مجله علمی - پژوهشی "پدافند الکترونیکی و سایبری"
سال سوم، شماره ۴، زمستان ۱۳۹۴، ص ۶۰- ۴۳

تشخیص بات‌نت برای شبکه‌های نظیربه‌نظیر

رضا عزمی[۱]، مریم قلی‌نژاد[۲*]، محسن صابری[۳]

۱- استادیار، گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه الزهرا تهران

۲- دانشجو کارشناسی ارشد، گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه الزهرا تهران

۳- مربی، گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه بزرگمهر قائنات

(دریافت:۹۴/۴/۱۶، پذیرش: ۹۴/۱۰/۲۲)

چکیده

بات‌نت‌ها از جمله جدیدترین نوع بدافزارها در مقیاس اینترنت می‌باشند که در سال‌های اخیر بیش‌ترین تهدیدات را متوجه سامانه‌های اینترنتی نموده‌اند. بات، رایانه‌ای آلوده‌شده به یک بدافزار است که بدون آگاهی و اراده کاربر و از راه دور توسط یک یا چند عامل انسانی کنترل می‌شود. به این عامل کنترل‌کننده، سرکرده یا مدیربات گویند و گاهی سیستم آلوده را قربانی نیز می‌نامند. بات‌نت نظیربه‌نظیر یکی از انواع بات‌نت است که از پروتکل‌های نظیربه‌نظیر برای کنترل بات‌های خود استفاده می‌کند و شناسایی این نوع از بات‌نت نسبت به انواع دیگر مشکل‌تر است. رویکرد پیشنهادی ما یک راه‌حل برای شناسایی چنین بات‌نت‌هایی است. این رویکرد با استفاده از تحلیل جریان شبکه و روش‌های خوشه‌بندی در داده‌کاوی، بات‌نت‌های نظیربه‌نظیر را شناسایی می‌کند. رویکرد ارائه‌شده، مبتنی بر جریان بوده و با مقایسه شباهت بین جریان‌ها و خوشه‌بندی جریان‌های مقایسه‌شده با استفاده از الگوریتم K-Means و در نهایت مقایسه خوشه‌های ترافیک جدید با خوشه‌های قبلی، می‌تواند وجود یا عدم وجود حمله را تشخیص دهد. به این ترتیب که ابتدا یک ترافیک از جنس حمله و یک ترافیک معمولی خوشه‌بندی می‌شود، سپس ترافیک جدید که نوع آن مشخص نیست خوشه‌بندی می‌شود. درنهایت خوشه‌بندی ترافیک جدید با دو ترافیک قبلی مقایسه شده و مشخص می‌شود که ترافیک جدید از چه نوعی است. رویکرد ارائه‌شده برای تشخیص بات‌نت از ترافیک ازدحام ناگهانی، عملکرد خوبی را از خود نشان می‌دهد و این خصوصیت، وجه تمایز الگوریتم پیشنهادی، نسبت به الگوریتم‌های مشابه آن است. درنهایت عملکرد رویکرد ارائه‌شده، با ترافیک‌های مختلف موردبررسی قرارگرفته می‌شود.

واژه‌های کلیدی: بات‌نت، شناسایی بات‌نت، بات، بات‌نت نظیربه‌نظیر، تحلیل جریان شبکه

۱- مقدمه

بات‌نت‌ها[۱] ازجمله جدیدترین نوع بدافزارها[۲] در مقیاس اینترنت می‌باشند که در سال‌های اخیر بیش‌ترین تهدیدات را متوجه سامانه‌های اینترنتی نموده‌اند. بات، رایانه‌ای آلوده‌شده به یک بدافزار است که بدون آگاهی و اراده کاربر و از راه دور توسط یک یا چند عامل انسانی کنترل می‌شود. به این عامل کنترل‌کننده، سرکرده یا مدیربات[۳] گویند و گاهی سیستم آلوده را قربانی[۴] نیز می‌نامند.

بات‌نت‌ها اهداف مخرب زیادی دارند و اخیراً مهاجمین، از بات‌نت‌های نظیربه‌نظیر[۵] برای راه‌اندازی انواع حملات مانند حملات DDOS، پخش بدافزار، توزیع هرزنامه‌ها و غیره، استفاده می‌کنند. بات‌نت‌های نظیربه‌نظیر نسل جدیدی از بات‌نت‌ها هستند که از شبکه‌های نظیربه‌نظیر برای کنترل از راه دور بات‌های خود استفاده می‌کنند. در این نوع از بات‌نت‌ها تمامی بات‌ها می‌توانند نقش کارگزار[۶] فرمان- کنترل[۷] را بازی کنند و مهاجمین می‌توانند حملات خود را از هر رایانه در بات‌نت نظیربه‌نظیر، راه‌اندازی کنند؛ بنابراین شناسایی آن‌ها در مقایسه با دیگر بات‌نت‌ها مشکل‌تر است.

ویژگی منحصربه‌فردی که یک بات‌نت را از دیگر بدافزارها متمایز می‌سازد، زیرساخت ارتباطی آن یعنی کانال فرمان- کنترل است. مدیر بات از کانال فرمان- کنترل برای دستوردادن به بات‌ها به‌منظور انجام فعالیت‌های مخرب استفاده می‌کند [۱].

مدیران بات برای نگه‌داشتن بات‌نت خود از فن‌های غیرقانونی زیادی استفاده می‌کنند از جمله پنهان‌سازی کد[۸]، رمزنگاری حافظه[۹]، فنّاوری پیاده‌سازی نظیربه‌نظیر و یا تقلید ازدحام ناگهانی[۱۰]. در اینجا تمرکز ما روی حمله DDOS ای است که با استفاده از تقلید ازدحام ناگهانی صورت می‌گیرد. حمله DDOS

*ایمیل نویسنده پاسخگو: M.gholinezhad84@gmail.com

1- Botnet
2- Malware
3- Bot Master
4- Victim
5- Peer to Peer
6- Server
7- Command and Control Server
8- Code Obfuscation
9- Memory Encryption
10- Flash Crowd Mimicking

یک حمله بحرانی برای اینترنت است و اغلب، بات‌نت‌ها در پشت این حملات قرار دارند. مدیران بات ماهر، با تقلید الگوی ترافیک ازدحام ناگهانی، فنّاوری تشخیص را غیرفعال می‌کنند؛ که این مسئله باعث می‌شود برای کسانی که به‌دنبال دفع حملات بات‌نت هستند یک چالش عظیم ایجاد شود. نکته‌ای که وجود دارد این است که جریان‌های حمله جاری در مقایسه با جریان ازدحام ناگهانی، شباهت بیش‌تری به هم دارند؛ و طبق همین اصل، الگوریتمی ارائه‌شده که از ضریب هم‌بستگی جریان، به‌عنوان یک پارامتر شباهت برای جریان‌های مشکوک استفاده می‌شود.

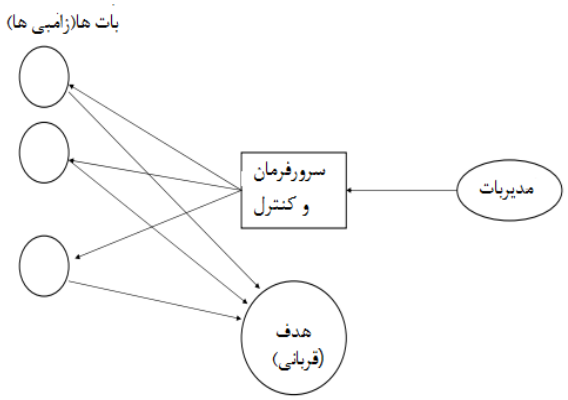
ازدحام ناگهانی، یک اتفاق ناگهانی و ناخواسته اما درعین‌حال مشروع است، مثل هجوم ناگهانی برای خواندن یک خبر جدید. یک راه‌حل خوب برای رخنه‌گرها[1] شبیه‌شدن به الگوی ازدحام ناگهانی است تا از دید آشکار کننده‌ها مخفی بماند که به این نوع، حمله ازدحام ناگهانی[2] می‌گویند.

پژوهش‌های دیگری قبلاً صورت گرفته که تشخیص حمله DDOS را با استفاده از مشخص‌کردن و استخراج ویژگی‌های حمله DDOS انجام می‌دهند، چنین رویکردهایی به‌صورت فعال قادر به شناسایی حملات نیستند. راه‌حل‌های دیگری برای مقابله با چنین حملاتی استفاده می‌شود که در آن از جورچین‌های[3] گرافیکی برای تفاوت قائل‌شدن بین انسان و بات‌ها استفاده می‌کنند. بدی این روش این است که استفاده از پاسخ‌های انسانی، می‌تواند برای کاربر آزاردهنده باشد. استفاده از CAPTCHA و رفتارهای انسانی در لایه کاربرد، خوب عمل می‌کند. با این حال؛ هیچ روش تشخیصی در لایه شبکه مشاهده نشده که بتواند سیستم دفاعی را توسعه دهد. یک تعداد پژوهش‌هایی انجام شده [۴-۲] که در آن‌ها با استفاده از هانی پات، بات‌ها به دام انداخته شده‌اند و روی آن‌ها تحلیل انجام‌داده‌شده تا فعالیت‌های بات‌نت را مورد مطالعه قرار دهند.

بات‌ها از سه عنصر بات، کارگزار فرمان- کنترل و مدیر بات تشکیل شده‌اند و تهدیدات آن‌ها نیز توسط این عناصر سازماندهی می‌شود. این عناصر به‌شرح زیر است [۶-۵]

- بات، یک برنامه نرم‌افزاری است که برروی میزبان‌های آسیب پذیر نصب شده و قادر است اقدامات مخربی را انجام دهد. پس از این‌که برنامه بات برروی رایانه نصب گردید، آن رایانه به یک بات یا زامبی[4] تبدیل می‌شود.
- مدیربات یا هدایت‌کننده بات، شخص یا گروهی از اشخاص هستند که با ارسال دستورات خود بات‌ها را، از راه دور کنترل می‌کنند تا فعالیت‌های غیرقانونی یا مخرب انجام دهند.
- هسته ارتباطی بات‌ها، کارسازی به‌نام "فرمان- کنترل" است که بات‌ها فرمان‌های اجرایی خود را از آن دریافت می‌نمایند. کارگزار فرمان- کنترل، دستورات مدیران بات را برای بات‌ها در رایانه‌های آلوده‌شده توزیع می‌کند یا به‌عبارتی دستورات را از مدیر بات دریافت و برای بات‌ها ارسال می‌کند. البته همچنان‌که در بخش‌های بعدی به آن پرداخته خواهد شد، این معماری تنها معماری مرسوم در بات‌نت‌ها نیست ولی عمومی‌ترین معماری است.

عناصر بات‌نت در شکل (۱) نشان داده شده است:



شکل (۱). عناصر یک بات‌نت

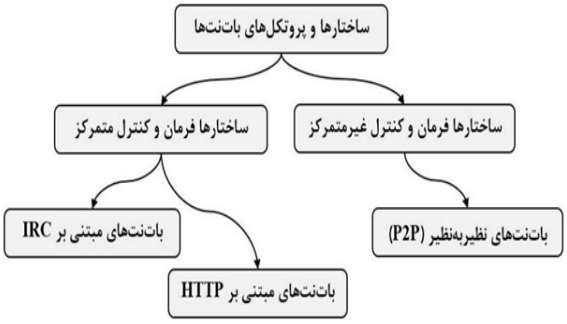
عملیات بات‌نت به‌وسیله ارسال دستورات خاصی از سمت مدیر بات و از طریق کانال فرمان- کنترل آغاز می‌شود. عناصر یک بات‌نت ارتباط نزدیکی باهم دارند و برای این‌که حمله‌ای بخواهد اتفاق بیفتد، زیرساخت ارتباطی یعنی کانال فرمان- کنترل، نیاز خواهد بود. زیرساخت فرمان کنترل[5] مهم‌ترین مؤلفه بات‌نت و وجه تمایز آن‌ها با دیگر بدافزارهاست که اجازه می‌دهد بات‌ها، دستورات مخرب را از جانب مدیر بات دریافت کنند و امکان هدایت و کنترل بات‌ها را درون یک بات‌نت برای مدیر بات فراهم می‌کند [۷].

۲- انواع معماری بات‌نت‌ها

بات‌نت‌ها با توجه به سازوکاری که برای فرمان- کنترل خود استفاده می‌کنند از دو معماری کلی استفاده می‌کنند. معماری متمرکز و نامتمرکز.

