





دانشگاه الزهراء (س)
دانشکده فنی و مهندسی

پایان نامه

جهت اخذ درجه کارشناسی ارشد
رشته مهندسی کامپیوتر - گرایش هوش مصنوعی

عنوان

تشخیص ناهنجاری از طریق کنترل جامعیت داده هسته سیستم عامل مبتنی بر رویکرد سیستم
ایمنی مصنوعی

استادان راهنما

دکتر رضا عزمی
دکتر محسن افشارچی

دانشجو

سوده بهروزی نیا

اسفند ماه سال ۱۳۹۲

کلیه دستاوردهای این تحقیق متعلق به

دانشگاه الزهراء (س) است.

تقدیم

به پاس تعبیر عظیم و انسانی شان از کلمه اثار و از خودگذشتگی شان

به پاس عاطفه سرشار و گرمای امید بخش وجودشان که در این سردترین روزگار ان، بهترین پشتیبان است

به پاس قلب های بزرگ شان که فریاد رس است و سرگردانی و ترس در پناهشان به شجاعت می گراید

و

به پاس محبت های بی دریغشان که هرگز فروکش نمی کند

این مجموعه را به پدر و مادر فداکار و خواهر و برادر مهربانم تقدیم می کنم.

قدردانی

پاس بی کران پروردگار یکتا را که هستی مان، بخشد و به طریق علم و دانش، رهنمونان شد و به بهمنشینی رهروان علم و دانش مستخرمان نمود و

خوشه چینی از علم و معرفت را روزیای ساخت.

پاس خدای را که سخنوران، دستوران او بانند و شمارندگان، شردن نعمت های او ندانند و کوشندگان، حق او را کزاردن نتوانند. و سلام

و دور در محمد و خاندان پاک او، طاهران معصوم، هم آنان که وجودمان و مدار وجودشان است.

و با پاس از پدر و مادری به استواری کوه و مهربانی الهی که مرا الفبای زندگی آموختند و اتنان بیکران از مساعدت های بی شائبه ی استاد

عالیقدرم دکتر رضا عزمی که از محضر پر فیض تدریستان، بهره با برده ام.

باشد که بتوانم بخشی از زحمات بگی اتان را پاس گویم.

چکیده

یکی از چالش‌های مهم در مبحث امنیت سیستم‌های کامپیوتری، تشخیص نفوذ و فعالیت‌های نفوذی به سیستم می‌باشد. در همین راستا، سیستم‌عامل که به عنوان منبع مهم اجرای برنامه‌های کامپیوتری به حساب می‌آید، نقش مهمی در تامین امنیت اطلاعات ایفا می‌کند. سیستم‌های کامپیوتری به علت پیچیدگی و گستردگی، همیشه در معرض حملات و روت‌کیت‌ها قرار دارند. به همین جهت تشخیص نفوذ، هم‌اکنون به یکی از رویکردهای فعال در تحقیقات امنیتی تبدیل شده است. این تحقیقات دارای دو چالش عمده می‌باشند: نخست بستر جمع‌آوری امن اطلاعات و دوم ایجاد روشی دقیق و مبتنی بر رفتار روت‌کیت‌ها برای تشخیص نفوذ.

در این پژوهش، به منظور تشخیص نفوذ و مقابله با حملات و روت‌کیت‌های سطح هسته، سیستم تشخیص نفوذی تحت عنوان $KLrtD^1$ ارائه شده است. این سیستم نوعی معماری امن و مبتنی بر ناظر دارد که از یک سو با بکارگیری ابزار طراحی شده‌ی $KLdG^2$ به جمع‌آوری داده‌های مورد نیاز از سطح هسته می‌پردازد و از سویی دیگر با بهره‌گیری از روشی الهام گرفته از تئوری خطر و خروجی سه ابزار طراحی شده‌ی $KLrtD_IC^3$ ، $KLrtD_AD^4$ و $KLrtD_SB^5$ که به ترتیب از روش‌های کنترل جامعیت، تشخیص ناهنجاری و تشخیص مبتنی بر امضا برای تولید سیگنال‌های ورودی استفاده می‌کنند، حملات، روت‌کیت‌ها و ناهنجاری‌های سیستمی را تشخیص می‌دهد.

