تشخیص ناهنجاری در ترافیک شبکه با استفاده از سامانه توزیع‌شده مبتنی بر سیستم‌های چندعامله خودسازمانده

نیلوفر شکیبا[۱]، دانشجوی کارشناسی ارشد، اکرم بیگی[۲]، استادیار

۱- دانشکده مهندسی کامپیوتر - دانشگاه تربیت دبیر شهید رجایی - تهران - ایران - n_shakiba@vatanmail.ir
۲- دانشکده مهندسی کامپیوتر - دانشگاه تربیت دبیر شهید رجایی - تهران - ایران - akrambeigi@sru.ac.ir

چکیده: امروزه چالش‌های حوزه امنیت اطلاعات و ارتباطات بسیار مورد توجه محققین است. گسترش مرزهای شبکه، افزایش و پیچیدگی حملات امنیتی شبکه، نیاز به وجود سامانه‌های هوشمند، خودکار و بی‌درنگ کشف ناهنجاری و تهدیدات شبکه را دوچندان نموده است. برای کشف ناهنجاری، لازم است ترافیک شبکه به صورت بی‌درنگ مورد پایش قرار گیرد. ناهنجاری شامل تغییرات قابل‌توجه و غیرمعمول رفتار ترافیک شبکه در مقایسه با الگوهای رفتار نرمال آن است. در این مقاله به‌منظور کشف ناهنجاری، یک سامانه مبتنی بر سیستم‌های چندعامله خودسازمانده ارائه‌شده است. سیستم‌های چندعامله از عامل‌هایی که با یکدیگر برای رسیدن به هدف مشخصی تعامل دارند تشکیل‌شده‌اند. از این سیستم‌ها برای حل مسائلی استفاده می‌شود که حل آن برای یک عامل و یا به‌صورت یکپارچه مشکل است. معماری سامانه پیشنهادی مقیاس‌پذیر است و می‌تواند خود را با تغییرهای شبکه‌های امروزی وفق دهد. ارزیابی و تحلیل انجام شده روی سامانه پیشنهادی در مجموعه-داده NSL-KDD، نشان می‌دهد نرخ کشف ناهنجاری در ترافیک شبکه در مقایسه با روش‌های مطرح اخیر بهبود یافته است. همچنین با پیشنهاد الگوریتم‌هایی برای بهینه کردن انتخاب عامل‌ها و تعیین وزن تصمیم به طور هوشمند برای عامل‌ها، علاوه بر افزایش نرخ تشخیص ناهنجاری، زمان تحلیل رخدادها نیز کاهش داده شده است.

واژه‌های کلیدی: امنیت شبکه، تشخیص ناهنجاری، تشخیص نفوذ، سیستم‌های چندعامله، سامانه‌های مقیاس‌پذیر.

Anomaly Detection in Network Traffic using Distributed Self-Organizing Multi Agent Systems

Niloofar Shakiba, MSc[1], Akram Beigi, Assistant Professor [2]

1- Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, Email: n_shakiba@vatanmail.ir
2- Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, Email: akrambeigi@sru.ac.ir

Abstract: Challenges in the field of information and communication security are of great interest to researchers. The expansion of network boundaries, the intensification and complexity increase of network security attacks, has amplified the need for intelligent, automated and real-time systems to detect network anomalies and threats. To detect anomalies, network traffic needs to be monitored immediately. The anomaly involves significant and unusual changes in network traffic behavior compared to its normal behavior patterns. In this paper, in order to detect anomalies, a system based on self-organizing multi agent systems is presented. Multi agent systems are made up of agents that interact with each other to achieve a specific goal. These systems are used to solve problems that are difficult for a single agent to solve or integrate. The proposed system architecture is scalable and can adapt to changes in today's networks. The evaluation and analysis of the proposed system in the NSL-KDD dataset shows that the rate of anomalies detection has improved compared to the recently proposed methods. Also, by proposing an algorithm to optimize the agents' choices and another one for intelligent agents' decision weighting, the rate of anomaly detection is increased and the time of event analysis is reduced.

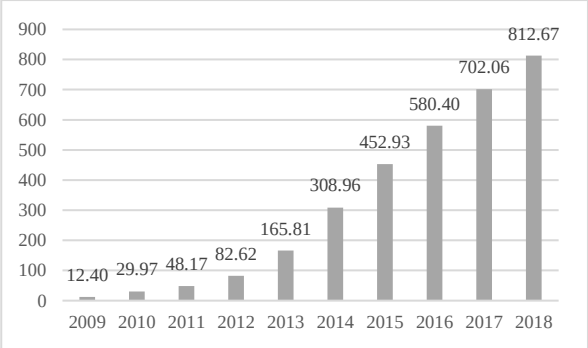
Keywords: Network security, Anomaly detection, Intrusion detection, Multi agent systems, Saleable systems.

نام نویسنده مسئول: اکرم بیگی
نشانی نویسنده مسئول: ایران - تهران - لویزان - دانشگاه تربیت دبیر شهید رجایی - دانشکده مهندسی کامپیوتر

۱- مقدمه

امنیت اطلاعات یکی از مسائل حائز اهمیت در دنیای امروز است. امنیت اطلاعات را می‌توان به حفاظت و صیانت از داده‌ها و جلوگیری از هرگونه فعالیت غیرمجاز تعبیر کرد. فعالیت غیرمجاز هرگونه فعالیتی است که یکی از سه مؤلفه امنیت اطلاعات یعنی یکپارچگی[۱]، محرمانگی[۲] و دسترس‌پذیری را مختل نماید. به‌منظور استقرار امنیت در شبکه‌های کامپیوتری، راه‌حل‌های متعددی وجود دارد که در این بین، سامانه‌های تشخیص نفوذ[۳] (IDS) بیشترین کاربرد را دارند. این سامانه‌ها یک راه حل خودکار دفاعی و یک سیستم امنیتی برای نظارت، پایش، تحلیل و کشف رفتارهای خصمانه و ناهنجار در شبکه هستند که می‌توانند مبتنی بر میزبان[۴] و یا مبتنی بر شبکه باشند[۱]. سامانه‌های تشخیص نفوذ با توجه به روش تشخیص نفوذ می‌توانند به سه دسته مبتنی بر امضا[۵]، مبتنی بر ناهنجاری[۶] و یا ترکیبی طبقه‌بندی شوند[۲]. IDSهای مبتنی بر امضا به یک پایگاه داده از اطلاعات انواع حملات نیاز دارند. نگهداری و به‌روزرسانی این پایگاه داده یکی از موضوعات چالش‌برانگیز در این نوع سامانه‌هاست. همچنین آنها نمی‌توانند حملات روز صفر[۷] که ناشناخته هستند و در پایگاه داده برای آن الگویی وجود ندارد را تشخیص دهند. بااین‌وجود این سیستم‌ها به علت دارا بودن نرخ پایین مثبت کاذب[۸] (تعداد رخدادهایی که از نظر امنیتی عادی و بی‌خطر هستند اما سامانه آنها را مخرب تشخیص داده و هشدار تولید نموده است)، هنوز هم در صنعت استفاده می‌شوند. در IDSهای مبتنی بر ناهنجاری، الگوهای رفتار هنجار تشخیص داده می‌شوند و سیستم تشخیص نفوذ به‌منظور پیدا کردن مغایرت با این الگوها ترافیک شبکه را پایش می‌کند [۳]. درواقع با تحلیل ترافیک شبکه و ساخت نمایه[۹] از رفتار شبکه می‌توانند هرگونه تغییر در رفتار معمول شبکه را تشخیص دهند.

اولین گام در تشخیص ناهنجاری، تعریف مسئله و مدل‌سازی درست آن است [۴]. بنابراین تعریف و تعیین رفتار هنجار یا نرمال شبکه مهم‌ترین گام برای تشخیص ناهنجاری است. به این منظور تعاریف و مصادیق مختلفی ارائه‌شده است. ناهنجاری در شبکه می‌تواند با قصد ایجاد تخریب و نفوذ صورت گرفته باشد و یا صرفاً یک رفتار غیرمعمول ولی بدون قصد تخریب باشد. از کار انداختن سرویس‌ها توسط مهاجمین و تلاش برای دسترسی به اطلاعات محرمانه شبکه مثال‌هایی از ناهنجاری مخرب هستند. رخدادهایی مثل انتقال یک فایل حجیم نیز مثالی از ناهنجاری در شبکه است که باوجود اینکه ماهیت خصمانه ندارد می‌تواند برای شبکه تهدید محسوب شود و سرویس‌دهی آن را مختل کند. از سویی دیگر، می‌توان مشاهداتی که با سایر داده‌های یک مجموعه همخوانی ندارد را ناهنجاری دانست [۵]. در حالت کلی، می‌توان الگوهایی از رفتار شبکه را که با رفتار هنجار تعریف شده تطابق ندارند را ناهنجاری به شمار آورد [۶]. با توجه به افزایش سریع حملات در شبکه و رشد صعودی آن در سالیان اخیر، همان‌طور که در شکل ۱ نشان داده شده است [۶]، کشف و پاسخ بی‌درنگ به عملیات نفوذ به شبکه، نیاز به فرآیندهای خودکار و هوشمندانه دارد.



شکل ۱: نرخ وقوع آلودگی با بدافزار (میلیون بار) [۷]

گسترش استفاده از شبکه و همچنین افزایش نیاز به اشتراک‌گذاری داده‌های مهم در انواع کسب‌وکارها باعث شده است که سیستم‌های سنتی کشف نفوذ که تا حد زیادی به فرد خبره وابسته هستند قادر به پاسخگویی به نیازهای امنیت نباشند. همچنین افزایش استفاده از موبایل و فنّاوری‌های بی‌سیم مرزهای شبکه را به‌وسیله معرفی تجهیزات پویا که در آن کاربران و ابزارها مدام در حال پیوستن به شبکه و جدا شدن از آنها هستند به‌شدت تغییر و گسترش داده است. برای سازگاری با این تغییرات، لازم است IDSها قابلیت مقیاس‌پذیری داشته باشند. عموماً IDSهای سنتی از یک واحد پردازش مرکزی به‌منظور تحلیل ترافیک شبکه استفاده می‌کنند که این امر احتمال خطای نقطه‌شکست مرکزی[۱۰] را افزایش می‌دهد. به‌منظور استقرار امنیت در شبکه‌های امروزی نیاز است تا از سیستم‌های تشخیص نفوذ بی‌درنگ، نامتمرکز، خودکار و مقیاس‌پذیر استفاده نمود. در این مقاله یک سامانه توزیع‌شده مبتنی بر سیستم‌های چندعامله خودسازمانده به‌منظور کشف ناهنجاری های ترافیک شبکه ارائه‌شده است. این سامانه از تعدادی عامل هوشمند تشکیل‌شده است که بنا بر ماهیت توزیع‌شدگی سامانه، هر عامل به‌تنهایی قادر به تحلیل و جمع‌آوری اطلاعات شبکه است. همچنین عامل‌های این سیستم می‌توانند به‌صورت هوشمند به شبکه اضافه و یا از آن کم شوند تا بتوانند با کمترین پیچیدگی و سربار، خود را با تغییرات موردنیاز شبکه وفق دهند. علاوه براین، تعداد عامل‌های درگیر در تحلیل شبکه بنا بر میزان اهمیت یک رخداد و گستردگی آن تعیین می شود. عامل‌های سیستم پیشنهادی قادرند ناهنجاری‌های ترافیک شبکه را یافته و از این طریق حمله‌های ناشناخته و یا پیچیده امنیتی را تشخیص دهند.

