مجله علمی- پژوهشی "پدافند الکترونیکی و سایبری"
ویژه‌نامه ترکیبیات، رمزنگاری و محاسبات، ص ۲۱۱ -۲۰۱

دسته‌بندی خانواده باج‌افزارهای رمزگذار با استفاده از پرونجاهای دام

محمدهادی علائیان، سعید پارسا[۲*]

۱- دانشجوی دکتری، ۲- دانشیار، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران
(دریافت: ۱۳۹۷/۰۸/۲۵ پذیرش: ۱۳۹۸/۰۷/۰۲)

چکیده

بدافزارها یکی از تهدیدات همیشگی برای دستگاه‌های رایانه‌ای به شمار می‌آیند. بدافزارها با ورود به دستگاه‌های رایانه‌ای بسته به اهدافشان، سعی دارند در روند عادی دستگاه‌های رایانه‌ای اخلال ایجاد کنند. در این بین، بدافزارهایی به نام باج‌افزار وجود دارند که پس از ورود به دستگاه‌های رایانه‌ای و محدود کردن دسترسی قربانی به دستگاه رایانه‌ای خود با رمزگذاری پَروَنجاهای قربانی یا به‌صورت قفل‌گذاری دستگاه درصدد اخاذی از قربانی بر می‌آید. این نوع بدافزار، یک تفاوت بسیار آشکار با دیگر بدافزارها دارد، باج‌افزارها صراحتاً قربانی را از وجود خود بر روی دستگاه با خبر می‌سازند. برخلاف آسیب‌های جدی که بر روی دستگاه‌های قربانی وارد می‌سازند، می‌توان با ویژگی‌های منحصربه‌فردی که بر روی دستگاه برجای می‌گذارند شناسایی گردند. در این مقاله، با ارائه محیطی جهت اجرای باج‌افزارها و دستگاهی که ویژگی‌های مختلفی از تعاملات باج‌افزارها با فرمت پَروَنجا را استخراج می‌کند، نشان داده می‌شود که با کمک این ویژگی‌ها می‌توان با دقت ۹۸/۸۸ درصد علاوه بر شناسایی باج‌افزارها، به کمک الگوریتم‌های یادگیری ماشین خانواده باج‌افزارها را نیز تعیین کرد.

واژه‌های کلیدی: بدافزار، باج‌افزار، باج‌افزارهای رمزگذار، باج‌افزارهای قفل‌گذار، دسته‌بندی خانواده باج‌افزارها

۱- مقدمه

امروزه نمی‌توان تأثیر نرم‌افزارها در زندگی روزمره خود را انکار کرد. نرم‌افزارها با کارایی و دقت بالای خود از ملزومات زندگی امروزی ما به حساب می‌آیند و استفاده از آن‌ها در محیط‌های محاسباتی نه فقط یک امتیاز، بلکه به یک نیاز ضروری تبدیل‌شده است. اما در کنار این نرم‌افزارهای سودمند نرم-افزارهایی هم همواره با عنوان بدافزار وجود دارند که امنیت دستگاه‌های رایانه‌ای را تهدید می‌کنند [۲-۱] این برنامه‌ها صدمات جدی و در برخی موارد خسارات جبران‌ناپذیری به دستگاه‌های رایانه‌ای وارد می‌کنند. بدافزارها انواع مختلفی دارند که باج‌افزارها یکی از آن‌ها به حساب می‌آید. این دسته از بدافزارها که هدف آن‌ها بیشتر جنبه مالی دارند با حمله به پَروَنجاهای شخصی بر روی دستگاه قربانی آن‌ها را غیرقابل‌دسترس می‌کند [۳] که در قبال دریافت مبلغی، این پَروَنجاها را دوباره به حالت قبل برمی‌گرداند. به‌عبارت‌دیگر باج-افزار را می‌توان بدافزاری برای اخاذی از کاربر به‌صورت دیجیتالی دانست. اولین باج‌افزار در سال ۱۹۸۹ با نام AIDS شناسایی شد [۴].

به گزارش سیمنتک در ۶ ماه اول سال ۲۰۱۷ سازمان‌ها ۴۲ درصد از اهداف باج‌افزارها را تشکیل می‌دهند. این در حالی است که در سال‌های ۲۰۱۶ و ۲۰۱۵ سازمان‌ها به ترتیب ۳۰ و ۲۹ درصد از اهداف این نوع از بدافزارها را شامل می‌شدند. دلیل اصلی این افزایش مربوط به دو باج‌افزار WannsCry و Petya می‌باشد [۵]. از سال ۲۰۰۵ تا سال ۲۰۱۴ تنها ۱۴ خانواده از باج‌افزارها تولیدشده بودند. این در حالی است که تنها در فصل اول سال ۲۰۱۵ شاهد ایجاد ۲۷ خانواده از باج‌افزارهای جدید بودیم [۶].

به‌طورکلی، باج‌افزارها به دو دسته تقسیم می‌شوند: باج‌افزارهای قفل‌گذار و باج‌افزارهای رمزگذار: باج‌افزارهای قفل‌گذار سرویس‌های متعددی مانند صفحه‌نمایش، دستگاه‌های ورودی بر روی سیستم‌عامل قربانی را قفل کرده و فعالیت‌های کاربر را به تعدادی فعالیت مرتبط با فرآیند واریز مبلغ معین‌شده توسط باج‌افزار محدود می‌سازد [۳]. اما نوع دیگر باج‌افزار یعنی رمزگذار، با

[۱] رایانامه نویسنده مسئول: hadi_alaeiyan@comp.iust.ac.ir

رمزگذاری داده‌ها و پَروَنجاهای شخصی کاربر سعی در اعمال محدودیت برای کاربر می‌کند. می‌توان گفت که یک حمله موفق از سوی باج‌افزار از ۵ مرحله تشکیل می‌یابد: ۱- استقرار، ۲- نصب و راه‌اندازی، ۳- دستور و کنترل[1]، ۴- تخریب و ۵- اخاذی [۷]. در نخستین مرحله حمله، باج‌افزار نیازمند به بارگذاری پَروَنجا اصلی بر روی دستگاه قربانی می‌باشد که برای این کار می‌تواند از بردارهای حمله متعددی مانند ایمیل[2]، بهره‌گیری از آسیب-پذیری‌های متعدد، بارگذاری و راه‌اندازی خودکار استفاده کند. گام بعدی یعنی مرحله نصب و راه‌اندازی می‌تواند بسیار پیچیده باشد. در بسیاری از موارد، گونه‌های جدید باج‌افزارهای رمزنگار ابتدا از ویروس‌های ماکرو و یا پی‌دی‌اف آلوده جهت ورود به دستگاه استفاده می‌کنند [۸]. سپس به‌محض این‌که بدافزار وارد دستگاه شد، کد بدخواه خود را اجرا کرده و سپس دستگاه میزبان را مورد تحلیل قرار می‌دهد تا واقعی بودن و یا جعبه‌شنی[3] بودن دستگاه را شناسایی کند. بعدازآن باج‌افزار با به‌کارگیری روش-هایی مانند استفاده از آدرس MAC[4] سعی می‌کند خود را منحصربه‌فرد کند تا طراح باج‌افزار اطلاع داشته باشد که کدام دستگاه آلوده‌شده است. در این مرحله باج‌افزار چندین اسکریپت برای اطمینان از غیرفعال بودن دستگاه‌های حفاظتی بر روی دستگاه قربانی اجرا می‌کند. ازجمله کارهایی که انجام می‌دهد هم می‌توان به موارد زیر اشاره کرد: از کار انداختن ویژگی کپی سایه[5] بر روی پَروَنجاها و دیوارک‌ها، از کار انداختن ویژگی‌های بازیابی و درنهایت غیر فعال‌سازی نرم‌افزارهای ضد بدافزاری و گزارش‌گیری بر روی دستگاه. در حملات باج‌افزاری، به‌محض این‌که کد مخرب نصب و راه‌اندازی شد [۹]، باج‌افزار درصدد ایجاد ارتباط با سرور دستور و کنترل برمی‌آید و منتظر دستور از جانب سرور می‌شود [۷]، این دستورات بسته به نوع باج‌افزار و اهداف آن می‌تواند هر چیزی باشد مانند شناسایی نوع پَروَنجاهایی که بر روی دستگاه قرار دارند، مدت انتظار باج‌افزار تا شروع آلودگی دستگاه و دستورات متعدد دیگر که طراح باج‌افزار تمایل دارد قبل از شروع آلودگی از آن‌ها مطلع شود.