نتایج حاصل از ارزیابی‌ها، حاکی از این است که سیستم $KLrtD$ به علت استفاده از یک روش ترکیبی از تکنیک‌های مختلف و رویکردی مبتنی بر ناظر، به خوبی روت‌کیت‌های سطح هسته را شناسایی کرده و قادر است هم‌ردیف سیستم‌های تشخیص نفوذ مبتنی بر میزبان و در مواقعی، بهتر از آن‌ها، به تشخیص روت‌کیت‌ها و بدافزارهای سطح هسته بپردازد.

کلمات کلیدی: روت‌کیت، کنترل جامعیت، تشخیص ناهنجاری، تشخیص مبتنی بر امضا، سیستم ایمنی مصنوعی، تئوری خطر، تکنولوژی ناظر

¹ Kernel Level rootkit Detection (KLrtD)

² Kernel Level data Gathering (KLdG)

³ KLrtD_Integrity Checking (KLrtD_IC)

⁴ KLrtD_Anomaly Detection (KLrtD_AD)

⁵ KLrtD_Signature Based (KLrtD_SB)

فهرست مطالب

فصل اول : مقدمه و هدف پژوهش	۱
۱-۱- مقدمه	۲
۲-۱- رویکردهای کلیدی	۳
۳-۱- سازماندهی مستند	۷
۴-۱- خلاصه فصل	۸
فصل دوم : مفاهیم مرتبط	۸
۱-۲- مقدمه	۹
۲-۲- نفوذ	۹
۱-۲-۲- روت کیت‌ها	۱۰
۲-۲-۲- سیر تکاملی روت کیت‌ها	۱۱
۳-۲-۲- مکانیزم‌های متداول حملات	۱۴
۳-۲- سیستم‌های تشخیص نفوذ	۱۶
۱-۳-۲- منبع اطلاعات دریافتی	۱۷
۲-۳-۲- روش تشخیص نفوذ	۱۹
۳-۳-۲- زمان پاسخ	۲۰
۴-۲- سیستم‌های ایمنی انسانی	۲۰
۵-۲- سیستم‌های ایمنی مصنوعی	۲۲
۱-۵-۲- ویژگی‌های سیستم ایمنی مصنوعی برای سیستم‌های تشخیص نفوذ	۲۳
۶-۲- فناوری مجازی سازی	۲۴
۱-۶-۲- تکنیک‌های مجازی سازی	۲۵
۲-۶-۲- مانیتور ماشین مجازی	۲۶
۳-۶-۲- ماشین ناظر Xen	۲۸
۷-۲- خلاصه فصل	۳۱
فصل سوم : پژوهش‌های مرتبط	۳۲

۳۳	۱-۳-مقدمه
۳۳	۲-۳-سیستم‌های تشخیص مبتنی بر محیط عملکرد
۳۴	۱-۲-۳-تکنیک‌های مبتنی بر میزبان
۳۶	۲-۲-۳-تکنیک‌های مبتنی بر مجازی‌سازی
۳۸	۳-۲-۳-تکنیک‌های مبتنی بر ناظر خارجی
۴۰	۳-۳-انواع روش‌های تشخیص
۴۰	۱-۳-۳-تشخیص بر مبنای امضا
۴۱	۲-۳-۳-تشخیص بر مبنای جامعیت
۴۵	۳-۳-۳-تشخیص بر مبنای ناهنجاری
۴۹	۴-۳-۳-تشخیص بر مبنای سیستم ایمنی مصنوعی
۵۱	۴-۳-خلاصه فصل
۵۲	فصل چهارم : جمع‌آوری مجموعه داده‌ها
۵۳	۱-۴-مقدمه
۵۳	۲-۴-معماری ابزار <i>KLdG</i>
۵۵	۳-۴-ثبات‌ها
۵۵	۱-۳-۴-ثبات‌های کنترلی
۵۶	۲-۳-۴-ثبات دیباگ و پرچم، ثبات‌های توصیف‌گر و ثبات‌های خاص منظوره
۵۷	۴-۴-فراخوانی‌های سیستمی
۵۷	۱-۴-۴-جدول فراخوانی سیستمی
۵۸	۲-۴-۴-دنباله‌ای از فراخوانی‌های سیستمی
۶۰	۵-۴-ساختار داده‌های حیاتی
۶۱	۱-۵-۴-توصیف‌گر پردازش
۶۲	۲-۵-۴-مدیریت حافظه
۶۳	۳-۵-۴-نگاشت پردازش
۶۵	۴-۵-۴-ساختار داده <i>zone_struct</i>
۶۵	۵-۵-۴-مولد تولید اعداد تصادفی

