



# تشخیص میزبان‌های بدخواه در سیستم پلیس عامل‌ها بر اساس استخراج نشانه‌های 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 قرار گرفته 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 عامل‌های متحرک پیشنهاد شده است که در آن تشخیص بدخواه بودن میزبان‌ها بر عهده پلیس عامل‌ها است. در این مقاله علاوه بر اصلاح نحوه کار عامل‌های پلیس، روش جدیدی برای تصمیم‌گیری راجع به میزبان بدخواه بودن میزبان‌ها با استفاده از یک سیستم خبره مبتنی بر قوانین غیر قطعی ارائه نموده ایم. برای انجام این کار با اعزام عامل‌های بی‌نشان پلیس به میزبان‌های مشکوک، نشانه‌های حمله براساس تأثیری که روی عامل به جای می‌گذارند استخراج و دسته‌بندی شده، سپس به کمک سیستم خبره پلیس میزبان بدخواه بودن میزبان‌ها تعیین می‌شود.

## واژه‌های کلیدی

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

## ۱- مقدمه

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 و ۵]: اشیاء تشخیصی<sup>۵</sup>، تاریخچه مسیر<sup>۶</sup>، ثبت دو جانبه سفرنامه<sup>۷</sup>، تکرار میزبان<sup>۸</sup>، حالت‌های مرجع<sup>۹</sup>.

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 مکانیسم‌های جلوگیری از مداخله هنوز ادامه دارد. تکنیک‌های تشخیص نیز در واقع بعد از وقوع حمله است و نیاز عامل را در همان زمان برآورده نمی‌کند.

روش‌های زیادی در طی سال‌های اخیر برای محافظت از عامل‌های متحرک پیشنهاد شده است [۵-۱] که برخی روی تشخیص حمله و برخی روی جلوگیری از حمله به عامل تأکید دارند. هدف از مکانیسم‌های جلوگیری از مداخله این است که مداخله در عامل توسط میزبان یا بسیار مشکل باشد و یا برای او هزینه بالایی را در پی داشته‌باشد، این مکانیسم‌ها زمینه تحقیقاتی فعلی اکثر محققان امروزی است. تعدادی از مکانیسم‌هایی که به این منظور ارائه شده عبارتند از [۴، ۵]: سخت‌افزار مقاوم در برابر حمله<sup>۱</sup>، تماس با خانه<sup>۲</sup>، استفاده از سیستم چندعاملی<sup>۳</sup> و به‌هم‌ریختن کد<sup>۴</sup>.

<sup>5</sup> Detection object

<sup>6</sup> Path history

<sup>7</sup> Mutual itinerary recording

<sup>8</sup> Replication server

<sup>9</sup> References state

<sup>1</sup> Tamper Resistant Hardware

<sup>2</sup> Phone Home

<sup>3</sup> Multi Agent System

<sup>4</sup> Code Obfuscation

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

در ادامه مقاله، ابتدا در بخش ۲ به بررسی انواع حمله‌ها و دسته‌بندی آنها می‌پردازیم. در بخش ۳ نشانه‌های قابل محاسبه حمله‌ها را معرفی می‌کنیم. در بخش ۴ روند کار سیستم پلیس برای جمع‌آوری نشانه‌ها شرح داده می‌شود، در بخش ۵ الگوریتم‌هایی برای استخراج نشانه‌ها معرفی می‌شود، در بخش ۶ نحوه تصمیم‌گیری سیستم پلیس براساس نشانه‌های موجود معرفی می‌شود و در بخش ۷ جمع‌بندی 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 دسته‌بندی ارائه شده در [۴،۵] و روش‌های موجود برای تشخیص حمله‌ها، 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<sup>۱</sup>:

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 نمونه‌هایی از این نوع حمله هستند. این نوع حمله کد و داده عامل را تغییر نمی‌دهد، تأثیر این نوع حمله روی نتیجه اجرای عامل است. بنابراین تکنیک‌هایی قادر به تشخیص این نوع حمله هستند که اجرای صحیح عامل را با اجرای عامل در میزبان مورد بررسی مقایسه می‌کنند.

