تشخیص فوری حملات DDoS با استفاده از NetFlow در IOS دستگاه های سیسکو

Immediate detection of DDoS attacks with using NetFlow on Cisco devices IOS

نویسنده: دکتر محمود رضا طهماسب پور
مشاور ارشد سخت افزار و زیرساخت فناوری اطلاعات بانک تجارت
دکترای علوم کامپیوتر و فناوری اطلاعات - دانشگاه مالایا (UM)
mahmoudtahmasebpour@live.com

چکیده

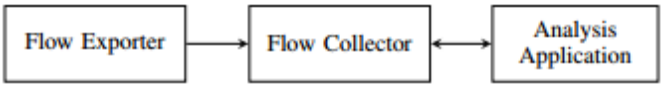
تشخیص جریان حملات DDoS به طور معمول توسط برنامه های کاربردی تجزیه و تحلیل کننده که در سیستم نصب شده اند یا نزدیک به جمع کننده جریان هستند انجام می شود. اگر چه این روش اجازه استقرار آسان را می دهد، ولی تشخیص غیر آنی و حساس به حملات DDoS را می سازد که دلایل آن عبارتند از اول، واقعیت این است که فرآیند ارسال جریان بر اساس وقفه است که گردآورنده جریان به طور معمول داده ها را برای برنامه کاربردی تجزیه و تحلیل در تکه هایی تهیه می کند که باعث تاخیر چند دقیقه ای در جهت تشخیص می گردد. دوم، ترافیک حمله ممکن است بوسیله فرآیند ارسال جریان تقویت شود اگر بسته های اصلی به اندازه کافی کوچک و بخشی از جریان های کوچک باشند.

در این مقاله تحقیقی نشان داده می شود چگونگی تشخیص حملات DDoS بر روی ارسال کننده جریان بجای گردآوری جریان، یعنی نزدیک به منبع داده و در مد آنی، که با این حال دسترسی به یک جریان زیرساختی نظارت با قابلیت توسعه را لازم دارد. در این تحقیق، به بررسی اینکه آیا ممکن است عمل سیستم تشخیص در همان پلتفرم شبکه IOS دستگاه های سیسکو به طور گسترده مستقر گردد. از آنجا که هدف نهایی تحقیق شناسایی حضور حمله مهاجمان و اهداف آن است، از NetFlow استفاده می گردد. در این زمینه، تشخیص نمونه اولیه حمله DDoS نشان داده است که برای تولید یک بار ثابت بر روی پلتفرم های زمینه، حتی زیر حملات، تاکید است که تشخیص حمله DDoS می تواند در یک دستگاه سیسکو مدل Catalyst 6500 در شبکه های تولید انجام شود، اگر ظرفیت مازاد کافی در دسترس باشد.

۱- مقدمه

حملات DDoS در حال تبدیل شدن به یک تحدید بزرگ فنی و اقتصادی، اضافه بار شبکه ها و سرورها با مقادیر زیادی از ترافیک شبکه هستند. در اوایل سال ۲۰۱۴، سایت CloudFlare بوسیله یک حمله سیل آسا UDP تقویت شده با رسیدن به نزدیک به ۴۰۰ گیگابیت در ثانیه ترافیک در پهنای باند آن مختل شد [۱]. اگر چه حملات سیل آسا UDP به طور معمول با هدف اضافه بار از طریق ازدحام زیاد بایت دارند، همچنین حملات دیگری مانند حملات سیل آسا TCP SYN وجود دارند که حاصل از تعداد زیاد اتصالات هستند. با تعریف یک جریان، "مجموعه ای از بسته های عبوری از یک نقطه شبکه در طول یک بازه زمانی خاص، به طوری که تمامی بسته های متعلق به یک جریان خاص مجموعه ای از خواص مشترک دارند" [۲]، این نیز نتیجه تعداد

زیادی جریان است. این امکان را برای فن اوری های مبتنی بر جریان برای تشخیص چنین حملات مبتنی بر حجم می سازد [۳]. علاوه بر این، استفاده از فن آوری های صادر کردن جریان، مانند NetFlow و به تازگی استاندارد IETF تلاش برای IPFIX، به طور خاص مفید برای تولید مجموعه ترافیک هستند. این رویکرد باعث کاهش مقدار داده ها به طور قابل توجهی برای تجزیه و تحلیل [۴] و همچنین قدرت پردازش لازم برای صادر کردن و گردآوری می شود. علاوه بر این، این فن آوری ها به طور گسترده ای در دستگاه های ارسال بسته موجود است و داده های جریان را به راحتی در دسترس برای استقرار در شبکه ها موجود می سازد.



شکل ۱ – نمونه معماری نظارت جریان

به طور کلی تشخیص نفوذ مبتنی بر جریان، به طور سنتی توسط برنامه های کاربردی تجزیه و تحلیل انجام می شود به غیر از حمله DDoS [۵] و [۷]، همان طور که در شکل ۱ نشان داده شده است. این برنامه های کاربردی بر روی داده های جریان صادر شده بوسیله صادر کنندگان جریان و گردآوری توسط گردآورندگان جریان عمل می کنند. از آنجا که صادرات داده های جریان به شدت مبتنی بر وقفه زمانی است و گردآوری اغلب برای کار در فواصل زمانی چند دقیقه ای طراحی شده است، برنامه های کاربردی تجزیه و تحلیل در فرآیند تشخیص تاخیرهای مختلف را دارا هستند [۸]. به خصوص در مورد تشخیص حمله DDoS، که در آن می تواند بار اضافی در زیرساخت شبکه خیلی سریع اتفاق افتد، این همان چیزی است که باید از آن اجتناب کرد.

