بسمه تعالی

معرفی، بررسی، نصب و پیکربندی چارچوب Apache Metron

فهرست مطالب

۱ مقدمه

آمارهای منتشر شده از حملات اخیر در دنیا از جمله گزارش شرکت Yahoo مبنی بر افشای حساب‌های کاربری افراد، آمار گسترده حملات بر روی حساب‌های کاربری در Twitter، همچنین سرعت و ابعاد وسیع حملات اخیر باج‌افزارها و نفوذ به شبکه‌های رایانه‌ای و بسیاری دیگر از حوادث اخیر، مدیران را با این حقیقت روبرو نموده است که تشخیص و کاهش تهدیدات یک موضوع کلیدی برای تیم‌های مرکز عملیات امنیت می‌باشد. امنیت سایبری از چالش‌های عمده سازمان‌ها بوده و موجب نگرانی آن‌ها درباره نفوذ به داده و گزینش روش‌های جدید برای امن‌سازی دارایی‌های آسیب‌پذیر و تحلیل و پاسخ‌گویی به حوادث سایبری به‌منظور مقابله با آن‌ها شده است.

یکی از چالش‌هایی که سازمان‌ها با آن روبرو هستند، افزایش حجم تولید داده‌ها (داده‌های کلان) و گسترش خطرات ناشی از حرکت سازمان‌ها به سمت استفاده از فن‌آوری‌های داده‌های بزرگ جهت ذخیره‌سازی و تحلیل داده برای افزایش بینش و آگاهی تیم‌های پاسخ‌گویی به حوادث سایبری است. همچنین مقدار زیادی از داده‌ها از منابع متعدد مانند حسگرها تولید شده و ردپای مهاجمان می‌تواند همچنان ناشناخته باقی بماند. با وجود آن‌که برخی سازمان‌ها در مسیر بهبود تجربه مشتری با نوآوری در محصولات خود قرار گرفته‌اند اما میزان و حجم بالای داده‌های جریان، آن‌ها را از پیمایش کلیه اطلاعات باز می‌دارد. راه‌حل‌های موجود و سنتی SIEM قادر به مدیریت تولید حجم بالای داده نیستند و نیاز به تحلیل این داده‌ها با کمترین تأخیر ممکن و همچنین افزایش فشار محاسباتی و پردازش داده، سازمان‌ها را به استفاده از نسل جدید SIEM مبتنی بر معماری کلان داده سوق می‌دهد.

در SIEMهای نسل جدید مبتنی بر کلان داده معمولا دو رویکرد مطرح می‌شود:

۱. استفاده از معماری‌ها و فن‌آوری‌های مقیاس‌پذیر و مبتنی بر جامعه[۱] مانند OpenSOC و Apache Metron

[۱] Community-oriented scalable technologies

۲. امکان و استفاده از فن‌آوری‌ها و چارچوب‌هایی که سازمان خود مبادرت به ساخت و ارائه آن‌ها می‌کند، که اصطلاحاً به آنها DIY[۲] گفته می‌شود.

با توجه به انفجار اطلاعات، SOCها نیاز به نمایش بی‌درنگ داده‌ها دارند تا قادر به پردازش و تحلیل آن ها باشند. در این حالت رویکرد مبتنی بر جامعه براساس فن‌آوری‌های مقیاس‌پذیر به سازمان نه تنها قابلیت مقیاس‌پذیری را افزایش می‌دهد بلکه به سازمان در حل مؤثر معضلات مقابله با حوادث سایبری کمک می‌کند. یادگیری ماشین، خودکارسازی و غنی‌سازی بی‌درنگ کلیه داده‌ها بایستی از مؤلفه‌های کلیدی چنین فن‌آوری‌هایی باشند تا باعث ایجاد یک ساختار مقیاس‌پذیر در پاسخ‌گویی به حملات سایبری مدرن امروزی گردند. در این سناریو سازمان‌ها قادر به تحلیل سریع‌تر تهدیدات، ضبط داده با هزینه کمتر و مقابله آشکار با چالش‌های جدید امنیت سایبری در مقیاس وسیع‌تر خواهند بود. رویکرد دیگر DIY است که سازمان خود اقدام به طراحی و تولید یک برنامه برای تحلیل داده‌ها می‌کند. اما استفاده از محصولات مبتنی بر جامعه که با یکدیگر برای مقابله با حملات همکاری می‌کنند همیشه گزینه بهتری است. یکی از این تلاش‌ها توسط بنیاد آزاد نرم‌افزار آپاچی[۳] (ASF) پشتیبانی گردید که بر پایه Apache Metron، چارچوب برنامه کاربردی امنیت سایبری کلان داده ایجاد شد تا امکان مشاهده یکپارچه داده‌های مختلف امنیتی را در مقیاس بزرگ‌تر به‌منظور کمک به SOCها در تشخیص سریع و پاسخ‌گویی به تهدیدات فراهم کند و تلاش دیگر نیز بهره‌گیری از پشته متن باز ELK توسط الستیک می‌باشد.

بنابراین فن‌آوری‌های گذشته به مبارزه با چالش‌های امنیت سایبری امروزی کمکی چندانی نمی‌کنند. از آنجایی که جرایم سایبری از فیشینگ ساده و کلاهبرداری پست‌های الکترونیکی به حلقه‌های اخاذی پیچیده پیشرفت نموده‌اند، بایستی از فن‌آوری‌های متن بازی که رویکرد مبتنی بر جامعه دارند بهره برد. از همه مهم‌تر یک رویکرد مبتنی بر کلان داده از اهمیت اساسی برخوردار است.

---

[۲] Do-it-Yourself

[۳] Apache Software Foundation

۲ معرفی Apache Metron

Apache Metron یک چارچوب کاربردی امنیت سایبری متن باز می‌باشد که توسط Apache برای ارائه چارچوب امنیتی پیشرفته و نسل جدید معرفی گردید تا ریسک‌های امنیتی بی‌درنگ تشخیص داده شود. Aapache Metron تکنولوژی‌های مختلفی از کلان داده را به منظور ارائه ابزاری متمرکز برای پایش و تحلیل امنیتی تجمیع می‌کند و قابلیت‌هایی را برای سازمان‌ها به منظور گردآوری فایل‌های ثبت رویداد، شاخص‌گذاری ضبط کامل بسته، پردازش و ذخیره‌سازی مقدار زیادی از منابع مختلف امنیتی، تحلیل رفتاری پیشرفته و غنی‌سازی جریان داده‌ها، یکپارچگی با منابع هوشمندی تهدید و طبقه‌بندی تهدیدات فراهم می‌آورد. این قابلیت‌ها امکان پردازش بی‌درنگ در سرعت بالا، تشخیص ناهنجاری‌ها و پاسخ‌گویی سریع به آن‌ها را ایجاد می‌کنند. Apache Metron به تیم‌های عملیات امنیت امکان کاهش مؤثر مقدار کلان داده در چارچوب DIY و ابزار علم داده لازم برای تشخیص تهدیدات در زمان واقعی را می‌دهد.

عملکرد کلی Apache Metron در ٤ حوزه زیر قرار می‌گیرد:

۱. سازوکاری برای ضبط، ذخیره‌سازی و نرمال‌سازی هرگونه تله‌متری امنیتی در سرعت‌های بسیار بالا می‌باشد. از آنجا که تله‌متری امنیتی به‌طور مداوم تولید می‌گردد، نیازمند به روشی برای به‌دست‌آوردن داده در سرعت‌های بالا و تحویل آن به واحدهای پردازش متفاوت برای محاسبات و تحلیل‌های پیشرفته است.

۲. پردازش بی‌درنگ و کاربرد غنی‌سازی از جمله اطلاعات جغرافیایی، هوشمندی تهدید و DNS به تله‌متری جمع‌آوری انجام می‌گیرد. استفاده فوری از این اطلاعات به دریافت تله‌متری ورودی، زمینه و آگاهی موقعیتی و همچنین شخص و مکانی را که برای تحقیقات ضروری باشد، فراهم می‌کند.

۳. ذخیره‌سازی کارآمد اطلاعات براساس چگونگی اطلاعات مورد استفاده:

- استخراج فایل‌های ثبت رویداد و تله‌متری ذخیره‌شده انجام می‌گیرد و برای دید اجمالی نسبت به امنیت تحلیل می‌شوند.
- قابلیت استخراج و بازسازی مجدد کامل بسته‌ها به تحلیل‌گر امکان پاسخ‌گویی به سوالاتی از جمله مهاجم حقیقی چه کسی بوده است، چه داده‌ای از بین رفته و داده به کجا ارسال گردیده است، را می‌دهد.
- ذخیره‌سازی درازمدت نه تنها در طول زمان افزایش می‌یابد، بلکه امکان تحلیل‌های پیشرفته از جمله فنون یادگیری ماشین را فراهم می‌کند تا برای ایجاد مدل‌ها بر روی اطلاعات مورد استفاده

قرار گیرد. داده دریافتی می‌تواند در برابر این مدل‌های ذخیره‌شده برای تشخیص پیشرفته ناهنجاری رتبه‌بندی شود.

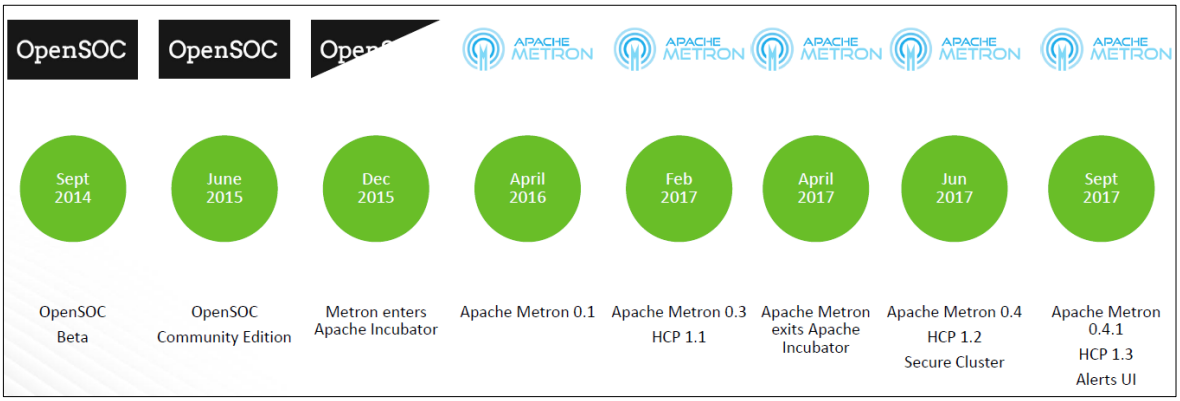
٤. واسط کاربری Metron، خلاصه هشدارها را با هوشمندی تهدید و غنی‌سازی خاص برای آن هشدار بر روی یک صفحه مجزا ارایه می‌دهد. به‌علاوه، قابلیت‌های پیشرفته جست‌وجو و ابزارهای استخراج کامل بسته به تحلیل‌گر برای تحقیقات بدون نیاز به ابزارها ارایه می‌گردد.

کلان داده گزینه‌ای مناسب برای تحلیل‌های امنیتی قدرتمند است. چارچوب Metron تعداد زیادی از عناصر را از اکوسیستم هادوپ جهت فراهم نمودن چارچوبی مقیاس‌پذیر برای تحلیل‌های امنیتی، شامل قابلیت‌هایی مانند ضبط کامل بسته، پردازش جریان، پردازش دسته‌ای، جست‌وجوی بی‌درنگ و گردآوری تله‌متری یکپارچه‌سازی می‌کند.

۲-۱ تاریخچه

همزمان با افزایش چشم‌گیر در فعالیت‌های مخرب و تهدیدات سایبری، از سال ۲۰۰۵ تا ۲۰۰۸ میلادی دوره‌ای وجود داشت که Apache Hadoop در حال توسعه بود و با کمبود نیروی متخصص امنیتی مواجه بوده‌است. پس از سال ۲۰۰۸ میلادی، با پیدایش کلان داده، حجم زیادی از داده جاری در مراکز داده در سراسر جهان وجود داشت.

شرکت سیسکو به دلیل دارا بودن متخصصین در حوزه‌های امنیتی و زیرساخت این کمبود را احساس کرد، چرا که ابزارهای سنتی SIEM دیگر پاسخگوی نیازهای آنان نبوده و در نتیجه OpenSOC متولد شد. بین سپتامبر ۲۰۱۳ تا آوریل ۲۰۱۵ میلادی، James Sirota دانشمند ارشد داده همزمان در سیسکو و تیم Hortonworks، مبادرت به ایجاد SOC مدیریت شده نسل جدید و ساخته شده بر پایه فن‌آوری‌های متن باز کلان داده نمود. در دسامبر ۲۰۱۵ میلادی، OpenSOC ارائه گردید و به‌عنوان پروژه جدید و نوپا و محل رشد در Apache پذیرفته شد و به Apache Metron تغییر نام یافت. در آوریل ۲۰۱٦ میلادی، اولین نسخه رسمی Apache Metron یعنی ۰,۱ توسط جامعه Metron منتشر گردید. تصویر ۲-۱ نمودار تولد Apache Metron را از آغاز تاکنون نشان می‌دهد.

