دانشگاه صنعتی مالک اشتر

مجتمع دانشگاهی فن‌آوری اطلاعات، ارتباطات و امنیت

پژوهشکده امنیت اطلاعات و ارتباطات

پایان‌نامه دوره کارشناسی ارشد مهندسی فنآوری اطلاعات–امنیت اطلاعات

تجسم حملات شبکه‌ای چند مرحله‌ای مبتنی بر رفتار

جواد کریمی

استاد راهنما:

دکتر علی‌جبار رشیدی

استاد مشاور:

دکتر کوروش داداش تبار

آبان ۱۳۹۴

بسم الله الرحمن الرحيم

دانشگاه صنعتی مالک اشتر

مجتمع دانشگاهی فن‌آوری اطلاعات، ارتباطات و امنیت

پژوهشکده امنیت اطلاعات و ارتباطات

پایان‌نامه دوره کارشناسی ارشد مهندسی فن‌آوری اطلاعات–امینت اطلاعات

تجسم حملات شبکه‌ای چند مرحله‌ای مبتنی بر رفتار

جواد کریمی

استاد راهنما:

دکتر علی‌جبار رشیدی

استاد مشاور:

دکتر کوروش داداش تبار

آبان ۱۳۹۴

دانشگاه صنعتی مالک اشتر

مجتمع دانشگاهی فن‌آوری اطلاعات، ارتباطات و امنیت

پژوهشکده امنیت اطلاعات و ارتباطات

پایان‌نامه‌ی کارشناسی ارشد رشته‌ی مهندسی فن‌آوری اطلاعات-امنیت اطلاعات
مربوط به آقای جواد کریمی

با عنوان

تجسم حملات شبکه‌ای چند مرحله‌ای مبتنی بر رفتار

در تاریخ ۱۳۹۴/۰۸/۱۲ توسط کمیته‌ی تخصصی زیر، مورد بررسی قرار گرفت و با نمره‌ی ۱۹ و درجه‌ی عالی به تصویب رسید:

۱	استاد راهنما	دکتر علی‌جبار رشیدی	(درجه‌ی علمی)	دانشیار
۲	استاد مشاور	دکتر کوروش داداش تبار	(درجه‌ی علمی)	استادیار
۳	استاد داور	دکتر فرید صمصامی	(درجه‌ی علمی)	استادیار
۴	استاد داور	دکتر مرتضی براری	(درجه‌ی علمی)	استادیار
۵	مدیر تحصیلات تکمیلی دانشکده	دکتر محمد فخر دانش	(درجه‌ی علمی)	استادیار

تقدیم به پدر و مادر عزیزم

سپاسگذاری

در ابتدا از خدای متعال که در تمام مسیر این کار مرا یاری داد تشکر می کنم. در مقـام دوم از استاد راهنمای بنده جناب آقای دکتر علی‌جبار رشیدی به خاطر حمایت ها و راهنمایی هایشـان و از جناب آقای دکتر کوروش داداش‌تبار بـه خـاطر حمایـت‌هـا و همراهـی‌هایشـان کمـال تشـکر و سپاسگذاری را دارم. در انتها نیز از تمام دوستانی که در مرکز ادغام داده بـه بنـده یـاری رسـاندند تشکر و قدردانی می نمایم.

جواد کریمی

چکیده

امنیت شبکه بسیار زیاد بر سامانه‌هایی مثل سیستم تشخیص نفوذ تکیه دارند، درحالی‌که هدف این ابزارها شناسایی ترافیک‌های مخربانه و استفاده‌های خرابکارانه از کامپیوترها است. به علت حجم بالای این هشدارها حرکت به سمت همبسته سازی آغاز شد. ولی مشکل آگاه نبودن تحلیلگران از وضعیت واقعی شبکه و حملات ممکن در شبکه هنوز باقی است. هدف این تحقیق افزایش دانش جاری تحلیلگران از حملات شبکه‌ای جاری و حملات محتمل‌تر در آینده است. ما در این تحقیق با استفاده از دادگان حاصل از مسابقه CDX2009 و انجام مراحل همبسته سازی بر روی این دادگان، به تولید رد حملاتی پرداخته‌ایم. همچنین در فرآیند تولید این رد حملات، ما به‌جای الگوی زنجیره مرگ و الگوهای مشابه، از الگویی جدید بهره برده‌ایم. در انتها با استفاده از ویژگی دنباله‌ای بودن این ردهای حمله و مدل‌های مارکوف با طول متغیر، رفتار بعدی یا گام بعدی حمله را پیش‌بینی نموده‌ایم. نتایج حاصل از این تحقیق نشان می‌دهد که الگوریتم پیش‌بینی تطبیق جزئی از نوعA ، با داده آموزشی ۶۰ درصد، در۵۰,۷ درصد موارد و روش از نوع S، در ۵۳,۵ درصد موارد موفق به پیش‌بینی صحیح شده‌اند.

کلیدواژه‌ها: آگاهی وضعیتی، تجسم، الگوی راهنما، رد حمله، زنجیره مارکوف، الگوریتم پیش‌بینی تطبیق جزئی

فهرست عنوان‌ها

عنوان **صفحه**

فهرست جدول‌ها

عنوان **صفحه**

فهرست شکل‌ها

عنوان **صفحه**

فصل ۱– مقدمه

به‌طورکلی وظیفه یک تحلیل‌گر حوزه سایبری، پشتیبانی امنیتی از ساختارهای اطلاعاتی و ارتباطی تحت اختیارش است. تضمین در مورد اینکه این شبکه‌ها در حال حاضر به‌درستی کار می‌کنند و همچنین از دسترس افراد غیرمجاز مانند هکرها یا کرم‌های رایانه‌ای درامان‌اند.

آن‌ها این نتیجه‌گیری را از روی سیاست‌های امنیتی کامل، طراحی سیستم خوب، تنظیمات صحیح سیستم و مدیریت وصله‌های نرم‌افزاری به دست می‌آورند و تا حدی مطمئن می‌شوند. ولی می‌دانیم امنیت امری مطلق نیست بلکه نسبی است. درواقع امنیت صد در صد، به‌خصوص در حوزه دفاع سایبری معنا ندارد.

استفاده از سرویس‌ها و محصولات مبتنی بر IT در بخش‌های مختلف نیاز به فضای سایبری امن را نیز به همراه خود می‌آورد. سامانه‌های شناسایی و حفاظت موجود مثل فایروال‌ها، سامانه‌های تشخیص نفوذ قادر به تشخیص تهدیدهای نسل جدید که هدفمند، ماندگار، پنهان‌کار و ناشناخته هستند نیستند[۱]. حملات سایبری مانند استاکس نت علیه برنامه هسته‌ای ایران، ویروس سرقت کننده دکستر و حملات دیگری که در جنگ‌های سایبری به کشورهای مختلف در سطوح مختلف آسیب می‌رسانند. برای اینکه از کاربران اینترنت و زیرساخت سایبری در برابر تهدیدات مختلف محافظت کنیم به سامانه‌های دفاعی سایبری مقاوم و منعطفی نیاز است که توانایی تصمیم سازی هوشمند برای شناسایی و پیش‌بینی حملات و تهدیدات گوناگون را داشته باشند. سپس اطلاعات به‌دست‌آمده می‌توانند با گروه واکنش و هماهنگی رخدادهای رایانه‌ای (CERT) یا دیگر سازمان‌ها به اشتراک گذاشته شوند تا به‌صورت بلادرنگ آن‌ها آگاهی بهتری از آخرین تهدیدات سایبری پیداکرده و بتوانند اقدامات پیشگیرانه‌ای را قبل از بروز آسیب انجام دهند. در چرخه تأمین امنیت در دنیای مجازی، برای اینکه بتوان فعالیت‌های مشکوک را هشدار داد حسگرهای امنیتی مانند سامانه‌های تشخیص نفوذ، فایروال و... تولید و ارائه شدند. تحلیل‌گر شبکه به این اطلاعات برای رسیدن به یک آگاهی وضعیتی کامل نیاز دارد.

آگاهی وضعیتی، یک قابلیت شناختی انسانی است که در محیط‌های پیچیده خصوصاً در مکان‌هایی که رویدادهای خطرناک با تعداد تکرار زیاد اتفاق می‌افتد اهمیت بیشتری پیدا می‌کند.

از طرفی همزمان با پیچیده شدن و رشد اندازه شبکه‌ها، تعداد و تنوع حسگرهای امنیتی نیز بیشتر می‌شود و به طبع آن، تعداد هشدارهای بیشتری نیز تولید خواهد شد. این مسئله کار را برای تحلیل‌گرهای امنیتی دشوارتر ساخت. در سال ۲۰۰۰، بس[۲] نیاز به ادغام اطلاعات را برای حل این مشکل پیشنهاد کرد. بعدازآن، کارهای زیادی برای همبسته سازی هشدارهای سیستم تشخیص نفوذ انجام شد تا بدین‌وسیله بتوانند، حجم این مقدار داده انبوه را کاهش دهند.

قابل‌ذکر است که این ابزارها و فن‌ها هیچ‌کدام آگاهی وضعیتی نیستند و هیچ‌کدام آگاهی وضعیتی را به‌صورت مستقیم برای تحلیل‌گر انسانی فراهم نمی‌کنند. بلکه به‌جای آن، این مواد تنها کمک یار برای تحلیل‌گر در جهت تأمین اطلاعات برای رسیدن به آگاهی وضعیتی بهتر برای تحلیل‌گر هستند. همچنین با ابزارهای امنیتی در حال حاضر دفاع فعالانه یا ممکن نیست و یا به‌سختی امکان‌پذیر است؛ زیرا همیشه تحلیلگران یک گام قبل‌تر از مهاجمین هستند و پس از عمل او واکنش نشان می‌دهند.

یکی از گام‌هایی که اندسلی در مدل خود به‌عنوان مرحله‌ای مهم در جهت رسیدن به آگاهی وضعیتی معرفی می‌کند، تجسم است [۳]. این ویژگی امکان دفاع فعالانه را به تحلیلگران می‌دهد. تجسم به تحلیل‌گر کمک می‌کند که بتواند حملات سایبری آینده و موجودیت‌های مورد خطر در آینده را پیش‌بینی کرده و عکس‌العمل‌های مناسب (نظیر مهیاکردن وصله‌های مناسب برای موجودیت‌های درخطر، تغییر تنظیمات فایروال‌ها و..) را در کوتاه‌ترین زمان ممکن و قبل از اینکه مهاجم ضربات جبران‌ناپذیری را به آن شبکه وارد کند انجام دهد. تجسم از طریق بررسی چهار ویژگی فرصت، قابلیت، نیت و رفتار ممکن می‌شود. در ادامه این پایان‌نامه ما تنها بر تجسم از طریق رفتار تمرکز خواهیم کرد.

در فصل دوم دلیل نیاز به آگاهی وضعیتی بیان‌شده و این مفهوم تشریح شده است. همانطور که گفتیم هدف این پایان‌نامه تجسم مبتنی بر رفتار است. به همین دلیل فصل سوم این مفهوم را بیشتر بسط داده و به بحث رفتار در حوزه سایبری پرداخته است. در این فصل همچنین مروری بر بر روش های پیش‌بینی حملات و مدل های رفتاری در حوزه سایبر انجام‌شده است.

فصل چهارم به معماری پیشنهادی ما برای تجسم مبتنی بر رفتار می‌پردازد. همچنین فرایندی که در انتهای آن تجسم رفتاری صورت خواهد گرفت بیان می‌شود. در این فصل سطوح مختلف این معماری یعنی سطح درک، فهم و تجسم به صورت جداگانه تشریح شده است. در این فصل الگوی راهنمای و الگوریتم پیش‌بینی کننده پیشنهادی ما نیز معرفی و توضیح داده‌شده است. شایان ذکر است ازآنجاکه هدف اصلی این پایان‌نامه سطح پایانی آگاهی وضعیتی یعنی تجسم بوده است، فرض بر صحت عمل سطوح پیشین گذاشته‌شده است.

فصل پنج به بیان نتایج شبیه‌سازی‌ها و نحوه انجام ارزیابی می‌پردازد. در انتها، ما در فصل شش به جمع‌بندی و نتیجه‌گیری از کلیت کار پرداخته‌ایم و پیشنهادات مان را برای کارهای آینده بیان کرده ایم.

فصل ۲–تجسم حملات سایبری مبتنی بر رفتار

۲–۱– مقدمه

در این فصل بیان می‌شود که چرا بـه آگـاهی وضـعیتی نیازمنـدیم. همچنـین مفهـوم آگـاهی وضعیتی و سطوح آن تعریف و تشریح می‌شود. به مفهوم تجسم در حـوزه‌هـای مشـابه بـا آگـاهی وضعیتی پرداخته شده و مفهوم تجسم بیشتر توضیح داده می‌شود. سپس بیان می‌شـود کـه بـرای تجسم یک رفتار، به چه چیزهایی نیاز است و انواع تجسم بیان مـی‌شـود. مفهـوم رفتـار در حـوزه سایبری بیشتر توضیح داده می‌شود و انـواع مـدلهای رفتـاری کـه پـیش تـر در مقـالات مختلـف بررسی‌شده‌اند نیز بیان می‌شوند. در نهایت نیز رابطه میان انواع تجسم‌ها بیان خواهد شد.

۲–۲–چرا نیازمند طراحی کاربر محور هستیم؟

به‌طور کلی روش طراحی سیستم ها را می‌توان در دو دسته قرار داد: طراحی فناوری محـور[1] و طراحی کاربر محور[2][۴].

به‌طور سنتی در طراحی اکثریت سیستم های موجود از طراحـی فنـاوری محـور اسـتفاده‌شـده است. در این نوع طراحی ابتدا مهندسین حسگر یا سیسـتم هـایی را بـرای انجـام هـر کـاری کـه موردنیاز است طراحی می‌کنند. سپس برای هر یک از این سیسـتم هـا و حسـگرها، نمایشـگری را تعبیه می‌کنند. نمایشگر وظیفه‌اش این است که به اپراتور وضعیت سیستم را اطلاع دهد. به عنوان مثال در یک هواپیما، نمایشگرهایی برای دما، فشار، طول و عرض جغرافیایی و .. طراحی‌شده است.

این طرح با قابلیت توسعه‌پذیر بودن فناوری، مشکل‌زا خواهد بود؛ زیرا هـر چقـدر کـه فنـاوری توسعه پیدا می‌کند و بهبود می‌یابد، نیاز به نمایشگرهای بیشتری نیز احساس می‌شود. در واقع این مشکل همان ویژگی این قرن یعنی عصر اطلاعات است. تا پیش از این عصر، انسان مشـکلش تنهـا

[1] Technology-Centered Design
[2] User-Centered Design

نداشتن اطلاعات کافی بود ولی با آمدن تکنولوژی مسئله دیگری مطرح شد. مسئله‌ای به نام غرق شدن در اطلاعات؛ یعنی به حدی به هر شخص اطلاعات ارائه می‌شود که باز نوعی سردرگمی و استیصال را با خود برای انسان به ارمغان می‌آورد. آکادمی ملی دانش با بررسی ۴۴۰۰۰ از مرگ‌های اتفاق افتاده سالانه در یک سال آمریکا، در حوزه پزشکی بیان کرده اند که حداقل دو سوم این تعداد قابل جلوگیری بوده است[۵]. اکثریت این خطاها ناشی از مسئله‌ای به نام خطای انسانی نبوده است بلکه اکثریت آن نتیجه مستقیم طراحی‌های فناوری محوری بوده است که وظیفه شان بالا بردن سطح عملکردی انسان‌ها بوده است.

جایگزین طراحی بالا، طراحی کاربر محور است. هدف این طراحی پر کردن حفره اطلاعاتی میان انسان‌ها و سیستم ها و رسیدن به سیستم هایی کاراتر است. این سبک طراحی، طراحان را ترغیب به ایجاد واسطهایی می‌کند که با توجه به قابلیت‌ها و نیازمندی‌های اپراتورها ساخته‌شده باشند. نتیجه یک طراحی کاربر محور می‌بایست کاهش بیشتر خطاها و بهبود بهره وری بدون نیاز به قابلیت‌های فناوری جدیدتر باشد. به صورت دقیق‌تر می‌توان گفت هدف این طراحی فراهم کردن واسطهایی است که به افراد این اجازه را بدهد که به صورت کاراتری بفهمند که چه چیز در حال اتفاق افتادن است. در حقیقت انسان‌ها به سیستم هایی نیاز دارند که از آگاهی وضعیتی آن‌ها حمایت کند. آگاهی وضعیتی[1] را می‌توان کلید طراحی کاربر محور نامید[۵].

۲-۳- آگاهی وضعیتی چیست؟

در بخش پیشین متوجه ضرورت طراحی کاربر محور شدیم و فهمیدیم که یک طراحی زمانی حقیقتا کاربر محور است که بتواند آگاهی وضعیتی را فراهم نماید. مبدا این اصطلاح حوزه ناوبری نظامی است. البته این مفهوم در حوزه‌های دیگر نیز مطرح است ولی ممکن است بانام‌های دیگری به آن اشاره شود.

اندسلی در سال ۱۹۸۸[۶] آگاهی وضعیتی را این گونه تعریف کرد:

«درک[2] اجزای محیط داخل یک بازه زمانی و مکانی، فهم[3] معنای آن‌ها و تجسم[4] حالت آن‌ها در آینده نزدیک»

در واقع می‌توان گفت هدف آگاهی وضعیتی رسیدن به این واقعیت است که چه چیز در اطراف ما در حال اتفاق افتادن است و این اطلاعات چه معنایی و مقصودی در حال و آینده خواهند داشت[۵].

اجزای آگاهی وضعیتی با توجه به دامنه‌های مختلف می‌توانند تغییر کنند. اهمیت آگاهی

[1] Situation Awareness
[2] Perception
[3] Comprehension
[4] Projection

وضعیتی به این است که عملاً به بنیان های تصمیم سازی می‌پردازد و رسیدن به کارایی بیشتر را ممکن می‌سازد. این مفهوم، ماورای حوزه‌های دانشی است. هر حوزه‌ای که در آن نیاز به تصمیم سازی و عمل کارا باشد، به آگاهی وضعیتی نیز نیاز است[۵]. طبق تعریف آگاهی وضعیتی، این مفهوم سه سطح دارد:

سطح ۱- درک اجزای محیط

سطح ۲- فهم وضعیت جاری

سطح ۳- تجسم حالت آینده

در ادامه به شرح بیشتر این سه سطح خواهیم پرداخت.

۲-۳-۱- درک

اولین گام رسیدن به آگاهی وضعیتی، درک حالت، خصوصیت‌ها و پویایی‌های اجزای مرتبط در محیط است. این اجزاء بسته به هر موضوع و حوزه مختلف می‌تواند متفاوت باشد. یک خلبان نیاز دارد که اجزای مهمی مثل هواپیماهای دیگر، عوارض مسیر، حالت سیستم و چراغ‌های هشدار و مشخصات آن‌ها را بداند. یک افسر ارتش نیاز دارد تا اجزائی مثل دشمن، مکان‌های غیر نظامی و خودی‌ها و اعمال آن‌ها، ویژگی‌های عوارض، موانع و وضعیت آب‌وهوا را مطلع باشد.

۲-۳-۲- فهم

گام دوم در رسیدن به یک آگاهی وضعیتی مناسب، فهم این است که داده ها و سرنخ‌های دریافت شده در ارتباط با اهداف و مقاصد ما چه معنایی می‌توانند داشته باشند. فهم از یک تلفیق اجزای سطح ۱ و مقایسه آن اطلاعات با اهداف به دست می آید. در این گام قطعات داده یکپارچه شده تا اطلاعات را تشکیل دهند. سپس این اطلاعات ترکیب شده بر حسب اهمیت و معنایی که آن را با اهداف در حال حاضر مرتبط می‌کند اولویت‌بندی می‌شوند. سطح دوم آگاهی وضعیتی برای یک فرمانده جنگی مثل این می ماند که فرماندهان جنگی با دیدن خطوط و ردهای برجامانده خودروها و زرهی‌ها بر روی زمین پی به تعداد نیروها و تجهیزات جنگی آنها ببرد.

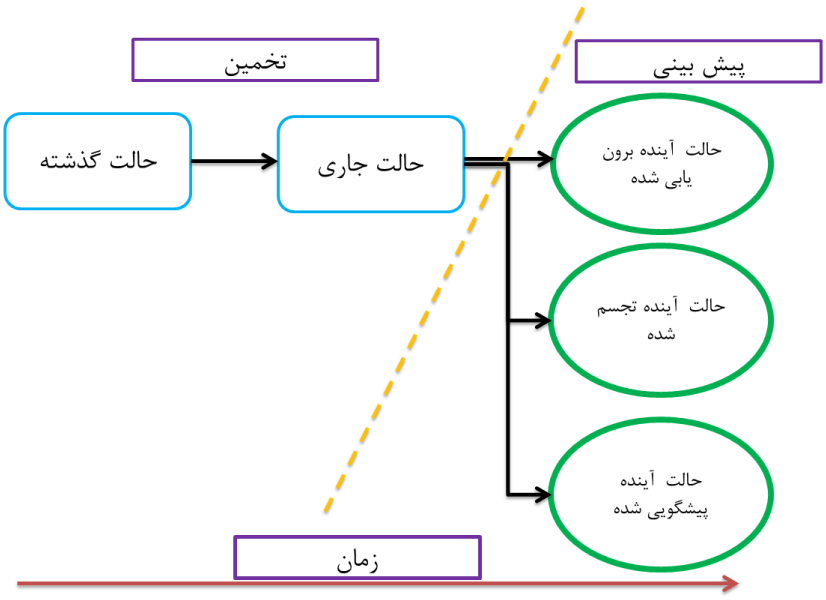
۲-۳-۳- تجسم

به محض اینکه شخص دانست که چه اجزائی وجود دارند و آن‌ها متناسب با اهداف جاری چه معنایی می‌دهند، گام بعد توانایی پیش‌بینی این است که آن اجزاء در آینده (حداقل در یک زمان کوتاه) چه انجام خواهند داد. استفاده از فهم وضعیت جاری برای تشکیل تجسم، نیازمند یک فهم بسیار خوب از آن حوزه است. داشتن این قابلیت به کارشناسان این اجازه را می‌دهد که بتوانند در مقابل رویدادهای مختلف آماده باشند و استراتژی ها و پاسخ‌های مناسبی را برای مدیریت آن‌ها در نظر بگیرند.

۲-۴- تجسم

برای توضیح بهتر تجسم، ما از مفهومی به نام تحلیل هوشمندانه[1] که فرایند مشابهی با اگاهی وضعیتی دارد استفاده می‌کنیم. طبق تعریف کلارک[۷]، تحلیل هوشمندانه، فرآیند پیچیده‌ای است که در آن حقایق، توسعه داده‌شده و به استنتاج منجر می‌شود. در این نوع تحلیل در حقیقت با توسعه اطلاعات موجود، معنایی به دست می آید که می‌تواند مورد استفاده تصمیم سازان[2] و برنامه ریزان[3] قرار گیرد. تحلیل هوشمندانه، بر توصیف حالت یک موجودیت و هر تغییری که حالت یک شی در طی زمان به خود می‌بینند تمرکز دارد. دو فرآیند در تحلیل هوشمندانه برای ارزیابی تغییرات وضعیت در یک موجودیت وجود دارد. این دو فرآیند عبارتند از: تخمین و پیش‌بینی. شکل ۲-۱ مفاهیم تخمین و پیش‌بینی را نشان می‌دهد.

هدف تخمین رسیدن به نتایجی درباره داده، خصوصا برای حالت جاری یا گذشته یک موجودیت است. درحالی‌که هدف پیش‌بینی رسیدن به نتایجی درباره حالت آینده یک موجودیت است.



شکل ۲-۱: نمایشی از تخمین و پیش‌بینی[۷]

[1] Intelligence Analysis
[2] Decision Maker
[3] Planner

کلارک در [۷] برای توضیح بیشتر مفهوم پیش‌بینی از متدولوژی کالمن فیلتر[1] و گام هایی که در متدولوژی کالمن فیلتر استفاده می‌شود بهره می‌گیرد. همانطور که درشکل ۲-۲ دیده می‌شود، گام یک این متدولوژی به تخمین حداقل یک حالت جاری و گذشته یک موجودیت که تحت مشاهده است می‌پردازد. گام دوم به تعیین نیروهایی که بر موجودیت تاثیر دارند و آن را به حالت جاری رسانده اند می‌پردازد. همچنین معیاری که برای تعیین این نیروها استفاده می‌شود نیز مشخص می‌شود. در گام سوم نیز به پیش‌بینی می‌پردازد. کلارک انواع روشهای پیش‌بینی را در سه دسته قرار می‌دهد:

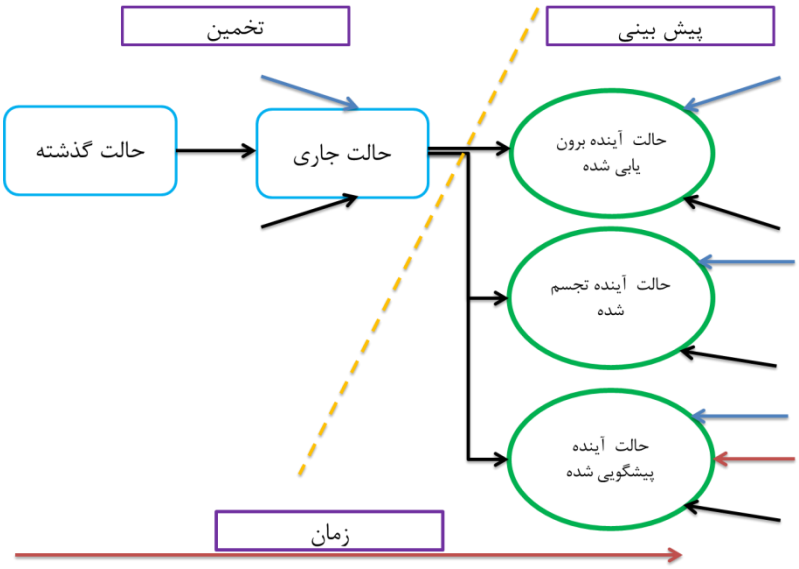
- برون‌یابی[2]
- تجسم
- پیشگویی[3]

در برون‌یابی، نیروهای مشابه با یک روش ثابت بر موجودیت تاثیر می‌گذارند. در حقیقت هر هنگام که یک سطح بالایی از اطمینان[4] و یک عدم قطعیت[5] کم در مورد نیروهای تأثیرگذار بر موجودیت وجود داشته باشد، ازاین روش استفاده می‌شود.

در تجسم هم اگر نیروهای مشابهی را در نظر بگیریم که بر موجودیت تاثیر گذارند این بار برخلاف برون‌یابی تغییرات محتمل در این نیروها در آینده نیز در پیش‌بینی در نظر گرفته می‌شود. به عبارت دیگر در این روش نیروها همیشه ثابت فرض نمی‌شوند و نقش متغیرها و عوامل دیگر نیز در پیش‌بینی در نظر گرفته می‌شود. ویژگی مثبت این روش این است که می‌تواند در یک بازه زمانی طولانی‌تری نسبت به برون‌یابی مورد استفاده قرار گیرد و می‌تواند متغیرها و عامل‌های بیشتری را در تعیین نتیجه محتمل دخالت دهد.

و در نهایت روش پیشگویی ویژگی‌اش این است که به تعیین نیروهای جدیدی که ممکن است بر موجودیت تاثیر بگذارد می‌پردازد. در حقیقت این روش یک طیف گسترده‌تری از احتمالات تأثیرگذار بر یک موجودیت را در نظر می‌گیرد و به تعیین نیروهایی که ممکن است در حالت آینده یک موجودیت اثرگذار باشند نیز می‌پردازد.

[1] Kalman Filter Methodology
[2] Extrapolation
[3] Forecast
[4] Confidence
[5] Uncertainty



شکل ۲-۲: متدولوژی کالمن فیلتر [۷]

۲-۵- کارهای مرتبط در حوزه پیش‌بینی حملات

به‌طورکلی روش‌های مرسومی که در پیش‌بینی حملات استفاده می‌شود را می‌توان به چهار دسته تقسیم کرد. این روش‌ها عبارت‌اند از استخراج الگوهایی برای سلسله اعمال دشمن (ازجمله گراف‌های حمله، الگوهای حمله و ...)، فن‌های یادگیری ماشین و داده‌کاوی، تئوری بازی‌ها و زنجیره‌های مارکوف. البته ممکن است از ترکیبی از این دسته‌ها نیز استفاده شود.

در حوزه سایبر کارهای زیادی بر روی ارزیابی دنباله‌ای از آسیب‌پذیری‌های سیستم و گام‌هایی که می‌تواند به‌منظور یک حمله استفاده شود انجام شده است. لی و همکاران [۸] در سال ۲۰۰۷ یک الگوریتم همبسته سازی مبتنی بر آمار برای پیش‌بینی استراتژی‌های حمله‌ای جدید ارائه کرده‌اند. این روش از دو مکانیسم همبسته سازی هشدار مکمل به نام‌های همبسته سازی مبتنی بر استنتاج بیزین و همبسته سازی مبتنی بر GCT در کارش بهره می‌گیرد. سناریوهای حمله با استفاده از این گراف‌های همبسته سازی شده ایجاد می‌شوند. البته این الگوریتم نمی‌تواند به دلیل نتایج به همراه مثبت ناصحیح بالا برای پیش‌بینی حملات ترکیبی استفاده شود.

وو و همکاران[۹] در سال ۲۰۱۲ یک مدل پیش‌بینی حملات سایبری مبتنی بر شبکه بیزین پیشنهاد کرد. در این مدل علاوه‌بر آسیب‌پذیری‌های در شبکه که به‌وسیله گراف حمله در نظر گرفته می‌شوند، سه عامل محیطی وضعیت شبکه، ارزش دارایی‌های شبکه و تاریخ حمله شبکه نیز لحاظ شده است. بعد از ترکیب این عامل‌ها با گراف حمله، الگوریتم احتمالی بیزین میزان احتمال حمله هر گراف را تعیین می‌کند.

لی و همکاران [۱۰] در سال ۲۰۰۷ پیشنهاد کرده‌اند که یک مدل طرح حمله با استفاده از شبکه بیزین طراحی شود تا اینکه اعمال حمله شناسایی شوند. یک مسئله در این روش‌ها

پیچیدگی توسعه و نگهداری مدل‌های مسیر حمله است که دارای روش‌های حمله، نیت‌ها و تنظیمات شبکه‌ای گوناگون می‌شوند.

چی‌ین و همکاران [۱۱] در سال ۲۰۱۲ چارچوبی برای پیش‌بینی تهدیدات با استفاده از استخراج الگوهای خاص حمله ارائه می‌کنند. در این کار یک متدولوژی فرمال نیز برای تولید طرح‌های حمله شبکه‌ای خاص نیز ارائه می‌شود.

هالسوپل و همکاران [۱۲] در سال ۲۰۰۶ نیز مدلی برای پیش‌بینی حملات ارائه کردند. پیشنهاد این افراد برای ساده‌سازی کار جداسازی مدل‌سازی تنظیمات شبکه و روش‌های حمله‌ای سایبری است. در این کار نیز از گراف‌های حمله استفاده می‌شود. ارزیابی‌ها در هر یک به‌صورت جداگانه انجام شده و نتایج درنهایت با همدیگر ترکیب شده و هدف محتمل برای حمله آینده شناسایی می‌شود. روش های توسعه الگوهای سلسله اعمال دشمن که بر رفتارهای گذشته حمله‌کننده تمرکز دارند زمانی خوب کار می‌کنند که الگوها به‌درستی تعریف شده باشند و گاهی اوقات نیاز به یادگیری خودکار هم در این روش‌ها وجود دارد.

دسته‌ای دیگر از روش‌ها بر استفاده از قواعد یادگیری ماشین و داده‌کاوی برای پیش‌بینی رفتارهای آینده تمرکز کرده‌اند. این روش ها بر استخراج اطلاعات مفید و الگوها از مجموعه داده‌های زیاد تکیه‌دارند و دارای پیچیدگی محاسباتی بالایی نیز هستند. کیپریانو و همکاران [۱۳]در سال ۲۰۱۱ یک روش برای پیش‌بینی رفتارهای مهاجمین ارائه کردند. در این روش از قواعد یادگیری ماشین برای یادگیری رفتارهای گذشته مهاجمین استفاده کرده و از این اطلاعات برای تخمین رفتارهای آینده مهاجمین استفاده می‌کنند.

ژی تانگ لی در سال ۲۰۰۷[۱۴] از روش‌های داده‌کاوی برای تولید گراف‌های داده استفاده کردند. این الگوریتم برای هر دنباله حمله یک درجه پیش‌بینی محاسبه می‌کند. این درجه‌بندی‌ها درنهایت به پیش‌بینی محتمل‌ترین نتیجه آینده یاری می‌رساند.

تئوری بازی‌ها نیز از روش‌هایی است که می‌تواند به پیش‌بینی رفتارهای آینده مهاجم یاری برساند. لیو و همکاران [۱۵] در سال ۲۰۰۵ یک مدل تئوری بازی مبتنی بر تشویق را برای استنتاج نیت، اهداف و استراتژی‌های مهاجم (AIOS) ارائه کرده‌اند. این مقاله این نکته را بیان می کند که انتخاب بهترین مدل بازی، وابسته به درجه دقت سیستم تشخیص نفوذ به کار گرفته‌شده و درجه همبسته سازی گام‌های حمله است. این کار یک متدولوژی برای مدل کردن تعاملات بین یک مهاجم DDoS و مدیر شبکه ارائه کرده‌اند.

تانگ و همکاران [۱۶] در سال ۲۰۱۱ الگوریتمی برای آگاهی وضعیتی تهدید نفوذکننده ارائه کرده‌اند. این الگویتم مبتنی بر تئوری بازی و ادغام اطلاعات است. در این مقاله از یک ساختار شبکه بیزین پویا و استنتاج دقیق برای کسب و ادغام اطلاعات رفتاری مربوط به نفوذ گر استفاده می‌کنند. سرانجام پیش‌بینی رفتار آینده نفوذ گر را با استفاده از محاسبات موازنه پاسخ کوانتومی انجام می‌شود.

درنهایت دسته چهارم روش‌ها بر استفاده از زنجیره‌های مارکوف تأکیددارند. گاوو و همکاران [۱۷] در سال ۲۰۰۳ از مدل‌های مارکوف مخفی برای پیش‌بینی حملات در لایه کاربرد استفاده کرده‌اند.

فاوا و همکاران[۱۸] در سال ۲۰۰۸یک سیستم پیش‌بینی و یادگیری قابل انطباق با استفاده از مدل مارکوف با طول متغیر (VLMM)پیشنهاد کردند. درحالی‌که رویکردهای مشابه یادگیری ماشین و مدل‌سازی برای تشخیص ناهنجاری و تشخیص نفوذ استفاده شده بود، (برای مثال لی و همکارانش[۱۹] در سال ۱۹۹۷، لین و برودلی [۲۰]درسال ۱۹۹۹، یی و همکارانش[۲۱] در سال ۲۰۰۴) در کار دنیل و همکارانش[۱۸] در سال ۲۰۰۸ نخستین بار از فن VLMM برای تجسم حملات استفاده شد. پیش‌بینی VLMM یکی از شاخه‌های پیش‌بینی کننده‌ی جهانی (جاکوئت و همکارانش[۲۲] در سال ۲۰۰۲، شالیزی و شالیزی [۲۳]در سال ۲۰۰۴) است که در اصل برای کاربردهای دیگر نظیر فشرده‌سازی متن (بل و همکارانش [۲۴]) توسعه داده شد. VLMMبه خوبی دستگاه‌های یادگیرنده آنلاین سازگار است، زیرا کارایی محاسباتی بسیار خوبی در مقایسه با مدل مارکوف مخفی و انعطاف‌پذیری بالایی در مقایسه با مدل مارکوف با طول ثابت دارد.

مان و همکاران [۲۵] در سال ۲۰۱۰ از ARMA (که یک روش پیش‌بینی سری زمانی است) و مدل مارکوف برای پیش‌بینی وضعیت‌های امنیتی شبکه استفاده کرد. نتایج پیش‌بینی از دو مدل با مقادیر وزنی مناسبی باهم ترکیب می‌شوند تا پیش‌بینی دقیق‌تری به دست آید.

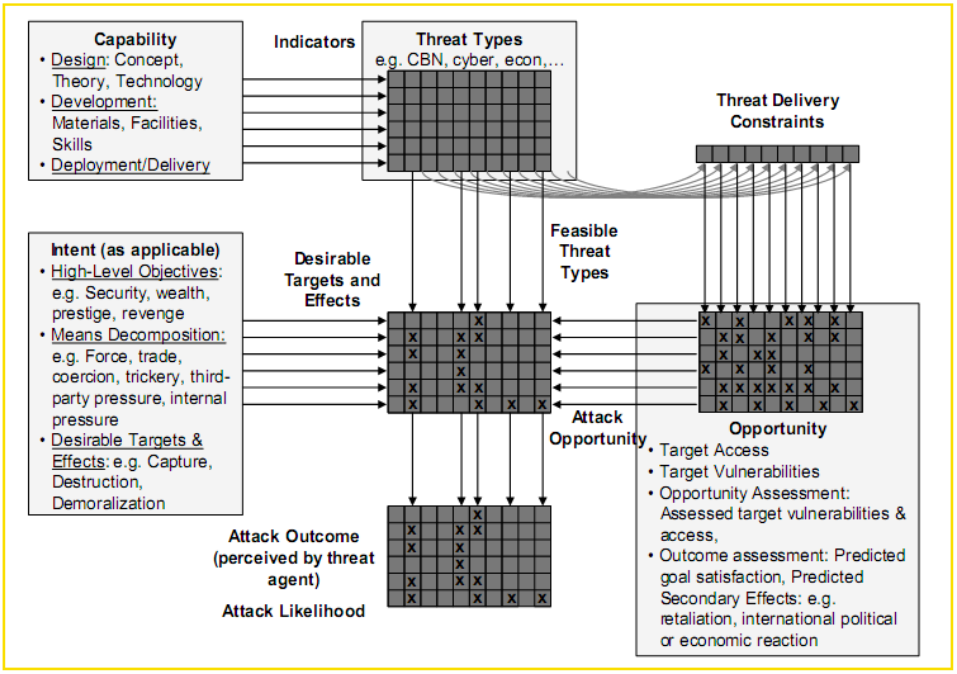
۲-۶- انواع تجسم

همانطور که پیش تر نیز اشاره شد، عنصری که در مدل آگاهی وضعیتی، تجسم می‌شود، وضعیت است. از طرفی نیز می‌توان چندین وضعیت را در یک محیط تصور کرد[۲۶]. وضعیت مورد نظر و مهم در حوزه امنیت سایبر، وضعیت تهدید[1] است. وضعیت تهدید، وضعیتی است که در آن احتمال انواع معینی از رویدادهای بالقوه(یعنی حملات)، به وسیله یک یا چند عامل (که ممکن است یک شخص یا عامل انسانی و یا ماشینی باشند) بر علیه موجودیت‌هایی آسیب‌پذیر(که شامل انسان یا دارایی هایشان است) انجام می‌شود.

