تشخیص PDF های مخرب با استفاده از قوانین رابطه‌ای

حبیب‌رضا غلامی[1]، اسماعیل زینالی‌خسرقی[2]، منصور احمدی[3]

۱. کارشناس ارشد مهندسی صنایع، دانشگاه آزاد اسلامی واحد قزوین، hr_gholami@qiau.ac.ir

۲. استادیار کامپیوتر، دانشگاه آزاد اسلامی واحد قزوین

۳. کارشناس ارشد کامپیوتر، دانشگاه آزاد اسلامی واحد قزوین

تاریخ دریافت: ۹۲/۱۰/۲۳ تاریخ پذیرش: ۹۳/۵/۱

چکیده

فایل‌های PDF مخرب طی ۲-۳ سال اخیر به منظور زیان رساندن به امنیت کامپیوتر بیشتر مورد استفاده قرار گرفته‌اند و بسیاری از ضدویروس‌های اخیر در مقابل این نوع تهدید ناکارآمد هستند. در این پژوهش روشی جدید به منظور تشخیص این نوع حملات ارائه شده است که در آن خصیصه‌های مربوط به ساختار فایل‌های PDF به صورت ایستا استخراج شده و با استفاده از استخراج الگوهای رایج از آنها و در نظر گرفتن الگوها به عنوان خصیصه، یک سیستم طبقه‌بند موثر ایجاد شده است. این سیستم به جای تحلیل جاوااسکریپت واقع در فایل‌های PDF، از ساختار آن استفاده می‌کند و نشان داده می‌شود که ساختار می‌تواند به طور موثر فایل‌های مخرب و سالم را از یکدیگر متمایز کند. فایل‌هایی که مورد تحلیل واقع شده است در حدود ۶۰۰۰ PDF مخرب و ۶۰۰۰ PDF خوش‌خیم هستند. این سیستم از دیگر پژوهش‌هایی که در این زمینه انجام شده است و همچنین از بقیه ضدویروس‌ها، نتایج بهتری را کسب کرده است. علاوه بر آن، می‌توان از این روش بعنوان افزونه‌ای برای ضدویروس‌های رایج استفاده کرد. همچنین این روش دربرابر فرار از تشخیص به صورت بسیار کارآمدی عمل می‌کند.

کلیدواژه

PDF، مخرب، داده کاوی، تشخیص، Association Rule

مقدمه

روش‌هایی که در آن هکرها سعی می‌کنند در امنیت سیستم‌های کامپیوتری اختلال ایجاد کنند روز به روز در حال توسعه است. سیستم عامل‌ها روز به روز امن‌تر می‌شوند و وصله‌های امنیتی مرتبا ارائه می‌شوند و امکان یافتن آسیب‌پذیری‌های روز-صفر[1] در سیستم عامل‌ها کاسته می‌شود. بنابراین مهاجمان، برنامه‌های شرکای سوم[2] (از قبیل Adobe Reader، Microsoft Outlook و ...) و فایل‌های با فرمت آنها را مورد هدف قرار می‌دهند. فایل‌های PDF امروزه به طور گسترده به منظور خواندن متون مورد استفاده قرار گرفته و بسیار مرسوم هستند. کاربران تصور می‌کنند که آنها امن هستند هرچند که امنیت آنها در سال‌هال اخیر بیشتر مورد تهدید واقع شده است. برنامه‌هایی که به طور مداوم به منظور بازکردن آنها استفاده می‌شوند، توسط مهاجمان مورد حمله قرار می‌گیرند و از آنها به منظور دسترسی به سیستم استفاده می‌کنند. علاوه بر آن، طیف حملات از زمانی که افزونه‌ها برای آن زیاد شده است نیز افزایش یافته است. زمانی که سیستم توسط مهاجم مورد حمله قرار گرفت، آن سیستم را بعنوان بخشی از شبکه ربات‌ها قرار می‌دهد. این مشکل توسط شرکت های Symantec و IBM در سال‌های ۲۰۰۹ و۲۰۱۰ گزارش شد[۱،۲]. مهاجمان مرتبا باهوش‌تر می‌شوند و اقدامات امنیتی شرکت Adobe را دور می‌زنند و همچنان به عنوان یک تهدید واقعی باقی مانده‌اند. واضح بودن فرمت PDF مهاجمان را قادر می‌سازد تا با کمترین تلاش بتوانند از شناخته شدن فرار کنند. حملات زیادی اخیرا با استفاده از آسیب پذیری‌های اخیر شرکت Adobe انجام شده است[۳،۵]. این حملات بر خلاف حملات قدیمی‌تر تاثیر زیادی را بر روی کابران معمولی کامپیوتر گذاشته است. علاوه بر آن، آسیب‌پذیری‌های زیادی اخیرا در برنامه‌های Adobe Reader کشف شده است[۶] که مکانیزم‌های دفاعی موجود برعلیه آن حملات هنوز ناکافی است و از این رو قابلیت فرار PDF مخرب از آنها وجود دارد. حتی اکثر ضدبدافزارهای جدید تنها حملات خاص PDF را تشخیص می‌دهند زیرا روش آنها مبتنی بر امضا است. بنابراین آنها نمی‌توانند با حملات جدیدی که تغییری جزئی بر روی آنها بوجود می‌آورند، مقابله کنند. پژوهش‌های کمی در زمینه تشخیص PDFهای مخرب انجام شده است که در بخش کارهای مرتبط به تفسیل به آن پرداخته می‌شود. بر خلاف کارهای قبلی، روش ارائه شده در این پژوهش قصد بر تشخیص ایستای PDFهای مخرب را