۶۶	max_thread	پارامتر	۶-۵-۴
۶۶	rtc_fops	ساختار داده	۷-۵-۴
۶۷	/dev/kmem و /dev/mem	فایل‌های	۸-۵-۴
۶۷	Proc-sys	مجموعه داده	۶-۴
۶۹		خلاصه فصل	۷-۴
۷۰	KLrtD	ارائه سیستم تشخیص	فصل پنجم
۷۱		مقدمه	۱-۵
۷۳		الهام زیستی	۲-۵
۷۶		معماری مفهومی	۳-۵
۷۹		فاز استخراج داده و کدگشایی	۱-۳-۵
۸۱		فاز آموزش و تولید سیگنال‌های ورودی	۲-۳-۵
۹۳		فاز تشخیص	۳-۳-۵
۹۵		خلاصه فصل	۴-۵
۹۷		ارزیابی سیستم	فصل ششم
۹۸		مقدمه	۱-۶
۹۹		معیارهای ارزیابی	۲-۶
۱۰۰		معیار نرخ مثبت اشتباه	۱-۲-۶
۱۰۰		معیار نرخ مثبت صحیح	۲-۲-۶
۱۰۱		معیار نرخ منفی اشتباه	۳-۲-۶
۱۰۱		معیار نرخ منفی صحیح	۴-۲-۶
۱۰۱		معیار دقت	۵-۲-۶
۱۰۲	ROC	معیار مساحت زیر نمودار	۶-۲-۶
۱۰۲		منابع تولید سیگنال‌های ورودی	۳-۶
۱۰۳		ارزیابی سیگنال امن	۱-۳-۶
۱۱۰	PAMP	ارزیابی سیگنال	۱-۳-۶
۱۱۱		ارزیابی سیگنال خطر	۲-۳-۶

۱۱۴	۴-۶- تحلیل نتایج و قدرت تشخیص سیستم پیشنهادی
۱۱۸	۵-۶- تحلیل سیستم پیشنهادی
۱۲۳	۶-۶- خلاصه فصل
۱۲۴	فصل هفتم : نتیجه‌گیری
۱۲۵	۱-۷- مقدمه
۱۲۶	۲-۷- پژوهش‌های آینده
۱۲۷	۳-۷- خلاصه فصل

فهرست جدول ها

- جدول ۱-۲ جزئیات معماری مجازی سازی در *xen*..... ۲۹
- جدول ۱-۳ مقایسه تکنیک های متفاوت محیط عملکرد با معیارهای گوناگون..... ۳۴
- جدول ۲-۳ دسته بندی روش های موجود در حوزه ی تشخیص..... ۴۰
- جدول ۱-۶ نمایش چهار مفهوم پایه در ارزیابی قدرت تفکیک یک الگوریتم..... ۱۰۰
- جدول ۲-۶ جزئیات پیکربندی محیط ارزیابی سیستم..... ۱۰۳
- جدول ۳-۶ روت کیت های قابل شناسایی توسط ابزار *KLrtD_IC*..... ۱۰۷
- جدول ۴-۶ نتایج ارزیابی روش درخت تصمیم برای طول دنباله های مختلف..... ۱۰۸
- جدول ۵-۶ نتایج ارزیابی روش درخت تصمیم برای دنباله هایی به طول ۳..... ۱۰۹
- جدول ۶-۶ روت کیت های قابل تشخیص توسط امزاهای لیست سیاه..... ۱۱۰
- جدول ۷-۶ نتایج ارزیابی ابزار *KLrtD_AD* با تعداد درخت های متفاوت..... ۱۱۲
- جدول ۸-۶ نتیجه ارزیابی سیستم *KLrtD*..... ۱۱۷
- جدول ۹-۶ لیست روت کیت های قابل شناسایی در سیستم *KLrtD*..... ۱۱۷
- جدول ۱۰-۶ مقایسه سه ابزار *HookSafe*، *Gibraltar* و *KLrtD* با روت کیت های قابل تشخیص..... ۱۲۰

