

## مدلی جدید مبتنی بر مارکوف جهت انتخاب بهترین راه کار امن سازی برنامه های کاربردی وب

سید جواد فتحی<sup>۱\*</sup>، تقی نوشی فرد<sup>۲</sup>، مسعود باقری<sup>۳</sup>، سعید پارسا<sup>۴</sup>

۱ و ۲ - کارشناس ارشد کامپیوتر نرم افزار، دانشگاه جامع امام حسین (ع)

۳- دانشجوی دکتری نرم افزار، دانشگاه علم و صنعت ایران ۴- دانشیار، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران

(دریافت: ۹۱/۷/۵، پذیرش: ۹۱/۱۲/۲۱)

### چکیده

در سال های اخیر با گسترش روز افزون فضای وب برای تبادل اطلاعات و ارائه خدمات، تولید برنامه های کاربردی تحت وب، رشد قابل ملاحظه ای پیدا کرده است. مطابق گزارش های آماری در حال حاضر بیش از ۷۰ درصد حملات اینترنتی از طریق وب انجام می شود. بنابراین بخش عمده ای از تهدیدات ناشی از عدم تأمین امنیت کافی در برنامه های کاربردی و به طور خاص برنامه های کاربردی تحت وب است. برای مقابله با این تهدیدات راه کارهای متنوعی ارائه گردیده است. در این مقاله راه کارهای مختلف برای امن سازی برنامه های کاربردی وب مورد بررسی قرار گرفته و با توجه به معیارهای تدوین شده و مدل مارکوف پیشنهادی، روش جدیدی برای انتخاب راه کار مناسب برای امن سازی برنامه های کاربردی وب در دوره های مختلف چرخه حیات سامانه، ارائه و مورد ارزیابی قرار گرفته است. روش پیشنهادی جواب گوی یکی از مهم ترین نیازهای کشور در حوزه پدافند سایبری برای مدل کردن تهدیدات و نحوه امن سازی برنامه های کاربردی وب می باشد. در روش پیشنهادی برای بیان توانمندی هر کدام از راه کارهای مطرح شده از طیف لیکرت به عنوان ارزیاب توانمندی استفاده شده است.

**واژه های کلیدی:** برنامه های کاربردی وب، امن سازی، بازبینی کد، دیواره آتش برنامه های کاربردی وب، سامانه تشخیص و جلوگیری از نفوذ،

پویشگر امنیتی

### ۱. مقدمه

اخیراً سازمان ها و مؤسسات مختلفی در زمینه امنیت برنامه های کاربردی به صورت تخصصی، فعالیت می نمایند که بخشی از این فعالیت ها منجر به تولید پویشگرهای برنامه های کاربردی وب<sup>۳</sup>، روش های برنامه نویسی امن<sup>۴</sup> و دیواره آتش های برنامه کاربردی وب<sup>۵</sup> شده است.

یکی دیگر از فعالیت های این سازمان ها، دسته بندی و گروه بندی حملات یا آسیب پذیری های برنامه های کاربردی تحت وب است که این امر، منجر به توسعه دانش شده و کمک بسزایی در تولید محصولات امنیتی نموده است.

انجمن OWASP<sup>۶</sup> و WASC<sup>۷</sup> از جمله سازمان هایی هستند که در این زمینه، دسته بندی هایی را ارائه نموده اند.

در سال های اخیر با گسترش روز افزون فضای وب به منظور تبادل اطلاعات و ارائه خدمات، تولید برنامه های کاربردی تحت وب، رشد قابل ملاحظه ای پیدا کرده است. در این بین با وجود این که راه کارهای امنیت شبکه رشد خوبی داشته اند و محصولات متنوعی نظیر دیواره آتش، شبکه خصوصی مجازی<sup>۱</sup>، مدیریت یکپارچه تهدیدات<sup>۲</sup> و غیره ارائه می شوند، موضوعات مرتبط با امنیت فضای وب به اندازه کافی مورد توجه نبوده اند.

مطابق گزارش های آماری در حال حاضر بیش از ۷۰ درصد حملات اینترنتی از طریق وب انجام می شود [۱]. بنابراین بخش عمده ای از تهدیدات ناشی از عدم تأمین امنیت کافی در برنامه های کاربردی و به طور خاص برنامه های کاربردی تحت وب است.

4- Secure Programming Methodology

5- Web Application Firewall

6- Open Web Application Security Project (www.owasp.org)

7- Web Application Security Consortium (www.webappsec.org)

1- Virtual Private Network

2- Unified Threat Management

3- Web Application Scanner

\* رایانامه نویسنده پاسخگو: sjfathi@ihu.ac.ir

وبی دارای چالش‌های عمده هستند. راه‌حل‌های سامانه تشخیص نفوذ نوعاً برای بازرسی بعد از وقوع حمله استفاده می‌شود و راه‌حل‌های سامانه جلوگیری از نفوذ، برای شناخت و انسداد ترافیک حمله تمرکز دارند. هر دو سامانه نقش حیاتی در جلوگیری از حملات خارجی دارند ولی حفاظت کاملی در مورد تهدیدات داخلی ندارند [۵]. این سامانه‌ها برای ایجاد امنیت داخلی دارای مسائل زیر می‌باشند:

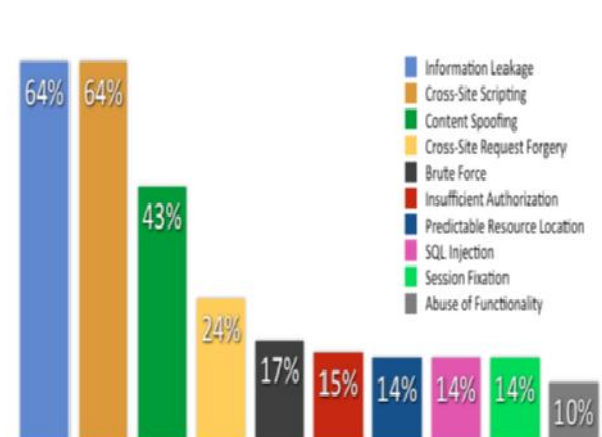
سامانه‌های مذکور قابلیت درک منطق قراردادهای ارتباطاتی برنامه‌های وب، مانند HTTP, SOAP و غیره را ندارند و نمی‌توانند بین یک درخواست طبیعی و غیرطبیعی مربوط به برنامه کاربردی وب در لایه شبکه تفاوتی قائل شوند [۶].

بسیاری از برنامه‌های کاربردی مبتنی بر وب از قرارداد SSL<sup>۴</sup> استفاده می‌کنند. این قرارداد، امکانات امنیتی تشخیص هویت، جامعیت و محرمانگی را برای برنامه‌های تحت وب فراهم می‌کند. NIPSها عموماً هیچ‌گونه بازرسی برای ترافیک‌های رمز شده نظیر SSL را انجام نمی‌دهند. با توجه به این که یک محصول NIPS الگوهای حمله را روی ترافیک رمز شده نظیر ترافیک رمز نشده اعمال می‌کند، بنابراین می‌توان نتیجه گرفت که این محصول نمی‌تواند هیچ حمله وابسته به محتوای لایه کاربرد را در قرارداد SSL تشخیص دهد. این واقعیت به دلیل رمزنگاری محتوای لایه کاربرد در قرارداد SSL است [۷].

