مجله علمی ـ پژوهشی «پدافند الکترونیکی و سایبری»
سال اول، شماره ۳، پاییز ۱۳۹۲، ص ۲۴- ۱۳

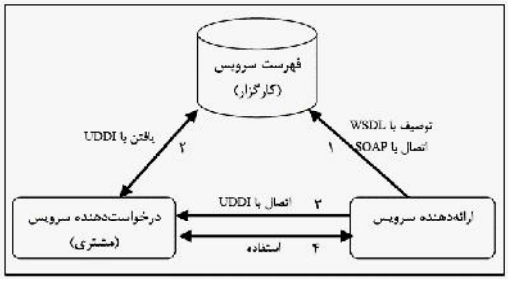
ارتقای امنیت سرویس‌های وب با استفاده از فنون تحمل‌پذیری خطا با تأکید بر فن تنوع طراحی

صادق بجانی[1*]، محمد عبداللهی ازگمی[2]
۱- دانشجوی دکتری مهندسی کامپیوتر، دانشگاه جامع امام حسین (ع)،
۲- استادیار دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران
(دریافت۹۱/۱۰/۰۳، پذیرش: ۹۲/۰۲/۲۸)

چکیده
بی‌توجهی به امنیت سیستم‌های نرم‌افزاری مبتنی بر سرویس‌های وب، به‌علت چالش‌های امنیتی آنها، نتایج زیان‌بار و جبران‌ناپذیری به‌همراه دارد. تاکنون استانداردهای امنیتی بر اساس مکانیزم‌های سنتی امنیتی، نظیر رمزنگاری و امضاء دیجیتال، در رابطه با امنیت سرویس‌های وب و سیستم‌های مبتنی بر آنها مطرح شده است. همچنین فنون کلاسیک تحمل‌پذیری خطا و روش‌های متعارف امنیتی برای ارتقای امنیت سرویس‌های وب قابل بهره‌برداری است. علی‌رغم استفاده از استانداردهای امنیتی و فنون یادشده، تاکنون امنیت کامل سرویس‌های وب فراهم نشده و شاهد استمرار نفوذ در سرویس‌های وب هستیم. در مقاله حاضر برای ارتقای امنیت سرویس‌های وب، رویکردی مبتنی بر روش‌های متعارف امنیتی و تکنیک‌های کلاسیک تحمل‌پذیر خطا، با تأکید بر تکنیک‌های افزونگی و تنوع طراحی برای طراحی سرویس‌های وب پیشنهادشده است. عملکرد سیستم نرم‌افزاری مبتنی بر سرویس وب که با استفاده از تکنیک‌های تحمل‌پذیری خطا و سیستم نرم‌افزاری مبتنی بر وب و بدون استفاده از تکنیک‌های تحمل‌پذیری خطا طراحی شده، با بهره‌گیری از زنجیره‌های مارکوف مدل‌سازی شده و کارایی آنها با استفاده از نرم‌افزار Maple محاسبه شده است. نتایج ارزیابی نشان می‌دهد که میزان سرویس‌دهی سیستم‌های مبتنی بر وب که از تکنیک‌های تحمل‌پذیری نفوذ بهره می‌برند، به‌طور قابل توجهی افزایش می‌یابد.

واژه‌های کلیدی: نفوذ، تکنیک‌های تحمل‌پذیری خطا، تحمل‌پذیری نفوذ، تکنیک‌های افزونگی، تنوع طراحی.

۱. مقدمه
سرویس‌های وب راه حل اصلی تحقق معماری سرویس‌گرا محسوب شده و برسکوهای متفاوت توزیع‌شده در سطح اینترنت اجرا می‌شوند. رابط سرویس‌های وب براساس XML[1] تعریف می‌شود [۱]. سرویس‌های وب از طریق ارسال پیام‌های XML، با سیستم‌های نرم‌افزاری تبادل اطلاعات نموده و یک تعریف قابل پردازش ماشین، به‌نام WSDL[2] دارند [۲]. آنها به‌طور عمومی از سیستم‌های ناهمگن تشکیل شده‌اند و برای توسعه آنها، از معماری سرویس‌گرا استفاده می‌شود [۲]. شکل **۱**، تعامل اجزای سرویس‌های وب را نشان می‌دهد.



شکل ۱: تعامل اجزای سرویس وب [۳]

در معماری مبتنی بر سرویس‌های وب بر پایه SOAP[3] که فراهم کننده قالب تعاملات آنهاست، پروتکل UDDI[4]، امکانات انتشار، جستجو و استفاده از این سرویس‌ها را فراهم می‌کند و از XML به‌عنوان شالوده توصیف، انتشار و ارتباط خود استفاده می‌نماید [۴]. با وجود اهمیت این سرویس‌ها در اثربخشی سیستم‌های توزیع شده در زمینه‌های کاربردی مختلف، چالش‌ها و مسائل امنیتی خاصی نیز دارند که به برخی از آنها در ذیل اشاره می‌شود.

۱-۱. مسائل امنیتی و حملات ویژه سرویس‌های وب
پروتکل HTTP از پروتکل‌های پرکاربرد انتقال در وب است و آسیب‌پذیری‌های آن، تهدیدات سرویس‌های وب تلقی می‌شود [۲]. ده مورد از بیشترین آسیب‌پذیری کاربردهای وب که از طرف مؤسسه OWASP در سال ۲۰۱۰ منتشر شده، در مورد سرویس‌های وب نیز صادق است [۵]. براساس مرجع [۶]، حملات رایجی که هدف اصلی آنها از کار انداختن یا سوءاستفاده از سرویس‌های وب است، عبارتند از:

سوءاستفاده از فایل توصیف سرویس وب: مشخصه در دسترس بودن توصیف سرویس وب در WSDL در محیط باز اینترنت، می‌تواند مبنایی برای سوءاستفاده و تدارک حمله باشد.

* ایمیل نویسنده پاسخگو: sbejani@ihu.ac.ir
[1] eXtensible Markup Language
[2] Web Service Description Language
[3] Simple Object Access Protocol
[4] Universal Description, Discovery and Integration

حمله جلوگیری از سرویس[1]: اگر نرخ درخواست‌های ارسالی به سرویس‌دهنده وب از حد آستانه بیشتر شود و امکان پاسخ‌گویی به درخواست‌های بعدی وجود نداشته باشد، یعنی سیستم وارد حالت حمله جلوگیری از سرویس شده است.

حملات مبتنی بر XML: این حملات به‌منظور جلوگیری از سرویس و ایجاد دسترسی‌های غیرمجاز است. برای استفاده دنباله‌های دریافتی XML، تجزیه لازم است. برای تجزیه از دو روش SAX[2] , DOM[3] استفاده می‌شود. تجزیه‌گر DOM، کل دنباله را در حافظه بارگذاری می‌کند و مهاجم می‌تواند با ارسال فایل‌های بزرگ XML، بخش بزرگی از حافظه را اشغال نموده و از منابع سرویس‌دهنده استفاده کند و شرایط جلوگیری از سرویس را فراهم نماید [۲]. تجزیه‌گرهای SAX، دنباله‌های XML را تنها در زمان نیاز پردازش می‌کنند و در برابر حمله تزریق XML آسیب‌پذیرند. در این حمله مهاجم به‌دنبال ایجاد خرابی در فایل XML اصلی و بازنویسی روی مقادیر قبلی است.

حمله منع سرویس با استفاده از آدرس‌های IP جعلی: از ساز وکارهای رمزنگاری در سرویس‌هایی که به‌نوعی از مکانیزم‌های اعتبارسنجی، در بخش اختیاری سرآیند پیام SOAP استفاده می‌کنند، سوءاستفاده می‌شود.

حملات انکار سرویس: این حملات شامل حملات سیل‌آسا، ارسال دنباله‌های بازگشتی به تجزیه‌گرهای XML، ارسال دنباله‌های حجیم به تجزیه‌گرهای XML و مسمومیت الگو است.

۱-۲. اهداف طرح پیشنهادی

هدف اصلی مقاله حاضر، ارائه روش طراحی سرویس‌های وب تحمل‌پذیر نفوذ، برای کاربردهای عمومی است. بدین منظور، از تلفیق روش‌های متعارف امنیتی و تکنیک‌های کلاسیک تحمل‌پذیری خطا استفاده شده است. استفاده از سیستم‌های پیشگیری از نفوذ[4] و تشخیص نفوذ[5] که جزئی از روش‌های متعارف امنیتی هستند، بعضی از نفوذهای پیشگیری نشده را تشخیص داده شده و از بروز برخی نیز پیشگیری می‌شود. اما این روش‌ها به‌تنهایی نمی‌توانند همه مشکلات امنیتی سرویس‌های وب را حل کنند، زیرا همچنان امکان نفوذ وجود دارد. از طرف دیگر روش‌های متعارف، برای مقابله با نفوذهای شناخته شده، مناسب است و در مقابل نفوذهای جدید ناکارآمد است. در طرح پیشنهادی، استفاده از فنون‌های افزونگی، تنوع طراحی و رأی‌گیری که از فنون‌های کلاسیک تحمل‌پذیری خطا هستند، پیشنهاد شده است. استفاده از فنون افزونگی در طرح پیشنهادی برای پاسخگویی به درخواست کاربر ایجاب می‌کند به‌جای یک سرویس وب، از سه سرویس وب استفاده شود.

