تحلیلگر حمله انکار سرویس توزیع شده (DDoS): با استفاده از JPCAP و WinCap

چکیده

امروزه کامپیوترها و شبکه‌های آنها با سرعت و انعطاف‌پذیری باعث پیچیده و متنوع شدن سیستم‌هایی شده‌اند که با یکدیگر ارتباط برقرار می‌کنند. همیشه نیازی به ابزار خاص پیچیده و با سرعت بالا برای تحلیل بسته/شبکه وجود دارد. نظارت بر ترافیک شبکه همیشه به سادگی و صراحتی نیست که در تئوری مطرح می‌شود، بلکه منجر به ایجاد تغییر در بسیاری از موارد می‌گردد. این ابزار همچنین قادر به ارائه‌ی نمایش گرافیکی و آمارهایی از خطرها و گزارش‌ها می‌باشند. بسیار مهم است که در حین توسعه‌ی این ابزار، درک درستی از شرایط حملات نفوذی، پروتکل‌های شبکه و رفتار سیستم‌ها، درک کاربر و انطباق و سازگاری کاربر داشته باشیم. ما قصد داریم که آن دسته از ابزار تشخیص و تحلیل ترافیک شبکه را ارائه دهیم که به طور عمده بر روی حمله‌ی DDoS تمرکز دارند. ما در این مقاله از کتابخانه‌ی JPCAP به همراه WinCap برای ضبط و ثبت بسته‌های شبکه جهت تشخیص و تحلیل ترافیک شبکه بر روی هدف اصلی حمله‌ی DDoS استفاده می‌کنیم.

کلمات کلیدی: گوش‌دهنده به شبکه (Sniffer)؛ ضبط بسته‌ها؛ Jpcap؛ WinCap؛ تشخیص نفوذ، حمله‌های انکار سرویس توزیع شده[1] (DDoS).

تحلیلگر بسته یا sniffer (گوش‌دهنده به شبکه) در واقع ابزاری برای جمع‌آوری داده‌ها (درخواست و پاسخ) است که میان وسایل مختلف شبکه و همچنین کاربران نهایی شبکه رد و بدل می‌شوند [۱۱]. این sniffer‌ها با پروتکل‌های مختلف داده‌ها سروکار دارند. به محض جاری شدن داده‌ها در سرتاسر شبکه، sniffer شروع به ضبط هر بسته می‌کند، و در صورت نیاز به رمزگشایی داده‌های خام بسته‌ها نموده، و مقادیر فیلدهای مختلف موجود در بسته را نمایش می‌دهد، و محتوای آن را طبق RFC مناسب یا دیگر مشخصات تجزیه و تحلیل می‌نماید [۲][۳]. در حین تحلیل، sniffer‌ها با بسته‌هایی مواجه می‌شوند که نباید بخشی از داده‌ها باشند یا حاوی داده‌هایی هستند که برای نفوذ، ویروس‌ها، رفتارهای مخرب یا هر گونه نقض سیاست شبکه در نظر گرفته

[1] Distributed Denial of Service (DDoS)

شده‌اند. این ابزار شامل ابزارهای آماری شبکه، تشخیص نفوذ، دیمون‌های ورود و بررسی پورت[1]، snifferهای مربوط به رمز عبور[2]، مسموم‌کننده‌های بسته‌های ARP[3]، مسیریاب‌های ردیاب، و غیره می‌باشند [۱].

یک حمله‌ی انکار سرویس توزیع شده (DDoS) در واقع تلاشی است برای از دسترس خارج نمودن سرویس‌های آنلاین به وسیله‌ی غرق ساختن آنها با ترافیکی که از مبدأهای مختلف می‌آیند [۱۰][۵]. آنها طیف وسیعی از منابع مهم از جمله بانک‌ها و وب‌سایت‌های خبری را هدف قرار می‌دهند و چالش عمده‌ای را برای این امر به وجود می‌آورند که افراد مطمئنی بتوانند اطلاعات مربوطه را منتشر نموده و به آنها دسترسی داشته باشند [۴].

این مقاله به این صورت سازماندهی شده است که در بخش ۱ به ارائه‌ی درک درستی از WinCap و Jpcap می‌پردازیم، بخش ۲ برای ارائه‌ی الگوریتم پیشنهادی و بخش ۳ نیز برای جزئیات آزمایشی در نظر گرفته شده است و در ادامه در بخش ۴ نیز نتایج آزمایشات ارائه می‌گردد.

