شناسایی حملات انکار سرویس توزیع شده با استفاده از تولید امضای بسته داده

یونس توحیدیان[۱]، جواد جهانشیری[۲]، هادی شرکاء[۳]، امیرحسین زاهد[۴]

۱-کارشناس ارشد، گروه فناوری اطلاعات گرایش تجارت الکترونیک، دانشگاه اترک، قوچان، ایران
unes.tohid@gmail.com

۲-استادیار گروه فتا، دانشگاه علوم انتظامی امین، تهران، ایران
jahanshiri@nict.ir

۳-کارشناس ارشد مهندسی فناوری اطلاعات، دانشکده مهندسی، دانشگاه بین المللی امام رضا(ع)، مشهد، ایران
h.shoraka.b@gmail.com

۴-کارشناس مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بین المللی امام رضا(ع)، مشهد، ایران
zahed.amir2@gmail.com

چکیده: با توسعه فناوری‌های نوین اطلاعاتی و ارتباطی و فراگیر شدن بهره‌برداری از فضای سایبر، حملات انکار سرویس توزیع‌شده (**DDoS**) یک تهدید جدی برای سازمان‌های آنلاین می باشد. این حملات می توانند اثرات مخرب داشته باشد به عنوان مثال بروی یکی از اثرات آن می تواند از جنبه چهره عمومی نماد تجاری و درآمدهای آن باشد. یک روش مناسب برای مقابله برای دفاع در برابر حملات **DDoS**، ذخیره یک امضا برای هر حمله است. امروزه بیشتر سازمان‌ها و نهادها اعم از کشوری و لشکری (نظامی و انتظامی) با چنین حملاتی روبرو هستند لذا در این تحقیق تلاش بر آن است تا ضمن شناخت سیستم‌های شناسایی نفوذ در روش پیشنهادی به منظور تولید امضا لازم تبیین شود. با استفاده از این روش به محض پیداشدن این امضا در ترافیک، هر حمله را می‌توان شناسایی کرد. هرچند، این فرآیند چندان ساده نیست و فرآیند تولید امضا معمولا وقت‌گیر و نیازمند تلاش فراوان است. برای کمک به منظور حل این چالش حل این مشکل تولید امضا، در این تحقیق، یک روش خودکار را برای تولید امضاهای برپایه بسته داده برای حملات **DDoS** پیشنهاد می‌شود. این تحقیق همچنین روابط بین بسته‌های داده مختلف یک حمله یکسان را بررسی می‌کند. درواقع امضای دیجیتال مبتنی بر بسته داده برای حملاتی استفاده می شود که الگو و هسته اولیه آنها یکی می باشد و تنها تفاوت جزئی در ساختار حمله تغییر می کند. پژوهش با پیشنهاد یک الگوریتم تولید امضا و اعتبارسنجی آن با استفاده از ابزارهای کاربردی، به پایان می‌رسد.

واژه های کلیدی: امضای بسته داده، تولید، حملات انکار سرویس توزیع شده، شناسایی.

تاریخ ارسال مقاله : ۱۳۹۹/۱۲/۰۱
تاریخ پذیرش مقاله : ۱۴۰۰/۰۱/۱۵
نویسنده مسئول : یونس توحیدیان

۱- مقدمه

انکار سرویس توزیع‌شده[1] (DDoS) یک حمله هماهنگ‌شده است که قصد تضعیف‌کردن سرویس‌ها و خارج از دسترس‌کردن آنها برای کاربران قانونی را دارد. حملات DDoS با ارسال درخواست به صورت سیل آسا سعی در اشغال پهنای باند سرویس دهنده، سرریزکردن پهنای باند سرویس‌ها و یا خالی‌کردن منابع آنها، به اهدافشان می‌رسند. اثر حملات DDoS بروی درآمد و محبوبیت عمومی کسب‌وکارهای آنلاین می‌تواند بشدت مخرب باشد. حملات DDoS می‌تواند منجربه ازدست‌رفتن بیش از ۳۰۰ هزار دلار در ساعت شده [۱] و باعث اعتمادزدایی مشتری شود.

یکی از مهمترین روش‌ها برای مبارزه با حملات DDoS، استفاده از سیستم شناسایی نفوذ[2] (IDS) است که حملات DDoS را هرچه سریع‌تر شناسایی کرده و درنتیجه اثرات مخرب این حملات را بروی کسب‌وکار کم می‌کند.

IDS ها براساس مکانیزم شناسایی حمله‌شان، به دو دسته اصلی طبقه‌بندی می‌شوند. دسته اول، سیستم شناسایی نفوذ برپایه امضا[3] (S-IDS) است که حملات را با مقایسه ترافیک فعلی شبکه و امضاهای حمله شناخته‌شده، شناسایی می‌کند؛ S-IDS یک روش مؤثر را برای شناسایی حملات شناخته‌شده ارائه می‌دهد، هرچند، این روش در مواجهه با حملات (ناشناخته) جدید، بی‌فایده است.

دسته دوم، سیستم شناسایی نفوذ برپایه رفتار غیرعادی[4] (A-IDS) است که با داشتن یک الگو از رفتار عادی (فاقدحمله) شبکه و نیز یک الگو از رفتار جاری شبکه، کار می‌کند. A-IDS با مقایسه مستمر الگو جاری با الگو عادی شبکه، عمل می‌کند و اگر مقدار تفاوت این دو از یک مقدار آستانه بیشتر شود، هشدار حمله داده می‌شود.

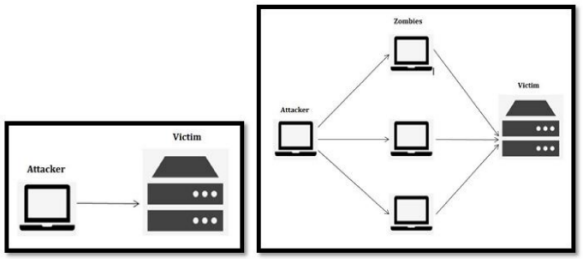
یک ترکیب از هر دو IDS را می‌توان برای داشتن یک خط دفاعی قوی دربرابر حملات DDoS استفاده کرد، در این ترکیب، A-IDS برای شناسایی یک حمله جدید، و S-IDS برای شناسایی حملات قدیمی استفاده می‌شود.

امروزه بیشتر سازمان‌ها و نهادها اعم از کشوری و لشکری (نظامی و انتظامی) از چنین حملاتی مستثنی نیستند، این تحقیق، یک روش کارآمد تولید خودکار امضا برپایه بسته داده را پیشنهاد می‌دهد که حملات جدید را بدون نیاز به کار دستی، مشخص می‌کند و مورد بهره‌برداری سازمان‌ها و نهادهای موصوف خواهد بود.

