opentext™ | Cybersecurity

Informe de posicionamiento

Seguridad de aplicaciones impulsada por desarrolladores: Seguridad a la velocidad de DevOps

Índice

El problema actual de la seguridad para aplicaciones

En los últimos 10 años, el software ha pasado de ser una función de asistencia empresarial a ser un centro de innovación y se ha convertido en el diferenciador competitivo más importante para la mayoría de las empresas de cualquier tamaño y sector vertical. Debido a este cambio de importancia respecto al software, las empresas de hoy en día están aumentando drásticamente el número de aplicaciones y la frecuencia de sus versiones. Según el estado de DevOps en 2020 por Puppet, el **46 %** de las organizaciones tienen lanzamientos cada semana o incluso con mayor frecuencia. Además, la complejidad del código sigue aumentando a medida que los desarrolladores intentan satisfacer la demanda empresarial utilizando el código comercial y el código abierto, además de su código personalizado. El informe Estado del sistema de suministro del software de Sonatype indica que, de media, el **80 %** del código de una aplicación proviene de bibliotecas de código abierto. Además, el informe también demostró que, de media, cada aplicación contiene 38 vulnerabilidades conocidas de código abierto. Esta situación tiene enormes implicaciones para los equipos de seguridad a la hora de buscar y gestionar estas vulnerabilidades. Como consecuencia, algunas de las vulneraciones de seguridad más considerables de los últimos años se debieron a vulnerabilidades en los componentes de código de otros fabricantes.

El 90 % de los incidentes de seguridad se deben a exploits contra defectos del diseño o el código del software.

Con las necesidades empresariales como prioridad, proliferan las aplicaciones a través de sitios web, plataformas de redes sociales y aplicaciones móviles y en la nube. Además, algunas aplicaciones están impulsadas por equipos de marketing y creadas con software de otros fabricantes. Normalmente, a estas aplicaciones se les excluye de los procesos empresariales habituales y se las controla poco o nada.

Además de todos los desafíos provocados por el número creciente de aplicaciones, lo cual incrementa la complejidad y la velocidad de lanzamientos, las normativas como el RGPD y la recopilación de datos del cliente con fines comerciales se han convertido en la norma. La multiplicidad de instancias de datos de clientes incrementa la probabilidad y el impacto de las vulneraciones. Esto es especialmente preocupante debido a que la mayoría de las vulneraciones de seguridad actuales se deben a las vulnerabilidades de las aplicaciones. Según el Informe de riesgos de seguridad en aplicaciones de 2019 de nuestro equipo Software Security Research, el 80 % de las aplicaciones contiene al menos una vulnerabilidad crítica o alta, y el 90 % de los incidentes de seguridad se deben a exploits contra defectos del diseño o el código del software.

Estos problemas no van a dejar de crecer

Dado que el tiempo necesario hasta la comercialización sigue siendo vital para un negocio, las organizaciones están adoptando DevOps u otras metodologías ágiles similares para acelerar el desarrollo de su creciente éxito. Todo esto significa que, si la seguridad no se convierte en una parte esencial del ciclo de vida del software, las organizaciones continuarán lanzando aplicaciones con más vulnerabilidades al mercado a una velocidad endiablada.

Motivos por los que las prácticas tradicionales de seguridad para aplicaciones ya no funcionan

En muchas organizaciones, la seguridad de las aplicaciones se limita a un equipo especializado que se involucra en las últimas fases de desarrollo y que suele ser percibido como un inhibidor de velocidad. Estos equipos de seguridad no dan abasto ya que los equipos de desarrollo crecen 80 veces más rápido que ellos. Cuando se encuentran vulnerabilidades de seguridad durante las fases finales, las organizaciones se enfrentan a una considerable presión, que desemboca en encontronazos entre los equipos, retrasos en las fechas de lanzamiento o cosas peores. También se está precipitando la producción de versiones con problemas de seguridad conocidos con el fin de cumplir los plazos del proyecto, de modo que la empresa y los clientes se encuentran expuestos a posibles ataques.

Las mejores prácticas y pruebas de seguridad de aplicaciones deben integrarse a la cadena de herramientas del desarrollador.