۲- مروری بر ادبیات موضوع

بررسی تحقیقات انجام‌شده در تشخیص ناهنجاری شبکه، مشخص‌کننده این است که بسیاری از روش‌های پیشنهادی در این حوزه از الگوریتم‌های خوشه‌بندی[۱۱]، طبقه‌بندی[۱۲]، تکاملی، تحلیل آماری و یا ترکیب چند الگوریتم مختلف استفاده کرده‌اند که عمده این روش‌ها ماهیت متمرکز دارند [۱]. در این حوزه رویکردهای دیگری نیز

در دسته مدل‌سازی با سیستم‌های چندعامله و همچنین الگوریتم‌های هوش جمعی[۱۳] مشاهده می‌شوند. با توجه به ویژگی‌های شبکه‌های امروزی، استفاده از سیستم‌های چندعامله به دلیل قابلیت توزیع شدن راه‌حل مناسبی به نظر می‌رسد.

طبق تعریف، عامل موجودیتی است که بتواند از طریق حسگر از محیط ادراک داشته و توسط عملگر بر محیط تأثیر بگذارد. عامل‌ها باید خودمختار بوده و دارای قدرت تصمیم‌گیری در رابطه با اینکه چه کاری باید انجام دهند تا به اهداف تعیین‌شده برسند داشته باشند و نیز بتوانند با دیگر عامل‌ها تعامل داشته باشند. سیستم‌های چندعامله از چندین عامل تشکیل‌شده‌اند که با یکدیگر برای رسیدن به هدف مشخص تعامل دارند. از سیستم‌های چندعامله در مسائلی استفاده می شود که حل آنها برای یک سیستم تک‌عامله و یا یکپارچه مشکل باشد [۸]. به صورت ایده‌آل انتظار می‌رود سیستم‌های مبتنی بر عامل مزیت های مقیاس‌پذیری و سهولت استقرارپذیری که معماری مبتنی بر عامل ارائه می‌دهد را دارا باشند. همچنین مشکلات رایجی مثل وجود خطای شکست‌مرکزی، سخت‌افزار گران برای فراهم نمودن مقیاس‌پذیری و ساختار عملیاتی نامنعطف را نداشته باشند. در این بخش تعدادی از پژوهش‌هایی که از سیستم‌های مبتنی بر عامل به‌منظور تحلیل امنیت شبکه استفاده نموده‌اند مرور خواهد شد.

در [۹] یک سیستم تشخیص نفوذ مبتنی بر عامل که در آن از چندین منبع اطلاعاتی استفاده می‌شود، پیشنهاد شده است. عامل‌ها از اطلاعاتی که از منابع نظامی به‌منظور استخراج فاکتورهایی مثل مکان جغرافیایی، چشم‌انداز سیاسی و انگیزه‌های ممکن حمله استخراج شده است، استفاده می‌نمایند. مزیت مهم این سیستم این است که از منابع اطلاعاتی باکیفیت بالا به‌منظور نتیجه‌گیری در رابطه با حمله استفاده می‌نماید. این سیستم داده‌ها را به دو دسته حقیقت و پیش‌فرض (به‌عنوان حقیقت تائید نشده) رده‌بندی می‌نماید. معماری پیشنهادی این مقاله به عدم وجود یک داده و یا اشتباه بودن آن حساس است. همچنین، عامل‌ها داده‌ها را به‌صورت زنده جمع‌آوری می‌کنند تا بتوانند به‌روزترین اطلاعات در دسترس را به‌منظور جلوگیری از افت کیفیت داده[۱۴]، که عموماً در مخازن اطلاعات مرکزی رخ می‌دهد، فراهم نمایند.

در [۱۰] از یک سیستم چندعامله سلسله مراتبی برای نظارت و گزارش نقض سیاست‌ها در یک شبکه‌امنیتی استفاده شده است. این سیستم از چندین نوع عامل تشکیل‌شده است که هرکدام کار بخصوصی را انجام می‌دهند (مانند نظارت بر رخدادها، تولید هشدار و ساخت گزارش). مدیر شبکه سیاست‌هایی را برای عامل‌ها تعریف می‌کند تا مدل سلسله مراتبی آنها پیاده سازی شود. این مدل ذاتاً به‌طور مرکزی اداره می شود و معایب IDSهای مرکزی که قبلاً ذکر شده است، در آن وجود دارد. عامل‌های بالاترین لایه، رخدادهای امنیتی را از طریق داده هایی که توسط عامل‌های لایه پایین‌تر جمع‌آوری‌شده است را رده‌بندی می‌نماید. استفاده از ساختار لایه‌ای به این‌گونه که یک دسته از عامل‌ها داده جمع‌آوری کنند و دسته دیگر آنها را تحلیل نمایند ممکن است دسترس‌پذیری را مختل کرده و گلوگاه کارایی ایجاد نماید.

در [۱۱] یک محیط برای جمع‌آوری شواهد قانونی[۱۵] معرفی‌شده است. در این محیط از سه نوع عامل برای جمع‌آوری داده، تحلیل داده و تولید هشدار استفاده‌شده است. نویسنده این مقاله عنوان کرده است که الگوی سیستم‌های چندعامله برای جمع‌آوری داده‌های قانونی بسیار مناسب هستند. زیرا می‌توانند در فضای شبکه پخش شوند و به جمع آوری شواهد بپردازند. بسیاری از IDSها فاقد این ویژگی هستند و تنها ارتباطات شبکه‌ای قابل مشاهده را نظارت می‌کنند.

در [۱۲] یک سیستم جمع‌آوری شواهد قانونی توزیع شده با روش سلسله‌مراتبی و حسگرهای قابل پیکربندی چندگانه توسعه داده شده است. این سیستم از انواع حسگرها برای جمع‌آوری و تجمیع داده‌ها به‌منظور استخراج طبیعت رخدادهای امنیتی استفاده می‌نماید. این سیستم نوع حمله را بر اساس اینکه کدام قسمت شواهد در طول جستجو وجود ندارد تشخیص می‌دهد. در محیط‌های سایبری پیچیده این راه‌حل مناسب است زیرا نبود اطلاعات لزوماً به معنای اینکه حمله ای وجود ندارد، نیست.

در [۱۳] کاربردهای رایج در زیرساخت‌های حیاتی شامل سیستم های تشخیص نفوذ مرور شده است. انعطاف‌پذیری سیستم به‌عنوان یک فاکتور مهم در سیستم چندعامله مورد تأکید قرارگرفته است. مهاجمین می‌توانند عامل‌ها را از طریق حمله منع سرویس از دسترس خارج نمایند. عامل‌ها باید قادر باشند که خود را با تغییرات شبکه وفق دهند چراکه ممکن است تعدادی از عامل‌ها قادر به ارتباط با سایر عامل‌ها نباشند. مدل‌های سلسله‌مراتبی اگر مسیر ارتباطی آنها سالم نباشد درست عمل نخواهند کرد.

در [۱۴] با استفاده از سیستم‌های چندعامله، روشی برای تشخیص حمله منع سرویس توزیع‌شده ارائه‌شده است که در آن عامل‌ها از روش بهینه‌سازی هوش‌جمعی برای برقراری ارتباط و تصمیم‌گیری دقیق استفاده می‌کنند. چندین نوع عامل تعریف‌شده‌اند که هرکدام وظایف خاصی دارند. به‌طور مثال عامل‌های تشخیص با استفاده از روش‌های تحلیل بی‌نظمی[۱۶] و کوواریانس، حملات را تشخیص می‌دهند و عامل های هماهنگ‌کننده ارتباط بین عامل‌ها و تصمیم‌گیری را برقرار می‌نمایند. همچنین عامل‌های ناظر در صورت مشاهده هرگونه رخداد غیرمعمول، عامل‌های تشخیص را فعال می‌نمایند.

در [۱۵] از سیستم‌های چندعامله به‌منظور کشف حملات مانای پیشرفته[۱۷] استفاده‌شده است. در این سیستم تشخیص نفوذ با به‌کارگیری معماری چندعامله و با استفاده از منابع داده خارجی، منشأ اتصالات مشکوک را جستجو می‌کند. در این محیط سه نوع عامل تعریف‌شده است: ۱) عامل مشاور به‌منظور پیدا کردن مکان آدرس‌های IP ۲) عامل تحلیل‌کننده به‌منظور مقایسه اتصالات مشکوک با الگوی ترافیک مشاهده‌شده ۳) عامل تمایزدهنده بین اتصالات ایجادشده توسط انسان و ربات.

در [۱۶] سیستم تشخیص نفوذی با سه لایه پیشنهادشده است. در این سیستم از هر دو روش مبتنی بر امضا و مبتنی بر ناهنجاری استفاده‌شده است. لایه اول با استفاده از قوانین ابتکاری برخی نفوذها را که بسیار شبیه نمونه‌های هنجار هستند تشخیص می‌دهد. لایه دوم با استفاده از خوشه‌بندی، نمونه ورودی که به مرکز خوشه ناهنجار نزدیک‌تر از مرکز خوشه هنجار هستند را نفوذ تشخیص می‌دهد. مرکز خوشه توسط الگوریتم ژنتیک محاسبه می‌شود. لایه سوم نیز با استفاده از یک رده‌بند جنگل تصادفی، نفوذها را تشخیص می‌دهد.

در [۴] یک سیستم چندعامله خودسازمانده به‌منظور جمع‌آوری شواهد حقوقی در سطح شبکه پیشنهادشده است. این سیستم با هدف سازگاری با شبکه‌های امروزی از بعد مقیاس‌پذیری ارائه شده است. در این سیستم‌عامل‌ها خودمختارند و نقش‌های مختلفی در محیط امنیت شبکه می‌پذیرند. ایجاد و به‌کارگیری خودکار عامل‌ها که در سیستم‌های تشخیص نفوذ سنتی توسط کارشناس صورت می‌گیرد یکی از وجوه تمایز برجسته این سیستم است.

سامانه پیشنهادی این مقاله ترافیک شبکه را به‌منظور کشف انواع ناهنجاری پایش می‌نماید. معماری این سامانه توزیع‌شده و مبتنی بر سیستم‌های چندعامله است. این سامانه با به‌کارگیری عامل‌های خودمختار و هوشمند می‌تواند خود را با تغییرات شبکه سازگار نماید. وجود قابلیت تشخیص ناهنجاری می‌تواند ضمن کشف انواع نفوذها و حملات، هرگونه نقض سیاست‌های شبکه را نیز گزارش دهد.

۳- سامانه تشخیص نفوذ مبتنی بر سیستم‌های چندعامله

در این پژوهش یک سامانه مبتنی بر سیستم‌های چندعامله با ساختار توزیع شده و خود سازمانده به‌منظور کشف ناهنجاری ارائه شده است. الگوریتم‌های توسعه‌یافته در این سامانه، با استفاده از ویژگی و قابلیت عامل‌ها، ترافیک شبکه را پایش می‌نمایند. الگوریتم انتخاب هوشمند گروه که در این سامانه توسعه داده‌شده است، با بهینه نمودن عامل‌های مشارکت‌کننده در پایش یک رخداد، سبب افزایش نرخ تشخیص ناهنجاری و همچنین کاهش زمان موردنیاز برای تحلیل رخداد موردنظر می‌شوند. الگوریتم تصمیم‌گیری وزن‌دار هوشمند نیز با هوشمند نمودن وزن‌دهی تصمیم عامل‌ها سعی در بهبود نرخ تشخیص ناهنجاری و زمان تحلیل دارد. در ادامه به توصیف سامانه پیشنهادی می‌پردازیم.

