هفتمین کنفرانس بین المللی انجمن رمز ایران
۲۴ و ۲۵ شهریورماه ۱۳۸۹، دانشگاه صنعتی خواجه نصیرالدین طوسی
!
انجمن رمزایران
دانشگاه صنعتی خواجه نصیرالدین طوسی

استخراج آنتولوژی حملات به روش داده‌کاوی جهت استفاده در سیستم‌های تشخیص نفوذ

مهدی هاشمی شهرکی ، محسن کاهانی

گروه مهندسی کامپیوتر، دانشگاه فردوسی مشهد
me_ha6@stu-mail.um.ac.ir, kahani@um.ac.ir

چکیده

یکی از کاربردهای جدید وب معنایی در سیستم‌هایی است که با استفاده از یک سری قوانین سعی در دسته‌بندی خروجی دارند. در مورد سیستم‌های تشخیص نفوذ نیز پژوهش‌هایی صورت گرفته است ولی در اغلب کارها از یک تاکسونومی جهت نمایش ویژگی حملات استفاده شده است. بکارگیری تاکسونومی مشکلات و محدودیت‌های فراوانی در سیستم تشخیص نفوذ ایجاد می‌کند که برای جلوگیری از بروز آنها می‌توان از یک آنتولوژی جهت طبقه‌بندی و بیان ویژگی حملات استفاده نمود. در این مقاله با استفاده از تکنیک‌های داده‌کاوی ابتدا یک آنتولوژی برای حملات کامپیوتری طراحی می‌شود. سپس با استفاده از این آنتولوژی یک سیستم چند عامله برای تشخیص نفوذ طراحی و پیاده‌سازی می‌گردد. نتایج بدست آمده حاکی از دقت خوب این سیستم در مقایسه با سیستم‌های مشابه می‌باشد.

واژه های! کلیدی

تشخیص نفوذ در شبکه‌های کامپیوتری، داده‌کاوی، آنتولوژی حملات کامپیوتری، سیستم‌های تشخیص نفوذ توزیع شده، تشابه معنایی.

۱- مقدمه !

امروزه با گسترش شبکه‌های کامپیوتری، بحث امنیت شبکه بیش از گذشته مورد توجه پژوهشگران قرار گرفته است. در این راستا تشخیص نفوذ به عنوان یکی از اجزای اصلی برقراری امنیت در شبکه‌های کامپیوتری شناخته می‌شود که هدف اصلی آن کنترل ترافیک شبکه و تحلیل رفتارهای کاربران می‌باشد. در این خصوص تحقیقات زیادی انجام پذیرفته و از تکنیک‌های مختلف برای شناسایی نفوذ به شبکه استفاده شده است.

با توجه به رشد روز افزون وب معنایی استفاده از این روش هم مورد توجه محققین قرار گرفته است. استفاده از یک زبان تعریف استاندارد و توجه به معنا و قابلیت بکارگیری در شرایطی که درک معنایی و نه نحوی دقیقی از مسئله وجود دارد از مزایای بکار گیری مفاهیم مرتبط با وب معنایی مانند آنتولوژی است.

به طور کلی در اغلب کارهای انجام گرفته در این حوزه از یک تاکسونومی جهت نمایش ویژگی حملات استفاده شده است. بکارگیری تاکسونومی مشکلات و محدودیت‌های فراوانی در سیستم تشخیص نفوذ ایجاد می‌کند که برای جلوگیری از بروز آنها می‌توان از یک آنتولوژی جهت طبقه‌بندی و بیان ویژگی حملات استفاده نمود. آنتولوژی علاوه بر دارا بودن ویژگی‌های تاکسونومی مزایای عمده دیگری نیز دارد و با استفاده از آن می توان مدل داده ای تشخیص نفوذ را از منطق سیستم کشف نفوذ تفکیک نمود.

از این رو هدف اصلی این تحقیق بر روی استخراج آنتولوژی حملات در حوزه‌ی تشخیص نفوذ شبکه‌های کامپیوتری بنا شده است. برای این منظور تکنیک‌های داده کاوی متفاوتی مانند الگوریتم‌های طبقه‌بندی RIPPER و خوشه بندی HotSpot بر روی مجموعه‌ی داده‌ای NSL-KDD بکار گرفته می شود. با بهرهگیری از این الگوریتم‌ها می‌توان قوانین لازم جهت مشخص نمودن ویژگی کلاس‌های مختلف آنتولوژی حملات را تولید نمود. آنتولوژی بدست آمده به عنوان یک طبقه بند در سیستم تشخیص نفوذ توزیع شده به کار گرفته می‌شود.

با توجه به اینکه اخیرا مقالات متعددی در خصوص بکارگیری آنتولوژی در سیستم های تشخیص نفوذ نگارش شده است و مرور مطالب خوبی در آنها انجام شده است، در این مقاله از مرور ادبیات

(بعلت کمبود جا) صرفنظر شده است. علاقه مندان می‌توانند به [1] مراجعه نمایند.

در این مقاله پس از توضیح نحوه طراحی آنتولوژی حملات با استفاده از تکنیک‌های داده‌کاوی بـه معمـاری سیـستم و نیـز ارزیابی و مقایسه آن با سیستم‌های مشابه پرداخته می‌شود.

۲- طراحی آنتولوژی حملات!

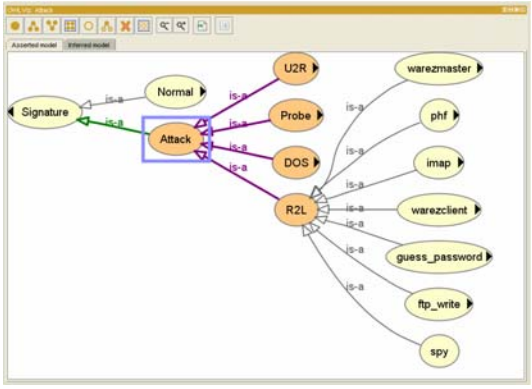
برای طراحی آنتولوژی مورد نظر از نرم‌افزار Protégé (http://protege.stanford.edu) استفاده شده است. این نرم افزار یک نرم افزار منبع باز رایگان مبتنی بر پایگاه دانش می‌باشد، همچنین برای بررسی میزان سازگاری اجزای آنتولوژی با یکدیگر از نرم‌افزار Racer (http://www.sts.tu-harburg.de/~r.f.moeller/racer/) به عنوان نرم افزار "استدلال کننده " استفاده شده است.

