هشتمین کنفرانس ملی مهندسی برق و الکترونیک ایران
دانشگاه آزاد اسلامی کتاباد
۲۷ و ۲۸ مردادماه ۹۵

مروری برتکنیکهای شناسایی بدافزارها

الهام نیکبخت
دانشگاه آزاد اسلامی واحد مشهد
مشهد،خراسان رضوی
el.nikbakht@gmail.com

سید محمدحسین معطر
دانشگاه آزاد اسلامی واحد مشهد
مشهد،خراسان رضوی
Moattar@mshdiau.ac.ir

مجید وفایی جهان
دانشگاه آزاد اسلامی واحد مشهد
مشهد،خراسان رضوی
Vafaeijahan@mshdiau.ac.ir

چکیده —بدافزار کد مخربی است که برای صدمه زدن به کامپیوتر و یا شبکه کامپیوتری ارتقا یافته است. تعداد بدافزارها بسرعت در حال افزایش است و این مقدار رشد باعث شده که محققین امنیت کامپیوتر روشهای جدیدی برای محافظت کامپیوترها و شبکه ها اختراع کنند.روشهای عمده ای که برای شناسایی malware وجود دارند عبارتند از:روش مبتنی بر امضاء ، مبتنی بررفتارومبتنی بر تکامل.روش برمبنا امضا متداولترین روشی است که بوسیله آنتی ویروسهای تجاری استفاده شده است اما فقط در حالتهایی که اسناد کاملا شناخته شده هستند میتواند استفاده بشوند.روش مبتنی بر رفتار برای پوشش معایب روش مبتنی برامضاء معرفی شد وباز بعلت بعضی نقص‌ها روش تکاملی معرفی گردید.این مقاله نگاه کلی روی ویروس های کامپیوتر و تکنیک های دفاعی را ارائه می دهد. نویسندگان ویروس های کامپیوتری از تکنیک های چند ریختی استفاده می کنند تا ویروس هایی که ساختار درونیشان را برای سرایت تغییر می دهند تولید کنند. از طرف دیگر تکنولوژی آنتی ویروس دائما نیرنگ های ویروس ها را تعقیب می کند تا بر تهدیدهای آنان غلبه کند.امروزه کارشناسان آنتی ویروس متدهایشان را طراحی و توسعه می دهند تا هر روز بیشتر و بیشتر قدرتمند شوند هدف این مقاله نگاه کلی به این روش ها و بیان ضعف هاست تا کسانی را که علاقه مند به پژوهش های بیشتر روی این بحث هستند ترغیب کند.

واژه‌های کلیدی —ویروس چند شکلی، رمزنگاری،تشخیص براساس ناهنجاری،شبیه سازی،

مقدمه

از زمان پیدایش اولین کد مخرب کامپیوتری در سال ۱۹۸٦، هر ساله تعداد زیاد و قابل توجه‌ای از کدهای مخرب جدید، ظهور پیدا می‌کنند. این رشد سریع کدهای مخرب، همیشه یک گام جلوتر از تلاش‌های متخصصین امنیتی برای ارائه راه حل‌هایی به منظور تشخیص و حذف آن‌ها از سیستم کاربران بوده است. اگرچه بدون در نظر گرفتن این رشد سریع نیز، روش سنتی مقابله با کدهای مخرب، شامل انتظار برای آلوده شدن تعدادی

کامپیوتر، تشخیص کد مخرب، طراحی راه حل مقابله و سپس ارائه آن به کاربران و مشتریان، فرآیندی است طولانی و ناکارآمد؛ چرا که در طول این فرآیند امکان وارد آمدن خسارت توسط کدهای مخرب وجود دارد.

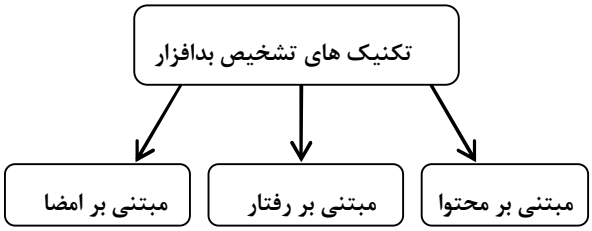
با توسعه اینترنت، سرعت تولید و گسترش کدهای مخرب نسبت به قبل، بسیار بیشتر شده است. طبق یک بررسی انجام شده توسط شرکت اف-سکیور تعداد کدهای مخرب کشف شده، فقط در سال ۲۰۰۷ برابر با مجموع تعداد کدهای مخرب کشف شده در ۲۰ سال قبل از آن بوده است و نسبت به سال ۲۰۰٦ رشد صد درصدی را تجربه کرده است[۱]. در بررسی دیگری که توسط شرکت امنیتی پاندا صورت پذیرفته، نشان داده است که روزانه در حدود ٦٥ هزار کد مخرب توسط موتور امنیتی این شرکت در اینترنت رصد می‌شود.

شرکت معتبر سمانتک معتقد است، مسأله تشخیص مخرب یا سالم بودن یک فایل یک مسئله حل نشدنی است[۲]. این شرکت معتقد است، تولید نرم افزاری که به صورت قطعی فایل‌های مخرب را شناسایی کند و خطای مثبت کاذب صفر داشته باشد و همچنین نیاز به بروز رسانی منظم نیز نداشته باشد عملی غیر ممکن است[۲]. از طرفی سیستم‌های کامپیوتری و ارتباطات زیربنایی دیجیتال امروزی به طور شدیدی در معرض انواع مختلف حملات مخرب قرار دارند. حملاتی که جنبه‌های مختلفی از جمله باج خواهی، انتقام، تروریستی، دزدی اطلاعات، سوء استفاده‌های مختلف و ... را شامل می‌شود. یکی از شایع‌ترین روش‌های این کار استفاده از کد مخرب است؛ که انواع و کاربردهای مختلفی دارد. از جمله می‌توان به ویروس، کرم، تروجان، جاسوس ابزار و ... اشاره کرد؛ که در این پایان نامه به اختصار کد مخرب نامیده می‌شوند.

تکثیر و انتشار کد مخرب ممکن است اثرات ناخوشایندی برای انواع مختلف کاربران عادی، شرکت‌های تجاری و دولت‌ها که از سیستم‌های کامپیوتری استفاده می‌کنند داشته باشد. به عنوان مثال، اگر یک کپی از یک کد مخرب به کامپیوتری که به شبکه‌ای متصل باشد نفوذ کند، می‌تواند منجر به از دست رفتن، استفاده غیر مجاز و یا تغییر مقدار زیادی از داده‌ها شود. در نتیجه این امر سبب بروز عدم اطمینان کاربران، نسبت به صحت اطلاعات بر روی شبکه خواهد شد.

روشهای شناسایی بدافزار

روشهای شناسایی بدافزار از دیدگاههای مختلفی طبقه بندی شده‌اند. همانطور که در شکل (۱) نشان داده شده دراین تحقیق روشهای شناسایی بدافزار را به سه گروه تقسیم می‌شود. [۳]



شکل(۱) روشهای شناسایی بدافزار[۳]

روشهایی برپایه امضاء:

امروزه انطباق الگو متداول ترین روش تشخیص بدافزار و شناسایی برمبنای امضاء عمومی ترین روش در این حوزه است.امضاء یک خصوصیت منحصربفرد برای هرفایل است.روشهایی مبتنی بر امضاء از الگوهای استخراج شده از بدافزارهای مختلف استفاده می‌کنند و این باعث می‌شود که آنها نسبت به دیگر روش‌های تشخیص ویروس بسیار سریع‌تر، موثر و کارا عمل کنند.این امضاها را اغلب با حساسیت خاصی جمع‌آوری می‌کنند تا منحصربفرد باشند و بنابراین روشهای شناسایی که از این روش(امضاء) استفاده می کنند نرخ خطای کمی دارند. و به دلیل همین میزان خطای کم متداولترین آنتی ویروسهای تجاری ازاین روش استفاده کنند. از سویی دیگر این روشها نمی‌توانند انواع بدافزار ناشناخته را شناسایی کنند و نیاز به مقدار زیادی نیروی انسانی،زمان و پول دارند تا امضاءهای منحصربفردی را بدست آورند که این از معایب این روشها بحساب می آید. از دیگر نقص‌های این روش ناتوانی در مقابله با بدافزارهایی است که کدهایشان در حملات مختلف تغییر میکنند نظیر پلی مورفیک و متامورفیک [٤].