۳-۱- تعاریف

- تعریف ۱: یک رخداد را هر نوع اتفاق قابل‌تشخیص در شبکه درنظر می‌گیریم که از نظر امنیتی نیاز به توجه دارد.

مثلاً یک رخدادنما از نوع پروتکل Syslog با درجه سختگیری[۱۸] ۱ می‌تواند یک رخداد نیازمند به توجه باشد (درجه سختگیری میزان اهمیت رخدادنما را نشان می‌دهد که می‌تواند عددی بین ۰ تا ۷ باشد. هرچقدر این عدد کمتر باشد رخدادنما نیاز به توجه بیشتری دارد).

- تعریف ۲: اطلاعات جمع‌آوری شده از ترافیک شبکه به صورت زوج مرتب (f, v) از ویژگی و مقدار مطابق رابطه (۱) است:

$$(f,v)\ where\ \ f \in F,\ \ v \in V \qquad (۱)$$

در رابطه (۱)، F را به عنوان مجموعه تمام ویژگی‌های قابل‌استخراج از شبکه بوده در نظر می‌گیریم (f ها اعضای این مجموعه هستند) و به طور کلی V محدوده ممکن مقادیر متناظر با این ویژگی‌ها درنظر گرفته می‌شوند. v ها مقادیر ممکن مربوط به هر ویژگی هستند و لازم به ذکر است که مقادیر مجاز و ممکن هر ویژگی دارای محدوده خاص خود است و V به ازای هر ویژگی محدوده متفاوتی را تعریف می‌کند و به منظور تسهیل در درک عبارت به این شکل نشان داده شده است.

- تعریف ۳: سامانه تشخیص‌نفوذ مبتنی بر سیستم‌های چندعامله از تعدادی عامل تشکیل شده است که در دامنه‌های مختلف شبکه پخش‌شده‌اند. این سامانه طبق رابطه (۲) نمایش داده می‌شود:

$$System = \{G_1, G_2, \ldots, G_n\} \qquad (۲)$$

G_iها نماد عامل‌ها هستند. تعداد آنها برحسب نیاز تعیین می شود. هر عامل یک مجموعه ورودی و یک مجموعه خروجی دارد. مجموعه ورودی، شرایط شروع به فعالیت عامل را نشان می‌دهد. تولید خروجی توسط عامل می‌تواند شرایط سایر عامل‌ها را جهت شروع به فعالیت برقرار سازد. به‌طور مثال یک عامل می‌تواند با تحقق رخداد پویش درگاه فعال‌شده و پس از جمع‌آوری اطلاعات و تحلیل آنها، یک خروجی مبنی بر مشکوک بودن IP منبع تولید نماید. در ادامه عامل دیگری که شرایط فعالیت آن با رخداد فعالیت IP مشکوک محقق می شود شروع به فعالیت خواهد نمود. نحوه شروع فعالیت عامل i در رابطه (۳) آمده است. C_i مجموعه ورودی یا شرایط عامل i را نشان می‌دهد. I نیز کلیه رخدادهای محقق شده را نشان می‌دهد:

$$\forall (f_j, v_j) \in C_i \ \ , (f_j, v_j) \in I \qquad (۳)$$

با توجه به رابطه (۳) بدیهی است که برای شروع فعالیت عامل باید تمام مجموعه شرایط محقق شوند.

- تعریف ۴: تابع همبستگی در هر عامل، رخداد امنیتی را دریافت کرده و با توجه به رخدادهای امنیتی قبلی که در حافظه عامل وجود دارد میزان همبستگی را محاسبه می‌نماید. تابع همبستگی با رابطه (۴) تعریف می‌شود:

$$corr(e_{id})[0,1] \ \&e_{id} \{e_0, e_1, \ldots, e_n\} \qquad (۴)$$

در رابطه (۴)، $corr$ تابع محاسبه همبستگی در هر عامل را نشان می‌دهد. e_{id} نیز رخدادی با شناسه id است $\{e_0, e_1, \ldots, e_n\}$ رخدادهای ثبت‌شده قبل از e_{id} هستند.

- تعریف ۵: تابع تصمیم‌گیری در هر عامل، میزان مخرب بودن یک رخداد را محاسبه می‌نماید. رابطه (۵) این تابع را نشان می‌دهد:

$$p(c, coll, corr(e_{id}))[0,1] \qquad (۵)$$

p تابع تصمیم‌گیری است که میزان همبستگی محاسبه‌شده توسط واحد همبستگی را به همراه اطلاعات جمع‌آوری شده و شرایط محقق شده را به‌عنوان ورودی دریافت می‌کند و احتمال مخرب بودن رخداد

م شکوک را محا سبه می‌کند. c مجموعه شرایط عامل و $coll$ اطلاعات جمع‌آوری‌شده است. هر عاملی که شرایط آن محقق شود به جمع‌آوری اطلاعات از شبکه خواهد پرداخت.

- تعریف ۶: وضـــعیت دامنه فعالیت و گزارش محلی که درنهایت توسط عامل تولید می‌شود عبارت است از:

$$R_{local} = \{e_{id}, ts, g_{id}, coll, p(c, coll, corr(e_{id}))\} \quad (۶)$$

در رابطه (۶) متغیر ts زمان وقوع رخداد را نشان می‌دهد، g_{id} عامل تولیدکننده گزارش اســـت و $coll$ خروجی واحد جمع‌آوری‌کننده اطلاعات اســـت که شـــامل اطلاعات تکمیلی در مورد رخداد موردنظر است و ممکن است شرایط عامل‌های دیگری را محقق نموده و آنها را در فرآیند پایش مشـــارکت دهد. تابع p نیز میزان همبســـتگی محاسبه شده توسط واحد همبستگی را به همراه اطلاعات جمع‌آوری شـده و شـرایط محقق شـده به‌عنوان ورودی دریافت می‌کند و احتمال مخرب بودن رخداد امنیتی را محاسبه می‌کند. برای تصمیم‌گیری نهایی در رابطه با مخرب بودن و یا نبودن یک رخداد، گزارش نهایی تولید می شود.

- تعریف ۷: گزارش نهایی شـــامل n گزارش محلی اســـت. n تعداد عامل‌هایی است که در رابطه با یک رخداد امنیتی فعال‌شده‌اند و گزارش محلی تولید نموده‌اند. گزارش نهایی R_{global} طبق رابطه (۷) تعریف می شود:

$$R_{global} = \{R_{local\,1}, R_{local\,2}, \ldots, R_{local\,n}\} \quad (۷)$$

- تعریف ۸: برای محاســـبه میزان کفایت شـــواهد جمع‌آوری‌شـــده از رابطه (۸) اســـتفاده می‌گردد. بطوریکه مجموع قدرمطلق اختلاف میزان ناهنجاری محاسبه‌شده توسط هر عامل با مقدار ۰/۵، میزان قطعیت شواهد را نشان می‌دهد. به‌طورکلی هرچقدر میزان احتمال ناهنجاری با مقدار ۰/۵ تفاوت داشته باشد به معنای تصمیم دقیق تر است.

$$certainty = \sum_{R_{local}\,i=1}^{i=n} w_i \left| p_i - 0.5 \right| \quad (۸)$$

در رابطه (۸) P_i خروجی تابع تصمیم‌گیری عامل i است. این متغیر احتمال ناهنجار بودن رخداد را از دید عامل i نشان می‌دهد. w_i نیز وزن تصمیم عامل i را نشان می‌دهد. w_i به‌طور پیش‌فرض برابر ۱ است.

۳-۲- معماری عامل‌ها

هر عامل دارای واحدهای[۱۹] مختلفی به شرح زیر است:

- *به‌روزرسانی نمایه:*

یکی از روش‌های تشـخیص رفتار ناهنجار مدل‌ســازی رفتار هنجار شبکه است. شناخت دامنه‌های مختلف شبکه و ساخت نمایه برای آنها می‌تواند نرخ هشـــدارهای کاذب را نیز کاهش دهد. در این راســـتا واحد نمایه برای دامنه فعالیت عامل یک نمایه می‌سازد و به‌صورت پویا آن را به‌روزرسانی می‌کند.

- *حافظه:*

هر عامل دارای یک حافظه کوچک اســـت که در آن امتیاز عملکرد خود و تعدادی از آخرین گزارش‌های دریافتی را نگهداری می‌کند. تعداد گزارش‌های نگهداری‌شـده به‌صـورت ایسـتا در مرحله تنظیمات قبل از شبیه‌سازی تعیین می‌گردد. گزارش‌های نگهداری‌شده در روند محاسبه همبســتگی اســتفاده می‌شـود. با گذشـت زمان مشـخصـی، اعتبار این گزارش ها از بین می‌رود. زمان عدم اعتبار این گزارش ها می‌تواند به‌صورت ایستا و یا پویا در نظر گرفته‌شود که در آزمایش‌های انجام‌شده به‌صورت ایستا تعیین‌شده است.

- *همبستگی:*

این واحد، همبستگی بین رخدادها را محاسبه و تحلیل می‌نماید و می‌تواند زمینه تشخیص حملات چندگامی را فراهم کند.

- *جمع‌آوری اطلاعات:*

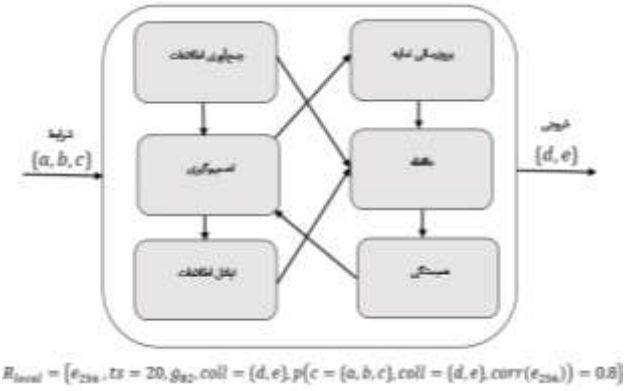
اطلاعات موردنیاز عامل از دامنه فعالیت آن را جمع‌آوری می‌کند.

- *تصمیم‌گیری:*

در این واحد می‌توان از کلیه روش هایی که برای تشـــخیص ناهنجاری وجود دارد استفاده نمود.

- *تبادل اطلاعات:*

این واحد وظیفه تبادل اطلاعات بین عامل‌ها را بر عهده دارد. عامل ها اطلاعات محلی جمع‌آوری‌شـــده را در قالب گزارش محلی با یکدیگر تبادل می‌نمایند. همچنین عامل‌ها نتیجه گزارش‌های نهایی را نیز به اطلاع یکدیگر می‌رسانند تا بتواند امتیاز عملکرد خود و نیز نمایه دامنه فعالیت را به‌روزرســـانی کنند. شـــکل ۲ نمای کلی معماری عامل‌ها را نشان می‌دهد. در این شکل a,b,c,d,e اطلاعات جمع‌آوری‌شده از شبکه هستند و یک نمونه از گزارش محلی تولیدشده نیز نشان داده‌شده است.



