International Journal for Scientific Research (IJSR)
المجلة الدولية للبحوث العلمية
Vol. (2), No. (4) April 2023 الإصدار (2)، العدد (4)

نهج مقترح لاختبار الأمان لمشروعات البرمجيات القائمة على منهجية Scrum

إيهاب محمد عبد الوهاب
ماجستير نظم المعلومات، الأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري، مصر

ملخص البحث:

تتميز الأساليب الرشيقة لتطوير البرامج Agile Software Development بالتكيف مع متطلبات العملاء المتغيرة وتسليم المنتجات البرمجية في وقت أقل من الطرق التقليدية. وتعد Scrum واحدة من أكثر أساليب التطوير الذكية شيوعاً والتي تستخدم في شركات برامج الكمبيوتر الكبيرة مثل HP، Yahoo، Google... إلخ. تحقق Scrum مزايا من حيث الوقت والتكلفة لكنها قد تفشل في إنتاج برامج ذات خصائص أمنية جيدة. قد يرجع ضعف خصائص الأمان إلى عدم وجود معيار أو إطار أمان واضح يمكن اعتماده منذ بداية مشروع تطوير البرمجيات. وبالإضافة إلى ذلك، ذكرت عدة دراسات أن معظم نقاط الضعف الأمنية التي تواجدت في البرمجيات أثناء عمليات التطوير تسبب التهديدات والجرائم السيبرانية. وتقترح الورقة نهجاً لاختبارات الأمان في مشروعات تطوير البرمجيات المعتمدة على منهجية التطوير Scrum، ويركز النهج المقترح على اختبار أمن Security Testing البرمجيات. وعلاوة على ذلك، يمكن للإطار المقترح أن يساعد فريق العمل على تعزيز أمن البرمجيات الناتجة من هذه المشروعات، والحد من مخاطر التهديدات، وخفض تكلفة إصلاح الأخطاء البرمجية.

الكلمات المفتاحية: اختبار الأمان، المراقبة، تهديدات الأمان، الجرائم الإلكترونية، نقاط الضعف، الطرق الرشيقة، مشروعات البرمجيات.

International Journal for Scientific Research (IJSR)

المجلة الدولية للبحوث العلمية

Vol. (2), No. (4)

April 2023

الإصدار (2)، العدد (4)

A Proposed Approach for Security Testing of Scrum-based Software Projects

Ehab Mohamed Abdel Wahab

Master of Information Systems, Arab Academy for Science, Technology and Maritime Transport, Egypt

Abstract:

Agile software development methods are characterized by adapting to changing customer requirements and delivering software products in less time. Scrum is one of the most common agile development methods that are used in large software companies like HP, Yahoo, Google, etc. Scrum achieves advantages in time and cost, but it may fail in producing software that has good security properties. The weakness in security properties may be due to the lack of a clear security standard or framework that can be adopted from the beginning of the project. In addition, several studies mentioned that most security vulnerabilities that were left in software during development processes cause threats and cybercrimes. The paper proposes a Scrum security approach that focuses on testing the security of software in Scrum projects. Moreover, the proposed approach can help the team to enhance the security of the software product, minimize the risk of threats, and reduce the cost of fixing software bugs.

Keywords: Security Testing, Scrum, Security Threats, Cybercrime, Vulnerabilities, Agile Methods, Software Projects.

1. مقدمة

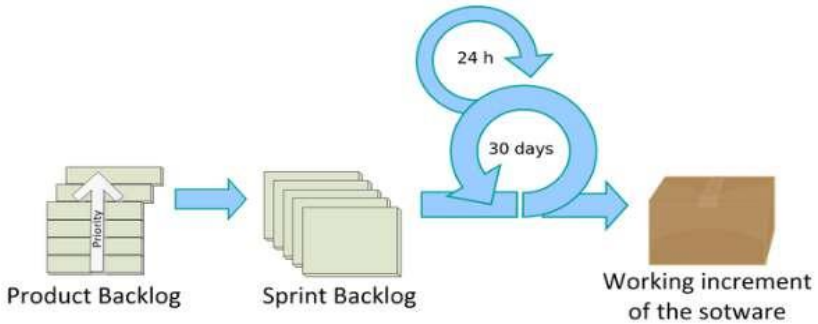
أصبحت المنهجيات الرشيقة أكثر الطرق شيوعاً التي تستخدمها شركات البرمجيات في مشروعات تطوير البرمجيات. وتهدف إدارة مشروعات تطوير البرمجيات أساساً إلى دعم عمليات تطوير البرمجيات بالتكلفة والوقت المناسبين وتحقيق الجودة العالية التي تلبي احتياجات العملاء وقت التسليم. ويدعي المدافعون عن

الأساليب الرشيقة أنه يمكن إنجاز هذه الأساليب باستخدام عمليات تطوير برمجيات تتكيف باستمرار مع [1]:

- الخبرات والمهارات الجماعية للمطورين، بما في ذلك الخبرة والمهارات المكتسبة حتى الآن في تطوير مشروع.
- التغييرات في متطلبات البرمجيات.
- التغيرات في بيئات التطوير والتشغيل المستهدفة.