روش مبتنی بر محتوا

طبقه بندی نرم افزارهای مخرب بر اساس آثار اولیه محتوای نرم افزارهای مخرب در این روش کد باینری مورد نیاز مخرب برای اولین بار جدا شده است. سپس محتوای مبتنی بر پروفایل یا ویژگی ها از داده ها جدا شده استخراج و به عنوان ورودی الگوریتم های طبقه بندی استفاده می شود. به عنوان مثال، Kolter و Maloof با استفاده از n–گرم از کدهای بایتی به

عنوان ویژگی های دستگاه، اعمال یادگیری طبقه بندی بدافزار را اجرا کردند [۵]

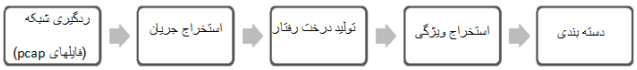
اخیراً تیان و همکاران، با استفاده از ویژگی های رشته های چاپ موجود در نمونه های بدافزار به تمایز بین بد افزار اجرایی و خوش خیم رسیدند[٦]. مانند بسیاری از سیستم های طبقه بندی موجود، فرض می کنیم که نمونه‌های نرم افزارهای مخرب های غیر بسته ای همه ی رویکردهای مبتنی بر محتوا نیاز به پیاده کردن دودویی دارند که این به علت وجود مخرب ها بسته بندی اغلب دشوار، آهسته و مبهم است.

علاوه بر این یک نوع جدید تا حدودی گیج کننده نرم افزارهای مخرب موجود ممکن است مطالب مختلف یا امضاء متفاوت داشته باشد اما همان رفتار گروه خود را نشان دهد. دقت در طبقه بندی مبتنی بر محتوا مبهم و تاریک بودن را کاهش می‌دهد. بنابراین رویکردهای مبتنی بر محتوا در شناسایی بدافزارهای جدید که در خانواده های موجود وجود دارد یک نقطه ضعف دیگر است.

روشهایی بر پایه رفتار

به منظور مقابله با محدودیت های روش مبتنی بر محتوا، روش ها، تجزیه، تحلیل و طبقه بندی مبتنی بر رفتار مخرب‌ها مطرح شده است. در این روش نمونه رفتار بدافزار در زمان اجرا توسط یک مشخصه رفتاری ارائه شده است. پس از آن طبقه بندی رفتاری پروفایل ها برای تجزیه و تحلیل بیشتر استفاده می شود. به عنوان مثال، لی و مُدی [۷] نمونه بدافزارها را با استفاده از توالی فراخوانی های سیستمی و ویرایش رشته از راه دور رده بندی کردند. بیلی و همکاران [۸]. رفتار بدافزارها در شرایط پایدار که باعث تغییرات روی سیستم می شود و تراکم فاصله نرمال (NCD) را به عنوان یک واحد برای طبقه بندی نمونه های مشابه نرم‌افزارهای مخرب تعریف کردند. Rieck و همکاران[۹و۱۰] از اطلاعات موجود در گزارش تجزیه و تحلیل ایجاد شده توسط CWS and Box برای تولید پروفایل های رفتاری و قطار SVM برای ساخت طبقه بندی خانواده بدافزارها استفاده گردید. بیشتر علائم تبعیض آمیز (مشخصه ممتاز) براساس طبقه بندی بدافزارها بین فایل‌های اجرایی مخرب و خوش خیم است.

یک مسیر دیگر در تحقیقات تمرکز بر خوشه ی نمونه ویروس نسبت به گروه با رفتارهای مشابه است[۸و۱۰و۱۱]. تجزیه و تحلیل خوشه به منظور شناسایی خانواده های جدید نرم افزارهای مخرب، زمان تجزیه و تحلیل و تلاش مقررات آنالیز را کاهش می دهد. یک گام مهم در تجزیه و تحلیل خوشه بدافزار پروفایل رفتاری و ارائه رفتار بدافزار است. به عنوان مثال، بایر و همکاران[۱۱] پروفایل رفتاری ایجاد کردند که از دنبال کردن گزیده علائم فراخوانی اشیا سیستم عامل و اینکه چه عملکردی روی آن اشیا انجام شد.



شکل(۲) مرور سیستم[۱۲]

روشهای شناسایی براساس رفتاربرنامه ای را جهت تشخیص اینکه یک نرم افزار بدخواه و مخرب است یا نه در نظر می گیرد[۱۲]. با توجه به اینکه روشهای مبتنی بر رفتار؛ عملکرد فایلهای اجرایی را در نظرمیگیرد در نتیجه روی نقاط ضعف روش‌های مبتنی بر امضاء حساس نشدند. به عبارتی دیگر در روش شناسایی بر اساس رفتار الگوریتم تشخیص روی عملکرد حساس می شود نه گفته ها.

در این روشها برنامه های با رفتار یکسان جمع آوری میشوند.بنابراین می‌توان فقط با یک امضاء رفتاری نمونه های مختلفی از بدافزار را شناسایی کرد.این نوع از مکانیسم شناسایی؛ به شناسایی بدافزارهایی کمک می‌کند که معمولا از منابع و خدمات سیستم به روش مشابه استفاده میکنند.

یک تشخیص دهنده بدافزار مبتنی بر رفتار به طور اساسی از اجزاء زیر تشکیل شده است: [۱۳]

۱–جمع آوری کنندهداده: این قسمت اطلاعات استاتیکی و دینامیکی در مورد برنامه اجرایی را جمع آوری می کند.

۲–مفسر: این قسمت اطلاعات خام جمع آوری شده توسط ماژول جمع آوری کننده داده را به فرم نمایشی واسط تبدیل می‌کند.

۳–تطبیق دهنده: این ماژول جهت مقایسه نتیجه مفسربا امضاهای رفتاری استفاده می‌شود.

مزیت عمده روشهای شناسایی بدافزار بر اساس رفتارآن است که توانایی شناسایی بدافزارهای چندریخت که تکنیک‌های مبتنی بر امضاء نمی-توانند آنها را تشخیص دهند را دارد و از طرفی دیگر طولانی بودن زمان اسکن از معایب اصلی روشهای شناسایی بدافزار مبتنی بر رفتار می-باشد.[۱٤]

در برخی مقالات پیچیدگی برخی آنالیزها با در نظر گرفتن سادگی محاسبات و نرخ صحت تشخیص بالا گزارش شده که آنان نمی توانند بدافزار ناشناخته که هنوز امضاء آن تولید نشده راکشف کنند.

در مواجه با بدافزارهای ناشناس، روش کشف رفتاری توسعه یافته است، که از دانش تشکیل رفتار طبیعی جهت تصمیم تخریب یک برنامه استفاده می کند.

اکثرا این روش ها از تکنیک های داده کاوی استفاده می کنند. داده کاوی برای تشخیص بدافزار از طریق امضاءها به صورت اتومات می باشد.

در[۱۵] یک چارچوب طبقه بندی خودکار برای نمونه های نرم افزارهای مخرب در شبکه خود رفتار ارائه شده است. این مقاله، روش معنایی قدرتمندی، جهت کشف بد افزارهای ناشناس بر اساس ترکیب مدل گرافی(CFG) و API فرا خوانده شده را نشان می دهد. هدف اصلی این مقاله، استخراج CFG از برنامه ها و ترکیب آن با فراخوان های APIاستخراج شده جهت اطلاعات بیشتر در مورد فایل های قابل اجرا می باشد. مدل نمایشی جدید API-CFG نامیده می شود. علاوه بر این جهت داشتن یادگیری و فرایند دسته بندی سریع گرافهای کنترل جریان به مجموعه بردارهای تابع با ترفندی مناسب، تبدیل می شوند روش فوق، قادر به دسته بندی کردن کد مخرب و غیرمخرب غیرقابل رویت با درجه ی دقت و صحت بالایی می باشد. نتایج، پیشرفت آماری قابل توجهی را در روش کشف n–گرمی نشان می دهد.

