Journal of Information and Communication Technology

Volume 16, Issue 59-60, Spring and Summer 2024, pp. 144-166

A Survey on Computer Security Patterns and Proposing a New Perspective

Seyed Hadi Sajjadi [1*], Reza Kalantari[1]

[1] Assistant Professor, ICT Research Institute (Iran Telecommunication Research Center), Tehran, Iran

Received: 28 September 2022, Revised: 20 January 2023, Accepted: 29 May 2023
Paper type: Research

Abstract

In this article, at the beginning, the use of computer security models and its benefits are discussed in a new way. Then, while briefly introducing the space of computer security encounters in the form of ontology, three perspectives in the study of patterns in this field have been identified and distinguished from each other. These three perspectives are secure models, security models, and the framework and system to security models. The first and last perspectives are briefly explained and the second perspective is studied in detail from the perspective of the organization of patterns, including the five types of organization. The five types mentioned include software-based lifecycle organization, logical-level organization-based organization, threat-based classification-based organization, attack-based classification-based organization, and application-based organization. In this type of introduction of patterns, the audience acquires a comprehensive view of the discourse of computer security patterns and acquires the necessary knowledge to make better use of these patterns. Finally, the analysis and idea of this research are presented in the form of introducing a new type of organization in order to facilitate the proper use and addressing of patterns. It is stated that the existing categories are mostly static and forward-looking and do not have the necessary dynamism and backwardness, and the idea of covering all stakeholders and security ontology can have this feature and, include agile patterns as well. Based on this idea and related analyzes, the atmosphere of future research activities will be revealed to the audience.

Keywords: Information Security, Security Model, Threat, Vulnerability, Attack, Ontology.

[*] Corresponding Author's email: h.sadjadi@itrc.ac.ir

دوفصلنامه
فناوری اطلاعات و ارتباطات ایران

سال شانزدهم، شماره های ۵۹ و ۶۰، بهار و تابستان ۱۴۰۳، صفحه ۱۴۴ الی ۱۶۶

مرور الگوهای امنیت رایانه‌ای و پیشنهاد یک دیدگاه جدید

سیدهادی سجادی[1]*، رضا کلانتری[1]

[1]استادیار، پژوهشگاه ارتباطات و فناوری اطلاعات (مرکز تحقیقات مخابرات ایران)، تهران، ایران

تاریخ دریافت: ۱۴۰۱/۰۷/۰۶ تاریخ بازبینی: ۱۴۰۱/۱۰/۳۰ تاریخ پذیرش: ۱۴۰۲/۰۳/۰۸

نوع مقاله: پژوهشی

چکیده

در این مقاله، نخست به شیوه‌ای بدیع به موضوع چرایی استفاده از الگوهای امنیت رایانه‌ای و مزایای آن پرداخته شده است. سپس ضمن معرفی اجمالی فضای مواجهات امنیت رایانه‌ای در قالب هستان‌شناسی، برای اولین بار سه دیدگاه در زمینهٔ مرور الگوهای این حوزه، شناسائی و از یکدیگر تمیز داده شده است. این سه دیدگاه شامل دیدگاه الگوهای امن که ناظر بر طراحی امن الگوهای متداول طراحی نرم‌افزار است؛ دیدگاه الگوهای امنیت که صرفاً به الگوهای امن‌سازی با کارکرد تماماً امنیتی اشاره دارد؛ و دیدگاه چارچوب و سیستم به الگوهای امنیت است که این دسته نیز کارکرد تماماً امنیتی داشته ولی نظام دسته‌بندی آن متفاوت از دیدگاه دوم است. دو دیدگاه اول و سوم به‌طور خلاصه توضیح داده‌شده و دیدگاه دوم نیز ازمنظر سازمان الگوها شامل پنج نوع سازماندهی، مورد تحقیق مفصل قرار گرفته است. در این نوع معرفی الگوها، مخاطب از منظری جامع با انواع و حوزه‌های عمل الگوهای امنیت رایانه‌ای آشناشده و آگاهی موضوعی و زمینه‌ای لازم برای استفاده بهتر از این الگوها را کسب می‌نماید. در انتها، ایدهٔ این پژوهش در قالب معرفی نوعی جدید از سازماندهی به منظور تسهیل در استفاده و آدرس‌دهی مناسب‌تر الگوها ارائه شده‌است. در این ایده بیان شده‌است که دسته‌بندی‌های موجود، عمدتاً ایستا و پیشانِگر بوده و از پویایی لازم و خصلت پسانِگری برخوردار نیستند و ایدهٔ مبتنی بر پوشش همهٔ ذی‌نفعان و هستان‌شناسی امنیت، می‌تواند این خاصیت را داشته باشد و به‌علاوه، الگوهای چابک را نیز در خود جای دهد. مبتنی بر این ایده و تحلیل‌های مرتبط، فضای فعالیت‌های پژوهشی آینده نیز برای مخاطب آشکار می‌گردد.

کلیدواژگان: امنیت اطلاعات، الگوی امنیت، تهدید، آسیب‌پذیری، حمله، هستان‌شناسی.

* رایانامهٔ نویسنده مسؤول: h.sadjadi@itrc.ac.ir

۱- مقدمه

عموماً هدف اصلی در انجام یک پژوهش مروری[1] (رجوع شود به مفهوم مرور در[۱]) در یک حوزهٔ علمی، درک دقیق از اقدامات صورت‌گرفته و نیز آگاهی یافتن نسبت به دایره و قلمرو کارهای مورد نیاز این حوزه است. این مقاله در خصوص مرور الگوهای امنیت رایانه‌ای است. پس‌از انجام این فعالیت، روش جدیدی از معرفی الگوها ارائه می‌گردد. در ابتدا لازم است تعریف و منظور خود را از الگوی امنیت رایانه‌ای بیان کنیم.[۲] یک الگوی امنیت[3]، توصیف کنندهٔ یک مشکل تکراری ویژه در امنیت است که در یک حوزه‌خاص اعم از طراحی و پیاده‌سازی سامانه‌ها ارائه شده و دربردارندهٔ بهترین راه‌حل پذیرفته شده، تاکنون است[۲].

وجود الگو موجب می‌شود طراحان و توسعه‌دهندگان در مسائل مشترک نیازی به تولید مجدد راه‌حل‌های امنیتی نداشته باشند[۳و۴]. امروزه استفاده از الگوها در بسیاری از موضوعات جدید مثل امنیت ابری، امنیت اینترنت اشیاء[۵] و زنجیره بلوکی[۳] مورد استفاده قرار می‌گیرد. در زمینه الگوهای امنیت پرسش‌های زیر را مطرح می‌کنیم: دلیل بوجود آمدن الگوهای امنیت چیست؟ چه مزایایی توسط آنها حاصل می‌شود؟ چگونه می‌توان از آنها استفاده نمود؟ معایب روش‌های فعلی در استفاده و مواجهه با آنها چیست؟ و آخر اینکه آیا می‌توان روشی نو در زمینه معرفی و ارجاع به استفاده از آنها ارائه نمود؟

در این مقاله به همه پرسش‌های فوق پاسخ داده‌شده است و روش جدیدی نیز در معرفی و دسترسی به الگوها ارائه می‌شود.

هدف از انجام این تحقیق، طرح یک تصویر بزرگتر و ترسیم فضای وسیع‌تری از کاربرد الگوهای امنیت رایانه‌ای برای توسعه‌دهندگان سامانه‌های رایانشی، مهندسان و مدیران امنیت رایانه‌ای است تا قادر باشند با هزینه کمتری به ایجاد و نگهداری امنیت سامانه‌ها و سامانه‌های امن بپردازند و تجارب بشری را در خلق الگوها مورد استفاده قرارداده و نیز الگوهای جدیدی را خلق نمایند.

در این مقاله ابتدا در بخش دوم مفهوم امنیت و فضای مواجهات امنیت را با استفاده از هستان‌شناسی کلان امنیت فضای رایانه‌ای تشریح می‌کنیم. این بخش دیدی کلی به خواننده در باره امنیت داده و کمک می‌کند تا تفکیک لازم را بین مفاهیم برقرار نماید. در بخش سوم با بیانی جدید و استدلالی متفاوت از استدلال‌های موجود در گفتمان الگوهای امنیت رایانه‌ای، ضرورت و چرایی توجه به الگوهای امینت و اهمیت آنرا تشریح می‌کنیم. این نحو تشریح در جای دیگری تاکنون نیامده و به نوعی ابداع این مقاله است. در بخش چهارم مقاله گذری سریع به تاریخچه مطرح شدن مفهوم الگوی امنیت رایانه‌ای پرداخته و زمینه را برای ورود به مبحث اصلی مرور الگوها فراهم می‌کنیم.

در بخش پنجم ما سه دیدگاه را در خصوص الگوهای زمینه شناسایی کردیم. این دسته‌بندی نیز اولین بار در این مقاله انجام شده است. اولین دسته شامل الگوهای امن یعنی همان الگوهای طراحی معروف به گنگز آو فور[4] هستند که دارای اثر امنیتی هستند و جوری طراحی شدند که در کنار ارائه کارکرد سیستمی مورد نظر، صفت امنیت در آنها برجسته بود و به‌عبارتی در بخش صفات کیفی آنها، صفت کیفی امنیت موردنظر قرار گرفته است. در حوزه مباحث تخصصی امنیت ممکن است به این الگوها توجه نشود ولی طرح آن توسط این مقاله توجه مخاطب را به این موضوع جلب می‌کند که الگوهایی وجود دارند که امنیت در آنها درونی شده است و اگر این ملاحظه درنظر گرفته نمی‌شد، چه بسا امنیت باید به صورت امری عارضی برای آن الگوها طراحی می‌شد. در دسته دوم الگوهای امنیت همانطور که به صورت متعارف در متون و گفتمان امنیت رایانه‌ای مطرح می‌شود، مد نظر است که موضوع بخش ششم مقاله است و بصورت مفصل مورد بحث و معرفی قرار گرفته است.

در دیدگاه سوم یعنی دیدگاه چارچوب، به‌کمک دیدگاه دوم مجموعه‌ای همبسته از الگوهای امنیت طوری در کنار هم قرار می‌گیرند که با بکارگیری توأمان آنها می‌توان سیستم امن را ایجاد نمود، به عبارتی با یک چارچوب مواجه هستیم. این دیدگاه موید نگاه کاربردی است برای بکارگیری مجموعه‌ای از الگوها در ایجاد یک سیستم.

بخش ششم مقاله همانطور که پیشتر هم توضیح داده شد به تفصیل مروری بر انواع الگوهای امنیت نموده و سعی در آشنا نمودن مخاطب با دنیای این الگوها در حد شناسایی عنوانی و آدرسی است. بدیهی است فهم دقیق آنها باید با مراجعه به متن تخصصی مربوطه که آدرس‌دهی شده است، صورت پذیرد. در نهایت بخش هفتم مقاله پیشنهاد مولفان را از یک منظر جدید مطرح نموده و مسیر جدیدی را برای دسته بندی الگوها، تعریف آنها و کاربست آسان آنها که مبتنی بر هستان‌شناسی است، ارائه می‌نماید.

---

[1] Survey

[۲] برای اختصار از واژهٔ «امنیت» استفاده می‌کنیم.

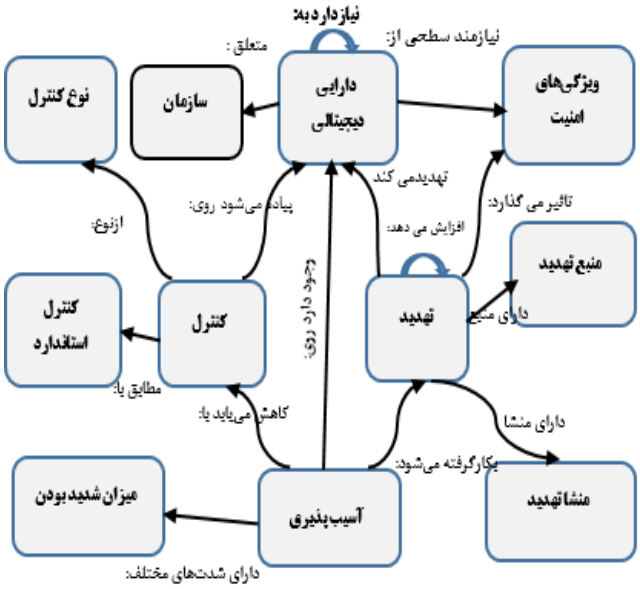
[3] Security pattern

4 GOF: Gangs of Four(It got nicknamed as Gangs of Four design patterns because of four authors: Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides)

۲- تشریح اجمالی فضای مواجهات امنیت

قبل‌از ورود به بحث الگوهای امنیت، ابتدا به‌طور خلاصه، فضای مسئلهٔ امنیت تشریح می‌گردد. اگر بخواهیم خیلی سریع و به‌طور اجمال، مسئلهٔ امنیت را تشریح نماییم، بهترین امکان در این خصوص بهره‌گیری از یک هستان‌شناسی سطح بالا است. مقالهٔ [۶] هستان‌شناسی سطح بالایی از امنیت رایانه‌ای را ارائه داده‌است که در شکل ۱ مشاهده می‌شود. به کمک این هستان‌شناسی می‌توان فضای مواجهات امنیت را در سطح کلان تبیین نمود.

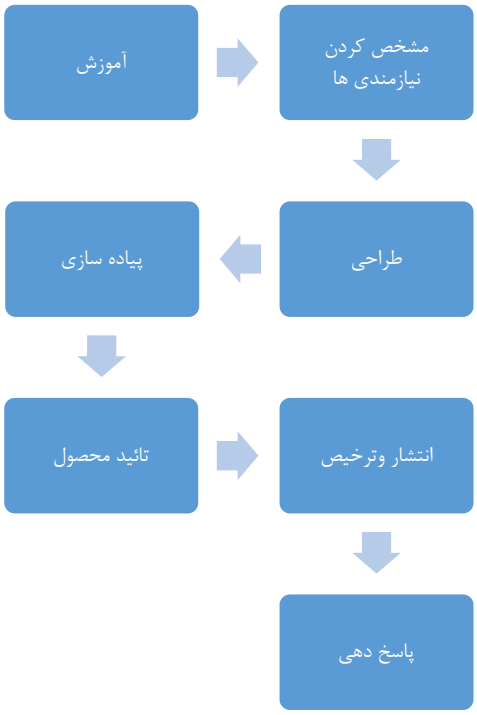
مطابق با تفسیر ارائه‌شده در[۶]، تهدیدات همواره دارائی‌های سازمان را هدف قرار می‌دهند و با بهره‌برداری از آسیب‌پذیری‌های موجود روی دارائی‌ها، موجب تأثیر روی ویژگی‌های امنیت در وجوه مختلف محرمانگی، صحت و دسترس‌پذیری می‌شوند (گوشه بالایی سمت راست شکل ۱). آسیب‌پذیری‌ها (فیزیکی، مدیریتی، فنی) که می‌توانند از شدت‌های مختلفی برخوردار باشند (بخش پایینی شکل ۱)، بوسیلهٔ اِعمال انواع (پیش‌گیرانه، بازدارنده، بازیابنده و آشکارساز) کنترل‌های مبتنی بر استانداردهای امنیت کاهش می‌یابند[۶]. تهدیدات نیز می‌توانند دارای منابع عمدی یا سهوی باشند و منشاءهای مختلف انسانی یا طبیعی داشته باشند(گوشه پایینی و وسطی سمت راست شکل ۱). مرتبط با الگوهای امنیت در مقالهٔ[۷] نیز مفاهیم امنیت به صورت مجزا مورد بررسی قرار گرفته‌اند که برای درک مفاهیم این زمینه می‌تواند مورد استفاده قرار گیرد.



