ششمین کنفرانس بین المللی انجمن رمز ایران
۱۵ و ۱۶ مهر ماه ۱۳۸۸، دانشگاه اصفهان
انجمن رمز ایران
دانشگاه اصفهان

تشخیص میزبان‌های بدخواه در سیستم پلیس عامل ها بر اساس استخراج نشانه های حمله

نسیبه محمودی، بهروز ترک لادانی

دانشکده فنی مهندسی، گروه کامپیوتر، دانشگاه اصفهان، اصفهان، ایران
ladani@eng.ui.ac.ir, mahmoodi@comp.ui.ac.ir

چکیده

یکی از مهمترین موانع در مقابل گسترش کاربردهای مبتنی بر عامل، مشکل امنیتی عامل‌ها و میزبان‌های عامل است. تشخیص میزبان‌های بدخواه موضوعی است که در سال‌های اخیر بیشتر مورد توجه محققان قرارگرفته است. اما اکثر روش‌های پیشنهادی، تشخیص بدخواه بودن میزبان‌ها را بر عهده خود عامل‌ها قرار می‌دهد که نه تنها برای عامل زمانگیر است، بلکه عامل را از هدف اصلی خود دور می‌کند. در کارهای قبلی مدلی برای سیستم پلیس در محیط عامل های متحرک پیشنهاد شده است که در آن تشخیص بدخواه بودن میزان‌ها بر عهده پلیس عامل ها است. در این مقاله علاوه بر اصلاح نحوه کار عامل های پلیس، روش جدیدی برای تصمیم گیری راجع به میزان بدخواه بودن میزبان ها با استفاده از یک سیستم خبره مبتنی بر قوانین غیر قطعی ارائه نموده ایم. برای انجام این کار با اعزام عامل های بی نشان پلیس به میزبان های مشکوک، نشانه های حمله براساس تاثیری که روی عامل به جای می‌گذارند استخراج و دسته بندی شده، سپس به کمک سیستم خبره پلیس میزان بدخواه بودن میزبان ها تعیین می شود.

واژه‌های کلیدی

سیستم خبره پلیس، عامل پلیس، تشخیص حمله، اعتماد، نشانه حمله، میزبان بدخواه.

۱- مقدمه

روش های زیادی در طی سال های اخیر برای محافظت از عامل های متحرک پیشنهاد شده است[۵-۱] که برخی روی تشخیص حمله و برخی روی جلوگیری از حمله به عامل تاکید دارند. هدف از مکانیسم‌های جلوگیری این است که مداخله در عامل توسط میزبان یا بسیار مشکل باشد و یا برای او هزینه بالائی را درپی داشته‌باشد، این مکانیسم ها زمینه تحقیقاتی فعلی اکثر محققان امروزی است. تعدادی از مکانیسم هایی که به این منظور ارائه شده عبارتنداز[۴،۵]: سخت‌افزار مقاوم در برابر حمله[۱]، تماس با خانه[۲]، استفاده از سیستم چندعاملی[۳] و به‌هم‌ریختن کد[۴].

در روش‌های تشخیص مداخله، هویت میزبان بدخواه ردیابی شده و سوء رفتار آن اثبات می‌شود. برخی از روش‌های تشخیص مداخله نیز عبارت است از [۴ و۵]: اشیاء تشخیصی[۵]، تاریخچه مسیر[۶]، ثبت دو جانبه سفرنامه[۷]، تکرار میزبان[۸]، حالت های مرجع[۹].

روش‌های جلوگیری از مداخله اکثراً نیاز به توابع محاسباتی سنگین دارند و بنا بر این با اهداف ایجاد عامل (متحرک بودن و نیاز به پهنای باند کم) تناقض دارند. تحقیقات روی مکانیسم های جلوگیری از مداخله هنوز ادامه دارد. تکنیک های تشخیص نیز در واقع بعد از وقوع حمله است و نیاز عامل رادر همان زمان برآورده نمی‌کند.

[1] Tamper Resistant Hardware
[2] Phone Home
[3] Multi Agent System
[4] Code Obfuscation
[5] Detection object
[6] Path history
[7] Mutual itinerary recording
[8] Replication server
[9] References state

تعدادی از روش های جلوگیری از مداخله نیز وجود دارند که از مکانیسم های تشخیص استفاده می‌کنند. این روش‌ها، مکانیسم-های جلوگیری غیرفعال نامیده می‌شوند که سعی می‌کنند از روش‌های سازمانی و وابسته به معماری استفاده کنند. مکانیسم-های غیرفعال مزیت‌های عامل مثل خودمختاری و مهاجرت را تضمین می‌کنند. در مقابل، مکانیسم‌های جلوگیری فعال، عامل را از حمله میزبان بدخواه دور می کند اما مزیت های عامل را حفظ نمی‌کند [۷].

