مجله علمی- پژوهشی "پدافند الکترونیکی و سایبری"
سال ششم، شماره ۳، پاییز ۱۳۹۷، ص ۱۵۱-۱۴۱

تشخیص حملات سایبری پیشرفته با استفاده از مدل‌سازی رفتاری مبتنی بر پردازش زبان طبیعی

کوروش داداش‌تبار احمدی[۱*]، مرجان خیرخواه[۲]، علی جبار رشیدی[۳]
۱- استادیار، ۲- دانشجوی کارشناسی ارشد، ۳- دانشیار، دانشگاه صنعتی مالک اشتر
(دریافت: ۹۶/۰۹/۱۰، پذیرش: ۹۶/۱۱/۱۴)

چکیده

رشته حملات پیچیده و ماندگار نفوذ به شبکه از مراحل نامحسوس و مخفی متعددی تشکیل شده‌اند. یکی از دلایل ناکارآمدی سامانه‌های تشخیص نفوذ در برابر این حملات، استفاده از سازوکار دفاعی مبتنی بر آنالیز ترافیک شبکه‌ای سطح پایین است که در آن به روابط پنهان بین هشدارها توجه نمی‌شود. فرض ما این است که اطلاعات ساختاری پنهان در داده‌های ترافیکی وجود دارند و ما می‌خواهیم در ترافیک شبکه‌ای قواعدی مانند قواعد زبان تعریف کنیم و آن‌را برای توصیف الگوهای فعالیت‌های شبکه‌ای بدخواهانه به‌کار بگیریم. به این وسیله می‌توانیم مسئله کشف الگوهای سوء استفاده و ناهنجاری را همانند مسئله یادگیری ساختارهای نحوی و قطعات مفهومی "زبان شبکه" حل کنیم. در این مقاله برای مدل‌سازی در مرحله تولید دنباله‌ها برای اولین بار در حوزه سایبری از یک خوشه‌بندی جدید به‌عنوان خوشه‌بندی MD_DBSCAN که یکی از انواع بهبودیافته خوشه‌بندی DBSCAN است، استفاده شده است. علاوه‌بر این، از یک الگوریتم حریصانه با الهام از القاء گرامر در پردازش زبان طبیعی استفاده شده تا با ادغام فعالیت‌های سطح پایین بتوانیم فعالیت‌های سطح بالا را کشف کنیم و روابط بین فعالیت‌های سطوح مختلف را تعریف کنیم. در بخشی از الگوریتم پیشنهادی برای کشف فعالیت‌های سطح بالا، برای اولین بار معیار شباهت ویرایش در خوشه‌بندی سلسله مراتبی به معیارهای موجود در الگوریتم پایه اضافه شده است. نتایج نشان می‌دهد دقت تشخیص در فعالیت‌های سطح بالا نسبت به فعالیت‌های سطح پایین با توجه به نمودار ROC حدود ۳۰ ٪ بیشتر است. همچنین، با تنظیم بهترین حد آستانه در الگوریتم تشخیص حملات، با درنظرگرفتن معیار F1 ، برای لغات سطوح یک تا سه به ترتیب به نتایج ۷۲/۳ و ۹۶/۲ و ۹۶/۴ در پنجره پیش‌بینی با اندازه سه رسیده‌ایم که به‌طورکلی حدود ۰/۲ نسبت به الگوریتم پایه بهبود نشان می‌دهد.

واژگان کلیدی: رفتار، حملات ماندگار پیشرفته، حملات سایبری، ادغام داده، پردازش زبان طبیعی

۱- مقدمه

تشخیص فعالیت‌های سطح بالای رفتار از طریق ادغام داده‌های حسگرهای مختلف، موضوعی است که کاربردهای متنوعی دارد. در این نوع سامانه‌های چندحسگری تشخیص دقیق فعالیت‌ها امکان‌پذیر شده است. در جاهایی‌که داده‌های ورودی معمولاً دنباله‌ای تک‌بعدی نیستند، ادغام حسگرهای چندگانه می‌تواند با پیش‌پردازش صورت‌گرفته بر روی این داده‌های چندبعدی تشخیص فعالیت‌ها را بهبود بخشد. طبق مطالعات انجام‌شده چالشی که در به‌کارگیری این سامانه‌ها با آن مواجه هستیم، حسگرآمیزی ناهمگن[۱] است.

چالش بعدی در این حوزه، نیاز به تشخیص فعالیت‌های سطح بالا بدون نظارت است؛ زیرا روش‌های نظارت‌شده از برچسب‌های از پیش تعیین‌شده استفاده می‌کنند که هم واقع‌گرایانه نیست و هم محدود به فعالیت‌های از پیش تعیین‌شده است. درحالی‌که معمولاً تشخیص فعالیت‌های سطح بالا به‌صورت عام‌تر، سودمندتر است.

تاکنون از روش‌های متعددی برای شناسایی فعالیت‌های سطح بالا مانند ادغام اطلاعات سطح بالا و پردازش زبان طبیعی استفاده شده است. روش به‌کاررفته در این مقاله برای استخراج فعالیت‌های سطح بالا استفاده از مفاهیم NLP و القاء گرامر متناظر با زبان شبکه می‌باشد.

پردازش زبان‌های طبیعی یکی از زیرشاخه‌های با اهمیت در حوزه علوم رایانه، هوش مصنوعی و نیز دانش زبان‌شناسی محاسباتی است که به تعامل بین رایانه و زبان‌های (طبیعی) انسانی می‌پردازد؛ پردازش زبان‌های طبیعی عبارت است از استفاده از رایانه برای پردازش زبان گفتاری و زبان نوشتاری. بدین معنی که رایانه‌ها را قادر سازیم که گفتار یا نوشتار

* رایانامه نویسنده مسئول: Dadashtabar@mut.ac.ir
1- Heterogeneous sensor fusion

تولیدشده در قالب و ساختار یک زبان طبیعی را تحلیل و درک نموده یا آن را تولید نمایند. از کاربردهای NLP می‌توان به فهم زبان طبیعی، برچسب‌گذاری اجزای کلام، رفع ابهام و استخراج روابط[1] اشاره کرد. در روش پیشنهادی، علاوه‌بر استخراج روابط میان دنباله‌های دریافتی از شبکه، از

روش عمقی که در رفع ابهام در زبان طبیعی وجود دارد الهام می‌گیریم. زیرا روش‌های عمقی، دانش عمیق‌تری از کلمه را متصور می‌شوند. رفع ابهام از معنای یک کلمه برمی‌گردد به این‌که آن کلمه در چه جمله‌ای به‌کاررفته و با چه کلماتی هم‌نشین گردیده است، چراکه یک کلمه در یک جمله یک معنی می‌دهد، درحالی‌که همان کلمه در جمله دیگر معنای دیگری به خود می‌گیرد.

در نهایت، هدف ما کشف الگوهای رفتاری سطح بالای اصلی در ترافیک شبکه است که احتمال دارد نشانه‌های اولیه حملات سایبری باشند. سامانه ما قادر به کشف حملات سایبری در مراحل اولیه آن است به‌طوری‌که برخی از اقدامات دفاعی می‌توانند قبل از به خطر افتادن واقعی سامانه انجام شوند.

بعد از این‌که در بخش اول مزایای به‌کارگیری روش‌های پردازش زبان طبیعی و استخراج رفتارهای سطح بالا را بیان شد، در بخش دوم، کارهای مرتبط گذشته را با تمرکز بر روش‌های استفاده‌شده برای تشخیص حملات سایبری چندمرحله‌ای مرور می‌کنیم. در بخش سوم، مراحل روش پیشنهادی را به همراه نوآوری‌های اضافه‌شده به الگوریتم پایه شرح می‌دهیم. در بخش چهارم به ارائه نتایج آزمایش‌ها و ارزیابی عملکرد روش پیشنهادی می‌پردازیم و در بخش پنجم در رابطه با نتایج و کارهای آینده بحث می‌کنیم.

