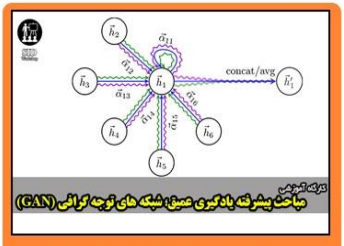
کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی
بین المللی و
ترفند های جستجو



مباحث پیشرفته یادگیری عمیق؛
شبکه های توجه گرافی
(Graph Attention Networks)

کارگاه آنلاین مقاله نویسی IEEE و ISI
ویژه فنی و مهندسی

سیزدهمین کنفرانس ملی انجمن کامپیوتر ایران
جزیره کیش، خلیج فارس، ایران ۱۹ الی ۲۱ اسفند ۱۳۸۶
انجمن کامپیوتر ایران
Computer Society of Iran
دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر
مقاله کامل

افزودن ویژگی‌های امنیتی به فرآیندهای چابک

حسین کرامتی
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف
keramati_h@mehr.sharif.edu

سیّد حسن میریان حسین‌آبادی
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف
hmirian@sharif.edu

چکیده: وجود آسیب‌پذیری‌های امنیتی متعدد در نرم‌افزارهای تولیدی و خسارات فراوانی که به این دلیل به مشتریان و شرکت‌های تولید کننده محصولات وارد آمده است، توسعه‌دهندگان سیستم‌های نرم‌افزاری را وادار به استفاده از روش‌هایی برای کاهش این مخاطرات امنیتی کرده است. از این رو، برای بالا بردن سطح امنیت در نرم‌افزارها، باید در کلیه مراحل تولید و توسعه محصول نرم‌افزاری، نیازمندی‌های امنیتی سیستم را مدنظر قرار داد. در این پژوهش، روشی ارائه شده است که طی آن، فرآیندهای چابک برای تولید نرم‌افزارهایی با سطح امنیتی بالاتر بهبود می‌یابند. در این روش، فعالیت‌هایی که برای کاهش میزان آسیب‌پذیری سیستم در چرخه تولید و توسعه نرم‌افزار می‌توانند اجرا شوند، به دقت انتخاب و به متدلوژی چابک اضافه می‌شوند به طوری که خاصیت چابکی خود را تا حد ممکن از دست ندهند. با استفاده از این روش، مهندس فرآیند می‌تواند برحسب سازمان و پروژه مورد نظر، فرآیند چابک خود را به گونه‌ای بهبود داده و تنظیم کند که محصول تولیدی از امنیت بالاتری برخوردار باشد و در عین حال برای تیم توسعه‌دهنده قابل اجرا باشد.

کلمات کلیدی: فرآیند توسعه نرم‌افزار امن، امنیت نرم‌افزار، مهندسی متدلوژی

1- مقدمه

تحقیقات در مورد قابلیت استفاده از نرم‌افزار و امنیت آن، بیشتر بر روی مسائل مربوط به کاربران سیستم و نیازهای آنان تمرکز دارد، در صورتی که برای تولید نرم‌افزارهای ایمن، باید تولیدکنندگان و توسعه‌دهندگان آن نیز مورد توجه قرار گیرند. این در حالی است که بنابر گزارشات منتشر شده، تعداد آسیب‌پذیری‌های امنیتی موجود در انواع نرم‌افزارها، روز به روز در حال افزایش است و این نشان‌دهنده ضعف در فرآیند تولید این نرم‌افزارهاست. از طرف دیگر، با گسترش استفاده از نرم‌افزارها در محیط‌های حساس و اعتماد به کارکرد آن، خسارت و ضرر ناشی از وجود آسیب‌پذیری امنیتی، برای مشتری و کاربران و در قبال آن، برای تولیدکننده نرم‌افزار هزینه بسیار زیادی را به همراه خواهد داشت. این نکته روشن است که توسعه‌دهندگان یک نرم‌افزار، نقش کلیدی را در بدست‌آمدن امنیت در آن بازی می‌کنند. در بسیاری از موارد، عدم آشنایی انجام‌دهندگان پروژه با پیش‌زمینه‌های امنیت، سبب بروز آسیب‌پذیری‌های جدی در محصول تولید می‌گردد.