بنابراین این سامانه‌ها در تحلیل قرارداد HTTP مشابه قراردادهای شبکه‌ای، نمی‌توانند به طور مناسب داده‌های وب را در لایه کاربرد، تجزیه و تحلیل کنند و زیرمؤلفه‌های آن را مانند سرآیندهای درخواست، کوکی‌ها، نام متغیرها و محتوای اجرایی صفحات بشناسند. سامانه‌های جلوگیری از نفوذ با داده‌های HTTP معمولاً به صورت یک متن بزرگ برخورد کرده، که باعث افزایش نرخ خطاهای نادرست مثبت<sup>۵</sup> و منفی<sup>۶</sup> می‌شود [۶]. اصولاً سامانه‌های تشخیص و جلوگیری از نفوذ سامانه‌های امنیتی بر پایه نشانه هستند، بنابراین محدودیت حفاظت در مقابل حملات ناشناخته<sup>۷</sup> را دارند. متأسفانه بیشتر نشانه‌های مورد استفاده در سامانه‌های جلوگیری از نفوذ، بر مبنای آسیب‌پذیری‌های عمومی نرم‌افزارها می‌باشند که برای برنامه‌های کاربردی وب مؤثر نیستند [۶].

بیشتر سامانه‌های جلوگیری از نفوذ قابلیت‌های نظارت ترافیک شبکه را داشته و تا حدودی می‌توانند انحراف‌هایی را تشخیص دهند

شکل ۱ نمایشگر میزان وقوع این آسیب‌پذیری‌ها در برنامه‌های کاربردی طبق آمارهای ارائه‌شده در سال ۲۰۱۱ است [۲].



شکل ۱. آمار درصد وقوع ده آسیب‌پذیری مهم برنامه‌های کاربردی [۲]

## ۲. راه‌کارهای تأمین امنیت در لایه کاربرد

### ۱.۲. سامانه‌های تشخیص و جلوگیری از نفوذ

یکی از تجهیزات امنیت شبکه که برای امن‌سازی برنامه‌های کاربردی وب در نظر گرفته می‌شود، سامانه تشخیص و جلوگیری از حملات است. این سامانه‌ها در دو نوع مبتنی بر میزبان و شبکه ارائه می‌شوند. برای امن‌سازی برنامه‌های کاربردی بر روی سرور می‌توان تنها سامانه‌های تشخیص و جلوگیری از نفوذ مبتنی بر شبکه<sup>۱</sup> (NIPS) را مورد بحث قرار داد. این سامانه‌ها غالباً امکان تشخیص حملات را با کمک دو شیوه تشخیص ناهنجاری<sup>۲</sup> و تشخیص سوء استفاده<sup>۳</sup> (مبتنی بر الگوی حملات) انجام می‌دهند [۳]. در واقع سامانه‌های NIPS با تمرکز بر تشخیص الگوهای تعریف شده ای از حملات فعالیت می‌کنند. این الگوها بیشتر حملات در سطح شبکه را پوشش می‌دهند. در حالی که در برنامه‌های کاربردی وبی تمرکز روی قرارداد HTTP بوده و باید حملات مخصوص این قرارداد را تشخیص داد. برای رسیدن به این هدف، NIPS باید کل ترافیک HTTP را موشکافی نموده و سپس برای تشخیص حملات بررسی کند. با توجه به اینکه برخی از مهاجمان از جزئیات خاص در قرارداد HTTP مطلع هستند، به راحتی از سامانه‌های NIPS گذر می‌کنند. برای نمونه در حمله SQL Injection، مهاجم با کمک مفهوم Unicode در قرارداد HTTP می‌تواند از الگوهای NIPS فرار کند [۴]. سامانه‌های تشخیص و جلوگیری از نفوذ، مکانیزم‌های امنیت محیطی را فراهم می‌کنند و در ایجاد امنیت داخلی برای شبکه‌ها و همچنین امنیت نرم‌افزارهای

4- Secure Socket Layer  
5- False Positive  
6- False Negative  
7- Zero Day Attack

1- Network Intrusion Protection System (NIPS)  
2- Anomaly Detection  
3- Misuse Detection

شناسایی می‌کنند که قبلاً با نشانه‌های<sup>۷</sup> آن آشنا شده و روش کشف آن را آموزش دیده باشند. در غیر این صورت قادر به شناسایی آسیب-پذیری نیستند، از طرف دیگر این ابزارها به فناوری ساخت برنامه‌های وب وابسته نیستند و آنها را با وجود انواع فناوری‌های توسعه مورد ارزیابی قرار می‌دهند [۱۰].

پوشگرهای برنامه‌های کاربردی وب به هر دو صورت جعبه سیاه<sup>۸</sup> و جعبه سفید<sup>۹</sup> قادر به انجام ارزیابی هستند [۱۱]. منظور از حالت جعبه سفید این است که بعضی از پوشگرهای برنامه‌های کاربردی وب در صورت دسترسی به کد برنامه و اضافه کردن کد ارائه شده توسط شرکت به برنامه کاربردی، برنامه را با دقت بالاتری مورد ارزیابی قرار میدهد [۱۲].

در سند [۸] ارزیابی توانایی‌های پوشگرهای وب برای شناسایی آسیب‌پذیری‌های برنامه‌های کاربردی وب ارائه شده است. همچنین سند NIST ۲۶۹-۵۰۰ ویژگی‌های حداقلی برای یک پوشگر برنامه‌های کاربردی وب را بیان کرده است [۹].

تحقیقات [۱۰ و ۱۱] نشان می‌دهد که پوشگرها همیشه قادر به کشف آسیب‌پذیری‌های موجود در برنامه‌های کاربردی وب نیستند.

همچنین [۱۳] بیان می‌کند که تعداد آسیب‌پذیری‌هایی که پوشگرها کشف می‌کنند کم بوده و دارای خطای بالایی است. در سال ۲۰۰۷ تحقیقاتی در مورد توانایی پوشگرهای وب برای شناسایی آسیب‌پذیری‌ها انجام شد که نتیجه آن نشان داد بین ۲۰ تا ۷۷ درصد شناسایی‌های آنها اشتباه است و همچنین ۹ درصد آسیب-پذیری‌ها نیز شناسایی نشده‌اند [۱۴].

توانایی پوشگرها در شناسایی آسیب‌پذیری‌های وب سرویس‌ها نیز توسط [۱۵] مورد آزمایش قرار گرفته است که نتیجه آن نشان‌دهنده درصد کم شناسایی آسیب‌پذیری‌ها و میزان خطای بالای پوشگرها است. تحقیقاتی نیز روی دسته خاصی از حملات انجام شده است که نشان می‌دهد، پوشگرها در شناسایی حملات Stored XSS، تزریق‌های پیشرفته SQL و حملات علیه مکانیزم‌های مدیریت جلسه دچار ضعف‌های شدیدی<sup>۱۰</sup> هستند [۱۶].

