بسمه تعالی

طراحی و پیاده‌سازی نرم‌افزار تشخیص وب‌سایت‌های مخرب با استفاده از یادگیری ماشین مبتنی بر تحلیل ایستا و پویا

نگارش:

بهزاد مرادی

استاد راهنما:

دکتر عبدالله چاله چاله

شهریورماه ۱۳۹۸

فهرست مطالب

عنوان **صفحه**

چکیده

تهدیدهای امنیتی وب به طور روزافزون در حال افزایش هستند. ماهیت شبکه اینترنت به صفحات وب بدخواه این اجازه را می‌دهد تا خود را به عنوان "صفحات امن" نشان دهند و متعاقباً برخی از کاربرانی که آگاهی کافی ندارند در دام این وب‌سایت‌ها گرفتار شوند. یکی از حملات رایج این حوزه، حمله Cross-Site Scripting (XSS) است. این حمله با تزریق اسکریپت‌های مخرب به ورودی‌های صفحات وب رخ می‌دهد و هنگامی که کاربر، پیوندِ صفحه‌یِ آلوده مورد نظر را بازدید کند انجام می‌شود. روش مرسوم برای شناسایی صفحات مخرب وب، استفاده از فهرست‌های سیاه است. این فهرست‌های سیاه، توسط سازمان‌های مورد اعتماد و داوطلب تهیه می‌شود و سپس توسط مرورگرهای مدرن مانند کروم و فایرفاکس استفاده می‌شود. با توجه به اینکه، ماهیت صفحات وب به‌طور مداوم در حال تغییر است این روش، در شناسایی تهدیدهای جدید ناکارآمد است رویکرد دیگر، استفاده از روش‌های یادگیری ماشین است که تصمیم‌گیری‌های پیچیده‌تری نسبت به روش انسانی می‌توانند اتخاذ کنند. روش‌های یادگیری ماشین با تحلیل ایستای متن (بدون اجرای کد) این کار را انجام می‌دهند اما درحال حاضر، عدم شناسایی صحیح در بسیاری از برنامه‌های جاری، منجر به فعال شدن کدهای مخرب می‌شود. در این پژوهش هدف ما شناسایی پیوندهای وب‌سایت‌های مخرب با استفاده از ترکیب تحلیل ایستا و پویا (با اجرای کد) است، که به کمک این دو رویکرد ابتدا، چالش‌های رمزگشایی و مبهم‌سازی پیوند را حل کرده و سپس ویژگی‌های استخراج شده را تحلیل می‌کنیم. نتایج این تجزیه و تحلیل نشان می‌دهد که رویکرد پیشنهاد شده با الگوریتم طبقه‌بندی درخت تصادفی، پیوندهای صفحات وب را با دقت ۹۷.۱۱ درصد شناسایی می‌کند.

مقدمه

امنیت اطلاعات در محیط‌های مجازی همواره به عنوان یکی از زیرساخت‌ها و الزامات اساسی مورد تأکید قرارگرفته است. گرچه امنیت مطلق چه در محیط واقعی و چه در فضای مجازی دست‌نیافتنی است، ولی ایجاد سطحی از امنیت که به اندازه کافی و متناسب با نیازها و سرمایه‌گذاری انجام شده باشد تقریباً در تمامی شرایط محیطی امکان‌پذیر است. در سال‌های اخیر، توسعه‌دهنده‌های برنامه‌های تحت وب استفاده از تصاویر، جاوا اسکریپت و دیگر عناصر را بیشتر کرده‌اند به عنوان مثال، موتور جستجوی گوگل یک نمونه واضح از این دسته است، ابتدا از عناصر کمی تشکیل شده بود درحالی‌که اکنون از عناصر بیشتری از قبیل گرافیک، CSS و HTML ساخته شده است. زمانی که جاوا اسکریپت به صفحات وب افزوده شد توانایی ایجاد رابط‌کاربری تعاملی را به همراه خود به صفحات وب افزود. فنّاوری‌های ActiveX، Silverlight، Java Applets و غیره نیز هر کدام ویژگی‌هایی به آن افزودند به عنوان مثال، قابلیت ActiveX امکان اجرای نرم‌افزارهای متنوعی مثل فلش، PDF، DivX و ... را برای مرورگرها فراهم می‌کرد. با گذر زمان، هنگامی که