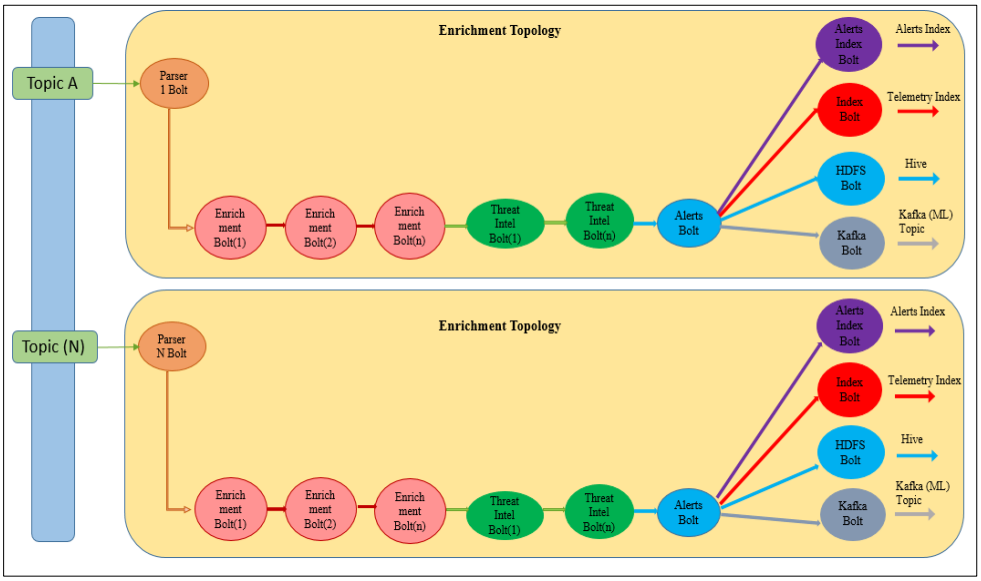


شکل ۲-۱: سیر تحول Apache Metron

۲-۲ تفاوت Apache Metron و OpenSOC

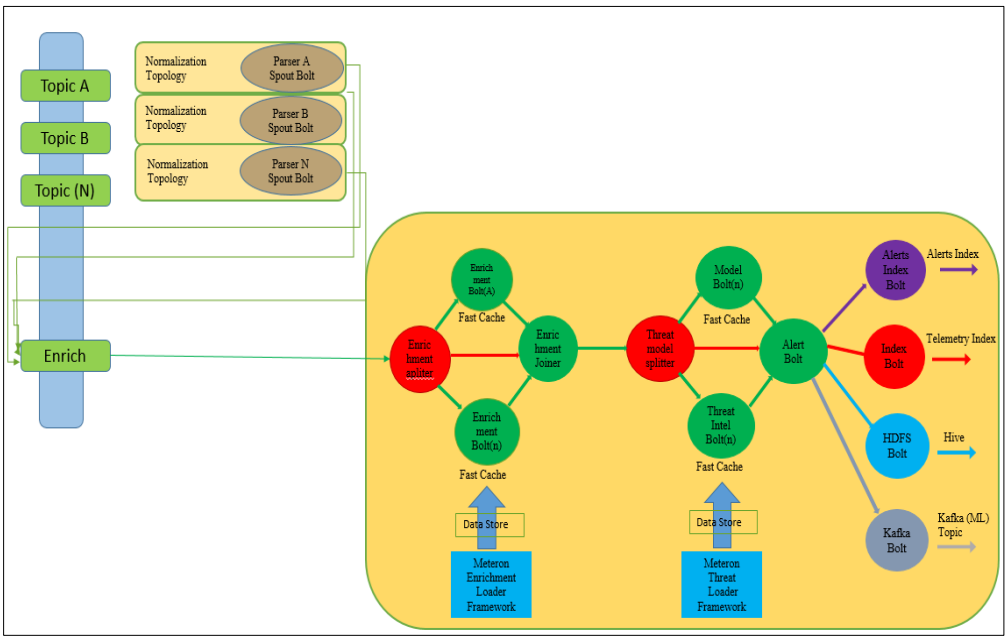
Apache Metron مزایای OpenSOC شامل پردازش سریع رویدادها از منابع مختلف را به ارث می‌برد. یکی از اهداف آن غلبه بر ضعف OpenSOC است. مهمترین چالش‌های معماری OpenOC از دیدگاه متخصصین حوزه‌ی امنیت، عبارتند از:

- **موازی‌بودن کامل، هیچ ثمره‌ای برای آن ندارد.** برای هر topic، توپولوژی غنی‌سازی شامل یک مجموعه از boltها است که به‌صورت سریالی اجرا می‌شوند. این معماری هیچ بهره‌ای از موازی‌بودن که storm می‌توانست فراهم کند، نمی‌برد.
- **مشکل در گسترش و افزودن یک منبع جدید:** باید یک توپولوژی جدید برای کامل‌کردن این نیازمندی ایجاد کند.
- **مشکل در نگهداری:** با افزایش در تعداد منابع داده مقدار زیادی توپولوژی‌های تکراری وجود دارد. بیشتر منطق‌های bolt در توپولوژی غنی‌سازی یکسان است، با این حال، با توجه به معماری OpenSOC، هر topic باید خط کد خود را حفظ کند. در صورتی که هر تغییری برای منطق خاص و مشترک برای عناوین نیاز باشد، باید در بین چندین خط کد تغییر کند که پیچیدگی و هزینه نگهداری را افزایش می‌دهد.
- **عدم وجود آزمون:** هیچ واحد یکپارچه‌ای برای آزمون در داخل OpenSOC وجود ندارد. کیفیت کدها به‌صورت کامل وارسی نشده است. شکل ۲-۲، ساختار سنتی OpenSOC را نشان می‌دهد.



شکل ۲-۲: ساختار سنتی OpenSOC

با معماری جدید Metron، همان‌طور که شکل ۲-۳ نشان می‌دهد، هدف Metron بایگانی بهتر و توسعه‌پذیر، حفاظت و نگهداری بهتر و کارایی و عملکرد قوی‌تر است.



شکل ۲-۳: معماری غنی‌سازی در Apache Metron

- **توسعه‌پذیری بهتر:** Metron قابلیت افزودن تجزیه‌کننده‌های جدید داده را بدون نوشتن کد دارد، که این مسئله را با بهره‌گیری از تجزیه‌کننده چارچوب Grok بهبود داده است. با Grok، می‌تواند الگوهای فایل‌هایی را بیفزاید که جوهره منابع جدید را بیرون کشیده و به‌آسانی منابع توسعه می‌یابند. با تعریف انتزاع توپولوژی نرمال‌سازی که در شکل ۲-۳ نشان داده شده است، Metron ویژگی دریافت منابع

داده را خلاصه کرده و در نتیجه نیازمندی‌ها را به‌منظور پیاده‌سازی توپولوژی منفرد برای هر منبع داده حذف می‌کند.

- **نگهداری و محافظت بهتر:** Metron کلیه توپولوژی‌های Storm را برای استفاده از پیکربندی Flux تبدیل می‌کند (روش اعلانی برای ارتباط توپولوژی‌ها با یکدیگر). Metron یک واحد و چارچوب آزمون یکپارچه را با کاهش هزینه نگهداری، بهبود کیفیت کدها و همچنین اشتباهات در زمان اجرا به‌وجود می‌آورد. با پیشرفت اطلاعات پیکربندی به سمت Zookeeper، Metron نیازمندی‌ها را به منظور توقف توپولوژی‌ها در بیشتر موارد حذف می‌کند. در حالت تغییر در پیکربندی توپولوژی‌ها هیچ نیازی به شروع مجدد توپولوژی بی‌تأثیر در محیط تولید نیست.
- **کارایی بهتر:** همان‌طور که در تصویر شماره ۲-۳ می‌بینید، با تعریف الگوی متصل و تقسیم‌کننده Storm، Metron توانایی پردازش چند bolt غنی‌سازی و bolt مقابل تهدیدات هوشمند را به‌صورت موازی دارد، که انتظار می‌رود کارایی را بهبود دهد. همچنین با معرفی سازوکار حافظه، Metron قادر به نگهداری بیشتر اطلاعات متقابل مورد استفاده برای غنی‌سازی و تهدید هوشمند در حافظه است، که در عوض هزینه دست‌یابی به داده را از HBase و MySQL کاهش می‌دهد. معماری جدیدی که Metron به تصویب رسانده هزینه افزودن منابع جدید را کاهش می‌دهد، در حالی‌که از کارایی بالای OpenSOC نیز پشتیبانی می‌کند. با حمایت از جامعه متن باز، Metron به‌سرعت در حال تکامل بوده و ویژگی‌های بیشتری را می‌افزاید.

۲-۳ قابلیت‌های کلیدی

همان‌گونه که در تصویر ۲-٤ مشاهده می‌کنید چارچوب Apache Metron، چهار قابلیت کلیدی را فراهم می‌نماید:

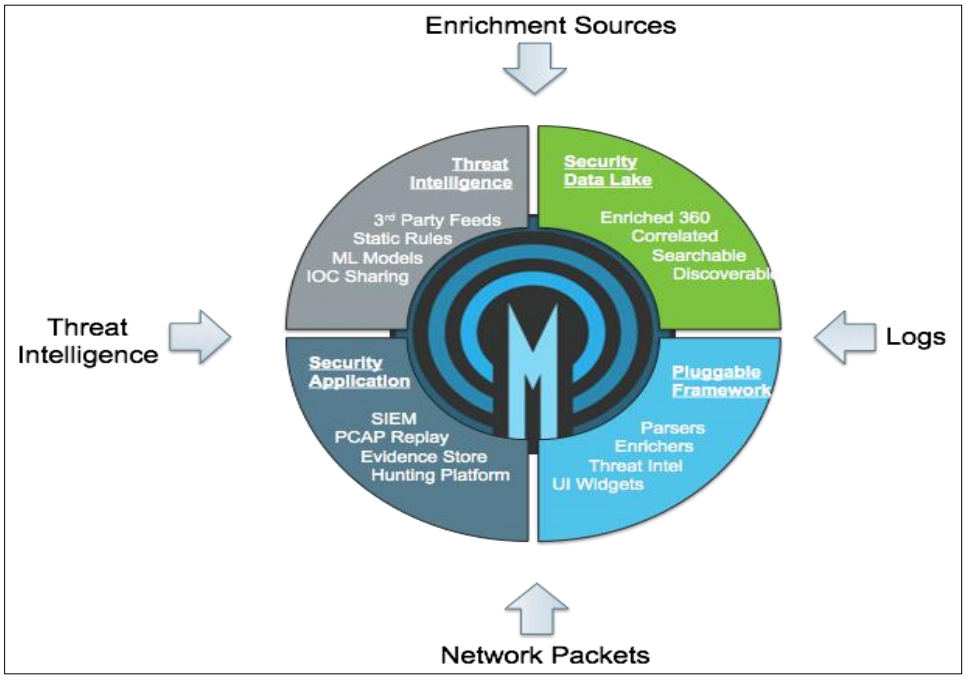
۱. **منبع داده امنیتی:** بستر Apache Metron روشی به‌صرفه به‌منظور ذخیره‌سازی داده تله‌متری برای دوره‌های طولانی‌مدت فراهم می‌کند. این منبع داده، بدنه داده لازم برای انجام تحلیل مهندسی را فراهم می‌آورد و سازوکاری را برای جست‌وجو و پرس‌وجو برای تحلیل‌های عملیاتی ارائه می‌دهد.

۲. **چارچوب قابل تنظیم:** این بستر نه تنها مجموعه‌ای از تجزیه‌کننده‌ها را برای منابع داده امنیتی رایج از جمله pcap، netflow، bro، snort، fireye، sourcefire فراهم می‌کند بلکه چارچوبی قابل تنظیم برای افزودن تجزیه‌کننده‌های سفارشی برای منابع داده جدید و سرویس‌های غنی‌سازی جدید را فراهم

می‌نماید تا اطلاعات زمینه‌ای بیشتر به داده جاری خام، افزونه‌های قابل تنظیم برای منابع هوشمندی تهدید، و توانایی سفارشی‌سازی داشبورهای امنیتی را مهیا کند.

۳. **برنامه کاربردی امنیتی:** قابلیت‌های یک SIEM استاندارد از جمله هشداردهی، چارچوب هوشمندی تهدید، عامل‌ها برای دریافت منابع داده را فراهم می‌آورد اما امکانات بازیابی بسته، ذخیره‌سازی رویداد را دارد که عموماً توسط تحلیل‌گران SOC انجام می‌گیرند.

٤. **چارچوب هوشمندی تهدید:** Metron فن‌آوری‌های دفاعی نسل جدید را فراهم می‌کند که شامل استفاده از یک کلاس از تشخیص ناهنجاری و الگوریتم‌های یادگیری ماشین می‌شود که می‌توانند به صورت بی‌درنگ همانند وقایع در حال جریان انجام گیرند.



شکل ۲-٤: قابلیت‌های کلیدی Apache Metron

موضوعات وظیفه‌ای اصلی عبارتند از:

- **چارچوب:** یک‌سری وظایف برای مقاوم نمودن چارچوب به‌منظور کارآیی، مقیاس‌پذیری، قابلیت توسعه و نگهداری انجام می‌شوند.
- **جمع‌آوری داده:** منابع داده طوری تنظیم می‌شود که Metron قابلیت‌هایی را برای streaming، دریافت و تجزیه در چارچوب فراهم می‌کند.

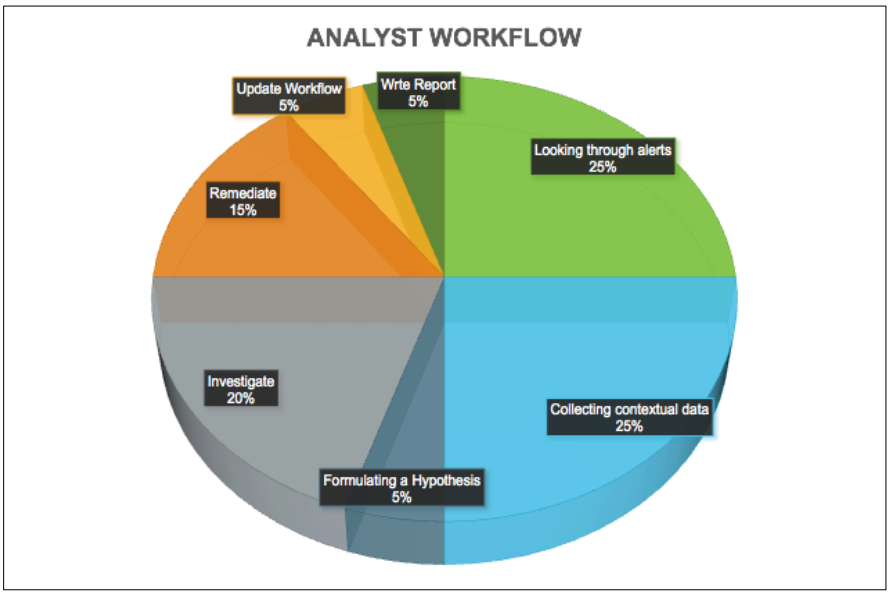
- **پردازش داده:** مجموعه‌ای از توپولوژی‌های Storm جهت انجام فعالیت‌های مختلف به‌صورت بی‌درنگ شامل نرمال‌سازی داده تله‌متری، غنی‌سازی، مرجع متقابل با منابع هوشمندی تهدید، هشداردهی، شاخص‌گذاری و ادامه دادن به ذخیره‌سازی تاریخی فراهم می‌کند.
- **واسط کاربری (UI):** پرتال، داشبورد و واسط‌های کاربر برای اشخاص مختلف قابل تنظیم و شخصی سازی می‌باشد.