یک حمله انکار سرویس (DoS)، به حمله ای گفته می شود که هدف آن جلوگیری دسترسی کاربران عادی یک وبسایت هدف از ارائه خدمات به کاربرانش می باشد. این حمله بوسیله بارگذاری اضافی وب سرور های سیستم هدف و مصرف پهنای باند و/ یا منابع محاسبه‌گر انجام می‌شود. این مصرف بیش از حد منابع، یا باعث می‌شود که وب‌سایت از دسترس خارج شود، و یا خدمات آن را بشدت تضعیف می‌کند.

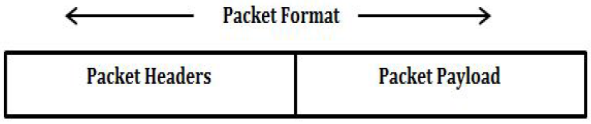
تفاوت کلیدی بین یک حمله DoS و یک حمله DDoS در تعداد مهاجمان و ماشین‌های درگیر در حمله است. در یک حمله DoS تنها یک ماشین برای اجرای فعالیت خرابکارانه استفاده می‌شود، درحالی‌که در یک حمله DDoS از تعداد زیانی ماشین، برای انجام حمله استفاده می‌شود. آن ماشین‌ها معمولا ماشین‌های آلوده‌شده و تغییریافته کاربران ناآگاه (که زامبی‌ها نام دارند) هستند، که بوسیله مهاجم اصلی کنترل می‌شوند تا حمله را انجام دهند.

یک نمایش از حملات DoS و DDoS در شکل ۱ نشان داده شده است.



شکل (۱) : حملات DoS در برابر DDoS

در ساده‌ترین حالت، حملات DoS ارسال پشت سرهم ترافیک می‌باشد. ترافیک از پاکت‌هایی درست شده است که هرکدام ساختار آن در شکل ۲ آمده است در آن یک پاکت را می‌توان به بخش سربرگ‌ها و بخش بسته داده تقسیم کرد. سربرگ‌ها برای لایه‌های مختلف درون مجموعه پروتکل ارتباط یعنی پروتکل کنترل انتقال/ پروتکل اینترنت[5] (TCP/IP) استفاده می شوند [۲]، درحالی‌که بخش بسته داده باتوجه به پروتکل برنامه کاربردی مورد استفاده، متغیر است.



شکل (۲): ساختار پاکت

رایج‌ترین حملات DDoS در جدول (۱) بررسی شدند. در سطح سازمان‌ها و نهادها اعم از کشوری و لشکری (نظامی و انتظامی) چنین حملاتی قادرند در ارایه خدمات برخط جلوگیری و عدم ارایه خدمات امنیتیف انتظامی و ... موجب ایجاد بروز نارضایتی و زیر سوال بردن قابلیت‌های شازمان‌ها و نهادهای موصوف گردد، به عنوان مثال در نیروی انتظامی جمهوری اسلامی ایران ارایه خدمات در سامانه‌های خبرگزاری پلیس، شماره‌گذاری، گذرنامه، اجراییات و ... از جمله ظرفیت های مورد استفاده در حملات موصوف است؛ لذا این تحقیق با

[1] Distributed Denial of Service
[2] Intrusion Detection System
[3] Signature based Intrusion Detection System
[4] Anomaly based Intrused Detection System
[5] Transmission Control Protocol/ Internet Protocol

هدف بررسی حملات موصوف و به میزان اثرگذاری آن پرداخته و در نهایت یک روش خودکار را برای تولید امضاهای برپایه بسته داده برای حملات DDoSارایه خواهد داد؛ لذا در این تحقیق این سوال مطرح می‌گردد که : «کدام حملات دارای بسته‌های داده تکراری یکسان هستند، و می‌توانند گزینه مناسبی برای تولید امضا باشند؟».

جدول (۱): بررسی حملات DDoS

DNS	بله	همه پاسخ‌ها، درخواست DNS را حمل می‌کنند
NTP	نه	پاسخ monlist وابسته به سرور است
SNMP	نه	پاسخ وابسته به شبکه پرس‌وجوشده است
SSDP	نه	پاسخ وابسته به شبکه پرس‌وجوشده است
NetBIOS	نه	پاسخ وابسته به شبکه پرس‌وجوشده است
Portmap	نه	پاسخ وابسته به سرور است
Chargen	بله	پاسخ‌ها، رشته‌های تکرارشونده از کاراکترهای با طول‌های مختلف هستند
QoTD	نه	QotD یک پروتکل قدیمی است و امروزه به‌ندرت در حملات استفاده می‌شود
TCP SYN	نه	این یک حمله برپایه سربرگ است

نتایج نشان می‌دهند که دربین حملات بررسی‌شده، DNS و Chargen مناسب‌ترین گزینه ها برای تولید امضای برپایه بسته داده هستند. به همین دلیل، بخش تجربی بیشتر بروی حملات DNS و Chargen تمرکز خواهد کرد. هرچند، ما انتظار داریم که الگوریتم‌های طراحی‌شده بروی دیگر پروتکل‌ها، کار کنند. اگر منبع یکسان یا بدنه درخواست POST یکسان در نسبت بزرگی از ترافیک حمله استفاده شود، انتظار می‌رود که برای حملات HTTP GET و HTTP POST نیز جهت بازرسی بسته داده مناسب باشند، هرچند، که این موارد بررسی نشده است. این نکته همچنین در گزارش وضعیت [۳] بیان شده است که حملات DNS و Chargen بیش از ۳۰ درصد حملات DDoS در فصل اول سال ۲۰۱۷ را تشکیل دادند.

شناسایی نفوذ، فرآیند دنبال‌کردن رویدادهای رخ‌دهنده در یک سیستم یا شبکه رایانه‌ای و تحلیل آنها برای نشانه‌های رخداد های احتمالی است، که این تهدیدات حتمی نقض قوانین امنیت رایانه، قوانین استفاده قابل قبول و یا اعمال اجرایی امنیتی استاندارد هستند.

۲- طبقه‌بندی IDS

سیستم‌های شناسایی نفوذ را می‌توان به‌صورت زیر طبقه‌بندی کرد:

۲-۱- IDS برپایه میزبان

IDS برپایه میزبان، در سطح میزبان استفاده می‌شود. این سیستم رویدادهایی مانند هویت‌های فرآیند، درخواست‌های سیستم، و اطلاعات ترافیک شخصی را برای یک میزبان مشخص پایش و تحلیل می‌کند.

۲-۲- IDS برپایه شبکه

IDS برپایه شبکه در سطح شبکه استفاده می‌شود. این سیستم اطلاعات ترافیک شبکه مثل حجم ترافیک، آدرس‌های IP و درگاه‌های سرویس برای یک شبکه خاص را پایش و تحلیل می‌کند.

۲-۳- IDS برپایه امضا

این سیستم همچنین با نام سیستم شناسایی سوءاستفاده شناخته می‌شود. این سیستم برپایه جستجوی الگوهای سوءاستفاده شناسایی‌شده مشخص در ترافیک ورودی قرار دارد؛ الگوهای حمله شناسایی‌شده می‌توانند هر ترکیبی از شرایط، پرچم‌ها و ویژگی‌های پاکت باشند، و امضاهای حمله نام دارند، و در بانک اطلاعاتی S-IDS ذخیره می‌شوند.