رویدادهای تهدید[2] ممکن است عمدی یا غیر عمدی (طوفان‌ها یا خطاهای انسانی) ایجاد شده باشند. در این پایان‌نامه تمرکز بر رویدادهای عمدی است[۲۷].

بر مبنای هستی‌شناسی[3] ارائه‌شده توسط لیتل و همکاران[۲۸]، نشانگرهای وضعیت‌های تهدید با قابلیت، فرصت و (جایی که عامل‌های هدفمند وجود داشته باشد) نیت عامل‌های مهاجم رابطه مستقیم دارند. این مدل درشکل ۲-۳ مشاهده می‌شود.

[1] Threat Situation
[2] Threat Events
[3] Ontology



شکل ۲-۳: اجزاء یک وضعیت تهدید[۲۸]

به عبارت دقیق‌تر یک تهدید (رفتار تهدیدکننده) از سه جزء پایه ای تشکیل شده است: قابلیت، فرصت و نیت [۲۹].

قابلیت یعنی توانایی و ظرفیت برای حمله به یک هدف و ایجاد آثار مخربانه[۳۰]. تشخیص قابلیت یک عامل، به شناسایی توانایی آن عامل در طراحی، توسعه و استقرار[۱] یا تحویل[۲] منابع(تجهیزات) استفاده شده در آن فعالیت وابسته است[۲۷]. در حقیقت در قابلیت در پی فهمیدن میزان دانش، مهارت و تجهیزات فرد مهاجم هستیم.

فرصت مهاجم به این اشاره دارد که چقدر احتمال دارد که مهاجم بتواند به هدف برای انجام یک حمله نزدیک شود[۳۱]. در فرصت تمرکز بر محدودیت‌های مهاجم است. یکی از جنبه‌های فرصت، نوع تهدید است. هر نوع تهدید خودش محدودیت‌هایی را برای مهاجم به وجود می‌آورد(مثلا تنها با ابزار پویش نمی‌توان نفوذ انجام داد). بقیه فرصت‌های مهاجم از طریق بررسی آسیب‌پذیری دارایی‌های هدف، میزان دسترس‌پذیری هدف و ارزیابی خود مهاجم از طریق فرصت‌ها و نتایج قابل دست‌یابی است[۲۷].

تمایل[۳] یا انگیزه[۴] یک دشمن برای حمله به یک هدف به منظور ایجاد اثرات خرابکارانه را نیت

[1] Deploy
[2] Deliver
[3] Desire
[4] Motivation

گویند[۳۰]. نیت می‌تواند به وسیله جداسازی[1] مقاصد(مثل فریبکاری، کاسب‌کاری و..)، اهداف سطح بالا(ثروت،انتقام و...) و هدف‌ها و تأثیرات مورد علاقه(مانند دستگیر کردن، تخریب کردن، تضعیف روحیه و...) به دست آید[۲۸].

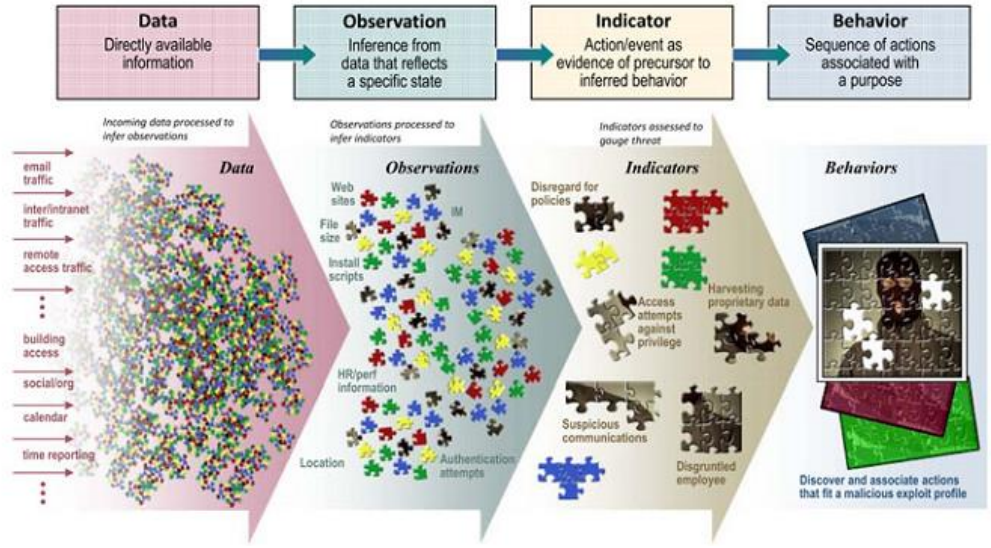
۲-۷- رفتار چیست؟

همانطور که می‌دانیم تهدید، به‌طور کلی یک نوع رفتار است که می‌تواند از یک زیر رفتار یا دنباله‌ای از زیر رفتارها بوجود آید. به عنوان مثال برای ایجاد یک درب پشتی در یک میزبان، ابتدا می‌بایست رفتارهایی مانند شناسایی[2] و نفوذ انجام شود تا بتوان به امکان ایجاد درب پشتی فراهم شود. هر رفتار از دنباله‌ای از اعمال[3] یا نشان‌گرها[4] تشکیل می‌شود که با توجه به یک هدف باهم مرتبط شده اند[۳۲]. هدف می‌تواند خوب یا بد باشد که به‌تبع آن رفتار خرابکارانه و خوب خواهد بود.

هر نشانگر از پردازش یک یا چند مشاهده[5] به دست می آید. به عنوان مثال با مشاهده ترافیک شبکه و مشاهداتی مثل افزایش بیشتر از حالت عادی دانلودها در ساعت‌های کاری دیر یا غیرعادی، می‌توان نشانگرهایی مانند «فعالیت‌های خرابکارانه محتمل» را فعال کرد.

هر مشاهده نیز از پردازش و کنار هم قرار گرفتن داده‌ها به دست می آید. به عنوان مثال می‌توان با محاسبه مقدار ترافیک در بازه زمانی مشخص به نرخ دانلود به آپلود در زمان‌های مشخص پی برد. این نتایج می‌تواند مشاهداتی مانند «میزان وب گردی در اینترنت در ساعات مختلف» را به دست آورد. یا مثلا مشاهده‌ای مانند «اسکرین سیور۶ غیرفعال شد» می‌تواند از داده ای مثل مقدار کلید رجیستری اسکرین سیور در سیستم شخص به دست آید. رابطه بین این اجزاء (داده، مشاهده، نشانگر و رفتار) در شکل ۳-۴ مشاهده می‌شود.

[1] Decomposition
[2] Reconnaissance
[3] Actions
[4] Indicators
[5] Observation
[6] Screen Saver



شکل ۲-۴: رابطه میان داده، مشاهده‌ها، نشانگرها و رفتارها[۳۲]

۲-۸- کارهای مرتبط در حوزه مدل‌سازی رفتاری مهاجمین

همانطور که پیش تر دیدیم، هر رفتار را از دو جنبه می توان بررسی کرد. یک جنبه از دیدگاه خود عامل است. این دیدگاه بیان می کند یک رفتار زمانی بروز می کند که سه جزء قابلیت، فرصت و نیت حاضر باشند. جنبه دیگر از دیدگاه خود عمل است. یعنی چه رفتار هایی پشت سر هم قرار گرفته اند تا یک رفتار نهایی رخ دهد. بر همین اساس می‌توان کارهای انجام‌شده در حوزه مدل‌سازی رفتاری مهاجمین را به دو دسته کلی تقسیم نمود. مدلهایی عامل محور و مدلهای عمل محور. از دسته اول این مدل ها، می‌توان به مدل زیر اشاره کرد.

هالد و پترسون در سال ۲۰۱۲[۳۳] طبقه‌بندی از گروه‌های هکرها شامل نه طبقه اصلی مبتنی بر انگیزه‌ها و توانایی هایشان ارائه کردند. این طبقات عبارتند از:

۱- نوآموز[1]
۲- ولگردهای سایبری[2]
۳- داخلی‌ها[3]
۴- دزدان حقیر[4]
۵- ویروس نویسان[5]
۶- هکرهای نگهبان سنتی[6]

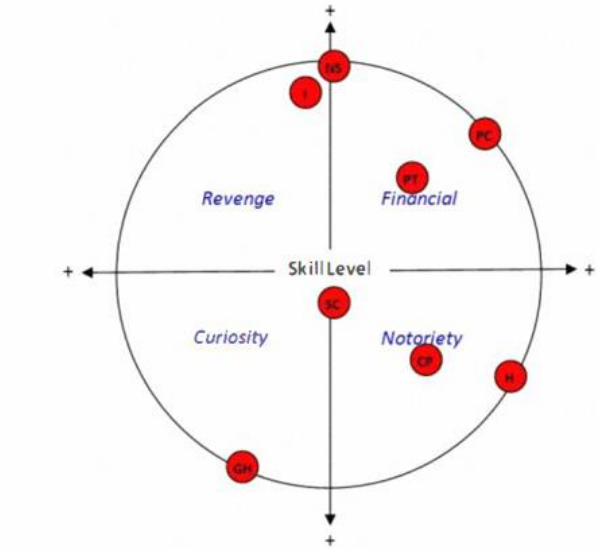
[1] Novice (NV)
[2] Cyber-Punks (CP)
[3] Internals (IN)
[4] Petty Thieves (PT)
[5] Virus Writers (VW)
[6] Old Guard Hackers (OG)

۷- مجرمان حرفه‌ای[۱]

۸- جنگجویان اطلاعات[۲]

۹- فعالان سیاسی[۳]

این مقاله برای بررسی نیت مهاجمین از چهار خصوصت کنجکاوی[۴]، انتقام[۵]، مالی[۶] و بدنام کردن[۷] استفاده کرد. همچنین هالد و همکاران از ویژگی‌هایی که در مقاله[۳۴] در مورد هر تهدید بیان شده بود برای بررسی رفتار تهدیدکننده هر یک از این مهاجمین استفاده کردند. مقاله [۳۴] برای هر تهدید، شش جنبه را معرفی کرده بود. این جنبه‌ها عبارتند از: نوع، نیت، محرک‌ها، قابلیت، روش ها و تمایلات. در نهایت نمودار و نتایج نهایی آن‌ها را درشکل ۲-۵ می‌بینید.



شکل ۲-۵: نمودار سطح مهارت/انگیزه[۳۳]

[1] Professional Criminals (PC)
[2] Information Warriors (IW)
[3] Political Activists (PA)
[4] Curiosity
[5] Revenge
[6] Financial
[7] Notoriety

دسته دوم این مقالات همانطور که گفته شد بر خود زیر رفتارها و رابطه میان آنها تمرکز دارند. چی ین و همکاران در سال ۲۰۰۷ [۳۵] در کارشان یک هستی‌شناسی حمله‌ای شبکه‌ای را پیشنهاد می‌کنند. طبقه‌بندی این هستی‌شناسی، مبتنی بر زاویه دید[۱] مهاجم است. بر طبق این هستی‌شناسی، یک حمله شبکه‌ای می‌تواند به داخل کلاسهای شناسایی، نفوذ[۲] و فعالیت بدون مجوز طبقه‌بندی شود. کلاس شناسایی به گام هایی از حمله اشاره دارد که مهاجم برای جمع‌آوری اطلاعات درباره اهدافش (برای نمونه اطلاعات توپولوژی شبکه) استفاده می‌کند. کلاس نفوذ به گام هایی از حمله اشاره دارد که در آن مهاجم از آسیب‌پذیری‌های محتمل در اهداف، برای بالابردن[۳] اولویتش به صورت غیرقانونی بهره‌کشی می‌کند تا یکی از موارد محرمانگی، صحت و یا دسترس‌پذیری آن سیستم ها را نابود کند. کلاس آخر یعنی فعالیت بدون مجوز اشاره به فعالیت‌هایی دارد که به وسیله مهاجم بر هدف نفوذ شده تحمیل می‌شود. مواردی مانند بازنویسی فایل های سیستمی یا نصب عامل‌های خرابکار روی اهداف. این مقاله از این هستی‌شناسی حمله، برای یکپارچه‌سازی کارای هشدارهای نامتجانس حسگرها و بازشناسی سناریوهای حمله مهاجمین بهره می‌گیرد.

هاتکینز و همکاران[۳۶] در سال ۲۰۱۳ مدلی به نام زنجیره مرگ نفوذ[۴] را پیشنهاد کردند. این مدل متناسب با حملات تهدید ماندگار پیشرفته (APT)[۵] طراحی شده است. حملاتی که در آن مهاجم بعد از نفوذ و گرفتن مجوز غیر قانونی برای مدت طولانی به صورت ناشناس باقی می‌ماند. هدف اصلی در این نوع حملات دزدی اطلاعات است. همانطور که درشکل ۲-۶ دیده می‌شود این مدل دارای هفت گام است. این گام ها عبارتند از:

۱- شناسایی:

فاز اولیه نفوذ شامل جمع‌آوری داده مرتبط با سیستم یا موجودیت مورد هدف است. این فاز می‌تواند شامل مهندسی اجتماعی و جستجوی شخصیت های کلیدی داخل یک سازمان از طریق روشهای اطلاعاتی منبع آزاد[۶] انجام شود.

۲- مجهز سازی[۷]:

در این گام بدافزارهای سفارشی شاخته شده توسعه داده می شوند و به صورت پنهان به عنوان یک افزونه در یک برنامه مانند مایکروسافت آفیس ذخیره می‌شوند. این گام بعد از آگاهی که از مرحله قبل به دست می آید به منظور دسترسی از راه دور به یک سیستم تحت تسلط

[1] Perspective
[2] Penetration
[3] Escalate
[4] Intrusion kill Chain
[5] Advanced Persistent Threat
[6] Open Source Intelligence Techniques
[7] Weaponization

قرارگرفته[1] انجام می‌شود.

۳- تحویل[2]:

افزونه خرابکار بایست به سیستم مورد هدف منتقل شود تا اینکه زنجیره مرگ ادامه پیدا کند. سه روش معمول روش های تحویل، ایمیل، محتوی وب و یا فایل‌های چندرسانه‌ای است[۳۷].

۴- بهره‌کشی[3]:

به محض اینکه افزونه تحویل شد، افزونه می‌بایست اجرا شود و قابلیت تغییر بر روی سیستم هدف را به مهاجم بدهد. این حالت در حضور آسیب‌پذیری داخل سیستم کاربر ممکن می‌شود. در اکثر این مواقع کاربر در مقابل افزونه‌های خوش‌ساخت که به اسنادی به ظاهر مهم متصل می‌شوند تسلیم خواهند شد.

۵- نصب[4]:

موجودیت خرابکار یک اتصال ماندگار[5] داخل سیستم تحت تسلط قرارگرفته ایجاد می‌کند. به محض اینکه سیستم بهره‌کشی شد و افزونه نصب شد، سیستم هدف تغییر داده می‌شود تا فرمان‌های از راه دور را دریافت کند و اعمال را از طرف مهاجم انجام دهد.

۶- فرماندهی و کنترل[6]:

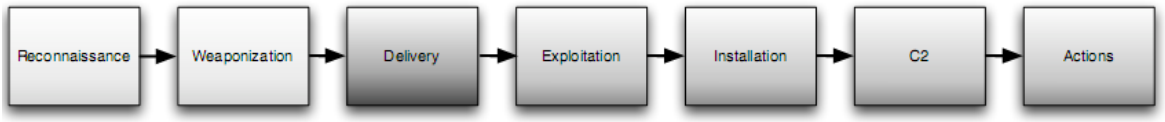
بدافزار طراحی‌شده برای کسب سود مالی اغلب به طور خودکار عمل می‌کنند و داده را به صورت خودکار به یک میزبان جمع‌آوری‌کننده می‌فرستد. این نوع از اتصال خیلی پرحرف است و به‌آسانی در طی زمان کوتاهی شناسایی می‌شود. اغلب تهدیدات ماندگار کانال های کنترلی را از طریق تعاملات دستی بر قرار می‌کنند. به این شکل که سیستم منتظر دریافت ترافیک کنترلی می ماند به‌جای اینکه یک پیغام بیکن[7] را در فواصل زمانی منظمی بفرستد. این شناسایی که یک تحت تسلط قرار گرفتن را کاهش می‌دهد و یک تفاوت کلیدی این نوع تهدیدات با تهدیدات دیگر است.

۷- اعمال مورد هدف[8]:

این فاز بعدازاینکه مهاجم بر سیستم تحت تسلط قرارگرفته کنترل پیدا می‌کند انجام می‌شود. در این گام استخراج داده از سیستم قربانی انجام می‌شود.

---

[1] Compromised
[2] Delivery
[3] Exploitation
[4] Installation
[5] Persistent
[6] Command and Control (C2)
[7] Beacon
[8] Actions on the Objective



شکل ۲-۶: فازهای زنجیره مرگ[۳۶]

در سال ۲۰۱۳ شرکت ماندیانت[۳۷] مدلی به نام چرخه زندگی حمله APT را پیشنهاد کرد. این مدل شش گام دارد که در آن یک حلقه در گام پنجم وجود دارد که احتمال بازگشت را در نظر می‌گیرد. شکل ۲-۷ این مدل را نشان می‌دهد.

این گام ها عبارتند از:

۱- شناسایی اولیه:

مشابه با گام اولیه در زنجیره مرگ هاتکینز است.

۲- تحت تسلط قراردادن اولیه:

در این گام، گام های دو تا پنج مدل پیشین یعنی مجهز سازی، تحویل، بهره‌کشی و نصب تلفیق‌شده‌اند.

۳- برقراری جای پا[۱]:

مشابه با گام شش مدل پیشین یعنی فرماندهی و کنترل.

۴- بالا بردن اولویت‌ها[۲]:

اولین تمایز بین مدل پیشین و این مدل این گام است. این مقاله برای این گام تعدادی از ابزارها را معرفی می‌کند که برای بالابردن اولویت استفاده می‌شود و می‌توانند نشانه هایی از تحت تسلط قرار گرفتن و فعالیت تهدید ماندگار باشد. در ادامه این فاز APT ممکن است به تزریق به داخل سیستم ها به وسیله ادامه دادن به گام پنج و زیر فازهایش ادامه دهد و یا ممکن است این فرآیند را رد کرده و به گام ششم منتج شود.

۵- شناسایی اولیه (انتخابی):

در این گام شناسایی در شبکه برای مشخص کردن امکان تزریق‌های آینده انجام می‌شود. برخلاف گام شناسایی اولیه این مرحله عمدتا مبتنی بر شبکه است و ممکن است همراه با پویش شبکه داخلی نیز همراه شود. این فاز اغلب شامل انگشت‌نگاری[۳] است که در آن مهاجم یک بچ اسکریپت[۴] را برای گرفتن اطلاعاتی درباره اعضای گروه کاربری اجرا می‌کند.

الف- حرکت جانبی[۵]: این گام شامل اکتشاف[۶] دیگر منابع شبکه مثل فایل های اشتراکی یا

[1] Establish Foothold
[2] Escalate Privileges
[3] Fingerprinting
[4] Batch Script
[5] Lateral Movement
[6] Exploration

دیگر سیستم های آسیب‌پذیر شبکه است.

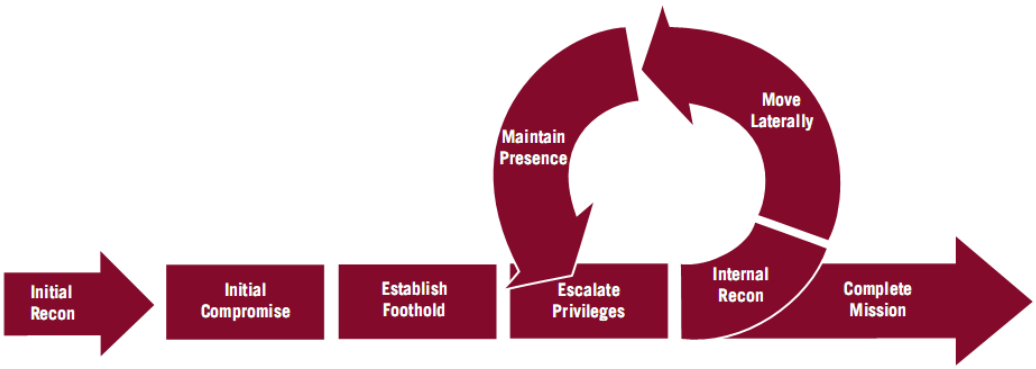
ب- نگهداری حضور[1]: مهاجم وسایل اضافی را برای دسترسی به شبکه مورد هدف برقرار می‌کند. وسایلی مانند نرم افزارهای درب پشتی که روی آن سیستم نصب می‌شود. انتقال داده همچنین با برقراری یک کانال مخفی[2] برای انتقال داده به صورت ماندگار می‌تواند انجام گیرد. کانال هایی که از سوءاستفاده از ارتباط VPN از راه دور موجود در سیستم قربانی انجام می‌شود. راه دیگر، تونل سازی با استفاده از پروتکل‌هایی مثل ICMP، DNS و HTTP برای انتقال داده است.

ج- برگشت به گام چهار: اگر مهاجم نگهداری حضور را از طریق تزریق جانبی انجام داد، بالا بردن اولویت بر روی ماشین‌های برای ایجاد دسترسی پایدار انجام خواهد گرفت. این گام دوباره امکان جمع‌آوری داده مرتبط با تحت تسلط قرار گرفتن را فراهم می‌کند.

۶- تکمیل ماموریت:

داده ای که ممکن است از طریق گام های قبلی داخل چرخه جمع‌آوری شده باشد در گام نهایی آن به وسیله آماده‌سازی داده و انتقال آن به شبکه خارجی تکمیل می‌شود. آماده‌سازی اغلب شامل فشرده‌سازی داده و رمزنگاری پیش از انتقال است. انتقال هم از طریق کانال مخفی که در گام ۵-ب انجام‌شده صورت می‌گیرد.

این مدل به طور کلی دانه‌دانگی[3] رویدادها را نسبت به مدل پیشین بیشتر رعایت می‌کند



شکل ۲-۷: چرخه زندگی حمله APT [37]

در سال ۲۰۱۴، برایانت و همکاران[۳۸] ترکیبی از دو مدل زنجیره مرگ هاتکینز[۳۶] و چرخه زندگی حمله APT ماندیانت[۳۷] را پیشنهاد کردند. مدل برایانت گامهای ابتدائی تحت تسلط قرار دادن تا نصب موفق را با گام های بیشتری نشان داد. مدل ماندیانت یک چارچوب برای تشخیص

[1] Maintain Presence
[2] Covert Channel
[3] Granularity

اعمال تزریق و به‌حساب آوردن اعمال تهدیدکننده داخل شبکه‌ها که با حرکات جانبی و توزیع تزریق به حفظ ماندگاری اقدام می‌کنند می‌پردازد.

مدل ترکیبی پیشنهادی این مقاله درشکل ۲-۸ دیده می‌شود. این مدل سه گام کلی دارد که هر یک در ادامه شرح داده می‌شود:

۱- گام کلی تحت تسلط قرار گرفتن:

این گام خود شامل چهار زیر گام از گام‌های مدل هاتکینز [۳۶] که به آن افزایش اولویت نیز اضافه‌شده است می‌شود. این زیرگام‌ها عبارتند از:

۱- شناسایی: برخلاف گام شناسایی مدل پیشنهادی هاتکینز [۳۶]، این گام شامل اعمالی مانند بررسی شبکه[1]، پویش پورت ها، تلاش های شکست‌خورده احراز هویت[2] و انگشت‌نگاری سیستم عامل و دیگر فنون تست نفوذ در داخل شبکه می‌شود.

۲- تحویل: این گام مشابه با مدل هاتکینز [۳۶] است.

۳- نصب: این گام مشابه با مدل هاتکینز [۳۶] است.

۴- بالا بردن اولویت: در این گام مهاجم در گیر تغییر مجوزهای امنیتی حساب‌ها می‌شود.

۵- فرماندهی و کنترل: این گام از طریق ارتباط با یک آدرس IP خارجی انجام می‌شود.

۲- گام کلی حرکت جانبی/ماندگاری

گام حرکت جانبی بعد از گام های شناسایی داخلی، احراز هویت و فرماندهی کنترل انجام می‌گیرد. جزئیات این گام عبارتست از:

۱- شناسایی داخلی:

این گام انجام می‌گیرد تا بتوان به خصوصیات هویتی مانند آدرس IP، نام میزبان، آدرس های MAC و بقیه موارد را به واسطه سیستم تحت تسلط قرارگرفته از دیگر میزبان‌های داخلی به دست آورد.

۲- احراز هویت:

این گام همان گام حرکت جانبی در مقاله ماندیانت[۳۷] است که به وسیله احراز هویت موفق به یک میزبان داخلی انجام می‌شود.

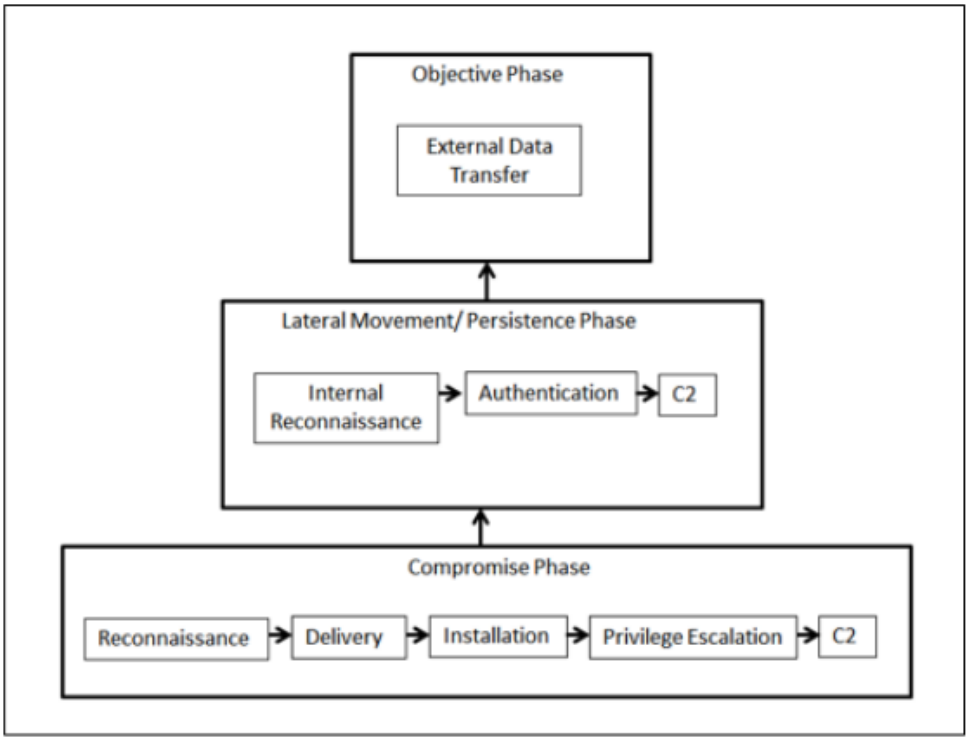
۳-فرماندهی و کنترل:

این گام هم بااتصال با یک آدرس IP خارجی انجام می‌گیرد.

۳-گام کلی هدف:

این گام ترکیب همان گام های نهایی هاتکینز [۳۶] و ماندیانت[۳۷] انجام می‌شود.

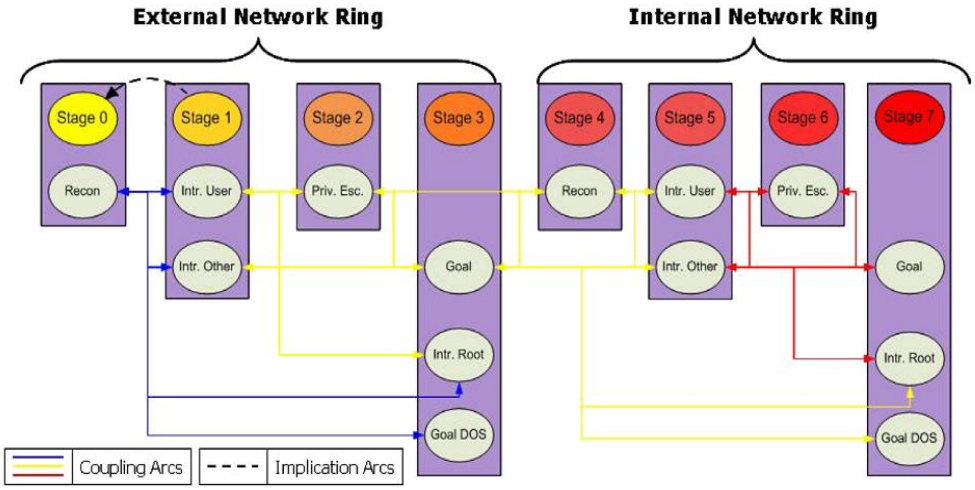
[1] Network Probing
[2] Authentication



شکل ۲-۸: مدل زنجیره مرگ ترکیبی[۳۸]

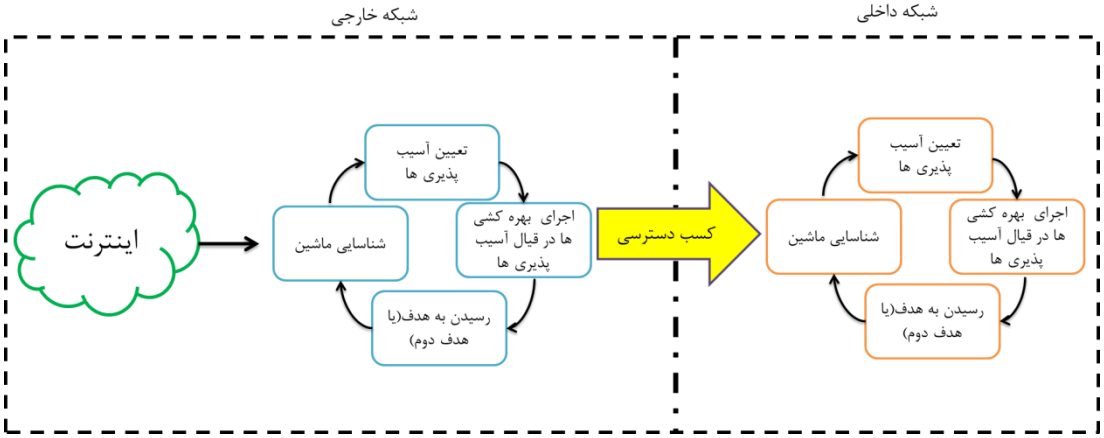
در سال ۲۰۰۹ یانگ و همکاران [۳۹] مدلی به نام الگوی راهنما[1] را پیشنهاد کردند. این مدل که با مدل های هم نوعش تفاوت‌هایی دارد یک گراف جهت‌دار است که هدف آن هدایت تولید ردهای[2] حمله است. این مدل مستقل از شبکه خاص، تنظیمات سیستم ها و نوع حملات مختلف است. این راهکار تا حد زیادی پیچیدگی مدل را کاهش می‌دهد و به همبسته سازی نزدیک به بی‌درنگی و ردیابی حملات کمک می‌کند. این مدل در شکل ۲-۹ دیده می‌شود. مدل الگوی راهنما یک شبکه را به طور پیش فرض در دو لایه بررسی می‌کند (البته به تعداد لایه های موجود در یک شبکه این مدل قابل توسعه است). از بیرون شبکه تا زمانی که یک موجودیت در لبه شبکه داخلی تحت تسلط قرار بگیرد و ازآنجا به داخل شبکه.

[1] Guidance Template
[2] Track



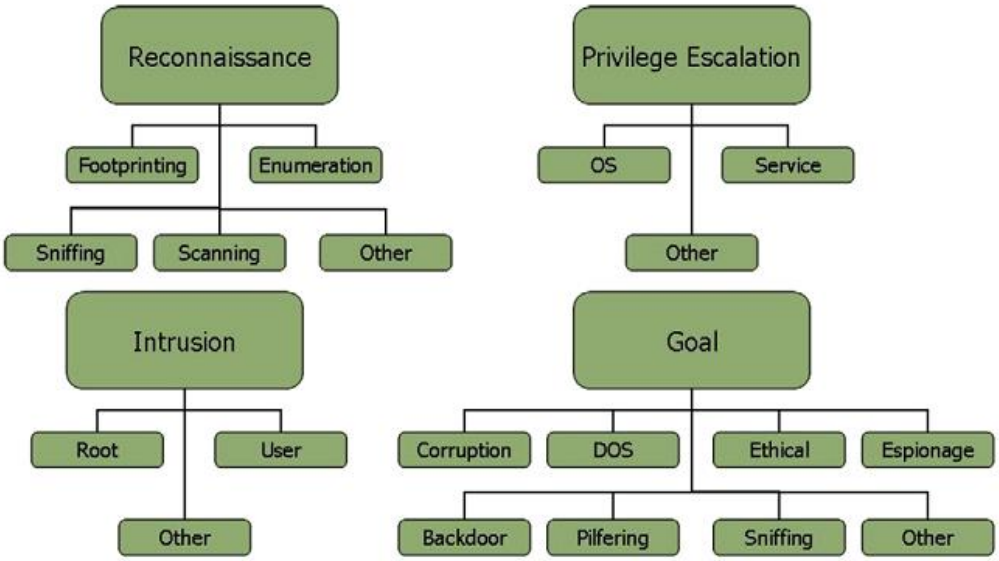
شکل ۲-۹: الگوی راهنمای حمله[۳۹]

تعداد این گام ها در هر لایه حداکثر چهار گام مشخص‌شده است. این گام ها در شکل ۲-۱۰ دیده می شوند. البته این تعداد گام می‌تواند کمتر نیز باشد. تعداد کمتر این گام ها به وسیله یال‌های جهت‌دار در شکل ۲-۹ ممکن خواهد بود.



شکل ۲-۱۰: گام های موجود در الگوی راهنما[۴۰]

اعمالی که در هر گام می‌تواند انجام شود به طور کلی در چهار طبقه کلی قابل قرار گرفتن است. این طبقات عبارت است از شناسایی، بالا بردن اولویت، نفوذ و هدف. هر دسته نیز شامل چندین زیر طبقه است. این طبقات و زیر طبقات در شکل ۲-۱۱ دیده می‌شود.



شکل ۲-۱۱: طبقات اعمال حمله[۳۹]

در جدول ۱ ویژگی مدل های رفتاری مطرح‌شده در این بخش بیان‌شده است. برای بررسی این مدل ها از سه معیار استفاده‌شده است:

- وابستگی به نوع تهدید: آیا این مدل برای یک نوع تهدید خاص طراحی شده است؟
- سطح دانه‌دانگی: چقدر مدل وارد جزئیات شده است؟ اگر فقط در سطح کلی بحث کرده باشد مدل از نوع دانه‌دانگی درشت[1] نام‌گذاری می‌شود و اگر وارد جزئیات هم شده باشد به آن دانه‌دانگی ریز[2] منتسب شده است.
- مقیاس‌پذیری: آیا مدل در زمانی که شبکه مورد نظر شامل چندین شبکه تودرتو باشد، آیا جواب می‌دهد. آیا این حالت بررسی‌شده است؟
- انعطاف‌پذیری: آیا مدل در نبود یک گام یا عنصر تشکیل‌دهنده‌اش می‌تواند جوابگو باشد؟ آیا مدل می‌تواند تمام حالات ممکن و صحیح را پوشش دهد؟
- نوع مدل رفتاری: همانطور که گفتیم آیا از نوع مدل های عمل محور است یا از نوع مدل های عامل محور.

[1] Coarse Granularity
[2] Fine Granularity

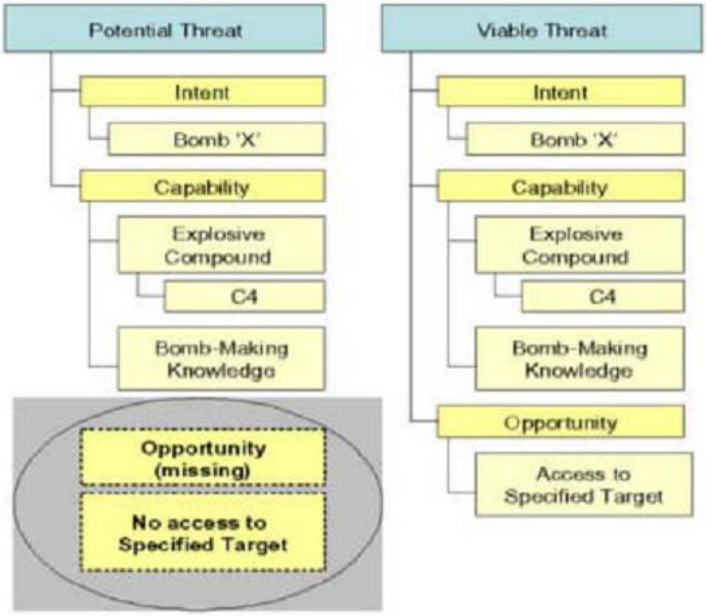
جدول ۱: مقایسه مدلهای رفتاری مطرح‌شده

معیارها / مدلها	وابستگی به نوع تهدید	سطح دانه‌دانگی	مقیاس‌پذیری	انعطاف‌پذیری	نوع مدل رفتاری
هالد و پترسون [۳۳]	ندارد	دانه‌دانگی درشت	دارد	نامشخص	عامل محور
چی ین و همکاران [۳۵]	ندارد	دانه‌دانگی درشت	نامشخص	ندارد	عمل محور
هاتکینز و همکاران[۳۶]	دارد	دانه‌دانگی درشت	نامشخص	ندارد	عمل محور
ماندیانت[۳۷]	دارد	دانه‌دانگی درشت	نامشخص	ندارد	عمل محور
برایانت و همکاران[۳۸]	دارد	دانه‌دانگی درشت	نامشخص	ندارد	عمل محور
یانگ و همکاران [۳۹]	ندارد	دانه‌دانگی ریز	دارد	دارد	عمل محور

۲-۹- رابطه بین انواع تجسم

از بخش‌های پیشین متوجه شدیم که یک تهدید (که خود یک رفتار است) از سه جزء پایه ای قابلیت، فرصت و نیت تشکیل شده است. تهدیدات نیز به دو نوع عملی[۱] و غیر عملی[۲] یا بالقوه[۳] تقسیم می‌شود[۲۹]. تهدیدات عملی هنگامی به وجود می‌آیند که هر سه جزء تشکیل‌دهنده تهدید (یعنی فرصت، قابلیت و نیت) حاضر باشند و با همدیگر در یک روش صحیحی متصل شوند. یک تهدید غیر عملی (یا بالقوه)، تهدیدی است که در آن تنها دو تا از سه جزء حضور دارند. برای مثال همانطور که درشکل ۲-۱۲ دیده می‌شود در تهدید بالقوه، نیت ایجاد بمب و قابلیت ساخت و مواد اولیه بمب مانند C4 که قابلیت انفجار دارد مهیا است ولی فرصتی که بتوان بمب را در مکان هدف قرار داد وجود ندارد.

