۱. مقدمه

امروزه سیستم‌های کامپیوتری به طور مداوم مورد حمله‌های فزاینده و پیچیده قرار می‌گیرند. حمله‌کننده‌ها با توجه به نقاط آسیب‌پذیر سیستم، بدافزارها[۱] را تولید می‌کنند. اگر چه تلاش‌های مهمی در مقابله با بدافزارها انجام شده که آسیب‌پذیری سیستم‌ها را بسیار مشکل‌کرده است. اما همچنان تعداد نقاط آسیب‌پذیر و راه‌های ورودی به سیستم بسیار زیاد است. افزایش پیچیدگی بدافزارها شناسایی آنها را مشکل کرده است و یکی از چالش‌های اصلی ضد بدافزارها به محدودیت منابع مربوط می‌شود. برای تشخیص بی‌درنگ[۲] بدافزار باید تمام برنامه‌های در حال اجرا در تمام لحظات نظارت شوند. این مشکل در تلفن‌های همراه و یا دستگاه‌های مبتنی بر اینترنت اشیا که محدودیت حافظه وجود دارد و هزینه انرژی مصرفی قابل توجه است، حیاتی‌تر می‌شود.

طراحی و پیاده‌سازی سیستمی که سخت‌افزار را با توجه به ویژگی‌های سطح پایین[۳] آن نسبت به بدافزار آگاه کند تا بتواند بدافزارها را از برنامه‌های عادی در زمان اجرا جدا کند، می‌تواند کمک فزاینده ای به تشخیص بدافزارها نماید. منظور از ویژگی‌های سطح پایین اطلاعاتی هستند که بدست آوردن آنها در زمان اجرا به هیچ پردازشی احتیاج نداشته باشد. مانند الگوهای ارجاع داده[۴] و نرخ خطا حافظه نهان[۵].

---

[۱] Malwares

[۲] Real-time

[۳] Low-level features

[۴] Data reference patterns

[۵] Cache miss rates

۲. معرفی پروژه

همانطور که ذکر شد با تمرکز بر روی ویژگی‌های سطح پایین می‌توان برنامه‌ها را دسته بندی کرد. در این پروژه نیز از همین ویژگی‌ها برای پیاده‌سازی سیستمی استفاده می‌شود که بتواند در زمان مناسب و با استفاده از منابع کم احتمال بدافزار بودن برنامه‌ها را مشخص کند تا لایه نرم‌افزاری که هزینه محاسبه بیشتری دارد برنامه‌های خطرناک‌تر را بررسی کند و در نتیجه هزینه استفاده از آن کاهش یابد و عملکرد بهتری داشته باشد.

۳. معماری سیستم

در این سیستم ویژگیهای سطح پایین سخت افزاری که مورد بررسی قرار گرفته می‌شود به دو دسته تقسیم می‌شوند. دسته اول ویژگی های هستند که به آدرس‌دهی حافظه ارتباط دارند، به طور مثال هیستوگرام فرکانس آدرس حافظه[۶]. دسته دوم ویژگی های مربوط دستورالعمل‌ها[۷] هستند، برای نمونه می‌توان به وجود کد عملیاتی پردازه در حال اجرا، اشاره کرد[2].

در مرحله آموزش از داده‌هایی که توسط اجرای بدافزارهای مختلف بدست آمده اند استفاده می‌گردد و با استفاده از روش‌های یادگیری (به طور مثال رگرسیون لجستیک[۸]) سیستم را با توجه به ویژگی‌های که مشخص شده آموزش داده می‌شود.

این سیستم از اطلاعات سخت‌افزاری که در انتهای خط‌لوله‌ی پردازنده[۹] قرار می‌گیرد، بهره می‌برد و پس از این که دستورالعمل در مرحله انجام قرار گرفت پردازش لازم را بر روی ویژگی‌های ذکر شده انجام می‌دهد.

معماری کلی سیتسم تشخیص بد افزار به صورت دولایه است و این سیستم طراحی شده اولین لایه دفاعی می‌باشد.

---

[۶] Frequency of memory address distance histogram

[۷] Instructions

[۸] Logistic Regression

[۹] Processor pipeline

هدف این است که این لایه پردازه‌های در حال اجرا را بر اساس میزان خطرناک بودن مرتب کند به طوری که لایه دوم دفاعی که برنامه‌ای نرم افزاری بر روی سیستم عامل می‌باشد به بررسی پردازه‌های مشکوک‌تر اختصاص داده شود.

۴. کاربرد سیستم

امروزه با توجه به همه‌گیر شدن اینترنت اشیا و همچین ابزارک‌های همراه که شامل اطلاعات مهمی از کاربران آنها هستند، عملکرد نادرست سیستم می‌تواند خسارت‌های جبران ناپذیری را موجب شود، تامین امینت این اشیاء از اهمیت زیادی برخوردار است. همچین با توجه به اینکه منابعی که این اشیاء دارند محدود است باید بتوان با حداقل منابع امنیت اینگونه سیستم‌ها را به صورت بی‌درنگ تامین کرد. و همانطور که اشاره شد این پروژه می‌تواند کمک بسزایی به محقق شدن این امر بکند.

۵. نحوه ارزیابی

مجموعه ای از داده‌هایی که از اجرا شدن برنامه‌های سالم و بدافزار بدست آمده اند (به طور مثال داده‌های استفاده شده در مقاله [1]) و شامل ویژگی‌های سطح پایین می‌شوند را برای شبیه سازی به سیستم ارائه می تا عملکرد سیستم را در مواجهه با آنها بررسی کنیم. برای سنجش عملکرد شاخص‌های $true\ positives$ ، $true\ negatives$ ، $false\ positives$ و $false\ negatives$ مورد بررسی قرار می‌گیرند.

۶. مراحل انجام پروژه

۱- تعیین مشخصه‌های سخت‌افزاری مناسب برای شناخت بدافزار.
۲- انتخاب روش مناسب برای پیاده‌سازی با درنظر گرفتن سرعت و دقت تشخیص.
۳- پیاده‌سازی نرم‌افزار تشخیص‌دهنده و مستقرکردن آن بر روی برد Raspberry Pi.
۴- تولید و ارائه خروجی مناسب و کاربردی.

۷. منابع

[1] M. Ozsoy, K. N. Khasawneh, C. Donovick, I. Gorelik, N. Abu-Ghazaleh, and D. V. Ponomarev, “Hardware-based malware detection using low level architectural features,” in IEEE Trans. on Computers, vol. PP, no. 99, 2016.

[2] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Sethumadhavan, and S. Stolfo, “On the feasibility of online malware detection with performance counters,” in Proceedings of the 40th Annual International Symposium on Computer Architecture, ser. ISCA ’13. New York, NY, USA: ACM, 2013, pp. 559–570.