Journal of Information and Communication Technology

Volume 17, Issue 63-64, Spring and Summer 2025, pp. 123-138

Confidential Payload Attribution on Encrypted Traffic of Enterprise Networks

Seyed Mohammad Hosseini[1*], Amir Hossein Jahangir[2], Mahdi Soltani[3]

[1] Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran
[2] Computer Engineering Department, Sharif University of Technology, Tehran, Iran
[3] Computer Engineering Department, Sharif University of Technology, Tehran, Iran

Received: 26 November 2024, Revised: 10 January 2025, Accepted: 10 January 2025
Paper type: Research

Abstract

The widespread use of encryption protocols is accompanied by an increased risk of organizational-level security devices becoming ineffective. When network traffic is encrypted, many security tasks such as intrusion detection and network forensics that rely on processing content of flows' payloads become ineffective. Existing practical approaches to this problem are based on TLS interception methods, which not only violate confidentiality but also impose security issues. This paper introduces a confidential payload attribution system called "JormYab". JormYab is a practical approach to enable data attribution on standard encrypted traffic for organizational networks. JormYab, which can be easily deployed in an enterprise network, is based on a simple traffic digesting mechanism and does not violate confidentiality. Our practical and realistic evaluations show that JormYab can store a history of standard encrypted traffic of an enterprise network for use in network forensic investigations. The realistic scenarios we have used in our research also reveal common challenges and problems in the process of payload attribution investigations, and based on them, we discuss effective methods to address the issues.

Keywords: Network forensics; Payload attribution; Encrypted traffic, Confidentiality.

[*] Corresponding Author's email: m-hosseini@sbu.ac.ir

دوفصلنامه
فناوری اطلاعات و ارتباطات ایران

سال هفدهم، شماره‌های ۶۳ و ۶۴ ، بهار و تابستان ۱۴۰۴، صفحه ۱۲۳ الی ۱۳۸

انتساب داده روی ترافیک رمز شده سازمانی بدون نقض محرمانگی

سید محمد حسینی[1*]، امیرحسین جهانگیر[2]، مهدی سلطانی[3]

[1] دانشکده مهندسی و علوم کامپیوتر، دانشگاه شهید بهشتی، تهران، ایران

[2] دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

[3] دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

تاریخ دریافت: ۱۴۰۳/۰۹/۰۶ تاریخ بازبینی: ۱۴۰۳/۱۰/۲۱ تاریخ پذیرش: ۱۴۰۳/۱۰/۲۱

نوع مقاله: پژوهشی

چکیده

استفاده فراگیر از پروتکل‌های رمزنگاری با افزایش خطر ناتوانی دستگاه‌های امنیتی سطح سازمانی همراه است. وقتی ترافیک شبکه رمزگذاری می‌شود، بسیاری از وظایف امنیتی مانند تشخیص نفوذ و جرم‌شناسی شبکه که به پردازش محتوای جریان‌ها وابسته‌اند، بی‌اثر می‌گردند. رویکردهای عملی موجود برای این مشکل بر اساس روش رهگیری TLS هستند که نه تنها محرمانگی را نقض ، بلکه مشکلات امنیتی نیز ایجاد می‌کنند. این مقاله یک سامانه انتساب داده محرمانه به نام «جرم‌یاب» را معرفی می‌کند. جرم‌یاب یک رویکرد عملی برای فراهم کردن امکان انتساب داده بر روی ترافیک رمزگذاری شده استاندارد برای شبکه‌های سازمانی است. جرم‌یاب که به راحتی در شبکه‌های سازمانی قابل استقرار است، بر اساس یک سازوکار ساده مبتنی بر چکیده‌سازی ترافیک عمل می‌کند و محرمانگی را نقض نمی‌کند. ارزیابی‌های عملی و واقع‌گرایانه ما نشان می‌دهند که جرم‌یاب می‌تواند تاریخچه‌ای از ترافیک رمزگذاری شده استاندارد یک شبکه سازمانی را برای استفاده در تجسس‌های جرم‌شناسی شبکه ذخیره کند. سناریوهای واقع‌گرایانه‌ای که ما در تحقیقات خود استفاده کرده‌ایم، چالش‌ها و مشکلات عمومی در فرآیند تجسس‌های انتساب داده را نیز آشکار می‌کند و بر اساس آن‌ها، روش‌های موثری را برای رفع مشکلات مورد بحث قرار می‌دهیم.

کلیدواژگان: جرم‌شناسی شبکه، انتساب داده، ترافیک رمز شده، محرمانگی.

* رایانامهٔ نویسنده مسؤول: m-hosseini@sbu.ac.ir

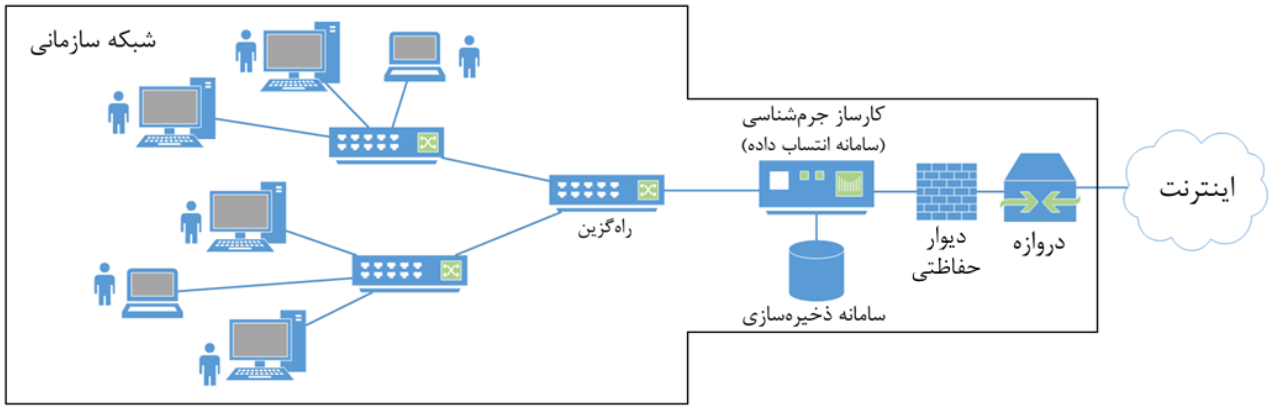
۱- مقدمه

در سال‌های اخیر، استفاده از پروتکل‌ها و برنامه‌های رمزگذاری شده مانند HTTPS به‌طور چشمگیری افزایش یافته است [۱]. از دیدگاه کاربران، استفاده از رمزگذاری در تمام ارتباطات شبکه برای اطمینان از محرمانگی[۱]، یکپارچگی[۲]، و سایر ویژگی‌های امنیتی ضروری است. این امر به رشد سریع و استفاده گسترده از پروتکل‌های رمزگذاری منجر شده است. با این حال، مزایای رمزگذاری بدون هزینه نیست. بکارگیری رمزگذاری در تمام پروتکل‌های استاندارد باعث بروز مشکلات و هزینه‌های مهمی می‌شود. مهم‌ترین و چالش‌برانگیزترین مسئله، ناتوانی دستگاه‌های امنیتی سطح سازمانی است. دستگاه‌هایی که وظایف امنیتی مانند تشخیص نفوذ، تشخیص نشت داده[۳]، نظارت والدین، بازرسی عمیق بسته‌ها[۴]، و جرم‌شناسی شبکه[۵] را انجام می‌دهند، در مواجهه با ترافیک رمزگذاری شده بی‌اثر می‌شوند. این مسائل منجر به تحمیل هزینه قابل‌توجهی به سازمان‌ها می‌شود.

در این مقاله، ما بر روی یکی از روش‌های جرم‌شناسی شبکه به نام «انتساب داده[۶]» تمرکز کرده، و به مشکلی که ترافیک رمزگذاری شده برای آن ایجاد می‌کند می‌پردازیم. روش‌های انتساب داده برای شناسایی منبع و مقصد یک رشته داده که از طریق شبکه منتقل می‌گردد، استفاده می‌شود. همان‌طور که در شکل ۱ نشان داده شده است، سامانه انتساب داده به طور معمول در لبه یک شبکه سازمانی قرار می‌گیرد و وظیفه ضبط و ثبت ترافیک شبکه سازمان برای بررسی‌های پس از حادثه[۷] را بر عهده دارد. برای بررسی حادثه، یک رشته داده که مشخصه یا نشانگر ویژگی حادثه است به سامانه انتساب داده ارائه می‌شود و سامانه با جستجوی ترافیک ذخیره‌شده، همه اتصال‌های[۸] برقرار شده شبکه (و در نتیجه مبدا و مقصدی) که آن داده را منتقل کرده‌اند، شناسایی و گزارش می‌کند. به عنوان مثال، رشته داده مورد تجسس می‌تواند امضای یک کرم[۹] باشد و سامانه انتساب داده می‌تواند منبع حمله و همچنین ماشین‌های آلوده را تشخیص دهد. به عنوان مثالی دیگر، رشته مورد تجسس می‌تواند اطلاعات محرمانه افشا شده سازمان باشد. در این حالت، سامانه انتساب داده می‌تواند برای شناسایی کارمند داخلی که اطلاعات را افشا کرده است، استفاده شود.

مانند هر سامانه امنیتی دیگری، ترافیک رمزگذاری شده باعث بی‌اثر شدن سامانه انتساب داده می‌شود. به‌طور واضح، جستجوی داده‌های آشکار[۱۰] در ترافیک ذخیره‌شده از اتصال‌های رمزگذاری شده نتیجه مفیدی ندارد. شناخته‌شده‌ترین روش برای این مشکل، روش بازرسی TLS[۱۱] [۲-۵] است که نه تنها نگرانی‌های مربوط به حریم خصوصی و محرمانگی را افزایش، بلکه امنیت را نیز به میزان زیادی کاهش می‌دهد [۲،۶-۹]. در حالی که راه‌حل‌های دیگری در مقالات برای بازرسی عمیق بسته‌ها با حفظ حریم خصوصی بر روی ترافیک رمزگذاری شده پیشنهاد شده‌اند [۲۳-۲۶]، این راه‌حل‌ها در حال حاضر عملی نیستند زیرا نیاز به استفاده از پروتکل رمزگذاری پیشنهادی آنها در هر دو طرف اتصال دارند. به عبارت دیگر، آنها نیاز به تغییر در پروتکل‌ها و برنامه‌های کاربردی فعلی در هر دو سمت اتصال (کارخواه[۱۲] و کارساز[۱۳]) دارند. علاوه بر این، آنها ترافیک را بازرسی و محتوای مخرب را بر اساس قوانین از پیش تعیین‌شده پالایش می‌کنند که برای مسئله انتساب داده قابل استفاده نیست. بنابراین، با توجه به معماری و پروتکل‌های استاندارد فعلی اینترنت، هیچ راه حل عملی برای مسئله انتساب داده با قابلیت حفظ محرمانگی در ترافیک TLS وجود ندارد.

