شناسایی بدافزارهای فراریخت با ترکیب تحلیل ایستا و پویا

هادی گلباغی[۱]، مجتبی وحیدی اصل[*۲] و علیرضا خلیلیان[۳]

[۱] مرکز آپای دانشگاه کردستان
h.golbaghi@uok.ac.ir

[۲] دانشکده مهندسی و علوم کامپیوتر، دانشگاه شهید بهشتی
mo_vahidi@sbu.ac.ir

[۳] گروه مهندسی نرم‌افزار، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان
khalilian@eng.ui.ac.ir

چکیده

بدافزارنویسان از فنون متعددی استفاده می‌کنند تا روش کشف نرم‌افزارهای ضد بدافزار را خنثی کنند. یکی از این روش‌های مؤثر، فراریخت‌کردن بدافزار با فنون مبهم‌سازی است. فراریختی ساختار کد را آن‌چنان تغییر می‌دهد که ضمن حفظ رفتار بدافزار، ساختار و الگوی کد آن عوض شود. پژوهش‌گران به‌تازگی روشی برای کشف بدافزارهای فراریخت پیشنهاد کرده‌اند که بر اساس تحلیل ایستای کد بدافزار کار می‌کند. مسئله اینجاست که کاربست بعضی از فنون مبهم‌سازی، اثربخشی تحلیل‌های ایستا را در کشف بدافزار فرایخت کم می‌کند. برای غلبه بر این مشکل، مقالهٔ حاضر علاوه بر تحلیل ایستا، تحلیل پویایی نیز روی بدافزار انجام می‌دهد. روش جدید، اطلاعاتی از تحلیل ایستا و تحلیل پویا استخراج و سپس این دو گونه اطلاع را با هم ترکیب می‌کند و حاصل برای آموزش یک دسته‌بند مورد استفاده قرار می‌گیرد. دسته‌بند حاصل برای شناسایی نمونهٔ فراریخت‌شده جدیدی از یک خانوادهٔ بدافزار مورد استفاده قرار می‌گیرد. درحقیقت، ترکیب اطلاعات حاصل از تحلیل ایستا و پویا سعی می‌کند بر نقاط ضعف هر کدام غلبه کند و درمجموع اثربخشی بهتری داشته باشد. به‌منظور ارزیابی روش پیشنهادی، آزمایش‌هایی بر روی ۴۵۰ فایل متشکل از فایل‌های سالم و پنج خانواده بدافزار فراریخت از ویروس‌ها و کرم‌های G2, MPCGEN, MWOR, NGVCK, VLC انجام شده است. آزمایش‌ها در سه حالت انجام شده‌اند: تحلیل ایستا، تحلیل پویا و ترکیب آن‌دو. نتایج مقایسه نشان می‌دهد که شناسایی بر پایه فقط تحلیل ایستا یا پویا اغلب با دقت صددرصد انجام نمی‌شود. با این حال، کشف بدافزار فراریخت با ترکیب اطلاعات حاصل از تحلیل ایستا و پویا به‌طور سازگار توانسته به دقت کشف صددرصدی دست پیدا کند که با معیار ROC اندازه‌گیری شده است.

واژگان کلیدی: بدافزار، فراریختی، مبهم‌سازی کد، تحلیل ایستا، تحلیل پویا

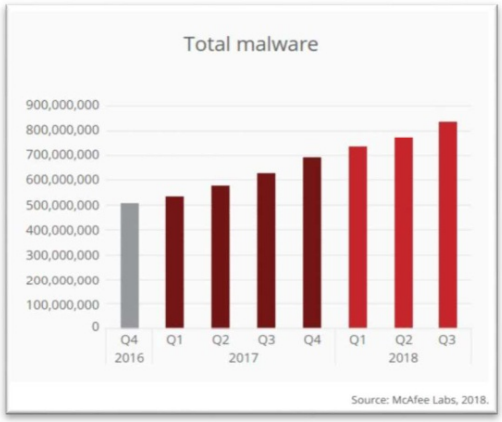
۱- مقدّمه

بدافزارها برنامه‌های رایانه‌ای هستند که هدف آن‌ها آسیب‌رسانی به سیستم‌های کامپیوتری است [۱، ۲]. بدافزارها با عملیات خراب‌کارانه، به منابع و اطلاعات سیستم‌ها دسترسی پیدا کرده و آن‌ها را تخریب می‌کنند. علاوه‌براین، می‌توانند به خدمات‌دهی و فعالیت‌های عادی سیستم‌ها آسیب وارد کنند و باعث سرقت غیرمجاز اطلاعات شخصی افراد و افشای حریم خصوصی آن‌ها شوند [۳]. به همین دلیل، مسئلهٔ شناسایی بدافزارها در مقیاس دانشگاهی و صنعتی مورد توجه بسیار بوده است. به‌طور متوسط، در هر حادثهٔ جرایم رایانه‌ای ۱۹۷ دلار خسارت وارد می‌شود [۴]. در شکل (۱)، آمار مجموع تعداد بدافزارها و رشد آن‌ها از سه ماهه چهارم سال ۲۰۱۶ تا سه ماه سوم ۲۰۱۸ نشان داده شده است.

بر طبق گزارش آزمایشگاه کسپرسکی تعداد بدافزارهای مالی ثبت‌شده در ماه سپتامبر سال ۲۰۱۸ نسبت به ماه جولای سال ۲۰۱۸ رشدی دوازده درصدی داشته است [۵]. با توجه به این گزارش‌ها و روند رشد بدافزارها در حوزه‌های مختلف، می‌توان پی برد که به‌طوراصولی شناسایی بدافزارها کاری پیچیده و بسیار سخت است. علّت این است که نویسندگان بدافزار به‌صورت مداوم روش‌های جدیدی برای مقابله با روش‌های کشف موجود طراحی می‌کنند. روش‌های سنّتی شناسایی بدافزار از روش‌های مبتنی بر امضا برای کشف بدافزارها استفاده می‌کردند؛ اما در حال حاضر پیچیدگی و

* نویسندهٔ مسؤول

تنوع بدافزارها روزبه‌روز در حال افزایش بوده و دسته‌ای از بدافزارها به نام بدافزارهای فراریخت با تغییر ساختار و امضا تکثیر می‌شوند که این باعث شده است که پژوهش‌گران تمرکز خود را بر روی معرفی روش‌هایی مؤثر برای شناسایی این گونه بدافزارها بگذارند [۲۲].



(شکل-۱): مجموع بدافزارهادر پایگاه داده آزمایشگاه مک‌آفی[۶]

فراریخت‌کردن بدافزار یکی از روش‌های مؤثر برای جلوگیری از کشف بدافزار بوده است [۷]. فراریختی ساختار داخلی کد بدافزار را با روش‌های مبهم‌سازی کد، آن‌چنان تغییر می‌دهد که ضمن تغییر ظاهر و امضای کد، همچنان رفتار بدافزار حفظ شود [۸]. بدافزارهای فراریخت به‌صورت عادی سعی در بازنویسی کد خود در هر انتشار داشته و با این کار هم مانایی خود را در محیط هدف تضمین کرده و هم روند عدم شناسایی خود را در محیط آلوده افزایش داده است [۲۰].