Más allá del incumplimiento de plazos y las dinámicas de equipo, el enfoque reactivo respecto a la seguridad para aplicaciones supone un sobrecoste a las organizaciones. Según el NIST, el gasto de la corrección de fallos de seguridad es 30 veces mayor durante la fase de producción y 10 veces mayor durante las pruebas en comparación con su detección durante las fases iniciales del desarrollo. Estos problemas y riesgos potenciales indican que la única manera de proteger las aplicaciones sin poner en riesgo los gastos es expandir la seguridad hacia la izquierda y adoptar un enfoque de seguridad de las aplicaciones impulsado por los desarrolladores.

¿Qué es la seguridad de aplicaciones impulsada por los desarrolladores?

La seguridad de aplicaciones impulsada por los desarrolladores consiste en hacer de la seguridad de las aplicaciones una parte integral del ciclo de vida del software sin crear una carga adicional para los participantes. Tanto si se trata de adoptar un enfoque DevSecOps como de crear un programa de seguridad más eficaz, lo primordial es pensar en la seguridad desde las primeras fases del ciclo de vida. Las mejores prácticas y pruebas de seguridad de aplicaciones deben integrarse a la cadena de herramientas del desarrollador. Cuando se aplican correctamente, esto también significa que no es necesario poner en peligro la seguridad de las aplicaciones para cumplir con los ciclos de lanzamiento tan rápidos que demanda el mercado.

La seguridad de las aplicaciones impulsada por los desarrolladores para su organización

Alcanzar el éxito con la seguridad impulsada por los desarrolladores conlleva tiempo y esfuerzo, pero el mayor obstáculo en el camino es el cambio cultural necesario para incluir la seguridad a lo largo de todo el ciclo de vida del desarrollo del software. Es importante apaciguar la tensión entre los equipos de seguridad y los desarrolladores. Muchas personas creen que los equipos de desarrollo y seguridad tienen prioridades contrapuestas que a menudo se convierten en el mayor obstáculo para el éxito de un programa de seguridad de aplicaciones. Los desarrolladores suelen resistirse a que su organización cree un programa de seguridad de aplicaciones por temor a que se frene la entrega de su código. Esta mentalidad negativa sobre la seguridad se debe a menudo a que los profesionales de la seguridad dictan reglas, flujos de trabajo y herramientas para los desarrolladores en lugar de crear asociaciones sólidas, objetivos comunes y herramientas que se integren perfectamente con la cadena de herramientas de desarrollo.

Al igual que en DevOps, los equipos tienen que acabar con los silos entre ellos, adoptar la transparencia y colaborar. Si bien es más fácil de decir que de hacer, contar con una buena implicación ejecutiva y la presencia de expertos en seguridad en la organización puede ayudar a impulsar esta iniciativa. Más allá del cambio cultural necesario, a continuación se indican algunos pasos importantes para que la transición a la seguridad de aplicaciones impulsada por los desarrolladores tenga éxito:

1 de cada 10 descargas de componentes de código abierto contienen una vulnerabilidad de seguridad conocida.

Paso 1: Desarrollar sin olvidarse de la seguridad

Dado que la relación entre desarrolladores y especialistas en seguridad es de alrededor de 80:1, es imprescindible proporcionar recursos a los desarrolladores para que se responsabilicen de su propio código. Al buscar y solucionar los defectos de seguridad durante el proceso de codificación, los desarrolladores pueden eliminar las posibles vulnerabilidades de seguridad antes de que lleguen a las pruebas y la producción, lo que ahorra tiempo y dinero a la organización. Esta nueva perspectiva requiere formar a los desarrolladores para que codifiquen con la seguridad en mente y proporcionarles las herramientas adecuadas para que puedan recibir comentarios en tiempo real sobre su código. Hay muchas opciones para la formación en seguridad de los desarrolladores, pero las herramientas que proporcionan información de seguridad en tiempo real sobre el código (como el complemento Fortify Security Assistant by OpenText, que actúa de forma muy similar a un corrector ortográfico, proporcionando información de seguridad en tiempo real sobre el código a medida que se desarrolla), o la formación gamificada e integrada de los desarrolladores tales como Secure Code Warrior, facilitan la adopción y aceleran el aprendizaje.

También es importante que los equipos de seguridad ayuden a los desarrolladores compartiendo con ellos información sobre amenazas conocidas, proporcionándoles comentarios, y adoptando la transparencia y la visibilidad en su trabajo. El hecho de que los responsables del desarrollo hayan recibido formación en materia de seguridad para aplicaciones y se hayan asociado con ellos como expertos de seguridad proporciona resultados positivos. De esta forma, los responsables de desarrollo pueden aportar la perspectiva de seguridad durante las primeras fases del ciclo de vida del desarrollo, además de los aspectos funcionales y de calidad tradicionales.