۲-۴- IDS برپایه رفتار غیرعادی

این سیستم همچنین با نام IDS برپایه رفتار شناخته می‌شود. این سیستم به توانایی شناسایی رفتار مشکوک ترافیک سیستم متکی است.

۳-کارهای مرتبط

تحقیقات در زمینه امنیت اینترنت و حملات DDoS، از زمان ظهور اینترنت به‌دلیل اثر جدی جرایم مجازی بروی اشخاص و کسب‌وکارها، فعال بوده است. کار انجام‌شده در این تحقیق را می‌توان به دو زیرشاخه فضای تحقیقاتی DDoS مرتبط ساخت، که یکی تحقیقاتی است که بسته داده پاکت را برای اهداف مختلفی مثل شناسایی چیز غیرعادی یا طبقه‌بندی ترافیک بازرسی می‌کند، که در این کار با نام زیرشاخه بازرسی بسته داده شناخته می‌شود. زیرشاخه دیگر، آن زیرشاخه‌ای است که با تولید امضای یک حمله (D)DoS سروکار دارد، که با نام زیرشاخه تولید امضا شناخته می‌شود.

۳-۱- زیرشاخه بازرسی بسته داده

در مرجع [۴] یک سیستم شناسایی برپایه رفتار غیرعادی، برمبنای بسته‌های داده پاکت پیشنهاد می‌دهند.

در مرجع [۵] نسبت به PAYL جلوتر رفته و استفاده از مرتبه بالاتر n-گرام‌ها (n>1) را برای شناسایی رویدادهای مخرب و تولید امضاهای مخرب پیشنهاد می‌کنند.

یک روش برای شناسایی پاکت‌های مخرب برپایه بازرسی بسته داده پیشنهاد می‌دهند، که در آن، آنها به مدل‌سازی پاکت‌های برپایه به‌ازای هر سرویس، اتکا می‌کنند، و این کار را با بهره‌برداری از فرکانس‌های قابل پیش‌بینی بایت انجام می‌دهند، که این فرکانس‌ها بین پاکت‌های سرویس یکسان، به‌اشتراک گذاشته شده است [۶].

۳-۲- زیرشاخه تولید امضا

زیرشاخه دیگر تحقیق، در رابطه با تولید امضای حمله مخرب است، خواه این حملات، کرم‌ها باشند، حمله DoS باشد و خواه نوع دیگر از حملات باشد.

در مرجع [۷] یک روش نیمه-خودکار پیشنهاد کردند، تا حملات را دنبال کرده و امضاهای حمله را تولید کند. این روش از سه جزء تشکیل شده است: دیتالاگین یا داده‌برداری، تحلیل داده، و استخراج امضا، که در آن داده‌های حمله با استفاده از روندکاری هانی‌پات[1] جمع‌آوری می‌شود، و این داده‌ها براساس ماشین بردار حمایت[2] (SVM) در دو دسته مشکوک و قانونی طبقه‌بندی می‌شوند [۸].

یک روش برای شناسایی حملات مصرف منابع (D)DoS جدید، و تولید یک امضا برای آنها را ارائه می‌دهند [۹].

استفاده از S-IDS در ترکیب با یک A-IDS شخصی‌سازی‌شده را پیشنهاد می‌دهند، و یک روش تولید امضای وزن‌دار را برای شناسایی حملات DoS، علاوه بر دیگر نوع حملات، معرفی می‌کند [۱۰].

در مرجع [۱۲] یک سیستم خوشه‌بندی بدافزار رفتاری را ارائه می‌دهند، که ترافیک HTTP مخرب را هدف گرفته است. روش پیشنهادشده، هدف تولید امضای خودکار کارآمدتر و بهتر برای بدافزار برپایه HTTP را دارد.

یک مکانیزم تولید امضا برای ترافیک شبکه مخرب پیشنهاد می‌دهند، که از روی یک سیستم هانی‌پات گرفته شده است (یک سیستم پایش‌شده که بروی اینترنت استفاده می‌شود، و هدف آن به‌تله‌انداختن هکرها و ترافیک مخرب برای مطالعات بیشتر است) [۱۴].

در مرجع [۱۵] به روی تولید امضای خودکار برای کرم‌های چندریختی روز صفر[3] متمرکز می‌شوند، که درهر تلاش برای آلوده‌سازی، بسته داده‌شان را تغییر می‌دهند.

یک شبیه‌ساز که امضاها را برای بهره‌برداری از حملات روز صفر، بدون تمرکز زیاد بروی بسته‌های داده پاکت، تولید می‌کند در مرجع [۱۶] بررسی شده است.

جدول ۲. خلاصه تولید امضا

Signature Generation Method	Traffic Classification Method	Designed For	Automation Level	مرجع
Longest common Substring (LCS)	SVM semi-supervised classifier	حملات به سرویس های وب	نیمه خودکار	[۷]
One pass incremental apriori algorithm	Server live ping	حملات مصرف منابع DDoS و Dos	خودکار	[۹]
Weighted frequent items generated	Customized A-IDS	حملات شبکه	خودکار	[۱۰]
NA	Incremental feature based clustering	خوشه بندی ترافیک مخرب HTTP	خودکار	[۱۲]
LCS	Honeypots	ترافیک شبکه مخرب	خودکار	[۱۴]
LCS variations	Honeypots	کرم های چند شکل	خودکار	[۱۵]
LCS and critical exploit string detection	Dynamic Taint Analysis [13]	حملات صفر روزه	خودکار (مبتنی بر شبیه ساز)	[۱۶]

با دنبال‌کردن معیار ارائه‌شده در جدول ۲، تحقیق ما را می‌توان به‌این صورت دسته‌بندی کرد: تولید امضا برای حملات (D)DoS، با استفاده از امضای به‌اشتراک گذاشته‌شده مینیمم به‌عنوان یک روش طبقه‌بندی ترافیک، و نیز استفاده از طولانی‌ترین زیررشته مشترک [۱۱] برای تولید امضا. روش ما به این صورت از کارهای قبلی متفاوت است که این روش بسته‌های داده پاکت را به‌عنوان رشته‌ها درنظر می‌گیرد، و آنها را برپایه خصوصیات متشابه رشته‌ای آنها، خوشه‌بندی می‌کند. این روش همچنین راه‌هایی را برای کاهش دادن سرعت و قدرت پردازش پیشنهاد می‌دهد، و این کار را با درنظرگرفتن یک زیرمجموعه انتخاب‌شده از ترافیک حمله به‌صورت دقیق انجام می‌دهد، و همزمان یک نمایش دقیق از همه حمله را حفظ می‌کند.