۲- کارهای مرتبط

در تحقیق انجام‌شده توسط Kremmer و همکاران، سامانه کشف نفوذی براساس یکپارچه‌سازی روش‌های کشف ناهنجاری و سوء استفاده ارائه شده است [۱]. در کار ارائه‌شده، الگوی فعالیت‌های یک حمله سایبری استخراج می‌شود. روش ارائه‌شده برای تشخیص الگوهای رفتاری در این تحقیق به مانند روش‌هایی که در تحقیقات Debar و Sperotto و همکاران بوده عمدتاً از روش‌های سنتی یعنی کشف فعالیت‌های سطح پایین استفاده می‌کنند [۳-۲]. ما به دنبال روشی هستیم به مانند آن‌چه در تحقیقات Lee و همکاران [۴] و Yan و همکاران [۵] پیشنهاد شده تا بتوانیم با روش‌های نوین داده‌کاوی به استخراج الگوهای رفتاری و کشف فعالیت‌های سطح بالای یک حمله سایبری بپردازیم. Yan و همکاران برای کشف فعالیت‌های سطح بالا از روش‌های داده‌کاوی نوین مانند استفاده از آمار مربعات خی دو بهره گرفته‌اند. به غیر از روش‌های پیشین، آنالیز مؤلفه اصلی [۶]، الگوریتم‌های ژنتیک [۷]و شبکه‌های عصبی مصنوعی [۸]، استنتاج فازی [۹] همگی روش‌های کشف نفوذ معروفی هستند که در آنالیز ترافیک شبکه برای کشف فعالیت‌های سطح بالای یک حمله سایبری پیچیده استفاده می‌شوند اما اغلب از کشف روابط میان دنباله‌های دریافتی از شبکه غافل می‌مانند.

تا کنون از روش‌های مبتنی بر زبان در کشف ناهنجاری‌ها در موضوعات مختلف استفاده شده است. Adler و همکاران با ترکیب علوم پردازش زبان طبیعی و ابر داده[2] قادر به تشخیص تحریف علمی در متون Wikipedia شده‌اند [۱۰]. در تحقیق Lekhac و همکاران با استفاده از پردازش زبان طبیعی به تحقیق و کشف جرائم در اطلاعات بزرگ پرداخته شده و سرعت و دقت در این روند بهبود یافته است [۱۱]. Peng و همکاران از روش‌های مبتنی بر زبان برای کشف فعالیت‌های سطح بالای انسانی در یک بازی پینگ‌پنگ استفاده کرده و نتایج آن‌را با استفاده از روش مدل مخفی مارکف مقایسه کرده اند [۱۲]. با الهام از این رویکردها ما به دنبال روشی هستیم که با استفاده از کشف فعالیت‌های سطح بالا به تشخیص ناهنجاری در یک شبکه منجر می‌شود.

تاکنون کمتر از روش‌های مبتنی بر زبان برای کشف نفوذ حملات سایبری استفاده شده است. تنها محققانی همچون Rieck [۱۳] پیشنهاد استخراج‌کردن ویژگی‌های زبان نظیر n-gramها و کلمات را از روی حداکثر بار ارتباط و اعمال کشف ناهنجاری بدون نظارت را اعمال نمودند. تفاوت این کار با تحقیق ما در این است که آن‌ها فقط اطلاعات محلی دنباله‌های بایت را استفاده می‌کردند اما رویکرد ما وابستگی در فواصل دوربین الگوهای زبانی و فعالیت‌های حمله را به‌منظور تشخیص تهدیدهای سایبری مدل‌سازی می‌کند.

۳- روش پیشنهادی برای تشخیص فعالیت‌های سطح بالای حملات سایبری

روش به‌کار گرفته‌شده برای تشخیص حملات سایبری روش حریصانه‌ای است که با الهام از القاء گرامر در پردازش زبان طبیعی به‌دست آمده است. ما به دنبال ارتباط پنهان موجود بین هشدارها هستیم و قصد داریم با داشتن یک ترکیب منطقی از

۱- استخراج روابط عبارت است از تعیین نسبت‌ها و ارتباط‌ها میان هستارهای اسمی

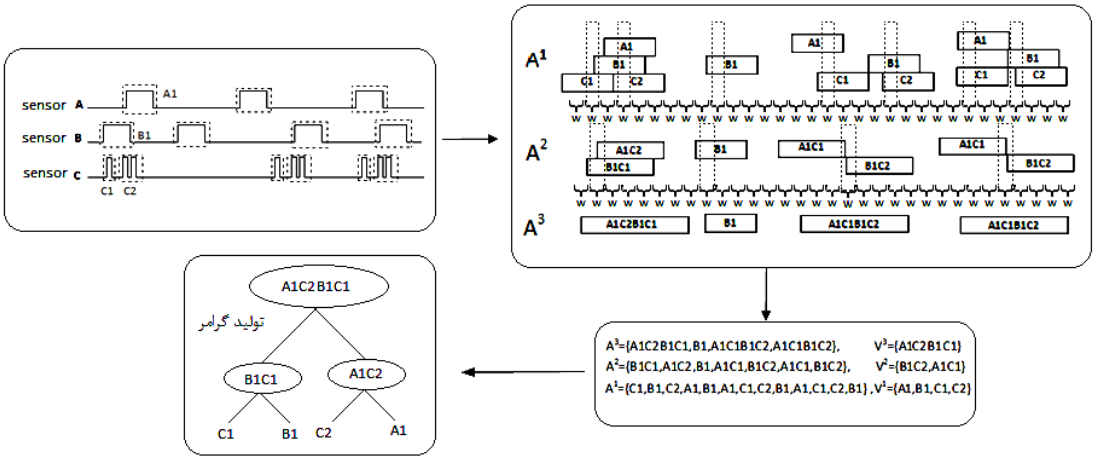
2- Metadata

هشدارها به مانند آنچه در کشف bigramها در متون و اسناد انجام می‌شود، کار کنیم. در این روش، ابتدا فعالیت‌های سطح بالا از طریق در پرانتز قراردادن دو زیرفعالیت که از لحاظ آماری مکرراً در کنار هم قرار گرفته‌اند، شناسایی می‌شوند در مرحله بعد، فعالیت‌های مشابه به لحاظ شکلی یا معنایی را به فعالیت‌های سطح بالاتر تعمیم می‌دهیم. برای استفاده از رویکرد مبتنی بر زبان در یادگیری بدون نظارت ساختار فعالیت‌ها از روابطی که در ادامه براساس شکل (۱) توضیح داده شده است استفاده می‌کنیم. با فرض داشتن حسگرهای A و B و C که هرکدام دارای فعالیت‌های مرتبط هستند، ما متن فعالیت‌ها در سطوح مختلف را به شکل زیر تعریف می‌کنیم: $A=\{A^1,\ldots,A^L\}$ که در آن $A^l=\{a_1^l,\ldots,a_{|A^l|}^l\}$ و در آن هر a_i^l بیانگر فعالیت iام در سطح l است. در این‌جا A^1 نشان‌دهنده فعالیت‌های پایین‌ترین سطح است، درحالی‌که AL نشان‌دهنده فعالیت‌های بالاترین سطح است. با توجه به A، در پایین شکل (۱) بعد از کشف با هم گذاری‌ها و خوشه‌بندی سلسله مراتبی، با داشتن درخت روابط، یک گرامر فعالیت G={V,R} تعریف می‌کنیم که در آن، V بیان‌گر روابطی است که ساختار گرامر را مشخص می‌کنند. به‌طور خاص، $V=\{V^1,\ldots,V^L\}$ است که در آن، V^l بیان‌گر فهرست واژگان فعالیت Lامین سطح معنایی است به‌طوری‌که هر فعالیت سطح L به شکل $a_i^l \in \{V^1,\ldots,V^L\}$ است؛ بنابراین، هر رابطه مستقیم $(V_i^{l1} \rightarrow V_j^{l2})$ اگر $l_1>l_2$ و V_i^{l1} یک فعالیت ترکیبی[1] یا یک فعالیت تعمیم‌یافته[2] از V_j^{l2} است.