[1] Viable
[2] Non-Viable
[3] Potential

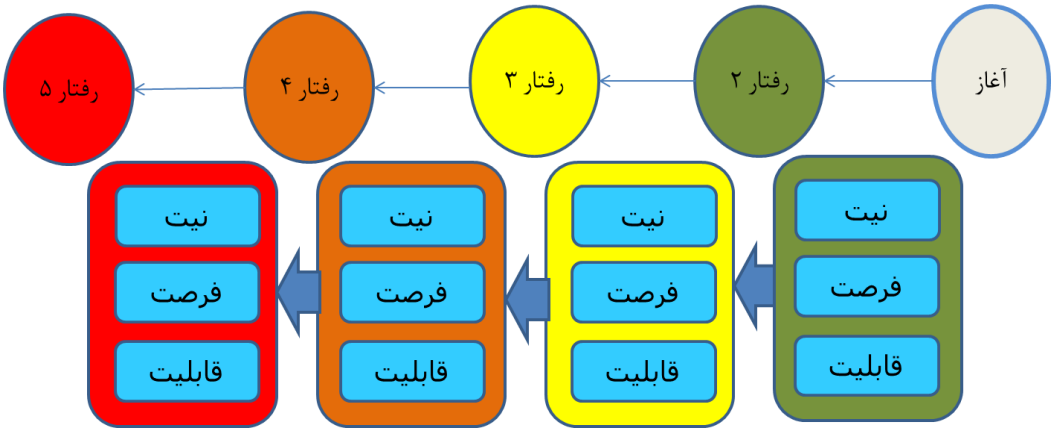


شکل ۲-۱۲: تهدیدات عملی و غیرعملی[۲۸]

هنگامی که به حوزه سایبر و پیچیدگی‌های آن می‌نگریم، می‌بینیم که جنس تهدیدات در حوزه سایبر عموما از جنس تهدیدات بالقوه است. زیرا در اغلب اوقات بررسی یکی از این اجزاء یا ممکن نیست و یا این که با خطای بالایی می‌تواند همراه باشد (مانند بررسی نیت).

توجه به دو نکته می‌تواند در بهبود نتایج و کاهش خطا مفید باشد. اولین نکته این است که با توجه بخش قبل، دیدیم که یک حمله (تهدید یا رفتار نهایی) زمانی به وجود می‌آید که رفتارهایی پیش از آن پشت سرهم قرار گیرند تا آن رفتار بتواند به وجود آید. از طرفی هم دیدیم که بین رفتار و سه جزء ایجادکننده‌اش نیز رابطه‌ای وجود دارد؛ بنابراین این نتیجه را می‌توان گرفت که همانطور که میان رفتارها روابط علت و معلولی وجود دارد، می‌بایست میان سه جزء تشکیل‌دهنده دو رفتار پشت سر هم نیز رابطه‌ای وجود داشته باشد. این روابط در شکل ۲-۱۳ قابل مشاهده است.

نکته دوم این است که همانطور که هر رفتار برای بروزش به سه جزء نیت، فرصت و قابلیت وابسته است رابطه بالعکسی نیز وجود دارد؛ یعنی یک رفتار با ظهورش می‌تواند نشان دهنده وقوع و آماده بودن سه گانه نیت، فرصت و قابلیت مشخصی نیز باشد. این دو ویژگی می‌توانند نقش کمک‌کننده‌ای را در تجسم دقیق‌تر رفتارهای آینده و حل بهتر پیچیدگی های این حوزه ایفا نمایند. در این کار ما تنها به جنبه دنباله‌ای بودن رفتار ها پرداختیم و در واقع تنها تجسم با استفاده از رفتار را مورد بررسی قرار داده ایم.



شکل ۲-۱۳: رابطه بین رفتارها و اجزایشان

۲-۱۰- جمع بندی

در این فصل بیان شد که داشتن تصمیمی کارا و عملی بهینه نیازمند یک آگاهی وضعیتی مناسب است. آگاهی وضعیتی هدفش ایجاد و طراحی واسطهایی متناسب با قابلیت‌ها و نیازمندی‌های کاربران است. همچنین در این فصل سطوح آگاهی وضعیتی معرفی و تعریف شدند. تجسم که یکی از سطوح آگاهی وضعیتی است، نوعی از پیش‌بینی است که در آن به پیش‌بینی وضعیت نیروهای موثر بر یک موجودیت (که می‌توانند میزان اثرشان تغییر کند) پرداخته شده و بدین وسیله محتمل ترین وضعیت خود موجودیت در آینده نیز مشخص می‌شود. می‌پردازیم. در این فصل بیان شد که به طور کلی یک رفتار از دنباله‌ای از نشانگرها یا رفتارهای دیگر تشکیل می‌شود که هدف مشخصی را دنبال می‌کنند. همچنین هر رفتار برای انجام شدن از سمت یک عامل هوشمند نیازمند سه جزء پایه ای فرصت، قابلیت و نیت است؛ بنابراین برای اینکه بتوان یک رفتار را دقیق‌تر تجسم کرد نیاز است که علاوه بر استفاده از ویژگی دنباله‌ای و پی‌درپی بودن رفتارها و تجسم رفتارها، از تجسم با استفاده از قابلیت، نیت و فرصت نیز استفاده نمود. البته در این کار تمرکز تنها بر تجسم با استفاده از رفتار خواهد بود.

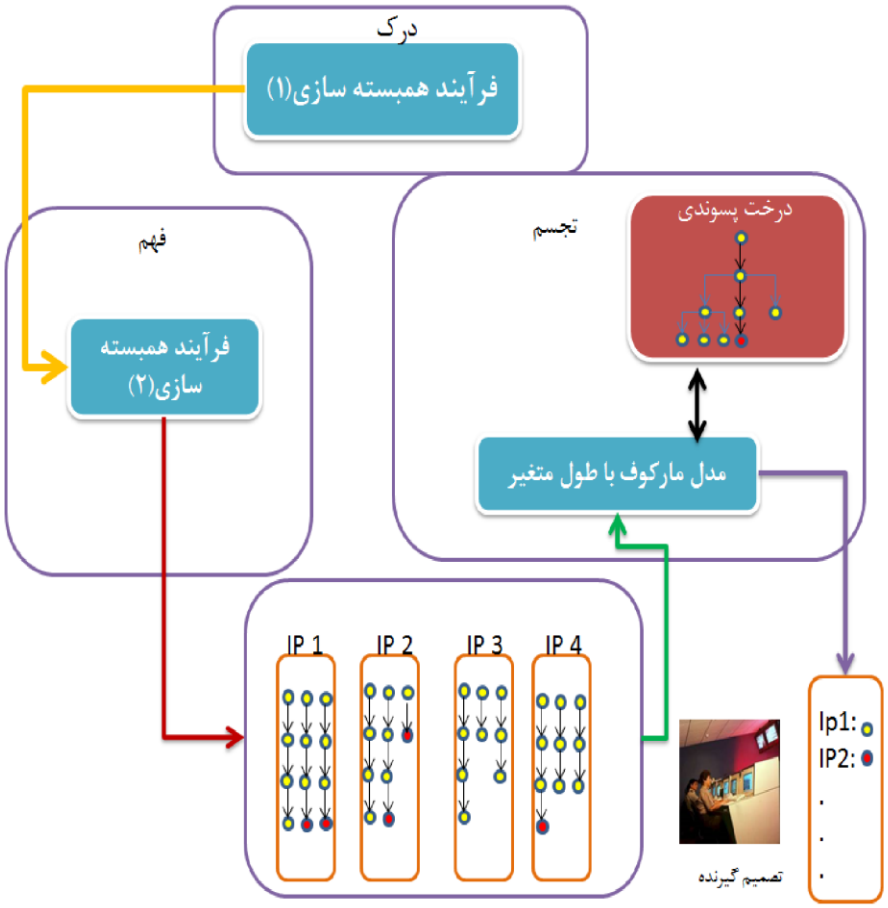
فصل ۳– معماری پیشنهادی

۳–۱–مقدمه

در این فصل معماری کلی از کلیت کار ارائه می‌شود. این معماری مبتنی بر سطوح آگاهی وضعیتی طراحی شده است. همچنین در این فصل فرآیند همبسته سازی که در این کار استفاده‌شده است معرفی می‌شود. الگوی راهنمای پیشنهادی ارائه‌شده و تشریح می‌گردد. در لایه سوم این مدل که لایه تجسم است، به مفاهیم ریاضیاتی الگوریتم های استفاده‌شده در این بخش پرداخته می‌شود.

۳–۲– معماری کلی

معماری پیشنهادی کلی در این گزارش در شکل ۴-۱دیده می‌شود. این معماری از سه سطح اصلی درک، فهم و تجسم تشکیل شده است. در سطح درک و فهم، فرآیند همبسته سازی جریان دارد که بخش‌هایی از آن در سطح درک و بخش‌هایی در سطح فهم انجام می‌شود. سطح تجسم شامل دو بخش درخت پسوندی و الگوریتم مدل مارکوف با طول متغیر قرار دارد. نتیجه نهایی این معماری، پیش‌بینی رفتار بعدی مهاجم (مهاجمین) در هر میزبان بخش و ارائه آن به تصمیم سازان است. خطوط سیاه، سبز و بنفش به ترتیب نشان دهنده داده آموزش، آزمون و داده نهایی است. در ادامه به هر یک از این بخش‌ها بیشتر خواهیم پرداخت.



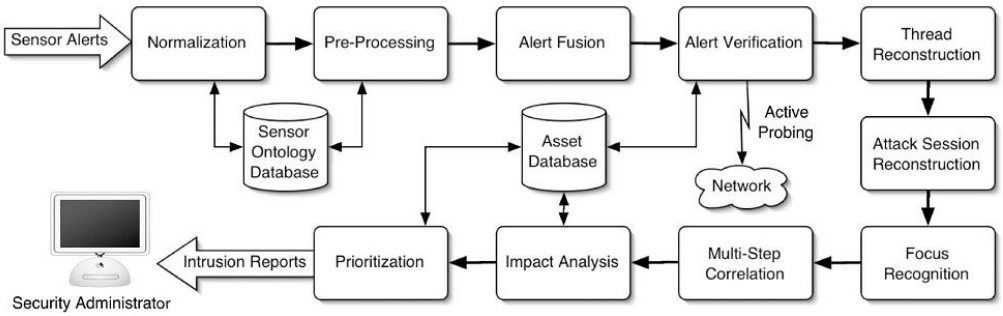
شکل ۳-۱: معماری کلی

۳-۳- فرآیند همبسته سازی

ولیوئر و همکاران[۴۱] در سال ۲۰۰۴ مدلی برای فرآیند همبسته سازی پیشنهاد کردند. این مدل درشکل ۳-۲ دیده می‌شود. در ابتدا دو کار بر همه هشدارها انجام می‌گیرد. در فاز اولیه بخش نرمال‌سازی[۱] هر هشدار را که دریافت می‌کند، به یک فرمت استاندارد که قابل فهم برای بخش‌های دیگر همبسته سازی باشد ترجمه می‌کند. در فاز بعدی بخش پیش‌پردازش[۲] فعال‌شده و به بررسی فیلدهای هشدارها می‌پردازد تا همه فیلدها مقادیر معناداری داشته باشند. بخش بعدی، بخش ادغام است. این بخش مسئول ترکیب کردن هشدارهایی است که از حسگرهای مختلف ولی برای یک حمله مشخص صادرشده‌اند. بخش بعدی ممیزی هشدار[۳] است.این بخش یک هشدار را می‌گیرد و موفقیت حمله‌ای که باعث بوجود آمدن این هشدار شده است را مشخص می‌کند. دلیل این کار این است که بتوان حملات و هشدارهای موفق را از غیر آن تشخیص داد. بازسازی

[1] Normalization
[2] Preprocessing
[3] Alert Verification

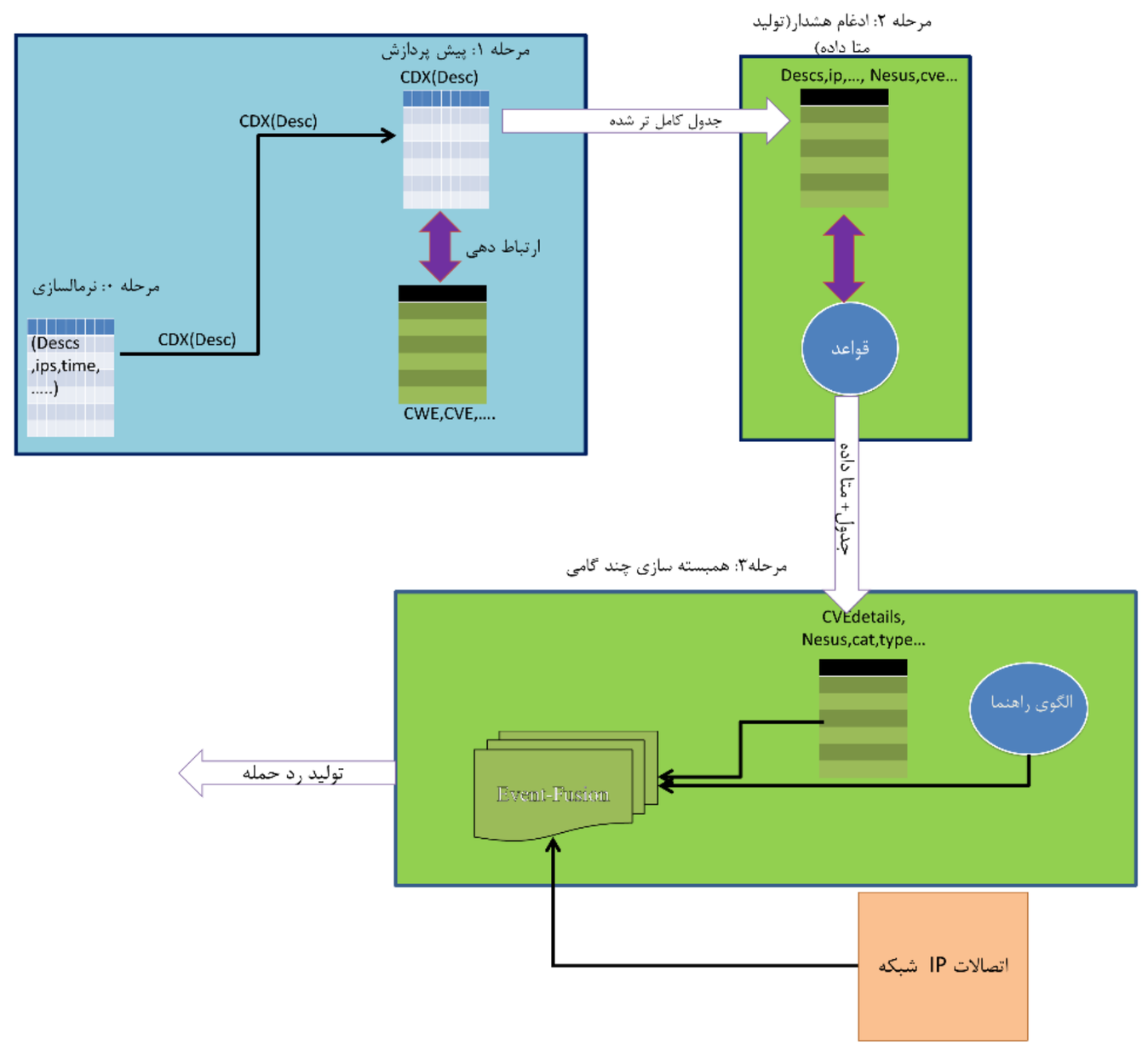
ریسمان[1]بخش بعدی در این فرآیند است. وظیفه این بخش ترکیب یک سری از هشدارها است که از حملاتی صادرشده‌اند که در آن یک مهاجم، یک مقصد را مورد حمله قرار داده است. بخش بعدی که بازسازی نشست حمله [2] است مسئولیتش مرتبط کردن هشدارهای مبتنی بر شبکه با هشدارهای مبتنی بر میزبان است. بخش بازشناسی تمرکز، وظیفه شناسایی میزبان‌هایی را دارند که مبدا یا مقصد تعداد زیادی از حملات بودند. این مورد برای تشخیص حملات انکار سرویس[3] استفاده می‌شود. بخش دیگر در این فرآیند همبسته سازی چند مرحله ای[4] است که وظیفه‌اش مشخص کردن حملاتی سطح بالا است که از روی سناریوهایی شناسایی می‌شوند. این سناریوها از دنباله‌ای از رویدادها ایجادشده‌اند. بخش تحلیل اثر[5] میزان تاثیر حملات شناسایی‌شده را بر عملکرد شبکه‌ای که در حال نظارت شدن است ارزیابی می‌کند. در نهایت بر اساس همین ارزیابی‌ها، بخش اولویت‌دهی[6] یک اولویت مناسب را به هر یک از این هشدارها اختصاص می‌دهد. شایان ذکر است که ترتیب قرار گرفتن اجزاء این فرآیند در این مدل می‌تواند لزوما به این شکل نباشد[۴۱].



شکل ۳-۲: فرآیند همبسته سازی[۴۱]

شکل ۳-۳ مدل پیشنهادی ما برای فرآیند تولید رد حملات مطابق با مدل ارائه‌شده توسط ولیوئر [۴۱] است. بخش‌هایی از این فرآیند در سطح درک و بخش‌هایی در سطح فهم قرار می‌گیرد.

[1] Thread Reconstruction
[2] Attack Session Reconstruction
[3] Denial-of-Service(DoS)
[4] Multistep Correlation
[5] Impact Analysis
[6] Prioritization



شکل ۳-۳: فرآیند تولید رد حملات شبکه‌ای

۳-۴- دادگانCDX

هدف این پایان‌نامه این است که چه طور می‌توان از روی رابطه علت و معلولی رفتارهای موجود در یک حمله به یک موجودیت، به پیش‌بینی رفتار آینده بر روی آن موجودیت پرداخت. یک دادگان مناسب برای این هدف، دادگانی است که در آن ردهای حملات موجود باشد. هر رد هم حاوی رویدادهای مختلف است. رویدادها می‌توانند انواع هشدارها باشند. ولی متاسفانه چنین دادگانی در حال حاضر موجود نیست. از دادگان موجود می‌توان به دادگان MIT [42][43]، KDD Cup99[۴۴] و Defcon [۴۵]اشاره کرد.

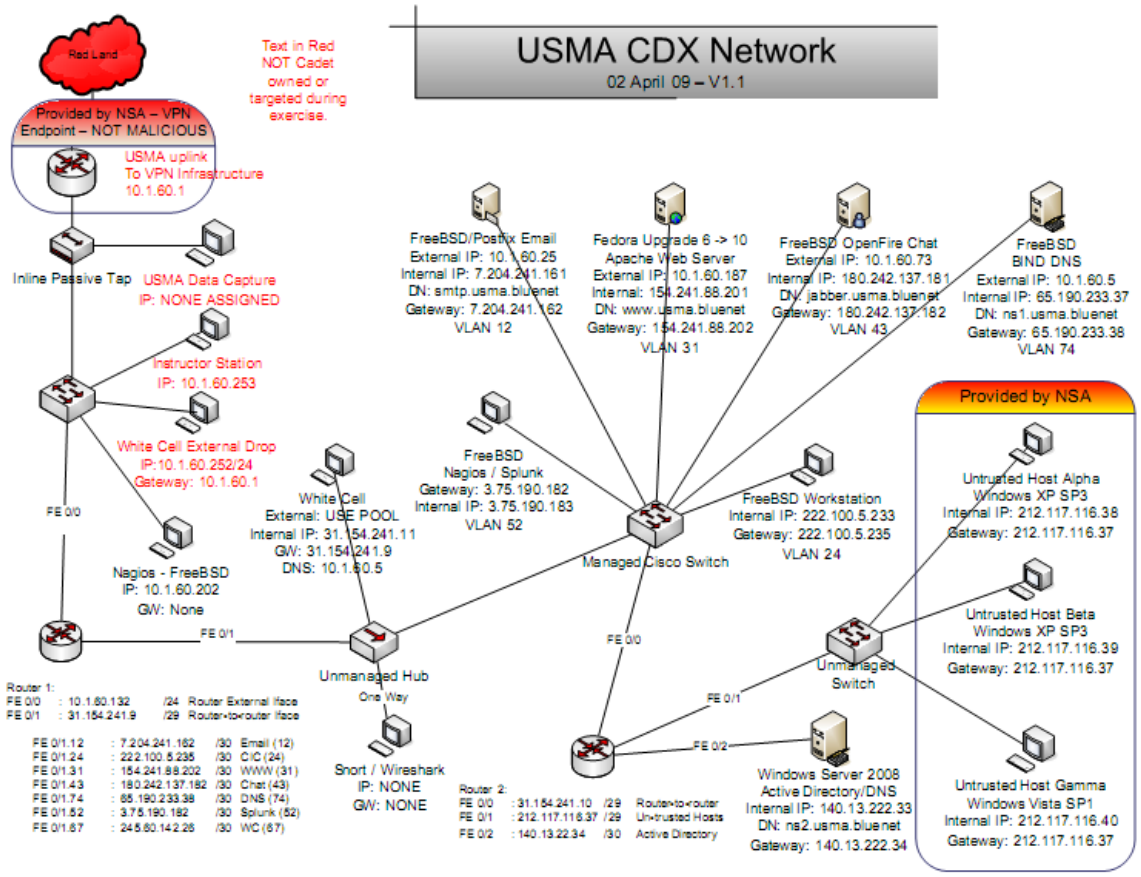
ما برای کار خود از دادگان CDX2009[۴۶] استفاده کردیم. دادگان CDX حاوی لاگ های حاصل از نرم افزار جلوگیری از نفوذ Snort است. این دادگان در طی یک رقابت مبارزه‌ای شبکه‌ای که در آن یکی از اهدافش تولید یک دادگان برچسب خورده بود تولید شده است. زیرساخت شبکه شامل چهار سرور است که در جدول ۲سرویس ها و اطلاعات موجود از این سرورها قرار داده شده است. دو آدرس IP در این جدول وجود دارد:

- آدرس IP داخلی که متناظر با لاگ Snort است
- آدرس IP خارجی که متناظر با یک TCP dump شبکه خارج از مرز این شبکه است.

جدول ۲: لیستی از سرورهای آسیب‌پذیر

سرویس	سیستم عامل	IP داخلی	IP خارجی
Postfix Email	**FreeBSD**	**7.204.241.161**	**10.1.60.25**
Apache Web Server	**Fedora10**	**154.241.88.201**	**10.1.60.187**
OpenFire Chat	**FreeBSD**	**180.242.137.181**	**10.1.60.73**
BIND DNS	**FreeBSD**	**65.190.233.37**	**10.1.60.5**

در شکل ۳-۴ شبکه طراحی‌شده در این مسابقه را مشاهده می‌شود.



شکل ۳-۴: شبکه CDX [۴۷]

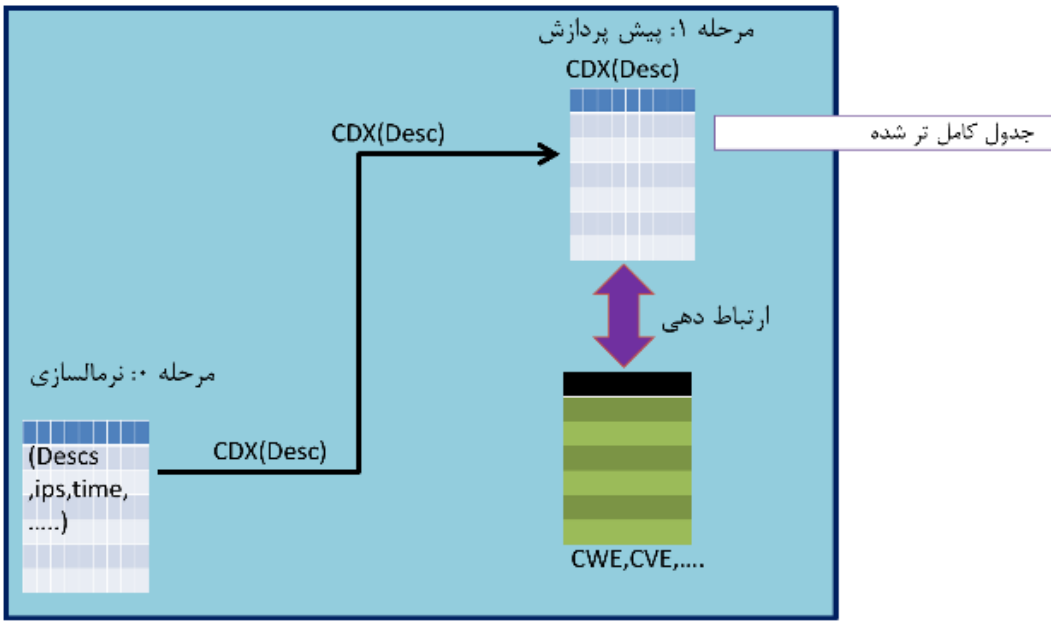
در زیر نیز یک مثال از لاگ های این دادگان قرار داده شده است:

```
[**] [1:3638:7] WEB-CGI SoftCart.exe CGI buffer overflow attempt [**]
[Classification: Web Application Attack] [Priority: 1]
11/08-13:05:38.382425 10.2.190.254:60089 -> 154.241.88.201:80
TCP TTL:61 TOS:0x0 ID:47842 IpLen:20 DgmLen:1200 DF
***A**** Seq: 0xFF9D1147  Ack: 0x9AEB5140  Win: 0xB7  TcpLen: 32
TCP Options (3) => NOP NOP TS: 961453 78007145
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2004-2221][Xref => http://www.securityfocus.com/bid/10926]
```

از دلایل استفاده از این دادگان، اعتبار آن، جدیدتر بودن تاریخ آن، داشتن مراجع مناسب برای هشدارها و مشخص بودن ساختار شبکه تحت آزمایش می‌توان اشاره کرد.

۳-۵- درک

بخش درک در شکل ۳-۵ مشاهده می‌شود. در این بخش دو عمل نرمال‌سازی و پیش‌پردازش انجام می‌شود.



شکل ۳-۵ :بخش درک معماری کلی

۳-۵-۱- نرمال‌سازی

در بخش نرمال‌سازی، تمام هشدارها را در یک فرمت قرار می‌دهیم. فیلدهای این فرمت را در جدول ۳مشاهده می‌کنید.

جدول ۳: فرمت استاندارد اعمال شده بر دادگان

فیلدها	توضیح
IDs	**ID هر رکورد**
Description	**توصیف Snort**
Classification	**طبقه‌بندی Snort**
Priority	**اولویت اختصاصی از Snort**
sDate	**تاریخ صدور این هشدار از Snort**
sTime	**زمان صدور این هشدار ازSnort**
SourceIP	**IP مبدا**
SourcePort	**پورت مبدا**
DestinationIP	**IP مقصد**

DestinationPort	پورت مقصد
CVEID	ID در سایت CVE
NessusID	ID در سایت Nessus
BugtraqID	ID در سایت Bugtraq
CVED	توصیف نوشته‌شده در سایت CVE
Score	امتیاز اختصاص داده‌شده براساس CVSS
CImpact	تاثیر بر محرمانگی از CVSS
IImpact	تاثیر بر صحت از CVSS
AImpact	تاثیر بر دسترس‌پذیری از CVSS
AccessC	پیچیدگی دسترسی از CVSS
Authentication	لزوم احراز هویت از CVSS
GAccess	سطح دسترسی
Vulnerability	بهره‌کشی
CWEID	ID در سایت CWE
App	نرم افزار آسیب‌پذیر
Synopsis	خلاصه‌ای در مورد آسیب‌پذیری (Nessus)
Des	توصیف بیشتر از Nessus
RiskFactor	اعلام CVSS از Nessus
BugtraqD	نرم افزار و نسخه آسیب‌پذیر (Bugtraq)
Class	دسته‌بندی پیشنهادی از Bugtraq
Remote	آیا حمله از خارج شبکه قابل انجام است؟ (Bugtraq)
Local	آیا حمله از داخل شبکه قابل انجام است؟ (Bugtraq)
Info	اطلاعاتی در مورد این ضعف (Bugtraq)

۳-۵-۲- پیش‌پردازش

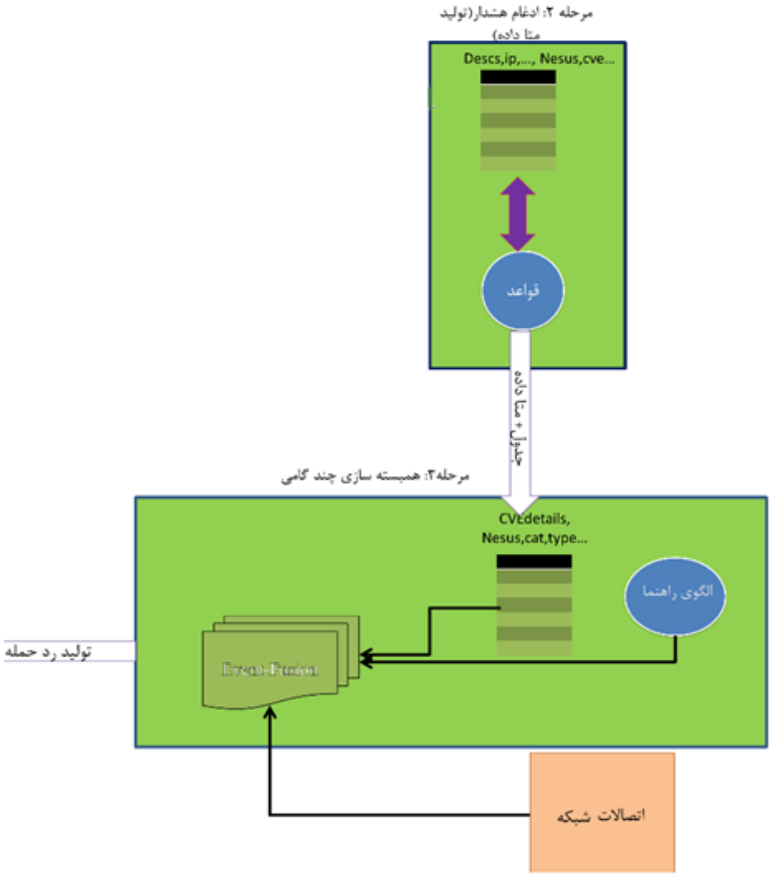
در بخش پیش‌پردازش از طریق روابط میان فیلدها، فیلدهای خالی در بعضی از رکوردهای پایگاه داده را کامل می‌کنیم. به عنوان مثال بعضی از رکوردهای پایگاه داده، دارای فیلد CVE

بودند ولی فیلد CVSS نداشتند. در این کار ما از طریق ایجاد ارتباط میان جداول دیگر، این مقدار را به دست آوردیم. به‌طور کلی ما از اطلاعات زیر در مرحله نرمال‌سازی بهره برده‌ایم[۴۸]:

- سایت CVEdetails
- سایت Bugtraq
- سایت Nessus
- اطلاعات موجود در هشدارهای Snort

۳-۶- فهم

بخش فهم در شکل ۴-۶دیده می‌شود. در این بخش دو عمل ادغام هشدار (تولید متا-داده) و همبسته سازی چند مرحله‌ای صورت می‌گیرد.



شکل ۳-۶: بخش فهم معماری کلی

۳-۶-۱- ادغام هشدار (تولید متا-داده)

در بخش ادغام هشدار با استفاده از قوانین[۱] از پیش تعریف‌شده (که با زبان قاعده منظم[۲] نوشته می‌شود) و مجموعه فیلدهای دارای توصیف در هر رکورد پایگاه داده (فیلد هایی که در مورد هشدار توضیح ارائه می‌کنند)، به هر رکورد یک یا چند برچسب سطح بالا از نوع طبقه‌بندی[۳] و بهره‌کشی[۴] زده می‌شود.

برچسب‌های طبقه‌بندی (با علامت ✓) و زیر طبقه‌بندی (با علامت •) در زیر آورده شده است:

- ✓ شناسایی: در این مرحله به دنبال جمع اوری اطلاعات هستیم.
  - شنود[۵]: شنود بر روی خطوط
  - ردیابی[۶]: به‌طور کلی شبکه چه ویژگی‌هایی دارد؟
  - پویش: چه سیستم‌هایی فعال‌اند؟ اهداف فعلا به صورت دقیق واضح نشده اند.
  - سرشماری[۷]: چه چیز در مورد حساب‌های کاربری افراد فهمید؟
- ✓ نفوذ: مهاجم قصد وارد شدن دارد یا هر ضعفی که امکان نفوذ مستقیم را در یکی از سطوح زیر بدهد.
  - نفوذ در حساب ریشه[۸]
  - نفوذ در حساب کاربر[۹]
  - نفوذ در دیگران[۱۰]
- ✓ بالا بردن اولویت: دو مرحله‌ای است؛ یعنی مهاجم ابتدا واردشده (local) فرض می‌شود و سپس برای رفتن به سطوح دیگر تصمیم می‌گیرد.
  - بالا بردن اولویت به حساب ریشه
  - بالا بردن اولویت به حساب کاربری
  - بالا بردن اولویت به حساب دیگری
- ✓ هدف
  - تخریب[۱]: هر نوع خرابکاری بر روی ماشین، داده، حافظه و... . یا عملی که

---

[1] Rules
[2] Regular Expression
[3] Category
[4] Exploit
[5] Sniffing
[6] Footprinting
[7] Enumeration
[8] Intrusion Admin
[9] Intrusion User
[10] Intrusion Other

منجر به خرابکاری شود. به غیر از انکار سرویس.

- حمله انکار سرویس: موجب جلوگیری از ارائه سرویس به همه کاربران می‌شود.
- دزدی اطلاعات حساس[۲]: اطلاعاتی مانند پسوردها و...
- دزدی‌های کم‌ارزش (دله‌دزدی)[۳]: هر کاری که هدفش به دست اوردن و خارج کردن فایل و یا جداول پایگاه‌های داده باشد.
- ایجاد درب پشتی[۴]

نمونه‌ای از قوانین را هم در جدول ۴ مشاهده می‌کنید.

جدول ۴: نمونه‌ای از قوانین

توضیح	قوانین	برچسب
از راه دور عمل overflow را انجام دهد. (البته اگر فیلد remote فقط yes باشد؛ یعنی آسیب‌پذیری این نرم افزار از طریق local ممکن نباشد.)	"(*.)remote(.*)overflow(*.)"	نفوذ
در جمله اگر کلمه اول enumerate و بعد آن برای یکبار users آمده باشد. قبل و بعدش هر چیزی می‌تواند باشد.	(.*)enumerate(.*?)users(.*)	سرشماری
اگر در یک جمله دو کلمه stack و corrupt باشد. بینشان و قبلشان هر چیزی می‌تواند باشد. حروف بزرگ و کوچک هم اهمیت ندارد.	?i:(.*)stack(.*)corrupt(.*)	هدف-تخریب

[1] Corruption
[2] Espionage
[3] Pilfering
[4] Backdoor

برچسب‌های بهره‌کشی، از لحاظ انتزاع در سطح پایین‌تری نسبت به طبقه‌بندی و زیر طبقه‌بندی قرار دارند. در حقیقت بهره‌کشی‌ها حاوی نشانگرهای طبقات هستند. به عبارت دیگر این برچسب‌ها حاوی قواعد منظم هستند. این برچسب‌ها را در جدول ۵ مشاهده می‌کنید.

جدول ۵: برچسب‌های بهره‌کشی و طبقات آن‌ها

بهره‌کشی	طبقه‌بندی
backdoor	نفوذ
Overflow remote	
Authentication bypass	
injection	
Cross-site	
Code execute	
Overflow local-	بالا بردن اولویت
Authentication bypass	
privilage	
Memory Stack	هدف-تخریب
Modify Database	
Deny service	هدف-انکار سرویس

Authentication files access	هدف- دزدی اطلاعات حساس
Cross site	
access files	هدف- دزدی‌های کم‌ارزش
backdoor	هدف-درب پشتی

ما در ادامه، هر رکورد پایگاه داده را یک رخداد[1] می‌نامیم. هر رخداد به ازای هر برچسب بهره‌کشی حاوی یک رویداد خواهد بود؛ زیرا توصیفاتی که برای هر رخداد وجود دارد هر کدام ممکن است به طبقات و بهره‌کشی‌های مختلفی اشاره کنند. به عنوان مثال اگر یک رخداد با دو طبقه نفوذ و پویش برچسب خورده باشد و برای زیر طبقه نفوذ، دو برچسب بهره‌کشی و برای پویش یک برچسب بهره‌کشی داشته باشیم، می‌بایست سه رویداد از این رخداد وارد چرخه تولید رد حمله شود.

۳-۶-۲- همبسته سازی چند مرحله‌ای

به طور کلی هر حمله ای می تواند تک مرحله ای یا چند مرحله ای باشد. اگر هدف مهاجم از انجام حمله، به عنوان مثال تنها اجرای عمل شنود باشد، به این حمله یک حمله تک مرحله ای گویند. از طرف دیگر اگر مهاجم ابتدا عمل شنود را انجام دهد، سپس اقدام به نفوذ در سطح کاربر کند و پس از آن اقدام به نفوذ در سطح ریشه نماید، این نوع حمله را حمله چند مرحله ای می‌نامند. این نوع حمله می‌تواند با گذشت زمان توسعه پیدا کند و از ماشینی به ماشین دیگر و از دامنه ای به دامنه دیگر منتقل شود.

[1] Incident

بخش همبسته سازی چند مرحله‌ای همانطور که در شکل ۳-۶ دیده می‌شود شامل دو مؤلفه الگوی راهنما و پایگاه داده‌ای حاوی داده‌های حاصل از مراحل قبل است. به این بخش اتصالات شبکه نیز به عنوان ورودی وارد می‌شود.

الگوی راهنما الگویی کلی، ممکن و صحیح را برای ایجاد ردهای چند مرحله ای نشان می‌دهد. اطلاعات ورودی اتصالات شبکه نیز برای مشخص کردن موجودیت‌های شبکه استفاده می‌شود؛ به عبارت دیگر در نهایت قرار است ردهای حمله بر موجودیت‌های مختلف شبکه با توجه به الگوی راهنما ایجاد شوند. در دو بخش پیش رو الگوی راهنما و فرآیند تولید رد حمله بیشتر شرح داده می‌شود.