تاکنون روش‌های مختلفی برای شناسایی بدافزارهای فراریختی پیشنهاد شده‌اند. در یک دسته‌بندی کلی می‌توان روش‌های کشف بدافزار را به دو دسته ایستا و پویا تقسیم کرد. روش‌های ایستا محبوب‌تر و رایج‌ترند؛ زیرا بدون نیاز به اجرای بدافزار و تنها با تحلیل ساختار کد بدافزار، اقدام به کشف آن‌ها می‌کنند. درمقابل، روش‌های پویا با اجرای بدافزار و جمع‌آوری اطلاعاتی از اجرای بدافزار سعی می‌کنند، بدافزار را شناسایی کنند [۹].

نویسندگان این مقاله در مقالهٔ دیگری در همین اواخر روشی پیشنهاد کرده‌اند [۱۸] که بر مبنای جمع‌آوری اطلاعاتی حاصل از تحلیل ایستای کد بدافزار کار می‌کند. این روش اطلاعات به‌دست آمده را در قالب چهار معیار مناسب خلاصه و سپس با استفاده از دسته‌بند مناسب برای شناسایی گونهٔ جدید فراریخت از یک بدافزار شناخته‌شده اقدام می‌کند.

اگرچه روش پیشنهادی در آزمایش‌ها سودمندی و استحکام خود را نشان داده است، ولی شرایطی هست که اثربخشی هر روش مبتنی بر تحلیل ایستا از جمله روش پیشنهادی قبلی نویسندگان [۱۸] را می‌تواند تضعیف کند. برای نمونه، روش‌های جدیدتری از مبهم‌سازی کد می‌توانند استفاده شوند، طوری‌که ساختار کد نیز با توجه به معیارهای مورد محاسبه تغییر محسوسی کرده و روش کشف را کم‌اثر کند. به‌همین دلیل گاهی پیشنهاد می‌شود که بدافزار در محیط مجازی هم‌چون جعبهٔ شنی اجرا شود تا رفتار واقعی آن پایش شده و اطلاعات حاصل از آن برای شناسایی بدافزار مورد استفاده قرار گیرد. این موضوع انگیزه گسترش روش قبلی بوده که به آن اطلاعات حاصل از تحلیل پویا نیز اضافه شده است.

روش پیشنهادی در این مقاله ابتدا با تحلیل ایستا و تکیه بر تفاوت نرخ تکرار کدهای عملیاتی و ثبات‌ها در خانواده‌های مختلف بدافزارِ فراریخت و فایل‌های سالم، اطلاعاتی را جمع‌آوری می‌کند؛ سپس، در مرحلهٔ تحلیل پویا با پایش اجرای برنامه و پشته زمان اجرا، میزان دسترسی به حافظه اصلی و مقادیر ثبات‌ها در زمان‌های مختلف را نیز بررسی و چهار معیار نیز محاسبه می‌کند. کشف گونهٔ جدید بدافزار فراریخت و جداکردن آن از فایل‌های سالم، با استفاده از هشت معیار استخراج شده و بهره‌گیری از دسته‌بند مناسب صورت می‌گیرد.

روش پیشنهادی در مرحله ایستا با تأکید بر روی پالایش و انتخاب صحیح کدهای عملیاتی در صدد افزایش کارایی است و در شمارش کدهای عملیاتی، با همسان در نظر گرفتن دستورهای مشابه، سعی دارد در مقابل روش مبهم‌سازیِ جانشینیِ دستورهای مشابه، مقاوم باشد. همچنین با ترکیب این اطلاعات حاصل از تحلیل ایستا با اطلاعات حاصل از تحلیل پویا به‌دنبال رسیدن به شناسایی با دقت بالاتر است. هنگام تحلیل پویا به‌طورمعمول بدافزار رفتار مخرب واقعی خود را نشان می‌دهد و این، مزیت بزرگی برای کشف آن است. همچنین اگر بدنه بدافزار، رمزنگاری یا بسته‌بندی‌شده باشد، در موقع اجرا باید بدنهٔ بدافزار رمزگشایی و باز شود و این‌کار جمع‌آوری اطلاعات درست و واقعی از بدافزار را میسر می‌سازد.

برای نشان‌دادن عملکرد روش در موقعیت‌های عملی و ارزیابی آن آزمایش‌هایی بر روی ۴۵۰ فایل متشکل از فایل‌های سالم و پنج خانواده بدافزار فراریخت از ویروس‌ها و کرم‌های *G2, MPCGEN, MWOR, NGVCK, VLC* انجام شده‌اند.

نتایج در آزمایش‌ها در سه حالت گزارش شده‌اند: کشف با اطلاعات ایستا، کشف با اطلاعات پویا و کشف با ترکیب این دو دسته اطلاعات. دقت کشف با کمک اطلاعات ایستا بالاست؛ ولی اغلب اوقات صد درصد نمی‌شود؛ اما با ترکیب ایستا و پویا، روش پیشنهادی بهبود یافته و دقت صددرصدی بر مبنای معیار ROC به دست می‌آید.

دستاوردهای این مقاله در قالب نتایج آزمایش‌ها، شواهد مناسبی برای طراحی روش‌های جدیدتر کشف بدافزار در اختیار پژوهش‌گران قرار می‌دهد. همچنین گواه دیگری بر اثربخشی روش‌های ترکیبی ارائه می‌کند. علاوه‌بر این‌ها، سادگی نسبی روش پیشنهادی بهبودیافته باعث می‌شود، بتوان آن را در توسعهٔ محصولات تجاری به‌کار گرفت. نوآوری‌های این مقاله عبارتند از:

- ترکیب و تقویت روش کشف ایستای بدافزار فراریخت با اطلاعاتی حاصل از تحلیل پویا؛
- نتایج آزمایش‌هایی از روش پیشنهادی بهبودیافته روی چندین خانواده بدافزار فراریخت محک و مقایسه عملکرد روش کشف بدافزار در سه حالت استفاده از اطلاعات ایستا، پویا و ترکیب این دو.

ساختار ادامهٔ مقاله به این شرح است: در بخش دوم روش پیشنهادی بهبودیافته به‌طور کامل تشریح می‌شود. بخش سوم به ارزیابی و تفسیر نتایج اختصاص دارد. در بخش چهارم، برخی از مهم‌ترین کارهای مرتبط مورد بررسی قرار می‌گیرد. درنهایت بخش پنجم نیز به نتیجه‌گیری اختصاص دارد.

۲- روش پیشنهادی

بررسی روش‌های موجود در حوزه شناسایی بدافزار فراریخت و بررسی نقاط قوت و ضعف آن‌ها، نشان می‌دهد که موتورهای فراریختی همه چیز را در ساختار کد بدافزار تغییر نمی‌دهند. عملیات بر روی تعداد بسیاری از ثبات‌ها عوض نمی‌شود که درنتیجه محتوای برخی از ثبات‌ها به‌هیچ‌وجه تغییر نمی‌کنند؛ و بسیاری از کدهای عملیاتی در بدنه کد نیز بلاتغییر می‌مانند. در ارتباط با جریان اجرایی بدافزار، این موتورها به‌طورمعمول روالی مشترک را دنبال می‌کنند و فعالیت‌ها دارای وجه‌اشتراک زیادی هستند. این بینش در ساختار بدافزارهای فراریخت، انگیزهٔ طراحی روش جدیدی برای کشف بدافزارهای فراریخت با ترکیب روش‌های ایستا و پویا بوده است.