برپایه ایده اصلی مقاله، لازم است برای توسعه سرویس‌های وب افزونه، از فنون تنوع طراحی شود. این امر موجب می‌شود سرویس‌های وب افزونه کار یکسانی داشته، اما نحوه تولید، ابزار و محیط توسعه آنها متفاوت باشد. استفاده از فنون تنوع طراحی در توسعه سرویس‌های وب، زمینه کاهش احتمالات خرابی‌های حالت مشترک سرویس‌های وب افزونه را فراهم می‌کند و می‌توان امیدوار بود که آسیب‌پذیری‌های هر یک از سرویس‌های وب افزونه با آسیب‌پذیری سرویس‌های وب دیگر متفاوت است. در این شرایط، اگر مهاجم موفق به شناسایی آسیب‌پذیری در یکی از سرویس‌های وب افزونه شود و حمله‌ای را به آن تدارک دیده و عملی کند، از این‌رو به دلیل کاهش احتمال، وجود همان آسیب‌پذیری (مبدأ حمله) در سایر سرویس‌های وب افزونه، احتمال تکرار حمله روی سرویس‌های وب افزونه بسیار کمتر خواهد بود.

همچنین به‌جای یک سرویس وب از سه سرویس وب استفاده می‌شود، برای تعیین نتیجه نهایی رأی‌گیری می‌شود که نتیجه آن، قابل پوشش بودن حمله به یکی از سرویس‌های وب است. بهره‌برداری توأم افزونگی و تنوع طراحی برای تعیین نتیجه نهایی، استفاده از فرآیندهای رأی‌گیری را لازم نموده و موجب افزایش هزینه توسعه سرویس‌های وب وافزایش هزینه پردازش درخواست کاربران می‌شود. افزایش این هزینه‌ها، یک مبادله بین هزینه و افزایش تحمل‌پذیری نفوذسرویس‌های وب است و می‌تواند براساس فاکتورهای مختلف در معماری پیشنهادی مورد پذیرش قرار گیرد. بنابراین استفاده توأم تکنیک‌های افزونگی و تنوع طراحی، تحمل‌پذیری نفوذ سرویس‌های وب و احتمال رسیدن به نتیجه درست را افزایش می‌دهد.

در بخش ۲ مقاله، مروری بر تحقیقات مرتبط خواهد شد. در بخش ۳، کلیات طرح پیشنهادی بیان و در بخش ۴ طرح تشریح می‌شود. در بخش ۵، مدل‌سازی و ارزیابی ارائه و در بخش ۶، نتیجه‌گیری و پیشنهادها بیان می‌شود.

۲. پیشینه تحقیق

سرویس‌های وب از نظر آسیب‌پذیری‌ها و نوع حملاتی که آنها را تهدید می‌نمایند، با سیستم‌های متعارف نرم‌افزاری متفاوتند و چون در محیط باز اینترنت اجرا می‌شوند، درمعرض خطرات بیشتری قرار دارند. به این علت، روش‌ها، فنون‌ها ومعماری‌های ارائه شده برای سیستم‌های متعارف نرم‌افزاری، به‌تنهایی برای سرویس‌های وب کارساز نیستند. در تحقیقات انجام شده، از فنون برنامه‌نویسی تک نگارشی، روش‌های متعارف امنیتی و تنوع ضمنی مورد استفاده قرار می‌گیرد. با وجود اینکه استاندارد امنیتی براساس ساز وکارهای سنتی (نظیر رمزنگاری) برای سرویس‌های وب[6] ارائه شده، از این‌رو به

[1] Denial Of Service
[2] Simple Api For Xml
[3] Dynamic Object Model
[4] Prevention systems
[5] Detection systems
[6] Ws-Security

راه‌حل‌های جامع‌تری برای امنیت سرویس‌های وب با کاربرد عمومی نیاز است، زیرا با وجود این استانداردها، شاهد ادامه نفوذ به سرویس‌های وب هستیم. لزوماً ارائه راه‌کارها و ابزارهایی برای مقابله با این نفوذها، در موارد متعددی ذکر شده است.

در مرجع [۷] برای جلوگیری از رخداد «خرابی حالت مشترک[1]»، استفاده از فنون «تنوع طراحی»پیشنهاد و در مرجع [۸] برای ایجاد سرویس‌های وب تحمل‌پذیر نفوذ، فنون تنوع ضمنی[2]معرفی شده است. در این معماری، معیار ارسال درخواست کاربر به یک یا چند سرویس‌دهنده، میزان تهدید فعلی است و بدترین حالت هم، زمانی است که درخواست متقاضی، سرویس بدخواهانه[3] است که مهاجم قادر خواهد بود فقط یکی از سرویس‌دهنده‌ها را به مصالحه[4] با خود درآورد و برای انجام عملیات بدخواهانه نیاز به گام‌های بیشتری دارد.

در مرجع [۹] برای تحمل‌پذیری نفوذ سرویس‌های وب، فنون‌های برنامه‌نویسی چندنگارشی و تنوع طراحی پیشنهاد شده که حاصل آن، کاهش شدید احتمال رخداد خرابی‌های یکسان در مؤلفه‌های افزونه تولید شده با فنون تنوع طراحی است. در مرجع [۱۰] راه‌حلی برای ایجاد سرویس‌های وب اتکاپذیر، با استفاده از فنون تنوع طراحی ارائه شده است. در این کار، الگویی برای پیاده‌سازی سرویس وب اتکاپذیر پیشنهاد شده که بر مبنای استفاده از فنون تنوع طراحی در سطوح مختلف معماری استوار است. همچنین به‌منظور مقابله با نفوذ در سیستم‌های نرم‌افزاری، معماری‌های خاصی از جمله SITAR، MAFTIA و SCIT ارائه شده که هر یک از آنها دارای قابلیت‌ها و محدودیت‌های خاصی می‌باشند [۱۱ و ۱۲]. معماری ارائه شده در مرجع [۱۳]، از کارهای مهم و شاخص حوزه تحمل‌پذیری نفوذ سرویس‌های وب محسوب می‌شود که برای سرویس‌های وب تحمل‌پذیر نفوذ، باکاربردهای ویژه قابل استفاده است.در این معماری، از فنون تک‌نگارشی (یکی از فنون‌های کلاسیک تحمل‌پذیر خطا) و تطبیق آن با شرایط خرابی بدخواهانه، استفاده شده و برای موارد استفاده سرویس‌های منحصر به‌فرد مطلوب بوده و فقط دارای یک نسخه پشتیبان می‌باشد که به‌منظور بازسازی سرویس وب مصالحه‌ای، مورد استفاده قرار می‌گیرد. معماری چندلایه‌ای سرویس وب تحمل‌پذیر نفوذ، معیار سنجش کارآمد بودن طرح پیشنهادی است.

۳. کلیات طرح پیشنهادی

کلیات طرح پیشنهادی، بر اساس تلفیق روش‌های متعارف امنیتی و فنون‌های تحمل‌پذیری خطا با تأکید بر استفاده از فنون‌های تنوع طراحی است. شناسایی ویژگی‌های سیستم‌تحمل‌پذیر نفوذ و آگاهی از قابلیت‌های فنون تنوع طراحی، برای ارتقای طرح پیشنهادی ضروری است که در ادامه به آنها پرداخته می‌شود.

۳-۱. ویژگی‌های سیستم‌تحمل‌پذیرنفوذ

این سیستم، دارای ویژگی‌های زیر است [۸]:

- عدم مداخله مؤلفه‌های سیستم در کارکرد یکدیگر: برای تأمین این ویژگی، باید عملیات هر مؤلفه سیستم، مستقل از سایر مؤلفه‌ها باشد تا آسیب‌پذیری‌های آنها متفاوت بوده و درصورت نفوذ در یکی از آنها، راه نفوذ به دیگر مؤلفه‌ها باز نشود.
- حفظ جامعیت داده‌ها: این ویژگی با استفاده از ساز وکارهای آزمون‌های پذیرش، بررسی و اصلاح موارد نقض جامعیت داده‌ها تحقق می‌یابد.
- محدودسازی نفوذ: این ویژگی ایجاب می‌کند حیطه تحت کنترل نفوذگر، محدود سازی شده و از گسترش دامنه آن جلوگیری شود.
- پایداری پس از نفوذ: این ویژگی سبب می‌شود حملات جلوگیری از سرویس، روی الگوریتم‌های پیکربندی مجدد، خنثی‌سازی شده و پس از نفوذ، پایداری حاصل شود.
- بازیابی نفوذ: به‌معنی بازیابی داده‌ها و مسدودسازی آسیب‌پذیری‌های منجر به نفوذ است.