1. Zero Day
2. Third Party

دارد بدون اینکه بخواهد توجهی به محتوای جاوااسکریپت آن شود. اگر چه تعداد زیادی از روش‌های سوء استفاده از PDFها، متکی به جاوااسکریپت هستند، دو چالش مهم با این قضیه وجود دارد که تشخیص مبتنی بر جاوااسکریپت را مشکل می‌سازد. اولین آن یافتن جاوااسکریپت در فایل PDF است زیرا ممکن است در ساختار منطقی PDF پنهان شود حتی فراتر از مکان‌هایی که در استاندارد PDF وجود دارد[۷]. هر محتوای متنی در فایل PDF می‌تواند با استفاده از تابع eval و یا مشابه آن بعنوان جاوااسکریپت تفسیر شود. از این رو، هیچ چیز نمی‌تواند یک مهاجم را از انتشار کدهای جاوااسکریپت در متن باز دارد. دومین چالش درهم‌سازی کدهای جاوااسکریپت است. اگر چه روش‌هایی به منظور تشخیص کدهای درهم جاوااسکرپت ارائه شده است[۸] اما هنوز عملی‌بودن این روش‌ها مبهم است. در این مقاله یک روش جایگزین برای روش‌های تشخیص مبتنی بر جاوااسکریپت ارائه شده است که از خصوصیات مستندات PDF به منظور متمایز کردن فایل‌های سالم و مخرب استفاده می‌کند. به جای اینکه به محتوی مخرب توجه شود، به ساختار مخرب توجه می‌شود و دلیل آن پیچیدگی ساختار و ساختار منطقی مستندات PDF است. ایجاد سیستمی که در برابر مهاجمان مقاوم باشد نیز یک چالش است. در این پژوهش، ابزار جدیدی به منظور تشخیص PDFهای مخرب به صورت ایستا با استفاده از داده کاوی ارائه می‌شود. به دلیل کارایی آن از لحاظ سرعت، می‌توان این روش را بعنوان یک تشخیص‌دهنده جاسازی شده درون هر آنتی‌ویروسی و یا ایجاد کننده PDF قرار داد. کارایی روش پیشنهادی بر روی داده‌های زیادی در حدود ۱۲۰۰۰ فایل PDF مخرب و سالم ارزیابی شد که نرخ تشخیص بالای ۹۹٪ بدست آمد. این نتایج نشان دهنده قدرت تشخیص بسیار خوب روش پیشنهادی در برابر PDF مخرب حاضر در دنیای واقعی است. علیرغم اینکه این روش محتوی را لحاظ نمی‌کند، فرار از آن به طور بدیهی امکان‌پذیر نیست. حتی اگر مهاجم بداند چه خصیصه‌هایی برای تشخیص لازم است، نمی‌تواند آنها را به سادگی حذف کند زیرا حذف خصیصه‌ها ممکن است که در حمله، اختلال ایجاد کند. علاوه بر این، اضافه‌کردن خصیصه‌های فایل‌های سالم نمی‌تواند منجر به فرار شود. این نوع حملات را با چندین اعمال آزمایشگاهی بررسی کردیم و نشان داده می‌شود که سیستم در مقابل این حملات مقاوم است.

در مجموع، سهم این پژوهش در زیر ارائه شده است:

۱. ارائه قوانین رابطه‌ای به عنوان مجموعه‌ای از خصیصه‌های جدید به منظور تمایز میان فایل‌های PDF سالم و مخرب که در هیچ کدام از کارهای قبلی استفاده نشده است.

۲. بهتر بودن نتایج بدست آمده (از لحاظ سرعت) در مقایسه با کارهای مرتبط و بدست آوردن دقت بالای ۹۹٪

۳. ارائه یک طبقه‌بند جدید که وابسته به محتوی فایل‌ها نیست

۴. مقاوم بودن در برابر روش‌های فرار از تشخیص

کارهای مرتبط

مستندات مخرب در سالیان اخیر موضوع پژوهش‌های زیادی بوده‌اند. در تحلیل ایستا، مقاله‌های[۹، ۱۰] با استفاده از نمایش n-gram داده‌های فایل‌های مستند ورد، نمونه مخرب آن را تشخیص داده‌اند. تحلیل پویا نیز نسبت به اندازه کد مخرب، محدودیت‌هایی را نیز به همراه دارد. در حالی که تکنیک آنها و نوع فرمت فایل با کار ما تفاوت دارد، اما هدف‌ها بسیار شبیه به هم هستند. علاوه بر آنها، تحلیل امضاء نیز به طور گسترده[۱۱، ۱۲] برای آسیب‌پذیری‌های فایل‌های PDF مورد مطالعه قرار گرفته شده است[۱۳].

به منظور بررسی انتشار بدافزار از طریق مستندات PDF، کارهای بعدی بر روی تحلیل کدهای جاوااسکریپت تمرکز داشته‌اند. راه‌حل‌های زیادی برای تشخیص کدهای جاوااسکریپت ارائه شده است. برای مثال Jsand[۱۴]، Cujo[۱۵]، Zozzle[۱۶]، Prophiler[۱۷] ابزارهای مشهور برای تحلیل ایستا و پویا در این زمینه هستند. این ابزارها اغلب به منظور تشخیص تهدیدهای جاسازی‌شده در مستندات به کار برده می‌شوند.

Wepawet[۱۸]، یک چارچوب برای تحلیل تهدیدهای تحت وب است که با استفاده از JSand کدهای جاوااسکریپت درون فایل‌های PDF را تحلیل می‌کند. JSand با استفاده از مطابقت با HtmlUnit[3]، یک شبیه ساز مرورگر مبتنی بر جاوا و Mozilla's Rhino[4] به منظور استخراج خصیصه‌های رفتاری مرتبط با اجرای کدهای جاوااسکریپت کار می‌کند که در آن از یک طبقه‌بندی آماری برای تشخیص الگوهای ناهنجار استفاده شده است.

یک رویکرد مشابه MalOffice[۱۹] است. این رویکرد با استفاده از pdftk[5] کدهای جاوااسکریست را استخراج می‌کند و CWSandbox[۲۰] رفتار کد را تحلیل می‌کند و طبقه‌بندی در آن با استفاده از یک سری قوانین انجام می‌شود. CWSandbox همچنین برای طبقه‌بندی رفتار بدافزارها استفاده می‌شود[۲۱]. MDScan[۲۲] یک رویکرد متفاوت را دنبال می‌کند و رفتارهای مخرب را از طریق Nemu، که ابزاری برای تفسیر کدهای تزریق شده در حافظه است، تشخیص می‌دهد. همچنین یک ایده مشابه با پیاده‌سازی متفاوت در ShellOS[۲۳] ارائه شده است.