ولی به اندازه کافی قابلیت کاربرد و نظارت هر برنامه مستقل را ندارند. حملات برنامه‌های وب مانند جلوگیری از سرویس، جستجوی کامل و حملات تکه تکه شده قرارداد HTTP برای هر سایتی، آستانه‌ای (حد و مرز) دارد. هر وقت که یک برنامه از سطح آستانه‌اش تجاوز کرد، سامانه امنیتی باید درخواست‌هایش را متوقف کند. NIPS‌ها از انجام این عمل عاجزند [۶].

سامانه‌های تشخیص و جلوگیری از نفوذ، بیشتر روی داده‌های ورودی و کمتر روی داده‌های خروجی از برنامه‌های وب تمرکز دارند. مهاجمین اغلب از پیغام‌های خطای صادر شده، برای تنظیم حملات خود به منابع بانک‌های اطلاعاتی استفاده می‌کنند [۷].

## ۲.۲. پوشگرهای برنامه‌های کاربردی وب<sup>۱</sup>

از جمله ابزارهایی که در چرخه امن‌سازی برنامه‌های کاربردی وب تاثیرگذارند، پوشگرهای برنامه‌های کاربردی وب هستند که در مراحل تست قبل از تحویل برنامه یا آزمون‌های بعد از اعمال تغییرات، از آنها استفاده می‌شود. این پوشگرها ابزارهای اتوماتیکی هستند که برنامه‌های کاربردی وب را با هدف کشف آسیب‌پذیری‌های عمومی نظیر تزریق اسکریپت‌های مخرب<sup>۲</sup>، تزریق دستورات SQL<sup>۳</sup>، پیمایش دایرکتوری<sup>۴</sup>، تنظیمات غیر امن و اجرای دستورات از راه دور<sup>۵</sup> مورد آزمون قرار می‌دهند [۱].

این ابزارها توسط دو دسته از افراد استفاده می‌شوند. دسته اول، متخصصان امنیت برنامه‌های وب هستند که برای کشف آسیب‌پذیری‌ها و اصلاح آنها از این ابزارها استفاده می‌کنند و دسته دوم نفوذگرانی هستند که به منظور سوء استفاده، از این ابزارها برای کشف آسیب‌پذیری‌ها بهره می‌برند.

پوشگرها به دو دسته تجاری و متن‌باز تقسیم می‌شوند و استفاده از آنها بخش مهمی از ارزیابی امنیتی برنامه‌های کاربردی وب را تشکیل می‌دهد. همچنین به‌عنوان تکمیل‌کننده بخشی از استانداردهای امنیتی، نظیر بخش ۶.۶ از استاندارد امنیتی داده صنایع کارت‌های پرداخت الکترونیکی<sup>۶</sup>، مورد استفاده قرار می‌گیرند [۹]. پوشگرهای برنامه‌های کاربردی وب فقط آسیب‌پذیری‌هایی را

7- Signature  
8- Black Box  
9- White Box  
10- Session

1- Web Application Security Scanner  
2- XSS  
3- Standard Query Language  
4- Directory Traversal  
5- Remote Command Execution  
6- PCIDSS

روی سایت‌های پیشنهاد شده برای تست، از قبل برنامه‌ریزی کرده‌اند. برای اثبات این موضوع می‌توان با تست این ابزارها بر سایت‌های [۱۷ و ۱۸] که از آن‌ها به‌منظور آزمایش پوششگرها و همچنین آموزش کاربران در مورد امنیت برنامه‌های کاربردی وب استفاده می‌شود، به این نتیجه رسید [۱۹].

### ۳.۲. کدنویسی امن و بازبینی کد برنامه‌های کاربردی<sup>۳</sup>

بسیاری از آسیب‌پذیری‌های موجود در برنامه‌های کاربردی، ناشی از عدم دقت در تولید این برنامه‌هاست. ملاحظات امنیتی در تمام مراحل شیوه‌های تولید نرم‌افزار ارائه شده است. با این حال تولید امن برنامه‌های کاربردی هنوز هم یک چالش اساسی در امنیت نرم‌افزار است. علاوه بر ارائه رهیافت‌های امنیتی در شیوه‌های تولید، به‌طور خاص بازبینی کد و ارائه رهیافت در نحوه کدنویسی امن، راه کار مناسبی برای بهبود وضعیت نرم‌افزار تولید شده می‌باشد.

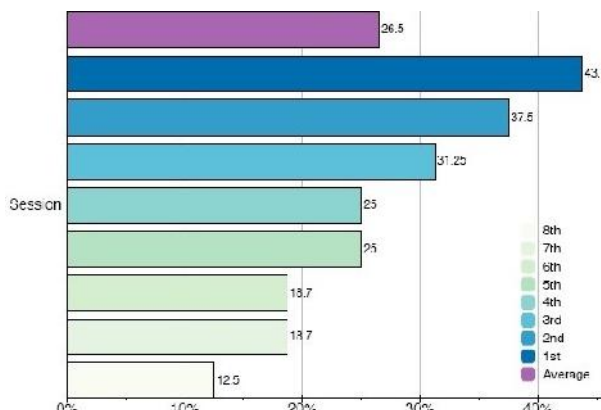
گرچه راه کارهای کدنویسی امن و بازبینی کد در راستای امنیت برنامه کاربردی بسیار مناسب می‌باشند، در مقایسه با راه کارهای دیگر هنوز نمی‌توانند جایگزین کاملی باشند [۲۰]. در زیر دلایل آن ذکر شده است.

به‌طور معمول در فرآیند تولید نرم‌افزار، راه کارهای کدنویسی امن و بازبینی آن کمتر مورد توجه مدیران تولید قرار می‌گیرند. برای نمونه با وجود تأخیر و نزدیکی به زمان‌های حساس در تولید، امنیت به سادگی قربانی خواهد شد.

گرچه ممکن است در فرآیند تولید اولین نسخه برنامه کاربردی، ملاحظات امنیتی در تولید لحاظ شود، در زمان حل مشکلات برنامه و ارائه نسخ جدید که ملاحظات زمان و هزینه مطرح می‌شود، باید منتظر نقص کدنویسی امن بود. به عنوان مثال، طبق گزارش تهدیدات اینترنت شرکت سیمان تک [۲۱] میانگین زمان لازم برای اعمال وصله<sup>۴</sup> امنیتی برای سازمان‌ها ۵۵ روز می‌باشد. در حالی که در گزارش آماری امنیت وب گروه امنیتی Whitehat زمان لازم برای اصلاح آسیب‌پذیری تزریق کد SQL کشف شده در برنامه‌های کاربردی وب ۱۳۸ روز می‌باشد [۲].

