علوم و مهندسی کامپیوتر
نشریه علمی پژوهشی انجمن کامپیوتر ایران
مجلد ۱۱، شماره ۱ (الف)، ۱۳۹۲
صفحات ۵۴-۶۰
مقاله عادی

پیش‌بینی نفوذ به سیستم‌ها بوسیله‌ی متن کاوی

عاطفه خزاعی اشکان سامی

دانشکده مهندسی برق و کامپیوتر، دانشگاه شیراز، شیراز، ایران

چکیده

وجود بسیاری از مشخصه‌ها در نرم‌افزارها می‌تواند باعث آسیب‌پذیر شدن سیستم‌ها شود. این آسیب‌پذیری‌ها راه نفوذی برای مهاجمان هستند. سالانه میلیون‌ها دلار در سرتاسر دنیا صرف مقابله با نفوذ مهاجمان به سیستم‌های نرم‌افزاری می‌شود. همواره محققان به دنبال یافتن روش‌هایی برای رده‌بندی این آسیب‌پذیری‌ها بوده‌اند تا بتوانند از میزان هزینه‌های مالی و انسانی که برای نظارت بر آسیب‌پذیری‌ها صرف می‌شود بکاهند. بیشتر تحقیق‌های پیشین از روش‌های تجربی و آماری برای حل این مسئله استفاده کرده‌اند اما در این تحقیق ما به دنبال پیش‌بینی پاسخ این سوال با استفاده از ابزارهای متن کاوی هستیم: آیا از گزارشات آسیب‌پذیری موجود در یک سیستم می‌توان فهمید که آسیب‌پذیری منجر به نفوذ مهاجمان خواهد شد؟ در این تحقیق برداری از مشخصات برای هر آسیب‌پذیری از میان گزارش‌های مستند استخراج و مدلی برای پیش‌بینی نفوذپذیری ارائه شده است. نتایج بهبود قابل قبولی را در دقت پیش‌بینی احتمال نفوذها نسبت به کارهای مشابه پیشین نشان می‌دهد.

کلمات کلیدی: آسیب‌پذیری، متن کاوی، نفوذ به سیستم.

۱- مقدمه

یک خطا، نقص، رفتار، خروجی یا رویداد درون برنامه، سیستم یا دستگاه که بتواند منجر به آسیب ضمنی یا صریح در محرمانگی، تمامیت یا دسترسی‌پذیری یک سیستم شود، آسیب‌پذیری سیستم محسوب می‌گردد. هر ساله هزاران آسیب‌پذیری کشف و گزارش می‌شوند. وجود آسیب‌پذیری‌ها در سیستم‌ها می‌تواند منجر به نفوذ مهاجمان شود. در سال‌های اخیر تحقیقات زیادی بر روی آسیب‌پذیری‌های سیستم‌ها انجام شده است و محققان از دیدگاه‌های مختلفی از جمله موارد زیر به بررسی آسیب‌پذیری‌ها پرداخته‌اند:

- نقش افراد و فرآیندهای موثر بر آسیب‌پذیری نرم‌افزارها (برای مثال [۱]، [۲] و [۳]): گروه‌های مختلفی از افراد، به صورت مستقیم یا غیرمستقیم، در ارتباط با آسیب‌پذیری‌ها هستند. از جمله این افراد می‌توان به کاربران، مهاجمان و سازندگان سیستم‌های نرم‌افزاری اشاره کرد. کشف و افشای آسیب‌پذیری‌ها، به اشکال مختلفی بر فعالیت این افراد تاثیر می‌گذارد. این تاثیر به صورت معکوس نیز می‌باشد، یعنی فعالیت‌های این افراد بر کشف و افشای آسیب‌پذیری موثر است.
- بررسی پایگاه داده‌های آسیب‌پذیری‌ها و تاثیر ساختار و دقت آن‌ها بر نتایج تحقیقات (برای مثال [۴]): پایگاه داده‌های مختلفی وجود دارد که اطلاعات آسیب‌پذیری‌ها در آن‌ها ثبت می‌شود. این پایگاه داده‌ها از نظر دقت ذخیره اطلاعات با یکدیگر متفاوت هستند. نتایج [۴] نشان داد که تفاوت‌های موجود در پایگاه داده‌ها می‌تواند، گاهی منجر به نتیجه‌گیری‌های کاملاً متفاوت شود.
- دسته‌بندی آسیب‌پذیری‌ها (برای مثال [۷] و [۸]): اگرچه تعداد بسیار زیادی آسیب‌پذیری وجود دارد، اما علت‌های وقوع آسیب‌پذیری‌ها نسبت به تعدادشان بسیار محدود است. تحقیقات زیادی برای اندازه‌گیری میزان شباهت میان آسیب‌پذیری‌ها و ارائه دسته‌بندی‌هایی برای آن‌ها انجام شده است، ارائه این دسته‌بندی‌ها، انجام مطالعات و ارائه راه حل برای آسیب‌پذیری‌ها را ساده‌تر می‌کند.
- تمرکز بر نوع خاصی از آسیب‌پذیری‌ها و ارائه راه حل برای آن‌ها (برای مثال [۹] و [۱۰]): ارائه راه حلی که برای همه آسیب‌پذیری‌ها مناسب باشد، غیر ممکن است. بنابراین محققان معمولاً با در نظر گرفتن یک نوع آسیب‌پذیری، از آسیب‌پذیری‌های یک سیستم مشخص، به ارائه راه حل برای آن می‌پردازند.

• ارائه راه حل‌هایی برای پیش‌گیری از حمله مهاجمان تا زمان ارائه بسته نرم‌افزاری اصلاح شده (برای مثال [۱۱]): فاصله زمانی میان کشف تا ارائه بسته نرم‌افزاری اصلاح شده، فرصتی برای نفوذ مهاجمان به سیستم آسیب‌پذیر است، این تحقیق به دنبال راه حلی برای به تاخیر انداختن بهره‌کشی از سیستم، به منظور ایجاد فرصتی برای ارائه بسته اصلاح شده است.