در برخی از باج‌افزارها اطلاعات بسیار مهمی از دستگاه قربانی به سرور کنترل و دستور ارسال می‌شود که ازجمله این اطلاعات می‌توان به آدرس IP، سیستم‌عامل، مرورگرها و محصولات ضد بدافزاری نصب‌شده اشاره کرد. در مرحله کلیدی که برای قفل-گذاری دستگاه و یا رمزنگاری پَروَنجاها برای استفاده توسط بدافزارها بر روی دستگاه قربانی لازم است حاضر می‌باشد. در این مرحله تمامی پَروَنجاهایی که توسط فرآیندهای کنترل و دستور شناسایی می‌شوند [۱۰] توسط باج‌افزارگذاری می‌شوند.

پَروَنجاهای موردنظر باج‌افزار نسبت به نوع خانواده باج‌افزارها متفاوت می‌باشد اما ازجمله انواع پَروَنجاهای بسیار محبوب بین باج‌افزارها می‌توان به مستندات آفیس مایکروسافت، JPG، GIF و هر نوع پَروَنجا دیگری اشاره کرد. بسیاری از گونه باج‌افزارها نه‌تنها پَروَنجاها بلکه اسامی پَروَنجاها را هم رمزنگاری می‌کنند تا قربانی را از پی بردن به این‌که کدام پَروَنجاها رمزگذاری شده‌اند را ناتوان سازد. رمزگذاری پَروَنجاها به سه صورت متقارن، نامتقارن و یا ترکیبی از این دو انجام می‌پذیرد. درنهایت در مرحله اخاذی آگاه-سازی کاربر از نوع حمله انجام‌شده بر روی دستگاه خود توسط باج‌افزار صورت می‌گیرد. بعد از آن‌که همه پَروَنجاهای موردنظر باج‌افزار رمزگذاری شدند، به قربانی صفحه‌ای مبنی بر آسیب دستگاه نمایش داده می‌شود [۷].

در این مقاله، یک دستگاه تحلیل باج‌افزار، جهت تحلیل و همچنین دسته‌بندی باج‌افزارها ارائه‌شده است. در رویکرد مطرح‌شده، با استخراج ویژگی‌هایی همچون درگاشت شانون[6]، تعداد پَروَنجاهای اضافه‌شده در مسیر پَروَنجاهای عسل[7]، دسترسی به VSSAdmin، دسترسی به Recent files، تغییر حجم پَروَنجاها و نحوه رمزگذاری پَروَنجاهای عسل و سپس با ایجاد یک مجموعه داده از این ویژگی‌ها و درنهایت استفاده از الگوریتم جنگل‌ها تصادفی جهت آموزش یک مدل دسته‌بند، می‌توان رفتارهای باج-افزاری را شناسایی کرده و باج‌افزارها را بر اساس ویژگی‌های مطرح‌شده دسته‌بندی کرد. به‌منظور مقایسه نتایج به‌دست‌آمده

[1] C2(Command & Control)
[2] E-mail
[3] Sandbox
[4] media access control address
[5] Shadow Copy : فنّاوری که در سیستم‌عامل ویندوز تعبیه‌شده که امکان کپی‌برداری از پرونجاهای پشتیبان را حتی در مواقع کار کردن با پرونجاها را برای کاربر فراهم می‌سازد.

[6] Shannon Entropy
[7] Honey files: این پرونجاهای پرونجاهایی قلابی می‌باشند که تأثیرات باج‌افزارهای بر روی این نوع از پَروَنجاها بررسی می‌شوند.

در این مقاله از مجموعه نمونه باج‌افزارهای به کار برده شده در [۱۱] استفاده‌شده است.

نوآوری‌های مطرح‌شده در این مقاله در زیر لیست شده‌اند:

۱. استفاده از پَروَنجاهای عسل جهت تحریک باج‌افزارها
۲. استخراج ویژگی‌هایی جهت شناسایی باج‌افزارها
۳. دسته‌بندی باج‌افزارها بر اساس خانواده‌هایشان به کمک ویژگی‌های استخراج‌شده.

در ادامه، در بخش دوم به کارهای مرتبط در حوزه شناسایی باج‌افزارها پرداخته‌شده و نحوه کار دستگاه‌های ارائه‌شده در این مقالات و نقاط قوت و ضعف این دستگاه‌ها پرداخته می‌شود و همچنین روش پیشنهادی جهت شناسایی خانواده باج‌افزارها ارائه‌شده و در قسمت سوم نتایج به‌دست‌آمده از اعمال روش پیشنهادی بر روی دیتاست موجود در [۱۱] ارائه می‌شود.

۲- روش تحقیق

در این قسمت به کارهای مرتبط پرداخته‌شده و دستگاه‌های ارائه‌شده در مقالات موردبررسی قرارگرفته و درنهایت روش پیشنهادی جهت تحلیل و دسته‌بندی باج‌افزارها ارائه می‌شود.

۲-۱- کارهای مرتبط

در این قسمت به کارهای مرتبط در حوزه شناسایی باج‌افزارها پرداخته‌شده و به نحوه کار دستگاه‌های ارائه‌شده در این مقالات و نقاط قوت و ضعف این دستگاه‌ها پرداخته می‌شود.

باج‌افزارهای رمزگذار سعی می‌کنند، به‌صورت هدفمند یا به‌صورت تصادفی (بسته به نوع باج‌افزار) پَروَنجاهای کاربر را رمزگذاری کنند. در این صورت، دسترسی کاربر به پَروَنجاهای شخصی خود محدودشده و درنتیجه کاربر برای رمزگشایی پَروَنجاها خود باید مبلغی را جهت دریافت کلید رمز واریز کند. کلید بدافزارها می‌تواند به‌صورت محلی به‌وسیله خود باج‌افزار و یا از طریق سرور دستور و کنترل از راه دور تولید شود.

در [۱۲] به مطالعه رفتارهای ترس‌افزارها[1] و باج‌افزارها پرداخته‌شده است و سازوکارهای رمزگذاری، تعاملات دستگاه پَروَنجا بسیاری از باج‌افزارها در طی یک سال رصد شده است. روش‌های متعددی در این مقاله جهت مقابله با این نوع از بدافزارها پیشنهادشده اما به ارزیابی روش‌های پیشنهادی در این مقاله پرداخته نشده است.