حتی در صورت آشنایی توسعه‌دهندگان نرم‌افزار با مبانی کلی امنیت نرم‌افزار و دانستن تکنیک‌هایی برای برآورده کردن برخی نیازهای امنیتی و استفاده از برخی روش‌های طراحی امن، عدم وجود فرآیندی کامل ناظر بر کلیه قسمت‌های تولید و توسعه نرم‌افزار، دست‌یابی به محصولی ایمن و قابل اعتماد را با مشکل مواجه می‌کند. نکته اصلی این است که نیازمندی‌های امنیتی چیزی نیست که بتوان در برخی مراحل تولید از آن صرف‌نظر کرد و باید در طول چرخه حیات نرم‌افزار، از شروع پروژه تا تحلیل، طراحی، پیاده‌سازی، تست و استقرار و حتی مرگ سیستم، امنیت به صورتی صحیح مورد توجه قرار گیرد. بدین منظور، توسعه‌دهندگان نرم‌افزار نیاز به یک فرآیند، روش و ساز و کار تعریف شده دارند تا آنها را در شناخت و تأمین نیازمندی‌های امنیتی نرم‌افزار یاری دهد.

طی سالیان اخیر، تلاش‌هایی برای دست‌یابی به نرم‌افزارهای امن انجام شده و فرآیندهایی نیز برای تولید نرم‌افزارهای حساس به امنیت ارائه شده‌اند که شرح مختصر کارهای انجام شده در این زمینه، در قسمت بعد (کارهای مرتبط) آمده است.

در این مقاله، روشی برای بهبود امنیت در نرم‌افزارهای تولیدی توسط فرآیندهای چابک ارائه شده است. استخراج فعالیت‌های امنیتی از متدلوژی‌ها و منابع موجود، محاسبه درجه چابکی فعالیت‌ها و مقایسه‌پذیری آن با چابکی سایر فعالیت‌ها و تعریف درجه چابکی فرآیند، از جمله مفاهیمی هستند که در این مقاله ارائه شده‌اند. همچنین تعریف پارامتر میزان تحمل‌پذیری کاهش چابکی و در نهایت ارائه الگوریتمی برای افزودن فعالیت‌های امنیتی به متدلوژی چابک و کنترل میزان کاهش چابکی فرآیند در ترکیب فعالیت‌ها، هسته اصلی روش ارائه شده در این مقاله هستند.

در ادامه، کارهای مرتبط در این زمینه را بیان کرده و پس از معرفی مختصر فرآیندهای چابک و ویژگی‌های آنها، نحوه استخراج فعالیت‌های امنیتی و افزودن آن به یک متدلوژی چابک ارائه می‌گردد. تنظیم میزان انحراف فرآیند از چابکی و نتیجه‌گیری در پایان، قسمت‌های بعدی این مقاله را تشکیل می‌دهند.

2- کارهای مرتبط

در پاسخ به رشد میزان آسیب‌پذیری‌های امنیتی در محصولات نرم‌افزاری و بروز خسارات فراوان ناشی از آنها، تلاش‌هایی در زمینه تولید نرم‌افزارهایی با امنیت بالاتر انجام شده است. حاصل این تلاش‌های به صورت راهنماها، تجربیات موفق، روش‌های مدل‌سازی و مدیریت خطر، ابزار تحلیل و تست امنیتی و الگوهای طراحی امن ارائه شده است. به علاوه کارهایی در زمینه ایمن‌سازی چرخه حیات نرم‌افزار و تولید و توسعه نرم‌افزارهایی امن انجام شده که در ادامه به برخی از آنها اشاره می‌شود.

فرآیند AEGIS [2,5] متدلوژی‌ای است که بر مبنای مدل‌سازی دارایی، شناسایی نیازمندی‌های امنیتی، تحلیل مخاطرات و زمینه[1] مورد استفاده آن سیستم طراحی شده است. این متدلوژی، بعد از فاز طراحی نرم‌افزار وارد کار شده و سعی در کاهش آسیب‌پذیری‌های آن از طریق ساز و کارهای تحلیل و رفع مخاطرات دارد که البته بسیاری از قسمت‌های آن به عهده تیم انجام‌دهنده پروژه واگذار شده است و در مجموع، یک متدلوژی کامل محسوب نمی‌شود.

فرآیند دیگر برای تولید و توسعه امن نرم‌افزارها، کار انجام شده در قالب پروژه CLASP است که نگارش دوم آن در سال 2006 ارائه شده است [9]. هدف این فرآیند، فراهم کردن مجموعه‌ای از مؤلفه[2]های فرآیندی مبتنی بر فعالیت و نقش است که هسته اصلی آن شامل تجربیاتی موفق[3] برای افزودن امنیت به فرآیندهای موجود یا جدید توسعه نرم‌افزار می‌باشد و در قالب 24 فعالیت ارائه شده است. CLASP در واقع یک فرآیند نیست بلکه مجموعه‌ای از فعالیت‌هاست که می‌تواند در فرآیندهای دیگر مورد استفاده قرار گیرد.

