BigFlow: Real-time and Reliable Anomaly based Intrusion Detection for High-Speed Networks

Future Generation Computer Systems

2019

73 citations

Keywords: datastreaming

روشهای فعلی تشخیص برای داده‌های حجیم، بسته ها را ضبط کرده و آنها را به یک فایل سیستم دیگر که از نوع HDFS برای آنالیز می فرستند. الگوریتم‌های یادگیری ماشین فعلی در سرعتهای بالا و عملکرد بلادرنگ[1] مشکل دارند روشهای یادگیری ماشین غیرنظارتی[2] برای کشف آنومالی‌ها، به دلیل ضبط بسته‌ها در یک بازه زمانی و سپس تشخیص بی نظمی نیز مشکل حافظه خواهند داشت و همچنین نرخ مثبت کاذب بالایی خواهند داشت. لذا روشهای نظارتی پیشنهاد شدند، که بر اساس یک مدل به دست‌آمده (بر اساس یک پروسه تمرین[3] اغلب طولانی و هزینه بر) ترافیک های بد و نابد را با استفاده از یک الگوریتم classifier دسته بندی می‌کنند. اما با توجه به ماهیت متغیر ترافیک، این مدل را باید مرتباً بازطراحی کرد. در این آزمایش بررسی می‌کنیم که دقت این سامانه‌ها در طول سال به ۲۳درصد کاهش می‌یابد. لذا برای نشان دادن آن به عنوان یک سیستم قابل اتکا باید مدام آموزش و تست شود.

راهکار ما در این مقاله از این روش استفاده می‌کند که آیا خروجی classifier که همان تعیین مهاجم بودن یا سالم بودن ترافیک است، باید قبول شود(rejecting low confidence classifications)؟(با کمک event class(normal or attack) probability(confidence)). مثلاً اگر رویدادی مهاجم طبقه بندی شده بود، تنها در صورتی می‌تواند قبول شود که معیار اطمینان آن بالای ۹۰٪ باشد. اگریکی قبول نشد، ادمین می فهمد که رفتار شبکه تغییر پیداکرده است. تغییر در رفتار شبکه می‌تواند به خاطر ظهور سرویس‌های جدید باشد. همچنین راهکارمان یک stream learning می‌باشد که ترافیک را در لحظه بررسی خواهد کرد و به روزرسانی افزایشی مدل بر اساس نمونه‌های ردشده را خواهیم داشت. این مدلهای افزایشی باعث افزایش دقت و کاهش زمان یادگیری(به دلیل دور نیانداختن مدل قبلی) در شبکه های پهن باند خواهند شد. اما مشکلشان این است که نظارتی هستند یعنی نیاز به داده‌های از قبل طبقه‌بندی‌شده[4] خواهند داشت. روش‌ما میخواهد مداخله انسان را کاهش دهد و همچنین میزان دیتایی که باید ضبط شود را کم کند.

نوآوری‌های ما در این مقاله:

---

[1] Real-time
[2] Unsupervised
[3] training
[4] Classified

- تولید اولین دیتاست برای بررسی IDS در مدت زمان طولانی یکسال که شامل رکوردهای برچسب گذاری شده(نرمال یا مهاجم) می‌باشد که هربسته ۱۵۸ تا feature خواهد داشت (براساس دیتاست MAWIFlow می‌باشد). دیتاست‌های که برای بررسی سامانه‌های تشخیص نفوذ استفاده می‌شوند باید دارای ویژگی‌های زیر باشند:realism، validity: با این که پکت‌ها پیلودشان حذف شده است و داده‌های حساس موجود در سرایندها حذف شده‌اند اما همچنان بازسازی جریان‌ها ممکن است چون‌که feature ها دستکاری نشده اند،prior labeling، high variablity، and reproducibiltiy public availibilty
- تست الگوریتم های classifier مختلف با این دیتاست (الگوریتم‌های decision tree, random forest, gradient boosting,hybrid) . برای هر کدام نیز دو گونه آپدیت (آموزش مدل) مدل درنظر گرفتیم: بدون آپدیت و آپدیت هفتگی. از Apache Spark برای پیاده‌سازی و ارزیابی این الگوریتم‌ها استفاده کرده‌ایم.
- طراحی bigflow که آن را در یک بازه زمان یکساله بررسی کرده و دقت بالایی و میزان استفاده کم از منابع و همچنین زمان کم برای آموزش دارا می‌باشد.

هنگام استفاده از روش‌های یادگیری ماشین، رفتار ترافیک بر اساس یک سری ویژگی‌هایی که از آن استخراج می‌شود، نمایش داده می شود:

Table 1
Network-level feature set used in the experiments throughout this work [18].