در ابتدا، بسیاری از کارها برروی کشف بدافزار با استفاده از توالی ساختار و تجزیه و تحلیل کنترل جریان، انجام می شد[۱٦و۱۷]. علاوه بر این فراخوانی های API بدلیل اطلاعات مفیدشدرموردفعالیت بدافزار،بررسیمی شدند[۱۸و۱۹] .همچنین تکنیک یادگیری ماشین در فازهای های مختلف تشخیص، استفاده می شده است[۲۰].

دراوایل پیدایش ویروس، فقط ویروس های ساده و استاتیک به جهان معرفی شدند. بنابراین، روش های ساده ی مبتنی بر امضاء قادر بودند بر آنان غلبه کنند[۲۱و۲۲]. این روش های ساده، در ابتدا مناسب بودند، اما تحول سریع در فعالیت های مخرب بد افزار، محققان را به سوی روشهای جدید سوق داد. یکی از جذاب ترین روشهایی که در روزهای اولیه بوجود آمد، داده کاوی را بر روی n–گرم ها به کار می گیرد.

جرالد و دستیارانش، روشی را پیشنهاد دادند که تجریه و تحلیل های n–گرم ها را جهت شناسایی ویروس های bootsector با استفاده از شبکه های عصبی، توسعه می داد. n–گرم ها بر اساس فراوانی های وقوع در برنامه های ویروسی غیرمخرب، انتخاب می شدند. کاهش ویژگی جهت ٤-cover به این صورت انجام می شود که هر ویروس برای اینکه در مجموعه داده ها در نظر گرفته شود باید حداقل دارای چهار تکرار n–گرم باشد.

با توجه به اینکه n–گرم ها جهت بررسی معنایی برنامه، ضعف دارند، ویژگی دیگری باید به جای آن استفاده شود. هافمیر و همکارانش از توالی فراخوانی های سیستمی برای تشخیص کدهای مخرب، استفاده کردند[۱۷].

توالی فراخوانی API، وابستگی های نهان بین توالی کد را نشان می داد. برگرون و دستیارانش، ویژگی های پویا و رفتاری را با توجه به مبارزه با تغییر شکل استفاده کردند[۲۳و۲٤]. جائو و دستیارانش روش جدیدی را جهت یافتن تفاوت ها در برنامه دودوئی ارائه دادند. در واقع تفاوت های ترکیبی دارای پتانسیل کافی جهت ایجاد اختلال (صدا) می باشند، بنابراین یافتن تفاوت های معنایی یک چالش است. آنها روش جدید هم ریختی گراف و اجرای نمادین را جهت تحلیل گراف کنترل جریان فایل های PE استفاده کردند، که این کار با تشخیص زیرگراف مشترک ماکسیمم، انجام دادند. روش آنان، تفاوت معنایی بین یک فایل PE و نسخه پچ شده اش را کشف کرد، اگرچه ، نرخ مثبت کاذبش ، نیازمندی های zero-day را برطرف نمی کند[۲۵].

سزار و زیانگ روش طبقه بندی جدیدی را ارائه دادند که گراف های جریان، جهت شناسایی بدافزارهای چند شکلی استفاده می شدند. آنان از الگوریتم اکتشافی یا هیوریستیک جهت انطباق گراف جریان هنگام یافتن هم ریختیهای گراف، استفاده کردند. بنابراین، آنان توانستند شباهت بین فایل های PE را تخمین بزنند و در نهایت آنان الگوریتم دسته بندی بر اساس روش خود ارائه دادند[۲٦].

جئونگ و لی گرافی را توسط رشته ی فراخوانی هایAPI استخراج کردند و آن را کد گراف نامیدند، کار آنان مستقیما روی فایل های دودوئی و واکشی پرش ها و دستورالعمل های فراخوانی از فایل های PE ایجاد کد گراف بود. روش آنان می تواند ٦۷٪ بدافزارهای ناشناس را کشف کنند[۲۷]. بنظر می رسد که دستیابی آنان تحقق گرا و دارای کاربرد عملی می باشد، اما هنوز سرعت اکتشاف آن پایین می باشد.

دولین و رولس هر تابع را بعنوان یک گراف جریان در نظر گرفتند، و سپس رابطه ی فراخوان بین آنان را ترسیم نمودند. (یعنی از گره ی که فراخوانی را انجام می دهد به گره ی فراخوانده شده)؛ بنابراین هر فایل PE بعنوان گرافی از گرافها نشان داده می شود. تحت این گرایش، آنان توانستند به یک هم ریختی اصلاح شده بین مجموعه ای از بلاک های اصلی و مجموعه ای از توابع، در دو فایل PE مجزا برسند. بنابراین آنان توانستند سرقت کد را کشف کنند[۲۸]. آبادی و دستیارانش معنای یکپارچگی کنترل جریان (CFI) را با بازنویسی کد ماشین، فرموله کردند. CFI روشی را جهت بررسی فرآیندهای اجرایی درون یک گراف کنترل جریان معین را ارائه می دهد که از تجزیه و تحلیل های برنامه ایستا گرفته شده است. به عبارت دیگر، CFI بررسی های پویا برای درستی کنترل جریان اجرائی تاکید دارد، و دارای اجرای موثر و کارآمد در سایه پنهان برنامه با محافظت بالایی باشد[۲۹] .

مقاله [۳۰] یک روش جهت مدل کردن و تحلیل سیستم‌های بدافزار فراهم می‌کند. که یک مدل هستی شناسی فازی نوع دو زمانبند را طراحی می‌کند و یک مدل توزیع شده بدافزار را در محیط واقعی پیاده سازی می‌کند. این سیستم می‌تواند انواع بدافزار را که شامل گونه های مختلف ماشین مجازی می‌باشد تحلیل کند بنابراین خیلی از چالش‌های امنیتی با استفاده از تحلیل رفتار می‌تواند بهتر درک شود.

استراتژی های اختفاء[۱]

وقتی توسعه دهندگان بدافزار ظاهر می شوند که بدافزارشان قرار هست شناخته شود وسعی میکنند با بکار بردن انواع روشهای اختفا از استراتژیهای شناسایی آنتی ویروس فرار کنند.در این بخش بعضی از شناخته شده ترین استراتژی های اختفا معرفی می‌شود.

مبهم کردن کد[۲]

در این روش توسعه دهندگان بدافزار فعالیتهایی را براساس روشهای مبتنی بر امضا جهت جلوگیری از شناسایی بدافزار بیان میکنند.این فعالیت‌ها شامل اضافه کردن فرامین اشتباه وپرشهای غیرلازم و غیره میباشند.

رمزنگاری کد[۳]

اینگونه برنامه های مخرب از مکانیسم دفاعی رمز نگاری برای خودشان یا فعالیتهای مخرب‌شان استفاده می‌کنند. برنامه های مخرب رمز نگاری شده مجموعه ای پیچیده از الگوریتمهای رمزگشا و رمزنگاری و کلید های رمزنگاری و کد مخرب رمزنگاری شده است.وقتی که برنامه مخربی اجرا میشود الگوریتم راهنما و رمزگشا برای رمز گشایی از بخش مخربش استفاده میکند.برنامه های مخرب تکثیرمیشوند و با راهنمایی های جدید تولید شده و الگوریتم رمزنگاری راهنمای جدیدی تولید می‌کند.نسخه جدید رمز نگاری شده تولید می شود.این نسخه شامل الگوریتمهای رمز نگاری و راهنمای جدید است. بنابراین حتی راهنمای رمزنگاری وکد رمزنگاری شده پیوسته تغییر میکند اما بخاطرالگوریتم‌های رمزگشا فیکس شده میتوانند شناسایی شوند.