هر یک از تحقیقاتی که تاکنون روی آسیب‌پذیری‌ها انجام شده است، به نوعی به دنبال کمک به ناظران و کاربران سیستم جهت مقابله با آسیب‌پذیری‌ها بوده‌اند. با توجه به این نکته که تاکنون هیچ ابزار مطمئنی که بتواند جایگزین نیروی انسانی در نظارت بر سیستم‌ها بشود، ارائه نشده است و حتی با در نظر گرفتن ساختارهای ناپایدار آسیب‌پذیری‌ها، ارائه چنین ابزاری غیر ممکن به نظر می‌رسد، بخشی از نیروی انسانی در سازمان‌ها وظیفه نظارت مستقیم بر سیستم‌ها و کنترل دسترسی‌ها را برعهده دارند، تا بتوانند از نفوذ مهاجمانی که با استفاده از آسیب‌پذیری‌های سیستم قصد سوء استفاده از سیستم‌هایشان را دارند، جلوگیری کنند. سیستم‌های کامپیوتری معمولاً به طور همزمان دارای چندین آسیب‌پذیری هستند، با توجه به محدود بودن امکانات مالی و نیروی انسانی عموماً امکان مقابله همزمان با تمام آسیب‌پذیری‌ها ممکن نمی‌باشد. با در نظر گرفتن این نکته که ساختار برخی آسیب‌پذیری‌ها به گونه‌ای است که مهاجمان نمی‌توانند از آن‌ها بهره کشی کنند، این سوال برای ناظران، کاربران و حتی سازندگان سیستم‌ها مطرح می‌شود که کدام آسیب‌پذیری‌ها مستعد بهره کشی هستند و باید در اولویت قرار گیرند؟ هدف ما در این تحقیق پاسخ دهی به این سوال با استفاده از گزارشات مربوط به آسیب‌پذیری‌ها می‌باشد.

پایگاه داده‌های مختلفی وجود دارند که گزارشات کشف آسیب‌پذیری‌ها در آن‌ها ثبت می‌شود. برخی از این پایگاه داده‌ها متعلق به سازندگان نرم‌افزارها هستند که گزارشات آسیب‌پذیری‌های محصولات خود را در آن‌ها ثبت می‌کنند، معمولاً این پایگاه داده‌ها در دسترس همه‌ی محققان نیست. پایگاه داده‌های عمومی نیز وجود دارند که گزارشات آسیب‌پذیری محصولات متنوع با کاربری بالا را در خود ثبت می‌کنند. پایگاه داده‌های عمومی، در دسترس همه‌ی محققان می‌باشد و نتایج حاصل از تحقیق بر روی این پایگاه داده‌ها قابلیت تعمیم بیشتری نیز دارند. در این پژوهش، با استفاده گزارشات مستند در پایگاه داده‌های عمومی و با استفاده از متن کاوی، به دنبال ارائه روشی خودکار برای پاسخ دهی به این سوال هستیم: "کدام آسیب‌پذیری‌ها منجر به بهره کشی از سیستم خواهند شد؟". با استفاده از روش‌های متن کاوی برداری از مشخصات، از میان گزارشات موجود در پایگاه داده‌ها استخراج شده است و پس از آن با استفاده از روش‌های کلاس‌بندی، کلاس‌بندی کننده‌هایی آموزش داده شدند و پاسخ این سوال را با دقت خوبی پیش‌بینی کردند.

در ادامه مقاله ابتدا به کارهای قبلی انجام شده در این زمینه اشاره می‌شود، سپس درباره داده‌های تحقیق، نحوه استخراج مشخصه‌ها از آن‌ها، نحوه انجام آزمایشات و نتایج‌شان توضیح داده خواهد شد.

۲- پیشینه تحقیق

تعداد کمی از سازمان‌ها وجود دارند که منابع مالی و انسانی لازم برای مقابله با همه آسیب‌پذیری‌هایی که سیستم‌هایشان را تهدید می‌کنند، در اختیار دارند. در زمینه اولویت دهی آسیب‌پذیری‌ها تحقیقات زیادی انجام شده است و معیارهای مختلفی برای انجام این اولویت بندی ارائه شده است. برای مثال می‌توان به CERT/CC [۱۷]، SANS [۱۸]، سیستم نمره دهی مایکروسافت [۱۹] و CVSS [۲۰] اشاره کرد. اما برای اینکه بتوانیم آسیب‌پذیری‌های مختلف را با هم مقایسه کنیم، نیاز به یک معیار استاندارد داریم. در میان این روش‌های رده‌بندی CVSS (Common Vulnerability Scoring System) تقریباً به عنوان یک استاندارد از جانب سازندگان و محققان پذیرفته شده است. به منظور بهبود این روش تحقیقات زیادی انجام شده است، از آن جمله می‌توان به [۵] و [۶] اشاره کرد. CVSS از سه گروه معیار تشکیل شده است: پایه (Base)، زمانی (Temporal) و محیطی (Environmental). معیار پایه بدون در نظر گرفتن موقعیت زمانی و محیطی آسیب‌پذیری به ارزیابی آن‌ها می‌پردازد، درحالی که در ارزیابی با معیار زمانی، زمان و در معیار محیطی، مکان آسیب‌پذیری نیز در نظر گرفته می‌شود. با وجود معیارهای زمانی و محیطی، عموماً معیار پایه که نشان دهنده خصوصیات ذاتی یک آسیب‌پذیری است، در تحقیقات و پایگاه داده‌ها استفاده می‌شود [۲۰].

با وجود تحقیقات فراوانی که برروی CVSS انجام شده است، مهران بزرگی و همکارانش در [۱۲] نشان دادند که نمره پایه CVSS یک معیار مناسب برای پیش‌بینی اینکه آیا یک آسیب‌پذیری منجر به بهره کشی از سیستم خواهد شد یا نه، نیست. بزرگی و همکارانش در [۱۲] برای اولین بار از روش‌های آموزش ماشین و داده کاوی برای پیش‌بینی بهره کشی از آسیب‌پذیری‌ها استفاده کردند. در [۱۲] اغلب مشخصات باینری هستند و از بین متن‌های گزارشات آسیب‌پذیری‌ها، استخراج شده‌اند. این مشخصات بیانگر وجود یا عدم وجود کلمات، در متن گزارشات هستند. در [۱۲] پس از استخراج مشخصات و ساختن بردار مشخصات برای هر آسیب‌پذیری، کلاس بندی کننده‌ای آموزش داده شد که امکان وقوع بهره کشی از آسیب‌پذیری‌ها را با دقتی بهتر از CVSS پیش‌بینی می‌کرد. می‌توان گفت در زمینه استفاده از متن کاوی برای پیش‌بینی بهره کشی از آسیب‌پذیری‌ها، تحقیق مهران بزرگی و همکارانش تنها کار پیشین تحقیق ما محسوب می‌شود. در قسمت‌های مختلف مقاله به تفاوت‌های میان کار بزرگی و کار خودمان اشاره خواهیم کرد.