۴- روش پژوهش

۴-۱- تولید امضا

یک امضا دنباله‌ای از کاراکترها است که به‌طور مؤثری یک حمله را شناسایی می‌کند؛ این دنباله به‌اندازه کافی بزرگ است که نمایش‌های مختلف یک حمله خاص را دربرمی‌گیرد، و به‌اندازه کافی کوچک است که در همه‌جای ترافیک عادی وجود ندارد.

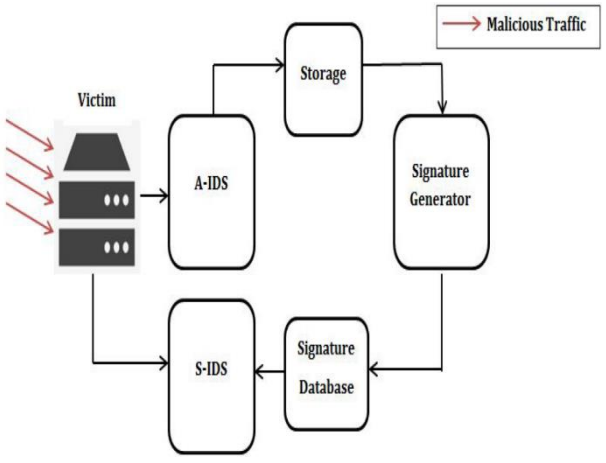
[1] Honeypot framework
[2] Support Vector Machine
[3] Zero-day

مسئله تولید امضا برپایه بسته داده خودکار، با تکنیک‌های پردازش رشته سروکار دارد. این مسئله تصمیم بر یافتن یک الگوی رشته را دارد، که ترافیک حمله مخرب را مشخص می‌کند، و به این سوال پاسخ می دهد که این ترافیک آیا یک کرم است، یک حمله DoS است، و یا نوع دیگری از حمله شبکه است.

داده‌های ورودی را می‌توان به یکی از سه نوع فرمت زیر پردازش کرد: فرمت باینری، فرمت هگز، فرمت اسکی

فرضیه اصلی در این تحقیق این است که حملات DoS به‌عنوان یک رفتار غیرعادی برای شبکه قربانی محسوب می‌شود. همچنین حملات، حامل تعداد زیادی پاکت تکراری مشابه هستند، برای مثال اگر حمله، یک حمله بازتاب DNS باشد، تعداد زیادی از پاکت‌های DNS در زمان حمله وجود خواهد داشت، و اگر حمله، یک حمله بازتاب NTP باشد، پس تعداد قابل توجهی از پاکت‌های NTP در زمان حمله وجود خواهد داشت، و اگر یک حمله HTTP GET باشد، تعداد بیشتری از درخواست‌های HTTP GET نسبت به حالت عادی، در زمان حمله رخ خواهد داد و غیره. این توزیع غیرعادی ترافیک موجب سیگنال‌دادن به A-IDS می‌شود، به این معنی که یک فعالیت مخرب، احتمالا به روی شبکه درحال وقوع است، و این سیستم شروع به ذخیره داده‌های پاکت‌های مشکوک و IPهای منبع آنها خواهد کرد. این فعالیت مخرب ذخیره‌شده، ورودی برای زیرسیستم تولید امضای ما خواهد بود.

در شکل ۳ یک دیدکلی از سیستم اجرایی نشان داده شده است، یک مثال از سناریوی کلی حمله داده شده است، که در آن، نقش هر بخش سیستم توضیح داده شده است، تا یک فهم بهتر از ساختار سیستم حاصل شود.



شکل(۳): ساختار عمومی سیستم

سناریوی کلی به‌صورت زیر اجرا می‌شود:

۱. ترافیک مخرب به سرور فرستاده شده و توسط A-IDS شناسایی می‌شود.

۲. A-IDS، ترافیک مخرب را ذخیره می‌کند.

۳. مولد امضا، ترافیک حمله را خوانده و یک امضا تولید می‌کند.

۴. امضای تولیدشده سپس به بانک اطلاعاتی امضای حمله S-IDS اضافه می‌شود.

۵. اگر هنوز حمله ادامه دار باشد S-IDS می‌تواند ترافیک حمله را فیلتر و خارج کند، و همچنین اگر دوباره حمله روی دهد می تواند آن را متوقف کند. به این شکل، حلقه شناسایی بسته خواهد شد.

۵- روش کلی

در روش پیشنهادی به منظور تولید امضا لازم است مراحل ذیل انجام شود که عبارتند از:

۵-۱- فیلترکردن حمله

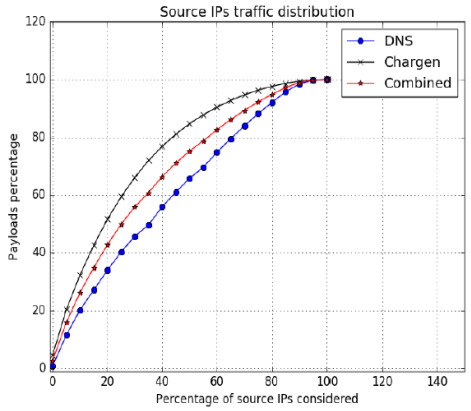
ورودی، یک ردپای حمله است که حامل تعداد زیادی پاکت است. هدف این فاز کاهش دادن تعداد پاکت‌هایی است که باید پردازش شوند. این کار بوسیله استخراج بسته داده های کلیدی از رد پایی انجام می‌شود که به‌طور ایده‌آل نمایشگر همه یا بخش زیادی از پاکت‌های متغیر است که متعلق به حمله می باشند. دو فرض را در نظر می گیریم و فرضیه ها را با آزمایش اعتبارسنجی می‌کنیم. فرضیه ما دو وجه دارد که آن را می‌توان به‌صورت زیر خلاصه کرد:

درصد کمتری از IPهای منبع، درصد زیادتری از ترافیک حمله را می‌فرستند، و ترافیک آنها به بهترین شکل می‌تواند نمایشگر حمله باشد.

بسته‌های داده دارای بیشترین تکرار، نمایشگر درصد زیادی از ترافیک حمله است، و آن بسته‌های داده به بهترین شکل می‌توانند نمایشگر حمله باشند.

۵-۲- روش IPهای منبع

تقریبا ۶۰ درصد یک حمله DNS، توسط ۴۰ درصد IPهای مشارکت‌کننده فرستاده می‌شود. درصد ترافیک در یک حمله Chargen زیاد است، بیش از ۷۵ درصد ترافیک حمله، بوسیله ۴۰ درصد IPهای مشارکت‌کننده فرستاده می‌شود.



شکل (۴): روش IPهای منبع

۵-۳- طبقه‌بندی برپایه شباهت

هدف عمل طبقه‌بندی بسته داده، این است که بتواند آن الگوهای یکتا را استخراج کند، که در آنها بسته‌های داده که نمایشگر هر کدام از الگوهای یکتا هستند، مشابه درنظر گرفته می‌شوند، و نسبت به بسته‌های داده هر الگوی دیگر با بیشترین دقت ممکن، غیرمشابه فرض می‌شوند. بسته‌های داده مشابه قرار است که انواع مختلف یک حمله یکسان باشند، که دارای یک امضای مشترک هستند.

