیت بالا است. یکی از روش‌های رایج بهبود امنیت، به‌کارگیری معماری‌های مختلف شبکه نرم‌افزارمحور است که در ادامه آن را تشریح می‌کنیم.

۴-۳- انواع معماری کنترل‌کننده در شبکه نرم‌افزارمحور

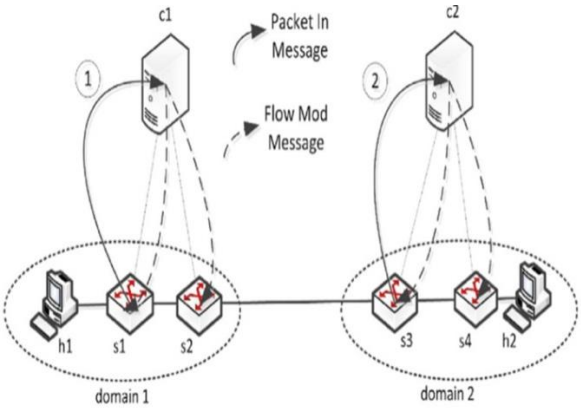
با توجه به عدم وجود استاندارد اجرایی برای استفاده از معماری‌های مختلف شبکه نرم‌افزارمحور در طراحی شبکه، مشتریان نسخه‌های متفاوتی را از کنترل‌کننده شبکه نرم‌افزارمحور برای رفع نیاز خود استفاده می‌کنند. انواع معماری شبکه نرم‌افزارمحور از لحاظ ساختار کنترل‌کننده شامل کنترل‌کننده‌های متمرکز واحد و چندگانه (معماری سلسله‌مراتبی، معماری توزیع‌شده کامل با دیدگاه محلی و جهانی) است. در ادامه این ساختارها را تشریح می‌کنیم.

۱-۴-۳- استفاده از یک کنترل‌کننده متمرکز (کنترل‌کننده متمرکز واحد)

کنترل‌کننده شبکه نرم‌افزار محور واحد، بخشی از شبکه با عنوان دامنه را کنترل کرده و اطلاعات محدودی از شبکه دارد. دامنه مجموعه‌ای از تمام سویچ‌هایی است که به‌طور مستقیم به یک کنترل‌کننده متصل شده و همهٔ میزبان‌ها به این سوئیچ‌ها متصل هستند. برای مثال در شکل (۳) c1 به‌صورت مجزا به‌عنوان کنترل‌کننده متمرکز واحد، فقط قادر به دیدن s2 و s1 و h1 است. حمله منع سرویس و منع سرویس توزیع‌شده می‌تواند در کنترل‌کننده با ساختار بالا رخ دهد که تنها نقطه شکست است [۱۵،۱۸].

۲-۴-۳- استفاده از کنترل‌کننده‌های چندگانه (سلسله‌مراتبی و توزیع‌شده)

به‌کارگیری کنترل‌کننده‌های متعدد می‌تواند ضمن افزایش میزان اعتماد و تحمل شکست، مشکل شکست نقطه‌ای را برطرف کرده و در صورت شکست یک کنترل‌کننده، امکان تداوم در کنترل شبکه را میسّر سازد. همچنین با ساختار بالا کنترل‌کننده‌ها برای تداوم در کنترل شبکه با همدیگر همکاری کرده و اطلاعات را مبادله می‌کنند. درضمن استفاده از ساختار بالا که از کنترل‌کننده‌های زیادی بهره می‌گیرد، سربار را افزایش می‌دهد [۱۵]. شکل (۳) ساختار شبکه نرم‌افزارمحور با کنترل‌کننده‌های چندگانه را نشان می‌دهد.



(شکل-۳): کنترل‌کننده‌های چندگانه با دامنه‌های مختلف و قابلیت مسیریابی مستقل بسته‌ها [۱۸]

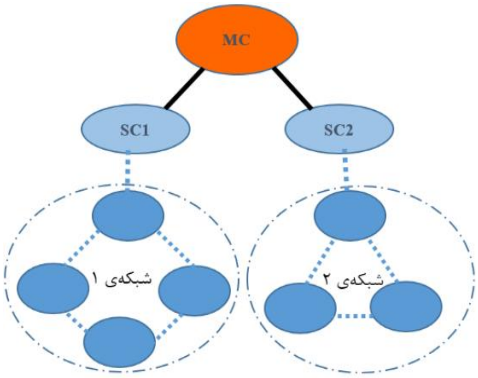
[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

معماری کنترل‌کننده چندگانه می‌تواند با استفاده از دو طرح کنترل‌کننده چندگانهٔ سلسله‌مراتبی و چندگانهٔ توزیع‌شده کامل (مسطح) با دیدگاه محلی و جهانی طراحی و اجرا شود.

۱- ساختار کنترل‌کنندهٔ چندگانه‌ی سلسله‌مراتبی

مدل سلسله‌مراتبی به‌عنوان یک معماری عمودی است و در ساختار بالا کنترل‌کننده ریشه (اصلی) هماهنگی‌های لازم را بین کنترل‌کننده‌های محلی ایجاد کرده و آن‌ها را مدیریت می‌کند. در این ساختار کنترل‌کنندهٔ ریشه به پایگاه داده جهانی دسترسی داشته و کنترل‌کننده‌های محلی قبل از این‌که بتوانند هر عملیات بین دامنه را اجرا کنند، می‌بایست اطلاعات شبکه را از کنترل‌کننده ریشه درخواست کند. در این مدل کنترل‌کنندهٔ ریشه به‌عنوان یک سرور و کنترل‌کننده‌های محلی به‌عنوان مشتری عمل می‌کنند. همچنین هیچ‌یک از کنترل‌کننده‌های محلی خوشه اتصال برنامه‌نویسی کاربردی شرقی - غربی با سایرین نداشته و فقط با کنترل‌کنندهٔ ریشه متمرکز ارتباط دارند.

در شکل (۴) کنترل‌کننده MC[1] به‌عنوان ریشه (اصلی) و SC1[2] و SC2 کنترل‌کننده‌های محلی (ثانویه) هستند. لازمه برقراری ارتباط و پاسخ به درخواست‌ها این است که هر SC توسط MC تأیید شود. فرایند برقراری ارتباط در دو مرحله ادامه می‌یابد. در مرحله نخست، یک SC درخواست خود را برای مدیر ارائه کرده و در مرحله دوم، MC با دسترسی به پایگاه داده، مجوز یا رد دسترسی SC درخواست‌کننده را صادر می‌کند [۱۵]. در این ساختار کنترل‌کنندهٔ ریشه، وضعیت شبکه جهانی و کنترل‌کننده‌های محلی، فقط حالت شبکه محلی خود را حفظ می‌کنند؛ بنابراین در ساختار بالا کنترل‌کنندهٔ ریشه با مدیریت بر کنترل‌کننده‌های محلی، اتصال و هماهنگی بین خوشه را فراهم می‌کند [۱۵،۱۸].

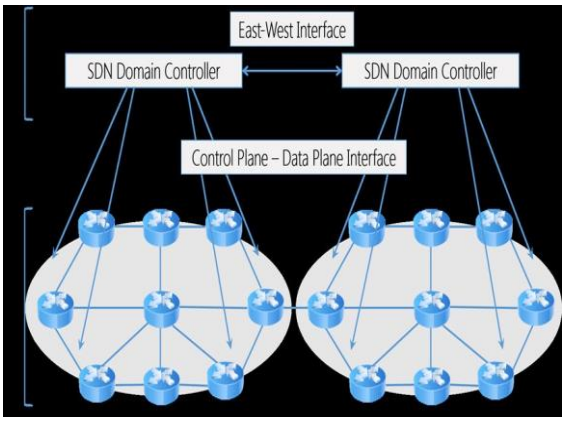


(شکل-۴): ساختار کنترل‌کننده‌های سلسله‌مراتبی [۱۵]

۲- ساختار کنترل‌کنندهٔ توزیع‌شدهٔ کامل (مسطح)[3]

کنترل‌کننده با ساختار توزیع‌شده کامل یا مسطح، برنامه کاربردی - کنترلی واقع در چند سرور موجود در شبکه را به‌منظور جمع‌آوری اطلاعات اجرا می‌کند. همچنین یک رابط برنامه‌نویسی برای پیاده‌سازی ویژگی‌های مدیریتی مانند مسیریابی و کنترل دسترسی ارائه می‌کند. شکل (۵) نمونه‌ای از ساختار شبکه نرم‌افزارمحور مجهز به کنترل‌کنندهٔ توزیع‌شدهٔ کامل (با دیدگاه محلی و جهانی) را نشان می‌دهد [۱۵].

کنترل‌کنندهٔ چندگانهٔ توزیع‌شده کامل با دیدگاه محلی، نشان‌دهندهٔ وضعیت فعلی شبکه در یک دامنه است که به‌صورت محلی در کنترل‌کننده ذخیره می‌شود. این حالت به کنترل‌کننده کمک کرده تا از هر رویداد مانند پیوستن یا ترک میزبان و اتصال به بالا یا پایین آگاه باشد؛ اما کنترل‌کنندهٔ چندگانهٔ به‌منظور به‌روزشدن، دسترسی به اطلاعات بیشتر و ساختن یک وضعیت شبکه جهانی (دیدگاه جهانی) باید بخش‌هایی از وضعیت شبکه محلی خود را به کنترل‌کننده‌های دیگر توزیع کند؛ بنابراین برخی از اطلاعات از قبیل حالت شبکه محلی ایستا و پویا، فهرست برنامه‌های نصب‌شده و دستگاه‌های متصل در کنترل‌کننده مبادله می‌شود.



(شکل-۵): ساختار کنترل‌کننده توزیع‌شدهٔ کامل (مسطح) [۱۹]

۵-۳- مقایسه ساختارهای مختلف کنترل‌کننده

به‌کارگیری ساختارهای مختلف کنترل‌کننده در معماری شبکه نرم‌افزارمحور، مطابق پارامترهای کلیدی زیر بسیار مؤثر است [۱۸]:

۱- مقیاس‌پذیری:

با توجه به وجود محدودیت در رسیدگی به درخواست‌ها و تعدّد درخواست‌های مسیر حمل و نقل، توزیع بار شبکه در میان کنترل‌کننده‌ها بسیار حائز اهمیت است.