۳- داده‌ها

نتایج تحقیقات [۴] در زمینه بررسی منابع آسیب‌پذیری‌ها نشان داد که ضعیف بودن برخی گزارش‌ها می‌تواند منجر به نتایج ناصحیحی در تحقیقات شود. بنابراین به منظور به دست آوردن داده‌های مطمئن‌تر، از دو پایگاه داده معروف در زمینه آسیب‌پذیری‌ها به نام‌های OSVDB (Open Source Vulnerability Database) و CVE (Common Vulnerabilities and Exposure) استفاده کردیم.

OSVDB یک پایگاه داده مستقل و منبع باز است و در حال حاضر با بیش از ۷۰ هزار گزارش آسیب‌پذیری در دسترس عموم می‌باشد. این پایگاه داده دارای ۱۵ جدول است، که در این جدول‌ها، برای هر آسیب‌پذیری اطلاعاتی مانند توصیف و راه حل مقابله با آن، مشخصات محصول و سازنده محصول، تاریخ‌های کشف و افشاء آسیب‌پذیری محصول، تاریخ نفوذ به سیستم و... جمع آوری شده است [۱۴].

CVE، بیش از آنکه یک پایگاه داده محسوب شود، یک استاندارد نام گذاری برای آسیب‌پذیری‌ها است، که در حال حاضر، حاوی بیش از ۴۵ هزار گزارش آسیب‌پذیری می‌باشد [۱۳]. تنوع اطلاعات موجود برای آسیب‌پذیری‌ها در این پایگاه داده نسبت به OSVDB کمتر است و بیشتر بر توصیف آسیب‌پذیری تمرکز دارد و در حقیقت، در این تحقیق از CVE برای تقویت داده‌های OSVDB استفاده شده است.

برای انجام این تحقیق از گزارشات آسیب‌پذیری‌های مشترک موجود در هر دو پایگاه داده OSVDB و CVE تا پایان سال ۲۰۱۰ میلادی استفاده شده است و تعداد این گزارش‌های مشترک موجود ۱۲۷۱۰ مورد می‌باشد.

۴- استخراج مشخصه

پس از دریافت پایگاه داده‌ها و انجام مراحل اولیه آماده‌سازی داده‌ها، باید از میان داده‌های موجود برداری از مشخصات، برای گزارشات آسیب‌پذیری‌ها استخراج شود. شکل شماره ۱ شمای کلی مراحل انجام این تحقیق را نشان می‌دهد. در این شکل مراحل استخراج مشخصات نیز به شکل خلاصه نمایش داده شده است.

شکل ۱- شمای کلی مراحل انجام تحقیق

پیش از این نیز اشاره شد که، بخش زیادی از اطلاعات موجود در پایگاه داده‌های مورد استفاده این تحقیق به صورت متن می‌باشد. بنابراین با استفاده از ابزارها و روش‌های متداول متن کاوی اقدام به استخراج مشخصه‌ها از میان این متن‌ها کردیم.

به منظور استخراج مشخصه‌ها ابتدا لغات موجود در متن‌ها استخراج و از میان آن‌ها کلمات توقف (stop words) حذف شد. کلمات توقف مجموعه‌ای از کلمات هستند که تقریباً در تمامی متن‌ها تعدادی از آن‌ها به چشم می‌خورد، از نظر معنایی، مفهوم قابل توجهی را منتقل نمی‌کنند و استفاده بسیار زیاد آن‌ها باعث می‌شود که نقش تمیزدهندگی در بین متن‌ها را نیز نداشته باشند. از جمله کلمات توقف می‌توان به حروف اضافه و ضمایر اشاره کرد. پس از حذف کلمات توقف، کلمات باقی مانده ریشه‌یابی شدند. ریشه‌یابی به این معناست که کلیه‌ی مشتقات کلمه، به عنوان یک کلمه در نظر گرفته شوند، به عنوان مثال رفتم، رفتی و رفت، هر سه به عنوان کلمه رفت حساب شوند. پس از انجام ریشه‌یابی برای کلمات باقی مانده، مقدار TF-IDF محاسبه و به عنوان مقدار مشخصه استخراج شد. لازم به ذکر است که وزن TF-IDF (Term Frequency – Inverse Document Frequency) اغلب در بازیابی اطلاعات و متن کاوی مورد استفاده قرار می‌گیرد و یک روش اندازه گیری آماری برای ارزیابی اهمیت یک کلمه در یک سند می‌باشد.

به منظور تشکیل بردار مشخصات برای هر نمونه از آسیب‌پذیری‌ها اگر واژه مورد نظر در متن موجود بود مقدار TF-IDF و در غیر این صورت مقدار صفر برای آن مشخصه در نظر گرفته می‌شد. برای استخراج این مشخصه‌ها از ابزار WVT (Word Vector Tool) [۱۶] که به زبان جاوا می‌باشد، استفاده شد.

تعداد محدودی مشخصه نیز از فیلدهای غیر متنی استخراج شد، که عموماً بیان‌گر مقدار یک صفت برای آسیب‌پذیری می‌باشند. مقادیر همه این مشخصات نیز همانند اعداد TF-IDF اعداد اعشاری نرمال شده بین صفر و یک می‌باشند. جدول شماره ۱ مشخصات استخراج شده و تعداد آن‌ها را نشان می‌دهد. همان طور که در این جدول مشخص است در نهایت برای هر نمونه آسیب‌پذیری بردار مشخصاتی با طول ۳۱۸۳۹ به دست آمد. این تعداد مشخصه نسبت به مشخصه‌های [۱۲] تقریباً یک سوم می‌باشد. کاهش تعداد مشخصات در این تحقیق نسبت به [۱۲] یک ویژگی مثبت برشمرده می‌شود، زیرا باعث افزایش سرعت فرآیندهای آموزش و تست می‌گردد.