یافتن الگوهای تکرارشونده در سری‌های زمانی می‌تواند به ایجاد یک روش برای ساختن فرهنگ واژگان اولیه کمک کند. در بالای سمت چپ شکل (۱) مشاهده می‌شود که به زیر دنباله‌های مشابه برچسب‌هایی برای ساختن فرهنگ واژگان انتساب داده شده که با توجه به قرارگرفتن در یک پنجره زمانی می‌توان برای ادغام آن‌ها تصمیم‌گیری کرد (بالای سمت راست شکل). این پنجره از ابتدای دنباله شروع به حرکت روبه‌جلو کرده و فعالیت‌های نزدیک به هم را پیدا می‌کند. بدین ترتیب، می‌توان فعالیت‌های سطح بالا را استخراج نموده و با تولید گرامر از روابط به‌دست‌آمده، لغات سطح بالاتر را تولید کرد. در ادامه الگوریتم تولید گرامر را می‌بینیم:

Algorithm: grammer induction on network trace sequence
Input: S,network trace sequence obtained from MD_DBscan clustering
Input:αT, merging threshold parameter
Input: δT, generalization threshold parameter
Output: G={V,R}, discovered hierarchical grammar
Output: A, hierarchical network activities labeled using V

1 (A^1,V^1)= initialize (S)
2 $(A',V')=(A^1,V^1)$
3 l=1
4 While True do
5 l=l+1
6 (A^l,V^l)= discover super-activities from (A',V')(merge each a_x^1, a_y^1 if collocation $(a_x^1, a_y^1) >\alpha T$)
7 Break if $|V^l|==0$
8 For $v_i \in V^l$, $v_j \in V'$ do
9 Add edges(v_i, v_j) into R for all collocations
10 End
11 (A',V')= generalize vocabulary from (A^l,V^l)
12 for each a_i in A^1
13 assign a_i to cluster c_i
14 Compute distance between clusters(maximum distance)
15 Find the closest (most similar) pair of clusters and merge them into a single cluster in V'
16 Compute distances (similarities) between the new cluster and each of the old clusters.
17 Repeat steps 15,16 until all distances are greater than (1-δT)
18 End
19 $V=(V^1\ldots V^{l-1})$
20 Return G={V,R},A

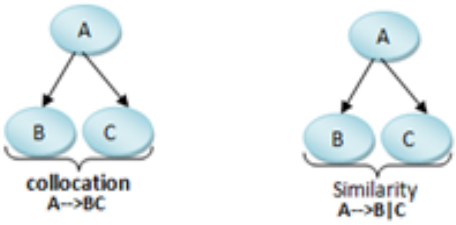


شکل (۱): رویکرد مبتنی بر زبان در یادگیری بدون نظارت براساس القاء گرامر G

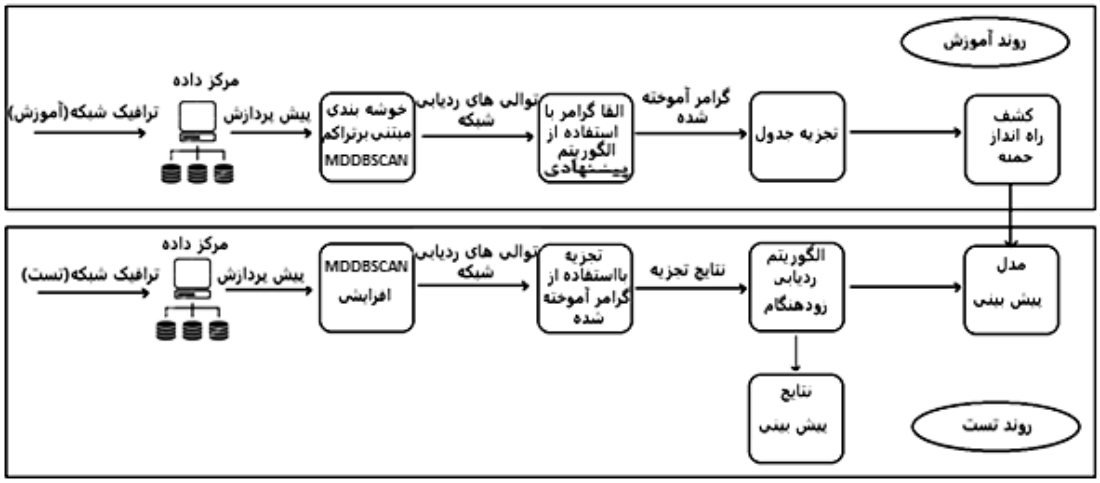
1- Super Activity
2- Generalized Activity

تشخیص رفتار خاص حملات سایبری را ارائه دهیم از روابط و

تعاریفی که در ادامه آمده است، استفاده می‌کنیم: فرض می‌کنیم سری‌های زمانی ترافیک شبکه‌ای به‌صورت $S=\{S_1,\ldots,S_N\}$ به طول N باشند. هر رکورد S_i با دریافت اطلاعات از حسگرهای تشخیص نفوذ d بعدی به‌صورت $S_i=\{O_i^1,\ldots,O_i^d\}$ تعریف می‌شود. دنباله ردیابی[1] حاصل از ترافیک شبکه‌ای اخذشده را می‌توان به‌صورت $T=\{T_1,\ldots,T_N\}$ نشان داد که در آن، $T_i= F_M(S_i)$ است و F_M در این رابطه تابع نگاشتی است که دریافت اطلاعات از حسگرهای d بعدی S_i را به یک نماد ردیابی واحد T_i ترافیک شبکه‌ای تبدیل می‌کند. به بیان دیگر، دنباله ردیابی T یک نمایش تک‌بعدی از ترافیک شبکه‌ای خام است که انجام محاسبات بیشتر را به‌دلیل کاهش ابعاد آسان‌تر می‌سازد. به‌علاوه، در این الگوریتم فعالیت شبکه‌ای چندسطحی[2] را با یک مجموعه به شکل $A=\{A^1,\ldots,A^L\}$ با L سطح تعریف نمودیم که $A^1=\{a_1^l,\ldots,a_{|A^l|}^l\}$ و در آن هر a_i^l بیان‌گر iامین فعالیت در سطح l است. توجه کنید که A^1 در این‌جا به معنای فعالیت‌های پایین‌ترین سطح (یعنی $A_1=T$) است درحالی‌که A^L دلالت بر بالاترین سطح دارد؛ چنانچه بخواهیم یک نگاشت مفهومی از سطوح مختلف معنایی از زبان طبیعی در زبان شبکه ارائه کنیم می‌توانیم پایین‌ترین سطح فعالیت را برابر با ویژگی‌های کلی ارتباطات در شبکه درنظر بگیریم، سطح بالاتر از معنا که در زبان طبیعی کلمات هستند می‌توانند به فعالیت‌های واحد تجزیه‌ناپذیر در شبکه نگاشت شوند و سطح بالاتر که در زبان طبیعی جملات هستند به وظایف در شبکه که دنباله‌ای از عملیات هستند، نگاشت می‌شوند. گرامر فعالیت شبکه‌ای G به‌صورت $G=\{V,R\}$ تعریف می‌شود که V دلالت بر فهرست واژگان فعالیت[3] در سطوح مختلف با L سطح دارد و R دلالت بر روابطی که ساختار گرامری را مشخص می‌کنند. دقیق‌تر بگوییم، $V=\{V^1,\ldots,V^L\}$ که V^L بیان‌گر فهرست واژگان فعالیت در سطح مفهومی L ام می‌باشد به‌طوری‌که $a_i^l \in \{V^1 U \ldots U V^L\}$. پس، تعریف می‌کنیم که هر رابطه مستقیم $(V_i^{l1} \rightarrow V_j^{l2})$ اگر $l_1>l_2$ و V_i^{l1} یک فعالیت ترکیبی[4] یا یک فعالیت تعمیم‌یافته از V_j^{l2} است. همان‌طور که بیان شد گرامر G یک گرامر مستقل از متن است. گرامرهای مستقل از متن در زبان‌شناسی کاربردهای زیادی دارند و برای توصیف ساختار جملات و کلمات در زبان طبیعی به‌کار می‌روند. در گرامرهای مستقل از متن قواعد به شکل یک به یک، یک به چند و یک به هیچ تعریف می‌شوند و سمت چپ هر قاعده همیشه یک غیرپایانه قرار دارد. قواعد از یک نماد شروع به‌نام S آغاز می‌شوند. در گرامر مورد استفاده ما تمام نمادهای غیرپایانی در مجموع نمادهای شروع قرار می‌گیرند و قواعد می‌توانند از هرکدام از غیرپایانه‌ها شروع شوند. در مرحله اول ساخت گرامر که تمرکز به باهم‌گذاری لغات سطح پایین و ادغام آن‌ها در یک لغت داریم می‌توانیم قاعده تشکیل‌شده را بدین شکل تعریف کنیم: $A \rightarrow BC$ در مرحله دوم که بعد از اندازه‌گیری شباهت لغات آن‌ها را خوشه‌بندی کرده و لغات جدید را تعمیم می‌دهیم، می‌توانیم قواعد را به این شکل تعریف کنیم: $A \rightarrow B|C$



شکل (۲): ساخت قاعده تولید از درخت روابط (سمت چپ). این قاعده بدین معنا است که غیرپایانه A هم‌زمان B و C را در کنار هم تولید می‌کند. ساخت قاعده تعمیم از درخت روابط (سمت راست) . این قاعده بدین معنا است که غیرپایانه A می‌تواند B یا C را تولید کند.