۲- شکست:

[1] Main Controller
[2] Secondary Controller
[3] Flat

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

کنترل‌کننده متمرکز واحد دارای مشکل شکست نقطه‌ای در برابر تهدیداتی از قبیل منع سرویس و منع سرویس توزیع شده است.

۳- ثبات:

کنترل‌کننده‌های شبکه نرم‌افزارمحور چندگانه باید در هر زمان، همان وضعیت شبکه جهانی را داشته باشند.

۴- حریم خصوصی:

اشتراک‌گذاری اطلاعات محرمانه به‌وسیله هر کنترل‌کننده، نقض قوانین امنیتی در ارسال اطلاعات است؛ بنابراین هر کنترل‌کننده فقط مجاز به اشتراک‌گذاری اطلاعات فاقد طبقه‌بندی با سایرین است.

اکتیان[1] و همکارانش [۱۸] با درنظرگرفتن ساختارهای مختلف کنترل‌کننده در شبکه نرم‌افزارمحور، آن‌ها را بر حسب پارامترهای مهمی مانند مقیاس‌پذیری، شکست، ثبات و حریم خصوصی مقایسه کردند. نتیجه بررسی‌ها به‌صورت جدول (۱) خلاصه شد؛ اما نکته مهم در معماری‌های بالا این است که برای دست‌یابی به بهترین ساختار طی اولویت‌های کاری مختلف می‌توان روش‌های ترکیبی را استفاده کرد.

(جدول-۱): مقایسه معماری‌های مختلف شبکه نرم‌افزار محور از حیث ساختار کنترل‌کننده [۱۸]

نوع پارامتر	اولویت اول	اولویت دوم	اولویت سوم	اولویت چهارم
مقیاس‌پذیری	توزیع‌شده تخت با دید محلی	سلسله مراتبی	توزیع‌شده تخت با دید جهانی	متمرکز واحد
شکست	توزیع‌شده تخت با دید محلی و جهانی	------	سلسله مراتبی	متمرکز واحد
ثبات	توزیع‌شده تخت با دید جهانی	سلسله مراتبی	توزیع‌شده تخت با دید محلی	متمرکز واحد
حریم خصوصی	سلسله مراتبی	متمرکز واحد	توزیع‌شده تخت با دید محلی	توزیع‌شده تخت با دید جهانی

۴- تأثیر حملات منع سرویس توزیع‌شده در رایانش ابری

حملات منع سرویس و منع سرویس توزیع‌شده روش‌های اصلی مهاجمان برای ایجاد ممانعت از دسترسی کاربران به سرویس رایانش ابری است. حملات منع سرویس توسط یک نفر و حملات منع سرویس توزیع‌شده توسط دو یا چند نفر (بات) اجرا می‌شوند. به‌طورکلی، حمله منع سرویس توزیع‌شده شامل رئیس، دستیار، عامل (بات) و قربانی است. رئیس از طریق دستیاران با بات (زامبی) که یک عامل خطرناک است، ارتباط برقرار کرده و از آن‌ها استفاده می‌کند. دستیاران برنامه‌هایی هستند که بر روی تعدادی از دستگاه‌های در معرض خطر مانند Server نصب شده و مهاجمان برای ارسال دستورها با آن‌ها ارتباط برقرار می‌کنند. مهاجمان جهت یافتن یک دستگاه آسیب‌پذیر از روش‌های اسکن متنوعی استفاده کرده و با کمک دستیاران، عامل (بات) را کنترل می‌کنند. بات‌ها در هنگام استفاده از نرم‌افزار به‌وسیلهٔ یک کد بدافزاری نفوذ کرده و به سیستم قربانی حمله می‌کنند. در شکل‌گیری شبکه بات چنانچه زامبی‌ها بیشتر باشند، حمله مختل‌کننده‌تر است [۱۲].

۴-۱- طبقه‌بندی حملات منع سرویس توزیع‌شده

انواع عمده این قبیل حملات شامل حملات مبتنی بر تخلیه پهنای باند و تخلیه‌ی منبع است. حملات بالا طبق بررسی‌های انجام‌شده کل پهنای باند و منابع شبکه را مصرف کرده و بر اساس نوع آسیب‌پذیری به طرق زیر تقسیم می‌شوند:

۱- حملات تخلیه پهنای باند

حملات تخلیهٔ پهنای باند اغلب به‌صورت حملات سیلاب با ارسال حجم زیادی از ترافیک مخرب توسط ابزاری مانند «ترینو» به سمت قربانی اجرا می‌شوند. انسداد پهنای باند، اشباع و کندشدن سرعت شبکه قربانی به‌دلیل ترافیک مخرب روی نشانی IP و به‌کارگیری بسته‌های UDP و ICMP نشانهٔ آن است. حملات Smurf و Fraggle نمونه‌هایی از این‌گونه حملات هستند [۱۲].

۲- حملات تخلیه منبع

حملات تخلیه منبع جهت تخلیه منابع و جلوگیری از سرویس‌دهی به سیستم قربانی به‌صورت زیر دسته‌بندی می‌شوند [۱۲].

- پروتکل استفاده از آسیب‌پذیری‌ها (حملات سوءاستفاده)

هدف این نوع حملات، مصرف مازاد مقدار منابع قربانی است. حملات TCP SYN مثالی برای این نوع حملات است.

[1] Oktian

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

• حملات بسته ناقص

مهاجم بسته اطلاعاتی ناقص یا همراه با داده‌های مخرب را به‌سمت قربانی ارسال کرده و سبب هرج و مرج می‌شود. این حملات به دو شکل حملات نشانی IP و گزینه‌های بسته IP اجرا می‌شوند.

۴-۲- سناریوهای مرسوم حملات منع سرویس توزیع‌شده در شبکه نرم‌افزار محور

سناریوهای مختلف حمله منع سرویس توزیع‌شده در شبکه نرم‌افزار محور با توجه به شکل (۶) عبارتند از [۲۰]:

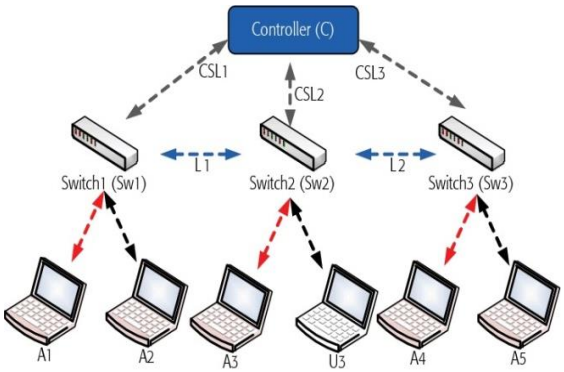
سناریوی ۱- ارتباط کنترل‌کننده و سوئیچ، هدف حمله باشد.

سناریوی ۲- منابع سیستم کنترل‌کننده هدف مهاجمان باشد.

سناریوی ۳- حافظه سوئیچ هدف حمله باشد.

سناریوی ۴- اتصال (ارتباط) سوئیچ‌ها هدف حمله باشد.

سناریوی ۵- کاربر مجاز هدف حمله مهاجم یا مهاجمان قرار گیرد.



(شکل-۶): توپولوژی SDN در برابر تهدیدات DDoS [۲۰]

۴-۲-۱- ارتباط کنترل‌کننده و سوئیچ هدف حمله باشد

در این سناریو مهاجمان تحت سوئیچ شماره (۱) می‌توانند حجم ترافیک را با نشانی IP دروغین افزایش داده و همهٔ بسته‌ها را همان‌طور که از IP ناشناخته می‌آید به کنترل‌کننده ارسال کند. در ادامه اتصال بین سوئیچ و کنترل‌کننده به‌طور بالقوه با حجم زیاد ترافیک روبه‌رو می‌شود. در سناریوی بالافوق ارتباط کنترل‌کننده با سوئیچ شماره یک (CSL1) هدف حمله بوده و مهاجم A1 است.

۴-۲-۲- منابع کنترل‌کننده هدف مهاجمان باشند

با توجه به این‌که ترافیک حمله از چندین سوئیچ که تحت مدیریت کنترل‌کننده هستند وارد شده، بار حمله تقسیم شده و بدین ترتیب تشخیص حمله دشوارتر می‌شود. حمله فوق، منع سرویس توزیع‌شده کور[1] نامیده شده است [۲۱]. هدف حمله در سناریوی بالا کنترل‌کننده و منابع آن و مهاجم A1، A3 یا A4 است.

۴-۲-۳- حافظه سوئیچ هدف حمله باشد

هر سوئیچ ضمن داشتن حافظه محدود، نیازمند ذخیره یک ورودی جدید برای هر جریان غیرمنطبق است. با ایجاد جریان‌های جدید توسط مهاجم، کنترل‌کننده قوانین ورودی‌های جدید را به سوئیچ ارسال می‌کند. چنانچه مهاجم از همهٔ ورودی‌های جدول جریان استفاده کند، ترافیک مجاز قادر به دسترسی نیست.

در توپولوژی بالا دو مهاجم A1 و A2 می‌خواهند L1 را مسدود کند؛ درحالیکه A4 و A5 از منابع L2 استفاده می‌کنند؛ بنابراین هدف یعنی سوئیچ شماره (۲) برای کاربران مجاز غیرقابل‌دسترسی است. در این سناریو هدف سوئیچ شماره (۲) و مهاجمان A1، A2، A4 و A5 هستند.

۴-۲-۴- اتصال (ارتباط) سوئیچ‌ها هدف حمله باشد

اجرای این حمله وقتی ساده‌تر می‌شود که مهاجمان تحت سوئیچ‌های مختلفی با یکدیگر در ارتباط باشند. به‌عنوان‌مثال A1 و A3 می‌توانند از تمام منابع ارتباطی قابل‌دسترسی سوئیچ‌های شمارهٔ (۱) و (۲) استفاده کند. این نوع حمله، حمله کورملت[2] [۲۲] نامیده شده است. در این سناریو L1 به‌عنوان هدف و A1 و A3 مهاجمان هستند.