طبقه‌بندی برپایه تشابه، بسته‌های داده را به‌عنوان رشته می‌گیرد، و روش‌هایی را بررسی می‌کند تا آن بسته‌های داده را برپایه شباهت ساختار و محتوایشان به هم مرتبط کند. هدف این طبقه‌بندی، خوشه‌بندی بسته‌های داده مرتبطی است که بخشی از یک نوع حمله می‌باشند، و متفاوت از ترافیک دیگر است.

سنجش تشابه رشته، عملکردی است که بر روی دنباله‌های رشته و شکل‌بندی کاراکتر کار می‌کند. این معیار، تشابه یا عدم تشابه (فاصله) بین دو شیء را سنجیده، و تخمین این مسئله را آسان می‌کند که آیا این دو شیء به هم مرتبط هستند یا خیر. سنجش تشابه رشته، در زمینه بازیابی اطلاعات، طبقه‌بندی متن، فیلترکردن هزرنامه، و خوشه‌بندی اسناد، کارکردهای مختلفی دارد.

ما بسته‌های داده را به‌شکل برداری با ۱۶ نمایه نشان می‌دهیم، که این نمایه‌ها، مقادیر 0 تا f را باتوجه به الفبای هگزادسیمال دارند.

برای نشان‌دادن این مسئله به‌شکل ریاضی، فرض کنید

P = {p1, p2, · · · , pn}دسته‌ای از بسته‌های داده، و

T = {t1, t2, · · · , tm} دسته‌ای از مقادیر مشخص درحال‌رخ دادن در P هستند.

بر اساس ریاضی یک بسته داده را می توانیم به صورت یک بردار چند بعدی $\overrightarrow{tp}$ نشان داد.

فرض کنیم $\overrightarrow{tf}\ (p.t)$ تعداد تکرار یک رخداد مقدارT ∈ t در بسته داده p ∈ P باشد نمایش برداری بسته داده p در رابطه (۱) نشان داده می شود:

$$\overrightarrow{tp} = \left(\overrightarrow{tf}(p.t1).\overrightarrow{tf}(p.t2).\ldots\overrightarrow{tf}(p.tm)\right) \quad (۱)$$

برای هر سنجش تشابه، یک آزمایش انجام می‌شود تا فرکانس سنجش تشابه برای چنین کارکردی تخمین زده شود. یک بسته داده آزمایشی مرجع با ۸۰ بسته داده مقایسه می‌شود، که ۴۰ عدد از این بسته‌ها، باهم مشابه‌اند، زیرا آنها از یک حمله یکسان و دارای امضای واحد هستند، و ۴۰ عدد دیگر، بسته‌های غیرمشابه‌اند، زیرا آنها از یک حمله دیگر هستند و ویژگی‌های مشترکی با بسته داده آزمایشی ندارند. بسته‌های داده مختلف طوری انتخاب شدند که تا حد ممکن موارد متنوع‌تری را شامل شوند، به‌طوری که دارای طول‌های مختلف بوده و امضا در افست‌های مختلف روی می‌دهد. هدف این آزمایش این است که نشان دهیم چه نوع سنجشی برای مقایسه بسته داده DoS مناسب‌تر است.

سنجش‌های مختلف را از دو جنبه اصلی ارزیابی می‌کنیم:

* دقت سنجش
* زمان صرف‌شده برای اجرای سنجش

دقت سنجش: دقت براساس تعداد نتایج درست که سنجش می‌دهد، تصمیم‌گیری می‌شود که در رابطه (۲) نشان داده شده است.

$$\text{دقت} = \frac{\boldsymbol{NTruePositive + NTrueNegative}}{\boldsymbol{NTotal}} \quad (۲)$$

N: تعداد کل مقایسه‌ها

Positive: دو بسته داده‌ای که مشابه هستند.

Negative: دو بسته داده‌ای که غیرمشابه هستند.

زمان صرف‌شده برای اجرای سنجش. به‌دلیل اینکه این عملیات چندین بار تکرار خواهد شد، پس یک الگوریتم سریع بسیار مطلوب نیاز است. زمان در اینجا، به‌شکل پیچیدگی زمان الگوریتم سنجش در نشانه‌گذاری O ارزیابی می‌شود.

۵-۴- انواع سنجش:

۵-۴-۱ - تشابه کسینوسی

این مدل احتمالا ساده‌ترین سنجش تشابه رشته است.

درجه تشابه بین دو بسته داده مختلف، به‌شکل کسینوس زاویه بین دو بردار سنجیده می‌شود، و این کسینوس ارتباط بین دو بردار بسته داده را نشان می‌دهد. رابطه (۳) بیانگر این موضوع است.

$$\textbf{تشابه کسینوسی } (\boldsymbol{ta.tb}) = \frac{\overrightarrow{\boldsymbol{ta}}.\overrightarrow{\boldsymbol{tb}}}{\|\overrightarrow{\boldsymbol{ta}}\|\|\overrightarrow{\boldsymbol{tb}}\|} \quad (۳)$$

یک خصوصیت مهم تشابه کسینوسی این است که مستقل از ترتیب است.

به این معنی است که کسینوس زاویه بین دو بسته داده برابر صفر است، و این یعنی اینکه بسته‌های داده مشابه‌اند. تشابه کسینوسی نشان می‌دهد که دو بسته داده، برخلاف اینکه آنها در ترتیب معکوس هستند، باهم یکسان‌اند.

مثال : دو بسته داده مثل p1= 012ab و p2= ba210 منجر به مقدار تشابه ۱ می‌شود.

الگوریتم تشابه کسینوسی در پیچیدگی زمان O(max(n,m)) اجرا می‌شود، که در آن n طول بسته داده اول و m طول بسته داده دوم است.

تشابه کسینوسی در چنین دامنه‌ای زیاد کارآمد نیست زیرا یک مقدار آستانه عالی ممکن نیست بسته‌های داده مشابه و غیرمشابه دارای نواحی مشترک باشند.

یک مقدار آستانه بهینه که دقت را به بالاترین حد ممکن می‌رساند، محاسبه می‌شود. رفتار تشابه کسینوسی را می‌توان به خصوصیت استقلال ترتیبی این سنجش ربط داد، که زیاد مناسب برای یک اندازه الفبای محدود نیست، مانند کاراکترهای هگزادسیمال که در آن بیشتر کاراکترهای الفبا، در بسته‌های داده کاملا متفاوت ظاهر می‌شوند.

جدول (۳)نتایج تشابه کسینوسی

مقدار آستانه بهینه	True Positives	False Positives	True Negatives	False Negatives	دقت
۰،۸۵	۱۳	۱	۳۹	۲۷	٪۶۵

۵-۴-۲- شاخص ژاکارد

از تقسیم تعداد اشتراک دو بسته داده بر تعداد اجتماع دو بسته داده به دست می‌آید.