شکل ۲: معماری عامل‌های سامانه

۳-۳- سامانه توزیع‌شده خودسازمانده

برای مدل سازی شبکه‌های پیشرفته امروزی می‌توان از مفهومی به نام دامنه اسـتفاده کرد. با تعریف دامنه‌های مختلف می‌توان پیچیدگی‌های شبکه‌های امروزی را پیاده سازی نمود. با استقرار عامل‌ها در دامنه‌های شبکه می‌توان آن را پایش کرد. عامل‌های این سامانه وظیفه جمع‌آوری اطلاعات و تحلیل آنها را بر عهده‌دارند. هرکدام از عامل‌ها قادرند با در اختیار داشتن یک گزارش سراسری، در مورد ناهنجار بودن و یا نبودن

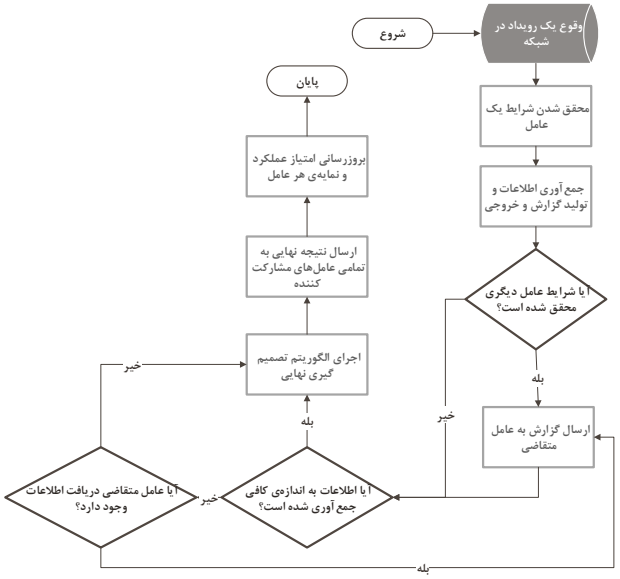
یک رخداد تصمیم‌گیری نمایند. در سامانه‌های متمرکز تصمیم‌نهایی در واحد پردازش‌مرکزی انجام می‌شود که در این صورت امکان بروز خطای شکست‌مرکزی وجود دارد. اما در سامانه توزیع شده پیشنهادی تمام عامل‌ها قابلیت تصمیم‌گیری نهایی دارند و امکان خطای شکست‌مرکزی از بین می‌رود. در این سامانه حتی اگر تعدادی از عامل‌ها از کار بیافتند باوجود احتمال افت عملکرد، همچنان شبکه پایش می‌شود.

به‌محض اینکه رخدادی در شبکه باعث تحقق شرایط یک عامل شود آن عامل شروع به جمع‌آوری اطلاعات در رابطه با آن رخداد می نماید. در روند جمع‌آوری اطلاعات، گزارش محلی و خروجی توسط عامل تولید می‌شود که در حقیقت جزئیات رخداد را از دید آن عامل نشان می‌دهد. خروجی تولیدشده توسط عامل ممکن است شرایط سایر عامل‌ها را برقرار کرده و آنها را در فرآیند پایش مشارکت دهد. در ابتدای تحلیل یک رخداد اطلاعات کمی از جزئیات آن وجود دارد. با وارد عمل شدن عامل‌ها و افزایش مشارکت آنها اطلاعات بیشتری درمورد رخداد موردنظر جمع‌آوری می‌گردد. برای تصمیم‌گیری نهایی برای برچسب رخداد پس از جمع‌آوری شواهد موجود به‌حدکفایت (طبق رابطه (۸))، برای آن یک گزارش نهایی تولید می‌شود.

برای بررسی ناهنجاری رخداد توسط هر عامل می‌توان از کلیه روش‌های موجودی که در ادبیات ارائه‌شده است استفاده کرد. بعد از تولید گزارش نهایی و تعیین برچسب نهایی رخداد، نتیجه این گزارش به اطلاع تمام عامل‌هایی که در فرآیند بررسی آن رخداد شرکت داشته اند خواهد رسید. در گام بعدی عامل‌ها عملکرد خود را با نتیجه نهایی مقایسه کرده و امتیاز خود را که g_{score} نام دارد به‌روزرسانی می‌کنند. این متغیر به ازای هر تصمیم‌نهایی مقدار احتمال صحیح محاسبه شده توسط عامل مذکور را به‌صورت جمع‌تجمعی نگهداری می‌کند. نگهداری تجمعی امتیاز عامل می‌تواند دید خوبی از عملکرد عامل در تحلیل رخدادهای پیشین ارائه دهد. هرچقدر این امتیاز بیشتر باشد به معنای عملکرد بهتر عامل است. عامل‌هایی که عملکرد ضعیفی دارند به‌مرور از فرآیند پایش دامنه حذف خواهند شد. حذف و اضافه شدن خودکار عامل‌ها به فرآیند تشخیص ناهنجاری باعث سازگار شدن سامانه با تغییرات شبکه خواهد شد.

عامل‌ها پس از دریافت نتیجه گزارش‌نهایی، نمایه دامنه‌فعالیت خود را نیز به‌روزرسانی می‌نمایند. عدم ناهنجار بودن یک رخداد در دامنه خاص باعث می شود تا این مورد روی تصمیم‌گیری‌های آینده در رابطه با رخداد مشابه تأثیرگذار باشد. زیرا ممکن است رخدادی که در وهله اول از دید عامل‌ها ناهنجار است یک رخداد معمول در شبکه مورد پایش باشد. عامل‌ها با به‌روزرسانی نمایه دامنه مورد فعالیت خود می توانند ویژگی‌های آن را نگهداری کنند. همچنین اگر رخدادی ناهنجار باشد می‌تواند ویژگی‌های ثبت‌شده در نمایه دامنه را تحت تأثیر قرار دهد و در صورت تکرار ممکن است تعدادی ویژگی را از نمایه حذف نماید. به‌عبارت‌دیگر خصوصیات ثبت‌شده در نمایه دامنه می‌توانند حذف یا اضافه گردند.

شکل ۳ یک نمونه از نحوه پایش رخداد در سامانه را نشان می‌دهد که در آن مثالی از جمع‌آوری اطلاعات در مورد یک رخداد و تولید گزارش نهایی و در انتها تصمیم‌گیری در مورد آن ارائه‌شده است. سیاست در پیش‌گرفته شده در این روش این است که شواهد موجود در رابطه با رخداد موردنظر به اندازه کفایت جمع‌آوری شود. در فرآیند اجرای الگوریتم نهایی و نحوه جمع‌آوری اطلاعات توسط عامل‌ها می توان از روش‌های مختلفی استفاده کرد. در ادامه الگوریتم‌های پیشنهادی که به این منظور توسعه داده‌شده‌اند آمده است.



شکل ۳: مثالی از نحوه تولید گزارش نهایی

۳-۴- الگوریتم‌های توسعه داده‌شده

مطابق با معماری سیستم پیشنهادی برای تعامل عامل‌ها باهم می‌توان از سیاست‌های متفاوتی استفاده کرد. همچنین به‌منظور تصمیم‌گیری نهایی در رابطه با ناهنجار بودن و یا نبودن یک رخداد نیز می‌توان از الگوریتم‌های مختلفی استفاده نمود. در ادامه الگوریتم‌هایی که برای این منظور توسعه داده‌شده‌اند معرفی می‌گردد.

۳-۴-۱- الگوریتم طبقه‌بندی پایه

الگوریتم پایه صرفاً به‌منظور مقایسه عملکرد الگوریتم‌های پیشنهادی و اندازه‌گیری میزان بهبود آنها ارائه شده است (شکل ۴). در این الگوریتم وقتی شرایط یک عامل محقق می‌گردد عامل موردنظر شروع به فعالیت نموده و خروجی تولید می‌نماید. الگوریتم پایه گزارش محلی تولید شده توسط یک عامل را به‌صورت تصادفی به یکی از عامل‌های متقاضی در یافت گزارش ارسال می‌نماید. هر گزارش نهایی مجموعه‌ای از گزارش‌های محلی است که توسط عامل‌ها تولید می‌شود. الگوریتم تصمیم‌گیری نهایی به ازای هر R_{global} و به روش اکثریت آرا به‌کار گرفته می‌شود و رخداد موردنظر در کلاس مخرب یا معمولی طبقه‌بندی می‌شود (خط ۷ تا ۱۰ شکل ۴). شایان‌ذکر است الگوریتم تصمیم‌گیری نهایی در هر یک از عامل‌ها می‌تواند اجرا شود و تنها شرط اجرای آن

جمع‌آوری شواهد در حد قابل‌قبول است که این مقدار می‌تواند به‌وسیله تعریف یک حد آستانه پویا و یا ایستا مشخص شود. عدم تصمیم‌گیری نهایی در یک واحد پردازش مرکزی باعث می شود که سامانه مذکور به شکل توزیع‌شده عمل کند و وابسته به پردازشگر مرکزی نباشد و دچار خطای شکست‌مرکزی نیز نشود. همچنین اگر به هر دلیلی عاملی که در حال اجرای الگوریتم تصمیم‌گیری نهایی است از کار بیافتد عاملی که بلافاصله قبل از آن مشغول فعالیت بوده به پایش شبکه ادامه خواهد داد. هر عامل بعد از ارسال گزارش به عامل دیگر، مدت‌زمان مشخصی منتظر دریافت پاسخ (سیگنال) از عامل دریافت‌کننده گزارش خواهد ماند. دریافت این سیگنال به این معنی است که الگوریتم تصمیم‌گیری نهایی اجرا شده و یا گزارش به عامل دیگری ارسال‌شده است.

بعد از طبقه‌بندی رخداد، برچسب آن به اطلاع عامل‌هایی که در فرآیند جمع‌آوری اطلاعات و تحلیل رخداد مشارکت داشته‌اند خواهد رسید. سپس عامل‌ها عملکرد خود را با نتیجه نهایی مقایسه کرده و امتیاز خود را (g_{score}) به‌روزرسانی می‌کنند (سطر ۱۱ تا ۱۷ شکل ۴).

Algorithm: Baseline Algorithm

Inputs

R_{global} // a global report
$R_{local} = \{e_{id}, ts, g_{id}, (f, v), p\}$
g_p //participant agents
$a \in (0,1)$ //a threshold for marking event as malicious
$Num_{malicious} = 0$
$Num_{innocouoc} = 0$

1: **Procedure: Final decision**
2: For $R_{global} = \{R_{local}, \ldots, R_{local}\}$ do
3: For $R_{local} = \{e_{id}, ts, g_{id}, (f, v), p\} \in R_{global}$ do
4: if $p \geq a$ then //the event is malicious in local report
5: $Num_{malicious}$ ++
6: else
7: $Num_{innocouoc}$ ++
8: if $Num_{malicious} \geq Num_{innocouoc}$ then
9: Return <e_{id}, decision =malicious>
10: else
11: Return <e_{id}, decision =innocuous>
12: For agent in G_p do
13: If decision =malicious:
14: $g_{sore} = g_{sore} + p$
15: update(profile)
16: else
17: $g_{sore} = g_{sore} + (1-p)$
18: update(profile)

شکل ۴: الگوریتم طبقه‌بندی پایه

۳-۴-۲- الگوریتم انتخاب هوشمند گروه

برای اینکه عامل‌ها بتوانند خود را با بهتر و سریع‌تر با تغییرات شبکه وفق دهند و زمان تحلیل رخدادها نیز کاهش یابد، الگوریتم انتخاب هوشمند گروه ارائه شده است (شکل ۵). عملکرد این الگوریتم به این صورت است که وقتی یک گزارش محلی توسط یک عامل تولید شد، تعدادی عامل که شرایط آنها محقق شده است، درخواست دریافت این گزارش را خواهند داد. الگوریتم انتخاب هوشمند گروه با توجه به امتیاز عملکرد عامل‌ها، سه عامل با بالاترین امتیاز را انتخاب کرده (اولویت با عامل‌هایی است که در دامنه مشترک با رخداد جاری هستند) و گزارش را به آنها ارسال می‌کند (سطر ۲ و۳ شکل ۵). این کار باعث می شود فقط عامل‌هایی که کارا هستند انتخاب شوند تا فرآیند تشخیص ناهنجاری دقیق‌تر و در زمان کمتری صورت گیرد. نحوه به‌روزرسانی امتیاز عامل‌ها مشابه الگوریتم پایه است. با گذشت زمان، امتیاز عامل‌ها با شیب کم کاهش می‌یابد. بنابراین اگر عاملی برای مدت طولانی در فرایند پایش شرکت نکرده باشد امتیاز عملکرد پایینی خواهد داشت. از سوی دیگر، امتیاز عملکرد پایین شانس انتخاب شدن عامل در فرآیند پایش را کاهش می‌دهد. زمانی که عامل برای مدت طولانی در فرآیندهای پایش شرکت ننماید در عمل حذف‌شده است.