۴-۲-۵- کاربر مجاز هدف حمله مهاجمان قرار گیرد

مهاجم می‌تواند روی همان سوئیچ یا سوئیچ دیگر قرار گیرد. اگر کنترل‌کننده یا سوئیچ نتواند مهاجم را شناسایی کند، منابع Server (کاربر مجاز) از بین می‌رود. در این سناریو U3 هدف و A1، A2، A3، A4 و A5 مهاجمان هستند.

۴-۳- چالش‌های شبکه نرم‌افزارمحور برابر حملات منع سرویس توزیع‌شده

رایانش ابری روشی مؤثر بر کاهش هزینه‌های سرمایه‌ای، هزینه‌های عملیاتی، سرویس‌دهی مناسب، دسترسی به پهنای باند شبکه، بانک اطلاعاتی و مقیاس‌پذیری بالا

[1] Blind
[2] Coremelt Attack

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

است. رایانش ابری ممکن است در معرض تهدیداتی شامل دسترسی غیرمجاز، نشت و تغییر داده، برنامه‌های مخرب، حمله به آسیب‌پذیری‌ها و غیره قرار گیرد.

نتایج پژوهش‌های مختلف نشان می‌دهد که استفاده از قابلیت‌های شبکه نرم‌افزار محور شامل تجزیه و تحلیل ترافیک نرم‌افزاری، کنترل متمرکز منطقی، به‌روزرسانی پویای قوانین و دید جهانی از شبکه سبب تسهیل تشخیص و مقابله با حملات می‌شود؛ اما دو ویژگی کنترل نرم‌افزاری و کنترل متمرکز شبکه، زمینه حملات منع سرویس توزیع‌شده در رایانش ابری را فراهم می‌کند [۲،۱۷،۲۰]. همچنین مواردی مانند جداسازی سطح داده از سطح کنترل، غیرفعال و محدودشدن قابلیت‌های سوئیچ‌ها (فقدان هوشمندی و تصمیم‌گیری جریان) از دلایل دیگر آسیب‌پذیری شبکه نرم‌افزار محور در برابر حملات بالا است [۲۰].

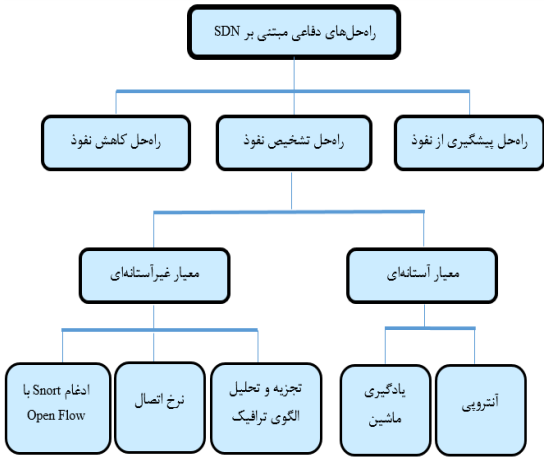
در ضمن شبکه نرم‌افزار محور با چالش‌های مختلفی در برابر حملات منع سرویس توزیع‌شده روبه‌رو است. مهم‌ترین چالش‌های شبکه نرم‌افزار محور مقابل این قبیل حملات در رایانش ابری، شامل موارد زیر است:

۱- ایجاد آسیب‌پذیری‌های بالقوه حملات منع سرویس توزیع‌شده در سیستم‌عامل‌های شبکه نرم‌افزارمحور
۲- میزان دسترسی به کنترل‌کننده
۳- مقیاس‌پذیری
۴- امنیت

بنابراین با توجه به آسیب‌پذیری ابر محاسباتی مبتنی بر شبکه نرم‌افزار محور در سرویس‌دهی و سهولت اجرای حملات منع سرویس توزیع‌شده، نیازمند به‌کارگیری راه‌حل‌های امنیتی - دفاعی مناسب در رایانش ابری هستیم که در ادامه به آن می‌پردازیم [۲،۲۰].

۵- انواع راه‌حل‌های دفاعی در برابر حملات منع سرویس توزیع‌شده

در اغلب موارد، مهاجمان همگام با رشد فزاینده حملات منع سرویس توزیع‌شده تلاش می‌کنند تا دسترسی غیرمجاز به سیستم قربانی داشته باشند. استفاده از راه‌حل دفاعی کارآمد می‌تواند از نفوذ مهاجمان جلوگیری کرده و ضمن تشخیص سریع نفوذ، تأثیر آن‌ها را به‌طور قابل‌توجهی کاهش دهد. راه‌حل‌های دفاعی همان‌طور که در شکل (۷) مشاهده می‌شود، به سه دسته راه‌حل‌های پیش‌گیری، تشخیص و کاهش نفوذ تقسیم می‌شوند. در این مقاله ضمن این‌که روش‌های مختلف پیشگیری و کاهش نفوذ را بیان می‌کنیم، انواع راه‌حل‌های تشخیص نفوذ مبتنی بر شبکه نرم‌افزارمحور را بر اساس روش آشکارسازی تهدیدات، به دو دسته کلی راه‌حل‌های تشخیص نفوذ آستانه‌ای و غیرآستانه‌ای طبقه‌بندی کرده و تکنیک‌های شناخته‌شده مربوط به هر یک را تشریح می‌کنیم.



(شکل-۷): دسته‌بندی راه‌حل‌های دفاعی مبتنی بر شبکه نرم‌افزار محور

در راستای جلوگیری، تشخیص و کاهش حملات منع سرویس توزیع‌شده اقدامات زیادی صورت گرفته و روش‌های دفاعی مناسبی توسط پژوهش‌گران ارائه شده است. در ادامه به بررسی اجمالی هر یک از این راه‌کارها می‌پردازیم.

۵-۱- راه‌حل‌های پیشگیری از نفوذ

بهترین استراتژی در برابر تهدیدات، جلوگیری از وقوع حملات است. یکی از این روش‌ها، استفاده از فیلتر است؛ بنابراین از انواع فیلترهای ورودی، فیلترهای خروجی و فیلترهای توزیع‌شده مبتنی بر مسیر استفاده می‌شود. فیلترهای توزیع‌شده مبتنی بر مسیر از اطلاعات مسیر برای مسدودکردن یا فیلتر نشانی IP بسته‌های جعلی استفاده می‌کنند. سایر فنون پیش‌گیری شامل غیرفعال‌کردن سرویس بدون استفاده، استفاده از بسته امنیتی، تغییر نشانی IP، غیرفعال‌کردن پخش همگانی IP و تله‌عسل است [۱۲].

۵-۲- راه‌حل‌های تشخیص نفوذ

سیستم تشخیص نفوذ از انتشار حملات منع سرویس توزیع‌شده و خرابی‌های دیگر جلوگیری نموده و با شناسایی رفتارهای غیرطبیعی و ناهنجار در عملکرد سیستم و سوءاستفاده، با حفظ پایگاه داده[1]، حفظ امضاهای شناخته‌شده و الگوهای مورداستفاده، نفوذ را

[1] Database

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

تشخیص می‌دهد. راه‌حل‌های تشخیص نفوذ به دو روش آستانه‌ای و غیرآستانه‌ای طبقه‌بندی می‌شوند. در ادامه برخی از معروف‌ترین راه‌حل‌های تشخیص نفوذ با معیارهای آستانه‌ای و غیرآستانه‌ای مبتنی بر شبکه نرم‌افزارمحور را معرفی می‌کنیم که عبارتند از:

۵-۲-۱- راه‌حل‌های تشخیص نفوذ با معیار آستانه‌ای

در روش آستانه‌ای، مدیر ابر یک سطح آستانه برای مجموعه‌ای از جریان‌های ارسالی تعیین می‌کند. اگر این مقدار بیش از سطح آستانه در یک اسلات زمانی باشد، این سیستم الگوریتم تشخیص نفوذ را راه‌اندازی می‌کند. درغیراین‌صورت، کنترل‌کنندهٔ شبکه نرم‌افزارمحور درخواست کاربر را به ابر ارائه می‌کند. مهم‌ترین روش‌های تشخیص نفوذ آستانه‌ای شناخته‌شده عبارتند از:

- تشخیص به روش آنتروپی[1]

از روش آنتروپی برای اندازه‌گیری تصادفی یک ویژگی در یک دوره زمانی خاص استفاده می‌شود. مقادیر آنتروپی بالا نشان‌دهندهٔ توزیع احتمال پراکنده‌تر و مقادیر آنتروپی پائین‌تر نشان‌دهندهٔ غلظت یک توزیع است؛ بنابراین از روش فوق برای تشخیص ناهنجاری گسترده در سامانه‌های مرسوم تشخیص نفوذ استفاده می‌شود. آنتروپی را می‌توان در ویژگی‌هایی مانند جریان‌های شبکه، آدرس‌های IP و تعداد بسته‌ها با میزان سربار پائین برای تشخیص نفوذ محاسبه کرد. به‌طورمعمول، طرح‌های تشخیص مبتنی بر آنتروپی تغییرات غیرقابل‌پیش‌بینی در سری‌های زمانی را از ویژگی‌های ترافیکی دقیق تشخیص می‌دهند [۱۱،۲۳].

- تشخیص حملات با روش یادگیری ماشین[2]

شبکه‌های عصبی مصنوعی، شبکه بیزی، نقشهٔ خودسازمان‌دهی و غیره به‌طور گسترده برای تشخیص ناهنجاری‌ها در سیستم تشخیص نفوذ[3] استفاده می‌شوند [۱۱]. این راه‌حل‌ها برای تشخیص حملات در شبکه‌های سیمی، شبکه‌های بی‌سیم و در تشخیص حمله منع سرویس توزیع‌شده مبتنی بر شبکه نرم‌افزارمحور بسیار مؤثر بوده و به‌طور گسترده استفاده می‌شوند [۲۴].

به‌طورکلی، راه‌حل مبتنی بر یادگیری ماشین جریان‌های شبکه را مطابق برخی خصوصیات مرتبط با ویژگی‌های ترافیک متمایز کرده و آن‌ها را به‌صورت بدخیم و خوش‌خیم بر اساس یک سطح آستانه محاسبه و طبقه‌بندی می‌کند [۲۵]. بااین‌حال، عملکرد این راه‌حل‌ها به مجموعه دادهٔ مورداستفاده برای آموزش وابسته است.