Paso 2: Realizar pruebas pronto, a menudo y rápidamente

Durante el ciclo de vida de desarrollo del software, existen varias estrategias para mantener la velocidad necesaria para no quedarse atrás respecto a la frecuencia actual de lanzamientos. Estas estrategias son realizar pruebas pronto, a menudo y rápidamente.

Realizar pruebas pronto

Las pruebas estáticas de seguridad de la aplicación (SAST) identifican el origen de los problemas de seguridad y ayudan a solucionar los defectos de seguridad subyacentes desde las primeras fases del desarrollo. Para mantener la velocidad de los lanzamientos, los desarrolladores necesitan capacidades inteligentes muy a mano para ser capaces de enviar el código rápida y fácilmente. Fortify Static Code Analyzer by OpenText lidera este método porque:

- Identifica y elimina las vulnerabilidades en código fuente, binario o de bytes.
- Cubre los idiomas que utilizan los desarrolladores con el servicio técnico; es decir, más de 27 idiomas

- Ofrece una detección y solución temprana de los defectos, lo que permite reducir los gastos de corrección.
- Revisa los resultados de los análisis en tiempo real gracias al acceso a recomendaciones y la navegación por las líneas de código para encontrar vulnerabilidades más rápidamente y permitir la auditoría colaborativa.
- Más bien un enfoque "en segundo plano": tener los análisis disponibles en todas partes, incluidos los canales IDE y CI/CD de los desarrolladores.

Fortify Security Assistant by OpenText va un paso más allá al proporcionar a los desarrolladores información y recomendaciones sobre las vulnerabilidades del código en tiempo real, mientras se escribe. Esto no solo sirve de "corrector ortográfico" de seguridad al desarrollador para las vulnerabilidades más comunes, sino que además le permite dejar de cometer esos errores por completo en el futuro.

El análisis más inteligente se refiere a la validación de DAST de los hallazgos de SAST y la sintonización de DAST mediante los resultados de SAST.

Más allá del análisis estático, todavía existe una creciente preocupación sobre las vulnerabilidades conocidas dentro de los componentes de código abierto. Durante casi 10 años, el uso de componentes vulnerables conocidos ha estado en la lista OWASP Top 10. La comunidad DevSecOps ha descubierto recientemente que 1 de cada 10 descargas de componentes de código abierto contienen una vulnerabilidad de seguridad conocida. Se ha producido un aumento del 71 % en las vulneraciones verificadas o sospechosas entre 2014 y 2020, y 1 de cada 5 organizaciones experimentó al menos una vulneración de código abierto en los últimos 12 meses.

Aunque esto es alarmante, muchas organizaciones han estado utilizando Software Composition Analysis para compensar estos riesgos. Sin embargo, priorizar los hallazgos de código abierto sigue siendo un reto importante con Software Composition Analysis. Al igual que con los hallazgos de SAST, la auditoría manual de los hallazgos es un proceso que requiere mucho tiempo y que aumenta el tiempo para solucionar los problemas para los desarrolladores. Según un informe de Sonatype, las organizaciones dedicarán una media de 20 minutos a investigar manualmente un hallazgo de código abierto y la aplicación promedio contiene 38 problemas de código abierto. Debido a que la mayoría de las organizaciones cuentan con cientos o miles de aplicaciones, esto podría suponer miles de horas dedicadas a investigar hallazgos de código abierto que podrían no tener ningún impacto real en la seguridad de su aplicación. Los equipos deben poder centrarse en los problemas que no solo son vulnerables, sino también en aquello susceptible de ser vulnerado.

El análisis de susceptibilidad significa ilustrar rápidamente los componentes vulnerables que se invocan directa o indirectamente y, por lo tanto, son utilizables o "susceptibles". Ser capaz de priorizar los problemas de código abierto ahorra tiempo a la hora de investigar problemas conocidos y aún más tiempo en actualizar una biblioteca que tiene un beneficio de seguridad casi nulo.