در تحلیل پویا، مقاله[۲۴] از برنامه reader ابزاری استفاده کرده و با استفاده از تحلیل پویا خصیصه‌های ساختاری مستندات PDF را استخراج کرده و به عنوان خصیصه‌های طبقه‌بند در یادگیری

3. http://htmlunit.sourceforge.net
4. http://www.mozilla.org/rhino/
5. http://www.pdflabs.com/tools/pdftk-the-pdftoolkit

ماشین استفاده می‌کند. در حالی که رویکرد بالا مشابه چیزی است که در اینجا ارائه می‌شود با این تفاوت که آنها درصد تشخیص بالایی را بدست نیاورده‌اند. Laskov کدهای جاوااسکریپت درون PDFها را استخراج کرده و از آنها در طبقه بند SVM استفاده کرده‌اند که درصد نسبتا متوسطی را بدست آورده‌اند[۲۵]. در هر دو روش، عامل مهم توانایی، parse کردن فایل‌های مستند مخرب است که کار مشکلی است. در این مقاله با استفاده از روش‌های مختلف استخراج خصیصه، عمل parse کردن تسهیل شده است. مقاله[۲۲] نشان داده است که عملکرد آنتی ویروس‌های موجود در برابر فایل‌های مخرب نسبتا کم است. به منظور افزایش نرخ تشخیص، آنها از ترکیب روش‌های ایستا و پویا استفاده کرده‌اند. این روش نیازی به اجرا ندارد و همچنین مستقل از آسیب‌پذیری‌ها است.

جدیدترین کار در این زمینه Malware Slayer[۲۶] و PDFRate[۲۷، ۲۸] هستند که با ارائه راهکار یادگیری ماشین بر روی ساختار PDF ارائه شده‌اند. این دو روش کدهای مخرب را تحلیل نمی‌کند بلکه با در نظر گرفتن الگوهای متفاوت در ساختار فایل PDF، PDF مخرب و سالم را از یکدیگر مجزا می‌کنند. Malware Slayer بر روی nameها object تمرکز دارد در حالی که PDFRate هر چقدر که می‌تواند اطلاعات مختلفی را از ساختار PDF استخراج می‌کند از قبیل تعداد objectها، streamها، حروف کوچک و بزرگ و غیره. هر دو ابزار نرخ تشخیص بالا و false positive کمی را در تشخیص PDFهای مخرب نشان داده‌اند. یک رویکرد مشابه دیگر[۲۹] با استفاده از ساختار سلسله مراتبی name objectها نیز ارائه شده است. در این مقاله نیز از ساختار PDF استفاده شده است با این تفاوت که مرحله استخراج خصیصه با تمامی روش‌های قبلی متفاوت است. در بخش نتایج نشان داده می‌شود که از لحاظ دقت نتیجه‌ای مشابه و از لحاظ سرعت و فرار از تشخیص بهتر از روش‌های قبلی عمل شده است. ابزارهای متفاوتی نیز برای تحلیل مستندات PDF ارائه شده‌اند و همگی آنها اطلاعات مختلفی را استخراج می‌کنند و بر روی تشخیص PDFهای مخرب تاکیدی ندارند. از جمله این ابزارها می‌توان به PDF Tools[۳۰]، PeePDF[۳۱] و Origami[۳۲] اشاره کرد.

ساختار PDF

PDF یک ساختار سلسله مراتبی از objectها دارد که به یکدیگر متصل شده‌اند. ما در اینجا ساختار فایل PDF را به ۴ بخش اصلی تقسیم‌بندی می‌کنیم[۱۷]: objectها، ساختار فایل، ساختار مستند و Streamهای محتوایی.

Objectها

Objectها به دو بخش objectهای مستقیم، یعنی آنهایی که بوسیله یک شماره ارجاع داده می‌شوند (و آنهایی که بوسیله reader به منظور ساختن ساختار منطقی استفاده می‌شوند) و objectهای غیرمستقیم، یعنی آنهایی که بوسیله شماره ارجاع داده نمی‌شوند، تقسیم‌بندی می‌شوند. اساسا objectهای PDF، ۸ نوع هستند: ۱- Boolean objectهایی که مقادیر آنها True یا False است. ۲- Numeric objectهایی که با یک مقدار عددی صحیح نمایش داده می‌شوند. دو نوع شی عددی وجود دارد که به صورت صحیح و حقیقی است. همچنین می‌توانند مثبت یا منفی باشند. ۳- String یک دنباله از کاراکترها است در پرانتز ( ) و یا داده‌های hexadecimal در براکت‌های گوشه‌ای < >. ۴- Name دنباله‌ای از کاراکترها است که با / شروع می‌شوند. ۵- Array دنباله‌ای از اشیاء که بین براکت‌های مربعی [ ] قرار می‌گیرد. مثال:

[12 false (Literal String) /NameObject]

۶- Dictionary دنباله‌ای از جفت‌های یک کلمه کلیدی (name object) و یک مقدار (Boolean، numeric، کلمه کلیدی و یا یک آرایه دیگر) را می‌سازد. آنها بین <<>> قرار می‌گیرند. مثال:

<< /Type /Example /Version 0.03 /String (Literal String) /Array [12 false (Literal String) /NameObject] >>

۷- Stream یک شی dictionary خاص بین کلمات کلیدی stream و endstream قرار می‌گیرد که برای ذخیره عکس، متن، کد اسکریپت استفاده می‌شود و می‌تواند با استفاده از فیلترهای خاص محصور شود. مثال:

<< /Length 43 >> stream BT /F1 24 Tf 100 700 Td (Hello World) Tj ET endstream

۸- یک شی خالی که یا کلمه کلیدی null نمایش داده می‌شود.

ساختار فایل

ساختار فایل تعیین‌کننده این است که چگونه اشیاء درون فایل PDF، قابل دسترس و قابل بروزرسانی هستند. هر فایل PDF از ۴ قسمت تشکیل شده است.

شکل ۱. ساختار داخلی فایل PDF

۱- Header اولین خط از مستند است که اطلاعاتی درباره ورژن PDF استفاده شده بوسیله فایل را نشان می‌دهد. فرمت header به صورت "PDF-X.X%" است که کاراکترهای X با شماره ورژن استاندارد PDF جایگزین می‌شوند. ۲- Body بخش اصلی فایل است و شامل تمامی objectهای PDF است. ۳- Cross-reference table این بخش دقیقا در زیر body واقع شده است و نشان‌دهنده موقعیت هر object غیرمستقیم در حافظه است. هدف این بخش دسترسی تصادفی به اشیاء است و بنابراین در کارایی نرم‌افزارهای خواندن PDF تاثیر می‌گذارد. با استفاده از این جدول، یک PDF reader نیازی ندارد که تمامی فایل را به منظور یافتن یک شی خاص جستجو کند. ۴- Trailer اطلاعاتی درباره جدول تداخل و نیز object کلیدی همانند ریشه و همچنین تعداد revisionهایی که برای مستند ساخته شده است می‌دهد. آخرین خط فایل PDF با "EOF%%" نشان داده می‌شود.