استراتژیالیگومورفی[٤]

برنامه های مخربی که از این استراتژی برای رمزنگاری بعنوان مکانیسم دفاعی برای محافظت از خودشان استفاده میکنند قادر هستند الگوریتم رمزنگاریشان را در مدت محدودی تغییر دهند.بعنوان مثال ویروسی که حلقه هایی به تعداد محدود و کوچک و رمز گشاینده مختلفی دارد.

استراتژی پلی مورفی[٥]

برنامه های مخربی که از این استراتژی استفاده میکنند معمولا خودشان رابا یک الگوریتم رمزنگاری رمزنگاری میکنند.بنابراین با هرآلودگی کلیدی با رمز نگارنده های مختلف مورد استفاده قرار میگیرد.همچنین می توانند تعداد نامحدودی الگوریتمهای رمز نگاربرای جلوگیری از شناسایی شدن استفاده کنند.در هراجرا بخشی از کد رمزگشا تغییر میکند.با توجه به نوع برنامه مخرب،فعالیتهای مخرب یا سایر فعالیتهایی که بوسیله برنامه مخربی که جایگزین عملیات رمزنگاری میتوانند بشوند انجام میگیرند.معمولا موتور انتقال در برنامه های مخرب رمزنگاری شده؛ با هرتغییری الگوریتم رمزنگار میسازد.سپس این موتورو برنامه های مخرب به کمک الگوریتمهای تولید شده و راهنمای جدید رمزگشا کد به آنها متصل شده و رمز نگاری میشود.

استراتژی متامورفی[٦]

[۱] CONCEALMENT STRATEGIES
[۲] Obfuscation
[۳] Codeencryption
[٤] Oligomorphic strategy
[٥] Polymorphic strategy

برنامه های مخربی که از این نوع استراتژی استفاده میکنند جزء پیچیده ترین انواع برنامه ها هستند.این نوع برنامه ها خودشان را بصورتی تغییر میدهند که نمونه های جدید هیچ شباهتی به نمونه اصلی ندارد.این برنامه ها موتور کد گذاری ندارد و در هر انتقالی بطور خودکار کد سورس malware تغییر میکند.

روشهای خیلی زیادی هستند که میتوانند ظاهر کد سورس را تغییر دهند[۳۱]. بعنوان مثال:

فرمتهای جایگزین کامنتها و یا فضاهای خالی در کد سورس را پاک یا اضافه میکند.این روش آسانترین و دارای کمترین کارامدی درمیان روشها میباشد.گاهی الگوریتم شناسایی را میتواند گمراه کند.

با تغییر نام دادن مستمر نام شناسنده تغییر پیدا میکند بدون اینکه به درستی برنامه خدشه ای وارد شود.تغییر اسامی باعث گیج شدن انسان ممکن است بشود اما اکثرا هیچ اثری برروی روشها ندارد.

چیدمان دوباره میتواند نتیجه عباراتی را در برنامه تغییر دهد در صورتی که خطایی در برنامه ایجاد نکند.

جایگزین عبارتی درصورتی میتواند بعضی عبارات را با عبارات دیگر که عملکرد مشابه دارد جایگرین کند که اشتباه منطقی در برنامه تولید نکند.این روش از قبلی پیچیده تر است.

تغییرکنترل بعضی از عملکردهای ساختار کنترل با سایرعملکردهای کنترل که کارمشابهی انجام میدهد تغییر می کند. مثلا for در حلقه و while حلقه قابلیت تغییر دارند.

ورود جانک کددر کد بی اهمیت برنامه منجر به تشخیص اشتباه میشود درصورتی که منطق برنامه اصلی مختل نشود.بعبارت دیگر کد جانک اجرایی برروی منطق کد سورس اثر نمی گذارد[۳۲] .

سابروتین داخلی و خارجی روشهایی هستند که کد سابروتین را با تماس سابروتین تعویض میکنند و برعکس.کد خارجی روشی هست که کد سابروتین رابا فراخوانی سابروتین[۷] جایگزین میکند.کد داخلی گزینه متقابل هست و روشی هست که برای جلوگیری از آورهد(سربار) بودن سابروتین کال استفاده شده است.این گونه انتقالات از کد اصلی محافظت میکند اما به روشهای مختلف اینکار انجام میشود.

[۶] Metamorphic strategy

[۷] subroutine call

تحلیل بد افزارها، به روش های تحلیل ایستا و پویا

تحلیل بد افزارها، به روش های تحلیل ایستا و پویا صورت می گیرد. در تحلیل ایستا امضای نرم افزار با امضاء های موجوددر پایگاه داده ابزار ضد بدافزار مقایسه می گردد و در صورتی که شباهتی با یکی از امضاهای موجود در پایگاه داده وجود داشت برنامه بعنوان یک بدافزار شناسایی و معرفی می گردد و اگر مشابه نبود آن برنامه به عنوان برنامه ی بی خطر معرفی می‌گردد. [۳۳]

مزیت این روش آن است که اگر بدافزارها به صورت ایستا تحلیل شوند نیازی به اجرا ندارند. در نتیجه، آسیب احتمالی ناشی از اجرای بدافزار به سیستم وارد نمی شود. اما در این روش بدافزار با استفاده از تکنیک های رمز نگاری، مبهم سازی و چند ریختی، پیچیدگی خود را بالا برده، در نتیجه تحلیل آن به روش ایستا، سخت و در بسیاری از اوقات غیر ممکن می شود. روش ایستا کند بود. اگر تعداد بسیار زیادی بدافزار داشته باشیم بایستی امضای تمام این بدافزارها را در پایگاه داده نگهداری نماییم و هر برنامه ای (کدی) که بخواهیم کشف نماییم، امضایش با تمام رکوردهای موجود در پایگاه داده مقایسه می شود. که این کار بسیار زمان بر است. حجم پایگاه داده ای که امضای بدافزارهای شناسایی شده را نگهداری می کنند روز به روز در حال افزایش بود و نگهداری و جابه جایی این پایگاه داده بین کلاینت و سرور مشکل می بود. روش ایستا ناامن بود. نویسنده های بدافزار به راحتی در کدهای بدافزار تغییر جزیی ایجاد کرده و در نتیجه امضای بدافزار تغییر پیدا می کرد. لذا این امضاء در پایگاه داده موجود نبوده و ابزار ضد بدافزار نمی توانست بدافزار را شناسایی نماید، بنابراین، بدافزار به عنوان نرم افزار امن، معرفی می شد.

اساس این روش بر این نظریه استوار بود که متخصصان اعتقاد داشتند کلیه بدافزارها، در رفتار، شباهت هایی با یکدیگر دارند و همگی یک سری رفتارها را به صورت مشابه با یکدیگر انجام می دهند، لذا با توجه به این رفتارهای مشابه و بدون توجه به امضاهای ذخیره شده در پایگاه داده (و مقایسه بین امضای برنامه و امضاهای ذخیره شده)، رفتار برنامه را مشاهده و تحلیل می نمایند. اگر رفتار برنامه در حال اجرا، مشابه رفتار بدافزارهایی که قبلاً شناسایی شده بودند، باشد، برنامه در حال اجرا، به عنوان بدافزار معرفی می شد در غیر اینصورت به عنوان یک برنامه امن معرفی می گردد[۱۰]. (بدافزارها حدود ۱۰۰۰ تا ۲۰۰۰ رفتار مشابه با یکدیگر دارند لذا مقایسه یک

رفتار با ۲۰۰۰ رفتار، ساده تر از مقایسه یک امضاء با میلیون ها امضاء می باشد).

