دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۱، جلد ۲۰، شماره ۱ (۵۹-۶۶)
https://monadi.isc.org.ir

ارائه‌شده در هجدهمین کنفرانس بین‌المللی انجمن رمز ایران در دانشگاه اصفهان، ۱۰ و ۱۱ شهریور ۱۴۰۰

ابزاری برای شناسایی حملات فردی در مرورگر با استفاده از تحلیل پویای صفحات وب★

مجید ایرانپور مبارکه* و بهروز ترک لادانی

گروه مهندسی فناوری اطلاعات، دانشگاه اصفهان، اصفهان، ایران

اطلاعات مقاله

کلمات کلیدی:
حمله فردی در مرورگر
بدافزار
تشخیص بدافزار
تحلیل پویا

dor: 20.1001.1.24763047.1401.11.1.7.4

نوع مقاله: پژوهشی

چکیده

امروزه تشخیص حملات سطح مرورگر چالشی جدی برای حفاظت از اطلاعات کاربران محسوب می‌شود. حمله فردی در مرورگر (MitB)، نوع مهمی از این حملات است که با استفاده از تروجان‌ها می‌تواند منجر به تغییر در محتویات صفحه وب، مداخله در ترافیک شبکه، سرقت نشست و سرقت اطلاعات کاربر شود. در این مقاله ابزاری کارآمد برای شناسایی بلادرنگ حملات MitB از طریق تحلیل پویای صفحات وب بر اساس توصیف الگوی حملات ارائه می‌شود. مزیت ابزار پیشنهادی نسبت به روش‌های مشابه این است که محدود به شناسایی یک یا چند حمله خاص نیست و کد روش شناسایی در ابزار تعبیه نشده است، بلکه الگوی حملات مختلف به‌صورت جداگانه توصیف می‌شود. جهت ارزیابی ابزار ارائه‌شده، دو سرویس وب آسیب‌پذیر ارائه‌شده توسط OWASP که دارای طیف وسیعی از آسیب‌پذیری‌های شناخته‌شده هستند، به همراه چارچوب آزمون نفوذپذیری BeEF مورد استفاده قرار گرفت و مجموعه‌ای از حملات MitB به‌صورت عملی پیاده‌سازی و توسط ابزار مورد ارزیابی قرار گرفت. همین آزمایش‌ها با استفاده از سه ابزار مشابه دیگر نیز عیناً انجام و با ابزار ارائه‌شده مقایسه شد. علاوه بر برتری ابزار ارائه‌شده از جهت استقلال توصیف حملات از خود ابزار، نتایج حاصل نشان‌دهنده بهتر بودن معیارهای دقت و بازخوانی تشخیص آن نسبت به ابزارهای مشابه است.

© ۱۴۰۱ انجمن رمز ایران

۱ مقدمه

امروزه سوءاستفاده مهاجمان از طریق کدهای مخربی که اصطلاحاً بدافزار[1] نام دارد، تهدیدی جدی برای سازمان‌ها و کاربران است که می‌تواند صدمات جبران‌ناپذیری به سیستم‌ها وارد کند. سرقت نشست‌های[2] کاربر، اطلاعات هویتی، اطلاعات مالی، مختل کردن سیستم و پاک کردن فایل‌ها تنها بخشی از آثار منفی بدافزارها است. امروزه محیط شبکه‌های کامپیوتری و اینترنت بستر مناسبی برای فعالیت و انتشار بدافزارهای گوناگون محسوب می‌شوند و مرورگرها یکی از نقاط تمرکز اصلی مهاجمان برای اجرای حملات است. قدرتی که بدافزارها در اختیار مهاجمین قرار می‌دهند، باعث شده تا آن‌ها به اجرای حملات مختلف در بستر مرورگرها روی آورند. حملات اجراشده می‌تواند به ارسال داده‌ها برای مهاجم با ویژگی پنهان‌سازی ارتباط خود بپردازند. در واقع این نوع حملات، تروجان‌هایی[3] هستند که محیط مرورگر را آلوده ساخته و می‌توانند به تغییر در محتویات صفحه وب، مداخله در ترافیک شبکه [۱]، سرقت نشست‌های کاربر [۲] و سرقت اطلاعات محرمانه کاربر نظیر نام کاربری و کلمه عبور بپردازند؛ که نتیجه آن بسیار کاراتر از حملات متعارف مرد میانی[4] است [۳]. به این نوع حملات، حملات فردی در مرورگر (MitB[5])، می‌گویند. برای

★از کمیته علمی هجدهمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.

*نویسنده مسئول
آدرس‌های رایانامه: majidiranpour@eng.ui.ac.ir (مجید ایرانپور مبارکه)، ladani@eng.ui.ac.ir (بهروز ترک لادانی)

© ۱۴۰۱ تمامی حقوق متعلق به انجمن رمز ایران است.

[1] malware [2] session

[3] trojan [4] Man-in-the-Middle [5] Man-in-the-Browser

مثال تروجان‌های Zeus و URLZone Bebloh، برای سرقت اعتبارنامه و تغییر محتوای ارسال شده به میزبان نمونه‌هایی از این نوع حملات را در عمل پیاده‌سازی کرده‌اند [۳].