وتستخدم الأساليب الرشيقة للتطوير للحد من مخاطر فشل المشروعات؛ إلا أنها تحتاج إلى اتباع عدة قواعد مثل: المرونة، مستندات أقل، تفاعلات أكثر بين أعضاء فريق العمل، والاتصال الجيد مع العميل والمستخدم النهائي للبرمجيات الناتجة. وأكثر الطرق استخداماً على نطاق واسع هي: "Scrum" و"Hybrid Framework" التي تجمع بين "Scrum" و"Extreme Programming". وقد وضع شوابر وسثرلاند Scrum وصف ذلك في دليل Scrum [2].

Scrum هو إطار لتطوير البرمجيات الرشيقة و يستخدم أساسا في تطوير البرمجيات بشكل متكرر وتدريجي كما هو مبين في الشكل (1). يتمثل الهدف الرئيسي من Scrum في متطلبات العملاء التي يمكن تغييرها بسرعة أثناء تطوير البرامج. ويمنح هذا السباق فريق التطوير بعض المرونة فيما يتعلق بالوظائف المنفذة في الدورة sprint [3]. كل دورة sprint تبدأ بتخطيط متطلبات الدورة بالمعايير المعروفة حيث يقوم العميل بمراجعة المتطلبات وتحديد أولوياتها. وتنتهي الدورة sprint بمراجعة هذه الدورة، حيث تمثل هذه المراجعة بوابة لجودة المخرجات. ولها دور هام جدا في إدارة مخاطر أمن المنتج [4].



الشكل (1): إطار العمل Scrum [3]

وقد لوحظ من خلال التقارير والتجارب المنشورة أن أمن البرمجيات لا يأخذ الأهمية المطلوبة عند تطوير البرمجيات باستخدام الأساليب الرشيقة. نظرا لأن أسلوب Scrum يركز على تلبية الاحتياجات بسرعة فائقة للوفاء باحتياجات العملاء المباشرة وأمن البرمجيات هو أحد احتياجات العملاء، فمن المهم عدم إغفاله [5]. في عالم اليوم الشديد الترابط، حيث توجد متطلبات تنظيمية وخصوصية قوية لحماية البيانات الخاصة بالبرامج. كما يجب أن يعامل أمن البرمجيات كأولوية عالية وأن يفحص بشكل جيد. وقد أشار بعض الباحثين والممارسين تلك المشكلة الهامة في البرمجيات - أمن البرمجيات [6]. قد يعتبر التأمين متطلباً غير عملي في عمليات التطوير المرنة لمطوري Scrum. وقد سلطت دراسات عديدة الضوء على وجود نقاط ضعف في البرمجيات قد تؤدي في نهاية المطاف إلى تهديدات وتسبب جريمة إلكترونية. كما بدا الأمر للمطورين متعارضا بين الحفاظ على جودة مقبولة للبرمجيات التي تعالج قضايا الأمن أو سرعه تنفيذ البرامج وفقا للجدول الزمني للمشروع [5].

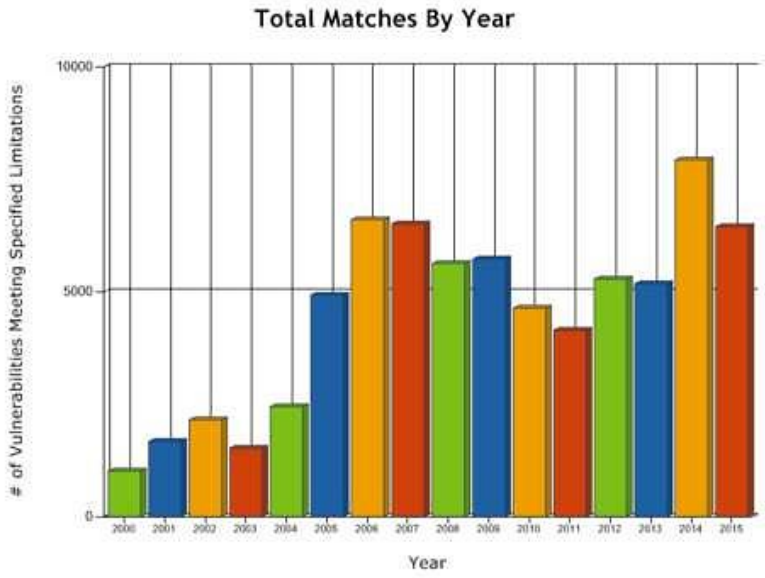
الجزء المتبقي من هذه الدراسة تم تنظيمه في ستة أقسام، حيث يعرض القسم (2) نقاط الضعف والتهديدات الخاصة بأمن البرمجيات. ويعرض القسم (3) تكاليف اختبار أمن البرمجيات والجرائم السيبرانية. ويبين القسم (4) الأعمال والدراسات السابقة ذات الصلة بموضوع الدراسة، كما يعرض أثر دمج أمن البرمجيات في المنهجية الرشيقة Scrum. وفي القسم (5) يقترح الباحث نهج اختبارات الأمان ويصف هذا النهج. ويعرض القسم (6) خلاصة الدراسة وأهم الموضوعات التي يمكن للباحثين في هذا المجال التركيز عليها في المستقبل.