۳-۲. مزایای استفاده از فنون تنوع طراحی

فنون تنوع طراحی یک ساز وکار طبیعی مهم برای کاهش حملات موفق است [۱۴]. از مسائل مهم امنیت نرم‌افزارها این است که نباید احتمال حملات واحد همزمان به مؤلفه‌ها را نادیده گرفت. اگر مؤلفه‌های مورد حمله، دارای آسیب‌پذیری‌های واحدی باشند، مهاجم با استفاده از یکسانی آسیب‌پذیری‌ها، قادر خواهد بود مؤلفه‌های بیشتری را به مصالحه بکشاند و این امر منجر به کاهش سطح امنیت سیستم می‌شود. هدف از تأکید بر استفاده مؤثر از فنون‌های تنوع طراحی در توسعه سرویس‌های وب، کاهش احتمال خرابی‌های حالت مشترک است. براساس مفاهیم برگرفته از [۱۴]، فنون تنوع طراحی، جایگزین مناسبی برای استفاده از رویه‌های درستی‌یابی و اعتبارسنجی (V&V)[5]، زمینه‌سازی کاهش خطاهای حالت مشترک در مؤلفه‌های تکراری، افزایش دادن قابلیت اطمینان سیستم و فراهم‌سازی شرایط مناسب تحمل‌پذیری خطا در سیستم است.

۴. طراحی پیشنهادی

روش‌های متعارف امنیتی، پاسخگوی همه نیازهای امنیتی سرویس‌های وب نیستند و «تحمل‌پذیری نفوذ» به‌عنوان یک راهبرد

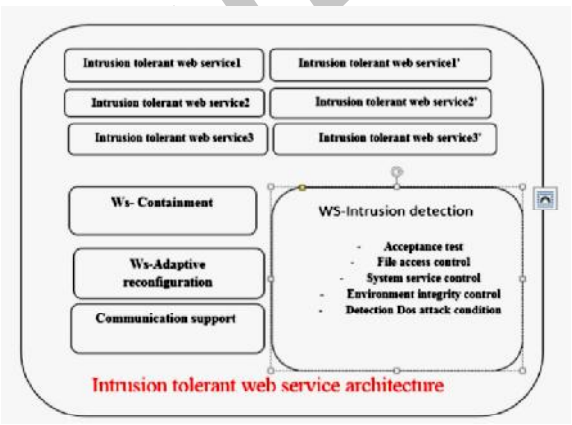
[1] Common Mode Failure
[2] Incident Diversity
[3] Malicious
[4] Compromise
[5] Validation And Verification

اساسی برای ارتقای امنیت سرویس‌های وب محسوب می‌شود و رویکرد پیشنهادی بر مبنای این راهبرد قرار گرفته است.

۴-۱. رویکرد پیشنهادی برای طراحی سرویس‌های وب تحمل‌پذیر نفوذ

طی مراحل مختلف تولید سیستم، سعی بر این است که از ایجاد آسیب‌پذیری‌های جدید جلوگیری شده و آسیب‌پذیری‌های شناخته شده رفع یا مسدود شوند. با وجود همه این اقدامات، آسیب‌پذیری‌هایی باقی می‌مانند که زمینه‌ساز بروز نفوذ بوده و نتیجه تحریکات خارجی عمدی بدخواهانه یا خطای عملیاتی هستند [۱۵]. فنون تحمل‌پذیری نفوذ، نوعی توانمندی است که برای مقابله با کلاس‌های ناشناخته آسیب‌پذیری مورد استفاده قرار می‌گیرد [۱۶].

رویکرد پیشنهادی مطابق شکل **۲** بر پایه تلفیق روش‌های متعارف امنیتی و فنون‌های تحمل‌پذیری خطا استوار است.



شکل ۲: طرح کلی سرویس وب تحمل‌پذیر نفوذ

در این طرح پیشنهادی، علاوه بر روش‌های متعارف امنیتی، از سه افزونه سرویس وب استفاده می‌شود که با فنون تنوع طراحی، توسعه داده شده است. در کنار هرکدام از افزونه‌ها، یک نسخه تکرار، به‌عنوان نسخه پشتیبان در نظر گرفته شده است. در این طرح، درخواست سرویس به‌صورت هم‌زمان در اختیار هرسه سرویس وب افزونه قرار گرفته و نتایج پردازش درخواست آنها، وارد تصمیم‌گیری شده و نتیجه نهایی تعیین‌گردد.

۴-۲. اجزای طرح پیشنهادی

طرح شامل اجزا و وظایف زیر است:

۱- مؤلفه پیشگیری از نفوذ[1]: این مؤلفه، وظیفه شناسایی و پیشگیری از نفوذ را بر عهده دارد و به این منظور از زیر مؤلفه‌های دیواره آتش سرویس وب، پایگاه‌داده الگوی نفوذ و مدیر نشست استفاده می‌کند [۱۶]. وظایف دیواره آتش سرویس وب چنین است:

- تشخیص و جلوگیری از حملات جلوگیری ضد سرویس مبتنی بر XML
- پردازش پیام‌های SOAP به منظور تشخیص ضمائم و دستورات اجرایی بدخواهانه
- اعتبارسنجی منبع درخواست[2] سرویس، به منظور شناسایی حملات جلوگیری ضد سرویس توزیع‌شده.

۲- مؤلفه پیکربندی مجدد: وظیفه این مؤلفه؛ محدودسازی فضای نفوذ، قطع ارتباط ناحیه مورد نفوذ از سایر نواحی، مکان‌یابی نفوذ و در نهایت حذف نفوذ و ترمیم مؤلفه مصالحه‌ای است [۱۶]. زیرمؤلفه‌های این مؤلفه چنین است:

- واحد محدودساز نفوذ[3]: این واحد وظیفه مکان‌یابی نفوذ و محصور کردن منطقه آن را دارد تا خرابی ایجاد شده گسترش نیابد.
- واحد پیکربندی مجدد[4]: واحدپیکربندی‌مجدد وظایف خود را با دریافت هشدارهای نفوذ یا مصالحه و با درنظر گرفتن اهداف تحمل‌پذیری نفوذ انجام می‌دهد. پیکربندی مجدد این امکان را فراهم می‌کند که طیف گسترده‌ای از راهبردهای تحمل‌پذیری نفوذ در سیستم وجود داشته و مورد استفاده قرارگیرند. این واحد تضمین می‌کند که پیکربندی سرویس‌های وب، سطح مناسبی از امنیت را داشته باشد.

۳- مؤلفه پشتیبان ارتباطات[5]: هدف این مؤلفه ترکیب روش‌های متداول امنیتی با فنون‌های تنوع طراحی است [۱۱]. اجزاء این مؤلفه عبارتند از:

- پروتکل‌های ارتباط جمعی: هدف اصلی آن، ترکیب فنون‌های کلاسیک تحمل‌پذیری خطا با فنون‌های تنوع طراحی و پیاده‌سازی است[۱۱]. استفاده مؤثر از این روش، نوعی محافظت در مقابل خرابی‌های تصادفی تکرار، محسوب می‌شود و بر پایه مفروضاتی از این قبیل که، خطاها مستقل از هم شوند و آسیب‌پذیری‌های تکرارها متفاوت باشند تا احتمال رخداد خطای طراحی یکسان در تکرارها کاهش یابد، استوار است.

۴- مؤلفه تشخیص نفوذ[6]: هدف اصلی آن، ایجاد توانایی تشخیص انواع مختلف سوءاستفاده‌ها و استفاده منفی کاربران است [۱۷]. در سیستم‌هایی که در آنها از فنون برنامه‌نویسی چندنگارشی استفاده می‌شود، تشخیص نفوذ و رفتارهای غیرعادی، با کمک آزمون‌های پذیرش و مقایسه خروجی نگارش‌های مختلف انجام می‌شود.

۵- مؤلفه تصمیم‌گیری: پس از اجرای درخواست هر یک از سرویس‌های وب و اجرای آزمون پذیرش، نتایج اجرای آن، که آزمون پذیرش، آنها را غیرمصالحه‌ای تشخیص داده‌است، به واحد

[1] Intrusion Prevention Component

[2] Source-Invoke-Authenticator
[3] Containmenter
[4] Reconfigure
[5] Communication Support
[6] Intrusion Detection Unit

تصمیم‌گیری ارسال می‌شود تا نتیجه نهایی اجرای درخواست، با استفاده از رأی‌گیری براساس الگوریتم‌های مناسب تعیین گردد [۱۸].

۴-۳. سطوح دفاعی طراحی پیشنهادی

استفاده از تنوع طراحی در توسعه سرویس‌های وب، موجب می‌شود نفوذگر با شناسایی یک آسیب‌پذیری و تدارک یک حمله موفق، قادر به تکرار نفوذ در مؤلفه‌های دیگر نباشد.

۱- درستی‌یابی درخواست‌های ورودی که با این کنترل، می‌توان نشانه‌های برخی از حملات را شناسایی و از بروز آنها جلوگیری کرد .

۲- آزمون پذیرش پاسخ‌های سرویس‌های وب به‌منظور تعیین سرویس وب مصالحه‌ای و جلوگیری از نقض تمامیت داده‌های سیستم، به عنوان وسیله تشخیص نفوذ نیز استفاده می‌شود.

۳- رأی‌گیری اکثریت که تا حدی قادر به پوشش خرابی‌ها (نتایج نفوذ) است.

۴- پیکربندی مجدد مؤلفه‌های مصالحه‌ای سرویس‌های وب.