En OpenText, nos asociamos con Sonatype para conseguirlo. Fortify recopila los métodos y las firmas de las funciones en función de las solicitudes que se reciben para las indicaciones de Sonatype de componentes conocidos. A medida que Sonatype analiza varios componentes de código abierto, Fortify entiende que para cualquiera de esas vulnerabilidades conocidas particulares que han tenido actualizaciones, lo que significa que se han aplicado parches, Fortify genera una firma para esa función o método para que podamos ver que la función está en su propio código personalizado y que está utilizando ese componente vulnerable de la dependencia. Esto significa que los desarrolladores no solo saben que tienen la dependencia de su ruta de clase, sino que la utilizan de una manera que los hace susceptibles a esta vulnerabilidad en particular.

Realizar pruebas a menudo

Las pruebas dinámicas de seguridad de la aplicación (DAST) simulan ataques contra una aplicación web en funcionamiento para identificar las vulnerabilidades que los exploits podrían aprovechar. Esto proporciona una visión completa de la seguridad de la aplicación al centrarse en aquello susceptible de ser vulnerado y cubrir todos los componentes (servidor, código personalizado, código abierto, servicios, etc.). Al integrar las herramientas DAST en el desarrollo, el control de calidad y la producción, se puede obtener una visión holística de forma continuada. Fortify WebInspect by OpenText ofrece una solución eficaz ya que:

- Identifica rápidamente los riesgos en aplicaciones existentes.
- Automatiza las pruebas dinámicas de seguridad de aplicaciones para cualquier tecnología, desde el desarrollo hasta la producción.
- Cumple con los estándares de conformidad y seguridad con las políticas y los informes preconfigurados para las principales normativas.
- Valida las vulnerabilidades en aplicaciones en funcionamiento, priorizando los problemas más graves en el análisis de causa principal.
- Marcos de trabajo modernos y las API.

Gracias a los análisis automatizados dinámicos o estáticos, puede identificar eficazmente las vulnerabilidades de seguridad en el código fuente, reduciendo la laboriosidad de las evaluaciones de seguridad.

SAST y DAST realmente se complementan entre sí. Al aplicar por niveles el análisis dinámico sobre el análisis estático, los clientes obtienen una valiosa métrica del riesgo adicional que les permite ver una imagen del riesgo real y más completa. Si bien es importante identificar las vulnerabilidades en las primeras etapas del SDLC mediante tecnologías como el análisis estático, es de suma importancia crear bucles de respuesta que puedan identificar cuándo surgen esos hallazgos en entornos de ejecución mediante una exploración DAST.

Una organización que identifica hallazgos como el XSS al principio del SDLC y continúa detectando esos problemas en la producción, puede centrar sus recursos de formación y desarrollo en abordar los problemas sistémicos.

La verdadera integración de SAST y DAST significa que las herramientas SAST y DAST se integran en una única plataforma centrada en el desarrollador con una consola de gestión única. La administración unificada de vulnerabilidades crea bucles de respuesta. Una plataforma unificada de administración de vulnerabilidades no solo es fundamental en cuanto a los flujos de trabajo simplificados de priorización y triaje que introduce, sino también en cuanto a los patrones que se pueden derivar de los datos. El análisis más inteligente se refiere a la validación de DAST de los hallazgos de SAST y la sintonización de DAST mediante los resultados de SAST.

Realizar pruebas rápidamente

Las pruebas interactivas de seguridad de la aplicación (IAST) son un tipo de pruebas de seguridad que combina las DAST con los comentarios del tiempo de ejecución de las aplicaciones probadas a la vez que se realizan las pruebas. Sin embargo, incluso con un enfoque IAST, buscar las vulnerabilidades solo es un tercio del trabajo. Los dos tercios restantes se invierten a menudo en la validación de falsos positivos y la corrección.
Otro argumento en contra de las IAST es el hecho de que este método de prueba suele obviar verdaderos positivos debido a las limitaciones técnicas de este enfoque. Como método alternativo y más eficaz, los algoritmos de aprendizaje automático aplicados y la automatización de auditorías pueden ahorrar tiempo y esfuerzo de auditoría, además de mejorar la precisión del análisis estático.

Fortify Audit Assistant by OpenText es nuestra tecnología de aprendizaje automático. Disponible tanto in situ como en la nube, Audit Assistant utiliza los metadatos del resultado del análisis para predecir y eliminar falsos positivos, y reducir el tiempo de corrección hasta en un 50 %. Un cliente comprobó como 8000 problemas de Java se veían reducidos a 3000 gracias a esta tecnología. Nuestra versión reciente sigue optimizando el proceso para los clientes al añadir la predicción automática en la versión de aplicación para solicitar predicciones automáticamente cuando se añaden nuevos problemas.