راه‌کار ارائه‌شده در [۱۳] بر پایه رصد فراخوانی‌های فرمت پَروَنجا باج‌افزارها در محیط جعبه‌شنی می‌باشد. روش معرفی‌شده زمانی که باج‌افزار با پَروَنجاهای شخصی کاربر در تعامل باشد آن‌ها را رصد می‌کند و به‌صورت موازی هم همه تغییرات صفحه‌نمایش کاربر را زیر نظر می‌گیرد و سعی دارد تا رفتارهای شبه باج افزاری فرمت پَروَنجا پَروَنجاهای اجرایی را شناسایی کند. این روش حتی قادر به شناسایی باج‌افزارهای قفل‌گذار نیز می‌باشد. روش پیشنهادی یک روش بلادرنگ شناسایی باج‌افزار نبوده و می‌توان این روش را به‌عنوان یک روش تحلیلی در نظر گرفت.

اما روش پیشنهادی در [۱۴] برای شناسایی باج‌افزار بر روی سیستم‌عامل اندروید می‌باشد. این روش، اقدام به شناسایی رفتار باج‌افزارها در سطح کاربر می‌کند. روش پیشنهادی در [۱۴] به دو صورت ایستا و پویا برنامه‌های اندروید را تحلیل می‌کنند. این روش از تحلیل لکه‌گذاری ایستا و شبیه‌سازی سبک جریانی از فراخوانی‌های توابع را که منجر به رمزنگاری پَروَنجاها یا قفل صفحه‌نمایش می‌شوند را مشخص می‌کند. این روش برای شناسایی رفتار تهدیدآمیز بر پایه یادگیری و همچنین بر پایه روش پردازش زبان طبیعی (NLP) می‌باشد. روش پیشنهادی مختص خانواده خاصی از باج‌افزارها نمی‌باشد و می‌توان آن را برای هر خانواده‌ای از باج‌افزارها استفاده کرد.

روش EldeRan یک فن تحلیل پویا با رویکرد یادگیری ماشین می‌باشد که در [۶] پیشنهادشده است. این روش باج‌افزارها را بر اساس مجموعه فعالیت‌هایی که برنامه‌ها در مرحله نصب انجام می‌دهند شناسایی می‌کند. فرضیه اصلی این روش این است که باج‌افزارها شامل ویژگی‌های یکتایی هستند که در زمان تحلیل پویای آن‌ها شباهت زیادی به هم دارند پس می‌توان از این

[1]: این نوع از بدافزارها با نمایش گزارش‌های قلابی، به کاربر هشدار می‌دهد Scareware تا به‌منظور جلوگیری از آسیب به دستگاه خود ابزار ضد بدافزاری معرفی‌شده را دانلود و نصب کند تا بدافزارهای موجود بر روی دستگاه خود را به کمک این ابزار از بین ببرد اما در اصل از این طریق بدافزار را وارد دستگاه قربانی می‌کند.

ویژگی‌ها برای شناسایی باج‌افزارها استفاده کرد. این روش ابتدا ویژگی‌های مرتبط به رفتار باج‌افزارها را انتخاب کرده و سپس هر برنامه تازه نصب‌شده بر روی رایانه را به‌وسیله الگوریتم یادگیری ماشین بدون استفاده از فن‌های بر پایه اکتشافی یا امضا دسته-بندی می‌کند. این مقاله می‌خواهد نشان دهد که یک باج‌افزار می‌تواند با دقت بالایی بر اساس ویژگی‌های محدودی قبل از آلودگی شناسایی شود.

روش مطرح‌شده در [۱۱] از روش یادگیری ماشین برای فعالیت-های دیسک استفاده می‌کند. ویژگی‌های انتخاب‌شده برای یادگیری ماشین از میلیون‌ها درخواست ورودی/خروجی استخراج‌شده‌اند. این ویژگی‌ها در دو حالت جمع‌آوری‌شده‌اند: ۱- زمانی که دستگاه وضعیت عادی دارد و هیچ‌گونه باج‌افزاری بر روی آن موجود نمی‌باشد؛ ۲- زمانی که دستگاه موردحمله باج-افزار قرارگرفته باشد. راه‌حل پیشنهادی از ۳ راه‌انداز[1] ساخته‌شده است. راه‌انداز اول که مسئول ترمیم پَروَنجاهای می‌باشد، برای هر عملیات نوشتن و بازنام‌گذاری[2] از پَروَنجا مربوطه یک نسخه پشتیبان تولید می‌شود. راه‌انداز دوم هم مسئول تشخیص پایه رمزگذاری[3] توکاری شده در فرآیندها می‌باشد و درنهایت راه‌انداز سوم به کمک الگوریتم جنگل‌های تصادفی[4] اقدام به شناسایی باج‌افزار می‌کند. این روش در مقابل باج‌افزارهایی که از توابع رمزگذاری ناشناخته استفاده می‌کنند کارساز نمی‌باشد.

یکی دیگر از روش‌های پیشنهادی در این حوزه CryptoDrop می‌باشد. این روش که در [۸] پیشنهادشده است روشی پیش هشدار برای شناسایی باج‌افزار می‌باشد. این روش تمامی فعالیت-های پَروَنجا را موردبررسی قرار داده و در موارد شناسایی چیزهایی مشکوک به کاربر هشدار می‌دهد. شناسایی با این روش با استفاده از سه ویژگی تغییر در نوع پَروَنجا، مقدار شباهت پَروَنجا و درگاشت صورت می‌گیرد. البته این روش هیچ تضمینی در رابطه با حداقل داده‌هایی که قبل از شناسایی از دست می‌رود نداده و همچنین امکان ترمیم پَروَنجاهای رمزنگاری‌شده هم در این روش وجود ندارد. یکی از ویژگی‌هایی که در این روش در نظر گرفته‌شده است، ویژگی تغییر در نوع پَروَنجاها بعد از رمزگذاری پَروَنجاها می‌باشد اما این ویژگی نمی‌تواند برای تمامی باج‌افزارها صادق باشد به‌عنوان‌مثال، باج‌افزار satana هیچ تغییری در نوع پَروَنجا ایجاد نکرده و تنها نام آن را تغییر می‌دهد.

۲-۲- روش پیشنهادی جهت شناسایی خانواده باج‌افزارها

در این بخش تمامی مراحل روش پیشنهادی برای شناسایی خانواده باج‌افزارها به‌طور کامل توضیح داده می‌شود.

همان‌طور که در شکل (۱) مشاهده می‌شود، ابتدا هرکدام از باج-افزارها و برنامه‌های بی‌آزار بر روی جعبه‌شنی که ساختار آن در بخش‌های بعد توضیح داده خواهد شد اجرا می‌شوند.

در زمان اجرا، تمامی تعاملات فرمت پَروَنجا برای برنامه اجراشده توسط یک راه‌انداز مینی‌فیلتر ثبت‌شده و ویژگی‌های شناساگر باج‌افزار که در ادامه به بررسی آن‌ها پرداخته خواهد شد استخراج می‌شوند، در گام بعد نیز یک ماتریس از تمامی ویژگی‌های استخراج‌شده از کل برنامه‌ها تهیه می‌شود. درنهایت با ایجاد یک مدل به‌وسیله الگوریتم جنگل‌های تصادفی توسط ماتریس تشکیل یافته، اقدام به دسته‌بندی نمونه‌ها می‌شود.