۱. مزایا و معایب Metron

مزایای Apache Metron از دو جهت قابل بررسی است:

- از نظر تحلیل‌گر و محقق
- از نظر متخصص داده

۲-۳-۱ تحلیل‌گر و محقق



شکل ۲-۵: مراحل جریان کاری تحلیل‌گر

نمودار ۲-۵ مراحل کلیدی را در جریان کاری محقق یا تحلیل‌گر نشان می‌دهد. برای مراحل خاص در جریان کاری، Apache Metron قابلیت‌های کلیدی را فراهم می‌کند که در ابزارهای امنیتی سنتی یافت نمی‌شوند:

۱. جست‌وجو از طریق هشدارها

a. **کنسول متمرکز هشدارها:** دارای یک داشبورد متمرکز برای هشدارها و رویدادهای تله‌متری مرتبط با هشدارها در کلیه منابع داده امنیتی در سازمان می‌باشد که یک ویژگی قوی درداخل Metron است که از پرش تحلیل‌گر از یک کنسول به دیگری جلوگیری می‌نماید.

b. **هشدارهای Metron:** چشم‌انداز درازمدت Metron فراهم نمودن مجموعه‌ای از مدل‌ها و بسته‌ها شامل Alerts Relevancy Engine و Meta-Alerts می‌شوند. Meta-Alertها توسط مدل‌های تحلیل یا گروه‌بندی تولید می‌شوند و سازوکاری را برای محافظت از کاربر نهایی در برابر ۱۰۰۰ نوع هشدار فراهم می‌سازند.

c. **هشدارهای نشانه‌گذاری شده با داده هوشمندی تهدید:** مشاهده هشدارهای نشانه‌گذاری شده با هوشمندی تهدید از منابع شخص ثالث به تحلیل‌گر امکان رمزگشایی سریع‌تر را می‌دهد که این هشدارها قانونی یا مثبت کاذب هستند.

۲. **جمع‌آوری داده‌های متنی**

a. **پیام‌های کاملاً غنی‌شده:** تحلیل‌گر زمان زیادی را برای غنی‌سازی دستی هشدارهای خام یا رویدادها می‌گذراند، در حالی‌که Metron با پیام‌های کاملاً غنی‌شده‌ای کار می‌کند.

b. **رابط کاربری شفاف UI:** رابط کاربری شفاف که نه تنها دارای کلیه هشدارها از تمام منابع داده امنیتی است، بلکه دیدگاه یکسانی را از داده غنی‌شده ارائه می‌دهد.

c. **جست‌وجوی بی‌درنگ متمرکز:** کلیه هشدارها و رویدادهای تله‌متری به‌طور بی‌درنگ شاخص‌گذاری می‌شوند، زیرا تحلیل‌گر دسترسی فوری به جست‌وجوی کلیه رویدادها دارد.

d. **کلیه وقایع در یک مکان متمرکز:** کلیه وقایع با غنی‌سازی‌ها و نشانه‌ها در یک منبع واحد ذخیره می‌شوند.

۳. **تحقیق**

a. **دسترسی دانه‌ریزی‌شده به PCAP:** پس از شناسایی یک تهدید قانونی، محققان پیشرفته SOC خواهان توانایی برای بارگذاری داده بسته خامی هستند که موجب تولید هشدار شدند.

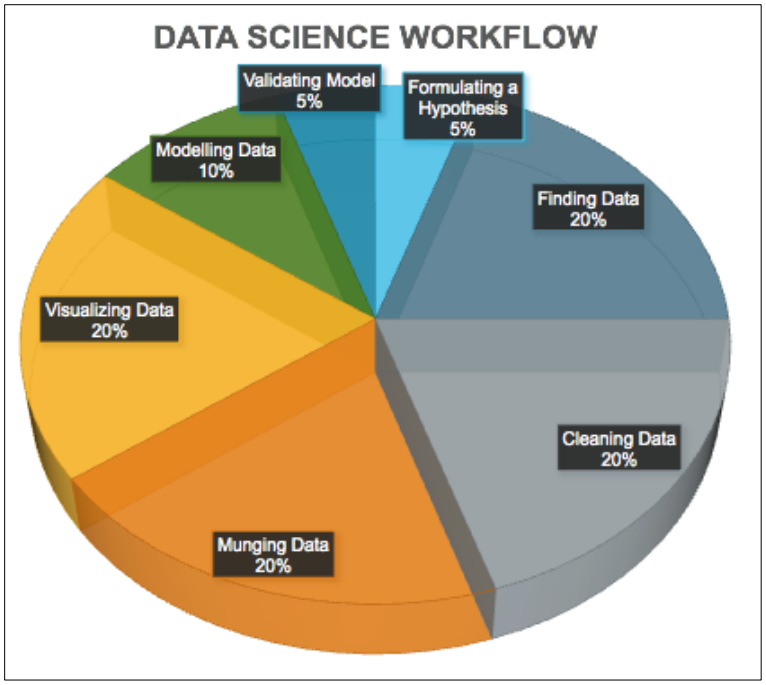
b. بازسازی PCAPهای قدیمی در مقابل امضاهای جدید: Metron امکان پیکربندی برای ذخیره‌سازی داده PCAP خام در هادوپ را دارد. کالبد داده PCAP می‌تواند پس از بازسازی مجدد به‌منظور آزمون مدل‌های جدید تحلیل و امضاهای جدید به‌کار رود.

c. پیام‌های خام به‌عنوان منبع معتبر به‌کار می‌روند.

d. دارایی‌ها و هویت کاربر به‌عنوان منابع غنی‌سازی استفاده می‌شوند.

توجه داشته باشید که سه مرحله بالا در جریان کاری تحلیل‌گر تقریباً ۷۰٪ از زمان را به خود اختصاص می‌دهند. Metron به‌طور قابل ملاحظه‌ای زمان جریان کاری صرف شده تحلیل‌گر را کاهش می‌دهد زیرا هر آن‌چه که تحلیل‌گر نیاز به دانستن دارد در یک مکان واحد متمرکز شده است.

۲-۳-۲ متخصص داده



شکل ۲-۶:مراحل جریان کاری علم داده

نمودار بالا مراحل کلیدی در جریان کاری داده‌ها را نشان می‌دهد. برای مراحل خاص در جریان کاری، Apache Metron، قابلیت‌های کلیدی را فراهم می‌کند که در ابزارهای امنیتی سنتی نبوده و یا به‌سختی یافت می‌شوند:

۱. **پیداکردن داده**

a. **کلیه داده‌ها در یک مکان قرار دارند:** یکی از بزرگ‌ترین چالش‌هایی که متخصصین علم تحلیل داده با آن روبرو می‌شوند پیدا کردن داده لازم برای تمرین و ارزیابی مدل‌های رتبه‌بندی است. Metron یک منبع مجزا را در مکانی که داده تله‌متری در سازمان ذخیره شده است فراهم می‌کند.

b. **داده‌ها در معرض انواع مختلف APIها:** مخزن امنیتی Metron موتورهای مختلفی را برای دسترسی و کار با داده شامل SQL، زبان‌های اسکریپت‌نویسی، in-memory، جاوا و REST APIها و غیره را فراهم می‌کند.

c. **سیاست‌های کنترل دسترسی استاندارد:** همه داده‌های ذخیره شده در مخازن امنیتی Metron با Apache Ranger از طریق سیاست‌های دسترسی در سطح سیستم‌فایل از جمله HDFS و در سطح موتور پردازش از جمله Spark، Hive، Solr و غیره امن‌سازی می‌شود.

۲. **پاک‌سازی داده**

a. **Metron رویدادهای تله‌متری را نرمال‌سازی می‌کند:** Metron کلیه داده تله‌متری را به صورت حداقل یک ساختار JSON با ۷ ستون تبدیل می‌کند که به متخصصین علم داده امکان سهل یافتن و همبسته‌سازی داده با یکدیگر را می‌دهد.

b. **وارسی اسکیمای جزئی در دریافت:** چارچوب metron داده را بر روی دریافت و فیلتر داده بی‌اعتبار به‌طور خودکار اعتبارسنجی می‌کند، که این قابلیتی است که متخصصین علم داده، به‌طور سنتی زمان زیادی را صرف انجام آن می‌کنند.

۳. **تغییر داده‌های ناقص**

a. **غنی‌سازی خودکار داده:** به‌طور معمول به‌صورت دستی داده را برای ایجاد و تست ویژگی‌ها یا غنی نمایند یا بایستی با تیم چارچوب/داده برای انجام آن، کار کنند. با Metron رویدادها در زمان واقعی غنی و ذخیره می‌شوند.

b. **برنامه کاربردی خودکار برچسب‌های کلاس**: انواع مختلفی از ابرداده (اطلاعات هوشمندی تهدید و غیره) بر روی رویداد برچسب‌گذاری می‌شوند که به متخصصین امکان ایجاد آسان تر ماتریس‌های ویژگی را برای مدل‌ها فراهم می‌آورد.

c. **چارچوب محاسباتی عظیم موازی:** کلیه عملیات پاک‌سازی و تغییر داده‌ها از فن‌آوری‌های توزیع یافته‌ای استفاده می‌کنند که امکان پردازش مقدار بسیار بزرگ یا سرعت بالای اجراکننده را فراهم کرده و مقیاس‌پذیر است.

٤. **مصورسازی داده**

a. **جست‌وجوی بی‌درنگ همراه با رابط کاربری:** Metron کلیه وقایع و هشدارها را شاخص‌گذاری کرده و داشبورد UI برای انجام جست‌وجوی بی‌درنگ را فراهم می‌کند.

b. **داشبوردهای Apache Zeppelin**: داشبوردهای خارج از سرویس Zeppelin نیز می‌توانند توسط تحلیل‌گر SOC به‌کار روند. با Zeppelin شما می‌توانید داشبوردها و مقادیر جایگزین را به اشتراک گذارید و به‌سرعت انواع گراف‌ها را تغییر دهید.

c. **یکپارچه‌سازی با Jupyter**: یادداشت‌های Jupyter برای وظایف مشترک از جمله شناسایی، مصورسازی، ترسیم، ارزیابی ویژگی‌ها و غیره به دانشمندان علم داده ارائه شده است.

چهار مرحله بالا در جریان علم داده حدوداً ۸۰٪ از زمان متخصصین تحلیل علم داده را به خود اختصاص می‌دهد. Metron به‌طور قابل ملاحظه‌ای زمان را از لحظه فرضیه تا مدل کاهش می‌دهد.

۲-٤ مزایا به‌عنوان یک ابزار SIEM

هدف از تولید Metron ارائه یک چارچوب متن باز است تا افراد بتوانند همکاری کنند و مجموعه داده‌ها، مدل‌های یادگیری ماشین، تجزیه‌کننده‌های تله‌متری، منابع غنی‌سازی تله‌متری و مخازن هوشمندی تهدید را به اشتراک گذارند. در امنیت سایبری مقابله به تنهایی با تهدیدات، ایجاد سازگاری با ابزارهای فعلی و برقراری تجمیع میان واحدهای مجزا به‌صورت یک چارچوب واحد و قدرتمند، برای یک سازمان چالشی بزرگ است.

علاوه بر متن باز بودن و تسهیل پیشرفت در تحلیل‌های امنیتی، Metron مزایای متعددی نسبت به سیستم SIEM معمولی دارد:

- **هزینه کم**: Metron از کلیه پشته‌های متن باز تحت یک فرآیند تأمین اطلاعات علم داده[٤] استفاده می‌کند و بر روی سخت‌افزار مناسب اجرا می‌شود. این بدان معنی است که Metron جهت اجرا بسیار کم‌هزینه است. هزینه در امنیت فاکتور مهمی است، زیرا هزینه اقدامات متقابل برای پایش و واکنش

[٤] under the hood: این اصطلاح بیان می‌کند که فرآیند تأمین اطلاعات علم داده دارای چند بخش است که شامل مراحل زیر می‌باشد:

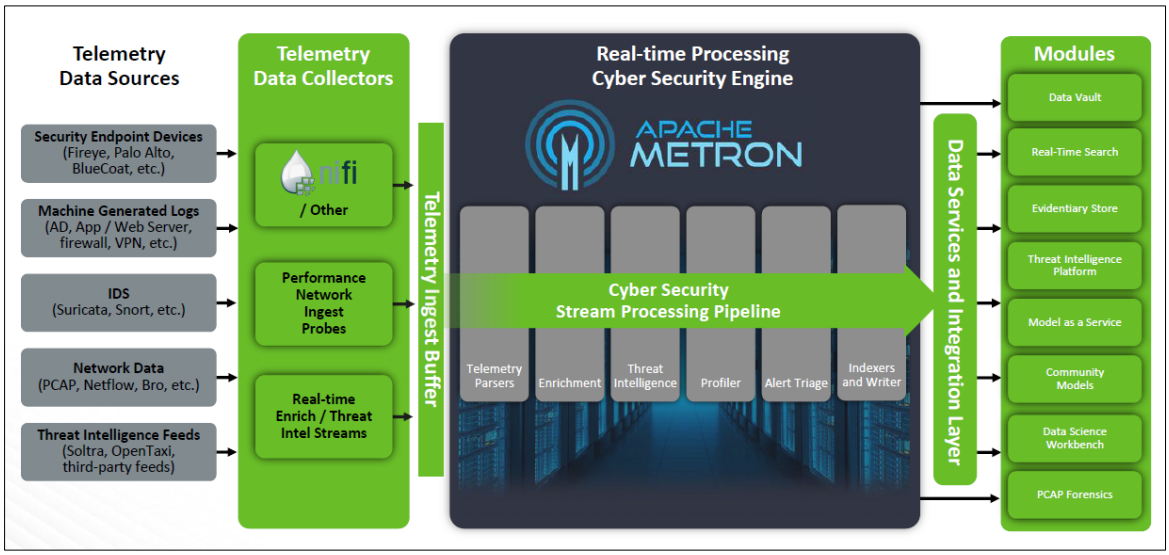
۱. ضبط
۲. سازماندهی
۳. مصورسازی
٤. تحلیل
۵. پیاده‌سازی

به تهدید نباید از هزینه چیزهایی که بایستی محافظت شود، تجاوز نماید. با صرف هزینه‌های امنیتی پایش دارایی‌های بیشتر در این صورت اقتصادی است و مراکز داده امن‌تر خواهد شد.

- **باز بودن:** متن باز بودن Metron امکان بررسی و رسیدگی دقیق‌تر جامعه متن باز را برای کلیه اجزای آن ایجاد می‌کند. بازبودن برای یک ابزار امنیت محور، مدل بهتری نسبت به بسته‌بودن است. کلیه مشکلات بایستی از بین رفته و باز بودن آن کاملاً رعایت گردد، اما رقابت در بسته‌بودن این سختی را به همراه ندارد و با بازاریابی و فروش، انگیزه را افزایش می‌دهد بنابراین از لحاظ امنیتی هیچ اعتماد به نفسی را به‌وجود نمی‌آورد.
- **مبتنی بر هادوپ:** Metron می‌تواند حجم بی‌سابقه‌ای از داده در حال جریان را با Apache Storm پردازش نماید. هنگامی‌که یک سازمان با رفتار یا نرم‌افزارهای مخرب مواجه می‌شود که اغلب به‌عنوان بخشی از عملیات عمومی بدافزار رخ می‌دهد، امضاها شناخته می‌شوند و از منابع هوشمندی تهدید شخص ثالث تهیه می‌شوند. توانایی در گرفتن از منابع و مراجعه به آن‌ها در برابر هر پیام تله‌متری پردازش شده توسط Metron در زمان واقعی نه تنها تشخیص چنین عملیاتی را تسهیل می‌بخشد، بلکه اقتصاد را برای موارد ناهنجار تغییر می‌دهد. اگر اجباری به سفارشی نمودن بدافزار برای هر هدف باشد، این حملات عمومی بسیار گران و غیرقابل تحمل خواهد بود.
- Metron تلاش می‌کند تا جریان کار متداول SOC را بیشتر از یک رویکرد قانون‌محور به سمت یک رویکرد داده‌محور که شامل یادگیری ماشین و میزان بالایی از خودکارسازی و تشخیص مستقل می‌باشد، تغییر دهد.
- چشم‌انداز تهدید مدرن برای کنترل فقط با قوانین ایستا، بسیار پویا می‌باشد و این همان چیزی است که SIEMهای معمولی به آن وابسته هستند. اگر کنترل‌های مبتنی بر قانون به‌درستی نگهداری نشوند تبدیل به منابعی از هشدارهای مثبت کاذب می‌شوند.

