مجله علمی- پژوهشی "پدافند الکترونیکی و سایبری"
سال پنجم، شماره ۴، زمستان ۱۳۹۶، ص ۱۵- ۱

تشخیص شبکه بات با رویکرد تحلیل رفتاری جریان شبکه و بهره‌گیری از الگوریتم‌های داده‌کاوی

سعید پارسا[1*]، حامد مرتاضی[2]

۱- دانشیار، دانشگاه علم و صنعت ایران ۲- کارشناس ارشد کامپیوتر، دانشگاه آزاد اسلامی شبستر

(دریافت: ۹۵/۰۵/۰۴، پذیرش: ۹۵/۱۱/۲۵)

چکیده

«شبکه بات» شبکه‌ای از رایانه‌های آلوده متصل به اینترنت است که تحت مدیریت سرور فرماندهی و کنترل قرار دارد و برای حملات انکار سرویس، فرستادن هرزنامه و عملیات مخرب دیگر مورد استفاده قرار می‌گیرد. با وجود ویژگی‌های خاص هر شبکه بات، بات‌ها در داخل شبکه رفتارهای همسانی از خود نشان می‌دهند و این می‌تواند نقطه آغاز شناسایی یک بات در داخل شبکه باشد و با شناسایی این رفتار همگون می‌توان ترافیک تولیدی بات‌ها را از ترافیک عادی شبکه تفکیک کرد و از مشکلاتی مانند یافتن الگوریتم‌های رمزگشایی کانال‌های ارتباطی رمزنگاری‌شده در امان بود. رفتار همسان بات‌ها در داخل شبکه بات می‌تواند منجر به تولید ویژگی‌ها و خصیصه‌هایی شود که بتوان با تحلیل این ویژگی‌ها، جریان بدخواه را از جریان سالم تشخیص داد. منطق اصلی روش استفاده‌شده در این پژوهش بر این پایه استوار است که شبکه‌های بات، الگوهای ترافیکی قابل‌تشخیصی از خود به‌جای می‌گذارند که به کمک روش‌های یادگیری ماشین قابل شناسایی بوده و می‌توان ترافیک تولیدی توسط آن‌ها را از ترافیک عادی شبکه جدا کرد. در این مقاله، ویژگی‌ها و رفتار شبکه‌های بات مشهور همچون Weasel در جهت تولید خصیصه‌ها مطالعه شد. سپس، بعد از تهیه مجموعه داده‌های واقعی که ترکیبی از ترافیک سالم و ترافیک تولیدی توسط چندین شبکه بات مشهور است، جریان بسته‌ها در پنجره‌های زمانی ۳۰۰ ثانیه‌ای تحلیل شده و با توجه به الگوهای ترافیکی قابل تشخیص، خصیصه‌های مختلفی استخراج (تولید) شد. این خصیصه‌ها در ابزار وکا و به کمک الگوریتم‌های یادگیری ماشین داده‌کاوی شده و نتایج طبقه‌بندی به‌عنوان خروجی ارائه می‌شود. نتایج خروجی‌ها نشان‌دهنده نرخ تشخیص بالاتر در مقایسه با کارهای مشابه و در حدود ۹۹ ٪ می‌باشد. درنهایت نیز روشی برای شناسایی بلادرنگ شبکه‌های بات ارائه خواهد شد.

واژه‌های کلیدی: شبکه بات، تشخیص شبکه بات، سرور فرماندهی و کنترل، پنجره زمانی، یادگیری ماشین

۱- مقدمه

واژه بات از روبات برگرفته شده و در حقیقت یک کد دودویی بدخواه است که بر روی میزبان‌های آسیب‌پذیر اجرا شده و به مهاجم یا مدیر بات امکان می‌دهد تا از راه دور آن میزبان‌ها را با فرامین خود هدایت نماید. شبکه بات نیز به معنی شبکه ای از میزبان‌های آلوده به بات می‌باشد. زمانی‌که یک رایانه به بات آلوده می‌شود، دیگر قادر نخواهد بود در برابر دستورات مدیر بات مقاومت کرده یا از اجرای آن‌ها سرباز زند. درنتیجه، مهاجم می‌تواند از توان پردازشی میزبان‌های به تصرف درآمده، به‌صورت توزیع‌شده به نفع خود بهره‌برداری کرده و انواع مختلفی از حملات را به صورت هماهنگ و با قدرت تخریبی بسیار بالا بر روی قربانی سازمان‌دهی کند. این در حالی است که معمولاً هویت وی مخفی می‌ماند. اقداماتی که تا به امروز برای مقابله با فعالیت‌های مخرب شبکه‌های بات به‌ویژه در مقابله با حملات ارسال هرزنامه و حملات انکار سرویس انجام شده است را می‌توان به دو گروه واکنشی و جلوگیری تقسیم نمود [۱].

روش‌های واکنشی رایج‌ترین روش‌ها هستند. راهبرد مورد استفاده در این روش‌ها آن است که ابتدا فعالیت‌های مخرب تشخیص داده شوند و سپس اقدام مناسب برای کاهش ترافیک مخرب به مقداری قابل قبول، انجام گیرد. این روش‌ها دارای دو عیب عمده می‌باشند: ۱) نیاز به ساخت یک زیربنای کامل با قدرت پردازشی قابل‌توجه و هم‌چنین فضای ذخیره‌سازی داده برای تحلیل حجم عظیمی از اطلاعات جمع‌آوری‌شده. ۲) با توجه به این‌که حمله در زمان تشخیص هم‌چنان درحال اجرا است، کاربران عادی حداقل از بخشی از تأثیرات منفی آن تا انجام واکنش مناسب، رنج خواهند برد. روش‌های جلوگیری، روش‌هایی هستند که سعی می‌کنند احتمال انجام فعالیت‌های مخرب را تا حد ممکن کاهش دهند. این

* رایانامه نویسنده مسئول: parsa@iust.ac.ir

راه‌کارها می‌توانند شامل افزایش منابع کاربران و یا ایجاد تغییر در زیربنای شبکه به گونه‌ای باشد که کاربران را وادار به احراز هویت کند. هرچند که در مقابله با این راه‌کارها، مهاجمان نیز می‌توانند منابع و ابزارهای خود را بهبود بخشند. در این مقاله رویکرد ارائه‌شده، برمبنای روش واکنشی بوده و بدخواه‌بودن یا سالم‌بودن بسته‌های آزمایشی بدین صورت تشخیص داده می‌شود که پس از تحلیل جریان بسته‌ها و استخراج و تولید خصیصه‌های وابسته به جریان، این خصیصه‌ها در جهت شناسایی الگوهای رفتاری مشابه، داده‌کاوی شده و درنهایت، الگوی بسته‌های سالم از بسته‌های مخرب متمایز می‌شود.

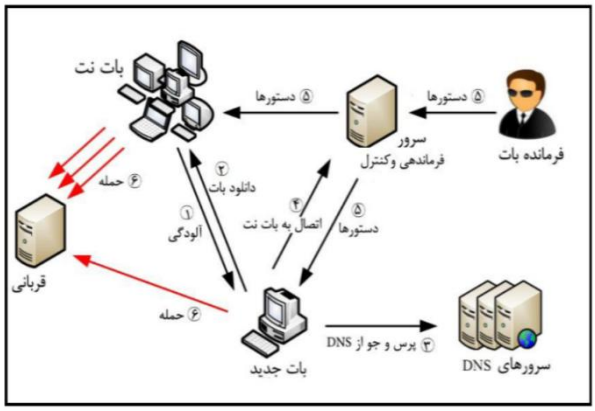
در بخش‌های دوم و سوم این مقاله به تشریح ویژگی‌های شبکه‌های بات و انواع روش‌های تشخیص آن‌ها پرداخته می‌شود. در بخش چهارم، نمونه کارهای مشابه در جهت تشخیص شبکه‌های بات مطالعه می‌شود. در بخش پنجم، روش پیشنهادی تشخیص شبکه‌های بات توضیح داده می‌شود. در بخش ششم، شرایط و نحوه انجام آزمایش‌ها بیان‌شده و در نهایت در بخش هفتم، نتایج ارزیابی به همراه مقایسه آن با کارهای مشابه ارائه خواهد شد.

۲- شبکه‌های بات

«شبکه بات» شبکه‌ای از رایانه‌های آلوده متصل به اینترنت است که تحت مدیریت سرور فرماندهی و کنترل قرار دارد و برای حملات انکار سرویس، فرستادن هرزنامه و عملیات مخرب دیگر مورد استفاده قرار می‌گیرد. ممکن است شبکه‌های بات دارای کارکردهای قانونی نیز باشند، ولی در اغلب موارد با فعالیت‌های مجرمانه برای انتشار هرزنامه، بدافزار یا حملات سرقت هویت در ارتباط هستند [۲]. اندازه یک شبکه بات، به پیچیدگی و تعداد رایانه‌های استخدام‌شده در آن بستگی دارد. معمولاً کاربران کامپیوترها از این موضوع که دستگاه‌هایشان از راه دور کنترل‌شده و مورد سوءاستفاده قرار می‌گیرند اطلاعی ندارند. در مقایسه شبکه بات با بدافزارهای موجود مانند کرم و ویروس، وجود کانال‌های فرماندهی و کنترل، تفاوت کلیدی است؛ چون بات‌ها تحت کنترل مهاجم، دستور را دریافت و رفتارهای مخرب انجام می‌دهند [۳]. برخلاف سایر بدافزارها، شبکه‌های بات چرخه حیات شفاف‌تری دارند که می‌تواند به سه مرحله اصلی شکل‌گیری، فرمان و کنترل، و حمله تقسیم شود. شکل (۱) شمای کلی یک شبکه بات و مفاهیم مرتبط با آن را نمایش می‌دهد.

بات[1]: کلمه بات از کلمه روبات مشتق شده است. بات‌ها طراحی شده‌اند تا برخی از توابع از قبل تعریف‌شده را به‌صورت خودکار انجام دهند. به‌عبارت‌دیگر، بات‌های منفرد برنامه‌های نرم‌افزاری هستند که روی رایانه میزبان اجرا می‌شوند و موجب می‌شوند مهاجم فعالیت‌های میزبان را از راه دور کنترل کند [۴]. برخی اوقات به‌جای کلمه بات، اصطلاح زامبی[2] به‌کار برده می‌شود.