۲-۲-۱- محیط جعبه‌شنی

یکی از مهم‌ترین مراحل، آماده‌سازی محیطی مناسب جهت اجرای پَروَنجاهای اجرایی بر روی جعبه‌شنی می‌باشد. همان‌طور که در مقدمه بیان شد بسیاری از باج‌افزارها پَروَنجاهای شخصی موجود بر روی دستگاه قربانی را مورد هدف قرار داده و با رمزگذاری بر روی پَروَنجاهای کاربر سعی می‌کنند تا درازای واریز مبلغی از سوی قربانی کلید پَروَنجاهای رمزگذاری شده بر روی دستگاه را به کاربر ارائه دهند. پس به‌منظور تحلیل باج‌افزار باید محیطی آماده شود که بتوان در یک مدت محدود رفتارهای باج-افزار را مشاهده کرد.

- **پَروَنجاهای عسل**[5]: پَروَنجاهای عسل جهت شناسایی فعالیت-های مخرب بر روی دستگاه قرار می‌گیرند. این نوع پَروَنجاها در صورت دسترسی و تغییر از سوی افراد متخلف [۱۶-۱۵] یا

[1] Driver
[2] Rename
[3] Cryptographic primitive
[4] Random forest classifier
[5] Honey Files

بدافزارها می‌توانند دریافتن افراد و یا برنامه‌های مخرب ما را یاری کنند.

این پَروَنجاها در اصل هیچ ارزشی برای کاربر نداشته و هدف از آن‌ها شناسایی اقدامات مخرب می‌باشد. با قرار دادن این پَروَنجاها در نواحی که پتانسیل بالایی در مواجه‌شدن با اقدامات مخرب دارند می‌توانند کمک بسیار زیادی در شناسایی باج‌افزارها داشته باشند. می‌توان با قرار دادن این نوع از پَروَنجاها در نقاط مختلف دستگاه و رصد مداوم این پَروَنجاها افراد متخلف و همچنین بدافزارهای مختلفی که بر روی پَروَنجاها تأثیر مخربی دارند را شناسایی کرد. ازجمله ویژگی‌هایی که این نوع از پَروَنجاها باید داشته باشند می‌توان به موارد زیر اشاره کرد [۱۵]:

۱. **واقعی به نظر رسیدن**: به این معنی که باج‌افزارها این نوع از پَروَنجاها را به‌عنوان پَروَنجا واقعی ببینند و همانند دیگر پَروَنجاهای اصلی با آن‌ها رفتار کنند.

۲. **تحریک‌کننده باشند**: بایستی این نوع از پَروَنجاها برای باج‌افزارها و در حالت کلی برای بدافزارها طوری تعیین شوند تا بدافزارها رفتار واقعی خود را نشان دهند.

۳. **قابلیت دسترسی**: پَروَنجاهای عسل باید در مسیرهایی قرار گیرند تا باج‌افزارها به‌راحتی به آن‌ها دسترسی داشته باشند.

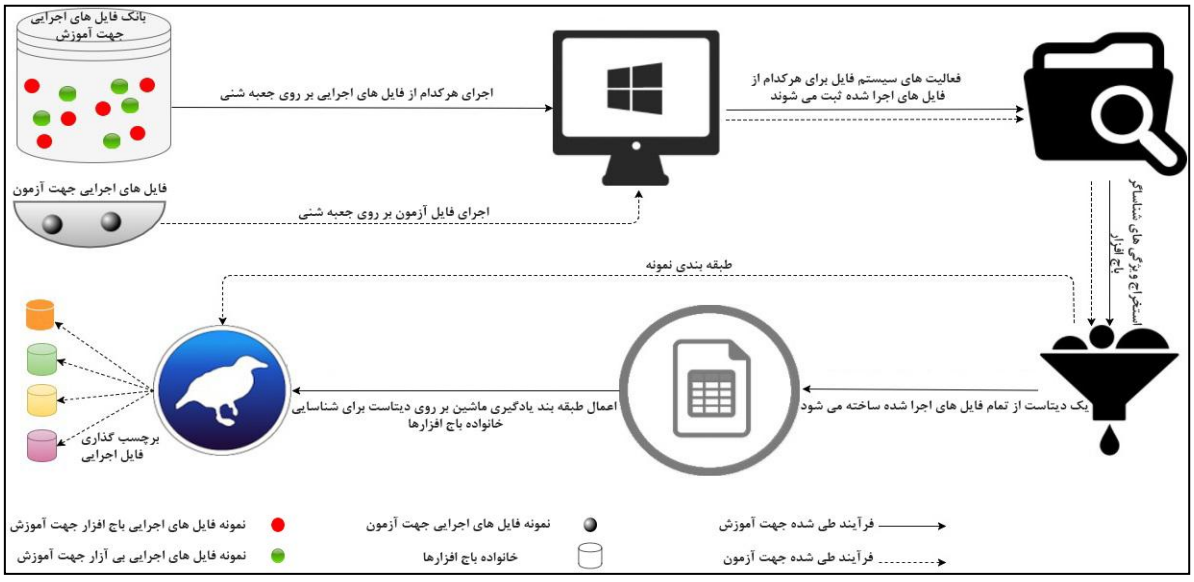
۴. **قابلیت شناسایی**: این نوع از پَروَنجاها باید به‌گونه‌ای باشند تا با رصد این نوع پَروَنجاها، پَروَنجا اجرایی مخرب شناسایی شود.

۵. **تنوع**: باج‌افزارهای مختلف پَروَنجاهای مختلفی را مورد هدف قرار می‌دهند پس برای اینکه محیط اجرا را برای اکثر باج‌افزارها تحریک‌پذیر کنیم باید پَروَنجاهای عسل متنوع باشند.

پَروَنجاهای عسل در نظر گرفته‌شده بر روی جعبه‌شنی در این مقاله را می‌توان به ۴ دسته زیر تقسیم کرد که در جدول (۱) بیان‌شده است. این نوع پَروَنجاها بیشترین آسیب‌دیدگی را در مقابل حملات باج‌افزارها متحمل می‌شوند [۱۳].

پس برای این‌که بتوان از پَروَنجاهای عسل به‌منظور شناسایی باج-افزارها استفاده کرد بایستی این نوع از پَروَنجاها در مسیرهای مشخصی قرار گیرند تا با تحریک باج‌افزارها، بتوان رفتار تخریبی آن‌ها را رصد کرد.

– **جانمایی پَروَنجاهای عسل** ازآنجاکه باج‌افزارها از روش‌های مختلفی برای پیمایش تمام پَروَنجاهای موجود بر روی دستگاه استفاده می‌کنند بایستی این پَروَنجاها چنان جانمایی شوند تا در



شکل (۱): فرآیند طرح پیشنهادی (در معماری پیشنهادی ابتدا پَروَنجا اجرایی بر روی جعبه‌شنی اجرا می‌شود و سپس بر اساس ویژگی‌های مطرح‌شده که از تعاملات باج‌افزار با فرمت پَروَنجا استخراج می‌شوند و اعمال آن بر مدل ساخته‌شده می‌توان خانواده باج‌افزار را شناسایی کرد.)

مدت‌زمان محدودِ تحلیل یعنی ۲۰ دقیقه اجرا برای هر باج‌افزار [۱۳]، بتوان رفتار بدخواهانه باج‌افزارها را به‌دست‌آورد.