در بسیاری از موارد برنامه‌های کاربردی از طریق برون سپاری تولید شده است و در زمان کشف یک آسیب‌پذیری نمی‌توان کد برنامه را تغییر داد. بنابراین باید راه کار امن‌سازی مستقل از تغییر کد و ارائه نمود. باید توجه کرد که حجم زیادی از برنامه‌های کاربردی در

نمونه‌ای از نتایج [۱۶] را در شکل ۲ مشاهده می‌کنید که در بهترین حالت ۴۳/۷ درصد آسیب‌پذیری‌های مربوط به مدیریت نشست را شناسایی کرده است.



شکل ۲. درصد شناسایی مشکلات امنیتی مربوط به مدیریت جلسه [۱۶]

پوششگرهای برنامه‌های کاربردی وب از نظر نحوه به‌کارگیری به دو صورت مورد استفاده قرار می‌گیرند. حالت اول نشانه‌گیری و شلیک<sup>۱</sup>، حالت دوم آموزش داده‌شده<sup>۲</sup> [۱۰]. در حالت اول استفاده‌کننده نیاز به هیچ تنظیمی روی ابزار ندارد و در حقیقت پوششگر با همان تنظیمات اولیه اجرا می‌شود. با توجه به نتایج تحقیقات [۱۰]، می‌توان نتیجه گرفت که در این حالت به‌طور میانگین ۵۵ درصد آسیب‌پذیری‌ها توسط پوششگرها شناسایی نمی‌شوند. در حالت دوم استفاده‌کننده باید به‌طور میانگین ۱ تا ۱/۵ ساعت صرف تنظیمات پوششگر برای انجام آزمون برنامه کاربردی وب نماید. البته این زمان شامل زمان صرف شده برای خواندن مستندات ابزار و همچنین میزان تبحر استفاده‌کننده نسبت به ابزار نیز می‌باشد. در این حالت نزدیک به ۴۸ درصد آسیب‌پذیری‌ها توسط این ابزارها شناسایی نمی‌شوند.

باتوجه به این مطالب، برای سازمان‌های بزرگ به‌صرفه نیست که این ابزارها را با این سطح کارایی فقط به‌منظور یک بار تست یا بعد از هر تغییری در برنامه وب استفاده کنند. علاوه بر این به‌دلیل میزان بالای خطا در گزارش‌های تولید شده این ابزارها، و عدم توانایی تفکیک گزارش‌های صحیح از غلط، زمان زیادی از متخصصین امنیت برنامه‌های کاربردی وب برای تحلیل این گزارشات هدر می‌رود. تحقیقات [۱۰] نشان می‌دهد که این زمان به‌طور میانگین معادل ۲۰ ساعت است. نکته جالبی که از روش انجام تحقیق [۱۰] به‌دست می‌آید این است که تولیدکنندگان پوششگرهای تجاری ابزارهای خود را

3- Secure Coding  
4- Patch

1- Point and Shoot  
2- Trained

Overall Summary		Acunetix		Appscan		Burp Suite Pro		Haistorm		NTOSpider		Qualys		Web Inspect	
		PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained	
<b>Grand Totals</b>	<b>Vuln Found</b>	۶۴	۷۳	۸۴	۸۵	۳۲	۵۶	۶۱	۹۶	۱۴۲	۱۴۶	۴۴	۴۶	۵۲	
	<b>Found%</b>	%۴۱.۵۶	%۴۷.۴۰	%۵۴.۵۵	%۵۵.۱۹	%۲۰.۷۸	%۳۶.۳۶	%۳۹.۶۱	%۶۲.۳۴	%۹۲.۲۱	%۹۴.۱۶	%۲۸.۵۷	%۳۹.۸۷	%۳۳.۷۷	
<b>Valid Vulns</b>	<b>Vulns Missed</b>	۹۰	۸۱	۷۰	۶۹	۱۲۲	۹۸	۹۳	۵۸	۱۲	۹	۱۱۰	۱۰۸	۱۰۲	
	<b>Missed%</b>	%۵۸.۴۴	%۵۲.۶۰	%۴۵.۴۵	%۴۴.۸۱	%۷۹.۲۲	%۶۳.۶۴	%۶۰.۳۹	%۳۷.۶۶	%۷.۷۹	%۵.۸۴	%۷۱.۴۳	%۷۰.۱۳	%۶۶.۲۳	
<b>Pages</b>	<b>FP Reported</b>	۳	۳	۵	۳	۲	۶	۹	۷	۳	۳	۲	۳	۲	
	<b>Scan Time</b>	۸:۳۳	۱۰:۴۴	۶:۵۴	۶:۱۸	۰:۴۲	۱:۴۹	۲:۳۱	۹:۲۸	۸:۰۳	۷:۴۵	۱:۲۸	۲:۵۳	۴:۱۸	
	<b>Training Time</b>	N/A	۱:۱۰	N/A	۱:۳۰	N/A	۲:۰۵	N/A	۴:۱۰	N/A	۰:۰۵	N/A	N/A	۱:۲۵	
	<b>Total Time</b>	۸:۳۳	۱۱:۵۴	۶:۵۴	۷:۴۸	۰:۴۲	۳:۵۴	۲:۳۱	۱۳:۳۸	۸:۰۳	۷:۵۰	۱:۲۸	۲:۵۳	۵:۴۳	

شکل ۳. نتایج حاصل از بررسی نمونه‌های مختلف از پویسگرهای برنامه‌های کاربردی وب [۲]

آتش، نوع شبکه‌ای آن از محبوبیت بالایی در امن‌سازی برخوردار است. تکنولوژی این دیواره‌های آتش با حضور محصولات UTM، امکان بازرسی محتوای لایه کاربرد را برای انواع ویروس‌ها و Spam‌ها فراهم می‌کند. با این وجود دیواره‌های آتش شبکه‌ای درک کاملی از برنامه‌های کاربرد موجود در شبکه‌ها ندارند. برای نمونه زمانی که در یک سازمان برنامه اتوماسیون اداری استفاده می‌شود، دیواره آتش شبکه‌ای درک کاملی از این برنامه ندارد و تنها می‌تواند نوع سرویس، آدرس‌های شبکه و آدرس‌های سرویس گیرنده را بازرسی نماید. در محصولات UTM نیز تنها بازرسی ویروس و Spam به این قابلیت‌ها اضافه می‌شوند. بنابراین باید از دیواره‌های آتشی استفاده نمود که برنامه‌های کاربردی را از این تهدیدات محافظت نماید. بدیهی است برای بسیاری از این تهدیدات، ابزار امنیتی باید بتواند بیش از گزینه‌های قرارداد HTTP، برنامه کاربردی را درک و بازرسی نماید.