کار اخیر نشان داده است که حرکت تشخیص نزدیکتر به منبع داده، باعث کاهش به طور قابل توجهی در تاخیرهای تشخیص از حداقل ۱۰ ثانیه تا ۱۶۵ ثانیه می شود [۹]. ارائه الگوریتم تشخیص حمله DDoS بر روی یک پلتفرم هدف با داده های غیر فعال مبتنی بر صادرات بر اساس جریان ها، یعنی پلتفرم INVEA-TECH's FlowMon قابل اجرا است. هدف از این مقاله بررسی این است که آیا الگوریتم تشخیص ارائه شده در [۹] را می توان در یک پلتفرم شبکه در دسترس گسترده مستقر نمود. در این زمینه، پلتفرم IOS سیسکو را هدف قرار داده شده و به ویژه مدل Catalyst 6500، که یکی از دستگاه های ارسال بسته به طور گسترده مستقر است [۱۰]. بویژه در تجربه عملیات انجام تشخیص نفوذ در دستگاه های ارسال بسته در شبکه های تولید تمرکز می گردد.

باقی مانده مطالب این مقاله به این شرح است. بخش دوم، اصطلاحات مربوط به NetFlow و IPFIX که در سراسر این مقاله استفاده شده است. یک نمای کلی از الگوریتم تشخیص اصلی از [۹] در بخش سوم داده شده است. در بخش چهارم، چگونگی اطلاعات نظارت مورد نیاز توضیح داده شده است، که به عنوان ورودی به الگوریتم تشخیص بکار رفته را می توان از IOS سیسکو بدست آورد. نمونه اولیه اجرا شده در بخش پنجم مورد بحث قرار می گیرد که برای اعتبار سنجی ارائه شده در بخش ششم استفاده خواهد شد. در بخش هفتم، امکانات بیشتر برای تشخیص حمله DDoS و کاهش مخاطرات در IOS سیسکو توضیح داده شده است. در نهایت، نتایج بدست آمده در بخش هشتم طرح گردیده است.

۲- صادر کردن و اندازه گیری جریان

در این بخش، اصلاحات مربوط به صادر کردن و اندازه گیری جریان معرفی می گردند که در طول این مقاله مورد استفاده قرار خواهند گرفت. برای یک مرور جامع از NetFlow و IPFIX، به آموزش در [۴] مراجعه شود.

صادر کردن و اندازه گیری جریان دو وظیفه انجام شده توسط یک صادر کننده جریان هستند [۴]، همان طور که در شکل ۱ نشان داده شده است. بسته ها در شبکه داخل جریان ها بوسیله فرآیند اندازه گیری گردآوری شده اند. هنگامی که یک جریان

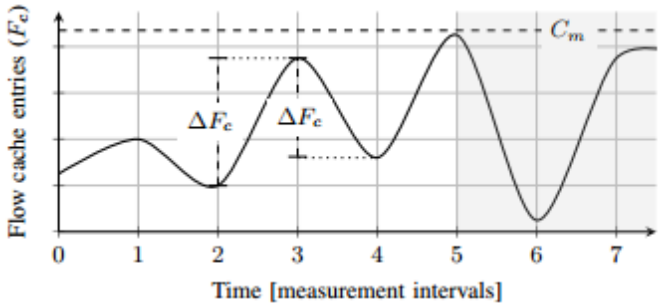
جدید مشاهده می شود، برای ورود به این جریان یک جریان نهان شده ایجاد می شود. این چیز نهان شده یک جدول است که اطلاعات را در جریان های فعال شبکه ذخیره می کند [۴]. گذشته از کلید جریان، یعنی زمینه های شناسایی یک جریان، برخی اطلاعات اضافی معمول، از قبیل تعداد بسته ها و بایت ها در یک جریان حساب شده اند. به این رویداد که در آن چیز نهان شده کامل است و یک جریان نهان شده ورودی نمی تواند ایجاد شود مراجعه شود، که می تواند در طول دوره ها از ترافیک بالا اتفاق بیافتد اگر جریان نهان شده زیر ابعاد، به عنوان یک خطای یادگیری جریان باشد [۱۱]. هنگامی که یک جریان نهان شده ورودی منقضی می شود، برای مثال زمانی که جریان فعال و یا بیکار برای مدت طولانی بوده است یا به دلیل محدودیت منابع، یک سابقه جریان صادر می شود، یعنی آن را در یک پیام NetFlow یا IPFIX قرار داده و به یک گردآورنده برای ذخیره سازی و پیش پردازش ارسال می شود.