این مقاله یک سامانه انتساب داده محرمانه به نام «جرم‌یاب» را ارائه می‌کند. جرم‌یاب یک راه‌حل عملی و ساده است که انتساب داده را بر روی ترافیک رمزگذاری شده استاندارد (TLS) یک شبکه سازمانی بدون نقض محرمانگی فراهم می‌کند. این راه‌حل بر اساس تجهیز



شکل ۱. شبکه سازمانی مجهز به سامانه انتساب داده در لبه شبکه

کامپیوترهای شبکه سازمانی به یک افزونه نرم‌افزاری است که با پردازش داده‌های آشکار بالای لایه TLS، چکیده‌های فشرده‌ای از داده‌های ترافیک ایجاد می‌کند. افزونه مزبور چکیده‌های تولیدشده را به یک کارساز جرم‌شناسی ارسال می‌کند تا برای تجسس‌های پس از حادثه ذخیره شود. کارساز جرم‌شناسی که در لبه شبکه سازمانی مستقر شده است، اتصال‌های رمز شده‌ای را که چکیده‌های خود را با استفاده از افزونه مجاز[۱۴] و معتبر[۱۵] جرم‌یاب ارائه نمی‌دهند مسدود می‌کند. چکیده‌ها قابل بازگشت به داده‌های آشکار اصلی نیستند و در نتیجه محرمانگی حفظ می‌شود. با این حال، چکیده‌ها می‌توانند برای جستجوی یک رشته داده و تعیین اینکه آیا رشته با آن‌ها مرتبط است یا خیر، استفاده شوند. هزینه این راه حل، نرخ پایینی از پاسخ‌های مثبت کاذب[۱۶] است. ما در این مورد بحث می‌کنیم که چگونه یک تحلیل‌گر جرم‌شناسی[۱۷] می‌تواند با پاسخ‌های مثبت کاذب برخورد کند و همچنین چگونه می‌توان به این راه‌حل از نظر محرمانگی اعتماد کرد.

به‌عنوان نمونه‌ای از روش پیشنهادی، ما افزونه‌ای توسعه داده‌ایم که بر روی پیمانه TLS مرورگر فایرفاکس قرار می‌گیرد. همچنین کارساز جرم‌شناسی را با استفاده از DPDK [۱۰] پیاده‌سازی کرده‌ایم تا تاخیر در انتقال بسته‌های ترافیکی را به حداقل برسانیم. آزمایش‌های عملی و واقع‌گرایانه ما نشان داده‌اند که زمانی که رمزگذاری تنها با استفاده از امنیت لایه انتقال (TLS) انجام شود، که روش معمول در اکثر برنامه‌ها مانند وب‌سایت‌ها (HTTPS) است، جرم‌یاب می‌تواند به طور موثری منبع یا مقصد رشته‌های مورد تجسس را شناسایی کند. این راه‌حل داده‌هایی را که در لایه کاربرد رمزگذاری شده‌اند، مانند برخی برنامه‌های پیام‌رسان، پوشش نمی‌دهد.

دستاوردهای این پژوهش به صورت زیر فهرست شده است:

- معرفی ابزاری به نام جرم‌یاب که راهکاری عملی و جدید برای انتساب داده بر روی ترافیک رمز شده استاندارد بدون نقض محرمانگی است. جرم‌یاب به راحتی در شبکه‌های سازمانی مستقر می‌شود بدون اینکه نیازی به تغییر پروتکل‌های استاندارد موجود به ویژه TLS باشد.

- ما برای اولین بار، آزمایش‌های واقع‌گرایانه‌ای برای ارزیابی سامانه‌های انتساب داده انجام داده‌ایم. سناریوهایی که در ارزیابی‌هایمان استفاده کردیم، چالش‌ها و مشکلات مهمی را آشکار می‌کنند که باید در هنگام اجرای فرآیند انتساب داده مورد توجه قرار گیرند. همچنین روش‌های مقابله با این چالش‌ها را مورد بحث و بررسی قرار می‌دهیم.

ساختار ادامه مقاله بدین صورت است: در بخش ۲، به طور خلاصه روش‌های انتساب داده را مرور کرده، به بررسی کارهای مرتبط و رویکردهای موجود برای مشکلی که رمزگذاری برای سامانه‌های امنیتی ایجاد می‌کند می‌پردازیم. در بخش ۳، راهکار پیشنهادی و در بخش ۴، ارزیابی‌های انجام شده را ارائه می‌کنیم. در نهایت، در بخش ۵، مقاله را جمع‌بندی می‌کنیم.

۲- پیش‌زمینه

۲-۱- انتساب داده

«انتساب» مسئله شناسایی منبع و/یا مقصد یک رشته داده است که از طریق شبکه منتقل می‌شود. سامانه انتساب داده که معمولا با استفاده از یک کارساز جرم‌شناسی در لبه یک شبکه سازمانی مستقر می‌شود، تاریخچه‌ای از ترافیک را ذخیره می‌کند تا به وسیله آن بتوان مبداء و مقصد هر رشته داده منتقل شده از طریق شبکه را جستجو کرد. در تجسس‌های پس از حادثه، تحلیل‌گر جرم‌شناسی تاریخچه ذخیره‌شده برای یک رشته داده (مثلا امضای یک کرم) را بررسی می‌کند. سامانه انتساب داده اطلاعات (مانند منبع و مقصد) تمام اتصال‌هایی که آن رشته را حمل کرده‌اند گزارش می‌دهد.

ساده‌ترین سامانه انتساب داده یک ضبط‌کننده ترافیک خام شبکه است. با ضبط ترافیک می‌توان هر حادثه‌ای در شبکه را بررسی کرد. به‌عنوان مثال، تحلیل‌گر جرم‌شناسی می‌تواند با جستجوی اطلاعات افشا شده یا امضای یک کرم در ترافیک ضبط‌شده، مبداء و مقصد آن را بیابد. چالش‌برانگیزترین مشکل این راهکار، ذخیره حجم بالای داده است که بسیار پرهزینه است. مشکل دیگر آن نقض محرمانگی است. امکان دسترسی به اطلاعات شخصی کاربران یا اطلاعات حساس یک سازمان از طریق ضبط ترافیک شبکه، مانع استفاده از ضبط‌کننده‌های ترافیک در بسیاری از موقعیت‌ها می‌شود.

برای حل مشکلات ذکر شده، «کولش» و همکاران [۱۱] اولین سامانه انتساب داده بر پایه ذخیره چکیده‌هایی از ترافیک را پیشنهاد کردند. این سامانه به نام HBF با استفاده از توابع درهم‌سازی[۱۸] و فیلترهای بلوم [۱۲،۱۳] چکیده‌هایی از ترافیک می‌سازد. HBF محتوای هر بسته ترافیکی را به چند قطعه تقسیم و برای هر کدام چکیده‌ای با استفاده از توابع درهم‌سازی تولید می‌کند. چکیده‌های تولید شده بطور فشرده در ساختاری داده‌ای مشابه فیلتر بلوم ادغام می‌شوند.

عملیات چکیده‌سازی یک‌طرفه است زیرا دامنه خروجی توابع درهم‌سازی بسیار کوچک‌تر از دامنه ورودی آنها است. این بدان

معناست که چکیده ترافیک که به‌طور قابل‌توجهی کوچک‌تر از ترافیک اصلی است، نمی‌تواند به داده‌های اصلی بازگردانده شود و در نتیجه، محرمانگی حفظ می‌شود. با توجه به اینکه چکیده نمی‌تواند به ترافیک اصلی بازگردانده شود، از آن برای استعلام یک رشته داده استفاده می‌شود. چکیده می‌تواند به استعلام برای یک رشته پاسخ دهد و فهرستی از اتصال‌هایی که آن رشته را حمل کرده‌اند برگرداند. فرآیند استعلام مشابه با فرآیند چکیده‌سازی است: رشته را به قطعه‌هایی تقسیم کرده و از فیلتر بلوم برای آن قطعه‌ها استعلام می‌گیرند.

هزینه‌ای که بابت فشرده‌سازی چکیده‌ها پرداخت می‌شود پاسخ‌های مثبت کاذب است، بدین معنا که برخی از اتصال‌هایی که توسط تابع استعلام بازگردانده می‌شوند، رشته مورد تجسس را حمل نکرده‌اند. با این حال، تا زمانی که مرزهای یک قطعه از رشته مورد جستجو در دو بسته متوالی قرار نگیرد، منجر به منفی کاذب نمی‌شود. «پونک» و همکاران [۱۴،۱۵] روش‌های بهبود یافته انتساب داده WBS و WMH را پیشنهاد کردند که منجر به کاهش قابل‌توجهی در نرخ مثبت کاذب با نسبت کاهش داده تا ۱۰۰:۱ شدند. آن‌ها از الگوریتم بهبود یافته‌ای برای تقسیم رشته‌ها به قطعه‌ها استفاده می‌کنند. «حقیقت» و همکاران [۱۶] روش CMBF را پیشنهاد دادند که از جستجوهای نویسه عام[۱۹] پشتیبانی می‌کند. جستجوی نویسه عام به معنای جستجوی رشته داده‌ای است که برخی از نویسه‌های[۲۰] آن ناشناخته است، مانند امضای کرم‌های چندریختی[۲۱]. CBID [۱۷] یک سامانه انتساب داده دیگر است که بر اساس ترکیبی از فیلتر بلوم، جدول شاخص بیتی و روش نمونه‌برداری ترافیک طراحی شده است. این سامانه نرخ مثبت کاذب را در مقایسه با روش‌های قبل بطور قابل‌توجهی کاهش می‌دهد. DSPAS جدیدترین راهکار انتساب داده پیشنهاد شده است [۱۸] که با روش‌های پردازش سیگنال دیجیتال پیاده‌سازی شده است. سازوکار چکیده‌سازی DSPAS امکان یافتن رشته‌های داده مشابه با رشته مورد تجسس را فراهم می‌کند، که البته پیچیدگی محاسباتی آن به مراتب بیشتر از روش‌های قبل است. همه روش‌های ذکر شده ترافیک را در سطح بسته[۲۲] پردازش و چکیده‌سازی می‌کنند این بدین معنی است که زمانی که مرزهای یک رشته مورد جستجو در دو بسته متوالی قرار گیرد سامانه دچار پاسخ منفی کاذب می‌شود.

با وجود همه این پژوهش‌ها در زمینه انتساب داده، همان‌طور که در بخش اول ذکر شد، ترافیک رمزگذاری شده سامانه‌های انتساب داده را بی‌اثر می‌کند. به‌طور واضح، جستجوی داده آشکار در محتوای یک اتصال رمزگذاری شده نتیجه مفیدی ندارد، چه برسد به چکیده یک اتصال رمزگذاری شده.

۲-۲- چالش ترافیک رمز شده

پروتکل‌های رمزگذاری نقش حیاتی در رشد شبکه‌های کامپیوتری، به ویژه اینترنت و برنامه‌های کاربردی آن ایفا کرده‌اند. این پروتکل‌ها حریم خصوصی و محرمانگی کاربران را هنگام انتقال اطلاعات از طریق پیوند[۲۳] نا امن اینترنت فراهم می‌کنند. اما رمزگذاری یک شمشیر دو لبه است. این امر مانع از کارکرد موثر دستگاه‌های نظارتی و امنیتی سازمان‌ها، به ویژه آنهایی که بر پایه بازرسی عمیق بسته‌ها هستند می‌شود. برای حل این مشکل، سازمان‌ها با استفاده از روش بازرسی TLS، تعادلی[۲۴] بین حریم خصوصی افراد داخلی و امنیت سازمانی ایجاد می‌کنند.