جدول ۱- مشخصات استخراج شده از نمونه‌های آسیب‌پذیری

صفت مورد استفاده	تعداد مشخصه‌ها	پایگاه داده
Company-URL	۳۰۸	OSVDB
Company	۳۳۶	OSVDB
Comments	۱۴۳	CVE
Description	۷۰۴۸	CVE
Phase	۲	CVE
Reference	۴۹۹۲	CVE
Votes	۲۲	CVE
Description	۳۵۶۵	OSVDB
Email	۷۶۸	OSVDB
Manual-Notes	۲۰۰۰	OSVDB
Author-Name	۱۰۴۲	OSVDB
Products-Name	۱۴۲۵	OSVDB
Short-Description	۳۰۲۴	OSVDB
Solution	۸۹۸	OSVDB
T-Description	۹۵۱	OSVDB
Title	۳۸۴۸	OSVDB
Status	۲	CVE
Ext-refrences	۳۸	OSVDB
Obj-affect-name	۴	OSVDB
Version-Name	۱۳۲۹	OSVDB
Vendor-Name	۹۴	OSVDB
مجموع	**۳۱۸۳۹**	

۵- آزمایشات و نتایج

در این بخش نحوه انجام آزمایشات و نتایج حاصل از آن‌ها ارائه شده است. برای کلاس بندی نمونه‌ها از SVM استفاده شد. کلاس‌بندی کننده SVM (Support Vector Machines) با فرض اینکه دسته‌ها به صورت خطی جدایی پذیر هستند به دنبال پیدا کردن اَبَر صفحه‌ای، با حداکثر حاشیه که نمونه‌های کلاس‌های مختلف را از یکدیگر جدا کند، است. در مسائلی که به صورت خطی جداپذیر نباشند، داده‌ها به فضایی با ابعاد بیشتر نگاشت پیدا می‌کنند، تا بتوان آن‌ها را در این فضای جدید به صورت خطی جدا نمود. پیاده سازی‌های مختلفی از SVM موجود است. با توجه به زیاد بودن تعداد مشخصات در این تحقیق از پیاده سازی LIBLINEAR [۱۵] استفاده کردیم. SVM خطی برای حالتی که طول بردار مشخصات نسبت به تعداد نمونه‌ها زیاد باشد، مناسبتر است.

۵-۱- روش کار

همان طور که پیش از این اشاره شد در این تحقیق می‌خواهیم پاسخ این سوال را که "آیا آسیب‌پذیری موجود در سیستم منجر به نفوذ مهاجمان به سیستم خواهد شد یا نه؟" را پیش‌بینی کنیم. در ادامه کار به نمونه‌های موجود برچسب‌های مثبت و منفی اختصاص داده شد. برچسب مثبت به این معناست که آسیب‌پذیری موجود، منجر به بهره کشی از سیستم می‌شود و برچسب منفی عکس این مسئله را نشان می‌دهد.

برچسب‌های مثبت و منفی بر اساس وضعیت دسته بندی بهره‌کشی (Exploit Classification) موجود در پایگاه داده OSVDB مشخص می‌شود. در حال

حاضر دسته بندی بهره‌کشی دارای مقادیر عمومی (public)، شایعه شده (rumored)، مجهول (unknown)، کرم خورده (wormified)، تجاری (commercial) و خصوصی (private) می‌باشد. با توجه به این نکته که OSVDB تغییراتی را در جهت بهبود پایگاه داده خود ایجاد کرده و مقادیر فعلی دسته بندی بهره‌کشی با مقادیر مورد استفاده در [۱۲] متفاوت است، از روش برچسب زنی [۱۲] نمی‌توان استفاده کرد. با در نظر گرفتن این تغییر، برای برچسب زدن به آسیب‌پذیری‌ها، با مهران بزرگی (دانشجوی دانشگاه کالیفرنیا) که یکی از نویسندگان [۱۲] است و برایان مارتین (Brian Martin) که یکی از مدیران OSVDB است، مشورت کردیم. پیشنهاد مهران بزرگی این بود که همه‌ی نمونه‌ها به جز نمونه‌هایی که دسته بندی بهره‌کشی آن‌ها مجهول است، برچسب مثبت و نمونه‌های مجهول برچسب منفی بگیرند [۲۱]. اما نظر برایان مارتین درباره نحوه برچسب زنی به نمونه‌ها با نظر مهران بزرگی متفاوت است. به نظر او نمونه‌های شایعه شده نیز باید برچسب منفی بگیرند، زیرا این دسته نمونه‌هایی هستند که بهره کشی از آن‌ها شایعه شده است، ولی گزارش‌های نفوذ موجود برای این آسیب‌پذیری‌ها چندان قابل استناد نمی‌باشد [۲۲]. با وجود اینکه در [۱۲] نمونه‌های شایعه شده برچسب مثبت داشتند، اما با اولویت دادن به نظر برایان مارتین داده‌ها را به صورتی که در جدول ۲ نشان داده شده است برچسب زنی کردیم.

جدول ۲- نحوه برچسب زنی به نمونه‌ها

نوع Exploit Classification	تعداد نمونه‌ها	برچسب
Exploit Public	۸۵۲۶	+۱
Exploit Rumored	۱۲۵۴	-۱
Exploit Unknown	۲۳۵۶	-۱
Exploit Wormified	۴	+۱
Exploit Commercial	۳۰۴	+۱
Exploit Private	۲۶۶	+۱
مجموع	**۱۲۷۱۰**	

همان‌طور که در جدول ۲ نیز مشخص است تعداد ۹۱۰۰ نمونه با برچسب مثبت و ۳۶۱۰ نمونه با برچسب منفی به دست آمده است.

در انجام آزمایشات از دسته بندی کننده SVM همراه با روش بگینگ (bagging) استفاده شد. در روش بگینگ، چند کلاس‌بندی کننده با زیر مجموعه‌هایی از داده‌ها، آموزش داده می‌شوند و برای پیش‌بینی بوسیله‌ی این کلاس‌بندی کننده‌ها، بین آن‌ها رأی گیری می‌شود و نتیجه رأی پیش‌بینی را مشخص می‌کند. تجربه نشان داده است که این روش نسبت به حالتی که فقط از یک کلاس بندی کننده استفاده می‌شود، دقت بالاتری دارد.