2. نقاط الضعف والتهديدات الخاصة بأمن البرمجيات

إن قابلية البرامج بوجود ثغرات تشكل نقطة ضعف في المنتج قد تسمح للمهاجم باختراق البرنامج (المنتج) أو إتاحة الخدمة أو سريته [7]. وأحد الأسباب المباشرة لضعف الحواسيب هو تعقيد البرمجيات الذي يعني وجود وظائف أكثر ووجود تعليمات برمجية أكثر وبالتالي أخطاء في التعليمات البرمجية. ويوفر التعقيد كل الفرصة للإخفاء المهاجمين والتسبب في الإخفاق الأمني للبرمجيات [8].

ويبدو أن الثغرات البرمجية القابلة للتأثر أكثر تعقيداً من كتابة البرمجيات الخاطئة [8]. ويرتبط العامل الآخر الذي قد يتسبب في ضعف البرامج بمرحلة التطوير عند قيام المطورين بإنتاج كود جديد أو إدخال كود تم تطويره في الأصل لتطبيقات أخرى. على سبيل المثال: تضمين كود برمجي متاح مجاناً على الانترنت بدون معرفة مصدره، أو استخدام مكونات برمجية من مصادر خارجية مع عدم كفاية اختبارات الأمان وتحليل

المخاطر الأمنية. وبالتالي، فإن هذه العوامل تزيد من المخاطر والتهديدات [21]. ووفقا لقاعدة البيانات الوطنية لنقاط ضعف البرمجيات في الولايات المتحدة، بلغ عدد حالات الضعف المكتشفة في عام 2015 كما هو مبين في الشكل (2) [9].



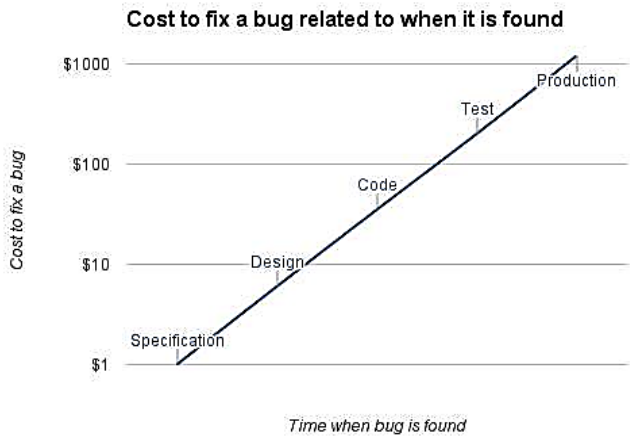
الشكل (2): عدد مواطن الضعف في السنة في الفترة من كانون الثاني/يناير 2000 إلى كانون الأول/ديسمبر 2015 في وزارة الصحة الوطنية [9]

وقد أظهرت النتيجة مقارنة بالسنوات السابقة أن عدد حالات الضعف لا يزال مرتفعا جدا. بالرغم من وجود الكثير من تقنيات الأمان وجود طبقات مختلفة للحماية. وأدى ذلك إلى استنتاج أن أمن البرمجيات ينبغي أن يكون وظيفة رئيسية في تطوير البرمجيات. وبعبارة أخرى، فإن التعامل مع السبب الحقيقي للتهديدات التي تمت ملاحظتها بشكل صحيح سيوفر الوقت والمال والمخاطر.

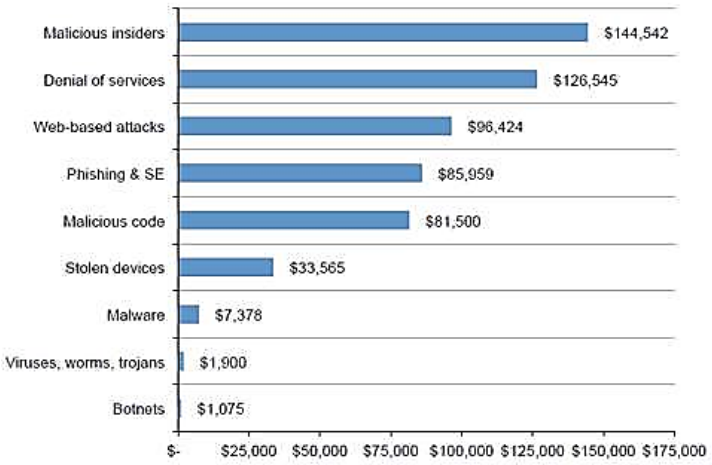
3. تكاليف اختبار أمن البرامج والجرائم الإلكترونية

