On High-Speed Flow-based Intrusion Detection using Snort-compatible Signatures

IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING(rank 7)

Oct 2020

Felix Erlacher (Paderborn University)

22 citation

روشهای تشخیص نفوذ مبتنی بر امضا فعلی در شبکه‌های با نرخ انتقال بالا مناسب نمی‌باشند. مانیتور جریان که بر مبنی استاندارد اطلاعات لایه اینترنت[1] ، در بحث جمع‌آوری اطلاعات بسته ها در یک جریان به خوبی عمل می‌کند. همچنین قادر خواهیم بود که از اطلاعات پیلود جریان ها نیز استفاده کنیم. و ما نیز روشی ارایه می‌دهیم که از جریانهای HTTP در لایه اپلیکیشن و پیلود آنها استفاده می‌کند. روشی که ارایه می‌دهیم یک سامانه تشخیص نفوذ مبتنی بر امضا با کمک IPFIX می‌باشد.FIXIDS از امضاهای HTTP مربوط به اسنتورت که بر مبنای IPFIX می باشد استفاده می کندو آنهارا برای فلوهای HTTP عبوری که IPFIX دارند بررسی می‌کند.FIXID دیتاهارا با نرخ بالاتری در مقایسه با snort و نرخ دورانداختن کم ارایه می‌دهد.

سیتم‌های NIDS امروزی بر دو پایه هستند: anomaly-based(behavior-based) , knowledge-based(signature/rule base) . مهمترین نوع از knowledge-based هاNIDS هستند.snort هم در این دسته قرار می‌گیرد که می‌تواند از DPI ها هم استفاده کند. به دلیل اینکه امضاهای مختلفی حتی بر مبنای regex و از محدوده بایت‌ها گرفته تا بسته‌های موجود در یک جریان قابل اعمال هستند، در شبکه‌های با نرخ انتقال بالا کارایی نخواهند داشت. اما یک روش جاگزین تشخیص بر مبنای جریان است که بسته‌هایی با ویژگی یکسان را جمع‌آوری کرده و سپس آن را به عنوان یک جریان بررسی می‌کند.[2]پروتکل IPFIX استاندارد اصلی برای جمع‌آوری اطلاعات بسته‌ها در قالب جریان برای پردازش‌های بیشتر می‌باشد. این که چه بسته‌هایی را در یک فلو قرار دهیم می‌تواند بر اساس یک اپلیکیشن خاص و با انتخاب درست Inforamtion Element ها باشد. ما نیز چون HTTP پروتکل مورد استفاده زیاد در اینترنت می‌باشد از IPFIX IE های بر مبنای آن استفاده می:نیم.

درFIXIDS به کمک امضاهای بر پایه HTTP و اعمال آنها به جریان‌های IPFIX که شامل اطلاعات HTTP می‌باشند یک روش تشخیص مبتنی بر جریان ارایه میدهیم که از روش DPI نیز سریعتر می‌باشند.همچنین در مقایسه با آن از قابلیت موازی سازی تسک‌های جداسازی بسته‌ها و انباشت آنها به عنوان یک جریان IPFIX با تسک آنالیز آنها بهره می‌برد. IPFIX نمی‌تواند روش‌های مبتنی بر DPI مثل snort را جایگزین کند و درواقع مکملی برای‌انهاست. همچنین روش‌ما برای ترافیک‌های رمز شده با استفاده از پروکسی های میانی نیز موفق خواهد شد و چون اطلاعات کمتری نسبت به DPI میخواهد، لذا حریم خصوصی کاربر نیز حفظ خواهد شد.

همچنین پس از بررسی متوجه خواهیم شد که روش ما نرخ گذر بالا و کارایی و دقت بالایی در تشخیص خواهد داشت.

---

[1] Internet Protocol Flow Information Export (IPFX)
[2] Flow Monitroing

روشهای قبلی ارایه شده برای افزایش سرعت عمل تطبیق امضا بر مبنای سخت افزار خاص یا نرم افزار(استفاده از الگوریتم‌های بهبود یافته) می‌بودند. تا به امروز روش‌های شناسایی مبتنی بر جریان تنها بر اساس اطلاعات سرآیند بسته‌ها بودند و یک حجم کمی از انواع مختلف حملات را تشخیص می‌دادند. و هیچکدام قابلیت تشخیص بر مبنای پیلود بسته‌ها و استفاده از امضاهای تعریف شده شخصی کاربران را نداشتند.

معرفی سیستم: از همان مجموعه امضاهای به روز پیش فرض خود snort استفاده می‌کنیم. مشخص است که FIXIDS تنها برای رولهای مبتنی بر HTTP کاربرد خواهد داشت. رولهای دیگر نیز در مرحله پردازش اولیه کنارگذاشته خواهند شد. همچنین snort به دنبال الگوهای محتوا در پیلود ترافیک‌ها نیز می‌گردد. snort قابلیتی به نام content modifier را در اختیار ما قرار می‌دهد که با استفاده از آن الگوها را می توان در محدوده کمتری و با سرعت بیشتری در پیلودها جستجو کرد. و اینها اکثرشان کاری می‌کنند که اسنورت در محدوده فیلدهای HTTP عمل کند.