ساختار مستند

ساختار مستند نشان می‌دهد که اشیاء چگونه برای نمایش قسمت‌های مختلف مستند PDF استفاده می‌شوند از قبیل صفحه‌ها، فونت، انیمیشن‌ها و غیره. سلسله مراتب اشیاء در body فایل PDF تشریح می‌شود. شی اصلی در این سلسله مراتب، شی catalog است که با یک dictionary نشان داده می‌شود. اغلب اشیاء غیرمستقیم در فایل PDF، dictionaryها هستند. هر صفحه مستند یک شی page است که شامل ارجاع‌هایی به اشیاء دیگر که جزئی از آن صفحه هستند، است. Catalog dictionary بوسیله root/ در قسمت trailer علامت‌گذاری شده است.

Content Streamها

اشیاء stream شامل دنباله‌ای از دستورالعمل‌ها است که ظاهر صفحه و موجودیت گرافیکی آن را تشریح می‌کنند. اگرچه آنها بعنوان objectها تعریف می‌شوند، با objectهایی که ساختار مستند را نشان می‌دهند متفاوت است. دستورالعمل‌ها نیز می‌توانند به دیگر objectهای غیرمستقیم ارجاع داده شوند که شامل اطلاعاتی درباره منابع مطابق با آن stream است. ساختار منطقی می‌تواند پیچیده باشد زیرا تعدادی درجه آزادی در تشکیل ارجاع بین objectها وجود دارند. علاوه بر آن، به استثناء فایلهای خطی، ترتیب objectها درون فایل کاملا اختیاری است. معمولا غیرممکن است که objectهای درون فایل را ویرایش کرد زیرا آنها ارجاع به حافظه خود دارند و درون جدول Cross-Reference آورده شده است. به منظور انجام این کار، یک نسخه جدید از object باید ایجاد شود و بعد از trailer، با یک trailer جدید و یک جدول Cross-Reference جدید اضافه می‌شود. یعنی objectهای اصلی درون فایل نگه‌داری می‌شود. این روال نیز version update نامیده می‌شود.

داده کاوی

داده‌کاوی بعنوان استخراج‌کننده غیربدیهی، شناخته‌نشده و بالقوه اطلاعات از داده‌های با مقادیر زیاد [۳۳] شناخته می‌شود. داده‌کاوی به دو دسته پهناور تقسیم‌بندی می‌شود. پیش‌بینی، که شامل پیش‌بینی مقادیر ناشناخته با استفاده از اطلاعات داده شده است و توصیفی، که شامل پیداکردن الگوها جهت توصیف داده‌ها هستند. در حالت کلی، الگوریتم‌های داده‌کاوی به ۳ دسته کاوش الگوهای رایج، طبقه‌بندی و خوشه‌بندی تقسیم می‌شوند. از آنجا که در این پژوهش از الگوهای رایج و طبقه‌بندی استفاده شده است، این دو را را به صورت خلاصه بیان می‌کنیم.

کاوش الگوهای رایج

کاوش الگوهای رایج به معنای یافتن الگوهایی[۳۴] است که زیاد تکرار می‌شوند. الگوهای رایج به عنوان نام پیشنهاد شده برای الگوهایی می‌باشد که بطور رایج و زیاد در داده‌ها وجود دارند. انواع مختلفی از الگوهای رایج شامل اقلام، زیردنباله‌ها و زیرساختارها می‌باشد. اقلام رایج معمولا اشاره به مجموعه‌ای از عناصر دارد که با یکدیگر در تراکنش‌های پایگاه اطلاعاتی تکرار می‌شود، برای مثال شیر و نان که ترتیب اهمیتی ندارد. زیردنباله رایج از قبیل الگویی که مشتری‌ها تمایل به خرید آن دارند بعنوان مثال اول کامپیوتر و سپس دوربین دیجیتال و سپس یک کارت حافظه می‌خرند که این یک الگوی ترتیبی رایج است. زیر-ساختار می‌تواند اشاره به شکل‌های ساختاری متفاوت داشته باشد از قبیل گراف‌ها، درخت‌ها و شبکه‌ها. اگر یک زیر-ساختار به طور متداول رخ دهد به آن الگوی ساختاری رایج گویند.

طبقه‌بندی

طبقه‌بندی به معنی پیش‌بینی برچسب‌ها[۳۲] برای داده‌ها بر اساس داده‌های برچسب خورده قبلی است. طبقه‌بندی فرایندی است برای یافتن مدل (یا تابع) که داده‌ها را تشریح و کلاس‌های آنها را تشخیص می‌دهد. برای رسیدن به این هدف از مدل ساخته شده استفاده می‌شود که می‌تواند برچسب کلاس‌های نامشخص را بدست آورد. مدل بیان شده براساس تحلیل مجموعه‌ای از داده‌های آموزشی (داده‌هایی که برچسب کلاس آنها مشخص می‌باشد) بدست می‌آید.

روش کار

در این بخش روش پیشنهادی برای تشخیص فایل‌های PDF مخرب ارائه می‌شود. فرایند داده‌کاوی را در ۵ مرحله بیان مسئله، جمع‌آوری داده، پیش‌پردازش داده، تخمین مدل و تشریح مدل می‌توان تشریح کرد. همانطور که در اینجا مشخص است مسئله، تشخیص مستندات PDF مخرب به کمک طبقه‌بندی است. همانطور که در کارهای مرتبط مشاهده شد، قسمت مهم در مسئله تشخیص این فایل‌ها مرحله استخراج خصیصه است. نکته اصلی در این پژوهش استخراج الگوهای رایج از اشیاء نام درون فایل‌های PDF است که در هیچ کدام از کارهای قبلی انجام نشده است و همچنین این نوع الگوها می‌توانند باعث تمایز بین فایل‌های خوش‌خیم و مخرب شوند.