یک جعبه میانی[۲۵] بازرسی TLS به سازمان‌ها امکان را می‌دهد که شکل آشکار ترافیک رمزگذاری شده را بررسی کنند و آن‌ها را برای تحلیل‌های جرم‌شناسی ذخیره و یا محتوای مخرب را پالایش کنند. ابزارها و محصولات مختلفی این روش را پیاده‌سازی کرده‌اند، مانند mitmproxy [۱۹]، SSLSplit [۲۰]، دستگاه‌های Symantec SSL Visibility [۵] و دیوار حفاظتی Sophos XG [۴]. بازرسی TLS در واقع یک روش مرد میانی[۲۶] است که توسط مرورگر[۲۷] کامپیوتری که درخواست اتصال امن دارد به عنوان یک حمله شناسایی نمی‌شود. برای بررسی اتصال رمزگذاری شده یک کاربر داخلی، جعبه میانی بازرسی TLS که به‌عنوان نایب (میانجی)[۲۸] در لبه شبکه سازمانی عمل می‌کند، ابتدا اتصال درخواست شده توسط کاربر داخلی را خاتمه داده[۲۹]، داده‌های آن را رمزگشایی می‌کند. پس از بررسی داده‌های آشکار یا ذخیره آن‌ها برای کاربردهای جرم‌شناسی، جعبه میانی آن را دوباره رمزگذاری و از طریق یک اتصال امن دیگر به کارساز مورد نظر کاربر ارسال می‌کند. این فرآیند در جهت معکوس نیز انجام می‌شود، یعنی برای داده‌هایی که از کارساز خارجی به کاربر داخلی ارسال می‌شود. لازم به ذکر است که یک گواهی‌نامه CA خاص بطور بر روی تمام کامپیوترهای سازمانی نصب می‌شود تا این بازرسی به عنوان حمله شناسایی نشود.

روش بازرسی TLS که به آن «تقسیم جلسه TLS[۳۰]» هم گفته می‌شود، دو جلسه TLS کاملا جداگانه (کارخواه به جعبه میانی و جعبه میانی به کارساز) ایجاد می‌کند، بنابراین ویژگی‌های PFS[۳۱] و AEAD[۳۲] در TLS تاثیری بر آن ندارند. به همین دلیل، این روش برای هر دو نسخه ۱,۲ و ۱,۳ پروتکل TLS کار می‌کند. روش‌های بازرسی TLS دیگری نیز وجود دارند که بعضی از آن‌ها با همه نسخه‌های TLS سازگار نیستند. به عنوان مثال، روش‌های غیرفعال

اتصال کارخواه را خاتمه نمی‌دهند و بنابراین تاخیری ایجاد نمی‌کنند. این روش‌ها با گواهی RSA کارساز ارائه می‌شوند تا بتوانند اتصال را به‌طور غیرفعال رمزگذاری کنند. اما این روش زمانی که مجموعه رمزهای[۳۳] PFS استفاده می‌شوند (برخی از مجموعه رمزهای نسخه ۱,۲ پروتکل TLS و همه مجموعه رمزهای نسخه ۱,۳ پروتکل TLS) کار نمی‌کند. برخی روش‌های دیگر، مانند [۲۱]، لایه‌های رمزگذاری و احراز هویت را در TLS جدا می‌کنند. این روش به دلیل اجباری بودن حفظ یکپارچگی داده مبتنی بر AEAD در نسخه ۱,۳ پروتکل TLS قابل استفاده نیست. خوانندگان علاقه‌مند می‌توانند به مرجع [۲] برای بررسی جامع روش‌های بازرسی TLS مراجعه کنند.

نکته مهمی که می‌خواهیم بر آن تمرکز کنیم این است که روش بازرسی TLS محرمانگی را نقض می‌کند. جعبه میانی اتصال‌های کاربران را رمزگشایی کرده و می‌تواند اطلاعات محرمانه آن‌ها را ببیند. برخی تحقیقات، مانند [۲۲]، پیشنهاد می‌کنند که کارخواه‌ها به طور انتخابی کلید TLS را به جعبه‌های میانی مجاز ارائه دهند. با این حال، این مشکل را حل نمی‌کند. در مدل تهدید ما، نهادی خارج از کامپیوتر کاربر، مثل جعبه میانی، قابل اعتماد در نظر گرفته نمی‌شود و بنابراین نمی‌خواهیم اطلاعات آشکار اتصال‌های رمزگذاری شده را به جعبه میانی ارائه دهیم.

شایان ذکر است که راهکارهای رمزنگاری مانند BlindBox [۲۳]، SPABox [۲۴]، PrivDPI [۲۵]، و Pine [۲۶] امکان بازرسی مستقیم ترافیک رمزگذاری شده را بدون نقض محرمانگی فراهم می‌کنند. این سامانه‌ها ترافیک را رمزگشایی نمی‌کنند. در عوض، قالب‌های رمزگذاری شده الگوهای ترافیکی مخرب، که به آن‌ها قوانین گفته می‌شود، در اختیار جعبه میانی گذاشته می‌شود. این قوانین می‌توانند شامل امضای بدافزار (برای کاربرد تشخیص نفوذ)، اطلاعات محرمانه (برای تشخیص نشت داده)، و محتوای ممنوعه (برای نظارت والدین) باشند. ترافیک هر اتصال رمزگذاری شده با قوانین رمزگذاری شده مقایسه می‌شود و در صورت تطابق، هشدار ایجاد می‌شود. راه‌حل آن‌ها مبتنی بر روش‌های «رمزنگاری قابل جستجو»[۳۴] است که البته مستعد آسیب‌پذیری در برابر حملات هستند [۲۷،۲۸]. علاوه بر این، این رویکرد تاخیر قابل توجهی در برقراری اتصال و ارسال ترافیک ایجاد می‌کنند و با افزایش تعداد قوانین، تاخیر نیز افزایش می‌یابد. مهم‌ترین جنبه منفی این رویکرد این است که تنها در صورتی کار می‌کند که هر دو طرف هر اتصال از پروتکل ویژه آنها استفاده کنند. بنابراین، این راهکار در حال حاضر عملی نیست، زیرا نمی‌توانیم همه کاربران و کارسازهای اینترنت را مجبور به استفاده از پروتکل جدید کنیم مگر اینکه به یک استاندارد جهانی تبدیل و جایگزین پروتکل TLS موجود شود. علاوه بر این، فرض اساسی این سامانه این است که جعبه میانی قوانین را می‌داند، که برای کاربردهای جرم‌شناسی و روش‌های انتساب داده که نقش خود را زمانی ایفا می‌کنند که قوانین امنیتی نتوانسته‌اند از وقوع حادثه جلوگیری کنند قابل اجرا نیست. بنابراین، ما به راه‌حلی نیاز داریم که ترافیک رمزگذاری شده را به صورت قابل استفاده برای تحلیل‌های جرم‌شناسی ضبط کند و در عین حال محرمانگی را نیز نقض نکند.

در نهایت، شایان ذکر است که چندین مطالعه نشان داده‌اند که جعبه‌های میانی و روش‌های بازرسی TLS باعث کاهش حفاظت ارائه شده توسط TLS می‌شوند [۲،۶،۷،۹]. بر اساس مطالعه‌ای معروف که توسط تیمی از دانشگاهیان و متخصصان انجام شده است [۶]، ۶۲ درصد از اتصالاتی که از طریق یک جعبه میانی بازرسی TLS عبور می‌کنند دارای امنیت کمتری هستند و ۵۸ درصد از اتصال‌های جعبه میانی دارای آسیب‌پذیری‌های بحرانی هستند.

۲-۳- تعریف مسئله و مدل تهدید

هدف ما ارائه راه‌حلی برای مسئله انتساب داده در جریان[۳۵]های TLS یک شبکه سازمانی بر اساس مدل تهدید زیر است. بیان مسئله به صورت کلی عبارت است از:

«اگر k بایت از جریان f (که TLS است) از طریق کارساز جرم‌شناسی S منتقل شود، باید بتوانیم کارساز جرم‌شناسی را برای هر زیررشته‌ای از $k-\tau$ بایت اول محموله f جستجو کنیم و سرآیند f را به دست آوریم.» واضح است که مقدار مطلوب برای τ صفر است، اما اگر هیچ راه‌حلی برای $\tau = 0$ یافت نشود، مقدار کوچک τ نیز قابل قبول است.

در مدل تهدید ما، کارساز جرم‌شناسی «صادق اما کنجکاو»[۳۶] است، یعنی وظایف مورد انتظار را به طور کامل انجام می‌دهد، اما ممکن است سعی کند اطلاعاتی از تاریخچه ترافیک ذخیره شده استخراج کند (مثلا بر اثر جاسوسی یا نشت اطلاعات). علاوه بر این، ممکن است تاریخچه ترافیک از سامانه ذخیره‌سازی نشت کند. یک مهاجم ممکن است تاریخچه ترافیک را از سامانه ذخیره‌سازی سرقت کند. با این حال، نرم‌افزار اصلی کارساز جرم‌شناسی در برابر هرگونه تغییرات و نفوذهای مخرب ایمن در نظر گرفته می‌شود. در مورد کامپیوترهای کاربران، ما نرم‌افزاری که بر روی آن‌ها نصب می‌کنیم را امن در نظر می‌گیریم، یعنی مهاجم نمی‌تواند به نرم‌افزار نفوذ و کلیدهای مخفی را استخراج کند. با این حال، یک مهاجم می‌تواند

نرم‌افزار را حذف یا ترافیک آن را مسدود کند تا مانع از بدست آوردن و ذخیره تاریخچه ترافیک توسط کارساز جرم‌شناسی شود. علاوه بر این، مهاجم ممکن است تلاش کند با ارائه تاریخچه ترافیک جعلی کارساز جرم‌شناسی را فریب دهد.

بر اساس مدل تهدید، یک راه حل مناسب باید ویژگی‌های زیر را داشته باشد:

صحت[۳۷]: تاریخچه ترافیک ذخیره شده باید از داده اصلی جریان‌های رمزگذاری شده نشأت گرفته باشد. به عبارت دیگر، راه حل باید از ذخیره چکیده‌های جعلی تولید شده توسط کامپیوترهای مخرب یا آلوده شده جلوگیری کند. علاوه بر این، چکیده‌ها در مسیر خود به سمت کارساز جرم‌شناسی ممکن است تغییر کنند. هرگونه تغییر غیر مجاز نیز باید شناسایی شود.

تمامیت[۳۸]: داده‌های همه جریان‌های TLS که از طریق کارساز جرم‌شناسی منتقل می‌شوند باید در تاریخچه ترافیک گنجانده شوند. به عبارت دیگر، یک جریان TLS نباید منتقل شود مگر اینکه چکیده محموله آن بدست آمده و ذخیره شود.

محرمانگی: راه‌حل نباید محرمانگی کاربران داخلی را نقض کند.

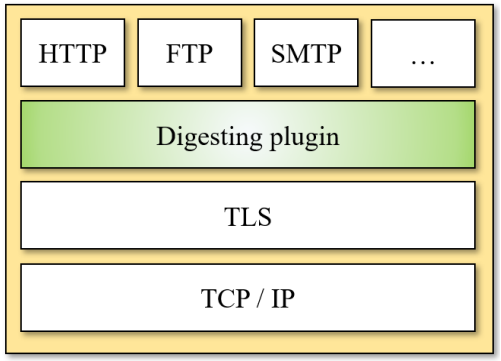
تاکید می‌کنیم که ما فقط ترافیک رمزگذاری شده استاندارد (TLS) و برنامه‌هایی که فقط از آن استفاده می‌کنند (مانند وب‌سایت‌های معمولی مبتنی بر HTTPS) را در نظر می‌گیریم و داده‌هایی که در لایه کاربرد با استفاده از روش‌های رمزگذاری خاص رمزگذاری شده‌اند را پوشش نمی‌دهیم. برای مثال، اگر کاربری یک سند را قبل از ارسال آن از طریق شبکه با استفاده از یک ابزار فشرده‌سازی مثل 7-zip فشرده و رمزگذاری کند، محتوای سند نمی‌تواند انتساب داده شود. در واقع، هیچکدام از روش‌های بازرسی TLS نیز نمی‌توانند اطلاعاتی را که در لایه کاربرد رمزگذاری شده است رمزگشایی کنند.