شرکت مایکروسافت نیز تجربیات خود در زمینه امنیت نرم‌افزارهای تولیدی را جمع‌بندی و نهایتا فرآیندی برای مهندسی امنیت نرم‌افزار به نام SDL ارائه کرده است. این فرآیند به مرور تکمیل و آخرین نسخه آن در اوایل سال 2006 ارائه شد [1]. این فرآیند از 13 مرحله تشکیل شده است که به صورت ترتیبی اجرا شده و چرخه حیات نرم‌افزار را پوشش می‌دهد. در عین انسجام فعالیت‌ها در این فرآیند و قابلیت اجرای عملی آنها، فاقد مرحله‌ای برای تحلیل نیازمندی‌های امنیتی است، در صورتی که این موضوع، یکی از قسمت‌های اصلی فرآیند توسعه نرم‌افزار را تشکیل می‌دهد. همین امر باعث شده تا به جنبه‌های مختلف امنیتی توجهی نشود و تحلیل و برطرف کردن بسیاری از مخاطرات و آسیب‌پذیری‌ها، و شناخت انواع تهدیدها در این فرآیند انجام نپذیرد. محصول تولید شده توسط این فرآیند، با وجود اینکه برخی از آسیب‌پذیری‌های امنیتی در آن بسیار کم وجود دارد، مخاطرات بسیار دیگری را ممکن است در بر داشته باشد.

برای ایمن‌سازی فرآیندهای چابک، مایکروسافت در ضمیمه فرآیند SDL خود، رهنمودهایی را ارائه کرده است که طی آن، فعالیت‌های فرآیند SDL در متدلوژی چابک اجرا می‌شوند [1]. همچنین در مقاله Beznosov [3]، فرآیندهای چابک و تطبیق‌پذیری روش‌ها و تکنیک‌های امنیتی با این فرآیندها مورد بحث قرار گرفته است که در آن به متدلوژی چابک XP به صورت خاص توجه شده است و این موضوع در مقاله دیگر وی بروز بیشتری دارد [4].

زیرساخت SSE-CMM [13]، معیارهای Common Criteria [12] و سند NIST [7]، روش‌هایی برای ارزیابی فعالیت‌های امنیتی ارائه داده‌اند. پروژه UMLSec و مقالات و تحقیقات پیرامون آن در دانشگاه آکسفورد [11] برای دست‌یابی به زبانی برای توصیف و مدل‌سازی جنبه‌های امنیتی نرم‌افزار در طول فرآیند تولید آن، از دیگر کارهای انجام شده در این زمینه است. بستر TSP-Secure برای طرح‌ریزی امنیتی و بالا بردن امنیت نرم‌افزار [14] و همچنین مجموعه منابع ایمن‌سازی توسعه نرم‌افزار تحت عنوان پروژه BSI از US-CERT [10] و سند دیگری از NIST [6]، دیگر فعالیت‌ها در این زمینه را تشکیل می‌دهند.

3- فرآیندهای چابک[4] و ویژگی‌های آنها

در مقابل فرآیندهای سنتی و شی‌گرا در توسعه نرم‌افزار، خانواده جدیدی از متدلوژی‌ها تحت عنوان متدلوژی‌های چابک در سال‌های اخیر پا به عرصه نهادند که اتفاقاً در جامعه مهندسی نرم‌افزار با اقبال خوبی مواجه شدند. هدف این فرآیندها، بهینه‌سازی سرعت انجام و جلب رضایت مشتری در طول حیات نرم‌افزار می‌باشد [8]. از متدلوژی‌های XP، Scrum، FDD و DSDM می‌توان به عنوان نمونه‌هایی از فرآیندهای چابک نام برد.

در متدلوژی‌های چابک، تیم‌های پروژه نسبتاً کوچک تشکیل شده و با تعامل زیاد با مشتری و محوریت روابط انسانی بین اعضای تیم به جای مدل‌سازی و مستندات مرسوم در فرآیندهای سنتی، سعی در افزایش سرعت تولید نرم‌افزار و بالا رفتن میزان رضایت مشتری از محصول تولیدی می‌کنند. در این فرآیندها، پروژه در تکرار[5]های متعدد انجام می‌گیرد که در هر تکرار، کلیه مراحل انجام پروژه شامل تحلیل نیازمندی‌ها، طراحی، پیاده‌سازی، تست و حتی مجتمع‌سازی اجرا شده و محصول اجرایی در فواصل کوتاه آماده و در صورت امکان، در اختیار مشتری قرار می‌گیرد.

این فرآیندها از فعالیت‌هایی ساده، با کمترین مقدار مدل‌سازی و تولید مستندات حجیم، انعطاف و تکرارپذیر، دارای سرعت اجرای بالا و همراه با تعاملات شفاهی بین اعضاء تیم تشکیل شده است که در مقابل تغییرات نیازمندیها مقاوم بوده و در اکثر آنها، مشتری به صورت بلاواسطه در جریان روند انجام پروژه قرار می‌گیرد و در آن نقش ایفا می‌کند.