۲-۱- مرحلهٔ تحلیل ایستا

در مرحلهٔ شناسایی مبتنی بر تحلیل ایستا فرض می‌شود که معیار مشترکی میان بسیاری از موتورهای فراریخت وجود دارد؛ به‌گونه‌ای که در بدنهٔ ویروس‌ها تعداد زیادی از ثبات‌ها، دستورالعمل‌ها یا کدهای عملیاتی تکرار می‌شوند. این معیار می‌تواند پایه و اساسی برای تمایز بین خانواده‌های مختلف بدافزارها باشد. روش پیشنهادی در مرحله ایستا مبتنی بر تحلیل نرخ تکرار دستورهای برنامه یا همان کدهای عملیاتی و ثبات‌ها است؛ سپس براساس دسته‌بندی نرخ تکرارها، فایل‌های سالم و خانواده‌های مختلف بدافزارهای فراریخت را طبقه‌بندی می‌کند. ترتیب انجام فعالیت‌ها در این مرحله به صورت الگوریتم (۱) در شکل (۲) است. در ابتدا پیش‌پردازش انجام شده و معیارهای n ، m و k محاسبه می‌شوند. روند کلی روش پیشنهادی برای شناسایی بدافزارهای فراریخت در الگوریتم (۲) در شکل (۳) آمده است.

الگوریتم ۱: مرحله پیش‌پردازش
ورودی: فایل مورد بررسی
خروجی: کدهای عملیاتی و ثبات‌های شمارش شده
۱.برگردان فایل مورد بررسی به کد اسمبلی.
۲.استانداردسازی کد اسمبلی فایل مورد بررسی.
۳.شمارش تمامی کدهای عملیاتی ریزپردازنده ۸۰۸۶.
۴.پالایش و انتخاب n کد عملیاتی و حذف m مورد که نرخ تکرار آن‌ها به حد آستانه تعیین شده k نرسیده باشد.
۵.هنجارسازی نرخ تکرارها.
۶.محاسبه چهار معیار روش پیشنهادی برای شناسایی و طبقه‌بندی فایل‌های سالم و خانواده بدافزارهای فراریخت.
۷.ذخیره در بانک اطلاعاتی مربوط به شمارش ثبات‌ها و کدهای عملیاتی خانواده‌های مختلف.

(شکل-۲): ترتیب انجام فعالیت‌ها در مرحلهٔ پیش پردازش مرحلهٔ ایستا

در روش پیشنهادی از معیارهای مشابه با معیارهای مقالهٔ [۸] که شناسایی بر روی کدهای عملیاتی بوده است، استفاده می‌شود. برای انتخاب کدهای عملیاتی، توجه به انتخاب حدّ آستانه مناسب و حذف تعداد بیش‌تری از کدهای عملیاتی که در روند شناسایی تأثیری ندارند، باعث افزایش کارایی روش پیشنهادی می‌شود. با انتخاب حد آستانه مناسب و حذف m مورد از کدهای عملیاتی، کدهای عملیاتی که در ساختار کد یک فایل، اهمیت بسیار کمی دارند حذف و پس از فرایند انتخاب و پالایش و حذف m مورد که تعداد تکرار آن‌ها کمتر از حد آستانه k است، n کد عملیاتی انتخاب می‌شوند. پس از پالایش کدهای عملیاتی، چون نرخ تکرار آن‌ها در فایل‌های مختلف بسیار متفاوت است، باید تعداد دستورها هنجارسازی شوند. برای هنجارسازی، از فرمول (۱) استفاده شده است.

الگوریتم ۲: شناسایی بدافزار در مرحله تحلیل ایستا
ورودی: فایل مورد بررسی
خروجی: اعمال معیارها و طبقه‌بندی فایل مورد بررسی به فایل سالم یا بدافزار
۱. تا مرحله ششم روال پیش‌پردازش نشان داده شده در شکل (۲)، مجدداً تکرار خواهد شد.
۲.محاسبه چهار معیار به‌صورت زیر:
معیار α) شمارش n کد عملیاتی انتخاب شده طبق حد آستانه k
معیار β) شمارش تمامی ثبات‌های ریزپردازنده ۸۰۸۶ به جز ثبات فلگ، که ۲۱ مورد هستند.
معیار γ) شمارش کدهای عملیاتی مجموعه A که نرخ تکرار بالایی در فایل‌های سالم داشته‌اند اما در بدافزارها وجود ندارند.
معیار δ) شمارش کدهای عملیاتی مجموعه B که نرخ تکرار بالایی در بدافزارها داشته‌اند اما در فایل‌های سالم وجود ندارند.
۳.رسم نمودار معیارهای محاسبه شده به‌صورت جداگانه.
۴.محاسبه مساحت زیر نمودار هر معیار به‌صورت جداگانه.
۵.طبقه‌بندی بر اساس نتایج به‌دست آمده از معیارها.
۶. تحلیل نتایج شناسایی.

(شکل-۳): روند کلی فعالیت‌ها پس از پیش پردازش مرحله ایستا

$$W = \frac{F}{T} \qquad (۱)$$

در فرمول (۱)، F نرخ تکرار محاسبه‌شدهٔ هر کد عملیاتی، T مجموع تکرار تمامی کدهای عملیاتی فایل مورد نظر و W حالت هنجارشدهٔ نرخ تکرار هر کد عملیاتی است. پس از هنجارسازی، چهار معیار اصلی برای شناسایی و طبقه‌بندی فایل‌ها در مرحله ایستا محاسبه می‌شوند. برای هر یک از چهار معیار، کدهای عملیاتی مختلف و ثبات‌هایی در نظر گرفته شده‌اند:

معیار α: کل n کد عملیاتیِ انتخاب‌شده.

معیار β: برای این معیار، ۲۱ ثبات در نظرگرفته شده‌اند.

معیار γ: این معیار به کدهای عملیاتی می‌پردازد که با توجه به نرخ تکرار آن‌ها و تحلیل‌هایی که بعد از شمارش تمامی فایل‌ها صورت گرفته است، نرخ تکرار بالایی در فایل‌های سالم دارند و در بدافزارها یا به‌هیچ‌وجه تکرار نشده‌اند و یا تعداد تکرارشان زیر سه بار است.

معیار δ: این معیار به کدهای عملیاتی می‌پردازد که با توجه به نرخ تکرار آن‌ها و تحلیل‌هایی که بعد از شمارش تمامی فایل‌ها صورت گرفته است، نرخ تکرار بالایی در بدافزارها دارند و در فایل‌های سالم یا به‌هیچ‌وجه تکرار نشده‌اند و یا تعداد تکرارشان زیر سه بار است.

سپس نمودار خطی تکرار کدهای عملیاتی رسم می‌شود که نمودار Yها مربوط به ثبات‌ها یا کدهای عملیاتی و نمودار Yها به تعداد تکرار هر ثبات یا کد عملیاتی اختصاص دارد. برای محاسبه چهار معیار، مساحت زیرنمودار مربوط به تکرار همان کدهای عملیاتی و ثبات‌ها محاسبه می‌شود. پس از شمارش کلیه فایل‌ها و محاسبه تمامی معیارها، اطلاعاتِ حاصل، مورد تحلیل و طبقه‌بندی قرار می‌گیرند.

