Identifying Application-Layer DDoS Attacks Based on Request Rhythm Matrices

IEEE ACCESS 2019

11 citations

حملات منع خدمت توزیع شده لایه اپلیکیشن به دلیل گریز از سامانه‌های تشخیص نفوذ و همچنین افزایش کاربرد، تهدیدی جدی در برابر سایت‌ها به حساب می‌آیند. یک روش آماری ارایه می‌دهیم که اشیا تقاضاشده و مقادیر متناظرشان را در یک ساختار به نامRM به صورت فشرده ذخیره می‌کند که در واقع نشانگر رفتار کاربر است. حملات AL-DDOS را با افزایش آنومالی در این ساختار تشخیص[1] می‌دهیم و همچنین به کمک میزان دوری[2] از نرخ تغییر، هاستهای متخاصم را شناسایی می‌کنیم و مقابله[3] می‌کنیم.

برای شناسایی الگوی اپلیکشن‌ها (ترافیک اپلیکیشن ها هم در content و هم در dwell time الگوی ریتمی دارند) از سایز بسته و زمان بین ورودی[4] بسته‌های HTTP-Request متوالی در یک جریان به عنوان مشخصه‌های اصلی استفاده می‌کنیم. و از استخراج این ویژگی‌ها برای ساخت RM استفاده می‌کنیم. به طور مثال برای کاربران وب ، RM نشان‌دهنده access trajectory fragments مثل ترتیب صفحات مشاهده شده و زمان گذرانده شده در هر صفحه می‌باشد. برای مهاجم تقلید این رفتار سخت می‌باشد.

در این مقاله هدفمان حمله‌های سیل اسای HTTP که به وب‌سایت می‌باشد، است

ماتریسی که می‌سازیم در لایه پکت می‌باشد(مثل برخی روش‌های قبلی) اما ویژگی‌هایی استخراج می‌کند که اطلاعات ارزشمندتری از اپلیکیشن می‌دهد. در واقع اطلاعات سایز بسته‌ها و زمان بین ورودی بسته‌ها که اطلاعات لایه شبکه هستند، به ترتیب متناظر بعد فضایی و بعد زمانی که اطلاعات خوبی در زمینه اپلیکیشن می‌باشد، به ما می‌دهند.

تشریح رویکرد ما:

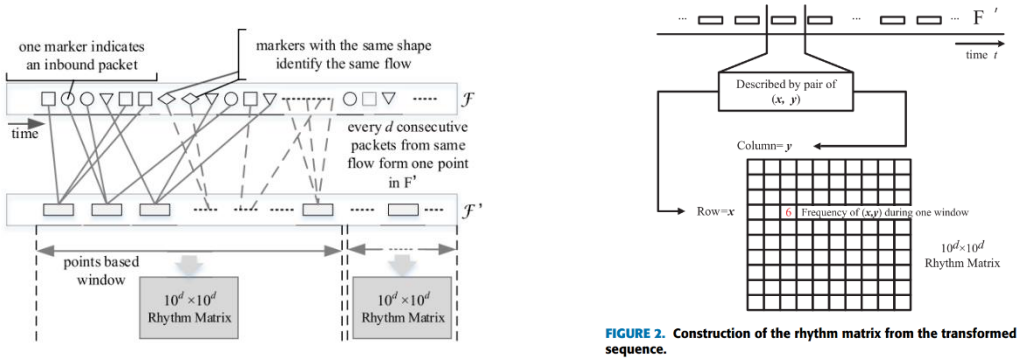

FIGURE 2. Construction of the rhythm matrix from the transformed sequence.

استخراج ویژگی: در هر پنجره زمانی بسته‌ای که بیاد، هش آدرس مبدا آن را حساب کرده و دوتایی(سایز بسته، زمان آمدن بین بسته و بسته قبلی اش) در فلو f قرار می‌دهیم.به کل جریان‌ها نیز F گوییم اما برخی بسته‌ها را در F قرار نمی‌دهیم برای مثال، از بسته‌های TCP با سایز پیلود صفر، صرفنظر می‌کنیم چون اثری در شناسایی رفتار کاربر نخواهند داشت.

[1] Detection
[2] Change outlier
[3] Mitigation
[4] Interarrival

تبدیل داده‌ها: در یک بسته سایز هدرها برای یک کلاینت-سرور ثابت است ولی سایز پیلود بسته برای هر اپلیکیشن و پروتکل متفاوت می‌باشد. برای بحث فاصله زمانی بسته‌ها(که به عوامل مختلفی بستگی دارد)، از یک تبدیل لگاریتمی استفاده می‌کنیم که بازه مقادیر قابل قبول برای سایز بسته و زمان ورود بین دو بسته را کوچک کنیم. با یک نرخ نمونه d، جریانات را تبدیل به f′ میکنیم.