۳ معماری HCP[۵]

بخش اصلی در معماری بسته امنیت سایبری Hortonworks (HCP)، Apache Metron است که موتور امنیتی پردازش بی‌درنگ مدرن بوده و متشکل از Apache Hadoop، Apache Storm و تکنولوژی‌های وابسته می‌باشد که در ادامه گزارش به بررسی ساختار آن خواهیم پرداخت. HCP چارچوب و ابزارهایی را برای مراکز عملیات امنیت (SOC) جهت تشخیص سریع‌تر و بهتر تهدید در ابعاد بسیار وسیع به‌صورت بی‌درنگ فراهم می‌کند.



شکل ۳-۱: چارچوب HCP

چارچوب HCP، دریافت، تجزیه و نرمال‌سازی داده کاملاً غنی‌شده و متنی، منابع هوشمندی تهدید، طبقه‌بندی و یادگیری ماشین بر پایه تشخیص را فراهم می‌کند، همچنین داشبوردهای تقریباً بی‌درنگ از کاربر نهایی براساس شالوده قوی در پشته‌های چارچوب داده Hortonworks[۶](HDP) و جریان کاری Hortonworks[۷](HDF) را فراهم می‌آورد و چارچوب پیشرفته یکپارچه‌ای را برای تحلیل‌های امنیتی ارائه کرده است. جریان کاری در HCP نیز در زمان واقعی انجام می‌گیرد و شامل مراحل زیر است:

---

[۵] Hortonworks Cybersecurity Package

[۶] Hortonworks Data Platform

[۷] Hortonworks Dataflow

۱. اطلاعات از منابع داده تله‌متری به Kafka topic (Kafka در این‌جا بافر رویداد تله‌متری را دارد). Kafka topic برای هر منبع داده تله‌متری ایجاد می‌شود. این اطلاعات داده خام تله‌متری است که شامل فایل‌های ثبت وقایع میزبان، دیواره آتش، ایمیل‌ها و داده‌های شبکه می‌باشد.
۲. همین‌که اطلاعات به Kafka topicها وارد می‌شود، داده به‌صورت یک ساختار نرمال‌شده JSON و قابل خواندن برای Metron تجزیه می‌شود.
۳. سپس اطلاعات موجود با اطلاعات هوشمندی تهدید، جغرافیایی، دارایی و غیره غنی می‌شوند.
٤. اطلاعات شاخص‌گذاری و ذخیره‌سازی شده و نتیجه هر هشدار به داشبورد Metron، رابط کاربری هشدار و همچنین تله‌متری ارسال می‌گردد.

٤ معماری Apache Metron

به‌طور کلی معماری Apache Metron بر پایه عملیات و نیازمندی‌های زیر طراحی شده است:

- جمع‌آوری داده
- پردازش بی‌درنگ داده (تجزیه، غنی‌سازی، هشداردهی، شاخص‌گذاری، ذخیره‌سازی، اجرای قوانین/مدل)
- تحلیل‌ها
- داشبوردها برای تحلیل‌گران

Apache Metron هسته‌اصلی معماری Kappa با Apache Storm مؤلفه پردازش و Aapache Kafka گذرگاه داده واحد را تشکیل می‌دهد.

مؤلفه‌های عمده Apache Metron شامل ماژول‌ها و زبان‌های خاص دامنه می‌باشند. بنابراین ساختار Apache Metron از ماژول‌های مختلفی شامل موارد زیر تشکیل شده است:

- **تجزیه‌کننده**: توپولوژی برای نرمال‌سازی تله‌متری از قالب‌های بومی حس‌گر به‌صورت Metron JSON
- **غنی‌سازی**: توپولوژی برای غنی‌سازی پیام‌های JSON در Metron، مراجع آن‌ها در برابر ذخیره‌سازی‌های هوشمندی تهدید و انتشار هشدارها
- **PCAP**: توپولوژی برای جریان بسته‌های شبکه به HDFS برای استفاده با سرویس PCAP

- **API**: سرویس برای اجرای تحلیل‌ها/فیلترکردن بر روی فایل‌های PCAP در ورودی HDFS با توپولوژی PCAP
- **حس‌گرها**: حس‌گرهای منبع داشبودهای Metron و تحلیل‌ها
- **مدیریت داده**: بارگذارها برای بارگذاری عمده غنی‌سازی و ذخیره‌سازی هوشمندی تهدید
- **رابط کاربری**: رابط کاربری برای تحلیل‌گر SOC
- **استقرار**: اسکریپت‌ها برای خودکارسازی استقرارهای Metron

بنابراین ساختار آن دارای زیربخش‌های زیر است که در ادامه به بررسی روند کار آن‌ها خواهیم پرداخت:

تجزیه‌کننده‌ها: تجزیه داده از Kafka به مدل داده Metron و عبوردادن آن به دیگر بخش‌ها

غنی‌سازی: غنی‌ساختن پسا-تجزیه[۸] داده و امکان برچسب‌گذاری یک پیام به‌عنوان هشدار و تخصیص سطح شاخص‌گذاری با HDFS، Elasticsearch و Solr

برخی از ابزارهایی که در تمام بخش‌های معماری وجود دارند عبارتنداز:

Stellar: یک زبان سفارشی تبدیل داده است که در سراسر ساختار Metron از تبدیل ساده فیلد تا بیان قوانین طبقه‌بندی به‌کار می‌رود.

مدل به‌عنوان سرویس: برنامه کاربردی Yarn می‌تواند مدل‌های آماری و یادگیری ماشین را به خوشه همراه با توابع Stellar مربوطه گسترش دهد تا بتواند با آن‌ها به روشی مقیاس‌پذیر ارتباط برقرار کند.

مدیریت داده: مجموعه‌ای از ابزارهای مدیریت داده با هدف تهیه داده برای HBase در قالبی است که امکان جریان داده از طریق Metron را فراهم می‌کند تا با نتایج حاصل شده غنی گردد. این ابزارها شامل اجتماع با منابع دردسترس هوشمندی تهدید از طریق TAXII با ساختارهای فایل مسطح ساده می‌باشد.

Profiler: سازوکار استخراج ویژگی است که می‌تواند پروفایلی را تولید کند که توصیف‌کننده رفتار یک هویت باشد. موجودیت ممکن است یک سرویس‌دهنده، کاربر، زیرشبکه یا برنامه کاربردی باشد. هنگامی‌که

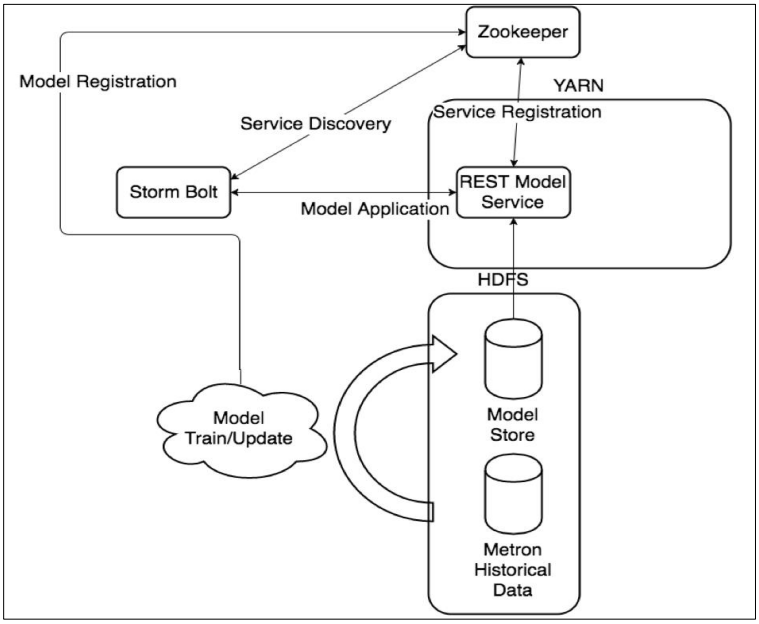
[۸] post-parsing

پروفایل تولید گردد آن‌چه را که رفتار طبیعی به نظر می‌رسد، مدل‌هایی که می‌تواند ساخته شوند تا رفتار غیرنرمال شناسایی گردد، را تعریف می‌کند.

٤-١ زیرساخت مدیریت مدل

یکی از مهمترین ویژگی‌های پیش‌بینی شده و مورد انتظار توانایی رشد در فرآیند هوشمندی تهدید و غنی‌سازی به همراه بینش‌های حاصل از مدل آماری یا یادگیری ماشین است. چالش‌های موجود در این زیرساخت عبارتند از:

- اجرای مدل ممکن است که به‌اندازه کافی محاسباتی باشد یا با رشد و تشدید منابع درخواستی روبرو گردد و نیازمند به پشتیبانی از مقیاس‌بندی از طریق تعادل بار می‌باشد که در این صورت ملزم به مدیریت و کشف سرویس است.
- مدل‌ها نیاز به تمرین‌های بیش از حد و پی در پی دارند تا به رشد تهدیدات و الگوهای جدید ایجاد شده برسند. مدل‌ها تا جای ممکن بایستی سازگار با زبان/محیط[٩] باشند.



شکل ٤-١: زیرساخت مدیریت مدل

[٩] language/environment agnostic

به‌منظور پشتیبانی از الزامات فنی در مدل‌های موجود، مؤلفه‌های زیر ایجاد می‌شوند:

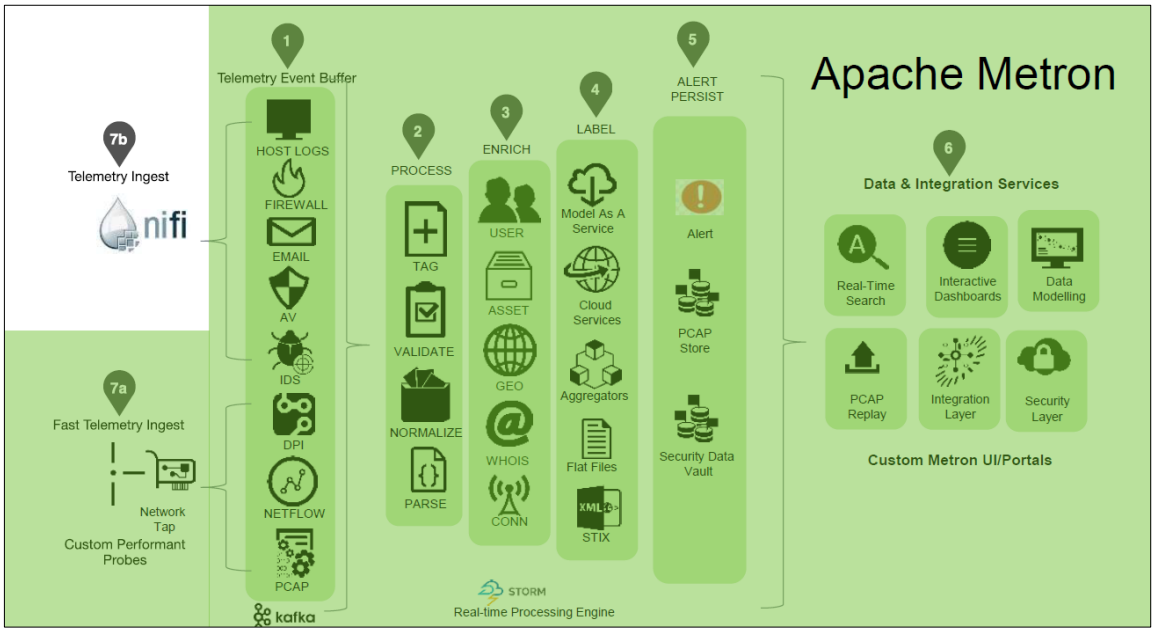
- برنامه کاربردی Yarn که درخواست‌های پیاده‌سازی و اجرای مدل را رصد می‌کند و نقاط پایانی آنها را در Zookeeper ثبت می‌کند.
  - انواع عملیات: ADD، REMOVE، LIST
  - نام مدل
  - نسخه مدل
  - نیازمندی‌های حافظه (به مگابایت)
  - تعداد موارد
- مجموعه‌ای از توابع Stellar برای تعامل با مدل‌ها از طریق زیرساخت مدل به‌عنوان سرویس استقراریافته‌اند.

چند اسکریپت در رابطه با زیرساخت فوق ارائه شده است:

- **maas_service.sh**: اسکریپت maas_service.sh برنامه کاربردی Yarn را آغاز می‌کند که درخواست‌ها را رصد می‌کند. در حال حاضر صف درخواست‌ها یک صف توزیع‌شده ذخیره‌شده برای راحتی کار در zookeeper می‌باشد.
- **maas_deploy.sh**: اسکریپت maas_deploy.sh اجازه می‌دهد تا مدل‌ها و تضمین‌هایشان را از طریق دیسک محلی در خوشه فراهم کند.

٤-٢ معماری منطقی Apache Metron

شکل ٤-٢ مؤلفه‌های منطقی چارچوب Metron را نمایش می‌دهد.