1- Hacker
2- Flash Crowd Attack
3- Puzzle
4- Zombie

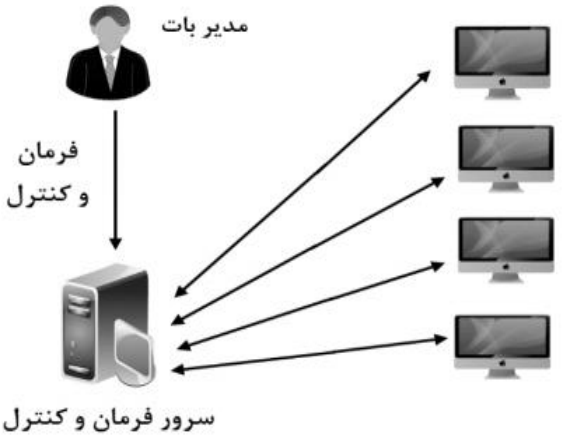
5- Command and Control Infrastructure



شکل (۲). ساختارها و پروتکل‌های بات‌نت‌ها

۲-۱- معماری متمرکز

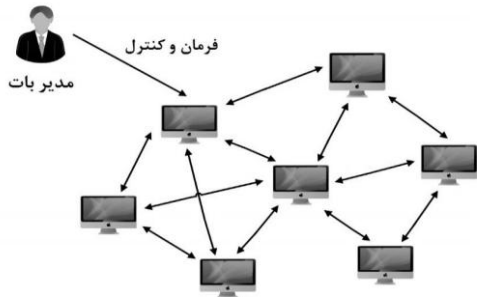
در معماری متمرکز[1] که عمومی‌ترین معماری نیز است، یک یا چند کارگزار فرمان- کنترل وجود دارد و تمام بات‌ها دستورات خود را از این کارگزار دریافت می‌کنند. حسن معماری متمرکز پیاده‌سازی آسان و امکان آرایش‌دهی سریع بات‌ها است به این معنی که با ارسال دستورات در مدت‌زمان کوتاهی می‌توان بات‌ها را برای هدف خاصی آرایش داد. همچنین بازخورد مستقیم، مدیر بات را قادر می‌سازد به‌راحتی وضعیت بات‌نت را بررسی کند و به اطلاعات مهمی درباره بات‌نت از قبیل تعداد بات‌های فعال، دست یابد [۶]. عیب اساسی این روش این است که اگر به‌دلیلی کارگزار فرمان- کنترل از دسترس خارج شود، عملاً بات‌نت از کار می‌افتد. دوم این‌که این روش به‌راحتی قابل‌شناسایی است. زیرا دستورات مشابه‌دادن به یک تعداد گره در شبکه و یا اتصال چندین گره از یک شبکه به یک سیستم خاص، شک دستگاه‌های تشخیص نفوذ[2] (IDS) را برخواهد انگیخت.



شکل (۳). ساختار فرمان- کنترل متمرکز

۲-۲- معماری غیرمتمرکز

در معماری نامتمرکز بات‌ها به‌صورت یک شبکه نظیربه‌نظیر باهم در تعامل می‌باشند. سرکرده دستورات خود را به یک یا چند بات ارسال می‌کند و با استفاده از قراردادهای نظیربه‌نظیر این دستورات در تمام بات‌نت منتشر می‌شود. ویژگی اصلی بات‌نت‌های نظیربه‌نظیر این است که تمام همتاها می‌توانند نقش یک کارگزار فرمان- کنترل را بازی کنند [۸] به‌عبارت‌دیگر در بات‌نت‌های نظیربه‌نظیر کارگزار فرمان- کنترل پنهان است [۹]. حسن این روش این است که بات‌ها به کارگزار فرمان- کنترل وابسته نیستند. توزیع ترافیک در این بات‌نت‌ها شناسایی آن‌ها را دشوار کرده است. این بات‌نت‌ها مقیاس‌پذیری بالایی دارند اما پیاده‌سازی آن‌ها پیچیده و دشوار است. بات‌ها به‌خاطر تأخیر در انتشار دستورات نمی‌توانند به‌سرعت آرایش بگیرند. حالت خاصی هم از بات‌نت‌های نظیربه‌نظیر موجود است که در آن هر بات فقط با یک بات دیگر در ارتباط است. درنتیجه گراف اتصال حاصل از این شبکه به‌صورت زنجیر است. این پیاده‌سازی گرچه از مدل کامل نظیربه‌نظیر ساده‌تر است ولی تأخیر انتشار در آن زیاد است و تضمینی نیز برای رسیدن پیام به همه بات‌ها وجود ندارد زیرا این روش از این ضعف رنج می‌برد که در صورت قطع‌شدن زنجیر بخشی از بات‌نت از دسترس خارج می‌گردد درنتیجه چنین ساختاری از استحکام مطلوبی برخوردار نیست.



شکل (۴). ساختار فرمان- کنترل غیرمتمرکز

جدول (۱) ساختار فرمان- کنترل متمرکز و غیرمتمرکز را باتوجه به عوامل بیان‌شده، مورد مقایسه قرار داده است.

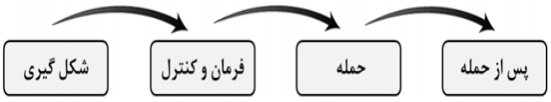
جدول (۱). مقایسه ساختار فرمان- کنترل متمرکز و غیرمتمرکز

ساختار فرمان و کنترل عامل مقایسه	متمرکز	غیرمتمرکز
پیچیدگی طراحی	کم	زیاد
زمان تاخیر پیام	کم	زیاد
قدرت آشکارسازی	زیاد	کم
انعطاف‌پذیری	کم	زیاد

1 -Centralized C&C Structure
2- Intrusion Detection System

۳-چرخه حیات باتنت

به‌منظور این‌که یک میزبان آسیب‌پذیر به یک بات و بخشی از یک باتنت تبدیل شود، باید از یک چرخه از مراحل عبور کند. این مراحل تحت عنوان چرخه حیات باتنت[1] بیان می‌شوند. چرخه حیاط باتنت‌ها در شکل (۵) نشان داده‌شده است:



شکل (۵). چرخه حیات یک باتنت

چرخه حیات باتنت شامل مراحل زیر است [۱۰]:

۱. شکل‌گیری[2]: مرحله شکل‌گیری مرحله‌ای است که مهاجم از آسیب‌پذیری مشخصی در رایانه هدف، سوءاستفاده کرده و از دسترسی به‌دست‌آمده برای نصب برنامه مخرب، استفاده می‌کند. پس از نصب برنامه مخرب، رایانه قربانی برنامه را اجرا کرده و به یک بات تبدیل می‌شود.

۲. فرمان- کنترل[3](C&C) : در مرحله فرمان- کنترل باتها تلاش می‌کنند ارتباطی را با کارگزار فرمان- کنترل خود برقرار سازند و از طریق ارتباط ایجادشده به باتنت بپیوندند.

۳. حمله[4]: پس از این‌که باتها ارتباط خود را با کارگزار فرمان- کنترل برقرار کردند، مرحله حمله آغاز می‌شود. باتها از طریق کانال فرمان- کنترل، دستورات مدیریت را دریافت می‌کنند و فعالیت‌های مخرب را براساس دستورات دریافت شده، انجام می‌دهند.

۴. پس از حمله[5]: در این مرحله ممکن است مدیریت، برنامه بات را برای بهبود قابلیت‌ها یا برخی تغییرات دیگر به‌روزرسانی کند. همچنین ممکن است روش‌های جدیدی را برای جلوگیری از تشخیص باتها و مخفی نگه‌داشتن آن‌ها به‌کار ببرد.

۴- کاربرد داده‌کاوی در شناسایی باتنت‌ها

امروزه شرکت‌ها و سازمان‌ها حجم انبوهی از داده‌ها را در پایگاه‌های داده خود در اختیار دارند. در این راستا نیاز است تا روش‌های خودکاری وجود داشته باشد تا بتواند دانش مفید را از میان انبوه داده‌ها، کشف و استخراج کنند. داده‌کاوی ابزار مؤثری است که برای این منظور مورد استفاده قرار می‌گیرد. در زمینه‌ی شناسایی باتنت‌ها، روش‌های داده‌کاوی می‌توانند با کشف الگوهایی از رفتار مخرب باتنت‌ها، آن‌ها را از ترافیک عادی شبکه متمایز کرده و شناسایی کنند [۱۱]. روشی که در اینجا برای شناسایی باتنت از آن استفاده شده است روش خوشه‌بندی داده‌هاست.

در خوشه‌بندی، داده‌ها براساس قاعده "حداکثر مشابهت درون کلاسی و حداقل مشابهت برون کلاسی" خوشه‌بندی یا گروه‌بندی می‌شوند. درنهایت خوشه‌هایی از داده‌ها شکل می‌گیرند که داده‌های درون یک خوشه بالاترین مشابهت را با یک‌دیگر و کم‌ترین مشابهت را با داده‌های دیگر خوشه‌ها دارند. هر خوشه‌ای که شکل‌گرفته، می‌تواند به‌عنوان یک کلاس از داده‌ها در نظرگرفته‌شود و قواعدی از آن استنتاج شوند [۱۲].

خوشه‌بندی، فرایند گروه‌بندی مجموعه‌ای از داده‌ها به کلاس دارای داده‌های مشابه است. در خوشه‌بندی، هیچ اطلاعی از کلاس‌های موجود درون داده‌ها وجود ندارد و به‌عبارتی خود خوشه‌ها نیز از داده‌ها استخراج می‌شود [۱۳].

۵-دسته‌بندی رویکردهای شناسایی باتنت براساس روش استفاده‌شده برای تشخیص

از این دیدگاه، دو رویکرد کلی می‌توان برای تشخیص باتنت در نظر گرفت:

۱- استفاده از هانی‌پات[6]: هانی‌پات شبکه‌ای از کوزه‌های عسل است که به‌عنوان طعمه در داخل شبکه قرار می‌گیرد و اگر به یک بات آلوده شد، تمام جزئیات تعاملی و رفتاری آن را ثبت می‌نماید و می‌تواند این آلودگی را گزارش کند، که منجر به شناسایی باتنت خواهد شد.

۲- استفاده از شنود غیرفعال: این رویکرد با شنود و تحلیل ترافیک شبکه وجود شبکه‌های بات را تشخیص می‌دهد. این رویکرد نیز به چهار دسته تقسیم می‌شود:

- مبتنی بر امضا[7]: در این روش سعی می‌شود که با تحلیل بسته‌های ورودی/خروجی و یافتن الگوهای[8] خاصی در آن‌ها به‌نام امضا و مقایسه این الگوها با بانک الگوی از قبل آماده‌شده، وجود باتنت بررسی شود. این روش در دستگاه‌های تشخیص نفوذ مانند snort به‌کار برده می‌شود.
- مبتنی بر ناهنجاری[9]: در این روش از ناهنجاری‌هایی که فعالیت باتنت بر روی ترافیک طبیعی شبکه ایجاد می‌کند، سعی در شناسایی آن دارند. مقاله بینکلی[10] و

1- Botnet Life Cycle
2- Formation
3- Command and Control
4- Attack
5- Post-Attack

6- Honeynet
7- Signature
8- Pattern
9- Anomaly
10- Binkley

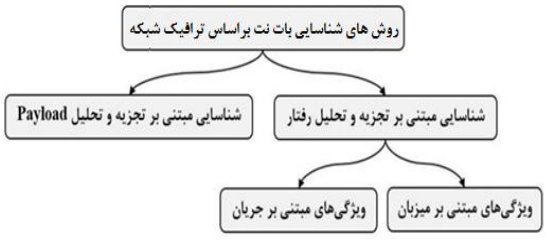
همکاران [۱۴] که توضیح داده خواهد شد نمونه‌ای از این روش‌ها است.

- مبتنی برDNS : این روش در حقیقت یک روش مبتنی بر ناهنجاری است ولی از آنجاکه در بسیاری موارد، پرس‌وجوهایDNS عاملی برای تشخیص بات‌نت است به‌عنوان روشی مستقل نیز از آن یاد می‌شود. بات‌ها برای یافتن کارگزار فرمان- کنترل خود، درخواست‌هایی به DNS می‌دهند. از آنجاکه اغلب کارگزار فرمان- کنترل از فن‌آوری DNS پویا برای مخفی نگه‌داشتن خود استفاده می‌کند، درخواست‌های مکرر بات‌ها یک نوع ناهنجاری در ترافیک DNS ایجاد می‌کند که تشخیص این ناهنجاری به شناسایی بات‌نت منتهی خواهد شد.
- مبتنی بر داده‌کاوی: این روش‌ها گستره‌ای را در برمی‌گیرند و معمولاً بر تحلیل‌های آماری و داده‌کاوی و هوش‌مصنوعی استوار می‌باشند. به‌عنوان نمونه هسنا[1] و همکاران [۱۵] در حملات هرزنامه‌ای ناشی از بات‌نت‌ها ویژگی‌هایی را مدنظر قرار داده‌اند و توانسته‌اند با تحلیل مؤلفه‌های اصلی[2] (PCA)، ویژگی‌های با واریانس بیش‌تر را یافته و حملات را تشخیص دهند. الگوریتم پیشنهادی که در فصل‌های بعدی ارائه خواهد شد، نیز مبتنی بر داده‌کاوی است.

روش‌های نامبرده هرکدام نقاط قوت و ضعف خود را دارند. جدول (۲)، مقایسه‌ای کلی بر روش‌های مطرح‌شده نشان می‌دهد.