شبکه بات: شبکه بات، شبکه‌ای از رایانه‌های آلوده به نام بات‌ها است که تحت کنترل مهاجم قرار دارند و برای حملات انکار سرویس توزیع‌شده، کلاه‌برداری، تقلب کلیک و انتشار بدافزارها مورد استفاده قرار می‌گیرند. شبکه بات را ارتش زامبی‌ها نیز می‌گویند [۲ و ۵].



شکل (۱): شمای کلی یک شبکه بات و مفاهیم مرتبط با آن بر مبنای IRC [۶]

سرور فرماندهی و کنترل: بات دستورات خود را از سرور فرماندهی و کنترل که توسط مهاجم هدایت می‌شود، دریافت می‌نماید. استفاده از این سرور موجب گمنامی مهاجم می‌شود تا به‌راحتی قابل ردیابی نباشد.

فرمانده بات یا مهاجم[3]: مهاجم به فردی گفته می‌شود که تمامی کارهای یک شبکه شبکه بات -از ایجاد تا کنترل- را به دست دارد؛ بدین ترتیب که بات را پیکربندی و پیاده‌سازی می‌نماید، سپس بات را برروی رایانه قربانی نصب‌کرده و درنهایت، بات‌ها را از طریق کانال کنترلی هدایت و رهبری کرده و دستورهای حمله را صادر می‌نماید.

۳- طبقه‌بندی روش‌های تشخیص شبکه بات

شکل (۲)، طبقه‌بندی روش‌های مختلف تشخیص شبکه بات براساس نوع پیاده‌سازی آن‌ها را نمایش می‌دهد. در ادامه، این روش‌ها به‌صورت اجمالی توضیح داده شده است.

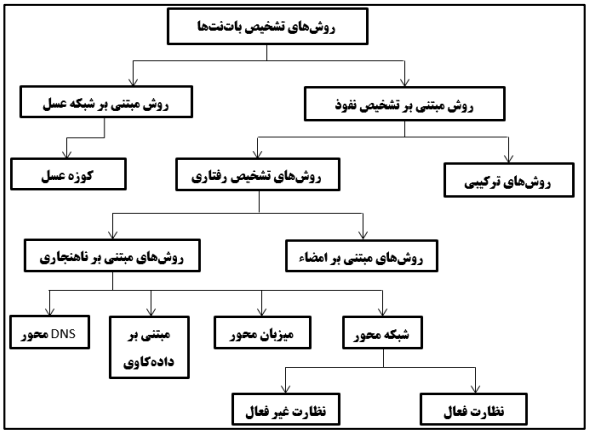
1- Bot

2- Zombie
3- Botmaster

۳-۱- تله عسل

یک تله عسل[1] یا یک شبکه عسل برای جمع‌آوری اطلاعات بات‌ها به‌منظور انجام تحلیل‌های مختلف به کار گرفته می‌شود. تله‌های عسل در جامعه امنیتی تحت عنوان دستگاه‌های کامپیوتری آسیب‌پذیر شناخته می‌شوند که در مکان خاصی از شبکه قرار می‌گیرند و حمله‌کنندگان را تحریک به حمله می‌کنند. بعد از حمله، کارکنان امنیتی سیستم با استفاده از مهندسی معکوس اقدام به بررسی بار مفید[2] بات و نحوه رفتار آن می‌کنند [۷]. تله‌های عسل هیچ‌گونه محصول تولیدی ندارند. بنابراین، هرگونه ارتباط میان آن‌ها و دیگر اعضای شبکه، ترافیک مشکوکی است که باید به‌دقت بررسی شود. این روش، نقاط ضعفی دارد که نمی‌توان از آن به‌عنوان یک روش مستقل برای تشخیص شبکه‌های بات استفاده کرد. ازجمله این ضعف‌ها می‌توان به موارد زیر اشاره کرد[۸]:

- مقیاس‌پذیری محدود به علت نیاز به تجهیزات سخت‌افزاری قدرتمند مانند مسیریاب‌ها و دستگاه‌های کامپیوتری پیچیده.
- عدم توانایی پیداکردن حملات اینترنتی به علت ایزوله‌بودن.
- چالش کشف تله‌های عسل آلوده که به‌عنوان دام قرار داده شده‌اند.
- احتمال آسیب به تله‌های عسل برای تخریب سایر بخش‌های شبکه وجود دارد.



شکل (۲). طبقه‌بندی روش‌های مختلف تشخیص شبکه بات

۳-۲- روش‌های مبتنی بر تشخیص نفوذ

یک سیستم تشخیص نفوذ، یک نرم‌افزار کاربردی یا یک ماشین سخت‌افزاری است که برای نظارت بر سرویس‌های سیستم به‌منظور کشف فعالیت‌های بدخواهانه، نقض امنیت و گزارش آن به مدیر سایت استفاده می‌شود. روش‌های مبتنی بر تشخیص نفوذ به دو بخش مبتنی بر امضاء[3] و مبتنی بر ناهنجاری[4] تقسیم می‌شود که در ادامه توضیح داده خواهد شد.

۳-۲-۱- روش‌های مبتنی بر امضاء

در این روش، سیستم تشخیص نفوذ شامل امضاءها و الگوهای شبکه‌های باتی مشهور است تا با مقایسه جریان‌های عبوری از یک ایستگاه با این امضاءها، بات‌بودن یا نبودن آن ایستگاه را گزارش دهد. عمده‌ترین عیب این روش، نیاز به به‌روزرسانی دوره‌ای مخزن امضاءها برای تشخیص شبکه‌های باتی جدید می‌باشد و چون سرعت به‌روزرسانی امضاءها همیشه کندتر از شبکه‌های باتی جدید است، لذا سیستم‌هایی که از این روش برای تشخیص شبکه‌های بات استفاده می‌کنند، معمولاً یک‌قدم از بات‌های جدید عقب هستند. نمونه بارز این نوع سیستم‌ها SNORT[5] می‌باشد. SNORT یک برنامه‌ی متن‌باز است که به سه روش قابل تنظیم است: سیستم تشخیص نفوذ، Sniffer و Packet Logger.

۳-۲-۲- روش‌های مبتنی بر ناهنجاری

ایده اصلی این روش بر این پایه استوار است که کل ترافیک غیرعادی عبوری از شبکه مانند ترافیک عبوری از درگاه‌های غیرمعمول، تأخیر بالا، بار ترافیکی غیرعادی و ... تحلیل و در جهت شناسایی ترافیک بدخواه از ترافیک سالم استفاده شود [۹]. به‌عبارت دیگر، رفتارهای معمول و عادی شبکه مطالعه و براساس آن ترافیک غیرعادی شبکه از ترافیک عادی جداسازی می‌شود. این روش خود به چهار بخش تقسیم می‌شود [۱۰]:

۱) روش میزبان محور[6]: در این روش، تک‌تک ایستگاه‌های موجود در شبکه در جهت کشف فعالیت‌های بدخواهانه نظارت می‌شوند. به دلیل این‌که در این روش باید تمام ایستگاه‌های موجود در شبکه به ابزارهای پیشرفته مانیتورینگ مانند آنتی‌ویروس‌ها مجهز شوند، مقیاس‌پذیری و به‌تبع آن کارایی این روش پایین می‌آید.

۲) روش شبکه محور[7]: در این روش، برخلاف روش میزبان محور، جریان عبوری از کل شبکه مورد نظارت قرار می‌گیرد. این روش به دو شکل انجام می‌شود ؛ نظارت فعال و نظارت غیرفعال. در نظارت فعال، بسته‌های آزمایشی در جهت سنجش کارایی و مقاومت شبکه، بدون تأثیر در کارایی آن به داخل شبکه تزریق‌شده و مکان‌های

1- HoneyNet (HoneyPot)
2- Payload
3- Signature-Based
4- Anomaly-Based
5- (www.snort.org)
6- Host-Base
7- Network-Base

آسیب‌پذیر شبکه شناسایی می‌شود. در نظارت غیرفعال، کل جریان عبوری از شبکه به سمت یک درگاه سخت‌افزاری خاص هدایت شده و کلیه بسته‌های این جریان برای تشخیص فعالیت‌های بدخواهانه مورد وارسی قرار می‌گیرند.

۳) **روش مبتنی بر داده‌کاوی**: یکی از روش‌هایی که برمبنای نظارت غیرفعال بر ترافیک شبکه کار می کند بحث داده کاوی است. در این روش، ابتدا الگوهای ارتباطی مشابه دسته‌بندی می‌شوند و سپس بررسی برای میزبان‌هایی که براساس این الگوها فعالیت می کنند؛ آغاز می شود. این روش مستقل از پروتکل ها و ساختارهای شبکه‌های بات کار می کند.

۴) **روش DNS محور**: یکی از مکانیسم‌های برقراری ارتباط با سرور فرماندهی و کنترل و ارسال پیام فعال‌بودن، استفاده از DNS است. در شبکه‌های باتی که از این روش استفاده می کنند، تعداد مشخصی از بات‌ها در زمان یکسان شروع به فعالیت کرده و حتی می توانند سرور فرماندهی و کنترل خود را نیز تغییر دهند؛ این کار باعث تولید ترافیک DNS می‌شود که یک رفتار گروهی بوده و می‌توان از آن برای پیداکردن ترافیک‌های DNS مرتبط با شبکه بات استفاده کرد. با توجه به این‌که در این مقاله جریان بسته‌های عبوری از شبکه تحلیل شده است، لذا از روش نظارت غیرفعال به کمک داده‌کاوی برای تشخیص شبکه‌های بات استفاده می‌شود.

۴- نمونه کارهای انجام‌شده

با مطالعه انواع روش‌های شناسایی شبکه‌های بات، این نتیجه حاصل شد که باید برای مقایسه و تعیین نقاط ضعف و قوت روش‌های مختلف، شاخص‌هایی تعیین‌شده و براساس آن نتیجه‌گیری نمود. برای مقایسه کارهای گذشته در تشخیص شبکه‌های بات از پنج شاخص موجود در جدول (۱)، استفاده شده است.