شکل (۳): مراحل تشخیص فعالیت‌های سطح بالای یک حمله‌ی سایبری

1- Trace Sequence
2- Multi-Level Network Activity Text
3- Activityvocabulary
4- Super Activity ای

حمله، می‌پردازیم. در حقیقت به‌مانند آن‌چه در شکل (۳) نشان داده شده است، عمل می‌کنیم. در ادامه مراحل الگوریتم را شرح می‌دهیم.

مرحله اول: استخراج نمادهای اولیه از ترافیک شبکه اخذشده با استفاده از MDDBSCAN

در این مرحله، ابتدا پیش‌پردازشی بر روی ترافیک شبکه‌ای، انجام می‌گیرد. ما براساس رویدادنگاری‌هایی[1] که ورودی سامانه کشف نفوذ پیشنهادی هستند، به جمع‌آوری آمار و ویژگی‌های مرتبط، خواه پیوسته یا گسسته می‌پردازیم تا بتوانیم مفاهیم سطحی[2] ترافیک شبکه را دریافت کنیم. این ویژگی‌ها برای هر ردیابی شبکه به مبانی تراکنش‌های شبکه نظیر انواع پروتکل‌ها محدود نمی‌شوند بلکه ویژگی‌های سطح بالای ارتباطات شبکه‌ای نظیر مدت ارتباط و درصد خطاهای ارتباطاتی را نیز می‌توان تعریف کرد که به تشخیص ارتباطات عادی از حملات کمک می‌کنند. بعد از مرحله پیش‌پردازش داده، ترافیک شبکه با استفاده از الگوریتم‌های خوشه‌بندی[3] به یک دنباله ردیابی تبدیل می‌شود. به‌طوری‌که ردیابی‌های شبکه‌ای مشابه توسط نماد یکسانی در دنباله ردیابی نمایش داده می‌شوند؛ یعنی، هر رکورد تراکنش ترافیک شبکه به یک نماد واحد مشابه یا یک کلمه در زبان طبیعی تبدیل می‌شود که منجر به یک دنباله ردیابی $T=\{T_1,\ldots,T_N\}$می‌شود. دنباله T بیان‌گر ترافیک شبکه شامل N تراکنش است که در آن، $T_i=F_M(S_i)$ و F_M تابع تبدیل از ترافیک شبکه‌ای خام به نمادهای ردیابی مبتنی بر نتایج خوشه‌بندی است. در همین مرحله برای انجام خوشه‌بندی ما از الگوریتم MD_DBSCAN[4] [۱۴] که از الگوریتم‌های بهبودیافته DBSCAN[5] است، استفاده کرده‌ایم. الگوریتم MD_DBSCAN می‌تواند خوشه‌هایی با چگالی‌های متفاوت، یعنی خوشه‌هایی که در تراکم متفاوت باشند را پیدا کند. همچنین، در دنباله‌های جدید به‌علت پویایی فضای داده از الگوریتم IMD_DBSCAN[6] استفاده کرده‌ایم. این الگوریتم برای به‌روزرسانی خوشه‌ها نیاز به پرس‌وجوی ناحیه‌ای بسیار کمتری دارد و دارای افزایش سرعت بسیار خوبی است. برای محاسبه شباهت هنگام خوشه‌بندی، به‌دلیل وجود ویژگی‌های پیوسته و گسسته در مجموعه دادگان، از معیار شباهت Gower [۱۶] در الگوریتم MD_DBSCAN استفاده شده است.

مرحله دوم: کشف فعالیت‌های ترکیبی مبتنی بر فعالیت‌های سطح پایین

برای کشف ابرفعالیت از میان فعالیت‌های شبکه‌ای سطح پایین، ابتدا فرض می‌کنیم که جفت فعالیت‌هایی که به‌طور مشترک در یک پنجره اتفاق می‌افتند ممکن است مؤلفه‌هایی از یک فعالیت ترکیبی باشند. برای یک جفت فعالیت معین $[v_i^l,v_j^l]$، فرکانس رخدادن مرزی v_i^l و v_j^l و نیز فرکانس رخدادن مشترک آن‌ها را در دنباله ردیابی جمع‌آوری می‌کنیم. بر اساس این فرکانس‌های مرزی و مشترک، باهم‌گذاری[7] محاسبه می‌شود تا تعیین شود که آیا این دو تشکیل یک فعالیت ترکیبی را می‌دهند یا خیر. برای مسئله ما به‌طور خاص، باهم‌گذاری بین دو فعالیت اشاره می‌کند که رخدادن هم‌زمان آن‌ها ناشی از وابستگی ذاتی بین آن‌ها است و صرفاً تصادفی نیست. اهمیت مجموعه آماری می‌تواند با ساخت جدول احتمال وقوع به ازای هر جفت فعالیت $[v_i^l,v_j^l]$ و محاسبه آمار مجذور کای (X^2) اندازه‌گیری شود [۱۷] که بعداً با یک پارامتر آستانه‌ای (نظیر $X^2_{0.05}(1)=3.841$ که مقادیر بحرانی آن در جدول آمار مربعات کای آورده‌شده) مقایسه شود تا برای ادغام $[v_i^l,v_j^l]$ و ساخت یک فعالیت ترکیبی تصمیم‌گیری کند.

آزمون کای دوی دومتغیره مواقعی استفاده می‌شود که اطلاعات جمع‌آوری‌شده به فراوانی مشاهده شده و طبقات دو متغیر تعلق دارد بدین معنی که یک گروه آزمودنی براساس دو متغیر در طبقات مختلف طبقه‌بندی می‌شوند و سؤالی که در چنین مواردی مطرح است، وجود یا عدم وجود ارتباط بین متغیرها است؛ به‌عبارت دیگر، هدف این آزمون پاسخگویی به این سؤال است که آیا رابطه معنی‌داری بین دو متغیر وجود دارد یا خیر.

برای آزمون این فرضیه غالباً مشاهدات در یک جدول توافقی (r*c) که r تعداد سطرها و c تعداد ستون‌ها است جمع‌آوری می‌شوند. آماره آزمون مشابه آزمون نیکویی برازش به‌صورت رابطه (۱) به‌دست می‌آید:

$$X^2=\sum_{i,j}\frac{(O_{ij}-E_{ij})^2}{E_{ij}} \qquad (۱)$$

مرحله سوم: تعمیم فهرست واژگان با استفاده از خوشه‌بندی سلسله مراتبی

کشف باهم‌گذاری زمانی تأثیر دارد که دنباله‌های تکرارشونده‌ای از زیرفعالیت‌ها[8] وجود داشته باشند. دقیق‌بودن این تکرارها برای فعالیت‌های شبکه‌ای به‌دلیل از دست‌رفتن اطلاعات در زمان انجام خوشه‌بندی احتمال کمی دارد. به‌منظور کشف بهتر فعالیت‌های سطح بالا در چنین شرایطی، ما باید قبل از ساخت جداول

1- Log
2- Shallow Semantics
3- Clustering
4- Multi_Density_Dbscan
5- Density Based Spatial Clustering Application With Noise
6- Incremental Md_Dbscan
7- Collocation
8- Sub-Activity

احتمال وقوع، فعالیت‌های مشابه را تعمیم دهیم. همان‌طوری که در [۱۲] پیشنهاد شده است، شباهت بین دو فعالیت در سطح یکسان می‌تواند برحسب شباهت محتوایی[1] θ_e و شباهت متنی[2] θ_x اندازه‌گیری شود. همچنین، در [۱۸] معیار فاصله ویرایش[3] برای محاسبه شباهت دو فعالیت پیشنهاد شده است. ما این سه معیار را با محاسبه میانگین حسابی در قالب یک معیار شباهت واحد θ تجمیع می‌کنیم. این تجمیع θ سپس در یک الگوریتم خوشه‌بندی لینک کامل[4] استفاده می‌شود که در آن یک لینک بین دو فعالیت وجود دارد تنها اگر θ شباهت آن‌ها بزرگ‌تر از آستانه θt باشد. درنتیجه، همه فعالیت‌های دسته‌بندی‌شده در خوشه یکسان به‌شدت شبیه به یکدیگر می‌باشند. دلیل استفاده از چند معیار شباهت، دستیابی به جنبه‌های متفاوتی از شباهت بین فعالیت‌ها است.