فهرست شکل‌ها

- شکل ۱-۲ محیط ماشین مجازی. (الف) نوع اول: قرارگیری ناظر، مستقیماً بالای سخت‌افزار. (ب) نوع دوم: قرارگیری ناظر بر روی سیستم‌عامل میزبان ۲۷
- شکل ۲-۲ ساختار ماشین مجازی *Xen* ۲۸
- شکل ۳-۲ ساختار لایه ای *Xen* ۳۰
- شکل ۱-۳ معماری بستر تست *Copilot* ۳۹
- شکل ۱-۴ معماری جمع‌آوری داده در سطح ناظر توسط *KLdG* ۵۴
- شکل ۲-۴ ثبت داده‌های ثابت کنترلی ۵۶
- شکل ۳-۴ ثبت‌های دیباگ ۵۶
- شکل ۴-۴ ثبت‌های توصیف‌گر ۵۶
- شکل ۵-۴ ثبت‌های خاص منظوره ۵۷
- شکل ۶-۴ جدول فراخوانی سیستمی ۵۸
- شکل ۷-۴ اسکریپت ثبت فراخوانی‌های سیستمی ۵۹
- شکل ۸-۴ قسمتی از فایل حاوی فراخوانی‌های سیستمی ۵۹
- شکل ۹-۴ قسمتی از دنباله‌های استخراج‌شده به طول ۳ ۶۰
- شکل ۱۰-۴ دنباله‌ای از ارتباط برخی از ساختار داده‌ها ۶۰
- شکل ۱۱-۴ سورس کد ماژول بازیابی آفست ۶۱
- شکل ۱۲-۴ دسترسی به مقادیر ساختار داده *task_struct* ۶۲
- شکل ۱۳-۴ دسترسی به مقادیر ساختار داده‌ی *mm_struct* ۶۳
- شکل ۱۴-۴ دسترسی به مقادیر ساختار داده‌ی *vm_area_struct* ۶۳
- شکل ۱۵-۴ ارتباط ساختار داده‌های اساسی مرتبط با فایل ۶۴
- شکل ۱۶-۴ نحوه دسترسی به ساختار داده *zone_struct* ۶۵
- شکل ۱۷-۴ نحوه دستیابی به ساختارهای *random_state_ops* و *unrandom_state_ops* ۶۶
- شکل ۱۸-۴ نحوه دستیابی به ساختار *rtc_fops* ۶۷