۶- دسته‌بندی رویکردهای شناسایی بات‌نت براساس ترافیک شبکه

روش‌های شناسایی بات‌نت‌ها، روش‌های شناسایی مبتنی بر ترافیک شبکه هستند که به نظارت بر ترافیک شبکه تمرکز دارند. این روش‌ها به دودسته تجزیه و تحلیل مبتنی بر محتوا و تجزیه و تحلیل مبتنی بر رفتار تقسیم‌بندی می‌شوند [۱۶] [۱۷] که در ادامه بیش‌تر توضیح داده شده است.



شکل (۶). دسته‌بندی روش‌های شناسایی بات‌نت براساس ترافیک شبکه

۶-۱- شناسایی مبتنی بر تجزیه‌وتحلیل محتوا

در روش‌های شناسایی مبتنی بر تجزیه‌وتحلیل محتوا[3]، محتوای بسته‌های مبادله‌شده در شبکه، تجزیه و تحلیل می‌شوند تا براساس تجزیه و تحلیل انجام‌شده و ویژگی‌های به‌دست‌آمده، ترافیک بات‌نت شناسایی شود. این روش‌ها اگرچه دقت تشخیص بالایی را نشان می‌دهند، اما از وجود برخی محدودیت‌ها رنج می‌برند [۶ و ۱۷]:

- در روش‌های شناسایی مبتنی بر تجزیه و تحلیل محتوا، نیاز است حجم زیادی از داده‌های بسته‌های ترافیک تجزیه‌وتحلیل شوند، بنابراین روش‌هایی زمان‌بر هستند و حجم پردازشی[4] زیادی را دربردارند.
- روش‌های شناسایی مبتنی بر تجزیه و تحلیل محتوا نمی‌توانند بات‌نت‌هایی را که از بسته‌های رمزنگاری‌شده در کانال‌های ارتباطی خود استفاده می‌کنند، شناسایی کنند. زیرا بسته‌های ترافیک رمزنگاری‌شده و اطلاعات محتوای بسته در دسترس نیست.
- در روش‌های شناسایی مبتنی بر تجزیه و تحلیل محتوا، ممکن است به‌دلیل مسائل قانونی و حریم خصوصی، امکان خواندن محتوای بسته‌های ترافیک و دست‌یابی به اطلاعات محتوا وجود نداشته باشد،

۶-۲- شناسایی مبتنی بر تجزیه‌وتحلیل رفتار

میزبان‌های آلوده به بات در یک بات‌نت به‌خاطر از پیش کدشدن، رفتار ترافیکی مشابهی را از خود نشان می‌دهند. در روش‌های شناسایی مبتنی بر تجزیه و تحلیل رفتار ترافیک از این ایده استفاده می‌شود که بات‌ها درون یک بات‌نت معمولاً رفتار ترافیکی یکنواختی دارند و الگوهای ترافیکی خاصی را به‌منظور برقراری ارتباط در شبکه از خود نشان می‌دهند [۱۷].

این روش‌ها با استخراج ویژگی‌های مختلفی که در ترافیک شبکه رخ می‌دهد، مانند اندازه بسته، مدت‌زمان جریان و غیره، رفتارها و الگوهای ترافیکی بات‌نت‌ها را مشخص کرده و ترافیک آن‌ها را از ترافیک عادی شبکه متمایز می‌سازند [۱۱]. مزیت روش‌های شناسایی مبتنی بر تجزیه و تحلیل رفتار ترافیک در این است که به محتوای بسته‌های ترافیک وابسته نیستند و می‌توانند بات‌نت‌های با ترافیک رمزگذاری‌شده را شناسایی کنند. همچنین، اطلاعات مربوط به ترافیک شبکه معمولاً می‌تواند به‌راحتی از دستگاه‌های مختلف شبکه بازیابی شود، بدون این‌که تأثیر قابل‌توجهی بر عملکرد شبکه یا دسترسی خدمات داشته باشد [۱۸] ویژگی‌ها می‌توانند براساس خصوصیات میزبان‌ها و یا

1 - Husna
2- Principal Component Analysis

3- Payload
4- Overhead

جریان‌های ترافیک شبکه باشند و به دو دسته ویژگی‌های مبتنی بر میزبان و ویژگی‌های مبتنی بر جریان گروه‌بندی می‌شوند [۴ و ۱۱]که در ادامه در مورد آن‌ها توضیح داده شده است.

ویژگی‌های مبتنی بر میزبان: ویژگی‌هایی هستند که از الگوهای ارتباطی میزبان‌ها استخراج می‌شوند. این ویژگی‌ها در ارتباطات بین یک میزبان و دیگر میزبان‌های شبکه رخ می‌دهند و برای شناسایی میزبان‌های با الگوهای ارتباطی مشترک، مفید هستند. استخراج ویژگی‌های مبتنی بر میزبان، نیازمند تجزیه و تحلیل هر بسته متعلق به یک میزبان خاص است و زمان زیادی را نسبت به استخراج ویژگی‌های مبتنی بر جریان، صرف می‌کند.

ویژگی‌های مبتنی بر جریان: ویژگی‌هایی هستند که از جریان‌های شبکه استخراج می‌شوند و برای نسبت دادن جریان‌ها به کلاس خاصی از ترافیک شبکه مثل ترافیک بات یا ترافیک غیربات، مورداستفاده قرار می‌گیرند. جریان به مجموعه‌ای از بسته‌ها اطلاق می‌شود که آدرس IP مبدأ، درگاه مبدأ، آدرس IP مقصد، درگاه مقصد و پروتکل یکسان دارند [۱۶]. استخراج ویژگی مبتنی بر جریان، زمان کمتری را نسبت به ویژگی‌های مبتنی بر میزبان صرف می‌کند، زیرا فضای تجزیه و تحلیل، فقط اطلاعات مربوط به جریان است.

جدول (۲). مقایسه روش‌های تشخیص بات‌نت (۱)

روش تشخیص	مزایا	معایب
مبتنی بر امضا	الگویتم ساده و دقیق	عدم شناسایی بات‌های جدید افزایش نرخ منفی کاذب برای بات‌های جدید افزایش نرخ مثبت کاذب هنگام تنوع بات‌ها سنگین بودن الگوریتم به‌دلیل داشتن پایگاه داده بزرگ از امضاها عبور بات‌نت از سیستم، با استفاده از فن‌های مبهم کدگذاری
مبتنی بر ناهنجاری	داشتن عمومیت قادر بودن به شناسایی بات‌های جدید مقاوم در برابر ترافیک رمز شده سبک‌تر نسبت به روش مبتنی بر امضا	سربار محاسباتی بیش‌تر نسبت به الگوریتم‌های مبتنی بر امضا اعلان خطای بالا
مبتنی بر DNS	داشتن تمام مزایای مربوط به الگوریتم‌های مبتنی بر ناهنجاری پیاده یازی راحت‌تر نسبت به الگوریتم‌های مبتنی بر ناهنجاری	داشتن تمام معایب مربوط به الگوریتم‌های مبتنی بر ناهنجاری
مبتنی بر داده‌کاوی	شناسایی بات‌های جدید مقاوم در برابر رمزنگاری	پیچیدگی الگوریتم سرعت عمل کم‌تر

۷- معرفی چند پژوهش برای تشخیص بات‌نت نظیربه نظیر

تارنج[1] و همکاران [۱۹] یک روش شش مرحله‌ای را برای شناسایی جریان‌های ترافیک بات‌نت‌های نظیربه‌نظیر در مرحله فرمان- کنترل از چرخه حیات ارائه داده‌اند که مبتنی بر محتوا است. برای تقسیم‌بندی جریان‌های ترافیک استفاده‌شده است.در ادامه تعدادی از رویکردهای شناسایی بات‌نت‌های نظیربه‌نظیر که مبتنی بر تجزیه و تحلیل رفتار ترافیک هستند، بیان شده است.

چن[2] و همکاران [۲۰] رویکردی را برای شناسایی بات‌های نظیربه‌نظیر مبتنی بر میزبان، ارائه داده‌اند. رویکرد آن‌ها تشخیص رفتارهای مخرب و ارتباطات نظیربه‌نظیر را هدف قرار داده است. آن‌ها برای تشخیص رفتارهای مخرب، فراخوانی توابع API تولیدشده توسط برنامه بات بر روی میزبان را تجزیه و تحلیل کرده‌اند. برای شناسایی ارتباطات نظیربه‌نظیر، از ویژگی‌های رفتاری میزبان مانند آدرس IP، شماره پورت و نوع پروتکل استفاده‌شده است. در رویکرد آن‌ها، ویژگی‌های استفاده‌شده مبتنی بر میزبان بوده و استخراج آن‌ها زمان زیادی را صرف می‌کند. رویکرد آن‌ها برای شناسایی نیاز دارد تا فعالیت مخرب توسط بات انجام شود و برای شناسایی در مرحله فرمان- کنترل مناسب نیست. به‌عبارت دیگر، امکان شناسایی بات در مرحله فرمان- کنترل وجود ندارد. نتایج ارزیابی آن‌ها نشان‌دهنده نرخ تشخیص ۹۵/۷ درصد و نرخ مثبت کاذب[3] ۳/۵ درصد برای شناسایی بات‌های نظیربه‌نظیر است.

استوانویک[4] و همکاران [۲۱] یک سیستم تشخیص مبتنی بر جریان را برای شناسایی ترافیک بات‌نت‌های نظیربه‌نظیر ارائه داده‌اند. سیستم آن‌ها از دو جزء تشکیل شده است: پیش‌پردازش و طبقه‌بند[5]. در مرحله پیش‌پردازش، با تجزیه و تحلیل ترافیک شبکه در سطح جریان، ویژگی‌های استخراج شده‌اند و در مرحله طبقه‌بند، جریان‌های ترافیک با استفاده از الگوریتم‌های یادگیری ماشین[6] نظارتی، به‌صورت مخرب و غیرمخرب طبقه‌بندی می‌شوند. در سامانه‌ی آن‌ها، تعداد ۳۹ ویژگی از جریان‌های ترافیک استخراج‌شده و عملکرد ۸ طبقه‌بند، مورد ارزیابی

1- Tarng
2- Chen

3- False Positive
4 - Stevanovic
5- Classifier
6- Machine Learning

قرارگرفته است. سه طبقه‌بند درخت تصمیم C4.5، Random Forest و Random Tree عملکرد بهتری را نسبت به دیگر طبقه‌بندها نشان داده‌اند. عملکرد طبقه‌بندها، نرخ تشخیص ترافیک را بین ۹۵/۵ تا ۹۶/۵ درصد نشان داده است.

سعد[1] و همکاران [۱۸] رویکردی را برای شناسایی بات‌نت‌های نظیربه‌نظیر در مرحله فرمان- کنترل با استفاده از رفتار ترافیک شبکه ارائه داده‌اند. در رویکرد آن‌ها از ویژگی‌های مبتنی بر میزبان و مبتنی بر جریان برای شناسایی ترافیک بات‌نت‌های نظیربه‌نظیر استفاده شده است. آن‌ها تعداد ۱۷ ویژگی را از جریان‌های شبکه و الگوهای ارتباطی میزبان‌ها استخراج کرده‌اند. استخراج ویژگی‌های مبتنی بر میزبان زمان بیشتری را نسبت به ویژگی‌های مبتنی بر جریان صرف می‌کند. در رویکرد آن‌ها، عملکرد پنج روش یادگیری ماشین برای طبقه‌بندی ترافیک شبکه، بررسی و ارزیابی‌شده است. نتایج رویکرد آن‌ها به‌منظور تشخیص ترافیک فرمان- کنترل بات‌نت‌های نظیربه‌نظیر، نرخ تشخیص بالای ۹۰ درصد و نرخ خطای کم‌تر از ۷ درصد را برای طبقه‌بندهای SVM، ANN و Nearest Neighbors نشان می‌دهد.

پژوهش‌های مختلفی به‌منظور شناسایی بات‌نت‌های نظیربه‌نظیر موردبررسی و ارزیابی قرار گرفتند. نتایج ارزیابی نشان می‌دهد که اگرچه تاکنون رویکردهای مختلفی برای شناسایی بات‌نت‌های نظیربه‌نظیر ارائه شده است، اما این رویکردها با برخی مشکلات و محدودیت‌ها مواجه‌اند که عبارتنداز:

برخی رویکردها (مانند رویکرد چن و همکاران)، بات‌نت‌ها را باتوجه به فعالیت مخرب آن‌ها شناسایی می‌کنند. چنین رویکردهایی فقط قادرند بات‌نت را در مرحله حمله از چرخه حیات شناسایی کنند و امکان شناسایی در مرحله فرمان- کنترل وجود ندارند. رویکردی مؤثر است که بتواند بات‌ها را پیش از این که اقدام به حمله کنند، شناسایی کند.