با توجه به اینکه که در حال حاضر آنتولوژی کامل و جامعی در زمینه ی حملات کامپیوتری وجود ندارد، کلیه ی مراحل ایجاد آنتولوژی از ابتدای فاز طراحی به طور کامل انجام شده است. مراحل عملی طراحی و ایجاد یک آنتولوژی عبارت است از: [2]

۱. تعریف کلاس ها در آنتولوژی
۲. مرتب کردن سلسله مراتب کلاس ها در آنتولوژی
۳. تعریف ویژگی ها و خصایص و همچنین معین کردن مقادیر مجاز برای آن ها
۴. مقدار دهی به خصایص برای نمونه های تعریف شده در آنتولوژی

آنتولوژی طراحی شده باید پوشش دهنده حملات معمول واقع شده در شبکه ها و سیستم های کامپیوتری باشد. این آنتولوژی باید بتواند ویژگی ها و شرایط وقوع حمله در شبکه ها را پوشش داده و به تشخیص حملات کمک کند. برای اینکه سیستم تشخیص نفوذ قدرت بیشتری داشته و توانایی تشخیص نفوذ های جدید را نیز داشته باشد علاوه بر رفتار های نفوذی بایستی الگوی رفتار های نرمال نیز در دل این آنتولوژی گنجانده شوند.انجام هر یک از این مراحل و جمع آوری اطلاعات برای آن ها، استخراج دانش و استنتاج روابط ما بین شان جزء امور بسیار پیچیده در طراحی آنتولوژی می باشد، البته تست و ارزیابی آنتولوژی طراحی شده نیز دارای پیچیدگی های خاص خود است.

برای تعیین سلسله مراتب کلاس های آنتولوژی روش های مختلفی وجود دارد: روش بالا به پایین، روش پایین به بالا و روشی که ترکیبی از این دو روش می باشد[2]. در این پژوهش در طراحی آنتولوژی "حملات کامپیوتری" از روش بالا به پایین استفاده شده است. در "آنتولوژی حملات کامپیوتری" مفهوم Signature کلی‌ترین مفهوم در دامنه‌ی تشخیص حملات کامپیوتری می‌باشد، که می‌تواند دارای زیرشاخه‌های Attack و Normal باشد. زیر شاخه ی Attack در واقع شامل امضای تمام نفوذهای شناخته شده و موجود در داده‌های آموزشی NSL-KDD بوده که خود دارای چهار زیرشاخه ی کلی DoS، U2R، R2L و Probe می‌باشد. هر کدام از این کلاس های حملات نیز خود دارای انواع مختلفی می‌باشند، مثلاً کلاس حملات R2L دارای زیر کلاس‌های ftp_write، guess_password، imap، phf، spy، warezclient و warezmaster می‌باشد. در این مرحله دسته بندی اصلی حملات کامپیوتری انجام شده است، در ادامه نیز به بررسی و تکامل حملات و زیرشاخه های حملات پرداخته می شود. در شکل ۱ شمایی از کلاس حملات R2L و زیرکلاس های آن نشان داده شده است.



شکل ۱: شمایی از کلاس حملات R2L و زیرکلاس های آن

در تعریف کلاس های آنتولوژی، کلاس عمومی دیگری تحت عنوان ValuePartition برای بیان ویژگی ها و ارتباطات حملات مختلف با یکدیگر تعریف شده است. هدف از طراحی این کلاس کمک گرفتن برای بیان ویژگی ها و ارتباطات سایر کلاس ها با یکدیگر می باشد. کلاس هایی مثل کلاس ValuePartition در واقع جزء آنتولوژی اصلی نبوده و تنها برای کمک به طراحی آنتولوژی اصلی و مقدار دهی ویژگی های کلاس های مختلف موجود در آنتولوژی تعریف می شوند.

در آنتولوژی طراحی شده در این پژوهش، کلاس ValuePartition دارای ۴۱ زیر کلاس بوده که هر کدام از آن ها دارای نمونه هایی هستند که این نمونه ها مقادیر مجاز برای ویژگی های ۴۱ گانه ی اتصالات مربوط به حملات کامپیوتری می باشند.

در مرحله ی سوم طراحی می بایست اصطلاحات و موارد مهم و حساس در آنتولوژی لیست شوند. مثلاً در رابطه با حملات کامپیوتری ویژگی هایی چون "پروتکل مورد استفاده"، "سرویس مورد استفاده" و... می توانند جزء موارد اصلی و مهم در رابطه با حملات در آنتولوژی "حملات کامپیوتری" باشند.

در این مرحله مقادیر مجاز برای هر کدام از این ویژگی ها نیز بایستی مورد بررسی قرار گرفته و ویژگی ها و خصایص کلاس های آنتولوژی تعیین شوند. تنها تعریف کلاس ها برای داشتن آنتولوژی کافی نیست، برای اینکه آنتولوژی بتواند اهداف مورد نظر طراحی را برآورده کند و پاسخ مناسبی برای سوال ها و پرس و جوها داشته باشد می بایست خصایص و ویژگی های کلاس های آن تعیین شده و محدودیت های آن ها نیز مشخص شود.

به عنوان مثال در آنتولوژی "حملات کامپیوتری" از ۲۶۴۶ رکورد اتصال مربوط به کلاس حملات Smurf تمامی آن ها دارای سرویس ecr_i بوده و تعداد بایت های داده ای فرستاده شده از مبدا به مقصد بیشتر از ۵۲۰ بایت می باشد و یا اینکه در کلاس حملات Land حتماً می بایست خصوصیت land برابر با یک باشد.

1) (service = ecr_i) and
(src_bytes >= 520) => class=smurf
2)(land = 1) => class=land

اگر داده‌های آزمایشی مورد استفاده در این تحقیق بسیار کمتر از مقدار کنونی بود یک شخص خبره با آنالیز آن ها و انجام محاسبات آماری بر روی آن‌ها می توانست مجموعه ای از این قوانین را بدست آورد. همان طور که در [۱] بیرون کشیدن این قوانین از دل داده های آزمایشی توسط یک فرد خبره انجام شده بود. این روش علاوه بر وقت گیر بودن، دقت پایینی نیز خواهد داشت و در مجموعه ی داده های آزمایشی عظیم روش دستی عملاً غیر ممکن می باشد. مجموعه ی این عوامل باعث گرایش به سمت تکنیک های داده کاوی گردید.