۱. WinCap و Jpcap

ضبط بسته‌ها در Windows (WinCap) امکان دسترسی به شبکه را در سطح پایین در محیط سیستم‌عامل ویندوز ارائه می‌دهد. این ابزار می‌تواند ترافیک شبکه را به همراه پشته‌ی پروتکلی و فرآیند انتقال دهد. همچنین اجازه‌ی فیلتر کردن بسته‌ها را در سطح هسته می‌دهد. این ابزار یکی از درایورهای مفید شبکه در Windows است، که دسترسی به شبکه را در سطح پایین به همراه کنترل هسته‌ای ارائه می‌دهد که آن را انجام می‌دهد. این ابزار برای بسیاری از سیستم‌ها به صورت تجاری مورد استفاده قرار گرفته و همچنین دارای نسخه‌ی UNIX با نام libcap می‌باشد. ابزارهای تحلیلگر بسته، تحلیلگر ترافیک، نظارت ترافیک شبکه، سیستم تشخیص نفود و به طور عمده آنچه در snifferها موجود است توسط WinCap به صورت یکپارچه در آمده است [۱۳][۱۴][۶].

Jpacap در واقع یک کتابخانه‌ی متن‌باز مبتنی بر جاوا است که به زبان C و جاوا پیاده‌سازی شده و به منظور دسترسی به ترافیک شبکه از جمله ثبت و ضبط و ارسال بسته‌ها بر روی شبکه می‌باشد. این کتابخانه به طور عمده به همراه WinCap (در Windows) / libcap (در UNIX) در برنامه‌های کاربردی مبتنی بر جاوا مورد

[1] port knocking daemons
[2] password sniffers
[3] ARP poisoners

استفاده قرار می‌گیرد. Jpcap به ضبط بسته‌های Ethernet، TCP، UDP، IPv4، IPv6، ICMP، ARP/RARP و غیره می‌پردازد. این کتابخانه بر روی سیستم‌های عامل‌هایی از جمله Windows (نسخه‌های 98، 2000، XP و Vista)، لینوکس (نسخه‌های Fedora و Ubuntu)، مکینتاش (سیستم‌عامل Mac نسخه‌ی Darwin)، FreeBSD، و Solaris [۹] آزمایش شده و نتایج موفقی را به همراه داشته است.

Jpcap مجموعه‌ای از کلاس‌ها و واسط‌های جاوا است. این مجموعه جزئیات غیرضروری را از کاربر مخفی می‌کند، جزئیاتی از ترافیک شبکه‌ی در حال ضبط، ارسال بسته بر روی شبکه و پروتکل‌های در حال نمایش. Jpcap به صورت داخلی توسط تعداد زیادی از کلاس‌ها و واسط‌های شبکه به همراه ملحقات آنها کنترل و اجرا می‌شود. رابط بومی جاوا نقش بسیار مهمی در اتصال مجموعه‌ی شامل تمام کلاس‌ها و واسط‌ها به یکدیگر به عنوان یک جزء دارد [۸].

Jpcap در سمت جاوا از کلاس‌های متعدد جاوا تشکیل شده است. این کلاس‌ها در واقع نظیر و همتای ساختارهای بومی C هستند که توسط libpcap فراهم شده‌اند [۷]. به عنوان مثال، وقتی کاربر یک نمونه از شئ Pcap را بازیابی می‌کند، این شئ شامل یک اشاره‌گر حافظه به ساختار pcap_t در C است [۷]. هنگامی که هر روش غیر-ایستا به فراخوانی کلاس جاوا می‌پردازد، از مرجع ذخیره شده به ساختار بومی C استفاده خواهد کرد تا تابع درخواست شده را اجرا نماید. در صورت نیاز، همان شئ به تمام دیگر ساختارها از قبیل Pcap نیز اعمال می‌شود. تمام آنها دارای جفتی از زبان C هستند و یک مرجع حافظه را به ساختار C متناظر خود نگه‌داری می‌کنند [۷]. برای اهداف امنیتی و محافظتی جاوا، خواننده مجاز نیست که به طور مستقیم به این ساختارهای C دسترسی داشته باشد، و تمام توابع متناظر کتابخانه‌ی *libpcap* به عنوان روش‌های جاوا ارائه شده است. بنابراین، رابطه‌ی بسیار نزدیکی بین هر شئ جاوا و ساختار بومی C متناظر آن وجود دارد، و همین رابطه نیز در توابع *libpcap* و روش‌های جاوای متناظر با آنها اعمال می‌شود.