۲-۲- مرحلهٔ تحلیل پویا

در روش پیشنهادی، ترکیب روش‌های ایستا و پویا مدنظر بوده است؛ دلیل، آن است که در مواردی بدافزارها بدنه کد خود را رمزنگاری می‌کنند که با توجه به این مورد، روش‌های شناسایی مبتنی بر تحلیل ایستا به‌طور کامل در مقابل این فنون دچار شکست می‌شوند. نکتهٔ کلیدی این است که حتی اگر بدافزاری رمزنگاری شده باشد، پس از اجرا در یک محیط جعبه شنی یا محیط مجازی امن، خود را رمزگشایی کرده و با استفاده از ابزار می‌توان به کد اصلی بدافزار و روند اجرایی آن دسترسی پیدا کرد. روش پیشنهادی در مرحله تحلیل پویا، با بررسی اجرای فایل مورد نظر در یک محیط امن، ردیابیِ اجرا و مدنظر قراردادن چهار معیار به طبقه‌بندی فایل‌ها می‌پردازد. چهار معیار در مرحله تحلیل پویا در الگوریتم (۳) در شکل (۴) نشان داده شده است. در این مرحله، چهار معیار نشان داده‌شده در شکل (۴)، در چهار زمان مختلف $T1$ تا $T4$ ثبت می‌شود.

الگوریتم ۳: مرحله تحلیل پویا در روش پیشنهادی
ورودی: فایل مورد بررسی
خروجی: محاسبه معیارهای چهارگانه مرحله تحلیل پویا
۱. شمارش تعداد عملیات انجام گرفته در پشته زمان اجرا.
۲. شمارش تعداد دسترسی‌های صورت گرفته به حافظه اصلی.
۳. شمارش تعداد متغیرهایی که در جریان اجرایی برنامه مورد استفاده قرار می‌گیرند.
۴. مقادیر ثبات‌ها در زمان‌های مختلف تعیین شده.

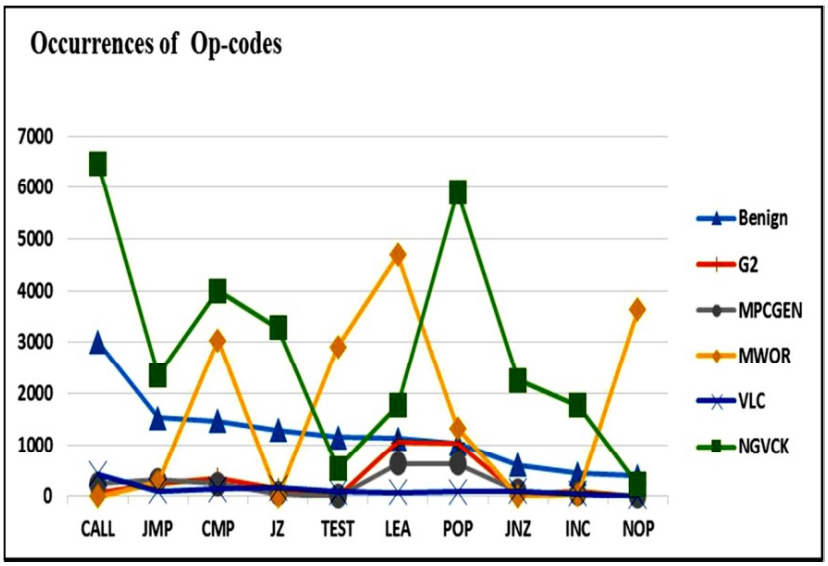
(شکل-۴):. چهار معیار شناسایی در مرحله تحلیل پویا

دلایل مختلفی برای انتخاب معیارهای چهارگانه در بخش پویا وجود داشته که به‌دلیل این‌که روش پیشنهادی دارای دو بخش ایستا و پویا است و درکل محاسبهٔ هشت معیار بایستی انجام شود، استفاده از معیارهایی که سربار زیادی برای روش پیشنهادی نداشته و کارایی روش را پایین نیاورد، چندین معیار در مرحلهٔ بررسی اجرای برنامه‌های مخرب مدنظر قرار گرفت که افزایش کارایی دلیل انتخاب این معیارها از بین چندین معیار بوده است. برای مثال، تعداد عملیات انجام‌گرفته

در پشتهٔ زمان اجرا، دارای مقادیر *S1* تا *S4*، که منطبق با زمان‌های ۳۰، ۶۰، ۹۰ و ۱۲۰ ثانیه است، ثبت می‌شود. برای معیارهای دیگر نیز به همین صورت این عمل تکرار می‌شود. طبق معیارهای در نظر گرفته‌شده، درنهایت طبقه‌بندی فایل‌ها به فایل سالم و بدافزار، انجام خواهد شد. در روش پیشنهادی اساس طبقه‌بندی و درنهایت شناسایی بر طبق چهار معیار تحلیل ایستا و چهار معیار تحلیل پویا است.

۳- ارزیابی

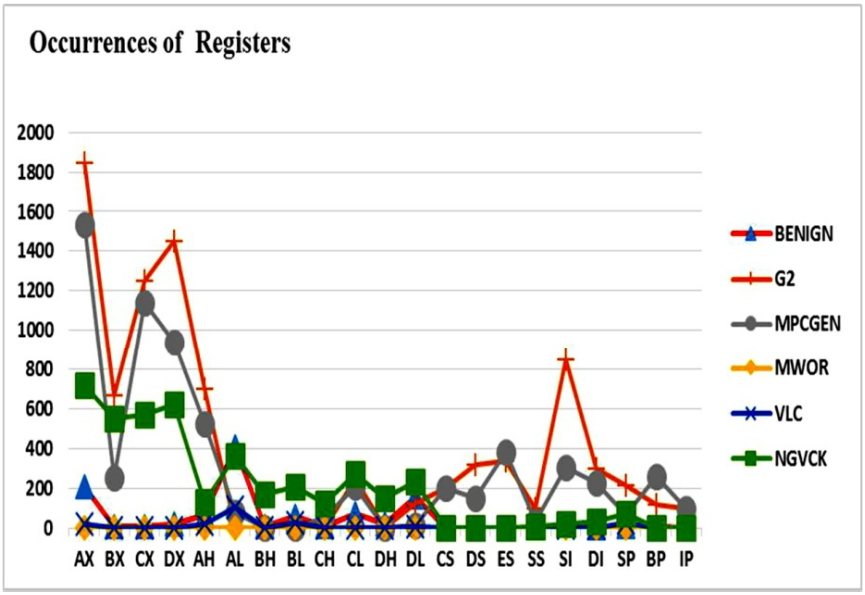
به‌منظور ارزیابی روش پیشنهادی، ۴۵۰ فایل از [۱۶،۲۱] استخراج شده که متشکل از چهل فایل سالم یا BENIGN که با استفاده از ابزار Cygwin تولید شده و ۴۱۰ فایل بدافزار فراریخت بوده است. بدافزارها از پنج خانواده بدافزارهای فراریخت G2, MPCGEN, MWOR, NGVCK, VLC تشکیل شده‌اند. در مرحله ایستا ابتدا همه ۴۵۰ فایل مورد بررسی، با استفاده از ابزار IDA PRO به کد اسمبلی برگردانده می‌شوند. در مرحله پویا نیز در محیط مجازی امن با استفاده از ابزار IDA PRO و EMU8086 روند اجرایی فایل مورد نظر نسبت به چهار معیار روش پیشنهادی مورد بررسی قرار گرفته است و در زمان‌های $T1$=30s، $T2$=60s ، $T3$=90s و $T4$=120s اطلاعات معیارهای نامبرده‌شده در بخش ۳-۲ ثبت می‌شوند. پایه و اساس روش پیشنهادی در مرحله ایستا در شمارش کدهای عملیاتی است که تعدادشان در ریزپردازنده ۸۰۸۶، ۱۹۱ مورد است. تمامی ۱۹۱ مورد برای تمامی ۴۵۰ فایل مورد بررسی، در مرحله پیش‌پردازش شمارش می‌شوند. با آزمایش‌ها و ارزیابی‌های صورت‌گرفته و مشاهده تأثیر انتخاب حد آستانه در کارایی روش پیشنهادی، حد آستانه مقدار k=۵ تعیین شد. با درنظرگرفتن k=۵ ، تعداد کدهای عملیاتی که پالایش و انتخاب شدند، n=۸۳ است و تعداد کدهای عملیاتی حذف‌شده نیز m=۱۰۷ است. در مقایسه تحلیل‌ها مشهود است که برخی از کدهای عملیاتی فقط در برنامه‌های سالم هستند و هرگز در بدافزارها وجود ندارند. بعضی کدهای عملیاتی و ثبات‌ها ممکن است، در بدافزارها وجود داشته باشند، اما هرگز در برنامه‌های سالم وجود ندارند. برخی از کدهای عملیاتی و ثبات‌ها، نه در بدافزارها وجود دارند و نه در برنامه‌های سالم که این موارد در شکل‌های (۵ و ۶) که شمارشی در مرحله پیش‌پردازش صورت گرفته مشهود و در ادامه، معیارهایی برای طبقه‌بندی‌ها در روش پیشنهادی بوده است.