ما در کار قبلی خود [۱]، مدلی به نام سیستم پلیس عامل ها ارائه نموده ایم که مبنای کار آن بر اساس محاسبه اعتماد است. سیستم پلیس به سابقه همه میزبان‌ها و عامل‌ها دسترسی دارد و تصمیمی که درباره قابل اعتماد بودن می‌گیرد دقیق است. سیستم پلیس عامل ها با مجهز بودن به روش‌های مختلف تشخیص میزبان های بدخواه (از طریق عامل‌های پلیس) و دسترسی به اطلاعاتی از آنها که در اختیار همه نیست، میزان امن بودن هر میزبان را مشخص می‌نماید. بنابراین، هر عامل بدون نیاز به داشتن سابقه‌ی میزبان و یا نیاز به حمل کد های سنگین، فقط با استعلام از سیستم عامل های پلیس به نحو مناسب می تواند به راحتی و امنیت به فعالیت خود ادامه دهد. در این سیستم، پلیس برای محاسبه اعتماد از ۵ نوع مختلف از مفاهیم اعتماد، شهرت و خطر برای محاسبه نهایی اعتماد استفاده می کند:

۱- گزارش میزبان همسایه درباره میزبان مورد بررسی (شهرت).
۲- عامل هایی که با این میزبان تراکنش انجام داده‌اند. (شهرت)
۳- مدت فعالیت (اعتماد)
۴- گزارش عامل‌های پلیس (خطر)
۵- تجربه مستقیم عامل مراجعه کننده (اعتماد)

در مقاله [۱] به‌طور مفصل درباره محاسبه و ترکیب این ۵ مورد بحث شده است ولی در آن برای محاسبه گزارش عامل‌های پلیس از روش پیشنهادی در [۸] استفاده شده است. در [۸] فقط احتمال وقوع حمله را در میزبان با استفاده از درخت AND/OR بدست می‌آورد و درباره نوع حمله صحبتی نمی‌شود. همچنین از نشانه هایی استفاده می‌شود که غالبا قابل محاسبه نیستند، درباره نحوه بدست آوردن نشانه ها در این مقاله صحبتی نشده است. ما در این مقاله روش جدیدی را برای نحوه فعالیت عامل های پلیس، برای گزارش درباره هر میزبان، معرفی می‌کنیم.

در این مقاله حمله ها براساس تاثیری که روی عامل به جای می-گذارند (نشانه های حمله) مورد بررسی قرار می‌گیرند و دسته بندی می‌شوند. عامل های پلیس به میزبان فرستاده می‌شوند، پس از بازگشت نشانه های آنها استخراج می‌شود و نشانه های بدست آمده به سیستم خبره پلیس وارد می‌شود و پس از آن حمله‌هایی که میزبان روی عامل انجام داده تشخیص داده می‌شود.

مزیت روش پیشنهادی ما نسبت به روش مقاله [۶] اینست که اولا نوع حمله در سیستم خبره تشخیص داده می‌شود، دوما از نشانه-های قابل محاسبه استفاده می‌شود و نحوه بدست آوردن آنها نیز ذکر شده است.

در ادامه مقاله، ابتدا در بخش ۲ به بررسی انواع حمله ها و دسته بندی آنها می‌پردازیم. در بخش ۳ نشانه های قابل محاسبه حمله ها را معرفی می کنیم. در بخش۴ روند کار سیستم پلیس برای جمع‌آوری نشانه ها شرح داده می‌شود، در بخش۵ الگوریتم هایی برای استخراج نشانه ها معرفی می‌شود، در بخش ۶ نحوه تصمیم-گیری سیستم پلیس براساس نشانه‌های موجود معرفی می‌شود و در بخش ۷ جمع بندی کلی روش پیشنهاد شده و کارهای آتی ذکر شده است.

۲- دسته‌بندی انواع حمله‌ها به عامل از نظر روش‌های موجود برای تشخیص حمله‌ها

با توجه به دسته بندی ارائه شده در [۴،۵] و روش های موجود برای تشخیص حمله ها، در این بخش حمله‌ها را از دید نشانه‌ها و روش‌های تشخیص موجود برای آنها بررسی می‌کنیم:

حمله علیه صحت عامل

تداخل در کد، حالت یا داده عامل درستی عامل را مورد تهدید قرار می‌دهد، که می‌تواند با اهداف بدخواهانه یا تصادفی باشد. این نوع تهدیدات در دو دسته قرار میگیرند.

تداخل در صحت [1]:

در این نوع حمله، میزبان عامل را به درستی اجرا نمی‌کند ولی با این وجود هیچ تغییری در اطلاعات عامل نمی‌دهد. انتقال نادرست عامل، اجرا نکردن کامل کد عامل، یا اجرای عامل به صورت اختیاری نمونه هایی از این نوع حمله هستند. این نوع حمله کد و داده عامل را تغییر نمی دهد، تأثیر این نوع حمله روی نتیجه اجرای عامل است. بنابراین تکنیک هایی قادر به تشخیص این نوع حمله هستند که اجرای صحیح عامل را با اجرای عامل در میزبان مورد بررسی مقایسه می کنند.