جدول (۱): پَروَنجاهای هدف باج‌افزارها

دسته پَروَنجاها	پسوندهای موردنظر
مستندات	txt, doc(x), ppt(x), xls(x), pdf
لایسنس	key, pem, crt, cer
آرشیو	zip, rar
مدیا	jp(e)g, mp3, avi

ازجمله چالش‌هایی که در شناسایی باج‌افزار با استفاده از پَروَنجاهای عسل روبرو هستیم، می‌توان به موارد زیر اشاره کرد:

۱- باج‌افزارها قادر به شناسایی پَروَنجاهای عسل باشند که در این صورت اقدام به رمزگذاری آن‌ها نخواهند کرد. در این حالت باج‌افزار با شناسایی این پَروَنجاها به‌عنوان پَروَنجاهای دامی و قلابی آن‌ها را رمزگذاری نمی‌کند.

۲- پَروَنجاهای عسل، در زمان محدودی که برای تحلیل باج‌افزار اختصاص داده‌شده است، رمزگذاری نشوند. در این صورت تأثیر باج‌افزارها بر روی پَروَنجاها عسل غیرقابل رصد خواهد بود.

ازآنجاکه ما هیچ دسترسی به کد منبع[1] باج‌افزارها نداریم [۱۷] پس اطلاعی از توانایی شناسایی پَروَنجاهای عسل توسط باج-افزارها را در اختیار نداریم و حتی در صورت آگاهی از این امکان در باج‌افزار، چگونگی انجام این عمل توسط باج‌افزار هم برای ما مبهم می‌باشد. اما می‌توان با فرض داشتن چنین امکانی در باج-افزارها به‌صورت زیر عمل کرد:

۱- همان‌طور که در جدول (۱) مشاهده می‌شود پَروَنجاها متنوع می‌باشد برخی از باج‌افزارها پَروَنجاهایی با یک حجم کمتر از یک مقدار پیش‌فرض را رمزگذاری نمی‌کنند پس لازم است تا پَروَنجاهای عسل حجم‌های متنوعی داشته باشند.

۲- یکی دیگر از کارهایی که می‌توان انجام داد، استفاده از نام‌های یکتا و بامعنی برای پَروَنجاهای عسل می‌باشد. به این منظور پَروَنجاهای عسل نباید اسامی تکراری یا اسامی خاصی داشته باشند. در بسیاری از موارد باج‌افزارها می‌توانند با شناسایی پَروَنجاها با اسامی خاص و یا تکراری در محیط‌های تحلیل، رفتار خود را پنهان کنند.

اما چالش دیگری که همواره در استفاده از پَروَنجاهای عسل برای شناسایی باج‌افزارها وجود دارد، رمزگذاری و دست‌کاری پَروَنجاهای عسل می‌باشد. ازآنجاکه ما هیچ اطلاعی در مورد چگونگی پیمایش پَروَنجاها توسط باج‌افزارها نداریم امکان دارد در زمان محدود اجرای باج‌افزار بر روی جعبه‌شنی نتوان رفتار بدخواهانه آن‌ها را شاهد بود و درنتیجه باعث بروز منفی اشتباه شود.

ازآنجاکه نمی‌توان مطمئن بود که کدام پَروَنجاها جز اولین پَروَنجاهایی خواهند بود که توسط باج‌افزار رمزگذاری می‌شوند. اما مسیرها[2] به‌صورت الفبایی پردازش و پیمایش می‌شوند [۷] به این خاطر، معمولا C:\$Recycle.Bin در اکثر موارد جز اولین دایرکتوری‌هایی است که توسط باج‌افزار مورد حمله قرار می‌گیرد. لذا، ما در این پروژه پَروَنجاهایی عسل را در دو پَروَنجا با نام‌های C:\$$0Decoyfiles\ و C:\ZDecoyfiles\ جاگذاری کرده‌ایم. علت اینکه پَروَنجاهایی عسل را در دو پوشه قرار داده‌ایم این است که برخی از باج‌افزارها همانند باج‌افزار shadow پَروَنجاها را به‌صورت الفبایی از پایین به بالا (Z تا A) پیمایش می‌کنند.

۲-۲-۲- ثبت تمام فعالیت‌های فرمت پَروَنجا

در زمان اجرای باج‌افزار بر روی جعبه‌شنی تمامی فعالیت‌های فرمت پَروَنجا پَروَنجاهای اجرایی ثبت می‌شوند. روش‌های مختلفی می‌توان برای رصد فعالیت‌های فرمت پَروَنجا پَروَنجاهای اجرایی وجود دارد. به‌عنوان‌مثال، فعالیت‌های فرمت پَروَنجا می‌توانند با قلاب اندازی[3] مجموعه‌ای از توابع API[4] مرتبط با فرمت پَروَنجا یا فراخوانی‌های سیستم‌عاملی با استفاده از جدول SSDT[5] رصد شوند. اما این راه‌کارها می‌توانند معایب زیر را به همراه داشته باشد [۱۳]:

[1] Source code

[2] Directory
[3] Hooking
[4] Application Programming Interface
[5] System Service Descriptor Table

۱. باج‌افزارها می‌توانند از روش‌های رمزنگاری [1]منحصربه‌فرد خود به‌جای APIهای استاندارد برای گذشتن از قلاب اندازی API به‌منظور رمزگذاری پَروَنجاهای کاربر استفاده کنند.

۲.قلاب اندازی SSDT بر روی دستگاه‌های ۶۴ بیتی به علت KPP[2] نمی‌تواند انجام شود.

۳.بسیاری از توابع SSDT غیر مستند می‌باشند و در نسخه‌های مختلف ویندوز می‌توانند تغییر یابند.

لذا، روش‌های قلاب اندازی در این روش به‌منظور رصد فعالیت-های ورودی/ خروجی فرمت پَروَنجا از راه‌انداز مینی‌فیلتر[3] استفاده‌شده است. این روش راه‌کاری استاندارد مبتنی بر هسته سیستم‌عامل برای رصد فعالیت‌های فرمت پَروَنجا در نسخه‌های مختلف ویندوز می‌باشد [۱۳] در سیستم‌عامل ویندوز درخواست‌های ورودی/ خروجی به‌صورت بسته‌های درخواست ورودی/ خروجی (IRP) می‌باشند.

جدول (۲): فهرست ویژگی‌های در نظر گرفته‌شده برای روش پیشنهادی

ردیف	ویژگی
۱	درگاشت پَروَنجاها
۲	تعداد پَروَنجاهای اضافه‌شده
۳	دسترسی به VSSAdmin
۴	پَروَنجاهای اجرایی اضافه‌شده
۵	دسترسی به RecentFiles
۶	تغییر حجم پَروَنجاها
۷	نحوه رمزگذاری پَروَنجاها توسط باج‌افزار

۲-۲-۳- استخراج ویژگی‌ها

در معماری پیشنهادی، در زمان ثبت فعالیت‌های فرمت پَروَنجا توسط مینی‌فیلتر استخراج ویژگی‌های شناساگر باج‌افزار صورت می‌گیرد. ویژگی‌های در نظر گرفته‌شده را می‌توان به‌صورت جدول (۲) بیان کرد. درگاشت که میزان تصادفی بودن داده‌ها در پَروَنجا را نشانی‌دهی، تعداد پَروَنجاهای اضافه‌شده به مسیرهای پَروَنجاهای عسل، دسترسی به VSSAdmin جهت پاک‌سازی پَروَنجاهای پشتیبان[4]، پَروَنجاهای اجرایی اضافه‌شده توسط پَروَنجا اجرایی اولیه باج‌افزار جهت ادامه کار رمزگذاری پَروَنجاها توسط این پَروَنجاهای اجرایی، رمزگذاری آخرین پَروَنجاهای دست‌کاری شده توسط قربانی، تغییرات حجم پَروَنجاها پس از رمزگذاری و درنهایت نحوه رمزگذاری پَروَنجاها به‌صورت بازنویسی و یا ایجاد یک پَروَنجا جدید اشاره کرد. در این قسمت تمام ویژگی‌های ذکرشده در بالا موردبررسی قرار می‌گیرند.