۵-۲-۲- راه‌حل‌های تشخیص نفوذ با معیار غیرآستانه‌ای

در روش‌های تشخیص نفوذ غیرآستانه‌ای برعکس روش آستانه‌ای هیچ‌گونه سطح آستانه‌ای برای تشخیص نفوذ انتخاب نشده و از پارامترهای دیگری از قبیل مشخصات بسته، الگوی امضا و غیره استفاده می‌شود. برخی از مهم‌ترین راه‌حل‌های تشخیص نفوذ غیرآستانه‌ای شناخته‌شده عبارتند از:

- تشخیص حملات به روش تجزیه و تحلیل الگوی ترافیک

روش بالا بر پایه این اصل که میزبان‌های آلوده الگوهای رفتاری مشابهی را نشان داده که با میزبان‌های خوش‌خیم متفاوتند، استوار است. در الگوهای ترافیکی مشابه، درنتیجهٔ فرمانی که برای بسیاری از اعضای همان شبکهٔ بات ارسال می‌شود، باعث رفتار مشابه مانند ارسال بسته‌های غیرمجاز، شروع‌کردن پویش و غیره می‌شود.

- تشخیص به روش نرخ اتصال

طبقه‌بندی روش‌های تشخیص ناهنجاری مبتنی بر نرخ اتصال شامل دو معیار ضریب موفقیت اتصال و تعداد اتصالات ایجادشده است. دسته نخست بیان‌گر ارتباطات موفقیت‌آمیز نسبت به کل اتصالات است؛ اما دومی تعداد اتصالات در یک پنجره معین از زمان را نشان می‌دهد [۱۱].

- تشخیص حملات با ادغام Snort و OpenFlow

Snort [۲۶] یک سیستم پیش‌گیری از نفوذ شبکه منبع باز و سیستم تشخیص نفوذ شبکه است.

Snort [۲۷] سیستمی است که می‌تواند با OpenFlow ادغام شود و با پیکربندی مجدد در حین اجرا این امکان را برای سیستم ابری فراهم سازد تا ضمن تشخیص نفوذ، اقدامات متقابل را انجام دهد [۱۱].

NICE [۲۸] راه‌حل انتخاب اقدامات متقابل، اندازه‌گیری و تشخیص آسیب‌پذیری توزیع‌شده چندفازی را ارائه کرده است. این نرم‌افزار از قابلیت‌های Snort و OpenFlow برای تشخیص نفوذ در سیستم مبتنی بر رایانش ابری استفاده می‌کند. Snort از روش تشخیص امضا برای شناسایی نفوذ استفاده و بسته‌های شبکه را برای شناسایی حملات شناخته‌شده و تطبیق آن‌ها کنترل می‌کند. هدف این راه‌حل، جلوگیری از به‌خطرافتادن ماشین‌های مجازی در سیستم مبتنی بر رایانش ابری است.

[1] Entropy
[2] Machine Learning
[3] Intrusion Detection Systems (IDS)

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

۳-۵- راه‌حل‌های کاهش نفوذ (پاسخ به تشخیص)

پس از تشخیص، روش‌های کاهش نفوذ جهت مسدودکردن یا پاسخ به تشخیص نفوذ استفاده می‌شود. جهت ردیابی و یافتن هویت مهاجم، روش‌های خاصی مانند ردیابی IP و ICMP، آزمایش پیوند و شماره‌گذاری تصادفی بسته کاربرد دارد [۱۲]؛ درضمن روش‌های رایج کاهش نفوذ شامل انداختن بسته[1]، مسدودکردن پورت[2]، تغییر مسیر[3]، کنترل پهنای باند[4]، تغییر توپولوژی شبکه[5]، بازرسی دقیق بسته[6]، تغییر نشانی IP و/ یا آدرس MAC، جداسازی (قرنطینه) ترافیک[7] است.

بنابراین در راستای بهبود امنیت شبکه و رفع آسیب‌پذیری ابر محاسباتی مبتنی بر شبکه نرم‌افزار محور برخلاف قابلیت‌های فراوان این قبیل شبکه‌ها در سرویس‌دهی نیازمند به‌کارگیری چارچوب یا راه‌حل امنیتی مناسب و مبتنی بر شبکه نرم‌افزارمحور در رایانش ابری هستیم [۲،۲۰]. در ادامه برخی مثال‌های کاربردی از انواع راه‌حل‌ها یا چارچوب‌های دفاعی تشخیص نفوذ آستانه‌ای یا غیرآستانه‌ای مبتنی بر شبکه نرم‌افزارمحور که در پژوهش‌های مختلف ارائه و پیشنهادشده را بررسی می‌کنیم.

۶- چارچوب دفاعی پیش‌فعال مبتنی بر شبکه نرم‌افزارمحور توزیع‌شده

اکثر راه‌حل‌های دفاعی در برابر حملات منع سرویس توزیع‌شده مبتنی بر محاسبه یک مقدار آستانه است. این مقدار آستانه به‌عنوان پایه‌ای برای تشخیص نفوذ بوده و برای همهٔ برنامه‌ها ثابت است و نمی‌توان آن را با توجه به برنامه‌های مختلفی که از شبکه استفاده می‌کنند، سفارشی کرد [۱۱].

همچنین با وجود یک مرکز داده (شبکه) در مقیاس بزرگ با برنامه‌های متنوع نیازمند یک راه‌حل قابل تنظیم با میزان حساسیت مختلف به‌منظور تشخیص حملات هستیم [۲۹]؛ بنابراین جهت رفع نیازمندی‌ها، چارچوب دفاعی پیش‌فعال مبتنی بر شبکه نرم‌افزارمحور توزیع‌شده را در ادامه شرح می‌دهیم [۱۱].

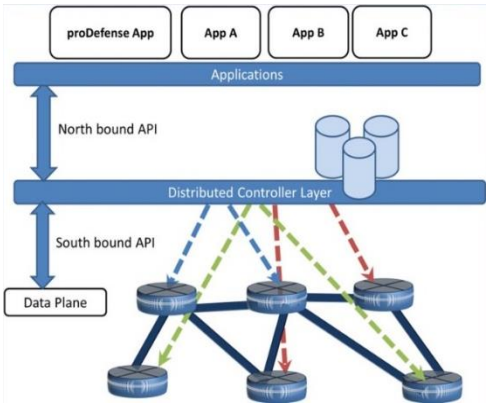
۶-۱- شناسایی چارچوب دفاعی پیش‌فعال

چارچوب دفاعی پیش‌فعال [۱۱]، با توجه به نیاز امنیت سایبری برنامه‌های مختلف، هشدارهای حمله منع سرویس توزیع‌شده را ارسال می‌کند؛ بنابراین راه‌حل تشخیص حمله یادشده شامل یک راه‌حل واکنشی قابل تنظیم برای تولید هشدارهایی است که با نیازهای خاص برنامه مطابقت می‌کند. چارچوب بالا از برنامه‌نویسی و ماهیت پویای شبکه نرم‌افزارمحور بهره برده و یک راه‌حل محافظتی و قابل تنظیم در برابر حملات منع سرویس توزیع‌شده ارائه می‌کند.

۶-۱-۱- معماری چارچوب دفاعی پیش‌فعال مبتنی بر شبکه نرم‌افزارمحور توزیع‌شده

در چارچوب پیش‌فعال کنترل‌کننده‌های شبکه نرم‌افزارمحور به‌صورت توزیع‌شده مستقر می‌شوند. ارتباط هر کنترل‌کننده با سوئیچ‌های متناظر توسط پروتکل امن OpenFlow برقرار می‌گردد. کنترل‌کننده پس از جمع‌آوری و به‌روزرسانی قوانین جریان، طی مراحلی مجاز یا غیرمجازبودن درخواست‌ها را مشخص کرده و دستورهای لازم را به سوئیچ OpenFlow ارسال می‌کند.

در این چارچوب با توجه به میزان اهمیت امنیت سایبری برنامه‌ها که اغلب مبتنی بر سیستم شهر هوشمند است، از راه‌حل تشخیص نفوذ قابل تنظیم و آستانه‌ای استفاده می‌شود. راه‌حل مورداستفاده از فیلترهای تطبیقی هشداردهنده در برابر حملات منع سرویس توزیع‌شده در کنترل‌کننده‌های شبکه نرم‌افزارمحور توزیع‌شده بهره می‌گیرد. شکل (۸) معماری چارچوب دفاعی پیش‌فعال را نشان می‌دهد [۱۱]. استفاده از کنترل‌کننده‌های توزیع‌شده، قابلیت اطمینان و مقیاس‌پذیری راه‌حل را برای رفع نیازهای یک مرکز دادهٔ (شبکه) بزرگ افزایش می‌دهد.



(شکل-۸): معماری چارچوب پیش‌فعال مبتنی بر SDN توزیع‌شده [۱۱]

[1] Drop packets
[2] Block port
[3] Redirection
[4] Control band width
[5] Network reconfiguration and topology change
[6] Deep packet inspection
[7] Quarantine or traffic isolation

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

۲-۱-۶- جنبه‌های امنیتی در برنامه‌های شهر هوشمند

شهر هوشمند نیازمند یک راه‌حل امنیتی قابل تنظیم جهت اجرای برنامه‌های متنوع است. این برنامه‌ها مانند سامانه حمل و نقل هوشمند، خدمات نجات هوشمند، شبکه امنیتی هوشمند با توجه به الزامات امنیت سایبری مختلف، به سه دسته بسیار بحرانی، بحرانی و متوسط تقسیم می‌شوند.

برنامه‌های ارائه‌دهنده خدمات بسیار بحرانی مانند شبکه امنیتی هوشمند و سامانه کنترل ترافیک، نیاز جدی برای برقراری امنیت سایبری دارند. این قبیل برنامه‌ها نمی‌توانند خطا را تحمل کنند. در ضمن برنامه‌های ارائه‌دهندهٔ خدمات بحرانی مانند مراقبت‌های درمانی و بهداشتی، روش‌های کاهش تهدیدات را در طول زمان مشخصی بعد از تشخیص حمله فعال می‌کنند. در این قبیل برنامه چون تأثیر حمله مهم و حیاتی است، نیازمند یک راه‌حل امنیتی چالاک هستیم. در مقابل برنامه‌هایی مانند هواشناسی، اخبار و ورزش الزامات امنیتی سایبری سخت‌گیرانه‌ای ندارند. این برنامه‌ها به یک چارچوب به‌نسبه چابک با میزان خطای مثبت - غلط بسیار پایین نیاز دارند [۱۱]. فیلترهای دفاعی موردنیاز برای برنامه‌هایی با میزان حساسیت سایبری بسیار بحرانی، بحرانی و متوسط به‌ترتیب شامل فیلترهای بسیار واکنش‌پذیر، متوسط و با کمترین واکنش‌پذیری است که در ادامه آن‌ها را تشریح می‌کنیم.