برای بهبود برخی مشکلات ذاتی موجود در مجموعه داده های آزمایشی KDD مجموعه داده های آزمایشی NSL-KDD پیشنهاد شده است[۳]. اگر چه این مجموعه جدید از برخی مشکلات بیان شده مبری نبوده و نمی تواند یک نمونه واقعاً کامل و عملی برای رکورد های اتصال شبکه های موجود باشد، با این وجود این مجموعه می تواند توسط محققان به عنوان یک مجموعه مناسب برای مقایسه سیستم های کشف نفوذ مختلف مورد استفاده قرار گیرد.

در این تحقیق برای استخراج قوانین از مجموعه ی داده های آموزشی پس از انجام آزمایشات متعدد و بررسی روش های مختلف، روش RIPPER برای یادگیری قوانین انتخاب شد.

برای داده کاوی در داده‌های عظیم نرم‌افزارهای متعددی طراحی شده که بسیاری از روش‌های داده کاوی را پیادسازی کرده و کار با آن‌ها نیز بسیار راحت می‌باشد. از جمله معروفترین و پرکاربردترین آن‌ها می‌توان به Weka و RapidMiner اشاره نمود. در این تحقیق از RapidMiner برای بدست آوردن ویژگی های حملات و نفوذ های کامپیوتری مختلف با استفاده از تکنیک های داده کاوی، استفاده شده است.

پس از پیش پردازش‌های اولیه، فایل متنی مربوط به اتصالات مختلف به الگوریتم RIPPER داده شد. نتیجه این داده کاوی ها مجموعه ای از قوانینی بوده که ویژگی‌های حملات مختلف را در قالب تعدادی قانون ساده و قابل فهم ارائه می‌نماید. در شکل ۲ چند نمونه از این قوانین نمایش داده می شود.

1) (num_shells >= 1) and (dst_host_same_src_port_rate <= 0.01) => class=perl (4.0/1.0)
2) (root_shell >= 1) and (src_bytes <= 51) and (service = http) => class=phf (4.0/0.0)
3) (num_file_creations >= 1) and (dst_host_srv_count <= 1) and (num_shells >= 1) => class=multihop (3.0/1.0)
4) (dst_host_count <= 2) and (service = login) => class=ftp_write (2.0/0.0)
5) (dst_host_count <= 2) and (service = ftp_data) and (dst_host_srv_count >= 84) and (dst_host_srv_count <= 85) => class=ftp_write (2.0/0.0)
6) (num_file_creations >= 1) and (src_bytes <= 116) and (src_bytes >= 104) => class=ftp_write (2.0/0.0)
...

شکل ۲: چند نمونه از قوانین RIPPER مستخرج از داده کاوی!

قوانین مستخرج از داده کاوی هم در فاز سوم (تعریف ویژگی ها و خصایص و معین کردن مقادیر مجاز برای آن ها) و هم در فاز چهارم (مقدار دهی به خصایص برای نمونه های تعریف شده) مورد استفاده قرار می گیرند. پس از تولید قوانین توسط RapidMiner تمام شرط های تک تک قانون ها (قسمت مقدم قانون ها) به عنوان مقادیر مجاز برای آن خصیصه در نظر گرفته شده و در بخش ValuePartition در زیر کلاس مربوطه به عنوان یک نمونه اضافه می شود. به عنوان مثال قانون زیر را در نظر بگیرید:

(service = ecr_i) and (src_bytes >= 520) => class=smurf

این قانون دارای دو شرط یا مقدم بوده و حداکثر مقدار ویژگی src_bytes برابر با ۱۳۷۹۹۶۳۸۸۸ می باشد. لذا برای ویژگی service در بخش ValuePartition مقدار ecr_i به عنوان یک نمونه ی مجاز در کلاس service_ValuePartition افزوده شده و برای ویژگی src_bytesنیز در بخش ValuePartition مقدار زیر به عنوان یک نمونه ی مجاز در کلاس src_bytes_ValuePartition افزوده می شود. پس از افزودن تمام مقادیر مجاز به کلاس های ValuePartition با استفاده از شروط یا همان بخش مقدم قوانین فاز سوم طراحی آنتولوژی کامل شده و نوبت به آخرین مرحله ی ساخت آنتولوژی می رسد.

آخرین مرحله و یا فاز چهارم از طراحی آنتولوژی مربوط به تعیین محدودیت‌ها و مقادیر مجاز ویژگی‌ها و خصایص کلاس‌های آنتولوژی می‌باشد. برای ایجاد این قیود و محدودیت ها از ویژگی شئی استفاده می‌شود. برای این کار برای خصیصه‌ی src_bytesویژگی شئی has_src_bytes با دامنه‌ی Signature و برد src_bytes_ValuePartition از نوع تابعی

تعریف می شود ، این کار برای تمام خصیصه های ۴۱ گانه انجام می گیرد.

در آخرین مرحله نیز با انتخاب کلاس مربوطه قیود مورد نظر برای آن تنظیم می شود. این مرحله از کار در مقایسه با بخش های قبلی بسیار وقت گیر بوده و احتیاج به دقت زیادی دارد. برای مثال قانون زیر را در نظر بگیرید:

(dst_host_srv_count <= 2) and (rerror_rate >= 0.25) and
(service = private) and (dst_host_diff_srv_rate <= 0.04) and (count <= 8) => class=portsweep

برای این قانون قیود و محدودیت های زیر به آنتولوژی افزوده می شوند:

portsweep:
has_dst_host_srv_count {dst_host_srv_count_0-2}
has_rerror_rate {rerror_rate_0.25-1}
has_service {private}
has_dst_host_diff_srv_rate {dst_host_diff_srv_rate_0-0.04}
has_count {count_0-8}

این عمل برای تمام کلاس ها انجام می گیرد. به واسطه ی قیود و محدودیت های تعیین شده برای هر یک از کلاس ها، در آنتولوژی حملات کامپیوتری، می توان روابط و ارتباط های کلاس های مختلف حملات را در آنتولوژی پیشنهادی بیان کرد.