۳-۶-۳- الگوی راهنمای پیشنهادی

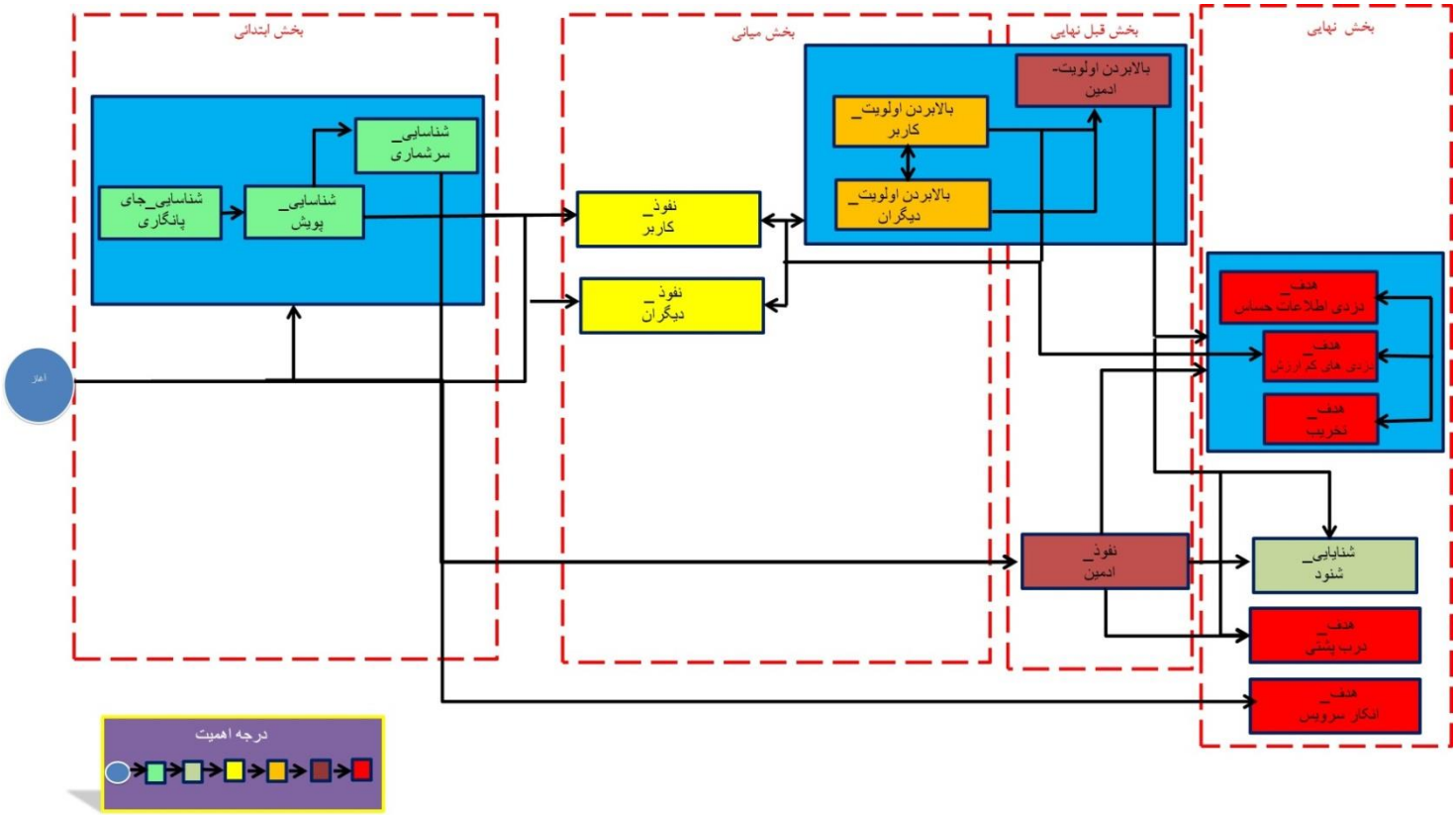
الگوی راهنمای پیشنهادشده در این مقاله که در شکل ۳-۷ دیده می‌شود، توسعه داده‌شده الگوی راهنمای در[۳۹] است. طبقات این الگو، همان برچسب‌های زیر طبقه‌بندی معرفی‌شده در بخش ادغام هشدار هستند. این الگو دارای چهار بخش[1] کلی است. بخش اول [2] شامل اعمال اولیه شامل ردیابی، پویش و سرشماری است. در حقیقت هدف اصلی در این بخش صرفا جمع‌آوری اطلاعات است. بخش دوم یا همان بخش میانی[3] شامل اعمالی مانند نفوذ به سطح کاربر، نفوذ به سطح کاربری دیگر، بالا بردن سطح اولویت به سطح کاربر و بالا بردن سطح اولویت به سطح دیگر می‌شود. هدف این بخش ایجاد جای پا برای انجام مراحل بعدی است. بخش سوم یا بخش قبل نهایی[4] بخشی است که در آن مهاجم به حساب ادمین یا ریشه دست پیدا می‌کند. رسیدن به این بخش می‌تواند به صورت مستقیم(نفوذ به حساب ادمین) یا غیر مستقیم(بالا بردن سطح اولویت به ادمین) انجام گیرد. در نهایت هم بخش پایانی[5] قرار دارد که در آن اهداف مهاجم دنبال می‌شود. اهدافی مانند تخریب، حمله انکار سرویس، دزدی اطلاعات حساس، دزدی‌های کم‌ارزش و ایجاد درب پشتی قرار دارد.

در این الگو بر خلاف مدل [۳۹]، برای حرکت میان یالها تنها دو حرکت مجاز است. یکی حرکت افقی (تنها از چپ به راست) و دیگری حرکت عمودی (فرقی در بالا به پایین یا پایین به بالا وجود ندارد). در حقیقت در این الگو چرخه وجود ندارد. دلیل حرکت به یک سمت این است که بتوان از این الگو بهتر در هنگام ایجاد ردهای حمله استفاده نمود. هر رد حمله قرار است نشان دهنده یک حمله باشد. این که آیا دو حمله آیا باهم مرتبط هستند یا نه (تولید سناریو) از بحث این الگو خارج است. رنگ ها نیز در این الگو از سبز به قرمز، میزان اهمیت هر

[1] Section
[2] Initial Section
[3] Intermediate Section
[4] Pre-Final Section
[5] Final Section

گام را نشان می‌دهد. به‌طور کلی می‌توان تفاوت‌های این مدل را با مدل [۳۹] در موارد زیر خلاصه کرد:

- تفاوت در چهار گام هر حمله (همان چهار بخش کلی)
- قرار دادن طبقات و زیر طبقات متناسب با این چهار بخش کلی
- تغییر زیر طبقات طبقه بالا بردن اولویت
- حذف چرخه‌ها در مدل و یک طرفه کردن آن



شکل ۳-۷: الگوی راهنمای پیشنهادی

۳-۶-۴- تولید رد حمله

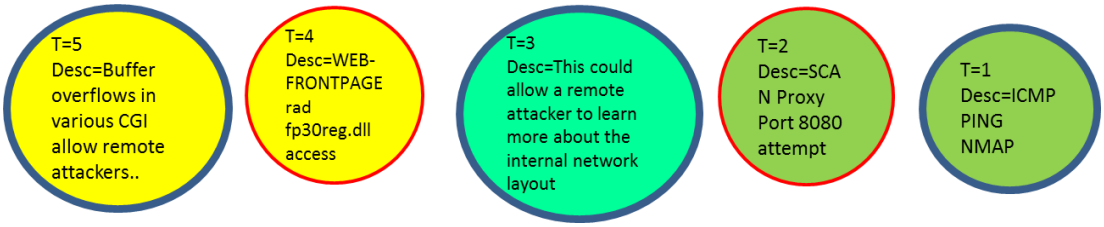
این مرحله وظیفه‌اش تولید رد حملات است. در زیر سودوکد تولید رد حمله را مشاهده می‌کنید:

```
main(event e){
IP ip=find_ip(e);//ابتدا آی پی این رویداد را پیدا کن. به عبارتی کجا این رویداد رخ داده.
trackslist=tp.tracks; //رد های حمله داخل این آی پی را استخراج کن.
  while( trackslist.hasnext ){// تا زمانی که رد حمله ای وجود دارد
    track t=trackslist.gettrack;// هر رد حمله را بگیر
    checktracks (e,t);// و برای بررسی به تابع چک بفرست.
  }
}
checktracks(input event,track t){
  if event.subcat ∈ t.events.get_subcatslist{
    if event.type ∈t.events.gettypeslist{//اگر از یک زیر طبقه بودند
      sorted_exploitlist=sortbytime(event.type,track.similarexploit_events);//رویداد هایی از رد حمله را که از نظر برچسب بهره کشی مشابهت با
      //رویداد ورودی دارند را جداکن و آنها را براساس زمان اعلامشان مرتب کن
      insert(event,t.getplace(lastevent_of(sorted_exploitlist)));// آخرین رویداد داخل این لیست را بگیر
      // و سپس مکان این رویداد را در رد حمله پیدا کن. سرانجام این رویداد ورودی را در بعد این مکان اضافه کن.
    }else{// نشان می دهد که مشابهت تنها از نوع زیر طبقه است.
       sorted_eventlist=sortbytime(event.subcat,track.similarsubcat_events);//رویدادهایی از رد حمله را که
       زیر طبقه مشابهی با زیر طبقه رویداد ورودی دارند را جدا کن و آنها را بر اساس زمان مرتب کن.
       insert(event,t.getplace(last event of(sorted eventlist))); آخرین رویداد داخل این لیست را بگیر
    // و سپس مکان این رویداد را در رد حمله پیدا کن. سرانجام این رویداد ورودی را در بعد این مکان اضافه کن.
    }
  }else{//زیر طبقات قبل این رویداد ورودی را بدست آورده و بررسی می کنیم که آیا این هیچ کدام از
       //این طبقات در لیست زیر طبقات رد حمله وجود دارد یا نه
     if(event.past_subcatlist exist in t.events.getsubcatlist){
          if (check between two events){// بررسی می کنیم که آیا می توان بین دو رویداد در لیست ، رویداد ورودی را قرار داد؟
           insert between two events;//بین دو رویداد ، رویداد ورودی را قرار بده
           delete past relation;// رابطه سابق بین ان دو رویداد را حذف کن.
          }else{//مشخص می شود که بین هیچ دو رویدادی نمی توان این ورودی را قرار داد
           insert(event,last(t.eventslist));//رویداد ورودی را به آخرین رویداد رد حمله اضافه کن.
          }
     }else{//نشان می دهد که نمی تواند به چیزی وصل شود، پس آن را به لیست آرشیو اضافه کن.
        insertarchive(event,archivelist)
     }
  }
}
```

شکل ۳-۸: سودوکد تولید رد حملات شبکه‌ای

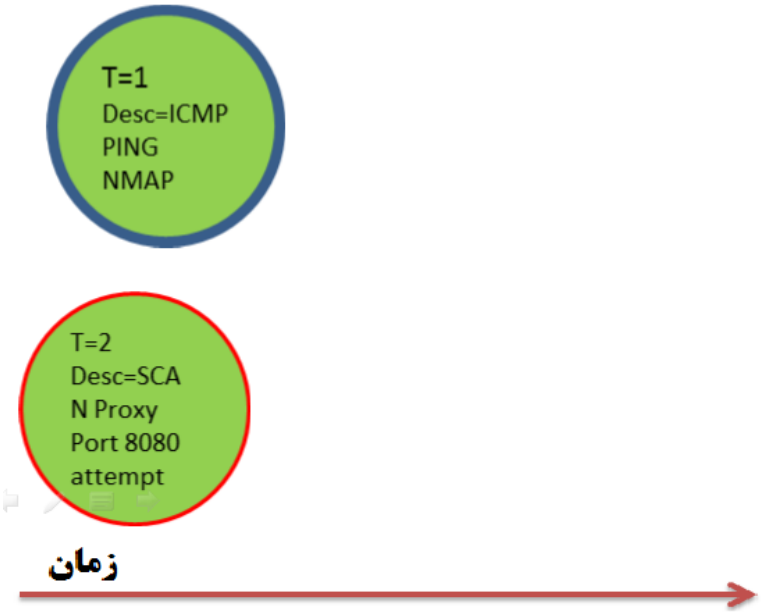
در ادامه با ذکر یک مثال ساده این روند تشریح می‌شود. شکل ۳-۹ بخشی از الگوی راهنمای پیشنهادی را نشان می‌دهد. شکل ۳-۱۰ رویدادهای ورودی را نشان می‌دهد این ورودی‌ها هم از پایگاه داده اطلاعات ما می‌آید. در داخل هر ورودی زمان ورود آن با حرف T و توصیف آن با حرف Desc مشخص شده است. رنگ هر رویداد نشان دهنده این است که آن رویداد متعلق به کدام طبقه در الگوی راهنما است. همچنین در این مثال ما دو IP(موجودیت) داریم. که با رنگ های آبی و قرمز مشخص شده اند.

شکل ۳-۹: بخش از الگوی راهنما



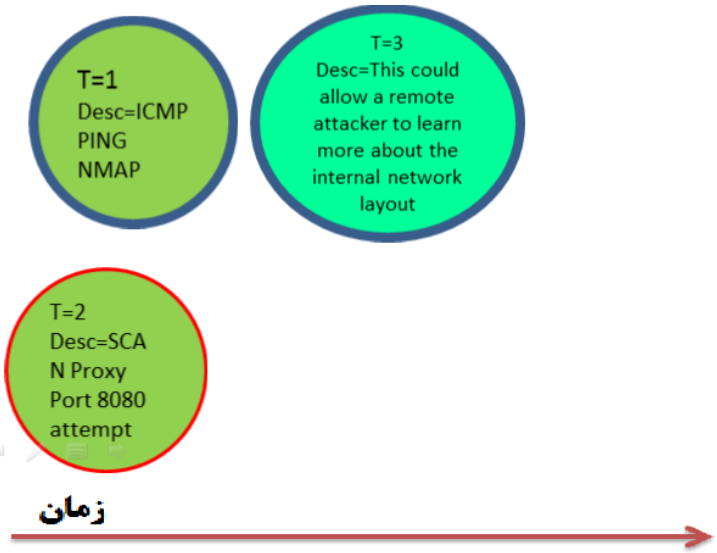
شکل ۳-۱۰: ورودی‌ها به ترتیب زمان

در مرحله اول رویدادهای t=1 و t=2 وارد می شوند(شکل ۳-۱۱).

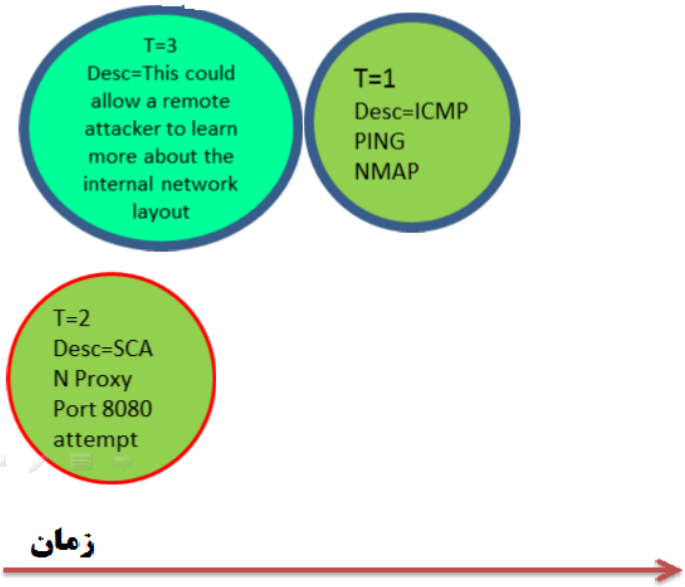


شکل ۳-۱۱: مرحله اول تولید یک رد حمله

در مرحله دوم رویداد t=3 وارد می‌شود و ابتدا به صورت عادی در مکـانی جلـوتر از رویـداد t=1 قرار می‌گیرد (شکل ۳-۱۲). ولی در مرحله سوم جای این دو رویداد به دلیل ترتیب موجود در الگوی راهنما عوض می‌شود (شکل ۳-۱۳).

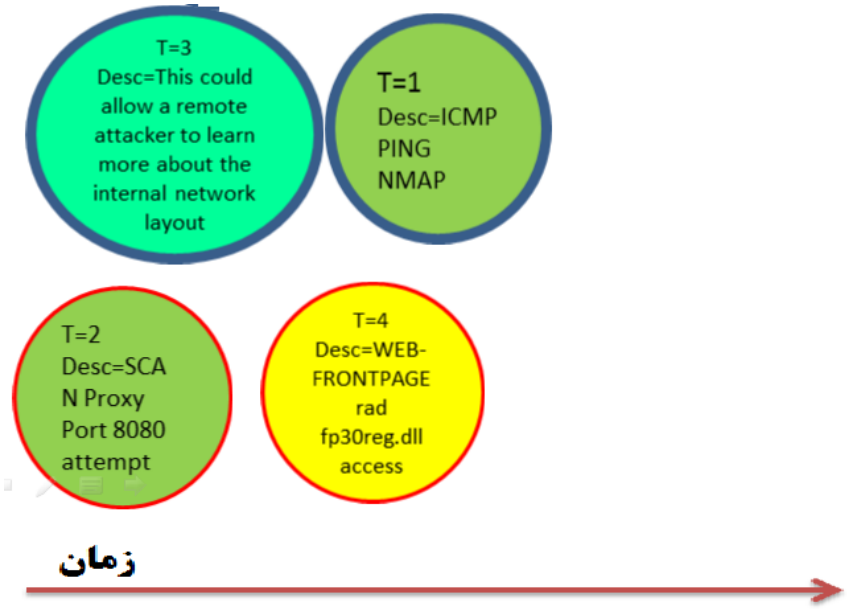


شکل ۳-۱۲: مرحله دوم تولید یک رد حمله

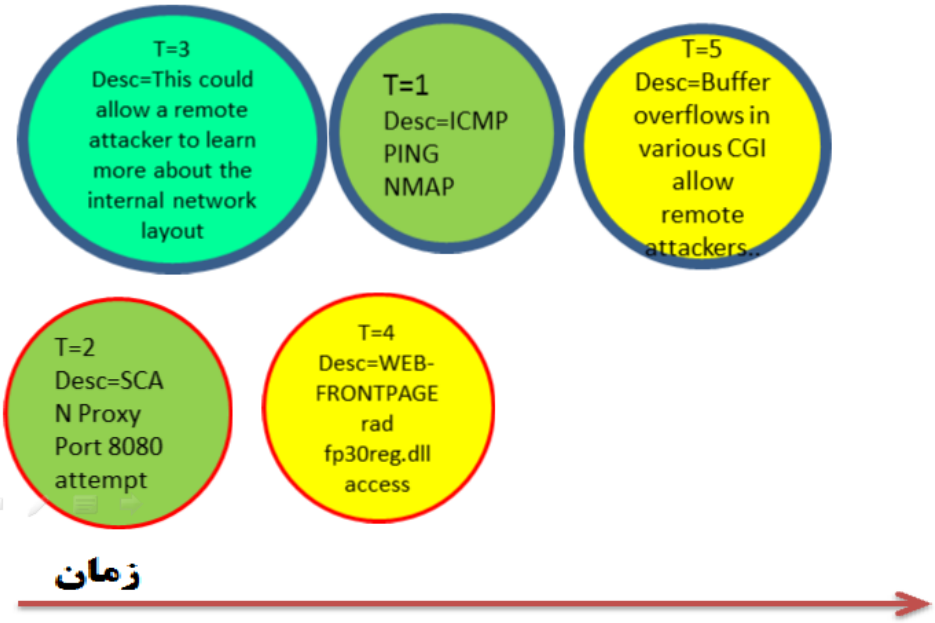


شکل ۳-۱۳: مرحله سوم تولید یک رد حمله

در نهایت هم رویداد t=4 وارد می‌شود که بر طبق الگوی راهنما، در جلـوی رویـداد t=2 در IP دوم قرار می‌گیرد. در نهایت همانطور که در شکل ۳-۱۵ مشاهده می‌شود ما دو رد حمله، در دو IP مقصد خواهیم داشت.



شکل ۳-۱۴: مرحله چهارم تولید یک رد حمله



شکل ۳-۱۵: مرحله پنجم تولید یک رد حمله

در شکل ۳-۱۶ نمونه‌ای از یک رد حمله مشاهده می‌شود:

{"id_attack_track":1,"events":[{"fields_base":{"CVED":"","Description":"ET SCAN NMAP -f -sS","Classification":"Attempted Information Leak","Synopsis":"","BugtraqD":"","Des":"","Info":""},"subcat":"Scanning","sDate":"11/08","sTime":"11:08:32.588672","destip":"7.204.241.161","stype":"","RiD":"2"},{"fields_base":{"CVED":"clamav-milter in ClamAV before 0.91.2, when run in black hole mode, allows remote attackers to execute arbitrary commands via shell metacharacters that are used in a certain popen call, involving the \u0026quot;recipient field of sendmail.\u0026quot;","Description":"SMTP ClamAV recipient command injection attempt","Classification":"Attempted Administrator Privilege Gain","Synopsis":"","BugtraqD":"ClamAV Popen Function Remote Code Execution Vulnerability","Des":"","Info":"OpenSSL is prone to an off-by-one buffer-overflow vulnerability because the library fails to properly bounds-check user-supplied input before copying it to an insufficiently sized memory buffer.Successfully exploiting this issue may allow attackers to execute arbitrary machine code in the context of applications that use the affected library, but this has not been confirmed. Failed exploit attempts may crash applications, denying service to legitimate users. NOTE: This issue was introduced in the fix for the vulnerability described in BID 20249 (OpenSSL SSL_Get_Shared_Ciphers Buffer Overflow Vulnerability).","subcat":"intrusion_admin","sDate":"11/08","sTime":"13:03:38.552291","destip":"7.204.241.161","stype":"2","RiD":"34"}]}]}

شکل ۳-۱۶: نمونه ای از رد حمله تولید شده

این رد حمله در قالب json ساخته‌شده است. به‌طور کلی در این کار هر مقصد، دارای لیستی از ردهای حمله است که هر رد حمله با یک id مشخص شده است. هر رد حمله نیز حاوی لیستس از رویدادها است. هر رویداد هم دارای اطلاعاتی است. این اطلاعات شامل یک لیست به نام fields_base است. این لیست حاوی اطلاعات توصیفی درباره هر هشدار است. فیلد های دیگر هر رویداد عبارت است از:

- Subcat : زیر طبقه آن هشدار را نشان می‌دهد.
- sDate: تاریخی را نشان می‌دهد که این رویداد صادر شده است.
- sTime: زمانی را نشان می‌دهد که این رویداد صادر شده است.
- Destip: IP مقصد را نشان می‌دهد.
- Stype: حاوی برچسب بهره‌کشی این رویداد است (این مقدار عددی است).
- RiD: به رکورد این هشدار در پایگاه داده اشاره می‌کند.

۳-۷- تجسم

همانطور که در معماری کلی (شکل ۳-۱) دیده می‌شود این سطح دارای دو بخش اصلی الگوریتم مدل مارکوف با طول متغیر[1] و درخت پسوندی است. یکی از دلایل تاکید بر دنباله‌ای سازی رویدادها، استفاده از مدل های ریاضیاتی است که قابل استفاده در این حوزه هستند.

[1] VLMM

۳-۷-۱- مدل مارکوف با طول متغیر

همانطور که از نام این مدل پیداست مدل مارکوف با طول متغیر، یک مدل از خانواده مدل های مارکوف است. مدل های مارکوف نیز خودشان نوع پیچیده‌تری از مدل‌های n-gram هستند.

۳-۷-۱-۱- مدل‌سازی N-Gram

مدل‌سازی N-Gram یک روش تحلیل داده برای جریان‌های دنباله‌ای داده[1] است. یک جریان داده شامل چندین واژه[2] یا n-gram است. یک n-gram دنباله‌ای به صورت $\{x_{(i-n)},.....x_i, x_{i+1}\}$ یا $x^{i+1}_{(i-n)}$ به طول n است که شامل یک محتوی[3] $x^{i}_{(i-n)}$ و یک پیش‌بینی رویداد x_{i+1} می‌شود. هدف این نوع مدل‌سازی این است که با تولید واژگان مختلف و ارتباط میان آن‌ها، اطلاعات مورد نظر خود را از آن‌ها استخراج کند.

به عنوان مثال فرض کنید دنباله S به صورت S={ABAACAABAA}باشد. در مرحله اول اگر uni-gramها را در نظر بگیریم، در این دنباله ما ۷ عدد A، ۲ عدد B و ۱ عدد C خواهیم داشت. می‌توانیم بگوییم که احتمال A برابر 7/10 است. حال برای اینکه بتوانیم واژه بعدی در این دنباله را حدس بزنیم، نیاز داریم که احتمال bi-gramها را محاسبه کنیم. به عنوان مثال برای محاسبه احتمال اینکه واژه بعدی این دنباله،A باشد ما باید به دنبال bi-gram AA باشیم؛ بنابراین با توجه به جدول ۶ خواهیم داشت:

$$P(A|A)=P(AA)/(P(AA)+P(AB)+P(AC))=3/(3+2+1)=50\% \quad (۱)$$

می‌بینیم که این بار برخلاف احتمال قبلی، احتمال اینکه حرف بعدی A باشد 50% اعلام شده است. هر چقدر که ما این n-gramها را تغییر دهیم ممکن است با مقادیر متفاوت‌تری نیز برخورد کنیم. از طرفی هم باید توجه داشت که هر چقدر n-gram ها بلندتر می‌شوند تعداد تکرار آن‌ها نیز کمتر و کمتر می‌شود تا اینکه سرانجام به سطحی می‌رسد که دیگر تعداد تکرارشان با حالت تصادفی فرق چندانی ندارد؛ به عبارت دیگر بعد از یک طول معین، اطلاعات استخراج‌شده از n-gramهای مشاهده‌شده، خیلی خاص می‌شود و چندان برای تعیین اینکه چه دنباله‌هایی ممکن است در آینده اتفاق بیافتند مفید نیست.

[1] Sequential Streams of Data
[2] Term
[3] context

جدول ۶: مقادیر Bi-grams

Bi-gram	AA	AB	AC	BA	BB	BC	CA	CB	CC
تعداد تکرار	3	2	1	2	0	0	1	0	0

یک پارامتر n-gram، احتمال اتفاق افتادن یک پیش‌بینی بعد از محتوی است. چندین روش متفاوت برای تخمین پارامترهایn-gram وجود دارد که ساده ترین آنها تخمین احتمال بیشینه (ML)[1] نامیده می‌شود. رابطه(۲) این تخمین را نشان می‌دهد:

$$p(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{c(x_{i+1} \mid x^{i}_{(i-n)+1})}{\sum_{x \in \Omega} c(x \mid x^{i}_{(i-n)+1})} \qquad (۲)$$

در این رابطه c(s)تعداد تکرار n-grams را نشان می‌دهد. در مدل‌سازی n-gram میزان احتمال دنباله‌ای از رویدادها به صورت ضربی از احتمالات تخمینی رویدادها به شکل رابطه (۳) به دست می آید:

$$p(x_1^j) = \prod_{i=0}^{j} p(x_{i+1} \mid x^{i}_{(i-n)+1}) \qquad (۳)$$

مشکلات از آنجایی شروع می‌شود که از مدلهای ML با مرتبه ثابت استفاده می‌شود و رویدادهایی رخ می‌دهند که تا پیش از این اتفاق نیفتاده اند.

۳-۷-۱-۲- مدل‌های مارکوف:

نوع پیچیده‌تر مدل‌های n-gram، مدل‌های مارکوف است. مدل مارکوف، مدلی آماری از یک سری رویدادهای وابسته به زمان است. هر یک از این رویدادهای محتمل با یک نماد منحصر بفرد نشان داده می‌شود. در این مدل مجموعه‌ای از حالت‌ها تعریف می‌شود، به هر حالت یک نماد اختصاص‌یافته و احتمال انتقال از یک حالت به حالت دیگر مشخص می‌گردد. مدل مارکوف به وسیله مشاهده یک دنباله از رویدادها و محاسبه احتمالات انتقالی ما بین حالت‌ها آموزش داده می‌شود. دنباله رویدادها در این مدل باید ویژگی مارکوفی داشته باشند. این ویژگی می‌گوید که با فرض حالت جاری، حالت بعدی مستقل از حالت‌های پیشین است:

$$P(x_{t+1}|x_1 \dots x_t) = P(x_{t+1}|x_t) \qquad (۴)$$

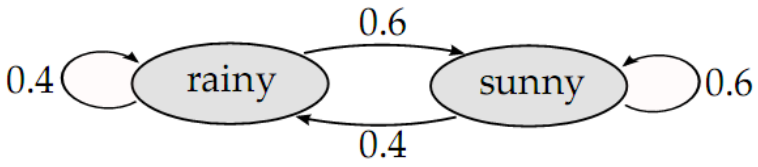
در رابطه(۴) ، $P(x_t)$نشان‌دهنده احتمال نماد بعدی است. به عبارتی در این دنباله نماد

[1]Maximum Likelihood

بعدی تنها به حالت جاری وابسته است و وابستگی به حالت گذشته(با فرض حالت جاری) و یـا آینده ندارد. این نوع دنباله‌ها که ویژگی مارکوفی در آن‌ها برقرار است، زنجیره مارکوف[1] یا مدل مارکوف مرتبه اول گفته می‌شود. پیش‌بینی کننده‌هایی که از یک چنـین مـدلهای پـیش‌بینـی استفاده می‌کنند در اصطلاح پیش‌بینی کننده‌های مارکوفی نامیده می‌شوند.در ادامه به‌سـادگی می‌توان رابطه (۴) را نیز توسعه داده تا اینکه حالت بعدی، به n حالت قبلی وابسته باشد:

$$P(x_{t+1} \mid x_1 .... x_t) = P(x_{t+1} \mid x_{(t-n)+1} ..... x_t) \qquad (۵)$$

در رابطه (۵)، به n ، مرتبه[2] گفته می‌شود. با این فرض ما یک مـدل مـارکوف از مرتبـه (n-1ام یا همان مدل n-gram را خواهیم داشت.یک مثال معروف برای مدل مارکوف،مثال آب‌وهوا است. در این مثال فرض می‌کنیم که دو نوع آب‌وهوا (دو حالت) داریم: بارانی[3] و آفتابی[4] که بـه ترتیب با R و S نشان می‌دهیم. شکل ۳-۱۷ احتمـالات انتقـالی میـان ایـن حالـت‌هـا رانشـان می‌دهد.



شکل ۳-۱۷:نمودار تغییر حالت

حال فرض کنید ما برای یک مدت ۱۱ روزه آب و هوای یک منطقه را زیر نظر گرفتـه‌ایـم و چنین دنباله‌ای به دست آمده است: S={RSSRRSSRRSS}. سـؤال ایـن اسـت کـه وضـعیت آب‌وهوا در روز دوازدهم به چه شکل است؟

ساده‌ترین مدل مارکوف که همان مارکوف مرتبه اول است به وسیله مشاهده bi-gramها ایجاد می‌شود زیرا در این مدل تنها به یک گام عقب‌تر نگاه می‌شود.این bi-gramها را می‌توانیم در جدول ۷ ببینیم:

جدول ۷ :نمودار bi-gram دنباله S={RSSRRSSRRSS}

Bi-gram	RR	RS	SR	SS
Count	2	3	2	3

همچنین می‌توانیم این جدول را به یک ماتریس انتقالی (جدول ۸) تبـدیل کنـیم. در ایـن ماتریس a_{ij}، احتمال انتقال از حالت جاری i در زمان t به حالت بعدی j در زمان t+1 است.به صورت ریاضی a_{ij}به این صورت تعریف می‌شود:

[1] Markov Chain
[2] Order
[3] Rainy
[4] Sunny

$$a_{ij} = P(\omega_{t+1} = j | \omega_t = i) \qquad (۶)$$

به‌طوری‌که ω نشان‌دهنده حالت مدل در زمان t است.

جدول ۸: ماتریس انتقال

$a_{ij} =$

	R	S
R	0.4	0.6
S	0.4	0.6

حال می‌دانیم حالت بعدی این دنباله یا $S \rightarrow R$ خواهد بود یا $S \rightarrow S$. با توجه به ماتریس انتقال خواهیم داشت یا $P_s.P_{SS} = 1.a_{SS} = 0.6$ و یا $P_s.P_{SR} = 1.a_{SR} = 0.4$. یعنی احتمال ۶۰٪ وجود دارد که در روز دوازدهم نماد بعدی S و تنها ۴۰ درصد احتمال وجود دارد که نماد بعدی R باشد. بسته به کاربرد پیش‌بینی کننده ممکن است تنها پیش‌بینی کننده تنها نمادهایی با بالاترین احتمال را برگردانند یعنی به عبارتی سیاست پیش‌بینی محتمل ترین را پیاده کنند یا اینکه ممکن است نمادهایی که m احتمال بالاتر را دارا هستند را برگردانند. در این صورت سیاست پیش‌بینی m-بالا[1] اجرا شده است که m پارامتری است که به صورت دلخواه تعیین می‌شود.

همان طور که دیدیم این نتایج کاملاً بر اساس رفتارهای گذشته نتیجه‌گیری شده است ولی عملاًمحیط‌هایی مانند موسیقی، متن و یا سایبر چندان چنین ویژگی را ندارند. یعنی این قدر سخت علت و معلولی نیستند. حال سؤال این است که آیا با وجود انسان و قدرت خلاقیت آن به طور کلی پیش‌بینی ممکن خواهد بود؟ جواب سؤال این است که درست است که انسان دارای خلاقیت است ولی در صورتی که مشاهدات زیاد باشد خیلی از الگوهای رفتاری پنهانی انسان را می‌توان به دست آورد. مثلاً درست است که یک آهنگساز می‌تواند آهنگ‌های مختلفی ایجاد کند ولی به‌هرحال یک استانداردها و قواعدی برای این کار وجود دارد. در مورد نوشتن نیز قواعد گرامری بر جملات حکم‌فرماست و همچنین در حوزه سایبر که هر حمله دارای گام‌های اصلی مشخصی است.

بعضی از پیش‌بینی کننده‌های مارکوفی ثابت هستند. به‌عبارتی‌دیگر طبق رابطه(۷)مدل تنها تا یک مرتبه ثابت از پیش تعیین‌شده(L=k)K ایجاد می‌شود. این کار به دلیل کاهش اندازه و پیچیدگی مدل انجام می‌شود. این نوع پیش‌بینی کننده‌های مارکوفی، در اصطلاح زنجیره‌های مارکوف با طول ثابت[2] یا پیش‌بینی کننده‌های مرتبه k نامیده می‌شوند. این ساختار به دلیل عدم پویایی و مشخص نبودن چگونگی مقداردهی بهینه مقدار K ، چندان مفید فایده

[1] Top-m Prediction Policy
[2] Fixed-length Markov Chains

نیست.دسته دیگری از پیش‌بینی کننده‌های مارکوفی، پیش‌بینی کننده‌هایی هستند کـه دیگـر درآنها طول حافظه ثابت نیست و مقدار آن تابعی از مقادیر گذشته‌اش می‌باشد. به عبارت دیگر مقدارL در رابطه(۷)می‌تواند از ۱ تا t متغیر باشد. به این نـوع پـیش‌بینـی کننـده در اصـطلاح زنجیره‌های مارکوف با طول متغیر گفته می‌شود. این نوع پـیش‌بینـی کننـده‌هـا ممکـن اسـت دارای یک کران بالایی باشندیا نباشند.

$$p(x_{t+1} \mid x_t, x_{t-1}, \ldots\ldots x_{t-L+1}) \qquad (۷)$$

دو ویژگی در مورد تعداد مرتبه‌ها در مدل مارکوف وجود دارد. هر چقدر مرتبـه‌هـا افـزایش می‌یابد تعداد تکرارها کمتر می‌شود و خیلی به‌سختی یک دنباله بیشتر از یک‌بار دیده می‌شـود. به عبارت دیگر خیلی جزئی می‌شوند. از طرفی هم مرتبه‌های پایین‌تر معمولاً تعـداد تکرارشـان بیشـتر اسـت و همـین طـور خیلـی کلـی‌انـد . اینجـا اسـت کـه مـدلهای مـارکوف بـا طـول متغیر(VLMM) می‌توانند کار ساز باشند. مدلهای مارکوف با طول متغیـر کـه از زنجیـره‌هـای مارکوف با طول متغیر بهره می‌گیرند از مخلوط کردن[1] مدلهای مارکوف با مرتبه‌هـای مختلـف ایجاد می‌شوند. یک VLMM ساده این است که وزن‌های ثابتی را به هر مرتبه مدل احتصـاص بدهیم و نتایج را باهم ترکیب کنیم ولی روش پیچیده‌تر آن این است که این وزن‌ها حالت پویـا داشته باشند و بسته به اهمیتشان، هر مرتبه، وزن متفاوتی را بگیرد. تعداد زیادی از الگـوریتم ها وجود دارند که با روش VLMM کار می‌کنند. از جمله این الگوریتم ها می‌توان به الگـوریتم های درخـت‌هـای پسـوندی احتمـالی[2](PST)[۴۹]، وزن دهـی درخـت محتـوا[3](CTW)[۵۰]و پیش‌بینی با استفاده از تطبیق جزئی[4](PPM)[۵۱] اشاره کرد. فرکانس صفر و عـدم یکنـواختی دو مسئله مهم در الگوریتم های مارکوف با طول متغیر هستند که نحوه برخورد هر یک از ایـن الگوریتم ها با این دو مسئله در واقع ساختار محاسباتی آن‌ها را شکل می‌دهد. در ادامه به بیان این دو مسئله پرداخته وراه حل پیشنهادی در مورد آن‌ها که همان روش یکدست‌سازی[5] اسـت شرح داده خواهد شد.

۴-۷-۱-۲-۱ عدم یکنواختی

اولین مسئله، مسئله‌ای به نام عدم یکنواختی است. در PPM این مسئله برای مدل‌هـای بـا مرتبه بالا مطرح است. به‌طور خلاصه این مسئله می‌گوید که هـر چقـدر مرتبـه مـدل و عمـق درخت بیشتر می‌شود تعداد تکرار در آن کمتر شده و احتمالات این دنباله‌ها مقادیر کمتـری را

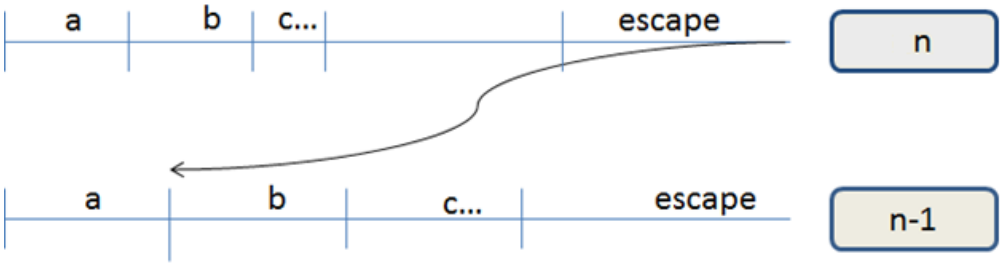
[1] blending
[2] Probabilistic Suffix Tree
[3] Context Tree Weighting
[4] Prediction by Partial Match
[5] Smoothing

خواهند گرفت. از طرف دیگر هم اطلاعات در سطح پایین‌تر(عمق کمتر) هم اغلب خیلی کلی‌اند و چون احتمالات بیشتری نیز دارند باعث می‌شوند که پیش‌بینی‌ها و الگوهای تکراری مدام استفاده شود و سیستم تنها به یک الگوهای خاصی حساس شود.