حملات MitB، از توابع و ویژگی‌های مختلف یک مرورگر استفاده می‌کنند. از این رو افزونه‌های مرورگر که با اهداف گوناگون، گشت و گذار در وب را برای کاربران راحت‌تر می‌کنند، فرصت خوبی را برای مهاجمین فراهم کرده تا با توسعه یک افزونه بدافزارگونه، بعد از نصب شدن آن در مرورگر کاربر، به اجرای حملات MitB بپردازند [۴]. افزونه‌های بدافزارگونه، همچنین می‌توانند، با استفاده از ساز و کار API hooking، با استفاده از تغییر آنچه کاربر در مرورگر می‌بیند، با تغییر مقادیر فرم‌ها، تغییر مقادیر دریافتی و ...، حملات MitB را اجرا نمایند [۵]. وجود برخی از آسیب‌پذیری‌ها در صفحه وب نیز به عنوان فرصت مناسبی برای بهره‌برداری از این حملات، با استفاده از ویژگی‌های DOM صفحات وب است [۴]. از این رو حملاتی مثل فیشینگ [۱]، کلیک دزدی [۲] و استخراج‌کنندگان رمزارز [۳]، از جمله حملات خانواده MitB، هستند.

با توجه به عملکرد حملات MitB، از آنجا که این حملات در سمت مشتری انجام می‌گیرند، بسیاری از سیستم‌های سطح شبکه مانند دیواره آتش [۴] و سیستم‌های تشخیص نفوذ [۵]، در تشخیص این حملات با مشکل روبرو هستند [۴] و همچنین ساز و کارهای امنیتی، مانند SSL و یا احراز هویت دوعاملی [۶]، نیز مقاومتی در برابر این حملات ایجاد نمی‌کنند [۳].

استفاده روزمره کاربران از مرورگر باعث شده که اهمیت شناسایی و جلوگیری از آسیب‌های این نوع حملات، بیش‌تر از پیش مورد توجه قرار گیرد. استفاده از روش‌های پنهان‌کارانه، نظیر شبیه‌سازی رفتار بدخواهانه به رفتار کاربر واقعی، مبهم کردن کد، استفاده از ارتباطات شبکه‌ای رمزگذاری شده، تغییر رفتار بدافزارها در شرایط مختلف و هدایت کاربر به وب‌سایت اصلی، به منظور عدم جلب توجه کاربر از وجود حمله، باعث شده تا حملات MitB به سختی قابل تشخیص باشند [۲].

در این پژوهش، هدف شناسایی حملات MitB است که از طریق سوءاستفاده از افزونه‌های مخرب، هدف مهاجم را برآورده می‌کنند. از این رو این مقاله به ارائه یک ابزار جهت تشخیص این حملات پرداخته که با استفاده از تحلیل پویای صفحات وب بر اساس توصیف الگوی حملات عمل می‌کند. مزیت ابزار پیشنهادی نسبت به روش‌های مشابه این است که فقط به شناسایی یک یا چند حمله خاص محدود نیست و کد روش شناسایی در ابزار تعبیه نشده است، بلکه الگوی حملات مختلف به صورت جداگانه توصیف می‌شود.

جهت ارزیابی ابزار پیشنهادی، از سرویس‌های آسیب‌پذیر bWAPP و Mutillidae و همچنین از چارچوب تست نفوذ BeEF استفاده شده است. همچنین آزمایش‌هایی با هدف بررسی کارایی ابزار پیشنهادی بر اساس رویکرد GQM [۷] طراحی و انجام شده است. ابزار پیشنهادی علاوه بر مزیت ذکر شده توانست به‌دقت ٪۸۰ و بازخوانی ٪۱۰۰ دست یابد که بهتر از ابزارهای قبلی مشابه است.

در ادامه مقاله ابتدا مرور مختصری بر کارهای مرتبط ارائه می‌شود، سپس به تشریح معماری و روش عملکرد ابزار پیشنهادی می‌پردازیم. در ادامه پیاده‌سازی و نحوه ارزیابی ابزار مورد بحث قرار می‌گیرد و نهایتاً مقاله با جمع‌بندی به انتها می‌رسد.

۲ مرور کارهای مرتبط

کارهای که در جدول ۱ مورد بررسی قرار گرفته است، مهم‌ترین نمونه‌ها از کارهای پیشین و مشابه با هدف این پژوهش انتخاب شده‌اند که به اختصار روش تشخیص حملات و نقاط قوت و ضعف هر کدام بیان شده است. همان طور که مشاهده می‌شود، تمامی روش‌های فوق دارای نقاط ضعف بوده و هیچ روشی از تحلیل صفحات وب بر اساس توصیف الگوی حملات MitB، به جای استفاده از یک روش، جهت تشخیص حملات، استفاده نکرده است.