برای هر آزمایش ۵ دسته بندی کننده SVM را که هر کدام از آن‌ها با ۳۰۰۰ نمونه که به صورت تصادفی از میان نمونه‌های موجود انتخاب شده بودند آموزش دادیم. واضح است انتخاب تصادفی همراه با جایگذاری از میان نمونه‌ها منجر به همپوشانی در میان داده‌های آموزش کلاس بندی کننده‌های مختلف می‌شود. در ادامه برای سنجش میزان کارایی، نمونه‌هایی که در هیچ دسته بندی کننده‌ای برای آموزش استفاده نشدند را به عنوان داده تست به کلاس بندی کننده‌ها دادیم. سپس از میان کلاس بندی کننده‌ها، برای برچسب زنی به داده‌های تست، رای گیری شد و نتیجه رای که ۱+ یا ۱- بود نشان دهنده پیش‌بینی دسته بندی کننده‌ها درباره نفوذ یا عدم نفوذ به سیستم می‌باشد. لازم به ذکر است اعدادی که در ادامه مقاله ارائه می‌شود، میانگینی از نتایج چندین بار تکرار آزمایشات است.

۵-۲- پیش‌بینی برون خط (offline) نفوذها

در بخش قبل درباره نحوه برچسب زنی نمونه‌ها توضیح داده شد. داده‌ها به روش ارائه شده در جدول ۲ برچسب زده شدند و آزمایش‌ها به روشی که در قسمت قبل توضیح دادیم انجام شد. جدول ۳ خلاصه‌ای از نتایج حاصل از این آزمایشات را نشان می‌دهد.

جدول ۳- دقت پیش‌بینی در آزمایش برون خط

	داده‌های آموزش	داده‌های تست
تعداد نمونه‌ها	۳۰۰۰ (برای هر کلاس بندی کننده)	۳۳۲۵ (میانگین تعداد در آزمایشات)
منفی صحیح (True Negative)	٪۹۹.۵۵	٪۸۱.۵۱
مثبت صحیح (True Positive)	٪۹۹.۶۹	٪۸۹.۰۱
منفی غلط (False Negative)	٪۰.۴۵	٪۱۸.۴۹
مثبت غلط (False Positive)	٪۰.۳۱	٪۱۰.۹۹
دقت	**٪۹۹.۶۵**	**٪۸۷.۱۱**

در جدول ۳ مثبت صحیح به معنای نمونه‌های مثبتی است که به درستی مثبت تشخیص داده شده اند، منفی صحیح نیز نمونه‌هایی هستند که به درستی منفی پیش‌بینی شده‌اند. اما مثبت غلط نمونه‌های منفی که به اشتباه به عنوان نمونه‌های مثبت پیش‌بینی شدند، را نشان می‌دهند و این به این معناست که یک آسیب‌پذیری که منجر به بهره‌کشی نمی‌شود به اشتباه به عنوان یک آسیب‌پذیری خطرناک پیش‌بینی شده است. منفی غلط به نمونه‌های مثبتی اشاره دارد که به اشتباه به عنوان نمونه منفی شناخته شده‌اند، این یعنی یک آسیب‌پذیری که منجر به بهره‌کشی خواهد شد، به اشتباه به عنوان یک آسیب‌پذیری امن پیش‌بینی شده است. همان طور که در این جدول مشخص است دقت حاصل از این آزمایشات برای داده‌های تست تقریباً ۸۷٪ می‌باشد که این مقدار از نتایج [۱۲] ۲.۶۹٪ کمتر است. در این تحقیق برای اینکه اثر برچسب زنی داده‌های شایعه شده بر دقت پیش‌بینی را بررسی کنیم، این آزمایش را در حالتی که همانند [۱۲] نمونه‌های شایعه شده برچسب مثبت داشتند، نیز تکرار شد. نتایج آن دقت ۹۰.۳۸٪ را برای داده‌های تست نشان داد، که این مقدار از دقت به دست آمده در [۱۲] بیشتر است و این مسئله بیانگر برتری نتایج حاصل از این آزمایش، در این تحقیق نسبت به [۱۲] می‌باشد.

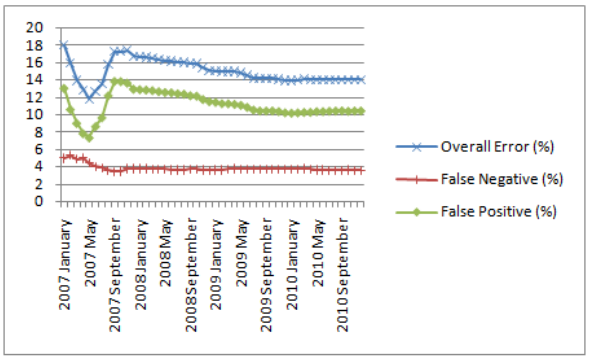
۵-۳- پیش‌بینی برخط (online) نفوذها

آزمایشات برون خط که در قسمت قبل توضیح داده شد بیان‌گر پتانسیل گزارش‌های آسیب‌پذیری برای پیش‌بینی امکان نفوذ به سیستم‌ها می‌باشد. ولی در دنیای واقعی گزارش‌های آسیب‌پذیری و نتایج نفوذ و عدم نفوذ مهاجمان به سیستم‌ها در گذر زمان و به تدریج به دست می‌آید. بنابراین باید کلاس‌بندی کننده را با داده‌های موجود آموزش داد تا برای پیش‌بینی نمونه‌هایی که پس از آن به دست می‌آیند، استفاده شود. گذر زمان نتیجه واقعی نفوذ یا عدم نفوذ برای نمونه‌های تست را مشخص می‌کند و پس از آن می‌توان از این نمونه‌ها نیز برای

آموزش کلاس بندی کننده‌ها استفاده و آن‌ها را در زمان‌های مختلف به روزرسانی کرد.