برای بررسی صحت و درستی آنتولوژی طراحی شده در این پژوهش از دو روش استفاده شده است. ابتدا صحت و درستی آنتولوژی از نظر منطقی بررسی شده است. برای این کار از نرم افزار استنتاج گر Racer استفاده شده است. به کمک این نرم افزار می توان سازگاری کلاس های مختلف تعریف شده در آنتولوژی را بررسی کرد. مثلاً دو کلاس "نرمال" و "حمله" باید به صورت جدا از هم تعریف شوند چراکه یک وضعیت رخ داده در شبکه نمی تواند به طور همزمان هم در دسته ی نرمال قرار بگیرد و هم در دسته ی حمله.

لازم به ذکر است که کار طراحی آنتولوژی یک عمل تکرار شونده بوده و به مرور زمان و با انجام تست های مختلف در سیستم های واقعی می توان به یک آنتولوژی کامل دست پیدا کرد. انتظار این است که با تکامل آنتولوژی طراحی شده میزان خطای آن کاهش یابد. پس از انجام تست-ها و بازبینی های فراوان بر روی آنتولوژی طراحی شده، آنتولوژی حملات کامپیوتری برای استفاده در سیستم تشخیص نفوذ پیشنهادی آماده شده و فایل OWL آن قابل استفاده در پیاده سازی این سیستم می باشد.

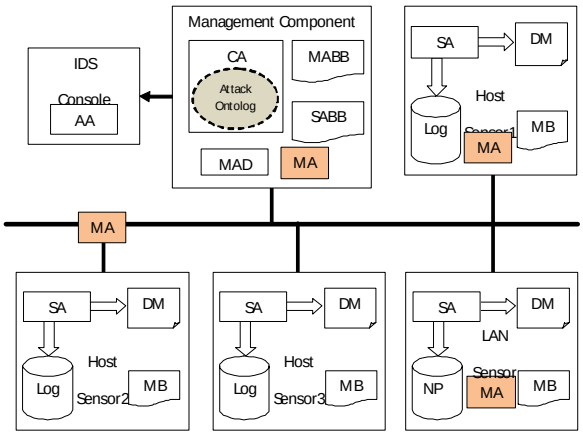
۳- طراحی معماری سیستم!

در این بخش معماری یک سیستم تشخیص نفوذ مبتنی بر وب معنایی ارائه شده است که ار آنتولوژی طراحی شده برای انجام عمل تشخیص نفوذ استفاده می شود. مدل پیشنهادی یک سیستم چند عاملی است که دارای تعدادی عامل تشخیص نفوذ ثابت و متحرک مختلف و یک عامل مرکزی بوده که برای تشخیص یک نفوذ به صورت بهینه با یکدیگر همکاری می نمایند

برای استفاده از آنتولوژی طراحی شده ی و تست آن سعی شد تا شرایط یک شبکه واقعی برای آن شبیه سازی شود. بنابراین یک محیط چند عاملی برای رسیدن به این هدف در نظر گرفته شد. استفاده از محیط مبتنی بر عامل کمک می کند تا از قابلیت ها و رفتار های عامل ها در برقراری ارتباط با یکدیگر و تعامل با یکدیگر جهت شبیه سازی بهتر محیط شبکه برای تست سیستم پیشنهادی استفاده شود. برای طراحی محیط مبتنی بر عامل از زبان برنامه نویسی Java به همراه کتابخانه ی JADE استفاده شده است.

در تحقیق حاضر در طراحی سیستم پیشنهادی (DIDMO)مسائلی مانند انعطاف پذیری، مقیاس پذیری، مستقل بودن از پلتفرم و قابلیت اطمینان در نظر گرفته شده است. معماری کلی DIDMO در شکل ۳ نشان داده شده است و در ادامه ویژگی های هر یک از مولفه های آن تشریح شده و به طرز کار سیستم و طریقه ی تعاملات این عامل ها پرداخته می شود.

عامل ایستا :(SA) نقش SA در معماری پیشنهادی یک مانیتور کننده یا حسگر می باشد و در مولفه های نصب شده بر روی تمام میزبان های تحت نظارت و مؤلفه های موجود بر روی حسگر های شبکه یک SA قرار دارد که در ادامه به نقش آن پرداخته می شود.



شکل ۳- معماری DIDMO

عامل سیار(MA): این عامل ها وظیفه‌ی حرکت بین عامل های ایستا، دریافت اطلاعات مربوط به اتصالات، حذف اطلاعات اضافی، ادغام آنها و تحویل این اتصالات به عامل مرکزی را به عهده دارند.

مولفه‌ی مدیریت(MC) : این مؤلفه شامل آنتولوژی حملات در قالب یک فایل OWL بوده و با کمک یک عامل مرکزی استدلال و استنتاج بر روی این آنتولوژی را انجام داده، وضعیت اتصالات دریافتی از عامل های سیار را تشخیص داده و در صورت نفوذی بودن یک اتصال، هشدار مربوطه را به عامل هشدار تحویل می دهد.

عامل مرکزی(CA) : عامل اصلی در سیستم DIDMO می باشد که مجهز به آنتولوژی "حملات کامپیوتری" طراحی شده در این پایان نامه بوده و در واقع نقش نهایی تشخیص نفوذ را بر عهده دارد.

عامل هشدار(AA) : این عامل هشدار های دریافتی از مؤلفه ی مدیریت را به میز فرمانی که توسط مدیر امنیت شبکه مشاهده می شود می فرستد.

گسیل کننده ی عامل های سیار(MAD) : نقش اصلی MAD اعزام عامل های سیار (برای جمع آوری داده ها) به میزبان هایی که قبلاً هشدار اتصال مشکوک داشته اند می باشد.

تابلوی اعلانات عامل های ایستا :(SABB) نقش این مؤلفه در سیستم پیشنهادی تبادل اطلاعات بین عامل های ایستا و عامل مرکزی بوده و در واقع بخشی از مؤلفه ی مدیریت می باشد.

تابلوی اعلانات عامل های سیار(MABB) : نقش این مؤلفه در سیستم پیشنهادی تبادل اطلاعات بین عامل های سیار و عامل مرکزی بوده و در واقع بخشی از مؤلفه ی مدیریت می باشد.

