<https://resources.whitesourcesoftware.com/blog-whitesource/how-to-secure-your-sdlc-the-right-way>

**كيف تقوم بحماية دورة حياة البرمجيات SDLC بالطريقة الصحيحة؟**

الزيادة الملحوظة في نسبة التهديدات و المخاطر أدت إلى وضع بعض ضوابط أمن العلومات للتأكد من حماية بيانات المنظمات من المهاجمين. ولهذه الأسباب تم إدراج أمن دورة حياة البرمجيات. ينبغي على المنظمات أن تضع بعين الاعتبار حماية بيانات العملاء في جميع مراحل تطوير الأنظمة بشكل يوازي تقديم خدمات مبتكرة وجديدة.

ينبغي علينا اختيار المقاييس والأدوات المناسبة التي تمكننا من الحفاظ على أمن جميع الإجراءات في تطوير البرمجيات.

تهديدات أمن البرمجيات

خلال السنوات الماضية كثرت التهديدات على طبقة التطبيقات على نظام OSI ، كما أن نظام تأمين التطبيقات OWASP يقدر بأن نحو ثلث تطبيقات الويب تحتوي على نقاط ضعف أمنية. وفي تقرير WhiteHat "تقرير احصائيات أمن المواقع الإلكترونية ٢٠١٥" ٨٦٪ من المهاجمين يستخدمون الثغرات الأمنية ذاتها سهلة الوصول للحصول على وصول غير مصرح له لشبكة الجهة المستهدفة.

وعند النظر لواحدة من أحدث الهجمات الإلكترونية على شركة Equifax والتبعات المترتبة عليها، نجد أنه مازالت بعض المنظمات تتباطأ في تطبيق استراتيجية تأمين دورة حياة البرمجيات.

كيف نتأكد من أن دورة حياة البرمجيات آمنه؟

أحد أهم المبادئ هي ادخال أمن المعلومات في مراحل متقدمة. وهذا يشمل ادخال إجراءات وأدوات أمن المعلومات خلال جميع مراحل تطوير البرمجيات بداية من أولى المراحل، حيث أنه سيضمن للمنشأة حفظ الوقت والجهد والمال في المراحل المتأخرة لأنه كلما اكتشفت الثغرة في مراحل متأخرة ستكون كلفة اغلاقها أكبر وأصعب.

ادخال أمن المعلومات في جميع مراحل تطوير البرمجيات

كل مرحلة من مراحل تطوير البرمجيات لها متطلباتها وأدوتها للحفاظ على أمن المعلومات. وخلال جميع مراحل التطوير يجب استخدام أدوات اكتشاف وإصلاح الثغرات الأمنية لضمان اغلاق جميع نقاط الضعف فور اكتشافها.

1. التخطيط

خلال المرحلة الأولى وهي التخطيط، ينبغي على المطورين ومتخصصي أمن المعلومات العمل على استخراج المخاطر المحتملة التي ينبغي التركيز عليها خلال دورة حياة النظام.

1. التحليل وجمع المتطلبات

في المرحلة الثانية، يتم اتخاذ القرارات حول التقنية المستخدمة وإطار العمل ولغة البرمجة. وينبغي على المتخصصين تحديد الثغرات المحتملة في التقنيات والأساليب التي من المقرر استخدامها للتأكد من اتخاذ قرارات أمنية سليمة.

1. بناء النظام والتصميم

في هذه المرحلة ينبغي على الفريق اتباع إرشادات التصميم والتأكد من اغلاق الثغرات المحتملة التي تم تحليلها في المراحل السابقة. عند اغلاق الثغرات في مرحلة متقدمة من تصميم النظام تستطيع التأكد من عدم حدوث أي مشاكل أو تغييرات في مرحلة التطوير. بعض الإجراءات مثل التنبؤ بالمخاطر وتحليل المخاطر في حذه المرحلة يجعل عملية تطوير النظام أسهل وأكثر أمان.

1. التطوير

خلال مرحلة التطوير ينبغي على الفريق استخدام معايير كتابة آمنة. كما ينبغي على المبرمجين عند مراجعة البرنامج والتأكد من أنه كتب بطريقة تضمن تقديم الخصائص المتوقعة أن يعطوا الانتباه للثغرات الأمنية في كتابة البرنامج.

1. الاختبار

خلال مرحلة الاختبار ينبغي استخدام أدوات لأتمته الاختبارات الأمنية مثل SAST and DAST لتحسين أمن النظام.

ينبغي عدم التوقف عن اجراء الاختبارات الأمنية بعد مرحلة التطوير والبقاء على استعداد لإصلاح أي ثغرة لم يتم اكتشافها في المراحل السابقة. كما ينبغي التأكد من اجراء جميع الاعدادات بالطريقة الصحيحة.

1. الصيانة والإصلاح

حتى بعد الانتهاء من مرحلة التطوير والتطبيق، ينبغي تطبيق معايير أمن المعلومات في مرحلة صيانة النظام. يجب التأكد من أن المنتجات محدثه باستمرار للتأكد من خلوها من الثغرات الجديدة.

التركيز على مصادر أمن المعلومات المفتوحة

أحد المخاطر التي ينبغي التركيز عليها هو التأكد من وجود دورة حياة برمجية آمنة هو وجود عناصر مفتوحة المصدر حيث أن المنتجات البرمجية تحتوي على ٦٠-٨٠٪ من النص البرمجي عن طريق مصادر مفتوحة. الاستخدام الدائم لأدوات متابعة الأجزاء مفتوحة المصدر من الممكن يعطي تنبيهات عن بعض المخاطر وطريقة التعامل معها.

ادخال أمن المعلومات في مرحلة مبكرة من دورة حياة البرمجيات

تطبيق ممارسات وأدوات أمن المعلومات بدءًا من المراحل المتقدمة من تطوير النظام والحرض على وجودها في جميع المراحل سيسهم في تقديم منتجات وخدمات آمنة للمستخدم. كما أن الاختبارات المستمرة تسهم في التأكد من استمرارية أمان المنتج من تاريخ الاطلاق.

ادخال أمن المعلومات في دورة حياة البرمجيات ينبغي أن يؤخذ بعين الاعتبار حيث أن معظم الهجمات في الوقت الحالي موجهة على طبقة التطبيقات على نظام OSI مما يجعل الحصول على منتجات آمنة أكثر أهمية.