أحد أهداف اختبار البرمجيات هو إيجاد الثغرات الأمنية. وترتبط تكلفة إصلاح الخلل ارتباطاً كبيرا بالمكان الذي يتم فيه العثور على الخلل أثناء العملية كما يمكن رؤيته في الشكل (3) [10]. وتظهر دراسة أخرى أعدها

معهد بونيمون في الولايات المتحدة أن برمجيات الشفرات الخبيثة هي المشكلة الأكثر تكلفة بالنسبة للشركات الأمريكية [11]. أما البلدان التي تتحمل أعلى التكاليف المرتبطة بالهجمات التي تمنع تقديم الخدمات فهي المملكة المتحدة وأستراليا. تعتبر البرمجيات الخبيثة أكثر تكلفة في الاتحاد الروسي. وفي معظم البلدان، وتعتبر شبكات الإنترنت أقل أنواع الهجمات تكلفة، وأكثر الجرائم الحاسوبية تكلفة هي تلك التي يتسبب فيها مخبرون خبيرون، وحرمان من الخدمات، والهجمات القائمة على شبكة الإنترنت على النحو المبين في الشكل (4) [11]. ويتطلب التخفيف من هذه الهجمات تكنولوجيات تمكينية مثل حلول اختبار أمن التطبيقات، ونظام SIEM، ونظم منع الاختراق IPS [11]. وينبغي إعطاء الأولوية لاختبار البرمجيات للتقليل إلى أدنى حد من الجريمة السيبرانية والآثار المترتبة عليها.



الشكل (3): تكلفة إصلاح الأخطاء المرتبطة بمكان العثور عليها. هذا الرقم هو نسخة معدلة من النسخة الأصلية من [10].



الشكل (4): متوسط التكلفة السنوية للجرائم السيبرانية المرجحة حسب تواتر الهجمات، وجهة نظر موحدة، n = 252 شركة منفصلة [11]

4- الأعمال ذات الصلة

وخلال السنوات الأخيرة، قدمت عدة دراسات تتضمن تقنيات مقترحة لتعزيز أمن البرمجيات في إطار مشاريع التطوير الرشيدة وقياس فعالية الأنشطة الأمنية في المشاريع الرشيقة.

- تقترح سونيا وآخرون [12] نهجاً جديداً يوفر قياساً كمياً لمرونة أنشطة الأمن من حيث درجة سرعة الاستجابة. تحدد درجة توافق نشاط الأمان مع عملية المرونة. وعلاوة على ذلك، قدم تحليل مقارن للأنشطة الأمنية فيما بينها في سياق عامل الكفاءة في إدارة الكوارث وإزالة المخاطر. RREF هو تقييم لمدى فعالية نشاط أمني لإزالة الخطر ومساعدة مطور خلال تطوير البرمجيات في تقرير أي نشاط أمني مفيد من الآخر للتكامل.
- Chowdhury et al. [13] أثبت من خلال إشارة تجريبية أن الكيانات البرمجية المعقدة والمقترنة وغير المتماسكة أقل أمنا بشكل عام.
- يونغي وآخرون [8] تحقق من إذا كانت فرضية مقاييس التعقيد صحيحة. تظهر النتائج الأولية أن مقاييس التعقيد التسعة لها ارتباط ضعيف (ρ=0.30 على أفضل تقدير) مع مشكلات الأمان لمحرك Mozilla

JavaScript. وحسب نتائج الدراسة، فإن الاستنتاج بأن تعقيد البرمجيات هو عدو لأمن البرمجيات. ومع ذلك، يبدو أن البرمجة الضعيفة أكثر تعقيدا من التعليمات البرمجية الخاطئة.