۱- شباهت محتوایی ($\emptyset_e$): فعالیت‌های مشابه از نظر محتوا شامل دو حالت اصلی را تشکیل می‌دهد. حالت اول، وقتی است که فعالیت‌های چندگانه، هر دو تکرارهای زیرفعالیت یکسانی باشند. نظیر A0A0A0A0A0 و A0A0A0A0A0. حالت دوم وقتی است که دو فعالیت تنها در بخش جزئی از ترکیب خود باهم تفاوت دارند. برای مثال، A0A0A0A0A0 و A0A0A0A0A1 برای درنظرگرفتن این دو حالت برای دو عبارت (فعالیت) مشابه V_1 و V_2 ابتدا n-gram آن‌ها را محاسبه کرده‌ایم. فرض کنید $|v_1|\leq|v_2|$ و ۲ باشد.

$$\text{Precision}=\frac{|nGram(v_1)\cap nGram(v_2)|}{|nGram(v_1)|} \quad (۲)$$

$$\text{Recall}=\frac{|nGram(v_1)\cap nGram(v_2)|}{|nGram(v_2)|} \quad (۳)$$

که در آن‌ها، صورت کسر نشان‌دهنده 2-gram های مشترک بین رشته V_1 و V_2 می‌باشند.

برای به‌دست‌آوردن شباهت، دقت و بازخوانی با یک میانگین هارمونیک که در رابطه (۴) نشان داده شده است، ادغام می‌شوند تا F1 از رابطه (۵) به‌دست آید.

$$(a_1,\ldots,a_n)=\frac{n}{\frac{1}{a_1}+\cdots+\frac{1}{a_n}} \quad (۴)$$

$$F1=2*\frac{precision*recall}{precision+recall} \quad (۵)$$

درحالتی‌که دو عبارت از نظر محتوا بسیار شبیه بوده لیکن از نظر طول متفاوت هستند، رابطه (۵) را به شکل رابطه (۶) نرمالیزه می‌کنیم:

$$\theta_e=2*\frac{precision*\sqrt{recall}}{precision+\sqrt{recall}} \quad (۶)$$

۲- شباهت زمینه ($\emptyset_x$): در شباهت زمینه فعالیت‌هایی که در محیطی مشابه رخ می‌دهند را بررسی می‌کنیم. ما زمینه یک فعالیت را به‌صورت $\vec{c}=(n_1,\ldots,n_{|v|})$ ارائه می‌کنیم که بیان‌گر تعداد دفعات رخدادن مشترک هریک از فعالیت‌ها با فعالیت مورد نظر است. سپس شباهت زمینه θ_x بین دو فعالیت را با فاصله کسینوسی بین بردارهای زمینه آن فعالیت‌ها به‌مانند آنچه در رابطه (۷) آمده است، درنظر می‌گیریم.

$$\theta_x=\frac{\overrightarrow{c_1}.\overrightarrow{c_2}}{|\overrightarrow{c_1}||\overrightarrow{c_2}|} \quad (۷)$$

۳- شباهت با معیار فاصله ویرایش (d_e): فاصله ویرایش یا فاصله لون‌اشتاین برای محاسبه میزان تفاوت میان دو رشته در علوم رایانه و نظریه داده‌ها استفاده می‌شود. فاصله ویرایش عبارت است از کمترین تعداد عملیات مورد نیاز برای تبدیل یک رشته به رشته دیگر. این عملیات شامل حذف، اضافه و تبدیل است. یک الگوریتم رایج برنامه‌نویسی پویا برای محاسبه فاصله لون‌اشتاین شامل استفاده از یک ماتریس $(n+1)\times(m+1)$ می‌شود، که n و m طول دو رشته هستند. این الگوریتم برپایه الگوریتم wagner_fischer برای ویرایش فاصله است. به‌عنوان مثال، برای تبدیل رشته سمت چپ به رشته سمت راست، سه مرحله عملیات نیاز داریم:

A6A2A3A3A7A4→A1A2A3A3A2A4A5

مرحله اول: جایگزینی A1 به‌جای A6 ، مرحله دوم: جایگزینی A2 به‌جای A7 و مرحله سوم: اضافه‌کردن A5 در انتها؛ بنابراین، فاصله ویرایش بین این دو رشته ۳ است. برای نرمالیزه‌کردن معیار فاصله ویرایش در فاصله بین صفر و یک و تبدیل آن به معیاری برای شباهت از رابطه (۸) استفاده می‌کنیم:

$$\text{Edit distance similarity}=\frac{\max(length(v_1),lenght(v_2))-editdistance(v_1,v_2)}{\max(length(v_1),lenght(v_2))} \quad (۸)$$

خوشه‌بندی، نیازمند تجمیع شباهت‌های محتوا و زمینه و ویرایش در یک مقیاس شباهت واحد $\emptyset$ هستیم. براساس مشاهده، میانگین‌های حسابی، هندسی و هارمونیک، همگی گزینه‌های معقولی هستند. اگرچه به‌منظور این‌که یکی از شباهت‌ها (و نه لزوماً هر سه) را برای تعمیم کافی بسازیم، به میانگین حسابی نیاز داریم. این $\emptyset$ تجمیع سپس در یک الگوریتم خوشه‌بندی پیوند کامل[5] استفاده می‌شود که طبق آن یک پیوند بین دو فعالیت

1- Content Similarity
2- Context Similarity
3- Edit Distance
4- Complete Link Algorithm
5- Complete Link

وجود دارد تنها اگر شباهت آن‌ها $\emptyset$ بزرگ‌تر از یک آستانه $\emptyset_{thr}$ باشد. درنتیجه، همه فعالیت‌هایی که باهم در یک خوشه دسته‌بندی می‌شوند، همگی بسیار شبیه به یکدیگر هستند.

مرحله چهارم: کشف "نشانه‌های شروع رفتار"[1]

وقتی گرامر ترافیک شبکه القاء گردید، می‌توان از قواعد گرامری برای تجزیه[2] دنباله‌های ردیابی شبکه استفاده کرد و در سطوح مختلف در طول دنباله ردیابی با استفاده از الگوریتم‌های تجزیه، همه ساختارهای معتبر زیردرخت تجزیه را به‌دست آورد که در پردازش زبان طبیعی (NLP) به‌عنوان سازندگان[3] نامیده می‌شوند.

برای کشف سازندگان در متن، از الگوریتم تجزیه چارت CYK بهره می‌بریم که این الگوریتم از برنامه‌نویسی پویا استفاده می‌کند. الگوریتم CYK یک الگوریتم اشتقاق است که روی گرامرهای مستقل از متن عمل می‌کند. این الگوریتم برای تشخیص عضویت یا عدم عضویت یک رشته خاص در یک زبان مستقل از متن به کار می‌رود. روش این الگوریتم بالا به پایین بوده و از الگوریتم‌های پویا است. نسخه استاندارد این الگوریتم فقط روی گرامرهایی به شکل فرم نرمال چامسکی[4] عمل می‌کند. البته تمام گرامرهای مستقل از متن قابلیت تبدیل به فرم نرمال چامسکی را دارند. زمان اجرایی این الگوریتم در بدترین حالت $O(N^3.G)$ است که در آن N طول رشته مورد نظر و G اندازه گرامر موجود است. هدف ما از اجرای این الگوریتم، یافتن تمام رشته‌های موجود در دنباله S است که در زبان مستقل از متن وجود دارند، علاوه‌بر آن، مکان این رشته‌ها در دنباله برای ما حایز اهمیت است. همچنین، به‌دلیل وجود قواعد تعمیم در گرامر، که قواعد یک به یک می‌باشند، قواعد به شکل فرم چامسکی نیست. با این توصیف، ما نیاز به ایجاد تغییراتی در الگوریتم اصلی داریم که درنهایت آن‌را به شکل زیر بازسازی می‌کنیم: فرض کنید گرامر مستقل از متن القاءشده یک تاپل چهار جزیی $G=(N,\sum,P,R_s)$ باشد که $N,\sum,P,R_s$ به‌ترتیب بیان‌گر «مجموعه نمادهای غیرپایانی»، «مجموعه نمادهای پایانی»، «مجموعه قواعد تولید گرامر» و «نماد شروع شناسایی‌شده» باشند. می‌خواهیم بدانیم آیا دنباله‌ای از نمادهای w وجود دارد که به $L(G)$ تعلق داشته باشد یا خیر. به‌این‌ترتیب می‌توانیم همه سازنده‌های معتبر سطوح مختلف را در دنباله ردیابی پیدا کنیم. $L(G)$ همه دنباله‌هایی است که در تعداد متناهی از گام‌هایی از نماد آغازین متعلق به R_s قابل استخراج است که در مورد ما معادل با N است یعنی همه نمادها در فهرست واژگان فعالیت سطح بالا می‌توانند نمادهای آغازین معتبر یک اشتقاق از گرامر G باشند. درنهایت، نتایج در یک ساختار قابل استفاده مجدد به نام چارت ذخیره می‌شوند. وقتی گرامر G القاء می‌شود و بعد از اجرای الگوریتم CYK همه n سازنده معتبر سطح l یعنی $C^l = \{C_1^l, \dots, C_n^l\}$ در طول دنباله آموزش پیدا می‌شوند، دنباله آموزش را می‌توان توسط مجموعه‌ای از زوج‌های (c_i^l, w_i^l) به ازای سطح l نشان داد که W پنجره‌ای با طول ثابت است که شامل دنباله فعالیت‌های شبکه‌ای است که به c_i^l مربوط می‌شود و بعد از c_i^l رخ می‌دهد (شکل ۴).