جدول(۱): شاخص‌های تشخیص شبکه‌های بات

شاخص	توضیحات
تشخیص شبکه بات ناشناخته	نشان‌دهنده قابلیت تشخیص شبکه‌های باتی جدید و حملات ناشناخته می‌باشد.
مستقل از ساختار و پروتکل	نشان‌دهنده قابلیت تشخیص ترافیک فرماندهی و کنترل درصورتی‌که حتی مدیر بات نوع پروتکل و ساختار سرور فرماندهی و کنترل را تغییر دهد، می‌باشد.
مقدار False Positive	نشان‌دهنده تشخیص ترافیک شبکه بات به‌صورت غلط و غیر شبکه بات می‌باشد که خطرناک است.
مقدار نرخ تشخیص	نشان‌دهنده میزان دقت تشخیص شبکه بات در روش مورد نظر می‌باشد.
تشخیص بات‌های رمزنگاری‌شده	نشان‌دهنده امکان تشخیص شبکه‌های باتیی با ارتباطات فرماندهی و کنترل رمزنگاری‌شده می‌باشد.
تشخیص بلادرنگ (نظارت فعال)/ غیربلادرنگ (نظارت غیرفعال)	نشان‌دهنده نوع تشخیص روش پیشنهادی می‌باشد که می‌تواند بلادرنگ بوده یا از نظارت غیرفعال استفاده شود.

جدول (۲) نمونه‌هایی از کارهای انجام‌شده در راستای تشخیص شبکه‌های بات را با توجه به شاخص‌های تعیین‌شده، طبقه‌بندی می‌کند. با توجه به مقدار شاخص‌های تشخیص مربوط به این روش‌ها، روش ارائه‌شده در این مقاله نرخ تشخیص بالاتر و مثبت کاذب کمتری داشته و قادر به تشخیص بلادرنگ باتنت‌ها نیز می‌باشد.

۵- روش پیشنهادی تشخیص شبکه‌های بات

در این مقاله برای تشخیص شبکه‌های بات از روش تحلیل جریان[1] استفاده شده است. با وجود ویژگی‌های خاص هر شبکه بات، بات‌ها در داخل شبکه‌های بات، رفتارهای همسانی از خود نشان می‌دهند و این می‌تواند نقطه آغاز شناسایی یک شبکه بات در داخل شبکه باشد. با شناسایی این رفتار همگون، می‌توان ترافیک تولیدی شبکه‌های بات را از ترافیک عادی شبکه تفکیک کرد و از مشکلاتی مانند یافتن الگوریتم‌های رمزگشایی کانال‌های ارتباطی رمزنگاری‌شده در امان بود. رفتار همسان بات‌ها در داخل شبکه‌های بات می‌تواند منجر به تولید ویژگی‌ها[2] و خصیصه‌هایی[3] شود که بتوان با تحلیل این ویژگی‌ها، جریان بدخواه را از جریان سالم تشخیص داد. همچنین، می‌توان روشی ارائه کرد که با توجه به گروه‌بندی بسته‌های با تعداد مناسب (و بسیار کم) در جریان‌ها، بتوان پیش‌زمینه طراحی یک سیستم بلادرنگ برای تشخیص شبکه‌های بات را فراهم کرد. تمامی این نتایج به‌علت تولید خصیصه‌ها و ویژگی‌های مناسب و تحلیل این ویژگی‌ها به‌دست آمده است.

1- Flow Analysis

2- Attributes

3- Features

۵-۱- مفروضات

قبل از تشریح روش ارائه‌شده در این مقاله، تعاریف اولیه‌ای که در این تحقیق مورداستفاده قرار خواهد گرفت، بیان می‌شود :

جریان (Flow): مجموعه بسته‌های رد و بدل‌شده مابین دو آدرس IP منحصربه‌فرد در پنجره زمانی T (یا شامل تعداد محدود و مشخص از بسته‌ها) و با استفاده از یک جفت درگاه و پروتکل لایه انتقال[1] UDP یا TCP است [۱۱].

پنجره زمانی: جریان بسته‌های رد و بدل‌شده مابین دو آدرس IP، در پنجره‌های زمانی مشخصی مورد وارسی قرار می‌گیرند زیرا ممکن است این ارتباط از چند ثانیه تا چندین روز به طول انجامد. هنگام انتخاب مقدار عددی این پنجره زمانی نیز باید به این نکته توجه شود که اگر مقدار بسیار کوچکی انتخاب شود ممکن است به‌دست‌آوردن ویژگی‌های منحصربه‌فرد ترافیک که تنها در طی یک دوره طولانی‌تر از زمان آشکار می‌شود، امکان‌پذیر نباشد. درصورتی‌که پنجره زمانی مقدار بزرگی انتخاب شود، تا حصول این فاصله زمانی هیچ ویژگی استخراج نخواهد شد و درنتیجه، عملیات تشخیص شبکه بات به طول خواهد انجامید. در این تحقیق، مقدار مناسب برای پنجره زمانی با توجه به تجربه و مطالعه کارهای مشابه مانند [۳۶ و ۳۹]، ۳۰۰ ثانیه انتخاب شده است.

۵-۲- خصیصه‌ها و ویژگی‌های مبتنی بر جریان

یک ویژگی یا یک خصیصه به مشخصه‌ای از یک جریان بسته در پنجره زمانی T اطلاق می‌شود که می‌تواند یک مقدار عددی[2] یا غیرعددی[3] داشته باشد. اگرچه تعداد زیادی از خصیصه‌های مبتنی بر جریان در جهت تشخیص انواع مختلف شبکه‌های بات پیشنهاد شده است، امّا هنوز به‌طور قطعی نمی‌توان در مورد ارزش واقعی و تعداد مناسب این خصیصه‌ها نتیجه‌گیری کرد.

جدول (۲): نمونه کارهای انجام‌شده در راستای تشخیص شبکه‌های بات

نویسنده/نام طرح پیشنهادی	نوع روش تشخیص				شاخص تشخیص					
	مبتنی بر امضاء	مبتنی بر نامنجاری	مبتنی بر DNS	مبتنی بر داده‌کاوی	تشخیص شبکه بات ناشناخته	مستقل از ساختار و پروتکل	مقدار False Positive	مقدار نرخ تشخیص	تشخیص بات‌های رمزنگاری‌شده	تشخیص بلادرنگ
Snort (2006) [14]	✓				×	×	بالا	ضعیف	×	×
Wang و همکاران [15] (2009)	✓				×	×	بالا	ضعیف	×	×
Botzilla (2010) [16]	✓				×	×	پایین	متوسط	×	×
BotSwat (2007) [17]		✓			✓	×	پایین	٪ ۸۰	×	×
BotHunter (2007) [18]		✓			✓	✓	بالا	ضعیف	×	×
BotSniffer (2008) [19]		✓			✓	×	پایین	متوسط	✓	×
BotMiner (2008) [20]		✓		✓	✓	✓	پایین	خوب	✓	✓
Strayer و همکاران [21] (2008)		✓		✓	✓	×	بالا	٪ ۷۰	×	×
BotProbe (2009) [22]		✓			×	×	بالا	٪ ۷۵	×	×
BotGAD (2009) [23]			✓		✓	×	بالا	ضعیف	×	✓
Shahrestani و همکاران [24] (2009)			✓		✓	✓	بالا	ضعیف	×	×
Masud و همکاران [25] (2008)		✓		✓	✓	✓	پایین	٪ ۹۸	✓	×
Saad و همکاران [13] (2011)		✓		✓	✓	✓	پایین	٪ ۹۷	✓	✓
Bilge و همکاران [26] (2012)		✓		✓	✓	✓	پایین	٪ ۷۰	✓	×
Zhao و همکاران [12] (2013)		✓		✓	✓	✓	متوسط	٪ ۹۸	✓	×
عزمی و همکاران (۱۳۹۴) [27]		✓		✓	✓	✓	متوسط	٪ ۹۰	×	×
فتحیان و همکاران (۱۳۹۵) [28]		✓		✓	✓	✓	متوسط	٪ ۹۸	✓	×

1- Transport Layer
2- Numeric
3- Nominal

در این مقاله سعی شده است تا با مطالعه رفتارهای مختلف شبکه‌های باتی مشهور مانند Storm، Waledac، Nugache و ... ، ویژگی‌ها و خصیصه‌های متفاوتی از بسته‌های نمونه، تولید و استخراج شده و به‌عنوان بردار ورودی روش‌های یادگیری ماشین در جهت شناسایی شبکه‌های بات به کار رود. منطق اصلی روش استفاده‌شده در این پژوهش، بر این پایه استوار است که شبکه‌های بات الگوهای ترافیکی قابل‌تشخیصی از خود به‌جای می‌گذارند که به کمک روش‌های یادگیری ماشین قابل شناسایی بوده و می‌توان ترافیک تولیدی توسط شبکه‌های بات را از ترافیک عادی شبکه جدا کرد. در ادامه، منطق تولید خصیصه‌ها از بسته‌های نمونه توضیح داده شده است.

۵-۲-۱- آدرس‌های آی‌پی فرستنده و گیرنده جریان

آدرس آی‌پی فرستنده و گیرنده جریان به‌صورت گسترده در دستگاه‌های تشخیص نفوذ استفاده می‌شود. این خصیصه به‌عنوان تعیین‌کننده تعداد ارتباطات مجزا استفاده شده و نیز در محاسبات سایر خصیصه‌ها کاربرد دارد. این ویژگی جزء بردار خصیصه‌ها در ورودی الگوریتم‌های یادگیری ماشین به‌کار برده نمی‌شود. در این پژوهش، از یک لیست سیاه شامل آدرس آی‌پی‌های مربوط به شبکه‌های باتی مشهور و یک لیست سفید شامل آدرس آی‌پی‌های ترافیک عادی شبکه برای برچسب‌گذاری مجموعه داده آزمایشی استفاده شده است.

۵-۲-۲- درگاه‌های مربوط به فرستنده و گیرنده جریان

در حالت کلی، این ویژگی برای گروه‌بندی مناسب بسته‌ها در جریان‌ها به‌کار گرفته می‌شود. این ویژگی، خصیصه مناسبی برای استفاده به‌عنوان بردار ورودی الگوریتم یادگیری ماشین نیست زیرا به‌عنوان مثال، شبکه باتی مانند Nugache به‌صورت دوره‌ای درگاه خود را عوض می‌کند و شناسایی این شبکه بات به کمک ویژگی درگاه، عملاً امکان‌پذیر نخواهد بود.

۵-۲-۳- پروتکل

همانند ویژگی درگاه، پروتکل نیز برای گروه‌بندی ترافیک‌های هم‌شکل به کار گرفته می‌شود. همچنین مهم‌ترین استفاده از این خصیصه، جداسازی ترافیک (جریان) مرتبط و غیرمرتبط می‌باشد. به‌عنوان مثال، اکثر شبکه‌های بات از پروتکل‌های TCP و UDP به‌عنوان پروتکل‌های لایه ارتباطی استفاده می‌کنند و از ویژگی پروتکل در جهت جداسازی این ترافیک‌ها استفاده می‌شود.