۳- راهکار پیشنهادی

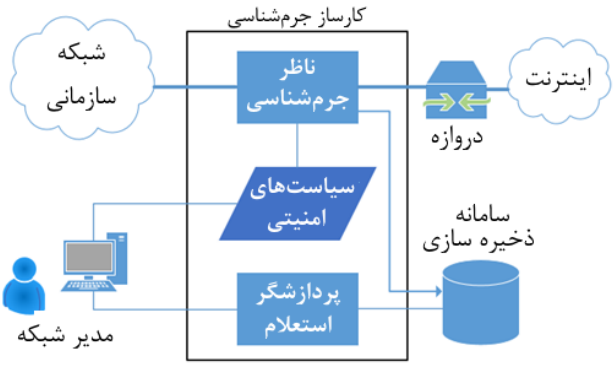
«جرم‌یاب» یک راهکار عملی برای مسئله مطرح شده است. ایده پشت جرم‌یاب بر این واقعیت بنا نهاده شده است که اگرچه ترافیک رمز شده در مسیر خود در شبکه نامفهوم و غیرقابل استفاده است، اما ترافیک در هر طرف اتصال، به صورت غیر رمزشده و آشکار است. بنابراین، اگر یک طرف اتصال، ترافیک ورودی و خروجی ساده خود را به یک کارساز جرم‌شناسی ارائه دهد، کارساز می‌تواند ترافیک را برای تجسس‌های پس از وقوع جرم ذخیره کند. مشکل واضح این است که کاربران نمی‌خواهند اطلاعات حساس خود را حتی برای یک طرف معتبر فاش کنند. اگر راهی وجود داشته باشد که ترافیک کاربران بدون فاش کردن اطلاعات درون آن به کارساز جرم‌شناسی ارسال شود، مشکل حل می‌شود. راه‌حل استفاده از روش‌های چکیده‌سازی ترافیک است.

در راهکار جرم‌یاب، کارساز جرم‌شناسی پشت دروازه[۳۹] شبکه سازمان قرار می‌گیرد و مسئول نظارت بر ترافیک ورودی/خروجی و همچنین ذخیره تاریخچه ترافیک است. هر کامپیوتر متصل به شبکه به یک افزونه نرم‌افزاری به نام «افزونه چکیده‌ساز» مجهز است که مطابق شکل ۲ در بالای لایه TLS قرار می‌گیرد. برنامه‌ها معمولا از پروتکل TLS برای ارسال و دریافت داده‌ها به صورت امن از طریق یک اتصال رمزگذاری شده استفاده می‌کنند. آن‌ها داده‌های آشکار را به لایه TLS می‌دهند و لایه TLS داده‌ها را رمزگذاری و با استفاده از پروتکل TCP ارسال می‌کند. در راه حل پیشنهادی ما، افزونه چکیده‌ساز هر دستگاه حالت آشکار داده‌ای را که یک برنامه به لایه TLS می‌دهد می‌گیرد و از آن‌ها چکیده تولید می‌کند. چکیده‌های تولید شده توسط افزونه به کارساز جرم‌شناسی ارسال می‌شوند تا در یک فیلتر بلوم ادغام و ذخیره شوند.

شکل ۳ اجزای کارساز جرم‌شناسی را نشان می‌دهد. «ناظر جرم‌شناسی» مسئول دریافت چکیده‌ها و انتقال آن‌ها به سامانه ذخیره‌سازی است. اگر چکیده یک اتصال رمزگذاری شده دریافت نشود، ناظر جرم‌شناسی اتصال را مسدود می‌کند. واحد پردازش استعلام در زمان وارسی و تجسس استفاده می‌شود. هنگامی که تحلیل‌گر جرم‌شناسی یک رشته داده را استعلام می‌کند، واحد «پردازشگر استعلام» چکیده‌های ذخیره شده را پردازش و جریان‌هایی که رشته را حمل کرده‌اند گزارش می‌کند. برای برآورده کردن ویژگی‌های مطرح شده در بخش ۳-۲، شبکه سازمانی باید یک سری سیاست‌های امنیتی اتخاذ کند که در ادامه، مورد بحث قرار می‌گیرد. مدیر شبکه سیاست‌ها را در کارساز جرم‌شناسی تنظیم می‌کند تا توسط ناظر جرم‌شناسی و افزونه چکیده‌ساز هر کامپیوتر سازمان قابل دسترس باشد.



شکل ۲. موقعیت افزونه چکیده‌ساز در پشته شبکه



شکل ۳. اجزای کارساز جرم‌شناسی

تا بدینجا، یک سامانه انتساب داده در سطح سازمانی داریم که بر روی ترافیک رمزگذاری شده استاندارد کار می‌کند و همچنین ویژگی سوم، یعنی حفظ محرمانگی، را برآورده می‌کند. با این حال، دو ویژگی دیگر، یعنی صحت و تمامیت نیز باید برآورده شوند. در ادامه، به چگونگی تامین این دو ویژگی می‌پردازیم.

۳-۱- صحت

برای دستیابی به ویژگی صحت، باید از پذیرش چکیده‌های جعلی توسط ناظر جرم‌شناسی جلوگیری کنیم. یک چکیده جعلی می‌تواند توسط یک افزونه چکیده‌ساز جعلی تولید یا توسط یک مهاجم بازپخش شود. علاوه بر این، چکیده‌ها در مسیر خود به سمت کارساز جرم‌شناسی ممکن است دچار تغییرات مخرب شوند. این مشکل را می‌توان به‌سادگی با استفاده از علامت احراز هویت پیام درهم‌سازی شده[۴۰] یا HMAC حل کرد. علامت احراز هویت یکپارچگی و اصالت داده را تایید می‌کند. افزونه چکیده‌ساز یک شماره توالی و یک شناسه اتصال را به هر چکیده اضافه و سپس علامت احراز هویت ترکیب حاصل را با استفاده از کلید مخفی آن کامپیوتر محاسبه می‌کند و در نهایت چکیده و علامت احراز هویت را برای کارساز جرم‌شناسی ارسال می‌کند. کلید مخفی خاص هر کاربر یا کامپیوتر در افزونه چکیده‌ساز ثابت‌گذاری[۴۱] می‌شود و باید محافظت شود (با روش‌هایی مانند مبهم‌سازی[۴۲] [۲۹] که خارج از محدوده این مقاله است). بر اساس مدل تهدید مطرح شده، مهاجمان نمی‌توانند به افزونه چکیده‌ساز نفوذ و کلیدهای مخفی را سرقت کنند. با این حال، آن‌ها می‌توانند سعی در ارسال چکیده‌های جعلی داشته باشند. کارساز جرم‌شناسی می‌تواند با استفاده از علامت احراز هویت یکپارچگی و اصالت چکیده‌های دریافت شده را تایید کرده و تغییرات غیرمجاز یا تولید چکیده‌های جعلی را شناسایی کند. به عبارت دیگر، کارساز جرم‌شناسی یک چکیده را می‌پذیرد اگر بتواند یکپارچگی و اصالت آن را با استفاده از علامت احراز هویت پیام تایید کند. لازم به ذکر است که به دلیل استفاده از شماره توالی، از دست رفتن یا تکرار چکیده‌ها قابل شناسایی است. علاوه بر این، اگر چکیده یک اتصال برای اتصال دیگر توسط یک مهاجم بازپخش شود، شناسه اتصال مانع از پذیرش آن می‌شود.

۳-۲- تمامیت

ویژگی تمامیت به این معنی است که چکیده هر جریانی که از طریق کارساز جرم‌شناسی منتقل می‌شود باید ذخیره شود. این هدف می‌تواند با اتخاذ برخی سیاست‌های امنیتی محقق شود. اعمال سیاست‌های امنیتی یک وظیفه اساسی و حیاتی برای شبکه‌های سازمانی است [۳۰،۳۱]، که ابزارهای پیچیده‌ای برای آن پیشنهاد شده است [۳۲،۳۳].

مدیر شبکه قوانین و سیاست‌های امنیتی را تنظیم و برقرار می‌کند. این قوانین از واحد سیاست امنیتی دریافت شده و باید توسط هر افزونه چکیده‌ساز رعایت شوند. در غیر این صورت، اتصال رمزگذاری شده آن‌ها توسط ناظر جرم‌شناسی مسدود می‌شود.

یک قانون ساده می‌تواند این باشد:

قانون ۱: «هر بسته ترافیکی یک اتصال رمزگذاری شده، باید چکیده داده خود را نیز به همراه داشته باشد. در غیر این صورت، کارساز جرم‌شناسی مانع از عبور بسته می‌شود.»

این قانون دو مشکل دارد. مشکل اول این است که افزونه چکیده‌ساز در بالای لایه انتقال قرار دارد و نمی‌تواند چکیده‌ای برای بار داده هر بسته ترافیکی که در لایه شبکه ساخته می‌شود، تولید کند. به عبارت دیگر، مرزهای بخش داده بسته‌های شبکه در لایه‌های بالای لایه شبکه قابل تعیین نیستند. علاوه بر این، همگام‌سازی چکیده‌ها و بسته‌ها دشوار و تقریبا غیر ممکن است. بنابراین، قانون باید بر داده هر اتصال به جای هر بسته اعمال شود. مشکل دوم این است که چکیده یک بسته ورودی نمی‌تواند تا زمانی که کاربر داخلی مربوطه آن را دریافت نکند، بدست آید. با توجه به این دو مسئله، ما از قانون تعدیل شده زیر استفاده می‌کنیم:

قانون ۲: «یک جریان رمزگذاری شده تا زمانی که بخشی از بار داده جریان که چکیده آن دریافت نشده است از یک آستانه تعریف‌شده، به نام آستانه قطع ارسال، تجاوز نکند، اجازه عبور از ناظر جرم‌شناسی را دارد.»

این بدین معنی است که افزونه چکیده‌ساز می‌تواند چکیده‌های چندین بسته را ذخیره کند تا زمانی که اندازه کل بار داده آن‌ها به نزدیکی آستانه قطع ارسال برسد و سپس چکیده‌ها را با یک تراکنش

ارسال کند. از طرف دیگر، تا زمانی که چکیده‌ها به طور منظم دریافت شوند و اختلاف بین اندازه کل بار داده‌های ارسال شده و اندازه کل بار داده‌هایی که چکیده‌های آن‌ها دریافت شده است از آستانه تجاوز نکند ناظر جرم‌شناسی بسته‌ها را بدون هیچ تاخیری انتقال می‌دهد. ما از نماد τ برای نشان دادن آستانه قطع ارسال استفاده می‌کنیم.