شکل ۱. هستان‌شناسی سطح کلان امنیت[۶]

۳- تبیینی بدیع از ضرورت و چرایی توجه به الگوهای امنیت

امنیت در همهٔ حوزه‌های فناوری اطلاعات و ارتباطات به مقوله‌ای مهم و حیاتی تبدیل شده است و اعتماد به سامانه‌ها و نرم‌افزارها و حتی اعتبار سازمان‌ها شدیداً به موضوع امنیت گره خورده‌است. تلاش‌های فراوانی برای توسعهٔ دانش امنیت و تسهیل بکارگیری فناوری‌های مرتبط با آن در سامانه‌ها انجام می‌گیرد. بررسی گزارشات مرتبط با تعداد حملات و نیز آسیب‌پذیری‌های کشف‌شده درانواع سامانه‌ها [۸و۹] و روند روبه‌افزایش آن، حاکی است که هنوز بلوغ لازم در عوامل و ذی‌نفعان مختلف در کشف و اِعمال نیازمندی امنیت در سامانه‌ها بخصوص سامانه‌های نرم‌افزاری بوجود نیامده است. توسعه‌دهندگان و ایجادکنندگان نرم‌افزارها، دانش و خبرگی لازم در این زمینه را ندارند و در بسیاری موارد نرم‌افزارها بدون ملاحظات امنیتی توسعه می‌یابند. سؤال اساسی این است که چگونه می‌توان خلاء ناشی از فقدان دانش امنیت در توسعه‌دهندگان و ایجادکنندگان نرم‌افزارها و سامانه‌ها را برطرف نمود؟ شرکت مایکروسافت در چرخهٔ عمر تعریف‌شده برای توسعه نرم‌افزارهای امن، مرحلهٔ آموزش را در ابتدای کار قرار داده [۱۰] که در شکل ۲ نشان داده شده است. این روش توسط اُواسپ[1] نیز به‌طور مشابه بکار گرفته می‌شود[۱۱].



شکل ۲. مراحل چرخه عمر ایجاد نرم‌افزار امن[۱۰]

[1] OWASP

در این مرحله موارد بنیادی برای ساخت نرم‌افزار مطلوب، شامل طراحی امن، مدل‌سازی تهدید، کُدنویسی امن، آزمون امنیت و بهترین تجربیات مرتبط با حفظ حریم خصوصی آموزش داده می‌شود[۱۰].

به نظر می‌رسد این توصیه بیشتر برای نوشتن نرم‌افزارهای امن (نرم‌افزارهایی که هنگام مصالحه، وزن امنیت بیشتر است) مناسب است، یعنی در مواقعی که هزینه-فایده و مصالحهٔ[1] مرتبط با این زمینه به نفع امنیت است، ناچار برای گروه توسعه نرم‌افزار این مرحله باید انجام شود. اما آیا فرایند تولید همهٔ نرم‌افزارها و سامانه‌ها از فرایند تولید نرم‌افزار امن تبعیت می‌کنند؟ پاسخ این پرسش قطعاً مثبت نیست، زیرا از نظر عقلانیت[2] اقتصادی در همه‌جا این فرصت بدست نمی‌آید. لذا باید به دنبال راه‌حل دیگری برای این موضوع بود. روش دیگری که بعضاً برای این موضوع بکار گرفته شده است، استفاده از خطوط راهنما و توصیه‌ها است که می‌توان به NIST-SP 800-64 اشاره نمود [۱۲]. البته کل فرایند توصیه شده توسط شرکت مایکروسافت [۱۱] را نیز می‌توان در این زمره قرار داد.

در مستندNIST، مرتبط با هر مرحلهٔ چرخهٔ عمر نرم‌افزار، تعدادی فعالیت معرفی شده‌است که برای هرکدام توصیه شده‌است که از کدام راهنما و استاندارد استفاده گردد. برای مثال درخصوص طبقه‌بندی امنیت، FIPS 199 را معرفی نموده است. در FIPS199 طبقه‌بندی به‌صورت مجموعه‌ای از دوتایی‌های «خصوصیت» و «پیامد» تعریف شده است که خصوصیت از مجموعهٔ (محرمانگی، صحت و دسترسی‌پذیری) انتخاب می‌شود و پیامدها از مجموعهٔ (کم، متوسط، بالا و غیر قابل کاربرد) بدست می‌آید [۱۳]. در جدول ۱ خلاصه فعالیت‌های مذکور به همراه نمونه‌هایی از راهنماها و توصیه‌ها ارایه شده‌است.

در این روش متناسب‌با فعالیت‌های مراحل چرخهٔ عمر ایجاد نرم‌افزار (SDLC[3]) ، اقدامات اصلی مرتبط با امنیت معرفی شده‌اند، ولی در روش معرفی‌شده از سوی مایکروسافت [۱۱]، چرخهٔعمر مورد نظر، مختص ایجاد نرم‌افزار امن (SDL[4]) است. اگر بخواهیم نرم‌افزار امن تولید کنیم، دغدغهٔ فقدان دانش امنیت، کمتر است اما اگر هدف، تولید نرم‌افزار و سامانه متعارف باشد و قصد داشته باشیم ملاحظات امنیت را رعایت‌کنیم، در این صورت مشکل فقدان دانش و متخصص امنیت، خود را شدیداً نمایان می‌سازد.

جدول ۱. فعالیت‌های چرخهٔ عمر نرم‌افزار و اسناد راهنمای امنیت متناسب با آن[۱۳]

مرحله	ملاحظات امنیتی	استانداردها و راهنماهای کمکی
شروع	طبقه‌بندی امنیت	FIPS PUB 199
	ارزیابی مقدماتی ریسک	NIST SP 800-30 , NIST SP 800-53
مرحله شناخت و طراحی	ارزیابی ریسک	NIST SP 800-30 , NIST SP 800-53
	تحلیل نیازمندی‌های وظیفه‌ای امنیت	Privacy Act, FISMA, OMB circulars, agency enabling acts, NIST Special Publications and FIPS
	تحلیل نیازمندی‌های اطمینان امنیت	Common Criteria(CC)
	ملاحظات هزینه‌ای و گزارش‌دهی	OMB Circular A-11, Part 3, Planning, Budgeting, and Acquisition of Capital Assets,"
	برنامه‌ریزی امنیت	NIST Special Publication 800-18
	ایجاد کنترل امنیت	NIST Special Publication 800-53
	ارزشیابی و آزمون امنیت از منظر ایجاد	
	دیگر مؤلفه‌های برنامه‌ریزی	
مرحله پیاده سازی	شروع و پذیرش	
	یکپارچه‌سازی کنترل امنیت	
	گواهی امنیت	NIST SP 800-37
	مجوز امنیت	NIST SP 800-37
مرحله عملیات/نگهداری	مدیریت پیکربندی و کنترل	
	مانیتورینگ بلانقطاع	NIST Special Publication 800-53A
مرحله عرضه و استقرار	حفظ اطلاعات	
	محو اطلاعات از روی رسانه	
	در اختیار قراردادن سخت‌افزار و نرم‌افزار	

[1] Trade-offs
[2] Rationality
[3] Software Development Life Cycle
[4] Security Development Lifecycle

با نگاه به جدول ۱ و انواع ارجاعات به راهنماها و مستندات مختلف، می‌توان فهمید که چه دانشی و نیز چه تلاشی برای مراقبت و بکارگیری این موارد مورد نیاز است. به‌علاوه در استفاده از خطوط راهنما و توصیه‌ها مشکل جدی عدم قابلیت استفاده مجدد[1] مطرح است و نمی‌توان قواعدی که به زبان غیررسمی طبیعی نوشته‌شده را برای حل مشکلات برنامه‌نویسی امن که مشترک بین همه است، بکار گرفت [۷]. از دیدگاه مطرح شده در [۱۴] که بر اساس آن شناسایی و رفع آسیب‌پذیری‌های نرم‌افزار نباید به بعد از اتمام مراحل تولید و توسعه واگذار شود و توصیه‌های جلوگیری از ایجاد آسیب‌پذیری در همه مراحل توسعه را ارائه می‌نماید، هزینهٔ رفع آسیب‌پذیری‌های سیستم و ریسک مرتبط با آنها، بعد از استقرار سیستم، هم برای توسعه‌دهندگان سیستم (برنامه‌نویسان) و هم برای استفاده‌کنندگان بالا است.

در تأیید مطالب پیش‌گفته با استناد به [۱۵]، به‌طور خلاصه می‌توان گفت که روش‌های مواجههٔ نظام‌مند مانند خط‌مشی امنیتی، معیارهای مشترک، نمایش‌های درختی، روش‌های صوری[2] و روش‌های نیمه‌صوری[3] برای همیشه بکار گرفته نمی‌شوند، زیرا دغدغهٔ امنیت برای توسعه‌دهندگان آنقدر برجسته نمی‌شود که از این روش‌ها استفاده نمایند یا لااقل در نرم‌افزارهای متعارف فرصت نمی‌یابند از این روش‌های نظام‌مند استفاده‌کنند. این نحوهٔ استدلال در بیان چرایی استفاده از الگوها برای اولین بار در اینجا ارائه شده است.

۴- تاریخچه و مزایای بکارگیری

از آنجایی که امنیت در چرخهٔحیات نرم‌افزار به صورت یک مقولهٔ اضافی[4] دیده شده و توجه کافی به آن صورت نمی‌گیرد، فعالان زمینهٔ الگوها به این مشکل پی‌برده و در صدد ثبت تجارب افراد خبره در این خصوص برآمده‌اند [۱۵]. اولین تلاش دراین زمینه در سال ۱۹۹۷ توسط یادر و باراکلو با انتشار مقاله‌ای درباب الگوهای امنیت صورت‌پذیرفت[۱۴]. پیش از این‌ها، تلاش‌ها عمدتاً صرف بیان الگوها، به‌طور عام شده‌است که اولین بار کریستوفر الکساندر مفهوم الگوهای طراحی را مرتبط با حوزه ساختمان مطرح ساخت [۱۴]. بک و کانینگهام بکارگیری الگوها را در برنامه‌نویسی تجربه‌کرده و در اوپ‌اسلا[5] ارائه نمودند [۱۴]. کتاب منتشر شده توسط گنگ‌اُوفور در سال 1995 تلاش مهم بعدی است [۱۴]. آنها به شکل ساختارمندی تجارب افراد خبره را اخذ و به‌صورت سه‌تایی (زمینه، مشکل، راه‌حل) ارائه کردند [۱۴]. درالگوهای امنیت ارائه‌شده توسط افراد مختلف ازروش گنگ‌اوفور با تمرکز بر مشکل امنیت اقدام شده‌است. الگوهای امنیت بسته‌های با قابلیت استفاده مجدد هستند که دانش خبره را با خود به‌همراه دارند[۷]. الگوها در همهٔ مراحل باید علاوه بر خطوط راهنما، مورد اهتمام قرار گرفته و ترویج گردند[۱۴]. با استفاده از الگوهای امنیتی با قابلیت استفاده مجدد، تولیدکنندگان می‌توانند هزینهٔ تولید محصولات امن را هم‌زمان برای استفاده‌کنندگان و پیاده‌سازان پایین‌آورند[۱۴]. الگوهای امنیت مزایای‌زیر را دربردارند[۱۵]:

- افراد تازه‌کار و ناوارد در امنیت می‌توانند مثل خبرگان در مواجهه با مشکل امنیت عمل کنند؛
- کارشناسان امنیت می‌توانند هم مشکل و هم راه‌حل را به‌طور مؤثرتر مورد بحث قرار دهند یا آنها را مشخص نموده و نام‌گذاری کنند؛
- مشکلات به صورت نظام‌مندتری حل می‌شوند؛
- مشکل وابستگی بین مؤلفه‌ها می‌تواند به‌طور مناسب‌تری مشخص و درنظر گرفته‌شوند.

به علاوه، با استفاده از بکارگیری الگوهای امنیت، نقش عوامل انسانی را در مهندسی امنیت کاهش می‌دهیم. مهمترین دلائل این‌گفته از نظر[۱۵] عبارتنداز:

- مهندسی امنیت عموماً توسط خبره‌های امنیت انجام نمی‌شود، بلکه نیاز به امنیت توسط افراد مطرح شده و پاسخ داده می‌شود، یعنی با مسئله امنیت به صورت یک امر اضافی برخورد می‌شود، زیرا عموماً دغدغهٔ اصلی ایجادکنندگان نرم‌افزار، وظیفه‌مندی سامانه است نه امنیت؛
- عموماً افراد به صورت اقتضایی[6] با حل مسئله امنیت برخورد می‌کنند. مثلاً برای امنیت ورود به سامانه‌ها، افراد را مجبور می‌کنند کلمهٔ عبورقوی تعیین‌کنند، ولی کلمهٔ عبورقوی، زود فراموش شده و افراد برای اجتناب از مشکل فراموشی، کلمه عبور را در جاهای ناامن یادداشت می‌کنند؛
- راه‌حل‌ها عموماً مبتنی بر زمان هستند. یک راه‌حل فعلی ممکن است برای زمان‌های آتی اعتبار نداشته باشد؛
- اگر همهٔ وابستگی‌ها هم مشخص شوند، باز هم نمی‌شود همه را به‌طور مناسب در نظر گرفت. مثلاً یک کارپذیر وب را می‌توان بررسی کرد که امن است یا خیر، ولی نمی‌توان مطمئن

[1] Reuse
[2] Formal Method
[3] Semi Formal Method
[4] Add-on
[5] OOPSLA (Object-Oriented Programming, Systems, Languages & Applications) is an annual ACM research conference.
[6] Ad-hoc

شد که سایر ارتباطات با سیستم عامل و مؤلفه‌های دیگر مثل سخت‌افزار امن هستند یا خیر.

۵- بررسی الگوهای امنیتی و دیدگاه‌های غالب

در بررسی‌های به‌عمل‌آمده می‌توان سه دیدگاه متفاوت در زمینه الگوها را شناسائی و ارائه نمود:

- الگوهای امن[1]
- الگوهای امنیت[2]
- دیدگاه سیستم و چارچوب به الگوهای امنیت

این دسته‌بندی (سه دیدگاه)، اولین بار در این نوشتار ارائه شده است. در زیربخش‌های این قسمت، هرکدام از دیدگاه‌ها با شرح بیشتری مورد بررسی قرارگرفته‌اند.