در گذشته در زمینه کشف رفتاری بدافزارها، روش های مختلفی مطرح گردیده است.در[۳۳] روشی ارائه گردید که براساس دسترسی بدافزار به کتابخانه های پیوند پویا، ماهیت برنامه را تشخیص می داد. این روش بدین صورت بود که ابتدا قوانینی از یک مجموعه بدافزار استخراج گردید تا بر اساس آن صفات مخرب تشخیص داده شود اما در این روش از نرم افزارهای سالم جهت حذف رفتارهای مشترک نرم افزارهای سالم با بدافزارها استفاده نشده بود. همچنین این روش دارای معایبی همچون الف) نرخ مثبت کاذب بسیار بالا بدلیل عدم استفاده از برنامه های بی خطر جهت مدلسازی ب)عدم استفاده از محیط امن جهت ترمیم خسارتهای وارده ناشی از اجرای بدافزار ج)استفاده از مجموعه داده آزمون قدیمی و نادرست (بصورتی که داده های آزمون استفاده شده اغلب دارای رفتار یکسان بوده اند) می باشد[۳٤].

روش دیگر ارائه شده در[۳۵و۳٦] از فراخوانی های سیستمی جهت ساخت مدل استفاده کرده است.

روش دیگر ارائه شده در[۳۷]از مقادیر درون ثبات های سیستم عامل، جهت ثبت رفتار نرم افزار استفاده شده است. بطوریکه بوسیله تکنیک های داده کاوی، رفتارهای مخرب را بر اساس مقادیری که در حین اجرای بدافزار در ثبات های سیستم ذخیره می گردد، شناسایی می نماید. این روش دارای نرخ مثبت نادرست بسیار بالایی بوده است، بدلیل آنکه ممکن است برای دو عمل متفاوت، مقادیر ثبات های سیستم، یکسان باشند و دقت روش بشدت کاهش پیدا نماید.

تکنیک‌های هوشمند در استخراج الگو و مدل رفتاری بدافزار

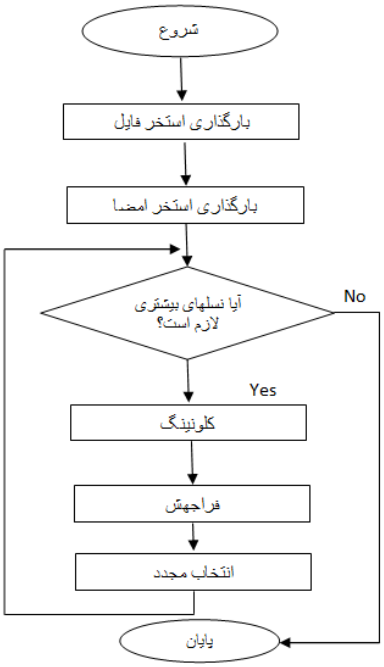
امروزه در همه روش های امنیت کامپیوتر تکنیک های پایه ای هوش مصنوعی متفاوتی استفاده می شود[۳۸]. همچنین روشهای هوش ازدحامی، الگوریتم ژنتیک، بهینه سازی کلونی مورچه ها کاربردهای متفاوتی در دسته بندی الگو و تصویر و پردازش سیگنال دارد[۳۹و٤۰و٤۱]. سیستم ایمنی مصنوعی (AIS) بسیار شبیه به نمونه ها در ساختار و مکانیسم است، با این حال روشی تازه است. AIS در زمینه های مختلف در کامپیوتر اعمال می شود، که مهمترین آنها تشخیص درست ویروس هاست. محافظت در برابر ویروس ها هر روز سخت تر و سخت تر می شود و آنها هرکسی را که از رایانه استفاده میکند را تهدید می کنند. هوش ویروس ها در طول زمان افزایش می یابد، و امضای آنها را به طور مداوم در حال تغییر است[٤۲و٤۳]. که این ماموریت ضد ویروس را سخت تر میکند[٤٤]. (AIS) دارای چندین مفهوم انتخاب کلونال، انتخاب منفی و تئوری ایمنی شبکه است. پیشنهاد این مقاله الگوریتم (VDC) که از الگوریتم انتخاب کلونال الهام شده و CLONALG به طور دقیق بررسی شده را جهت کشف ویروس ها پیشنهاد میکند[٤۵].

مطالعات نشان داده است که ۲۵ درصد از افرادی که از کامپیوتر استفاده می کنند به گونه ای به بدافزار آلوده شده اند که حدود نیمی از این کامپیوترها, کامپیوتر های تجاری هستند که از این آلودگی رنج میبرند[٤٦]. ساده ترین و رایج ترین روش برای حفاظت شبکه از حملات ویروسی استفاده از فن آوری امضا است.

این مقاله باید یک شیوه کمکی با ارائه پیشنهاد الگوریتم کلونال کشف ویروس و بهبود پارامترهای استفاده از GA پیشنهاد می کند، الگوریتم VDC یک موضوع جدید است ولی ویروس یک موضوع قدیمی است به هر حال مسئله کشف ویروس یک مسئله رو به رشد است چون تمام افرادی که از کامپیوتر استفاده می کنند تحت تاثیر آن قرار می گیرند.

الگوریتم های انتخاب منفی برای کشف ویروس استفاده شده است[۳۸و٤۷و٤۸و٤۹]. اما الگوریتم انتخاب کلونال با انجام یک جستجوی گسترده وب و یک تحقیق در گستره وسیع از ژورنال های خاص هنوز الگوریتم انتخاب تولیدی کلونال برای برنامه های کاربردی کامپیوتر استفاده نشده است این طور فهمیده میشود که استفاده از الگوریتم انتخاب کلونال یک همکاری جدید است. اصول انتخاب کلونال رویکرد پاسخ ایمنی به توصیف محرک های آنتی ژن را توضیح میدهد. چیزی که میتوان به صورت زیر بیان کرد: فقط سلول هایی که آنتی ژن را می شناسند تکثیر و انتخاب می شوند و سایر سلول ها تکثیر نمیشوند. این سلول های B تولید شده که کپی والدینشان هستند جهش پیدا کرده اند. وقتی آنتی بادی ها با قدرت با آنتی ژن ها جورمیشوند این سلول های B تحریک به ایجاد کلونال از خودشان می شوند[۵۰]. در [۵۱] آنتی ژن ها ویروس های کامپیوتری را در فایل های آلوده نشان می دهند و آنتی بادی ها امضاءها را نشان می دهند. امضاء ها دارای مقادیر تطبیق شده بالایی هستند (دارای تابع هدف بالاتر) جهت شبیه سازی (سوپر جهش انتخاب مجدد) انتخاب می شوند بنابراین

شبیه سازی کپی هایی از امضاء های دارای بهترین تناسب را تولید می کنند که جهت جهش و تولید و ارائه شناسایی ویروس هایی که دارای ویژگی های متفاوتی هستند (ژنهای متفاوت)، قبلاً حمله نکرده اند استفاده می کنند. در این تحقیق انتخاب مجدد به صورت تصادفی به الگوریتم انتخابی وابسته به کلونال اضافه می شوند تا انتخاب بهترین امضاء تغییر یافته را تضمین کنند. شکل زیر مراحل الگوریتم را نشان می دهد.



شکل (۱) فلوچارت الگوریتم VDC[۵۱]

از نکات مهم در چرخه تکاملی بدافزارها به وجود آمدن دسته های مختلف مانند ویروس، نرم افزارهای جاسوسی، تروجان و غیره است. امروزه شاهد توزیع این تهدیدات از طریق شبکه های اجتماعی و پتانسیل گسترش آن به ناحیه های دیگری در جهان هستیم. در دهه گذشته حملاتی مانند بدافزار استاکس نت با هدف گروه کاربری خاص مانند صنعت را شاهد بودیم[۵۲].

این بدافزار در ابتدا برای حمله به تجهیزات صنعتی به وجود آمد و امروز نمونه های مختلفی از آن را در حوزه های کاربری دیگر بوجود آمده است .