بنابراین، افزونه چکیده‌ساز در سطح جریان کار می‌کند نه سطح بسته. برای اجرای این قانون، ناظر جرم‌شناسی باید دو متغیر برای هر اتصال فعال نگه دارد، d_c و u_c. متغیر d_c اندازه (تعداد بایت‌ها) بخشی از بار داده اتصال که چکیده‌های آن دریافت شده را ذخیره می‌کند و متغیر u_c اندازه بخشی از بار داده که از طریق ناظر جرم‌شناسی منتقل شده است را ذخیره می‌کند. هنگامی که چکیدهδ برای اتصالc دریافت می‌شود، کارساز جرم‌شناسیd_c را بروزرسانی می‌کند. از طرف دیگر، هنگامی که کارساز جرم‌شناسی یک بسته را منتقل می‌کند، u_c، به اندازه بار داده بسته افزایش می‌یابد. بنابراین، کارساز جرم‌شناسی می‌تواند با استفاده از این دو متغیر از انتقال جریانی که چکیده آن دریافت نشده است جلوگیری کند. به‌طور خاص، برای هر بسته ترافیکی p که می‌خواهد از لبه شبکه از طریق کارساز جرم‌شناسی عبور کند، بسته تنها در صورتی ارسال می‌شود که:

$$u_c + PayloadLength(p) - d_c \leq \tau \qquad (۱)$$

که در رابطه (۱) $PayloadLength(p)$ اندازه بار داده بسته p را نشان می‌دهد. لازم به ذکر است که وقتی شرط معادله (۱) برای بسته p برقرار نباشد، ناظر جرم‌شناسی بسته را میانگیر[۴۳] می‌کند و منتظر دریافت چکیده جریان آن می‌ماند. اگر چکیده پس از زمان از پیش تعریف شده‌ای دریافت نشود، اتصال جریان مسدود و گزارش می‌شود. شبه‌کدهای افزونه چکیده‌ساز و ناظر جرم‌شناسی در الگوریتم ۱ و الگوریتم ۲ نشان داده شده‌اند.

بنابراین، راه‌حل پیشنهادی دارای سطح قابل قبولی از ویژگی تمامیت است. به عبارت دیگر، تنها τ بایت از یک جریان ترافیک رمز شده می‌تواند بدون ارائه چکیده آن منتقل شود. امکان ارسال اطلاعات مهم بدون ارائه چکیده می‌تواند با تنظیم مقدار کوچک برای آستانه قطع ارسال، مانند ۴ کیلوبایت، به‌طور قابل توجهی کاهش یابد. اگر یک کاربر داخلی چکیده ترافیک رمزگذاری شده خود را پس از τ بایت ارسال نکند، ناظر جرم‌شناسی اتصال را متوقف کرده و فعالیت مشکوک را گزارش می‌کند.

یک سیاست تکمیلی نیز می‌تواند برای اتصال‌های داخلی یک سازمان تنظیم شود تا چکیده‌های ترافیک داخلی نیز جمع‌آوری شود. در این صورت، افزونه چکیده‌ساز باید چکیده هر اتصال با هر فرد در داخل شبکه سازمانی را نیز به کارساز جرم‌شناسی ارسال کنند. با این حال، نقض این سیاست توسط ناظر جرم‌شناسی قابل تشخیص نیست زیرا اتصال‌های داخلی از طریق کارساز جرم‌شناسی عبور نمی‌کنند. راه حل ممکن برای این مشکل استفاده از عوامل جرم‌شناسی توزیع‌شده است، یعنی استقرار یک ناظر جرم‌شناسی در هر مجموعه شبکه سازمانی. راه‌حل‌های مبتنی بر SDN و NFV برای تسهیل در پیاده‌سازی و اعمال چنین سیاست‌های امنیتی به‌صورت توزیع شده وجود دارد [۳۲،۳۳]. اگر یک شبکه سازمانی از راه‌گزین‌های[۴۴] SDN استفاده کند، ناظر جرم‌شناسی می‌تواند به‌عنوان یک برنامه نظارتی برای کار در سراسر شبکه پیاده‌سازی شود. OpenSec [۳۴]، که یک سکوی امنیتی مبتنی بر سیاست برای SDN است، گزینه مناسبی برای پیاده‌سازی این راه‌حل و سیاست‌های آن است.

افزونه چکیده‌ساز: یک مسئله مهم در راه حل پیشنهادی، پیاده‌سازی و استقرار افزونه چکیده‌ساز است. ما یک افزونه برای مرورگر فایرفاکس طراحی و پیاده‌سازی کردیم که تمام داده‌های آشکار ارسال یا دریافت شده از طریق هر اتصال TLS مرورگر را ضبط می‌کند. این افزونه را با استفاده از جاوااسکریپت و توابع webRequest موزیلا [۳۵] پیاده‌سازی کردیم. رویداد onBeforeRequest از توابع webRequest می‌تواند حالت آشکار همه درخواست‌ها و پاسخ‌های لایه کاربرد را در اختیار ما بگذارد. افزونه داده‌ها را با استفاده از الگوریتم CBID [۱۷] پردازش و به ناظر جرم‌شناسی ارسال می‌کند. بنابراین، همه اتصال‌های رمزگذاری شده استاندارد که با استفاده از مرورگر برقرار می‌شوند، مانند HTTPS، FTPS و SSMTP، به‌راحتی پردازش و در کارساز جرم‌شناسی ذخیره می‌شوند.

یک مزیت مهم دیگر ضبط درخواست‌ها و پاسخ‌ها در بالای پیمانه TLS این است که داده‌ای که قرار است پردازش شود، تحت تاثیر پروتکل‌های لایه پایین‌تر قرار نمی‌گیرد. به عبارت دیگر، ما اثرات قطعه‌های TCP و بسته‌های IP را نمی‌بینیم و افزونه می‌تواند داده خام لایه کاربرد را به صورت کامل ضبط کند. سامانه‌های انتساب داده موجود که در لبه شبکه کار می‌کنند، ترافیک را در سطح بسته پردازش می‌کنند. آن‌ها می‌توانند بار داده هر بسته را به طور جداگانه پردازش کنند یا بار داده بسته‌های هر جریان را ترکیب کرده و بار داده جریان حاصل را به‌صورت کامل پردازش کنند. مورد اول می‌تواند در زمان تجسس به نتایج منفی کاذب منجر شود، زیرا

الگوریتم ۱- شبه‌کد افزونه چکیده‌ساز

DigestingPlugIn_DataTransferHandler
Input: *plainData*: Data from either application or TLS layer
cID: Connection identifier (5-tuple)

$\delta[cID].digest \leftarrow \delta[cID].digest \parallel$ **MakeDigest**(*plainData*)
$\delta[cID].dataLen \leftarrow \delta[cID].dataLen +$ **Length**(*plainData*)
if ($\delta[cID].dataLen \geq \tau - T$) **then**
    digestMsg.digest $\leftarrow \delta[cID]$
    digestMsg.hMAC $\leftarrow$ **CalculateHMAC**($\delta[cID]$, *hKey*)
    Send(*digestMsg*, *fsIP*)
    $\delta[cID].digest \leftarrow$ NULL
    $\delta[cID].dataLen \leftarrow 0$
    $\delta[cID].seq \leftarrow \delta[cID].seq + 1$
end if

Notation:
$\parallel$: Append operator
δ : Map of digest objects for different connections.
τ : Forwarding block threshold
hKey : HMAC secret key
T : Digest pass threshold
seq : Sequence number for digest messages
fsIP : IP address of the forensic server

مرزهای یک قطعه از رشته جستجو شده ممکن است در دو بسته متوالی قرار بگیرد. مورد دوم منجر به هزینه پردازش و نیاز به حافظه بسیار زیاد می‌شود زیرا سامانه انتساب بار داده باید بسیاری از اتصال‌ها را دنبال و بار داده بسته‌های هر جریان را ترکیب کند. بنابراین، سامانه‌های انتساب بار داده موجود، رویکرد اول را انتخاب می‌کنند [۱۸-۱۵] که گاهی منجر به نتایج منفی کاذب می‌شود.

افزونه چکیده‌ساز می‌تواند به‌راحتی برای مرورگرهای متن‌باز دیگر نیز توسعه یابد. علاوه بر این، چکیده‌ساز می‌تواند به صورت‌های دیگر، مانند کتابخانه‌های برنامه‌نویسی توسعه یابد تا توسط برنامه‌های دیگر مورد استفاده قرار گیرد. به‌عنوان مثال، فرض کنید یک شرکت امنیتی برنامه خاص خود را برای ارتباط بین شعبه‌های خود دارد. توسعه‌دهندگان برنامه می‌توانند به سادگی این کتابخانه را به برنامه اضافه و آن را به افزونه چکیده‌ساز مجهز کنند.

۴- ارزیابی

ما ارزیابی‌های خود را در دو مرحله انجام دادیم. در مرحله اول، توانایی جرم‌یاب در شناسایی و انتساب رشته‌های جستجو شده را ارزیابی کردیم. در مرحله دوم، سایر ویژگی‌های کارایی آن مانند سربار ترافیکی، تاخیر افزوده شده به ترافیک، و بار پردازش پردازنده کاربران داخلی را مورد ارزیابی قرار دادیم.

۴-۱- ارزیابی قابلیت انتساب

ما افزونه چکیده‌ساز را برای مرورگر فایرفاکس توسعه دادیم و آن را

الگوریتم ۲- شبه‌کد ناظر جرم‌شناسی

ForensicAgent_DigestReceiveHandler
Input: *digestMsg*: Digest message from a client machine

$\delta \leftarrow digestMsg.digest$
if (**CalculateHMAC**(δ, *hKey*) = *digestMsg.hMAC*) **then**
    $cID \leftarrow digestMsg.connectionID$
    if (*digestMsg.seq* = $seq[cID] + 1$) **then**
        StoreInBloomFilter($\delta.digest$, *cID*)
        $d_c[cID] \leftarrow d_c[cID] +$ *digestMsg.dataLen*
        $seq[cID] \leftarrow seq[cID] + 1$
        if ($pktBuffer[cID] \neq$ EMPTY) **then**
            do
                $pkt \leftarrow$ **Dequeue**(pkt*Buffer*[*cID*])
                Forward(*pkt*)
                $u_c[cID] \leftarrow u_c[cID] +$ **PayloadLength**(*pkt*)
            while($u_c[cID]$+**PayloadLength**(*pkt*)-$d_c[cID] \leq \tau$)
        end if
    end if
end if

Packet_Receive_Handler
Input: *pkt*: Packet from either inside or outside of network
cID: Connection identifier of the packet

if ($u_c[cID]$ + **PayloadLength**(*pkt*) - $d_c[cID] \leq \tau$) **then**
    Forward(*pkt*)
    $u_c[cID] \leftarrow u_c[cID] +$ **PayloadLength**(*pkt*)
else
    Enqueue(*pkt*, *pktBuffer*[*cID*])
end if

بر روی مرورگر فایرفاکس سه کامپیوتر آزمایشگاه تحقیقاتی خود نصب کردیم. الگوریتم چکیده‌سازی با نسبت کاهش داده ۱:۱۰۰ تنظیم شد. برنامه ناظر جرم‌شناسی را نیز توسعه دادیم. برای پردازش سریع بسته‌ها، نیاز داشتیم که هسته سیستم‌عامل را دور بزنیم و بسته‌ها را بدون وقفه‌های سیستم‌عامل پردازش کنیم. بنابراین، از DPDK [۱۱] و حالت سرکشی[۴۵] آن استفاده کردیم تا ناظر جرم‌شناسی را پیاده‌سازی کنیم. یک کامپیوتر مجزا این برنامه را میزبانی و نقش کارساز جرم‌شناسی را ایفا می‌کرد که از طریق آن سه کامپیوتر دیگر به اینترنت متصل شدند. به کاربران سه رایانه اجازه دادیم تا کارهای روزمره خود را با استفاده از اینترنت انجام دهند. در این مدت، کارساز جرم‌شناسی در مجموع چکیده‌ای به اندازه ۶۰ مگابایت از کل ترافیک را جمع‌آوری کرد. علاوه بر کارهای روزمره کاربران، سه سناریوی زیر را در طول روز انجام دادیم:

۱. کاربر شماره ۱ به وب‌سایت https://www.mehrnews.com مراجعه کرد و یکی از مقالات آن را باز کرد.