۲-۱-۶- فیلترهای کاربردی در چارچوب دفاعی پیش‌فعال

ضمن این‌که به‌کارگیری فیلترهای متوسط متحرک وزنی نمایی[1] به‌عنوان یک نوآوری است [۳۰]، چارچوب دفاعی پیش‌فعال، این قبیل فیلترها را برای سفارشی‌کردن تشخیص استفاده می‌کند.

معادله (۱) فرم اصلاح‌شده از معادله فیلتر EWMA مورداستفاده در چارچوب دفاعی پیش‌فعال را نشان می‌دهد.

$$PT_t = \alpha PT_{t-1} + (1-\alpha)CT_t + c \quad (۱)$$

PT_t: ترافیک پیش‌بینی‌شده، CT_t: ترافیک فعلی، α: بهره، c: مقدار ثابت و وابسته به ویژگی ترافیک است. به‌طورمعمول این فیلترها در صورت تشخیص حمله، سریع یا به‌آرامی واکنش نشان می‌دهند. اگر مقدار α زیاد باشد، حمله تشخیص داده نشده و برای α با مقادیر پایین‌تر، به‌صورت فوری هشدار حمله منع سرویس توزیع‌شده را اعلام می‌کند. انواع فیلترهای هشداردهنده‌ی مورداستفاده در این چارچوب شامل موارد زیر است [۱۱]:

۱- فیلتر بسیار واکنش‌پذیر[2]

چنانچه در معادله فیلتر α = 0/1 باشد، باعث غالب‌شدن اثر نرخ ترافیک فعلی می‌شود. فیلتر HR برای برنامه‌هایی در دسته بسیار بحرانی استفاده شده و بلافاصله هشدارهای امنیتی را صادر می‌کند.

۲- فیلتر واکنشی متوسط[3]

اگر در معادله α = 0/5 باشد، نرخ ترافیک فعلی و قبلی مساوی در نظر گرفته می‌شود. فیلتر IR (متوسط) ضمن استفاده برای برنامه‌ها در دسته بحرانی، جهت بیش‌تر شبکه‌ها مناسب است.

۳- فیلتر با کمترین واکنش‌پذیری[4]

اگر در معادله فیلتر α = 0/9 باشد، دارای بیشترین پایداری بوده و برابر حملات بسیار آهسته واکنش نشان می‌دهد. در برنامه‌های دسته متوسط که نیاز به چابکی امنیتی کمتری دارند، استفاده می‌شود. استفاده از فیلتر LR (با کمترین واکنش‌پذیری) تلاشی آگاهانه برای پایین‌آمدن نرخ خطای مثبت - غلط است.

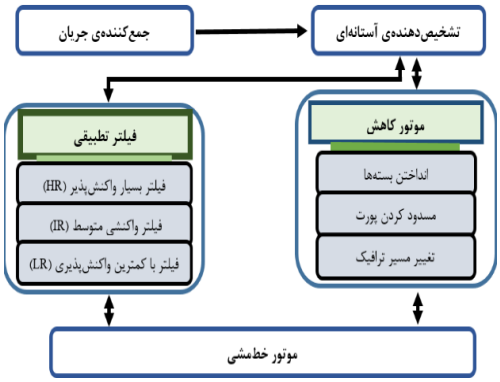
۲-۶- چارچوب دفاعی پیش‌فعال مبتنی بر شبکه نرم‌افزارمحور توزیع‌شده

چارچوب یادشده ماژولار است و مطابق با شکل (۹) مؤلفه‌های اصلی آن شامل جمع‌کننده جریان، موتور خط‌مشی، تشخیص‌دهندهٔ حمله و موتور کاهش است. موتور کاهش، فیلترهای تطبیقی و توابع تشخیص‌دهنده آستانه‌ای از هم جدا شده و می‌توانند با نیازهای امنیتی آینده سازگار شوند. جمع‌کننده جریان، ترافیک ورودی‌های جریان را از سوئیچ‌های OpenFlow جمع کرده و از موتور خط‌مشی برای تعریف خط‌مشی و پیکربندی نوع فیلتر (جهت تشخیص و کاهش حمله در شبکه) استفاده می‌شود [۱۱].

[1] Exponentially Weighted Moving Average (EWMA) Filters

[2] Highly Reactive (HR) Filter
[3] Intermediate Reactive (IR) Filter
[4] Least Reactive (LR) Filter

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]



(شکل-۹): طرح چارچوب دفاعی پیش‌فعال [۱۱]

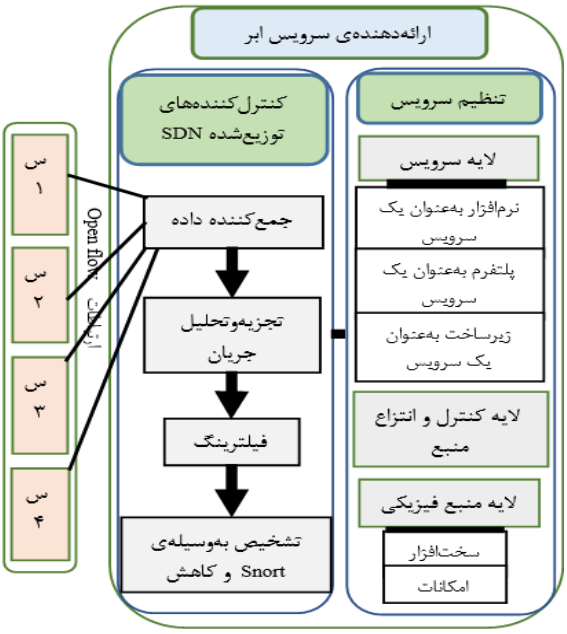
در این چارچوب بخش تشخیص، ورودی را از جمع‌کننده جریان دریافت کرده و هشدارهای امنیتی را با توجه به خط‌مشی تعریف‌شده و انتخاب فیلتر قابل تنظیم مناسب، تولید می‌کند. هشدارهای امنیتی زمینهٔ فعال‌شدن بخش کاهش توسط موتور خط‌مشی را برای انجام اقدامات مناسب فراهم می‌سازد. جهت کاهش حمله از برخی استراتژی‌های دفاعی مانند انداختن بسته، مسدودکردن پورت و تغییر مسیر ترافیک بهره می‌گیرد. چارچوب فوق جهت پاسخ‌گویی به الزامات یک راه‌حل دفاعی مؤثر در برابر حملات منع سرویس توزیع‌شده طراحی شده است [۱۱].

۳-۶- چالش‌های چارچوب دفاعی پیش‌فعال مبتنی بر شبکه نرم‌افزار محور

شبکه نرم‌افزار محور یک فناوری برجسته و در حال تکامل است و برای نسل آینده معماری‌های شبکه مفید است؛ اما وجود چالش‌های مختلف در چارچوب بالا، نشان‌دهندهٔ آسیب‌پذیری شبکه نرم‌افزار محور در برابر تهدیداتی مانند منع سرویس توزیع‌شده در محیط ابری است. اهم این چالش‌ها عبارتند از [۱۱]:

۱- مشکل‌بودن استقرار سخت‌افزار شبکه نرم‌افزار محور
۲- سربار نظارت بر ترافیک شبکه در زمان واقعی
۳- اشباع‌شدن کنترل‌کننده از حجم بالای ترافیک
۴- آسیب‌پذیری ناشی از برنامه‌های شبکه نرم‌افزار محور
۵- امنیت کنترل‌کننده و کانال‌های ارتباطی آن
۶- نوع راه‌حل و نحوه‌ی تشخیص ترافیک مخرب (نفوذ)

انتخاب راه‌حل و نحوهٔ تشخیص نفوذ بسیار مهم است. بسیاری از راه‌حل‌های مورداستفاده برای تشخیص نفوذ بر اساس سطح آستانه است. هشدارها زمانی ارسال می‌شوند که ترافیک شبکه از حداکثر حد مجاز آستانه عبور کند. چنانچه ترافیک غیرمجاز زیر سطح آستانه باشد، ناشناخته می‌ماند. همچنین حملات پیچیده‌ای که شبیه به الگوهای ترافیک مجاز هستند، تشخیص آن بسیار دشوار است؛ بنابراین به‌کارگیری الگوهای تشخیص نفوذ آستانه‌ای قابلیت اعتماد و اطمینان‌پذیری راه‌حل را بسیار می‌کاهد.



(شکل-۱۰): معماری SecCloudDD [۳۱]

۷- راه‌حل آنتروپی مبتنی بر شبکه نرم‌افزار محور

راه‌حل آنتروپی مبتنی بر معماری شبکه نرم‌افزار محور به‌منظور برقراری امنیت رایانش ابری در برابر حملات منع سرویس توزیع‌شده طی پژوهش‌هایی در [۳۱] پیشنهاد شد و به‌عنوان روش SecCloudDD معرفی شد. درواقع ویژگی‌های کنترل و برنامه‌ریزی متمرکز شبکه نرم‌افزار محور نظارت بر ترافیک ابر را مؤثرتر و امن‌تر می‌سازد؛ بنابراین مطابق شکل (۱۰) معماری SecCloudDD قادر به تشخیص مهاجمان در میان کاربران مجاز بوده و آن‌ها را جهت تثبیت سیستم در زمان واقعی مسدود می‌کند. در شکل زیر، سوئیچ‌های استفاده‌شده در سطح داده با س ۱، س ۲ و س ۳ مشخص شده است.