۵-۲-۴- تعداد بسته‌های ردوبدل شده در هر جریان

بات‌ها سعی می‌کنند تا ارتباط‌شان با سرور فرماندهی و کنترل و نیز مدیر بات همیشه در حالت برخط و زنده باقی بماند، لذا معمولاً تعداد زیادی بسته را در یک بازه زمانی ارسال می‌کنند. این ویژگی کمک شایانی برای شناسایی این الگوی ترافیکی می‌کند.

۵-۲-۵- تعداد بسته‌ها با بار مفید به طول صفر

با توجه به توضیح ارائه‌شده در مورد خصیصه قبلی، بات‌ها برای زنده نگه‌داشتن ارتباط خود با کانال‌های فرماندهی و کنترل، تعداد زیادی بسته را در یک بازه زمانی ارسال می‌کنند که این بسته‌ها معمولاً دارای بار مفید با طول صفر می‌باشند. برای شناسایی این رفتار، از خصیصه تعداد بسته‌ها با بار مفید به طول صفر استفاده شده است.

۵-۲-۶- تعداد بسته‌ها با بار مفید با طول کوچک/درصد تعداد بسته‌های با بار مفید با طول کوچک

با توجه به این‌که شبکه‌های بات سعی در مخفی نگه‌داشتن ارتباطات خود دارند، به همین دلیل، ترافیک رد و بدل‌شده بین اجزای شبکه‌های بات دارای بار مفید با طول بسیار کم می‌باشد. این درحالی است که ارتباطات عادی شبکه از این قاعده پیروی نمی‌کند. لذا برای شناسایی این الگوی ترافیکی از این خصیصه‌ها استفاده شده است.

۵-۲-۷- نرخ تعداد بسته‌های ورودی به تعداد بسته‌های خروجی

به علت وجود الگوهای ترافیکی خاص در شبکه‌های بات، تعداد بسته‌های ورودی و خروجی در یک جریان و در یک بازه زمانی مشخص، از نظم خاصی برخوردار است ولی در مورد ترافیک عادی شبکه نرخ مابین تعداد بسته‌های ورودی به تعداد بسته‌های خروجی دارای رفتار مشخص و قابل پیش‌بینی نیست. لذا از این خصیصه به‌منظور تشخیص رفتارهای شبکه‌های بات در مورد ارسال و دریافت بسته‌ها استفاده شده است.

۵-۲-۸- تعداد بسته‌های ورودی و تعداد بسته‌های خروجی

در برخی شبکه‌های بات مانند Weasel، ارتباط داخل شبکه بات برای یافتن بات‌های نظیر به نظیر، به شکلی انجام می‌شود که برخی جریانات دارای تعداد بسته‌های ورودی و یا خروجی صفر می‌باشند. در این حالت، عملاً خصیصه نرخ تعداد بسته‌های ورودی به تعداد بسته‌های خروجی بدون استفاده خواهد بود. برای حل این مشکل، از این دو خصیصه به‌صورت مجزا استفاده شد.

۵-۲-۹- مدت‌زمان جریان

یکی از خصیصه‌های مهم و پرکاربرد در زمینه شناسایی شبکه‌های بات –به‌کمک تحلیل رفتاری ترافیک شبکه- ویژگی مدت‌زمان جریان می‌باشد. شبکه باتی مانند Weasel در ارتباطات خود، بین مدیر بات و بات‌ها، بسیار کند عمل می‌کند و بسته‌های ارسالی با فاصله‌های زمانی کوتاه و پراکنده ارسال می‌شود تا سیستم‌های

کشف شبکه بات قادر به شناسایی آن نباشند. این درحالی است که ترافیک سالم شبکه از این نظم تبعیت نمی‌کند و لذا از خصیصه مدت‌زمان جریان می‌توان برای شناسایی الگوهای زمانی شبکه‌های بات استفاده کرد.

۵-۲-۱۰- طول اولین بسته جریان

بسیاری از شبکه‌های بات اطلاعات خاصی مانند اطلاعات پروتکل و یا اطلاعات مربوط به کانال‌های ارتباطی خود را در اولین بسته از جریان قرار می‌دهند و استفاده از این ویژگی می‌تواند کمک به‌سزایی در تشخیص انواع مختلف شبکه‌های بات داشته باشد.

۵-۲-۱۱- ویژگی‌های مبتنی بر اندازه جریان / ویژگی‌های مبتنی بر طول زمان

با توجه به این‌که ترافیک تولیدی توسط شبکه‌های بات از الگوهای ارتباطی مشابهی برخوردار بوده و یکنواختی بیشتری نسبت به ترافیک کاربران عادی شبکه دارد، ویژگی‌های مبتنی بر اندازه و طول زمان جریان برای طبقه‌بندی ترافیک و نیز شناسایی ترافیک بدخواه از ترافیک سالم مورد استفاده قرار می‌گیرد.

جدول (۳) نشان‌دهنده فهرستی از ویژگی‌های استخراج‌شده برای عملیات داده‌کاوی می‌باشد. برخی از ویژگی‌ها مانند آدرس آی‌پی مبدأ و مقصد و شماره درگاه فرستنده و گیرنده جریان می‌تواند به‌طور مستقیم از طریق بررسی سرآیند[1] TCP/UDP به دست آید. امّا برخی دیگر از ویژگی‌ها مانند متوسط طول بار مفید بسته، نیاز به پردازش و محاسبات بیشتری دارند.

۵-۳- مدل طبقه‌بندی

روش‌های یادگیری ماشین به‌دلیل استفاده از توابع خودکار برای انجام عملیات پیش‌بینی، می‌توانند ابزار مناسبی به‌جای تولید دستی قوانین و الگوها باشند. به کمک الگوریتم‌های یادگیری ماشین، می‌توان ترافیک شبکه را به دو بخش بدخواه و سالم طبقه‌بندی کرد. برای این منظور ابتدا باید داده‌های ورودی اولیه آموزش‌دیده[2] و سپس داده‌های آزمایشی به کمک الگوریتم یادگیر، آزمایش شود. برای نیل به این هدف، ابتدا بردار ویژگی شامل خصیصه‌های تولیدشده در بخش قبل را به‌عنوان ورودی، به الگوریتم یادگیر داده و نتایج به‌دست می‌آید. بردار ورودی در الگوریتم‌های مختلف آزمایش شده و نتایج ارائه می‌شود. در این مقاله از الگوریتم‌های معروف یادگیری ماشین شامل: Bayesian Network (BNet)، Decision Tree (j48)، Naive Bayesian (NB) و Random Forest (RF) استفاده شده است. این الگوریتم‌ها، الگوریتم‌های بهینه و پرکاربرد در زمینه یادگیری ماشین و داده‌کاوی می‌باشند.

در این مقاله روشی برای شناسایی بلادرنگ و برخط شبکه‌های بات ارائه شده است. بدین ترتیب که با توجه به استخراج مناسب خصیصه‌ها از مجموعه داده آزمایشی، به‌جای گروه‌بندی بسته‌ها در پنجره‌های زمانی ۳۰۰ ثانیه‌ای، تعداد مناسب (و البته بسیار کم) از بسته‌ها در هر جریان گروه‌بندی شده و به کمک روش‌های یادگیری ماشین، طبقه‌بندی می‌شود. بر این اساس و با تغییر در تعداد بسته‌ها در هر جریان می‌توان سامانه‌ای طراحی کرد که در فاصله‌های زمانی کوتاه، جریان‌هایی از بسته‌ها با تعداد کم بسته را مورد وارسی قرار داده و بدخواه‌بودن یا سالم‌بودن ترافیک عبوری را به‌صورت آنی تشخیص دهد.

۵-۴- معرفی مجموعه داده‌های نمونه

یکی از نقاط کلیدی این مقاله که در بسیاری از کارهای مشابه توجه زیادی به آن نشده است، استفاده از مجموعه داده‌های دنیای واقعی برای رسیدن به بهترین کارایی، می‌باشد. در این مقاله از مجموعه داده‌های iscx که در دانشگاه unb کانادا جمع‌آوری شده است، استفاده شد[3] که حدود ۵ میلیون بسته برچسب‌گذاری‌شده را در بر می‌گیرد. این مجموعه داده‌ها هم ترافیک بدخواه و هم ترافیک سالم شبکه را شامل می‌شود و دارای نمونه بسته‌های واقعی از شبکه‌های باتی معروف مانند Zeus، Weasel، RBot و... می‌باشد. فایل مورد نظر با پسوند pcap. است که استاندارد شرکت وایرشارک[4] است.

وایرشارک نرم‌افزار گزارش‌گیری، تحلیل و بررسی اطلاعات در حال تبادل شبکه و یا اینترنت است. این نرم‌افزار با استفاده از توابع آماری، امکان تولید گزارش‌ها و تحلیل نحوه عملکرد و اطلاعات درحال تبادل شبکه را فراهم می‌سازد. در این پروژه ابتدا قبل از بهره‌برداری از بسته‌های موجود در فایل نمونه، از نرم‌افزار وایرشارک برای بررسی این بسته‌ها و درک ویژگی‌های اولیه آن‌ها، استفاده شد.

۶- شرایط و نحوه انجام آزمایش‌ها

در این فصل فرآیند انجام آزمایش بر روی ترافیک نمونه و نحوه واکشی ویژگی‌ها از بسته‌های نمونه، تشریح شده است.