۲. کاربر شماره ۲ از طریق دو کارساز ایمیل مختلف دو نامه الکترونیک ارسال کرد. متن هر نامه حدود ۲۵۰۰ شناسه بود. بدین منظور از دو کارساز ایمیل زیر استفاده کردیم:

- https://pmail.sbu.ac.ir

- https://mail.google.com

۳. کاربر شماره ۳ به وب‌سایت‌های https://web.telegram.org و https://web.whatsapp.com وارد شد که برنامه‌های پیام‌رسان تلگرام و واتساپ مبتنی بر وب هستند و چند پیام ارسال کرد.

پس از جمع‌آوری چکیده‌ها، هر یک از سناریوها را به صورت زیر بررسی کردیم.

بررسی سناریو ۱: هدف این تجسس آزمایشی، شناسایی کاربری است که مقاله را دریافت کرده است. بدین منظور، از چکیده ترافیک برای کل متن مقاله استعلام گرفتیم، اما نتیجه مثبتی به دست نیامد. دلیل بدست نیامدن نتیجه این است که مقاله به صورت یک صفحه HTML دانلود شده بود و در نتیجه، فراداده‌ها و به ویژه برچسب‌های HTML بین پاراگراف‌های مقاله وجود داشت. این مسئله مانع از تطابق کامل بین رشته جستجو شده و رشته دریافت شده توسط کاربر می‌شد. بنابراین، برای یکی از پاراگراف‌های مقاله که حدود ۲۵۰ کاراکتر داشت استعلام کردیم. دو نتیجه مثبت به دست آمد: کاربران شماره ۱ و ۳. سپس، برای پاراگراف دیگری که حدود ۳۰۰ کاراکتر داشت استعلام گرفتیم و کاربران شماره ۱ و ۲ به عنوان نتایج مثبت گزارش شدند. از آنجا که سامانه‌های انتساب داده پاسخ منفی کاذب ندارند [۱۷-۱۱]، می‌توانیم نتایج مثبت کاذب را با اشتراک گرفتن از نتایج حذف کنیم. اشتراک مجموعه نتایج اول و دوم، کاربر شماره ۱ را به عنوان نتیجه نهایی مشخص می‌کند. باید توجه داشت که نرخ مثبت کاذب سامانه‌های انتساب داده برای رشته‌های کوچک جستجو شده بالا است. بر اساس نتایج مقالات قبلی، برای نسبت کاهش داده ۱:۱۰۰، زمانی که چکیده‌ها برای رشته‌هایی بزرگ‌تر از ۴۰۰ بایت جستجو می‌شوند، نرخ مثبت کاذب نزدیک به صفر است. ما برای پاراگراف دیگری از مقاله که حدود ۵۰۰ کاراکتر داشت جستجو کردیم و تنها پاسخ مثبت درست (کلاینت ۱) بدست آمد.

بررسی سناریو ۲: هدف از این تجسس، شناسایی کاربر ارسال کننده نامه الکترونیک است. ابتدا از چکیده برای یک پاراگراف از نامه ارسال شده توسط pmail.sbu.ac.ir استعلام گرفتیم، اما نتیجه مثبتی بدست نیامد. برای پیدا کردن مشکل، درخواست‌ها و پاسخ‌های کارساز ایمیل را در بخش ناظر شبکه فایرفاکس در حین ارسال ایمیل بررسی کردیم. مشاهده کردیم که نوع محتوا application/x-www-form-urlencoded است. در این فرمت، کاراکتر فاصله با علامت + جایگزین می‌شود. همچنین نشان‌های غیرالفبایی با نشانه %HH، که شامل یک نویسه درصد و دو رقم مبنای ۱۶ که نمایانگر کد ASCII کاراکتر است جایگزین می‌شوند. این موضوع در بخش محتوا نیز قابل مشاهده است. بنابراین، کاراکترهای فاصله پاراگراف مورد نظر را با علامت + جایگزین کرده و از چکیده‌ها دوباره برای پاراگراف اصلاح شده استعلام گرفتیم. کاربر شماره ۲ به درستی و بدون نتیجه مثبت کاذب شناسایی و گزارش شد.

سپس، چکیده‌ها را برای یک پاراگراف از نامه ارسال شده توسط کارساز گوگل استعلام کردیم. پس از عدم دریافت پاسخ مثبت، درخواست‌ها به کارساز گوگل را بررسی کردیم. مشاهده کردیم که نوع محتوا application/json است و متن ایمیل به‌عنوان یک سند HTML در یک شیء از ساختار JSON ارسال می‌شود. از آنجا که متن ایمیل به عنوان یک سند HTML در نظر گرفته می‌شود، برچسب‌های HTML در متن ایمیل وجود دارند، به ویژه بین پاراگراف‌ها. علاوه بر این، کاراکترهای فاصله با نشانه   جایگزین می‌شوند که شناسه فاصله غیر قابل شکست در HTML است. بنابراین، کاراکترهای فاصله پاراگراف را با   جایگزین و چکیده‌ها را دوباره برای پاراگراف اصلاح شده استعلام کردیم. کاربر شماره ۲ به درستی و بدون هیچ نتیجه مثبت کاذبی گزارش شد.

این سناریوها نشان می‌دهد که گاهی نیاز است که برای انتساب یک رشته داده جستجو شده، پیش‌پردازشی انجام شود و بازرسان امنیتی باید نقش خود را در مواجهه با وضعیت و انجام یک بررسی جامع به درستی و با دقت ایفا کنند. باید توجه داشت که این روال حتی برای ابتدایی‌ترین سامانه‌های انتساب داده، یعنی ابزارهای ضبط ترافیک خام، اجتناب‌ناپذیر است. یک راه‌حل برای ساده‌تر کردن این روال این است که افزونه با انجام پیش‌پردازش‌هایی فراداده‌های شناخته شده را قبل از اعمال توابع درهم‌سازی از داده‌ها حذف کند. به عنوان یک راه‌حل دیگر، روش چکیده‌سازی افزونه که مبتنی بر الگوریتم CBID است را می‌توان با روش چکیده‌سازی DSPAS که به تفاوت‌های کوچک حساس نیست، جایگزین کرد. همان‌طور که قبلا گفته شد، DSPAS از روش‌های پردازش سیگنال دیجیتال برای چکیده‌سازی یک رشته داده استفاده می‌کند. هنگامی که چکیده‌ها را برای یک رشته با استفاده از DSPAS استعلام کنیم، کاربرانی که رشته‌های داده مشابه با رشته استعلام شده را منتقل کرده‌اند هم به عنوان پاسخ گزارش می‌شوند.

بررسی سناریو ۳: هدف در این سناریو، شناسایی کاربری است که پیام را ارسال کرده است. از چکیده برای یکی از پیام‌ها استعلام گرفتیم، اما نتیجه مثبتی بدست نیامد. درخواست‌ها به کارسازهای

تلگرام و واتساپ را بررسی کردیم. مشاهده کردیم که نوع محتوا application/octet-stream است و داده‌های درخواست‌ها نامشخص است. تلگرام از پروتکل خاصی به نام MTProto برای انتقال پیام‌ها استفاده می‌کند. یکی از وظایف اصلی این پروتکل رمزگذاری پیام‌ها قبل از ارسال از طریق پروتکل لایه انتقال است. به عبارت دیگر، تلگرام پیام‌ها را در لایه کاربرد رمزگذاری می‌کند. از سوی دیگر، واتساپ از یک سازوکار رمزگذاری انتها به انتها[۴۶] استفاده می‌کند که در لایه کاربرد عمل می‌کند. بنابراین، در این مورد، چکیده‌هایی که افزونه از داده‌های رمزگذاری شده تولید می‌کند بی‌فایده هستند. همان‌طور که در بخش مقدمه بیان شد، روش پیشنهادی فقط زمانی موثر است که داده‌ها تنها با استفاده از پروتکل امنیت لایه انتقال رمزگذاری شوند.

۴-۲- ارزیابی خصوصیات دیگر کارایی

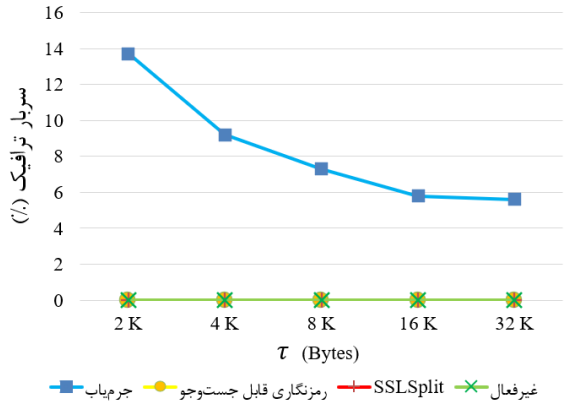
برای ارزیابی دیگر ویژگی‌های کارایی جرم‌یاب، به یک شبکه سازمانی مجهز به افزونه چکیده‌ساز و ناظر جرم‌شناسی نیاز داشتیم. کاربران داخلی شبکه باید اتصال‌های TLS به خارج از شبکه برقرار کنند و مدیر شبکه باید قادر به تنظیم سیاست‌های امنیتی روی ناظر جرم‌شناسی باشد. در این مدت، ویژگی‌های مختلفی مانند سربار ترافیک، تاخیر افزوده به ترافیک، و سربار پردازش پردازنده کاربران داخلی باید اندازه‌گیری شود.

برای انجام این ارزیابی، ما یک شبکه سازمانی را با استفاده از سکوی Docker [۳۶،۳۷] شبیه‌سازی کردیم. Docker یک سکوی مجازی‌سازی است که برای شبیه‌سازی و ارزیابی سامانه‌ها و شبکه‌های توزیع شده استفاده می‌شود [۳۸،۳۹]. شبکه شبیه‌سازی شده دارای ۵۰۰ کامپیوتر به عنوان کاربران داخلی است. هر کامپیوتر یک سامانه لینوکسی است. برای شبیه‌سازی این تعداد زیاد کامپیوترها، از توزیع Alpine که یک لینوکس سبک و قابل تنظیم است استفاده کردیم. همچنین افزونه چکیده‌ساز را روی مرورگر فایرفاکس هر رایانه نصب کردیم. در شبکه شبیه‌سازی شده، یک رایانه لینوکسی ویژه را به عنوان دروازه شبکه تنظیم کرده و کل شبکه را از طریق دروازه به اینترنت متصل کردیم. ناظر جرم‌شناسی نیز در کامپیوتر دروازه پیاده‌سازی شده است.

فهرستی از وظایف برای هر رایانه آماده کردیم. این وظایف شامل بارگیری صفحات وب HTTPS تصادفی از لیست ۱۰۰ وب‌سایت پربازدید و همچنین بارگیری و بارگذاری اسناد به یک کارساز FTPS است. فهرست وظایف هر رایانه شامل ۱۰۰ وظیفه است که در یک بازه زمانی یک ساعته پخش شده‌اند. همچنین برنامه‌ای برای کامپیوترها نوشتیم که وظایف تعریف شده را اجرا کند. این برنامه ابتدا حافظه نهان[۴۷] مرورگر رایانه‌ها را پاک می‌کند، سپس وظایف آنها را دریافت و در زمان مشخص شده اجرا می‌کند.