۳ روش و ابزار پیشنهادی

بنیاد OWASP، اجرای حملات فردی در مرورگر را در قالب سوءاستفاده از چهار جزء Browser Helper Objects، افزونه‌ها، API-Hooking و کرم‌های JavaScript تقسیم‌بندی کرده است. Browser Helper Object کتابخانه‌های پویایی است که مرورگر Internet Explorer از آن برای افزایش کارایی کاربران استفاده می‌کند و با توجه به استفاده بسیار کم کاربران از این مرورگر، سوءاستفاده از این روش کمتر مورد استقبال نفوذگران است. برای سوءاستفاده از API-Hooking، نیاز به نوشتن بدافزار برای سیستم‌عامل‌های مختلف است که مهاجم را درگیر مباحثی مثل مبهم سازی کد، درک APIهای سیستم‌عامل، دور زدن آنتی‌ویروس و ... می‌کند. از این رو این روش نیز کمتر مورد استقبال نفوذگران است. از طرفی هم کرم‌های نوشته‌شده با زبان JavaScript، اغلب فقط روی یک سایت آسیب‌پذیر و یک حمله خاص کارآمد خواهند بود. در مقابل، در روش بکارگیری افزونه‌ها، مهاجم با حداقل دانش از JavaScript می‌تواند بدافزار خود را طراحی کند و درگیر مسائلی که در طراحی بدافزار برای سیستم‌عامل است، نخواهد شد همچنین به خاطر دسترسی داشتن به APIهای مرورگر و DOM صفحات وب، کارایی بیشتری خواهند داشت و امکان اجرای طیف وسیعی از حملات مختلف از طریق ارتباط با مرکز کنترل (C2) را دارند. از این رو در این پژوهش شناسایی حملات MitB که توسط افزونه‌های مخرب اجرا می‌شوند، بر اساس توصیفی از الگوی حمله که کاربر به ابزار معرفی می‌کند، انجام می‌شود.

با توجه به پیامدهای حملات MitB که به آن اشاره شد، مهم‌ترین موردی که در این حملات بسیار مهم است، تشخیص به‌موقع آن‌هاست. استفاده از سیستم‌های برون‌خط برای تشخیص این حملات اگر چه کارآمد است اما با تشخیص دیر موقع، می‌تواند اثرات مخرب زیادی را به بار آورد. از

[1] phishing [2] clickjacking [3] Cryptocurrency mining [4] firewalls [5] Intrusion Detection System(IDS) [6] Two factor authentication [7] Goal, Question, Metric

جدول ۱. مقایسه کارهای مرتبط با پژوهش

پژوهشگران	سال انتشار مقاله	روش تشخیص	نقاط قوت	نقاط ضعف
پوادی و همکاران [۶]	۲۰۱۶	تشخیص حملات کلیک دزدی با استفاده از افزونه ClickProtect	تشخیص حملات کلیک دزدی شامل چهارچوب نامرئی و تغییر مکان اشارگر	عدم تحلیل پیامدهای کد مخرب JavaScript
ساشاگری و همکاران [۷]	۲۰۱۶	شناسایی اسکریپت‌های JavaScript بدخواه با روش AMA	بررسی پیوندهای کد JavaScript بر اساس کاراکترهای مشکوک	عدم شناسایی حملات تزریق کد
ژانگ و همکاران [۸]	۲۰۱۶	کشف بدافزارهای پنهان با ایجاد درخت رخداد و بررسی ریشه آن	تمایز میان رفتار واقعی کاربر از رفتار بدخواه بدافزارها	تشخیص نادرست در حملاتی مثل فیشینگ
لالیا و همکاران [۹]	۲۰۱۹	توسعه افزونه CSRFDetector و آنالیز درخواست‌ها و محتوای وب، جهت تشخیص حملات CSRF	تشخیص گام‌به‌گام حملات CSRF، با بررسی محتویات صفحه وب و درخواست‌های ارسالی	سربار زیاد
رحیمی [۱۰]	۲۰۱۹	شناسایی جملات کلیک دزدی، فیشینگ و بدافزارهای MitB، با توسعه ابزار Xsec	تشخیص حملات کلیک دزدی با تحلیل برچسب‌های مختلف، تشخیص حملات فیشینگ و تشخیص بدافزارهای MitB، از روی گراف رابطه رخدادی	عدم شناسایی حملات جدید MitB

شکل ۱. معماری ابزار

طرف دیگر استفاده از سیستم‌های بلادرنگ، نیازمند مدیریت دقیق منابع است تا با بالا رفتن میزان استفاده از پارامترهایی مانند RAM و CPU، باعث کاهش سرعت پاسخگویی و اختلال در مرورگرها نشوند. از این رو برای جلوگیری از مشکلات مطرح شده و قابل اجرا بودن ابزار پیشنهادی برای سیستم‌های ضعیف و یا سازمان‌ها، از معماری سرویس محور برای توسعه ابزار استفاده شده است.

در ادامه این بخش، ابتدا معماری کلی ابزار پیشنهادی مرور خواهد شد و سپس دو ماژول اصلی مورد استفاده در آن به تفصیل توضیح داده می‌شود.

۱.۳ معماری ابزار پیشنهادی

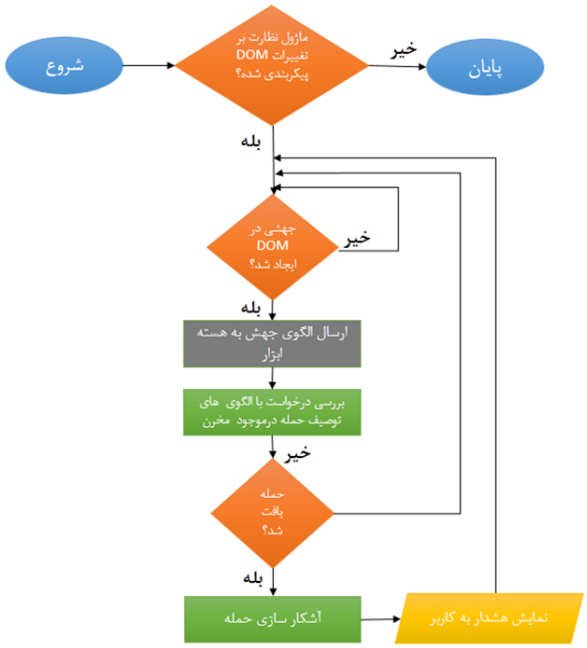
معماری ابزار پیشنهادی از سه جزء اصلی تشکیل‌شده که در شکل ۱ نشان داده شده است.