Table 1
Snort content modifiers and their correspondig IPFIX IE

Content modifier		HTTP IE	IANA IE ID
http_method	→	httpRequestMethod	459
http_uri	→	httpRequestTarget	461
http_raw_uri	→	httpRequestTarget	461
http_stat_code	→	httpStatusCode	457
http_stat_msg	→	httpReasonPhrase	470

Table 2
Modifiers for pcre content definitions supported by FIXIDS

Modifier	Description
i	pcre pattern searches are by default case sensitive; this turns case insensitive pattern matching on
U, I	The pcre pattern search is applied to httpRequestTarget
M	The pcre pattern search is applied to httpRequest-Method
S	The pcre pattern search is applied to httpStatusCode
Y	The pcre pattern search is applied to httpReasonPhrase

در FIXIDS الگوهای محتوا می توانند text یا دیتای باینری به شکل هکزادسیمال باشند و همچنین قابلیت پشتیبانی از regex را نیز خواهد داشت. رولهای مخصوص کاربر که برای اپلیکیشن‌های خاص نیز می‌تواند استفاده شود را نیز پشتیبانی می‌کند.

پیاده سازی: برای پیاده سازی FIXIDS از ابزار متن باز مخصوص مانیتور شبکه Vermont استفاده می کنیم که FIXIDS به عناون یک ماژول به آن اضافه شده است.

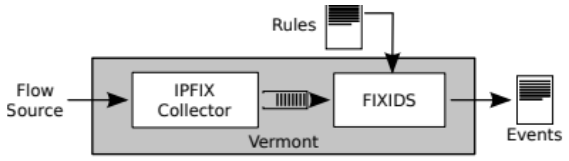

Figure 1. Minimal Vermont configuration with FIXIDS functionality

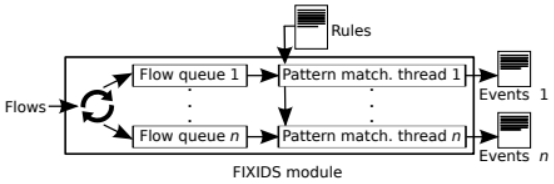

Figure 2. Sketch of the internals of the FIXIDS module

یک IPFIX Exporter خارجی جریان های IPFIX حاوی IE های HTTP را به Vermont می فرستد (در حال حاضر TCP، SCTP، UDP و DTLS روی UDP و SCTP پروتکل های انتقال پشتیبانی می‌شوند). در Vermont، ماژول IPFIX Collector جریان‌ها را دریافت کرده و از طریق بافر جریان به ماژول FIXIDS تحویل می‌دهد. در هنگام راه اندازی، ماژول FIXIDS همه امضاها و تنظیمات را از فایل قوانین ارائه شده در پیکربندی parse می کند. و سپس در زمان اجرا IE هر یک از جریان‌های عبوری را با الگوهای امضای هر قانون مقایسه می‌کند. همچنین همانطور که در شکل ۲ نشان داده شده است ما از یک قابلیت تطابق الگوی موازی نیز بهره می‌بریم. فقط باید توجه داشت که هر هسته پردازنده به صورت اختصاصی توسط یک ماژول vermont زیر بار زیاد است و لذا نباید برای تطابق الگو از آنها استفاده کرد. هر صف به صورت FIFO پیاده سازی شده است و برای مقایسه الگوهای رشته ای هر قانون (همه قانونها از فایل مربوطه parse شده‌اند و در مموری قرار گرفته‌اند) با IE ها از تابع

strstr() استفاده می‌کند(این تابع به دلیل این که به صورت اسمبلی پیاده سازی شده است و از ثبات‌های پردازنده استفاده می کند از بقیه مقایسه گرهای رشته‌ای عملکردی بهتر خواهد داشت). اگر یکی از الگوهای تعیین شده در رولها نقض شود، بقیه نیز بررسی نخواهند شد. اگر تمامی الگوها تطبیق پیدا کنند، یک هشدار داده خواهدشد و اطلاعات رویداد در فایل مربوط به نخ ثبت خواهد شد و تطابق الگو قوانین باقی مانده (به غیر از HTTP) ادامه خواهد یافت.