۳-الگوریتم تشخیص

یک الگوریتم موجود را برای برآوردن الزاماتی که کم وزن، دقیق و در زمینه تشخیص حمله DDoS آنی بودند بکار گرفته می شود، در [۹] شرح داده شده است. الگوریتم فوق بر روی یک بازه زمانی ثابت و با اندازه گیری تعداد جریان نهان ایجاد شده ورودی، به عنوان اندازه های قابل استفاده در بیش از چهار معیار ارائه شده قابل اجرا است [۹]. بر اساس این اندازه گیری، یک پیش بینی برای اندازه گیری مقدار فاصله بعدی ایجاد شده است. تعداد جریان نهان ایجاد شده ورودی در مقایسه با مقادیر اندازه گیری گذشته بیش از حد بالا است، نمونه اندازه گیری در نظر گرفته شده غیر عادی است. با این حال، به دلیل ترافیک اینترنت که الگوی روزانه، مانند افزایش قوی و کاهش در تعداد جریان نهان ایجاد شده در شروع و پایان یک روز کاری به ترتیب نشان می دهد، الگوریتم فوق نیز رفتار عادی شبکه را در دوره ۲۴ ساعته می آموزد. ارزش پیش بینی به این صورت تعریف می شود:

$$\hat{x}_{t+1} = b_t + s_t \qquad \text{(معادله ۱)}$$

که در آن $\hat{x}_{t+1}$ ارزش پیش بینی برای فاصله بعدی است، b_t جزء پایه، گاهی اوقات به عنوان جزء دائمی معرفی می شود، که نشان دهنده روند ترافیک اینترنت و S_t جزء فصلی که نشان دهنده الگوی روزانه است.



شکل ۲ – جریان ورودی نهان ایجاد شده در IOS سیسکو روی زمان

چند پیشرفت برای این الگوریتم در [۹] بحث شده است. اول، به منظور کاهش استفاده حافظه، از ارزش حفظ الگوهای فصلی استفاده می شود. S_t ، در هر ساعت ذخیره می شود و برای تخمین مقدار برای یک زمان معین داخل می شود. دوم، برای جلوگیری الگوریتم فوق از یادگیری الگوهای ترافیکی مخرب، مقدارهایی مانند S_t و b_t در هنگام یک حمله دور ریخته می شوند. آخر، از الگوهای ترافیکی در تعطیلات پایان هفته به طور معمول متفاوت از الگوهای در طول روزهای هفته، یک فرق بین تعطیلات آخر هفته و روزهای هفته برای حافظه فصلی ساخته شده است. این نتایج در دو الگوی تمرینی، یکی برای روزهای هفته و دیگری تعطیلات آخر هفته می باشد.

۴- نظارت اطلاعات موجود در IOS

الگوریتم تشخیص در این مقاله که در بخش سوم در نظر گرفته شده است، در یک اندازه واحد است، یعنی تعداد جریان نهان شده ورودی در هر بازه زمانی ایجاد می شود. این اندازه به راحتی در دسترس بر روی پلتفرم نظارت جریان در کار اصلی از نمونه اولیه (INVEA-TECH's FlowMon [۹])، می باشد. از آنجا که پلتفرم با توسعه پذیری ذهنی طراحی شده است، این اطلاعات به طور مستقیم در دسترس از پلتفرم API است. با این حال، میزان اطلاعات موجود در IOS به شدت وابسته به مسیر بسته است، یا به جریان در روتر و سوئیچ بستگی دارد. به طور دقیق تر، بسته ها هم به صورت سخت افزاری و هم نرم افزاری سوئیچ می شوند، اگر چه بسیاری از بسته ها سخت افزاری سوئیچ می شوند. به طور مثال در شبکه دانشگاه Twente، ۹۹,۶ درصد از ترافیک مربوط به سوئیچ سخت افزاری است [۱۱]. علت این که بسته به صورت نرم افزاری سوئیچ می شود، تکه تکه شدن بسته ها است، بسته ها به مقصد دستگاه خودشان ارسال می شوند، به طور مثال بسته هایی که نیاز به ARP دارند [۱۲]. برای جریان پردازش سخت افزاری، اطلاعات بر روی تعدادی از جریان نهان ایجاد شده ورودی به طور مستقیم در دسترس نیست. برای نزدیک شدن به این متریک، از اطلاعات در دسترس از متریک جریان و پردازش صادرات استفاده می شود:

- تعداد ورودی های جریان نهان شده (F_c).
- تعداد رکوردهای جریان سوئیچ نرم افزاری صادر شده (F_e).
- تعداد خطاهای جریان یادگیری (F_f). این متریک به جای جریان در بسته ها قرار دارد.