- شکل ۴-۱۹ نحوه دسترسی به *mem_fops* و *kmem_fops* ۶۷
- شکل ۴-۲۰ ویژگی‌های مجموعه داده *Proc_sys* ۶۸
- شکل ۴-۲۱ اسکریپت ثبت اطلاعات مجموعه داده *Proc_sys* ۶۸
- شکل ۵-۱ نحوه عملکرد سیستم ایمنی طبیعی ۷۵
- شکل ۵-۲ معماری سیستم *KLrtD* ۷۷
- شکل ۵-۳ فلوجارت سیستم *KLrtD* ۷۸
- شکل ۵-۴ مولفه‌های مهم ابزار *KLrtD* ۸۰
- شکل ۵-۵ قالب مشترک برای امضاهای لیست سفید مبتنی بر ویژگی ۸۵
- شکل ۵-۶ قالب مشترک برای امضاهای لیست سفید مبتنی بر درخت تصمیم ۸۸
- شکل ۵-۷ نمونه امضاهای جمع‌آوری شده برای شناسایی روت‌کیت‌ها ۸۹
- شکل ۶-۱ گراف *ROC* برای نمایش قدرت یک طبقه‌بند فرضی ۱۰۲
- شکل ۶-۲ نمونه امضاهای تولید شده با روش مبتنی بر ویژگی ۱۰۴
- شکل ۶-۳ درخت تصمیم تولید شده از دنباله فراخوانی‌های سیستمی ۱۰۵
- شکل ۶-۴ نمونه امضاهای تولید شده با روش مبتنی بر ویژگی ۱۰۶
- شکل ۶-۵ نمودار *ROC* برای *KLrtD_AD* در اثر افزایش تعداد طول دنباله‌ها ۱۰۸
- شکل ۶-۶ نمودار *ROC* برای *KLrtD_AD* در اثر افزایش داده‌های آموزش برای طول دنباله ۳..... ۱۰۹
- شکل ۶-۷ نمونه امضاهای لیست سیاه ۱۱۰
- شکل ۶-۸ نمودار *ROC* برای *KLrtD_AD* در اثر افزایش تعداد درخت‌ها ۱۱۳
- شکل ۶-۹ نمودار میزان پیشرفت *KLrtD_AD* در اثر افزایش تعداد درخت‌ها بر اساس *FP* و *TP* ۱۱۳
- شکل ۶-۱۰ نمودار میزان پیشرفت *KLrtD_AD* در اثر افزایش تعداد درخت‌ها بر اساس *Acc* ۱۱۴
- شکل ۶-۱۱ نمودار *ROC* سیستم *KLrtD* ۱۱۷

فصل اول : مقدمه و هدف پژوهش

۱-۱- مقدمه

رشد باورنکردنی بدافزارهای کامپیوتری و از طرفی تمایل بیش از پیش کاربران به استفاده از محیط‌های شبکه‌ای و توزیع شده به منظور اشتراک گذاشتن اطلاعات، سبب شده است که اهمیت امنیت در سیستم‌های کامپیوتری خودنمایی بیشتری داشته باشد. با توجه به همین موضوع، یکی از چالش‌های اساسی در این حوزه، بحث امنیت و تشخیص فعالیت‌های خرابکارانه و هرگونه نفوذ به آن‌ها می‌باشد که جامعیت^۱، محرمانگی^۲ و یا دسترسی^۳ به یک منبع را در یک سیستم کامپیوتری به خطر می‌اندازد. با توجه به تلاش‌های فراوان برای مقابله با فعالیت‌های خرابکارانه، امروزه دیگر صحبت از طراحی یک سیستم اطلاعاتی بدون در نظر گرفتن جوانب امنیتی آن امری منسوخ است.

حملات و نفوذ به سیستم‌های کامپیوتری از طریق بدافزارها^۴ انجام می‌شوند. بدافزارها، یکی از ابزارها و اقدامات ضد امنیتی هستند و به برنامه‌هایی اطلاق می‌شوند که بدون اجازه صاحب سیستم، قصد انجام کارهای ناخواسته یا خرابکارانه را در سیستم دارند. ویروس‌ها، کرم‌واره‌ها، اسب‌های تراوا^۵، شماره‌گیرها^۶، درهای پشتی^۷، نرم‌افزارهای جاسوسی^۸ و روت‌کیت‌ها^۹ نمونه‌ای از انواع بدافزارهای موجود در دنیای امنیت می‌باشند. به دلیل طبیعت تغییرپذیر، روبه‌رشد و گسترده این مقوله، مقابله با آن‌ها یکی از مسائل چالش‌برانگیز و ضروری در امنیت سیستم‌های کامپیوتری محسوب می‌شود.

