تشخیص بات‌نت با استفاده از مدل مخفی مارکوف در وقفه‌های جریان

سارا السادات زمانی دانالو[۱]، کارشناسی ارشد؛ محسن افشارچی[۲]، دانشیار؛ وحید سلوک[۳]، استادیار

۱- دانشکده کامپیوتر و فناوری اطلاعات - دانشگاه تحصیلات تکمیلی علوم پایه زنجان- زنجان- ایران- szamani925@gmail.com
۲- دانشکده مهندسی برق و کامپیوتر- دانشگاه زنجان - زنجان- ایران- afsharchim@znu.ac.ir
۳- دانشکده مهندسی کامپیوتر و فناوری اطلاعات - دانشگاه صنعتی ارومیه - ارومیه - ایران- v.solouk@it.uut.ac.ir

چکیده: بات‌نتها یکی از محبوبترین انواع بدافزارها در میان مجرمان اینترنتی هستند، به‌طوریکه اخیراً پایه‌ی اصلی بیشتر جرائم سایبری بوده‌اند. اغلب روش‌های تشخیص بات‌نت موجود نمی‌توانند آنها را به‌صورت بلادرنگ و قبل از مشارکت در یک حمله سایبری، تشخیص دهند. در این مقاله یک سیستم تشخیص بات‌نت مبتنی‌بر مدل مخفی مارکوف ارائه می‌شود. این سیستم قادر به تشخیص بات‌نت در بازه‌های زمانی خیلی کوچک از جریان شبکه بدون نیاز به بررسی کل جریان است. همچنین این روش علاوه‌بر تشخیص بات‌نت در مراحل اولیه از چرخه حیات، مرحله فعالیت آن (کانال فرمان و کنترل یا حمله) را نیز در هر لحظه تعیین می‌کند. بات‌نت BlackEnergy یکی از خطرناک‌ترین انواع بات‌نتهای مبتنی‌بر HTTP است، که در این پژوهش ترافیک شبکه آن مورد تحلیل و بررسی قرار می‌گیرد. ویژگی‌های شاخص و الگوهای رفتاری این بات‌نت در مراحل مختلف چرخه حیاتش استخراج می‌شود. سپس مدل مخفی مارکوف پیشنهادی جهت تشخیص بات‌نت BlackEnergy براساس ویژگی‌ها و الگوهای رفتاری آن ارائه می‌شود. برای ارزیابی مدل ارائه‌شده، از مجموعه داده‌جامع و معتبری از ترافیک شبکه استفاده می‌شود که نشان می‌دهد روش پیشنهادی حتی در پنجره‌های زمانی خیلی کوچک، دقت تشخیص بالایی نسبت به بسیاری از روش‌های دیگر دارد.

واژه‌های کلیدی: تشخیص بات‌نت، مدل مخفی مارکوف، وقفه زمانی، جریان شبکه، مرحله فرمان و کنترل.

BotNet Detection Using Hidden Markov Model within Flow Intervals

Sara Sadat Zamani Danalou[1], MSc; Mohsen Afsharchi[2], Associate Professor; Vahid Solouk[3], Assistant Professor

1- Faculty of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences, Gava Zang, Zanjan, Iran, Email: szamani925@gmail.com
2- Faculty of Electrical and Computer Engineering, University of Zanjan, Zanjan, Iran, Email: afsharchim@znu.ac.ir
3- Department of Computer and Information Technology Engineering, Urmia University of Technology, Urmia, Iran, Email: v.solouk@it.uut.ac.ir

Abstract: Botnets are known to be among the most popular malwares in cyber criminals for their practicality in carrying many cyber-crimes as reported in the recent news. While many detection schemes have been developed, botnets remain the most powerful attack platform by constantly and continuously adopting new techniques and strategies. Thus, early identification and timely detection of botnets can take an effective step towards making perfect defense system. Most of existing botnet detection methods cannot detect botnets in real-time and in an early stage of their lifecycle before participating in a cyber-crime. In this work, we propose a novel approach to detect the BlackEnergy botnet tra□c using Hidden Markov Model (HMM) within flow Intervals. In BlackEnergy, bots are controlled by attackers under a HTTP base command and control (C&C) infrastructure. First we analysis BlackEnergy's network traffic and extract its main features and network behavior patterns. Then we adapt the proposed HMM model with BlackEnergy botnet patterns and features. In addition to detecting the botnet communication tra□c in both Attack and C&C stages, inferred HMM defines the stage of botnet lifecycle. Our proposed method detects botnet activity in small time intervals without having seen a complete network flow. Using existing datasets, we show experimentally that it is possible to identify the presence of botnets activity with high accuracy even in very small time windows.

Keywords: Botnet detection, Hidden Markov Model, time interval, flow interval, network flow, command and control stage.

تاریخ ارسال مقاله: ۱۳۹۷/۰۲/۰۶
تاریخ اصلاح مقاله: ۱۳۹۷/۰۵/۱۲، ۱۳۹۷/۰۶/۱۶و ۱۳۹۷/۰۹/۰۹
تاریخ پذیرش مقاله: ۱۳۹۸/۰۱/۱۸
نام نویسنده مسئول: وحید سلوک
نشانی نویسنده مسئول: دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی ارومیه، ارومیه، ایران.

۱- مقدمه

بات‌نت[۱] ها یکی از جدیدترین تهدیدهای امنیت سایبری هستند که تشخیص آن‌ها از مهمترین مسائل مطرح در زمینه امنیت شبکه‌های اینترنتی است. بات یک برنامه بدخواه است که بر روی میزبان‌های آسیب‌پذیر اجرا می‌شود [۱]. بات‌نت مجموعه‌ای از سیستم‌های آلوده به بات است که در آن بات‌ها از طریق یک زیرساخت ارتباطی به‌نام کانال فرمان و کنترل[۲] (C&C) توسط مهاجم که با عنوان مدیربات[۳] شناخته می‌شود، از راه دور کنترل می‌شوند [۲]. مدیربات می‌تواند در حالی که هویت‌اش مخفی می‌ماند، توسط سیستم‌های بات، طیف گسترده‌ای از فعالیت‌های بدخواهانه از قبیل انجام حملات جلوگیری از سرویس توزیع‌شده[۴] (DDoS)، ارسال هرزنامه، گسترش بدافزار، نشت اطلاعات، کلاه‌برداری در تعداد کلیک‌ها و سرقت هویت را به‌صورت هماهنگ و با قدرت تخریبی بسیار بالا بر روی قربانی سازمان‌دهی کند.

چرخه حیات بات‌نت‌ها را می‌توان به چهار مرحله اصلی شکل‌گیری، فرمان و کنترل، حمله و پس از حمله، تقسیم نمود [۳]. در مرحله شکل‌گیری، مهاجم توسط مکانیزم‌های انتشار از قبیل مرورگر وب، پست الکترونیکی و اشتراک فایل، میزبان‌های آسیب‌پذیر را آلوده می‌کند. در مرحله فرمان و کنترل، بات با سرویس‌دهنده بات‌نت از طریق کانال C&C ارتباط برقرار می‌کند و مدیربات دستورات را توسط این کانال به بات‌های خود ارسال می‌کند. در مرحله حمله، بات با توجه به دستورات دریافتی از مهاجم، فعالیت‌های بدخواهانه‌اش را انجام می‌دهد. در مرحله پس از حمله، بات‌کدهای باینری خود را به‌روزرسانی می‌کند. بهترین زمان تشخیص بات‌نت مربوط به مراحل پیش از حمله است که بات‌ها هنوز رفتار بدخواهانه‌ای از خود نشان نداده‌اند.

بات‌نت‌ها از نظر پروتکل مورد استفاده در کانال‌های فرمان و کنترل به سه نوع مبتنی‌بر پروتکل IRC، HTTP و نظیربه‌نظیر (P2P) تقسیم‌بندی می‌شوند. اخیراً رایج‌ترین نوع بات‌نت‌ها، مبتنی‌بر پروتکل HTTP هستند. در این نوع بات‌نت‌ها، بات‌ها از URL مخصوص یا آدرس IP تعیین شده توسط مدیربات، استفاده می‌کنند تا به سرویس‌دهنده وب خاصی که نقش سرویس‌دهنده C&C را دارد، متصل شوند [۴]. یکی از مزیت‌های این نوع بات مخفی‌ماندن ترافیک فرمان و کنترل در میان ترافیک HTTP معمول مرور کاربران در صفحات وب است. در این مقاله، روشی جهت تشخیص یکی از انواع بات‌نت‌های مبتنی‌بر پروتکل HTTP ارائه می‌شود. این روش قادر است بات‌ها را علاوه‌بر مرحله حمله، در مرحله فرمان و کنترل نیز شناسایی کند. روش پیشنهادی در برابر کانال‌های رمزنگاری مقاوم است و می‌تواند بات‌ها را در کمترین وقفه زمانی تشخیص دهد. جهت طراحی این سیستم تشخیص اهداف زیر برآورده شده‌اند:

- معرفی یک سیستم تشخیص بات‌نت مبتنی‌بر مدل مخفی مارکوف[۵] (HMM) و رویکرد تحلیل جریان شبکه در وقفه‌های زمانی.
- تحلیل ترافیک شبکه بات‌نت BlackEnergy و استخراج ویژگی‌های شاخص و الگوهای رفتاری متعدد این بات‌نت در مراحل مختلف.
- ارائه مدل مخفی مارکوف براساس ویژگی‌ها و الگوهای رفتاری بات‌نت BlackEnergy و رفتار ترافیک نرمال شبکه.
- ترکیب مجموعه‌داده آزمایشگاه ملی لارنس برکلی[۶] [۵] که شامل طیف گسترده‌ای از ترافیک نرمال شبکه است، با مجموعه‌داده بات‌نت BlackEnergy [۶] و به‌دست آوردن مجموعه‌داده‌ای حجیم که نمونه معتبر و جامعی از ترافیک شبکه دنیای واقعی است.
- ارزیابی روش تشخیص پیشنهادی در پنجره‌های زمانی خیلی کوچک بین ۵ تا ۱۵۰ ثانیه، و به‌دست آوردن نتایج بهتر در میزان نرخ مثبت صحیح[۷] و نرخ مثبت کاذب[۸] نسبت به روش‌های دیگر.
- تشخیص مرحله فعالیت بات‌نت علاوه‌بر شناسایی بات‌نت. در این پژوهش برای اولین بار در کنار تشخیص بات، مرحله فعالیت آن نیز با دقت بالای ۹۰ درصد تشخیص داده می‌شود.

ساختار این مقاله بدین صورت است که در بخش دوم مفاهیم اولیه و مروری بر کارهای گذشته در زمینه تشخیص بات‌نت ارائه می‌شود. در بخش سوم، روش تشخیص بات‌نت پیشنهادی معرفی می‌شود. در بخش چهارم، مجموعه داده‌ها، معیارها و نتایج ارزیابی سیستم تشخیص بات‌نت پیشنهادی، به‌تفصیل بیان می‌شود. در نهایت، بخش پنجم مقاله شامل جمع‌بندی و ارائه رویکرد آینده می‌شود.

۲- پیش‌زمینه و کارهای مرتبط

۲-۱- بات‌نت BlackEnergy

بات‌نت BlackEnergy مبتنی‌بر پروتکل HTTP بوده و به‌طور عمده برای حملات ممانعت از سرویس توزیع‌شده، استفاده می‌شود. البته در سال‌های اخیر برای انواع دیگر حملات سایبری از جمله سرقت اطلاعات و کلاه‌برداری‌های بانکی نیز به‌کار می‌رود. این بات‌نت فعالیت خود را از سال ۲۰۰۷ آغاز کرده و تا کنون ۳ نسخه از آن شناسایی شده است. نسخه اول این بات‌نت، وب سایت‌های روسی را با حملات DDoS مورد هدف قرار می‌داد [۷]، همچنین در سال ۲۰۰۸ و هنگام مقابله روسیه با گرجستان، منجربه حملات سایبری علیه گرجستان شد. در سال ۲۰۱۰ نسخه دوم این بات‌نت پدیدار گشت. در سال ۲۰۱۴ نسخه سوم مشاهده شد که با انگیزه‌های سیاسی، وب‌سایت‌های اکرین را مورد هدف قرار می‌داد و علاوه‌بر حملات DDoS اقدام به سرقت اطلاعات می نمود [۸]. در سال ۲۰۱۵، یک مطالعه سنجشی [۹] بر روی ۱۶ خانواده از بات‌نت‌های بسیار رایج و فعال انجام شد. این پژوهش نشان داد که بات‌نت BlackEnergy با رتبه سوم جزء دسته خانواده بات‌نت‌های با اندازه بزرگ است، همچنین پایداری و انعطاف‌پذیری بالایی دارد و از نظر فعال‌بودن مقام اول را در میان بات‌نت‌های مورد بررسی دارد.

۲-۲- کارهای مرتبط

تا کنون مقالات بسیاری در زمینه تشخیص باتنت ارائه شده است. در میان این پژوهش‌ها، روشی ارزشمند است که قادر باشد باتنت‌ها را به‌صورت بسیار دقیق، بلادرنگ و قبل از حمله سایبری، تشخیص دهد. بدین منظور، روش ارائه‌شده در این پژوهش مبتنی‌بر دو رویکرد تحلیل رفتار ترافیک شبکه در وقفه‌های جریان و استفاده از مدل مخفی مارکوف است. اکثر پژوهش‌های پیشین در زمینه تشخیص باتنت، براساس روش تحلیل بسته‌های شبکه بوده‌اند. این روش‌ها به دلیل پردازش محتوای بسته‌ها، بسیار کند هستند؛ و در برابر باتنت‌های مبتنی‌بر کانال‌های رمزنگاری کارایی ندارند. همچنین به‌دلیل بررسی محتوای بسته‌ها، بحث رعایت حریم خصوصی را نقض می‌کنند. با وجود چنین مشکلاتی، محققین اخیراً به سمت استفاده از روش تحلیل ترافیک شبکه تمایل یافتند.