۶-۱- فرآیند تولید ترافیک از فایل نمونه

در این مرحله، نرم‌افزارهای مختلفی مانند wireshark، winpcap و Ethereal برای تحلیل، مقایسه و شبیه‌سازی ترافیک موجود در فایل مذکور آزمایش شد و درنهایت برای عملیات دریافت و فیلترسازی بسته‌ها، از تحلیلگر نحوی مربوط به شرکت مایکروسافت به نام Microsoft Network Monitor در پلتفرم دات نت و زبان برنامه‌نویسی #C استفاده شد. Microsoft Network Monitor ابزاری

1- Header
2- Train
3- https://www.cs.unb.ca/downloads/iscx/
4- Wireshark

جهت دریافت، نمایش و تحلیل داده‌های شبکه و بازگشایی پروتکل‌های آن می‌باشد. این ابزار دارای فیلترینگ کامل و هوشمند بوده و می‌تواند ابزار مفیدی برای مدیریت شبکه باشد. این نرم‌افزار توانایی نمایش فریم به فریم کلیه بسته‌ها را در لایه دو شبکه دارد. تحلیلگر نحوی این نرم‌افزار نیز به‌صورت متن‌باز برای استفاده در محیط VisualStudio در دسترس عموم قرارگرفته است تا ابزاری قدرتمند برای توسعه‌دهندگان نرم‌افزار در جهت افزایش امنیت فراهم سازد. این تحلیلگر نحوی دارای توابع از پیش آماده است تا بسته‌های موجود در فایل‌های با پسوند cap. و pcap. را تحلیل کند. برای انجام این هدف برنامه‌ای در پلتفرم دات نت و زبان برنامه‌نویسی C# نوشته شد تا بسته‌های موجود در مجموعه داده‌های نمونه را با توجه به ویژگی‌های هر بسته مانند آدرس آی‌پی مبدأ و مقصد، شماره درگاه، اندازه بار مفید و... گرفته و برای انجام عملیات بعدی در داخل پایگاه داده ذخیره کند.

جدول (۳): ویژگی‌های استخراجی از جریان بسته‌ها

توضیحات فارسی	توضیحات انگلیسی	ویژگی
آدرس آی‌پی مربوط به ارسال‌کننده جریان	Flow source IP address	SrcIP
آدرس آی‌پی مربوط به دریافت‌کننده جریان	Flow destination IP address	DestIP
آدرس درگاه مربوط به ارسال‌کننده جریان	Flow source port address	SrcPort
آدرس درگاه مربوط به دریافت‌کننده جریان	Flow destination port address	DestPort
پروتکل لایه انتقال (UDP یا TCP)	Transport layer protocol	Protocol
متوسط اندازه بار مفید بسته در پنجره زمانی	Average payload packet length for time interval	APL
واریانس اندازه بار مفید بسته در پنجره زمانی	Variance of payload packet length for time interval	PV
تعداد بسته‌های رد و بدل‌شده در بازه زمانی مشخص	Number of packets exchanged for time interval	PX
تعداد بسته‌های رد و بدل‌شده با بار مفید خالی (بسته‌های با طول بار مفید صفر)	Number of null packets exchanged	NNP
تعداد بسته‌های رد و بدل‌شده با بار مفید کوچک (بسته‌های با طول بار مفید، ۶۳ الی ۴۰۰ بایت)	Number of small packets exchanged	NSP
درصد بسته‌های ردوبدل شده با بار مفید کوچک	Percentage of small packets exchanged	PSP
فاصله زمانی شروع تا پایان جریان (در این تحقیق این بازه مابین ۰ تا ۳۰۰ ثانیه متغیر خواهد بود)	Flow duration	Duration
نرخ رد و بدل‌شدن بسته‌ها در هر ثانیه در پنجره زمانی	Number of packets exchanged per second in time interval T	PPS
اندازه اولین بسته رد و بدل‌شده در جریان	The size of the first packet in the flow	FPS
نرخ مابین تعداد بسته‌های ورودی بر تعداد بسته‌های خروجی در هر جریان	Ratio between the number of incoming packets over the number of outgoing packets	IOPR
تعداد بسته‌های ورودی در هر جریان	number of incoming packets	InPktCount
تعداد بسته‌های خروجی در هر جریان	number of outgoing packets	OutPktCount
کل بایت‌های رد و بدل‌شده در جریان	Total number of bytes	TBT
نسبت تعداد کل بسته‌ها با طول یکسان به تعداد کل بسته‌ها	Total number of packets with the same length over the total number of packets	DPL
متوسط تعداد بیت بر ثانیه	Average bits-per-second	BS

الگوریتم (۱)، شبه‌کد این برنامه را نشان می‌دهد. برای این‌که ویژگی‌های معرفی‌شده در فصل ۶، از بسته‌ها استخراج شوند، ابتدا این بسته‌ها در یک پایگاه داده ذخیره‌شده و به کمک امکانات پرس‌وجوها، در جریانی از بسته‌ها گروه‌بندی می‌شوند. سپس می‌توان به کمک این جریان‌ها، ویژگی‌های موردنظر استخراج شوند. در بخش‌های بعدی این مراحل نیز توضیح داده خواهد شد. در این مرحله بسته‌های خام پس از واکشی از فایل‌های شکسته‌شده، در جدولی در پایگاه داده SQL Server ذخیره شدند.

الگوریتم(۱) : شبه‌کد فرآید تولید ترافیک از فایل نمونه

```
1: procedure Parser(filePath)
2:     ArrayList <Packet> p;
3:     ArrayList <Packet> Parser;
4:     ArrayList <Packet> packetlist;
5:     Pcap pcap1 ← Pcap.openOffline (path, errbuf);
6:     if pcap1 := NULL then errbuf.toString();
7:     end if
8:     packetlist ← new ArrayList < Packet >;
9:     for Packet p : packets do
10:    if packet has IP then
11:    packet.getHeader(ip);
12:    if packet has TCP then
13:    packet.getHeader(tcp);
14:    p ← Packet(tcp.srcport, tcp.dstport, ip.src,
ip.dst,
15:                pkt.hdr.length, pkt.length,
16:                pkt.timestamp);
17:            packetlist.add(p);
18:        else
19:            packet.getHeader(udp);
20:            p ← Packet(udp.srcport, udp.dstport,
ip.src, ip.dst,
21:                pkt.hdr.length, pkt.length,
22:                pkt.timestamp);
23:            packetlist.add(p);
24:        end if
25:    end if
26:  end for
27:  pcap1.loop;
28: return packetlist;
29: end procedure
```

۶-۲- فرآیند گروه‌بندی بسته‌ها در جریان‌ها[1]

همان‌طورکه در تعریف جریان آمده است، جریان عبارت از مجموعه‌ای از بسته‌های رد و بدل‌شده مابین دو آدرس آی‌پی منحصربه‌فرد در پنجره زمانی T (یا شامل تعداد محدود و مشخص از بسته‌ها) و با استفاده از یک جفت درگاه و پروتکل لایه انتقال UDP یا TCP است. نحوه گروه‌بندی بسته‌ها بدون درنظرگرفتن پنجره زمانی، ساده امّا غیرکاربردی می‌باشد زیرا ممکن است یک جریان مابین ۱ ثانیه تا چندین روز به طول انجامد. پس قبل از هر چیز باید ترافیک بسته‌ها در پنجره زمانی 300، گروه‌بندی شوند. برای این‌کار از یک پرس‌وجوی Sql ای استفاده شده است. در این پرس‌وجو، بسته‌هایی که دارای جفت آدرس آی‌پی و درگاه مبدأ / مقصد یکسان هستند در یک گروه به صورتی گروه‌بندی می‌شوند که فاصله زمانی شروع تا پایان کمتر از ۳۰۰ ثانیه باشد. به‌عبارت دیگر، پنجره زمانی ۳۰۰ ثانیه انتخاب شد.

۶-۳- فرآیند استخراج خصیصه‌ها

پس از این‌که جریان بسته‌ها در پنجره زمانی ۳۰۰ ثانیه گروه‌بندی شد، نوبت به واکشی خصیصه‌های انتخابی از لیست جریان‌ها می‌رسد. این خصیصه‌ها براساس ویژگی‌های ارائه‌شده در بخش ۶-۲ تولید می‌شوند. حال بعد از استخراج ویژگی‌های جریان‌ها، حدود ۲۵۰۰۰ داده ورودی برای انجام عملیات داده‌کاوی تهیه شد و در مرحله بعدی به کمک این داده‌ها عملیات یادگیری انجام خواهد پذیرفت.

۶-۴- فرآیند ارزیابی خصیصه‌ها

در این مرحله از یک ابزار قدرتمند در زمینه داده‌کاوی به نام وکا[2] استفاده شده است. ابزار وکا نرم‌افزاری است متن‌باز که توسط دانشگاه Waikato در کشور نیوزلند ارائه شده است [۲۹]. از ویژگی‌های بارز این نرم‌افزار می‌توان به پوشش تقریباً کامل الگوریتم‌های داده‌کاوی در آن اشاره کرد. این نرم‌افزار همان‌طورکه گفته شد به صورت متن باز و با زبان جاوا از سال ۱۹۹۹ به جامعه علمی در دنیا عرضه شده است که می‌توان از این ویژگی جهت توسعه و یا حتی سفارشی‌سازی آن برای کاربردهای خاص استفاده نمود. این ابزار برخلاف دیگر نرم‌افزارهایی که در حوزه داده‌کاوی فعالیت دارند -مانند متلب[3]- رایگان می‌باشد. وکا دارای محیطی کاملاً گرافیکی و کاربرپسند بوده و از لحاظ کاربری، نرم‌افزاری بسیار ساده و روان می‌باشد [۳۰]. به‌طور خلاصه، وکا مجموعه تقریباً کاملی از جدیدترین الگوریتم های یادگیری ماشین و پیش‌پردازش داده را برای داده‌کاوی فراهم کرده است. امکانات این ابزار در تمام گام های داده‌کاوی از قبیل آماده‌سازی داده‌های ورودی، اعمال روش‌های یادگیری و ارزیابی نتایج، بصری‌سازی داده‌ها و بصری‌سازی نتایج خروجی بدون نیاز به کدنویسی دردسترس می‌باشد.

در این مرحله خصیصه‌های به‌دست‌آمده در مرحله قبل در یک فایل با پسوند CSV. ذخیره‌شده و به‌عنوان ورودی ابزار وکا مورد استفاده قرار می‌گیرد. مدل طبقه‌بندی در وکا به کمک اعتبارسنجی متقابل[4] و به روش k-fold که در آن k=10 می‌باشد انجام می‌گیرد. در این حالت، مجموعه ورودی به ۱۰ زیرمجموعه تقسیم و برای هر زیرمجموعه، ۹ قسمت از داده‌ها به‌عنوان یادگیر و قسمت باقی‌مانده به‌عنوان داده آزمایش استفاده می‌شود. این مراحل ۱۰ بار تکرار شده و بهترین کارایی حاصل می‌شود و رفتار شبکه‌های باتی ناشناخته به کمک این روش به دست می‌آید. البته به‌علت این‌که خصیصه‌هایی مانند آدرس آی‌پی و شماره درگاه مربوط به جریان‌های ورودی و خروجی، نمی‌تواند خصیصه‌های مناسبی برای داده‌کاوی باشد و فقط

1- Flow Creation

2- Weka
3- Matlab
4- Cross Validation

برای انجام عملیات محاسباتی مورد استفاده قرار گرفتند، از فایل ورودی حذف و فقط تعداد ۱۶ خصیصه به‌عنوان بردار ورودی نرم‌افزار وکا انتخاب شد. نکته بسیار مهم دیگر مشکلی است که در این مرحله رخ داد و پس از بارگذاری فایل ورودی در نرم‌افزار وکا و قبل از شروع عملیات طبقه‌بندی، نرم‌افزار وکا با خطای کمبود حافظه مواجه گردید. برای حل این مشکل، فایل RunWeka.ini موجود در مسیر نصب وکا را در محیط نوت‌پد باز کرده و مقدار maxheap افزایش‌یافته و به ۱۲۸۰۰ تغییر می‌یابد و به این ترتیب مشکل حل خواهد شد.