4- بهبود امنیتی فرآیند چابک

همانطور که قبلا ذکر شد، برای بالابردن سطح امنیت در تولید و توسعه نرم‌افزار، کارهایی انجام شده است که می‌توان از آنها برای بهبود ویژگی‌های امنیتی فرآیند چابک استفاده نمود. برای این کار، یک فرآیند چابک را که در شرکت یا سازمان مربوطه برای تولید و توسعه نرم‌افزار استفاده می‌شود و تیم انجام‌دهنده پروژه به صورت معمولی با آن کار می‌کنند را انتخاب کرده و با افزودن قابلیت‌های امنیتی، آن را بهبود می‌دهیم.

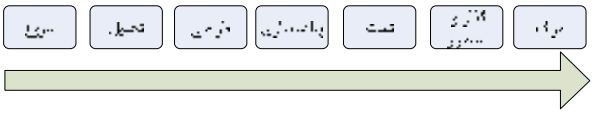
در این روش، ابتدا فعالیت‌های امنیتی که در منابع مختلف برای افزایش امنیت محصول تولیدی پشنهاد شده است را گردآوری کرده و آنها را در قالب چرخه عمومی حیات نرم‌افزار دسته‌بندی و تدوین می‌کنیم و طی الگوریتمی، آنها را به فرآیند چابک افزوده و با فعالیت‌های آن ترکیب می‌کنیم. پارامتر درجه *چابکی* را برای کلیه فعالیت‌های امنیتی و همچنین فعالیت‌های فرآیند چابک محاسبه کرده و سپس با تعیین پارامتر *میزان تحمل‌پذیری* و اعمال آن در الگوریتم ذکر شده، میزان چابکی فرآیند حاصل را کنترل می‌کنیم. در نهایت به فرآیندی چابک با قابلیت‌های امنیتی و کاهش حداقلی میزان چابکی آن دست می‌یابیم که برای انجام پروژه اجرا می‌شود.

4-1- استخراج لیست فعالیت‌های امنیتی

در نخستین مرحله، با بررسی کارهای انجام شده در زمینه تولید توسعه نرم‌افزارهای امن، فعالیت‌ها، پیشنهادات و ساز و کارهایی که برای تولید نرم‌افزارهای با امنیت بالاتر ارائه شده است، جمع‌آوری و لیست شده‌اند. همانطور که قبلاً اشاره شد، کارهای انجام شده و به خصوص تحقیقات آکادمیک در این زمینه بسیار کم است، بنابراین برای بدست آوردن این لیست، منابع متعددی شامل کارهای دانشگاهی و علمی، تحقیقات و محصولات صنعتی در زمینه تولید نرم‌افزارهای امن و همینطور مقالات پراکنده و سایتهای مرتبط با این موضوع مورد بررسی قرار گرفته و موارد قابل استفاده آنها استخراج گردیده است که در قالب70 فعالیت و زیرفعالیت استخراج شده‌اند و از این پس تحت عنوان *فعالیت‌های امنیتی* از آنها نام می‌بریم.

4-2- تنظیم فعالیت‌ها در قالب چرخه عمومی حیات نرم‌افزار

لیست فعالیت‌های امنیتی استخراج شده در مرحله قبل، از منابع متعدد و مختلفی بدست‌آمده است و بنابراین آنها را ساماندهی و طبقه‌بندی می‌کنیم تا بتوان به شکلی منظم و منطقی آنها را مورد استفاده قرار داد. این دسته‌بندی را در قالب مدل عمومی فرآیندهای تولید نرم‌افزار انجام می‌دهیم که می‌توان آنها را در یک مدل عمومی، در هفت دسته فعالیت به صورت شکل 1 توصیف کرد که از ابتدای شروع پروژه تا کنار گذارده شدن آن را پوشش می‌دهد.



شکل 1 - چرخه عمومی حیات نرم‌افزار

فعالیت‌های بدست آمده در مرحله قبل را با توجه به تطابق آن با قسمت‌های مختلف چرخه حیات نرم‌افزار، دسته‌بندی می‌کنیم و با این کار، جایگاه هر یک در روند انجم پروژه روشن و فعالیت‌های مرتبط درکنار هم قرار می‌گیرند. به عنوان مثال، شکل 2 فعالیت‌های جای گرفته در دسته فعالیت‌های مربوط به تحلیل را نشان می‌دهد که البته برخی از آنها دارای چند زیرفعالیت نیز می‌باشند که در این شکل دیده نمی‌شود.