تغییر اطلاعات[2]:

این نوع حمله شامل موارد متعددی می شود که میزبان اجرا کننده از طریق دسترسی غیر مجاز، به عامل حمله می‌کند. مثل تغییر، خراب کردن یا حذف کد، داده، حالت یا جریان اجرای[3] عامل یا دخالت در ارتباطات عامل بامیزبان یا عامل های دیگر. چون تغییراتی که این نوع حمله ایجاد می کند ممکن است روی نتیجه اجرای عامل تأثیر بگذارد، بنابراین تکنیک هایی که اجرای

[1] Integrity attacks
[2] Information modification
[3] Control flow

صحیح عامل را با اجرای عامل در میزبان مورد بررسی مقایسه میکنند، قادر به تشخیص این نوع حمله هستند. همچنین می‌توان از داده های پوچ برای تشخیص تغییرات غیر مجاز استفاده کرد.

ممانعت از حق دسترسی

هنگامی که عامل مجاز وارد یک میزبان می شود باید اجازه دسترسی به منابع مورد نیازش را داشته باشد. اگر یک عامل مجاز از دسترسی به منابعی که اجازه دسترسی به آنها را دارد منع شود، حمله منع دسترسی اتفاق افتاده است. در واقع فعالیت‌های عمدی میزبان هستند که باعث مانع شدن عامل از رسیدن به هدفش میشوند و به سه دسته تقسیم می‌شود:

عدم ارئه خدمات[1]:

درخواست منابعی که عامل برای اتمام فعالیت هایش نیاز دارد رد می‌شود. یا اینکه میزبان بدخواه عامل را با اطلاعات نامربوط زیاد بمباران میکند تا عامل از کامل کردن هدفش باز بماند. چون هدف این نوع حمله عدم رسیدن عامل به هدفش (اجرای صحیح) است. بنابراین تکنیک هایی که اجرای صحیح عامل را با اجرای عامل درمیزبان مورد بررسی مقایسه میکنند قادر به تشخیص این نوع حمله هستند.

تاخیر در ارائه خدمات[2]

در این حمله میزبان عامل را برای ارائه خدمات منتظر میگذارد وفقط بعد از یک مدت زمان مشخص خدمات را در اختیار عامل قرار میدهد. این تأخیر می‌تواند تأثیر منفی روی اهداف عامل متحرک داشته باشد. چون هدف این نوع حمله تآخیر یا عدم رسیدن عامل به هدفش (اجرای صحیح) است، بنابراین تکنیک هایی که اجرای صحیح عامل را با اجرای عامل در میزبان مورد بررسی مقایسه میکنند قادر به تشخیص این نوع حمله هستند. قرار دادن یک زمان مشخص برای اجرای عامل هم می تواند روشی برای تشخیص این نوع حمله باشد.

امتناع از انتقال[3]

هنگامی این حمله اتفاق می افتد که میزبان از انتقال عامل به میزبان بعدی مشخص شده در سفرنامه امتناع کند. روش هایی که سفرنامه عامل را بررسی می کنند می توانند این حمله را تشخیص دهند.

حمله علیه محرمانگی عامل

این حمله وقتی اتفاق می افتد که قسمت با ارزشی از عامل مثل کد یا داده به طور غیرقانونی مورد دستیابی قرار گیرد. این حمله ها سه دسته اند:

شنود[4]

در این حمله میزبان با جاسوسی کردن روی عامل سعی می‌کند اطلاعاتی در باره عامل یا ارتباطات عامل با دیگر عامل ها جمع آوری کند تا از این اطلاعات به نفع خودش استفاده کند. این حمله با روش های ارائه شده قابل تشخیص نیست.

دزدیدن اطلاعات[5]

میزبان نه تنها اطلاعات عامل را شنود می کند بلکه آن اطلاعات را حذف نیز می کند. میزبان بدخواه ممکن است عامل را بدزدد تا از آن برای اهداف خودش استفاده کند یا اینکه عامل را نابود کند. راه تشخیص این حمله، مقایسه اجرای عامل در میزبان مشکوک با اجرای صحیح عامل در میزبان دیگری است.

جدول ۱: دسته بندی حمله ها بر اساس روش‌های تشخیص

روش های تشخیص	تهدیدات
روش هایی اجرای صحیح عامل را شبیه سازی میکنند سپس از مقایسه استفاده می کنند: تکرار سرور حالت های مرجع	تداخل در صحت تغییر اطلاعات عامل عدم خدمات دهی تأخیر در خدمات
روش هایی سفرنامه عامل را ثبت می کنند و اجرای عامل را رد یابی می کنند: تاریخچه مسیر ثبت دوطرفه سفرنامه کپسوله کردن نتایج میانی	تداخل در صحت تغییر اطلاعات عامل عدم خدمات دهی دزدیدن اطلاعات جعل هویت امتناع از انتقال
استفاده از اشیاء تشخیصی	تغییر اطلاعات عامل
محدودیت زمانی	عدم خدمات دهی تاخیر در خدمات دهی مهندسی معکوس شبیه سازی

مهندسی معکوس[6]

هنگامی که میزبان بدخواه عامل متحرک را بدست آورد، داده ها یا حالت آن را تحلیل میکند تا در آینده عامل را دستکاری کند یا حتی خود عامل را ایجاد کند. تنها راه تشخیص زمانی است که صرف انجام مهندسی معکوس عامل می‌شود.