تابلوی پیغام(MB) : این تابلو برای تبادل اطلاعات بین SAها بوده و در تمام میزبان های تحت نظارت و حسگر های شبکه موجود می باشد.

مدل داده ای (DM): مدل داده ای نیز مانند تابلوی پیغام بخشی از مولفه ی نصب شده بر روی تمام میزبان-های تحت نظارت و حسگر ها بوده و به تشخیص مشکوک بودن یک اتصال کمک کرده و باعث افزایش کارایی سیستم می شود.

۳-۱-تعاملات عامل ها و تشخیص نفوذ!

همان طور که گفته شد در معماری پیشنهادی SA نقش یک حسگر را به عهده داشته و در تمام مولفه‌های نصب شده بر روی میزبان های تحت نظارت و مؤلفه‌های موجود بر روی حسگرهای شبکه یک SA تعبیه شده است. مؤلفه‌های نصب شده بر روی میزبان‌های تحت نظارت با مشاهده‌ی فایل های ثبت وقایع میزبان‌ها که شامل System ،Application Logs Calls و System Status بوده سعی در بیرون کشیدن ویژگی‌های مبتنی بر میزبان یک اتصال می‌نماید. در هر بخش از شبکه نیز یک مولفه حسگر شبکه وجود دارد که شامل یک SAبوده که نقش بیرون کشیدن خصوصیات مبتنی بر شبکه ی یک اتصال (خصوصیات ترافیکی و ذاتی یک اتصال) بر اساس محتویات بسته های عبوری در شبکه را برعهده دارد.

عامل های SA پس از بدست آوردن این ویژگی ها برای یک اتصال آن ها را با مدل داده ای خود مقایسه و در صورت مطابقت، این اتصال به عنوان اتصال مشکوک تشخیص داده می شود و SA پیغام هشداری برای MAD ارسال می نماید. این پیغام شامل شماره ی شناسایی اتصال مربوطه، آدرس میزبان و مجموع ویژگی های مشکوک اتصال بوده و نقش اطلاع رسانی به MAD برای ادامه ی روند تشخیص را بر عهده دارد.

برای جلوگیری از بمباران مؤلفه ی مدیریت توسط تمام ویژگی های یک اتصال بایستی تنها ویژگی های لازم و ضروری را به مؤلفه ی مدیریت تحویل داده تا این مؤلفه سریعتر و کارامد تر عمل نماید. برخی مواقع ممکن است که برای تشخیص یک نفوذ تنها ۲ یا ۳ خصیصه از ۴۱ خصیصه ی یک اتصال مورد نیاز باشد و با ارسال تمام ۴۱ خصیصه به مؤلفه ی مدیریت سربار محاسباتی زیادی بر روی آن اعمال شود. برای رسیدن به این هدف ابتدا عامل مرکزی پیغام های دریافتی از SABBرا برداشته و به دنبال شباهت های معنایی بین آنتولوژی و خصیصه های موجود در این پیام ها می گردد. پس از یافتن بیشترین شباهت ها مجموعه ی خصیصه های مورد نیاز را برای تصمیم گیری نهایی در رابطه با وضعیت یک اتصال به همراه شماره ی شناسایی اتصال و آدرس میزبان در MABB قرار می دهد. عامل MAD نیز با توجه به اطلاعات موجود در MABBیک عامل سیار (MA) به میزبان مربوط گسیل می کند.

این MA تمام خصیصه های مورد نیاز مبتنی بر میزبان و خصیصه های مورد نیاز مبتنی بر شبکه (خصیصه های ذاتی و خصیصه های ترافیکی) را ازSA های موجود در میزبان های تحت نظارت و حسگر های شبکه بدست آورده و پس از تجمع این خصوصیات و ایجاد اتصال مورد نظر با خصیصه های درخواست شده توسط مؤلفه ی مدیریت، این اتصال را به مؤلفه ی مدیریت تحویل می دهد.

در این مرحله مؤلفه ی مدیریت به دنبال پیدا کردن شباهت های معنایی بین این اتصال و آنتولوژی حملات می گردد و پس از پیدا کردن بیشترین شباهت، نوع اتصال را بر اساس این شباهت تعیین و در صورت نفوذی بودن آن، هشدار مناسب را به عامل هشدار (AA) تحویل می دهد. عامل هشدار نیز هشدار های دریافتی از عامل مرکزی را به میز فرمانی که توسط مدیر امنیتی شبکه مشاهده می شود ارسال می کند. علاوه بر این AA مسئول پیشگیری از نمایش هشدار های مشابه در یک بازه ی زمانی معین بوده و اطلاعات

مربوط به این هشدار ها را نیز برای تحلیل های احتمالی آینده نگهداری می‌نماید.

۳-۲-مدل داده‌ای موجود در عامل های ایستا!

برای تولید مدل داده‌ای موجود در عامل های ایستا از نتایج داده کاوی استفاده شده است. با بررسی های آماری بر روی قوانین حاصل از اجرای الگوریتم داده کاوی RIPPER بر روی مجموعه داده های آزمایشی NSL-KDD مشاهده می شود که برخی از خصیصه های اتصال هیچگونه تاثیری در استخراج قوانین نداشته و در برخی از خصیصه های دیگر تنها مقادیر خاصی از آن ها در قوانین مشاهده شده است. خصیصه هایی که در تولید قوانین نقشی نداشته اند عبارتند از:

is_guest_login, is_host_login, num_access_files, su_attempted, num_outbound_cmds, urgent, srv_rerror_rate

به زبان ساده‌تر می‌توان گفت هیچکدام از این هفت خصیصه جزء بخش مقدم هیچ قانونی نبوده و در کارایی روش ارائه شده هیچگونه تاثیری نداشته و جزء خصایص زائد می باشند. با حذف این خصیصه ها از یک اتصال به جای ۴۱ خصیصه در اتصالات ۳۴ خصیصه خواهیم داشت که این کاهش خصیصه به افزایش سرعت تشخیص و کاهش حجم داده های ارسالی بین عامل های مختلف کمک می کند.