بنابراین از مسائل مهم در شناسایی بدافزارهای ناشناخته، وجود بانک داده ای جامع و بروز از نمونه های شناخته شده بر اساس حوزه های کاربری و منطقه جغرافیایی برای متخصصین است.

نکته مهم دیگر در شناسایی بدافزارها استخراج الگوهایی از نمونه های شناخته شده است. متخصصین این حوزه با استفاده از فن های داده کاوی[۸] و یادگیری ماشین[۹] تلاش در ارائه الگوهایی با قابلیت شناسایی نمونه های ناشناخته دارند. کارشناسان با بزرگ شدن بانک داده ای بدافزارها با مشکل محدودیت و وابستگی برخی از الگوریتم های کارآمد مانند « تحلیل مولفه های اصلی [۱۰]»، « ماشین بردار پشتیبانی[۱۱]»، رگرسیون[۱۲] و غیره به پردازنده و حافظه اصلی مواجه هستند[۵۳]. وجود این مشکل منجر به زمان بر شدن استخراج الگو مخرب ها، عدم شناسایی به موقع بدافزارهای جدید و تحمیل هزینه مالی بالا به محققین و شرکت های فعال در این حوزه می شود. بنابراین دو چالش اساسی وجود دارد:

بانک داده ای بروز بر اساس حوزه کاربری و نواحی جغرافیایی

ازدیاد نمونه ها برای تحلیل و استخراج الگو

به منظور استخراج الگو بدافزار توسط فن های داده کاوی و یادگیری ماشین، ابتدا می بایست آنالیزی در قالب یک سری خصوصیات از آن ارائه گردد. اولین راهکار آنالیز مخرب تکنیک های ایستا است [٥٤]. از نمونه تکنیک تحلیل ایستا، می توان به « بصری سازی بدافزار[۱۳]» اشاره نمود که تاثیر قابل توجهی در کاهش زمان آنالیز دارد [۵۵و۵۶]. پایین بودن میزان خطا[۱٤] فرمول زیر یکی از شاخص های یک الگوریتم شناسایی است. مخرب ها با رعایت تکنیک هایی از قبیل ساختار استاندارد باینری اجرایی، پنهان سازی کد مخرب، اجرای کد مخرب در زمان اجرا از طریق منبع دیگر و غیره منجر به افزایش معیارهای تشخیص نادرست۱۵ و موارد تشخیص داده نشده[۱٦] می شوند و در نتیجه دقت الگوریتم کاهش می یابد [۵۷].

$$\text{Error rate} = \frac{(FP + FN)}{ALL}$$

برای غلبه بر مشکل دقت پایین راهکارهای ایستا، از روش های پویا معروف به « آنالیز رفتاری[۱۷] » استفاده می گردد [٥٤و٥٨]. یکی از پرکاربردترین روش ها اجرای برنامه در یک محیط ایزوله به نام سندباکس[۱۸]

[۸] Data mining
[۹] Machine Learning
[۱۰] Principal Component Analysis (PCA)
[۱۱] Support vector machines (SVM)
[۱۲] Regression
[۱۳] Malware Images
[۱٤] Error rate
[۱۵] False Native (FN)
[۱٦] False Positive(FP)
[۱۷] Malware Behavior Analysis
[۱۸] Sandbox

است. در محیط سندباکس فعالیت های زمان اجرای برنامه از قبیل توالی توابع سیستمی فراخوانی شده، تغییرات فایل سیستم، تغییرات مقادیر رجیستری در سیستم عامل ویندوز و غیره مانیتور و ثبت می گردد [۵۹]. از مشکلات عمده این روش نیاز به زمان بالا برای فرایند آنالیز و تولید گزارش است. بنابراین توسط تیم های تحقیقاتی و شرکت های تجاری در دهه گذشته محصولاتی مانند CWSandbox & Norman-G۲ Anubisو غیره جهت هوشمند سازی فرآیند مانیتور و ثبت گزارش ها ارائه شده است[٦۰و٦۱و٦۲]. ابزارهایی مذکور این قابلیت را دارند تا از طریق وب سرویس و واسط کاربری وب بتوان فایل را به آن ها ارسال نمود و امکان جمع آوری آنلاین فایل های مشکوک را از هر نقطه جغرافیایی برای تیم های تحقیقاتی فراهم کنند. هیچ یک از ابزارهای یاد شده الگوریتم های داده کاوی برای استخراج مدل از خروجی آنالیزها را فراهم نکرده اند. البته ابزارهای جانبی توسط تیم های تحقیقاتی مانند Malheur برای داده کاوی بر روی خروجی ابزارهایی مانند CWSandbox ارائه شده است[٦۳و٦٤]. در این ابزار متأسفانه الگوریتم های داده کاوی به طور کامل پیاده سازی نشده است و همچنان مشکلات محدودیت پردازنده و حافظه اصلی در الگوریتم ها وجود دارد. در داده کاوی و یادگیری ماشین با توجه به افزایش اطلاعات موردنیاز برای تحلیل و استخراج مدل، چارچوب مختلفی از قبیل scikit-learn ارائه شده است[٦۵]. برخی از الگوریتم‌های محیط‌های ارائه شده همچنان دارای محدودیت‌ها و مشکلات حافظه هستند. در همین راستا مبحث پردازش موازی[۱۹] داده کاوی بر روی پردازنده کارت گرافیک[۲۰] مطرح شده است[٦٦] اما متأسفانه اکثر الگوریتم ها هنوز پیاده سازی نشده اند و بیشتر در سطح مقالات مطرح است .

در دهه گذشته با افزایش محبوبیت محاسبات ابری[۲۱] و با توجه به قابلیت های آن برای رفع محدودیت های پردازنده و حافظه اصلی، چارچوب های داده کاوی و یادگیری ماشین در محیط پردازش ابری پیاده سازی شده اند. بطور مثال می توان به پروژه کد باز Apache Mahout اشاره نمود[٦۷].

[۱۹]Parallel Computing

[۲۰]Graphics Processing Unit (GPU)

[۲۱]Cloud Computing

قابل ذکر است در سال های اخیر شرکت های تجاری حوزه امنیت سیستم های رایانه ای از قبیل کسپراسکی، بیت دفندر و غیره اقدام به اضافه نمودن پردازش ابری به محصولات خود کرده اند.

جمع بندی

بدون شک بدافزارها یکی از مهم‌ترین تهدیدهای امنیتی برای فن‌آوری اطلاعات بوده و هستند. در طی سالیان گذشته، از زمان بدافزارهای ساده تا تهدیدهای پیشرفته‌تری همچون ویروس‌های پیشرفته امروزی، همواره یکی از مهم‌ترین دلایل رخدادهای امنیتی این بدافزارها بوده‌اند. در این فصل مروری بر انواع روش‌های حمله و نحوه مقابله با آنها انجام شده است.

مراجع

[۱] F-Secure, Reports amount of malware grew by ۱۰۰% during ۲۰۰۷. Availablefrom:http://www.fsecure.com/en_IN/abouts/pressroom/news/۲۰۰۷/fs_news_۲۰۰۷۱۲۰٤_۱_eng.html [Accessed ۸ febraury ۲۰۱۱].۲۰۰۷

[۲] Symantec corporation. ‘Understanding heuristics’, volume XXXIV, ۲۰۰۳.

[۳] P. Szor.” The Art of Computer Virus Research and Defense”. Addison Wesley for Symantec Press, New Jersey, ۲۰۰۵.

[٤] P. Gutmann. “The Commercial Malware Industry.”in proceeding of the ۲۰۰۷ DEFCON conference, DEFCON' ۰۷ Las Vegas ۲۰۰۷

[۵] J. Z. Kolter and M. A. Maloof, “Learning to Detect and Classify Malicious Executables in the Wild,” J. Mach. Learn. Res., vol. ۷, pp. ۲۷۲۱–۲۷۴۴, Dec. ۲۰۰۶.