یادگیری ماشین

مسئله تشخیص یک PDF بعنوان مخرب و یا خوش‌خیم بعنوان یک مسئله طبقه‌بندی معرفی می‌شود. نظریه یادگیری چنین مسائلی را تشریح می‌کند که فرایند یادگیری از داده به ۲ مرحله تقسیم می‌شود. ۱- ساختن مدل از روی نمونه داده‌های ورودی ۲- پیش‌بینی داده‌های جدید با استفاده از مدل بدست آمده.

مدل‌ها

در این مقاله از سه مدل معروف طبقه‌بندی برای انجام آزمایش‌ها استفاده شده است. در اینجا هر کدام بصورت مختصر تشریح می‌شود.

درخت تصمیم: یک درخت تصمیم فضای پیشگویی مدل را به صورت بازگشتی قسمت‌بندی کرده تا رابطه بین متغیرهای آن را مدل کند. با استفاده از نمونه‌ها درختی ساخته می‌شود. سیستم یادگیر یک رویکرد بالا به پایین است. با استفاده از پیمایش درخت حاصل شده، مجموعه‌ای از قوانین بدست می‌آید که به کمک آن می‌توانید برچسب کلاس هر کدام از نمونه‌ها را پیش‌بینی کنید.

ماشین بردار پشتیبان: ماشین‌های بردار پشتیبان مجموعه‌ای از ابزارها هستند که می‌تواند بهینه‌ترین ابرصفحه را برای جداسازی خطی و یا غیرخطی داده‌ها به ۲ دسته[۳۵] انجام دهد. این ابرصفحه فاصله بین دو ابرصفحه موازی را بیشینه می‌کند.

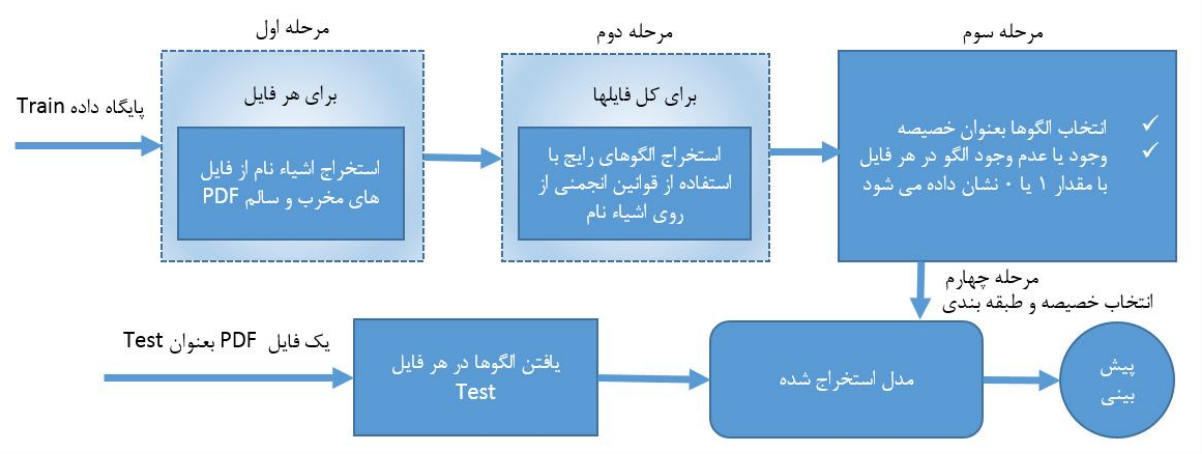
جنگل تصادفی: جنگل تصادفی یک طبقه‌بندی گروهی است که از تعداد زیادی درخت تصمیم تشکیل شده است که خروجی آن مد خروجی تمامی درخت‌های تصمیم به صورت تکی است.

روش پیشنهادی

روش ارائه‌شده در اینجا مبتنی بر رویکرد داده‌کاوی است و برای اولین بار است که از الگوهای تکراری به عنوان یک پارامتر موثر در تشخیص PDFهای مخرب استفاده می‌شود. ساختار کلی روش پیشنهادی در شکل ۲ آورده شده است.

در بخش جمع‌آوری داده، فایل‌های benign و malicious هر دو برای مرحله training استفاده شده‌اند زیرا نقاط مشترکی بین این دو دسته وجود داشت. بنابراین، استفاده از هر دو کلاس برای یادگیری می‌توانست گزینه خوبی باشد. داده‌ها به صورت زیر جمع‌آوری شدند: یک واسط به موتور جستجوی Yahoo[۳۶]، که به صورت تصادفی فایل‌های PDF، با نام‌های تصادفی از یک dictionary، را دانلود می‌نمود و همچنین یک پایگاه‌داده عظیم فراهم شده توسط تیم Contagio[۳۷] (شامل فایل‌های سالم و مخرب) که در اکثر کارهای قبلی از آن استفاده شده است. پیداکردن نمونه‌های مخرب از موتورهای جستجو کار راحتی نیست زیرا اکثر PDFهای مخرب از طریق Spam ارسال می‌شوند. بنابراین، استفاده از موتور جستجو Yahoo نتایج خوبی را نمی‌دهد. تعداد فایل‌ها شامل ۵۹۹۳ PDF مخرب و ۵۹۵۱ PDF خوش‌خیم (در حدود ۱۲۰۰۰) است. فایل‌های مخرب در مجموعه یادگیر شامل تعداد مختلفی از حملات (یعنی کدهای جاسازی شده JavaScript و ActionScript) است. جهت استخراج خصیصه‌های اشیاء Name، از ابزار PDFid استفاده شده است. این ابزار به منظور تحلیل فایل‌های PDF توسط آقای Stevens نوشته شده و هم اکنون جزو نرم‌افزارهای لینوکس Backtrack است. یکی از قابلیت‌های آن استخراج اشیاء Name است. در این مقاله استخراج اشیاء نام برای تک تک PDFها انجام می‌شود. در واقع هر فایل به صورت مجموعه‌ای از نام‌ها ذخیره شد.

در مرحله بعد از انتخاب خصیصه استفاده می‌شود. انتخاب خصیصه فرایندی است که مجموعه‌ای از خصیصه‌های درون مجموعه یادگیر انتخاب می‌شوند. این مجموعه شامل خصیصه‌هایی هستند که میان کلاس‌ها متمایزتر هستند و می‌توانند به بهبود دقت در طبقه‌بندی کمک کنند.