۴-۴. قابلیت‌های طراحی پیشنهادی

۱- اعتبارسنجی درخواست کاربران در قالب بررسی درستی و مجاز بودن درخواست‌ها، این کارها با استفاده از تطبیق الگوی درخواست با الگوی تقویت شده انجام می‌گیرد و مانع از رخداد حملات تزریق SQL,XML در سرویس‌های وب می‌شود.

۲- جامعیت اطلاعاتی سرویس‌های وب.

۳- با اعمال آزمون پذیرش مناسب روی نتایج خروجی، اجرای درخواست توسط سرویس‌های وب تأمین می‌شود. با توجه به اینکه نقض جامعیت اطلاعاتی جزء اهداف مهاجمان است، در مدل پیشنهادی با استفاده از تعیین دقیق نتیجه نهایی و دوری از توافق روی خروجی نادرست، از حملات نقض کننده جامعیت، جلوگیری می‌شود.

۴- اعتبارسنجی منبع درخواست سرویس، به‌منظور شناسایی حملات DDOS انجام می‌گیرد، این کار بر عهده دیواره‌آتش سرویس وب است.

۵- از وظایف اصلی دیواره آتش سرویس وب، تشخیص علائم و جلوگیری از حملات منع سرویس مبتنی بر XML است.

۶- فراهم‌سازی امکان پشتیبانی طیف وسیعی از راهبردهای تحمل‌پذیری‌نفوذ.

۷- طراحی با قابلیت پیکربندی مجدد، امکان می‌دهد که طیف وسیعی از راهبردهای تحمل‌پذیری خطا و نفوذ در سرویس‌های وب وجود داشته باشد. بخش پیکربندی مجدد و ابزارهای در اختیار آن، از جمله نگارش‌های پشتیبان مؤلفه‌های مختلف، به‌ویژه سرویس‌های وب، موجب آماده بودن شرایط معماری برای ارتقای زمینه‌های تحمل‌پذیری نفوذ می‌شود.

۵. مدل‌سازی و ارزیابی طرح پیشنهادی

مدل‌سازی طرح پیشنهادی در دو بخش زیر انجام می‌گیرد:

۱- مدل‌سازی برای بررسی عملکردی طرح پیشنهادی

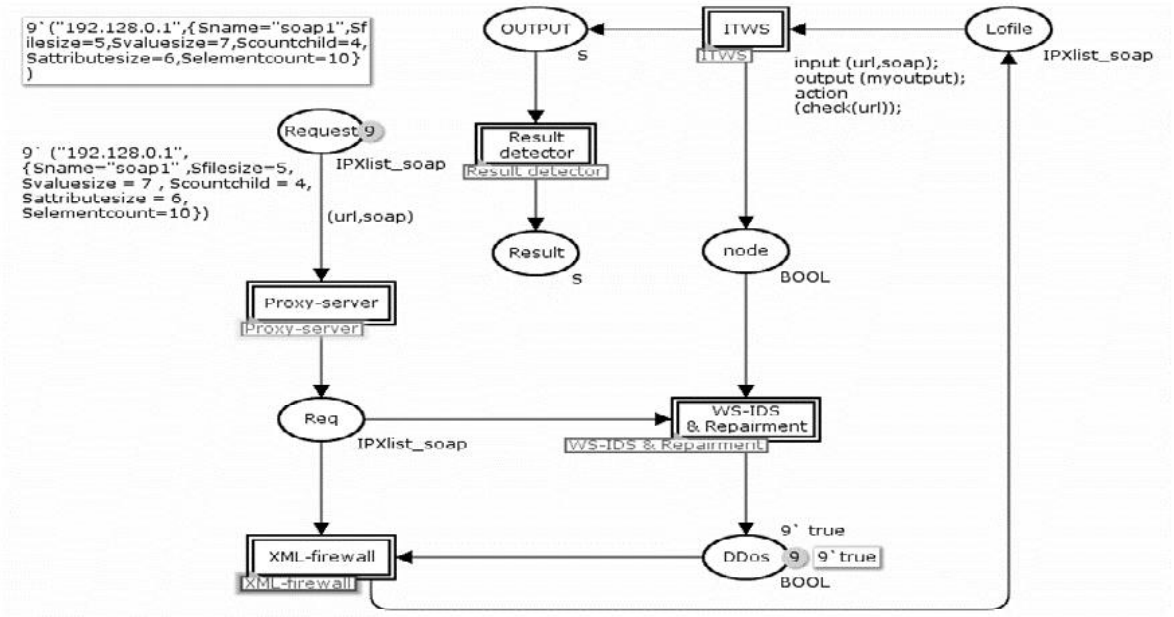
۲- مدل‌سازی و ارزیابی سرویس‌های وب توسعه داده شده با فنون تنوع طراحی.

۵-۱. تحلیل عملکرد اجزای اصلی سیستم در طرح پیشنهادی

استفاده از شبکه‌های پتری رنگی[1](CPN) برای مدل‌سازی طرح پیشنهادی، اجرای گرافیکی مدل و بررسی تغییرات نشانه‌گذاری را در هر مرحله از اجرا امکان‌پذیر کرده و شرایط دنبال کردن نحوه عملکرد هر یک از بخش‌های مدل را فراهم می‌کند. به‌منظور بررسی صحت عملکرد طرح پیشنهادی، مدل معماری آن، با استفاده از CPN Tools تهیه و در شکل ۳ ارائه شده است. بخش‌های اصلی مدل، شامل سرویس دهنده نماینده، دیواره آتش سرویس وب، واحد تشخیص نفوذ، واحد محدودسازی نفوذ و پیکربندی مجدد، واحد سرویس وب و واحد تصمیم‌گیر است که هریک در قالب گذرهای جانشینی نشان داده شده است. مکان‌های موجود در این مدل، درگاه‌های ورودی، خروجی و یا ورودی/ خروجی گذرهای جانشینی را تشکیل می‌دهند. درخواست ورودی در مکان Request قرار می‌گیرد. آنگاه گروه سرویس وب متناظرآن درخواست، توسط گذر سرویس‌دهنده نماینده Proxy-server تعیین می‌شود و سپس آن درخواست، جهت بررسی‌های لازم، در اختیار گذر جانشینی دیواره آتش سرویس وب XML-firewall قرار می‌گیرد و چنان‌چه مجاز باشد به مکان logfile انتقال داده می‌شود و در غیر این صورت قبل از ورود به داخل سیستم، رد می‌شود.

گذر جانشینی دیواره آتش سرویس وب، با اعتبارسنجی آدرس IP درخواست‌ها، درخواست‌های جعلی را از واقعی تفکیک‌کرده و درخواست‌های جعلی را قبل از ورود به بخش پردازشی سیستم، رد می‌کند. گذر جانشینی تشخیص نفوذ، در صورت تشخیص حمله جلوگیری از سرویس، مکان DDos را مقداردهی می‌کند. درخواست ورودی، پس از طی مراحل کنترلی، توسط واحدهای مختلف پیش‌بینی شده در معماری، سرویس وب لازم و مناسب را دریافت می‌نماید. در این معماری، هر درخواست توسط گروهی از سرویس‌های وب یکسان که با استفاده از تکنیک تنوع طراحی ایجاد شده‌اند، پاسخ داده می‌شوند و پاسخ هریک از آنها به درخواست ورودی، جهت تعیین نتیجه نهایی، وارد گذر جانشینی تصمیم‌گیری Result-detector می‌شود. نتیجه نهایی تعیین شده توسط گذر جانشینی تصمیم‌گیری، از طریق گذر جانشینی توزیع‌کننده امکانات، در اختیار متقاضی درخواست قرار می‌گیرد. اجرای این مدل و زیرمدل‌های مربوط به اجزای اصلی (که به‌دلیل محدودیت فضا امکان ارائه زیرمدل‌ها در این مقاله وجود ندارد)، عملکرد صحیح اجزای آن را تأیید می‌کند. براساس بررسی عملکرد هر جزء طرح پیشنهادی، روشن می‌شود که سیستم، در مقابل حملات DOS و حمله به جامعیت داده‌ای سرویس‌های وب، مقاوم است.

[1] Coloured Petri Nets



شکل ۳: صفحه اصلی مدل طرح پیشنهادی

۵-۲. مدل‌سازی بخش سرویس وب تحمل‌پذیر نفوذ طرح پیشنهادی

نمودار بلوکی طرح پیشنهادی سرویس وب تحمل‌پذیر نفوذ در فن افزونگی معمولاً برای مقابله با نفوذ و خرابی استفاده می‌شود تا درصورت خرابی یکی از مؤلفه‌های تکراری، از مؤلفه‌های افزونه، جایگزین شده و از خرابی سیستم جلوگیری شود که این از ویژگی‌های مثبت فنون افزونگی است. باید در نظر داشت استفاده از فنون افزونگی با در نظرگرفتن سربار آن، در همه شرایط مطلوب نیست و علت آن، یکسان بودن مؤلفه‌های افزونه و در نتیجه آسیب‌پذیری‌های یکسان آنها است. طرح پیشنهادی ضمن استفاده از ویژگی‌های مثبت فنون افزونگی، با بهره‌گیری از فنون تنوع طراحی، ویژگی منفی فنون افزونگی را به حداقل ممکن رسانده است. انتظار می‌رود با استفاده از افزونه‌سازی سرویس وب و بهره‌برداری از فنون‌های تنوع طراحی در توسعه آنها، یکسانی آسیب‌پذیری‌های سرویس‌های وب افزونه‌ای، به حداقل برسد تا در صورت رخداد حمله موفق به یکی از سرویس‌های وب، از توسعه و تکرار آن حمله به سایر سرویس‌های وب افزونه جلوگیری شود. برای بررسی میزان تأثیر استفاده از مؤلفه افزونه توسعه داده شده با تکنیک تنوع طراحی و تعداد آن در قابلیت سرویس‌دهی (توان عملیاتی) سیستم، طرح پیشنهادی را در سه سطح، سیستم دربرگیرنده «یک سرویس وب»، «دو سرویس وب» و سیستم دربرگیرنده «سه سرویس وب» ارائه شده است، آنگاه هریک از آنها مدل‌سازی و ارزیابی می‌شود.