- **درگاشت شانون:** درگاشت، اطلاعاتی در مورد عدم قطعیت داده ارائه می‌دهد. به این معنی که مقدار تصادفی بودن یک مجموعه داده مشخص را نمایش می‌دهد. هرچقدر داده تصادفی‌تر باشد در صورت محاسبه درگاشت، درگاشت آن مقدار زیادی را نشان خواهد داد [۸]. برخی از انواع داده‌ها مانند داده‌های رمزگذاری شده و فشرده‌شده[5] درگاشت بالایی دارند پس بنابراین، حمله باج‌افزار به پَروَنجاها همواره باعث افزایش درگاشت می‌شود زیرا که در حملات باج‌افزار، باج‌افزار پَروَنجا قربانی را خوانده و محتوای رمزگذاری شده را می‌نویسد. همین امر باعث افزایش درگاشت می‌شود. درگاشت شانون برای آرایه‌ای از بایت‌ها را می-توان در رابطه (۱) مشاهده کرد.

e

در این فرمول $P_{B_i} = \frac{F_i}{\text{totalbytes}}$ که F_i تعداد نمونه‌های بایت با مقدار i در آرایه را نشان می‌دهد. این فرمول مقداری بین ۰ تا ۸ را به‌عنوان خروجی می‌دهد که مقدار ۸ نشان می‌دهد که مقادیر بایت در آرایه دارای توزیع یکنواخت می‌باشد. مقادیر درگاشت پَروَنجاهای رمزگذاری شده بیشتر میل به مقدار ۸ را دارند [۸] به همین علت وقوع هر بایت در پَروَنجا رمزگذاری شده بایستی یک احتمال یکنواخت داشته باشد.

- **تعداد پَروَنجاهای اضافه‌شده:** شاید یکی از مهم‌ترین ویژگی‌هایی که باج‌افزارها را از دیگر بدافزارها متمایز می‌کند این است که باج‌افزار بعد از انجام مراحل رمزگذاری پَروَنجاهای کاربر،

[1] Cryptosystem
[2] Kernel path protection
[3] Minifilter driver
[4] File backup
[5] Compressed

به‌صورت کاملاً واضح قربانی را در جریان حمله قرار می‌دهد. یکی از روش‌های انجام این کار اضافه کردن پَروَنجاهای مختلف جهت آگاه‌سازی کاربر در مسیرهای مختلفی است که باج‌افزار برای انجام رمزگذاری به آن‌ها وارد می‌شود. به‌عنوان‌مثال، باج‌افزار satana بعد از ورود به یک مسیر و انجام کارهای رمزگذاری خود پَروَنجا متنی با نام !satana!.txt در هر مسیر واردشده، قرار می‌دهد که در این پَروَنجا متنی تمامی اطلاعات در رابطه با نحوه رمزگذاری پَروَنجاها و چگونگی رمزگشایی پَروَنجاها آمده است. (الزاماً مطالب موجود در این پَروَنجا متنی هم نمی‌تواند درست باشد).

دسترسی به VSSAdmin: بسیاری از باج‌افزارهای جدید سعی می‌کنند تمامی پَروَنجاهایی که در VSC[1] ذخیره می‌باشد را با استفاده از VSS[2] حذف کنند [۷] این ویژگی یکی از ویژگی‌هایی است که می‌توان گفت مختص باج‌افزارها می‌باشد. به این معنی که هیچ برنامه یا سرویس دیگری در تلاش برای حذف این حجم از اطلاعات از VSC نمی‌کند. به‌عنوان‌مثال، باج‌افزار Cerber از دستورات زیر برای حذف همه پَروَنجاها استفاده می‌کند:

C:\Windows\system۳۲\vssadmin.exe" delete shadows /all /quiet

دستور بالا تمامی پَروَنجاهای ذخیره‌شده در VSC را حذف می‌کند. در اثر اجرای این دستور پَروَنجا اجرایی vssadmin.exe به اجرا درمی‌آید.

پَروَنجاهای اجرایی اضافه‌شده: بسیاری از باج‌افزارها به‌محض اجراشدن، یک پَروَنجا اجرایی دیگری که می‌توان گفت تمام‌کارهای مخرب را آن پَروَنجا انجام می‌دهد در مسیرهایی مانند: \Downloads، \Temp یا مسیرهایی در %Appdata% قرار می‌دهد. به‌عنوان‌مثال باج‌افزار CTB Locker به‌محض اجراشدن، یک پَروَنجا اجرایی در مسیر c:\Users\alha\AppData\Local\Temp می‌سازد. سپس با اجرای پَروَنجا جدید اضافه‌شده تمام‌کارهای رمزنگاری توسط آن انجام می‌شود.

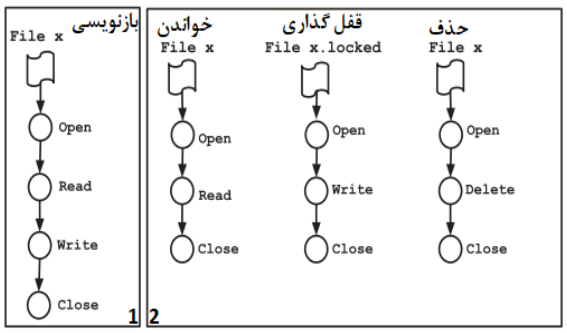
-دسترسی به Recent Files: همان‌طور که قبلاً هم اشاره شد، راهبرد حملات در باج‌افزارها می‌تواند متفاوت باشد. بدین معنی که برخی از باج‌افزارها سعی دارند تا پَروَنجاها را طبق شرایط مختلفی موردحمله قرار دهند. حال دراین‌بین باج‌افزارهایی هستند که پَروَنجاهای کاربر را بر طبق آخرین دسترسی، رمزنگاری می‌کنند. مانند باج‌افزار Cerber که پَروَنجاهای کاربر را بر طبق آخرین دسترسی‌ها رمزگذاری می‌کنند.

- **تغییر حجم پَروَنجاها:** باج‌افزارها پس از رمزگذاری بر روی پَروَنجاهای کاربر باعث می‌شوند تا حجم آن‌ها تغییر کند. این ویژگی هم می‌تواند یک ویژگی دیگر در شناسایی باج‌افزارها باشد.

- **نحوه رمزگذاری پَروَنجاها:** باج‌افزارها به‌منظور رمزگذاری پَروَنجاهای می‌توانند به دو صورت عمل کنند [۱۳]: ۱- بازنویسی روی پَروَنجا اصلی، ۲- ایجاد یک پَروَنجا جدید.