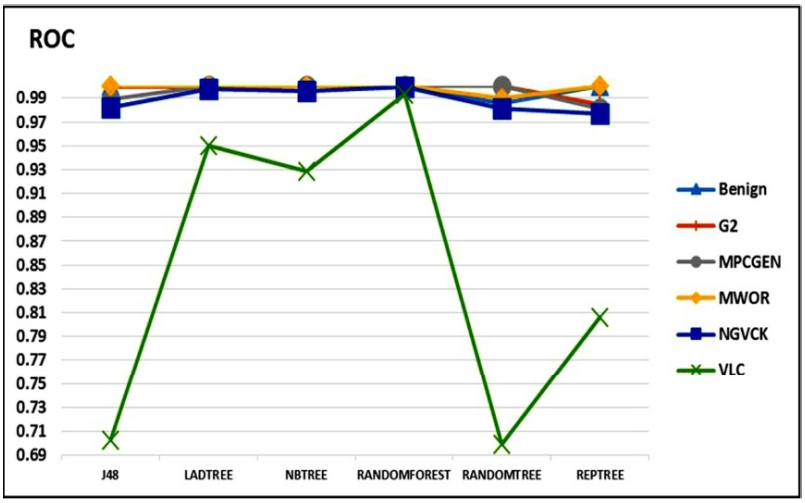
(شکل-۵): تفاوت در رخداد برخی از کدهای عملیاتی در خانواده بدافزارهای فراریخت و فایل‌های سالم در مرحله پیش‌پردازش

۳-۱- تحلیل نتایج

ابتدا تحلیل نتایج به‌صورت جداگانه صورت می‌پذیرد و سپس ترکیب دو روش پیشنهادی مورد ارزیابی قرار می‌گیرد. در ارزیابی و تحلیل نتایج معیار *FP* نرخ مثبت کاذب است. همچنین نمودار *ROC* جهت تخمین دقت دسته‌بندی استفاده می‌شود که بیشینه مقدار ممکن برای آن مقدار یک و کمترین مقدار آن صفر است. در شکل (۷) معیار *ROC* برای روش پیشنهادی در مرحله تحلیل ایستا و در شکل (۸) معیار *ROC* برای روش پیشنهادی در مرحله تحلیل پویا نشان داده شده است.



(شکل-۶):. تفاوت در رخداد ثبات‌های مختلف در خانواده‌های متفاوت بدافزارهای فراریخت و فایل‌های سالم در مرحله پیش‌پردازش



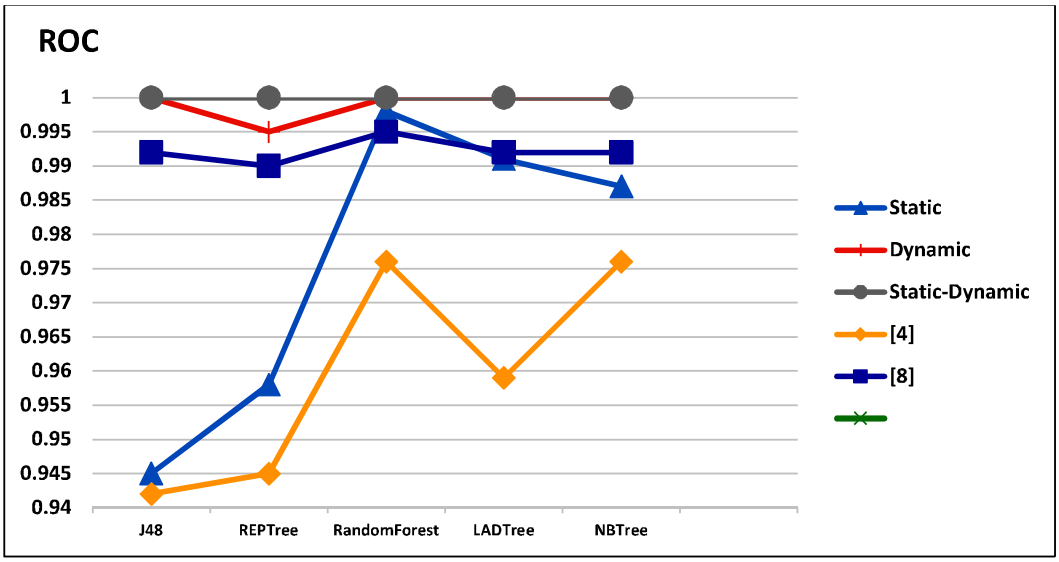
(شکل-۷): مقایسه میزان ROC برای پنج خانواده بدافزارهای فراریخت و فایل‌های سالم در مرحله تحلیل ایستا

تحلیل نتایج طبقه‌بندی‌ها در ترکیب روش‌های ایستا و پویا با استفاده از الگوریتم‌های دسته‌بندی مختلف در جدول (۱) نشان می‌دهد که روش پیشنهادی قادر است، طبقه‌بندی بین بدافزارهای فرایخت و فایل سالم را با دقت بسیار بالا انجام دهد. با توجه به شکل (۹)، مشاهده می‌شود که دقت شناسایی در مرحله‌های ایستا، پویا و ترکیب ایستا-پویا به‌صورت جداگانه به‌نسبه خوب و مقایسه‌ای با مقاله [۴] و [۸] انجام گرفته است. دلیل مقایسه با این دو مقاله شرایط یکسان نمونه بدافزارها و الگوریتم‌های مورد استفاده برای دسته‌بندها بوده است. با بررسی جدول (۱)، روش پیشنهادی که ترکیب روش‌های ایستا و پویا است، در الگوریتم‌های مختلف دسته‌بندی، نتایج $ROC=1$ و $FP=0$ برای آن ثبت شده است.