این شاخص، رشته‌ها را به نمایه‌های باطول یک کاراکتر تقسیم می‌کند، و بررسی می‌کند که چگونه کاراکترهای بین دو بسته داده درمقابل مجموعه کلی کاراکترهای موجود، مشترک‌اند.

شاخص ژاکارد مقدار ۱ را برای بسته‌های داده مشابه، و صفر را برای بسته‌های داده غیرمشابه تولید می‌کند.

$$\textbf{ژاکارد شاخص(۴)} \quad (ta.tb) = \frac{|ta \cap tb|}{|ta \cup tb|}$$

دو خصوصیت مهم شاخص ژاکارد عبارتنداز:

این سنجش، تنها بخشی از نمایه‌هایی که مشترک بین دو بسته داده هستند، بدون احتساب فرکانس آنها، را آزمایش می‌کند و دارای دقت کمتری نسبت به تشابه کسینوسی است این موضوع در رابطه (۴) بیان شده است. ترتیب ظاهرشدن کاراکترها را درنظر نمی‌گیرد، و بنابراین، نتیجه کاملا یکسانی برای دو رشته معکوس می‌دهد.

الگوریتم شاخص ژاکارد، در پیچیدگی زمان O(max(n,m)) اجرا می‌شود، که در آن n طول بسته داده اول و m طول بسته داده دوم است.

نتایج شاخص ژاکارد، امیدوارکننده به نظر می رسد تا زمانیکه ما تعداد زیاد بسته‌های داده را می‌بینیم که با یک مقدار زیاد تشابه، با هم منطبق شدند.

این اتفاق، امتیاز کلی سنجش را بدتر می‌کند. این طبقه‌بندی ناکارآمد، به‌خاطر خصوصیات ذکرشده این سنجش، قابل انتظار بود، مخصوصا خصوصیت حاضر/ غایب در ترکیب با الفبای هگزادسیمال محدود، که شانس یافتن تعداد زیادی از کاراکترهای مجاز را در بسیاری از بسته‌های داده زیاد می‌کند.

جدول (۴) نتایج شاخص ژاکارد

مقدار آستانه بهینه	True Positives	False Positives	True Negatives	False Negatives	دقت
۰،۸۲	۱۲	۰	۴۰	۲۸	٪۶۵

۵-۴-۳-فاصله لونشتاین

فاصله لونشتاین، یک سنجش تشابه رشته‌ای "فاصله- ویرایش" است. این سنجش، کمترین تعداد مورد نیاز واردکردن‌ها، حذف‌کردن‌ها، و جایگزین‌ها برای تبدیل رشته a به رشته b را برمی‌گرداند.

امتیاز فاصله لونشتاین طوری تنظیم شد تا یک امتیاز بین صفر برای بسته‌های داده غیرمشابه، و ۱ برای بسته‌های داده مشابه برگرداند، به‌شکلی که:

$$\textbf{New Score} = 1 - \frac{\textbf{old score}}{\textbf{max}(\textbf{length}(\textbf{payload1}),\textbf{lenght}(\textbf{payload2}))} \quad (۵)$$

اگر دو بسته داده مشابه باشند، مقدار oldScore صفر، و مقدار NewScore، ۱ خواهد بود. اگر دو بسته داده غیرمشابه باشند، مقدار oldScore برابر جمع جایگزین‌های هر کاراکتر در بسته داده کوتاهتر، و ورودی‌های کاراکترهای باقی‌مانده از بسته داده بلندتر است، به‌طوری که oldScore برابر طول رشته بلندتر و درنتیجه مقدار NewScore برابر صفر خواهد بود. رابطه (۵) برای محاسبه مقدار NewScore در فاصله لونشتاین می‌باشد.

الگوریتم فاصله لونشتاین در پیچیدگی زمان O(n*m)اجرا می‌شود، که در آن، n: طول بسته داده اول، و m: طول بسته داده دوم است.

بیشتر بسته‌های داده غیرمشابه، دارای مقدار تشابه کم هستند و برعکس. فاصله لونشتاین، به‌طور غیرمستقیم، شمار و ترتیب کاراکترها را درنظر می‌گیرد.

تشابه لونشتاین دارای نتایج بهتری است، چراکه بیشتر بسته‌های داده غیرمشابه، دارای مقدار تشابه کم هستند و برعکس. این اتفاق می‌تواند به این دلیل باشد که فاصله لونشتاین، به‌طور غیرمستقیم، شمار و ترتیب کاراکترها را درنظر می‌گیرد. هرچند، برای کارکرد ما، نتایج هنوز بهینه نیستند.

جدول (۵)نتایج فاصله لونشتاین

مقدار آستانه بهینه	True Positives	False Positives	True Negatives	False Negatives	دقت
۰،۲۵	۲۴	۰	۴۰	۱۶	٪۸۰

۵-۴-۴- امتیاز هم‌راستایی اسمیت-واترمن

الگوریتمی است که هدف تعیین نواحی مشابه بین دو رشته را دارد. این کار را توسط یافتن بهترین هم‌راستای محلی آن دو رشته انجام می‌دهد.

امتیاز انطباق: وقتی افزوده می‌شود که دو کاراکتر هم‌راستا شده فعلی در دو رشته، برهم منطبق باشند.

امتیاز عدم‌انطباق: به‌عنوان جریمه درنظر گرفته می‌شود و وقتی کم می‌شود که دو کاراکتر هم‌راستاشده فعلی در دو رشته، نامنطبق باشند.

امتیاز شکاف: به‌عنوان جریمه درنظر گرفته می‌شود و وقتی کم می‌شود که به یک عملیات واردکردن یا حذف‌کردن کاراکتر نیاز باشد.

امتیاز انطباق: ۱+

جریمه عدم‌انطباق: ۱-

جریمه شکاف: ۱-

الگوریتم اسمیت-واترمن تلاش می‌کند تا بهترین هم‌راستایی بسته‌های داده مقایسه‌شده را بیابد. این الگوریتم در مورد مسئله ما، امضا را در دو بسته داده مشابه هم‌راستا خواهد کرد، اما در انجام این کار در بسته‌های داده غیرمشابه شکست خواهد خورد.

جدول (۶)نتایج امتیاز هم‌راستایی اسمیت-واترمن

مقدار آستانه بهینه	True Positives	False Positives	True Negatives	False Negatives	دقت
۰,۲۲	۴۰	۰	۴۰	۰	٪۱۰۰

۵-۴-۵- طولانی‌ترین زیردنباله مشترک

این روش هرچند که دقیقا یک سنجش تشابه نیست، اطلاعات خوبی درمورد تشابه محتوای دو بسته داده بدست می‌دهد. هدف مسئله طولانی‌ترین زیردنباله مشترک، یافتن طول طولانی‌ترین زیردنباله کاراکترهای پشت‌سرهم یا غیرپشت‌سرهم است که در دو رشته، تکرار شده‌اند.