ما از WinCap برای گرفتن ترافیک در داخل سیستم استفاده می‌کنیم تا آن را تحلیل نموده و DDoS خاصی را تشخیص دهیم. در سطح عملکردی، jpcap با بررسی مقدار آستانه به تحلیل بسته‌ها خواهد پرداخت تا تصمیم بگیرد که بسته‌های وارد شده مشکوک هستند یا ترافیک نرمالی می‌باشند.

۲. الگوریتم پیشنهادی

۱. به دست آوردن و باز کردن شبکه.

```
NetworkInterface[] device = jpcapCaptor.getDeviceList();
captor = JNetPcapCaptor.openDevice(device[index], snaplen, promics, timeout);
```

که snaplen برابر با حداکثر تعداد بایت‌ها، promics برابر با حالت ترافیک و timeout نیز برابر با مقادیر زمانی timeout بر حسب میلی‌ثانیه می‌باشند.

۲. قرار دادن نوع ترافیک به صورت بی‌سیم یا اترنت (Ethernet)

این امر به کاربر اجازه خواهد داد تا بر نوع ترافیک نظارت داشته باشد، اینکه نوع آن یک LAN بی‌سیم است یا یک LAN از نوع اترنت (Ethernet) می‌باشد.

۳. تنظیم مقدار آستانه، در غیر این صورت مقدار آن به صورت پیش‌فرض محاسبه خواهد شد.

۴. بررسی ترافیک ورودی

بررسی آدرس IP مبدأ و مقصد،

اعمال پشته پروتکلی مناسب و ایجاد نخ،

بررسی آدرس IP مبدأ و مقصد به همراه شماره Port

بررسی اینکه آیا آدرس IP و شماره Port یکسانی در صف پشته یافت می‌شود

اجرای توابع محاسبه‌ی DDoS.

۵. محاسبه احتمال DDoS و نمایش گزارش

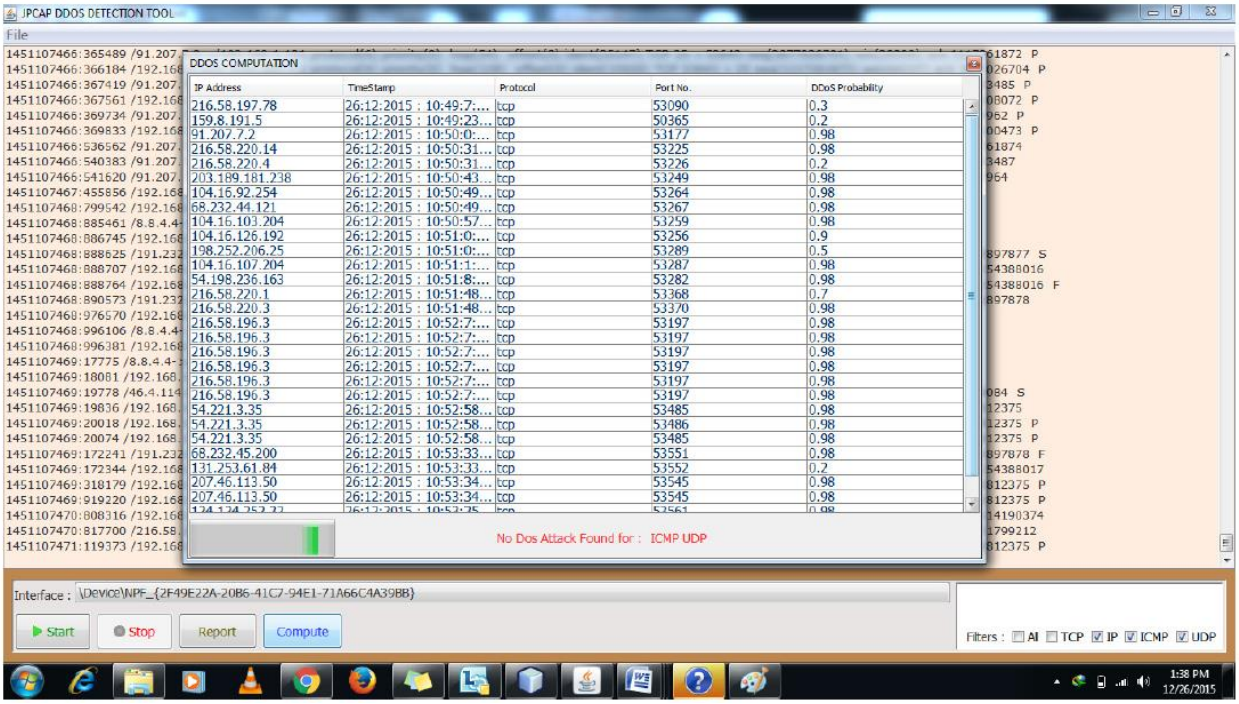
۶. تولید گزارش تحلیلگر بر اساس هشدارهای تشخیص و اطلاع‌رسانی به کاربر