برخی رویکردها (مانند رویکرد تارنج و همکاران)، شناسایی را براساس محتوای بسته‌های ترافیک انجام می‌دهند. این رویکردها نمی‌توانند بات‌نت‌هایی را که از بسته‌های رمزنگاری‌شده در کانال‌های ارتباطی خود استفاده می‌کنند، شناسایی کنند. برخی رویکردها که شناسایی را براساس ویژگی‌های مبتنی بر جریان انجام می‌دهند، تعداد زیادی ویژگی را از جریان‌های شبکه استخراج کرده‌اند (مانند رویکرد استوانویک و همکاران که تعداد ۳۹ ویژگی را از جریان‌های ترافیک استخراج کرده‌اند). نکته مهم در شناسایی بات‌نت‌ها، انتخاب بهترین و کم‌ترین تعداد ویژگی است تا با کم‌ترین حجم محاسباتی، امکان شناسایی بات‌نت‌ها به‌صورت سریع و بلادرنگ وجود داشته باشد. مسئله این پژوهش شناسایی بات‌نت‌های نظیربه‌نظیر است. باتوجه به این‌که شناسایی براساس تجزیه‌وتحلیل رفتار ترافیک صورت می‌گیرد، شناسایی مستقل از محتوای بسته‌های ترافیک بوده و امکان تشخیص بات‌نت‌های نظیربه‌نظیر با ترافیک رمزنگاری‌شده وجود دارد. در این پژوهش از ویژگی‌های مبتنی بر جریان برای شناسایی بات‌نت استفاده‌شده است. و بر اساس میزان شباهت[2] بین جریان‌ها، طبق قواعد خاصی کار شناسایی انجام می‌شود. نکته‌ای که باعث شده این پژوهش نسبت به پژوهش‌های مشابه انجام‌شده، برتری داشته باشد این است که با استفاده از این الگوریتم می‌توان ترافیک بات‌نت را از ترافیک ازدحام ناگهانی تشخیص داد. ترافیک ازدحام ناگهانی ترافیک مشروع اما ناخواسته‌ای است که به‌دلیل سیل عظیم همزمان کاربران به یک سایت به‌وجود می‌آید. این مسئله در بخش‌های بعدی به‌طور مفصل شرح داده خواهد شد. در ادامه چند پژوهش دیگر در زمینه شناسایی بات‌نت معرفی می‌شود.

جدول (۳) خلاصه رویکردهای بیان‌شده برای شناسایی بات‌نت‌ها را نشان می‌دهد.

جدول (۳). خلاصه رویکردهای شناسایی بات‌نت‌ها

رویکرد	روش شناسایی از دیدگاه اول	روش شناسایی از دیدگاه دوم	مرحله شناسایی	نوع معماری شبکه بات‌نت	نرخ تشخیص
تارنج و همکاران [۱۹]	مبتنی بر داده‌کاوی	مبتنی بر محتوا	فرمان- کنترل	نظیربه نظیر	-
چن و همکاران [۲۰]	مبتنی بر ناهنجاری	مبتنی بر میزبان	حمله	نظیربه نظیر	نرخ تشخیص ۹۵/۷ درصد و نرخ مثبت کاذب[3] ۳/۵ درصد
استوانویک و همکاران [۲۱]	مبتنی بر داده‌کاوی	مبتنی بر جریان	فرمان- کنترل	نظیربه نظیر	نرخ تشخیص ترافیک رابین ۹۵/۵ تا ۹۶/۵ درصد
سعد و همکاران [۱۸]	مبتنی بر داده‌کاوی	مبتنی بر میزبان و جریان	فرمان- کنترل	نظیربه نظیر	نرخ تشخیص بالای ۹۰درصد و نرخ خطای کم‌تر از ۷درصد
روش پیشنهادی	مبتنی بر داده‌کاوی	مبتنی بر جریان	فرمان- کنترل	نظیربه‌نظیر	نرخ تشخیص ۸۵ درصد

1 - Saad

2- Similarity
3- False Positive

۸- طرح پیشنهادی (روش تشخیص مبتنی بر شباهت)

ایده اولیه رویکردی که در این پژوهش ارائه‌شده، برگرفته از کاری است که شوئی یو[1] و همکاران [۲۲] انجام داده‌اند. ابتدا کار شوئی یو و همکاران مطرح می‌شود سپس الگوریتم پیشنهادی که توسعه‌شده الگوریتم شوئی یو وهمکاران است مطرح خواهد شد.

حقایقی در زمینه باتنت وجود دارد که در طول این پژوهش مدنظر قرارگرفته شده‌اند:

- ابزارهای حمله معمولاً در یک باتنت مشابه است. مدیر بات یک فرمان را برای تمام بات‌های باتنتش برای شروع یک حمله درنظر می‌گیرد.
- جریان‌های حمله‌ای که در سمت قربانی مشاهده می‌کنیم، تراکم تعداد زیادی از جریان‌های حمله اولیه است و جریان حمله متراکم شده[2] یک انحراف استاندارد[3] مشابه را به اشتراک می‌گذارد. انحراف استاندارد جریان حمله اغلب کوچک‌تر از جریان ازدحام ناگهانی اصلی است. دلیل این مسئله این است که تعداد بات‌های زنده باتنت به‌مراتب کم‌تر از تعداد کاربران همزمان قانونی ازدحام ناگهانی است (باتنت حدود چندصد یا سطح‌های پایینی از چندهزار است و کاربران همزمان ازدحام ناگهانی در سطح صدهاهزار نفر هستند).

پس نتیجه این‌که، انحراف استاندارد در جریان حمله کم‌تر از جریان ازدحام ناگهانی است پس شباهت بین جریان‌ها در حمله DDOS بیش‌تر از ازدحام ناگهانی است. بنابراین روش تشخیصی ارائه می‌شود که در آن از ضریب هم‌بستگی جریان برای تشخیص استفاده‌شده است. در این پژوهش، شناسایی باتنت نظیربه‌نظیر در مرحله فرمان- کنترل از چرخه حیات، براساس تجزیه و تحلیل ترافیک انجام می‌شود که در ادامه با جزئیات بیش‌تر شرح داده می‌شود.

۸-۱- تعاریف

جریان شبکه: برای یک مسیریاب[4] داده شده در یک شبکه محلی، بسته‌های[5] شبکه را کلاستربندی می‌کنیم، به‌گونه‌ای که آن‌هایی آدرس‌های مقصد و مبدا یکسانی دارند به‌عنوان یک جریان شبکه تعریف می‌شوند.

جریان شبکه را با $X_i[n]$ نشان می‌دهیم که در آن i $(i \geq 1)$ ایندکس جریان‌های شبکه است و n نشان‌دهنده n امین المان درداده‌های متوالی است. به‌عنوان مثال اگر اندازه جریان شبکه داده‌شده X_i ، N باشد، جریان شبکه به‌صورت زیر نمایش داده می‌شود:

$$X_i = \{x_i[1], x_i[2], \ldots\ldots, x_i[N]\} \quad (۱-۸)$$

که در آن، $(1 \leq k \leq n), x_i[k]$ تعداد بسته‌هایی است که در k امین بازه زمانی برای جریان شبکه، شمارش شده است.

شدت جریان[6]: برای یک جریان شبکه X_i که اندازه جریان شبکه، N باشد (N≥1)، امیدریاضی جریان را به‌عنوان شدت جریان تعریف می‌کنیم:

$$E[X_i]= \frac{1}{N}\sum_{n=1}^{N} x_i[n] \quad (۲-۸)$$

شدت جریان، میانگین نرخ بسته‌های یک جریان شبکه را نشان می‌دهد. اگر X_i یک جریان حمله DDOS باشد، به $E[X_i]$، شدت حمله می‌گوییم.

اثر انگشت جریان[7]: برای جریان شبکه داده‌شده X_i با اندازه N، اثر انگشت جریان X_i' نمایش یک‌دستی از X_i است:

$$X_i' = \{x_i'[1], x_i'[2], \ldots\ldots, x_i'[N],\} = \left\{\frac{x_i[1]}{N.E[X_i]}, \frac{x_i[2]}{N.E[X_i]}, \ldots\ldots, \frac{x_i[N]}{N.E[X_i]}\right\} \quad (۳-۸)$$

طبق این تعریف می‌دانیم که، $\sum_{k=1}^{N} x_i'[k] = 1$

اگر Xi و Xj $(i \neq j)$ دو جریان شبکه با اندازه یکسان N باشند همبستگی بین دوجریان[8] به‌صورت زیر تعریف می‌شود:

$$r_{x_i x_j} = \frac{1}{N}\sum_{n=1}^{N} x_i[n]x_j[n] \quad (۴-۸)$$

همبستگی جریان برای نشان دادن میزان شباهت بین جریان‌های مختلف استفاده می‌شود. در بعضی موارد ممکن است همبستگی، مقدار صفر را نشان دهد درحالی‌که دو جریان کاملاً هم بسته باشند، دلیل این امر تفاوت فاز بین دوجریان است، بنابراین این تعریف به‌صورت زیر توسعه داده شد:

$$r_{x_i x_j}[k] = \frac{1}{N}\sum_{n=1}^{N} x_i[n]x_j[n+k] \quad (۵-۸)$$

که در آن، k (k=0,1,2,3,…,N-1) مکان جریان X_j بعد از اعمال تغییر فاز است.

ضریب هم‌بستگی جریان[9]: اگر X_i و X_j $(i \neq j)$ دو جریان شبکه با اندازه یکسان N باشند، ضریب هم‌بستگی جریان را به صورت زیر تعریف می‌کنیم:

$$\rho_{x_i x_j}[k] = \frac{r_{x_i x_j}[k]}{\frac{1}{N}\left[\sum_{n=1}^{N} x_i^2[n]\sum_{n=1}^{N} x_j^2[n]\right]^{1/2}} \quad (۶-۸)$$

1- Shui Yu
2- Aggregated Attack Flows
3- Standard Deviation
4- Router
5- Packet

6- Flow Strength
7- Flow Fingerprint
8- Flow Correlation
9- Flow Correlation Coefficient

ضریب هم‌بستگی جریان، برای مشخص‌کردن شباهت بین دو جریان استفاده می‌شود. زمانی که دو جریان مشابه داشته باشیم که تفاوت فازی داشته باشند، مقدار این ضریب کاهش پیدا می‌کند. در اینجا طبق رابطه (۸-۵) این مشکل با تغییر فاز حل می‌شود.

۸-۲- شناسایی مبتنی بر شباهت

در اینجا روشی برای تشخیص بات‌نت ارائه می‌شود که اساس آن مبتنی برشباهت جریان‌هاست، و از آن برای مقابله با حمله ازدحام ناگهانی استفاده می‌شود.

هنگامی‌که یک هشدار حمله DDOS در شبکه اعلام می‌شود، رویترها با استفاده از شمارش تعداد بسته‌ها در یک بازه زمانی داده شده (مثلاً ۱۰۰ میلی‌ثانیه)، شروع به نمونه‌گیری از جریان‌های مشکوک می‌کنند. هنگامی‌که اندازه جریان موردنظر N به‌دست آمد، ضریب همبستگی جریان بین جریان‌های مشکوک محاسبه می‌شود.

فرض کنیم که M جریان شبکه وجود دارد، X_1 و X_2 ، X_3، X_4 ..... و X_M ، بنابراین می‌توان ضریب هم‌بستگی را بین هر دوجریان ، $(1 \le i \le M)X_i$ و $(1 \le j \le M)X_j$ به‌دست آورد.

$I_{X_iX_j}$ به‌عنوان شاخص شباهت بین دو جریان X_i و X_j در نظر گرفته می‌شود، این شاخص می‌تواند دو مقدار داشته باشد: ۱- برای حمله DDOS و صفر برای حالات دیگر. ∂ نیز به‌عنوان حد آستانه برای تصمیم‌گیری درنظر گرفته می‌شود به‌طوری‌که :

$$I_{X_iX_j}=\begin{cases}1, & \rho_{x_ix_j}[k] \ge \partial \\ 0, & otherwise\end{cases} \qquad (۸-۷)$$

که درآن، $1 \le j,i \le M$ و $i \ne j$. ضریب هم‌بستگی بین هر دو جریان متفاوت طبق رابطه (۸-۷) محاسبه می‌شود و مجموع آن‌ها طبق رابطه (۸-۸) به‌عنوان احتمال وجود حمله درنظر گرفته می‌شود.

$$\Pr(I_A=1)=\frac{\sum_{1\le i,j\le M, i\ne j} I_{X_iX_j}}{\binom{M}{2}} \qquad (۸-۸)$$

که در آن، I_A شاخص حمله DDOS است. در اینجا هم یک حد آستانه درنظر گرفته می‌شود که اگر مقدار احتمال از این حد آستانه بیش‌تر شود نشان‌دهنده وجود حمله در شبکه است.

$$I_A=\begin{cases}1, & \Pr(I_A=1) \ge \partial' \\ 0 & \Pr(I_A=1) < \partial'\end{cases} \qquad (۸-۹)$$

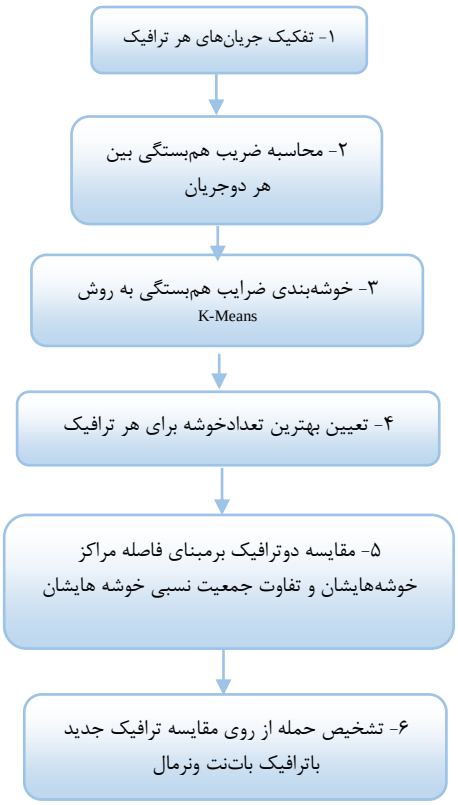
به‌عنوان مثال، اگر حد آستانه ۰/۶ باشد، درصورتی حمله DDOS داریم که حداقل ۶۰ درصد مقایسات مثبت باشد. حمله DDOS و ازدحام ناگهانی ازضریب هم‌بستگی جریان متفاوتی استفاده می‌کنند. لازم است که یک حد بالا برای ضریب هم‌بستگی جریان به‌دست بیاوریم که اگر ضریب همبستگی از این مقدار بیش‌تر باشد این‌طور درنظر می‌گیریم که حمله رخ داده است.