شکل ٤-٢: معماری منطقی Apache Metron

Apache Metron شامل مؤلفه‌ها و بخش‌های پیش‌فرض زیر است:

- Apache Flume
- Apache Hadoop
- Apache HBase
- Apache Hive
- Apache Kafka
- Apache Spark
- Apache Storm
- ElasticSearch
- MySQL

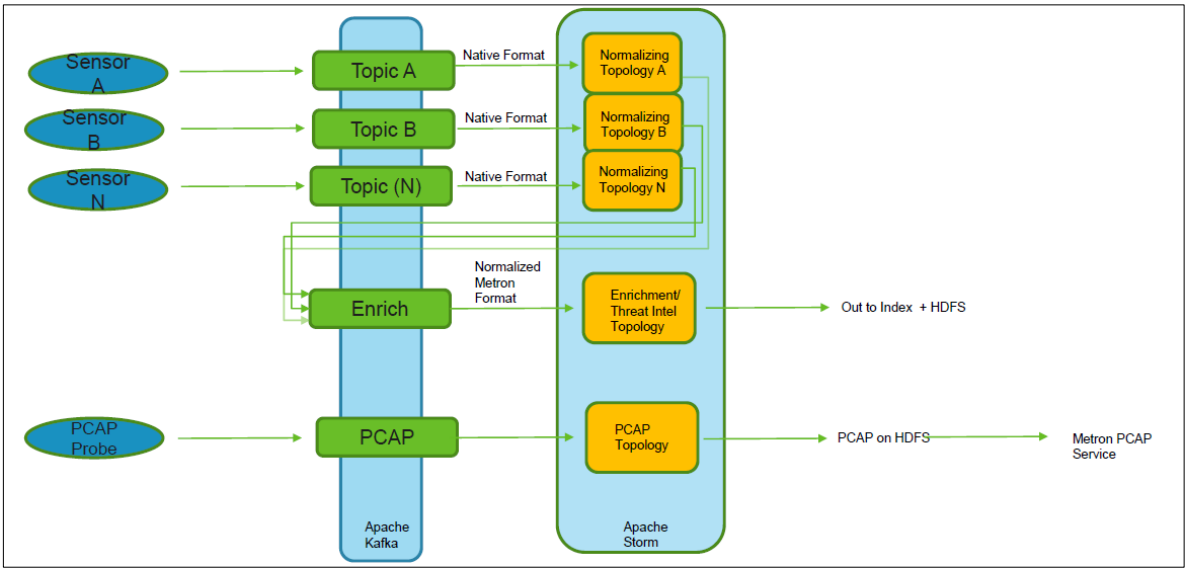
در ساختار Apache Metron حس‌گرهای تله‌متری داده را تولید نموده و به Kafka topic ها وارد می‌کنند. Metron از پردازش‌گر جریان Storm استفاده می‌کند. توپولوژی Storm داده خام را گرفته آن را با قالب JSON عبور می‌دهد. قالب JSON دارای ۸ فیلد اضافی است. خروجی تجزیه‌شده دوباره به Kafka داده شده و خروجی آن به توپولوژی غنی‌سازی Storm تحویل داده می‌شود. با استفاده از توپولوژی غنی‌سازی می‌توانیم فیلدها را با دیگر اطلاعات مانند آدرس IP با اطلاعات مکانی یا اطلاعات لازم برای DNS غنی کنیم. برای پردازش سریع، هر غنی‌سازی از طریق حافظه محلی پشتیبانی می‌شود. سپس مخزن هوشمندی تهدید، مخازن متفاوتی را نگه‌داری می‌کند که دربرگیرنده اطلاعات درباره آدرس‌های IP بدخواه است که در برابر داده تولید شده وارسی می‌شود. سرانجام، داده‌های تحت ترمیم با استفاده از موتور جست‌وجوی Elasticsearch یا Solr شاخص‌گذاری می‌شوند. پشته Metron از Kibana به‌عنوان رابط کاربری سمت کاربر در Elasticsearch

استفاده می‌کند. برای تحلیل جریان زنده، داده خام منتقل شده در سراسر شبکه نیز می‌تواند با از ابزار PCAP (ضبط بسته) ضبط گردد. این داده در HDFS ذخیره شده و در هر زمانی قابل بازسازی است.

٤-٢-١ پردازش جریانی (Streaming)

Streaming ماژولی است که منجر به پردازش جریان شامل دریافت تله‌متری، غنی‌سازی، مراجع هوشمندی تهدید، هشداردهی و رتبه‌بندی بی‌درنگ مدل‌های یادگیری ماشین می‌شود. streaming در Metron بر پایه Apache Storm ساخته شده است که یک موتور پردازش جریان مقیاس‌پذیر با ویژگی‌های منحصربه‌فرد می‌باشد و آن برای پردازش شبکه‌بندی و رویدادنگاری داده مناسب می‌سازد. تله‌متری توسط حس‌گرهایی که با توپولوژی‌های Storm پردازش شده‌اند، تولید می‌گردد. سه نوع توپولوژی وجود دارد که در شکل ٤-٣ معماری فیزیکی آن نشان داده شده است:

- توپولوژی نرمال‌سازی/تجزیه
- توپولوژی هوشمندی تهدید/غنی‌سازی
- توپولوژی PCAP

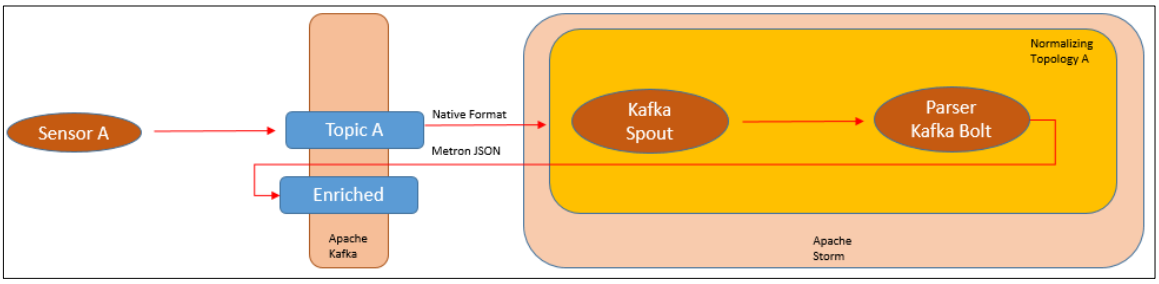


شکل ٤-٣: انواع توپولوژی در Apache Metron

٤-٢-١-١ توپولوژی تجزیه

توپولوژی تجزیه Metron به‌منظور گرفتن ورودی حس‌گر در قالب بومی و تبدیل آن به JSON Object طراحی شده‌است. توپولوژی از دو مؤلفه تشکیل شده است: Storm Kafka Spout که داده را از Kafka

topic خوانده و آن را به توپولوژی storm می‌دهد و Parser Kafka bolt که پیام را تجزیه نموده و به topic غنی‌سازی Kafka باز می‌گرداند.



شکل ٤-٤: توپولوژی تجزیه

Parser Kafka bolt یک bolt توسعه‌پذیر است که می‌تواند دو نوع تجزیه‌کننده بگیرد: GROK و Java برای اهداف خاص. اما تجزیه‌کننده‌های Grok کندتر از تجزیه‌کننده‌های Java هستند و گاهی اوقات تله‌متری در نوشتن وضعیت Grok بسیار پیچیده می‌شود. هر خروجی هر تجزیه‌کننده به‌دلیل ساختار داده انعطاف‌پذیر و امکان حلقه و لیست، با JSON قالب‌بندی شده است.

٤-٢-١-٢ تجزیه‌کننده‌های Grok

تجزیه‌کننده ورودی را که معمولاً آرایه‌ای از بایت‌های دریافت شده از Kafka Spout است، می‌گیرد و آن را به JSON Object تبدیل می‌کند. تجزیه‌کننده Grok این عمل را با بهره‌گیری از کتابخانه Grok داخل Parser Kafka Bolt Adaptor انجام می‌دهد. بنابراین به‌جای کدنویسی تجزیه‌کننده در Java، Grok Adapter امکان ایجاد Grok Statementها یا خروجی‌های قابل فهم توسط Storm bolt را می‌دهد که قابل تبدیل به قالب JSON هستند.

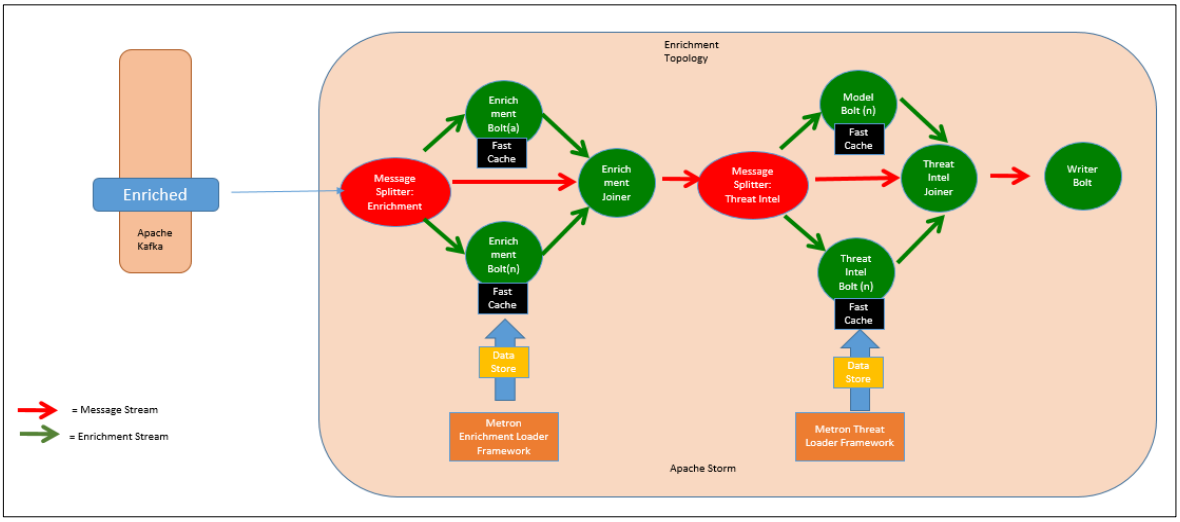
٤-٢-١-٣ تجزیه‌کننده Java

اغلب تجزیه‌کننده‌ها برای Metron، به‌غیر از زمانی که تله‌متری یا با سرعت بسیار زیاد یا خیلی پیچیده باشد، بایستی به‌عنوان تجزیه‌کننده‌های Grok نوشته شوند، در این صورت نوشتن بیان Grok برای آن غیرممکن

است. کلیه تجزیه‌کننده‌های Java باید مبنای تجزیه‌کننده Metron[۱۰] و خروجی JSON Object را توسعه دهند.

٤-۲-۱-٤ توپولوژی هوشمندی تهدید/غنی‌سازی

ورودی توپولوژی غنی‌سازی قالب JSON تولید شده توسط تجزیه‌کننده در توپولوژی تجزیه می‌باشد. خروجی این توپولوژی با مقدار زیادی از منابع داده قابل پشتیبانی Metron نوشته می‌شود. دو نوع جریان وجود دارد: جریان پیام و جریان‌های غنی‌سازی. جریان پیام، پیام اصلی را حمل می‌کند در حالی‌که جریان غنی‌سازی بر روی غنی‌سازی‌ها یا قطعات هوشمندی تهدید بر روی پیام ضمیمه می‌شود.



شکل ٤-۵: توپولوژی هوشمندی تهدید/غنی‌سازی

انواع boltها شامل موارد زیر است:

- جداکننده غنی‌سازی[۱۱]
- غنی‌سازی[۱۲]
- پیونددهنده غنی‌سازی[۱۳]

---

[۱۰] Metron Parser Base

[۱۱] Enrichment Splitter

[۱۲] Enrichment Bolt

[۱۳] Enrichment Joiner Bolt

- هوشمندی تهدید[۱۴]
- پیونددهنده هوشمندی تهدید[۱۵]

٤-۲-۱-٤-۱ غنی‌سازی

غنی‌سازی‌ها، متن اضافه را به پیام streaming اضافه می‌کنند. برای مثال، اگر پیام داده‌شده یک IP خارجی داشته باشد یک غنی‌سازی، داده geo را به پیام برچسب می‌زند. مثال دیگر زمانی خواهد بود که اگر پیام حاوی نام دامنه باشد می‌توان پاسخ‌های whois را به پیام برچسب زد. سه مزیت اصلی نسبت به افزودن متن از طریق غنی‌سازی‌ها به پیام وجود دارد:

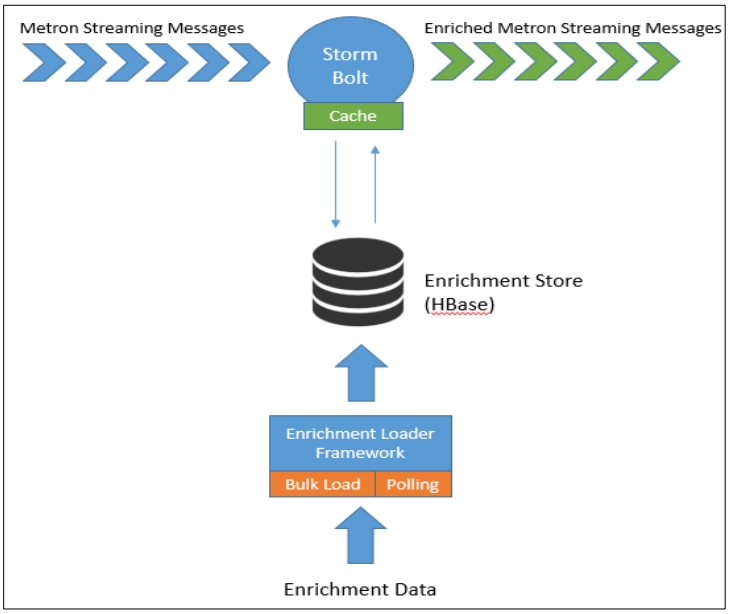
- **همبستگی**: اگر بدانید که کدام کاربر و دارایی پیام را در نظر می‌گیرد و از کجا می‌آید برای همبسته‌سازی آن با دیگر پیام‌ها آسان‌تر و کاربردی‌تر است.
- **یادگیری ماشین**: داشتن کل متن از طریق streaming امکان رتبه‌بندی در برابر مدل‌های یادگیری ماشین به‌طور بی‌درنگ را امکان پذیر می‌کند.
- **دقت**: اطلاعات اساسی غنی‌سازی همیشه تغییر می‌کند (کاربران وارد و خارج می‌شوند، آدرس IP ماشین‌ها تغییر می‌کند و غیره) و تحلیل‌گر نیز قصد دارد تا جایی که امکان دارد، نزدیک به زمان ضبط غنی‌سازی را انجام دهد.
- **تحقیقات**: با داشتن متن کامل برای قطعه داده شده از ابرداده یا هشدار نیاز به کنسول‌های کمتری است و فرد را به رابط کاربری شفاف نزدیک‌تر می‌کند.

Metron چارچوب توسعه‌پذیر را برای اتصال غنی‌سازی فراهم می‌کند. هر غنی‌سازی دو مؤلفه دارد: منبع داده غنی‌سازی و bolt غنی‌سازی.