افزونه مرورگر: افزونه مرورگر علاوه بر بهره بردن از APIهای وب، به مجموعه APIهای خود JavaScript نیز دسترسی دارد. این بدان معناست که توسعه‌دهنده می‌تواند در یک افزونه مرورگر کارهای بسیار بیشتری نسبت به یک صفحه وب انجام دهد. از این رو با توسعه یک افزونه، به جمع‌آوری الگوهای مختلف از صفحه وب، برای کشف حملات MitB که توسط افزونه بدافزارگونه بر روی صفحه وب انجام می‌شود، پرداخته شد. استفاده از این قابلیت مزیت‌های زیر را همراه دارد:

- اطمینان از اینکه تمام الگوهای درخواست‌های ارسال شده به ابزار پیشنهادی از طریق مرورگر انجام شده و نیاز به مشاهده تمام ترافیک عبوری از سیستم و تفکیک ترافیک مرورگر نیست.
- چنانچه که یک افزونه بدافزارگونه از پروتکل‌های رمز شده مثل HTTPS و یا WSS استفاده کند، نمی‌توان از ترافیک خروجی مرورگر به بررسی وجود الگوی حمله پرداخت اما با استفاده از افزونه توسعه داده شده، می‌توان الگوی درخواست‌ها را پیش از رمز شدن توسط مرورگر دریافت و تحلیل کرد.
- افزونه‌های بدافزارگونه از طریق دسترسی داشتن به مدل DOM برای هر صفحه، می‌توانند به راحتی به تغییر آن برای اهداف مخرب خود بپردازند. برای مثال به تغییر آدرس ارسال یک فرم لاگین بپردازند تا نام کاربری و رمز عبور کاربر قربانی را به دست آورند. از این رو با توسعه افزونه پیشنهادی از طریق APIهای JavaScript می‌توان به جمع‌آوری الگوی تغییرات در DOM پرداخت.

هسته ابزار: این بخش از ابزار به عنوان یک سرویس بر روی سیستم کاربران نصب می‌گردد. افزونه ابزار پیشنهادی نیز، با این سرویس در ارتباط بوده و الگوهای جمع‌آوری‌شده از هر صفحه وب را برای آن ارسال می‌کند. هسته وظیفه تشخیص حمله با استفاده از مقایسه الگوی دریافتی از صفحه وب با الگوی توصیف حملات در مخزن و ارسال نتیجه به افزونه مرورگر را دارد.

مخزن توصیف الگوی حملات: این مخزن به عنوان پایگاه داده‌ای از تمامی الگوهای توصیف حملات MitB محسوب می‌شود. هسته ابزار همواره با این بخش از ابزار در تعامل بوده تا با استفاده از الگوهای ذخیره شده در آن به کشف حملات بپردازد.



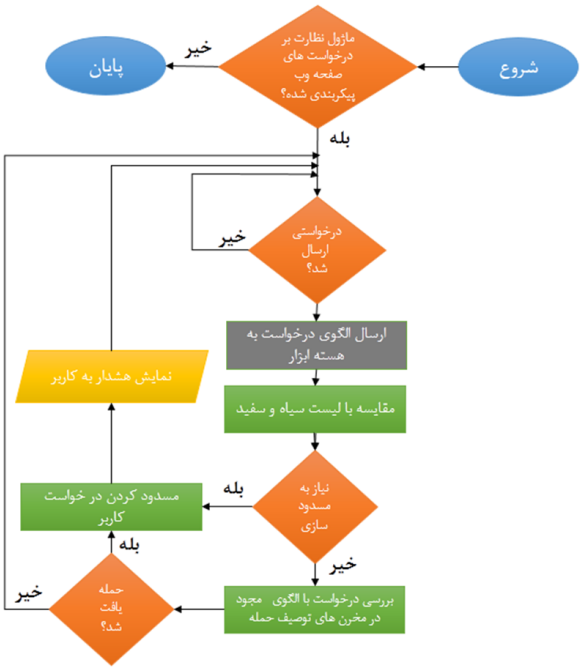
شکل ۲. عملکرد ماژول نظارت بر تغییرات DOM

۲.۳ فرآیند تشخیص مخرب بودن الگوها

همان‌طور که اشاره شد، افزونه مرورگر وظیفه جمع‌آوری جهش‌ها [1] و درخواست‌هایی که از صفحه وب ارسال می‌شوند را بر عهده دارد. پس‌ازآن توصیف الگوی هر یک به هسته ارسال می‌شود. هسته ابزار نیز با تبدیل الگو به یک دستور جست‌وجو در مخزن داده، به دنبال تشخیص مخرب بودن یا نبودن آن بوده و نتیجه را به افزونه اعلام می‌نماید. این فرایند در ابزار پیشنهادی توسط دو ماژول انجام می‌شود:

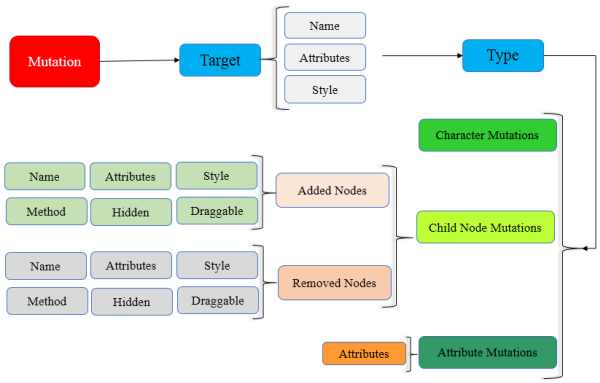
ماژول نظارت بر تغییرات DOM: این ماژول وظیفه جمع‌آوری جهش‌هایی که در صفحه وب رخ می‌دهند و بررسی مخرب بودن یا نبود آن را بر عهده دارد. کد این بخش افزونه می‌تواند به APIهای صفحه وب دسترسی داشته باشد و از رابط MutationObserver که به‌عنوان جایگزینی برای ویژگی قدیمی Mutation Events طراحی‌شده، استفاده شده است. این رابط توانایی نظارت بر تغییرات ایجادشده در درخت DOM را فراهم می‌کند. از این رو می‌توان به جمع‌آوری جهش‌هایی که در صفحه وب رخ می‌دهد پرداخت. بعد از اینکه جهشی در صفحه وب ایجاد شد، توصیف الگوی آن به هسته ابزار ارسال می‌شود. اکنون ابزار که به مخزن توصیف الگوی حملات متصل است، به تبدیل توصیف الگو به یک دستور جست‌وجو می‌پردازد. چنانچه حمله‌ای برای آن کشف شد، هشدار لازم را به افزونه مرورگر ارسال نموده تا توسط افزونه به کاربر نمایش داده شود. با توجه به آنچه شرح داده شد و اینکه این ماژول از روی رفتار افزونه‌ها و ساختار تغییراتی که انجام می‌دهند، در حین اجرای افزونه به تشخیص حمله می‌پردازد، پس عملکرد این ماژول در تشخیص حملات از نوع تحلیل رفتاری پویا و تحلیل ساختاری پویا است. شکل ۲ نحوه عملکرد این ماژول را نمایش می‌دهد.

[1] mutation



شکل ۳. عملکرد ماژول نظارت بر درخواست‌های صفحه وب

ماژول نظارت بر درخواست‌های صفحه وب: افزونه‌ها می‌تواند به APIهای صفحه وب دسترسی داشته باشند. از این رو یک افزونه بدافزار گونه می‌تواند از طریق شئ XMLHttprequest و یا fetch، به ارسال درخواست‌های مخرب بپردازد. پس در ابزار پیشنهادی وظیفه این ماژول بررسی مخرب بودن یا نبودن درخواست‌های ارسالی از صفحه وب است و در صورت مواجه شدن با یک درخواست مخرب، به مسدود کردن آن بپردازد و هشدار لازم را به کاربر نمایش دهد. در افزونه ابزار جهت شنود درخواست‌ها و ارسال الگوی توصیف هر آن‌ها به هسته ابزار، از رخداد onBeforeRequest، استفاده شده است که می‌تواند قبل از ارسال درخواست، اطلاعات مربوط به سرایند و بدنه را شنود کند. در بعضی از موارد تحلیل‌گرهای بدافزارها و محققین امنیتی بعد از شناسایی حملاتی که به ارتباط با مرکز کنترل خود می‌پردازند و یا وب‌سایت‌هایی که عملکرد مخرب دارند، به انتشار لیست دامنه و یا آدرس IP آن می‌پردازند. از این رو می‌توان برای تسریع در بررسی مخرب بودن یا نبود درخواست‌ها از یک لیست سیاه از دامنه و IPهای مخرب استفاده کرد و چنانچه درخواست ارسالی به یکی از موارد آن بود، درخواست مسدود شود. به طریق مشابه کاربر می‌تواند با معرفی لیست سفید از دامنه یا آدرس IP آن‌ها، به روند کار ابزار سرعت ببخشد. همان‌طور که بیان شده این ماژول درواقع از روی امضاء هر درخواست ارسالی از صفحه وب و هنگامی‌که افزونه‌های دیگر در حال اجرا هستند، درخواست‌های مخرب را شناسایی می‌کند. از این رو رویکرد این ماژول در شناسایی حملات از نوع تحلیل مبتنی بر امضاء پویا است. شکل ۳، نحوه عملکرد این ماژول را نمایش می‌دهد.