Fortify Audit Assistant simplifica la fase de las pruebas de seguridad que consume más tiempo: la auditoría de los resultados del análisis. Fortify Audit Assistant aplica un extensivo conocimiento de seguridad y el aprendizaje automático para automatizar la eliminación de falsos positivos, priorizar los hallazgos e identificar las vulnerabilidades de seguridad relevantes para la organización. Esto implica que, tras iniciar un análisis estático, pueden obtenerse los resultados validados y trasladar a la fase de desarrollo para su corrección.

Paso 3: Utilizar integraciones para integrar la seguridad de la aplicación en su ciclo de vida

La seguridad de las aplicaciones debe integrarse sin problemas a su canal de SDLC y CI/CD para lograr el éxito. Al integrarse en las herramientas que su organización y los desarrolladores utilizan para desarrollar y probar sus aplicaciones, puede encontrar problemas con antelación y de manera frecuente, y solucionarlos como parte de los ciclos de pruebas de desarrollo. Fortify cuenta con un ecosistema de integración que a los desarrolladores les resulta fácil de usar, aprovecha su inversión en las herramientas actuales y reduce la fricción al integrar la seguridad en sus procesos. La seguridad de la aplicación Fortify está incorporada a su proceso de DevOps. La velocidad de DevOps a escala empresarial no significa sacrificar la seguridad y poner en riesgo su negocio.

Fortify aprovecha Swagger en todas nuestras API para proporcionar documentación / referencia automática de la API. La página Fortify GitHub tiene varios proyectos con ejemplos de cómo aprovechar nuestras diversas API para realizar las tareas más solicitadas. La referencia API está incorporada en los productos y se puede acceder a ella a través de la interfaz web de los respectivos productos.

Implantación de software más rápida
Gracias a las opciones de automatización para análisis estáticos y dinámicos, y a la disponibilidad de integraciones para las herramientas de desarrollo más populares (como Visual Studio, Eclipse y Jenkins), los equipos de desarrollo pueden ahorrar tiempo y reducir los malentendidos. Las integraciones con sistemas de gestión de defectos, como JIRA o Bugzilla, mejoran la gestión y la solución de problemas de seguridad, además de garantizar que se puedan gestionar de la misma forma que la organización se encarga de los problemas funcionales. Este eficaz enfoque conlleva un desarrollo y una implantación de software más rápidos, que satisfacen las necesidades empresariales en cuanto a velocidad.

Riesgos reducidos
Al expandir la seguridad hacia la izquierda (primeras fases) para cubrir el ciclo de vida de desarrollo del software de forma integrada y automatizada, las organizaciones reducen sus riesgos y gastos asociados ya que es más barato corregir las vulnerabilidades al principio del proceso. El complemento Fortify Security Assistant y la automatización de los análisis de seguridad impulsados por Jenkins o Azure DevOps ayudan a la organización de desarrollo a realizar las pruebas de seguridad en una etapa temprana del proceso, así como en cualquier momento del mismo.

Retorno de la inversión mejorado
Fortify trabaja con herramientas de desarrollo existentes para proteger su inversión y permite a los equipos de desarrollo el uso continuado de sus herramientas favoritas. Gracias al complemento Fortify Security Assistant, por ejemplo, los desarrolladores no necesitan aprender a usar una herramienta distinta para llevar a cabo análisis de seguridad del código ya que funciona dentro de su IDE existente. Con las integraciones de análisis estático, por otro lado, los análisis de seguridad se realizan como parte del proceso de compilación, y los desarrolladores reciben los problemas de seguridad dentro del sistema de gestión de defectos, sin necesidad de añadir complejidad a las herramientas y procesos existentes.

Paso 4: Automatizar la seguridad como parte de los procesos de desarrollo y pruebas

La automatización del desarrollo, los procesos, la provisión de servidores y el despliegue de aplicaciones es la clave para alcanzar la eficiencia con la iniciativa de DevOps. La automatización permite a las organizaciones desarrollar y lanzar aplicaciones de alta calidad de manera más rápida. Para conseguir la seguridad de las aplicaciones impulsada por los desarrolladores, la automatización se puede utilizar del mismo modo con las pruebas de seguridad para mantener la misma calidad a una velocidad mayor. La automatización consiste en incluir la seguridad como parte de las cadenas de herramientas de DevOps. Esto puede ocurrir en el IDE durante la codificación, en las fases de confirmación, creación y prueba. Este es un énfasis importante de cada programa de seguridad de aplicaciones. Al automatizar las pruebas de seguridad, puede crear y ejecutar pruebas de seguridad automatizadas del mismo modo que lo haría con pruebas de unidades o de integración.