با انجام تجزیه و تحلیل بیشتر بر روی قوانین حاصل از داده کاوی دریافت می شود که در برخی از خصیصه های چند مقداری، مقدار این خصیصه حالت بولی دارد یعنی تمامی مقادیر این خصیصه را می توان به دو گروه با مقادیر صفر و مقادیر غیر صفر تبدیل نمود که عملاً تمام مقادیر غیر صفر هیچ تفاوتی با یکدیگر ندارند. لیست خصایصی که دارای این ویژگی می باشند همراه با ویژگی های آن ها در جدول ۱ نشان داده شده است و پیشنهادی نیز برای جایگذاری این خصیصه ها ارائه شده است. احساس می شود که این جایگذاری خصیصه ها در مجموعه ی آزمایشی KDD به کاهش پیچیدگی های زائد این مجوعه کمک شایانی نماید.

جدول ۱ : لیست خصایص قابل اصلاح

نام خصیصه	مفهوم کنونی	مقادیر فعلی	مفهوم جدید پیشنهادی	مقادیر جدید
num_compromised	تعداد وضعیت‌های به خطر افتاده	0-7479	آیا وضعیت خطر افتاده وجود دارد؟	0-1
num_file_creations	تعداد عملیات تولید فایل	0-43	آیا عملیات تولید فایل صورت پذیرفته است؟	0-1
num_root	تعداد دسترسی‌ها به ROOT	0-7468	آیا دسترسی به ROOT صورت گرفته است؟	0-1
root_shell	تعداد دستورات پوسته	0-2	آیا دستورات پوسته اجرا شده اند؟	0-1

سومین نکته ای که در ادامه ی بررسی مشاهده گردید و به تولید مدل داده ای کمک زیادی نمود این بود که برای یک خصیصه ی خاص تنها برخی از مقادیر ممکن برای آن در قوانین آموخته شده به کار رفته است و سایر مقادیر آن بی اهمیت بوده اند (این مقادیر هیچ نقشی در تصمیم گیری وضعیت یک اتصال ندارند). با بهره گیری از این نگرش می توان به سادگی مشکوک یا غیر مشکوک بودن یک اتصال را تعیین نمود. اتصالات مشکوک اتصالاتی هستند که ممکن است جزء یکی از کلاس های نفوذ باشند که تصمیم گیری نهایی درباره وضعیت این اتصالات در عامل مرکزی انجام می پذیرد. ولی اتصالات غیر مشکوک بدون شک جزء کلاس های نرمال بوده و ضرورتی به ارسال این اتصال ها به عامل مرکزی وجود ندارد. برای تشخیص مشکوک بودن یک اتصال کافی است یک مدل داده ای ساده شامل تمام حالات ممکن برای هر یک از خصایص یک اتصال را در اختیار داشت، با یک مقایسه ی ساده می توان مشکوک بودن یک اتصال را تشخیص داد. یک اتصال زمانی مشکوک می باشد که حداقل یکی از خصایصش یکی از مقادیر ممکن مربوط به آن خصیصه در مدل داده ای را داشته باشد، و زمانی غیر مشکوک خواهد بود که هیچکدام از خصیصه هایش یکی از مقادیر ممکن مربوط به آن خصیصه در مدل داده ای را نداشته باشد.

برای پیاده سازی اموری که مربوط به ارتباط با آنتولوژی، استخراج اطلاعات از آن و... می باشد، از بسته ی نرم افزاری Jena (*http://jena.sourceforge.net/*) استفاده شده است. Jenaیک چارچوب به زبان Java است که مناسب برای ایجاد برنامه های کاربردی مبتنی بر وب معنایی می باشد. بسته ی نرم افزاری Jena محیط برنامه نویسی برایRDF ، RDFS، OWLو SPARQL بوده و شامل یک موتور استنتاج مبتنی بر قانون نیز می باشد.

عامل مرکزی حساس ترین بخش سیستم طراحی شده می‌باشد. عامل مرکزی با دریافت یک گزارش از هر یک از عامل‌های سیار، موقعیت وضعیت گزارش شده را در آنتولوژی "حملات کامپیوتری" بررسی می‌کند که این عمل را به کمک زبان پرس و جو SPARQL انجام می‌دهد. عامل مرکزی به کمک این گونه پرس‌وجوها می‌تواند موقعیتِ وضعیت گزارش شده را در آنتولوژی "حملات کامپیوتری" استخراج کند. مثلاً در این پرس‌وجوها ویژگی‌هایی نظیر نوع پروتکل استفاده شده، نوع سرویس استفاده شده، وضعیت پرچم ها اعم از نرمال یا خطا بودن، تعداد بایت اطلاعاتی که برای مقصد ارسال شده و.... با وضعیت‌ها و ویژگی های تعریف شده در آنتولوژی مطابقت داده شده و به این صورت، وضعیت اتصال گزارش شده، توسط عامل مرکزی آنتولوژی "حملات کامپیوتری" تعیین می‌شود. به

عبارت دیگر کلاسی از آنتولوژی که کاملاً با وضعیت گزارش داده شده مطابقت داشته باشد به عنوان کلاس این اتصال برگردانده می شود.

یکی از معایب این روش این است که در صورتی که وضعیت جدیدی در شبکه رخ دهد که تفاوت بسیار اندکی با یکی از وضعیت های مورد استفاده در ساخت آنتولوژی داشته باشد، این سیستم قادر به طبقه بندی آن نمی باشد. در این تحقیق برای رفع این مشکل از یک تکنیک ساده برای یافتن شباهت-های معنایی استفاده شده است. این تکنیک به جای تولید جواب قطعی یک جواب غیر قطعی تولید می نماید. یعنی اگر یک اتصال با هیچ یک از کلاس های آنتولوژی حملات مطابقت نداشته باشد، گره ای که بیشترین شباهت را با این اتصال دارد به عنوان کلاس مورد نظر برای طبقه بندی این اتصال انتخاب می نماید. همچنین میزان این شباهت را نیز در قالب یک عدد اعشاری مثبت کوچک تر از یک بیان می نماید. بیشترین مقدار شباهت یک بوده و زمانی اتفاق می افتد که یک کلاس از "آنتولوژی حملات" با اتصال مورد آزمایش کاملاً مطابقت داشته باشد.