دیواره‌های آتش برنامه‌های کاربردی وب، گونه جدیدی از فناوری‌های نوظهور در عرصه امنیت اطلاعات و ارتباطات است که به‌منظور محافظت ویژه از برنامه‌های تحت وب در مقابل حملات مختلف و رو به افزایش در این حوزه، ایجاد شده است. سامانه‌های WAF قادر به متوقف کردن حملاتی هستند که دیواره‌های آتش شبکه و سامانه‌های تشخیص نفوذ مرسوم و معمول، در حوزه امنیت از دفاع در مقابل آن‌ها عاجزند. این سامانه‌ها به‌گونه‌ای ساخته می‌شوند که به ویرایش کدهای منبع نرم‌افزارهای تحت وبی که این سامانه از آن‌ها محافظت می‌کند نیازی نداشته باشد [۲۰]. این دیواره‌های آتش به‌طور خاص بازرسی ترافیک برنامه‌های کاربردی مبتنی بر وب را انجام می‌دهند. در این بازرسی علاوه بر اطلاعات شبکه‌ای ترافیک، اطلاعات خاص برنامه‌های کاربردی نیز مورد نیاز

سازمان‌ها با برون‌سپاری<sup>۱</sup> تولید می‌شود و هزینه تغییر کد بعد از تولید نسخ اولیه، منوط به تمدید قرارداد تولیدکننده و حتی انتخاب پیمان‌کار دیگر برای درک کد برنامه و رفع مشکل می‌باشد [۲۰].

بسیاری از سازمان‌ها چندین نوع از خدمات خود را به صورت برنامه‌های کاربردی تحت وب ارائه می‌کنند. با کشف یک آسیب‌پذیری روی برنامه‌های کاربردی، اصلاح یک به یک این برنامه‌ها هزینه بالایی خواهد داشت.

#### ۴.۲. دیواره‌های آتش برنامه کاربردی وب<sup>۲</sup>

در حیطه ابزارهای امنیت اطلاعات، محصولات دیواره آتش را می‌توان یکی از عمومی‌ترین تکنولوژی‌ها در راستای امن‌سازی نام برد. البته این محصولات توان امنیتی خود را بیشتر در زمینه امن‌سازی شبکه نشان داده‌اند. ولی واقعیت این است که دیواره‌های آتش تنها در مباحث امنیت شبکه مورد استفاده قرار نمی‌گیرند. تکنولوژی دیواره‌های آتش از زمان اولین نسل آن‌ها، یعنی حدود سال ۱۹۹۵ میلادی تاکنون، سیر توسعه و تحول ویژه‌ای را طی نموده است. این سیر توسعه در دو زمینه کلی رشد قابلیت‌های امنیتی در محصولات دیواره آتش و افزایش انواع دیواره‌های آتش با توجه به کاربردها و حیطه استفاده از آن‌ها، قابل دسته‌بندی است. محصولات دیواره آتش از نظر حیطه کاربرد شامل دیواره آتش شبکه<sup>۳</sup>، دیواره آتش شخصی<sup>۴</sup>، دیواره آتش برنامه کاربردی وب و دیواره آتش پایگاه‌داده<sup>۵</sup> می‌باشد. همان‌طور که از نام این محصولات مشخص است، هر کدام در حیطه خاصی کاربرد دارند. بدیهی است که دیواره‌های آتش شبکه‌ای، نیاز به دیواره آتش شخصی یا بقیه دیواره‌های آتش را مرتفع نمی‌سازد. البته از تمام انواع دیواره‌های

- 1- Outsource
- 2- Web Application Firewall
- 3- Network Firewall
- 4- Personal Firewall
- 5- Database Firewall

نشست، حفاظت فیلدهای مخفی در فرم‌ها و غیره می باشد. محصول WAF علاوه بر کشف آسیب پذیری، امکان ممانعت از تهدید را نیز دارد. یک محصول WAF می تواند چندین برنامه کاربردی را به طور مداوم حفاظت نماید.

با توجه به گزارش ارائه شده در [۲۲] دیواره‌های آتش برنامه کاربردی وب بدون مکانیزم آموزش درصد پایینی از حملات را شناسایی و جلوگیری می کنند و موارد شناسایی شده توسط آنها دارای میزانی از خطا است. بنابراین برای بهبود عملکرد دیواره آتش برنامه کاربردی وب در مقابل آسیب پذیری‌ها بهتر است این سامانه در ترکیب با یک پوششگر آسیب پذیری برنامه‌های کاربردی وب، به منظور تولید قوانین بازدارنده استفاده شود [۲۳].

### ۳. روش پیشنهادی

برای بیان توانمندی هر کدام از راه کارهای مطرح شده از مکانیزم طیف لیکرت به عنوان ارزیاب توانمندی استفاده شده است. جدول شماره ۱ بیانگر نداشت یک معیار عددی به معیار کمی در طیف لیکرت می باشد [۲۴ و ۲۵]:

جدول ۱. طیف لیکرت

وزن	معیار
۱	کم
۳	نسبتاً کم
۵	متوسط
۷	زیاد
۹	خیلی زیاد

سعی شده است با استفاده از این طیف، مقادیر کیفی به دست آمده در جدول ۲ را به یک مقدار کمی نگاشت کرد. در مدل مارکوف پیشنهادی از این مقادیر استفاده می گردد.

با توجه به مطالب فوق، معیارهای ارزیابی لازم در جهت استفاده از راه کارهای امنیتی بیان شده در مقاله را در دوره‌های مختلف عمر یک نرم افزار در جدول ۲ ارائه می دهیم. وزن هر کدام از معیارهای انتخابی، با بررسی‌ها و آزمایش‌های انجام شده و مقایسه نتایج آن‌ها به دست آمده است [۲۰ و ۲۲]. با توجه به اینکه توانمندی هر کدام از معیارها می تواند در یک بازه نوسان داشته باشد، از طیف لیکرت برای بیان توانمندی و مقایسه استفاده گردیده است. در مواردی که معیار انتخاب شده در راه کار مورد نظر امکان اجرا نداشته یا با شرایطی امکان چشم پوشی را دارد، از مقدار کم برای بیان اهمیت معیار مورد

می باشد. برای نمونه محصولات WAF کنترل ویژه‌ای روی مفهوم نشست<sup>۱</sup> انجام می دهند. باید توجه داشت که این مفهوم کاملاً مربوط به برنامه کاربردی است و به عبارت دیگر بالاتر از قراردادهای لایه کاربرد است.

به طور معمول یک محصول WAF یا به صورت نرم افزاری و یکپارچه در داخل کارگزار وب<sup>۲</sup> یا به صورت یک Appliance و به عنوان یک ابزار شبکه‌ای می تواند مستقر شود. معمولاً محصولات تجاری در بازار به صورت یک Appliance ارائه می شوند. این محصولات می توانند مستقل از نوع کارگزار وب، خدمات WAF را ارائه کنند. همچنین با حضور چندین نوع برنامه کاربردی می توان با استفاده از یک WAF به صورت Appliance از تمام برنامه‌های کاربردی موجود محافظت نمود. برخی ملاحظات مربوط به استفاده از WAF عبارتند از:

محصولات WAF تمرکز روی قرارداد HTTP داشته و حملات مخصوص این قرارداد را تشخیص می دهند. برای رسیدن به این هدف، WAF کل ترافیک HTTP را موشکافی نموده و سپس آن را برای تشخیص حملات بررسی می کند.