برنامه‌نویس‌های برنامه‌های وب از محیط‌های توسعه‌ای (IDE) برای ایجاد برنامه‌ها استفاده کردند، کدهای HTML قابل‌توجهی تولید شد، به مرور با افزایش تعداد مرورگرهای وب، مخصوصاً اینترنت اکسپلورر که خط فکری جداگانه‌ای داشت و نیازمند کار بیشتری برای برنامه‌نویسان وب بود این فاکتورها به همراه افزایش استفاده از اسکریپت‌ها، پیچیدگی صفحات وب را افزایش دادند تا جایی که باعث افزایش امکان آسیب‌دیدگی و تبدیل به مخرب شدن برای آن‌ها فراهم شد. درضمن، توسعه‌دهنده‌های وب ممکن است یک صفحه وب را صرفاً با استفاده از HTML بسازند، اما یک مهاجم هنوز این امکان را دارد که یک اسکریپت به آن تزریق کند و آن را مستعد حمله کند.

نوعی از حملات مبتنی بر جاوا اسکریپت وجود دارد که هویت کاربر و سیستم کاربر را هدف قرار می‌دهد، این حمله XSS[۱] نام دارد. این حمله جزء ۱۰ آسیب‌پذیری شایع وب‌سایت‌ها است که توسط سازمان OWASP[۲] ارائه شده است و می‌تواند مخاطرات بسیاری برای کاربر فراهم کند. حملات XSS به سه دسته کلی تقسیم می‌شوند که عبارتنداز: ذخیره شده، بازتابی و مبتنی بر DOM[۳]. حمله ذخیره شده بعد از تزریق اسکریپت درون صفحه‌وب، داخل پایگاه داده وب‌سایت‌ها به صورت دائم ذخیره می‌شود در حالی که در حمله بازتابی همان اسکریپت صرفاً بعد از تزریق به صورت کامل یا بخشی از آن به سمت کاربر بازتاب داده می‌شود و درون پایگاه داده به شکل دائم ذخیره نمی‌شود. در نوع سوم، حمله تزریق صرفاً درون مرورگر کلاینت انعکاس داده می‌شود که توسط DOM پردازش روی آن انجام شده است و سرور هیچ پی‌لودی از کلاینت دریافت نمی‌کند به همین دلیل تشخیص آن را دشوارتر است.

این حمله می‌تواند علاوه بر سرقت اطلاعات باعث کنترل سیستم قربانی از طریق وب شود. این حمله کدهای مخرب را از طرف منابع تایید نشده به سایت تزریق می‌کند که می‌تواند کنترل سیستم قربانی را به دست گیرد، یا اینکه کوکی‌ها، تاریخچه مرورگر و سایر اطلاعات قربانی را دریافت کرده و به سایت‌های بدخواه ارسال کند. به عنوان مثال مجرمان می‌توانند با استفاده از کوکی‌ها، هویت کاربر را جعل و محصولی را خریداری کنند بدون اینکه کاربر اطلاعی داشت باشد. تمام این کارها از طریق کوکی‌ها انجام می‌شود.

در این پژوهش، یک رویکرد براساس الگوریتم‌های یادگیری ماشین، برای طبقه‌بندی صفحات سالم از صفحاتی که حمله XSS بازتابی با استفاده از URLها انجام می‌دهند را پیشنهاد می‌کنیم. الگوریتم‌های متعددی برای رده‌بندی صفحات وب ارائه شده‌اند. اغلب روش‌های ارائه شده، از ویژگی‌های ایستا و روش‌های متن‌کاوی صفحات وب استفاده می‌کنند که ویژگی‌های رفتاری (پویای) صفحات وب را به‌درستی استخراج نمی‌کنند. در این پژوهش با استفاده از ویژگی‌های رفتاری و ایستا، تشخیص انجام می‌شود.