قبل از انجام عملیات داده‌کاوی در وکا، معیارهای ارزیابی که توسط الگوریتم‌های یادگیری ماشین ایجاد و مقایسه خواهد شد، معرفی می‌شود. ابزار وکا، معیارهایی که در ادامه معرفی می‌شوند را به‌صورت خودکار بعد از تولید مدل، به‌عنوان خروجی نمایش می‌دهد.

$$\text{Precision (PRC)} = \frac{\text{TP}}{\text{TP+FP}} \quad (۱)$$

$$\text{Recall (RCL)} = \frac{\text{TP}}{\text{TP+FN}} \quad (۲)$$

$$\text{F} - \text{measure (FM)} = 2 * \frac{\text{RCL} * \text{PRC}}{\text{RCL} + \text{PRC}} \quad (۳)$$

که در آن، TP، FP، FN به ترتیب مثبت واقعی[1]، مثبت کاذب[2]، منفی کاذب[3] می‌باشند. دو پارامتر Precision و Recall از اهمیت یکسانی برخوردار نیستند. Precision نشان‌دهنده این است که چند درصد از 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. پارامتر Recall از پارامتر Precision مهم‌تر است زیرا: اگر Recall کم باشد، یعنی الگوریتم توانسته است درصد کمی از ترافیک‌های بات را شناسایی کند. به‌عبارت‌دیگر، 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 تعداد زیادی ترافیک نرمال را به‌عنوان ترافیک بات شناسایی کرده است. 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.

1- True Positive
2- False Positive
3- False Negative

۷- نتایج ارزیابی و مقایسه نتایج

بعد از بارگذاری فایل csv. به‌عنوان ورودی وکا، نوبت به طبقه‌بندی و اعمال الگوریتم‌های یادگیر می‌رسد که از زیرمنوی Classify قابل دسترس می‌باشند. جدول (۴) نشان‌دهنده درصد معیارهای ارزیابی مورد نظر برای ترافیک بدخواه و ترافیک سالم که از ۴ الگوریتم یادگیری انتخابی به‌دست آمده است، می‌باشد. این نتایج از طبقه‌بندی جریان‌های دارای پنجره زمانی ۳۰۰ ثانیه و به‌صورت 10- Fold محاسبه شده است. این جدول نشان می‌دهد که الگوریتم RandomForest با دقت بیشتری نسبت به سه الگوریتم دیگر، سالم‌بودن یا مخرب‌بودن بسته‌ها را تشخیص می‌دهد.

قبل از ادامه این بخش، به بررسی مناسب‌ترین پنجره زمانی T برای گروه‌بندی بسته‌ها در جریان‌ها می‌پردازیم. با گروه‌بندی بسته‌ها در پنجره‌های زمانی مختلف و مقایسه TP و FP مربوط به ترافیک بدخواه، این نتیجه حاصل شد که پنجره زمانی حدود ۳۰۰ ثانیه بهترین کارایی را حاصل می‌کند. اشکال (۴-۳) نشان‌دهنده تأثیر اندازه پنجره زمانی بر نرخ مثبت صحیح و مثبت کاذب الگوریتم RandomForest می‌باشد. شکل (۵)، مقایسه الگوریتم‌های مختلف برای طبقه‌بندی جریان‌ها را نشان می‌دهد. این شکل نشان می‌دهد که الگوریتم Random Forest دقت بیشتر داشته و معیارهای ارزیابی دارای کمترین تفاوت نسبت به همدیگر می‌باشند. همچنین شکل (۶)، نشان‌دهنده مقایسه زمان‌های تولید مدل در هر الگوریتم می‌باشد.

در این مقاله روشی برای شناسایی بلادرنگ و برخط شبکه‌های بات ارائه شده است. بدین ترتیب که تعداد مناسب (و البته بسیار کم) از بسته‌ها در هر جریان گروه‌بندی شده و به کمک روش‌های یادگیری ماشین، طبقه‌بندی می‌شود. می‌توان نتیجه گرفت که به کمک این روش، بتوان سیستمی طراحی کرد که در فاصله‌های زمانی کوتاه، جریان‌هایی از بسته‌ها با تعداد کم بسته را مورد وارسی قرار داده و بدخواه‌بودن یا سالم‌بودن ترافیک عبوری را به‌صورت آنی تشخیص داد. برای این منظور، الگوریتم گروه‌بندی بسته‌ها تغییر کرده و به‌جای گروه‌بندی بسته‌ها در پنجره‌های زمانی ۳۰۰ ثانیه‌ای، در هر جریان تعداد محدود بسته قرار می‌گیرد. این تعداد به ترتیب ۱۰، ۵۰، ۱۰۰ و ۱۰۰۰ بسته در هر جریان را شامل می‌شود. سپس نتایج گروه‌بندی بسته‌ها به‌عنوان ورودی به الگوریتم تولید خصیصه‌ها ارسال و نتایج به‌صورت فایل csv. به نرم‌افزار وکا داده

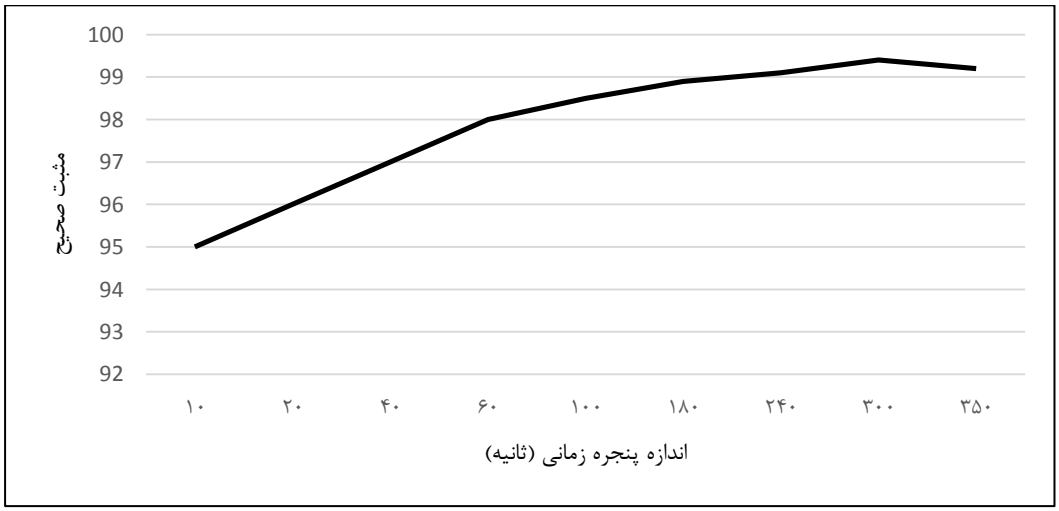
می‌شود. با توجه به این‌که الگوریتم Random Forest از نتیجه قابل‌قبول‌تر و با انحراف کمتر برخوردار است، نتایج این گروه‌بندی توسط این الگوریتم طبقه‌بندی شده و نتیجه در جدول (۵) نمایش داده شده است. اطلاعات به‌دست‌آمده نشان می‌دهد که با گروه‌بندی تعداد کم بسته‌ها در یک جریان (بین ۵۰ تا ۱۰۰ بسته) و با درنظرگرفتن این‌که شبکه‌های باتیی مانند weasel در ارتباطات خود بسیار کند عمل کرده و تعداد بسته‌های کمی بین مدیر بات و بات‌ها ردوبدل می‌شود، بتوان سیستمی طراحی کرد که ترافیک عبوری را به‌صورت بلادرنگ و با نرخ تشخیص بالا در دو گروه بدخواه و سالم طبقه‌بندی کند.

پروژه‌های انجام‌شده توسط Masud[۲۵]، Saad[۱۳]، Bilge[۲۶] و Zhao[۱۲] مشابه با مقاله حاضر می‌باشند. با مطالعه نتایج به‌دست‌آمده از این مقالات و مقایسه با پژوهش حاضر، نتایج زیر حاصل می‌شود:

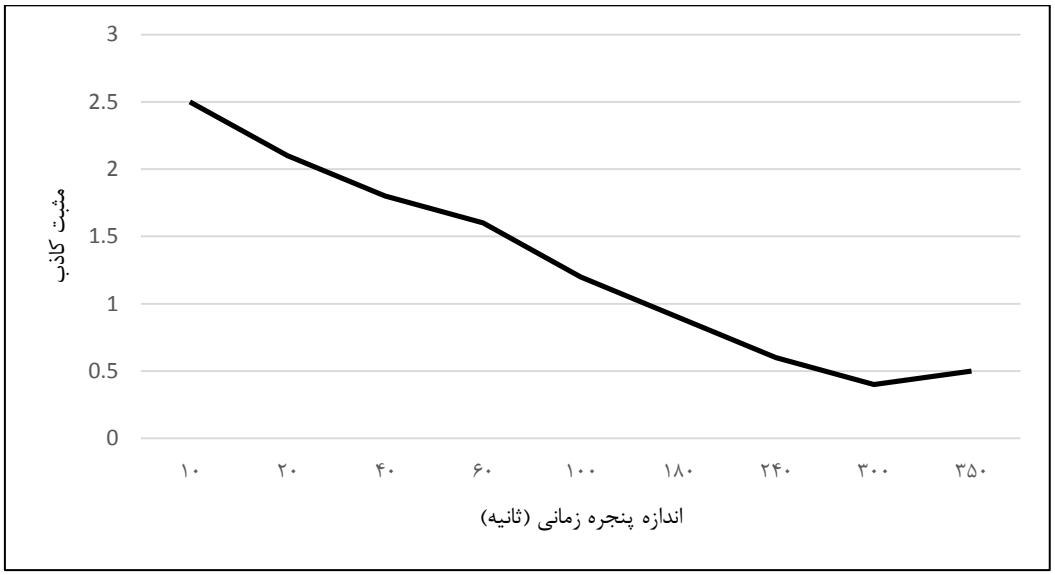
- با مطالعه ویژگی‌های شبکه‌های باتی مشهور مانند Weasel و ...، خصیصه‌هایی برای شناسایی الگوهای مشابه شبکه‌های بات تولید و استخراج شد که منجر به دست‌یابی به بالاترین دقت و کمترین مثبت کاذب نسبت به نتایج پژوهش‌های مذکور شد. به‌عنوان مثال، Saad و همکاران، به کمک روش SVM به نرخ تشخیص حدود ۹۷ ٪ و Bilge و همکاران به کمک روش Random Forest به نرخ تشخیص ۷۰ ٪ دست پیدا کرده‌اند. این درحالی است که پس از تحلیل جریان بسته‌ها و استخراج و تولید خصیصه‌های وابسته به جریان، این خصیصه‌ها در جهت شناسایی الگوهای رفتاری مشابه، داده‌کاوی شده و درنهایت الگوی بسته‌های سالم از بسته‌های مخرب متمایز شد و نرخ تشخیص در حدود ۹۹ ٪ به دست آمد. این افزایش دقت به علت تولید خصیصه‌های مناسب و در نتیجه جداسازی رفتار شبکه‌های بات از جریان عادی شبکه می‌باشد.

جدول (۴): درصد معیارهای ارزیابی برای تشخیص شبکه بات براساس خصیصه‌های استخراجی (T=300s)

الگوریتم یادگیری	Bayesian Network			Decision Tree			Naive Bayesian			Random Forest		
پارامتر ارزیابی	PRC	RCL	FM	PRC	RCL	FM	PRC	RCL	FM	PRC	RCL	FM
ترافیک بدخواه	۹۶/۴	۸۹/۵	۹۲/۸	۹۷/۷	۹۷/۸	۹۷/۸	۷۳/۲	۱۳/۵	۲۲/۷	۹۷/۷	۹۹/۳	۹۸
ترافیک سالم	۸۲/۸	۹۳/۸	۸۷/۹	۹۵/۹	۹۵/۸	۹۵/۸	۳۶/۱	۹۰/۸	۵۱/۷	۹۶/۸	۹۵/۵	۹۶/۳

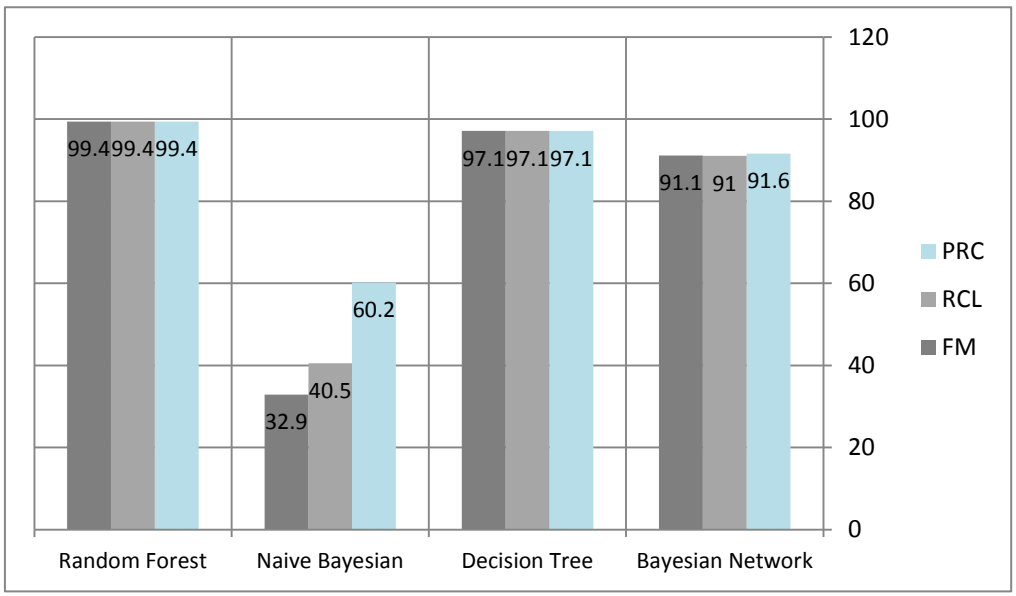


شکل (۳): تأثیر اندازه پنجره زمانی بر نرخ مثبت صحیح ترافیک بدخواه

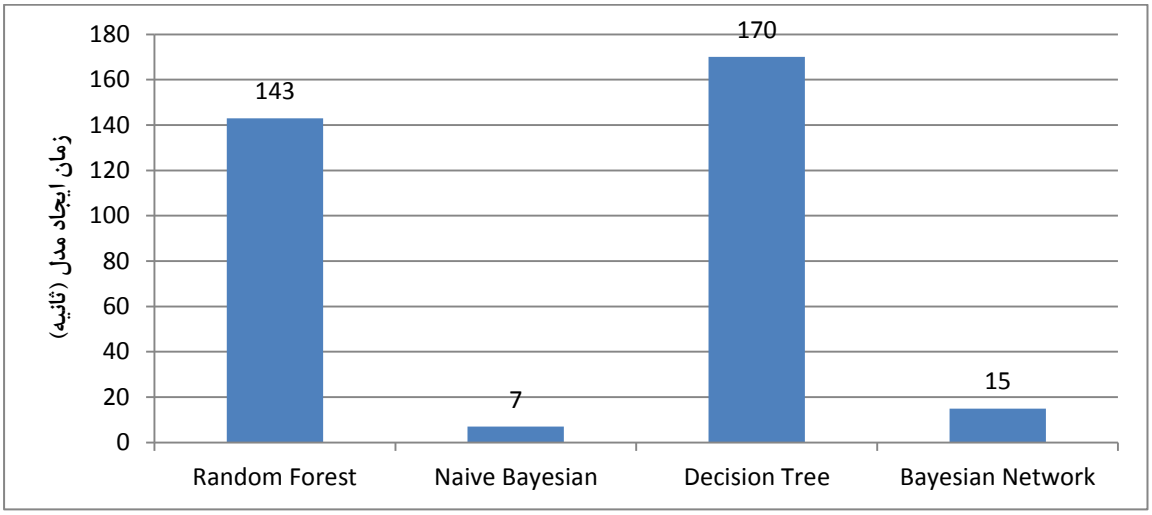


شکل (۴): تأثیر اندازه پنجره زمانی بر نرخ مثبت کاذب ترافیک بدخواه

- الگوریتم پیشنهادی در مقاله برای تولید ترافیک از فایل نمونه با توجه به استفاده از یک تحلیل‌گر نحوی قابل اطمینان و قدرتمند مانند Microsoft Network Monitor Parser از سرعت بسیاری نسبت به کارهای مشابه مانند روش‌های تحت وب و فیتون دارد. با توجه به این‌که مجموعه داده‌های مورد استفاده در مقالات مذکور تا حدودی ازلحاظ ساختاری شبیه به مجموعه بسته‌های استفاده‌شده در این مقاله می‌باشد، با استخراج خصیصه‌های مناسب، زمان تولید مدل نسبت به کارهای مشابه در وضعیت مناسب‌تری قرار دارد.
- روش پیشنهادی برای کشف بلادرنگ شبکه‌های بات در این مقاله، در هیچ‌یک از کارهای مشابه مدنظر قرار داده نشده است و از این لحاظ می‌توان این پژوهش را پیش‌زمینه‌ای برای تولید یک سیستم برخط برای شناسایی بلادرنگ شبکه‌های بات دانست.



شکل (۵): مقایسه الگوریتم‌های مختلف برای طبقه‌بندی جریان‌ها در پنجره زمانی ۳۰۰ ثانیه



شکل (۶): مقایسه زمان‌های تولید مدل در هر الگوریتم

جدول (۵): مقایسه معیارهای ارزیابی در گروه‌بندی تعداد مختلف بسته در هر جریان

	۱۰ بسته در هر جریان	۵۰ بسته در هر جریان	۱۰۰ بسته در هر جریان	۱۰۰۰ بسته در هر جریان
Precision	۹۱/۵	۹۵/۶	۹۷/۸	۹۷/۸
Recall	۹۱/۱	۹۵/۶	۹۷/۸	۹۸
F-measure	۹۱/۱	۹۵/۶	۹۷/۸	۹۷/۹

۸- نتیجه‌گیری

در این مقاله سعی شده است تا یک روندکاری مناسب در جهت تشخیص کامپیوترهای آلوده به شبکه بات اتخاذ شود تا به کمک روش‌های یادگیری ماشین و نیز با استفاده از داده‌های برچسب‌گذاری‌شده، خصیصه‌های تولیدی از جریان بسته‌ها به کمک ابزار متن‌باز وکا طبقه‌بندی شود. منطق اصلی روش استفاده‌شده در این پژوهش بر این پایه استوار است که شبکه‌های بات الگوهای ترافیکی قابل تشخیصی از خود به‌جای می‌گذارند که به کمک روش‌های یادگیری ماشین قابل شناسایی بوده و می‌توان ترافیک تولیدی توسط شبکه‌های بات را از ترافیک عادی شبکه جدا کرد. در این مسیر نکات زیر را می‌توان به‌عنوان عامل اصلی موفقیت این مقاله قلمداد کرد:

استفاده از "مجموعه داده‌های نمونه جهان واقعی[1]" از بسته‌های شبکه شامل شبکه‌های باتی معروف و نیز بسته‌های سالم برچسب‌گذاری‌شده.

گروه‌بندی مناسب بسته‌های نمونه در جریان‌ها با توجه به پنجره زمانی متناسب.

مطالعه ویژگی‌های شبکه‌های باتی مشهور مانند Weasel و ...، در جهت تولید خصیصه‌هایی برای شناسایی الگوهای مشابه شبکه‌های بات.

دست‌یابی به بالاترین دقت و کمترین مثبت کاذب نسبت به نتایج پژوهش‌های مذکور به‌دلیل استخراج ویژگی‌های مناسب از جریان‌های ترافیکی برچسب‌دار.