همان‌طور که در شکل (۲) حالت ۱، مشاهده می‌شود باج‌افزار پَروَنجاهای کاربر را با نسخه رمزگذاری شده پَروَنجا بازنویسی می‌کند. ازجمله این نوع باج افزارها می‌توان به Satana، CTB اشاره کرد.

اما در حالت ۲، باج افزار ابتدا یک پَروَنجا جدید می‌سازد، داده‌های پَروَنجا قربانی را خوانده و نسخه رمزگذاری شده پَروَنجا قربانی را تولید کرده و نسخه رمزگذاری شده را در پَروَنجا اصلی نوشته و درنهایت پَروَنجا اصلی کاربر را حذف می‌کند. مانند باج‌افزار Locky که به همین صورت عمل می‌کند.



شکل (۲): نحوه رمزگذاری پَروَنجاهای توسط باج‌افزارها
۱) بازنویسی پَروَنجا، ۲) ایجاد یک پَروَنجا جدید [۱۳]

- ایجاد مجموعه داده: بدیهی است که در فرآیند آموزش بایستی مجموعه داده‌ای برای آموزش حاضر شود. در این قسمت از تمامی ویژگی‌های استخراج‌شده از اجرای تمامی پَروَنجاهای اجرایی، ماتریسی ایجاد می‌شود.

[1] Volume Shadow Copy
[2] Volume Shadow Copy Service

۲-۲-۴- یادگیری دسته‌بند

در این مرحله با استفاده از دیتاست ایجادشده در مرحله پیشین دسته‌بندی ایجاد می‌شود. دسته‌بند در نظر گرفته‌شده در این روش برای یادگیری، دسته‌بند جنگل‌های تصادفی[1] می‌باشد. جنگل‌های تصادفی یک دسته‌بند ترکیبی[2] نظارت‌شده[3] است که شامل درختان تصمیم متعدد می‌باشد. خروجی این دسته‌بند با محاسبه مد[4] خروجی‌های به‌دست‌آمده از کل درختان تصمیم تولید می‌شود. یک ارتباط مستقیمی بین تعداد درختان جنگل و نتیجه آن وجود دارد: هرچقدر تعداد درختان زیاد باشد، نتیجه دقیق‌تر خواهد بود.

مزایای دسته‌بندِ جنگل‌های تصادفی:

۱- یکی از مشکلات اساسی در فرآیند آموزش مسئله بیش برازش[5] می‌باشد. اما در جنگل‌ها تصادفی در صورت کافی بودن تعداد درختان در جنگل، مشکل بیش برازش به وجود نمی‌آید.

۲- مقادیری که در دیتاست موجود نمی‌باشند را می‌تواند مدیریت کند.

۳- بر روی پایگاه داده‌های بسیار بزرگ به‌صورت مؤثر عمل می‌کند.

۲-۲-۵- فرآیند آزمون

همان‌طور که در شکل (۱) مشاهده می‌شود فرآیند آزمون و فرآیند آموزش تقریباً فرآیند مشابهی را طی می‌کنند به این منظور که در فرآیند آموزش هم ابتدا می‌بایست نمونه پَروَنجاهای اجرایی آزمون بر روی جعبه‌شنی به اجرا درآیند در هنگام اجرای این نمونه‌ها می‌بایست فعالیت‌های فرمت پَروَنجا به کمک مینی‌فیلتر ثبت‌شده و در گام بعد سعی می‌شود تمام ویژگی‌های شناساگر باج‌افزار از گزارش‌های فعالیت‌های فرمت پَروَنجا استخراج شوند بعدازاین مرحله، ویژگی‌های استخراج‌شده بر دسته‌بند جنگل تصادفی آموزش‌دیده در مرحله آموزش اعمال می‌شود. که خروجی این عمل برچسب خانواده آن نمونه خواهد بود.

[1] Random forests classifier
[2] Ensemble
[3] Supervised
[4] Mode
[5] Overfitting

۳- نتایج و بحث

برای ارزیابی این روش از نمونه باج‌افزارهای موجود در [۱۱] استفاده‌شده است. مشخصات نمونه‌ها در جدول (۳) نمایش داده‌شده است. به‌منظور مقایسه رفتار باج‌افزارها با برنامه‌های بی‌آزار، شصت‌وشش برنامه بی‌آزار هم به‌منظور رصد فعالیت‌های فرمت پَروَنجا به اجرا درآمدند.

جدول (۳): مشخصات باج‌افزارها

خانواده باج‌افزار	تعداد
CryptoWall	۱۵۷ (۴۱٪)
Crowti	۱۲۵ (۳۲/۶٪)
CrytptoDefense	۷۷ (۲۰/۱٪)
Critroni	۱۴ (۳/۷٪)
TeslaCrypt	۱۰ (۲/۶٪)
مجموع تعداد نمونه‌ها ۳۸۳ عدد می‌باشد.	

در این ویندوز ۸ به‌عنوان سیستم‌عامل جعبه‌شنی به‌کاربرده شده است. تعداد انواع مختلف پَروَنجاهای عسل استفاده‌شده برای این روش هم در جدول (۴) آمده است:

جدول (۴): تعداد پَروَنجاهای عسل

دسته پَروَنجاها	تعداد
مستندات	۲۱
لایسنس	۹
آرشیو	۴
مدیا	۱۵
مجموع تعداد پَروَنجاهای عسل ۴۹ عدد می‌باشد.	

این تعداد پَروَنجا همان‌طور که قبلاً هم اشاره شد در دو مسیر قرار می‌گیرند. هرکدام از باج‌افزارها به مدت ۲۰ دقیقه بر روی جعبه‌شنی اجرا می‌شود. در این مدت تمامی ویژگی‌های اشاره‌شده استخراج می‌شوند. برای ویژگی درگاشت از اختلاف درگاشت پَروَنجاهای عسل استفاده‌شده است. به این معنی که ابتدا درگاشت پَروَنجاها قبل از اجرای باج‌افزار محاسبه‌شده سپس اقدام به اجرای باج‌افزار می‌شود و بعد از مدت ۲۰ دقیقه دوباره درگاشت پَروَنجاهای عسل محاسبه می‌شود درنهایت اختلاف این دو درگاشت به‌عنوان ویژگی در نظر گرفته می‌شود. برای ویژگی تعداد پَروَنجاهای اضافه‌شده هم از اختلاف تعداد پَروَنجاهای قبل

و بعد از اجرای باج‌افزار استفاده‌شده است. دیگر ویژگی‌های مطرح‌شده هم با یک مقدار دودویی مقداردهی می‌شوند (۰یا ۱). پس از جمع‌آوری یک مجموعه داده از تمامی باج‌افزارها اقدام به ایجاد یک ماتریس جهت آموزش دسته‌بند جنگل‌های تصادفی شده است. برای آموزش دسته‌بند از ابزار وِکا[1] استفاده‌شده است. خروجی وکا برای مجموعه داده جمع‌آوری‌شده برای این روش در جدول (۵) ارائه‌شده است. این در حالی است که دقت تشخیص در [۱۱] برابر با ۹۷,۷ درصد می‌باشد.