در سال ۲۰۰۶ [۱۰]، یک روش مبتنی‌بر جریان شبکه برای تشخیص ترافیک C&C باتنت‌های مبتنی‌بر IRC ارائه شد. در این روش ابتدا ترافیک IRC از کل ترافیک شبکه جدا می‌شود. سپس ترافیک باتنت مبتنی‌بر IRC از ترافیک IRC نرمال تشخیص داده می‌شود. در سال ۲۰۰۸ [۱۱]، روش تشخیص باتنت با نام BotMiner معرفی شد. این روش متکی بر تشخیص رفتار گروهی و یکنواخت بات‌ها در یک باتنت است. روش BotMiner می‌تواند باتنت‌ها را مستقل از ساختار و پروتکل کانال فرمان و کنترل، فقط در مرحله حمله تشخیص دهد. در سال ۲۰۱۱ [۱۲]، روش‌های مختلف یادگیری ماشین۹ برای تشخیص باتنت‌های نظیربه‌نظیر به‌کار گرفته شدند. در این پژوهش، در بهترین حالت، مقادیر نرخ مثبت صحیح بالای ۹۰ درصد و نرخ خطای کمتر از ۷ درصد برای روش‌های ماشین بردار پشتیبان[۱۰]، شبکه عصبی مصنوعی[۱۱] و نزدیکترین همسایه[۱۲]، به‌دست آمد. ضعف عمده این پژوهش، عدم توانایی تشخیص باتنت‌های جدید است.

بیشتر سیستم‌های مبتنی‌بر تحلیل ترافیک شبکه پیشین، با بررسی کل جریان میان دو میزبان، باتنت‌ها را تشخیص می‌دهند. در این چند سال، پژوهشگران برای تشخیص برخط و بلادرنگ باتنت‌ها، سیستم‌های تشخیص باتنت مبتنی‌بر تحلیل جریان در وقفه‌های زمانی را ارائه کردند. در سال ۲۰۰۹ [۱۳]، روشی مبتنی‌بر وقفه‌های جریان برای تشخیص باتنت نظیربه‌نظیر Storm توسط وانگ و همکاران ارائه شد. این روش تشخیص براساس نظارت بر پایداری جریان‌های کنترل در فواصل زمانی ۱۰ دقیقه است. در این پژوهش پایداری جریان‌های باتنت Storm اندازه گرفته شد، و از این اصل که بات‌ها در جستجوی فرمان و اجرای وظایف‌شان مکرراً رفتار مشابهی از خود نشان می‌دهند، استفاده شد.

در سال ۲۰۱۲ [۱۴]، از وقفه‌های زمانی برای تشخیص بات استفاده شد. در این پژوهش، یک روش تشخیص مشابه BotMiner، بر اساس مشاهده ویژگی‌های ترافیک شبکه ارائه شد. در این روش ۳ مرحله فرآیند شامل فیلترکردن ترافیک، تشخیص فعالیت‌های مخرب و نظارت بر ترافیک برای گروه‌بندی بات‌ها براساس رفتار گروهی آن‌ها، استفاده شده است. این رویکرد مفهوم جریان‌ها را به دوره‌های زمانی ۶ ساعته تقسیم می‌کند. در سال ۲۰۱۳ [۶]، روش تشخیص باتنت براساس وقفه‌های زمانی و با استفاده از درخت تصمیم، توسط ژائو و همکاران مطرح شد. نتایج رویکرد پیشنهادی در وقفه‌های زمانی بین ۱۰ تا ۳۰۰ ثانیه بر روی مجموعه‌داده‌ی جامع و معتبری ارائه شد. در این پژوهش با افزایش پنجره زمانی دقت تشخیص و نرخ خطا بهبود می‌یابد. در این روش بعد از پایان هر دوره زمانی مجموعه‌ای از بردارهای ویژگی از جریان‌های ترافیک شبکه استخراج می‌شود. سپس با استفاده از این بردارهای ویژگی و روش درخت تصمیم هرس شده[۱۳]، بات‌ها در مراحل حمله و فرمان و کنترل تشخیص داده می‌شوند.

در سال ۲۰۱۶ [۱۵]، رویکرد جدیدی برای تشخیص باتنت براساس وقفه‌های جریان و تکنیک‌های یادگیری ماشین، ارائه شد. در این پژوهش، روش‌های درخت تصمیم ارتقاءیافته۱۴، نایو بیز۱۵ و ماشین بردار پشتیبان برای تشخیص باتنت‌ها به‌کار گرفته شدند. این روش‌ها قادر به شناسایی باتنت‌ها با نرخ مثبت صحیح بالای ۹۲ درصد و نرخ خطای کمتر از ۱ درصد هستند. برای ارزیابی این پژوهش از مجموعه‌داده ی مقاله [۶]، به‌همراه مجموعه‌داده باتنت‌های متنوع دیگر، استفاده شد. در این پژوهش، نتایج رویکرد پیشنهادی در وقفه‌های زمانی بین ۶۰ تا ۳۰۰ ثانیه ارائه شد. این نتایج نشان دادند که با افزایش پنجره زمانی تا ۱۸۰ ثانیه، دقت تشخیص و نرخ خطا بهبود می‌یابد؛ اما با افزایش پنجره زمانی از ۱۸۰ ثانیه تا ۳۰۰ ثانیه، دقت تشخیص کاهش و نرخ خطا افزایش می‌یابد. این مشاهدات نشان می‌دادند که الگوی منحصربفردی توسط باتنت‌ها طی می‌شود؛ به‌طوریکه پنجره زمانی ۱۸۰ ثانویه بیشترین شباهت را به این الگو دارد.

اگرچه تحقیقات بسیاری در حوزه تشخیص باتنت انجام گرفته، ولی در تعداد اندکی از آن‌ها، از مدل مارکوف استفاده شده است. در صورتی که این ابزار قدرتمند ریاضی، به طور وسیعی در پژوهش‌های مختلف از جمله تشخیص دست‌خط [۱۶، ۱۷]، صحبت [۱۸] و اشیاء [۱۹]، و پیش‌بینی درخواست [۲۰] به‌کار رفته است. در سال ۲۰۱۱ [۲۱]، روشی مبتنی‌بر مدل مخفی مارکوف توسط لو و بروکس ارائه شد که می‌توانست بات‌های مبتنی‌بر پروتکل HTTP را با دقت بالای ۹۰ درصد تشخیص دهد. در این پژوهش جریان‌های ترافیک شبکه سیستم اسکادا۱۶ مورد بررسی قرار گرفت، و با تحلیل داده‌های زمانی ترافیک بات Zeus، مدل مخفی مارکوفی با چهار حالت ارائه شد. این مدل می‌توانست باتنت Zeus را با دقت قابل قبولی تشخیص دهد.

در سال ۲۰۱۲ [۲۲]، بر اساس الگوهای رفتاری خاص باتنت‌ها در اسکن‌کردن پورت‌ها یک روش مبتنی‌بر مدل مخفی مارکوف توسط کیم و همکاران ارائه شد. در این پژوهش، ابتدا توسط یک دسته‌بند ساده متنی، الگوهای رفتاری اسکن پورت باتنت‌ها استخراج شد؛ سپس از این الگوها برای آموزش مدل مخفی مارکوف استفاده شد. این روش توانست با نرخ بیش از ۳۰ درصد، تشخیص زودهنگام و قبل از حمله داشته باشد.

در سال ۲۰۱۶ [۲۳]، یک رویکرد جدید جهت پیش‌بینی حملات باتنت توسط عبید و همکاران معرفی شد. این روش با ارائه هشدارهای اولیه به مدیران شبکه، به آن‌ها کمک می‌کند تا در اسرع وقت و قبل از حمله، میزبان‌های آلوده را آشکار سازند. روش پیشنهادی مبتنی‌بر مدل‌سازی توالی آلودگی باتنت به‌عنوان یک زنجیره مارکوف[۱۷] است. هدف این روش شناسایی رفتارهای است که احتمالاً منجربه حملات باتنت می شود. آن‌ها نشان دادند که با استفاده از این روش می‌توان بیش از ۹۸ درصد از حملات انواع خانواده‌های باتنت را پیش‌بینی کرد.

۳- روش تشخیص باتنت پیشنهادی

سیستم تشخیص باتنت ارائه‌شده در این پژوهش، براساس دو رویکرد تحلیل رفتار ترافیک شبکه در وقفه‌های جریان و مدل مخفی مارکوف است. این سیستم قادر است با استفاده از الگوهای رفتار ترافیک شبکه باتنت و در وقفه های زمانی خیلی کم، بات ها را در مراحل اولیه از چرخه حیات باتنت شناسایی کند. در این بخش ابتدا مسئله تشخیص باتنت براساس مدل مخفی مارکوف، مدل‌سازی می‌شود. سپس معماری سیستم تشخیص باتنت و گام‌های تشخیص معرفی می شود. در مرحله بعد ترافیک شبکه و همچنین ویژگی‌های جریان‌های آن مورد تحلیل و بررسی قرار می‌گیرد. سپس الگوهای رفتاری استخراج‌شده از ترافیک شبکه باتنت BlackEnergy توصیف می شود. درنهایت سیستم تشخیص باتنت BlackEnergy ارائه می‌شود.

۳-۱- مدل‌سازی مسئله تشخیص باتنت

مدل مخفی مارکوف از لحاظ ساختار ریاضی بسیار قدرتمند است، که اگر به‌صورت مناسبی پیاده‌سازی شود نتایج خوبی خواهد داشت. در این بخش مدل سیستم تشخیص باتنت مبتنی‌بر HMM معرفی می‌شود. که پارامترهای آن به‌صورت زیر تعیین می‌شوند:

- **مجموعه حالات**[۱۸]: هر جریان از ترافیک شبکه در هر وقفه زمانی می‌تواند در حالت نرمال یا در حالت فرمان و کنترل و یا در حالت حمله قرار گرفته باشد. بدین ترتیب مجموعه حالات S به‌صورت زیر بیان می‌شود:

$$S = \{\text{Normal, C\&C, Attack}\} \quad (۱)$$

- **مجموعه مشاهدات**[۱۹]: ویژگی‌ها و الگوهای رفتاری هریک از جریان‌های ترافیک شبکه، مجموعه مشاهدات را تشکیل می‌دهند. بدین ترتیب مجموعه مشاهدات V به‌صورت زیر بیان می‌شود:

$$V = \{\text{Normal Sign, C\&C Sign, Attack Sign1, Attack Sign2}\} \quad (۲)$$

مشاهده NormalSign، مجموعه‌ی k عضوی از ویژگی‌های منتخب ترافیک شبکه نرمال است:

$$\text{NormalSign} = \{v_0, ..., v_{k-1}\} \quad (۳)$$

مشاهده C&CSign، مجموعه‌ی l عضوی از الگوهای رفتاری ترافیک شبکه بات در کانال C&C است:

$$\text{C\&CSign} = \{v_k, ..., v_{k+l-1}\} \quad (۴)$$

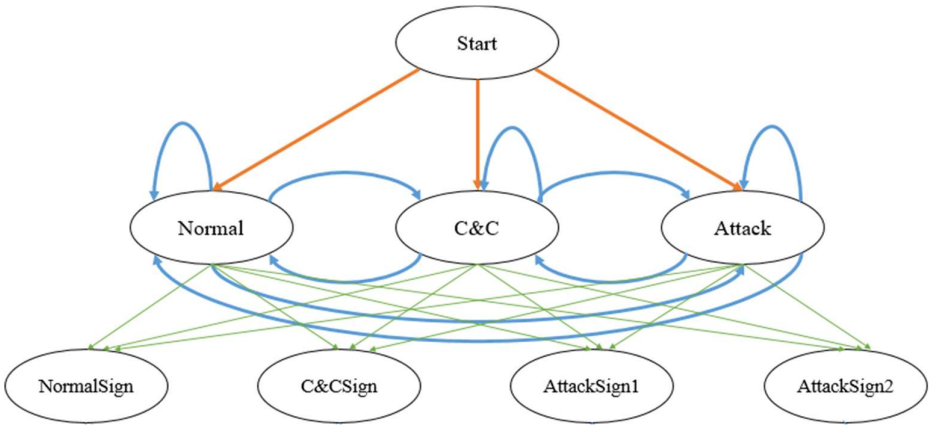
مشاهده AttackSign1، مجموعه‌ی m عضوی از الگوهای رفتاری ترافیک شبکه بات در حالت حمله است:

$$\text{AttackSign1} = \{v_{k+l}, ..., v_{k+l+m-1}\} \quad (۵)$$

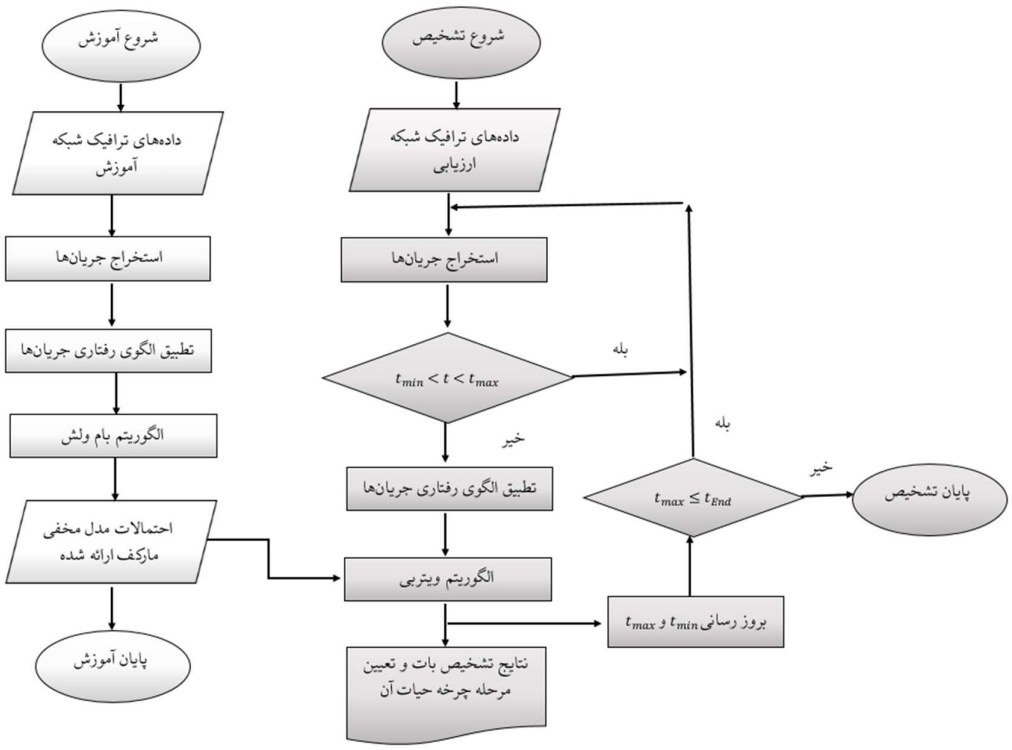
مشاهده AttackSign2، مجموعه‌ی n عضوی از الگوهای رفتاری ترافیک شبکه بات در حالت حمله است:

$$\text{AttackSign2} = \{v_{k+l+m}, ..., v_{k+l+m+n-1}\} \quad (۶)$$

در شکل (۱) مدل ارائه‌شده جهت تشخیص باتنت نشان داده شده است. همانطور که در شکل ملاحظه می‌شود هر کدام از مشاهدات شامل یک مجموعه است. اعضای این مجموعه‌ها با توجه به ویژگی‌ها و الگوهای رفتاری انواع باتنت‌ها با هدف تشخیص آن‌ها تعیین می‌شوند. در بخش بعدی، مدل ارائه‌شده را برای تشخیص باتنت BlackEnergy به‌کار گرفته و تغییرات مورد نیاز را به آن اعمال می‌شود.



شکل ۱: مدل مخفی مارکوف سیستم تشخیص باتنت پیشنهادی



شکل ۲: روندنمای سیستم تشخیص باتنت پیشنهادی

۳-۲- معماری سیستم تشخیص باتنت

سیستم تشخیص باتنت پیشنهادی شامل دو مرحله‌ی اصلی شامل آموزش و تشخیص است. شکل (۲) این مراحل و گام‌های تشخیص باتنت را نشان می‌دهد. در مرحله آموزش احتمالات گذر مدل HMM پیشنهادی محاسبه می‌شود. این مرحله از سه گام تشکیل شده است. در گام نخست، از داده‌های ترافیک شبکه ورودی، بردارهای جریان استخراج می‌شود. در گام بعدی الگوهای رفتاری بردارهای جریان با ویژگی‌ها و الگوهای رفتاری باتنت BlackEnergy مقایسه می‌شود و در صورت تطبیق هر الگو، مشاهده‌ای به بردار مشاهدات اضافه می‌گردد. در گام سوم بردار مشاهدات همراه با احتمالات اولیه مدل مخفی مارکوف پیشنهادی به‌عنوان ورودی به الگوریتم بام‌ولش[۲۰] داده می‌شود که خروجی این الگوریتم بهترین مقدار احتمالات گذر مدل مخفی مارکوف پیشنهادی خواهد بود.

در مرحله تشخیص، مشخص می‌شود که هر جریان در هر وقفه زمانی، در کدامیک از سه حالت: نرمال، فرمان و کنترل و یا حمله قرار دارد. بدین ترتیب بات و مرحله حیات آن شناسایی می‌شود. این مرحله نیز دارای سه گام است. ابتدا بردارهای جریان از داده‌های ترافیک شبکه در بازه زمانی $[t_{\min}\ t_{\max}]$ استخراج می‌شود. سپس با مقایسه رفتار بردارهای جریان و الگوهای رفتاری باتنت، بردار مشاهدات به‌دست می‌آید. در نهایت، بردار مشاهدات همراه با مقدار احتمالات گذر مدل HMM، که خروجی الگوریتم بام‌ولش در مرحله آموزش بود، به‌عنوان ورودی به الگوریتم ویتربی[۲۱] داده می‌شود. خروجی این الگوریتم، بردار حالات بهینه است که همان نتایج تشخیص مدل می‌باشد. سپس میزان $t_{\min}$ و $t_{\max}$ به‌روزرسانی می‌شود، و در صورتی که مقدار $t_{\max}$ از زمان آخرین بسته مجموعه‌داده ارزیابی، t_{End}، بیشتر باشد؛ مرحله تشخیص پایان می‌یابد. در بخش ۳-۵، مرحله‌های آموزش و تشخیص باتنت BlackEnergy، به‌صورت مفصل بیان خواهد شد.

۳-۳- تحلیل ترافیک شبکه

تحلیل ترافیک شبکه بر این اصل استوار است که بات‌های یک باتنت معمولاً رفتار ترافیکی مشابهی دارند، که می‌توان باتنت را براساس آن تشخیص داد. در حوزه تحلیل ترافیک شبکه، منظور از جریان، ارتباط میان دو میزبان منحصربه‌فرد است که شامل تبادل بسته‌های شبکه بین آن دو می‌شود [۶]. سیستم‌های تشخیص باتنت مرسوم مبتنی‌بر تحلیل ترافیک شبکه، کل جریان میان دو میزبان را بررسی می‌کردند، در نتیجه برای تشخیص برخط و بلادرنگ باتنت مناسب نیستند. روشی که در این مقاله برای تشخیص باتنت معرفی می‌شود، براساس تحلیل ترافیک شبکه در وقفه‌های جریان است. این روش با بررسی خصوصیات جریان‌های موجود در پنجره‌های زمانی، باتنت‌ها را به‌صورت بلادرنگ تشخیص می‌دهد. انتخاب اندازه این پنجره زمانی، براساس توافق میان سرعت و دقت تشخیص خواهد بود [۶]. همچنین چرخه الگوهای رفتاری باتنت‌ها نقش بسیار عمده‌ای در انتخاب اندازه این پنجره زمانی دارد.

ویژگی، خصوصیتی از یک جریان یا مجموعه‌ای از جریان‌ها در پنجره زمانی تعیین شده است. در جدول (۱) مجموعه‌ای از ویژگی‌هایی نشان

داده می‌شود که برای ساخت سیستم تشخیص باتنت مورد توجه قرار گرفته است. مجموعه ویژگی‌ها براساس رفتار عموم بات‌ها خصوصاً بات BlackEnergy انتخاب شده‌اند. برخی ویژگی‌ها مانند طول بسته تبادل شده و پروتکل مورداستفاده، مستقیماً از سربارهای TCP/UDP استخراج می‌شوند. در حالی که سایر ویژگی‌ها مانند متوسط زمان بین بسته‌ها در وقفه زمانی، نیازمند محاسبات اضافی هستند. در انتخاب ویژگی‌ها، ویژگی آدرس‌های IP مبداء و مقصد، درنظر گرفته نشدند، چرا که متناقض با هدف طراحی سیستم تشخیص بات با قابلیت جابجایی است.

الگوهای رفتاری باتنت BlackEnergy به‌طور مستقیم براساس ویژگی‌های پروتکل مورداستفاده در انتقال، طول بسته تبادل‌شده، فاصله زمانی بین بسته‌ها و اندازه اولین بسته انتقالی، استخراج می‌شوند. با بررسی‌هایی که انجام شد، دو ویژگی زیر به‌عنوان ویژگی‌های شاخص این بات انتخاب شدند تا در کنار الگوهای رفتاری بات، در مدل HMM سیستم تشخیص بات استفاده شوند:

- تعداد بسته‌های مبادله‌شده در هر واحد زمان از وقفه زمانی
- محدوده طول بسته‌ها در وقفه زمانی

جدول ۱: ویژگی‌های منتخب باتنت BlackEnergy

توضیحات	ویژگی
پروتکل مورداستفاده در لایه انتقال	Protocol
طول بسته تبادل‌شده	Packet_Lenght
طول اولین بسته انتقالی در جریان	First_Packet_size
محدوده طول بسته‌ها در وقفه زمانی	Packet_Lenght_Range
متوسط زمان بین بسته‌ها در وقفه زمانی	Time_between_Packet
تعداد بسته‌های مبادله‌شده در هر واحد زمان از وقفه زمانی	Packet_Exchanged

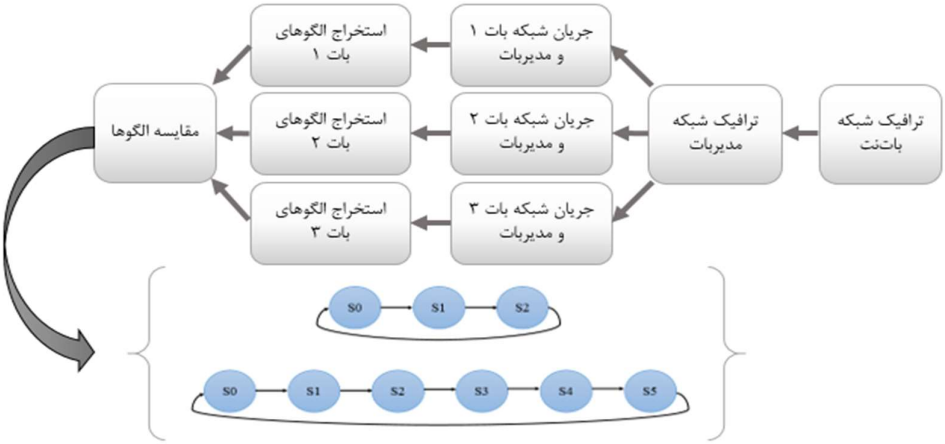
۳-۴- تحلیل رفتار ترافیکی بات BlackEnergy

در این مقاله برای ارزیابی سیستم تشخیص باتنت پیشنهادی از نسخه ۱/۷ باتنت BlackEnergy استفاده شده است [۷]. این بات، برنامه کم حجم (کمتر از ۵۰ کیلو بایت) برای سیستم عامل ویندوز بوده و از یک گرامر ساده برای ارتباط و رمزنگار زمان اجرا برای مقابله با تشخیص آنتی ویروس استفاده می‌کند. این بات برای ارتباط با مدیربات، پیام POST را به سرویس‌دهنده‌ی C&C ارسال می‌کند. پاسخ دریافتی از سرویس‌دهنده به‌صورت کدگذاری‌شده بر پایه۶۴ است که با تبدیل و کدگشایی آن، فرمان به‌دست می‌آید. سپس بات فعالیت‌اش را براساس فرمان دریافتی، به‌روزرسانی و اجرا می‌کند. در این پژوهش پس از تحلیل و بررسی مجموعه داده باتنت BlackEnergy [۷]، الگوهای رفتار ترافیکی آن استخراج شد. منظور از الگوی رفتار ترافیکی باتنت، روند تبادل پیامی است که به‌صورت متناوب در کل ارتباطات باتنت تکرار می‌شود. در ادامه الگوهای رفتاری این باتنت در مرحله C&C و مرحله حمله معرفی می‌شوند.

۳-۴-۱- الگوی رفتار ترافیکی بات در مرحله C&C

همانطور که قبلاً اشاره شد، مرحله فرمان و کنترل (C&C) مرحله‌ای از چرخه حیات بات است که با مدیربات در ارتباط است. برای استخراج الگوی رفتاری بات در این مرحله، ابتدا ترافیک شبکه مربوط به آدرس IP مدیربات را از کل ترافیک شبکه باتنت جدا شد. سپس جریان‌های متعدد بات‌ها از ترافیک مدیربات استخراج شدند. مشاهده شد که جریان‌های شبکه بین مدیربات با هریک از بات‌ها بسیار مشابه هستند. این جریان‌ها از الگوهای رفتاری تقریباً یکسانی پیروی می‌کنند، که در طول زمان به‌صورت متناوب تکرار می‌شوند. با محاسبه و بررسی مقادیر ویژگی‌هایی که در بخش قبل معرفی شدند، دو الگوی خاص رفتار ترافیکی برای ارتباط بات با مدیربات استخراج شد. شکل (۳) روند استخراج این دو الگو را نشان می‌دهد.

الگوی رفتاری اول بات در کانال C&C، شامل سه حالت S0، S1 و S2 است. در جدول (۲) مشخصات این حالات نشان داده می‌شود.



شکل ۳: روند استخراج الگوی رفتاری مرحله فرمان و کنترل

الگوی رفتاری دوم بات در کانال C&C، شامل شش حالت است. در جدول (۳) مشخصات حالات S0، S1، ... و S5 بیان می‌شود.