ساخت RM: پنجره زمانی را بر روی f′ تنظیم می‌کنیم. اندازه جدول RMنیز به مقدار drop point ها بستگی دارد. و در حالت عادی، توزیع droppoint ها به صورت نرمال و درحالت پایداری میباشد. و هر مقدار خانه آن، بیانگر تعداد دفعات تولید (x,y) از F′ می‌باشد. در واقع، برابر فرکانس آن خانه در F′ می‌باشد. سایز پنچره‌ها (که آن را برای جلوگیری از اتلاف حافظه بر روی F′ تعریف می‌کنیم) نیز به تعداد drop point بستگی دارد که در اینجا آن را با PktW نشان می‌دهیم.

به طور ساده تر هر تاپلی از بسته‌ها که می‌آید، بر اساس d، آنها را تقسیم بندی کرده و با هم در نظر می‌گیرد و مختصات x,y که برای drop point متناظر می‌باشد را محاسبه می‌کند و به مقدار آن یک واحد اضافه می‌کند و این کار آپدیت droppoint را تا زمانی که تعداد droppoint ها به مقدار PktW برسد (سایز پنجره) ، انجام می‌دهد.

شرح ترافیک از روی اطلاعات ذخیره شده در RM: مزیت اصلی روش‌ما، شناسایی و درک سیر دسترسی کاربر و تغییر در آن می‌باشد. در اینجا نشان می‌دهیم که چگونه RM می‌تواند مشخصه‌های ترافیک از انواع مختلف را توصیف کند . دیتاست ما شامل ترافیک عادی و انواع مختلف حملات به صورت زیر می‌باشد:

- Single URL flood
- Multi-URL flood
- Random-URL flood

۴ تا جدول RM با مقدارآستانه یکسان PktW برای trace می‌سازیم و فاصله اقلیدسی بین هر کدام از آنها را حساب می‌کنیم. بر اساس فاصله اقلیدسی میانگین بین دوگروه، ماتریس فاصله را می‌سازیم. از جدول فاصله ماتریسی نتایج زیر را استنتاج می‌کنیم:

- RMها اگر از یک مبدا یکسان باشند، شبیه به هم خواهند بود
- RMهای حملات منع خدمت با ترافیک معمولی تفاوت خواهند داشت

مقابله با حملات منع خدمت توزیع‌شده: در زمان حمله، چندین کاربر متخاصم با یک رفتار یکسان به یک سایت قربانی متصل می‌شوند که باعث تغییر به خصوصی در یک سری خصایص ماتریس RM می‌شود. به کمک این عناصر، می‌توان هاستهای متخاصم را شناست.

1. نرخ تغییرات و شناسایی حملات منع خدمت[5]: بین چندین RM، المنتهای با بیشترین تغییر غیرنرمال را می‌یابد. برای مقایسه RM فعلی را با RM دو تا نمونه‌ی قبلی مقایسه می‌کند
2. داده‌های پرت و شناسایی هاستهای متخاصم[6]: یک مقدارآستانه‌ای تعریف می‌کنیم که اگر تعداد droppoint های یک هاست که به عنوان outlier شناسایی شدند، بیشتر از آن مقدار شد، ما آن را به عنوان یک هاست متخاصم شناسایی می‌کنیم. مقدار این آستانه را در ابتدا بر روی مقادیر بالا تنظیم می‌کنیم(مثلاً ۸۰ درصد)، و زمانی که یک حمله شناسایی شد مقدار آن را تا زمانی که سایت بتواند به صورت طبیعی رفتار کند، کاهش می‌دهیم.

[5] Change rates and DdoS detection
[6] Outliers and malicious host identification

آنالیز پیچیدگی: پیچیدگی محاسباتی تولید RM برابر O(n) می‌باشد که n تعداد drop point های در RM است که بیشتر از PktW نمی‌باشد. جدول هش برای ذخیره اطلاعات پکت‌ها نیز از مکان‌یابی استفاده شده است و لذا پیچیدگی فضایی آن O(1) می‌باشد.

آزمایش و ارزشیابی: از ابزارهای LOIC و HOIC برای شبیه‌سازی حمله AL-DDOS و اندازه گیری معیارهای میزان دقت و زمان استفاده می‌کنیم. با افزایش Pkt، نرخ TPR افزایش و FPR کاهش خواهد یافت. موارد زیر را بررسی می‌کنیم:

- کارایی در شناسایی AL-DdoS
- کارایی در مقابل flas crowd ها: همانطور که میدانیم تشخیص دقیق و تمایز ایجاد کردن flash coward ها از افزایش ناگهانی ترافیک عادی از ویژگی‌های اصلی تشخیص دهنده AL می‌باشد.روش ارایه شده نیز از این ویژگی برخوردار می‌باشد. برای این مود نیز از دیتاست‌های شامل flash coward استفاده می‌کنیم.