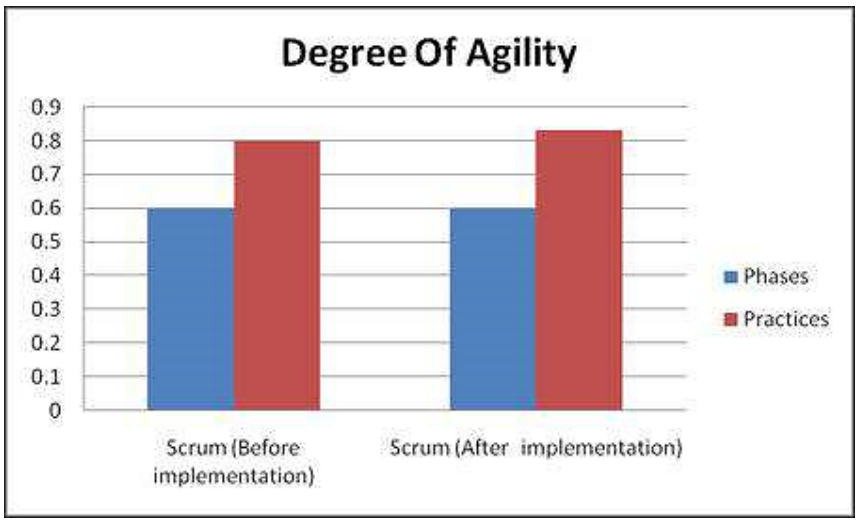
- ويقترح C.Pohland وآخرون [14] نموذجا من "Secure Scrum" يختلف عن إطار "Scrum" مع تركيز خاص على تطوير البرمجيات الآمنة في جميع مراحل عملية تطوير البرمجيات. ويركز النموذج المقترح على تنفيذ المسائل المتعلقة بالأمن دون الحاجة إلى تغيير عملية المراقبة الأساسية أو التأثير على ديناميكيات الفريق. وتسمح تقنية Secure Scrum حتى لغير الخبراء في مجال الأمان باكتشاف مشكلات الأمان، وتنفيذ ميزات الأمان، والتحقق من تنفيذ عمليات التنفيذ. ويظهر اختبار Secure Scrum أن مستوى أمن البرامج التي تم تطويرها باستخدام Secure Scrum أعلى من مستوى أمان البرامج التي تم تطويرها باستخدام Standard Scrum.
- A.Jøsang وآخرون [15] توفير طريقة مرنة لتصميم البرامج الآمنة .تتضمن الطريقة مراجعة الأمان في مرحلة دورة التكرار الدوري حيث يتم تقييم النسخة الحالية من النظام. يتطلب أيضا أن يكون أعضاء الفريق قد حصلوا على تعليم وتدريب أمنيين كافيين.
- S.Jürimäe et al [16] مقارنة الفروق بين SDL و CLASP و Touchpoints والمجال الذي يغطيه لتحسين أمن البرامج حيث أنه لا يغطي سوى مرحلة واحدة من عملية التطوير.
- D. Mougouei et al [17] عرض نسخة معززة أمنياً من Scrum، أي Secure Scrum (S-Scrum) لدمج تحليل الأمان وأنشطة التصميم في عمليات Scrum. وقد عدلت المنهجية المقترحة عمليات التدقيق لمراعاة إجراءات الوثائق التي تعكس الجوانب الأمنية لخدمة الويب المستهدفة. وعلاوة على ذلك، تهتم المنهجية المقترحة بالأمن وكذلك بالاحتياجات المتغيرة أثناء التخطيط متطلبات المتغيرة للإصدار ودورة إنشاء البرمجيات (Sprint).
- آي. غاني في [18] قد أثبت نجاح الأبحاث التي تم إجراؤها على نموذج الفحص المحسن الذي اقترحوه، ويتم تقييم هذا النموذج في مراحل المتطلبات والتطوير والاختبار. وقد أظهرت النتائج أن المرونة تتحسن في حالة تنفيذ الأعمال المتأخرة في مجال الأمن، مما يعني أن المرونة لا تتأثر سلبا في حالة إضافة هذا النموذج إلى نموذج الفحص.

- كما اقترح السيد تومانيك وآخرون [2] إمكانية إثراء Scrum لإطار تطوير البرمجيات الذكية من خلال النظر في اختبارات الاختراق ومتطلبات الأمان ذات الصلة خلال دورة حياة تطوير البرمجيات. M.Tomanek et al. [2] تطبيق المعارف والخبرات المستمدة من أعمالهم السابقة التي ركزت على تطوير منهجية اختبار اختراق نظام المعلومات الجديدة PETA مع التركيز على استخدام COBIT 4.1 كإطار لإدارة هذه الاختبارات، وعلى الأعمال السابقة التي ركزت على تكييف إطار إدارة المشروع PRINCE2 مع Scrum.
- أ.بروستروم وآخرون [10] عرض التحقيق في الكيفية التي يمكن بها مواصلة إدماج الأمن في عملية تطوير رشيقة في تحليل شفرة المصدر والاختبار التلقائي للأمن. وتنتج الورقة البحثية دليلاً للمطورين يخبرهم عن كيفية إعداد اختبارات أمنية، ومساعدتهم على فهم التداعيات والمخاطر الأمنية، وتوضح لهم كيف يمكنهم تخفيف مخاطر معينة للحد من نقاط الضعف.
- تقترح سونيا وآخرون [19] نهجا يسعى إلى مساعدة دوائر الهندسة الأمنية في قياس فعالية كل نشاط أمني على حدة. ويستنتج من الورقة التي تستخدم هذا النموذج أن قياس فعالية النشاط الأمني سيساعد صانعي القرارات أيضا بتوفير مبادئ توجيهية لاختيار الضمان نشاط يمكن أن يكون أكثر فائدة للاندماج في عملية التنمية.

وتثبت الدراسات السابقة أنه يمكن تحسين أمن البرامج في Scrum. كما أن الأمن متضمن في عملية تطوير رشيقة دون أن يؤثر سلباً على منهجية مرنه ورشيقة.