در این آزمایش یک وضعیت برخط شبیه سازی شده است. یک تاریخ به عنوان مبداء در نظر گرفته شد و کلاس بندی کننده‌ها با نمونه‌های متعلق به قبل از این تاریخ آموزش داده شدند. سپس داده‌های ماه بعد، به عنوان نمونه‌های تست به این کلاس بندی کننده‌ها نشان داده شدند. پس از آن کلاس‌بندی کننده‌ها مجدد با مجموع نمونه‌های جدید به دست آمده از ماه تست و نمونه‌های قبل از آن آموزش داده شدند و این روند ماه به ماه، برای یک بازه زمانی چند ساله، تکرار شد.

برای بررسی کارایی در این آزمایش همانند [۱۲] از خطای تجمعی استفاده شد. برای اینکه بتوانیم کارمان را با [۱۲] مقایسه کنیم همانند آن، ابتدا یک بازه ۳ ساله را در نظر گرفتیم (سال‌های ۲۰۰۸ تا ۲۰۱۱ میلادی). لازم به ذکر است که در [۱۲] بازه زمانی ۳ ساله‌ای بین سال‌های ۲۰۰۵ تا ۲۰۰۸ در نظر گرفته شده است. خطای تجمعی در این آزمایشات در انتهای سال ۲۰۱۰ کمتر از ۱۲٪ بود، که این مقدار، نسبت به [۱۲] تقریباً ۲٪ کمتر است، اما برخلاف [۱۲] در این بازه سه ساله، خطای تجمعی به یک مقدار ثابت نهایی میل نکرد. در نتیجه برای پیدا کردن مقدار ثابت، برای این آزمایش بازه زمانی ۴ ساله‌ای (سال‌های ۲۰۰۷ تا ۲۰۱۱ میلادی) را مورد بررسی قرار دادیم. شکل شماره ۲ نمودار خطای تجمعی این آزمایش را برای بازه ۴ ساله را نشان می‌دهد که خطای تجمعی آن در نهایت همانند [۱۲] به مقدار تقریبی ۱۴٪ میل کرده است.



شکل ۲- نمودار خطای تجمعی، منفی غلط و مثبت غلط برای آزمایش برخط

همان‌طور که در شکل ۲ مشخص است خطای تجمعی مثبت غلط برای این آزمایش بیشتر از خطای منفی غلط می‌باشد، در حالی که در [۱۲] عکس این نکته صادق است. خطای مثبت غلط به این معناست که کلاس بندی کننده پیش‌بینی می‌کند که نفوذ رخ می‌دهد در حالی که این اتفاق نمی‌افتد. لازم به ذکر است که در مسئله پیش‌بینی نفوذها، خطای مثبت غلط از خطای منفی غلط برای کاربران کم خطرتر است، زیرا با این پیش بینی، کاربران و ناظران سیستم برای جلوگیری از نفوذ به سیستم‌هایشان آماده می‌شوند، و حداقل با قرار دادن افرادی جهت مانیتور کردن مستقیم سیستم‌ها سعی در پیش‌گیری و مقابله با نفوذ را خواهند داشت. درحالیکه در خطای منفی غلط وضعیت سیستم امن پیش‌بینی می‌شود، اما سیستم در شرایطی که ناظران و کاربران آمادگی مقابله با آن را ندارند، مورد بهره کشی توسط مهاجمان قرار می‌گیرد. در نتیجه کمتر بودن خطای منفی غلط در این تحقیق، نسبت به [۱۲] ویژگی مثبت با ارزشی محسوب می‌شود.

۵-۴- مقایسه OSVDB و CVE

در این تحقیق از گزارشات مشترک بین OSVDB و CVE استفاده شد. CVE استاندارد نام‌گذاری آسیب‌پذیری‌ها است و با دقت بسیار بر ثبت گزارش‌های آسیب پذیری نظارت می‌کند. از طرف دیگر OSVDB اطلاعات زیادی را برای هر آسیب پذیری ثبت می‌کند و دارای چندین جدول در پایگاه داده‌اش می‌باشد. در این بخش به این سوال پاسخ داده می‌شود: کدام پایگاه داده در نتایج حاصل از آزمایشات موثرتر بوده است، OSVDB یا CVE؟ برای پاسخ دهی به این سوال، آزمایشات برای OSVDB و CVE به طور جداگانه تکرار شد. طول بردار ویژگی‌های OSVDB ۱۹۶۳۰ و برای CVE ۱۲۲۰۹ بود. این بردارهای ویژگی به روش مشابه با بخش ۴ استخراج شدند. داده‌های تست استفاده شده در این آزمایشات بخشی از داده‌های مشترک بین OSVDB و CVE بودند. سایر جزئیات در روش انجام آزمایشات مشابه بخش‌های پیشین است.

جدول ۴ نتایج پیش بینی بهره کشی را برای OSVDB و CVE با هم و جدای از هم نشان می‌دهد. همانطور که در جدول مشخص است دقت OSVDB و CVE تقریباً با هم برابر است و این دقت تقریباً ۲٪ کمتر از حالتی است که بردار ویژگی‌ها همزمان حاوی اطلاعات OSVDB و CVE می‌باشد. این نتیجه نشان می‌دهد که سیاست سختگیرانه و کنترل کیفیت CVE باعث ایجاد اطلاعات بهتری نسبت به OSVDB که دارای یک سیاست باز در دریافت اطلاعات است، نمی‌شود. از طرف دیگر OSVDB حجم زیادی از اطلاعات را برای هر آسیب پذیری ثبت می‌کند، اما این حجم از اطلاعات نیز تاثیری در به دست آوردن نتایج بهتر ندارد.