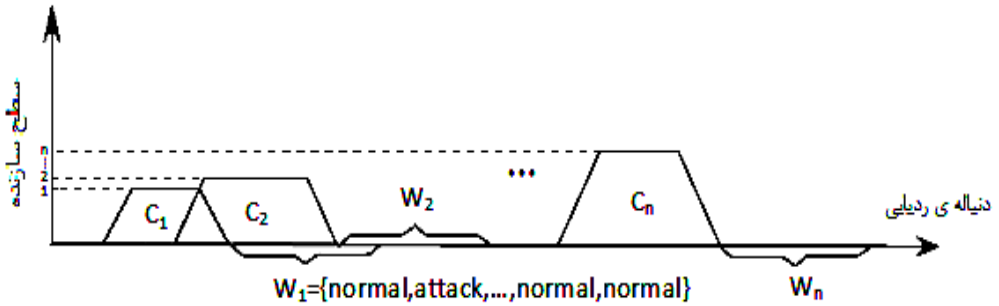
الگوریتم CYK برای تشخیص سازندگان به شکل زیر استفاده شده است:

```
algorithm:structured trigger discovery with modified CYK parsing
    input: S, trace sequence of N symbols S1,…,SN
    input: the set of M non-terminal symbols R1,…,RM
    input: maxSpanLen, the maximum length of parsing span to be considered
    output: the location of all valid constituent with corresponding derivation start symbols
1       let P be an N*N*M array of Booleans. Initialize all elements of P to false.
2       For (i=1; i=< n; i++) do
3       For each unit production RmSi do
4       P[i,1,m]= True
5       End
6       End
7       // i iterates over the length of the span
8       For(i=2; i=< maxSpanLen; i++) do
9       // j iterates over the start of the span
10      For (j=1; j=<n-i+1; j++) do
11      // k iterates over the partition of the span
12      For (k=1; k<i ; k++) do
13      For each generalization rule RARB do
14      If P[j,k,B] then
15      P[j,k,A]= true
16      End
17      End
18      For eachproduction rule RARBRc do
19      IfP[j,k,B] andP[j+K,i-k,C] then
20      P[j,K,A]= true
21      End
22      End
23      End
24      End
25      End
26      If any of P[j,i,x] is true (x is iterated over the set of starting symbols R) then
27      Return
end
```

برای شناسایی "نشانه‌های شروع رفتار" سازنده که فعالیت‌های حمله در حال وقوع احتمالی را پیش‌بینی می‌کنند،

1- Trigger
2- Parse
3- Constituents
4- Chamesky Normal Form (CNF)

مجدداً جدول احتمال رخداد را ایجاد کرده و آمار مجذور کای (X^2) را برای اندازه‌گیری همبستگی بین هر سازنده و رفتار شبکه محاسبه می‌کنیم. نشانه‌های شروع رفتار یا Trigger ها در علم NLP به معنای نمایه‌هایی است که قبل از انجام هر کار، عامل آن را در ذهن خود می‌سازد. به‌عبارت دیگر، این نمایه‌های داخلی مانند حرف‌هایی است که عامل قبل از انجام عمل با خود می‌گوید و یا تصویری که از عمل در ذهن خود تجسم می‌کند. در واقع، Triggerهایی که ما در دنباله آموزش پیدا می‌کنیم، سازندگانی هستند که با رفتار شبکه همبستگی خاصی دارند و معمولاً بعد از پدیدارشدن در دنباله، رفتار خاصی را به‌دنبال دارند که مقدار این همبستگی با رفتار را می‌توان از روش‌های آماری که در ادامه گفته می‌شود، محاسبه کرد.



شکل (۴): سازندگان سطوح مختلف و پنجره‌های فعالیت پیش روی آن‌ها در دنباله‌ی ردیابی

با جستجوی در دنباله ردیابی آموزش از چپ به راست، فرکانس‌های مشترک همه جفت‌های (c_i^l,A_i) به ازای سطح l را جمع‌آوری می‌کنیم که $A_i \in w_i^l$ بیان‌گر برچسب نماد ردیابی است که می‌تواند دودویی (نظیر «عادی» یا «حمله») یا چندتایی (نظیر «عادی»، «DoS»، «probe») باشد. چنین تعدادی در یک جدول فرکانس مشترک[1] (سمت چپ شکل ۵) ثبت می‌شود. جدول فرکانس حاشیه[2] (وسط شکل ۵) نیز به‌طور مشابه ساخته می‌شود. سپس جدول احتمال وقوع (سمت راست شکل ۵) بر طبق جداول فرکانس مشترک و مرزی ساخته می‌شود. آمار مجذور کای هر جفت (c_i^l,A_i) براساس جدول احتمال وقوع محاسبه می‌شود که برای استفاده بعدی یعنی پیش‌بینی رفتار در حال وقوع شبکه با داشتن دنباله جدید ردیابی در یک جدول، ذخیره می‌شود.

جدول احتمال وقوع

	162	~16	
normal	8	3	11
attack	2	5	7
	10	8	18

جدول فرکانس حاشیه

سازنده	نوع فعالیت	تکرار
162	-------	10
721	-------	8
-----	normal	11
-----	attack	7

جدول فرکانس مشترک

سازنده	نوع فعالیت	تکرار
162	normal	8
162	attack	2
721	normal	3
721	attack	5

شکل (۵): مثال از نحوه محاسبه آمار مجذور کای(X^2) ، جداول فرکانس مشترک و مرزی و جداول احتمال وقوع با استفاده از رابطه (۱)

1- Joint Frequency Table
2- Marginal Frequency Table

$$\frac{\left(8-\left(10*\frac{11}{18}\right)\right)^2}{\left(10*\frac{11}{18}\right)}+\frac{\left(2-\left(10*\frac{7}{18}\right)\right)^2}{\left(10*\frac{7}{18}\right)}+\frac{\left(3-\left(8*\frac{11}{18}\right)\right)^2}{\left(8*\frac{11}{18}\right)}+\frac{\left(5-\left(8*\frac{7}{18}\right)\right)^2}{\left(8*\frac{7}{18}\right)}$$

$$=11.61 \geq \propto_{0.95}= 0.384 \qquad (۹)$$

مرحله پنجم: کشف حمله سایبری در مرحله اولیه حمله

در این مرحله ابتدا همبستگی بین هر" نشانه شروع رفتار" سازنده شناسایی‌شده در دنباله ردیابی را با انواع حملات احتمالی بررسی می‌کنیم؛ یعنی با هر بار خواندن یک نماد ردیابی جدید تست، قادر به یافتن همه " نشانه شروع رفتار" سازنده معتبر سطوح مختلف به ازای هر پنجره فعالیت شبکه خواهیم بود.

فرض کنید که یک " نشانه شروع رفتار" سازنده معتبر C در دنباله تست شناسایی شود که W پنجره فعالیت مرتبط با آن است. ابتدا با مراجعه به جدول نگاشت‌های ذخیره‌شده که در گام کشف "نشانه شروع رفتار" تهیه‌کرده‌ایم، به دنبال همه انواع فعالیت‌های شبکه‌ای که ممکن است با C آغاز شوند، می‌گردیم که آن‌را با A نشان می‌دهیم. سپس، به‌منظور انجام پیش‌بینی خود، آمار مجذور کای را بین هر جفت (C,Ai) که $Ai \epsilon A$ با آستانه‌های ثابت مقایسه می‌کنیم. در این‌جا فرض می‌کنیم که A حاوی هم برچسب منفی (به معنای «عادی») و هم برچسب مثبت (به معنای «حمله»، «DoS») است زیرا اگر C فقط برای

شروع فعالیت‌های عادی امکان‌پذیر باشد، نیازی به گرفتن آمار از آن نداریم و با دیدن C در دنباله می‌توانیم به‌سادگی فعالیت بعدی را پیش‌بینی کنیم.