---

[۱۴] Threat Intel Bolt

[۱۵] Threat Intel Joiner Bolt



شکل ٤-٦: نحوه غنی‌سازی در Apache Metron

پیش از فعال‌سازی قابلیت غنی‌سازی در داخل Metron، منبع غنی‌سازی Metron که عمدتاً Hbase هستند، با داده غنی‌سازی بارگذاری می‌شوند. داده غنی‌سازی می‌تواند یا به‌صورت انبوه از HDFS بارگذاری گردد یا به منبع غنی‌سازی با چارچوب قابل جابجایی بارگذاری جاری شود. بارگذار غنی‌سازی، پیام‌های غنی‌سازی را به صورت قالب JSON که قابل درک می‌باشد، به Metron انتقال می‌دهد. چارچوب بارگذاری، قابلیت‌های اضافی برای داده‌های پیرامون منابع غنی‌سازی مبتنی بر زمان دارد. هنگامی که منابع بارگذاری می‌شوند، یک bolt غنی‌سازی که می‌تواند با منبع غنی‌سازی ارتباط برقرار کند، می‌تواند به توپولوژی غنی‌سازی متصل گردد. هر bolt غنی‌سازی می‌تواند برچسب/فیلد خاصی را در داخل پیام Metron غنی‌سازی نماید. زمانی که یک bolt تشخیص دهد که قادر به غنی‌سازی یک فیلد است، به منبع غنی‌سازی می‌رسد، غنی‌سازی را از بین می‌برد و پیام را با غنی‌سازی برچسب‌گذاری می‌کند. سپس غنی‌سازی داخل حافظه in-memory در bolt ذخیره می‌شود. Metron از قابلیت‌های اصولی مسیریابی در Storm بهره می‌گیرد تا اطمینان حاصل کند که مقادیر مشابه غنی‌سازی به boltهای متناسبی ارسال گردد که قبلاً این مقادیر در حافظه قرار گرفته‌اند، در نتیجه Metron سرعت و مقیاس فوق‌آلعاده آن را می‌گیرد نسبت به دیگر سیستم‌های پردازش کلان داده که این قابلیت را ندارند.

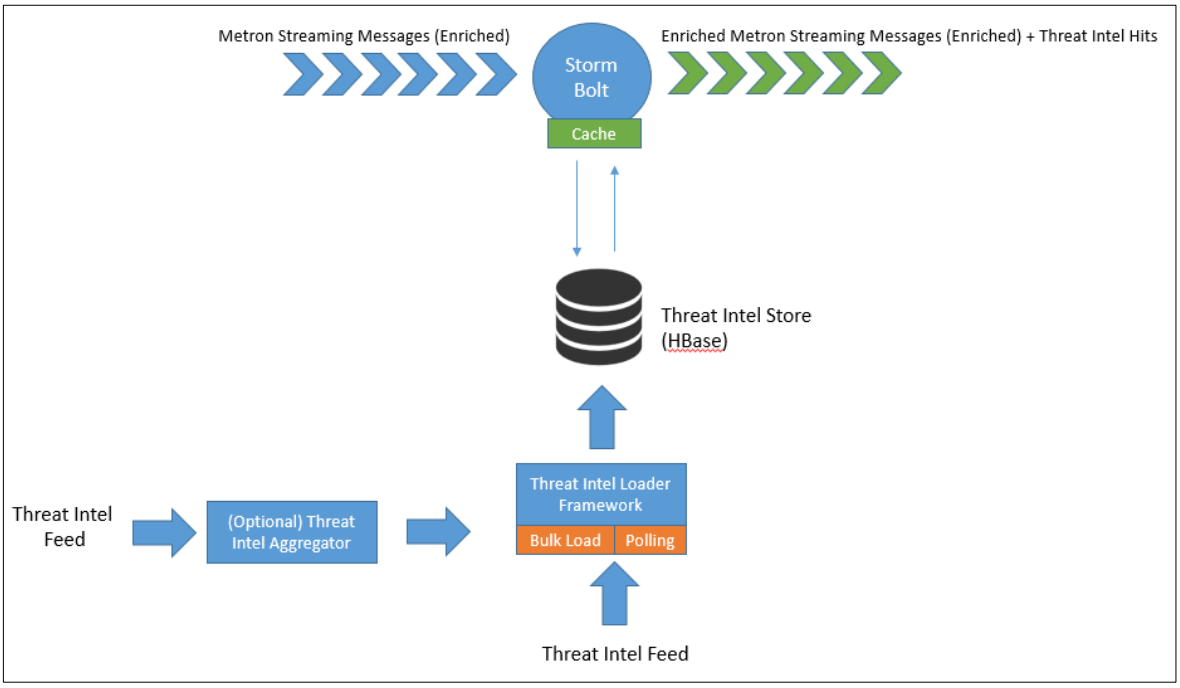
جدول ٤-١ لیست غنی‌سازی‌هایی را نشان می‌دهد که در Metron پشتیبانی می‌شوند.

جدول ٤-١: لیست غنی‌سازی‌ها در Metron

غنی‌سازی	منبع ذخیره‌سازی	منبع غنی‌سازی	فیلد پیام Metron	نوع بارگذار	نرخ تازه‌سازی	معماری غنی‌سازی Metron
GeoIP	MySQL	Maxmind Geolite /http://dev.maxmind.com/geoip/legacy/geolite	src_ip, dst_ip	انبوه و طبقه از HDFS	هر سه ماه یکبار	Geo Enrichment
Asset	HBase	LDAP, AD, DNS logs, enterprise inventory stores	src_ip, dst_ip	هنوز فراهم نشده	در هر ساعت	Asset Enrichment
User	Hbase	LDAP, AD, proxy logs	src_ip + application	هنوز فراهم نشده	هر ٥ ماه یکبار	User Enrichment

٤-٢-١-٤-٢ هوشمندی تهدید

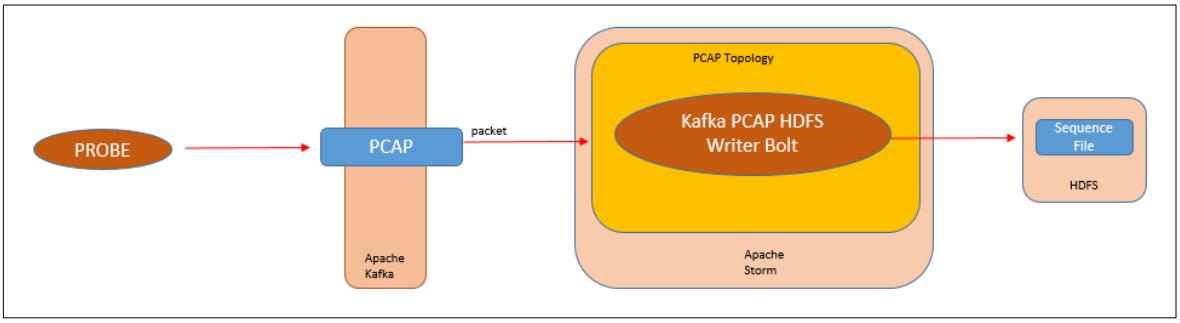
Metron چارچوب گسترده را برای اتصال به منابع تهدید امنیتی فراهم می‌کند. هر منبع هوشمندی تهدید دو مؤلفه دارد: یک منبع داده غنی‌سازی و یک bolt غنی‌سازی. منابع هوشمندی تهدید به‌صورت انبوه بارگذاری شده و به منبع هوشمندی تهدید جاری می‌شوند، مشابه با آن‌چه در بارگذاری منابع غنی‌سازی انجام می‌شود. کلیدها در قالب مقدار-کلید بارگذاری می‌شوند. کلید نشان‌گر است و مقدار، توضیح JSON قالب‌بندی شده است از آن‌چه که نشان‌گر می‌باشد. توصیه می‌شود که از گردآورکننده منبع تهدید از جمله Soltra استفاده شود تا منابع با Stix یا Taxii نرمال‌سازی شوند. Metron یک آداپتور را فراهم می‌کند که قادر به خواندن منابع Stix/Taxii تولید شده توسط Soltra باشد و آن‌ها را به سمت HBase جاری نماید. Hbase منبع داده جهت بازگشت به عقب فوق سریع هوشمندی تهدید در Metron است. به‌علاوه، Metron یک فایل مسطح و بارگذار انبوه Stix را فراهم می‌کند تا بتواند نرمال‌سازی، حذف و بارگذاری انبوه یا streaming داده هوشمندی تهدید را به HBase انجام دهد، حتی اگر بدون استفاده از یک گردآورکننده منبع تهدید باشد.



شکل ٤-٧: نحوه هوشمندی تهدید در Apache Metron

٤-٢-١-٥ توپولوژی PCAP

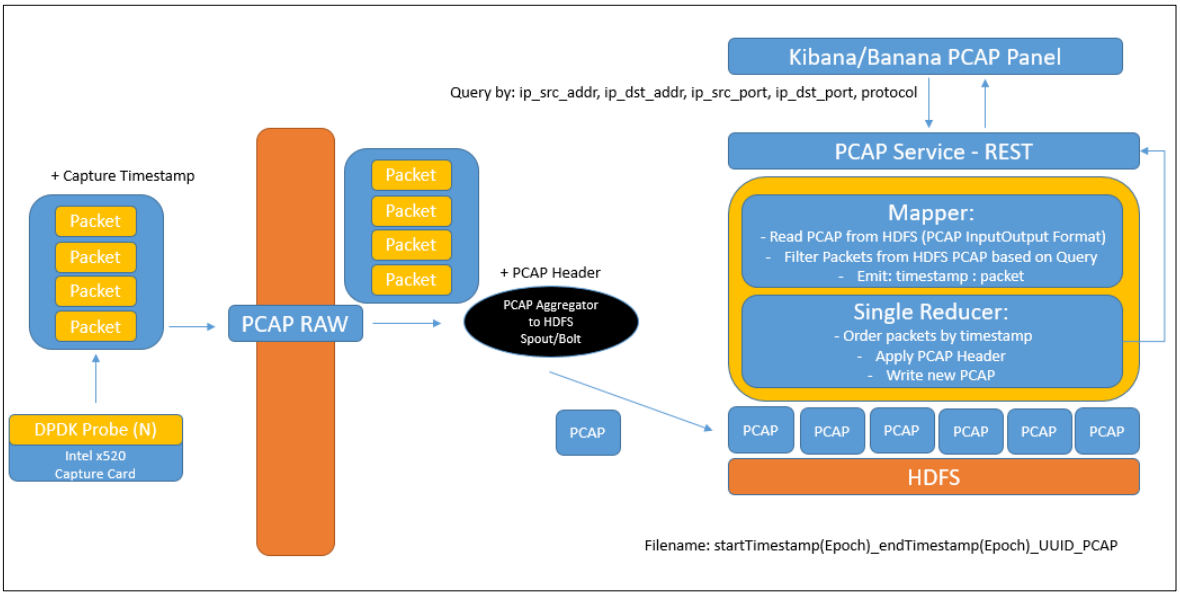
توپولوژی PCAP به‌منظور گرفتن بسته‌های شبکه از PCAP Kafka topic و ذخیره‌سازی آن‌ها در HDFS به عنوان فایل‌های توالی، طراحی می‌شود. آن یک bolt مجزا دارد که همانند Kafka Spout عمل می‌کند و همچنین HDFS Writer در جایی بسته‌ها در HDFS ذخیره‌شده باشند. توپولوژی PCAP وابستگی زیادی به PCAP Probe دارد.



شکل ٤-٨: توپولوژی PCAP

٤-٢-١-٦ سرویس PCAP

شرط لازم برای داشتن سرویس PCAP، داشتن توپولوژی PCAP آماده و در حال اجرا است.



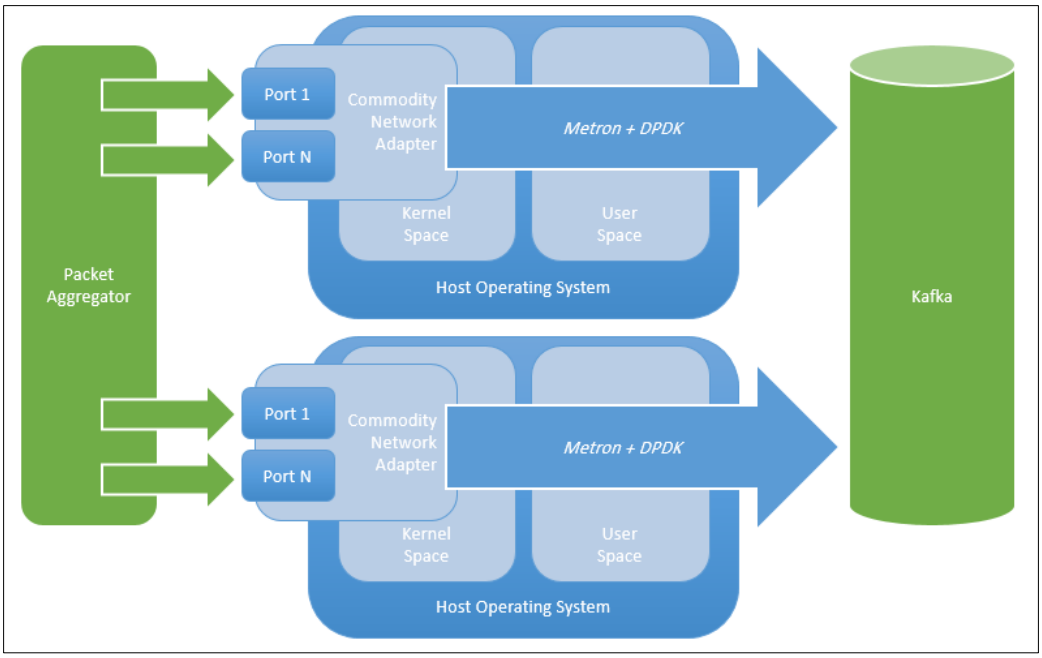
شکل ٤-٩: سرویس PCAP در Apache Metron

سرویس PCAP شامل پنل PCAP در Kibana است که توسط یک Restful API پشتیبانی می‌شود. پنل فوق شامل اطلاعات زیر است:

- Source IP
- Dest IP
- Source port
- Dest port
- Protocol
- Timeframe

هرگاه یک کوئری در پنل PCAP وارد گردد، پنل یک سرویس REST PCAP را منتقل می‌کند. سرویس REST PCAP کار MR را از طریق فایل‌های PCAP ذخیره‌شده بر روی HDFS توسط توپولوژی PCAP پر می‌کند و آن‌ها را براساس پرس‌وجوی پنل Kibana فیلتر می‌کند و PCAP جدید را از پرس‌وجوی PCAP کامپایل نموده و از طریق سرویس REST PCAP به پنل Kibana می‌گرداند.