جدول ۴- مقایسه OSVDB و CVE در پیش بینی بهره کشی برون خط

	فقط ویژگی‌های OSVDB		فقط ویژگی‌های CVE	
	آموزش	تست	آموزش	تست
منفی صحیح	٪۹۹.۴۳	٪۷۸.۴۶	٪۹۸.۶	٪۷۸.۱۹
مثبت صحیح	٪۹۹.۴۵	٪۸۷.۲۷	٪۹۸.۹۹	٪۸۷.۳۷
منفی غلط	٪۰.۵۷	٪۲۱.۵۴	٪۱.۴	٪۲۱.۸۱
مثبت غلط	٪۰.۵۵	٪۱۲.۷۳	٪۱.۰۱	٪۱۲.۶۳
دقت	٪۹۹.۴۴	٪۸۵.۱۱	٪۹۸.۸۸	٪۸۵.۲

۵-۵- ارزیابی ویژگی‌ها

در این بخش به اینکه چه ویژگی‌هایی نقش مهم‌تری در پیش بینی بهره‌کشی دارند، پرداخته می‌شود. به منظور انتخاب بهترین ویژگی‌ها از دو روش مختلف استفاده شد. در اولین روش از الگوریتم SVM-RFE استفاده شده است. در SVM خطی، هر ویژگی در انتهای آموزش به یک وزن مثبت یا منفی می‌رسد که اندازه این وزن نشان دهنده سهم آن ویژگی در قوانین تصمیم گیری است و وزن بیشتر بیان‌گر تاثیر بیشتر آن ویژگی است و الگوریتم SVM-RFE برمبنای این وزن‌ها کار می‌کند. با این روش ۲۲۶ ویژگی برتر انتخاب شد.

دومین روش، استفاده از روش واریانس در انتخاب ویژگی است. در روش واریانس ویژگی‌هایی که بیشترین مقدار واریانس را دارند انتخاب می‌شوند. ویژگی‌هایی با واریانس بالا عموماً تمیز دهنده‌های خوبی هستند. با این روش ۲۲۷ مشخصه انتخاب شد. ویژگی‌های انتخاب شده توسط هیچ یک از این روش‌ها نتوانست، به دقتی برابر یا حتی نزدیک به دقت پیش بینی بهره کشی با کل ویژگی‌های انتخاب شده برسد. بویژه در روش اول که براساس الگوریتم SVM-RFE بود فقط دقتی بین ۷۲ تا ۷۴ درصد حاصل شد. اما جالب‌ترین نتیجه حاصل از این آزمایشات مربوط به مشخصات انتخاب شده به روش واریانس می‌باشد. این روش با تعداد ۲۲۷ ویژگی انتخاب شده، به دقت ۸۰.۰۹٪ در پیش بینی بهره کشی رسید. اگر چه این مقدار دقت تقریباً ۷٪ کمتر از دقت پیش بینی با کل ویژگی‌های

استخراج شده است اما این نکته را نیز باید در نظر داشت که ۲۲۷ ویژگی کمتر از یک صدم کل تعداد ویژگی‌ها است.

به منظور پیدا کردن یک دید کلی نسبت به بهترین ویژگی‌ها، ویژگی‌هایی که در لیست انتخاب شده‌ی هر دو روش مشترک بودند، انتخاب شدند. تعداد ویژگی‌های مشترک به دست آمده ۲۸ ویژگی است و نتیجه جالب اینجاست که تعداد ۱۴ ویژگی از آن‌ها متعلق به CVE و ۱۴ ویژگی باقی مانده متعلق به OSVDB است. جدول ۵ گزارش مختصری از ویژگی‌های مشترک در هر دو روش و روش واریانس را نشان می‌دهد. در جدول ۵ مشخص است که "توصیف" در CVE و OSVDB، و نیز "مرجع" در CVE موثرترین گروه ویژگی‌ها هستند.

جدول ۵- گزارش مختصری از ویژگی های انتخاب شده

نام صفت در پایگاه داده	تعداد ویژگی‌های روش واریانس	تعداد ویژگی‌های مشترک بین دو روش	تعداد کل ویژگی‌های استخراج شده	پایگاه داده
Company-URL	۱	۰	۳۰۸	OSVDB - Authors
Company	۳	۱	۳۳۶	OSVDB - Authors
Comments	۰	۰	۱۴۳	CVE
Description	۴۸	۶	۷۰۴۸	CVE
Phase	۱	۰	۲	CVE
Reference	۴۲	۷	۴۹۹۲	CVE
Votes	۲	۰	۲۲	CVE
Description	۷۴	۵	۳۵۶۵	OSVDB - vulnerabilities
Email	۰	۰	۷۶۸	OSVDB - Authors
Manual-Notes	۳	۳	۲۰۰۰	OSVDB - vulnerabilities
Author-Name	۰	۰	۱۰۴۲	OSVDB - Authors
Products-Name	۱	۰	۱۴۲۵	OSVDB – object_products
Short-Description	۲۴	۴	۳۰۲۴	OSVDB - vulnerabilities
Solution	۱۷	۰	۸۹۸	OSVDB - vulnerabilities
T-Description	۰	۰	۹۵۱	OSVDB - vulnerabilities
Title	۱۴	۱	۳۸۴۸	OSVDB - vulnerabilities
Status	۲	۱	۲	CVE
Ext-References	۳۸	۰	۳۸	OSVDB – Ext_references
Obj-Affect-Name	۴	۰	۴	OSVDB – object_affect_types
Version-Name	۰	۰	۱۳۲۹	OSVDB – object_versions
Vendor-Name	۳	۰	۹۴	OSVDB – object_vendors
مجموع	**۲۷۷**	**۲۸**	**۳۱۸۳۹**	

۶- نتیجه‌گیری

وجود آسیب‌پذیری‌ها در سیستم‌های نرم‌افزاری امری بسیار متداول است. برخی از این آسیب‌پذیری‌ها در واقع راه نفوذی برای مهاجمان سیستم‌های نرم‌افزاری می‌باشند. اطلاعات مربوط به آسیب‌پذیری و نفوذ به سیستم‌ها، در پایگاه داده‌های عمومی و تخصصی مختلفی ذخیره شده است. بخش قابل توجهی از داده‌های ذخیره شده در این پایگاه داده‌ها متن گزارشات کاربران، ناظران یا سازندگان نرم‌افزارها است. در این تحقیق مشخصاتی را از میان این گزارشات استخراج کردیم و با استفاده از این مشخصات امکان نفوذ به سیستم توسط مهاجمان را پیش‌بینی کردیم. نتایج حاصل از این تحقیقات نسبت به کار مشابه پیشین بهبود قابل قبولی را داشته است.

سپاسگزاری

صمیمانه از مهران بزرگی (دانشجوی دانشگاه کالیفرنیا) و برایان مارتین (یکی از مدیران OSVDB) که در طول انجام تحقیق ما را راهنمایی کردند، سپاسگذاری می‌کنیم.