درنهایت، ما قادر به ارائه نتایج پیش‌بینی مبتنی بر الگوریتم زیر خواهیم بود که در آن normalThr و attackThr بیانگر دو مقدار آستانه‌ای آمار مجذور کای هستند. برای مثال اگر آمار مجذور کای جفت سازنده-فعالیت(C,Ai) بزرگ‌تر از آستانه مربوط به خود باشد، پیش‌بینی مطابق با Ai تعیین می‌گردد.

الگوریتم تصمیم‌گیری نهایی برای پیش‌بینی نتایج به شکل زیر خواهد بود:

Algorithm: prediction of network behavior on the trace sequence
Input: S, the trace sequence , indexed by [0,1,…,N-1]
Input: chart, derived from CYK parsing on the sequence
Input: chiSqrMapping, the mapping from each constituent trigger to its X^2 statistics at each vocabulary level
Input: window size
Input: maxLevel, the maximum level of constituent
Output: prediction label for each valid trigger along the test sequence

```
1   startIndex=0
2   while startIndex <= N-windowSize do
3   triggerFound= false
4   prediction window=
5   S_test [startIndex+windowSize]
6   possible Triggers= find all valid constituent triggers for prediction window by looking up chart
7   target Trigger=get the constituent trigger up to level maxLevel from possible Triggers
8   A= get all possible network activity types that can be triggered by target trigger
9   If normal ϵ A then
10  If chiSqr Mapping
11  [(target Trigger,normal)]> normalThr then
12  Prediction= normal
13  Else
14  Prediction= attack
15  End
16  Else
17  Maxchisqr=max(getChisqr Values(chisqrMapping,A))
18  If maxchisqr>attachThr then
19  Prediction= attack
20  Else
21  Prediction= normal
22  End
23  End
24  startIndex= startIndex+1
25  end
```

۴- نتایج و بحث

در ادامه به شرح آزمایش و روش‌های ارزیابی آن می‌پردازیم.

۴-۱- آزمایش‌ها

در شبیه‌سازی الگوریتم از دادگان KDD99 برای آموزش و تست الگوریتم پیشنهادی استفاده شده است[۱۵] که بر روی این دادگان، ابتدا نمونه‌گیری و سپس خوشه‌بندی با استفاده از الگوریتم MDDBSCAN اجرا شده است. از خروجی این مرحله برای ساخت گرامر همان‌گونه که گفته شد، استفاده شده است. درنهایت، برای ارزیابی دقت تشخیص فعالیت‌های سطح بالای یک حمله سایبری از ارزیابی گرامر ساخته‌شده نهایی بهره بردیم.

۴-۲- ارزیابی

ارزیابی نتایج آزمایش‌ها در دو مرحله برای خوشه‌بندی MDDBSCAN و برای گرامر ساخته‌شده نهایی انجام شده است.

۴-۲-۱- ارزیابی خوشه‌بندی MDDBSCAN

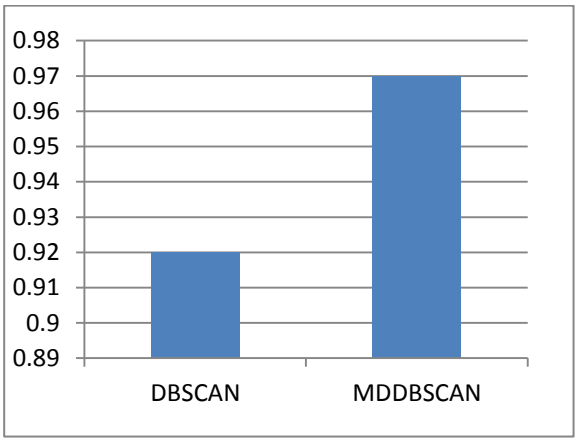
برای این‌که ارزیابی MDDBSCAN روی این دادگان را داشته باشیم و میزان بهبود نتایج آن‌را نسبت به الگوریتم DBSCAN بسنجیم، در یک آزمایش از ۸۰ ٪ داده‌ها (یک‌بار به شکل تصادفی و بار دیگر به شکل ترتیبی) به‌عنوان داده آموزشی استفاده کردیم و در دادگان تست، دادگان بدون برچسب را برحسب رأی اکثریت خوشه برچسب زدیم (نرمال و حمله). سپس نتایج به‌دست‌آمده را با نتایج به‌دست‌آمده از الگوریتم DBSCAN روی همین دادگان با minpts=4 و Eps=0.02 مقایسه کردیم. برای این مقایسه از سه معیار Fowlkes.Mallows و Jaccard و درصد خطا استفاده کرده‌ایم.

$$\text{Fowlkes.Mallows\_index}=\sqrt{\frac{TP}{TP+FP}.\frac{TP}{TP+FN}} \qquad (۱۰)$$

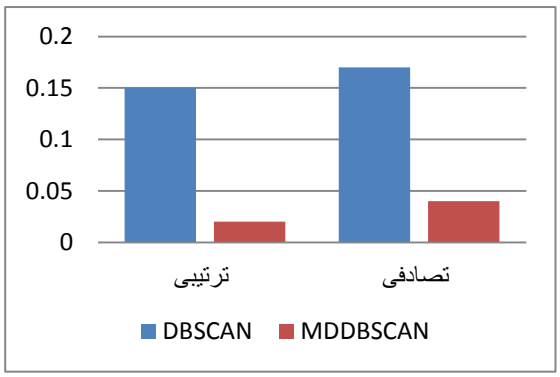
جدول (۱): نتایج ارزیابی خوشه‌بندی با معیار Fowlkes Mallow

دادگان/الگوریتم	KDD99 ٪ ۸۰- ٪ ۲۰ آموزشی- تست (تصادفی)	KDD99 ٪ ۸۰- ٪ ۲۰ آموزشی- تست (ترتیبی)
DBSCAN	۰/۸۷	۰/۹۴
MDDBSCAN	۰/۹۱	۰/۹۸

$$\text{Jaccard\_Index}=\frac{|A\cap B|}{|A\cup B|}=\frac{TP}{TP+FP+FN} \qquad (۱۱)$$



شکل (۶): نتایج ارزیابی خوشه‌بندی براساس معیار میانگین Jaccard



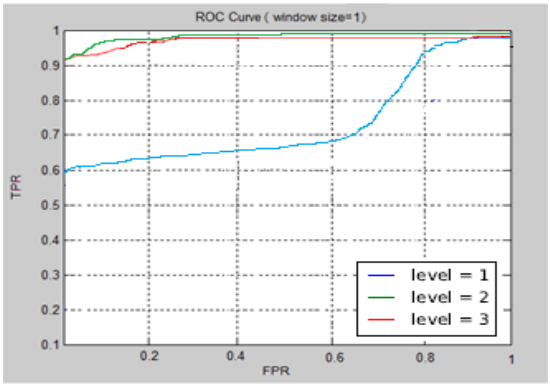
شکل (۷): نتایج ارزیابی خوشه‌بندی با معیار درصد خطا

۴-۲-۲- ارزیابی الگوریتم

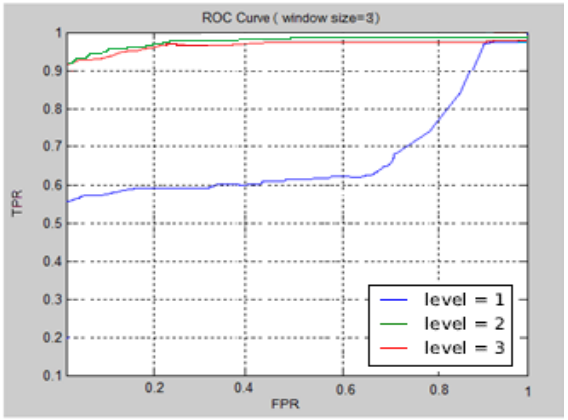
بعد از انجام خوشه‌بندی با هربار کشف لغات در سطوح یک تا سه، الگوریتم ارائه‌شده برای تشخیص سریع حملات، اجرا شد و برای هر لغت کشف‌شده، تشخیص به‌صورت حمله یا نرمال صورت گرفت. در الگوریتم ارائه‌شده به متغیر آستانه حمله و نرمال مقادیر مختلفی دادیم و درنهایت، نتایج کلی الگوریتم را در نمودار ROC، با هر دو آزمایش با پنجره‌های پیش‌بینی ۱ و ۳ رسم کردیم.

شکل (۸) نتایج آزمایش با استفاده از پنجره پیش‌بینی با اندازه ۱ است. همان‌طورکه در شکل دیده می‌شود نتایجی که از لغات سطوح دو و سه به‌دست‌آمده تفاوت زیادی با نتایجی که از سطوح اول به‌دست‌آمده دارد.