---

[۱] Cross-Site Scripting

[۲] Open Web Application Security Project

[۳] Document Object Model

تشخیص حملات XSS بازتابی

رایج‌ترین نوع حمله XSS، نوع بازتابی است. این حمله دو بخش اصلی دارد: بخش نخست URL تزریق شده و بخش دوم مربوط به صفحه وب است. برای تشخیص حمله به هر دو عامل نیاز است. زیرا هنگامی که اسکریپتی از طریق پیوند به وب سایت تزریق شود و بازتاب کل آن یا بخشی از آن در محتوای صفحه وجود نداشته باشد عملیات حمله با شکست مواجه شده است. رویکردهای موجود غالبا بدون توجه به محتوای صفحه صرفا با استفاده از تحلیل ایستا و روش‌های مبتنی بر عبارات باقاعده، تشخیص را انجام می‌دهند درحالی که بسیاری از این نوع حملات به طور کلی توانایی به وقوع پیوستن را ندارند و حمله‌ای رخ نداده است بلکه تلاشی صورت پذیرفته است که به حمله منجر نشده است.

مهاجمین برای دور زدن سیستم‌های تشخیص موجود، از روش‌های رمزگذاری URL، هدایت سایت و مبهم‌سازی استفاده می‌کنند. روش‌های رمزگذاری را می‌توان با سیستم تشخیص نوع رمزگذار و معکوس‌سازی عملیات رفع کرد، همچنین رویکرد هدایت سایت هم به سادگی با دنبال کردن URL هدف، می‌توان رفع کرد درحالی‌که با روش مبهم‌سازی نمی‌توان به سادگی مقابله کرد. رویکردهای معرفی شده برای رفع مبهم‌سازی به صورت دستی و خودکار معرفی شده‌اند، که در رویکرد دستی عملیات‌های روتین، توسط فرد خبره به صورت معکوس انجام می‌شود که از معایب این روش زمان بالا و امکان خطا است. در رویکرد خودکار این کار توسط ابزارهایی مانند SpiderMonkey انجام می‌شود. این ابزار یک پیاده‌سازی از زبان جاوا اسکریپت است که بدون نیاز به مرورگر کدهای مورد نظر را اجرا می‌کند. این رویکرد هم با توجه به محدودیت توابع که رفع مبهم (در کل فقط سه تابع eval، document.write و console.log) می‌کند برای بقیه توابع گزارشی به ما نمی‌دهد و به صورت جامع قابل استفاده نیست در رویکرد پیشنهادی با استفاده از تکنیک هوک کردن (منظور توقف توابع قبل از اجرا و گزارش‌گیری از آن‌ها) توابع جاوا اسکریپت، توابع را رفع مبهم کرده سپس به همراه الگوریتم تشخیص بازگشتی رمزگذاری و هدایت سایت ویژگی‌هایی را استخراج می‌کند. درنهایت با کمک مدل‌سازی مبتنی بر الگوریتم‌های درختی (ID3، C4.5 و جنگل تصادفی) و K-NN مدل طبقه‌بندی کننده‌ای ارائه شده است. ابتدا پارامترهای مدل‌های مورد نظر را بهینه کرده سپس از مدل بهینه استفاده می‌کنیم. نقطه قوت این پژوهش، استفاده از مکانیزم گزارش‌گیری رخدادهای صفحه است که مزیت‌هایی دارد که، عبارتنداز:

- **توانایی عبور از مبهم‌سازی**: با توجه به اینکه دقیقا قبل از اجرای عملیات، تابع هوک فراخوانی می‌شود، کل پارامترهای توابع و عملیات‌ها رفع مبهم‌سازی شده‌اند.
- **توانایی دنبال کردن انتقال سایت**: توسط مرورگر مجازی به صورت خودکار آدرس URL در صورت انتقال، دنبال می‌شود.