از مرکز تحقیقات مخابرات ایران که به نمایندگی از وزارت ارتباطات و فناوری اطلاعات، این پروژه را مورد حمایت قرار دادند سپاسگذاریم.

مراجع

[1] S. Frei, D. Schatzmann, B. Plattner, and B. Trammel. "Modeling the Security Ecosystem — The Dynamics of (In) Security," *Proc, Int'l Workshop on the Economics of Information Security (WEIS),* pp. 79-106, 2009.

[2] A. Arora, R. Krishnan, R. Telang, and Y. Yang. "An Empirical Analysis of Software Vendors' Patch Release Behavior: Impact of Vulnerability Disclosure," *Journal of Information Systems Research,* vol. 21, no. 1, pp. 115-132, 2010.

[3] G. Vache, "Vulnerability Analysis for a Quantitative Security Evaluation," *Proc, Int'l Symposium on Empirical Software Engineering and Measurement*, pp. 526-534, 2009.

[4] F. Massacci, and V. H. Nguyen, "Which is the Right Source for Vulnerability Studies? An Empirical Analysis on Mozilla Firefox," *ACM Trans. MetriSec, 2010.*

[5] L. GALLON, "On the impact of environmental metrics on CVSS scores," *Proc, IEEE Int'l Conf. Privacy, Security, Risk and Trust (PASSAT10)*, pp. 987-992, 2010.

[6] C. Frühwirth, and T. Mannisto, "Improving CVSS-based vulnerability prioritization and response with context information," *Proc, IEEE Int'l Workshop on Security Measurement and Metrics*, pp. 535-544, 2009.

[7] J. A. Wang, L. Zhou, M. Guo, H. Wang, and J. Camargo. "Measuring Similarity for Security Vulnerabilities," *Proc, Int'l Conf. System Sciences,* pp. 1-10, 2010.

[8] S. Huang, H. Tang, M. Zhang, and J. Tian, "Text Clustering on National Vulnerability Database," *Proc, IEEE*

Int'l Conf. Computer Engineering and Applications (ICCEA), pp. 295-299, 2010.

[9] T. Wang, T. Wei, Z. Lin, and W. Zou, "IntScope: Automatically Detecting Integer Overflow Vulnerability in X86 Binary Using Symbolic Execution," *Proc, IEEE Int'l Conf. Network Distributed Security Symposium (NDSS)*, pp. 67-82, 2009.

[10] T. Huynh, and J. Miller, "An empirical investigation into open source web applications' implementation vulnerabilities," *Journal of Empirical Software Engineering*, vol. 15, no. 5, pp. 556 - 576, 2010.

[11] M. Chandrasekaran, M. Baig, and S. Upadhyaya. "AVARE: Aggregated Vulnerability Assessment and Response against Zero-day Exploits," *Proc, IEEE Int'l Conf. Performance, Computing, and Communications (IPCCC)*, pp. 8-14, 2006.

[12] M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker. "Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits," *Proc, ACM Int'l Conf. Knowledge discovery and data mining (SIGKDD)*, pp. 25-28, 2010.

[13] Common Vulnerabilities and Exposures: The Standard for Information Security Vulnerability Names, CVE Editorial Board, http://cve.mitre.org/.

[14] OSVDB. The Open Source Vulnerability Database, http://osvdb.org/.

[15] R.-E. Fan, K.-W. Chang, C.-J Hsieh, X,-R. Wang, and C.-J Lin, LIBLINEAR –A Library for Large Linear Classification, http://www.csie.ntu.edu.tw/~cjlin/liblinear/.

[6] M. Wurst, "The Word Vector Tool: User Guide, Operator, Developer Tutorial," http://wvtool.sf.net, May 2009.

[17] United States Computer Emergency Readiness Team (US-CERT), http://wwww.kb.cert.org/vuls/html/fieldhelp, 2006.

[18] SANS Critical Vulnerability Analysis Archive, http://www.sans.org/newsletters/cva/.

[19] Microsoft Security Response Center Security Bulletin Severity Rating System, http://www.microsoft.com/technet/bulletin/rating.mspx.

[20] Forum of Incident Response and Security Teams (FIRST), Common Vulnerability Scoring System (CVSS). http://www.first.org/cvss/.

[21] Personal Communications with Mehran Bozorgi via Email (Feb 9, 2011).

[22] Personal Communications with Brian Martin via Email (Feb 13, 2011).

عاطفه خزاعی مدرک کارشناسی خود را در رشته مهندسی کامپیوتر – نرم‌افزار از دانشگاه امام رضا (ع) مشهد و مدرک کارشناسی‌ارشد خود را نیز در همین رشته از دانشگاه شیراز اخذ نموده است. او بر روی پیش‌بینی بهره‌کشی به عنوان پایان‌نامه کارشناسی‌ارشد خود تحت نظر دکتر اشکان سامی کار کرده است. امنیت و پردازش زبان طبیعی زمینه‌های تحقیقاتی مورد علاقه او می‌باشند.

آدرس پست‌الکترونیکی ایشان عبارت است از:

atefeh.khazaei@gmail.com

اشکان سامی مدرک کارشناسی خود را از Virginia Tech آمریکا، مدرک کارشناسی‌ارشد خود را از دانشگاه شیراز، و مدرک دکترای خود را دانشگاه توکیو ژاپن اخذ کرده است. او به داده کاوی، کیفیت و امنیت نرم‌افزار علاقه‌مند است. او عضو کمیته فنی چندین کنفرانس بین‌المللی مثل PAKDD، ADMA، HumanCon و Future Tech می‌باشد و بیش از ۴۰ مقاله کنفرانسی و نزدیک به ۱۰ مقاله ژورنال دارد. او عضو IEEE و یکی از بنیان گذاران CERT دانشگاه شیراز است.

آدرس پست‌الکترونیکی ایشان عبارت است از:

asami@ieee.org

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۰/۴/۱

تاریخ اصلاح: ۹۲/۳/۱۴

تاریخ قبول شدن: ۹۲/۴/۳۰

نویسنده مرتبط: عاطفه خزاعی، دانشکده مهندسی برق و کامپیوتر، دانشگاه شیراز، شیراز، ایران.