#### تغییر اطلاعات<sup>۲</sup>:

این نوع حمله شامل موارد متعددی می‌شود 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<sup>۳</sup> عامل یا دخالت در ارتباطات عامل بامیزبان یا عامل‌های دیگر. چون تغییراتی که این نوع حمله ایجاد می‌کند ممکن است روی نتیجه اجرای عامل تأثیر بگذارد، بنابراین تکنیک‌هایی که اجرای

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

ما در کار قبلی خود [۱]، مدلی به نام سیستم پلیس عامل‌ها 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 عامل‌ها با مجهز بودن به روش‌های مختلف تشخیص میزبان‌های 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 کد‌های سنگین، فقط با استعلام از سیستم عامل‌های پلیس به نحو مناسب می‌تواند 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 می‌کند:

- ۱- گزارش میزبان همسایه درباره میزبان مورد بررسی (شهرت).
  - ۲- عامل‌هایی 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 نشانه‌هایی استفاده می‌شود که غالباً قابل محاسبه نیستند، درباره نحوه بدست آوردن نشانه‌ها در این مقاله صحبتی نشده است. ما در این مقاله روش جدیدی را برای نحوه فعالیت عامل‌های پلیس، برای گزارش درباره هر میزبان، معرفی می‌کنیم.
- در این مقاله حمله‌ها براساس تأثیری که روی عامل به جای می‌گذارد (نشانه‌های حمله) مورد بررسی قرار می‌گیرند و دسته‌بندی می‌شوند. عامل‌های پلیس به میزبان فرستاده می‌شوند، پس از بازگشت نشانه‌های آنها استخراج می‌شود و نشانه‌های بدست آمده به سیستم خبره پلیس وارد می‌شود و پس از آن حمله‌هایی که میزبان روی عامل انجام داده تشخیص داده می‌شود.

<sup>1</sup> Integrity attacks

<sup>2</sup> Information modification

<sup>3</sup> Control flow

#### شنود<sup>۴</sup>

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<sup>۵</sup>

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 روش های تشخیص

تهدیدات	روش های تشخیص
تداخل در صحت تغییر اطلاعات عامل عدم خدمات دهی تأخیر در خدمات	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 دزدیدن اطلاعات جعل هویت امتناع از انتقال	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
تغییر اطلاعات عامل	استفاده از اشیاء تشخیصی
عدم خدمات دهی تأخیر در خدمات دهی مهندسی معکوس شبیه سازی	محدودیت زمانی

#### مهندسی معکوس<sup>۶</sup>

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.

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 ارائه خدمات<sup>۱</sup>:

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 در میزبان مورد بررسی مقایسه میکنند قادر به تشخیص این نوع حمله هستند.

#### تأخیر در ارائه خدمات<sup>۲</sup>

در این حمله میزبان عامل را برای ارائه خدمات منتظر میگذارد فقط 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 تأخیر 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 می‌تواند روشی برای تشخیص این نوع حمله باشد.

#### امتناع از انتقال<sup>۳</sup>

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 می‌توانند 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:

<sup>4</sup> Eavesdropping

<sup>5</sup> Information theft

<sup>6</sup> Reverse engineering

<sup>1</sup> Denial of service

<sup>2</sup> delay of service

<sup>3</sup> Transmission refusal

```
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 می‌تواند درخواست پول برای میزبان بدخواه باشد که با چنین درخواستی home تشخیص می‌دهد که شنود اتفاق افتاده است.

نشانه‌های قابل تشخیص توسط سیستم پلیس در جدول ۲ فهرست شده است. در بخش بعد الگوریتم‌های محاسبه هر کدام از این نشانه‌ها شرح داده شده است.

جدول ۲: نشانه‌های قابل استخراج از عامل‌های پلیس

Malicious Elapsed Time(MET)	عددی بین ۱-۰۰ که مشخص می‌کند مدت زمانی که عامل برای سفرش طی کرده است تا چه میزان بدخواهانه بوده است.
Wrong state (WST)	در صورتی ۱ است که وضعیت عامل با وضعیت صحیح آن متفاوت باشد.
Wrong data (WDT)	در صورتی ۱ است که داده‌های عامل با داده‌های صحیح آن متفاوت باشد
Wrong code (WCD)	در صورتی ۱ است که کد عامل با کد صحیح آن متفاوت باشد
Wrong itinerary (WIT)	در صورتی ۱ است که سفرنامه عامل با سفرنامه صحیح آن متفاوت باشد.
Wrong host ID (WHI)	در صورتی ۱ است که شناسه شناخته شده از میزبان با شناسه واقعی میزبان متفاوت باشد
Eavesdrop? (ED)	در صورتی 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 عامل‌های پلیس را از طریق یک سیستم بی‌نشان کننده<sup>۲</sup> به شبکه می‌فرستد تا هر کدام یک میزبان را مورد بررسی قرار دهند و پس از اجرا به سیستم بازگردند. اطلاعات اولیه عامل‌ها در یک جدول به نام جدول عامل‌ها (AT) ذخیره می‌شود. علاوه بر این یک عامل برای مقایسه‌های آتی، در سیستم پلیس نگه داشته می‌شود. توضیح

### حملات احراز اصالت

#### جعل هویت<sup>۱</sup>

میزبان می‌تواند با تظاهر به هویت میزبانهای مطمئن، سعی در به دام انداختن عاملها کرده و با این حيله 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<sup>۲</sup> به همراه ثبت سفرنامه است.

#### شبیه‌سازی

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 حمله‌ها بر اساس روش‌های تشخیص آنها در جدول ۱ ارائه شده است.

نتیجه‌ای که می‌توان از این جدول بدست آورد این است که با ترکیب همه روش‌های تشخیص می‌توان همه حمله‌ها به جز شنود را تشخیص داد. البته ما در سیستم پلیس روشی برای تشخیص آن پیشنهاد می‌کنیم. در بخش بعد به بررسی نشانه‌های قابل تشخیص حملات می‌پردازیم.

### ۳- نشانه‌های حملات

برخلاف روش‌های قبلی، ما در سیستم پلیس حمله‌ها را از روی نشانه آنها تشخیص می‌دهیم. عامل‌های 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 قانون‌های تعریف شده تشخیص داده شود چه حمله‌هایی روی عامل اتفاق افتاده است.

همانطور که در بخش قبل مشخص شد، روش‌های موجود حمله‌ها را تشخیص نمی‌دهند. برای تشخیص شنود می‌توان 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:

<sup>1</sup> Masquerading

<sup>2</sup> Digital signature

<sup>3</sup> Anonymizer System

Execute <i>part1</i> , <i>part2</i> , <i>part3</i> and <i>part4</i> concurrently
<p><b>Part1:</b> //execute sedentary agent</p> <ol style="list-style-type: none"> <li>1. Start <b>et</b> timer</li> <li>2. Execute <b>agent<sub>n+1</sub></b></li> <li>3. End <b>et</b> timer //calculating executing time</li> </ol> <p>// agent after execute: <math>agen_{n+1}(id_{n+1}, Cs, Ds, Ss)</math></p>
<p><b>Part2:</b> //set ED for each host</p> <ol style="list-style-type: none"> <li>1. If received any sign from <b>agent<sub>i</sub></b> then set <b>ED</b> in <b>t<sub>i</sub></b> in <b>AT</b> to be 1;</li> </ol>
<p><b>Part3:</b> //set bt of each agent</p> <ol style="list-style-type: none"> <li>1. If each agent back to police system then set <b>bt</b> of <b>t<sub>i</sub></b> in <b>AT</b> equal to <b>pst<sub>i</sub></b></li> </ol>
<p><b>Part4:</b> //if deadline is met go for extract step</p> <ol style="list-style-type: none"> <li>1. For <math>i=1</math> to <math>n</math> <ol style="list-style-type: none"> <li>1.1. if <b>pst<sub>i</sub></b> is equal to <b>Mt</b> then           <ol style="list-style-type: none"> <li>1.1.1. End <b>pst<sub>i</sub></b></li> <li>1.1.2. Execute &lt;extract symptoms function&gt;</li> </ol> </li> </ol> </li> </ol>

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

سیستم مدت زمان مشخصی (MT) را برای بازگشت همه عامل‌ها صبر می‌کند و پس از اتمام سفر عامل‌ها و استخراج نشانه‌ها از عامل‌ها، به ازاء هر عامل یک سطر در جدول نشانه‌های میزبان‌ها (TSAH) ذخیره می‌کند، مانند آنچه در جدول ۳ 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 می‌آورد.

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 می‌شود 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

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

هر عامل به یک میزبان فرستاده می‌شود و پس از اجرا از طریق سیستم بی‌نشان کننده به میزبان پلیس باز میگردد. (شکل ۱).

<p>Agents table: <b>AT</b>          Table of Symptoms of Attacks in Hosts: <b>TSAH</b>          Sending time: <b>st</b>          GoBack time: <b>bt</b></p> <ol style="list-style-type: none"> <li>1- Create <b>Dp</b> and <b>Dp</b></li> <li>2- Create <b>Cp</b>= {                //first part of agent codes                Request <b>secret</b> from host                If <b>secret</b> is true then                {                    Request Honey data from home                }                //end part of agent codes              }</li> <li>3- For <math>i=1</math> to <math>n</math>                //                3.1 Create <b>agent<sub>i</sub></b> with (<math>id_i, Cp, Dp, Sp</math>)                3.2 set <b>St</b> to be time of now                3.3 set <b>bt</b> to be 0                3.4 send <b>agent<sub>i</sub></b> to <b>host<sub>i</sub></b> (<math>h_i</math>) through anonymizer system                3.5 set <b>ED</b> to be 0;                3.6 insert <math>t(id_i, PoH_i, PoN, h_i, st, bt, ED)</math> into <b>AT</b>                3.7 start <b>pst<sub>i</sub></b> timer //police system timer</li> <li>3- Create <b>agent<sub>n+1</sub></b> with (<math>id_{n+1}, Cp, Dp, Sp</math>) // sedentary agent</li> <li>4- End</li> </ol>
---

شکل ۱: الگوریتم سیستم پلیس برای ایجاد و ارسال عامل‌ها

جمع آوری اطلاعات لازم توسط عامل‌های پلیس انجام می‌شود و سپس پردازش این اطلاعات در سیستم پلیس انجام می‌پذیرد. چون عامل‌ها فقط یک کد عادی را در میزبان اجرا میکنند، هدف عامل از میزبان مخفی می‌ماند 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 (**part1**) **et**
- در صورت دریافت نشانه‌ای مبتنی بر وقوع شنود، مقدار دهی به **ED** در سطر مربوط به عامل (**part2**)
- در صورت بازگشت عامل‌ها، محاسبه زمان رفت و برگشت برای هر عامل (**part3**)
- در صورت تمام شدن مهلت انتظار برای بازگشت عامل‌ها رفتن به مرحله استخراج نشانه‌ها (**part4**).

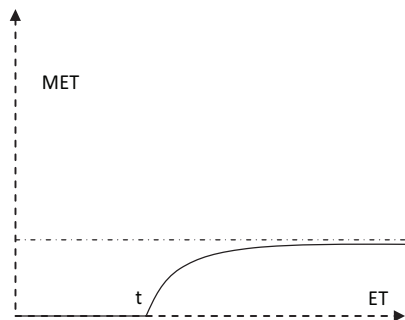