[٦] M. Gheorghescu, “An automated virus classification system,” in Virus Bulletin Conference, ۲۰۰۵, pp. ۲۹۴–۳۰۰.

[۷] T. Lee and J. J. Mody, “Behavioral classification,” in EICAR Conference, ۲۰۰۶.

[۸] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F.Jahanian, and J. Nazario, “Automated classification and analysis of internet malware,” in Proceedings of the۱۰th international conference on Recent advances in intrusion detection, Gold Goast, Australia, ۲۰۰۷, pp.۱۷۸-۱۹۷.

[۹] K. Rieck, T. Holz, C. Willems, P. Düssel, and P.Laskov, “Learning and Classification of Malware Behavior,” in Detection of Intrusions and Malware, and Vulnerability Assessment, vol. ۵۱۳۷, Springer Berlin /Heidelberg, ۲۰۰۸, pp. ۱۰۸-۱۲۵.

[۱۰] K. Rieck, P. Trinius, C. Willems, and T. Holz, “Automatic analysis of malware behavior using machine learning,” Journal of Computer Security, vol. ۱۹, no. ۴,pp. ۶۳۹-۶۶۸, Jan. ۲۰۱۱.

[۱۱] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, “Scalable, behavior-based malware clustering,” in Network and Distributed System Security Symposium (NDSS), ۲۰۰۹

[۱۲] KALPA, “Introduction to Malware”,“http://securityresearch.in/index.php/projects/malware_lab/introduction-to-malware/۸/”, ۲۰۱۱.

[۱۳] G. Jacob, H. Debar, and E. Filiol, “Behavioral detection of malware:

[۱٤] from a survey towards an established taxonomy,” Journal in Computer Virology, pp. ۲۵۱–۲۶۶, ۲۰۰۸

[۱۵] A. Ahmed, E . Elhadi, M. A. Maarof and A. H. Osman, “Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph Information Assurance and Security Research Group.” Journal, A., Sciences, A., & Publications, S., Faculty of Computer Science and Information Systems, ۹(۳), ۲۸۳–۲۸۸,۲۰۱۲

[۱٦] Eskandari, Mojtaba and Hashemi, Sattar, "A Graph Mining Approach for Detecting Unknown Malwares" Journal of Visual Languages and Computing ۲۳ (۲۰۱۲) ۱٥٤–۱٦۲.

[۱۷] V. Sathyanarayan, P. Kohli, B. Bruhadeshwar, Signature generation and detection of malware families, in: Information Security and Privacy, Springer, ۲۰۰۸, pp. ۳۳٦–۳٤۹.

[۱۸] P. Vinod, V. Laxmi, M. Gaur, G. Kumar, Y. Chundawat, Static cfg analyzer for metamorphic malware code, in: Proceedings of the Second International Conference on Security of Information and Networks, ACM, ۲۰۰۹, pp. ۲۲٥–۲۲۸.

[۱۹] F. Chen, Y. Fu, Dynamic detection of unknown malicious executables base on api interception, in: First International Workshop on Database Technology and Applications, IEEE, ۲۰۰۹, pp. ۳۲۹–۳۳۲.

[۲۰] J. Xu, A. Sung, P. Chavez, S. Mukkamala, Polymorphic malicious executable scanner by api sequence analysis, in: Fourth International Conference on Hybrid Intelligent Systems. HIS'۰٤, IEEE, ۲۰۰٤, pp. ۳۷۸–۳۸۳.

[۲۱] A. Shabtai, R. Moskovitch, Y. Elovici, C. Glezer, Detection of malicious code by applying machine learning classifiers on static features: a state-of-the-art survey, Information Security Technical Report ۱٤ (۱) (۲۰۰۹) ۱٦–۲۹.

[۲۲] G. Necula, Proof-carrying code, in: Proceedings of the ۲٤th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, ACM, ۱۹۹۷, pp. ۱۰٦–۱۱۹.

[۲۳] V. Paxson, Bro: a system for detecting network intruders in realtime, Computer Networks ۳۱ (۲۳–۲٤) (۱۹۹۹) ۲٤۳٥–۲٤٦۳.

[۲٤] S. Hofmeyr, S. Forrest, A. Somayaji, Intrusion detection using sequences of system calls, Journal of Computer Security ٦ (۳) (۱۹۹۸) ۱٥۱–۱۸۰.

[۲٥] J. Bergeron, M. Debbabi, M. Erhioui, B. Ktari, Static analysis of binary code to isolate malicious behaviors, in: IEEE Eighth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. (WET ICE'۹۹) Proceedings, IEEE, ۱۹۹۹, pp. ۱۸٤–۱۸۹.

[۲٦] D. Gao, M. Reiter, D. Song, Binhunt: automatically finding semantic differences in binary programs, Information and Communications Security (۲۰۰۸) ۲۳۸–۲٥٥.

[۲۷] S. Cesare, Y. Xiang, A fast flowgraph based classification system for packed and polymorphic malware on the endhost, in: ۲٤th IEEE International Conference on Advanced Information Networking and Applications (AINA), IEEE, ۲۰۱۰, pp. ۷۲۱–۷۲۸.

[۲۸] K. Jeong, H. Lee, Code graph for malware detection, in: International Conference on Information Networking. ICOIN ۲۰۰۸, IEEE, ۲۰۰۸, pp. ۱–٥.

[۲۹] T. Dullien, R. Rolles, Graph-based comparison of executable objects (English version), Symposium sur la se´ curite´ des technologies de l'information et des communications ٥ (۲۰۰٥) ۱–۳.

[۳۰] M. Abadi, M. Budiu, U. Erlingsson, J. Ligatti, Control-flow integrity principles, implementations, and applications, ACM Transactions on Information and System Security (TISSEC) ۱۳ (۱) (۲۰۰۹) ٤.

[۳۱] Hsien-De Huang, Chang-Shing Lee. TWMAN+: A Type-۲ Fuzzy Ontology Model for Malware Behavior Analysis, IEEE October ۱۴-۱۷, ۲۰۱۲.

[۳۲] K. Kim and B. R. Moon, "Malware detection based on dependency graph using hybrid genetic algorithm." In Proceedings of the ۱۲th annual conference on Genetic and evolutionary computation, July ۰۷-۱۱, ۲۰۱۰.

[۳۳] Aycock.J : "Computer Viruses and Malware". Springer, Heidelberg, ۲۰۰٦.

[۳٤] N.Idika, P.Mathur, "A Survey of Malware Detection Techniques", Department of Computer Science, Purdue University, West Lafayette, (۲۰۰۷), Vol.۲, ۱-۴۸.

[۳٥] M. Egele, T. Scholte, E. Kirda, C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools", In Proceedings of ACM Computer, (۲۰۱۲), Vol. ۴۴ ۶-۸.

[۳٦] M.Alazab, S.Venkataraman, P.Watters, "Towards Understanding Malware Behavior by the Extraction of API Calls", In Proceedings of the ۲۰۱۰ Second Cybercrime and Trustworthy Computing Workshop, Washington, DC,USA, (۲۰۱۰), ٥۲-٥۹.

[۳۷] B.Schreiber, Undocumented Windows ۲۰۰۰ Secrets: A Programmer's Cookbook, Addison Wesley Longman Publishing Co, Boston, MA, USA, ۲۰۰۱.

[۳۸] M.Ghiasi, A.Sami, Z.Salehi, "Dynamic Malware Detection Using Registers Values", ۹th International ISC Conference on Information Security and Cryptology, ۲۰۱۲

[۳۹] K. Loukhaoukha, J.Y. Chouinard, M.H. Taieb, Optimal image watermarking algorithm based on LWT-SVD via multi-objective ant colony optimization, Journal of Information Hiding and Multimedia Signal Processing ۲ (October (۴)) (۲۰۱۱) ۳۰۳–۳۱۹.