همچنین برای بدست آوردن میزان قرابت بین تمام مقادیر خصیصه های اسمی هم نوع با یکدیگر یک الگوریتم ساده طراحی شد. در این الگوریتم تمام فرضیات بالا در نظر گرفته شد و با بررسی های آماری بر روی مجموعه ی آزمایشی NSL-KDD و مقادیر مجاز برای خصایص اسمی آن ها میزان قرابت بین تک تک مقادیر مجاز یک خصیصه ی اسمی بدست آمد. لازم به ذکر است که از تمام ۴۱ خصیصه ی داده های آزمایشی KDD تنها سه خصیصه ی flag، service و protocol_type دارای مقادیر اسمی می باشند. الگوریتم پیشنهادی در محیط C ++ پیاده سازی شد و نتایج قابل توجهی تولید نمود. خروجی این الگوریتم برای مجموعه داده های آزمایشی سه ماتریس دو بعدی بود که هر ماتریس به یک خصیصه ی اسمی تعلق داشت و مقدار درایه موجود بر روی سطر iام و ستون jام میزان قرابت بین iامین و jامین مقدار این خصیصه را بیان می نمود که یک عدد اعشاری مثبت کوچک تر از یک خواهد بود. مقدار صفر عدم وجود قرابت و مقدار یک وجود قرابت کامل را بیان می نماید. این روابط در جدول شماره ۲ نشان داده شده است.

جدول ۲: قرابت موجود بین پروتکلهای icmp، tcp و udp

	icmp	**tcp**	**Udp**
icmp	1	0	0.81
tcp	0	1	0.16
udp	0.81	0.16	1

۴- بررسی نتایج و ارزیابی سیستم!

برای آزمایش کارایی سیستم از میان مجموعه داده های تست NSL-KDD، ده نمونه داده ی تست هر یک شامل ۱۲۰۰۰ رکورد به طور تصادفی انتخاب کرده و کارایی سیستم پیشنهادی با استفاده از این مجموعه بررسی می شود

نتایج طبقه بندی مجموعه داده های تست براساس معیار های تعریف شده و هر یک از نسخه های تست سیستم DIDMO در جدول ۳ ذکر شده است. در این جدول نتایج قابل توجهی برای نسخه های حاصل از آموزش با سری داده های مختلف تست ارائه شده که حاکی از مؤثر بودن سیستم پیشنهادی و کارایی بالای آن می باشد.

جدول ۳: ماتریس برهم ریختگی برای مجموع تمام ده نمونه ی داده های تست مجموعه داده های NSL-KDD بر روی سیستم تشخیص نفوذ DIDMO

		Predicted					دقت
		N	PRB	DoS	U2R	R2L	
Actual	N	64015	67	45	2	42	0.99757
	PRB	177	10893	23	0	3	0.98171
	DoS	78	18	43643	0	1	0.99778
	U2R	47	0	0	7	3	0.12281
	R2L	136	7	0	0	816	0.85089
هشدار غلط		0.0067	0.0083	0.0015	0.2222	0.0566	CPE= 0.1129

در ادامه برای ارزیابی هر چه بیشتر رویکرد فوق، سیستم DIDMO با چند روش یادگیری ماشین که نتایج آزمایشات خود را بر روی داده های KDD ارائه کرده اند مقایسه شد. جدول ۴ کارایی هر یک از این روش ها را مقایسه می نماید. روش ارائه شده در بعضی از کلاس-های حمله کارایی بهتری نسبت به دیگر رقیبان ارائه کرده و مقدار CPE برابر با ۰,۱۱۲۹ بیانگر توانایی این سیستم در تشخیص نفوذ می باشد. از نتایج بدست آمده می توان استنباط نمود که سیستم DIDMO کارایی خوبی در تشخیص نفوذ در شبکه داشته و نرخ تشخیص و نرخ هشدار های غلط قابل قبولی نیز ارائه می نماید.

جدول ۴: مقایسه سیستم های مختلف تشخیص نفوذ

CPE	**FA**	**DTR**	**R2L**	**U2R**	**DoS**	**Prb**	**N**	*الگوریتم*
0.1129	**0.2**	**99.2**	**85**	**12.2**	**99.7**	**98.1**	**99.7**	**DIDMO (ours)**
0.015	3	99.9	n/r	n/r	99.9	n/r	n/r	SWIDS[1]
0.1579	1.9	95.3	31.5	14.1	99.5	84.1	98.2	ESC-IDS[4]
n/r	3.5	94.4	12.4	76.3	99.7	86.8	96.5	RSS-DSS[5]
0.2024	n/r	n/r	31.2	93.6	96.7	99.2	97.4	Parzen-Window[6]
0.2285	n/r	n/r	9.6	29.8	97.3	88.7	n/r	Multi-Classifier[7]
0.2331	0.6	91.8	8.4	13.2	97.1	83.3	99.5	Winner of KDD[8]
0.2356	0.6	91.5	7.3	11.8	97.5	84.5	99.4	Runner Up of KDD[9]
0.2371	0.4	91.1	10.7	6.6	96.9	73.2	99.5	PNrule[10]

البته نتایج جدول در بعضی از سطر ها می تواند غیر عادلانه باشد، برای مثال در [۱] تنها به تشخیص رفتار حملات DoS پرداخته شده است و سیستم تنها قادر به تشخیص DoS یا غیر DoS بودن رکورد های اتصال بوده و هیچ اطلاعاتی را در مورد نوع رفتار های غیر DoS ارائه نمی کند. بیان حقیقت فوق دلیلی برای این مدعا نیست که چون DIDMO یک طبقه بندی چهار کلاسه را برای حملات ارائه می کند روش مناسب تری می باشد، بلکه توجه ی خواننده را به این نکته جلب می نماییم که در روش فوق وقتی رکوردی به عنوان غیر DoS شناخته می شود کلاس آن هر چه که باشد به عنوان یک الگوی طبقه بندی شده ی درست در آن کلاس قرار می گیرد ولی در روش ارائه شده در این مطالعه بعضی از الگو ها علیرغم تشخیص درست به عنوان غیر DoS ممکن است در کلاس نادرست طبقه بندی شوند. به عبارت دیگر در روش فوق تشخیص یک رفتار غیر DoS به معنی طبقه بندی درست است در حالی که در روش های دیگر که همانند روش ما عمل می کنند تشخیص حمله به معنای طبقه بندی درست نیست و امکان طبقه بندی اشتباه وجود دارد. اثبات این مدعا نرخ تشخیص حملات بالاتر در روش ارائه شده است که نشان می دهد DIDMO در تشخیص حملات بسیار خوب عمل می نماید در حالیکه ممکن است در طبقه بندی کلاس دچار اشتباه شود.