تعداد جریان نهان ایجاد شده ورودی از آخرین اندازه گیری را می توان با استفاده از تعریف زیر تخمین زد:

$$F = \Delta F_c + \Delta F_e + \Delta F_f / c_f \quad \text{(معادله ۲)}$$

هنگامی که جریان نهان شده ورودی صادر می شود، F_c کاهش خواهد یافت که باعث تقریب با دقت کمتر خواهد شد اگر فواصل اندازه گیری خیلی طولانی باشند. برای مثال، در شکل ۲، اگر اندازه گیری برای پوشش دو بازه باشد، از $t = 2$ تا $t = 4$ ، ΔF_c در پیک $t = 3$ در نظر گرفته نخواهد شد. بوسیله نمونه گیری از F_c در فواصل بیشتر، می توان تغییرات را با دقت بیشتری مشاهده نمود، به طوری که مثبت ΔF_c در $t = 3$ و منفی ΔF_c در $t = 4$ مشاهده می گردد، که توسط صادرات ایجاد می شود. سپس، اگر ΔF_c منفی باشد، یک تخمین از مقادر قبلی ΔF_c به جای آن استفاده می شود. هنگامی که جریان نهان شده در حال نزدیک شدن به حد ظرفیت خود است، صادر کننده یک انقضای اضطراری را انجام می دهد [۴]. در شکل ۲ این موضوع در منطقه سایه دار به تصویر کشیده شده است. همچنین F_c به C_m ظرفیت جریان نهان شده می رسد که می بایست از جریان نهان شده ورودی منقضی بشود. اگر یک اندازه گیری بین $t = 6$ و $t = 7$ انجام شود، الگوریتم ممکن است آن را به عنوان یک حمله برای یک فاصله اندازه گیری با توجه به افزایش گسترده در تعداد نهان شده ورودی نسبت به $t = 6$ شناسایی کند. برای مقابله با آن، اجرای شناسایی، منتظر اندازه گیری بعدی می ماند اگر آن را یک حمله مشکوک اعتبار دهد بجای یک حمله واقعی. این مسئله باعث ایجاد تاخیر در تشخیص می شود.

از آنجا که تعداد ورودی در جریان نهان شده (F_c) تنها در مورد جریان سوئیچ سخت افزاری است، تعدادی تغییر جریان صادرات نرم افزاری (F_e)، که می تواند به طور مستقیم از IOS به دست آورد را اضافه می گردد. در نهایت اضافه کردن F_f اجازه داده می شود برای مورد جریان ها که باید ایجاد شده باشند اما نشده اند، بویژه در مورد حملات DDoS با شدت بالا. برای مثال، برای جبران این واقعیت که F_f در بسته ها بیان شده است که متریک های دیگر در جریان ها بین می کنند، F_f تقسیم می شود بوسیله متوسط تعداد بسته ها در جریان، توسط cf که در معادله ۲ ارائه شده است.

۵- اجرا

EEM بخشی از IOS سیسکو است که تشخیص رویدادهای شبکه را بصورت آنی بکار می برد، تعریف سیاست ها را اجازه می دهد که می تواند مورد استفاده برای اجرای یک اپلت یا اسکریپت وقتی که رویدادها حادث می شوند بکار برده می شود. برای مثال در زمانی که میزان بار شبکه به حد خاص می رسد یا تغییراتی در مسیر شبکه رخ می دهد به مدیران شبکه می توان ایمیل ارسال نمود. یک نوع دیگر رویداد مبتنی بر زمان است. این رویداد می تواند در میان دیگر رویدادها در فواصل زمانی ثابت برنامه ریزی شود. در این کار، با استفاده از دو سیاست مبتنی بر زمان اجرا می شود همچون TCL اسکریپت ها:

- سیاست اندازه گیری: تعیین مولفه اول برای برآورد متریک مبتنی بر جریان. تعداد ورودی های جریان نهان شده (F_c)، در بخش چهارم توصیف شد.
- سیاست تشخیص: بازیابی اجزای باقی مانده. تعداد جریان های نرم افزاری صادرات (F_e) و تعداد خطاهای یادگیری جریان (F_f). همچنین پیاده سازی الگوریتم تشخیص حمله DDoS واقعی.

برای به دست آوردن هر سه مولفه، که همه با استفاده از SNMP موجود ساخته شده اند، با استفاده از ویژگی های محیط EEM که دسترسی به اشیاء SNMP را فراهم می کند. دلیل تقسیم سیاست اندازه گیری از سیاست تشخیص این است که نیاز به وضوح بیشتر برای تشخیص دقیق تر تغییرات وجود دارد که در بخش چهارم توصیف شد.

فراخانی سیاست ها از طریق حافظه است و چون می خواهیم داده ها را به اشتراک بگذاریم که هر دو بین سیاست و سیاست ها اجرا می شوند، یک روش برای به اشتراک گذاری داده ها نیاز به اجرا دارد. با توجه به این واقعیت که سیستم فایل مبینی بر فلش است به طور معمول از اقدام نوشتن بیش از حد که باعث کوتاه شدن طول عمر حافظه می شود جلوگیری شود. بنابر این محیط EEM یک کتابخانه متن برای این منظور ارائه می دهد؛ آن برای ذخیره سازی تغییرات TCL به حافظه به جای نوشتن آنها روی دیسک اجازه می دهد. علاوه بر این برای نگهداری تراک از داده ها بین سیاست اجرا می شود، همچنین از این ویژگی برای تبادل اطلاعات بین دو سیاست استفاده می شود، به عنوان نتیجه سیاست اندازه گیری بوسیله سیاست تشخیص مورد نیاز می باشد.