[٤۰] H.C. Huang, Y.H. Chen, Genetic fingerprinting for copyright protection of multicastmedia, Soft Computing ۱۳ (February (۴)) (۲۰۰۹) ۳۸۳–۳۹۱.

[٤۱] P. Puranik, P. Bajaj, A. Abraham, P. Palsodkar, A. Deshmukh, Human perceptionbased color image segmentation using comprehensive learning particle swarm optimization, Journal of Information Hiding and Multimedia Signal Processing ۲ (July (۳)) (۲۰۱۱) ۲۲۷–۲۳٥.

[٤۲] F.C. Chang, H.C. Huang, A refactoring method for cache-efficient swarmintelligence algorithms, Information Sciences, in press http://dx.doi.org/ ۱۰.۱۰۱۶/j.ins.۲۰۱۰.۰۲.۰۲٥

[٤۳] Secure Computing Corporation, Virus Signature Solutions from Secure Computing, http://www.securecomputing.com/, ۲۰۰۸.

[٤٤] M. Unterleitner, Computer Immune System for Intrusion and Virus Detection: Adaptive Detection Mechanisms and their Implementation, VMD Verlag Dr. Muller Aktiengesellschaft & Co., Germany, ۲۰۰۸, http://www.amazon.com/ Computer-Immune-System-Intrusion Detection/dp/۳۸۳۶۴۶۱۰۸۰

[٤٥] Z. Yu, L. Tao, Q. Renchao, Unknown computer virus detection inspired by immunity, Journal of Frontiers of Computer Science and Technology (۲۰۰۹), http://dx.doi.org/۱۰.۳۷۷۸/j.issn.۱۶۷۳-۹۴۱۸.۲۰۰۹.۰۲.۰۰۴, ISSN۱۶۷۳-۹۴۱۸/۲۰۰۳/۰۳ (۰۲)-۰۱٥۴-۰۸ http://www.ceaj.org/wes/qikan/manage/wenzhang/T۰۸۱۱۰۶۰.pdf

[٤٦] L. Castro, F. Zuben, Learning and optimization using the clonal selection principle, IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems ۶ (۳) (۲۰۰۲) ۲۳۹–۲٥۱ ftp://ftp.dca.fee.unicamp.br/ pub/docs/vonzuben/lnunes/ieee tec۰۱.pdf.

[٤۷] M. Creeger, The battle is bigger than most of us realize: CTO Roundtable:Malware Defense, Article development led by queue.acm.org, Communications of the ACM ٥۳ (April (۴)) http://portal.acm.org/citation.cfm?id=۱۷۲۱۶٥۴&coll=DL&dl=GUIDE&CFID=۱۰٥۳۳۴۳۳&CFTOKEN=۲۰۰٥۷۲۷۳, ۲۰۱۰.

[٤۸] K. Edge, G. Lamont, R. Raines, A Retrovirus Inspired Algorithm for Virus Detection & Optimization, Wright-Patterson AFB, Dayton, OH USA ۴٥۴۳۳, GECCO'۰۶, Washington, USA, ACM ۱-٥۹٥۹۳-۱۸۶-۴/۰۶/۰۷, http://portal.acm.org/citation.cfm?id=۱۱۴۴۰۱۶, ۲۰۰۶.

[٤۹] S. Forrest, A. Perelson, L. Allen, R. Cherukuri, Self-nonself discrimination in acomputer, in: Proceedings of IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Alamitos,CA,۱۹۹۴,pp.۳۶۰–۳۶٥, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=۱۰.۱.۱.۴۸.۳۲٥۸

[٥۰] J. Kephart, G. Sorkin, M. Swimmer, S. White, Blueprint for a computer immune system, in: This Paper was Originally Presented at the Virus Bulletin International Conference in San Francisco, California, USA, IBM ThomasJ.Watson Research Center, ۱۹۹۷, http://www.research.ibm.com/antivirus/SciPapers/Kephart/VB۹۷/

[٥۱] U. Aickelin, Artificial Immune Systems (AIS) – A New Paradigm for Heuristic Decision Making, The University of Nottingham, Nottingham,NG۸ ۱BB, United Kingdom, ۲۰۰۴, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=۱۰.۱.۱.۶۰.٥۹۲۳

[٥۲] Suha Afaneh,Raed Abu Zitar,Alaa Al-Hamami– Virus detection using clonal selection algorithm with Genetic Algorithm (VDC algorithm), Applied Soft Computing ۱۳(۲۰۱۳) ۲۳۹-۲۴۶

[٥٣] T.M. Chen, S. Abu-Nimeh. Lessons from stuxnet. Computer, IEEE, ۴۴(۴):۹۱–۹۳, ۲۰۱۱.

[٥٤] Muazzam Siddiquim, Morgan C. Wang, Joohan Lee. A survey of data mining techniques for malware detection using file features. in ACM-SE ۴۶ Proceedings of the ۴۶th Annual Southeast Regional Conference on XX, pp.۵۰۹–۵۱۰. ACM Press, ۲۰۰۸.

[٥٥] Nwokedi Idika, Aditya P. Mathur. A survey of malware detection techniques. http://www.serc.net/system/files/SERC-TR-۲۸۶.pdf, ۲۰۰۷. [Online; accessed ۱۵-Aug-۲۰۱۳].

[٥٦] Lakshmanan Nataraj, Vinod Yegneswaran, Phillip Porras Jian Zhang. A comparative assessment of malware classification using binary texture analysis and dynamic analysis. in AISec '۱۱ Proceedings of the ۴th ACM workshop on Security and artificial intelligence, pp. ۲۱–۳۰. ACM Press, ۲۰۱۱.

[٥٧] Lakshmanan Nataraj, S. Karthikeyan, Gregoire Jacob B.S. Manjunath. Malware images: Visualization and automatic classification. in International Symposium on Visualization for Cyber Security (VizSec), ۲۰۱۱.

[٥٨] A. Moser, C. Kruegel, E. Kirda. Limits of static analysis for malware detection. in Computer Security Applications Conference, ۲۰۰۷. ACSAC ۲۰۰۷. Twenty-Third Annual, pp. ۴۲۱–۴۳۰, ۲۰۰۷.

[٥٩] Konrad Rieck, Thorsten Holz, Carsten Willems Patrick Düssel Pavel Laskov. Learning and classification of malware behavior. in Detection of Intrusions and Malware, and Vulnerability Assessment, pp. ۱۰۸–۱۲۵.Springer Berlin Heidelberg, ۲۰۰۸.

[٦٠] C. Willems, T. Holz, F. Freiling. Toward automated dynamic malware analysis using cwsandbox. Security & Privacy, IEEE, ۵(۲):۳۲–۳۹, ۲۰۰۷.

[٦١] NormanShark.Normansandboxwhitepaper.http://download.norman.no/whitepapers/whitepaper_Norman_SandBox.pdf.[Online; accessed ۱۵-Aug-۲۰۱۳].

[٦٢] Anubis: Analyzing unknown binaries. http://anubis.seclab.tuwien.ac.at. [Online; accessed ۱۵- Aug-۲۰۱۳].۱۲۳

[٦٣] Gettys, Jim, Karlton, Phil, and McGregor, Scott. Automatic analysis of malware behavior using machine learning. Journal of Computer Security, ۱۹(٤):٦٣٩–٦٦٨, ۲۰۱۱.

[٦٤] Malheur: Automatic analysis of malware behavior. http://www.mlsec.org/malheur/. [Online; accessed ۱۵-Aug-۲۰۱۳].

[٦٥] scikit-learn: Machine learning in python. http://scikitlearn. org/stable/. [Online; accessed ۱۵-Aug-۲۰۱۳].

[٦٦] gpuminer: Parallel data mining on graphics processors. http://code.google.com/p/gpuminer/. [Online; accessed ۱۵-Aug-۲۰۱۳].

[٦٧] Apache mahout™ - scalable machine learning libraries. http://mahout.apache.org/. [Online; accessed ۱۵-Aug-۲۰۱۳].