شکل ۴. مراحل توصیف الگو در ماژول نظارت بر تغییرات DOM

۳.۳ توصیف الگو برای ماژول نظارت بر تغییرات DOM

در این بخش به نحوه‌ی توصیف الگوی حملات بر روی DOM و توصیف الگوی جهش‌ها پرداخته شده است. برای این منظور به ترتیب گام‌های زیر برای هر الگو باید اجرا شود:

- ابتدا باید بررسی شود که تغییر روی چه برچسبی[1] است. برای این منظور برچسب مورد نظر باید با استفاده از نام برچسب، صفت‌های[2] آن و ویژگی‌های CSS مشخص شود.
- پس از آن باید نوع تغییری که انجام گرفته است مشخص گردد. این تغییر می‌تواند یکی از حالات زیر باشد:
  - تغییر متن یک برچسب
  - اضافه یا حذف یک برچسب فرزند
  - تغییر صفت‌های یک برچسب
- اگر تغییر از نوع اضافه یا حذف یک برچسب فرزند باشد، باید نوع برچسبی که حذف یا اضافه شده است با استفاده از ترکیب نام برچسب، صفت‌ها، ویژگی‌های CSS، نوع متد ارسال درخواست، قابل مشاهده بودن یا نبودن و قابل‌جابه‌جایی بودن یا نبودن آن مشخص شود.
- اگر تغییر از نوع تغییر صفت‌های یک برچسب باشد، باید نام صفت مشخص شود.

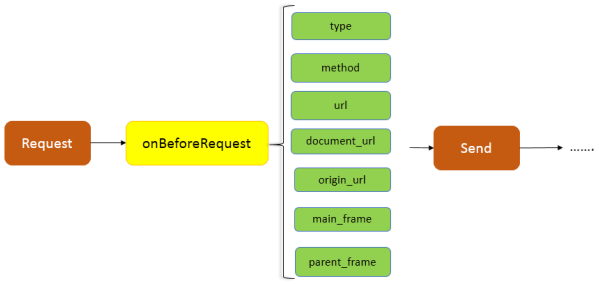
با استفاده از موارد گفته شده می‌توان به توصیف الگوی حمله یا تغییرات ایجادشده در DOM پرداخت. شکل ۴، مراحل گفته شده را نمایش می‌دهد.

۴.۳ توصیف الگو برای ماژول نظارت بر درخواست‌های صفحه وب

در این بخش به نحوه‌ی توصیف الگوی حملات از طریق ارسال درخواست و توصیف الگوی درخواست‌های صفحه وب، پرداخته شده است. برای این منظور هر درخواست در رخداد onBeforeRequest بررسی شده که مهم‌ترین پارامترهای آن، موارد زیر است:

- **type**: نوع درخواست را بیان می‌کند مثلاً درخواست می‌تواند از نوع دریافت فونت، عکس یا یک فریم باشد.

[1]tag [2]attribute



شکل ۵. عملکرد رخداد onBeforeRequest

- **method**: نوع متد ارسال درخواست را مشخص می‌کند.
- **url**: مشخص‌کننده مقصد ارسال درخواست است.
- **documentUrl**: آدرس سندی است که منابع در آن بارگیری می‌شوند.
- **originUrl**: آدرس مرجعی که درخواست را ارسال کرده است.
- **document**: مشخص‌کننده اینکه آیا درخواست از سند اصلی ارسال شده است.
- **origin**: مشخص‌کننده اینکه آیا درخواست از مرجع اصلی ارسال شده است.
- **mainFrame**: مشخص‌کننده اینکه آیا درخواست از فریم اصلی صفحه ارسال شده است.
- **parentFrame**: مشخص‌کننده اینکه آیا درخواست از فریم پدر صفحه ارسال شده است.

۴ پیاده‌سازی ابزار و ارزیابی روش پیشنهادی

برای پیاده‌سازی ابزار پیشنهادی از زبان JavaScript برای ایجاد افزونه مرورگر استفاده شد. خروجی این بخش می‌تواند به صورت یک افزونه بر روی مرورگر FireFox نصب شود اما با توجه به قابلیت‌های یکسان APIهای توسعه افزونه، می‌توان با اندکی تغییر این خروجی را بر روی دیگر مرورگرها نیز نصب و استفاده کرد. برای پیاده‌سازی هسته ابزار از زبان Python3 استفاده شد که به صورت یک سرویس در سیستم کاربر اجرا می‌شود. افزونه‌ها برای ارتباط با سیستم‌عامل باید از قوانین مرورگر در این خصوص پیروی کنند. از این رو ارتباط بین افزونه مرورگر و هسته ابزار از فرآیند Native Messaging استفاده می‌کند و افزونه می‌تواند پیام‌ها را در قالب JSON با هسته تبادل کند. ابزار پیشنهادی به گونه‌ای توسعه داده شده تا بتواند با مخازن متفاوت ارتباط برقرار کند. از این رو می‌توان از یک فایل یا یک پایگاه داده استفاده شود. به دلیل اینکه از قالب JSON برای توصیف الگوی حملات استفاده شده است، چنانچه از پایگاه داده برای این منظور استفاده شود، باید حتماً از نوع NoSQL باشد. برای تبدیل الگوی دریافتی به یک دستور جست‌وجو درون فایل از کتابخانه py-jsonq استفاده شده است. همچنین به طور پیش‌فرض امکان استفاده از ElasticSearch نیز توسط کتابخانه elasticsearch نیز در ابزار فراهم شده است. در حالت کلی نیز امکان استفاده از هر کتابخانه‌ی دیگر وجود دارد که برای این کار کافی است در کد هسته ابزار، تابع "send_query"

جدول ۲. فهرست حملات استفاده شده در ارزیابی

	BeEF	Mutillidae	bWAPP
حملات	Clickjacking	HTML Injection	iframe Injection
	Replace HREFs	JavaScript Injection (Password Generator)	Clickjacking
	Replace HREFs (HTTPS)	JavaScript Injection (Discussion of Back Button)	
	Redirect Browser (iframe)	Sensitive data expose	
	iframe Event Logger		
	Create invisible ifram		

در مسیر src/utils.py/ مطابق با هر کتابخانه‌ی دلخواهی رونویسی گردد. کد این ابزار به‌صورت نرم‌افزار آزاد و در سایت Github انتشاریافته است [۱۲].