انتخاب الگوریتم‌های یادگیری ماشین با کارایی مناسب در ابزار وکا.

ارائه روشی برای کشف بلادرنگ شبکه‌های بات در جهت تولید یک سیستم برخط برای شناسایی بلادرنگ شبکه‌های بات.

در این پروژه با مطالعه مقالات مختلف درزمینه تشخیص شبکه‌های بات، یک راه‌حل عملی برای شناسایی میزبان‌های آلوده ارائه‌شده و از کلی‌گویی پرهیز شد. در اکثر کارهای مشابه، فرآیند تولید ترافیک به‌صورت نظری توضیح داده شده است و ترافیک تولیدی نیز کمتر شامل شبکه‌های بات مشهور بوده و تعداد بسته‌های موجود برای عملیات داده‌کاوی نیز کافی نیست؛ درحالی‌که در این پروژه برای بالابردن کارایی، از مجموعه داده‌های کامل استفاده شده است. همچنین، با آزمایش تعداد مختلف بسته در هر جریان سعی شد تا بهترین کارایی حاصل شده و پیش‌زمینه برای کارهای آینده در جهت تولید یک سیستم شناسایی برخط شبکه‌های بات فراهم شود.

1- Real World Sample Dataset

۹- پژوهش‌های آینده

در پژوهش‌های مختلف، روش‌های تشخیص شبکه بات، ساختار خود را با جزئیات تشریح می کنند. بنابراین، این خطر همواره وجود دارد که فن‌های گریز مختلفی برای آن‌ها، از طرف مهاجمین ارائه شود. در سوی دیگر، شبکه‌های باتی نسل آینده به احتمال بسیار زیاد از فن‌های مختلفی برای گریز از تشخیص بهره می‌گیرند. از این‌رو، مقاوم‌سازی هرچه بیشتر روش‌های پیشنهادی در برابر فن‌های مختلف گریز یکی از مهم‌ترین چالش‌های موجود برای تشخیص شبکه‌های بات خواهد بود. هرچند شبکه‌های بات و بات‌ها روزبه‌روز در حال پیشرفت بوده و پیچیده‌تر می‌شوند، امّا بررسی‌های انجام‌شده در رابطه با تشخیص شبکه‌های بات هنوز در مسیر ابتدایی قرار دارد و می‌توان با ترکیب چند روش به نتایج بهتری دست یافت.

یکی دیگر از مباحثی که جای مطالعه و تحقیق بسیاری دارد، مطالعه بر روی روش‌های تشخیص برخط شبکه‌های بات می‌باشد. اکثر روش‌های ارائه‌شده برای شناسایی شبکه‌های بات، به‌صورت غیربرخط و برمبنای پروتکل کانال فرماندهی و کنترل بنا نهاده شده و بسیار کند می‌باشند. این روش‌ها اکثراً براساس اطلاعات جمع‌آوری‌شده درگذشته عمل می‌کنند و همیشه یک‌قدم عقب‌تر از مهاجم قرار دارند. لذا ارائه یک روش برخط برای تشخیص شبکه‌های بات می‌تواند گام بسیار بزرگی در جهت مقابله با حملات سایبری باشد.

در نهایت، با افزایش روزافزون استفاده از تبلت‌ها و گوشی‌های هوشمند، شبکه‌های باتی مختلف این حوزه نیز روزبه‌روز در حال گسترش می‌باشد و به‌خصوص شبکه‌های اجتماعی مانند تلگرام و فیس‌بوک، بیش از گذشته در معرض حملات و سوءاستفاده توسط شبکه‌های بات قرار دارند. لذا لزوم مطالعه و تحقیق بر روی انواع روش‌های کشف و شناسایی شبکه‌های بات در حوزه گوشی‌های هوشمند و سیستم‌عامل‌های اندروید و IOS بیش از پیش حس می‌شود.

۱۰- مراجع

[1] Available https://en.wikipedia.org/wiki/Botnet

[2] J. Liu, et al., "Botnet: Classification, Attacks, Detection, Tracing.and Preventive Measures," Journal on Wireless Communications and Networking, 2009.

[3] L. Heng-Feng and H. Ru-Xin, "A Survey of botnet Detection," Computers & Security, 2010.

[4] M. Feily, A. Shahrestani, and S. Ramadass, "A Survey of Botnet and Botnet Detection," 09 Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies, 2009.

[5] M. Feily, A. Shahrestani, and S. Ramadass, "A Survey of Botnet and Botnet Detection, IEEE 3rd Int. Conf. On Emerging Security Information, Systems and Technologies, 2009.

[6] J. Park, "Acquiring Digital Evidence from Botnet Attacks: Procedures and Methods," M. Sc Thesis, 2011.

[7] F. C. Freiling, T. Holz, and G. Wicherski, "Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks," Springer Berlin Heidelberg, 2005.

[8] C. C. Zou and R. Cunningham, "Honeypot-aware advanced botnet construction and maintenance," IEEE Int. Conf. on Dependable Systems and Networks, 2006.

[9] J. R. Binkley and S. Singh, "An Algorithm for Anomalybased botnet detection. Proc. USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop, 2006.

[10] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and M. Ronaldo Salles, "Botnets: A survey, Computer Networks," The International Journal of Computer and Telecommunications Networking, 2013.

[11] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An Overview of IP Flow-based Intrusion Detection," IEEE Communications Surveys & Tutorial, 2010.

[12] D. Zhao, I. Traore, A. Ghorbani, B. Sayed, S. Saad, and W. Lu, "Peer to Peer Botnet Detection Based on Flow Intervals, Part of the IFIP Advances in Information and Communication Technology Book Series, 2012.

[13] S. Saad, I. Traore, A. Ghorbani, B. Sayed, 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 NinthAnnual International Conference on, july 2011.

[14] Snort IDS [Online] Retrieved on January 2013 from http://www.snort.org

[15] W. Wei, F. Binxing, et al., "A Novel Approach to Detect IRC-Based Botnet," International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC), 2009.

[16] R. Konrad, S. Guido, et al., "Botzilla: Detecting the "Phoning Home" Of Malicious Software," Proceedings of the Symposium on Applied Computing, Sierre, Switzerland, ACM, 2010.

[17] E. Stinson and J. C. Mitchell, "Characterizing bots remote control behavior," In: Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2007.

[18] G. Gu, P. Porras, V. Yegneswaran, et al., "Bothunter: Detecting Malware Infection Through IDS-driven Dialog Correlation," Proc. 16th USENIX Security Symp., pp. 167-182, 2007.

[19] G. Gu, J. Zhang, and W. Lee, "A BotSniffer: detecting Botnet Command and Control Channels in Network Traffic," Proc. 15th Annual Network and Distributed System Security Symp., pp. 2-19, 2008a.

[20] G. Gu, R. Perdisci, J. Zhang, et al., "BotMiner: Clustering Analysis of Network Traffic for Protocol and Structure-Independent Botnet Detection," USENIX Security Symp., pp. 139-154, 2008b.

[21] W. T. Strayer, R. Walsh, C. Livadas, et al., "Detecting botnets with tight command and control," Proc. 31st IEEE Conf. on Local Computer Networks, pp. 195-202, 2006.

[22] A. G. Tokhtabayev and V. A. Skormin, "Non-stationary Markov models and anomaly propagation analysis in IDS," IEEE 3rd Int. Symp. on Information Assurance and Security, pp. 203-208, 2007. [doi:10.1109/IAS.2007.72]

[23] H. Choi, H. Lee, and H. Kim, "BotGAD: detecting botnets by Capturing Group Activities in Network Traffic," Proc. 4th Int. ICST Conf. on Communication System Software and Middleware, pp. 1-8, 2009. [doi:10.1145/1621890.1621893]

[24] A. Shahrestani, M. Feily, R. Ahmad, et al., "Architecture for applying data mining and visualization on network flow for botnet traffic detection," IEEE Int. Conf. on Computer Technology and Development, pp. 33-37, 2009.

[25] M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K. Hamlen, "Flow-based Identification of Botnet Traffic by Mining Multiple log Files," In Distributed Framework and Applications, DFmA 2008, First International Conference on, pp. 200 –206, Oct. 2008.

[26] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis," In Proceedings of the 28th Annual Computer Security Applications Conference, ser. ACSAC '12, New York, NY, USA: ACM, pp. 129–138, 2012.

[27] R. Azmi, M. Gholinezhad, and M. Saberi, "Botnet Detection for Peer to Peer Networks," Journal of Electronical & Cyber Defence, vol. 3, n o. 4, Serial No.12, 2016. (In Persian)

[28] M. Fathian, M. Abdollahi Azgomi, and H. Dehghani, "Modeling Browsing Behavior Analysis for Malicious Robot Detection in Distributed Denial of Service Attacks," Journal of Electronical & Cyber Defence, vol. 4, no. 2, 2016, Serial no. 14 (In Persian)

[29] I. H. Witten and E. Frank, "Data Mining, 3th ed, SanFrancisco: Morgan Kafman, pp. 403-468, 2011.

[30] J. Han and M. Kamber, "Data Mining Concepts and Techniques," SanFrancisco :Morgan Kafman, pp. 18-21, 2011.

Botnet Detection with Flow Behavior Analysis Approach

S. Parsa*, H. Mortazi

*Iran University of Science and Technology

(Received: 25/07/2016, Accepted: 13/02/2017)

ABSTRACT

"Botnet"is a network of infected computers connected to the Internet that is under management of the command and control server and is used for denial of service attacks, for sending spams and other malicious operations. The size of a botnet depends on the complexity and number of computers employed. Users usually do not know that their systems are remotely controled and abused. Botnets are attractive for cyber criminals, because they are capable of being reset for various offenses, moved to new hosting services, or they are reprogrammed in response to new developments in security. Despite the specific characteristics of each botnet, bots in a botnet exhibit homogeneous behaviors and this can be the starting point for identifying a botnet within a network. Discoverable behavior of bots in a botnet can lead to production of features and attributes. Analyzing of these features, we can classify traffic to malicious and non-malicious traffic.This approach uses network flow analysis and machine learning methods to detect peer to peer botnets. Furthermore,this approach is flow-based and analyzes features extracted from flows based on the behavior of well-known botnets such as Weasel, etc and determines that the new traffic is an attack or not.

Keywords: Botnet, Bot, Peer to Peer Botnet, Network Flow Analysis, Machine Learning.

*Corresponding Author Email: parsa@iust.ac.ir