شکل ۲. روش کار پیشنهادی

الگوریتم CfsSubsetEval[۳۸] یکی از این دسته الگوریتم‌ها است که نتایج خوبی را بر روی پایگاه داده‌های مختلف بدست آورده است. این الگوریتم خصیصه‌ها را براساس تابع ارزیابی اکتشافی مبتنی بر وابستگی رتبه‌بندی می‌کند. گرایش به سمتی خواهد بود که خصیصه‌ها تا حد زیادی به آن کلاس وابسته باشند و به دیگر کلاس‌ها وابسته نباشند و در نهایت خصیصه‌های نامرتبط حذف می‌شوند. رتبه‌بندی ذکر شده با استفاده فرمول زیر صورت می‌گیرد:

$$M_s = \frac{k\overline{rcf}}{\sqrt{k + k(k-1)\overline{rff}}} \quad (۱)$$

در فرمول فوق، M_s شایستگی یا همان رتبه‌بندی مجموعه‌ای (S) از k خصیصه را نشان می‌دهد. $\overline{rcf}$ میانگین وابستگی کلاس به خصیصه را نشان می‌دهد که f عضو S است. $\overline{rff}$ میانگین وابستگی داخلی خصیصه به خصیصه را نشان می‌دهد. صورت کسر فوق نشان‌دهنده قدرت پیش‌بینی مجموعه‌ای از خصیصه‌ها برای یک کلاس است و مخرج نشان‌دهنده زائد بودن مجموعه‌ای از خصیصه‌هاست. این الگوریتم برای تمامی حالتهای مختلف خصیصه‌های این مقدار را می‌سازد و بهترین را انتخاب می‌کند. این الگوریتم در نرم افزار WEKA[۳۹] پیاده‌سازی شده است.

در مرحله بعد از الگوهای رایج همانند Apriori استفاده می‌شود. در جستجوی اولیه، تعداد اقلام تکی مشخص می‌شوند. تمامی اقلامی که رایج نیستند (یعنی اقلامی که در تراکنش‌های کمتر از تعداد تعیین‌شده توسط کاربر ظاهر می‌شوند) از تراکنش‌ها حذف می‌شوند زیرا این اقلام هرگز جزو مجموعه اقلام تکراری دیگر نخواهند بود.

در نهایت از طبقه‌بندی برای تشخیص استفاده می‌شود. طبقه‌بندی مجموعه الگوهای تکراری که از مرحله قبلی بدست آمد را به عنوان خصیصه انتخاب می‌کند. همانطور که قبلا عنوان شد، استفاده از الگوهای تکراری برای طبقه‌بندی می‌تواند نتایج خوبی را حاصل کند. در این حالت ۱۸۲ خصیصه بدست آمده به منظور طبقه‌بندی استفاده می‌شوند. به منظور اینکه تعداد خصیصه‌ها متمایزتر باشند، از الگوریتم انتخاب خصیصه CfsSubsetEval استفاده می‌شود که تعداد خصیصه‌ها از ۱۸۲ به ۷ تقلیل یافت. تعداد خصیصه کمتر می‌تواند سرعت را در زمان آزمون افزایش دهد. این ۷ خصیصه از آن جهت موثر هستند که هر کدام یک الگو هستند و از چندین اشیاء نام تشکیل شده‌اند که علاوه بر سرعت، فرار از تشخیص را برای مهاجم سخت‌تر می‌کنند.

نتایج و آزمایش‌ها

در این بخش، مجموعه آزمایش‌هایی را که بر روی پایگاه اطلاعاتی انجام گرفته بیان می‌شود و نتایج بدست آمده را تفسیر خواهیم کرد.

نتایج

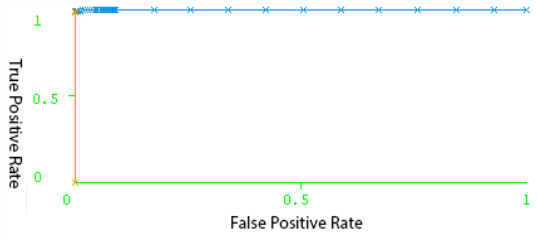
جهت بررسی کارایی روش ارائه شده لازم است که با یک سری نتایج آزمایشگاهی، قدرت آن نشان داده شود. به همین منظور از سه معیار مختلف دقت، سرعت و مقاوم بودن در برابر فرار از روش استفاده می‌شود که هر کدام را در زیر توضیح داده‌ایم. دقت روش پیشنهادی کاملا بررسی شده و نتایج بسیار خوبی حاصل شده است. همچنین از چندین طبقه‌بند برای ساختن مدل استفاده می‌شود که بهترین نتایج مربوط به جنگل تصادفی است. جهت ارزیابی روش پیشنهادی، از روش اعتبارسنجی صلیبی ۱۰تایی استفاده کرده‌ایم. پایگاه اطلاعاتی ما شامل ۵۹۹۳ PDF مخرب و ۵۹۵۱ PDF خوش‌خیم است. پایگاه اطلاعاتی را به ۱۰ قسمت مساوی تقسیم کرده و هر بار ۹تای آن را برای آزمایش و یکی را

برای آزمون در نظر می‌گیریم که نتایج آن در جدول ۱ آورده شده است.

جدول ۱. دقت و نرخ مثبت کاذب (اعتبارسنجی صلیبی)

Classifiers	Accuracy	False Positive
Naïve Bayes	٪۹۷	٪۵/۲
SVM	٪۹۹	٪۰/۶
Random Forest	٪۹۹	٪۰/۳

بهترین نتیجه از طبقه بند Random Forest بدست آمده است که نمودار ROC آن در شکل زیر آورده شده است.



شکل ۳. نمودار ROC

اکثر کارهای مرتبط نیز نتیجه بالای ٪۹۹ را بدست آورده‌اند یعنی روش ما بهبودی را در دقت ایجاد نکرده است که ممکن است بدلیل تعداد کم خصیصه‌ها باشد و دقتی بسیار نزدیک به روش‌های قبلی بدست آورده است. به همین دلیل نرخ تشخیص نمی‌تواند به تنهایی عاملی برای مقایسه باشد. به منظور مقایسه دقیق‌تر از پارامترهای سرعت و فرار از تشخیص استفاده می‌شود.