```

Extract symptoms function:
Inputs: agentn+1 = (idn+1, Cs, Ds, Ss)
For each ti(idi, PoHi, PoNi, hi, st, bt, ED) in AT
    eti = CalculateElapsedTime (PoN, PoH, st, bt)
    WCD, WDT, WST, WHI, MET, WIT set to be 0
    if bt != 0 then
        Find agent with idi
        Ch is code of agent
        Dh is Data of agent
        Sh is state of agent
        Hii is decrypt of id of hi with public key of hi
        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 hi != Hii then WHI set to be 1
        if eti <= t then MET set to be 0
        else MET = 1 - e-(eti-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 شکل ۴ 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 می‌شود 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 میزبان‌هایی که از آنها عبور کرده است اطلاع دارد. از طرفی

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: داده‌های عامل مستقر بعد از اجرا

Cs: کد عامل مستقر بعد از اجرا

t: مدت زمان اجرای عامل مستقر

st: زمان فرستادن عامل‌ها به میزبان

PoN: خصوصیات ترافیک شبکه

PoH: خصوصیات میزبان

hi: شناسه میزبان

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

Sh: وضعیت عامل پس از بازگشت

Dh: داده‌های 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 شکل ۳ نشان داده شده است می‌توانیم نشانه‌های MST و MDT و ED و MCD و MET و WHI 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 می‌کند. 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 می‌شود:

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 می‌ماند:

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



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		

- 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, May15-17<sup>th</sup>, 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

## ۷- نتیجه گیری

در این مقاله برای اولین بار سعی شده است بدخواه بودن میزبان‌ها 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 روش‌ها این است که به وضعیت عامل قبل و بعد اجرا در میزبان دسترسی داریم و همچنین به سابقه طولانی مدت میزبان‌ها 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 نشانه‌های قابل محاسبه بیشتری مثل تکراری بودن شناسه عامل تحقیق انجام شود و برای دقیق تر شدن قوانین به قانون‌ها اضافه شود.

## مراجع

- [۱] نسیمه محمودی، بهروز ترک‌لادانی: "مدلی برای سیستم پلیس در محیط عامل‌های متحرک"، مجموعه مقالات پنجمین کنفرانس بین‌المللی انجمن رمز ایران، دانشگاه صنعتی مالک اشتر تهران، ۱۳۸۷.
- [2] Kamalrulnizam Abu Bakar and B. S. Doherty" Evaluation of the Recorded State Mechanism for