٤-٢-١-٧ کاوش‌گر ضبط بسته



شکل ٤-١٠: کاوش‌گر ضبط بسته

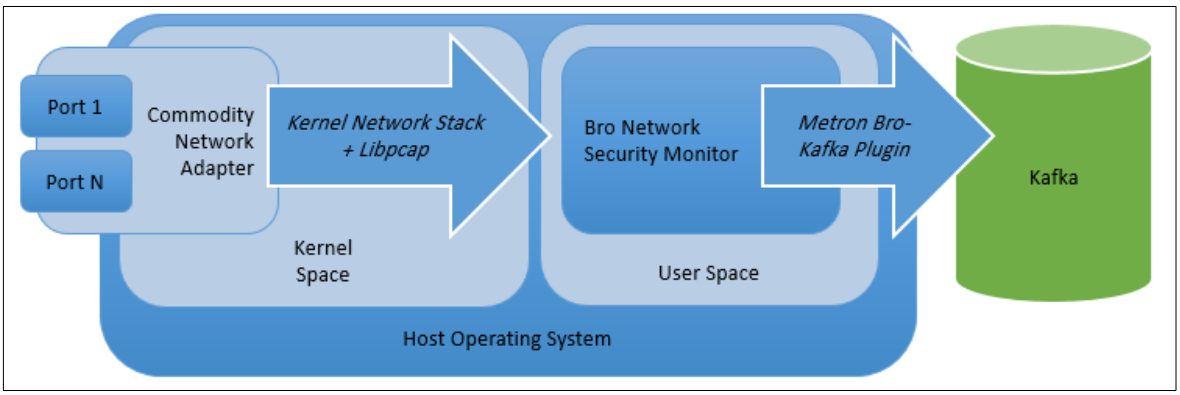
یک کاوش‌گر برای استفاده با سرویس PCAP در Metron برای ضبط با سرعت بالای بسته از شبکه، طراحی می‌شود. این حس‌گر برای ضبط بسته‌های خام از طریق سیم و بارگذاری انبوه آن‌ها برای Kafka با نرخ بسیار بالا طراحی شده است. تله‌متری که این حس‌گر تولید می‌کند یک تله‌متری با حداکثر گذردهی در Metron است. در حالی‌که کاوش‌گر به سرویس PCAP اتصال محکمی ندارد، استفاده از هر دو با یکدیگر برای بازیابی PCAP بسیار توصیه می‌شود.

در معماری این حس‌گر، فایل‌های Kafka توسط توپولوژی PCAP Storm برداشته شده و به‌صورت انبوه به HDFS بارگذاری می‌گردد. هر فایل در HDFS به‌عنوان فایل توالی ذخیره می‌شود. هنگامی‌که در HDFS، سرویس PCAP برای خواندن به‌کار می‌رود و فایل‌های توالی، فایل‌های PCAP سازگار را از طریق یک restful API تحویل می‌دهد. می‌توان کاوش‌گرهای متعددی را به Kafka متصل نمود. سخت‌افزار توصیه شده از خانواده آداپتورهای شبکه اینتل است که با DPDK پشتیبانی می‌گردد.

٤-١-٢-٨ Bro

Bro عمدتاً به‌عنوان مولد داده و بازرسی عمیق بسته[١٦] (DPI) به‌کار می‌رود. در حال حاضر Metron از ویژگی‌های هشدارهای IDS در Bro استفاده نمی‌کند. Metron با Bro از طریق افزونه Bro کار می‌کند و نیازی به تغییر در کدهای Bro نیست. افزونه Bro پیام‌های خروجی Bro به JSON را قالب‌بندی نموده و آن ها را بر روی یک Kafka topic قرار می‌دهد. پیام‌های JSON که توسط افزونه Bro منتشر شده‌اند به‌منظور تجزیه با توپولوژی تجزیه Metron Bro طراحی می‌شوند.

استخراج ابرداده DPI (قابلیت دید لایه ٧) پرهزینه است، بنابراین تنها برای پروتکل‌های انتخاب‌شده انجام‌پذیر می‌باشد. توصیه می‌شود که DPI برای پروتکل‌های DNS و HTTP فعال گردد. از این رو، وقتی کاوش‌گر PCAP هر بسته منفرد را روی سیم مشاهده کند، رکوردگذاری می‌کند و ابر داده DPI تنها برای زیرمجموعه‌ای از این بسته‌ها استخراج می‌گردد. همچنین Bro تنها ابزار استخراج DPI نیست. ابزار Qosmos، ابزار مشهور دیگری است که به زودی از آن پشتیبانی خواهد شد.



شکل ٤-١١: ضبط Bro در Meton

٤-١-٢-٩ Yet Another Flowmeter (YAF)

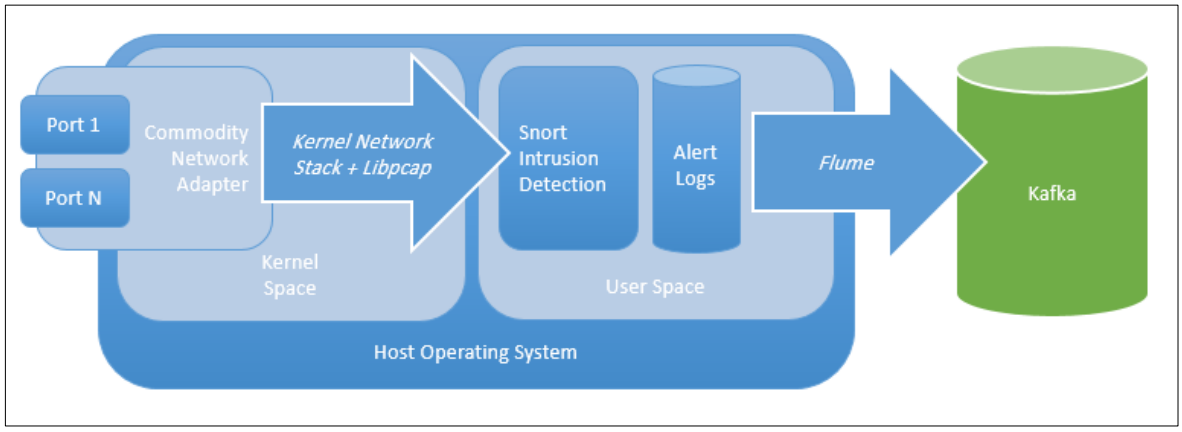
به‌دلیل محدودیت فضا و بار پردازشی بر روی تمام اجزای زیربنایی، هیچ‌کس تمایل به استفاده از PCAP ندارد. در حالی‌که Netflow جایگزینی برای PCAP نیست، اما خلاصه سطح بالایی از جریان شبکه را ارائه می‌دهد که در فایل‌های PCAP وجود دارد. در صورت عدم استفاده از PCAP، حداقل می‌توان Netflow را فعال نمود. Metron از YAF برای تولید داده‌های IPFIX (نسخه ١٠ پروتکل Netflow) از کاوش‌گر PCAP استفاده

[١٦] Deep Packet Inspection

می‌کند و خروجی آن به جای بسته‌های خام شبکه IPFIX خواهد بود. اگر Netflow به‌جای PCAP تولید گردد، داده‌های جریان Netflow به‌جای توپولوژی PCAP به توپولوژی تجزیه خواهد رفت.

در ابزار YAF ترافیک دو طرفه به دلیل لزوم کارایی در شبکه‌های بزرگ و گسترش یافته و نیاز به پایش ترافیک در لینک‌های با سرعت بالا، بسیار اهمیت دارد و امکان تحلیل را در هر دو جهت ارتباطی ایجاد می نماید و با حذف اطلاعات اضافی باعث افزایش در کارایی صدور داده می‌شود؛ در نتیجه ابزار فوق توانایی ضبط بسته‌ها به صورت زنده را داشته و جریان‌های دو جهته IPFIX را صادر می‌کند. این ابزار می‌تواند اطلاعات بسته IPFIX را به‌صورت قالب ASCII به دو روش در خروجی تولید کند: YAFAscii و yaf-file- yaf-file-mediator mediator می‌تواند فایل‌های IPFIX را با استفاده از libfixbuf خوانده[۱۷] داده جریان را به صورت یک فایل متنی چاپ کند و قادر است همه عناصر DPI استخراج شده را نشان دهد.

۱۰-۱-۲-٤ Snort



شکل ٤-۱۲: ضبط Snort

Snort یکی از مشهورترین سیستم‌های امروزی IDPS شبکه است. Snort ترافیک شبکه را پایش کرده و هشدارهایی را که براساس امضاها از قوانین عمومی تولید می‌شوند، ایجاد می‌نماید. Metron خروجی کاوش گر ضبط بسته را به Snort ارسال نموده و هر زمان که هشدارهای Snort به‌وجود آیند، Metron از Apache Flume به‌منظور استفاده از این هشدارها به Kafka topic بهره می‌برد. هنگامی‌که هشدارهای Snort به topic Kafka برسند، توسط توپولوژی تجزیه برداشته می‌شوند.

[۱۷] Read

٤-٣ قابلیت پشتیبانی

قابلیت پشتیبانی در Metron از سه جهت مطرح می‌شود:

- **منابع ذخیره سازی مورد پشتیبانی:** منابع ذخیره‌سازی که در ساختار Metron پشتیبانی می‌شود در جدول ٤-٢ بیان شده‌اند.

جدول ٤-٢: قابلیت پشتیبانی از منابع داده مختلف

منبع داده	دریافت انبوه	نرخ دریافت داده	نسخه
HBase	موجود نیست	موجود نیست	+ 2.0
HDFS	بله	بله	Hadoop 2.7.2
Elastic Search	بله	بله	Elastic Search 1.x

- **پشتیبانی از غنی‌سازی:** غنی‌سازی در سه حیطه اطلاعات جغرافیایی (Geo)، دارایی (Asset) و کاربر (User) قابل پشتیبانی است.
- **پشتیبانی از منابع هوشمندی تهدید:** که قبلاً به‌طور مفصل راجع به آن بیان شد.

٥ نصب و پیکربندی اولیه

٥-١ مبتنی بر Docker

هیچ یک از اجزای اصلی Metron به‌طور خودکار قابل نصب یا راه اندازی با پشتیبان‌های داکر نیستند. در این جا انتظار هیچ‌گونه تله‌متری را نباید داشت تا بتواند تجزیه، غنی‌سازی یا شاخص‌گذاری در آن انجام گیرد. چنان‌چه قصد آزمون یا اجرای دموی قابلیت‌های Metron را بر روی تک گره دارید، ماشین مجازی مبتنی بر Vagrant[١٨] گزینه مناسبی است. در موارد زیر می‌توان از آن به جای Vagrant استفاده کرد:

- محیطی که بتوان به سرعت آن را ساخت و راه‌اندازی کرد.
- نیاز به ساخت و شروع مجدد سرویس به‌صورت پی در پی داشت.

[١٨] Vagrant-driven VM

داکر شامل پشتیبان‌های زیر است که برای Metron سفارشی شده‌اند.

- Kafka (با Zookeeper)
- HBase
- Storm
- Elasticsearch
- Kibana
- HDFS

۵-۱-۱ نصب و پیکربندی داکر

نسخه‌های زیر داکر مورد آزمون قرار گرفته‌اند، که می‌توانید آن‌ها را نصب و راه‌اندازی نمایید:

- Docker version 1.12.0
- Docker-machine version 0.8.0
- Docker-compose version 1.8.0

دستورات زیر را گام به گام اجرا نمایید.

۱. ساخت Metron از طریق مسیر اصلی:

- $ cd $METRON_HOME
- $ mvn clean install -DskipTests

۲. ایجاد میزبان و منابع بیشتر با اسکریپت زیر:

- $ export METRON_DOCKER_HOME=$METRON_HOME/metron-contrib/metron-docker
- $ cd $METRON_DOCKER_HOME
- $ ./scripts/create-docker-machine.sh

۳. دستور بالا یک میزبان به نام "metron-machine" را ایجاد می‌کند. هرگاه که بخواهید دستورات داکر در این میزبان را اجرا کنید، مطمئن گردد که ابتدا مقادیر مربوط به محیط داکر تنظیم شده باشد:

- $ eval "$(docker-machine env metron-machine)"

٤. اگر از نصب docker-engine محلی استفاده می‌کنید، مقدار محیطی BROKER_IP_ADDR را به آدرس IP ماشین میزبان خود تنظیم نمایید.

روش استفاده:

کلیه مراحل زیر به صورت گام به گام طی خواهد شد.

۱. در مسیر اصلی نرم‌افزار:

- $ cd $METRON_DOCKER_HOME/compose/

۲. چرخه محیط داکر توسط دستور docker-compose کنترل می‌گردد. نام سرویس‌ها در فایل docker-compose.yml یافت می‌گردد. به‌عنوان مثال برای ساخت و شروع محیط دستور زیر اجرا می‌شود:

```
$ eval "$(docker-machine env metron-machine)"
$ docker-compose up -d
```

۳. دستور زیر وضعیت کلیه سرویس‌ها را نمایش می‌دهد:

```
$ docker ps --format 'table {{.Names}}\t{{.Status}}'
NAMES                STATUS
metron_storm_1       Up 5 minutes
metron_hbase_1       Up 5 minutes
metron_kibana_1      Up 5 minutes
metron_kafkazk_1     Up 5 minutes
metron_elasticsearch_1   Up 5 minutes
```

٤. سرویس‌های مختلف از طریق http بر روی میزبان داکر به صورت محلی نمایش داده می‌شود:

```
$ docker-machine ls
NAME ACTIVE DRIVER STATE URL SWARM DOCKER ERRORS
metron-machine * virtualbox Running tcp://192.168.99.100:2376 v1.12.5
```

۵. با فرض آن‌که آدرس IP میزبان 192.168.99.100 باشد، رابط کاربری و APIها به صورت زیر هستند:

- Storm - http://192.168.99.100:8080/
- HBase - http://192.168.99.100:16010/
- Elasticsearch - http://192.168.99.100:9200/_plugin/head/
- Kibana - http://192.168.99.100:5601/
- HDFS (Namenode) - http://192.168.99.100:50070/

٦. کلاس تجزیه‌کننده جدید اضافه می‌گردد:

```
$ cd $METRON_HOME
$ mvn clean install -DskipTests
```

۷. اجرای دستورات زیر به‌منظور پیاده‌سازی مجدد تجزیه‌کننده‌ها به پشتیبان Storm انجام می‌گیرد:

```
$ cd $METRON_DOCKER_HOME/compose
$ docker-compose down
```

```
$ docker-compose build storm
$ docker-compose up –d
```

۸. اتصال به container با دستور زیر انجام می‌شود:

```
$ cd $METRON_DOCKER_HOME/compose
$ docker-compose exec kafkazk bash
```

۹. ایجاد حس‌گر از طریق داده نمونه:

پس از ایجاد فایل داده در مسیر kafkazk/data/. و ساخت مجدد پشتیبان Kafka/Zookeeper داریم:

```
$ cd $METRON_DOCKER_HOME/compose
$ printf 'first test data\nsecond test data\nthird test data\n' > ./kafkazk/data/TestData.txt
$ docker-compose down
$ docker-compose build kafkazk
$ docker-compose up -d
```

دستور بالا فایل داده تست در container مربوط به Kafka/Zookeeper را پیاده می‌کند.