محصول WAF به طور کامل، ترافیک رمز شده SSL را بازرسی می کند. برای این منظور این محصولات به عنوان یک نماینده<sup>۳</sup> قرارداد SSL در شبکه قرار گرفته و ترافیک را رمزگشایی، بازرسی و سپس رمزنگاری می کنند.

گرچه در مفاهیم شبکه‌ای بالاترین لایه در پشت قراردادها شبکه‌ای<sup>۴</sup>، لایه کاربرد<sup>۵</sup> است، در زمانی که مفاهیم برنامه نویسی شبکه مطرح می شود، و به طور خاص در برنامه نویسی وب، لایه‌های بالاتر از لایه کاربرد نیز وجود دارد. البته این لایه وابسته به نوع برنامه کاربردی وب متفاوت است. برای نمونه مفاهیمی نظیر مدیریت نشست، Web Service، اعتبارسنجی ورودی<sup>۶</sup> و ... همگی بالاتر از سطح قرارداد HTTP قرار می گیرند. بخش زیادی از قابلیت‌های صافی کردن در WAF مربوط به بازرسی در لایه‌های بالاتر از لایه کاربرد است. برای نمونه WAF می تواند مدیریت نشست را در یک برنامه کاربردی بررسی نماید. برخی از قابلیت‌های امنیتی محصول WAF که خاص مفاهیم لایه بالاتر کاربرد می باشد، شامل کاهش حملات جستجوی کامل<sup>۷</sup>، حفاظت از کوکی‌ها<sup>۸</sup>، کاهش حملات مبتنی بر

1- Session  
2- Embeded  
3- Proxy  
4- Network Stack Protocol  
5- Input Validation  
6- Brute Force  
7- Brute Force  
8- Cookies

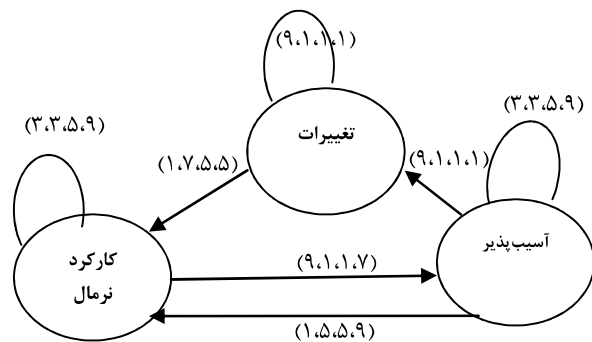
جدول ۲. معیارهای مربوط به ارزیابی راه کارهای امن سازی برنامه‌های کاربردی وب

ردیف	معیارهای ارزیابی		بازبینی کد	پوششگر	NIDS/IPS	WAF
۱	زمان مورد استفاده	توانمندی کشف آسیب پذیری در زمان تغییر کد	خیلی زیاد	متوسط	کم	زیاد
		واکنش سریع به حمله	نسبتاً کم	نسبتاً کم	متوسط	خیلی زیاد
۲	مهارت‌های مورد نیاز	آشنایی با زبان‌های برنامه‌نویسی	خیلی زیاد	نسبتاً کم	کم	متوسط
		آشنایی با حملات	خیلی زیاد	متوسط	زیاد	خیلی زیاد
		آشنایی با پروتکل‌های مربوط	زیاد	متوسط	زیاد	خیلی زیاد
۳	زمان و هزینه	نصب و راه‌اندازی	کم	متوسط	زیاد	زیاد
		آموزش	خیلی زیاد	کم	متوسط	متوسط
		تنظیمات دوره‌ای	کم	نسبتاً کم	متوسط	متوسط
۴	مزایا	سرعت تست نرم‌افزار	متوسط	زیاد	کم	کم
		تشخیص حملات فعال	کم	کم	نسبتاً کم	خیلی زیاد
		جلوگیری از حملات فعال	کم	کم	زیاد	خیلی زیاد
		Zero Day Detection	خیلی زیاد	کم	کم	زیاد
		بررسی کانال رمز شده	کم	کم	کم	خیلی زیاد
		اعمال سیاست‌های مستقیم امنیتی	کم	کم	کم	زیاد
		محافظت هم‌زمان از چندین برنامه کاربردی وب	متوسط	متوسط	زیاد	خیلی زیاد
		در دسترس نبودن کد (جعبه سیاه)	کم	متوسط	متوسط	خیلی زیاد
		تشخیص حملات منطقی	خیلی زیاد	کم	کم	کم
		قابلیت همبستگی با سایر ابزارهای امنیتی	کم	متوسط	زیاد	زیاد
		قابلیت یادگیری / تجربه‌نگاری	زیاد	کم	کم	متوسط
		False Positive	کم	متوسط	زیاد	نسبتاً کم
		False Negative	کم	متوسط	زیاد	نسبتاً کم
		انسان خبره در دسترس	نسبتاً کم	زیاد	متوسط	متوسط
۵	معایب	سرپار در بلندمدت	کم	نسبتاً کم	متوسط	زیاد
		عدم توانایی در پوشش نقصان‌های معماری نرم‌افزار	کم	خیلی زیاد	خیلی زیاد	خیلی زیاد

برای انتخاب راه کار مناسب در شرایط مختلف از مدل مارکوف استفاده می‌کنیم. برای به دست آوردن مدل مارکوف، فرض شده که برنامه کاربردی وب در حال استفاده است. بدین منظور که مدل ارائه شده، برای انتخاب راه کار مناسب در زمان استفاده

نظر استفاده می‌شود. به عنوان مثال زمان و هزینه مورد نیاز نصب و راه‌اندازی در روش بازبینی کد، قابل چشم‌پوشی بوده ولی با توجه به انتخاب طیف لیکرت به عنوان معیار ارزیابی می‌توان مقدار کم را به آن نسبت داد.

امن سازی در طیف لیکرت می باشد. مقادیر ارائه شده برای هر کدام از چهار تایی ها از نگاشت وزن های ارائه شده در جدول ۲، با معیارهای مطرح طیف لیکرت در جدول ۱ به دست آمده است.



شکل ۵. مدل مارکوف؛ انتخاب راه کار مناسب امن سازی برنامه کاربردی وب

باتوجه به مدل ارائه شده می توان از روی چهار تایی مربوط به زمان استفاده یعنی (۳,۳,۵,۹) نتیجه گرفت که ترکیب استفاده از پویشگر با دیواره آتش برنامه کاربردی وب یا ترکیب پویشگر با سامانه تشخیص و جلوگیری نفوذ به ترتیب می توانند راه کارهای مناسبی برای امن سازی برنامه کاربردی وب باشند.