در رابطه‌های (۸-۷) و (۸-۹) ، زمانی‌که از حد آستانه استفاده می‌کنیم، یک تعداد از جریان‌ها را به‌طور کامل کنار می‌گذاریم. در واقع این اطلاعات کلاً دور ریخته می‌شود که این باعث می‌شود دقت الگوریتم کم‌تر شود. برای مقابله با این مشکل، الگوریتم فوق را کمی بهبود داده‌ایم که در ادامه آن را توضیح می‌دهیم.

بعد از این که همبستگی بین هر دوجفت جریان محاسبه شد، حجم زیادی از ضرایب همبستگی بین جفت جریان‌ها را داریم که با استفاده از الگوریتم K-Means خوشه‌بندی می‌شوند. و بهترین خوشه‌بندی با تعداد مراکز مناسب انتخاب می‌شود. روش کار الگوریتم K-Means و چگونگی انتخاب بهترین خوشه‌بندی در فصل بعد بیش‌تر توضیح داده خواهدشد. این کار برای دو ترافیک بات‌نت و معمولی به‌صورت جدا انجام می‌شود. ترافیک معمولی می‌تواند ترافیک ازدحام ناگهانی و یا یک ترافیک معمولی از شبکه باشد. چراکه الگوریتم ارائه‌شده برای هر دوحالت می‌تواند جواب‌گو باشد. مراحل کار به این‌صورت است که ابتدا جریان‌ها جدا می‌شوند و هر جریان به‌شکل رابطه (۸-۱) نمایش داده می‌شود، یعنی تعداد بسته‌های جریان برای بازه‌های زمانی مختلف شمارش می‌شود. سپس همبستگی بین هر دو جریان از جریان‌های ترافیک داده‌شده، طبق رابطه (۸-۵) محاسبه می‌شود و با داشتن مقدار همبسنگی، ضریب همبستگی بین دو جریان طبق رابطه (۸-۶) محاسبه می‌شود. و درنهایت بعد از اعمال الگوریتم K-Means روی این ضریب‌ها، بهترین خوشه‌بندی انتخاب می‌شود. تمام مراحل فوق برای هر دو ترافیک بات‌نت و ترافیک معمولی انجام می‌شود. مراحل اجرای الگوریتم در شکل (۷) نشان داده شده است:

تا اینجا بهترین خوشه‌بندی برای ترافیک‌ها مشخص شد. حال برای این که بتوان برای یک ترافیک جدید، حمله‌بودن یا نبودن آن‌را تشخیص داد، به دو طریق می‌توان عمل کرد: اول این‌که برای ترافیک جدید به‌ازای هر دو جریان موجود در آن، ضریب همبستگی دو جریان محاسبه شود و به خوشه‌بندی‌های ترافیک بات‌نت موجود اضافه شود. چنان‌چه باعث افزایش پراکندگی خوشه‌های موجود شد، بدین معنی است که ترافیک جدید از جنس ترافیک بات‌نت نیست. راه حل دوم که نسبت به راه‌حل قبلی عملکرد بهتری دارد، این است که ترافیک جدید

همانند ترافیک باتنت و معمولی، خوشه‌بندی شود و خوشه‌های جدید با خوشه‌های ترافیک معمولی و باتنت مقایسه شود. و براساس شباهتی که با هردوی این ترافیک‌ها دارد می‌توان تشخیص داد که از کدام نوع است. برای تعیین شباهت بین دوترافیک می‌توان چند معیار را بررسی کرد. ممکن است این سؤال پیش آید که برای مقایسه، تعداد خوشه‌ها را باید ثابت درنظر گرفت؟ جواب خیر است، چراکه بهترین تعداد خوشه‌ها طبق آزمایش، به‌دست‌آمده و ممکن است برای هر ترافیک تعداد خوشه متفاوتی داشته باشیم.



شکل (۷). مراحل اجرای الگوریتم تشخیص باتنت

برای مقایسه ترافیک ناشناخته با ترافیک باتنت فاصله هریک از مراکز خوشه‌های ترافیک جدید با تمام مراکز خوشه‌های ترافیک باتنت محاسبه می‌شود. مرکزی که کم‌ترین مقدار را دارد، شبیه‌ترین خوشه به این مرکز است. درنهایت مجموع فاصله مراکز ترافیک جدید به شبیه‌ترین خوشه‌هایشان، محاسبه می‌شود:

$$\forall 1 \le i \le m, 1 \le j \le n \qquad (۸-۱۰)$$
$$D_i = \min(d_{ij})$$

$$d_{ij} = d(v_i, v_j) = \left| V_i - V_j \right| \qquad (۸-۱۱)$$
$$1 \le i \le m \text{ ،} 1 \le j \le n$$

$$T = \sum_{i=1}^{m} D_i \qquad (۸-۱۲)$$

که در آن، V_i مرکز خوشه iام از ترافیک جدید و V_j مرکز خوشه jام از ترافیک باتنت، m تعداد خوشه‌های ترافیک جدید و n تعداد خوشه‌های ترافیک باتنت می‌باشد.

مقدار T را نسبت به ترافیک جدید و ترافیک معمولی نیز محاسبه می‌کنیم. بدیهی است هرچه T کمتر باشد ترافیک‌ها به هم شبیه‌ترند. حال با مقایسه این مقدار برای دو ترافیک معمولی و باتنت می‌توان نتیجه گرفت که ترافیک جدید از چه نوعی است. اگر مقدار این شاخص برای ترافیک جدید نسبت به ترافیک باتنت کمتر از ترافیک معمولی باشد نتیجه می‌گیریم که ترافیک جدید از نوع باتنت است. برای روشن‌تر شدن موضوع، در فصل بعد به‌صورت عملی این را نشان خواهیم داد.

اما مساله‌ای هست که تا اینجا درنظرگرفته نشده‌است و آن، مساله‌ی جمعیت هر خوشه است. هرچه جمعیت دو خوشه از دو ترافیک متفاوت به هم نزدیک‌تر باشد دلیلی بر شباهت بیشتر این دو خوشه است. از آنجایی که جمعیت کلی، بین دو ترافیک متفاوت است، برای مقایسه، جمعیت هر خوشه تقسیم بر کل جمعیت را به‌عنوان جمعیت نسبی در نظر می‌گیریم و جمعیت نسبی را به‌عنوان عامل مقایسه در نظر می‌گیریم:

$$P_i = \frac{\|c_i\|}{\binom{M}{2}} \qquad (۸-۱۳)$$

که در آن، M تعداد جریان‌های شبکه است و $\|c_i\|$ تعداد عناصر خوشه iام.

پس اگر بخواهیم دو خوشه را شبیه به‌هم درنظر بگیریم اولاً مراکز دوخوشه باید به‌هم شبیه باشند و ثانیاً، جمعیت نسبی دو خوشه باید به هم نزدیک باشد. بنابراین بعد از این‌که طبق رابطه (۸-۱۰) خوشه‌هایی که مراکز آن‌ها به‌هم تزدیک است، مشخص شد، جمعیت این دو خوشه با هم مقایسه می‌شود و با اضافه کردن این مقدار به‌عنوان یک ضریب در رابطه (۸-۱۲) تأثیر جمعیت درنظر گرفته می‌شود. یعنی احتمال شبیه‌بودن دو خوشه، به‌وسیله تفاوت جمعیت نسبی‌شان مشخص می‌شود. پس رابطه (۸-۱۲) به‌صورت زیر تعمیم داده می‌شود:

$$T_A = \sum_{i=1}^{m} A_i D_i \qquad (۸-۱۴)$$

$$A_i = \left| P_i - P_j \right| \qquad (۸-۱۵)$$

که در آن، $1 \le i \le m$ و j اندیس خوشه‌ای است که به‌ازای آن d_{ij} کم‌ترین مقدار را دارد. بدیهی است که هرچه مقدار فوق کم‌تر باشد، دو ترافیک به‌هم شبیه‌ترند. و برای تعیین این‌که ترافیک جدید از چه نوعی است باید مشخص کنیم که به کدام

ترافیک شبیه‌تر است

در ادامه به بیان چگونگی پیاده‌سازی الگوریتم، جمع‌آوری بسته‌های ترافیک، استخراج جریان‌ها و محاسبه همبستگی می‌پردازیم. سپس با استفاده از الگوریتم K-Means داده‌ها را خوشه‌بندی خواهیم کرد و براساس پارامترهای مشخص شده، عملکرد الگوریتم را با یک نمونه ترافیک نشان می‌دهیم.

۸-۳- استخراج جریان

برای اجرای الگوریتم نیاز به ترافیک بات‌نت و ترافیک معمولی داریم. برای ترافیک بات‌نت، از مجموعه‌داده‌ای[1] که ISCX[2]، در اختیار عموم قرار داده[3]، استفاده شده‌است. قسمتی از این مجموعه‌داده که حاوی ۲۰۰,۰۰۰ بسته می‌باشد و در بازه زمانی ۸۰۰ ثانیه‌ای تولید شده، برای این کار انتخاب شده است. طبق آن‌چه که برای تعریف جریان قبلاً گفتیم، بسته‌هایی که مبدأ و مقصد یکسان دارند، جدا شده و تحت عنوان یک جریان شناخته می‌شوند. بدین‌ترتیب ۶۶۰ جریان در این بازه زمانی از مجموعه داده مذکور، استخراج گردید.

ترافیک معمولی شبکه هم ترکیبی از ترافیک برنامه‌های مختلف را شامل می‌شود که عبارتند از: برنامه‌های کاربردی (BitTorrent)، برنامه‌های چت (Skype) و ترافیک وب (Web Browsing). برای این کار، از ترافیک دانشگاه بزرگمهر قاین که حدود ۱۲۰۰۰ دانشجو دارد، در اولین ساعات انتخاب واحد، استفاده شده‌است. طبق لاگی که در اختیار داریم، در یک بازه زمانی یک ساعته یعنی ۳۶۰۰ ثانیه‌ای، حدود ۲۰۰,۰۰۰ بسته در آن ردوبدل شده‌است که از این تعداد بسته، ۳۸۰ جریان استخراج شده‌است. استخراج جریان و اجرای الگوریتم ارائه‌شده در محیط پایتون انجام شده است و برای ذخیره اطلاعات از پایگاه داده SQL Server استفاده‌شده است.

برای هر جریان، بازه‌های زمانی که بسته‌ها درآن بازه شمارش می‌شوند، ۵ ثانیه‌ای درنظر گرفته شده، بنابراین مقدار N، برای ترافیک بات‌نت۱۶۰ و برای ترافیک معمولی ۷۲۰ خواهد بود. (طبق رابطه (۸-۱)). پس از این‌که جریان‌ها مشخص شد، وضعیت هر بسته، در مورد اینکه بسته مربوط به کدام جریان است، مشخص می‌شود. سپس برای هر دو جریان از ترافیک موجود، مقدار هم‌بستگی بین آن‌ها طبق رابطه (۸-۵) محاسبه می‌شود. مقدار k در این محاسبات صفر درنظر گرفته شده است، چرا که جریان‌های موجود در این مجموعه داده هم‌فاز هستند و چنان‌چه روی آن تغییر فاز اعمال شود، مقادیر نادرستی را برای همبستگی نتیجه می‌دهند. مقدار ضریب همبستگی بین دو جریان براساس مقدار همبستگی، طبق رابطه (۸-۶) محاسبه می‌شود. حال برای هر دو جریان داده شده، مقدار ضریب همبستگی جریان را داریم که عددی بین ۰ تا ۱ است. الگوریتم K-Means را برای کلاستربندی جریان‌ها، براساس ضریب همبستگی‌شان استفاده می‌کنیم.