[1] Denial of service
[2] delay of service
[3] Transmission refusal
[4] Eavesdropping
[5] Information theft
[6] Reverse engineering

حملات احراز اصالت

جعل هویت[1]

میزبان می‌تواند با تظاهر به هویت میزبانهای مطمئن, سعی در به دام انداختن عاملها کرده و با این حیله اطلاعات حساس آنها را استخراج نماید. خطر بستر بدخواه باعث دستکاری کد و حالت یک عامل می‌شود که در بخشهای قبلی توضیح داده شد. راه تشخیص این نوع حمله استفاده از امضای الکترونیکی[2] به همراه ثبت سفرنامه است.

شبیه سازی

هر عامل در حین سفرش شناسه‌ای را حمل می‌کند تا میزبانها با احراز اصالت او اجازه استفاده از سرویسهایشان را بدهند. حال اگر میزبانی یک کپی از عامل متحرک را ایجاد کند, احراز هویت عامل با مشکل مواجه می‌شود زیرا یکتا بودن شناسه عامل نقض می‌شود. تنها راه تشخیص زمانی است که صرف انجام کپی عامل می‌شود.

دسته بندی حمله ها بر اساس روش‌های تشخیص آنها در جدول ۱ ارائه شده است.

نتیجه ای که می‌توان از این جدول بدست آورد این است که با ترکیب همه روش های تشخیص می‌توان همه حمله ها به جز شنود را تشخیص داد. البته ما در سیستم پلیس روشی برای تشخیص آن پیشنهاد می‌کنیم. در بخش بعد به بررسی نشانه های قابل تشخیص حملات می‌پردازیم.

۳- نشانه های حملات

برخلاف روش‌های قبلی، ما در سیستم پلیس حمله ها را از روی نشانه آنها تشخیص می‌دهیم. عامل های پلیس بعد از سفر به میزبان به سیستم پلیس باز میگردند تا نشانه‌های قابل محاسبه آنها استخراج شود. سپس این نشانه‌ها وارد سیستم خبره می‌شود تا از طریق قانون های تعریف شده تشخیص داده شود چه حمله-هایی روی عامل اتفاق افتاده است.

همانطور که در بخش قبل مشخص شد، روش های موجود حمله شنود را تشخیص نمی دهند. برای تشخیص شنود می توان از طعمه استفاده کرد. به این صورت که قسمتی از عامل حاوی یک راز باشد و اجرای قسمت دوم کد عامل وابسته به این راز باشد، در واقع وظیفه قسمت دوم این است که تشخیص دهند آیا میزبان این راز را در اختیار دارد یا خیر. میزبان می‌تواند راز را از طریق دسترسی غیر مجاز یا از طریق شنود ارتباطات عامل بدست آورد. به طور مثال می‌توان در قسمت دوم کد عامل، چنین کدی قرار گیرد:

```
Agent code {
//first part of agent codes
If (secret)
{
    Request Honey data from home;
}
 //end part of agent codes
}
```

به طور مثال این honey data می تواند درخواست پول برای میزبان بدخواه باشد که با چنین درخواستی home تشخیص می دهد که شنود اتفاق افتاده است.

نشانه‌های قابل تشخیص توسط سیستم پلیس در جدول ۲ فهرست شده است. در بخش بعد الگوریتم های محاسبه هرکدام از این نشانه ها شرح داده شده است.

جدول ۲: نشانه‌های قابل استخراج از عامل‌های پلیس

Malicious Elapsed Time(MET)	عددی بین ۰-۱ که مشخص می کند مدت زمانی که عامل برای سفرش طی کرده است تا چه میزان بدخواهانه بوده است.
Wrong state (WST)	درصورتی ۱ است که وضعیت عامل با وضعیت صحیح آن متفاوت باشد.
Wrong data (WDT)	درصورتی ۱ است که داده‌های عامل با داده‌های صحیح آن متفاوت باشد
Wrong code (WCD)	درصورتی ۱ است که کد عامل با کد صحیح آن متفاوت باشد
Wrong itinerary (WIT)	درصورتی ۱ است که سفرنامه عامل با سفرنامه صحیح آن متفاوت باشد.
Wrong host ID (WHI)	درصورتی ۱ است که شناسه شناخته شده از میزبان با شناسه واقعی میزبان متفاوت باشد
Eavesdrop? (ED)	درصورتی ۱ است که شناسه شناخته شده از میزبان با شناسه واقعی میزبان متفاوت باشد