۷-۱- نگرش امنیتی در معماری SecCloudDD

در این بخش مراحل رویکرد را تشریح می‌کنیم:

۱- جمع‌آوری داده

برای هر درخواست جدید، سوئیچ‌های شبکه نرم‌افزارمحور ویژگی‌های درخواست از قبیل آدرس IP منبع، آدرس IP ابر و شمارنده را در جدول جریان به‌روزرسانی کرده و سپس به کنترل‌کننده ارسال می‌کنند. مرحله‌ی بعدی

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

جمع‌آوری آمارها از همهٔ سوئیچ‌های مرتبط با آن، جهت ثبت و ذخیره‌ی آن‌ها در جداول جریان در سطح جهانی است [۳۱].

۲- تجزیه و تحلیل داده

پس از انجام عملیات جمع‌آوری داده، کنترل‌کننده نحوهٔ تکمیل ترافیک را با استفاده از ستون شمارش جریان از جدول جریان جهانی آن نظارت می‌کند. مدیر ابر یک سطح آستانه برای مجموعه‌ای از جریان‌های ارسالی تعیین می‌کند. اگر این مقدار بیش از سطح آستانه در یک اسلات زمانی باشد، سامانه بالا الگوریتم تشخیص حمله را راه‌اندازی می‌کند. درغیراین‌صورت، کنترل‌کننده درخواست کاربر را به ابر ارائه می‌دهد [۳۱].

۳- تشخیص حمله

الگوریتم تشخیص بر اساس رویکرد آنتروپی سریع است و اندازهٔ پنجره[1] و آستانه دو جزء ضروری برای تشخیص حملات منع سرویس توزیع‌شده هستند. اندازه پنجره بر اساس یک فاصلهٔ زمانی است؛ درضمن چنانچه مقدار آنتروپی بیش از یک مقدار آستانه شود، سامانه ابری تحت حملهٔ منع سرویس توزیع‌شده قرار داشته و درصورتی‌که مقدار به‌دست‌آمدهٔ آنتروپی از سطح آستانه‌ی تعیین‌شده کمتر شود، سامانه ابری تحت حملهٔ منع سرویس توزیع‌شده قرار نگرفته است [۳۱].

آنتروپی $E(i, wt)$ برای یک شمارش جریان $C(i, wt)$ از یک اتصال خاص i در یک پنجره زمانی wt به‌وسیلهٔ معادلهٔ (۲) و (۳) محاسبه می‌شود [۳۱].

$$E(i, wt) = -\log \frac{C(i,wt)}{\sum_{i=1}^{n} C(i,wt)} + Q(i, wt) \qquad (۲)$$

Where

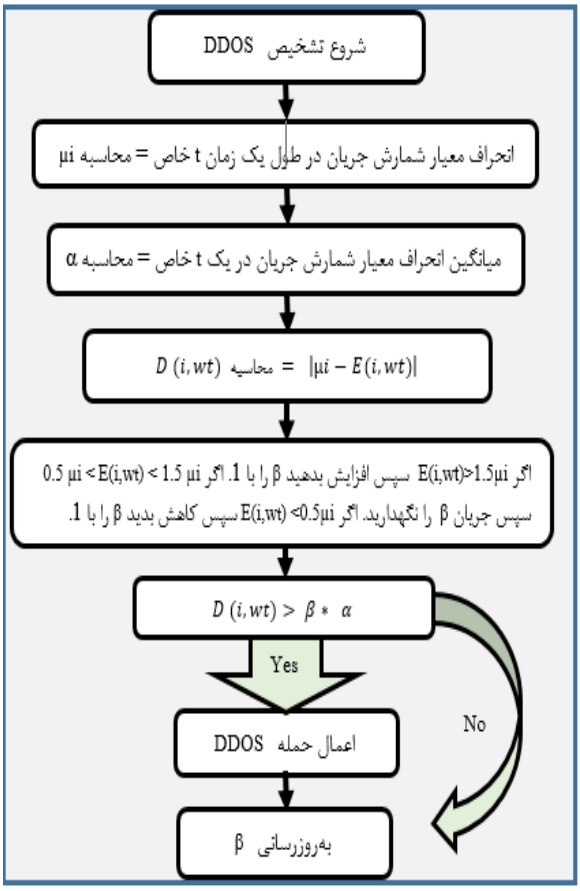
$$Q(i, wt) = \begin{cases} \left|\log \frac{C(i,wt+1)}{C(i,wt)}\right|, C(i, wt) \geq C(i, wt+1) \\ \left|\log \frac{C(i,wt)}{C(i,wt+1)}\right|, C(i, wt) < C(i, wt+1) \end{cases} \qquad (۳)$$

پس از محاسبه آنتروپی تمام شمارنده‌های جریان در فاصلهٔ زمان wt، الگوریتم تشخیص حملهٔ نشان‌داده‌شده در شکل (۱۱) مجموعه‌ای از متغیرهای زیر را ایجاد می‌کند [۳۱].

- μi = انحراف معیار شمارش جریان در طول یک زمان t خاص.
- $D(i, wt)$ = مقدار قدرمطلق تفاوت بین μi و $E(i, wt)$ به‌عنوان‌مثال:

$Di = |\mu i - E(i, wt)|$.

- α = میانگین انحراف معیار شمارش جریان در یک زمان خاص t.
- β = مقدار ضریب آستانه و عدد صحیح مثبت.
- ω = آستانه $(\omega = \beta * \alpha)$



(شکل-۱۱): الگوریتم آنتروپی سریع تطبیقی [۳۱]

در ادامه پس از محاسبه μi، Di را محاسبه کرده که نشان‌دهنده مقدار قدرمطلق تفاوت بین μi و $E(i, wt)$ است. اگر $Di \geq \omega$ باشد، شبکه ابر فعلی تحت یک حملهٔ منع سرویس توزیع‌شده در پنجره زمان فعلی wt است. درغیراین‌صورت، وضعیت ترافیک همواره نرمال بوده و حمله تشخیص داده نمی‌شود. هنگامی‌که کنترل‌کننده بسته‌های حمله را تشخیص می‌دهد، حمله منع سرویس توزیع‌شده را در سوئیچ‌های SDN اعلام کرده تا نشانی IP منبع ارتباط خاص (i) را قبل از رسیدن به سرور ابر مسدود کند.

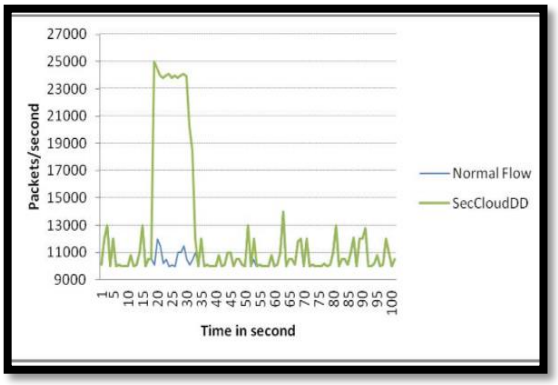
۲-۷- نتایج ارزیابی معماری SecCloudDD

ارزیابی و مقایسه‌ی نتایج حاصل از طرح بالا با الگوهای مشابه نشان داد که راه‌حل مبتنی بر معماری بالا، ترافیک ابر را کنترل کرده و از لحظه‌ای که مهاجم تعداد بسته‌ها را برای مصرف تمام منابع قربانی افزایش می‌دهد، به‌صورت فعال عمل می‌کند. در شکل (۱۲) خط آبی رفتار ترافیکی

[1] Window size

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

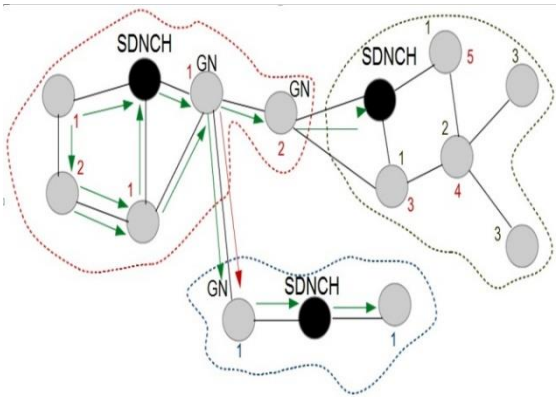
نرمال و خط سبز رفتار ترافیکی مخرب و تحت حمله منع سرویس توزیع‌شده را نشان می‌دهد. همچنین راه‌حل SecCloudDD ضمن کنترل و مسدودنمودن ترافیک مخرب و ارسال نتایج به تمام سوئیچ‌ها، آن را به‌موقع و به‌صورت نرمال تثبیت می‌کند. شبیه‌سازی نشان داد که تشخیص ناهنجاری و مسدودکردن آن می‌تواند در مدت پانزده ثانیه هنگامی‌که تعداد بسته‌های ارسالی به ۲۵۰۰۰ بسته در ثانیه برسد، انجام شود [۳۱].



(شکل-۱۲): تشخیص حمله DDOS به‌روش SecCloudDD [۳۱]

۸- معماری خوشه‌ای شبکه نرم‌افزارمحور

گونزالس[1] [۸] معماری خوشه‌ای شبکه نرم‌افزارمحور برای شبکه موردی سیار و اینترنت اشیا را طبق شکل (۱۳) پیشنهاد کرد. این معماری مبتنی بر شبکه حسگر خوشه‌ای نرم‌افزارمحور[2] است.



(شکل-۱۳): ارتباطات دادهٔ شبکه حس‌گر بی‌سیم مبتنی بر SDN [۸]

خوشه‌بندی شامل سازمان‌دهی شبکهٔ داخلی گروهی از گره‌ها، تحت یک ساختار سلسله‌مراتبی است. برای توسعه این معماری، کنترل‌کننده را در سرخوشه[3] قرار می‌دهیم.

در این معماری هر گره در فاصلهٔ بیش از (۱) از سرخوشه مستقر بوده و بیشینه قطر هر خوشه (۲) است. کاربران شبکه موردی با گره‌های دیگر از طریق سوئیچ سازگار تعبیه‌شده، اتصال برقرار می‌کنند. به‌طورمعمول شبکه بزرگ فاقد ساختار سازمانی، کارآمد نیست؛ بنابراین خوشه‌بندی شبکه با فرض هر سرخوشه به‌عنوان یک کنترل‌کننده پیشنهاد شده است. در هر سرخوشه از الگوریتم‌های تشخیص نفوذ با معیارهای غیرآستانه‌ای استفاده می‌شود. حالت‌های مختلف گره‌ها شامل گره ساده[4]، گره دروازه[5] و سرخوشه است. در رویکرد پیشنهادی، سرخوشه‌ی معماری **SDCSN**، سرخوشه شبکه نرم‌افزارمحور[6] نامیده می‌شود. هر خوشه، یک دامنه شبکه نرم‌افزارمحور نام دارد که با موارد زیر تعریف می‌شود.