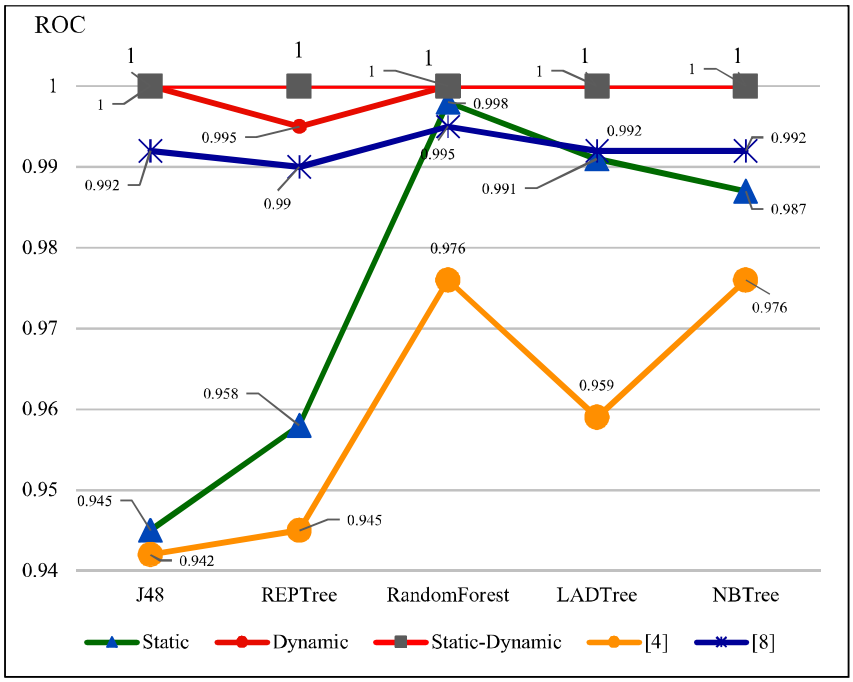
روش پیشنهادی دارای چند نقطه قوت اساسی است:

- در مقابل روش مبهم‌سازیِ جایگزینیِ دستورهای مشابه، که در روش‌های تحلیل ایستای قبلی باعث شکست روش‌های شناسایی می‌شد، مقاوم است و اثر آن را خنثی می‌کند و دچار افت در دقت شناسایی نمی‌شود. دلیل آن نیز این است که تمامی دستورات مشابه را در روند شمارش، یکسان در نظر می‌گیرد.

- روش پیشنهادی در مقابل مواردی که بدافزارها بدنهٔ کد خود را رمزنگاری می‌کنند و باعث شکست روش‌های ایستا می‌شوند، مقاوم است؛ به این دلیل که در مرحلهٔ پویا با اجرای فایل مورد نظر، هم به کد و هم به روند اجرایی فایل، دست‌یابی خواهد داشت.
- روش پیشنهادی دارای دقت شناسایی صددرصدی در اغلب الگوریتم‌های مختلف طبقه‌بندی بوده است.
- روش پیشنهادی هم در تحلیل ایستا و هم در تحلیل پویا معیارهای مختلفی را برای شناسایی مدنظر قرار می‌دهد که این مورد، روش را در مقابل برخی از فنون مبهم‌سازی مقاوم می‌کند. به‌عنوان مثال، روش‌های مبهم‌سازی کد که فقط بر روی ثبات‌ها کار می‌کنند، تنها بر معیار β در بخش ایستا تأثیرگذار هستند و در بقیه معیارها تأثیر نخواهند گذاشت که در روند کلی شناسایی تأثیر اندکی را خواهد گذاشت.



(شکل-۸): مقایسه میزان ROC برای پنج خانواده بدافزارهای فراریخت و فایل‌های سالم در مرحله تحلیل پویا



(شکل-۹): مقایسه جداگانه میانگین نتایج ROC برای مرحله‌های ایستا، پویا، ترکیب ایستا-پویا و مقاله [۴، ۸]

(جدول-۱): مقایسه نتایج روش پیشنهادی در مرحله‌های ایستا و پویا با حالت ترکیبی ایستا-پویا با الگوریتم‌های مختلف طبقه‌بندی

فایل‌ها	ترکیب ایستا-پویا		نتایج مرحله پویا		نتایج مرحله ایستا		الگوریتم
	ROC	FP	ROC	FP	ROC	FP	
BENIGN	۱	۰	۱	۰	۱	۰	J48
G2	۱	۰	۱	۰	۰/۹۹۹	۰/۰۰۳	
MPCGEN	۱	۰	۱	۰	۰/۹۹	۰/۰۰۳	
MWOR	۱	۰	۱	۰	۱	۰	
VLC	۱	۰	۱	۰	۰/۷۰۳	۰/۰۰۵	
NGVCK	۱	۰	۱	۰	۰/۹۸۱	۰/۰۲۵	
BENIGN	۱	۰	۰/۹۷۱	۰	۰/۹۸۱	۰/۰۰۲	Random Tree
G2	۱	۰	۱	۰	۱	۰	
MPCGEN	۱	۰	۰/۹۹۶	۰/۰۰۸	۱	۰	
MWOR	۱	۰	۱	۰	۰/۹۹	۰	
VLC	۱	۰	۱	۰	۰/۶۹۹	۰/۰۰۲	
NGVCK	۱	۰	۰/۹۵	۰	۰/۹۸۱	۰/۰۳۵	
BENIGN	۱	۰	۰/۹۸۱	۰	۱	۰	REP Tree
G2	۱	۰	۰/۹۹	۰/۰۰۳	۰/۹۸۱	۰/۰۰۵	
MPCGEN	۱	۰	۱	۰	۰/۹۸۱	۰/۰۰۳	
MWOR	۱	۰	۰/۹۹۵	۰/۰۰۳	۱	۰	
VLC	۱	۰	۱	۰	۰/۸۰۶	۰/۰۰۲	
NGVCK	۱	۰	۰/۹۹۶	۰/۰۰۵	۰/۹۷۱	۰/۰۵۵	
BENIGN	۱	۰	۱	۰	۱	۰	Random Forest
G2	۱	۰	۱	۰	۱	۰	
MPCGEN	۱	۰	۱	۰	۱	۰	
MWOR	۱	۰	۱	۰	۱	۰	
VLC	۱	۰	۱	۰	۰/۹۹۲	۰	
NGVCK	۱	۰	۱	۰	۰/۹۹۹	۰/۰۲۵	
BENIGN	۱	۰	۱	۰	۱	۰/۰۰۲	LAD Tree
G2	۱	۰	۱	۰	۰/۹۹۹	۰/۰۰۵	
MPCGEN	۱	۰	۱	۰	۱	۰/۰۰۳	
MWOR	۱	۰	۱	۰	۱	۰	
VLC	۱	۰۰۵ /۰	۱	۰/۰۰۵	۰/۹۵	۰/۰۰۲	
NGVCK	۱	۰	۱	۰	۰/۹۹۶	۰/۰۳	
BENIGN	۱	۰	۱	۰	۱	۰	NB Tree
G2	۱	۰	۱	۰	۱	۰	
MPCGEN	۱	۰	۱	۰	۱	۰/۰۰۳	
MWOR	۱	۰	۱	۰	۱	۰	
VLC	۱	۰	۱	۰	۰/۹۲۹	۰/۰۰۲	
NGVCK	۱	۰	۱	۰	۰/۹۹۶	۰/۰۳	