برای مثال، فرض کنید s1=012ab44458 و s2= 0932ab48323 بنابراین:

LCSubsequence (s1, s2) = 02ab48

یک خصوصیت مهم طولانی‌ترین زیردنباله مشترک این است که، ترتیب ظاهرشدن کاراکترها در رشته‌ها را درنظر می‌گیرد در نتیجه دو رشته معکوس را تشخیص می دهند.

پیچیدگی زمان O(n*m) حل می‌شود، که در آن n طول بسته داده اول و m طول بسته داده دوم است.

$$NewScore = \frac{oldscore}{min(lenght(payload1), lenght(payload2))} \quad (۶)$$

نتایج طولانی‌ترین زیردنباله مشترک که با استفاده از رابطه (۶) محاسبه می‌شود، بدتر از مقدار مورد انتظار بود، مخصوصا با بسته‌های داده بشدت غیرمشابه احتمالا به‌دلیل الفبای محدود هگزادسیمال است، که احتمال حضور همه یک بسته داده را به‌عنوان زیردنباله یکی دیگر، بیشتر می‌کند، خصوصا وقتی که یک بسته داده به‌مقدار قابل توجهی، کوتاهتر از دیگری است.

جدول (۷)نتایج طولانی‌ترین زیردنباله مشترک

مقدار آستانه بهینه	True Positives	False Positives	True Negatives	False Negatives	دقت
۰,۷۲	۴۰	۲۸	۱۲	۰	٪۶۵

۵-۴-۶-طولانی‌ترین زیررشته مشترک

گرچه این معیار، یک سنجش تشابه سرراست نیست، طولانی‌ترین زیررشته مشترک می‌تواند تخمین خوبی از مقدار تشابه بسته‌های داده براساس طول طولانی‌ترین زیررشته‌ای بدهد که بسته‌ها در آن مشترک‌اند، که در آن، یک زیررشته، گروهی از کاراکترهای پشت‌سرهم است.

با فرض همان مثال داده‌شده برای سنجش طولانی‌ترین زیردنباله مشترک، فرض کنید s1=012ab44458و s2= 0932ab48323، بنابراین:

LCSubstring (s1, s2) = 2ab4

این به آن دلیل است که بیشترین امتیازی که بوسیله طولانی‌ترین زیررشته مشترک بازگردانی می‌شود، وقتی که دو بسته داده، یکسان‌اند، یا یکی زیرمجموعه دیگری است، برابر طول بسته داده کوتاهتر است، درحالی‌که، وقتی که دو بسته داده غیرمشابه‌اند، کمترین امتیاز برابر صفر است.

طولانی‌ترین زیردنباله مشترک با استفاده از برنامه‌نویسی پویا، در یک روش پیچیدگی زمان O(n*m) حل می‌شود، که در آن، n: طول بسته داده اول، و m: طول بسته داده دوم است.

$$NewScore = \frac{oldscore}{min(lenght(payload1).lenght(payload2))} \quad (۷)$$

روش طولانی‌ترین زیررشته مشترک نیز به نتایج بهینه رسید. این نتیجه را می‌توان باتوجه به این واقعیت توضیح داد که در بسته‌های داده مشابه، طولانی‌ترین زیررشته مشترک، همان امضا است، درحالی‌که در بسته‌های داده غیرمشابه، این مورد برقرار نیست و طولانی‌ترین زیررشته مشترک، دارای طول کمتری است. رابطه (۷) بیانگر این موضوع می باشد.

به‌دلیل نتایج دقیق کسب شده در روش طولانی‌ترین زیررشته مشترک، آزمایشی دیگر بروی همان دسته داده انجام شد، تا طول‌های طولانی‌ترین زیررشته‌های مشترک بین بسته داده اصلی و بسته‌های داده مشابه و غیرمشابه، بدست آید. بسته‌های داده مشابه، دارای زیررشته مشترک طولانی‌تری نسبت به بسته‌های داده غیرمشابه هستند، یک الگوریتم در یک روش پیچیدگی زمان O(max(n,m)) اجرا می‌شود، که در آن، n: طول بسته داده اول، و m: طول بسته داده دوم است، و این الگوریتم، در این تحقیق، با نام الگوریتم "کمترین زیررشته مشترک" شناخته می‌شود.

جدول (۸)نتایج طولانی‌ترین زیررشته مشترک

مقدار آستانه بهینه	True Positives	False Positives	True Negatives	False Negatives	دقت
۰,۲	۴۰	۰	۴۰	۰	٪۱۰۰

جدول (۹)جمع بندی نتایج

سنجش تشابه	دقت	پیچیدگی زمانی
تشابه کسینوسی	٪۶۵	O(max(n, m))
شاخص ژاکارد	٪۶۵	O(max(n, m))
فاصله لونشتاین	٪۸۰	O(n ∗ m)
امتیاز هم‌راستایی اسمیت-واترمن	٪۱۰۰	O(n ∗ m)
طولانی‌ترین زیردنباله مشترک	٪۶۵	O(n ∗ m)
طولانی‌ترین زیررشته مشترک	٪۱۰۰	O(n ∗ m)
الگوریتم کمترین زیررشته مشترک	٪۱۰۰	O(max(n, m))

۶- ارزیابی روش پیشنهادی و نتایج

۶-۱- تولید امضا

تولید امضا، فرآیند استخراج صورت مشترک که نمایانگر حمله است. این امضا، رایج‌ترین محتوای اشتراکی در ردپای حمله اصلی می باشد. برای توانایی تولید یک امضا، ما از روش‌های ذکرشده برای فیلترکردن حمله و طبقه‌بندی بسته‌های داده استفاده کردیم.

الگوریتم طولانی‌ترین زیررشته مشترک، برای استخراج امضا از بسته‌های داده مختلف، انتخاب گردید.

دو الگوریتم ارائه شده است، اولی، براساس روش IPهای منبع، به‌عنوان یک روش برای فیلترکردن حمله، درحالی‌که دومی براساس تکراری‌ترین بسته‌های داده برای فیلترکردن حمله قرار دارد. هردو الگوریتم برای طبقه‌بندی بسته‌های داده، روش کمترین زیررشته اشتراکی را استفاده می‌کند.

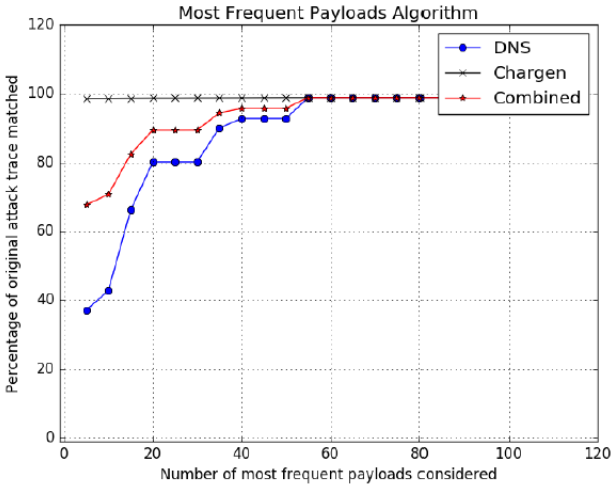
نتایج تولید امضا برای DNS نتایج دقیقتری حاصل شده که با بازرسی IPهای کمتری بدست آمده است.