۴- روند کار سیستم پلیس برای جمع‌آوری نشانه‌ها

فرض می‌کنیم در مدل عامل‌های پلیس، میزبان‌ها خدمات مشابهی را ارائه میدهند. سیستم پلیس ابتدا گروهی از عامل های پلیس را از طریق یک سیستم بی نشان کننده[3] به شبکه می‌فرستد تا هر کدام یک میزبان را مورد بررسی قرار دهند و پس از اجرا به سیستم بازگردند. اطلاعات اولیه عامل ها در یک جدول به نام جدول عامل‌ها (AT) ذخیره می‌شود. علاوه بر این یک عامل برای مقایسه های آتی، در سیستم پلیس نگه داشته می شود. توضیح

[1] Masquerading

[2] Digital signature

[3] Anonymizer System

اینکه سیستم بی‌نشان کننده، سیستمی است که به کمک آن می‌توان یک عامل را به صورت بی نشان (یعنی بدون اینکه مالک واقعی عامل و یا مسیر طی شده توسط عامل از طریق میزبان قابل تشخیص باشد) به میزبان های مختلف ارسال نمود. نمونه ای از این سیستم ها در [۹] ارائه شده است.

هر عامل به یک میزبان فرستاده می شود و پس از اجرا از طریق سیستم بی نشان کننده به میزبان پلیس باز میگردد.(شکل ۱) .

```
Agents table: AT
Table of Symptoms of Attacks in Hosts: TSAH
Sending time: st
GoBack time: bt
1-  Create Dp and Sp
2-  Create Cp= {
        //first part of agent codes
        Request secret from host
        If secret is true then
        {
                Request Honey data from home
        }
        //end part of agent codes
    }
3-  For i=1 to n
        //
        3.1 Create agent_i with (id_i, Cp , Dp, Sp)
        3.2 set St to be time of now
        3.3 set bt to be 0
        3.4 send agent_i to host_i (h_i) through anonymizer system
        3.5 set ED to be 0;
        3.6 insert t(id_i , PoH_i , PoN, h_i ,st,bt,ED) into AT
        3.7 start pst_i timer//police system timer
3- Create agent_{n+1} with (id_{N+1} , Cp , Dp, Sp) // sedentary agent
4-  End
```

شکل ۱: الگوریتم سیستم پلیس برای ایجاد و ارسال عامل‌ها

جمع آوری اطلاعات لازم توسط عامل های پلیس انجام می شود و سپس پردازش این اطلاعات در سیستم پلیس انجام می پذیرد. چون عامل ها فقط یک کد عادی را در میزبان اجرا میکنند، هدف عامل از میزبان مخفی می ماند

همانگونه که در شکل ۲ نشان داده شده است، سیستم پلیس بعد از فرستادن عامل‌های پلیس به میزبان‌ها ۴ فعالیت را باید به صورت موازی انجام دهد:

- اجرای عامل مستقر در سیستم ($agent_{n+1}$) و محاسبه زمان اجرای آن توسط زمانسنج et (part1)
- در صورت دریافت نشانه‌ای مبتنی بر وقوع شنود، مقدار دهی به ED در سطر مربوط به عامل (part2)
- درصورت بازگشت عامل‌ها، محاسبه زمان رفت و برگشت برای هر عامل (part3)
- در صورت تمام شدن مهلت انتظار برای بازگشت عامل-ها رفتن به مرحله استخراج نشانه ها (part4) .

```
Execute part1, part2, part3 and part4 concurrently

Part1: //execute sedentary agent
    1. Start et timer
    2. Execute agent_{n+1}
    3. End et timer //calculating executing time
// agent after execute: agen_{n+1} (id_{n+1},Cs, Ds, Ss )

Part2: //set ED for each host
    1. If received any sign from agent_i then set ED
       in t_i in AT to be 1;

Part3: //set bt of each agent
    1. If each agent back to police system then
       set bt of t_i in AT equal to pst_i

Part4: //if deadline is met go for extract step
    1. For i=1 to n
    1.1. if pst_i is equal to Mt then
        1.1.1.End pst_i
        1.1.2. Execute <extract symptoms function>
```

شکل ۲: فعالیت‌های موازی سیستم پلیس در زمان اجرای عامل‌ها در میزبان‌ها

سیستم مدت زمان مشخصی (MT) را برای بازگشت همه عامل ها صبر می‌کند و پس از اتمام سفر عامل‌ها و استخراج نشانه‌ها از عامل‌ها، به ازاء هر عامل یک سطر در جدول نشانه های میزبان‌ها (TSAH) ذخیره می‌کند ، مانند آنچه درجدول۳ به عنوان نمونه نشان داده شده است.

برای محاسبه هر نشانه، سیستم با استفاده از اطلاعات بدست آمده از عامل فرستاده شده و مقایسه آن با عامل نگهداری شده یک احتمال بین ۰ و ۱ را بدست می آورد.

مقادیر بدست آمده در مرحله استخراج نشانه‌ها به سیستم خبره فرستاده می‌شود تا حمله‌های به وقوع پیوسته در هر میزبان تشخیص داده شود. در بخش بعد نحوه استخراج نشانه‌ها شرح داده شده است.