در واقع ما به سطوح پایین‌تر برای تولید مواد جدید و به سطوح بالاتر برای تولید ترکیبات مختلف و مشخصه سازی آن‌ها نیازمندیم. در حقیقت در این جا به یک نوع روش متعادل‌سازی نیازمندیم.

۴-۷-۱-۲-۲- مسئله فرکانس صفر[1]

این مسئله می‌گوید هنگامی که مرتبه‌ها افزایش می‌یابند دنباله‌هایی دیده می‌شوند که تا پیش از این دیده نشده است. در این حالت الگوریتم به ناچار به این دنباله که تابه‌حال(در دنباله‌های آموزشی پیشین)دیده نشده است،احتمال صفر می‌دهد. دادن احتمال صفر سبب می‌شود که این الگوی جدید به طور کامل نادیده گرفته شود. همین طور می‌دانیم که هر چقدر هم پایگاه داده ما بزرگ باشد باز این مشکل ممکن است اتفاق بیفتد. برای حل این مشکل مفهومی به نام احتمال گریز[2] معرفی شد. احتمال گریز، احتمال مواجه‌شدن با یک کاراکتر جدید(که در مرتبه‌های پیش تر دیده نشده است) است. دلیل نام‌گذاری این پارامتر این است که در انجام پیش‌بینی با استفاده از اطلاعات گذشته(درخت پسوندی از پیش ایجادشده)، هنگامی که شروع به جستجو در مورد یک محتوی در مرتبه‌های مختلف این درخت می‌نماییم، تعدادی از مرتبه‌های درخت هیچ پیش‌بینی در مورد این محتوی نمی‌توانند بکنند. در این مواقع یا می‌بایست یک احتمال صفر برگردانده شود که در این حالت حضور آن مرتبه نادیده گرفته خواهد شد و یا مقداری غیر صفر برگردانده شده و سپس برای تعیین پیش‌بینی به یک محتوی کوچک‌تر "گریز" کرده و دوباره به بررسی درخت در سطحی دیگر برای پیش‌بینی در مورد این محتوی بپردازیم. این فضای احتمال همان طور که در شکل ۳-۱۸ دیده می‌شود، در هر سطح معمولا با توجه توزیع احتمالی نمادهای دیگر درهمان سطح تعیین می‌شود.



شکل ۳-۱۸: احتمال گریز

[1]Zero Frequency Problem
[2]Escape Probabilities

۴-۷-۱-۲-۳- یکدست‌سازی[1]

در دو بخش پیشین به دو مسئله مهم در الگوریتم‌های با طول متغیر اشاره کردیم. در مدل‌سازی زبانی آماری، مجموعه‌ای از تکنیک ها که به یک دست سازی شناخته می‌شوند برای حل این مسائل پیشنهادشده‌اند. این پروسه اغلب با استفاده از یک ترکیب خطی از پیش‌بینی‌های ایجادشده از محتوی‌های با طول متفاوت انجام می‌شود. یکدست‌سازی، همانطور که در رابطه (۸)می‌بینید پروسه‌ای است که پیش‌بینی‌های مختلف ارائه‌شده توسط سطوح مختلف درخت را با استفاده از ضرایبی مشخص باهم ترکیب می‌کند. در این رابطه s نماد جدید، C نماد محتوی، λ نماد ضریب وزنی برای هر سطح و $p_i(.)$ نماد توزیع احتمالی ایجادشده در هر سطح می‌باشد. همچنین در این رابطه مجموع ضرایب باید برابر یک باشد($\sum_i \lambda_i(C)=1$)تا این که در نهایت توزیع‌های احتمالی نرمال شده‌ای به دست آید. نام این روش از این حقیقت می‌آید که این تکنیک تمایل دارد تا توزیع‌های احتمالی را به وسیله تطبیق دادن احتمالات پایین مانند احتمالات صفر به سمت بالا و احتمالات بالا به سمت پایین یکنواخت‌تر کند.

$$\mathrm{P}_n(s\,|C)=\lambda_n(C)\,p_n(s\,|C)+\lambda_{n-1}(C)p_{n-1}(s\,|C)+....+\lambda_0(C)p_0(s\,|C) \qquad (۸)$$

تمرکز اصلی این فن حل مسئله فرکانس صفر است ولی به همراه آن مشکل عدم یکنواختی نیز تا حدودی حل می‌شود. دو فن مشهور یکدست‌سازی، یکدست‌سازی عقب گردی[2] و یکدست‌سازی درون‌یابی شده[3] است. در هر دو این روش‌ها از دو پارامتر α که تخمین‌های احتمالی را محاسبه می‌کند و eکه نماد احتمال گریز است برای محاسبه احتمال پیش‌بینی رویداد x_{i+1} استفاده می‌شود.

با فرض یک دنباله آزمایشی، مدل‌های عقب گردی از بیش‌ترین مرتبه در درخت کار خود را آغاز می‌کنند و به دنبال یک دنباله که با این دنباله منطبق باشد و تعداد تکرار آن از یک مقدار معین بیشتر باشد می‌گردد. اگر هیچ تطبیقی صورت نگرفت، اولین عضو دنباله حذف‌شده و این دنباله آزمایشی جدید در n-gramهای با یک اندازه کوچک‌تر جستجو می‌شود. این پروسه به همین شکل ادامه می‌یابد تا یک انطباق مناسب پیدا شود. چارچوب ریاضیاتی این فن را در رابطه(۹) می‌بینید:

$$p(x_{i+1}\,|\,x^{\,i}_{(i-n)+1})=\begin{cases}\alpha(x_{i+1}\,|\,x^{\,i}_{(i-n)+1}) & if\;\; c(x_{i+1}\,|\,x^{\,i}_{(i-n)+1})>0\\ e(x^{\,i}_{(i-n)+1})p(x_{i+1}\,|\,x^{\,i}_{(i-n)+2}) & if\;\; c(x_{i+1}\,|\,x^{\,i}_{(i-n)+1})=0\end{cases} \qquad (۹)$$

[1] Smoothing
[2] Back-off Smoothing
[3] Interpolated Smoothing

در این رابطه اگر تعداد تکرار x_{i+1}، یک مقدار غیر صفر باشد(یعنی $c(x_{i+1}|x^{i}_{(i-n)+1})>0$) سپس تخمین احتمالی $\alpha(x_{i+1}|x^{i}_{(i-n)+1})$استفاده می‌شود. در غیر این صورت یک مقدار کوچک به نام احتمال گریز($e(x^{i}_{(i-n)+1})$) برگشت داده‌شده و برای بررسی احتمال x_{i+1}به یک مرتبه محتوی کوچک‌تر مراجعه می‌کنیم.

درنهایت اگر این گام‌های برگشتی به مقدار مشخصی نرسید مرتبه نهایی مرتبه‌ای به نام مرتبه ۱- خواهد بود که در آن احتمال همه حروف الفبا یکسان است(از توزیع احتمالی یکنواخت استفاده می‌شود).

در مدل‌های درون‌یابی احتمال یک n-gram به‌وسیله محاسبه بازگشتی یک ترکیب وزنی از توزیع مرتبه n-1 و n-2 به دست می‌آید. چارچوب ریاضیاتی این فن را در رابطه (۱۰) می‌بینید:

$$p(x_{i+1}|x^{i}_{(i-n)+1})=\alpha(x_{i+1}|x^{i}_{(i-n)+1})+e(x^{i}_{(i-n)+1})p(x_{i+1}|x^{i}_{(i-n)+2}) \qquad (۱۰)$$

البته در رابطه (۱۰) مقدار α شامل یک مضرب $1-e(x^{i}_{(i-n)+1})$نیز می‌باشد.

تفاوت اصلی میان مدل‌های عقب‌گرد و درون‌یابی در چگونگی استفاده از توزیع‌های مرتبه پایین‌تر است. مدل‌های درون‌یابی از اطلاعات توزیع‌های مرتبه پایین‌تر در همه حالات بهره می‌گیرند درحالی‌که مدل‌های عقب گردی تنها زمانی از اطلاعات توزیع‌های مرتبه پایین‌تر استفاده می‌کنند که تکرار دنباله به طول n، صفر باشد.

در ادامه به یکی از الگوریتم‌های با طول متغیر به نام PPMکه از فن یکدست‌سازی عقب گردی در الگوریتمش بهره می‌گیرد خواهیم پرداخت.

PPM -۳-۱-۷-۳

به‌طور کلی ارتباط مستقیمی بین پیش‌بینی دنباله‌های گسسته و الگوریتم‌های فشرده‌سازی بدون اتلاف وجود دارد به‌طوری‌که هر الگوریتم فشرده‌سازی بدون اتلاف می‌تواند برای پیش‌بینی و بالعکس هر الگوریتم پیش‌بینی برای فشرده‌سازی بدون اتلاف استفاده شود. بهترین روش ازنظر عملکرد در فشرده‌سازی برای محاسبه تخمین‌های احتمالی دنباله‌های بر خط با استفاده از یک مدل درخت پسوندی، تکنیک مخلوط سازی استفاده‌شده توسط PPM است[۵۲]. استراتژی‌های مخلوط سازی را نیز می‌توان به سه دسته تقسیم کرد:

مخلوط سازی کامل: این روش در پی این است که از همه محتوی‌های موجود استفاده کند. مسلما این روش زمان محاسباتی زیادتری را به خود اختصاص می‌دهد . به این روش، روش درون‌یابی نیز گفته می‌شود.

استثنا[1]: در این روش مخلوط سازی از بلندترین محتوایی که تطبیق می‌کند برای پیش‌بینی یک رویداد استفاده می‌کند. اگر این محتوی رویداد موردنظر را با احتمال غیر صفر

[1] Exclusion

پیش‌بینی کرد، این مقدار برای میزان احتمال وقوع آن رویداد استفاده می‌شود. وگرنه مقداری (احتمال گریز) برگردانده شده و دومین بلندترین محتوی برای پیش‌بینی آن رویداد استفاده می‌شود. این روند تا آنجا که پیش‌بینی انجام شود گریز به محتوی کوچک‌تر را انجام می‌دهد. درنهایت مرتبه -۱ تضمین می‌کند که مقداردهی سرانجام خواهد شد. دلیل اصلی نام‌گذاری این روش این است که هنگامی که مرتبه‌های بالاتر در مورد یک رویداد پیش‌بینی را انجام می‌دهند دیگر پیش‌بینی‌های مرتبه‌های پایین‌تر را در نظر نمی‌گیرد یا به‌اصطلاح آن‌ها را استثنا می‌کند[۵۳]. دلیل این کار این است که سطوح پایین‌تر دارای اطلاعات جزئی بیشتری هستند و به نظر آن‌ها برای پیش‌بینی بسنده می‌شود. به این روش، روش عقب گردی نیز گفته می‌شود. همچنین در این روش تعداد رخداد یک رویداد به شرطی افزایش می‌یابد که این رویداد توسط مرتبه‌های بالاتر پیش‌بینی نشده باشد.

استثناء تنبل گونه[1]: این روش از همان مکانیزم گریز و شناسایی بلند ترین محتوی به‌مانند روش استثنا برای پیش‌بینی استفاده می‌کند با این تفاوت که دیگر تعداد آن‌ها را به این دلیل که مرتبه‌های بالاتر آن را پیش‌بینی کرده اند مستثنی نمی‌کند[۵۳].

پیش‌بینی به وسیله تطبیق جزیی، یک طرح فشرده‌سازی داده است که ایده اصلی آن استفاده از یک الگوریتم مبتنی بر یک دست سازی عقب‌گردی برروی توزیع‌های n-gram است. طرح‌های مختلفی در PPM با توجه به تفاوت محاسبه α, و e وجود دارد. در تمام روابط زیر پارامتر α، احتمال وزن دهی شده در یک سطح معین برای یک نماد جدید را نشان می‌دهد. به عبارت دیگر α یک احتمال است که در عبارت $(1-e_i)$ ضرب شده است و مقدار احتمال گریز فضای احتمالی مربوط به آن در محاسبات اعمال‌شده است. در ادامه به تعدادی از آن‌ها اشاره خواهیم کرد:

روش A: در این روش مقدار احتمال گریزی که به نمادهای جدید اختصاص می‌یابد و همچنین مقدار α به صورت زیر محاسبه می‌شود:

$$e(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{1}{\sum_{x \in \Omega} c(x^{i}_{(i-n)+1}) + 1} \qquad (۱۱)$$

$$\alpha(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{c(x_{i+1} \mid x^{i}_{(i-n)+1})}{\sum_{x \in \Omega} c(x \mid x^{i}_{(i-n)+1}) + 1} \qquad (۱۲)$$

هر چقدر تعداد رویدادهای یک محتوی افزایش می‌یابد احتمال گریز کوچک و کوچک‌تر می‌شود. به عبارت دیگر بعدازاینکه یک تعداد کاراکتر در یک محتوی ظاهر می‌شود، احتمال اینکه یک کاراکتر کاملا جدید ظاهر شود خیلی کم می‌شود. همچنین میزان احتمال

[1] Lazy Exclusion

$\alpha(x_{i+1}|x^{i}_{(i-n)+1})$ به تخمین ML نزدیک‌تر می‌شود.

روش B: در رابطه (۵)، $q(x^{i}_{(i-n)+1})$ تعداد کل نمادهای متفاوتی که با تعداد تکرار غیر صفر در محتوی $x^{i}_{(i-n)+1}$ اتفاق افتاده است را نشان می‌دهد. در این رابطه $q(x^{i}_{(i-n)+1})$رابطه مستقیم با $e(x_{i+1}|x^{i}_{(i-n)+1})$ دارد. این بدین معناست که اگر تعداد کاراکترهای متفاوت کمتری پیش از این دیده شده است، احتمالش کم است که در آینده یک کاراکتر جدید هم دیده شود.در این روش به صورت پیش فرض یک نماد هنگامی که در یک محتوی معین روی می‌دهد به عنوان یک رویداد جدید در نظر گرفته می‌شود و با احتمال صفر پیش‌بینی می‌شود مگر اینکه قبلا آن نماد در آن محتوی دو بار اتفاق افتاده باشد. به عبارت دیگر این روش زمانی یک مشاهده را به صورت کارا به کار می‌گیرد که آن نماد ۲ بار اتفاق افتاده باشد. این اقدام ازاینجا ناشی می‌شود که رویدادی که فقط یکبار ظاهرشده است و از بین رفته است ممکن است یک خطا باشد. این فکر در رابطه (۶) با تفریق یک از تعداد نمادها در هنگام محاسبه $\alpha(x_{i+1}|x^{i}_{(i-n)+1})$ اعمال می‌شود.

$$e(x_{i+1}|x^{i}_{(i-n)+1}) = \frac{q(x^{i}_{(i-n)+1})}{\sum_{x\in\Omega} c(x^{i}_{(i-n)+1})} \qquad (۱۳)$$

$$\alpha(x_{i+1}|x^{i}_{(i-n)+1}) = \frac{c(x_{i+1}|x^{i}_{(i-n)+1})-1}{\sum_{x\in\Omega} c(x^{i}_{(i-n)+1})} \qquad (۱۴)$$

مدل PPM که از این روش بهره می‌گیرد به نام PPMB شناخته می‌شود.

روش C: این روش در واقع توسعه داده‌شده روش A است و به‌نوعی برای ترکیب کردن روش های A و B طراحی شده است:

$$e(x_{i+1}|x^{i}_{(i-n)+1}) = \frac{q(x^{i}_{(i-n)+1})}{\sum_{x\in\Omega} c(x^{i}_{(i-n)+1}) + q(x^{i}_{(i-n)+1})} \qquad (۱۵)$$

$$\alpha(x_{i+1}|x^{i}_{(i-n)+1}) = \frac{c(x_{i+1}|x^{i}_{(i-n)+1})-1}{\sum_{x\in\Omega} c(x^{i}_{(i-n)+1}) + q(x^{i}_{(i-n)+1})} \qquad (۱۶)$$

در این روش مقدار احتمال گریز هر هنگام که نوع جدیدی مشاهده می‌شود افزایش می‌یابد ولی این افزایش مقدار به دلیل اضافه شدن در مخرج، به مقدار کمتری خواهد بود. در این مدل که به نام PPMC شناخته می‌شود به‌روزرسانی استثنائی نیز انجام می‌شود. در این روش عمل به‌روزرسانی محتوی‌های مرتبه پایین‌تر به شرطی انجام می‌شود که این سطوح بعد از یک گریز ملاقات شوند. به عبارت دیگر هیچ محتوی مرتبه بالاتری با داده آموزشی تطبیق پیدا نکند. در حقیقت در این استراتژی به‌روزرسانی، زمانی تعداد شمارش یک رویداد در یک محتوی افزایش

می‌یابد که آن رویداد در محتوی‌های با مرتبه بالاتر پیش‌بینی نشده باشد. این استراتژی در واقع در راستای همان سیاست استثنا است.

روش D: این روش که به نام PPMD نیز شناخته می‌شود، عملکردی شبیه به روش B دارد با این تفاوت که از مقدار ۰٫۵ در دو رابطه استفاده شده است. در این روش هنگامی که یک کاراکتر جدید برای اولین بار اتفاق می‌افتد ۰٫۵ به‌جای ۱ به صورت رابطه احتمال گریز اضافه می‌شود:

$$e(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{\frac{1}{2} q(x^{i}_{(i-n)+1})}{\sum_{x \in \Omega} c(x^{i}_{(i-n)+1})} \qquad (۱۷)$$

$$\alpha(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{c(x_{i} \mid x^{i}_{(i-n)+1}) - \frac{1}{2}}{\sum_{x \in \Omega} c(x^{i}_{(i-n)+1})} \qquad (۱۸)$$

روش P :پنجمین روش، روش P است. ایده این روش از این فرض نشات می‌گیرد که رویدادهای جدید بر طبق یک مدل پروسه پواسون اتفاق می‌افتند. ویتن و بل در این رابطه را پیشنهاد کردند:

$$e(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{q_1(x_{i+1} \mid x^{i}_{(i-n)+1})}{\sum_{x \in \Omega} c(x \mid x^{i}_{(i-n)+1})} - \frac{q_2(x_{i+1} \mid x^{i}_{(i-n)+1})}{\left(\sum_{x \in \Omega} c(x \mid x^{i}_{(i-n)+1})\right)^2} + \frac{q_3(x_{i+1} \mid x^{i}_{(i-n)+1})}{\left(\sum_{x \in \Omega} c(x \mid x^{i}_{(i-n)+1})\right)^3} + ... \qquad (۱۹)$$

در ادامه می‌توان این رابطه را به صورت رابطه تقریب زد:

$$e(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{q_1(x_{i+1} \mid x^{i}_{(i-n)+1})}{\sum_{x \in \Omega} c(x \mid x^{i}_{(i-n)+1})} \qquad (۲۰)$$

یکی از ضعف‌های این دو روش این است که هنگامی که $q_1(x_{i+1} \mid x^{i}_{(i-n)+1}) = 0$ یا $\sum_{x \in \Omega} c(x \mid x^{i}_{(i-n)+1}) = q_1(x_{i+1} \mid x^{i}_{(i-n)+1})$ شود، تخمین احتمالی به ترتیب صفر یا یک خواهد شد. موفات یک راه حل برای این مشکل پیشنهاد کرد:

$$e(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{q_1(x_{i+1} \mid x^{i}_{(i-n)+1}) + 1}{\sum_{x \in \Omega} c(x \mid x^{i}_{(i-n)+1}) + 1} \qquad (۲۱)$$

$$\alpha(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{q_1(x_{i+1} \mid x^{i}_{(i-n)+1})}{\sum_{x \in \Omega} c(x \mid x^{i}_{(i-n)+1}) + 1} \qquad (۲۲)$$

به عبارت دیگر به سادگی به صورت و مخرج احتمال گریز ۱ اضافه می‌کند این روش به نام

روش PPMP نیز شناخته می‌شود

روش S:

همانطور که پیش تر اشاره کردیم یک روش مرسوم برای اجرای یکدست‌سازی استفاده از نمادهای گریز است. در هر طول مختلف از محتوی، توزیع‌های پیش‌بینی کننده یک احتمال معینی را به یک نماد اضافی به نام نماد گریز اختصاص می‌دهند که به آن احتمال گریز گفته می‌شود. این مقدار احتمالی مطابق با پیش‌بینی‌ها در یک سطح پایین‌تر یعنی سطح پدر آن سطح اختصاص می‌یابد. کووانس در سال ۲۰۰۶ [۵۴] نوع عمومی‌سازی بر روی روش PPMA ارائه کرد و این روش را PPMA عمومی‌سازی شده[1] نامید. در این روش رابطه احتمال گریز ارائه‌شده در روش PPMA به صورت زیر تبدیل می‌شود:

$$e(x_{i+1} \mid x^{i}_{(i-n)+1}) = \frac{\wp}{\sum_{x \in \Omega} c(x^{i}_{(i-n)+1}) + \wp} \qquad (۲۳)$$

هنگامی که $\wp = 1$ شود PPMA و هنگامی که $\wp = q(x^{i}_{(i-n)+1})$ PPMC حاصل می‌شود. ازآنجاکه مقدار احتمال اختصاص داده‌شده به نماد گریز ، در مقدار پیش‌بینی سطح فرزند ضرب می‌شود، مقدار احتمال گریز را می‌توان به واقع سهم مشارکت احتمال پیش‌بینی فرزند در احتمال پیش‌بینی پدر دانست. این ویژگی در روش PPMA که تنها به تعداد تکرار نمادها در سطح فرزند توجه می‌شود به دست نمی آید. ابتدا به ویژگی زیر توجه نمایید:

هنگامی که محتوای فرزند توزیع‌های احتمالی مشابهی با توزیع‌های احتمالی مربوط به سطح پدر ارائه می‌کند در این موارد ما علاقه داریم که بیشتر به محتواهای کوچک‌تر(محتوای پدر) مراجعه کنیم. زیرا محتواهای کوچک‌تر شامل اطلاعات آماری کلی بیشتری نسبت به محتواهای بلندتر هستند. در این مواقع مقدار $\wp$ می‌بایست کوچک‌تر شود تا تاکید بر محتواهای کوچک‌تر باشد(به عبارتی مقدارe بزرگ‌تر شود). بالعکس هنگامی که محتوای فرزند توزیع احتمالی متفاوتی نسبت به توزیع پیشنهادی پدر در مورد نمادهای بعدی دارد در این موارد بهتر است که به محتواهای بلندتر(محتوی فرزند) مراجعه شود زیرا محتواهای بلندتر جزئی‌تر هستند و امکان تمییز دادن بین پیشنهادات را بیشتر فراهم می‌کند. در این مواقع مقدار $\wp$ می‌بایست بزرگ‌تر شود تا تاکید بر محتوای بلندتر شود (در واقع مقدار eکوچک‌تر شده و احتمال کمتری به احتمال پیشنهادی پدر داده شود)

شفلر در سال ۲۰۰۸[۵۵] پیشنهاد کرد که مقدار $\wp$ به صورت زیر محاسبه شود:

$$\wp = \frac{\sigma}{d(\alpha(x_{i+1} \mid x^{i}_{(i-n)+1}), \alpha(x_{i} \mid x^{i-1}_{(i-n)+1}))} \qquad (۲۴)$$

[1] PPMA Generalised

d در این عبارت میزان فاصله میان دو توزیع را محاسبه می‌کند. دو روش واگرایی کالبک-لیبلر[1] و روش p-norms برای این کار پیشنهاد شده است. مقدار σ نیز یک مقدار ثابت است. وی نام این روش را PPMS نامید.

ولی این روش دارای یک ضعف است زیرا عملاً در حالت‌هایی ویژگی موجود در روش PPMA که به بیان رابطه احتمال گریز و تعداد نمادهای در سطح فرزند می پرداخت از بین می‌رود. به بیان دقیق‌تر با جایگزینی رابطه (۲۴) در رابطه (۲۳) خواهیم داشت:

$$e(x_{i+1}|x^{i}_{(i-n)+1}) = \frac{\sigma}{d(\alpha(x_{i+1}|x^{i}_{(i-n)+1}),\alpha(x_i|x^{i-1}_{(i-n)+1}))} = \frac{\sigma}{\sum_{x\in\Omega} c(x^{i}_{(i-n)+1}) + \frac{\sigma}{d(\alpha(x_{i+1}|x^{i}_{(i-n)+1}),\alpha(x_i|x^{i-1}_{(i-n)+1}))}} \quad (۲۵)$$

$$\frac{\frac{\sigma}{d(\alpha(x_{i+1}|x^{i}_{(i-n)+1}),\alpha(x_i|x^{i-1}_{(i-n)+1}))}}{\frac{d(\alpha(x_{i+1}|x^{i}_{(i-n)+1}),\alpha(x_i|x^{i-1}_{(i-n)+1}))\sum_{x\in\Omega} c(x^{i}_{(i-n)+1}) + \sigma}{d(\alpha(x_{i+1}|x^{i}_{(i-n)+1}),\alpha(x_i|x^{i-1}_{(i-n)+1}))}} = \frac{\sigma}{d(\alpha(x_{i+1}|x^{i}_{(i-n)+1}),\alpha(x_i|x^{i-1}_{(i-n)+1}))\sum_{x\in\Omega} c(x^{i}_{(i-n)+1}) + \sigma}$$

در این حالت در مواردی که مقدار d برابر صفر باشد، نقش پارامتر c به طور کلی نادیده گرفته می‌شود. پیشنهاد ما این است که رابطه (۲۳) به صورت زیر تغییر نماید:

(۲۶)

$$e(x_{i+1}|x^{i}_{(i-n)+1}) = \frac{d(\alpha(x_{i+1}|x^{i}_{(i-n)+1}),\alpha(x_i|x^{i-1}_{(i-n)+1}))}{\sum_{x\in\Omega} c(x^{i}_{(i-n)+1}) + d(\alpha(x_{i+1}|x^{i}_{(i-n)+1}),\alpha(x_i|x^{i-1}_{(i-n)+1}))}$$

در واقع رابطه (۲۶) به بیان میزان تفاوت توزیع احتمالی یک محتوا نسبت به محتوای پدرش علاوه بر تاثیر تعداد نمادها در سطح فرزند پرداخته است. هر چه تفاوت توزیع احتمالی میان فرزند و پدر بیشتر باشد، مقدار $\wp = d(\alpha(x_{i+1}|x^{i}_{(i-n)+1}),\alpha(x_i|x^{i-1}_{(i-n)+1}))$ کمتر شده و در واقع به محتوی‌های بلندتر توجه می‌شود و هرچه شباهت بیشتر باشد مقدار $\wp$ بیشتر شده و به محتوی کوچک‌تر توجه بیشتری خواهد شد.

ما از این روش برای محاسبه احتمال گریز در این کار استفاده کرده ایم. همچنین در روش ما مقدار (1-e) در مقدار احتمال α، ضرب نشده است. به عبارت دیگر مقدار احتمال گریز تنها در تمام سطوحی که تکرار دنباله صفر است استفاده شده است.

[1] Kullback-Leibler Divergence

۳-۷-۱-۴- سنجه‌های تشابه[1]

سنجه‌های تشابه، درجه نزدیکی یا دوری اشیای مورد هدف را منعکس می‌کنند. روش های زیادی برای محاسبه میزان مشابهت میان دو توزیع احتمالی پیشنهاد شده است[۴۶]. در این مقاله ما برای محاسبه میزان مشابهت، از روش واگرایی کالبک-لیبلر استفاده می‌کنیم.

۴-۷-۱-۴-۱- واگرایی کالبک-لیبلر[2]

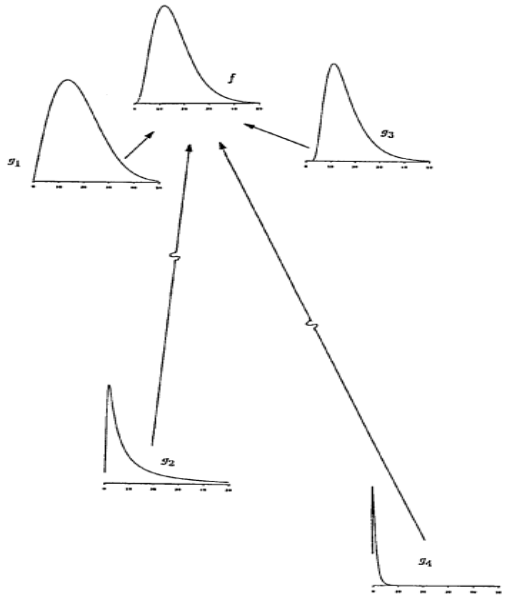
روش واگرایی کالبک-لیبلر (KL divergence)که به نام آنتروپی نسبی نیز شناخته می‌شود یک معیار معروف برای ارزیابی میزان تفاوت میان دو توزیع احتمالی است. با فرض دو توزیع احتمالی P و Q، واگرایی KL از توزیع P به توزیع Q به صورت زیر تعریف می‌شود:

$$D_{KL}(p\,||Q) = \sum_i P(i)\log(\frac{P(i)}{Q(i)}) \qquad (۲۷)$$

رابطه KL یک سنجه متقارن نیست و یک روش محاسبه یک طرفه است. به این معنا که سنجش از P به Q، مشابه سنجش از Q به P نیست. در این روش در پی این هستیم که میزان فاصله[3] میان یک مدل تقریبی (Q) و یک مدل حقیقی(P) را بیابیم. این معیار کمک می‌کند که میان چند مدل تقریبی از یک مدل حقیقی بتوانیم مدل تقریبی با کمترین فاصله نسبی و تخمینی را بیابیم. به عنوان مثال در شکل ۳-۱۹دانستن فاصله کالبک میان مدل های تقریبی g به مدل f به انتخاب نزدیک‌ترین مدل تقریبی نسبت به مدل f یاری می رساند[۵۶]. در واقع این رابطه می‌خواهد بگوید که اگر از توزیع Q به‌جای توزیع Pاستفاده شود به چه میزان هدر رفت اطلاعات خواهیم داشت.

[1] Similarity Measures
[2] Kullback–Leibler

[3] Distance



شکل ۳-۱۹: واگرایی کالبک-لیبلر

هنگام استفاده از این روش ممکن است با دو مشکل روبه‌رو شویم:

۱- زمانی که q_i یا p_i مساوی صفر باشد چه باید کرد؟

۲- می‌دانیم که P وQ دو توزیع هستند که بر روی رویدادهای متفاوتی تعریف‌شده‌اند. در این وضعیت چگونه می‌توان فاصله میان این دو توزیع را محاسبه کرد؟

سوال اول هنگامی که $p_i = 0$ یا $\frac{p_i = 0}{q_i = 0}$ است چون عبارت به صورت $p_i \log \frac{p_i}{q_i}$ است در هر دو حالت مقدار برابر صفر می‌شود. ولی مشکل همچنان در حالتی که $q_i = 0$ و $p_i != 0$ است باقی است. واگرایی در این حالت بی نهایت[1] خواهد شد. در مسئله دوم هم مشکلی مشابه با مسئله اول وجود دارد. زیرا همیشه رویدادهایی ممکن است در یک سطح دیده شوند ولی در سطح دیگر دیده نشده باشند. این دو مشکل را می‌توان با استفاده از روش کاهش تمام عیار[2] که یک نوع روش یکدست‌سازی است حل نمود.

برای دو توزیع احتمالی P و Q دو مجموعه S_p و S_q را به ترتیب به عنوان مجموعه رویدادهای مشاهده‌شده در توزیع P و مجموعه رویدادهای مشاهده‌شده در توزیع Q فرض می‌کنیم. توزیع احتمالی یکدست شده برای هر عضو S_p و S_q که به ترتیب با $p(i)'$ و $q(i)'$ نشان می دهیم طبق رابطه به دست می آید:

[1] Infinite
[2] Absolute Discounting

(۲۸)

$$p(i)' = \begin{cases} p(i) - pc & if\ i \in S_p \\ eps & otherwise\ i \in S_p / S_q \end{cases}$$

$$q(i)' = \begin{cases} q(i) - qc & if\ i \in S_q \\ eps & otherwise\ i \in S_q / S_p \end{cases}$$

$$eps = 0.1\ , Pc = eps * \frac{|S_u - S_p|}{|S_p|},\ qc = eps * \frac{|S_u - S_q|}{|S_q|}$$

که در رابطه بالا p(i) و q(i) احتمالات پیشین هر رویداد و $S_u = S_p \cup S_q$است. مقدار eps یک مقدار دلخواه کوچک (اینجا ۰,۱ فرض شده است) فرض شده است. عبارت $i \in S_p / S_q$ به معنای عنصری(i) که در S_p است و در S_q نیست. به عنوان مثال فرض کنید.

P: a=۱/۲, b=۱/۴, c=۱/۴

Q: a=۶/۱۲, b=۲/۱۲, d=۴/۱۲

که با عمل یکدست‌سازی خواهیم داشت:

$pc = qc = eps / 3 = 1/30,$

P: a= 1/2-1/30 =0.47 ,b=1/4-1/30= 0.22 ,c=1/4-1/30=0.22 ,d=1/30=0.03

Q: a=6/12-1/30= 0.47 ,b=2/12-1/30=0.13 ,c= 1/30=0.03 ,d=4/12-1/30=0.72

۳-۷-۲- درخت‌های پسوندی[1]

پیش تر گفتیم که می‌توان مدل مارکوف مرتبه اول را توسعه داده و n امین مدل مارکوف را تولید کرد. مدلی که در آن یک رویداد جدید به n رویداد پیشینش وابسته است. ولی با این توسعه مشکل جدیدی مطرح می‌شود. با افزایش مرتبه‌ها، ماتریس انتقال نیز ابعادش بزرگ‌تر می‌شود. برای یک مدل مارکوف با مرتبه n ام و با Nحالت، ماتریس انتقال a، N^nمقدار دارد.این مقدار داده و ابعاد بزرگ ماتریس،پیچیدگی محاسباتی و همچنین میزان فضای حافظه موردنیاز را بسیار افزایش می‌دهد. مخصوصاً زمانی که لازم باشد از چند مدل مارکوف استفاده کرده و داده‌ها را باهم ترکیب کنیم.

برای حل این مشکل درختان پسوندی معرفی شدند. ریشه این درخت، حالت مدل را در زمان t=0 نشان می‌دهد. در درخت پسوندی هر نماد به شکل یک گره که کنارش تعداد تکرار و احتمالش ذکرشده نمایش داده می‌شود. در هر سطح یک گره ممکن است والد یک یا چند گره

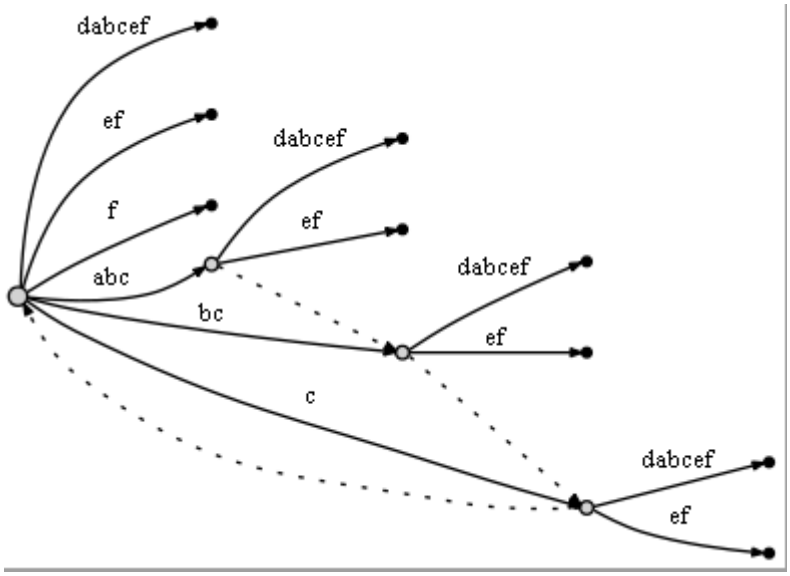
[1] Suffix Trees(ST)

دیگر باشد. سطح اول این درخت شامل نمادهای دیده‌شده در دنباله مورد نظر است و سطح دوم در واقع نشان‌دهنده همان bi-gramها است و همین طور تا آخر. در واقع با یک پیمایش از ریشه تا یک برگ می‌توان یک n-gram موجود در دنباله را مشاهده کرد. نکته مهم در این درخت این است که هر شاخه از درخت، نشان‌دهنده همه دنباله‌های منحصربه‌فرد حاضر در داده آموزشی است. ما در کارمان از درخت پسوندی آکونن[1] استفاده کرده ایم. از ویژگی‌های این درخت، کمتر بودن تعداد گره‌ها در آن است[۵۷]. نمونه‌ای از این درخت را در شکل ۳-۲۰ مشاهده می‌کنید. این درخت از روی ورودی abcdabcef ساخته‌شده است.در واقع به وسیله درخت پسوندی از ایجاد دنباله‌های تکراری و بیهوده جلوگیری می‌شود. از این بابت چون این درخت تنها شامل دنباله‌هایی است که تابه‌حال دیده‌شده، از نظر محاسباتی و همچنین فضای حافظه موردنیاز خیلی بهتر از مدل مارکوف با مرتبه n ام کار می‌کند که در آنجا از ماتریس‌ها برای این کارها استفاده می‌شد. اینجا نیز مانند ماتریس‌ها اگر بخواهیم احتمال یک bi-gram یا همان مدل مارکوف مرتبه اول را محاسبه کنیم خواهیم داشت:

$$P(AB)=P(A).P(B|A) \qquad (۲۹)$$

P(B|A) در رابطه (۲۹) در واقع همان نقش احتمال انتقال a_{ij} را ایفا می‌کند.

یک نکته مهم دیگر در این ساختارها این است که مدل مرتبه o شامل همه اطلاعاتی است که موردنیاز مدل مرتبه o-1 است.



شکل ۳-۲۰:درخت آکونن حاصل از رسم جمله abcdabcef

[1] Ukkonen suffix tree

۳-۸- جمع‌بندی

در این فصل معماری کلی کار ابتدا ارائه شد. سپس هر یک از این سطوح معماری (که مطابقت با مدل آگاهی وضعیتی) دارد، با جزئیات بیشتری شرح داده شد.