دو سیاست بحث قبل توسط EEM در فواصل مربوطه خود که بر اساس زمان اجرا می شوند از سیاست های مربوطه انتخاب می شوند. هنگامی که سوئیچ با این حال زیر بار سنگین استفاده شود از سیاست CPU قوی تر استفاده سی شود. برای جلوگیری از این سیاست ها از پرش اجرایی وقتی که سیاست بیش از طول نمونه اولیه فاصله دارد با بهره گیری از EEM می تواند حداکثر

زمان اجرا را تنظیم نماید. اگر زمان اجرا بیش از حد طول بکشد سیاست خاتمه اجباری باعث از دست رفتن اطلاعات می شود. در مورد سیاست تشخیص، الگوریتم شروع کردن دوباره را با فاز یادگیری برای همه داده های از دست رفته اقدام می کند. اگر سیاست های اندازه گیری قبل از موعد مقرر خاتمه یابد، تعداد اندازه گیری ورودی های جریان نهان ایجاد شده کمتر خواهد شد، آن را به عنوان یک اندازه گیری از دست رفته که کمی تحت تاثیر دقت و صحت الگوریتم قرار خواهد گرفت. برای جلوگیری از از دست رفتن سیاست تشخیص، یک حاشیه به فاصله اضافه شده است که اجازه می دهد آن را دیگری اجرا کند اگر لازم باشد، اما بیش از آن فاصله که در آن اجرا شده است نیست. متوسط زمان اجرای سیاست های تشخیص بین ۲ تا ۳ ثانیه در شرایط عادی است و در شرایط فشار بین ۷ تا ۸ ثانیه خواهد بود. بنابر این فاصله نهایی انتخاب شده برای سیاست تشخیص ۱۰ ثانیه است. برای سیاست های اندازه گیری، اندازه گیری نشان داده می شود که ۲ ثانیه تعادل مطلوب بین اندازه گیری های دقیق و از دست رفتن داده ها برای خاتمه دادن فراهم می کند.

۶- اعتبار

در این بخش، اعتبار این کار توصیف می گردد، شروع بوسیله شناسایی نیازمندیها در بخش ششم قسمت الف، بعد از آن یک توصیف از راه اندازی اعتبار و همچنین جزئیات در مورد استقرار در قسمت ب و در نهایت نتایج را در قسمت ج مورد بحث قرار می گیرد.

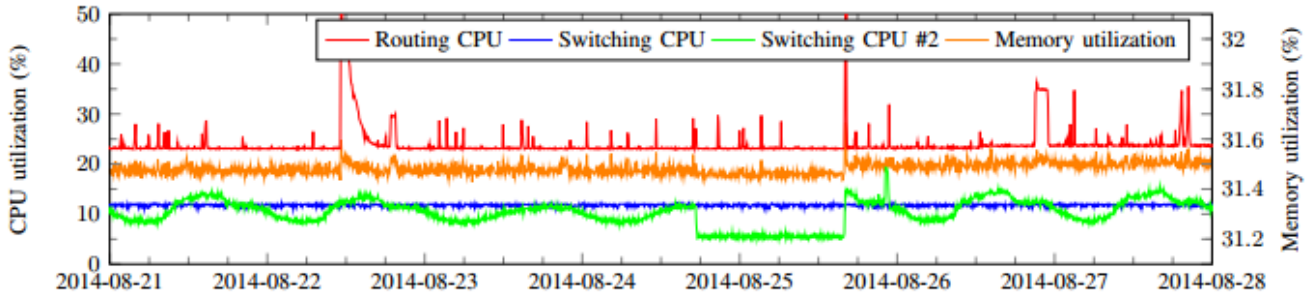
الف) نیازمندیها

سه الزام مورد نیاز برای الگوریتم تشخیص اصلی تعریف شده اند: ۱- آن باید بسیار سبک از نظر CPU و حافظه استفاده باشد، ۲- دقت باید به اندازه کافی بالا برای تعیین تعداد کم نادرست مثبت و منفی باشد، ۳- تاخیر تشخیص باید تقریبا ۱۰ درصد از روش های تشخیص نفوذ معمولی [۹] کمتر باشد. با این حال از آنجا که دستگاه سیسکو مدل Caralyst 6500 با سرعت بالا ارسال بسته می نماید برای وظایف تشخیص نفوذ طراحی نشده است. مراقبت های مناسب باید انجام شود تا دستگاه از حد مجاز بار تجاوز نکند و دچار قطع ارسال نگردد. بنابراین نیازمندی برای سنجش آنی تشخیص با ۳۰ ثانیه است که CPU و باید ۱۰ درصد یا کمتر مورد استفاده شوند.