۴- کارهای مرتبط

در سالیان اخیر روش‌ها و فنون متفاوت و متعددی برای کشف بدافزارهای فراریخت پیشنهاد شده‌اند. روش‌های شناسایی موجود دارای نقاط قوت و ضعفی هستند. برخی از روش‌ها [۳، ۱۰] به‌دلیل سربار محاسباتی بالا، زمانی در مقابل بدافزارها مؤثر هستند که زمان کافی برای شناسایی داشته و ضدبدافزار و بانک اطلاعاتی آن، به نرخ سریعی به‌روزرسانی شوند؛ درغیراین صورت، تأثیر مثبت آن‌ها کاهش پیدا می‌کند [۱۱]. روش پیشنهادی ران‌وال و همکاران [۹] از جمله روش‌های اندازه‌گیری شباهت بر اساس گراف کدهای عملیاتی بوده، ولی کارآمدتر از روش ارائه‌شده توسط اندرسون و همکاران [۱۲] عمل می‌کند. این روش با دریافت یک فایل اجرایی، رشته کدهای عملیاتی را استخراج می‌کند و از روی آن گراف وزن‌دار را می‌سازد؛ اما به‌جای استفاده از هسته گراف‌ها، به‌طور مستقیم گراف‌های کد عملیاتی را مقایسه می‌کند. بر اساس نتایج، روش پیشنهادی دارای دقت به‌نسبه خوبی بوده و کارایی بهتری نسبت به روش‌های قبلی دارد. البته اگر از روش جایگزینیِ دستورهای مشابه استفاده شود، تشخیص فراریختی توسط روش پیشنهادی این مقاله دشوار خواهد بود.

روش پیشنهادی کانفورما و همکاران [۸] به معرفی فنون شناسایی تکیه کرده است که مبتنی بر فرض وجود یک اثر جانبی مشترک بین بسیاری از موتورهای فراریخت هستند. این روش برای حدود هزار برنامه، مورد آزمایش و بررسی قرار داده شده و بر اساس نتایج آن به‌طور دقیق ویروس‌های فراریخت و غیر فراریخت را دسته‌بندی کرده است. از معایب روش این است که اگر از روش جایگزینیِ دستورهای مشابه یا اضافه‌کردن کد زائد به کد بدافزار و یا رمزنگاری بدنه کد بدافزار استفاده شود، توزیع دستورهای تکراری تغییر می‌کند و روش پیشنهادی دچار شکست در شناسایی خواهد شد.

مارک استمپ و همکاران [۱۳] از روش‌های قبلی مبتنی بر بردار [۱۴] برای تشخیص فراریختی استفاده کرده‌اند. در این روش، بردارهای ویژهٔ حاصل از بایت‌های خام فایل‌های اجرایی متعلق به خانواده‌ای از ویروس‌های فراریخت تهیه و سپس از این بردارهای ویژه جهت امتیازدهی به فایل‌هایی متشکل از خانواده‌ای از ویروس‌ها و نیز فایل‌های سالم استفاده می‌شود. این روش قدرتمند است و برای کدهایی که درجه فراریختی آن‌ها بالاست و از روش‌های تشخیص آماری گریخته‌اند، مفید است. البته در مورد بدافزارهایی که بتوانند با حفظ آمار دستورها، شکل کد را عوض کنند، این روش با شکست روبه‌رو خواهد شد.

گامال محمد و نورافیدا بنتی در [۱۹] قالبی را پیشنهاد می‌کنند که منجر به ایجاد رویکردی جدید بر اساس روش‌های مبتنی بر امضا و مبتنی بر رفتار رشته‌محور برای بهبود شناسایی بدافزارهای فراریخت‌شده است. شناسایی با استفاده از مجموعه داده‌های استاندارد از نمونه بدافزارهای

شناخته‌شده به‌صورت فرمت رشته‌ای، توابع و پارامترهای مختلف انجام می‌شود. نتایج نشان می‌دهد که بخش‌های پر خطر کدها و فایل‌های بدافزار از قطعه کدهایی هستند که به همراه دستورهای سالم به کد اصلی تزریق شده‌اند. درنتیجه، این روش، شناسایی بدافزارهای ناشناخته را نیز تسهیل می‌کند و سرعت و دقت را با کاهش پیچیدگی محاسباتی در زمان تشخیص بدافزار و کاهش حافظه مصرفی افزایش می‌دهد.

رویکرد مهرا و همکاران [۱۵] بر روی شناسایی و طبقه‌بندی بدافزارهای فراریخت بر اساس خانواده‌های آن‌ها تمرکز دارد. روش پیشنهادی به این صورت است که گراف جریان کنترلی رسم شده و گراف فراخوانی APIها ایجاد می‌شود. این رویکرد هر بدافزار فراریخت را براساس ویژگی‌های خانوادهٔشان که از هیستوگرام و فرمول اندازه‌گیری کای دو، که بر اساس تحلیل پویا است، طبقه‌بندی می‌کنند. در این مقاله، دقت در الگوریتم‌های مختلف طبقه‌بندی از ۸۹ تا ۹۹/۱۰ درصد به‌دست آمده است.

به اعتقاد محمد بن خمس و همکاران [۱۷] هنوز هم دست‌یابی به دقت کامل و کارایی مناسب برای شناسایی بدافزارهای فراریخت یک چالش محسوب می‌شود. روش پیشنهادی این مقاله ویژگی‌های تغییر داده نشده در ساختار بدافزار را برای استفاده در فرایند شناسایی با استفاده از ماشین بردار پشتیبانی استخراج می‌کند. خصوصیات n-gram به‌طور مستقیم از ساختار دودویی بدافزار استخراج شده، که این خصوصیات به‌عنوان امضا در نظر گرفته می‌شوند. این خصوصیات مقادیر قابل توجهی از تعداد خصوصیات انتخاب n-gram در حالت اصلی را کاهش می‌دهد. این روش ترکیبی از استخراج امضای n-gram و ماشین بردار پشتیبانی است. نتایج ارزیابی‌های روش پیشنهادی برای شناسایی بدافزارهای فراریخت نشان می‌دهد که این روش قادر است، دقتی در حدود ۹۹ درصد و نرخ منفی کاذب پایینی داشته باشد.