جدول (۵): میزان دقت روش پیشنهادی

Accuracy	٪۹۸/۸۸
F-Measure	٪۰/۹۸۹
Precission	٪۰/۹۸۹
Recall	٪۰/۹۸۹

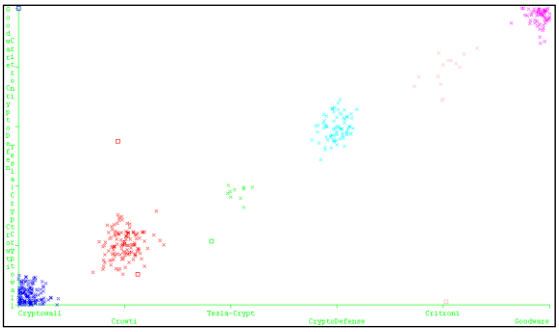
ماتریس درهم‌ریختگی[2] برای این روش به‌صورت جدول (۶) می‌باشد.

جدول (۶): ماتریس درهم‌ریختگی برای روش پیشنهادی

	f	e	d	c	b	a
a= C-wall	۱	۰	۰	۰	۰	۱۵۵
b=Crowti	۰	۰	۱	۰	۱۲۴	۱
c=Tesla-C	۰	۰	۰	۱۰	۱	۰
d=C-Def	۰	۰	۷۶	۰	۰	۰
e=Critroni	۰	۱۴	۰	۰	۰	۱
f=GW	۶۵	۰	۰	۰	۰	۰

شکل (۳) نمودار دسته‌بند جنگل‌های تصادفی را نشان می‌دهد. همان‌طور که در این نمودار مشاهده می‌شود نمونه‌های اجراشده با دقت بالایی (٪۹۸,۸) دسته‌بندی‌شده‌اند.

پس می‌توان نتیجه گرفت که این روش می‌تواند نمونه‌های مختلفی را در خانواده‌های خود دسته‌بندی نماید.



شکل (۳): نمودار دسته‌بند

۴- نتیجه‌گیری

امروزه، سخت‌افزارها و نرم‌افزارهای رایانه‌ای به‌شدت در حال رشد هستند. همچنین، نرم‌افزارهای بدخواه یک تهدید مهم در این حوزه به‌حساب می‌آیند. کاربران بدخواه برای رسیدن به هدف بدخواهانه خود، تعداد بدافزارهای هم‌ریخت را افزایش می‌دهند تا از شناسایی توسط نرم‌افزارهای امنیتی رهایی یابند. در همین راستا، روزانه هزاران بدافزار با روش‌های تحلیل متفاوت، شناسایی می‌شوند که از این تعداد حدود ۹۰٪ آن‌ها بدافزارهای هم‌ریخت هستند. به گفته محققین اکثر بدافزارها، جزء تعداد محدودی از خانواده‌های بدافزار هستند و یکی از مسائل روز تشخیص خانواده بدافزار است. در روش پیشنهادی جهت شناسایی خانواده باج-افزارها از ویژگی‌های استخراج‌شده از فعالیت‌های فرمت پَروَنجا همچون درگاشت پَروَنجاها، پَروَنجاهای اضافه‌شده در مسیر پَروَنجاهای عسل، دسترسی به VSSAdmin جهت پاک‌سازی پَروَنجاهای پشتیبان، پَروَنجاها اجرایی اضافه‌شده، رمزگذاری آخرین پَروَنجاهای دستکاری‌شده توسط کاربر، تغییرات حجمی ایجادشده در پَروَنجاهای عسل در اثر اجرای باج‌افزار و درنهایت نحوه رمزگذاری پَروَنجاها استفاده‌شده است. روش پیشنهادی همان‌طور که نشان داده شد توانست با دقت بالای ٪۹۸/۸ خانواده باج‌افزارهای اجراشده را به‌درستی شناسایی کند که در مقایسه با روش پیشین از دقت بیشتری برخوردار است.

[1] Weka: نرم‌افزاری که شامل مجموعه‌ای از الگوریتم‌های یادگیری ماشینی و داده‌کاوی می‌باشد. این نرم‌افزار توسط دانشگاه وایکاتو توسعه داده‌شده و برای تحلیل داده‌های عظیم کاربرد دارد.

[2] Confusion matrix

۵- مراجع

[1] A. B. Razak, M. Faizal, N. B. Anuar, R. Salleh, and Ahmad Firdaus, “The rise of “malware”: Bibliometric analysis of malware study.” Journal of Network and Computer Applications, vol. 75, pp. 58-76, 2016.

[2] D. D. Zovi, C. Eagle, I. Guilfanov, S. Porst, D. Quist, and P. Engbretson, “Practical Malware Analysis: The Hands-On Guide to Dissecting,” online: https://www.cise.ufl.edu/~jnw/MalwareReverseEngineeringSyllabus.pdf .

[3] S. Song, B. Kim, and S. Lee, “The Effective Ransomware Prevention Technique Using Process Monitoring on Android Platform,” Mobile Information Systems, 2016.

[4] B. A. S. Al-rimy and M. A. Maarof, “A 0-Day Aware Crypto-Ransomware Early Behavioral Detection Framework,” In International Conference of Reliable Information and Communication Technology, pp. 758-766, 2017.

[5] I. Security and T. Report, “Ransomware 2017,” 2017.

[6] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, “Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection,” arXiv preprint arXiv:1609.03020, 2016

[7] A. Liska and T. Gallo, “Ransomware: Defending Against Digital Extortion,” O'Reilly Media, Inc., 2016.

[8] H. Carter, P. Traynor, and K. R. B. Butler, “CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data,” IEEE 36th International Conference on Distributed Computing Systems (ICDCS), pp. 303-312, 2016.

[9] M. Lindorfer, C. Kolbitsch, and P. Milani Comparetti, “Detecting environment-sensitive malware,” In International Workshop on Recent Advances in Intrusion Detection, Springer, Berlin, Heidelberg, pp. 338–357, 2011.

[10] N. Idika, “A Survey of Malware Detection Techniques,” Purdue University, vol. 48, 2007.

[11] A. Continella, A. Guagnelli, G. Zingaro, G. De Pasquale, A. Barenghi, S. Zanero, and F. Maggi, “ShieldFS: a self-healing, ransomware-aware filesystem,” In Proceedings of the 32nd Annual Conference on Computer Security Applications, pp. 336-347, ACM, 2016.

[12] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda. “Cutting the gordian knot: A look under the hood of ransomware attacks,” In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, Cham, pp. 3-24, 2015.

[13] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, “{UNVEIL}: A Large-Scale, Automated Approach to Detecting Ransomware,” In 25th {USENIX} Security Symposium ({USENIX} Security 16), pp. 757-772. 2016.

[14] N. Andronio, S. Zanero, and F. Maggi, “Heldroid: Dissecting and detecting mobile ransomware,” In International Symposium on Recent Advances in Intrusion Detection, Springer, Cham, pp. 382-404, 2015.

[15] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, “Baiting inside attackers using decoy documents,” In International Conference on Security and Privacy in Communication Systems, Springer, Berlin, Heidelberg, pp. 51-70, 2009.

[16] J. Yuill, M. Zappe, D. Denning, and F. Feer, “Honeyfiles: deceptive files for intrusion detection,” In Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, IEEE, pp. 116-122, 2004.

[17] P. M. Comparetti, G. Salvaneschi, E. Kirda, C. Kolbitsch, C. Kruegel, and S. Zanero, “Identifying dormant functionality in malware programs,” In 2010 IEEE Symposium on Security and Privacy, IEEE, pp. 61-76, 2010.