Type	Grouping	Features
Host-based	Host to All	Number of Packets, Number of Bytes, Average Packet Size, Percentage of Packets (PSH Flag), Percentage of Packets (SYN and FIN Flags), Percentage of Packets (FIN Flag), Percentage of Packets (SYN Flag), Percentage of Packets (ACK Flag), Percentage of Packets (RST Flag), Percentage of Packets (ICMP Redirect Flag), Percentage of Packets (ICMP Redirect Flag), Percentage of Packets (ICMP Time Exceeded Flag), Percentage of Packets (ICMP Unreachable Flag), Percentage of Packets (ICMP Other Types Flag), Average Packet Size, Throughput in Bytes, Protocol
Flow-based	Source to Destination Destination to Source Both	Number of Packets, Number of Bytes, Average Packet Size, Percentage of Packets (PSH Flag), Percentage of Packets (SYN and FIN Flags), Percentage of Packets (FIN Flag), Percentage of Packets (SYN Flag), Percentage of Packets (ACK Flag), Percentage of Packets (RST Flag), Percentage of Packets (ICMP Redirect Flag), Percentage of Packets (ICMP Redirect Flag), Percentage of Packets (ICMP Time Exceeded Flag), Percentage of Packets (ICMP Unreachable Flag), Percentage of Packets (ICMP Other Types Flag), Throughput in Bytes

BigFlow بر پایه پلتفرم‌های پردازش استریمی[5] بنیان شده است که اینها اطلاعاتی را که دریافت میکنند توسط یک سری PE(Processing Element) پردازش می کنند. ارتباطات و نحوه پیغام دهی بین PE ها سه حالت دارد: shuffle : که با احتمال یکسان به یک PE دیگر می فرستد، keyed: که پیغام ها را مثلن بر اساس آدرس IP گروه بندی کرد و آن را به PE مربوطه ارسال می‌کند، broadcast: که به تمام PE های همنوع ارسال می‌کند. بلادرنگی زمانی برقرار می شود که هر PE حافظه محدود داشته باشد و به صورت موازی چندین مورد از آنها باهم پردازش داشته باشند

روش کلی کار BigFlow به دو قسمت تقسیم می‌شود:(*نکته! event (رویداد): یکای چیزی که باید آنالیز شود و می‌تواند از نوع packet header یا netflow record باشد*)

[5] Apache Flink & Apache Storm

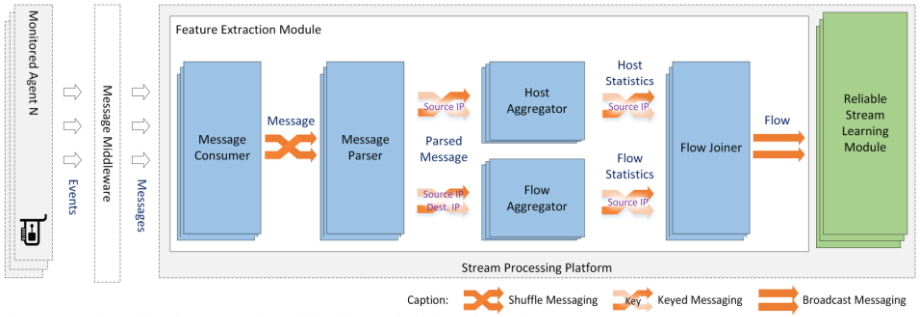

Figure 2 – *BigFlow* real-time feature extraction module architecture for high-speed networks.

Feature extraction: یک فریمورک پردازش استریم میباشد. آمار جریان به صورت بلادرنگ که feature vector نامیده می‌شود (در یادگیری ماشین event یا instance گویند) را به دست می‌آورد و شامل اطلاعات ترافیک ردوبدل شده بین دو هاست در یک بازه زمانی[6] می‌باشد.روش‌های قدیمی استخراج اطلاعات برای classifeier در پهن‌باند کاربرد ندارند. Bigflow ۱۵۸ ویژگی که در سطح میزبان (مثلاً اطلاعات دریافتی و یا ارسالی یک هاست مشخص) و جریان(بین دو هاست مثلاً میانگین حجم بسته‌های تبادل شده بین دو هاست) می‌باشند، استخراج می‌کند. این بخش از اجزای زیر تشکیل شده است:

Monitored Agent: هاست‌ها، سوییچ‌ها، روترها. مثلاً سوییچ ها هدرهای بسته‌ای ارسال می‌کنند و روترها NetFlow record ارسال می‌کنند. رویدادها را از طریق Message Middleware به فریمورک ارسال می‌کنند.

MessageMiddleWare: همه رویدادها از طریق اینترفیس این می‌روند.

MessageConsumer: دریافت تمامی انواع رویدادهای موجود و سپس ارسال آنها به صورت shuffle توسط PE های Stream Processing به ماژول اصلی.

MessageParser: بر اساس نوع رویداد دریافتی یکسری اطلاعات از آن استخراج می‌کند: event source/field/type(netflow record or network packet). روی PE های مختلفی بالانس شده است.

Aggregator : از دو بخش اصلی تشکیل شده‌است. کار اصلی استخراج ویژگی را این دو انجام میدهند. هر دو از طریق keyed پیغام ها رو دریافت می‌کنند. بر اساس کلیدی که به دست‌آمده ، به PE مشخصی ارسال می‌کنند(هر کدام از این بخشها از یکسری PE تشکیل شده اند):

- HostAggregator: کلید آن از هش آدرس آی‌پی مبدا به دست می‌آید
- FlowAggregator: کلید آن از طریق XOR بر روی آدرس مبدا و مقصد به دست می‌آید.