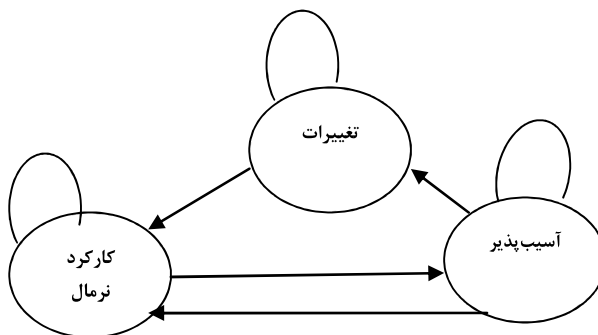
همچنین می توان این نتیجه را گرفت که در صورت وقوع آسیب پذیری با شرایط در دسترس نبودن کد برنامه، راه کار مناسب استفاده از دیواره آتش برنامه کاربردی وب می باشد و در شرایطی که کد برنامه در دسترس باشد بازبینی کد به عنوان راه کار مناسب می تواند انتخاب شود. همان طور که در مدل مشاهده می شود، در شرایط مختلف و با توجه به معیارهای معایب، مزایا و هزینه زمان می توان گزینه های دیگر را با اولویت خاص انتخاب کرد.

آشنایی با زبان های برنامه نویسی تقریباً برای تمامی روش های مطرح لازم می باشد ولی در صورت استفاده از روش هایی که نیاز بالایی به این پارامتر دارند، باید محدودیت ها و معایب مربوط به آن روش مانند کمبود نیروی انسانی متخصص را مد نظر داشت.

زمان و هزینه مورد نیاز برای هر کدام از راه کارهای مطرح، از دو نگاه مورد بررسی قرار گرفته است. اولی مربوط به هزینه و زمان مورد نیاز برای نصب و راه اندازی و آموزش در جهت استفاده بهینه از راه کار و دومی مربوط به هزینه و زمان مورد نیاز برای به روز رسانی هر کدام از راه کارهای مطرح که در قالب تنظیمات دوره ای آورده شده است.

در بخش مزایا توانایی راه کارهای مختلف در شناسایی حملات ناشناخته، محافظت از چندین برنامه در آن واحد، میزان همبستگی با

برنامه کاربردی است و شرایط تولید برنامه کاربردی مد نظر نمی باشد. برای یک برنامه کاربردی در زمان استفاده، حالات کارکرد نرمال، آسیب پذیر و تغییرات در برنامه، مطابق شکل متصور می باشد.



شکل ۴. مربوط به حالات مختلف یک برنامه کاربردی در زمان استفاده

همان طور که در شکل ۴ نمایش داده شده است، وقتی یک برنامه کاربردی که در حالت کارکرد نرمال است با گزارش یک آسیب پذیری به حالت آسیب پذیر می رود، در حالت آسیب پذیر در صورتی که کد برنامه کاربردی برای رفع آسیب پذیری در دسترس باشد برنامه به حالت تغییرات منتقل شده و پس از رفع آسیب پذیری مجدداً به حالت کارکرد نرمال باز می گردد.

در صورت در دسترس نبودن کد برنامه، باید از راه کارهای امن سازی که نیاز به کد برنامه ندارند استفاده کرد و با رفع آسیب پذیری، برنامه را دوباره به حالت کارکرد نرمال بازگرداند.

حال با استفاده از پارامترهای مطرح شده در جدول ۳ و با استفاده از شکل ۴، مدل مارکوف را مربوط به انتخاب راه کار مناسب در شرایط مختلف چرخه استفاده یک برنامه کاربردی ارائه می دهیم (شکل ۵).

جدول ۳. نحوه محاسبه امتیاز مربوط به هر کدام از روش ها

SOLUTION1	WAF+SC=9+3=12
SOLUTION2	IDS+SC=5+3=8

در مدل ارائه شده چهار تایی، (CA, SC, IDS, WAF) به ترتیب بیانگر امتیاز کسب شده استفاده از بازبینی کد، پویشگر، سامانه تشخیص و جلوگیری از نفوذ و دیواره آتش برنامه کاربردی برای

1. Brute Force
2. Cookies



Security به تنهایی از بین ۱۱۹ سایت آسیب پذیر توانست ۴۴ سایت را به درستی آسیب پذیر تشخیص دهد. در صورتی که در ترکیب با پویشرگ آسیب پذیری W3af این تعداد به ۹۰ سایت افزایش یافت.

پس می توان نتیجه گرفت نتایج به دست آمده از ارزیابی منطبق با مدل مارکوف پیشنهادی بوده و جمع مقادیر مربوط به چهار تایی ها در شرایط مختلف میزان کارا بودن آن راه کار را نشان می دهد.

## ۵. نتیجه گیری

در این مقاله ابتدا راه کارهای مختلف امن سازی برنامه های کاربردی وب مطرح و در شرایط مختلف، چرخه حیات سامانه با هم مقایسه گردیدند. سپس با توجه به معیارهای تدوین شده و مدل مارکوف پیشنهادی نحوه انتخاب راه کار مناسب برای امن سازی برنامه های کاربردی ارائه گردید و مورد ارزیابی قرار گرفت. نتایج حاصله نشان داد که در زمان استفاده از برنامه کاربردی وب برای انجام واکنش سریع به حمله، مهارت های آشنایی با زبان های برنامه نویسی، آشنایی با حملات و آشنایی با پروتکل های مربوطه، راه کار مناسبی برای تأمین امنیت برنامه کاربردی، استفاده از دیواره آتش برنامه کاربردی وب با ترکیب پویشرگ های آسیب پذیری می باشد. در صورت استفاده از این گزینه حملات Zero Day و حملات فعال با درصد احتمال بالایی قابل شناسایی می باشند. این روش به عنوان یکی از راه کارهای مناسب برای امن سازی برنامه های کاربردی وب که کد آنها در دسترس نمی باشد مطرح است. معایب مهم این روش عدم توانایی پوشش نقصان های مربوط به معماری برنامه های کاربردی وب است. در صورت کشف آسیب پذیری در دوره استفاده از نرم افزار نیز با توجه به پارامترهای مزایا، معایب، هزینه و زمان بیان شده در جدول ارزیابی، دیواره آتش برنامه کاربردی وب گزینه مناسب تری می باشد.

روش پیشنهادی جواب گوی یکی از مهم ترین نیازهای کشور در حوزه پدافند سایبری برای مدل کردن تهدیدات و نحوه امن سازی برنامه های کاربردی وب می باشد.

ابزارهای امنیتی دیگر (به عنوان مثال دیواره های آتش برنامه کاربردی وب برای محافظت از بارگذاری فایل های آلوده در سرور از آنتی ویروس ها کمک می گیرند)، توانایی یادگیری و میزان وابستگی به در دسترس بودن کد برنامه مد نظر قرار گرفته است. بخش معایب از دو منظر کلی عملکرد (False Negative, False Positive) سربار و پوشش نقصان های معماری نرم افزار) و هزینه (انسان خبره) انتخاب شده اند.