۵-۱- دیدگاه اول، الگوهای امن

مطابق با [۱۴] روش یافتن الگوهای امن، جستجو در الگوهای مهندسی نرم‌افزار مانند گنگ‌اُوفور و استخراج الگوهای امن از متن الگوهایی است که دارای تأثیر امنیتی هستند و سپس مستندکردن آنها برای استفاده در مسائل تکراری مرتبط با ارتقاء امنیت است. از این نقطه‌نظر سه‌دسته الگو مطرح‌شده‌اند. (مطالبی که در ادامه آمده- است شامل بخش الف، ب و ج از منبع[۱۴] است):

۵-۱-۱- الگوهای سطح معماری

۵-۱-۱-۱- الگوی تجزیه بی‌اعتماد(کاهش امتیاز)[3]

بسیاری از حملات، ناشی از دسترسی سطح بالا به سیستم می‌باشد که از طریق یک حساب‌کاربری مثل ریشه در یونیکس یا مدیر در ویندوز انجام می‌شود. با استفاده ازاین الگو، کاربر به کل سامانه دسترسی نخواهد داشت و دسترسی به کل سامانه محدود به بخش کوچک‌تری می‌شود. این کار از طریق منتقل کردن وظایف مجزا به داخل برنامه‌هایی که دارای عدم اعتماد متقابل نسبت به هم هستند، صورت می‌گیرد. ساختار کلی این الگو بصورتی است که سیستم را به دو یا چندین برنامه تقسیم می‌کند، به‌طوریکه هر کدام پردازش‌های جداگانه‌ای را انجام می‌دهند. در حقیقت هر پردازش سطح دسترسی و حق امتیاز مربوط به خود را دارا بوده و یک زیرمجموعه‌ای از وظایف سامانه را انجام می‌دهد. موجودیت‌هایی که این الگو از آنها تشکیل شده است، عبارتند از:

- تعدادی از برنامه‌های مستقل و جداگانه از یکدیگر که هرکدام پردازش‌های مستقل خود را دارا می‌باشد و برای هریک، شناسه کاربری‌هایی تعریف و سطح دسترسی و حق امتیاز هر کدام مستقل از دیگری است؛
- ارتباط شبکه‌ای کاربرمحلی و کاربر راه‌دور؛
- یک مکانیزم ارتباطی بین پردازش‌ها مانند SOAP، RPC و سوکت دامنه یونیکس.

این الگو باعث می‌شود که دسترسی نفوذگران به کل سامانه محدود شود، مخصوصاً در زمان‌هایی که به یک عنصر از سامانه توسط هکر دسترسی پیدا شود که در این صورت بقیهٔ سامانه از نفوذ این نفوذگر در امان خواهد بود.

۵-۱-۱-۲- الگوی جداسازی مجوز[4]

هدف این الگو کاهش میزان کُدهایی است که با دسترسی مشخصی اجرا می‌شوند و این سطح دسترسی‌ها هیچگونه تأثیر یا محدودیتی در کاربرد برنامه ایجاد نمی‌کنند. درحقیقت این الگو یک نمونه ویژه و خاص‌تری از الگوی تجزیه بی‌اعتماد[5] می‌باشد. رخنه امنیتی در حالتی که از این الگو استفاده نشود زمانی رخ می‌دهد که یک کارپذیر از طریق سیستم احراز هویت و سیستم کاربری غیرقابل‌اطمینانی که دارد، تلاش می‌کند بوسیله یک پردازش فرزند که همان امتیاز کارپذیر را دارا است، کاربران را احراز هویت کند. در این زمان یک‌کاربر با نیت و قصد سوء سعی می‌کند با سوءاستفاده از این آسیب‌پذیری و با حق‌دسترسی، کنترل پردازش فرزند را در دست گیرد و اعتبار حق‌دسترسی خود را بر روی کارپذیر افزایش دهد. به‌طورکلی این الگو برای سیستم‌هایی با شرایط و کاربردهای ذیل مناسب‌است:

۱. به یک سطح دسترسی بالا نیاز نداشته باشد.
۲. ارتباط مهم و اساسی با منابع غیر قابل اطمینان داشته باشد.

۵-۱-۱-۳- الگوی موکول به کرنل[6]

هدف این الگو جداسازی شفاف آن قسمت از وظایف سیستم است که حق‌دسترسی بالا داشته و باید از قسمت‌های کاربردی دیگری که نیازمند حق‌دسترسی بالا نمی‌باشند، جداشده و همچنین امکان بهره بردن از تأیید کاربر که در هستهٔ سیستم موجود است، بوجود آید. انگیزهٔ اصلی استفاده از این الگو کاهش یا اجتناب از آن دسته از

[1] Secure Patterns
[2] Security Patterns
[3] Distrustful Decomposition (Privilege reduction)
[4] PrivSep (Privilege Separation[
[5] Distrustful Decomposition
[6] Defer to Kernel

برنامه‌های کاربردی است که اجرای آنها با افزایش سطح دسترسی همراه است و ممکن است به‌طور بالقوه، احتمال حملهٔ افزایش امتیاز را بالا ببرد. در سیستم‌های یونیکس این به معنی کاهش یا اجتناب از برنامهٔ Setuid می‌باشد و در سیستم عامل ویندوز نیز به معنای اجتناب از اجرای برنامه، تحت‌کاربری مدیر می‌باشد. به‌علاوه تمرکز این الگو بر تأیید کاربر که توسط هسته سیستم عامل مهیا می‌شود، قرار دارد. استفادهٔ مجدد از کارکرد و وظیفه هستهٔ سیستم در تأیید کاربر، فواید و مزایای ذیل را دربر دارد:

- توسعه‌دهندگان مجبور به نوشتن برنامه‌ای برای احراز هویت کاربران و تأیید آنها نیستند.
- این راه‌حل یک راه‌حل قابل انتقال می‌باشد، زیرا به هر سیستم عاملی اجازه می‌دهد که کاربر را به روش ثابت و با هر پلتفرمی تأیید کند.

مباحث ارائه شده در مقاله [۱۶] در این زمینه اطلاعات مفید و جدیدی را به خواننده ارائه می‌کند.

۵-۱-۲- الگوهای سطح طراحی

۵-۱-۲-۱- الگوی کارخانه امن[1]

هدف این الگو جدا ساختن منطق وابسته به امنیت از وظایف اساسی و اصلی مرتبط با ایجاد یا انتخاب یک شئ است. این الگو شکل توسعه‌داده‌شدهٔ الگوی متد سازنده مجرد گنگ‌اوفور است که جنبه‌های امنیت را مورد توجه ویژه قرار داده‌است. یکی از انگیزه‌های اصلی این الگو، کاهش چسبندگی بین وظایف امنیتی یک شئ و دیگر وظایف آن است.

۵-۱-۲-۲- الگوی کارخانه راهبرد امن[2]

هدف این الگو ارائه روش اصلاح و استفاده آسان در انتخاب شئ استراتژی مناسب برای انجام یک تکلیف براساس اختیارات و اعتبار امنیتی یک کاربر یا محیط است. این الگو بر اساس الگوی استراتژی گنگ‌اوفور و نیز الگوی سازنده امن (الگوی قبلی) ساخته شده است.

۵-۱-۲-۳- الگوی کارخانه سازنده امن[3]

هدف این الگو جداسازی قواعد وابسته به امنیت از فرایند پایه‌ای ایجاد شئ پیچیده و سنگین مطابق با الگوی سازنده در گنگ‌اوفور است. این کار باعث کاهش چسبندگی وظایف امنیتی از وظایف دیگر شئ پیچیده می‌شود.

۵-۱-۲-۴- الگوی زنجیره امن مسئولیت[4]

مطابق گنگ‌اوفور، الگوی زنجیرهٔ مسئولیت، الگویی است که شامل فهرست پیوندی از نگهدارنده‌ها است که هرکدام قادر به پردازش درخواست‌ها می‌باشند. هنگامی که درخواستی به زنجیره اضافه می‌شود، به اولین نگهدارنده‌ای که قادر به پردازش آن می‌باشد، منتقل می‌گردد. در الگوی حاضر که مبتنی بر گنگ‌اوفور است، آنچه را که به جنبه‌های مرتبط با اعتماد به کاربر یا محیط مربوط است، از وظیفه‌مندی اصلی جدا شده و چسبندگی بین وظایف اعتماد و امنیت از دیگر وظایف الگو حذف می‌گردد.

۵-۱-۲-۵- الگوی ماشین حالت امن[5]

این الگو نیز بسط‌یافتهٔ الگوی State ارائه‌شده در گنگ‌اوفور است و نام دیگر آن حالت امن است. در مدل حالت امن ماشین، وضعیت ماشین به منظور بررسی امنیت سیستم، ثبت می‌شود. وضعیت ارائه داده‌شده، متشکل از تمام مجوزهای فعلی و تمام موارد حال‌حاضر می‌باشد. اگر مورد مدنظر، بتواند به مقصد خود فقط از طریق سیاست‌های امنیتی دست یابد، سیستم امن‌است. هدف این الگو این است که اجازه می‌دهد یک جداسازی واضح بین مکانیسم‌های امنیتی و قابلیت‌های سطح کاربر بوسیله پیاده‌سازی وظایف امنیتی و وظایف سطح‌کاربر به‌صورت دو حالت مجزا صورت پذیرد.

۵-۱-۲-۶- الگوی بازدیدکننده امن[6]

این الگو به طراحان ابزاری ارائه می‌دهد تا بتوانند دسترسی کاربرانی که امتیاز دسترسی مجاز و مناسب به اشیاء ندارند را منتفی نمایند. زمانی که از این الگو استفاده می‌کنیم، سلسله مراتب داده‌ها، برای ساختار اطلاعات در گره‌های[7] متفاوت استفاده می‌شود. این بدان معنی است که تمام گره‌ها برای تمام ناظران به‌جز کسانی که اعتبارنامهٔ امنیتی مناسب را دارا هستند، قفل می‌باشد. نتیجهٔ این کار جدایی واضح منطق امنیتی، از عملکرد کاربر است. در این الگو، مشاهده‌گر امن باید از سیستمی استفاده کند که اطلاعات، به صورت سلسله مراتب داده‌ای مرتب شده باشند. این الگو همچنین محدودیت‌های دسترسی مختلف را بر روی هر گره ایجاد می‌کند.

۵-۱-۳- الگوهای سطح پیاده‌سازی

۵-۱-۳-۱- الگوی ثبت‌کننده امن[8]

[1] Secure Factory
[2] Secure Strategy Factory
3 Secure Builder Factory
[4] Secure Chain of Responsibility
[5] Secure State Machine
[6] Secure Visitor
[7] Node
[8] Secure Logger

هدف این الگو جلوگیری از جمع‌آوری اطلاعات مفید توسط مهاجم از طریق لاگ‌های سیستمی و نیز جلوگیری از ویرایش لاگ‌های سیستمی توسط مهاجم برای پنهان نمودن کارهای مخرب صورت داده شده توسط وی است. این الگو در لاگ‌فایل‌ها یا قالب‌های دیگری از نگهداری لاگ‌ها مورد استفاده قرارمی‌گیرد و نیز برای سدنمودن مهاجم جهت ورود به سیستم و پیداکردن و تشخیص حملات روی سیستم بکار گرفته می‌شود. نتایج بکارگیری این الگو عبارتنداز:

- دسترسی مهاجمان محدود می‌شود یا اصلاً متن واقعی لاگ‌ها را نمی‌توانند ببینند، که در این صورت نمی‌توانند حملات ماهرانه‌ای را انجام‌دهند.
- تغییراتی که مهاجم در لاگ‌ها ایجاد می‌کند، توسط کاربر مجاز قابل یافتن است.

۵-۱-۳-۲- الگوی پاک‌کردن اطلاعات حساس[1]

اگر اطلاعات حساس قبل‌از خالی کردن منابع قابل استفادهٔ مجدد، مشخص نشوند، ممکن است این اطلاعات که در منابع با قابلیت استفادهٔ مجدد ذخیره شده‌اند، به‌وسیلهٔ دسترسی غیرمجاز مورد استفاده مهاجمان قرار بگیرند. کاربرد این الگو باعث می‌شود که قبل‌از اینکه منابع دوباره مورداستفاده قرار بگیرند، اطلاعات‌حساس مشخص‌گردد. منابع با قابلیت‌استفادهٔ مجدد، شامل حافظهٔ به‌صورت پویا اختصاص داده شده، حافظهٔ ایستای اختصاص‌یافته، حافظهٔ خودکار اختصاص‌یافته، مخزن حافظه، دیسک و مخزن دیسک می‌باشد. زیرا اطلاعاتی که در منابع وجوددارد، زمانی علامت‌گذاری می‌شود که منبع مورد نظر قابل استفادهٔمجدد است، محتوای فعلی منبع دست نخورده باقی می‌ماند تا روی آن اطلاعات جدید نوشته شود که در طول این مدت کاربر غیرمجاز می‌تواند به آن دسترسی داشته‌باشد. اگر برنامه، اطلاعات حساس را در منابع با قابلیت استفادهٔمجدد ذخیره‌کرده باشد، این الگو کاربرد دارد. نتیجهٔ استفاده از این الگو آن است که اگر دسترسی غیرمجاز به منابع هم پیدا شود، باز هم نفوذگر قادر به خواندن اطلاعات حساس نخواهد بود.

۵-۱-۳-۳- الگوی پوشه امن[2]

هدف این الگو آن است که اطمینان دهد، مهاجم نمی‌تواند فیلدهای مورد استفاده یک برنامهٔ درحال اجرا را دستکاری‌کند. هر برنامه‌ای که اجرا می‌شود ممکن‌است به برنامه‌های دیگری وابسته باشد. یک توسعه‌دهندهٔ نرم‌افزار فرض را بر این می‌گذارد که برنامه توسط کاربران خارجی در حین اجرا دستکاری نمی‌شود. حال اگر این فرض نقض شود، فایل می‌تواند توسط کاربران متفاوتی تغییر پیدا کند یا فایل در یک زمان بسیار حساس، پاک شود یا تغییر یابد. این الگو اطمینان می‌دهد که پوشه‌هایی که فایل‌های آن، توسط برنامه‌ها استفاده می‌شود، فقط می‌توانند توسط کاربر معتبر برنامه، نوشته یا خوانده شوند. کاربردهای مهم این الگو را می‌توان به صورت زیر فهرست نمود:

- برنامه ممکن است در محیطی ناامن اجرا شود، محیطی که کاربران بدخواه می‌توانند به آن‌برنامه و فایل سیستم‌هایی که توسط برنامه‌ها استفاده می‌شود، دسترسی داشته باشند.
- برنامه، فایل‌ها را می‌خواند یا می‌نویسد.
- اجرای برنامه می‌تواند به صورت منفی صورت گیرد، یعنی اگر فایل‌ها توسط برنامه‌ای که بوسیلهٔ کاربر خارجی در حین اجرای برنامه تغییر یافته، خوانده یا نوشته شوند.

۵-۱-۳-۴- الگوی متعارف‌سازی نام‌مسیر[3]

هدف از ایجاد این الگو اطمینان از این است که تمام ارتباطات یک برنامه (شامل ارتباط میان ماژول‌ها و کلاس‌های یک برنامه] از طریق لینک‌های معتبری که فاقد هرگونه علامت یا میانبر است صورت پذیرد. از آنجایی‌که استفاده از لینک‌های کمکی و میانبر، ممکن است معنای‌یک برنامه را از بین ببرد، لذا پیشنهاد می‌گردد برای هرگونه ارتباط داخلی در یک برنامه از یک لینک و مسیر معتبر و منحصر بفرد استفاده شود. این الگو قادر است در موارد بسیاری استفاده گردد که از جمله آنها می‌توان به موارد زیر اشاره‌کرد:

- پذیرش برنامه از منابع غیرقابل اطمینان
- جلوگیری از دسترسی‌های غیرمجاز توسط هکرها به‌وسیله ایجاد مسیرهای جعلی
- اجرای برنامه در یک محیط که در آن هرفایل، یک مسیر منحصربفرد و معتبر دارد.