با دستور زیر داده‌ها در ماژول Kafa Topic جریان می‌یابد:

```
$ docker-compose exec kafkazk ./bin/produce-data.sh
Usage:  produce-data.sh data_path topic [message_delay_in_seconds]

# Stream data in TestData.txt to the 'test' Kafka topic at a frequency of 5 seconds (default is 1 second)
$ docker-compose exec kafkazk ./bin/produce-data.sh /data/TestData.txt test 5
```

پشتیبان Kafka/Zookeeper با داده Suid و Bro همراه می‌شود:

```
# Stream Bro test data every 1 second
$ docker-compose exec kafkazk ./bin/produce-data.sh /data/BroExampleOutput.txt bro
# Stream Squid test data every 0.1 seconds
$ docker-compose exec kafkazk ./bin/produce-data.sh /data/SquidExampleOutput.txt squid 0.1
```

۱۰. بارگذاری تنظیمات در Zookeeper:

```
$ docker-compose exec kafkazk bash
# $METRON_HOME/bin/zk_load_configs.sh -z localhost:2181 -m PUSH -i $METRON_HOME/config/zookeeper
# exit
$ docker-compose exec kafkazk bash
```

```
# $METRON_HOME/bin/zk_load_configs.sh -z localhost:2181 -m DUMP
# exit
```

۱۱. مدیریت توپولوژی:

امکان شروع و توقف هریک از توپولوژی‌ها با دستورات و اسکریپت‌ها وجود دارد که در زیر به چند نمونه اشاره می‌کنیم. استفاده از یک اسکریپت برای شروع توپولوژی‌های تجزیه‌کننده:

```
docker-compose exec storm ./bin/start_docker_parser_topology.sh sensor_name
```

شروع توپولوژی غنی‌سازی:

```
docker-compose exec storm ./bin/start_enrichment_topology.sh
```

شروع توپولوژی شاخص‌گذاری:

```
docker-compose exec storm ./bin/start_elasticsearch_topology.sh
```

توقف توپولوژی غنی‌سازی:

```
docker-compose exec storm storm kill enrichments -w 0
```

اجرای حس‌گر به‌صورت نقطه به نقطه:

```
$ cd $METRON_DOCKER_HOME/compose
$ docker-compose exec kafkazk ./bin/produce-data.sh /data/BroExampleOutput.txt bro
```

وارسی حس‌گر با استفاده از یک پیام در Kafka:

```
$ export METRON_DOCKER_HOME=$METRON_HOME/metron-contrib/metron-docker
$ cd $METRON_DOCKER_HOME/compose
$ docker-compose exec kafkazk ./bin/kafka-console-consumer.sh --zookeeper
localhost:2181 --topic bro
```

از بین بردن مصرف‌کننده (consumer) شروع توپولوژی تجزیه‌کننده Bro:

```
$ docker-compose exec storm ./bin/start_docker_parser_topology.sh bro
```

داده Bro بایستی از طریق توپولوژی تجزیه‌کننده bro و به topicهای محیط Kafka جریان یابد. Topic غنی‌سازی بایستی به‌طور خودکار ایجاد گردد:

```
$ docker-compose exec kafkazk ./bin/kafka-topics.sh --zookeeper localhost:2181 --list
bro
enrichments
indexing
```

وارسی داده تجزیه‌شده Bro در topic غنی‌سازی Kafka:

```
docker-compose exec kafkazk ./bin/kafka-console-consumer.sh --zookeeper localhost:2181
--topic enrichments
```

شروع توپولوژی غنی‌سازی:

```
docker-compose exec storm ./bin/start_enrichment_topology.sh
```

وارسی داده تجزیه‌شده Bro در topic شاخص‌گذاری Kafka:

```
docker-compose exec kafkazk ./bin/kafka-console-consumer.sh --zookeeper localhost:2181
--topic indexing
```

شروع توپولوژی شاخص‌گذاری:

```
docker-compose exec storm ./bin/start_elasticsearch_topology.sh
```

داده Bro غنی‌شده بایستی در Elasticsearch container قابل نمایش باشد:

```
$ docker-machine ls
NAME           ACTIVE   DRIVER      STATE     URL                         SWARM
DOCKER    ERRORS
metron-machine   *        virtualbox   Running   tcp://192.168.99.100:2376           v1.12.5

$ curl -XGET http://192.168.99.100:9200/_cat/indices?v
health status index                       pri rep docs.count docs.deleted store.size pri.store.size
yellow open   .kibana                     1   1          1            0      3.1kb          3.1kb
yellow open   bro_index_2016.12.19.18     5   1        180            0      475kb          475kb
```

۲-۵ استقرار Apache Metron

برای دسترسی به نسخه‌های مختلف Metron می‌توان به آدرس زیر مراجعه نمود:

```
http://metron.apache.org/documentation/#releases
```

می‌توان Metron را به سه روش خودکار راه‌اندازی نمود:

- Ansible براساس نصب ماشین مجازی تک گره Vagrant
- Ansible با ۱۰ گره کاملاً خودکار براساس نصب بر روی AWS با استفاده از Ambari Blueprints و AWS API ها
- نصب کامل خودکار Metron بر روی هر کلاستر تحت مدیریت Ambari

Metron در محیط‌های زیر قابل نصب است:

۱-۲-۵ Dev VM Install

Metron برای نصب بر روی میزبان مجازی قابل اجرا بر روی Virtualbox، نیاز به مؤلفه‌های زیاد و نصب کلیه آن‌ها بر روی میزبان منفرد خصوصاً مجازی‌سازی شده دارد. برای کار به اندازه کافی نیاز به ۸ گیگابایت حافظه دارد. دستورالعمل‌ها برای دو سیستم CentOS 6 و Ubuntu 14.04 در github.com موجود است. که در این گزارش توضیحات مبتنی بر Ubuntu 14.04 خواهند بود.

در سیستم کامپیوتری که Apache Metron پیاده‌سازی می‌شود مؤلفه‌های زیر بایستی نصب شده باشند:

- Ansible (2.0.0.2, 2.2.2.0, or 2.5.0)
- Docker
- Vagrant 2.0+
- Vagrant Hostmanager Plugin
- Virtualbox 5.0+
- Python 2.7
- Maven 3.3.9
- کامپایلر سازگار با C++ مانند GCC

اسکریپت زیر را برای تأیید اعتبار پیش‌فرض‌های بیان‌شده اجرا نمایید:

```
metron-deployment/scripts/platform-info.sh
```

نصب بر روی MacOS نیاز به پروژه Homebrew دارد. بنابراین مراحل زیر را بایستی گام به گام اجرا نمود:

۱. نصب Homebrew با دستورالعمل‌های سایت آن انجام گیرد.

۲. دستورات زیر را در ترمینال برای نصب ابزارهای مورد نیاز اجرا نمایید:

```
brew cask install vagrant virtualbox docker
brew cask install caskroom/versions/java8
brew install maven@3.3 git
pip install ansible==2.2.2.0
vagrant plugin install vagrant-hostmanager
open /Applications/Docker.app
```

پیاده‌سازی Metron:

۱. اطمینان حاصل شود که سرویس داکر در حال اجراست.

۲. Metron پیاده‌سازی گردد:

```
cd metron-deployment/development/ubuntu14
vagrant up
```

دستورات زیر نیز در ادامه فرآیند استقرار بدون نصب مجدد میزبان اجرا می‌شوند:

```
vagrant provision
```

بررسی Metron:

برای بررسی محیط جدید Aache Metron به منابع زیر مراجعه گردد:

- Metron Alerts
- Ambari

با اجرای دستور زیر از طریق SSH به میزبان متصل شوید:

```
vagrant ssh
```

روش کار با Metron:

علاوه بر اجرای مجدد کلیه منابع تأمین‌شده، شاید نیاز به اجرای مجدد برچسب Ansible فردی یا مجموعه‌ای از برچسب‌ها به روش زیر باشد. دستور زیر کلیه مؤلفه‌ها را نصب نموده و رابط کاربری آغاز به کار می‌کند.

```
./ run_ansible_role.sh web
```

یا

```
vagrant --ansible-tags="web" provision
```

استفاده از برچسب‌ها:

```
./run_ansible_role.sh "sensors,enrichment"
```

دستور فوق لیست کلیه برچسب‌ها را ارائه می‌دهد:

- hdp-install - Install HDP
- hdp-deploy - Deploy and Start HDP Services (will start all Hadoop Services)
- sensors - Deploy and Start Sensors.
- enrichment - Deploy and Start Enrichment Topology.

دستور زیر Vagrant را با برچسب environment اجرا می‌کند:

```
./run_enrichment_role.sh
```

ایجاد کل پروژه و اجرای آزمون‌ها:

```
$ mvn clean install
```

ایجاد بدون تست:

```
$ mvn clean install –DskipTests
```

ایجاد با پروفایل HDP:

```
$ mvn clean install -PHDP-2.5.0.0
```

ایجاد و اجرای گزارش‌گیری Metron:

```
$ mvn clean install
$ mvn site site:stage-deploy site:deploy
```

اجرای کد بدون آزمون:

```
$ mvn clean install -DskipTests site site:stage-deploy site:deploy
```

۲-۲-۵ نصب Metron نسخه 0.3.1 بر روی Ubuntu

در این بخش نصب Metron نسخه 0.3.1 بر روی Ubuntu یا Debian و ماشین مجازی تک گره با Vagrant و Ambari تشریح می‌گردد. همان‌گونه که در ابتدا توضیح داده شد دستورالعمل‌ها برای میزبان Ubuntu هستند و دستورات مشابه در CentOS نیز قابل اجرا هستند. این دستورات از Vagrant به‌منظور نصب ماشین مجازی تک گرهی پشتیبانی می‌کنند. این دستورات در سیستم‌عامل Debian نیز قابل اجرا بوده و کاملاً مشابه با Ubuntu است. نیازمندی منابع عبارتند از:

- اختصاص ۲۰ گیگابایت حافظه به ماشین مجازی Vagrant
- ۱۰۰ گیگابایت دیسک سخت
- ۲ عدد CPU

نصب با منابع کمتر ممکن است کار کند اما حافظه ناکافی ممکن است منجر به از کارافتادن برخی فرآیندها گردد. مراحل گام به گام نصب بدین شرح است:

۱. نصب pip:

```
sudo apt-get install python-pip python-dev build-essential
sudo pip install --upgrade pip
```

۲. نصب virtual box (Ubuntu، centos)

۳. ویرایش فایل sources. List:

```
sudo vim /etc/apt/sources.list
```

a) افزودن دو خط زیر در انتهای فایل:

```
# for virtual box
deb http://download.virtualbox.org/virtualbox/debian xenial contrib
```

b) ذخیره فایل و خروج

٤. نصب کلید عمومی اوراکل:

```
wget –q https://www.virtualbox.org/download/oracle_vbox_2016.asc-O- | sudo apt-key add
-
wget –q https://www.virtualbox.org/download/oracle_vbox.asc -O- | sudo apt-key add -
```

٥. نصب Oracle Virtualbox:

```
sudo apt-get update
sudo apt-get install virtualbox-5.0
```

٦. نصب جاوا:

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt update
sudo apt install oracle-java8-installer
```

٧. نصب Vagrant:

```
Version issue:
1.8.1 -> Plugin issue, can not download the plugin
1.8.5 -> Authentication issue while connecting to vmbox
Default apt-get install vagrant will give v 1.8.1
```

a) دانلود هر نسخه غیر از نسخه ۱٫۸٫۱ و ۱٫۸٫۲ با دستور زیر:

```
wget https://releases.hashicorp.com/vagrant/1.8.x (replace x with the desired version)
```

٨. نصب Maven

٩. نصب ansible:

a) تنها نسخه ۲٫۰٫۰٫۲ توسط metron پشتیبانی می‌شود.

b) در زمان نصب sible ممکن است خطاهای مربوط به بسته‌های رمزنگاری رخ دهد که با دستورات زیر قابل حل است:

```
sudo apt-get install python-pip python-dev libffi-dev libssl-dev libxml2-dev libxslt1-dev
libjpeg8-dev zlib1g-dev
For Centos this should probably solve it
```

c) برای CentOS:

```
sudo pip install -U setuptools
sudo pip install distribute
```

d) سعی کنید که دستور نصب ansible را اجرا نمایید. اگر کار نکرد از دستور زیر استفاده کنید:

```
sudo yum install gcc libffi-devel python-devel openssl-devel
```

e) در غیر این‌صورت از دستور زیر استفاده کنید:

```
yum search python | grep -i devel
```

f) سپس ببینید که کلیه بسته‌ها این مشکل را حل کرده اند یا نه.

g) مسیر جاوا و Maven را تنظیم کنید:

```
mvn -v in terminal should show the maven path
Edit /etc/environments
Add: JAVA_HOME="java path"
Add: MAVEN_HOME="maven path"
```

۱۰. پروژه را از مسیر زیر اجرا نمایید:

```
https://github.com/apache/metronNow build the packages and launch the Vagrant VM:
```

a) سپس از دستورات زیر استفاده کنید:

```
cd incubator-metron
mvn clean package -DskipTests
vagrant plugin install vagrant-hostmanager
cd metron-deployment/vagrant/(select the module you want to run)
vagrant up
```

۱۱. اگر موارد با شکست مواجه شد، دستور زیر را اجرا کنید:

```
vagrant provision (multiple times)
```

۱۲. هنگامی که موفقیت‌آمیز باشد بایستی نتایج زیر را دریافت نمایید:

- ماشین مجازی Vagrant به عنوان "node 1" اجرا گردد.
- Ambari بایستی در مرورگر با آدرس http://node1:8080/ در دسترس باشد.
- Kibana بایستی در مرورگر با آدرس http://node1:5000/ در دسترس باشد.
- Monit بایستی در مرورگر با آدرس http://node1:2812/ در دسترس باشد.

۱۳. اگر Monit کار نکرد، یا داده مورد نظر را تأمین نکرد، دستورات زیر را اجرا نمایید:

```
vagrant ssh
sudo vi (or nano or vim) /etc/hosts
```

a) هنگامی که فایل را باز کنید در خط بالای آن بایستی متن زیر را بیابید که صحیح می‌باشد.

```
127.0.0.1 localhost
```

۱٤. در صورتی‌که مراحل نصب به‌صورت کامل اجرا و بدون خطا باشد، سرویس‌دهنده بایستی به درستی اجر شود.

٦ مراجع

[1] https://metron.apache.org/current-book/index.html
[2] https://wiki.apache.org/incubator/MetronProposal
[3] https://hortonworks.com/blog/solving-cyber-security-quagmire-prism-big-data/
[4] http://www.hackinclude.com/introduction-cybersecurity-warheads/
[5] https://community.hortonworks.com/articles/26731/apache-metron-vs-opensoc-2.html
[6] https://github.com/apache/metron
[7] https://hortonworks.com/blog/20-questions-big-data-cybersecurity-experts-apache-metron-webinar-recap/
[8] https://cwiki.apache.org/confluence/display/METRON/About+Metron
[9] https://archive.apache.org/dist/metron/0.4.1/site-book/metron-deployment/vagrant/full-dev-platform/index.html