- سرخوشه شبکه نرم‌افزارمحور، هماهنگ‌کننده دامنه است.
- گره دروازه، پل بین گره‌های حس‌گر و سرخوشه است.
- گره‌های حس‌گر، گروهی از گره‌ها در یک دامنه با گره‌های دروازه‌ی آن‌ها هستند.

در این معماری سرخوشه علاوه‌بر وظیفه مدیریتی روی دامنه شبکه و تشخیص نفوذ غیرآستانه‌ای [۸،۳۲،۳۳،۳۴]، به‌عنوان نگهبان امنیتی، نظارت و ایمنی مؤثر دامنه را حفظ و از تهدیدات داخلی و خارجی جلوگیری می‌کند.

در ادامه پژوهش‌گران پی بردند که خوشه پیشنهادی قادر به حل مشکلات مربوط به فرایندهای مسیریابی در معماری بالا نیست؛ بنابراین با ترکیب پروتکل‌های مسیریابی و قابلیت‌های شبکه نرم‌افزارمحور، معماری شبکه نرم‌افزار محور خوشه‌ای توزیع‌شده مبتنی بر پروتکل مسیریابی توسط گونزالس و همکارانش [۸] پیشنهاد شد.

۹- راه‌حل تشخیص نفوذ مبتنی بر Snort

Snort یک سامانه تشخیص نفوذ متن باز[7] است که توانایی تحلیل بدون تأخیر ترافیک شبکه و ثبت رویداد بسته‌ها روی IP شبکه را دارد. همچنین سامانه بالا از ارسال فوری هشدار به کاربر پشتیبانی کرده و حتی می‌تواند به‌عنوان

[1] Gonzalez
[2] Software Defined Cluster Sensor Network (SDCSN)
[3] Cluster Head
[4] Simple Node
[5] Gateway Node
[6] Software Defined Network Cluster - Head (SDNCH)
[7] Open Source

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

یک سامانه پیشگیری از نفوذ مورداستفاده قرار گیرد. این سامانه برای تشخیص بسیاری از تهدیدات و کاوش‌ها مانند سرریز بافر[1]، پویش درگاه[2]، رابط دروازه‌ی مشترک[3]، کاوش بلوک پیام سرور[4] و منع سرویس توزیع‌شده به‌کار گرفته می‌شود. Snort یک نرم‌افزار امنیتی پیشرفته است که در سه حالت به‌عنوان یک استراق‌سمع‌کننده[5] برای بسته‌های شبکه، ثبت‌کنندهٔ بسته‌ها و یک سامانه تشخیص نفوذ کامل مبتنی بر شبکه قابل برنامه‌ریزی و قابل استفاده است. این سامانه از روش تشخیص امضا برای شناسایی نفوذ استفاده کرده و بسته‌های شبکه را برای شناسایی حملات شناخته‌شده و تطبیق آن‌ها کنترل می‌کند. همچنین سامانه بالا می‌تواند مطابق قوانین نوشته‌شده، به‌صورت غیرآستانه‌ای و بر اساس مشخصات بسته از قبیل پهنای باند اشغالی، طول، نوع پروتکل، محتوا، سرآیند و غیره نفوذ را تشخیص دهد [۲۶].

به‌منظور تشخیص حملات منع سرویس توزیع‌شده در شبکه نرم‌افزارمحور، چارچوب Ryu SDN امکان ادغام با Snort را برای برقراری ارتباط با یکدیگر فراهم می‌کند. کنترل‌کننده شبکه نرم‌افزارمحور و Snort در میزبان‌های مشخصی که Snort بیشتر به‌عنوان Client و کنترل‌کننده به‌عنوان Server هستند، نصب می‌شوند. Snort هشدارهای امنیتی را برای تجزیه و تحلیل بیشتر با اجرای pigrelay.py که می‌تواند در https://github.com/John-Lin/ pigrelay. دریافت شود را به کنترل‌کننده شبکه نرم‌افزارمحور منتقل می‌کند. دستور Sudo python pigrelay.py$ مجموعه اطلاعات sFlow و هشدارهای Snort جهت حمله سیل ICMP را جمع‌آوری کرده و سپس به کنترل‌کننده ارسال می‌کند [۳۵]. در ادامه مدیر قوانین موردنظر را به‌منظور تشخیص حمله سیل ICMP در Snort تنظیم می‌کند. در [۳۵] جهت شبیه‌سازی حمله سیل به میزبان از دستور Hpig3 استفاده شده است.

نتایج آزمایش نشان می‌دهد که کنترل‌کننده شبکه نرم‌افزار محور می‌تواند با ایجاد فهرست سیاه حاصل از تجزیه و تحلیل راه‌حل دفاعی - امنیتی اطلاعات، نشانی‌های IP مشکوک مرتبط با نام‌های دامنه را نمایش دهد. همچنین به‌روزرسانی قوانین تشخیص نفوذ در زمان واقعی به‌وسیلهٔ امضاهای حمله می‌تواند به‌عنوان یک راه‌حل تشخیص نفوذ غیرآستانه‌ای مؤثر جهت کشف اتصالات مشکوک شبکه و تجزیه و تحلیل رفتار ناهنجار در Snort انجام شود.

[1] Buffer OverFlow
[2] Port Scan
[3] Common Gateway Interface (CGI)
[4] Server Message Block (SMB)
[5] Sniffer

۱۰- جمع‌بندی و پیشنهادها

برخی شبکه‌های مبتنی بر رایانش ابری به‌علت فقدان زیرساخت ثابت و مدیریت متمرکز با چالش‌های امنیتی مختلفی روبه‌رو هستند. شبکه نرم‌افزارمحور توزیع‌شده با درنظرداشتن ایده برنامه‌ریزی، کنترل متمرکز، تجزیه و تحلیل ترافیک، مدیریت کیفیت سرویس و غیره ضمن تزریق هوشمندی به شبکه می‌تواند به‌عنوان نگهبان امنیتی شبکه فاقد زیرساخت ثابت در رایانش ابری عمل نماید.

بررسی نتایج تحقیقات مختلف در سال‌های اخیر [۲،۱۸] نشان می‌دهد که استفاده از معماری شبکه نرم‌افزارمحور توزیع‌شده مسطح (کامل) ضمن افزایش مقیاس‌پذیری و ثبات، مشکل شکست نقطه‌ای ندارد؛ اما این فناوری به‌تنهایی در برابر حملات منع سرویس توزیع‌شده و حفظ حریم خصوصی آسیب‌پذیر است؛ بنابراین با توجه به رشد روزافزون حملات بالا در محیط ابری و گسترش شبکه‌ها، جهت بهبود امنیت شبکه‌های فاقد زیرساخت ثابت مبتنی بر شبکه نرم‌افزار محور، نیازمند به‌کارگیری راه‌حل‌های دفاعی مناسب مبتنی بر شبکه نرم‌افزار محور توزیع‌شده مسطح با دیدگاه جهانی هستیم [۲،۱۱،۱۸].

راه‌حل‌های دفاعی حملات منع سرویس توزیع‌شده اغلب شامل سه دسته راه‌حل‌های پیشگیری، تشخیص و کاهش نفوذ هستند. در این مقاله با بررسی مقالات مختلف، راه‌حل‌های تشخیص نفوذ مبتنی بر شبکه نرم‌افزارمحور را به دو دسته آستانه‌ای و غیرآستانه‌ای طبقه‌بندی کردیم. سپس راه‌حل‌های شناخته‌شده تشخیص نفوذ با معیار آستانه‌ای شامل روش آنتروپی و یادگیری ماشین و با معیار غیرآستانه‌ای شامل تجزیه و تحلیل الگوی ترافیک، نرخ اتصال و ادغام Snort و OpenFlow را تشریح کردیم.

در ادامه با بررسی برخی مثال‌های کاربردی به این نتیجه رسیدیم که آستانه‌ای‌بودن روش تشخیص نفوذ از قبیل چارچوب دفاعی قابل تنظیم پیش‌فعال [۱۱] و عملکرد راه‌حل‌های دفاعی تشخیص نفوذ با الگوی آنتروپی که ترافیک غیرمجاز را از مجاز بر اساس سطح آستانه متمایز می‌سازند [۶،۳۱]، درصد ریسک‌پذیری تشخیص نفوذ را افزایش داده و میزان آسیب‌پذیری شبکه را تشدید می‌کند. همچنین حملات پیچیده‌ای که شبیه به الگوهای ترافیک مجاز هستند، تشخیص آن بسیار دشوار است و چنانچه ترافیک غیرمجاز زیر سطح آستانه باشد، ناشناخته

[Downloaded from monadi.isc.org.ir on 2026-10-06]
[DOR: 20.1001.1.24763047.1399.9.1.1.4]

می‌ماند؛ در نتیجه به‌کارگیری الگوهای تشخیص نفوذ آستانه‌ای، قابلیت اعتماد و اطمینان‌پذیری راه‌حل را بسیار می‌کاهد؛ بنابراین به‌عنوان پژوهش‌های آتی و جهت بهبود امنیت شبکه می‌توان ضمن بررسی نقاط ضعف و قوت راه‌حل‌های مختلف دفاعی در برابر حملات منع سرویس توزیع‌شده، بر مسائل زیر تمرکز کرد.

الف) به‌کارگیری چارچوب دفاعی قابل تنظیم با معیارهای غیرآستانه‌ای مبتنی بر معماری شبکه نرم‌افزارمحور توزیع‌شده مسطح که در آن مسائل مربوط به حفظ حریم خصوصی رعایت شود؛ به‌عبارت‌دیگر، تلفیق چارچوب دفاعی قابل تنظیم مبتنی بر معیارهای غیرآستانه‌ای با ساختار شبکه نرم‌افزارمحور توزیع‌شده مسطح، میزان اعتماد و اطمینان‌پذیری را افزایش داده و سبب بهبود امنیت شبکه‌های فاقد زیرساخت ثابت و مدیریت متمرکز می‌شود.