استفاده از این الگو در برنامه‌نویسی باعث بهبود دقت و امنیت دسترسی به فایل‌ها می‌گردد. از طرفی استفاده از لینک‌های معتبر و منحصر بفرد باعث افزایش سرعت و کاهش سربار در برنامه می‌شود.

۵-۱-۳-۵- الگوی تائید ورودی[4]

به کمک این الگو می‌توان از بسیاری از آسیب‌پذیری‌هایی که بوسیله

[1] Clear Sensitive Information
[2] Secure Directory
[3] Pathname Canonicalization
[4] Input Validation

سنجش داده‌های ورودی صورت می‌گیرد، جلوگیری کرد. اعتبارسنجی ورودی، مستلزم آن است که یک توسعه‌دهنده، به‌درستی تمام ورودی‌های خارجی را از منابع داده‌ای غیرقابل اطمینان، شناسایی و اعتبارسنجی کند. اعتبارسنجی ورودی، آزمون صحت هر ورودی است که در آن ممکن است مواردی غیر از آنچه صحیح است، وارد شود. ورودی کاربر می‌تواند از انواع منابع، کاربر نهایی، برنامه دیگر، یک کاربر مخرب، یا هرتعداد از منابع دیگر آمده باشد. کاربر مخرب هیچگاه اعلام نمی‌کند که او به عنوان مهاجم نرم‌افزار شما است. به همین دلیل همه ورودی‌ها باید بررسی و اعتبارسنجی شوند. برای اینکه دقیقاً مشخص شود چه کسی یا چه چیز ورودی را برای پردازش به شما می‌دهد، برنامه‌های کاربردی و نرم‌افزار باید تمام ورودی وارد شده توسط کاربر را بررسی کند، اما این نباید تنها ورودی برای بررسی باشد. ممکن است شما ورودی را از یک بانک داده‌ای گرفته باشید. مشکلات ناشی از اعتبارسنجی ورودی نادرست، می‌تواند به تمام انواع مشکلات و آسیب‌پذیری‌ها منجر شود. عدم استفاده از اعتبارسنجی ورودی کاربر توسط یک برنامه، علت ریشه‌ای بسیاری ازسوء استفاده‌های امنیتی جدی مانند حملات سرریز بافر، حملات تزریق SQL و حملات CSS است. اعتبارسنجی ورودی برای امنیت نرم‌افزار، امری حیاتی است. الگوهای اعتبارسنجی ورودی، اغلب ساده هستند و نیازمند شناسایی و سنجش اعتبار صحت می‌باشند.

مزایای اعتبارسنجی ورودی‌های سیستم، باعث افزایش امنیت و قابلیت اطمینان سیستم شده و در عوض باعث کاهش کارایی سیستم می‌شود، زیرا برای تعیین‌کردن و مدیریت‌کردن تمام مکان‌هایی که ورودی نامعتبر ایجادمی‌شود، نیازمند کار اضافه می‌باشد.

۵-۱-۳-۶- الگوی مقداردهی اکتساب منابع[1]

هدف از الگوی RAII این است که اطمینان حاصل شود که منابع سیستم به درستی اختصاص داده شده یا آزاد می‌شوند و تمام مسیرهای اجرای برنامه امکان‌پذیر است. همچنین تضمین می‌کند که منابع برنامه به درستی به‌کار گرفته شده‌اند. به‌طور معمول هرمنبعی که در سیستم استفاده می‌شود باید به‌موقع نیز آزاد شود. این موضوع برای جلوگیری از اتلاف منابع می‌باشد. جلوگیری از آزادشدن منابع در زمان استفاده برنامه نیز بسیار مهم است، زیرا عدم رعایت این مورد می‌تواند نتایج بغرنجی را به همراه داشته باشد. همچنین تعیین زمانی که دیگر سیستم به منابع احتیاجی ندارد و بایستی آزاد شوند نیز بسیار دشوار می‌باشد، بنابراین طول مدت زمان استفاده از منابع بایستی در طراحی اصلی برنامه دیده شده باشد. یک نمونه از استفاده از الگوی RAII برنامه‌ای است که حافظه یا منابع را در هنگام اجرا اختصاص می‌دهد و قبل از اتمام برنامه، منابع را آزاد می‌نماید. مثال دیگر استفاده از الگوی RAII تشکیل یک اتصال شبکه‌ای در هنگام شروع برنامه و قطع اتصال در هنگام بسته شدن تابع یا برنامه می‌باشد. الگوی RAII برای هر سیستمی که از منابع استفاده می‌کند و لازم است به‌صورت پی‌درپی، منبع تخصیص داده و سپس آزاد کند، کاربرد دارد. این منابع شامل بخش‌هایی از حافظه، فایل‌های باز، منابع شبکه و غیره می‌تواند باشد. همچنین این الگو برای زمانی که منابع سیستم محدود است و هنگام آزادکردن منابع با مشکل مواجه می‌شود و سبب اتلاف منابع یا حتی موجب اخلال در سرویس‌دهی می‌شود، نیز بسیار مفید است.

نکتهٔ مهم: الگوهایی که در بخش ۵-۱ معرفی شدند، صرفاً الگوهای امنیتی نیستند، بلکه الگوهای‌طراحی هستند که از نظر امنیت بهبود داده شده‌اند.

۵-۲- دیدگاه دوم، الگوهای امنیت

در دیدگاه دوم، یعنی الگوهای امنیت، با انواع و اقسام رده‌بندی‌ها و روش‌های طبقه‌بندی الگوها مواجه هستیم که مستقل از الگوهای گنگ‌اوفور و تقسیم‌بندی مطابق با آن هستند. هنگامی که از الگوهای امنیت بحث می‌کنیم، عموماً به این مسئله توجه داریم که فضای مواجههٔ ما مجموعه‌ای از الگوها هستند که قرار است در بارهٔ پیاده‌سازی و استفادهٔ عملی از آنها تلاش‌کنیم. چهار دسته فعالیت‌هایی که می‌توان در حوزهٔ الگوها انجام داد و غالباً با ابزار نیز انجام می‌شود[۱۵] عبارتند از:

- فعالیت‌های نگهداری الگوها شامل ایجاد، ویرایش، نشر و خواندن الگوها
- رده‌بندی شامل طبقه‌بندی‌های مسئله و راه‌حل یا انواع دسته‌بندی‌های دیگر
- استفاده از آنها در مدل‌سازی معماری‌ها و سیستم‌های فناوری اطلاعات
- استنتاج: آیا این الگو مناسب است یا الگوهای جایگزین دیگری داریم؟ از منظر نیازمندی‌های غیروظیفه‌ای، مثلاً کارایی کدام الگو مناسب حل مسئله مورد نظر است؟

در این مقاله و در بخش‌های پیش‌رو (بخش۶) بررسی تفصیلی

[1] Resource Acquisition Initialization

درخصوص دسته‌بندی رده‌بندی الگوها ارائه شده است و به علاوه دسته‌بندی جدیدی نیز در این مقاله مطرح شده‌است.

۵-۳- دیدگاه سوم، دیدگاه سیستم و چارچوب به الگوهای امنیت

در دیدگاه سوم یعنی دیدگاه چارچوب، به‌کمک دیدگاه دوم مجموعه‌ای همبسته از الگوهای امنیت طوری در کنار هم قرار می‌گیرند که با بکارگیری توأمان آنها می‌توان سیستم امن را ایجاد نمود، به عبارتی با یک چارچوب مواجه هستیم. این نوع مواجهه در سند [۱۷] مطرح شده است. از نظرگاه این گروه، الگوهای امنیت در دو بخش قرار می‌گیرند: آنهایی که دسترس‌پذیری سیستم را تأمین می‌کنند و آنهایی که حفاظت از سیستم شامل محرمانگی و صحت را تأمین می‌نمایند. با بکارگیری همهٔ این الگوها به‌صورت توأمان و در قالب یک چارچوب، امنیت سیستم هدف تأمین‌می‌گردد.

۵-۳-۱- گروه اول، الگوهای مرتبط با دسترس‌پذیری

۵-۳-۱-۱- الگوی سیستم ایست بازرسی[1]

هدف این الگو آن‌است که یک سیستم را طوری ساختاردهی کند که حالتش قابل بازیابی و بازگشت به یک حالت صحیح شناخته شده در حالتی که یک مؤلفه دچار عجز شود، باشد.

۵-۳-۱-۲- الگوی آماده به کار[2]

هدف این الگو آن‌است که طوری سیستم را سازماندهی نماید که سرویس ارائه شده توسط یک مؤلفه بتواند توسط مؤلفه دیگری از سر گرفته شود.

۵-۳-۱-۳-الگوی‌سامانه مقاوم در برابر خطا با بررسی قیاسی[3]

در این الگو هدف ساخت سیستمی است که شکست غیروابسته در یک مؤلفه، سریعاً آشکار شود، به‌طوری که آن شکست در یک مؤلفه، موجب شکست کل سیستم نمی‌شود.

۵-۳-۱-۴-الگوی سامانه تکراری[4]

این الگو، سیستم را طوری ساختاردهی می‌کند که اجازهٔ آماده‌سازی سیستم از چند نقطه قابل انجام است تا در مواقع شکست یک یا چند مؤلفه یا ارتباط، سیستم بتواند به‌راحتی بازیابی شود.

۵-۳-۱-۵-الگوی آشکارساز/مصحح ایراد[5]

این الگو افزونگی را به داده اضافه می‌کند تا بتواند در مواقع ایراد، آشکارسازی و بازیابی را انجام دهد.

۵-۳-۲- گروه دوم، الگوهای مرتبط با حفاظت

الگوهای مرتبط با حفاظت که در ارتباط با تأمین محرمانگی و صحت هستند، عبارتند از:

۵-۳-۲-۱-الگوی سامانه حفاظت‌شده[6]

این الگو طوری سیستم را می‌سازد که همهٔ دسترسی‌های کلاینت‌ها به منابع، بوسیلهٔ واسطها انجام می‌گیرد که آن واسطها، سیاست‌های امنیتی لازم را تحمیل می‌کنند.

۵-۳-۲-۲-الگوی خط‌مشی[7]

این الگو برای هر مؤلفهٔ غیرهمبسته از یک سیستم‌اطلاعاتی، اجبار خط‌مشی امنیتی را جداسازی‌می‌کند تا از اجرای درست فعالیت‌های مرتبط با خط‌مشی اطمینان حاصل‌کند.

۵-۳-۲-۳-الگوی احراز هویت[8]

عموماً الگوهای‌دیگر روی این الگو گسترش‌می‌یابند، مثل الگوی خط‌مشی که معروف‌ترین استفاده‌کنندگان از این الگو JAAS و PAM هستند.

۵-۳-۲-۴-الگوی توصیف‌گر موضوع

این الگو دسترسی به خصوصیات مرتبط با امنیت یک موجودیت که قرار است عملیات روی آنها انجام شود را ارائه می‌کند.

۵-۳-۲-۵-الگوی ارتباط امن[9]

وقتی که لازم است دو بخش در مواجهه با یک تهدید با هم مرتبط شوند، این الگو اطمینان از تحقق اهداف سیاست‌های امنیتی را برآوده می‌سازد.

۵-۳-۲-۶-الگوی زمینه امنیت[10]

این الگو مرتبط با یک زمینه اجرایی مشخص یا یک فرایند، یک

[1] Checkpointed System
[2] Standby
[3] Comparator-Checked Fault-Tolerant System
[4] Replicated System
[5] Error Detection/Correction
[6] Protected System
[7] Policy
[8] Authenticator
[9] Secure communication
[10] Security Context

عملیات یا اقدام، امکانی را برای نگهداری خصوصیات و داده‌های امنیتی، فراهم می‌نماید.

۵-۳-۲-۷-الگوی انجمن امنیت[1]

این الگو ساختاری را تعریف می‌کند که براساس آن امنیت ارتباطی بین هر مشارکت‌کننده و اطلاعات مورد نیازش را برقرار می‌کند و وارسی‌های لازم برای اطمینان از دریافت حفاظت شدهٔ اطلاعات توسط بخش‌های شرکت‌کنندهٔ دیگر را انجام می‌دهد.

۵-۳-۲-۸-الگوی نایب امن[2]

این الگو روابط بین حفاظ‌های دو نمونه از یک سیستم حفاظت‌شده را هنگامی که یک نمونه در دل دیگری جا داده شده باشد، تعریف می‌کند.

۶- الگوهای امنیت و سازماندهی آنها

دیدگاه‌های مختلفی برای سازماندهی الگوهای امنیت مطرح شده است. هدف از سازماندهی و طبقه‌بندی الگوهای امنیت، ارائه تسهیلات به استفاده‌کنندگانی است که انتظار نمی‌رود خبرهٔ حوزه امنیت باشند و با این سازماندهی‌ها قادر خواهند شد بهترین الگو را از میان الگوها، متناسب با کار خود پیدا نمایند. در این بخش سازماندهی‌های معروف و مهم مورد بررسی قرار می‌گیرند و در بخش ۷ این مقاله سازماندهی مد نظر مولفان ارائه شده است و مزایا و معایب آن با سازماندهی‌های موجود تشریح شده است.

۶-۱- سازماندهی براساس مراحل چرخه عمر

در مقاله [۶]، نویسندگان از نقطه‌نظر چرخهٔ عمر نرم‌افزار به الگوهای امنیت پرداخته‌اند. الگوهای امنیت در مراحل سه‌گانهٔ استخراج نیازمندی‌ها، طراحی و پیاده‌سازی مورد بررسی قرارگرفته و یک بخش هم به مقوله مهندسی الگوهای امنیت از منظر چرخهٔ عمر الگوها تخصیص داده شده است. نویسندگان این مقاله بر این باورند که در مرحلهٔ تعیین نیازمندی‌ها، «دارائی‌ها» باید مد نظر قرارگرفته و دلائل حفاظت از این دارائی‌ها بررسی و نهایتاً نیازمندی‌های امنیتی به صورت قسمتی از نیازمندی‌های سیستم مشخص گردند. آنها الگوهای امنیت در این مرحله را به دو گروه الگوهای فرایند تحلیل و الگوهای مدل-پایه تقسیم نموده و در هر کدام تعدادی الگو را مورد بررسی قرار داده‌اند.