جهت ارزیابی ابزار پیشنهادی، از سرویس‌های آسیب‌پذیر bWAPP و Mutillidae که توسط بنیاد OWASP طراحی و معرفی شده، استفاده شده است. این سرویس‌ها بیش از صدها آسیب طبقه‌بندی‌شده دارند که می‌توان دسته‌های مختلف مورد تمرین و آزمایش قرار گیرد. یکی دیگر از ابزارهای استفاده شده BeEF است. BeEF یک چارچوب تست نفوذ است که بر روی محیط مرورگر تمرکز دارد. این ابزار در واقع یک چارچوب برای پیاده‌سازی و اجرای حملات بر روی مرورگرها است. به‌طورکلی ابزار BeEF برای اجرای حملات نوع تزریق، حملات فیشینگ، کلیک دزدی و آسیب‌پذیری مرورگرها ساخته شده است.

ارزیابی انجام شده بر اساس رویکرد GQM طراحی شده و هدف آن بررسی کارایی ابزار پیشنهادی است. GQM، رویکردی برای بهبود و اندازه‌گیری کیفیت نرم‌افزار است که با ارائه یک مدل اندازه‌گیری در سه سطح مفهومی، عملیاتی و کیفی تعریف می‌شود. از این رو سعی شده به این سؤال پاسخ داده شود که آیا ابزار می‌تواند حملات مختلفی که یک بدافزار MitB اجرا می‌کند را در بازه زمانی مناسب و به طور صحیح شناسایی کند؟ معیارها در این ارزیابی موارد زیر است:

- دقت [۱] و بازخوانی [۲] قابل‌قبول
- زمان شناسایی قابل‌قبول در شناسایی حملات
- مقایسه دقت و بازخوانی ابزار با سه ابزار Check iframe، tackker و Pixm در شناسایی حملات

در جدول ۲ فهرست تمامی حملاتی که در این ارزیابی بررسی شد، ذکر شده است.

برای اجرای آزمایش، ابتدا بر روی یک سرور Linux Ubuntu 19.04،

[1] precision [2] recall

جدول ۳. مقایسه نتایج ابزارهای مشابه با نتیجه ابزار پیشنهادی

	ابزار پیشنهادی	Check iframe	tackker	Pixm
بازخوانی	۱۰۰٪	۱۰۰٪	۱۰۰٪	۶۶٪
دقت	۸۰٪	۷۷٪	۴۴٪	۱۰۰٪

جدول ۴. بازه زمانی وقوع تا تشخیص حملات در ابزار پیشنهادی

Mutillidae	زمان اجرای حمله t1	زمان تشخیص حمله t2	اختلاف زمان t2–t1
HTML Injection	00:22:43	00:23:03	00:00:20
JavaScript Injection (Password Generator)	00:25:33	00:26:01	00:00:28
JavaScript Injection (Discussion of Back Button)	00:18:56	00:19:28	00:00:32
Sensitive data expose	00:20:23	00:20:01	00:00:22

ابزار BeEF، bWAPP و Mutillidae مطابق دستورالعمل نصبی که سازنده ابزارها معرفی کرده‌اند، نصب شد. پس از آن در یک سیستم دیگر و بر روی مرورگر Firefox، افزونه ابزار پیشنهادی نصب گردیده و توصیف الگوی هر یک از حملات مطرح شده در مخزن ابزار قرار گرفت. سپس ۱۰۰ سایت برتر الکسا مشاهده شد و پس از آن به‌طور تصادفی به اجرای حملات توسط چارچوب BeEF و استفاده از این چارچوب در یک افزونه بدافزارگونه برای اجرا روی سرویس‌های bWAPP و Mutillidae پرداخته شد. در نهایت مقدار دقت و بازخوانی ابزار محاسبه و با سه ابزار Check iframe، tackker و Pixm مقایسه شد. نتایج این ارزیابی در جدول ۳ ملاحظه می‌شود. همان‌طور که مشخص است، ابزار پیشنهادی توانسته نتایج بهتری را کسب کند.

در هنگام اجرای حملات بر روی Mutillidae، زمان شناسایی حملات توسط ابزار اندازه‌گیری شد. همان‌طور که در جدول ۴ مشاهده می‌شود، ابزار تهیه‌شده توانست حملات اجراشده بر روی Mutillidae را در بازه زمانی کمتر از نیم ثانیه پس از وقوع شناسایی کند که نتیجه قابل قبولی است.