Gracias a los análisis automatizados dinámicos o estáticos, puede identificar eficazmente las vulnerabilidades de seguridad en el código fuente, reduciendo la laboriosidad de las evaluaciones de seguridad. El hecho de contar con un análisis automatizado del código no solo reduce los tiempos de revisión, evaluación de seguridad y pruebas, sino que conlleva un ahorro de gastos de corrección al encontrar antes las vulnerabilidades.

Paso 5: Pensar en el futuro

Con el cambio actual en el que el desarrollo moderno es más dinámico que nunca, con mayor velocidad y complejidad, existe una migración continua a las API, los microservicios, IaC y más. Garantizar la seguridad de este panorama cambiante será cada vez más crucial en los próximos meses y años. Para obtener más información acerca de algunas de estas tendencias y aspectos que se deben considerar para ellas, consulte nuestro Informe de tendencias de seguridad de las aplicaciones de 2021.

Introducción

La seguridad de aplicaciones impulsada por los desarrolladores, integrada a lo largo de todo el ciclo de vida de desarrollo del software, crea procesos controlados y de riesgo reducido, lo que, en última instancia, reduce los gastos, mejora el tiempo de comercialización y optimiza el esfuerzo. Disponer de un itinerario claro hacia la integración y automatización de la seguridad de aplicaciones con KPI cuantificables aumentará las probabilidades de éxito de su organización. La seguridad de las aplicaciones proporciona beneficios más fáciles de demostrar en comparación con otras inversiones en ciberseguridad. La evidencia del progreso realizado y del retorno de la inversión garantizará una inversión continuada en seguridad de aplicaciones.

Estos son algunos de los aspectos importantes que debe tener en cuenta a la hora de trazar la hoja de ruta de ese proceso.

- Identifique a sus expertos en seguridad de aplicaciones.
- Desarrolle su estrategia y sus procesos principales antes de la implantación.
  - Defina el alcance inicial y las métricas clave, como: Las aplicaciones y los equipos de desarrollo con los que empezar
  - La posibilidad de emplear las SAST, DAST o ambas
  - Qué integraciones utilizar
  - La posibilidad de usar herramientas de seguridad in situ, a pedido, o un enfoque híbrido
  - Cuáles son las mejoras esperadas en un plazo de 12 meses respecto a la línea de base
- Encuentre las herramientas adecuadas para su organización.

Así, medir su éxito es crucial. Tener unos KPI adecuados permite que su organización no solo mida eficazmente la situación que tiene en materia de seguridad, sino que justifique el gasto y la inversión continua en su programa de seguridad. Los KPI deben ajustarse a los objetivos de la empresa/programa. A continuación, se indican algunos que se deben tener en cuenta:

- **Tendencia de riesgo ponderado**: representación del riesgo basada en el negocio de los defectos de seguridad de aplicaciones web evaluados durante un periodo de tiempo especificado o casos repetidos del desarrollo de aplicaciones.
- **Ventana de corrección de defectos de seguridad**: tiempo transcurrido desde que se identifica un defecto de seguridad en una aplicación web verificada hasta que se verifica su cierre. Se puede hacer referencia a esto como tiempo medio de espera hasta la corrección (MTTR).
- **Tasa de recurrencia de defectos de seguridad**: tasa, a lo largo del tiempo, en la que los defectos de seguridad de aplicaciones web previamente cerrados se vuelven a introducir en una aplicación, organización u otra unidad lógica determinada.
- **Relación de seguridad respecto a los defectos de calidad**: relación entre los defectos de seguridad y el número total de defectos de calidad del software que se generan (funcional + rendimiento + seguridad).

¿Por qué OpenText?

Las personas, los procesos y la tecnología son los componentes esenciales de la seguridad de aplicaciones por los desarrolladores. OpenText tiene la experiencia y los recursos; así como la tecnología, las personas y los procesos (a través de Fortify on Demand y servicios profesionales) para ayudarle en cada paso del proceso.