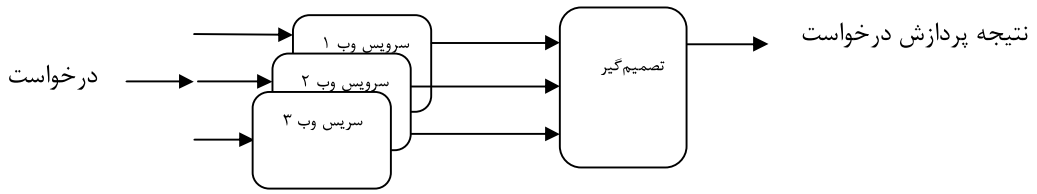
مدل‌سازی طرح پیشنهادی با زنجیره‌های مارکوف: برای مدل‌سازی رفتار سیستم نرم‌افزاری مبتنی بر سرویس وب و تعیین صفات امنیتی آن، لازم است حالات مرتبط سیستم و تعاملات بین آنها، در قالب نمودار گذر حالت و با استفاده از زنجیره مارکوف مدل شود، زیرا فاصله بین دوگذر در چنین سیستم‌هایی، معمولاً تصادفی است و نمودارهای مدل آماری مانند زنجیره مارکوف، برای این منظور، ابزار مناسبی است و به‌همین دلیل ابزار مدل‌سازی طرح پیشنهادی، زنجیره مارکوف انتخاب شده است و مفروضات بخش سرویس‌های وب طرح پیشنهادی به شرح زیر است:

۱- در هریک از سه سطح طرح پیشنهادی، هریک از سرویس‌های وب یا افزونه‌های آن، دارای چهار حالت بدون نفوذ، مورد نفوذ، آماده ترمیم و مصالحه‌ای است.

۲- سرویس وب و افزونه(های) آن، با استفاده از فنون تنوع طراحی و تولید می‌شوند.

۳- نرخ نفوذ، نرخ تشخیص نفوذ و نرخ ترمیم هریک از سرویس‌های وب و افزونه(های) آن برابر نبوده و این نابرابری به‌عنوان مصداقی برای یکسان نبودن آسیب‌پذیری‌های حالت مشترک آنها است.

۴- نسخه پشتیبان سرویس وب یا افزونه(های) آن، نسخه دوم است.



شکل ۴: ساختار بلوکی سرویس وب تحمل‌پذیر نفوذ (پیشنهادی)

مفروضات بخش سرویس‌های وب طرح پیشنهادی:

۱- در هریک از سه سطح طرح پیشنهادی، هریک از سرویس‌های وب یا افزونه‌های آن، دارای چهار حالت بدون نفوذ، مورد نفوذ، آماده ترمیم و مصالحه‌ای است

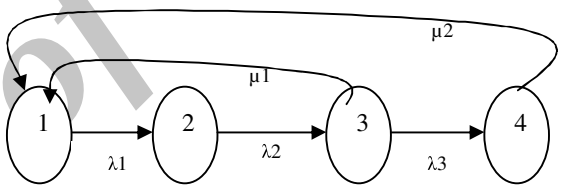
۲- سرویس وب و افزونه(های) آن، با استفاده از تکنیک تنوع طراحی و تولید می‌شوند

۳- نرخ نفوذ، نرخ تشخیص نفوذ و نرخ ترمیم هریک از سرویس‌های وب و افزونه(های) آن برابر نبوده و این نابرابری به‌عنوان مصداقی برای یکسان نبود آسیب‌پذیری‌های حالت مشترک آنها است

۴- نسخه پشتیبان سرویس وب یا افزونه(های) آن، نسخه دوم است.

سرویس وب تحمل‌پذیر نفوذ (به همراه یک نسخه پشتیبان):

در این طرح که مربوط به سیستم پیشنهادی در [۱۳] می‌باشد، از فنون تنوع طراحی استفاده نشده است و برای ترمیم آثار نفوذ در سرویس وب، از نسخه پشتیبان استفاده می‌شود. قابل توجه است که مدل سیستم، در معماری ارائه شده [۱۳] با استفاده از زنجیره مارکوف، مطابق شکل **۵** است:



شکل ۵: نمودار گذر- حالت سرویس وب با یک نسخه پشتیبان

فرض می‌شود که سرویس وب دارای مشخصات زمانی متفاوت به شرح زیر باشد:

- مورد نفوذ واقع شدن سرویس وب طبق توزیع نمایی و با نرخ λ_1 است
- تشخیص نفوذ در سرویس وب طبق توزیع نمایی و با نرخ λ_2 است
- به مصالحه درآمدن سرویس وب طبق توزیع نمایی و با نرخ $\lambda 3$ است
- زمان ترمیم سرویس وب که نفوذ در آن تشخیص داده شده است، طبق توزیع نمایی μ_1 است
- زمان ترمیم سرویس وب مصالحه‌ای، طبق توزیع نمایی μ_2 است.

در این صورت، فضای حالت این مدل مطابق جدول **۱** است و ماتریس مولد این سیستم به‌صورت زیر است:

$$Q1=\begin{bmatrix} -\lambda 1 & \lambda 1 & 0 & 0 \\ 0 & -\lambda 2 & \lambda 2 & 0 \\ \mu 1 & 0 & -(\mu 1+\lambda 3) & \lambda 3 \\ \mu 2 & 0 & 0 & -\mu 2 \end{bmatrix}$$

جدول ۱: حالات یک سرویس وب به‌همراه یک نسخه پشتیبان

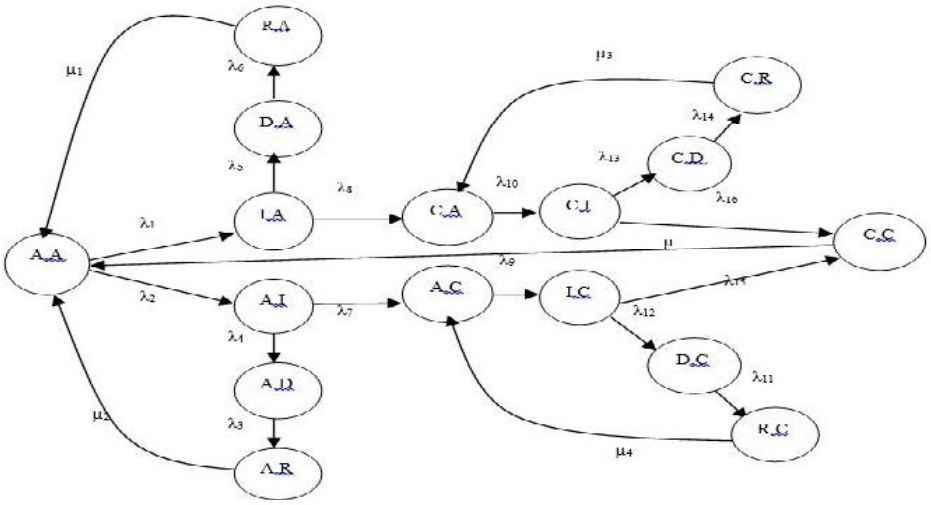
حالت	وضعیت سیستم
۱	فعال (بدون نفوذ) ws
۲	مورد نفوذ ws
۳	آماده ترمیم ws
۴	مصالحه‌ای ws

معادلات جریان‌های ورودی و خروجی سیستم چنین است:

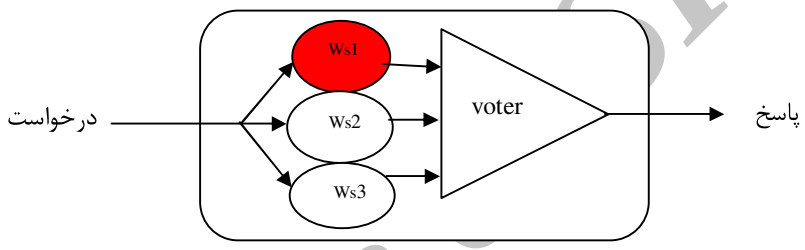
$$\begin{cases} \pi 4.\mu 2 + \pi 3.\mu 1 = \pi 1.\lambda 1 \\ \pi 1.\lambda 1 = \pi 2.\lambda 2 \\ \pi 2.\lambda 2 = \pi 3.\mu 1 + \pi 3.\lambda 3 \\ \pi 4.\mu 2 = \pi 3.\lambda 3 \\ \pi 1 + \pi 2 + \pi 3 + \pi 4 = 1 \end{cases}$$