هر کدام از این‌ها یک قسمتی از مقادیر هش را مسیول می‌باشند.

BigFlow برای این که مقادیر رویدادهای مربوط به یک گروه را محاسبه کند، آنها را به یک بازه زمینی تقسیم می‌کند که ماژولهای Tumbling Window نامیده می‌شوند. هر کدام از این ماژول‌ها مقادیر ویژگی‌ها را برای یک بازه‌زمانی خاصی و رویداد دریافتی

[6] Time interval

آپدیت یا ذخیره می‌کنند زمانی که Tumbling Window منقضی شد،مقادیر ویژگی‌ها با فرمت آمار هاست یا جریان به ماژول بعدی ارسال می‌کند. خاصیت این ماژول این است که ابتدا مطمین می‌شود که تمامی جریان‌ها منقضی می‌شوند

FlowJoiner: این ماژولها تمامی ویژگیهای انواع مختلف رویدادها را دریافت کرده و آنهارا در استریم واحد جریان پیوند می‌کنند. باید توجه داشت که یک هاست واحد ممکن است شامل چندین ویژگی آماری جریانی دریک Tumbling Window داشته باشد در حالی که تنها یک ویژگی آماری هاستی داشته باشد.(مثلاً هاستی که از سرویس‌های چندین هاست دیگر استفاده می‌کند). پس وظیفه این ماژول این است که همه اینها را ذخیره کند و در آینده به هم متصل کند.

Reliable stream learning: بر اساس ورودی feature vector از ماژول قبلی آن را به مهاجم یا طبیعی classify می‌کند. Streamlearning classifier به همراه یک ماژول verifier می‌باشد. زمانی که خروجی classifier تایید نشد، اون رویداد را ذخیره می‌کند تا بعداً توسط یک عامل انسانی(که بر اساس اطلاعاتی که از رفتار جدید شبکه به دست می‌آورد مثلاً سرویس‌های جدید و یا سایت‌های CVE) برچسپ گذاری شود. سپس از این نمونه برچسب گذاری شده برای به روزرسانی افزایشی classifier استفاده می‌شود.

این ماژول در ابتدا با یک دیتاست آموزش داده می‌شود و classifier model ساخته می‌شود و مقدار آستانه برای هر کلاس نرمال یا مهاجم تعیین می‌شود.

پیاده سازی: کل طرح ما روی Apache Flink که فریمورک مخصوص پردازش استریم می‌باشد پیاده سازی شده‌است و همچنین مکانیزم Tumbling Window نیز در آن شبیه‌سازی شده است و زمان آن نیز به حالت بهینه ۱۵ ثانیه تنظیم شده است. پیغام رسانی keyed نیز با استفاده از اینترفیس KeySelector آن پیاده سازی شده است.(برای آشنایی با بخش‌های دیگر به مقاله مراجعه شود)

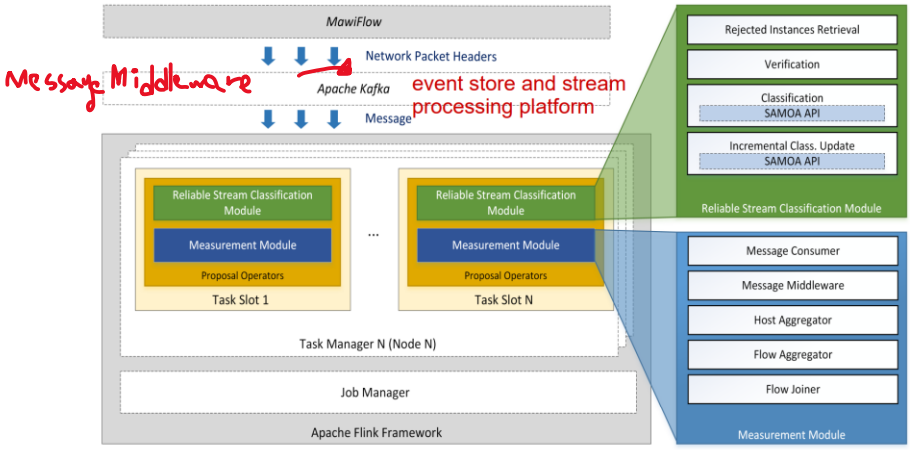

Figure 5 – *BigFlow* architecture.

ارزیابی: در ابتدا ماژول Reliable Stream Classification با معیار دقت در طول زمان، با دیتاست MAWIFlow ارزیابی شده است و سپس کارایی و مقیاس پذیری BigFlow و همچنین هزینه به روزرسانی ماژول stream learning بررسی شده است.

راهکارهای پیشنهادی برای بهبود:

استفاده از اسکچ‌ها برای ذخیره ویژگی‌ها؛

تعیین خودکار اینکه که یک event مهاجم است یا نه؛