روت‌کیت‌ها بدافزارهایی هستند که اغلب، آن‌ها را به خودی خود نمی‌توان مخرب یا خطرناک دانست، بلکه قرارگرفتن آن‌ها در کنار ویروس‌ها یا کرم‌واره‌های اینترنتی یا نوع

¹ Integrity

² Confidentiality

³ Availability

⁴ Malware

⁵ Trojan Horse

⁶ Dialer

⁷ Backdoor

⁸ Spyware

⁹ Rootkit

استفاده از آنها است که به آنان ماهیتی خطرناک می‌بخشد. شناسایی روت‌کیت بسیار مشکل‌تر از بدافزارهای دیگر است زیرا روت‌کیت‌ها جایگزین برنامه‌های اجرایی مهم سیستم‌عامل شده و در گاهی مواقع جایگزین خود هسته می‌شوند و به دیگر بدافزارها این اجازه را می‌دهند که از طریق درب‌پستی و پنهان‌شدن در سیستم‌عامل به آن نفوذ کنند. با توجه به همین مشکلات و همین‌طور قدرت غیرقابل انکار روت‌کیت‌ها در حمله و نفوذ به سیستم، تشخیص و مقابله با آنها یکی از مباحث مهم امنیتی به‌شمار می‌رود.

در همین راستا، هدف اصلی این پژوهش ارائه‌ی یک سیستم تشخیص نفوذ مبتنی بر میزان است که تا حدی با بهبود کارهای پیشین، گامی در جهت رسیدن به یک سیستم امن در شناسایی بدافزارهای پنهان سطح هسته بردارد. در ادامه، رویکردهای کلیدی این پژوهش در بخش ۱-۲ و شیوه‌ی سازماندهی مستند در بخش ۱-۳ بررسی می‌شود. در نهایت بخش ۱-۴ خلاصه‌ای از مطالب ذکرشده در این فصل را ارائه می‌دهد.

۱-۲- رویکردهای کلیدی

رویکرد اصلی این پژوهش، ارائه‌ی یک سیستم تشخیص نفوذ مبتنی بر میزان می‌باشد. ارائه و پیاده‌سازی یک سیستم جدید در حوزه‌ی تشخیص نفوذ، نیازمند بررسی دقیق مسائل گوناگونی نظیر مکان پیاده‌سازی، داده‌های موردنیاز، تحلیل روت‌کیت‌ها، نحوه‌ی تولید امضاها، جمع‌آوری امضاها و موجود و نیز استفاده از روش تشخیصی مناسب می‌باشد.

سیستم‌های تشخیص نفوذ، بر اساس موقعیت قرارگیری و دریافت منبع اطلاعاتی، به دو دسته‌ی مبتنی بر شبکه و مبتنی بر میزان تقسیم می‌شوند. سیستم‌های مبتنی بر شبکه^۱ (*NIDS*) [۳]، از بسته‌ها و ترافیک شبکه به عنوان منبع اصلی اطلاعات برای تصمیم‌گیری پیرامون نفوذ استفاده می‌کنند. سیستم‌های مبتنی بر میزان^۲ (*HIDS*) [۴] که در این پژوهش مدنظر قرار گرفته شده‌اند، با بررسی و ارزیابی رفتارهای میزان، دید سیستمی و قدرت بالایی در تشخیص نفوذ دارند.

¹ Network based Intrusion Detection System (*NIDS*)

² Host based Intrusion Detection System (*HIDS*)

سیستم‌های تشخیص مبتنی بر میزبان را از لحاظ مکان پیاده‌سازی می‌توان به سه سطح کاربری، هسته سیستم‌عامل و سیستم ناظر طبقه‌بندی کرد. این سیستم‌ها در سطح کاربر مانند یک نرم‌افزار کاربردی می‌توانند پردازش‌های در حال اجرا و رفتار آن‌ها را کنترل کنند، اما نمی‌توانند از خود محافظت کرده و دید جامعی به اطلاعات درون هسته داشته باشند [۵،۶]. دسته‌ی دیگری از این روش‌ها مانند [۷] در سطح هسته سیستم‌عامل و با ایجاد تغییرات در کد هسته بر عملیات حساس نظارت دارند. این دسته عموماً به بررسی داده‌های ایستا و تنها نحوه‌ی دسترسی به داده‌های حساس می‌پردازند و از دید روت‌کیت‌ها کاملاً محافظت نمی‌شوند [۸،۹]. گفتنی است برخی از راه‌حل‌های محدود نرم‌افزار مانند [۱۰] که برای حفاظت از زیر بخش‌های هسته معرفی شده‌اند، در حال حاضر هیچ‌کدام اثبات کامل و مطمئنی نسبت به امنیت روش خود ارائه نکرده‌اند.