ب) بهبود عملکرد سامانه‌های تشخیص نفوذ از طریق کاهش نرخ تشخیص اشتباه و انجام اقداماتی در زمینه کاهش زمان تشخیص.

ج) به‌کارگیری سامانه‌های تشخیص نفوذ هوشمند توزیع‌شده جهت غلبه بر چالش‌های امنیتی شبکه‌های مبتنی بر شبکه نرم‌افزارمحور و حذف جریان‌های مشکوک.

۱۱- مراجع

[1] Allam, Hesham, Nasser Nassiri, Amala Rajan, and Jinesh Ahmad, "A critical overview of latest challenges and solutions of Mobile Cloud Computing", In 2017 Second international conference on fog and mobile edge computing (FMEC) ،IEEE, pp. 225-229, 2017.

[2] Yan, Qiao, and F. Richard Yu, "Distributed denial of service attacks in software-defined networking with cloud computing", IEEE Communications Magazine 53, pp. 52-59, 2015.

[3] Bellavista, Paolo, Alessandro Dolci, and Carlo Giannelli, "MANET-Oriented SDN: motivations, challenges, and a solution prototype", IEEE , pp. 14-22, 2018.

[4] Alam, Tanweer, "Middleware Implementation in Cloud-MANET Mobility Model for Internet of Smart Devices", International jornal of Computer Science and Network Security, pp. 1902.09744, 2019.

[5] Poularakis, Konstantinos, George Iosifidis, and Leandros Tassiulas, "SDN-enabled tactical ad hoc networks: Extending programmable control to the edge", IEEE Communications Magazine 56, p. 132138, 2018.

[6] Huang, Xueli, Xiaojiang Du, and Bin Song, "An effective DDoS defense scheme for SDN", In 2017 IEEE International Conference on Communications (ICC)،IEEE, pp. 1-6, 2017.

[7] Yan, Q, Q. Gong, and F. Richard Yu, "Effective software-defined networking controller scheduling method to mitigate DDoS attacks", Electronics Letters 53, no, p. 469471, 2017.

[8] Gonzalez, Carlos, Salim Mahamat Charfadine, Olivier Flauzac, and Florent Nolot, "SDN-based security framework for the IoT in distributed grid", In 2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech), IEEE, pp. 1-5, 2016.

[9] Yang, Peng, Ning Zhang, Yuanguo Bi, Li Yu, and Xuemin Sherman Shen, "Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approac", IEEE, pp. 14-20 , 2017.

[10] Guesmi, Houda, and Leila Azouz Saidane, "Using sdn approach to secure cloud servers against flooding based ddos attacks", IEEE, pp. 309-315, 2017.

[11] Bawany, Narmeen Zakaria, Jawwad A. Shamsi, and Khaled Salah., "DDoS attack detection and mitigation using SDN: methods, practices, and solutions", Arabian Journal for Science and Engineering 42, pp. 425-441, 2017.

[12] Deshmukh, Rashmi V, and Kailas K. Devadkar, "Understanding DDoS attack & its effect in cloud environment", Procedia Computer Science 49, pp. 202-210, 2015.

[13] Nunes, Bruno Astuto A, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks", IEEE Communications Surveys & Tutorials 16, pp. 1617-1634, 2014.

[14] Raza, Muhammad H, Shyamala C. Sivakumar, Ali Nafarieh, and Bill Robertson, "A comparison of software defined network (SDN) implementation strategies", Procedia Computer Science 32, pp. 1050-1055, 2014.

[15] Alqallaf, Maha, and Bin Wang, "Software defined collaborative secure ad hoc wireless networks", In 2015 International Conference on Collaboration Technologies and Systems (CTS), IEEE, 2015.

[16] Ayesh aImran, "SDN Controllers Security

[Downloaded from monadi.isc.org.ir on 2026-10-06]

[DOR: 20.1001.1.24763047.1399.9.1.1.4]

Issues", University of Jyväskylä, MS Thesis document in Web Intelligence and Service Engineering, November 2017.

[17] Reddy, V. KRISHNA, and D. Sreenivasulu, "Software-defined networking with ddos attacks in cloud computing", International Journal of innovative Technologies (IJIT), pp. 3779-3783, 2016.

[18] Oktian, Yustus Eko, SangGon Lee, HoonJae Lee, and JunHuy Lam, "Distributed SDN controller system: A survey on design choice", computer networks, pp. 100-111, 2017.

[19] Allybokus, Zaid, Konstantin Avrachenkov, Jérémie Leguay, and Lorenzo Maggi., "MultiPath Alpha-Fair Resource Allocation at Scale in Distributed Software-Defined Networks", IEEE Journal on Selected Areas in Communications 36, pp. 2655-2666, 2018.

[20] Kalkan, Kubra, Gurkan Gur, and Fatih Alagoz, "Defense mechanisms against DDoS attacks in SDN environment", IEEE Communications Magazine 55, pp. 175-179, 2017.

[21] Ma, Duohe, Zhen Xu, and Dongdai Lin, "Defending blind DDoS attack on SDN based on moving target defense", In International Conference on Security and Privacy in Communication Networks, Springer, pp. 463-480, 2014.

[22] Studer, Ahren, and Adrian Perrig, "The coremelt attack", In European Symposium on Research in Computer Security,Springer, pp. 37-52, 2009.

[23] Wang, Rui, Zhiping Jia, and Lei Ju , "An entropy-based distributed DDoS detection mechanism in software-defined networking", IEEE, pp. 310-317, 2015.

[24] Abduvaliyev, Abror, Al-Sakib Khan Pathan, Jianying Zhou, Rodrigo Roman, and WaiChoong Wong, "On the vital areas of intrusion detection systems in wireless sensor networks", IEEE Communications Surveys & Tutorials 15, pp. 1223-1237, 2013.

[25] Xu, Yang, and Yong Liu, "DDoS attack detection under SDN context", In IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE, pp. 1-9, 2016 .

[26] Roesch, M, "Snort: lightweight intrusion detection for networks", In: LISA'99:13th Systems Administration Conference, pp. 229–238, 1999.

[27] Lakhina, Anukool, Mark Crovella, and Christophe Diot, "Mining anomalies using traffic feature distributions", In ACM SIGCOMM computer communication review, pp. 217-228, 2005.

[28] Chung, Chun-Jen, Pankaj Khatkar, Tianyi Xing, Jeongkeun Lee, and Dijiang Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems", IEEE transactions on dependable and secure computing 10, pp. 198-211, 2013.

[29] Bawany, Narmeen Zakaria, and Jawwad A. Shamsi, "Smart city architecture: Vision and challenges", International Journal of Advanced Computer Science and Applications, pp. 246-255, 2015.

[30] White, Joshua S., Thomas Fitzsimmons, and Jeanna N. Matthews, "Quantitative analysis of intrusion detection systems: Snort and Suricata", In 2013 Cyber Sensing International Society for Optics and Photonics, pp. 704-875, 2013.

[31] Guesmi, Houda, and Leila Azouz Saidane, "Using sdn approach to secure cloud servers against flooding based ddos attacks", IEEE, pp. 309-315, 2017.

[32] De Gante, Alejandro, Mohamed Aslan, and Ashraf Matrawy, "Smart wireless sensor network management based on software-defined networking", In 2014 27th Biennial Symposium on Communications (QBSC), IEEE, pp. 71-75, 2014.

[33] Yang, Fan, Vamsi Gondi, Jason O. Hallstrom, Kuang-Ching Wang, and Gene Eidson, "OpenFlow-based load balancing for wireless mesh infrastructure", In 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC), IEEE, pp. 444-449, 2014.

[34] El-Mougy, Amr, Mohamed Ibnkahla, and Lobna Hegazy, "Software-defined wireless network architectures for the Internet-of-Things", In 2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops), IEEE, pp. 804-811, 2015.

[35] Hsiao-Chung, L. I. N, and W. A. N. G. Ping, "Implementation of an SDN-based security defense mechanism against DDoS attacks", DEStech Transactions on Economics, Business and Management iceme-ebm, 2016.

مسعود محمدعلی پور تحصیلات خود را در مقاطع کارشناسی در سال ۱۳۸۱ در دانشگاه امام علی (ع) تهران و کارشناسی ارشد رشته مهندسی مخابرات امن و رمزنگاری در سال ۱۳۹۸ در دانشگاه شهید بهشتی تهران به پایان رسانده است. موضوعات پژوهشی مورد علاقه ایشان امنیت در شبکه‌های مخابراتی، اینترنت اشیا، زنجیره بلوکی، زیرساخت‌های SDN – NFV و مباحث مربوط به جنگ الکترونیک است.

[Downloaded from monadi.isc.org.ir on 2026-10-06]

[DOR: 20.1001.1.24763047.1399.9.1.1.4]

سعید شکرالهی تحصیلات خود را در مقطع کارشناسی رایانه-نرم‌افزار در سال ۱۳۸۱ از دانشگاه اصفهان و در مقاطع کارشناسی ارشد و دکتری رایانه-نرم‌افزار به‌ترتیب در سال‌های ۱۳۸۴ و ۱۳۹۳ از دانشگاه شهید بهشتی به پایان رسانده است. ایشان دوره فرصت مطالعاتی خود را در سال ۱۳۹۱ در آزمایشگاه امنیت دانشگاه میلان سپری کرده است. وی در حال حاضر استادیار گروه امنیت شبکه و رمزنگاری پژوهشکده فضای مجازی در دانشگاه شهید بهشتی است. زمینه‌های پژوهشی مورد علاقه ایشان عبارتند از: سامانه‌های فوق مقیاس وسیع، معماری نرم‌افزار، معماری سرویس‌گرا، معماری سازمانی، امنیت و کنترل دسترسی، اینترنت اشیا، میان‌افزارهای مبتنی بر رویداد و شبکه‌های بین خودرویی.

[Downloaded from monadi.isc.org.ir on 2026-10-06]

[DOR: 20.1001.1.24763047.1399.9.1.1.4]