۵- استخراج نشانه‌ها

در سیستم پلیس برای محاسبه نشانه هر حمله از مقایسه عامل با اجرای صحیح آن استفاده می‌کنیم. همانطور که ذکر شد، در ابتدا n عامل پلیس از طریق سیستم بی‌نشان کننده به n میزبان در محیط فرستاده می شود و یک عامل ($agent_{n+1}$) در خود سیستم پلیس مستقر می‌ماند و در آنجا اجرا می‌شود. سپس با مقایسه نتیجه عامل با نتیجه حاصل از اجرای عامل مستقر در

سیستم می‌توان نشانه‌های حمله را استخراج کرد. با مقدار دهی اولیه به عامل‌های پلیس و اجرای عامل مستقر، موارد زیر در اختیار سیستم پلیس قرار می‌گیرد:

Ss: وضعیت عامل مستقر بعد از اجرا
Ds: داده های عامل مستقر بعد از اجرا
Cs: کد عامل مستقر بعد از اجرا
t: مدت زمان اجرای عامل مستقر
st: زمان فرستادن عامل ها به میزبان
PoN: خصوصیات ترافیک شبکه
PoH: خصوصیات میزبان
hi: شناسه میزبان

اطلاعاتی که بعد از بازگشت عامل های پلیس، برای هر میزبان در اختیار داریم:

Sh: وضعیت عامل پس از بازگشت
Dh: داده های عامل پس از بازگشت
Ch: کد عامل پس از بازگشت
bt: زمان رفت و برگشت
HiI: شناسه ای که میزبان از خود به عامل معرفی کرده است.

با استفاده از این اطلاعات همانطور که در الگوریتم شکل۳ نشان داده شده است می‌توانیم نشانه های MST و MDT و ED و MCD و MET وWHI را محاسبه کنیم:

اگر تفاوتی بین هریک از کد یا داده و یا وضعیت عامل پلیس به ترتیب با کد یا داده و یا وضعیت مورد انتظار (عامل مستقر) مشاهده شود نشانه مربوط به آنها در سیستم پلیس یک می‌شود (MST یا MCD یا MDT) . اگر شناسه ای که میزبان به عامل معرفی کرده با شناسه واقعی عامل متفاوت باشد نشانه WHI برابر ۱ می‌شود.

سیستم پلیس برای محاسبه MET از تابعی به نام CalculateElapsedTimeInHost استفاده می‌کند. این تابع زمانی که عامل در میزبان سپری کرده است را محاسبه می کنید. ورودی‌های این تابع شامل خصوصیات ترافیک شبکه، خصوصیات میزبان مورد بررسی و زمان فرستادن عامل به میزبان و زمان سپری شده برای رفت و برگشت عامل پلیس می شود:

ET=CalculateElapsedTimeInHost(PoH, PoN, st, bt)

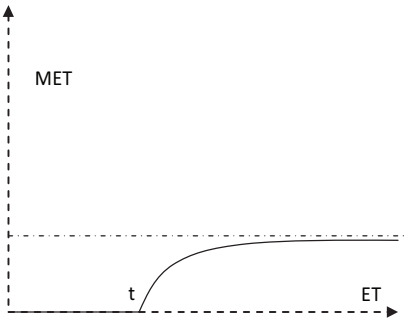
اگر ET از t (زمان اجرای عامل مستقر) بیشتر باشد احتمال اینکه این زمان صرف فعالیت‌های بدخواه شده باشد، زیاد است. اگر MT حداکثر زمانی باشد که سیستم منتظر بازگشت عامل می ماند:

$$MET = \begin{cases} 0 & if \quad TE \le t \\ 1 - e^{-(TE-t)} & if \quad TE > t \end{cases}$$

```
Extract symptoms function:
Inputs: agent_{n+1} = (id_{n+1} , Cs, Ds, Ss)
For each t_i(id_i , PoH_i , PoN, h_i ,st,bt,ED) in AT
        et_i = CalculateElapsedTime (PoN, PoH, st,bt)
        WCD, WDT, WST, WHI, MET, WIT set to be 0
        if bt != 0 then
                Find agent with id_i
                Ch is code of agent
                Dh is Data of agent
                Sh is state of agent
                HiI is decrypt of id of h_i with public key of h_i
                if Ch != Cs  then  WCD set to be 1
                if Dh != Ds  then  WDT set to be 1
                if Sh != Ss  then  WST set to be 1
                if h_i!=HiI  then  WHI set to be 1
                if et_i<= t then MET set to be 0
                else  MET = 1 - e^{-(et_i - t)}
                if ED = 1 then set ED to be 1
                insert (WCD, WDT, WST, WHI, MET) into SAHT
        else MET set to be 1  //agent didn't come back to system
```

شکل ۳: الگوریتم استخراج نشانه‌های WCD ، WDT ، WST ، WHI و MET

نمودار تغییرات MET بر حسب ET در شکل۴ نشان داده شده است. هرچه TE از t بزرگتر باشد در نتیجه MET بزرگتر می‌شود و در صورتی که از t خیلی بزرگتر باشد به سمت ۱ میل می‌کند.