وتناقش بعض الدراسات مؤخراً العلاقة بين الأمن كوظيفة ومنهجية تطوير البرمجيات الرشيقة (Agile). ومعروف لمن يهتم بأمن البرامج قد يتعارض مع إحدى السمات الأساسية في منهجية Agile للتسليم السريع والتغييرات السريعة والمرونة. واستناداً إلى الشكل (5)، تحسنت درجة المرونة في الممارسات من 0.8 (قبل التنفيذ) إلى 0.83 (بعد التنفيذ) [18]. وقد ظهر التحسن بعد التنفيذ مما يدل على أن تراكم الأمن لا يضيف أي تأخير إلى السرعة، أو المرونة، أو التعلم، أو الاستجابة إذا كان الضمان مطبقا كجزء من طريقة "Scrum". والتنفيذ الفعلي ذو صلة ويمكن تنفيذه دون أي خوف من التأثير سلبا [18]. كما اقترحت سونيا وآخرون [19] نهجا يسعى إلى مساعدة دوائر الهندسة الأمنية في قياس فعالية كل نشاط أمني على حدة. وتقدم

دراسات أخرى ورقات مماثلة تتعلق بأساليب رشيقة أخرى مثل XP، FDD، DSDM، وما إلى ذلك. وستشجع هذه الدراسات المطورين على اعتبار الضمان شرطا وظيفيا دون خوف من تأخير التسليم.



الشكل (5): درجة المرونة

5. نهج مقترح لاختبارات أمن مشاريع البرمجيات المعتمدة على SCRUM

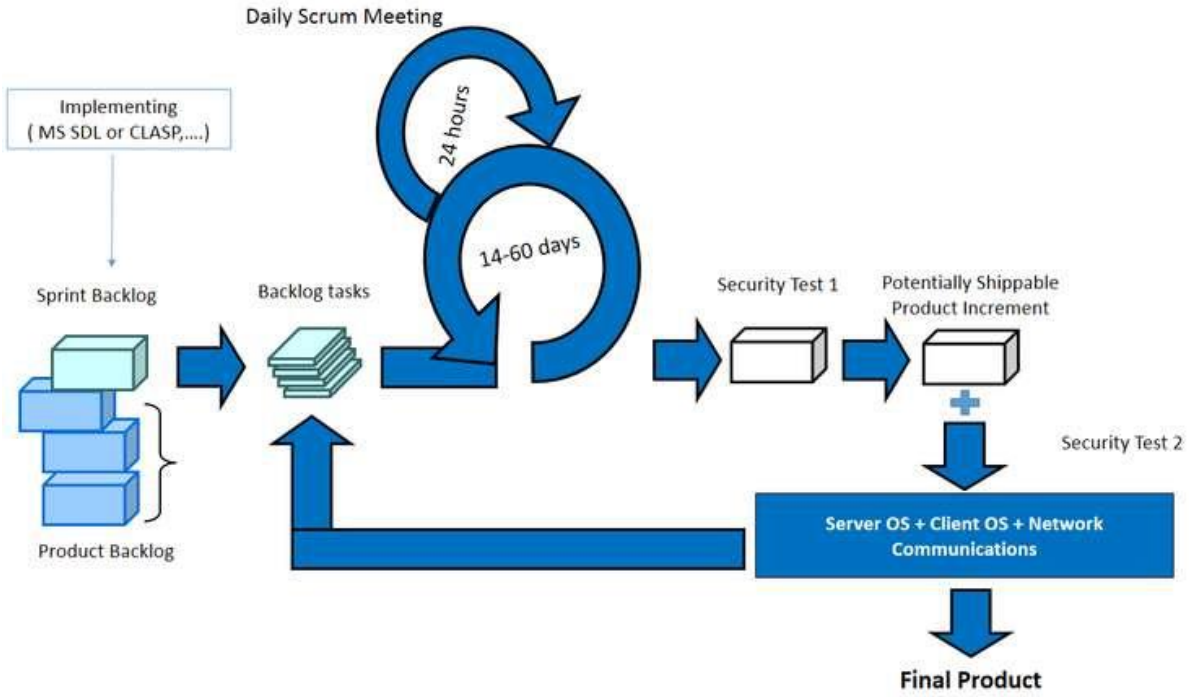
هناك بعض الافتراضات: التنفيذ (MS SDL أو OWASP CLASP أو Cigital's Security Touchpoints... الخ). من الضروري اختبار أمن البرامج لمشروع تطوير البرامج الذكية.(SCRUM) هناك ممارسات مقترحة مختلفة في عملية تطوير الأمن مثل MS SDL و OWASP CLASP و Cigital's Security Touchpoints... إلخ. وتقترح هذه الممارسات لتعزيز أمن منتجات البرمجيات بالتوصية بمجموعة من الأنشطة الأمنية. غير أن هذا سيساعد المطور على تحديد النشاط الذي يوصي به تبعا لمرحلة التطوير. وتثبت وجهة النظر العملية أن ممارسات الأمان (MS SDL ، OWASP clasp) غير كافية وقد ظهر العديد من أخطاء الأمان في البرنامج ربما لأن الاختبار الذى تم في بيئة مثالية (مستقلة) دون اختباره في بيئة حياة الأعمال للنظر في عوامل مختلفة مثل (نظام التشغيل Server و Client OS و Network Commination... الخ). هذا