جدول ۲: مشخصات الگوی رفتاری اول بات در کانال C&C

حالت	پروتکل	طول بسته
S0	TCP	62
S1	TCP	62
S2	HTTP	---

جدول ۳: مشخصات الگوی رفتاری دوم بات در کانال C&C

حالت	پروتکل	طول بسته
S0	TCP	62
S1	TCP	62
S2	TCP	60
S3	HTTP	---
S4	HTTP	---
S5	TCP	60

۳-۴-۲- الگوی رفتار ترافیکی بات در مرحله حمله

بات پس از مرحله فرمان و کنترل، وارد مرحله حمله می‌شود. در این مرحله، بات بر روی سیستم قربانی حملات متنوع DDoS پیاده می‌کند. برای استخراج الگوی رفتاری بات در این مرحله، ابتدا ترافیک شبکه مربوط به آدرس IP مدیربات از کل ترافیک شبکه بات‌نت حذف می‌شود. سپس ترافیک شبکه هریک از بات‌ها را از ترافیک بات‌نت جدا می‌کنیم. مشاهده شد که ترافیک شبکه همه بات‌ها از الگوهای رفتاری تقریباً یکسانی پیروی می‌کند که در طول زمان به‌صورت متناوب تکرار می‌شود. با محاسبه و بررسی مقادیر ویژگی‌ها، ۷ الگوی خاص رفتار ترافیکی برای حالت حمله بات استخراج می‌شود. در شکل (۴) روند استخراج این ۷ الگوی رفتاری نشان داده شده است.

الگوی رفتاری اول بات در مرحله حمله، شامل سه حالت S0، S1 و S2 است، که در جدول (۴) مشخصات این حالات نشان داده می‌شود. الگوی رفتاری دوم بات در این مرحله، شامل چهار حالت S0، S1، S2 و S3 است. در این الگو، کل مدت زمان رخداد حالت‌های S0 تا S3 کمتر از ۰/۱۲ ثانیه است. جدول (۵) مشخصات حالات این الگو را بیان می‌کند. الگوی رفتاری سوم بات در مرحله حمله، شامل پنج حالت است. در این الگو، متوسط طول زمان رخداد حالت‌های S0 تا S3، در بازه زمانی ۰/۰۰۱ تا ۰/۰۰۶ ثانیه قرار دارد. همچنین فاصله زمانی بین رخداد حالت‌های S3 و S4، در بازه ۰/۱ تا ۰/۲ ثانیه قرار گرفته است. این حمله به صورت منظم هر ۲۰ ثانیه یک بار اتفاق می‌افتد. در جدول (۶) مشخصات حالات این الگوی حمله ارائه شده است.

جدول ۴: مشخصات الگوی رفتاری اول بات در مرحله حمله

حالت	پروتکل	طول بسته
S0	HTTP	824
S1	TCP	54
S2	TCP	60

جدول ۵: مشخصات الگوی رفتاری دوم بات در مرحله حمله

حالت	پروتکل	طول بسته
S0	TCP	62
S1	TCP	54
S2	HTTP	824
S3	TCP	54

جدول ۶: مشخصات الگوی رفتاری سوم بات در مرحله حمله

حالت	پروتکل	طول بسته
S0	NCP	84
S1	NCP	70
S2	NCP	100
S3	NCP	70
S4	TCP	60

الگوی رفتاری چهارم بات در مرحله حمله، شامل شش حالت است. جدول (۷) مشخصات این حالات را نشان می‌دهد. در این جدول، حالت‌های S0، S1 و S2 در سطر اول و حالت‌های S3، S4 و S5 در سطر دوم قرار دارند. دلیل این نحوه نمایش، یکسانی پروتکل و طول بسته انتقالی حالت‌های موجود در هر سطر است. الگوی رفتاری پنجم بات در مرحله حمله، شامل ۱۲ حالت است. جدول (۸) مشخصات این حالات را بیان می‌کند. در این جدول، حالت‌هایی که در آن‌ها پروتکل و طول بسته انتقالی یکسان است، در یک سطر قرار گرفته‌اند. الگوی رفتاری ششم بات در مرحله حمله، شامل ۹ حالت است. جدول (۹) مشخصات این حالات را نشان می‌دهد. الگوی رفتاری هفتم بات در مرحله حمله، شامل ۱۸ حالت است. در جدول (۱۰) مشخصات این حالات ارائه شده است. در این جدول نیز حالت‌های S0، S1، S2 و S3 به‌دلیل داشتن پروتکل و طول بسته انتقالی یکسان، در سطر اول قرار گرفته‌اند.

جدول ۷: مشخصات الگوی رفتاری چهارم بات در مرحله حمله

حالت	پروتکل	طول بسته
S0, S1, S2	TCP	60
S3, S4, S5	TCP	66

جدول ۸: مشخصات الگوی رفتاری پنجم بات در مرحله حمله

حالت	پروتکل	طول بسته
S0, S1, S2, S3, S4, S5	TCP	60
S6, S7, S8, S9, S10, S11	TCP	66

جدول ۹: مشخصات الگوی رفتاری ششم بات در مرحله حمله

حالت	پروتکل	طول بسته
S0	TCP	62
S1	TCP	62
S2	TCP	60
S3	HTTP	---
S4	TCP	60
S5	TCP	54
S6	HTTP	824
S7	TCP	54
S8	TCP	60

جدول ۱۰: مشخصات الگوی رفتاری هفتم بات در مرحله حمله

حالت	پروتکل	طول بسته
S0, S1, S2, S3	TCP	62
S4	TCP	60
S5	HTTP	254
S6	TCP	60
S7	TCP	60
S8	HTTP	254
S9	TCP	60
S10	TCP	54
S11	TCP	54
S12	HTTP	824
S13	TCP	54
S14	TCP	60
S15	HTTP	824
S16	TCP	54
S17	TCP	60

۳-۵- سیستم تشخیص بات‌نت BalckEnergy

در این بخش، سیستم تشخیص بات‌نت BlackEnergy مبتنی‌بر مدل مخفی مارکوف، معرفی می‌شود. در بخش گذشته، با تحلیل ترافیک شبکه این بات‌نت، الگوهای رفتاری آن در مراحل C&C و حمله، استخراج شد. باید در نظر داشت که در دنیای واقعی بات‌نت ممکن است از الگویی استفاده کند که قبلاً مشاهده نشده است. بنابراین الگوهای استخراج‌شده برای تشخیص بات‌نت در تمام شرایط کافی نیستند. در نتیجه، باید ویژگی‌های جریان بات نیز در مدل HMM به‌کار گرفته شوند. همان‌طور که در روابط (۱) و (۲) نشان داده شد، مدل مخفی مارکوف پیشنهادی دارای ۳ حالت و ۴ مشاهده است. همچنین در روابط (۳)، (۴)، (۵) و (۶) نشان داده شد که هر کدام از مشاهدات شامل یک مجموعه است. در این بخش، ابتدا اعضای مجموعه‌های مدل HMM ارائه شده، با توجه به ویژگی‌ها و الگوهای رفتاری بات‌نت BlackEnergy تعیین می‌شود. سپس مرحله‌های آموزش و تشخیص بات‌نت BlackEnergy، و همچنین الگوریتم‌های بام‌ولش و ویتربی با جزئیات بیشتر بیان می‌شود.

مشاهده NormalSign: ترافیک نرمال دارای بازه خصوصیات بسیار وسیعی است، و الگوهای رفتاری بی‌شماری را شامل می‌شود. در این پژوهش برای کاهش پیچیدگی مدل تشخیص، فقط یکی از ویژگی‌های ترافیک نرمال به‌عنوان مشاهده در نظر گرفته می‌شود. دلیل انتخاب این ویژگی، طبق این واقعیت است که بات‌نت‌ها دارای رفتار یکنواخت‌تری نسبت به سایر جریان‌ها در شبکه هستند. آن‌ها معمولاً بسته‌های کوچک و با اندازه یکنواخت و تقریباً یکسانی را به‌صورت پیوسته مبادله می‌کنند [۶]. با بررسی‌های انجام شده بر روی بات‌نت BlackEnergy، ویژگی مهم محدودیت اندازه بسته‌های انتقالی در ترافیک این بات‌نت بین اندازه ۵۴ تا ۸۲۴ به‌دست آمد. بدین ترتیب، مشاهده NormalSign یک مجموعه تک عضوی شامل مشاهده حداقل یک بسته انتقالی در یک جریان با اندازه‌ای خارج از محدوده تعیین‌شده برای ترافیک بات‌نت BlackEnergy است. بنابراین v_0، به‌عنوان مشاهده‌ی حداقل یک بسته بزرگتر از ۸۲۴ یا کوچکتر از ۵۴، در طول پنجره زمانی تعریف می‌شود. مجموعه مشاهده NormalSign به‌صورت زیر بیان می‌شود:

$$\text{NormalSign} = \{v_0\} \quad (۷)$$

مشاهده C&CSign: همانطور که در بخش قبل بیان شد، بات‌نت BlackEnergy در مرحله فرمان و کنترل از چرخه حیاتش دارای ۲ الگوی رفتاری منحصربفرد است. بنابراین مشاهده C&CSign، مجموعه این ۲ الگوی رفتار ترافیکی بات‌نت BlackEnergy در کانال فرمان و کنترل با نام‌های v_1 و v_2 در نظرگرفته می‌شود:

$$\text{C\&CSign} = \{v_1, v_2\} \quad (۸)$$

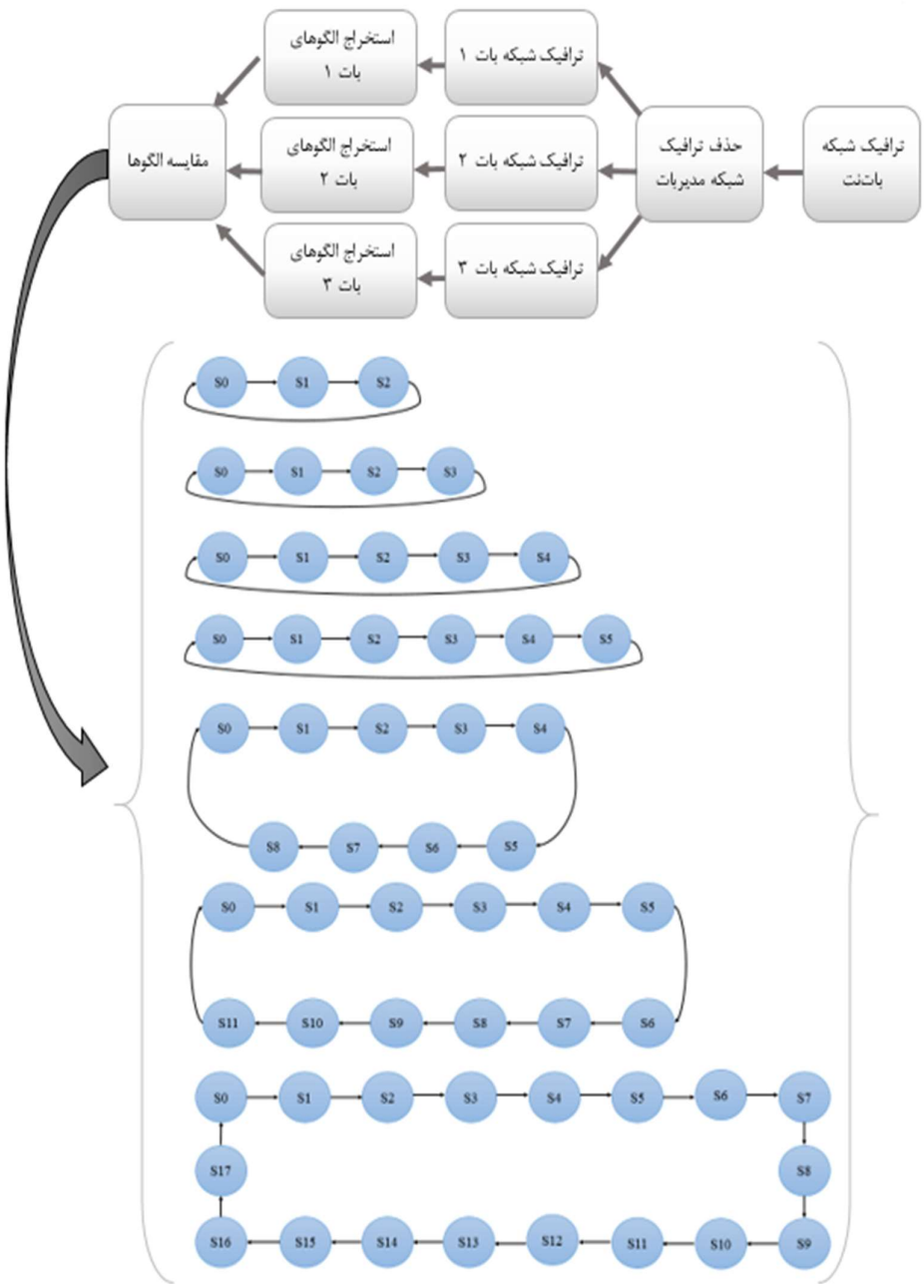
مشاهده AttackSign1: این مشاهده، مجموعه‌ی ۷ عضوی از الگوهای رفتاری ترافیک شبکه بات‌نت BlackEnergy در حالت حمله است که در قسمت قبل توضیح داده شدند:

$$\text{Attack Sign1} = \{v_3, v_4, v_5, v_6, v_7, v_8, v_9\} \quad (۹)$$

مشاهده AttackSign2: هنگام تشخیص بات‌نت‌ها همیشه احتمال مواجه شدن با الگوهای حمله‌ی جدید وجود دارد. با توجه به این‌که حمله اصلی بات‌نت BlackEnergy از نوع DDoS است؛ بنابراین ویژگی اصلی این بات‌نت، تعداد زیاد بسته‌های انتقالی در واحد زمان است. براساس آمارهای گرفته‌شده، ویژگی مهم بیشتر حملات این بات‌نت، مبادله بیش از ۲۰ بسته در واحد زمان (یک ثانیه) در طول یک جریان است. این ویژگی با نام v_{10}، به‌عنوان مشاهده AttackSign2 در نظر گرفته می‌شود:

$$\text{Attack Sign1} = \{v_{10}\} \quad (۱۰)$$

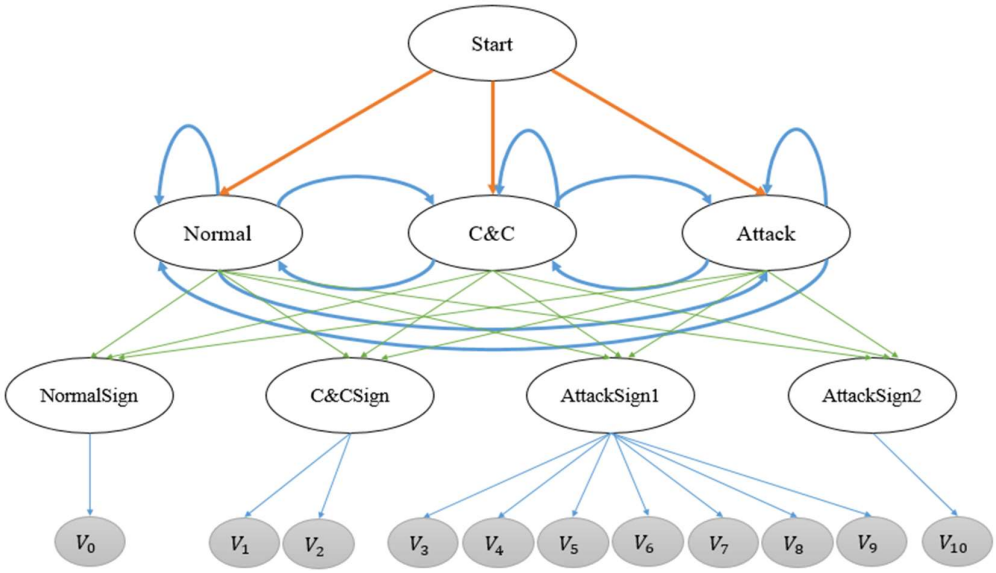
با اعمال مجموعه مشاهدات بات‌نت BlackEnergy در مدل مخفی مارکوف ارائه شده در شکل (۱)، سیستم تشخیص بات‌نت BlackEnergy به‌صورت شکل (۵) نمایش داده می‌شود.



شکل ۴: روند استخراج الگوی رفتاری مرحله حمله

۳-۵-۱- مرحله آموزش

هدف مرحله آموزش این است که با داشتن دنباله مشاهدات و ساختارکلی مدل HMM شامل تعداد حالت‌ها و تعداد مشاهدات، پارامترهای مدل $\lambda = (A, B, \pi)$ طوری محاسبه شوند که بیشترین تناسب را با داده‌های آموزشی داشته باشند. در مدل λ، ماتریس احتمالات گذر با متغیر A، ماتریس احتمالات مشاهدات با متغیر B، و بردار احتمالات اولیه با متغیر π نمایش داده می‌شود. روند این مرحله و گام‌های آن پیش‌تر در شکل (۲) نشان داده شده است. در این مرحله ابتدا از داده‌های ترافیک شبکه ورودی، بردارهای جریان استخراج می‌شود. سپس ویژگی‌ها و الگوهای رفتاری بردارهای جریان با هریک از اعضای مجموعه مشاهدات $V = \{v_0, v_1, v_2, ..., v_{10}\}$ که در بخش قبل مشخص شدند، مقایسه می‌شود. در صورت رویت هریک از مشاهدات در بردارهای جریان، مشاهده‌ای به بردار مشاهدات اضافه می‌گردد. در نهایت بردار مشاهدات همراه با احتمالات اولیه به‌عنوان ورودی به الگوریتم بام‌ولش داده می‌شود که خروجی این الگوریتم بهترین مقدار احتمالات گذر مدل مخفی مارکوف پیشنهادی خواهد بود.



شکل ۵: سیستم تشخیص باتنت BlackEnergy

الگوریتم بام‌ولش

در این پژوهش برای آموزش از الگوریتم حداکثرسازی[۲۲] امید ریاضی[۲۳] (EM) به‌عنوان یک نمونه از الگوریتم بام‌ولش استفاده می‌شود. همگرایی این الگوریتم مطلوب بوده و نتایج خوبی برای مدل پیشنهادی داشت. الگوریتم EM شامل دو گام مهم امید ریاضی و حداکثرسازی است. مراحل آموزش مدل در این الگوریتم به‌صورت زیر است:

۱) مرحله مقداردهی اولیه: پارامترهای اولیه مدل λ تعیین می‌شود.

۲) مرحله امید ریاضی: متغیرهای کمکی $\xi_t(i,j)$ و $\lambda_t(i)$ با استفاده از روابط (۱۲) و (۱۴) محاسبه می‌شوند. متغیر $\xi_t(i,j)$، احتمال وقوع در حالت i، در زمان t و در حالت j، در زمان $t+1$ است. این متغیر به‌صورت رابطه (۱۱) تعریف می‌شود، که می‌توان آن را به‌صورت رابطه (۱۲) نیز بیان کرد. متغیر $\lambda_t(i)$ بیانگر احتمال حضور در حالت i، در زمان t، با داشتن دنباله مشاهدات O و مدل مخفی مارکوف λ است. این متغیر به‌صورت رابطه (۱۳) تعریف می‌شود، که می‌توان آن را توسط $\xi_t(i,j)$ به‌صورت رابطه (۱۴) بیان کرد.

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j \mid O, \lambda) \qquad (۱۱)$$

$$\xi_t(i,j) = \frac{P(q_t = i, q_{t+1} = j, O \mid \lambda)}{P(O \mid \lambda)} \qquad (۱۲)$$

$$\gamma_t(i) = P(q_t = i \mid O, \lambda) \qquad (۱۳)$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j), \quad 1 \le i \le N, \ 1 \le t \le T \qquad (۱۴)$$

در رابطه فوق، متغیر N نشان‌دهنده تعداد حالات است که در مدل تشخیص باتنت پیشنهادی برابر با ۳ است. همچنین متغیر T نشان‌دهنده تعداد اعضای دنباله ورودی مشاهدات است.

۳) مرحله حداکثرسازی: مدل $\bar{\lambda}$ با استفاده از مدل λ و فرمول‌های باز تخمین (۱۵)، (۱۶) و (۱۷) محاسبه می‌شود.

$$\bar{\pi}_{i_t} = \gamma_1(i), 1 \le i \le N \qquad (۱۵)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i,j)}, \quad 1 \le i \le N, \ 1 \le j \le N \qquad (۱۶)$$

$$\bar{b}_j(k) = \frac{\sum_{t=1, Ot=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \ 1 \le i \le N, \ 1 \le k \le M \qquad (۱۷)$$

در روابط فوق، متغیرهای N، M و T به‌ترتیب نشان‌دهنده تعداد حالات، تعداد مشاهدات که در این مدل برابر با ۴ است، و تعداد اعضای دنباله ورودی مشاهدات است.

۴) مرحله به‌روز رسانی: $\lambda \leftarrow \bar{\lambda}$

۵) بازگشت به مرحله امید ریاضی

روند فوق تا زمانی که میزان نسبت شباهت بهبود مناسبی داشته باشد ادامه می‌یابد.

۳-۵-۲- مرحله تشخیص

هدف مرحله تشخیص این است که با داشتن دنباله مشاهدات O و مدل $\lambda = (A, B, \pi)$، دنباله حالات بهینه‌ی Q محاسبه شود. در نتیجه با استفاده از بردار مشاهدات می‌توان تشخیص داد که سیستم در هنگام

رویداد هر مشاهده در کدامیک از حالت‌های نرمال، کانال فرمان و کنترل و یا حمله قرار داشته است. برای حل این مسئله از الگوریتم ویتربی استفاده می‌شود. روند کار مرحله تشخیص و گام‌های آن پیش‌تر در شکل (۲) نشان داده شده است. در این مرحله، ابتدا در بازه زمانی $[t_{\min}\ t_{\max}]$ بردارهای جریان از داده‌های ترافیک شبکه استخراج می‌شود. سپس مانند مرحله آموزش، ویژگی‌ها و الگوهای رفتاری بردارهای جریان با هریک از اعضای مجموعه $V=\{v_0,v_1,v_2,...,v_{10}\}$ مقایسه می‌شود. در صورت رویت هریک از مشاهدات در بردارهای جریان، مشاهده‌ای به بردار مشاهدات اضافه می‌گردد. در گام بعد، بردار مشاهدات همراه با مقدار احتمالات گذر مدل HMM که خروجی الگوریتم بام‌ولش در مرحله آموزش بود، به‌عنوان ورودی به الگوریتم ویتربی داده می‌شود.

خروجی الگوریتم ویتربی، بردار حالات بهینه یا همان نتایج تشخیص مدل پیشنهادی خواهد بود. گاهی ممکن است در حین روند تشخیص در طول وقفه زمانی هیچ مشاهده‌ای در بردارهای جریان رویت نشود، در این صورت روش تشخیص پیشنهادی هیچ پیام هشداری نداده و حالت ترافیک شبکه را در این مورد، نرمال تشخیص می‌دهد. پس از اجرای الگوریتم ویتربی و ارائه گزارش تشخیص بات‌نت در هر وقفه زمانی، مقادیر $t_{\min}$ و $t_{\max}$ به‌روزرسانی می‌شوند. گام‌های مرحله تشخیص تا زمانی که مقدار $t_{\max}$ از زمان آخرین بسته مجموعه‌داده ارزیابی، t_{End}، کمتر یا مساوی باشد تکرار می‌شود. در این پژوهش برای اینکه دقت روش تشخیص پیشنهادی ارزیابی شود، در حین هر مشاهده، حالت واقعی جریان نیز به بردار حالات واقعی اضافه می‌شود. در نهایت، با مقایسه بردار حالات بهینه‌ی به‌دست‌آمده از الگوریتم ویتربی و بردار حالات واقعی، دقت تشخیص محاسبه می‌شود.

الگوریتم ویتربی

در مسأله تشخیص، با داشتن دنباله مشاهدات $O=\{O_0,O_1,...,O_t\}$ و مدل $\lambda=(A,B,\pi)$، باید دنباله حالات بهینه‌ی $Q=\{q_0,q_1,...,q_t\}$ را که دنباله مشاهدات O را تولید کرده‌اند، به‌دست آید. الگوریتم ویتربی دنباله حالات بهینه را با بیشترین مقدار نسبت شباهت محاسبه می‌کند. با استفاده از این روش می‌توان توسط روابط (۱۸)، (۱۹)، (۲۰) و (۲۱) بردار حالات مورد نظر را به‌دست آورد.

$$\delta_t(i)=\max_{q_1,q_2,...,q_{t-1}} P(q_1,...,q_{t-1},q_t=i,o_1,...,o_{t-1}\mid\lambda) \quad (۱۸)$$

$\delta_t(i)$، در شرایطی که حالت فعلی برابر با i باشد، بیشترین مقدار احتمال برای دنباله حالات و دنباله مشاهدات در زمان t را به‌دست می‌دهد. به‌همین ترتیب می‌توان روابط بازگشتی زیر را نیز به‌دست آورد:

$$\delta_{t+1}(j)=b_j(ot+1)\left[\max_{1\le i\le N}\delta_t(j)a_{ij}\right],\quad 1\le i\le N,\ 1\le t\le T-1 \quad (۱۹)$$

$$\delta_1(j)=\pi_j b_j(o_1),\ 1\le j\le N \quad (۲۰)$$

در این روش در هر زمان یک اشاره‌گر به حالت برنده قبلی خواهد بود. در نهایت حالت j^* با داشتن شرط زیر به دست می‌آید:

$$j^*=\arg\max_{1\le j\le N}\delta_t(j) \quad (۲۱)$$

با شروع از حالت j^* دنباله حالات به شکل بازگشت به عقب و با دنبال کردن اشاره‌گر به حالات قبلی به‌دست می‌آید. با استفاده از این روش می‌توان مجموعه حالات مورد نظر را به‌دست آورد.

۴- نتایج آزمایش‌ها و تحلیل‌ها

در این بخش، نحوه پیاده‌سازی روش تشخیص پیشنهادی، مجموعه داده‌ها، معیارهای ارزیابی و نتایج آزمایش‌های انجام‌شده، شرح داده می‌شوند. تمامی فرایند پیاده‌سازی این پژوهش در سیستم عامل لینوکس نسخه اوبونتو ۲۰۱۵/۱۰ و با زبان برنامه‌نویسی C انجام گرفته است. دلیل عمده پیاده‌سازی روش پیشنهادی به زبان C، مربوط به گزارشی می‌شود که در [۲۴] ارائه شده است. این گزارش شامل مقایسه میان سرعت زبان‌های مرسوم برنامه‌نویسی در پیاده‌سازی الگوریتم بام‌ولش است. که در آن، زبان C رتبه اول را در میان تمام زبان‌های برنامه نویسی برای پیاده‌سازی مدل‌های مخفی مارکوف با تعداد حالات و مشاهدات کم، متوسط و زیاد دارد. در این پژوهش از نرم‌افزار پایش و مدیریت شبکه‌ی Wireshark و همچنین از نرم‌افزار Tcpdump کامپیوتری استفاده شده است. از برنامه‌ی Editcap نیز برای مشاهده و بررسی بسته‌های مجموعه داده‌ها، ویرایش و ترکیب آن‌ها با یکدیگر و ساختن مجموعه‌داده مناسب برای ارزیابی روش پیشنهادی استفاده می‌شود.