Algorithm: Self-selected group Algorithm

Inputs

$R_{global} = \{R_{local}, \ldots, R_{local}\}$ //a set of local reports
$R_{local} = \{e_{id}, ts, g_{id}, (f, v), p\}$
$g_{req} \subseteq G$ //agents whose conditions are satisfied and requested R_{global}
$g_{sel} \subseteq G$ //selected agents to receive R_{global}
$a \in (0,1)$ //a threshold for marking event as malicious
$Num_{malicious} = 0$
$Num_{innocouoc} = 0$

1: **Procedure: Agent Selection**
2: For $g \in g_{req}$ do
3: Return threeHighest $((g,g_{score}) \in g_{req})$ as g_{se}

4: **Procedure: Final decision**
5: For R_{local} in R_{global} do
6: if $p \geq a$ then //the event is malicious in local report
7: $Num_{malicious}$ ++
8: else
9: $Num_{innocouoc}$ ++
10: if $Num_{malicious} \geq Num_{innocouoc}$ then
11: Return <e_{id}, decision =malicious>
12: else
13: Return <e_{id}, decision =innocuous>
14: For agent in g_{se} do
15: If decision =malicious:
16: $g_{sore} = g_{sore} + p$
17: update(profile)
18: else
19: $g_{sore} = g_{sore} + (1-p)$
20: update(profile)

شکل ۵: الگوریتم انتخاب هوشمند گروه

در این سامانه برای ضعیف قلمداد کردن یک عامل با توجه به امتیاز آن نیازی به تعریف حد آستانه نیست. زیرا امتیاز عملکرد برای مشارکت در فرآیند پایش صرفاً با مقایسه امتیاز عامل‌ها موردا ستفاده قرار می‌گیرد. عملکرد ضعیف عامل‌ها ممکن است به دلایل مختلفی مانند نامناسب بودن روش تشخیص ناهنجاری، تنظیم نادرست پارامترهای عامل و دچار نقص شدن آن باشد. برچسب نهایی یک رخداد با روش اکثریت آراء بین سه عامل انتخاب‌شده صورت می‌گیرد. روش دیگر این است که شرط تصمیم‌گیری نهایی، مشابه الگوریتم پایه،

جمع‌آوری شواهد به‌اندازه کافی باشد. همچنین اگر دامنه‌ای از شبکه مورد تهاجم‌های پی‌درپی قرار گیرد عامل‌های بیشتری به تحلیل آن رخداد خواهند پرداخت و یا اگر دامنه‌ای از شبکه حذف شود با کاهش امتیاز عملکرد عامل‌های آن با توجه به تعریف یک حد آستانه، به‌مرور حذف خواهند شد و منابع زیر ساخت آزاد خواهند گردید. نهایتاً بعد از این‌که طبقه رخداد مشخص گردید، نتیجه به اطلاع تمام عامل‌های مشارکت‌کننده در فرآیند خواهد رسید تا عامل‌ها امتیاز خود را به‌روزرسانی نمایند (سطر ۱۰ تا ۲۰ شکل ۵). همچنین عامل نمایه دامنه خود را به‌روزرسانی می‌نماید تا در مراحل بعدی فرآیند بررسی ناهنجاری لحاظ گردد. یعنی وقتی عاملی با احتمال بالا رخدادی را ناهنجار تشخیص می‌دهد اما این رخداد توسط سامانه هنجار تشخیص داده می‌شود رخداد مذکور از دید عامل غیرمعمول بوده است.

با به‌روزرسانی نمایه توسط عامل در دفعات بعدی مواجهه با این رخداد، عامل تجربیات گذشته (این تجربیات در نمایه ذخیره می‌گردد) را در تصمیم‌گیری دخالت می‌دهد. این فرآیند باعث کاهش نرخ هشدارهای کاذب می‌گردد. همچنین اگر رخدادی توسط عامل هنجار تشخیص داده شود اما برچسب‌نهایی این رخداد ناهنجار باشد، عامل نمایه را به‌روزرسانی می‌نماید تا با تغییر در سیاست‌های شبکه و غیره سازگار گردد.

الگوریتم انتخاب هوشمند گروه همچنین این امکان را فراهم می کند که عامل‌ها در گروهی که بهترین عملکرد را دارند فعالیت نمایند. این امر سبب افزایش کارایی عامل‌ها می‌شود چراکه آنها از دامنه فعالیت خود اطلاعات دقیق‌تری خواهند داشت که درنتیجه آن، ناهنجاری‌ها دقیق‌تر تشخیص داده خواهد شد. در این صورت تابع همبستگی نیز مقدار دقیق‌تری را محاسبه خواهد کرد زیرا گزارش‌های تبادل شده بین عامل‌ها عموماً در یک دامنه اتفاق افتاده‌اند.

۳-۴-۳- الگوریتم تصمیم‌گیری وزن‌دار هوشمند

حمله‌ها و نفوذها در شبکه عموماً در دامنه خاصی از شبکه رخ می دهند. براین اساس، در مواجهه با یک رخداد مخرب ممکن است عموم عامل‌هایی که در سایر دامنه‌ها قرار دارند رأی بر هنجار بودن آن بدهند. به‌منظور پیشگیری از این مشکل و کاهش نرخ منفی کاذب الگوریتم تصمیم‌گیری وزن‌دار هوشمند ارائه شده است (شکل ۶). عامل‌هایی که در یک دامنه قرار دارند و دقیقاً پشت سر هم وارد فرآیند پایش می شوند و گزارش محلی تولید می‌کنند را همسایه در نظر می‌گیریم.

این الگوریتم می‌تواند تعدادی عامل که در یک دامنه و در همسایگی یکدیگر هستند و تمام آنها رأی بر مخرب بودن یک رخداد دارند (به آنها سری متوالی می‌گوییم) را در گزارش نهایی تشخیص دهد (سطر ۴ و ۵ شکل ۶). در الگوریتم‌های پیشین نتیجه تصمیم هر عامل با وزن ۱ روی تصمیم نهایی تأثیرگذار بوده است. در این الگوریتم هرچقدر طول سری متوالی عامل‌ها بیشتر شود ۱ واحد به وزن تصمیم عامل اضافه می‌گردد (سطر ۷ شکل ۶). به‌محض اینکه سری شکسته شود وزن تصمیم به مقدار پیش‌فرض ۱ تنظیم می‌گردد (سطر ۱۱ و ۱۴ شکل ۶).

Algorithm: Smart Series-weighting algorithm

Inputs

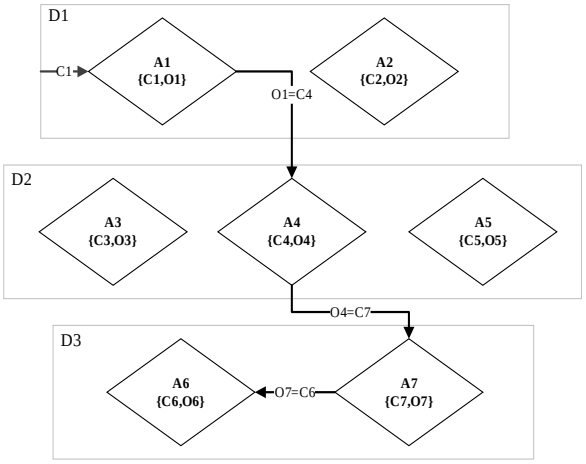
R_{global} //a global report
$a \in (0,1)$ //a threshold for marking event as malicious
$w = 1$ //initial weight
$Num_{malicious} = 0$
$Num_{innocouoc} = 0$
Domain =0 //domain of agent

1: **Procedure: Final decision**
2: For $R_{global} = \{R_{local}, \ldots, R_{local}\}$ do
3: For $R_{local} = <e_{id}, ts, g_{id,} (f, v), p) \in R_{global}$ do
4: if $p \geq a$ then //the event is malicious in local eport
5: if gid. domain == Domain then
6: $Num_{malicious} + w$
7: $W = w+1$
8: $Domain = g_{id}$. domain
9: else
10: $Num_{malicious}$++
11: $W = 1$ //the series is broken
12: $Domain = g_{id}$. domain
13: else //the event is innocuous in local reports
14: $W = 1$ //the series is broken
15: Domain = 0
16: $Num_{innocouoc}$ ++
17: if $Num_{malicious} \geq Num_{innocouoc}$ then
18: Return <e_{id}, decision =malicious>
19: else
20: Return <e_{id}, decision =innocuous>
21: For agent in g_{se} do
22: If decision =malicious:
23: $g_{sore} = g_{sore} + p$
24: update(profile)
25: else
26: $g_{sore} = g_{sore} + (1-p)$
27: update(profile)

شکل ۶: الگوریتم انتخاب هوشمند گروه

در این الگوریتم هرچه تعداد عامل‌های همسایه که نظر بر مخرب بودن یک رخداد دارند بیشتر شود وزن تصمیم آنها و درنتیجه اثرگذاری آنها بر نتیجه نهایی بیشتر خواهد شد. علت صعودی بودن وزن‌دهی به این علت است که عامل‌های جدیدتر سری با استفاده از اطلاعات بیشتری تصمیم‌گیری نموده‌اند و احتمال اینکه تصمیم آنها دقیق‌تر باشد بیشتر است. شایان‌ذکر است در حالت عادی تصمیم هر عامل وزن ۱ دارد. همچنین کمترین مقدار وزن تأثیر تصمیم برابر ۱ است.

به عنوان مثال ۷ عامل را در نظر بگیرید که به منظور پایش ترافیک شبکه در ۳ دامنه پخش شده‌اند (شکل ۷). همانطور که در شکل مشاهده می‌شود، با تحقق رخداد C1 عامل A1 فعال شده و پس از جمع‌آوری اطلاعات و تولید خروجی عامل A4 فعال شده و وارد فرآیند پایش می‌شود. به همین ترتیب در ادامه عامل A7 و A6 نیز وارد فرآیند پایش می‌شوند. در نهایت پس از جمع‌آوری اطلاعات به حد کفایت و تولید گزارش نهایی، الگوریتم تصمیم‌گیری نهایی اجرا شده و هنجار بودن و یا نبودن ترافیک مشخص می‌گردد.