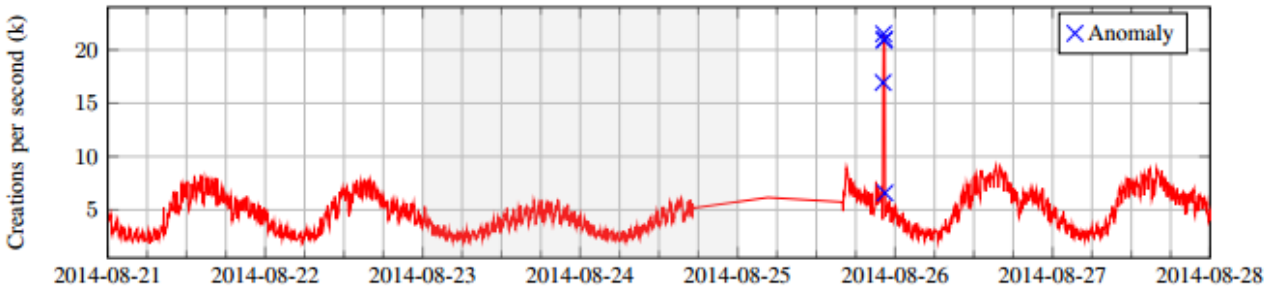
ب) راه اندازی و توسعه

اجرای توصیف شده در بخش پنجم بر روی یک دستگاه سیسکو مدل Catalyst 6500 با موتور سوپروایز ۷۲۰ و IOS ورژن 15.1(2)SY1 توسعه داده شده است. به این دستگاه کارت WS-X6708-10G-3C با اتصال اترنت اضافه می گردد. ترافیک مورد استفاده برای اعتبارسنجی از اتصال بالایی از پردیس شبکه دانشگاه UT به مرکز ملی تحقیقات و آموزش و پرورش شبکه SURFnet هلند که شامل هر دو ترافیک آموزشی یعنی ترافیک تولید شده توسط دانشکده ها و دانشجویان و ترافیک خوابگاه دانشگاه می باشد. لینک دارای سرعت ۱۰ گیگابیت بر ثانیه با توان عملیاتی متوسط ۱,۸ گیگابیت در ثانیه در طول ساعات کاری است. علاوه بر این، داده های جریان به گردآوری جریان صادر شده می پردازد به طوری که حملات تشخیص داده شده توسط نمونه اولیه می تواند به صورت دستی تایید شود.

ترافیک شبکه مورد استفاده در [۹] متفاوت از ترافیک مورد استفاده در این کار است، بنابراین روشن است که باید به تنظیم پارامترهای الگوریتم تشخیص برای رسیدن به دقت مشابه در [۹] پرداخته شود. به این ترتیب مقدارهای پارامتر مطلوب برای نقطه مشاهدات انتخاب می گردد. پارامتر c_f، برای تخمین تعداد جریان ورودی نهان ایجاد شده استفاده می شود، که در بخش چهارم توصیف گردیده است. اندازه $c_f = 59.8133$ بسته در هر جریان در متوسط راه اندازی است.



شکل ۳ – بار دستگاه سیسکو مدل Catalyst 6500 روی زمان



شکل ۴ – جریان ورودی نهان ایجاد شده در ثانیه

ج) نتایج

مهمترین نیازمندی در این کار اعتبار است که باید با اجرای سبک انجام شود به طوری که به فعالیت های اصلی دستگاه ارسال بسته یعنی مسیریاب و سوئیچ اختلال وارد نشود. اندازه گیری مصرف منابع هم از CPU و حافظه صورت می گیرد. در شکل ۳ بار CPU همراه دستگاه و استفاده از حافظه نشان داده شده است. که به طور متوسط بیش از ۱۵۰ ثانیه است. با استفاده از SNMP بار CPU را در سه جزء یعنی مسیریابی CPU که از دسته ترافیک L3 است و دو CPU سوئیچینگ که ترافیک روند در L2 را اندازه گیری می کند.

در شکل ۳ سیاست در طول کل دوره اندازه گیری فعال است. از آنجا که استفاده از CPU در اکثر فرآیندها مقدار صفر تا یک درصد گزارش شده و تنها قله یک درصد است. برای اندازه گیری این، سوئیچ را مجدد بوت و تمامی حافظه و CPU استفاده شده پاک می گردد. در طی اندازه گیری یک بار بر روی CPU مسیریابی با ۴ درصد و حافظه ۳۱,۳ درصد استفاده می گردد که پس از فعال شدن سیاست های مورد نظر یک افزایش ۲۰ درصدی در استفاده از CPU و افزایش ۰,۲ درصدی در حافظه مشاهده می گردد.

در طول دوره که در آن الگوریتم تشخیص مستقر است یک حمله به شبکه در ۲۵ اوت اعتبارسنجی شده است. مدت این حمله ۲۰ دقیقه بوده و شامل ترافیک DNS و TCP می باشد. با توجه به اندازه گیری های فوق نتیجه گرفته می شود که استفاده از

حافظه برآورد مورد نیاز ۱۰ درصد یا کمتر است. با این حال ۲۰ درصد بار ناشی از CPU از اجرا با توجه به ۱۰ درصد راضی کننده نیست.

دومین نیازمندی تاخیر تشخیص می باشد. این نیازمندی باید دقیق باشد و در نمونه اولیه اعتبار سنجی شده باشد [۹]. نتایج در مورد تاخیرهای تشخیص چند گانه ۱۰ ثانیه است که حداقل می باشد. حمله مشاهده شده در شکل ۴ در سومین فاصله تشخیص داده شده است و در ۳۰ ثانیه عمل تشخیص انجام گرفته است. نیازمندی نهایی در تشخیص حمله DDoS دقت می باشد. در شکل ۴ تعداد ورودی جریان نهان ایجاد شده در فاصله اندازه گیری را نشان می دهد که متوسط فاصله بیش از ۵ دقیقه است.