در الگوهای مرتبط با فرایند تحلیل، موارد زیر توسط آنها شناسایی شده‌است و ما به دلیل اهمیت این الگوها، توضیحات بیشتری از آنها را در این نوشتار می‌آوریم:

۶-۱-۱- الگوی مشخص نمودن نیازهای‌امنیت برای دارائی‌های سازمان[3]

توضیحات زیر در خصوص این الگو از فصل ششم کتاب[۱۸] به‌طور اجمال اخذ شده‌است. این الگو با توصیف و شرح دارائی‌ها، به ما کمک می‌کند تا بفهمیم که دارائی‌های‌ما در سیستم چه مشخصاتی دارند. این الگو شامل شناخت و هویت‌شناسی دارائی‌های تجاری عوامل کسب‌وکار با تأثیر امنیتی آنها، رابطهٔ بین دارائی‌ها و عوامل تجاری و نوع هر یک از آنهاست. دارائی‌های امنیتی باید دارای چهار خصوصیت باشند که عبارتند از محرمانگی، صحت، دسترس‌پذیری و جوابگو بودن به نیازها. سازمان انواع دارائی‌های کسب‌وکار که نیازمند حفاظت هستند را باید به طور نظام‌مند و واضح مشخص‌کرده و همچنین تعیین‌کند هرکدام به چه نوع حفاظتی نیاز دارند. این فعالیت اصولاً به وسیلهٔ یک معمار سازمانی یا برنامه‌ریز راهبردی انجام می‌شود که شامل پنج‌مرحله است:

۱. مشخص‌نمودن دارائی‌های کسب‌وکار (اطلاعات مربوط به دارایی‌ها مانند پرسنل و اطلاعات مالی، دارایی‌های فیزیکی مانند ساختمان‌ها)
۲. مشخص نمودن عوامل کسب‌وکار که بر روی نیازهای امنیتی دارائی‌ها هم در داخل و هم در خارج سازمان تأثیر می‌گذارند، قوانین و مقررات مانند قوانین حفظ حریم-خصوصی، روابط با شرکای سازمانی، مأموریت سازمان، اهداف و مقاصد و خط‌مشی، میل به سرمایه‌گذاری قوی مالی، فرآیندهای کسب‌وکار مانند حسابداری و فرآیندهای سفارش‌ها، وقایع حساس کسب‌وکار مانند فرآیندهای پرداخت‌های ماهانه، مکان‌هایی که در آن فرآیندهای کسب‌وکار و حوادث رخ می‌دهند.
۳. تعیین این که کدام یک از دارائی‌ها به کدام یک از عوامل کسب‌وکار مرتبط هستند. در این بخش ملاحظات زیر مورد توجه قرار می‌گیرند: (قانون حفظ حریم خصوصی که به اطلاعات کارکنان اعمال می‌شود، برخی از انواع دارائی‌های فیزیکی ممکن است فقط در یک محل خاص وجود داشته باشند، ممکن است اطلاعات مالی منتخبی نیاز به اشتراک‌گذاری با دیگر شرکاء را داشته‌باشند.)

[1] Security Association
[2] Secure Proxy
[3] Security Needs identification for Enterprise Assets Pattern

۴. شناسایی این که چه نوع امنیتی ممکن‌است موردنیاز باشد:

- محرمانگی: حفاظت در مقابل افشای غیرعمد و غیرمجاز
- صحت: حفاظت در مقابل تغییر غیرعمد و غیرمجاز
- دسترس‌پذیری: ساخت دارائی‌های کسب‌وکار قابل استفاده مجاز
- پاسخگویی: مجوز مسئولیت برای انجام اعمال

۵. بر اساس عوامل کسب‌وکار، تعیین اینکه هر نوع دارائی نیاز به چه نوع امنیتی دارد.

شناسایی دارائی‌های شرکت و نیازهای امنیتی از بهترین روش‌ها برای بهتر کردن عملکرد شرکت است، اما اغلب به‌صورت غیررسمی یا در راستای تجزیه و تحلیل مخاطرات امنیتی انجام می‌شود. مدل SSE-CMM[1] سطح توانایی فرآیند مهندسی امنیت مرتبط با ارزیابی ریسک را مشخص می‌کند. این مدل عناصری شبیه به این الگو را داراست:

- امنیت را در کل حوزهٔ سازمان مشخص می‌کند.
- نیازهای هماهنگ امنیتی برگرفته از نهادهای خارجی که شامل قوانین، سیاست‌ها و استانداردها هستند، را مشخص می‌کند.
- روند تأثیر دارائی‌ها شامل شناسایی و توصیف دارائی‌های سازمان و نیاز به یکپارچگی، محرمانگی، دسترس‌پذیری، اصالت و قابلیت اطمینان را شامل است.

فواید این الگو عبارت است از:

- تسهیل در ایجاد تعادل و تصمیم‌گیری‌های آگاهانه برای نیازهای امنیتی شرکت با استفاده از نیروهای رقیب و عوامل صریح کسب‌وکار را فراهم می‌سازد.
- نتیجهٔ مفید دیگر استفاده از این الگو قابلیت ردیابی از حفاظت دارائی‌هایی است که در کسب‌وکار ایجاد می‌شود.

این مدل همچنین مضراتی دارد که هزینه‌بر بودن، یکی‌از آنها است.

۶-۱-۲- الگوی ارزشیابی دارایی[2]

توضیحات زیر در خصوص این الگو از فصل ششم کتاب [۱۸] به‌طور اجمال اخذ شده است:

ارزشیابی دارایی کمک می‌کند که شرکت‌ها و سازمان‌های بزرگ دارائی‌هایی را که مالکیت آن‌را دارند یا آن را کنترل می‌کنند، مشخص و ارزش‌گذاری‌کنند. در این ارزشیابی شش سطح و درجه تعیین می‌گردد: از کم‌اهمیت‌ترین تا باارزش‌ترین و از سه دیدگاه مختلف نیازمندی‌های امنیتی، ارزش مالی و تأثیر بر کسب‌وکار مورد بررسی قرار می‌گیرند. بنابراین ما برای هر دارائی یک ارزش تعیین می‌کنیم که نه تنها ارزش بلکه مخاطره هر دارائی هم درنظر گرفته می‌شود. انواع دارائی‌ها شامل دارائی‌های اطلاعاتی مثل اطلاعات پرسنل، داده‌های مالی، داده‌های تحقیقاتی و دارائی‌های فیزیکی مثل ساختمان، وسایل نقلیه و کارمندان هستند. سازمان‌ها باید اهمیت کلی این دارائی‌ها را مشخص‌کنند. هدف در این الگو این است که یک سازمان بزرگ باید مشخص کند چه دارائی‌هایی در فرایند ارزیابی ریسک دخیل هستند و باید ارزش هر یک از آن دارائی‌ها را معین‌کند. مشکل اصلی این است که توانایی تعریف کردن یک مقدار برای دارائی، یک عنصر اساسی در هر قسمت از ارزشیابی فرایند تعیین مخاطره می‌باشد. تهدیدات و آسیب‌پذیری‌ها هم دارائی‌ها را افشاء می‌کنند، بنابراین باید دارائی‌ها، ارزش‌گذاری شوند. بدون تعیین این مسئله، یک سازمان قادر به ارزشیابی درستی از ریسک‌هایی که اتفاق می‌افتد، نیست. یک شرکت بزرگ باید یک روش استاندارد برای ارزشیابی و توصیف دارایی‌هایش داشته باشد و به‌علاوه باید قادر باشد هزینه‌هایی که منجر به از دست‌دادن یک دارایی می‌شود را ارزشیابی کند. راه‌حل این است که به‌صورت نظام‌مند یک مقدارکلی را برای هر دارائی در حیطه ارزیابی مخاطره آن، تعیین‌کنیم. این بدین معناست که چهار مرحلهٔ زیر را باید انجام دهیم:

۱. تعیین ارزش امنیتی آن: این مقدار بستگی به میزان اطلاعات امنیتی آن دارائی دارد، مثل محرمانگی، حفظ جامعیت، دسترس‌پذیری و پاسخگویی.
۲. تعیین ارزش‌مالی آن: شامل هزینه‌های تعمیر و جایگزینی یا نگهداری و راه‌اندازی. هزینه‌های برق و میزان فضای آن، موردی است که احتمالاً بین همهٔ دارائی‌ها توزیع شده-است.
۳. تعیین تأثیر آن بر کسب‌وکار: تعیین یک ارزش برای دارائی به جهت میزان تأثیری که بر کسب‌وکار می‌گذارد.
۴. تعیین یک ارزش‌کلی و ساخت یک جدول مقدار: تلفیق نتایج مقادیر امنیت، مالی، کسب‌وکار و تعیین یک مقدارکلی که یک سازمان روی یک دارائی می‌گذارد و در نهایت باید این نتایج را در جدول مقدار دارایی وارد کرد.

ارزشیابی دارائی‌ها یک جزء کلیدی از کلیهٔ ارزشیابی‌های

[1] The Systems Security Engineering Capability Maturity Model

[2] Asset Valuation Pattern

ریسک‌هایی است که به‌طور گسترده‌ای پذیرفته شده‌اند از جمله [Pel01], [ISO17799], [ISO13335-3] [IST800-30]. این الگو محاسن زیر را دارا می‌باشد:

- منجر به بدست‌آوردن یک دیدگاه واقع‌بینانه و کامل از دارائی‌هایی می‌شود که در کسب وکار حیاتی هستند.
- نتایج ارزیابی دارائی‌ها می‌تواند به منظور توسعه یا بروزرسانی سازمان و طرح بازیابی و تداوم کسب وکار استفاده شود.
- ارزش‌کیفی، به‌نسبت بدست آوردن هزینه‌های سنگین مقادیر کمی ارزشیابی دارائی، ساده‌تر بدست می‌آید و درنتیجه منجر به تسریع روند کلی ارزشیابی مخاطره می‌گردد.
- یک سازمان بزرگ ممکن است مجبور به تغییر شیوهٔ خود گردد، اگر یک دارائی بیشتر از آنچه فکرش را می‌کردند، ارزش داشته باشد. این مسئله منجر به سود بیشتر در درازمدت می‌گردد.

۶-۱-۳- الگوی ارزیابی تهدید[1]

توضیحات زیر در خصوص این الگو از فصل ششم کتاب[۱۸] و مقاله[۱۶] به‌طور اجمال اخذ شده است:

تهدیدات، شرایط بالقوه بروز حوادث هستند. آنها می‌توانند بر روی هر دارائی که در موقعیت مرتبط با تهدید باشند، تأثیر بگذارند. ارزیابی تهدید یک‌سازمان، تهدیدهای وابسته به دارائی‌های سازمان را شناسایی می‌کند و احتمال یا وسعت وقوع حوادث روی آنها را تعیین می‌نماید. مسئله اصلی این است که یک سازمان ممکن است نتواند وسعت و دامنهٔ دارائی‌های درمعرض خطر را تشخیص دهد و لازم است مطابق یک روش نظام‌مند دامنهٔ تهدیدات و خطر بروز آنها را همواره مورد سنجش و ارزیابی قرار دهد. مطابق این الگو، نقشه راهبردی کار تهیه و مراحل زیر مطابق آن انجام می‌گردد:

۱. شناسایی و تشخیص تهدید شامل منبع تهدید، کاری که صورت می‌گیرد و پیامدهای آن
۲. ساخت جدول تهدیدات
۳. تعیین احتمالات
۴. درجه‌بندی هر تهدید

ویژگی‌های زیر درخصوص این الگو قابل ذکر است:

- عواملی که باعث کم شدن وسعت تهدید و میزان مخاطره می‌شوند، مشخص می‌گردند.
- تشخیص پیامدهای تهدیدات امکان‌پذیر می‌شود.
- تلاش لازم برای درک تمام تهدیدات احتمالی می‌تواند زمان زیادی را به‌خود تخصیص‌دهد.

۶-۱-۴- الگوی رویکردهای امنیت سازمان[2]

توضیحات زیر در خصوص این الگو از فصل ششم کتاب [۱۸] به‌طور اجمال اخذ شده‌است:

این الگو شامل پیشگیری، ردیابی و واکنش به مسائل و مشکلات امنیتی است و اساس و پایه‌ای است برای تصمیم‌گیری اینکه چه سرویس‌های امنیتی باید توسط سازمان‌ها بنا نهاده شوند. در این زمینه دارائی‌های یک کسب‌وکار که نیاز به محافظت دارند و خصوصیات و عناصر امنیت مورد نیاز آن (محرمانگی، صحت، دسترس‌پذیری) باید مورد شناسایی دقیق واقع گردند. برای هر یک از انواع دارائی‌ها که نیاز به محافظت دارند یک مجموعه از رویکردهای یکپارچه مشخص می‌شود. فرآیند کار بر روی دو منظر تکیه دارد که عبارتنداز منظر و دیدگاه انفرادی از نوع دارائی و منظر و دید جامع و سراسری سازمان. برای هر نوع دارائی به‌طور نظام‌مند و صریح و آشکار مجموعه‌ای از مخاطرات امنیتی مورد امتحان قرارگرفته و بدینوسیله رویکرد امنیتی مناسب و اولویت‌های پیشنهادی کسب‌وکار، استفاده می‌شود. فرآیند تعریف رویکردها به‌طور نمونه توسط یک معمار یا طراح استراتژیک اجرا می‌گردد. قدم اول جمع‌آوری تمام اطلاعات لازم شامل انواع دارائی و ملزومات امنیتی آنها می‌باشد. سپس اطلاعات مربوط به شرایط مخاطره که بر رویکردها تأثیر دارند، جمع‌آوری می‌شوند و در نهایت رویکردهایی که انتخاب شده‌اند یکپارچه می‌گردند.

مزایای حاصل از بکارگیری این الگو عبارتند از:

- سطح آگاهی مدیریت را ارتقاء می‌دهد.
- اطلاعات لازم برای نحوهٔ تصمیم‌گیری درباره رویکرد امنیتی مربوط به شناسایی نیازمندی‌های امنیتی را ارائه می‌دهد و به عبارتی آگاهی‌های لازم را فراهم می‌کند.
- به تخصیص بهتر منابع برای حفاظت از دارائی‌ها کمک می‌کند.
- امکان بازخورد در فرایند تصمیم‌گیری را فراهم می‌کند.
- ایجاد تعادل منطقی در انتخاب‌های بین امنیت و کارایی را فراهم می‌کند.
- نشان می‌دهد که می‌توان رویکردها را به منظور حفاظت ساده‌تر از دارائی‌ها با یکدیگر تلفیق نمود.

در ادامه الگوهای مدل-پایه مورد اشاره قرار می‌گیرند:

[1] Threat Assessment Pattern

[2] Enterprise Security Approaches Pattern

۶-۱-۵- الگوهای مبتنی بر تروپوس[1]

تروپوس یک متدولوژی توسعه سیستم‌های نرم‌افزاری امنیت-گرا می‌باشد که مفاهیم مهندسی نیازمندی‌ها مانند بازیگر، هدف، برنامه را با مفاهیم مهندسی امنیتی مانند تهدید، محدودیت امنیتی و مکانیزم امنیتی، تحت یک فرآیند واحد برای پشتیبانی از تجزیه و تحلیل و توسعه سیستم‌های نرم‌افزاری امن و قابل اعتماد ترکیب می‌کند و شامل الگوهای پیشنهادی توس گیورگینی[2] که مشخص‌کنندهٔ یک نوع محرمانگی خاص هستند و الگوهای پیشنهادی موراتیدیس[3] برای حفاظت در برابر عوامل بدخواه می‌باشد.

۶-۱-۶- الگوهای مبتنی برقالب مسئله

این الگو توسط هیتبر[4] و همکاران وی ارائه شده است. در مرحله طراحی، معتقدند که باید کارکردهایی را طراحی کنیم که خصوصیات امنیت مانند دسترس‌پذیری، محرمانگی، صحت و پاسخگویی که در مرحله نیازمندی‌ها معلوم شدند را تأمین و برآورده کنند. طراحی مواردی مثل کنترل دسترسی، تصدیق اصالت، رمزنگاری، امضای دیجیتالی و ثبت رخداد، به‌طور خاص در اینجا طرح شده‌است. آنها متعقدند طراحی کارکردهای امنیت نه تنها به خصوصیات چهارگانهٔ مذکور، بلکه به استراتژی‌های امنیت مانند استراتژی «پیش‌گیری در قبال حملات» یا «آشکارسازی حملات» وابسته است.