۵ جمع‌بندی

استفاده روزمره کاربران از مرورگر برای اعمالی نظیر خرید برخط، انتقال وجه، ارسال اطلاعات به سامانه‌های درون شبکه و مناسب بودن محیط مرورگر برای اجرای حملات، باعث شده تا تشخیص حملات MitB یک چالش جدی برای حفاظت از اطلاعات کاربران، محسوب شود. انواع مختلف حملات MitB به سرعت در حال افزایش است. از این رو وجود یک روش کارا و جامع برای تشخیص این خانواده از حملات، همواره یک

چالش جدی محسوب می‌شود؛ بنابراین در این پژوهش به ارائه ابزاری برای شناسایی حملات MitB با استفاده از تحلیل پویای صفحات وب پرداخته شد که با استفاده از توصیف الگوی حملات ارائه‌شده در ابزار، به تحلیل پویای صفحات وب برای شناسایی الگوی حمله پرداخته و در صورت تشخیص حمله، هشدار لازم را به کاربر اعلام می‌کند. از ویژگی‌های این ابزار، دقت و بازخوانی بالا، محدود نبودن به شناسایی نوع خاصی از حملات، عدم تعبیه کد روش شناسایی در ابزار و توانایی تشخیص حملات جدید بر اساس توصیف الگوی آن، است.

مراجع

[1] Rauti, Sampsa and Leppänen, Ville. Man-in-the-browser attacks in modern web browsers. in *Emerging Trends in ICT Security*, pp. 469–480. Elsevier, 2014.

[2] Eisen, Ori. Catching the fraudulent man-in-the-middle and man-in-the-browser. *Network Security*, 2010(4):11–12, 2010.

[3] Goyal, Puneet, Bansal, Naman, and Gupta, Neeraj. Averting man in the browser attack using user-specific personal images. in *2013 3rd IEEE International Advance Computing Conference (IACC)*, pp. 1283–1286. IEEE, 2013.

[4] Cain, Chris. Analyzing man-in-the-browser (mitb) attacks. *dated Dec*, 2014.

[5] Utakrit, Nattakant. Review of browser extensions, a man-in-the-browser phishing techniques targeting bank customers. 2009.

[6] Pawade, Dipti, Reja, Divya, Lahigude, Abhilasha, and Johri, Era. Implementation of extension for browser to detect vulnerable elements on web pages and avoid clickjacking. in *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*, pp. 226–230. IEEE, 2016.

[7] Seshagiri, Prabhu, Vazhayil, Anu, and Sriram, Padmamala. Ama: static code analysis of web page for the detection of malicious scripts. *Procedia Computer Science*, 93:768–773, 2016.

[8] Zhang, Hao, Yao, Danfeng Daphne, Ramakrishnan, Naren, and Zhang, Zhibin. Causality reasoning about network events for detecting stealthy malware activities. *computers & security*, 58:180–198, 2016.

[9] Pawade, Dipti, Reja, Divya, Lahigude, Abhilasha, and Johri, Era. Implementation of extension for browser to detect vulnerable elements on web pages and avoid click-jacking. in *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*, pp. 226–230. IEEE, 2016.

[۱۰] رحیمی حسین‌آبادی، علی. روشی جهت تشخیص بدافزارهای پنهان کار با استفاده از تحلیل رفتاری، ۱۳۹۸.

[11] Owasp, man-in-the-browser attack. https://owasp.org/www-community/attacks/Man-in-the-browser_attac. Accessed: 2019.

[12] [online]. https://github.com/majidmc2/Dobiel. Accessed: 2020.

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2022, Volume 20, Number 1 (pp. 59–66)
https://monadi.isc.org.ir

Selected paper at the ISCISC 2021 in University of Isfahan, Isfahan, Iran

A Tool for Detecting Man in the Browser (MitB) Attacks using Dynamic Analysis of Web Pages★

Majid Iranpour Mobarakeh *and Behrouz Tork Ladani

Model Based Security Analysis (MBSA) Research Group, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran

ARTICLE INFO.

Keywords:
MitB attacks
malware
malware detection
dynamic analysis

dor: 20.1001.1.24763047.1401.11.1.7.4

Type: research paper

ABSTRACT

Detection of browser attacks is considered a serious challenge in today's web applications. Man in the Browser (MitB) attack is an important type of these attacks that can lead to changes in web page contents, interference in network traffic, session hijacking, and user information theft by using Trojans. In this paper, an efficient tool for real-time detection of MitB attacks through dynamic analysis of web pages based on the description of attack patterns is presented. The advantage of the proposed tool is that it is not limited to identifying one or more specific attacks and the identification method code is not embedded in the tool, but the patterns of different attacks are specified separately. In order to evaluate the presented tool, two vulnerable web services provided by OWASP, which have a wide range of known vulnerabilities, were used along with the BeEF penetration test framework, and a set of MitB attacks were practically implemented and evaluated by the tool. The same tests were performed using three other similar tools and compared with the developed tool. In addition to the superiority of the presented tool in terms of the independence of attack descriptions from the tool itself, the results show that the accuracy and readability of its diagnosis are better than similar tools.

© 2022 ISC

★ The ISCISC 2021 Program Committee effort is highly acknowledged for reviewing this paper.
* Corresponding author
Email addresses: majidiranpour@eng.ui.ac.ir (Majid Iranpour Mobarakeh), ladani@eng.ui.ac.ir (Behrouz Tork Ladani)
© 2022 ISC. All rights reserved.