۷- بحث

نمونه اولیه ارائه شده در بخش پنجم اطلاعات پلتفرم های زمینه را با استفاده از SNMP بازیابی می کند. بازیابی اطلاعات با استفاده از SNMP می تواند بوسیله هر دستگاه دیگری، حتی یک Raspberry Pi، با به حداکثر رساندن قدرت پردازش در دسترس از دستگاه ارسال برای مسیریابی و سوئیچینگ انجام شود. با این حال، از آنجا که هدف نهایی انجام کاهش حملات است که نیاز به اطلاعات در مورد حمله کنندگان و اهدافشان دارد، به طور معمول تشخیص انجام شده بر روی خود دستگاه ارسال (که در آن NetFlow در دسترس است)، برای استقرار سریع در محیط های تولید بدون هیچ هزینه اضافی اجازه می دهد.

تشخیص حملات اولین گام بسیار مهم است، که صرفا در خدمت هدف نهایی یعنی کاهش حملات است. در [۹]، نه تنها تشخیص حمله مورد بحث است، بلکه کاهش آن نیز مطرح است. هنگامی که الگوریتم تشخیص اجرا می شود و یک نمونه اندازه گیری غیر عادی در نظر گرفته می شود، کاهش با شمارش تعداد رکوردهای جریان صادر در منبع آدرس IP آغاز شده؛ به محض این که بیش از ۲۰۰ پرونده جریان با سه بسته یا کمتر در هر ثانیه از آدرس IP مبدا خاص صادر شده باشد، آدرس IP منبع در لیست سیاه قرار می گیرد. آدرس های IP در لیست سیاه به یک فایروال برای جلوگیری از ترافیک اضافه مهاجم اضافه می شوند. علاوه بر این، برای جلوگیری از گرآوری جریان از سربار، سوابق جریان با این آدرس IP به گردآوری ارسال نمی شود. وقتی که الگوریتم، پایان حمله را تشخیص می دهد، قوانین ایجاد شده از فایروال حذف می شود.

اطلاعات مورد استفاده برای شناسایی مهاجمان در IOS در دسترس نیست [۹]؛ تنها تعداد کل ورودی های نهان شده صادرات دردسترس است. یک روش جایگزین برای شناسایی مهاجمان، تجزیه و تحلیل محتویات جریان نهان شده می باشد. با این حال، آدرس های IP حمله کنندگان در حافظه نهان شده در طول حمله DDoS ضعیف خواهند شد در نتیجه تعداد زیادی از جریان نهان شده ورودی از مهاجمان که تولید مقدار زیادی ترافیک می کنند. با این حال، زمان مورد نیاز برای بازیابی و پردازش تمام جریان نهان شده زیر بار که متشکل از حداقل ۱۲۸ کیلو ورودی، بسته به سخت افزار استفاده می شود می تواند ده ثانیه طول بکشد، کاهش آن به سختی امکان پذیر است.

یک رویکرد متفاوت برای اجرای کاهش بکارگیری ویژگی های IOS است که ردیابی برترین $x \in$ (۰ و ۲۰۰) جریان های شامل بیشترین مقدار را نگه می دارد، چه از نظر بسته ها و یا بایت ها، اشاره به مطلب Top Talkers NetFlow دارد. این ویژگی نمی تواند بوسیله تعدادی از جریان های تولید شده توسط میزبان نشان دهد، که خیلی زیاد برای منابع حملات DDoS بشوند. علاوه بر این، این احتمال وجود دارد که کاربران مشروع در لیست باشند، آنها می توانند فقط بسیاری از بسته ها و بایت ها را تولید کنند. پس نتیجه می شود که شناسایی مهاجمان در کنار کاهش در این کار دشوار است.

۸- نتیجه گیری

هدف از این پژوهش، بررسی استفاده از دستگاه های ارسال بسته در سطح High-End برای تشخیص حملات DDoS و در نهایت با کاهش زمان بصورت آنی بود که توسط به کارگیری نمونه دستگاه سیسکو مدل Catalyst 6500 بصورت آنی انجام شد. نتایج نشان می دهد که تشخیص سیل حملات در ده ثانیه امکان پذیر است، تشخیص آنی به طور گسترده در دسترس به امکانات پلتفرم سوئیچ بستگی دارد. با این حال، نمونه نیز نشان داده است تداخل مسیریابی و سوئیچینگ می تواند باعث ایجاد پردازش بار CPU به میزان ۲۰ درصد شود. با توجه به اپراتورهای شبکه های مختلف با موجود بودن ظرفیت دستگاه ارسال بسته می توانند تشخیص حمله DDoS را در محیط تولید انجام دهند. در حالی که امکان استقرار اجرای آن فقط با ۲۰ تا ۳۰ درصد ظرفیت CPU در دسترس است، برای مثال نیاز به یک اولویت پایین تر با عدم دخالت در فرآیندهای مسیریابی و سوئیچینگ اجرا می شود. از آنجا که این موضوع امکان دارد باعث بی ثباتی در نمونه شود، حداقل ظرفیت در دسترس CPU، ۴۰ درصد توصیه می شود.