۶-۱-۷- الگوهای مرتبط با معرفی معماری مفهومی

تلاش‌های یادروبارکالو[5] برای معرفی معماری مفهومی امنیت به صورت الگو شامل ارائه یک توصیف زبان طبیعی از هفت الگوی امنیت به نام‌های Single Access Point, Check Point, Roles, Session, Full View with Errors, Limited View, Secure Access Layer است.

۶-۱-۸- الگوهای امنیت نشان داده شده با UML

این بخش شامل معرفی تلاش‌های صورت گرفته توسط فرناندز و پن[6] برای نشان‌دادن الگوهای امنیت به‌وسیله UML به‌خصوص مواردی مانند اجازه دسترسی، کنترل دسترسی نقش-پایه، امنیت چندسطحی و اجازهٔ دسترسی فایل است.

۶-۱-۹- الگوهای ۲۵گانه

این بخش شامل ۲۵ مورد الگوی امنیت در سطح معماری و طراحی، دسته‌بندی شده در کتاب[۱۸] است.

علاوه بر موارد فوق، نویسندگان [۱۳] تلاش نمودند تا در این مرحله نحوهٔ استفاده از الگوهای طراحی را همراه با ملاحظات امنیتی تبیین نمایند. برای مثال به کمک الگوهای معرفی‌شده در کتاب[۱۸]، سه الگوی دیواره آتش در زمینه ویژگی دسترس‌پذیر بودن خدمات امنیت معرفی شده‌اند که عبارتنداز:

- سطح IP: الگوی دیواره آتش پالایش بسته[7]
- سطح انتقال: الگوی دیواره آتش نیابتی[8]
- سطح خدمت: الگوی دیواره آتش با حالت کامل[9]

این الگوها در تأمین ویژگی محرمانگی نیز کمک می‌کنند[۷]. در ادامه برای ویژگی پاسخگویی که یکی از ویژگی‌های چهارگانه مطرح‌شده در بخش خصوصیات امنیت است نیز تعدادی الگو را از کتاب[۱۸] معرفی نموده‌اند که همهٔ آنها از نوع الگوهای فرایندی هستند[۷] و عبارتند از:

- الزامات حسابداری امنیتی[10]
- الزامات ممیزی[11]
- دنباله‌های ممیزی[12]
- الزامات تشخیص نفوذ[13]
- الزامات عدم انکار[14]

برای مرحله پیاده‌سازی، آنها معتقدند وجود خطوط راهنما کفایت نمی‌کند چون خطوط راهنما به آسانی قابلیت استفاده مجدد نداشته و به سفسطه‌گویی گرایش دارند. به زعم آنها ما به الگوهای امنیت در سطح پیاده‌سازی نیاز داریم که مستندات نیمه‌ساخت‌یافته یا واژگان استانداردی هستند که به‌طور واضح خطوط راهنما را بیان می‌کنند و محصولات مرتبط با آنها را ارائه می‌دهند[۷]. در مرحله پیاده‌سازی نویسندگان به تعدادی الگوی مرتبط با حملات اشاره

[1] Tropos
[2] Giorgini
[3] Mouratidis
[4] Hatebur
[5] Yoder and Barcalow
[6] Fernandez and Pan
[7] IP Level: Packet Filter Firewall pattern
[8] Transport Level: Proxy Firewall pattern
[9] Service Level: Stateful Firewall pattern
[10] Security Accounting Requirements
[11] Audit Requirements
[12] Audit Trail
[13] Intrusion Detection Requirements
[14] Non-Repudiation Requirements

می‌کنند که بیان‌کننده این است که چه وقتی به چه چیزی و چگونه حمله انجام می‌شود و موجب درهم شکستن نرم‌افزار می‌گردد. دانستن این الگوهای حمله برای برنامه‌نویسان این امکان را ایجاد می‌کند که بهبودهای لازم را برای در امان بودن از این الگوهای حمله ایجاد نمایند[۷]. الگوهای بازآرایی[1] که وجوهی از امنیت را تأمین می‌کنند نیز به عنوان الگوهای امنیتی مرحله پیاده‌سازی، مورد بحث قرار گرفته‌اند.

مهندسی الگوی امنیت اشاره به فعالیت‌هایی دارد که شبیه چرخه عمر الگوهای نرم‌افزار هستند و شامل دو فرآیند می‌باشند، فرایند استخراج و فرآیند بکارگیری[۷]. در این مقاله درخصوص نحوهٔ استخراج الگوها و ارتباط آنها با شناخت نیازمندی‌های امنیت، پیداکردن ماشینی الگو برای یک مسئله و زمینهٔ داده‌شده با استفاده از هستان‌شناسی و نیز معیارهای غیروظیفه‌ای[2]، دسته‌بندی الگوها با روش‌های مختلف مانند مدل CIA و استراید[3] بحث نموده و نیز به مواردی مانند ذخیره‌سازی الگوها و دستیابی به آنها فراخور مسئله، آنالیز کیفیت و متدلوژی ایجاد به‌وسیله الگوهای امنیت، پرداخته و نهایتاً کفایت موضوع را بررسی و ابراز داشته است که در موارد زیادی نیاز به کشف الگوها داریم. در مقاله فوق بیان شده‌است که در مرحلهٔ طراحی، جهت کوتاه‌کردن فاصله بین مرحله شناخت نیازمندی‌ها، طراحی و مرحله پیاده‌سازی به الگوهایی برای مخاطره و حمله نیاز است. نیاز به متدولوژی برای طراحی الگوهای‌حمله در سطح طراحی و نیز یکسان‌سازی الگوهای امنیت با مدل‌های نظام، از جمله نیازهای مهم آتی برشمرده شده است.

۶-۲- سازماندهی براساس بازنمایی سطوح منطقی

در [۱۹] الگوهای امنیت براساس بازنمایی سطوح منطقی کاربردهای مبتنی‌بر وب تقیسم‌بندی شده است و این تقسیم‌بندی کلی را برای فناوری‌های J2EE تعمیم داده و به‌طور خاص ارائه می‌کند. براساس تقسیم‌بندی عمومی این روش، الگوها در سطوح وب، کسب‌وکار، یکپارچه‌سازی و قابلیت‌های زیرساختی کیفیت سرویس، مورد تفکیک قرار می‌گیرند. جداول خلاصه زیر این الگوها را برای سه سطح وب، کسب‌وکار و یکپارچه‌سازی بیان می‌کند.

الف- الگوهای لایهٔ وب:

در این بخش شش الگو به اختصار معرفی می‌شوند که عمدتاً درجهت تأمین امنیت لایهٔ وب معرفی و استفاده می‌شوند (جدول ۲).

ب- الگوهای لایه کسب وکار:

در جدول ۳ الگوهای مرتبط با لایه کسب و کار آمده است.

ج- الگوهای لایهٔ یکپارچه‌سازی:

در جدول ۴ الگوهای لایه یکپارچه‌سازی نام برده شده است.

جدول ۲. الگوهای لایهٔ وب[۱۹]

ردیف	نام الگو	توضیح الگو
۱	ارتباط امن	این الگو استفاده از لایهٔ انتقال داده امن را برای ارتباطات مشتری به کارپذیر و کارپذیر به کارپذیر ارائه و تشریح می‌کند.
۲	انجمن امن	نشان می‌دهد بین دو موجودیت چگونه تعامل امن برقرار نمائیم. برای مثال حفاظت از نشست بین مرورگر و کارپذیروب‌با استفاده از SSL یا TSL و ایمیل امن با استفاده از رمزنگاری و پراکسی.
۳	نقطه دسترسی منفرد	این الگو اجبار می‌کند که یک مدخل ورودی منفرد برای خدمات کسب وکار و کاربردها داشته باشیم و یک صفحه یا اعلان لاگین را برای این کار ارائه می‌کند.
۴	نقطه بررسی	فرایند احراز هویت و مجوزدهی را بوسیله یک نقطه وارسی، متمرکز می‌کند. این الگو استفاده از JAAS را برای پیاده‌سازی سیستم Check Point مفروض می‌داند.
۵	نشست	در کاربردهای امن همواره نیاز داریم اطلاعات سراسری را در کل چرخه حیات کاربرد بدست آوردیم. این الگو اطلاعات نشست را (برای مثال متغیرهای نشست http اطلاعات فراخوانی RPC، جزئیات سفارش دهنده سرویس در JMS یا SOAP] که باید برای پی‌گیری‌های امنیتی نگهداری شوند، مشخص می‌کند.
۶	تامین کننده امنیت	تشریح می‌کند که یک کلاینت در قبال ادعای احراز هویت و مجوز دهی ارائه کننده سرویس شناسایی، چه باید بکند.

[1] Refactoring
[2] Non Functional Requirements
[۳] STRIDE مدلی برای شناسایی تهدیدات امنیتی رایانه است که توسط Praerit Garg و Loren Kohnfelder در مایکروسافت توسعه یافته است و شامل شش دسته است.

جدول ۳. لگوهای لایهٔ کسب‌وکار[۱۹]

ردیف	نام الگو	توضیح الگو
۱	نقش	تفکیک و جدا شدن یک کاربر خاص از امتیازاتش را با استفاده از نقش او نشان می‌دهد.
۲	توضیف کننده موضوع	این الگو اجازه می‌دهد دسترسی به صفات امنیتی یک موضوع از طریق عملیات انجام پذیرد. این الگو مطابق استبا javax.security.auth.Subject در JAAS این الگو می‌تواند برای بررسی مجوزها و اختیارات مورد استفاده قرار گیرد.
۳	بافتار امنیت	امکان لازم برای دسترسی به خصوصیات امنیتی مثل شناسه کاربر و شناسه گروه را ارائه می‌کند.
۴	دیدکامل با خطا	این الگو یک دید کامل به کاربر در باره ایرادات رخ داده شامل استثنات ضروری را ارائه می‌دهد.
۵	دید محدود	اجازه می‌دهد کاربران فقط آنچه را که دسترسی دارند ببینند.
۶	ثبت رخداد امنیت	این الگو مرتبط است با بدست آوردن و پیگیری کردن رخدادهای مرتبط با امنیت برای لاگ کردن و دنبال کردن ممیزی. اطلاعات لاگ شده برای ارزیابی ریسک یا تحلیل مورد استفاده قرار می‌گیرد.

جدول ۴. الگوهای لایه یکپارچه‌سازی[۱۹]

ردیف	نام الگو	توضیح الگو
۱	Source of Data Authoritative	این الگو منبع داده را برای احراز هویت و صحت داده وارسی می‌کند.
۲	Third-Party Communication	این الگو کمک می‌کند ریسک‌های روابط طرف ثالث مشخص شده و معیارهای حفاظت امنیت برای ارتباطات طرف ثالث را بکار می‌گیرد.

۶-۳- سازماندهی براساس طبقه‌بندی تهدیدپایه

حفیظ و همکارانش در[۲۰] به کمک طبقه‌بندی توانستند سازماندهی مناسبی از الگوهای امنیت ارائه دهند. آنها در این مقاله متعقدند سازماندهی مناسب الگوهای امنیت هم برای نویسندگان الگوها و هم استفاده‌کنندگان مفید هستند و هنوز سازماندهی مناسبی وجود ندارد و به‌علاوه وجود تعداد زیاد الگوها امکان پیدا کردن مناسب‌ترین آنها را سخت کرده است. آنها با بهره‌گیری از تعریف الگو که عبارت است از یک راه‌حل برای یک مسئله در یک زمینه[۲۰]، سه عنصر کلیدی دامنه[۱]، زمینه[۲] و مسئله[۳] را از این تعریف استخراج نموده و به عنوان سنجه‌های سازماندهی در نظر گرفتند:

۱. طرح رده‌بندی براساس مفاهیم دامنه شامل مفاهیم پایه‌ای محرمانگی، صحت، دسترس‌پذیری (مدل CIA)

۲. طرح رده‌بندی براساس زمینه:

- زمینهٔ برنامه: قسمتی از سیستم که الگو در آنجا بکار گرفته می‌شود. این بخش از مدل‌های نظامی که در سه بخش مدخل ورودی، پیرامون و هستهٔ اصلی دربردارندهٔ زیرساخت نسبت به استقرار امنیت اقدام می‌کنند، اخذ شده است. لذا سه بخش هسته، پیرامون و بخش بیرونی درنظر گرفته شده است.
- ذی‌نفعان و دیدهای آنها: دسته‌بندی براساس چارچوب جدولی زکمن

۳. طرح رده‌بندی براساس دامنه مسئله و مدل تهدید: براساس مشکلی که الگوها حل می‌کنند، طبقه‌بندی صورت می‌گیرد. در این تحقیق، ۱۴ الگو برای نمونه انتخاب شده است، شامل موارد مندرج در جدول ۵ است:

جدول ۵. الگوهای چهاردهگانهٔ نمونه برای بررسی[۲۰]

شماره	نام الگو	شماره	نام الگو
۱	Authenticator	۸	Policy Enforcement Point
۲	Authorizatin	۹	Replicated System
۳	Checkpointed System	۱۰	Safe Data Buffer
۴	Defense in Depth	۱۱	Secure Pre-Forking
۵	Exception Shielding	۱۲	Single Access Point
۶	Minefield	۱۳	Subject Descriptor
۷	Password Synchronizer	۱۴	Grey Hats

طرح اول براساس مدل CIA:

- محرمانگی شامل هفت الگوی ۱و ۲ و ۵ و ۸ و ۱۱ و ۱۲ و ۱۳
- صحت یک الگو شامل ۱۰
- دسترس پذیری یک الگو شامل ۳
- صحت و دسترس‌پذیری دو الگو شامل ۷ و ۹

طرح دوم براساس زمینه: زمینه برنامه (کاربرد):

- هسته برنامه۶ الگو شامل ۳ و ۵ و ۶ و ۱۰ و ۱۱ و ۱۳
- محیط برنامه چهار الگو شامل ۱ و ۲ و ۸ و ۱۲
- بیرون از برنامه دو الگو شامل ۷ و ۹

[1] Domain Concepts
[2] Context
[3] Problem

- همه طبقات دو الگو شامل ۴ و ۱۴

در جدول ۶ تطابق الگوهای چهارده گانه جدول ۵ با جدول زکمن تشریح شده است.

جدول ۶. تطابق الگوهای چهاردهگانهٔ فوق با جدول زکمن [۲۰]

دید	وظیفه	داده	آزمون
دید معمار	الگوهای ۳ و ۸ و ۹ و ۱۱ و ۱۲		
دید طراح	۱ و ۷	۵ و ۱۳	
دید پیاده‌ساز		۱۰	۱۴

در این خصوص مطابق شش بُعد مدل استراید موارد زیر تطبیق داده شده است:

- جعل: سه الگو شامل ۱ و ۷ و ۱۳
- دستکاری سه الگو شامل ۳ و ۱۰ و ۱۲
- انکار: هیچکدام
- افشای اطلاعات: دو الگو شامل ۲ و ۵
- منع سرویس: یک الگو شامل ۹
- بالابردن امتیاز: یک الگو شامل ۱۱
- همه طبقات: چهارالگو شامل ۴ و ۶ و ۸ و ۱۴

کار اساسی که در این مقاله انجام گرفته است، علاوه بر ارائه انواع رده‌بندی‌ها، ارائه یک درخت سلسله‌مراتبی از الگوها مطابق با مدل تهدید استراید (برگرفته از حروف اول شش مفهوم آمده در شعبه‌های شکل ۳ است شامل جعل[۱]، دستکاری[۲]، انکار[۳]، افشای اطلاعات[۴]، منع‌سرویس[۵] و بالابردن امتیاز[۶]) و زمینهٔ کاربرد[۷] است که صورت پذیرفته است. (درخت شکل ۳ برگرفته شده از [۲۰] می‌باشد.)