OpenText proporciona una solución de seguridad de aplicaciones integral y flexible con modelos in situ, a pedido e híbridos. Con beneficios cuantificables, como el tiempo de comercialización un 30 % más rápido, un 95 % menos de positivos, análisis entre un 10 y un 15 % más rápidos, soluciones 10 veces más rápidas y la detección del doble de vulnerabilidades, Fortify sigue siendo líder en el sector de las herramientas de seguridad de aplicaciones.

Elija Fortify por:
Su fácil puesta en marcha: puede iniciarse en tan solo un día con Fortify on Demand.

Su facilidad de uso e intuitiva integración en los procesos existentes: Fortify se integra fácilmente en el entorno que a sus desarrolladores les resulta familiar, haciendo de la seguridad un añadido natural para sus herramientas y procesos.

Sus capacidades de velocidad, automatización y ampliación: la mayoría de los análisis de Fortify se completan en cuestión de minutos. Además, puede procesar los datos brutos de los análisis para obtener resultados de auditoría con funciones asistidas por ordenador también en pocos minutos. Los análisis automatizados pueden iniciarse como parte de las comprobaciones de código, las comprobaciones, las compilaciones, las actualizaciones u otros componentes de los canales de la integración continua y la implantación continua (CI/CD). Los clientes de Fortify disfrutan de una gran facilidad de ampliación in situ al usar técnicas de análisis centralizado, Fortify on Demand o un enfoque híbrido.

Su precisión y cobertura en cuanto a lenguajes de programación: los usuarios de Fortify experimentan un número mayor de verdaderos positivos (más hallazgos validados) y menos falsos positivos (menos ruido) en comparación con otros productos. Fortify ofrece la cobertura de lenguajes de programación más amplia (27 compatibles) desde mayo de 2021.

Su reconocimiento unánime en el sector: Fortify se considera una de las mejores herramientas en materia de seguridad de aplicaciones desde los últimos 15 años. Reconocido como líder en el Gartner Magic Quadrant for Application Security por octavo año consecutivo. Fortify cuenta con la confianza de las empresas más importantes de numerosos mercados verticales de todo el mundo.

Cree software seguro con rapidez con la ayuda de Fortify con estas características clave:

- El **complemento Fortify Security Assistant** proporciona un análisis de seguridad del código en tiempo real. Solucione cada problema con la confianza de saber que solo se marcan los problemas importantes.
- Las **plantillas GitHub Actions y GitLab CI** permiten integrar y automatizar las pruebas de seguridad de las aplicaciones estáticas (SAST) en sus flujos de trabajo de CI/CD.

Conecte con nosotros en
www.opentext.com

- El **análisis de susceptibilidad** permite a los desarrolladores o profesionales de la seguridad comprobar si alguien ha invocado una vulnerabilidad en su código personalizado. Lo que es más importante, pueden comprobar si la entrada que controla el atacante alcanza la función del código.
- **Speed Dial** en Fortify SCA permite a los desarrolladores controlar mejor la profundidad y la velocidad de sus análisis estáticos.
- **Commit Scan** ofrece a los desarrolladores análisis automatizados y ligeros en su flujo de trabajo, integrando pruebas estáticas en el proceso de confirmación de Git, proporcionándoles información inmediata sobre el código que se está comprobando en GitHub, GitLab y Bitbucket.
- **Fortify Audit Assistant** minimiza la carga de trabajo del auditor con el aprendizaje automático para identificar las vulnerabilidades a partir de los resultados de Fortify SCA. De este modo, se reducen los problemas que requieren un examen manual profundo.
- **Smart View** en Audit Workbench ayuda a los desarrolladores a comprender rápidamente cómo se relacionan varios problemas desde la perspectiva del flujo de datos, con la posibilidad de clasificar los problemas de seguridad y después solucionarlos en el punto más eficiente.

opentext™ | Cybersecurity

OpenText Cybersecurity ofrece soluciones integrales de seguridad para empresas y socios de todos los tamaños. Desde la prevención, la detección y la respuesta a la recuperación, la investigación y el cumplimiento, nuestra plataforma unificada integral ayuda a los clientes a crear resiliencia cibernética con ayuda de una cartera de seguridad integral. Gracias a los conocimientos prácticos de nuestra inteligencia de amenazas contextual y en tiempo real, los clientes de OpenText Cybersecurity se benefician de productos de alta eficacia, una experiencia compatible y seguridad simplificada para ayudar a administrar el riesgo empresarial.

762-ES0040-002 | O | 12/23 | © 2023 Open Text