شکل ۷: مثالی از روند پایش شبکه توسط ۷ عامل در ۳ دامنه

۴- پیاده‌سازی و تحلیل نتایج

همان‌طور که گفته شد در این پژوهش یک سامانه توزیع‌شده خود سازمانده مبتنی بر سیستم‌های چندعامله ارائه شده است. برای آزمودن این سامانه ابتدا شبیه ساز این سامانه پیاده سازی شده است. شبیه ساز پیاده سازی شده از طریق [۱۷] قابل‌دسترس است. از این شبیه‌ساز برای مدل‌سازی داده‌ها، پیاده‌سازی کلاس‌های عامل، الگوریتم های پیشنهادی و نوع داده‌های مختص این پژوهش مثل ناهنجاری استفاده‌شده است.

۴-۱- مدل‌سازی داده‌ها و شبیه‌سازی

برای بررسی و تحلیل عملکرد سامانه پیشنهادی مجموعه داده آموزشی NSL_KDD به کار گرفته‌شده است. به گزارش [۱۸] که مرجعی معتبر برای انواع مجموعه داده‌ها برای سامانه‌های تشخیص نفوذ است، این مجموعه از معیارهای مؤثر ارزیابی در این حوزه است. برای سنجش سامانه داده‌های +KDDTrain که دارای ۱۲۵۹۷۴ رکورد شامل ۶۷۳۴۳ داده نرمال و ۵۸۶۳۱ داده حمله است، استفاده‌شده اند. در گام اول با استفاده از نرم‌افزار وکا [۱۹] تعداد ۶ عدد از مؤثرترین ویژگی های مجموعه‌داده انتخاب و به مقادیر عددی تبدیل شدند (جدول ۱).

جدول ۱: ویژگی‌های منتخب مجموعه داده +KDDTrain

ردیف	شماره ویژگی	نام ویژگی
۱	۴	flag
۲	۵	Src_bytes
۳	۶	Dst_bytes
۴	۱۲	Logged_in
۵	۲۶	Srv_serror_rate
۶	۳۰	Diff_srv_rate

همان‌طورکه پیش‌تر ذکر شد عامل‌های این سامانه می‌توانند از روش‌های مختلفی برای رده‌بندی رخدادهای شبکه استفاده نمایند. ازاین‌رو به‌منظور سهولت در بررسی عملکرد عامل‌های سامانه و مقایسه الگوریتم‌های مختلفی که در این محیط توسعه داده می‌شوند رخدادهای ناهنجار در یک قالب از مجموعه داده +KDDTrain مدل‌سازی شده‌اند. قالب رخدادها به‌شکل $<e_{id}, type, detectability, validity>$ هستند که در آن، e_{id} شناسه رخداد است که به صورت تصادفی و یکتا تولید می‌شود. رکوردهای مجموعه‌داده با توجه به برچسب آنها (هنجار و یا ناهنجار) و نیز نوع ناهنجاری و میزان سختی تشخیص به ۲۴۷ خوشه تقسیم‌شده‌اند. متغیر *type* نشان می‌دهد که هر رخداد عضو کدام خوشه است. متغیر *detectability* با توجه به فاصله هر رخداد از مرکز خوشه متناظر و نیز برچسب میزان سختی تشخیص آن محاسبه شده است. مرکز خوشه‌ها میانگین مقادیر ویژگی‌های اعضای خوشه‌ها هستند. هرچقدر مقدار متغیر *detectability* کمتر باشد به معنی وجود حمله‌ای است که تشخیص آن دشوار است. حملات روز صفر را می‌توان نوعی از این حملات دانست که به علت ناشناخته بودن به‌سختی قابل‌تشخیص هستند. متغیر *validity* میزان ناهنجاری یک رخداد را نشان می‌دهد. به‌منظور مدل سازی دقیق‌تر رخدادها و به‌تبع آن تحلیل بهتر آنها به جای نگاه دودویی به هنجار بودن و یا نبودن، از رویکرد فازی برای محاسبه میزان ناهنجاری استفاده‌شده است و میزان ناهنجاری یک رخداد در بازه [۰،۱] تعیین‌شده است (۱ برای ناهنجاری و ۰ برای نرمال). این متغیر با توجه به برچسب رکورد و نیز فاصله رکورد از مرکز خوشه‌ای که به آن تعلق دارد، محاسبه شده‌است.

تابع تصمیم‌گیری در عامل با توجه به دو متغیر *detectability* و *validity* میزان هنجار بودن رخداد را مشخص می‌کند. در گام اول ابتدا دو عدد تصادفی با توزیع نرمال و با میانگین مقادیر *detectability* و *validity* تولید می‌گردد. واریانس این دو توزیع باهم برابر است و به صورت ایستا تعیین می‌گردد. سپس عدد خروجی تابع تصمیم‌گیری بر اساس میانگین این دو عدد محاسبه می‌شود.

با تغییر واریانس توزیع‌های نرمال می‌توان انواع روش‌های تشخیص ناهنجاری را نیز مدل‌سازی نمود. به‌طور مثال هر چه واریانس توزیع کمتر باشد به این معناست که از روش تشخیص دقیق‌تری در عامل‌ها استفاده شده‌است. مزیت مدل سازی ارائه شده این است که اولاً نمای کلی رخدادهای مدل‌سازی شده واقعی است و تفسیر دقیق‌تری نسبت به رخدادهای صرفاً تصادفی و انتزاعی ارائه می‌دهد. ثانیاً این امکان را فراهم می‌کند تا سنجش سامانه مستقل از روش تشخیص ناهنجاری به کارگرفته شده در عامل باشد. درنتیجه می‌توان بر توسعه الگوریتم‌هایی که هوشمندی ارتباط عامل‌ها و کارایی آنها را افزایش می‌دهد که هدف اصلی این پژوهش است تمرکز کرد.

۴-۲- پیاده‌سازی

سامانه پیشنهادی با زبان پایتون، سیستمی با پردازنده corei5-CPU و حافظه RAM 8G پیاده‌سازی شده است. این سامانه از چهار کلاس اصلی مدل‌ساز داده‌ها، نوع داده ناهنجاری، عامل و شبیه‌ساز تشکیل‌شده است. جدول ۲ متغیرهای مربوط به کلاس عامل را نشان می‌دهد که به‌منظور تنظیم عملکرد عامل و به‌صورت ایستا تعیین می‌گردند.

جدول ۲: متغیرهای تنظیم عامل

متغیر	توضیح	مقدار
buffer_size	مشخص‌کننده تعداد رخدادهایی که در حافظه عامل نگهداری می‌شوند	۲۰
threshold_analysis	حد آستانه مربوط به ناهنجار قلمداد کردن یک رخداد	۰/۶
threshold_ts	مقدار اختلاف زمانی تأثیرگذار بر میزان همبستگی رخدادها	۱۰
variance	واریانس توزیع‌های نرمال مورداستفاده در تابع تصمیم‌گیری را مشخص می‌نماید	۰/۲
profile_size	مقدار اطلاعات ذخیره‌شده در نمایه را نشان می‌دهد	۲۰

جدول ۳ متغیرهای مربوط به کلاس شبیه سازی را نشان می‌دهد. این متغیرها نحوه شبیه‌سازی و مدل‌سازی شبکه را نشان می‌دهند. در جدول ۳ متغیر num_of_domain تعداد دامنه‌های موجود در شبکه را نشان می‌دهد. هرچقدر تعداد دامنه در شبکه بیشتر باشد به معنای پیچیده‌تر شدن شبکه و حضور فنّاوری‌های مختلف در آن است.

جدول ۳: متغیرهای تنظیم شبیه‌ساز

متغیر	توضیح	مقدار
Num_of_itration	تعداد شبیه‌سازی‌های انجام‌شده	۱۰۰۰
num_of_agents	تعداد عامل‌های مستقر در شبکه	[۵۰،۱۰۰]
num_of_condition	تعداد اعضای مجموعه شرایط هر عامل	{۱،۲،۳}
num_of_effect	تعداد اعضای مجموعه خروجی هر عامل	{۱،۲،۳}
num_of_domain	تعداد دامنه‌های موجود در شبکه	[۲۰،۵۰]
agent_domain_range	محدوده IP های عامل‌ها	[۲۰،۵۰]
time_stamp_range	محدوده زمان	(۰،۱۰۰)
global_decision_threshold	حد آستانه کفایت شواهد برای تصمیم‌گیری نهایی	۱۰
domain_association_factor	میزان ارتباط بین دامنه-ها	[۰،۱]
def_w	مقدار پیش‌فرض وزن تصمیم عامل‌ها	۱

متغیر domain_assosiation_factor میزان ارتباط منطقی بین دامنه‌ها را کنترل می‌نماید. در این متغیر مقدار صفر به معنی این است که تمام دامنه‌ها مستقل‌اند. در این حالت عامل‌ها فقط با عامل‌های هم دامنه خود ارتباط دارند. افزایش مقدار این متغیر به معنای وجود ارتباط بین دامنه‌های مختلف شبکه است. با این متغیر می‌توان پیچیدگی‌های مختلف شبکه و نیز تغییرات آن را مدل‌سازی نمود.

۴-۳- نتایج

هدف اصلی این پژوهش فراهم نمودن ویژگی مقیاس پذیری برای سیستم‌های تشخیص نفوذ مبتنی بر ناهنجاری است. به‌منظور تحلیل عملکرد الگوریتم‌های پیشنهادی، ناهنجاری‌های مدل‌سازی شده از مجموعه داده KDDTrain+ به سامانه تزریق‌شده است. برای هر رخداد ممکن است یکی از چهار حالت زیر رخ دهد:

- مثبت صحیح (TP): رخدادهای ناهنجار که سامانه آنها را به‌درستی تشخیص می‌دهد.
- منفی صحیح (TN): رخدادهای نرمال که سامانه آنها را به‌درستی تشخیص می‌دهد.
- مثبت کاذب (FP): رخدادهای نرمال که سامانه آنها را به‌اشتباه ناهنجاری تشخیص می‌دهد.
- منفی کاذب (FN): رخدادهای ناهنجار که سامانه آنها را به‌اشتباه نرمال تشخیص می‌دهد.

برای ارزیابی الگوریتم‌ها از معیارهای نرخ‌تشخیص‌ناهنجاری (DR)، نرخ‌مثبت‌کاذب (FRR) و دقت‌کل (AC) که به ترتیب با رابطه های (۹)، (۱۰) و (۱۱) محاسبه می‌شوند، استفاده‌شده‌است. دقت‌کل برابراست با نرخ نمونه‌هایی که به درستی برچسب‌گذاری شده اند.

$$DR = \frac{TP}{TP + FN} \qquad (۹)$$

$$FPR = \frac{FP}{FP + TN} \qquad (۱۰)$$

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \qquad (۱۱)$$

همچنین برای ارزیابی زمان تصمیم‌گیری در رابطه با برچسب یک رخداد، تعداد تصمیم‌های محلی اتخاذشده سنجیده‌شده‌است. جدول ۴ ماتریس ابهام حاصل از شبیه‌سازی انجام‌شده را نشان می‌دهد.