۵- نتیجه‌گیری

روش پیشنهادی این مقاله در مرحله ایستا با شمارش و تحلیل ثبات‌ها و کدهای عملیاتی و در مرحله پویا با شمارش تعداد عملیات‌های انجام‌گرفته در پشتهٔ زمانِ اجرا، تعداد دسترسی‌ها به حافظهٔ اصلی، تعداد متغیرهای مورد استفاده در جریان اجرایی برنامه و مقادیر ثبات‌ها، به شناسایی بدافزارهای فراریخت می‌پردازد. یافته‌های آزمایش‌ها نمایان‌گر این است که با وجود سادگی، روش ارائه‌شده بسیار دقیق است. در روش پیشنهادی سربار محاسباتی بسیار کم است؛ ایدهٔ اساسی روش به‌سادگی قابل فهم و در ضدبدافزارها به‌آسانی قابل پیاده‌سازی است؛ در مقابل روش‌های جایگزینی دستورهای مشابه، رمزنگاری بدنه کد بدافزارها و فنون مختلف مبهم‌سازی با توجه به شناسایی بر مبنای معیارهای مختلف، مقاوم است و دارای کارایی مطلوبی از نظر حافظه مصرفی و سرعت شناسایی است. هم‌چنین در مقایسه با روش‌های مشابه موجود دارای بهبودهایی در ابعاد مختلف است.

۶- مراجع

[1] Baysa, D., Low, R. M., & Stamp, M.Structural entropy and metamorphic malware. Journal of computer virology and hacking techniques, 9(4), 179-192, 2013.

[2] Konstantinou, E., &Wolthusen, S. Metamorphic virus: Analysis and detection. Royal Holloway University of London, 15, 2008.

[3] Chen, L., Li, T., Abdulhayoglu, M., & Ye, Y. Intelligent malware detection based on file relation graphs. In Semantic Computing (ICSC), International Conference on (pp. 85-92). IEEE,2015.

[4] Canfora, G., Mercaldo, F., Visaggio, C. A., & Di Notte, P. Metamorphic malware detection using code metrics. Information Security Journal: A Global Perspective, 23(3), 57-67,2014.

[5] IT threat evolution Q3 2018. Statistics: https://securelist.com/it-threat-evolution-q3-2018-statistics/88689/

[6] https://www.mcafee.com/enterprise/enus/assets/reports/rp-quarterly-threats-dec-2018.pdf

[7] Aycock, J. Computer Viruses and Malware (Advances in Information Security). Sec-aucus,2006.

[8] Canfora, G., Iannaccone, A. N., &Visaggio, C. A. Static analysis for the detection of metamorphic computer viruses using repeated-instructions counting heuristics. Journal of Computer Virology and Hacking Techniques,10(1), 11-27,2014.

[9] Runwal, N., Low, R. M., & Stamp, M. Opcode graph similarity and metamorphic detection. Journal in Computer Virology, 8(1-2), 37-52,2012.

[10] Toderici, A. H., & Stamp, M. Chi-squared distance and metamorphic virus detection. Journal of Computer Virology and Hacking Techniques, 9(1), 1-14,2013.

[11] Al Daoud, E., Jebril, I. H., &Zaqaibeh, B. Computer virus strategies and detection methods. Int. J. Open Problems Compt. Math, 1(2), 12-20,2008.

[12] B. Anderson, D. Quist, J. Neil, C. Storlie, T. Lane. Graph-based malware detection using dynamic-analysis, J. Comput. Virol. Vol.7, No. 4, pp. 247-258,2011.

[13] Deshpande, S., Park, Y., & Stamp, M. Eigenvalue analysis for metamorphic detection. Journal of computer virology and hacking techniques, 10(1), 53-65,2014.

[14] M. E. Saleh, A. B. Mohamed, A. A. Nabi. Eigenviruses for metamorphic virus recognition, IET information security Vol. 5, No. 4, pp. 191-198, 2011.

[15] Mehra, V., Jain, V., &Uppal, D. DaCoMM: Detection and Classification of Metamorphic Malware. Fifth International Conference on (pp. 668-673). IEEE,2015.

[16] Mark Stamp Website in San Jose State University, [Online], http://cs.sjsu.edu/~stamp/viruses/

[17] Khammas, B. M., Monemi, A., Ismail, I., Nor, S. M., &Marsono, M. N. Metamorphic Malware Detection Based on Support Vector Machine Classification of Malware SubSignatures. TELKOMNIKA(Telecommunication Computing Electronics and Control), 14(3), 2016.

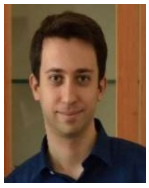
[18] Golbaghi, H., Vahidi-Asl, M., Khalilian, A., "A New Approach for Metamorphic Malware Detection by Static Analysis of Registers and Opcodes", Computing Science Journal, Vol. 4, pp. 3-15, (in Persian), 2017

[19] Mohamed, G. A., &Ithnin, N. BSBRT: API Signature Behaviour Based Representation Technique for Improving Metamorphic Malware Detection. In International Conference of Reliable Information and Communication Technology (pp. 767-777). Springer, Cham. . 2017, April.

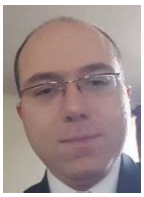
[20] Irshad, M., Al-Khateeb, H. M., Mansour, A.,Ashawa, M., &Hamisu, M. Effective methods to detect metamorphic malware: a systematic review. International Journal of Electronic Security and Digital Forensics, 10(2), 138-154, 2018.

[21] https://github.com/opsxcq/mirror-vxheaven.org

[22] Nar, M., Kakisim, A. G., Çarkaci, N., Yavuz, M. N., &Sogukpinar, I. Analysis and Comparison of Opcode-based Malware Detection Approaches. In 2018 3rd International Conference on Computer Science and Engineering (UBMK) (pp. 498-503). IEEE,2018, September.



مجتبی وحیدی اصل استادیار گروه مهندسی نرم‌افزار در دانشکده مهندسی کامپیوتر دانشگاه شهید بهشتی است. زمینه‌های علاقه‌مندی او آزمون و اشکال‌زدایی نرم‌افزار و امنیت نرم‌افزار است.



علیرضا خلیلیان دانشجوی دکترای مهندسی کامپیوتر گرایش نرم‌افزار در دانشگاه اصفهان است. وی مدارک کارشناسی ارشد و کارشناسی خود را نیز در گرایش نرم‌افزار دریافت کرده است. زمینه‌های پژوهشی وی آزمون و اشکال‌زدایی نرم‌افزار، امنیت نرم‌افزار، زبان‌های برنامه‌سازی و تحلیل کد است. از ایشان تاکنون بیش از چهل مقاله در نشریه‌ها و کنفرانس‌های داخلی و خارجی به‌چاپ رسیده و همچنین مؤلف چهار کتاب تخصصی در رشته مهندسی کامپیوتر است. وب‌گاه شخصی او در نشانی www.khalilian.net در دسترس است.

هادی گلباغی دارای کارشناسی ارشد مهندسی کامپیوتر گرایش نرم‌افزار از دانشگاه شهید بهشتی است و مدرک کارشناسی خود را نیز در گرایش نرم‌افزار دریافت کرده است. زمینه‌های پژوهشی او امنیت نرم‌افزار، شناسایی بدافزارها، شناسایی اپلیکیشن‌های مخرب اندرویدی و شناسایی نرم‌افزارهای جعلی است. از وی تاکنون مقالات متعددی در نشریه‌ها و کنفرانس‌های داخلی و خارجی به‌چاپ رسیده و هم‌اکنون در مرکز آپای دانشگاه کردستان مشغول پژوهش است.