شکل (۹) فوق نتایج آزمایش با استفاده از پنجره پیش‌بینی با اندازه ۳ است. همان‌طورکه انتظار می‌رود با افزایش اندازه پنجره پیش‌بینی دقت نتیجه افت می‌کند. زیرا با افزایش اندازه پنجره پیش‌بینی، ارتباط معنادار بین سازنده‌ها و رفتار شبکه کمتر خواهد شد. در واقع، افزایش اندازه پنجره منجر به تشخیص با فاصله بیشتر اما با دقت کمتر می‌شود.



شکل (۸): نتایج الگوریتم و مقایسه نتایج به‌دست‌آمده از لغات سطوح ۱، ۲ و ۳



شکل (۹): نتایج افزایش اندازه پنجره پیش‌بینی تا اندازه ۳

۵- نتیجه‌گیری

هدف اصلی این مقاله تشخیص بهتر و سریع‌تر حملات سایبری چندمرحله‌ای است. بدین منظور از مدل‌سازی ساخت یافته رفتاری با الهام‌گرفتن از پردازش زبان طبیعی استفاده شده است. در روش پیشنهادی علاوه‌بر استخراج روابط میان دنباله‌های دریافتی از شبکه، از روش عمقی که در رفع ابهام در زبان طبیعی وجود دارد، الهام گرفته‌ایم. ما برای انجام کار خود از دادگان KDD99 که بیشترین دادگان مورد استفاده در این حوزه می‌باشد، استفاده کرده‌ایم. به‌علت داشتن ویژگی‌های پیوسته و گسسته در دادگان KDD99، ما برای خوشه‌بندی اولیه از معیار شباهت Gower استفاده کردیم که هر دو نوع ویژگی را در رابطه خود در برمی‌گیرد. خروجی این خوشه‌بندی لغات ابتدایی یا سطح اول در دنباله زبان شبکه را تشکیل می‌دهد. با تناظری که بین زبان طبیعی و زبان شبکه تعریف کردیم می‌توان با ترکیب لغات سطوح پایین‌تر به لغات سطوح بالاتر دست پیدا کرد. برای

ترکیب لغات از فرمول آماری مربعات کای دو استفاده کردیم که با داشتن یک حد آستانه برای تصمیم‌گیری ترکیب یا عدم ترکیب دو لغت از سطح پایین‌تر به ما کمک کرد. در ادامه کار، با خوشه‌بندی سلسله مراتبی لینک کامل این ترکیبات را به لغات مشابه نیز تعمیم دادیم تا روابط و لغات جدید بیشتری تولید کنیم. درنهایت، با به‌دست‌آوردن لغات سطوح مختلف و روابط بین آن‌ها با استفاده از الگوریتم CYK توانستیم لغات را در دنباله آموزش پیدا کرده و به همراه پنجره رویدادهای بعد از آن در یک جدول ذخیره کنیم. برای تشخیص زودهنگام حملات، تصمیم‌گیری برای هر لغت به‌منظور قرارگرفتن در دسته حمله یا دسته نرمال صورت گرفت. همان‌طورکه در نمودارهای ROC مشاهده شد، نتایجی که لغات سطح دوم و سوم در تشخیص حملات داشتند نسبت به لغات سطح یک تفاوت و بهبود قابل توجهی داشت اما لغات سطح سوم نسبت به سطح دوم در به‌دست‌آوردن نتایج کارایی بهتری نداشته‌اند. علت این موضوع می‌تواند تنوع و گوناگونی بیشتر لغات در سطوح بالاتر باشد. این موضوع در زبان طبیعی نیز به چشم می‌خورد که هرچه سطح لغات بالاتر باشد، گوناگونی و پراکندگی در آن‌ها بیشتر خواهد بود. این موضوع می‌تواند در کارهای آینده مورد بررسی و تحقیق بیشتری قرار بگیرد.

۶- منابع

[1] R. Kremmerer and G. Vigna, "Intrusion Detection: A Brief History and Overview," 2002.

[2] D. Marc Dacier and A. wespi, "Towards a taxonomy of intrusion-detection systems," 1999.

[3] A. Sperotto G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An Overview of IP Flow-Based Intrusion Detection," pp. 343-356, 2010.

[4] W. Lee, S. J. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems," ACM Transaction on Information and System Security 3, pp. 227-261, 2000.

[5] X. Yan and J. Ying Zhang, "Early Detection of Cyber Security Threats using Structured Behavior Modeling," ACM Transaction on Information and System Security, vol. V, 2013.

[6] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," New York : s. n, conference on Applications, technologies, architectures, and protocols, pp. 219-230, 2004.

[7] M. Saniee Abadeh, J. Habibi, and C. Lucas," Intrusion detection using a fuzzy genetics-based learning algorithm," pp. 414-428, 2007.

[8] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," pp. 6225–6232, 2010.

[9] A. J. Rashidi, K. Dadashtabar Ahmadi, and F. Samsami Khodadad, "Projection of Multistage Cyber Attack Based on Belief Model and Fuzzy Inference," Journal of electronical & cyber defence, vol. 3, no. 2, 2015. (In Persian)

[10] B. Thomas Adler, L. de Alfaro, S. M. Mola-Velasco, P. Rosso, and A. G. West, "Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Feature," International Conference on Intelligent Text Processing and Computational Linguistics, pp. 277-288, 2011.

[11] N.-A. Le-Khac, M.-T. Kechadi, and M. Banerveld, "Performance Evaluation of a Natural Language Processing approach applied in White Collar crime investigation," School of Computer Science & Informatics, University College Dublin Belfield, Dublin, Ireland, 2014.

[12] H.-K. Peng, P. Wu, J. Zhu, and J. Ying Zhang, "Helix:Unsupervised Grammar Induction for Structured Activity Recognition," 11th IEEE International Conference on Data Mining. pp. 1195-1199, 2011.

[13] K. Rieck and P. Laskov, "Detecting unknown network attacks using language models," Third international conference on Detection of Intrusions, pp. 74-90, 2006.

[14] Sunoallah, Wesam Ashour and Saad, "Multi Density DBSCAN," International Conference on Intelligent Data Engineering and Automated Learning, pp. 446-453, 2011.

[15] A. Özgür and H. Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," Ankara : s.n., Apr. 2016.

[16] J. C. Gower, "A general coefficient of similarity and some of its properties," pp. 857–871, 1971.

[17] W. G. Cochran, "The X2 test of goodness of fit," Annals of Mathematical Statistics, vol. 25 , pp. 315–345, 1952.

[18] Peterson, A. C. Lin, and L. Gilbert, "Activity Pattern Discovery from Network Captures," IEEE Symposium on Security and Privacy Workshop, 2016.

Journal of Electronical & Cyber Defence
Vol. 6, No. 3, 2018, Serial No. 23

Detection of advanced Cyber Attacks, Using Behavior Modeling Based on Natural Language Processing

K. Dadashtabar Ahmadi*, M. Kheirkhah, A. J. Rashidi

*Malek-Ashtar University of Technology

(Received: 01/12/2017, Accepted: 03/02/2018)

ABSTRACT

The complex and persistent attacks of network have been made up of numerous hidden stages. One of the reasons for the ineffectiveness of intrusion detection systems against these attacks is the use of a defense mechanism based on low-level network traffic analysis, in which the hidden relationships between alerts are not addressed. Our assumption is that there is a hidden structural information in traffic data, and we want to define rules in network traffic similar to linguistic rules and use it to describe the patterns of malicious network activity. In this way, the discovery of misuse and anomalous patterns can be well treated as the problem of learning syntactic structures and semantic fragments of the "network language". In this paper, for the first time in cybersecurity, a new clustering is used named as the clustering of MD_DBSCAN; one of the most advanced types of DBSCAN clustering. In addition, a greedy algorithm inspired by the induction of grammar in natural language processing has been used to recgnize high-level activities and define the relations between activities in different levels, by integrating low-level activities. In the recognition section of high-level activities of the proposed algorithm, for the first time, similarity edition criterion in hierarchical clustering has been added to the existing criteria in the base algorithm. According to ROC curves the results show that the accuracy of detection in higher-level activities are about 30% higher than low-level activities. Also by choosing the best setting for threshold parameters in attack detection algorithms, we had the highest F1 score in different levels from 1 to 3: 72.3 , 96.2, 96.4. which means that in general we have had the improvement of about 0.2 compared to the base algorithm.

Keywords: Behavior, Advanced Persistent Threats,Cyber Attacks, Data Fusion, Natural Language Processing

* Corresponding Author Email: Dadashtabar@mut.ac.ir