جدول ۴: ماتریس ابهام شبیه‌سازی

الگوریتم	TP	TN	FP	FN
پایه	۴۱۶۲۸	۵۹۹۳۶	۷۴۰۷	۱۷۰۰۳
تصمیم‌گیری وزن‌دار هوشمند	۵۳۳۵۴	۵۵۹۰۱	۱۱۴۴۲	۵۲۷۷
انتخاب هوشمند گروه	۵۰۴۲۲	۵۸۵۸۹	۸۷۵۴	۸۲۰۹

جدول ۵ مقدار معیارهای ارزیابی نرخ‌تشخیص، نرخ‌مثبت‌کاذب و دقت‌کل را برای سه الگوریتم پایه، انتخاب هوشمندگروه و تصمیم‌گیری وزن‌دار هوشمند نشان می‌دهد. در الگوریتم تصمیم‌گیری وزن‌دار هوشمند درواقع با مدنظر قرار دادن محل فعالیت عامل و در نظرگرفتن وزن بیشتر برای رأی عامل‌هایی که در دامنه رخداد مشکوک هستند نرخ تشخیص ناهنجاری در این الگوریتم در مقایسه با الگوریتم پایه افزایش‌یافته است. در الگوریتم هوشمندگروه نیز با توجه به‌اینکه تعداد محدودی از عامل‌هایی که بهترین عملکرد را در دامنه رخداد مشکوک داشته‌اند انتخاب‌شده است نرخ تشخیص افزایش‌یافته است. ستون سوم در جدول ۵، مقدار پارامتر نرخ مثبت‌کاذب را برای سه الگوریتم پایه،

انتخاب هوشمندگروه و تصمیم‌گیری وزن‌دار هوشمند نشان می‌دهد. در الگوریتم‌های انتخاب هوشمندگروه و تصمیم‌گیری وزن‌دار هوشمند نرخ مثبت‌کاذب در مقایسه با الگوریتم پایه اندکی افزایش‌یافته است. بنابر [۲۷] به‌طورکلی به علت نیاز به حفظ زیرساخت شبکه از حملات و تهدیدات مختلف، پارامتر نرخ‌تشخیص از اهمیت بیشتری در مقایسه با نرخ هشدارهای کاذب برخوردار است.

جدول ۵: نتایج الگوریتم‌های توسعه داده‌شده

الگوریتم	DR	FP	AC%
پایه	۰/۷۱	۰/۱۱	۸۱
تصمیم‌گیری وزن‌دار هوشمند	۰/۹۱	۰/۱۷	۸۸
انتخاب هوشمند گروه	۰/۸۶	۰/۱۳	۸۶

در جدول ۶ نتایج حاصل از شبیه‌سازی تعدای از سامانه‌های تشخیص نفوذ مطرح اخیر آورده شده‌است. نتایج حاصل از آزمایش‌های انجام شده توسط این سامانه‌ها روی مجموعه داده +KDDTrain در جدول گزارش شده‌ا ست. م شخص‌ا ست که میزان دقت‌کل در سامانه توزیع‌شده چندعامله پیشنهادی این مقاله با در نظر گرفتن الگوریتم ت صمیم‌گیری وزن‌دار هو شمند در مقای سه با این سامانه‌ها بهبود یافته ا ست. همچنین با در نظر گرفتن الگوریتم انتخاب هو شمندگروه، میزان دقت‌کل در مقای سه با سامانه‌های عنوان شده بهبود یافته‌ا ست. لازم به ذکر ا ست که اکثر سامانه‌های مبتنی بر سی ستم‌های چندعامله که در مرور ادبیات بررسی شده‌اند، از پایگاه‌داده‌های رایج برای ارزیابی کار خود استفاده نکرده اند. بنابراین روش پیشنهادی با به‌روزترین تحقیقات اخیر مقایسه شده‌است.

جدول ۶: میزان دقت‌کل برای روش‌های مطرح اخیر

روش	سال	AC%
روش پیشنهادی	۲۰۲۰	۸۸/۰۰
مبتنی بر یادگیری عمیق [۲۰]	۲۰۱۷	۸۳/۲۸
مبتنی بر یادگیری نیمه‌نظارتی فازی [۲۱]	۲۰۱۷	۸۴/۱۲
مدل حافظه کوتاه مدت - بلندمدت دوطرفه [۲۲]	۲۰۲۰	۸۴/۲۵
مبتنی بر یادگیری نیمه‌نظارتی فازی [۲۳]	۲۰۱۸	۸۴/۵۴
مبتنی بر یادگیری عمیق [۲۴]	۲۰۱۸	۸۵/۰۰
انتخاب ویژگی و الگوریتمهای هوش جمعی [۲۵]	۲۰۱۹	۸۵/۸۰
مبتنی بر درخت تصمیم گیری [۲۶]	۲۰۱۹	۸۵/۸۱
سیستم تشخیص نفوذ سه لایه [۱۶]	۲۰۱۹	۸۶/۷۵

۴-۴- پیچیدگی زمانی

از بین مقالات مقایسه‌شده با تحقیق حاضر پژوهش‌های[۲۰]، [۲۴]، [۲۵] و [۲۶] زمانی برای آموزش مدل و تحقیق [۲۱] زمان آزمایش داده‌ها را ارائه‌داده اند. لازم به‌ذکر است که زمان‌های گزارش‌شده با استفاده از پردازنده‌های متفاوتی بدست آمده‌اند و امکان مقایسه بین آنها از این طریق ممکن نیست. در ادامه به تحلیل پیچیدگی زمانی سامانه پی شنهادی می پردازیم. زمان تحلیل رخدادها در این سامانه از سه قسمت به‌صورت رابطه (۱۲) تشکیل‌شده است:

$$T = T_C + T_D + T_U \quad (۱۲)$$

وقتی رخدادی در شبکه اتفاق می‌افتد عامل‌ها شروع به تحلیل آن و جمع‌آوری اطلا عات می‌نمایند که زمان این تحلیل و جمع‌آوری اطلاعات T_C نامیده می‌شود. بعد از جمع‌آوری اطلاعات به‌حدکفایت، باید الگوریتم تصمیم‌گیری نهایی اجرا شود که T_D این زمان را نشان می‌دهد. بعد از اجرای الگوریتم تصمیم‌گیری نهایی باید نتیجه به‌اطلاع عامل‌های مشارکت‌کننده برسد تا آن‌ها بتوانند خود را به‌روزرسانی نمایند که T_U ن شانگر این زمان ا ست. پیچیدگی هرکدام از این زمان‌ها با رابطه های (۱۳) تا (۱۵) نشان داده‌شده‌است:

$$T_C = O(n\ d_1) \quad (۱۳)$$

$$T_D = O(n\ d_2) \quad (۱۴)$$

$$T_U = O(n\ d_3) \quad (۱۵)$$

در رابطه های فوق، n تعداد عامل‌های مشارکت‌کننده در فرآیند پایش رخداد است. d_1 متوسط زمان جمع‌آوری اطلاعات و تحلیل آن توسط هر عامل، d_2 متوسط زمان بررسی هر گزارش محلی در الگوریتم تصمیم‌گیری نهایی و d_3 متوسط زمان به‌روزرسانی هر عامل است. همان‌طور که از پیچیدگی‌های زمانی مشخص است مدت‌زمان تحلیل رویدادها به تعداد عامل‌های مشارکت‌کننده و یا همان تعداد تصمیم های محلی اتخاذشده بستگی دارد. واضح است که:

$$d_1 > d_3 > d_2 \quad (۱۶)$$

نمودار شکل ۸ مقدار پارامتر نرخ تعداد تصمیم‌های محلی که متناسب با زمان اجرا است را برای الگوریتم پایه، الگوریتم انتخاب هو شمندگروه و الگوریتم ت صمیم‌گیری وزن‌دار هو شمند نشان می‌دهد. تعداد ت صمیم‌های محلی اتخاذ شده تو سط عامل‌ها رابطه م ستقیمی با مدت‌زمان اجرای برنامه دارد. تعداد عامل‌های م شارکت‌کننده در فرآیند پایش بر اساس رابطه (۸) تعیین می‌گردد.

شکل ۸: مقایسه نرخ تعداد تصمیم‌های محلی

الگوریتم انتخاب هوشمندگروه با انتخاب عامل‌های کارا سعی در بهبود متغیر p_i دارد. هرچقدر این متغیر که خروجی تابع تصمیم‌گیری است قطعی‌تر باشد (به این معنا که به صفر یا یک نزدیک‌تر باشد) مقدار *certainty* زودتر به‌حدکفایت از پیش تعیین‌شده خواهد رسید. الگوریتم تصمیم‌گیری وزن‌دار هوشمند نیز با افزایش مقدار w_i سعی دارد متغیر *certainty* را زودتر به‌حدکفایت برساند. همان‌طور که در شکل ۸ م شخص ا ست نرخ تعداد ت صمیم‌های محلی در الگوریتم‌های

تصمیم‌گیری وزن‌دار هوشمند و انتخاب هوشمندگروه در مقایسه با الگوریتم پایه کاهش‌یافته‌است. به‌عبارت‌دیگر زمان تحلیل رخدادها در این دو الگوریتم کاهش‌یافته‌است. این امر به‌این دلیل است که الگوریتم انتخاب هوشمندگروه از عامل‌های کاراتری استفاده می‌نماید و در زمان کمتری شواهد کافی جمع‌آوری می‌کند. در الگوریتم تصمیم‌گیری وزن دار هوشمند نیز به علت وزن بیشتر تصمیم عامل‌های همسایه نیز شواهد جمع‌آوری‌شده زودتر به‌حدکفایت می‌رسد.

۴-۵- تحلیل کارایی

سامانه ارائه‌شده در این مقاله علاوه بر هدف بهبود نرخ‌تشخیص ناهنجاری در ترافیک شبکه سعی دارد ویژگی مقیاس‌پذیری و همچنین جلوگیری از بروز خطای نقطه‌شکست‌مرکزی را نیز فراهم کند. همانطور که پیش‌تر اشاره شد از یک سو، وجود قابلیت حذف و اضافه شدن هوشمند عامل‌ها به فرآیند پایش ترافیک شبکه، که بنابر تغییرات و نیازهای شبکه رخ می‌دهد، ویژگی مقیاس‌پذیری را فراهم آورده است و این سامانه بدون نیاز به یک فرد خبره می‌تواند خود را با تغییرات شبکه سازگار نماید. از سوی دیگر نحوه ارتباط عامل‌ها و فعال شدن آن‌ها نیز قابلیت مقیاس‌پذیری را پشتیبانی می‌کنند. علی‌رغم اهمیت مقیاس‌پذیری سامانه‌های تشخیص‌نفوذ، در کارهای انجام شده در این حوزه توجه کمی به این ویژگی معطوف بوده است (مقالات عنوان شده در جدول ۶ به این موضوع نپرداخته اند).

در سامانه توزیع شده پیشنهادی، عامل‌ها ساختار مشابه داشته و هریک قابلیت جمع‌آوری داده، تحلیل داده‌ها و تصمیم‌گیری نهایی دارند و اگر یکی از عامل‌ها به هر دلیلی از کار بیافتد، عامل‌های دیگر می‌توانند به فرآیند پایش ادامه دهند. وجود این ویژگی از بروز خطای نقطه‌شکست‌مرکزی می‌کاهد. با توجه به اینکه پایش یک امر ۲۴×۷ است جلوگیری از وقوع خطای نقطه‌شکست‌مرکزی با کمترین هزینه نیازمند توجه است. که این موضوع نیز کمتر در مقالات مورد برر سی قرار گرفته است (مقالات مورد مقایسه به این موضوع نپرداخته اند).