طریقه راه اندازی: در این قسمت به نرم افزارهای مورد نیاز و نحوه تنظیم آنها، قوانین و ترافیک مورد استفاده می پردازیم:

- Snort :event ها آن را با event های تولید شده توسط FIXIDS مقایسه می‌کنیم. تنها تغییری که بر روی تنظیمات پیشفرض انجام داده ایم افزایش سایز صفها بوده است، چون در غیر اینصورت اگر تعداد event ها به ازای هر بسته از طول صف بیشتر شود، دیگر گزارشی داده نمی‌شود.
- IPFIX-based Signature-based Intrusion Detection System(FIXIDS) : ماژول مربوطه در Vermont می‌باشد. روی یک پورت به جریان‌های موردنظر گوش می‌دهیم. هر چه تعداد هسته های فیزیکی اختصاص داده‌شده به نخ‌های مربوط به تطابق الگو افزایش یابد، دقت و سرعت بالاتری خواهیم داشت.(اما hyperthreading تاثیر به سزایی ندارد)
- Vermont Flow Probe: از این ماژول vermont برای انباشته کردن بسته‌ها به جریانهای IPFIX استفاده می‌کنیم. توجه شود که Vermont به گونه‌ای تنظیم شده است که تنها ترافیک‌های HTTP را export کند. تشخیص ترافیک HTTP نیز با بررسی صحت سرایند بسته‌ها انجام می‌شود. همچنین فیلدهای HTTP IE جریان‌ها که استخراج می کند برای بررسی تشخیص نفوذ از مهمترین هاش URI می‌باشند که طبق کانفیگی که انجام داده‌ایم ۱۵۰ بایت اولیه آن را استخراج می‌کند.
- Nprobe Flow Probe: به عنوان نمونه و به منظور بررسی تطبیق پذیری IPFIX با flow probe های دیگر استفاده می‌شود. منظورمان flow exporter های دیگر به غیر از vermont، که ترافیک IPFIX تولید می‌کنند، می‌باشد
- Network Setup: سه تا workstation داریم که اولی و سومی به دومی متصل شده اند با لینک مسبتقیم که لذا از سربار مسیریابی جلوگیری شود.
- مجموعه قوانین تشخیص: برای هردو FIXIDS و snort از یک مجموعه قوانین مربوط به HTTP استفاده می‌کند.
- ترافیک مهاجم: ترافیک‌های موجود حجم خوبی از حملات را ندارند و همچنین پیلودها شامل داده‌های حریم خصوصی نمی باشند. لذا یک روش اینست که از ترکیب چندین دیتاست با هم یک دیتاست شخصی بسازیم و از ابزارهایی مثل متااسپلویت نیز بدین منظور استفاده کنیم. اما بسیار زمانبر خواهد بود. راه‌حل استفاده از فریمورک GENESIDS می‌باشد. می‌توان با آن ترافیک های HTTP شخصی‌سازی شده با فرمت snort تولید کرد.
- ترافیک واقعی: از CISCO Trex استفاده می‌کنیم که قابلیت تولید پیلود سطح اپلیکیشن را نیز دارا می‌باشد. روش ما برای افزایش ترافیک عبوری شبکه، افزایش Connection Per Second به جای Packets Per Second می‌باشد.

ارزیابی عملکردی: دراینجا قصد داریم ارزیابی الگوریتم‌های استفاده‌شده در FIXIDS و بررسی دقت آنها در شناسایی حجم وسیعی از حملات را در مقایسه با snort را انجام دهیم . نتیجه می‌شود هردو دقت بالای ۹۰ درصد خواهند داشت.

ارزیابی کارایی FIXIDS: دقت تشخیص و میزان بسته‌های دور انداخته شده در زیر نرخ پهنای‌باند بالا را بررسی می‌کنیم.با Export های مختلف آن را تست کرده ایم و درآخر نیز با snort مقایسه کرده‌ایم.

در آخرین مرحله نیز بررسی می‌کنیم که آیا FIXIDS امکان پیاده سازی در محیطی که از قبل snort در آن موجود می باشد را دارد یا نه تا بدین منظور میزان بار روی snort را کاهش دهد. به این صورت عمل می‌کند که قوانین مربوط به HTTP به FIXIDS و قوانین دیگررا به snort محول می کنیم و برای ارزیابی نیز تغییرات نرخ گذر را بررسی می‌کنیم.

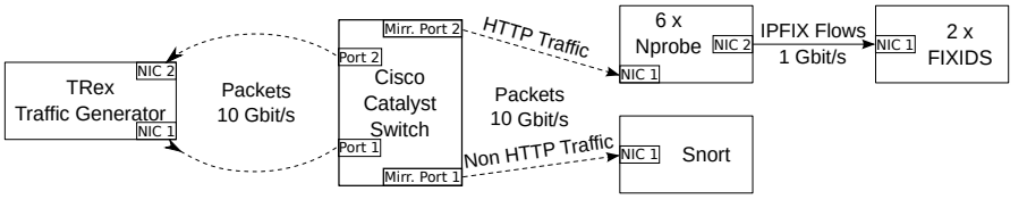

Figure 13. Realistic application scenario; FIXIDS analyzes all HTTP traffic and Snort analyzes all non HTTP traffic. This way Snort processes few rules and less traffic.