۳. آزمایش‌ها و نتایج

ما از KDD استفاده می‌کنیم، ما در شبکه‌ای با اینترنت به اشتراک گذاشته شده در WAN و LAN از طریق یک شبکه‌ی بی‌سیم و سیمی هستیم. آزمایش بر روی سیستم‌عامل Windows 7، ۶۴ بیتی با پردازنده‌ی i3 انجام شده است. برای آزمایش، ما یک وب سرور اختصاصی را برای میزبانی یک وب‌سایت محلی مستقر

نموده‌ایم. سیستم حمله‌کننده با برنامه‌ی ping به اجرای حمله بر روی سیستم میزبان موردنظر می‌پردازد. ما هر دو برنامه‌ی مهاجم و میزبان را اجرا نموده و به تحلیل سیستم برای تشخیص حمله DDoS پرداخته‌ایم.

از آنجایی که ما بر روی حمله‌ی DDoS تمرکز نموده‌ایم، به همین دلیل تحلیل راه‌حل پیشنهادی خود را بر روی LAN انجام می‌دهیم، چرا که در LAN چندین میزبان به یکدیگر متصل شده‌اند و میزبان خاصی را هدف قرار داده‌اند تا ارسال بسته‌ها را به صورت سیل‌آسا به آن انجام دهند، یعنی در اینجا، در آزمایش ما به طور پیوسته عمل ping را انجام دهند. به طور کلی در حین حمله‌ی DDoS، ما می‌توانیم حمله را مانند شکل زیر تشخیص دهیم:



شکل ۱. تشخیص حمله DDoS

در هر آزمایش، ما ترافیک زنده را از شبکه‌ی دارای اینترنت به اشتراک گذاشته شده، ثبت و ضبط نموده‌ایم. ما حمله‌ی DDoS انجام شده با سیستم برنامه‌ی ping را توسط آدرس IP آنها، برچسب‌زمانی پورت (timestamp port) و با احتمال تشخیص داده‌ایم، احتمالی که شدت حمله را نشان می‌دهد. تشخیص ما برای DDoS بسیار کارآمد بوده است، به گونه‌ای که قادر به تشخیص حملات سیل‌آسای TCP و UDP به صورت موثرتر و کارآمدتر بوده‌ایم، این حملات به طور عمده در حمله‌ی DDoS یافت می‌شوند. بسته‌های ICP می‌توانند

برای چندین حمله از حمله‌ی smurf مورد استفاده قرار گیرند؛ حمله‌ی سیل‌آسای SYN نیز به صورت موثر در راه‌حل ترافیکی ما تشخیص داده شده است.

۴. نتیجه‌گیری

ما در این مقاله، یک راه‌حل الگوریتمی برای تشخیص DDoS از روی ترافیک شبکه را ارائه کردیم. از آنجا که پارامترهای متعددی در یک حمله‌ی DDoS تغییر می‌کنند، به همین دلیل این حمله باید به طور موثری برای رسیدگی به آسیب وارد شده به ترافیک شبکه در نظر گرفته شود. ما تاکید زیادی بر روی بسته‌های TCP، ICMP و UDP داریم، چرا که این بسته‌ها مستعد استفاده شدن در حمله‌های DDoS به صورت سیل‌آسا یا پینگ شده (ping) هستند. برای رسیدگی به حمله DDoS، ما علاوه بر بررسی مقدار آستانه‌ی تعیین شده برای بسته‌های مستعد، همچنین به بررسی آدرس IP مبدأ و مقصد می‌پردازیم تا مشخص شود که کاربر ارسال‌کننده یا دریافت‌کننده‌ی این بسته‌ها در شبکه احراز هویت شده و تایید شده می‌باشد یا خیر. در این نقطه‌ی بررسی، ما می‌توانیم یک راه‌حل واقعی را برای تشخیص DDoS ارائه دهیم که قادر به تنظیم واسط، پروتکل و فایل‌های ترافیک شبکه باشد.