بدیهی است که هدف اصلی این سامانه بهبود نرخ‌تشخیص است. در مقالات مختلف سعی شده‌است که با بهره‌گیری از روش‌های متنوعی مانند روش‌های یادگیری ماشین و یا روش‌های هیوریستیک و غیره مدلی ساخته شود و با داده‌های آموزشی، آموزش داده شود تا در نهایت بتواند نرخ تشخیص را بهبود دهد. ماهیت معماری پیشنهادی این مقاله و همچنین ساختار عامل‌های به‌کارگرفته‌شده در آن به‌این صورت است که می‌تواند از روش‌های مختلفی برای تشخیص ناهنجاری در ترافیک شبکه در ماژول تصمیم‌گیری عامل‌ها استفاده نمود. به عبارت دیگر عامل های به‌کارگرفته‌شده در این سامانه می‌توانند هر نوع روش تشخیص ناهنجاری را با استفاده از رویکرد بررسی داده‌های بیشتر و به کارگیری عامل‌های بیشتر در تحلیل بهبود دهند. عامل‌ها فراخور نیاز با یکدیگر همکاری می‌کنند و هریک از زاویه دید خود و نیز با استفاده از اطلاعات محلی خود، رخداد را پایش کرده و نظر خود را با استفاده از ماژول تصمیم‌گیری اعلام کرده و نهایتا پس از اجرای الگوریتم تصمیم‌گیری نهایی هنجار بودن و یا نبودن رخداد مشخص می‌شود. همچنین عامل‌ها از محیط مورد پایش خود یک نمایه می سازند که این نمایه به ماژول تصمیم‌گیری در اخذ تصمیم دقیق‌تر کمک می‌نماید. بدیهی است که استفاده از چند عامل و به تبع آن استفاده از داده‌های بیشتر که در نهایت نوع رخداد را بوسیله الگوریتم رای‌اکثریت بین نظرات عامل‌ها مشخص می‌کند کاراتر از یک روش متمرکز است؛ هرچند این امر هزینه زمانی بیشتری را می‌طلبد. البته سعی ما بر این بوده است که با گروه‌بندی هوشمند و وزن‌دهی هوشمند به تصمیم عامل‌ها، زمان را حتی الامکان کاهش دهیم.

با توجه به این توضیحات می‌توان ادعا کرد که معماری پیشنهادی این مقاله نسبت به روش‌های موجود بهبود داشته است و نتایج پیاده‌سازی نیز گواه این موضوع است.

۵- ۵- نتیجه و کارهای آینده

سامانه پیشنهادی ارائه‌شده قادر است ضمن فراهم کردن ویژگی مقیاس پذیری، از روش‌های موجود تشخیص ناهنجاری بهره برده و عملکرد آنها را بهبود دهد. در سامانه مذکور عامل‌ها می‌توانند بدون پیش شرط وارد سامانه شده و شروع به فعالیت نمایند و با سایر عامل‌ها به تعامل بپردازند. قابلیت اضافه و حذف شدن هوشمند عامل‌ها ویژگی مقیاس پذیری را در سامانه فراهم نموده است. با این روش سامانه تشخیص نفوذ قادر است خود را با تغییرات شبکه سازگار نماید. در عموم سامانه های تشخیص نفوذ از یک پردازشگر مرکزی استفاده می شود که عامل ها عموماً وظیفه جمع‌آوری اطلاعات و ارسال آن به پرداز شگر را دارند. در سامانه پیشنهادی این مقاله عامل‌ها می‌توانند ضمن جمع‌آوری اطلاعات به تحلیل آنها نیز بپردازند. وجود ویژگی توزیع‌شدگی از بروز نقطه شکست‌مرکزی جلوگیری می‌نماید. وجود فاکتور امتیاز عملکرد در عامل‌ها باعث می‌شود که هر عامل در دامنه فعالیت خود تخصصی‌تر عمل نماید. همچنین عامل‌هایی که امتیاز عملکرد پایین دارند به‌مرور از سامانه حذف‌شده و منابع تحت اختیار مثل حافظه و CPU را عملاً آزاد می‌نمایند. وجود چنین قابلیت‌هایی باعث می‌شود که این سامانه، خود سازمانده باشد.

به‌منظور بررسی رخدادهای ترافیک شبکه، در این سامانه تعدادی الگوریتم برای تصمیم‌گیری نهایی پیشنهاد شده است. این الگوریتم‌ها برچسب نهایی یک رخداد را با توجه به گزارش محلی تولیدشده توسط هر عامل تعیین می‌نمایند. ارزیابی صورت گرفته روی این الگوریتم‌ها نشان داده است که میزان نرخ تشخیص در مقایسه با سامانه تشخیص نفوذ سه‌لایه ارائه‌شده اخیر بهبودیافته است. نرخ مثبت کاذب اندکی افزایش یافته است که در مجموع میزان قابل قبولی دارد. شایان ذکر است که در صنعت، نرخ تشخیص از اهمیت بیشتری برخوردار است. همچنین ارزیابی صورت‌گرفته بر مقدار تحلیل مورد نیاز برای تعیین برچسب یک رخداد روی الگوریتم‌های پیشنهادی نشان داده است که

دو الگوریتم انتخاب هو شمندگروه و ت صمیم‌گیری وزن‌دار هو شمند در مقایسه با الگوریتم پایه عملکرد بهتری دارند.

در آ ینده به منظور بهبود نرخ هشـــدار های کاذب بر توســـعه الگوریتمی در سیستم چندعامله توزیع‌شده خودسازمانده تمرکز خواهد شـــد. این امر می‌تواند با تغییر در معماری عامل‌ها و یا الگوریتم‌های تصـــمیم‌گیری نهایی صـــورت گیرد. در این پژوهش تابع تشـــخیص ناهنجاری در عامل‌ها به صورت آماری مدل‌سازی شده‌است که در آینده می‌توان به کارگیری انواع روش‌های تشـــخیص ناهنجاری در عامل‌ها را مورد ارزیابی قرار داد.

مراجع

[1] G. Fernandes, J.J. Rodrigues, L.F. Carvalho, J.F. Al-Muhtadi and M.L. Proença, "A comprehensive survey on network anomaly detection", Telecommunication Systems, vol. 70, no. 3, pp. 447-489, 2019.

[2] A. Nisioti, A. Mylonas, P.D. Yoo and V. Katos, "From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods", IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3369-3388, 2018.

[3] M.H. Bhuyan, D.K. Bhattacharyya and J.K. Kalita, "Network anomaly detection: methods, systems and tools", IEEE communications surveys & tutorials, vol. 16, no. 1, pp. 303-336, 2013.

[4] P. Kendrick, N. Criado, A. Hussain and M. Randles, "A self-organising multi-agent system for decentralised forensic investigations", Expert Systems with Applications, vol. 102, pp. 12-26, 2018.

[5] V. Rothamsted, T. Lewis and V. Barnett, *Outliers in statistical data*, John Wiley & Sons, 1996.

[6] N. Hoque, M.H. Bhuyan, R.C. Baishya, D.K. Bhattacharyya and J.K. Kalita, "Network attacks: Taxonomy, tools and systems", Journal of Network and Computer Applications,vol. 40, pp. 307-324, 2014.

[7] *2020 Cyber Security Statistics The Ultimate List Of Stats, Data & Trends*, June 2020, https://purplesec.us/resources/cyber-security-statistics.

[8] M. Wooldridge, *an Introduction to MultiAgent Systems*. Second ed., John wiley & sons, 2009.

[9] P. Shakarian, G.I. Simari, G. Moores and S. Parsons, "Cyber attribution: An argumentation-based approach", Cyber Warfare, Springer, pp. 151-171, 2015.

[10] A.D. McKinnon, S.R. Thompson, R.A. Doroshchuk, G.A. Fink and E.W. Fulp, "Bio-inspired cyber security for smart grid deployments", IEEE PES Innovative Smart Grid Technologies Conference (ISGT), pp. 1-6, 2013.

[11] A. Jahanbin, A. Ghafarian, S.A. Hosseini Seno and S. Nikookar, "A computer forensics approach based on autonomous intelligent multi-agent system", International Journal of Database Theory and Application, vol. 6, no.5, pp. 1-12, 2013.

[12] K. Shanmugasundaram, N. Memon, A. Savant and H. Bronnimann, "ForNet: A distributed forensics network" International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, Springer, 2003.

[13] Z.A. Baig, "Multi-agent systems for protecting critical infrastructures: A survey", Journal of Network and Computer Applications, vol. 35, no. 3, pp. 1151-1161, 2012.

[14] R. Kesavamoorthy and K.R. Soundar, "Swarm intelligence based autonomous DDoS attack detection and defense using multi agent system", Cluster Computing, vol. 22, no. 4, pp. 9469-9476, 2019.

[15] W. Mees, "Multi-agent anomaly-based APT detection", Proceedings of Information Systems Technology Panel Symposium, 2012.

[۱۶] آمره‌ای، بیگی، «یک سیستم تشخیص نفوذ چندلایه با رویکرد ترکیبی»، پانزدهمین کنفرانس بین‌المللی انجمن رمز ایران، تهران، انجمن رمز ایران، ۱۳۹۷.

[17] *Distributed Self-organizing Multi-agent System*, July 2020, https://gitlab.com/N_shakiba/dsms/

[18] *NSL-KDD dataset*, June 2020, https://www.unb.ca/cic/datasets/nsl.html.

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten, "The WEKA data mining software: an update", ACM SIGKDD explorations newsletter, vol. 11, no. 1, pp. 10-18, 2009.

[20] C. Yin, Y. Zhu, J. Fei and X. He, "A deep learning approach for intrusion detection using recurrent neural networks" IEEE Access, vol. 5, pp. 21954-21961, 2017.

[21] R.A.R. Ashfaq, X.Z. Wang, J.Z. Huang, H. Abbas and Y.L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system", Information Sciences, vol. 378, pp. 484-497, 2017.

[22] T. Su, H. Sun, J. Zhu, S. Wang and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset". IEEE Access,vol. 8, pp. 29575-29585, 2020

[23] Y. Gao, Y. Liu, Y. Jin, J. Chen, and H. Wu, "A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system", IEEE Access, vol. 6, pp. 50927-50938, 2018.

[24] S. Naseer, Y. Saleem, S. Khalid, M.K. Bashir, J. Han, M.M. Iqbal and K. Han, "Enhanced network anomaly detection based on deep neural networks". IEEE Access. Vol 6, pp. 48231- 48246, 2018.

[25] B.A. Tama, M. Comuzzi and K.H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system". IEEE Access, vol. 7, pp. 94497-507, 2019.

[26] P. Illy, G. Kaddoum, C.M. Moreira, K. Kaur and S. Garg, "Securing fog-to-things environment using intrusion detection system based on ensemble learning". IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-7, 2019.

[27] K. Alsubhi, N. Bouabdallah and R. Boutaba, "Performance analysis in intrusion detection and prevention systems", IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops, 2011.

زیرنویس‌ها

[1] Integrity
[2] Confidentiality
[3] Intrusion detection system
[4] Host based
[5] Signature based
[6] Anomaly based
[7] Zero day attacks
[8] False positive
[9] Profile
[10] Single point of failure
[11] Clustering
[12] Classification
[13] Swarm intelligence
[14] Degradation
[15] Forensic
[16] Entropy
[17] Advanced persistent threat
[18] Severity level
[19] Module