با توسعه‌ی فناوری ماشین مجازی که در بسیاری از محیط‌ها مورد استفاده قرار گرفته شده است، نظارت بر رویدادهای سطح هسته و حفاظت از داده‌ها نیز بر بستر این فناوری و در ناظر توسعه یافته است. به طوری که می‌توان ادعا کرد یکی از عوامل پیشرفت این فناوری امکان نظارت بیرونی بر ماشین مجازی (VMI) بوده است [۲،۱۱،۱۲،۱۳،۱۴]. استفاده از این فناوری توسط [۱۱] و با بکارگیری VMI در شناسایی بدافزارها مورد توجه بیشتری قرار گرفته است. پس از آن *Lares* [۵]، *SIM* [۱۰]، *TraDic* [۲]، *Wizard* [۹]، *HyperSafe* [۱۲] و *HyperCheck* [۱۳] برای کنترل جامعیت هسته در ناظر قرار گرفته و از فناوری مجازی‌سازی بهره بردند.

بنابراین این پژوهش به دلایل گوناگون از قبیل وجود روت‌کیت‌های سطح هسته و نیز نیاز به اجازه‌ی دسترسی سطح بالا به هسته و ساختار داخلی آن، سیستم تشخیص نفوذ پیشنهادی را درون لایه‌ی ناظر تعبیه نموده است. استفاده از تکنولوژی مجازی‌سازی، سیستم تشخیص نفوذ را به‌وسیله‌ی ایزوله‌سازی کامل آن از سایر لایه‌های نرم‌افزار سیستم، در برابر حملات مقاوم‌تر می‌سازد [۱۴].

این سیستم پیشنهادی برای اعمال روش‌های تشخیص، نیازمند جمع‌آوری داده‌های مناسبی برای تشخیص و تفکیک رفتار هنجار یا ناهنجار سیستم میزبان می‌باشد. برای بررسی همه‌جانبه و جامع از هسته‌ی سیستم‌عامل میزبان و پردازش‌های در حال اجرا، بایستی اطلاعات

کاملی نظیر ساختار داده‌های حیاتی شامل داده‌های کنترلی و غیرکنترلی، ثبات‌ها، دنباله فراخوانی‌های سیستمی و همین‌طور اطلاعات پدازه‌های در حال اجرا را از سراسر هسته‌ی سیستم جمع‌آوری کرد. تاکنون جمع‌آوری داده‌ها در محیط‌های گوناگون نظیر محیط بیرونی، محیط داخل میزبان و محیط ناظر صورت پذیرفته است [۱۵، ۱۶، ۱۷، ۱۸]. این پژوهش جمع‌آوری داده‌ها را در لایه‌ی ناظر انجام داده است تا عملیات را به صورت امن و به‌دور از گزند روت‌کیت‌های سطح هسته به انجام برساند.

به طور کلی، سیستم‌های تشخیص نفوذ با دو رویکرد کلی، عملیات تشخیص را دنبال می‌نمایند. در رویکرد اول که اصطلاحاً تشخیص مبتنی بر امضا^۱ نامیده می‌شود، رفتارهای ناهنجار تحت عنوان امضای بدافزار مدل می‌شوند و یک نفوذ، زمانی کشف می‌شود که با یکی از امضاهای بدافزار، تطبیق داشته باشد. این سیستم‌ها به جای استفاده از امضاهای بدافزار می‌توانند از امضاهای امن استفاده کنند. در این حالت سیستم زمانی در حالت امن تشخیص داده می‌شود که با امضاهای امن تطبیق داشته باشد. اما رویکرد دوم که اصطلاحاً تشخیص ناهنجاری^۲ نامیده می‌شود، به مدل‌سازی رفتارهای هنجار سیستم می‌پردازد و یک نفوذ، زمانی کشف می‌شود که با مدل هنجار تطبیق نداشته باشد.