در آزمایش اول، افزونه چکیده‌ساز و ناظر جرم‌شناسی را غیرفعال کرده و سپس وظایف را انجام دادیم. در طول شبیه‌سازی یک ساعته، رایانه‌ها در مجموع ۴۷۴۲ مگابایت ترافیک بارگیری کردند. از سوی دیگر، ترافیک بارگذاری شده در مجموع ۳۹۵۷ مگابایت بود. پس از آن، افزونه‌های چکیده‌ساز و ناظر جرم‌شناسی را که برای نسبت کاهش داده ۱۰۰:۱ تنظیم شده بودند، فعال و آزمایش را برای ارزیابی خصوصیات کارایی تکرار کردیم. ما همچنین نرم‌افزار بازرسی TLS، یعنی SSLsplit [۲۰] و روش رمزگذاری غیرفعال [۲۲] را برای مقایسه با روش خود مورد ارزیابی عملی قرار دادیم. علاوه بر این، کارایی روش‌های رمزگذاری قابل جستجو را هر زمان که لازم بود بیان و با روش پیشنهادی جرم‌یاب مقایسه کرده‌ایم. ما روش‌های رمزگذاری قابل جست‌وجو را پیاده‌سازی نکردیم زیرا آنها نیاز دارند که دو طرف اتصال رمزگذاری شده از پروتکل رمزگذاری پیشنهادی خودشان استفاده کنند و بنابراین در آزمایش ما نمی‌توانند کار کنند. علاوه بر این، آنها برای کاربرد دیگری، یعنی تطبیق امضا برای تشخیص نفوذ، طراحی شده‌اند.

سربار ترافیک: شکل ۴ سربار ترافیک ناشی از تولید و ارسال چکیده‌ها را نشان می‌دهد. سربار ترافیک را بدین صوتر تعریف می‌کنیم: اندازه کل ترافیک چکیده‌های ارسال شده به ناظر جرم‌شناسی تقسیم بر حجم کل ترافیک ورودی و خروجی اصلی کامپیوترها (بدون در نظر گرفتن ترافیک چکیده‌ها) است. اولین سوالی که ممکن است مطرح شود این است که «با توجه به نسبت کاهش داده ۱۰۰:۱، چرا سربار ترافیک بیش از یک درصد از کل ترافیک است؟» دلیل آن به سازوکار چکیده‌سازی مربوط می‌شود. الگوریتم چکیده‌سازی ترافیک را با استفاده از فیلترهای بلوم چکیده

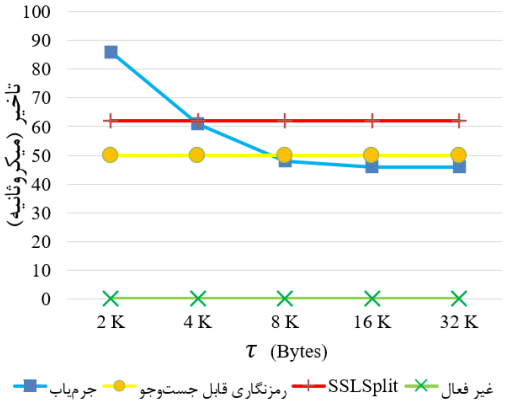


شکل ۴. سربار ترافیکی چکیده‌ها

می‌کند. به‌طور خلاصه، یک فیلتر بلوم از یک آرایه بیتی تشکیل شده است که مقدار همه آنها در ابتدا صفر است. در طول فرآیند چکیده‌سازی، بسته‌های ترافیک به قطعاتی تقسیم و قطعه‌ها درهم‌سازی می‌شوند. بیت‌های فیلتر بلوم متناظر با نتایج درهم‌سازی به یک مقداردهی می‌شوند. هنگامی که نتایج درهم‌سازی در یک فیلتر بلوم جمع‌آوری می‌شوند، اندازه نهایی چکیده کوچک‌تر از کل درهم‌سازی‌ها می‌شود. بنابراین، سربار ترافیک چکیده‌ها بیشتر از چکیده نهایی است. برای کاهش سربار ترافیک، افزونه چکیده‌ساز را به گونه‌ای طراحی کردیم که مقادیر درهم‌سازی که در مقادیر ارسالی قبل بوده‌اند را ارسال نکند. با جلوگیری از ارسال مقادیر درهم‌سازی یکسان توسط افزونه چکیده‌ساز، یعنی درهم‌سازی‌هایی که فیلتر بلوم ناظر جرم‌شناسی را تغییر نمی‌دهند، سربار ترافیک کاهش پیدا می‌کند. برای پیاده‌سازی این سازوکار، افزونه چکیده‌ساز مقادیر درهم‌سازی که قبلا ارسال شده‌اند را در یک فیلتر بلوم داخلی ذخیره و درهم‌سازی‌های جدید را قبل از ارسال به ناظر جرم‌شناسی با آن بررسی می‌کند. علاوه بر این، لازم به ذکر است که بخش قابل‌توجهی از سربار ترافیک به دلیل سربار سطح شبکه در ارسال پیام‌های درهم‌سازی است، یعنی سربار سرآیندهای IP و TCP.

همان‌طور که شکل ۴ نشان می‌دهد، با افزایش τ، سربار ترافیک کاهش می‌یابد. مقدار بزرگتر برای τ به معنای آن است که افزونه چکیده‌ساز می‌تواند چکیده‌های بیشتری را ذخیره و در یک تراکنش ارسال کند و در نتیجه، سربار سطح شبکه کاهش می‌یابد. با این حال، تاثیر سربار سطح شبکه در $\tau = 16\,KB$ ناچیز می‌شود.

از سوی دیگر، SSLsplit که یک روش بازرسی TLS به روش مرد میانی است و همچنین روش رمزگذاری غیرفعال هیچ‌گونه سربار ترافیکی ایجاد نمی‌کنند. روش‌های رمزگذاری قابل جستجو به دلیل تنظیمات خاص اتصال و تبادل نشانه در حین انتقال داده، دارای مقداری سربار ترافیک هستند، اما این سربار ناچیز است.



شکل ۵. تاخیر انتقال بسته‌های شبکه توسط ناظر جرم‌شناسی

تاخیر ترافیک: یکی دیگر از خصوصیات مهم کارایی، تاخیر افزوده به ترافیک است، یعنی زمانی که ناظر جرم‌شناسی صرف پردازش و ارسال یک بسته می‌کند. انتظار می‌رود تاخیر کم باشد زیرا ناظر جرم‌شناسی بیشتر بسته‌ها را فوراً ارسال می‌کند. تنها زمانی که شرط معادله ۱ برای یک بسته برآورده نشود، بسته در میانگیر ناظر نگهداری می‌شود. احتمال وقوع این رویداد برای مقادیر پایین τ بیشتر است. ما میانگین تاخیر بسته را در ناظر جرم‌شناسی اندازه‌گیری کردیم. نتایج در شکل ۵ نشان داده شده‌اند. همان‌طور که انتظار می‌رفت، تاخیر کم و ناچیز است و در محدوده تاخیر دیوارهای حفاظتی سخت‌افزاری قرار دارد و حتی کمتر است. مقادیر ۲ و ۴ کیلوبایت برای τ منجر به نرخ میانگیری بسته بالا و در نتیجه تاخیر بیشتر می‌شود. با این حال، $\tau = 8\,KB$ و بالاتر، نرخ میانگیری بسته زیادی را اعمال نمی‌کند و باعث می‌شود که ناظر جرم‌شناسی بیشتر بسته‌ها را بدون معطلی ارسال کند، که منجر به تاخیر کمتر بسته می‌شود.

در مورد SSLsplit، تأخیر حدود ۶۰ میکروثانیه مشاهده شد. تاخیر SSLsplit بیشتر از حداقل تاخیر روش پیشنهادی است زیرا SSLsplit ترافیک را قبل از ارسال رمزگشایی و رمزگذاری می‌کند. روش‌های رمزگذاری قابل جستجو که تطبیق امضا انجام می‌دهند، می‌توانند تاخیر بیشتری داشته باشند که به تعداد قوانین و امضاهای آن‌ها بستگی دارد. با این حال، در بهترین حالت، تاخیر آن‌ها به اندازه ارسال فوری است. علاوه بر این، در فرایند پیچیده برقراری اتصال، تاخیر قابل‌توجهی ایجاد می‌کنند که می‌تواند تا یک ثانیه افزایش یابد. در نهایت، روش رمزگذاری غیرفعال تاخیر انتقال بسته‌ها را افزایش نمی‌دهد زیرا در وسط اتصال‌ها قرار ندارد و ترافیک را منتقل نمی‌کند.

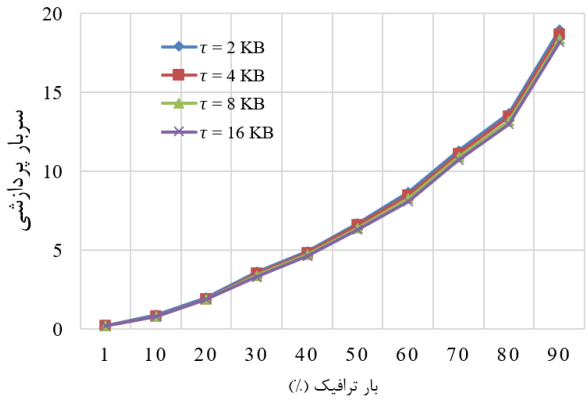
سربار پردازش: برای ارزیابی سربار پردازش افزونه چکیده‌ساز، میانگین استفاده از پردازنده کامپیوترها را در هر آزمایش اندازه‌گیری کردیم. شکل ۶ سربار پردازشی افزونه چکیده‌ساز را برای مقادیر مختلف τ نشان می‌دهد. از آنجا که بار پردازشی فرآیند چکیده‌سازی به بار ترافیک وابسته است، آزمایش‌ها را با تعداد وظایف مختلف تکرار کردیم که منجر به بارهای ترافیکی متفاوت شد. بنابراین، سربار پردازش به عنوان تابعی از بار ترافیک نشان داده شده است. به دلیل محدودیت پهنای باند اینترنت، تمام کامپیوترها به جز یکی را غیرفعال کردیم تا بتوانیم بار ترافیک یک کامپیوتر را به بیشینه نزدیک کنیم.

همان‌طور که مشاهده می‌شود، سربار پردازش برای بارهای ترافیکی

جدول ۱. مقایسه رویکردهای مختلف برای انتساب داده روی ترافیک رمز شده

	ضبط ترافیک خام	مرد میانی [۵،۱۹،۲۰]	رمزگشایی غیر فعال [۲۲]	رمزنگاری قابل جست‌وجو [۲۳-۲۶]	جرم‌یاب
موثر روی ترافیک رمز شده	خیر	بله	بله	بله	بله
حفظ محرمانگی	بله	خیر	خیر	بله	بله
عملی بودن	بله	بله	بله	خیر	بله
سربار ترافیک	ندارد	ندارد	ندارد	ندارد	دارد
تاخیر انتقال ترافیک	ندارد	دارد	ندارد	دارد	دارد
تاخیر برقراری اتصال	ندارد	دارد	ندارد	دارد	ندارد
سربار پردازشی روی کاربران	ندارد	ندارد	ندارد	دارد	دارد
سربار پردازشی	ندارد	ندارد	ندارد	ندارد	دارد

رنگ سبز و قرمز به ترتیب نشان دهنده ویژگی مثبت و منفی است



شکل ۶. سربار پردازشی افزونه چکیده‌ساز

در محدوده ۱٪ تا ۹۰٪ بین۰٫۰۴٪ تا ۱۸٪ متغیر است. علاوه بر این، سربار پردازش تقریبا مستقل از τ است. از سوی دیگر، SSLsplit هیچ‌گونه سربار پردازشی روی کامپیوترهای شبکه اعمال نمی‌کند. روش‌های رمزگذاری قابل جستجو به طور متوسط و بیشینه سربار پردازشی به ترتیب حدود ۱۰% و ۲۵٪ را گزارش کرده‌اند.