- **توانایی کاوش عمیق سایت**: توسط مرورگر پیوندهای داخلی و خارجی دنبال می‌شود. هنگامی که اسکریپتی درون آن باشد و اجرا شود توسط تابع هوک تبدیل به گزارش می‌شود. این اسکریپت ممکن است به شکل پویا یا تعاملی با کاربر، فعال و غیرفعال باشد.
- **توانایی کشف رویدادهای پویا و ایستا**: بخش ایستا توسط توابع جاوا اسکریپت دریافت می‌شود و بخش پویا توسط تابع هوک (توسط شرایط ایجاد و حذف) دریافت می‌گردد.
- **توانایی بلوکه کردن درخواست‌های شبکه**: توسط تابع هوک می‌توان برخی یا همه ارتباطات شبکه را متوقف کرد.

در ادامه براساس ویژگی‌های استخراج شده مدل‌سازی انجام می‌شود. به طور کلی ۴۰ ویژگی استخراج شده است که غالبا از تحلیل پویای کد بدست آمده‌اند و از رفتار جاوا اسکریپت تشکیل شده است. در بخش بعدی نتایج بدست آمده نشان داده می‌شود.

نتیجه‌گیری و پیشنهادات

داده‌های مورد استفاده در بخش مدل‌سازی به صورت دستی (۹۰۰ سایت نرمال) و خودکار (۴۰۰۰ سایت نرمال و ۲۰۰۰ سایت مخرب) از سایت‌های خزنده الکسا و XSSed فراهم شده است. نتایج بدست آمده در جدول زیر نشان می‌دهد که الگوریتم جنگل تصادفی بیشترین صحت، دقت تشیخص درست نمونه درست و کمترین خطا را بدست آورده است. در نهایت، براساس مدل برتر یک نرم‌افزار به صورت درخواست و پاسخ بر مبنای پروتکل HTTP به صورت سرویس POST فراهم شده است. این نرم‌افزار را هم می‌توان تحت سرور به صورت سرویس استفاده کرد و هم به عنوان یک دیوارآتش برای درخواست‌های مرورگر درون سیستم مشتری استفاده نمود.

جدول۱. نتایج ارزیابی مدل‌سازی‌های چهارگانه

الگوریتم	صحت	دقت	Recall-TPR	FPR	F1-measure	نرخ خطا
جنگل تصادفی	٪۹۷,۱۷	٪۹۷,۶۴	٪۹۸,۲۸	٪۹۴,۶۳	٪۹۷,۹۶	٪۱,۷۱
C4.5	٪۹۶,۳۴	٪۹۷,۱۱	٪۹۷,۶۳	٪۹۳,۳۸	٪۹۷,۳۷	٪۲,۳۶
ID3	٪۹۵,۱۳	٪۹۶,۸۰	٪۹۶,۲۴	٪۹۲,۴۹	٪۹۶,۵۲	٪۳,۷۵
K-NN	٪۹۳,۹۱	٪۹۷,۹۷	٪۹۳,۶۱	٪۹۴,۴۸	٪۹۵,۷۴	٪۶,۳۸

این سیستم زمان اجرای بالایی دارد، که برای افزایش بازدهی با استفاده از روش‌های چند نخی، می‌توان تعداد درخواست را افزایش داد ولی زمان پاسخ را نمی‌توان به صورت قابل ملاحظه‌ای کاهش داد. زمان معمول برای پاسخ به یک درخواست کمتر از ۵۰ ثانیه است که به دلیل استفاده از مرورگر مجازی برای استخراج ویژگی و بارگذاری صفحه وب توسط شبکه است که با توجه به سرعت شبکه و سیستمی که مرورگر در آن اجرا می‌شود کاهش یا افزایش می‌یابد.