4-3- محاسبه درجه چابکی فعالیت‌ها

فرآیندهای چابک دارای ویژگی‌هایی هستند که آنها را از سایر انواع فرآیندهای تولید و توسعه نرم‌افزار متمایز می‌سازد. همین ویژگی‌ها هستند که این فرآیندها را در بسیاری از پروژه‌ها نسبت به متدلوژی‌های سنتی و حتی شی‌گرا برتری داده و با هزینه کمتر و سرعت بیشتر، محصول مورد پسند مشتری را تولید می‌کنند. برای تولید و توسعه محصول با امنیت مناسب، انجام فعالیت‌های خاصی در این زمینه مورد نیاز می‌باشد که آنها را از لیست فعالیت‌های بدست آمده در مراحل قبل انتخاب می‌کنیم. فعالیت‌های امنیتی انتخاب شده باید با فرآیند چابک مورد نظر قابل تجمیع و استفاده باشند، به همین منظور، پارامتری تحت عنوان درجه *چابکی* را برای هر یک از این فعالیت‌ها محاسبه می‌کنیم. درجه چابکی هر فعالیت براساس میزان تناسب آن با فرآیندهای چابک تخمین زده می‌شود، بدین صورت که ویژگی‌های عمومی فرآیندهای چابک را در نظر گرفته و به ازای هر فعالیت از لیست فعالیت‌های امنیتی، میزان تطبیق هر ویژگی با آن فعالیت را با عددی بین صفر تا 5 معین می‌کنیم. هر چه آن فعالیت با ویژگی مورد نظر تطابق بیشتر داشته باشد، این عدد به 5 نزدیک‌تر می‌شود و برای فعالیتی که آن ویژگی را ندارد، عدد صفر منظور می‌گردد. این مقادیر با بررسی دقیق نحوه اجرای هر فعالیت و مقایسه آن با ویژگی مورد نظر بدست می‌آید. در نهایت، جمع مقادیر مشخص شده برای هر ویژگی، درجه چابکی آن فعالیت را تشکیل می‌دهد. جدول 1، نحوه محاسبه درجه چابکی برای چند فعالیت امنیتی را نشان می‌دهد.

ویژگی چابک / فعالیت	سادگی	قابلیت تعامل با مشتری	عدم نیاز به تولید مستند و مدل‌سازی	تحمل تغییرات نیازمندیها	سرعت اجرا	غیر رسمی بودن	مبتنی بر افراد	تکرار پذیری	انعطاف پذیری	درجه چابکی فعالیت
شناسایی حملات	2	3	3	4	2	5	2	5	4	30
مدل‌سازی تهدید	1	1	0	3	2	3	1	4	1	16
تحلیل نیازمندی‌های امنیتی	4	4	3	3	4	5	4	4	4	35
آموزش و اطلاع رسانی امنیتی	3	4	5	5	5	5	5	5	5	42
تشکیل تیم امنیت	4	4	5	5	5	5	5	5	5	43
شناسایی منابع	2	3	3	1	3	3	2	4	3	24
شناسایی نقش‌ها	3	4	3	2	3	4	4	4	3	30
مرور جنبه‌های امنیتی طراحی	3	3	2	3	5	4	4	5	5	34
تحلیل ایستای کدهای برنامه تولیدی	5	5	2	5	5	4	5	5	1	37
استفاده از قابلیت‌های امنیتی کامپایلرها	5	0	5	5	5	5	5	5	2	37
تست آسیب‌پذیری	0	3	2	4	3	4	4	5	5	30
طرح ریزی نحوه پاسخگویی به حملات	2	5	3	3	2	4	4	2	4	29

جدول 1 - محاسبه درجه چابکی برای چند فعالیت امنیتی

4-4- ترکیب‌پذیری فعالیت‌های امنیتی و چابک

برای استفاده از فعالیت‌های امنیتی در فرآیندهای چابک، باید آنها را با موتور اصلی و فعالیت‌های آن فرآیند ترکیب کرد و با این کار، جایگاه و نحوه اجرای فعالیت‌های امنیتی در چرخه تولید و توسعه نرم‌افزار مشخص می‌گردد. بدین منظور، فرآیندهای چابک را مورد تحلیل می‌دهیم و فعالیت‌های آن و ترتیب و نحوه اجرای آنها را بدست می‌آوریم که این کار را برای هفت فرآیند چابک شامل DSDM، Scrum، XP، ASD، dX، FDD و Crystal Clear انجام دادیم و فازها، فعالیت‌ها و زیرفعالیت‌های اصلی و نحوه اجرای هر یک از آنها استخراج گردیده است.