شکل ۴: نمودار تغییرات MET

برای محاسبه WIT، عامل پلیس با سفرنامه ای مشخص (شامل n میزبان) به میزبان i فرستاده می شود در عین حال عامل پلیس دیگری با همین سفرنامه ولی با حذف میزبان i به محیط فرستاده می‌شود:

itinerary include hosti (IIH):

سفرنامه ثبت شده توسط عاملی که از میزبان i گذر می‌کند

itinerary exclude hosti (IEH):

سفرنامه ثبت شده توسط عاملی که از میزبان i گذر نمی‌کند.

هر میزبان هنگام عبور عامل از آن، شناسه خود را که با کلید خصوصی میزبان رمز شده است را در عامل ثبت می‌کند بنابراین عامل از میزبان هایی که از آنها عبور کرده است اطلاع دارد. از طرفی

//For calculating WIT:
1. Select n host to calculate WIT for them
2. For i=1 to n
   2.1. Set itinerary of **agent1**$_i$= all selected host except **host**$_i$
   2.2. Set itinerary of **agent2**$_i$= all selected hosts
3. Sent each agent to environment

//after backing agent to police system:
//extracting WIT
1. **IEH** is Visited hosts of **agent1**$_i$
2. **IIH** is Visited hosts of **agent2**$_i$
3. If **IIH** is wrong AND **IEH** isn't wrong then
   3.1. Set **WIT** to be 1
4. If **IIH** isn't wrong AND **IEH** isn't wrong then
   4.1. Set **WIT** to be 0
5. If **IIH** isn't wrong AND **IEH** is wrong then nothing
   //This is not possible because **IEH** is subset of **IIH**
6. If **IIH** is wrong AND **IEH** is wrong then
   6.1. If for other hosts in itinerary **IEH** aren't wrong then
      6.1.1. Set WIT to be 0
   6.2. Else //for all host in itinerary **IEH** is wrong
      6.2.1. We only can understand that 2 or more hosts in itinerary are malicious

شکل ۵: الگوریتم محاسبه WIT توسط سیستم پلیس

چون هر میزبان با کلید خصوصی خود شناسه را رمز می‌کند، نمی‌تواند در آینده منکر آن شود. در حالتی که عامل از میزبان i عبور کند مجموعه میزبان هایی که از آن عبور کرده IIH نامیده شده است و در حالتی که از میزبان i عبور نکند IEH نامیده می‌شود. الگوریتم استخراج WIT در شکل ۵ نشان داده شده است.

در این الگوریتم برای هر میزبان، مقدار دهی به WIT با مقایسه درست بودن IIH و IEH انجام می‌پذیرد. در صورتی که IIH درست نباشد، یعنی یکی از میزبان‌های درون سفرنامه، مسیر حرکت عامل را تغییر داده است در این حالت اگر IEH درست باشد به این معنی است که بقیه میزبان‌ها سفرنامه را تغییر نمی‌دهند بنابر این WIT برای این میزبان با ۱ مقداردهی می‌شود. اگر هر دو IEH و IIH درست بود WIT با ۰ مقدار دهی می‌شود. اگر هردو نادرست بود به این معناست که یک یا چند میزبان موجود در سفرنامه بدخواه هستند. حال اگر IEH میزبان‌های دیگر درست بود چون این میزبان هم جزو IEH بقیه میزبان هاست بنابراین WIT آن را با ۰ مقدار دهی می مکنیم در بقیه موارد با این الگوریتم نمی‌توان نظری داد.

۶- نحوه تصمیم‌گیری سیستم پلیس بر اساس نشانه‌ها

همانطور که در بخش‌های قبل مشاهده کردیم سیستم پلیس تعدادی عامل پلیس را برای بررسی وضعیت میزبان‌ها به محیط میفرستد، سیستم مدت زمان مشخصی (MT) را برای بازگشت همه عامل ها صبر می‌کند و پس از اتمام سفر عامل‌ها و استخراج نشانه‌ها از عامل‌ها، به ازاء هر عامل یک سطر در جدول نشانه های میزبان‌ها (TSAH) ذخیره می کند ، مانند آنچه در جدول۳ به عنوان نمونه نشان داده شده است.

برای محاسبه هر نشانه سیستم با استفاده از اطلاعات بدست آمده از عامل فرستاده شده و مقایسه آن با کپی عامل نگهداری شده یک احتمال بین ۰ و ۱ را بدست می آورد.