فصل ۴– شبیه‌سازی و نتایج

۴–۱–مقدمه

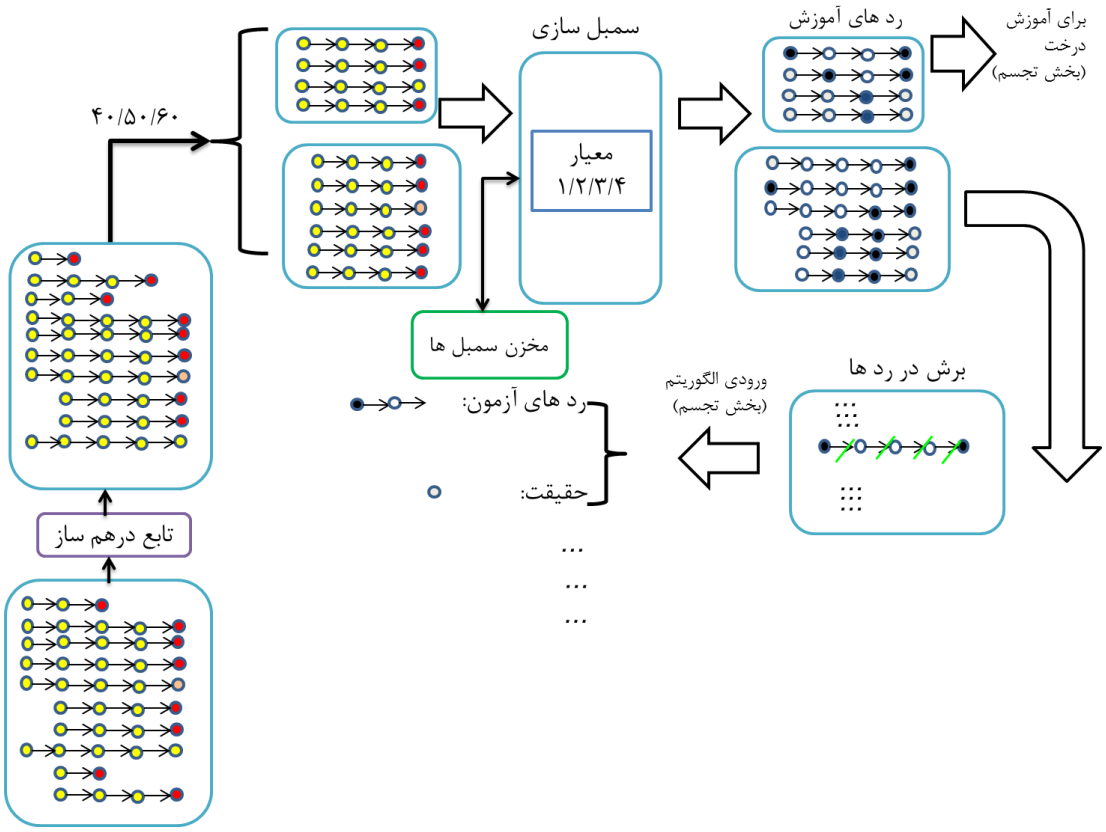
در این فصل به تشریح مراحل تولید داده آموزشی[۱] و آزمون[۲] و نتایج نهایی دو الگوریتم مارکوف با طول متغیر A وS خواهیم پرداخت.

۴–۲– فرآیند تولید داده آموزشی و آزمون

همانطور که در فصل سوم در شکل ۳-۱ که نشان دهده معماری کلی است، قابل مشاهده است، ردهای حمله(خطوط سبز و سیاه) وارد بخش تولید داده آموزشی و آزمون می‌شود. این بخش داده های موردنیاز برای تولید داده آموزشی و آزمون را به عهده دارد.

روند کلی تولید داده آموزش و آزمون در شکل ۴-۱ دیده می‌شود. ردهای حمله ابتدا با استفاده از یک تابع درهم سازی[۳]، به صورت تصادفی درهم می‌شوند. این عمل پنج بار در این کار تکرارشده است. سپس این مجموعه حاوی ردهای حمله در هم شده برای تقسیم‌بندی به مرحله دوم می رود.

[1] Train Data
[2] Test Data
[3] Shuffle



شکل ۴-۱: تولید داده های آموزشی و آزمون

برای تقسیم‌بندی ردهای حمله به دو بخش داده های آزمون و آموزشی، ما از سه تقسیم‌بندی مختلف استفاده کردیم.

- تقسیم‌بندی ۴۰-۶۰: ۴۰٪ داده آموزشی و ۶۰٪ داده آزمون
- تقسیم‌بندی ۶۰-۴۰: ۶۰٪ داده آموزشی و ۴۰٪ داده آزمون
- تقسیم‌بندی ۵۰-۵۰: ۵۰٪ داده آموزشی و ۵۰٪ داده آزمون

بنابراین با توجه به هر یک از این تقسیم‌بندی‌ها ما دو دسته داده خواهیم داشت. مرحله بعد وظیفه‌اش نمادسازی برای رویدادهای داخل هر رد حمله است. این کار برای ساده تر کردن عملیات‌ها و کاهش زمان و فضای موردنیاز در زمان پیاده‌سازی انجام می‌شود. ما از چهار معیار زیر برای تولید نمادها استفاده کردیم:

- معیار۱: تولید نماد بر اساس برچسب طبقه‌بندی
- معیار۲: تولید نماد بر اساس برچسب زیر طبقه‌بندی
- معیار۳: تولید نماد بر اساس برچسب بهره‌کشی
- معیار ۴: تولید نماد بر اساس توصیفات رویدادها

در واقع ما به ازای هر تقسیم‌بندی، چهار دسته رد حمله (با نمادهایی متفاوت) خواهیم

داشت. به عنوان مثال اگر رد حمله ما رد نشان داده‌شده در شکل ۳-۱۶ باشد، و بخـواهیم ایـن رد را بر اساس معیار زیر طبقه‌بندی سمبل سازی کنیم، رد زیر ایجاد خواهد شد:

1,2,#,

ما در سمبل سازی از اعداد برای این کار استفاده کردیم که هـر رد انتهـایش بـا علامـت # مشخص می‌شود. هر سمبل نیز با علامت ویرگـول جـدا مـی‌شـود. بـدیهی اسـت در ایـن گـام می‌بایست نمادهای اختصاص‌یافته به هر یک از رویدادها به واسطه هر معیـار، درجـایی ذخیـره شود. مخزن نمادها وظیفه ذخیره‌سازی این نمادها را به عهده می‌گیرد.

حال ما دو دسته رد حمله داریم که بـا نمادهـایی نمادگـذاری شـده انـد. ردهـای آمـوزش تولیدشده آماده استفاده هستند و جهت آموزش درخت، به بخش تجسم فرستاده می‌شوند. ولی ردهای آموزش می‌بایست مراحل دیگری را هم طی کنند. این ردها وارد بخش دارایی می‌شوند.

مرحله چهارم، وظیفه‌اش برش زدن در هر رد حمله است. در هر رد حمله، ما در چهار نقطه آن عمل برش زدن را انجـام مـی‌دهـیم. انتخـاب چهـار نقطـه بـه دلیـل وجـود چهـار مرحلـه اصلی(شکل ۳-۷) در یک حمله است. به‌طور دقیق‌تر در این مرحله ابتـدا یـک رد حملـه را بـه چهار بخش تقسیم کرده و سپس با استفاده از یک تابع تصادفی در هر یک از این بخش‌ها، یک نقطه از آن بخش را انتخاب می‌کنیم(برش می‌زنیم). در مواردی کـه طـول رد حملـه کمتـر از چهار باشد، عمل برش به صورت تصادفی در چهار دفعه بر روی آن رد صـورت خواهـد گرفـت. خروجی این مرحله، یک رد حمله از ابتدای رد حمله تا نقطه برش و رویداد پس از نقطه بـرش که ما به آن حقیقت[1] می‌گوییم می باشد. این دو پارامتر نیز برای محاسبات بـه بخـش تجسـم فرستاده می‌شوند.

۴-۳- بررسی نتایج

در خروجی ما در انتها دو دسته داده خواهیم داشت. یک دسته که مبتنی بر الگوریتم مدل مارکوف با طول متغیر-روش A(VLMM-A) تولیدشده‌اند و دسته دیگـر داده‌هـای حاصـل از الگوریتم مدل مـارکوف بـا طـول متغیـر-روش S(VLMM-S). فـاوا و همکـاران[۱۸] در کـار خودشان برای تجسم مبتنی بر رفتار از روش مدل مارکوف بـا طـول متغیـر- روش A اسـتفاده کرده اند. در حقیقت معیار مقایسه ما با این کار تنها در قسمت محاسبات پیش‌بینی می‌توانست باشد، زیرا دسترسی به دادگان مورد استفاده‌شده در این مقاله ممکن نبود.

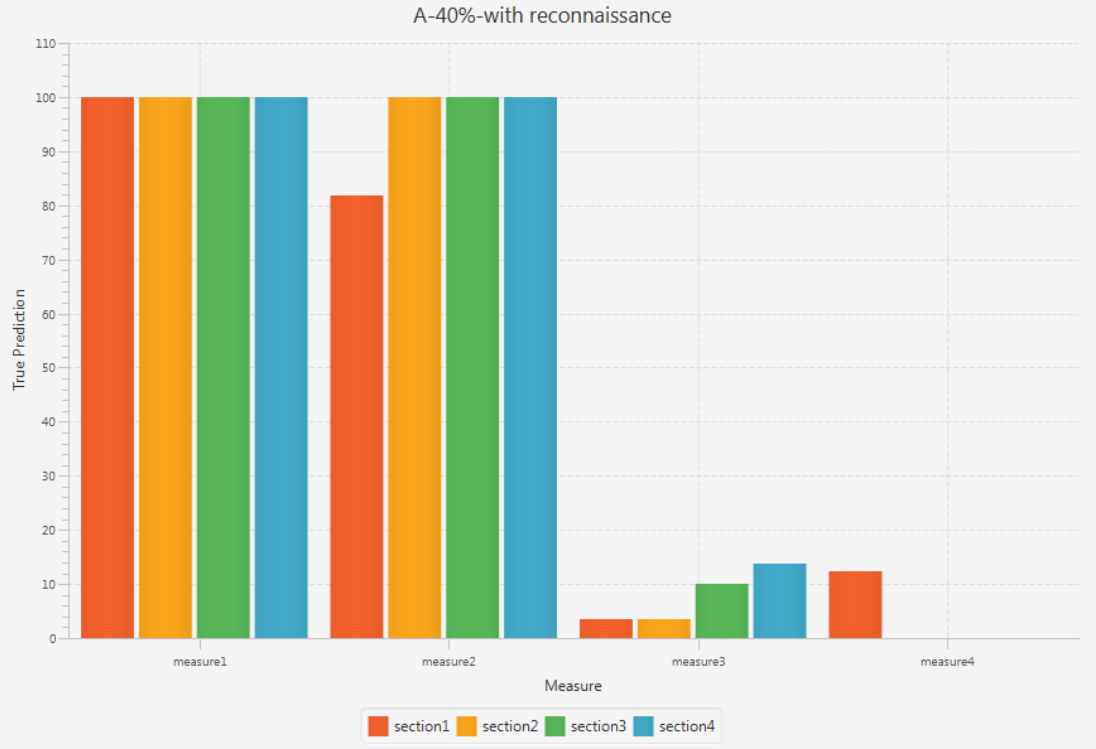
معیار مقایسه این دو روش، مقدار احتمال پیش‌بینی صحیح هر یک از این دو الگوریتم، نسبت به رویداد حقیقت است. به عبارت دیگر این هر یک از این الگوریتم ها، با چه احتمالی رویداد

[1] Ground Truth

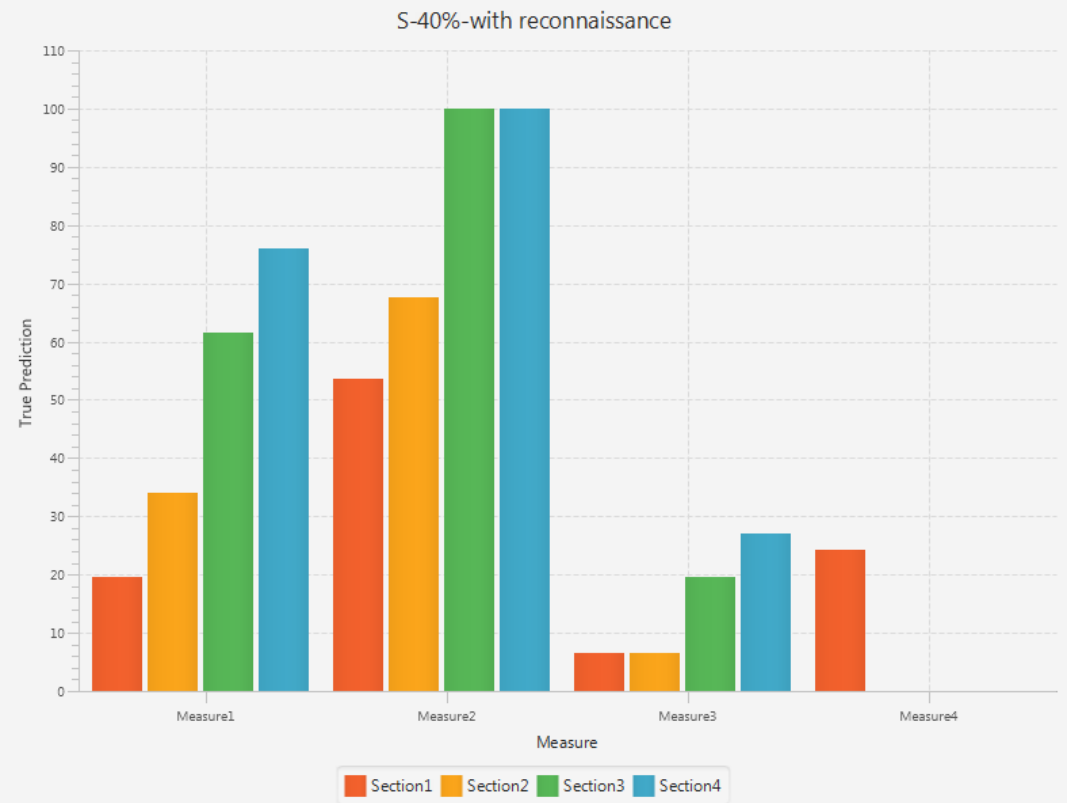
حقیقت را پیش‌بینی کرده اند.

ما برای این ارزیابی از دو دسته رد حمله به تعداد ۳۲۸۲ رد حمله استفاده کردیم. در یک دسته از رد حملات، رویدادهای مربوط به عمل شناسایی وجود دارد. البته به ازای هر ipمقصد، یکبار فقط عمل شناسایی اجازه داده شده است. این ایجاد محدودیت به این دلیل است که عمل شناسایی عموما در حملات از تکرار بالایی برخوردارند و می‌توانند نتایج پیش‌بینی نهایی را تحت تاثیر خود قرار دهند. دسته دیگر از رد حملات، دارای هیچ رویداد شناسایی نیست. حال این دو بخش داده به صورت جداگانه به بخش تولید داده آموزشی و آزمون و در ادامه بخش تجسم داده می‌شود.

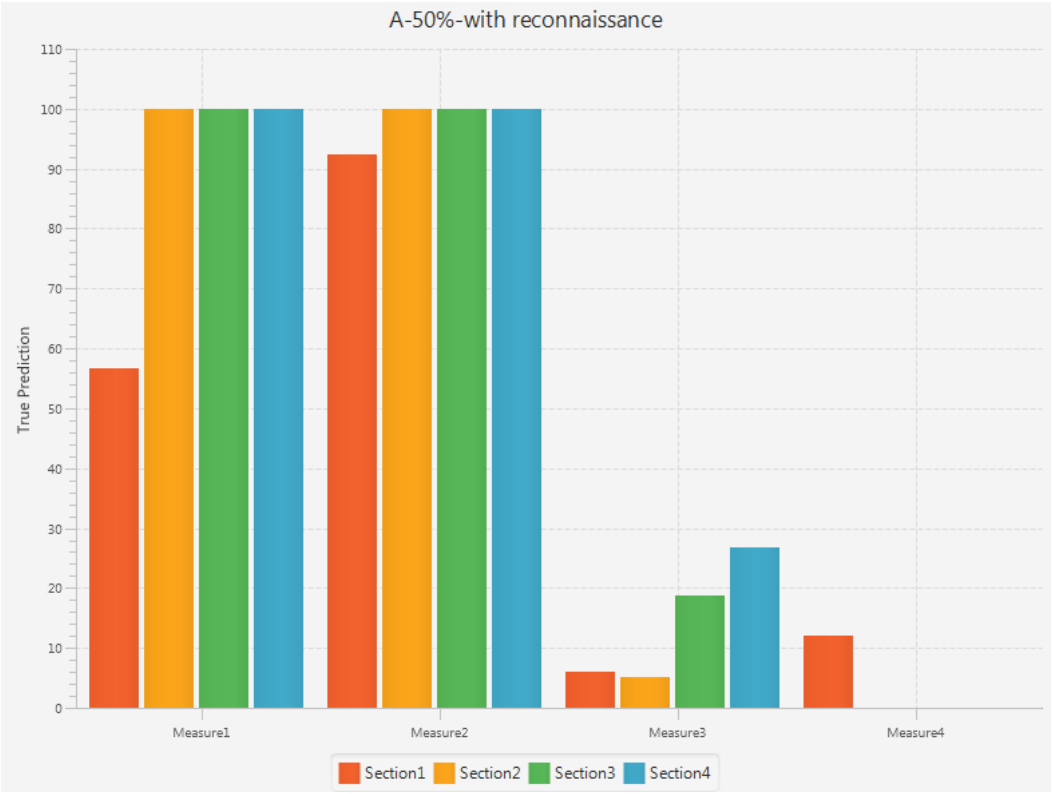
در شکل ۴-۲،شکل ۴-۳، شکل ۴-۴، شکل ۴-۵، شکل ۴-۶، شکل ۴-۷، شکل ۴-۸، شکل ۴-۹، شکل ۴-۱۰، شکل ۴-۱۱، شکل ۴-۱۲ وشکل ۴-۱۳ نتایج با توجه به داده همراه با شناسایی و میزان داده آموزشی متفاوت نشان داده شده است. نتایج، بالا بودن پیش‌بینی‌ها را در سطوح اول و دوم (که در این سطوح جواب‌ها کلی‌ترند) نسبت به سطوح سوم و چهارم (که جواب‌ها می‌بایست با جزئیات بیشتری باشند) نشان می‌دهد.



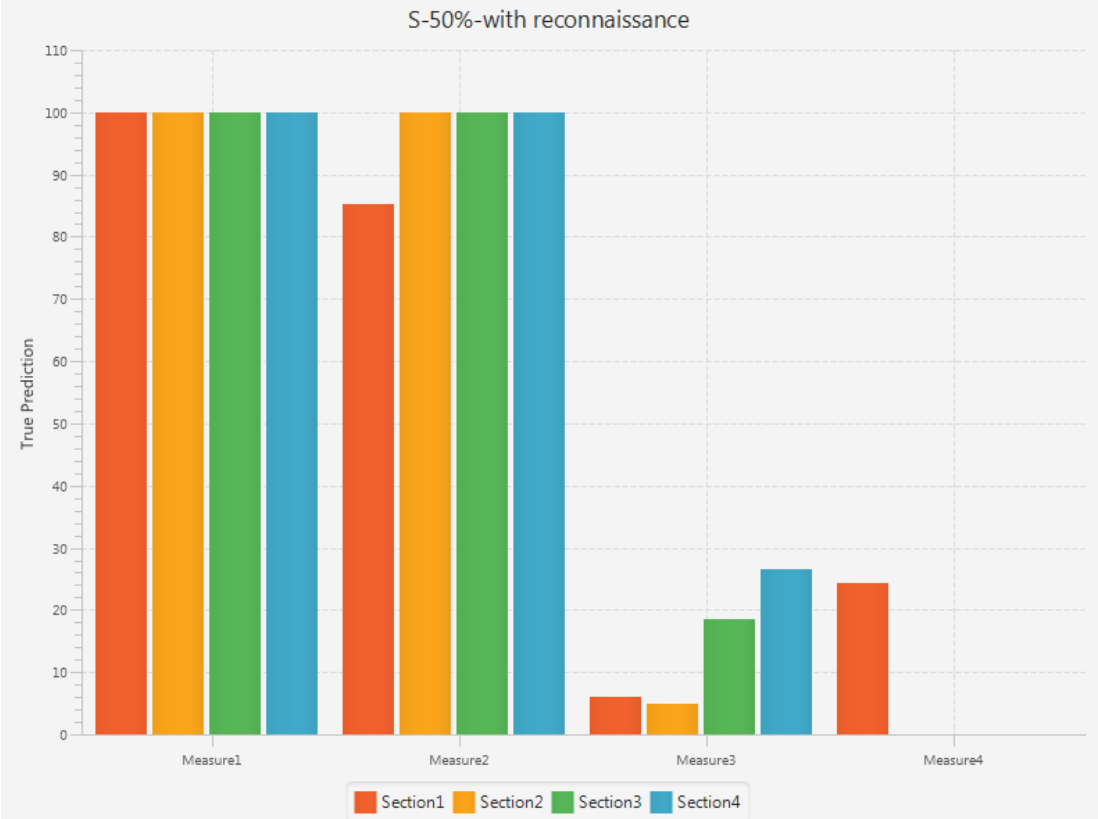
شکل ۴-۲: پیش‌بینی‌های حاصل از روش PPMA– با شناسایی و داده آموزشی ۴۰٪



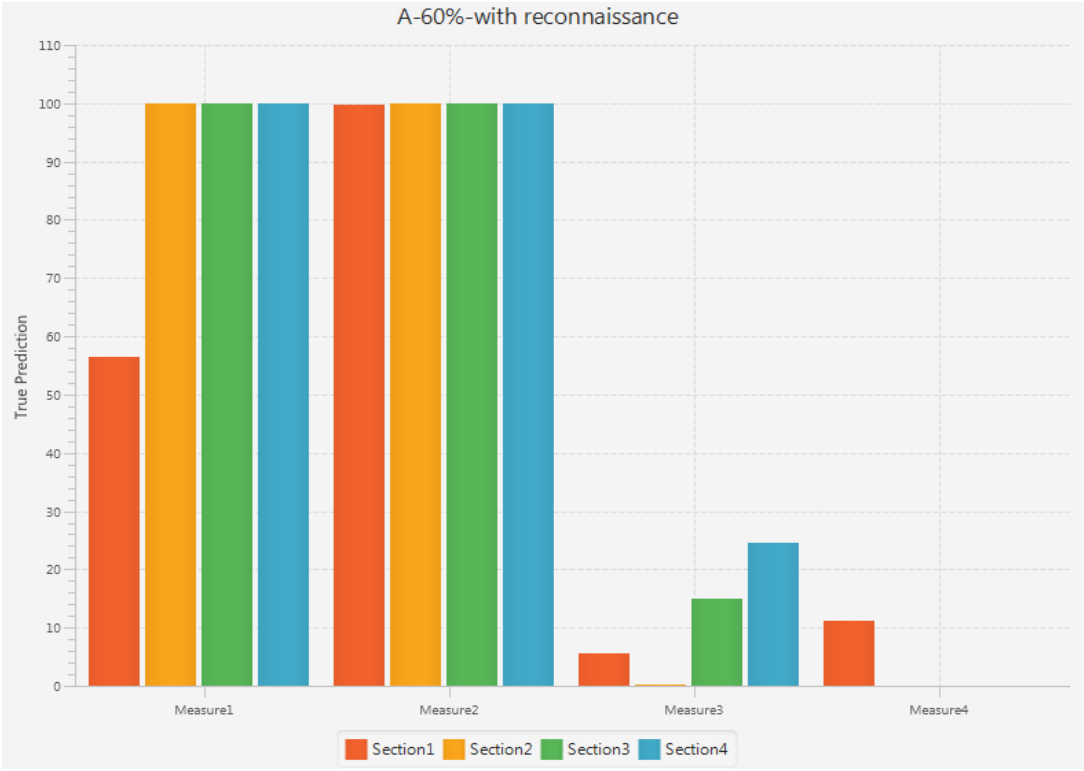
شکل ۴-۳ پیش‌بینی‌های حاصل از روش PPMS- با شناسایی و داده آموزشی ۴۰٪



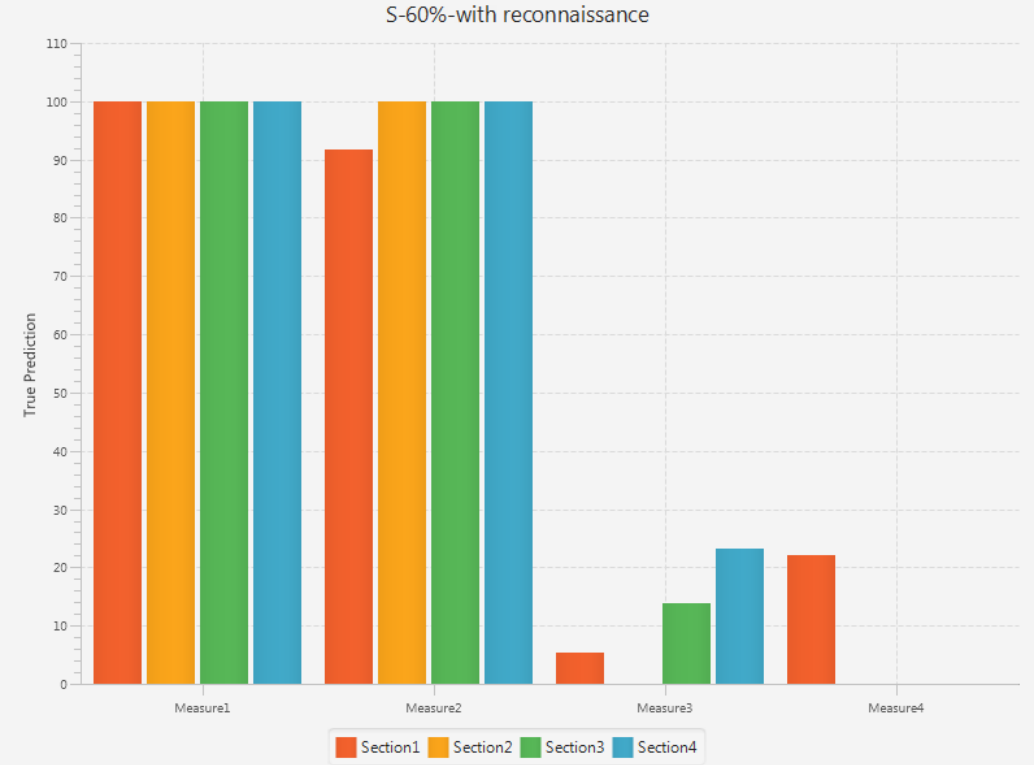
شکل ۴-۴: پیش‌بینی‌های حاصل از روش PPMA- با شناسایی و داده آموزشی ۵۰٪



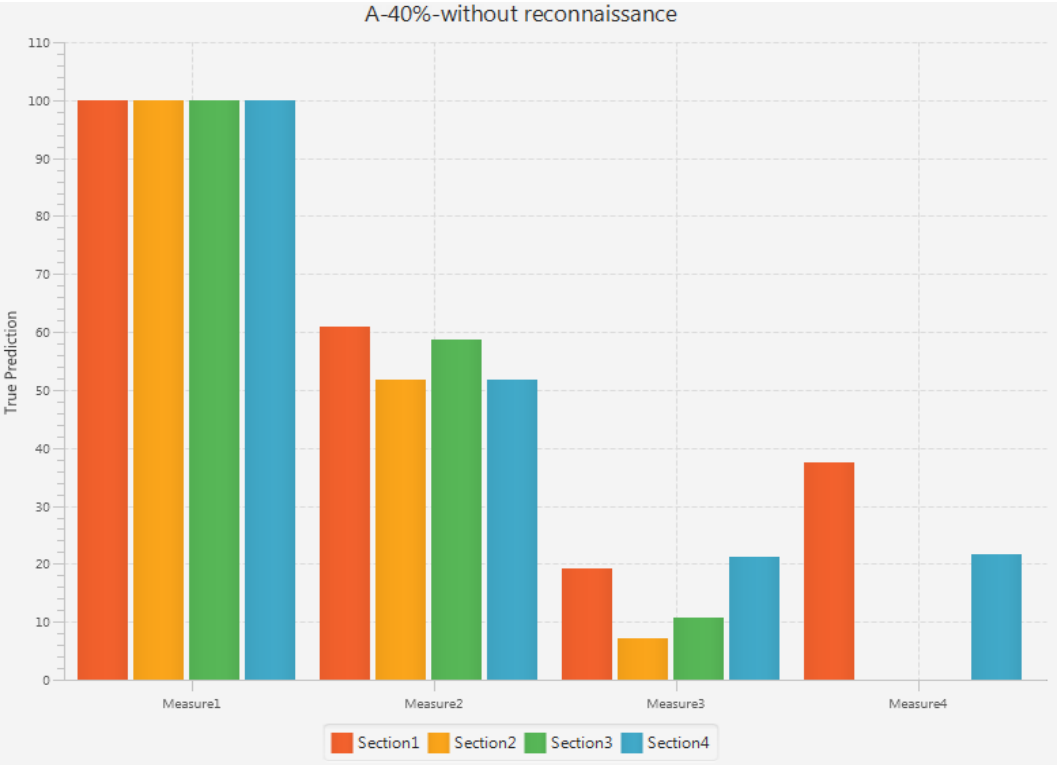
شکل ۴-۵: پیش‌بینی‌های حاصل از روش PPMS– با شناسایی و داده آموزشی ۵۰٪



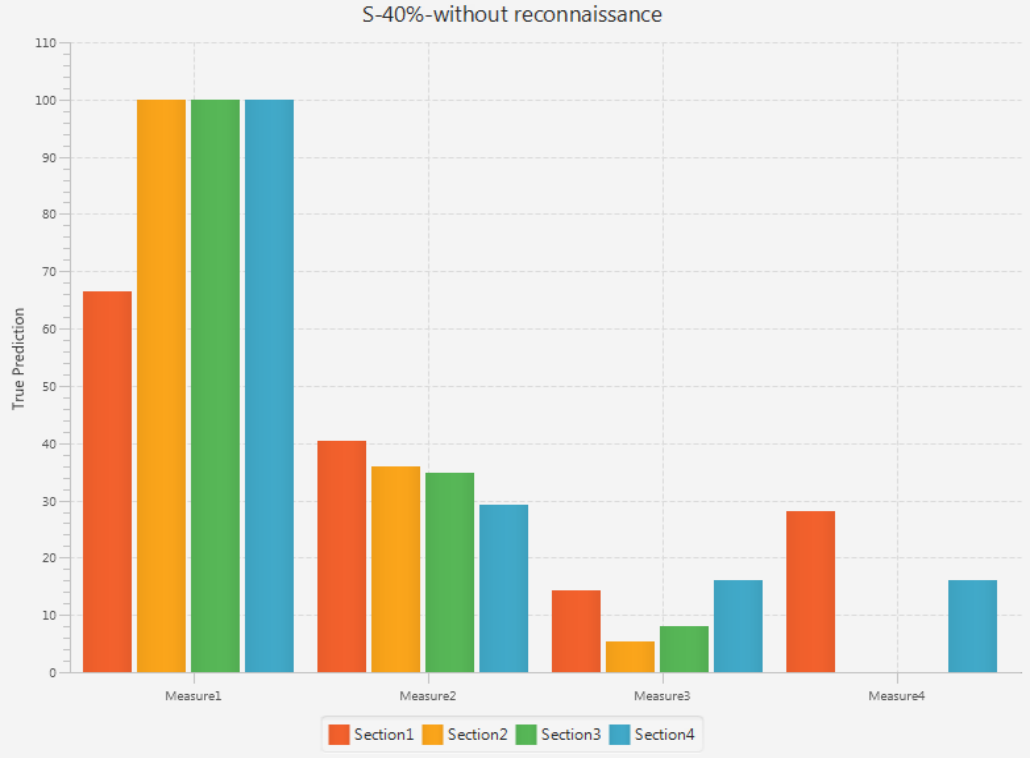
شکل ۴-۶: پیش‌بینی‌های حاصل از روش PPMA– با شناسایی و داده آموزشی ۶۰٪



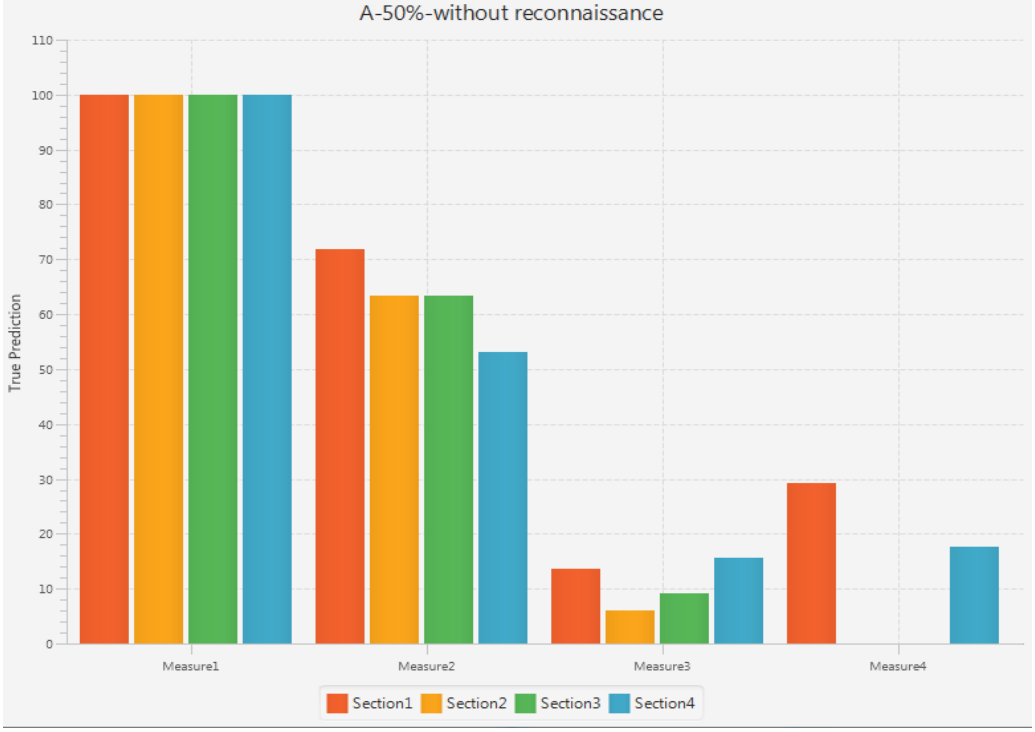
شکل ۴-۷: پیش‌بینی‌های حاصل از روش PPMS- با شناسایی و داده آموزشی ۶۰٪



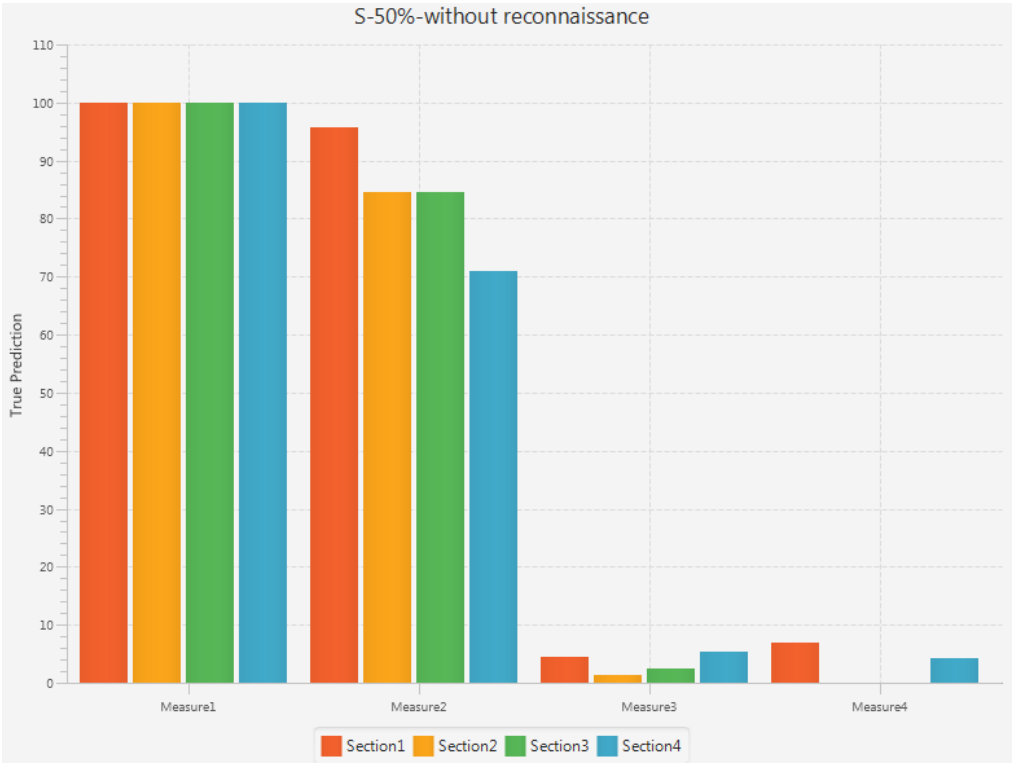
شکل ۴-۸: پیش‌بینی‌های حاصل از روش PPMA- بدون شناسایی و داده آموزشی ۴۰٪



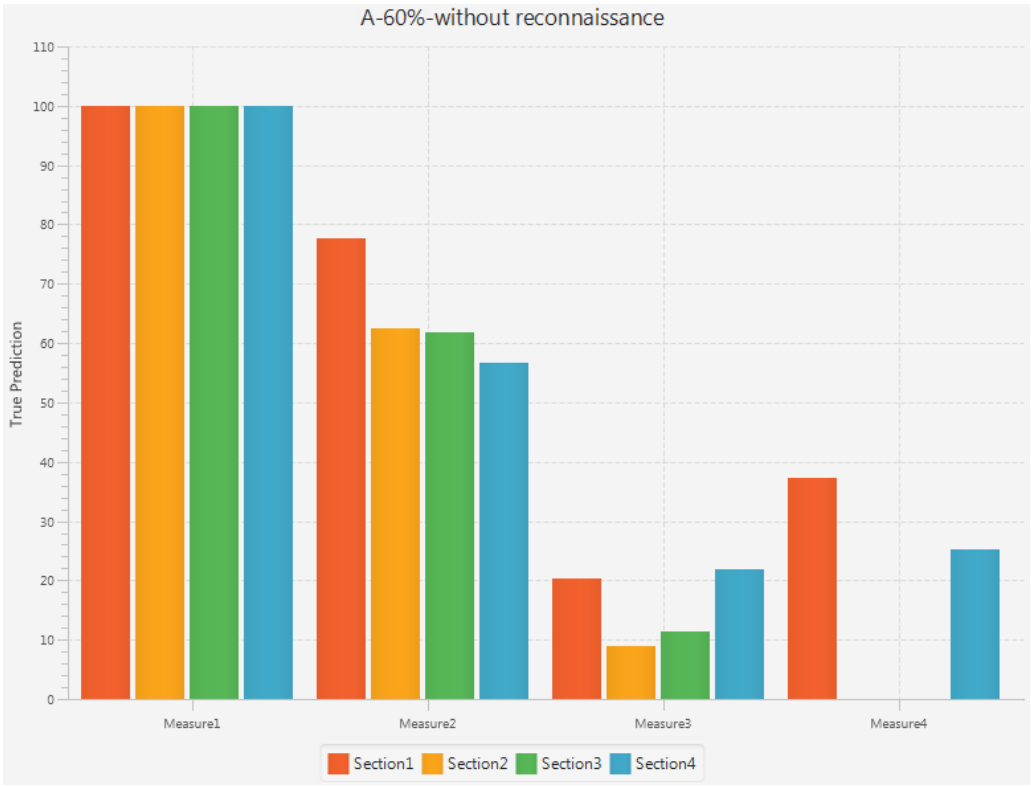
شکل ۴-۹: پیش‌بینی‌های حاصل از روش PPMS- بدون شناسایی و داده آموزشی ۴۰٪