شکل 2 - فعالیت‌های مربوط به تحلیل امنیتی سیستم

پس از آن، فعالیت‌های بدست‌آمده از فرآیند چابک مورد نظر با فعالیت‌های امنیتی بدست آمده در مراحل قبل را در یک جدول متقاطع قرار داده و خانه‌های آن را با مقدار صفر یا یک پر می‌کنیم. به ازای هر

فعالیت امنیتی، کلیه فعالیت‌های بدست آمده از متدلوژی چابک در نظر گرفته می‌شود و در صورت امکان‌پذیر بودن ترکیب آنها با یکدیگر از نظر مفهومی و ساختاری و قابلیت توسعه فعالیت چابک با آن فعالیت امنیتی، مقدار یک و در غیر این صورت، مقدار صفر برای تقاطع آنها منظور می‌گردد. به عنوان مثال، فعالیت امنیتی "*تست آسیب‌پذیری*" با فعالیت "*تست نهایی*" از فرآیند چابک dX قابل ترکیب می‌باشد ولی با فعالیت "*طراحی*" و یا حتی فعالیت "*پیاده‌سازی مبتنی بر تست*" از همان فرآیند ترکیب‌پذیر نیست. این مقدار صفر یا یک را *پارامتر ترکیب‌پذیری* برای دو فعالیت می‌نامیم.

4-5- ترکیب فعالیت‌ها

با توجه به ذات چابک بودن فرآیند، فعالیت‌های امنیتی که برای ترکیب با فعالیت‌های فرآیند چابک انتخاب می‌شوند، باید تا حد امکان، باعث سنگین شدن فرآیند و کاهش ویژگی چابک بودن آن نشوند. برای هر یک از فعالیت‌های فرآیند چابک مورد نظر،
درجه چابکی را همانند فعالیت‌های امنیتی، محاسبه می‌کنیم و میانگین آنها را به عنوان درجه *چابکی فرآیند* در نظر می‌گیریم. همچنین پارامتر جدیدی به نام *میزان تحمل‌پذیری* را برای فرآیند چابک تعریف می‌کنیم که هر چه مقدار آن بیشتر باشد، فعالیت‌های امنیتی با درجه چابکی کمتر، شانس بیشتری برای ترکیب با فعالیت‌های فرآیند چابک پیدا خواهند کرد. با ترکیب دو فعالیت امنیتی و چابک، درجه چابکی فرآیند حاصل، برابر با حداقل درجه چابکی آن دو خواهد شد. ترکیب دو فرآیند را با علامت ⊕ نشان خواهیم داد.

4-6- الگوریتم انتخاب و ترکیب فعالیت‌ها

انتخاب فعالیت‌های امنیتی برای افزودن به فرآیند چابک و ترکیب با فعالیت‌های آن توسط الگوریتم زیر و در 6 مرحله انجام می‌شود:

1. از بین فعالیت‌های امنیتی، فعالیت با بالاترین درجه چابکی انتخاب می‌گردد:

$$ad(SA_x) = \max[\ ad(SA_i)\ ] \qquad i = 1\ ..\ n$$

که در آن فعالیت‌های امنیتی با علامت SA و فعالیت‌های چابک را با PA نشان داده و درجه چابکی هر فعالیت با تابع ad(x) نمایش داده شده‌اند و n تعداد فعالیت‌های موجود در لیست مجموعه فعالیت‌های امنیتی و x، فعالیت انتخاب شده از آن مجموعه می‌باشد.

2. فعالیت‌های چابکی که دارای پارامتر ترکیب‌پذیری مخالف صفر با فعالیت امنیتی انتخاب شده باشند، مجموعه‌ای را تشکیل می‌دهند و از بین آنها، فعالیتی که درجه چابکی آن از همه بیشتر باشد، با فعالیت امنیتی انتخاب شده ترکیب می‌گردد.

با ترکیب دو فعالیت امنیتی و چابک SA_x و PA_i و در صورتی که پارامتر ترکیب‌پذیری آنها، CP_{ix} مخالف صفر باشد، فعالیت جدیدی با درجه چابکی برابر با مینیمم درجه چابکی آنها بدست می‌آید و جایگزین فعالیت چابک اصلی در فرآیند می‌گردد (PA_i'). در صورتی که پارامتر ترکیب‌پذیری آنها مقدار صفر داشته باشد، آن دو فرآیند ترکیب نمی‌گردند. Set(PA) نشان‌دهنده مجموعه فرآیندهای چابک می‌باشد.

$$PA_i = y \mid y \in Set(PA) \wedge CP_{ix} \neq 0 \wedge \forall_{z \in Set(PA)}\ ad(y) \geq ad(z)$$

$$ad(PA_i') = \min[\ ad(PA_i)\ ,\ ad(SA_x)\ ]$$
$$PA_i' = PA_i \oplus SA_x$$
$$Set(PA') = Set(PA) - PA_i + PA_i'$$