IJSR, VSRP Publishing, UK ISSN 2755-3418 (Online)
Website: https://ijsr.vsrp.co.uk

وتتطلب العوامل وجود إعداد معين للمحافظة على أمان البيئة بأكملها ويجب أخذه بعين الاعتبار في تطوير البرامج مثل استخدام منافذ معينة في جهاز اتصال الشبكة (جدار الحماية، نظام منع الاختراق (IDS)، جهاز التوجيه ... إلخ) أو بعض إعدادات الأمان في نظام تشغيل جانب العميل (أجهزة الكمبيوتر، الجهاز المدمج ...). لذا؛ فمن دون اختبار التأثيرات المترتبة على هذه العوامل، لن يصبح بوسعنا أن نرى العديد من التهديدات في النظام بالكامل.

يساعد اختبار البرامج داخل بيئة العمل على اكتشاف نقاط الضعف التي قد لا تظهر. على سبيل المثال، المستخدم الضار الذي يستخدم (الباب الخلفي أو الخطأ) للبرنامج للقيام بعمليات غير قانونية أو تعليمات برمجية ضارة التي تسبب تهديدا من المستخدم الخارجي أو الداخلي (محتوى نشط خبيث، باب خلفي).

يفترض إطار العمل المقترح كما هو مبين في الشكل (6) استخدام إحدى ممارسات أمان البرامج على سبيل المثال (OWASP CLASP، Cigital's Security Touchpoints، SAN...). يتم استخدام إطار الأمان في كل دورة ويتم تطبيق اختبار الأمان باستخدام برامج1 (أدوات ثابتة أو ديناميكية) قبل أن نقوم بدمج منتج البرمجيات المحتمل في بيئة العمل. في هذه المرحلة تم إجراء اختبار الأمان 2 إما بشكل ثابت أو ديناميكي. ويفحص الإطار المقترح أي ضعف أمني أو أي خلل يظهر داخل بيئات العمل. ونتيجة لذلك، سيتم التعامل مع هذه العناصر والنظر فيها مرة أخرى في مهمة التأخيرات الدورية أو إصدار نهائي من البرنامج. وينصح أيضا بتطبيق دراسة هيكلية أمنية لتقييم تصميم التطبيق وبيئة نشره (مكونات النظام الرئيسية).



الشكل (6): إطار العمل الأمني لتطوير برامج Scrum

استناداً إلى هذه الدراسة الاستقصائية يمكن أن تكون مفيدة في تحديد المجالات التي تشكل أكبر خطر على الأمن، ونقاط الضعف في تصميم التطبيقات تمكنك من تحديد أولويات المنطقة التي تحتاج إلى تحليل أمني أكثر تعمقا. يمكن تحقيق تقليل المخاطر من خلال تعديل التصميم ليتوافق مع أفضل الممارسات الأمنية. والاختبارات الأمنية المقترحة هي:

- اختبار الأمن 1 تصف النقاط التالية 12 فئة فرعية لمنهجية اختبار اختراق التطبيقات التي تستخدم كل أو جزء من هذه الفئات الفرعية [20]:
  - المقدمة والأهداف.
  - جمع المعلومات.
  - اختبار إدارة التهيئة والتطبيق.
  - اختبار إدارة الهوية.

- اختبار المصادقة.
- اختبار الاعتماد.
- اختبار التحقق من صحة الإدخال.
- حدث خطأ أثناء المعالجة.
- التشفير.
- اختبارات منطق إجراءات العمل.
- اختبار من جانب العميل.

- اختبار الأمان 2: يكتشف "اختبار اختراق الشبكة" (تلقائي أو يدوي) مكامن الضعف في الشبكة التي يمكن مهاجمتها ووضعها في موضع التهديد. سيغطي الاختبار كل أو جزء من العناصر التالية:
  - نظام التشغيل.
  - تطبيق الخادم.
  - نظام منع الاختراق.
  - الموجه.
  - جدار الحماية.
  - خدمات الشبكة.
  - إدارة الوصول.

بعد إجراء اختبارات الأمان، تظهر النتائج معرفة مفصلة تماماً حول نقاط الضعف وترتيبها بحسب تأثير مخاطر الأعمال. أيضاً، اقترح إستراتيجية واضحة للتخفيف لتعزيز أمن البرامج والحفاظ على بيئة الأعمال آمنة.