رابطه (۱): معادلات جریان‌های ورودی و خروجی سیستم با یک سرویس وب

طرح سیستم با سرویس وب تحمل‌پذیر نفوذ و یک افزونه آن: در طرح جاری، سرویس وب و افزونه آن که با فنون تنوع طراحی تولید شده‌اند، مورد استفاده قرار می‌گیرند و مدل مارکوف آن مانند شکل ۶ است. با توجه به ویژگی‌هایی که استفاده از فنون تنوع طراحی ایجاد می‌کند، آسیب‌پذیری سرویس وب و افزونه آن یکسان نیست و این امر سبب می‌شود نفوذ احتمالی و مصالحه یکی از سرویس‌های وب، قابل پوشش و تحمل‌پذیر باشد. فرض می‌شود که سرویس وب و افزونه آن دارای مشخصات زمانی متفاوت زیر باشند: حالت A فعال بودن سرویس وب، I مورد نفوذ واقع شدن سرویس وب، D تشخیص نفوذ و R تعمیر سرویس وب و C سرویس وب مصالحه‌ای را نشان می‌دهد. براساس شکل **۷**، نرخ گذر از حالت تعمیر به حالت فعال از μ_n و برای تبدیل هرحالت غیر تعمیر به هر حالت غیرفعال، از نرخ λ_m استفاده می‌شود. جدول **۲**، حالات نمودار گذر- حالت مدل مارکوف سرویس وب و افزونه آن را نشان می‌دهد. در تکمیل طرح پیشنهادی، تعداد دو افزونه، برای هر سرویس وب تحمل‌پذیر نفوذ، در نظر گرفته شده است. قابل توجه اینکه نه تنها تعداد افزونه‌های توسعه یافته با فنون تنوع طراحی، بلکه تعداد و ترکیب محورهای تنوع طراحی نیز، در کاهش احتمال آسیب پذیری‌های یکسان مؤثر است. برای ترمیم آثار نفوذ در سرویس وب یا افزونه‌های آن، از نسخه پشتیبان استفاده می‌شود. حاصل‌ضرب دکارتی حالت‌های ممکن برای هر سرویس وب و افزونه‌های آن، حالت‌های ممکن سیستم طرح پیشنهادی را مشخص می‌کند. تعداد حالات در این شرایط برابر حاصل‌ضرب دکارتی حالات ممکن و برابر ۶۴ حالت است که به علت حجیم بودن، از ارائه آن در اینجا خودداری می‌شود ونتایج ارزیابی آن ارائه می‌شود.



شکل ۶: نمودار گذر- حالت مدل مارکوف سرویس وب با یک افزونه



شکل ۷: مدل حمله به سیستم مبتنی بر سرویس وب (استفاده از تکنیک‌های افزونگی و تنوع طراحی)

جدول ۲: جدول حالات سرویس وب و افزونه آن

حالت	Ws1	Ws2	حالت	Ws1	Ws2
۱	فعال	فعال	۹	فعال	مصالحه‌ای
۲	مورد نفوذ	فعال	۱۰	آماده ترمیم	مصالحه‌ای
۳	تشخیص نفوذ	فعال	۱۱	تشخیص نفوذ	مصالحه‌ای
۴	آماده ترمیم	فعال	۱۲	مورد نفوذ	مصالحه‌ای
۵	فعال	مورد نفوذ	۱۳	مصالحه‌ای	مورد نفوذ
۶	فعال	تشخیص نفوذ	۱۴	مصالحه‌ای	تشخیص نفوذ
۷	فعال	آماده ترمیم	۱۵	مصالحه‌ای	آماده ترمیم
۸	مصالحه‌ای	فعال	۱۶	مصالحه‌ای	مصالحه‌ای

مدل حمله به سیستم با طرح پیشنهادی مقابله با آن و مقایسه آن با سیستم ساده مبتنی بر سرویس وب: برای ارائه مدل حمله، سیستم نرم‌افزاری مبتنی بر سرویس وب در دو حالت زیر در نظر گرفته می‌شود:

۱- سیستم ساده مبتنی بر سرویس وب: در این سیستم از فنون‌های افزونگی و تنوع طراحی استفاده نشده است. در این حالت امکان تحمل‌پذیری حمله وجود ندارد و سیستم نرم‌افزاری در مقابل حمله موفق دچار شکست می‌شود. یعنی در حالتی که سیستم نرم‌افزاری مبتنی بر سرویس وب دارای سرویس‌های وب ساده باشد و در آن از فنون‌های افزونگی و تنوع طراحی استفاده نشود، میزان نفوذ به سرویس‌های سیستم، متناسب با میزان آسیب‌پذیری سرویس‌های آن است.

۲- در حالتی که سیستم نرم‌افزاری مبتنی بر سرویس وب مانند طرح پیشنهادی، دارای یک سرویس وب و دو افزونه توسعه داده شده با فنون تنوع طراحی باشد، حمله به سرویس‌های وب دارای حالات مختلف زیر است:

- حالتی که حمله‌ای رخ ندهد و سیستم دچار خرابی نمی‌شود.

- مانند شکل ۷ سه حالت وجود دارد که در هر یک از آن حالات آسیب‌پذیری یکی از سرویس‌های وب، توسط مهاجم تشخیص داده شود و حمله‌ای بر مبنای آن رخ دهد. در هر یک از این سه حالت، به علت استفاده از فنون افزونگی در توسعه سرویس‌های وب و همچنین استفاده از رأی‌گیری با الگوریتم اکثریت، حمله، پوشش داده می‌شود و سیستم دچار خرابی نمی‌شود. باید دقت نمود که پوشش حمله در شرایط بدون استفاده از فنون های افزونگی، تنوع طراحی و رأی‌گیری ممکن نیست.

- سه حالت ممکن دیگر اینکه دو سرویس و باز سه سرویس وب، دچارحمله یکسان شود. در این شرایط، سیستم دچارخرابی خواهد شد. اما استفاده از فنون تنوع طراحی در توسعه سرویس‌های وب، موجب کاهش احتمال آسیب‌پذیری‌های یکسان در آنها می‌شود. یعنی احتمال وقوع هر یک از سه حالت یاد شده نسبت به حالتی که از فنون تنوع طراحی در توسعه سرویس‌های وب استفاده نشود، کمتر است و می‌توان گفت استفاده از فنون تنوع طراحی موجب می‌شود که سیستم در مقابل نفوذ تکراری به سرویس وب مقاوم‌تر شود.

- حالت دیگر اینکه با شناسایی آسیب‌پذیری‌های متفاوت در سه سرویس وب، حملاتی به هریک از آنها صورت می‌پذیرد. در این شرایط، پاسخ نهایی قابل تعیین نخواهد بود زیرا هر سه ورودی الگوریتم رأی‌گیری، متفاوت از بقیه بوده و اکثریتی حاصل نمی‌شود، اما وجود مصالحه در بیش از یک سرویس وب تشخیص داده می‌شود ولی سرویس وب‌های مصالحه‌ای قابل تشخیص نخواهد بود.

- حالت دیگر اینکه روی هر سه سرویس وب، حمله واحدی رخ دهد و سیستم را به خرابی بکشاند. احتمال رخداد این حالت نیز به‌دلیل استفاده از فنون تنوع طراحی در توسعه سرویس‌های وب و کاهش احتمال آسیب‌پذیری‌های یکسان در سه سرویس وب، بسیار کمتر است. نتیجه اینکه در حالت a,b، سیستم دچار خرابی نمی‌شود و حمله احتمالی به‌دلیل استفاده از فنون افزونگی پوشش داده می‌شود و نشان دهنده افزایش تحمل‌پذیری نفوذ سرویس‌های وب است، زیرا در صورت عدم استفاده از فنون تنوع طراحی، امکان پوشش نفوذ نیست. در سه حالت بیان شده در c، احتمال وقوع این حالات به‌دلیل استفاده از فنون تنوع طراحی در توسعه سرویس‌های وب افزونه، نسبت به حالتی که از فنون‌های افزونگی و تنوع طراحی استفاده نشده باشد، کاهش می‌یابد. حالت d، تنها امکان تشخیص نفوذ در سرویس‌های وب به‌دلیل استفاده از فنون افزونگی فراهم می‌شود.

حالت e، احتمال رخداد این حالت نیز به‌دلیل استفاده از فنون تنوع

با توجه به تبیین آثار استفاده از فنون‌های تنوع طراحی، افزونگی و رأی‌گیری در حالات مختلف حمله و مقایسه آن با سیستم ساده مبتنی بر سرویس وب که در آن از این فنون‌های استفاده نشده است، قابل قبول است که سیستم با طراحی پیشنهادی به دلیل بهره‌برداری از فنون‌های افزونگی و تنوع طراحی در مقابل حملات به سرویس‌های وب مقاوم‌تر است که همان هدف اصلی طراحی پیشنهادی می‌باشد.