3. در صورتی که در مرحله قبل، هیچ فعالیت چابکی با فعالیت امنیتی انتخاب شده ترکیب نشده باشد، فعالیت انتخاب شده را از لیست فعالیت‌های امنیتی حذف می‌کنیم و به مرحل 1 الگوریتم باز می‌گردیم.
4. پس از ترکیب فعالیت انتخاب شده با فعالیتی در فرآیند چابک، درجه چابکی فرآیند جدید دوباره محاسبه می‌گردد. در صورتی که اختلاف درجه چابکی جدید با درجه چابکی فرآیند اولیه کمتر از پارامتر میزان تحمل‌پذیری باشد، فعالیت امنیتی انتخاب شده از لیست فعالیت‌های امنیتی حذف شده و برای ترکیب فعالیت بعدی، به مرحله 1 الگوریتم برمی‌گردیم.
5. در صورتی که اختلاف درجه چابکی جدید و اولیه، بیشتر از میزان تحمل‌پذیری باشد، آخرین ترکیب انجام شده، لغو می‌گردد و الگوریتم متوقف می‌شود.
6. در صورت خالی شدن لیست فعالیت‌های امنیتی، الگوریتم متوقف می‌شود.

پس از اجرای الگوریتم و ترکیب فعالیت‌ها به صورت فوق، به فرآیندی جدید می‌رسیم که تا حد امکان فعالیت‌های امنیتی به آن اضافه شده است و میزان چابکی فرآیند تا حد قابل قبول حفظ شده است.

4-7- پارامتر میزان تحمل‌پذیری

فعالیت‌هایی که به بالاتر رفتن سطح امنیت نرم‌افزار کمک می‌کنند، در بسیاری از موارد همراه با کار نسبتاً زیاد، تولید مستندات و انجام عملیات تحلیلی رسمی[6] و شبه رسمی هستند. بنابراین ترکیب آنها با فرآیندهای چابک، باعث کاهش خاصیت چابکی آنها شده و ممکن است برای تیم چابک انجام دهنده پروژه قابل تحمل نباشد و یا با هزینه قابل توجه و کاهش سرعت انجام پروژه همراه باشد. از طرف دیگر، امنیت در

سیستم تولیدی از اهمیت خاصی برخوردار است و برای نیل به آن، می‌توان مقداری از کاهش چابکی و پیرو آن، افزایش احتمالی هزینه‌ها و کاهش سرعت اجرای پروژه را تحمل کرد.

در روش ارائه شده، تعریف درجه چابکی برای فعالیت‌های امنیتی، فعالیت‌های چابک و کل فرآیند، به منظور کنترل پذیری و قابل مقایسه کردن چابکی فعالیت‌ها نسبت به هم و فرآیند جدید نسبت فرآیند اولیه انجام شده است. به علاوه در الگوریتم انتخاب فعالیت‌های امنیتی، به صورت بهینه عمل می‌شود و با افزودن هر فعالیت، کمترین کاهش چابکی در فرآیند اصلی اتفاق می‌افتد. همچنین پارامتر میزان تحمل‌پذیری، وظیفه کنترل میزان کاهش چابکی سیستم در طی افزودن فعالیت‌های امنیتی به فرآیند چابک را بر عهده دارد. هر چه مقدار این پارامتر کمتر باشد، فرآیند حاصل از چابکی بیشتری برخوردار خواهد بود و مقدار صفر برای آن، به معنی عدم کاهش چابکی فرآیند اولیه می‌باشد. افزایش مقدار این پارامتر، به معنی کاهش چابکی فرآیند و از طرف دیگر، افزودن فعالیت‌های امنیتی بیشتر و در نتیجه آن بالا رفتن سطح امنیتی محصول تولید می‌باشد.

مهندس متدلوژی برای تعیین مقدار پارامتر تحمل‌پذیری باید بتواند بین هزینه ناشی از افزایش این پارامتر برای سازمان و تیم انجام‌دهنده پروژه از یک طرف و خسارت ناشی از پائین بودن سطح امنیتی سیستم تولیدی از طرف دیگر تعادل برقرار کرده و این پارامتر را طوری تنظیم کند که در مجموع، منفعت آن غالب باشد. به عنوان مثال، خسارت ناشی از امنیت پائین محصول نهایی، می‌تواند با استفاده از روش‌های مدیریت مخاطره[7] و محاسبه احتمال و میزان خسارت ناشی از مخاطرات احتمالی بدست آید. از طرف دیگر، کاهش هر واحد چابکی فرآیند یا هر یک از فرآیندها، هزینه‌ای به همراه خواهد داشت که البته مقدار آن برای هر سازمان، تیم انجام دهنده و پروژه خاص متفاوت است. با برقراری تعادل بین هزینه/منفعت و تنظیم پارامتر میزان تحمل‌پذیری در این روش، به امنیت مطلوب در نرم‌افزار تولیدی توسط فرآیند چابک خود، دست می‌یابیم.