سرعت تشخیص

علاوه بر دقت، سرعت نیز نقش موثری را در یک تشخیص‌کننده ایفا می‌کند زیرا ضد بدافزارها معولا فایل‌های زیادی را پویش می‌کنند و بدنبال فایل‌های مخرب هستند. به همین دلیل سرعت پردازش یک فایل آزمایش، بسیار حائز اهمیت است. از آنجا که در روش‌های یادگیری ماشین مهمترین عامل برای سرعت، خصیصه‌ها و استخراج آنها است، تعداد خصیصه و نوع آن نیز اهمیت دارد. جدول زیر تعداد خصیصه‌های استفاده شده در روش‌های دیگر را نشان می‌دهد.

جدول ۲. تعداد خصیصه‌های استخراج شده در هر روش

رویکرد پیشنهادی ما	Malware Slayer [26]	PDF Rate [27]
۷	۱۱۵	۶۷

جدیدترین روش ارائه‌شده در سال ۲۰۱۳ است که مبتنی بر گراف است. در جدول زیر مقایسه‌ای میان روش ما و آن روش در زمان آزمون انجام شده است.

همانطور که ملاحظه می‌شود، زمان یافتن الگو در یک فایل آزمون در روش پیشنهادی ما بسیار کمتر از روش پرهزینه Srndic ۲۰۱۳ است.

جدول ۳. مقایسه روش ما با Srndic از لحاظ سرعت

روش پیشنهادی ما	Srndic [29]
استخراج نام ها (W)	استخراج نام ها (W)
هیچ (۰)	استخراج گراف (X)
هیچ (۰)	پیمایش گراف برای استخراج مسیرهای مختلف
تطابق الگو – جستجو در یک مسیر (Z') در نتیجه Z' < Z	تطابق الگو – جستجو در چندین مسیر (Z)

فرار از تشخیص و مقاوم بودن

قدرت تشخیص بالا یکی از معیارهای مهم و قابل قبول است اما برای هر روش مبتنی بر تشخیص، مقاوم بودن در برابر فرار از آن نیز اهمیت دارد. بنابراین مقاومت خصیصه‌های انتخابی در برابر حملات فرار و تقلید نیز بسیار مهم و حیاطی است و نشان‌دهنده نرخ تشخیص واقعی در محیط‌های واقعی، یعنی محیط‌هایی که مهاجم سعی در فرار از روش را دارد، است. مکانیزم تشخیص در این مقاله بر اساس مستندات گذشته و قبلی است و این شباهت میان مستندات می‌تواند توسط مهاجم دستکاری شود زیرا بعضی از خصیصه‌ها به راحتی قابل تغییر هستند و بعضی دیگر نیز مشکل‌تر هستند. برای مثال، اکثر مهاجمان از جاوااسکریپت به منظور سوءاستفاده از آسیب‌پذیری‌ها استفاده می‌کنند بنابراین حذف این خصیصه توسط مهاجمان به سختی امکان‌پذیر است.

طبقه‌بندی و تشخیص مستندات مخرب براساس شباهت خصیصه های استخراج شده آنها است. یک تکنیک واضح، به کارگیری حمله تقلید است به این صورت که مستندات مخرب به صورت هدفمند دستکاری شوند به‌طوری‌که خصیصه‌های آنها به خصیصه‌های مستندات نرمال نزدیک شوند درحالیکه هنوز محتوای مخرب خود را دارا باشند. اگر مهاجم بداند که چه خصیصه هایی در طبقه بند استفاده می‌شود و خصیصه های مهم کدامیک هستند، مهاجم می‌تواند آن خصیصه‌ها را به منظور اینکه خود را نرمال نشان دهد به خود اضافه کند و طبقه بندی را به نفع خود تغییر دهد. به منظور شبیه‌سازی این نوع حمله، مهاجمان خصیصه هایی را که متمایز کننده میان دو کلاس هستند را بدست آورده و آنها را تغییر می‌دهند. ما با استفاده از الگوریتم CfsSubsetEval خصیصه های متمایز را انتخاب کردیم و تک تک آنها را در ۶ مرحله به مستندات مخرب اضافه کردیم. نرخ تشخیص اشتباه مستندات را در جدول زیر آورده‌ایم و با روش ارائه شده در PDFRate ۲۰۱۱ که تنها کاری است این آزمایش را انجام داده است، مقایسه کردیم. همانطور که ملاحظه می شود، بدون اضافه کردن خصیصه روش

PDFRate نتـایج بهتـری را گرفتـه اسـت امـا بـا اضـافه‌کـردن خصیصه‌های متمایز به نمونه‌های مخرب روش پیشنهادی ما نتـایج بهتری را کسب کرده است و در برابـر حمـلات mimcry مقـاوم تـر نشان داده است.

جدول ۴. مقایسه از نظر فرار از تشخیص

PDFRate		روش پیشنهادی ما	
None	٪۰/۱۲	None	٪۰/۸
(+) count font	٪۱۳/۶۱	(+) /L /Font	٪۱/۸
(+) count javascript	٪۱۸/۹۰	(+) /L /Font /ProcSet	٪۳/۲
(+) count stream diff	٪۱۸/۹۹	(+) /L /Producer	٪۴/۳
(+) count js	٪۲۱/۳۵	(+) /FontDescriptor /OPM	٪۶/۳
(+) pos box max	٪۲۳/۲۷	(+) /L /Producer /Font	٪۱۳/۳
(+) image totalpx	٪۲۳/۳۰	(+) /Producer /OPM /Page	٪۱۵/۹

نتیجه‌گیری

در این مقاله، یک سیستم جدید برای تشخیص فایل‌های PDF مخرب ارائه شده است. این سیستم مبتنی بر یادگیری ماشین و داده‌کاوی بوده و مرتبط به ساختار داخلی فایل است که به راحتی ایجاد می‌شود و در مقابل حملات مختلف مقاوم است. همچنین مقدار دقت آن بسیار نزدیک به سیستم‌های مشابه است. در حقیقت، این ابزار به طور خاص فایل‌های PDF را مورد بررسی قرار می‌دهد و برای فایل‌های دیگر مانند exe و یا doc می‌توان ماژول‌های دیگری طراحی نمود. مزیت این ابزار این است که تمامی فایل‌های PDF را می‌تواند مورد تحلیل قرار دهد و این‌گونه نیست که تنها آنهایی که کدهای جاوااسکریپت دارند را مورد هدف قرار دهد. سیستم ارائه شده نتایج بهتری را از نظر مجموع دقت، سرعت و مقاومت در برابر فرار را در مقایسه با روش‌های دیگر نشان داده است. این ابزار می‌تواند بعنوان یک افزونه برای ضدبدافزارها مورد استفاده قرار گیرد. در آینده می‌خواهیم از الگوهای قوی‌تر و سریع‌تری بعنوان خصیصه استفاده کنیم.