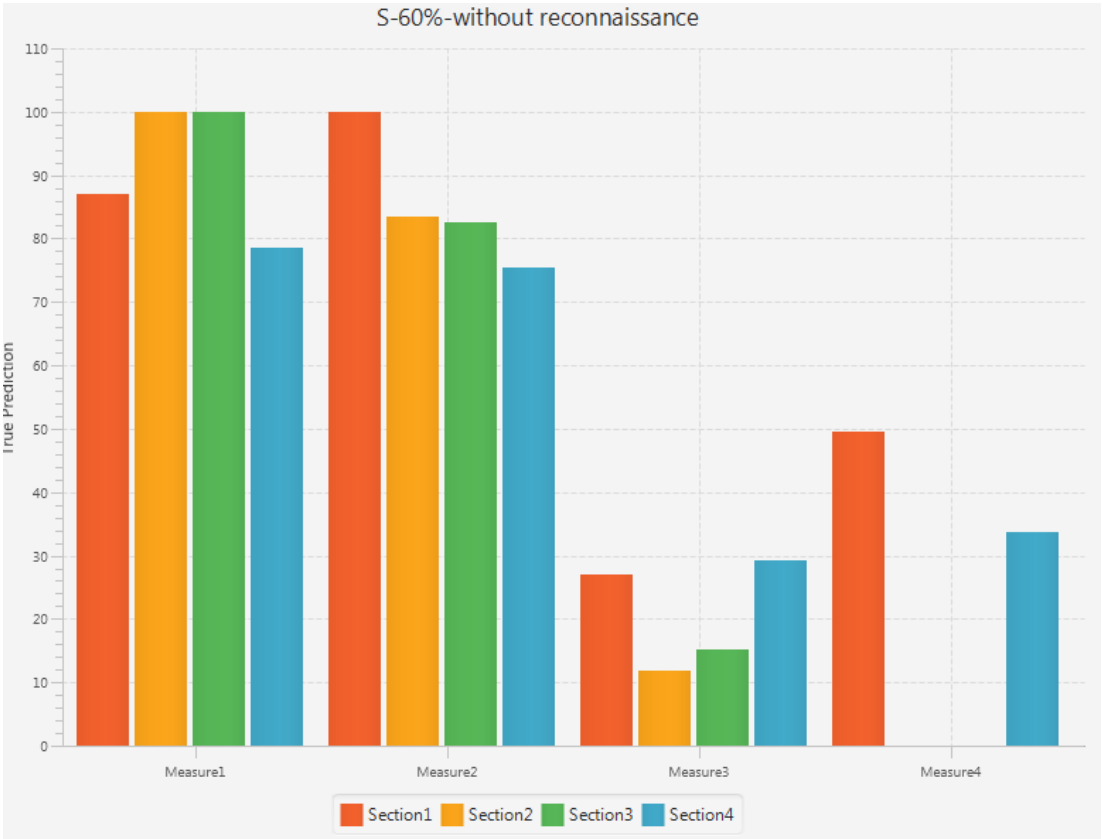
شکل ۴-۱۰: پیش‌بینی‌های حاصل از روش PPMA- بدون شناسایی و داده آموزشی ۵۰٪



شکل ۴-۱۱: پیش‌بینی‌های حاصل از روش PPMS- بدون شناسایی و داده آموزشی ۵۰٪

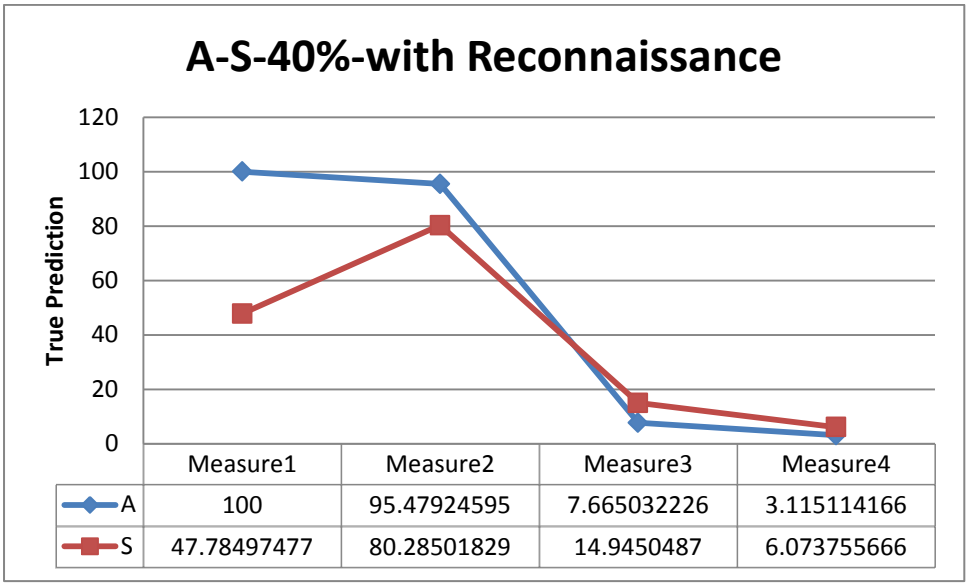


شکل ۴-۱۲: پیش‌بینی‌های حاصل از روش PPMA- بدون شناسایی و داده آموزشی ۶۰٪

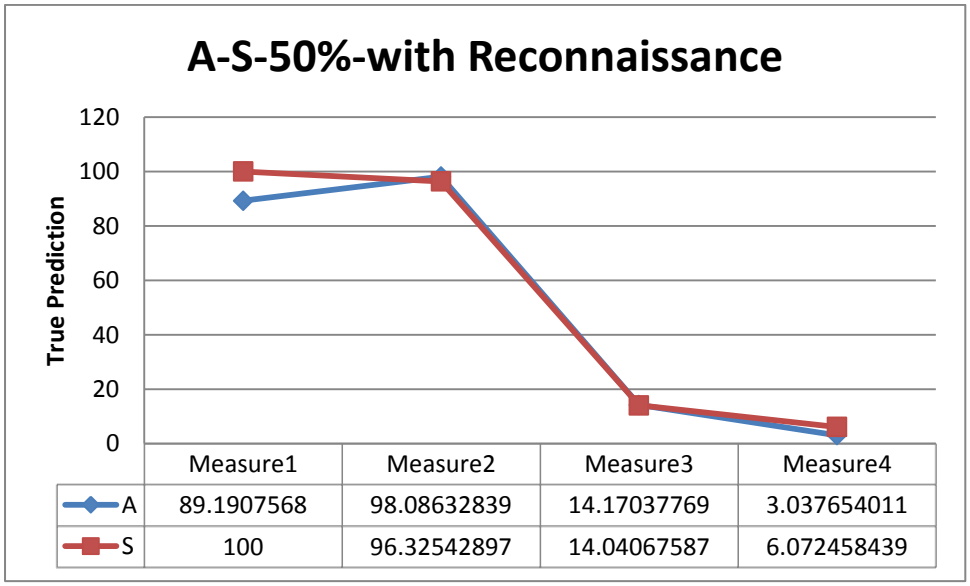


شکل ۴-۱۳: پیش‌بینی‌های حاصل از روش PPMS- بدون شناسایی و داده آموزشی ۶۰٪

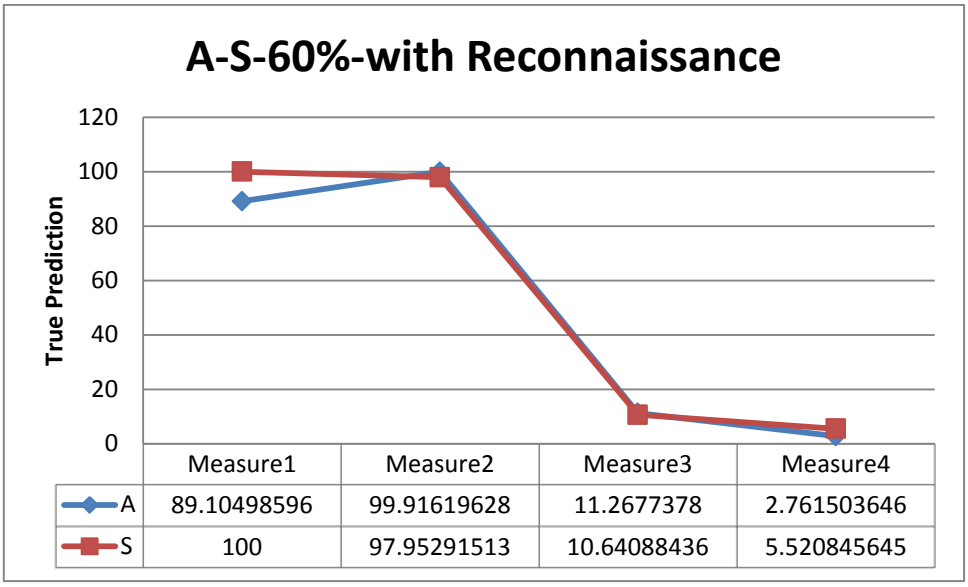
شکل ۴-۱۴، شکل ۴-۱۵،شکل ۴-۱۶، شکل ۴-۱۷، شکل ۴-۱۸و شکل ۴-۱۹ نمودارهـایی مقایسه‌ای از نتایج در هر دسته ارائه می‌کنند. در این شکل‌ها در هر معیار، از نتایج پیش‌بینی‌ها در بخش‌های مختلف، میانگین گرفته شده است و مقادیر این میانگین‌ها به صـورت مقایسـه‌ای در کنار هم قرار داده شده است. همانطور که در شکل ۴-۱۴، شکل ۴-۱۵ و شکل ۴-۱۶ دیـده می‌شود در داده همراه با شناسایی، هنگامی‌که داده آموزشی ۴۰٪ است، نتـایج کـاملا بـه نفـع روش PPMA است. ولی هنگامی‌که داده‌های آموزشی افزایش می‌یابد، روش PPMS در سـطح اول نتیجه بهتر و در باقی سطوح یا مقداری نزدیک و یا مقداری بیشتر ارائه می‌کند.



شکل ۴-۱۴: مقایسه روش PPMA و PPMS با داده همراه با شناسایی و داده آموزشی ۴۰٪

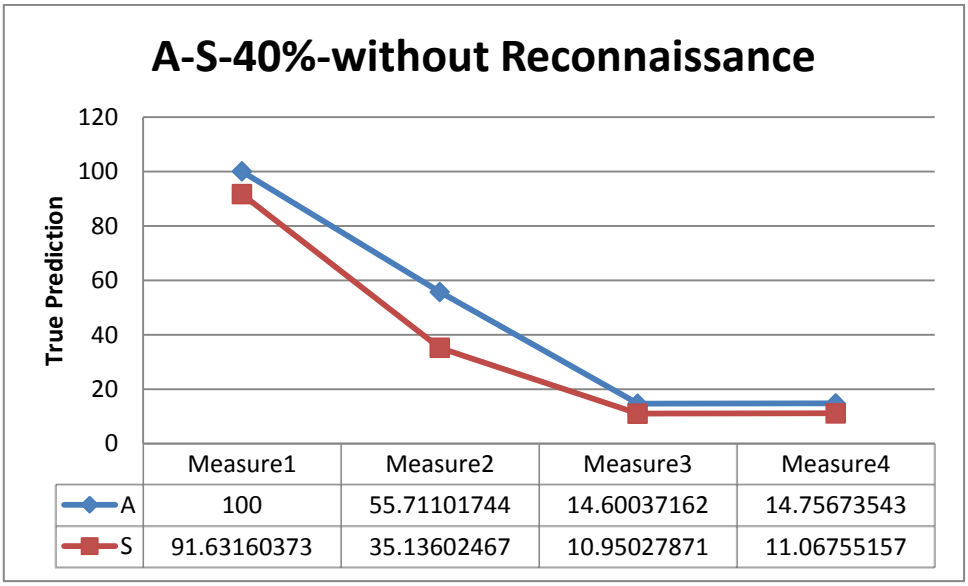


شکل ۴-۱۵: مقایسه روش PPMA و PPMS با داده همراه با شناسایی و داده آموزشی ۵۰٪

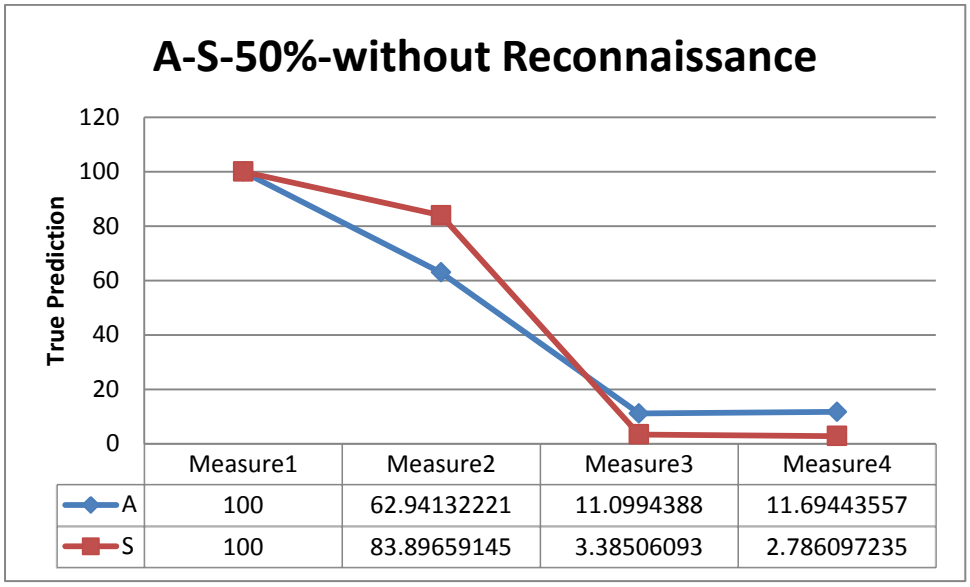


شکل ۴-۱۶: مقایسه روش PPMA و PPMS با داده همراه با شناسایی و داده آموزشی ۶۰٪

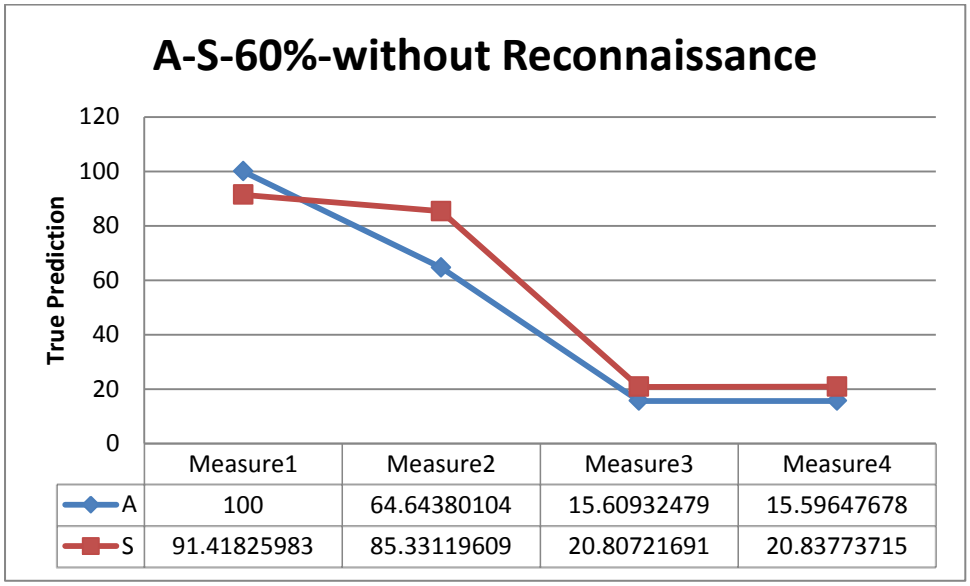
در شکل ۴-۱۷، شکل ۴-۱۸ و شکل ۴-۱۹ که تکرار رویدادهای شناسایی نیز وجود ندارد، با داده آموزشی ۴۰٪ برتری روش PPMA دیده می‌شود ولی در با افزایش داده آموزشی، روش PPMS نتیجه بهتر را نشان می‌دهد.



شکل ۴-۱۷: مقایسه روش PPMA و PPMS با داده بدون شناسایی و داده آموزشی ۴۰٪

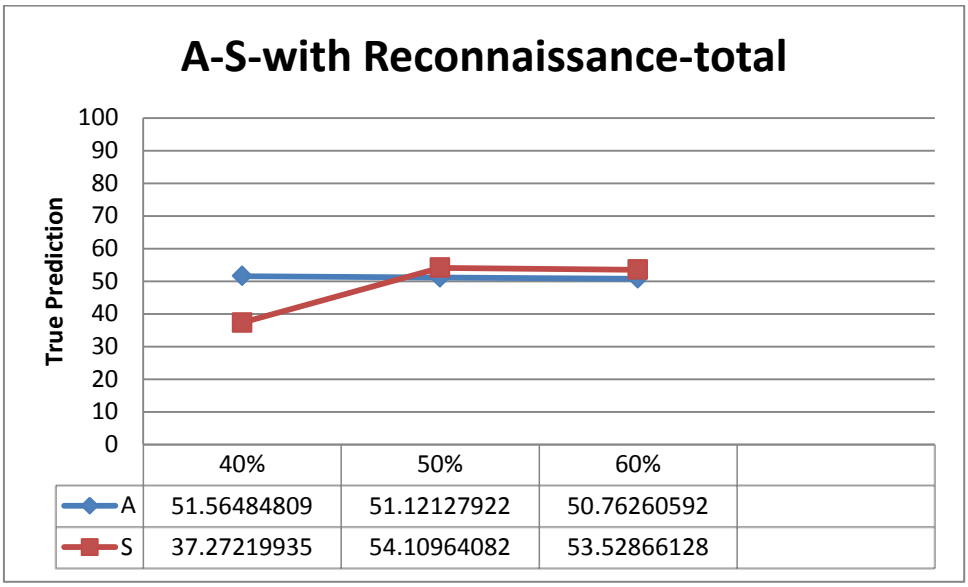


شکل ۴-۱۸: مقایسه روش PPMA و PPMS با داده بدون شناسایی و داده آموزشی ۵۰٪

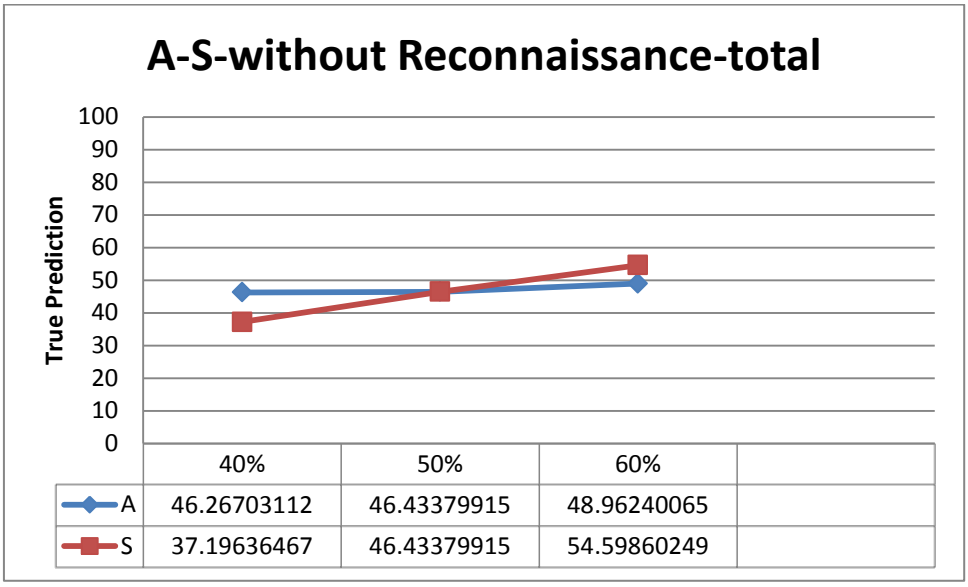


شکل ۴-۱۹: مقایسه روش PPMA و PPMS با داده بدون شناسایی و داده آموزشی ۶۰٪

شکل ۴-۲۰ و شکل ۴-۲۱ به مقایسه دو روش PPMA و PPMS به صورت کلی تر پرداخته است. در این جا میانگینی از میانگین‌های گرفته‌شده از هر معیار برای دو روش PPMA و PPMS، به عنوان معیار برای مقایسه دو روش PPMA و PPMS استفاده می‌شود. مشاهده می‌شود که در هر دو دسته داده بدون شناسایی و با شناسایی، هنگامی که از داده آموزشی ۴۰٪ استفاده می‌شود، روش PPMA و هنگامی‌که داده آموزشی زیاد می‌شود، روش PPMS نتایج بهتری را ارائه می‌کند. بررسی بیشتر این نتایج در پیوست آ قرار داده‌شده است.



شکل ۴-۲۰: مقایسه کلی روش PPMA و PPMS همراه با شناسایی و داده‌های آموزشی



شکل ۴-۲۱: مقایسه کلی روش PPMA و PPMS بدون شناسایی و داده‌های آموزشی

۴-۴ جمع‌بندی

در این فصل نحوه آماده‌سازی داده‌های آموزشی و آزمون تشریح شد. همچنین نتایج در قالب نمودارهایی نمایش داده شد. نتایج نشان می‌دهد که الگوریتم مارکوف که از روش PPMS استفاده می‌کند در داده‌های آموزش ۵۰٪ و ۶۰٪ پیش‌بینی‌های بهتری را انجام می‌دهد.

فصل ۵– جمع‌بندی و نتیجه‌گیری

هدف اصلی این پایان‌نامه، مدل‌سازی ویژگی‌های رفتاری حملات و استفاده از آن‌ها در پیش‌بینی رفتار های بعدی بود. همانطور که می‌دانیم یکی از چالش ها در این حوزه، داشتن یک دادگان مناسب است. ما در این کار به داده هایی نیاز داشتیم که بتوان با استفاده از توصیفات مربوط به آن‌ها، رفتارهایی را به آن‌ها اختصاص داد؛ به عبارت دیگر داده های خام برای این کار مفید نبودند و می‌بایست داده ها به اطلاعات تبدیل شوند. برای این کار ما از فرآیند همبسته سازی استفاده کردیم. الگوی کلی ما برای این فرآیند، مدل معرفی‌شده در [۴۱] بود.

داده های ورودی ما را هشدارهای صادرشده از سیستم تشخیص نفوذ Snort تشکیل می‌دهد. این داده ها از یک مسابقه به نام CDX که در سال ۲۰۰۹ انجام شد به دست آمده است. در مرحله بعدی ما به پیش‌پردازش این هشدارها پرداختیم و توصیفات این هشدارها را به عبارت دیگر غنی‌تر کردیم[۴۸]. در گام سوم از این داده ها، برچسب‌هایی(متا داده) ایجاد کردیم. این برچسب‌ها عبارتند از طبقه‌بندی، زیر طبقه‌بندی و بهره‌کشی. این سه برچسب بر اساس قواعدی که بر روی توصیفات هشدارها اعمال می‌شود، به هر هشدار اختصاص می‌یابد. گام چهارم به همبسته سازی چند مرحله ای این هشدارها می‌پردازد. در این گام ما از یک الگو به نام الگوی راهنما استفاده کردیم. الگوی استفاده‌شده در این کار توسعه داده‌شده الگوی پیشنهادی در[۳۹] است. از ویژگی‌های این الگو می‌توان به عدم وابستگی به شبکه، تنظیمات سیستم و نوع حمله، مقیاس‌پذیر بودن، عمومیت و انعطاف‌پذیری آن اشاره کرد. خروجی این گام را ردهای حمله می‌نامیم. ردهای حمله خود شامل رویدادهایی هستند که هر رویداد شامل اطلاعاتی از یک هشدار است.

بعد از آماده شدن این ردها، ما اقدام به آماده‌سازی این داده ها برای بررسی امکان‌پذیر بودن پیش‌بینی رفتاری یک مهاجم پرداختیم. ابتدا برای پیش‌بینی رفتاری ما از مدل مارکوف با طول متغیر استفاده کردیم. این روشها در فشرده‌سازی متنی که با دنباله‌ای از داده‌ها سروکار دارند بسیار استفاده‌شده‌اند. دلیل استفاده از این مدل‌ها هم وجود چنین دنباله‌هایی در کار ما بود.

ردهای حملات حاصل از مرحله پایانی فرآیند همبسته سازی، عملاً خاصیت دنباله‌ای یا علـت و معلولی را دارند. دلیل استفاده از مدل مارکوف با ویژگی طول متغیر، استفاده از نتـایج مختلـف حاصل از این مدل، در مرتبه‌های مختلف و ترکیـب کـردن آن‌هـا اسـت. در ایـن بخـش مـا از الگوریتم جدیدی به نام PPMS برای ترکیب نتایج مدل مارکوف در مرتبه‌های مختلف اسـتفاده کرده ایم. این روش بر یکنواخت سازی احتمالات تمرکز بیشتری دارد. فاوا و همکـاران[۱۸] در تحقیق خود از روش PPMA برای ترکیب کردن نتایج مـدل مـارکوف در مرتبـه‌هـای مختلـف استفاده کرده بودند. ما در این پایان‌نامـه الگـوریتم پیشـنهادی خـود را(PPMS) بـا الگـوریتم آنها(PPMA) مقایسه کردیم. نتایج نهایی نشان می‌دهد که روش PPMS هنگامی‌که داده‌هـای آموزش افزایش می‌یابد پیش‌بینی‌های صحیح بهتری را ارائه می‌کند.

در پایان می‌توان موارد زیر را برای کارهای آینده پیشنهاد نمود:

- استفاده از سه‌گانه قابلیت، فرصت و نیت در کنار رفتار می‌تواند به رسیدن نتیجه دقیق‌تر یاری برساند.
- استفاده از ساختار شبکه و تنظیمات آن و رابطه میان سرویس ها می‌تواند به پیش‌بینی یک رخداد در آینده کمک کند. این موارد خصوصاً زمانی که هدف ما پیش‌بینی رفتار بعدی با جزئیات بیشتر و دقیق‌تر است می‌تواند کمک‌کننده باشد و از میزان خطاها بکاهد.
- استفاده از تجارب حوزه‌های دیگر در موضوع پیش‌بینی. از جمله می‌توان به الگوریتم پیش‌بینی کننده کالمن-فیلتر اشاره کرد که برای ردگیری و پیش‌بینی رفتارها و رویدادها استفاده می‌شود.
- استفاده از یک روش بهتر برای برچسب‌گذاری رویدادها. روشهایی مانند NLP روش‌های مناسبی برای این کار هستند.

پیوست أ – بررسی گام های حمله

Rconnaissance

Sniffing

یک شنودگر[1] یک ابزار گیرنده بسته یا فریم است. این ابزار اصولا داده را هنگامی که از یک میزبان به میزبان دیگر در شبکه منتقل می شود گرفته و نمایش می دهد. به طور کلی یک شنودگر به ترافیک روی شبکه گوش کرده و آن را در یک قالب خط-فرمان یا گرافیکی در معرض دید هکر قرار می دهد. اکثریت شنودگرها سرآیند و بخش داده لایه ۲(فریم) یا لایه ۳(بسته) را نمایش می دهند.

شنودگرها همانطور که گفتیم برای گرفتن ترافیک ما بین دو سیستم به کار برده می شوند ولی آنها همچنین می توانند اطلاعات بیشتری را نیز فراهم کنند. بسته به اینکه چگونه شنودگر استفاده شود و معیارهای امنیتی در چه سطحی باشند، یک هکر می تواند از یک شنودگر برای کشف نام های کاربری، پسوردها و دیگر اطلاعات محرمانه ای که روی شبکه منتقل می شود استفاده کند. حال ببینیم یک نرم افزار شنودگر چگونه عمل می کند.

نرم افزار شنودگر کارش را به وسیله گرفتن بسته های ارسال شده برای آدرس های مک[2] انجام می دهد، نه لزوما فقط بسته ای که برای سیستم شنودگر ارسال می شود. این نحوه شنود به حالت بی قاعده[3] معروف است. به صورت عادی یک سیستم روی یک شبکه تنها ترافیک‌هایی که مستقیم به آدرس مکش ارسال شده را می خواند و پاسخ می دهد. در این بین تعدادی از ابزارهای هک کارت شبکه سیستم را به حالت بی قاعدع تغییر می دهند. در این حالت، یک کارت شبکه همه ترافیک را می خواند و آن را به شنودگر برای پردازش ارسال می کند. این حالت بر روی کارت شبکه با نصب درایور های نرم افزاری خاصی فعال می شود. هر پروتکلی که داده را رمز نمی کند مستعد مورد شنود قرار گرفتن است. پروتکل هایی مانند HTTP، POP3، SNMPو FTP معروف ترین پروتکل هایی هستند که شنود می شوند و می توانند به وسیله یک هکر برای جمع اوری اطلاعات ارزشمند مثل نام کاربری و پسورد مورد استفاده قرار گیرند.

به طور کلی دو نوع نرم افزار شنودگر وجود دارد: فعال و غیر فعال. شنود غیر فعال شامل گوش کردن و گرفتن ترافیک می شود و در یک شبکه که به وسیله هاب هایی متصل شده اند مفید است. شنود فعالانه شامل ایجاد یک جعل ARP [4] یا یک حمله ترافیک سیل آسا در برابر یک

[1] Sniffer
[2] MAC
[3] Promiscuous
[4] Addres Resolution Protocol

سوئیچ برای گرفتن ترافیک می شود. همانطور که از نامها نیز مشخص است شنود فعال، قابل شناسایی و شنود غیر فعال قابل شناسایی نیست.

در شبکه هایی که از هاب ها یا واسط بی سیم برای اتصال سیستم ها استفاده می شود همه میزبان های روی شبکه می توانند همه ترافیک ها را مشاهده کنند. بنابراین یک شنودگر غیرفعال می تواند تمام ترافیک‌هایی که از طریق هاب عبور می کند را بگیرد. یک شبکه سوئیچ شده به طور متفاوتی عمل می کند. سوئیچ به داده فرستاده شده به سمتش نگاه می کند و سعی می کند تا بسته ها را از روی آدرس مک شان به دریافت کننده های مورد نظرشان هدایت کند. این سوئیچ را قادر می سازد تا ترافیک شبکه را قطعه بندی کرده و ترافیک را تنها به آدرس های مک مقصد صحیح بفرستد.

روش دیگر برای شنود داده ها از طریق یک سوئیچ استفاده از منعکس سازی پورت[1] یا پورت SPAN[2] است. با فعال سازی منعکس سازی پورت، سوئیچ یک کپی از همه بسته های دیده شده روی یک پورت به پورت دیگر می سازد تا آن بسته بتواند تحلیل شود. در بسیاری از موارد پورت های SPAN توسط ادمین های شبکه به منظور نظارت بر شبکه برای اهداف قانونی مورد استفاده قرار می گیرد.

امروزه نیز بیشتر شبکه ها از استفاده از هاب ها به سوئیچ ها گرایش و تغییر پیدا کرده اند. به دلیل ساختاری که سوئیچ ها عمل می کنند جمع آوری اطلاعات از طریق شنود از یک شبکه سوئیچ شده سخت تر شده است. از راههای پیشنهادی برای شنود از طریق سوئیچ می توان به جعل ARP [3](مسموم سازی ARP[4]) وجعل DNS (مسموم سازی DNS) اشاره کرد.

✓ جعل ARP

این روش شامل استفاده از آدرس مک دروازه[5] شبکه به منظور منحرف کردن ترافیک فرستاده شده به دروازه، به سمت سیستم شنودگر می شود. یک هکر می تواند یک سوئیچ را با ترافیک بسیار زیادی (سیل آسا) مورد حمله قرار دهد تا عملا از کارش به عنوان سوئیچ بماند و تبدیل به یک هاب شود و همه ترافیک ها را به همه پورت ها ارسال کند. البته امروزه خیلی از سوئیچ ها دیگر در برابر این حملات با ترافیک سیل آسا مقاوم شده اند.

✓ جعل DNS

در این روش یک سرور DNS فریب داده می شود تا به اطلاعاتی که دریافت کرده است به دید

[1] Port Mirroring
[2] Switched Port Analyzer
[3] ARP Spoofing
[4] ARP Poisoning
[5] Gateway

اطلاعاتی موثق نگاه کند در حالی که حقیقتا چنین نیست. حال هنگامی که یک کاربر آدرس URL یک سایت خاص را در خواست می کند، آدرس روی یک سرور DNS جستجو شده تا آدرس IP متناظر را پیدا کند. اگر سرور DNS، تحت تسلط قرار گرفته باشد، کاربر به یک وب سایت دیگر مثلا یک وب سایت جعلی هدایت خواهد شد. برای اجرای یک حمله DNS، مهاجم از یک رخنه در نرم افزار سرور DNS برای این که بتواند اطلاعات ناصحیح را به ان بقبولاند بهره کشی می کند . اگر سرور به طور صحیحی پاسخ های DNS را تایید نکند تا تضمین کند که انها از یک منبع معتبر می آیند، سرور ذخیره سازی ورودی های ناصحیح را پایان داده و به روز رسانی را انجام داده و این اطلاعات را برای کاربرانی که تقاضاهای بعدی را می دهند استفاده می کند.

این روش می تواند برای جایگزین کردن محتوای دلبخواه برای یک مجموعه از قربانیان با یک مجموعه از محتویات مهاجم استفاده شود. برای مثال یک مهاجم ورودی های آدرس IP یک DNS برای یک وب سایت هدف روی یک سرور DNS معین را مسموم می کند و آنها را با آدرس IP یک سروی که هکر آن را کنترل می کند جایگزین می کند. هکر بعد از این می تواند ورودی های جعلی برای فایل های روی این سرور به همراه نامهایی مشابه با آنچه که روی سرور هذف بود قرار دهد. این فایل ها ممکن است حاوی محتوبات خرابکارانه مثل یک ویروس یا کرم باشد. با فریب خوردن قربانی وبارگذاری این فایل ها کاربر می تواند به مقصود خود برسد.

انواع روش های جعل DNS از قرار زیر است:

جعل اینترانت: عمل کردن به عنوان یک دستگاه روی یک شبکه داخلی مشابه

جعل اینترنت: عمل کردن به عنوان یک دستگاه روی اینترنت

مسموم سازی پروکسی سرور DNS: تغییر ورودی های DNS روی یک سرور پروکسی تا اینکه کاربر به یک میزبان سیستمی متفاوتی هدایت شود.

مسموم سازی کش DNS[1] : تغییر ورودی های DNS روی هر سیستم تا اینکه کاربر به یک میزبان متفاوتی هدایت شود

[1] Cache

Footprinting

ردیابی[1] پروسه ایجاد یک طرح اولیه یا یک نقشه از شبکه و سیستم های سازمان است. جمع آوری اطلاعات نیز با نام ردیابی یک سازمان نیز شناخته می شود. ردیابی با تعیین سیستم هدف، نرم افزار ها یا مکان فیزیکی هدف آغاز می شود. به محض این که این اطلاعات پیدا شد، اطلاعات خاص تر در مورد سازمان با استفاده از روش های غیر نفوذی انجام می شود. برای مثال صفحه وب شخصی سازمان ممکن است شامل یک دایرکتوری شخصی یا یک لیستی از کارمندان باشد که می تواند برای هکر هنگان انجام حملات اجتماعی مفید باشد.

اطلاعاتی که هکر در طی فاز ردیابی به دنبال آن است شامل هر چیزی است که سرنخی از معماری شبکه، سرور و انواع نرم افزارهایی که داده های ازشمندی را در خود ذخیره کرده اند، در اختیار هکر قرار دهد. قبل از اینکه یک حمله یا یک بهره کشی انجام شود، سیستم عامل و نسخه انواع نرم افزارها می بایست هویدا گردد تا بتوان حمله ای کاراتر بر روی هدف انجام داد. در زیر تعدادی از اطلاعاتی که نیاز است در این فاز در باره یک هدف جمع اوری شود آورده شده است:

- نام دامنه
- بلاک های شبکه
- سرویس های شبکه و نرم افزارهای آن
- معماری سیستم
- سیستم تشخصص نفوذ
- آدرس های IP خاص
- مکانیزم های کنترل دسترسی
- شماره تلفن ها
- آدرس های تماس

از ابزارهایی که می تواند در ردیابی به مهاجم کمک کند ابزار tracerout است.

✓ استفاده از tracerout در ردیابی

Tracerout یک ابزار packet-tracking است که در اکثر سیستم عامل ها موجود است. این نرم افزار با فرستادن یک ICMP echo به هرگام(مسیریاب یا دروازه) روی مسیر تا زمانی که به آدرس مقصد برسد کار می کند. هنگامی که پیغام های ICMP از هر مسیر یاب برگشت داده

[1] Footprinting

می شود، TTL[1] آن به ازای هر مسیر یاب به اندازه یکی کم می شود. با این روش هکر می تواند تعیین کند که چه تعداد گام مسیریاب از فرستنده تا مقصد وجود دارد.
یک مسئله در استفاده کردن از این روش این است که هنگامی که این بسته ها با یک فایروال یا یک مسیریاب فیلتر کننده بسته مواجه می شود Time out می شوند. اگر چه که فایروال این ابزار را از رسیدن به مقصودش منع می کند ولی به هکر از وجود یک فایروال اطلاع می دهد که می بایست از روش هایی برای دور زدن فایروال استفاده شود.