۴-۱- مجموعه داده‌ها

به‌منظور ارزیابی روش پیشنهادی توسط مجموعه داده‌ی مشابه دنیای واقعی، مجموعه داده‌های ترافیک بات‌نت BlackEnergy [۶] و ترافیک نرمال آزمایشگاه ملی لارنس برکلی (LBNL) [۵] با هم ترکیب می‌شوند. مجموعه داده بات‌نت BlackEnergy، شامل ترافیک داده بدخواه با حجم ۱۹ مگا بایت است که توسط دو بات‌نت تولید می‌شود. بات‌نت اول دارای یک مدیربات و ۳ بات است که به یک سیستم قربانی حمله می‌کنند. بات‌نت دوم نیز دارای یک مدیربات و یک بات است که به سیستم قربانی دیگری حمله می‌کند. تعداد کل بسته‌های تبادل‌شده در این مجموعه داده برابر با ۱۰۹۰۰۲ است.

داده‌های شبکه LBNL در طول دوره سه ماهه از اکتبر ۲۰۰۴ تا ژانویه ۲۰۰۵ جمع‌آوری شده است که ۲۲ زیر شبکه را پوشش می‌دهد و ۱۱ گیگا بایت حجم دارد. این مجموعه داده شامل انواع مختلف سرویس‌های شبکه از قبیل وب، ایمیل، گرفتن نسخه پشتیبان و جریان رسانه است. دو زیرمجموعه داده‌ی D1 و D2 به‌صورت تصادفی از این مجموعه داده بزرگ، انتخاب می‌شوند. در جدول (۱۱) اطلاعات کلی مربوط به این دو زیرمجموعه داده و مجموعه داده BlackEnergy ارائه می‌شود. برای انجام آزمایش‌ها، دو مجموعه داده از ترکیب مجموعه داده‌های D1، D2 و BlackEnergy ساخته می‌شوند. این مجموعه داده‌ها دارای فیلدهای زمان دریافت بسته، آدرس مبداء بسته، آدرس مقصد بسته، پروتکل و

جدول ۱۲: مشخصات مجموعه‌داده‌های ساخته‌شده برای آزمایش

مجموعه‌داده	حجم مجموعه‌داده	ترافیک نرمال (٪)	ترافیک باتنت (٪)	تعداد بسته‌های آموزش	تعداد بسته‌های آزمون	تعداد کل بسته‌ها
مجموعه‌داده۱	۱۷۱ مگابایت	۹۱	۹	۲۲۹۷۰۵	۱۰۰۰۰۰۰	۱۲۲۹۷۰۵
مجموعه‌داده۲	۱/۴ گیگابایت	۹۸/۹۲	۱/۰۸	۱۰۱۰۹۰۰	۹۰۹۸۱۰۰	۱۰۱۰۹۰۰۰

طول بسته است. اطلاعات این مجموعه داده‌ها در جدول (۱۲) نشان داده می‌شود. برای آزمایش سیستم تشخیص باتنت پیشنهادی، ابتدا مجموعه‌داده۱ ساخته می‌شود. سپس برای ارزیابی بهتر و بیشتر این سیستم، مجموعه‌داده۲ ساخته شده و از روش اعتبارسنجی متقابل ۱۰ لایه‌ای[۲۴] استفاده می‌شود.

۴-۲- معیارهای ارزیابی

در اکثر تحقیقات حوزه تشخیص باتنت، برای ارزیابی روش تشخیص از معیارهای مرسوم نرخ مثبت صحیح و نرخ مثبت کاذب استفاده می‌شود. در این مقاله نیز جهت ارزیابی روش پیشنهادی و مقایسه آن با سایر روش‌ها، این معیارها محاسبه می‌شوند. همچنین برای ارزیابی دقیق‌تر روش پیشنهادی معیارهای جدید زیر معرفی می‌شوند:

نرخ مثبت صحیح حالت حمله باتنت (TPAR): این معیار برابر است با نسبت تعداد تشخیص‌های صحیح حالت حمله باتنت (TPA) به تعداد کل فعالیت‌های باتنت در حالت حمله و به صورت زیر محاسبه می‌شود:

$$TPAR = \frac{TPA}{TPA+FNA} \quad (۲۲)$$

نرخ مثبت صحیح حالت فرمان و کنترل (TPCR) : این معیار برابر است با نسبت تعداد تشخیص‌های صحیح حالت فرمان و کنترل باتنت (TPC) به تعداد کل فعالیت‌های باتنت در حالت فرمان و کنترل که به‌صورت زیر محاسبه می‌شود:

$$TPCR = \frac{TPC}{TPC+FNC} \quad (۲۳)$$

برای اولین بار در حوزه تشخیص باتنت، توسط معیارهای فوق دقت تشخیص مراحل مختلف چرخه حیات باتنت در کنار تشخیص باتنت بررسی می‌شود. نتایج حاصل نشان‌دهنده این است که روش ارائه‌شده می‌تواند علاوه‌بر تشخیص باتنت، مرحله چرخه حیات آن را نیز با دقت مطلوبی شناسایی کند.

جدول ۱۱: مشخصات مجموعه داده‌های آزمایش

مشخصات	BlackEnergy	D1	D2
تاریخ	2008/09/08	2005/01/07	2004/12/15
مدت زمان	00:04:43	01:12:00	02:27:30
تعداد زیرشبکه‌ها	-	۱۸	۲۲
تعداد میزبان‌ها	۸	۱۵۵۸	۲۱۰۲
تعداد بسته‌ها	۱۰۹۰۰۲	۱۰۰۰۰۰	۱۰۰۰۰۰۰۰

معیار AUC: معیار AUC[۲۵]، معیار مؤثر دیگری است که برای تعیین میزان کارایی یک دسته‌بند استفاده می‌شود. این معیار نشان‌دهنده سطح زیر نمودار ROC[۲۶] است. نمودار ROC، روشی معمول برای ارزیابی کارایی دسته‌بندها است. عدد AUC یک مقدار بین ۰ و ۱ است که هر اندازه بزرگ‌تر باشد، کارایی نهایی دسته‌بند مطلوب‌تر ارزیابی می‌شود. نمودار ROC، منحنی دو بعدی است که محور Y آن نرخ مثبت صحیح (TPR)، و محور X آن نرخ مثبت کاذب (FPR) را نشان می‌دهد. در ادامه این بخش، در شکل (۱۲) نمودار ROC روش پیشنهادی نشان داده می‌شود که بیانگر کارایی مدل مخفی مارکوف ارائه‌شده در تشخیص باتنت است.

۴-۳- نتایج آزمایش‌ها

سیستم تشخیص باتنت پیشنهادی بر روی مجموعه داده‌های ۱ و ۲ که در جدول (۱۲) مشخصات آن‌ها بیان شد، مورد آزمایش قرار گرفت. شکل‌های (۶)، (۷) و (۸) به‌ترتیب نمودارهای نرخ مثبت صحیح، نرخ مثبت کاذب و نرخ منفی کاذب سیستم تشخیص باتنت پیشنهادی را برای تمام وقفه‌های زمانی بین ۵ تا ۱۵۰ ثانیه و با گام ۵ ثانیه، نشان می‌دهند. در شکل (۶)، نرخ مثبت صحیح بر روی مجموعه‌داده۱ در زمان ۵ ثانیه برابر با ۹۸/۶۶ درصد است، که در وقفه زمانی ۱۵ ثانیه برای اولین بار، مقدار آن به ۱۰۰ درصد می‌رسد. سپس با افزایش مدت زمان وقفه این میزان در بازه‌ی ۹۹/۹ و ۱۰۰ درصد تغییرات جزئی دارد. همچنین نرخ مثبت صحیح بر روی مجموعه‌داده۲ نیز درتمام وقفه‌های زمانی مطلوب بوده و مقدار آن بیش از ۹۹/۶۳ درصد است. با توجه به نمودار شکل (۶)، اکثر اوقات نرخ مثبت صحیح بر روی هر دو مجموعه‌داده برابر با ۱۰۰ درصد است. به‌طور کلی در هر دو مجموعه داده، با افزایش وقفه زمانی، میزان نرخ مثبت صحیح افزایش یافته و به ۱۰۰ درصد می‌رسد.

در شکل (۷) مشاهده می‌شود، با افزایش مدت زمان وقفه، میزان نرخ مثبت کاذب از وقفه زمانی ۵ ثانیه تا وقفه زمانی ۸۵ ثانیه در مجموعه‌داده۱ و ۹۰ ثانیه در مجموعه‌داده۲ تقریباً روندی کاهشی دارد. اما طبق نمودار، از این زمان‌ها به بعد، نرخ مثبت کاذب دچار برخی بی‌نظمی‌ها می‌شود. پیش از این نیز چنین نتایجی در [۱۵] مشاهده شده است. در ادامه این بخش، به‌طور مفصل دلیل این تغییرات با توجه به افزایش طول پنجره زمانی بیان می‌شود. در شکل (۸) مشاهده می‌شود که در هر دو مجموعه‌داده با افزایش طول وقفه زمانی، به‌طورکلی میزان نرخ منفی کاذب کاهش می‌یابد، و حتی در شماری از وقفه‌های زمانی به ۰ هم می‌رسد.

همان‌طور که در شکل‌های (۶)، (۷) و (۸) ملاحظه می‌شود، روند تغییرات نمودارها برای مجموعه‌داده۱ تا پنجره‌زمانی ۸۵ ثانیه مطلوب است، به‌طوری که در این پنجره زمانی نرخ مثبت صحیح ۱۰۰ درصد و نرخ مثبت کاذب ۰/۰۰۱ است. هر چند در اکثر پنجره‌های زمانی نیز نرخ مثبت صحیح ۱۰۰ درصد و نرخ مثبت کاذب پایین برقرار است. این روند تغییرات نمودارها برای مجموعه‌داده۲ نیز تا پنجره‌زمانی ۹۰ ثانیه مطلوب است به‌طوری که در این پنجره زمانی نرخ مثبت صحیح ۹۹/۹۹ درصد و نرخ مثبت کاذب ۰/۰۰۶ است. هرچند در سایر پنجره‌های زمانی نیز نتایج خوبی برای این مجموعه‌داده به‌دست آمد. به‌طوری که در پنجره زمانی ۴۵ ثانیه نیز، نرخ مثبت صحیح ۹۹/۹۹ درصد است و در پنجره‌های زمانی بزرگتر به ۱۰۰ درصد هم می‌رسد. با توجه به اینکه مجموعه‌داده۲ حجیم‌تر است، و هنگام آزمایش با آن از روش اعتبارسنجی متقابل ۱۰ لایه‌ای استفاده شده است. بنابراین براساس این مجموعه‌داده، پنجره زمانی ۹۰ ثانیه به‌عنوان مناسب‌ترین وقفه زمانی برای تشخیص روش پیشنهادی انتخاب می‌شود.

۴-۳-۱- بررسی میزان معیارهای TPAR و TPCR

یکی از نقاط قوت روش پیشنهادی این است که می‌تواند علاوه‌بر تشخیص باتنت، مرحله‌ی فعالیت آن را نیز شناسایی کند که در روش‌های تشخیص باتنت گذشته، تاکنون به این نکته مهم پرداخته نشده است. برای ارزیابی روش پیشنهادی براساس معیارهای جدید، از مجموعه‌داده۲ که حجیم‌تر و به دنیای واقعی نزدیک‌تر است، استفاده می‌شود. در شکل (۹)، نمودار نرخ مثبت صحیح حالت حمله (TPAR) و نرخ مثبت صحیح حالت فرمان و کنترل (TPCR)، نشان داده شده است. در تمام وقفه‌های زمانی، میزان نرخ مثبت صحیح تشخیص حالت C&C برابر با ۱۰۰ درصد است. دلیل اصلی این نتیجه مربوط به الگوهای محدود، خاص و منحصربه‌فرد رفتار بات در مرحله فرمان و کنترل است که در این پژوهش به‌درستی از مجموعه داده باتنت BlackEnergy استخراج شده و با نسبت احتمالات اولیه مناسب برای آموزش به الگوریتم بام‌ولش ارائه شده است.

از سوی دیگر، مرحله حمله باتنت می‌تواند الگوهای رفتاری بسیار متنوعی داشته باشد. بنابراین دقت تشخیص آن نسبت به مرحله C&C کمتر می‌شود. تشخیص دقیق‌تر مرحله حمله باتنت نیازمند استخراج الگوها با جزئیات ریزتر است که ممکن است پیچیدگی مدل و زمان تشخیص را افزایش دهد. همان‌طور که در شکل (۹) ملاحظه می‌شود، در تمام آزمایش‌ها میزان نرخ مثبت صحیح تشخیص حالت حمله (TPAR) بیش از ۹۱/۷ درصد است، به‌طوریکه در پنجره‌های زمانی بزرگتر از ۸۵ ثانیه به بیش از ۹۵ درصد هم می‌رسد. باتوجه به شکل (۹)، میزان نرخ مثبت صحیح تشخیص حالت حمله با افزایش طول وقفه زمانی، در مجموع روندی منظم و افزایشی دارد. این روند تغییرات نمودار TPAR بسیار مشابه با روند تغییرات نرخ مثبت صحیح مجموعه‌داده۲ در شکل (۶) است. این قضیه نشان می‌دهد که با افزایش طول پنجره زمانی دقت تشخیص مدل نیز افزایش می‌یابد.