اثربخشی استفاده ازطرح پیشنهادی در افزایش امنیت سرویس وب:

فرض۱: سرویس وب Ws1 دارای آسیب‌پذیری‌های زیر باشد:
Ws1: v1, v2,v3, v4,v8

فرض ۲: سرویس‌های مثالی وب Ws1، Ws2 و Ws3 که با استفاده از فنون تنوع طراحی توسعه یافته‌اند، دارای آسیب‌پذیری‌های زیر باشند:

Ws1: v1, v2,v3, v4,v8
Ws2: v2, v5, v6,v8
Ws3: v3, v6, v7, v8

برای نشان دادن اثربخشی استفاده از طراحی پیشنهادی در افزایش امنیت سرویس وب، یک بار سیستمی در نظر گرفته می‌شود که با ارائه سرویس وب Ws1 کار می‌کند و بار دیگر از سه سرویس وب توسعه یافته با فنون تنوع طراحی به ارائه سرویس پرداخته می‌شود. وضعیت سیستم استفاده کننده از سرویس وب Ws1 و سیستم استفاده کننده از سه سرویس وب توسعه داده شده با فنون تنوع طراحی (مربوط به مثال) را نشان می‌دهد.

جدول ۳: وضعیت سیستم استفاده کننده از یک سرویس وب و سیستم استفاده کننده از سه سرویس وب در مقابل حمله

نتیجه حمله به سیستم استفاده کننده از سه سرویس وب	وضعیت سیستم استفاده کننده از سه سرویس وب درمقابل حمله	وضعیت سرویس‌های وب موجود در سیستم استفاده کننده از سه سرویس وب در شرایط رخداد حمله	نتیجه حمله به سیستم استفاده کننده از یک سرویس وب	حمله براساس آسیب‌پذیری
Masked	Normal	Ws1 Compromised Ws2 Normal Ws3 Normal	Failed	V1
Succ	Failed	Ws1 Compromised Ws2 Compromised Ws3 Normal	Failed	V2
Succ	Failed	Ws1 Compromised Ws2 Normal Ws3 Compromised	Failed	V3
Masked	Normal	Ws1 Compromised Ws2 Normal Ws3 Normal	Failed	V4
Attack detected, but not masked	Failed	Ws1 Compromised Ws2 Compromised Ws3 Compromised	Failed	V8

۵-۳. ارزیابی و مقایسه

هدف از ارزیابی طرح‌های موجود و پیشنهادی، محاسبه انواع معیارهای کارایی مربوط به هریک از آنها است که با حل حالت پایدار[1] مدل ایجاد شده و محاسبه احتمالات حالت پایدار به‌دست می‌آید. در این رابطه می‌توان معیارهای «مبتنی بر حالت[2]» مانند بهره‌وری[3] و معیارهای «مبتنی بر نرخ» مانند توان عملیاتی[4] را مورد توجه قرار داد. برای ارزیابی سیستم‌هایی که مدل آنها ارائه شده، مجموعه احتمالات متناظر با حالاتی را که در آن حالات سرویس‌دهی جریان دارد، محاسبه کرده و نتایج حاصل از آن مقایسه و تحلیل می‌شود.

ارزیابی طرح‌های مختلف سیستم مبتنی بر سرویس وب:

ارزیابی سیستم با معماری یک سرویس وب (سیستم با معماری ارائه شده در[۱۳])، با فرض پارامترها به‌صورت زیر:

λ_1	λ_2	λ_3	μ_1	μ_2
۰/۰۰۰۰۰۱	۰/۰۰۰۰۰۲	۰/۰۰۰۰۰۱	۰/۰۰۲	۰/۰۰۳

و با حل معادلات جریان‌های ورودی و خروجی، بردار جواب طرح سیستم با یک سرویس وب (بدون استفاده از افزونگی) چنین است:

P=[0.66644, 0.33322, 0.00033, 0.00000011]

ارزیابی سیستم با طرح سرویس وب تحمل‌پذیر نفوذ و یک افزونه آن: بردار جواب طرح سیستم با دو سرویس وب (با استفاده از یک سرویس وب افزونه) و با انتخاب پارامترها به صورت زیرمحاسبه شده است.

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	λ_{11}
0.00000001	0.00000002	0.00000001	0.00000002	0.00000001	0.00000002	0.00000001	0.00000002	0.00000001	0.00000002	0.00000001
λ_{12}	λ_{13}	λ_{14}	λ_{15}	λ_{16}	μ_1	μ_2	μ_3	μ_4		
0.00000002	0.00000001	0.00000002	0.00000001	0.00000002	0.0003	0.0002	0.0003	0.003		

P = [0.21092 0.07031 0.03515 0.00004 0.14061 0.28123 0.00014 0.00769 0.08413 0.00663 0.00673 0.08167 0.00842 0.00734 0.03549 0.03329]

۵-۴. مقایسه نتایج ارزیابی

مقایسه نتایج ارزیابی سیستم با طرح یک سرویس وب که نمونه آن در مرجع [۱۳] آورده شده است، طرح یک سرویس وب و یک افزونه آن، طرح یک سرویس وب و دو افزونه آن، تأثیر استفاده از افزونگی‌های حاصل از به کارگیری تکنیک تنوع طراحی در تولید آنها را نشان داده و درستی این فرضیه را اثبات می‌کند که استفاده از

[1] Steady-State Solution
[2] State-Based Measures
[3] Utilization
[4] Throughput

تکنیک‌های افزونگی و تنوع طراحی در تولید افزونه‌های سرویس وب، در افزایش تحمل‌پذیری نفوذ آنها مؤثر است.

- مجموع احتمالات متناظر با حالت‌هایی که در آن، طرح سیستم با یک سرویس وب (بدون استفاده از افزونگی) مطرح شده [۱۳] در حالت سرویس‌دهی می‌باشد، چنین است:

$$U_1 = \pi 1 = 0.66644$$

- مجموع احتمالات متناظر با حالت‌هایی که در آن، طرح سیستم با دو سرویس وب (با استفاده از یک سرویس وب و افزونه آن) در حالت سرویس‌دهی می‌باشد، چنین است:

$$U_2 = \pi 1 + \pi 2 + \pi 3 + \pi 4 + \pi 5 + \pi 6 + \pi 7 = 0.73791$$

جدول حالات ممکن طرح سیستم با یک سرویس وب و دو افزونه آن، از حاصل‌ضرب دکارتی حالات ممکن یک سرویس وب و دو افزونه آن به‌دست می‌آیند.حالت‌هایی که در آن سیستم در حالت سرویس‌دهی می‌باشد، حالت‌هایی هستند که حداقل دو مورد از سه مورد سرویس یا افزونه آن، فعال بوده و در سرویس‌دهی نقش دارند. مجموع احتمالات متناظر با این حالات چنین است:

$$U_3 = \pi 1 + \pi 2 + \pi 3 + \pi 4 + \pi 5 + \pi 9 + \pi 13 + \pi 17 + \pi 33 + \pi 49 = 0.79441$$

مجموع احتمالات متناظر با حالت‌هایی که در آن، طرح سیستم با یک سرویس وب (بدون استفاده از افزونگی) مطرح شده [۱۳] در حالت سرویس‌دهی می‌باشد، چنین است:

$$U_1 < U_2 < U_3 \text{ یا } 0.66644 < 0.73791 < 0.79441$$

که در آن، رابطه بین حالات سرویس‌دهی سه حالت سیستم پیشنهادی بیان شده است.

۵-۵. علت اعتبار پارامترهای ورودی و نتایج حاصل از محاسبات و ارزیابی‌ها

علت اعتبار مقادیر پارامترها و نتایج ارزیابی چنین است:

- درنمودارهای گذر- حالت، مقادیر نرخ‌های گذر از حالتی به حالت دیگر، براساس نمونه کار آزمایشگاهی معرفی شده در مرجع [۱۹] است.
- نمودارهای گذر- حالت مدل مارکوف سیستم در شرایط مختلف، نشان‌دهنده حالات مختلف سیستم و نرخ تبدیلات در آن است. حالات ممکن هر سیستم، نحوه گذر از حالتی به حالت دیگر براساس تعریف حالات سیستم است.
- برای ارزیابی سیستم‌های مفروض، معادلات جریان‌های ورودی و خروجی هر مدل، با توجه به نمودارهای گذر- حالت مدل مارکوف سیستم تهیه و مورد استفاده قرار گرفته است.
- محاسبات انجام گرفته برای یافتن مقادیر π_1 تا π_n، بر اساس معادلات جریان‌های ورودی و خروجی مدل‌های مارکوف است و از نرم‌افزار Maple برای محاسبات استفاده شده است. یعنی مبانی محاسبات مقادیر مورد نظر و ابزار محاسبه نرم‌افزار مطمئن است. ارزیابی سیستم در حالات مختلف، براساس مقادیر محاسبه شده، استوار است و نتایج ارزیابی با صحیح بودن مقادیر ورودی و درستی روش ارزیابی قابل قبول می‌باشند.
- استفاده از تکنیک‌های افزونگی و تنوع طراحی در توسعه سرویس‌های وب طرح پیشنهادی، موجب افزایش هزینه توسعه سرویس‌های وب و همچنین موجب افزایش تحمل‌پذیری نفوذ سرویس‌های وب می‌شود و لازم است پذیرش این دو ویژگی در کنار هم، به‌عنوان پارامترهای یک مبادله در نظر گرفته شود.