6. خلاصة الدراسة والأعمال المستقبلية

وتكشف هذه الورقة البحثية أنه يجب إيلاء اهتمام كبير لمسألة اختبارات الأمن أثناء تنفيذ مشروعات البرمجيات المعتمدة على منهجية التطوير الرشيقة Scrum. ويوفر الإطار المقترح أداة فعالة في استعراض وتعزيز مشروعات Scrum لإنتاج برمجيات مأمونة. يقوم إطار العمل بعرض المخاوف الأمنية في مراحل ومنظورات مختلفة بالنظر إلى نظام البرامج لتجنب أي تهديد أمني. ويمكن للإطار المقترح أن يساعد الفريق على تعزيز أمن منتج البرامج، والحد من مخاطر التهديدات، وخفض تكلفة إصلاح الأخطاء البرمجية.

من الأعمال المستقبلية المقترحة: أنه يمكن العمل على تحديد إطار أكثر تفصيلاً لتزويد فريق عمل Scrum بإطار تدريجي. وبالإضافة إلى ذلك، يمكن استخدام مجموعة من مقاييس الأمن لتقييم نوعية تنفيذ أمن منتجات البرمجيات في مشاريع Scrum.

7. المراجع

[1] D. Turk, R. France and B. Rumpe, "Assumptions Underlying Agile Software-Development Processes", Journal of Database Management, vol. 16, no. 4, pp. 62-87, 2005.

[2] M. Tomanek and T. Klima, "Penetration Testing in Agile Software Development Projects", International Journal on Cryptography and Information Security, vol. 5, no. 1, pp. 01-07, 2015.

[3] K. Schwaber, and J. Sutherland, "The scrum guide, The Definitive Guide to Scrum: The Rules of the Game", (1991st–2013th Ed.). Scrum.org

[4] A. Vaha-Sipila, "Product Security Risk Management in Agile Product Management", Stockholm, Sweden, 2010.

[5] "Agile Security Successful Application Security Testing for Agile Development", white paper, Veracode, Inc, 2010.

[6] I. Ghani and Izzaty Yasin, "Software Security Engineering in Extreme Programming Methodology: A Systematic Literature Review", Sci.Int. (Lahore), 25 (2), P.P. 215-221, 2013.

[7] Microsoft MSDN, "Definition of a Security Vulnerability", 2016. [Online]. Available: https://msdn.microsoft.com/en-us/library/cc751383.aspx. [Accessed: 13- Jan- 2016].

[8] Y. Shin and Laurie Williams, "Is Complexity Really the Enemy of Software Security?", ACM QoP 08, October 27 2008

[9] "NVD - Statistics Results", 2016. [Online]. Available: https://web.nvd.nist.gov/view/vuln/statistics-results?adv_search=true&cves=on&pub_date_start_month=0&pub_date_start_year=2000&pub_date_end_month=11&pub_date_end_year=2015. [Accessed: 13- Jan- 2016].

[10] A. Broström, "Integrating Automated Security Testing in the Agile Development Process", KTH Royal Institute of Technology, Stockholm, Sweden, 2015.

[11] "2015 Cost of Cyber Crime Study: Global", by Ponemon Institute, October 2015.

[12] Sonia and Singhal, "Integration Analysis of Security Activities from the Perspective of Agility", International Conference on Agile and Lean Software Methods, Bengaluru, India, February 17–19 (2012).

[13] I. Chowdhury, M. Zulkernine, "Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities", Journal of Systems Architecture, vol. 57, Issue 3, pp. 294–313, March 2011

[14] C. Pohland, H. Hof, "Secure Scrum: Development of Secure Software with Scrum", arXiv preprint: 1507.02992, 2015.

[15] A. Josang and M. odegaard, E. Oftedal, "Cybersecurity Through Secure Software Development", 9th World Conference on Information Security Education (WISE9), Hamburg, May 2015.

[16] S. Jurimae, "A Literature Survey of the Development Processes for Secure Software", Bachelor's Thesis, Faculty of Mathematics and Computer Science, University of Tartu 2015.

[17] D. Mougouei, N. Fazlida, M. Sani and M. Almasi, "S-Scrum: A Secure Methodology for Agile Development of Web Services", World of Computer Science and Information Technology Journal (WCSIT), ISSN: 2221-0741, Vol. 3, No. 1, PP. 15-19, 2013.

[18] I. Ghani1, Z. Azham and S. Jeong, "Integrating Software Security into Agile-Scrum Method", Ksii Transactions on Internet and Information Systems, vol. 8, no. 2, February 2014.

[19] Sonia and Singhal, "An Evaluation Approach: Measuring Effectiveness of Security Activities", ICDMW 2013, PP. 202–210, 2013.

[20] Owasp.org, "Web Application Penetration Testing - OWASP", 2016. [Online]. Available: https://www.owasp.org/index.php/Web_Application_Penetration_Testing. [Accessed: 19- Jan- 2016].

[21] Cigital, "Third Party Security for Apps & Software", 2016. [Online]. Available: https://www.cigital.com/solutions/by-security-need/third-party-security/. [Accessed: 01- Feb- 2016].