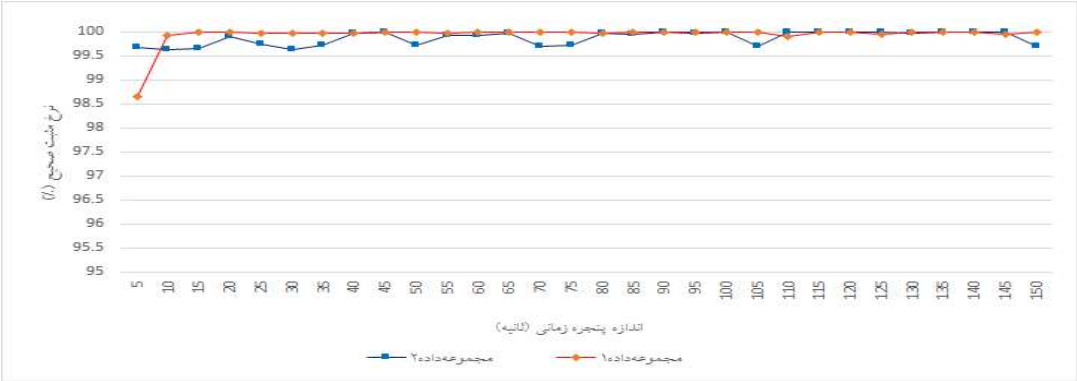
۴-۳-۲- بررسی تأثیر اندازه پنجره زمانی در نتایج آزمایش

در این قسمت سعی شده است که تأثیر اندازه پنجره زمانی یا وقفه زمانی به‌صورت دقیق‌تر و واضح‌تر بر روی آزمایش‌ها مشاهده و بررسی شود. بدین منظور میانگین نرخ‌های مثبت صحیح و مثبت کاذب هر دو مجموعه‌داده در تمام پنجره‌های زمانی از ۵ تا ۱۵۰ ثانیه محاسبه گردید. سپس شاخص‌های زمانی ۵، ۳۰، ۶۰، ۹۰، ۱۲۰، ۱۵۰ ثانیه تعیین شد. این شاخص‌ها براساس تحلیل و بررسی میزان تغییرات نتایج انتخاب شده‌اند، تا اینکه بتوانند نماینده خوبی برای ارزیابی اثر اندازه پنجره زمانی بر روی نتایج تشخیص مدل پیشنهادی باشند. همان‌طور که در شکل (۱۰) ملاحظه می‌شود، طبق انتظار، با افزایش اندازه پنجره زمانی، به‌دلیل افزایش حجم داده‌های در دسترس برای تصمیم‌گیری، میزان نرخ مثبت صحیح روند منظم افزایشی دارد.

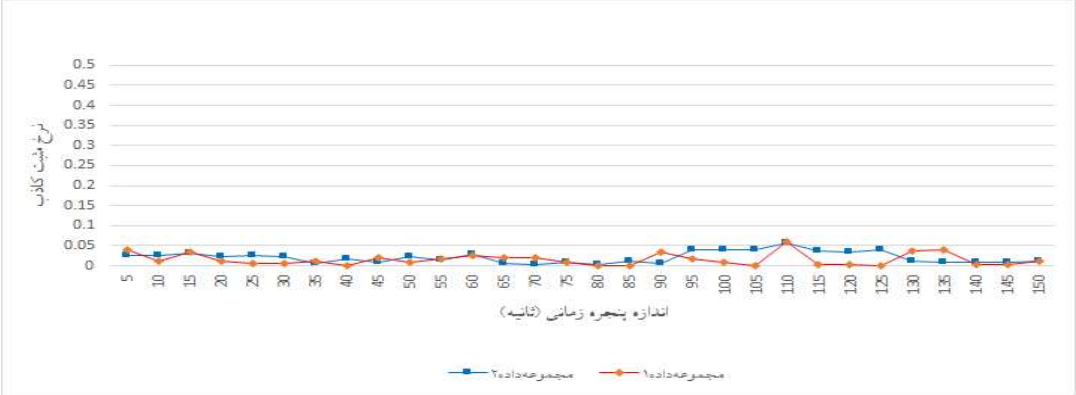
از سوی دیگر، مطابق شکل (۱۱)، با افزایش اندازه پنجره زمانی از ۵ تا ۹۰ ثانیه، میزان تغییرات نرخ مثبت کاذب یا نرخ هشدار نادرست روندی کاهشی دارد. ولی از ۹۰ ثانیه تا ۱۲۰ ثانیه افزایش نامطلوبی در نرخ مثبت کاذب مشاهده می‌شود. این بی‌نظمی در نمودار شکل (۷) نیز قابل‌مشاهده است. این مشاهدات نمایان‌گر آن است که باتنت‌ها الگوهای منحصربه‌فردی را طی می‌کنند، به‌طوریکه برخی از پنجره‌های زمانی بیشترین شباهت را به زمان چرخه باتنت دارند و برخی دیگر کمترین. براساس این واقعیت، در سال ۲۰۱۶، مقاله [۱۵] نتایجی مشابه با این پژوهش در تغییرات میزان نرخ مثبت کاذب ارائه کرده است. از طرفی، در وقفه‌های زمانی بزرگ به‌دلیل ترافیک متنوع و حجیم مجموعه‌داده LBNL، احتمال تشابه رفتار جریان‌های نرمال با فعالیت بات افزایش می‌یابد. دلایل فوق، باعث افزایش نرخ مثبت کاذب در بدترین شرایط به مقدار بیش از ۰/۰۳ برای مجموعه‌داده۱ و ۰/۰۵ برای مجموعه‌داده۲ می‌شود. البته همانگونه که در شکل (۱۱) مشاهده می‌شود به‌مرور با افزایش اندازه پنجره زمانی دوباره میزان نرخ مثبت کاذب کاهش می‌یابد.

۴-۳-۳- بررسی میزان معیار AUC

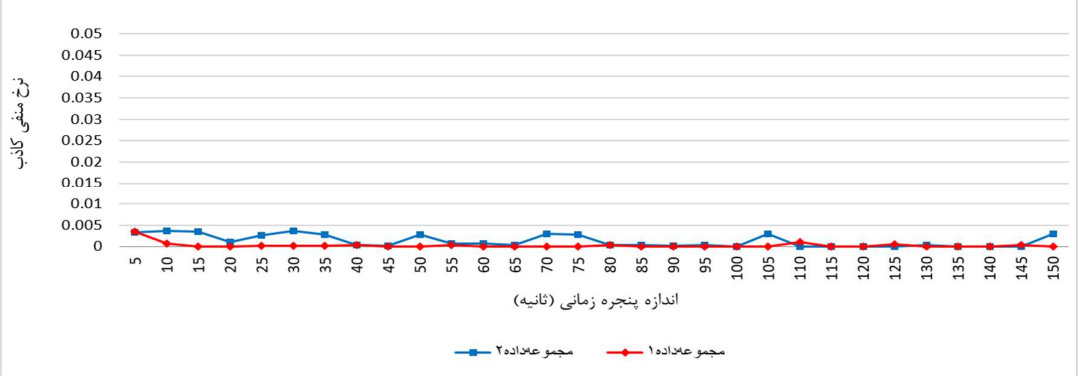
براساس نتایج آزمایش‌ها، روش تشخیص باتنت ارائه‌شده، در تمام وقفه‌های زمانی، دقت تشخیص بالا و نرخ مثبت کاذب پایین دارد. برای ارزیابی بیشتر این روش، نمودار ROC آن بر اساس جابجایی حد آستانه‌ی ویژگی‌های تعداد بسته‌های مبادله‌شده در واحد زمان و اندازه طول بسته‌ها، ترسیم می‌شود. منحنی ROC روش پیشنهادی در شکل (۱۲) نشان داده شده است. همانطور که در این شکل ملاحظه می‌شود، میزان معیار AUC در این روش برابر با ۰/۹۸۳۶۵ است که بسیار مطلوب و نزدیک به ۱ است.



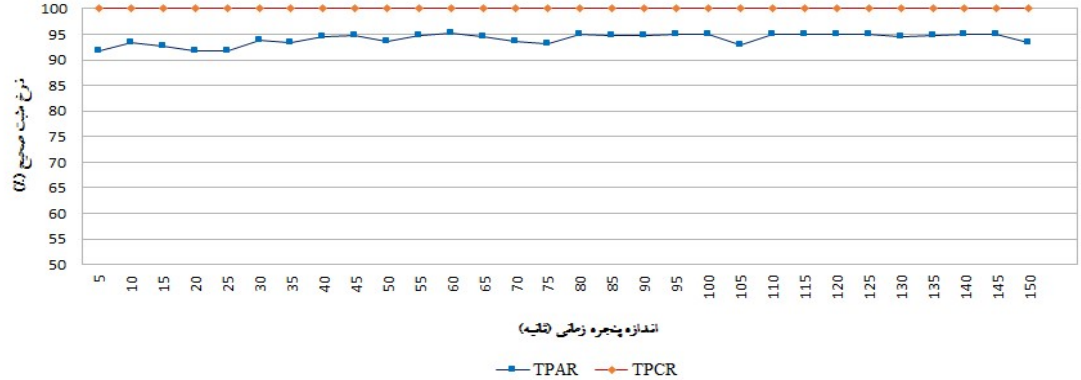
شکل ۶: نرخ مثبت صحیح مجموعه‌داده‌ها در وقفه‌های زمانی مختلف



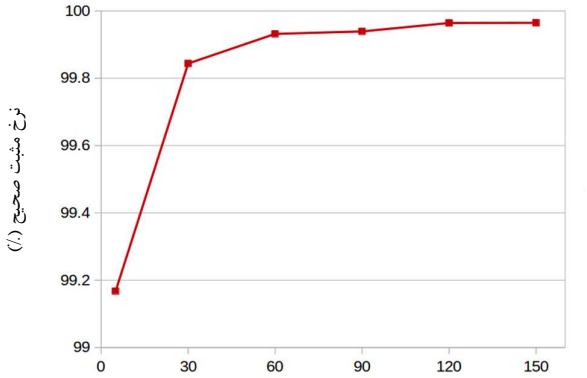
شکل ۷: نرخ مثبت کاذب مجموعه‌داده‌ها در وقفه‌های زمانی مختلف



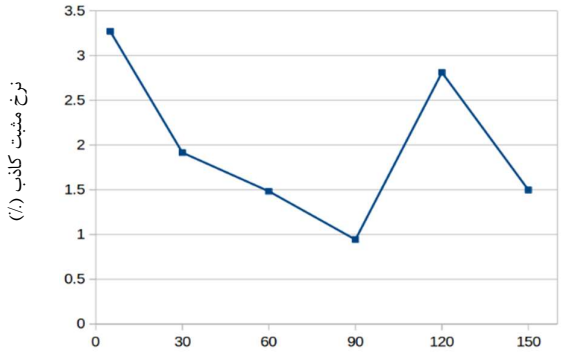
شکل ۸: نرخ منفی کاذب مجموعه‌داده‌ها در وقفه‌های زمانی مختلف



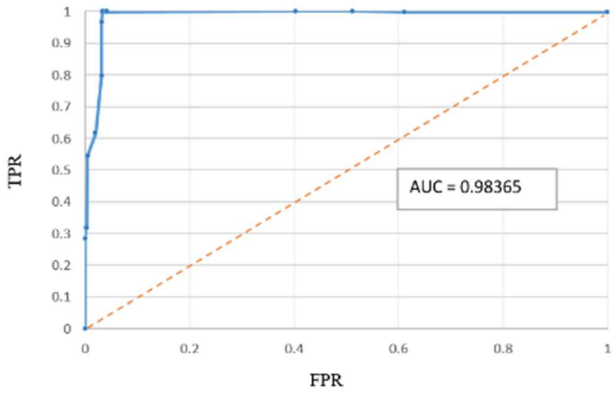
شکل ۹: نرخ مثبت صحیح حالت حمله (TPAR) و و نرخ مثبت صحیح حالت فرمان و کنترل (TPCR)



شکل ۱۰: میانگین نرخ مثبت صحیح مجموعه‌داده‌ها در پنجره‌های زمانی شاخص



شکل ۱۱: میانگین نرخ مثبت کاذب مجموعه‌داده‌ها در پنجره‌های زمانی شاخص



شکل ۱۲: منحنی ROC روش تشخیص باتنت پیشنهادی

جدول ۱۳: مقایسه روش پیشنهادی با سایر روش‌های مبتنی‌بر وقفه‌های جریان

روش تشخیص	سطح تشخیص	مرحله تشخیص	مدت زمان وقفه	نرخ مثبت صحیح (٪)	نرخ مثبت کاذب (٪)
وانگ و همکاران [۱۳]	انفرادی	C&C	۱۰ دقیقه	۹۸/۱۱	۳۰/۱۹
ژائو و همکاران [۶]	انفرادی	C&C - حمله	۳۰۰ ثانیه	۹۸/۱	۲/۱
کیروبواتی و آنیتا [۱۵]	انفرادی	C&C - حمله	۱۸۰ ثانیه	۹۶/۱	۴/۸۱
روش پیشنهادی	انفرادی	C&C - حمله	۴۵ ثانیه	۹۹/۹۹	۰/۸۷

جدول ۱۴: مقایسه روش پیشنهادی با سایر روش‌های مبتنی‌بر مدل مخفی مارکوف

روش تشخیص	پروتکل تشخیص	مرحله تشخیص	تشخیص بلادرنگ	نرخ مثبت صحیح مرحلهC&C (٪)	نرخ مثبت صحیح (٪)	نرخ مثبت کاذب (٪)
لو و بروکس [۲۱]	HTTP	C&C - حمله	×	مشخص نشده	۹۵	۲
کیم و همکاران [۲۲]	HTTP- IRC	C&C - حمله	×	۳۰/۶	۹۵ - ۹۴	۱۹ -۱۰
روش پیشنهادی	HTTP	C&C - حمله	√	۱۰۰	۹۹/۹۹	۰/۸۷

۴-۴- مقایسه‌ها

در این بخش، روش پیشنهادی بر اساس برخی از ویژگی‌ها با تعدادی از روش‌های تشخیص بات‌نت مبتنی‌بر وقفه‌های جریان مقایسه می‌شود. دلیل انتخاب این روش‌ها برای مقایسه با روش پیشنهادی، برتری‌های آن‌ها از لحاظ سرعت و دقت تشخیص نسبت به اکثر پژوهش‌های پیشین است. روش‌های گذشته‌ی مبتنی‌بر بازرسی بسته‌های شبکه، به‌دلیل نیاز به زمان پردازش بیشتر، سرعت تشخیص پایینی دارند. این روش‌ها در برابر کانال‌های C&C رمزنگاری شده ناتوان هستند. برای غلبه بر این مشکلات، روش‌های مبتنی‌بر تحلیل جریان شبکه معرفی شدند. اکثر این روش‌ها به‌دلیل اینکه کل جریان را تحلیل می‌کنند، بلادرنگ نیستند [۶]. در سال‌های اخیر، روش‌های مبتنی‌بر تحلیل جریان شبکه در وقفه‌های جریان ارائه شدند. این روش‌ها قادر به تشخیص بات‌نت‌ها در کمترین زمان و در مراحل اولیه از چرخه حیات آن‌ها هستند. نتایج مقایسه روش پیشنهادی با روش‌های تشخیص بات‌نت مبتنی‌بر تحلیل جریان شبکه در وقفه‌های جریان، در جدول (۱۳) نشان داده شده است.