رویکرد تشخیص مبتنی بر امضا در شناسایی نفوذهای شناخته‌شده بسیار قدرتمند است و به همین سبب، هم‌اکنون در بسیاری از سیستم‌های تجاری به‌کار گرفته شده است؛ اما نقص آن در عدم توانایی کشف و شناسایی رفتارهای نفوذی جدیدی می‌باشد که امضای آن‌ها برای سیستم شناخته شده نیست. رویکرد تشخیص ناهنجاری با هدف برطرف‌سازی این نقص و تشخیص نفوذهای ناشناخته ارائه شده است. با توجه به همین موضوع، سیستم ارائه شده از تلفیق دو روش برای تشخیص نفوذ در سیستم استفاده می‌کند تا همزمان بتواند از مزایای هر دو استفاده نموده و کاستی‌های هر یک را تحت پوشش قرار دهد.

این پژوهش درصدد است تا سیستم پیشنهادی خود را مبتنی بر سیستم‌های ایمنی مصنوعی^۳ ارائه دهد. سیستم‌های ایمنی مصنوعی به دلیل شباهت زیاد با سیستم ایمنی انسانی

¹ Signature based Detection

² Anomaly Detection

³ Artificial Immune System (AIS)

و شبیه‌سازی مکانیزم دفاعی بدن، گزینه‌ی مناسبی برای سیستم‌های تشخیص به‌حساب می‌آیند.

به منظور جمع‌آوری کلیه‌ی داده‌های مورد نیاز برای سیستم تشخیص، در این پژوهش ابزاری تحت عنوان *KLdG* طراحی شده است که با نوعی معماری امن و مبتنی بر ناظر به جمع‌آوری امن اطلاعات در لایه‌ی ناظر یا مانیتور ماشین مجازی^۱ می‌پردازد. در ادامه سیستمی نوین تحت عنوان *KLrtD* مبتنی بر تئوری خطر [۱۹،۲۰] در سیستم ایمنی ارائه می‌شود که با سه مولفه‌ی مختلف با عناوین *KLrtD_IC*، *KLrtD_AD* و *KLrtD_SB* به جمع‌آوری سیگنال‌های لازم در تشخیص نفوذ می‌پردازد.

سیستم *KLrtD* برای تولید سیگنال امن^۲ از روش تشخیص مبتنی بر امضای لیست سفید^۳ برای کنترل جامعیت هسته‌ی سیستم‌عامل در مولفه *KLrtD_IC* استفاده می‌کند. این سیستم از سویی دیگر برای تولید سیگنال خطر^۴ و *PAMP* به ترتیب از روش‌های تشخیص ناهنجاری و تشخیص مبتنی بر امضای لیست سیاه^۵ در مولفه‌های *KLrtD_AD* و *KLrtD_SB* استفاده می‌نماید.

سیستم *KLrtD* همان‌طور که بیان شد، سیگنال‌های مورد نیاز خود را از طریق مولفه‌های مختلف طراحی‌شده، تولید می‌کند. این سیستم به‌دلیل استفاده از داده‌های آموزش جامع و همچنین ترکیبی مناسب از انواع روش‌های تشخیص، کارایی خوبی در میان سیستم‌های تشخیص نفوذ مبتنی بر میزبان در تشخیص روت‌کیت‌های سطح هسته دارد. به این ترتیب سیستم *KLrtD* می‌تواند با بهره‌گیری همزمان از مزایای تکنولوژی ناظر و تئوری خطر، از گزند روت‌کیت‌های سطح هسته مصون بماند و با روشی دقیق و الهام گرفته از سیستم ایمنی انسانی، به تفکیک رفتار هنجار و ناهنجار سیستم بپردازد.

¹ *Virtual Machine Monitor*

² *Safe*

³ *White list*

⁴ *Danger*

⁵ *Black list*