۶-۴- سازماندهی براساس طبقه‌بندی حمله-پایه

در این راستا تلاش مهمی توسط آندراس ویسار[۸] و همکارانش انجام شده که در مقاله[۲۱] ارائه شده‌است. از دیدگاه این مقاله همهٔ طرح‌های رده‌بندی ارائه‌شده دارای عیب هستند. رده‌بندی‌های مبتنی بر مفهوم امنیت برای مثال دقیق نیستند و یک الگو ممکن است ذیل چندین طبقه قرارگیرد. رده‌بندی براساس لایهٔ منطقی موجب می‌شود که یک چشم‌انداز و منظر از الگوها پدید آید، ولی کمکی به انتخاب مناسب نمی‌کند زیرا معلوم نمی‌شود کجا و به کدام دلیل یک الگو بکارگرفته شده است.

شکل ۳. شکل سلسله‌مراتبی تطابق الگوها با مدل تهدید استراید[۲۰]

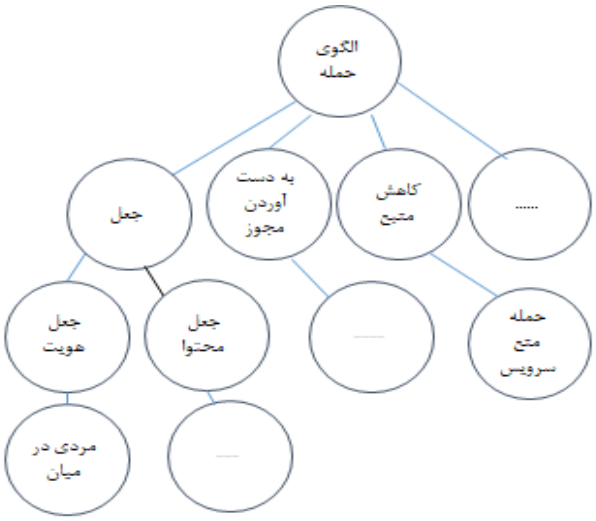
رده‌بندی جدولی براساس انواع دیدگاه‌ها، دید خوبی ارائه می‌کند ولی با نیازمندی‌های امنیتی هم‌راستا نشده است[۲۱]. ایدهٔ این مقاله استفاده از الگوهای حمله برای دسته‌بندی الگوها است. این مقاله به کمک الگوهای حملهٔ معرفی شده در سایت کپک[۹] طبقه‌بندی خود را ارائه می‌دهد. در این سایت، شرکت MITRE طبقه‌بندی کاملی از حملات ارائه نموده و هدفش را از ارائه این اطلاعات به این صورت بیان می‌دارد (خلاصه شده از[۲۲]): «نهاد کپک این اطلاعات را برای برنامه‌نویسان و طراحان، آزمون‌کننده‌ها، آموزش‌دهنده‌ها و همهٔ کسانی‌که می‌خواهند در چرخهٔ عمر سیستم، قابلیت‌های امنیت را توسعه دهند، در قالب طبقه‌بندی و به‌صورت عمومی ارائه می‌نماید. هر الگوی حمله دربردارندهٔ دانستنی‌های مرتبط با طراحی و نحوهٔ اجرای حمله توسط حمله‌کننده و راهنمای کاهش اثرات آن‌است.»

در روش کپک هر الگوی حمله دارای یک شناسه است و اطلاعات کاملی از سناریوی اجرای حمله تا راه‌حل‌های مقابله و نیز دیگر اطلاعات مثل رابطهٔ الگوی حمله با دیگر الگوهای حمله داده شده است. در [۲۱] به مدد این اطلاعات و استفاده از مدل تهدید استراید، طبقه‌بندی جدیدی معرفی شده‌است که نمونهٔ شکل ۴ را به عنوان گلچین ارائه داده‌است. این نمونه ساختار کار را نشان می‌دهد. در لایهٔ دوم از طبقه‌بندی، شش‌بُعد مدل تهدید استراید ارائه‌شده و ذیل هرکدام حملات مرتبط و الگوهای حمله درج شده‌اند.

[1] Spoofing
[2] Tampering
[3] Reputation
[4] Information disclosure
[5] Dos
[6] Elevation of Privilege
[7] Application Context
[8] Andreas Wiesauer
[9] capec.mitre.org

مزیت این روش نسبت به بقیه آن است که استفاده‌کنندگان پس از مشخص شدن الگوی یک حمله خاص، به‌آسانی می‌توانند الگوی امنیت مرتبط را پیداکرده و به‌علاوه این روش کمک می‌کند الگوهای مشابه مشخص شوند[۲۱]. الگوهای طراحی امنیت در یک دسته الگوی حمله یکسان، ممکن است شبیه یا حتی از افزونگی برخوردار باشند و طبقه‌بندی ارائه شده کمک می‌کند منظر و چشم‌انداز الگوهای طراحی امنیت به شکل منطقی‌تری ارائه شود[۲۱]. مثال زیر از [۲۲] برای کمک به فهم بهتر مطلب فوق آورده شده‌است.

«یکی از حمله‌های تعریف شده در کاتالوگ کپک، حملهٔ مردی در میانه با ID94 است. این حمله ارتباط کانال بین سرویس‌دهنده و مشتری را هدف قرار می‌دهد. حمله‌کننده سعی دارد از اطلاعات مبادله‌شده بین دو طرف استراق سمع کند. هدف الگوهای لوله امن و کانال امن، امن نمودن ارتباطات بین موجودیت‌های مختلف است. هر دو الگو شبیه هم هستند و کمک می‌کنند تا از فعالیت مردی در میانه پیش‌گیری گردد. اگر طراحان نسبت به حساس بودن سیستم طراحی شده نسبت به چنین حملاتی واهمه داشته باشند، طبقه‌بندی پیشنهادی به آنها کمک می‌کند تا الگوی مواجهه مناسب را پیداکنند.»



شکل ۴. طبقه‌بندی برای جایدهی الگوها مطابق با الگوهای حمله[۲۱]

۶-۵- سازماندهی براساس دامنه کاربرد

مقاله[۲۳] تلاش وسیعی برای بررسی همهٔ منابع مرتبط با الگوهای امنیت را بخرج داده است. میخاییل بونکه[1] و همکاران مدعی هستند کلیهٔ نوشتگان زمینه را از سال ۱۹۹۷تا سال ۲۰۱۰ بصورت نظام‌مند بررسی‌نموده و توانستند ۴۰۹ الگو را مطابق با دامنهٔ کاربرد مشتمل بر ۱۶۲ الگو در حوزه نرم‌افزار، ۸۴ الگو در حوزه سازمان، ۵۶ الگو در زمینهٔ شبکه، ۲۳ الگو در زمینه کاربَر و ۳۵ الگو در زمینهٔ رمزنگاری دسته‌بندی نمایند. شکل۴ این دامنه‌ها را نشان می‌دهد [۲۳].

۷- پیشنهاد چارچوب جدید: چارچوب پویا و پسانِگر در مقابل چارچوب ایستا و پیشانِگر

دسته‌بندی‌های الگوهای امنیت باید مبتنی بر یک روش نظام‌مند باشند تا بتوانند الگوها که از تعدد بالایی برخوردار هستند را دسته‌بندی کنند و به‌علاوه، یک دسته‌بندی باید قادر باشد الگوهای جدید را در خود جای دهد[۲۳]. بررسی دسته‌بندی‌های ارائه شده نشان می‌دهد که همگی آنها از ویژگی‌های زیر به‌طورکامل برخوردار نیستند:

- پوشش همه ذی‌نفعان برای امنیت سامانه در کل چرخه حیات سیستم
- همهٔ وجوه و حوزه‌های اساسی امنیت مدنظر قرار نمی‌گیرند.
- براساس الگوهای موجود هستند و قابلیت توسعه برای پذیرش الگوهای آینده را مشخص نمی‌سازند

به‌علاوه اگر چارچوبی باشد که قابلیت پذیرش و جایدهی الگوهای امنیت چابک را بهتر از چارچوب دیگر کسب نماید برای ما مطلوب‌تر است. منظور از الگوهای چابک امنیت، الگوهایی هستند که خصوصیات شش‌گانهٔ زیر با عنوان خصوصیات SAREPH را داشته باشند[۲۴]:

- S : خودسازمانده[2]
- A : وفق‌دهنده خود با وضعیت‌های پیش‌بینی‌نشده[3]
- R : دارای واکنش بازگشت‌پذیر[4]
- E : تحول‌پذیر در مقابل تغییرات محیط[5]
- P : نوع‌آوری پیش‌گیرانه[6]
- H : هماهنگ با اهداف سامانه[7]

علی‌رغم ارائه انواع طبقه‌بندی‌ها و بحث‌های مختلف صورت‌گرفته در این‌خصوص، هنوز نقشهٔ کاملی که هر ذی‌نفع بداند چگونه می‌تواند به الگوی مناسب برای کارش رهنمون شود، وجود ندارد. اگرچه مطابق چارچوب زکمن، به این موضوع پرداخته شده است[۲۰] ولی نه تنها دسته‌بندی ذی‌نفعان کامل نیست بلکه نظام مواجهه‌اش به

[1] Michaela Bunke
[2] Self-organizing
[3] Adapting to unpredictable situations
[4] Reactively resilient
[5] Evolving in concert with a changing environment
[6] Proactively innovative
[7] Harmonious with system purpose

مفاهیم امنیت ناقص بوده و حداکثر به‌صورت افزودن یک ستون اضافی برای امنیت است[۱۸]. یعنی فضای مواجهات امنیت به‌عنوان یک لایهٔ واسط برای تسریع در آدرس‌دهی منظور نشده است. برای مثال معمار به عنوان یک ذی‌نفع از کدام فضای گفتمانی امنیت مثل تهدید، آسیب‌پذیری، کنترل، دارائی، خصوصیت، استاندارد و غیره باید به وظایف، داده‌ها و آزمون بپردازد؟ این یعنی نیاز به تلاقی دیدگاه‌ها (ذی‌نفع) با هستان‌شناسی امنیت. این همان ایدهٔ جدید است. برای نیل به این منظور ابتدا مجموعه ذی‌نفعان و حوزه‌های اصلی امنیت را بدست می‌آوریم. مطابق با [۲۵] در چرخهٔ عمر سیستم انواع ذی‌نفعان وجود دارند مانند مدیریت عالی، مسئولین واحد کسب‌وکاری، مدیر فناوری اطلاعات، متخصصین امنیت، مالکان برنامه کاربردی، برنامه‌نویسان/توسعه‌دهندگان، مدیران پروژه/رهبران تیم، معماران فنی، مدیران تضمین کیفیت، تحلیل‌گران کسب‌وکار، مسئولین تحویل، مدیر پروژه سمت مشتری، و ممیزان.

مطابق‌با هستان‌شناسی امنیت[۶] تعداد یازده حوزه کلان و اصلی تهدید، منبع تهدید، منشاء تهدید، آسیب‌پذیری، شدت آسیب‌پذیری، کنترل، نوع کنترل، کنترل استاندارد، سازمان، دارائی، و خصوصیت امنیتی قابل احصاء است.

چارچوب کلی طرح در زیر نشان داده شده است. مطابق این چارچوب ما تعداد ۱۳x۱۱مدخل داریم که هر مدخل باید دقیقاً مطابق خصوصیات و نقش‌های هر ذی‌نفع و فضای مواجهه امنیتی او یعنی یک یا چند تا از هستان‌های گفته شده و روابط آن هستان با بقیه هستان‌ها موردبررسی و تحقیق قرار گیرد. لزوماً همه مدخل‌ها در این چارچوب ممکن است معنی‌دار نباشند، ولی هر مدخل به‌تنهایی می‌تواند موضوع یک تحقیق باشد. پیشنهاد تحقیقات آتی این نوشتار بر تکمیل این مدخل‌ها بنا می‌شود.

راهنمای استخراج هر مدخل بصورت کلی در زیر آمده‌است:

الف-مشخص‌کردن وظایف ذی‌نفعان مختلف در چرخه حیات سیستم مانند موارد زیر[۲۶]:

- استخراج نیازمندی‌ها
- تحلیل نیازمندی‌ها
- نوشتن مستندات نیازمندی‌های نرم‌افزار
- ساخت و تحلیل نمونه
- انجام طراحی‌های نرم‌افزار
- نوشتن مستندات طراحی‌های نرم‌افزار
- برنامه‌نویسی
- تحقیق در خصوص تکنیک‌های مهندسی نرم‌افزار یا به‌دست آوردن اطلاعات در بارهٔ دامنه‌کاربرد
- ایجاد استراتژی‌های آزمون و موردهای آزمون
- آزمون نرم‌افزار و ثبت نتایج
- جداسازی مسائل[1] و حل‌آنها
- یادگیری استفاده یا نصب و تنظیم ابزار جدید نرم‌افزاری و سخت‌افزاری
- نوشتن مستندات مانند راهنمای استفاده
- شرکت در جلسات با همکاران، مشتریان و سرپرستان
- آرشیو نرم‌افزار یا آماده‌سازی آن برای توزیع

ب- تبیین حوزه‌امنیتی مرتبط با مدخل شامل مواردی مانند:

- مفهوم
- رابطه‌اش با مفاهیم دیگر در هستان‌شناسی
- مستندات ارائه شده تاکنون، مانند استانداردها و راهنماها

در این بخش بررسی استانداردهای مرتبط، کمک مهمی برای کشف الگوها می‌نماید. بدست آوردن الگوها روش‌های مختلفی دارد که استفاده از استانداردها در این خصوص بسیار ارزشمند جلوه می‌کند. اولین گام، کشف روابط بین مفاهیم و واژگان موجود در استانداردها و عناصر الگوها یعنی زمینه، مسئله و راه‌حل است[۲]. ممکن است به چند استاندارد برای استخراج یک الگو مراجعه کنیم. این روش این مزیت را دارد که استانداردها توسط افراد خبره خلق شده‌اند و قابل‌اعتماد و استناد هستند[۲]. وجود استاندارد، امکان بدست‌آوردن واژگان استاندارد و ساختار رسمی را به‌دست می‌دهد و به‌علاوه موجب کمک و بهبود یکپارچگی بین الگوهای مختلف نوشته‌شده، توسط افراد مختلف می‌شود.

ج- رابطه وظایف ذی‌نفع با حوزهٔ امنیتی مرتبط با مدخل

با توجه به نقش ذی‌نفع در چرخهٔ حیات سیستم، رابطه‌اش با مقولات امنیت در این مرحله استخراج می‌گردد.

د- کشف یا تعریف الگوهای مورد نیاز

با توجه به موارد فوق، به‌طور خلاصه فرایند استخراج الگوها می‌تواند به صورت زیر باشد:

۱. استخراج وظایف ذی‌نفع
۲. درک عمومی از مقوله و هستان امنیت
۳. استخراج تکالیف مرتبط با یک وظیفه نسبت به یک

[1] Isolating problems

هستان امنیت

۴. انتخاب الگوهای کمک‌کننده و متناسب از میان الگوها یا تعریف نیاز به الگوها

حاصل کار را به‌صورت خلاصه می‌توان در قالب جدول ۷ نشان داد.