Scanning

بعد از اینکه گام ردیابی کامل شد، گام پویش[2] اجرا می شود. در واقع اگر ردیابی را گامی در جهت شناسایی مکان های حاوی اطلاعات بدانیم، پویش معادل سر زدن به همه دیوارها برای پیدا کردن همه درها و پنجره ها است. گام پویش، فرآیند مشخص کردن سیستم هایی است که روشن هستند و در حال پاسخگویی بر روی شبکه می باشند. به طور کلی سه نوع پویش وجود دارد:
پویش پورت: پویش پورت، فرآیند مشخص کردن پورت های TCP/IP موجود و باز روی یک سیستم است. ابزارهای پویش پورت، یک هکر را قادر می سازند تا یک هکر در باره سرویس های موجود روی یک سیستم معین اطلاع کسب کند.
پویش شبکه: پویش شبکه یک رویه برای مشخص کردن میزبان های فعال روی یک شبکه خواه برای حمله به آنها یا بررسی امنیتی شبکه می باشد. میزبان ها به وسیله آدرس های IP معینی مشخص می شوند. این ابزراها سعی می کنند تا همه میزبان های زنده یا پاسخگو را روی شبکه و ادرس های IP متناظرشان بیابند.
پویش آسیب پذیری: پویش آسیب پذیری فرآیند مشخص کردن کنش گرایانه آسیب پذیری های سیستم های کامپیوتری روی یک شبکه است. یک پویشگر آسیب پذیری مواردی مثل سیستم عامل و شماره نسخه به همراه اینکه که کدام سرویس پک آن نصب شده را مشخص می کند. بنا براین پویشگر مشخص ضعف ها یا آسیب پذیری ها در سیستم عامل را مشخص می کند که در فاز حمله، مهاجم می تواند از آن ضعف ها برای به دست آوردن دسترسی به سیستم بهره برداری نماید. پویش آسیب پذیری و شبکه معمولا می تواند شناسایی شود. زیرا پویشگر می باست با سیستم مورد نظر روی شبکه تعملاتی را داشته باشد. بسته به نوع نرم افزار پویشگر و سرعت پویش آن، یک سیستم تشخیص نفوذ(IDS) پویش را تشخیص داده و آن را به عنوان یک رویدادIDS علامت گذاری می کند. تعدادی از ابزارها برای پویش از روشهای متفاوتی بهره می گیرند تا بتوانند سیستم های تشخیص نفوذ را فریب بدهند و بتوانند به

[1] Time To Live
[2] Scanning

صورت ناشناخته تر به پویش خود ادامه دهند.
معمولا روش شناسی پویش با بررسی سیستم هایی که روی شبکه روشن هستند و به تقاضاهای اتصال یا کاوش[1] ها پاسخ می دهند آغاز می شود. پویشICMP یا یک تجسس به وسیله پینگ[2]، فرآین فرستادن یک تقاضای ICMP یا پینگ به همه میزبان های روی شبکه برای تعیین اینکه که کدام یک ازآنها روشن است و به تقاضا ها پاسخ می دهد می باشد. البته تقریبا هر سیستم IDS یا IPS یک تجسس به وسیله پینگ را شناسایی و به ادمین امنیتی آن را هشدار می دهد. اکثر فایروال ها و سرورهای پروکسی پاسخ های پینگ را بلاک کرده و بنابراین یک هکر نمی تواند به درستی تعیین کند که آیا یک سیستم واقعا روشن هست یا نه. بنابراین اگر پینگ کردن به ما نشان می دهد که به عنوان مثال سیستمی روشن نیست به این معنا نیست که واقعا سیستم روشنی در شبکه وجود ندارد. در این مواقع نیاز است تا از یم روش جایگزین در این مواقع استفاده شود. گام بعدی روش شناسی پویش، بررسی پورت های باز است. پویش پورت، اطلاعات ارزشمند تری نسبت به یک تجسس به وسیله پینگ درباره میزبان و آسیب پذیری های آن به هکر می دهد. شناسایی سرویس نیز گام سوم روش شناسی پویش است. معمولا این گام با استفاده از یک ابزار مشابه با پویش پورت انجام می شود. یک مهاجم به وسیله تعیین پورت های باز می تواند وضعیت سرویس های مرتبط با آن شماره پورت را بفهمد. Nmap یک ابزار متن باز است که به صورت کارایی می تواند تجسس به وسیله پینگ، پویش پورت، شناسایی سرویس، شناسایی آدرس IP و سیستم عامل را انجام دهد.
گام چهارم روش شناسی پویش، کسب نشانه[3] یا انگشت نگاری[4] است. فرآیند انگشت نگاری به مهاجم این توانایی را می دهد تا اهداف با آسیب پذیری خاص یا با ارزش بالا را شناسایی نماید. هکرها به دنبال آسانترین راه برای کسب دسترسی به یک سیستم یا شبکه هستند. کسب نشانه، یک فرآیند باز کردن اتصال و خواندن نشانه یا پاسخ فرستاده شده توسط نرم افزار است. تعدادی از سرورهای وب، میل و FTP به اتصال Telnet با نام و نسخه نرم افزار پاسخ می دهند. این به یک هکر کمک می کند که بتواند سیستم عامل و نرم افزارهای کاربردی را بهتر شناسایی نماید.
انگشت نگاری پشته فعالانه[5] رایج ترین نوع انگشت نگاری است. این روش شامل فرستادن یک داده به یک سیستم برای دیدن این است که چه طور سیستم به آن پاسخ می گوید. این کار به این دلیل انجام می شود که فروشندگان سیستم عامل پشته TCP را به طور متمایزی پیاده سازی می کنند و با سیستم عامل های متفاوت، پاسخ ها نیز متفاوت خواهد بود. انگشت نگاری

[1] Probe
[2] Ping Sweep
[3] Banner Grabbing
[4] Fingerprinting
[5] Active stack Fingerprinting

پشته فعالانه، قابل شناسایی است زیرا آن به طور مکرر سعی بر اتصال با یک سیستم هدف یکسان را دارد.
انگشت نگاری پشته منفعل[1] پنهان کارتر است و از روی بررسی ترافیک روی شبکه به نوع سیستم عامل پی می برد. این روش از تکنیک های شنود به جای تکنیکهای پویش بهره می برد. انگشت نگاری منفعلانه معمولا به وسیله یک IDS یا دیگر سیستم های امنیتی ناشناخته می ماند ولی در عوضش دقتش از انگشت نگاری فعالانه پایین تر است.
گام آخر در روش شناسی پویش، استفاده از سرورهای پروکسی در هنگام پویش است. یک سرور پروکسی یک کامپیوتر است که به عنوان یک واسط ما بین یک مهاجم و کامپیوتر مقصد قرار می گیرد. استفاده از یک سرور پروکسی به کاربر این اجازه را می دهد تا کاربر بر روی شبکه مخفی باشد. هکر ابتدا یک اتصال به سرور پروکسی ایجاد می کند و سپس یک تقاضا برای اتصال به یک کامپیوتر مقصد از طریق اتصال با این پروکسی می دهد. لزوما تقاضاهای دسترسی پروکسی به کامپیوتر هدف از کامپیوتر مهاجم نیست. این روش به یک هکر این اجازه را می دهد که بتواند به صورت ناشناس وب گردی کرده یا در غیر این صورت حملات شان را پنهان کنند.

Enumeration

سرشماری[2] پس از مرحله پویش انجام می گیرد و در آن به جمع آوری و پردازش نام های کاربری، نام های ماشین ها، منابع شبکه ای، منابع به اشتراک گذاشته شده و سرویس ها پرداخته می شود. همچنین سرشماری، به یک تقاضای فعالانه یا اتصال به یک سیستم هدف برای کسب این اطلاعات نیز گفته می شود. هدف سرشماری شناسایی حساب کاربری یا سیستمی برای استفاده بالقوه در هک کردن یک سیستم مورد هدف است. در این گام لزوما هدف، پیدا کردن حساب ادمین سیستم نیست زیرا از اکثر حساب ها می توان بالا رفته و به حساب هایی که دسترسی بیشتر را ممکن می سازد رسید. تعداد زیادی از ابزارهای هک وجود دارند که برای پویش IP های شبکه به منظور مشخص کردن اطلاعات NetBios name طراحی شده اند. برای هر میزبان پاسخ دهنده، ابزارهایی اطلاعات آدرسIP، نام کامپیوتری NetBios، نام کاربری logged-in و آدرس های MAC را لیست می کنند.
به عنوان مثال بر روی دامنه Windows 2000، یک ابزار از پیش نصب شده ای به نام net view برای سرشماری NetBIOS وجود دارد. یکی از آسیب پذیری های معروف در مرحله سرشماری، Null Session است که در CISF[3] یا SMB بسته به سیستم عامل آن اتفاق می افتد. SMB و استانداردهای NetBIOSدر ویندوز شامل API هایی است که اطلاعاتی را درباره

[1] Passive stack fngerprinting
[2] Enumeration
[3] Common Internet File System

یک سیستم از طریق پورت۱۳۹ بر می گردانند.Null session TCP زمانی اتفاق می افتد که شما به یک سیستم بدون هیچ نام کاربری یا پسوردی وارد می شوید. به محض اینکه یک هکر بتواند یک اتصال NetBIOS با استفاده از یک null session برقرار نماید می تواند به آسانی یک dump کامل از همه نام های کاربری، گروه ها، اجزای به اشتراک گذاشته ده، مجوزها، سیاست ها،سرویس ها و هر آن چه که یک حساب کاربری null استفاده می کند داشته باشد.
نوع دیگر سرشماری، سرشماری SNMP است. در این فرایند از SNMP برای سرشماری حساب های کاربری روی یک سیتم هدف استفاده می شود.

Intrusion

مهاجمین از یک پیشروی منطقی برای حمله استفاده می کنند. یعنی ابتدا از خارج از یک آسیب پذیری در یک سرویس سوء استفاده کرده، سپس دسترسی شل داخلی را بدست می آورند. در نهایت به محض اینکه دسترسی شل بدست آمد، دیگر مهاجمین به عنوان یک کاربر داخلی در سیستم فرض می شوند و از آن به بعد می توان هدف نهایی حمله را انجام داد(ایجاد درب پشتی، انکار سرویس و...). به طور کلی حملات را می توان با دو معیار تقسیم بندی نمود:

- دسترسی از خارج (Remote Access)
- دسترسی از داخل (Local Access)

. دسترسی داخلی نیز هنگامی بدست می آید که مهاجم توانسته باشد وارد یک سیستم شود یا یک خط شل حقیقی را کسب کرده باشد. حملات دسترسی از داخل به عنوان حملات افزایش اولویت نیز شناخته می شوند(که در بخش آینده به آن پرداخته است).

دسترسی از خارج

در دسترسی از خارج، کسب دسترسی از طریق شبکه (برای مثال یک سرویس گوش کننده) یا دیگر کانال های ارتباطی صورت می گیرد. راه های سوء استفاده از بیرون شبکه را می توان در ۴ دسته تقسیم بندی نمود[۵۸]:

- **بهره کشی از یک سرویس گوش کننده**: هر سرویسی(مانند TCP/UDP، SSH و...) که به مهاجم این امکان را بدهد که بتواند از راه دور به وسیله آن به یک سیستم وارد شود، می تواند مورد استفاده توسط مهاجم برای نفوذ باشد. به طور بالقوه مهاجمین برای کسب دسترسی می بایست از یک سرویس گوش کننده استفاده کنند.

زیرا یک مهاجم بدون یک سرویس گوش کننده نمی تواند برای کسب دسترسی از راه دور اقدام نماید.

- **مسیر یابی از طریق یک سیستم**: مهاجمین می توانند از طریق دور زدن فایروال، بدون اینکه به داخل فایروال هم نفوذ کنند به مسیر یابی در داخل شبکه بپردازند. در این نوع حملات مهاجمین دور زدن فایروال ها را به کمک بسته های Source-routing انجام می دهند. routing Source مکانیزمی است که اجازه می دهد بسته ها بتوانند اطلاعات اضافی حاوی چندین آدرس IP باشند و عملا یک مسیر یاب با دیدن این بسته ها متوجه می شود که در چه مسرهایی باید بسته را بفرستد. بدین وسیله فرستنده می تواند مسیری را که یک بسته طی بکند را در قسمت مبدا بسته مشخص نماید. مهاجمین از این طریق می توانند شبکه مورد نظرشان به وسیله فرستادن این بسته ها به قسمت های مختلف شبکه بررسی نمایند و اطلاعاتی در مورد توپولوژی شبکه، سیستم عامل یک هدف و ... را کسب نمایند. این کار زمانی میسر است که IP forwarding بر روی سیستم مورد نظر فعال شده باشد.
- **اجرای از راه دور توسط یک کاربر**: این نوع حملات و بهره کشی ها معمولا از طریق وارد شدن قربانی به یک سایت آلوده و یا اجرای یک برنامه آلوده توسط قربانی که مهاجم آن را ارسال کرده است میسر می شود.
- **حملات حالت** بی قاعده: مهاجم می تواند با سوء استفاده از ابزارهایی که کارت شبکه را در حالت بی قاعده قرار می دهند(به عنوان مثال tcpdump یا wireshark) با فرض وجود آسیب پذیری، از آن ها عملا در جهت اهداف خود استفاده نماید.

دسترسی از داخل

همانطور که در مرحله قبل هم گفته شد اکثر مهاجمین تلاش می کنند تا از طریق سوء استفاده از آسیب پذیری ها از راه دور اقدام به کسب دسترسی داخلی پیدا کنند. در جایی که مهاجمین یک خط فرمان شل روی سیستم قربانی داشته باشد، مهاجم در داخل سیستم قربانی فرض می شود. اگرچه دسترسی ریشه از طریق سوء استفاده از آسیب پذیری ها از راه دور ممکن است ولی اغلب مهاجمین ابتدا سعی در کسب دسترسی در سطح کاربر می نمایند. بعد از آن می بایست اولویت خود را به سطح بالاتر(یا سطح دیگر) برسانند که به آن افزایش اولویت می گویند. که بخش بعد به این موضوع خواهد پرداخت.

Privilege Escalation

گام چهارم سیکل حمله، افزایش اولویت ها است. هدف انجام این گام کسب حقوق و مجوزهای بیشتر است. به طور ساده تر، در افزایش اولویت، یک حساب کاربری عادی به یک حساب کاربری ادمین(یا یک حساب کاربری دیگر) تبدیل می شود. در اکثر موارد به دلیل سخت تر بودن نیازمندیهای امنیتی برای یک ادمین، مهاجم معمولا این امکان را ندارد تا نام کاربری و پسورد یک حساب کاربری با اولویت ادمین را بیابد. در این مواقع هکر می بایست یک حساب کاربری با اولویت پایین تر را برای شروع کارش انتخاب نماید. سپس از این حساب کاربری به حساب های کاربری بالاتر صعود کند. بنابراین گام اول این است که مهاجم ابتدا باید به یکی از حساب های کاربری غیر ادمین دسترسی پیدا کند و سپس به فکر افزایش سطح اولویت باشد. یکی از دلایل اهمیت افزایش اولویت این است که اگر مهاجم بخواهد ابزاری را بر یک سیستم نصب نماید، می بایست سطح دسترسی ادمین را داشته باشد.

به طور کلی دو نوع افزایش اولویت وجود دارد:

- بالابردن اولویت افقی[1]: در این نوع افزایش اولویت، یک کاربر عادی با حساب کاربری یا بدون حساب کاربری روی یک سیستم، از یک آسیب پذیری در سیستم در جهت به دست آوردن توانایی ها یا محتویات رزرو شده برای کاربر عادی دیگر سوء استفاده می کند.
- بالابردن اولویت عمودی[2]: یک کاربر با سطوح دسترسی پایین تر با داشتن یک حساب کاربری روی یک سیستم از رخنه های موجود برای کسب قابلیت ها و محتویات رزرو شده برای کاربران اولویت دار بالاتر یا کاربران ریشه سوء استفاده می کند.

برای اینکه یک مهاجم بتواند به سطح دسترسی ریشه(یا سطح کاربری دیگر) از سطح کاربری برسد، می بایست حمله اش را بر نرم افزارهایی(سرویس هایی) متمرکز کند که دارای سطح دسترسی مورد نظر باشند تا بعد از تسلط بر آن بتواند سطح اولویت آنها را قرض بگیرد. به طور کلی می توان افزایش اولویت را از سه طریق ایجاد کرد:

- سوء استفاده از اشتباهات در پیکربندی
- سوء استفاده از افراد
- حمله به یک نرم افزار

[1] Horizontal privilege escalation
[2] Vertical privilege escalation

در صورتی که در مورد پیکربندی یک نرم افزار سهل انگاری شود، مهاجم می تواند امکان این را داشته باشد که اولویت بالاتری به دست آورد. از شناخته ترین این موارد، حملات Brute-force برای پیدا کردن پسورد یک سیستم عامل است. اگر کاربر در هنگام تنظیم نام کاربری و پسورد موارد امنیتی را رعایت نکرده باشد و پسورد قوی را قرارنداده باشد حساب کاربری اش می تواند مورد سوءاستفاده قرار بگیرد.

همچنین یک مهاجم با استفاده از حملاتی مشابه حملات شبکه اجتماعی می تواند اطلاعات مورد نظر برای دسترسی به یک سطح اولویت بالاتر را بدست آورد.

دسته سوم از راههای افزایش اولویت به حملاتی مربوط می شود که تمرکز آنها بر یک نرم افزار است.در این نوع حملات هدف این است که اولویت یک نرم افزار در سطح دلخواه مهاجم با استفاده از یک حمله توسط مهاجم قرض گرفته شود. تعداد حملات زیادی را می توان در این دسته قرار داد مانند حمله به فایل های SAM و حملات سر ریز بافر.

GOAL

Memory Corruption

آسیب پذیری تخریب حافظه یکی از روش های معمول مهاجمین برای ایجاد خرابی در صحت یک نرم افزار است. یک حمله تخریب حافظه، مکانیزمی است که از آسیب پذیری های تخریب حافظه برای تغییر خصمانه اشیاء داخل فضای حافظه اجرایی آسیب پذیر یک برنامه، سوء استفاده کرده تا اینکه مهاجم بتواند رفتار برنامه را به یک شکل خاصی تغییر دهد. هنگامی که مهاجم قادر به تغییر محتویات حافظه برنامه مورد نظر به صورت دلخواه باشد، نتیجه حمله می تواند از بی ثباتی سیستم تا اجرای کد اختیاری توسط مهاجم باشد. بهره کشی از آسیب پذیری های تخریب حافظه به طور کلی در دو دسته حمله به داده -کنترلی و یا حمله به داده-خالص قرار می گیرد. حملات داده-کنترلی داده کنترلی را تخریب می کنند. یعنی حمله به داده ای که نقش مدیریت حافظه یک فرآیند در طی اجرایش را به عهده دارد انجام می شود. از این نوع داده ها می توان به اشاره گر دستورالعمل ذخیره شده[1]، اشاره گر فریم ذخیره شده، یک اشاره گر تابعی در GOT و غیره اشاره نمود. حملات داده-کنترلی، قصدشان انتقال جریان اجرایی یک پروسه هدف به داخل کد ماشینی خرابکار که به فضای آدرس آن پروسه یا به داخل دستورالعمل های موجود آن پروسه تزریق شده می باشد که با این کار پروسه وادار می شود تا آرگومان هایی را که مهاجم تامین می کند را قبول کند. حملات داده-خالص داده ای را که نقشی در محاسبات دقیق پروسه هدف دارد تخریب می کند[۵۹].

[1] Saved Instruction Pointer

برای ذکر مثال می توان به سر ریز بافر استک[1] ، حملات رشته قالبی[2]و حملات دوتایی-آزاد[3] اشاره کرد. این حملات سعی در تغییر اشیاء حافظه در استک(مثلا آدرس بازگشت) یا هیپ[4](داده مدیریت هیپ) در یک برنامه آسیب پذیر دارند.
آسیب پذیری ها با این نوع حملات، عموما به وسیله یک عدم صحت سنجی ورودی[5] در زبان برنامه نویسی C به این دلیل که برنامه نویسان این آزادی را دارند که تصمیم بگیرند چه موقع و چه طور ورودی ها را کنترل کنند ایجادمی شود. این انعطاف پذیری اغلب منجر به بهبود کارایی نرم افزار می شود. البته تعداد آسیب پذیری های ایجاد شده به وسیله خطای صحت سنجی ورودی نشان می دهد که خطاهای برنامه نویسی این نوع، آسان در موقع ایجاد و سخت برای تعمیر است[۶۰].

DOS

طی یک حمله انکار سرویس[6] یک مهاجم می تواند سیستم را با گران بار[7]کردن منابع یا ممانعت از فعالیت کاربران قانونی از دسترسی به سیستم، آن را به یک سیستم غیر قابل استفاده یا بسیار کند تبدیل نماید. این حملات می توانند بر روی یک سیستم یا بر روی کل شبکه انجام شوند.
دزدی نشست[8]یک روش هک است که یک انکار سرویس موقت را برای یک کاربر نهایی هنکامی که مهاجم یک نشست را در دست می گیرد، ایجاد می کند. مهاجم از دزدی نشست برای دست گرفتن نشست جاری بعد این که کاربر یک نشست تصدیق اصالت شده را بر قرار کرد استفاده می کند. دزدی نشست می تواند همچنین برای اجرای یک حمله فردی در میان در هنگامی که مهاجم بین سرور و کلاینت قانونی قرار گرفته و به ترافیک گوش می کند استفاده شود. به طور کلی دو طبقه بندی اصلی برای حملات انکار سرویس وجود دارد:

- حملاتی که به وسیله یک سیستم تنها، به یک هدف تنها صورت می گیرد(انکار سرویس ساده)
- حکلاتی که به وسیله چندین سیستم به یک سیستم تنها صورت می گیرد(انکار سرویس توزیع شده یا DDOS)

هدف حملات انکار سرویس کسب یک دسترسی بدون مجوز به ماشین ها یا داده نیست بلکه

[1] Stack Buffer Overflow
[2] Format String Attacks
[3] Double-free Attacks
[4] Heap
[5] Input Validation
[6] Denial of Service
[7] overload
[8] Session hijacking

هدف جلوگیری از کابران قانونی یک سرویس از اجرای آن است. یک حمله انکار سرویس ممکن است به صورت های زیر انجام گیرد:

- یک شبکه را با ترافیک سیل زده نماید تا از دسترسی به یک سرویس جلوگیری شود.
- ارتباطات بین دو ماشین قطع گردد تا اینکه از دسترسی بـه یـک سیسـتم جلـوگیری شود.
- ممانعت از دسترسی یک شخص خاص از دسترسی به یک سرویس
- خراب کردن یک سرویس برای یک سیستم یا شخص خاص

یک حمله انکار سرویس معمولا به عنوان یک حمله از روی ناچاری به حساب می آید و به عنوان یک حمله ساده در نظر گرفته می شود. زیرا در این حمله مهاجم به هیچ اطلاعاتی دست پیدا نمی کند بلکه فقط هدف را اذیت کرده و سرویس هایشان را مختل می کند. حملات انکار سرویس زمانی می توانند مخرب باشند و تاثیر زیادی بگذارند که از چندین سیستم و در یک زمان یکسان انجام گیرند. حملات DDOS می توانند به وسیله بات ها[1] و بات نت ها[2] برای حملاتی علیه یک قربانی مورد استفاده قرار گیرند. یک بات به طور خلاصه یک روبات وبی و یک برنامه نرم افزاری خود کار است که به طور خودکار و هوشمندانه عمل می کند. بات ها عامل های نرف افزاری وبی هستند که با صفحات وب تعامل بر قرار می کنند. به عنوان مثال اسپم کنندگان اغلب از بات ها برای پست کردن پیغام های اسپم روی گروه های جدید یا فرستادن میل ها بهره می گیرند. یک بات نت گروهی از سیستم های باتی است. بات نت ها چندین هدف شامل حملات DDOS، ایجاد یا سوء استفاده از رله کنندگان میل SMTP برای اسپم کردن، داده وستد جعلی، دزدی شماره سریال های نرم افزارها، شناسه لاگین ها و اطلاعات مالی مثل شماره های کارت اعتباری را به عهده می گیرند. سیستمی که تحت تسلط قرار گرفته در واقع یک قربانی ثانویه به حساب می آید.

Espionage

هدف دشمن در این دسته سرقت اطلاعات مهم است. حملاتی که به قصد به دست آوردن فایل های پسورد، کلید های دسترسی و به طور کلی اطلاعات احراز هویت در این دسته قرارمی گیرند.

- حمله برای آشکار سازی اطلاعات احراز هویت
- حمله برای آشکار سازی اطلاعات سیستم

[1] BOTs
[2] BOTNETs

- حمله برای آشکار سازی اطلاعات کاربری

Backdoor

یک درب پشتی برنامه یا مجموعه ای از برنامه های مرتبط است که یک مهاجم آن را بر روی سیستم هدف نصب می کند تا بتواند اجازه دسترسی به آن سیستم را در زمانی بعد تر داشته باشد. اضافه کردن یک سرویس جدید معمولترین تکنیک برای پنهان سازی درب های پشتی در سیستم عامل ویندوز است. قبل از نصب یک درب پشتی، یک هکر می بایست درباره سیستم تحقیق کند تا سرویس هایی را که در حال اجرا روی آن سیستم است را پیدا کند. استفاده خوب از فنون جمع آوری اطلاعات برای دانستن اینکه چه سرویس ها یا برنامه هایی در حال حاضر بر روی سیستم هدف در حال اجرا است حیاتی است. در اکثریت موارد هکر با نصب درب پشتی سرویس جدیدی را به آن سیستم اضافه می کند و به آن یک نام ناآشنا و یا حتی بهتر به آن نام یک سرویسی را که تا به حال استفاده نشده و آن به صورت دستی فعال می شود و یا کاملا غیر فعال است را می دهد. فن درب پشتی ساده ولی کارا است. هکر می تواند به داخل ماشین با حداقل مقدار دیدپذیری در لاگ های سرور وارد شود[۶۱].

Pilfering

هدف در این بخش به دست آوردن و استخراج اطلاعات از یک ماشین است. از این نوع حملات می توان به حملات تزریق SQL اشاره کرد.

فهرست مراجع

[1] E. Gandotra, D. Bansal, and S. Sofat, “Malware analysis and classification: A survey,” *J. Inf. Secur.*, vol. 2014, 2014.

[2] T. Bass, “Intrusion detection systems and multisensor data fusion,” *Commun. ACM*, vol. 43, no. 4, pp. 99–105, 2000.

[3] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 37, no. 1, pp. 32–64, 1995.

[4] M. R. Endsley, *Designing for Situation Awareness: An Approach to User-Centered Design, Second Edition*. CRC Press, 2011.

[5] L. T. Kohn, J. M. Corrigan, and M. S. Donaldson, *To err is human:: building a Safer Health System*, vol. 6. National Academies Press, 2000.

[6] M. R. Endsley, “Design and evaluation for situation awareness enhancement,” presented at the Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 1988, vol. 32, pp. 97–101.

[7] R. M. Clark, *Intelligence Analysis: estimation and prediction*. American Literary Press, 1996.

[8] X. Qin and W. Lee, “Discovering novel attack strategies from INFOSEC alerts,” in *Data Warehousing and Data Mining Techniques for Cyber Security*, Springer, 2007, pp. 109–157.

[9] J. Wu, L. Yin, and Y. Guo, “Cyber attacks prediction model based on bayesian network,” presented at the Proceedings of the 2012 IEEE 18th International Conference on Parallel and Distributed Systems, 2012, pp. 730–731.

[10] X. Qin and W. Lee, “Attack plan recognition and prediction using causal networks,” presented at the Computer Security Applications Conference, 2004. 20th Annual, 2004, pp. 370–379.

[11] S.-H. Chien and C.-S. Ho, “A Novel Threat Prediction Framework for Network Security,” in *Advances in Information Technology and Industry Applications*, Springer, 2012, pp. 1–9.

[12] J. Holsopple, S. J. Yang, and M. Sudit, “TANDI: Threat assessment of network data and information,” presented at the Defense and Security Symposium, 2006, p. 62420O–62420O.

[13] C. Cipriano, A. Zand, A. Houmansadr, C. Kruegel, and G. Vigna, “Nexat: A history-based approach to predict attacker actions,” presented at the Proceedings of the 27th Annual Computer Security Applications Conference, 2011, pp. 383–392.

[14] Z. Li, J. Lei, L. Wang, and D. Li, “A data mining approach to generating network attack graph for intrusion prediction,” presented at the Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on, 2007, vol. 4, pp. 307–311.

[15] P. Liu, W. Zang, and M. Yu, “Incentive-based modeling and inference of attacker intent, objectives, and strategies,” *ACM Trans. Inf. Syst. Secur. TISSEC*, vol. 8, no. 1, pp. 78–118, 2005.

[16] K. Tang, M. Zhao, and M. Zhou, “Cyber Insider Threats Situation

Awareness Using Game Theory and Information Fusion-based User Behavior Predicting Algorithm," *J. Inf. Comput. Sci.*, vol. 8, no. 3, pp. 529–545, 2011.
[17] F. Gao, J. Sun, and Z. Wei, "The prediction role of hidden markov model in intrusion detection," presented at the Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on, 2003, vol. 2, pp. 893–896.
[18] D. S. Fava, S. R. Byers, and S. J. Yang, "Projecting cyberattacks through variable-length markov models," *Inf. Forensics Secur. IEEE Trans. On*, vol. 3, no. 3, pp. 359–369, 2008.
[19] W. Lee, S. J. Stolfo, and P. K. Chan, "Learning patterns from unix process execution traces for intrusion detection," presented at the AAAI Workshop on AI Approaches to Fraud Detection and Risk Management, 1997, pp. 50–56.
[20] T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," *ACM Trans. Inf. Syst. Secur. TISSEC*, vol. 2, no. 3, pp. 295–331, 1999.
[21] N. Ye, Y. Zhang, and C. M. Borror, "Robustness of the Markov-chain model for cyber-attack detection," *Reliab. IEEE Trans. On*, vol. 53, no. 1, pp. 116–123, 2004.
[22] P. Jacquet, W. Szpankowski, and I. Apostol, "A universal predictor based on pattern matching," *Inf. Theory IEEE Trans. On*, vol. 48, no. 6, pp. 1462–1472, 2002.
[23] C. R. Shalizi and K. L. Shalizi, "Blind construction of optimal nonlinear recursive predictors for discrete sequences," presented at the Proceedings of the 20th conference on Uncertainty in artificial intelligence, 2004, pp. 504–511.
[24] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text compression*, vol. 348. Prentice Hall Englewood Cliffs, 1990.
[25] D. Man, Y. Wang, W. Yang, and W. Wang, "A combined prediction method for network security situation," presented at the Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on, 2010, pp. 1–4.
[26] J. Holsopple, S. J. Yang, M. Kuhl, D. Hall, R. Nagi, S. Shapiro, M. Sudit, B. Panulla, M. Kandefer, and P. Seyed, "National Center for Multisource Information Fusion," DTIC Document, 2009.
[27] E. Shahbazian, G. Rogova, and M. J. de Weert, *Harbour Protection Through Data Fusion Technologies*. Springer Netherlands, 2008.
[28] E. Little, G. Rogova, and A. Boury-Brisset, "Theoretical Foundations of Threat Ontology (ThrO) for Data Fusion Applications," TR-2005-269, Nov, 2005.
[29] B. Bennett and C. Fellbaum, *Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*. IOS Press, 2006.
[30] C. G. Chittester and Y. Y. Haimes, "Risks of terrorism to information technology and to critical interdependent infrastructures," *J. Homel. Secur. Emerg. Manag.*, vol. 1, no. 4, 2004.
[31] R. Radvanovsky and J. Brodsky, *Handbook of SCADA/Control Systems Security*. Taylor & Francis, 2013.
[32] F. L. Greitzer and D. A. Frincke, "Combining traditional cyber security

audit data with psychosocial data: towards predictive modeling for insider threat mitigation," in *Insider Threats in Cyber Security*, Springer, 2010, pp. 85–113.

[33] S. L. Hald and J. M. Pedersen, "An updated taxonomy for characterizing hackers according to their threat properties," presented at the Advanced Communication Technology (ICACT), 2012 14th International Conference on, 2012, pp. 81–86.

[34] J. Moteff, "Risk management and critical infrastructure protection: Assessing, integrating, and managing threats, vulnerabilities and consequences," 2005.

[35] S.-H. Chien, E.-H. Chang, C.-Y. Yu, and C.-S. Ho, "Attack subplan-based attack scenario correlation," presented at the Machine Learning and Cybernetics, 2007 International Conference on, 2007, vol. 4, pp. 1881–1887.

[36] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Lead. Issues Inf. Warf. Secur. Res.*, vol. 1, p. 80, 2011.

[37] M. I. Center, "APT1: Exposing one of China's cyber espionage units," *Mandian Com*, 2013.

[38] B. Bryant, "A Method for Implementing Intention-Based Attack Ontologies with SIEM Software," 2014.

[39] S. J. Yang, A. Stotz, J. Holsopple, M. Sudit, and M. Kuhl, "High level information fusion for tracking and projection of multistage cyber attacks," *Inf. Fusion*, vol. 10, no. 1, pp. 107–121, 2009.

[40] M. E. Kuhl, J. Kistner, K. Costantini, and M. Sudit, "Cyber attack modeling and simulation for network security analysis," presented at the Proceedings of the 39th Conference on Winter Simulation: 40 years! The best is yet to come, 2007, pp. 1180–1188.

[41] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer, "Comprehensive approach to intrusion detection alert correlation," *Dependable Secure Comput. IEEE Trans. On*, vol. 1, no. 3, pp. 146–169, 2004.

[42] K. Kendall, "A database of computer attacks for the evaluation of intrusion detection systems," DTIC Document, 1999.

[43] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Comput. Netw.*, vol. 34, no. 4, pp. 579–595, 2000.

[44] "KDDCup," *KDDCupdata*. 1999.

[45] "DEFCONconference," *DEFCON capture the flag(CTF) contest.* .

[46] B. Sangster, T. O'Connor, T. Cook, R. Fanelli, E. Dean, C. Morrell, and G. J. Conti, "Toward Instrumenting Network Warfare Competitions to Generate Labeled Datasets.," presented at the CSET, 2009.

[47] "CDX_2009_Network_USMA.," 2009.

[48] "مجموعه گزارشات مرکز پژوهشی علوم وفناوری پردازش و ادغام داده و تصمیم," ۱۳۹۴ ۱۳۹۳.

[49] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia: Learning probabilistic automata with variable memory length," *Mach. Learn.*, vol. 25, no. 2–3, pp. 117–149, 1996.

[50] F. M. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree

weighting method: basic properties," *Inf. Theory IEEE Trans. On*, vol. 41, no. 3, pp. 653–664, 1995.
[51] K. M. Curewitz, P. Krishnan, and J. S. Vitter, "Practical prefetching via data compression," presented at the ACM SIGMOD Record, 1993, vol. 22, pp. 257–266.
[52] S. Bunton, "A generalization and improvement to PPM's blending," *UW-CSE Tech. Rep. UW-CSE-97-01-10 Univ. Wash.*, 1997.
[53] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text compression*. Prentice-Hall, Inc., 1990.
[54] P. J. Cowans, "Probabilistic document modelling," 2006.
[55] C. Scheffler, "Improving PPM," 2008.
[56] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, 2002.
[57] E. Ukkonen, "On-line construction of suffix trees," *Algorithmica*, vol. 14, no. 3, pp. 249–260, 1995.
[58] S. McClure, J. Scambray, and G. Kurtz, *Hacking Exposed, Sixth Edition: Network Security Secrets& Solutions*. McGraw-Hill Education, 2009.
[59] J. Lopez, R. Setola, and S. Wolthusen, *Critical Infrastructure Protection: Advances in Critical Infrastructure Protection: Information Infrastructure Models, Analysis, and Defense*. Springer, 2012.
[60] C. Kil and North Carolina State University, *Mechanisms for Protecting Software Integrity in Networked Systems*. North Carolina State University, 2008.
[61] K. Graves, *CEH certified ethical hacker study guide*. John Wiley & Sons, 2010.
[62] M. Gregg, S. Watkins, G. Mays, C. Ries, R. M. Bandes, and B. Franklin, *Hack the Stack: Using Snort and Ethereal to Master The 8 Layers of An Insecure Network*. Elsevier Science, 2006.
[63] N. Cuppens-Boulahia, F. Cuppens, S. Jajodia, A. A. E. Kalam, and T. Sans, *ICT Systems Security and Privacy Protection: 29th IFIP TC 11 International Conference, SEC 2014, Marrakech, Morocco, June 2-4, 2014. Proceedings*. Springer, 2014.

MUT

Degree: Master of Science (M.Sc.)

Title: The projection of the Multistage Network Attacks based on the Attack Behaviors

Author: Javad Karimi

Supervisor: Dr. Ali Jabar Rashidi

Department: Institute for Research on Information and Communications Security

Date: November 03, 2015

Abstract:

Network Security rely on the systems such as intrusion detection systems extremely, while the purpose of these tools is identifying malicious traffics and uses of the computers. Due to the high volume of alerts began moving towards the correlation. But the problem of not awareing of the analysts from real network situation and possible attacks on the network is remained still. The aim of this research is increasing of the current knowledge of analysts from current network attacks and more likely attacks in the future. In this research, we used Dataset of CDX2009. We correlated this dataset and produced attack tracks from those. We also used a new template in the process of attack track production instead of templates like Kill Chain. Finally, we predict future behaviors or stage of the attack by using of sequence characteristic of the attack tracks and Variable Length Markov Model. The results of this research show that the Prediction by Partial Matching algorithm type A and S, with 60 percent data train respectively have been able to predict in 50.7 and 53.5 percent of cases correctly.

Keywords: Situation Awareness, Projection, Guidance Template, Attack Track, Markov Chain, Prediction by Partial Matching

In The Name Of Allah

Malek-e-Ashtar University of Technology

Academic Complex of Information, Communications and Security Technologies
Institute for Research on Information and Communications Security (IRICS)

The projection of the Multistep Network Attacks based on the Behavior

A Thesis
Submitted in partial fulfillment of the requirements
For the degree of Master of Science (M.Sc.) in
Information Technology Engineering, Secure Information

By:
Javad Karimi

Evaluated and approved by the thesis committee, on November 03, 2015

No.	Title	Responsibility	Signature
1	Dr. Ali Jabar Rashidi	Supervisor	
2	Dr. Kourosh Dadashtabar	Advisor	
3	Dr. Farid Samsami	Examiner	
4	Dr. Morteza Barrari	Examiner	
5	Dr. Mohammad FakhrDanesh	Department Graduate Coordinator	

The projection of the Multistep Network Attacks based on the Behavior

A Thesis
Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science (M.Sc.) in
Information Technology Engineering, Secure Information
from the
Institute for Research on Information and Communications Security (IRICS)
Academic Complex of Information, Communications and Security Technologies
Malek-e-Ashtar University of Technology

By

Javad Karimi

Supervised by

Dr. Ali Jabar Rashidi

Advised by

Dr. Kourosh Dadashtabar

November 2015