روش K-Means یکی از روش‌های خوشه‌بندی داده‌ها در داده‌کاوی است. این روش علی‌رغم سادگی آن، یک روش پایه برای بسیاری از روش‌های خوشه‌بندی دیگر (مانند خوشه‌بندی فازی) محسوب می‌شود. برای این الگوریتم شکل‌های مختلفی بیان شده است. ولی همه آن‌ها دارای روالی تکراری هستند که برای تعدادی ثابت از خوشه‌ها سعی در تخمین به‌دست آوردن نقاطی به‌عنوان مراکز خوشه‌ها دارند. این نقاط در واقع همان میانگین نقاط متعلق به هر خوشه هستند. هر نمونه داده، به یک خوشه نسبت داده می‌شود، به‌گونه‌ای که آن داده، کم‌ترین فاصله تا مرکز آن خوشه را دارا باشد.

۸-۴- تعیین بهترین تعداد خوشه‌ها

هدف از اعتبارسنجی خوشه‌ها یافتن خوشه‌هایی است که بهترین تناسب را با داده‌های موردنظر داشته باشند. دو معیارِ پایهٔ اندازه‌گیری پیشنهاد شده برای ارزیابی و انتخاب خوشه‌های بهینه عبارتند از: [۲۳]

- تراکم[4]: داده‌های متعلق به یک خوشه بایستی تا حد ممکن به یک‌دیگر نزدیک باشند. معیار رایج برای تعیین میزان تراکم داده‌ها، واریانس داده‌ها است.
- جدایی[5]: خوشه‌ها خود بایستی به اندازه کافی از یک‌دیگر جدا باشند. سه راه برای سنجش میزان جدایی خوشه‌ها مورد استفاده قرار می‌گیرد که عبارتند از:
  - فاصلهٔ بین نزدیک‌ترین داده‌ها از دو خوشه.
  - فاصلهٔ بین دورترین داده‌ها از دو خوشه.
  - فاصلهٔ بین مراکز خوشه‌ها.

جدول (۴). مجموعه علائم به‌کاررفته در این بخش

n_c	تعداد خوشه‌ها
$d(x,y)$	فاصله بین دو داده
v_i	نقطه مرکز خوشه i ام
c_i	I امین خوشه
$\|c_i\|$	تعداد داده‌های درون i امین خوشه

1- Dataset
2- Information Security Center of Excellence
3- http://www.iscx.ca/datasets

4- Compactness
5- Separation

۸-۵- شاخص دیویس بولدین[1]

این معیار از شباهت بین دو خوشه (R_{ij}) استفاده می‌کند که براساس پراکندگی یک خوشه (S_i^*) و عدم شباهت بین دو خوشه (d_{ij}) تعریف می‌شود. شباهت بین دو خوشه را می‌توان به‌صورت‌های مختلفی تعریف کرد ولی بایستی شرایط زیر را دارا باشد.

- $R_{ij} \geq 0$.
- $R_{ij} = R_{ji}$.
- اگر S_j^* و S_i^* هر دو برابر صفر باشند آن‌گاه R_{ij} نیز برابر صفر باشد.
- اگر $S_j^* > S_k^*$ و $d_{ij} = d_{ik}$ آن‌گاه $R_{ij} > R_{ik}$.
- اگر $S_j^* = S_k^*$ و $d_{ij} < d_{ik}$ آن‌گاه $R_{ij} > R_{ik}$.

معمولاً شباهت بین دو خوشه به‌صورت زیر تعریف می‌شود:

$$R_{ij} = \frac{S_i^* + S_j^*}{d_{ij}} \qquad (۱۶-۸)$$

که در آن، d_{ij} طبق رابطه (۸-۱۱) و S_i^* (فاصله میانگین) با روابط زیر محاسبه می‌شوند.

$$S_i^* = \frac{1}{\|c_i\|} \sum_{x \in c_i} d(x, v_i) \qquad (۱۷-۸)$$

که در آن، v_i نقطه مرکز خوشه i ام، c_i، خوشه i ام و $\|c_i\|$ تعداد داده‌های خوشه iام است.

باتوجه به مطالب بیان‌شده و تعریف شباهت بین دو خوشه، شاخص دیویس بولدین به‌صورت زیر تعریف می‌شود.

$$DB = \frac{1}{n_c} \sum_{i=1}^{n_c} R_i \qquad (۱۸-۸)$$

که در آن، n_c تعداد خوشه‌هاست و R_i هم به‌صورت زیر محاسبه می‌شود:

$$R_i = \max_{j \to 1..n_c, i \neq j} (R_{ij}), i = 1...n_c \qquad (۱۹-۸)$$

این شاخص در واقع میانگین شباهت بین هر خوشه با شبیه‌ترین خوشهٔ به آن را محاسبه می‌کند. می‌توان دریافت که هرچه مقدار این شاخص کم‌تر باشد، خوشه‌های بهتری تولید شده است.

۹- پیاده‌سازی

الگوریتم فوق را با تعداد مراکز مختلف برای هر دو ترافیک معمولی و بات‌نت اجرا کردیم. اجرای الگوریتم بر روی ترافیک بات‌نت، همراه با جزئیات در ادامه شرح داده شده‌است. تمام مراحلی که روی این ترافیک انجام شده، بر روی ترافیک معمولی هم اعمال شده و نتایج نهایی برای ترافیک معمولی هم در انتها نشان داده شده است.

۹-۱- اجرای الگوریتم K-Means

برای اجراهای مختلف الگوریتم با تعداد مراکز متفاوت، نتایج نهایی را در جداول زیر نشان می‌دهیم. در هریک از موارد زیر، برای تست پایداربودن مراکز، اجرای الگوریتم را تا ده مرتبه بعد از پایداری، ادامه داده‌ایم. S_i^* طبق رابطه (۸-۱۷) محاسبه شده است.

جدول (۵). اجرای الگوریتم K-Means با دو مرکز (تعداد تکرار الگوریتم برای رسیدن به پایداری: ۵ بار)

مراکز اولیه	مراکز نهایی	جمعیت $\|c_i\|$	فاصله میانگین(S_i^*)
۰/۳۳	۰/۲۰	۱۲۹۱۷۲	۰/۰۸۳
۰/۴۶	۰/۷۴	۸۸۲۹۸	۰/۱۷۹

جدول (۶). اجرای الگوریتم با سه مرکز (تعداد تکرار الگوریتم برای رسیدن به پایداری: ۹ بار)

مراکز اولیه	مراکز نهایی	جمعیت $\|c_i\|$	فاصله میانگین(S_i^*)
۰/۳۳	۰/۱۴	۱۰۰۳۸۷	۰/۰۶۱
۰/۷۸	۰/۵۲	۸۰۴۱۶	۰/۰۹۵
۰/۹۲	۰/۹۶	۳۶۶۶۷	۰/۰۵۸

جدول (۷). اجرای الگوریتم با چهار مرکز (تعداد تکرار الگوریتم برای رسیدن به پایداری: ۱۳ بار)

مراکز اولیه	مراکز نهایی	جمعیت $\|c_i\|$	فاصله میانگین(S_i^*)
۰/۱۱	۰/۰۶	۵۹۹۸۳	۰/۰۳۲
۰/۲۳	۰/۳۲	۶۶۲۵۱	۰/۰۵۹
۰/۵۴	۰/۵۹	۵۶۳۹۵	۰/۰۷۲
۰/۸۷	۰/۹۷	۳۴۸۴۱	۰/۰۴۴

الگوریتم را تا ۸ مرکز اجرا کردیم که از بیان جزئیات آن صرفنظر می‌شود و تنها نتایج نهایی محاسبات را برای این اجراها در زیر نشان می‌دهیم.

۹-۲- تعیین بهترین خوشه‌بندی

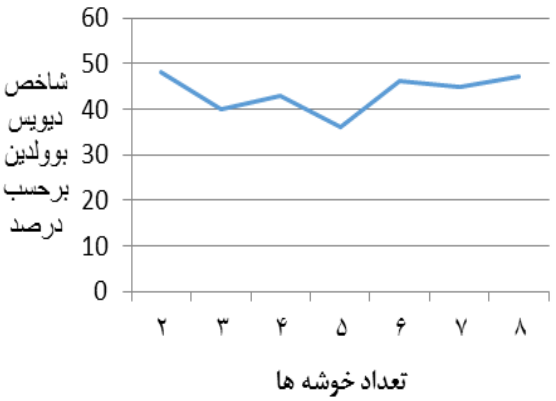
تا اینجا برای تعداد خوشه‌های مختلف الگوریتم اجرا شد. حال باید بهترین تعداد خوشه‌ها انتخاب شود. برای تعیین بهترین تعداد خوشه، مقدار DB طبق رابطه (۸-۱۹) برای خوشه‌بندی‌های فوق محاسبه گردیده و نتایج در جدول (۸)

1- Davies Bouldin Index

نشان داده شده است: همان‌طورکه گفته شد هرچه این شاخص کم‌تر باشد، خوشه‌بندی‌های بهتری تولید شده است. طبق نمودار شکل (۸) کم‌ترین مقدار شاخص در خوشه‌بندی با ۵ مرکز به‌دست آمده است، یعنی بهترین تعداد خوشه برای ترافیک بات‌نت ۵ خوشه می‌باشد. P_i طبق رابطه (۸-۱۳) و S_i^* طبق رابطه (۸-۱۷) محاسبه شده‌است. نتایج در جدول (۹) نشان داده شده است.

جدول (۸). مقدار شاخص دیویس بولدین برای تعداد مراکز مختلف ترافیک بات‌نت

تعداد مراکز	شاخص دیویس بولدین (DB)
۲	۰/۴۸
۳	۰/۴۰
۴	۰/۴۳
۵	۰/۳۶
۶	۰/۴۶
۷	۰/۴۵
۸	۰/۴۷



شکل (۸). نمودار مقدار شاخص دیویس بولدین بر حسب تعدادمراکز

جدول (۹). بهترین خوشه‌بندی برای ترافیک بات‌نت

مرکز خوشه (V_i)	جمعیت خوشه ($\|\|c_i\|\|$)	فاصله میانگین (S_i^*)	جمعیت نسبی (P_i)
۰/۰۵	۵۴۶۶۱	۰/۰۲۸	۰/۲۵
۰/۲۸	۵۷۹۵۱	۰/۰۵	۰/۲۶
۰/۹۴	۴۷۴۲۳	۰/۰۸۴	۰/۲۲
۰/۶۹	۲۵۶۰۶	۰/۰۶۴	۰/۱۲
۰/۹۸	۳۱۸۲۹	۰/۰۲۰	۰/۱۵

مراحل فوق را بر روی ترافیک معمولی در دسترس، نیز اجرا کردیم و پس از گذشت تمام مراحل، نتیجه آزمایشات این بود که بهترین خوشه‌بندی برای ترافیک معمولی، خوشه‌بندی با ۴ مرکز می‌باشد. جدول (۱۰) خوشه‌بندی ترافیک معمولی با چهار خوشه را نشان می‌دهد.

جدول (۱۰). خوشه‌بندی ترافیک معمولی

مرکز خوشه (V_i)	جمعیت خوشه ($\|\|c_i\|\|$)	فاصله میانگین (S_i^*)	جمعیت نسبی (P_i)
۰/۰۵	۲۶۳۱۰	۰/۰۷۴	۰/۳۶
۰/۱۳	۲۱۴۵۲	۰/۰۸۳	۰/۲۹
۰/۳۶	۱۸۷۴۷	۰/۰۷۵	۰/۲۹
۰/۵۷	۵۵۰۱	۰/۱	۰/۰۷

۹-۳- تشخیص حمله

تا اینجا بهترین خوشه‌بندی را برای یک ترافیک بات‌نت و یک ترافیک معمولی پیداکردیم. حال می‌خواهیم با استفاده از داده‌هایی که داریم، برای یک ترافیک جدید مخرب‌بودن یا نبودن آن را تعیین کنیم. ابتدا ترافیک جدید را همانند روال قبلی بررسی کرده و پس از محاسبه ضریب همبستگی جریان‌های ترافیک، الگوریتم K-Means را روی آن اجرا می‌کنیم. همان‌طورکه در فصل قبل گفته شد، سپس فاصله بین خوشه‌های ترافیک ناشناخته را با خوشه‌های ترافیک بات‌نت و معمولی طبق رابطه (۸-۱۲) محاسبه می‌کنیم. نتایج محاسبات به‌صورت زیر می‌باشد:

جدول (۱۱). خوشه‌بندی ترافیک جدید

مرکز خوشه (V_i)	جمعیت خوشه ($\|\|c_i\|\|$)	فاصله میانگین (S_i^*)	جمعیت نسبی (P_i)
۰/۰۳	۲۶۳۷۶	۰/۰۴۸	۰/۲۱
۰/۱۸	۲۰۶۳۹	۰/۰۴۳	۰/۱۶
۰/۴۵	۳۲۲۱۲	۰/۰۳۷	۰/۲۶
۰/۶۳	۱۹۶۵۹	۰/۰۴۱	۰/۱۶
۰/۸۹	۲۵۸۶۴	۰/۰۳۲	۰/۲۱

ترافیک ناشناخته جدید حاوی ۸۰۰۰۰ بسته در بازه زمانی ۲۰۰۰ ثانیه‌ای است که ۵۰۰ جریان از آن استخراج شده‌است. برای ترافیک جدید بهترین تعداد مراکز همان‌طور که در جدول نشان داده شده ۵ مرکز می‌باشد.