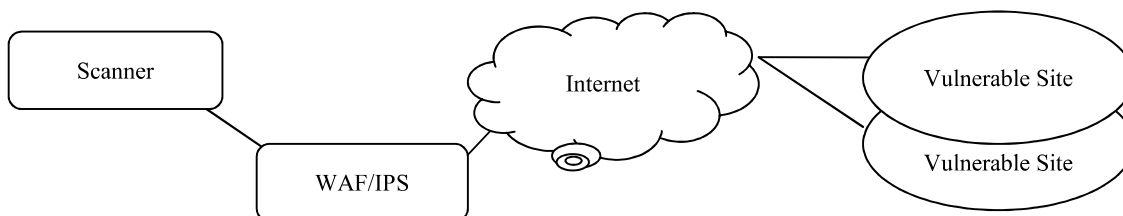
## ۴. ارزیابی روش پیشنهادی

برای ارزیابی روش پیشنهادی از ابزارهای متن باز Mode Security، W3af و Snort استفاده گردیده است. برای این منظور ابزارهای مطرح در معماری شکل ۶ استقرار یافتند.

پس از استقرار معماری، حملات مطرح در حوزه برنامه های کاربردی وب بر روی ۳۱۵ وب سایت مورد آزمون قرار گرفت. نتایج حاصل از این ارزیابی در جدول ۴ ارائه گردیده است. همان طور که مشاهده می شود محصولات دیواره آتش و سامانه های تشخیص نفوذ، به تنهایی در مقابل حملات تا حدی موفق بوده اند. در صورتی که ترکیب این محصولات با پویشرگ های آسیب پذیری درصد موفقیت چشمگیری را نشان می دهد. به عنوان مثال دیواره آتش Mode

جدول ۴. نتایج حاصل از ارزیابی روش پیشنهادی

		WAF		IDS	
		Mode Security	W3af	Snort	W3af
تعداد کل سایت های مورد ارزیابی ۳۱۵	تشخیص صحیح	۴۴	۹۰	۳۱	۸۶
	درصد تشخیص صحیح	%۳۶.۹۷	%۷۵.۶۳	%۲۶.۰۵	%۷۲.۲۶
تعداد سایت های آسیب پذیر ۱۱۹	تشخیص نادرست	۷۵	۲۹	۸۸	۳۳
	درصد تشخیص نادرست	%۶۳.۰۲	%۲۴.۳۶	%۷۳.۹۵	%۲۷.۷۳



شکل ۶. معماری استقرار ابزارها برای ارزیابی روش پیشنهادی

۵. مراجع
- [17] D. Rhoades. WebMaven. Available at: <http://www.mavensecurity.com/WebMaven.php>, 2002
- [18] McAfee, Inc. Hacme Bank v2.0. Available at: <http://www.foundstone.com/us/resources/proddesc/hacmebank.htm>, May 2006.
- [19] Using a Web Server Test Bed to Analyze the Limitations of Web Application Vulnerability Scanners, David A. Shelly, Blacksburg, Virginia, 2010
- [20] Barnett. R, "WAF Virtual Patching Challenge: Securing WebGoat with ModSecurity." s.l: Blackhat DC Conference, 2009.
- [21] Internet Threat Report, H3: Symantec, 2007.
- [22] Suto. L, "Analyzing the Effectiveness of Web Application Firewalls." 2011.
- [23] Lemos. R, Time To Automate Web Defenses?, [Http://www.darkreading.com/vulnerability-management/167901026/security/attacks-breaches/231901651/time-to-automate-web-defenses.html](http://www.darkreading.com/vulnerability-management/167901026/security/attacks-breaches/231901651/time-to-automate-web-defenses.html). Oct 25, 2011.
- [24] Likert. R, "A Technique for the Measurement of Attitudes," Archives of Psychology, p. 140:1-55, 1932.
- [25] Wuensch. K. L, "What is a Likert Scale? and How Do You Pronounce 'Likert?," East Carolina University, 2009.
- [26] Bryant. Ch, The Key Difference Between IPS And IDS. Streetdirectory. [Online] 2011. [http://www.Streetdirectory.com/travel\\_guide/149387/networking/the\\_key\\_difference\\_between\\_ips\\_and\\_ids.html](http://www.Streetdirectory.com/travel_guide/149387/networking/the_key_difference_between_ips_and_ids.html).
- [27] Palka. D, Zachara. M, "Learning Web Application Firewall - Benefits and Caveats." 2011.[28] A. Robert, R. C. Barnett and etc., Web Application Firewall Evaluation Criteria. s.l.: Web Application Security Consortium, 2009.
- [28] Barnett. R, "XSS Street-Fight:The Only Rule Is There Are No Rules." s.l. : Blackhat Dc Conference, 2010.
- [1] Web Application Security Consortium., WEB APPLICATION SECURITY STATISTICS, <http://www.Webappssec.org/>. s.l. : (WASC), 2011.
- [2] WhiteHat Security., 11th Edition Website Security Statistic Report. s.l. : WhiteHat Security, Inc., 2011.
- [3] Scarfone. K, Mell. P, Guide to Intrusion Detection and Prevention Systems (IDPS). s.l: Computer Security Resource Center, 800-94, 2007.
- [4] Kruegel. Ch, Valeur. F, Vigna. G, INTRUSION DETECTION AND CORRELATION Challenges and Solutions. s.l: Springer Science + Business Media, Inc, 2005.
- [5] What is the Difference Between IPS, IDS and Internal Security? checkpoint. [Online] Check Point Software Technologies, [http://www.checkpoint.com/securitycafe/readingroom/internal\\_security/ips\\_ids\\_internal\\_security.html](http://www.checkpoint.com/securitycafe/readingroom/internal_security/ips_ids_internal_security.html), 2011.
- [6] McMillan. J, Intrusion Detection FAQ: What is the difference between an IPS and a Web Application Firewall? sans.org. [Online] <http://www.sans.org/security-resources/faq/ips-web-app-firewall.php>, November 2009.
- [7] OWASP German Chapter., Best Practices:Use of Web Application Firewalls. s.l. : Open Web Application Security Project(OWASP), 2011.
- [8] Web application Security Consortium,Web Application Security Scanner Evaluation Criteria,Version 1.0, 2009
- [9] <http://samate.nist.gov>
- [10] Suto. L, Analyzing the accuracy and time costs of web application security scanners, February 2011.
- [11] Fong. E, Gaucher. R, Okun. V, Black. P. E, and Dalci. E, Building a test suite for web application scanners. Hawaii International Conference on System Sciences, 2008
- [12] Antunes. N, Vieira. M, Detecting SQL Injection Vulnerabilities in Web Services. In Dependable Computing, 2009. LADC '09. Fourth Latin-American Symposium on, pages 17-24, Sept. 2009.
- [13] Fonseca. J, Vieira. M, Madeira. H, Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks. In Dependable Computing, 2007.PRDC 2007. 13th Pacific Rim International Symposium on, pages 365-372, Dec. 2007
- [14] Vieira. M, Antunes. N, Madeira. H, Using web security scanners to detect vulnerabilities in web services. In Dependable Systems & Networks, 2009. DSN '09. IEEE/IFIP International Conference on, pages 566-571, 29 2009-July 2 2009.
- [15] <http://www.owasp.org>
- [16] State of the Art: Automated Black-Box Web Application Vulnerability Testing, Jason Bau, Elie Bursztein, Divij Gupta, John Mitchell, Stanford University Stanford, CA, 2009