مقادیر بدست آمده در مرحله استخراج نشانه‌ها به سیستم خبره فرستاده می‌شود تا حمله‌های به وقوع پیوسته در هر میزبان تشخیص داده شود. بر اساس تعریف حمله ها و نشانه‌های موجود قوانین سیستم خبره تعریف می‌شوند. به طور مثال حمله تغییر اطلاعات زمانی صورت میگیرد که وضعیت عامل یا سفرنامه عامل یا کد عامل یا داده‌های عامل به طور ناصحیح تغییر کرده باشد و یا اگر وضعیت عامل تغییر کرده باشد و زمانی هم که عامل در میزبان سپری کرده باشد بدخواهانه باشد این حمله منع خدمات‌دهی است. ممکن است همزمان چند حمله صورت بگیرد و طرف اول چند قانون همگی اتفاق بیفتد که در این صورت سیستم پلیس همه حمله‌ها را تشخیص می‌دهد. بنا بر این قوانین سیستم خبره را می‌توان به صورت زیر مشخص کرد:

$$WST \vee WDT \vee WCD \vee WIT \rightarrow Information\ modification\ attack$$

$$WST \wedge \sim WDT \wedge \sim WCD \rightarrow Integrity\ attack$$

$$WST \wedge MET \rightarrow Denial\ of\ service\ attack$$

$$MET \rightarrow Delay\ of\ service\ attack \vee Cloning\ attack$$

$$WIT \rightarrow Transmision\ refusal\ attack$$

$$ED \rightarrow eavesdrop\ attack$$

$$ED \wedge (WDT \vee WCD \vee WST) \rightarrow Theft\ information\ attack$$

$$MET \rightarrow reverse\ engineering$$

$$WHI \rightarrow Masqurading\ attack$$

جدول ۳: یک سطر نمونه از جدول نشانه‌ها در میزبان‌ها

Agent#	Host1								Host2	...
	time	Malicious Elapsed Time(MET)	Wrong state(WST)	Wrong data(WDT)	Wrong code(WCD)	Wrong itinerary(WIT)	Wrong host ID(WHI)	Eavesdrop? (ED)	…	...
Agenti	1:1:30	0.7	1	0	0	1	0	1		

۷- نتیجه گیری

در این مقاله برای اولین بار سعی شده است بدخواه بودن میزبان ها در یک سیستم چند عاملی را از روی نشانه‌های حمله قابل محاسبه آنها تشخیص دهیم. نحوه بدست آوردن نشانه‌ها نیز معرفی شده است. این نشانه‌ها به سیستم خبره فرستاده می‌شود تا حمله‌های احتمالی تشخیص داده شوند. مزیت این روش سیستم پلیس نسبت به دیگر روش ها این است که به وضعیت عامل قبل و بعد اجرا در میزبان دسترسی داریم و همچنین به سابقه طولانی مدت میزبان ها نیز دسترسی داریم.

در آینده نحوه دخالت دادن روش جدید تشخیص حمله در محاسبه دقیقتر اعتماد و پیاده سازی سیستم کامل پلیس و تحلیل دقیق آن کار می‌شود. همچنین برای فعالیت‌های آتی می‌توان روی نشانه های قابل محاسبه بیشتری مثل تکراری بودن شناسه عامل تحقیق انجام شود و برای دقیق تر شدن قوانین به قانون ها اضافه شود.

مراجع

[۱] نسیبه محمودی، بهروز ترک‌لادانی :" مدلی برای سیستم پلیس در محیط عامل‌های متحرک "، مجموعه مقالات پنجمین کنفرانس بین المللی انجمن رمز ایران، دانشگاه صنعتی مالک اشتر تهران، ۱۳۸۷.

[2] Kamalrulnizam Abu Bakar and B. S. Doherty" Evaluation of the Recorded State Mechanism for Protecting Agent Integrity Against Malicious Hosts ", International Journal of Computers, Communications & Control Vol. III (2008), No. 1, pp. 60-68

[3] Abdelmorhit El Rhazi, Samuel Pierre and Hanifa Boucheneb " A Secure Protocol Based on a Sedentary Agent for Mobile Agent Environments", Journal of Computer Science 3 (1): 35-42, 2007 ISSN 1549-3636 © 2007 Science Publications

[4] Nitschke. L and Paprzycki. M and Ren.M "*Mobile Agent Security*", Research Note of KBN grant 0 T00A 003 23, 2005.

[5] ELMARIE BIERMAN and ELSABE CLOETE:" *Classification of Malicious Host Threats in Mobile Agent Computing* " © 2002 SAICSIT

[6] J. Todd McDonald, Alec Yasinsac, Willard C. Thompson: "*Mobile Agent Data Integrity Using Multi-agent Architecture*". ISCA PDCS 2004: 536-542

[7] Merwe, J. and Solms, S.H,"*Electronic Commerce with Secure Intelligent Trade Agents*". Proceedings of ICICS'97, LNCS Vol. 1334, Springer-Verlag (1997) 452−462

[8] Magdy Saeb, Meer Hamza, Ashraf Soliman," *Protecting Mobile Agents against Malicious Host Attacks Using Threat Diagnostic AND/OR Tree*" , Smart Objects Conference, SOC2003, Grenoble, France, May15-17th, 2003

[9] Raji, F.; Ladani, B.T.; Brenjkoub, M."Anonymous Agent with Anonymous Itinerary" Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on Volume , Issue , 15-19 Dec. 2007 Page(s):534 – 537