به مثال زیر برای برای نمونه و درک نحوه تدوین خصوصیات یک مدخل می‌پردازیم:

عنوان مدخل: (مدیریت سطح بالا=S_1 ، تهدید=O_1]

جدول ۷. شناسائی الگوها مطابق چارچوب ذی‌نفع – هستان

	وظایف	هستان O_j	الگوهای مورد نیاز P_s
ذی‌نفع S_i	وظیفهٔ 1	تکلیف ۱	
		...	
		تکلیف n	
	...	تکلیف ۱	
		...	
		تکلیف n	
	وظیفه n	تکلیف ۱	
		...	
		تکلیف n	

- مرحلهٔ ۱: استخراج وظایف S1: مدیریت سطح بالا شامل مواردی از قبیل اعضای هیئت مدیره، رئیس، قائم‌مقام و معاونین رئیس، مدیران اجرائی سازمان‌ها، و مدیران ارشد است. مدیریت سطح بالا در خصوص چرخهٔ عمر نرم‌افزار وظایف مهمی از قبیل موارد زیر را بر عهده دارد:
  - تدوین استراتژی شامل مواردی مانند:
    - تولید کامل: تمام اجزاء و مؤلفه‌های سیستم در فرایند تولید نرم‌افزار به‌صورت داخلی و بدون کمک از منابع آماده انجام می‌شود.
    - تولید نسبی (استفاده از کاتس[1])
    - تولید متن باز
    - تولید متن بسته
  - سازماندهی
  - تأمین منابع
  - برنامه‌ریزی
  - نظارت عالیه
- مرحلهٔ ۲: تبیین هستان‌امنیت (در این مثال تهدید): استاندارد ISO تهدید را عامل بالقوه ایجاد یک واقعه می‌داند که می‌تواند (با استفاده از یک آسیب‌پذیری) موجب صدمه رساندن به سازمان شود[۲۷]. تهدید می‌تواند عمدی یا غیرعمدی باشد. هنگامی که تهدید به رویداد بالفعل تبدیل می‌شود، می‌تواند موجب به‌وجود آمدن واقعه شود. مؤسسهٔ استاندارد ملی آمریکا تهدید را هر پیشامد یا موقعیتی تعریف می‌کند که کارایی سازمانی (شامل مأموریت‌ها، کارکردها، وجهه یا شهرت) ، اموال سازمانی، یا افراد را از راه سامانه‌های اطلاعاتی بصورت منفی تحت تأثیر قرار دهد، که این خود می‌تواند از طریق دسترسی غیرمجاز، تخریب، آشکارسازی اطلاعات، تغییر اطلاعات و/یا جلوگیری از دسترسی به سرویس باشد. تهدید همچنین شامل وجود یک منشاء تهدید که به‌طور بالقوه از یک نقطه ضعف امنیتی سامانه می‌تواند استفاده کند نیز می‌شود [۲۸].
- تهدید عبارت است از: عامل بالقوه ایجاد واقعه‌ای که ممکن است منجر به صدمه به سازمان گردد [۲۹].
- مرحله ۳: احصاء تکالیف مرتبط با وظایف ذی‌نفع در قبال مقولهٔ تهدید: تکالیف مرتبط با وظیفهٔ «تدوین استراتژی» برای نمونه در اینجا تشریح شده است:
- همانطور که در بخش تعیین وظایف اشاره شد، تدوین استراتژی می‌تواند شامل تعیین تکلیف در موارد مهمی مانند تولید کامل، تولید نسبی و استفاده از کاتس تولید متن باز، تولید متن بسته، تولید مبتنی بر SPL و... باشد. در ارتباط با این وظیفه دو دسته تکالیف زیر می‌تواند متناسب با مقولهٔ تهدید استخراج گردد:
  - مراقبت از تهدیدات مرتبط با متن‌باز بودن: یک مدیر سطح بالا در چرخه‌حیات نرم‌افزار باید نسبت به تهدیدات استراتژی متن باز بودن واقف بوده و نسبت به مراقبت از آنها تمهیدات لازم را بیاندیشد. متن‌باز بودن سامانه امکان دسترسی هکرها به کُد برنامه‌ها و مشخصات سامانه را فراهم می‌کند. با دسترس‌پذیر بودن کُد سامانه امکان بررسی‌عمیق آسیب‌پذیری‌ها وجود دارد. البته ما عموماً دو دسته مواجهه را شاهد خواهیم بود. مواجهه با هکرهای نجیب[2] و هکرهای نانجیب[3]. یک مدیر سطح بالا باید فرایندهای لازم برای استفاده از دانش هکرهای نجیب را فراهم نماید و فرصت لازم به هکرهای نانجیب را فراهم نکند.
  - مراقبت از تهدیدات مرتبط با کاتس: یک مدیر سطح بالا باید تهدیدات منبعث از تهیه مؤلفه‌های مبتنی برکاتس را به‌دقت درنظرگیرد. عرضهٔ مؤلفه‌ها به این‌صورت،

[1] Commercial Off-The-Shelf (COTS)
[2] Benign Hackers
[3] Malign Hackers

فراوانی عرضهٔ آنها و تعداد زیاد جامعهٔ استفاده‌کننده را شامل می‌شود. این امر موجب می‌شود هکرها نسبت به دامنهٔ گسترده استفاده از آنها در دنیا آگاهی یافته و انگیزه لازم برای نفوذ به سامانه‌ها از این طریق را پیدا نمایند.

- مرحله ۴: الگوهای مورد نیاز: در این بخش متناسب با تکالیف تعریف شده برای وظیفهٔ تدوین استراتژی، باید الگوهای موجود کمک‌کننده به این تکالیف و نیز فضای نیاز به الگوها استخراج گردد.
- در این روش الگوهای غیرموجود یا نیاز آتی به الگوها در فضاهای‌خالی نیز کشف می‌گردد. در این چارچوب با تغییر هستان‌شناسی و ذی‌نفعان، پشتیبانی از انعطاف‌پذیری و ایجاد الگوهای‌جدید به‌راحتی قابل انجام است. تکمیل همه مدخل‌ها می‌تواند موضوع یک پروژه باشد، یا اینکه در پژوهش‌های آتی به تدریج استخراج گردد. هدف ما از انجام یک نمونهٔ مدخل، بیشتر با هدف اثبات مفهوم مورد نظر صورت گرفته‌است. جدول ۸ برای دو تکلیف استخراج شدهٔ فوق ارائه شده است.

جدول ۸. جدول خلاصه برای مدخل ذی‌نفع-هستان مرتبط با وظیفهٔ «تدوین استراتژی»

وظایف ذینفع (مدیریت سطح بالا)	تکالیف مرتبط با هستان تهدید	راهنمای کشف یا تعریف الگوها
تدوین استراتژی-تولید شامل: - تولیدکامل - استفاده‌از کاتس - تولیدمتن‌باز - تولیدمتن بسته - تولید مبتنی برخط تولید	۱-مراقبت از تهدیدات مرتبط با متن‌باز بودن	اهداف (Intents): ۱- تضمین‌حفاظت‌پذیری‌درعین‌افشای(closure)مشخصات سامانه ۲- اعتماد استفاده کننده گان به حفظ خصوصیات امنیتی در فرایندهای استفاده و توسعه مبتنی بر متن باز انگیزه: باز بودن و افشای مشخصات سامانه موجب آسانی استفاده وسیع از سامانه و قابلیت تغییرات محلی و نیز ارزان بودن و بهره مندی از دامنه وسیع توسعه دهندگان و نگهداری کنندگان است. این امر نیازمند اعمال رویه های کارا برای حفظ خصوصیات امنیتی و تضمین اعتماد است. فضای نیازمندی به الگوها: - الگوهای فرایندی از پیشنهاد تا تولید مولفه امن تر و انتشار آن - الگوهای فرایندی کمک کننده به مدیریت پیکربندی امن تر - الگوهای فرایندی انتخاب امن ترین محصول متن باز(open artifact) در بین پیشنهادات مختلف توضیح : یا باید الگوهای موجود از نظرامنیتی بهبود داده شوند یا الگوهای جدیدی تعریف شوند تا نیاز این بخش تامین شود.
	۲-مراقبت از تهدیدات مرتبط با کاتس	اهداف (Intents): ۱- اطمینان از عدم وجود آسیب‌پذیری های شناخته شده ۲- تضمین وجود فرایندهای موثر در مقابله آسیب‌پذیری های روز صفر(zero day) مرتبط با اعمال فضای نیازمندی به الگوها : -الگوهای Input Validation -الگوهای تست جعبه سیاه -الگوهای مرتبط با امنیت وب سرویس

۸- نتیجه‌گیری و پیشنهاد

همانطور که در یکی از تحقیقات انجام گرفته از سال ۱۹۹۷ تا سال ۲۰۱۰ آمده است، تعداد الگوهای امنیت به ۴۰۹ فقره رسیده است و کشف و معرفی آنها همچنان روند روبه‌افزایشی را طی می‌کند. مواجههٔ با آنها برای استفاده‌کنندگان باید سهل و سریع باشد و ارائه دسته‌بندی‌های مختلف و سازماندهی‌های انجام‌گرفته یا در حال انجام، وسیلهٔ مناسبی برای ارائه این تسهیلات است. از دیدگاه‌های مختلف می‌توان به این الگوها نگاه‌کرد و می‌توان گفت نگاه به این الگوها برای مخاطبان مختلف، کامل نشده و هر طرحی که بتواند به‌طور واضح‌تری نحوهٔ استفاده و کاربرد آنها را برای استفاده‌کنندگان که عموماً تخصص زیادی در زمینه امنیت ندارند، ارائه‌کند، موفق‌تر خواهد بود. ما با اتکاء به این نکته توانستیم به شیوه‌ای جدید به مقولهٔ الگوها پرداخته و روشی که همهٔ ذی‌نفعان را در همهٔ حوزه‌های امنیت تحت پوشش قرار دهد، ارائه نمودیم و یک نمونه را برای کمک به درک این روش مورد بررسی قرار داده‌ایم. تعداد زیادی مدخل قابل پژوهش برای فعالیت‌های آینده پیش روی محققان این زمینه باز نمودیم و پیشنهاد ما به علاقه‌مندان این حوزه تکمیل راهنمای انتخاب الگوهای امنیت مطابق با این دیدگاه و نیز تبیین فضای نیاز به الگوها مطابق این چارچوب است.

مراجع

[1] http://www.oxforddictionarries.com,[Online]. Sunday, 12 February 2023

[2] Schumacher, Markus. "Security Patterns and Security Standards." In EuroPLoP, pp. 289-300. 2002.

[3] Xu, Xiwei, HMN Dilum Bandara, Qinghua Lu, Ingo Weber, Len Bass, and Liming Zhu. "A decision model for choosing patterns in blockchain-based applications." In 2021 IEEE 18th International Conference on Software Architecture (ICSA), pp. 47-57. IEEE, 2021.

[4] Marko, Nadja, Joaquim Maria Castella Triginer, Christoph Striecks, Tobias Braun, Reinhard Schwarz, Stefan Marksteiner, Alexandr Vasenev et al. "Guideline for Architectural Safety, Security and Privacy Implementations Using Design Patterns: SECREDAS Approach." In International Conference on Computer Safety, Reliability, and Security, pp. 39-51. Springer, Cham, 2021.

[5] Chifor, Bogdan-Cosmin, Ștefan-Ciprian Arseni, and Ion Bica. "IoT Cloud Security Design Patterns." In Big Data Platforms and Applications, pp. 113-164. Springer, Cham, 2021.

[6] Fenz, Stefan, Thomas Pruckner, and Arman Manutscheri. "Ontological mapping of information security best-practice guidelines." In International Conference on Business Information Systems, pp. 49-60. Springer, Berlin, Heidelberg, 2009.

[7] Yoshioka, Nobukazu, Hironori Washizaki, and Katsuhisa Maruyama. "A survey on security patterns." Progress in informatics 5, no. 5 (2008): 35-47.

[8] http://www.symantec.com/security_response/publications/threatreport.jsp, [Online]. Sunday, 12 February 2023

[9] https://www.us-cert.GOV, [Online]. Sunday, 12 February 2023

[10] https://docs.microsoft.com/en-us/previous-versions/windows/desktop/cc307406 (v=msdn.10], [Online]. Sunday, 12 February 2023

[11] https://www.owasp.org/index.php/Secure_SDLC_Cheat_Sheet, [Online]. Sunday, 12 February 2023

[12] Grance, Tim, Joan Hash, and Marc Stevens. Security considerations in the information system development life cycle. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2004.

[13] National Bureau of Standards, and National Bureau of Standards. Federal Information Processing Standards Publication: Standard Security Label for Information Transfer. US Department of Commerce, National Institute of Standards and Technology, 1994.

[14] Chad Dougherty, Kirk Sayre, Robert C. Seacord, David Svoboda, Kazuya Togashi (JPCERT/CC), Secure Design Patterns, TECHNICAL REPORT CMU/SEI-2009-TR-010 ESC-TR-2009-010, http://cert.org/,[Online], March 2009; Updated October 2009.

[15] Schumacher, Markus. Security engineering with patterns: origins, theoretical models, and new applications. Vol. 2754. Introduction section, Pagee 6, Springer Science & Business Media, 2003.

[16] Barabanov, Alexander, and Denis Makrushin. "Security audit logging in microservice-based systems: survey of architecture patterns." arXiv preprint arXiv:2102.09435 (2021).

[17] Blakley, Bob, and Craig Heath. "Security design patterns technical guide–Version 1." Open Group (OG] (2004).

[18] Schumacher, Markus, Eduardo Fernandez-Buglioni, Duane Hybertson, Frank Buschmann, and Peter Sommerlad. Security Patterns: Integrating security and systems engineering. John Wiley & Sons, 2013.

[19] Steel, Christopher, Ramesh Nagappan, and Ray Lai. "The alchemy of security design methodology, patterns, and reality checks." Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management, Prentice Hall 1088 (2005).

[20] Hafiz, Munawar, Paul Adamczyk, and Ralph E. Johnson. "Organizing security patterns." IEEE software 24, no. 4 (2007): 52-60.

[21] Wiesauer, Andreas, and Johannes Sametinger. "A security design pattern taxonomy based on attack patterns." In International Joint Conference on e-Business and Telecommunications, pp. 387-394. 2009.

[22] "https://capec.mitre.org," [Online]. Sunday, 12 February 2023

[23] Bunke, Michaela, Rainer Koschke, and Karsten Sohr, "Application-Domain Classification for Security Patterns," In Proceedings of the International Conferences on Pervasive Patterns and Applications, IARIA Conferences. XPS , 2011.

[24] Dove, Rick, and Laura Shirey. "On discovery and display of agile security patterns." In Conf Syst Eng Res, Stevens Institute of Technology, Hoboken, NJ. 2010.

[25] "www.ISC2.org," [Online]. Sunday, 12 February 2023

[26] Laplante, Philip A. What every engineer should know about software engineering. CRC Press, 2007.

[27] Huang, Chien-Cheng, Kwo-Jean Farn, and Frank Yeong-Sung Lin, " A Study on Information Security Management with Personal Data Protection," In 2011 IEEE 17th International Conference on Parallel and Distributed Systems (pp. 624-630]. IEEE, (2011, December).

[28] Pub,FIPS(FIPS Publication 200). "Minimum security requirements for federal information and information systems." (2005].

[29] A. I. Standard, " In InformationTechnology-security Techniques-Code of Practice for Information Security Controls", (AS ISO/IEC 27002: 2015), 2015.