برای هردو مورد DNS و Chargen، درصد بالایی از ردپای حمله، تا ٪۹۹، را می‌توان با امضاهای تولیدشده انطباق داد.

با پردازش ترافیک IPهای منبع بیشتر، یک امضای دقیقتر تولید می‌شود. این بهبود دقت به‌خاطر این است که با درنظرگرفتن IPهای منبع بیشتر، صورت‌های بیشتری از حمله پردازش می‌شود، تا اینکه یک نقطه اشباع حاصل شود.

این نکته را نیز باید گفت که از هر ۱۰ حمله Chargen آزمایشی، الگوریتم IPهای منبع نتوانستند یک امضا با طول بیش یا مساوی کمترین طول مشخص‌شده (۲۰ کاراکتر) استخراج کنند.

نتایج الگوریتم تکراری‌ترین بسته‌های داده، در شکل آمده است. می‌توان دید که برای Chargen، یک امضای دقیق، که نمایشگر ٪۹۹ ترافیک حمله است، درهمان زمان ابتدایی بازرسی ۵ عدد از تکراری‌ترین بسته‌های داده تولید شده است.



شکل ۵ نتایج الگوریتم تکراری‌ترین بسته‌های داده

از آزمایش‌های پایانی، می‌توان نتیجه گرفت که الگوریتم تکراری‌ترین بسته‌های داده منجربه نتایج دقیقتری نسبت به الگوریتم IPهای منبع بیشترین فرستنده می‌شود. این نتیجه‌گیری به این دلیل است که حملات یکسان برای هردو الگوریتم اجرا شد، و الگوریتم تکراری‌ترین بسته‌های داده، هیچ بسته داده خارجی‌ای را نشان نداد، و توانست به‌طور مؤثری یک امضای حمله تولید کند.

۷- نتیجه‌گیری

هدف تحقیق در این مقاله یک نمایش کامل برای تولید خودکار یک امضا برای حملات DDoS برپایه بسته داده می باشد. این کار ابتدا با شناسایی انواع حملاتی که مناسب برای تولید امضای برپایه بسته داده هستند، انجام شد و یک معیار برای ارزیابی اینکه آیا یک حمله، مناسب برای بازرسی بسته داده، است یا نه، پیشنهاد شد. دربین رایج‌ترین نوع حملات در چند سال گذشته، حملات برپایه DNS و برپایه Chargen دارای احتمال بیشتری در داشتن امضاهای اشتراکی بسته داده هستند.

شش نوع سنجش تشابه بررسی شدند تا بتوان روابط بین ترافیک را یافته و تکراری‌ترین الگو درون ترافیک را شناسایی کرد، که این الگو همان حمله است. الگوریتم‌های اسمیت- واترمن و طولانی‌ترین زیررشته مشترک، بیشترین دقت را در طبقه‌بندی بسته داده نشان دادند. سپس یک امضا برای الگوی حمله تولید شد که بیشترین مقدار ممکن از پاکت‌های حمله را شناسایی می‌کند.

درنهایت امضاهای تولیدشده، به‌عنوان قوانین جدید برای Snort IDSاضافه شدند، و با توانایی شناسایی درست و مدیریت پاکت‌های حمله ارزیابی شدند. درطول فرآیند ارائه‌شده، هدف تولید خودکار امضا حاصل شد.

مراجع

[1] What is a DDoS attack?, 2017, https://www. verisign.com/en_US/security-services/ddos-protection/ what-is-a-ddos-attack/index.xhtml.

[2] Fall, K., Stevens, R., "TCP/IP illustrated", The protocols, Pearson Education, vol. 1, Nov. 2011.

[3] Security report, Akamai's [state of the internet], Volume 4, Number 1, Quarter1, 2017, https:// www. akamai.com/us/en/multimedia/documents/state-of-the-internet/q1-2017-state-of-the-internet-security-report.pdf.2010.

[4] Wang, K., Stolfo, S., "Anomalous Payload-Based Network Intrusion Detection - Recent Advances in Intrusion Detection", Springer, Vol. 4, pp. 203–222, 2004.

[5] Wang, K., Parekh, J., Stolfo, S., "Anagram: A Content Anomaly Detector Resistant to Mimicry Attack - Recent Advances in Intrusion Detection", Springer, pp. 226–248, 2006

[6] Nwanze, N., Summerville, D., "Detection of Anomalous Network Packets Using Lightweight Stateless Payload Inspection - 2008 33rd IEEE Conference on Local Computer Networks", IEEE, pp. 911–918, Oct. 2008.

[7] Thakar, U., Dagdee, N., Varma, S., "Pattern Analysis and Signature Extraction for Intrusion Attacks on Web Services - International Journal of Network Security & Its Applications", Vol. 2, pp. 190-205, Jul. 2010.

[8] Burges, C., "A tutorial on support vector machines for pattern recognition - Data mining and knowledge discovery", pp. 121-167, 1998.

[9] Katkar, V., Bhirud, S., "Novel DoD/DDoS Attack Detection and Signature Generation - International Journal of Computer Applications", Vol. 47, pp. 18–24, Jun. 2012.

[10] Hwang, K., Cai, M., Chen, Y., Qin, M., "Hybrid intrusion detection with weighted signature generation over anomalous internet episodes - IEEE Transactions on Dependable and Secure Computing", IEEE, Vol. 4, pp. 41-5

[11] Gusfield, D., "Algorithms on strings, trees and sequences: computer scienceand computational biology", ACM SIGACT News, Cambridge University Press, vol. 28, Dec. 1997.

[12] Perdisci, R., Lee, W., Feamster, N., "Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces - In NSDI", Vol. 10, pp. 14, Apr. 2010.

[13] Schwartz, E. J., Avgerinos, T., & Brumley, D. (2010, May). All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask). In 2010 IEEE symposium on Security and privacy (pp. 317-331). IEEE.

[14] Kreibich, C.,Crowcroft, J., "Honeycomb: Creating Intrusion Detection Signatures Using Honeypots - ACM SIGCOMM computer communication review", Vol. 34, pp. 51–56, Jan. 2004.

[15] Mohammed, M., Chan, H., Ventura, N., "Honeycyber: Automated Signature Generation for Zero-day Polymorphic Worms - 2008 IEEE Military Communications Conference", IEEE, pp. 1–6, Nov. 2008.

[16] Portokalidis, G., Slowinska, A., Bos, H., "Argos: An Emulator for Fingerprinting Zero-Day Attacks for Advertised Honeypots With Automatic Signature Generation - ACM SIGOPS Operating Systems Review", Vol. 40, pp. 15–27, Apr. 2006.