جدول (۱۲). فاصله مراکز ترافیک جدید نسبت به ترافیک باتنت

D_1	۰/۰۲
D_2	۰/۱۰
D_3	۰/۰۵
D_4	۰/۰۶
D_5	۰/۰۹

پس مقدار T_A را برای ترافیک جدید نسبت به ترافیک باتنت، براساس داده‌های فوق محاسبه می‌کنیم:

$$T_A = (0.02*0.04)+(0.1*0.1)+(0.14*0.04) \quad (۱-۹)$$
$$+(0.06*0.04)+(0.09*0.06) = 242*10^{-4}$$

با انجام همین اعمال روی ترافیک معمولی، مقدار $644*10^{-4}$ برای T_A نسبت به ترافیک معمولی به‌دست آمد. طبق آن‌چه قبلاً هم گفته‌شد، هرچه این مقدار کم‌تر باشد به‌معنی شبیه‌تربودن دو ترافیک است. پس نتیجه می‌گیریم که ترافیک جدید باید از نوع باتنت باشد. و در واقع هم همین‌طور است. ترافیک تست‌شده، ترافیک یک باتنت است که در دانشگاه CTU گرفته شده است[1].

همان‌طورکه قبلاً گفته شد یکی از چالش‌هایی که در زمینه پژوهش پیرامون باتنت با آن مواجه هستیم داشتن مواد اولیه است که در اینجا مواد اولیه ما ترافیک باتنت واقعی، ترافیک ازدحام ناگهانی واقعی و ترافیک معمولی شبکه است. به‌دست آوردن ترافیک معمولی شبکه کار مشکلی نیست اما برای ترافیک باتنت از ترافیک‌هایی که توسط دانشگاه CTU ارائه شده استفاده کرده‌ایم و برای ترافیک ازدحام ناگهانی از ترافیک دانشگاه بزرگمهر در روز انتخاب واحد و در بازه‌های زمانی مختلف استفاده شده است. الگوریتم ارائه‌شده را با استفاده از ترافیک‌های موجود آزمایش کردیم که در ۹۲ درصد موارد، الگوریتم به‌درستی شناسایی کرد. همین کار را برای الگوریتم ارائه‌شده توسط شوئی یو و همکاران انجام دادیم که نتیجه آزمایش این بود که الگوریتم مذکور در ۶۳ درصد موارد ، کار شناسایی را به‌درستی انجام داد. مشکل دیگری که در الگوریتم شوئی یو و همکاران وجود دارد، تعیین حد آستانه مناسب است. مقدار حد آستانه بستگی به تعداد جریان‌ها و اندازه ترافیک دارد. درهرحال تعیین حدآستانه، خود مساله‌ای است که باید به‌دقت مورد بررسی قرارگیرد، چراکه اگر درست انتخاب نشود تمام نتایج تحت تأثیر این اشتباه قرارخواهند گرفت.

۱۰- ارزیابی

برای ارزیابی الگوریتم از روش 10 Folds Cross Validation که به‌طور گسترده در پژوهش‌ها به‌کار گرفته می‌شود، استفاده‌شده است. این روش به‌منظور آموزش و تست، مجموعه داده را به ۱۰ زیرمجموعه تصادفی تقسیم‌بندی می‌کند. یک مجموعه برای تست و ۹ مجموعه دیگر برای آموزش به‌کار گرفته می‌شوند. فرایند آموزش و تست آن‌قدر تکرار می‌شود تا زمانی که تمام ۱۰ زیرمجموعه، یک‌بار به‌عنوان مجموعه تست و ۹ زیرمجموعه دیگر به‌عنوان مجموعه آموزشی، مورداستفاده قرار گیرند.

مزیت این روش در این است که تمام داده‌های موجود در مجموعه داده به‌عنوان داده آموزشی و تست، درنظر گرفته می‌شوند. مجموعه داده، به دو کلاس ترافیک باتنت نظیربه‌نظیر و ترافیک معمولی شبکه طبقه‌بندی می‌شوند، سپس عملکرد الگوریتم، مورد ارزیابی قرار می‌گیرد. پارامترهایی که برای ارزیابی مورد استفاده قرار می‌گیرند، پارامترهای F-Measure، Precision، Recall و Accuracy است که در ادامه شرح داده شده‌اند. جدول (۱۳) پارامترهای ارزیابی الگوریتم را نشان می‌دهد.

جدول (۱۳). پارامترهای ارزیابی الگوریتم

کلاس ترافیک پارامتر ارزیابی	ترافیک باتنت نظیربه نظیر	ترافیک معمولی شبکه
Recall	$\text{Recall}_{\text{Bot}}$	$\text{Recall}_{\text{Normal}}$
Precision	$\text{Precision}_{\text{Bot}}$	$\text{Precision}_{\text{Normal}}$
F_Measure	$\text{F-Measure}_{\text{Bot}}$	$\text{F\_Measure}_{\text{Normal}}$
Accuracy	Accuracy	

برای تعریف پارامترهای ارزیابی، ابتدا لازم است ماتریس درهم‌ریختگی معرفی شود. با این ماتریس می‌توان نتایج ارزیابی را بهتر نشان داد. ماتریس درهم‌ریختگی در جدول (۱۴) نشان داده شده است و پارامترهای آن به‌صورت زیر به‌دست می‌آیند:

۱- مثبت درست (TP)[2]: تعداد ترافیک‌های باتی که به‌عنوان ترافیک بات شناسایی شده‌اند.

۲- مثبت کاذب (FP)[3]: تعداد ترافیک‌های نرمالی که به‌عنوان ترافیک بات شناسایی شده‌اند.

۳- منفی درست (TN)[4]:تعداد ترافیک‌های نرمالی که به‌عنوان ترافیک نرمال شناسایی شده‌اند.

۴- منفی کاذب (FN)[5]: تعداد ترافیک‌های باتی که به‌عنوان ترافیک نرمال شناسایی شده‌اند.

جدول (۱۴). ماتریس درهم‌ریختگی

	بات شناسایی‌شده	نرمال شناسایی‌شده
ترافیک‌های بات	مثبت درست (TP)	منفی کاذب (FN)
ترافیک‌های نرمال	مثبت کاذب (FP)	منفی درست (TN)

1- http://mcfp.weebly.com/

2- True Positive
3- False Positive
4- True Negative
5- False Negative

ماتریس درهم‌ریختگی نشان می‌دهد که:

- هرچه پارامتر مثبت درست بیش‌تر باشد یا به‌عبارتی پارامتر منفی کاذب کم‌تر باشد، الگوریتم توانسته است تعداد ترافیک بات بیش‌تری را به‌درستی شناسایی کند.
- هرچه پارامتر منفی درست بیش‌تر باشد یا به‌عبارتی پارامتر مثبت کاذب کم‌تر باشد، الگوریتم توانسته است تعداد ترافیک نرمال بیش‌تری را به‌درستی شناسایی کند.
- باتوجه به این‌که در این پژوهش از روش 10 Folds Cross Validation برای آموزش و تست الگوریتم استفاده شده است، تمام داده‌های موجود در مجموعه داده، به‌عنوان داده تست درنظر گرفته می‌شوند، درنتیجه روابط زیر برقرار است:
- مثبت درست به‌علاوه منفی کاذب نشان‌دهنده تعداد کل ترافیک بات‌نت است.
- منفی درست به‌علاوه مثبت کاذب نشان‌دهنده تعداد کل ترافیک‌های نرمال است.

باتوجه به ماتریس درهم‌ریختگی، پارامترهای ارزیابی به‌صورت زیر تعریف می‌شوند:

۱- $\text{Recall}_{\text{Bot}}$: نشان‌دهنده این است که الگوریتم توانسته است چند درصد ترافیک بات را به‌درستی شناسایی کند. این پارامتر از نسبت تعداد ترافیک باتی که توسط الگوریتم به‌درستی به‌عنوان ترافیک بات شناسایی‌شده‌اند، به تعداد کل ترافیک‌های بات به‌دست می‌آید و از طریق رابطه (۱-۱۰) محاسبه می‌شود:

$$\text{Recall}_{\text{Bot}} = \frac{TP}{TP+FN} \qquad (۱-۱۰)$$

۲- $\text{Recall}_{\text{Normal}}$: نشان‌دهنده این است که الگوریتم توانسته است چند درصد ترافیک‌های نرمال را شناسایی کند. این پارامتر از نسبت تعداد ترافیک‌های نرمالی که توسط الگوریتم به‌درستی به‌عنوان ترافیک نرمال شناسایی‌شده‌اند، به تعداد کل ترافیک‌هایی که توسط الگوریتم به‌عنوان ترافیک نرمال شناسایی‌شده‌اند، به‌دست می‌آید و از طریق رابطه (۲-۱۰) محاسبه می‌شود:

$$\text{Recall}_{\text{Normal}} = \frac{TN}{TN+FP} \qquad (۲-۱۰)$$

۳- $\text{Precision}_{\text{Bot}}$: نشان‌دهنده این است که چند درصد ترافیک‌هایی که توسط الگوریتم به‌عنوان ترافیک بات شناسایی شده‌اند، واقعاً بات هستند. این پارامتر از نسبت تعداد ترافیک‌های باتی که توسط الگوریتم به‌درستی به‌عنوان ترافیک بات شناسایی‌شده‌اند، به تعداد کل ترافیک‌هایی که توسط الگوریتم به‌عنوان ترافیک بات شناسایی شده‌اند، به‌دست می‌آید و از طریق رابطه (۳-۱۰) محاسبه می‌شود.

$$\text{Precision}_{\text{Bot}} = \frac{TP}{TP+FP} \qquad (۳-۱۰)$$

۴- $\text{Precision}_{\text{Normal}}$: نشان‌دهنده این است که چند درصد ترافیک‌هایی که توسط الگوریتم به‌عنوان ترافیک نرمال شناسایی شده‌اند، واقعاً نرمال هستند. این پارامتر از نسبت تعداد ترافیک‌های نرمالی که توسط الگوریتم به‌درستی به‌عنوان ترافیک نرمال شناسایی‌شده‌اند، به تعداد کل ترافیک‌هایی که توسط الگوریتم به‌عنوان ترافیک نرمال شناسایی‌شده‌اند، به‌دست می‌آید و از طریق رابطه (۴-۱۰) محاسبه می‌شود.

$$\text{Precision}_{\text{Normal}} = \frac{TN}{TN+FN} \qquad (۴-۱۰)$$

۵- F_Measure: این پارامتر، ترکیبی از دو پارامتر Precision و Recall است و از طریق رابطه‌های (۵-۱۰) و (۶-۱۰) برای دو کلاس ترافیک بات‌نت نظیربه‌نظیر و ترافیک نرمال شبکه محاسبه می‌شود.

$$\text{F\_Measure}_{\text{Bot}} = 2 * \frac{Recall_{Bot} * Precision_{Bot}}{Recall_{Bot} + Precision_{Bot}} \qquad (۵-۱۰)$$

$$\text{F\_Measure}_{\text{Normal}} = 2 * \frac{Recall_{Normal} * Precision_{Normal}}{Recall_{Normal} + Precision_{Normal}} \qquad (۶-۱۰)$$

۶- Accuracy: این پارامتر نشان‌دهنده دقت کلی الگوریتم است و از نسبت تعداد کل ترافیک‌هایی که توسط الگوریتم به‌درستی شناسایی‌شده‌اند، به تعداد کل ترافیک‌ها به‌دست می‌آید. این پارامتر از طریق رابطه (۷-۱۰) محاسبه می‌شود.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (۷-۱۰)$$

برای ارزیابی الگوریتم ارائه‌شده توسط روش ارزیابی فوق ابتدا مقادیر ماتریس درهم‌ریختگی که توسط الگوریتم به‌دست آمده است، مورد بررسی قرار می‌گیرد. سپس عملکرد الگوریتم در شناسایی ترافیک بات‌نت‌های نظیربه‌نظیر و ترافیک نرمال شبکه براساس پارامترهای ارزیابی، موردبررسی و ارزیابی قرار می‌گیرد.

ماتریس درهم‌ریختگی نشان‌دهنده نتایج ارزیابی الگوریتم است. جدول (۱۵) مقادیر ماتریس درهم‌ریختگی را برای الگوریتم نشان می‌دهد.

جدول (۱۵). مقادیر ماتریس درهم‌ریختگی برای الگوریتم

	بات شناسایی‌شده	نرمال شناسایی‌شده
ترافیک‌های بات	۲۴	۶
ترافیک‌های نرمال	۳	۲۷

جدول (۱۶) نتایج ارزیابی عملکرد الگوریتم را در شناسایی ترافیک بات‌نت نظیربه‌نظیر براساس پارامترهای تعریف‌شده، نشان می‌دهد.