چند مورد نیازمندی قبل از شناسایی وجود دارد: اول، یک رد پای کوچک برای اجرای الگوریتم تشخیص لازم است.نتایج اعتبارسنجی نشان داده اند که افزایش قابل مشاهده ای در CPU و استفاده از حافظه در طول حملات وجود ندارد. با این حال، زمانی که نظارت اجرا می شود افزایش بار در CPU، ۲۰ درصد و استفاده از حافظه ۰,۲ درصد قابل مشاهده در نمونه است. در حالی که میزان استفاده از حافظه رضایت بخش است به علت استفاده از ۱۰ درصد و کمتر از منابع در دسترس، نیازمندی بار استفاده شده در CPU رضایت بخش نمی باشد. دوم، اعتبارسنجی از نمونه در شبکه دانشگاه UT نشان داده است که میزان تاخیر امکان تشخیص برای حملات با شدت زیاد ۳۰ ثانیه است، نیاز فوق برای تشخیص آنی رضایت بخش است. این مربوط به سه برابر ۱۰ ثانیه اندازه گیری انجام شده است. اندازه گیری فواصل کوچکتر ممکن است میزان تاخیر تشخیص را کاهش دهد اما آن بوسیله قرآیند مدیریتی به احتمال زیاد اجرای تشخیص را با زمان اضافی خواهد ساخت. آخرین مورد نیاز برای اجرا دقت تشخیص است. نتایج اعتبار سنجی نشان می دهد که تعداد موارد مثبت کاذب کم است، در حالی که نرخ تشخیص بالا است، پس نتیجه حاصل می شود که دقت بالا است.

کاهش قدم بعدی پس از تشخیص است. تحقیقات نشان داده است که در حالی که ممکن است برای به دست آوردن اطلاعات کافی برای شناسایی مهاجمان، فرمانی استفاده شود که برای بدست آوردن این اطلاعات ۱۰ ثانیه را زمانی که سوئیچ تحت بار سنگین در شرایط حملات سیل آسا قرار دارد اتفاق افتد. پس نتیجه گرفته می شود که کاهش زمان تا آنی بر روی سخت افزار مورد استفاده در این کار امکان پذیر نیست.

کار آینده شامل بررسی پیاده سازی های جایگزین بر روی سخت افزارهای مختلف خواهد بود که شامل سخت افزارهای قوی تر با قابلیت های اضافی باشند. این سخت افزارهای قوی تر به طور قطع باعث کاهش در زمان تشخیص خواهند شد.

مراجع

[1] CloudFlare, Inc., “Technical Details Behind a 400Gbps NTP Amplification DDoS Attack,” 2014, accessed on 21 January 2015. [Online]. Available: http://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack

[2] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," RFC 7011 (Internet Standard), Internet Engineering Task Force, September 2013. [Online]. Available: http: //www.ietf.org/rfc/rfc7011.txt

[3] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of IP flow-based intrusion detection," IEEE Communica- tions Surveys & Tutorials, vol. 12, no. 3, pp. 343–356, 2010.

[4] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow Monitoring Explained: From Packet Capture to Data Analysis with Netflow and IPFIX," IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 2037–2064, 2014.

[5] A. A. Galtsev and A. M. Sukhov, "Network attack detection at flow level," in Proceedings of the 11th international conference and 4th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking, 2011, pp. 326–334.

[6] H. A. Nguyen, T. Tam Van Nguyen, D. I. Kim, and D. Choi, "Network Traffic Anomalies Detection and Identification with Flow Monitoring", in 5th IFIP International Conference on Wireless and Optical Communications Networks, WOCN'08, 2008, pp. 1–5.

[7] N. Muraleedharan, A. Parmar, and M. Kumar, "A Flow based Anomaly Detection System using Chi-square Technique," in Proceedings of IEEE 2nd International Advance Computing Conference, IACC'10, 2010, pp. 285–289.

[8] R. Hofstede and A. Pras, "Real-Time and Resilient Intrusion Detection: A Flow-Based Approach," in Proceedings of the 6th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security, AIMS'12, Ph.D. Workshop, ser. Lecture Notes in Computer Science, vol. 7279. Springer Berlin Heidelberg, 2012, pp. 109–112.

[9] R. Hofstede, V. Bartoš, A. Sperotto, and A. Pras, "Towards Real-Time Intrusion Detection for NetFlow/IPFIX," in Proceedings of the 9th International Conference on Network and Service Management, CNSM'13, 2013, pp. 227–234.

[10] J. Follett, "Cisco: Catalyst 6500 The Most Successful Switch Ever," 2006, accessed on 21 January 2015. [Online]. Available: http://www.crn.com/news/networking/189500982/ cisco-catalyst-6500-the-most-successful-switch-ever.htm

[11] R. Hofstede, I. Drago, A. Sperotto, R. Sadre, and A. Pras, "Measurement Artifacts in NetFlow Data," in Proceedings of the 14th International Conference on Passive and Active Measurement, PAM'13, ser. Lecture Notes in Computer Science, vol. 7799. Springer Berlin Heidelberg, 2013, pp. 1–10.

[12] Cisco Systems, Inc., "Catalyst 6500/6000 Switch High CPU Utilization," 2012, accessed on 21 January 2015. [On-line]. Available: http://www.cisco.com/c/en/us/support/docs/switches/ catalyst-6500-series-switches/63992-6k-high-cpu.html