۵- نتیجه گیری!

در این مقاله یک روش تشخیص نفوذ توزیع شده ی مبتنی بر وب معنایی ارائه شده و برای تشخیص وضعیت های رخ داده در سیستم از آنتولوژی "حملات کامپیوتری" استفاده می شود. هر گزارشی که از وضعیت های رخ داده در شبکه برای عامل مرکزی ارسال شود، در آنتولوژی "حملات کامپیوتری" بررسی شده و نتیجه حاصل برای عامل هشدار ارسال می شود.

در سیستم پیشنهادی بر مقیاس پذیری سیستم توجه زیادی شده و می‌بایست بار محاسباتی عامل مرکزی کمترین مقدار ممکن باشد. برای نیل به این هدف بایستی از بمباران عامل مرکزی توسط تمام اتصالات رخداده در شبکه به همراه خصیصه‌های ۴۱گانه‌ی آنها جلوگیری کرده و از یک روش سرکشی استفاده شود. یعنی به جای اینکه یک عامل ایستا تمام اتصالات رخداده در شبکه را همراه با تمام خصوصیات آن ها به عامل مرکزی بفرستد، تنها اتصالات مشکوک را همراه با خصوصیات درخواست شده توسط عامل مرکزی به عامل مرکزی تحویل دهد. از این رو عامل های ایستا به یک مدل داده ای مجهز شده که این مدل نقش بسزایی در این امر ایفا می کند و باعث می شود که برخی از اتصالات با کمک مدل داده ای موجود در عامل های ایستا غیر مشکوک در نظر گرفته شده و عامل های ایستا از ارسال خصیصه های آن ها به عامل مرکزی خود داری نمایند. این کار باعث کاهش حجم تعاملات و افزایش کارایی عامل مرکزی می شود.

برای تولید آنتولوژی کاویده شده، بررسی صحت آنتولوژی "حملات کامپیوتری" و تست سیستم DIDMO طراحی شده در این پایان نامه از مجموعه داده های NSL-KDD استفاده شده است. نتایج حاصل از آزمایش ها و تست های انجام شده، نشان از عملکرد خوب سیستم DIDMO می باشد. معیار هایی مانند: معیار CPE و نرخ تشخیص بدست آمده، نشان از کارایی خوب سیستم DIDMO می باشد. عیب سیستم DIDMO در شناسایی کلاس U2R است که می تواند عملکرد سیستم را تحت تاثیر قرار دهد. البته این اتفاق دور از انتظار نیست و علت آن را می توان در پایین بودن تعداد نمونه های آموزشی در این کلاس حمله جستجو نمود. اما موضوع قابل توجه نرخ بسیار بالای تشخیص بوده که از نقاط قوت این سیستم است.

تحقیق انجام شده گام جدیدی است که از مفاهیم وب معنایی و بخصوص از آنتولوژی مستخرج از داده کاوی در جهت بهبود سیستم های تشخیص نفوذ استفاده کرده است. امکان تکمیل و ادامه ی تحقیق حاضر از جهت های دیگری نیز میسر می باشد. اولین کار دارای اولویت، توسعه و بهبود آنتولوژی حملات کامپیوتری می باشد به گونه ای که بتوان به سادگی بر اساس شباهت های موجود بین مقادیر یک خصیصه و استفاده از تکنیک های مختلف شباهت معنایی نفوذ های جدید و ناشناخته را نیز تشخیص داد.

از طرفی مدل داده‌ای استفاده شده در عامل‌های ایستا یا همان سنسورها بسیار ضعیف بوده و می‌توان با تقویت این مدل داده‌ای بر میزان تشخیص رکوردهای غیرمشکوک افزود، این کار باعث کاهش حجم داده‌های ارسالی بین عامل های ایستا و عامل مرکزی و افزایش مقیاس‌پذیری سیستم می‌شود.

از فعالیت‌های دیگری که در ادامه این پروژه می‌توان انجام داد توسعه آنتولوژی در جهتی است که عمل همکاری بین عامل‌ها توسعه پیدا کند. همچنین ادامه‌ی تحقیق بر روی عمل استنتاج بر روی آنتولوژی به گونه‌ای که بتوان تعامل بین عامل‌های موجود در محیط چند عامله را در جهت تشخیص حملات توزیع شده افزایش داد.

مراجع

[1] F. Abdoli, M. Kahani, "Ontology-based distributed intrusion detection system", 14th International CSI Conference (CSICC09), pp. 65-70, 2009.

[2] F. Natalya, D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and

Stanford Medical Informatics Technical Report SMI-2001-0880, 2001

[3] M. Tavallaee, E. Bagheri, W. Lu, A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set”, IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009 .

[4] A. Nadjaran-Toosi. , M. Kahani, R. Monsefi, “Network Intrusion Detection Based on Neuro-Fuzzy Classification”, ICOCI2006 (Kuala Lumpur, Malaysia, June 6-8, 2006.

[5] D. Song, M. I. Heywood, A. N. Zincir-Heywood, “Training Genetic Programming on Half a Million Patterns: An Example from Anomaly Detection”, IEEE Transactions on Evolutionary Computation, 2005 .

[6] *[Yeu02] D. Y. Yeung, C. Chow, “Parzen-window Network Intrusion Detectors”, Sixteenth International Conference on Pattern Recognition, Quebec City, Canada, pp. 11-15, August 2002.

[7] M. R. Sabhnani, G. Serpen, “Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context”, Proceedings of International Conference on Machine Learning: Models, Technologies, and Applications, Las Vegas, Nevada, 209-215, 2003 .

[8] B. Pfahringer, “Winning the KDD99 Classification Cup: Bagged Boosting”, SIGKDD explorations, 1(2), 65-66, 2000 .

[9] I. Levin, “KDD-99 Classifier Learning Contest LLSoft’s Results Overview”, SIGKDD Explorations, ACM SIGKDD, 1(2) 67-75, 2000 .

[10] R. Agarwal, M. V. Joshi, “PNrule: A New Framework for Learning classifier Models in Data Mining”, Technical Report TR 00-015, Depratment of Computer Science, University of Minnesota, 2000 .

.