در نهایت، جدول ۱ مقایسه‌ای از همه روش‌ها ارائه می‌دهد. به طور خلاصه، تا آنجا که ما اطلاع داریم، جرم‌یاب تنها راه حل موجود کارا برای مشکل انتساب داده در ترافیک رمزگذاری شده استاندارد است که هم محرمانگی را حفظ می‌کند و هم پیاده‌سازی آن با در نظر گرفتن استانداردهای فعلی شبکه امکان‌پذیر است.

۵- جمع‌بندی

در این مقاله، جرم‌یاب را به‌عنوان یک راه‌حل عملی و بدون نقض محرمانگی برای مسئله انتساب داده در ترافیک رمزگذاری شده استاندارد (TLS) در شبکه‌های سازمانی معرفی کردیم. برای استفاده از این راهکار، سازمان باید کامپیوترهای خود را به یک افزونه نرم‌افزاری چگیده‌ساز مجهز کند که از داده‌های آشکار بالای لایه TLS چکیده‌های فشرده‌ای ایجاد کند. ما در این مورد بحث کردیم که چگونه با اتخاذ برخی سیاست‌ها و روش‌های امنیتی در سطح شبکه می‌توان صحت و تمامیت این راهکار را تضمین کرد. تجسس‌های آزمایشی واقع‌گرایانه ما نشان داد که روش جرم‌یاب می‌تواند به‌طور موثری برای انتساب رشته‌های داده منتقل شده از طریق TLS استفاده شود. ارزیابی‌های انجام‌شده نشان داد که سربار ترافیک و همچنین سربار پردازش راهکار پیشنهادی معقول است. در کارهای آینده، قصد داریم سامانه‌های تشخیص نفوذ و همچنین سامانه‌های پیشگیری از نشت داده را با استفاده از روش خود پیاده‌سازی کنیم تا بتوانند وظایف خود را روی ترافیک رمزگذاری شده انجام دهند. این کار با ارائه چکیده‌های امضاهای مخرب (یا چکیده‌های اطلاعات محرمانه) به آنها و مقایسه این چکیده‌ها با چکیده‌های ترافیک در دروازه شبکه قابل انجام به نظر می‌رسد.

مراجع

[1] Porter Felt, A., Barnes, R., King, A., Palmer, C., Bentzel, C. and Tabriz, P., Measuring HTTPS Adoption on the Web. 26th USENIX Security Symposium (2017), 1323–1338.

[2] de Carnavalet, X. de C. and van Oorschot, P.C., A survey and analysis of TLS interception mechanisms and motivations. ACM Computing Surveys. (Jan. 2023)

[3] Erlacher, F., Woertz, S. and Dressler, F., A TLS Interception Proxy with Real-Time Libpcap Export. 41st IEEE Conference on Local Computer Networks (Nov. 2016), 1–3.

[4] Sophos XG Firewall: https://www.sophos.com/en-us/products/next-gen-firewall.

[5] Symantec SSL visibility appliances: https://www.broadcom.com/products/cybersecurity/network/encrypted-traffic-management/ssl-visibility-appliance.

[6] Durumeric, Z., Ma, Z., Springall, D., Barnes, R., Sullivan, N., Bursztein, E., Bailey, M., Halderman, J.A. and Paxson, V., The security impact of HTTPS interception. Proceedings 2017 Network and Distributed System Security Symposium (San Diego, USA, Feb. 2017), 1–14.

[7] O'Neill, M., Ruoti, S., Seamons, K. and Zappala, D., TLS Proxies: Friend or Foe? Proceedings of the 2016 Internet Measurement Conference (New York, NY, USA, Nov. 2016), 551–557.

[8] Waked, L., Mannan, M. and Youssef, A., The Sorry State of TLS Security in Enterprise Interception Appliances. Digital Threats: Research and Practice. 1, 2 (Jun. 2020), 1–26.

[9] Waked, L., Mannan, M. and Youssef, A., To Intercept or Not to Intercept: Analyzing TLS Interception in Network Appliances. Proceedings of the 2018 on Asia Conference on Computer and Communications Security (New York, NY, USA, May 2018), 399–412.

[10] DPDK: Data Plane Development Kit: https://www.dpdk.org

[11] Shanmugasundaram, K., Brönnimann, H. and Memon, N.D., Payload attribution via hierarchical Bloom filters. Proceedings of the 11th ACM Conference on Computer and Communications Security (Washington, USA, Oct. 2004), 31–41.

[12] Bloom, B.H., Space/time trade-offs in hash coding with allowable errors. Communications of the ACM. 13, 7 (1970), 422–426.

[13] Roussev, V., Hashing and data fingerprinting in digital forensics. IEEE Security & Privacy. 7, 2 (2009), 49–55.

[14] Ponec, M., Giura, P., Brönnimann, H. and Wein, J., Highly efficient techniques for network forensics. Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07) (Alexandria, USA, Oct. 2007), 150–160.

[15] Ponec, M., Giura, P., Wein, J. and Brönnimann, H., New payload attribution methods for network forensic investigations. ACM Transactions on Information and System Security. 13, 2 (2010), 1–32.

[16] Haghighat, M.H., Tavakoli, M. and Kharrazi, M., Payload attribution via character dependent multi-Bloom filters. IEEE Transactions on Information Forensics and Security. 8, 5 (2013), 705–716.

[17] Hosseini, S.M. and Jahangir, A.H., An effective payload attribution scheme for cybercriminal detection using compressed bitmap index tables and traffic downsampling. IEEE Transactions on Information Forensics and Security. 13, 4 (Apr. 2018), 850–860.

[18] Hosseini, S.M., Jahangir, A.H. and Kazemi, M., Digesting network traffic for forensic investigation using digital signal processing techniques. IEEE Transactions on Information Forensics and Security. 14, 12 (2019), 3312–3321.

[19] mitmproxy: https://mitmproxy.org/. Accessed: 2023-04-08.

[20] SSLsplit: https://www.roe.ch/SSLsplit.

[21] Lesniewski-Laas, C. and Kaashoek, M.F., SSL Splitting: Securely Serving Data from Untrusted Caches. 12th USENIX Security Symposium (2003).

[22] Wilkens, F., Haas, S., Amann, J. and Fischer, M., Passive, Transparent, and Selective TLS Decryption for Network Security Monitoring. (2022), 87–105.

[23] Sherry, J., Lan, C., Popa, R.A. and Ratnasamy, S., BlindBox: Deep packet inspection over encrypted traffic. Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication - SIGCOMM '15 (London, United Kingdom, Aug. 2015), 213–226.

[24] Fan, J., Guan, C., Ren, K., Cui, Y. and Qiao, C., SPABox: Safeguarding privacy during deep packet inspection at a middleBox. IEEE/ACM Transactions on Networking. 25, 6 (2017), 3753–3766.

[25] Ning, J., Poh, G. Sen, Loh, J.C., Chia, J. and Chang, E.C., PrivDPI: Privacy-Preserving Encrypted Traffic Inspection with Reusable Obfuscated Rules. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (New York, NY, USA, Nov. 2019), 1657–1670.

[26] Ning, J., Huang, X., Poh, G. Sen, Xu, S., Loh, J.-C., Weng, J. and Deng, R.H., Pine: Enabling Privacy-Preserving Deep Packet Inspection on TLS with Rule-Hiding and Fast Connection Establishment. European Symposium on Research in Computer Security (2020), 3–22.

[27] Damie, M., Hahn, F. and Peter, A., A Highly Accurate Query-Recovery Attack against Searchable Encryption using Non-Indexed Documents. 30th USENIX Security Symposium (2021), 143–160.

[28] Ning, J., Xu, J., Liang, K., Zhang, F. and Chang, E.-C., Passive Attacks Against Searchable Encryption. IEEE Transactions on Information Forensics and Security. 14, 3 (2019), 789–802.

[29] Xu, H., Zhou, Y., Ming, J. and Lyu, M., Layered obfuscation: a taxonomy of software obfuscation techniques for layered security. Cybersecurity. 3, 1 (Dec. 2020), 1–18.

[30] Palanisamy, R., Norman, A.A. and Kiah, M.L.M., Compliance with Bring Your Own Device security policies in organizations: A systematic literature review. Computers & Security. (2020), 101998.

[31] Safa, N.S., Von Solms, R. and Furnell, S., Information security policy compliance model in organizations. Computers & Security. 56, (2016), 70–82.

[32] Achleitner, S., Burke, Q., McDaniel, P., Jaeger, T., La Porta, T. and Krishnamurthy, S., MLSNet: A policy complying multilevel security framework for software defined networking. IEEE Transactions on Network and Service Management. 18, 1 (2021), 729–744.

[33] Perales, A.P., Adding Support for Automatic Enforcement of Security Policies in NFV Networks. IEEE/ACM Transactions on Networking. 27, 2 (2019), 707–720.

[34] Lara, A. and Ramamurthy, B., OpenSec: Policy-Based Security Using Software-Defined Networking. IEEE Transactions on Network and Service Management. 13, 1 (2016), 30–42.

[35] Mozilla's webRequest APIs: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/webRequest.

[36] Docker virtualization platform: https://www.docker.com.

[37] Merkel, D. 2014. Docker: lightweight Linux containers for consistent development and deployment. Linux Journal. 2014, 239 (2014).

[38] Naik, N., Migrating from virtualization to dockerization in the cloud: Simulation and evaluation of distributed systems. IEEE 10th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Environments (MESOCA) (Raleigh, USA, Oct. 2016), 1–8.

[39] Ramalho, F. and Neto, A., Virtualization at the network edge: A performance comparison. IEEE 17th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM) (Coimbra, Portugal, Jun. 2016), 1–6.

[1] Confidentiality
[2] Integrity
[3] Data exfiltration
[4] Deep packet inspection
[5] Network forensics
[6] Payload attribution
[7] Post-mortem investigations
[8] Connection
[9] Worm
[10] Plain

[11] Transport Layer Security
[12] Client
[13] Server
[14] Authorized
[15] Authenticated
[16] False positive
[17] Forensic analyst
[18] Hash
[19] Wild-card character
[20] Character
[21] Polymorphic worms
[22] Packet
[23] Link
[24] TLS interception
[25] Middlebox
[26] Man-in-the-middle
[27] Browser
[28] Proxy
[29] Terminate
[30] TLS session splitting
[31] Perfect forward secrecy
[32] Authenticated encryption with associated data
[33] Cipher suite
[34] Searchable encryption
[35] Flow
[36] Honest-but-curious
[37] Soundness
[38] Completeness
[39] Gateway
[40] Hash-based message authentication code
[41] Hard-code
[42] Obfuscation
[43] Buffer
[44] Switch
[45] Polling
[46] End-to-end
[47] Cache