5- کارهای آینده

روش ارائه شده در این مقاله برای بهبود جنبه‌های امنیتی فرآیند چابک با افزودن فعالیت‌های امنیتی را می‌توان با تغییراتی بهبود بخشید. با وزن دادن به فعالیت‌ها در محاسبه درجه چابکی کل فرآیند بر اساس تعداد تکرار اجرای آنها در کل چرخه حیات نرم‌افزار و همچنین وزن دادن به ویژگی‌های چابکی فعالیت‌ها، می‌توان دقت محاسبه درجه چابکی فرآیند و فعالیت‌ها را بهبود بخشید.

از آنجا که بسیاری از فعالیت‌های امنیتی، سنگین بوده و از چابکی پائینی برخوردار هستند، می‌توان با اعمال تغییراتی در آنها، درجه چابکی را افزایش داده تا شانس به کار رفتن در فرآیند چابک را بدست آورند. یکپارچه‌سازی و کنترل سازگاری فعالیت‌های انتخاب شده، فازی[8] کردن پارامتر ترکیب‌پذیری فعالیت‌های امنیتی با فعالیت‌های فرآیند چابک و نهایتاً تعریف وابستگی بین فعالیت‌های امنیتی و اطمینان از انتخاب شدن فعالیت‌های امنیتی مرتبط به هم در فرآیند چابک نهایی، از دیگر بهبودهایی است که برای دست‌یابی به فرآیندی با قابلیت‌های امنیتی بیشتر، در این روش قابل اعمال است.

6- نتیجه

با استفاده از روش ارائه شده در این مقاله، متدلوژی‌های چابکی که در تولید و توسعه محصولات نرم‌افزاری مورد استفاده قرار می‌گیرند، از جهت امنیتی بهبود یافته و فعالیت‌های امنیتی به آن افزوده می‌شوند. مهندس متدلوژی می‌تواند برحسب نیازمندی‌های امنیتی پروژه و تحلیل هزینه/منفعت کاهش چابکی و افزایش سطح امنیت، با مقداردهی صحیح به پارامتر تحمل‌پذیری، درجه چابکی فرآیند حاصل را کنترل کند.

مراجع

[1] Howard, M., Lipner, S., *"The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software"*, Microsoft Press, 2006.

[2] Flechais, I., *"Designing Secure and Usable Systems"*, Ph.D. Thesis, University of London, London, UK, 2005.

[3] Beznosov, K., Kruchten, P., *"Towards Agile Security Assurance"* In Proceedings of the 2004 Workshop on New Security Paradigms, 2005.

[4] Beznosov, K., *"Extreme Security Engineering: On Employing XP Practices to Achieve 'Good Enough Security' without Defining It."*, First ACM Workshop on Business Driven Security Engineering, 2003.

[5] Flechais, I., Sasse, M. A., Hailes, S. M. V., *"A process for developing secure and usable systems"*, In Proceedings of the 2003 Workshop on New Security Paradigms, 2003.

[6] Grance, T., Hash, J., Stevens, M., *"Security Considerations in the Information System Development Life Cycle"*, NIST, Computer Security Division, NIST Special Publication 800-64, REV. 1, 2004.

[7] Swanson, M., etc, *"Security Metrics Guide for Information Technology Systems"*, NIST, Computer Security Division, NIST Special Publication 800-55, 2003.

[8] Agile Alliance, *"Manifesto for Agile Software Development"*, 2005,

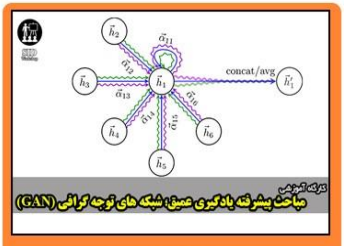
http://www.agilealliance.org.
[9] Secure Software Inc., *"CLASP: Comprehensive Lightweight Application Security Process"*, Version 2.0, 2006, http://www.securesoftware.com/process.
[10] US-CERT, Software Engineering Institute, *"Build Security In"*, 2006, https://buildsecurityin.us-cert.gov.
[11] Jürjens, J., *"Developing Secure Systems with UMLsec From Business Processes to Implementation"*, UMLsec homepage, 2002, http://www4.in.tum.de/~umlsec.
[12] The Common Criteria Portal, 2007, http://www.commoncriteriaportal.org.
[13] Systems Security Engineering – Capability Maturity Model (SSE-CMM) official web site, 2007, http://www.sse-cmm.org.
[14] TSP for Secure Systems Development – Presentation, 2002, http://www.sei.cmu.edu/tsp/tsp-secure-presentation.

زیرنویس‌ها

[1] Environment
[2] Component
[3] Best Practices
[4] Agile Methodologies
[5] Iteration
[6] Formal
[7] Risk Management
[8] Fuzzy

کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی
بین المللی و
ترفند های جستجو



مباحث پیشرفته یادگیری عمیق؛
شبکه های توجه گرافی
(Graph Attention Networks)

کارگاه آنلاین مقاله نویسی IEEE و ISI
ویژه فنی و مهندسی