همانطور که در جدول (۱۳) مشاهده می‌شود، روش‌های گذشته مبتنی‌بر وقفه جریان، در بهترین حالت، در وقفه زمانی ۱۸۰ ثانیه قادر به تشخیص بات‌نت با دقت مطلوب هستند [۱۵]. در صورتی‌که روش ارائه‌شده در این پژوهش، می‌تواند بات‌ها را در وقفه زمانی ۵ ثانیه با دقت بالایی تشخیص دهد. در جدول (۱۳)، وقفه زمانی ۴۵ ثانیه به‌عنوان شاخص مقایسه انتخاب می‌شود. زیرا ۴۵ ثانیه، کمترین وقفه‌ی زمانی است که در هر دو آزمایش مجموعه داده۱ (نرخ مثبت صحیح ۱۰۰ درصد و نرخ مثبت کاذب ۱/۸۶ درصد) و مجموعه داده۲ (نرخ مثبت صحیح ۹۹/۹۹ و نرخ مثبت کاذب ۰/۸۷ درصد) بهترین مقدار به‌دست می‌آید. نکته قابل‌توجه در مقایسه فوق این است که مجموعه‌داده نرمال استفاده شده [۵]، در روش‌های [۶، ۱۵] و روش پیشنهادی یکسان است. این امر می‌تواند تا حدودی اعتبار این مقایسه را نشان دهد. روش‌های ارائه‌شده در جدول مقایسه قادر هستند با تحلیل سطح انفرادی، برخلاف روش‌های مبتنی‌بر تحلیل سطح گروهی، در صورت وجود حتی یک بات در شبکه، آن را تشخیص دهند.

با وجود اینکه در تعداد اندکی از تحقیقات گذشته، از مدل مخفی مارکوف برای تشخیص بات‌نت استفاده شده است، اما فقط در روش پیشنهادی ویژگی‌های مختلف بات‌نت برای پیاده‌سازی مدل در نظر گرفته شده است و تنها به یک ویژگی زمان دریافت بسته‌های شبکه [۲۱] یا ویژگی اسکن پورت‌ها [۲۲، ۲۵]، اکتفا نشده است. که این امر باعث افزایش قدرت تشخیص مدل مخفی مارکوف پیشنهادی می‌شود. در جدول (۱۴)، روش پیشنهادی براساس برخی از ویژگی‌ها با روش‌های [۲۱، ۲۲] مقایسه می‌شود. همچنین باتوجه به ساختار ساده‌ی مدل پیشنهادی، برخلاف مدل [۲۱]، دارای قابلیت تعمیم جهت تشخیص انواع مختلف بات‌نت است. مدل [۲۱] به‌دلیل ساختار ریاضی پیچیده، زمان یادگیری و تشخیص طولانی، برای تشخیص بات‌نت‌های P2P مناسب نیست [۲۶].

۵- نتیجه‌گیری

در این مقاله، یک سیستم تشخیص بات‌نت مبتنی‌بر مدل مخفی مارکوف ارائه شد که قادر است بات‌ها را در وقفه‌های کوتاه جریان تشخیص دهد. مهم‌ترین مشخصات این سیستم عبارتند از: تشخیص بلادرنگ و سریع نسبت به روش‌های پیشین، تشخیص در مراحل اولیه چرخه حیات بات‌نت، تشخیص بات‌نت‌های مبتنی‌بر کانال‌های C&C رمزنگاری‌شده و تشخیص با دقت بالا و نرخ مثبت کاذب پایین. از سویی دیگر، روش‌هایی که بات‌نت‌ها را در مراحل مختلف چرخه حیاتش تشخیص می‌دهند، قادر به شناسایی مرحله فعالیت آن نیستند. ولی روش ارائه‌شده در این

پژوهش می‌تواند علاوه‌بر شناسایی باتنت، مرحله چرخه حیات آن (فرمان و کنترل یا حمله) را نیز با دقت بالایی تعیین کند.

در مدل مخفی مارکوف پیشنهادی، علاوه‌بر الگوهای رفتاری متفاوت باتنت در مراحل مختلف چرخه حیاتش، ویژگی‌های شاخص باتنت نیز در نظر گرفته می‌شود. درنتیجه، مدل پیشنهادی قادر به شناسایی باتنت‌هایی می‌شود که تاکنون الگوی رفتاری آن‌ها را مشاهده نکرده است. برخلاف روش پیشنهادی، اکثر روش‌های تشخیص نفوذ مبتنی‌بر مدل مخفی مارکوف، نیاز به زمان بالاتری برای یادگیری دارند [۲۱، ۲۶]. در این روش‌ها، در آغاز یادگیری تعداد حالت‌ها مشخص نیست. بنابراین برای ساخت مدل، مدل‌های مختلف با تعداد حالات متفاوت ساخته می‌شود، سپس با محاسبه‌ی نرخ مثبت کاذب و مثبت صحیح این مدل‌ها، بهترین آن‌ها انتخاب می‌شود. بدین ترتیب، ساخت تعداد زیادی مدل نیاز به زمان زیادی دارد.

باتنت BlackEnergy یکی از رایج‌ترین و زیان‌بارترین باتنت‌ها است. بدین منظور در این پژوهش رفتار ترافیک شبکه آن مورد تحلیل و بررسی قرار گرفت. در نهایت، سیستم تشخیص باتنت پیشنهادی براساس ویژگی‌ها و الگوهای رفتار ترافیکی باتنت BlackEnergy پیاده‌سازی شده است. برای تشخیص این باتنت، طبق نتایج آزمایش‌ها، نرخ مثبت صحیح درحدود ۱۰۰ درصد و نرخ مثبت کاذب کمتر از ۱ درصد برای اکثر وقفه‌های زمانی کوچکتر از ۱۵۰ ثانیه، به‌دست آمد. در این پژوهش، مهم‌ترین رویکرد آینده این است که مدل مخفی مارکوف پیشنهادی برای شناسایی انواع مختلف باتنت‌ها تعمیم داده شود. همچنین برای تسریع و تسهیل استخراج الگوهای رفتار ترافیکی باتنت‌ها از روش‌های یادگیری ماشین استفاده شود.

مراجع

[1] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in Internet Measurement Conference, pp. 41-52, 2006.

[2] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in Emerging Security Information, Systems, and Technologies Conference, pp. 268-273, Greece, 2009.

[3] J. Leonard, S. Xu, and R. Sandhu, "A framework for understanding botnets," in Availability, Reliability, and Security Conference, pp. 917-922, 2009.

[4] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A survey of botnet technology and defenses," in Cybersecurity Applications and Technology Conference, pp. 299-304, USA, 2009.

[5] Lawrence Berkeley National Laboratory and ICSI, LBNL/ICSI enterprise tracing project, LBNL enterprise trace repository, http://www.icir.org/enterprise-tracing/2016-11-08.

[6] D. Zhao, I. Traore, B. Sayed, W. Lu, Sh. Saad, A. Ghorbani, and D. Garant, "Botnet detection based on traffic behavior analysis and flow intervals," Computers & Security, vol. 39, Part A, pp. 2-16, 2013.

[7] J. Nazario, "BlackEnergy DDoS bot analysis," Technical report, Arbor networks, 2007.

[8] F-secure labs security response, BLACKENERGY amd QUEDAGH, the convergence of crimeware and APT attacks, Malware analysis white paper, 2014.

[9] W. Chang, A. Mohaisen, A. Wang, and S. Chen, "Measuring botnets in the wild: some new trends," in Information, Computer and Communications Security Conference, pp. 645-650, 2015.

[10] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Using machine learning technliques to identify botnet traffic", in Local Computer Networks Conference, pp. 967-974, 2006.

[11] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection," in Security Symposium, pp. 139-154, 2008.

[12] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, and W. Lu, "Detecting P2P botnets through network behavior analysis and machine learning," in Privacy, Security and Trust Conference, pp. 174-180, 2011.

[13] B. Wang, Z. Li, H. Tu, and J. MaWang, "Measuring peer-to-peer botnets using control flow stability," in Availability, Reliability and Security Conference, pp. 663-669, 2009.

[14] H. R. Zeidanloo and S. Rouhani, Botnet detection by monitoring common network behaviors, Lambert Academic Publishing, 2012.

[15] G. Kirubavathi and R. Anitha, "Botnet detection via mining of traffic flow characteristics," Computers and Electrical Engineering, vol. 50, pp. 91-101, 2016.

[16] B. K. Sin, J. Y. Ha, S. C. Oh, and J. H. Kim, "Network-based approach to online cursive script recognition," IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 29, no. 2, pp. 321-328, 1999.

[۱۷] مسعود فرکی و مازیار پالهنگ، «بازشناسی برخط حروف فارسی بر پایه مدل مخفی مارکوف»، مجله مهندسی برق دانشگاه تبریز، دوره ۴۰، شماره ۱، صفحه ۲۳- ۳۴، ۱۳۸۹.

[18] K. Pulasinghe, K. Watanabe, K. Izumi, and K. Kiguchi, "Modular fuzzy-neuro controller driven by spoken language commands," IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 34, no. 1, pp.293-302, 2004.

[19] T. Starner and A. Pentland. "Visual recognition of american sign language using hidden markov models," Perceptual Computing Section, the Media Laboratory, Massachusetts Institute of Technology, Technical Report, 1995.

[۲۰] سیامک عبداله‌زاده، محمدعلی بالافر و لیلی محمدخانلی، «استفاده از خوشه‌بندی و مدل مارکوف جهت پیش‌بینی درخواست آتی کاربر در وب»، مجله مهندسی برق دانشگاه تبریز، دوره ۴۵، شماره ۳، صفحه ۸۹- ۹۶، ۱۳۹۴.

[21] C. Lu and R. Brooks, "Botnet traffic detection using hidden markov models," in Cyber Security and Information Intelligence Research Conference, pp. 31-34, 201.

[22] D. H. Kim, T. Lee, J. Kang, H. Jeong, and H. Peter, "Adaptive pattern mining model for early detection of botnet-propagation scale," Security and Communicaion Networks, vol. 5, no. 8, pp. 917-927, 2012.

[23] Z. Abaid, D. Sarkar, M. A. Kaafar, and S. Jha, "The early bird gets the Botnet: A Markov chain based early warning system for botnet attacks," in Local Computer Networks Conference, pp. 61-68, 2016.

[24] A. Garivier, "The Baum-Welch algorithm for hidden Markov models: speed comparison between octave/python/R /scilab/matlab/C/C++," http://www.math.univ-toulouse.fr/~agariv ie/Telecom/code/index.php/2017-09-15.

[25] W. Gobel, "Detecting botnets using hidden Markov models on network traces," White paper, 2008.

[26] C. Lu and R. R. Brooks, "P2P hierarchical botnet traffic detection using hidden Markov models," in Learning from Authoritative Security Experiment Results Conference, pp. 41-46, 2012.

زیرنویس‌ها

[1] Botnet
[2] Command and control
[3] Bot master
[4] Distributed denial of service
[5] Hidden Markov Model
[6] Lawrence Berkeley National Lab (LBNL)
[7] True Positive Rate (TPR)
[8] False Positive Rate (FPR)
[9] Machine learning
[10] Support vector machine (SVM)
[11] Artificial neural network
[12] Nearest neighbor
[13] REP Tree
[14] Boosted Decision Tree
[15] Naïve Bayes
[16] SCADA
[17] Markov chain
[18] States
[19] Observations
[20] Baum-Welch
[21] Viterbi
[22] Maximization
[23] Expectation
[24] 10-fold cross validation
[25] Area Under Curve
[26] Receiver Operating Characteristic