مرجع‌ها

[1] Symantec, "Symantec Global Internet Security Threat Report (Trends for 2009)," 2010.

[2] IBM, "IBM X-Force 2010 Mid-Year Trend and Risk Report," 2010.

[3] H. security. (2012). Vorsicht bei angeblicher telekom-onlinerechnung. Available: http://heise.de/-1545909

[4] Symantec. (2012). PDF Malware Writers Keep Targeting Vulnerability. Available: http://www.symantec.com/connect/blogs/pdf-malware-writers-keep-targeting-vulnerability

[5] P. Muncaster. (2012). Blackhole crimeware kit drives web threat spike. Available: http://www.theregister.co.uk/2012/01/26/sophos_fake av_conficker/

[6] Google. (2012). Google warns of using Adobe Reader - particularly on Linux. Available: http://www.h-online.com/open/news/item/Google-warns-of-using-Adobe-Reader-particularly-on-Linux-1668153.html

[7] Adobe. (2009). PDF Reference. Available: http://www.adobe.com/devnet/pdf/pdf_reference.html

[8] B. L. Scott Kaplan, Ben Zorn, Christian Siefert, and Charlie Cursinger, ""NOFUS: Automatically Detecting" + String.fromCharCode(32) + "ObFuSCateD ".toLowerCase() + "JavaScript Code"," 2011.

[9] S. S. Wei-Jen Li, Angelos Stavrou, Elli Androulaki, Angelos D. Keromytis, "A study of malcode-bearing documents," presented at the DIMVA, Switzerland, 2007.

[10] S. ABISH, SHAFIQ, M., AND FAROOQ, M. , "Malware detection using statistical analysis of byte-level file content," presented at the SIGKDD Workshop on CyberSecurity and Intelligence Informatics 2009.

[11] S. STOLFO, WANG, K., AND LI, W. , "Fileprint analysis for malware detection," presented at the CCS WORM, 2005.

[12] S. A. K. M. Zubair Shafiq, Muddassar Farooq, "Embedded Malware Detection Using Markov n-Grams," presented at the DIMVA, Paris, France, 2008.

[24] J. S. CROSS, AND MUNSON, M. A. , "Deep PDF parsing to extract features for detecting embedded malware. Tech. Rep. SAND20117982, Sandia National Laboratorie," 2011.
[25] N. Š. Pavel Laskov, "Static detection of malicious JavaScript-bearing PDF documents," presented at the ACSAC, Texas, USA, 2011.
[26] G. G. D. Maiorca, and I. Corona, "A pattern recognition system for malicious pdf filles detection," presented at the 8th international conference on Machine Learning and Data Mining in Pattern Recognition, MLDM, Berlin, Heidelberg, 2012.
[27] C. S. a. A. Stavrou, "Malicious pdf detection using metadata and structural features," presented at the 28th Annual Computer Security Applications Conference, ACSAC 2012.
[28] Pdfrate. Available: http://pdfrate.com
[29] N. S. a. P. Laskov, "Detection of malicious pdf files based on hierarchical document structure," presented at the 20th Annual Network & Distributed System Security Symposium, 2013.
[30] Pdf tools. Available: http://blog.didierstevens.com/programs/pdftools
[31] Peepdf. Available: Http://eternal-todo.com/tools/peepdf-pdf-analysis-tool
[32] Origami framework. Available: Http://esec-lab.sogeti.com/pages/Origami
[33] a. M. K. Jiawei Han, Data Mining: Concepts and Techniques, 3 ed.: Morgan Kaufmann, 2011.
[34] R. S. Rakesh Agrawal, "Fast Algorithms for Mining Association Rules in Large Databases," presented at the VLDB, 1994.
[35] A. Webb, Statisitcal Pattern Recognition: Wiley, 2005.
[36] Yahoo. Yahoo Search Engine. Available: http://www.yahoo.com
[37] contagio. (2013). PDF Dataset. Available: http://contagiodump.blogspot.com/
[38] M. A. Hall, "Correlation-Based Feature Subset Selection for Machine Learning," University of Waikato., Hamilton, New Zealand, 1998.
[39] E. F. Mark Hall, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, "The WEKA Data Mining Software: An Update," SIGKDD Explorations, vol. 11, 2009.
[13] S. RAUTIAINEN, "A look at portable document format vulnerabilities," Information Security Technical Report 14, 1 2009.
[14] C. K. Marco Cova, Giovanni Vigna, "Detection and analysis of drive-by-download attacks and malicious JavaScript code," presented at the WWW, USA, 2010.
[15] T. K. Konrad Rieck, Andreas Dewald, "Cujo: efficient detection and prevention of drive-by-download attacks," presented at the ACSAC, Texas, USA, 2010.
[16] B. L. Charlie Curtsinger , Benjamin Zorn, Christian Seifert, "ZOZZLE: fast and precise in-browser JavaScript malware detection," presented at the USENIX Security Symposium, 2011.
[17] M. C. Davide Canali, Giovanni Vigna, Christopher Kruegel, "Prophiler: a fast filter for the large-scale detection of malicious web pages," presented at the WWW, Hyderabad, India, 2011.
[18] Wepawet. Available: Http://wepawet.iseclab.org/index.php
[19] C. W. Markus Engelberth, Thorsten Holz, "Detecting malicious documents with combined static and dynamic analysis," presented at the Virus Bulletin, Geneva, 2009.
[20] T. H. Carsten Willems, Felix Freiling, "Toward Automated Dynamic Malware Analysis Using CWSandbox," IEEE Security and Privacy, vol. 5, pp. 32-39, 2007.
[21] T. H. K. Rieck, C. Willems, P. Dussel, and P. Laskov, "Learning and classification of malware behavior," presented at the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA Berlin, Heidelberg, 2008.
[22] G. S. Zacharias Tzermias, Michalis Polychronakis, Evangelos P. Markatos, "Combining static and dynamic analysis for the detection of malicious documents," presented at the EUROSEC, 2011.
[23] K. S. a. S. K. a. F. M. a. N. Provos, "SHELLOS: Enabling Fast Detection and Forensic Analysis of Code Injection Attacks," presented at the USENIX Security Symposium, 2011.