۶-۵. دلایل مناسب‌بودن طرح پیشنهادی

نتایج ارزیابی طرح پیشهادی در سه حالت متفاوت نشان می‌دهد که با افزایش تعداد مؤلفه‌های افزونه، مقدار U (احتمال متناظر با حالت‌هایی که در آن حالات، سیستم سرویس‌دهی دارد) طرح سیستم افزایش می‌یابد. علت افزایش U این است که مؤلفه‌های مذکور با استفاده از تکنیک تنوع طراحی توسعه داده می‌شوند و استفاده از تکنیک تنوع طراحی، موجب کاهش احتمال یکسانی آسیب‌پذیری‌ها در سرویس‌های وب می‌شود و زمینه افزایش مقدار U فراهم می‌گردد؛ یعنی با افزایش تعداد افزونه‌های وب سرویس، مدت زمانی که سیستم در حال سرویس‌دهی باشد، افزایش می‌یابد. در نتیجه، استفاده هم‌زمان از روش‌های متعارف امنیتی و تکنیک تنوع طراحی در پیاده‌سازی سرویس‌های وب، موجب افزایش میزان سرویس‌دهی و افزایش امنیت سرویس‌های وب شده است. در ضمن هزینه پیاده‌سازی طرح پیشنهادی به‌دلیل استفاده از تکنیک‌های افزونگی و تنوع طراحی نسبت به حالتی که از این تکنیک‌های استفاده نمی‌شود، بیشتر است و به ازای آن تحمل‌پذیری نفوذ سرویس‌های وب حاصل می‌شود که در کاربردهای عمومی توافقی مطلوب است.

۶. نتیجه‌گیری

امروزه سرویس‌های وب نقش بسیار مهمی را در کاربردهای تحت وب ایفا می‌کنند. سیستم‌های مبتنی بر سرویس‌های وب، مجموعه بزرگی از سیستم‌های توزیعی را تشکیل می‌دهند. عموماً سرویس‌های وب از سیستم‌های ناهمگن و دارای سکوهای متفاوت توزیع شده در سطح اینترنت تشکیل شده‌اند و برای توسعه آنها از معماری سرویس‌گرا استفاده می‌شود. وجود آسیب‌پذیری در نرم‌افزارهای مختلف از جمله سیستم‌های نرم‌افزاری توزیع شده که مبتنی بر وب هستند، باعث می‌شود که در حین عملیات سیستم نرم‌افزاری، نفوذگرها از وجود آنها آگاه شده و تهدیدهای بالقوه‌ای برای سیستم ایجاد شود که در اینجا لزوم استفاده از تکنیک‌های سرویس وب تحمل‌پذیر نفوذ، مطرح می‌شود. به این صورت که استفاده از این فنون، علی‌رغم وجود آسیب‌پذیری‌های اجتناب‌ناپذیر، امکان بهره‌برداری از این آسیب‌پذیری‌ها و انجام موفقیت‌آمیز نفوذها را، به هکرها و حمله‌کننده‌ها نمی‌دهد.در مقاله حاضر، ضمن شناسایی سرویس‌های وب، قابلیت‌ها، ویژگی‌ها و حملات رایج به آنها، شیوه‌های متعارف

امنیتی، تکنیک‌های تحمل‌پذیری خطا، ویژگی‌های سرویس‌های وب تحمل‌پذیر نفوذ تبیین شدند تا در رویکرد پیشنهادی مورد استفاده قرار گیرند. درادامه برای افزایش قابلیت تحمل‌پذیری نفوذ سرویس‌های وب، راهکاری مرکب ازروش‌های متعارف امنیتی مقابله با نفوذ و تکنیک‌های تحمل‌پذیری خطا، معرفی شده است.

بر اساس چارچوب طرح پیشنهادی برای سرویس‌های وب تحمل‌پذیر نفوذ، شامل سرویس وب توسعه‌یافته با تکنیک تنوع طراحی، مؤلفه پیشگیری از نفوذ، مؤلفه تشخیص نفوذ، مؤلفه پیکربندی مجدد سرویس وب و مؤلفه تصمیم‌گیری هستند. مؤلفه‌های مذکور که سرویس وب را در مقابل نفوذ، مقاوم و تحمل‌پذیر می‌نمایند، جزئی از ساختار سرویس وب می‌باشند. تحمل‌پذیری نفوذ سرویس وب، با استفاده از تکنیک تنوع تکنیک طراحی در توسعه افزونه‌های سرویس‌های وب ایجاد می‌شود. این امر موجب کاهش احتمال ایجاد آسیب‌پذیری‌های یکسان در افزونه‌ها می‌شود که نتیجه آن موجب افزایش تحمل‌پذیری نفوذ سرویس‌های وب می‌شود. برای بررسی اثربخشی استفاده از تکنیک تنوع تکنیک طراحی در طرح پیشنهادی، سرویس وب بر اساس مدل مارکوف، مدل‌سازی و آنگاه توان عملیاتی سیستم طرح پیشنهادی مورد ارزیابی قرار گرفت. همچنین برای بررسی صحت عملکردی طرح، از ابزار CPNTools به‌منظور مدل‌سازی استفاده شد. پیشنهاد مقاله حاضر این است که برای کمک به تحقیق جاری و تقویت تحمل‌پذیری نفوذ سرویس‌های وب، تحقیقات حوزه تحمل‌پذیری سرویس‌های وب در دو محور زیر انجام گیرد:

۱) تقویت هر کدام از ساز و کارهای تشخیص نفوذ، محدودسازی نفوذ، پیکربندی مجدد تا نقش هریک از ساز و کارهای مذکور در تحمل‌پذیری نفوذ پررنگ‌تر شود.

۲) بررسی تأثیر تعداد محورهای تنوع طراحی و معیارهای ترکیب آنها، برای دست‌یابی به تحمل‌پذیری بیشتر سیستم نرم‌افزاری در مقابل نفوذ، که از آن مؤلفه‌ها استفاده می‌کند.

۷. مراجع

[1] E. Cerami, Web Services Essentials First Edition, O'Reilly, 2002.

[2] A. W. Zhao and E. Moser, "Building Dependable and Secure Web Services"; J.of Software, vol. 2, No. 1, Feb. 2007.

[3] R. Gopala, K. "Guptha, Analysis of Provisioning Dependable Spot Virtual Machines in Cloud Environments"; M.S. Thesis, University of Leeds, 2011.

[4] M. Kaplan, and G. Fox, "Access Control System Using Web Services for XML Messaging Systems"; Cloud computing and Distributed systems (CLOUDS), pp. 2-3, 2010.

[5] OWASP, The Ten Critical Web Services Security Vulnerability, Available:http://www.owasp.org.

[6] M. Pal, M. Correia, "self cleansing Intrusion Tolerance"; Proc. of the 4th Workshop on Recent Advances in Intrusion-Tolerant System, 2010.

[7] E. Dunrova, "Fault Tolerant Design: An Introduction"; Kluwer Academic Publishers, 2008.

[8] D. Gorton, "Extending Intrusion Detection with Alert Correlation and Intrusion Tolerance"; M.S.Thesis, Sweden, Chalers Universityof Technology, Goteborg, 2003.

[9] R. R. Obelheiro, A. N. Bessani, L. C. Lung, M. Correia, "How Practical are Intrusion-Tolerant Distributed Systems?"; Department of Information,University of Lisbon, 2006.

[10] M. Abdollahi Azgomi and E.Nourani, "A Dependable Web Service Architecture Based on DesignDiversity Techniques and WS-BPEL"; Iranian J. of Electrical and Computer Eng. (IJECE), Accepted: Apr. 2012, In Press.

[11] I. Welch, J. Warne, P. Rayan, R. Stroud, "Architectural Analysis of MAFTIA's Intrusion Tolerance Capabilities"; Architectural Analysis of MAFTIA Intrusion Tolerance CapabilitiesTechnical Report, 2003.

[12] A. Sood, Q. L. Nguyen, "Comparative Analysis of Intrusion-Tolerant System Architectures"; IEEE Security and Privacy, 2011.

[13] Z. Agajani, M. Abdollahi Azgomi, "A Multi-Layer Architecture for Intrusion tolerant Web Services"; Int. J. of u- and e- Service, Science and Tech.,Vol. 1, No. 1 pp. pp. 73-80, 2008.

[14] A. Avizienis,J. P. J. Kelly, "Fault Tolerance by Deisgn Diversity: Concepts and Experiments"; IEEE Computer, Vol. 17, No. 8, pp. 67-80, 1984.

[15] G. T. Santos, L. C. Lung and C. Monetez, "FTWeb: A Fault Tolerant Infrastructure for Web Services"; Proceedings of the Ninth IEEE International EDOC Enterprise Computing conf., 2005.

[16] K. A. Scarfone, M. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)"; NIST, pp. 800-94, 2007.

[17] A. Ribagorda, J. Carbo, A. Orfil, "Autonomous Decision on intrusion Detection with Trained BDI Agents"; Vol. 31, No. 1, pp. 1803-1813, 2008.

[18] N. Looker, "Increasing Web Service Dependability through Consensus Voting"; Proc. of the 29th Annual Int. Computer Software and Applications Conf., 2005.

[19] A. Javad talab, "Distributed Voting for More Fault Tolerance in TMR"; proc. of Tenth Annual CSICC, pp. 334-341, 2005 (In Persian).