جدول (۱۶). نتایج ارزیابی الگوریتم در شناسایی ترافیک‌های بات نت نظیربه نظیر

$Precision_{Normal}$	٪۸۱	$F\_Measure_{Bot}$	٪۸۳/۸۰
$Precision_{Bot}$	٪۸۸	$F\_Measure_{Normal}$	٪۸۵/۲۶
$Recall_{Normal}$	٪۹۰	Accuracy	٪۸۵
$Recall_{Bot}$	٪۸۰		

نتایج ارزیابی نشان می‌دهد که عملکرد الگوریتم باتوجه به پارامترهای مختلف، متفاوت است. دو پارامتر Recall و Precision از اهمیت یکسانی برخوردار نیستند. Precision نشان‌دهنده این است که چند درصد ترافیک‌های باتی که الگوریتم به‌عنوان ترافیک بات شناسایی کرده است، واقعاً بات هستند. پارامتر Recall نیز مشخص می‌کند که الگوریتم توانسته است چند درصد ترافیک‌های بات را شناسایی کند. باتوجه به توضیحات زیر، پارامتر Recall از پارامتر Precision مهم‌تر است:

اگر Recall کم باشد، یعنی الگوریتم توانسته است درصد کمی از ترافیک‌های بات را شناسایی کند. به‌عبارت دیگر، تعداد زیادی از ترافیک‌های بات به‌عنوان ترافیک نرمال شناسایی شده‌اند. این برای امنیت سیستم بسیار خطرناک است، زیرا الگوریتم نتوانسته است چنین ترافیک‌های باتی را تشخیص دهد.

اگر Precision کم باشد، یعنی الگوریتم توانسته است درصد کمی از ترافیک‌های باتی را که واقعاً بات هستند، شناسایی کند. به‌عبارت دیگر، الگوریتم تعداد زیادی ترافیک نرمال را به‌عنوان ترافیک بات شناسایی کرده‌است. این برای امنیت سیستم خطر چندانی ایجاد نمی‌کند، فقط ممکن است موجب ناراحتی کاربر شود، زیرا ترافیک‌های نرمال به‌عنوان ترافیک بات شناسایی شده‌اند و ممکن است یک برنامه عادی، غیرفعال شود.

پارامتر Accuracy دقت کلی الگوریتم را در شناسایی ترافیک باتنت نظیربه‌نظیر و ترافیک نرمال شبکه، نشان می‌دهد. این پارامتر مشخص می‌کند که الگوریتم توانسته است چند درصد ترافیک‌های باتنت نظیربه‌نظیر و ترافیک‌های نرمال شبکه را به‌درستی شناسایی کند.

۱۱- نتیجه‌گیری

در این مقاله، ما سیستم تشخیص خود را که مبتنی بر تجزیه و تحلیل رفتار ترافیک و مبتنی بر جریان است، را ارائه دادیم. نکته‌ای که از آن برای تشخیص باتنت استفاده شد این است که جریان‌های حمله جاری در مقایسه با جریان‌های ازدحام ناگهانی، شباهت بیش‌تری به هم دارند؛ و طبق همین اصل، الگوریتمی ارائه‌شده که از ضریب هم‌بستگی جریان، به‌عنوان یک پارامتر شباهت برای جریان‌های مشکوک استفاده می‌شود. راه‌حل‌های دیگری برای مقابله با چنین حملاتی استفاده می‌شود که در آن از جورچین‌های گرافیکی برای تفاوت قائل‌شدن بین انسان و بات‌ها استفاده می‌کنند. بدی این روش این است که استفاده از پاسخ‌های انسانی، می‌تواند برای کاربر آزاردهنده باشد. پژوهش ارائه‌شده مشکلات فوق را برطرف می‌کند. در این پژوهش، شناسایی باتنت نظیربه‌نظیر در مرحله فرمان-کنترل از چرخه حیات، براساس تجزیه و تحلیل ترافیک انجام می‌شود؛ و از الگوریتم K-Means برای خوشه‌بندی ضرایب هم‌بستگی استفاده‌شده است و درنهایت براساس شباهت ترافیک جدید با جریان‌های باتنت می‌توان مخرب‌بودن ترافیک جدید را تشخیص داد. برای ارزیابی الگوریتم از روش 10 Folds Cross Validation که به‌طور گسترده در پژوهش‌ها به‌کار گرفته می‌شود، استفاده شده است. پارامتر Accuracy دقت کلی الگوریتم را در شناسایی ترافیک باتنت نظیربه‌نظیر و ترافیک نرمال شبکه، نشان می‌دهد. این پارامتر مشخص می‌کند که الگوریتم توانسته است چند درصد ترافیک‌های باتنت نظیربه‌نظیر و ترافیک‌های نرمال شبکه را به‌درستی شناسایی کند که مقدار این پارامتر برای الگوریتم ارائه شده ۸۵ درصد است. اما همان‌طورکه گفته شد، داده‌های اولیه که برای انجام آزمایش مورد نیاز است، بسیار محدود است، بنابراین نرخ تشخیصی که از آزمایش این مجموعه داده به‌دست می‌آید نرخ قابل اطمینانی نیست. و برای دقیق‌ترشدن نتایج باید آزمایش را روی تعداد بیش‌تری از مجموعه داده انجام داد. که مطمئناً اگر مجموعه داده مورد آزمایش تنوع و تعداد بیش‌تری داشته باشد نرخ تشخیص نیز بالاتر خواهد رفت. اما با این‌حال، مزیتی که این الگوریتم نسبت به الگوریتم‌های مشابه دارد، تشخیص ترافیک باتنت از ترافیک ازدحام ناگهانی است. که این ویژگی خاص باعث می‌شود الگوریتم پیشنهادی نسبت به الگوریتم‌های مشابه اهمیت بیش‌تری داشته باشد.

۱۲- پیشنهادها

باتوجه به این‌که در شناسایی باتنت‌ها نرخ خطا در تشخیص وجود دارد، نیاز است برای تکمیل کار رویکردی به‌منظور شناسایی باتنت‌های نظیربه‌نظیر در مرحله حمله از چرخه حیات ارائه شود. برخی باتنت‌ها ممکن است در مرحله فرماندهی و کنترل از چرخه حیات شناسایی نشوند، بنابراین نیاز است رویکردی وجود داشته باشد تا بتواند آن‌ها را در مرحله حمله از چرخه حیات شناسایی کند. باتوجه به این موضوع، در تحقیقات آینده قصد داریم رویکردی را برای شناسایی باتنت‌های نظیربه‌نظیر در تمامی مراحل چرخه حیات ارائه دهیم. این رویکرد بایستی بتواند علاوه‌بر شناسایی باتنت‌های نظیربه‌نظیر در مرحله فرماندهی و کنترل از چرخه حیات، آن‌ها را در مرحله حمله نیز شناسایی کند.

امروزه استفاده از دستگاه‌های سیار مانند موبایل در میان

افراد افزایش یافته است. دستگاه‌های سیار قابلیت اتصال به اینترنت را دارند و افراد ممکن است بیش‌تر وقت خود را با استفاده از این دستگاه‌ها در اینترنت بگذرانند. همان‌طورکه فن‌آوری در این زمینه پیشرفت کرده است و امروزه شاهد این هستیم که بیش‌تر افراد کارهای خود را با استفاده از گوشی‌های هوشمند انجام می‌دهند، مهاجمین و توسعه‌دهندگان بات‌نت از این فرصت استفاده کرده‌اند و فعالیت در محیط سیار را هدف قرار داده‌اند. بات‌نت‌هایی هستند که در محیط سیار فعالیت‌های مخرب خود را انجام می‌دهند. این بات‌نت‌ها به دستگاه‌های سیار افراد حمله می‌کنند تا به اهداف خود از قبیل سرقت اطلاعات مهم، دست یابند. باتوجه به این موضوع، نیاز است تا رویکردی برای شناسایی این بات‌نت‌ها نیز ارائه شود، بنابراین مسیر دیگر ادامه این پژوهش، کار بر روی شناسایی بات‌نت‌ها در محیط سیار است.

۱۳- مراجع

[1] A. Cole, M. Michael, and D. Noyes, "Botnets: The rise of the machines," In Proceedings on the 6th Annual Security Conference, 2007.

[2] B. Assadhan, M. José, and D. Lapsley, "Periodic Behavior in Botnet Command and Control Channels Traffic," IEEE, 2009.

[3] H. Choi, H. Lee, and H. Kim, "BotGAD: detecting botnets by capturing group activities in network traffic," In Proceedings of the Fourth International ICST Conference on communication System software and middleware, 2009.

[4] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, vol. 7, 2007.

[5] J. Govil and J. Govil, "Criminology of botnets and their detection and defense methods," Electro/Information Technology, 2007 IEEE International Conference, 2007.

[6] S. S. Silva, R. M. Silva, and R. C. Pinto, "Botnets: A survey," Computer Networks, vol. 57, no. 2, pp. 372-403, 2013.

[7] H. R. Zeidanloo, M. Safar, M. Zamani, P. Vahdani Amoli, and M. J. Z. Shooshtari, "A taxonomy of botnet detection techniques," Computer Science and Information Technology (ICCSIT), 3rd IEEE International Conference on, vol. 2, 2010.

[8] K.-S. Han and G. I. Eul, "A survey on p2p botnet detection," Proceedings of the International Conference on IT Convergence and Security , Springer Netherlands, pp. 589-593, 2012.

[9] A. H. Lashkari and S. G. Ghalebandi, "A Wide Survey on Botnet," Digital Information and Communication Technology and Its Applications , Springer Berlin Heidelberg, pp. 445-454, 2011.

[10] J. Leonard, S. Xu, and R. Sandhu, "A framework for understanding botnets," In Availability, Reliability and Security, 2009 ARES'09, International Conference on, 2009.

[11] S. Garg, A. K. Sarje, and S. K. Pedd, "Improved Detection of P2P Botnets through Network Behavior Analysis," Recent Trends in Computer Networks and Distributed Systems Security ,Springer Berlin Heidelberg, pp. 334-345, 2014.

[12] J. Han, K. Micheline, and P. Jian, "Data mining," southeast asia edition: Concepts and techniques, Morgan kaufmann, 2006.

[13] L. Xinying and W. Peizhi, "Data Mining Technology and its Application in Electronic Commerce," In Wireless Communications, Networking and Mobile Computing, 2008 WiCOM'08, 4th International Conference on, 2008.

[14] J. R. Binkley and S. Singh, "An algorithm for anomaly-based botnet detection," Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), pp. 43-48, 2006.

[15] H. Husna, S. Phithakkitnukoon, and S. Pa , "Behavior analysis of spam botnets," Communication Systems Software and Middleware and Workshops, 2008 COMSWARE, 3rd International Conference on, 2008.

[16] N. Pratik, S. Ray, C. Hota, and V. Venkatakrishnan , "Peershark: detecting peer-to-peer botnets by tracking conversations," In Security and Privacy Workshops (SPW), 2014 IEEE, 2014.

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

[18] S. Sherif, I. Traore, A. Ghorbani, S. Bassam, D. Zhao, W. Lu, J. Felix, and P. Hakimian, "Detecting P2P botnets through network behavior analysis and machine learning," In Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on, 2011.

[19] W. Tarng, L.-Z. Den, K.-L. Ou, and M. Chen, "The analysis and identification of P2P botnet's traffic flows," International Journal of Communication Networks and Information Security (IJCNIS), vol. 2, no. 3, 2011.

[20] F. Chen, M. Wang, Y. Fu, and J. Zeng, "New detection of peer-to-peer controlled bots on the host," Wireless Communications, Networking and Mobile Computing WiCom'09, 5th International Conference, Beijing, 2009.

[21] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," In Computing, Networking and Communications (ICNC), 2014 International Conference on, 2014.

[22] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang, "Discriminating DDoS attacks from flash crowds using flow correlation coefficient," Parallel and Distributed Systems, IEEE Transactions, vol. 23, no. 6, 2012.

[23] F. Kovács, C. Legány, and A. Babos, "Cluster validity measurement techniques," 6th International symposium of hungarian researchers on computational intelligence, 2005.

Journal Of Electronical & Cyber Defence
Vol. 3, No. 4, 2016, Serial No.12

Botnet Detection for Peer to Peer Networks

R. Azmi, M. Gholinezhad*, M. Saberi

* Alzahra University, Engineering College

(Received: 07/07/2015, Accepted: 12/01/2016)

ABSTRACT

Botnets are the latest types of internet-scale malware in recent years that has been the greatest threats to the web servers. Bot is an infected computer by a malware that are controlled remotely by one or more human factors without the user's knowledge. This controller agent called "bot master" and sometimes the infected system is called "victim". Peer to peer botnet is one type of botnets that use peer to peer protocols and detection of this type of botnet is more difficult than other types. Our suggested approach is a new method to detect such botnets. This approach uses network flow analysis and clustering method in data mining to detect peer to peer botnets. This approach is flow-based and compares the similarity between flows and K-Means clustering algorithm and eventually determines that the new traffic is an attack or not. This approach has good performance in detection of botnets in flash crowd traffic and this characteristic is distinction of the suggested algorithm and similar algorithms. Finally, the suggested approach has been tested with different traffic.

Keywords: Botnet, Botnet Detection, Bot, Peer to Peer Botnet, Network Flow Analysis.

* Corresponding Author Email: M.gholinezhad84@gmail.com